

**RM0091**

リファレンスマニュアル

STM32F0x1/STM32F0x2/STM32F0x8
advanced ARM[®]-based 32-bit MCUs

概要

このリファレンスマニュアルは、アプリケーション開発者を対象としています。STM32F0x1/STM32F0x2/STM32F0x8 マイクロコントローラメモリとペリフェラルを使用する方法について、詳しく説明しています。

STM32F031x4/x6、STM32F051x4/x6/x8、STM32F071x8/xB、STM32F091xB/xC、STM32F042x4/x6、STM32F072x8/xB、STM32F038x6、STM32F048x6、STM32F058x8、STM32F078xB、および STM32F098xC デバイスに適用されます。

このマニュアルでは、STM32F0x1/STM32F0x2/STM32F0x8 マイクロコントローラを「STM32F0xx」と呼びます。

STM32F0xx は、さまざまなメモリサイズ、パッケージ、およびペリフェラルを持つマイクロコントローラファミリ製品になります。

注文情報、機械的および電気的特性については、対応するデータシートを参照してください。

ARM[®] CORTEX[®]-M0 コアについては、Cortex[®]-M0 テクニカルリファレンスマニュアルを参照してください。

関連ドキュメント

- Cortex[®]-M0 テクニカルリファレンスマニュアル (<http://infocenter.arm.com> から入手可能)。
- STM32F0xx Cortex-M0 プログラミングマニュアル (PM0215)
- STM32F0xx データシート (STMicroelectronics のウェブサイト www.st.com から入手可能)。

目次

1	このマニュアルにおける表記の規則	42
1.1	レジスタに関する略記	42
1.2	用語	42
1.3	使用可能なペリフェラル	43
2	システムおよびメモリの概要	44
2.1	システムアーキテクチャ	44
2.2	メモリ構成	46
2.2.1	概要	46
2.2.2	メモリマップとレジスタ境界アドレス	47
2.3	内蔵 SRAM	51
2.4	フラッシュメモリの概要	52
2.5	ブート設定	53
3	内蔵フラッシュメモリ	55
3.1	フラッシュの主な機能	55
3.2	フラッシュメモリの機能詳細	55
3.2.1	フラッシュメモリの構成	55
3.2.2	フラッシュのプログラムおよび消去操作	58
3.3	メモリ保護	65
3.3.1	読み出し保護	65
3.3.2	書き込み保護	67
3.3.3	オプションバイトの書き込み保護	67
3.4	フラッシュ割り込み	68
3.5	フラッシュレジスタの説明	68
3.5.1	フラッシュアクセス制御レジスタ (FLASH_ACR)	68
3.5.2	フラッシュキーレジスタ (FLASH_KEYR)	69
3.5.3	フラッシュオプションキーレジスタ (FLASH_OPTKEYR)	69
3.5.4	フラッシュステータスレジスタ (FLASH_SR)	70
3.5.5	フラッシュ制御レジスタ (FLASH_CR)	70
3.5.6	フラッシュアドレスレジスタ (FLASH_AR)	72
3.5.7	フラッシュオプションバイトレジスタ (FLASH_OBR)	72
3.5.8	書き込み保護レジスタ (FLASH_WRP)	73

3.5.9	フラッシュレジスタマップ	74
4	オプションバイト	75
4.1	オプションバイトの説明	76
4.1.1	ユーザオプションバイトおよび読み出し保護オプションバイト	76
4.1.2	ユーザデータオプションバイト	77
4.1.3	書き込み保護オプションバイト	78
4.1.4	オプションバイトマップ	79
5	巡回冗長検査計算ユニット (CRC)	80
5.1	概要	80
5.2	CRC の主な機能	80
5.3	CRC の機能説明	81
5.4	CRC レジスタ	82
5.4.1	データレジスタ (CRC_DR)	82
5.4.2	独立型データレジスタ (CRC_IDR)	83
5.4.3	制御レジスタ (CRC_CR)	83
5.4.4	CRC の初期値 (CRC_INIT)	84
5.4.5	CRC 多項式 (CRC_POL)	84
5.4.6	CRC レジスタマップ	85
6	電源制御 (PWR)	86
6.1	電源	86
6.1.1	独立した A/D および D/A コンバータ用電源と基準電圧	87
6.1.2	独立 I/O 電源レール	87
6.1.3	バッテリーバックアップドメイン	87
6.1.4	電圧レギュレータ	88
6.2	電源供給スーパーバイザ	88
6.2.1	パワーオンリセット (POR) / パワーダウンリセット (PDR)	88
6.2.2	プログラム可能な電圧検出器 (PVD)	89
6.3	低電力モード	90
6.3.1	システムクロックの低速化	91
6.3.2	ペリフェラルクロックゲーティング	92
6.3.3	SLEEP モード	92
6.3.4	STOP モード	93
6.3.5	STANDBY モード	95

6.3.6	低電力モードからの自動ウェイクアップ	96
6.4	電源制御レジスタ	97
6.4.1	電源制御レジスタ (PWR_CR)	97
6.4.2	電源制御/ステータスレジスタ (PWR_CSR)	98
6.4.3	PWR レジスタマップ	99
7	リセットおよびクロック制御 (RCC)	100
7.1	リセット	100
7.1.1	電源リセット	100
7.1.2	システムリセット	100
7.1.3	RTC ドメインリセット	102
7.2	クロック	102
7.2.1	HSE クロック	106
7.2.2	HSI クロック	107
7.2.3	HSI48 クロック	108
7.2.4	PLL	109
7.2.5	LSE クロック	109
7.2.6	LSI クロック	110
7.2.7	システムクロック (SYSCLK) の選択	110
7.2.8	クロックセキュリティシステム (CSS)	110
7.2.9	ADC クロック	111
7.2.10	RTC クロック	111
7.2.11	独立型ウォッチドッグクロック	111
7.2.12	クロック信号出力	111
7.2.13	TIM14 を使用した内部/外部クロックの測定	112
7.3	低電力モード	113
7.4	RCC レジスタ	115
7.4.1	クロック制御レジスタ (RCC_CR)	115
7.4.2	クロック設定レジスタ (RCC_CFGR)	117
7.4.3	クロック割り込みレジスタ (RCC_CIR)	120
7.4.4	APB ペリフェラルリセットレジスタ 2 (RCC_APB2RSTR)	123
7.4.5	APB ペリフェラルリセットレジスタ 1 (RCC_APB1RSTR)	124
7.4.6	AHB ペリフェラルクロック有効レジスタ (RCC_AHBENR)	127
7.4.7	APB ペリフェラルクロック有効レジスタ 2 (RCC_APB2ENR)	128
7.4.8	APB ペリフェラルクロック有効レジスタ 1 (RCC_APB1ENR)	130
7.4.9	RTC ドメイン制御レジスタ (RCC_BDCR)	133
7.4.10	制御/ステータスレジスタ (RCC_CSR)	135

7.4.11	AHB ペリフェラルリセットレジスタ (RCC_AHBSTR)	136
7.4.12	クロック設定レジスタ 2 (RCC_CFGR2)	138
7.4.13	クロック設定レジスタ 3 (RCC_CFGR3)	139
7.4.14	クロック制御レジスタ 2 (RCC_CR2)	140
7.4.15	RCC レジスタマップ	142
8	クロックリカバリシステム (CRS)	144
8.1	概要	144
8.2	CRS の主な機能	144
8.3	CRS の機能詳細	145
8.3.1	CRS ブロック図	145
8.3.2	同期入力	145
8.3.3	周波数誤差測定	146
8.3.4	周波数誤差評価および自動トリミング	147
8.3.5	CRS の初期化と設定	147
8.4	CRS 低電力モード	148
8.5	CRS 割り込み	148
8.6	CRS レジスタ	149
8.6.1	CRS 制御レジスタ (CRS_CR)	149
8.6.2	CRS 設定レジスタ (CRS_CFGR)	151
8.6.3	CRS 割り込みおよびステータスレジスタ (CRS_ISR)	152
8.6.4	CRS 割り込みフラグクリアレジスタ (CRS_ICR)	154
8.6.5	CRS レジスタマップ	155
9	汎用 I/O (GPIO)	156
9.1	概要	156
9.2	GPIO の主な機能	156
9.3	GPIO の機能説明	156
9.3.1	汎用 I/O (GPIO)	158
9.3.2	I/O ピンオルタネート機能マルチプレクサと配置	159
9.3.3	I/O ポート制御レジスタ	159
9.3.4	I/O ポートデータレジスタ	160
9.3.5	I/O データのビット単位の操作	160
9.3.6	GPIO ロック機構	160
9.3.7	I/O オルタネート機能の入力／出力	161
9.3.8	外部割り込み／ウェイクアップライン	161

9.3.9	入力設定	161
9.3.10	出力設定	162
9.3.11	オルタネート機能設定	162
9.3.12	アナログ設定	163
9.3.13	HSE または LSE オシレータのピンを GPIO として使用	164
9.3.14	GPIO ピンを RTC 供給ドメインで使用	164
9.4	GPIO レジスタ	165
9.4.1	GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ F)	165
9.4.2	GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A ~ F)	165
9.4.3	GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A ~ F)	166
9.4.4	GPIO ポートプルアップ/プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ F)	166
9.4.5	GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ F)	167
9.4.6	GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A ~ F)	167
9.4.7	GPIO ポートビットセット/リセットレジスタ (GPIOx_BSRR) (x = A ~ F)	168
9.4.8	GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ B)	168
9.4.9	GPIO オルタネート機能下位レジスタ (GPIOx_AFR1) (x = A ~ F)	169
9.4.10	GPIO オルタネート機能上位レジスタ (GPIOx_AFR2) (x = A ~ F)	170
9.4.11	GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A ~ F)	170
9.4.12	GPIO レジスタマップ	171
10	システム設定コントローラ (SYSCFG)	173
10.1	SYSCFG レジスタ	173
10.1.1	SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1)	173
10.1.2	SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1)	177
10.1.3	SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2)	177
10.1.4	SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG_EXTICR3)	178
10.1.5	SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4)	179
10.1.6	SYSCFG 設定レジスタ 2 (SYSCFG_CFGR2)	180
10.1.7	SYSCFG 割り込みライン 0 ステータスレジスタ (SYSCFG_ITLINE0)	181
10.1.8	SYSCFG 割り込みライン 1 ステータスレジスタ (SYSCFG_ITLINE1)	181

10.1.9	SYSCFG 割り込みライン 2 ステータスレジスタ (SYSCFG_ITLINE2)	182
10.1.10	SYSCFG 割り込みライン 3 ステータスレジスタ (SYSCFG_ITLINE3)	182
10.1.11	SYSCFG 割り込みライン 4 ステータスレジスタ (SYSCFG_ITLINE4)	183
10.1.12	SYSCFG 割り込みライン 5 ステータスレジスタ (SYSCFG_ITLINE5)	183
10.1.13	SYSCFG 割り込みライン 6 ステータスレジスタ (SYSCFG_ITLINE6)	183
10.1.14	SYSCFG 割り込みライン 7 ステータスレジスタ (SYSCFG_ITLINE7)	184
10.1.15	SYSCFG 割り込みライン 8 ステータスレジスタ (SYSCFG_ITLINE8)	184
10.1.16	SYSCFG 割り込みライン 9 ステータスレジスタ (SYSCFG_ITLINE9)	185
10.1.17	SYSCFG 割り込みライン 10 ステータスレジスタ (SYSCFG_ITLINE10)	185
10.1.18	SYSCFG 割り込みライン 11 ステータスレジスタ (SYSCFG_ITLINE11)	186
10.1.19	SYSCFG 割り込みライン 12 ステータスレジスタ (SYSCFG_ITLINE12)	186
10.1.20	SYSCFG 割り込みライン 13 ステータスレジスタ (SYSCFG_ITLINE13)	187
10.1.21	SYSCFG 割り込みライン 14 ステータスレジスタ (SYSCFG_ITLINE14)	187
10.1.22	SYSCFG 割り込みライン 15 ステータスレジスタ (SYSCFG_ITLINE15)	188
10.1.23	SYSCFG 割り込みライン 16 ステータスレジスタ (SYSCFG_ITLINE16)	188
10.1.24	SYSCFG 割り込みライン 17 ステータスレジスタ (SYSCFG_ITLINE17)	188
10.1.25	SYSCFG 割り込みライン 18 ステータスレジスタ (SYSCFG_ITLINE18)	189
10.1.26	SYSCFG 割り込みライン 19 ステータスレジスタ (SYSCFG_ITLINE19)	189
10.1.27	SYSCFG 割り込みライン 20 ステータスレジスタ (SYSCFG_ITLINE20)	189
10.1.28	SYSCFG 割り込みライン 21 ステータスレジスタ (SYSCFG_ITLINE21)	190
10.1.29	SYSCFG 割り込みライン 22 ステータスレジスタ (SYSCFG_ITLINE22)	190
10.1.30	SYSCFG 割り込みライン 23 ステータスレジスタ (SYSCFG_ITLINE23)	190
10.1.31	SYSCFG 割り込みライン 24 ステータスレジスタ (SYSCFG_ITLINE24)	191
10.1.32	SYSCFG 割り込みライン 25 ステータスレジスタ (SYSCFG_ITLINE25)	191
10.1.33	SYSCFG 割り込みライン 26 ステータスレジスタ (SYSCFG_ITLINE26)	191
10.1.34	SYSCFG 割り込みライン 27 ステータスレジスタ (SYSCFG_ITLINE27)	192
10.1.35	SYSCFG 割り込みライン 28 ステータスレジスタ (SYSCFG_ITLINE28)	192
10.1.36	SYSCFG 割り込みライン 29 ステータスレジスタ (SYSCFG_ITLINE29)	193
10.1.37	SYSCFG 割り込みライン 30 ステータスレジスタ (SYSCFG_ITLINE30)	193
10.1.38	SYSCFG レジスタマップ	194

11 **ダイレクトメモリアクセスコントローラ (DMA) 197**

11.1	概要	197
11.2	DMA の主な機能	197
11.3	DMA の機能説明	198
11.3.1	DMA トランザクション	198
11.3.2	アービタ	199



11.3.3	DMA チャンネル	199
11.3.4	プログラム可能なデータ幅、データの整列、およびエンディアン	201
11.3.5	エラー管理	202
11.3.6	DMA 割り込み	202
11.3.7	DMA リクエストマッピング	203
11.4	DMA レジスタ	208
11.4.1	DMA 割り込みステータスレジスタ (DMA_ISR および DMA2_ISR)	208
11.4.2	DMA 割り込みフラグクリアレジスタ (DMA_IFCR および DMA2_IFCR)	209
11.4.3	DMA チャンネル x 設定レジスタ (DMA_CCRx および DMA2_CCRx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。 ここで x = チャンネル数)	210
11.4.4	DMA チャンネル x データ数レジスタ (DMA_CNDTRx および DMA2_CNDTRx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。x = チャンネル番号)	212
11.4.5	DMA チャンネル x ペリフェラルアドレスレジスタ (DMA_CPARx および DMA2_CPARx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。 x = チャンネル番号)	212
11.4.6	DMA チャンネル x メモリアドレスレジスタ (DMA_CMARx および DMA2_CMARx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。 x = チャンネル番号)	213
11.4.7	DMA チャンネル選択レジスタ (DMA_CSELR および DMA2_CSELR)	214
11.4.8	DMA レジスタマップ	215
12	割り込みとイベント	217
12.1	ネスト化されたベクタ割り込みコントローラ (NVIC)	217
12.1.1	NVIC の主な機能	217
12.1.2	SysTick 較正值レジスタ	217
12.1.3	割り込みベクタと例外ベクタ	217
12.2	拡張割り込み/イベントコントローラ (EXTI)	219
12.2.1	主な特長	219
12.2.2	ブロック図	220
12.2.3	イベント管理	220
12.2.4	機能詳細	221
12.2.5	外部および内部の割り込み/イベントラインの配置	222
12.3	EXTI レジスタ	223
12.3.1	割り込みマスクレジスタ (EXTI_IMR)	223
12.3.2	イベントマスクレジスタ (EXTI_EMR)	224
12.3.3	立ち上がりトリガ選択レジスタ (EXTI_RTSR)	224
12.3.4	立ち下がりトリガ選択レジスタ (EXTI_FTSR)	225

12.3.5	ソフトウェア割り込みイベントレジスタ (EXTI_SWIER)	226
12.3.6	ペンディングレジスタ (EXTI_PR)	227
12.3.7	EXTI レジスタマップ	228
13	アナログデジタルコンバータ (ADC)	229
13.1	概要	229
13.2	ADC の主な機能	230
13.3	ADC ピンと内部信号	231
13.4	ADC の機能詳細	232
13.4.1	較正 (ADCAL)	232
13.4.2	ADC オン / オフ制御 (ADEN、ADDIS、ADRDY)	233
13.4.3	ADC クロック (CKMODE)	235
13.4.4	ADC の設定	236
13.4.5	チャンネル選択 (CHSEL、SCANDIR)	237
13.4.6	プログラム可能なサンプリング時間 (SMP)	237
13.4.7	シングル変換モード (CONT=0)	238
13.4.8	連続変換モード (CONT=1)	238
13.4.9	変換の開始 (ADSTART)	239
13.4.10	タイミング	239
13.4.11	実行中の変換の停止 (ADSTP)	240
13.5	外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN)	241
13.5.1	不連続モード (DISCEN)	242
13.5.2	プログラム可能な分解能 (RES) - 高速変換モード	242
13.5.3	変換の終了、サンプリングフェーズの終了 (EOC、EOSMP フラグ)	243
13.5.4	変換シーケンスの終了 (EOSEQ フラグ)	243
13.5.5	タイミング図の例 (シングル / 連続モードハードウェア / ソフトウェアトリガ)	244
13.6	データ管理	246
13.6.1	データレジスタおよびデータの配置 (ADC_DR、ALIGN)	246
13.6.2	ADC オーバーラン (OVR、OVRMOD)	246
13.6.3	DMA を使用しない変換データシーケンスの管理	247
13.6.4	オーバーランなしでの DMA を使用しない変換データの管理	247
13.6.5	DMA を使用した変換データの管理	247
13.7	低電力機能	249
13.7.1	ウェイトモード変換	249
13.7.2	オートオフモード (AUTOFF)	249

13.8	アナログウィンドウウォッチドッグ (AWDEN、AWDSGL、AWDCH、AWD_HTR/LTR、AWD)	251
13.9	温度センサと内部基準電圧	252
13.10	バッテリー電圧監視	254
13.11	ADC 割り込み	254
13.12	ADC レジスタ	255
13.12.1	ADC 割り込みおよびステータスレジスタ (ADC_ISR)	255
13.12.2	ADC 割り込み有効レジスタ (ADC_IER)	256
13.12.3	ADC 制御レジスタ (ADC_CR)	258
13.12.4	ADC 設定レジスタ 1 (ADC_CFGR1)	260
13.12.5	ADC 設定レジスタ 2 (ADC_CFGR2)	264
13.12.6	ADC サンプリング時間レジスタ (ADC_SMPR)	264
13.12.7	ADC ウォッチドッグ閾値レジスタ (ADC_TR)	265
13.12.8	ADC チャンネル選択レジスタ (ADC_CHSELR)	266
13.12.9	ADC データレジスタ (ADC_DR)	266
13.12.10	ADC 共通設定レジスタ (ADC_CCR)	267
13.12.11	ADC レジスタマップ	268
14	D/A コンバータ (DAC)	269
14.1	概要	269
14.2	DAC の主な機能	269
14.3	シングルモードの機能説明	271
14.3.1	DAC チャンネルイネーブル	271
14.3.2	DAC 出力バッファイネーブル	271
14.3.3	DAC データフォーマット	271
14.3.4	DAC チャンネル変換	272
14.3.5	DAC 出力電圧	273
14.3.6	DAC トリガ選択	273
14.4	デュアルモードの機能詳細 (STM32F07x および STM32F09x デバイス)	274
14.4.1	DAC チャンネルイネーブル	274
14.4.2	DAC 出力バッファイネーブル	274
14.4.3	DAC データフォーマット	274
14.4.4	デュアルモードでの DAC チャンネル変換	275
14.4.5	デュアル変換モードの説明	275
14.4.6	DAC 出力電圧	279
14.4.7	DAC トリガ選択	280

14.5	ノイズ生成 (STM32F07x および STM32F09x デバイス)	281
14.6	三角波生成 (STM32F07x および STM32F09x デバイス)	282
14.7	DMA リクエスト	283
14.8	DAC レジスタ	284
14.8.1	DAC 制御レジスタ (DAC_CR)	284
14.8.2	DAC ソフトウェアトリガレジスタ (DAC_SWTRIGR)	288
14.8.3	DAC チャンネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R1)	288
14.8.4	DAC チャンネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L1)	289
14.8.5	DAC チャンネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R1)	289
14.8.6	DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2)	289
14.8.7	DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L2)	290
14.8.8	DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R2)	290
14.8.9	デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD)	291
14.8.10	デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD)	291
14.8.11	デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD)	292
14.8.12	DAC チャンネル 1 データ出力レジスタ (DAC_DOR1)	292
14.8.13	DAC チャンネル 2 データ出力レジスタ (DAC_DOR2)	293
14.8.14	DAC ステータスレジスタ (DAC_SR)	293
14.8.15	DAC レジスタマップ	294
15	コンパレータ (COMP)	296
15.1	概要	296
15.2	COMP の主な機能	296
15.3	COMP の機能説明	297
15.3.1	COMP ブロック図	297
15.3.2	COMP ピンおよび内部信号	297
15.3.3	COMP のリセットおよびクロック	298
15.3.4	コンパレータのロック機構	298
15.3.5	ヒステリシス	298
15.3.6	電力モード	299

15.4	COMP 割り込み	299
15.5	COMP レジスタ	300
15.5.1	COMP 制御およびステータスレジスタ (COMP_CSR)	300
15.5.2	COMP レジスタマップ	303
16	高機能制御タイマ (TIM1)	304
16.1	TIM1 の概要	304
16.2	TIM1 の主な特長	304
16.3	TIM1 の機能詳細	306
16.3.1	タイムベースユニット	306
16.3.2	カウンタモード	308
16.3.3	繰り返しカウンタ	318
16.3.4	クロックソース	320
16.3.5	キャプチャ／比較チャネル	323
16.3.6	入力キャプチャモード	326
16.3.7	PWM 入力モード	327
16.3.8	強制出力モード	328
16.3.9	出力比較モード	328
16.3.10	PWM モード	330
16.3.11	相補出力とデッドタイム挿入	333
16.3.12	ブレーク機能の使用	335
16.3.13	外部イベントによる OCxREF 信号のクリア	338
16.3.14	6 ステップ PWM 生成	339
16.3.15	ワンパルスモード	340
16.3.16	エンコーダインタフェースモード	341
16.3.17	タイマ入力 XOR 機能	343
16.3.18	ホールセンサとのインタフェース	344
16.3.19	TIMx と外部トリガの同期	346
16.3.20	タイマの同期	349
16.3.21	デバッグモード	349
16.4	TIM1 レジスタ	350
16.4.1	TIM1 制御レジスタ 1 (TIM1_CR1)	350
16.4.2	TIM1 制御レジスタ 2 (TIM1_CR2)	351
16.4.3	TIM1 のスレーブモード制御レジスタ (TIM1_SMCR)	353
16.4.4	TIM1 DMA／割り込み有効レジスタ (TIM1_DIER)	355
16.4.5	TIM1 ステータスレジスタ (TIM1_SR)	357
16.4.6	TIM1 のイベント生成レジスタ (TIM1_EGR)	358

16.4.7	TIM1 のキャプチャ／比較モードレジスタ 1 (TIM1_CCMR1)	360
16.4.8	TIM1 のキャプチャ／比較モードレジスタ 2 (TIM1_CCMR2)	363
16.4.9	TIM1 のキャプチャ／比較有効レジスタ (TIM1_CCER)	364
16.4.10	TIM1 のカウンタ (TIM1_CNT)	368
16.4.11	TIM1 のプリスケラ (TIM1_PSC)	368
16.4.12	TIM1 の自動再ロードレジスタ (TIM1_ARR)	368
16.4.13	TIM1 繰り返しカウンタレジスタ (TIM1_RCR)	369
16.4.14	TIM1 のキャプチャ／比較レジスタ 1 (TIM1_CCR1)	369
16.4.15	TIM1 のキャプチャ／比較レジスタ 2 (TIM1_CCR2)	370
16.4.16	TIM1 のキャプチャ／比較レジスタ 3 (TIM1_CCR3)	370
16.4.17	TIM1 のキャプチャ／比較レジスタ 4 (TIM1_CCR4)	371
16.4.18	TIM1 ブレークおよびデッドタイムレジスタ (TIM1_BDTR)	371
16.4.19	TIM1 DMA 制御レジスタ (TIM1_DCR)	373
16.4.20	完全転送の TIM1 DMA アドレス (TIM1_DMAR)	374
16.4.21	TIM1 レジスタマップ	375
17	汎用タイマ (TIM2 および TIM3)	377
17.1	TIM2 および TIM3 の概要	377
17.2	TIM2 および TIM3 の主な機能	377
17.3	TIM2 および TIM3 の機能説明	379
17.3.1	タイムベースユニット	379
17.3.2	カウンタモード	381
17.3.3	クロックソース	392
17.3.4	キャプチャ／比較チャネル	396
17.3.5	入力キャプチャモード	398
17.3.6	PWM 入力モード	400
17.3.7	強制出力モード	401
17.3.8	出力比較モード	401
17.3.9	PWM モード	402
17.3.10	ワンパルスモード	405
17.3.11	外部イベントによる OCxREF 信号のクリア	407
17.3.12	エンコーダインタフェースモード	407
17.3.13	タイマ入力 XOR 機能	409
17.3.14	タイマと外部トリガの同期	410
17.3.15	タイマの同期	414
17.3.16	デバッグモード	419

17.4	TIM2 および TIM3 レジスタ	420
17.4.1	TIM2 および TIM3 制御レジスタ 1 (TIM2_CR1 および TIM3_CR1)	420
17.4.2	TIM2 および TIM3 制御レジスタ 2 (TIM2_CR2 および TIM3_CR2)	422
17.4.3	TIM2 および TIM3 のスレーブモード制御レジスタ (TIM2_SMCR および TIM3_SMCR)	423
17.4.4	TIM2 および TIM3 DMA/ 割り込み有効レジスタ (TIM2_DIER および TIM3_DIER)	426
17.4.5	TIM2 および TIM3 制御レジスタ (TIM2_SR および TIM3_SR)	427
17.4.6	TIM2 および TIM3 のイベント生成レジスタ (TIM2_EGR および TIM3_EGR)	429
17.4.7	TIM2 および TIM3 のキャプチャ/比較モードレジスタ 1 (TIM2_CCMR1 および TIM3_CCMR1)	430
17.4.8	TIM2 および TIM3 のキャプチャ/比較モードレジスタ 2 (TIM2_CCMR2 および TIM3_CCMR2)	433
17.4.9	TIM2 および TIM3 のキャプチャ/比較有効レジスタ (TIM2_CCER および TIM3_CCER)	434
17.4.10	TIM2 および TIM3 のカウンタ (TIM2_CNT および TIM3_CNT)	436
17.4.11	TIM2 および TIM3 のプリスケラ (TIM2_PSC および TIM3_PSC)	436
17.4.12	TIM2 および TIM3 自動再ロードレジスタ (TIM2_ARR および TIM3_ARR)	437
17.4.13	TIM2 および TIM3 のキャプチャ/比較レジスタ 1 (TIM2_CCR1 および TIM3_CCR1)	437
17.4.14	TIM2 および TIM3 のキャプチャ/比較レジスタ 2 (TIM2_CCR2 および TIM3_CCR2)	438
17.4.15	TIM2 および TIM3 のキャプチャ/比較レジスタ 3 (TIM2_CCR3 および TIM3_CCR3)	438
17.4.16	TIM2 および TIM3 のキャプチャ/比較レジスタ 4 (TIM2_CCR4 および TIM3_CCR4)	439
17.4.17	TIM2 および TIM3 DMA 制御レジスタ (TIM2_DCR および TIM3_DCR)	440
17.4.18	完全転送の TIM2 および TIM3 DMA アドレス (TIM2_DMAR および TIM3_DMAR)	441
17.4.19	TIM2 および TIM3 のレジスタマップ	442
18	汎用タイマ (TIM14)	444
18.1	TIM14 の概要	444
18.2	TIM14 の主な機能	444
18.3	TIM14 の機能詳細	445
18.3.1	タイムベースユニット	445
18.3.2	カウンタモード	447
18.3.3	クロックソース	450

18.3.4	キャプチャ／比較チャネル	450
18.3.5	入力キャプチャモード	452
18.3.6	強制出力モード	453
18.3.7	出力比較モード	453
18.3.8	PWM モード	454
18.3.9	デバッグモード	455
18.4	TIM14 レジスタ	456
18.4.1	TIM14 制御レジスタ 1 (TIM14_CR1)	456
18.4.2	TIM14 割り込み有効レジスタ (TIM14_DIER)	457
18.4.3	TIM14 ステータスレジスタ (TIM14_SR)	457
18.4.4	TIM14 のイベント生成レジスタ (TIM14_EGR)	459
18.4.5	TIM14 のキャプチャ／比較モードレジスタ 1 (TIM14_CCMR1)	460
18.4.6	TIM14 のキャプチャ／比較有効レジスタ (TIM14_CCER)	462
18.4.7	TIM14 のカウンタ (TIM14_CNT)	463
18.4.8	TIM14 のプリスケアラ (TIM14_PSC)	463
18.4.9	TIM14 の自動再ロードレジスタ (TIM14_ARR)	463
18.4.10	TIM14 のキャプチャ／比較レジスタ 1 (TIM14_CCR1)	464
18.4.11	TIM14 オプションレジスタ (TIM14_OR)	464
18.4.12	TIM14 レジスタマップ	465
19	汎用タイマ (TIM15/16/17)	467
19.1	TIM15/16/17 の概要	467
19.2	TIM15 の主な機能	467
19.3	TIM16 および TIM17 の主な機能	469
19.4	TIM15/16/17 の機能説明	470
19.4.1	タイムベースユニット	470
19.4.2	カウンタモード	472
19.4.3	繰り返しカウンタ	476
19.4.4	クロックソース	477
19.4.5	キャプチャ／比較チャネル	479
19.4.6	入力キャプチャモード	481
19.4.7	PWM 入力モード (TIM15 の場合のみ)	482
19.4.8	強制出力モード	483
19.4.9	出力比較モード	483
19.4.10	PWM モード	485
19.4.11	相補出力とデッドタイム挿入	486
19.4.12	ブレーク機能の使用	488

19.4.13	ワンパルスモード	491
19.4.14	TIM15 外部 トリガ同期	493
19.4.15	タイマ同期 (TIM15)	495
19.4.16	デバッグモード	495
19.5	TIM15 レジスタ	496
19.5.1	TIM15 制御レジスタ 1 (TIM15_CR1)	496
19.5.2	TIM15 制御レジスタ 2 (TIM15_CR2)	497
19.5.3	TIM15 のスレーブモード制御レジスタ (TIM15_SMCR)	499
19.5.4	TIM15 DMA / 割り込み有効レジスタ (TIM15_DIER)	500
19.5.5	TIM15 ステータスレジスタ (TIM15_SR)	501
19.5.6	TIM15 のイベント生成レジスタ (TIM15_EGR)	503
19.5.7	TIM15 のキャプチャ/比較モードレジスタ 1 (TIM15_CCMR1)	504
19.5.8	TIM15 のキャプチャ/比較有効レジスタ (TIM15_CCER)	507
19.5.9	TIM15 のカウンタ (TIM15_CNT)	510
19.5.10	TIM15 のプリスケアラ (TIM15_PSC)	510
19.5.11	TIM15 の自動再ロードレジスタ (TIM15_ARR)	510
19.5.12	TIM15 繰り返しカウンタレジスタ (TIM15_RCR)	511
19.5.13	TIM15 のキャプチャ/比較レジスタ 1 (TIM15_CCR1)	511
19.5.14	TIM15 のキャプチャ/比較レジスタ 2 (TIM15_CCR2)	512
19.5.15	TIM15 ブレークおよびデッドタイムレジスタ (TIM15_BDTR)	512
19.5.16	TIM15 DMA 制御レジスタ (TIM15_DCR)	514
19.5.17	完全転送の TIM15 DMA アドレス (TIM15_DMAR)	515
19.5.18	TIM15 レジスタマップ	516
19.6	TIM16 および TIM17 レジスタ	518
19.6.1	TIM16 および TIM17 制御レジスタ 1 (TIM16_CR1 および TIM17_CR1)	518
19.6.2	TIM16 および TIM17 制御レジスタ 2 (TIM16_CR2 および TIM17_CR2)	519
19.6.3	TIM16 および TIM17 DMA/ 割り込み有効レジスタ (TIM16_DIER および TIM17_DIER)	520
19.6.4	TIM16 および TIM17 ステータスレジスタ (TIM16_SR および TIM17_SR)	521
19.6.5	TIM16 および TIM17 のイベント生成レジスタ (TIM16_EGR および TIM17_EGR)	522
19.6.6	TIM16 および TIM17 のキャプチャ/比較モードレジスタ 1 (TIM16_CCMR1 および TIM17_CCMR1)	523
19.6.7	TIM16 および TIM17 のキャプチャ/比較有効レジスタ (TIM16_CCER および TIM17_CCER)	527
19.6.8	TIM16 および TIM17 のカウンタ (TIM16_CNT および TIM17_CNT)	529
19.6.9	TIM16 および TIM17 のプリスケアラ (TIM16_PSC および TIM17_PSC)	529

19.6.10	TIM16 および TIM17 自動再ロードレジスタ (TIM16_ARR および TIM17_ARR)	529
19.6.11	TIM16 および TIM17 繰り返しカウンタレジスタ (TIM16_RCR および TIM17_RCR)	530
19.6.12	TIM16 および TIM17 のキャプチャ/比較レジスタ 1 (TIM16_CCR1 および TIM17_CCR1)	530
19.6.13	TIM16 および TIM17 ブレークおよびデッドタイムレジスタ (TIM16_BDTR および TIM17_BDTR)	531
19.6.14	TIM16 および TIM17 DMA 制御レジスタ (TIM16_DCR および TIM17_DCR)	533
19.6.15	完全転送の TIM16 および TIM17 DMA アドレス (TIM16_DMAR および TIM17_DMAR)	533
19.6.16	TIM16 および TIM17 のレジスタマップ	536
20	基本タイマ (TIM6/TIM7)	538
20.1	TIM6/TIM7 の概要	538
20.2	TIM6/TIM7 の主な機能	538
20.3	TIM6/TIM7 の機能詳細	539
20.3.1	タイムベースユニット	539
20.3.2	カウンタモード	541
20.3.3	クロックソース	545
20.3.4	デバッグモード	545
20.4	TIM6/TIM7 レジスタ	546
20.4.1	TIM6/TIM7 制御レジスタ 1 (TIMx_CR1)	546
20.4.2	TIM6/TIM7 制御レジスタ 2 (TIMx_CR2)	548
20.4.3	TIM6/TIM7 DMA / 割り込み有効レジスタ (TIMx_DIER)	548
20.4.4	TIM6/TIM7 のステータスレジスタ (TIMx_SR)	549
20.4.5	TIM6/TIM7 のイベント生成レジスタ (TIMx_EGR)	549
20.4.6	TIM6/TIM7 のカウンタ (TIMx_CNT)	549
20.4.7	TIM6/TIM7 プリスケアラ (TIMx_PSC)	550
20.4.8	TIM6/TIM7 自動再ロードレジスタ (TIMx_ARR)	550
20.4.9	TIM6/TIM7 レジスタマップ	551

21	赤外線インタフェース (IRTIM)	552
22	独立型ウォッチドッグ (IWDG)	553
22.1	概要	553
22.2	IWDG の主な機能	553
22.3	IWDG の機能説明	553
22.3.1	IWDG ブロック図	553
22.3.2	ウィンドウオプション	554
22.3.3	ハードウェアウォッチドッグ	555
22.3.4	レジスタのアクセス保護	555
22.3.5	デバッグモード	555
22.4	IWDG レジスタ	555
22.4.1	キーレジスタ (IWDG_KR)	556
22.4.2	プリスケアラレジスタ (IWDG_PR)	557
22.4.3	再ロードレジスタ (IWDG_RLR)	558
22.4.4	ステータスレジスタ (IWDG_SR)	559
22.4.5	ウィンドウレジスタ (IWDG_WINR)	560
22.4.6	IWDG レジスタマップ	561
23	システムウィンドウ型ウォッチドッグ (WWDG)	562
23.1	概要	562
23.2	WWDG の主な機能	562
23.3	WWDG の機能説明	562
23.3.1	ウォッチドッグの有効化	563
23.3.2	ダウンカウンタの制御	563
23.3.3	高度なウォッチドッグ割り込み機能	563
23.3.4	ウォッチドッグタイムアウトをプログラムする方法	564
23.3.5	デバッグモード	565
23.4	WWDG レジスタ	565
23.4.1	制御レジスタ (WWDG_CR)	565
23.4.2	設定レジスタ (WWDG_CFR)	566
23.4.3	ステータスレジスタ (WWDG_SR)	566
23.4.4	WWDG レジスタマップ	567

24	リアルタイムクロック (RTC)	568
24.1	概要	568
24.2	RTC の主な機能	568
24.3	RTC の実装	569
24.4	RTC の機能説明	569
24.4.1	RTC ブロック図	569
24.4.2	RTC によって制御される GPIO	571
24.4.3	クロックとプリスケアラ	573
24.4.4	リアルタイムクロックとカレンダー	573
24.4.5	プログラム可能なアラーム	574
24.4.6	周期的自動ウェイクアップ	574
24.4.7	RTC の初期化と設定	575
24.4.8	カレンダーの読み出し	577
24.4.9	RTC のリセット	578
24.4.10	RTC の同期	578
24.4.11	RTC リファレンスクロック検出	579
24.4.12	RTC の高精度デジタル較正	580
24.4.13	タイムスタンプ機能	582
24.4.14	タンパ検出	582
24.4.15	較正クロック出力	584
24.4.16	アラーム出力	584
24.5	RTC 低電力モード	585
24.6	RTC 割り込み	585
24.7	RTC レジスタ	586
24.7.1	RTC 時刻レジスタ (RTC_TR)	586
24.7.2	RTC 日付レジスタ (RTC_DR)	587
24.7.3	RTC 制御レジスタ (RTC_CR)	588
24.7.4	RTC 初期化とステータスレジスタ (RTC_ISR)	591
24.7.5	RTC プリスケアラレジスタ (RTC_PRER)	593
24.7.6	RTC ウェイクアップタイマレジスタ (RTC_WUTR)	594
24.7.7	RTC アラーム A レジスタ (RTC_ALRMAR)	595
24.7.8	RTC 書き込み保護レジスタ (RTC_WPR)	596
24.7.9	RTC サブセカンドレジスタ (RTC_SSR)	596
24.7.10	RTC シフト制御レジスタ (RTC_SHIFTR)	597
24.7.11	RTC タイムスタンプ時刻レジスタ (RTC_TSTR)	598
24.7.12	RTC タイムスタンプ日付レジスタ (RTC_TSDR)	599

24.7.13	RTC タイムスタンプサブセカンドレジスタ (RTC_TSSSR)	600
24.7.14	RTC 較正レジスタ (RTC_CALR)	601
24.7.15	RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)	602
24.7.16	RTC アラーム A サブセカンドレジスタ (RTC_ALRMSSR)	605
24.7.17	RTC バックアップレジスタ (RTC_BKPxR)	606
24.7.18	RTC レジスタマップ	607
25	I2C (Inter-integrated circuit) インタフェース	609
25.1	概要	609
25.2	I2C の主な機能	609
25.3	I2C の実装	610
25.4	I2C の機能詳細	610
25.4.1	I2C1 ブロック図	611
25.4.2	I2C2 ブロック図	612
25.4.3	I2C クロックの要件	612
25.4.4	モード選択	613
25.4.5	I2C の初期化	614
25.4.6	ソフトウェアリセット	618
25.4.7	データ転送	619
25.4.8	I2C スレーブモード	621
25.4.9	I2C マスタモード	630
25.4.10	I2C_TIMINGR レジスタの設定例	642
25.4.11	SMBus 固有の機能	643
25.4.12	SMBus 初期化	646
25.4.13	SMBus : I2C_TIMEOUTR レジスタの設定例	648
25.4.14	SMBus スレーブモード	649
25.4.15	アドレス一致時に STOP モードからウェイクアップ	656
25.4.16	エラー条件	657
25.4.17	DMA リクエスト	659
25.4.18	デバッグモード	660
25.5	I2C 低電力モード	660
25.6	I2C 割り込み	660
25.7	I2C レジスタ	662
25.7.1	制御レジスタ 1 (I2C_CR1)	662
25.7.2	制御レジスタ 2 (I2C_CR2)	665
25.7.3	Own Address 1 レジスタ (I2C_OAR1)	668

25.7.4	Own Address 2 レジスタ (I2C_OAR2)	669
25.7.5	タイミングレジスタ (I2C_TIMINGR)	670
25.7.6	タイムアウトレジスタ (I2C_TIMEOUTR)	671
25.7.7	割り込みおよびステータスレジスタ (I2C_ISR)	672
25.7.8	割り込みクリアレジスタ (I2C_ICR)	674
25.7.9	PEC レジスタ (I2C_PECR)	675
25.7.10	受信データレジスタ (I2C_RXDR)	676
25.7.11	送信データレジスタ (I2C_TXDR)	676
25.7.12	I2C レジスタマップ	677

26

USART

(Universal synchronous asynchronous receiver transmitter) 679

26.1	概要	679
26.2	USART の主な機能	679
26.3	USART の拡張機能	680
26.4	USART の実装	681
26.5	USART の機能詳細	682
26.5.1	USART キャラクタの説明	684
26.5.2	トランスミッタ	685
26.5.3	レシーバ	688
26.5.4	ボーレート生成	695
26.5.5	クロック偏差に対する USART レシーバの許容誤差	697
26.5.6	自動ボーレート検出	698
26.5.7	マルチプロセッサ通信	699
26.5.8	Modbus 通信	701
26.5.9	パリティ制御	702
26.5.10	LIN (Local Interconnection Network) モード	703
26.5.11	USART 同期モード	705
26.5.12	単線半二重通信	708
26.5.13	スマートカードモード	708
26.5.14	Ir SIR ENDEC ブロック	713
26.5.15	DMA を使用した連続通信	715
26.5.16	RS232 ハードウェアフロー制御および RS485 ドライバ有効	717
26.5.17	STOP モードからのウェイクアップ	719

26.6	USART 低電力モード	720
26.7	USART 割り込み	720
26.8	USART レジスタ	722
26.8.1	制御レジスタ 1 (USARTx_CR1)	722
26.8.2	制御レジスタ 2 (USARTx_CR2)	725
26.8.3	制御レジスタ 3 (USARTx_CR3)	729
26.8.4	ボーレートレジスタ (USARTx_BRR)	733
26.8.5	ガード時間およびプリスケアラレジスタ (USARTx_GTPR)	733
26.8.6	レシーバタイムアウトレジスタ (USARTx_RTOR)	734
26.8.7	リクエストレジスタ (USARTx_RQR)	735
26.8.8	割り込みおよびステータスレジスタ (USARTx_ISR)	736
26.8.9	割り込みフラグクリアレジスタ (USARTx_ICR)	741
26.8.10	受信データレジスタ (USARTx_RDR)	742
26.8.11	送信データレジスタ (USARTx_TDR)	742
26.8.12	USART レジスタマップ	743
27	シリアルペリフェラルインタフェース / I2S (SPI/I2S)	745
27.1	概要	745
27.2	SPI の主な機能	745
27.3	I2S の主な機能	746
27.4	SPI/I2S の実装	746
27.5	SPI の機能説明	747
27.5.1	概要	747
27.5.2	マスタとスレーブの 1 対 1 の通信	748
27.5.3	標準マルチスレーブ通信	750
27.5.4	スレーブ選択 (NSS) ピンの管理	751
27.5.5	通信フォーマット	753
27.5.6	SPI の設定	755
27.5.7	SPI を有効にする手順	756
27.5.8	データの送受信手順	756
27.5.9	SPI ステータスフラグ	766
27.5.10	SPI エラーフラグ	767
27.5.11	NSS パルスモード	768
27.5.12	TI モード	769
27.5.13	CRC 計算	770

27.6	SPI 割り込み	771
27.7	I ² S の機能説明	772
27.7.1	I ² S の概要	772
27.7.2	サポートされるオーディオプロトコル	773
27.7.3	起動に関する説明	781
27.7.4	クロックジェネレータ	782
27.7.5	I ² S マスタモード	784
27.7.6	I ² S スレーブモード	786
27.7.7	I ² S ステータスフラグ	788
27.7.8	I ² S エラーフラグ	789
27.7.9	DMA の機能	789
27.8	I ² S 割り込み	790
27.9	SPI および I ² S レジスタ	791
27.9.1	SPI 制御レジスタ 1 (SPIx_CR1)	791
27.9.2	SPI 制御レジスタ 2 (SPIx_CR2)	793
27.9.3	SPI ステータスレジスタ (SPIx_SR)	796
27.9.4	SPI データレジスタ (SPIx_DR)	797
27.9.5	SPI CRC 多項式レジスタ (SPIx_CRCPR)	798
27.9.6	SPI Rx CRC レジスタ (SPIx_RXCRCR)	798
27.9.7	SPI Tx CRC レジスタ (SPIx_TXCRCR)	799
27.9.8	SPIx_I ² S 設定レジスタ (SPIx_I2SCFGR)	799
27.9.9	SPIx_I ² S プリスケアラレジスタ (SPIx_I2SPR)	801
27.9.10	SPI/I2S レジスタマップ	802
28	タッチセンシングコントローラ (TSC)	803
28.1	概要	803
28.2	TSC の主な機能	803
28.3	TSC の機能説明	804
28.3.1	TSC ブロック図	804
28.3.2	表面電荷移動取得の概要	804
28.3.3	リセットおよびクロック	806
28.3.4	電荷移動取得シーケンス	807
28.3.5	スペクトル拡散機能	808
28.3.6	最大カウントエラー	808
28.3.7	サンプリングコンデンサ I/O および チャネル I/O のモード選択	809
28.3.8	取得モード	810

28.3.9	I/O ヒステリシスおよびアナログスイッチの制御	810
28.3.10	静電容量検出 GPIO	811
28.4	TSC 低電力モード	811
28.5	TSC 割り込み	811
28.6	TSC レジスタ	812
28.6.1	TSC 制御レジスタ (TSC_CR)	812
28.6.2	TSC 割り込み有効レジスタ (TSC_IER)	815
28.6.3	TSC 割り込みクリアレジスタ (TSC_ICR)	815
28.6.4	TSC 割り込みステータスレジスタ (TSC_ISR)	817
28.6.5	TSC I/O ヒステリシス制御レジスタ (TSC_IOHCR)	817
28.6.6	TSC I/O アナログスイッチ制御レジスタ (TSC_IOASCR)	818
28.6.7	TSC I/O サンプリング制御レジスタ (TSC_IOSCR)	818
28.6.8	TSC I/O チャネル制御レジスタ (TSC_IOCCTSC_IOCCTCR)	819
28.6.9	TSC I/O グループ制御ステータスレジスタ (TSC_ILOGCSR)	820
28.6.10	TSC I/O グループ x カウンタレジスタ (TSC_ILOGxCR) (x = 1 ~ 8)	820
28.6.11	TSC レジスタマップ	821
29	コントローラエリアネットワーク (bxCAN)	823
29.1	概要	823
29.2	bxCAN の主な機能	823
29.3	bxCAN の概要	824
29.3.1	CAN 2.0B アクティブコア	824
29.3.2	制御レジスタ、ステータスレジスタ、設定レジスタ	824
29.3.3	送信メールボックス	824
29.3.4	受信フィルタ	825
29.4	bxCAN 動作モード	825
29.4.1	初期化モード	825
29.4.2	通常モード	826
29.4.3	スリープモード (低電力)	826
29.5	テストモード	827
29.5.1	サイレントモード	827
29.5.2	ループバックモード	828
29.5.3	サイレントモードとループバックモードの組み合わせ	828
29.6	デバッグモードでの挙動	829
29.7	bxCAN 機能の説明	829
29.7.1	送信処理	829

29.7.2	タイムトリガ通信モード	831
29.7.3	受信処理	831
29.7.4	識別子フィルタリング	832
29.7.5	メッセージストレージ	836
29.7.6	エラー管理	838
29.7.7	ビットタイミング	838
29.8	bxCAN の割込み	841
29.9	CAN レジスタ	842
29.9.1	レジスタのアクセス保護	842
29.9.2	CAN 制御/ステータスレジスタ	843
29.9.3	CAN メールボックスレジスタ	853
29.9.4	CAN フィルタレジスタ	860
29.9.5	bxCAN レジスタマップ	864
30	ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)	868
30.1	概要	868
30.2	USB の主な機能	868
30.3	USB の実装	869
30.4	USART の機能詳細	870
30.4.1	USB ブロックの説明	871
30.5	プログラミングに関する考慮事項	872
30.5.1	汎用 USB デバイスのプログラミング	872
30.5.2	システムリセットとパワーオンリセット	872
30.5.3	ダブルバッファエンドポイント	877
30.5.4	アイソクロナス転送	880
30.5.5	サスペンド/レジュームイベント	881
30.6	USB レジスタ	883
30.6.1	共通レジスタ	883
30.6.2	バッファディスクリプタテーブル	896
30.6.3	USB レジスタマップ	899

31	HDMI-CEC コントローラ (HDMI-CEC)	901
31.1	概要	901
31.2	HDMI-CEC コントローラの主な機能	902
31.3	HDMI-CEC の機能詳細	903
31.3.1	HDMI-CEC ピン	903
31.3.2	メッセージの説明	904
31.3.3	ビットタイミング	904
31.4	アービトレーション	905
31.4.1	SFT オプションビット	906
31.5	エラー処理	907
31.5.1	ビットエラー	907
31.5.2	メッセージエラー	907
31.5.3	ビット立ち上がりエラー (BRE)	908
31.5.4	ショートビット周期エラー (SBPE)	908
31.5.5	ロングビット周期エラー (LBPE)	908
31.5.6	送信エラー検出 (TXERR)	910
31.6	HDMI-CEC 割り込み	911
31.7	HDMI-CEC レジスタ	912
31.7.1	CEC 制御レジスタ (CEC_CR)	912
31.7.2	CEC 設定レジスタ (CEC_CFGR)	913
31.7.3	CEC Tx データレジスタ (CEC_TXDR)	916
31.7.4	CEC Rx データレジスタ (CEC_RXDR)	916
31.7.5	CEC 割り込みおよびステータスレジスタ (CEC_ISR)	916
31.7.6	CEC 割り込み有効レジスタ (CEC_IER)	918
31.7.7	HDMI-CEC レジスタマップ	920
32	デバッグサポート (DBG)	921
32.1	概要	921
32.2	ARM リファレンス資料	922
32.3	ピン名とデバッグポートピン	922
32.3.1	SWD ポートピン	922
32.3.2	SW-DP ピンの割り当て	922
32.3.3	SWD ピンでの内部プルアップ／プルダウン	922
32.4	ID コードとロック機構	923
32.4.1	MCU デバイス ID コード	923

32.5	SWD ポート	924
32.5.1	SWD プロトコルの概要	924
32.5.2	SWD プロトコルシーケンス	924
32.5.3	SW-DP ステートマシン (リセット、アイドル状態、ID コード)	925
32.5.4	DP と AP の読み出し/書き込みアクセス	926
32.5.5	SW-DP レジスタ	927
32.5.6	SW-AP レジスタ	928
32.6	コアデバッグ	928
32.7	BPU (ブレークポイントユニット)	929
32.7.1	BPU の機能	929
32.8	DWT (データウォッチポイント)	929
32.8.1	DWT の機能	929
32.8.2	DWT プログラムカウンタサンプルレジスタ	929
32.9	MCU デバッグコンポーネント (DBGMCU)	930
32.9.1	低電力モードのデバッグサポート	930
32.9.2	タイマ、ウォッチドッグ、および I ² C のデバッグサポート	930
32.9.3	デバッグ MCU 設定レジスタ (DBGMCU_CR)	931
32.9.4	デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ)	932
32.9.5	デバッグ MCU APB2 凍結レジスタ (DBGMCU_APB2_FZ)	934
32.9.6	DBG レジスタマップ	935
33	デバイス電子署名	936
33.1	ユニークデバイス ID レジスタ (96 ビット)	936
33.2	メモリサイズデータレジスタ	937
33.2.1	フラッシュサイズデータレジスタ	937
付録 A	コード例	938
A.1	概要	938
A.2	フラッシュ操作のコード例	938
A.2.1	フラッシュメモリアンロックシーケンスコード	938
A.2.2	メインフラッシュプログラミングシーケンスのコード例	938
A.2.3	ページ消去シーケンスのコード例	939
A.2.4	全体消去シーケンスのコード例	939
A.2.5	オプションバイトのアンロックシーケンスのコード例	940
A.2.6	オプションバイトのプログラミングシーケンスのコード例	940
A.2.7	オプションバイトの消去シーケンスのコード例	941

A.3	クロックコントローラ	942
A.3.1	HSE 起動シーケンスのコード例	942
A.3.2	PLL 設定変更のコード例	943
A.3.3	MCO 選択のコード例	943
A.3.4	TIM14 によるクロックの測定設定のコード例	944
A.4	GPIO	945
A.4.1	ロックシーケンスのコード例	945
A.4.2	オルタネート機能の選択シーケンスのコード例	945
A.4.3	アナログ GPIO 設定のコード例	946
A.5	DMA	946
A.5.1	DMA チャンネル設定シーケンスのコード例	946
A.6	割り込みとイベント	947
A.6.1	NVIC の初期化の例	947
A.6.2	外部割り込み選択のコード例	947
A.7	ADC	948
A.7.1	ADC 較正のコード例	948
A.7.2	ADC 有効化シーケンスのコード例	948
A.7.3	ADC 無効化シーケンスのコード例	948
A.7.4	ADC クロック選択のコード例	949
A.7.5	シングル変換シーケンスのコード例 - ソフトウェアトリガ	949
A.7.6	連続変換シーケンスのコード例 - ソフトウェアトリガ	950
A.7.7	シングル変換シーケンスのコード例 - ハードウェアトリガ	950
A.7.8	連続変換シーケンスのコード例 - ハードウェアトリガ	950
A.7.9	DMA ワンショットモードシーケンスのコード例	951
A.7.10	DMA サーキュラモードシーケンスのコード例	951
A.7.11	ウェイトモードシーケンスのコード例	952
A.7.12	オートオフおよびノーウェイトモードシーケンスのコード例	952
A.7.13	オートオフおよびウェイトモードシーケンスのコード例	952
A.7.14	アナログウォッチドッグのコード例	953
A.7.15	温度設定のコード例	953
A.7.16	温度計算のコード例	953
A.8	DAC	954
A.8.1	波形生成なしの独立トリガのコード例	954
A.8.2	1 つの LFSR 生成による独立トリガのコード例	954
A.8.3	異なる LFSR 生成による独立トリガのコード例	954
A.8.4	1 つの三角波生成による独立トリガのコード例	955

A.8.5	異なる三角波生成による独立トリガのコード例	955
A.8.6	同時ソフトウェア開始のコード例	955
A.8.7	波形生成なしの同時トリガのコード例	955
A.8.8	1 つの LFSR 生成による同時トリガのコード例	956
A.8.9	異なる LFSR 生成による同時トリガのコード例	956
A.8.10	1 つの三角波生成による同時トリガのコード例	956
A.8.11	異なる三角波生成による同時トリガのコード例	957
A.8.12	DMA 初期化のコード例	957
A.9	タイマ	958
A.9.1	TI2 の立ち上がりエッジのアップカウンタのコード例	958
A.9.2	2 回の ETR の立ち上がりエッジごとのアップカウンタのコード例 ..	958
A.9.3	入力キャプチャ設定のコード例	959
A.9.4	入力キャプチャのデータ管理のコード例	959
A.9.5	PWM 入力設定のコード例	960
A.9.6	DMA 設定による PWM 入力のコード例	960
A.9.7	入力比較設定のコード例	961
A.9.8	エッジアライン PWM の設定例	961
A.9.9	センターアライン PWM の設定例	962
A.9.10	OCxREF をクリアするための ETR 設定のコード例	962
A.9.11	エンコーダインタフェースのコード例	963
A.9.12	リセットモードのコード例	963
A.9.13	ゲートモードのコード例	963
A.9.14	トリガモードのコード例	964
A.9.15	外部クロックモード 2 + トリガモードのコード例	964
A.9.16	ワンパルスモードのコード例	965
A.9.17	別のタイマをプリスケリングするタイマのコード例	965
A.9.18	別のタイマを有効にするタイマのコード例	966
A.9.19	マスタとスレーブの同期のコード例	967
A.9.20	外部トリガによって同期された 2 つのタイマのコード例	968
A.9.21	DMA バースト機能のコード例	969
A.10	IRTIM のコード例	970
A.10.1	TIM16 および TIM17 設定のコード例	970
A.10.2	IRTIM の IRQHandler のコード例	971
A.11	bxCAN のコード例	972
A.11.1	bxCAN 初期化モードのコード例	972
A.11.2	bxCAN 送信のコード例	972
A.11.3	bxCAN 受信のコード例	973

A.12	DBG のコード例	973
A.12.1	DBG 読み出しデバイス ID のコード例	973
A.12.2	低電力モードの DBG デバッグのコード例	973
A.13	HDMI-CEC のコード例	974
A.13.1	CEC を設定する HDMI-CEC のコード例	974
A.13.2	割り込みが有効な HDMI-CEC 送信のコード例	974
A.13.3	HDMI-CEC の割り込み管理のコード例	974
A.14	I2C のコード例	975
A.14.1	受信するためにマスタモードに設定された I2C のコード例	975
A.14.2	送信するためにマスタモードに設定された I2C のコード例	975
A.14.3	スレーブモードで設定された I2C のコード例	975
A.14.4	I2C マスタトランスミッタのコード例	976
A.14.5	I2C マスタレシーバのコード例	976
A.14.6	I2C スレーブトランスミッタのコード例	976
A.14.7	I2C スレーブレシーバのコード例	977
A.14.8	DMA で送信するためにマスタモードに設定された I2C のコード例 ..	977
A.14.9	DMA で受信するためにスレーブモードに設定された I2C の コード例	977
A.15	IWDG のコード例	978
A.15.1	IWDG 設定のコード例	978
A.15.2	ウィンドウによる IWDG 設定のコード例	978
A.16	RTC のコード例	979
A.16.1	RTC カレンダ設定のコード例	979
A.16.2	RTC アラーム設定のコード例	979
A.16.3	RTC WUT 設定のコード例	980
A.16.4	RTC 読み出しカレンダのコード例	980
A.16.5	RTC 較正のコード例	980
A.16.6	RTC タンパおよびタイムスタンプ設定のコード例	981
A.16.7	RTC タンパおよびタイムスタンプのコード例	981
A.16.8	RTC クロック出力のコード例	981
A.17	SPI のコード例	982
A.17.1	SPI マスタ設定のコード例	982
A.17.2	SPI スレーブ設定のコード例	982
A.17.3	SPI 全二重通信のコード例	982
A.17.4	SPI 割り込みのコード例	983
A.17.5	DMA による SPI マスタ設定のコード例	983
A.17.6	DMA による SPI スレーブ設定のコード例	983

A.18	TSC のコード例	984
A.18.1	TSC 設定のコード例	984
A.18.2	TSC 割り込みのコード例	984
A.19	USART のコード例	985
A.19.1	USART トランスミッタ設定のコード例	985
A.19.2	USART 送信バイトのコード例	985
A.19.3	USART 転送完了のコード例	985
A.19.4	USART レシーバ設定のコード例	985
A.19.5	USART 受信バイトのコード例	985
A.19.6	USART LIN モードのコード例	986
A.19.7	USART 同期モードのコード例	986
A.19.8	USART 単線半二重のコード例	987
A.19.9	USART スマートカードモードのコード例	987
A.19.10	USART IrDA モードのコード例	988
A.19.11	USART DMA のコード例	988
A.19.12	USART ハードウェアフロー制御のコード例	988
A.20	WWDG のコード例	989
A.20.1	WWDG 設定のコード例	989
改版履歴		990

表の一覧

表 1.	STM32F0xx ペリフェラルレジスタ境界アドレス	47
表 2.	STM32F0xx メモリ境界アドレス	50
表 3.	ブートモード	53
表 4.	フラッシュメモリの構成 (STM32F03x、STM32F04x および STM32F05x デバイス)	56
表 5.	フラッシュメモリの構成 (STM32F07x、STM32F09x デバイス)	57
表 6.	フラッシュメモリの読み出し保護ステータス	65
表 7.	アクセス状態 対 保護レベルと実行モード	66
表 8.	フラッシュ割り込みリクエスト	68
表 9.	フラッシュインタフェース - レジスタマップとリセット値	74
表 10.	オプションバイトのフォーマット	75
表 11.	オプションバイトの構成	75
表 12.	オプションバイトマップおよび ST 出荷時の値	79
表 13.	CRC レジスタマップとリセット値	85
表 14.	低電力モードの概要	91
表 15.	Sleep-now	93
表 16.	Sleep-on-exit	93
表 17.	STOP モード	94
表 18.	STANDBY モード	95
表 19.	PWR レジスタマップとリセット値	99
表 20.	RCC レジスタマップとリセット値	142
表 21.	低電力モードが CRS に与える影響	148
表 22.	割り込み制御ビット	148
表 23.	CRS レジスタマップとリセット値	155
表 24.	ポートビット設定表	157
表 25.	GPIO レジスタマップとリセット値	171
表 26.	SYSCFG レジスタマップとリセット値	194
表 27.	STM32F09x デバイスの SYSCFG レジスタマップとリセット値	194
表 28.	プログラム可能なデータ幅およびエンディアン動作 (ビット PINC = MINC = 1 の場合)	201
表 29.	DMA 割り込みリクエスト	202
表 30.	チャンネルごとの DMA リクエストの概要 (STM32F03x、STM32F04x、および STM32F030x8STM32F05x デバイスの場合)	203
表 31.	STM32F07x デバイスでのチャンネルごとの DMA リクエストの概要	203
表 32.	STM32F09x デバイスでのチャンネルごとの DMA1 リクエストの概要	206
表 33.	STM32F09x デバイスでのチャンネルごとの DMA2 リクエストの概要	207
表 34.	DMA レジスタマップとリセット値	215
表 35.	DMA レジスタマップとリセット値 (レジスタは STM32F07x および STM32F09x デバイスでのみ使用可能)	216
表 36.	DMA レジスタマップとリセット値 (レジスタは STM32F09x デバイスでのみ使用可能)	216
表 37.	ベクターテーブル	217
表 38.	外部割り込み/イベントコントローラのレジスタマップとリセット値	228
表 39.	ADC 内部信号	231
表 40.	ADC ピン	231
表 41.	トリガから変換開始までの遅延	236
表 42.	トリガ極性の設定	241
表 43.	外部トリガ	241
表 44.	tSAR タイミングは分解能に依存	242
表 45.	アナログウォッチドッグ比較	251
表 46.	アナログウォッチドッグチャンネル選択	252

表 47.	ADC 割り込み	254
表 48.	ADC レジスタマップとリセット値	268
表 49.	DAC ピン	270
表 50.	外部トリガ	273
表 51.	DAC レジスタマップ とリセット値	294
表 52.	COMP レジスタマップとリセット値	303
表 53.	カウント方向とエンコーダ信号	342
表 54.	TIMx 内部トリガ接続	355
表 55.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能	367
表 56.	TIM1 レジスタマップとリセット値	375
表 57.	カウント方向とエンコーダ信号	408
表 58.	TIM2 および TIM3 内部トリガ接続	425
表 59.	標準 OCx チャンネルの出力制御ビット	436
表 60.	TIM2 および TIM3 のレジスタマップとリセット値	442
表 61.	標準 OCx チャンネルの出力制御ビット	463
表 62.	TIM14 レジスタマップとリセット値	465
表 63.	TIMx 内部トリガ接続	500
表 64.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能	509
表 65.	TIM15 レジスタマップとリセット値	516
表 66.	ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能	528
表 67.	TIM16 および TIM17 のレジスタマップとリセット値	536
表 68.	TIM6/TIM7 レジスタマップとリセット値	551
表 69.	IWDG レジスタマップとリセット値	561
表 70.	WWDG レジスタマップとリセット値	567
表 71.	STM32F0xx RTC implementation	569
表 72.	RTC ピン PC13 の設定	571
表 73.	LSE ピン PC14 の設定	572
表 74.	LSE ピン PC15 の設定	572
表 75.	低電力モードが RTC に与える影響	585
表 76.	割り込み制御ビット	585
表 77.	RTC レジスタマップとリセット値	607
表 78.	STM32F0xx I2C 実装	610
表 79.	アナログフィルタとデジタルフィルタの比較	614
表 80.	I2C-SMBUS 仕様のデータのセットアップおよびホールド時間	617
表 81.	I2C 設定表	621
表 82.	I2C-SMBUS 仕様のクロックタイミング	631
表 83.	fI2CCLK = 8 MHz でのタイミング設定の例	642
表 84.	fI2CCLK = 16 MHz でのタイミング設定の例	642
表 85.	fI2CCLK = 48 MHz でのタイミング設定の例	643
表 86.	SMBus タイムアウト仕様	645
表 87.	SMBUS の PEC 設定	647
表 88.	さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{\text{TIMEOUT}} = 25 \text{ ms}$)	648
表 89.	さまざまな I2CCLK 周波数での TIMEOUTB の設定例	649
表 90.	さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{\text{DLE}} = 50 \mu\text{s}$)	649
表 91.	低電力モード	660
表 92.	I2C 割り込みリクエスト	660
表 93.	I2C レジスタマップとリセット値	677
表 94.	STM32F0xx USART 実装	681
表 95.	サンプリングされたデータからのノイズ検出	693
表 96.	16 倍または 8 倍のオーバーサンプリングの両方の場合の $f_{\text{CK}} = 48 \text{ MHz}$ での プログラムされたボーレートの誤差計算	696

表 97.	BRR [3:0] = 0000 のときの USART レシーバの許容誤差	698
表 98.	BRR[3:0] が 0000 でないときの USART レシーバの許容誤差	698
表 99.	フレームフォーマット	702
表 100.	低電力モードが USART に与える影響	720
表 101.	USART 割り込みリクエスト	720
表 102.	USART レジスタマップとリセット値	743
表 103.	STM32F0xx SPI の実装	746
表 104.	SPI 割り込みリクエスト	771
表 105.	標準 8 MHz HSE を使用した場合のオーディオ周波数精度	783
表 106.	I ² S 割り込みリクエスト	790
表 107.	SPI レジスタマップとリセット値	802
表 108.	取得シーケンスの概要	806
表 109.	スペクトル拡散偏差と AHB クロック周波数	808
表 110.	モードおよび IODEF ビットの値に応じた I/O の状態	809
表 111.	静電容量検出 GPIO	811
表 112.	低電力モードが TSC に与える影響	811
表 113.	割り込み制御ビット	811
表 114.	TSC レジスタマップとリセット値	821
表 115.	送信メールボックスの配置	837
表 116.	受信メールボックスの配置	837
表 117.	bxCAN レジスタマップとリセット値	864
表 118.	STM32F0xx USB の実装	869
表 119.	ダブルバッファリングバッファフラグの定義	878
表 120.	バルクダブルバッファリングメモリバッファの用途	879
表 121.	アイソクロナスメモリバッファの使用	880
表 122.	レジャーイベント検出	882
表 123.	受信ステータスエンコード	895
表 124.	エンドポイントタイプエンコード	895
表 125.	エンドポイントの種類の意味	895
表 126.	送信ステータスエンコード	895
表 127.	割り当てられるバッファメモリの定義	898
表 128.	USB レジスタマップとリセット値	899
表 129.	HDMI ピン	903
表 130.	エラー処理タイミングパラメータ	909
表 131.	TXERR タイミングパラメータ	910
表 132.	HDMI-CEC 割り込み	911
表 133.	HDMI-CEC レジスタマップとリセット値	920
表 134.	SW デバッグポートピン	922
表 135.	パケットリクエスト (8 ビット)	924
表 136.	ACK 応答 (3 ビット)	925
表 137.	データ転送 (33 ビット)	925
表 138.	SW-DP レジスタ	927
表 139.	シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ	928
表 140.	コアデバッグレジスタ	928
表 141.	DBG レジスタマップとリセット値	935
表 142.	文書改版履歴	990

図の一覧

図 1.	システムアーキテクチャ	44
図 2.	メモリマップ	46
図 3.	プログラミング手順	60
図 4.	フラッシュメモリのページ消去手順	62
図 5.	フラッシュメモリの全体消去手順	63
図 6.	CRC 計算ユニットのブロック図	81
図 7.	電源の概要	86
図 8.	パワーオンリセット／パワーダウンリセット波形	89
図 9.	PVD の閾値	90
図 10.	簡略化されたリセット回路図	101
図 11.	クロックツリー (STM32F03x および STM32F05x デバイス)	104
図 12.	クロックツリー (STM32F04x、STM32F07x、および STM32F09x デバイス)	105
図 13.	HSE/LSE クロックソース	106
図 14.	キャプチャモードにおける TIM14 を使用した周波数測定	112
図 15.	CRS ブロック図	145
図 16.	CRS カウンタの動作	146
図 17.	I/O ポートビットの基本構造	157
図 18.	入力フローティング／プルアップ／プルダウン設定	161
図 19.	出力設定	162
図 20.	オルタネート機能設定	163
図 21.	ハインピーダンスアナログ設定	163
図 22.	DMA ブロック図	198
図 23.	STM32F09x デバイスの DMAx リクエストルーティングアーキテクチャ	205
図 24.	拡張割り込み／イベントコントローラ (EXTI) のブロック図	220
図 25.	外部割り込み／イベント GPIO の配置	222
図 26.	ADC ブロック図	232
図 27.	ADC 較正	233
図 28.	ADC の有効化 / 無効化	234
図 29.	ADC クロック構成	235
図 30.	アナログ / デジタル変換時間	239
図 31.	ADC 変換タイミング	240
図 32.	実行中の変換の停止	240
図 33.	シーケンスのシングル変換、ソフトウェアトリガ	244
図 34.	シーケンスの連続変換、ソフトウェアトリガ	244
図 35.	シーケンスのシングル変換、ハードウェアトリガ	245
図 36.	シーケンスの連続変換、ハードウェアトリガ	245
図 37.	データの配置と分解能	246
図 38.	オーバーラン (OVR) の例	247
図 39.	ウェイトモード変換 (連続モード、ソフトウェアトリガ)	249
図 40.	WAIT=0、AUTOFF=1 での動作	250
図 41.	WAIT=1、AUTOFF=1 での動作	250
図 42.	アナログウォッチドッグによって保護される領域	251
図 43.	温度センサおよび VREFINT チャンルのブロック図	252
図 44.	DAC ブロック図	270
図 45.	シングル DAC チャンネルモードのデータレジスタ	271
図 46.	トリガ無効 (TEN = 0) 時の変換タイミング図	272
図 47.	シングル DAC チャンネルモードのデータレジスタ	274
図 48.	デュアル DAC チャンネルモードのデータレジスタ	275
図 49.	DAC LFSR レジスタ計算アルゴリズム	281

図 50.	LFSR 波形生成による DAC 変換 (SW トリガ有効)	281
図 51.	DAC 三角波生成	282
図 52.	三角波生成による DAC 変換 (SW トリガ有効)	282
図 53.	コンパレータ 1 および 2 のブロック図	297
図 54.	コンパレータヒステリシス	298
図 55.	高機能制御タイマのブロック図	305
図 56.	プリスケラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	307
図 57.	プリスケラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	307
図 58.	内部クロック分周比が 1 の場合のカウンタのタイミング図	309
図 59.	内部クロック分周比が 2 の場合のカウンタのタイミング図	309
図 60.	内部クロック分周比が 4 の場合のカウンタのタイミング図	310
図 61.	内部クロック分周比が N の場合のカウンタのタイミング図	310
図 62.	ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)	311
図 63.	ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	311
図 64.	内部クロック分周比が 1 の場合のカウンタのタイミング図	312
図 65.	内部クロック分周比が 2 の場合のカウンタのタイミング図	313
図 66.	内部クロック分周比が 4 の場合のカウンタのタイミング図	313
図 67.	内部クロック分周比が N の場合のカウンタのタイミング図	313
図 68.	繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図 (不使用)	314
図 69.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	315
図 70.	内部クロック分周比が 2 の場合のカウンタのタイミング図	316
図 71.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	316
図 72.	内部クロック分周比が N の場合のカウンタのタイミング図	317
図 73.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図	317
図 74.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	318
図 75.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	319
図 76.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	320
図 77.	TI2 外部クロックの接続例	321
図 78.	外部クロックモード 1 の制御回路	322
図 79.	外部トリガ入力ブロック	322
図 80.	外部クロックモード 2 の制御回路	323
図 81.	キャプチャ/比較チャネル (例: チャネル 1 入力ステージ)	324
図 82.	キャプチャ/比較チャネル 1 メイン回路	324
図 83.	キャプチャ/比較チャネル (チャネル 1 から 3) の出力ステージ	325
図 84.	キャプチャ/比較チャネル (チャネル 4) の出力ステージ	325
図 85.	PWM 入力モードタイミング	327
図 86.	出力比較モード、OC1 の反転	329
図 87.	エッジアライン PWM 波形 (ARR=8)	331
図 88.	センターアライン PWM 波形 (ARR=8)	332
図 89.	デッドタイム挿入のある相補出力	334
図 90.	負のパルスより長い遅延があるときのデッドタイムの波形	334
図 91.	正のパルスより長い遅延があるときのデッドタイムの波形	334
図 92.	ブレークに対する出力の動作	337
図 93.	TIMx_OCxREF のクリア	338
図 94.	6 ステップ生成 COM の例 (OSSR=1)	339
図 95.	ワンパルスモードの例	340
図 96.	エンコーダインタフェースモードにおけるカウンタの動作例	343
図 97.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	343
図 98.	ホールセンサインタフェースの例	345

図 99.	リセットモードの制御回路	346
図 100.	ゲートモードの制御回路	347
図 101.	トリガモードの制御回路	348
図 102.	外部クロックモード 2 + トリガモードの制御回路	349
図 103.	汎用タイマのブロック図 (TIM2 および TIM3)	378
図 104.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	380
図 105.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	380
図 106.	内部クロック分周比が 1 の場合のカウンタのタイミング図	381
図 107.	内部クロック分周比が 2 の場合のカウンタのタイミング図	382
図 108.	内部クロック分周比が 4 の場合のカウンタのタイミング図	382
図 109.	内部クロック分周比が N の場合のカウンタのタイミング図	383
図 110.	ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)	383
図 111.	ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	384
図 112.	内部クロック分周比が 1 の場合のカウンタのタイミング図	385
図 113.	内部クロック分周比が 2 の場合のカウンタのタイミング図	386
図 114.	内部クロック分周比が 4 の場合のカウンタのタイミング図	386
図 115.	内部クロック分周比が N の場合のカウンタのタイミング図	387
図 116.	繰り返しカウンタが使用されていないときの更新イベント時のカウンタのタイミング図	387
図 117.	内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図	389
図 118.	内部クロック分周比が 2 の場合のカウンタのタイミング図	389
図 119.	内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図	390
図 120.	内部クロック分周比が N の場合のカウンタのタイミング図	390
図 121.	ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時、カウンタタイミング図	391
図 122.	ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図	392
図 123.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	393
図 124.	TI2 外部クロックの接続例	393
図 125.	外部クロックモード 1 の制御回路	394
図 126.	外部トリガ入力ブロック	395
図 127.	外部クロックモード 2 の制御回路	396
図 128.	キャプチャ/比較チャネル (例: チャネル 1 入力ステージ)	397
図 129.	キャプチャ/比較チャネル 1 メイン回路	397
図 130.	キャプチャ/比較チャネル (チャネル 1) の出力ステージ	398
図 131.	PWM 入力モードタイミング	400
図 132.	出力比較モード、OC1 の反転	402
図 133.	エッジアライン PWM 波形 (ARR=8)	403
図 134.	センターアライン PWM 波形 (ARR=8)	404
図 135.	ワンパルスモードの例	405
図 136.	TIMx OCxREF のクリア	407
図 137.	エンコーダインタフェースモードにおけるカウンタの動作例	409
図 138.	TI1FP1 の極性を反転したエンコーダインタフェースモードの例	409
図 139.	リセットモードの制御回路	410
図 140.	ゲートモードの制御回路	411
図 141.	トリガモードの制御回路	412
図 142.	外部クロックモード 2 + トリガモードの制御回路	413
図 143.	マスタ/スレーブタイマの例	414
図 144.	タイマ 1 の OC1REF によるタイマ 2 のゲート操作	415
図 145.	タイマ 1 の有効化によるタイマ 2 のゲート操作	416
図 146.	タイマ 1 の更新によるタイマ 2 のトリガ	417
図 147.	タイマ 1 の有効化によるタイマ 2 のトリガ	418
図 148.	タイマ 1 の TI1 入力によるタイマ 1 および 2 のトリガ	419

図 149.	汎用タイマのブロック図 (TIM14)	445
図 150.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	446
図 151.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	446
図 152.	内部クロック分周比が 1 の場合のカウンタのタイミング図	447
図 153.	内部クロック分周比が 2 の場合のカウンタのタイミング図	448
図 154.	内部クロック分周比が 4 の場合のカウンタのタイミング図	448
図 155.	内部クロック分周比が N の場合のカウンタのタイミング図	448
図 156.	ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)	449
図 157.	ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	449
図 158.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	450
図 159.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	450
図 160.	キャプチャ/比較チャンネル 1 メイン回路	451
図 161.	キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ	451
図 162.	出力比較モード、OC1 の反転	454
図 163.	エッジアライン PWM 波形 (ARR=8)	455
図 164.	TIM15 ブロック図	468
図 165.	TIM16 および TIM17 ブロック図	469
図 166.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	471
図 167.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	471
図 168.	内部クロック分周比が 1 の場合のカウンタのタイミング図	473
図 169.	内部クロック分周比が 2 の場合のカウンタのタイミング図	473
図 170.	内部クロック分周比が 4 の場合のカウンタのタイミング図	474
図 171.	内部クロック分周比が N の場合のカウンタのタイミング図	474
図 172.	ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)	475
図 173.	ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	475
図 174.	モードと TIMx_RCR レジスタの設定に応じた更新レートの例	476
図 175.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	477
図 176.	TI2 外部クロックの接続例	478
図 177.	外部クロックモード 1 の制御回路	479
図 178.	キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)	479
図 179.	キャプチャ/比較チャンネル 1 メイン回路	480
図 180.	キャプチャ/比較チャンネル (チャンネル 1) の出力ステージ	480
図 181.	キャプチャ/比較チャンネル (TIM15 の場合、チャンネル 2) の出力ステージ	481
図 182.	PWM 入力モードタイミング	483
図 183.	出力比較モード、OC1 の反転	484
図 184.	エッジアライン PWM 波形 (ARR=8)	486
図 185.	デッドタイム挿入のある相補出力	487
図 186.	負のパルスより長い遅延があるときのデッドタイムの波形	487
図 187.	正のパルスより長い遅延があるときのデッドタイムの波形	488
図 188.	ブレークに対する出力の動作	490
図 189.	ワンパルスモードの例	491
図 190.	リセットモードの制御回路	493
図 191.	ゲートモードの制御回路	494
図 192.	トリガモードの制御回路	495
図 193.	基本タイマブロック図	538
図 194.	プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図	540
図 195.	プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図	540
図 196.	内部クロック分周比が 1 の場合のカウンタのタイミング図	541

図 197.	内部クロック分周比が 2 の場合のカウンタのタイミング図	542
図 198.	内部クロック分周比が 4 の場合のカウンタのタイミング図	542
図 199.	内部クロック分周比が N の場合のカウンタのタイミング図	543
図 200.	ARPE = 0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタの タイミング図	543
図 201.	ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)	544
図 202.	内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図	545
図 203.	TIM16 と TIM17 との IR 内部ハードウェア接続	552
図 204.	独立型ウォッチドッグのブロック図	553
図 205.	ウォッチドッグのブロック図	563
図 206.	ウィンドウ型ウォッチドッグのタイミング図	564
図 207.	STM32F03x、STM32F04x、および STM32F05x デバイスの RTC ブロック図	569
図 208.	STM32F07x および STM32F09x デバイスの RTC ブロック図	570
図 209.	I2C1 ブロック図	611
図 210.	I2C2 ブロック図	612
図 211.	I2C バスプロトコル	613
図 212.	セットアップおよびホールドタイミング	615
図 213.	I2C 初期化フローチャート	618
図 214.	データ受信	619
図 215.	データ送信	620
図 216.	スレーブ初期化フローチャート	623
図 217.	I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0)	625
図 218.	I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1)	626
図 219.	I2C スレーブトランスミッタの転送バス図	627
図 220.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0)	628
図 221.	スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1)	629
図 222.	I2C スレーブレシーバの転送バス図	629
図 223.	マスタクロック生成	631
図 224.	マスタ初期化フローチャート	633
図 225.	HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス	633
図 226.	HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス	634
図 227.	N≤255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート	635
図 228.	N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート	636
図 229.	I2C マスタトランスミッタの転送バス図	637
図 230.	N≤255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート	639
図 231.	N>255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート	640
図 232.	I2C マスタレシーバの転送バス図	641
図 233.	t _{LOW:SEXT} 、t _{LOW:MEXT} のタイムアウト間隔	646
図 234.	N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート	650
図 235.	SMBus スレーブトランスミッタの転送バス図 (SBC=1)	650
図 236.	N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート	652
図 237.	SMBus スレーブレシーバのバス転送図 (SBC=1)	653
図 238.	SMBus マスタトランスミッタのバス転送図	654
図 239.	SMBus マスタレシーバのバス転送図	656
図 240.	I2C 割り込みマッピング図	661
図 241.	USART のブロック図	683
図 242.	ワード長のプログラミング	685
図 243.	設定可能なストップビット	686
図 244.	送信時の TC/TXE の動作	688
図 245.	16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出	689
図 246.	データサンプリング (16 倍のオーバーサンプリング)	693

図 247.	データサンプリング (8 倍のオーバーサンプリング).....	693
図 248.	アイドルライン検出を使用したミュートモード.....	700
図 249.	アドレスマーク検出を使用したミュートモード.....	701
図 250.	LIN モードでのブレーク検出 (11 ビットブレーク長、LBDL=1).....	704
図 251.	LIN モードでのブレーク検出とフレーミングエラー検出.....	705
図 252.	USART の同期送信の例.....	706
図 253.	USART データクロックタイミング図 (M ビット = 00).....	706
図 254.	USART データクロックタイミング図 (M ビット = 01).....	707
図 255.	RX データセットアップ/ホールド時間.....	707
図 256.	ISO 7816-3 非同期プロトコル.....	709
図 257.	ストップビット 1.5 個を使用したパリティエラー検出.....	710
図 258.	IrDA SIR ENDEC- ブロック図.....	714
図 259.	IrDA データ変調 (3/16) - 通常モード.....	714
図 260.	DMA を使用した送信.....	716
図 261.	DMA を使用した受信.....	717
図 262.	2 つの USART 間のハードウェアフロー制御.....	717
図 263.	RS232 RTS フロー制御.....	718
図 264.	RS232 CTS フロー制御.....	719
図 265.	USART 割り込みマッピング図.....	721
図 266.	SPI ブロック図.....	747
図 267.	全二重シングルマスタ/シングルスレーブアプリケーション.....	748
図 268.	半二重シングルマスタ/シングルスレーブアプリケーション.....	749
図 269.	単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ).....	750
図 270.	マスタと 3 つの独立したスレーブ.....	751
図 271.	ハードウェア/ソフトウェアスレーブ選択管理.....	752
図 272.	データクロックのタイミング図.....	754
図 273.	データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置.....	755
図 274.	FIFO での送受信のデータのパッキング.....	759
図 275.	マスタの全二重通信.....	762
図 276.	スレーブの全二重通信.....	763
図 277.	CRC のあるマスタの全二重通信.....	764
図 278.	バックされたモードでのマスタの全二重通信.....	765
図 279.	モトローラ SPI マスタモードでの NSSP パルス生成.....	768
図 280.	TI モードでの転送.....	769
図 281.	I ² S ブロック図.....	772
図 282.	I ² S フィリップスプロトコルの波形 (16/32 ビットフル精度).....	774
図 283.	I ² S フィリップス規格の波形 (24 ビットフレーム).....	774
図 284.	0x8EAA33 の送信.....	775
図 285.	0x8EAA33 の受信.....	775
図 286.	I ² S フィリップス規格 (32 ビットパケットフレームに拡張された 16 ビット).....	775
図 287.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例.....	775
図 288.	MSB 詰め 16 ビットまたは 32 ビット長フル精度.....	776
図 289.	MSB 詰め 24 ビットフレーム長.....	776
図 290.	32 ビットパケットフレームに拡張された MSB 詰め 16 ビット.....	777
図 291.	LSB 詰め 16 ビットまたは 32 ビットフル精度.....	777
図 292.	LSB 詰め 24 ビットフレーム長.....	777
図 293.	0x3478AE を送信するために必要な動作.....	778
図 294.	0x3478AE の受信に必要な動作.....	778
図 295.	32 ビットパケットフレームに拡張された LSB 詰め 16 ビット.....	778
図 296.	32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例.....	779
図 297.	PCM 規格の波形 (16 ビット).....	779

図 298.	PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)	780
図 299.	マスタモードでの開始シーケンス	781
図 300.	オーディオサンプリング周波数の定義	782
図 301.	I ² S クロックジェネレータのアーキテクチャ	782
図 302.	TSC ブロック図	804
図 303.	表面電荷移動アナログ I/O グループの構造	805
図 304.	サンプリングコンデンサの電圧の変化	806
図 305.	電荷移動取得シーケンス	807
図 306.	スペクトル拡散の変動の原理	808
図 307.	CAN ネットワークのトポロジ	824
図 308.	bxCAN 動作モード	827
図 309.	bxCAN (サイレントモード)	828
図 310.	bxCAN (ループバックモード)	828
図 311.	bxCAN (組み合わせモード)	829
図 312.	送信メールボックスの状態	830
図 313.	受信 FIFO の状態	831
図 314.	フィルタバンクのスケール設定 - レジスタ構造	834
図 315.	フィルタ番号の例	835
図 316.	フィルタのメカニズム - 例	836
図 317.	CAN エラー状態図	837
図 318.	ビットタイミング	839
図 319.	CAN フレーム	840
図 320.	イベントフラグと割り込み生成	841
図 321.	CAN メールボックスレジスタ	853
図 322.	USB ペリフェラルのブロック図	870
図 323.	パケットバッファ領域とバッファ記述テーブルの位置の例	874
図 324.	ブロック図	903
図 325.	メッセージの構造	904
図 326.	ブロック	904
図 327.	ビットタイミング	905
図 328.	信号フリータイム	905
図 329.	アービトラージフェーズ	906
図 330.	3 つの公称ビット周期の SFT	906
図 331.	エラービットタイミング	907
図 332.	エラー処理	909
図 333.	TXERR 検出	910
図 334.	ブロック図 - STM32F0xx MCU および Cortex [®] -M0 レベルのデバッグサポート	921

1 このマニュアルにおける表記の規則

1.1 レジスタに関する略記

レジスタの説明では、次の略記が使用されます。

読み出し／書き込み (rw)	これらのビットは、ソフトウェアによる読み出しと書き込みができます。
読み出し専用 (r)	これらのビットは、ソフトウェアによる読み出しのみが可能です。
書き込み専用 (w)	このビットは、ソフトウェアによる書き込みのみが可能です。ビットを読み出すと、リセット値が返されます。
読み出し／クリア (rc_w1)	このビットは、ソフトウェアによって読み出すことができ、“1”を書き込むことによってクリアできます。“0”を書き込んでも、ビットの値は変化しません。
読み出し／クリア (rc_w0)	このビットは、ソフトウェアによって読み出すことができ、“0”を書き込むことによってクリアできます。“1”を書き込んでも、ビットの値は変化しません。
読み出し／読み出しによるクリア (rc_r)	このビットは、ソフトウェアによって読み出すことができます。このビットを読み出すと、自動的に“0”にクリアされます。“0”を書き込んでも、ビットの値は変化しません。
読み出し／セット (rs)	このビットは、ソフトウェアによって読み出しとセットができます。“0”を書き込んでも、ビットの値は変化しません。
予約済み (Res.)	予約済みビットであり、リセット値に保持する必要があります。

1.2 用語

このセクションでは、本書で用いられる略語についての定義の概要を掲載しています。

- **ワード**：32 ビット長のデータ。
- **ハーフワード**：16 ビット長のデータ。
- **バイト**：8 ビット長のデータ。
- **SWD-DP (SWD デバッグポート)**：SWD-DP には、Serial Wire Debug (SWD) プロトコルに基づいた 2 ピン (クロックとデータ) のインタフェースを搭載しています。Cortex®-M0 Technical Reference Manual を参照してください。
- **IAP (アプリケーション内プログラミング)**：IAP は、ユーザプログラム実行中にマイクロコントローラのフラッシュメモリを再プログラムする機能です。
- **ICP (インサーキットプログラミング)**：ICP は、ユーザアプリケーションボードにデバイスが搭載された状態で、JTAG プロトコルや SWD プロトコルまたはブートローダを用いて、マイクロコントローラのフラッシュメモリをプログラムする機能です。
- **オプションバイト**：フラッシュメモリに格納された製品設定ビット。
- **OBL**：オプションバイトローダ。
- **AHB**：アドバンストハイパフォーマンスバス。
- **APB**：アドバンストペリフェラルバス。

1.3 使用可能なペリフェラル

すべての販売タイプで使用可能なペリフェラルとその型番については、該当するデバイスのデータシートを参照してください。

2 システムおよびメモリの概要

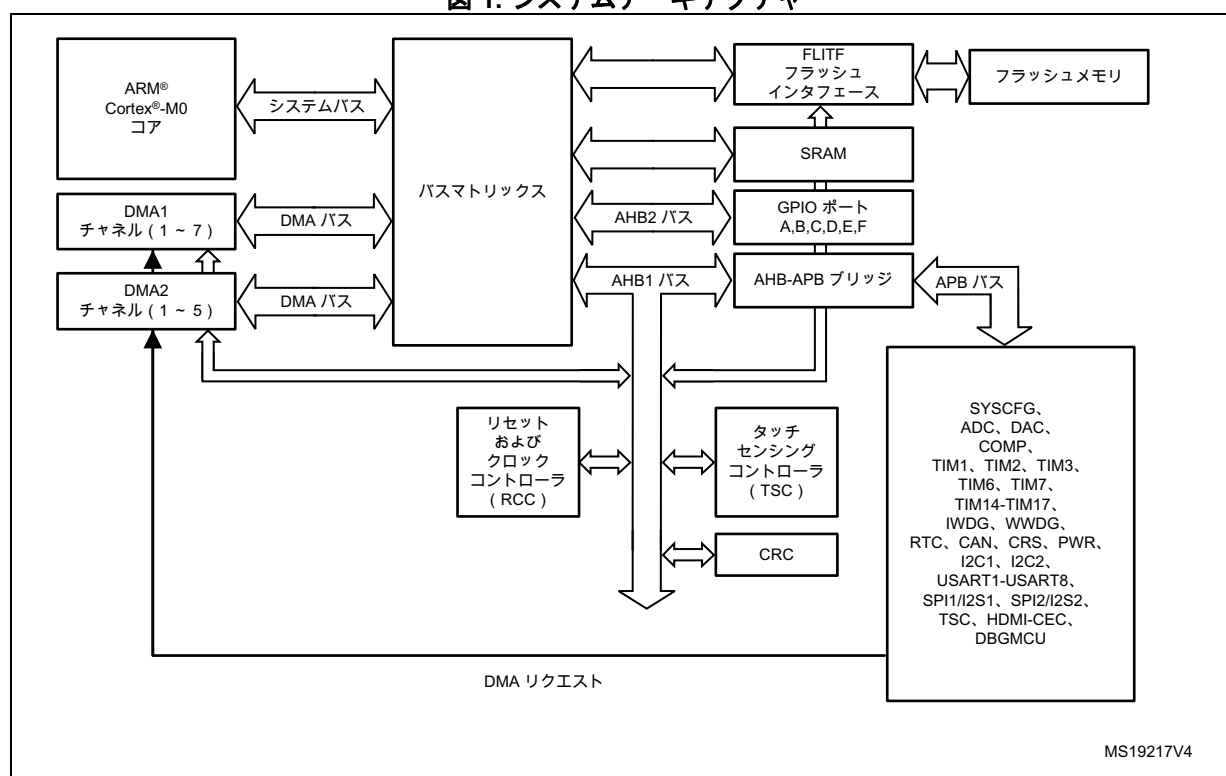
2.1 システムアーキテクチャ

メインシステムの構成は次のとおりです。

- 最大 3 つのマスタ :
 - Cortex[®]-M0 コア
 - 汎用 DMA1
 - 汎用 DMA2 (STM32F09x デバイスでのみ使用可能)
- 4 つのスレーブ :
 - 内部 SRAM
 - 内部フラッシュメモリ
 - AHB - APB ブリッジを含む AHB1、すべての APB ペリフェラルに接続
 - GPIO ポート専用の AHB2

図 1 に示すように、マルチレイヤ AHB バスアーキテクチャを使用して相互接続します。

図 1. システムアーキテクチャ



システムバス

このバスは、Cortex[®]-M0 コア（ペリフェラルバス）からコア - DMA の間のアービトレーションを管理するバスマトリックスに接続されます。

DMA バス

このバスは DMA の AHB マスタインタフェースが CPU と DMA アクセスから、SRAM、フラッシュメモリ、およびペリフェラルを管理するバスマスタへ接続します。

バスマトリックス

バスマトリックスでは、コアシステムバスと DMA マスタバスの間のアクセス調停を管理します。調停には、ラウンドロビン方式を使用します。バスマトリックスは、最大 3 つのマスタ（CPU、DMA1、DMA2）と 4 つのスレーブ（FLITF、SRAM、AHB - APB ブリッジを含む AHB1 および AHB2）で構成されます。

AHB ペリフェラルは、DMA アクセスを可能にするために、バスマトリックスを通じてシステムバスに接続されます。

AHB-APB ブリッジ（APB）

この AHB-APB ブリッジでは、AHB および APB バス間に完全同期接続を提供します。

このブリッジに接続されたペリフェラルのアドレスマッピングについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#)を参照してください。

各デバイスのリセット後、すべてのペリフェラルクロックは無効になります（SRAM とフラッシュは除く）。ペリフェラルを使用する前に、RCC_AHBENR、RCC_APB2ENR、または RCC_APB1ENR レジスタでクロックを有効にする必要があります。

注： **APB レジスタに 16 または 8 ビットアクセスが行われるときには、アクセスは 32 ビットアクセスに変換されます。すなわち、ブリッジが 16 または 8 ビットのデータを複製して、32 ビットのベクタを供給します。**

2.2 メモリ構成

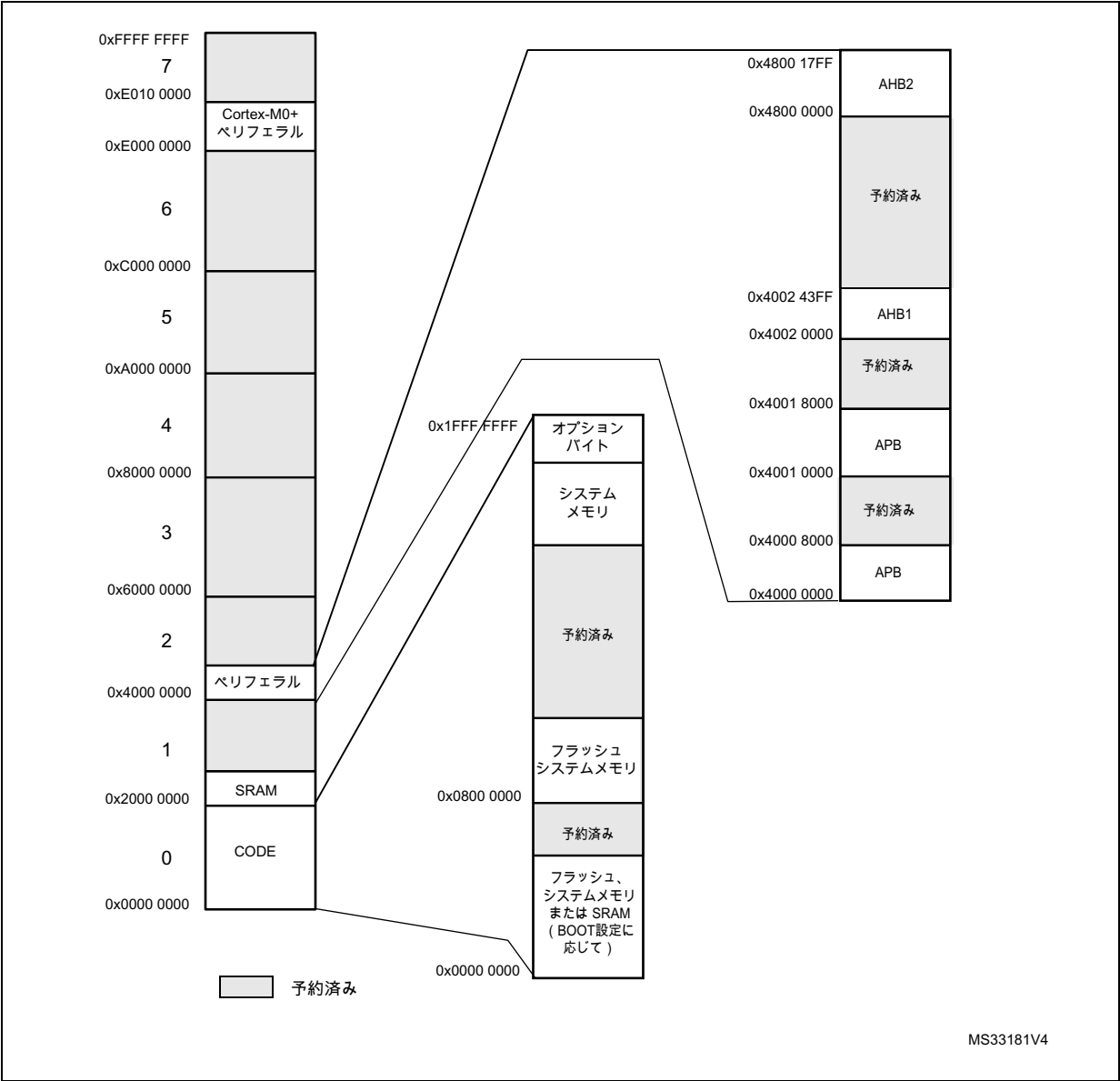
2.2.1 概要

プログラムメモリ、データメモリ、レジスタ、および I/O ポートは、同じ 4 GB のリニアなアドレス空間に配置されています。

バイトは、メモリ内でリトルエンディアン形式でコード化されます。ワード内で最も小さな番号のバイトがワードの最下位バイトとみなされ、最も大きな番号のバイトが最上位バイトとみなされます。

アドレス指定可能なメモリ空間は、それぞれ 512 MB の 8 つのメインブロックに分割されています。

図 2. メモリマップ



オンチップメモリとペリフェラルに割り当てられていないメモリ領域はすべて、「予約済み」とみなされます。使用可能なメモリ領域とレジスタ領域の詳細なマッピングについては、[メモリマップとレジスタ境界アドレス](#)およびペリフェラルの章を参照してください。

2.2.2 メモリマップとレジスタ境界アドレス

包括的なメモリマッピング図については、お使いのデバイスに対応するデータシートを参照してください。

次の表に、デバイスで使用可能なペリフェラルの境界アドレスを示します。

表 1. STM32F0xx ペリフェラルレジスタ境界アドレス

バス	境界アドレス	サイズ	ペリフェラル	ペリフェラルレジスタマップ
	0xE000 0000 - 0xE00F FFFF	1 MB	Cortex®-M0 内部ペリフェラル	
	0x4800 1800 - 0x5FFF FFFF	~384 MB	予約済み	
AHB2	0x4800 1400 - 0x4800 17FF	1 KB	GPIOF	セクション 9.4.12 (171 ページ)
	0x4800 1000 - 0x4800 13FF	1 KB	GPIOE	セクション 9.4.12 (171 ページ)
	0x4800 0C00 - 0x4800 0FFF	1KB	GPIOD	セクション 9.4.12 (171 ページ)
	0x4800 0800 - 0x4800 0BFF	1 KB	GPIOC	セクション 9.4.12 (171 ページ)
	0x4800 0400 - 0x4800 07FF	1 KB	GPIOB	セクション 9.4.12 (171 ページ)
	0x4800 0000 - 0x4800 03FF	1 KB	GPIOA	セクション 9.4.12 (171 ページ)
	0x4002 4400 - 0x47FF FFFF	~128 MB	予約済み	
AHB1	0x4002 4000 - 0x4002 43FF	1 KB	TSC	セクション 28.6.11 (821 ページ)
	0x4002 3400 - 0x4002 3FFF	3 KB	予約済み	
	0x4002 3000 - 0x4002 33FF	1 KB	CRC	セクション 5.4.6 (85 ページ)
	0x4002 2400 - 0x4002 2FFF	3 KB	予約済み	
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH インタフェース	セクション 3.5.9 (74 ページ)
	0x4002 1400 - 0x4002 1FFF	3 KB	予約済み	
	0x4002 1000 - 0x4002 13FF	1 KB	RCC	セクション 7.4.15 (142 ページ)
	0x4002 0800 - 0x4002 0FFF	2 KB	予約済み	
	0x4002 0400 - 0x4002 07FF	1 KB	DMA2	セクション 11.4.8 (215 ページ)
	0x4002 0000 - 0x4002 03FF	1 KB	DMA	セクション 11.4.8 (215 ページ)
	0x4001 8000 - 0x4001 FFFF	32 KB	予約済み	

表 1. STM32F0xx ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ	ペリフェラル	ペリフェラルレジスタマップ
APB	0x4001 5C00 - 0x4001 7FFF	9 KB	予約済み	
	0x4001 5800 - 0x4001 5BFF	1 KB	DBGMCU	セクション 32.9.6 (935 ページ)
	0x4001 4C00 - 0x4001 57FF	3 KB	予約済み	
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17	セクション 19.6.16 (536 ページ)
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16	セクション 19.6.16 (536 ページ)
	0x4001 4000 - 0x4001 43FF	1 KB	TIM15	セクション 19.5.18 (516 ページ)
	0x4001 3C00 - 0x4001 3FFF	1 KB	予約済み	
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1	セクション 26.8.12 (743 ページ)
	0x4001 3400 - 0x4001 37FF	1 KB	予約済み	
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1/I2S1	セクション 27.9.10 (802 ページ)
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1	セクション 16.4.21 (375 ページ)
	0x4001 2800 - 0x4001 2BFF	1 KB	予約済み	
	0x4001 2400 - 0x4001 27FF	1 KB	ADC	セクション 13.12.11 (268 ページ)
	0x4001 2000 - 0x4001 23FF	2 KB	予約済み	
	0x4001 1C00 - 0x4001 1FFF	1 KB	USART8	セクション 26.8.12 (743 ページ)
	0x4001 1800 - 0x4001 1BFF	1 KB	USART7	セクション 26.8.12 (743 ページ)
	0x4001 1400 - 0x4001 17FF	1 KB	USART6	セクション 26.8.12 (743 ページ)
	0x4001 0800 - 0x4001 13FF	2 KB	予約済み	
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI	セクション 12.3.7 (228 ページ)
	0x4001 0000 - 0x4001 03FF	1 KB	SYSCFG COMP	セクション 10.1.38 (194 ページ) セクション 15.5.2 (303 ページ)
	0x4000 8000 - 0x4000 FFFF	32 KB	予約済み	

表 1. STM32F0xx ペリフェラルレジスタ境界アドレス (続き)

バス	境界アドレス	サイズ	ペリフェラル	ペリフェラルレジスタマップ
APB	0x4000 7C00 - 0x4000 7FFF	1 KB	予約済み	
	0x4000 7800 - 0x4000 7BFF	1 KB	CEC	セクション 31.7.7 (920 ページ)
	0x4000 7400 - 0x4000 77FF	1 KB	DAC	セクション 14.8.15 (294 ページ)
	0x4000 7000 - 0x4000 73FF	1 KB	PWR	セクション 6.4.3 (99 ページ)
	0x4000 6C00 - 0x4000 6FFF	1 KB	CRS	セクション 8.6.5 (155 ページ)
	0x4000 6800 - 0x4000 6BFF	1 KB	予約済み	
	0x4000 6400 - 0x4000 67FF	1 KB	CAN	セクション 29.9.5 (864 ページ)
	0x4000 6000 - 0x4000 63FF	1 KB	USB/CAN SRAM	セクション 30.6.3 (899 ページ)
	0x4000 5C00 - 0x4000 5FFF	1 KB	USB	セクション 30.6.3 (899 ページ)
	0x4000 5800 - 0x4000 5BFF	1 KB	I2C2	セクション 25.7.12 (677 ページ)
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1	セクション 25.7.12 (677 ページ)
	0x4000 5000 - 0x4000 53FF	1 KB	USART5	セクション 26.8.12 (743 ページ)
	0x4000 4C00 - 0x4000 4FFF	1 KB	USART4	セクション 26.8.12 (743 ページ)
	0x4000 4800 - 0x4000 4BFF	1 KB	USART3	セクション 26.8.12 (743 ページ)
	0x4000 4400 - 0x4000 47FF	1 KB	USART2	セクション 26.8.12 (743 ページ)
	0x4000 3C00 - 0x4000 43FF	2 KB	予約済み	
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2	セクション 27.9.10 (802 ページ)
	0x4000 3400 - 0x4000 37FF	1 KB	予約済み	
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG	セクション 22.4.6 (561 ページ)
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG	セクション 23.4.4 (567 ページ)
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC	セクション 24.7.18 (607 ページ)
	0x4000 2400 - 0x4000 27FF	1 KB	予約済み	
	0x4000 2000 - 0x4000 23FF	1 KB	TIM14	セクション 18.4.12 (465 ページ)
	0x4000 1800 - 0x4000 1FFF	2 KB	予約済み	
	0x4000 1400 - 0x4000 17FF	1 KB	TIM7	セクション 20.4.9 (551 ページ)
	0x4000 1000 - 0x4000 13FF	1 KB	TIM6	セクション 20.4.9 (551 ページ)
	0x4000 0800 - 0x4000 0FFF	2 KB	予約済み	
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3	セクション 17.4.19 (442 ページ)
	0x4000 0000 - 0x4000 03FF	1 KB	TIM2	セクション 17.4.19 (442 ページ)

表 2. STM32F0xx メモリ境界アドレス

デバイス	境界アドレス	サイズ	メモリ領域	レジスタの説明
STM32F03x	0x2000 1000 - 0x3FFF FFFF	~512 MB	予約済み	
	0x2000 0000 - 0x2000 0FFF	4 KB	SRAM	セクション 2.3 (51 ページ)
	0x1FFF FC00 - 0x1FFF FFFF	1 KB	予約済み	
	0x1FFF F800 - 0x1FFF FBFF	1 KB	オプションバイト	セクション 4 (75 ページ)
	0x1FFF EC00 - 0x1FFF F7FF	3 KB	システムメモリ	
	0x0800 8000 - 0x1FFF EBFF	~384 MB	予約済み	
	0x0800 0000 - 0x0800 7FFF	32 KB	メインフラッシュメモリ	セクション 3 (55 ページ)
	0x0000 8000 - 0x07FF FFFF	~128 MB	予約済み	
	0x0000 0000 - 0x0000 7FFF	32 KB	メインフラッシュメモリ、システムメモリまたは SRAM (BOOT 設定に応じて)	
STM32F04x	0x2000 1800 - 0x3FFF FFFF	~512 MB	予約済み	
	0x2000 0000 - 0x2000 17FF	6 KB	SRAM	セクション 2.3 (51 ページ)
	0x1FFF FC00 - 0x1FFF FFFF	1 KB	予約済み	
	0x1FFF F800 - 0x1FFF FBFF	1 KB	オプションバイト	セクション 4 (75 ページ)
	0x1FFF C400 - 0x1FFF F7FF	13 KB	システムメモリ	
	0x0801 8000 - 0x1FFF C7FF	~384 MB	予約済み	
	0x0800 0000 - 0x0801 7FFF	32 KB	メインフラッシュメモリ	セクション 3 (55 ページ)
	0x0001 8000 - 0x07FF FFFF	~128 MB	予約済み	
	0x0000 0000 - 0x0000 7FFF	32 KB	メインフラッシュメモリ、システムメモリまたは SRAM (BOOT 設定に応じて)	
STM32F05x	0x2000 2000 - 0x3FFF FFFF	~512 MB	予約済み	
	0x2000 0000 - 0x2000 1FFF	8 KB	SRAM	セクション 2.3 (51 ページ)
	0x1FFF FC00 - 0x1FFF FFFF	1 KB	予約済み	
	0x1FFF F800 - 0x1FFF FBFF	1 KB	オプションバイト	セクション 4 (75 ページ)
	0x1FFF EC00 - 0x1FFF F7FF	3 KB	システムメモリ	
	0x0801 0000 - 0x1FFF EBFF	~384 MB	予約済み	
	0x0800 0000 - 0x0800 FFFF	64 KB	メインフラッシュメモリ	セクション 3 (55 ページ)
	0x0001 0000 - 0x07FF FFFF	~128 MB	予約済み	
	0x0000 0000 - 0x0000 FFFF	64 KB	メインフラッシュメモリ、システムメモリまたは SRAM (BOOT 設定に応じて)	

表 2. STM32F0xx メモリ境界アドレス（続き）

デバイス	境界アドレス	サイズ	メモリ領域	レジスタの説明
STM32F07x	0x2000 4000 - 0x3FFF FFFF	~512 MB	予約済み	
	0x2000 0000 - 0x2000 3FFF	16 KB	SRAM	セクション 2.3 (51 ページ)
	0x1FFF F800 - 0x1FFF FFFF	2 KB	オプションバイト	セクション 4 (75 ページ)
	0x1FFF C800 - 0x1FFF F7FF	12 KB	システムメモリ	
	0x0802 0000 - 0x1FFF C7FF	~384 MB	予約済み	
	0x0800 0000 - 0x0801 FFFF	128 KB	メインフラッシュメモリ	セクション 3 (55 ページ)
	0x0002 0000 - 0x07FF FFFF	~128 MB	予約済み	
	0x0000 0000 - 0x0001 FFFF	128 KB	メインフラッシュメモリ、システムメモリまたは SRAM (BOOT 設定に応じて)	
STM32F09x	0x2000 8000 - 0x3FFF FFFF	~512 MB	予約済み	
	0x2000 0000 - 0x2000 7FFF	32 KB	SRAM	セクション 2.3 (51 ページ)
	0x1FFF F800 - 0x1FFF FFFF	2 KB	オプションバイト	セクション 4 (75 ページ)
	0x1FFF D800 - 0x1FFF F7FF	8 KB	システムメモリ	
	0x0804 0000 - 0x1FFF D7FF	~384 MB	予約済み	
	0x0800 0000 - 0x0803 FFFF	256 KB	メインフラッシュメモリ	セクション 3 (55 ページ)
	0x0004 0000 - 0x07FF FFFF	~128 MB	予約済み	
	0x0000 0000 - 0x0003 FFFF	256 KB	メインフラッシュメモリ、システムメモリまたは SRAM (BOOT 設定に応じて)	

2.3 内蔵 SRAM

STM32F03x デバイスには、4 KB の SRAM を搭載しています。STM32F04x デバイスには、6 KB の SRAM を搭載しています。STM32F05x デバイスには、8 KB の SRAM を搭載しています。STM32F07xS デバイスには、16 KB の SRAM を搭載しています。STM32F09x デバイスには、32 KB の SRAM を搭載しています。

この SRAM には、バイト、ハーフワード（16 ビット）、またはフルワード（32 ビット）によるアクセスが可能です。このメモリは、最大システムクロック周波数（ウェイトステートなし）、つまり CPU と DMA の両方でアドレス指定できます。

パリティチェック

ユーザは、ユーザオプションバイトのオプションビット RAM_PARITY_CHECK を使用してパリティチェックを有効にできます（[セクション 4：オプションバイト](#)を参照）。

パリティチェックに使用できるのは 4 ビットであるため（1 バイトあたり 1 ビット）、データバス幅は 36 ビットです。これは、Class B や SIL 標準などで必要とされるメモリの信頼性を高めるためです。

パリティビットは、SRAM に書き込む際に計算され、格納されます。その後、読み出すときに自動的にチェックされます。1 つのビットがフェイルすると、NMI が生成されます。同じエラーが、[SYSCFG 設定レジスタ 2 \(SYSCFG_CFGR2\)](#) の SRAM_PARITY_LOCK の制御ビットにより、TIMER15/TIM16/TIM7 の BRK_IN ブレーク入力にリンクされている場合もあります。SRAM パリティエラーフラグ（SRAM_PEF）は[SYSCFG 設定レジスタ 2 \(SYSCFG_CFGR2\)](#) で使用可能です。

注： RAM パリティチェックを有効にするときに、コードの冒頭でソフトウェアを使用して RAM メモリ全体を初期化することを推奨します。これにより、初期化されていない位置を読み出す際にパリティエラーの発生を防ぐことができます。

2.4 フラッシュメモリの概要

フラッシュメモリは 2 つの別個の物理領域で構成されます。

- メインフラッシュメモリブロック。必要に応じてアプリケーションプログラムとユーザデータを含みます。
- 情報ブロック。2 つの部分で構成されます。
 - － ハードウェアとメモリ保護のユーザ設定のオプションバイト。
 - － 独自のブートローダコードを含むシステムメモリ。詳細については、[セクション 3：内蔵フラッシュメモリ](#)を参照してください。

フラッシュインタフェースは、AHB プロトコルに基づいて命令アクセスとデータアクセスを実装します。CPU コードの実行を加速するプリフェッチバッファを実装します。また、フラッシュメモリ操作（プログラム／消去）を実行するために必要な、フラッシュレジスタを使って制御されるロジックも実装します。

2.5 ブート設定

STM32F0xx では、BOOT0 ピンとブート設定ビット nBOOT1、BOOT_SEL および nBOOT0 のユーザオプションバイトに含まれるビットによって、次の表に示すように 3 種類のブートモードを選択できます。

表 3. ブートモード (1)

ブートモード設定				モード
nBOOT1 ビット	BOOT0 ピン	BOOT_SEL ビット	nBOOT0 ビット	
x	0	1	x	メインフラッシュメモリがブート領域として選択されます。 ⁽²⁾
1	1	1	x	システムメモリがブート領域として選択されます。
0	1	1	x	内蔵 SRAM がブート領域として選択されます。
x	x	0	1	メインフラッシュメモリがブート領域として選択されます。
1	x	0	0	システムメモリがブート領域として選択されます。
0	x	0	0	内蔵 SRAM がブート領域として選択されます。

- グレイオプションは STM32F04x および STM32F09x デバイスでのみ使用可能です。
- STM32F04x および STM32F09x デバイスについては、エンプティチェックの説明も参照してください。

ブートモード設定は、リセット後、SYSCLK の 4 番目の立ち上がりエッジでラッチされます。ユーザは、必要なブートモードに関連するブートモード設定をセットします。

ブートモード設定も、STANDBY モードの終了時に再サンプリングされます。したがって、これらのピンは STANDBY モードのときでも必要なブートモード設定に保たれる必要があります。この起動遅延時間が終了すると、CPU はアドレス 0x0000 0000 からスタック最上位の値をフェッチし、0x0000 0004 のブートメモリからコード実行を開始します。

選択したブートメモリに応じて、メインフラッシュメモリ、システムメモリ、または SRAM に、次のようにアクセスできます。

- メインフラッシュメモリからブート：メインフラッシュメモリのエイリアスがブートメモリ空間（0x0000 0000）に作成されますが、引き続き元のメモリ空間（0x0800 0000）からアクセスすることも可能です。言い換えると、フラッシュメモリの内容は、0x0000 0000 または 0x0800 0000 から始まるアドレスからアクセスできます。
- システムメモリからブート：システムメモリのエイリアスがブートメモリ空間（0x0000 0000）に作成されますが、引き続き元のメモリ空間（STM32F03x および STM32F05x デバイスの 0x1FFF EC00、STM32F04x デバイスの 0x1FFF C400、STM32F07x デバイスの 0x1FFF C800 および STM32F09x デバイスの 0x1FFF D800）からアクセスすることも可能です。
- 内蔵 SRAM からブート：SRAM のエイリアスがブートメモリ空間（0x0000 0000）に作成されますが、引き続き元のメモリ空間（0x2000 0000）からアクセスすることも可能です。

エンプティチェック

内部エンプティチェックフラグは、ブートローダによる未使用デバイスの簡単なプログラミングを可能にするために、STM32F03x および STM32F09x デバイスにのみ実装されています。このフラグは、BOOT0 ピンがターゲットブート領域としてメインフラッシュメモリで定義されている場合に使用されます。フラグをセットするとデバイスはエンプティとみなされ、ユーザがフラッシュメモリをプログラムできるよう、ブート領域としてメインフラッシュの代わりにシステムメモリ（ブートローダ）が選択されます。

このフラグはオプションバイトのロード中に更新されます。アドレス 0x08000 0000 が 0xFFFF FFFF として読み出された場合にのみセットされ、それ以外の場合はクリアされます。システムリセット後にユーザコードを実行するには、未使用デバイスのプログラミング後に、このフラグをクリアするために FLASH_CR レジスタの OBL_LAUNCH ビットをパワーオンまたは設定する必要があるということです。

注： デバイスを初めてプログラミングする際にオプションバイトが再ロードされていない場合、デバイスではシステムリセット後も引き続きブート領域としてシステムメモリを選択します。ブートローダコードではこの状況を検出することができるため、ブートメモリマッピングをメインフラッシュに変更し、そこでプログラミングされたユーザコードにジャンプします。

物理的な再割当て

ブートモードを選択すると、アプリケーションソフトウェアはコード領域でアクセス可能なメモリを変更できます。この変更は、SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1) で MEM_MODE ビットをプログラミングすることによって実行できます。Cortex® M3 および M4 とは異なり、M0 CPU ではベクタテーブルの再配置をサポートしていません。0x0800 0000 とは別のアドレスに配置されたアプリケーションコードでアプリケーション割り込みを処理するには、なんらかの追加コードが必要です。1つの方法として、内部 SRAM のベクタテーブルをソフトウェアで再配置することが考えられます。

- 0x2000 0000 で、フラッシュ（アプリケーションロードアドレスのベースにマッピングされた）から SRAM のベースアドレスにベクタテーブルをコピーします。
- SYSCFG 設定レジスタ 1 を使用して、アドレス 0x0000 0000 に SRAM を再配置します。
- 割り込みが発生すると、Cortex®-M0 プロセッサは SRAM の再配置されたベクタテーブルから割り込みハンドラの開始アドレスをフェッチし、フラッシュに配置された割り込みハンドラを実行するためにジャンプします。

この操作は、アプリケーションの初期化フェーズで実行してください。詳細については、AN4065 および付属の www.st.com の IAP コードを参照してください。

内蔵ブートローダ

内蔵ブートローダは、システムメモリに配置されるよう生産時に ST によってプログラムされています。次のシリアルインタフェースのいずれかを使用して、フラッシュメモリを再プログラムするために使用します。

- ピン PA14/PA15 または PA9/PA10 上の USART
- ピン PB6/PB7 上の I2C (STM32F07xxx および STM32F09x デバイスのみ)
- USB DFU インタフェース (STM32F07xxx デバイスのみ)

詳細については、AN2606 を参照してください。

3 内蔵フラッシュメモリ

3.1 フラッシュの主な機能

- 最大 256 KB のフラッシュメモリ
- メモリ構成：
 - － メインフラッシュメモリブロック：
最大 64 K ワード (64 K × 32 ビット)
 - － 情報ブロック：
最大 3 K ワード (3 K × 32 ビット) のシステムメモリ
 - － 最大 2 × 8 バイトのオプションバイト

フラッシュメモリインタフェースの機能：

- プリフェッチバッファ機能付き読み出しインタフェース
- オプションバイトローダ
- フラッシュのプログラム／消去操作
- 読み出し／書き込み保護
- 低電力モード

3.2 フラッシュメモリの機能詳細

3.2.1 フラッシュメモリの構成

フラッシュメモリは、32 ビット幅のメモリセルとして構成されており、コードとデータの両方の定数を格納するために使用できます。

STM32F03x、STM32F04x、および STM32F05x デバイスのメモリ構成は、1 KB のページを最大 64 ページ、または、4 KB のセクタ (4 ページ) を最大 16 セクタ含むメインフラッシュメモリブロックに基づいています。セクタは書き込み保護の最小単位です ([セクション 3.3 : メモリ保護 \(65 ページ\)](#) を参照)。

STM32F07x および STM32F09x デバイスのメモリ構成は、2 KB のページを最大 128 ページ、または、4 KB のセクタ (2 ページ) を最大 64 セクタ含むメインフラッシュメモリブロックに基づいています。書き込み保護の最小単位はセクタです ([セクション 3.3 : メモリ保護 \(65 ページ\)](#) を参照)。

情報ブロックは、2 つの部分に分かれます。

1. システムメモリ：システムメモリブートモードでデバイスをブートするために使用します。この領域は STMicroelectronics によって予約済みで、選択した通信インタフェースを介したフラッシュメモリの再プログラムに使用するブートローダを含みます。これはデバイスの製造時に ST によってプログラムされており、誤った書き込み／消去操作から保護されています。詳細については、AN2606 を参照してください。
2. オプションバイト

表 4. フラッシュメモリの構成 (STM32F03x、STM32F04x および STM32F05x デバイス)

フラッシュ領域	フラッシュメモリアドレス	サイズ (バイト)	名前	説明
メインフラッシュ メモリ	0x0800 0000 - 0x0800 03FF	1 KB	ページ 0	セクタ 0
	0x0800 0400 - 0x0800 07FF	1 KB	ページ 1	
	0x0800 0800 - 0x0800 0BFF	1 KB	ページ 2	
	0x0800 0C00 - 0x0800 0FFF	1 KB	ページ 3	
	.	.	.	.
	.	.	.	.
	.	.	.	.
	0x0800 7000 - 0x0800 73FF	1 KB	ページ 28	セクタ 7 ⁽¹⁾
	0x0800 7400 - 0x0800 77FF	1 KB	ページ 29	
	0x0800 7800 - 0x0800 7BFF	1 KB	ページ 30	
	0x0800 7C00 - 0x0800 7FFF	1 KB	ページ 31	
	.	.	.	.
	.	.	.	.
	.	.	.	.
	0x0800 F000 - 0x0800 F3FF	1 KB	ページ 60	セクタ 15
	0x0800 F400 - 0x0800 F7FF	1 KB	ページ 61	
	0x0800 F800 - 0x0800 FBFF	1 KB	ページ 62	
	0x0800 FC00 - 0x0800 FFFF	1 KB	ページ 63	
情報ブロック	0x1FFF EC00 - 0x1FFF F7FF	3 KB ⁽²⁾	-	システムメモリ
	0x1FFF C400 - 0x1FFF F7FF	13 KB ⁽³⁾	-	システムメモリ
	0x1FFF F800 - 0x1FFF F80F	2 × 8 バイト	-	オプションバイト

- 1. STM32F03x および STM32F04x デバイスのメインフラッシュメモリ領域は、セクタ 7までに制限されます。
- 2. STM32F03x および STM32F05x デバイス
- 3. STM32F04x デバイス



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 5. フラッシュメモリの構成 (STM32F07x、STM32F09x デバイス)

フラッシュ領域	フラッシュメモリアドレス	サイズ (バイト)	名前	説明
メインフラッシュ メモリ	0x0800 0000 - 0x0800 07FF	2 KB	ページ 0	セクタ 0
	0x0800 0800 - 0x0800 0FFF	2 KB	ページ 1	
	.	.	.	
	0x0801 F000 - 0x0801 F7FF	2 KB	ページ 62	セクタ 31 ⁽¹⁾
	0x0801 F800 - 0x0801 FFFF	2 KB	ページ 63	
	.	.	.	.
	0x0803 F000 - 0x0803 F7FF	2 KB	ページ 126	-
	0x0803 F800 - 0x0803 FFFF	2 KB	ページ 127	-
	0x1FFF C800 - 0x1FFF F7FF	12 KB ⁽²⁾	-	システムメモリ
	0x1FFF D800 - 0x1FFF F7FF	8 KB ⁽³⁾	-	システムメモリ
情報ブロック	0x1FFF F800 - 0x1FFF F80F	2 × 8 バイト	-	オプションバイト

1. STM32F07x のメインフラッシュメモリ領域は、セクタ 31 までに制限されます。
2. STM32F07x デバイスのみ。
3. STM32F09x デバイスのみ。

読み出し操作

内蔵フラッシュモジュールは、共通メモリ空間として直接アドレス指定できます。任意のデータ読み出し操作が専用の読み出し機構を通じてフラッシュモジュールの内容にアクセスし、リクエストされたデータを提供します。

命令フェッチとデータアクセスは、両方とも同じ AHB バスを通じて行われます。読み出しアクセスは、フラッシュアクセス制御レジスタ (FLASH_ACR) を通じて管理される次のオプションで行うことができます。

- 命令フェッチ：高速 CPU 実行用のプリフェッチバッファ
- 遅延：正確な読み出し操作に必要なウェイトステート数 (0 から 1 の間)

命令フェッチ

Cortex®-M0は AHB バスを通じて命令をフェッチします。プリフェッチブロックは、命令フェッチの効率の向上を目的としています。

プリフェッチバッファ

プリフェッチバッファの幅はブロック 3 つ分で、各ブロックは 4 バイトで構成されます。プリフェッチブロックは直接マッピングされています。ブロックは、そのサイズがフラッシュメモリの帯域幅と一致しているため、シングル読み出しでフラッシュメモリと完全に置き換えることができます。

このプリフェッチバッファの実装により、CPU の高速実行が可能になります。これは、CPU が一度に 1 ワードずつフェッチしますが、プリフェッチバッファに次のワードがすぐに使用可能な状態で、準備できているためです。これは、コードがジャンプのために 64 ビットの境界に整列されている場合、加速度の比率が 2 次となることを意味しています。



ただし、プリフェッチバッファはウェイトステート数が 1 の場合にのみパフォーマンスに影響を与えます。別のケース（ウェイトステートなし）では、プリフェッチバッファのステータスにかかわらず、パフォーマンスは維持されます。消費電力になんらかの影響を与える可能性があります、実際のアプリケーションコードに強く依存しています。

プリフェッチコントローラ

プリフェッチコントローラは、プリフェッチバッファで使用可能な空間に応じてフラッシュメモリにアクセスするかどうかを決定します。コントローラは、プリフェッチバッファで空いているブロックが少なくとも 1 つある場合に、読み出しリクエストを開始します。

リセット後のプリフェッチバッファの状態はオンです。

マイクロコントローラが内部 8 MHz の RC (HSI) オシレータで実行している間、プリフェッチバッファは初期化ルーチン中に通常オン/オフに切り替えられます。

アクセスの遅延

フラッシュメモリを読み出すための制御信号を維持するには、フラッシュアクセス制御レジスタの LATENCY[2:0] ビットで、プリフェッチコントローラのクロック周期に対するフラッシュメモリのアクセス時間の比率をプログラムする必要があります。この値によってフラッシュメモリの制御信号の維持に必要なサイクル数を得ることができ、また要求されたデータを正しく読み出すことができます。リセット後、値はゼロになり、フラッシュメモリには 1 サイクルのみでアクセスできます。追加ウェイトステートは不要です。

3.2.2 フラッシュのプログラムおよび消去操作

STM32F0xx 内蔵フラッシュメモリは、インサーキットプログラミングまたはアプリケーション内プログラミングを使用してプログラム可能です。

インサーキットプログラミング (ICP) 方法は、フラッシュメモリの内容全体を更新するために使用します。マイクロコントローラにユーザアプリケーションをロードするために SWD プロトコルやブートローダを使用します。ICP は、迅速かつ効率的な設計時の繰り返し操作を提供し、不要なバッファ処理やデバイスのソケット処理を排除できます。

ICP の方法とは対照的に、**アプリケーション内プログラミング (IAP)** では、メモリにプログラミングデータをダウンロードするために、マイクロコントローラでサポートされる任意の通信インタフェース (I/O、USB、CAN、USART、I²C、SPI など) を使用できます。IAP を使用して、ユーザはアプリケーション実行中にフラッシュメモリを再プログラムできます。ただし、アプリケーションの一部は ICP を使用してフラッシュメモリ内に事前にプログラムする必要があります。

プログラムおよび消去操作は製品電圧範囲全体にわたって実行できます。これらは、次の 7 つのフラッシュ関連のレジスタを通じて管理されます。

- キーレジスタ (FLASH_KEYR)
- オプションバイトキーレジスタ (FLASH_OPTKEYR)
- フラッシュ制御レジスタ (FLASH_CR)
- フラッシュステータスレジスタ (FLASH_SR)
- フラッシュアドレスレジスタ (FLASH_AR)
- オプションバイトレジスタ (FLASH_OBR)
- 書き込み保護レジスタ (FLASH_WRP)

進行中のフラッシュメモリ操作は、CPU がフラッシュメモリにアクセスしない限り、CPU をブロックしません。

逆に、フラッシュメモリへのプログラム／消去操作中にフラッシュメモリを読み出そうとすると、バスがストールされます。読み出し操作は、プログラム／消去操作が完了すると正しく処理されます。これは、プログラム／消去操作中はコードやデータをフェッチできないことを意味します。

フラッシュメモリ上でのプログラムおよび消去操作（書き込み／消去）では、内部 RC オシレータ（HSI）をオンにする必要があります。

フラッシュメモリのロック解除

リセット後、フラッシュメモリは不要な書き込みまたは消去操作から保護されます。FLASH_CR レジスタは、オプションビットの再ロードに使用する OBL_LAUNCH ビットを除き、書き込みモードではアクセスできません。FLASH_CR レジスタへのアクセスを開くには、FLASH_KEYR レジスタにロック解除シーケンスを書き込む必要があります。シーケンスは 2 つの書き込み操作で構成されます。

- KEY1 = 0x45670123を書き込む
- KEY2 = 0xCDEF89ABを書き込む

シーケンスを誤ると、次のリセットまで FLASH_CR レジスタがロックされます。

キーシーケンスを誤ると、バスエラーが検出され、ハードフォールト割り込みが発生します。これは最初の書き込みサイクルの後で KEY1 が一致しない場合、または 2 番目の書き込みサイクルで KEY1 が正しく書き込まれるが KEY2 が一致しない場合に行われます。

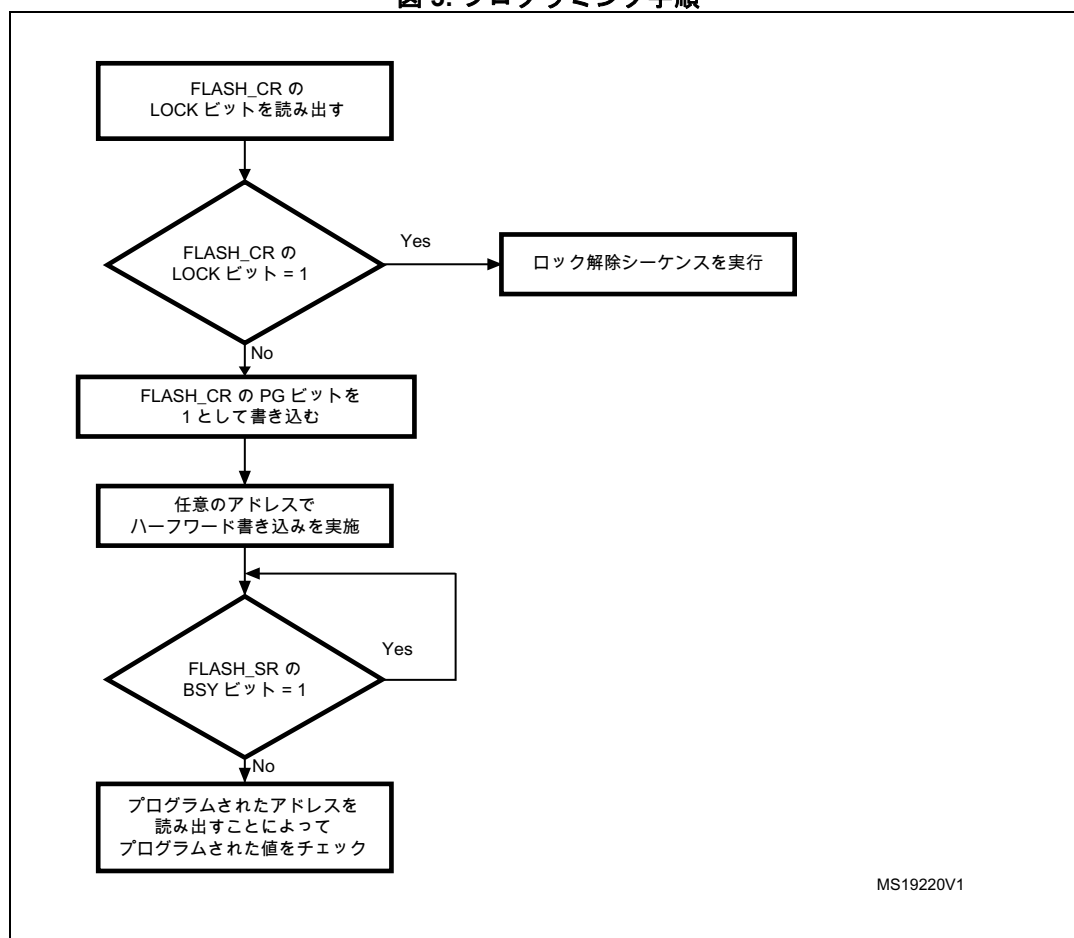
FLASH_CR レジスタは、ユーザソフトウェアで FLASH_CR レジスタの LOCK ビットに 1 を書き込むと再びロックできます。

コード例については、「付録」セクション [A.2.1：フラッシュメモリアンロックシーケンスコード](#) を参照してください。

メインフラッシュメモリプログラム

メインフラッシュメモリは、一度に 16 ビットずつプログラム可能です。プログラム操作は、FLASH_CR レジスタの PG ビットがセットされた状態で、CPU によってメインフラッシュメモリアドレスにハーフワードが書き込まれた場合に開始します。ハーフワード長以外のデータの書き込みは、バスエラーの原因となり、ハードフォールト割り込みが発生します。

図 3. プログラミング手順



フラッシュメモリインタフェースではメインフラッシュメモリのアドレス指定された位置の値を事前に読み出し、それが消去されたことをチェックします。消去されていない場合、プログラム操作はスキップされ、FLASH_SR レジスタの PGERR ビットによって警告が発行されます。唯一の例外は、0x0000 がプログラムされている場合です。この場合、位置は 0x0000 に正しくプログラムされ、PGERR ビットはセットされていません。

メインフラッシュメモリのアドレス指定された位置が FLASH_WRPR レジスタによって書き込み保護されている場合、プログラム操作はスキップされ、FLASH_SR レジスタの WRPRTERR ビットによって警告が発行されます。プログラム操作の終了は、FLASH_SR レジスタの EOP ビットによって示されます。

標準モードのメインフラッシュメモリのプログラミングシーケンスは、次のようになっています。

1. FLASH_SR レジスタの BSY ビットを確認し、進行中のメインフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの PG ビットをセットします。
3. 任意のアドレスでデータ書き込み（ハーフワード）を実施します。
4. FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます。
5. FLASH_SR レジスタの EOP フラグ（プログラミング操作が成功した場合にセットされる）をチェックし、その後ソフトウェアによってクリアします。

注： FLASH_SR レジスタの BSY ビットがセットされていると、書き込みモードではレジスタにアクセスできません。

コード例については、「付録」セクション [A.2.2：メインフラッシュプログラミングシーケンスのコード例](#) を参照してください。

フラッシュメモリの消去

フラッシュメモリはページごとに、または完全（全体消去）に消去できます。

ページ消去

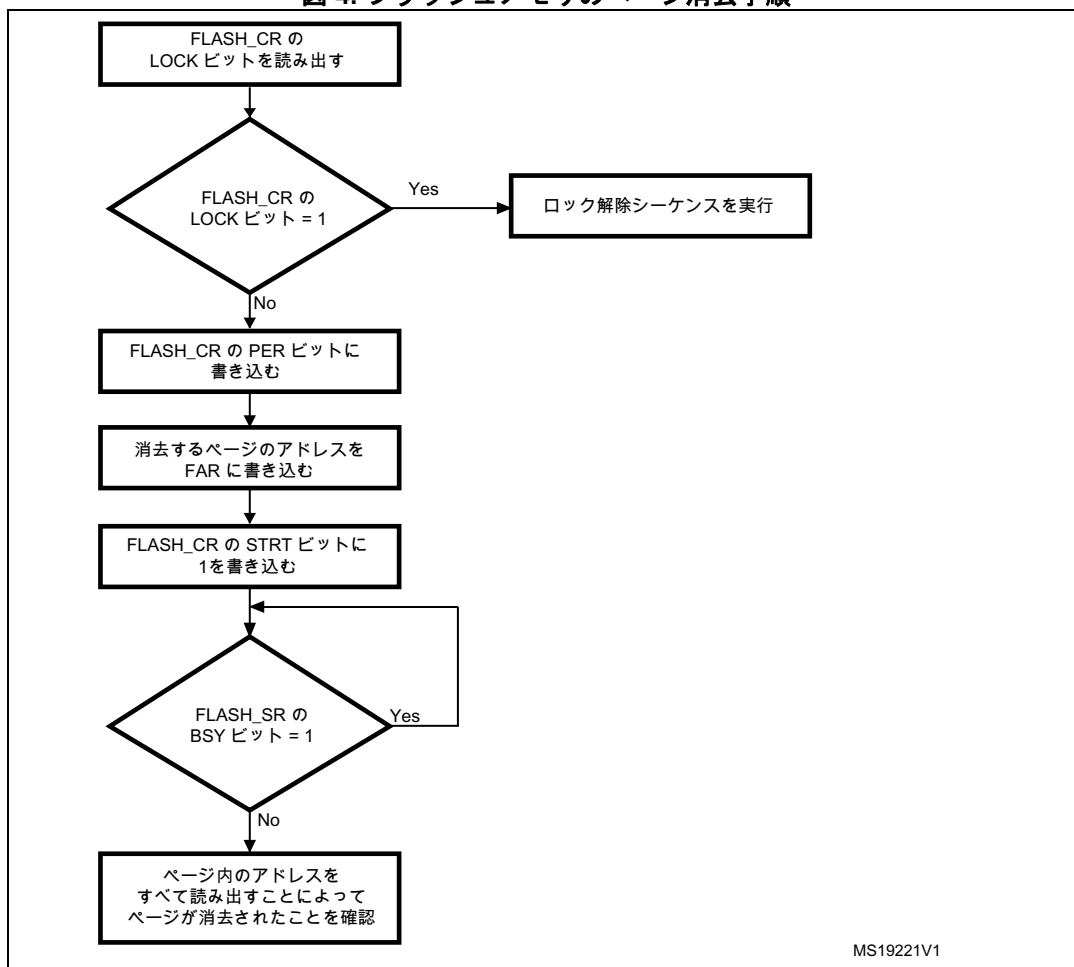
ページを消去するには、次の手順に従います。

1. FLASH_CR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの PER ビットをセットします。
3. 消去するページを選択するために、FLASH_AR レジスタをプログラムします。
4. FLASH_CR レジスタの STRT ビットをセットします（下の注を参照）。
5. BSY ビットがリセットされるのを待ちます。
6. FLASH_SR レジスタの EOP フラグ（消去操作が成功した場合にセットされる）をチェックします。
7. EOP フラグをクリアします。

注： STRT ビット設定後、少なくとも 1 CPU サイクル後に BSY ビットが「0」と等しいかどうかを、ソフトウェアでチェックを開始する必要があります。

コード例については、「付録」セクション [A.2.3：ページ消去シーケンスのコード例](#) を参照してください。

図 4. フラッシュメモリのページ消去手順



全体消去

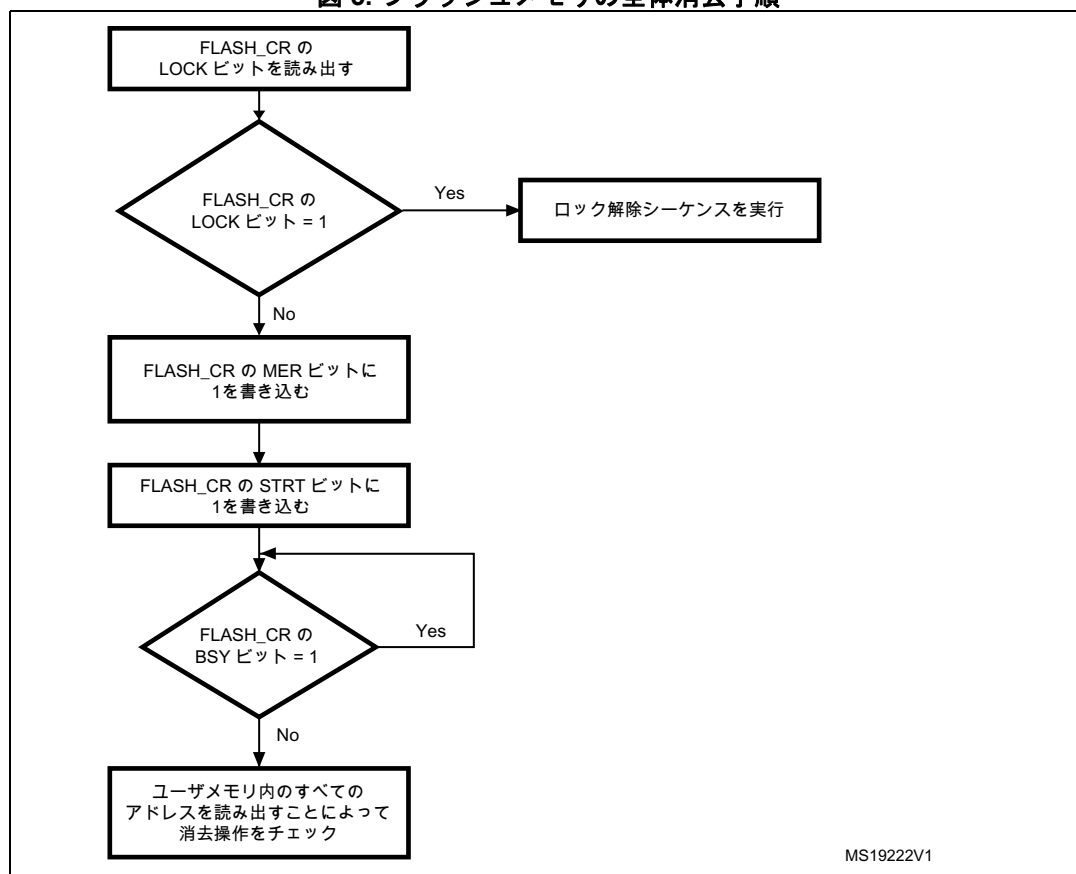
全体消去コマンドは、メインフラッシュメモリのすべてのページを完全に消去する場合に使用できます。情報ブロックは、この手順の影響を受けません。次のシーケンスを推奨します。

1. FLASH_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの MER ビットをセットします。
3. FLASH_CR レジスタの STRT ビットをセットします。
4. FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます。
5. FLASH_SR レジスタの EOP フラグをチェックします（プログラム操作が成功した場合にセットされる）。
6. EOP フラグをクリアします。

注： **STRT ビット設定後、少なくとも 1 CPU サイクル後に BSY ビットが「0」と等しいかどうかを、ソフトウェアでチェックを開始する必要があります。**

コード例については、「付録」セクション [A.2.4: 全体消去シーケンスのコード例](#) を参照してください。

図 5. フラッシュメモリの全体消去手順



オプションバイトのプログラミング

オプションバイトは、通常のユーザアドレスとは異なる方法でプログラムされます。オプションバイト数は、8 に制限されます（1、2、または 4 バイトは書き込み保護、1 バイトは読み出し保護、1 バイトはハードウェア設定、2 空きバイトはユーザデータ）。フラッシュアクセスのロック解除後、ユーザは FLASH_CR レジスタの OPTWRE ビットをセットするために FLASH_OPTKEYR レジスタに同じ KEYS のセット（KEY1 および KEY2）を書き込むことでオプションバイトのプログラミングを許可する必要があります（キー値については [フラッシュメモリのロック解除](#) を参照）。次に、ユーザは任意のフラッシュアドレスで、FLASH_CR レジスタの OPTPG ビットをセットし、ハーフワードの書き込み操作を実施する必要があります。

本当に消去されたことをチェックするために、アドレス指定されたオプションバイトの値が最初に読み出されます。消去されていない場合、プログラム操作はスキップされ、FLASH_SR レジスタの WRPRERR ビットによって警告が発行されます。プログラム操作の終了は、FLASH_SR レジスタの EOP ビットによって示されます。

オプションバイトは、プログラム操作の開始前に、次のフラッシュメモリアドレスに自動的に補完されます。これにより、オプションバイトとその補数は常に正しくなります。

シーケンスは、次のとおりです。

1. FLASH_SR レジスタの BSY ビットを確認し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの OPTWRE ビットをロック解除します。
3. FLASH_CR レジスタの OPTPG ビットをセットします。
4. 任意のアドレスにデータ（ハーフワード）を書き込みます。
5. BSY ビットがリセットされるのを待ちます。
6. プログラムされた値を読み出し、確認します。

コード例については、「付録」セクション [A.2.6 : オプションバイトのプログラミングシーケンスのコード例](#) を参照してください。

フラッシュメモリの読み出し保護オプションが保護された状態から保護されていない状態に変更された場合、読み出し保護オプションの再プログラミングの前にメインフラッシュメモリの全体消去が実施されます。読み出し保護オプション以外のオプションを変更したい場合は、全体消去は実施されません。読み出し保護オプションバイトが消去されていると、フラッシュメモリは保護状態となります。

消去手順

オプションバイトの消去シーケンスは、次のとおりです。

1. FLASH_SR レジスタの BSY ビットを読み出し、進行中のフラッシュメモリ操作がないことを確認します。
2. FLASH_CR レジスタの OPTWRE ビットをロック解除します。
3. FLASH_CR レジスタの OPTER ビットをセットします。
4. FLASH_CR レジスタの STRT ビットをセットします。
5. BSY ビットがリセットされるのを待ちます。
6. 消去されたオプションバイトを読み出し、検証します。

コード例については、「付録」セクション [A.2.7 : オプションバイトの消去シーケンスのコード例](#) を参照してください。

3.3 メモリ保護

フラッシュメモリのユーザエリアは、信頼されないコードによる読み出しから保護できます。フラッシュメモリのページは、プログラムカウンタの内容の損失による不要な書き込みから保護することもできます。書き込み保護の単位は 1 セクタ（4 ページ）です。

3.3.1 読み出し保護

読み出し保護は、RDP オプションバイトをセットし、新しい RDP オプションバイトを再ロードするシステムリセットを適用することで有効になります。

注： デバッグが SWD を通じて接続されている間に読み出し保護がセットされる場合、システムリセットの代わりに POR（パワーオンリセット）を適用してください。

読み出し保護には、保護なし（レベル 0）から最大保護またはデバッグなし（レベル 2）までの 3 つのレベルがあります。表 7：アクセス状態 対 保護レベルと実行モードを参照してください。

フラッシュメモリは、RDP オプションバイトとその補数に、表 6 に示す値のペアが含まれる場合に保護されます。

表 6. フラッシュメモリの読み出し保護ステータス

RDP バイト値	RDP 補数	読み出し保護レベル
0xAA	0x55	レベル 0（ST 出荷時の設定）
任意の値 0xAA または 0xCC を除く	任意の値 （相補である必要はない） 0x55 および 0x33 を除く	レベル 1
0xCC	0x33	レベル 2

保護レベルにかかわらず、システムメモリ領域は読み出しアクセス可能です。プログラム／消去操作の場合は、アクセスできません。

レベル 0：保護なし

メインフラッシュメモリ領域に、読み出し、プログラム、および消去操作を実行できます。

すべての操作でオプションバイトにもアクセスできます。

レベル 1：読み出し保護

これは、RDP オプションバイトが消去された場合のデフォルトの保護レベルです。RDP 値が 0xAA および 0xCC 以外の任意の値である場合に、または補数が正しくない場合でも、併せて定義されます。

- **ユーザモード：**ユーザモードで実行されるコードにより、すべての操作でメインフラッシュメモリやオプションバイトにアクセスできます。
- **デバッグ、ブート RAM、およびブートローダモード：**SWD を介したデバッグモード、またはコードをブート RAM またはブートローダから実行している場合に、メインフラッシュメモリやバックアップレジスタ（RTC の RTC_BKPxR）には完全にアクセスできません。これらのモードでは、簡単な読み出しアクセスでもバスエラーやハードフォールト割り込みが発生します。メインフラッシュメモリはプログラム／消去保護されており、悪意のある、または許可されていないユーザがダンブルーチンによってユーザコードを再プログラミングするのを防ぎます。任意にプログラム／消去操作が行われると、フラッシュステータスレジスタ（FLASH_SR）の PGERR フラグがセットされます。

レベル 0 に戻すために RDP が値 0xAA に再プログラムされると、メインフラッシュメモリの全体消去が実施され、バックアップレジスタ（RTC の RTC_BKPxR）がリセットされます。



レベル 2 : デバッグなし

このレベルでは、保護レベル 1 が保証されます。さらに、CortexM0 のデバッグ機能が無効化されます。結果として、デバッグポート（SWD）、RAM からのブート（ブート RAM モード）、およびシステムメモリからのブート（ブートローダモード）は使用できなくなります。

ユーザ実行モードでは、メインフラッシュメモリのすべての操作が許可されます。逆に、オプションバイトでは、読み出しおよびプログラム操作のみを実行できます。消去操作では、オプションバイトにアクセスできません。

さらに、RDP バイトはプログラムできません。そのため、レベル 2 は解除する方法がありません。これは、不可逆操作です。RDP バイトをプログラムしようとすると、Flash_SR レジスタの保護エラーフラグ WRPRERR がセットされ、割り込みが生成されます。

注： リセット後、デバッグ機能は無効化されます。
STMicroelectronics は、レベル 2 保護がセットされている不良部品には分析を実施できません。

表 7. アクセス状態 対 保護レベルと実行モード

領域	保護レベル	ユーザ実行			デバッグ／RAM からのブート／システムメモリからのブート		
		読み出し	書き込み	消去	読み出し	書き込み	消去
メインフラッシュメモリ	1	あり	あり	あり	なし	なし	なし ⁽⁴⁾
	2	あり	あり	あり	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾
システムメモリ ⁽²⁾	1	あり	なし	なし	あり	なし	なし
	2	あり	なし	なし	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾
オプションバイト	1	あり	あり ⁽³⁾	あり	あり ⁽⁴⁾	あり ⁽³⁾⁽⁴⁾	あり
	2	あり	あり ⁽⁵⁾	なし	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾
バックアップレジスタ ⁽⁶⁾	1	あり	あり	N/A	なし	なし	なし
	2	あり	あり	N/A	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾

- 1. 保護レベル 2 が有効な場合、デバッグポート、RAM からのブート、およびシステムメモリからのブートは無効化されます。
- 2. システムメモリは、保護レベル（0、1、または 2）および実行モードにかかわらず、読み出しアクセスのみ可能です。
- 3. メインフラッシュメモリは、RDP オプションバイトがレベル 1 からレベル 0（0xAA）に変更されると消去されます。
- 4. RDP レベル 1 がアクティブな場合、内蔵ブートローダではオプションバイトの読み出しまたは書き込みを許可しません。ただし、RDP 保護を解除する場合は除きます（レベル 1 からレベル 0 への移動）。
- 5. RDP バイトを除く、すべてのオプションバイトをプログラムできます。
- 6. これらのレジスタは、RDP のレベル 1 からレベル 0 に移動する場合にのみリセットされます。

読み出し保護レベルの変更

レベル 0 からレベル 1 への移動は、RDP バイトの値を任意の値（0xCC を除く）に変更することで、簡単に行うことができます。

RDP バイトに 0xCC 値をプログラミングすることで、直接レベル 0 またはレベル 1 からレベル 2 に移動することができます。

逆に、レベル 0（保護なし）への変更は、メインフラッシュメモリの全体消去操作なしでは行うことができません。この全体消去は、RDP バイトに 0xAA がプログラムされると直ちに実施されます。

注： 全体消去コマンドが使用されると、バックアップレジスタ（RTC の RTC_BKPxR）もリセットされます。

保護レベルの変更を確認するには、フラッシュ制御レジスタの「OBL_LAUNCH」ビットを通じてオプションバイトを再ロードする必要があります。

3.3.2 書き込み保護

書き込み保護は、1 セクタの単位で実装されています。これは、WRP_x オプションバイトを設定し、FLASH_CR レジスタの OBL_LAUNCH ビットをセットして再ロードすることで有効化されます。

保護されたセクタでプログラムまたは消去操作が実施された場合、フラッシュメモリはフラッシュメモリステータスレジスタ（FLASH_SR）の WRPRERR 保護エラーフラグを返します。

書き込み保護解除

書き込み保護を無効にするために、2 つの応用事例が提供されています。

- ケース 1：書き込み保護解除後に、読み出し保護を無効化します。
 - － フラッシュメモリ制御レジスタ（FLASH_CR）の OPTER ビットを使用して、オプションバイト領域全体を消去します。
 - － メモリの保護を解除するために、RDP バイトにコード 0xAA をプログラムします。この操作では、強制的にメインフラッシュメモリを全体消去します。
 - － フラッシュ制御レジスタ（FLASH_CR）の OBL_LAUNCH ビットをセットしてオプションバイト（および新しい WRP[1:0] バイト）を再ロードし、書き込み保護を無効化します。
- ケース 2：書き込み保護解除後に読み出し保護はアクティブ状態を維持。ユーザブートローダによるアプリケーション内プログラミングで有効です。
 - － フラッシュメモリ制御レジスタ（FLASH_CR）の OPTER ビットを使用して、オプションバイト領域全体を消去します。
 - － フラッシュ制御レジスタ（FLASH_CR）の OBL_LAUNCH ビットをセットしてオプションバイト（および新しい WRP[1:0] バイト）を再ロードし、書き込み保護を無効化します。

3.3.3 オプションバイトの書き込み保護

オプションバイトは、デフォルトで常に読み出しアクセス可能であり、また書き込み保護されています。オプションバイトへの書き込みアクセス（プログラム／消去）には、OPTKEYR にキーのシーケンス（ロックと同じ）を書き込む必要があります。正しいキーのシーケンスによってオプションバイトへの書き込みアクセスができ、これはFLASH_CR レジスタの OPTWREのセットによって示されます。書き込みアクセスは、ソフトウェアを通じてビットをリセットすることで無効にできます。

3.4 フラッシュ割り込み

表 8. フラッシュ割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
操作終了	EOP	EOPIE
書き込み保護エラー	WRPRTERR	ERRIE
プログラミングエラー	PGERR	ERRIE

3.5 フラッシュレジスタの説明

フラッシュメモリレジスタは、32 ビットワードでアクセスする必要があります（ハーフワードおよびバイトアクセスは許可されません）。

3.5.1 フラッシュアクセス制御レジスタ（FLASH_ACR）

アドレスオフセット：0x00

リセット値：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRFTBS	PRFTBE	Res.	LATENCY[2:0]		
										r	rw		rw	rw	rw

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **PRFTBS**：プリフェッチバッファステータス

このビットは、プリフェッチバッファのステータスを提供します。

0：プリフェッチバッファは無効です。

1：プリフェッチバッファは有効です。

注： プリフェッチステータスは、最初のフェッチリクエストが完了すると直ちに 1 にセットされます。

ビット 4 **PRFTBE**：プリフェッチバッファ有効化

0：プリフェッチは無効です。

1：プリフェッチは有効です。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **LATENCY[2:0]**：遅延

これらのビットは、フラッシュアクセス時間に対する SYSCLK（システムクロック）周期の割合を表します。

000：SYSCLK ≤ 24 MHz の場合、0 ウェイトステート

001：24 MHz < SYSCLK ≤ 48 MHz の場合、1 ウェイトステート

3.5.2 フラッシュキーレジスタ (FLASH_KEYR)

アドレスオフセット : 0x04
リセット値 : 0XXXXX XXXX

これらのすべてのレジスタビットは書き込み専用で、読み出すと 0 が返されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **FKEY** : フラッシュキー
これらのビットは、フラッシュをロック解除するキーを表します。

3.5.3 フラッシュオプションキーレジスタ (FLASH_OPTKEYR)

アドレスオフセット : 0x08
リセット値 : 0XXXXX XXXX

これらのすべてのレジスタビットは書き込み専用で、読み出すと 0 が返されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **OPTKEY** : オプションバイトキー
これらのビットは、OPTWRE をロック解除するキーを表します。

3.5.4 フラッシュステータスレジスタ (FLASH_SR)

アドレスオフセット : 0x0C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOP	WRPRT ERR	Res.	PG ERR	Res.	BSY
										rw	rw		rw		r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **EOP** : 操作終了

フラッシュ操作（プログラム／消去）の完了時に、ハードウェアによってセットされます。
1 を書き込むとリセットされます。

注： 成功したプログラムや消去操作が終了するたびに、EOP がアサートされます。

ビット 4 **WRPRTERR** : 書き込み保護エラー

フラッシュメモリの書き込み保護されたアドレスをプログラムすると、ハードウェアによってセットされます。
1 を書き込むとリセットされます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **PGERR** : プログラミングエラー

プログラムする前に、プログラム対象のアドレスに「0xFFFF」以外の値が含まれる場合、ハードウェアによってセットされます。
1 を書き込むとリセットされます。

注： FLASH_CR レジスタの STRT ビットはプログラム操作開始前にリセットする必要があります。

ビット 1 予約済みであり、リセット値のままにしておかなければなりません。

ビット 0 **BSY** : ビジー

これは、フラッシュ操作が進行中であることを示します。フラッシュ操作の開始時にセットされ、操作が終了するかエラーが発生するとリセットされます。

3.5.5 フラッシュ制御レジスタ (FLASH_CR)

アドレスオフセット : 0x10
リセット値 : 0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OBL_LAUNCH	EOPIE	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
		rw	rw		rw	rw		rw	rw	rw	rw		rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **OBL_LAUNCH** : オプションバイトローディングの強制

1 にセットすると、このビットではオプションバイトの再ロードを強制的に行います。この操作によって、システムリセットが生成されます。

0 : インアクティブ

1 : アクティブ

ビット 12 **EOPIE** : 操作終了割込み有効化

このビットは、FLASH_SR レジスタの EOP ビットが 1 になると割込み生成を有効にします。

0 : 割込み生成は無効です。

1 : 割込み生成は有効です。

ビット 11 予約済みであり、リセット値のままにしておかなければなりません。

ビット 10 **ERRIE** : エラー割り込みイネーブル

このビットは、FLASH_SR レジスタに PGERR/WRPRTERR がセットされた場合に、エラーによる割込み生成を有効にします。

0 : 割込み生成は無効です。

1 : 割込み生成は有効です。

ビット 9 **OPTWRE** : オプションバイトの書き込み有効化

セットすると、オプションバイトをプログラムできます。このビットは、FLASH_OPTKEYR レジスタに正しいキーシーケンスを書き込む際にセットされます。

このビットは、ソフトウェアでリセットできます。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **LOCK** : ロック状態

1 のみを書き込みます。このビットがセットされている場合は、フラッシュがロックされていることを示します。このビットは、ロック解除シーケンスが検出されると、ハードウェアによってリセットされます。ロック解除操作が成功しない場合には、このビットは次のリセットまでセットされたままとなります。

ビット 6 **STRT** : 開始

このビットがセットされると、ERASE 操作がトリガされます。このビットはソフトウェアによってのみセットされ、BSY ビットがリセットされるとリセットされます。

ビット 5 **OPTER** : オプションバイトの消去

オプションバイトの消去が選択されます。

ビット 4 **OPTPG** : オプションバイトのプログラミング

オプションバイトのプログラミングが選択されます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **MER** : 全体消去

選択したすべてのユーザページが消去されます。

ビット 1 **PER** : ページ消去

選択したページが消去されます。

ビット 0 **PG** : プログラミング

フラッシュプログラミングが選択されます。

3.5.6 フラッシュアドレスレジスタ (FLASH_AR)

アドレスオフセット : 0x14
リセット値 : 0x0000 0000

このレジスタは、現在使用しているアドレスと最後に使用したアドレスで、ハードウェアによって更新されます。ページ消去操作では、選択したページを示すためにソフトウェアで更新する必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAR[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:0 **FAR** : フラッシュアドレス

プログラミングを選択した場合はプログラムするアドレスを選択し、ページの消去を選択した場合は消去するページを選択します。

注 : *FLASH_SR レジスタのBSY ビットがセットされたときに、このレジスタへの書き込みアクセスがブロックされます。*

3.5.7 フラッシュオプションバイトレジスタ (FLASH_OBR)

アドレスオフセット : 0x1C
リセット値 : 0xFFFF XX0X

レジスタのリセット値は、オプションバイトでプログラムした値に依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA1								DATA0							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_SEL	RAM_PARITY_CHECK	VDDA_MONITOR	nBOOT1	BOOT0	nRST_STDBY	nRST_STOP	WDG_SW	Res.	Res.	Res.	Res.	Res.	RDPRT[1:0]		OPTERR
r	r	r	r	r	r	r	r						r	r	r

ビット 31:24 **DATA1**

ビット 23:16 **DATA0**

ビット 15:8 ユーザオプションバイト :

ビット 15 : **BOOT_SEL** (STM32F04x および STM32F09x デバイスでのみ使用可能)

ビット 14 : **RAM_PARITY_CHECK**

ビット 13 : **VDDA_MONITOR**

ビット 12 : **nBOOT1**

ビット 11 : **nBOOT0** (STM32F04x および STM32F09x デバイスでのみ使用可能)

ビット 10 : **nRST_STDBY**

ビット 9 : **nRST_STOP**

ビット 8 : **WDG_SW**

ビット 7:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:1 **RDPRT[1:0]** : 読み出し保護レベルステータス

- 00 : 読み出し保護レベル 0 は有効です (ST 出荷時の設定)。
- 01 : 読み出し保護レベル 1 は有効です。
- 11 : 読み出し保護レベル 2 は有効です。

ビット 0 **OPTERR** : オプションバイトエラー

セットされると、ロードされたオプションバイトとその補数が一致しないことが示されます。対応するバイトとその補数は、FLASH_OBR または FLASH_WRPR レジスタで 0xFF として読み出されます。

3.5.8 書き込み保護レジスタ (FLASH_WRPR)

アドレスオフセット : 0x20

リセット値 : 0xFFFF XXXX

レジスタのリセット値は、オプションバイトでプログラムした値に依存します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WRP[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRP[15:0]															

ビット 31:0 **WRP** : 書き込み保護

このレジスタには、OBL によってロードされた書き込み保護オプションバイトが含まれます。

3.5.9 フラッシュレジスタマップ

表 9. フラッシュインタフェース - レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRFTBS	PRFTBE	HLFCYA	LATENCY [2:0]			
	リセット値																											0	0	0	0	0	0	
0x004	FLASH_KEYR	FKEY[31:0]																																
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x008	FLASH_OPTKEYR	OPTKEY[31:0]																																
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
0x00C	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOP	WRPRTERR	Res.	PGERR	ERLYBSY		
	リセット値																											0	0		0	0	0	
0x010	FLASH_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OBL_LAUNCH	EOPIE	Res.	Res.	ERRIE	OPTWRE	Res.	LOCK	STRT	OPTER	OPTPG	Res.	MER	PER	PG
	リセット値																			0	0		0	0		1	0	0	0		0	0	0	
0x014	FLASH_AR	FAR[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	FLASH_OBR	Data1								Data0								BOOT_SEL	RAM_PARITY_CHECK	VDDA_MONITOR	nBOOT1	nBOOT0	nRST_STDBY	nRST_STOP	WDG_SW	Res.	Res.	Res.	Res.	Res.	Res.	RDPRT[1:0]		
	リセット値																	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X
0x020	FLASH_WRP	WRP[31:0]																																
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

4 オプションバイト

最大 8 つのオプションバイトがあります。これらは、アプリケーション要件によってエンドユーザが設定します。設定の例として、ウォッチドッグをハードウェアまたはソフトウェアモードに選択できます。

32 ビットワードは、オプションバイトで次に示すように分割されます。

表 10. オプションバイトのフォーマット

31-24	23-16	15:-8	7-0
オプションバイト 1 の補数	オプションバイト 1	オプションバイト 0 の補数	オプションバイト 0

情報ブロック内のこれらのバイトの構成は、表 11 に示すとおりです。

オプションバイトは、表 11 にリストされたメモリ位置またはオプションバイトレジスタ (FLASH_OBR) から読み出すことができます。

注： *新しくプログラムされたオプションバイト（ユーザ、読み出し／書き込み保護）は、システムリセット後にロードされます。*

表 11. オプションバイトの構成

アドレス	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFF F800	nUSER	USER	nRDP	RDP
0x1FFF F804	nData1	Data1	nData0	Data0
0x1FFF F808	nWRP1	WRP1	nWRP0	WRP0
0x1FFF F80C	nWRP3	WRP3	nWRP2	WRP2

システムリセットごとに、オプションバイトローダ（OBL）は情報ブロックを読み出して、オプションバイトレジスタ（FLASH_OBR）と書き込み保護レジスタ（FLASH_WRP1）にデータを格納します。各オプションバイトにも情報ブロックに補数があります。オプションのローディング中に、オプションビットとその補数を確認することで、ローディングが正しく実行されたことを確認できます。そうでない場合は、オプションバイトエラー（OPTERR）が生成されます。比較エラーが発生した場合は、対応するオプションバイトは強制的に 0xFF になります。オプションバイトとその補数が両方とも 0xFF（電氣的消去状態）と等しい場合に、比較回路は無効化されます。

4.1 オプションバイトの説明

4.1.1 ユーザオプションバイトおよび読み出し保護オプションバイト

フラッシュメモリアドレス : 0x1FFF F800
ST 出荷時の値 : 0x00FF 55AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nUSER								USER							
								BOOT_SEL	RAM_PARITY_CHECK	VDDA_MONITOR	nBOOT1	nBOOT0	nRST_STDBY	nRST_STOP	WDG_SW
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nRDP								RDP							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 nUSER : ユーザオプションバイトの補数



ビット 23:16 **USER** : ユーザオプションバイト (FLASH_OBR[15:8] に格納)

ビット 23 : **BOOT_SEL**

0 : BOOT0 信号は nBOOT0 オプションビットで定義されます。

1 : BOOT0 信号は BOOT0 ピン値で定義されます (レガシーモード)。

STM32F04x および STM32F09x デバイスでのみ使用できます。他のデバイスでは「1」とみなされます。

ビット 22 : **RAM_PARITY_CHECK**

0 : RAM パリティチェックは有効です。

1 : RAM パリティチェックは無効です。

ビット 21 : **VDDA_MONITOR**

0 : V_{DDA} 電源供給スーパバイザは無効です。

1 : V_{DDA} 電源供給スーパバイザは有効です。

ビット 20 : **nBOOT1**

BOOT0 信号とともに、デバイスブートモードを選択します。詳細については、[セクション 2.5 : ブート設定](#)を参照してください。

ビット 19 : **nBOOT0**

BOOT_SEL がクリアされると、nBOOT0 ビットではデバイスブートモードの選択に使用する BOOT0 信号値を定義します。詳細については、[セクション 2.5 : ブート設定](#)を参照してください。

STM32F04x および STM32F09x デバイスでのみ使用できます。

ビット 18 : **nRST_STDBY**

0 : STANDBY モードに移行するときにリセットを生成します。

1 : リセットは生成されません。

ビット 17 : **nRST_STOP**

0 : STOP モードに移行するときにリセットを生成します。

1 : リセットは生成されません。

ビット 16 : **WDG_SW**

0 : ハードウェアウォッチドッグ

1 : ソフトウェアウォッチドッグ

ビット 15:8 **nRDP** : 読み出し保護オプションバイトの補数

ビット 7:0 **RDP** : 読み出し保護オプションバイト

このバイトの値はフラッシュメモリの保護レベルを定義します。

0xAA : レベル 0 (ST 出荷時の設定)

0xFF (0xAA & 0xCC を除く) : レベル 1

0xCC : レベル 2

注 : **読み出し保護レベルステータスはフラッシュオプションバイトレジスタ (FLASH_OBR) の RDPRT[1:0] ビットに格納されます。読み出し保護の詳細については、セクション 3.3.1 : 読み出し保護を参照してください。**

4.1.2 ユーザデータオプションバイト

フラッシュメモリアドレス : 0x1FFF F804
ST 出荷時の値 : 0x00FF 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nData1								Data1							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nData0								Data0							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **nData1** : ユーザデータバイト 1 の補数
ビット 23:16 **Data1** : ユーザデータバイト 1 の値 (FLASH_OBR[31:24] に格納)
ビット 15:8 **nData0** : ユーザデータバイト 0 の補数
ビット 7:0 **Data0** : ユーザデータバイト 0 の値 (FLASH_OBR[23:16] に格納)

4.1.3 書き込み保護オプションバイト

このレジスタのセットは、フラッシュメモリの書き込み保護に使用します。WRPx フィールドでビットをクリア (および同時に nWRPx フィールドの対応するビットをセット) すると、特定のメモリセクタを書き込み保護できます。

STM32F03x、STM32F04x、STM32F05x、および STM32F07x デバイスでは、フラッシュメモリを 0 から 31 の WRP ビットによって 4 KB のセクタ単位で保護します。

STM32F09x デバイスでは、0 から 30 の WRP ビットによって最初の 124 KB を 4 KB のセクタ単位で保護し、ビット 31 で最後の 132 KB を保護します。

詳細については、[セクション 3.3.2 : 書き込み保護](#) を参照してください。

フラッシュメモリアドレス : 0x1FFF F808
ST 出荷時の値 : 0x00FF 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nWRP1								WRP1							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nWRP0								WRP0							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 **nWRP1** : フラッシュメモリ書き込み保護オプションバイト 1 の補数
ビット 23:16 **WRP1** : フラッシュメモリ書き込み保護オプションバイト 1 の値 (FLASH_WRP1[15:8] に格納)
ビット 15:8 **nWRP0** : フラッシュメモリ書き込み保護オプションバイト 0 の補数
ビット 7:0 **WRP0** : フラッシュメモリ書き込み保護オプションバイト 0 の値 (FLASH_WRP0[7:0] に格納)

注 : **STM32F03x および STM32F04x デバイスでは WRP0 と nWRP0 のみを内蔵します。**

次のオプションバイトは **STM32F07x および STM32F09x デバイスでのみ使用可能です。**

フラッシュメモリアドレス : 0x1FFF F80C
ST 出荷時の値 : 0x00FF 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
nWRP3								WRP3							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nWRP2								WRP2							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 31:24 **nWRP3** : フラッシュメモリ書き込み保護オプションバイト 1 の補数

ビット 23:16 **WRP3** : フラッシュメモリ書き込み保護オプションバイト 1 の値 (FLASH_WRP[15:8] に格納)

ビット 15:8 **nWRP2** : フラッシュメモリ書き込み保護オプションバイト 0 の補数

ビット 7:0 **WRP2** : フラッシュメモリ書き込み保護オプションバイト 0 の値 (FLASH_WRP[7:0] に格納)

4.1.4 オプションバイトマップ

次の表にオプションバイトの一覧を示します。

表 12. オプションバイトマップおよび ST 出荷時の値

オフ セット	オプション バイト	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	ユーザ保護 および 読み出し保護	nUSER								USER							nRDP								RDP								
										BOOT_SEL	RAM_PARITY_CHECK	VDDA_MONITOR	nBOOT1	BOOT0	nRST_STDBY	nRST_STOP																	WDG_SW
	ST 出荷時の値	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	1
0x04	ユーザデータ	nData1								Data1							nData0								Data0								
	ST 出荷時の値	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0x08	書き込み保護 機能	nWRP1								WRP1							nWRP0								WRP0								
	ST 出荷時の値	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0x0C	書き込み保護 機能	nWRP3								WRP3							nWRP2								WRP2								
	ST 出荷時の値	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

5 巡回冗長検査計算ユニット (CRC)

プログラム可能な多項式機能は、STM32F07x と STM32F09x デバイスにのみ適用されます。

5.1 概要

CRC (Cyclic Redundancy Check : 巡回冗長検査) 計算ユニットは、8、16、または 32 ビットデータワードと、ある生成多項式から、CRC コードを得るために使用されます。

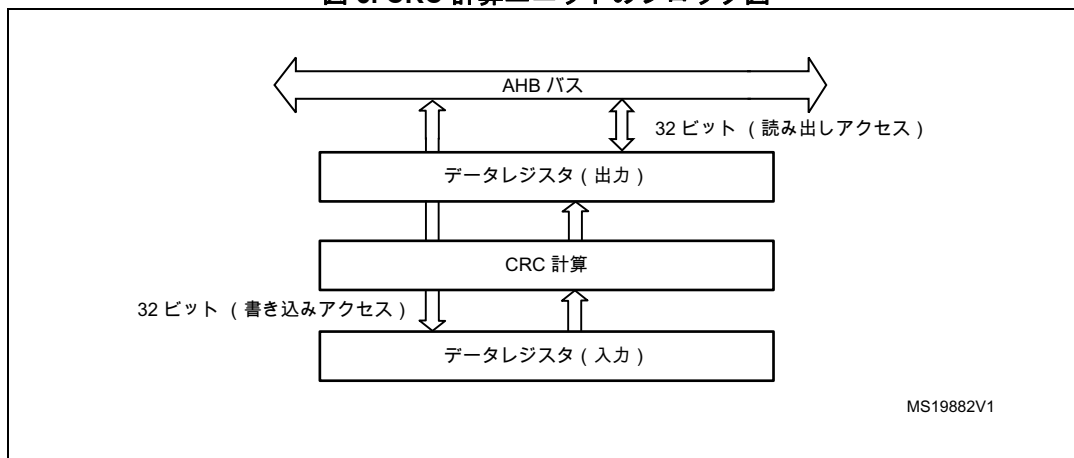
他のアプリケーションの中でも、CRC ベースのテクニックは、データ転送やストレージの整合性を確認するために使用されます。機能安全規格の範囲内では、このテクニックがフラッシュメモリの整合性を確認するひとつの手段となっています。CRC 計算ユニットは、実行時にソフトウェアのシグネチャ計算を支援します。リンク時に生成されて、特定のメモリ領域に保存されたりファレンスシグネチャと計算されたソフトウェアシグネチャが比較されます。

5.2 CRC の主な機能

- プログラム可能なサイズ (7、8、16、32 ビット) の完全にプログラム可能な多項式。
- サイズが 8、16、32 ビットのデータを取り扱います。
- プログラム可能な CRC の初期値
- シングル入力/出力 32 ビットデータレジスタ
- 計算時のバスのストールを避けるための入力バッファ
- データサイズが 32 ビットの場合、CRC の計算は AHB クロック 4 サイクル (HCLK) 以内に行われます。
- 汎用 8 ビットレジスタ (一時ストレージとして使用可能)
- I/O データの可逆性のオプション

5.3 CRC の機能説明

図 6. CRC 計算ユニットのブロック図



CRC 計算ユニットは、1 つの 32 ビット読み出し／書き込みデータレジスタ (CRC_DR) を持っています。このレジスタを使用して、新しいデータを入力し (書き込みアクセス)、前回の CRC 計算結果を保持します (読み出しアクセス)。

データレジスタへの書き込み操作のたびに、前回の CRC 値 (CRC_DR に格納) と新しい値の組み合わせが作成されます。CRC 計算は、書き込まれるデータのフォーマットに応じ、32 ビットデータワード全体に対して、またはバイト単位で行われます。

CRC_DR レジスタは、ワード、右詰め、ハーフワード、右詰め、バイトによってアクセスできます。他のレジスタについては、32 ビットアクセスのみ可能です。

計算の時間はデータ幅に依存します。

- 32 ビットの場合、4 AHB クロックサイクル
- 16 ビットの場合、2 AHB クロックサイクル
- 8 ビットの場合、1 AHB クロックサイクル

入力バッファを使うと、前回の CRC 計算によるウェイトステートを待つことなく、すぐに第 2 のデータを書き込むことができます。

与えられたバイト数に対する書き込みアクセス数を最小限に抑えるために、データサイズを動的に調節することができます。たとえば、5 バイトの CRC は、1 ワードの書き込みと、それに続く 1 バイトの書き込みで計算することができます。

入力データを逆にして、さまざまなエンディアンネス方式を管理することができます。逆転操作は、CRC_CR レジスタの REV_IN[1:0] ビットに応じて、8、16、および 32 ビットで行うことができます。

たとえば、入力データ 0x1A2B3C4D は、CRC 計算では以下のように使用されます。

- 0x58D43CB2 (バイト単位でビットが逆転)
- 0xD458B23C (ハーフワード単位でビットが逆転)
- 0xB23CD458 (フルワードでビットが逆転)

また、出力データも、CRC_CR レジスタの REV_OUT ビットをセットすることによって逆にすることができます。

操作はビットレベルで行われます。たとえば、出力データ 0x11223344 は 0x22CC4488 に変換されます。

CRC 計算機は、CRC_CR レジスタの RESET 制御ビットを使用して、プログラム可能な値に初期化することができます（デフォルト値は 0xFFFFFFFF）。

CRC の初期値は、CRC_INIT レジスタでプログラムすることができます。CRC_DR レジスタは、CRC_INIT レジスタの書き込みアクセス時に自動的に初期化されます。

CRC_IDR レジスタを使用して、CRC 計算に関する一時的な値を保持することができます。このレジスタは、CRC_CR レジスタの RESET ビットによる影響を受けません。

多項式のプログラミング可能性

多項式係数は CRC_POL レジスタを介して完全にプログラム可能であり、多項式のサイズは CRC_CR レジスタの POLYSIZE[1:0] ビットをプログラムすることにより、7、8、16、または 32 ビットに設定できます。偶多項式はサポートされていません。

CRC データが 32 ビット未満であれば、その値は CRC_DR レジスタの下位ビットから読み出すことができます。

信頼できる CRC 計算結果を得るために、CRC 計算の実行中に多項式の値やサイズを変更することはできません。そのため、CRC 計算が進行中である場合、アプリケーションは計算をリセットするか、または CRC_DR の読み出しを行ってから、多項式を変更しなければなりません。

多項式のデフォルト値は、CRC-32（イーサネット）多項式、0x4C11DB7。

5.4 CRC レジスタ

5.4.1 データレジスタ（CRC_DR）

アドレスオフセット：0x00

リセット値：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw															

ビット 31:0 **DR[31:0]**：データレジスタビット

このレジスタを使用して、CRC 計算機に新しいデータを書き込みます。

読み出し時には、前回の CRC 計算結果を保持します。

データサイズが 32 ビット未満であれば、下位ビットを使用して正しい値の書き込み／読み出しを行います。

5.4.2 独立型データレジスタ (CRC_IDR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]							
								rw							

ビット 31:8 予約済み、クリア状態を保つ必要があります。

ビット 7:0 **IDR[7:0]** : 汎用 8 ビットデータレジスタビット

これらのビットは、1 バイトの一時的なストレージ領域として使用できます。

このレジスタは、CRC_CR レジスタの RESET ビットによって生成される CRC リセットの影響を受けません。

5.4.3 制御レジスタ (CRC_CR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	リセット
								rw	rw	rw	rw	rw			rs

ビット 31:8 予約済み、クリア状態を保つ必要があります。

ビット 7 **REV_OUT** : 出力データを逆にします。

このビットは、出力データビット順序の反転を制御します。

0 : ビットの順序は変わりません。

1 : 出力フォーマットのビットが反転しています。

ビット 6:5 **REV_IN[1:0]** : 入力データを逆にします。

これらのビットは、入力データビット順序の反転を制御します。

00 : ビットの順序は変わりません。

01 : バイト単位でビットが反転しています。

10 : ハーフワード単位でビットが反転しています。

11 : ワード単位でビットが反転しています。



ビット 4:3 **POLY_SIZE[1:0]** : 多項式のサイズ
これらのビットは、多項式のサイズを制御します。
00 : 32 ビットの多項式
01 : 16 ビットの多項式
10 : 8 ビットの多項式
11 : 7 ビットの多項式

ビット 2:1 予約済み、クリア状態を保つ必要があります。

ビット 0 **RESET** : RESET ビット
このビットは、CRC 計算ユニットをリセットし、CRC_INIT レジスタに格納された値にデータレジスタをセットするために、ソフトウェアによってセットされます。このビットはセットのみが可能で、ハードウェアによって自動的にクリアされます。

5.4.4 CRC の初期値 (CRC_INIT)

アドレスオフセット : 0x10
リセット値 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC_INIT[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INIT[15:0]															
rw															

ビット 31:0 **CRC_INIT** : プログラム可能な CRC の初期値
このレジスタを使用して、CRC の初期値を書き込みます。

5.4.5 CRC 多項式 (CRC_POL)

アドレスオフセット : 0x14
リセット値 : 0x04C11DB7

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POL[15:0]															
rw															

ビット 31:0 **POL[31:0]** : プログラム可能な多項式
このレジスタを使用して、CRC 計算に使用される多項式の係数を書き込みます。
多項式のサイズが 32 ビット未満であれば、下位ビットを使用して正しい値をプログラムする必要があります。

5.4.6 CRC レジスタマップ

表 13. CRC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRC_DR	DR[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x04	CRC_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]								
	リセット値																										0	0	0	0	0	0	0
0x08	CRC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	リセット
	リセット値																									0	0	0	0	0			0
0x10	CRC_INIT	CRC_INIT[31:0]																															
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	CRC_POL	多項式係数																															
	リセット値	0x04C11DB7																															

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

6 電源制御 (PWR)

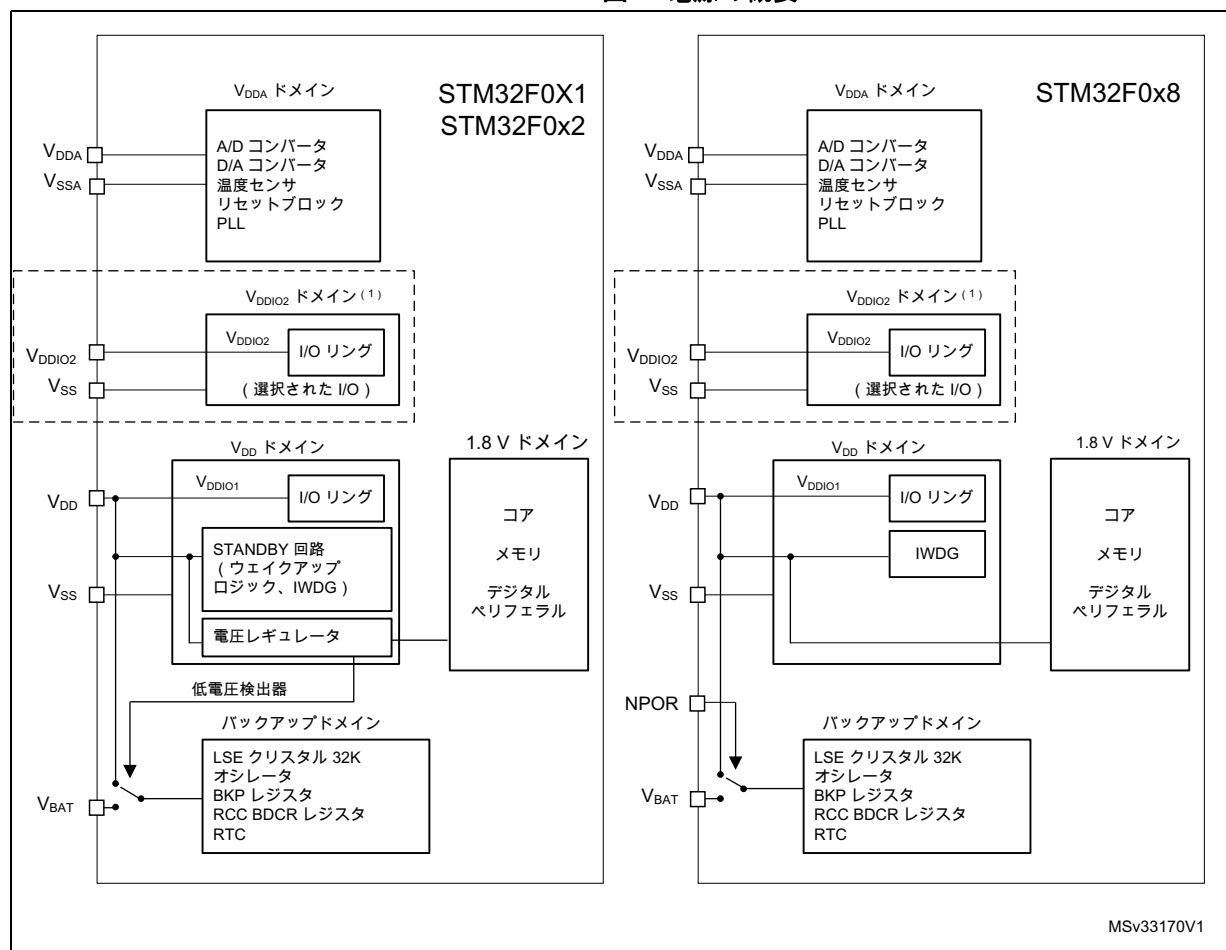
6.1 電源

アプリケーションで 1.8 V を安定して提供する必要がある STM32F0x8 サブファミリとは異なり、STM32F0x1/F0x2 サブファミリには内部 1.8 V デジタル電源ドメインを供給するために電圧レギュレータが内蔵されています。

- STM32F0x1/F0x2 デバイスでは、2.0 ~ 3.6 V 動作電源電圧 (V_{DD}) と 2.0 ~ 3.6 V アナログ電源電圧 (V_{DDA}) が必要です。
- STM32F0x8 デバイスでは、1.8 V $\pm 8\%$ の動作電源電圧 (V_{DD}) と 1.65 V ~ 3.6 V アナログ電源電圧 (V_{DDA}) が必要です。

リアルタイムクロック (RTC) およびバックアップレジスタは、主電源 V_{DD} がオフの場合、 V_{BAT} から供給することができます。

図 7. 電源の概要



1. STM32F04x、STM32F07x、および STM32F09x デバイスでのみ使用可能。

6.1.1 独立した A/D および D/A コンバータ用電源と基準電圧

変換の精度を向上させ、供給の柔軟性を拡張させるため、ADC および DAC には独立した電源が供給されます。この電源は、PCB のノイズを防ぐために、個別にフィルタシールドすることができます。

- ADC および DAC 用の電源入力は、独立した V_{DDA} ピンから供給されます。
- ADC 電源のグラウンドは、 V_{SSA} ピンとして用意されています。

V_{DDA} 電源／基準電圧は V_{DD} 以上である必要があります。

単電源を使用している場合、ノイズフリーの V_{DDA} 基準電圧を確保するために、外部フィルタ回路を通じて V_{DDA} を V_{DD} に外部的に接続することができます。

V_{DDA} が V_{DD} と異なる場合、 V_{DDA} は V_{DD} 以上である必要があります。パワーアップ／パワーダウン時に V_{DDA} と V_{DD} の間で安全な電位差を維持するために、 V_{DD} と V_{DDA} の間で外部ショットキーダイオードを使用できます。許容可能な差の最大値に関してはデータシートを参照してください。

6.1.2 独立 I/O 電源レール

STM32F04x、STM32F07x、および STM32F09x デバイスでより高い供給の柔軟性を実現するために、入出力の一部は独立した電源レールから供給されます。このレールの電源は 1.65 ~ 3.6 V の範囲で供給され、 V_{DDIO2} ピンを通じて外部から提供されます。 V_{DDIO2} 電圧レベルは V_{DD} または V_{DDA} から完全に独立していますが、 V_{DD} ピン上の有効な動作電源なしに供給することはできません。関係する I/O リストについては、関連するデバイスデータシートのピン配置図または表を参照してください。

V_{DDIO2} 電源は、内部基準電圧 (V_{REFINT}) により監視され、比較されます。 V_{DDIO2} がこの閾値を下回る場合、このレールから供給されるすべての I/O はハードウェアによって無効にされます。このコンパレータの出力は、EXTI ライン 31 に接続され、これは割り込みの生成に使用できます。

6.1.3 バッテリバックアップドメイン

V_{DD} がオフになった場合に、バックアップレジスタの内容を保持し、RTC ドメイン全体への電源供給を維持するために、 V_{BAT} ピンをバッテリーやその他の電源から供給されるオプションの予備電源に接続することができます。

V_{BAT} ピンから RTC ユニット、LSE オシレータ、および PC13 から PC15 の I/O に電源が供給され、主電源がオフの場合でも RTC が動作できるようにします。 V_{BAT} 電源への切り替えは、リセットブロックに組み込まれているパワーダウンリセット回路または (STM32F0x8 デバイス上で) 外部 NPOR 信号によって制御されます。

警告： $t_{RSTTEMPO}$ (V_{DD} 起動時の過渡期間) 中や PDR の検出後、 V_{BAT} と V_{DD} の間の電源スイッチは V_{BAT} に接続されたままになります。
起動フェーズ中、 V_{DD} が $t_{RSTTEMPO}$ 以内に規定値に達し ($t_{RSTTEMPO}$ の値については、データシートを参照)、かつ $V_{DD} > V_{BAT} + 0.6$ V である場合、電流は V_{BAT} まで V_{DD} と電源スイッチ (V_{BAT}) の間に接続された内部ダイオードを通して注入されます。
 V_{BAT} ピンに接続された電源/バッテリーがこの電流注入に対応できない場合は、この電源と V_{BAT} ピンの間に外部低電圧降下ダイオードを接続することを強く推奨します。

外部バッテリーを使用しないアプリケーションでは、 V_{BAT} を V_{DD} に外部接続し、デカップリング用の 100 nF のセラミックコンデンサを接続することを推奨します (詳細については、AN4080 を参照)。

RTC ドメインが V_{DD} から供給を受けている場合（アナログスイッチが V_{DD} に接続された状態）、次の機能が使用できます。

- PC13、PC14、および PC15 は、GPIO ピンとして使用可能
- PC13、PC14、および PC15 は、RTC または LSE で設定可能（[セクション 24.4 : RTC の機能説明 \(569 ページ\)](#) を参照）

注： アナログスイッチは限られた電流（3 mA）しか転送できないため、出力モードでの GPIO の PC13 から PC15 までの使用には制限があります。最大負荷 30 pF で最大速度 2 MHz に制限する必要があります。これらの I/O を電流ソースとして使用することはできません（たとえば、LED を駆動するなど）。

RTC ドメインが V_{BAT} から電源供給を受けている場合（ V_{DD} が印加されないため、アナログスイッチが V_{BAT} に接続された状態）、次の機能が使用できます。

- PC13、PC14、および PC15 は、RTC または LSE でのみ制御可能（[セクション 24.4 : RTC の機能説明 \(569 ページ\)](#) を参照）

6.1.4 電圧レギュレータ

リセット後、電圧レギュレータは常に使用可能になります。レギュレータは、アプリケーションのモードに応じて、3 種類のモードで動作します。

- RUN モードでは、レギュレータは 1.8 V ドメイン（コア、メモリ、デジタルペリフェラル）にフル電力を供給します。
- STOP モードでは、レジスタと SRAM の内容を保持したまま、レギュレータは 1.8 V ドメインに低電力を供給します。
- STANDBY モードでは、レギュレータの電源がオフになります。STANDBY 回路と RTC ドメインを除き、レジスタと SRAM の内容は失われます。

注： STM32F0x8 デバイスでは、電圧レギュレータはバイパスされ、マイクロコントローラは公称電源電圧 $V_{DD} = 1.8 \text{ V} \pm 8\%$ から電源供給されます。

6.2 電源供給スーパバイザ

6.2.1 パワーオンリセット (POR) / パワーダウンリセット (PDR)

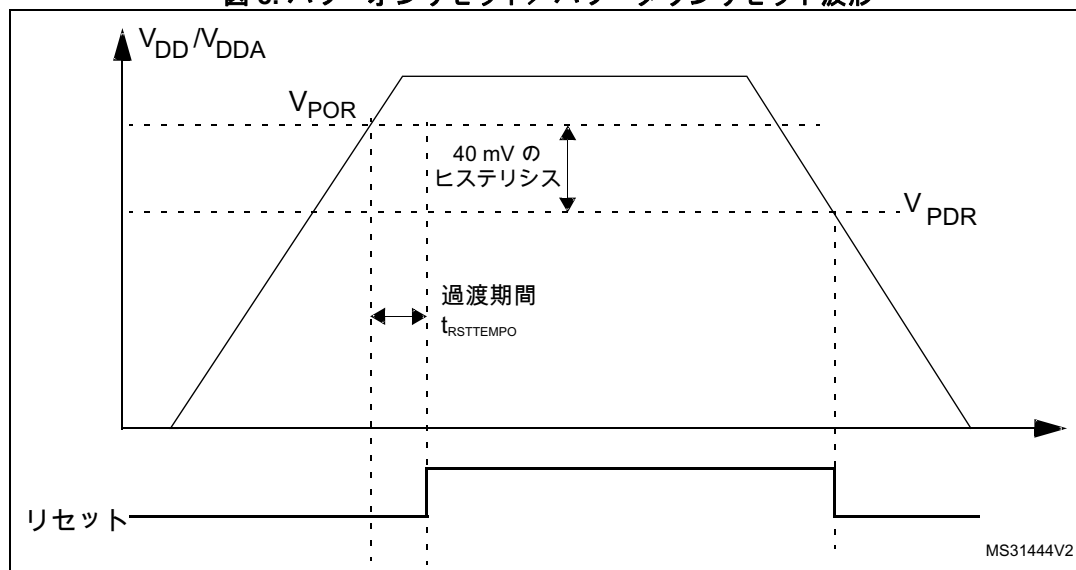
デバイスには、統合されたパワーオンリセット (POR) 回路とパワーダウンリセット (PDR) 回路があります。これらは常にアクティブで、2 V の閾値を上回る場合に正しい動作を保証します。

デバイスは、監視された電源電圧が規定の閾値 $V_{POR/PDR}$ を下回っている間は、外部のリセット回路を必要とせずに、リセットモードを維持します。

- POR では、 V_{DD} 電源電圧のみを監視します。起動フェーズ中、最初の入力は V_{DDA} である必要があり、この値は V_{DD} 以上である必要があります。
- PDR では、 V_{DD} および V_{DDA} 電源電圧を監視します。ただし、アプリケーションで V_{DDA} が V_{DD} 以上になるように設計されている場合、消費電力を低減するために（専用のオプションビット $V_{DDA\_MONITOR}$ をプログラムして） V_{DDA} 電源供給スーパバイザを無効にすることができます。

POR/PDR 閾値の詳細については、データシートの電気的特性セクションの項を参照してください。

図 8. パワーオンリセット/パワーダウンリセット波形



外部 NPOR 信号

STM32F0x8 デバイスでは、PB2 の I/O (小型のパッケージでは PB1) は使用できません。パワーオンリセットで使用する NPOR 機能に置き換えられます。

デバイスのパワーオンリセットとパワーダウンリセットを適切に行うために、 V_{DD} が安定するか電源をオフにするまで NPOR ピンをローにしておく必要があります。 V_{DD} が安定している場合、リセット状態は NPOR ピンをハイインピーダンスにすることで終了できます。NPOR ピンには、 V_{DDA} に接続された内部プルアップがあります。

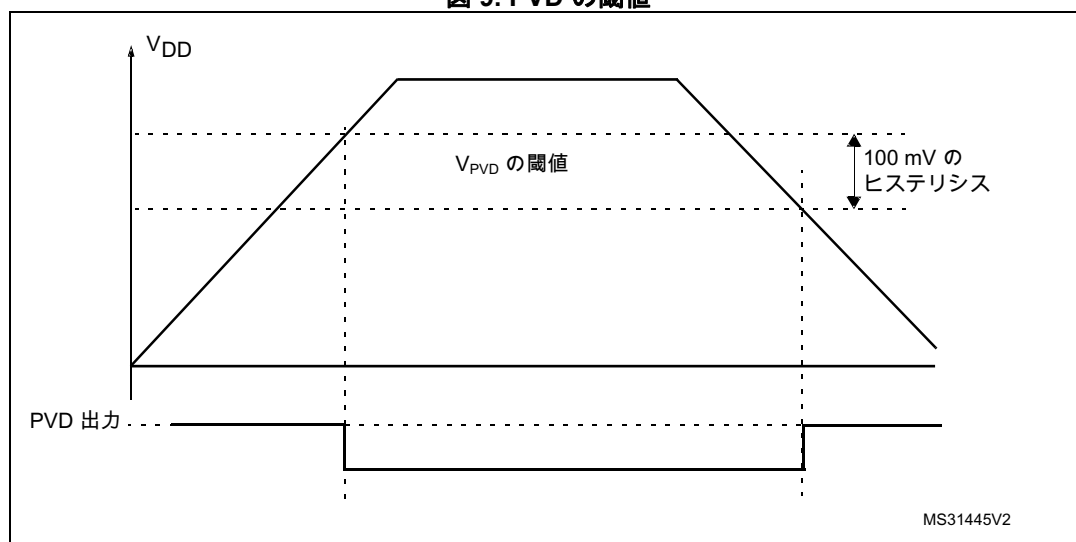
6.2.2 プログラム可能な電圧検出器 (PVD)

PVD を使用した V_{DD} 電源の監視が可能です。[電源制御レジスタ \(PWR_CR\)](#) の PLS[2:0] ビットで選択した閾値と比較することで監視します。

PVD は PVDE ビットをセットすることで有効になります。

[電源制御/ステータスレジスタ \(PWR_CSR\)](#) の PVDO フラグを使用して、 V_{DD} が PVD の閾値よりも高いか低いを示すことができます。このイベントは EXTI のライン 16 に内部接続され、EXTI レジスタで有効な場合は割り込みリクエストを生成させることができます。EXTI ライン 16 の立ち上がり/立ち下がりエッジの設定に応じて、 V_{DD} が PVD の閾値を下回るか、上回ったとき、あるいはその両方で、PVD 出力割り込みを生成させることができます。たとえば、サービスルーチンで、緊急停止処理を実行することなどが可能です。

図 9. PVD の閾値



6.3 低電力モード

デフォルトでは、マイクロコントローラは、システムリセットまたは電源リセット後は RUN モードです。外部イベント待ちなど、CPU の連続実行が不要なときの節電のために、いくつかの低電力モードが用意されています。消費電力の節減、スタートアップ時間の短縮、使用可能なウェイクアップソースを考慮した最適なモード選択はユーザに委ねられています。

このデバイスは、次の 3 つの低電力モードを備えています。

- SLEEP モード (CPU クロックはオフ。NVIC、SysTick などの Cortex®-M0 コアペリフェラルを含むすべてのペリフェラルは動作状態を維持)
- STOP モード (すべてのクロックが停止)
- STANDBY モード (1.8 V ドメインの電源オフ)

さらに、次の手段のいずれかによって RUN モードの消費電力を節減できます。

- システムクロックの低速化
- 使用しない APB および AHB ペリフェラルへのクロックのゲーティング

表 14. 低電力モードの概要

モード名	エントリ	ウェイクアップ	1.8V ドメイン クロックへの影響	V _{DD} ドメイン クロック への影響	電圧 レギュレータ
SLEEP (Sleep-now または Sleep-on-exit)	WFI	割り込み	CPU クロックオフ	なし	ON
	WFE	ウェイクアップイベント	他のクロックおよび アナログクロック ソースへの影響なし		
STOP	PDDS ビット および LPDS ビット + SLEEPDEEP ビット + WFI または WFE	任意の EXTI ライン (EXTI レジスタで 設定) 受信イベントでの 特定の通信 ペリフェラル (CEC、USART、 I2C)	すべての 1.8V ドメインクロック オフ	HSI および HSE オシレー タオフ	オンまたは低電 力モード (電源制御レジスタ (PWR_CR) に従う)
STANDBY	PDDS ビット + SLEEPDEEP ビット + WFI または WFE	WKUP ピンの 立ち上がりエッジ、 RTC アラーム、 NRST ピンによる 外部リセット IWDG リセット			オフ

注意： STM32F0x8 デバイスでは STOP モードを使用できます。ただし、レギュレータが使用されておらず、コアは外部ソースから直接供給されているため、低電力モードの電圧レギュレータと RUN モードの電圧レギュレータの識別は意味を持ちません。結果として、これらのデバイスでは STANDBY モードを使用できません。

6.3.1 システムクロックの低速化

RUN モードでは、プリスケアラレジスタをプログラミングすることによって、システムクロック (SYSCLK、HCLK、PCLK) の速度を下げることができます。SLEEP モードに移行する前にペリフェラルの速度を下げるため、これらのプリスケアラを使用することもできます。

詳細については、[セクション 7.4.2：クロック設定レジスタ \(RCC_CFGR\)](#) を参照してください。

6.3.2 ペリフェラルクロックゲーティング

RUN モードでは、消費電力を低減するため、個々のペリフェラルとメモリへの AHB クロック (HCLK) および APB クロック (PCLK) をいつでも停止することができます。

SLEEP モードで消費電力をさらに低減するため、WFI または WFE 命令を実行する前に、ペリフェラルクロックを停止することができます。

ペリフェラルクロックゲーティングは、[AHB ペリフェラルクロック有効レジスタ \(RCC_AHBENR\)](#)、[APB ペリフェラルクロック有効レジスタ 2 \(RCC_APB2ENR\)](#)、および [APB ペリフェラルクロック有効レジスタ 1 \(RCC_APB1ENR\)](#) で制御されます。

6.3.3 SLEEP モード

SLEEP モードへの移行

WFI (Wait For Interrupt) 命令または WFE (Wait for Event) 命令を実行することにより、SLEEP モードに移行します。SLEEP モードへの移行方法として、Cortex[®]-M0 システム制御レジスタの SLEEPONEXIT ビットによって、2 つのオプションが選択できます。

- Sleep-now : SLEEPONEXIT ビットがクリアされている場合、MCU は、WFI または WFE 命令の実行直後に SLEEP モードに移行します。
- Sleep-on-exit : SLEEPONEXIT ビットがセットされている場合、MCU は、最も優先度の低い ISR の終了直後に SLEEP モードに移行します。

SLEEP モードでは、すべての I/O ピンは RUN モードと同じ状態を保持します。

SLEEP モードへの移行方法の詳細については、[表 15](#) および [表 16](#) を参照してください。

SLEEP モードの終了

WFI 命令によって SLEEP モードに移行した場合、ペリフェラルの割り込みがネスト化されたベクタ割り込みコントローラ (NVIC) によって認識されると、デバイスは SLEEP モードを終了することができます。

WFE 命令によって SLEEP モードに移行した場合、MCU はイベントの発生直後に SLEEP モードを終了します。ウェイクアップイベントは、次のいずれかによって生成できます。

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、Cortex[®]-M0 システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードで設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャネルペンディングビットをクリアする必要はありません。

このモードは、割り込みの入口と出口で余分な時間を必要としないため、ウェイクアップ時間が最も短くなります。

SLEEP モードの終了方法の詳細については、[表 15](#) および [表 16](#) を参照してください。

表 15. Sleep-now

Sleep-now モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – SLEEPDEEP = 0 および – SLEEPONEXIT = 0 Cortex®-M0 システム制御レジスタを参照。
モードの終了	WFI 命令を使用して移行した場合 : 割り込み : 表 37 : ベクタテーブル を参照してください。 WFE 命令を使用して移行した場合 : ウェイクアップイベント : セクション 12.2.3 : イベント管理 を参照してください。
ウェイクアップ遅延時間	なし

表 16. Sleep-on-exit

Sleep-on-exit	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) – SLEEPDEEP = 0 および – SLEEPONEXIT = 1 Cortex®-M0 システム制御レジスタを参照。
モードの終了	割り込み : 表 37 : ベクタテーブル を参照してください。
ウェイクアップ遅延時間	なし

6.3.4 STOP モード

STOP モードは、ペリフェラルクロックゲーティングと組み合わせられた Cortex®-M0 のディープスリープ (deepsleep) モードに準拠しています。電圧レギュレータは、通常モードまたは低電力モードのいずれかに設定できます。STOP モードでは、1.8 V ドメインのすべてのクロックが停止し、PLL、HSI、HSE オシレータが停止します。SRAM とレジスタの内容は保持されます。

STOP モードでは、すべての I/O ピンは RUN モードと同じ状態を保持します。

STOP モードへの移行

STOP モードへの移行方法の詳細については、[表 17](#) を参照してください。

STOP モードで消費電力をさらに低減するには、内蔵電圧レギュレータを低電力モードにします。この設定は [電源制御レジスタ \(PWR_CR\)](#) の LPDS ビットで行います。

フラッシュメモリがプログラミング中の場合、メモリアクセスが終了してから、STOP モードに移行します。

APB ドメインにアクセス中の場合、APB アクセスが終了してから、STOP モードに移行します。

STOP モードでは、個別の制御ビットをプログラミングすることによって、次の機能を選択できます。

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。参照 : [セクション 22.3 : IWDG の機能説明](#)、[セクション 22 : 独立型ウォッチドッグ \(IWDG\)](#)

- リアルタイムクロック (RTC) : この設定は [RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI) : この設定は [制御/ステータスレジスタ \(RCC_CSR\)](#) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE) : この設定は [RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEON ビットで行います。

ADC や DAC は、STOP モードに移行する前に停止させない限り、STOP モード中にも電力を消費します。無効化する方法の詳細については、[ADC 制御レジスタ \(ADC_CR\)](#) および [DAC 制御レジスタ \(DAC_CR\)](#) を参照してください。

STOP モードの終了

STOP モードの終了方法の詳細については、[表 17](#) を参照してください。

割り込みまたはウェイクアップイベントの発行によって STOP モードを終了する場合、システムクロックとして HSI オシレータが選択されます。

電圧レギュレータが低電力モードで動作している場合、STOP モードからのウェイクアップの際、更なる起動時間が必要になります。STOP モードの間も内蔵レギュレータをオン状態に保つことによって、消費電力は増加しますが、起動時間は短縮されます。

表 17. STOP モード

STOP モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) - Cortex[®]-M0 システム制御レジスタの SLEEPDEEP ビットをセット - 電源制御レジスタ (PWR_CR) の PDDS ビットをクリア - PWR_CR の LPDS ビットを設定することによって電圧レギュレータのモードを選択 注 : STOP モードに移行するには、すべての EXTI ラインのペンディングビット (ペンディングレジスタ (EXTI_PR) 内)、すべてのペリフェラル割り込みのペンディングビット、および RTC アラームフラグをリセットする必要があります。そうしないと、STOP モード移行手順が無視され、プログラムが実行され続けます。 STOP モードに移行する前にアプリケーションが外部オシレータ (外部クロック) を無効にする必要がある場合、最初にシステムクロックソースを HSI に切り替え、次いで HSEON ビットをクリアする必要があります。 そうしないと、STOP モードに移行する前に HSEON ビットが 1 のままであった場合、STOP モードに移行するときに外部オシレータ (外部クロック) 障害を検出して誤動作を回避するべく、セキュリティシステム (CSS) 機能を有効にする必要があります。
モードの終了	WFI 命令を使用して移行した場合 : - 割り込みモードに設定されている任意の EXTI ライン (対応する EXTI 割り込みベクタが NVIC で有効になっている必要があります)。 - ウェイクアップモードでプログラムした場合は、特定の通信ペリフェラル (CEC、USART、I2C) 割り込みの一部 (ペリフェラルはウェイクアップモードでプログラムする必要があり、対応する割り込みベクタが NVIC で有効になっている必要があります)。 表 37 : ベクタテーブルを参照してください。 WFE 命令を使用して移行した場合 : イベントモードに設定されている任意の EXTI ライン セクション 12.2.3 : イベント管理 (220 ページ) を参照してください。
ウェイクアップ遅延時間	HSI ウェイクアップ時間 + 低電力モードからのレギュレータウェイクアップ時間

6.3.5 STANDBY モード

STANDBY モードでは、消費電力を最も少なくできます。このモードは、電圧レギュレータを無効にした状態の Cortex®-M0 のディープスリープ (deepsleep) モードに基づきます。結果として、1.8 V ドメインの電力がオフになります。PLL、HSI オシレータ、HSE オシレータもオフになります。RTC ドメインと STANDBY 回路のレジスタを除いて、SRAM とレジスタの内容は失われます (図 7 を参照)。

注意 : STM32F0x8 デバイスでは STANDBY モードを使用できません。

STANDBY モードへの移行

STANDBY モードへの移行方法の詳細については、表 18 を参照してください。

STANDBY モードでは、個別の制御ビットをプログラミングすることによって、次の機能を選択できます。

- 独立型ウォッチドッグ (IWDG) : IWDG は、キーレジスタへの書き込みによって、またはハードウェアオプションによって起動します。ウォッチドッグの動作がいったん開始されると、リセット以外では停止できません。参照: セクション 22.3 : IWDG の機能説明、セクション 22 : 独立型ウォッチドッグ (IWDG)
- リアルタイムクロック (RTC) : この設定は RTC ドメイン制御レジスタ (RCC_BDCR) の RTCEN ビットで行います。
- 内部 RC オシレータ (LSI) : この設定は 制御/ステータスレジスタ (RCC_CSR) の LSION ビットで行います。
- 外部 32.768 kHz オシレータ (LSE) : この設定は RTC ドメイン制御レジスタ (RCC_BDCR) の LSEON ビットで行います。

STANDBY モードの終了

マイクロコントローラは、外部リセット (NRST ピン)、IWDG リセット、有効な WKUPx ピンのうちの 1 つの立ち上がりエッジ、または RTC イベントのいずれかが発生すると、STANDBY モードを終了します。電源制御/ステータスレジスタ (PWR_CSR) を除くすべてのレジスタは、STANDBY モードからのウェイクアップ後にリセットされます。

STANDBY モードからのウェイクアップ後、プログラム実行はリセット後と同様にリスタートされます (ブートピン信号のサンプリング、オプションバイトローディング、リセットベクタのフェッチなど)。電源制御/ステータスレジスタ (PWR_CSR) の SBF ステータスフラグは MCU が STANDBY モードにあったことを示します。

STANDBY モードの終了方法の詳細については、表 18 を参照してください。

表 18. STANDBY モード

STANDBY モード	説明
モードへの移行	次の条件下での WFI (Wait for Interrupt) または WFE (Wait for Event) – Cortex®-M0 システム制御レジスタの SLEEPDEEP をセット – 電源制御レジスタ (PWR_CR) の PDDS ビットをセット – 電源制御/ステータスレジスタ (PWR_CSR) の WUF ビットをクリア
モードの終了	WKUP ピンの立ち上がりエッジ、RTC アラームイベントの立ち上がりエッジ、NRST ピンによる外部リセット、IWDG リセット
ウェイクアップ遅延時間	リセットフェーズ

STANDBY モードにおける I/O の状態

STANDBY モードでは、以下のピンを除き、すべての I/O ピンがハイインピーダンス状態となります。

- リセットパッド (有効)
- RTC または LSE で設定した場合、PC13、PC14、および PC15
- WKUPx ピン

デバッグモード

デフォルトでは、デバッグ機能を使用されているときにアプリケーションが MCU を STOP または STANDBY モードにすると、デバッグ接続は失われます。これは、Cortex[®]-M0 コアにクロックが供給されなくなるためです。

ただし、DBGMCU_CR レジスタの設定ビットをセットすることによって、低電力モードを多用しているときでも、ソフトウェアのデバッグを行うことができます。

6.3.6 低電力モードからの自動ウェイクアップ

RTC は、RTC アラームを使用して、外部割り込み (自動ウェイクアップモード) に頼ることなく、低電力モードから MCU をウェイクアップするために使用できます。RTC を、一定の時間間隔で STOP モードや STANDBY モードからウェイクアップさせるためのプログラム可能なタイムベースとすることができます。この目的のため、[RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1:0] ビットをプログラムすることによって、次の 3 つの代替 RTC クロックソースのうちから 2 つを選択できます。

- 低電力 32.768 kHz 外部クリスタルオシレータ (LSE OSC)
このクロックソースは、非常に少ない消費電力 (標準的な条件下で追加消費電力 1 μ A 未満) で高精度のタイムベースとなります。
- 低電力内部 RC オシレータ (LSI)
このクロックソースには、32.768 kHz クリスタルのコストを節約できるという利点があります。この内部 RC オシレータは、最小限の消費電力を追加するように設計されています。

RTC アラームイベントによって STOP モードからウェイクアップさせるには、次の設定が必要です。

- EXTI ライン 17 を立ち上がりエッジを検知するように設定します。
- RTC が RTC アラームを生成するように設定します。

STANDBY モードからウェイクアップするには、EXTI ライン 17 の設定は必要ありません。

6.4 電源制御レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

6.4.1 電源制御レジスタ (PWR_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000 (STANDBY モードからのウェイクアップによってリセット)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBP	PLS[2:0]		PVDE	CSBF	CWUF	PDDS	LPDS	
							rw	rw	rw	rw	rw	rc_w1	rc_w1	rw	rw

ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **DBP** : RTC ドメイン書き込み保護の無効化

リセット状態で、RTC およびバックアップレジスタは不要な書き込みアクセスから保護されます。これらのレジスタへの書き込みアクセスを可能にするには、このビットをセットする必要があります。

0 : RTC およびバックアップレジスタへのアクセスは無効です。

1 : RTC およびバックアップレジスタへのアクセスは有効です。

ビット 7:5 **PLS[2:0]** : PVD レベル選択

これらのビットは、電源電圧検出器によって検出される電圧閾値を選択するために、ソフトウェアで書き込みます。

PVD_LOCK が [SYSCFG 設定レジスタ 2 \(SYSCFG_CFGR2\)](#) で有効化されると、PLS[2:0] ビットはプログラムできなくなります。

000 : PVD 閾値 0

001 : PVD 閾値 1

010 : PVD 閾値 2

011 : PVD 閾値 3

100 : PVD 閾値 4

101 : PVD 閾値 5

110 : PVD 閾値 6

111 : PVD 閾値 7

詳細については、データシートの電気的特性を参照してください。

ビット 4 **PVDE** : 電源電圧検出器有効化

このビットは、ソフトウェアによってセット/クリアされます。PVD_LOCK が [SYSCFG 設定レジスタ 2 \(SYSCFG_CFGR2\)](#) レジスタで有効化されると、PVDE ビットはプログラムできなくなります。

0 : PVD は無効です。

1 : PVD は有効です。

ビット 3 **CSBF** : STANDBY フラグのクリア

このビットは常に 0 として読み出されます。

0 : 影響なし。

1 : SBF STANDBY フラグのクリア (書き込み)

- ビット 2 **CWUF** : ウェイクアップフラグのクリア
このビットは常に 0 として読み出されます。
0 : 影響なし。
1 : WUF ウェイクアップフラグを 2 システムクロックサイクル後にクリアします (書き込み)。
- ビット 1 **PDDS** : パワーダウンディープスリープ
このビットは、ソフトウェアによってセット/クリアされます。LPDS ビットとの組み合わせで動作します。
0 : CPU がディープスリープに移行すると、STOP モードに移行します。レギュレータの状態は LPDS ビットによって変化します。
1 : CPU がディープスリープに移行すると、STANDBY モードに移行します。
- ビット 0 **LPDS** : 低電力ディープスリープ
このビットは、ソフトウェアによってセット/クリアされます。このビットは PDDS ビットとの組み合わせで動作します
0 : 電圧レギュレータは STOP モードの間もオン状態を保ちます。
1 : 電圧レギュレータは STOP モードの間は低電力モードになります。
注 : **STOP モードで機能するペリフェラルでクロックが必要な場合、電源コントローラでは自動的に電圧レギュレータを低電力モードから通常モードに切り替え、このリクエストが消えるまでこのモードを維持します。**

6.4.2 電源制御/ステータスレジスタ (PWR_CSR)

アドレスオフセット : 0x04
リセット値 : 0x0000 000X (STANDBY モードからのウェイクアップによってリセットされない)
このレジスタを読み出すには、通常の APB 読み出しに対して、追加の APB サイクルが必要です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EWUP8	EWUP7	EWUP6	EWUP5	EWUP4	EWUP3 Res.	EWUP2	EWUP1	Res.	Res.	Res.	Res.	VREF INT RDY	PVDO	SBF	WUF
rw	rw	rw	rw	rw	rw	rw	rw					r	r	r	r

- ビット 31:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15:8 **EWUPx** : WKUPx ピンを有効化
これらのビットは、ソフトウェアによってセット/クリアされます。
0 : WKUPx ピンは汎用 I/O として使用されます。この WKUPx ピンでイベントが発生しても、デバイスは STANDBY モードからウェイクアップしません。
1 : WKUPx ピンは STANDBY モードからのウェイクアップに使用され、強制的にプルダウン入力モードに設定されます (WKUPx ピンの立ち上がりエッジで、システムは STANDBY モードからウェイクアップします)。
注 : **これらのビットは、システムリセットによってリセットされます。**
注 : **WKUP3 ~ WKUP8 ピンは STM32F07x および STM32F09x デバイスでのみ使用可能です。**
- ビット 7:4 予約済みであり、リセット値に保持する必要があります。

- ビット 3 **VREFINTRDY** : VREFINT 基準電圧レディ
- このビットは、内部基準電圧 VREFINT の状態を示すために、ハードウェアによってセット／クリアされます。
- 0 : VREFINT はレディ状態ではありません。
- 1 : VREFINT はレディ状態です。
- 注 :** このフラグは、**POR が外部から (NPOR ピンを通じて) 提供される STM32F0x8 デバイスでのみ有効です。STM32F0x1/F0x2 デバイスで、内部 POR は、リセットが解除される前に VREFINT が安定するまで待ちます。**
- ビット 2 **PVDO** : PVD 出力
- このビットは、ハードウェアによってセット／クリアされます。PVDE ビットによって PVD が有効化されている場合のみ有効です。
- 0 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より高いです。
- 1 : V_{DD} は PLS[2:0] ビットで選択された PVD 閾値より低いです。
- 注 :**
- PVD は STANDBY モードによって停止します。したがって、このビットは STANDBY またはリセット後、PVDE ビットがセットされるまで 0 になります。
 - PVD が有効化され、PWR_CR レジスタで設定されると、PVDO は外部割り込みコントローラを通じて割り込みを生成するために使用できます。
- ビット 1 **SBF** : STANDBY フラグ
- このビットは、デバイスが STANDBY モードに移行したときにハードウェアによってセットされ、POR/PDR (パワーオンリセット／パワーダウリセット)、または **電源制御レジスタ (PWR_CR)** の CSBF ビットをセットすることによってのみクリアされます。
- 0 : デバイスは STANDBY モードではありません。
- 1 : デバイスは STANDBY モードです。
- ビット 0 **WUF** : ウェイクアップフラグ
- このビットは、デバイスがウェイクアップイベントを受信したことを示すために、ハードウェアによってセットされます。これは、システムリセットによって、または **電源制御レジスタ (PWR_CR)** の CWUF ビットをセットすることによってクリアされます。
- 0 : ウェイクアップイベントは発生していません。
- 1 : 有効な WKUPx ピンのいずれか、または RTC アラームからウェイクアップイベントを受信しました。
- 注 :** **WKUPx ピンのレベルがすでにハイであっても、(EWUPx ビットをセットすることによって) このピンが有効化された場合、追加のウェイクアップイベントが検出されます。**

6.4.3 PWR レジスタマップ

次の表に、PWR レジスタマップとリセット値の一覧を示します。

表 19. PWR レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	PWR_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBP	PLS[2:0]			PVDE	CSBF	CWUF	2	1	0
	リセット値																								0	0	0	0	0	0	0	0	0	0
0x004	PWR_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWUP8	EWUP7	EWUP6	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1		Res.	Res.	Res.	Res.	VREFINTRDY	PVDO	SBF	WUF
	リセット値																	0	0	0	0	0	0	0	0					X	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

7 リセットおよびクロック制御 (RCC)

7.1 リセット

リセットには、システムリセット、電源リセット、RTC ドメインリセットの 3 種類があります。

7.1.1 電源リセット

電源リセットは、次のいずれかのイベントが発生したときに生成されます。

1. パワーオン/パワーダウンリセット (POR/PDR リセット)
2. STANDBY モードの終了

電源リセットは、RTC ドメインを除くすべてのレジスタをリセット値にセットします ([図 7 : 電源の概要](#)を参照)。

STM32F0x8 デバイスでは、POR/PDR リセットは機能しないため STANDBY モードは使用できません。電源リセットは、外部 NPOR ピンから供給する必要があります (すべての電源電圧が安定すると、アクティブロー、およびアプリケーションによって解放される)。

7.1.2 システムリセット

システムリセットは、クロックコントローラ CSR レジスタのリセットフラグと RTC ドメインのレジスタを除くすべてのレジスタをリセット値にセットします ([図 7 : 電源の概要](#)を参照)。

システムリセットは、次のイベントのいずれかの発生時に生成されます。

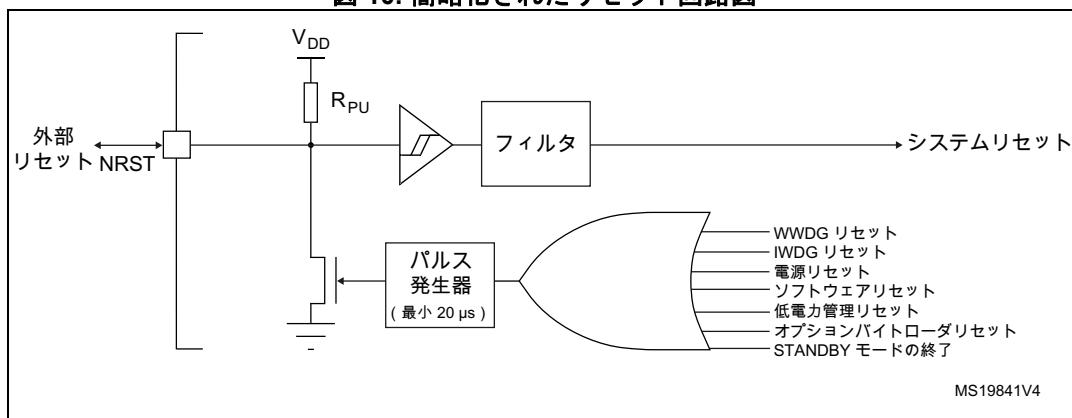
1. NRST ピンのローレベル (外部リセット)
2. ウィンドウ型ウォッチドッグイベント (WWDG リセット)
3. 独立型ウォッチドッグイベント (IWDG リセット)
4. ソフトウェアリセット (SW リセット) ([ソフトウェアリセット](#)を参照)
5. 低電力管理リセット ([低電力管理リセット](#)を参照)
6. オプションバイトローダリセット ([オプションバイトローダリセット](#)を参照)
7. 電源リセット

リセットソースは、制御/ステータスレジスタ (RCC_CSR) のリセットフラグを確認することによって識別できます ([セクション 7.4.10 : 制御/ステータスレジスタ \(RCC_CSR\)](#)を参照)。

これらのソースは NRST ピンに作用し、遅延フェーズの間、ピンをローに保ちます。リセットサービスルーチンベクタは、メモリマップのアドレス 0x0000_0004 に固定されています。

デバイスに与えられるシステムリセット信号は、NRST ピンに出力されます。パルス発生回路は、各内部リセットソースについて 20 μ s の最小リセットパルス期間を保証します。外部リセットの場合、リセットパルスは NRST ピンがローにアサートされているときに生成されます。

図 10. 簡略化されたリセット回路図



ソフトウェアリセット

デバイス上でソフトウェアリセットを実行するには、Cortex[®]-M0 のアプリケーション割り込みおよびリセット制御レジスタの SYSRESETREQ ビットをセットする必要があります。詳細については、Cortex[™]-M0 Technical Reference Manual を参照してください。

低電力管理リセット

低電力管理リセットを生成するには、2 つの方法があります。

1. STANDBY モードに移行するときにリセットを生成します。
この種類のリセットは、ユーザオプションバイトの nRST_STDBY ビットをリセットすることで有効になります。この場合、STANDBY モードへの遷移シーケンスが正常に実行されるたびに、STANDBY モードに移行する代わりにデバイスがリセットされます。
2. STOP モードに移行するときにリセットを生成します：
この種類のリセットは、ユーザオプションバイトの nRST_STOP ビットをリセットすることで有効になります。この場合、STOP モードへの遷移シーケンスが正常に実行されるたびに、STOP モードに移行する代わりにデバイスがリセットされます。

ユーザオプションバイトの詳細については、[セクション 4: オプションバイト](#)を参照してください。

オプションバイトローダリセット

オプションバイトローダリセットは、FLASH_CR レジスタの OBL_LAUNCH ビット (ビット 13) がセットされると生成されます。このビットは、ソフトウェアによるオプションバイトローディングを起動するために使用されます。

7.1.3 RTC ドメインリセット

RTC ドメインには、RTC ドメインだけに影響する 2 種類の特定のリセットがあります (図 7 : 電源の概要)。

RTC ドメインリセットは、LSE オシレータ、RTC、バックアップレジスタ、および RCC *RTC ドメイン制御レジスタ (RCC_BDCR)* にのみ影響します。これは、次のいずれかのイベントが発生したときに生成されます。

1. *RTC ドメイン制御レジスタ (RCC_BDCR)* の BDRST ビットをセットすることでトリガされるソフトウェアリセット
2. V_{BAT} がローのときに切断されると、 V_{DD} がパワーアップします。

バックアップレジスタは、次のいずれかのイベントが発生したときにリセットされます。

1. RTC タンパ検出イベント
2. 読み出し保護のレベル 1 からレベル 0 への変更

7.2 クロック

さまざまなクロックソースを使用して、システムクロック (SYSCLK) を駆動できます。

- HSI 8 MHz RC オシレータクロック
- HSE オシレータクロック
- PLL クロック
- HSI48 48 MHz RC オシレータクロック (STM32F04x, STM32F07x および STM32F09x デバイスでのみ使用可能)

デバイスには、次の追加のクロックソースがあります。

- 40 kHz 低速内蔵 RC (LSI RC)。独立型ウォッチドッグを駆動し、オプションで、STOP/STANDBY モードからの自動ウェイクアップに使用される RTC を駆動します。
- 32.768 kHz 低速外部クリスタル (LSE クリスタル)。オプションで、リアルタイムクロック (RTCCLK) を駆動します。
- ADC 専用の 14 MHz 高速内部 RC (HSI14)。

それぞれのクロックソースは、使用しないときに個別にオン/オフを切り替えて、電力消費を最適化可能です。

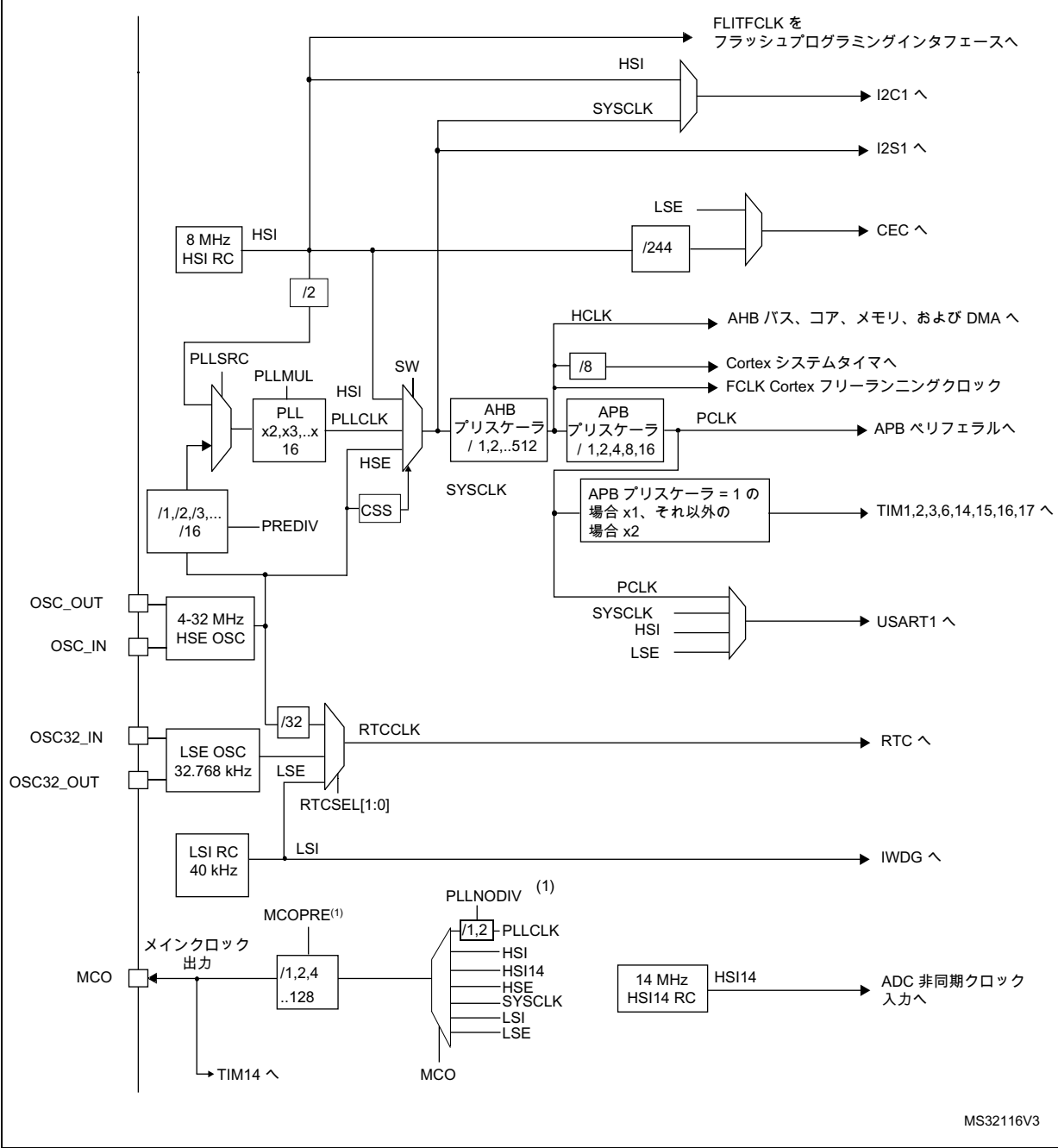
複数のプリスケラを使用して AHB および APB ドメインの周波数を設定できます。AHB および APB ドメインの最大周波数は 48 MHz です。

すべてのペリフェラルクロックは、次を除いてそれらのバスクロックから生成されます (AHB の HCLK または APB の PCLK)。

- 常に HSI クロックであるフラッシュメモリのプログラミングインタフェースクロック (FLITFCLK)
- 常に HSI クロックであるオプションバイトリダククロック
- 次の 2 つのうちのいずれかのソースから生成される ADC クロック (ソフトウェアによって選択される):
 - 常に最大サンプリングレートで実行される専用の HSI14 クロック
 - 2 または 4 分周された APB クロック (PCLK)
- 次の 4 つのうちのいずれかのソースから生成される USART1 クロック、USART2 クロック (STM32F07x および STM32F09x デバイスでのみ)、および USART3 クロック (STM32F09x デバイスでのみ) (ソフトウェアによって選択される):
 - システムクロック
 - HSI クロック
 - LSE クロック
 - APB クロック (PCLK)
- 次の 2 つのうちのいずれかのソースから生成される I2C1 クロック (ソフトウェアによって選択される):
 - システムクロック
 - HSI クロック
- 次の 2 つのうちのいずれかのソースから生成される USB クロック (ソフトウェアによって選択される):
 - PLL クロック
 - HSI48 クロック
- 244 分周された HSI クロックまたは LSE クロックから生成される CEC クロック
- 常にシステムクロックである I2S1 および I2S2 クロック
- LSE、LSI、または 32 分周された HSE クロックから生成される RTC クロック
- タイマクロック周波数は、ハードウェアによって自動的に固定されます。これには、2 つのケースがあります。
 - APB プリスケアラが 1 の場合、タイマクロック周波数は APB ドメインのクロックと同じ周波数に設定されます。
 - そうでない場合は、APB ドメインの周波数の 2 倍に設定されます。
- IWDG クロックは、常に LSI クロックから生成されます。

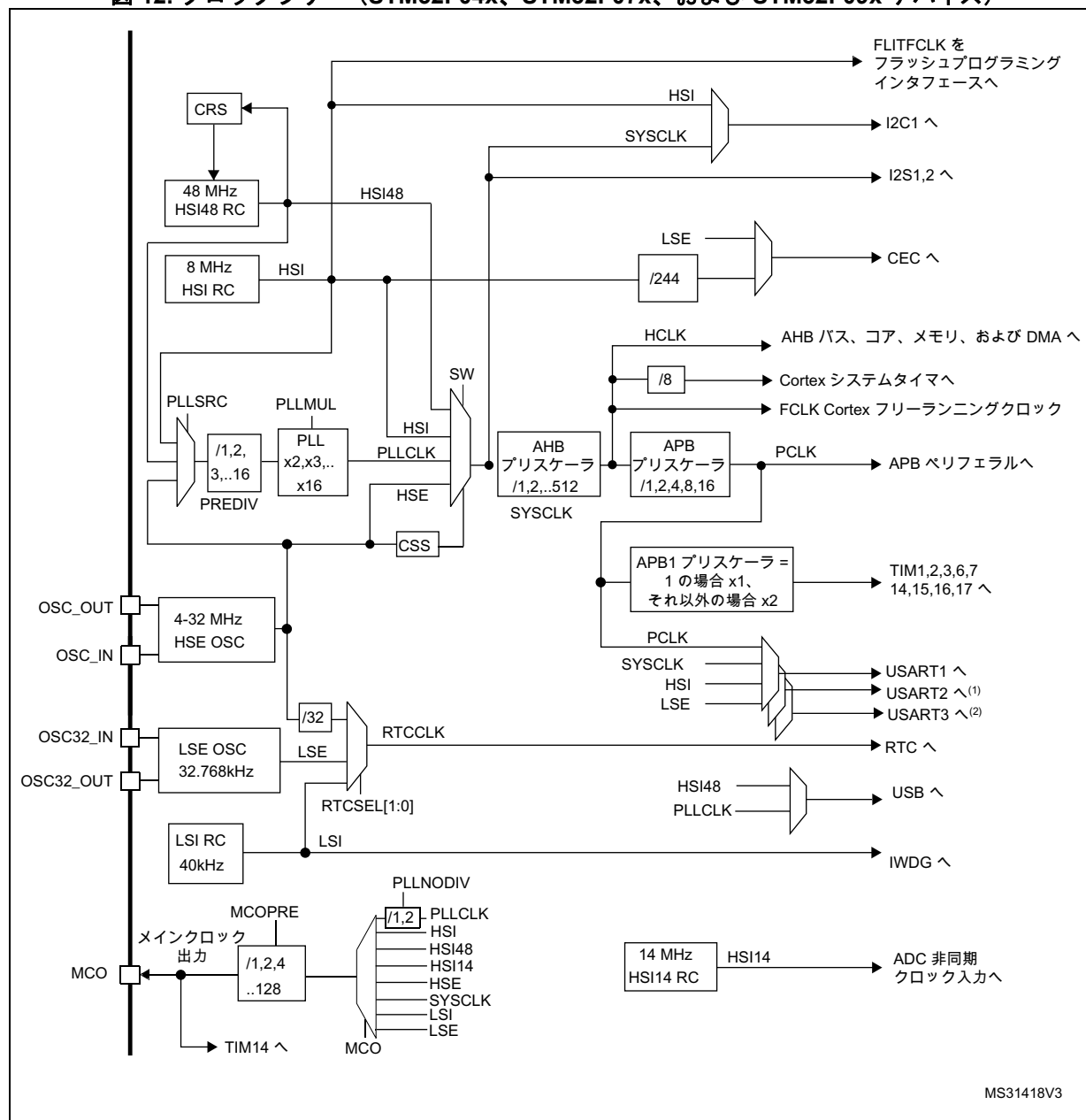
RCC は Cortex システムタイマ (SysTick) の外部クロックに、8 分周した AHB クロック (HCLK) を供給します。SysTick は、この分周されたクロックが直接 Cortex クロック (HCLK) で動作でき、SysTick 制御およびステータスレジスタで設定可能です。

図 11. クロックツリー (STM32F03x および STM32F05x デバイス)



1. STM32F05x デバイスでは使用できません。

図 12. クロックツリー (STM32F04x、STM32F07x、および STM32F09x デバイス)



1. STM32F04x デバイスでは使用できません。
2. STM32F04x および STM32F07x デバイスでは使用できません。

FCLK は Cortex®-M0 のフリーランニングクロックとして動作します。詳細については、*ARM Cortex™-M0 r0p0 Technical Reference Manual (TRM)* を参照してください。

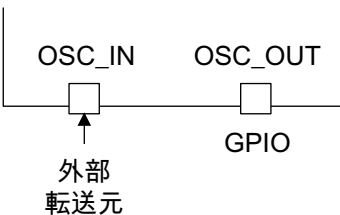
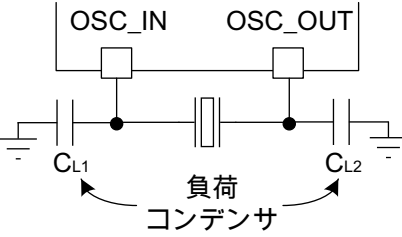
7.2.1 HSE クロック

高速外部クロック信号（HSE）は、次のどちらかのクロックソースから生成できます。

- HSE 外部クリスタル／セラミック発振子
- HSE ユーザ外部クロック

波形ひずみと発振開始時の安定化までの時間を少なくするために、発振子と負荷コンデンサはオシレータのピンのできるだけ近くに配置する必要があります。負荷コンデンサの値は、選択したオシレータに応じて調整する必要があります。

図 13. HSE/LSE クロックソース

クロックソース	ハードウェア構成
外部クロック	 MSv31915V1
クリスタル／セラミック発振子	 MSv31916V1

外部クリスタル／セラミック発振子 (HSE クリスタル)

4 から 32 MHz の外部オシレータには、メインクロックの周波数を非常に高い精度で生成できるという利点があります。

関連するハードウェア構成を図 13 に示します。詳細については、データシートの電気的特性のセクションを参照してください。

クロック制御レジスタ (RCC_CR) の HSERDY フラグは、HSE オシレータが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、クロックは出力されません。**クロック割り込みレジスタ (RCC_CIR)** で有効になっていれば、割り込みを生成することができます。

HSE クリスタルは、**クロック制御レジスタ (RCC_CR)** の HSEON ビットを使用してオン／オフできます。

コード例については、「付録」セクション **A.3.1 : HSE 起動シーケンスのコード例** を参照してください。

注意 : HSE オシレータをオンにするには、HSEON ビットのセット後に内部安定カウンタで 512 HSE クロックパルスを検出する必要があります。デバイスにクリスタルや発振子が接続されていない場合でも、OSC_IN ピンの過剰な外部ノイズによってオシレータが起動する場合があります。オシレータを起動すると、オフにするシーケンスを完了するために、別の 6 HSE クロックパルスが必要となります。何らかの理由で OSC_IN ピン上の振動がなくなった場合、オシレータをオフにすることはできず、OSC ピンは他で使用できないようロックされ、不要な消費電力が発生します。このような状況を避けるには、このような場合にオシレータをオフにすることができるクロックセキュリティシステム (CSS) を常に有効にすることを強くお勧めします。

外部ソース (HSE バイパス)

このモードでは、外部クロックソースが必要です。最大 32 MHz までの周波数を使用できます。このモードを選択するには、**クロック制御レジスタ (RCC_CR)** HSEBYP および HSEON ビットをセットします。周波数 (データシートを参照) に応じて 40 ~ 60% までのデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC_IN ピンを駆動する必要があり、その間、OSC_OUT ピンは GPIO として使用できます。図 13 を参照してください。

7.2.2 HSI クロック

HSI クロック信号は、内蔵 8 MHz RC オシレータから生成され、システムクロックとして直接、または PLL 入力として使用できます。

HSI RC オシレータには、低コスト (外付部品なし) でクロックソースを供給できるという利点があります。同時に、HSE クリスタルオシレータよりも起動時間を短縮できますが、較正を実施しても、外部クリスタルオシレータやセラミック発振子よりも周波数の精度は劣ります。

較正

RC オシレータの周波数は、製造工程でのばらつきのため、チップごとに異なります。このため、 $T_A = 25^{\circ}\text{C}$ で 1 % の精度を確保するよう、製造時に ST 社においてデバイスごとの較正が行われています。

リセット後、工場較正值が **クロック制御レジスタ (RCC_CR)** の HSICAL[7:0] ビットにロードされます。

アプリケーションが電圧または温度の変動を受ける場合は、これが RC オシレータの速度に影響を与えることがあります。**クロック制御レジスタ (RCC_CR)** の HSITRIM[4:0] ビットを使用して、アプリケーションで HSI 周波数をトリミングできます。



HSI 周波数変動の測定方法の詳細については、[セクション 7.2.13 : TIM14 を使用した内部／外部クロックの測定 \(112 ページ\)](#) を参照してください。

クロック制御レジスタ (RCC_CR) の HSIIRDY フラグは、HSI RC が安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、HSI RC 出力クロックは出力されません。

HSI RC は [クロック制御レジスタ \(RCC_CR\)](#) の HSION ビットを使用してオン／オフの切り替えができます。

HSI 信号は、HSE クリスタルオシレータに障害がある場合のバックアップソース（補助クロック）としても使用できます。[セクション 7.2.8 : クロックセキュリティシステム \(CSS\) \(110 ページ\)](#) を参照してください。

また、HSI クロックを MCO マルチプレクサに駆動することができます。その後、クロックはタイマ 14 に駆動することができ、これによりユーザはオシレータを較正できるようになります。

7.2.3 HSI48 クロック

STM32F04x、STM32F07x、および STM32F09x デバイスでのみ、HSI48 クロック信号は内部 48 MHz RC オシレータから生成され、システムクロックとして直接使用するか、分周され、PLL 入力として使用できます。

内部 48 MHz RC オシレータは、主に、特殊なクロックリカバリシステム (CRS) 回路によって USB ペリフェラルに高精度のクロックを提供します。非常に少ないステップで動作中のオシレータの周波数を自動的かつ迅速に調整するために、USB SOF 信号、LSE、または外部信号を使用できます。システムが RUN モードの場合は、オシレータをシステムクロックソースとして使用することもできます。これは、システムが STOP または STANDBY モードに移行した直後に無効化されます。CRS が使用されない場合、HSI48 RC オシレータは製造工程でのばらつきの影響を受けるデフォルトの周波数で実行されます。このため、 $T_A = 25^\circ\text{C}$ で 3 % 以下の精度を確保するよう、製造時に ST 社においてデバイスごとの較正が行われています。

CRS ペリフェラルの設定および使用方法の詳細については、[セクション 8](#)を参照してください。

クロック制御レジスタ (RCC_CR) の HSI48RDY フラグは、HSI48 RC が安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、HSI48 RC 出力クロックはリリースされません。

HSI48 RC は、クロック制御レジスタ (RCC_CR) の HSI48ON ビットを使用してオン／オフできます。オシレータも、USB のクロックソースとして選択され、ペリフェラルが有効になるとすぐに、(HSI48ON ビットを 1 に強制するハードウェアによって) 自動的に有効になります。

さらに、HSI48 クロックを MCO マルチプレクサに駆動することができ、他のアプリケーションコンポーネントのクロックソースとして使用できます。

7.2.4 PLL

内部 PLL は、HSI、分周された HSI48、または HSE 出力クロック周波数を逡倍するために使用できません。[図 10 : 簡略化されたリセット回路図](#)、[図 13 : HSE/LSE クロックソース](#)、および [クロック制御レジスタ \(RCC_CR\)](#) を参照してください。

PLL 設定 (入力クロック、前置分周器、および逡倍数の選択) は、PLL を有効にする前に実行する必要があります。PLL が有効になると、これらのパラメータは変更できません。

PLL 設定を変更するには、次の手順に従います。

1. PLLON を 0 にセットして PLL を無効にします。
2. PLLRDY がクリアされるまで待ちます。PLL が完全に停止します。
3. 任意のパラメータを変更します。
4. PLLON を 1 にセットして PLL を再度有効にします。
5. PLLRDY がセットされるまで待ちます。

[クロック割り込みレジスタ \(RCC_CIR\)](#) で有効になっていれば、PLL がレディ状態になると割り込みを生成することができます。

PLL 出力周波数は、16 ~ 48 MHz の範囲でセットする必要があります。

コード例については、「付録」セクション [A.3.2 : PLL 設定変更のコード例](#) を参照してください。

7.2.5 LSE クロック

LSE クリスタルは、32.768 kHz の低速外部クリスタルまたはセラミック発振子です。時計/カレンダー、その他のタイミング機能のためのリアルタイムクロックペリフェラル (RTC) に、低電力ながら高精度のクロックソースを供給できるという利点があります。

LSE クリスタルは、[RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEON ビットを使用してオン/オフの切り替えができます。最適な堅牢性と、起動時間の短縮、および消費電力の低減を実現するために、クリスタルオシレータの駆動能力は、[RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEDRV[1:0] ビットによって、実行時に変更できます。

[RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の LSERDY フラグは、LSE クリスタルが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、LSE クリスタル出力クロック信号はリリースされません。[クロック割り込みレジスタ \(RCC_CIR\)](#) で有効になっていれば、割り込みを生成することができます。

注意 : LSE オシレータをオンにするには、LSEON ビットのセット後に内部安定カウンタで 4096 LSE クロックパルスを検出する必要があります。デバイスにクリスタルや発振子が接続されていない場合でも、OSC32_IN ピンの過剰な外部ノイズによってオシレータが起動する場合があります。オシレータを起動すると、オフにするシーケンスを完了するために、別の 6 LSE クロックパルスが必要となります。何らかの理由で OSC_IN ピン上の振動がなくなった場合、オシレータをオフにすることはできず、OSC32 ピンは他で使用できないようロックされ、不要な消費電力が発生します。このような状況を回復する唯一の方法は、ソフトウェアによって RTC ドメインをリセットすることです。

外部ソース (LSE バイパス)

このモードでは、外部クロックソースが必要です。最大 1 MHz までの周波数を使用できます。このモードを選択するには、[RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の LSEBYP および LSEON ビットをセットします。50 % までのデューティサイクルを持つ外部クロック信号 (矩形波、正弦波、または三角波) で OSC_32_IN ピンを駆動する必要があり、その間、OSC_32_OUT ピンを GPIO として使用することができます。[図 13](#) を参照してください。



7.2.6 LSI クロック

LSI RC は、独立型ウォッチドッグ (IWDG) および RTC の STOP モードおよび STANDBY モードでの動作を可能にする低電力のクロックソースとして動作します。クロック周波数は約 40 kHz (30 kHz と 60 kHz の間) です。詳細については、データシートの電氣的特性のセクションを参照してください。

LSI RC は、**制御/ステータスレジスタ (RCC_CSR)** の LSION ビットを使用してオン/オフの切り替えができます。

制御/ステータスレジスタ (RCC_CSR) の LSIRDY フラグは、LSI オシレータが安定しているかどうかを示します。起動時、このビットがハードウェアによってセットされるまで、クロックは出力されません。**クロック割り込みレジスタ (RCC_CIR)** で有効になっていれば、割り込みを生成することができます。

7.2.7 システムクロック (SYSCLK) の選択

さまざまなクロックソースを使用して、システムクロック (SYSCLK) を駆動できます。

- HSI オシレータ
- HSE オシレータ
- PLL
- HSI48 オシレータ (STM32F04x、STM32F07x、および STM32F09x デバイスでのみ使用可能)

システムリセット後は、HSI オシレータがシステムクロックとして選択されています。クロックソースが直接、または PLL を経由してシステムクロックとして使用されているときに、このクロック信号を停止することはできません。

あるクロックソースから別のクロックソースへの切り替えは、切り替え後に使用するクロックソースの準備ができていない場合 (起動遅延時間を経てクロックが安定している状態、または、PLL がロックされている状態) にのみ行われます。準備ができていないクロックソースが選択された場合は、クロックソースの準備ができたときに切り替えが行われます。**クロック制御レジスタ (RCC_CR)** のステータスビットは、どのクロックの準備ができていないか、およびどのクロックがシステムクロックとして使用されているかを示します。

7.2.8 クロックセキュリティシステム (CSS)

クロックセキュリティシステムはソフトウェアで有効にできます。この場合、HSE オシレータのスタートアップ遅延時間の後にクロック検出回路が有効になり、オシレータが停止すると検出回路も無効になります。

HSE クロックに異常が検出された場合、HSE オシレータは自動的に無効になり、クロック障害イベントが高機能制御タイマ (TIM1) および汎用タイマ (TIM15、TIM16、および TIM17) のブレーク入力に送られます。また、ソフトウェアに異常を通知する割り込み (クロックセキュリティシステム割り込み (CSSI)) が生成されるので、MCU は対応処理を行うことができます。CSSI は、Cortex®-M0 NMI (ノンマスカブル割り込み) 例外ベクタにリンクされています。

注: **CSS を有効にすると、HSE クロックに障害が発生した場合に CSS 割り込みが発生し、NMI が自動的に生成されます。NMI は、CSS 割り込みペンディングビットがクリアされない限り、無限に実行されます。そのため、NMI 割り込みサービスルーチン (ISR) では、クロック割り込みレジスタ (RCC_CIR) の CSSC ビットをセットして CSS 割り込みをクリアする必要があります。**

HSE オシレータが直接的または間接的にシステムクロックとして使用されている場合 (間接的とは、PLL 入力クロックとして使用され、その PLL クロックがシステムクロックとして使用されることを意味する)、障害が検出されると、システムクロックが HSI オシレータに切り替えられ、HSE オシレータは無効になります。障害発生時に HSE クロック (分周されているかいないかにかかわらず) がシステムクロックとして使用されている PLL のクロック入力であった場合には、PLL も無効になります。

7.2.9 ADC クロック

ADC クロック選択は、ADC_CFGR2 内で実行されます（[セクション 13.12.5 : ADC 設定レジスタ 2 \(ADC_CFGR2\) \(264 ページ\)](#) を参照）。専用 14 MHz RC オシレータ（HSI14）または 2 または 4 分周された PCLK を選択できます。14 MHz RC オシレータは、ADC インタフェースでオン/オフ（「自動オフモード」）を切り替えるか、常に有効になるようソフトウェアで設定できます。APB クロックが ADC カーネルクロックとして選択されている場合は、ADC インタフェースで HSI 14 MHz RC オシレータをオンにすることはできません。

7.2.10 RTC クロック

RTCCLK クロックソースは HSE/32、LSE、または LSI クロックのいずれかです。これは、[RTC ドメイン制御レジスタ \(RCC_BDCR\)](#) の RTCSEL[1:0] ビットをプログラムすることによって、選択できます。この選択は、RTC ドメインをリセットしない限り変更できません。RTC を正常に動作させるには、PCLK 周波数が常に RTCCLK 周波数以上になるようにシステムを設定する必要があります。

LSE クロックは RTC ドメインに属しますが、HSE と LSI クロックはそうではありません。そのため、次のようになります。

- LSE が RTC クロックとして使用されている場合：
 - V_{DD} 供給がオフになった場合でも、 V_{BAT} の供給が保たれている限り、RTC は動作を続けます。
- LSI が RTC クロックとして使用されている場合：
 - V_{DD} 供給がオフになった場合、RTC の状態は保証されません。LSI 較正の詳細については、[セクション 7.2.6 : LSI クロック \(110 ページ\)](#) を参照してください。
- 32 分周された HSE クロックが RTC クロックとして使用されている場合：
 - V_{DD} 供給がオフになった場合や、内蔵されている電圧レギュレータがオフになった場合（1.8 V 電源ドメインへの電力供給が停止にされた場合）、RTC の状態は保証されません。

7.2.11 独立型ウォッチドッグクロック

独立型ウォッチドッグ (IWDG) がハードウェアのオプションまたはソフトウェアアクセスによって開始された場合、LSI オシレータは強制的にオンになり、オフにすることはできません。LSI オシレータの起動時の過渡期間が終わると、このクロックが IWDG に供給されます。

7.2.12 クロック信号出力

マイクロコントローラクロック出力 (MCO) 機能では、外部 MCO ピンにクロックを出力することができます。対応する GPIO ポートの設定レジスタは、オルタネート機能モードに設定されている必要があります。クロッキングシグナルのうちの 1 つを MCO クロックとして選択できます。

- HSI14
- SYSCLK
- HSI
- HSE
- 2 分周された PLL クロックまたは直接（STM32F05x デバイスでは直接接続は使用不可）
- LSE
- LSI
- HSI48（STM32F04x、STM32F07x、および STM32F09x デバイスのみ）

選択は [クロック設定レジスタ \(RCC_CFGR\)](#) の MCO[3:0] ビットによって制御されます。

コード例については、「付録」セクション [A.3.3 : MCO 選択のコード例](#) を参照してください。

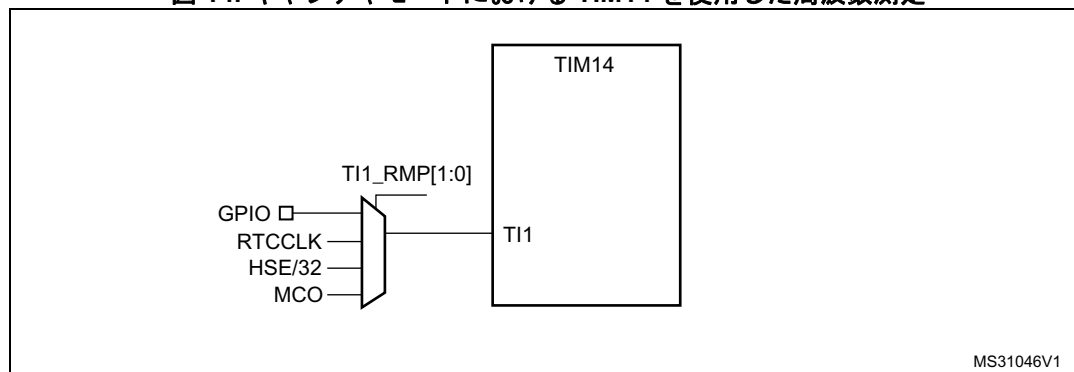


STM32F03x、STM32F04x、STM32F07x、および STM32F09x デバイスでは、このレジスタの追加ビット PLLNODIV が、MCO への PLL クロック入力に分周バイパスを制御します。MCO 周波数は、[クロック設定レジスタ \(RCC_CFGR\)](#) の MCOPRE[2..0] ビットで制御される設定可能なバイナリ分周器によって低減できます。

7.2.13 TIM14 を使用した内部／外部クロックの測定

TIM14 チャンネル 1 の入力キャプチャを使用することにより、ボード上のすべてのクロックソースの周波数を間接的に測定することができます。[図 14](#)を参照してください。

図 14. キャプチャモードにおける TIM14 を使用した周波数測定



タイマ 14 の入力キャプチャチャンネルは GPIO ラインまたは MCU の内部クロックです。この選択は、TIM14_OR レジスタの TI1_RMP [1:0] ビットで行います。以下を使用できます。

- TIM14 のチャンネル 1 は、GPIO に接続されます。デバイスのデータシートのオルタネート機能配置表を参照してください。
- TIM14 のチャンネル 1 は、RTCCLK に接続されます。
- TIM14 のチャンネル 1 は、HSE/32 クロックに接続されます。
- TIM14 のチャンネル 1 は、マイクロコントローラクロック出力 (MCO) に接続されます。MCO クロックの設定については、[セクション 7.2.12 : クロック信号出力](#)を参照してください。

コード例については、「付録」セクション [A.3.4 : TIM14 によるクロックの測定設定のコード例](#) を参照してください。

HSI の較正

MCO マルチプレクサを通じてチャネル 1 の入力キャプチャに LSE を接続する主な目的は、HSI システムクロックを正確に測定できるようにすることです（これには、システムクロックソースとして HSI を使用する必要があります）。LSE 信号の連続したエッジ間の HSI クロックカウント数により、内部クロック周期が測定されます。LSE クリスタルの高い精度（通常は数十 ppm）を利用することにより、同じ分解能で内部クロック周波数を判断でき、製造工程、温度、電圧に関連した周波数偏差に対してクロックソースを調整して補正することができます。

HSI オシレータには、上記の目的のためにユーザがアクセスできる専用の較正ビットがあります。

この基本概念は、相対測定（HSI/LSE 比など）ができることにあり、そのため、精度は 2 つのクロックソース間の比に密接にかかわっています。この比が大きいほど、測定精度は向上します。

LSE を使用できない場合、可能な限り正確な較正を達成するには、HSE/32 を選択することをお勧めします。

LSI の較正

LSI の較正は HSI と同じ方法で行いますが、リファレンスクロックを変更します。LSI クロックを TIM14 チャネル 1 の入力キャプチャに接続する必要があります。次に、HSE をシステムクロックソースとして定義し、LSI 信号の連続したエッジ間のクロックカウント数により、内部低速クロック周期が測定されます。

この基本概念は、相対測定（HSE/LSI 比など）ができることにあり、そのため、精度は 2 つのクロックソース間の比に密接にかかわっています。この比が大きいほど、測定精度は向上します。

HSI14 の較正

HSI14 では、周波数が高いため、正確な分解能を得ることはできません。ただし、1 つの方法として、HSE を使用し、PLL を経由してタイマ 14 で 48 MHz のクロックを生成し、より高い値に定義された HSI14 とキャプチャプリスケラを持つ入力キャプチャラインを使用することができます。その設定で、27 のイベントの比率を得られます。これは、正確な較正を行うにはまだ少し低い値です。測定の精度を上げるには、タイマ 14 の複数サイクルの後で HSI 周期をカウントすることを推奨します。この場合は、キャプチャイベントの処理にポーリングを使用する必要があります。

7.3 低電力モード

APB ペリフェラルクロックおよび DMA クロックは、ソフトウェアで無効にできます。

SLEEP モードでは CPU クロックを停止します。メモリインタフェースクロック（フラッシュおよび RAM インタフェース）は、ソフトウェアによって SLEEP モード中に停止できます。AHB - APB ブリッジのクロックは、SLEEP モード中にこれらに接続されたペリフェラルのクロックがすべて無効になった場合に、ハードウェアによって無効化されます。

STOP モードでは、コア供給ドメインのすべてのクロックを停止し、PLL と HSI、HSI48、HSI14、および HSE オシレータを無効にします。

HDMI CEC、USART1、USART2（STM32F07x および STM32F09x デバイスのみ）、USART3（STM32F09x デバイスのみ）、および I2C1 では、MCU が STOP モードの場合でも、HSI オシレータを有効にすることができます（HSI が当該ペリフェラルのクロックソースとして選択されている場合）。システムが STOP モードでレギュレータが LP モードの場合、3 つのペリフェラルのいずれから来るクロックリクエストによって、コアロジックの正確な電流駆動機能を得るために、レギュレータが MR モードに移行します。このリクエストが MCU をウェイクアップすることなく削除されると、レギュレータは LP モードに戻ります。



HDMI CEC、USART1、USART2 (STM32F07x および STM32F09x デバイスのみ) および USART3 (STM32F09x デバイスのみ) は、システムが STOP モードの場合 (LSE が当該ペリフェラルのクロックソースとして選択されている場合) で LSE オシレータが有効な場合 (LSEON) に、LSE オシレータで駆動することもできます。ただし、LSE オシレータをオンにすることはできません。

STANDBY モードでは、コア供給ドメインのすべてのクロックを停止し、PLL と HSI、HSI48、HSI14、および HSE オシレータを無効にします。

CPU のディープスリープモードは、DBGMCU_CR レジスタに DBG_STOP または DBG_STANDBY ビットをセットすることで、デバッグのためにオーバーライドすることができます。

割り込み (STOP モード) またはリセット (STANDBY モード) 後にディープスリープからウェイクアップすると、HSI オシレータがシステムクロックとして選択されます。

フラッシュプログラム操作が実行中の場合、フラッシュインタフェースアクセスが終了してから、ディープスリープモードに移行します。APB ドメインにアクセス中の場合、APB アクセスが終了してから、ディープスリープモードに移行します。

7.4 RCC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

7.4.1 クロック制御レジスタ (RCC_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 XX83 (X は未定義)。

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PLL RDY	PLLON	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
						r	rw					rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **PLL RDY** : PLL クロックレディフラグ

ハードウェアによってセットされ、PLL がロック状態であることを示します。

- 0 : PLL アンロック
- 1 : PLL ロック

ビット 24 **PLLON** : PLL イネーブル

PLL を有効にするために、ソフトウェアによってセット/クリアされます。

STOP または STANDBY モードに入るときに、ハードウェアによってクリアされます。このビットは、PLL クロックがシステムクロックとして使用されている場合、またはシステムクロックとして選択されている場合は、リセットできません。

- 0 : PLL オフ
- 1 : PLL オン

ビット 23:20 予約済みであり、リセット値に保持する必要があります。

ビット 19 **CSSON** : クロックセキュリティシステム有効化

クロックセキュリティシステムを有効にするために、ソフトウェアによってセット/クリアされます。CSSON がセットされているとき、HSE オシレータがレディになるとクロック検出回路がハードウェアによって有効にされ、HSE クロックの障害が検出された場合ハードウェアによって無効にされます。

- 0 : クロックセキュリティシステム無効 (クロック検出回路オフ)
- 1 : クロックセキュリティシステム有効 (HSE がレディであればクロック検出回路オン、そうでなければオフ)

ビット 18 **HSEBYP** : HSE クリスタルオシレータバイパス

外部クロックでオシレータをバイパスするために、ソフトウェアによってセット/クリアされます。デバイスで使用するには、HSEON ビットをセットして外部クロックを有効にする必要があります。HSEBYP ビットは、HSE オシレータが無効のときのみ、書き込みができます。

- 0 : HSE クリスタルオシレータはバイパスされません。
- 1 : HSE クリスタルオシレータが外部クロックによってバイパスされます。



ビット 17 HSERDY : HSE クロックレディフラグ

HSE オシレータが安定していることを示すために、ハードウェアによってセットされます。このビットでは HSEON リセット後に立ち上げるために、HSE オシレータクロックが 6 サイクル必要です。

- 0 : HSE オシレータはレディ状態ではありません。
- 1 : HSE オシレータはレディ状態です。

ビット 16 HSEON : HSE クロック有効化

ソフトウェアでセット/クリアされます。

STOP または STANDBY モードに入るときに、HSE オシレータを停止するためにハードウェアによってクリアされます。HSE オシレータが直接的または間接的にシステムクロックとして使用されている場合は、このビットをリセットできません。

- 0 : HSE オシレータオフ
- 1 : HSE オシレータオン

ビット 15:8 HSICAL[7:0] : HSI クロック校正

これらのビットは、スタートアップ時に自動的に初期化されます。これらは HSITRIM 設定を通じてソフトウェアによって調整されます。

ビット 7:3 HSITRIM[4:0] : HSI クロックトリミング

これらのビットにより、ユーザプログラミング可能なトリミング値が使用でき、この値は HSICAL[7:0] ビットに加算されます。HSI の周波数に影響する電圧や温度の変化に対応できるようにプログラミングできます。

デフォルト値は 16 で、HSICAL 値に追加すると、HSI を $8\text{ MHz} \pm 1\%$ にトリミングします。トリミングステップは、2 つの連続する HSICAL ステップの間で、40 kHz 前後です。

注： レジスタで増加した値によって、より高いクロック周波数を得られます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 HSIRDY : HSI クロックレディフラグ

HSI オシレータが安定していることを示すために、ハードウェアによってセットされます。HSION ビットがクリアされた後、HSIRDY は、HSI オシレータクロックの 6 サイクル後にローになります。

- 0 : HSI オシレータはレディ状態ではありません。
- 1 : HSI オシレータはレディ状態です。

ビット 0 HSION : HSI クロック有効化

ソフトウェアでセット/クリアされます。

STOP または STANDBY モードを終了するとき、または、直接的または間接的にシステムクロックとして使用されている HSE クリスタルオシレータに障害が発生した場合、HSI オシレータを強制的にオンにするために、ハードウェアによってセットされます。このビットは、HSI が直接的または間接的にシステムクロックとして使用されている場合、またはシステムクロックとして選択されている場合は、リセットできません。

- 0 : HSI オシレータオフ
- 1 : HSI オシレータオン

7.4.2 クロック設定レジスタ (RCC_CFGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : $0 \leq \text{ウェイトステート} \leq 2$ 、ワード、ハーフワード、およびバイトアクセス

クロックソースの切り替え中にアクセスが発生した場合に限り、1 または 2ウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLL NODIV	MCOPRE[2:0]			MCO[3:0]				Res.	Res.	PLLMUL[3:0]				PLL XTPRE	PLL SRC[1]
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PLL SMS[0]	ADC PRE	Res.	Res.	Res.	PPRE[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw	rw				rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

ビット 31 **PLLNODIV** : PLL クロックは MCO 用には分周されない (STM32F05x デバイスでは使用不可)
このビットは、ソフトウェアによってセット／クリアされます。MCO への PLL の接続は 2 分周でスイッチオフされます。
0 : MCO 用に PLL を 2 分周
1 : PLL は MCO 用には分周されない

ビット 30:28 **MCOPRE[2:0]** : マイクロコントローラクロック出力プリスケアラ (STM32F05x デバイスでは使用不可)
これらのビットは、MCO プリスケアラの分周比を選択するために、ソフトウェアによってセット／クリアされます。グリッチを回避するために、MCO 出力を無効にした場合にのみ、このプリスケアラを変更することを強く推奨します。
000 : MCO を 1 分周
001 : MCO を 2 分周
010 : MCO を 4 分周
.....
111 : MCO を 128 分周

ビット 27:24 **MCO[3:0]** : マイクロコントローラクロック出力
ソフトウェアでセット／クリアされます。
0000 : MCO 出力無効、MCO にクロックなし
0001 : 内部 RC 14 MHz (HSI14) オシレータクロックの選択
0010 : 内部低速 (LSI) オシレータクロックの選択
0011 : 外部低速 (LSE) オシレータクロックの選択
0100 : システムクロックの選択
0101 : 内部 RC 8 MHz (HSI) オシレータクロックの選択
0110 : 外部 4-32 MHz (HSE) オシレータクロックの選択
0111 : PLL クロックの選択 (PLLNODIV に応じて 1 または 2 で分周)
1000 : 内部 RC 48 MHz (HSI48) オシレータクロックの選択

注 : このクロック出力では、起動時または MCO クロックソースの切り替え時に、切り捨てのサイクルがある場合があります。

ビット 23:22 予約済みであり、リセット値に保持する必要があります。

ビット 21:18 **PLLMUL[3:0]** : PLL 通倍数

これらのビットは、PLL 通倍数を定義するために、ソフトウェアによって書き込まれます。これらのビットに書き込めるのは、PLL が無効なときだけです。

注意 : PLL 出力周波数は 48 MHz を超えてはなりません。

0000 : PLL 入力クロック × 2
0001 : PLL 入力クロック × 3
0010 : PLL 入力クロック × 4
0011 : PLL 入力クロック × 5
0100 : PLL 入力クロック × 6
0101 : PLL 入力クロック × 7
0110 : PLL 入力クロック × 8
0111 : PLL 入力クロック × 9
1000 : PLL 入力クロック × 10
1001 : PLL 入力クロック × 11
1010 : PLL 入力クロック × 12
1011 : PLL 入力クロック × 13
1100 : PLL 入力クロック × 14
1101 : PLL 入力クロック × 15
1110 : PLL 入力クロック × 16
1111 : PLL 入力クロック × 16

ビット 17 **PLLXTPRE** : PLL 入力クロックの HSE 分周

このビットは RCC_CFGR2 の PREDIV[0] ビットと同じビットです。意味については、RCC_CFGR2 の PREDIV ビットの説明を参照してください。

ビット 16:15 **PLLSRC[1:0]** : PLL 入力クロックソース

PLL または PREDIV クロックソースを選択するために、ソフトウェアによってセット/クリアされます。これらのビットに書き込めるのは、PLL が無効なときだけです。

00 : PLL 入力クロックとして HSI/2 を選択 (STM32F04x、STM32F07x、および STM32F09x デバイス上で PREDIV を強制的に 2 分周)
01 : PLL 入力クロックとして HSI/PREDIV を選択
10 : PLL 入力クロックとして HSE/PREDIV を選択
11 : PLL 入力クロックとして HSI48/PREDIV を選択

注 : PLLSRC[0] ビットは STM32F04x、STM32F07x、および STM32F09x デバイスでのみ使用でき、それ以外の場合は (値ゼロで) 予約済みです。

ビット 14 **ADCPRE** : ADC プリスケール

廃止された設定。適切な ADC クロック選択は、ADC_CFGR2 内で実行されます ([セクション 13.12.5 : ADC 設定レジスタ 2 \(ADC_CFGR2\) \(264 ページ\)](#) を参照)。

ビット 13:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **PPRE[2:0]** : PCLK プリスケール

APB クロック (PCLK) の分周比を制御するために、ソフトウェアによってセット/クリアされます。

0xx : HCLK は分周されません。
100 : HCLK は 2 分周されます。
101 : HCLK は 4 分周されます。
110 : HCLK は 8 分周されます。
111 : HCLK は 16 分周されます。

ビット 7:4 HPRE[3:0] : HLCK プリスケアラ

AHB クロックの分周比を制御するために、ソフトウェアによってセット／クリアされます。

- 0xxx : SYSCLK は分周されません。
- 1000 : SYSCLK は 2 分周されます。
- 1001 : SYSCLK は 4 分周されます。
- 1010 : SYSCLK は 8 分周されます。
- 1011 : SYSCLK は 16 分周されます。
- 1100 : SYSCLK は 64 分周されます。
- 1101 : SYSCLK は 128 分周されます。
- 1110 : SYSCLK は 256 分周されます。
- 1111 : SYSCLK は 512 分周されます。

ビット 3:2 SWS[1:0] : システムクロックスイッチステータス

どのクロックソースがシステムクロックとして使用されているかを示すために、ハードウェアによってセット／クリアされます。

- 00 : HSI オシレータがシステムクロックとして使用されています。
- 01 : HSE オシレータがシステムクロックとして使用されています。
- 10 : PLL がシステムクロックとして使用されています。
- 11 : HSI48 オシレータがシステムクロックとして使用されています（使用可能な場合）。

ビット 1:0 SW[1:0] : システムクロックスイッチ

SYSCLK ソースを選択するために、ソフトウェアによってセット／クリアされます。

STOP および STANDBY モードを終了するとき、またはシステムクロックとして直接的または間接的に使用されている HSE オシレータに障害が発生したときに、強制的に HSI を選択するようにハードウェアによってクリアされます。

- 00 : HSI がシステムクロックとして選択されます。
- 01 : HSE がシステムクロックとして選択されます。
- 10 : PLL がシステムクロックとして選択されます。
- 11 : HSI48 がシステムクロックとして選択されます（使用可能な場合）。

7.4.3 クロック割り込みレジスタ (RCC_CIR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	HSI48R DYC	HSI14 RDYC	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC
								w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	HSI48R DYIE	HSI14 RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	HSI48R DYF	HSI14 RDYF	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF
	rw	rw	rw	rw	rw	rw	rw	r	r	r	r	r	r	r	r

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **CSSC** : クロックセキュリティシステム割り込みクリア

このビットは、CSSF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : CSSF フラグをクリアします。

ビット 22 **HSI48RDYC** : HSI48 レディ割り込みクリア

このビットは、HSI48RDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : HSI48RDYF フラグをクリアします。

ビット 21 **HSI14RDYC** : HSI14 レディ割り込みクリア

このビットは、HSI14RDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : HSI14RDYF フラグをクリアします。

ビット 20 **PLLRDYC** : PLL レディ割り込みクリア

このビットは、PLLRDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : PLLRDYF フラグをクリアします。

ビット 19 **HSE RDYC** : HSE レディ割り込みクリア

このビットは、HSE RDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : HSE RDYF フラグをクリアします。

ビット 18 **HSIRDYC** : HSI レディ割り込みクリア

このビットは、HSIRDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : HSIRDYF フラグをクリアします。

ビット 17 **LSERDYC** : LSE レディ割り込みクリア

このビットは、LSERDYF フラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : LSERDYF がクリアされます。

- ビット 16 **LSIRDYC** : LSI レディ割り込みクリア
このビットは、LSIRDYF フラグをクリアするために、ソフトウェアによってセットされます。
0 : 影響なし。
1 : LSIRDYF がクリアされます。
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **HSI48RDYIE** : HSI48 レディ割り込み有効化
HSI48 オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : HSI48 レディ割り込みは無効です。
1 : HSI48 レディ割り込みは有効です。
- ビット 13 **HSI14RDYIE** : HSI14 レディ割り込み有効化
HSI14 オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : HSI14 レディ割り込みは無効です。
1 : HSI14 レディ割り込みは有効です。
- ビット 12 **PLLRDYIE** : PLL レディ割り込み有効化
PLL ロックによって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : PLL ロック割り込みは無効です。
1 : PLL ロック割り込みは有効です。
- ビット 11 **HSE RDYIE** : HSE レディ割り込み有効化
HSE オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : HSE レディ割り込みは無効です。
1 : HSE レディ割り込みは有効です。
- ビット 10 **HSIRDYIE** : HSI レディ割り込み有効化
HSI オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : HSI レディ割り込み無効
1 : HSI レディ割り込み有効
- ビット 9 **LSE RDYIE** : LSE レディ割り込み有効化
LSE オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : LSE レディ割り込みは無効です。
1 : LSE レディ割り込みは有効です。
- ビット 8 **LSIRDYIE** : LSI レディ割り込み有効化
LSI オシレータの安定化によって発生する割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。
0 : LSI レディ割り込みは無効です。
1 : LSI レディ割り込みは有効です。
- ビット 7 **CSSF** : クロックセキュリティシステム割り込みフラグ
HSE オシレータで障害が検出されたときに、ハードウェアによってセットされます。
CSSC ビットをセットすることによって、ソフトウェアによってクリアされます。
0 : HSE クロック障害によるクロックセキュリティ割り込みは発生していません。
1 : HSE クロック障害によってクロックセキュリティ割り込みが発生しました。

ビット 6 HSI48RDYF : HSI48 レディ割り込みフラグ

HSI48 が安定し、クロック制御レジスタ 2 (RCC_CR2) の HSI48ON ビットのセットに応じて HSI48RDYDIE がセットされたときに、ハードウェアによってセットされます。HSI48ON がセットされず、ただしクロックリクエストを通じてペリフェラルによって HSI48 オシレータが有効化された場合、このビットはセットされず、割り込みも生成されません。

HSI48RDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSI48 オシレータによるクロックレディ割り込みは発生していません。

1 : HSI48 オシレータによるクロックレディ割り込みが発生しました。

ビット 5 HSI14RDYF : HSI14 レディ割り込みフラグ

HSI14 が安定し、[クロック制御レジスタ 2 \(RCC_CR2\)](#) の HSI14ON ビットのセットに応じて HSI14RDYDIE がセットされたときに、ハードウェアによってセットされます。HSI14ON がセットされず、ただしクロックリクエストを通じてペリフェラルによって HSI14 オシレータが有効化された場合、このビットはセットされず、割り込みも生成されません。

HSI14RDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSI14 オシレータによるクロックレディ割り込みは発生していません。

1 : HSI14 オシレータによるクロックレディ割り込みが発生しました。

ビット 4 PLLRDYF : PLL レディ割り込みフラグ

PLL がロックされ、PLLRDYDIE がセットされているときに、ハードウェアによってセットされます。

PLLRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : PLL ロックによるクロックレディ割り込みは発生していません。

1 : PLL ロックによるクロックレディ割り込みが発生しました。

ビット 3 HSERDYF : HSE レディ割り込みフラグ

HSE クロックが安定し、HSERDYDIE がセットされているときに、ハードウェアによってセットされます。

HSERDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSE オシレータによるクロックレディ割り込みは発生していません。

1 : HSE オシレータによるクロックレディ割り込みが発生しました。

ビット 2 HSIRDYF : HSI レディ割り込みフラグ

HSI クロックが安定し、HSION ([クロック制御レジスタ \(RCC_CR\)](#) を参照) のセットに応じて HSIRDYDIE がセットされたときに、ハードウェアによってセットされます。HSION がセットされず、ただしクロックリクエストを通じてペリフェラルによって HSI オシレータが有効化された場合、このビットはセットされず、割り込みも生成されません。

HSIRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : HSI オシレータによるクロックレディ割り込みは発生していません。

1 : HSI オシレータによるクロックレディ割り込みが発生しました。

ビット 1 LSERDYF : LSE レディ割り込みフラグ

LSE クロックが安定し、LSERDYDIE がセットされているときに、ハードウェアによってセットされます。

LSERDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : LSE オシレータによるクロックレディ割り込みは発生していません。

1 : LSE オシレータによるクロックレディ割り込みが発生しました。

ビット 0 LSIRDYF : LSI レディ割り込みフラグ

LSI クロックが安定し、LSIRDYDIE がセットされているときに、ハードウェアによってセットされます。

LSIRDYC ビットをセットすることによって、ソフトウェアによってクリアされます。

0 : LSI オシレータによるクロックレディ割り込みは発生していません。

1 : LSI オシレータによるクロックレディ割り込みが発生しました。

7.4.4 APB ペリフェラルリセットレジスタ 2 (RCC_APB2RSTR)

アドレスオフセット : 0x0C

リセット値 : 0x00000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGMCURST	Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST
									rw				rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1RST	Res.	SPI1RST	TIM1RST	Res.	ADCRST	Res.	USART8RST	USART7RST	USART6RST	Res.	Res.	Res.	Res.	SYSCFGRST
	rw		rw	rw		rw		rw	rw	rw					rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DBGMCURST** : デバッグ MCU リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : デバッグ MCU をリセットします。

ビット 21:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM17RST** : TIM17 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM17 タイマをリセットします。

ビット 17 **TIM16RST** : TIM16 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM16 タイマをリセットします。

ビット 16 **TIM15RST** : TIM15 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM15 タイマをリセットします。

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **USART1RST** : USART1 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : USART1 をリセットします。

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **SPI1RST** : SPI1 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : SPI1 をリセットします。

- ビット 11 **TIM1RST** : TIM1 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM1 タイマをリセットします。
- ビット 10 予約済みであり、リセット値に保持する必要があります。
- ビット 9 **ADCRST** : ADC インタフェースリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : ADC インタフェースをリセットします。
- ビット 8 予約済みであり、リセット値に保持する必要があります。
- ビット 7 **USART8RST** : USART8 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : USART8 をリセットします。
- ビット 6 **USART7RST** : USART7 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : USART7 をリセットします。
- ビット 5 **USART6RST** : USART6 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : USART6 をリセットします。
- ビット 4:1 予約済みであり、リセット値に保持する必要があります。
- ビット 0 **SYSCFGRST** : SYSCFG リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : SYSCFG をリセットします。

7.4.5 APB ペリフェラルリセットレジスタ 1 (RCC_APB1RSTR)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000
アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CEC RST	DAC RST	PWR RST	CRS RST	Res.	CAN RST	Res.	USB RST	I2C2 RST	I2C1 RST	USART5 RST	USART4 RST	USART3 RST	USART2 RST	Res.
	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 RST	Res.	Res.	WWDG RST	Res.	Res.	TIM14R ST	Res.	Res.	TIM7 RST	TIM6 RST	Res.	Res.	TIM3 RST	TIM2 RST
	rw			rw			rw			rw	rw			rw	rw



ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **CECRST** : HDMI CEC リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : HDMI CEC をリセットします。

ビット 29 **DACRST** : DAC インタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : DAC インタフェースをリセットします。

ビット 28 **PWRRST** : 電源インタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : 電源インタフェースをリセットします。

ビット 27 **CRSRST** : クロックリカバリシステムインタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : CRS インタフェースをリセットします。

ビット 26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **CANRST** : CAN インタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : CAN インタフェースをリセットします。

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **USBRST** : USB インタフェースリセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : USB インタフェースをリセットします。

ビット 22 **I2C2RST** : I2C2 リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I2C2 をリセットします。

ビット 21 **I2C1RST** : I2C1 リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : I2C1 をリセットします。

ビット 20 **USART5RST** : USART5 リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : USART4 をリセットします。

ビット 19 **USART4RST** : USART4 リセット

ソフトウェアでセット/クリアされます。

0 : 影響なし。

1 : USART4 をリセットします。

ビット 18 **USART3RST** : USART3 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : USART3 をリセットします。

ビット 17 **USART2RST** : USART2 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : USART2 をリセットします。

ビット 16:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **SPI2RST** : SPI2 リセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : SPI2 をリセットします。

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **WWDGRST** : ウィンドウ型ウォッチドッグリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : ウィンドウ型ウォッチドッグをリセットします。

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TIM14RST** : TIM14 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM14 をリセットします。

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **TIM7RST** : TIM7 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM7 をリセットします。

ビット 4 **TIM6RST** : TIM6 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM6 をリセットします。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TIM3RST** : TIM3 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM3 をリセットします。

ビット 0 **TIM2RST** : TIM2 タイマリセット
ソフトウェアでセット／クリアされます。
0 : 影響なし。
1 : TIM2 をリセットします。

7.4.6 AHB ペリフェラルクロック有効レジスタ (RCC_AHBENR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0014

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

注 : ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの値はソフトウェアによって読み出しできず、返される値は常に 0x0 になります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN	Res.	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	Res.
							rw		rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC EN	Res.	FLITF EN	Res.	SRAM EN	DMA2E N	DMA EN
									rw		rw		rw	rw	rw

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TSCEN** : タッチセンシングコントローラクロック有効化
ソフトウェアでセット/クリアされます。
0 : TSC クロック無効
1 : TSC クロック有効

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **IOPFEN** : I/O ポート F クロック有効化
ソフトウェアでセット/クリアされます。
0 : I/O ポート F クロック無効
1 : I/O ポート F クロック有効

ビット 21 **IOPEEN** : I/O ポート E クロック有効化
ソフトウェアでセット/クリアされます。
0 : I/O ポート E クロック無効
1 : I/O ポート E クロック有効

ビット 20 **IOPDEN** : I/O ポート D クロック有効化
ソフトウェアでセット/クリアされます。
0 : I/O ポート D クロック無効
1 : I/O ポート D クロック有効

ビット 19 **IOPCEN** : I/O ポート C クロック有効化
ソフトウェアでセット/クリアされます。
0 : I/O ポート C クロック無効
1 : I/O ポート C クロック有効

ビット 18 **IOPBEN** : I/O ポート B クロック有効化
ソフトウェアでセット/クリアされます。
0 : I/O ポート B クロック無効
1 : I/O ポート B クロック有効



ビット 17 **IOPAEN** : I/O ポート A クロック有効化
ソフトウェアでセット/クリアされます。

0 : I/O ポート A クロック無効
1 : I/O ポート A クロック有効

ビット 16:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **CRCEN** : CRC クロック有効化
ソフトウェアでセット/クリアされます。

0 : CRC クロック無効
1 : CRC クロック有効

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **FLITFEN** : FLITF クロック有効化
SLEEP モード時に FLITF クロックを無効/有効にするために、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 FLITF クロック無効
1 : SLEEP モード時 FLITF クロック有効

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **SRAMEN** : SRAM インタフェースクロック有効
SLEEP モード時に SRAM インタフェースクロックを無効/有効にするために、ソフトウェアによってセット/クリアされます。

0 : SLEEP モード時 SRAM インタフェースクロック無効
1 : SLEEP モード時 SRAM インタフェースクロック有効

ビット 1 **DMA2EN** : DMA2 クロック有効化
ソフトウェアでセット/クリアされます。

0 : DMA2 クロック無効
1 : DMA2 クロック有効

ビット 0 **DMAEN** : DMA クロック有効化
ソフトウェアでセット/クリアされます。

0 : DMA クロック無効
1 : DMA クロック有効

7.4.7 APB ペリフェラルクロック有効レジスタ 2 (RCC_APB2ENR)

アドレス : 0x18

リセット値 : 0x0000 0000

アクセス : ワード、ハーフワード、およびバイトアクセス

APB ドメイン内のペリフェラルへのアクセス中にアクセスが発生する場合を除き、ノーウェイトステート。この場合、APB ペリフェラルへのアクセスが終了するまでウェイトステートが挿入されます。

注 : **ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの値はソフトウェアによって読み出しできず、返される値は常に 0x0 になります。**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG MCUEN	Res.	Res.	Res.	TIM17 EN	TIM16 EN	TIM15EN
									rw				rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	USART1 EN	Res.	SPI1EN	TIM1EN	Res.	ADCEN	Res.	USART8E N	USART7E N	USART6E N	Res.	Res.	Res.	Res.	SYSCFG COMPEN
	rw		rw	rw		rw		rw	rw	rw					rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **DBGMCUEN** MCU デバッグモジュールクロック有効化

ソフトウェアでセット／クリアされます。

0 : MCU デバッグモジュールクロック無効

1 : MCU デバッグモジュール有効

ビット 21:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **TIM17EN** : TIM17 タイマクロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM17 タイマクロック無効

1 : TIM17 タイマクロック有効

ビット 17 **TIM16EN** : TIM16 タイマクロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM16 タイマクロック無効

1 : TIM16 タイマクロック有効

ビット 16 **TIM15EN** : TIM15 タイマクロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM15 タイマクロック無効

1 : TIM15 タイマクロック有効

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **USART1EN** : USART1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : USART1 クロック無効

1 : USART1 クロック有効

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **SPI1EN** : SPI1 クロック有効化

ソフトウェアでセット／クリアされます。

0 : SPI1 クロック無効

1 : SPI1 クロック有効

ビット 11 **TIM1EN** : TIM1 タイマクロック有効化

ソフトウェアでセット／クリアされます。

0 : TIM1 タイマクロック無効

1 : TIM1P タイマクロック有効

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **ADCEN** : ADC インタフェースクロック有効
ソフトウェアでセット/クリアされます。
0 : ADC インタフェースクロック無効
1 : ADC インタフェースクロック有効

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **USART8EN** : USART8 クロック有効化
ソフトウェアでセット/クリアされます。
0 : USART8 クロック無効
1 : USART8 クロック有効

ビット 6 **USART7EN** : USART7 クロック有効化
ソフトウェアでセット/クリアされます。
0 : USART7 クロック無効
1 : USART7 クロック有効

ビット 5 **USART6EN** : USART6 クロック有効化
ソフトウェアでセット/クリアされます。
0 : USART6 クロック無効
1 : USART6 クロック有効

ビット 4:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **SYSCFGCOMPEN** : SYSCFG & COMP クロック有効化
ソフトウェアでセット/クリアされます。
0 : SYSCFG & COMP クロック無効
1 : SYSCFG & COMP クロック有効

7.4.8 APB ペリフェラルクロック有効レジスタ 1 (RCC_APB1ENR)

アドレス : 0x1C

リセット値 : 0x0000 0000

アクセス : ワード、ハーフワード、およびバイトアクセス

APB ドメイン内のペリフェラルへのアクセス中にアクセスが発生する場合を除き、ノーウェイトステート。この場合、APB ペリフェラルへのこのアクセスが終了するまでウェイトステートが挿入されます。

注 : ペリフェラルクロックがアクティブでない場合、ペリフェラルレジスタの値はソフトウェアによって読み出しできず、返される値は常に 0x0 になります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CEC EN	DAC EN	PWR EN	CRS EN	Res.	CAN EN	Res.	USB EN	I2C2 EN	I2C1 EN	USART5E N	USART4 EN	USART3E N	USART2 EN	Res.
	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI2 EN	Res.	Res.	WWDG EN	Res.	Res.	TIM14 EN	Res.	Res.	TIM7 EN	TIM6 EN	Res.	Res.	TIM3 EN	TIM2 EN
	rw			rw			rw			rw	rw			rw	rw



ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **CECEN** : HDMI CEC クロック有効化
ソフトウェアでセット/クリアされます。

0 : HDMI CEC クロック無効

1 : HDMI CEC クロック有効

ビット 29 **DACEN** : DAC インタフェースクロック有効
ソフトウェアでセット/クリアされます。

0 : DAC インタフェースクロック無効

1 : DAC インタフェースクロック有効

ビット 28 **PWREN** : 電源インタフェースクロック有効
ソフトウェアでセット/クリアされます。

0 : 電源インタフェースクロック無効

1 : 電源インタフェースクロック有効

ビット 27 **CRSEN** : クロックリカバリシステムインタフェースクロック有効
ソフトウェアでセット/クリアされます。

0 : CRS インタフェースクロック無効

1 : CRS インタフェースクロック有効

ビット 26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **CANEN** : CAN インタフェースクロック有効
ソフトウェアでセット/クリアされます。

0 : CAN インタフェースクロック無効

1 : CAN インタフェースクロック有効

ビット 24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **USBEN** : USB インタフェースクロック有効
ソフトウェアでセット/クリアされます。

0 : USB インタフェースクロック無効

1 : USB インタフェースクロック有効

ビット 22 **I2C2EN** : I2C2 クロック有効化
ソフトウェアでセット/クリアされます。

0 : I2C2 クロック無効

1 : I2C2 クロック有効

ビット 21 **I2C1EN** : I2C1 クロック有効化
ソフトウェアでセット/クリアされます。

0 : I2C1 クロック無効

1 : I2C1 クロック有効

ビット 20 **USART5EN** : USART5 クロック有効化
ソフトウェアでセット/クリアされます。

0 : USART5 クロック無効

1 : USART5 クロック有効

ビット 19 **USART4EN** : USART4 クロック有効化
ソフトウェアでセット/クリアされます。

0 : USART4 クロック無効

1 : USART4 クロック有効

ビット 18 **USART3EN** : USART3 クロック有効化
ソフトウェアでセット/クリアされます。
0 : USART3 クロック無効
1 : USART3 クロック有効

ビット 17 **USART2EN** : USART2 クロック有効化
ソフトウェアでセット/クリアされます。
0 : USART2 クロック無効
1 : USART2 クロック有効

ビット 16:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **SPI2EN** : SPI2 クロック有効化
ソフトウェアでセット/クリアされます。
0 : SPI2 クロック無効
1 : SPI2 クロック有効

ビット 13:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **WWDGEN** : ウィンドウ型ウォッチドッグクロック有効化
ソフトウェアでセット/クリアされます。
0 : ウィンドウ型ウォッチドッグクロック無効
1 : ウィンドウ型ウォッチドッグクロック有効

ビット 10:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TIM14EN** : TIM14 タイマクロック有効化
ソフトウェアでセット/クリアされます。
0 : TIM14 クロック無効
1 : TIM14 クロック有効

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **TIM7EN** : TIM7 タイマクロック有効化
ソフトウェアでセット/クリアされます。
0 : TIM7 クロック無効
1 : TIM7 クロック有効

ビット 4 **TIM6EN** : TIM6 タイマクロック有効化
ソフトウェアでセット/クリアされます。
0 : TIM6 クロック無効
1 : TIM6 クロック有効

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **TIM3EN** : TIM3 タイマクロック有効化
ソフトウェアでセット/クリアされます。
0 : TIM3 クロック無効
1 : TIM3 クロック有効

ビット 0 **TIM2EN** : TIM2 タイマクロック有効化
ソフトウェアでセット/クリアされます。
0 : TIM2 クロック無効
1 : TIM2 クロック有効

7.4.9 RTC ドメイン制御レジスタ (RCC_BDCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0018、RTC ドメインリセットによってリセットされます。

アクセス : 0 ≤ ウェイトステート ≤ 3、ワード、ハーフワード、およびバイトアクセス

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

注 : **RTC ドメイン制御レジスタ (RCC_BDCR) の LSEON、LSEBYP、RTCSEL、および RTCEN ビットは、RTC ドメインにあります。このため、リセット後、これらのビットは書き込み保護されるので、これらを変更するには、電源制御レジスタ (PWR_CR) の DBP ビットをセットする必要があります。詳細については、セクション 6.1.3 : バッテリバックアップドメインを参照してください。これらのビットは、RTC ドメインリセット後にのみリセットされます (セクション 7.1.3 : RTC ドメインリセットを参照)。内部または外部リセットは、これらのビットに影響しません。**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	Res.	Res.	LSEDRV[1:0]		LSE BYP	LSE RDY	LSEON
rw						rw	rw				rw	rw	rw	r	rw

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **BDRST** : RTC ドメインソフトウェアリセット

ソフトウェアでセット/クリアされます。

0 : リセットは有効ではありません。

1 : RTC ドメイン全体をリセットします。

ビット 15 **RTCEN** : RTC クロック有効化

ソフトウェアでセット/クリアされます。

0 : RTC クロック無効

1 : RTC クロック有効

ビット 14:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **RTCSEL[1:0]** : RTC クロックソース選択

RTC のクロックソースを選択するために、ソフトウェアによってセットされます。RTC クロックソースが選択されると、RTC ドメインがリセットされない限り、変更はできません。BDRST ビットを使用して、リセットできます。

00 : クロックなし。

01 : LSE オシレータクロックが RTC クロックとして使用されます。

10 : LSI オシレータクロックが RTC クロックとして使用されます。

11 : 32 分周された HSE オシレータクロックが RTC クロックとして使用されます。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。



ビット 4:3 LSEDRV LSE オシレータの駆動能力

LSE オシレータの駆動能力を変調するためにソフトウェアによってセットおよびリセットされます。RTC ドメインのリセットはデフォルト値を復元します。

- 00 : 「Xtal モード」 低駆動
- 01 : 「Xtal モード」 中低駆動
- 10 : 「Xtal モード」 中高駆動
- 11 : 「Xtal モード」 高駆動 (リセット値)

注： オシレータは、バイパスモードでない場合、Xtal モードです。

ビット 2 LSEBYP : LSE オシレータバイパス

デバッグモードのオシレータをバイパスするために、ソフトウェアによってセット/クリアされます。このビットに書き込めるのは、外部 32 kHz オシレータが無効なときだけです。

- 0 : LSE オシレータはバイパスされません。
- 1 : LSE オシレータはバイパスされます。

ビット 1 LSERDY : LSE オシレータはレディ状態です。

外部 32 kHz オシレータが安定していることを示すために、ハードウェアによってセット/クリアされます。LSEON ビットがクリアされた後、LSERDY は、外部低速オシレータクロックの 6 サイクル後にローになります。

- 0 : LSE オシレータはレディ状態ではありません。
- 1 : LSE オシレータはレディ状態です。

ビット 0 LSEON : LSE オシレータイネーブル

ソフトウェアでセット/クリアされます。

- 0 : LSE オシレータオフ
- 1 : LSE オシレータオン

7.4.10 制御／ステータスレジスタ (RCC_CSR)

アドレス : 0x24

リセット値 : 0xFFFF 0000。システムリセットによってリセットされます。ただし、リセットフラグは電源リセット時のみリセットされます。

アクセス : $0 \leq \text{ウェイトステート} \leq 3$ 、ワード、ハーフワード、およびバイトアクセス

このレジスタへの連続したアクセスの場合、ウェイトステートが挿入されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	OBLRS TF	RMVF	V18PWR RSTF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	rl_w	r							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSI RDY	LSION
														r	rw

ビット 31 LPWRRSTF : 低電力リセットフラグ

低電力管理リセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。

- 0 : 低電力管理リセットは発生していません。
- 1 : 低電力管理リセットが発生しました。

低電力管理リセットの詳細については、[低電力管理リセット](#)を参照してください。

ビット 30 WWDGRSTF : ウィンドウ型ウォッチドッグリセットフラグ

ウィンドウ型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。

- 0 : ウィンドウ型ウォッチドッグリセットは発生していません。
- 1 : ウィンドウ型ウォッチドッグリセットが発生しました。

ビット 29 IWDGRSTF : 独立型ウォッチドッグリセットフラグ

V_{DD} ドメインから独立型ウォッチドッグリセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。

- 0 : ウォッチドッグリセットは発生していません。
- 1 : ウォッチドッグリセットが発生しました。

ビット 28 SFTIRSTF : ソフトウェアリセットフラグ

ソフトウェアリセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。

- 0 : ソフトウェアリセットは発生していません。
- 1 : ソフトウェアリセットが発生しました。

ビット 27 PORRSTF : POR/PDR リセットフラグ

POR/PDR リセットが発生したときに、ハードウェアによってセットされます。
RMVF ビットに書き込むことによってクリアされます。

- 0 : POR/PDR リセットは発生していません。
- 1 : POR/PDR リセットが発生しました。



ビット 26 **PINRSTF** : PIN リセットフラグ

NRST ピンに外部リセット信号が入力され、リセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : NRST ピンからのリセットは発生していません。

1 : NRST ピンからリセットが発生しました。

ビット 25 **OBLRSTF** : オプションバイトローダリセットフラグ

OBL からリセットが発生したときに、ハードウェアによってセットされます。

RMVF ビットに書き込むことによってクリアされます。

0 : OBL からのリセットは発生していません。

1 : OBL からリセットが発生しました。

ビット 24 **RMVF** : リセットフラグ解除

RMVF を含むリセットフラグをクリアするために、ソフトウェアによってセットされます。

0 : 影響なし。

1 : リセットフラグをクリアします。

ビット 23 **V18PWRSTF** : 1.8 V ドメインのフラグをリセットします。

1.8 V ドメインの POR/PDR が発生したときに、ハードウェアによってクリアされます。

RMVF ビットに書き込むことによってクリアされます。

0 : 1.8 V ドメインの POR/PDR リセットは発生していません。

1 : 1.8 V ドメインの POR/PDR リセットが発生しました。

注意 : **STM32F0x8 ファミリーでは、このフラグは予約済みとして読み出す必要があります。**

ビット 22:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **LSIRDY** : LSI オシレータはレディ状態です。

LSI オシレータが安定していることを示すために、ハードウェアによってセット／クリアされます。LSION ビットがクリアされた後、LSIRDY は、LSI オシレータクロックの 3 サイクル後にローになります。

0 : LSI オシレータはレディ状態ではありません。

1 : LSI オシレータはレディ状態です。

ビット 0 **LSION** : LSI オシレータイネーブル

ソフトウェアでセット／クリアされます。

0 : LSI オシレータオフ

1 : LSI オシレータオン

7.4.11 AHB ペリフェラルリセットレジスタ（RCC_AHBRSTR）

アドレス : 0x28

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSC RST	Res.	IOPF RST	IOPE RST	IOPD RST	IOPC RST	IOPB RST	IOPA RST	Res.
							rw		rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.



ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **TSCRST** : タッチセンシングコントローラリセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : TSC をリセットします。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **IOPFRST** : I/O ポート F リセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : I/O ポート F をリセットします。

ビット 21 **IOPERST** : I/O ポート E リセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : I/O ポート E をリセットします。

ビット 20 **IOPDRST** : I/O ポート D リセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : I/O ポート D をリセットします。

ビット 19 **IOPCRST** : I/O ポート C リセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : I/O ポート C をリセットします。

ビット 18 **IOPBRST** : I/O ポート B リセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : I/O ポート B をリセットします。

ビット 17 **IOPARST** : I/O ポート A リセット
ソフトウェアでセット／クリアされます。

0 : 影響なし。
1 : I/O ポート A をリセットします。

ビット 16:0 予約済みであり、リセット値に保持する必要があります。

7.4.12 クロック設定レジスタ 2 (RCC_CFGR2)

アドレス : 0x2C

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV[3:0]			
													r/w	r/w	r/w

ビット 31:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **PREDIV[3:0]** PREDIV の分周比

これらのビットは、PREDIV の分周比を選択するために、ソフトウェアによってセット／クリアされます。これらに書き込めるのは、PLL が無効なときだけです。

注： **ビット 0 はクロック設定レジスタ (RCC_CFGR) のビット 17 と同じビットです。そのため、クロック設定レジスタ (RCC_CFGR) のビット 17 を変更すると、クロック設定レジスタ 2 (RCC_CFGR2) のビット 0 も変更されます (他の STM32 製品との互換性のため)。**

- 0000 : PREDIV 入力クロックは分周されません。
- 0001 : PREDIV クロックが 2 分周されます。
- 0010 : PREDIV クロックが 3 分周されます。
- 0011 : PREDIV クロックが 4 分周されます。
- 0100 : PREDIV クロックが 5 分周されます。
- 0101 : PREDIV クロックが 6 分周されます。
- 0110 : PREDIV クロックが 7 分周されます。
- 0111 : PREDIV クロックが 8 分周されます。
- 1000 : PREDIV クロックが 9 分周されます。
- 1001 : PREDIV クロックが 10 分周されます。
- 1010 : PREDIV クロックが 11 分周されます。
- 1011 : PREDIV クロックが 12 分周されます。
- 1100 : PREDIV クロックが 13 分周されます。
- 1101 : PREDIV クロックが 14 分周されます。
- 1110 : PREDIV クロックが 15 分周されます。
- 1111 : PREDIV クロックが 16 分周されます。



7.4.13 クロック設定レジスタ 3 (RCC_CFGR3)

アドレス : 0x30

リセット値 : 0x0000 0000

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART3SW[1:0]		USART2SW[1:0]	
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC SW	USB SW	CEC SW	Res.	I2C1 SW	Res.	Res.	USART1SW[1:0]	
							rw	rw	rw		rw			rw	rw

ビット 31:20 予約済みであり、リセット値に保持する必要があります。

ビット 19:18 **USART3SW[1:0]** : USART3 クロックソース選択 (STM32F09x デバイスでのみ使用可能)

このビットは、USART3 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00 : PCLK が USART3 クロックソースとして選択されます (デフォルト)。
- 01 : システムクロック (SYSCLK) が USART3 クロックとして選択されます。
- 10 : LSE クロックが USART3 クロックとして選択されます。
- 11 : HSI クロックが USART3 クロックとして選択されます。

ビット 17:16 **USART2SW[1:0]** : USART2 クロックソース選択 (STM32F07x および STM32F09x デバイスでのみ使用可能)

このビットは、USART2 クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 00 : PCLK が USART2 クロックソースとして選択されます (デフォルト)。
- 01 : システムクロック (SYSCLK) が USART2 クロックとして選択されます。
- 10 : LSE クロックが USART2 クロックとして選択されます。
- 11 : HSI クロックが USART2 クロックとして選択されます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **ADCSW** : ADC クロックソース選択

廃止された設定。HSI14 クロックを ADC 非同期クロック入力に接続することで、リセット値で保持します。適切な ADC クロック選択は、ADC_CFGR2 内で実行されます ([セクション 13.12.5 : ADC 設定レジスタ 2 \(ADC_CFGR2\) \(264 ページ\)](#) を参照)。

ビット 7 **USBSW** : USB クロックソース選択

このビットは、USB クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : HSI48 クロックが USB クロックソースとして選択されます (デフォルト)。
- 1 : PLL クロック (PLLCLK) が USB クロックとして選択されます。

ビット 6 **CECSW** : HDMI CEC クロックソース選択

このビットは、CEC クロックソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : 244 分周された HSI クロックが CEC クロックとして選択されます (デフォルト)。
- 1 : LSE クロックが CEC クロックとして選択されます。

ビット 5 予約済みであり、リセット値に保持する必要があります。



ビット 4 **I2C1SW** : I2C1 クロックソースの選択

このビットは、I2C1 クロックソースを選択するために、ソフトウェアによってセット／クリアされます。

0 : HSIクロック が I2C1 クロックソースとして選択されます（デフォルト）。

1 : システムクロック（SYSCLK）が I2C1 クロックとして選択されます。

ビット 3:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **USART1SW[1:0]** : USART1 クロックソースの選択

このビットは、USART1 クロックソースを選択するために、ソフトウェアによってセット／クリアされます。

00 : PCLK が USART1 クロックソースとして選択されます（デフォルト）。

01 : システムクロック（SYSCLK）が USART1 クロックとして選択されます。

10 : LSE クロックが USART1 クロックとして選択されます。

11 : HSI クロックが USART1 クロックとして選択されます。

7.4.14 クロック制御レジスタ 2（RCC_CR2）

アドレス : 0x34

リセット値 : 0xXX00 XX80（X は未定義）。

アクセス : ノーウェイトステート、ワード、ハーフワード、およびバイトアクセス

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSI48CAL[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	HSI48 RDY	HSI48 ON
r	r	r	r	r	r	r	r							r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSI14CAL[7:0]								HSI14TRIM[4:0]					HSI14 DIS	HSI14 RDY	HSI14 ON
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw	r	rw

ビット 31:24 **HSI48CAL[7:0]** : HSI48 工場クロック較正

これらのビットは、スタートアップ時に自動的に初期化され、読み出し専用になります。

ビット 23:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **HSI48RDY** : HSI48 クロックレディフラグ

HSI48 オシレータが安定していることを示すために、ハードウェアによってセットされます。HSI48ON ビットがクリアされた後、HSI48RDY は、HSI48 オシレータクロックの 6 サイクル後にローになります。

0 : HSI48 オシレータはレディ状態ではありません。

1 : HSI48 オシレータはレディ状態です。

ビット 16 **HSI48ON** : HSI48 クロック有効化

ソフトウェアまたはハードウェアによってクリアされます。USB ペリフェラルが有効化され、このソースでスイッチオンしたときに、ハードウェアによってセットされます。STOP または STANDBY モードに移行するときに、オシレータを停止するためにハードウェアによってリセットされます。このビットは、HSI48 が直接的または間接的にシステムクロックとして使用されている場合、またはシステムクロックとして選択されている場合は、リセットできません。

0 : HSI48 オシレータオフ

1 : HSI48 オシレータオン

ビット 15:8 **HSI14CAL[7:0]** : HSI14 クロック較正

これらのビットは、スタートアップ時に自動的に初期化されます。

ビット 7:3 HSI14TRIM[4:0] : HSI14 クロックトリミング

これらのビットにより、ユーザによるプログラムが可能なトリミング値が使用でき、この値は HSI14CAL[7:0] ビットに加算されます。HSI14 の周波数に影響する電圧や温度の変化に対応できるようにプログラミングできます。

デフォルト値は 16 で、HSI14CAL 値に追加すると、HSI14 を $14\text{ MHz} \pm 1\%$ にトリミングします。トリミングステップは、2 つの連続する HSI14CAL ステップの間で、50 kHz 前後です。

ビット 2 HSI14DIS ADC 無効化からの HSI14 クロックリクエスト

ソフトウェアでセット/クリアされます。

このビットをセットすると、ADC インタフェースで HSI14 オシレータが有効化されるのを防ぎます。

0 : ADC インタフェースでは、HSI14 オシレータをオンにすることができます。

1 : ADC インタフェースでは、HSI14 オシレータをオンにすることはできません。

ビット 1 HSI14RDY : HSI14 クロックレディフラグ

HSI14 オシレータが安定していることを示すために、ハードウェアによってセットされます。HSI14ON ビットがクリアされた後、HSI14RDY は、HSI14 オシレータクロックの 6 サイクル後にローになります。

0 : HSI14 オシレータはレディ状態ではありません。

1 : HSI14 オシレータはレディ状態です。

ビット 0 HSI14ON : HSI14 クロック有効化

ソフトウェアでセット/クリアされます。

0 : HSI14 オシレータオフ

1 : HSI14 オシレータオン

7.4.15 RCC レジスタマップ

次の表に、RCC レジスタマップとリセット値を示します。

表 20. RCC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	RCC_CR	Res.	Res.	Res.	Res.	Res.	Res.	PLL RDY	PLL ON	Res.	Res.	Res.	Res.	CSSON	HSEBYP	HSERDY	HSEON	HSICAL[7:0]							HSITRIM[4:0]				Res.	HSIRDY	HSION			
	リセット値							0	0					0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0		1	1	
0x04	RCC_CFGR	PLLMODIV	MCOPRE [2:0]		MCO [3:0]					Res.	Res.	PLLMUL[3:0]			PLLSRC[1]	PLLSRC[0]	ADCPRE	PPRE [2:0]				HPRE[3:0]			SWS [1:0]		SW [1:0]							
	リセット値	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x08	RCC_CIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSSC	HSI48RDY	HSI14 RDY	PLL RDY	HSERDY	HSIRDY	LSERDY	LSIRDY	Res.	Res.	HSI48RDYIE	HSI14 RDYIE	PLL RDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	HSI48RDYF	HSI14 RDYF	PLL RDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	RCC_APB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGMCURST	Res.	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	Res.	Res.	USART1RST	Res.	SPI1RST	TIM1RST	Res.	ADCRST	USART8RST	USART7RST	USART6RST	Res.	Res.	Res.	Res.	Res.	
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	RCC_APB1RSTR	Res.	CECRST	DACRST	PWRRST	CRSRST	Res.	CANRST	Res.	USBRST	I2C2RST	I2C1RST	USART5RST	USART4RST	USART3RST	USART2RST	Res.	Res.	SPI2RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0	0	0	0	0		0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	RCC_AHBENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSCEN	Res.	IOFEN	IOPEEN	USART4RST	IOPCEN	IOPBEN	IOPAEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値								0		0	0	0	0	0	0											0							
0x18	RCC_APB2ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGMCUEN	Res.	Res.	Res.	TIM17 EN	TIM16 EN	TIM15 EN	Res.	Res.	USART1EN	Res.	SPI1EN	TIM1EN	Res.	ADCEN	Res.	USART8EN	USART7EN	USART6EN	Res.	Res.	Res.	Res.	
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		1	0	0
0x1C	RCC_APB1ENR	Res.	CECEN	DACEN	PWREN	CRSEN	Res.	CANEN	Res.	USBEN	I2C2EN	I2C1EN	USART5EN	USART4EN	USART3EN	USART2EN	Res.	Res.	SPI2EN	Res.	Res.	Res.	Res.	Res.	Res.	TIM14EN	Res.	Res.	TIM7EN	TIM6EN	Res.	Res.	Res.	Res.
	リセット値	0	0	0	0	0		0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST	RTCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSE DRV [1:0]	LSEBYP	LSIRDY	LSION
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	RCC_CSR	LPWRSTF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	X	X	X	X	X	X	X	X																							0	0	
0x28	RCC_AHBRSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	RCC_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV[3:0]		
	リセット値																														0	0	0	0

表 20. RCC レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	RCC_CFGR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART3SW[1:0]		USART2SW[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADCSW	USBSW	CECSW	Res.	I2C1SW	Res.	Res.	USART1SW[1:0]	
	リセット値													0	0	0									0	0	0		0		0	0	
0x34	RCC_CR2	HSI48CAL[7:0]									Res.	Res.	Res.	Res.	Res.	HSI48RDY	HSI48ON	HSI14CAL[7:0]								HSI14TRIM[14:0]				HSI14DIS	HSI14RDY	HSI14ON	
	リセット値	X	X	X	X	X	X	X	X	X						0	0	X	X	X	X	X	X	X	X	1	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

8 クロックリカバリシステム (CRS)

このセクションは、STM32F04x、STM32F07x、および STM32F09x デバイスにのみ適用されます。

8.1 概要

クロックリカバリシステム (CRS) は、高精度トリミング可能な内蔵 RC オシレータ HSI48 に作用する、高度なデジタルコントローラです。選択可能な同期信号との比較に基づいて、オシレータ出力周波数チェックを行うための強力な手段を提供しています。CRS では、測定した周波数の誤差値に基づいてオシレータのトリミングを自動調整することが可能です。また、手動でトリミングすることも可能です。

CRS は、USB ペリフェラルに高精度なクロックを提供する場合に最適です。この場合、USB ホストによって正確に 1 ms 間隔で送信される USB バス上のフレーム開始 (SOF) パケット信号から同期信号を取得できます。

同期信号は、LSE オシレータ出力または外部ピンからも取得することができ、またユーザソフトウェアから生成することもできます。

8.2 CRS の主な機能

- 次のプログラム可能なプリスケールと極性を持つ、選択可能な同期ソース：
 - － 外部ピン
 - － LSE オシレータ出力
 - － USB SOF パケット受信
- ソフトウェアによって同期パルスを生成する可能性
- CPU 不要の自動オシレータトリミング機能
- より迅速なスタートアップのための手動制御オプション
- 自動誤差値のキャプチャと再ロードが可能な、16 ビットの周波数誤差カウンタ
- 自動周波数誤差値の評価とステータスレポートの、プログラム可能な制限値
- マスク可能な割り込み/イベント：
 - － 期待される同期 (ESYNC)
 - － 同期 OK (SYNCOK)
 - － 同期警告 (SYNCWARN)
 - － 同期またはトリミングエラー (ERR)

8.3.1 CRS ブロック図

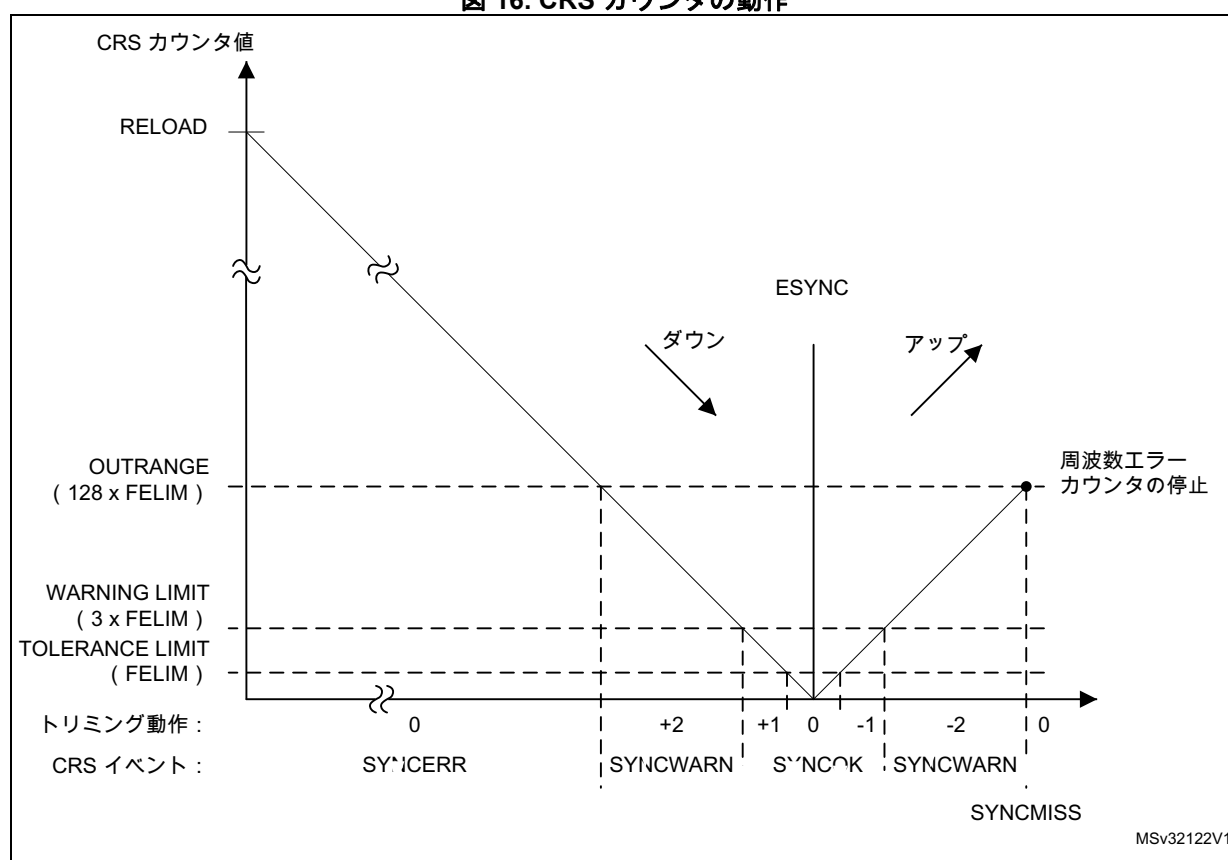
CRS_CR レジスタの SWSYNC ビットをセットして、ソフトウェアで同期イベントを生成することもできます。

8.3.3 周波数誤差測定

この周波数誤差カウンタは、各 SYNC イベントで RELOAD 値によって再ロードされる 16 ビットのダウン／アップカウンタです。ESYNC（期待される同期）イベントが生成されるゼロ値に達するまでカウントダウンします。次に、OUTRANGE 制限によって最終的に停止する（SYNC イベントが受信されない場合）までカウントアップし、SYNCMISS イベントを生成します。OUTRANGE 制限は、周波数誤差制限 (CRS_CFGR レジスタの FELIM フィールド) に 128 を乗算した値として定義されます。

SYNC イベントが検出されると、周波数誤差カウンタの実際の値とそのカウント方向が FECAP（周波数誤差キャプチャ）フィールドと CRS_ISR レジスタの FEDIR（周波数誤差方向）ビットに保管されます。ダウンカウントフェーズで（ゼロ値に達する前に）SYNC イベントが検出されたときは、実際の周波数がターゲットよりも小さいことを意味します（そのため、TRIM 値をインクリメントする必要があります）。アップカウントフェーズ中に検出されると、実際の周波数は高くなります（そのため、TRIM 値をデクリメントする必要があります）。

図 16. CRS カウンタの動作



8.3.4 周波数誤差評価および自動トリミング

測定した周波数誤差は、その値といくつかの制限を比較して評価されます。

- CRS_CFGR レジスタの FELIM フィールドに直接与えられる TOLERANCE LIMIT
- $3 * \text{FELIM}$ 値として定義される WARNING LIMIT
- $128 * \text{FELIM}$ 値として定義される OUTRANGE (エラー制限)

この比較の結果は、ステータスの表示を生成し、CRS_CR レジスタの AUTOTRIMEN ビットをセットすることで有効化される自動トリミングを制御するために使用されます。

- 周波数誤差が許容制限を下回る場合は、TRIM フィールドの実際のトリミング値が適切であり、トリミング動作が不要であることを意味します。
 - SYNCOK ステータスが示されます。
 - TRIM 値は、AUTOTRIM モードでは変更されません。
- 周波数誤差が警告制限を下回り、許容制限を上回るか同じである場合は、なんらかのトリミング動作が必要であることを意味します。ただし、1 回のトリミングステップによる調整で、十分適切な TRIM 値に達することができます。
 - SYNCOK ステータスが示されます。
 - AUTOTRIM モードで、1 回のトリミングステップによって TRIM 値が調整されます。
- 周波数誤差が警告制限を上回るか同じであり、エラー制限を下回る場合は、より強力なトリミング動作が必要であることを意味します。次の周期までに適切な TRIM 値に達することができない危険性もあります。
 - SYNCWARN ステータスが示されます。
 - AUTOTRIM モードで、2 回のトリミングステップによって TRIM 値が調整されます。
- 周波数誤差が誤差制限を上回るか同じである場合は、周波数がトリミング範囲外であることを意味します。これは、SYNC 入力クリーンではなく、SYNC パルスが一部欠落している場合に発生する可能性があります (たとえば、USB SOF が 1 つ破損している場合)。
 - SYNCERR または SYNCMISS ステータスが示されます。
 - TRIM 値は、AUTOTRIM モードでは変更されません。

注 : *TRIM フィールドの実際の値が制限値に近く、自動トリミングによってオーバーフローまたはアンダーフローが強制される場合、TRIM 値が誤差ちょうどに設定され、TRIMOVF ステータスが示されます。AUTOTRIM モード (CRS_CR レジスタにセットされた AUTOTRIMEN ビット) では、CRS_CR の TRIM フィールドがハードウェアによって調整され、読み出し専用になります。*

8.3.5 CRS の初期化と設定

RELOAD 値

RELOAD 値は、ターゲット周波数とプリスケール後の同期ソースの周波数間の比率に従って選択します。ゼロ値で期待される同期に達するために、1 ずつ減算します。計算式は、次のとおりです。

$$\text{RELOAD} = (f_{\text{TARGET}} / f_{\text{SYNC}}) - 1$$

RELOAD フィールドのリセット値は、48 MHz のターゲット周波数と 1 kHz の同期信号周波数 (USB からの SOF 信号) に対応します。



FELIM 値

FELIM 値の選択は、HSI48 発振特性とその標準的なトリミングステップサイズに、深く関係しています。最適値は、HSI48 オシレータクロック数を表すトリミングステップサイズの半分の値に対応します。次の計算式を使用できます。

$$FELIM = (f_{TARGET} / f_{SYNC}) * STEP[\%] / 100\% / 2$$

最高のトリム応答を取得するために、結果の小数点以下の値は常に切り上げられます。アプリケーション内でトリミング動作を頻繁に行う必要がない場合は、FELIM 値を少し上げることでトリミングのヒステリシスを増やすことができます。

FELIM フィールドのリセット値は、 $(f_{TARGET} / f_{SYNC}) = 48000$ と標準的なトリミングステップサイズ (0.14%) に対応しています。

注意 : *RELOAD および FELIM フィールドの誤った設定に対するハードウェアの保護はなく、不規則なトリム応答を引き起こす可能性があります。期待される動作モードでは、RELOAD 値を (同期ソース周波数に対して) 適切に設定することを求めます。この値は、128 * FELIM 値よりも大きい値です (OUTRANGE 制限)。*

8.4 CRS 低電力モード

表 21. 低電力モードが CRS に与える影響

モード	説明
SLEEP	影響はありません。 CRS 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	CRS レジスタを停止します。
STANDBY	CRS によって、動作は STOP または STANDBY モードが終了し、HSI48 オシレータがリスタートするまで停止します。

8.5 CRS 割り込み

表 22. 割り込み制御ビット

割り込みイベント	イベントフラグ	有効化 制御ビット	フラグビットの クリア
期待される同期	ESYNCF	ESYNCE	ESYNCC
同期 OK	SYNCOKF	SYNCOKE	SYNCOKC
同期警告	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同期またはトリミングエラー (TRIMOVF、SYNCMISS、SYNCERR)	ERRF	ERRIE	ERRC



8.6 CRS レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

8.6.1 CRS 制御レジスタ (CRS_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 2000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRIM[5:0]						SWSYN C	AUTOTR IMEN	CEN	Res.	ESYNCI E	ERRIE	SYNCW ARNIE	SYNCO KIE
		rw	rw	rw	rw	rw	rw	rt_w	rw	rw		rw	rw	rw	rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **TRIM[5:0]** : HSI48 オシレータ高精度トリミング

これらのビットにより、HSI48 オシレータにユーザプログラム可能なトリミング値が使用できます。HSI48 の周波数に影響する電圧や温度の変化に対応できるようにプログラミングできます。デフォルト値は 32 (トリミングインターバルの中間に相当) です。トリミングステップは、2 つの連続する TRIM ステップの間で、67 kHz 前後です。高い TRIM 値は、高い出力周波数に対応します。AUTOTRIMEN ビットがセットされると、このフィールドはハードウェアによって調整され、読み出し専用になります。

ビット 7 **SWSYNC** : ソフトウェア SYNC イベントの生成

このビットは、ソフトウェア SYNC イベントを生成するためにソフトウェアによってセットされます。ハードウェアによって自動的にクリアされます。
0 : 影響なし。
1 : ソフトウェア SYNC イベントが生成されます。

ビット 6 **AUTOTRIMEN** : 自動トリミング有効

このビットは、2 つの SYNC イベント間で測定した周波数誤差に応じて、TRIM ビットの自動ハードウェア調整を有効にします。このビットがセットされると、TRIM ビットが読み出し専用になります。TRIM 値は、測定した周波数誤差値に応じて、ハードウェアで 1 度に 1 または 2 ステップずつ調整できます。詳細については、[セクション 8.3.4: 周波数誤差評価および自動トリミング](#) を参照してください。
0 : 自動トリミングは無効です。ユーザは、TRIM ビットを調整できます。
1 : 自動トリミングは有効です。TRIM ビットは読み出し専用で、ハードウェアによって制御されます。

ビット 5 **CEN** : 周波数誤差カウンタ有効

このビットは、周波数誤差カウンタのオシレータクロックを有効にします。
0 : 周波数誤差カウンタは無効です。
1 : 周波数誤差カウンタは有効です。
このビットをセットすると、CRS_CFGR レジスタは書き込み保護され、変更できません。

ビット 4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **ESYNCE** : 期待される SYNC 割り込み有効

0 : 期待される SYNC (ESYNCF) 割り込みは無効です。
1 : 期待される SYNC (ESYNCF) 割り込みは有効です。



ビット 2 **ERRIE** : 同期またはトリミングエラー割り込み有効

0 : 同期またはトリミングエラー (ERRF) 割り込みは無効です。

1 : 同期またはトリミングエラー (ERRF) 割り込みは有効です。

ビット 1 **SYNCWARNIE** : SYNC 警告割り込み有効

0 : SYNC 警告 (SYNCWARNF) 割り込みは無効です。

1 : SYNC 警告 (SYNCWARNF) 割り込みは有効です。

ビット 0 **SYNCOKIE** : SYNC イベント OK 割り込み有効

0 : SYNC イベント OK (SYNCOKF) 割り込みは無効です。

1 : SYNC イベント OK (SYNCOKF) 割り込みは有効です。

8.6.2 CRS 設定レジスタ (CRS_CFGR)

このレジスタに書き込めるのは、周波数誤差カウンタが無効なときだけです (CEN ビットは CRS_CR でクリアされる)。このカウンタを有効にすると、このレジスタは書き込み保護されます。

アドレスオフセット : 0x04

リセット値 : 0x2022 BB7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNCPOL	Res.	SYNCSRC[1:0]		Res.	SYNCDIV[2:0]			FELIM[7:0]							
rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **SYNCPOL** : SYNC 極性選択

このビットは、SYNC 信号ソースの入力の極性を選択するために、ソフトウェアによってセット／クリアされます。

0 : SYNC は立ち上がりエッジでアクティブです (デフォルト)。

1 : SYNC は立ち下がりエッジでアクティブです。

ビット 30 予約済みであり、リセット値に保持する必要があります。

ビット 29:28 **SYNCSRC[1:0]** : SYNC 信号ソース選択

これらのビットは、SYNC 信号ソースを選択するために、ソフトウェアによってセット／クリアされます。

00 : SYNC 信号ソースとして GPIO を選択します。

01 : SYNC 信号ソースとして LSE を選択します。

10 : SYNC 信号ソースとして USB SOF を選択します (デフォルト)。

11 : 予約済み

注 : *USB LPM (Link Power Management、リンク電源管理) の使用時にデバイスが SLEEP モードである場合、周期的 USB SOF はホストから生成されません。そのため、実行中に HSI48 を校正するために SYNC 信号が CRS に供給されることはありません。SLEEP モードからのウェイクアップ後に必要なクロック精度を確保するために、GPIO の LSE や基準クロックを SYNC 信号として使用する必要があります。*

ビット 27 予約済みであり、リセット値に保持する必要があります。

- ビット 26:24 **SYNCDIV[2:0]** : SYNC 分周器
- これらのビットは、SYNC 信号の分周比を制御するために、ソフトウェアによってセット／クリアされます。
- 000 : SYNC は分周されません（デフォルト）。
- 001 : SYNC は 2 分周されます。
- 010 : SYNC は 4 分周されます。
- 011 : SYNC は 8 分周されます。
- 100 : SYNC は 16 分周されます。
- 101 : SYNC は 32 分周されます。
- 110 : SYNC は 64 分周されます。
- 111 : SYNC は 128 分周されます。
- ビット 23:16 **FELIM[7:0]** : 周波数誤差制限
- FELIM には、CRS_ISR レジスタの FECAP[15:0] ビットにラッチされた、キャプチャされた周波数誤差値を評価するために使用する値が含まれます。FECAP 評価の詳細については、[セクション 8.3.4 : 周波数誤差評価および自動トリミング](#)を参照してください。
- ビット 15:0 **RELOAD[15:0]** : カウンタ再ロード値
- RELOAD は、SYNC イベントごとに周波数誤差カウンタにロードされる値です。
- カウンタの動作の詳細については、[セクション 8.3.3 : 周波数誤差測定](#)を参照してください。

8.6.3 CRS 割り込みおよびステータスレジスタ（CRS_ISR）

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FECAP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEDIR	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
r					r	r	r					r	r	r	r

- ビット 31:16 **FECAP[15:0]** : 周波数誤差キャプチャ
- FECAP は、最後の SYNC イベント時にラッチされた、周波数誤差カウンタ値です。
- FECAP の使用の詳細については、[セクション 8.3.4 : 周波数誤差評価および自動トリミング](#)を参照してください。
- ビット 15 **FEDIR** : 周波数誤差方向
- FEDIR は、最後の SYNC イベント時にラッチされた、周波数誤差カウンタのカウント方向です。実際の周波数が、ターゲットを下回るか上回るかを示します。
- 0 : アップカウント方向。実際の周波数はターゲットを上回ります。
- 1 : ダウンカウント方向。実際の周波数はターゲットを下回ります。
- ビット 14:11 予約済みであり、リセット値に保持する必要があります。
- ビット 10 **TRIMOVF** : トリミングのオーバーフローまたはアンダーフロー
- このフラグは、自動トリミングによって TRIM 値のオーバーフローまたはアンダーフローが試みられた場合に、ハードウェアによってセットされます。CRS_CR レジスタの ERRIE ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることで、ソフトウェアによってクリアされます。
- 0 : トリミングエラーを信号化しません。
- 1 : トリミングエラーを信号化します。



ビット 9 SYNCMISS : SYNC の欠落

このフラグは、周波数誤差カウンタが値 FELIM * 128 に達し、SYNC が検出されない場合にハードウェアによってセットされます。これは、SYNC パルスが失われているということ、または TRIM 値を調整して補正するには周波数誤差が大きすぎ（内部周波数が高すぎる）、他の操作を行う必要があるということを意味します。この時点で、周波数誤差カウンタは停止し（次の SYNC を待つ）、CRS_CR レジスタの ERRIE ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC の欠落エラーを信号化しません。

1 : SYNC の欠落エラーを信号化します。

ビット 8 SYNCERR : SYNC エラー

このフラグは、SYNC パルスが ESYNC イベントより前に発生し、測定した周波数誤差が FELIM * 128 以上である場合に、ハードウェアによってセットされます。これは、TRIM 値を調整して補正するには周波数誤差が大きすぎ（内部周波数が低すぎる）、他の操作を行う必要があるということを意味します。CRS_CR レジスタの ERRIE ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC エラーを信号化しません。

1 : SYNC エラーを信号化します。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 ESYNCF : 期待される SYNC フラグ

このフラグは、周波数誤差カウンタがゼロ値に達したときに、ハードウェアによってセットされます。CRS_CR レジスタの ESYNCF ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの ESYNCC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : 期待された SYNC を信号化しません。

1 : 期待された SYNC を信号化します。

ビット 2 ERRF : エラーフラグ

このフラグは、同期またはトリミングエラー時に、ハードウェアによってセットされます。TRIMOVF、SYNCMISS、および SYNCERR ビットの論理和です。CRS_CR レジスタの ERRIE ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの ERRC ビットをセットすることへの反応として、ソフトウェアによってクリアされます。TRIMOVF、SYNCMISS、および SYNCERR ビットをクリアします。

0 : 同期またはトリミングエラーを信号化しません。

1 : 同期またはトリミングエラーを信号化します。

ビット 1 SYNCWARNF : SYNC 警告フラグ

このフラグは、測定した周波数誤差が FELIM * 3 以上で FELIM * 128 未満である場合に、ハードウェアによってセットされます。周波数誤差を補正するには、TRIM 値を複数のステップで調整する必要があります。CRS_CR レジスタの SYNCWARNIE ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの SYNCWARNIC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC 警告を信号化しません。

1 : SYNC 警告を信号化します。

ビット 0 SYNCOKF : SYNC イベント OK フラグ

このフラグは、測定した周波数誤差が FELIM * 3 未満である場合に、ハードウェアによってセットされます。これは、TRIM 値の補正が必要であること、または周波数誤差の補正は 1 つのトリミングステップで十分であることを意味します。CRS_CR レジスタの SYNCOKIE ビットがセットされている場合、割り込みが生成されます。CRS_ICR レジスタの SYNCOKIC ビットをセットすることで、ソフトウェアによってクリアされます。

0 : SYNC イベント OK フラグを信号化しません。

1 : SYNC イベント OK フラグを信号化します。

8.6.4 CRS 割り込みフラグクリアレジスタ (CRS_ICR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC
												rw	rw	rw	rw

ビット 31:4 予約済みであり、リセット値のままにしておかなければなりません。

ビット 3 **ESYNCC** : 期待 SYNC クリアフラグ

このビットに 1 を書き込むと、CRS_ISR レジスタの ESYNCF フラグをクリアします。

ビット 2 **ERRC** : エラークリアフラグ

このビットに 1 を書き込むと、TRIMOVF、SYNCMISS、および SYNCERR ビットがクリアされ、結果として CRS_ISR レジスタの ERRF フラグもクリアされます。

ビット 1 **SYNCWARNC** : SYNC 警告クリアフラグ

このビットに 1 を書き込むと、CRS_ISR レジスタの SYNCWARNF フラグをクリアします。

ビット 0 **SYNCOKC** : SYNC イベント OK クリアフラグ

このビットに 1 を書き込むと、CRS_ISR レジスタの SYNCOKF フラグをクリアします。

8.6.5 CRS レジスタマップ

表 23. CRS レジスタマップとリセット値

オフ	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CRS_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]						SWSYNC	AUTOTRIMEN	CEN	Res.	ESYNCE	ERRIE	SYNCWARNIE	SYNCOKIE
	リセット値																			1	0	0	0	0	0		0	0	0	0	0	0	0
0x04	CRS_CFGR	SYNCPOL	Res.	SYNC SRC [1:0]		Res.	SYNC DIV [2:0]		FELIM[7:0]							RELOAD[15:0]																	
	リセット値	0		1	0		0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1
0x08	CRS_ISR	FECAP[15:0]																FEDIR	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0					0	0	0
0x0C	CRS_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC
	リセット値																													0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

9 汎用 I/O (GPIO)

9.1 概要

各汎用 I/O ポートにはそれぞれ 4 つの 32 ビット設定レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR)、2 つの 32 ビットデータレジスタ (GPIOx_IDR、GPIOx_ODR)、および 1 つの 32 ビットセット/リセットレジスタ (GPIOx_BSRR) があります。ポート A および B にも、それぞれ 32 ビットロックレジスタ (GPIOx_LCKR) と 2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx_AFRH、GPIOx_AFRL) があります。

STM32F07x および STM32F09x デバイスでは、ポート C、D、および E にも 2 つの 32 ビットオルタネート機能選択レジスタ (GPIOx_AFRH および GPIOx_AFRL) があります。

ポート E は STM32F07x および STM32F09x デバイスでのみ使用可能です。

9.2 GPIO の主な機能

- 出力状態：プッシュプルまたはオープンドレイン + プルアップ/プルダウン
- 出力データレジスタ (GPIOx_ODR) またはペリフェラル (オルタネート機能出力) からの出力データ
- 各 I/O のスピード選択
- 入力状態：フローティング、プルアップ/プルダウン、アナログ
- 入力データレジスタ (GPIOx_IDR) またはペリフェラル (オルタネート機能入力) への入力データ
- GPIOx_ODR へのビット単位の書き込みアクセス用のビットセット/リセットレジスタ (GPIOx_BSRR)
- ポート A または B の I/O ポート設定を固定するロック機構 (GPIOx_LCKR)。
- アナログ機能
- オルタネート機能選択レジスタ (入出力ごとに最大 16 のオルタネート機能を使用可能)
- 2 クロックサイクルごとに変化可能な高速トグル
- 柔軟性の高いピンの多重化により、I/O ピンを GPIO またはいくつかのペリフェラル機能の 1 つとして使用可能

9.3 GPIO の機能説明

各 I/O ポートの特定のハードウェア特性については、データシートに記載されています。汎用 I/O (GPIO) ポートの各ポートビットは、ソフトウェアによって以下の動作モードを個別に設定できます。

- 入力フローティング
- 入力プルアップ
- 入力プルダウン
- アナログ
- プルアップまたはプルダウン機能を持つ出力オープンドレイン
- プルアップまたはプルダウン機能を持つ出力プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能プッシュプル
- プルアップまたはプルダウン機能を持つオルタネート機能オープンドレイン

各 I/O ポートビットは自由にプログラム可能ですが、I/O ポートレジスタには 32 ビットワード、ハーフワード、またはバイト単位でアクセスする必要があります。GPIOx_BSRR および GPIOx_BRR レジスタを使用すると、任意の GPIOx_ODR レジスタに不可分な読み出し／変更アクセスを行うことができます。これにより、読み出しと変更アクセスの間に IRQ が発生するリスクを回避できます。

図 17 は、標準 I/O ポートビットの基本構造を示しています。表 24 には、実行可能なポートビットの設定を示します。

図 17. I/O ポートビットの基本構造

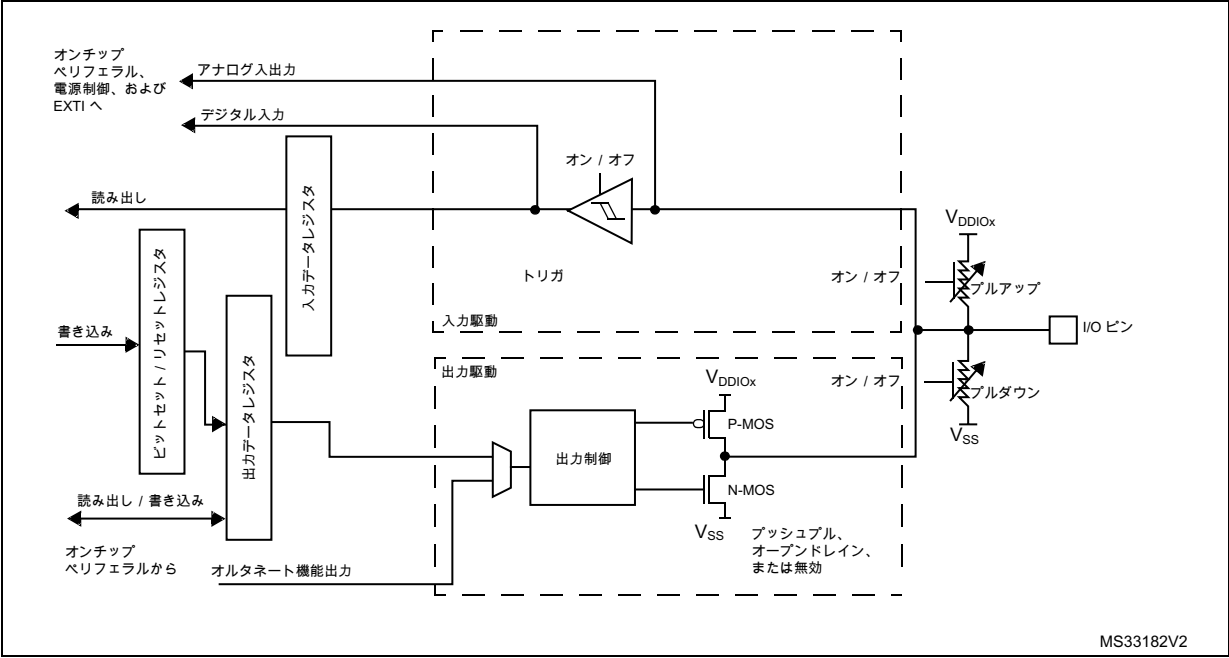


表 24. ポートビット設定表⁽¹⁾

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [1:0]	PUPDR(i) [1:0]		I/O 設定	
01	0	SPEED [1:0]	0	0	GP 出力	PP
	0		0	1	GP 出力	PP + PU
	0		1	0	GP 出力	PP + PD
	0		1	1	予約済み	
	1		0	0	GP 出力	OD
	1		0	1	GP 出力	OD + PU
	1		1	0	GP 出力	OD + PD
	1		1	1	予約済み (GP 出力 OD)	

表 24. ポートビット設定表⁽¹⁾ (続き)

MODER(i) [1:0]	OTYPER(i)	OSPEEDR(i) [1:0]		PUPDR(i) [1:0]		I/O 設定	
10	0	SPEED [1:0]		0	0	AF	PP
	0			0	1	AF	PP + PU
	0			1	0	AF	PP + PD
	0			1	1	予約済み	
	1			0	0	AF	OD
	1			0	1	AF	OD + PU
	1			1	0	AF	OD + PD
	1			1	1	予約済み	
00	x	x	x	0	0	入力	フローティング
	x	x	x	0	1	入力	PU
	x	x	x	1	0	入力	PD
	x	x	x	1	1	予約済み (入力フローティング)	
11	x	x	x	0	0	入力/出力	アナログ
	x	x	x	0	1	予約済み	
	x	x	x	1	0		
	x	x	x	1	1		

1. GP = 汎用、PP = プッシュプル、PU = プルアップ、PD = プルダウン、OD = オープンドレイン、AF = オルタネート機能

9.3.1 汎用 I/O (GPIO)

リセット中とリセット直後は、オルタネート機能は有効ではなく、ほとんどの I/O ポートは入力フローティングモードに設定されています。

リセット後、デバッグピンはオルタネート機能のプルアップ/プルダウンに設定されています。

- PA14 : SWCLK プルダウン
- PA13 : SWDIO プルアップ

ピンが出力として設定されている場合、出力データレジスタ (GPIOx_ODR) に書き込まれた値が I/O ピンに出力されます。出力ドライバをプッシュプルモードまたはオープンドレインモード (ローレベルのみが駆動され、ハイレベルはハイインピーダンス) で使用することができます。

入力データレジスタ (GPIOx_IDR) は、AHB クロックサイクルごとに、I/O ピン上のデータをキャプチャします。

すべての GPIO ピンに、内部ウィークプルアップ抵抗とウィークプルダウン抵抗があり、GPIOx_PUPDR レジスタの値によってこれらを有効化/無効化できます。

9.3.2 I/O ピンオルタネート機能マルチプレクサと配置

デバイスの I/O ピンは、マルチプレクサを介してオンボードのペリフェラル／モジュールに接続され、一度に 1 つのペリフェラルオルタネート機能 (AF) のみが 1 つの I/O ピンに接続可能となっています。この方法により、同じ I/O ピンを共有するペリフェラル間での競合を無くすることができます。

各 I/O ピンは、最大 16 のオルタネート機能入力 (AF0~AF15) を持つマルチプレクサを内蔵しており、GPIOx_AFRL (ピン 0~7) と GPIOx_AFRH (ピン 8~15) レジスタを介して設定することができます。

- リセット後、マルチプレクサの選択はオルタネート機能 0 (AF0) です。I/O は、GPIOx_MODER レジスタを通してオルタネート機能モードで設定されます。
- 各ピンに固有のオルタネート機能割り当てについての詳細は、デバイスデータシートに記載されています。

この柔軟性の高い I/O 多重化アーキテクチャに加え、各ペリフェラルではオルタネート機能がそれぞれの I/O ピンに配置されており、さらに小型のパッケージで利用できるペリフェラルの数を最適化します。

I/O を任意の設定で使用するには、次の手順に従います。

- **デバッグ機能：** 各デバイスのリセット後、これらのピンはデバッグホストによってすぐに使用可能なオルタネート機能ピンとして割り当てられます。
- **GPIO：** 必要とする I/O を、GPIOx_MODER レジスタで出力、入力、またはアナログとして設定します。
- **ペリフェラルオルタネート機能**
 - I/O を GPIOx_AFRL または GPIOx_AFRH レジスタのどちらかで必要とする AFx に接続します。
 - タイプ、プルアップ／プルダウン、出力スピードをそれぞれ GPIOx_OTYPER、GPIOx_PUPDR、GPIOx_OSPEEDER レジスタで選択します。
 - 必要とする I/O を、GPIOx_MODER レジスタでオルタネート機能として設定します。
- **追加機能：**
 - ADC および DAC については、GPIOx_MODER レジスタのアナログモードで必要とする I/O を設定し、ADC または DAC レジスタで必要な機能を設定します。
 - RTC、WKUPx、オシレータなどの追加機能については、関連する RTC、PWR、および RCC レジスタで必要な機能を設定します。これらの機能は、標準の GPIO レジスタの設定よりも優先されます。

オルタネート機能 I/O ピンの配置に関する詳細は、デバイスデータシートの「オルタネート機能配置」表を参照してください。

9.3.3 I/O ポート制御レジスタ

各 GPIO ポートには 32 ビットメモリマップド制御レジスタが 4 つ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR) あり、最大 16 個の I/O を設定します。GPIOx_MODER レジスタは I/O モード (入力、出力、AF、アナログ) を選択するために使用されます。GPIOx_OTYPER および GPIOx_OSPEEDR レジスタは、出力タイプ (プッシュプルまたはオープンドレイン) およびスピードを選択するために使用されます。I/O の方向がどちらであっても、GPIOx_PUPDR レジスタは、プルアップ／プルダウンを選択するために使用されます。

9.3.4 I/O ポートデータレジスタ

各 GPIO には、16 ビットメモリマップドデータレジスタが 2 つあります。入力データレジスタ (GPIOx_IDR) と出力データレジスタ (GPIOx_ODR) です。GPIOx_ODR は出力されるデータを格納し、読み出し/書き込みアクセスが可能です。I/O から入力されるデータは読み出し専用の入力データレジスタ (GPIOx_IDR) に格納されます。

レジスタの説明は、[セクション 9.4.5 : GPIO ポート入力データレジスタ \(GPIOx_IDR\) \(x = A ~ F\)](#) および [セクション 9.4.6 : GPIO ポート出力データレジスタ \(GPIOx_ODR\) \(x = A ~ F\)](#) を参照してください。

9.3.5 I/O データのビット単位の操作

ビットセット/リセットレジスタ (GPIOx_BSRR) は 32 ビットレジスタで、アプリケーションによる出力データレジスタ (GPIOx_ODR) のビット単位のセット/リセットを可能にします。ビットセット/リセットレジスタは GPIOx_ODR の 2 倍のサイズです。

GPIOx_ODR の各ビットには GPIOx_BSRR の 2 つの制御ビット BS(i) と BR(i) が対応します。ビット BS(i) に 1 を書き込むと、対応する ODR(i) ビットが**セット**されます。ビット BR(i) に 1 を書き込むと、対応する ODR(i) ビットが**リセット**されます。

GPIOx_BSRR のいかなるビットに 0 を書き込んでも GPIOx_ODR の対応するビットには影響しません。仮に、GPIOx_BSRR のビットに対してセットおよびリセットの両方を実行しようとした場合、セット動作が優先されます。

GPIOx_BSRR レジスタを使用した GPIOx_ODR 内の個々のビットの変更には、1 回限りの効果しかなく、GPIOx_ODR ビットを固定するものではありません。GPIOx_ODR のビットは常に直接アクセスすることができます。GPIOx_BSRR レジスタによって、ビット単位の不可分操作を行うことができます。

GPIOx_ODR をビットレベルでプログラムする場合は、ソフトウェアで割り込みを無効にする必要はありません。1 回の不可分な AHB 書き込みアクセスで 1 ビットまたは複数ビットを変更することができます。

9.3.6 GPIO ロック機構

GPIOx_LCKR レジスタへ特定の書き込みシーケンスを行うことにより、ポート A および B の GPIO 制御レジスタをロックすることができます。ロックされるレジスタは、GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH です。

GPIOx_LCKR レジスタに書き込むには、特定の書き込み/読み出しシーケンスを行う必要があります。このレジスタのビット 16 に適切な LOCK シーケンスを行う場合、LCKR[15:0] の値を使用して I/O の設定を固定します (この書き込みシーケンス中、LCKR[15:0] の値は同じである必要があります)。あるポートビットに LOCK シーケンスが行われると、次の MCU リセットまたはペリフェラルリセットまで、そのポートビットの値を変更できなくなります。GPIOx_LCKR の各ビットによって、制御レジスタ (GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL、GPIOx_AFRH) の対応するビットが停止されます。

GPIOx_LCKR ビット 16 を [15:0] ビットと同時に設定する必要があるため、この LOCK シーケンス ([セクション 9.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx_LCKR\) \(x = A ~ B\)](#)) を参照) は、GPIOx_LCKR レジスタへのワード (32 ビット長) アクセスを使用してのみ実行できます。

詳細については、[セクション 9.4.8 : GPIO ポート設定ロックレジスタ \(GPIOx_LCKR\) \(x = A ~ B\)](#) の LCKR レジスタの説明を参照してください。

9.3.7 I/O オルタネート機能の入力／出力

各 I/O が使用できるオルタネート機能入力／出力の 1 つを選択するため、2 つのレジスタが用意されています。これらのレジスタを使用し、必要に応じて、アプリケーションでオルタネート機能を他のピンに接続することができます。

つまり、GPIOx_AFRL および GPIOx_AFRH オルタネート機能レジスタを使用していくつかの使用可能なペリフェラル機能が、各 GPIO に多重化されることになります。こうして各 I/O に使用可能な機能のどれか 1 つをアプリケーションによって選択できます。AF 選択信号はオルタネート機能入力およびオルタネート機能出力に共通なので、任意の I/O が持つオルタネート機能入力／出力に対し 1 つのチャンネルが選択されます。

コード例については、「付録」セクション [A.4.2 : オルタネート機能の選択シーケンスのコード例](#) を参照してください。

どの機能が各 GPIO ピンに多重化されているかについてはデバイスデータシートを参照してください。

9.3.8 外部割り込み／ウェイクアップライン

すべてのポートに外部割り込み機能があります。外部割り込みラインを使用するには、該当ピンをアナログモードに設定したり、オシレータピンとして使用したりしないでください。そして、入力トリガは有効のままにしておいてください。[セクション 12.2 : 拡張割り込み／イベントコントローラ \(EXTI\)](#) を参照してください。

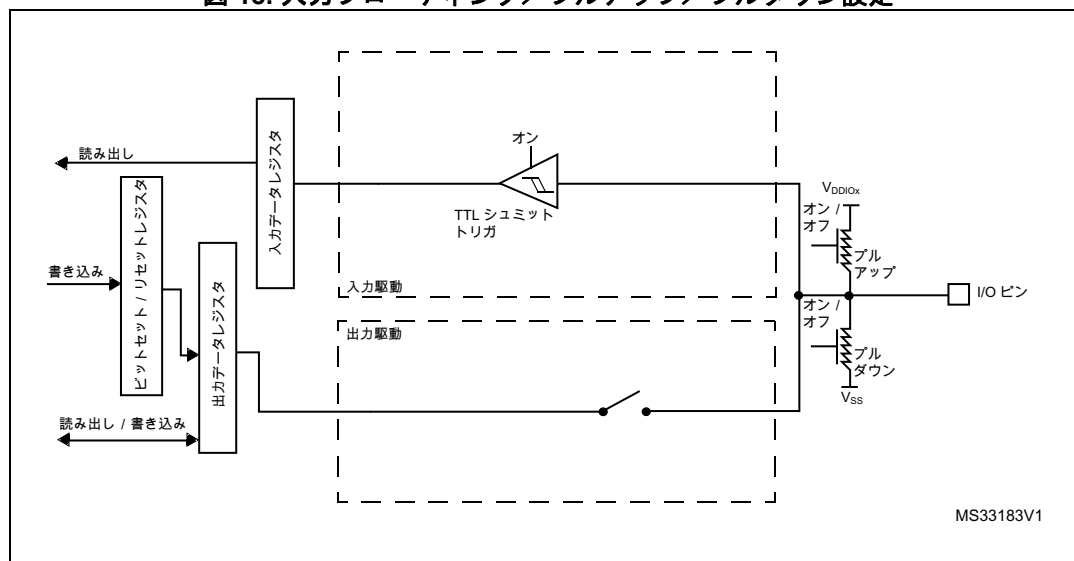
9.3.9 入力設定

I/O ポートが入力としてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力が有効になります。
- GPIOx_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

[図 18](#) は、I/O ポートビットの入力設定を示しています。

図 18. 入力フローティング／プルアップ／プルダウン設定



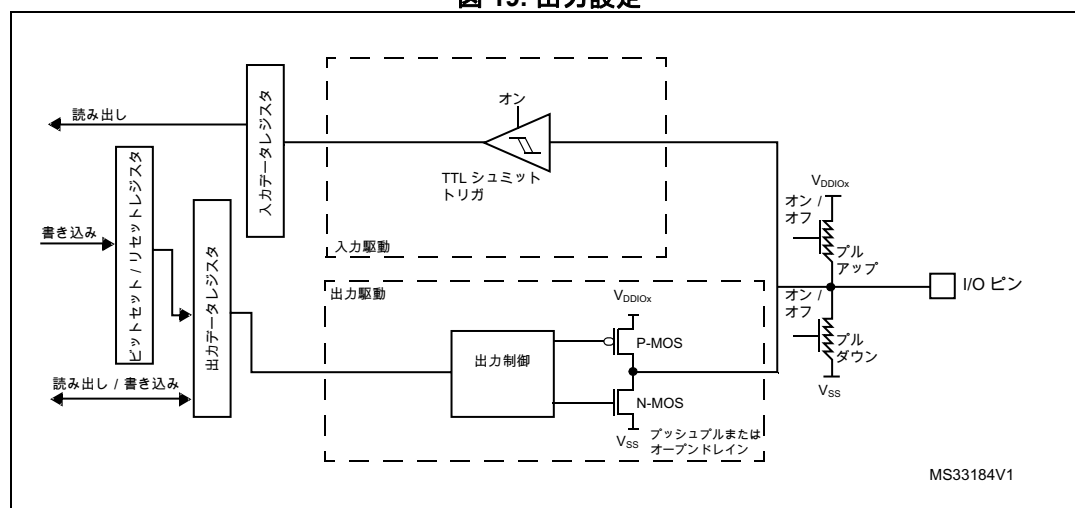
9.3.10 出力設定

I/O ポートが出力としてプログラムされた場合、

- 出力バッファが有効になります。
 - オープンドレインモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときはポートはハイインピーダンス状態のままです (P-MOS が有効になることはない)。
 - プッシュプルモード：出力レジスタが“0”のときは N-MOS が有効になり、“1”のときは P-MOS が有効になります。
- シュミットトリガ入力が有効になります。
- GPIOx_PUPDR レジスタの値に応じて、プルアップおよびプルダウン抵抗が有効になります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。
- 出力データレジスタの読み出しアクセスによって、最後に書き込まれたデータが得られます。

図 19 は、I/O ポートビットの出力設定を示しています。

図 19. 出力設定



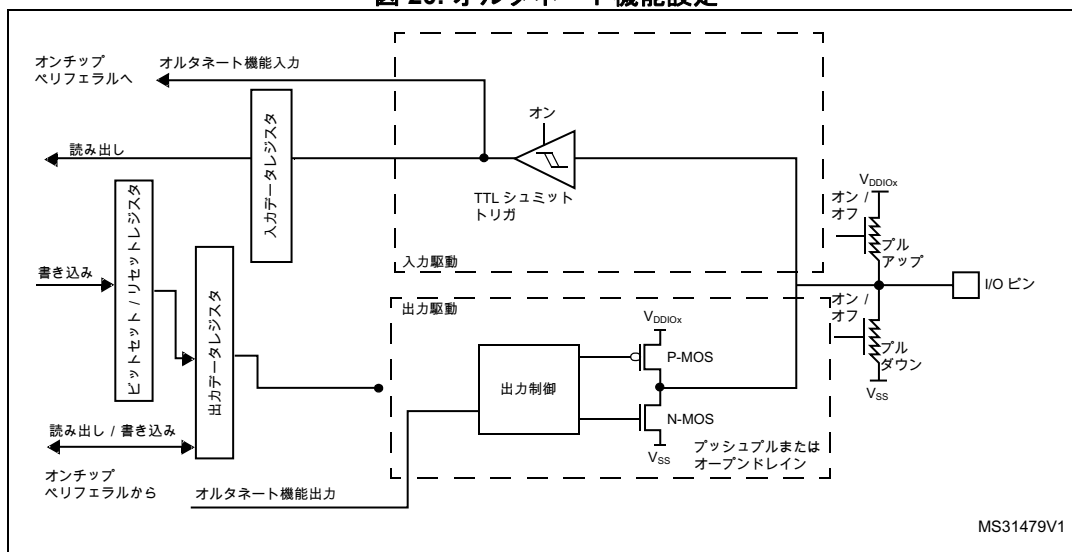
9.3.11 オルタネート機能設定

I/O ポートがオルタネート機能としてプログラムされた場合、

- 出力バッファをオープンドレインまたはプッシュプルモードに設定することができます。
- 出力バッファが、ペリフェラル（トランスミッタ、イネーブルおよびデータ）からの信号で駆動されます。
- シュミットトリガ入力が有効になります。
- ウィークプルアップ抵抗およびプルダウン抵抗が有効になるか否かは、GPIOx_PUPDR レジスタの値によって決まります。
- I/O ピン上のデータは、AHB クロックサイクルごとに入力データレジスタにサンプリングされます。
- 入力データレジスタへの読み出しアクセスによって、I/O 状態が得られます。

図 20 は、I/O ポートビットのオルタネート機能設定を示しています。

図 20. オルタネート機能設定



9.3.12 アナログ設定

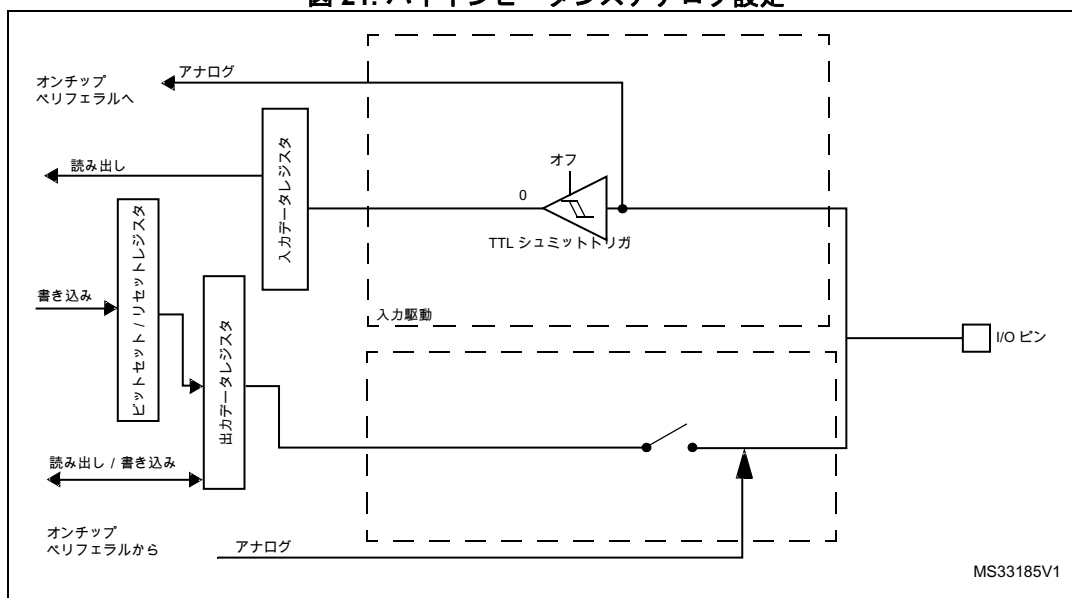
I/O ポートがアナログとしてプログラムされた場合、

- 出力バッファが無効になります。
- シュミットトリガ入力は無効になり、I/O ピンのどのようなアナログ値に対しても消費電力をゼロに抑えます。シュミットトリガ回路の出力は、常に“0”になります。
- ウィークプルアップ抵抗およびプルダウン抵抗はハードウェアによって無効にされます。
- 入力データレジスタの読み出しアクセスを行うと、値“0”が得られます。

コード例については、「付録」セクション [A.4.3: アナログ GPIO 設定のコード例](#) を参照してください。

図 21 は、I/O ポートビットの高インピーダンスアナログ入力設定を示しています。

図 21. ハイインピーダンスアナログ設定



9.3.13 HSEまたはLSE オシレータのピンを GPIO として使用

HSE または LSE オシレータがスイッチオフされた場合（リセット後のデフォルト状態）、関連のオシレータピンを 通常の GPIO として使用することができます。

HSE または LSE オシレータがスイッチオンされた場合（RCC_CSR レジスタの HSEON または LSEON ビットを設定することで）、オシレータは関連ピンを制御しますが、これらのピンの GPIO 設定は無効です。

オシレータがユーザ外部クロックモードに設定されている場合、ピンのみがクロック入力のために確保されますが、OSC_OUT または OSC32_OUT ピンは通常の GPIO として使用することができます。

9.3.14 GPIO ピンを RTC 供給ドメインで使用

コア供給ドメインの電源がオフになったとき（デバイスが STANDBY モードに移行したとき）、PC13/PC14/PC15 の GPIO が機能しなくなります。この場合、それらの GPIO 設定が RTC 設定によってバイパスされなければ、これらのピンはアナログ入力モードに設定されます。

RTC による I/O 制御の詳細は、[セクション 24.4: RTC の機能説明 \(569 ページ\)](#) を参照してください。

9.4 GPIO レジスタ

このセクションには、GPIO レジスタの詳細な説明が記載されています。

レジスタビット、レジスタアドレスオフセット、リセット値の概要は、表 25 を参照してください。

ペリフェラルレジスタはワード、ハーフワード、バイトのいずれかのモードで書き込むことができます。

9.4.1 GPIO ポートモードレジスタ (GPIOx_MODER) (x = A ~ F)

アドレスオフセット : 0x00

リセット値 :

- ポート A 0x2800 0000
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 2y+1:2y **MODERy[1:0]** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O モードを設定するために、ソフトウェアによって書き込まれます。

- 00 : 入力モード (リセット状態)
- 01 : 汎用出力モード
- 10 : オルタネート機能モード
- 11 : アナログモード

9.4.2 GPIO ポート出力タイプレジスタ (GPIOx_OTYPER) (x = A ~ F)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **OTy** : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O 出力タイプを設定するために、ソフトウェアによって書き込まれます。

- 0 : 出力プッシュプル (リセット状態)
- 1 : 出力オープンドレイン



9.4.3 GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) (x = A ~ F)

アドレスオフセット : 0x08

リセット値 :

- ポート A 0x0C00 0000
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3 [1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 2y+1:2y OSPEEDRy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O の出力スピードを設定するために、ソフトウェアによって書き込まれます。

- x0 : ロースピード
01 : ミディアムスピード
11 : ハイスピード

注 : 周波数仕様、およびスピード別の電源や負荷条件については、デバイスデータシートを参照してください。

9.4.4 GPIO ポートプルアップ／プルダウンレジスタ (GPIOx_PUPDR) (x = A ~ F)

アドレスオフセット : 0x0C

リセット値 :

- ポート A 0x2400 0000
- 他のポート 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 2y+1:2y PUPDRy[1:0] : ポート x 設定ビット (y = 0~15)

これらのビットは、I/O のプルアップまたはプルダウンを設定するために、ソフトウェアによって書き込まれます。

- 00 : プルアップ／プルダウンなし
01 : プルアップ
10 : プルダウン
11 : 予約済み

9.4.5 GPIO ポート入力データレジスタ (GPIOx_IDR) (x = A ~ F)

アドレスオフセット : 0x10

リセット値 : 0x0000 XXXX (X は未定義)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **IDRy** : ポート入力データビット (y = 0~15)

これらのビットは読み出し専用です。これらのビットには、対応する I/O ポートの入力値が格納されています。

9.4.6 GPIO ポート出力データレジスタ (GPIOx_ODR) (x = A ~ F)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **ODRy** : ポート出力データビット (y = 0~15)

これらのビットは、ソフトウェアによって読み出し/書き込みができます。

注 : アトミックビット単位のセット/リセットのために、GPIOx_BSRR または GPIOx_BRR レジスタ (x = A ~ F) への書き込みによって、ODR ビットを個々にセット/リセットすることができます。

9.4.7 GPIO ポートビットセット／リセットレジスタ (GPIOx_BSRR) (x = A ~ F)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 **BRy** : ポート x リセットビット y (y= 0~15)
これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。
0 : 対応する ODRx ビットの値は変化しません。
1 : 対応する ODRx ビットをリセットします。
注 : *BSx ビットと BRx ビットの両方がセットされた場合、BSx ビットが優先されます。*

ビット 15:0 **BSy** : ポート x セットビット y (y= 0~15)
これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。
0 : 対応する ODRx ビットの値は変化しません。
1 : 対応する ODRx ビットをセットします。

9.4.8 GPIO ポート設定ロックレジスタ (GPIOx_LCKR) (x = A ~ B)

このレジスタは、ビット 16 (LCKK) に正しい書き込みシーケンスが行われたときに、ポートビットの設定をロックするために使用されます。ビット [15:0] の値は、GPIO の設定をロックするために使用されます。書き込みシーケンスの間は、LCKR[15:0] の値を変更することはできません。あるポートビットに LOCK シーケンスが適用されると、次の MCU リセットまたはペリフェラルリセットまで、このポートビットの値を変更できなくなります。

注 : *GPIOx_LCKR レジスタへの書き込みには特定の書き込みシーケンスが使われます。このロックシーケンスではワードアクセス (32 ビット長) のみ可能です。*

各ロックビットによって、特定の設定レジスタ (制御レジスタおよびオルタネート機能レジスタ) が固定されます。

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **LCKK** : ロックキー

このビットは常時読み出しができます。ロックキー書き込みシーケンスを使用しなければ変更できません。

- 0 : ポート設定ロックキーは無効です。
 - 1 : ポート設定ロックキーは有効です。GPIOx_LCKR レジスタは、次の MCU リセットまたはペリフェラルリセットまでロックされます。
- ロックキー書き込みシーケンス :
- 書き込み LCKR[16] = “1” + LCKR[15:0]
- 書き込み LCKR[16] = “0” + LCKR[15:0]
- 書き込み LCKR[16] = “1” + LCKR[15:0]
- 読み出し LCKR
- 読み出し LCKR[16] = “1” (この読み出し操作は任意だが、ロックが有効であることを確認できる。)

注 : **ロックキー書き込みシーケンスの間は、LCK[15:0] の値を変更することはできません。**
 ロックシーケンス中にエラーが発生すると、ロックは中止されます。
 ポートの任意のビットの最初のロックシーケンスの後、次の MCU リセットまたはペリフェラルリセットまでは、LCKK ビットのかなる読み出しアクセスに対しても、“1”が返されます。

コード例については、「付録」セクション **A.4.1 : ロックシーケンスのコード例** を参照してください。

ビット 15:0 **LCKy** : ポート x ロックビット y (y= 0~15)

- これらのビットは読み出し／書き込みができますが、書き込めるのは LCKK ビットが“0”のときだけです。
- 0 : ポート設定はロックされません。
 - 1 : ポート設定はロックされます。

9.4.9 GPIO オルタネート機能下位レジスタ (GPIOx_AFRL) (x = A ~ F)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **AFRy[3:0]** : ポート x ピン y (y = 0~7) のオルタネート機能選択

これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRy 選択 :	1000 : 予約済み
0000 : AF0	1001 : 予約済み
0001 : AF1	1010 : 予約済み
0010 : AF2	1011 : 予約済み
0011 : AF3	1100 : 予約済み
0100 : AF4	1101 : 予約済み
0101 : AF5	1110 : 予約済み
0110 : AF6	1111 : 予約済み
0111 : AF7	



9.4.10 GPIO オルタネート機能上位レジスタ (GPIOx_AFRH)
(x = A ~ F)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 AFRy[3:0] : ポート x ピン y (y = 8~15) のオルタネート機能選択
これらのビットは、オルタネート機能 I/O を設定するために、ソフトウェアによって書き込まれます。

AFRy 選択 :	
0000 : AF0	1000 : 予約済み
0001 : AF1	1001 : 予約済み
0010 : AF2	1010 : 予約済み
0011 : AF3	1011 : 予約済み
0100 : AF4	1100 : 予約済み
0101 : AF5	1101 : 予約済み
0110 : AF6	1110 : 予約済み
0111 : AF7	1111 : 予約済み

9.4.11 GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A ~ F)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 予約済み
ビット 15:0 BRy : ポート x リセットビット y (y = 0 ~ 15)
これらのビットは書き込み専用です。これらのビットを読み出すと、値 0x0000 が返されます。
0 : 対応する ODx ビットの値は変化しません。
1 : 対応する ODx ビットをリセットします。

9.4.12 GPIO レジスタマップ

次の表に、GPIO レジスタマップとリセット値を示します。

表 25. GPIO レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	GPIOA_MODER	MODER15[1:0]																															
	リセット値	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00	GPIOx_MODER (x = B ~ F)	MODER15[1:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	GPIOx_OTYPER (x = A ~ F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOA_OSPEEDR	OSPEEDR15[1:0]																															
	リセット値	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOx_OSPEEDR (x = B ~ F)	OSPEEDR15[1:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	GPIOA_PUPDR	PUPDR15[1:0]																															
	リセット値	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	GPIOx_IDR (x = A ~ F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	リセット値																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	GPIOx_ODR (x = A ~ F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (x = A ~ F)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	GPIOx_LCKR (x = A ~ B)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (x = A ~ F)	AFRLAFR7[3:0]				AFRLAFR6[3:0]				AFRLAFR5[3:0]				AFRLAFR4[3:0]				AFRLAFR3[3:0]				AFRLAFR2[3:0]				AFRLAFR1[3:0]				AFRLAFR0[3:0]			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	GPIOx_AFRH (x = A ~ F)	AFRHAFR15[3:0]				AFRHAFR14[3:0]				AFRHAFR13[3:0]				AFRHAFR12[3:0]				AFRHAFR11[3:0]				AFRHAFR10[3:0]				AFRHAFR9[3:0]				AFRHAFR8[3:0]			
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	GPIOx_BRR (x = A ~ F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

10 システム設定コントローラ（SYSCFG）

デバイスは一連の設定レジスタを持っています。システム設定コントローラの主な目的は次の通りです。

- 一部の I/O ポート上での I²C 高速モードプラスの有効化／無効化
- 一部の DMA トリガソースの別の DMA チャンネルへの再配置
- コード領域の先頭に配置されたメモリの再配置
- STM32F09x デバイスの各割り込みライン用の保留中の割り込みステータスレジスタ
- 外部割り込みラインのGPIO への接続の管理
- 堅牢性の管理機能

10.1 SYSCFG レジスタ

10.1.1 SYSCFG 設定レジスタ 1（SYSCFG_CFGR1）

このレジスタは、メモリと DMA リクエストの再配置時の特定の設定と、特殊な I/O 機能を制御するために使用されます。

アドレス 0x0000 0000 でアクセス可能なメモリのタイプを設定するために、2 つのビットが使用されます。これらのビットは、ソフトウェアによる物理的な再配置を選択し、ハードウェア BOOT 選択をバイパスするために使用します。

リセット後、これらのビットは、実際のブートモード設定で選択された値になります。

アドレスオフセット：0x00

リセット値：0x0000 000X（X は実際のブートモード設定で選択されたメモリモード）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIM3_DMA_RMP	TIM2_DMA_RMP	TIM1_DMA_RMP	I2C1_DMA_RMP	USART3_DMA_RMP	USART2_DMA_RMP	SPI2_DMA_RMP	I2C_PA10_FMP	I2C_PA9_FMP	I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TIM17_DMA_RMP2	TIM16_DMA_RMP2	TIM17_DMA_RMP	TIM16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	IRDA_ENV_SEL[1:0]		Res.	PA11_PA12_RMP	Res.	Res.	MEM_MODE[1:0]	
	rw	rw	rw	rw	rw	rw	rw	rw			rw			rw	rw

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30 **TIM3_DMA_RMP**: TIM3 の DMA リクエスト再配置ビット。STM32F07x デバイスでのみ使用できます。
このビットは、ソフトウェアによってセット／クリアされます。TIM3 の DMA リクエストの再配置を制御します。

0: 再配置なし (TIM3_CH1 と TIM3_TRIG の DMA リクエストは DMA チャンネル 4 に配置される)

1: 再配置 (TIM3_CH1 と TIM3_TRIG の DMA リクエストは DMA チャンネル 6 に配置される)

ビット 29 **TIM2_DMA_RMP**: TIM2 の DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。
このビットは、ソフトウェアによってセット／クリアされます。TIM2 の DMA リクエストの再配置を制御します。

0: 再配置なし (TIM2_CH2 と TIM2_CH4 の DMA リクエストは、それぞれ DMA チャンネルの 3 と 4 に配置される)

1: 再配置 (TIM2_CH2 と TIM2_CH4 の DMA リクエストは DMA チャンネル 7 に配置される)

ビット 28 **TIM1_DMA_RMP**: TIM1 の DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。
このビットは、ソフトウェアによってセット／クリアされます。TIM1 の DMA リクエストの再配置を制御します。

0: 再配置なし (TIM1_CH1、TIM1_CH2、および TIM1_CH3 の DMA リクエストは、それぞれ DMA チャンネルの 2、3、および 4 に配置される)

1: 再配置 (TIM1_CH1、TIM1_CH2、および TIM1_CH3 の DMA リクエストは DMA チャンネル 6 に配置される)

ビット 27 **I2C1_DMA_RMP**: I2C1 の DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。
このビットは、ソフトウェアによってセット／クリアされます。I2C1 の DMA リクエストの再配置を制御します。

0: 再配置なし (I2C1_RX と I2C1_TX の DMA リクエストは、それぞれ DMA チャンネルの 3 と 2 に配置される)

1: 再配置 (I2C1_RX と I2C1_TX の DMA リクエストは、それぞれ DMA チャンネルの 7 と 6 に配置される)

ビット 26 **USART3_DMA_RMP**: USART3 の DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。

このビットは、ソフトウェアによってセット／クリアされます。USART3 の DMA リクエストの再配置を制御します。

0: 再配置なし (USART3_RX と USART3_TX の DMA リクエストは、それぞれ DMA チャンネルの 6 と 7 に配置される)

1: 再配置 (USART3_RX と USART3_TX の DMA リクエストは、それぞれ DMA チャンネルの 3 と 2 に配置される)

ビット 25 **USART2_DMA_RMP**: USART2 の DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。

このビットは、ソフトウェアによってセット／クリアされます。USART2 の DMA リクエストの再配置を制御します。

0: 再配置なし (USART2_RX と USART2_TX の DMA リクエストは、それぞれ DMA チャンネルの 5 と 4 に配置される)

1: 再配置 (USART2_RX と USART2_TX の DMA リクエストは、それぞれ DMA チャンネルの 6 と 7 に配置される)

ビット 24 **SPI2_DMA_RMP**: SPI2 の DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。
このビットは、ソフトウェアによってセット／クリアされます。SPI2 の DMA リクエストの再配置を制御します。

0: 再配置なし (SPI2_RX と SPI2_TX の DMA リクエストは、それぞれ DMA チャンネルの 4 と 5 に配置される)

1: 再配置 (SPI2_RX と SPI2_TX の DMA リクエストは、それぞれ DMA チャンネルの 6 と 7 に配置される)

- ビット 23:22 **I2C_PAx_FMP** : 高速モードプラス (FM+) 駆動機能有効化ビット。STM32F03x、STM32F04x、および STM32F09x デバイスでのみ使用可能。
- これらのビットは、ソフトウェアによってセット/クリアされます。各ビットは、PA10 および PA9 の I/O で I²C FM+ モードを有効にします。
- 0 : PAx ピンは標準モードで動作します。
 - 1 : PAx ピンで I²C FM+ モードを有効にし、速度制御をバイパスします。
- ビット 21 **I2C2_FMP** : I2C2 の FM+ 駆動機能の有効化。STM32F07x および STM32F09x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。このビットは、I2C_Pxx_FM+ ビットと論理和がとられます。
- 0 : FM+ モードは I2C_Pxx_FM+ ビットのみで制御されます。
 - 1 : FM+ モードは GPIOx_AFR レジスタの選択ビットを通じて選択されたすべての I2C2 ピンで有効化されます。これは、専用の I2C_Pxx_FM+ 制御ビットのないパッドで FM+ モードを有効化する唯一の方法です。
- ビット 20 **I2C1_FMP** : I2C1 の FM+ 駆動機能の有効化。STM32F05x デバイスでは使用できません。
- このビットは、ソフトウェアによってセット/クリアされます。このビットは、I2C_Pxx_FM+ ビットで論理和がとられます。
- 0 : FM+ モードは I2C_Pxx_FM+ ビットのみで制御されます。
 - 1 : FM+ モードは GPIOx_AFR レジスタの選択ビットを通じて選択されたすべての I2C1 ピンで有効化されます。これは、専用の I2C_Pxx_FM+ 制御ビットのないパッドで FM+ モードを有効化する唯一の方法です。
- ビット 19:16 **I2C_PBx_FMP** : 高速モードプラス (FM+) 駆動機能有効化ビット。
- これらのビットは、ソフトウェアによってセット/クリアされます。各ビットは、PB6、PB7、PB8、および PB9 の I/O で I²C FM+ モードを有効にします。
- 0 : PBx ピンは標準モードで動作します。
 - 1 : PBx ピンで I²C FM+ モードを有効にし、速度制御をバイパスします。
- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **TIM17_DMA_RMP2** : TIM17 のオルタネート DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。TIM17 のオルタネート DMA リクエストの再配置を制御します。
- 0 : オルタネート再配置なし (TIM17_DMA_RMP ビットに従って配置された TIM17 の DMA リクエスト)
 - 1 : オルタネート再配置 (DMA チャンネル 7 に配置された TIM17_CH1 と TIM17_UP の DMA リクエスト)
- ビット 13 **TIM16_DMA_RMP2** : TIM16 のオルタネート DMA リクエストの再配置ビット。STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。TIM16 のオルタネート DMA リクエストの再配置を制御します。
- 0 : オルタネート再配置なし (TIM16_DMA_RMP ビットに従って配置された TIM16 の DMA リクエスト)
 - 1 : オルタネート再配置 (DMA チャンネル 6 に配置された TIM16_CH1 と TIM16_UP の DMA リクエスト)
- ビット 12 **TIM17_DMA_RMP** : TIM17 の DMA リクエストの再配置ビット。STM32F03x、STM32F04x、STM32F05x、および STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。TIM17 の DMA リクエストの再配置を制御します。
- 0 : 再配置なし (TIM17_CH1 と TIM17_UP の DMA リクエストは DMA チャンネル 1 に配置される)
 - 1 : 再配置 (TIM17_CH1 と TIM17_UP の DMA リクエストは DMA チャンネル 2 に配置される)

- ビット 11 **TIM16_DMA_RMP**: TIM16 の DMA リクエストの再配置ビット。STM32F03x、STM32F04x、STM32F05x、および STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。TIM16 の DMA リクエストの再配置を制御します。
- 0: 再配置なし (TIM16_CH1 と TIM16_UP の DMA リクエストは DMA チャンネル 3 に配置される)
 - 1: 再配置 (TIM16_CH1 と TIM16_UP の DMA リクエストは DMA チャンネル 4 に配置される)
- ビット 10 **USART1_RX_DMA_RMP**: USART1_RX の DMA リクエストの再配置ビット。STM32F03x、STM32F04x、STM32F05x、および STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。USART1_RX の DMA リクエストの再配置を制御します。
- 0: 再配置なし (USART1_RX の DMA リクエストは DMA チャンネル 3 に配置される)
 - 1: 再配置 (USART1_RX の DMA リクエストは DMA チャンネル 5 に配置される)
- ビット 9 **USART1_TX_DMA_RMP**: USART1_TX の DMA リクエストの再配置ビット。STM32F03x、STM32F04x、STM32F05x、および STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。USART1_TX の DMA リクエストの再配置を制御します。
- 0: 再配置なし (USART1_TX の DMA リクエストは DMA チャンネル 2 に配置される)
 - 1: 再配置 (USART1_TX の DMA リクエストは DMA チャンネル 4 に配置される)
- ビット 8 **ADC_DMA_RMP**: ADC の DMA リクエストの再配置ビット。STM32F03x、STM32F04x、STM32F05x、および STM32F07x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。ADC の DMA リクエストの再配置を制御します。
- 0: 再配置なし (ADC の DMA リクエストは DMA チャンネル 1 に配置される)
 - 1: 再配置 (ADC の DMA リクエストは DMA チャンネル 2 に配置される)
- ビット 7:6 **IRDA_ENV_SEL[1:0]**: IRDA の変調エンベロープ信号ソース選択。STM32F09x デバイスでのみ使用できます。
- これらのビットにより、TIM16、USART4、および USART1 の間で変調エンベロープ信号を選択することができます。
- 00: TIMER16 選択
 - 01: USART1 選択
 - 10: USART4 選択
 - 11: 予約済み
- ビット 5 予約済みであり、リセット値に保持する必要があります。
- ビット 4 **PA11_PA12_RMP**: 小型のパッケージ (28 および 20 ピン) の PA11 と PA12 の再配置ビット。STM32F04x デバイスでのみ使用できます。
- このビットは、ソフトウェアによってセット/クリアされます。小型のピン数パッケージの PA9/10 または PA11/12 ピンのペアの配置を制御します。
- 0: 再配置なし (PA9/10 ピンのペアがピンに配置される)
 - 1: 再配置 (PA9/10 ではなく PA11/12 ピンのペアが配置される)
- ビット 3:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1:0 **MEM_MODE[1:0]**: メモリ配置選択ビット
- これらのビットは、ソフトウェアによってセット/クリアされます。これらは、アドレス 0x0000 0000 のメモリ内部配置を制御します。リセット後、これらのビットは、実際のブートモード設定で選択された値になります。詳細については、[章 2.5: ブート設定](#) を参照してください。
- x0: メインフラッシュメモリは 0x0000 0000 に配置されます。
 - 01: システムフラッシュメモリは 0x0000 0000 に配置されます。
 - 11: 内蔵 SRAM は 0x0000 0000 に配置されます。

10.1.2 SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1)

アドレスオフセット : 0x08

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定ビット (x= 0 ~ 3)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[x] ピン
x001 : PB[x] ピン
x010 : PC[x] ピン
x011 : PD[x] ピン
x100 : PE[x] ピン
x101 : PF[x] ピン
他の設定 : 予約済み

注 : 上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。

10.1.3 SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定ビット (x= 4 ~ 7)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

x000 : PA[x] ピン
x001 : PB[x] ピン
x010 : PC[x] ピン
x011 : PD[x] ピン
x100 : PE[x] ピン
x101 : PF[x] ピン
他の設定 : 予約済み

注 : 上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。



10.1.4 SYSCFG 外部割り込み設定レジスタ 3
(SYSCFG_EXTICR3)

アドレスオフセット：0x10
リセット値：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定ビット (x= 8 ~ 11)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[x] ピン
- x001 : PB[x] ピン
- x010 : PC[x] ピン
- x011 : PD[x] ピン
- x100 : PE[x] ピン
- x101 : PF[x] ピン

他の設定 : 予約済み

注： 上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。

10.1.5 SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4)

アドレスオフセット : 0x14

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **EXTIx[3:0]** : EXTI x 設定ビット (x= 12 ~ 15)

これらのビットは、EXTIx 外部割り込みのソース入力を選択するために、ソフトウェアで書き込みます。

- x000 : PA[x] ピン
- x001 : PB[x] ピン
- x010 : PC[x] ピン
- x011 : PD[x] ピン
- x100 : PE[x] ピン
- x101 : PF[x] ピン
- 他の設定 : 予約済み

注 : 上記レジスタの一部の I/O ピンは、小型のパッケージでは使用できない場合があります。

10.1.6 SYSCFG 設定レジスタ 2 (SYSCFG_CFGR2)

アドレスオフセット : 0x18

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM_PEF	Res.	Res.	Res.	Res.	Res.	PVD_LOCK	SRAM_PARITY_LOCK	LOCKUP_LOCK
							rc_w1						rw	rw	rw

ビット 31:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8 **SRAM_PEF** : SRAM パリティエラーフラグ

このビットは、SRAM パリティエラーが検出されたときに、ハードウェアによってセットされます。ソフトウェアで「1」を書き込むことによってクリアされます。

0 : SRAM パリティエラーは検出されていません。

1 : SRAM パリティエラーが検出されました。

ビット 7:3 予約済みであり、リセット値のままにしておかなければなりません。

ビット 2 **PVD_LOCK** : PVD ロックイネーブルビット

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。これは、TIM1/15/16/17 のブレーク入力への PVD 接続と、PWR_CR レジスタの PVDE と PLS[2:0] を有効にし、ロックするために使用できます。

0 : PVD 割り込みは、TIM1/15/16/17 のブレーク入力に接続されていません。PVDE および PLS[2:0] ビットは、アプリケーションでプログラムできます。

1 : PVD 割り込みは、TIM1/15/16/17 のブレーク入力に接続されています。PVDE および PLS[2:0] ビットは読み出し専用。

ビット 1 **SRAM_PARITY_LOCK** : SRAM パリティロックビット

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。これは、SRAM パリティエラー信号のTIM1/15/16/17 のブレーク入力への接続を有効にし、ロックするために使用できます。

0 : SRAM パリティエラーはTIM1/15/16/17 のブレーク入力から切断されています

1 : SRAM パリティエラーはTIM1/15/16/17 のブレーク入力に接続されています

ビット 0 **LOCKUP_LOCK** : Cortex-M0 LOCKUP ビットイネーブルビット

このビットは、ソフトウェアでセットされ、システムリセットでクリアされます。これは、Cortex-M0 LOCKUP (ハードフォールト) 出力のTIM1/15/16/17 ブレーク入力への接続を有効にし、ロックするために使用できます。

0 : Cortex-M0 LOCKUP 出力はTIM1/15/16/17 のブレーク入力から切断されています

1 : Cortex-M0 LOCKUP 出力はTIM1/15/16/17 のブレーク入力に接続されています

10.1.7 SYSCFG 割り込みライン 0 ステータスレジスタ (SYSCFG_ITLINE0)

STM32F09x には専用のレジスタのセットが実装されており、各割り込みラインに関連付けられたすべての保留中の割り込みソースを1つのレジスタへ収集します。これにより、割り込みラインに複数のソースが関連付けられている場合に、どのペリフェラルで処理が必要かを1回の読み出しでチェックできます。

これらのレジスタのビットはすべて読み出し専用で、対応する割り込みリクエストが保留中の場合にハードウェアによってセットされ、ペリフェラルレジスタで割り込みソースフラグをリセットすることでクリアされます。

アドレスオフセット : 80h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **WWDG** : ウィンドウ型ウォッチドッグ割り込み保留フラグ

10.1.8 SYSCFG 割り込みライン 1 ステータスレジスタ (SYSCFG_ITLINE1)

アドレスオフセット : 84h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VDDIO2	PVDOUT
														r	r

ビット 31:2 予約済み (0 として読み出される)

ビット 1 **VDDIO2** : VDDIO2 電源監視の割り込みリクエストの保留 (EXTI ライン 31)

ビット 0 **PVDOUT** : PVD 電源監視の割り込みリクエストの保留 (EXTI ライン 16)。このビットは STM32F0x8 では使用できません。

10.1.9 SYSCFG 割り込みライン 2 ステータスレジスタ (SYSCFG_ITLINE2)

アドレスオフセット : 88h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_ALRA	RTC_TSTAMP	RTC_WAKEUP
													r	r	r

ビット 31:3 予約済み (0 として読み出される)

ビット 2 **RTC_ALRA** : RTC アラームの割り込みリクエストの保留 (EXTI ライン 17)

ビット 1 **RTC_TSTAMP** : RTC タンパおよびタイムスタンプの割り込みリクエストの保留 (EXTI ライン 19)

ビット 0 **RTC_WAKEUP** : RTC ウェイクアップの割り込みリクエストの保留 (EXTI ライン 20)

10.1.10 SYSCFG 割り込みライン 3 ステータスレジスタ (SYSCFG_ITLINE3)

アドレスオフセット : 8Ch

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLASH_ITF
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **FLASH_ITF** : フラッシュインタフェースの割り込みリクエストの保留

10.1.11 SYSCFG 割り込みライン 4 ステータスレジスタ (SYSCFG_ITLINE4)

アドレスオフセット : 90h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRS	RCC
														r	r

ビット 31:2 予約済み (0 として読み出される)

ビット 1 **CRS** : クロックリカバリシステムの割り込みリクエストの保留

ビット 0 **RCC** : リセットおよびクロック制御の割り込みリクエストの保留

10.1.12 SYSCFG 割り込みライン 5 ステータスレジスタ (SYSCFG_ITLINE5)

アドレスオフセット : 94h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI1	EXTI0
														r	r

ビット 31:2 予約済み (0 として読み出される)

ビット 1 **EXTI1** : EXTI ライン 1 割り込みリクエストの保留

ビット 0 **EXTI0** : EXTI ライン 0 割り込みリクエストの保留

10.1.13 SYSCFG 割り込みライン 6 ステータスレジスタ (SYSCFG_ITLINE6)

アドレスオフセット : 98h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3	EXTI2
														r	r

ビット 31:2 予約済み (0 として読み出される)

ビット 1 **EXTI3** : EXTI ライン 3 割り込みリクエストの保留

ビット 0 **EXTI2** : EXTI ライン 2 割り込みリクエストの保留



10.1.14 SYSCFG 割り込みライン 7 ステータスレジスタ (SYSCFG_ITLINE7)

アドレスオフセット : 9Ch

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	EXTI15	EXTI14	EXTI13	EXTI12	EXTI11	EXTI10	EXTI9	EXTI8	EXTI7	EXTI6	EXTI5	EXTI4
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:10 予約済み (0 として読み出される)

ビット 11 **EXTI15** : EXTI ライン 15 割り込みリクエストの保留

ビット 10 **EXTI14** : EXTI ライン 14 割り込みリクエストの保留

ビット 9 **EXTI13** : EXTI ライン 13 割り込みリクエストの保留

ビット 8 **EXTI12** : EXTI ライン 12 割り込みリクエストの保留

ビット 7 **EXTI11** : EXTI ライン 11 割り込みリクエストの保留

ビット 6 **EXTI10** : EXTI ライン 10 割り込みリクエストの保留

ビット 5 **EXTI9** : EXTI ライン 9 割り込みリクエストの保留

ビット 4 **EXTI8** : EXTI ライン 8 割り込みリクエストの保留

ビット 3 **EXTI7** : EXTI ライン 7 割り込みリクエストの保留

ビット 2 **EXTI6** : EXTI ライン 6 割り込みリクエストの保留

ビット 1 **EXTI5** : EXTI ライン 5 割り込みリクエストの保留

ビット 0 **EXTI4** : EXTI ライン 4 割り込みリクエストの保留

10.1.15 SYSCFG 割り込みライン 8 ステータスレジスタ (SYSCFG_ITLINE8)

アドレスオフセット : A0h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCS_EOA	TCS_MCE
														r	r

ビット 31:2 予約済み (0 として読み出される)

ビット 1 **TCS_EOA** : タッチセンシングコントローラの取得完了時の割り込みリクエストの保留

ビット 0 **TCS_MCE** : タッチセンシングコントローラの最大カウントエラーの割り込みリクエストの保留

10.1.16 SYSCFG 割り込みライン 9 ステータスレジスタ (SYSCFG_ITLINE9)

アドレスオフセット : A4h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_CH1
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **DMA1_CH1** : DMA1 チャンネル 1 割り込みリクエストの保留

10.1.17 SYSCFG 割り込みライン 10 ステータスレジスタ (SYSCFG_ITLINE10)

アドレスオフセット : A8h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH2	DMA2_CH1	DMA1_CH3	DMA1_CH2
												r	r	r	r

ビット 31:4 予約済み (0 として読み出される)

ビット 3 **DMA2_CH2** : DMA2 チャンネル 2 割り込みリクエストの保留

ビット 2 **DMA2_CH1** : DMA2 チャンネル 1 割り込みリクエストの保留

ビット 1 **DMA1_CH3** : DMA1 チャンネル 3 割り込みリクエストの保留

ビット 0 **DMA1_CH2** : DMA1 チャンネル 2 割り込みリクエストの保留

10.1.18 SYSCFG 割り込みライン 11 ステータスレジスタ (SYSCFG_ITLINE11)

アドレスオフセット : ACh

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH5	DMA2_CH4	DMA2_CH3	DMA1_CH7	DMA1_CH6	DMA1_CH5	DMA1_CH4
									r	r	r	r	r	r	r

ビット 31:7 予約済み (0 として読み出される)

ビット 6 **DMA2_CH5** : DMA2 チャンネル 5 割り込みリクエストの保留

ビット 5 **DMA2_CH4** : DMA2 チャンネル 4 割り込みリクエストの保留

ビット 4 **DMA2_CH3** : DMA2 チャンネル 3 割り込みリクエストの保留

ビット 3 **DMA1_CH7** : DMA1 チャンネル 7 割り込みリクエストの保留

ビット 2 **DMA1_CH6** : DMA1 チャンネル 6 割り込みリクエストの保留

ビット 1 **DMA1_CH5** : DMA1 チャンネル 5 割り込みリクエストの保留

ビット 0 **DMA1_CH4** : DMA1 チャンネル 4 割り込みリクエストの保留

10.1.19 SYSCFG 割り込みライン 12 ステータスレジスタ (SYSCFG_ITLINE12)

アドレスオフセット : B0h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP2	COMP1	ADC
													r	r	r

ビット 31:3 予約済み (0 として読み出される)

ビット 2 **COMP2** : コンパレータ 2 割り込みリクエストの保留 (EXTI ライン 22)

ビット 1 **COMP1** : コンパレータ 1 割り込みリクエストの保留 (EXTI ライン 21)

ビット 0 **ADC** : ADC 割り込みリクエストの保留

10.1.20 SYSCFG 割り込みライン 13 ステータスレジスタ (SYSCFG_ITLINE13)

アドレスオフセット : B4h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_BRK	TIM1_UPD	TIM1_TRG	TIM1_CC
												r	r	r	r

ビット 31:4 予約済み (0 として読み出される)

ビット 3 **TIM1_BRK** : タイマ 1 ブレーク割り込みリクエストの保留

ビット 2 **TIM1_UPD** : タイマ 1 更新割り込みリクエストの保留

ビット 1 **TIM1_TRG** : タイマ 1 トリガ割り込みリクエストの保留

ビット 0 **TIM1_CC** : タイマ 1 コミュレーション割り込みリクエストの保留

10.1.21 SYSCFG 割り込みライン 14 ステータスレジスタ (SYSCFG_ITLINE14)

アドレスオフセット : B8h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_CC
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **TIM1_CC** : タイマ 1 キャプチャ/比較割り込みリクエストの保留

10.1.22 SYSCFG 割り込みライン 15 ステータスレジスタ (SYSCFG_ITLINE15)

アドレスオフセット : BCh

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM2
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **TIM2** : タイマ 2 割り込みリクエストの保留

10.1.23 SYSCFG 割り込みライン 16 ステータスレジスタ (SYSCFG_ITLINE16)

アドレスオフセット : C0h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **TIM3** : タイマ 3 割り込みリクエストの保留

10.1.24 SYSCFG 割り込みライン 17 ステータスレジスタ (SYSCFG_ITLINE17)

アドレスオフセット : C4h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DAC	TIM6
														r	r

ビット 31:1 予約済み (0 として読み出される)

ビット 1 **DAC** : DAC アンダーランの割り込みリクエストの保留

ビット 0 **TIM6** : タイマ 6 割り込みリクエストの保留

10.1.25 SYSCFG 割り込みライン 18 ステータスレジスタ (SYSCFG_ITLINE18)

アドレスオフセット : C8h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM7
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 TIM7 : タイマ 7 割り込みリクエストの保留

10.1.26 SYSCFG 割り込みライン 19 ステータスレジスタ (SYSCFG_ITLINE19)

アドレスオフセット : CCh

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM14
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 TIM14 : タイマ 14 割り込みリクエストの保留

10.1.27 SYSCFG 割り込みライン 20 ステータスレジスタ (SYSCFG_ITLINE20)

アドレスオフセット : D0h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM15
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 TIM15 : タイマ 15 割り込みリクエストの保留

10.1.28 SYSCFG 割り込みライン 21 ステータスレジスタ (SYSCFG_ITLINE21)

アドレスオフセット : D4h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM16
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **TIM16** : タイマ 16 割り込みリクエストの保留

10.1.29 SYSCFG 割り込みライン 22 ステータスレジスタ (SYSCFG_ITLINE22)

アドレスオフセット : D8h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **TIM17** : タイマ 17 割り込みリクエストの保留

10.1.30 SYSCFG 割り込みライン 23 ステータスレジスタ (SYSCFG_ITLINE23)

アドレスオフセット : DCh

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C1
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **I2C1** : EXTI ライン 23 と組み合わせられた I2C1 割り込みリクエストの保留

10.1.31 SYSCFG 割り込みライン 24 ステータスレジスタ (SYSCFG_ITLINE24)

アドレスオフセット : E0h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C2
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **I2C2** : I2C2 割り込みリクエストの保留

10.1.32 SYSCFG 割り込みライン 25 ステータスレジスタ (SYSCFG_ITLINE25)

アドレスオフセット : E4h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI1
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **SPI1** : SPI1 割り込みリクエストの保留

10.1.33 SYSCFG 割り込みライン 26 ステータスレジスタ (SYSCFG_ITLINE26)

アドレスオフセット : E8h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI2
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **SPI2** : SPI2 割り込みリクエストの保留



10.1.34 SYSCFG 割り込みライン 27 ステータスレジスタ (SYSCFG_ITLINE27)

アドレスオフセット : ECh

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART1
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **USART1** : EXTI ライン 25 と組み合わせられた USART1 割り込みリクエストの保留

10.1.35 SYSCFG 割り込みライン 28 ステータスレジスタ (SYSCFG_ITLINE28)

アドレスオフセット : F0h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART2
															r

ビット 31:1 予約済み (0 として読み出される)

ビット 0 **USART2** : EXTI ライン 26 と組み合わせられた USART2 割り込みリクエストの保留

10.1.36 SYSCFG 割り込みライン 29 ステータスレジスタ (SYSCFG_ITLINE29)

アドレスオフセット : F4h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART8	USART7	USART6	USART5	USART4	USART3
										r	r	r	r	r	r

ビット 31:6 予約済み (0 として読み出される)

ビット 5 **USART8** : USART8 割り込みリクエストの保留

ビット 4 **USART7** : USART7 割り込みリクエストの保留

ビット 3 **USART6** : USART6 割り込みリクエストの保留

ビット 2 **USART5** : USART5 割り込みリクエストの保留

ビット 1 **USART4** : USART4 割り込みリクエストの保留

ビット 0 **USART3** : EXTI ライン 28 と組み合わせられた USART3 割り込みリクエストの保留

10.1.37 SYSCFG 割り込みライン 30 ステータスレジスタ (SYSCFG_ITLINE30)

アドレスオフセット : F8h

システムリセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAN	CEC
														r	r

ビット 31:2 予約済み (0 として読み出される)

ビット 1 **CAN** : CAN 割り込みリクエストの保留

ビット 0 **CEC** : EXTI ライン 27 と組み合わせられた CEC 割り込みリクエストの保留

10.1.38 SYSCFG レジスタマップ

次の表に、SYSCFG レジスタマップとリセット値を示します。

表 26. SYSCFG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	SYSCFG_CFGR1	Res.	TIM3_DMA_RMP	TIM2_DMA_RMP	TIM1_DMA_RMP	I2C1_DMA_RMP	USART3_DMA_RMP	USART2_DMA_RMP	SPI2_DMA_RMP	I2C_PA10_FMP	I2C_PA9_FMP	I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP	Res.	TIM17_DMA_RMP2	TIM16_DMA_RMP2	TIM17_DMA_RMP	TIM16_DMA_RMP	USART1_RX_DMA_RMP	USART1_TX_DMA_RMP	ADC_DMA_RMP	IRDA_ENV_SEL		Res.	PA11_PA12_RMP	Res.				MEM_MODE[1:0]
	リセット値		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0			0			X	X	
0x08	SYSCFG_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[3:0]			EXTI2[3:0]			EXTI1[3:0]			EXTI0[3:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	SYSCFG_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7[3:0]			EXTI6[3:0]			EXTI5[3:0]			EXTI4[3:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	SYSCFG_EXTICR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11[3:0]			EXTI10[3:0]			EXTI9[3:0]			EXTI8[3:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	SYSCFG_EXTICR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15[3:0]			EXTI14[3:0]			EXTI13[3:0]			EXTI12[3:0]							
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	SYSCFG_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM_PEF	Res.	Res.	Res.	Res.	Res.	PVD_LOCK	SRAM_PARITY_LOCK	LOCKUP_LOCK		
	リセット値																							0						0	0	0		

表 27. STM32F09x デバイスの SYSCFG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1Dh から 7Fh	予約済み	予約済み																															
	リセット値																																
80h	SYSCFG_ITLINE0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
84h	SYSCFG_ITLINE1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VDDIO2	PVDOUT
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
88h	SYSCFG_ITLINE2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_ALRA	RTC_TSTAMP	RTC_WAKEUP
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



表 27. STM32F09x デバイスの SYSCFG レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
8Ch	SYSCFG_ITLINE3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLASH_ITF		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
90h	SYSCFG_ITLINE4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRS	RCC		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
94h	SYSCFG_ITLINE5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXT11	EXT10		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
98h	SYSCFG_ITLINE6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXT13	EXT12		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
9Ch	SYSCFG_ITLINE7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXT115	EXT114	EXT113	EXT112	EXT111	EXT110	EXT109	EXT108	EXT107	EXT106	EXT105	EXT104		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
A0h	SYSCFG_ITLINE8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCS_EOA	TCS_MCE		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
A4h	SYSCFG_ITLINE9	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA1_CH1		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
A8h	SYSCFG_ITLINE10	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH2	DMA2_CH1		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ACh	SYSCFG_ITLINE11	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH5	DMA2_CH4	DMA2_CH3	DMA2_CH2	DMA2_CH1	DMA2_CH0	DMA2_CHF		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
B0h	SYSCFG_ITLINE12	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH3	DMA2_CH2	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
B4h	SYSCFG_ITLINE13	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH3	DMA2_CH2	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
B8h	SYSCFG_ITLINE14	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH3	DMA2_CH2	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
BCh	SYSCFG_ITLINE15	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH3	DMA2_CH2
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
C0h	SYSCFG_ITLINE16	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA2_CH3	DMA2_CH2
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

[illegible]

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などに付きましては如何なる責任にも負いません。

11 ダイレクトメモリアクセスコントローラ (DMA)

11.1 概要

DMA (Direct Memory Access : ダイレクトメモリアクセスコントローラ) は、ペリフェラルとメモリ間、およびメモリ間で高速データ転送を行うために使用します。DMA により、CPU を動作させずに、データを高速で移動することができます。これにより、CPU リソースを他の操作のためにあけておくことができます。

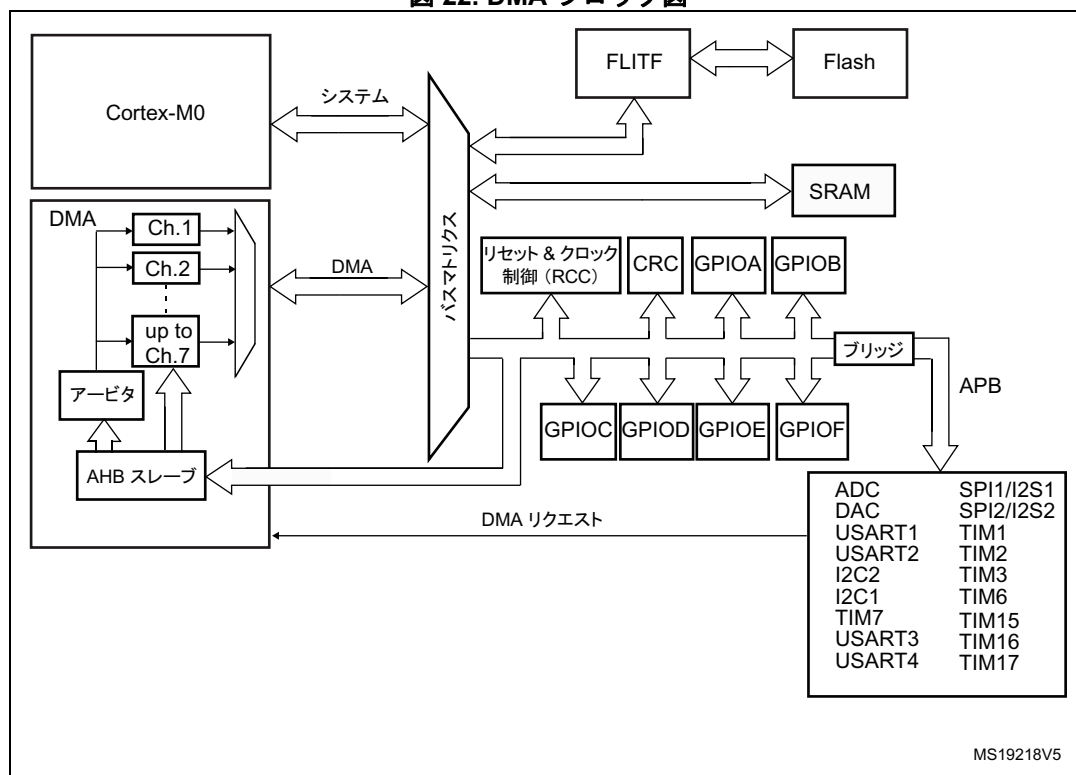
DMA コントローラは、12 本までのチャンネルを搭載し、それぞれが 1 つ以上のペリフェラルからのメモリアセスリクエストを管理する役割を担っています。また、DMA リクエスト間の優先順位を操作するためのアービタも内蔵しています。

11.2 DMA の主な機能

- 最大 7 本の個別に設定可能なチャンネル (リクエスト) を DMA 上に内蔵しています。
- STM32F09x は 5 本の個別に設定可能な追加のチャンネル (リクエスト) を DMA2 上で提供しています。
- 各チャンネルは、専用のハードウェア DMA リクエストに接続されており、ソフトウェアトリガもサポートしています。この設定はソフトウェアで行われます。
- DMA チャンネルからのリクエスト間の優先順位はソフトウェアでプログラムできます (最高、高、中、低の 4 レベル)。またはレベルが等しい場合はハードウェアによって決まります (リクエスト 1 はリクエスト 2 よりも優先、など)。
- 転送元および転送先の転送サイズ (バイト、ハーフワード、ワード) は個別に設定され、パッキング／アンパッキングをエミュレートします。転送元／転送先のアドレスは、データサイズに基づいて整列させてください。
- サークュラバッファ管理をサポートします。
- 各チャンネルに対する単一割り込みリクエストにおいて、3 つのイベントフラグ (DMA 1/2 転送、DMA 転送完了、DMA 転送エラー) の論理和がとられます。
- メモリ間の転送
- ペリフェラルからメモリ、メモリからペリフェラル、およびペリフェラル間の転送
- フラッシュ、SRAM、APB、AHB ペリフェラルに対して、転送元および転送先としてアクセス
- プログラム可能な転送データ数 : 最大 65535

以下にブロック図を示します。

図 22. DMA ブロック図



11.3 DMA の機能説明

DMA コントローラは、システムバスを Cortex[®]-M0 コアと共有することでダイレクトメモリアクセスを行います。DMA リクエストは、CPU と DMA のターゲット転送先が同じである場合に（メモリまたはペリフェラル）、多少のバスサイクルの間、CPU のシステムバスへのアクセスを停止することがあります。バスマトリックスはラウンドロビンスケジューリングを実装します。これにより、CPU のシステムバス帯域幅（メモリとペリフェラルの両方への）の少なくとも半分を確保できます。

11.3.1 DMA トランザクション

イベントの後、ペリフェラルは DMA コントローラにリクエスト信号を送信します。DMA コントローラは、チャネルの優先順位に応じて、リクエストを処理します。DMA コントローラがペリフェラルにアクセスするとすぐに、DMA コントローラはペリフェラルに確認応答を送信します。ペリフェラルは、DMA コントローラからの確認応答を受け取るとすぐに、そのリクエストを解除します。ペリフェラルによってリクエストが無効にされると、DMA コントローラは確認応答を解除します。さらにリクエストがある場合、ペリフェラルは次のトランザクションを開始できます。

要約すると、各 DMA 転送は次の 3 つの動作で構成されています。

- ペリフェラルデータレジスタまたは現在の内部ペリフェラル／メモリアドレスレジスタを介してアドレス指定されたメモリ位置からのデータのロード。最初の転送に使用される開始アドレスは、DMA_CPARx または DMA_CMARx レジスタでプログラムされたペリフェラル／メモリのベースアドレスです。

- ペリフェラルデータレジスタまたは現在の内部ペリフェラル／メモリアドレスレジスタを介してアドレス指定されたメモリ位置にロードされたデータの格納。最初の転送に使用される開始アドレスは、DMA_CPARx または DMA_CMARx レジスタでプログラムされたペリフェラル／メモリのベースアドレスです。
- まだ実行される予定のトランザクションの数を保持している DMA_CNDTRx レジスタのポストデクリメント。

11.3.2 アービタ

アービタは、チャネルリクエストをその優先順位に基づいて管理し、ペリフェラル／メモリアクセスシーケンスを起動します。

優先順位は 2 段階で管理されます。

- ソフトウェア：各チャネルの優先順位は、DMA_CCRx レジスタで設定できます。4 つのレベルがあります。
 - － 最優先
 - － 高優先
 - － 中優先
 - － 低優先
- ハードウェア：2 つのリクエストのソフトウェア優先順位レベルが同じである場合、小さな番号のチャネルが大きな番号のチャネルよりも優先されます。たとえば、チャネル 2 はチャネル 4 よりも優先されます。

11.3.3 DMA チャネル

各チャネルは、ある固定アドレスにあるペリフェラルレジスタとメモリアドレスの間の DMA 転送を処理します。転送データ数（最大 65535）はプログラム可能です。転送データ項目数を格納しているレジスタは、トランザクションの終了ごとにデクリメントされます。

プログラム可能なデータサイズ

ペリフェラルおよびメモリの転送データサイズは、DMA_CCRx レジスタの PSIZE および MSIZE ビットを介して完全にプログラム可能です。

ポインタのインクリメント

ペリフェラルおよびメモリのポインタは、DMA_CCRx レジスタの PINC および MINC ビットに応じて、各転送後に任意で自動的にポストインクリメントすることができます。インクリメントモードを有効にした場合、次の転送アドレスは、選択されたデータサイズに応じて 1、2、または 4 だけインクリメントされた前回の転送アドレスとなります。最初の転送アドレスは、DMA_CPARx/DMA_CMARx レジスタでプログラムされたアドレスとなります。転送動作中は、これらのレジスタは最初にプログラムされた値を保持します。現在の転送アドレス（現在の内部ペリフェラル／メモリアドレスレジスタ内）に対するソフトウェアによるアクセスはできません。

チャネルがノンサーキュラモードに設定されている場合、転送終了後（転送すべきデータ項目数がゼロに達したとき）、DMA リクエストは処理されません。新たな転送すべきデータ項目数を DMA_CNDTRx レジスタに再ロードするには、DMA チャネルを無効にする必要があります。

注： *DMA チャネルが無効にされると、DMA レジスタはリセットされません。DMA チャネルレジスタ (DMA_CCRx、DMA_CPARx、DMA_CMARx) は、チャネル設定フェーズでプログラムされた初期値のままです。*



サーキュラモードでは、転送終了後、DMA_CNDTRx レジスタには最初にプログラムした値が自動的に再ロードされます。現在の内部アドレスレジスタには、DMA_CPARx/DMA_CMARx レジスタからのベースアドレス値が再ロードされます。

チャンネル設定手順

DMA チャンネル x (x はチャンネル番号) の設定は、以下のシーケンスで行います。

1. DMA_CPARx レジスタに、ペリフェラルレジスタアドレスをセットします。ペリフェラルイベント後は、メモリとの転送に使用されるこのアドレスとデータをやり取りします。
2. DMA_CMARx レジスタにメモリアドレスをセットします。ペリフェラルイベント後は、このメモリに対してデータを読み書きします。
3. 転送すべきデータの総数を DMA_CNDTRx レジスタに設定します。毎回のペリフェラルイベント後に、この値はデクリメントされます。
4. DMA_CCRx レジスタの PL[1:0] ビットを使用して、チャンネルの優先順位を設定します。
5. データ転送方向、サーキュラモード、ペリフェラルとメモリのインクリメントモード、ペリフェラルとメモリのデータサイズ、1/2 転送やフル転送後の割り込みを DMA_CCRx レジスタに設定します。
6. DMA_CCRx レジスタの ENABLE ビットをセットすることでチャンネルを有効にします。

コード例については、「付録」セクション [A.5.1 : DMA チャンネル設定シーケンスのコード例](#) を参照してください。

チャンネルが有効になると、そのチャンネルに接続されているペリフェラルからの DMA リクエストを処理することができます。

バイトの半分が転送されると、1/2 転送フラグ (HTIF) がセットされ、1/2 転送割り込みイネーブルビット (HTIE) がセットされている場合には、割り込みが生成されます。転送が終了すると、転送完了フラグ (TCIF) がセットされ、転送完了割り込みイネーブルビット (TCIE) がセットされている場合には、割り込みが生成されます。

サーキュラモード

サーキュラモードを使用すると、サーキュラバッファや連続したデータフロー (ADC スキャンモードなど) を処理することができます。この機能は、DMA_CCRx レジスタの CIRC ビットを使用して有効にできます。サーキュラモードが有効になると、転送されるデータ項目数はチャンネル設定フェーズでプログラムされた初期値が自動的に再ロードされ、DMA リクエストの処理が続行されます。

メモリ間モード

DMA チャンネルは、ペリフェラルからのリクエストによってトリガされなくても機能します。このモードは、メモリ間モードと呼ばれます。

DMA_CCRx レジスタの MEM2MEM ビットがセットされると、チャンネルは、ソフトウェアが DMA_CCRx レジスタの Enable ビット (EN) をセットすることによってそのチャンネルを有効にするとすぐに、転送を開始します。DMA_CNDTRx レジスタがゼロに達すると、転送は停止します。メモリ間モードとサーキュラモードを同時に使用することはできません。

11.3.4 プログラム可能なデータ幅、データの整列、およびエンディアン

PSIZE と MSIZE が等しくない場合、DMA は、表 28 : プログラム可能なデータ幅およびエンディアンの動作 (ビット PINC = MINC = 1 の場合) に記載されているとおりに必要なデータの整列を行います。

表 28. プログラム可能なデータ幅およびエンディアンの動作 (ビット PINC = MINC = 1 の場合)

転送元 ポートの幅	転送先 ポートの幅	転送する データ 項目の数 (NDT)	転送元の内容: アドレス/データ	転送動作	転送先の内容: アドレス/データ
8	8	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1 : B0[7:0] を 0x0 で読み出し、次に B0[7:0] を 0x0 に書き込みます。 2 : B1[7:0] を 0x1 で読み出し、次に B1[7:0] を 0x1 に書き込みます。 3 : B2[7:0] を 0x2 で読み出し、次に B2[7:0] を 0x2 に書き込みます。 4 : B3[7:0] を 0x3 で読み出し、次に B3[7:0] を 0x3 に書き込みます。	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3
8	16	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1 : B0[7:0] を 0x0 で読み出し、次に 00B0[15:0] を 0x0 に書き込みます。 2 : B1[7:0] を 0x1 で読み出し、次に 00B1[15:0] を 0x2 に書き込みます。 3 : B3[7:0] を 0x2 で読み出し、次に 00B2[15:0] を 0x4 に書き込みます。 4 : B4[7:0] を 0x3 で読み出し、次に 00B3[15:0] を 0x6 に書き込みます。	@0x0 / 00B0 @0x2 / 00B1 @0x4 / 00B2 @0x6 / 00B3
8	32	4	@0x0 / B0 @0x1 / B1 @0x2 / B2 @0x3 / B3	1 : B0[7:0] を 0x0 で読み出し、次に 000000B0[31:0] を 0x0 に書き込みます。 2 : B1[7:0] を 0x1 で読み出し、次に 000000B1[31:0] を 0x4 に書き込みます。 3 : B3[7:0] を 0x2 で読み出し、次に 000000B2[31:0] を 0x8 に書き込みます。 4 : B4[7:0] を 0x3 で読み出し、次に 000000B3[31:0] を 0xC に書き込みます。	@0x0 / 000000B0 @0x4 / 000000B1 @0x8 / 000000B2 @0xC / 000000B3
16	8	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1 : B1B0[15:0] を 0x0 で読み出し、次に B0[7:0] を 0x0 に書き込みます。 2 : B3B2[15:0] を 0x2 で読み出し、次に B2[7:0] を 0x1 に書き込みます。 3 : B5B4[15:0] を 0x4 で読み出し、次に B4[7:0] を 0x2 に書き込みます。 4 : B7B6[15:0] を 0x6 で読み出し、次に B6[7:0] を 0x3 に書き込みます。	@0x0 / B0 @0x1 / B2 @0x2 / B4 @0x3 / B6
16	16	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1 : B1B0[15:0] を 0x0 で読み出し、次に B1B0[15:0] を 0x0 に書き込みます。 2 : B3B2[15:0] を 0x2 で読み出し、次に B3B2[15:0] を 0x2 に書き込みます。 3 : B5B4[15:0] を 0x4 で読み出し、次に B5B4[15:0] を 0x4 に書き込みます。 4 : B7B6[15:0] を 0x6 で読み出し、次に B7B6[15:0] を 0x6 に書き込みます。	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6
16	32	4	@0x0 / B1B0 @0x2 / B3B2 @0x4 / B5B4 @0x6 / B7B6	1 : B1B0[15:0] を 0x0 で読み出し、次に 0000B1B0[31:0] を 0x0 に書き込みます。 2 : B3B2[15:0] を 0x2 で読み出し、次に 0000B3B2[31:0] を 0x4 に書き込みます。 3 : B5B4[15:0] を 0x4 で読み出し、次に 0000B5B4[31:0] を 0x8 に書き込みます。 4 : B7B6[15:0] を 0x6 で読み出し、次に 0000B7B6[31:0] を 0xC に書き込みます。	@0x0 / 0000B1B0 @0x4 / 0000B3B2 @0x8 / 0000B5B4 @0xC / 0000B7B6
32	8	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1 : B3B2B1B0[31:0] を 0x0 で読み出し、次に B0[7:0] を 0x0 に書き込みます。 2 : B7B6B5B4[31:0] を 0x4 で読み出し、次に B4[7:0] を 0x1 に書き込みます。 3 : BBBAB9B8[31:0] を 0x8 で読み出し、次に B8[7:0] を 0x2 に書き込みます。 4 : BFBEBDBC[31:0] を 0xC で読み出し、次に BC[7:0] を 0x3 に書き込みます。	@0x0 / B0 @0x1 / B4 @0x2 / B8 @0x3 / BC
32	16	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1 : B3B2B1B0[31:0] を 0x0 で読み出し、次に B1B0[7:0] を 0x0 に書き込みます。 2 : B7B6B5B4[31:0] を 0x4 で読み出し、次に B5B4[7:0] を 0x1 に書き込みます。 3 : BBBAB9B8[31:0] を 0x8 で読み出し、次に B9B8[7:0] を 0x2 に書き込みます。 4 : BFBEBDBC[31:0] を 0xC で読み出し、次に BDBC[7:0] を 0x3 に書き込みます。	@0x0 / B1B0 @0x2 / B5B4 @0x4 / B9B8 @0x6 / BDBC
32	32	4	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC	1 : B3B2B1B0[31:0] を 0x0 で読み出し、次に B3B2B1B0[31:0] を 0x0 に書き込みます。 2 : B7B6B5B4[31:0] を 0x4 で読み出し、次に B7B6B5B4[31:0] を 0x4 に書き込みます。 3 : BBBAB9B8[31:0] を 0x8 で読み出し、次に BBBAB9B8[31:0] を 0x8 に書き込みます。 4 : BFBEBDBC[31:0] を 0xC で読み出し、次に BFBEBDBC[31:0] を 0xC に書き込みます。	@0x0 / B3B2B1B0 @0x4 / B7B6B5B4 @0x8 / BBBAB9B8 @0xC / BFBEBDBC

バイトまたはハーフワード書き込み動作をサポートしていない AHB ペリフェラルへの対応

DMA が AHB のバイトまたはハーフワード書き込み動作を起動すると、データは HWDATA[31:0] バスの未使用のレーンに複製されます。そのため、使用中の AHB スレーブペリフェラルがバイト/ハーフワード書き込み動作をサポートしておらず (HSIZE がペリフェラルで使用されていない)、かつエラーが 1 件も出ていない場合は、以下の2つの例に示されるように、DMA は 32 HWDATA ビットを書き込みます。

- ハーフワード “0xABCD” を書き込むには、DMA は HWDATA バスを “0xABCDABCD” (HSIZE = ハーフワード) にセットします。
- バイト “0xAB” を書き込むには、DMA は HWDATA バスを “0xABABABAB” (HSIZE = バイト) にセットします。

AHB/APB ブリッジが AHB 32 ビットスレーブペリフェラルである (HSIZE の値は考慮せず) と仮定した場合、以下のようなやり方で、AHB バイトまたはハーフワードのあらゆる動作を 32 ビット APB 動作に変換します。

- データ “0xB0” を 0x0 (または 0x1, 0x2, 0x3) に書き込む AHB バイト書き込み動作は、データ “0xB0B0B0B0” を 0x0 に書き込む APB ワード書き込み動作に変換されます。
- データ “0xB1B0” を 0x0 (または 0x2) に書き込む AHB ハーフワード書き込み動作は、データ “0xB1B0B1B0” を 0x0 に書き込む APB ワード書き込み動作に変換されます。

たとえば、APB バックアップレジスタ (32 ビットアドレス境界に整列された 16 ビットレジスタ) に書き込みたい場合は、転送元のメモリサイズ (MSIZE) を 16 ビットに、転送先のペリフェラルサイズ (PSIZE) を 32 ビットに設定する必要があります。

11.3.5 エラー管理

DMA 転送エラーは、予約済みアドレス空間に対する読み出しや書き込みによって発生する可能性があります。DMA 読み出しまたは書き込みアクセス中に DMA 転送エラーが発生した場合、障害のあるチャンネルは、ハードウェアが対応するチャンネル設定レジスタ (DMA_CCRx) の該当する EN ビットをクリアすることにより、自動的に無効化されます。DMA_IFR レジスタのチャンネルの転送エラー割り込みフラグ (TEIF) がセットされ、DMA_CCRx レジスタの転送エラー割り込みイネーブルビット (TEIE) がセットされている場合には、割り込みが生成されます。

11.3.6 DMA 割り込み

割り込みは、DMA チャンネルごとの 1/2 転送、転送完了、転送エラーに対して生成されます。高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 29. DMA 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
1/2 転送	HTIF	HTIE
転送完了	TCIF	TCIE
転送エラー	TEIF	TEIE



11.3.7 DMA リクエストマッピング

DMA コントローラ

ペリフェラル (TIMx、ADC、DAC、SPI、I2C、および USARTx) からのハードウェアリクエストは、DMA に入る前に単純に論理和が取られます。これは、1 つのチャンネルでは、一度に 1 つのリクエストのみ有効にする必要があることを意味しています。

ペリフェラル DMA リクエストは、対応するペリフェラルのレジスタの DMA 制御ビットをプログラムすることにより、個別に有効/無効にできます。

表 30 および表 31 に、チャンネルごとの DMA リクエストを示します。

表 30. チャンネルごとの DMA リクエストの概要
(STM32F03x、STM32F04x、および STM32F030x8STM32F05x デバイスの場合)

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5
ADC	ADC ⁽¹⁾	ADC ⁽²⁾	-	-	-
SPI	-	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX
USART	-	USART1_TX ⁽¹⁾	USART1_RX ⁽¹⁾	USART1_TX ⁽²⁾ USART2_TX	USART1_RX ⁽²⁾ USART2_RX
I2C	-	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX
TIM1	-	TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 TIM1_UP
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2	TIM2_CH4	TIM2_CH1
TIM3	-	TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 TIM3_TRIG	-
TIM6/DAC	-	-	TIM6_UP DAC_Channel1	-	-
TIM15	-	-	-	-	TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM
TIM16	-	-	TIM16_CH1 ⁽¹⁾ TIM16_UP ⁽¹⁾	TIM16_CH1 ⁽²⁾ TIM16_UP ⁽²⁾	-
TIM17	TIM17_CH1 ⁽¹⁾ TIM17_UP ⁽¹⁾	TIM17_CH1 ⁽²⁾ TIM17_UP ⁽²⁾	-	-	-

- DMA リクエストは、SYSCFG_CFGR1 レジスタの対応する再配置ビットがクリアされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 10.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG_CFGR1\) \(173 ページ\)](#) を参照してください。
- DMA リクエストは、SYSCFG_CFGR1 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 10.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG_CFGR1\) \(173 ページ\)](#) を参照してください。

表 31. STM32F07x デバイスでのチャンネルごとの DMA リクエストの概要

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5	チャンネル 6	チャンネル 7
ADC	ADC ⁽¹⁾	ADC ⁽²⁾	予約済み	予約済み	予約済み	予約済み	予約済み
SPI	予約済み	SPI1_RX	SPI1_TX	SPI2_RX ⁽¹⁾	SPI2_TX ⁽¹⁾	SPI2_RX ⁽²⁾	SPI2_TX ⁽²⁾
USART	予約済み	USART1_TX ⁽¹⁾ USART3_TX ⁽²⁾	USART1_RX ⁽¹⁾ USART3_RX ⁽²⁾	USART1_TX ⁽²⁾ USART2_TX ⁽¹⁾	USART1_RX ⁽²⁾ USART2_RX ⁽¹⁾	USART2_RX ⁽²⁾ USART3_RX ⁽¹⁾ USART4_RX	USART2_TX ⁽²⁾ USART3_TX ⁽¹⁾ USART4_TX
I2C	予約済み	I2C1_TX ⁽¹⁾	I2C1_RX ⁽¹⁾	I2C2_TX	I2C2_RX	I2C1_TX ⁽²⁾	I2C1_RX ⁽²⁾

表 31. STM32F07x デバイスでのチャンネルごとの DMA リクエストの概要 (続き)

ペリフェラル	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5	チャンネル 6	チャンネル 7
TIM1	予約済み	TIM1_CH1 ⁽¹⁾	TIM1_CH2 ⁽¹⁾	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_CH3 ⁽¹⁾ TIM1_UP	TIM1_CH1 ⁽²⁾ TIM1_CH2 ⁽²⁾ TIM1_CH3 ⁽²⁾	予約済み
TIM2	TIM2_CH3	TIM2_UP	TIM2_CH2 ⁽¹⁾	TIM2_CH4 ⁽¹⁾	TIM2_CH1	予約済み	TIM2_CH2 ⁽²⁾ TIM2_CH4 ⁽²⁾
TIM3	予約済み	TIM3_CH3	TIM3_CH4 TIM3_UP	TIM3_CH1 ⁽¹⁾ TIM3_TRIG ⁽¹⁾	予約済み	TIM3_CH1 ⁽²⁾ TIM3_TRIG ⁽²⁾	予約済み
TIM6/DAC	予約済み	予約済み	TIM6_UP DAC_Channel1	予約済み	予約済み	予約済み	予約済み
TIM7/DAC	予約済み	予約済み	予約済み	TIM7_UP DAC_Channel2	予約済み	予約済み	予約済み
TIM15	予約済み	予約済み	予約済み	予約済み	TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM	予約済み	予約済み
TIM16	予約済み	予約済み	TIM16_CH1 ⁽¹⁾ TIM16_UP ⁽¹⁾	TIM16_CH1 ⁽²⁾ TIM16_UP ⁽²⁾	予約済み	TIM16_CH1 ⁽³⁾ TIM16_UP ⁽³⁾	予約済み
TIM17	TIM17_CH1 ⁽¹⁾ TIM17_UP ⁽¹⁾	TIM17_CH1 ⁽²⁾ TIM17_UP ⁽²⁾	予約済み	予約済み	予約済み	予約済み	TIM17_CH1 ⁽³⁾ TIM17_UP ⁽³⁾

1. DMA リクエストは、SYSCFG_CFGR1 レジスタの対応する再配置ビットがクリアされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 10.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG_CFGR1\) \(173 ページ\)](#) を参照してください。
2. DMA リクエストは、SYSCFG_CFGR1 レジスタの対応する再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 10.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG_CFGR1\) \(173 ページ\)](#) を参照してください。
3. DMA リクエストは、SYSCFG_CFGR1 レジスタの追加の RMP2 再配置ビットがセットされた場合にのみ、この DMA チャンネルに配置されます。詳細については、[セクション 10.1.1 : SYSCFG 設定レジスタ 1 \(SYSCFG_CFGR1\) \(173 ページ\)](#) を参照してください。

STM32F09x デバイスの DMA1/DMA2 コントローラ

この章は、STM32F09x デバイスでのみ有効です。

STM32F09x には、DMA1 および DMA2 という 2 つの独立した DMA コントローラが内蔵されています。

ペリフェラル (TIMx、ADC、DAC、SPI、I2C、および USARTx) からのハードウェアリクエストは、DMAx チャンネル選択レジスタを通じて DMAx チャンネル (DMA1 1 から 7、および DMA2 1 から 5) に配置されます。1 つのチャンネルでは、一度に 1 つのリクエストのみ有効にしてください。[図 23 : STM32F09x デバイスの DMAx リクエストルーティングアーキテクチャ](#)を参照してください。

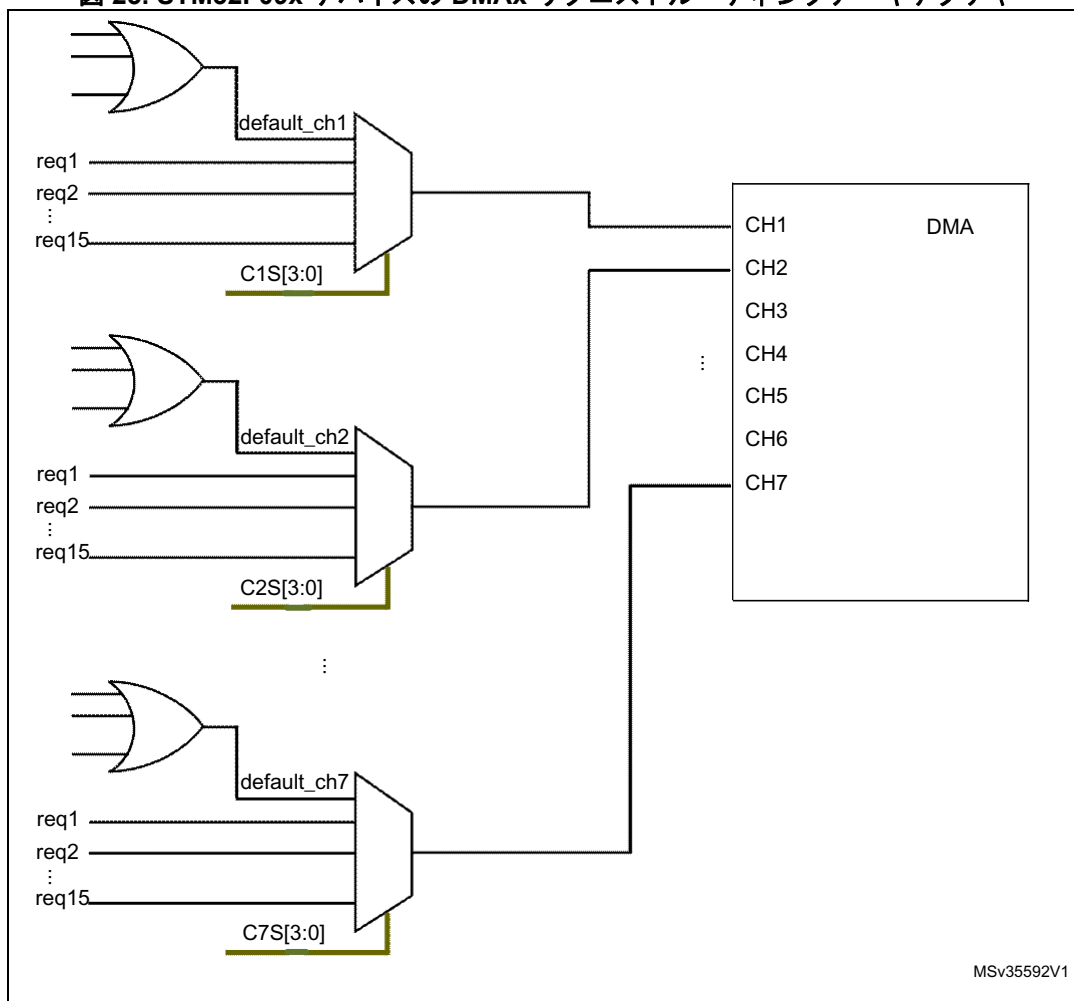
ペリフェラル DMA リクエストは、対応するペリフェラルのレジスタの DMA 制御ビットをプログラムすることにより、個別に有効/無効にできます。

デフォルトの配置位置 0 は、他の STM32F0xx 製品で使用する DMA 配置との互換性を確保します。ペリフェラル (TIMx、ADC、DAC、SPI、I2C、および USARTx) からのハードウェアリクエストは、DMA に入る前に単純に論理和が取られます。これは、1 つのチャンネルでは、一度に 1 つのリクエストのみ有効にする必要があることを意味しています。

オルタネート配置位置 1 から 15 により、DMA チャンネルでハードウェアリクエストをより柔軟に配置できるようになります。一部のペリフェラルにオルタネート配置位置が使用されると、競合を避けるため、同じリクエストはデフォルトの配置位置から削除されます。

[表 32](#) および [表 33](#) に、チャンネルとオルタネート位置ごとの DMA リクエストを示します。

図 23. STM32F09x デバイスの DMAx リクエストルーティングアーキテクチャ



1. チャンネル 6 および 7 は DMA2 では使用できません。
2. いずれかの DMA リクエストを位置 1 から 15 で選択すると、位置 0 のデフォルト位置から消えます。

表 32. STM32F09x デバイスでのチャンネルごとの DMA1 リクエストの概要

CxS [3:0]	チャンネル 1	チャンネル 2	チャンネル 3	チャンネル 4	チャンネル 5	チャンネル 6	チャンネル 7
0000	TIM2_CH3	TIM2_UP TIM3_CH3	TIM3_CH4 TIM3_UP	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	-	-
	-				TIM2_CH1	-	-
	-	-	-		TIM15_CH1 TIM15_UP TIM15_TRIG TIM15_COM	-	-
	ADC	-	TIM6_UP DAC_ Channel1	TIM7_UP DAC_ Channel2	-	-	-
	-	USART1_TX	USART1_RX	USART2_TX	USART2_RX	USART3_RX	USART3_TX
	-	-	-	-	-	USART4_RX	USART4_TX
	-	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX	-	-
	-	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX	-	-
	-	TIM1_CH1	TIM1_CH2	-	TIM1_CH3	-	-
	-	-	TIM2_CH2	TIM2_CH4	-	-	-
	TIM17_CH1 TIM17_UP	-	TIM16_CH1 TIM16_UP	TIM3_CH1 TIM3_TRIG	-	-	-
0001	ADC	ADC	TIM6_UP DAC_ Channel1	TIM7_UP DAC_ Channel2	-	-	-
0010	-	I2C1_TX	I2C1_RX	I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
0011	-	SPI1_RX	SPI1_TX	SPI2_RX	SPI2_TX	SPI2_RX	SPI2_TX
0100	-	TIM1_CH1	TIM1_CH2	-	TIM1_CH3	TIM1_CH1 TIM1_CH2 TIM1_CH3	-
0101	-	-	TIM2_CH2	TIM2_CH4	-	-	TIM2_CH2 TIM2_CH4
0110	-	-	-	TIM3_CH1 TIM3_TRIG	-	TIM3_CH1 TIM3_TRIG	-
0111	TIM17_CH1 TIM17_UP	TIM17_CH1 TIM17_UP	TIM16_CH1 TIM16_UP	TIM16_CH1 TIM16_UP	-	TIM16_CH1 TIM16_UP	TIM17_CH1 TIM17_UP
1000	USART1_RX	USART1_TX	USART1_RX	USART1_TX	USART1_RX	USART1_RX	USART1_TX
1001	USART2_RX	USART2_TX	USART2_RX	USART2_TX	USART2_RX	USART2_RX	USART2_TX
1010	USART3_RX	USART3_TX	USART3_RX	USART3_TX	USART3_RX	USART3_RX	USART3_TX
1011	USART4_RX	USART4_TX	USART4_RX	USART4_TX	USART4_RX	USART4_RX	USART4_TX
1100	USART5_RX	USART5_TX	USART5_RX	USART5_TX	USART5_RX	USART5_RX	USART5_TX
1101	USART6_RX	USART6_TX	USART6_RX	USART6_TX	USART6_RX	USART6_RX	USART6_TX
1110	USART7_RX	USART7_TX	USART7_RX	USART7_TX	USART7_RX	USART7_RX	USART7_TX
1111	USART8_RX	USART8_TX	USART8_RX	USART8_TX	USART8_RX	USART8_RX	USART8_TX

表 33. STM32F09x デバイスでのチャネルごとの DMA2 リクエストの概要

CxS[3:0]	チャネル 1	チャネル 2	チャネル 3	チャネル 4	チャネル 5
0000	なし				
0001	-	-	TIM6_UP DAC_Channel1	TIM7_UP DAC_Channel2	ADC
0010	I2C2_TX	I2C2_RX	-	-	-
0011	-	-	SPI1_RX	SPI1_TX	-
0100	-	-	-	-	-
0101	-	-	-	-	-
0110	-	-	-	-	-
0111	-	-	-	-	-
1000	USART1_TX	USART1_RX	USART1_RX	USART1_TX	USART1_TX
1001	USART2_TX	USART2_RX	USART2_RX	USART2_TX	USART2_TX
1010	USART3_TX	USART3_RX	USART3_RX	USART3_TX	USART3_TX
1011	USART4_TX	USART4_RX	USART4_RX	USART4_TX	USART4_TX
1100	USART5_TX	USART5_RX	USART5_RX	USART5_TX	USART5_TX
1101	USART6_TX	USART6_RX	USART6_RX	USART6_TX	USART6_TX
1110	USART7_TX	USART7_RX	USART7_RX	USART7_TX	USART7_TX
1111	USART8_TX	USART8_RX	USART8_RX	USART8_TX	USART8_TX

11.4 DMA レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、バイト (8 ビット)、ハーフワード (16 ビット)、またはワード (32 ビット) 単位でアクセスできます。

11.4.1 DMA 割り込みステータスレジスタ (DMA_ISR および DMA2_ISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27、23、19、**TEIFx** : チャンネル x 転送エラーフラグ (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

15、11、7、3 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャンネル x に転送エラーは発生していません。

1 : チャンネル x に転送エラー (TE) が発生しました。

ビット 26、22、18、**HTIFx** : チャンネル x 1/2 転送フラグ (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

14、10、6、2 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャンネル x に 1/2 転送 (HT) イベントは発生していません。

1 : チャンネル x に 1/2 転送 (HT) イベントが発生しました。

ビット 25、21、17、**TCIFx** : チャンネル x 転送完了フラグ (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

13、9、5、1 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャンネル x の転送完了 (TC) イベントは発生していません。

1 : チャンネル x に転送完了 (TC) イベントが発生しました。

ビット 24、20、16、**GIFx** : チャンネル x グローバル割り込みフラグ (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

12、8、4、0 このビットは、ハードウェアによってセットされます。ソフトウェアで DMA_IFCR レジスタの対応するビットに 1 を書き込むことによってクリアされます。

0 : チャンネル x に TE、HT、または TC イベントは発生していません。

1 : チャンネル x に TE、HT、または TC イベントが発生しました。

11.4.2 DMA 割り込みフラグクリアレジスタ (DMA_IFCR および DMA2_IFCR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
				w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27、23、19、**CTEIFx** : チャネル x 転送エラークリア (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

15、11、7、3 このビットは、ソフトウェアによってセット／クリアされます。

0 : 影響なし。

1 : DMA_ISR レジスタの対応する TEIF フラグをクリアします。

ビット 26、22、18、**CHTIFx** : チャネル x 1/2 転送クリア (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

14、10、6、2 このビットは、ソフトウェアによってセット／クリアされます。

0 : 影響なし。

1 : DMA_ISR レジスタの対応する HTIF フラグをクリアします。

ビット 25、21、17、**CTCIFx** : チャネル x 転送完了クリア (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

13、9、5、1 このビットは、ソフトウェアによってセット／クリアされます。

0 : 影響なし。

1 : DMA_ISR レジスタの対応する TCIF フラグをクリアします。

ビット 24、20、16、**CGIFx** : チャネル x グローバル割り込みクリア (DMA の場合 x = 1 ~ 7、DMA2 の場合 x = 1 ~ 5)

12、8、4、0 このビットは、ソフトウェアによってセット／クリアされます。

0 : 影響なし。

1 : DMA_ISR レジスタの GIF、TEIF、HTIF、および TCIF フラグをクリアします。

11.4.3 DMA チャネル x 設定レジスタ (DMA_CCRx および DMA2_CCRx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。 ここで x = チャネル数)

アドレスオフセット : 0x08 + 0d20 x (チャネル番号 - 1)

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **MEM2MEM** : メモリ間モード

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : メモリ間モードは無効です。
- 1 : メモリ間モードは有効です。

ビット 13:12 **PL[1:0]** : チャネル優先順位レベル

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 低
- 01 : 中
- 10 : 高
- 11 : 最優先

ビット 11:10 **MSIZE[1:0]** : メモリサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 8 ビット
- 01 : 16 ビット
- 10 : 32 ビット
- 11 : 予約済み

ビット 9:8 **PSIZE[1:0]** : ペリフェラルサイズ

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 8 ビット
- 01 : 16 ビット
- 10 : 32 ビット
- 11 : 予約済み

ビット 7 **MINC** : メモリインクリメントモード

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : メモリインクリメントモードは無効です。
- 1 : メモリインクリメントモードは有効です。

ビット 6 **PINC** : ペリフェラルインクリメントモード

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : ペリフェラルインクリメントモードは無効です。
- 1 : ペリフェラルインクリメントモードは有効です。

ビット 5 CIRC : サーキュラモード

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : サーキュラモードは無効です。
- 1 : サーキュラモードは有効です。

ビット 4 DIR : データ転送方向

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : ペリフェラルから読み出します。
- 1 : メモリから読み出します。

ビット 3 TEIE : 転送エラー割り込み有効化

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : TE 割り込みは無効です。
- 1 : TE 割り込みは有効です。

ビット 2 HTIE : 1/2 転送割り込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : HT 割り込みは無効です。
- 1 : HT 割り込みは有効です。

ビット 1 TCIE : 転送完了割り込み有効化

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : TC 割り込みは無効です。
- 1 : TC 割り込みは有効です。

ビット 0 EN : チャネルイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

- 0 : チャネルは無効です。
- 1 : チャネルは有効です。

11.4.4 DMA チャンネル x データ数レジスタ (DMA_CNDTRx および DMA2_CNDTRx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。x = チャンネル番号)

アドレスオフセット : $0x0C + 0d20 \times$ (チャンネル番号 - 1)

リセット値 : $0x0000\ 0000$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 NDT[15:0] : 転送データ項目の数

転送データ数 : 0~65535 このレジスタは、チャンネルが無効な場合にのみ書き込みできます。チャンネルが有効になると、このレジスタは読み出し専用になり、送信されるべき残りのバイトを指し示します。このレジスタは、各 DMA 転送後にデクリメントされます。

転送が完了すると、このレジスタは、ゼロのままにすることもできれば、チャンネルが自動再ロードモードに設定されている場合には、事前にプログラムされた値を自動的に再ロードすることもできます。

このレジスタの値が 0 の場合、チャンネルが有効であるか否かにかかわらず、トランザクションは処理できません。

11.4.5 DMA チャンネル x ペリフェラルアドレスレジスタ (DMA_CPARx および DMA2_CPARx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。x = チャンネル番号)

アドレスオフセット : $0x10 + 0d20 \times$ (チャンネル番号 - 1)

リセット値 : $0x0000\ 0000$

チャンネルが有効なときには、このレジスタへは書き込みを *行わない* てください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 PA[31:0] : ペリフェラルアドレス

データの読み出し／書き込みが行われるペリフェラルデータレジスタのベースアドレス。

PSIZE が 01 (16 ビット) のとき、PA[0] ビットは無視されます。アクセスは、自動的にハーフワードアドレスに整列されます。

PSIZE が 10 (32 ビット) のとき、PA[1:0] は無視されます。アクセスは、自動的にワードアドレスに整列されます。

11.4.6 DMA チャンネル x メモリアドレスレジスタ (DMA_CMARx および DMA2_CMARx) (DMA の場合 x = 1 ~ 7、および DMA2 の場合 x = 1 ~ 5。
x = チャンネル番号)

アドレスオフセット : 0x14 + 0d20 x (チャンネル番号 - 1)

リセット値 : 0x0000 0000

チャンネルが有効なときには、このレジスタへは書き込みを *行わない* てください。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA [31:16]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA [15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:0 **MA[31:0]** : メモリアドレス

データの読み出し／書き込みが行われるメモリ領域のベースアドレス。

MSIZE が 01 (16 ビット) のとき、MA[0] ビットは無視されます。アクセスは、自動的にハーフワードアドレスに整列されます。

MSIZE が 10 (32 ビット) のとき、MA[1:0] は無視されます。アクセスは、自動的にワードアドレスに整列されます。

11.4.7 DMA チャンネル選択レジスタ (DMA_CSELR および DMA2_CSELR)

このレジスタは、STM32F09x デバイスにのみ存在します。

アドレスオフセット : 0xA8

リセット値 : 0x0000 0000

このレジスタは、DMA チャンネルの再配置を管理するために使用されます (図 23を参照)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	C7S [3:0]				C6S [3:0]				C5S [3:0]			
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
C4S [3:0]				C3S [3:0]				C2S [3:0]				C1S [3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 **C7S[3:0]** : DMA チャンネル 7 選択
チャンネル 7 の DMA リクエストマッピング ⁽¹⁾。DMA2 では使用できません。

ビット 23:20 **C6S[3:0]** : DMA チャンネル 6 選択
チャンネル 6 の DMA リクエストマッピング ⁽¹⁾。DMA2 では使用できません。

ビット 19:16 **C5S[3:0]** : DMA チャンネル 5 選択
チャンネル 5 の DMA リクエストマッピング ⁽¹⁾。

ビット 15:12 **C4S[3:0]** : DMA チャンネル 4 選択
チャンネル 4 の DMA リクエストマッピング ⁽¹⁾。

ビット 11:8 **C3S[3:0]** : DMA チャンネル 3 選択
チャンネル 3 の DMA リクエストマッピング ⁽¹⁾。

ビット 7:4 **C2S[3:0]** : DMA チャンネル 2 選択
チャンネル 2 の DMA リクエストマッピング ⁽¹⁾。

ビット 3:0 **C1S[3:0]** : DMA チャンネル 1 選択
チャンネル 1 の DMA リクエストマッピング ⁽¹⁾。

- 具体的な DMA リクエストマッピングについては、表 32 : STM32F09x デバイスでのチャンネルごとの DMA1 リクエストの概要および表 33 : STM32F09x デバイスでのチャンネルごとの DMA2 リクエストの概要を参照してください。

次の表に、DMA レジスタマップとリセット値を示します。

[illegible]

表 34. DMA レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x58	DMA_CCR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]			MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x5C	DMA_CNDTR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		NDT[15:0]															
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x60	DMA_CPAR5	PA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x64	DMA_CMAR5	MA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表 35. DMA レジスタマップとリセット値
(レジスタは STM32F07x および STM32F09x デバイスでのみ使用可能)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x6C	DMA_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC		PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x70	DMA_CNDTR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]																
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	DMA_CPAR6	PA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x78	DMA_CMAR6	MA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	予約済み																																	
0x80	DMA_CCR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC		PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x84	DMA_CNDTR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]																
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	DMA_CPAR7	PA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	DMA_CMAR7	MA[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90	予約済み																																	

表 36. DMA レジスタマップとリセット値
(レジスタは STM32F09x デバイスでのみ使用可能)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0A8	DMA_CSELR	Res	Res	Res	Res	C7S[3:0]				C6S[3:0]				C5S[3:0]				C4S[3:0]				C3S[3:0]				C2S[3:0]				C1S[3:0]			
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

12 割り込みとイベント

12.1 ネスト化されたベクタ割り込みコントローラ (NVIC)

12.1.1 NVIC の主な機能

- 32 マスク可能な割り込みチャネル (Cortex[®]-M0 の 16 本の割り込みラインは数に含まれていない)
- 4 のプログラム可能な優先レベル (2 ビットの割り込み優先順位を使用)
- 遅延時間の少ない例外および割り込み処理
- 電源管理制御
- システム制御レジスタの実装

NVIC とプロセッサコアのインタフェースは密に結合され、割り込み処理の遅延時間を小さくし、遅れて到着した割り込みを効率的に処理できます。

コア例外を含むすべての割り込みは、NVIC によって管理されます。例外と NVIC プログラミングの詳細については、プログラミングマニュアル PM0215 を参照してください。

コード例については、「付録」セクション [A.6.1 : NVIC の初期化の例](#) を参照してください。

12.1.2 SysTick 較正值レジスタ

SysTick 較正值は 6000 にセットされますので、SysTick クロックを 6 MHz (最大 $f_{HCLK}/8$) に設定した状態での基準タイムベースは 1 ms になります。

12.1.3 割り込みベクタと例外ベクタ

[表 37](#) は、STM32F0xx デバイスのベクタテーブルです。特定のデバイスの使用可能なペリフェラルについて考慮してください。

表 37. ベクタテーブル

位置	優先順位	優先種別	項目 (略称)	説明	アドレス
-	-	-	-	予約済み	0x0000 0000
-	-3	固定	リセット	リセット	0x0000 0004
-	-2	固定	NMI	ノンマスカブル割り込み。RCC クロックセキュリティシステム (CSS) は NMI ベクタにリンクされます。	0x0000 0008
-	-1	固定	HardFault	あらゆる種類の異常	0x0000 000C
-	3	設定可能	SVCall	SWI 命令によるシステムサービスコール	0x0000 002C
-	5	設定可能	PendSV	ペンディング可能なシステムサービスリクエスト	0x0000 0038
-	6	設定可能	SysTick	システムティックタイマ	0x0000 003C
0	7	設定可能	WWDG	ウィンドウ型ウォッチドッグ割り込み	0x0000 0040
1	8	設定可能	PVD_VDDIO2	PVD および VDDIO2 電源のコンパレータ割り込み (EXTI ライン 16 および 31 との組み合わせ)	0x0000 0044
2	9	設定可能	RTC	RTC 割り込み (EXTI ライン 17、19、および 20 の組み合わせ)	0x0000 0048
3	10	設定可能	FLASH	フラッシュグローバル割り込み	0x0000 004C



表 37. ベクタテーブル（続き）

位置	優先順位	優先種別	項目（略称）	説明	アドレス
4	11	設定可能	RCC_CRS	RCC および CRS グローバル割り込み	0x0000 0050
5	12	設定可能	EXTI0_1	EXTI ライン [1:0] 割り込み	0x0000 0054
6	13	設定可能	EXTI2_3	EXTI ライン [3:2] 割り込み	0x0000 0058
7	14	設定可能	EXTI4_15	EXTI ライン [15:4] 割り込み	0x0000 005C
8	15	設定可能	TSC	タッチセンシング割り込み	0x0000 0060
9	16	設定可能	DMA_CH1	DMA チャンネル 1 割り込み	0x0000 0064
10	17	設定可能	DMA_CH2_3 DMA2_CH1_2	DMA チャンネル 2 および 3 割り込み DMA2 チャンネル 1 および 2 割り込み	0x0000 0068
11	18	設定可能	DMA_CH4_5_6_7 DMA2_CH3_4_5	DMA チャンネル 4、5、6、および 7 割り込み DMA2 チャンネル 3、4、および 5 割り込み	0x0000 006C
12	19	設定可能	ADC_COMP	ADC および COMP 割り込み (ADC 割り込みと EXTI ライン 21 および 22 の組み合わせ)	0x0000 0070
13	20	設定可能	TIM1_BRK_UP_ TRG_COM	TIM1 ブレーク、更新、トリガ、およびコミュレーション割り込み	0x0000 0074
14	21	設定可能	TIM1_CC	TIM1 キャプチャ / 比較割り込み	0x0000 0078
15	22	設定可能	TIM2	TIM2 グローバル割り込み	0x0000 007C
16	23	設定可能	TIM3	TIM3 グローバル割り込み	0x0000 0080
17	24	設定可能	TIM6_DAC_	TIM6 グローバル割り込みおよび DAC アンダーラン割り込み	0x0000 0084
18	25	設定可能	TIM7	TIM7 グローバル割り込み	0x0000 0088
19	26	設定可能	TIM14	TIM14 グローバル割り込み	0x0000 008C
20	27	設定可能	TIM15	TIM15 グローバル割り込み	0x0000 0090
21	28	設定可能	TIM16	TIM16 グローバル割り込み	0x0000 0094
22	29	設定可能	TIM17	TIM17 グローバル割り込み	0x0000 0098
23	30	設定可能	I2C1	I ² C1 グローバル割り込み (EXTI ライン 23 の組み合わせ)	0x0000 009C
24	31	設定可能	I2C2	I ² C2 グローバル割り込み	0x0000 00A0
25	32	設定可能	SPI1	SPI1 グローバル割り込み	0x0000 00A4
26	33	設定可能	SPI2	SPI2 グローバル割り込み	0x0000 00A8
27	34	設定可能	USART1	USART1 グローバル割り込み (EXTI ライン 25 の組み合わせ)	0x0000 00AC
28	35	設定可能	USART2	USART2 グローバル割り込み (EXTI ライン 26 の組み合わせ)	0x0000 00B0
29	36	設定可能	USART3_4_5_6_7_8	USART3、USART4、USART5、USART6、USART7、USART8 グローバル割り込み (EXTI ライン 28 の組み合わせ)	0x0000 00B4
30	37	設定可能	CEC_CAN	CEC および CAN グローバル割り込み (EXTI ライン 27 の組み合わせ)	0x0000 00B8
31	38	設定可能	USB	USB グローバル割り込み (EXTI ライン 18 の組み合わせ)	0x0000 00BC

12.2 拡張割り込み／イベントコントローラ (EXTI)

拡張割り込み／イベントコントローラ (EXTI) は、外部および内部で非同期的に発生するイベント／割り込みを管理し、CPU／割り込みコントローラへのイベントリクエスト、およびパワーマネージャへのウェイクアップリクエストを生成します。

EXTI を使用することで、外部／内部イベントラインを最大 32 本（23 本の外部イベントラインおよび 9 本の内部イベントライン）まで管理することが可能です。

各外部割り込みラインのアクティブエッジは個別に選択可能です。一方で、内部割り込みのアクティブエッジは常に立ち上がりエッジです。割り込みは保留のまま残すことができます。外部割り込みの場合、ステータスレジスタはインスタンス化され、割り込みのソースを示します。イベントは常に単純なパルスで、コアウェイクアップのトリガに使用されます（例：Cortex-M0 RXEV ピン）。内部割り込みの場合、保留ステータスは IP を生成することで保証され、特別なフラグは必要ありません。各入力ラインは、割り込みまたはイベント生成時に個別にマスクすることができます。また、内部ラインは STOP モードでのみサンプリングされます。このコントローラは、専用レジスタに書き込むことで、対応するハードウェアイベントラインによって多重化されているソフトウェアによる（唯一の）外部イベントのエミュレーションを可能にします。

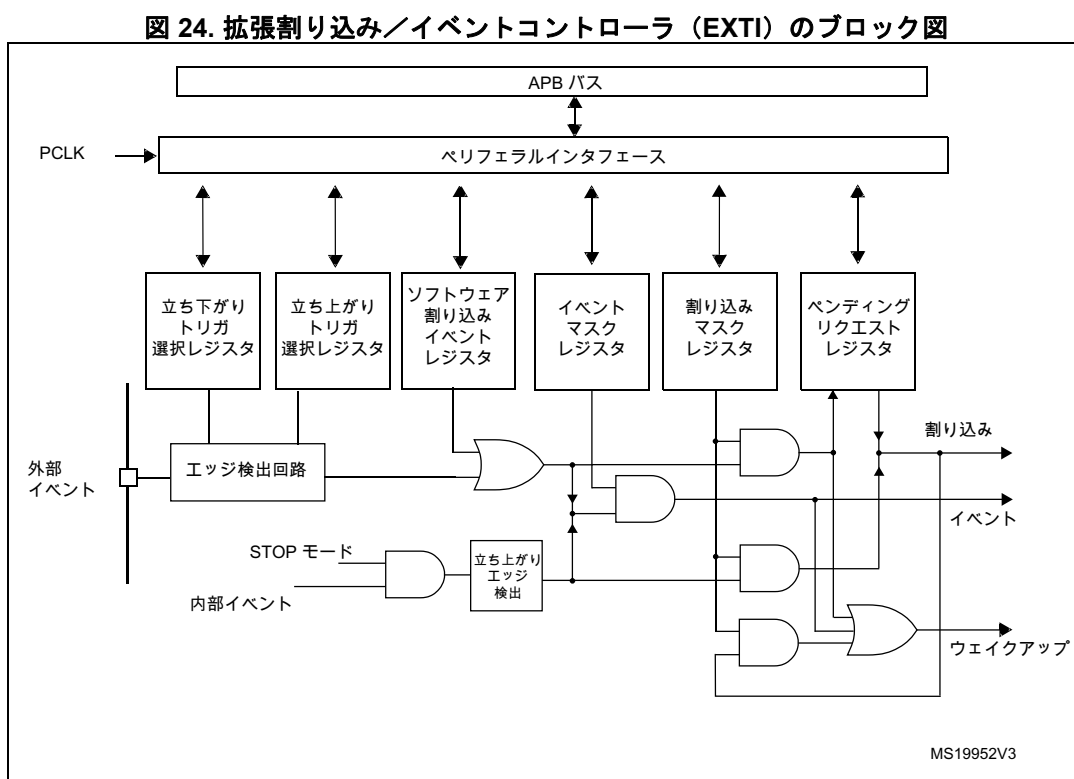
12.2.1 主な特長

EXTI の主な機能は以下のとおりです。

- 最大 32 のイベント／割り込みリクエストの生成をサポート
- 各イベント／割り込みラインの個別マスク
- システムが STOP モードでない場合に、内部ラインを自動的に無効化
- 外部イベント／割り込みラインの個別トリガ
- 外部割り込みライン専用のステータスビット
- すべての外部イベントリクエストのエミュレーション

12.2.2 ブロック図

図 24 に拡張割り込み／イベントブロック図を示します。



12.2.3 イベント管理

STM32F0xx は、コアをウェイクアップするために、外部イベントや内部イベントを処理することができます (WFE)。ウェイクアップイベントは、次のいずれかによって生成できます。

- ペリフェラル制御レジスタで割り込みを有効にし、NVIC では有効にせず、Cortex-M0 システム制御レジスタの SEVONPEND ビットを有効にします。MCU が WFE からリスタートするときには、EXTI ペリフェラル割り込みペンディングビットと (NVIC 割り込みクリアペンディングレジスタの) ペリフェラル NVIC IRQ チャンネルペンディングビットをクリアする必要があります。
- あるいは、外部または内部 EXTI ラインをイベントモードに設定します。CPU が WFE からリスタートするときには、イベントラインに対応するペンディングビットはセットされていないので、ペリフェラル割り込みペンディングビットや NVIC IRQ チャンネルペンディングビットをクリアする必要はありません。

12.2.4 機能詳細

外部割り込みラインで割り込みを生成するには、割り込みラインを設定し、有効にする必要があります。それには、2つのトリガレジスタに希望するエッジ検出をプログラムし、割り込みマスクレジスタの対応するビットに"1"を書き込むことによって割り込みリクエストを有効にします。選択されたエッジが外部割り込みラインで発生すると、割り込みリクエストが生成されます。割り込みラインに対応するペンディングビットもセットされます。このリクエストは、ペンディングレジスタに"1"を書き込むことによってリセットされます。

内部割り込みラインの場合、立ち上がりエッジが常にアクティブエッジです。割り込みは割り込みマスクレジスタのデフォルト値によって有効化され、ペンディングレジスタには対応するペンディングビットはセットされていません。

イベントを生成するには、イベントラインを設定し、有効にする必要があります。それには、2つのトリガレジスタに希望するエッジ検出をプログラムし、イベントマスクレジスタの対応するビットに"1"を書き込むことによってイベントリクエストを有効にします。選択されたエッジがイベントラインで発生すると、イベントパルスが生成されます。イベントラインに対応するペンディングビットはセットされません。

外部ラインの場合、割り込み／イベントリクエストは、ソフトウェア割り込み／イベントレジスタに"1"を書き込むことによって、ソフトウェアによって生成することもできます。

注： *内部ラインに関連する割り込みやイベントは、システムがSTOPモードのときのみトリガできます。システムがまだ動作している場合は、割り込み／イベントは生成されません。*

コード例については、「付録」セクション [A.6.2：外部割り込み選択のコード例](#) を参照してください。

ハードウェア割り込みの選択

ラインを割り込みソースとして設定するには、次の手順を使用します。

- EXTI_IMR レジスタの対応するマスクビットを設定します。
- 割り込みラインのトリガ選択ビットを設定します (EXTI_RTISR および EXTI_FTISR)。
- いずれかの EXTI ラインからの割り込みを正しく処理できるように、EXTI にマップされた NVIC IRQ チャネルを制御するイネーブルビットとマスクビットを構成します。

ハードウェアイベントの選択

ラインをイベントソースとして設定するには、次の手順を使用します。

- EXTI_EMR レジスタの対応するマスクビットを設定します。
- イベントラインのトリガ選択ビットを設定します (EXTI_RTISR および EXTI_FTISR)。

ソフトウェア割り込み／イベントの選択

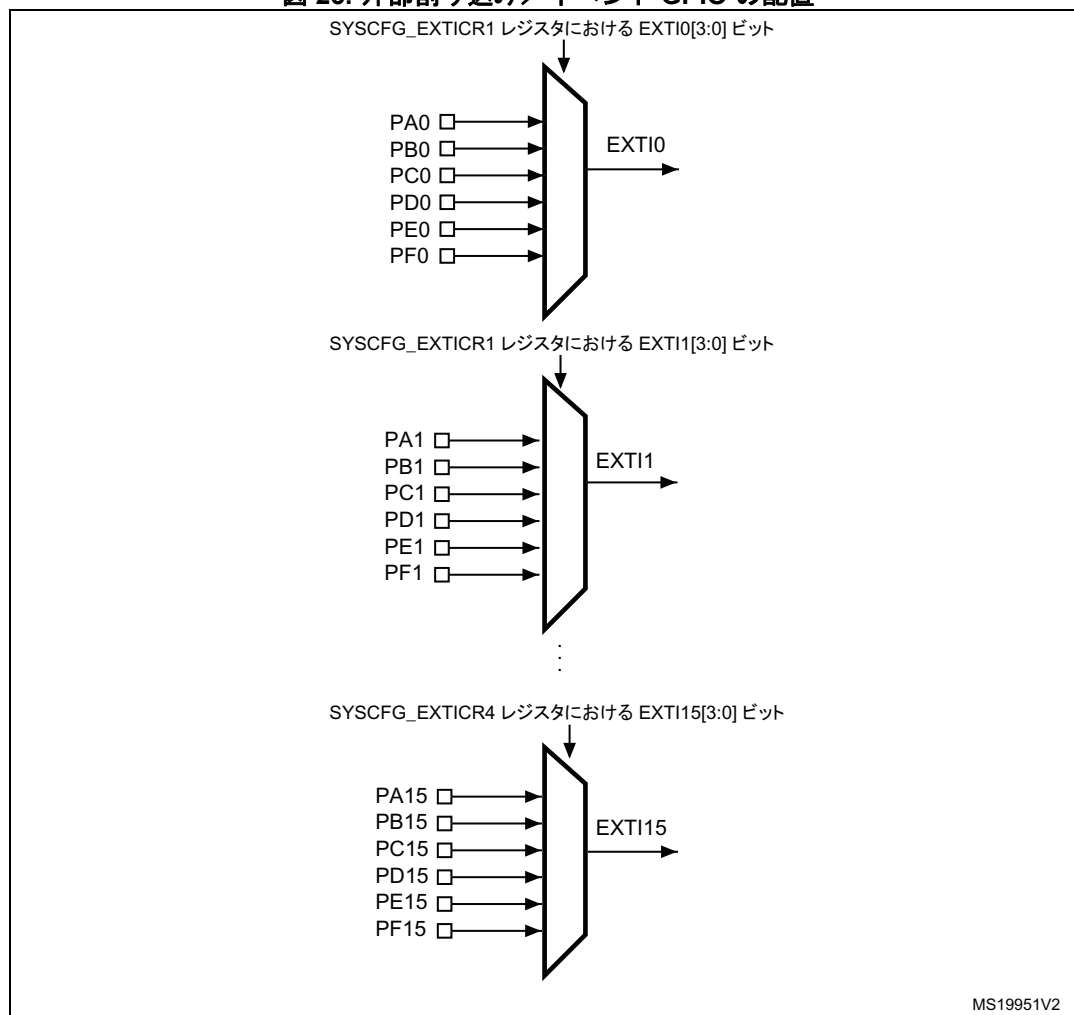
外部ラインのいずれかを、ソフトウェア割り込み／イベントラインとして設定できます。ソフトウェア割り込みの生成には、次の手順を使用します。

- 対応するマスクビット (EXTI_IMR、EXTI_EMR) を設定します。
- ソフトウェア割り込みレジスタの必要なビットをセットします (EXTI_SWIER)。

12.2.5 外部および内部の割り込み／イベントラインの配置

GPIO は、次のように、16 本の外部割り込み／イベントラインに接続されます。

図 25. 外部割り込み／イベント GPIO の配置



残りのラインは、以下のように接続されます。

- EXTI ライン 16 は PVD 出力に接続されます。
- EXTI ライン 17 は RTC アラームイベントに接続されます。
- EXTI ライン 18 は内部 USB ウェイクアップイベントに接続されます。
- EXTI ライン 19 は RTC のタンパイイベントおよびタイムスタンプイベントに接続されます。
- EXTI ライン 20 は RTC ウェイクアップイベント（STM32F07x および STM32F09x デバイスでのみ使用可能）に接続されます。
- EXTI ライン 21 はコンパレータ 1 出力に接続されます。
- EXTI ライン 22 はコンパレータ 2 出力に接続されます。
- EXTI ライン 23 は内部 I2C1 ウェイクアップイベントに接続されます。
- EXTI ライン 24 は予約済みです（内部的にローに保たれます）。
- EXTI ライン 25 は内部 USART1 ウェイクアップイベントに接続されます。

- EXTI ライン 26 は内部 USART2 ウェイクアップイベント（STM32F07x および STM32F09x デバイスでのみ使用可能）に接続されます。
- EXTI ライン 27 は内部 CEC ウェイクアップイベントに接続されます。
- EXTI ライン 28 は内部 USART3 ウェイクアップイベント（STM32F09x デバイスでのみ使用可能）に接続されます。
- EXTI ライン 29 は予約済みです（内部的にローに保たれます）。
- EXTI ライン 30 は予約済みです（内部的にローに保たれます）。
- EXTI ライン 31 は V_{DDIO2} 電源のコンパレータ出力（STM32F04x、STM32F07x、および STM32F09x デバイスでのみ使用可能）に接続されます。

注： 予約済みまたは一部のデバイスで使用されていない EXTI ラインは内部とみなされます。

12.3 EXTI レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード（32 ビット）単位でアクセスする必要があります。

12.3.1 割り込みマスクレジスタ（EXTI_IMR）

アドレスオフセット：0x00

リセット値： 0x0FF4 0000（STM32F03x デバイス）
0x7FF4 0000（STM32F04x デバイス）
0x0F94 0000（STM32F05x デバイス）
0x7F84 0000（STM32F07x および STM32F09x デバイス）

注： デフォルトで割り込みを有効にするには、内部ラインのリセット値は「1」にセットされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR31	MR30	MR29	MR28	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **MRx**：外部／内部ライン x の割り込みマスク

- 0：ライン x からの割り込みリクエストはマスクされます。
- 1：ライン x からの割り込みリクエストはマスクされません。

12.3.2 イベントマスクレジスタ (EXTI_EMR)

アドレスオフセット : 0x04
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR31	MR30	MR29	MR28	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:0 **MRx** : 外部/内部ライン x のイベントマスク
0 : ライン x からのイベントリクエストはマスクされます。
1 : ライン x からのイベントリクエストはマスクされません。

12.3.3 立ち上がりトリガ選択レジスタ (EXTI_RTSTR)

アドレスオフセット : 0x08
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR22	TR21	TR20	TR19	Res.	TR17	TR16
									r/w	r/w	r/w	r/w		r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:19 **TRx** : ライン x の立ち上がりトリガイベント設定ビット (x = 22~19)
0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。
1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17:0 **TRx** : ライン x の立ち上がりトリガイベント設定ビット (x = 17~0)
0 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは無効です。
1 : 入力ラインの (イベントと割り込みについて) 立ち上がりトリガは有効です。

注 : 外部ウェイクアップラインはエッジトリガされています。これらのライン上でグリッチが生成されないようにする必要があります。EXTI_RTSTR レジスタへの書き込み中に外部割り込みラインで立ち上がりエッジが発生した場合、ペンディングビットはセットされません。

同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

12.3.4 立ち下がりトリガ選択レジスタ (EXTI_FTSR)

アドレスオフセット : 0x0C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR22	TR21	TR20	TR19	Res.	TR17	TR16
									r/w	r/w	r/w	r/w		r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:19 **TRx** : ライン x の立ち下がりトリガイイベント設定ビット (x = 22~19)
0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です
1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17:0 **TRx** : ライン x の立ち下がりトリガイイベント設定ビット (x = 17~0)
0 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは無効です
1 : 入力ラインの (イベントと割り込みについて) 立ち下がりトリガは有効です

注 : 外部ウェイクアップラインはエッジトリガされています。これらのライン上でグリッチが生成されないようにする必要があります。EXTI_FTSR レジスタへの書き込み中に外部割り込みラインで立ち下がリエッジが発生した場合、ペンディングビットはセットされません。

同じ割り込みラインに対して、立ち上がりエッジトリガと立ち下がりエッジトリガを設定することができます。この場合、両方のエッジでトリガ条件が生成されます。

12.3.5 ソフトウェア割り込みイベントレジスタ (EXTI_SWIER)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER2 2	SWIER2 1	SWIER2 0	SWIER1 9	Res.	SWIER1 7	SWIER1 6
									rw	rw	rw	rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWIER1 5	SWIER1 4	SWIER1 3	SWIER1 2	SWIER1 1	SWIER1 0	SWIER9	SWIER8	SWIER7	SWIER6	SWIER5	SWIER4	SWIER3	SWIER2	SWIER1	SWIER0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:19 **SWIERx** : ライン x のソフトウェア割り込み (x = 22~19)

割り込みが EXTI_IMR のこのライン上で有効になっている場合、このビットが“0”のときに“1”を書き込むと EXTI_PR の対応する保留中のビットがセットされるため、割り込みリクエストが生成されます。

EXTI_PR レジスタの対応するビットをクリアする (対応ビットに“1”を書き込む) と、このビットはクリアされます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17:0 **SWIERx** : ライン x のソフトウェア割り込み (x = 17~0)

割り込みが EXTI_IMR のこのライン上で有効になっている場合、このビットが“0”のときに“1”を書き込むと EXTI_PR の対応する保留中のビットがセットされるため、割り込みリクエストが生成されます。

EXTI_PR レジスタの対応するビットをクリアする (対応ビットに“1”を書き込む) と、このビットはクリアされます。

12.3.6 ペンディングレジスタ (EXTI_PR)

アドレスオフセット : 0x14
リセット値 : 定義されていません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR22	PR21	PR20	PR19	Res.	PR17	PR16
									rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:23 予約済みであり、リセット値に保持する必要があります。

ビット 22:19 **PRx** : ライン x のペンディングビット (x = 22~19)

0 : トリガリクエストは発生していません。

1 : 選択されたトリガリクエストが発生しました。

このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。このビットは、ビットに 1 を書き込むことによってクリアされます。

ビット 18 予約済みであり、リセット値に保持する必要があります。

ビット 17:0 **PRx** : ライン x のペンディングビット (x = 17~0)

0 : トリガリクエストは発生していません。

1 : 選択されたトリガリクエストが発生しました。

このビットは、選択されたエッジイベントが外部割り込みラインで発生したときにセットされます。このビットは、ビットに 1 を書き込むことによってクリアされます。

12.3.7 EXTI レジスタマップ

次の表に、EXTI レジスタマップとリセット値を示します。

表 38. 外部割り込み／イベントコントローラのレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	EXTI_IMR	MR[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_EMR	MR[31:0]																															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_RTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR23	TR22	TR21	TR20	TR19	Res.	TR[17:0]																
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	EXTI_FTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR23	TR22	TR21	TR20	TR19	Res.	TR[17:0]																
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	EXTI_SWIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER23	SWIER22	SWIER21	SWIER20	SWIER19	Res.	SWIER[17:0]																
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	EXTI_PR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR23	PR22	PR21	PR20	PR19	Res.	PR[17:0]																
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。



13 アナログデジタルコンバータ (ADC)

13.1 概要

この 12 ビット ADC は、逐次比較型アナログデジタルコンバータです。最大 19 の多重化チャネルを持ち、16 の外部および3内部ソースからの信号を測定することができます。さまざまなチャネルの A/D 変換は、シングル、連続、スキャン、または不連続モードで行うことができます。ADC の結果は、左詰めまたは右詰めで 16 ビットのデータレジスタに格納されます。

アナログウォッチドッグ機能により、入力電圧が、ユーザ定義の上限値または下限値から逸脱していないかを、アプリケーションで検出することができます。

低周波数で非常に低い消費電力を可能にするために、効率的な低電力モードが実装されています。

13.2 ADC の主な機能

- 高性能
 - 12 ビット、10 ビット、8 ビット、または 6 ビットの設定可能な分解能
 - ADC 変換時間：12 ビット分解能（1 MHz）で 1.0 μ s、10 ビット分解能で 0.93 μ s。分解能を下げることによって、より高速な変換時間を達成可能。
 - 自己較正
 - プログラム可能なサンプリング時間
 - 組み込みのデーター貫性によるデータ配置
 - DMA サポート
- 低電力
 - アプリケーションは最適な ADC 性能を維持しながら、低電力動作の PCLK 周波数を低減できます。たとえば、PCLK の周波数にかかわらず、1.0 μ s の変換時間が維持されます。
 - ウェイトモード：低周波数 PCLK のアプリケーションの ADC オーバーランを防止します。
 - オートオフモード：アクティブ変換フェーズ中以外は、ADC の電源は自動的に切れます。これは ADC の消費電力を大幅に低減します。
- アナログ入力チャンネル
 - 16 の外部アナログ入力
 - 内部温度センサ用の 1 つのチャンネル (V_{SENSE})
 - 内部基準電圧用の 1 つのチャンネル (V_{REFINT})
 - 外部 V_{BAT} 電源ピンを監視するための 1 つのチャンネル
- 変換開始は、次のように開始できます。
 - ソフトウェアによって
 - 極性が設定可能なハードウェアトリガによって (TIM1、TIM2、TIM3、および TIM15 からの内部タイマイベント)
- 変換モード
 - 単一チャンネルを変換でき、または一連のチャンネルをスキャンできます。
 - シングルモードは、選択された入力をトリガごとに 1 回変換します。
 - 連続モードは、選択された入力を連続的に変換します。
 - 不連続モード
- サンプリング終了時、変換終了時、シーケンス変換終了時、およびアナログウォッチドッグイベントまたはオーバーランイベント時に割り込みを生成します。
- アナログウォッチドッグ
- ADC 電源仕様：2.4 から 3.6 V
- ADC 入力電圧範囲： $V_{SSA} \leq V_{IN} \leq V_{DDA}$

 26 に、ADC のブロック図を示します。

13.3 ADC ピンと内部信号

表 39. ADC 内部信号

内部信号名	信号タイプ	説明
TRGx	入力	ADC 変換トリガ
V _{SENSE}	入力	内部温度センサ出力電圧
V _{REFINT}	入力	内部電圧基準出力電圧
V _{BAT/2}	入力	2 分圧された V _{BAT} ピン入力電圧

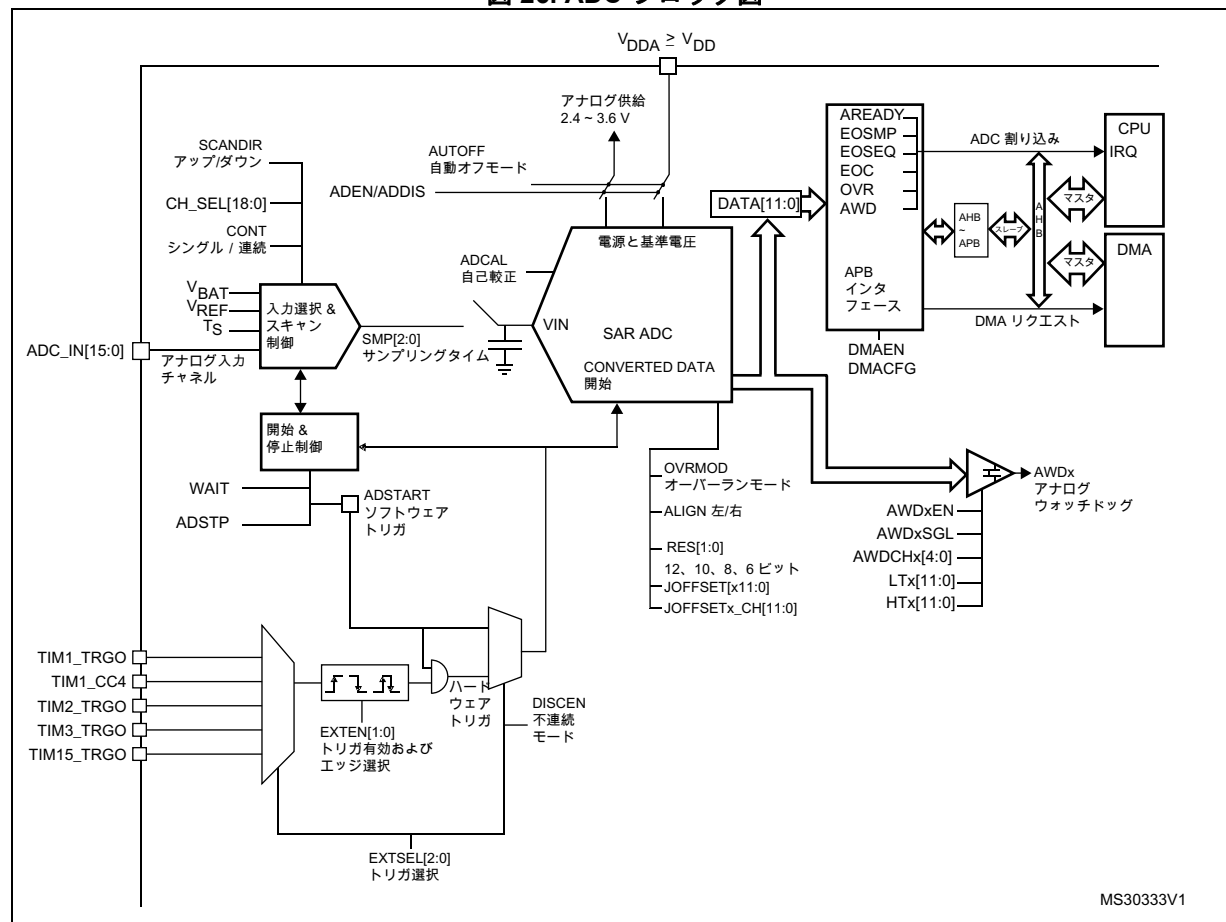
表 40. ADC ピン

名前	信号タイプ	説明
V _{DDA}	入力、アナログ電源	ADC、V _{DDA} 、V _{DD} のアナログ電源と正基準電圧
V _{SSA}	入力、アナログ供給グラウンド	アナログ電源のグラウンド。V _{SS} 電位である必要がある
ADC_IN[15:0]	アナログ入力信号	16 チャンネルのアナログ入力

13.4 ADC の機能詳細

図 26 に ADC ブロック図を、表 40 に ADC ピンの説明を示します。

図 26. ADC ブロック図



13.4.1 較正 (ADCAL)

ADC は較正機能を備えています。この手順時、ADC は、次の ADC の電源オフまで ADC に内部で適用される較正係数を計算します。アプリケーションは、較正中は ADC を使用してはならず、較正が完了するまで待つ必要があります。

較正は、A/D 変換を開始する前に行ってください。較正は、プロセスのばらつきによりチップごとに異なることがあるオフセットエラーを削除します。

較正は、ビット ADCAL=1 をセットすることによってソフトウェアによって開始されます。較正は、ADC が無効のとき (ADEN=0 のとき) だけ開始できます。すべての較正シーケンス時、ADCAL ビットは 1 のままです。較正が完了すると、ハードウェアによってクリアされます。その後、較正係数を ADC_DR レジスタ (ビット 6 ~ 0) から読み出すことができます。

ADC が無効の場合 (ADEN=0)、内部アナログ較正が保持されます。ADC 動作条件が変化したときには (V_{DDA} の変化が ADC オフセットのばらつきの主因であり、温度変化はそれほど影響しません)、較正サイクルを再実行することが推奨されます。

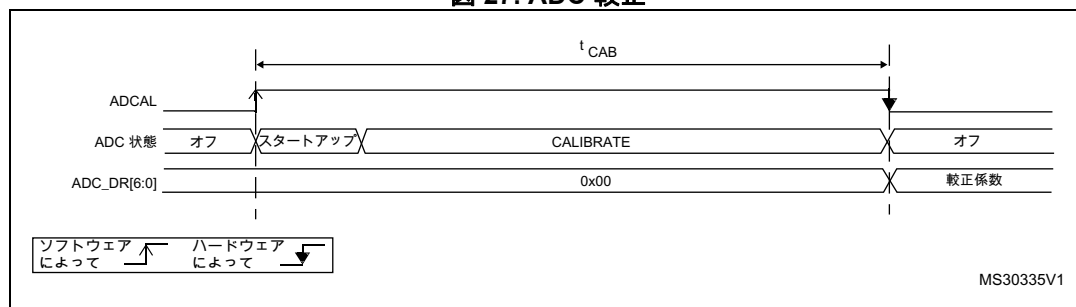
較正係数は ADC の電源がオフになるたびに失われます (たとえば、製品が STANDBY モードになった場合)。

較正ソフトウェア手順

1. ADEN=0であることを確認します。
2. ADCAL=1にセットします。
3. ADCAL=0になるまで待ちます。
4. 較正係数を ADC_DR のビット 6:0 から読み出すことができます。

コード例については、「付録」セクション [A.7.1 : ADC 較正のコード例](#) を参照してください。

図 27. ADC 較正



13.4.2 ADC オン / オフ制御 (ADEN、ADDIS、ADRDY)

MCU のパワーアップ時、ADC は無効になり、パワーダウンモードになります (ADEN=0)。

図 28 に示すように、ADC は正確な変換を開始する前に、安定時間 t_{STAB} を必要とします。

ADC を有効化または無効化するには、2 つの制御ビットが使用されます。

- ADC を有効にするには、ADEN=1 をセットします。ADC の動作準備ができると、ADRDY フラグがセットされます。
- ADC を無効にして、ADC をパワーダウンモードにするには、ADDIS=1 をセットします。ADC が完全に無効になると、ADEN および ADDIS ビットはハードウェアによって自動的にクリアされます。

その後、ADSTART=1 をセットすることによって (セクション 13.5 : 外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN) (241 ページ) を参照)、またはトリガが有効な場合は外部トリガイベントが発生したときに、変換を開始できます。

ADC を有効にするには、次の手順を実行します。

1. ADC_CR レジスタの ADEN=1 をクリアします。
2. ADC_ISR レジスタの ADRDY=1 になるまで待ちます (ADRDY は ADC 起動時間後にセットされます)。これは、ADC_IER レジスタの ADRDYIE ビットをセットすることによって割り込みを有効にした場合、割り込みによって処理できます。

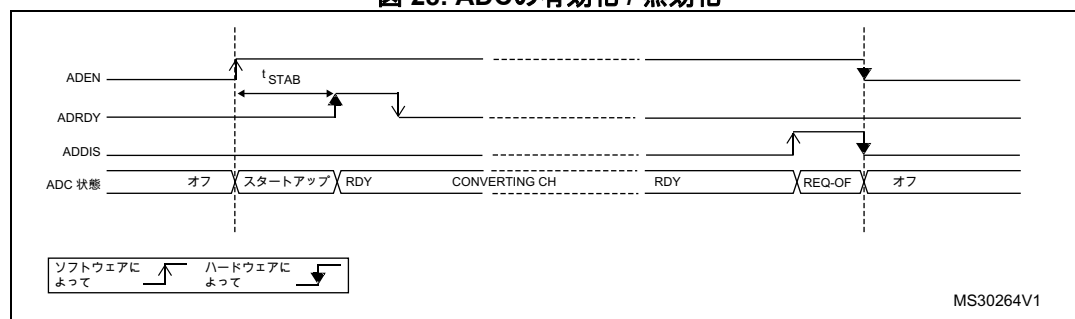
コード例については、「付録」セクション [A.7.2 : ADC 有効化シーケンスのコード例](#) を参照してください。

ADC を無効にするには、次の手順を実行します。

1. ADC_ISR レジスタの ADRDY ビットを"1"に設定してクリアします。
2. ADC_CR レジスタの ADSTART=0 を確認して、変換が実行中でないことを確認します。必要な場合は、ADC_CR レジスタの ADSTP ビットに 1 を書き込み、このビットが 0 として読み出されるまで待つことによって、実行中の変換を停止します。
3. ADC_CR レジスタの ADDIS=1 をセットします。
4. アプリケーションによって必要とされる場合、ADC_CR レジスタの ADEN=0 になり、ADC が完全に無効であることを示すまで待ちます (ADEN=0 になると、ADDIS は自動的にリセットされます)。
5. ADC_ISR レジスタの ADRDY ビットを"1"に設定してクリアします (オプション)。

コード例については、「付録」セクション [A.7.3 : ADC 無効化シーケンスのコード例](#) を参照してください。

図 28. ADCの有効化 / 無効化

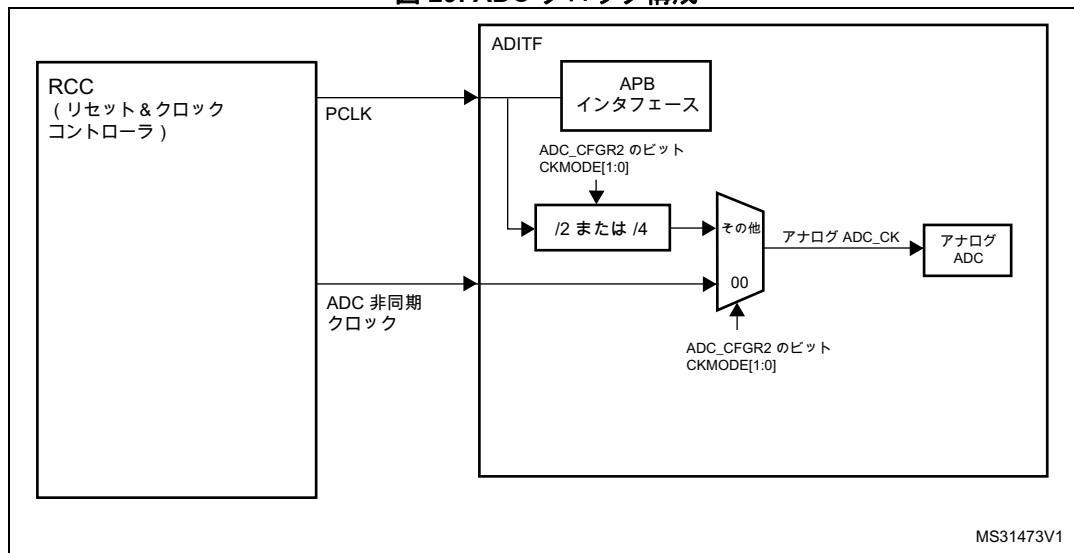


注： オートオフモード (AUTOFF=1) の場合、パワーオン/オフフェーズはハードウェアによって自動的に実行され、ADRDY フラグはセットされません。

13.4.3 ADC クロック (CKMODE)

ADC はデュアルクロックドメインアーキテクチャを持つので、ADC に APB クロック (PCLK) から独立したクロック (ADC 非同期クロック) を与えることができます。

図 29. ADC クロック構成



1. PCLK および ADC 非同期クロックを有効にする方法については、[セクション 7: リセットおよびクロック制御 \(RCC\) \(100 ページ\)](#) を参照してください。

アナログ ADC の入力クロックは、2 つのクロックソースから選択できます (PCLK および ADC 非同期クロックを有効にする方法については、[図 29 : ADC クロック構成](#)を参照してください)。

- a) ADC クロックは、APB クロックから独立し、非同期の「ADC 非同期クロック」という名前の特定のクロックソースにできます。
- このクロックソースの生成の詳細については、RCC のセクションを参照してください。
- この構成を選択するには、ADC_CFGR2 レジスタのビット CKMODE[1:0] をリセットする必要があります。
- コード例については、「付録」セクション [A.7.4 : ADC クロック選択のコード例](#)を参照してください。
- b) ADC クロックは ADC バスインタフェースの APB クロックから取得して、ビット CKMODE[1:0] に従ってプログラム可能な係数 (2 または 4) で分周することができます。
- この構成を選択するには、ADC_CFGR2 レジスタのビット CKMODE[1:0] が "00" 以外の値である必要があります。

オプション a) には、選択された APB クロック構成にかかわらず、最大の ADC クロック周波数に達するという利点があります。

オプション b) には、クロックドメインの再同期を迂回するという利点があります。これは、ADC がタイマによってトリガされるときと、アプリケーションが ADC の確実で精密なトリガを必要とする場合に便利で (そうしないと、トリガインスタンスの不確実性は、2 つのクロックドメイン間の再同期によって高まります)。

表 41. トリガから変換開始までの遅延

ADC クロックソース	CKMODE[1:0]	トリガイメントから変換開始までの遅延
専用の 14 MHz クロック	00	遅延は決定的ではありません (ジッタ)。
PCLK は 2 分周されます。	01	遅延は決定的であり (ジッタではない)、2.75 ADC クロックサイクルに等しい。
PCLK は 4 分周されます。	10	遅延は決定的であり (ジッタではない)、2.625 ADC クロックサイクルに等しい。

13.4.4 ADC の設定

ADC が無効な場合、ソフトウェアは ADC_CR レジスタの ADCAL および ADEN ビットに書き込む必要があります (ADEN は 0 である必要があります)。

ADC が有効であり、ADC を無効にするための保留中のリクエストがない場合のみ (ADEN=1 かつ ADDIS=0)、ソフトウェアは ADC_CR レジスタの ADSTART および ADDIS ビットに書き込む必要があります。

ADC_IER、ADC_CFGRi、ADC_SMPR、ADC_TR、ADC_CHSELR、および ADC_CCR レジスタのその他のすべての制御ビットについては、ADC が有効な場合 (ADEN = 1) と変換中でない場合 (ADSTART = 0) のみ、ソフトウェアは設定制御ビットに書き込む必要があります。

ADC が有効であり (おそらく変換中)、ADC を無効にするための保留中のリクエストがない場合のみ (ADSTART=1 かつ ADDIS=0)、ソフトウェアは ADC_CR レジスタの ADSTP ビットに書き込む必要があります。

注 : *上記のルールによって禁じられた書き込み操作をソフトウェアが行わないようにするハードウェア保護はありません。そのような禁止された書き込みアクセスが発生した場合、ADC は未定義の状態になることがあります。この場合に正しい動作を回復するには、ADC を無効にする必要があります (ADEN=0 と ADC_CR レジスタのすべてのビットをクリアします)。*

13.4.5 チャネル選択 (CHSEL、SCANDIR)

最大 19 の多重化チャネルがあります。

- GPIO ピンからの 16 のアナログ入力 (ADC_IN0...ADC_IN15)
- 3 つの内部アナログ入力 (温度センサ、内部基準電圧、V_{BAT} チャネル)

単一チャネルを変換でき、または一連のチャネルを自動的にスキャンできます。

変換されるチャネルのシーケンスは、ADC_CHSELR チャネル選択レジスタでプログラムする必要があります。各アナログ入力チャネルに専用の選択ビットがあります (CHSEL0...CHSEL18)。

チャネルがスキャンされる順序は、ADC_CFGR1 レジスタの SCANDIR ビットをプログラムすることによって設定できます。

- SCANDIR=0 : チャネル 0 からチャネル 18 へ前方スキャン
- SCANDIR=1 : チャネル 18 からチャネル 0 へ後方スキャン

温度センサ、V_{REFINT} および V_{BAT} 内部チャネル

温度センサは、チャネル ADC_IN16 に接続されます。内部電圧基準 V_{REFINT} は、チャネル ADC_IN17 に接続されます。V_{BAT} チャネルは ADC_IN18 チャネルに接続されています。

13.4.6 プログラム可能なサンプリング時間 (SMP)

変換を開始する前に、ADC は測定する電圧ソースと ADC の内蔵サンプリングコンデンサの間の直接接続を確立する必要があります。このサンプリング時間は、入力電圧ソースがサンプルをチャージし、コンデンサが入力電圧レベルを保持できるだけの十分な長さが必要です。

プログラム可能なサンプリング時間によって、入力電圧ソースの入力抵抗に従って変換速度を微調整することができます。

ADC は、一定の ADC クロックサイクル数だけ入力電圧をサンプリングしますが、この時間は、ADC_SMPR レジスタの SMP[2:0] ビットを使用して変更できます。

このプログラム可能なサンプリング時間は、すべてのチャネルに共通です。アプリケーションによって必要とされる場合、ソフトウェアは変換ごとにこのサンプリング時間を変更して適応できます。

合計変換時間は、次のように計算されます。

$$t_{\text{CONV}} = \text{サンプリング時間} + 12.5 \times \text{ADC クロックサイクル}$$

例 :

$$\text{ADC\_CLK} = 14 \text{ MHz, サンプリング時間} = 1.5 \text{ ADC クロックサイクル} :$$

$$t_{\text{CONV}} = 1.5 + 12.5 = 14 \text{ ADC クロックサイクル} = 1 \mu\text{s}$$

ADC は、EOSMP フラグをセットすることによって、サンプリングフェーズの終了を示します。

13.4.7 シングル変換モード (CONT=0)

シングル変換モードでは、ADC は単一シーケンスの変換を実行して、すべてのチャンネルを一度変換します。このモードは ADC_CFGR1 レジスタの CONT=0 のときに選択されます。変換は、次のいずれかによって開始されます。

- ADC_CR レジスタの ADSTART ビットのセット
- ハードウェアトリガイイベント

シーケンス内で、各変換の完了後、

- 変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
- EOC (end of conversion) フラグがセットされます。
- EOCIE ビットがセットされている場合、割り込みが生成されます。

変換シーケンスの完了後、

- EOSEQ (end of sequence) フラグがセットされます。
- EOSEQIE ビットがセットされている場合、割り込みが生成されます。

次に、ADC は、新しい外部トリガイイベントが発生するか、ADSTART ビットが再びセットされるまで停止します。

注： 単一チャンネルを変換するには、長さが1のシーケンスをプログラムします。

13.4.8 連続変換モード (CONT=1)

連続変換モードでは、ソフトウェアまたはハードウェアトリガイイベントが発生すると、ADC は一連の変換を実行して、すべてのチャンネルを一度変換した後、自動的に再起動して、同じ変換シーケンスを連続的に実行します。このモードは ADC_CFGR1 レジスタの CONT=1 のときに選択されます。変換は、次のいずれかによって開始されます。

- ADC_CR レジスタの ADSTART ビットのセット
- ハードウェアトリガイイベント

シーケンス内で、各変換の完了後、

- 変換されたデータは 16 ビットの ADC_DR レジスタに格納されます。
- EOC (end of conversion) フラグがセットされます。
- EOCIE ビットがセットされている場合、割り込みが生成されます。

変換シーケンスの完了後、

- EOSEQ (end of sequence) フラグがセットされます。
- EOSEQIE ビットがセットされている場合、割り込みが生成されます。

次に、新しいシーケンスをすぐに再開して、ADC は変換シーケンスを連続的に繰り返します。

注： 単一チャンネルを変換するには、長さが1のシーケンスをプログラムします。

不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方のビットをセットすることは禁じられています。

13.4.9 変換の開始 (ADSTART)

ソフトウェアは、ADSTART=1 をセットすることによって ADC 変換を開始します。

ADSTART がセットされると、変換は、

- EXTEN = 0x0 (ソフトウェアトリガ) の場合、すぐに開始します。
- EXTEN¹ 0x0 の場合、選択されたハードウェアトリガの次のアクティブエッジで開始します。

ADSTART ビットは、ADC の動作が実行中かどうかを示すためにも使用されます。ADSTART=0 であり、ADC がアイドルであることを示しているときには、ADC を再設定できます。

ADSTART ビットは、ハードウェアによってクリアされます。

- ソフトウェアトリガによるシングルモードのとき (CONT=0、EXTEN=0x0)
 - 連続シーケンスの終了時に (EOSEQ=1)
- すべての場合に (CONT=x、EXTEN=x)
 - ソフトウェアによって起動された ADSTP 手順の実行後 ([セクション 13.4.11: 実行中の変換の停止 \(ADSTP\) \(240 ページ\)](#) を参照)

注： 連続モード (CONT=1) では、EOSEQ フラグがセットされたとき、シーケンスは自動的に再起動されるので、ADSTART ビットはハードウェアによってクリアされません。

シングルモードでハードウェアトリガが選択されたとき (CONT=0 かつ EXTEN¹ 0x00)、EOSEQ フラグがセットされたとき、ADSTART はハードウェアによってクリアされません。このため、ソフトウェアは ADSTART ビットを再びセットする必要がなく、次のトリガイベントを見逃す恐れがありません。

13.4.10 タイミング

変換の開始から変換の終了までの経過時間は、設定されたサンプリング時間に逐次比較時間 (データ分解能に依存) を加えた合計です。

$$t_{ADC} = t_{SMPL} + t_{SAR} = [1.5 |_{min} + 12.5 |_{12bit}] \times t_{ADC\_CLK}$$

$$t_{ADC} = t_{SMPL} + t_{SAR} = 107.1 \text{ ns} |_{min} + 892.8 \text{ ns} |_{12bit} = 1 \mu\text{s} |_{min} (f_{ADC\_CLK} = 14 \text{ MHz の場合})$$

図 30. アナログ / デジタル変換時間

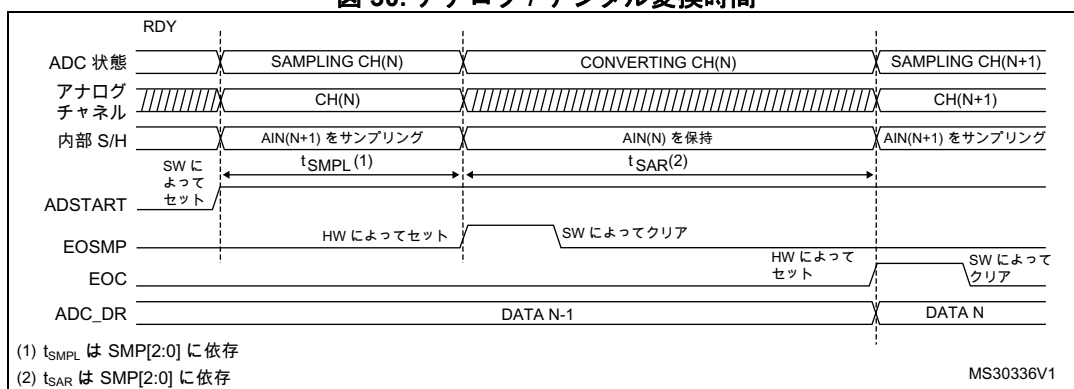
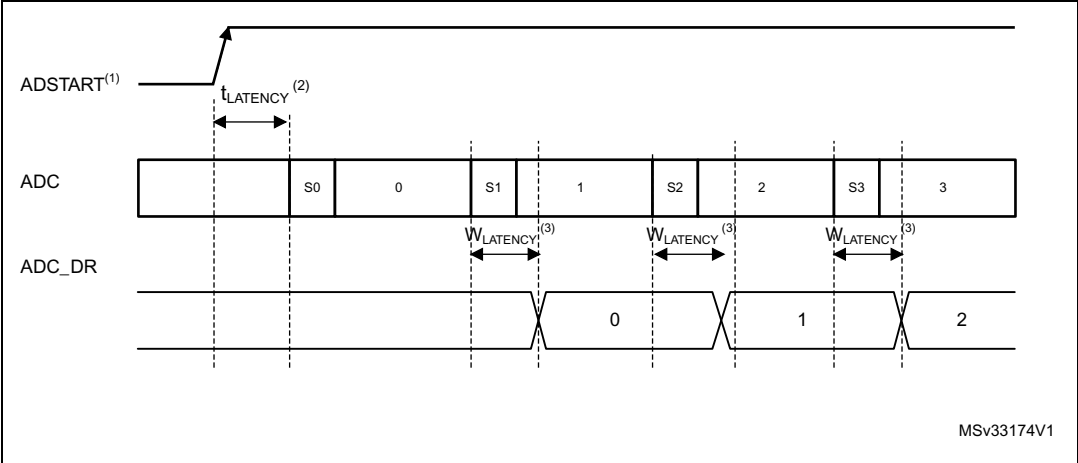


図 31. ADC 変換タイミング



1. EXTEN = 00 または EXTEN¹ 00
2. トリガ遅延 (詳細についてはデータシートを参照)
3. ADC_DR レジスタ書き込み遅延 (詳細についてはデータシートを参照)

13.4.11 実行中の変換の停止 (ADSTP)

ソフトウェアは、ADC_CR レジスタの ADSTP=1 をセットすることによって、実行中の変換を停止することができます。

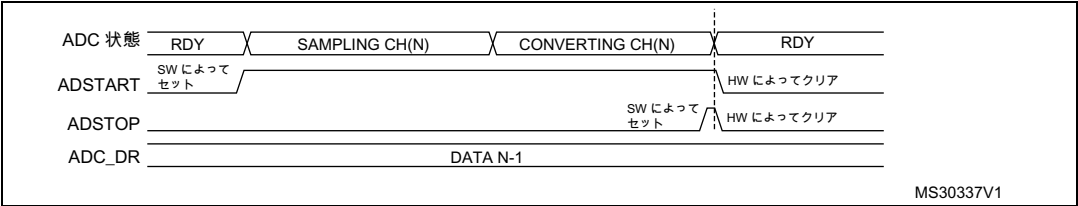
これによって ADC 動作がリセットされ、ADC はアイドルになり、新しい動作の準備ができます。

ADSTP ビットがソフトウェアによってセットされると、実行中の変換は中止され、結果は破棄されます (ADC_DR レジスタは現在の変換で更新されません)。

スキャンシーケンスも中止され、リセットされます (ADC を再起動すると、新しいシーケンスが再開されることを意味します)。

この手順が完了すると、ADSTP および ADSTART ビットの両方がハードウェアによってクリアされ、ソフトウェアは新しい変換を開始する前に ADSTART=0 になるまで待つ必要があります。

図 32. 実行中の変換の停止



13.5 外部トリガおよびトリガ極性での変換 (EXTSEL、EXTEN)

変換または変換シーケンスは、ソフトウェアによって、または外部イベント（タイマキャプチャなど）によってトリガされます。EXTEN[1:0] 制御ビットが “0b00” に等しくない場合、外部イベントは選択された極性で変換をトリガできます。トリガ選択は、ソフトウェアがビット ADSTART=1 をセットすると有効になります。

変換中に発生したハードウェアトリガは無視されます。

ビット ADSTART=0 の場合、発生したハードウェアトリガは無視されます。

表 42 にEXTEN[1:0] の値とトリガ極性の対応を示します。

表 42. トリガ極性の設定

転送元	EXTEN[1:0]
トリガ検出は無効です。	00
立ち上がりエッジで検出します。	01
立ち下がりエッジで検出します。	10
立ち上がりエッジと立ち下がりエッジの両方で検出します。	11

注： 外部トリガの極性は、ADC が変換中でない (ADSTART=0) のときだけ変更できます。

EXTSEL[2:0] 制御ビットは、8 つの可能なイベントのうち、変換をトリガするイベントを選択するために使用されます。

表 43 に、レギュラ変換に使用できる外部トリガを示します。

ソフトウェアソーストリガイベントは、ADC_CR レジスタの ADSTART ビットをセットすることによって生成できます。

表 43. 外部トリガ

名前	転送元	EXTSEL[2:0]
TRG0	TIM1_TRGO	000
TRG1	TIM1_CC4	001
TRG2	TIM2_TRGO	010
TRG3	TIM3_TRGO	011
TRG4	TIM15_TRGO	100
TRG5	予約済み	101
TRG6	予約済み	110
TRG7	予約済み	111

注： トリガ選択は、ADC が変換中でない (ADSTART=0) のときだけ変更できます。

13.5.1 不連続モード (DISCEN)

このモードは、ADC_CFGR1 レジスタの DISCEN ビットをセットすることによって有効になります。

このモード (DISCEN=1) では、シーケンスで定義された各変換を開始するには、ハードウェアまたはソフトウェアトリガイベントが必要です。逆に、DISCEN=0 の場合は、単一のハードウェアまたはソフトウェアトリガイベントが、シーケンス内で定義されたすべての変換を連続的に開始します。

例：

- DISCEN=1、変換されるチャネル = 0、3、7、10
 - 最初のトリガ：チャネル 0 が変換され、EOC イベントが生成されます。
 - 2 番目のトリガ：チャネル 3 が変換され、EOC イベントが生成されます。
 - 3 番目のトリガ：チャネル 7 が変換され、EOC イベントが生成されます。
 - 4 番目のトリガ：チャネル 10 が変換され、EOC イベントと EOSEQ イベントの両方が生成されます。
 - 5 番目のトリガ：チャネル 0 が変換され、EOC イベントが生成されます。
 - 6 番目のトリガ：チャネル 3 が変換され、EOC イベントが生成されます。
 - ...
- DISCEN=0、変換されるチャネル = 0、3、7、10
 - 最初のトリガ：シーケンス全体、すなわち、チャネル 0、3、7、および 10 が変換されます。各変換後に EOC イベントが生成され、最後の変換後には EOSEQ イベントも生成されます。
 - 後続のトリガイベントがあると、シーケンス全体が再開されます。

注： 不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方のビットをセットすることは禁じられています。

13.5.2 プログラム可能な分解能 (RES) - 高速変換モード

ADC の分解能を下げることによって、高速な変換時間 (t_{SAR}) が可能になります。

分解能は、ADC_CFGR1 レジスタの RES[1:0] ビットをプログラムすることによって、12、10、8、または 6 ビットに設定できます。分解能を下げることによって、高いデータ精度を必要としないアプリケーションの変換時間を高速にできます。

注： RES[1:0] ビットは、ADEN ビットがリセットされたときだけ変更する必要があります。

変換結果は常に 12 ビット幅であり、未使用の LSB ビットはゼロとして読み出されます。

分解能を下げると、表 44 に示すように、逐次比較ステップに必要な変換時間が短縮されます。

表 44. t_{SAR} タイミングは分解能に依存

RES[1:0] ビット	t_{SAR} (ADC クロック サイクル)	t_{SAR} (ns) ($f_{ADC} = 14$ MHz 時)	t_{SMPL} (min) (ADC クロック サイクル)	t_{ADC} (ADC クロック サイクル) (最小 t_{SMPL})	t_{ADC} (μs) ($f_{ADC} = 14$ MHz 時)
12	12.5	893 ns	1.5	14	1000 ns
10	11.5	821 ns	1.5	13	928 ns
8	9.5	678 ns	1.5	11	785 ns
6	7.5	535 ns	1.5	9	643 ns

13.5.3 変換の終了、サンプリングフェーズの終了 (EOC、EOSMP フラグ)

ADC は、各変換の終了 (EOC) イベントを示します。

ADC は、新しい変換データ結果が ADC_DR レジスタで使用可能になるとすぐに、ADC_ISR レジスタの EOC フラグをセットします。ADC_IER レジスタの EOCIE ビットがセットされている場合、割り込みが生成されます。EOC フラグは、ソフトウェアによって 1 を書き込むことによって、または ADC_DR レジスタを読み出すことによってクリアされます。

ADC は、ADC_ISR レジスタの EOSMP フラグをセットすることによって、サンプリングフェーズの終了も示します。EOSMP フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。ADC_IER レジスタの EOSMPIE ビットがセットされている場合、割り込みが生成されます。

この割り込みの目的は、処理を変換と同期するためです。一般に、アナログマルチプレクサには変換フェーズ中の隠された時間でアクセスできるため、マルチプレクサは次のサンプリング開始時に位置づけられます。

注： サンプリングの終了から変換の終了までは非常に短時間しか残されていないため、割り込みと WFI 命令ではなく、ポーリングまたは WFE 命令を使用することが推奨されます。

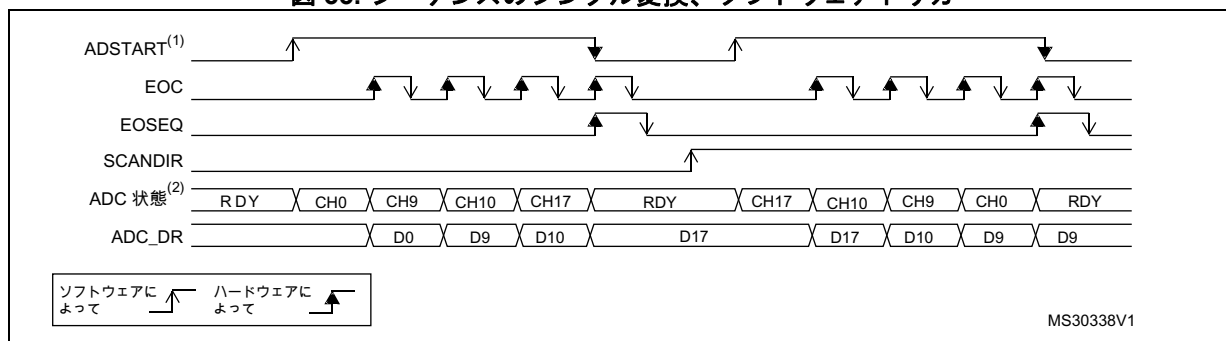
13.5.4 変換シーケンスの終了 (EOSEQ フラグ)

ADC は各シーケンスの終了 (EOSEQ) イベントをアプリケーションに通知します。

ADC は、変換の最後のデータ結果が ADC_DR レジスタで使用可能になるとすぐに、ADC_ISR レジスタの EOSEQ フラグをセットします。ADC_IER レジスタの EOSEQIE ビットがセットされている場合、割り込みが生成されます。EOSEQ フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

13.5.5 タイミング図の例 (シングル / 連続モードハードウェア / ソフトウェアトリガ)

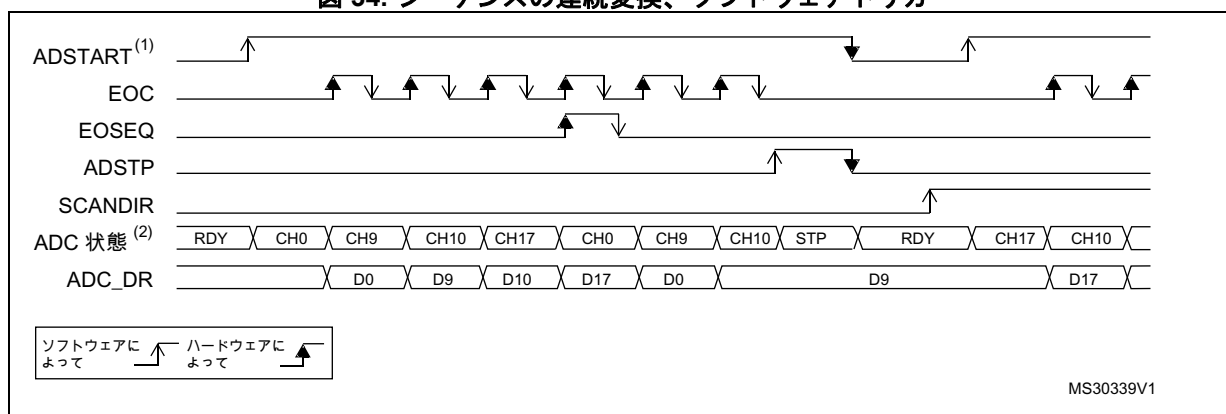
図 33. シーケンスのシングル変換、ソフトウェアトリガ



1. EXTEN=0x0、CONT=0
2. CHSEL=0x20601、WAIT=0、AUTOFF=0

コード例については、「付録」セクション [A.7.5 : シングル変換シーケンスのコード例 - ソフトウェアトリガ](#) を参照してください。

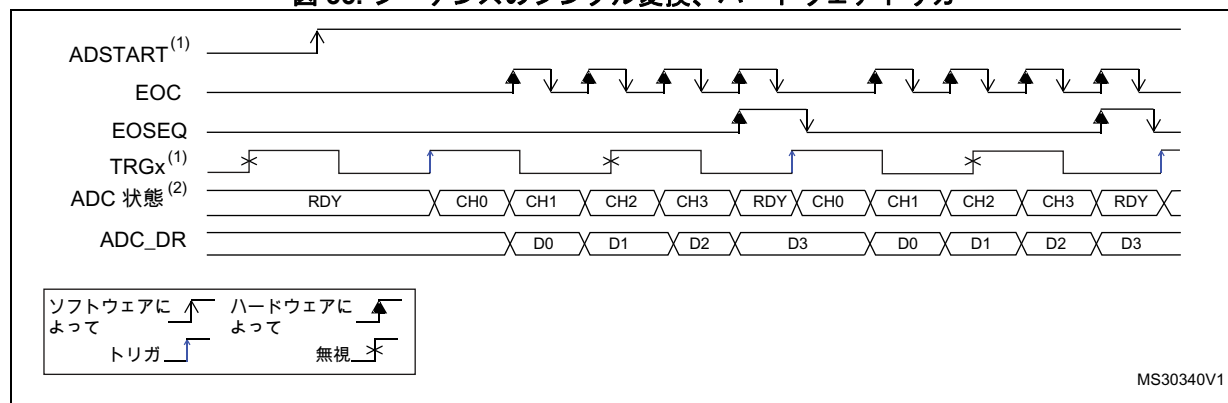
図 34. シーケンスの連続変換、ソフトウェアトリガ



1. EXTEN=0x0、CONT=1、
2. CHSEL=0x20601、WAIT=0、AUTOFF=0

コード例については、「付録」セクション [A.7.6 : 連続変換シーケンスのコード例 - ソフトウェアトリガ](#) を参照してください。

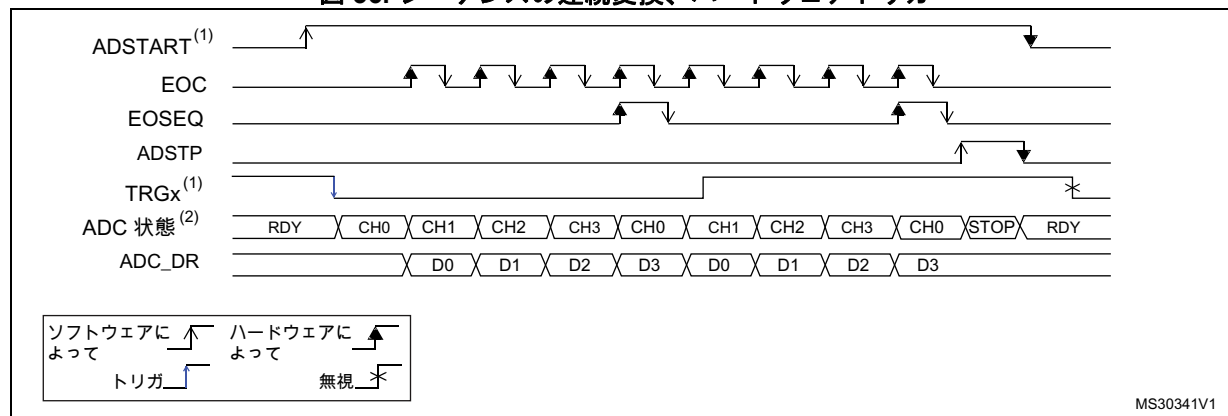
図 35. シーケンスのシングル変換、ハードウェアトリガ



- EXTSEL=TRGx (オーバー周波数)、EXTEN=0x1 (立ち上がりエッジ)、CONT=0
- CHSEL=0xF、SCANDIR=0、WAIT=0、AUTOFF=0

コード例については、「付録」セクション [A.7.7: シングル変換シーケンスのコード例 - ハードウェアトリガ](#)を参照してください。

図 36. シーケンスの連続変換、ハードウェアトリガ



- EXTSEL=TRGx、EXTEN=0x2 (立ち下がりエッジ)、CONT=1
- CHSEL=0xF、SCANDIR=0、WAIT=0、AUTOFF=0

コード例については、「付録」セクション [A.7.8: 連続変換シーケンスのコード例 - ハードウェアトリガ](#)を参照してください。

13.6 データ管理

13.6.1 データレジスタおよびデータの配置 (ADC_DR、ALIGN)

各変換の終了時 (EOC イベントの発生時)、変換されたデータの結果は 16 ビット幅の ADC_DR データレジスタに格納されます。

ADC_DR のフォーマットは、設定されたデータ配置と分解能に依存します。

ADC_CFGR1 レジスタの ALIGN ビットは、変換後に格納されるデータの配置を選択します。図 37 に示すように、データは右詰め (ALIGN=0) または左詰め (ALIGN=1) にできます。

図 37. データの配置と分解能

ALIGN	RES	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0x0	DR[11:0]															
	0x1	DR[9:0]															
	0x2	DR[7:0]															
	0x3	DR[5:0]															
1	0x0	DR[11:0]															
	0x1	DR[9:0]															
	0x2	DR[7:0]															
	0x3	DR[5:0]															

MS30342V1

13.6.2 ADC オーバーラン (OVR、OVRMOD)

オーバーランフラグ (OVR) は、新しい変換からのデータが使用可能になる前に、変換されたデータが CPU または DMA によって時間内に読み出されなかったときに、データオーバーランを示します。

OVR フラグは、新しい変換が完了した時点で EOC フラグが '1' のままであった場合に、ADC_ISR レジスタでセットされます。ADC_IER レジスタの OVRIE ビットがセットされている場合、割り込みが生成されます。

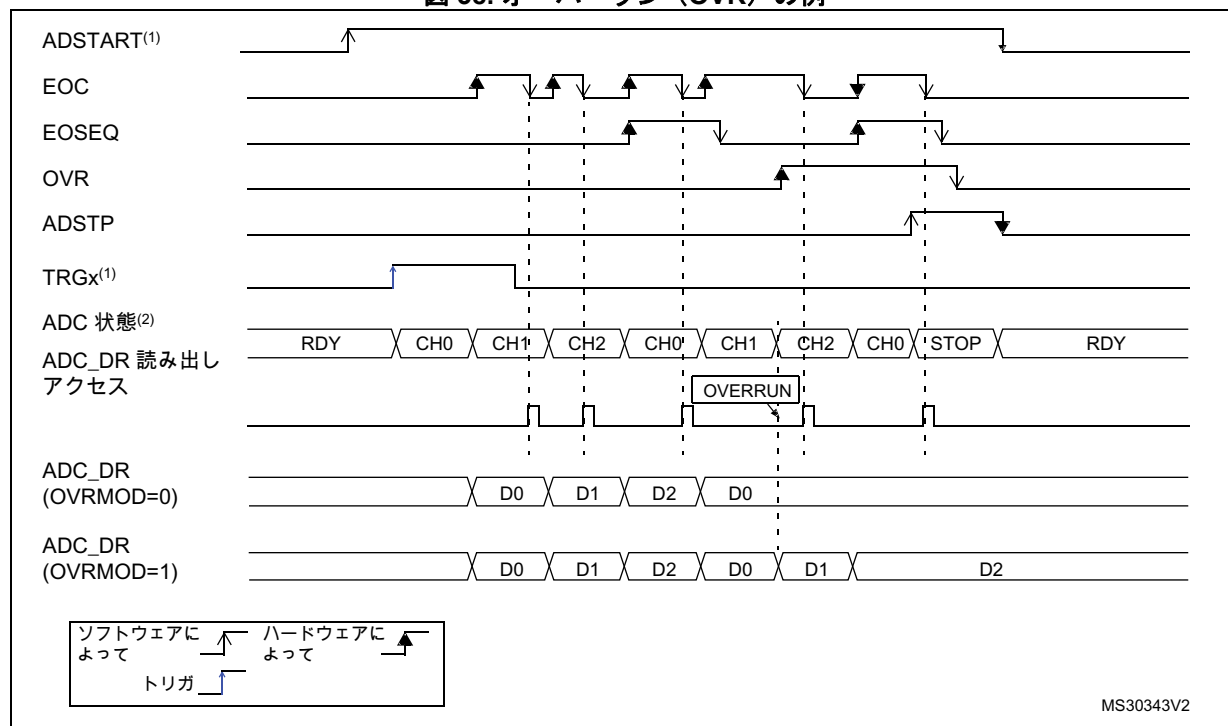
オーバーラン条件が発生すると、ADC は動作し続け、ソフトウェアが ADC_CR レジスタの ADSTP ビットをセットすることによってシーケンスの停止とリセットを決めるまで、変換を続行できます。

OVR フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

ADC_CFGR1 レジスタの OVRMOD ビットをプログラムすることによって、オーバーランイベントが発生したときにデータが保存されるか上書きされるかを設定できます。

- OVRMOD=0
 - － オーバーランイベントが発生しても、データレジスタは上書きされません。古いデータは保持され、新しい変換は破棄されます。OVR が 1 のままの場合、さらに変換を実行できますが、結果データは破棄されます。
- OVRMOD=1
 - － データレジスタは最後の変換結果で上書きされ、以前の未読データは失われます。OVR が 1 のままの場合、さらに変換を実行でき、ADC_DR レジスタは常に最新の変換からのデータを含みます。

図 38. オーバーラン (OVR) の例



13.6.3 DMA を使用しない変換データシーケンスの管理

変換が十分に遅い場合、ソフトウェアで変換シーケンスを処理することができます。この場合、ソフトウェアは EOC フラグと関連の割り込みを使用して、各データ結果を処理する必要があります。変換が完了するたびに、ADC_ISR レジスタの EOC ビットがセットされ、ADC_DR レジスタを読み出すことができます。オーバーラインイベントをエラーとして管理するには、ADC_CFGR1 レジスタの OVRMOD ビットを 0 に設定する必要があります。

13.6.4 オーバーランなしでの DMA を使用しない変換データの管理

変換のたびにデータの読み出しをせずに ADC に 1 つまたは複数のチャンネルを変換させると便利な場合があります。この場合、OVRMOD ビットを 1 に設定する必要があります。ソフトウェアは OVR フラグを無視する必要があります。OVRMOD=1 のとき、オーバーラインイベントが発生しても ADC は変換を続行し、ADC_DR レジスタは常に最新の変換データを含みます。

13.6.5 DMA を使用した変換データの管理

変換されたレギュラチャンネルの値はすべて、単一のデータレジスタに格納されるので、複数のチャンネルを変換するときには DMA を使用すると効率的です。これによって、ADC_DR レジスタに格納されている変換データ結果が失われるのを避けることができます。

DMA モードが有効なとき (ADC_CFGR1 レジスタの DMAEN ビットが 1 にセットされている)、各チャンネルの変換後、DMA リクエストが生成されます。これにより、変換されたデータを ADC_DR レジスタからソフトウェアで選択した場所へ転送することができます。

注: ADC_CFGR1 レジスタの DMAEN ビットは、ADC 較正フェーズ後にセットする必要があります。

これにもかかわらず、DMA が DMA 転送リクエストを時間内に処理できなかったためにオーバーランが発生した場合 (OVR=1)、ADC は DMA リクエストの生成を停止し、新しい変換に対応するデータ

は DMA によって転送されません。これは、RAM に転送されるすべてのデータを有効とみなすことができることを意味します。

OVRMOD ビットの設定に応じて、データは保存または上書きされます ([セクション 13.6.2 : ADC オーバーラン \(OVR、OVRMOD\) \(246 ページ\)](#) を参照してください)。

DMA 転送リクエストは、ソフトウェアが OVR ビットをクリアするまでブロックされます。

アプリケーションの用途に応じて 2 つの DMA モードがあり、ADC_CFGR1 レジスタのビット DMACFG で設定されます。

- DMA ワンショットモード (DMACFG=0)。
DMA が固定数のデータワードを転送するようにプログラムされたときには、このモードを選択してください。
- DMA サーキュラモード (DMACFG=1)
DMA をサーキュラモードまたはダブルバッファモードでプログラムするときには、このモードを選択してください。

DMA ワンショットモード (DMACFG=0)

このモードでは、ADC は新しい変換データワードが使用可能になるたびに DMA 転送リクエストを生成し、変換が再び開始された場合でも、DMA が最後の DMA 転送に達すると (DMA_EOT 割り込みが発生すると) ([セクション 11 : ダイレクトメモリアクセスコントローラ \(DMA\) \(197 ページ\)](#) を参照)、DMA リクエストの生成を停止します。

コード例については、「付録」セクション [A.7.9 : DMA ワンショットモードシーケンスのコード例](#) を参照してください。

DMA 転送完了後 (DMA コントローラで設定されたすべての転送後) :

- ADC データレジスタの内容が停止されます。
- 実行中の変換は中止され、その部分的な結果は破棄されます。
- DMA コントローラに対する新しいDMA リクエストは発行されません。これによって、開始された変換がある場合のオーバーランエラーの生成を回避します。
- スキャンシーケンスは中止され、リセットされます。
- DMA は停止します。

DMA サーキュラモード (DMACFG=1)

このモードでは、DMA が最後の DMA 転送に達した場合でも、ADC は新しい変換データワードがデータレジスタで使用可能になるたびに DMA 転送リクエストを生成します。これにより、DMA を連続的なアナログ入力データストリームを処理するようにサーキュラモードで設定できます。

コード例については、「付録」セクション [A.7.10 : DMA サーキュラモードシーケンスのコード例](#) を参照してください。

13.7 低電力機能

13.7.1 ウェイトモード変換

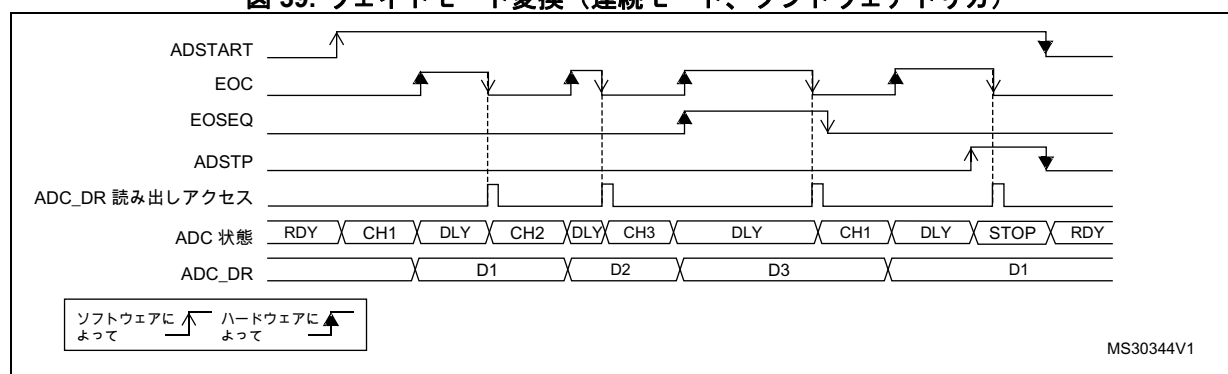
ウェイトモード変換を使用すると、ソフトウェアを単純化するだけでなく、ADC オーバーランが発生するリスクのある低周波数のクロックで動作しているアプリケーションのパフォーマンスを最適化できます。

ADC_CFGR1 レジスタの WAIT ビットが 1 にセットされているとき、新しい変換は、前のデータが処理された場合、ADC_DR レジスタが読み出された場合、または EOC ビットがクリアされた場合のみ開始できます。

これは、ADC の速度をデータを読み出すシステムの速度に自動的に適応させる方法です。

注： 変換中または読み出しアクセス前のウェイト時間中に発生したハードウェアトリガは無視されます。

図 39. ウェイトモード変換（連続モード、ソフトウェアトリガ）



1. EXTEN=1x0、CONT=0
2. CHSEL=0x3、SCANDIR=0、WAIT=1、AUTOFF=0

コード例については、「付録」セクション [A.7.11：ウェイトモードシーケンスのコード例](#) を参照してください。

13.7.2 オートオフモード (AUTOFF)

ADC にはオートオフモードと呼ばれる自動電源管理機能があり、ADC_CFGR1 レジスタの AUTOFF=1 をセットすることによって有効になります。

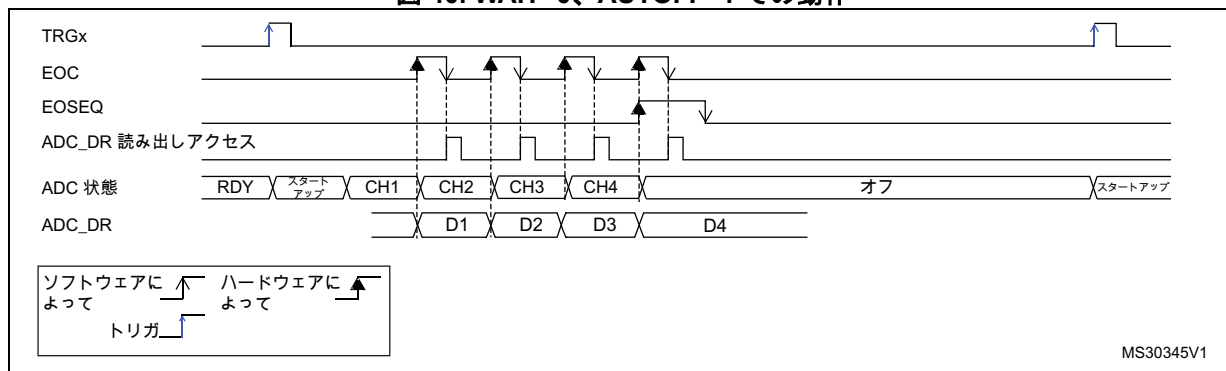
AUTOFF=1 のとき、ADC は、変換中でないときには常に電源がオフであり、（ソフトウェアまたはハードウェアトリガによって）変換が開始されると自動的にウェイクアップします。変換を開始するトリガイイベントと ADC のサンプリング時間の間に、スタートアップ時間が自動的に挿入されます。変換シーケンスが完了すると、ADC は自動的に無効になります。

オートオフモードは、アプリケーションが比較的少ない変換しか必要としないとき、または変換リクエストの間隔が十分に離れているとき（低周波数のハードウェアトリガによる場合など）に消費電力を大幅に削減できるため、ADC のオンとオフを切り替えるために余分な電力と余分な時間がかかっても引き合います。

低周波数のクロックで動作するアプリケーションの場合、オートオフモードとウェイトモード変換 (WAIT=1) を組み合わせることができます。この組み合わせは、ADC がウェイトフェーズで自動的に電源オフされ、ADC_DR レジスタがアプリケーションによって読み出されるとすぐに再起動する場合、大幅な節電を可能にします（[図 41：WAIT=1、AUTOFF=1 での動作](#) を参照してください）。

注： 専用 14 MHz 内部オシレータを管理する方法については、[セクション 7：リセットおよびクロック制御 \(RCC\) \(100 ページ\)](#) を参照してください。ADC インタフェースは、節電のために、14 MHz 内部オシレータの ON / OFF を自動的に切り替えることができます。

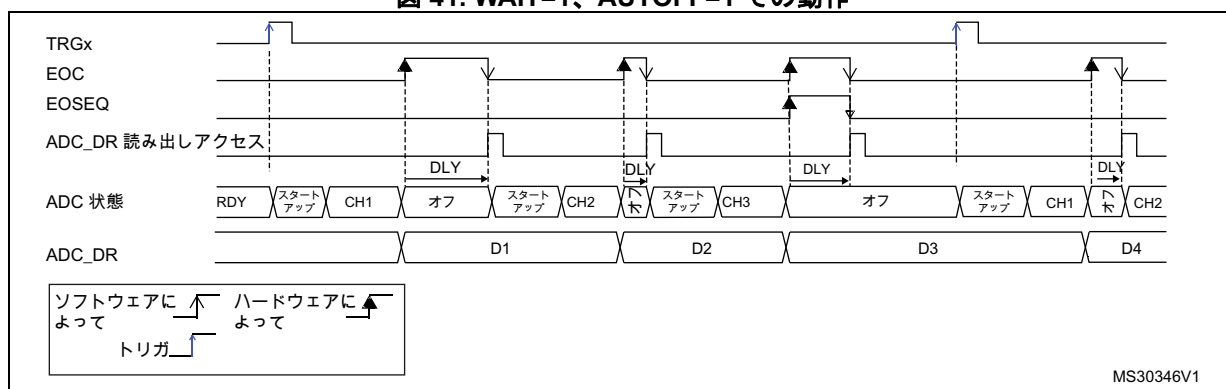
図 40. WAIT=0、AUTOFF=1 での動作



- EXTSEL=TRGx, EXTEN=0x1 (立ち上がりエッジ)、CONT=x, ADSTART=1, CHSEL=0xF, SCANDIR=0, WAIT=1, AUTOFF=1

コード例については、「付録」セクション [A.7.12：オートオフおよびノーウェイトモードシーケンスのコード例](#) を参照してください。

図 41. WAIT=1、AUTOFF=1 での動作



- EXTSEL=TRGx, EXTEN=0x1 (立ち上がりエッジ)、CONT=x, ADSTART=1, CHSEL=0xF, SCANDIR=0, WAIT=1, AUTOFF=1

コード例については、「付録」セクション [A.7.13：オートオフおよびウェイトモードシーケンスのコード例](#) を参照してください。

13.8 アナログウィンドウウォッチドッグ (AWDEN、AWDSGL、AWDCH、AWD_HTR/LTR、AWD)

AWD アナログウォッチドッグ機能は、ADC_CFGR1 レジスタの AWDEN ビットをセットすることによって有効になります。図 42 に示すように、選択された 1 つのチャンネルまたはすべての有効チャンネル（表 46：アナログウォッチドッグチャンネル選択を参照）が設定された電圧範囲（ウィンドウ）内にとどまっているかどうかを監視するために使用されます。

アナログウォッチドッグ (AWD) ステータスビットは、ADC によって変換されたアナログ電圧が低閾値を下回るか、高閾値を上回る場合にセットされます。これらの閾値は、ADC_HTR および ADC_LTR 16 ビットレジスタの下位 12 ビットに設定します。ADC_IER レジスタの AWDIE ビットをセットすることによって、割り込みを有効にできます。

AWD フラグは、ソフトウェアによって 1 を書き込むことによってクリアされます。

12 ビット未満の分解能でデータを変換するときには（ビット DRES[1:0] に従って）、内部比較は常に 12 ビット全体の 元の変換データ（左詰め）に対して実行されるため、プログラムされた閾値の LSB はクリアされたままである必要があります。

コード例については、「付録」セクション A.7.14：アナログウォッチドッグのコード例を参照してください。

表 45 に、可能なすべての分解能での比較方法を示します。

表 45. アナログウォッチドッグ比較

分解能ビット RES[1:0]	アナログウォッチドッグ比較：		コメント
	元の変換データ、 左詰め ⁽¹⁾	閾値	
00：12 ビット	DATA[11:0]	LT[11:0] と HT[11:0]	-
01：10 ビット	DATA[11:2],00	LT[11:0] と HT[11:0]	ユーザは LT1[1:0] と HT1[1:0] を “00” に設定する必要があります。
10：8 ビット	DATA[11:4],0000	LT[11:0] と HT[11:0]	ユーザは LT1[3:0] と HT1[3:0] を “0000” に設定する必要があります。
11：6 ビット	DATA[11:6],000000	LT[11:0] と HT[11:0]	ユーザは LT1[5:0] と HT1[5:0] を “000000” に設定する必要があります。

1. 配置計算の前に、元の変換データに対するウォッチドッグ比較が行われます。

表 46 に、ADC_CFGR1 レジスタの AWDSGL および AWDEN ビットを設定して、1 つ以上のチャンネルに対してアナログウォッチドッグを有効にする方法を示します。

図 42. アナログウォッチドッグによって保護される領域

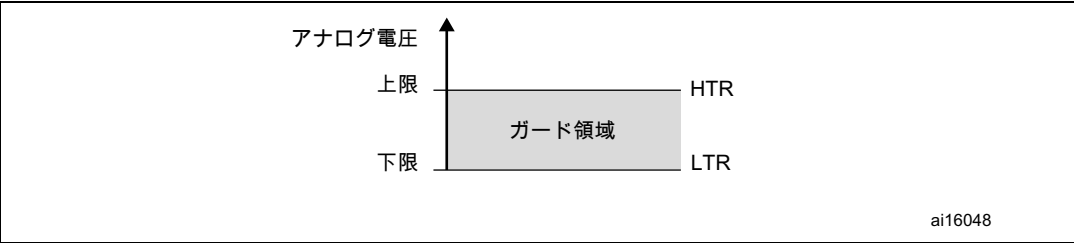


表 46. アナログウォッチドッグチャネル選択

アナログウォッチドッグによって保護されるチャネル	AWDSGL ビット	AWDEN ビット
なし	x	0
すべてのチャネル	0	1
単一の ⁽¹⁾ チャネル	1	1

1. AWDCH[4:0] ビットによって選択

13.9 温度センサと内部基準電圧

温度センサを使用して、デバイスの接合温度 (T_J) を測定できます。温度センサは、センサの出力電圧をデジタル値に変換する ADC_IN16 入力チャネルに内部接続されます。温度センサのアナログピンのサンプリング時間は、最小の T_{S_temp} より大きい値である必要があります。使用しないときには、センサをパワーダウンモードにすることができます。

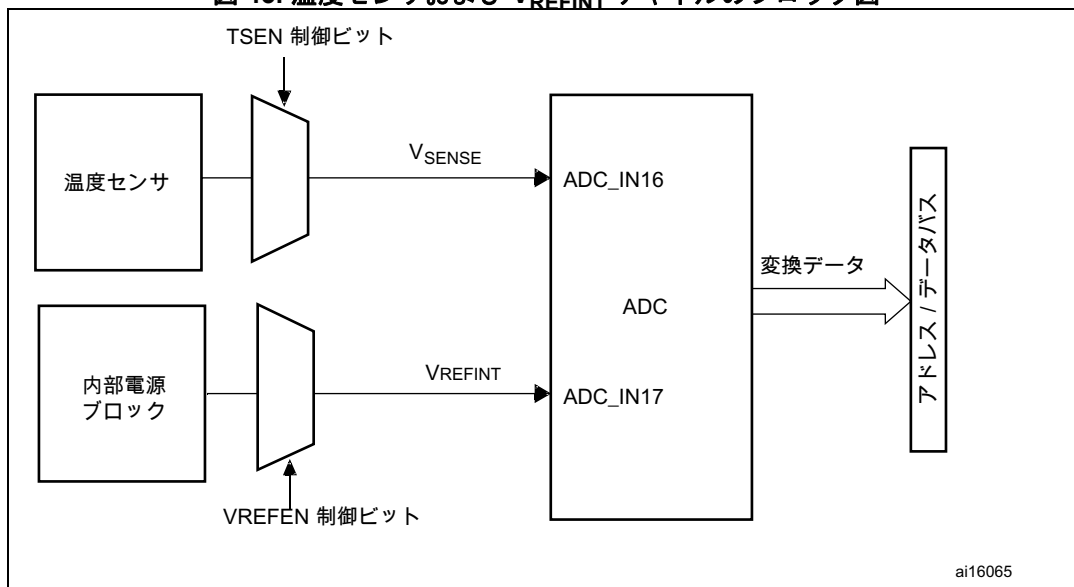
温度センサの出力電圧は、温度に比例して変化しますが、特性はプロセスのばらつきによりチップごとに大きく異なる場合があります。温度センサの測定精度を高めるために（特に絶対温度測定）、生産試験時に ST によって部品ごとに較正值が個別に測定され、システムメモリに格納されます。詳細については、特定のデバイスのデータシートを参照してください。

内部電圧基準 (VREFINT) は、ADC とコンパレータに安定した（バンドギャップ）電圧出力を提供します。VREFINT は、ADC_IN17 入力チャネルに内部接続されます。VREFINT の正確な電圧は、生産試験時に ST によって部品ごとに個別に測定され、システムメモリ領域に格納されます。

図 43 に、温度センサ、内部電圧基準、および ADC 間の接続のブロック図を示します。

ADC_IN16 (温度センサ) の変換を有効にするには、TSEN ビットをセットする必要があります。ADC_IN17 (VREFINT) の変換を有効にするには、VREFEN ビットをセットする必要があります。

図 43. 温度センサおよび VREFINT チャンネルのブロック図



温度の読み出し

1. ADC_IN16 入力チャネルを選択します。
2. デバイスのデータシートで指定されている適切なサンプリング時間 (T_{S_temp}) を選択します。
3. ADC_CCR レジスタの TSEN ビットをセットして、温度センサをパワーダウンモードからウェイクアップし、安定時間 (t_{START}) だけ待ちます。
コード例については、「付録」セクション [A.7.15: 温度設定のコード例](#) を参照してください。
4. ADC_CR レジスタの ADSTART ビットをセットすることによって(または外部トリガによって) ADC 変換を開始します。
5. ADC_DR レジスタから結果のデータを読み出します。
6. 次の式を使用して、実際の温度を計算します。

$$\text{温度} ((^{\circ}\text{C})) = \frac{110^{\circ}\text{C} - 30^{\circ}\text{C}}{\text{TS\_CAL2} - \text{TS\_CAL1}} \times (\text{TS\_DATA} - \text{TS\_CAL1}) + 30^{\circ}\text{C}$$

ここで、

- TS_CAL2 は、110°C で得られた温度センサの較正值です。
- TS_CAL1 は、30°C で得られた温度センサの較正值です。
- TS_DATA は、ADC によって変換された実際の温度センサの出力値です。
TS_CAL1 および TS_CAL2 較正ポイントの詳細については、特定のデバイスのデータシートを参照してください。

コード例については、[A.7.16: 温度計算のコード例](#) を参照してください。

注: センサがパワーダウンモードからウェイクアップして、正しいレベルで V_{SENSE} を出力できるようになるまでには時間がかかります (スタートアップ時間)。ADC にも起動後のスタートアップ時間があるので、遅延を最小にするには、ADEN ビットと TSEN ビットを同時にセットしてください。

内部基準電圧を使用した実際の V_{DDA} 電圧の計算

マイクロコントローラに印加される V_{DDA} 電源電圧は、ばらつきがあり、または正確にはわかりません。埋め込みの内部電圧基準 (VREFINT) と、製造プロセス時に $V_{DDA} = 3.3\text{ V}$ で ADC によって得られた較正データを使用して、実際の V_{DDA} 電圧レベルを評価することができます。

デバイスに印加される実際の V_{DDA} 電圧は、次の式で求められます。

$$V_{DDA} = 3.3\text{ V} \times \text{VREFINT\_CAL} / \text{VREFINT\_DATA}$$

ここで、

- VREFINT_CAL は、VREFINT の較正值です。
- VREFINT_DATA は、ADC によって変換された実際の VREFINT の出力値です。

供給に相対的な ADC 測定値から絶対電圧値への変換

ADC は、アナログ電源と変換されるチャネルに印加される電圧との比に対応するデジタル値を提供するように設計されています。ほとんどのアプリケーションの使用事例では、この比を V_{DDA} に依存しない電圧に変換する必要があります。 V_{DDA} が既知であり、ADC によって変換された値が右詰めされるアプリケーションでは、次の式を使用して、この絶対値を求めることができます。

$$V_{\text{CHANNEL}x} = \frac{V_{DDA}}{\text{FULL\_SCALE}} \times \text{ADC\_DATA}_x$$

V_{DDA} 値が既知ではないアプリケーションの場合、内部電圧基準を使用する必要があり、 V_{DDA} をセクション [内部基準電圧を使用した実際の \$V_{DDA}\$ 電圧の計算](#) に記載されている式で置き換えることができます、次の式を使用できます。

$$V_{CHANNELx} = \frac{3.3\text{ V} \times VREFINT\_CAL \times ADC\_DATA_x}{VREFINT\_DATA \times FULL\_SCALE}$$

ここで、

- $VREFINT\_CAL$ は、 $VREFINT$ の較正值です。
- ADC_DATA_x は、チャンネル x で ADC によって測定された値です（右詰め）。
- $VREFINT\_DATA$ は、ADC によって変換された実際の $VREFINT$ の出力値です。
- $FULL\_SCALE$ は、ADC 出力の最大デジタル値です。たとえば、12 ビット分解能では $2^{12} - 1 = 4095$ になり、8 ビット分解能では $2^8 - 1 = 255$ になります。

注： *ADC 測定が12 ビット右詰め以外の出力形式を使用して行われる場合、計算を行う前に、すべてのパラメータを互換性のある形式に変換しておく必要があります。*

13.10 バッテリ電圧監視

アプリケーションでは ADC_CCR レジスタの $VBATEN$ ビットによって、 V_{BAT} ピンのバックアップ バッテリ電圧を測定できます。 V_{BAT} 電圧は V_{DDA} より高くなることがあるので、ADC の正しい動作を確保するために、 V_{BAT} ピンはブリッジ 2 分圧回路に内部接続されています。このブリッジは、 $VBATEN$ ビットがセットされると自動的に有効になり、 $V_{BAT}/2$ を ADC_IN18 入力チャンネルに接続します。結果として、変換されたデジタル値は V_{BAT} 電圧の半分です。望ましくないバッテリ消費を避けるには、ADC 変換を行うときだけ、ブリッジ分圧回路を有効にすることが推奨されます。

13.11 ADC 割り込み

割り込みは、次のイベントによって生成できます。

- ADC パワーアップ、ADC の準備ができたとき ($ADRDY$ フラグ)
- 変換の終了 (EOC フラグ)
- 変換シーケンスの終了 ($EOSEQ$ フラグ)
- アナログウォッチドッグ検出の発生時 (AWD フラグ)
- サンプリングフェーズの終了が発生したとき ($EOSMP$ フラグ)
- データオーバーランの発生時 (OVR フラグ)

高い柔軟性を実現するため、個別の割り込みイネーブルビットを使用できます。

表 47. ADC 割り込み

割り込みイベント	イベントフラグ	イネーブル制御ビット
ADC レディ	$ADRDY$	$ADRDYIE$
変換の終了	EOC	$EOCIE$
変換シーケンスの終了	$EOSEQ$	$EOSEQIE$
アナログウォッチドッグステータスビットのセット	AWD	$AWDIE$
サンプリングフェーズの終了	$EOSMP$	$EOSMPIE$
オーバーラン	OVR	$OVRIE$

13.12 ADC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

13.12.1 ADC 割り込みおよびステータスレジスタ (ADC_ISR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD	Res.	Res.	OVR	EOSEQ	EOC	EOSMP	ADRDY
								r_w1			r_w1	r_w1	rc_w1	r_w1	r_w1

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **AWD** : アナログウォッチドッグフラグ

このビットは、変換された電圧が、ADC_LTR および ADC_HTR レジスタでプログラミングされた値を逸脱したときに、ハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : アナログウォッチドッグイベントは発生しませんでした（またはフラグイベントはソフトウェアによってすでに確認され、クリアされました）。

1 : アナログウォッチドッグイベントが発生しました。

ビット 6:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **OVR** : ADC オーバーラン

このビットは、オーバーランが発生したときにハードウェアによってセットされ、EOC フラグがすでにセットされているときに新しい変換が完了したことを意味します。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : オーバーランは発生しませんでした（またはフラグイベントはソフトウェアによってすでに確認され、クリアされました）。

1 : オーバーランが発生しました。

ビット 3 **EOSEQ** : シーケンス終了フラグ

このビットは、CHSEL ビットによって選択されたチャネルのシーケンスの変換終了時にハードウェアによってセットされます。ソフトウェアで 1 を書き込むことによってクリアされます。

0 : 変換シーケンスは完了していません（またはフラグイベントはソフトウェアによってすでに確認され、クリアされました）。

1 : 変換シーケンスが完了しました。

ビット 2 **EOC** : 変換終了フラグ

このビットは、チャンネルの各変換の終了時に、新しいデータ結果が ADC_DR レジスタで使用可能になったときに、ハードウェアによってセットされます。ソフトウェアによって 1 を書き込むことによって、または ADC_DR レジスタを読み出すことによってクリアされます。

- 0 : チャンネル変換は完了していません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされました)。
- 1 : チャンネル変換が完了しました。

ビット 1 **EOSMP** : サンプリング終了フラグ

このビットは、変換中、サンプリングフェーズの終了時にハードウェアによってセットされます。

- 0 : サンプリングフェーズの終了時ではありません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされました)。
- 1 : サンプリングフェーズの終了に達しました。

ビット 0 **ADRDY** : ADC レディ

このビットは、ADC が有効にされた後 (ビット ADEN=1)、ADC が変換リクエストを受け入れる準備ができた状態に達したときに、ハードウェアによってセットされます。

ソフトウェアで 1 を書き込むことによってクリアされます。

- 0 : ADC はまだ変換を開始する準備ができていません (またはフラグイベントはソフトウェアによってすでに確認され、クリアされました)。
- 1 : ADC は変換を開始する準備ができました。

注 : オートオフモード (AUTOFF=1) の場合、パワーオン/オフフェーズはハードウェアによって自動的に実行され、ADRDY フラグはセットされません。

13.12.2 ADC 割り込み有効レジスタ (ADC_IER)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDIE	Res.	Res.	OVRIE	EOSEQIE	EOCIE	EOSMPIE	ADRDYIE
								rw			rw	rw	rw	rw	rw

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **AWDIE** : アナログウォッチドッグ割り込みイネーブル

このビットは、アナログウォッチドッグ割り込みを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : アナログウォッチドッグ割り込みは無効です。
- 1 : アナログウォッチドッグ割り込みは有効です。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 6:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 OVRIE : オーバーラン割り込みイネーブル

このビットは、オーバーラン割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : オーバーラン割り込みは無効です。

1 : オーバーラン割り込みは有効です。OVR ビットがセットされると、割り込みが生成されます。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 3 EOSEQIE : 変換シーケンス終了割り込み有効

このビットは、変換シーケンス終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOSEQ 割り込みは無効です。

1 : EOSEQ 割り込みは有効です。EOSEQ ビットがセットされると、割り込みが生成されます。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 2 EOCIE : 変換終了割り込み有効

このビットは、変換終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOC 割り込みは無効です。

1 : EOC 割り込みは有効です。EOC ビットがセットされると、割り込みが生成されます。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 1 EOSMPIE : サンプリング終了フラグ割り込み有効

このビットは、サンプリングフェーズ終了時の割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : EOSMP 割り込みは無効です。

1 : EOSMP 割り込みは有効です。EOSMP ビットがセットされると、割り込みが生成されます。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 0 ADRDYIE : ADC レディ割り込みイネーブル

このビットは、ADC レディ割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : ADRDY 割り込みは無効です。

1 : ADRDY 割り込みは有効です。ADRDY ビットがセットされると、割り込みが生成されます。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

13.12.3 ADC 制御レジスタ (ADC_CR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD CAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADSTP	Res.	ADSTAR T	ADDIS	ADEN
											rs		rs	rs	rs

ビット 31 **ADCAL** : ADC 較正

このビットは、ADC の較正を開始するためにソフトウェアによってセットされます。

較正の完了後、ハードウェアによってクリアされます。

0 : 較正が完了しました。

1 : ADC を較正するには、1 を書き込みます。1 として読み出されたときには、較正が実行中であることを意味します。

注 : ソフトウェアは、ADC が無効のときだけ (ADCAL=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0)、ADCAL をセットできます。

ビット 30:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **ADSTP** : ADC 変換停止コマンド

このビットは、実行中の変換を停止および破棄するためにソフトウェアによってセットされます (ADSTP コマンド)。

変換が効果的に破棄され、ADC が新しい変換開始コマンドを受け入れる準備ができたときに、ハードウェアによってクリアされます。

0 : ADC 変換停止コマンドは実行中ではありません。

1 : ADC を停止するには、1 を書き込みます。1 として読み出されたときには、ADSTP コマンドが実行中であることを意味します。

注 : ソフトウェアは、ADSTART=1 かつ ADDIS=0 (ADC が有効であり、変換中である可能性があり、ADC を無効にする保留中のリクエストがない) のときだけ、ADSTP をセットできます。

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 ADSTART : ADC 変換開始コマンド

このビットは、ADC 変換を開始するためにソフトウェアによってセットされます。EXTEN [1:0] 設定ビットに応じて、変換はただちに開始されるか（ソフトウェアトリガ設定）またはハードウェアトリガイベントが発生したときに（ハードウェアトリガ設定）開始されます。

次のときに、ハードウェアによってクリアされます。

- シングル変換モードでは、ソフトウェアトリガが選択されたとき (EXTEN=0x0) : 変換シーケンス終了 (EOSEQ) フラグのアサート時。
- すべての場合 : ADSTP コマンドの実行後、ADSTP ビットがハードウェアによってクリアされると同時に。
 - 0 : ADC 変換は実行中ではありません。
 - 1 : ADC を開始するには、1 を書き込みます。1 として読み出されたときには、ADC が動作中であり、変換中である可能性があることを意味します。

注： ソフトウェアは、ADEN=1 かつ ADDIS=0 (ADC が有効であり、ADC を無効にする保留中のリクエストがない) のときだけ、ADSTART をセットできます。

ビット 1 ADDIS : ADC 無効化コマンド

このビットは、ADC を無効にして (ADDIS コマンド)、パワーダウン状態 (OFF 状態) にするためにソフトウェアによってセットされます。

ADC が効果的に無効化されると、ハードウェアによってクリアされます (ADEN もこの時点でハードウェアによってクリアされます)。

- 0 : ADDIS コマンドは実行中ではありません。
- 1 : ADC を無効にするには、1 を書き込みます。1 として読み出されたときには、ADDIS コマンドが実行中であることを意味します。

注： ソフトウェアは、ADEN=1 かつ ADSTART=0 (変換が実行中でない) のときのみ、ADDIS をセットできます。

ビット 0 ADEN : ADC 有効化コマンド

このビットは、ADC を有効にするために、ソフトウェアによってセットされます。ADRDY フラグがセットされると、ADC は動作する準備ができています。

ADDIS コマンドの実行後、ADC が無効になるとハードウェアによってクリアされます。

- 0 : ADC は無効です (OFF 状態)。
- 1 : ADC を有効にするには、1 を書き込みます。

注： ソフトウェアは、ADC_CR レジスタのすべてのビットが 0 のとき (ADCAL=0、ADSTP=0、ADSTART=0、ADDIS=0、および ADEN=0) だけ、ADEN をセットできます。

13.12.4 ADC 設定レジスタ 1 (ADC_CFGR1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWDCH[4:0]					Res.	Res.	AWDEN	AWDSGL	Res.	Res.	Res.	Res.	Res.	DISCEN
	rW	rW	rW	rW	rW			rW	rW						rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUTOFF	WAIT	CONT	OVRMOD	EXTEN[1:0]		Res.	EXTSEL[2:0]		ALIGN	RES[1:0]		SCANDIR	DMACFG	DMAEN	
rW	rW	rW	rW	rW			rW		rW	rW		rW	rW	rW	

ビット 31 予約済みであり、リセット値に保持する必要があります。

ビット 30:26 **AWDCH[4:0]** : アナログウォッチドッグチャンネル選択

これらのビットは、ソフトウェアによってセット／クリアされます。アナログウォッチドッグによって保護される入力チャンネルを選択します。

00000 : ADC アナログ入力チャンネル 0 が AWD によって監視されます。

00001 : ADC アナログ入力チャンネル 1 が AWD によって監視されます。

.....

10010 : ADC アナログ入力チャンネル 18 が AWD によって監視されます。

その他の値 : 予約済み。使用できません。

注 : **AWDCH [4:0] ビットによって選択されたチャンネルは、CHSELR レジスタにもセットされる必要があります。**

注 : **ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、これらのビットを書き込むことができます。**

ビット 25:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **AWDEN** : アナログウォッチドッグ有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : アナログウォッチドッグは無効です。

1 : アナログウォッチドッグは有効です。

注 : **ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。**

ビット 22 **AWDSGL** : ウォッチドッグを単一チャンネルまたはすべてのチャンネルで有効にします。

このビットは、AWDCH[4:0] ビットによって指定されたチャンネルまたはすべてのチャンネルに対するアナログウォッチドッグを有効にするために、ソフトウェアによってセット／クリアされます。

0 : すべてのチャンネルでアナログウォッチドッグは有効です。

1 : 単一チャンネルでアナログウォッチドッグは有効です。

注 : **ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。**

ビット 21:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **DISCEN** : 不連続モード

このビットは、不連続モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 不連続モードは無効です。

1 : 不連続モードは有効です。

注 : **不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方のビットをセットすることは禁じられています。**

注 : **ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。**

ビット 15 AUTOFF : オートオフモード

このビットは、オートオフモードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

- 0 : オートオフモードは無効です。
- 1 : オートオフモードは有効です。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 14 WAIT : ウェイト変換モード

このビットは、ウェイト変換モードを有効／無効にするために、ソフトウェアによってセット／クリアされます。

- 0 : ウェイト変換モードはオフです。
- 1 : ウェイト変換モードはオンです。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 13 CONT : シングル／連続変換モード

このビットは、ソフトウェアによってセット／クリアされます。このビットがセットされた場合、それがクリアされるまで連続的に変換が行われます。

- 0 : シングル変換モード
- 1 : 連続変換モード

注 : 不連続モードと連続モードの両方を有効にすることはできません。DISCEN=1 と CONT=1 の両方のビットをセットすることは禁じられています。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 12 OVRMOD : オーバーラン管理モード

このビットは、ソフトウェアによってセット／クリアされ、データオーバーランを管理する方法を設定します。

- 0 : オーバーランが検出されたとき、ADC_DR レジスタの古いデータが保存されます。
- 1 : オーバーランが検出されたとき、ADC_DR レジスタは最後の変換結果で上書きされます。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 11:10 EXTEN[1:0] : 外部トリガ有効および極性選択

これらのビットは、外部トリガ極性を選択し、トリガを有効にするために、ソフトウェアによってセット／クリアされます。

- 00 : ハードウェアトリガ検出は無効です (変換はソフトウェアによって開始できます)。
- 01 : 立ち上がりエッジでハードウェアトリガを検出します。
- 10 : 立ち下がりエッジでハードウェアトリガを検出します。
- 11 : 立ち上がりエッジと立ち下がりエッジの両方でハードウェアトリガを検出します。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

ビット 9 予約済みであり、リセット値に保持する必要があります。

ビット 8:6 EXTSEL[2:0] : 外部トリガ選択

これらのビットは、変換の開始をトリガするために使用される外部イベントを選択します（詳細については、[表 43 : 外部トリガ](#)を参照）。

000 : TRG0
001 : TRG1
010 : TRG2
011 : TRG3
100 : TRG4
101 : TRG5
110 : TRG6
111 : TRG7

注： ソフトウェアは、ADSTART=0（変換が実行中でない）ときのみ、これらのビットを書き込むことができます。

ビット 5 ALIGN : データの配置

このビットは、右詰めまたは左詰めを選択するために、ソフトウェアによってセット/クリアされます。
[図 37 : データの配置と分解能 \(246 ページ\)](#)を参照してください。

0 : 右詰め
1 : 左詰め

注： ソフトウェアは、ADSTART=0（変換が実行中でない）ときのみ、このビットに書き込むことができます。

ビット 4:3 RES[1:0] : データ分解能

これらのビットは、変換の分解能を選択するために、ソフトウェアによって書き込まれます。

00 : 12 ビット
01 : 10 ビット
10 : 8 ビット
11 : 6 ビット

注： ソフトウェアは、ADEN=0 のときだけ、これらのビットを書き込むことができます。

ビット 2 SCANDIR : スキャンシーケンス方向

このビットは、シーケンス内のチャネルをスキャンする方向を選択するために、ソフトウェアによってセット/クリアされます。

0 : 前方スキャン (CHSEL0 から CHSEL18)

1 : 後方スキャン (CHSEL18 から CHSEL0)

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 1 DMACFG : ダイレクトメモリアクセス設定

このビットは、2 つの DMA 動作モードを選択するためにソフトウェアによってセット/クリアされ、DMAEN=1 のときのみ有効です。

0 : DMA ワンショットモードが選択されています。

1 : DMA サーキュラモードが選択されています。

詳細については、[セクション 13.6.5 : DMA を使用した変換データの管理 \(247 ページ\)](#) を参照してください。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 0 DMAEN : ダイレクトメモリアクセス有効

このビットは、DMA リクエストの生成を有効にするために、ソフトウェアによってセット/クリアされます。これにより、DMA コントローラを使用して変換データを自動的に管理できます。詳細については、[セクション 13.6.5 : DMA を使用した変換データの管理 \(247 ページ\)](#) を参照してください。

0 : DMA は無効です。

1 : DMA は有効です。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

13.12.5 ADC 設定レジスタ 2 (ADC_CFGR2)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKMODE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:30 **CKMODE[1:0]** : ADC クロックモード

これらのビットは、アナログ ADC クロックの動作方法を定義するために、ソフトウェアによってセット/クリアされます。

00 : ADCCLK (非同期クロックモード)。製品レベルで生成されます (RCC セクションを参照)。

01 : PCLK/2 (同期クロックモード)

10 : PCLK/4 (同期クロックモード)

11 : 予約済み

すべての同期クロックモードにおいて、タイマトリガから変換開始までの遅延にジッタはありません。

注 : ソフトウェアは、ADC が無効のときだけ (ADCAL=0、ADSTART=0、ADSTP=0、ADDIS=0、および ADEN=0)、これらのビットを書き込みことができます。

ビット 29:0 予約済みであり、リセット値に保持する必要があります。

13.12.6 ADC サンプルング時間レジスタ (ADC_SMPR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMP[2:0]		
													rw		

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **SMP[2:0]** : サンプルング時間選択

これらのビットは、すべてのチャンネルに適用されるサンプルング時間を選択するために、ソフトウェアによって書き込まれます。

- 000 : 1.5 ADC クロックサイクル
- 001 : 7.5 ADC クロックサイクル
- 010 : 13.5 ADC クロックサイクル
- 011 : 28.5 ADC クロックサイクル
- 100 : 41.5 ADC クロックサイクル
- 101 : 55.5 ADC クロックサイクル
- 110 : 71.5 ADC クロックサイクル
- 111 : 239.5 ADC クロックサイクル

注： ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

13.12.7 ADC ウォッチドッグ閾値レジスタ (ADC_TR)

アドレスオフセット : 0x20

リセット値 : 0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **HT[11:0]** : アナログウォッチドッグ高閾値

これらのビットは、アナログウォッチドッグの高閾値を定義するために、ソフトウェアによって書き込まれます。セクション 13.8 : アナログウィンドウウォッチドッグ (AWDEN、AWDSGL、AWDCH、AWD_HTR/LTR、AWD) (251 ページ) を参照してください。

注： ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **LT[11:0]** : アナログウォッチドッグ低閾値

これらのビットは、アナログウォッチドッグの低閾値を定義するために、ソフトウェアによって書き込まれます。セクション 13.8 : アナログウィンドウウォッチドッグ (AWDEN、AWDSGL、AWDCH、AWD_HTR/LTR、AWD) (251 ページ) を参照してください。

注： ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、これらのビットを書き込むことができます。

13.12.8 ADC チャンネル選択レジスタ (ADC_CHSELR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL 18	CHSEL 17	CHSEL 16
													rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHSEL 15	CHSEL 14	CHSEL 13	CHSEL 12	CHSEL 11	CHSEL 10	CHSEL 9	CHSEL 8	CHSEL 7	CHSEL 6	CHSEL 5	CHSEL 4	CHSEL 3	CHSEL 2	CHSEL 1	CHSEL 0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18:0 **CHSELx** : チャンネル x 選択

これらのビットはソフトウェアによって書き込まれ、変換されるチャンネルのシーケンスの一部になるチャンネルを定義します。

0 : 入力チャンネル x は変換対象として選択されません。

1 : 入力チャンネル x は変換対象として選択されます。

注 : ソフトウェアは、**ADSTART=0 (変換が実行中でない)** ときのみ、これらのビットを書き込むことができます。

13.12.9 ADC データレジスタ (ADC_DR)

アドレスオフセット : 0x40

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **DATA[15:0]** : 変換データ

これらのビットは読み出し専用です。これらは、最後の変換チャンネルの変換結果を含んでいます。図 37 : データの配置と分解能 (246 ページ) に示すように、データは左詰めまたは右詰めされています。

較正の完了直後、DATA[6:0] は較正係数を含んでいます。

13.12.10 ADC 共通設定レジスタ (ADC_CCR)

アドレスオフセット : 0x308

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBAT EN	TS EN	VREF EN	Res.	Res.	Res.	Res.	Res.	Res.
							rw	rw	rw						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24 **VBATEN** : V_{BAT} イネーブル

このビットは、V_{BAT} チャンネルを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : V_{BAT} チャンネルは無効です。

1 : V_{BAT} チャンネルは有効です。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 23 **TSEN** : 温度センサ有効

このビットは、温度センサを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 温度センサは無効です。

1 : 温度センサは有効です。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 22 **VREFEN** : V_{REFINT} 有効

このビットは、V_{REFINT} を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : V_{REFINT} は無効です。

1 : V_{REFINT} は有効です。

注 : ソフトウェアは、ADSTART=0 (変換が実行中でない) ときのみ、このビットに書き込むことができます。

ビット 21:0 予約済みであり、リセット値に保持する必要があります。

13.12.11 ADC レジスタマップ

次の表は ADC レジスタの一覧です。

表 48. ADC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	ADC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD	0	Res.	0	EOSEQ	EOC	EOSMP	ADRDY
	リセット値																										0		0	0	0	0	0	
0x04	ADC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDIE	0	Res.	0	EOSEQIE	EOCIE	EOSMPIE	ADRDYIE
	リセット値																										0		0	0	0	0	0	
0x08	ADC_CR	ADCAL	Res.		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADSTP	Res.	ADSTART	ADDIS	ADEN	
	リセット値	0																										0		0	0	0	0	
0x0C	ADC_CFGR1	Res.	AWDCH[4:0]					Res.	Res.		AWDEN	AWDSGL	Res.	Res.	Res.	Res.	Res.	DISCEN	AUTOFF	WAIT	CONT	OVRMOD	EXTEN[1:0]			EXTSEL [2:0]		ALIGN	RES [1:0]	SCANDIR	DMACFG	DMAEN		
	リセット値		0	0	0	0	0			0	0							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	ADC_CFGR2	CKMODE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値	0	0																															
0x14	ADC_SMPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMP [2:0]			
	リセット値																														0	0	0	
0x18	予約済み																																	
0x1C	予約済み																																	
0x20	ADC_TR	Res.	Res.	Res.	Res.	HT[11:0]											Res.	Res.	Res.	Res.	LT[11:0]													
	リセット値					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0	
0x24	予約済み																																	
0x28	ADC_CHSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL18	CHSEL17	CHSEL16	CHSEL15	CHSEL14	CHSEL13	CHSEL12	CHSEL11	CHSEL10	CHSEL9	CHSEL8	CHSEL7	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	
	リセット値														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x2C 0x30 0x34 0x38 0x3C	予約済み																																	
0x40	ADC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x44 ... 0x304	予約済み																																	
0x308	ADC_CCR	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	TSEN	VREFEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値							0	0	0																								

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

14 D/A コンバータ (DAC)

このセクションは、STM32F05x、STM32F07x、および STM32F09x デバイスにのみ適用されます。2 番目の DAC チャンネル (DAC_OUT2) とその他の一部の機能は、STM32F07x および STM32F09x デバイスでのみ使用できます。

14.1 概要

DAC モジュールは、12 ビットの電圧出力デジタルアナログコンバータです。DAC は、8 または 12 ビットモードで設定でき、DMA コントローラと組み合わせて使用することもできます。12 ビットモードでは、データを左詰め右詰めどちらにも配置できます。入力基準電圧、 V_{DDA} (ADC と共用) が使用できます。より高い電流駆動力を得るために、任意で出力バッファを有効にすることができます。

14.2 DAC の主な機能

デバイスは、12 ビット DAC チャンネルの DAC_OUT1 を 1 個実装しています。2 番目のチャンネル DAC_OUT2 は STM32F07x および STM32F09x デバイスでのみ使用できます。

DAC の主な機能を以下に示します。

- 12 ビットモードでのデータの左詰めまたは右詰め
- 同期更新機能
- ノイズ波生成 (STM32F07x および STM32F09x デバイス)
- 三角波生成 (STM32F07x および STM32F09x デバイス)
- 独立または同時変換 (デュアルモードのみ)
- DMA 機能
- DMA アンダーランエラー検出
- 変換外部トリガ
- プログラム可能な内部バッファ
- 入力基準電圧、 V_{DDA}

 44 は DAC チャンネルのブロック図を、 49 はピンの概要を示します。

図 44. DAC ブロック図

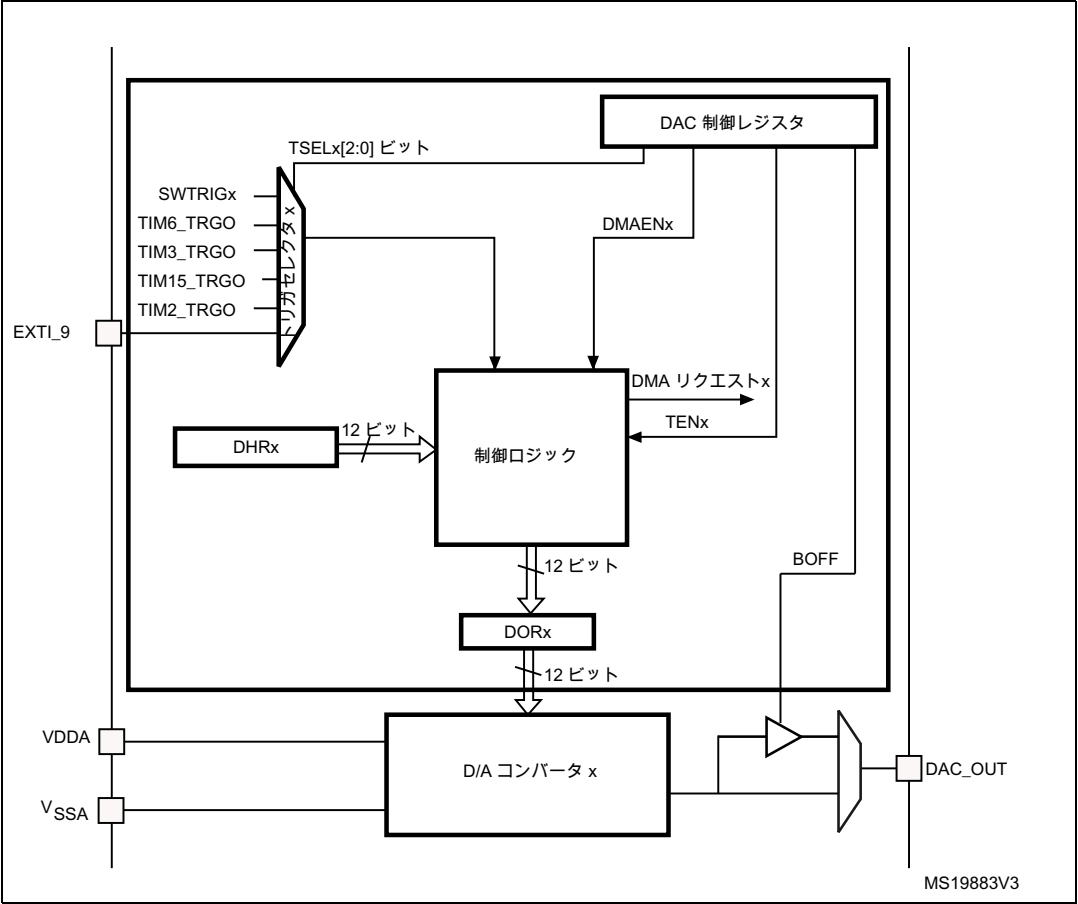


表 49. DAC ピン

名前	信号タイプ	説明
V _{DDA}	入力、アナログ電源供給	アナログ電源供給
V _{SSA}	入力、アナログ供給グラウンド	アナログ電源供給のグラウンド
DAC_OUT	アナログ出力信号	DAC チャネル x アナログ出力

注： DAC チャネル x が有効になると、対応する GPIO ピン (PA4 または PA5) が自動的にアナログコンバータ出力 (DAC_OUTx) に接続されます。アナログ出力をデジタルパートに入力する事による不要な消費電力増大を防ぐために、PA4 または PA5 ピンはまずアナログ (AIN) として設定してください。

14.3 シングルモードの機能説明

14.3.1 DAC チャンネルイネーブル

各 DAC チャンネルは、DAC_CR レジスタの対応する ENx ビットをセットすることによって起動できます。各 DAC チャンネルは、スタートアップ時間 t_{WAKEUP} 後に有効になります。

注： ENx ビットは、アナログ DAC チャンネル x のマクロセルのみを有効にします。DAC チャンネル x デジタルインタフェースは、ENx ビットがリセットされた場合でも有効になります。

14.3.2 DAC 出力バッファイネーブル

DAC は、出力インピーダンスを減らし、外部動作アンプがなくても外部負荷を直接駆動できる 1 つの出力バッファを内蔵しています。

DAC チャンネル出力バッファは、DAC_CR レジスタの BOFF1 ビットを通じて有効／無効にできます。

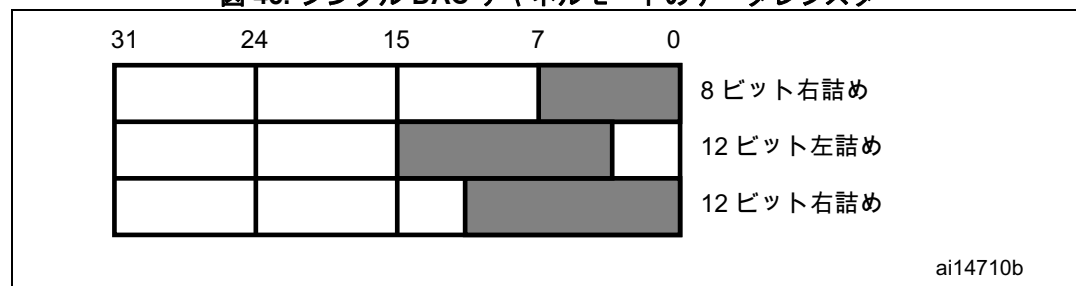
14.3.3 DAC データフォーマット

以下に示すように、選択された設定モードに応じて、指定されたレジスタにデータを書き込む必要があります。

- この場合、次の 3 つの設定が可能です。
 - － 8 ビット右詰め：データは、DAC_DHR8Rx [7:0] ビット (DHRx [11:4] ビットに格納) にソフトウェアによってロードされること。
 - － 12 ビット左詰め：データは、DAC_DHR12Lx [15:4] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。
 - － 12 ビット右詰め：データは、DAC_DHR12Rx [11:0] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。

ユーザによって書き込まれたデータは、ロードされた DAC_DHRyyyx レジスタに応じて、シフトされてから、DHRx (メモリマップされない内部レジスタであるデータ保持レジスタ x) に格納されます。その後、DHRx レジスタは自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、DORx レジスタにロードされます。

図 45. シングル DAC チャンネルモードのデータレジスタ



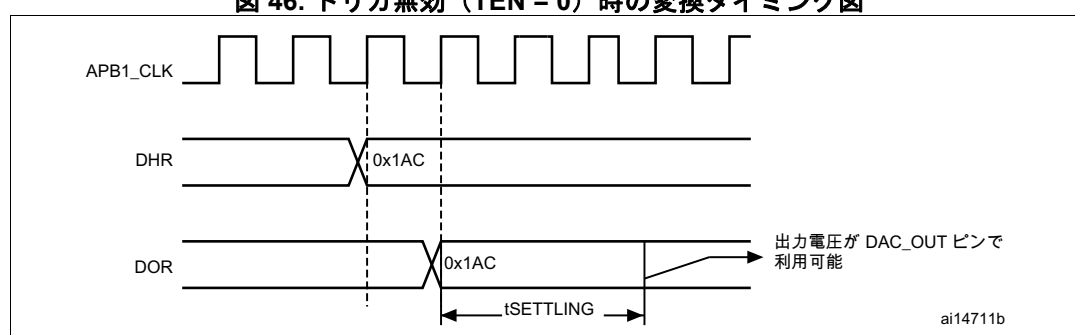
14.3.4 DAC チャンネル変換

DAC_DORx に直接書き込むことはできませんので、DAC_DHRx レジスタをロードする (DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx への書き込み) ことによって、DAC チャンネル x へのデータ転送を行う必要があります。

DAC_DHRx レジスタに格納されたデータは、ハードウェアトリガが選択されていない (DAC_CR レジスタの TENx ビットがリセットされている) 場合に、1 APB クロックサイクル後に DAC_DORx レジスタに自動的に転送されます。ただし、ハードウェアトリガが選択されている (DAC_CR レジスタの TENx ビットがセットされている) ときには、トリガが発生すると、転送は 3 APB クロックサイクル後に行われます。

DAC_DORx に DAC_DHRx の内容がロードされると、電源電圧とアナログ出力負荷に応じて決定される t_{SETTLING} 時間後にアナログ出力電圧が使用可能になります。

図 46. トリガ無効 (TEN = 0) 時の変換タイミング図



1 つの LFSR 生成による独立トリガ

DAC をこの変換モード ([セクション 14.5: ノイズ生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照) に設定するには、次の手順が必要です。

1. DAC チャンネルトリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットをセットすることによって、トリガソースを設定します。
3. DAC チャンネル WAVEx[1:0] ビットを "01" に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DAC_DHRx レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、DAC チャンネルデータをロードします。

DAC チャンネル x トリガが発生すると、同じマスクを持つ LFSRx カウンタが DHRx レジスタに加算され、合計が DAC_DORx に転送されます (3 APB クロックサイクル後)。その後、LFSR x カウンタが更新されます。

1 つの三角波生成による独立トリガ

DAC をこの変換モード ([セクション 14.6: 三角波生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照) に設定するには、次の手順が必要です。

1. DAC チャンネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットをセットすることによって、トリガソースを設定します。
3. DAC チャンネル x の WAVEx[1:0] ビットを "1x" に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. DAC チャンネル x データを、目的の DAC_DHRx レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、ロードします。

DAC チャンネル x トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャンネル x の三角波カウンタが更新されます。

14.3.5 DAC 出力電圧

デジタル入力は、0 から V_{DDA} までのリニア変換で出力電圧に変換されます。

各 DAC チャンネルピンのアナログ出力電圧は、次の式によって求められます。

$$\text{DACoutput} = V_{\text{DDA}} \times \frac{\text{DOR}}{4095}$$

14.3.6 DAC トリガ選択

TENx 制御ビットがセットされている場合、外部イベント (タイマカウンタ、外部割り込みラインなど) によって変換をトリガできます。表 50 に示すように、8 つの考えられるイベントのうち、どのイベントが変換をトリガするかは TSELx[2:0] 制御ビットによって決まります。

表 50. 外部トリガ

転送元	タイプ	TSEL[2:0]
TIM6_TRGO イベント	オンチップタイマからの内部信号	000
TIM3_TRGO イベント		001
TIM7_TRGO イベント		010
TIM15_TRGO イベント		011
TIM2_TRGO イベント		100
予約済み		101
外部割り込みライン 9 (EXTI ライン 9)	外部ピン	110
SWTRIG	ソフトウェア制御ビット	111

DAC インタフェースが選択されたタイマ TRGO 出力または選択された外部割り込みライン 9 の立ち上がりエッジを検出するたびに、DAC_DHRx レジスタに最後に格納されたデータが DAC_DORx レジスタに転送されます。DAC_DORx レジスタは、トリガが発生してから 3 APB サイクル後に更新されます。

ソフトウェアトリガが選択されている場合、変換は、SWTRIG ビットがセットされると開始されます。SWTRIG ビットは、DAC_DHRx レジスタの内容が DAC_DORx レジスタにロードされると、ハードウェアによってリセットされます。

注： **ENx ビットがセットされているときには、TSELx[2:0] ビットを変更することはできません。ソフトウェアトリガが選択されているときには、DAC_DHRx レジスタから DAC_DORx レジスタへの転送は、わずか 1 APB クロックサイクルで行われます。**



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

14.4 デュアルモードの機能詳細 (STM32F07x および STM32F09x デバイス)

14.4.1 DAC チャンネルイネーブル

各 DAC チャンネルは、DAC_CR レジスタの対応する ENx ビットをセットすることによって起動できます。DAC チャンネルは、スタートアップ時間 t_{WAKEUP} 後に有効になります。

注： ENx ビットは、アナログ DAC チャンネル x のマクロセルのみを有効にします。DAC チャンネル x デジタルインタフェースは、ENx ビットがリセットされた場合でも有効になります。

14.4.2 DAC 出力バッファイネーブル

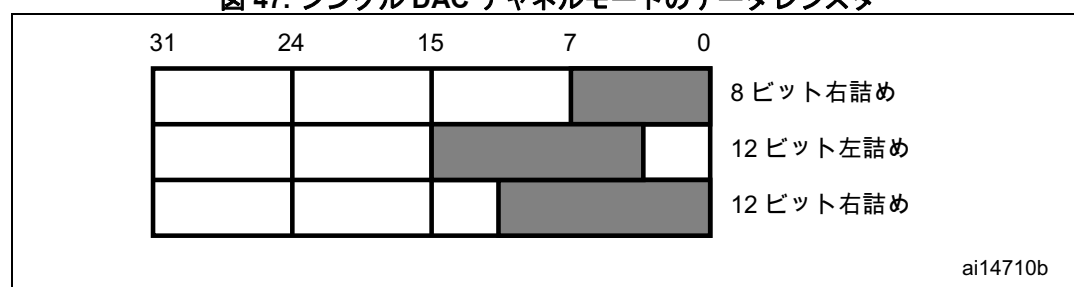
14.4.3 DAC データフォーマット

以下に示すように、選択された設定モードに応じて、指定されたレジスタにデータを書き込む必要があります。

- シングル DAC チャンネルモード
 - この場合、次の3つの設定が可能です。
 - 8 ビット右詰め：データは、DAC_DHR8Rx [7:0] ビット (DHRx [11:4] ビットに格納) にソフトウェアによってロードされること。
 - 12 ビット左詰め：データは、DAC_DHR12Lx [15:4] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。
 - 12 ビット右詰め：データは、DAC_DHR12Rx [11:0] ビット (DHRx [11:0] ビットに格納) にソフトウェアによってロードされること。

ユーザによって書き込まれたデータは、ロードされた DAC_DHRyyyx レジスタに応じて、シフトされてから、DHRx (メモリマップされない内部レジスタであるデータ保持レジスタ x) に格納されます。その後、DHRx レジスタは自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、DORx レジスタにロードされます。

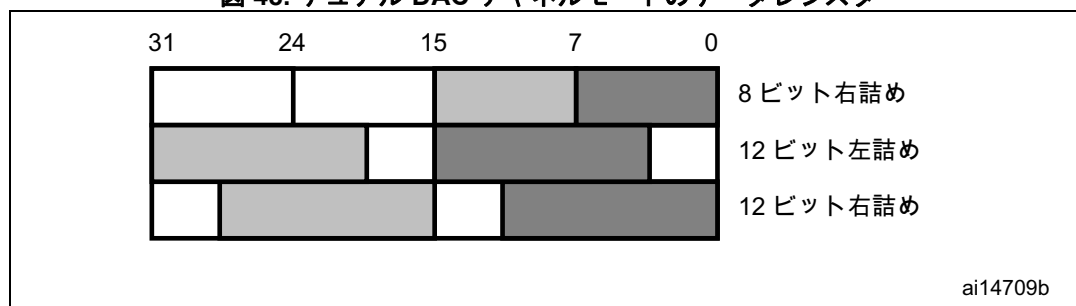
図 47. シングル DAC チャンネルモードのデータレジスタ



- デュアル DAC チャンネルモードでは、3 つの設定が可能です。
 - 8 ビット右詰め: DAC チャンネル 1 のデータは、DAC_DHR8RD [7:0] ビット (DHR1[11:4] ビットに格納) に、DAC チャンネル 2 のデータは、DAC_DHR8RD [15:8] ビット (DHR2[11:4] ビットに格納) にロードされます。
 - 12 ビット左詰め: DAC チャンネル 1 のデータは、DAC_DHR12LD [15:4] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC_DHR12LD [31:20] ビット (DHR2[11:0] ビットに格納) にロードされます。
 - 12 ビット右詰め: DAC チャンネル 1 のデータは、DAC_DHR12RD [11:0] ビット (DHR1[11:0] ビットに格納) に、DAC チャンネル 2 のデータは、DAC_DHR12LD [27:16] ビット (DHR2[11:0] ビットに格納) にロードされます。

ロードされた DAC_DHRyyyD レジスタに応じて、ユーザによって書き込まれたデータは、シフトしてから、DHR1 および DHR2 (データ保持レジスタ、メモリマップされない内部レジスタ) に格納されます。その後、DHR1 および DHR2 レジスタは、自動的に、ソフトウェアトリガによって、または外部イベントトリガによって、それぞれ DOR1 および DOR2 レジスタにロードされます。

図 48. デュアル DAC チャンネルモードのデータレジスタ



14.4.4 デュアルモードでの DAC チャンネル変換

デュアルモードでの DAC チャンネル変換は、シングルモードの場合と同様に行われます ([セクション 14.3.4](#) を参照)。ただし、データは DAC_DHR8Rx、DAC_DHR12Lx、DAC_DHR12Rx、DAC_DHR8RD、DAC_DHR12LD、または DAC_DHR12RD に書き込んでロードする必要があります。

14.4.5 デュアル変換モードの説明

同時に 2 つの DAC チャンネルを必要とするアプリケーションで、バスのバンド幅を有効に使用するために、DHR8RD、DHR12RD、および DHR12LD の 3 つのデュアルレジスタが搭載されています。両方の DAC チャンネルを同時に駆動するには、一意なレジスタアクセスが必要です。

2 つの DAC チャンネルとこれらのデュアルレジスタを使用することで、11 の変換モードが使用可能です。すべての変換モードは、必要な場合には、別の DHRx レジスタを使っても利用できます。

すべてのモードについて、以下の節で説明します。

DAC 変換を駆動する APB バス (APB または APB1) の詳細については、[セクション 14.3.4 : DAC チャンネル変換](#)を参照してください。

波形生成なしの独立トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

DAC チャンネル 1 トリガが発生すると、DHR1 レジスタが DAC_DOR1 に転送されます (3 APB クロックサイクル後)。

DAC チャンネル 2 トリガが発生すると、DHR2 レジスタが DAC_DOR2 に転送されます (3 APB クロックサイクル後)。

コード例については、「付録」セクション [A.8.1: 波形生成なしの独立トリガのコード例](#) を参照してください。

1 つの LFSR 生成による独立トリガ

DAC をこの変換モードに設定するには ([セクション 14.5: ノイズ生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、デュアル DAC チャンネルデータをロードします。

コード例については、「付録」セクション [A.8.2: 1 つの LFSR 生成による独立トリガのコード例](#) を参照してください。

DAC チャンネル 1 トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャンネル 2 トリガが発生すると、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

異なる LFSR 生成による独立トリガ

DAC をこの変換モードに設定するには ([セクション 14.5: ノイズ生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMP1[3:0] ビットと MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

コード例については、「付録」セクション [A.8.3 : 異なる LFSR 生成による独立トリガのコード例](#) を参照してください。

DAC チャネル 1 トリガが発生すると、MAMP1[3:0] によって設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

DAC チャネル 2 トリガが発生すると、MAMP2[3:0] によって設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

1 つの三角波生成による独立トリガ

DAC をこの変換モードに設定するには (セクション [14.6 : 三角波生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照)、次の手順が必要です。

1. DAC チャネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. DAC チャネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. DAC チャネル x データを、目的の DAC_DHRx レジスタ

コード例については、「付録」セクション [A.8.4 : 1 つの三角波生成による独立トリガのコード例](#) を参照してください。

DAC 変換を駆動する APB バス (APB または APB1) の詳細については、セクション [14.3.4 : DAC チャネル変換](#) を参照してください。

DAC チャネル x トリガが発生すると、同じ三角波振幅を持つ DAC チャネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャネル x の三角波カウンタが更新されます。

異なる三角波生成による独立トリガ

DAC をこの変換モードに設定するには (セクション [14.6 : 三角波生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照)、次の手順が必要です。

1. DAC チャネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットに異なる値を設定することによって、異なるトリガソースを設定します。
3. DAC チャネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで異なる最大振幅値を設定します。
4. DAC チャネル x データを、目的の DAC_DHRx レジスタ

コード例については、「付録」セクション [A.8.5 : 異なる三角波生成による独立トリガのコード例](#) を参照してください。

DAC チャネル x トリガが発生すると、MAMPx[3:0] によって設定された三角波振幅を持つ DAC チャネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャネル x の三角波カウンタが更新されます。

同時ソフトウェア開始

DAC をこの変換モードに設定するには、次の手順が必要です。

1. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) に、デュアル DAC チャネルデータをロードします。

この設定では、1 APB クロックサイクルです。



コード例については、「付録」セクション [A.8.6 : 同時ソフトウェア開始のコード例](#) を参照してください。

波形生成なしの同時トリガ

DAC をこの変換モードに設定するには、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) に、デュアル DAC チャンネルデータをロードします。

トリガが発生すると、DHR1 および DHR2 レジスタが DAC_DOR1 と DAC_DOR2 にそれぞれ転送されます (3 APB クロックサイクル後)。

コード例については、「付録」セクション [A.8.7 : 波形生成なしの同時トリガのコード例](#) を参照してください。

1 つの LFSR 生成による同時トリガ

DAC をこの変換モードに設定するには ([セクション 14.5 : ノイズ生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 2 つの DAC チャンネル WAVEx[1:0] ビットを“01”に設定し、MAMPx[3:0] ビットで同じ LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DHR12RD、DHR12LD、または DHR8RD) に、デュアル DAC チャンネルデータをロードします。

コード例については、「付録」セクション [A.8.8 : 1 つの LFSR 生成による同時トリガのコード例](#) を参照してください。

トリガが発生すると、同じマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。同時に、同じマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

異なる LFSR 生成による同時トリガ

DAC をこの変換モードに設定するには ([セクション 14.5 : ノイズ生成 \(STM32F07x および STM32F09x デバイス\)](#) を参照)、次の手順が必要です。

1. TEN1 と TEN2 の 2 つの DAC チャンネルトリガイネーブルビットをセットします。
2. TSEL1[2:0] および TSEL2[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. 2 つの DAC チャンネルの WAVEx[1:0] ビットを“01”に設定し、MAMP1[3:0] と MAMP2[3:0] ビットで異なる LFSR マスク値を設定します。
4. 目的の DHR レジスタ (DAC_DHR12RD、DAC_DHR12LD、または DAC_DHR8RD) にデュアル DAC チャンネルデータをロードします。

コード例については、「付録」セクション [A.8.9 : 異なる LFSR 生成による同時トリガのコード例](#) を参照してください。

トリガが発生すると、MAMP1[3:0] で設定されたマスクを持つ LFSR1 カウンタが DHR1 レジスタに加算され、合計が DAC_DOR1 に転送されます (3 APB クロックサイクル後)。その後、LFSR1 カウンタが更新されます。

同時に、MAMP2[3:0] で設定されたマスクを持つ LFSR2 カウンタが DHR2 レジスタに加算され、合計が DAC_DOR2 に転送されます (3 APB クロックサイクル後)。その後、LFSR2 カウンタが更新されます。

1 つの三角波生成による同時トリガ

DAC をこの変換モードに設定するには ([セクション 14.6 : 三角波生成 \(STM32F07x および STM32F09x デバイス\)](#)) を参照)、次の手順が必要です。

1. DAC チャンネル x トリガイネーブルビット TEN1x をセットします。
2. TSELx[2:0] ビットに同じ値をセットすることによって、両方の DAC チャンネルに同じトリガソースを設定します。
3. DAC チャンネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで同じ最大振幅値を設定します。
4. 目的の DAC_DHRx レジスタに、DAC チャンネル x データをロードします。

コード例については、「付録」セクション [A.8.10 : 1 つの三角波生成による同時トリガのコード例](#) を参照してください。

トリガが発生すると、同じ三角波振幅を持つ DAC チャンネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャンネル x の三角波カウンタが更新されます。

異なる三角波生成による同時トリガ

DAC をこの変換モードに設定するには ([セクション 14.6 : 三角波生成 \(STM32F07x および STM32F09x デバイス\)](#)) を参照)、次の手順が必要です。

1. DAC チャンネル x トリガイネーブルビット TENx をセットします。
2. TSELx[2:0] ビットに同じ値をセットすることによって、DAC チャンネル x に同じトリガソースを設定します。
3. DAC チャンネル x の WAVEx[1:0] ビットを“1x”に設定し、MAMPx[3:0] ビットで異なる最大振幅値を設定します。
4. 目的の DAC_DHRx レジスタに、DAC チャンネル x データをロードします。

コード例については、「付録」セクション [A.8.11 : 異なる三角波生成による同時トリガのコード例](#) を参照してください。

トリガが発生すると、MAMPx[3:0] によって設定された三角波振幅を持つ DAC チャンネル x の三角波カウンタが DHRx レジスタに加算され、合計が DAC_DORx に転送されます (3 APB クロックサイクル後)。その後、DAC チャンネル x の三角波カウンタが更新されます。

14.4.6 DAC 出力電圧

デジタル入力は、0 から V_{DDA} までのリニア変換で出力電圧に変換されます。

各 DAC チャンネルピンのアナログ出力電圧は、次の式によって求められます。

$$DAC_{output} = V_{DDA} \times \frac{DOR}{4095}$$



14.4.7 DAC トリガ選択

TENx 制御ビットがセットされている場合、外部イベント（タイマカウンタ、外部割り込みラインなど）によって変換をトリガできます。以下の表に示すように、8 つの中のどのイベントが変換をトリガするかは、TSELx[2:0] 制御ビットによって決まります。

DAC インタフェースが選択されたタイマ TRGO 出力または選択された外部割り込みライン 9 の立ち上がりエッジを検出するたびに、DAC_DHRx レジスタに最後に格納されたデータが DAC_DORx レジスタに転送されます。DAC_DORx レジスタは、トリガが発生してから 3 APB サイクル後に更新されます。

ソフトウェアトリガが選択されている場合、変換は、SWTRIG ビットがセットされると開始されず。SWTRIG ビットは、DAC_DHRx レジスタの内容が DAC_DORx レジスタにロードされると、ハードウェアによってリセットされます。

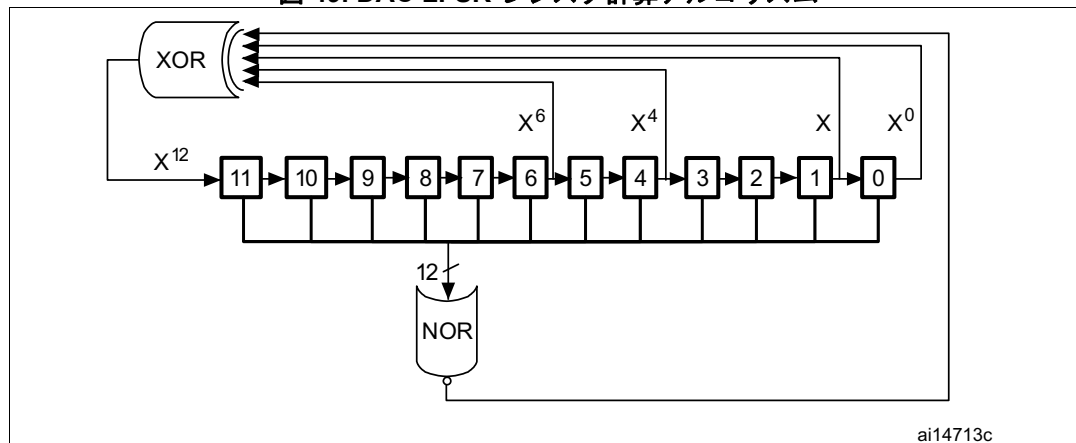
注： *ENx ビットがセットされているときには、TSELx[2:0] ビットを変更することはできません。*

ソフトウェアトリガが選択されているときには、DAC_DHRx レジスタから DAC_DORx レジスタへの転送は、わずか 1 APB クロックサイクルで行われます。

14.5 ノイズ生成 (STM32F07x および STM32F09x デバイス)

リニアフィードバックシフトレジスタ (LFSR) を使用して、可変振幅の擬似ノイズを生成することができます。DAC ノイズ生成を選択するには、WAVEx[1:0] に“01”をセットします。LFSRにプリロードされる値は 0xAAA です。このレジスタは、各トリガイベントの 3 APB クロックサイクル後に、以下の特定の計算アルゴリズムに従って更新されます。

図 49. DAC LFSR レジスタ計算アルゴリズム

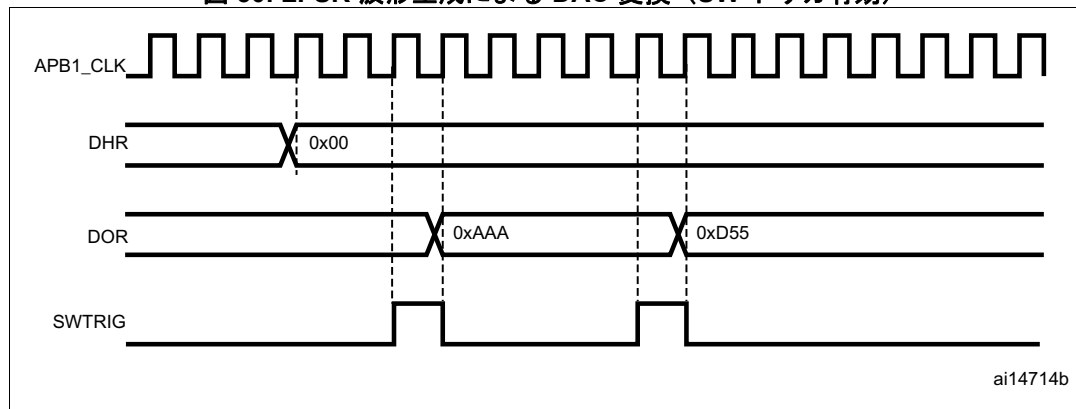


LFSR 値は、DAC_CR レジスタの MAMPx[3:0] ビットによって部分的または全体的にマスクでき、オーバーフローなしに DAC_DHRx の内容に加算され、DAC_DORx レジスタに格納されます。

LFSR が 0x0000 の場合、“1”がインジェクトされます (アンチロックアップメカニズム)。

WAVEx[1:0] ビットをリセットすることによって、LFSR 波形生成をリセットできます。

図 50. LFSR 波形生成による DAC 変換 (SW トリガ有効)



注： ノイズ生成のためには、DAC_CR レジスタの TENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

14.6 三角波生成 (STM32F07x および STM32F09x デバイス)

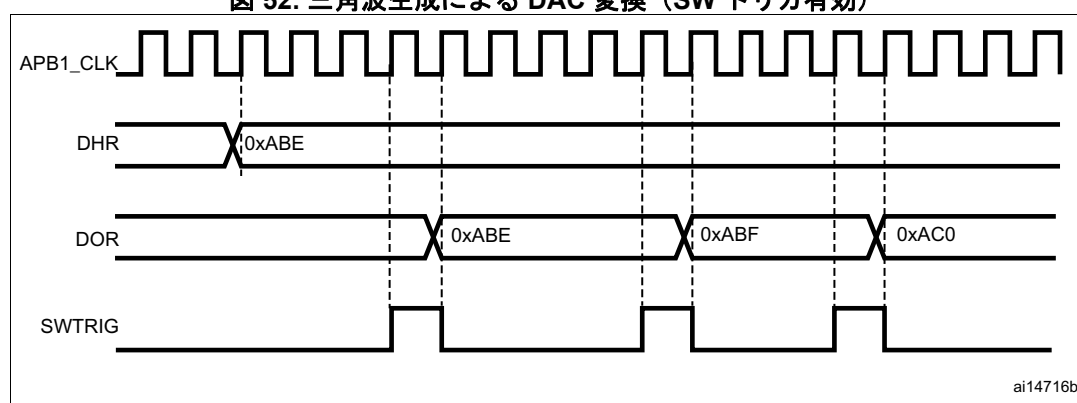
DC または低周波数信号上に、小さな振幅の三角波を追加することが可能です。DAC 三角波生成を選択するには、WAVEx[1:0] を“10”にセットします。振幅は、DAC_CR レジスタのMAMPx[3:0] ビットを介して設定されます。内部三角波カウンタは、各トリガイイベントの 3 APB クロックサイクル後にインクリメントされます。このカウンタの値は、オーバーフローなしに DAC_DHRx レジスタに加えられ、合計は DAC_DORx レジスタに格納されます。三角波カウンタは、MAMPx[3:0] ビットによって定義された最大振幅以上になるまでインクリメントされます。設定された振幅に達すると、カウンタは 0 にデクリメントされ、再びインクリメントが開始されます。

WAVEx[1:0] ビットをリセットすることによって、三角波生成をリセットすることができます。

図 51. DAC 三角波生成



図 52. 三角波生成による DAC 変換 (SW トリガ有効)



注： 三角波生成のためには、DAC_CR レジスタのTENx ビットをセットすることによって、DAC トリガを有効にしなければなりません。

DAC を有効にするには、その前に MAMPx[3:0] ビットを設定する必要があります。そうしないと、これらのビットは変更できません。

14.7 DMA リクエスト

各 DAC チャンネルは、DMA 機能を備えています。DAC チャンネルの DMA リクエストは、2 つの DMA チャンネルを使用して処理されます。

DAC DMA リクエストは、DMAENx ビットがセットされているときに、外部トリガ（ソフトウェアトリガでなく）が発生したときに生成されます。その場合、DAC_DHRx レジスタの値が DAC_DORx レジスタに転送されます。

デュアルモードでは、両方の DMAENx ビットがセットされている場合、2 つの DMA リクエストが生成されます。1 つの DMA リクエストしか必要ない場合には、対応する DMAENx ビットだけをセットしてください。これにより、アプリケーションは 1 つの DMA リクエストと一意な DMA チャンネルを使用して、両方の DAC チャンネルをデュアルモードで管理することができます。

コード例については、「付録」セクション [A.8.12 : DMA 初期化のコード例](#) を参照してください。

DMA アンダーラン

DAC DMA リクエストはキューされないため、最初の外部トリガに対する確認応答が受信される（最初のリクエスト）前に 2 番目の外部トリガが発生すると、新しいリクエストは発行されず、DAC_SR レジスタの DMA チャンネル x アンダーランフラグ DMAUDRx がセットされてエラー状態を報告します。続いて DMA データ転送が無効になり、その後の DMA リクエストは処理されません。DAC チャンネル x は、古いデータを変換し続けます。

ソフトウェアでは、“1”を書き込むことによって DMAUDRx フラグをクリアし、使用された DMA ストリームの DMAEN ビットをクリアし、DMA と DAC のチャンネル x を再初期化して転送を正しくリスタートさせてください。また、DAC トリガ変換周波数を変更するか、または DMA の負荷を軽減して、新しい DMA アンダーランを回避してください。最後に、DMA データ転送と変換トリガを有効にすることによって DAC 変換を再開することができます。

各 DAC チャンネルでは、DAC_CR レジスタの対応する DMAUDRIEx ビットが有効にされた場合、割り込みも生成されます。

14.8 DAC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

14.8.1 DAC 制御レジスタ (DAC_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2	EN2
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL1[2:0]			TEN1	BOFF1	EN1
		rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DMAUDRIE2** : DAC チャンネル2 DMA アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル2 DMA アンダーラン割り込みは無効です。

1 : DAC チャンネル2 DMA アンダーラン割り込みは有効です。

注 : このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 28 **DMAEN2** : DAC チャンネル2 DMA イネーブル

このビットは、ソフトウェアによってセット/クリアされます。

0 : DAC チャンネル2 DMA モードは無効です。

1 : DAC チャンネル2 DMA モードは有効です。

注 : このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 27:24 **MAMP2[3:0]** : DAC チャンネル 2 マスク/振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

0000 : LFSR/三角波振幅のアンマスクビット 0 は 1 に等しい。

0001 : LFSR/三角波振幅のアンマスクビット [1:0] は 3 に等しい。

0010 : LFSR/三角波振幅のアンマスクビット [2:0] は 7 に等しい。

0011 : LFSR/三角波振幅のアンマスクビット [3:0] は 15 に等しい。

0100 : LFSR/三角波振幅のアンマスクビット [4:0] は 31 に等しい。

0101 : LFSR/三角波振幅のアンマスクビット [5:0] は 63 に等しい。

0110 : LFSR/三角波振幅のアンマスクビット [6:0] は 127 に等しい。

0111 : LFSR/三角波振幅のアンマスクビット [7:0] は 255 に等しい。

1000 : LFSR/三角波振幅のアンマスクビット [8:0] は 511 に等しい。

1001 : LFSR/三角波振幅のアンマスクビット [9:0] は 1023 に等しい。

1010 : LFSR/三角波振幅のアンマスクビット [10:0] は 2047 に等しい。

≥ 1011 : LFSR/三角波振幅のアンマスクビット [11:0] は 4095 に等しい。

注 : このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 23:22 **WAVE2[1:0]** : DAC チャンネル 2 ノイズ／三角波生成イネーブル

これらのビットは、ソフトウェアによってセット／リセットされます。

00 : 波形生成は無効です。

01 : ノイズ波生成は有効です。

1x : 三角波生成は有効です。

注： ビット **TEN2=1** (DAC チャンネル2 トリガ有効) の場合のみ使用されます。

このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 21:19 **TSEL2[2:0]** : DAC チャンネル2 トリガ選択

これらのビットは、DAC チャンネル2 をトリガするために使用される外部イベントを選択します。

000 : タイマ 6 TRGO イベント

001 : タイマ 3 TRGO イベント

010 : タイマ 7 TRGO イベント

011 : タイマ 15 TRGO イベント

100 : タイマ 2 TRGO イベント

101 : 予約済み

110 : 外部割込みライン 9 (EXTI ライン 9)

111 : ソフトウェアトリガ

注： ビット **TEN2=1** (DAC チャンネル2 トリガ有効) の場合のみ使用されます。

このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 18 **TEN2** : DAC チャンネル2 トリガイネーブル

このビットは、DAC チャンネル2 トリガを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 2 トリガは無効であり、DAC_DHRx レジスタに書き込まれたデータは、1 APB クロックサイクル後に DAC_DOR2 レジスタに転送されます。

1 : DAC チャンネル 2 トリガは有効であり、DAC_DHRx レジスタから転送されたデータは、3 APB クロックサイクル後に DAC_DOR2 レジスタに転送されます。

注： ソフトウェアトリガが選択されているときには、DAC_DHRx から DAC_DOR2 レジスタへの転送には、わずか 1 APB クロックサイクルで行われます。

注： このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 17 **BOFF2** : DAC チャンネル2 出力バッファディセーブル

このビットは、DAC チャンネル2 出力バッファを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル2 出力バッファは有効です。

1 : DAC チャンネル2 出力バッファは無効です。

注： このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 16 **EN2** : DAC チャンネル2 イネーブル

このビットは、DAC チャンネル2 を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル2 は無効です。

1 : DAC チャンネル2 は有効です。

注： このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 15:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **DMAUDRIE1** : DAC チャンネル 1 DMA アンダーラン割り込みイネーブル

このビットは、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 DMA アンダーラン割り込みは無効です。

1 : DAC チャンネル 1 DMA アンダーラン割り込みは有効です。

ビット 12 **DMAEN1** : DAC チャンネル 1 DMA イネーブル

このビットは、ソフトウェアによってセット/クリアされます。

- 0 : DAC チャンネル 1 DMA モードは無効です。
- 1 : DAC チャンネル 1 DMA モードは有効です。

ビット 11:8 **MAMP1[3:0]** : DAC チャンネル 1 マスク/振幅セレクタ

これらのビットは、波形生成モードのマスクまたは三角波生成モードの振幅を選択するために、ソフトウェアによって書き込まれます。

- 0000 : LFSR/三角波振幅のアンマスクビット 0 は 1 に等しい。
- 0001 : LFSR/三角波振幅のアンマスクビット [1:0] は 3 に等しい。
- 0010 : LFSR/三角波振幅のアンマスクビット [2:0] は 7 に等しい。
- 0011 : LFSR/三角波振幅のアンマスクビット [3:0] は 15 に等しい。
- 0100 : LFSR/三角波振幅のアンマスクビット [4:0] は 31 に等しい。
- 0101 : LFSR/三角波振幅のアンマスクビット [5:0] は 63 に等しい。
- 0110 : LFSR/三角波振幅のアンマスクビット [6:0] は 127 に等しい。
- 0111 : LFSR/三角波振幅のアンマスクビット [7:0] は 255 に等しい。
- 1000 : LFSR/三角波振幅のアンマスクビット [8:0] は 511 に等しい。
- 1001 : LFSR/三角波振幅のアンマスクビット [9:0] は 1023 に等しい。
- 1010 : LFSR/三角波振幅のアンマスクビット [10:0] は 2047 に等しい。
- ≥ 1011 : LFSR/三角波振幅のアンマスクビット [11:0] は 4095 に等しい。

ビット 7:6 **WAVE1[1:0]** : DAC チャンネル 1 ノイズ/三角波生成イネーブル

これらのビットは、ソフトウェアによってセット/クリアされます。

- 00 : 波形生成は無効です。
- 01 : ノイズ波生成は有効です。
- 1x : 三角波生成は有効です。

注： ビット $TEN1 = 1$ (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

ビット 5:3 **TSEL1[2:0]** : DAC チャンネル 1 トリガ選択

これらのビットは、DAC チャンネル 1 をトリガするために使用される外部イベントを選択します。

- 000 : タイマ 6 TRGO イベント
- 001 : タイマ 3 TRGO イベント
- 010 : タイマ 7 TRGO イベント
- 011 : タイマ 15 TRGO イベント
- 100 : タイマ 2 TRGO イベント
- 101 : 予約済み
- 110 : 外部割込みライン 9 (EXTI ライン 9)
- 111 : ソフトウェアトリガ

注： ビット $TEN1 = 1$ (DAC チャンネル 1 トリガ有効) の場合のみ使用されます。

ビット 2 TEN1 : DAC チャンネル 1 トリガイネーブル

このビットは、DAC チャンネル 1 トリガを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 トリガは無効であり、DAC_DHRx レジスタに書き込まれたデータは、1 APB クロックサイクル後に DAC_DOR1 レジスタに転送されます。

1 : DAC チャンネル 1 トリガは有効であり、DAC_DHRx レジスタから転送されたデータは、3 APB クロックサイクル後に DAC_DOR1 レジスタに転送されます。

注： ソフトウェアトリガが選択されているときには、DAC_DHRx レジスタから DAC_DOR1 レジスタへの転送は、わずか 1 APB クロックサイクルで行われます。

ビット 1 BOFF1 : DAC チャンネル 1 出力バッファディセーブル

このビットは、DAC チャンネル 1 出力バッファを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 出力バッファは有効です。

1 : DAC チャンネル 1 出力バッファは無効です。

ビット 0 EN1 : DAC チャンネル 1 イネーブル

このビットは、DAC チャンネル 1 を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : DAC チャンネル 1 は無効です。

1 : DAC チャンネル 1 は有効です。

14.8.2 DAC ソフトウェアトリガレジスタ (DAC_SWTRIGR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
														w	w

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **SWTRIG2** : DAC チャンネル2 ソフトウェアトリガ

このビットは、ソフトウェアトリガを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : ソフトウェアトリガは無効です。

1 : ソフトウェアトリガは有効です。

注 : このビットは、DAC_DHR2 レジスタの値が DAC_DOR2 レジスタにロードされると、ハードウェアによってクリアされます (1 APB クロックサイクル後)。

このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 0 **SWTRIG1** : DAC チャンネル 1 ソフトウェアトリガ

このビットは、ソフトウェアトリガを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : ソフトウェアトリガは無効です。

1 : ソフトウェアトリガは有効です。

注 : このビットは、DAC_DHR1 レジスタの値が DAC_DOR1 レジスタにロードされると、ハードウェアによってクリアされます (1 APB クロックサイクル後)。

14.8.3 DAC チャンネル 1 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ

これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

14.8.4 DAC チャンネル 1 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L1)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												v	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ
これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

14.8.5 DAC チャンネル 1 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R1)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ
これらのビットは、DAC チャンネル 1 の 8 ビットデータを指定するために、ソフトウェアによって書き込まれます。

14.8.6 DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

14.8.7 DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ
(DAC_DHR12L2)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット左詰めデータ

これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

14.8.8 DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ
(DAC_DHR8R2)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **DACC2DHR[7:0]** : DAC チャンネル2 8ビット右詰めデータ

これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

14.8.9 デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット右詰めデータ
これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット右詰めデータ
これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

14.8.10 デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW				

ビット 31:20 **DACC2DHR[11:0]** : DAC チャンネル2 12ビット左詰めデータ
これらのビットは、DAC チャンネル2 の 12ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 19:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **DACC1DHR[11:0]** : DAC チャンネル 1 の 12 ビット左詰めデータ
これらのビットは、DAC チャンネル 1 の 12 ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

14.8.11 デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:8 **DACC2DHR[7:0]** : DAC チャンネル2 8ビット右詰めデータ
これらのビットは、DAC チャンネル2 の 8ビットデータを指定するために、ソフトウェアによって書き込まれます。

ビット 7:0 **DACC1DHR[7:0]** : DAC チャンネル 1 の 8 ビット右詰めデータ
これらのビットは、DAC チャンネル 1 の 8 ビットデータを指定するために、ソフトウェアによって書き込まれます。

14.8.12 DAC チャンネル 1 データ出力レジスタ (DAC_DOR1)

アドレスオフセット : 0x2C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC1DOR[11:0]** : DAC チャンネル 1 データ出力
これらのビットは読み出し専用であり、DAC チャンネル 1 のデータ出力を含みます。

14.8.13 DAC チャンネル 2 データ出力レジスタ (DAC_DOR2)

アドレスオフセット : 0x30
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **DACC2DOR[11:0]** : DAC チャンネル2 データ出力
これらのビットは読み出し専用であり、DAC チャンネル2 のデータ出力を含みます。

14.8.14 DAC ステータスレジスタ (DAC_SR)

アドレスオフセット : 0x34
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rc_w1													

ビット 31:30 予約済みであり、リセット値に保持する必要があります。

ビット 29 **DMAUDR2** : DAC チャンネル 2 DMA アンダーランフラグSTM32F07x および STM32F09x デバイスで使用できます。
このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。
0 : DAC チャンネル2 に DMA アンダーランエラー条件は発生しませんでした。
1 : DAC チャンネル2 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル2 変換を駆動しています)。
注: このビットはデュアルモードでのみ使用できます。シングルモードでは予約済みです。

ビット 28:14 予約済みであり、リセット値に保持する必要があります。

ビット 13 **DMAUDR1** : DAC チャンネル 1 の DMA アンダーランフラグ
このビットは、ハードウェアによってセットされ、(1 を書き込むことによって) ソフトウェアによってクリアされます。
0 : DAC チャンネル 1 に DMA アンダーランエラー条件は発生しませんでした。
1 : DAC チャンネル1 に DMA アンダーランエラー条件が発生しました (現在選択されているトリガは、DMA サービス機能のレートを上回る周波数で DAC チャンネル 1 変換を駆動しています)。

ビット 12:0 予約済みであり、リセット値に保持する必要があります。

14.8.15 DAC レジスタマップ

表 51 に DAC レジスタの要約を示します。

表 51. DAC レジスタマップ とリセット値

オフ セット	レジスタ名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x00	DAC_CR	Res.	Res.	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL2[2:0]			TEN2	BOFF2	EN2	Res.	Res.	DMAUDRIE1	DMAEN1	MAMP[3:0]				WAVE[1:0]		TSEL[2:0]			TEN1	BOFF1	EN1			
	リセット値			0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x04	DAC_SWTRIGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1			
	リセット値																														0	0				
0x08	DAC_DHR12R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]														
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0			
0x0C	DAC_DHR12L1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]											Res.	Res.	Res.	Res.				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0								
0x10	DAC_DHR8R1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]										
	リセット値																									0	0	0	0	0	0	0	0			
0x14	DAC_DHR12R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]														
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0			
0x18	DAC_DHR12L2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]											Res.	Res.	Res.	Res.				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0								
0x1C	DAC_DHR8R2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]										
	リセット値																									0	0	0	0	0	0	0	0			
0x20	DAC_DHR12RD	Res.	Res.	Res.	Res.	DACC2DHR[11:0]											Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]												
	リセット値					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0		
0x24	DAC_DHR12LD	DACC2DHR[11:0]											Res.	Res.	Res.	Res.	DACC1DHR[11:0]											Res.	Res.	Res.	Res.					
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0							
0x28	DAC_DHR8RD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							DACC1DHR[7:0]											
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x2C	DAC_DOR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DOR[11:0]														
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0			
0x30	DAC_DOR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DOR[11:0]														
	リセット値																					0	0	0	0	0	0	0	0	0	0	0	0			

表 51. DAC レジスタマップ (続き) とリセット値 (続き)

オフ セット	レジスタ名		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	DAC_SR	Res.	Res.	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値			0																	0													

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

15 コンパレータ (COMP)

このセクションは、STM32F05x、STM32F07x および STM32F09x デバイスにのみ適用されます。

15.1 概要

STM32F05x、STM32F07x、および STM32F09x デバイスは 2 個の汎用コンパレータ COMP1 および COMP2 を内蔵しており、それらはスタンダロンドバイス (すべての端子を I/O で使用可能) として、またはタイマと組み合わせて使用することができます。

コンパレータは以下のようなさまざまな機能に使用できます。

- アナログ信号によってトリガされる低電力モードからのウェイクアップ
- アナログ信号調節
- DAC と タイマからの PWM 出力を組み合わせた場合のサイクルごとの電流制御ループ

15.2 COMP の主な機能

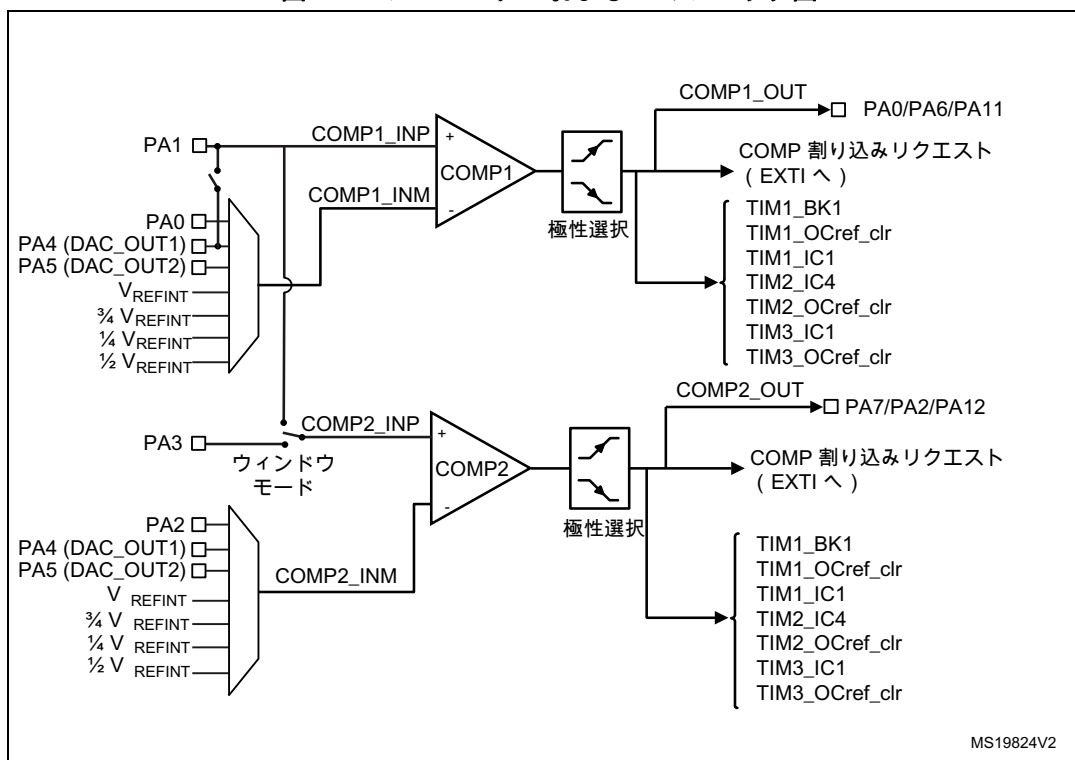
- レールツーレールコンパレータ
- 各コンパレータは、電圧を柔軟に選択できるように、次のような正入力および設定可能な負入力を備えています。
 - 3 本の I/O ピン
 - DAC
 - スケラ (バッファ付き分圧器) が提供する内部基準電圧および 3 つの約数 (1/4、1/2、3/4)
- プログラム可能なヒステリシス
- プログラム可能なスピード/消費
- 出力先を I/O またはトリガに使用するタイマ入力に変更することができます。
 - OCREF_CLR イベント (サイクルごとの電流制御)
 - 高速 PWM 停止のブレーキイベント
- COMP1 および COMP2 コンパレータをウィンドウコンパレータで結合することができます。
- 各コンパレータは、SLEEP および STOP モードからのウェイクアップ (EXTI コントローラ経由) を備えた割り込み生成機能を持っています。

15.3 COMP の機能説明

15.3.1 COMP ブロック図

コンパレータのブロック図を [図 53 : コンパレータ 1 および 2 のブロック図](#) に示します。

図 53. コンパレータ 1 および 2 のブロック図



15.3.2 COMP ピンおよび内部信号

コンパレータ入力として使用される I/O は、GPIO レジスタのアナログモードで設定する必要があります。

コンパレータの出力は、データシートの「オルタネート機能配置」表に示されているオルタネート機能チャネルを使用して I/O に接続することができます。

出力先を、以下の目的に使用される各種タイマ入力に内部で変更することも可能です。

- BKIN を使用した PWM 信号の緊急停止
- OCREF_CLR 入力を使用したサイクルごとの電流制御
- タイミング測定のための入力キャプチャ

コンパレータの出力先を内部および外部から一斉に変更することが可能です。

15.3.3 COMP のリセットおよびクロック

クロックコントローラによって提供される COMP クロックは、PCLK (APB クロック) と同期しています。

RCC コントローラには、クロックイネーブル制御ビットは提供されていません。リセットおよびクロックイネーブルビットは COMP と SYSCFG に共通です。

注： **重要：極性選択ロジックおよびポートへの出力先変更は、PCLK クロックから個別に機能します。これにより、コンパレータは STOP モードでも機能することができます。**

15.3.4 コンパレータのロック機構

コンパレータは、過電流保護や熱保護などの安全上の目的で使用されます。特定の機能安全要件があるようなアプリケーションの場合、万が一誤ったレジスタへのアクセスやプログラムカウンタの破壊が起こった場合に、コンパレータのプログラミングが絶対に変更されないようにすることが必要です。

このような目的で、コンパレータの制御レジスタおよびステータスレジスタを書き込み保護することができます (読み出し専用)。

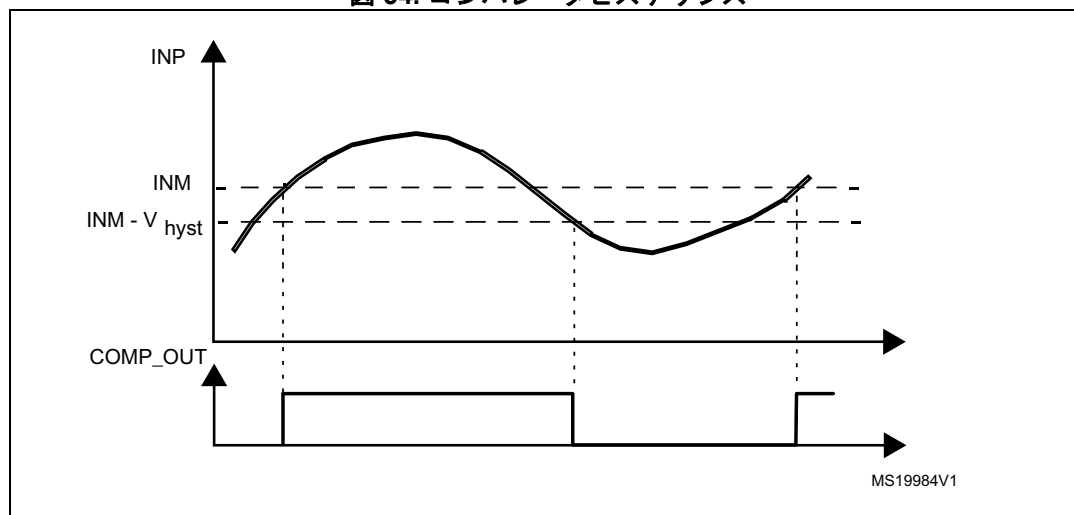
プログラミングが完了したら、COMP_CSR の 30:16 および 15:0 ビットを使用して、COMPxLOCK ビットを 1 にセットすることができます。これにより、COMPxLOCK ビットを含む COMP_CSR レジスタ全体が読み出し専用になります。

書き込み保護は MCU リセットによってのみリセット可能です。

15.3.5 ヒステリシス

コンパレータには、ノイズの多い信号での疑似出力遷移を避けるために、プログラム可能なヒステリシスが含まれています。不要な場合は、ヒステリシスを無効にすることができます (低電力モードを終了する場合など)。これにより、外部コンポーネントを使用してヒステリシス値を強制できます。

図 54. コンパレータヒステリシス



15.3.6 電力モード

コンパレータの電力消費と伝搬遅延を調節して、特定のアプリケーションに最適なトレードオフを実現させることができます。COMP_CSR レジスタの COMPxMODE[1:0] ビットは次のようにプログラミングすることができます。

- 00 : 高速／フル電力
- 01 : 中速／中電力
- 10 : 低速／低電力
- 11 : 超低速／超低電力

15.4 COMP 割り込み

コンパレータの出力は拡張割り込み／イベントコントローラに内部的に接続されます。各コンパレータには専用の EXTI ラインがあり、割り込みまたはイベントを生成することができます。低電力モードを終了するときにも同じ方法が使用されます。

詳細については、割り込みおよびイベントのセクションを参照してください。

15.5 COMP レジスタ

15.5.1 COMP 制御およびステータスレジスタ (COMP_CSR)

アドレスオフセット : 0x1C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP2LOCK	COMP2OUT	COMP2HYST [1:0]		COMP2POL	COMP2OUTSEL[2:0]			WNDWEN	COMP2INSEL[2:0]			COMP2MODE [1:0]		Res.	COMP2EN
rw	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r		rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP1LOCK	COMP1OUT	COMP1HYST [1:0]		COMP1POL	COMP1OUTSEL[2:0]			Res.	COMP1INSEL[2:0]			COMP1MODE [1:0]		COMP1SW1	COMP1EN
rw	r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r		rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

ビット 31 **COMP2LOCK** : コンパレータ 2 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

コンパレータ 2 のすべてのコントロールビットを読み出し専用にすることができます。

0 : COMP_CSR[31:16] ビットは読み出し／書き込み用です。

1 : COMP_CSR[31:16] ビットは読み出し専用です。

ビット 30 **COMP2OUT** : コンパレータ 2 出力

この読み出し専用ビットはコンパレータ 2 の出力状態のコピーです。

0 : 出力はロー (反転入力を下回る非反転入力)

1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 29:28 **COMP2HYST[1:0]** コンパレータ 2 ヒステリシス

これらのビットは、ヒステリシスレベルを制御します。

00 : ヒステリシスなし

01 : 低ヒステリシス

10 : 中ヒステリシス

11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

ビット 27 **COMP2POL** : コンパレータ 2 出力の極性

このビットは、コンパレータ 2 出力を反転させるために使用します。

0 : 出力は反転されません。

1 : 出力は反転されます。

ビット 26:24 **COMP2OUTSEL[2:0]** : コンパレータ 2 出力選択

これらのビットによってコンパレータ出力の転送先を選択します。

000 : 選択なし

001 : タイマ 1 ブレーク入力

010 : タイマ 1 入力キャプチャ 1

011 : タイマ 1 OCrefclear 入力

100 : タイマ 2 入力キャプチャ 4

101 : タイマ 2 OCrefclear 入力

110 : タイマ 3 入力キャプチャ 1

111 : タイマ 3 OCrefclear 入力

ビット 23 **WNDWEN** : ウィンドウモード有効

このビットは COMP2 の非反転入力に COMP1 の非反転入力に接続し、COMP1 は同時に PA3 から切断されます。

- 0 : ウィンドウモード無効
- 1 : ウィンドウモード有効

ビット 22:20 **COMP2INSEL[2:0]** : コンパレータ 2 反転入力選択

これらのビットにより、コンパレータ 2 の反転入力に接続されたソースを選択できます。

- 000 : V_{REFINT} の 1/4
- 001 : V_{REFINT} の 1/2
- 010 : V_{REFINT} の 3/4
- 011 : V_{REFINT}
- 100 : COMP2_INM4 (有効な場合は、PA4 に DAC_OUT1 が割り当てられる)
- 101 : COMP2_INM5 (存在する場合で有効な場合は、PA5 に DAC_OUT2 が割り当てられる)
- 110 : COMP2_INM6 (PA2)
- 111 : 予約済み

ビット 19:18 **COMP2MODE[1:0]** : コンパレータ 2 モード

これらのビットはコンパレータ 2 の動作モードを制御し、スピード／消費を調節できます。

- 00 : 高速／フル電力
- 01 : 中速／中電力
- 10 : 低速／低電力
- 11 : 超低速／超低電力

ビット 17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **COMP2EN** : コンパレータ 2 イネーブル

このビットは、コンパレータ 2 のオン／オフを切り替えます。

- 0 : コンパレータ 2 無効
- 1 : コンパレータ 2 有効

ビット 15 **COMP1LOCK** : コンパレータ 1 ロック

このビットは 1 回のみ書き込み可能です。ソフトウェアによってセットされます。システムリセットによってのみクリアされます。

コンパレータ 1 のすべてのコントロールビットを読み出し専用にすることができます。

- 0 : COMP_CSR[15:0] ビットは読み出し／書き込み用です。
- 1 : COMP_CSR[15:0] ビットは書き込み用です。

ビット 14 **COMP1OUT** : コンパレータ 1 出力

この読み出し専用ビットはコンパレータ 1 の出力状態のコピーです。

- 0 : 出力はロー (反転入力を下回る非反転入力)
- 1 : 出力はハイ (反転入力を上回る非反転入力)

ビット 13:12 **COMP1HYST[1:0]** コンパレータ 1 ヒステリシス

これらのビットは、ヒステリシスレベルを制御します。

- 00 : ヒステリシスなし
- 01 : 低ヒステリシス
- 10 : 中ヒステリシス
- 11 : 高ヒステリシス

ヒステリシス値の電気的特性を参照してください。

ビット 11 **COMP1POL** : コンパレータ 1 出力の極性

このビットは、コンパレータ 1 出力を反転させるために使用します。

- 0 : 出力は反転されません。
- 1 : 出力は反転されます。



ビット 10:8 COMP1OUTSEL[2:0] : コンパレータ 1 出力選択

これらのビットによってコンパレータ 1 出力の転送先を選択します。

- 000 : 選択なし
- 001 : タイマ 1 ブレーク入力
- 010 : タイマ 1 入力キャプチャ 1
- 011 : タイマ 1 OCrefclear 入力
- 100 : タイマ 2 入力キャプチャ 4
- 101 : タイマ 2 OCrefclear 入力
- 110 : タイマ 3 入力キャプチャ 1
- 111 : タイマ 3 OCrefclear 入力

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 COMP1INSEL[2:0] : コンパレータ 1 反転入力選択

これらのビットは、コンパレータ 1 の反転入力に接続されたソースを選択します。

- 000 : V_{REFINT} の 1/4
- 001 : V_{REFINT} の 1/2
- 010 : V_{REFINT} の 3/4
- 011 : V_{REFINT}
- 100 : COMP1_INM4 (有効な場合は、PA4 に DAC_OUT1 が割り当てられる)
- 101 : COMP1_INM5 (存在する場合で有効な場合は、PA5 に DAC_OUT2 が割り当てられる)
- 110 : COMP1_INM6 (PA0)
- 111 : 予約済み

ビット 3:2 COMP1MODE[1:0] : コンパレータ 1 モード

これらのビットはコンパレータ 1 の動作モードを制御し、スピード/消費を調節できます。

- 00 : 高速/フル電力
- 01 : 中速/中電力
- 10 : 低速/低電力
- 11 : 超低速/超低電力

ビット 1 COMP1SW1 : コンパレータ 1 非反転入力 DAC スイッチ

このビットは、PA0 と PA4 (DAC) I/O のコンパレータ 1 非反転入力間のスイッチを閉じます。

- 0 : スイッチを開きます。
- 1 : スイッチを閉じます。

注 : このスイッチは、COMP1 非反転入力 (高抵抗スイッチ) などの信号をハイインピーダンス入力に変更することのみを目的としています。

ビット 0 COMP1EN : コンパレータ 1 イネーブル

このビットは COMP1 のオン/オフを切り替えます。

- 0 : コンパレータ 1 無効
- 1 : コンパレータ 1 有効

15.5.2 COMP レジスタマップ

次の表に コンパレータレジスタの一覧を示します。

表 52. COMP レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1C	COMP_CSR	COMP2LOCK	COMP2OUT	COMP2HYST[1:0]		COMP2POL	COMP2OUTSEL[2:0]			WNDWEN	COMP2INSEL[2:0]			COMP2MODE[1:0]		Res.	COMP2EN	COMP1LOCK	COMP1OUT	COMP1HYST[1:0]		COMP1POL	COMP1OUTSEL[2:0]		Res.	COMP1INSEL[2:0]			COMP1MODE[1:0]		COMP1SW1	COMP1EN	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

16 高機能制御タイマ (TIM1)

16.1 TIM1 の概要

高機能制御タイマ (TIM1) は、プログラム可能なプリスケアラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較、PWM、デッドタイムを挿入した相補 PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケアラと RCC クロックコントローラプリスケアラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

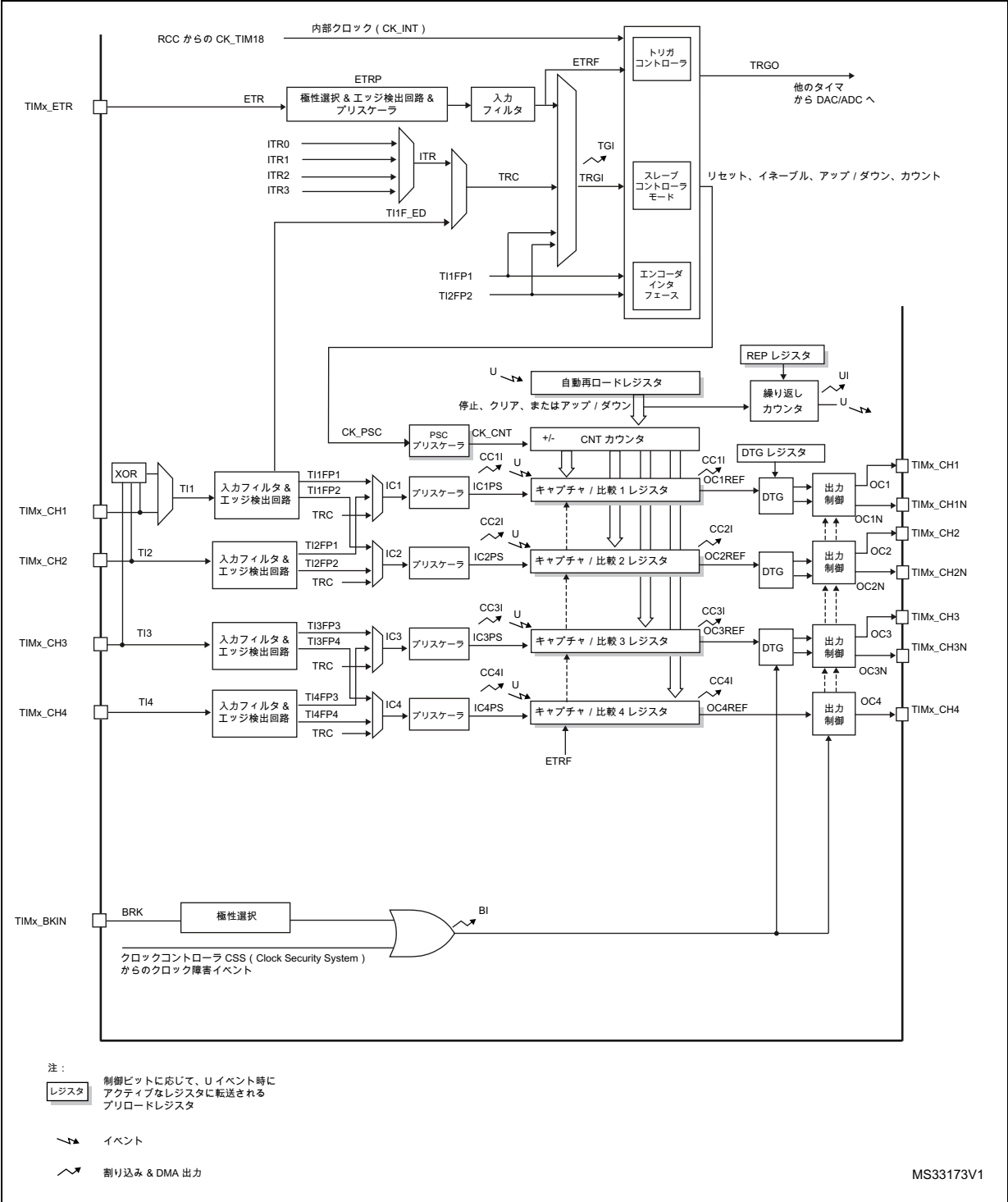
高機能制御タイマ (TIM1) と汎用タイマ (TIMx) は、互いに独立しており、リソースを共有しません。これらのタイマは、[セクション 16.3.20](#)に示すように、相互に同期させることができます。

16.2 TIM1 の主な特長

TIM1 タイマの主な機能は、次のとおりです。

- 16 ビットのアップカウンタ、ダウンカウンタ、アップ/ダウン自動再ロードカウンタ。
- 16 ビットのプログラム可能なプリスケアラ。カウンタクロック周波数を 1 から 65535 の間で分周でき、分周比の動作中の変更も可能。
- 次の機能を持つ、最大 4 つの独立チャネル。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成 (エッジアラインモードとセンターアラインモード)
 - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレイク入力。
- 以下のイベント時の割り込み/DMA 生成。
 - 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
 - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
 - 入力キャプチャ
 - 出力比較
 - ブレイク入力
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 55. 高機能制御タイマのブロック図



16.3 TIM1 の機能詳細

16.3.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)
- 繰り返しカウンタレジスタ (TIMx_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 57](#) と [図 58](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 56. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

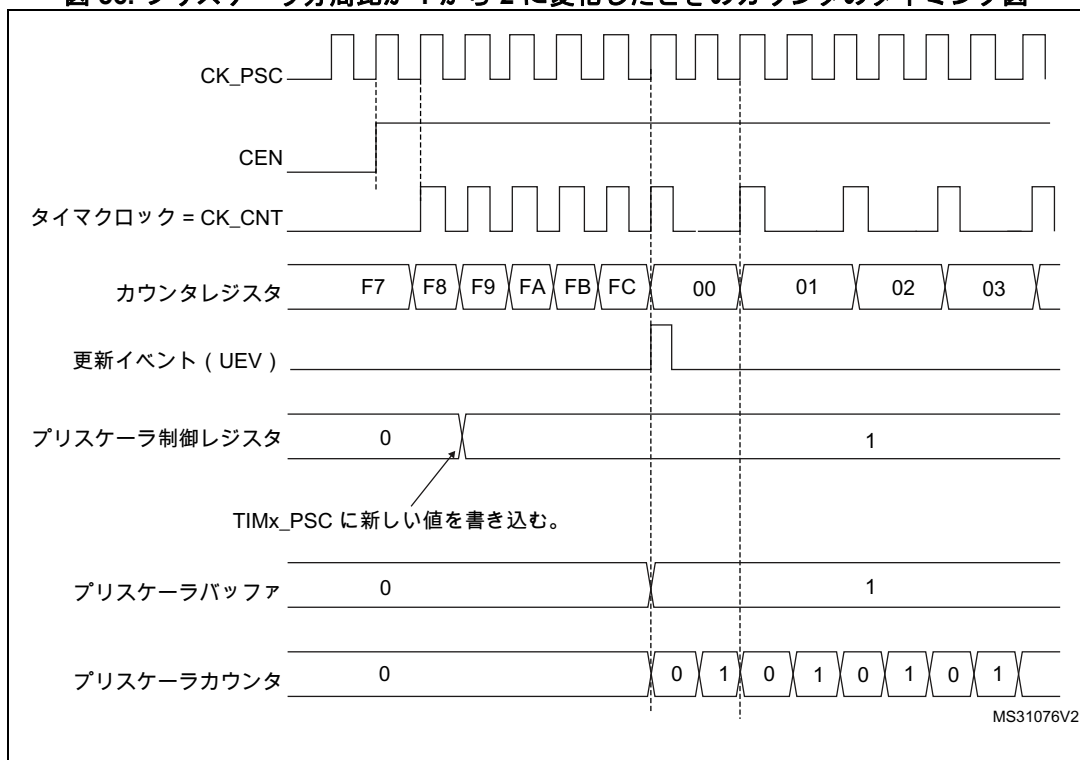
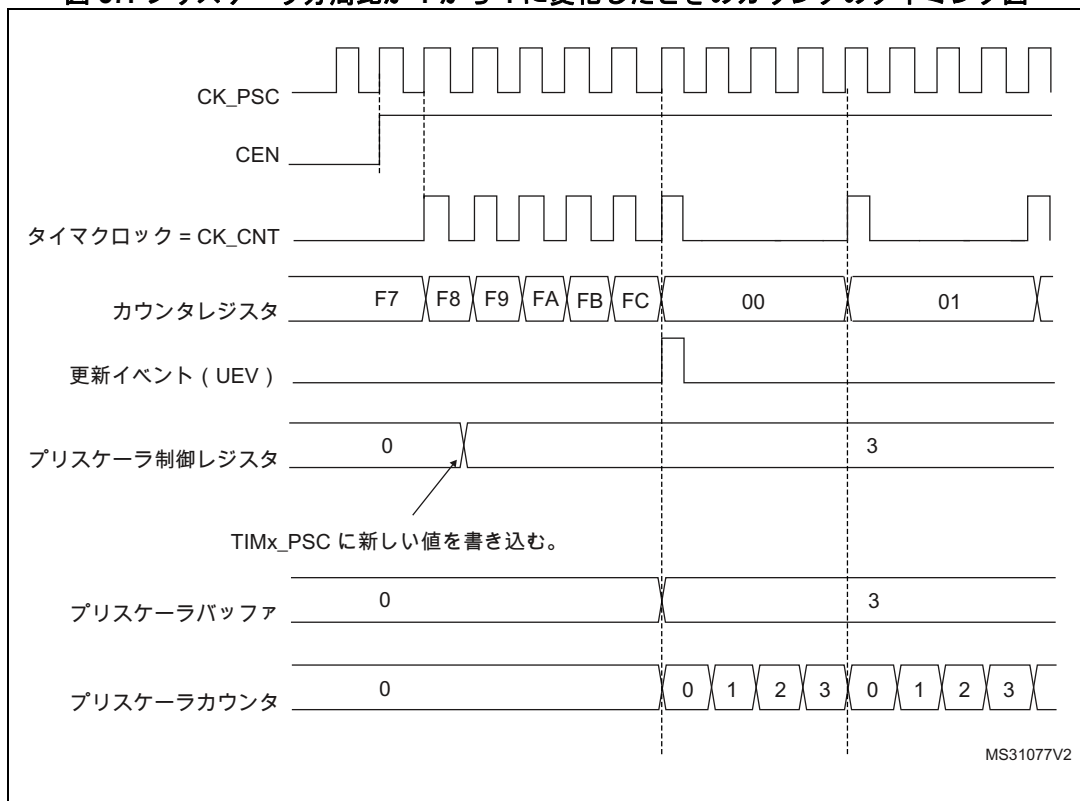


図 57. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



16.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx_RCR) までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされず。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 58. 内部クロック分周比が 1 の場合のカウンタのタイミング図

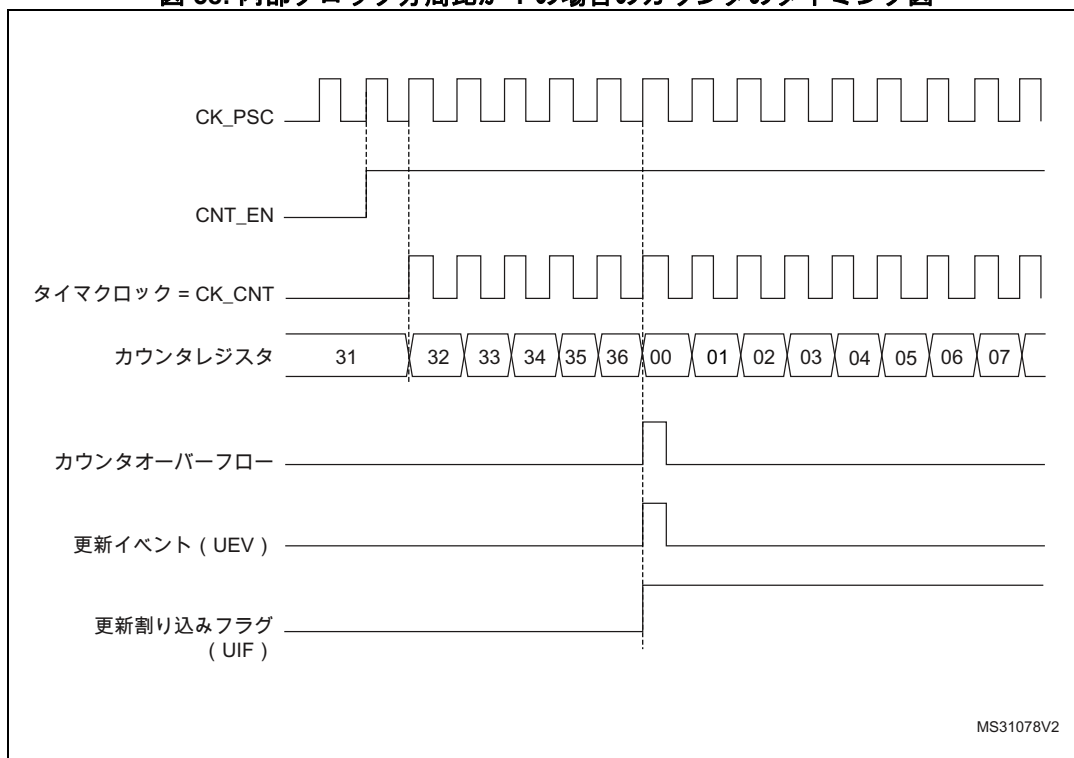


図 59. 内部クロック分周比が 2 の場合のカウンタのタイミング図

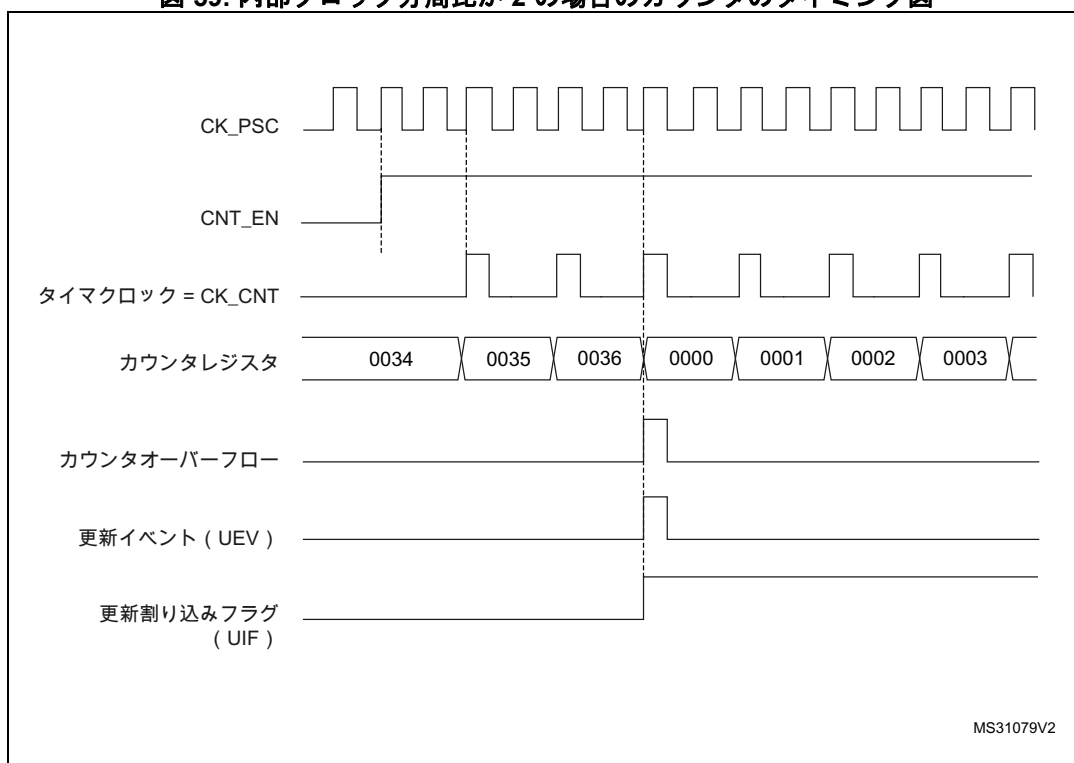


図 60. 内部クロック分周比が 4 の場合のカウンタのタイミング図

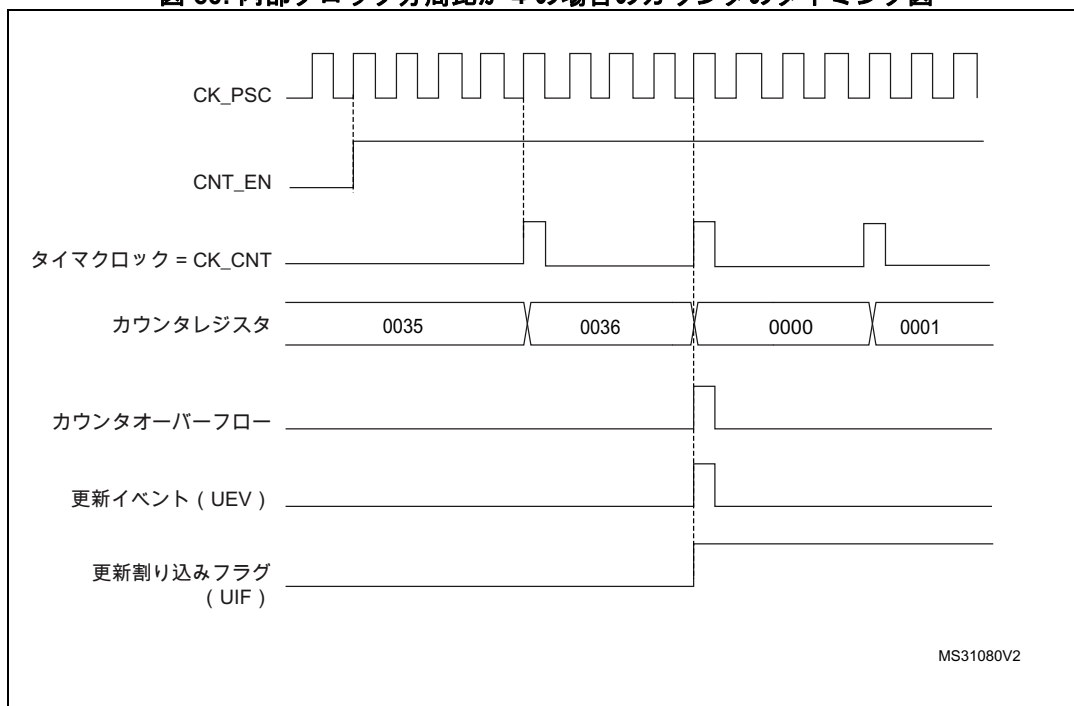


図 61. 内部クロック分周比が N の場合のカウンタのタイミング図

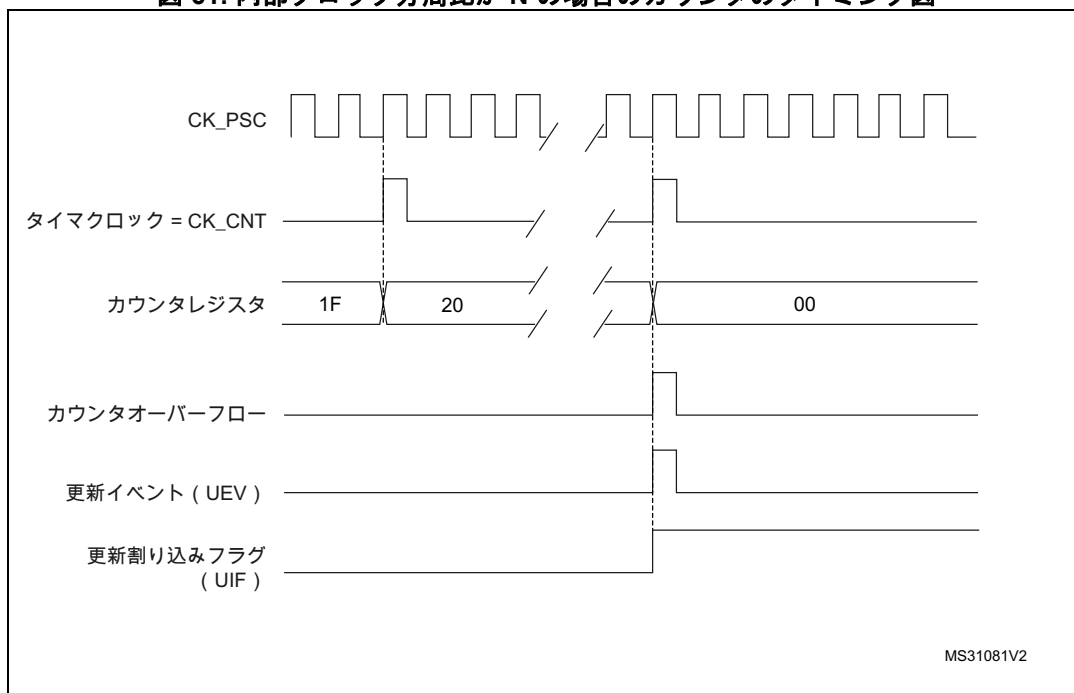


図 62. ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)

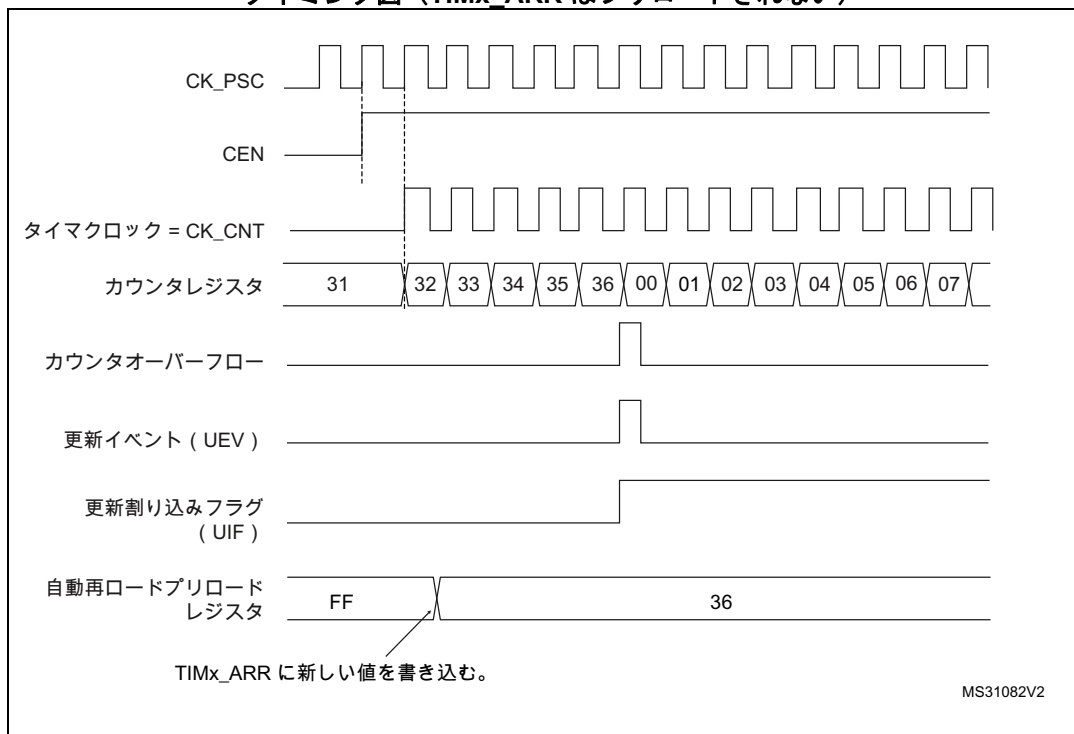
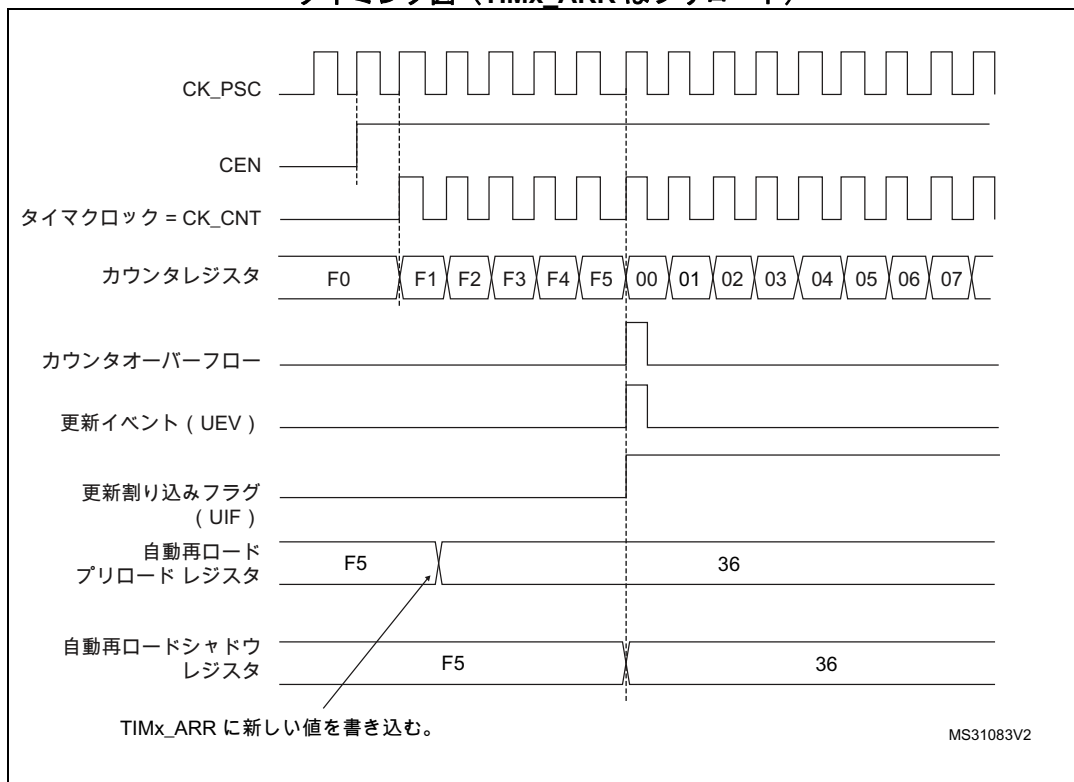


図 63. ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)



ダウンカウントモード

ダウンカウントモードでは、カウンタは自動再ロード値 (TIMx_ARR レジスタの内容) から 0 までカウントした後、自動再ロード値からカウントダウンをリスタートし、カウンタアンダーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx_RCR) までダウンカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのアンダーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値からリスタートしますが、プリスケアラのカウンタは 0 からリスタートします (しかし、プリスケアラ比は変化しません)。

さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタが再ロードされる前に自動再ロードが更新されるので、次の周期は予想通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 64. 内部クロック分周比が 1 の場合のカウンタのタイミング図

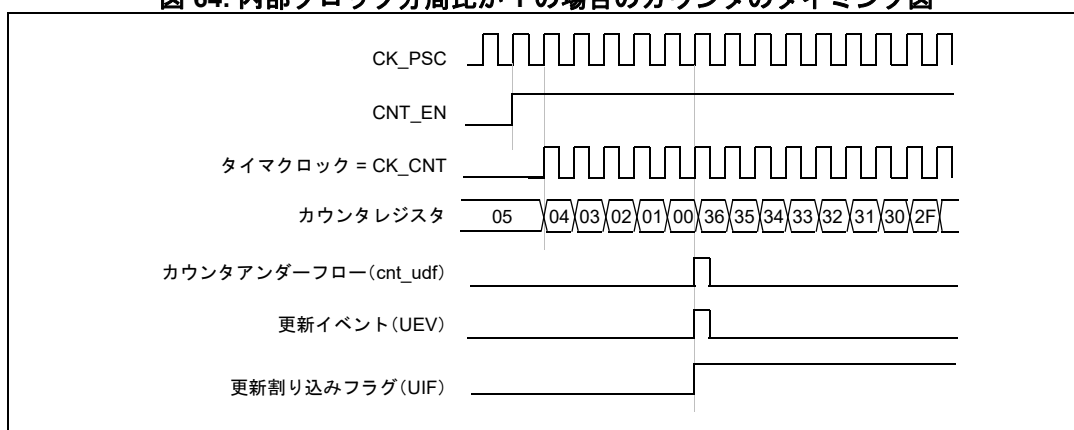


図 65. 内部クロック分周比が 2 の場合のカウンタのタイミング図

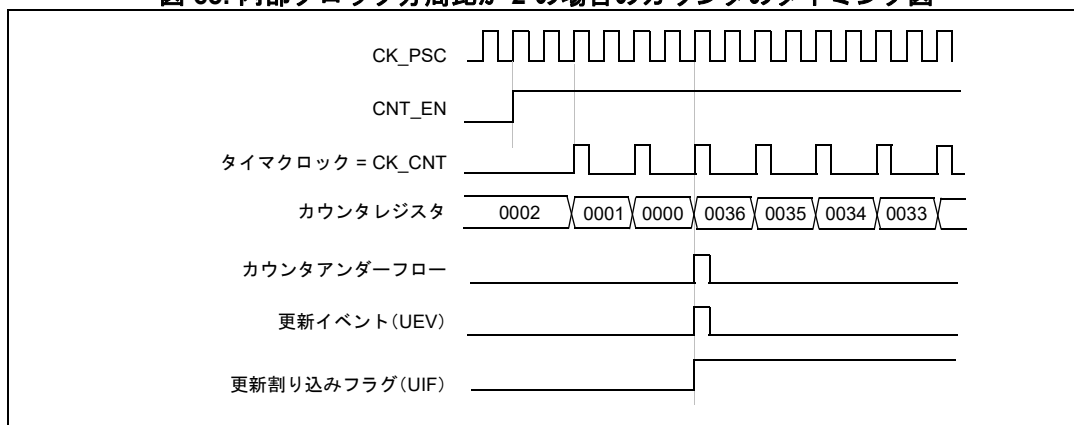


図 66. 内部クロック分周比が 4 の場合のカウンタのタイミング図

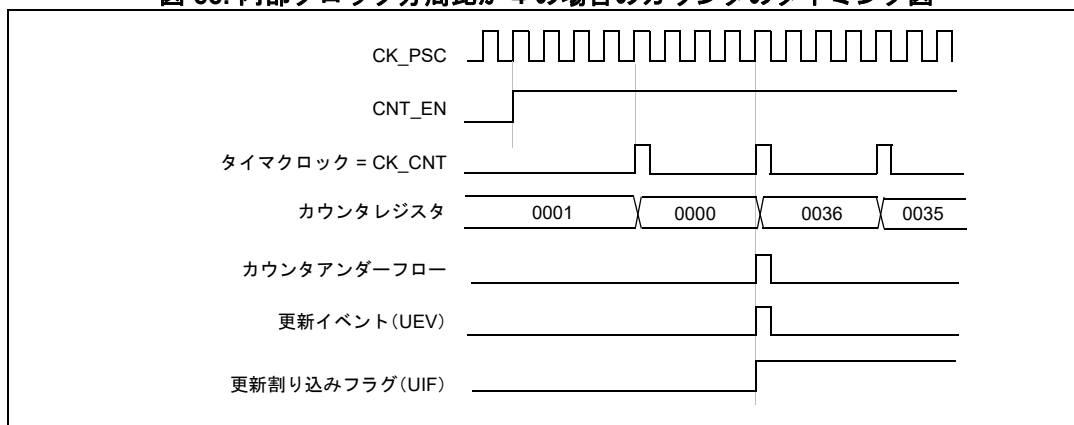


図 67. 内部クロック分周比が N の場合のカウンタのタイミング図

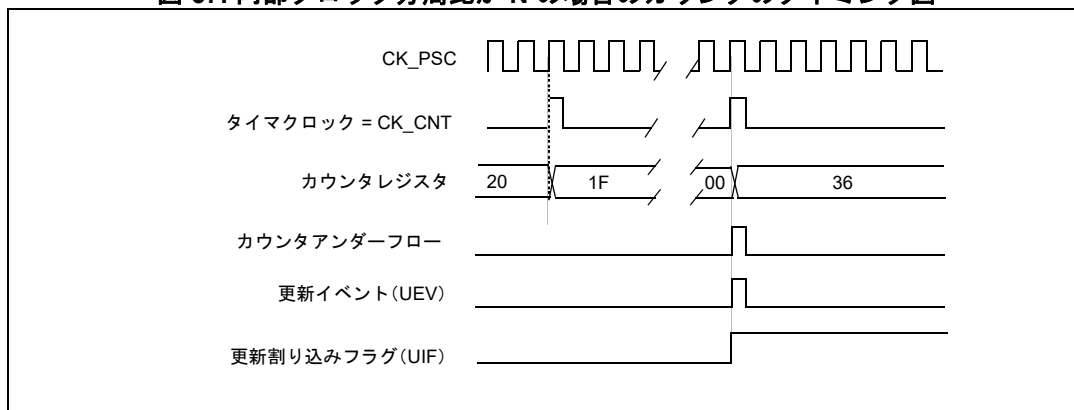
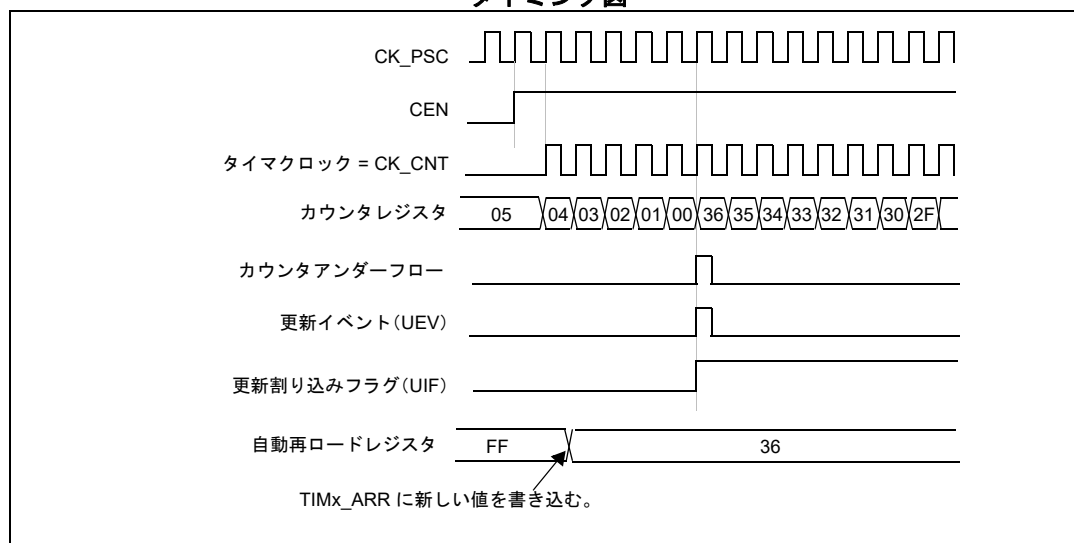


図 68. 繰り返しカウンタが使用されていない場合の更新イベント時のカウンタのタイミング図



センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが "00" に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS="01")、カウンタがカウントアップするとき (センターアラインモード 2、CMS="10")、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS="11") にセットされます。

このモードでは、TIMx_CR1 レジスタの方向ビット (DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この後 UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

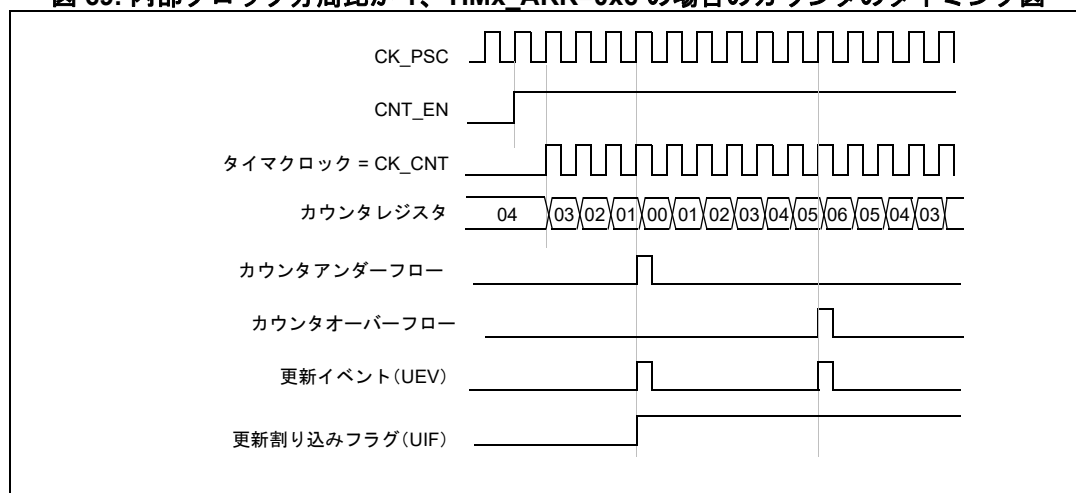
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合、UG ビットをセットすると UEV 更新イベントが生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 69. 内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターラインモード 1 が使用されています (詳細については、[セクション 16.4:TIM1 レジスタ \(350 ページ\)](#)を参照)。

図 70. 内部クロック分周比が 2 の場合のカウンタのタイミング図

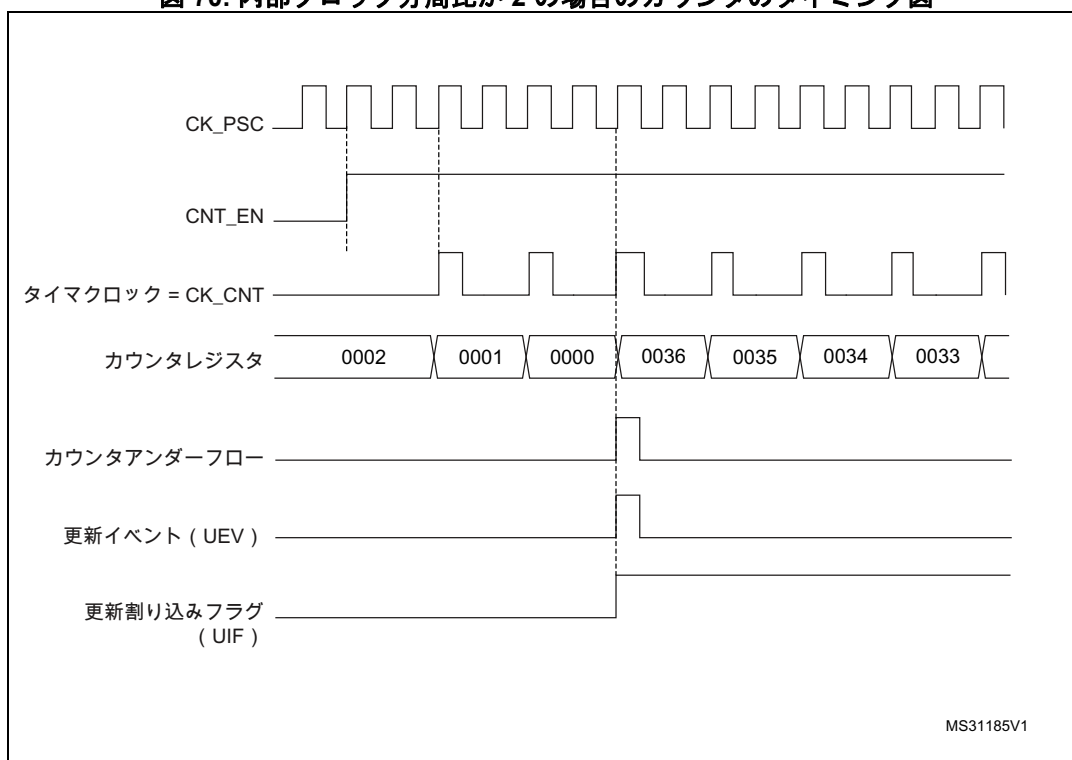
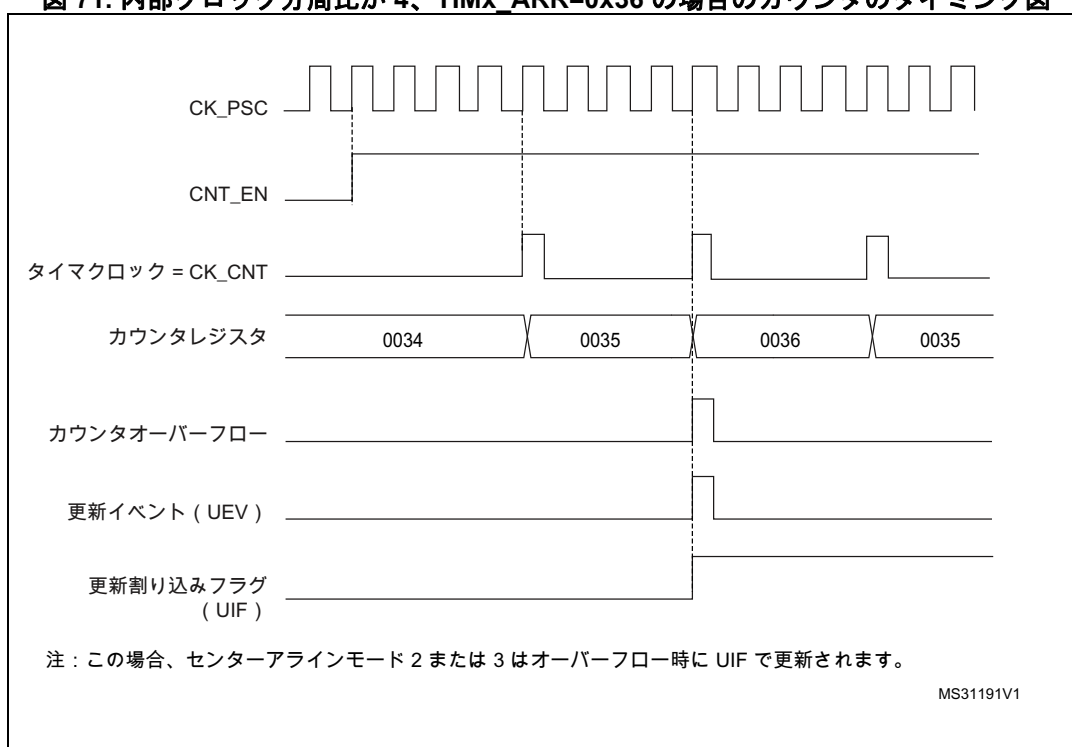


図 71. 内部クロック分周比が 4、TIMx_ARR=0x36 の場合のカウンタのタイミング図



1. センターアラインモード 2 または 3 が使用され、オーバーフロー時に UIF がセットされます。

図 72. 内部クロック分周比が N の場合のカウンタのタイミング図

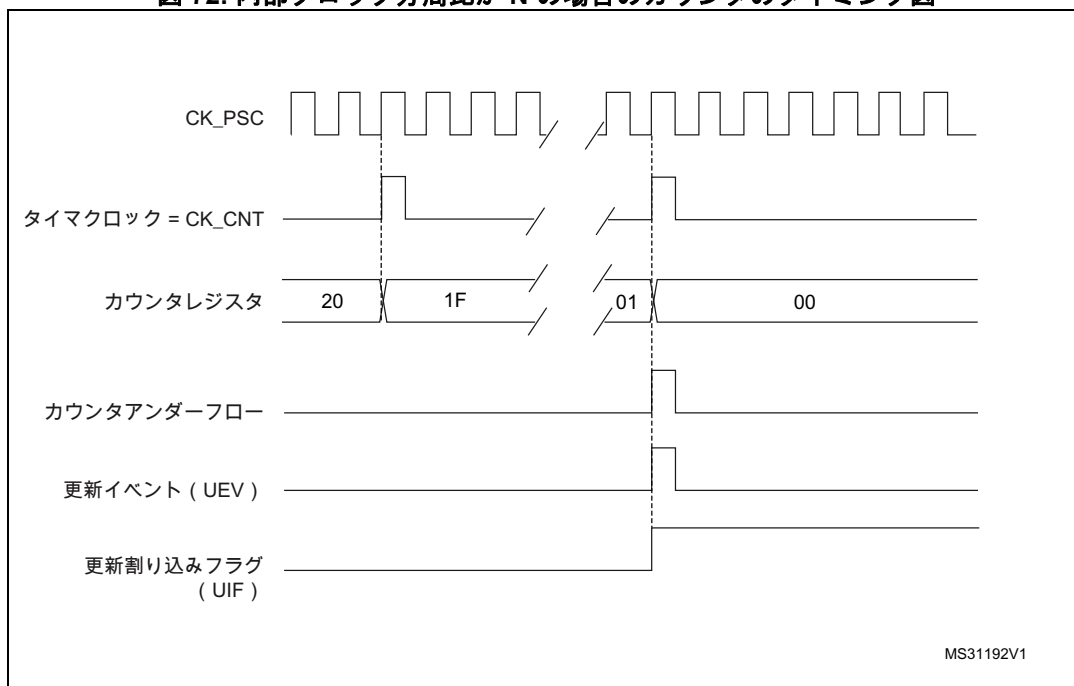


図 73. ARPE=1 (カウンタアンダーフロー) の場合の更新イベント時のカウンタのタイミング図

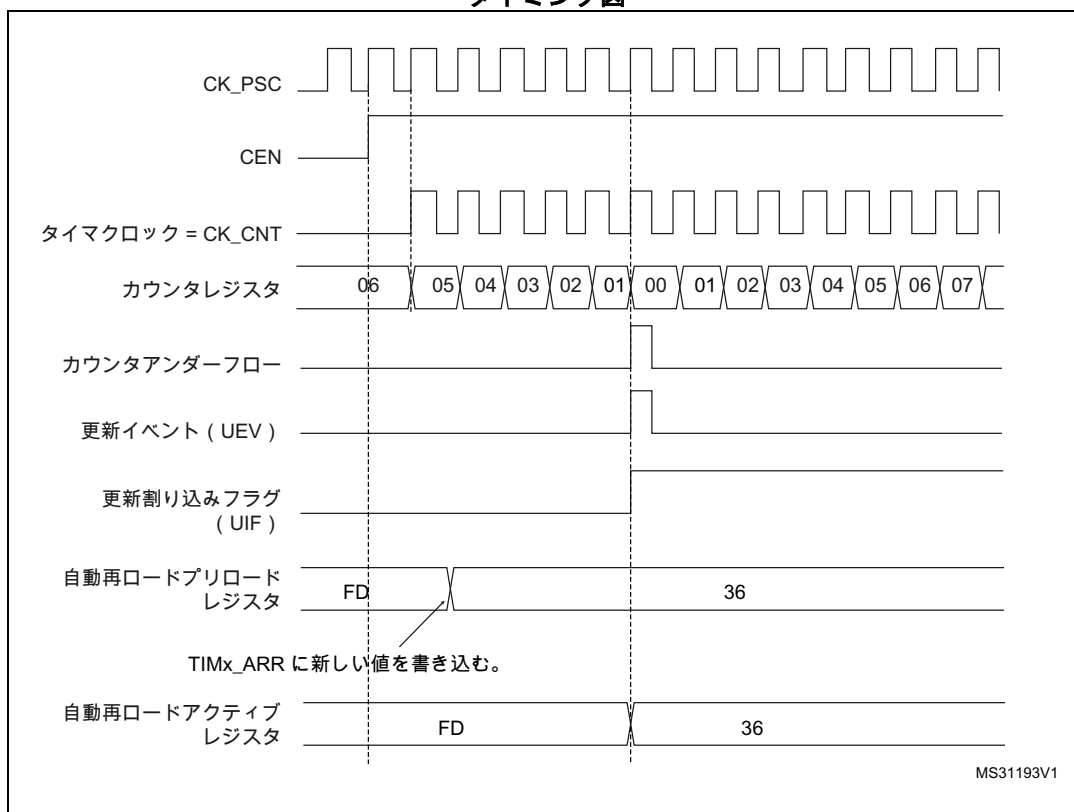
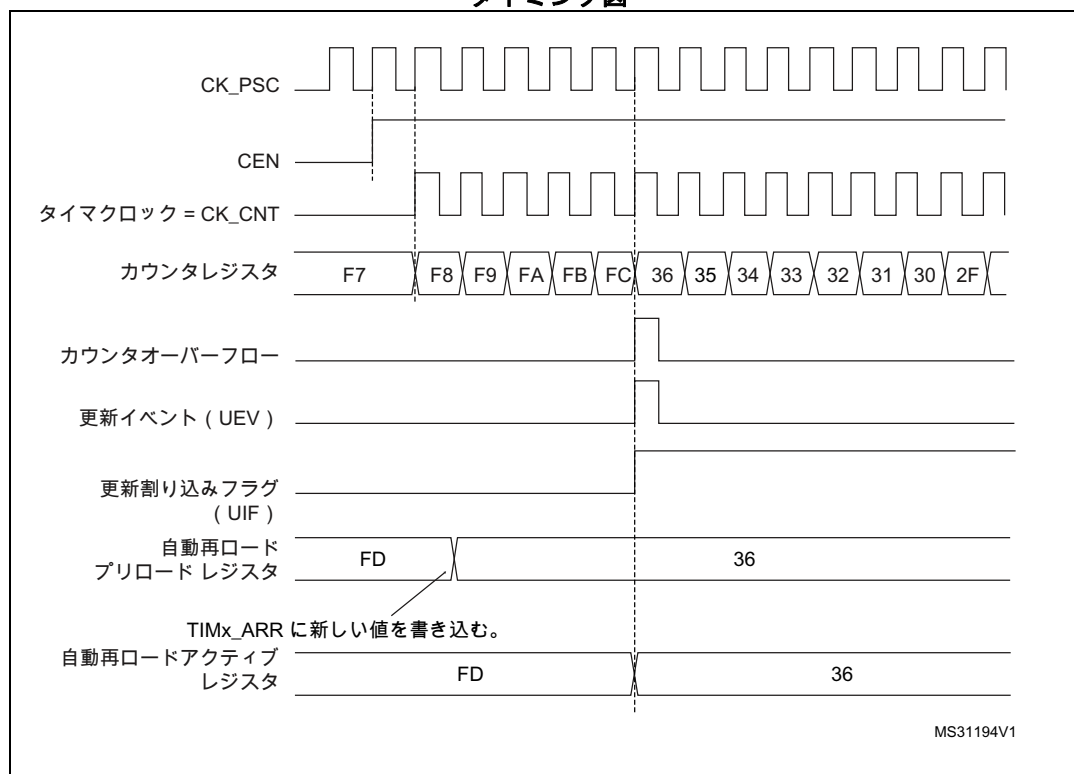


図 74. ARPE=1 (カウンタオーバーフロー) の場合の更新イベント時のカウンタのタイミング図



16.3.3 繰り返しカウンタ

セクション 16.3.1: タイムベースユニットに、カウンタオーバーフロー/アンダーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰り返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx_RCR 繰り返しカウンタレジスタの値を N とすると、N 回目のカウンタオーバーフローまたはアンダーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx_ARR 自動再ロードレジスタ、TIMx_PSC プリスケールレジスタ、比較モードの TIMx_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

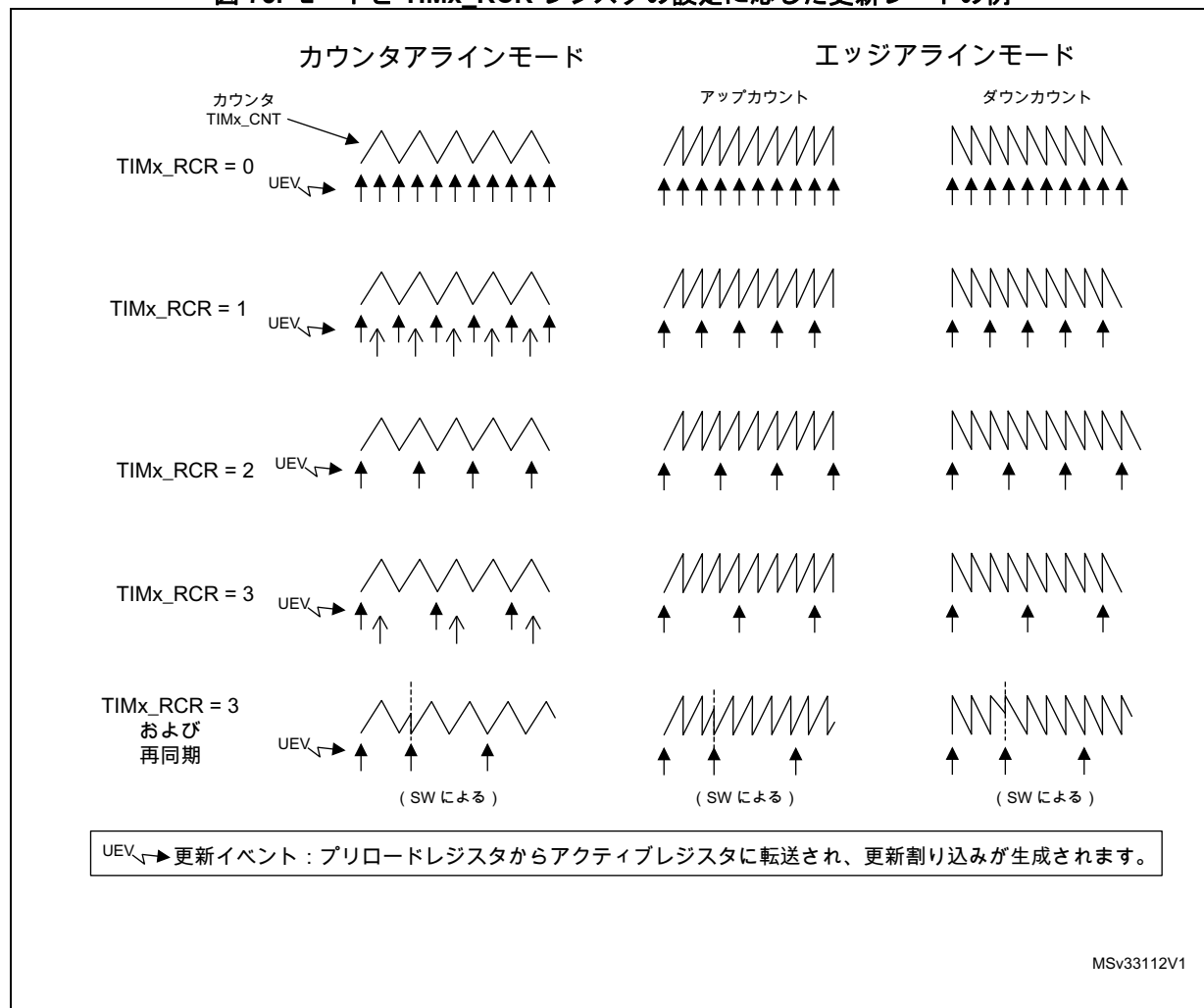
繰り返しカウンタは、次の場合にデクリメントします。

- アップカウントモードで、カウンタオーバーフローごと
- ダウンカウントモードで、カウンタアンダーフローごと
- センターアラインモードで、カウンタオーバーフローとカウンタアンダーフローごと最大繰り返し回数は 128 PWM サイクルに限られますが、PWM 周期ごとにデューティサイクルを 2 回更新することが可能になります。センタアラインモードで比較レジスタの値を PWM 周期あたり 1 回のみ更新するときには、パターンが対称なので、最大精度は $2 \times T_{ck}$ です。

繰り返しダウンカウンタは自動再ロードタイプです。繰り返しの回数は、TIMx_RCR レジスタの値によって定義されたとおりに維持されます (図 75 を参照してください)。ソフトウェアによって (TIMx_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰り返しカウンタの値にかかわらず直ちにイベントが発生し、繰り返しカウンタに TIMx_RCR レジスタの内容が再ロードされます。

センターアラインモードでは、TIMx_RCRが奇数の場合、TIMx_RCR レジスタが書き込まれたタイミングおよびカウンタが開始されたタイミングに応じてオーバーフローまたはアンダーフロー時に更新イベントが発生します。カウンタの開始前に TIMx_RCR が書き込まれた場合は、オーバーフローで、UEV が発生します。カウンタの開始後に TIMx_RCR が書き込まれた場合は、アンダーフローで UEV が発生します。たとえば、TIMx_RCR = 3 の場合、TIMx_RCR の書き込みタイミングに応じて4回目のオーバーフローイベントまたはアンダーフローイベントごとに UEV が発生します。

図 75. モードと TIMx_RCR レジスタの設定に応じた更新レートの例



16.3.4 クロックソース

カウンタクロックは、次のクロックソースによって供給されます。

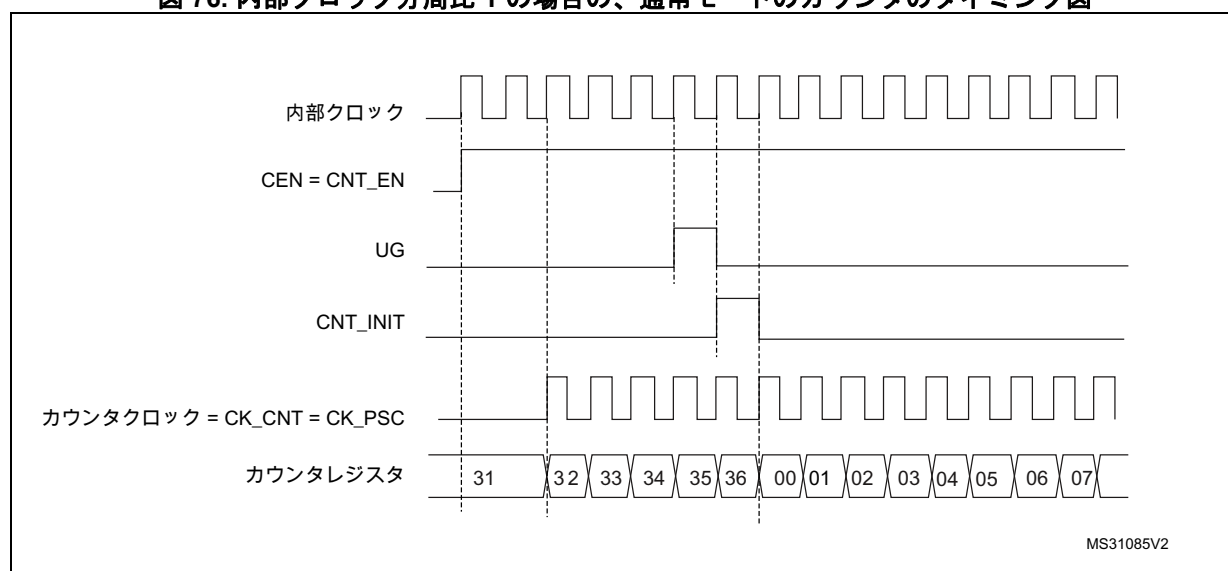
- 内部クロック (CK_INT)
- 外部クロックモード 1: 外部入力ピン
- 外部クロックモード 2: 外部トリガ入力 ETR
- 内部トリガ入力 (ITRx): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 1 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する \(414 ページ\)](#) を参照してください。

内部クロックソース (CK_INT)

スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx_CR1 レジスタ)、および UG ビット (TIMx_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 76](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

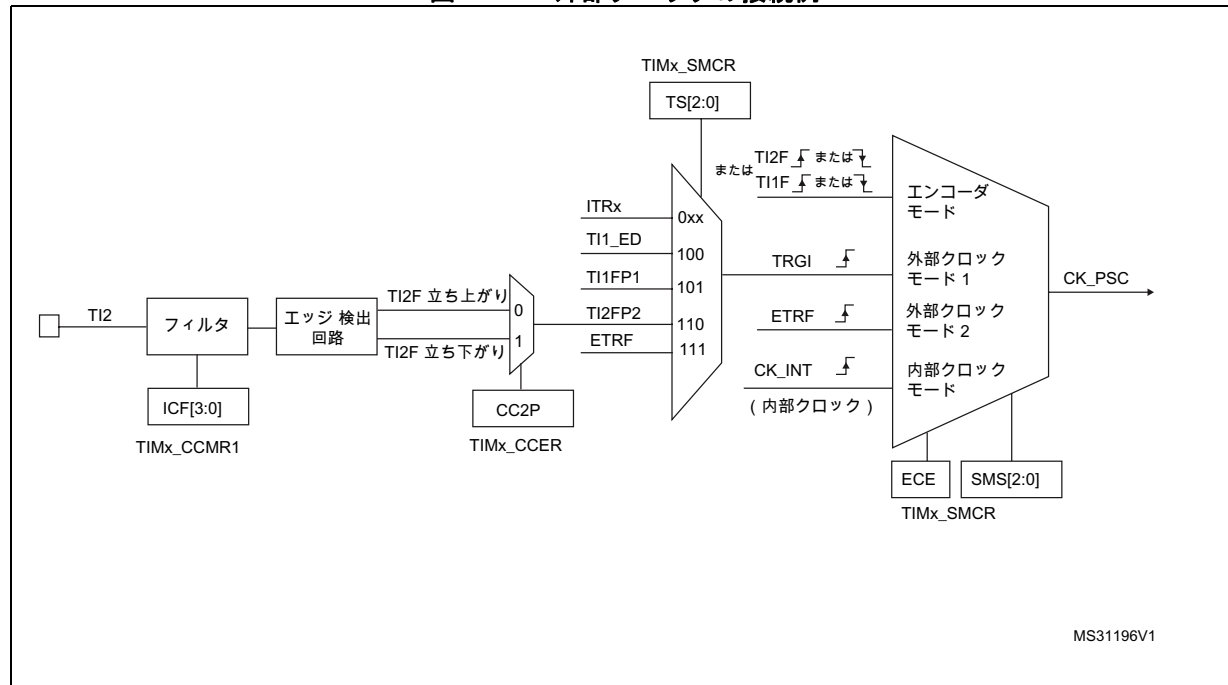
図 76. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 77. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_CCMR1 レジスタの CC2S ビットに"01"を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。
3. CC2P=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

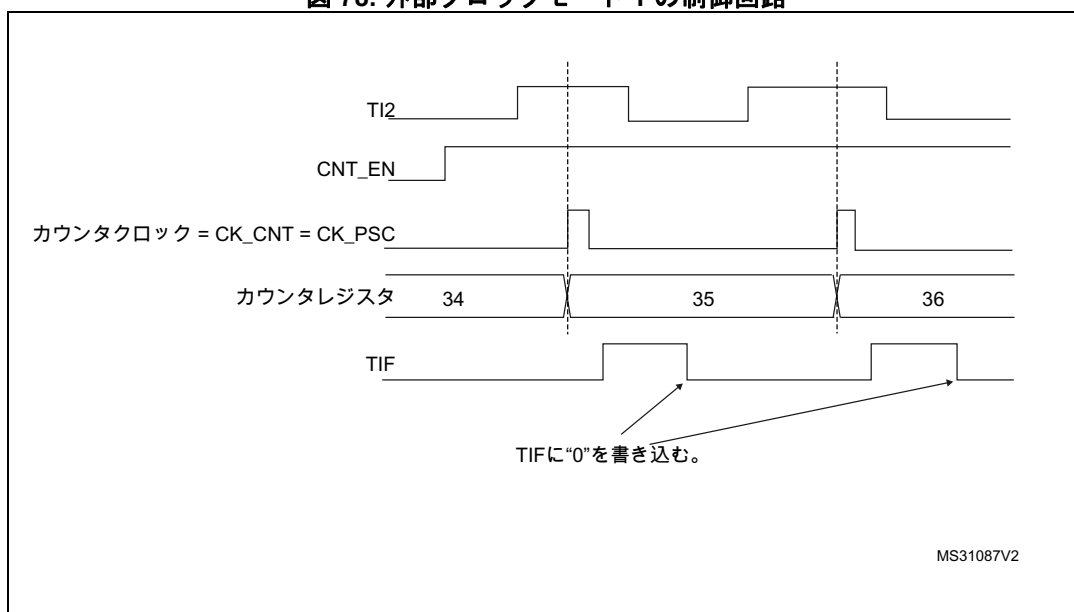
注： キャプチャプリスケアラはトリガには使用されないので、設定は不要です。

コード例については、「付録」セクション [A.9.1 : TI2 の立ち上がりエッジのアップカウンタのコード例](#)を参照してください。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウントクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 78. 外部クロックモード 1 の制御回路



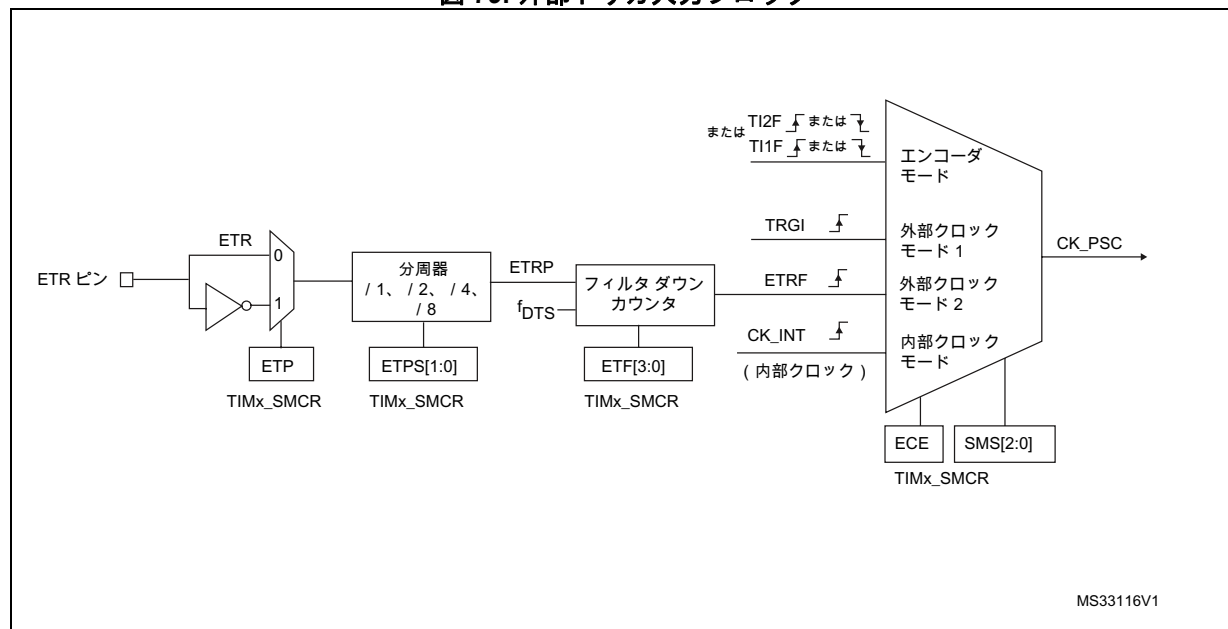
外部クロックソースモード 2

このモードは、TIMx_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 79 に、外部トリガ入力ブロックの概要を示します。

図 79. 外部トリガ入力ブロック



たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

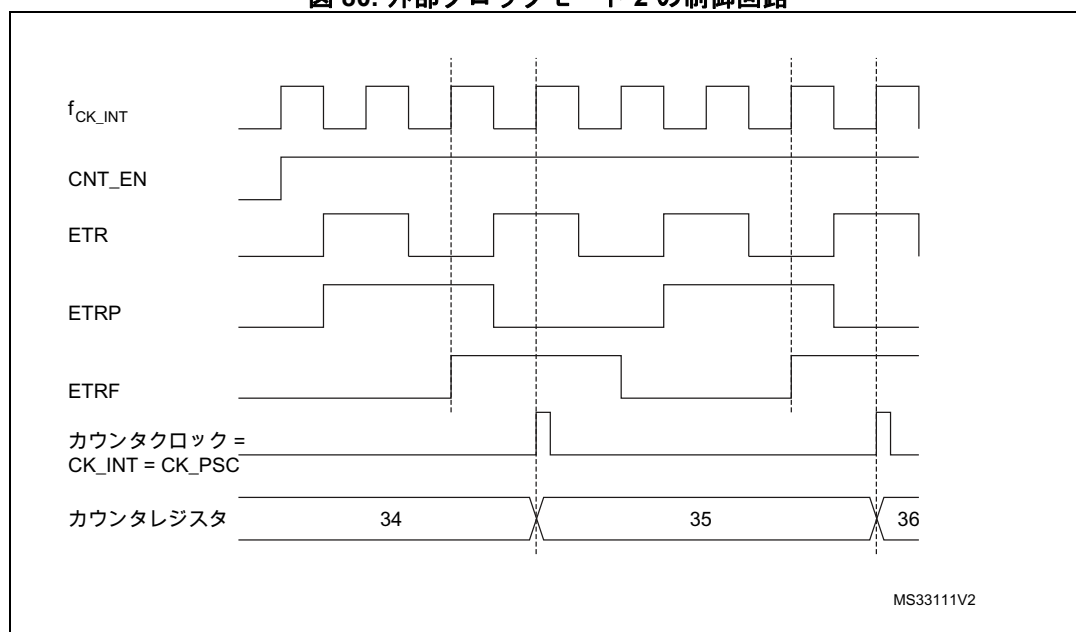
1. この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

コード例については、「付録」セクション [A.9.2 : 2 回の ETR の立ち上がりエッジごとのアップカウンタのコード例](#) を参照してください。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 80. 外部クロックモード 2 の制御回路



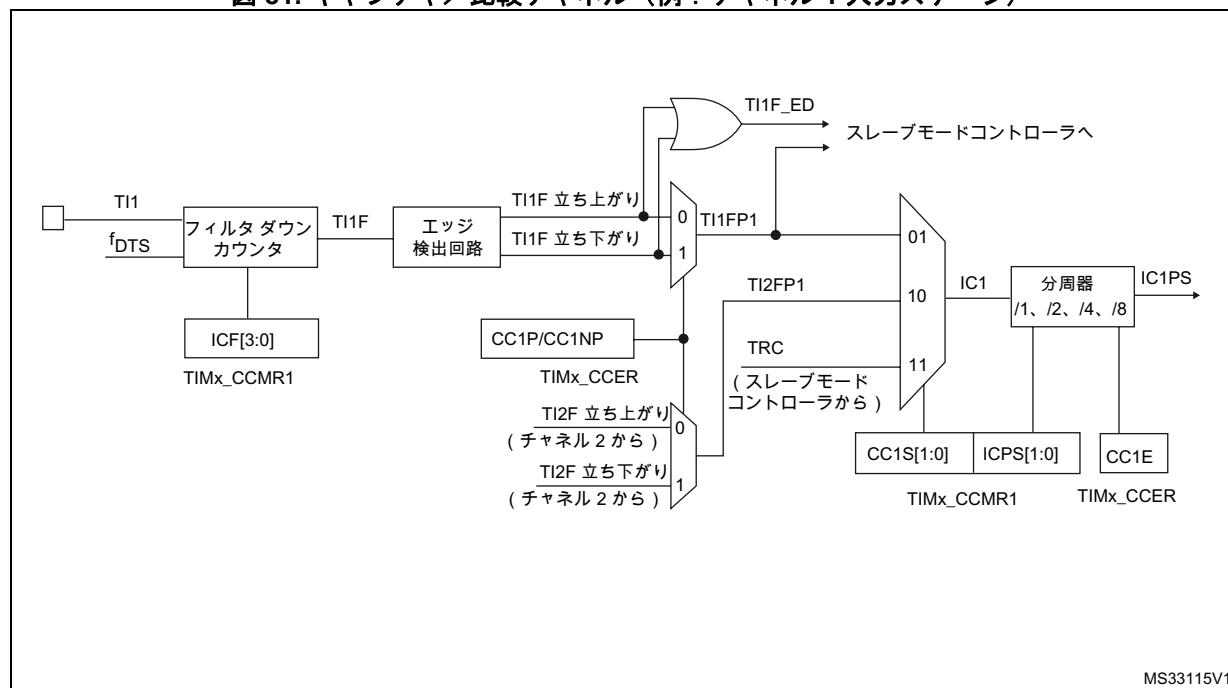
16.3.5 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、プリスケアラ）、および出力カステージ（比較回路と出力制御）から構成されています。

図 81 から 図 84 に、1 つのキャプチャ／比較チャネルの概要を示します。

入カステージは、対応する TIX 入力をサンプリングして、フィルタリングを行った TIXF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号（TIXFPx）を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ（ICxPS）に渡されます。

図 81. キャプチャ／比較チャネル（例：チャネル 1 入カステージ）



出力ステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 82. キャプチャ／比較チャネル 1 メイン回路

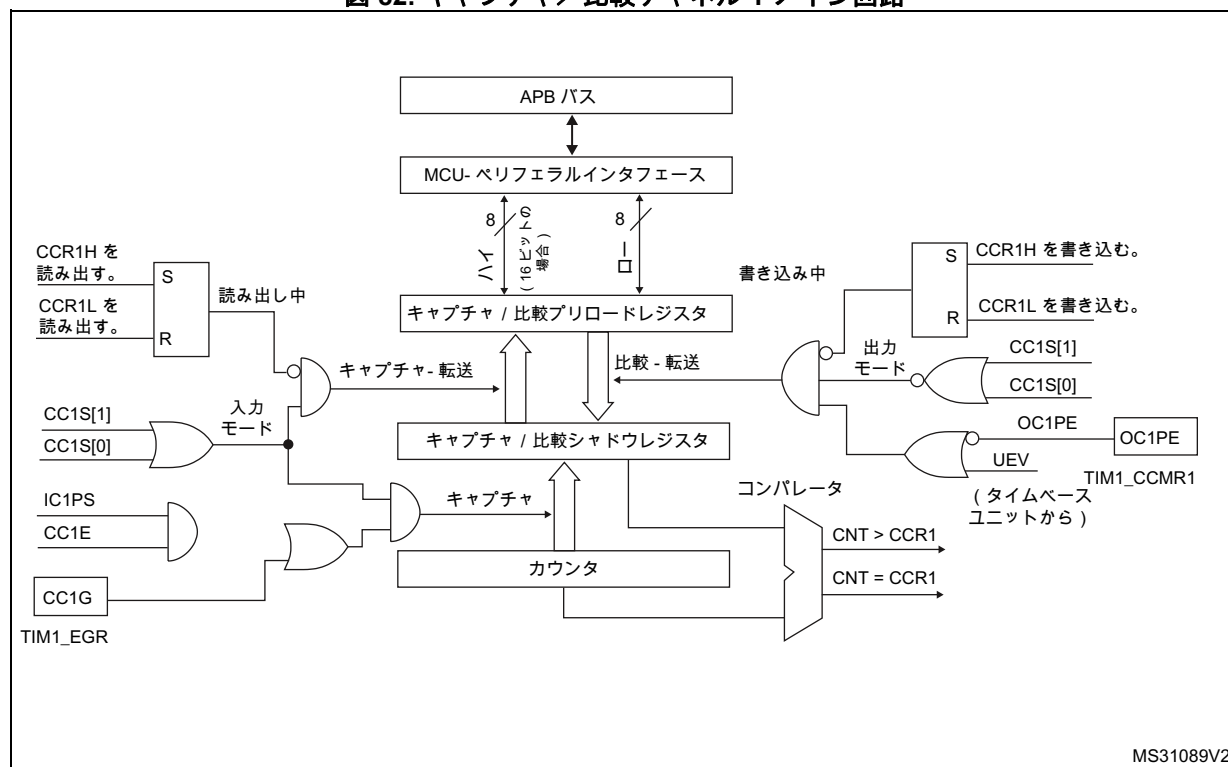
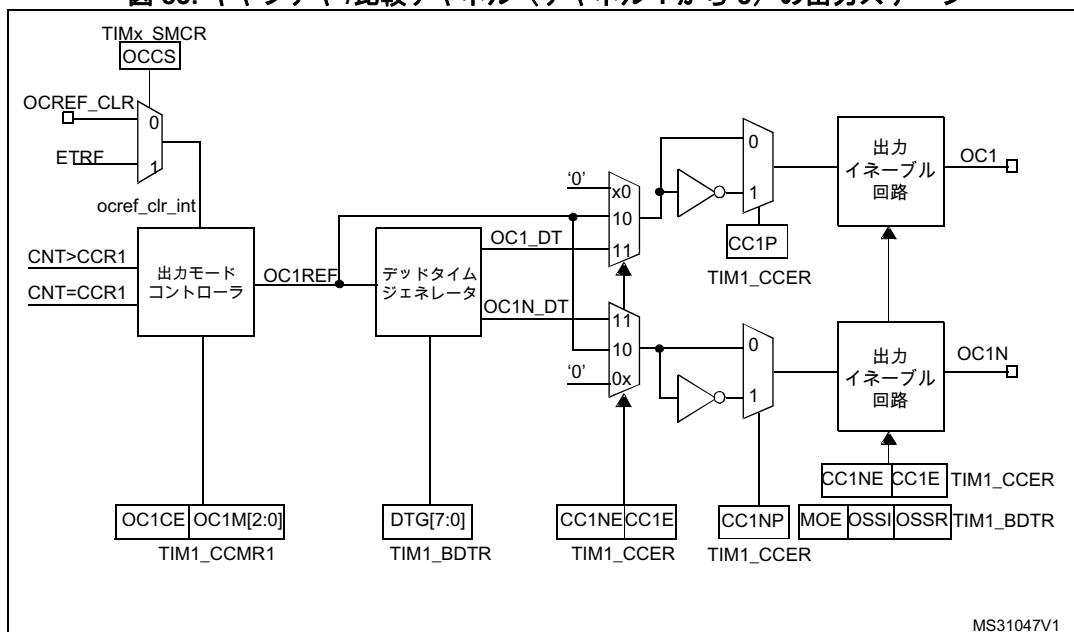
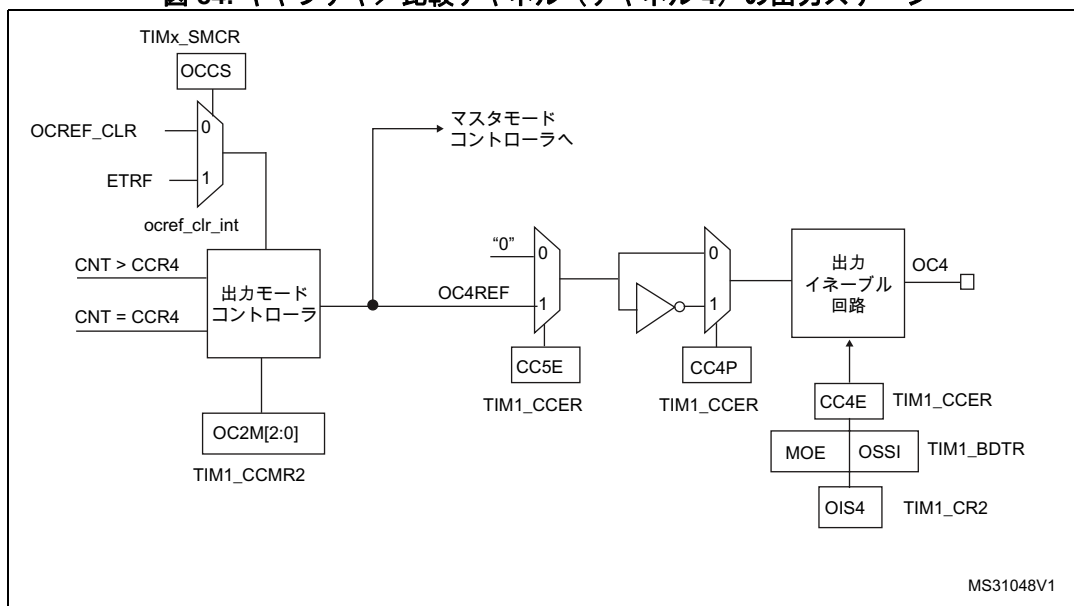


図 83. キャプチャ/比較チャンネル (チャンネル 1 から 3) の出力ステージ



MS31047V1

図 84. キャプチャ/比較チャンネル (チャンネル 4) の出力ステージ



MS31048V1

キャプチャ/比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

16.3.6 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならないので、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx_CCMR1 レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ変化のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“0”を書き込みます (この場合、立ち上がりエッジの選択)。
- 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要の場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

コード例については、「付録」セクション [A.9.3: 入力キャプチャ設定のコード例](#) を参照してください。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

コード例については、「付録」セクション [A.9.4: 入力キャプチャのデータ管理のコード例](#) を参照してください。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注: IC 割り込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

16.3.7 PWM 入力モード

このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

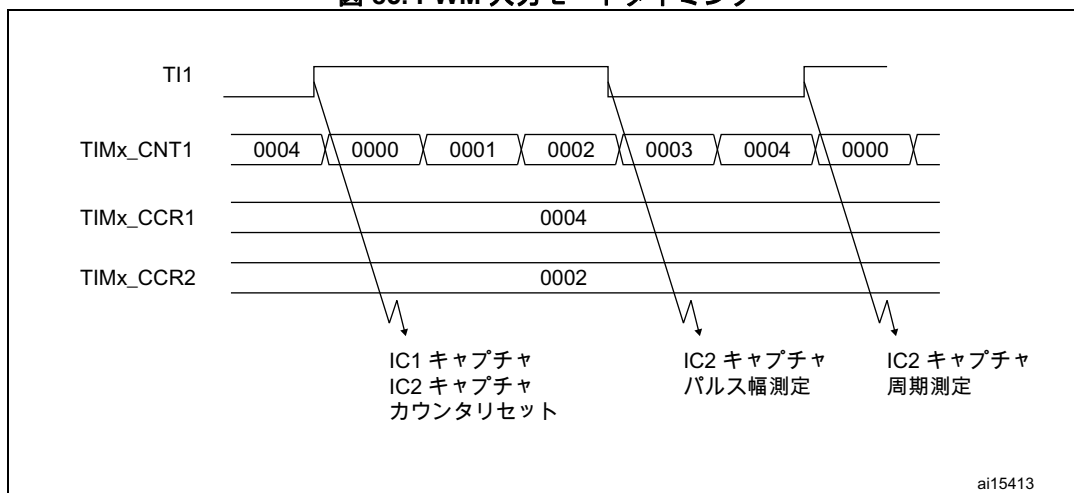
- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

- TIMx_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブな極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用)。
- TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
- CC2P ビットに“1”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

コード例については、「付録」セクション [A.9.5 : PWM 入力設定のコード例](#) を参照してください。

図 85. PWM 入力モードタイミング



16.3.8 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルとするには、対応する TIMx_OCMRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

16.3.9 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

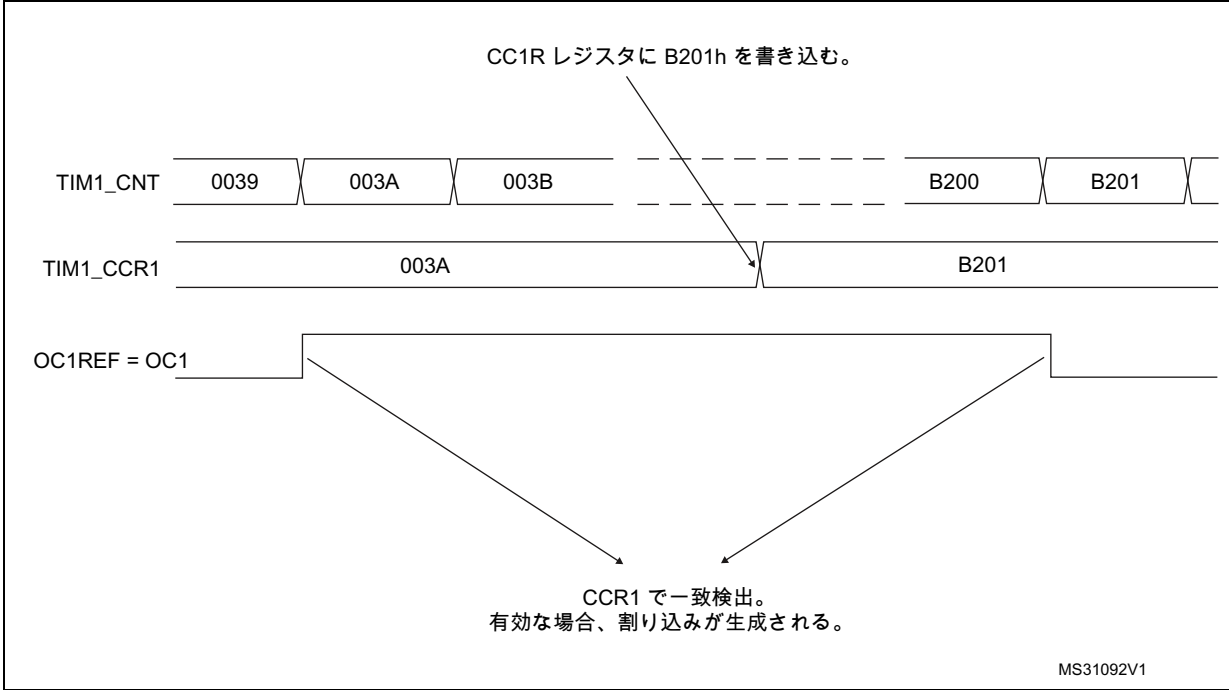
- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは、単一パルスを出力するためにも使用できます (ワンパルスモード)。

- 手順：
1. カウンタクロックを選択します（内部、外部、プリスケアラ）。
 2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
 3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
 4. 出力モードを選択します。例：
 - － CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに 011 を書き込みます。
 - － プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
 - － アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
 - － 出力を有効にするには、CCxE ビットに 1 を書き込みます。
 5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。
- コード例については、「付録」セクション [A.9.7：入力比較設定のコード例](#) を参照してください。
- いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り（OCxPE=0）。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 86](#) に示します。

図 86. 出力比較モード、OC1 の反転



16.3.10 PWM モード

パルス幅変調(PWM)モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx_CCER および TIMx_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx $\leq$ TIMx_CNT または TIMx_CNT $\leq$ TIMx_CCRx かどうか判断されます (カウントの方向によります)。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

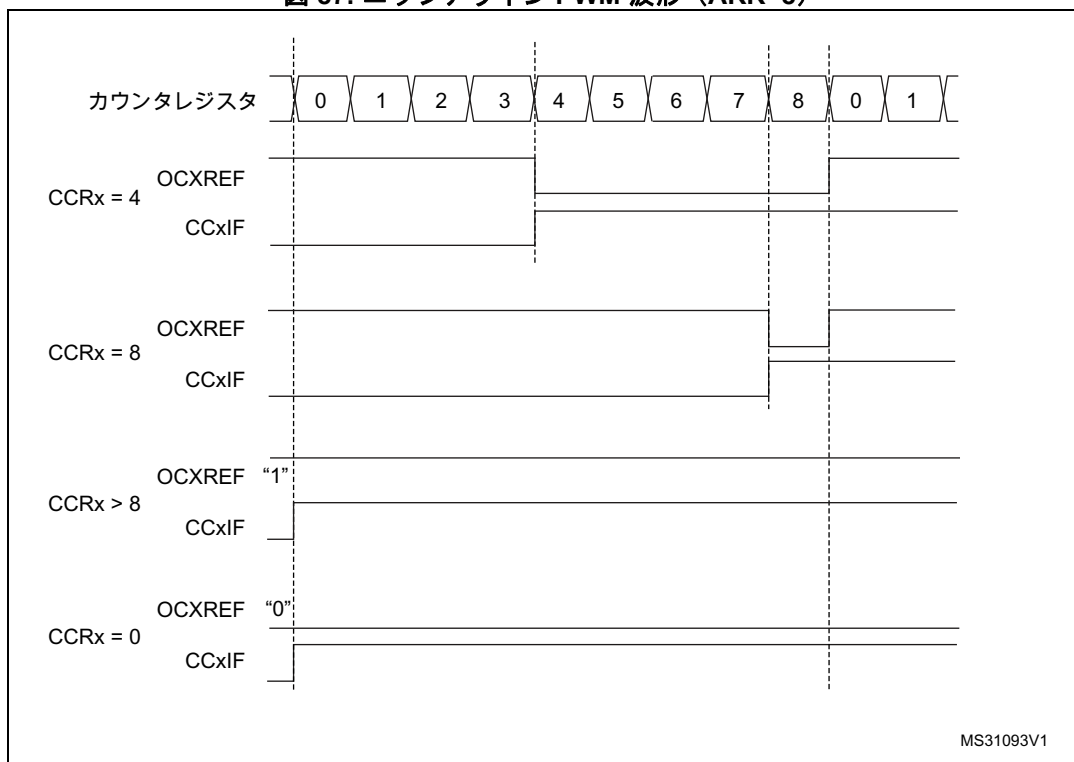
PWM エッジアラインモード

- アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。アップカウントモード (308 ページ) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。図 87 に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 87. エッジライン PWM 波形 (ARR=8)



コード例については、「付録」セクション [A.9.8: エッジライン PWM の設定例](#) を参照してください。

- ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。 [ダウンカウントモード \(312 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

PWM センターアラインモード

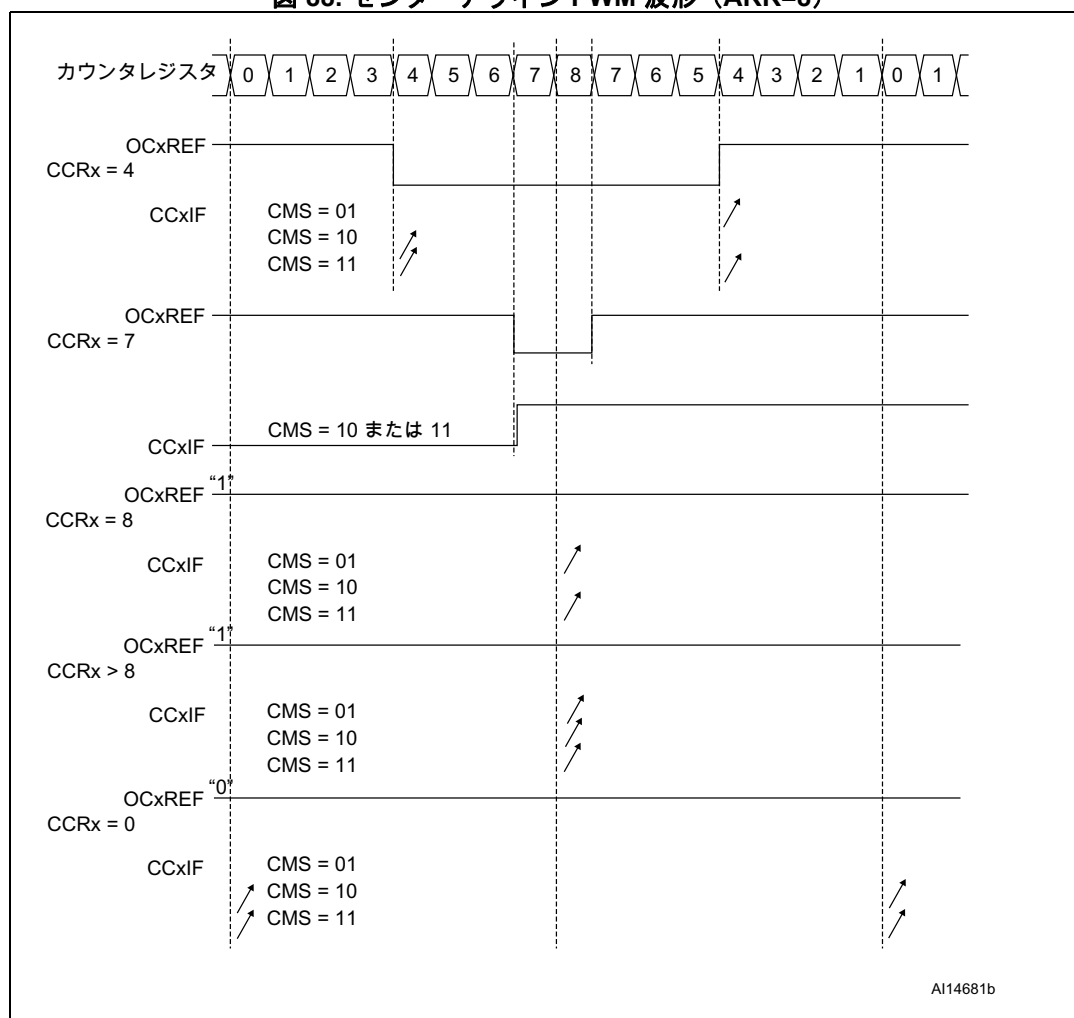
センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです（その他すべての構成は、OCxRef/OCx 信号に対して同じ効果を持ちます）。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット（DIR）はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。[センターアラインモード（アップ/ダウンカウント）（314 ページ）](#)を参照してください。

図 88 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

コード例については、「付録」セクション [A.9.9：センターアライン PWM の設定例](#) を参照してください。

図 88. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント：

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
 - － 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx_CNT > TIMx_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
 - － カウンタに 0 または TIMx_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

16.3.11 相補出力とデッドタイム挿入

高機能制御タイマ (TIM1) は、2 つの相補信号を出力して、出力時のスイッチオンおよびスイッチオフを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性 (レベルシフタの内在的な遅延、電源スイッチによる遅延など) に応じて調整する必要があります。

出力の極性 (主出力 OCx または補 OCxN) は、出力ごとに独自に選択できます。これは TIMx_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 OCx および OCxN は、TIMx_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx_BDTR レジスタと TIMx_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、[表 55: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能 \(367 ページ\)](#) を参照してください。特に、IDLE 状態に切り替わるとき (MOE が 0 になるときに)、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッドタイムジェネレータがあります。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点が異なります。
- OCxN 出力信号は、立ち上がりエッジが基準波形の立ち下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (OCx または OCxN) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 OCxREF との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 89. デッドタイム挿入のある相補出力

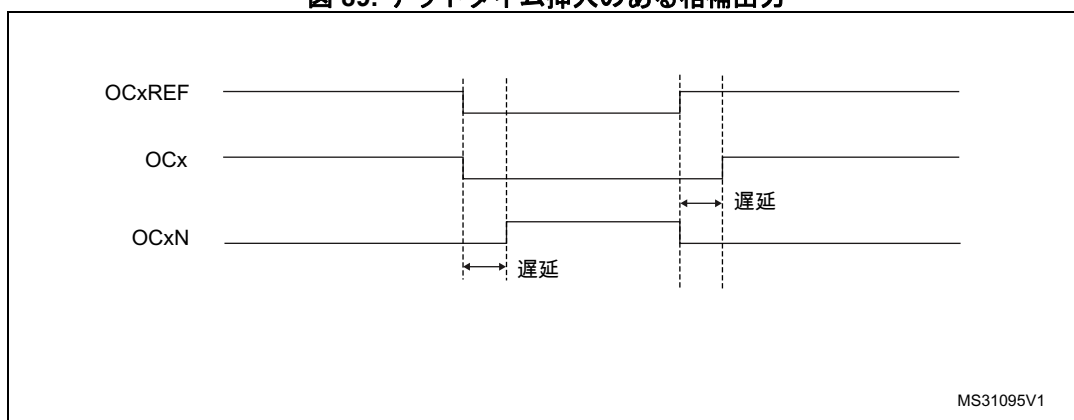


図 90. 負のパルスより長い遅延があるときのデッドタイムの波形

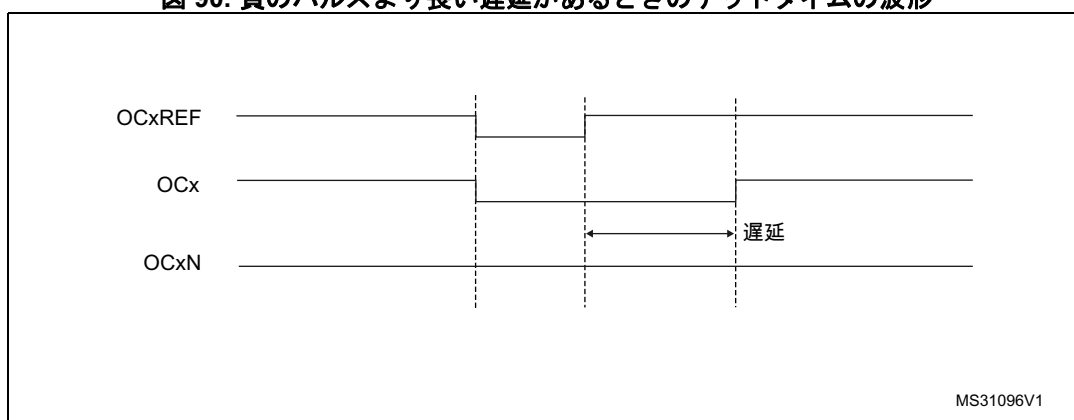
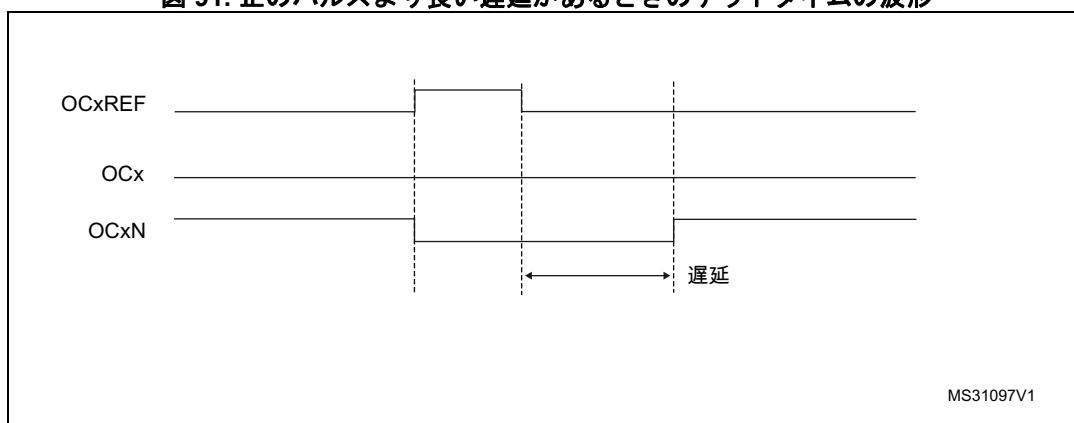


図 91. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャネルで同じであり、TIMx_BDTR レジスタの DTG ビットでプログラミングできます。遅延計算については、[セクション 16.4.18 : TIM1 ブレークおよびデッドタイムレジスタ \(TIM1_BDTR\) \(371 ページ\)](#) を参照してください。

OCxREF 信号の OCx または OCxN へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx_CCER レジスタの CCxE ビットおよび CCxNE ビットを構成することによって、OCxREF 信号を OCx 出力または OCxN 出力にリダイレクトできます。

これにより、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定することができます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができま

注： OCxN のみが有効なときには (CCxE=0, CCxNE=1)、相補にならず、OCxREF がハイレベルとなるとアクティブになります。たとえば、CCxNP=0 の場合は、OCxN=OCxRef です。他方、OCx と OCxN の両方が有効なときには (CCxE=CCxNE=1)、OCxREF がハイになると OCx はアクティブになり、OCxREF がローのときには、OCxN は補信号であり、アクティブになります。

16.3.12 ブレーク機能の使用

ブレーク機能を使用しているときには、出力イネーブル信号とインアクティブレベルは追加の制御ビット (TIMx_BDTR レジスタの MOE、OSSI、および OSSR ビットと TIMx_CR2 レジスタの OISx および OISxN ビット) に応じて変更されます。ただし、OCx および OCxN 出力の両方を同時にアクティブレベルに設定することはできません。詳細については、[表 55: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能 \(367 ページ\)](#) を参照してください。

ブレークソースは、ブレーク入力ピンか、リセットクロックコントローラ (RCC) のクロックセキュリティシステム (CSS) によって生成されたクロック障害イベントです。クロックセキュリティシステムの詳細については、[セクション 7.2.8: クロックセキュリティシステム \(CSS\)](#) を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。TIMx_BDTR レジスタの BKE ビットをセットすることによって、ブレーク機能を有効にできます。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。BKE および BKP ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がりがエッジは非同期のことがあるので、実際の信号（出力に作用する信号）と同期制御ビット (TIMx_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延（ダミー命令）を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

ブレークが発生すると（ブレーク入力で選択されたレベル）、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態、アイドル状態、またはリセット状態 (OSSI ビットで選択) になります。これは、MCU オシレータがオフの場合も同様です。
- 各出力チャネルは、MOE=0 になったとき、TIMx_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はハイのままです。

- 相補出力が使用されているときには：
 - － 出力は、まずリセット状態のインアクティブ状態に置かれます（極性に依存します）。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
 - － タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください（約 2 CK_TIM クロックサイクル）。
 - － OSSR=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はそのままか、CCxE または CCxNE ビットが高くなったときにハイになります。
- ブレーク状態フラグ (TIMx_SR レジスタの BIF ビット) がセットされます。TIMx_DIER レジスタの BIE ビットがセットされている場合は、割り込みを生成できます。TIMx_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント UEV で再び自動的にセットされます。これを使用して、たとえば、レギュレーションを行うことができます。そうでない場合、MOE ビットは、再び 1 が書き込まれるまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

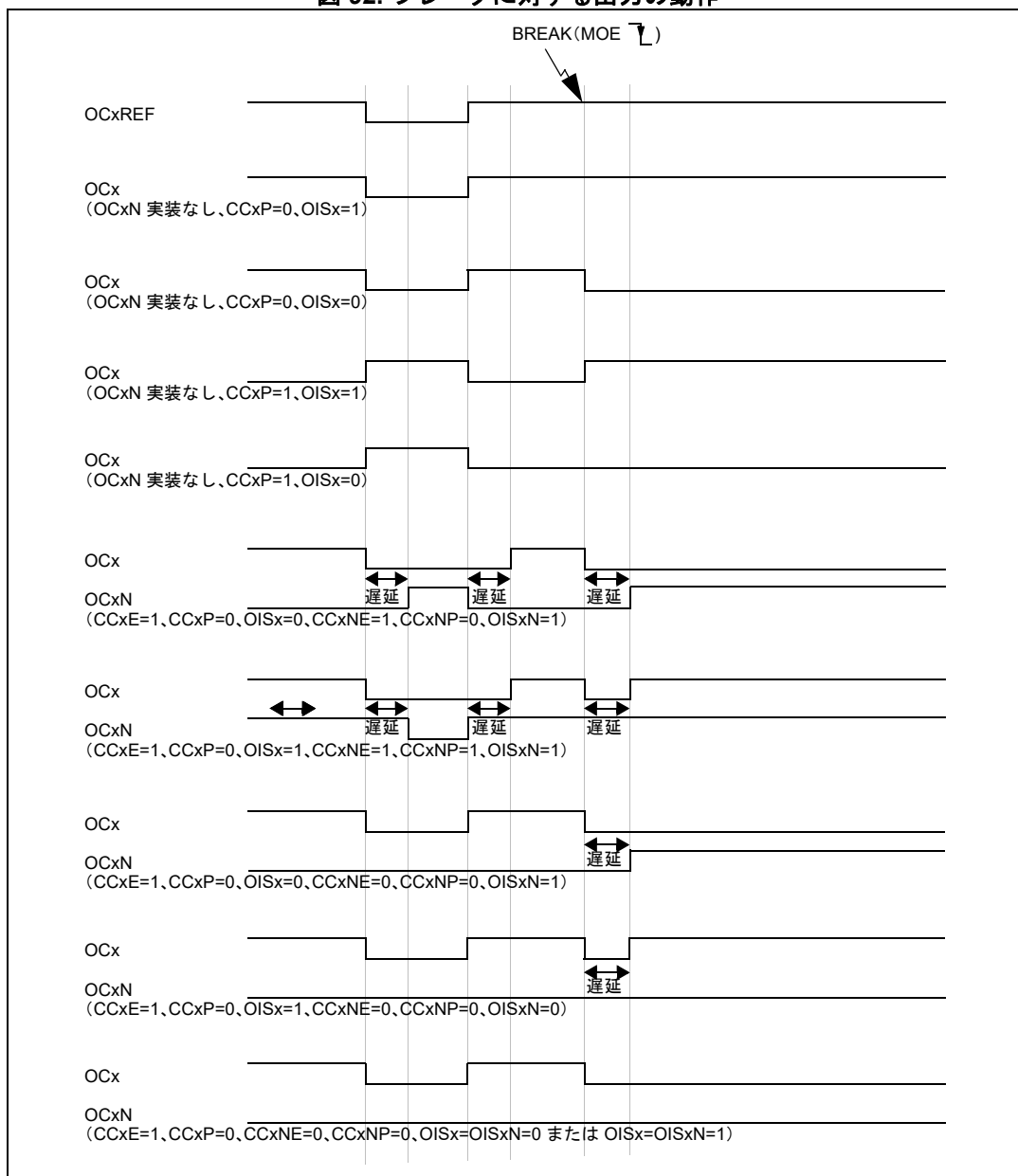
注： **ブレーク入力は、信号レベルに対して動作します。このため、ブレーク入力がアクティブな間は、MOE をセットできません（自動的にも、ソフトウェアによっても）。この間、ステータスフラグ BIF をクリアできません。**

ブレークは、BRK 入力によって生成でき、BRK はプログラミング可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ（デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性）を固定することができます。TIMx_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 16.4.18：TIM1 ブレークおよびデッドタイムレジスタ \(TIM1_BDTR\) \(371 ページ\)](#) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。

[図 92](#) に、ブレークに対する出力の動作例を示します。

図 92. ブレークに対する出力の動作



16.3.13 外部イベントによる OCxREF 信号のクリア

特定のチャンネルの OCxREF 信号は OCREF_CLR_INPUT にハイレベルを適用するとクリアされます (対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする)。OCxREF は、次の更新イベント (UEV) が発生するまで、ローレベルを保ちます。この機能は、出力比較モードと PWM モードでのみ使用可能です。強制モードでは動作しません。

OCREF_CLR_INPUT は、TIMx_SMCR レジスタで OCCS ビットを設定することで、OCREF_CLR 入力と ETRF (フィルタ後の ETR) の間で選択できます。

ETRF が選択された場合、ETRF は次のように設定する必要があります。

特定のチャンネルの OCxREF 信号は、ETRF 入力にハイレベルを適用する (対応する TIMx_CCMRx レジスタの OCxCE イネーブルビットを“1”にセットする) ことによってローに駆動できます。OCxREF 信号は、次の更新イベント UEV が発生するまで、ローレベルを保ちます。

この機能は、出力比較モードと PWM モードでのみ使用でき、強制モードでは機能しません。

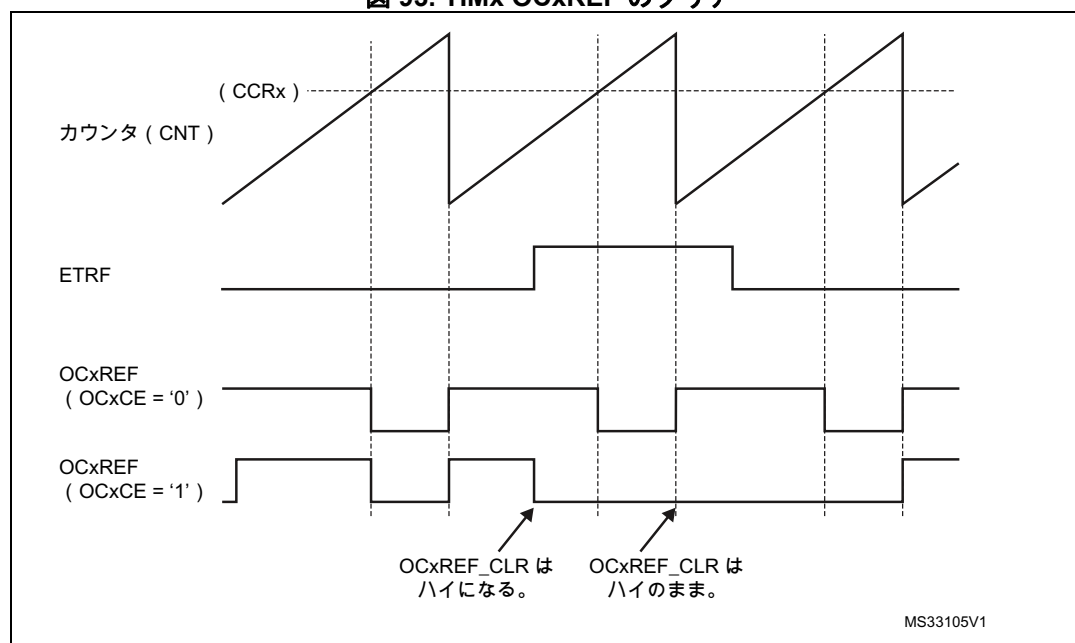
たとえば、OCxREF 信号を比較回路の出力に接続して、電流処理に使用することができます。この場合、ETRF は次のように設定する必要があります。

1. 外部トリガプリスケアラはオフ状態に維持します (TIMx_SMCR レジスタの ETPS[1:0] ビットを“00”にセット)。
2. 外部クロックモード 2 を無効にします (TIMx_SMCR レジスタの ECE ビットを“0”にセット)。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、ユーザのニーズに応じて設定できます。

コード例については、「付録」セクション [A.9.10 : OCxREF をクリアするための ETRF 設定のコード例](#) を参照してください。

[図 93](#) に、イネーブルビット OCxCE の両方の値について、ETRF 入力がハイレベルになったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 93. TIMx OCxREF のクリア



注 : 100% デューティサイクルの PWM の場合 (CCR_x > ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

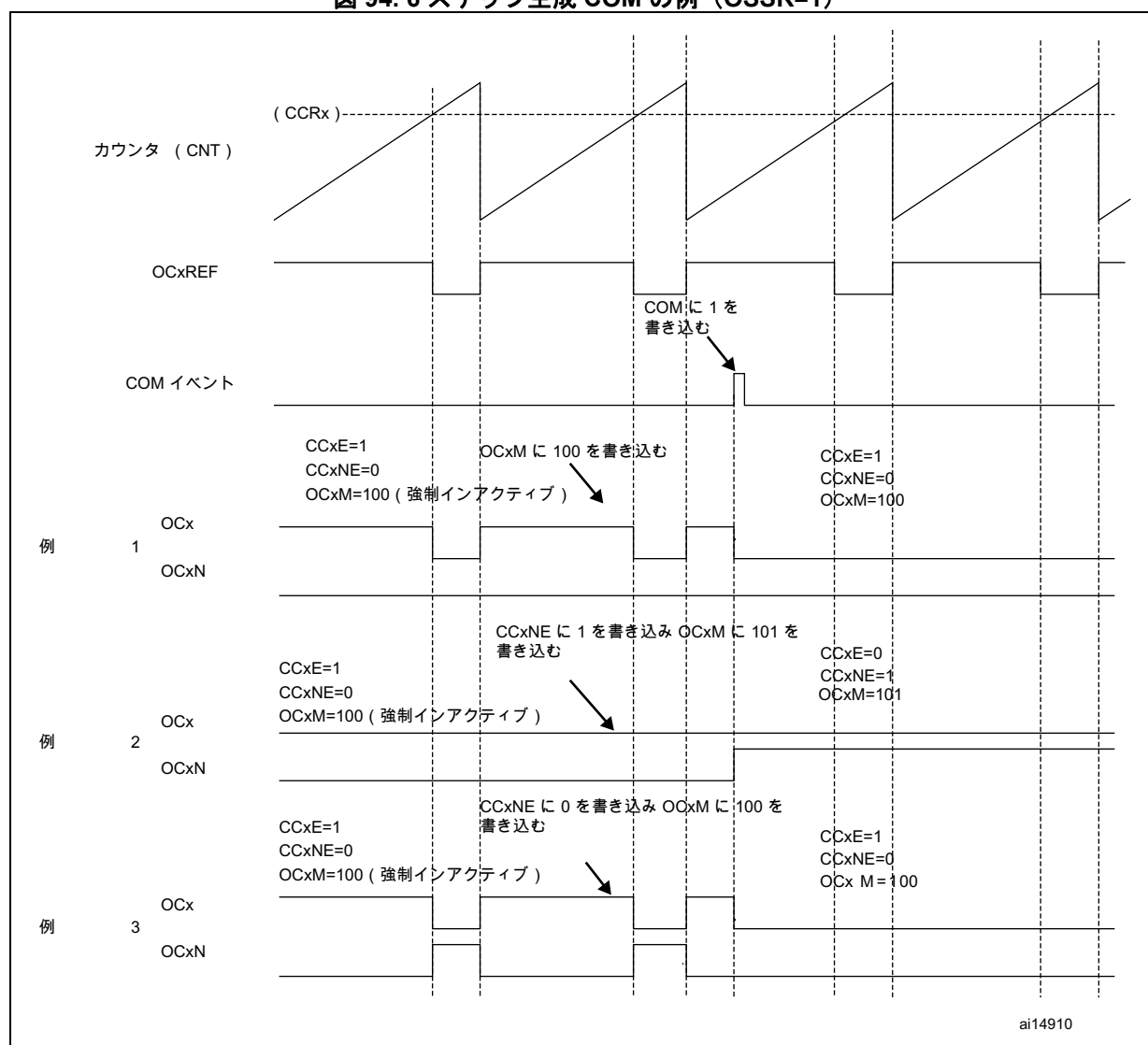
16.3.14 6 ステップ PWM 生成

チャンネルで相補出力が使用されているときには、OCxM、CCxE、および CCxNE ビットでプリロードビットが使用できます。プリロードビットは、COM 転換イベントでシャドウビットにコピーされます。これにより、次のステップの構成をあらかじめプログラミングして、すべてのチャンネルの構成を同時に変更することができます。COM は、TIMx_EGR レジスタの COM ビットをセットすることによってソフトウェアによって、またはハードウェアによって (TRGI 立ち上がりエッジで) 生成することができます。

フラグは、COM イベントが発生したときにセットされ (TIMx_SR レジスタの COMIF ビット)、これによって割り込み (TIMx_DIER レジスタの COMIE ビットがセットされている場合) または DMA リクエスト (TIMx_DIER レジスタの COMDE ビットがセットされている場合) を生成できます。

図 94 に、COM イベントが発生したときの OCx と OCxN 出力の動作を、3 種類のプログラミング構成の例で示します。

図 94. 6 ステップ生成 COM の例 (OSSR=1)



16.3.15 ワンパルスモード

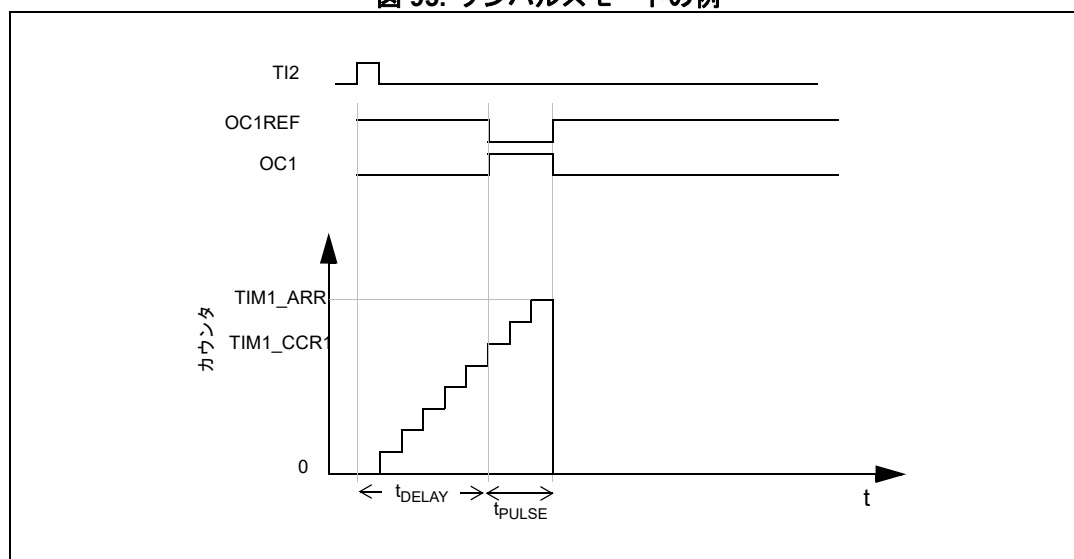
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に $0 < CCRx$ の場合)
- ダウンカウント時 : $CNT > CCRx$

図 95. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで CC2P=“0”と CC2NP=“0”を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx_SMCR レジスタの TS ビットに“110”を書き込みます。
- TI2FP2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに“110” (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1+1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

コード例については、「付録」セクション [A.9.16: ワンパルスモードのコード例](#) を参照してください。

必要なパルスは 1 つだけなので (シングルモード)、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、“0”に戻る時点)。TIMx_CR1 レジスタの OPM ビットが“0”にセットされると、繰り返しモードが選択されます。

特殊なケース：OCx 高速イネーブル

ワンパルスモードでは、TIx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

コード例については、「付録」セクション [A.9.16: ワンパルスモードのコード例](#) に記載された PULSE_WITHOUT_DELAY > 0 によって条件付けられたコードの一部を参照してください。

16.3.16 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS=“001”を、TI1 エッジのみをカウントしている場合は SMS=“010”を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS=“011”を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。必要なときには、入力フィルタもプログラミングできます。CC1NP と CC2NP はローに維持する必要があります。

2 つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。[表 53](#)を参照してください。カウンタのクロック供給は、TI1FP1 または TI2FP2 (入力フィルタおよび極性選択の後) は TI1 と TI2、フィルタされず反転されない場合は TI1FP1=TI1、フィルタされず反転されない場合は TI2FP2=TI2) での有効な遷移ごとに行われます。ただし、カウンタは有効である (TIMx_CR1 レジスタの CEN ビットに“1”が書き込まれている) ことが前提です。2 つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力 (TI1 または TI2) の遷移のたびに計算されます。



エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします（方向に応じて、0 から ARR まで、または ARR から 0 まで）。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、繰り返しカウンタ、トリガ出力の機能は通常どおりに機能が続けます。エンコーダモードと外部クロックモード 2 は互換性がないので、同時に選択することはできません。

このモードでは、カウンタはインクリメンタルエンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します（TI1 と TI2 は同時に切り替わらないと想定しています）。

表 53. カウント方向とエンコーダ信号

アクティブ エッジ	他方の信号の レベル (TI2 に対 する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	高	ダウン	アップ	カウントなし	カウントなし
	低	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	高	カウントなし	カウントなし	アップ	ダウン
	低	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の両方 をカウント	高	ダウン	アップ	アップ	ダウン
	低	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

 96 に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S="01" (TIMx_CCMR1 レジスタ、TI1FP1 は TI1 に配置)
- CC2S="01" (TIMx_CCMR2 レジスタ、TI1FP2 は TI2 に配置)
- CC1P="0" (TIMx_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1)
- CC2P="0" (TIMx_CCER レジスタ、TI1FP2 非反転、TI1FP2=TI2)
- SMS="011" (TIMx_SMCR レジスタ、両方の入力立ち上がり立ち下がり両エッジでアクティブ)
- CEN="1" (TIMx_CR1 レジスタ、カウンタ有効)

コード例については、「付録」セクション [A.9.11 : エンコーダインタフェースのコード例](#) を参照してください。

図 96. エンコーダインタフェースモードにおけるカウンタの動作例

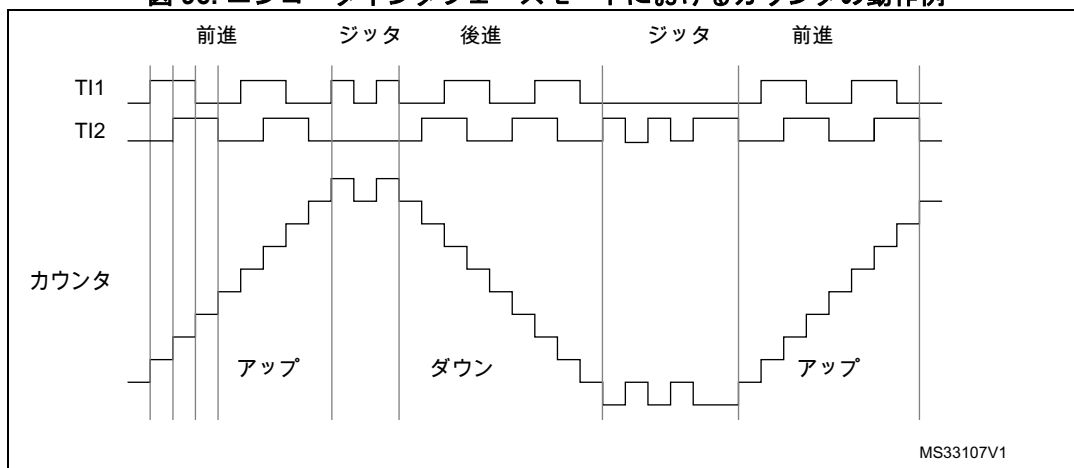
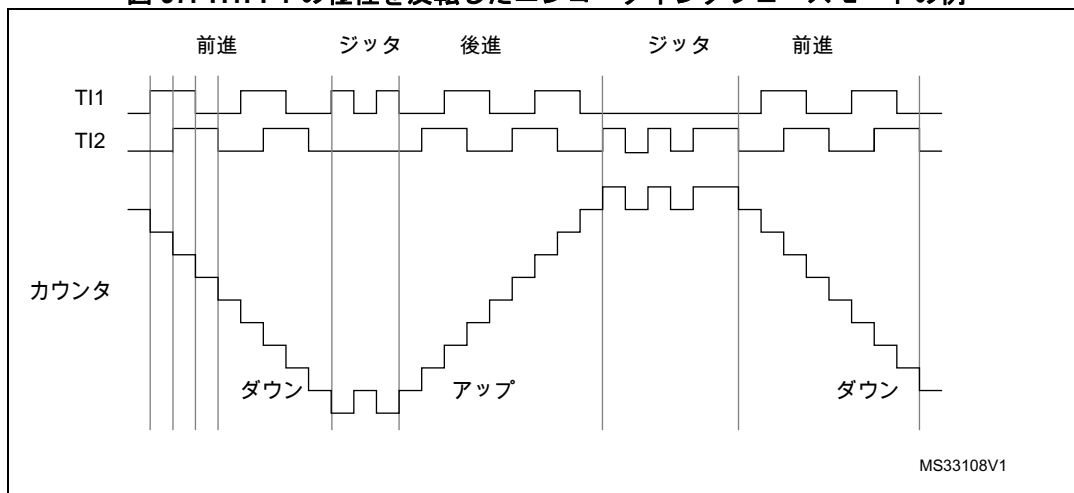


図 97 に、TI1FP1 の極性を反転したときのカウンタの動作を示します（上記と同じ設定ですが、CC1P="1"）。

図 97. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

16.3.17 タイマ入力 XOR 機能

TIMx_CR2 レジスタの TI1S ビットによって、チャンネル 1 の入力フィルタを 3 つの入力ピン TIMx_CH1、TIMx_CH2、および TIMx_CH3 を結合する XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。この機能をホールセンサのインタフェースに使用した例を、次の [セクション 16.3.18](#) に示します。

16.3.18 ホールセンサとのインタフェース

これは、高機能制御タイマ (TIM1) を使用して PWM 信号を生成し、モータと [図 98](#) で「インタフェースタイマ」と記されている別のタイマ TIMx (TIM2 または TIM3) を駆動することによって実現します。「インタフェースタイマ」は、XOR を通じて TI1 入力チャネル (TIMx_CR2 レジスタの TI1S ビットをセットすることで選択できます) に接続された 3 つのタイマ入力ピン (CC1、CC2、CC3) をキャプチャします。

スレーブモードコントローラはリセットモードに設定され、スレーブ入力は TI1F_ED です。したがって、3 つの入力のいずれかが反転するごとに、カウンタは 0 からカウントをリスタートします。これが、ホール入力の変化によってトリガされるタイムベースとなります。

「インタフェースタイマ」上で、キャプチャ/比較チャネル 1 がキャプチャモードで設定され、キャプチャ信号は TRC です ([図 81: キャプチャ/比較チャネル \(例: チャネル 1 入力ステージ\) \(324 ページ\)](#) を参照)。キャプチャされた値は、入力の 2 回の変化の間の経過時間に対応し、モータの速度情報を与えます。

「インタフェースタイマ」を出力モードで使用して、(COM イベントをトリガすることで) 高機能制御タイマ (TIM1) のチャネルの設定を変更するパルスを生成することができます。TIM1 タイマは、モータを駆動する PWM 信号を生成するために使用されます。このためには、プログラミングした遅延の後に正パルスが生成されるように (出力比較モードまたは PWM モードで) インタフェースタイマチャネルをプログラミングする必要があります。このパルスは、TRGO 出力を通じて高機能制御タイマ (TIM1) に送られます。

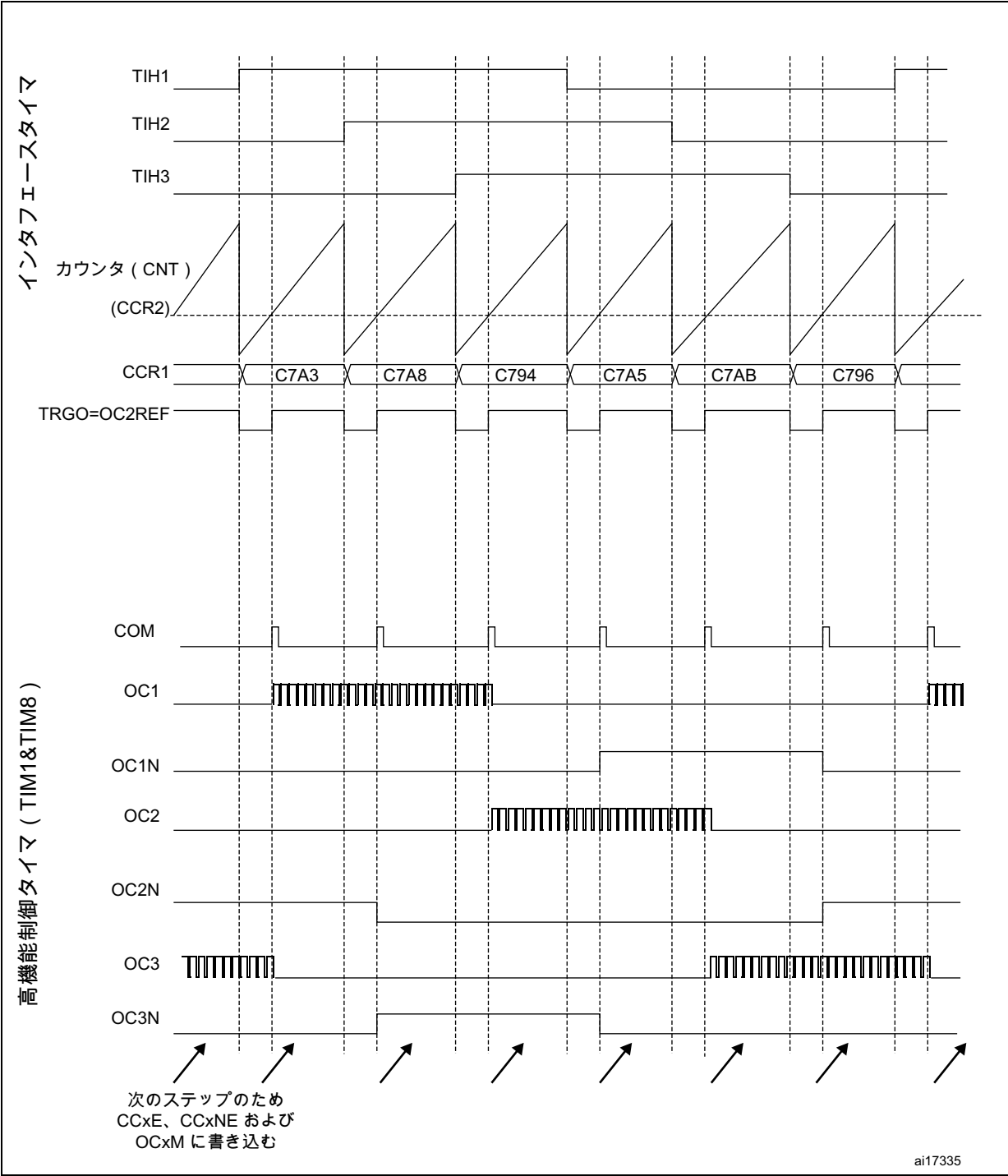
例: TIMx タイマの 1 つに接続されているホール入力に変化するたびに、プログラミングした遅延の後に高機能制御タイマ TIM1 の PWM 設定を変更するとします。

- 3 つのタイマ入力を TI1 入力チャネルに XOR 接続します。このためには、TIMx_CR2 レジスタの TI1S ビットに“1”を書き込みます。
- タイムベースをプログラムします。このためには、TIMx_ARR に最大値を書き込み、TI1 の変化でカウンタがクリアされるようにします。最大カウンタ時間がセンサの 2 回の変化の間の時間より長くなるように、プリスケアラを設定します。
- チャネル 1 をキャプチャモード (TRC 選択) にプログラムします。すなわち、TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。必要な場合は、デジタルフィルタをプログラムすることもできます。
- チャネル 2 を PWM 2 モードにプログラミングし、希望の遅延を指定します。このためには、TIMx_CCMR1 レジスタの OC2M ビットに“111”を、CC2S ビットに“00”を書き込みます。
- TRGO 上のトリガ出力として OC2REF を選択します。このためには、TIMx_CR2 レジスタの MMS ビットに“101”を書き込みます。

高機能制御タイマ TIM1 で、トリガ入力として適切な ITR 入力を選択する必要があり、タイマが PWM 信号を生成するようにプログラミングし、キャプチャ/比較制御信号がプリロードされ (TIMx_CR2 レジスタの CCPC=1)、COM イベントがトリガ入力によって制御されなければなりません (TIMx_CR2 レジスタの CCUS=1)。PWM 制御ビット (CCxE、OCxM) は、COM イベント後に次のステップのために書き込まれます (これは、OC2REF の立ち上がりエッジによって生成される割り込みサブルーチンで行うことができます)。

図 98 に、この例を示します。

図 98. ホールセンサインタフェースの例



16.3.19 TIMx と外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレープモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

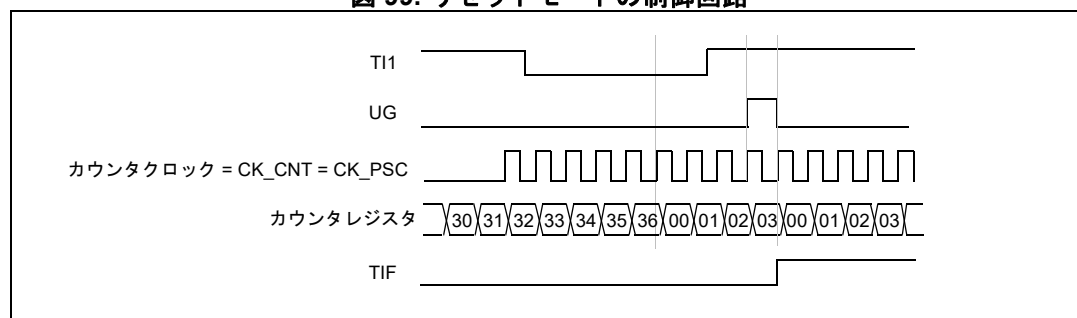
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないで、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタに CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

コード例については、「付録」セクション [A.9.12：リセットモードのコード例](#) を参照してください。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 99. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

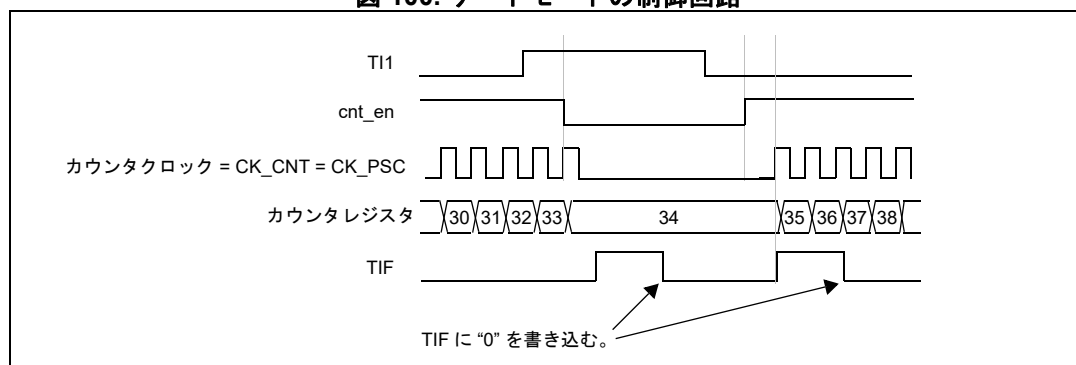
- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S=01 ビット）。TIMx_CCER レジスタで CC1P=1 と CC1NP=0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします（ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません）。

コード例については、「付録」セクション [A.9.13：ゲートモードのコード例](#) を参照してください。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 100. ゲートモードの制御回路



スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

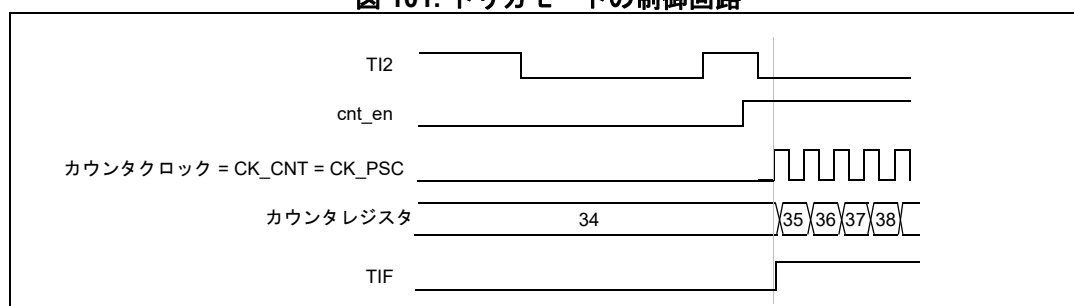
- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケアラはトリガには使用されないの、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

コード例については、「付録」セクション [A.9.14：トリガモードのコード例](#) を参照してください。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 101. トリガモードの制御回路



スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます（ただし、外部クロックモード 1 とエンコーダモードは除きます）。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます（リセットモード、ゲートモード、またはトリガモード）。TIMx_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000：フィルタなし
 - ETPS=00：プリスケアラ無効
 - ETP=0：ETR の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。

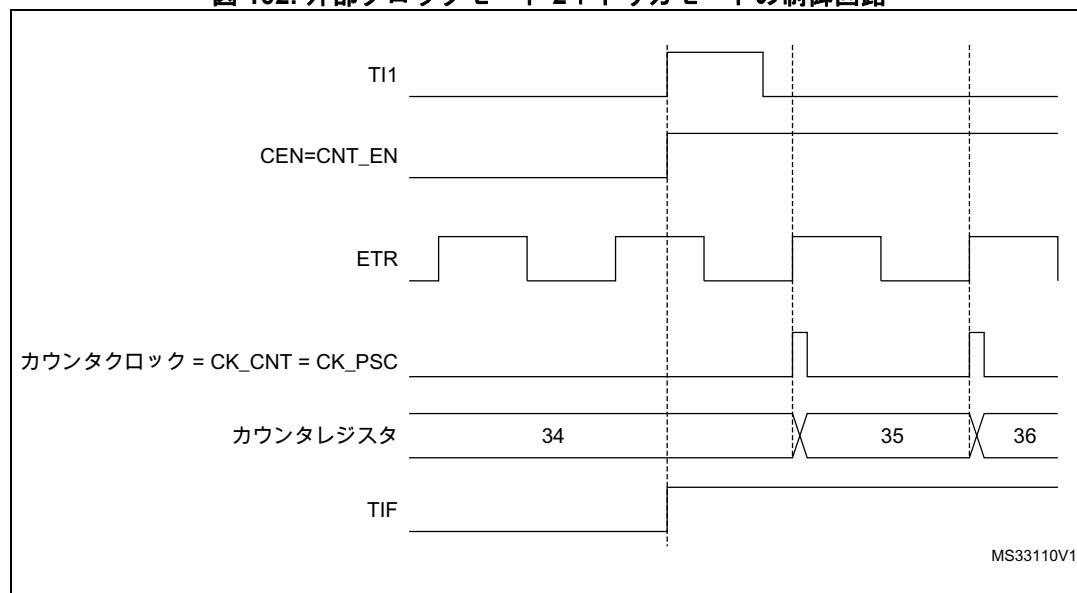
2. TIM1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000 : フィルタなし。
 - キャプチャプリスケラはトリガには使用されないなので、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

コード例については、「付録」セクション [A.9.15 : 外部クロックモード 2 + トリガモードのコード例](#) を参照してください。

TIM1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 102. 外部クロックモード 2+トリガモードの制御回路



16.3.20 タイマの同期

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 17.3.15 : タイマの同期 \(414 ページ\)](#) を参照してください。

16.3.21 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex™-M0 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

16.4 TIM1 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

16.4.1 TIM1 制御レジスタ 1 (TIM1_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (ETR、TIX) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

- 00 : $t_{DTS}=t_{CK\_INT}$
- 01 : $t_{DTS}=2*t_{CK\_INT}$
- 10 : $t_{DTS}=4*t_{CK\_INT}$
- 11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0 : TIMx_ARR レジスタはバッファされません。
- 1 : TIMx_ARR レジスタはバッファされます。

ビット 6:5 **CMS[1:0]** : センターアラインモード選択

- 00 : エッジアラインモードカウンタは、方向ビット (DIR) に応じて、カウントアップまたはカウントダウンします。
- 01 : センターアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。
- 10 : センターアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。
- 11 : センターアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル (TIMx_CCMRx レジスタの CCxS=00) の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

注 : カウンタが有効 (CEN = 1) なときに、エッジアラインモードからセンターアラインモードに切り替えることはできません。

ビット 4 **DIR** : 方向

- 0 : カウンタはアップカウンタとして使用されます。
- 1 : カウンタはダウンカウンタとして使用されます。

注 : このビットは、タイマがセンターアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。
0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します（有効な場合）。これらのイベントは、次のとおりです。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー／アンダーフローのみが更新割り込みまたは DMA リクエストを生成します（有効な場合）。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント（UEV）は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注： 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

16.4.2 TIM1 制御レジスタ 2 (TIM1_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **OIS4** : 出力アイドル状態 4 (OC4 出力)

OIS1 ビットの説明を参照してください。

ビット 13 **OIS3N** : 出力アイドル状態 3 (OC3N 出力)

OIS1N ビットの説明を参照してください。

ビット 12 **OIS3** : 出力アイドル状態 3 (OC3 出力)

OIS1 ビットの説明を参照してください。

ビット 11 **OIS2N** : 出力アイドル状態 2 (OC2N 出力)

OIS1N ビットの説明を参照してください。



ビット 10 **OIS2** : 出力アイドル状態 2 (OC2 出力)

OIS1 ビットの説明を参照してください。

ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)

0 : MOE=0 のとき、デッドタイム後に OC1N=0

1 : MOE=1 のとき、デッドタイム後に OC1N=0

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)

0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 7 **TI1S** : TI1 選択

0 : TIMx_CH1 ピンが TI1 入力に接続されます。

1 : TIMx_CH1、CH2、および CH3 ピンが TI1 入力に接続されます (XOR 接続)。

ビット 6:4 **MMS[1:0]** : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされるとき (すでにハイであった場合も)、トリガ出力は正のパルスを送信します (TRGO)。

100 : **比較** - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : **比較** - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

110 : **比較** - OC3REF 信号がトリガ出力 (TRGO) として使用されます。

111 : **比較** - OC4REF 信号がトリガ出力 (TRGO) として使用されます。

ビット 3 **CCDS** : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 **CCUS** : キャプチャ / 比較制御更新選択

- 0 : キャプチャ / 比較制御ビットがプリロードされるときには (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。
1 : キャプチャ / 比較制御ビットがプリロードされるときには (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **CCPC** : キャプチャ / 比較プリロード制御

- 0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。
1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込みの後、これらのビットは、転換イベント (COM) が発生した時にのみ更新されます (CCUS ビットに応じて、COMG ビットがセットまたは TRGI で立ち上がりエッジが検出されたとき)。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

16.4.3 TIM1 のスレーブモード制御レジスタ (TIM1_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15 **ETP** : 外部トリガ極性

このビットは、ETR と ETR のいずれがトリガ動作に使用されるかを選択します。

- 0 : ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブです。
1 : ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブです。

ビット 14 **ECE** : 外部クロックイネーブル

このビットは、外部クロックモード 2 を有効にします。

- 0 : 外部クロックモード 2 は無効です。
1 : 外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。

注 : 1: ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります (SMS=111、TS=111)。

2 : 外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません (TS ビットが 111 でないことが必要)。

3 : 外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力 ETRF です。

ビット 13:12 **ETPS[1:0]** : 外部トリガプリスケアラ

外部トリガ信号 ETRP の周波数は、TIMxCLK 周波数の 1/4 までに制限されます。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。

- 00 : プリスケアラオフ
01 : ETRP 周波数は 2 分周されます。
10 : ETRP 周波数は 4 分周されます。
11 : ETRP 周波数は 8 分周されます。



ビット 11:8 ETF[3:0] : 外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2

0010 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4

0011 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 8

注 : *ETF[3:0] = 1、2、または 3 のときに、式の f_{DTS} が CK_INT に置き換えられるよう注意する必要があります。*

ビット 7 MSM : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 TS[2:0] : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

000 : 内部トリガ 0 (ITR0)

001 : 内部トリガ 1 (ITR1)

010 : 内部トリガ 2 (ITR2)

011 : 内部トリガ 3 (ITR3)

100 : TI1 エッジ検出回路 (TI1F_ED)

101 : フィルタタイマ入力 1 (TI1FP1)

110 : フィルタタイマ入力 2 (TI2FP2)

111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、[表 54 : TIMx 内部トリガ接続 \(355 ページ\)](#) を参照してください。

注 : *設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。*

ビット 3 OCCS : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

0 : OCREF_CLR_INT は、OCREF_CLR 入力に接続されています。

1 : OCREF_CLR_INT は、ETRF に接続されています。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号（TRGI）のアクティブエッジが外部入力で選択された極性にリンクされます（入力制御レジスタおよび制御レジスタの説明を参照してください）。

000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。

001 : エンコーダモード 1 - カウンタは、TI1FP2 のレベルに応じて、TI2FP1 のエッジでカウントアップ/ダウンします。

010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。

011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。

100 : リセットモード - 選択されたトリガ入力（TRGI）の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

101 : ゲートモード - カウンタクロックは、トリガ入力（TRGI）がハイのときに有効になります。トリガがローになると、カウンタは停止します（リセットはされません）。カウンタの開始と停止の両方が制御されます。

110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します（リセットはされません）。カウンタの開始のみが制御されます。

111 : 外部クロックモード 1 - 選択されたトリガ（TRGI）の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合（TS=100）、ゲートモードを使用することはできません。TI1F_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

注： スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 54. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM15	TIM2	TIM3	TIM17

16.4.4 TIM1 DMA／割り込み有効レジスタ（TIM1_DIER）

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。
1 : トリガ DMA リクエストは有効です。

ビット 13 **COMDE** : COM DMA リクエストイネーブル

0 : COM DMA リクエストは無効です。
1 : COM DMA リクエストは有効です。

ビット 12 **CC4DE** : キャプチャ／比較 4 DMA リクエストイネーブル

0 : CC4 DMA リクエストは無効です。

1 : CC4 DMA リクエストは有効です。

ビット 11 **CC3DE** : キャプチャ／比較 3 DMA リクエストイネーブル

0 : CC3 DMA リクエストは無効です。

1 : CC3 DMA リクエストは有効です。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。

ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル

0 : CC1 DMA リクエストは無効です。

1 : CC1 DMA リクエストは有効です。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7 **BIE** : ブレーク割り込みイネーブル

0 : ブレーク割り込みは無効です。

1 : ブレーク割り込みは有効です。

ビット 6 **TIE** : トリガ割り込みイネーブル

0 : トリガ割り込みは無効です。

1 : トリガ割り込みは有効です。

ビット 5 **COMIE** : COM 割り込みイネーブル

0 : COM 割り込みは無効です。

1 : COM 割り込みは有効です。

ビット 4 **CC4IE** : キャプチャ／比較 4 割り込みイネーブル

0 : CC4 割り込みは無効です。

1 : CC4 割り込みは有効です。

ビット 3 **CC3IE** : キャプチャ／比較 3 割り込みイネーブル

0 : CC3 割り込みは無効です。

1 : CC3 割り込みは有効です。

ビット 2 **CC2IE** : キャプチャ／比較 2 割り込みイネーブル

0 : CC2 割り込みは無効です。

1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル

0 : CC1 割り込みは無効です。

1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

16.4.5 TIM1 ステータスレジスタ (TIM1_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **CC4OF** : キャプチャ/比較 4 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ/比較 3 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ/比較 2 オーバーキャプチャフラグ
CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ
このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。
0 : オーバーキャプチャは検出されていません。
1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BIF** : ブレーク割り込みフラグ
このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。
0 : ブレークイベントは発生していません。
1 : ブレーク入力アクティブレベルが検出されました。

ビット 6 **TIF** : トリガ割り込みフラグ
このフラグは、トリガイベント時（スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力アクティブエッジが検出されたとき）にハードウェアによってセットされます。ソフトウェアでクリアされます。
0 : トリガイベントは発生していません。
1 : トリガ割り込みが保留中です。

ビット 5 **COMIF** : COM 割り込みフラグ
このフラグは、COM イベント時にハードウェアによってセットされます（キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき）。ソフトウェアによってクリアされます。
0 : COM イベントは発生していません。
1 : COM 割り込みがペンディング中です。

ビット 4 **CC4IF** : キャプチャ/比較 4 割り込みフラグ
CC1IF の説明を参照してください。

ビット 3 **CC3IF** : キャプチャ/比較 3 割り込みフラグ
CC1IF の説明を参照してください。

ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます (センターアラインモードでは、例外もあります。TIMx_CR1 レジスタの CMS ビットの説明を参照してください)。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時 (アップカウントおよびアップ/ダウンカウントモードの場合)、またはアンダーフロー時 (ダウンカウントモードの場合) に CC1IF ビットはハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

– 繰り返しカウンタ値に関するオーバーフローまたはアンダーフロー (繰り返しカウンタ=0 の場合の更新)、および TIMx_CR1 レジスタの UDIS=0 の場合。

– TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

– TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき ([セクション 16.4.3 : TIM1 のスリープモード制御レジスタ \(TIM1_SMCR\)](#) を参照)。

16.4.6 TIM1 のイベント生成レジスタ (TIM1_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
								w	w	w	w	w	w	w	w

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 TG : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 COMG : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

注： このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4 CC4G : キャプチャ / 比較 4 イベント生成

CC1G の説明を参照してください。

ビット 3 CC3G : キャプチャ / 比較 3 イベント生成

CC1G の説明を参照してください。

ビット 2 CC2G : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

ビット 1 CC1G : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合：

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます（有効な場合）。

CC1 チャンネルが入力として設定されている場合：

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます（有効な場合）。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 UG : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。センターアラインモードが選択されている場合、または、DIR=0（カウントアップ）の場合、カウンタはクリアされます。そうでない場合、DIR=1（カウントダウン）であれば、自動再ロード値（TIMx_ARR）をとりまします。

16.4.7 TIM1 のキャプチャ／比較モードレジスタ 1 (TIM1_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル

ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード

ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル

ビット 10 **OC2FE** : 出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル

OC1CE : 出力比較 1 クリアイネーブル

0 : OC1Ref は ETRF 入力の影響を受けません。

1 : OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

- 000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。
- 001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。
- 010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。
- 011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。
- 100 : 強制非アクティブレベル - OC1REF は強制的にローになります。
- 101 : 強制アクティブレベル - OC1REF は強制的にハイになります。
- 110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。
- 111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注 :

- 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。
- 2 : PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。
- 3 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、OC1M アクティブビットは、COM が生成されたときにのみプリロードから新しい値をとります。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

- 0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。
- 1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 :

- 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。
- 2 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

- 0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。
- 1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

注 : **CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。**

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

注 : **CC2S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。**

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$, N = 2

0010 : $f_{SAMPLING} = f_{CK\_INT}$, N = 4

0011 : $f_{SAMPLING} = f_{CK\_INT}$, N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$, N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$, N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$, N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$, N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$, N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$, N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$, N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$, N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$, N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$, N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$, N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$, N = 8

注 : **ICx[F3:0] = 1, 2, または 3 のときに、式の f_{DTS} が f_{CK_INT} に置き換えられるよう注意する必要があります。**

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

- 00 : CC1 チャンネルは出力として設定されます。
- 01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。
- 10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。
- 11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

16.4.8 TIM1 のキャプチャ／比較モードレジスタ 2 (TIM1_CCMR2)

アドレスオフセット : 0x1C

リセット値 : 0x0000

上記の CCMR1 レジスタの説明を参照してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 15 **OC4CE** : 出力比較 4 クリアイネーブル

ビット 14:12 **OC4M** : 出力比較 4 モード

ビット 11 **OC4PE** : 出力比較 4 プリロードイネーブル

ビット 10 **OC4FE** : 出力比較 4 高速イネーブル

ビット 9:8 **CC4S** : キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

- 00 : CC4 チャンネルは出力として設定されます。
- 01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。
- 10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。
- 11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC4S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC4E=0) のときにのみ書き込み可能です。

ビット 7 **OC3CE** : 出力比較 3 クリアイネーブル

ビット 6:4 **OC3M** : 出力比較 3 モード

ビット 3 **OC3PE** : 出力比較 3 プリロードイネーブル

ビット 2 **OC3FE** : 出力比較 3 高速イネーブル

ビット 1:0 **CC3S** : キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

- 00 : CC3 チャンネルは出力として設定されます。
- 01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。
- 10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。
- 11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC3S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC3E=0) のときにのみ書き込み可能です。



入力キャプチャモード

- ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ
- ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケアラ
- ビット 9:8 **CC4S** : キャプチャ／比較 4 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC4 チャンネルは出力として設定されます。
- 01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。
- 10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。
- 11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。
- 注： **CC4S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC4E=0）のときにのみ書き込み可能です。
- ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ
- ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケアラ
- ビット 1:0 **CC3S** : キャプチャ／比較 3 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC3 チャンネルは出力として設定されます。
- 01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。
- 10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。
- 11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。
- 注： **CC3S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC3E=0）のときにのみ書き込み可能です。

16.4.9 TIM1 のキャプチャ／比較有効レジスタ（TIM1_CCER）

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 15:14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **CC4P** : キャプチャ / 比較 4 出力極性
- CC1P の説明を参照してください。
- ビット 12 **CC4E** : キャプチャ／比較 4 出力イネーブル
- CC1E の説明を参照してください。
- ビット 11 **CC3NP** : キャプチャ / 比較 3 相補出力極性
- CC1NP の説明を参照してください。
- ビット 10 **CC3NE** : キャプチャ / 比較 3 相補出力イネーブル
- CC1NE の説明を参照してください。
- ビット 9 **CC3P** : キャプチャ / 比較 3 出力極性
- CC1P の説明を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 8 **CC3E** : キャプチャ/比較 3 出力イネーブル

CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ/比較 2 相補出力極性

CC1NP の説明を参照してください。

ビット 6 **CC2NE** : キャプチャ/比較 2 相補出力イネーブル

CC1NE の説明を参照してください。

ビット 5 **CC2P** : キャプチャ/比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ/比較 2 出力イネーブル

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ/比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1N はアクティブハイです。

1 : OC1N はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

このビットは、TIM1FP1 と TIM2FP1 の極性を定義するために CC1P と組み合わせて使用されます。CC1P の説明を参照してください。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NP アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。

ビット 2 **CC1NE** : キャプチャ/比較 1 相補出力イネーブル

0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1NE アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

ビット 1 **CC1P** : キャプチャ / 比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

- 0 : OC1 はアクティブハイです。
- 1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 のアクティブ極性を選択します。

00 : 非反転/立ち上がりエッジ

この回路は TIxFP1 の立ち上がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードまたはエンコーダモードでのトリガ動作)。

01 : 反転/立ち下がりエッジ

この回路は TIxFP1 の立ち下がりエッジに反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されます (ゲートモードまたはエンコーダモードでのトリガ動作)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ

この回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (リセットモード、外部クロックモード、またはトリガモードにおけるキャプチャまたはトリガ動作)、TIxFP1 は反転されません (ゲートモードでのトリガ動作)。

この設定をエンコーダモードで使用することはできません。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1P アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 0 **CC1E** : キャプチャ / 比較 1 出力イネーブル

CC1 チャンネルが出力として設定されている場合 :

- 0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。
- 1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

- 0 : キャプチャは無効です。
- 1 : キャプチャは有効です。

注 : 相補出力を持つチャンネルでは、このビットはプリロードされます。TIMx_CR2 レジスタの CCPC ビットがセットされた場合、CC1E アクティブビットは、転換イベントが発生したときにのみ、プリロードされたビットから新しい値を取り込みます。

表 55. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能

制御ビット					出力状態 ⁽¹⁾	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	0	0	0	出力無効 (タイマによって駆動されない) OCx=0、OCx_EN=0	出力無効 (タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	0	1	出力無効 (タイマによって駆動されない) OCx=0、OCx_EN=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP、OCxN_EN = 1
		0	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	出力無効 (タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム OCxN_EN=1
		1	0	0	出力無効 (タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効 (タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
		1	0	1	オフ状態 (インアクティブ状態で出力有効) OCx=CCxP、OCx_EN=1	OCxREF + 極性 OCxN=OCxREF xor CCxNP、 OCxN_EN=1
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	オフ状態 (インアクティブ状態で出力有効) OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム OCxN_EN=1
0	0	X	0	0	出力無効 (タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効 (タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
	0		0	1	出力無効 (タイマによって駆動されない) 非同期: OCx=CCxP、OCx_EN=0、OCxN=CCxNP、 OCxN_EN=0	
	0		1	0		
	0		1	1	クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。 ただし、OISx と OISxN は、アクティブ状態における OCx と OCxN の両方 に対応しないことを前提とします。	
	1		0	0		
	1		0	1	出力無効 (タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効 (タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
	1		1	0	オフ状態 (インアクティブ状態で出力有効) 非同期: OCx=CCxP、OCx_EN=1、OCxN=CCxNP、 OCxN_EN=1	出力無効 (タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
	1		1	1	クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。 ただし、OISx と OISxN は、アクティブ状態における OCx と OCxN の両方 に対応しないことを前提とします。	

1. チャンネルの両方の出力が使用されないとき (CCxE=CCxNE=0)、OISx、OISxN、CCxP、および CCxNP はクリアされたままでなければなりません。

注： 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OC x および OC xN チャンネルの状態と、GPIO レジスタに依存します。

16.4.10 TIM1 のカウンタ (TIM1_CNT)

アドレスオフセット：0x24

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0]：カウンタ値

16.4.11 TIM1 のプリスケアラ (TIM1_PSC)

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値

カウンタクロック周波数 (CK_CNT) は $f_{CK\_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

16.4.12 TIM1 の自動再ロードレジスタ (TIM1_ARR)

アドレスオフセット：0x2C

リセット値：0xFFFF

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0]：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 16.3.1：タイムベースユニット \(306 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

16.4.13 TIM1 繰り返しカウンタレジスタ (TIM1_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **REP[7:0]** : 繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U_RC のみ、REP_CNT に REP 値がロードされるので、TIMx_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) は次のことを意味します。

- エッジアラインモードでは、PWM 周期の数
- センターアラインモードでは、PWM の 1/2 周期の数

16.4.14 TIM1 のキャプチャ／比較レジスタ 1 (TIM1_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

16.4.15 TIM1 のキャプチャ／比較レジスタ 2（TIM1_CCR2）

アドレスオフセット：0x38

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR2[15:0]**：キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合：

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合：

CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。

16.4.16 TIM1 のキャプチャ／比較レジスタ 3（TIM1_CCR3）

アドレスオフセット：0x3C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **CCR3[15:0]**：キャプチャ / 比較値

CC3 チャンネルが出力として設定されている場合：

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR3 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC3 出力に送信される値を含みます。

CC3 チャンネルが入力として設定されている場合：

CCR3 は、最後の入力キャプチャ 3 イベント（IC3）によって転送されたカウンタ値です。

16.4.17 TIM1 のキャプチャ／比較レジスタ 4 (TIM1_CCR4)

アドレスオフセット : 0x40

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 15:0 **CCR4[15:0]** : キャプチャ / 比較値

CC4 チャンネルが出力として設定されている場合 :
CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。
TIMx_CCMR4 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。
アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC4 出力に送信される値を含みます。
CC4 チャンネルが入力として設定されている場合 :
CCR4 は、最後の入力キャプチャ 4 イベント (IC4) によって転送されたカウンタ値です。

16.4.18 TIM1 ブレークおよびデッドタイムレジスタ (TIM1_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

注 : **ビット AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] は、LOCK 設定に応じて書き込みがロックされるので、TIMx_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。**

ビット 15 MOE : メイン出力イネーブル
このビットは、ブレーク入力がアクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャネルに対してのみ有効です。
0 : OC および OCN 出力は無効か、強制的にアイドル状態になります。
1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。
詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 16.4.9 : TIM1 のキャプチャ／比較有効レジスタ \(TIM1_CCER\) \(364 ページ\)](#))。

ビット 14 AOE : 自動出力イネーブル
0 : MOE はソフトウェアによってのみセットできます。
1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力 that アクティブでない場合)。
注 : **このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。**

ビット 13 BKP : ブレーク極性

- 0 : ブレーク入力 BRK はアクティブローです。
1 : ブレーク入力 BRK はアクティブハイです。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 BKE : ブレークイネーブル

- 0 : ブレーク入力 (BRK および CCS クロック障害イベント) は無効です。
1 : ブレーク入力 (BRK および CCS クロック障害イベント) は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 16.4.9 : TIM1 のキャプチャ/比較有効レジスタ \(TIM1_CCER\) \(364 ページ\)](#))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。
1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります。この場合、OC/OCN イネーブル出力信号 = 1 です。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 16.4.9 : TIM1 のキャプチャ/比較有効レジスタ \(TIM1_CCER\) \(364 ページ\)](#))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。
1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、まず強制的にアイドルレベルになります (OC/OCN イネーブル出力信号 = 1)。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 9:8 LOCK[1:0] : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

- 00 : LOCK オフ - どのビットも書き込み保護されません。
01 : LOCK レベル 1 = TIMx_BDTR レジスタの DTG ビット、TIMx_CR2 レジスタの OISx および OISxN ビット、および TIMx_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。
10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込めなくなります。
11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCMR レジスタの OCxM および OCxPE ビット) が書き込めなくなります。

注 : LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

$DTG[7:5]=0xx \Rightarrow DT=DTG[7:0] \times t_{dtg}$ 、ここで $t_{dtg}=t_{DTS}$ 。

$DTG[7:5]=10x \Rightarrow DT=(64+DTG[5:0]) \times t_{dtg}$ 、ここで $T_{dtg}=2 \times t_{DTS}$ 。

$DTG[7:5]=110 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=8 \times t_{DTS}$ 。

$DTG[7:5]=111 \Rightarrow DT=(32+DTG[4:0]) \times t_{dtg}$ 、ここで $T_{dtg}=16 \times t_{DTS}$ 。

例 : $T_{DTS}=125\text{ ns}$ (8 MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

16.4.19 TIM1 DMA 制御レジスタ (TIM1_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送回数 (タイマは、TIMx_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します) を指定します。

00000 : 1 回転送

00001 : 2 回転送

00010 : 3 回転送

...

10001 : 18 回転送

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx_CR1、

00001 : TIMx_CR2、

00010 : TIMx_SMCR

...

例 : 次の転送を考えます : DBL = 7 転送 かつ DBA = TIMx_CR1。この場合、転送は、TIMx_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

16.4.20 完全転送の TIM1 DMA アドレス（TIM1_DMAR）

アドレスオフセット：0x4C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **DMAB[15:0]**：DMA バーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります：

$(\text{TIMx\_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスはDMA 転送によって自動的に制御され、範囲は 0 から DBL です（DBL はTIMx_DCR 内で設定）。

DMA バースト機能の使用例

この例では、タイマ DMA バースト機能を使って CCRx レジスタ値を更新します（x = 2、3、4）。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
- DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。

– DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。

– 転送データ数 = 3 とします（下の注を参照）。

– サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
- DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします（DIER レジスタのUDE ビットをセット）。
4. TIMx を有効化
5. DMA チャンネルを有効化注：

注： この例は、各CCRx レジスタが1回更新される場合です。たとえば、各 CCRx レジスタが2回更新される場合は、転送データ数は6になります。データ1、データ2、データ3、データ4、データ5、データ6を格納するRAMのバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新DMA リクエストでデータ1がCCR2に転送され、データ2はCCR3に、データ3はCCR4にそれぞれ転送され、2番目の更新DMA リクエストでデータ4がCCR2に、データ5がCCR3に、データ6がCCR4にそれぞれ転送されます。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

16.4.21 TIM1 レジスタマップ

TIM1 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 56. TIM1 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
0x00	TIM1_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE		CMS [1:0]	DIR	OPM	URS	UDIS	CEN					
	リセット値																							0	0	0	0	0	0	0	0	0	0					
0x04	TIM1_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TIS	MMS[2:0]		CCDS	CCUS	Res.	CCPC						
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0					
0x08	TIM1_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETP	ECE	ETPS [1:0]		ETF[3:0]		MSM		TS[2:0]		OCS		SMS[2:0]								
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x0C	TIM1_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE					
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x10	TIM1_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF					
	リセット値																				0	0	0	0		0	0	0	0	0	0	0	0	0				
0x14	TIM1_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG					
	リセット値																									0	0	0	0	0	0	0	0	0				
0x18	TIM1_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2OE	OC2M [2:0]		OC2PE		OC2FE		CC2S [1:0]		OC1OE		OC1M [2:0]		OC1PE		OC1FE		CC1S [1:0]			
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	TIM1_CCMR1 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]		IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]		IC1 PSC [1:0]		CC1S [1:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1C	TIM1_CCMR2 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	O24OE	OC4M [2:0]		OC4PE		OC4FE		CC4S [1:0]		OC3OE		OC3M [2:0]		OC3PE		OC3FE		CC3S [1:0]			
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	TIM1_CCMR2 入力キャプチャモード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]		IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]		IC3 PSC [1:0]		CC3S [1:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x20	TIM1_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4P	CC4E	CC4NE	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E				
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x24	TIM1_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x28	TIM1_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x2C	TIM1_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x30	TIM1_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]												
	リセット値																									0	0	0	0	0	0	0	0	0	0			
0x34	TIM1_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 56. TIM1 レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x38	TIM1_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	TIM1_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	TIM1_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	TIM1_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]								
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	TIM1_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
	リセット値																				0	0	0	0	0				0	0	0	0	0
0x4C	TIM1_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

17 汎用タイマ (TIM2 および TIM3)

17.1 TIM2 および TIM3 の概要

この汎用タイマは、プログラム可能なプリスケラによって駆動される 16 ビットまたは 32 ビットの自動再ロードカウンタで構成されています。

入力信号のパルス長の測定 (入力キャプチャ) や出力波形の生成 (出力比較と PWM) など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

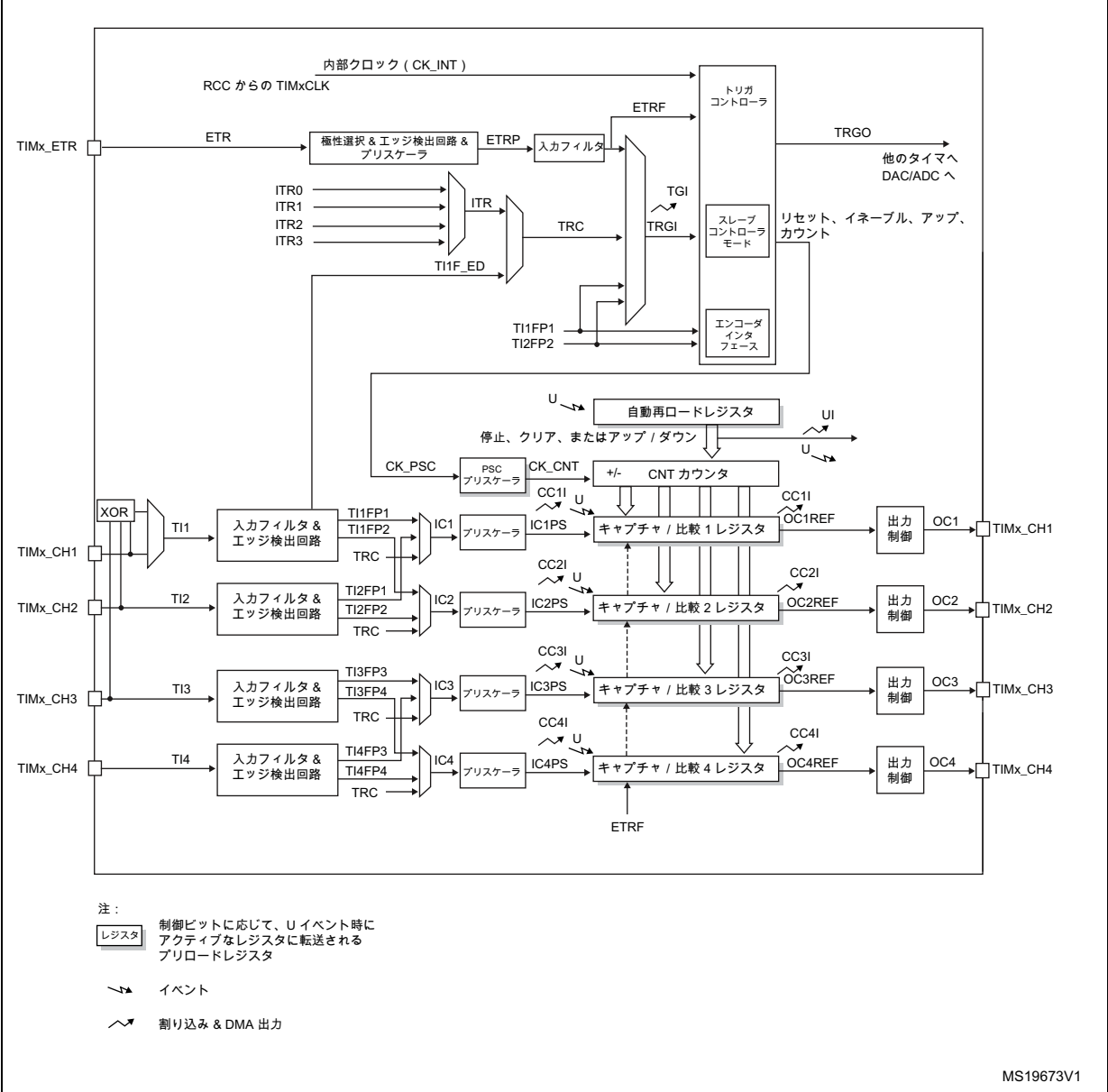
タイマは完全に独立していて、いかなるリソースも共有しません。これらのタイマは、[セクション 17.3.15](#)に示すように、相互に同期させることができます。

17.2 TIM2 および TIM3 の主な機能

汎用 TIMx タイマの主な機能は、次のとおりです。

- 16 ビット (TIM3) または 32 ビット (TIM2) のアップ、ダウン、アップ/ダウン自動再ロードカウンタ。
- カウンタクロック周波数を、1 から 65535 の間で分周する 16 ビットプログラム可能プリスケラ。
- 次の機能を持つ、最大 4 つの独立チャネル。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成 (エッジアラインモードとセンターアラインモード)
 - ワンパルスモード出力
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- 以下のイベント時の割り込み/DMA 生成。
 - 更新: カウンタオーバーフロー/アンダーフロー、カウンタの初期化 (ソフトウェアまたは内部/外部トリガによる)
 - トリガイベント (カウンタの開始、停止、初期化、または内部/外部トリガによるカウント)
 - 入力キャプチャ
 - 出力比較
- 位置決め目的のインクリメンタル (直交) エンコーダとホールセンサ回路をサポート
- 外部クロックまたはサイクルごとの電流管理のためのトリガ入力

図 103. 汎用タイマのブロック図 (TIM2 および TIM3)



17.3 TIM2 および TIM3 の機能説明

17.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビット/32 ビットカウンタです。カウンタはカウントアップしますが、カウントダウン、またはカウントアップとカウントダウンの両方を行う場合もあります。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウントイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

実際のカウンタイネーブル信号 CNT_EN は、CEN の 1 クロックサイクル後にセットされます。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビット/32 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 104](#) と [図 105](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 104. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

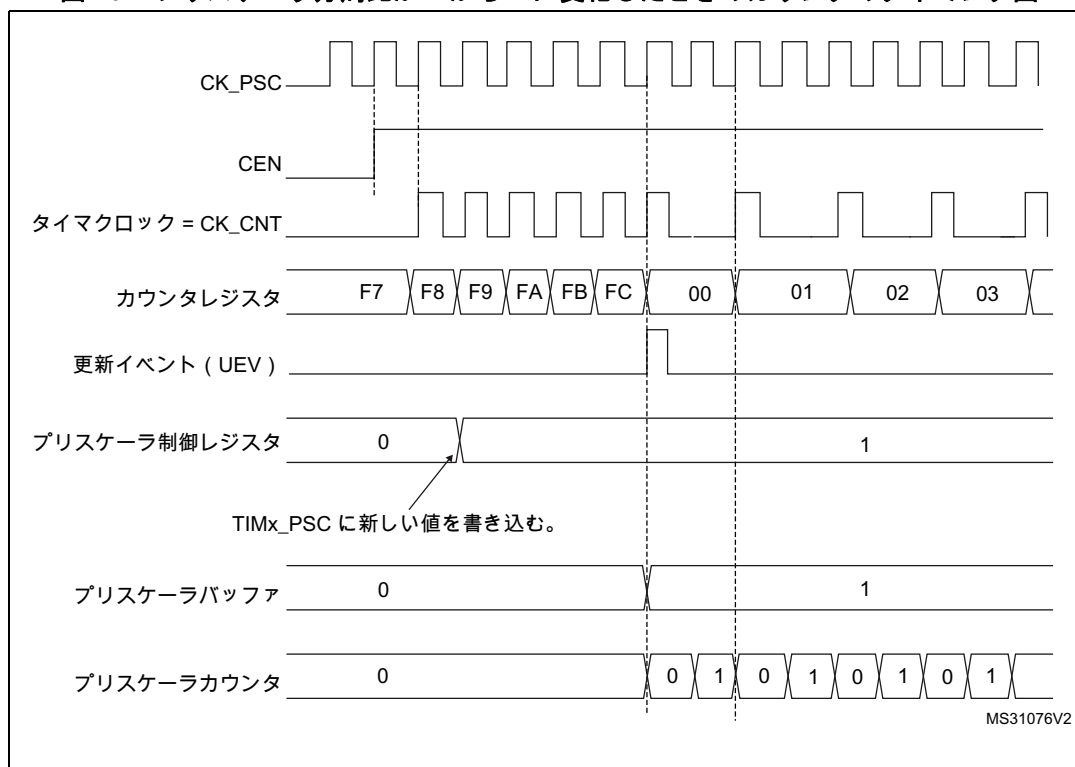
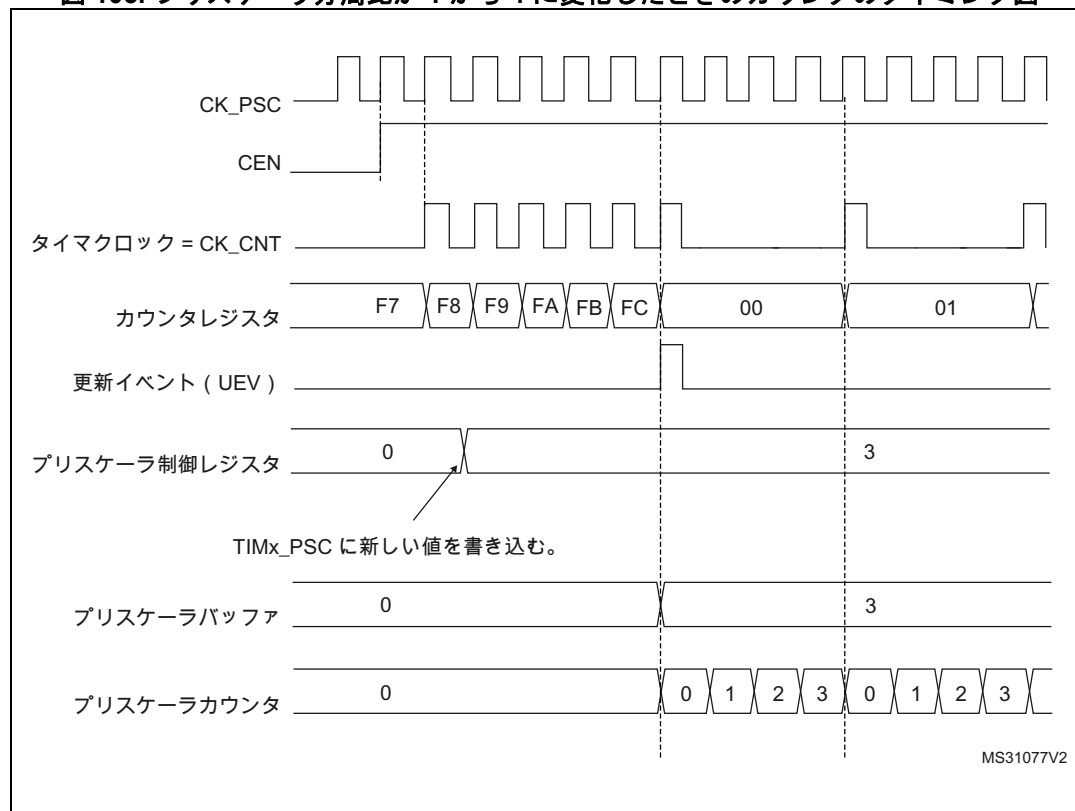


図 105. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



17.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、(ソフトウェアで、または、スレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることで生成できます。

UEV イベントは、TIMx_CR1 レジスタの UDIS ビットをセットすることにより、ソフトウェアで無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされません。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 106. 内部クロック分周比が 1 の場合のカウンタのタイミング図

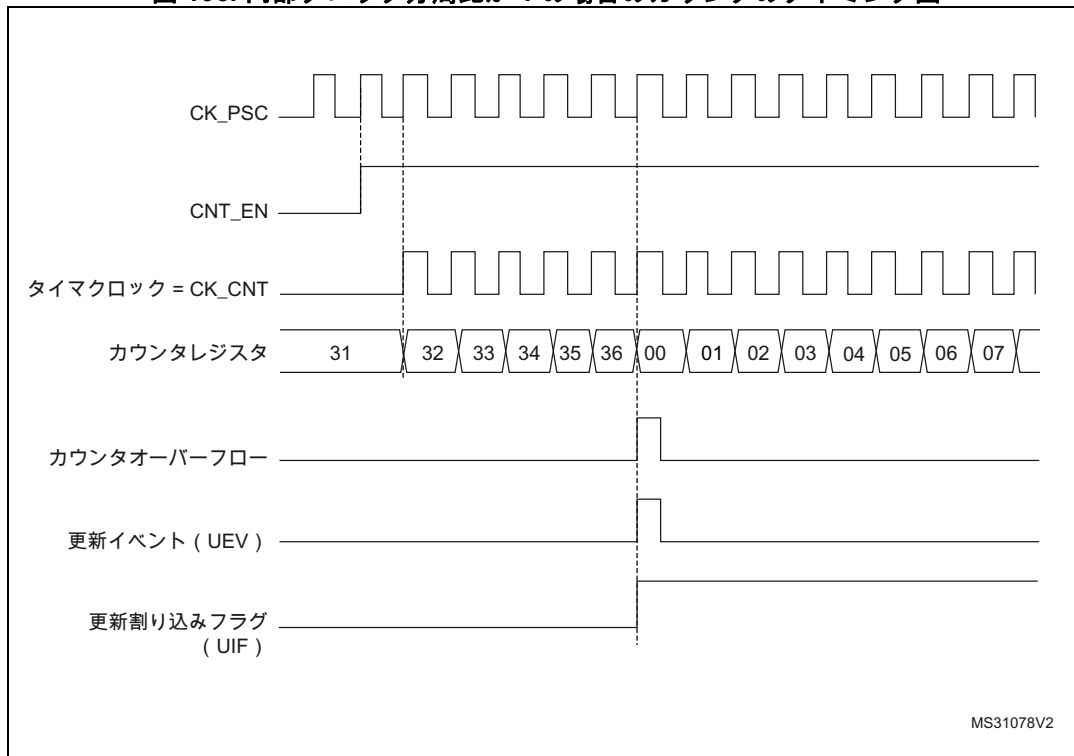


図 107. 内部クロック分周比が 2 の場合のカウンタのタイミング図

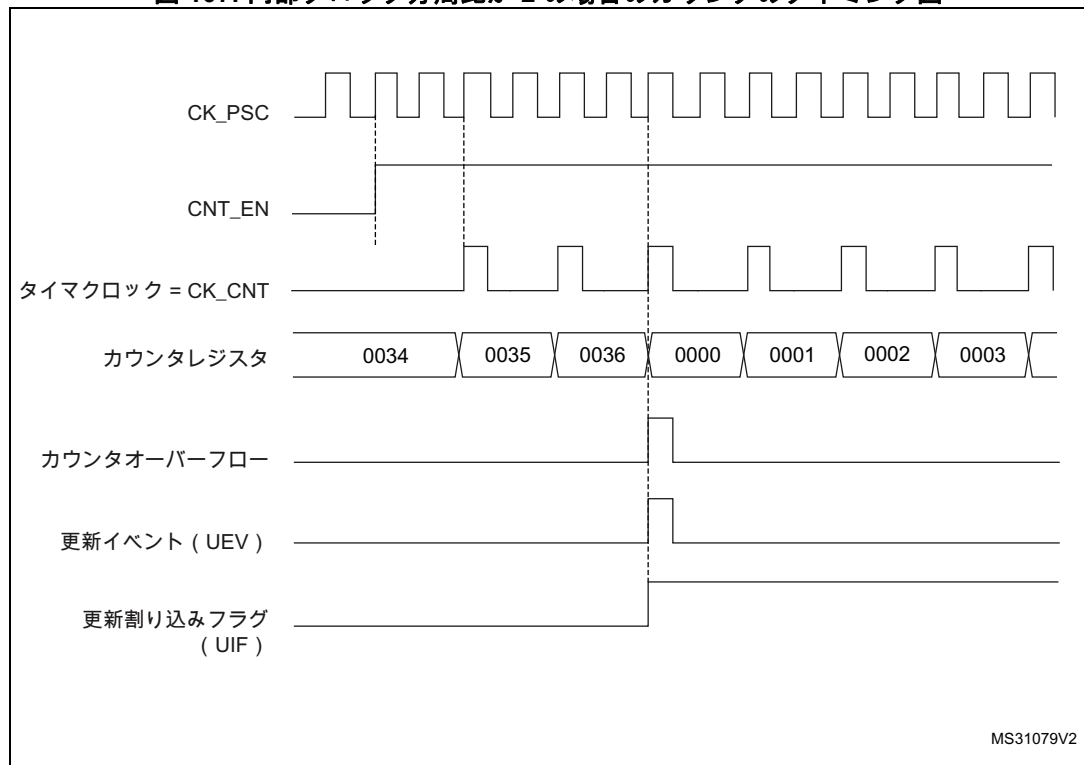


図 108. 内部クロック分周比が 4 の場合のカウンタのタイミング図

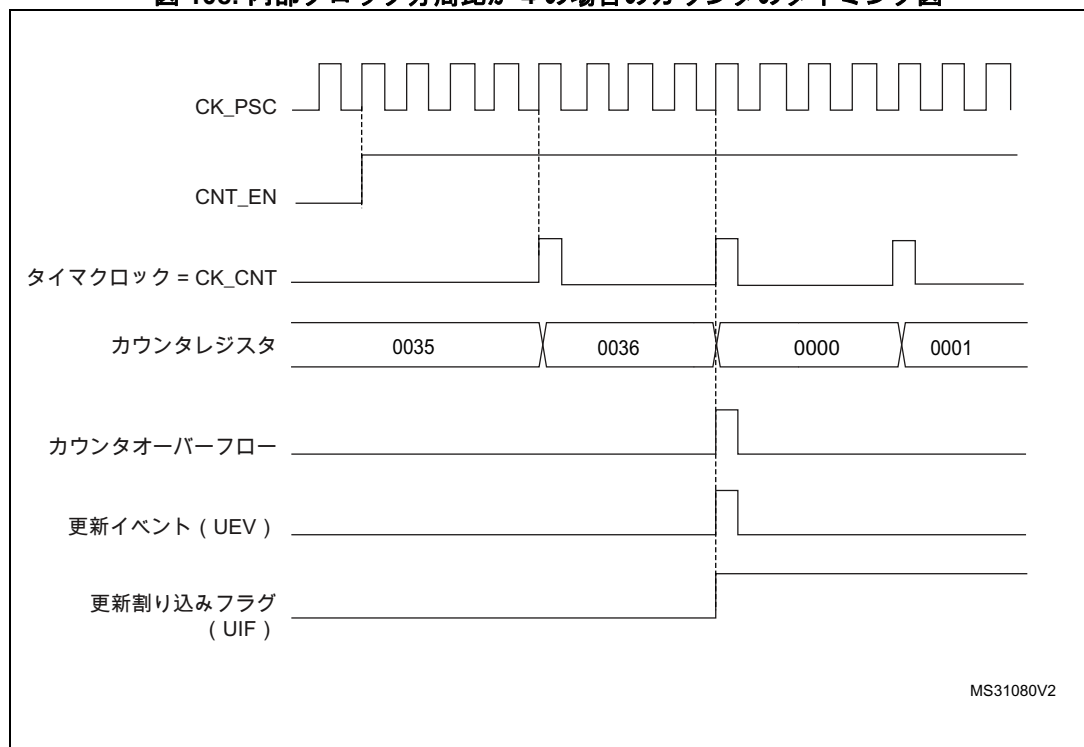


図 109. 内部クロック分周比が N の場合のカウンタのタイミング図

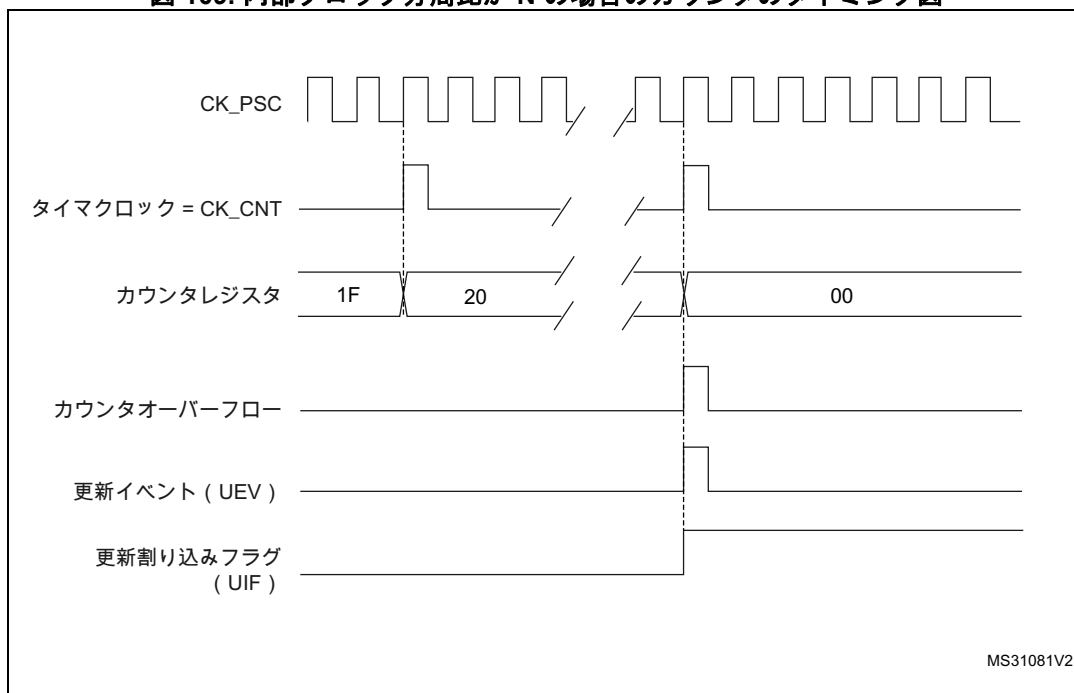
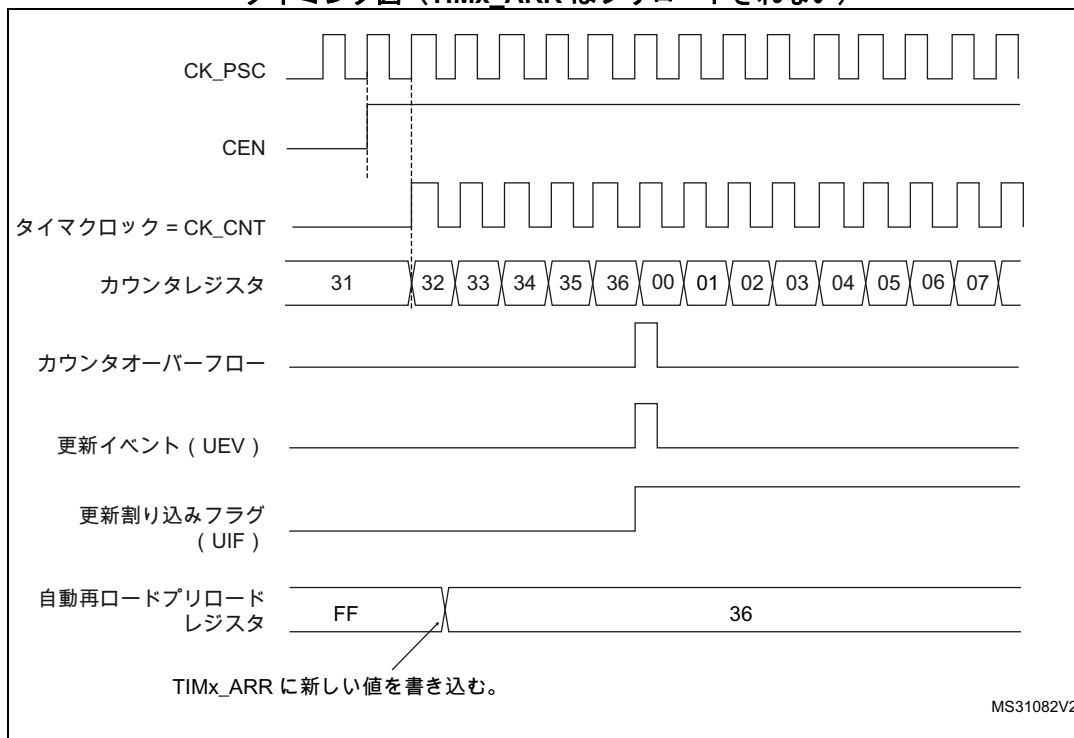


図 110. ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)



CK_PSC

CEN

タイムクロック = CK_CNT

カウンタレジスタ

カウンタオーバーフロー

更新イベント (UEV)

更新割り込みフラグ (UIF)

自動再ロード
プリロード レジスタ

自動再ロードシャドウ
レジスタ

TIMx_ARR に新しい値を書き込む。

MS31083V2

さらに、TIMx_CR1 レジスタの URS ビット（更新リクエスト選択）がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません（したがって、割り込みや DMA リクエストは送信されません）。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのパッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。カウンタがリロードされる前に自動再ロードが更新されるので、次の周期は期待通りの周期になります。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 112. 内部クロック分周比が 1 の場合のカウンタのタイミング図

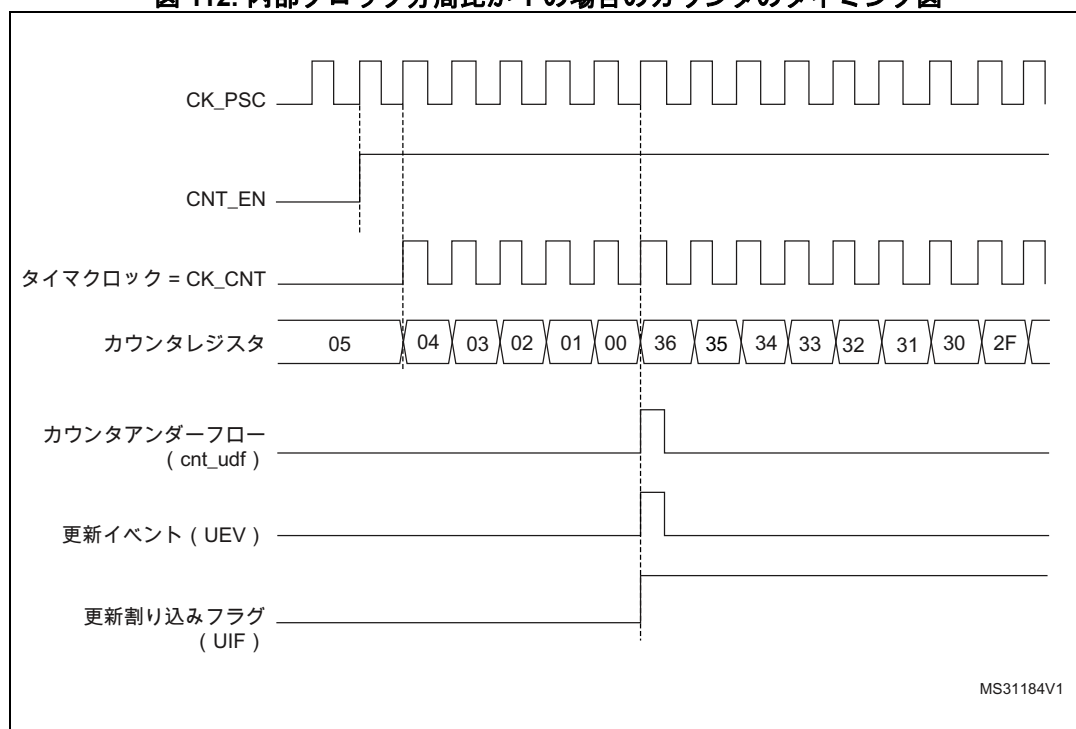


図 113. 内部クロック分周比が 2 の場合のカウンタのタイミング図

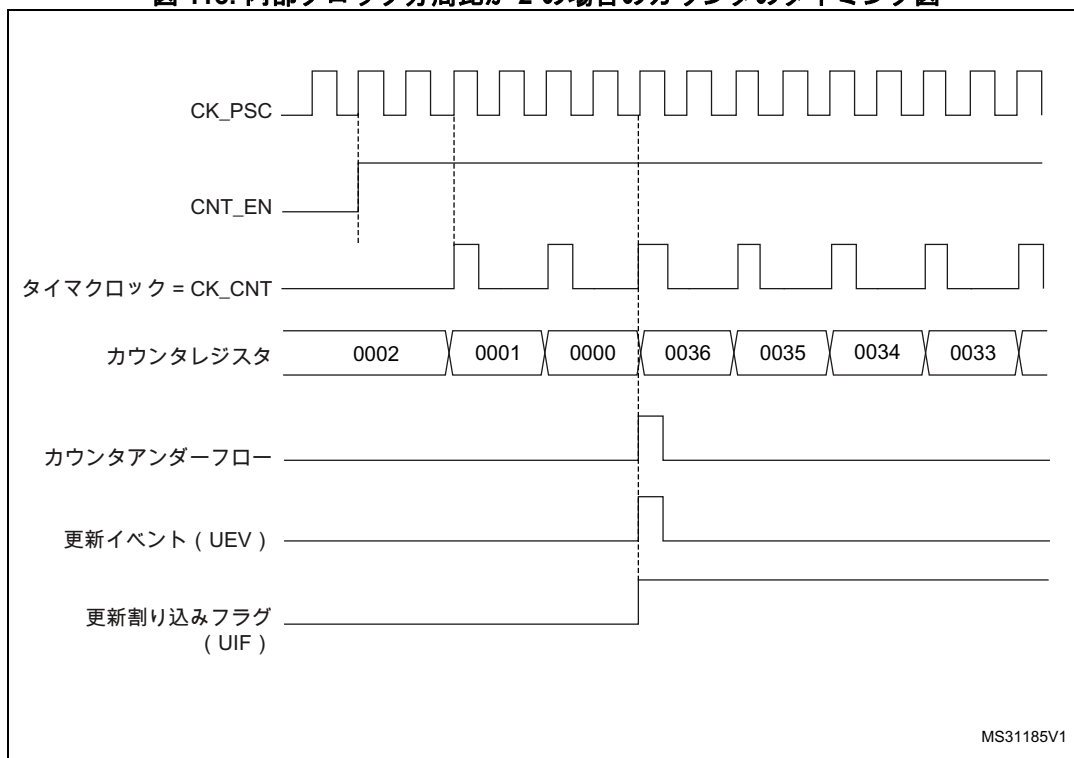


図 114. 内部クロック分周比が 4 の場合のカウンタのタイミング図

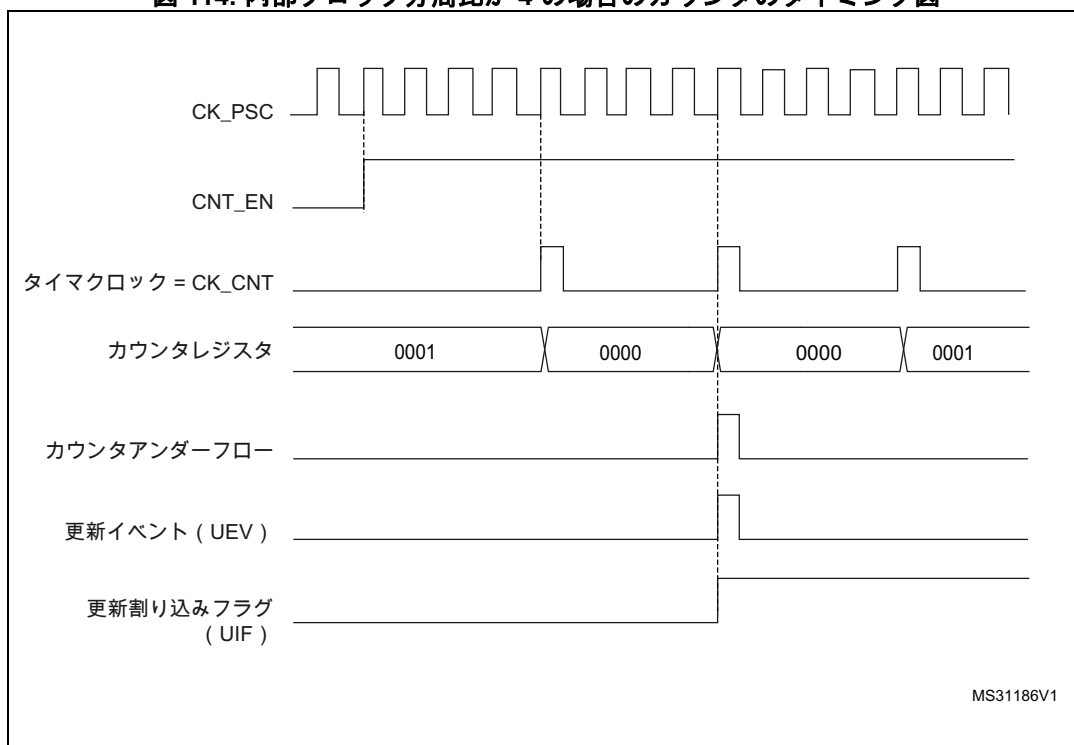


図 115. 内部クロック分周比が N の場合のカウンタのタイミング図

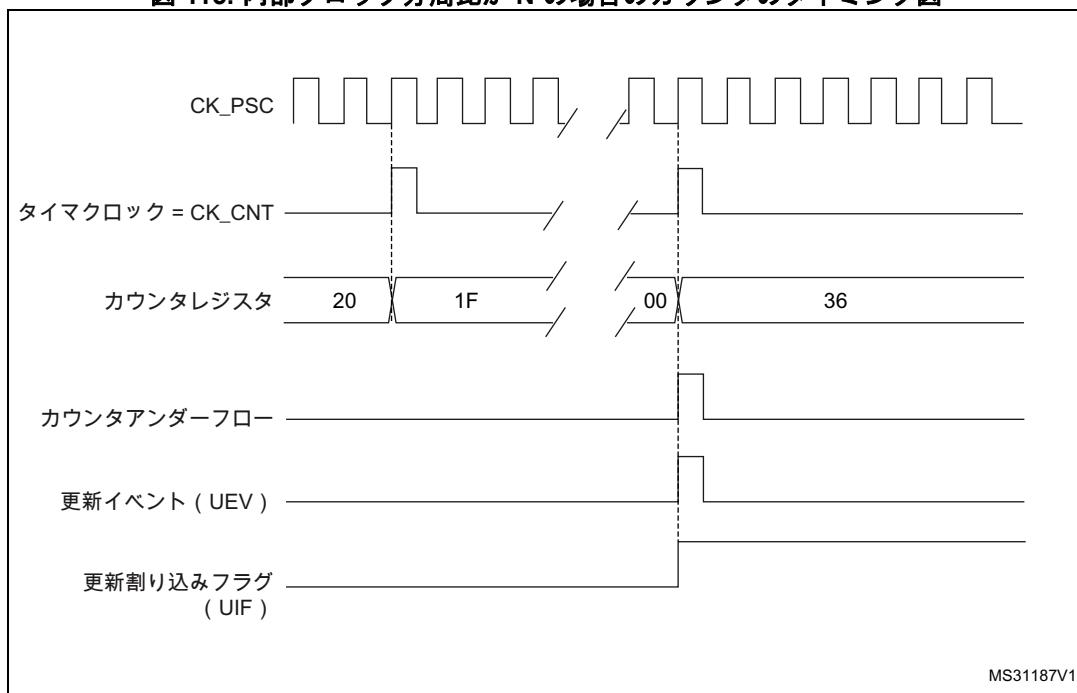
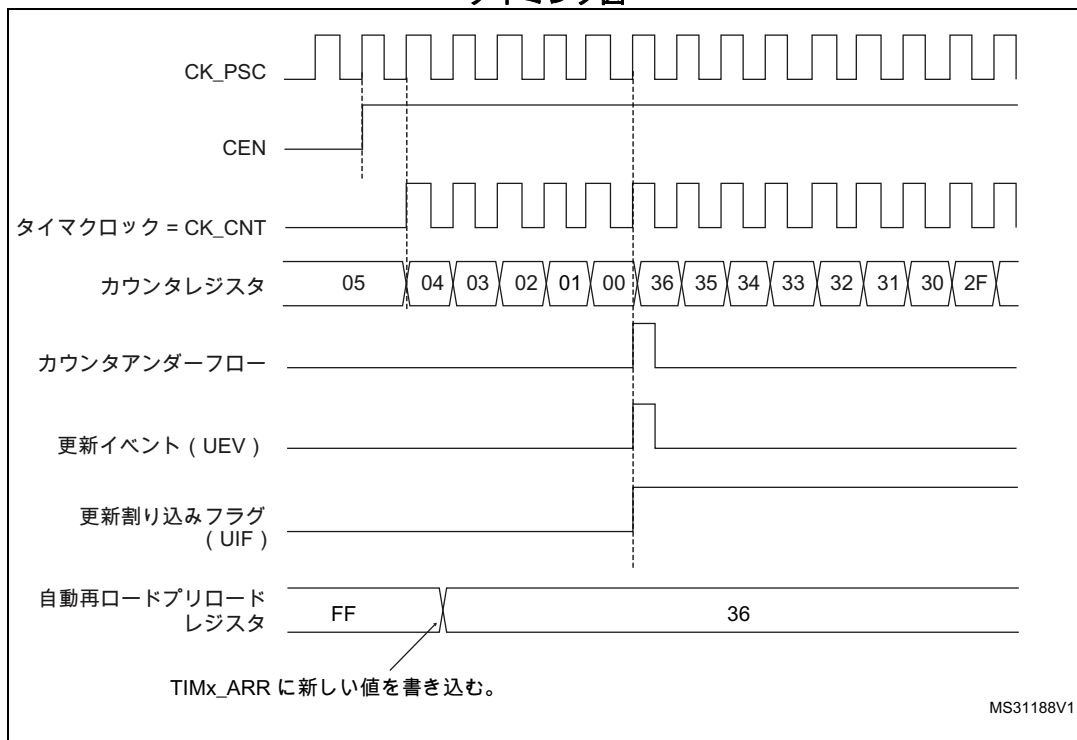


図 116. 繰り返しカウンタが使用されていないときの更新イベント時のカウンタのタイミング図



センターアラインモード (アップ/ダウンカウント)

センターアラインモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) -1 までカウントして、カウンタオーバーフローイベントを生成した後、自動再ロード値から 1 までカウントして、カウンタアンダーフローイベントを生成します。その後、0 からカウントをリスタートします。

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”に等しくないときにアクティブとなります。出力に設定されたチャンネルの出力比較割り込みフラグは、カウンタがカウントダウンするとき (センターアラインモード 1、CMS="01")、カウンタがカウントアップするとき (センターアラインモード 2、CMS="10")、またはカウンタがカウントアップしてカウントダウンするとき (センターアラインモード 3、CMS="11") にセットされます。

このモードでは、方向ビット (TIMx_CR1 レジスタの DIR) に書き込むことはできません。このビットは、ハードウェアによって更新されて、カウンタの現在の方向を示します。

更新イベントは、カウンタオーバーフローとカウンタアンダーフローごとに生成されます。または、(ソフトウェアで、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも、更新イベントが生成されます。この場合、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします。

UEV 更新イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることにより無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、カウンタは現在の自動再ロード値に基づいて、カウントアップとカウントダウンを続けます。

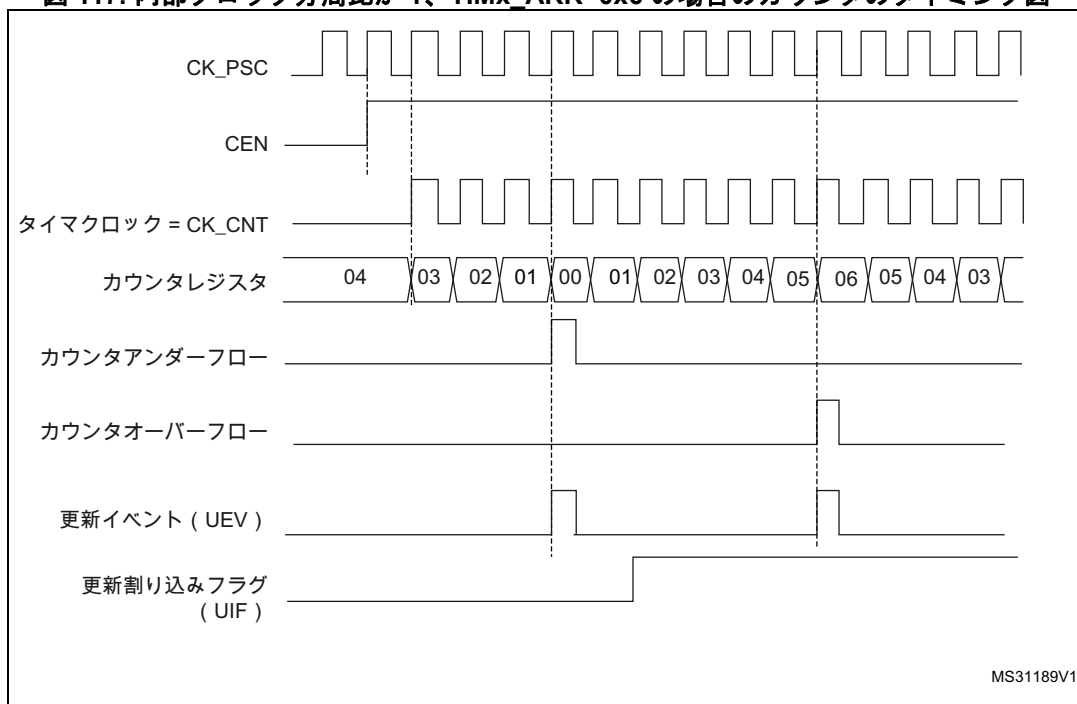
さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのパッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。
- 自動再ロードアクティブレジスタは、プリロード値 (TIMx_ARR レジスタの内容) で更新されます。更新の原因がカウンタオーバーフローである場合には、自動再ロードが更新されてからカウンタが再ロードされるので、次の周期は期待通りの周期になります (カウンタに新しい値がロードされます)。

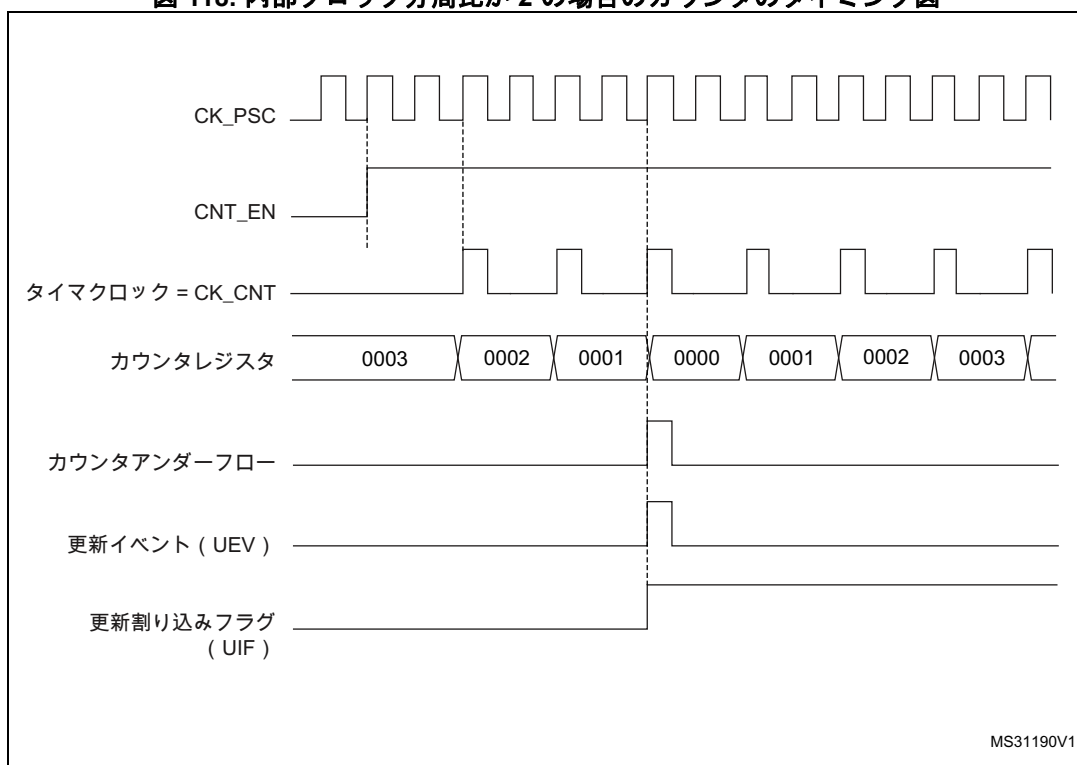
以下の図は、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 117. 内部クロック分周比が 1、TIMx_ARR=0x6 の場合のカウンタのタイミング図



1. ここでは、センターラインモード 1 が使用されています (詳細については、[セクション 17.4.1 : TIM2 および TIM3 制御レジスタ 1 \(TIM2_CR1 および TIM3_CR1\) \(420 ページ\)](#) を参照してください)。

図 118. 内部クロック分周比が 2 の場合のカウンタのタイミング図



注：この場合、センターアラインモード 2 または 3 はオーバーフロー時に UIF で更新されます。

MS31191V1

-
- CK_PSC
- タイマクロック = CK_CNT
- カウンタレジスタ
- 20 1F
- 01 00
- カウンタアンダーフロー
- 更新イベント (UEV)
- 更新割り込みフラグ (UIF)
- MS31192V1

図 121. ARPE=1（カウンタアンダーフロー）の場合の更新イベント時、
カウンタタイミング図

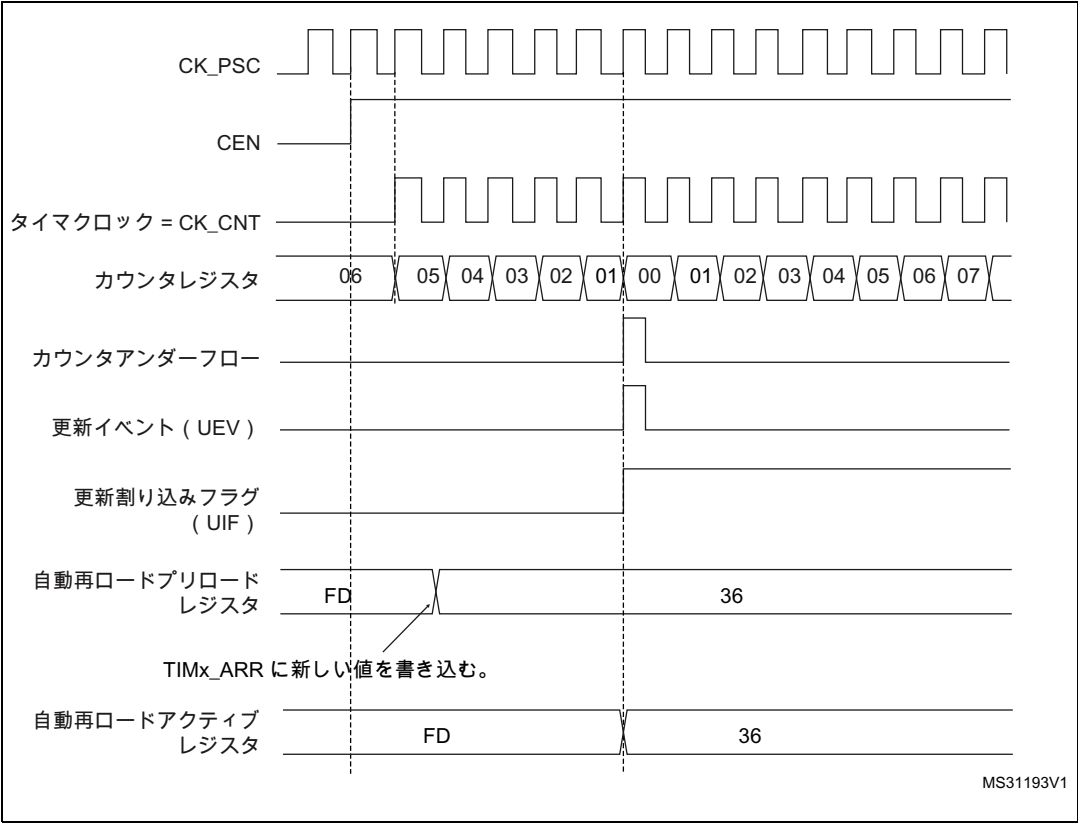
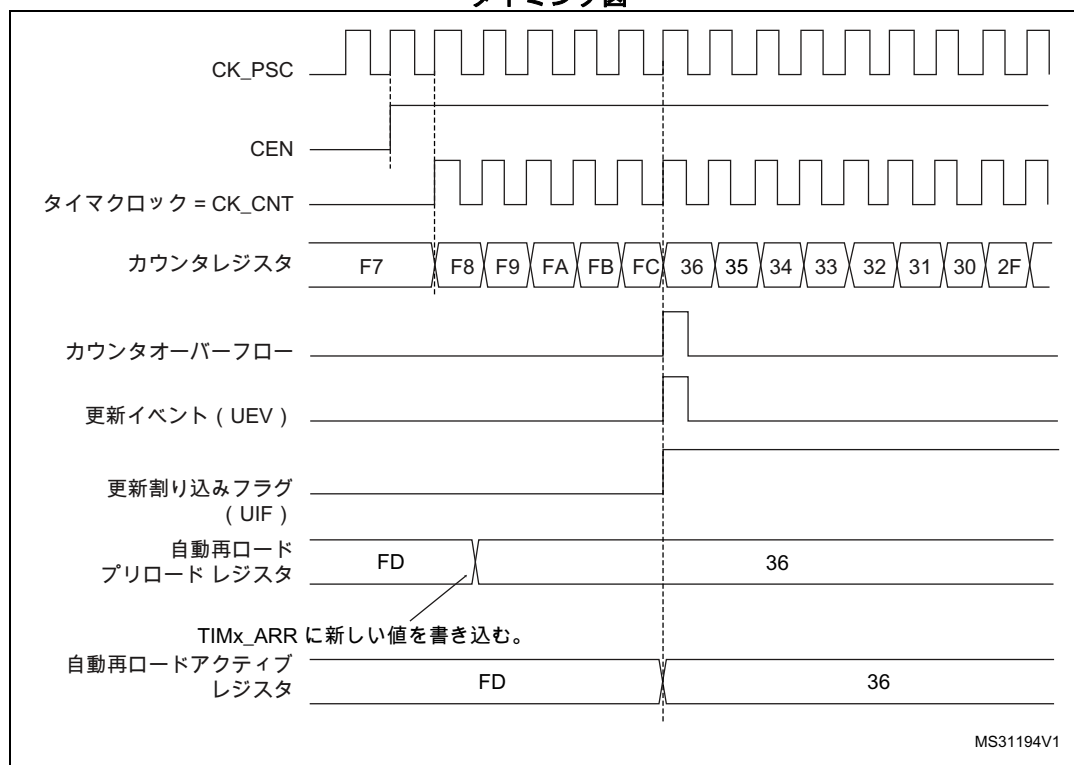


図 122. ARPE=1（カウンタオーバーフロー）の場合の更新イベント時のカウンタのタイミング図



17.3.3 クロックソース

カウンタクロックは、次のクロックソースによって供給されます。

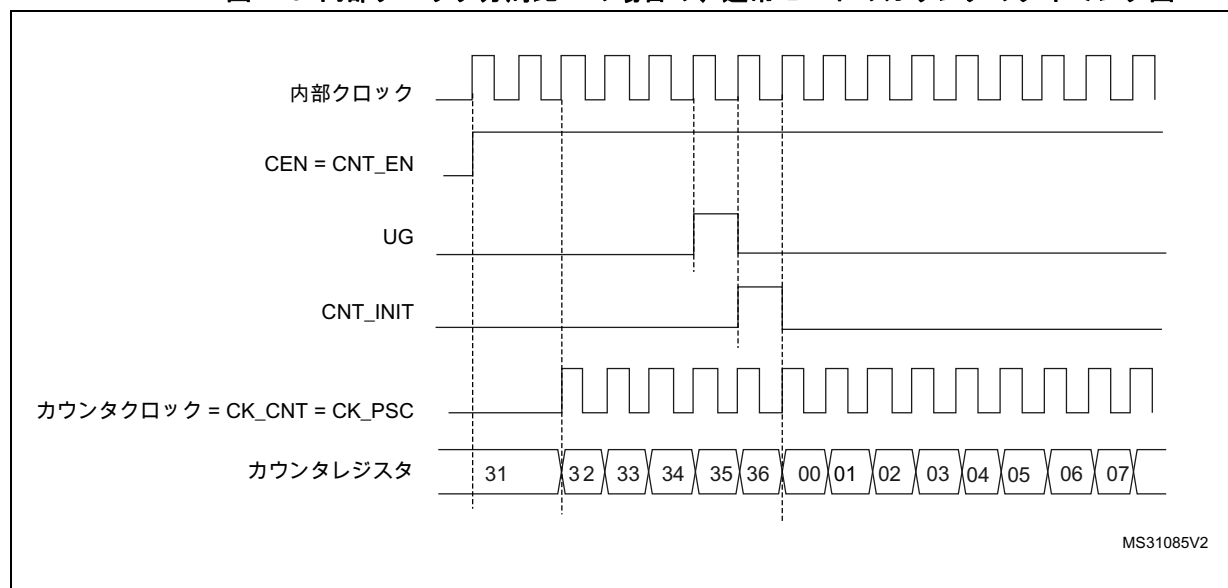
- 内部クロック（CK_INT）
- 外部クロックモード 1：外部入力ピン（Tix）
- 外部クロックモード 2：外部トリガ入力（ETR）
- 内部トリガ入力（ITRx）：あるタイマを別のタイマのプリスケアラとして使用します。たとえば、タイマ 1 がタイマ 2 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する（414 ページ）](#)を参照してください。

内部クロックソース（CK_INT）

スレーブモードコントローラが無効の場合（TIMx_SMCR レジスタの SMS=000）、CEN、DIR ビット（TIMx_CR1 レジスタ）と UG ビット（TIMx_EGR レジスタ）が実際の制御ビットであり、ソフトウェアでのみ変更できます（自動的にクリアされたままの UG ビットを除きます）。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 123](#)に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

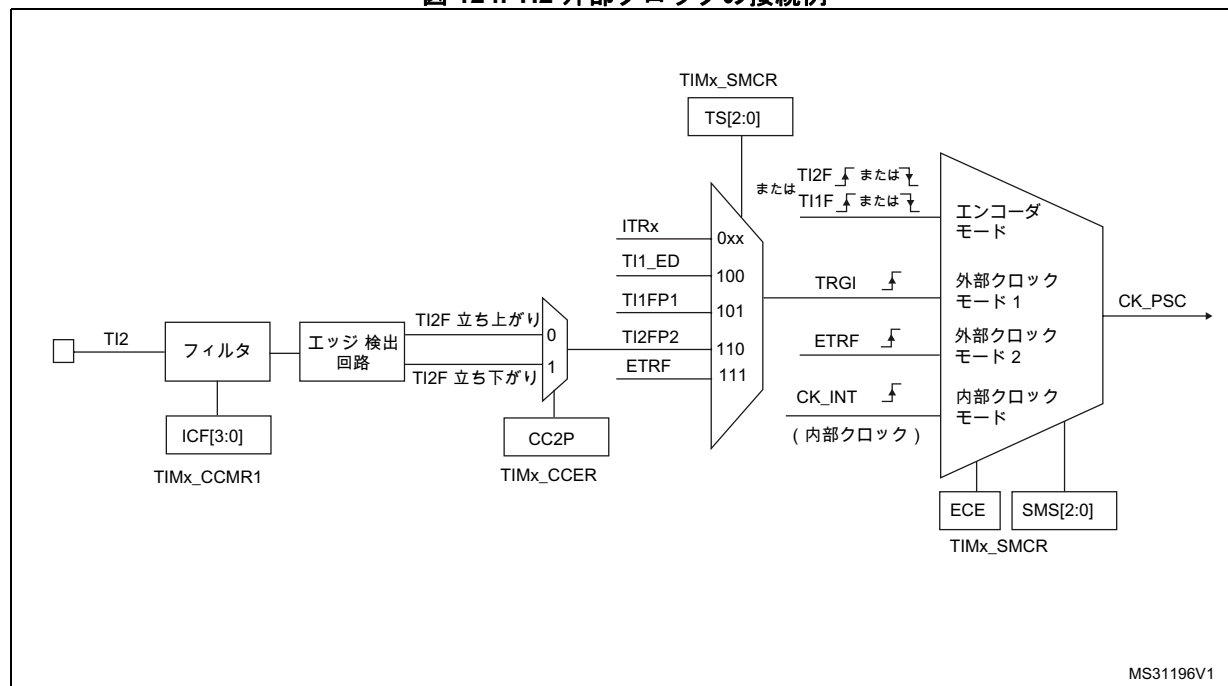
図 123. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 124. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_CCMR1 レジスタに CC2S=01 を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。

注： キャプチャプリスケアラはトリガには使用されないで、設定は不要です。

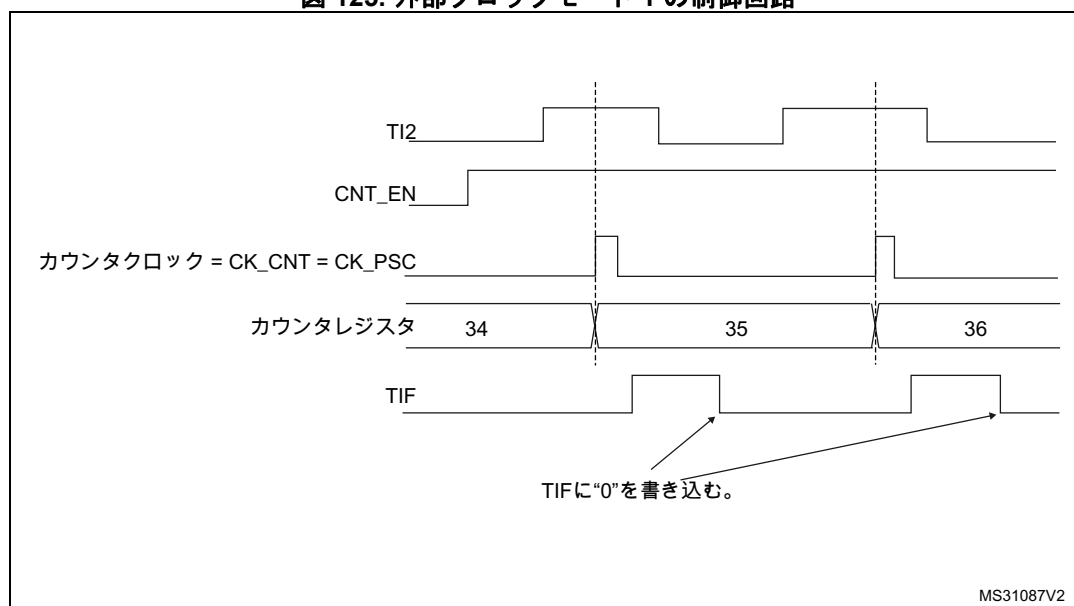
3. CC2P=0 と CC2NP=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

コード例については、「付録」セクション [A.9.2 : 2 回の ETR の立ち上がりエッジごとのアップカウンタのコード例](#) を参照してください。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 125. 外部クロックモード 1 の制御回路



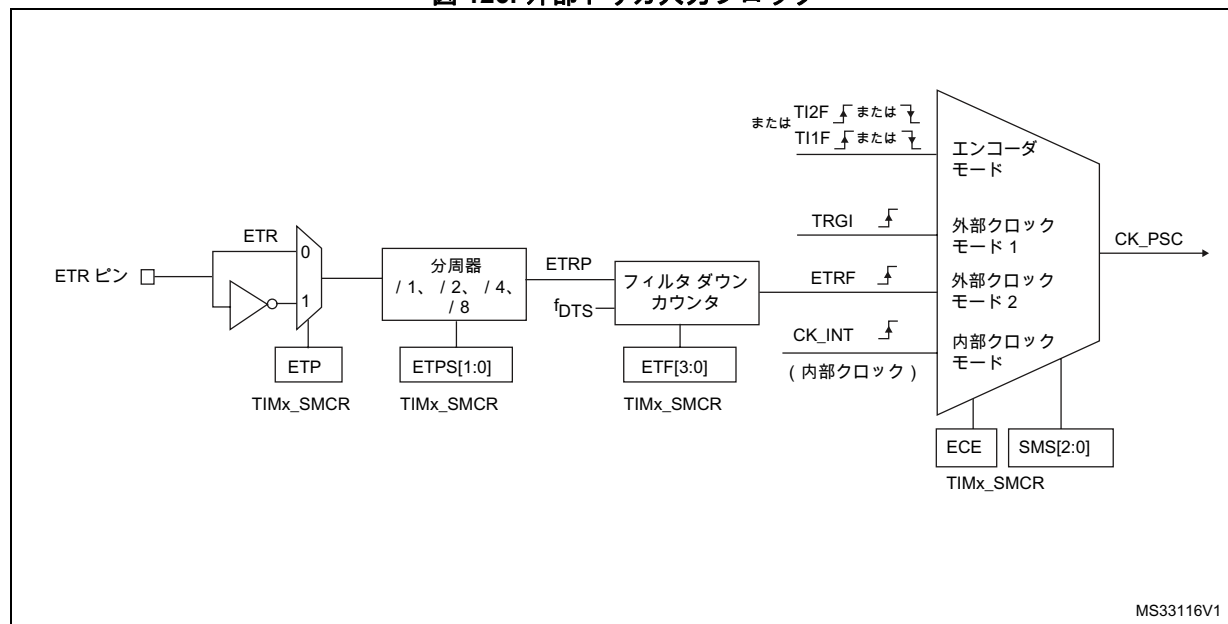
外部クロックソースモード 2

このモードは、TIMx_SMCR レジスタの ECE=1 を書き込むことによって選択されます。

カウンタは、外部トリガ入力 ETR の立ち上がりまたは立ち下がりエッジごとにカウントできます。

図 126 に、外部トリガ入力ブロックの概要を示します。

図 126. 外部トリガ入力ブロック



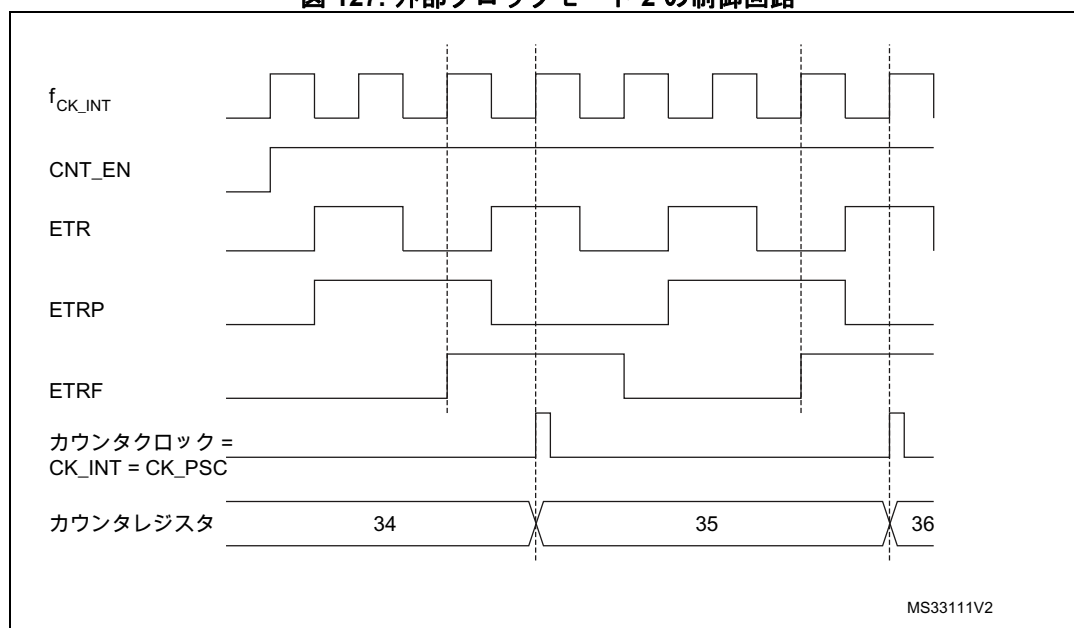
たとえば、ETR の 2 回の立ち上がりエッジごとにカウントするようにアップカウンタを設定するには、以下の手順に従います。

1. この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
2. TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
3. TIMx_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を選択します。
4. TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
5. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

カウンタは 2 回の ETR 立ち上がりエッジごとに 1 回カウントします。

ETR の立ち上がりエッジから実際のカウンタクロックまでの間に、ETRP 信号の再同期回路による遅延があります。

図 127. 外部クロックモード 2 の制御回路



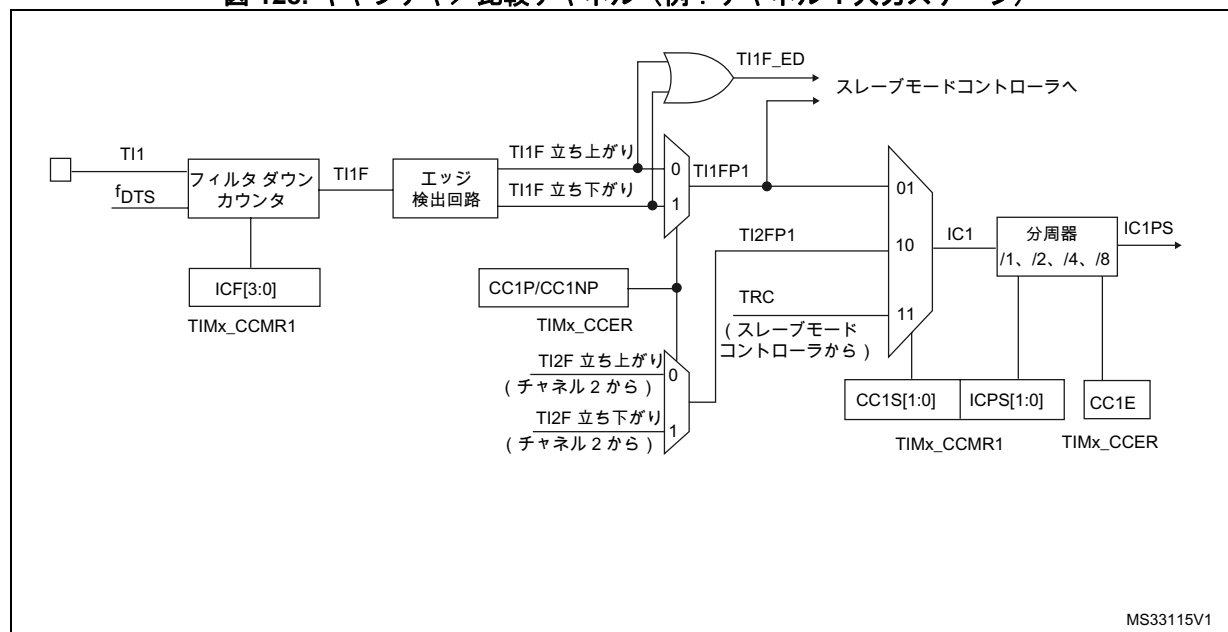
17.3.4 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ（シャドウレジスタを含む）、キャプチャの入カステージ（デジタルフィルタ、マルチプレクサ、ブリスケーラ）、および出力カステージ（比較回路と出力制御）から構成されています。

次の図に、キャプチャ／比較チャネルの概要を示します。

入カステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った $TIxF$ を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号（ $TIxFPx$ ）を生成します。この信号はブリスケーラを通じて、キャプチャレジスタ（ $ICxPS$ ）に渡されます。

図 128. キャプチャ/比較チャンネル (例: チャンネル 1 入力ステージ)



出力ステージは、OCxRef (アクティブハイ) として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 129. キャプチャ/比較チャンネル 1 メイン回路

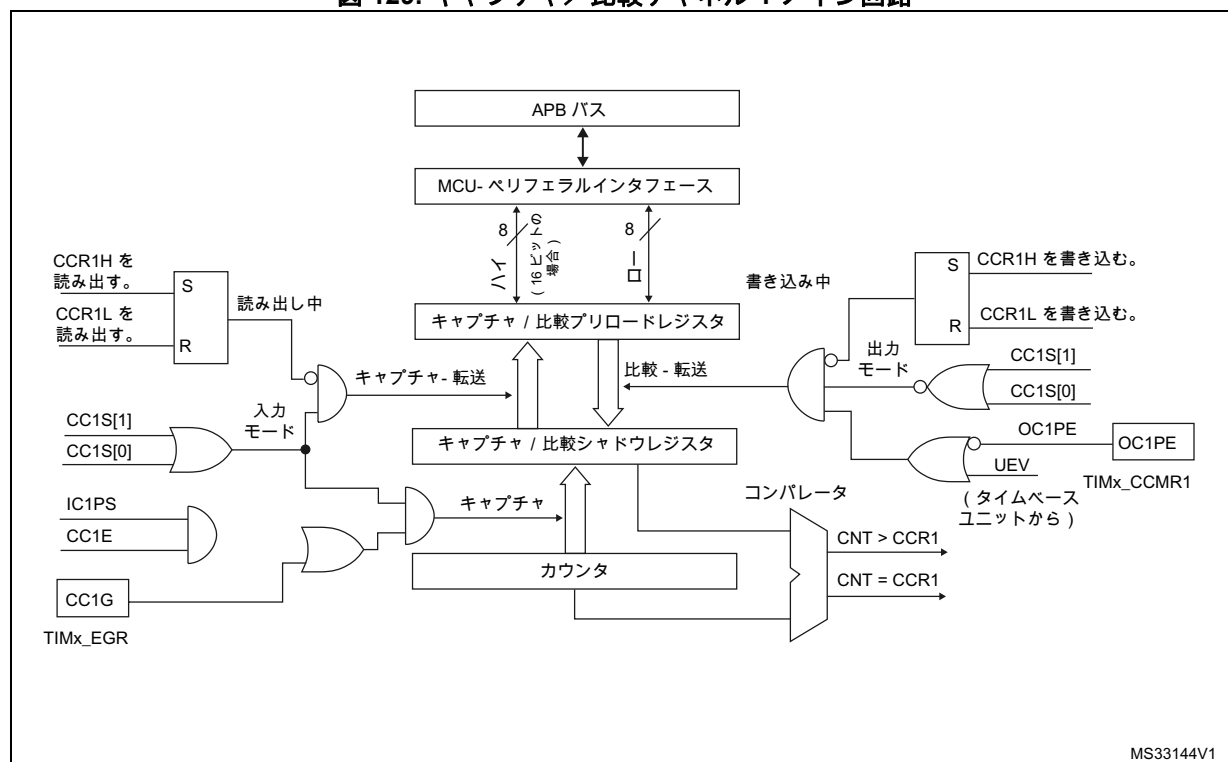
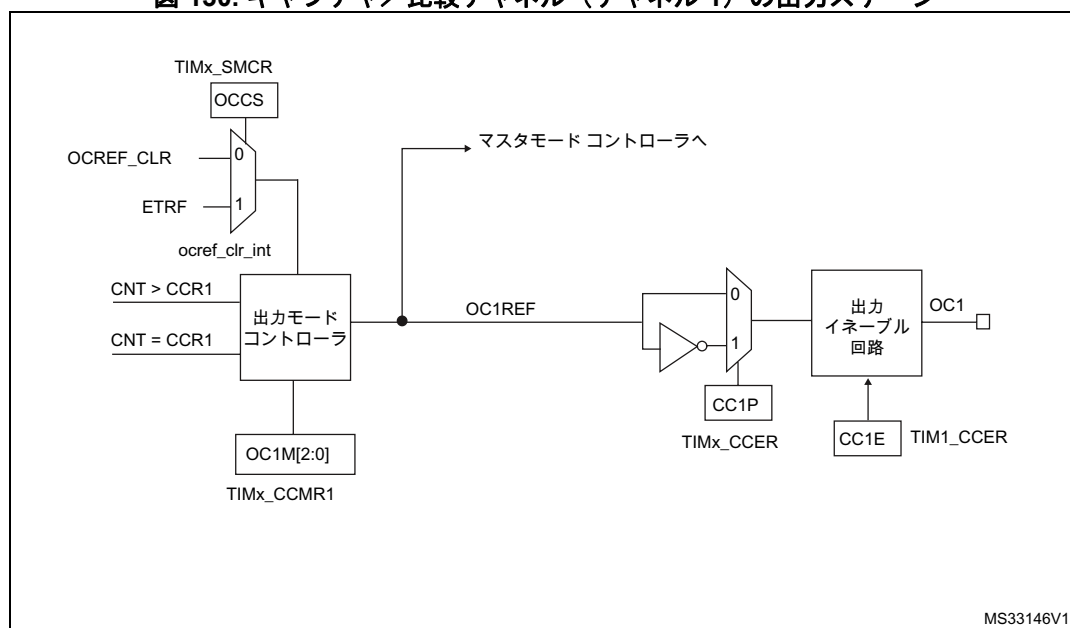


図 130. キャプチャ／比較チャンネル (チャンネル 1) の出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

17.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ／比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力が高くなったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならないので、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャンネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“0”を書き込みます (この場合、立ち上がりエッジの選択)。

- 入力プリスケアラをプログラムします。今回の例では、有効な信号変化ごとにキャプチャを行いますので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに 00 を書き込みます)。
- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要な場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

コード例については、「付録」セクション [A.9.2 : 2 回の ETR の立ち上がりエッジごとのアップカウンタのコード例](#) を参照してください。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

コード例については、「付録」セクション [A.9.4 : 入力キャプチャのデータ管理のコード例](#) を参照してください。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注 : IC 割り込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

17.3.6 PWM 入力モード

このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

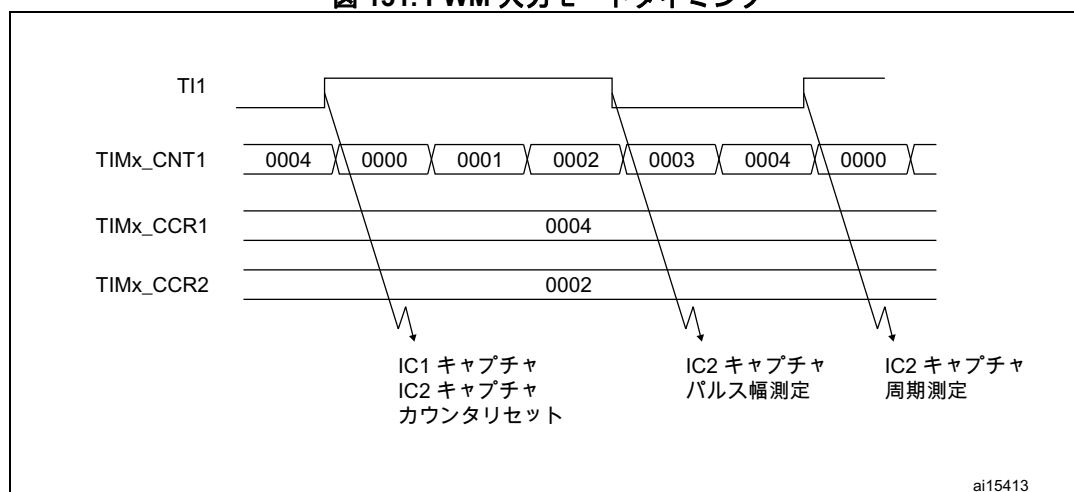
- 2つの ICx 信号が同じ TIx 入力にマッピングされます。
- この2つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2つの TIxFP 信号の1つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

- TIMx_CCMR1 レジスタの CC1S ビットに 01 を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
- CC1P ビットと CC1NP ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブ極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用します)。
- TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
- CC2P ビットと CC2NP ビットに“0”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに 1 を書き込むことによって、キャプチャを有効にします。

コード例については、「付録」セクション [A.9.5 : PWM 入力設定のコード例](#) を参照してください。

図 131. PWM 入力モードタイミング



17.3.7 強制出力モード

このモード (TIMx_CCMRx レジスタの CCxS=00) では、各出力比較信号 (OCxREF、そして OCx) はソフトウェアで強制的にアクティブ、非アクティブのいずれかの状態とされます。これは出力比較レジスタとカウンタの間の比較動作とはかかわりなく行われます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx_OC MRx レジスタの OCxM ビットに 101 を書き込みます。これにより、OCxREF は強制的にハイレベルになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

TIMx_CCMRx レジスタの OCxM ビットに 100 を書き込むことによって、OCxREF 信号を強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、出力比較モードのセクションで説明します。

17.3.8 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。
- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルス出力するためにも使用できます (ワンパルスモード)。

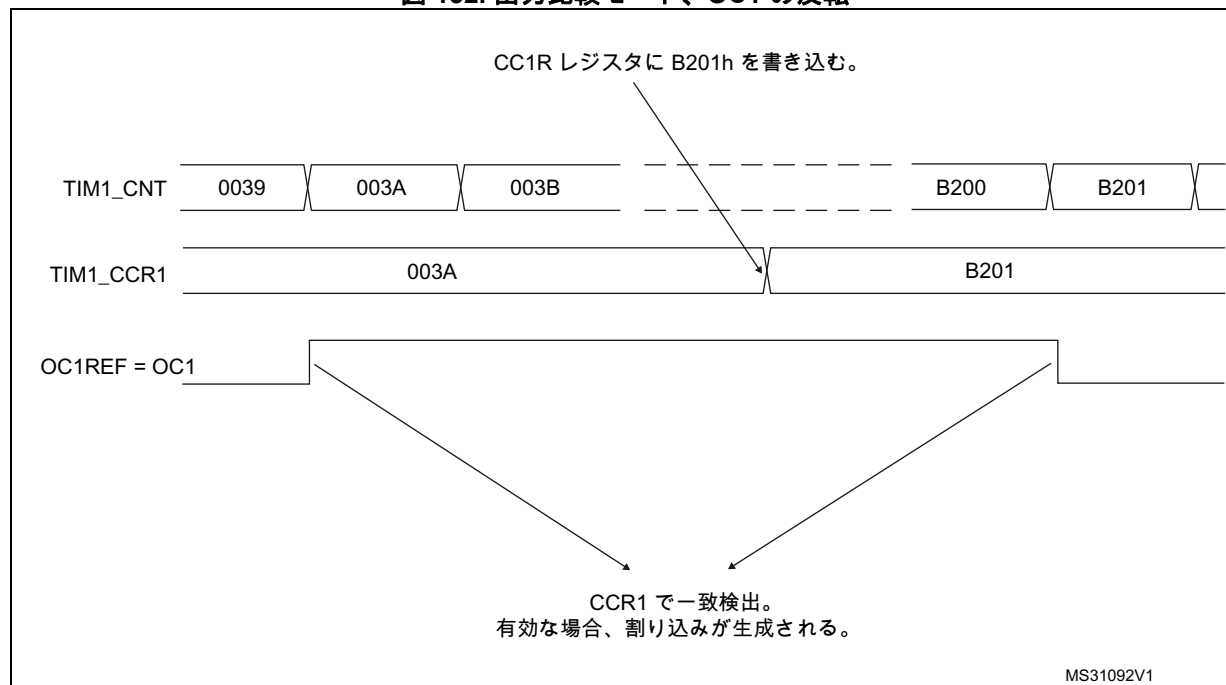
手順:

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエスト/DMA リクエストを生成する場合は、CCxIE ビット/CCxDE ビットをセットします。
4. 出力モードを選択します。たとえば、CNT が CCRx と一致したときに OCx 出力をトグルし、CCRx プリロードを使用せず、OCx が有効でアクティブハイのときには、OCxM=011、OCxPE=0、CCxP=0、CCxE=1 を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

コード例については、「付録」セクション [A.9.7: 入力比較設定のコード例](#) を参照してください。

いつでもソフトウェアで TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限ります (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 132](#) に示します。

図 132. 出力比較モード、OC1 の反転



17.3.9 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、個々のチャネル (OCx 出力ごとに PWM 1 波形) で、TIMx_CCMRx レジスタの OCxM ビットに "110" (PWM モード 1) や "111" (PWM モード 2) を書き込むことで、独自に選択できます。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx がどうか判断されます (カウントの方向によります)。

ただし、OCREF_CLR 機能 (OCREF は、次の PWM 周期までは ETR 信号を通じて外部イベントによってクリアできる) に従って、OCREF 信号は次の場合にのみアサートされます。

- 比較結果が変化するとき、または
- 出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) が停止構成 (比較なし、OCxM=000) から PWM モードの 1 つ (OCxM=110 または 111) へ切り替えられたとき。

タイマの動作中は、ソフトウェアで強制的に PWM になります。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

PWM エッジアラインモード

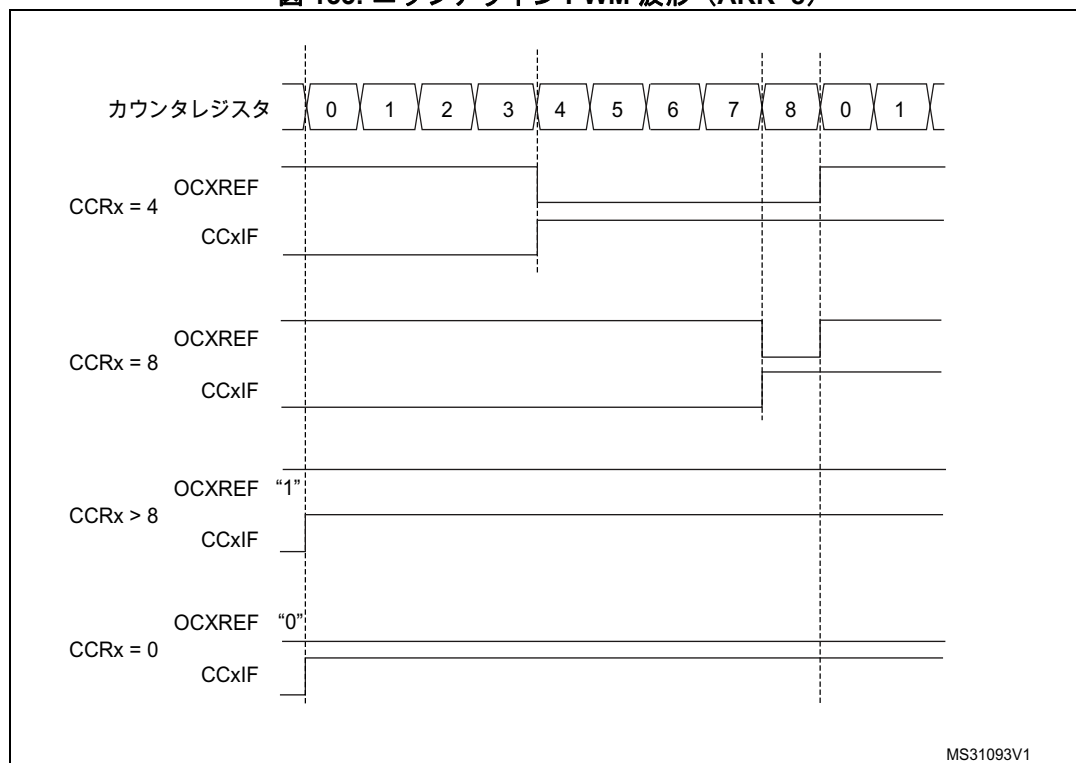
アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。 [セクション : アップカウントモード \(381 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”です。比較値が 0 の場合、OCxREF は“0”に保持されます。 [図 133](#) に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

コード例については、「付録」セクション [A.9.9 : センターアライン PWM の設定例](#) を参照してください。

図 133. エッジアライン PWM 波形 (ARR=8)



ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。 [ダウンカウントモード \(384 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

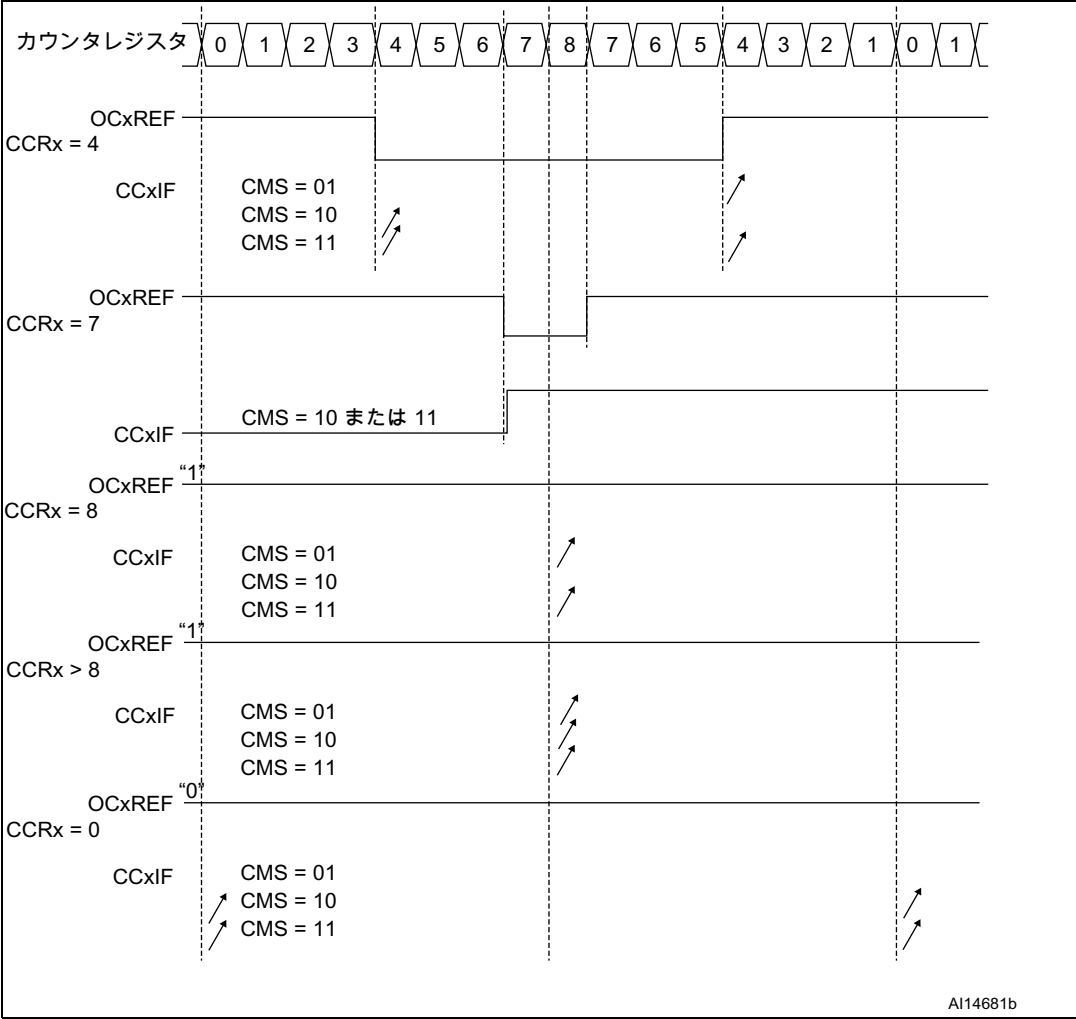
PWM センターアラインモード

センターアラインモードは、TIMx_CR1 レジスタの CMS ビットが“00”でないときにアクティブです (その他すべての設定は、OCxRef/OCx 信号に対して同じ効果を持ちます)。比較フラグは、CMS ビットの設定に応じて、カウンタがカウントアップ、カウントダウン、またはカウントアップとカウントダウンしているときにセットされます。TIMx_CR1 レジスタの方向ビット (DIR) はハードウェアによって更新されており、ソフトウェアで値を変更することはできません。センターアラインモード (アップ/ダウンカウント) (388 ページ) を参照してください。

図 134 に、次の条件でのセンターアライン PWM 波形の例を示します。

- TIMx_ARR=8
- PWM モードは PWM モード 1
- フラグは、TIMx_CR1 レジスタの CMS=01 で選択されたセンターアラインモード 1 に対応して、カウンタがカウントダウンするときにセットされます。

図 134. センターアライン PWM 波形 (ARR=8)



センターアラインモードの使用に関するヒント：

- センターアラインモードを開始するときには、現在のアップ/ダウン設定が使用されます。これは、TIMx_CR1 レジスタの DIR ビットに書き込まれた値に応じて、カウンタがカウントアップまたはカウントダウンすることを意味します。さらに、DIR ビットと CMS ビットをソフトウェアによって同時に変更することはできません。
- センターアラインモードで動作中のカウンタへの書き込みは、予期しない結果を招くことがあるので推奨されません。特に、
 - 自動再ロード値より大きい値をカウンタに書き込んだ場合 (TIMx_CNT > TIMx_ARR)、方向は更新されません。たとえば、カウンタがカウントアップしていた場合、カウンタはカウントアップを続けます。
 - カウンタに 0 または TIMx_ARR 値が書き込まれた場合、方向は更新されますが、更新イベント UEV は生成されません。
- センターアラインモードを使用する最も安全な方法は、カウンタを開始する直前に、ソフトウェアによって更新を生成して (TIMx_EGR レジスタの UG ビットをセットする)、動作中はカウンタへの書き込みを行わないことです。

17.3.10 ワンパルスモード

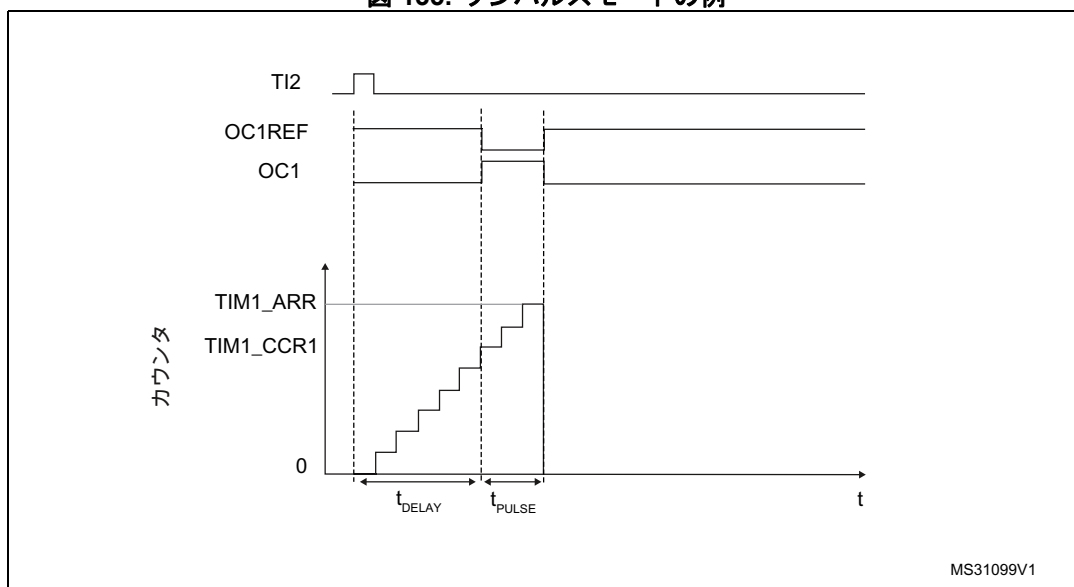
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 135. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタに CC2S=01 を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで CC2P=0 と CC2NP=0 を書き込みます。
- TI2FP2 をスレープモードコントローラのトリガ (TRGI) として構成するために、TIMx_SMCR レジスタの TS="110"を書き込みます。
- TI2FP2 を使用してカウンタを開始するために、TIMx_SMCR レジスタの SMS ビットに"110" (トリガモード) を書き込みます。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます (クロック周波数とカウンタプリスケアラを考慮に入れて)。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1 + 1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。オプションで、TIMx_CCMR1 レジスタの OC1PE=1 と TIMx_CR1 レジスタの ARPE=1 を書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイベントを待ちます。この例では、CC1P に"0"を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

コード例については、「付録」セクション [A.9.16: ワンパルスモードのコード例](#) を参照してください。

必要なパルスは 1 つだけなので (シングルモード)、TIMx_CR1 レジスタの OPM ビットに"1"を書き込みます。こうすると、カウンタは次の更新イベント時に停止します (カウンタが自動再ロード値に達して、"0"に戻る時点)。TIMx_CR1 レジスタの OPM ビットが"0"にセットされると、繰り返しモードが選択されます。

特殊なケース : OCx 高速イネーブル

ワンパルスモードでは、TIMx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF (および OCx) は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

コード例については、「付録」セクション [A.9.16: ワンパルスモードのコード例](#) に記載された PULSE_WITHOUT_DELAY > 0 によって条件付けられたコードの一部を参照してください。

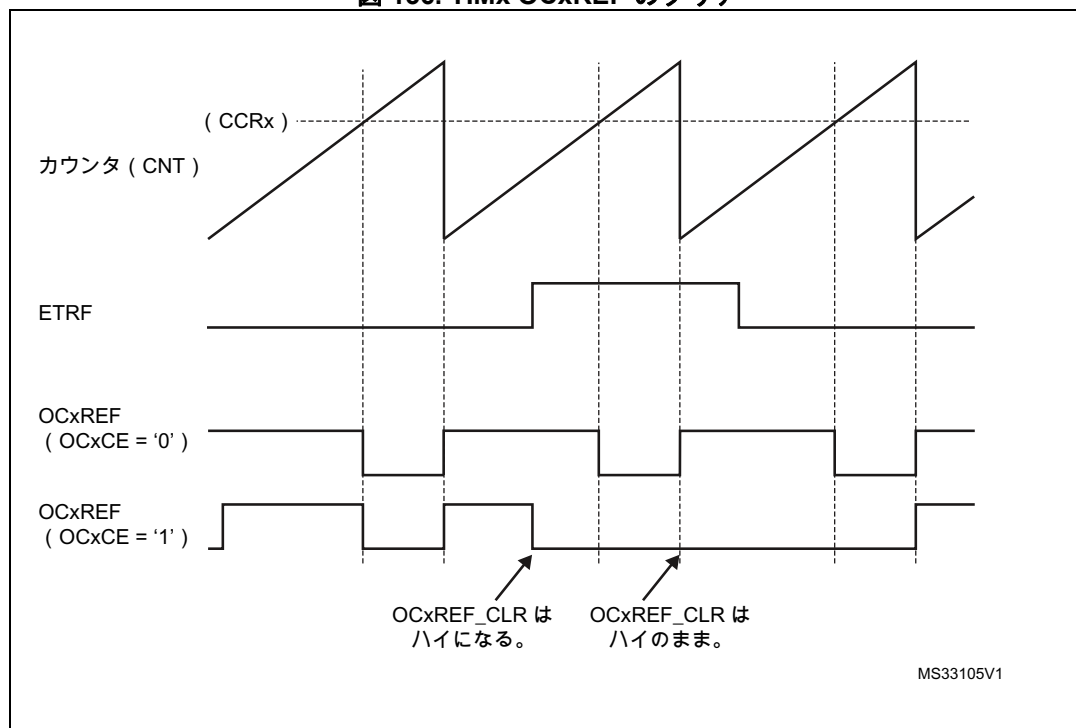
17.3.11 外部イベントによる OCxREF 信号のクリア

1. 外部トリガプリスケアラをオフに保つ必要があります。すなわち、TIMx_SMCR レジスタのビット ETPS[1:0] が 00 にクリアされます。
2. 外部クロックモード 2 を無効にする必要があります。すなわち、TIM1_SMCR レジスタのビット ECE が 0 にクリアされます。
3. 外部トリガ極性 (ETP) と外部トリガフィルタ (ETF) は、アプリケーションのニーズに応じて設定できます。

コード例については、「付録」セクション [A.9.10 : OCxREF をクリアするための ETR 設定のコード例](#) を参照してください。

[図 136](#) に、OCxCE イネーブルビットの両方の値について、ETRF 入力が高くなったときの OCxREF 信号の動作を示します。この例では、TIMx タイマは PWM モードにプログラミングされています。

図 136. TIMx OCxREF のクリア



1. 100% デューティサイクルの PWM の場合 (CCRx > ARR の場合)、次のカウンタオーバーフローで OCxREF が再度有効になります。

17.3.12 エンコーダインタフェースモード

エンコーダインタフェースモードを選択するには、TIMx_SMCR レジスタで、カウンタが TI2 エッジのみをカウントしている場合は SMS="001" を、TI1 エッジのみをカウントしている場合は SMS="010" を、TI1 と TI2 の両方のエッジをカウントしている場合は SMS="011" を書き込みます。

TI1 と TI2 の極性を選択するには、TIMx_CCER レジスタの CC1P ビットと CC2P ビットをプログラミングします。CC1NP と CC2NP はクリア状態に維持する必要があります。必要なときには、入力フィルタもプログラミングできます。

2つの入力 TI1 と TI2 は、インクリメンタルエンコーダとのインタフェースに使用されます。[表 57](#)を参照してください。カウンタのクロックは、TI1FP1 または TI2FP2（入力フィルタおよび極性選択後は TI1 および TI2。フィルタされず、反転されない場合は TI1FP1=TI1、フィルタされず、反転されない場合は TI2FP2=TI2）の有効な変化によって駆動されます。ただし、カウンタ有効なことが前提となります（TIMx_CR1 レジスタの CEN ビットが“1”）。2つの入力の遷移シーケンスが評価されて、カウントパルスと方向信号を生成します。シーケンスに応じて、カウンタはカウントアップまたはカウントダウンし、TIMx_CR1 レジスタの DIR ビットがハードウェアによって変更されます。カウンタが TI1 のみ、TI2 のみ、または TI1 と TI2 の両方をカウントしている場合でも、DIR ビットは、いずれかの入力（TI1 または TI2）の遷移のたびに計算されます。

エンコーダインタフェースモードは、方向選択を含む外部クロックとして動作します。カウンタは、0 と TIMx_ARR レジスタの自動再ロード値の間で連続的にカウントします（方向に応じて、0 から ARR まで、または ARR から 0 まで）。したがって、開始前に TIMx_ARR を設定する必要があります。同様に、キャプチャ、比較、プリスケアラ、およびトリガ出力機能は、通常動作を続けます。

このモードでは、カウンタはインクリメンタルエンコーダの速度と方向に応じて自動的に変更されます。したがって、カウンタの内容は、常にエンコーダの位置を表します。カウンタの方向は、接続されているセンサの回転方向に対応します。次の表は、カウント方向とエンコーダ信号の可能な組み合わせを示します（TI1 と TI2 は同時に切り替わらないと想定しています）。

表 57. カウント方向とエンコーダ信号

アクティブエッジ	他方の信号のレベル (TI2 に対する TI1FP1、 TI1 に対する TI2FP2)	TI1FP1 信号		TI2FP2 信号	
		立ち上がり	立ち下がり	立ち上がり	立ち下がり
TI1 のみカウント	高	ダウン	アップ	カウントなし	カウントなし
	低	アップ	ダウン	カウントなし	カウントなし
TI2 のみカウント	高	カウントなし	カウントなし	アップ	ダウン
	低	カウントなし	カウントなし	ダウン	アップ
TI1 と TI2 の両方を カウント	高	ダウン	アップ	アップ	ダウン
	低	アップ	ダウン	ダウン	アップ

外部インクリメンタルエンコーダは、外部インタフェースロジックなしに、MCU に直接接続できます。ただし、エンコーダの差分出力をデジタル信号に変換するために、通常、コンパレータが使用されます。これにより、耐ノイズ性が大幅に向上します。機械的なゼロ位置を示す 3 番目のエンコーダ出力は、外部割り込み入力に接続して、カウンタのリセットをトリガできます。

[図 137](#) に、カウント信号の生成と方向制御を含むカウンタの動作例を示します。また、両方のエッジが選択されているときの入力ジッタの補正方法も示します。この状況は、センサの位置が一方のスイッチングポイントの近くにあるときに生じることがあります。下の例では、以下のような設定となっています。

- CC1S= 01（TIMx_CCMR1 レジスタ、TI1FP1 は TI1 に配置）
- CC2S= 01（TIMx_CCMR2 レジスタ、TI2FP2 は TI2 に配置）
- CC1P=0、CC1NP=“0”（TIMx_CCER レジスタ、TI1FP1 非反転、TI1FP1=TI1）
- CC2P=0、CC2NP=“0”（TIMx_CCER レジスタ、TI2FP2 非反転、TI2FP2=TI2）
- SMS=011（TIMx_SMCR レジスタ、両方の入力が立ち上がり立ち下りの両エッジでアクティブ）
- CEN=1（TIMx_CR1 レジスタ、カウンタ有効）

コード例については、「付録」セクション [A.9.10 : OCxREF をクリアするための ETR 設定のコード例](#)を参照してください。

図 137. エンコーダインタフェースモードにおけるカウンタの動作例

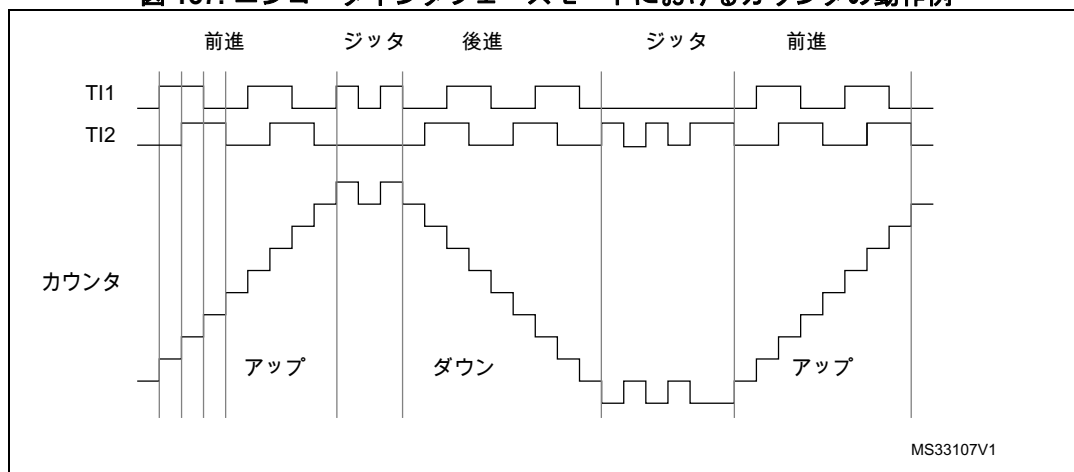
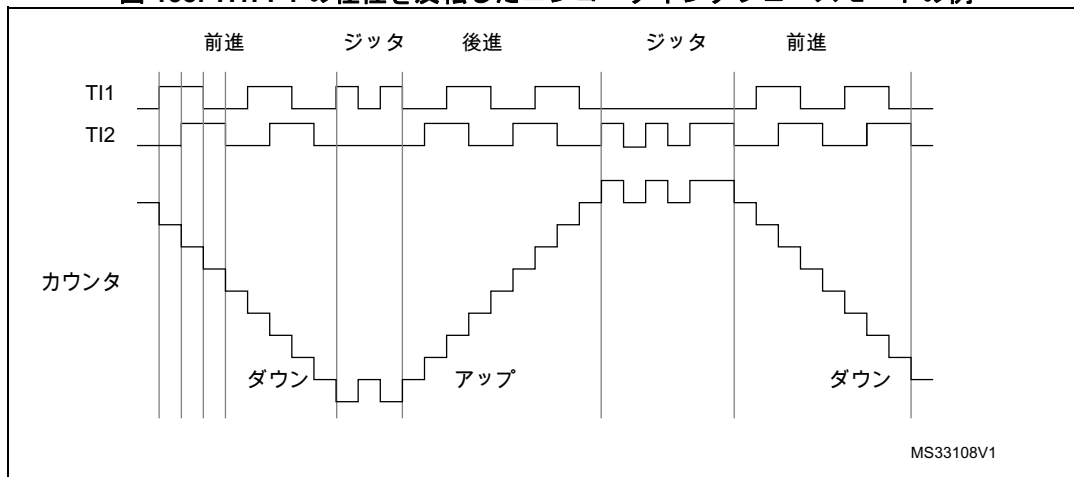


図 138 に、TI1FP1 の極性を反転したときのカウンタの動作を示します（上記と同じ設定ですが、CC1P=1）。

図 138. TI1FP1 の極性を反転したエンコーダインタフェースモードの例



タイマがエンコーダインタフェースモードに設定されている場合、タイマはセンサの現在位置に関する情報を提供します。キャプチャモードに構成した 2 番目のタイマを使用して、2 つのエンコーダイベントの時間差を測定することで、速度、加速度、減速度といった動的な情報を得ることができます。機械的なゼロ位置を示すエンコーダの出力をこの目的に使用できます。2 つのイベントの時間差に応じて、カウンタを定期的に読み出すこともできます。これを行うには、使用可能な場合、カウンタの値を 3 番目の入力キャプチャレジスタにラッチします（キャプチャ信号は周期的でなければならない、別のタイマによって生成できます）。使用可能なときには、リアルタイムクロックによって生成される DMA リクエストを通じて値を読み出すことも可能です。

17.3.13 タイマ入力 XOR 機能

TIM1_CR2 レジスタの TI1S ビットを使用して、チャンネル 1 の入力フィルタを、TIMx_CH1 から TIMx_CH3 までの 3 つの入力ピンを組み合わせさせた XOR ゲートの出力に接続できます。

XOR 出力は、トリガや入力キャプチャなど、すべてのタイマ入力機能で使用できます。

この機能をホールセンサのインタフェースに使用した例を [セクション 16.3.18 \(344 ページ\)](#) に示します。

17.3.14 タイマと外部トリガの同期

TIMx タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレープモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

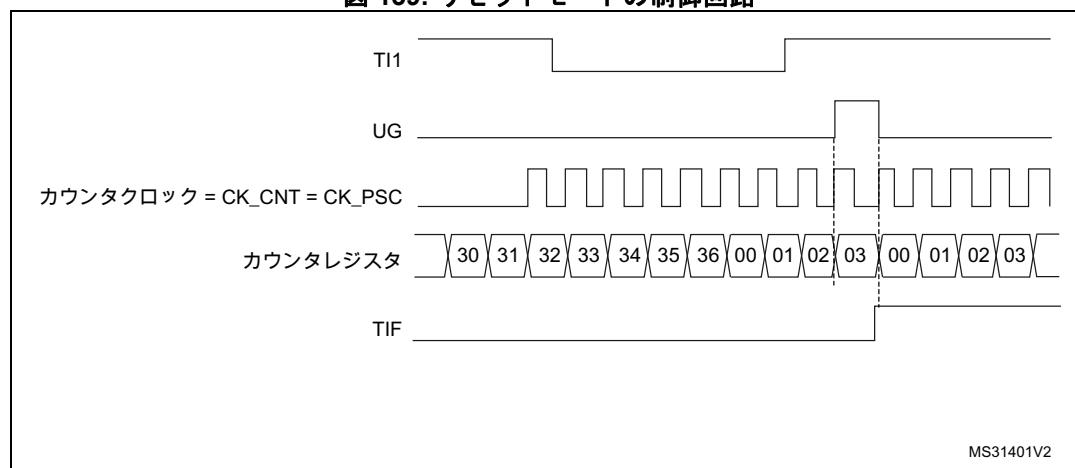
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタで CC1P=0 と CC1NP=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

コード例については、「付録」セクション [A.9.12：リセットモードのコード例](#) を参照してください。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 139. リセットモードの制御回路



スレープモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

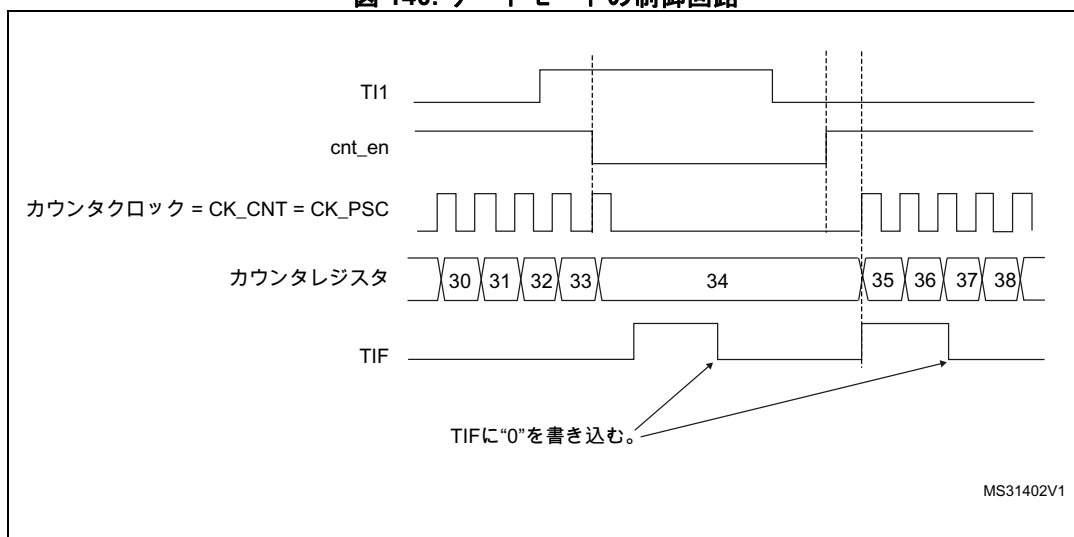
- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S=01 ビット）。TIMx_CCER レジスタで CC1P = 1 と CC1NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします（ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません）。

コード例については、「付録」セクション [A.9.13：ゲートモードのコード例](#) を参照してください。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 140. ゲートモードの制御回路



1. ゲートモードはエッジではなくレベルに対して動作するため、CCxP=CCxNP=1 の設定（立ち上がりと立ち下がり両エッジの検出）はゲートモードでは意味がありません。

スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

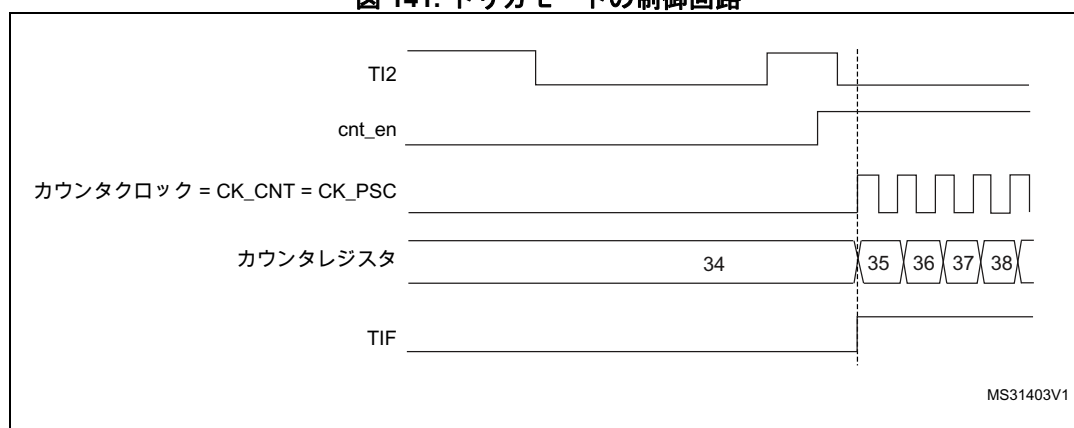
- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケラはトリガには使用されないので、設定は不要です。CC2S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタで CC2P = 1 と CC2NP = 0 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

コード例については、「付録」セクション [A.9.14：トリガモードのコード例](#) を参照してください。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 141. トリガモードの制御回路



スレーブモード：外部クロックモード 2 + トリガモード

外部クロックモード 2 は、他のスレーブモードとともに使用できます（ただし、外部クロックモード 1 とエンコーダモードは除きます）。この場合、ETR 信号は外部クロック入力として使用され、別の入力をトリガ入力として選択できます（リセットモード、ゲートモード、およびトリガモードで動作している場合）。TIMx_SMCR レジスタの TS ビットを通じて TRGI として ETR を選択しないようにしてください。

次の例では、アップカウンタは、TI1 の立ち上がりエッジが発生すると、ETR 信号の立ち上がりエッジのたびにインクリメントされます。

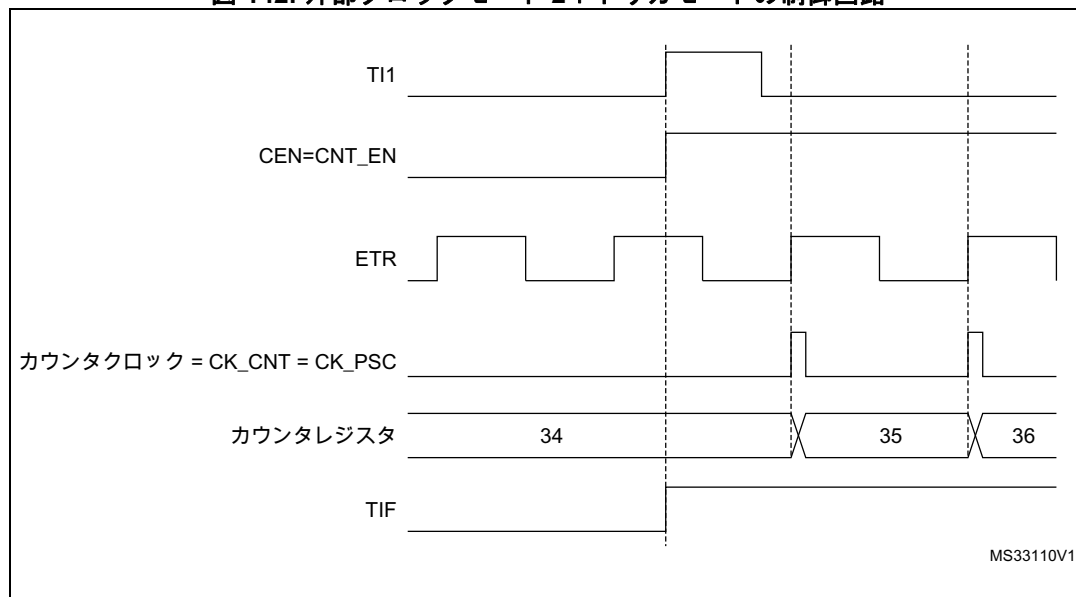
1. TIMx_SMCR レジスタで次のようにプログラミングすることによって、外部トリガ入力回路を構成します。
 - ETF = 0000 : フィルタなし
 - ETPS=00 : プリスケール無効
 - ETP=0 : ETR の立ち上がりエッジを検出。ECE=1 で外部クロックモード 2 を有効にします。
2. TI1 の立ち上がりエッジを検出するように、チャンネル 1 を次のように構成します。
 - IC1F=0000 : フィルタなし。
 - キャプチャプリスケールはトリガには使用されないで、設定する必要はありません。
 - TIMx_CCMR1 レジスタの CC1S=01 で、入力キャプチャソースのみを選択します。
 - TIMx_CCER レジスタの CC1P=0 と CC1NP=0 で、極性を有効にします（そして、立ち上がりエッジのみを検出します）。
3. TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。

コード例については、「付録」セクション [A.9.15 : 外部クロックモード 2 + トリガモードのコード例](#) を参照してください。

TI1 の立ち上がりエッジでカウンタが有効になり、TIF フラグがセットされます。カウンタは、ETR の立ち上がりエッジでカウントします。

ETR 信号の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、ETRP 入力の再同期回路によるものです。

図 142. 外部クロックモード 2+トリガモードの制御回路



タイマを使用して別のタイマを有効にする

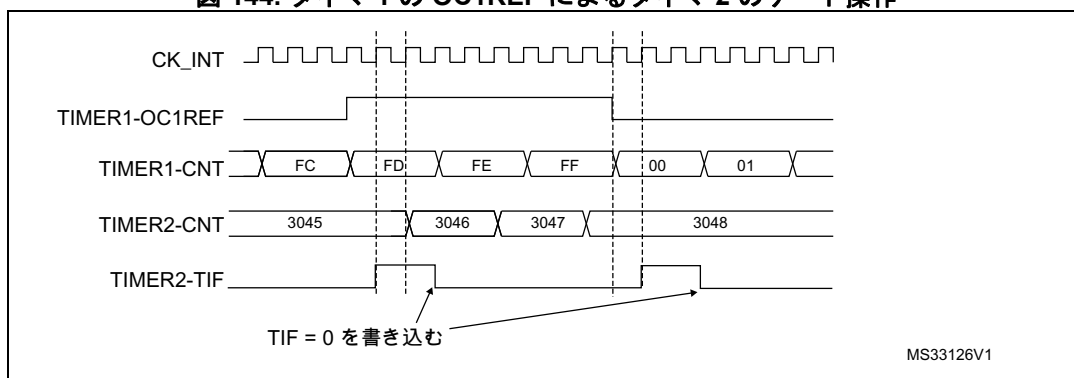
この例では、タイマ 1 の出力比較 1 でタイマ 2 の有効／無効を制御します。接続については、[図 143](#) を参照してください。タイマ 2 は、タイマ 1 の OC1REF がハイのときにのみ、分周された内部クロックをカウントします。両方のカウンタクロック周波数は、CK_INT をプリスケアラで 3 分周したものです ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

- タイマ 1 をマスタモードに設定して、スレーブタイマを有効化します (TIM1_CR2 レジスタの MMS=001)。
- タイマ 1 の OC1REF 波形を設定します (TIM1_CCMR1 レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をゲートモードに設定します (TIM2_SMCR レジスタの SMS=101)。
- CEN ビット (TIM2_CR1 レジスタ) に“1”を書き込んで、タイマ 2 を有効にします。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。

コード例については、「付録」セクション [A.9.18 : 別のタイマを有効にするタイマのコード例](#) を参照してください。

注 : カウンタ 2 のクロックはカウンタ 1 と同期しないので、このモードはタイマ 2 カウンタのイネーブル信号にのみ影響を与えます。

図 144. タイマ 1 の OC1REF によるタイマ 2 のゲート操作



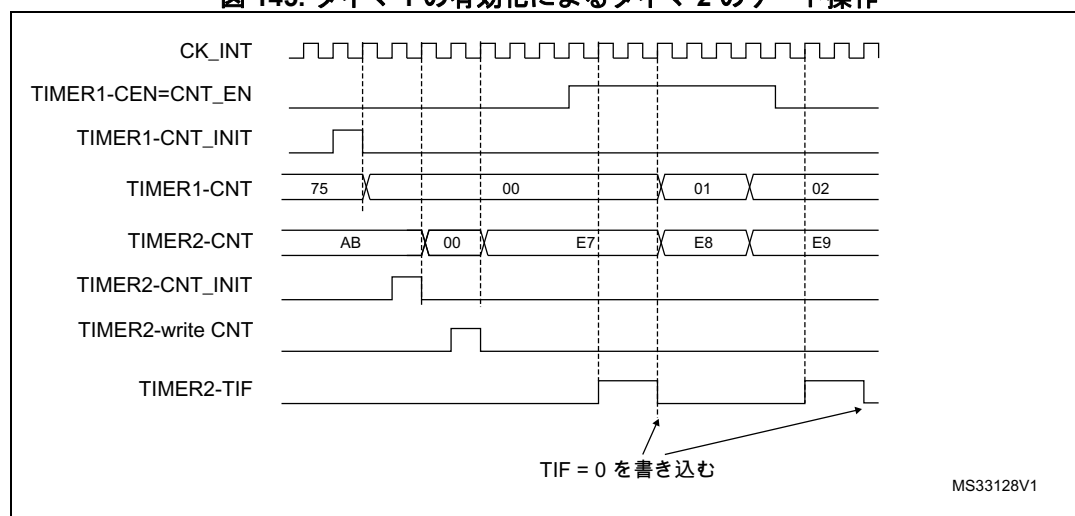
[図 144](#) の例では、タイマ 2 のカウンタとプリスケアラは、開始前に初期化されていません。したがって、現在値からカウントを開始します。タイマ 1 を開始する前に両方のタイマをリセットすることによって、特定の値から開始することが可能です。この場合、タイマカウンタに任意の値を書き込むことができます。TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで容易にタイマをリセットできます。

次の例では、タイマ 1 とタイマ 2 を同期させます。タイマ 1 はマスタであり、0 からカウントを開始します。タイマ 2 はスレーブであり、0xE7 から開始します。プリスケアラの分周比は両方のタイマで同じです。TIM1_CR1 レジスタの CEN ビットに“0”を書き込むことによってタイマ 1 を無効にすると、タイマ 2 は停止します。

- タイマ 1 をマスタモードに設定して、そのカウンタインエーブル信号 (CNT_EN) をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=001)。
- タイマ 1 の OC1REF 波形を設定します (TIM1_CCMR1 レジスタ)。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をゲートモードに設定します (TIM2_SMCR レジスタの SMS=101)。
- UG ビット (TIM1_EGR レジスタ) に“1”を書き込むことによって、タイマ 1 をリセットします。
- UG ビット (TIM2_EGR レジスタ) に“1”を書き込んで、タイマ 2 をリセットします。
- タイマ 2 のカウンタ (TIM2_CNT) に “0xE7” を書き込むことによって、タイマ 2 を 0xE7 に初期化します。
- CEN ビット (TIM2_CR1 レジスタ) に“1”を書き込んで、タイマ 2 を有効にします。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。
- CEN ビット (TIM1_CR1 レジスタ) に“0”を書き込むことによって、タイマ 1 を停止します。

コード例については、「付録」セクション [A.9.19: マスタとスレーブの同期のコード例](#) を参照してください。

図 145. タイマ 1 の有効化によるタイマ 2 のゲート操作

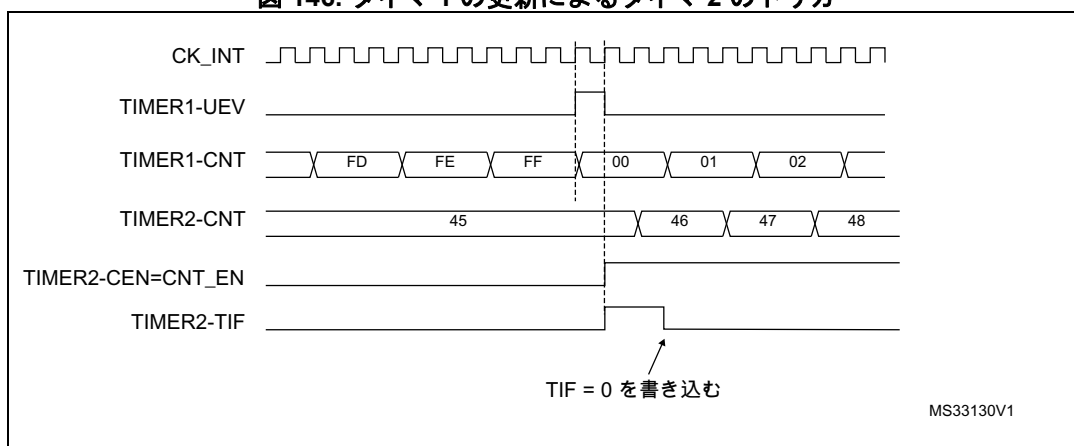


タイマを使用して別のタイマを開始する

この例では、タイマ 1 の更新イベントによってタイマ 2 の有効化を設定します。接続については、[図 143](#) を参照してください。タイマ 1 によって更新イベントが生成されると、タイマ 2 は、分周された内部クロックで現在値（ゼロである必要はありません）からカウントを開始します。タイマ 2 がトリガ信号を受信すると、その CEN ビットが自動的にセットされ、カウンタは TIM2_CR1 レジスタの CEN ビットに“0”が書き込まれるまでカウントします。両方のカウンタクロック周波数は、CK_INT をプリスケアラで 3 分周したものです ($f_{CK\_CNT} = f_{CK\_INT}/3$)。

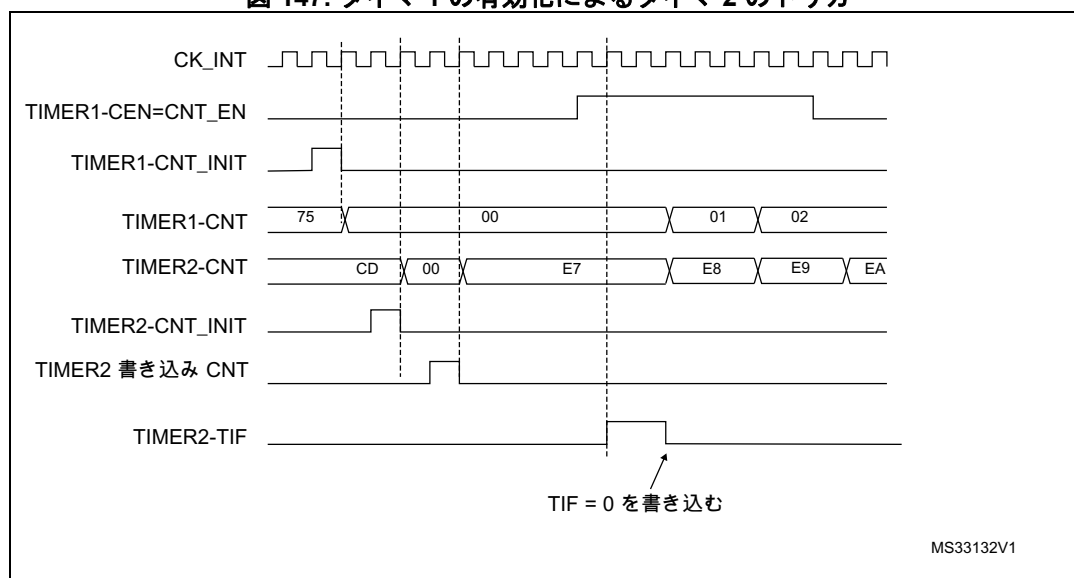
- タイマ 1 をマスタモードに設定して、その更新イベント (UEV) をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=010)。
- タイマ 1 の周期を設定します (TIM1_ARR レジスタ)。
- タイマ 2 がタイマ 1 から入カトリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をトリガモードに設定します (TIM2_SMCR レジスタの SMS=110)。
- CEN ビット (TIM1_CR1 レジスタ) に“1”を書き込んで、タイマ 1 を開始します。

図 146. タイマ 1 の更新によるタイマ 2 のトリガ



前の例と同じように、カウントを開始する前に両方のカウンタを初期化することができます。[図 147](#) は [図 146](#) と同じ設定ですが、ゲートモードではなくトリガモードでの動作を示します (TIM2_SMCR レジスタの SMS=110)。

図 147. タイマ 1 の有効化によるタイマ 2 のトリガ



外部トリガに対応して 2 つのタイマを同期して開始します。

この例では、TI1 入力の立ち上がりエッジでタイマ 1 を有効にし、タイマ 1 が有効になると、タイマ 2 を有効にします。接続については、[図 143](#) を参照してください。カウンタの同時性を確保するため、タイマ 1 はマスタ/スレーブモードに設定する必要があります (TI1 に対してはスレーブ、タイマ 2 に対してはマスタ)。

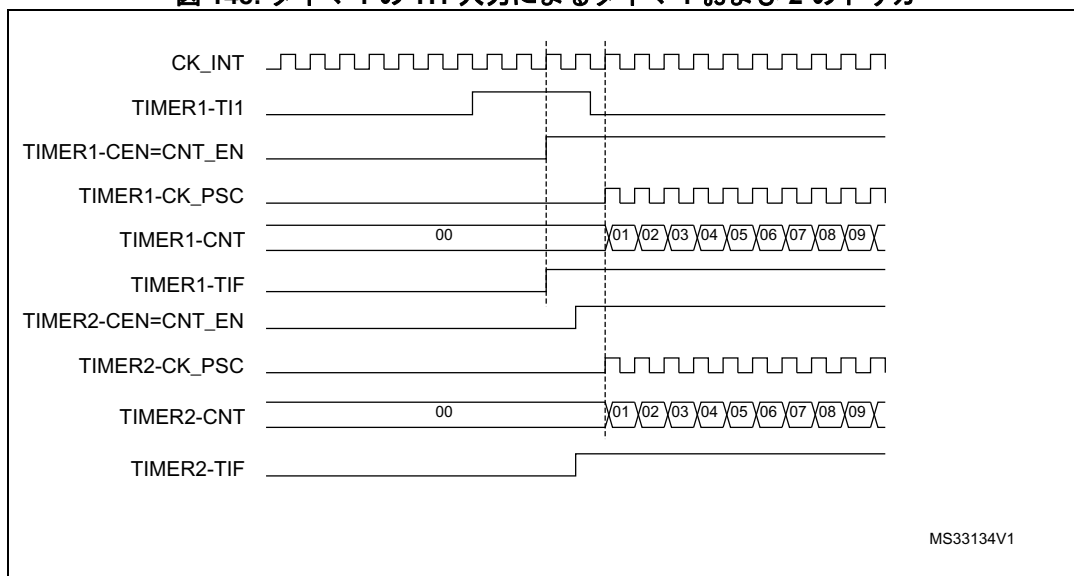
- タイマ 1 をマスタモードに設定して、その有効化をトリガ出力として送信します (TIM1_CR2 レジスタの MMS=001)。
- タイマ 1 をスレーブモードに設定して、TI1 から入力トリガを受け取るようにします (TIM1_SMCR レジスタの TS=100)。
- タイマ 1 をトリガモードに設定します (TIM1_SMCR レジスタの SMS=110)。
- MSM=1 (TIM1_SMCR レジスタ) を書き込むことによって、タイマ 1 をマスタ/スレーブモードに設定します。
- タイマ 2 がタイマ 1 から入力トリガを受け取るように設定します (TIM2_SMCR レジスタの TS=000)。
- タイマ 2 をトリガモードに設定します (TIM2_SMCR レジスタの SMS=110)。

コード例については、「付録」セクション [A.9.20: 外部トリガによって同期された 2 つのタイマのコード例](#) を参照してください。

TI1 (タイマ 1) で立ち上がりエッジが発生すると、両方のカウンタが同時に内部クロックによるカウントを開始し、両方の TIF フラグがセットされます。

注： この例では、両方のタイマが開始前に初期化されます (それぞれの UG ビットをセットすることによって)。両方のカウンタは 0 から開始しますが、カウンタレジスタ (TIMx_CNT) に書き込むことによって、容易にオフセットを挿入できます。マスタ/スレーブモードでは、タイマ 1 の CNT_EN と CK_PSC の間に遅延が挿入されます。

図 148. タイマ 1 の TI1 入力によるタイマ 1 および 2 のトリガ



17.3.16 デバッグモード

マイクロコントローラがデバッグモードになると（ARM® Cortex®-M0 コアは停止状態）、TIMx カウンタは、DBGMCU モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、停止します。

17.4 TIM2 および TIM3 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

17.4.1 TIM2 および TIM3 制御レジスタ 1（TIM2_CR1 および TIM3_CR1）

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:10 予約済み、常に 0 として読み出されます。

ビット 9:8 **CKD**：クロック分周

このビットフィールドは、タイマクロック（CK_INT）周波数と、デジタルフィルタ（ETR、TlX）によって使用されるサンプリングクロックとの間の分周比を示します。

00: $t_{DTS} = t_{CK\_INT}$
01: $t_{DTS} = 2 \times t_{CK\_INT}$
10: $t_{DTS} = 4 \times t_{CK\_INT}$
11: 予約済み

ビット 7 **ARPE**：自動再ロードプリロードイネーブル

0: TIMx_ARR レジスタはバッファされません。
1: TIMx_ARR レジスタはバッファされます。

ビット 6:5 **CMS**：センタアラインモード選択

00: エッジアラインモードカウンタは、方向ビット（DIR）に応じて、カウントアップまたはカウントダウンします。
01: センタアラインモード 1。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割り込みフラグは、カウンタがカウントダウンしているときのみセットされます。
10: センタアラインモード 2。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割り込みフラグは、カウンタがカウントアップしているときのみセットされます。
11: センタアラインモード 3。カウンタはカウントアップとカウントダウンを交互に行います。出力に設定されたチャネル（TIMx_CCMRx レジスタの CCxS=00）の出力比較割り込みフラグは、カウンタがカウントアップおよびカウントダウンしているときにセットされます。

注： カウンタが有効（CEN=1）なときに、エッジアラインモードからセンタアラインモードに切り替えることはできません。

ビット 4 **DIR**：方向

0: カウンタはアップカウンタとして使用されます。
1: カウンタはダウンカウンタとして使用されます。

注： このビットは、タイマがセンタアラインモードまたはエンコーダモードに設定されているときには読み出し専用です。

ビット 3 **OPM**：ワンパルスモード

0: カウンタは更新イベントで停止しません。
1: カウンタは次の更新イベントでカウントを停止します（CEN ビットをクリア）。

ビット 2 URS : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します（有効な場合）。これらのイベントは、次のとおりです。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー／アンダーフローのみが更新割り込みまたは DMA リクエストを生成します（有効な場合）。

ビット 1 UDIS : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント（UEV）は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 CEN : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注： 外部クロック、ゲートモード、およびエンコーダモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

17.4.2 TIM2 および TIM3 制御レジスタ 2（TIM2_CR2 および TIM3_CR2）

アドレスオフセット：0x04

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]				CCDS	Res.	Res.
								rw	rw	rw	rw	rw			

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TI1S**：TI1 選択

0：TIMx_CH1 ピンが TI1 入力に接続されます。

1：TIMx_CH1、CH2、および CH3 ピンが TI1 入力に接続されます（XOR 接続）。参照先：[セクション 16.3.18：ホールセンサとのインタフェース（344 ページ）](#)

ビット 6:4 **MMS**：マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます（TRGO）。組み合わせは、次のとおりです。

000：**リセット**- TIMx_EGR レジスタの UG ビットがトリガ出力（TRGO）として使用されます。トリガ入力によってリセットが生成される場合（スレーブモードコントローラがリセットモードに設定されているとき）、TRGO 信号は実際のリセットより遅延します。

001：**イネーブル**- カウンタイネーブル信号 CNT_EN がトリガ出力（TRGO）として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和（OR）によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ／スレーブモードが選択されている場合を除き、TRGO には遅延が存在します（TIMx_SMCR レジスタの MSM ビットの説明を参照してください）。

010：**更新**- 更新イベントがトリガ出力（TRGO）として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

011：**パルス比較**- キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされるとき（すでにハイであった場合も）、トリガ出力は正のパルスを送信します。（TRGO）。

100：比較- OC1REF 信号がトリガ出力（TRGO）として使用されます。

101：比較- OC2REF 信号がトリガ出力（TRGO）として使用されます。

110：比較- OC3REF 信号がトリガ出力（TRGO）として使用されます。

111：比較- OC4REF 信号がトリガ出力（TRGO）として使用されます。

ビット 3 **CCDS**：キャプチャ／比較 DMA 選択

0：CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1：CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2:0 予約済み、常に 0 として読み出されます。

17.4.3 TIM2 および TIM3 のスレーブモード制御レジスタ（TIM2_SMCR および TIM3_SMCR）

アドレスオフセット：0x08

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]		OCCS	SMS[2:0]			
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

- ビット 15 **ETP**：外部トリガ極性
このビットは、ETR と $\overline{\text{ETR}}$ のいずれがトリガ動作に使用されるかを選択します。
0：ETR は反転されず、ハイレベルまたは立ち上がりエッジでアクティブになります。
1：ETR は反転され、ローレベルまたは立ち下がりエッジでアクティブになります。
- ビット 14 **ECE**：外部クロックイネーブル
このビットは、外部クロックモード 2 を有効にします。
0：外部クロックモード 2 は無効です。
1：外部クロックモード 2 は有効です。カウンタは、ETRF 信号のアクティブエッジによってクロック供給されます。
1：ECE ビットをセットすることは、TRGI が ETRF に接続された状態で外部クロックモード 1 を選択することと同じ効果があります（SMS=111、TS=111）。
2：外部クロックモード 2 と次のスレーブモード、すなわち、リセットモード、ゲートモード、またはトリガモードを同時に使用することができます。ただし、この場合、TRGI を ETRF に接続することはできません（TS ビットが 111 でないことが必要）。
3：外部クロックモード 1 と外部クロックモード 2 が同時に有効な場合、外部クロック入力 は ETRF です。
- ビット 13:12 **ETPS**：外部トリガプリスケアラ
外部トリガ信号 ETRP の周波数は、最大でも CK_INT 周波数の 1/4 でなければなりません。プリスケアラを有効にすると、ETRP 周波数を低減できます。これは、高速な外部クロックを入力するときに役立ちます。
00：プリスケアラオフ
01：ETRP 周波数は 2 分周されます。
10：ETRP 周波数は 4 分周されます。
11：ETRP 周波数は 8 分周されます。

ビット 11:8 **ETF[3:0]** : 外部トリガフィルタ

このビットフィールドは、ETRP 信号をサンプルする周波数と、ETRP に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$ 、 $N = 2$

0010 : $f_{SAMPLING} = f_{CK\_INT}$ 、 $N = 4$

0011 : $f_{SAMPLING} = f_{CK\_INT}$ 、 $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、 $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、 $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、 $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、 $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、 $N = 8$

注 : **ETF[3:0] = 1、2、または 3 のときに、式の f_{DTS} が CK_INT に置き換えられるよう注意する必要があります。**

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)
- 100 : TI1 エッジ検出回路 (TI1F_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)
- 111 : 外部トリガ入力 (ETRF)

各タイマでの ITRx の詳細については、表 58 : TIM2 および TIM3 内部トリガ接続 (425 ページ) を参照してください。

注： 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 **OCCS** : OCREF クリア選択

このビットは、OCREF クリアソースを選択するために使用されます。

- 0 : OCREF_CLR_INT は、OCREF_CLR 入力に接続されています。
- 1 : OCREF_CLR_INT は、ETRF に接続されています。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号 (TRGI) のアクティブエッジが外部入力で選択された極性にリンクされます (入力制御レジスタおよび制御レジスタの説明を参照してください)。

- 000 : スレーブモードは無効です。CEN = "1" の場合、プリスケアラは内部クロックによって直接クロック供給されます。
- 001 : エンコーダモード 1 - カウンタは、TI1FP2 のレベルに応じて、TI2FP1 のエッジでカウントアップ/ダウンします。
- 010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。
- 011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。
- 100 : リセットモード - 選択されたトリガ入力 (TRGI) の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。
- 101 : ゲートモード - カウンタクロックは、トリガ入力 (TRGI) がハイのときに有効になります。トリガがローになると、カウンタは停止します (リセットはされません)。カウンタの開始と停止の両方が制御されます。
- 110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します (リセットはされません)。カウンタの開始のみが制御されます。
- 111 : 外部クロックモード 1 - 選択されたトリガ (TRGI) の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合 (TS=100)、ゲートモードを使用することはできません。TI1F_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

注： スレーブタイマのクロックは、マスタタイマからイベントを受信する前に有効化する必要があります、マスタタイマからトリガを受信している間は動作中に変更しないでください。

表 58. TIM2 および TIM3 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM15	TIM3	TIM14
TIM3	TIM1	TIM2	TIM15	TIM14

17.4.4 TIM2 および TIM3 DMA/割り込み有効レジスタ（TIM2_DIER および TIM3_DIER）

アドレスオフセット：0x0C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14 **TDE**：トリガ DMA リクエストイネーブル

- 0：トリガ DMA リクエストは無効です。
- 1：トリガ DMA リクエストは有効です。

ビット 13 予約済み。常に 0 として読み出されます。

ビット 12 **CC4DE**：キャプチャ／比較 4 DMA リクエストイネーブル

- 0：CC4 DMA リクエストは無効です。
- 1：CC4 DMA リクエストは有効です。

ビット 11 **CC3DE**：キャプチャ／比較 3 DMA リクエストイネーブル

- 0：CC3 DMA リクエストは無効です。
- 1：CC3 DMA リクエストは有効です。

ビット 10 **CC2DE**：キャプチャ／比較 2 DMA リクエストイネーブル

- 0：CC2 DMA リクエストは無効です。
- 1：CC2 DMA リクエストは有効です。

ビット 9 **CC1DE**：キャプチャ／比較 1 DMA リクエストイネーブル

- 0：CC1 DMA リクエストは無効です。
- 1：CC1 DMA リクエストは有効です。

ビット 8 **UDE**：更新 DMA リクエストイネーブル

- 0：更新 DMA リクエストは無効です。
- 1：更新 DMA リクエストは有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TIE**：トリガ割り込みイネーブル

- 0：トリガ割り込みは無効です。
- 1：トリガ割り込みは有効です。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4IE**：キャプチャ／比較 4 割り込みイネーブル

- 0：CC4 割り込みは無効です。
- 1：CC4 割り込みは有効です。

ビット 3 **CC3IE**：キャプチャ／比較 3 割り込みイネーブル

- 0：CC3 割り込みは無効です。
- 1：CC3 割り込みは有効です。



ビット 2 **CC2IE** : キャプチャ／比較 2 割り込みイネーブル

- 0 : CC2 割り込みは無効です。
- 1 : CC2 割り込みは有効です。

ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル

- 0 : CC1 割り込みは無効です。
- 1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

- 0 : 更新割り込みは無効です。
- 1 : 更新割り込みは有効です。

17.4.5 TIM2 および TIM3 制御レジスタ（TIM2_SR および TIM3_SR）

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

ビット 15:13 予約済み、常に 0 として読み出されます。

ビット 12 **CC4OF** : キャプチャ／比較 4 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 11 **CC3OF** : キャプチャ／比較 3 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 10 **CC2OF** : キャプチャ／比較 2 オーバーキャプチャフラグ

CC1OF の説明を参照してください。

ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。ソフトウェアで“0”を書き込むことによってクリアされます。

- 0 : オーバキャプチャは検出されていません。
- 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:7 予約済み、常に 0 として読み出されます。

ビット 6 **TIF** : トリガ割り込みフラグ

このフラグは、トリガイベント時（スレープモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力でアクティブエッジが検出されたとき）にハードウェアによってセットされます。ゲートモードが選択されている場合、カウンタが開始または停止したときにセットされます。ソフトウェアによってクリアされます。

- 0 : トリガイベントは発生していません。
- 1 : トリガ割り込みが保留中です。

ビット 5 予約済み、常に 0 として読み出されます。

ビット 4 **CC4IF** : キャプチャ／比較 4 割り込みフラグ

CC1IF の説明を参照してください。

ビット 3 **CC3IF** : キャプチャ／比較 3 割り込みフラグ

CC1IF の説明を参照してください。



ビット 2 **CC2IF** : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます (センターアラインモードでは、例外もあります。TIMx_CR1 レジスタの CMS ビットの説明を参照してください)。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時 (アップカウントおよびアップ/ダウンカウントモードの場合)、またはアンダーフロー時 (ダウンカウントモードの場合) に CC1IF ビットはハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました (IC1 で、選択された極性に一致するエッジが検出されました)。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。

このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

オーバーフローまたはアンダーフロー時と、TIMx_CR1 レジスタで UDIS=0 の場合。

TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイベントによって CNT が再初期化されたとき (同期制御レジスタの説明を参照)。

17.4.6 TIM2 および TIM3 のイベント生成レジスタ（TIM2_EGR および TIM3_EGR）

アドレスオフセット：0x14

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TG**：トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **CC4G**：キャプチャ／比較 4 生成

CC1G の説明を参照してください。

ビット 3 **CC3G**：キャプチャ／比較 3 生成

CC1G の説明を参照してください。

ビット 2 **CC2G**：キャプチャ／比較 2 生成

CC1G の説明を参照してください。

ビット 1 **CC1G**：キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：チャンネル 1 でキャプチャ／比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合：

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます（有効な場合）。

CC1 チャンネルが入力として設定されている場合：

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます（有効な場合）。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG**：更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0：影響なし。

1：カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。センターアラインモードが選択されている場合、または、DIR=0（カウントアップ）の場合、カウンタはクリアされます。そうでない場合、DIR=1（カウントダウン）であれば、自動再ロード値（TIMx_ARR）をとりま。

17.4.7 TIM2 および TIM3 のキャプチャ／比較モードレジスタ 1（TIM2_CCMR1 および TIM3_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]			IC1PSC[1:0]				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

出力比較モード

- ビット 15 **OC2CE** : 出力比較 2 クリアイネーブル
- ビット 14:12 **OC2M[2:0]** : 出力比較 2 モード
- ビット 11 **OC2PE** : 出力比較 2 プリロードイネーブル
- ビット 10 **OC2FE** : 出力比較 2 高速イネーブル
- ビット 9:8 **CC2S[1:0]** : キャプチャ／比較 2 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC2 チャンネルは出力として設定されます。
 - 01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。
 - 10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。
 - 11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。
- 注：** **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。
- ビット 7 **OC1CE** : 出力比較 1 クリアイネーブル
- OC1CE : 出力比較 1 クリアイネーブル
- 0 : OC1Ref は ETRF 入力の影響を受けません。
 - 1 : OC1Ref は ETRF 入力のハイレベルが検出されるとクリアされます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません (このモードはタイミングベースを生成するために使用されます)。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合は非アクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注 : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、CC1S=00 (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このような場合、OC は、比較結果に関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ/比較 2 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC2 チャネルは出力として設定されます。

01 : CC2 チャネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2

0010 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4

0011 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 8

注 : **ICxF[3:0] = 1、2、または 3 のときに、式の f_{DTS} が CK_INT に置き換えられるよう注意する必要があります。**

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E=0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ/比較 1 選択

このビットフィールドは、チャネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャネルは出力として設定されます。

01 : CC1 チャネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

17.4.8 TIM2 および TIM3 のキャプチャ／比較モードレジスタ 2（TIM2_CCMR2 および TIM3_CCMR2）

アドレスオフセット：0x1C

リセット値：0x0000

上記の CCMR1 レジスタの説明を参照してください。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]			IC3PSC[1:0]				
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

出力比較モード

ビット 15 **OC4CE**：出力比較 4 クリアイネーブル

ビット 14:12 **OC4M**：出力比較 4 モード

ビット 11 **OC4PE**：出力比較 4 プリロードイネーブル

ビット 10 **OC4FE**：出力比較 4 高速イネーブル

ビット 9:8 **CC4S**：キャプチャ／比較 4 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

- 00：CC4 チャンネルは出力として設定されます。
- 01：CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。
- 10：CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。
- 11：CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC4S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC4E=0）のときにのみ書き込み可能です。

ビット 7 **OC3CE**：出力比較 3 クリアイネーブル

ビット 6:4 **OC3M**：出力比較 3 モード

ビット 3 **OC3PE**：出力比較 3 プリロードイネーブル

ビット 2 **OC3FE**：出力比較 3 高速イネーブル

ビット 1:0 **CC3S**：キャプチャ／比較 3 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

- 00：CC3 チャンネルは出力として設定されます。
- 01：CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。
- 10：CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。
- 11：CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： **CC3S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC3E=0）のときにのみ書き込み可能です。



入力キャプチャモード

- ビット 15:12 **IC4F** : 入力キャプチャ 4 フィルタ
- ビット 11:10 **IC4PSC** : 入力キャプチャ 4 プリスケアラ
- ビット 9:8 **CC4S** : キャプチャ／比較 4 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC4 チャンネルは出力として設定されます。
- 01 : CC4 チャンネルは入力として設定され、IC4 は TI4 に配置されます。
- 10 : CC4 チャンネルは入力として設定され、IC4 は TI3 に配置されます。
- 11 : CC4 チャンネルは入力として設定され、IC4 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。
- 注： **CC4S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC4E=0）のときにのみ書き込み可能です。
- ビット 7:4 **IC3F** : 入力キャプチャ 3 フィルタ
- ビット 3:2 **IC3PSC** : 入力キャプチャ 3 プリスケアラ
- ビット 1:0 **CC3S** : キャプチャ／比較 3 選択
- このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。
- 00 : CC3 チャンネルは出力として設定されます。
- 01 : CC3 チャンネルは入力として設定され、IC3 は TI3 に配置されます。
- 10 : CC3 チャンネルは入力として設定され、IC3 は TI4 に配置されます。
- 11 : CC3 チャンネルは入力として設定され、IC3 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。
- 注： **CC3S** ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC3E=0）のときにのみ書き込み可能です。

17.4.9 TIM2 および TIM3 のキャプチャ／比較有効レジスタ（TIM2_CCER および TIM3_CCER）

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

- ビット 15 **CC4NP** : キャプチャ／比較 4 出力極性
- CC1NP の説明を参照してください。
- ビット 14 予約済み、常に 0 として読み出されます。
- ビット 13 **CC4P** : キャプチャ／比較 4 出力極性
- CC1P の説明を参照してください。
- ビット 12 **CC4E** : キャプチャ／比較 4 出カインエーブル。
- CC1E の説明を参照してください。
- ビット 11 **CC3NP** : キャプチャ／比較 3 出力極性
- CC1NP の説明を参照してください。
- ビット 10 予約済み、常に 0 として読み出されます。
- ビット 9 **CC3P** : キャプチャ／比較 3 出力極性
- CC1P の説明を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 8 **CC3E** : キャプチャ／比較 3 出力イネーブル。

CC1E の説明を参照してください。

ビット 7 **CC2NP** : キャプチャ／比較 2 出力極性

CC1NP の説明を参照してください。

ビット 6 予約済み、常に 0 として読み出されます。

ビット 5 **CC2P** : キャプチャ／比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ／比較 2 出力イネーブル。

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

この場合、CC1NP はクリアされたままでなければなりません。

CC1 チャンネルが入力として設定されている場合 :

このビットは、TI1FP1/TI2FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済み、常に 0 として読み出されます。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、トリガまたはキャプチャ操作の TI1FP1 および TI2FP1 の極性を選択します。

00 : 非反転／立ち上がりエッジ

回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ、エンコーダモード)。

01 : 反転／立ち下がりエッジ

回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ、エンコーダモード)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転／両エッジ

回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、外部クロックモード、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。この設定をエンコーダモードに使用することはできません。

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル。

CC1 チャンネルが出力として設定されている場合 :

0 : オフ - OC1 はアクティブではありません。

1 : オン - OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ／比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

表 59. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効（OCx=0、OCx_EN=0）
1	OCx=OCxREF + 極性、OCx_EN=1

注：標準 OCx チャンネルに接続されている外部 IO ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

17.4.10 TIM2 および TIM3 のカウンタ（TIM2_CNT および TIM3_CNT）

アドレスオフセット：0x24

リセット値：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16] (TIM2 のみ)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 CNT[31:16]：カウンタ値上位ビット (TIM2)

ビット 15:0 CNT[15:0]：カウンタ値下位ビット

17.4.11 TIM2 および TIM3 のプリスケアラ（TIM2_PSC および TIM3_PSC）

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値

カウンタクロック周波数 CK_CNT は $f_{CK\_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます。

17.4.12 TIM2 および TIM3 自動再ロードレジスタ（TIM2_ARR および TIM3_ARR）

アドレスオフセット：0x2C

リセット値：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16] (TIM2 のみ)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **ARR[31:16]**：自動再ロード値上位ビット（TIM2）

ビット 15:0 **ARR[15:0]**：自動再ロード値下位ビット

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 17.3.1：タイムベースユニット \(379 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

17.4.13 TIM2 および TIM3 のキャプチャ／比較レジスタ 1（TIM2_CCR1 および TIM3_CCR1）

アドレスオフセット：0x34

リセット値：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16] (TIM2 のみ)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR1[31:16]**：キャプチャ／比較 1 値上位ビット（TIM2）

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値下位ビット

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

17.4.14 TIM2 および TIM3 のキャプチャ／比較レジスタ 2（TIM2_CCR2 および TIM3_CCR2）

アドレスオフセット：0x38

リセット値：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16] (TIM2 のみ)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR2[31:16]**：キャプチャ／比較 2 値上位ビット（TIM2）

ビット 15:0 **CCR2[15:0]**：キャプチャ／比較 2 値下位ビット

CC2 チャンネルが出力として設定されている場合：

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合：

CCR2 は、最後の入力キャプチャ 2 イベント（IC2）によって転送されたカウンタ値です。

17.4.15 TIM2 および TIM3 のキャプチャ／比較レジスタ 3（TIM2_CCR3 および TIM3_CCR3）

アドレスオフセット：0x3C

リセット値：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16] (TIM2 のみ)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR3[31:16]** : キャプチャ／比較 3 値上位ビット（TIM2）

ビット 15:0 **CCR3[15:0]** : キャプチャ／比較 3 値下位ビット

CC3 チャンネルが出力として設定されている場合 :

CCR3 は、実際のキャプチャ／比較 3 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR3 レジスタの OC3PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 3 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC3 出力に送信される値を含みます。

チャンネル CC3 が入力として設定されている場合 :

CCR3 は、最後の入力キャプチャ 3 イベント（IC3）によって転送されたカウンタ値です。

17.4.16 TIM2 および TIM3 のキャプチャ／比較レジスタ 4（TIM2_CCR4 および TIM3_CCR4）

アドレスオフセット : 0x40

リセット値 : 0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16] (TIM2 のみ)															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 **CCR4[31:16]** : キャプチャ／比較 4 値上位ビット（TIM2）

ビット 15:0 **CCR4[15:0]** : キャプチャ／比較 4 値下位ビット

1. CC4 チャンネルが出力として設定されている場合（CC4S ビット） :
CCR4 は、実際のキャプチャ／比較 4 レジスタにロードされる値（プリロード値）です。
TIMx_CCMR4 レジスタの OC4PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 4 レジスタにコピーされます。
アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC4 出力に送信される値を含みます。
2. CC4 チャンネルが入力として設定されている場合（TIMx_CCMR4 レジスタの CC4S ビット） :
CCR4 は、最後の入力キャプチャ 4 イベント（IC4）によって転送されたカウンタ値です。

17.4.17 TIM2 および TIM3 DMA 制御レジスタ（TIM2_DCR および TIM3_DCR）

アドレスオフセット：0x48

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済み、常に 0 として読み出されます。

ビット 12:8 **DBL[4:0]**：DMA パースト長

この 5 ビットのベクタは、DMA 転送回数（タイマは、TIMx_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにパースト転送を認識します）を指定します。

00000：1 回転送
00001：2 回転送、
00010：3 回転送、
...
10001：18 回転送。

ビット 7:5 予約済み、常に 0 として読み出されます。

ビット 4:0 **DBA[4:0]**：DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します（TIMx_DMAR アドレスを通じて読み出し／書き込みアクセスが行われるとき）。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000：TIMx_CR1、
00001：TIMx_CR2、
00010：TIMx_SMCR
...

例：次の転送を考えます：DBL = 7 回転送 かつ DBA = TIMx_CR1。この場合、転送は、TIMx_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

17.4.18 完全転送の TIM2 および TIM3 DMA アドレス（TIM2_DMAR および TIM3_DMAR）

アドレスオフセット：0x4C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **DMAB[15:0]**：DMA パーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります：

$(\text{TIMx\_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御され、範囲は 0 から DBL です（DBL は TIMx_DCR 内で設定）。

DMA パースト機能の使用例

この例では、タイマ DMA パースト機能を使って CCRx レジスタ値を更新します（x = 2、3、4）。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

- 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします（下の注を参照）。
 - サーキュラモードは無効です。
- DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 転送、DBA = 0xE。
- TIMx 更新 DMA リクエストを有効にします（DIER レジスタのUDE ビットをセット）。
- TIMx を有効化
- DMA チャンネルを有効化注：

コード例については、「付録」セクション [A.9.20：外部トリガによって同期された 2 つのタイマのコード例](#) を参照してください。

注： この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。



17.4.19 TIM2 および TIM3 のレジスタマップ

TIM2 および TIM3 のレジスタは、次の表のようにマップされます。

表 60. TIM2 および TIM3 のレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIM2_CR1 および TIM3_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	CMS [1:0]	DIR	OPM	URS	UDIS	CEN	
	リセット値																							0	0	0	0	0	0	0	0	0	
0x04	TIM2_CR2 および TIM3_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THS	MMS[2:0]	CCDS	Res.	Res.	Res.			
	リセット値																								0	0	0	0	0				
0x08	TIM2_SMCR および TIM3_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETP	ECE	ETPS [1:0]		ETF[3:0]		MSM		TS[2:0]	OCCS		SMS[2:0]				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	TIM2_DIER および TIM3_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE		CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE	
	リセット値																		0		0	0	0	0	0	0	0	0	0	0	0	0	
0x10	TIM2_SR および TIM3_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF		Res.	TIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF	
	リセット値																				0	0	0	0			0	0	0	0	0	0	0
0x14	TIM2_EGR および TIM3_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	CC4G	CC3G	CC2G	CC1G	UG		
	リセット値																									0	0	0	0	0	0	0	0
0x18	TIM2_CCMR1 およ び TIM3_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2OE		OC2M [2:0]		OC2PE	OC2FE	CC2S [1:0]	OC1OE		OC1M [2:0]	OC1PE	OC1FE	CC1S [1:0]			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	TIM2_CCMR1 およ び TIM3_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		IC2F[3:0]		IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]		IC1 PSC [1:0]		CC1S [1:0]				
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	TIM2_CCMR2 およ び TIM3_CCMR2 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	O24OE		OC4M [2:0]		OC4PE	OC4FE	CC4S [1:0]	OC3OE		OC3M [2:0]	OC3PE	OC3FE	CC3S [1:0]			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM2_CCMR2 およ び TIM3_CCMR2 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		IC4F[3:0]		IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]		IC3 PSC [1:0]		CC3S [1:0]				
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	TIM2_CCER および TIM3_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	リセット値																	0		0	0	0		0	0	0		0	0		0	0	
0x24	TIM2_CNT および TIM3_CNT	CNT[31:16] (TIM2 のみ)																CNT[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	TIM2_PSC および TIM3_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	TIM2_ARR および TIM3_ARR	ARR[31:16] (TIM2 のみ)																ARR[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																

表 60. TIM2 および TIM3 のレジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	TIM2_CCR1 および TIM3_CCR1	CCR1[31:16] (TIM2 のみ)																CCR1[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x38	TIM2_CCR2 および TIM3_CCR2	CCR2[31:16] (TIM2 のみ)																CCR2[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x3C	TIM2_CCR3 および TIM3_CCR3	CCR3[31:16] (TIM2 のみ)																CCR3[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x40	TIM2_CCR4 および TIM3_CCR4	CCR4[31:16] (TIM2 のみ)																CCR4[15:0]															
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x44	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																
0x48	TIM2_DCR および TIM3_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	リセット値																				0	0	0	0	0				0	0	0	0	0
0x4C	TIM2_DMAR および TIM3_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

18 汎用タイマ (TIM14)

18.1 TIM14 の概要

TIM14 汎用タイマは、プログラム可能なプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

入力信号のパルス長の測定（入力キャプチャ）や出力波形の生成（出力比較、PWM）など、さまざまな目的に使用できます。

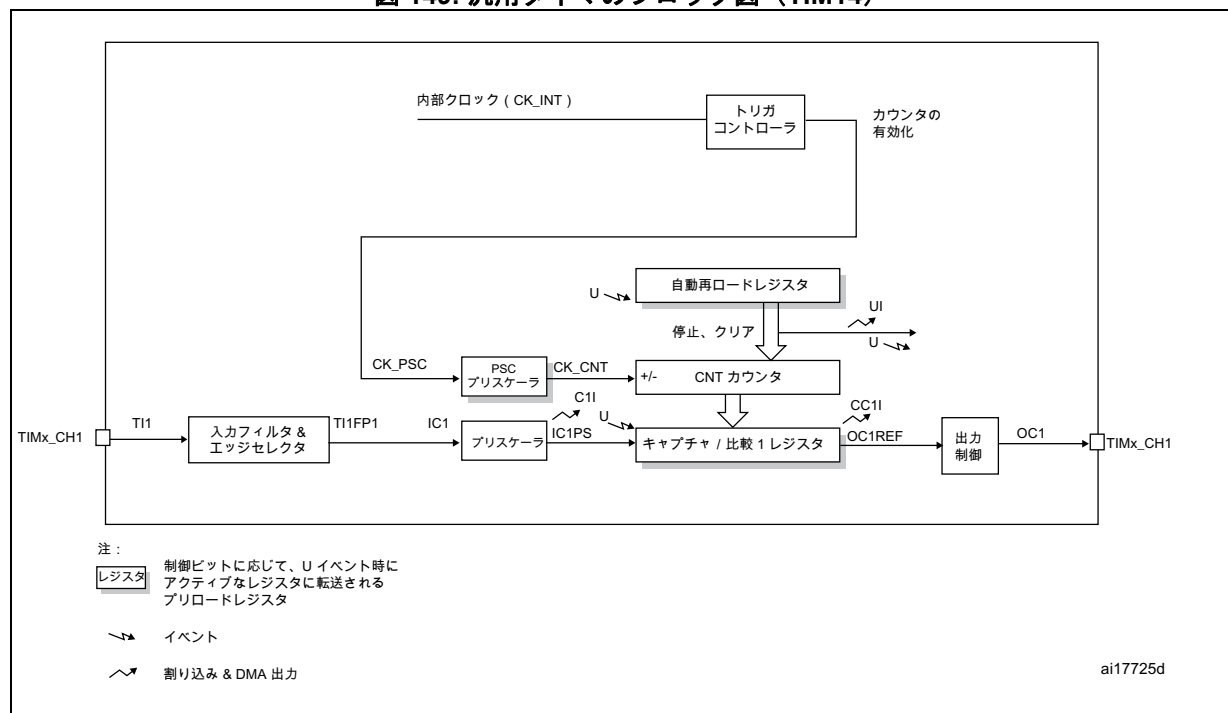
パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

TIM14 タイマは完全に独立していて、いかなるリソースも共有しません。[セクション 17.3.15](#) で説明するように、同期できます。

18.2 TIM14 の主な機能

- 16 ビット自動再ロードアップカウンタ
- カウンタクロック周波数を 1 から 65535 の間の値で分周するために使用される 16 ビットのプログラム可能なプリスケラ（動作中に変更可能）。
- 次の機能を持つ独立チャネル：
 - － 入力キャプチャ
 - － 出力比較
 - － PWM 生成（エッジアラインモード）
- 以下のイベント時の割り込み生成。
 - － 更新：カウンタオーバーフロー、カウンタの初期化（ソフトウェアによる）
 - － 入力キャプチャ
 - － 出力比較

図 149. 汎用タイマのブロック図 (TIM14)



18.3 TIM14 の機能詳細

18.3.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップの動作が可能です。カウンタのクロックは、プリスケラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、更新イベントはカウンタがオーバーフローしたときに送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケラ出力 CK_CNT から供給されます (カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください)。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

図 151 と 図 152 に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 150. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

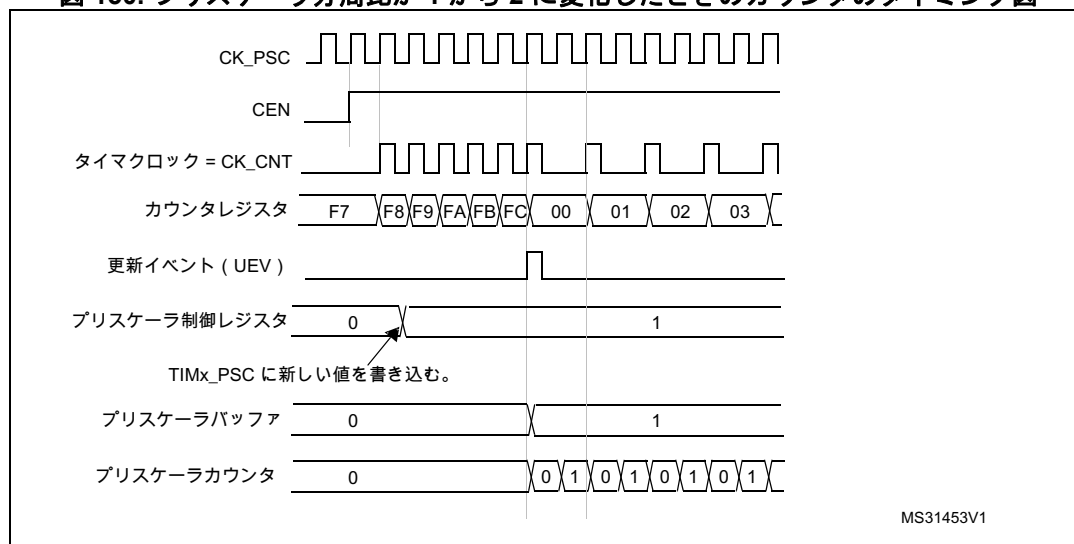
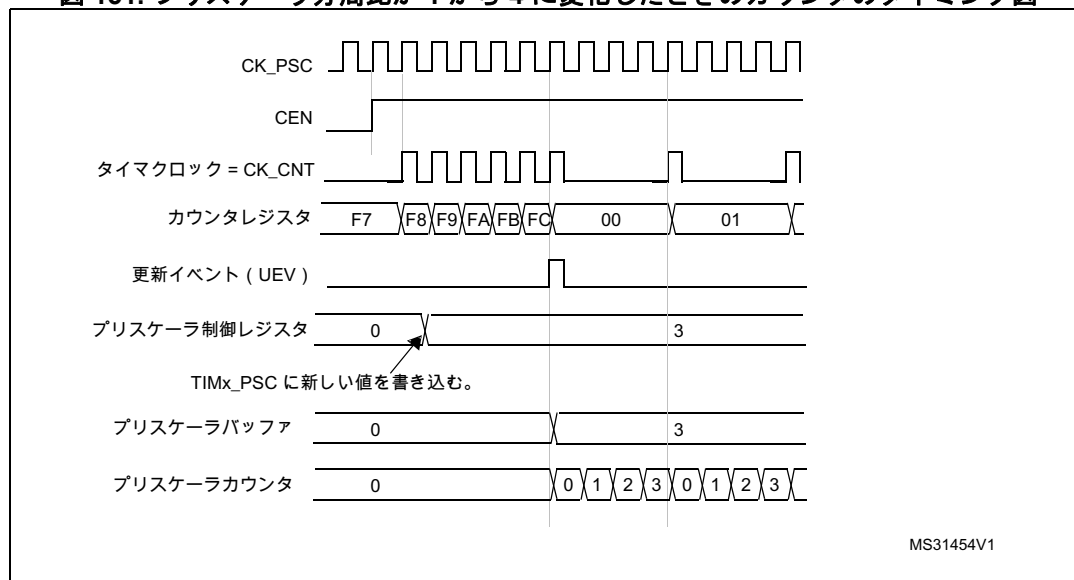


図 151. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



18.3.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

TIMx_EGR レジスタの UG ビットをセットすると、更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 152. 内部クロック分周比が 1 の場合のカウンタのタイミング図

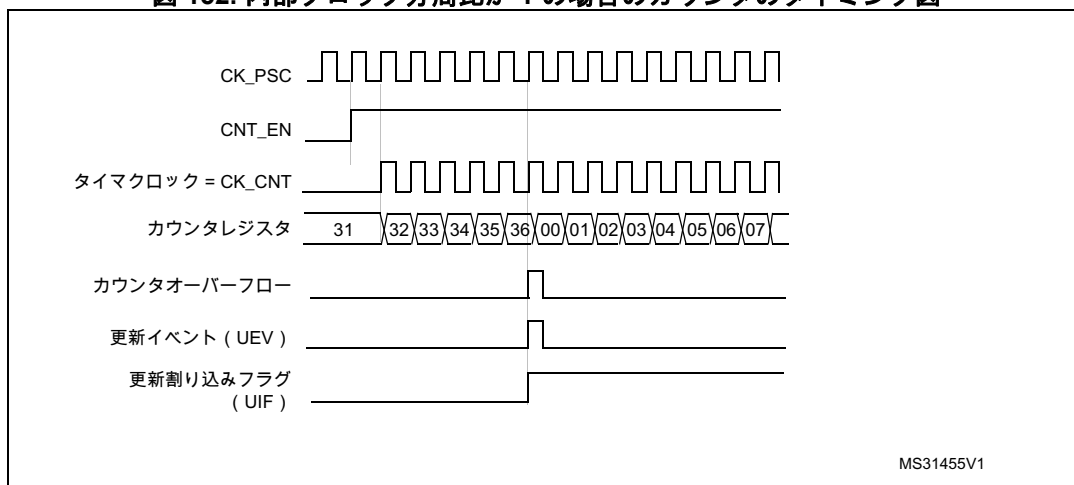


図 153. 内部クロック分周比が 2 の場合のカウンタのタイミング図

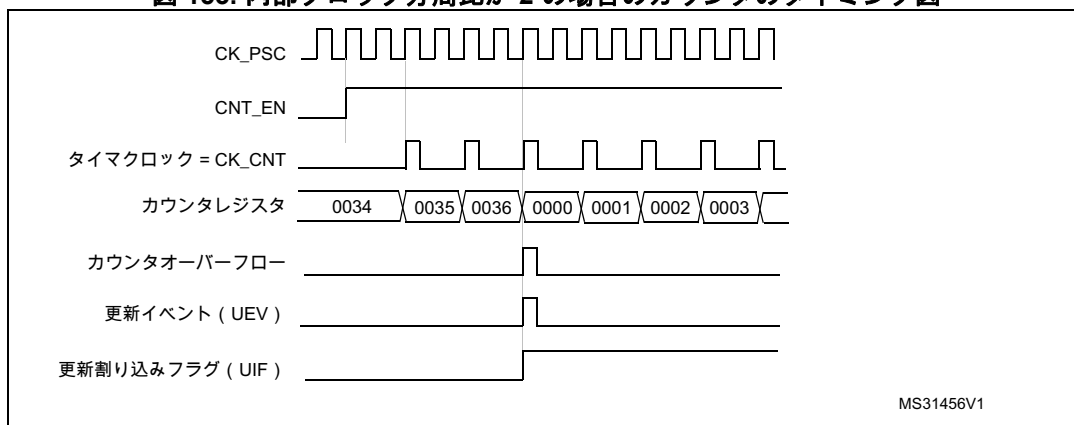


図 154. 内部クロック分周比が 4 の場合のカウンタのタイミング図

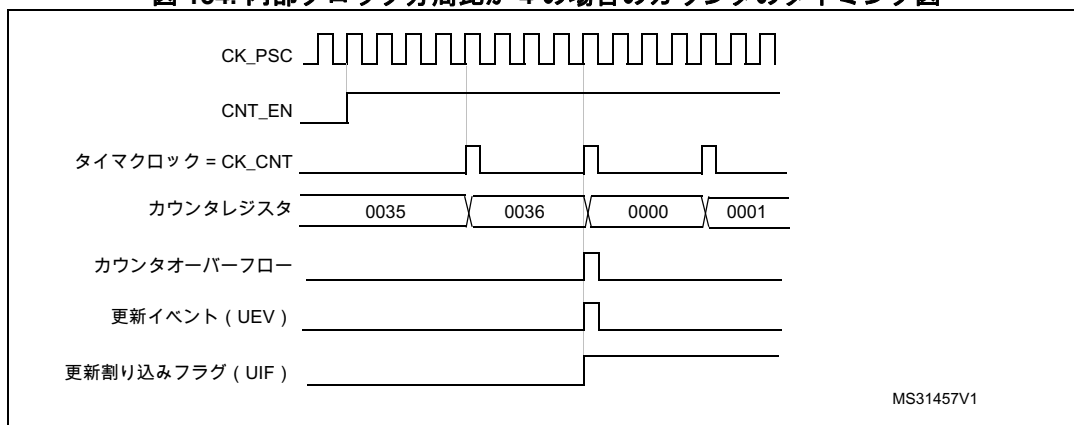


図 155. 内部クロック分周比が N の場合のカウンタのタイミング図

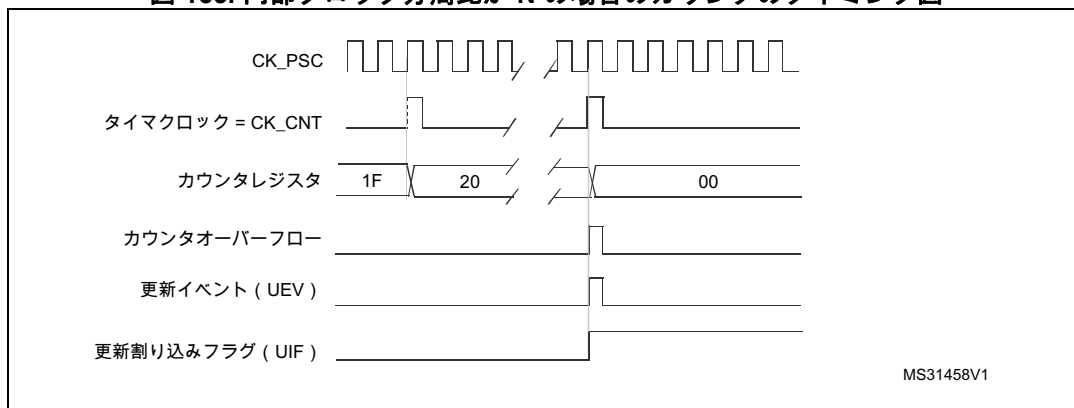


図 156. ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)

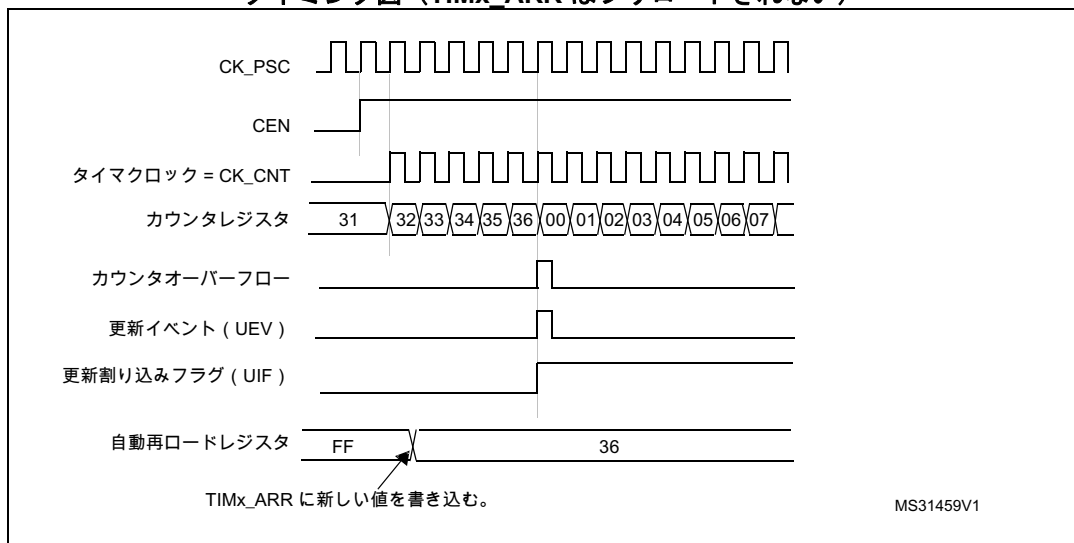
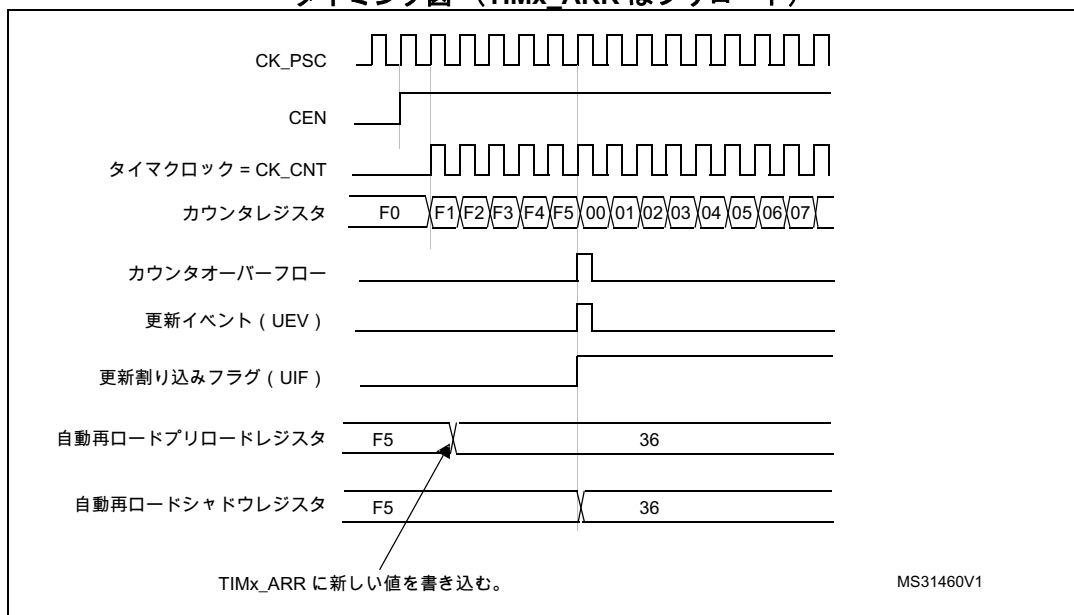


図 157. ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)



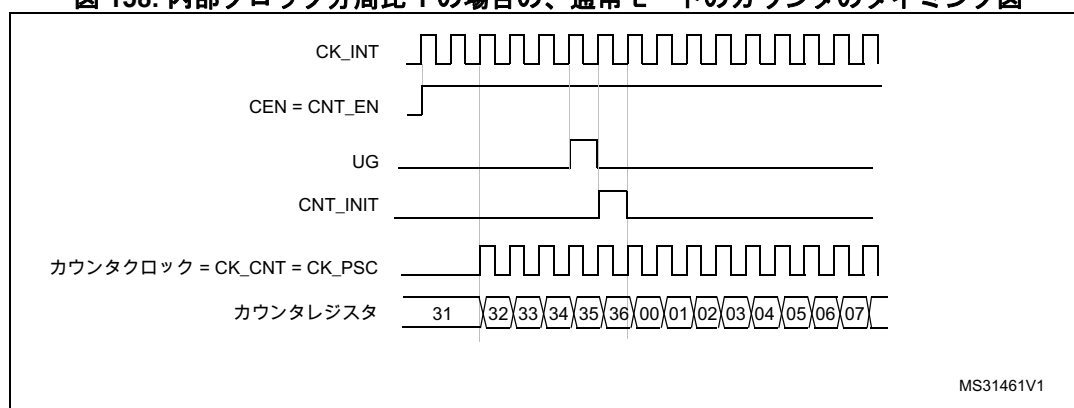
18.3.3 クロックソース

カウンタクロックは、内部クロック (CK_INT) ソースから供給されます。

TIMx_CR1 レジスタの CEN ビットと TIMx_EGR レジスタの UG ビットは実際の制御ビットであり、ソフトウェアによってのみ変更できます (ただし、自動的にクリア状態が保持される UG ビットを除く)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

図 158 に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

図 158. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



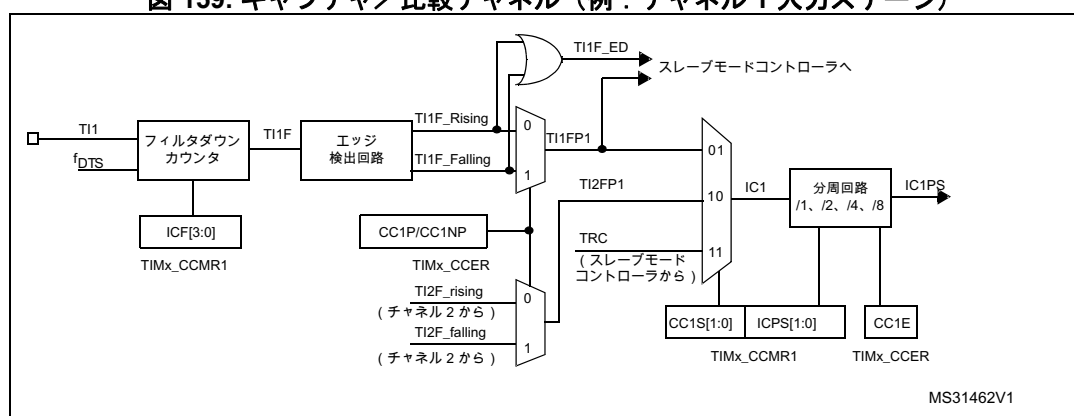
18.3.4 キャプチャ/比較チャネル

各キャプチャ/比較チャネルは、キャプチャ/比較レジスタ (シャドウレジスタを含む)、キャプチャの入カステージ (デジタルフィルタ、マルチプレクサ、プリスケアラ)、および出力カステージ (比較回路と出力制御) から構成されています。

図 159 から 図 161 に、1 つのキャプチャ/比較チャネルの概要を示します。

入カステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った TIxF を生成します。次に、極性選択付きのエッジ検出回路が、スレーブモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIxFPx) を生成します。この信号はプリスケアラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 159. キャプチャ/比較チャネル (例: チャネル 1 入カステージ)



出力ステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 160. キャプチャ／比較チャンネル 1 メイン回路

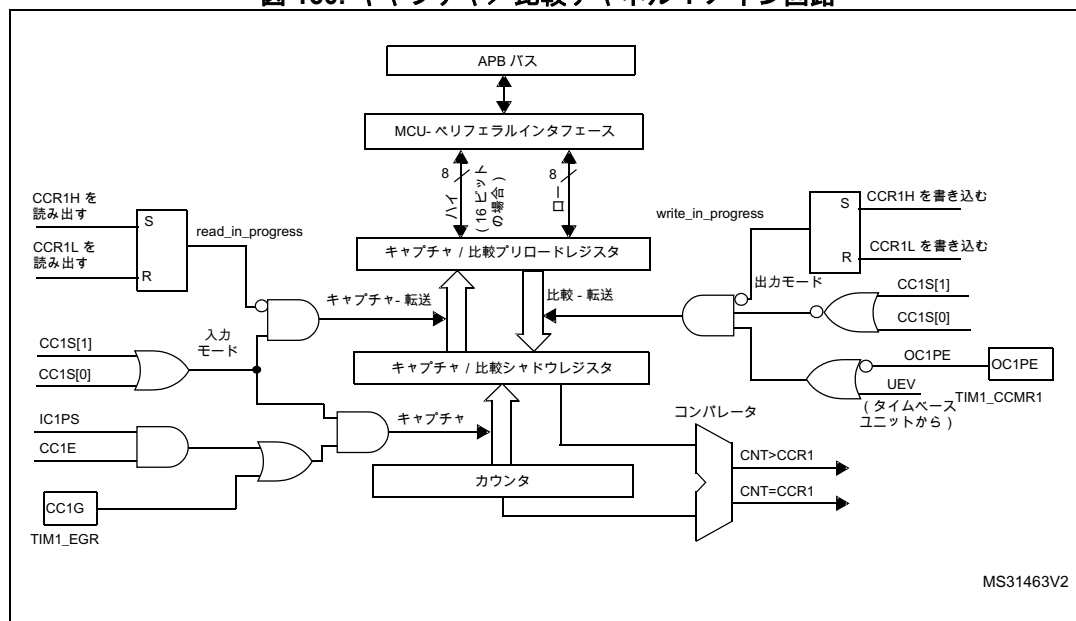
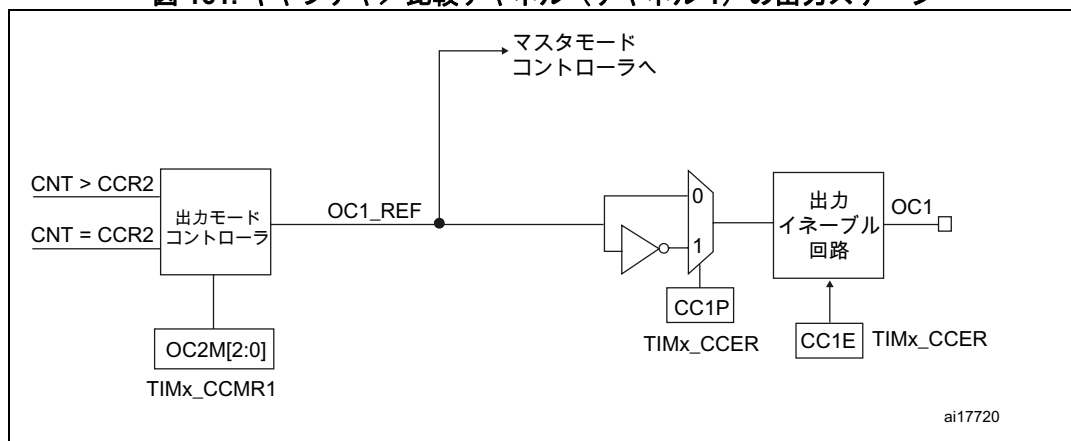


図 161. キャプチャ／比較チャンネル（チャンネル 1）の出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

18.3.5 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ/比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

1. アクティブ入力を選択します。TIMx_CCR1 は TI1 入力にリンクされていなければならないので、TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化するとすぐに、チャンネルは入力モードに設定され、TIMx_CCR1 レジスタは読み出し専用になります。
2. タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx_CCMR1 レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数 f_{DTS} でサンプリング)。次に、TIMx_CCMR1 レジスタの IC1F ビットに“0011”を書き込みます。
3. TI1 チャンネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットと CC1NP ビットに“00”を設定します (この場合、立ち上がりエッジの選択)。
4. 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。
5. TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
6. 必要に応じて、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にします。

コード例については、「付録」セクション [A.9.3: 入力キャプチャ設定のコード例](#) を参照してください。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。

コード例については、「付録」セクション [A.9.4: 入力キャプチャのデータ管理のコード例](#) を参照してください。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注: TIMx_EGR レジスタの対応する CCxG ビットをセットすることで、IC 割り込みリクエストをソフトウェアで発生させることができます。

18.3.6 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF、次に OCx) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルにするには、対応する TIMx_OC MRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP = 0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みリクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

18.3.7 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

1. 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM = 000)、アクティブにセットされるか (OCxM = 001)、インアクティブにセットされるか (OCxM = 010)、または反転されます (OCxM = 011)。
2. 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
3. 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

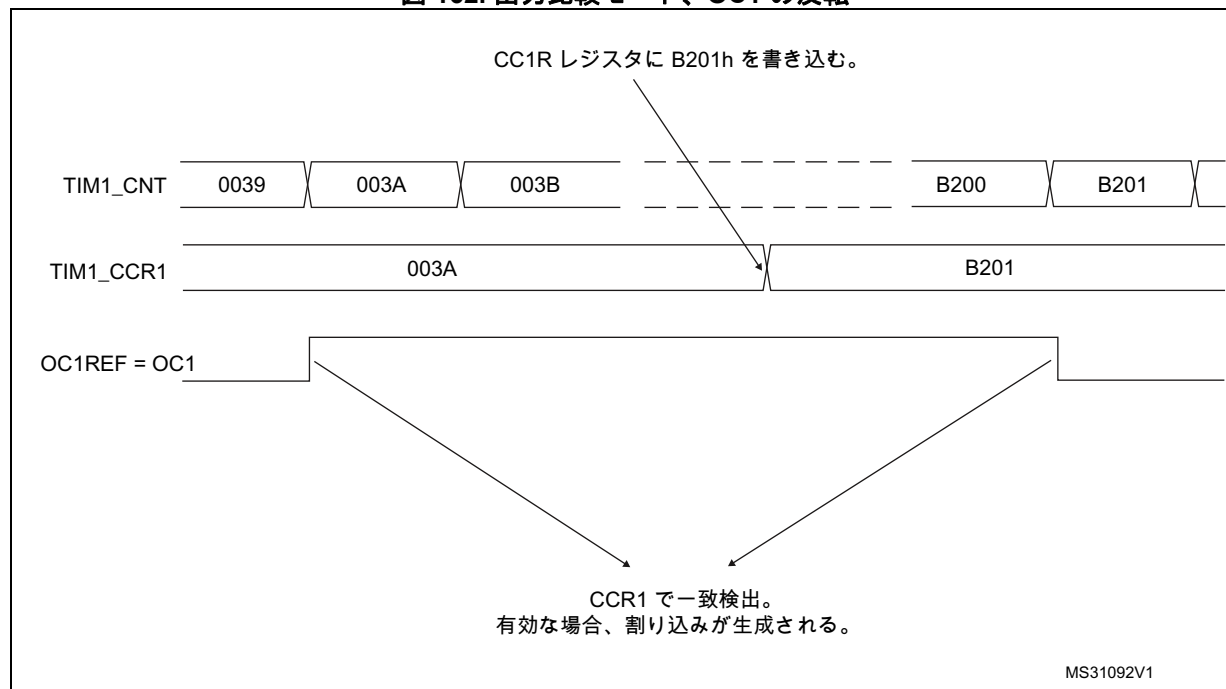
手順:

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例:
 - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに“011”を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに“0”を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに“0”を書き込みます。
 - 出力を有効にするには、CCxE ビットに“1”を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

コード例については、「付録」セクション [A.9.7: 入力比較設定のコード例](#) を参照してください。

いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り（OCxPE=0）。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 162](#) に示します。

図 162. 出力比較モード、OC1 の反転



18.3.8 PWM モード

パルス幅変調 (PWM) モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx の極性は、TIMx_CCER レジスタの CCxP ビットを使用して、ソフトウェアでプログラムできます。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、TIMx_CCER レジスタの CCxE ビットによって有効化されます。詳細については、TIMx_CCERx レジスタの説明を参照してください。

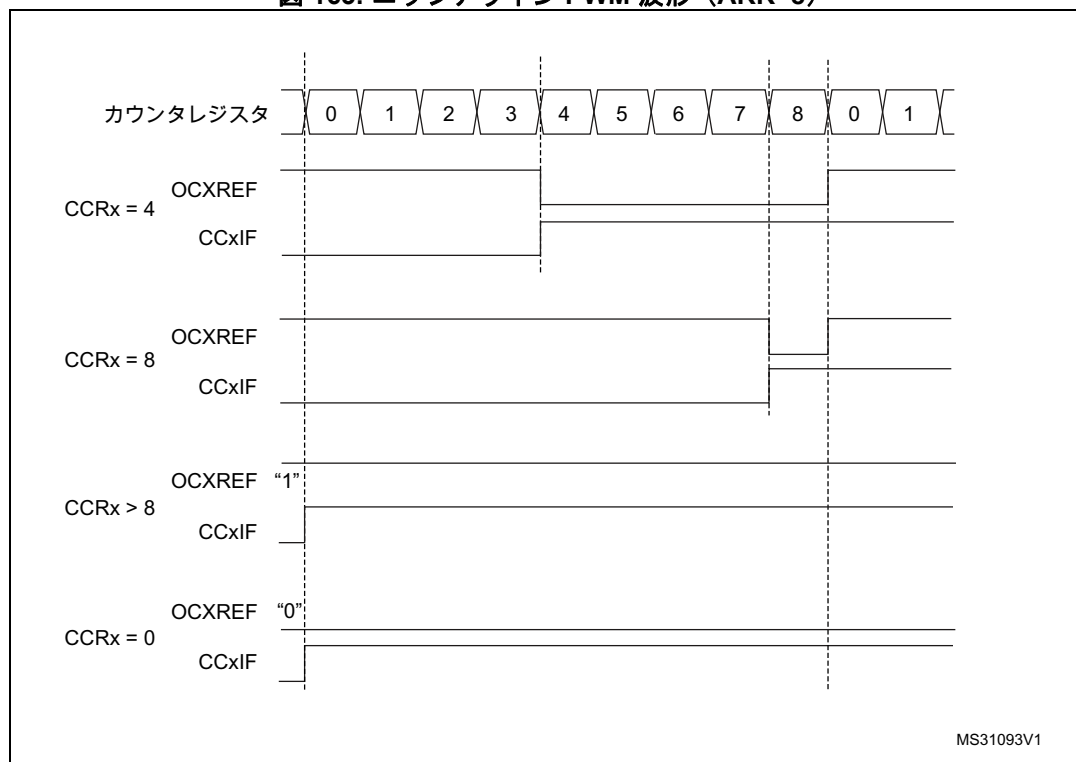
PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CNT ≤ TIMx_CCRx かどうか判断されます。

カウンタはカウントアップしているので、タイマはエッジアラインモードでのみ PWM を生成できません。

PWM エッジアラインモード

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。図 163 に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 163. エッジアライン PWM 波形 (ARR=8)



コード例については、「付録」セクション [A.9.8: エッジアライン PWM の設定例](#) を参照してください。

18.3.9 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex™-M0 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

18.4 TIM14 レジスタ

18.4.1 TIM14 制御レジスタ 1 (TIM14_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	Res.	URS	UDIS	CEN
						rw	rw	rw					rw	rw	rw

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:8 **CKD** : クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デジタルフィルタ (ETR、Tlx) によって使用されるサンプリングクロックとの間の分周比を示します。

- 00: $t_{DTS} = t_{CK\_INT}$
- 01: $t_{DTS} = 2 \times t_{CK\_INT}$
- 10: $t_{DTS} = 4 \times t_{CK\_INT}$
- 11: 予約済み

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0: TIMx_ARR レジスタはバッファされません。
- 1: TIMx_ARR レジスタはバッファされます。

ビット 6:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **URS** : 更新リクエストソース

このビットは、更新割り込み (UEV) ソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0: 次のイベントのいずれかが UEV を発生します (有効な場合)。
 - カウンタオーバーフロー
 - UG ビットのセット
- 1: カウンタオーバーフローのみが UEV を発生します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、更新割り込み (UEV) イベント生成を有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0: UEV は有効です。UEV は、次のいずれかのイベントによって生成されます。
 - カウンタオーバーフロー
 - UG ビットのセット

バッファを持つレジスタにはプリロード値がロードされます。

- 1: UEV は無効です。UEV は生成されず、シャドウレジスタ (ARR、PSC、CCRx) は値を維持します。カウンタとプリスケアラは、UG ビットがセットされた場合に再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0: カウンタは無効です。
- 1: カウンタは有効です。

18.4.2 TIM14 割り込み有効レジスタ (TIM14_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル

- 0 : CC1 割り込みは無効です。
- 1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

- 0 : 更新割り込みは無効です。
- 1 : 更新割り込みは有効です。

18.4.3 TIM14 ステータスレジスタ (TIM14_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバーキャプチャフラグ

- このフラグは、対応するチャネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。
- 0 : オーバーキャプチャは検出されていません。
 - 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1IF** : キャプチャ/比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときに、ハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時に CC1IF ビットがハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました (選択された極性に一致するエッジが IC1 で検出されました)。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- オーバーフローが発生し、かつ TIMx_CR1 レジスタで UDIS = 0 の場合。
- TIMx_CR1 レジスタで URS = 0 かつ UDIS = 0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

18.4.4 TIM14 のイベント生成レジスタ (TIM14_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1G** : キャプチャ／比較 1 生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ／比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割り込みが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。カウンタはクリアされます。

18.4.5 TIM14 のキャプチャ／比較モードレジスタ 1（TIM14_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]				OC1PE	OC1FE	CC1S[1:0]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]			
								r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード

ビット 15:7 予約済み

ビット 6:4 **OC1M**：出力比較 1 モード

これらのビットは、OC1 の元となる出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 のアクティブレベルは CC1P ビットに依存します。

000：停止。出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。

001：一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ／比較レジスタ 1（TIMx_CCR1）と一致したときに、強制的にハイになります。

010：一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ／比較レジスタ 1（TIMx_CCR1）と一致したときに、強制的にローになります。

011：反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100：強制非アクティブレベル - OC1REF は強制的にローになります。

101：強制アクティブレベル - OC1REF は強制的にハイになります。

110：PWM モード 1 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合はインアクティブになります。

111：PWM モード 2 - チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はインアクティブに、そうでない場合はアクティブになります。

注： PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときに、OCREF のレベルが変化します。

ビット 3 **OC1PE**：出力比較 1 プリロードイネーブル

0：TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1：TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注： PWM モードは、ワンパルスモード（TIMx_CR1 レジスタの OPM ビットがセットされている）のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 2 OC1FE : 出力比較 1 高速インエーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向（入力/出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : 予約済み

11 : 予約済み

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

入力キャプチャモード

ビット 15:8 予約済み

ビット 7:4 IC1F : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$, $N = 2$

0010 : $f_{SAMPLING} = f_{CK\_INT}$, $N = 4$

0011 : $f_{SAMPLING} = f_{CK\_INT}$, $N = 8$

0100 : $f_{SAMPLING} = f_{DTS}/2$, $N = 6$

0101 : $f_{SAMPLING} = f_{DTS}/2$, $N = 8$

0110 : $f_{SAMPLING} = f_{DTS}/4$, $N = 6$

0111 : $f_{SAMPLING} = f_{DTS}/4$, $N = 8$

1000 : $f_{SAMPLING} = f_{DTS}/8$, $N = 6$

1001 : $f_{SAMPLING} = f_{DTS}/8$, $N = 8$

1010 : $f_{SAMPLING} = f_{DTS}/16$, $N = 5$

1011 : $f_{SAMPLING} = f_{DTS}/16$, $N = 6$

1100 : $f_{SAMPLING} = f_{DTS}/16$, $N = 8$

1101 : $f_{SAMPLING} = f_{DTS}/32$, $N = 5$

1110 : $f_{SAMPLING} = f_{DTS}/32$, $N = 6$

1111 : $f_{SAMPLING} = f_{DTS}/32$, $N = 8$

注 : ICxF[3:0] = 1、2、または 3 のときに、式の f_{DTS} が CK_INT に置き換えられるよう注意する必要があります。

ビット 3:2 IC1PSC : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力エッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ／比較 1 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

0 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : 予約済み

11 : 予約済み

注 : **CC1S** ビットは、チャンネルがオフ (**TIMx_CCER** レジスタの **CC1E=0**) のときにのみ書き込み可能です。

18.4.6 TIM14 のキャプチャ／比較有効レジスタ (TIM14_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

ビット 15:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 **CC1NP** : キャプチャ／比較 1 相補出力極性

CC1 チャンネルが出力として設定されている場合 : CC1NP はクリアされたままにする必要があります。

CC1 チャンネルが入力として設定されている場合 : CC1NP ビットは、TI1FP1 の極性を定義するために CC1P と組み合わせて使用されます (CC1P の説明を参照してください)。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **CC1P** : キャプチャ／比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1P ビットは、トリガ動作またはキャプチャ動作での TI1FP1 および TI2FP1 の極性を選択します。

00 : 非反転／立ち上がりエッジ

回路は TI1FP1 の立ち上がりエッジに反応し (キャプチャモード)、TI1FP1 は反転されません。

01 : 反転／立ち上がりエッジ

回路は TI1FP1 の立ち上がりエッジに反応し (キャプチャモード)、TI1FP1 は反転されます。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転／両エッジ

回路は TI1FP1 の立ち上がりエッジに反応し (キャプチャモード)、TI1FP1 は反転されません。

ビット 0 **CC1E** : キャプチャ／比較 1 出力イネーブル。

CC1 チャンネルが出力として設定されている場合 :

0 : オフ - OC1 はアクティブではありません。

1 : オン - OC1 信号は、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ／比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。

表 61. 標準 OCx チャンネルの出力制御ビット

CCxE ビット	OCx 出力状態
0	出力無効 (OCx=0、OCx_EN=0)
1	OCx = OCxREF + 極性、OCx_EN = 1

注：標準 OCx チャンネルに接続されている外部入出力ピンの状態は、OCx チャンネルの状態と、GPIO レジスタに依存します。

18.4.7 TIM14 のカウンタ (TIM14_CNT)

アドレスオフセット：0x24

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0]：カウンタ値

18.4.8 TIM14 のプリスケアラ (TIM14_PSC)

アドレスオフセット：0x28

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0]：プリスケアラ値

カウンタクロック周波数 CK_CNT は $f_{CK\_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます。

18.4.9 TIM14 の自動再ロードレジスタ (TIM14_ARR)

アドレスオフセット：0x2C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0]：自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 18.3.1：タイムベースユニット \(445 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

18.4.10 TIM14 のキャプチャ／比較レジスタ 1（TIM14_CCR1）

アドレスオフセット：0x34

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]**：キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合：
CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。
TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。
アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

チャンネル CC1 が入力として設定されている場合：
CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

18.4.11 TIM14 オプションレジスタ（TIM14_OR）

アドレスオフセット：0x50

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM1_RMP	
														rW	rW

ビット 15:2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **TIM1_RMP [1:0]**：タイマ 入力 1 再配置

ソフトウェアでセット／クリアされます。

00：TIM14 のチャンネル 1 は、GPIO に接続されます。デバイスのデータシートのオルタネート機能配置表を参照してください。

01：TIM14 のチャンネル 1 は、RTCCLK に接続されます。

10：TIM14 のチャンネル 1 は、HSE/32 クロックに接続されます。

11：TIM14 のチャンネル 1 はマイクロコントローラクロック出力（MCO）に接続されます。この選択は、クロック設定レジスタ（RCC_CFGR）の MCO[2:0] ビットで制御されます（[セクション 7.4.2：クロック設定レジスタ（RCC_CFGR）](#)を参照）。

18.4.12 TIM14 レジスタマップ

TIM14 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 62. TIM14 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIM14_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	Res.	URS	UDIS	CEN	
	リセット値																							0	0	0					0	0	0	
0x08	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x0C	TIM14_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE	
	リセット値																															0	0	
0x10	TIM14_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF	
	リセット値																							CC1OF		Res.	Res.	Res.	Res.	Res.	Res.	0	0	
0x14	TIM14_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG	
	リセット値																															0	0	
0x18	TIM14_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.					OC1M [2:0]	OC1PE	OC1FE	CC1S [1:0]	
	リセット値																											0	0	0	0	0	0	
	TIM14_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.					IC1F[3:0]	IC1PSC [1:0]	CC1S [1:0]		
	リセット値																											0	0	0	0	0	0	
0x1C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x20	TIM14_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1P	CC1E
	リセット値																													0		0	0	
0x24	TIM14_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x28	TIM14_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	TIM14_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x30	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	
0x34	TIM14_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x38～ 0x4C	予約済み	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																																	

表 62. TIM14 レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x50	TIM14_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値																																0	0

19 汎用タイマ (TIM15/16/17)

TIM15 は STM32F03x では使用できません。

19.1 TIM15/16/17 の概要

TIM15/16/17 タイマは、プログラム可能なプリスケラによって駆動される 16 ビット自動再ロードカウンタで構成されています。

入力信号のパルス長の測定（入力キャプチャ）や出力波形の生成（出力比較、PWM、デッドタイムを挿入した相補 PWM）など、さまざまな目的に使用できます。

パルス幅と波形の周期は、タイマプリスケラと RCC クロックコントローラプリスケラを使用して、数マイクロ秒から数ミリ秒までの範囲で変化させることができます。

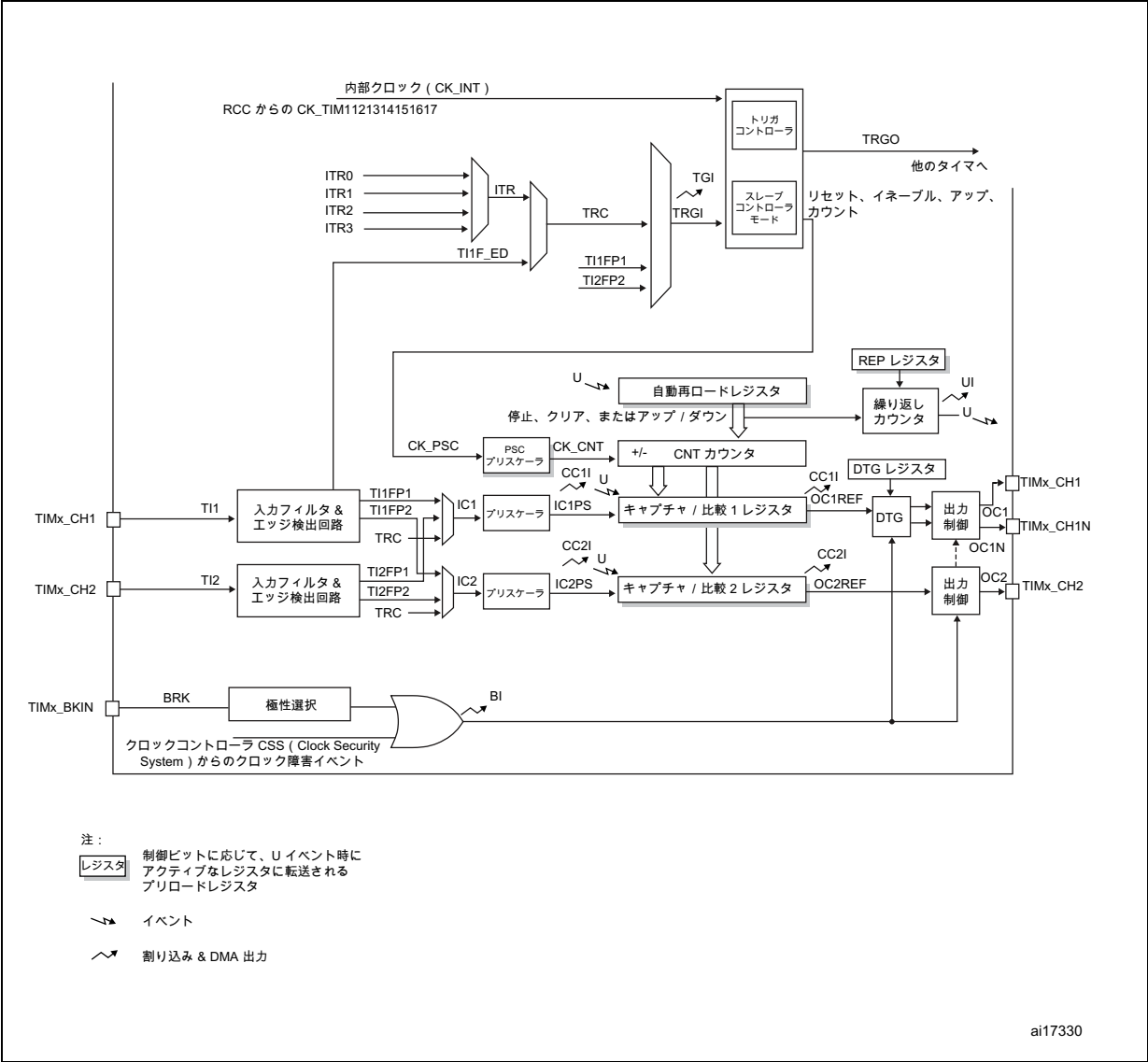
TIM15/16/17 タイマは完全に独立していて、いかなるリソースも共用しません。TIM15 は他のタイマと同期できます。

19.2 TIM15 の主な機能

TIM15 には以下の機能があります。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケラ（動作中も変更可能）で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- 次の機能を持つ、最大 2 つの独立チャネル。
 - － 入力キャプチャ
 - － 出力比較
 - － PWM 生成（エッジアラインモード）
 - － ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力（チャネル 1 のみ）
- 外部信号でタイマを制御し、複数のタイマを相互接続する同期回路。
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレイク入力。
- 以下のイベント時の割り込み/DMA 生成。
 - － 更新：カウンタオーバーフロー、カウンタの初期化（ソフトウェアまたは内部／外部トリガによる）
 - － トリガイベント（カウンタの開始、停止、初期化、または内部／外部トリガによるカウント）
 - － 入力キャプチャ
 - － 出力比較
 - － ブレイク入力（割り込みリクエスト）

図 164. TIM15 ブロック図

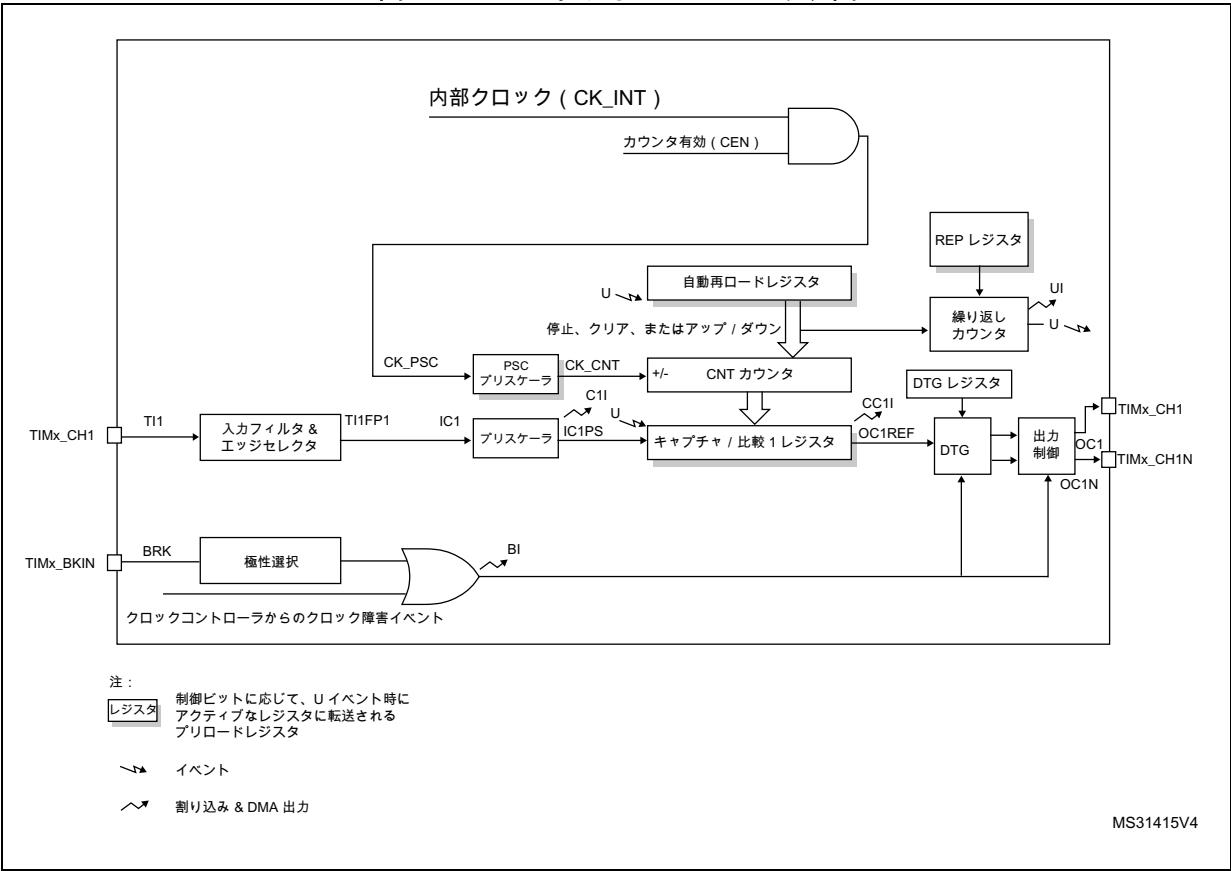


19.3 TIM16 および TIM17 の主な機能

TIM16 および TIM17 タイマには以下の機能があります。

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ（動作中も変更可能）で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- 次の機能を持つ、1 チャンネルタイマ。
 - 入力キャプチャ
 - 出力比較
 - PWM 生成（エッジアラインモード）
 - ワンパルスモード出力
- プログラム可能なデッドタイムを持つ相補出力
- カウンタの特定のサイクル数後にのみタイマレジスタを更新する繰り返しカウンタ。
- タイマの出力信号をリセット状態または既知の状態にするブレーク入力。
- 以下のイベント時の割り込み/DMA 生成。
 - 更新：カウンタオーバーフロー
 - 入力キャプチャ
 - 出力比較
 - ブレーク入力

図 165. TIM16 および TIM17 ブロック図



19.4 TIM15/16/17 の機能説明

19.4.1 タイムベースユニット

プログラマブル高機能制御タイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットカウンタです。カウンタはカウントアップ、カウントダウン、またはアップダウンします。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)
- 繰り返しカウンタレジスタ (TIMx_RCR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きは、プリロードレジスタへのアクセスになります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント (UEV) ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 の場合、カウンタがオーバーフロー（またはダウンカウント時はアンダーフロー）に達したときに、更新イベントが送られます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の中で詳しく説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウンタイネーブルビット (CEN) がセットされているときにのみ、プリスケアラ出力 CK_CNT から供給されます（カウンタの有効化の詳細については、スレーブモードコントローラの説明も参照してください）。

TIMx_CR1 レジスタの CEN ビットがセットされてから、カウンタがカウントを開始するまでに 1 クロックサイクルの遅延があることに注意してください。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。この制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 151](#) と [図 152](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 166. プリスケール分周比が 1 から 2 に変化したときのカウンタのタイミング図

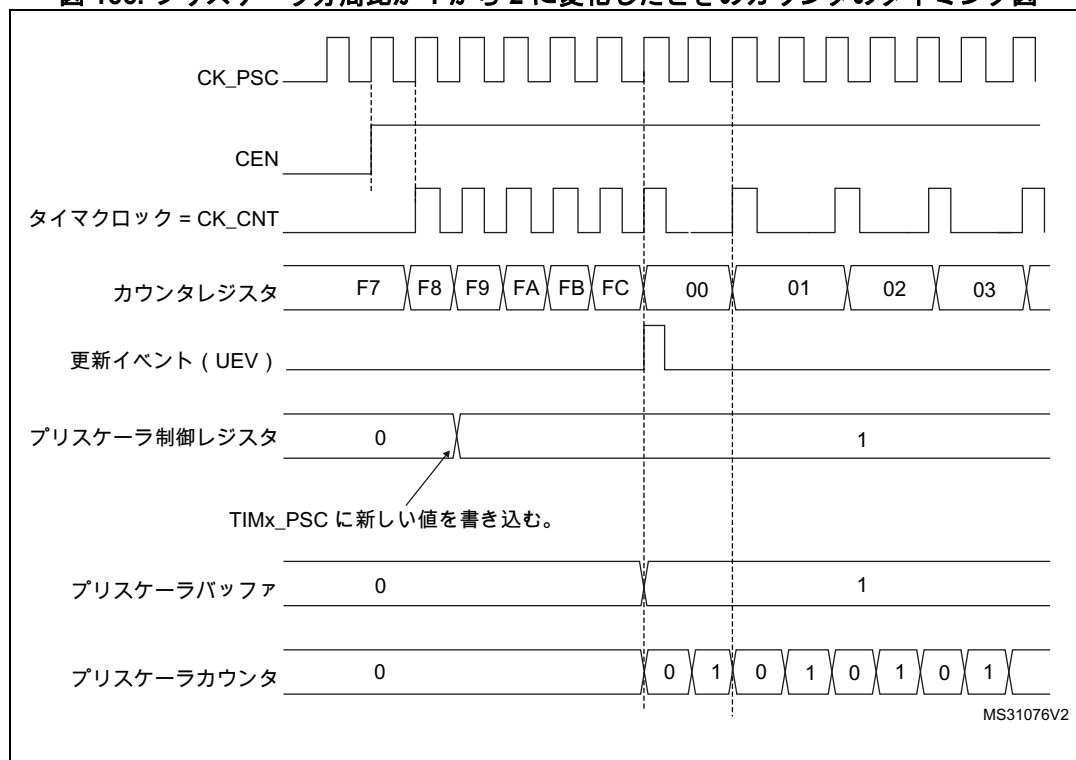
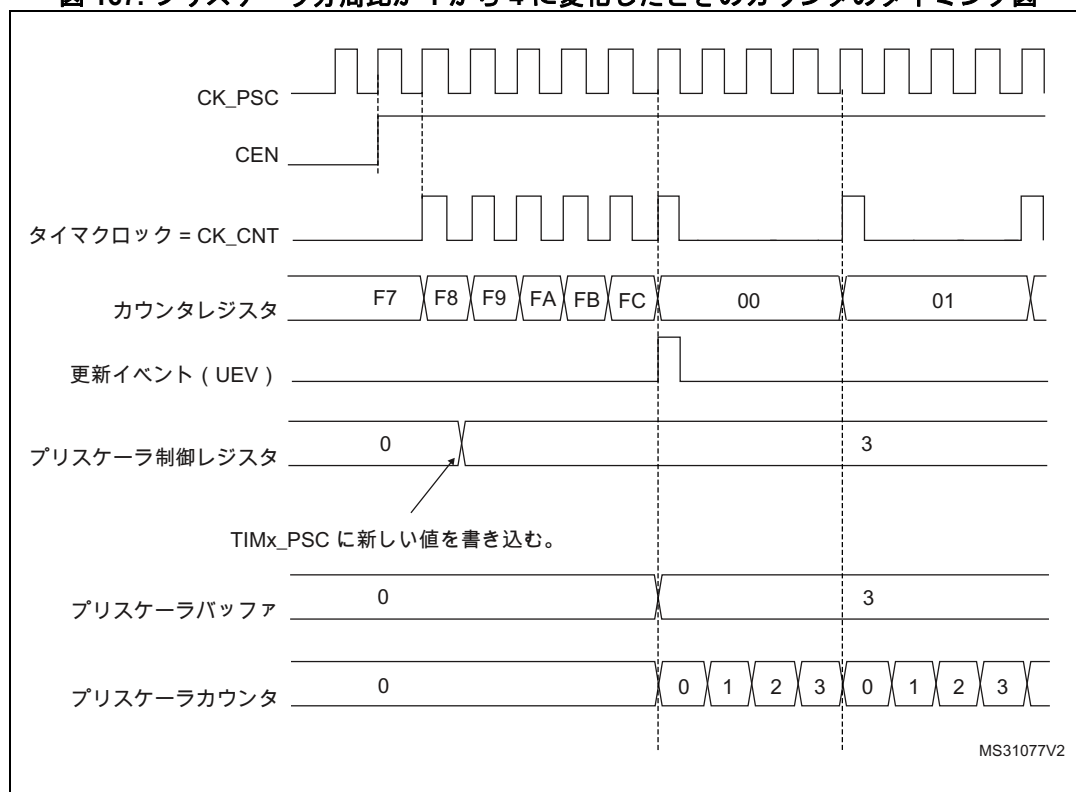


図 167. プリスケール分周比が 1 から 4 に変化したときのカウンタのタイミング図



19.4.2 カウンタモード

アップカウントモード

アップカウントモードでは、カウンタは 0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントし、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

繰り返しカウンタが使用されている場合には、繰り返しカウンタレジスタにプログラムされている回数 (TIMx_RCR) までアップカウント動作が繰り返され、その後に更新イベント (UEV) が生成されます。繰り返しカウンタが使用されていないときには、カウンタのオーバーフローごとに更新イベントが生成されます。

(ソフトウェアによって、またはスレーブモードコントローラを使用して) TIMx_EGR レジスタの UG ビットをセットすることでも更新イベントが生成されます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。この場合、UDIS ビットに 0 が書き込まれるまで、更新イベントは発生しません。ただし、プリスケアラのカウンタと同じく、カウンタは 0 からカウントをリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS ビット (更新リクエスト選択) がセットされている場合は、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (したがって、割り込みや DMA リクエストは送信されません)。これは、キャプチャイベント時にカウンタをクリアしているときに、更新とキャプチャの両方の割り込みを生成するのを防ぐためです。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットに応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- 繰り返しカウンタには TIMx_RCR レジスタの内容が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。
- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再びロードされます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 168. 内部クロック分周比が 1 の場合のカウンタのタイミング図

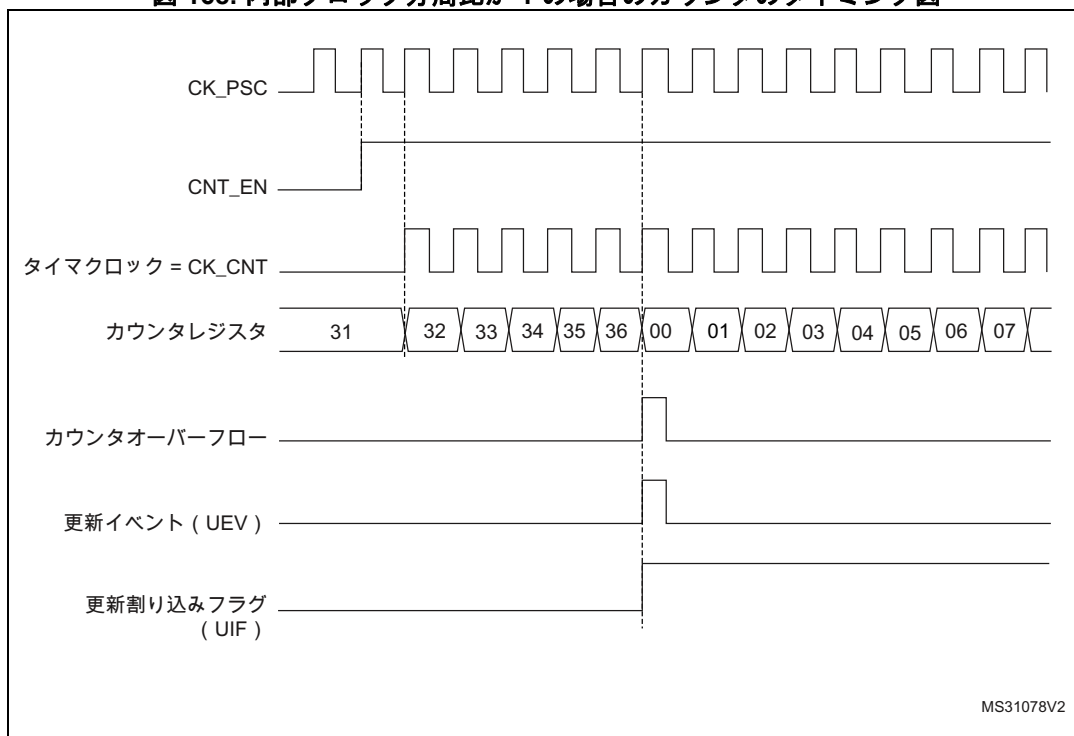


図 169. 内部クロック分周比が 2 の場合のカウンタのタイミング図

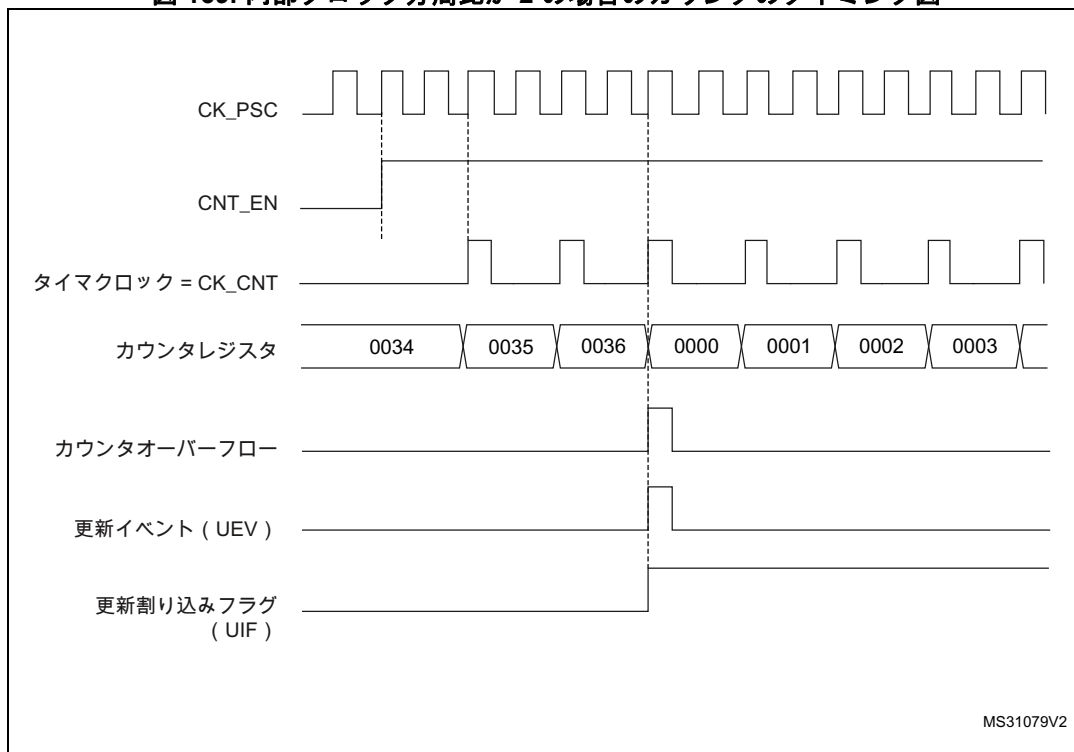


図 170. 内部クロック分周比が 4 の場合のカウンタのタイミング図

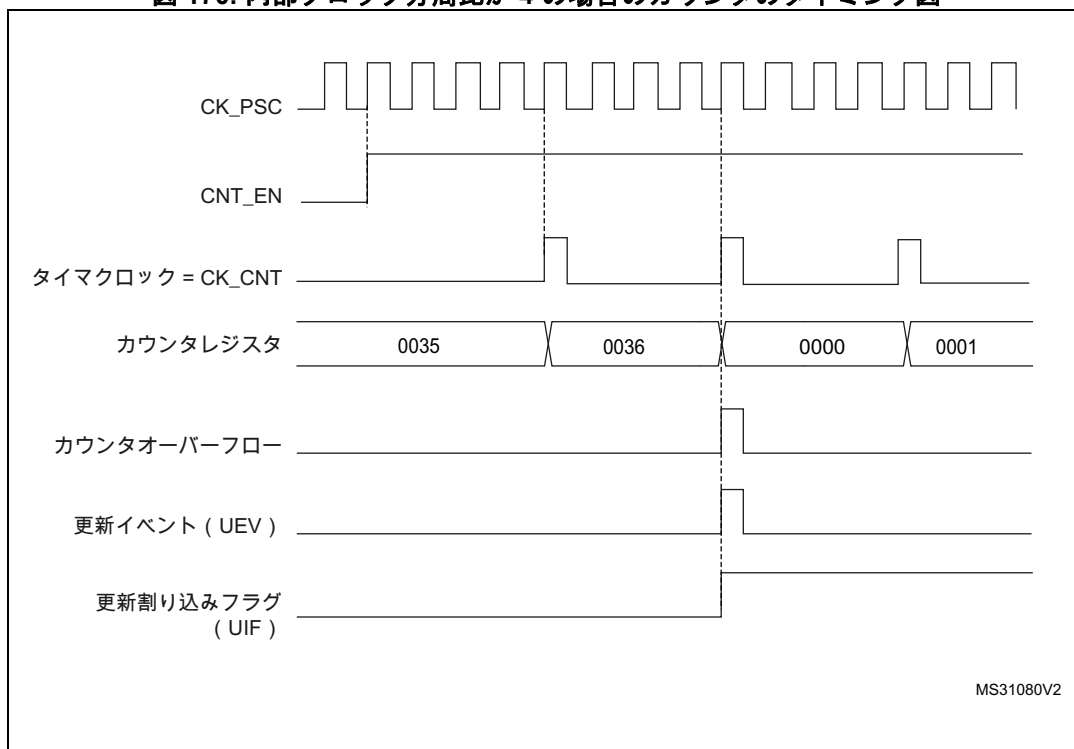


図 171. 内部クロック分周比が N の場合のカウンタのタイミング図

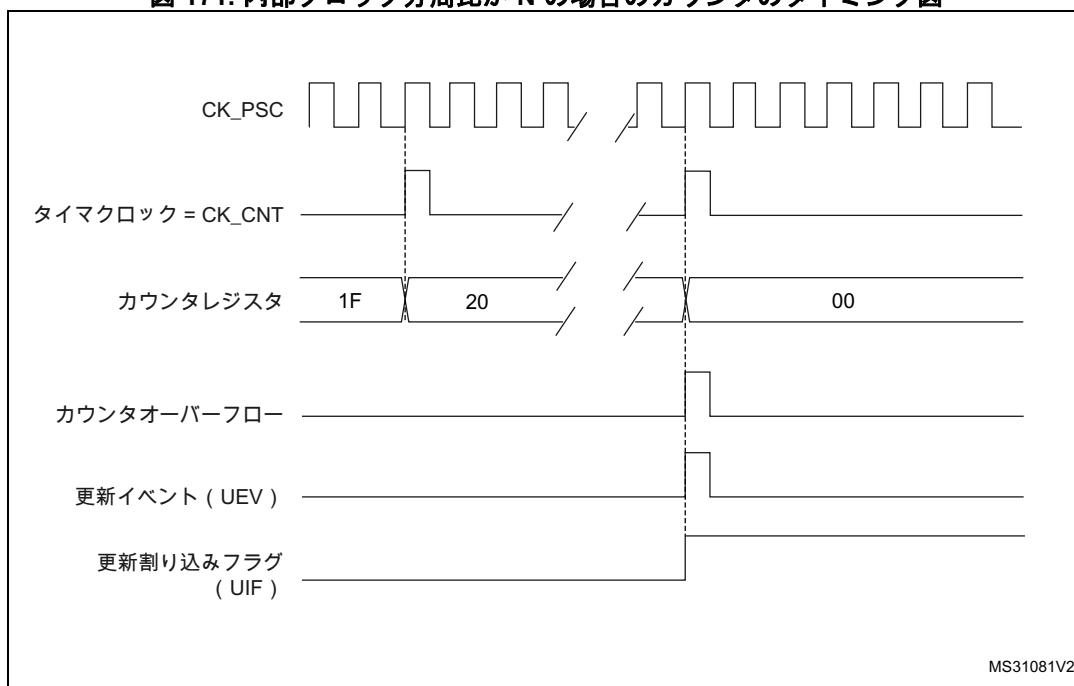


図 172. ARPE=0 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロードされない)

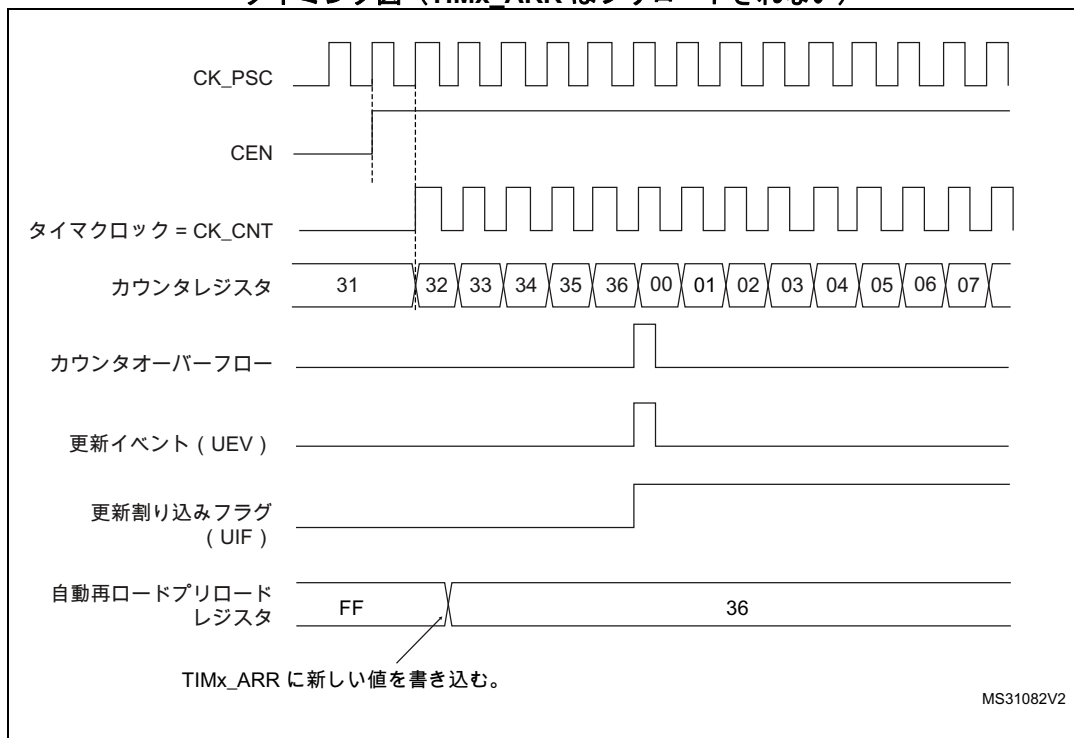
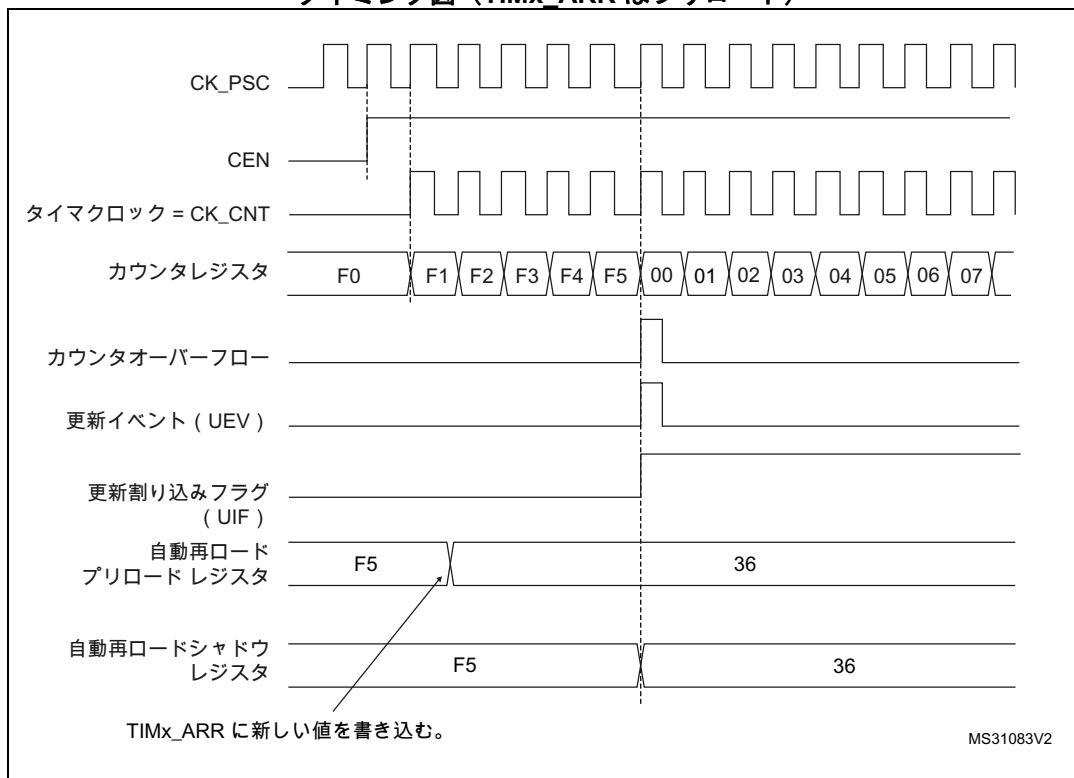


図 173. ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)



19.4.3 繰り返しカウンタ

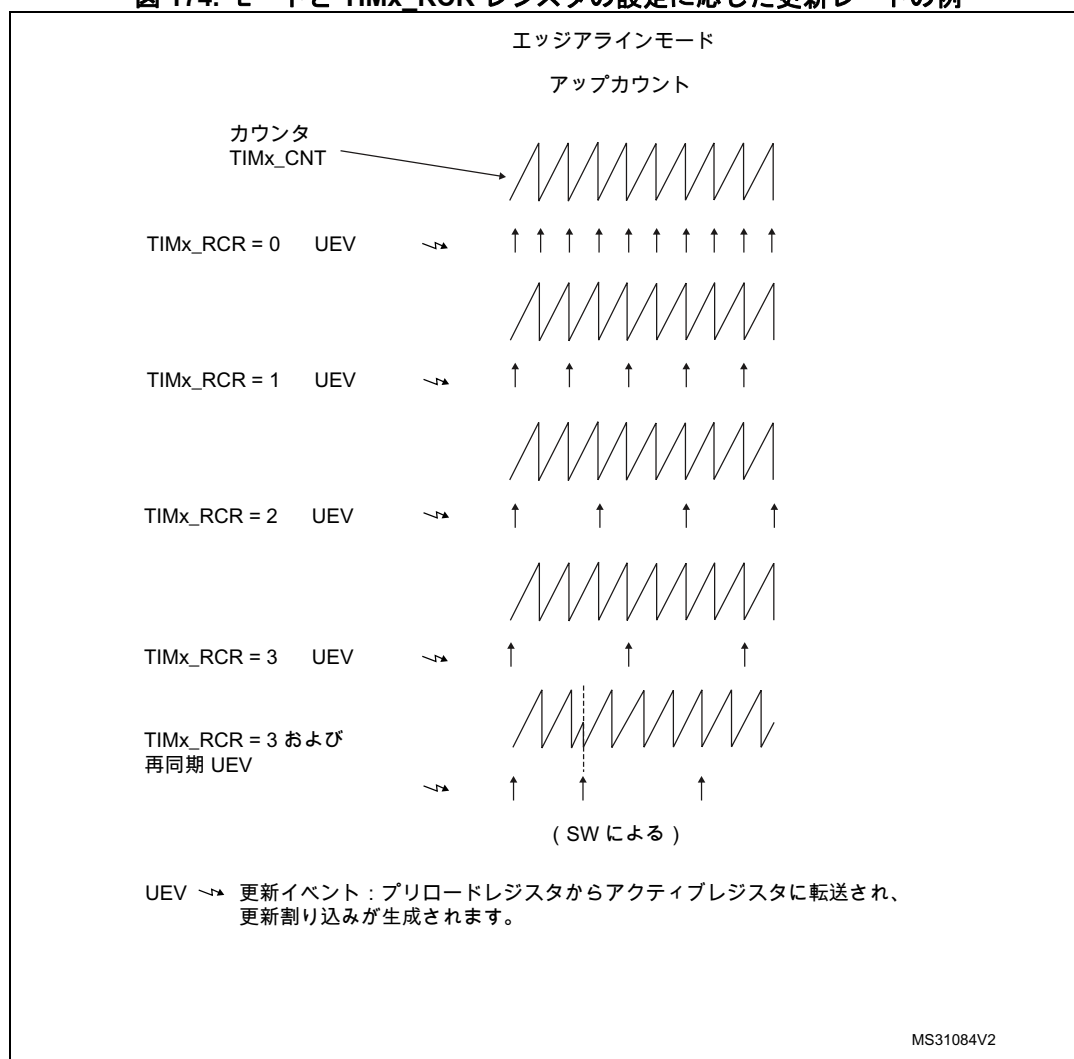
[セクション 18.3.1: タイムベースユニット](#)に、カウンタオーバーフロー/アンダーフローによって、どのように更新イベント (UEV) が生成されるかが説明されています。実際には、繰り返しカウンタが 0 に達したときにのみ、更新イベントが生成されます。これは、PWM 信号を生成する際に役立ちます。

これは、TIMx_RCR 繰り返しカウンタレジスタの値を N とすると、N 回目のカウンタオーバーフローまたはアンダーフローごとに、プリロードレジスタからシャドウレジスタ (TIMx_ARR 自動再ロードレジスタ、TIMx_PSC プリスケアラレジスタ、比較モードの TIMx_CCRx キャプチャ/比較レジスタ) ヘデータが転送されることを意味します。

繰り返しカウンタは、アップカウントモードで、カウンタオーバーフローごとにデクリメントされません。

繰り返しダウンカウンタは自動再ロードタイプです。繰り返しの回数は、TIMx_RCR レジスタの値によって定義されたとおりに維持されます ([図 174](#) を参照してください)。ソフトウェアによって (TIMx_EGR レジスタの UG ビットをセットすることによって)、またはスレーブモードコントローラを介してハードウェアによって更新イベントが生成されると、繰り返しカウンタの値にかかわらず直ちにイベントが発生し、繰り返しカウンタに TIMx_RCR レジスタの内容が再ロードされます。

図 174. モードと TIMx_RCR レジスタの設定に応じた更新レートの例



19.4.4 クロックソース

カウンタクロックは、次のクロックソースによって供給されます。

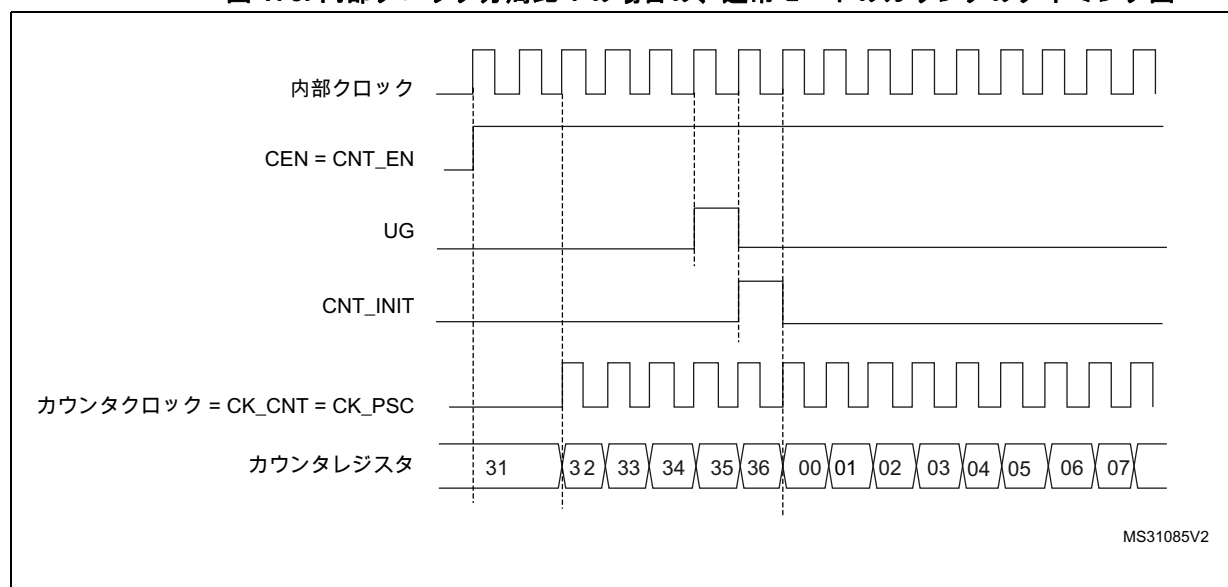
- 内部クロック (CK_INT)
- 外部クロックモード 1: 外部入力ピン (TIM15 の場合のみ)
- 内部トリガ入力 (ITRx) (TIM15 の場合のみ): あるタイマを別のタイマのプリスケアラとして使用します。たとえば、TIM1 が TIM15 のプリスケアラとして機能するように設定できます。詳細については、[タイマを別のタイマのプリスケアラとして使用する](#) を参照してください。

内部クロックソース (CK_INT)

TIM5 で、スレーブモードコントローラが無効の場合 (SMS=000)、CEN、DIR (TIMx_CR1 レジスタ)、および UG ビット (TIMx_EGR レジスタ) が実際の制御ビットとなり、ソフトウェアによってのみ変更できます (自動的にクリア状態に保たれる UG ビットを除きます)。CEN ビットに 1 が書き込まれると、プリスケアラにはクロックとして内部クロック CK_INT が供給されます。

[図 18.3.4](#) に、プリスケアラを使用しない場合の制御回路と通常モードのアップカウンタの動作を示します。

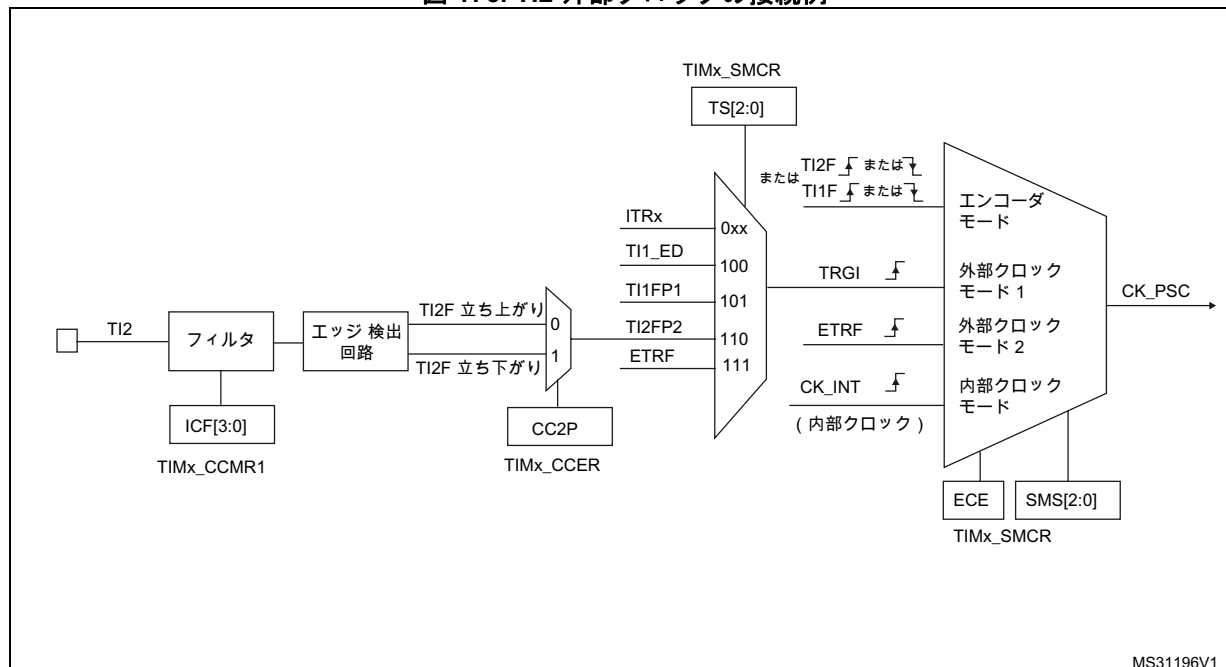
図 175. 内部クロック分周比 1 の場合の、通常モードのカウンタのタイミング図



外部クロックソースモード 1

このモードは TIMx_SMCR レジスタの SMS=111 のときに選択されます。カウンタは、選択された入力の立ち上がりまたは立ち下がりエッジでカウントすることができます。

図 176. TI2 外部クロックの接続例



たとえば、TI2 入力の立ち上がりエッジに反応してカウントするようにアップカウンタを設定するには、次の手順で行います。

1. TIMx_CCMR1 レジスタの CC2S ビットに“01”を書き込むことによって、チャンネル 2 が TI2 入力の立ち上がりエッジを検出するように設定します。
2. TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、入力フィルタ時間を設定します（フィルタを使用しない場合は、IC2F=0000 にしておきます）。
3. CC2P=0 を TIMx_CCER レジスタに書き込んで、立ち上がりエッジ極性を選択します。
4. TIMx_SMCR レジスタに SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します。
5. TIMx_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を選択します。
6. TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。

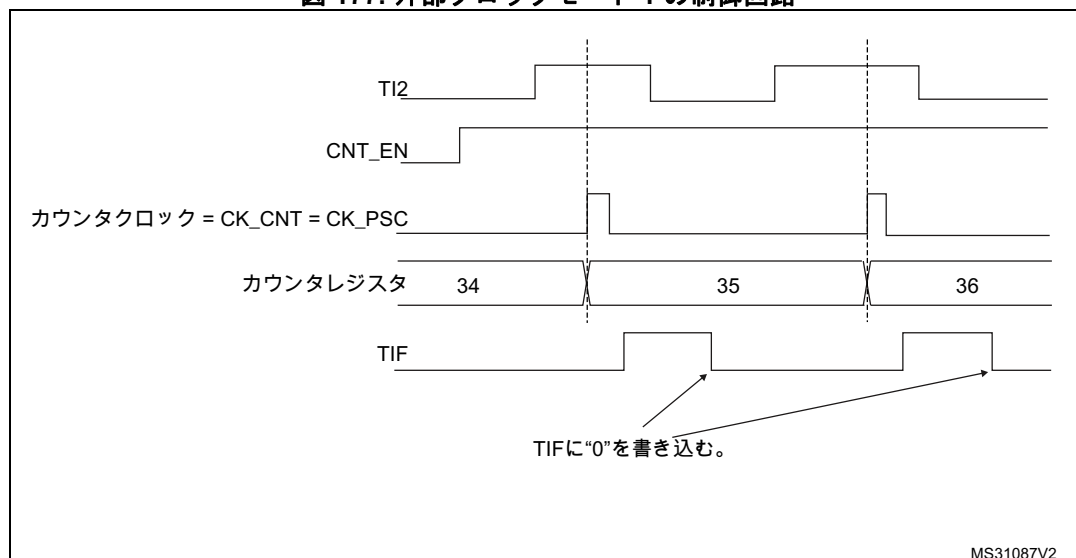
注： キャプチャプリスケラはトリガには使用されないの、設定は不要です。

コード例については、「付録」セクション **A.9.1 : TI2 の立ち上がりエッジのアップカウンタのコード例** を参照してください。

TI2 の立ち上がりエッジが発生すると、カウンタは 1 カウントを行い、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際のカウンタクロックまでの間には、TI2 入力の再同期回路による遅延があります。

図 177. 外部クロックモード 1 の制御回路



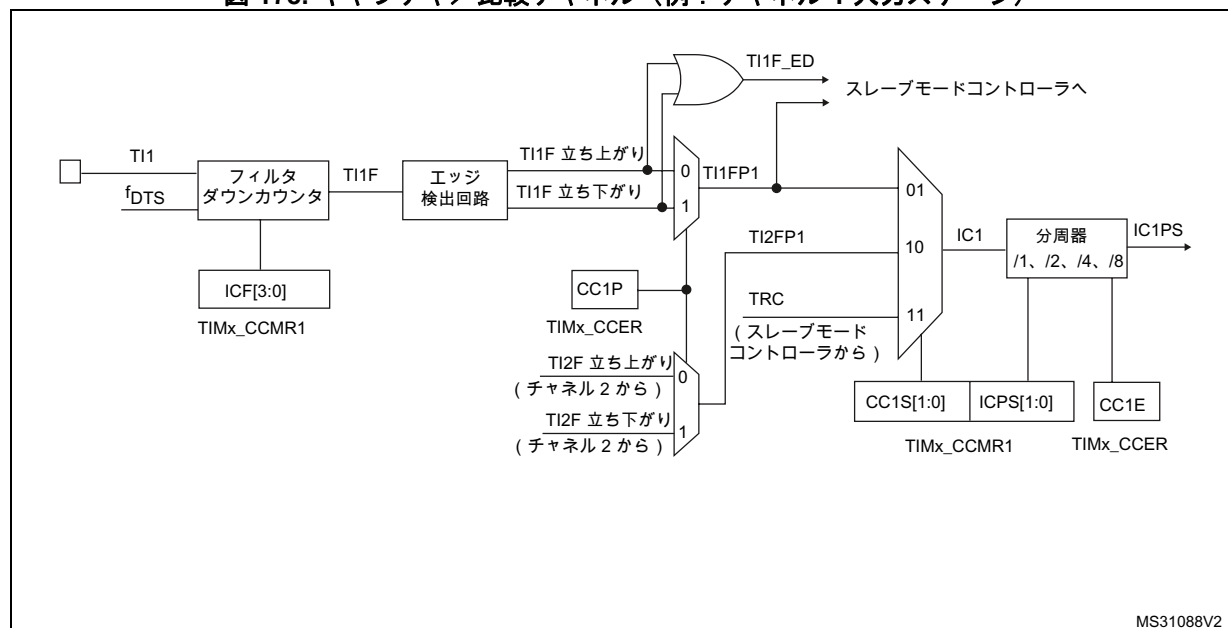
19.4.5 キャプチャ／比較チャネル

各キャプチャ／比較チャネルは、キャプチャ／比較レジスタ (シャドウレジスタを含む)、キャプチャの入カステージ (デジタルフィルタ、マルチプレクサ、プリスケラ)、および出力カステージ (比較回路と出力制御) から構成されています。

図 159 から 図 181 に、1 つのキャプチャ／比較チャネルの概要を示します。

入カステージは、対応する TIx 入力をサンプリングして、フィルタリングを行った TIxF を生成します。次に、極性選択付きのエッジ検出回路が、スレープモードコントローラによってトリガ入力として、またはキャプチャコマンドとして使用される信号 (TIxFPx) を生成します。この信号はプリスケラを通じて、キャプチャレジスタ (ICxPS) に渡されます。

図 178. キャプチャ／比較チャネル (例：チャネル 1 入カステージ)



出力ステージは、OCxRef（アクティブハイ）として使用される中間波形を生成します。信号の極性は最終出力に影響を与えます。

図 179. キャプチャ／比較チャンネル 1 メイン回路

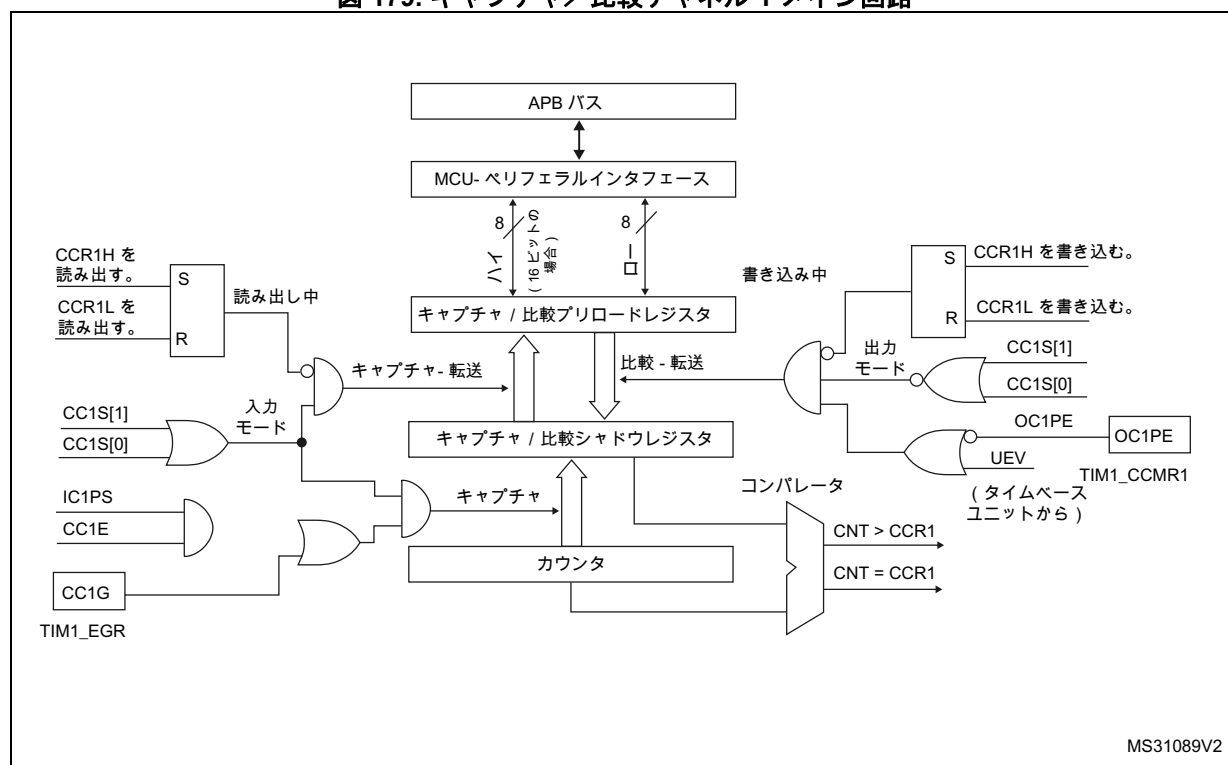


図 180. キャプチャ／比較チャンネル（チャンネル 1）の出力ステージ

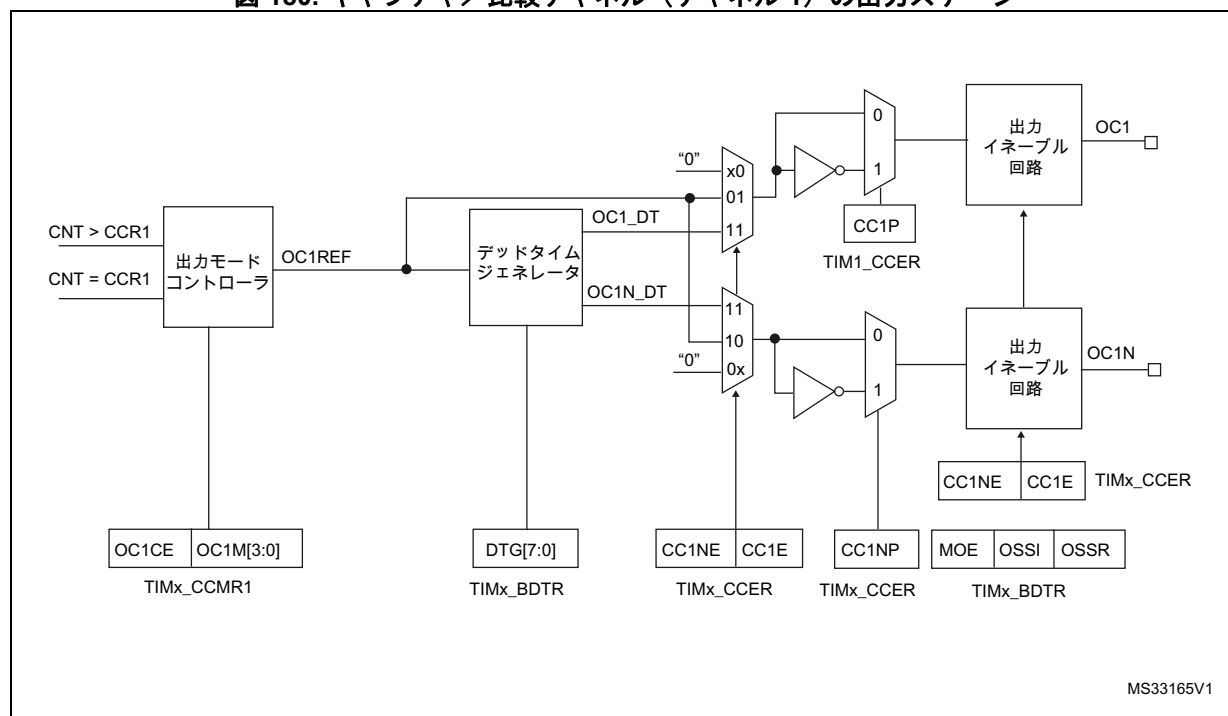
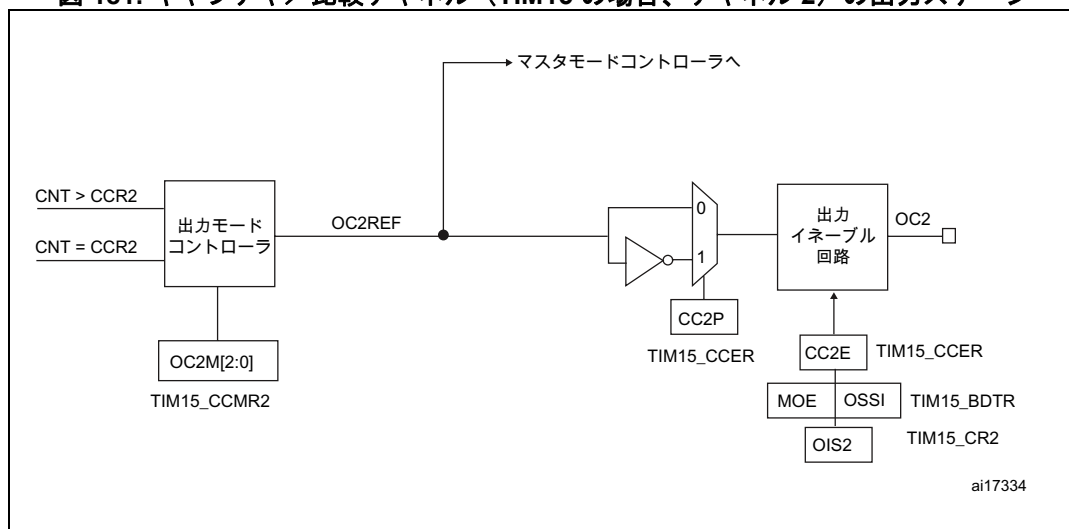


図 181. キャプチャ／比較チャネル (TIM15 の場合、チャネル 2) の出力ステージ



キャプチャ／比較ブロックは、1つのプリロードレジスタと1つのシャドウレジスタで構成されています。書き込みおよび読み出しアクセスは、常にプリロードレジスタに対して行われます。

キャプチャモードでは、キャプチャ動作は実際にはシャドウレジスタで行われ、その値がプリロードレジスタにコピーされます。

比較モードでは、プリロードレジスタの内容がシャドウレジスタにコピーされて、カウンタと比較されます。

19.4.6 入力キャプチャモード

入力キャプチャモードでは、対応する ICx 信号によって変化が検出された後、カウンタの値をラッチするために、キャプチャ／比較レジスタ (TIMx_CCRx) が使用されます。キャプチャが発生すると、対応する CCxIF フラグ (TIMx_SR レジスタ) がセットされ、割り込みまたは DMA リクエストを送信できます (有効な場合)。CCxIF フラグがすでにハイのときにキャプチャが発生した場合は、オーバキャプチャフラグ CCxOF (TIMx_SR レジスタ) がセットされます。CCxIF フラグは、ソフトウェアで“0”を書き込むことによって、または、TIMx_CCRx レジスタに格納されたキャプチャデータを読み出すことによってクリアできます。CCxOF は、“0”を書き込むとクリアされます。

次の例は、TI1 入力立ち上がったときに、カウンタの値を TIMx_CCR1 にキャプチャする方法を示します。このためには、次の手順を使用します。

- アクティブ入力を選択します。TIMx_CCR1 は TI1 入力とリンクされていなければならないので、このためには TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込みます。CC1S の値が“00”から変化すると、チャネルは入力に設定され、TIMx_CCR1 レジスタは読み出し専用になります。
- タイマに接続する信号に関して、必要な入力フィルタ時間をプログラムします (入力が TIx の 1 つである場合、TIMx_CCMRx レジスタの ICxF ビット)。入力信号の反転時、最大で内部クロックの 5 サイクルの間、信号が安定しないと想定してみます。この場合、フィルタ時間を 5 クロックサイクルより長くプログラミングする必要があります。新しいレベルの連続した 8 個のサンプルが検出されたときに、TI1 で遷移を検証できます (周波数 f_{DTS} でサンプリング)。この場合、TIMx_CCMR1 レジスタの IC1F ビットに 0011 を書き込みます。
- TI1 チャネルのアクティブ遷移のエッジを選択します。このためには、TIMx_CCER レジスタの CC1P ビットに 0 を書き込みます (この場合、立ち上がりエッジの選択)。
- 入力プリスケアラをプログラムします。この例では有効な遷移ごとにキャプチャを行いたいのので、プリスケアラを無効にします (TIMx_CCMR1 レジスタの IC1PS ビットに“00”を書き込む)。

- TIMx_CCER レジスタの CC1E ビットをセットすることによって、カウンタからキャプチャレジスタへのキャプチャを有効にします。
- 必要場合は、TIMx_DIER レジスタの CC1IE ビットをセットすることによって、関連する割り込みリクエストを有効にするか、TIMx_DIER レジスタの CC1DE レジスタをセットすることによって、DMA リクエストを有効にします。

コード例については、「付録」セクション [A.9.3: 入力キャプチャ設定のコード例](#) を参照してください。

入力キャプチャが発生すると、

- アクティブ遷移時に、カウンタの値が TIMx_CCR1 レジスタに格納されます。
- CC1IF フラグがセットされます (割り込みフラグ)。CC1OF ビットは、少なくとも 2 回連続でキャプチャが発生した場合にもセットされますが、フラグはクリアされません。
- CC1IE ビットに応じて、割り込みが生成されます。
- CC1DE ビットに応じて、DMA リクエストが生成されます。

コード例については、「付録」セクション [A.9.4: 入力キャプチャのデータ管理のコード例](#) を参照してください。

オーバキャプチャを処理するために、オーバキャプチャフラグの前にデータを読み出すことが推奨されます。これにより、フラグ読み出し後、データ読み出し前に発生するオーバキャプチャの見落としを避けることができます。

注: IC 割り込みと DMA リクエストは、TIMx_EGR レジスタの対応する CCxG ビットをセットすることによって、ソフトウェアによって生成することができます。

19.4.7 PWM 入力モード (TIM15 の場合のみ)

このモードは、入力キャプチャモードの特殊ケースです。操作手順は入力キャプチャモードと同様ですが、以下の点が異なります。

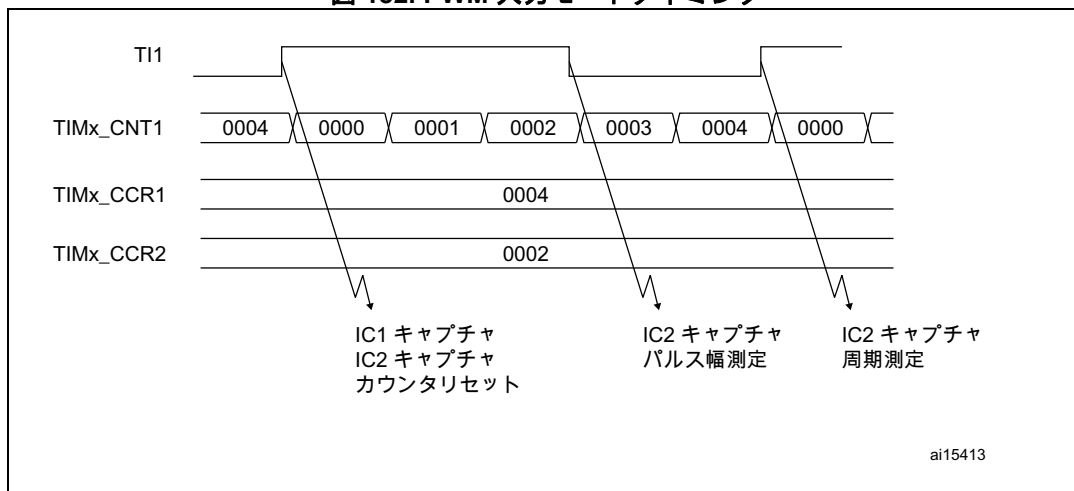
- 2 つの ICx 信号が同じ TIx 入力にマッピングされます。
- この 2 つの ICx 信号は、逆の極性のエッジでアクティブです。
- 2 つの TIxFP 信号の 1 つがトリガ入力として選択され、スレーブモードコントローラはリセットモードに設定されます。

たとえば、次の手順を使用して、TI1 に適用された PWM の周期 (TIMx_CCR1 レジスタ) とデューティサイクル (TIMx_CCR2 レジスタ) を測定できます (手順は、CK_INT 周波数とプリスケアラ値によって、若干異なることがあります)。

- TIMx_CCMR1 レジスタの CC1S ビットに“01”を書き込むことによって (TI1 を選択)、TIMx_CCR1 のアクティブ入力を選択します。
- CC1P ビットに“0”を書き込むことによって (立ち上がりエッジでアクティブ)、TI1FP1 のアクティブ極性を選択します (TIMx_CCR1 のキャプチャとカウンタクリアの両方に使用)。
- TIMx_CCMR1 レジスタの CC2S ビットに“10”を書き込むことによって (TI1 を選択)、TIMx_CCR2 のアクティブ入力を選択します。
- CC2P ビットに“1”を書き込むことによって (立ち下がりエッジでアクティブ)、TI1FP2 のアクティブ極性を選択します (TIMx_CCR2 のキャプチャに使用されます)。
- TIMx_SMCR レジスタの TS ビットに 101 を書き込むことによって (TI1FP1 を選択)、有効なトリガ入力を選択します。
- TIMx_SMCR レジスタの SMS ビットに 100 を書き込むことによって、スレーブモードコントローラをリセットモードに設定します。
- TIMx_CCER レジスタの CC1E と CC2E ビットに“1”を書き込むことによって、キャプチャを有効にします。

コード例については、「付録」セクション [A.9.5 : PWM 入力設定のコード例](#) を参照してください。

図 182. PWM 入力モードタイミング



1. TI1FP1 と TI2FP2 のみがスレーブモードコントローラに接続されているので、PWM 入力モードは TIMx_CH1/TIMx_CH2 信号でのみ使用できます。

19.4.8 強制出力モード

出力モード (TIMx_CCMRx レジスタの CCxS ビット = 00) では、出力比較レジスタとカウンタの間の比較に関係なく、各出力比較信号 (OCxREF と OCx/OCxN) をソフトウェアによって直接、強制的にアクティブまたはインアクティブレベルにできます。

出力比較信号 (OCxREF/OCx) を強制的にアクティブレベルとするには、対応する TIMx_OCMRx レジスタの OCxM ビットに“101”を書き込みます。これにより、OCxREF は強制的にハイになり (OCxREF は常にアクティブハイ)、OCx は CCxP 極性ビットと逆の値になります。

例: CCxP=0 (OCx アクティブハイ) => OCx は強制的にハイレベルになります。

OCxREF 信号は、TIMx_CCMRx レジスタの OCxM ビットに“100”を書き込むことによって、強制的にローにできます。

いずれにしても、TIMx_CCRx シャドウレジスタとカウンタの比較は実行されるので、フラグをセットできます。それに応じて、割り込みや DMA リクエストを送信できます。これについては、次の出力比較モードのセクションで説明します。

19.4.9 出力比較モード

この機能は、出力波形を制御したり、一定時間が経過したことを示すために使用されます。

キャプチャ/比較レジスタとカウンタの値が一致すると、出力比較は次のように機能します。

- 対応する出力ピンに、出力比較モード (TIMx_CCMRx レジスタの OCxM ビット) と出力極性 (TIMx_CCER レジスタの CCxP ビット) によって定義されたプログラム可能値を割り当てます。一致した際、出力ピンは、レベルを維持するか (OCxM=000)、アクティブにセットされるか (OCxM=001)、非アクティブにセットされるか (OCxM=010)、または反転されます (OCxM=011)。
- 割り込みステータスレジスタのフラグをセットします (TIMx_SR レジスタの CCxIF ビット)。
- 対応する割り込みマスク (TIMx_DIER レジスタの CCxIE ビット) がセットされている場合は、割り込みを生成します。

- 対応するイネーブルビット (TIMx_DIER レジスタの CCxDE ビット) がセットされている場合は、DMA リクエストを送信します (DMA リクエスト選択には、TIMx_CR2 レジスタの CCDS ビットが使用されます)。

TIMx_CCRx レジスタは、プリロードレジスタを使用するしないにかかわらず、TIMx_CCMRx レジスタの OCxPE ビットを使用してプログラミングできます。

出力比較モードでは、更新イベント UEV は OCxREF および OCx 出力には影響を与えません。タイミングの分解能はカウンタの 1 カウント分です。出力比較モードは単一パルスを出力するためにも使用できます (ワンパルスモード)。

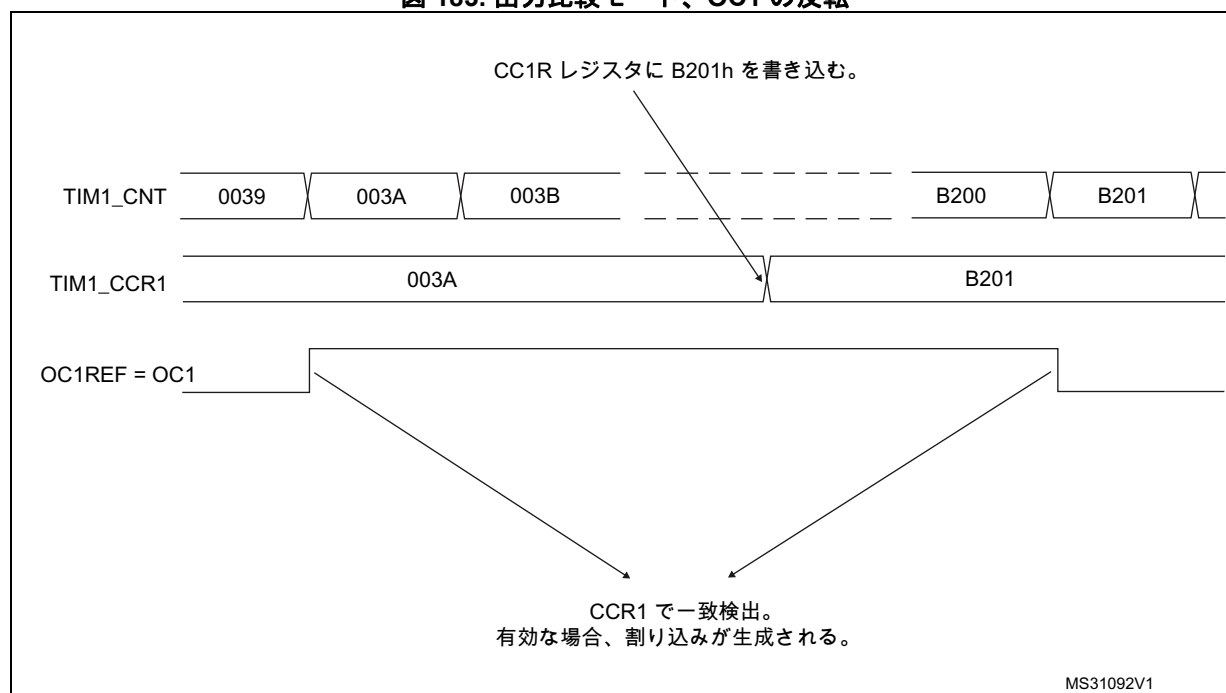
手順：

1. カウンタクロックを選択します (内部、外部、プリスケアラ)。
2. TIMx_ARR レジスタと TIMx_CCRx レジスタに目的のデータを書き込みます。
3. 割り込みリクエストを生成する場合は、CCxIE ビットをセットします。
4. 出力モードを選択します。例：
 - CNT と CCRx が一致したときに OCx 出力ピンを反転するには、OCxM ビットに 011 を書き込みます。
 - プリロードレジスタを無効にするには、OCxPE ビットに 0 を書き込みます。
 - アクティブハイ極性を選択するには、CCxP ビットに 0 を書き込みます。
 - 出力を有効にするには、CCxE ビットに 1 を書き込みます。
5. TIMx_CR1 レジスタの CEN ビットをセットすることによって、カウンタを有効にします。

コード例については、「付録」セクション [A.9.2 : 2 回の ETR の立ち上がりエッジごとのアップカウンタのコード例](#) を参照してください。

いつでもソフトウェアによって TIMx_CCRx レジスタを更新して、出力波形を制御できます。ただし、プリロードレジスタが有効でない場合に限り (OCxPE=0)。そうでない場合、TIMx_CCRx シャドウレジスタは、次の更新イベント UEV でのみ更新されます。例を [図 162](#) に示します。

図 183. 出力比較モード、OC1 の反転



19.4.10 PWM モード

パルス幅変調(PWM)モードでは、TIMx_ARR レジスタの値によって決められた周波数と TIMx_CCRx レジスタの値によって決められたデューティサイクルで信号を生成できます。

PWM モードは、TIMx_CCMRx レジスタの OCxM ビットに“110” (PWM モード 1) または“111” (PWM モード 2) を書き込むことによって、チャンネルごとに選択できます (OCx 出力ごとに 1 つの PWM)。TIMx_CCMRx レジスタの OCxPE ビットをセットすることによって、対応するプリロードレジスタを有効にする必要があります。また、TIMx_CR1 レジスタの ARPE ビットをセットすることによって、自動再ロードプリロードレジスタも (アップカウントまたはセンターアラインモードで) 有効にする必要があります。

プリロードレジスタは、更新イベントが発生したときにのみシャドウレジスタに転送されるので、カウンタを開始する前に、TIMx_EGR レジスタの UG ビットをセットすることによって、すべてのレジスタを初期化しておく必要があります。

OCx 極性は、TIMx_CCER レジスタの OCxP ビットを使用して、ソフトウェアでプログラム可能です。アクティブハイまたはアクティブローとしてプログラムできます。OCx 出力は、CCxE、CCxNE、MOE、OSSI、および OSSR ビット (TIMx_CCER および TIMx_BDTR レジスタ) の組み合わせによって有効になります。詳細については、TIMx_CCERx レジスタの説明を参照してください。

PWM モード (1 または 2) では、TIMx_CNT と TIMx_CCRx が常に比較されて、TIMx_CCRx ≤ TIMx_CNT または TIMx_CNT ≤ TIMx_CCRx かどうか判断されます (カウントの方向によります)。

タイマは、TIMx_CR1 レジスタの CMS ビットに応じて、エッジアラインモードまたはセンターアラインモードで PWM を生成できます。

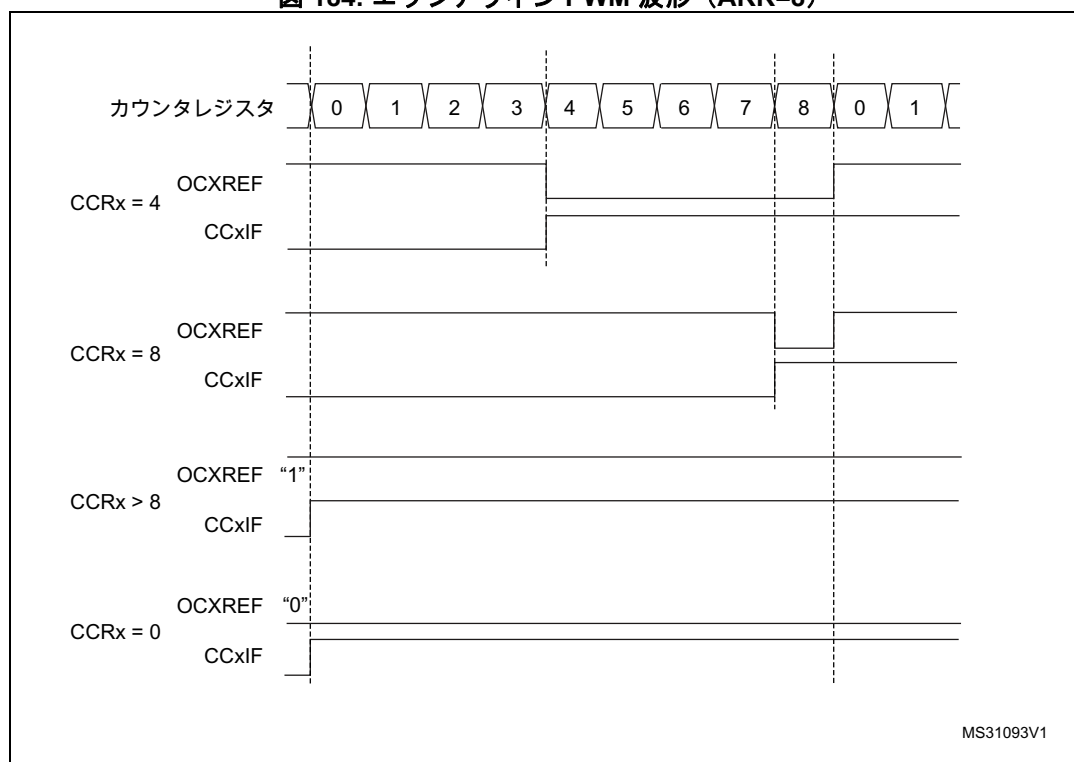
PWM エッジアラインモード

- アップカウント構成

TIMx_CR1 レジスタの DIR ビットがローのときには、アップカウントがアクティブです。[アップカウントモード \(447 ページ\)](#) を参照してください。

次の例では、PWM モード 1 を使用しています。PWM 基準信号 OCxREF は、TIMx_CNT < TIMx_CCRx の間はハイに、そうでない場合はローになります。TIMx_CCRx の比較値が自動再ロード値 (TIMx_ARR レジスタの) より大きい場合、OCxREF は“1”に保持されます。比較値が 0 の場合、OCxREF は“0”に保持されます。[図 163](#) に TIMx_ARR=8 のときのエッジアライン PWM 波形の例を示します。

図 184. エッジライン PWM 波形 (ARR=8)



コード例については、「付録」セクション [A.9.9 : センターアライン PWM の設定例](#) を参照してください。

- ダウンカウント構成

TIMx_CR1 レジスタの DIR ビットがハイのときには、ダウンカウントがアクティブです。 [繰り返しカウンタ \(476 ページ\)](#) を参照してください。

PWM モード 1 では、基準信号 OCxRef は、TIMx_CNT > TIMx_CCRx の間はローであり、そうでない場合はハイになります。TIMx_CCRx の比較値が TIMx_ARR の自動再ロード値より大きい場合、OCxREF は“1”です。このモードでは、0 % の PWM 信号を生成することはできません。

19.4.11 相補出力とデッドタイム挿入

TIM15/16/17 汎用タイマは、1 つの相補信号を出力して、出力時のスイッチオフおよびスイッチオンを管理できます。

この時間は、通常、デッドタイムと呼ばれ、出力に接続されているデバイスとその特性（レベルシフタの内在的な遅延、電源スイッチによる遅延など）に応じて調整する必要があります。

出力の極性（主出力 OCx または補 OCxN）は、出力ごとに独自に選択できます。これは TIMx_CCER レジスタの CCxP ビットおよび CCxNP ビットへの書き込みによって行います。

相補信号 OCx および OCxN は、TIMx_CCER レジスタの CCxE ビットと CCxNE ビット、TIMx_BDTR レジスタと TIMx_CR2 レジスタの MOE、OISx、OISxN、OSSI、および OSSR ビットといった複数の制御ビットの組み合わせによって有効になります。詳細については、[表 64: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能 \(509 ページ\)](#) を参照してください。特に、IDLE 状態に切り替わるとき（MOE が 0 になるときに）、デッドタイムが挿入されます。

デッドタイム挿入は、CCxE ビットと CCxNE ビットの両方をセットし、ブレーク回路がある場合は、さらに MOE ビットをセットすることによって有効になります。各チャンネルに 1 つの 10 ビットデッドタイムジェネレータがあります。この回路は、基準波形 OCxREF から OCx と OCxN の 2 つの出力を生成します。OCx と OCxN がアクティブハイの場合、

- OCx 出力信号は基準信号と同じですが、立ち上がりエッジが基準の立ち上がりエッジより遅い点が異なります。
- OCxN 出力信号は、立ち上がりエッジが基準波形の立ち下がりエッジから遅れている点を除けば、基準信号を反転させた波形と同じです。

遅延がアクティブ出力 (OCx または OCxN) の幅より大きい場合、対応するパルスは生成されません。

以下の図は、デッドタイム生成回路の出力信号と基準信号 OCxREF との関係を示します。(これらの例では、CCxP=0、CCxNP=0、MOE=1、CCxE=1、および CCxNE=1 を想定しています。)

図 185. デッドタイム挿入のある相補出力

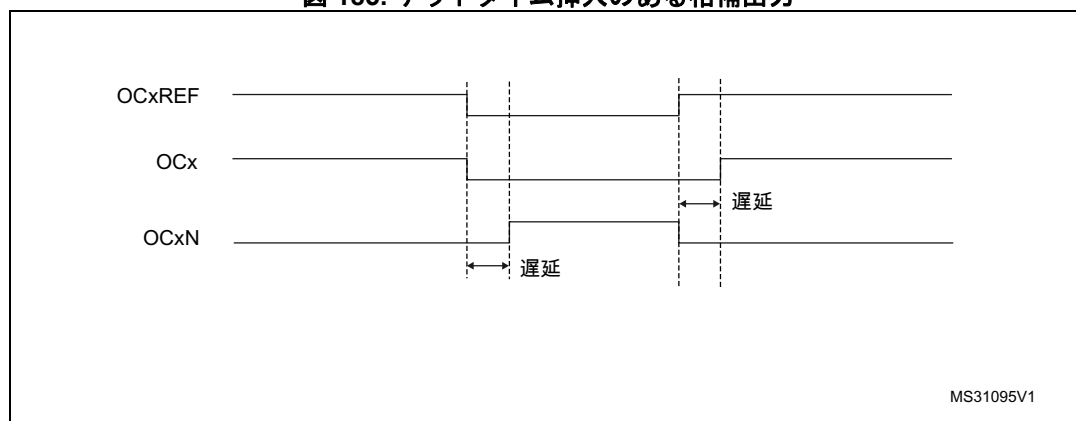


図 186. 負のパルスより長い遅延があるときのデッドタイムの波形

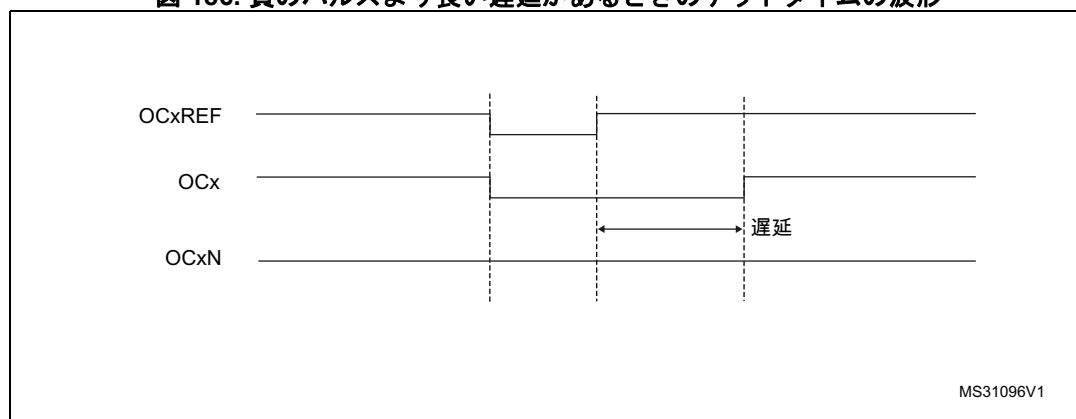
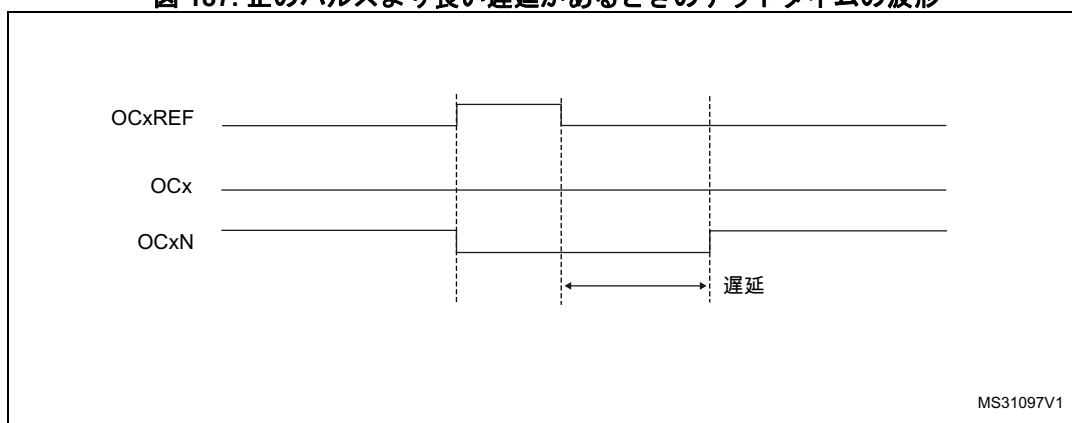


図 187. 正のパルスより長い遅延があるときのデッドタイムの波形



デッドタイム遅延は、各チャンネルで同じであり、TIMx_BDTR レジスタの DTG ビットでプログラミングできます。遅延計算については、[セクション 19.5.15 : TIM15 ブレークおよびデッドタイムレジスタ \(TIM15_BDTR\) \(512 ページ\)](#) を参照してください。

OCxREF 信号の OCx または OCxN へのリダイレクト

出力モード（強制、出力比較、または PWM）では、TIMx_CCER レジスタの CCxE ビットおよび CCxNE ビットを構成することによって、OCxREF 信号を OCx 出力または OCxN 出力にリダイレクトできます。

これにより、特定の波形（PWM または静的アクティブレベルなど）を一方の出力に送信し、補信号をインアクティブレベルに固定することができます。他の例としては、両方の出力をインアクティブレベルにしたり、両方の出力をアクティブにして、デッドタイムのある相補出力とすることができます。

注： OCxN のみが有効なときには (CCxE=0, CCxNE=1)、相補にならず、OCxREF がハイレベルとなるとアクティブになります。たとえば、CCxNP=0 の場合は、OCxN=OCxRef です。他方、OCx と OCxN の両方が有効なときには (CCxE=CCxNE=1)、OCxREF がハイになると OCx はアクティブになり、OCxREF がローのときには、OCxN は補信号であり、アクティブになります。

19.4.12 ブレーク機能の使用

ブレーク機能を使用しているときには、出力イネーブル信号とインアクティブレベルは追加の制御ビット (TIMx_BDTR レジスタの MOE、OSSI、および OSSR ビットと TIMxCR2 レジスタの OISx および OISxN ビット) に応じて変更されます。ただし、OCx および OCxN 出力の両方を同時にアクティブレベルに設定することはできません。詳細については、[表 64: ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能 \(509 ページ\)](#) を参照してください。

ブレークソースは、ブレーク入力ピンか、リセットクロックコントローラ (RCC) のクロックセキュリティシステム (CSS) によって生成されたクロック障害イベントです。クロックセキュリティシステムの詳細については、[セクション 7.2.8 : クロックセキュリティシステム \(CSS\)](#) を参照してください。

リセットが終了すると、ブレーク回路は無効になり、MOE ビットはローになります。TIMx_BDTR レジスタの BKE ビットをセットすることによって、ブレーク機能を有効にできます。ブレーク入力の極性は、同じレジスタの BKP ビットを設定することによって選択できます。BKE と BKP は、同時に変更できます。BKE および BKP ビットが書き込まれるとき、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が適用されます。そのため、書き込み動作の後、ビットを正しく読み出すためには 1 APB クロックサイクル待つ必要があります。

MOE の立ち下がりがエッジは非同期のことがあるので、実際の信号（出力に作用する信号）と同期制御ビット (TIMx_BDTR レジスタからアクセスできる) の間に、再同期回路が挿入されています。このため、非同期信号と同期信号の間に若干の遅延が発生します。特に、MOE がローになった後で 1 が書き込まれた場合、MOE を正しく読み出すためには、遅延（ダミー命令）を挿入する必要があります。これは、非同期信号を書き込んで、同期信号を読み出すからです。

ブレークが発生すると（ブレーク入力で選択されたレベル）、

- MOE ビットは非同期にクリアされ、出力は、インアクティブ状態、アイドル状態、またはリセット状態 (OSSI ビットで選択) になります。これは、MCU オシレータがオフの場合も同様です。
- 各出力チャネルは、MOE=0 になったとき、TIMx_CR2 レジスタの OISx ビットでプログラミングされたレベルで駆動されます。OSSI=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はハイのままです。
- 相補出力が使用されているときには：
 - － 出力は、まずリセット状態のインアクティブ状態に置かれます（極性に依存します）。これは非同期に行われるので、タイマにクロックが供給されていないときでも機能します。
 - － タイマクロックが供給されている場合、デッドタイム後に OISx および OISxN ビットでプログラミングされたレベルで出力を駆動するために、デッドタイムジェネレータが作動します。この場合でも、OCx と OCxN を同時にアクティブレベルに駆動することはできません。MOE の再同期により、デッドタイム時間が通常より少し長くなることに注意してください（約 2 CK_TIM クロックサイクル）。
 - － OSSI=0 の場合、タイマはイネーブル出力を解除し、そうでない場合、イネーブル出力はそのままか、CCxE または CCxNE ビットがハイになったときにハイになります。
- ブレーク状態フラグ (TIMx_SR レジスタの BIF ビット) がセットされます。TIMx_DIER レジスタの BIE ビットがセットされている場合は、割り込みを生成できます。TIMx_DIER レジスタの BDE ビットがセットされている場合、DMA リクエストを送信できます。
- TIMx_BDTR レジスタの AOE ビットがセットされている場合、MOE ビットは次の更新イベント UEV で再び自動的にセットされます。これを使用して、たとえば、レギュレーションを行うことができます。そうでない場合、MOE ビットは、再び 1 が書き込まれるまでローのままです。この場合、セキュリティ目的で使用でき、パワー駆動回路、温度センサ、またはセキュリティコンポーネントからのアラームにブレーク入力を接続できます。

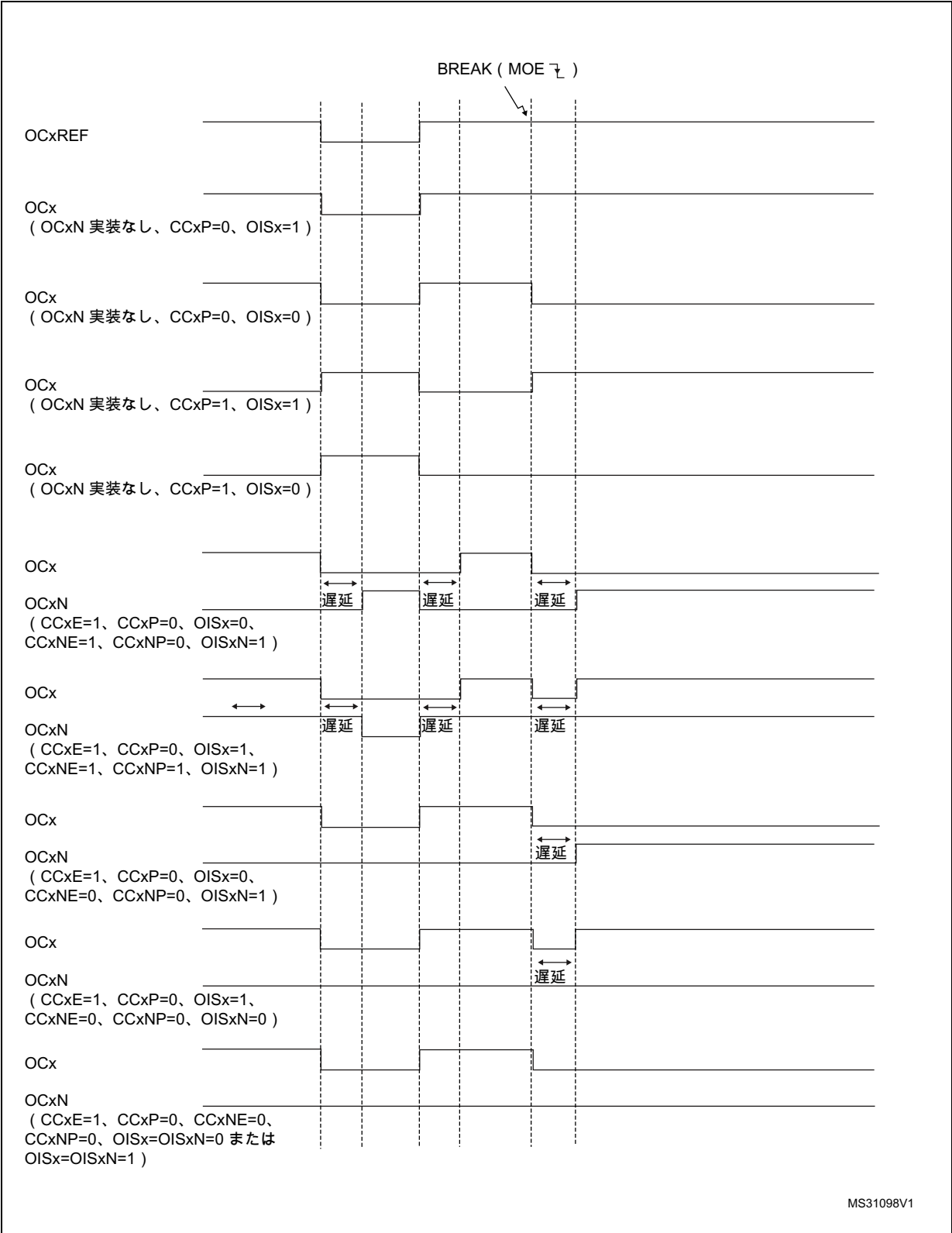
注： **ブレーク入力は、信号レベルに対して動作します。このため、ブレーク入力がアクティブな間は、MOE をセットできません（自動的にも、ソフトウェアによっても）。この間、ステータスフラグ BIF をクリアできません。**

ブレークは、BRK 入力によって生成でき、BRK はプログラミング可能な極性を持ち、TIMx_BDTR レジスタの BKE がイネーブルビットです。

ブレーク入力と出力管理に加えて、アプリケーションに対する安全策として、ブレーク回路内に書き込み保護機能を設けてあります。これにより、いくつかのパラメータ（デッドタイムの長さ、OCx/OCxN 極性、無効時の状態、OCxM 構成、ブレークイネーブルと極性）を固定することができます。TIMx_BDTR レジスタの LOCK ビットによって、3 レベルの保護を選択することができます。[セクション 19.5.15：TIM15 ブレークおよびデッドタイムレジスタ \(TIM15_BDTR\) \(512 ページ\)](#) を参照してください。LOCK ビットは、MCU リセット後に 1 回だけ書き込むことができます。

[図 188](#) に、ブレークに対する出力の動作例を示します。

図 188. ブレークに対する出力の動作



19.4.13 ワンパルスモード

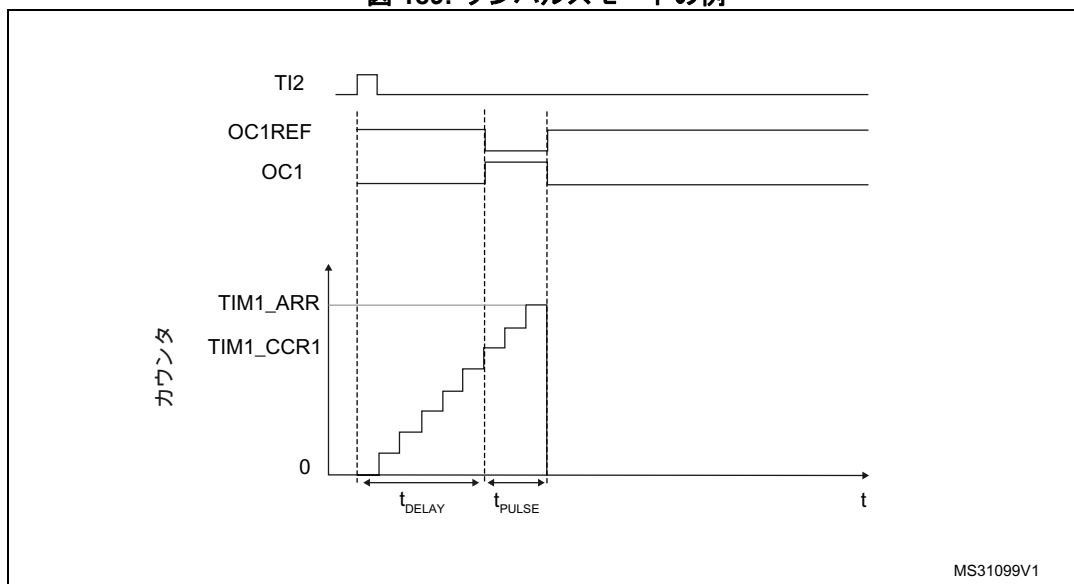
ワンパルスモード (OPM : One Pulse Mode) は、これまでに説明したモードの特殊ケースです。トリガに応じてカウンタを開始して、プログラム可能な遅延後にプログラム可能な長さのパルスを生成できます。

カウンタの開始は、スレーブモードコントローラを通じて制御できます。波形の生成は、出力比較モードまたは PWM モードで行うことができます。ワンパルスモードを選択するには、TIMx_CR1 レジスタの OPM ビットをセットします。これによって、カウンタは、次の更新イベント UEV で自動的に停止します。

パルスは、比較値がカウンタの初期値と異なる場合のみ、正しく生成されます。開始する前に (タイマがトリガを待っているときに)、設定が次のようであればなりません。

- アップカウント時 : $CNT < CCRx \leq ARR$ (特に、 $0 < CCRx$)
- ダウンカウント時 : $CNT > CCRx$

図 189. ワンパルスモードの例



たとえば、TI2 入力ピンで立ち上がりエッジが検出されたときに、OC1 にパルス幅が t_{PULSE} の正のパルスを遅延時間 t_{DELAY} 後に生成することもできます。

TI2FP2 をトリガ 1 として使用します。

- TIMx_CCMR1 レジスタの CC2S ビットに "01" を書き込むことによって、TI2FP2 を TI2 に配置します。
- TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタで CC2P="0" を書き込みます。
- TI2FP2 をスレーブモードコントローラのトリガ (TRGI) として設定します。このためには、TIMx_SMCR レジスタの TS ビットに "110" を書き込みます。
- TI2FP2 を使用してカウンタを開始します。このためには、TIMx_SMCR レジスタの SMS ビットに "110" (トリガモード) を書き込みます。

コード例については、「付録」セクション [A.9.16: ワンパルスモードのコード例](#) を参照してください。

OPM 波形は、次のように比較レジスタに書き込むことによって定義されます（クロック周波数とカウンタプリスケアラを考慮に入れて）。

- t_{DELAY} は、TIMx_CCR1 レジスタに書き込まれた値によって定義されます。
- t_{PULSE} は、自動再ロード値と比較値の差 (TIMx_ARR - TIMx_CCR1) によって定義されます。
- 比較一致が発生したときに 0 から 1 へ遷移し、カウンタが自動再ロード値に達したときに 1 から 0 へ遷移する波形を生成するとします。このためには、TIMx_CCMR1 レジスタの OC1M=111 を書き込むことによって、PWM モード 2 を有効にします。必要に応じて、TIMx_CCMR1 レジスタの OC1PE ビットに“1”を書き込み、TIMx_CR1 レジスタの ARPE ビットに書き込むことによって、プリロードレジスタを有効にすることもできます。この場合、TIMx_CCR1 レジスタに比較値を書き込み、TIMx_ARR レジスタに自動再ロード値を書き込みます。次に、UG ビットをセットすることによって更新を生成し、TI2 で外部トリガイイベントを待ちます。この例では、CC1P に“0”を書き込みます。

上の例では、TIMx_CR1 レジスタの DIR および CMS ビットはローでなければなりません。

必要なパルスは 1 つだけなので、TIMx_CR1 レジスタの OPM ビットに“1”を書き込みます。こうすると、カウンタは次の更新イベント時に停止します（カウンタが自動再ロード値に達して、“0”に戻る時点）。

特殊なケース：OCx 高速イネーブル

ワンパルスモードでは、TIx 入力のエッジ検出によって、カウンタを有効にする CEN ビットがセットされます。その後、カウンタと比較値の比較によって、出力が反転されます。ただし、このような動作には数クロックサイクルが必要なので、実現可能な最小遅延 ($t_{\text{DELAY min}}$) が制限されます。

最小遅延で波形を出力したい場合は、TIMx_CCMRx レジスタの OCxFE ビットをセットします。こうすると、OCxREF（および OCx）は、比較動作を行うことなく、強制的にトリガに反応します。新しいレベルは、比較が一致したときと同じです。OCxFE は、チャンネルが PWM1 または PWM2 モードに設定された場合のみ機能します。

コード例については、「付録」セクション [A.9.16：ワンパルスモードのコード例](#) に記載された PULSE_WITHOUT_DELAY > 0 によって条件付けられたコードの一部を参照してください。

19.4.14 TIM15 外部 トリガ同期

このセクションは、STM32F05x、STM32F07x、および STM32F09x デバイスにのみ適用されます。

TIM15 タイマは、いくつかのモードで外部トリガを使用して同期できます。そのモードは、リセットモード、ゲートモード、およびトリガモードです。

スレープモード：リセットモード

カウンタとそのプリスケアラは、トリガ入力のイベントに応じて再初期化できます。さらに、TIMx_CR1 レジスタの URS ビットがローの場合は、更新イベント UEV が生成されます。その場合、すべてのプリロードされたレジスタ (TIMx_ARR、TIMx_CCRx) が更新されます。

次の例では、TI1 入力の立ち上がりエッジに応じて、アップカウンタがクリアされます。

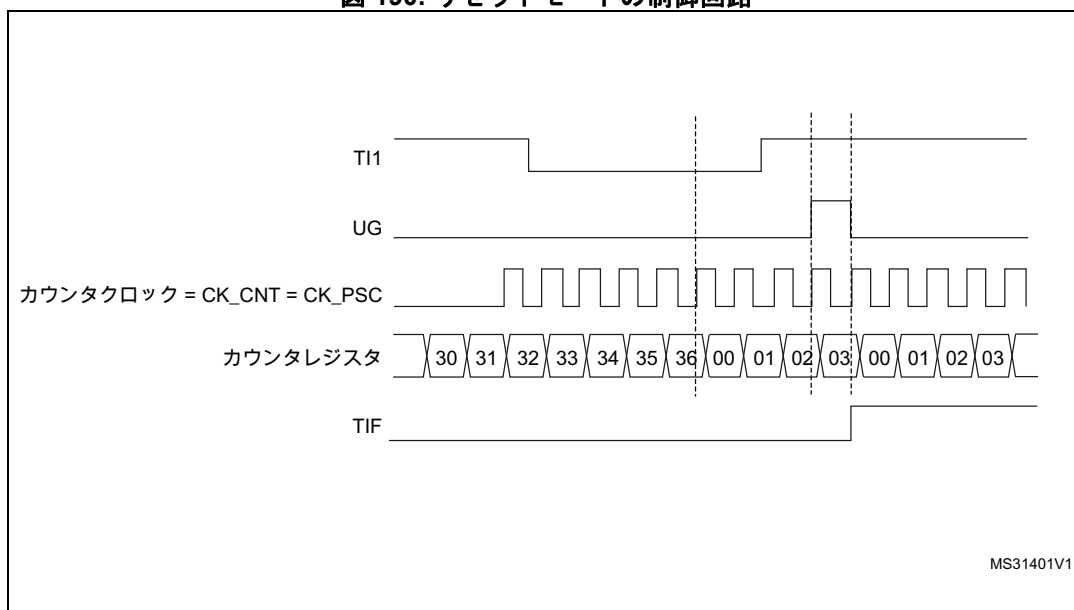
- TI1 の立ち上がりエッジを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します (この例ではフィルタは不要なので、IC1F=0000 のままにしておきます)。キャプチャプリスケアラはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します (TIMx_CCMR1 レジスタの CC1S=01)。TIMx_CCER レジスタに CC1P=0 を書き込んで、極性を有効にします (そして、立ち上がりエッジのみを検出します)。
- TIMx_SMCR レジスタに SMS=100 を書き込むことによって、タイマをリセットモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを開始します。

コード例については、「付録」セクション [A.9.12：リセットモードのコード例](#) を参照してください。

カウンタは内部クロックでカウントを開始し、TI1 の立ち上がりエッジまで通常の動作を行います。TI1 が立ち上がると、カウンタはクリアされ、0 からリスタートします。同時に、トリガフラグがセットされ (TIMx_SR レジスタの TIF ビット)、有効な場合は割り込みリクエストまたは DMA リクエストを送信できます (TIMx_DIER レジスタの TIE および TDE ビット)。

次の図は、自動再ロードレジスタ TIMx_ARR=0x36 の場合の動作を示します。TI1 の立ち上がりエッジから実際にカウンタがリセットされるまでの遅延は、TI1 入力の同期回路によるものです。

図 190. リセットモードの制御回路



スレーブモード：ゲートモード

選択された入力のレベルに応じて、カウンタを有効にできます。

次の例では、アップカウンタは TI1 入力が高レベルのときだけカウントします。

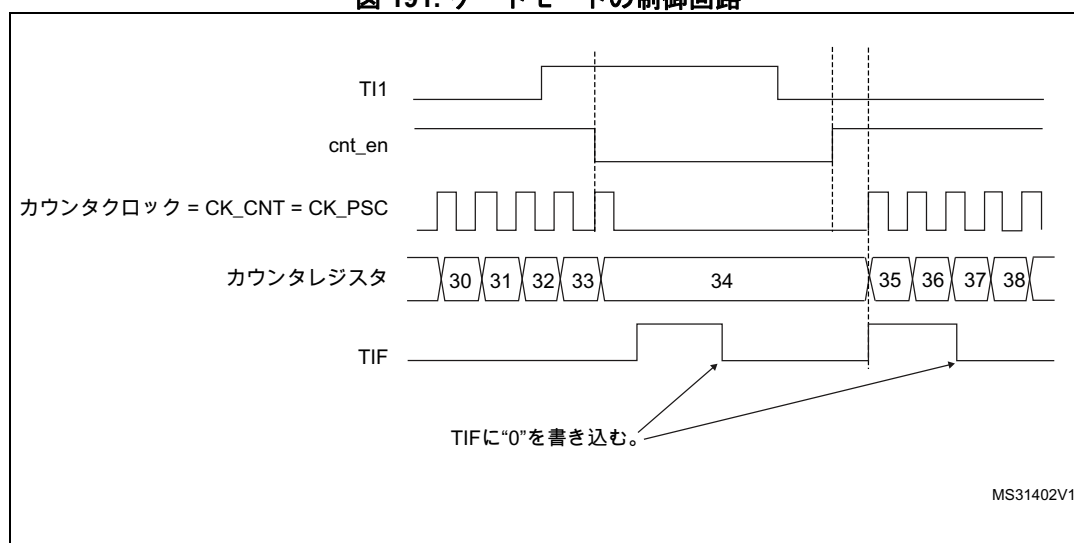
- TI1 のローレベルを検出するように、チャンネル 1 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC1F=0000 のままにしておきます）。キャプチャプリスケールはトリガには使用されないため、設定は不要です。CC1S ビットは、入力キャプチャソースのみを選択します（TIMx_CCMR1 レジスタの CC1S=01 ビット）。TIMx_CCER レジスタに CC1P=1 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=101 を書き込むことによって、タイマをゲートモードに設定します。TIMx_SMCR レジスタに TS=101 を書き込むことによって、入力ソースとして TI1 を選択します。
- TIMx_CR1 レジスタに CEN=1 を書き込んで、カウンタを有効にします（ゲートモードでは、CEN=0 の場合、トリガ入力のレベルにかかわらず、カウンタは開始しません）。

コード例については、「付録」セクション [A.9.13：ゲートモードのコード例](#) を参照してください。

カウンタは、TI1 がローになると内部クロックでカウントを開始して、TI1 がハイになると停止します。TIMx_SR レジスタの TIF フラグは、カウンタの開始時と停止時にセットされます。

TI1 の立ち上がりエッジから実際にカウンタが停止するまでの遅延は、TI1 入力の再同期回路によるものです。

図 191. ゲートモードの制御回路



スレーブモード：トリガモード

選択された入力のイベントに対応して、カウンタが開始できます。

次の例では、アップカウンタは、TI2 入力の立ち上がりエッジに応じて開始します。

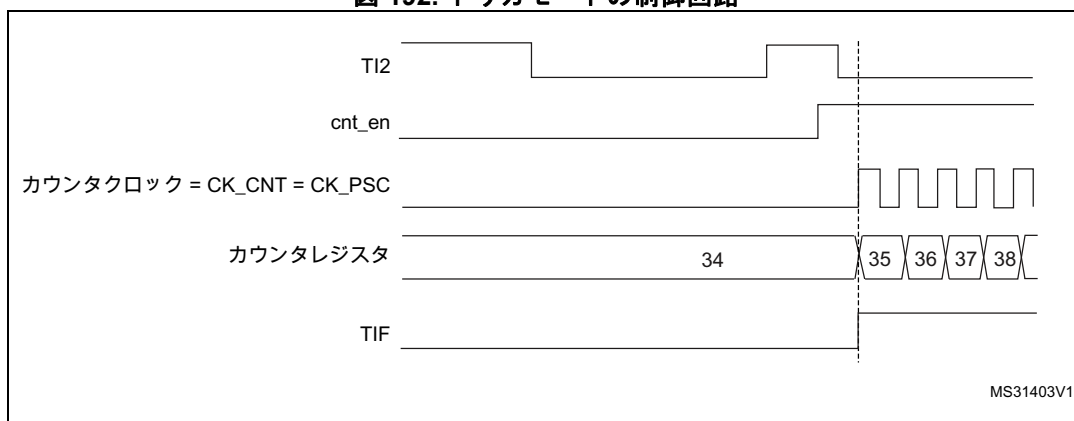
- TI2 の立ち上がりエッジを検出するように、チャンネル 2 を設定します。入力フィルタ時間を設定します（この例ではフィルタは不要なので、IC2F=0000 のままにしておきます）。キャプチャプリスケラはトリガには使用されないの、設定は不要です。CC2S ビットは入力キャプチャソースのみを選択するように設定されます (TIMx_CCMR1 レジスタの CC2S=01)。TIMx_CCER レジスタに CC2P=1 を書き込んで、極性を有効にします（そして、ローレベルのみを検出します）。
- TIMx_SMCR レジスタに SMS=110 を書き込むことによって、タイマをトリガモードに設定します。TIMx_SMCR レジスタに TS=110 を書き込むことによって、入力ソースとして TI2 を選択します。

コード例については、「付録」セクション [A.9.14：トリガモードのコード例](#) を参照してください。

TI2 で立ち上がりエッジが発生すると、カウンタは内部クロックでのカウントを開始し、TIF フラグがセットされます。

TI2 の立ち上がりエッジから実際にカウンタが開始するまでの遅延は、TI2 入力の再同期回路によるものです。

図 192. トリガモードの制御回路



TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 17.3.15：タイマの同期 \(414 ページ\)](#) を参照してください。

19.4.15 タイマ同期 (TIM15)

このセクションは、STM32F05x、STM32F07x、および STM32F09x デバイスにのみ適用されます。

TIM タイマは、タイマの同期または連結のために、内部で互いにリンクされます。詳細については、[セクション 17.3.15：タイマの同期 \(414 ページ\)](#) を参照してください。

19.4.16 デバッグモード

マイクロコントローラがデバッグモードになると (Cortex™-M0 コアは停止状態)、TIMx カウンタは、DBG モジュールの DBG_TIMx_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

19.5 TIM15 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

19.5.1 TIM15 制御レジスタ 1 (TIM15_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
						rw	rw	rw				rw	rw	rw	rw

ビット 15:10 予約済み、常に 0 として読み出されます。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (TIx) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

- 00 : $t_{DTS} = t_{CK\_INT}$
- 01 : $t_{DTS} = 2 * t_{CK\_INT}$
- 10 : $t_{DTS} = 4 * t_{CK\_INT}$
- 11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

- 0 : TIMx_ARR レジスタはバッファされません。
- 1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済み、常に 0 として読み出されます。

ビット 3 **OPM** : ワンパルスモード

- 0 : カウンタは更新イベントで停止しません。
- 1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット/クリアされます。

- 0 : 次のイベントのいずれかが更新割り込みを生成します (有効な場合)。これらのイベントは、次のとおりです。
 - カウンタオーバーフロー/アンダーフロー
 - UG ビットのセット
 - スレーブモードコントローラからの更新生成
- 1 : カウンタオーバーフロー/アンダーフローのみが更新割り込みを生成します (有効な場合)。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

- 0 : UEV は有効です。更新イベント (UEV) は、次のいずれかのイベントによって生成されます。
- カウンタオーバーフロー／アンダーフロー
 - UG ビットのセット
 - スレーブモードコントローラからの更新生成

パッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ (ARR、PSC、CCRx) は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

- 0 : カウンタは無効です。
1 : カウンタは有効です。

注： 外部クロックおよびゲートモードは、CEN ビットが事前にソフトウェアによってセットされている場合のみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

19.5.2 TIM15 制御レジスタ 2 (TIM15_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	Res.	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rw	rw	rw		rw	rw	rw	rw	rw		rw

ビット 15:11 予約済み、常に 0 として読み出されます。

ビット 10 **OIS2** : 出力アイドル状態 2 (OC2 出力)

- 0 : MOE = 0 のとき、OC2=0
1 : MOE = 1 のとき、OC2=0

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされている場合 (TIMx_BKR レジスタの LOCK ビット)、変更できません。

ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)

- 0 : MOE=0 のとき、デッドタイム後に OC1N=0
1 : MOE=1 のとき、デッドタイム後に OC1N=0

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BKR レジスタの LOCK ビット)。

ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)

- 0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)
1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

注： このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIMx_BKR レジスタの LOCK ビット)。

ビット 7 予約済み、常に 0 として読み出されます。



ビット 6:4 MMS[1:0] : マスタモード選択

これらのビットにより、同期のためにマスタモードでスレーブタイマに送信される情報を選択することができます (TRGO)。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが生成される場合 (スレーブモードコントローラがリセットモードに設定されているとき)、TRGO 信号は実際のリセットより遅延します。

001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除き、TRGO には遅延が存在しません (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として使用されます。たとえば、マスタタイマをスレーブタイマのプリスケールとして使用できます。

011 : **パルス比較** - キャプチャまたは比較一致が発生すると、CC1IF フラグがセットされる時 (すでにハイであった場合も)、トリガ出力は正のパルスを送信します (TRGO)。

100 : **比較** - OC1REF 信号がトリガ出力 (TRGO) として使用されます。

101 : **比較** - OC2REF 信号がトリガ出力 (TRGO) として使用されます。

ビット 3 CCDS : キャプチャ/比較 DMA 選択

0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。

1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。

ビット 2 CCUS : キャプチャ/比較制御更新選択

0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。

1 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 1 予約済み、常に 0 として読み出されます。

ビット 0 CCPC : キャプチャ/比較プリロード制御

0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。

1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込み後は、COM ビットがセットされた場合にのみ更新されます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

19.5.3 TIM15 のスレーブモード制御レジスタ (TIM15_SMCR)

アドレスオフセット : 0x08

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]				Res.	SMS[2:0]	
								rw	rw	rw	rw		rw	rw	rw

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7 **MSM** : マスタ/スレーブモード

0 : 影響なし。

1 : トリガ入力 (TRGI) に対するイベントの影響は、現在のタイマとそのスレーブとの間の完全な同期 (TRGO を通じて) を可能にするために遅延されます。これは、1 つの外部イベントで複数のタイマを同期したい場合に役立ちます。

ビット 6:4 **TS[2:0]** : トリガ選択

このビットフィールドは、カウンタの同期に使用されるトリガ入力を選択します。

- 000 : 内部トリガ 0 (ITR0)
- 001 : 内部トリガ 1 (ITR1)
- 010 : 内部トリガ 2 (ITR2)
- 011 : 内部トリガ 3 (ITR3)
- 100 : TI1 エッジ検出回路 (TI1F_ED)
- 101 : フィルタタイマ入力 1 (TI1FP1)
- 110 : フィルタタイマ入力 2 (TI2FP2)

各タイマでの ITRx の詳細については、表 63 : TIMx 内部トリガ接続 (500 ページ) を参照してください。

注 : 設定変更時の誤ったエッジ検出を避けるために、これらのビットは、使用されていないとき (SMS=000 のときなど) にのみ変更しなければなりません。

ビット 3 予約済み、常に 0 として読み出されます。

ビット 2:0 **SMS** : スレーブモード選択

外部信号が選択されると、トリガ信号（TRGI）のアクティブエッジが外部入力で選択された極性にリンクされます（入力制御レジスタおよび制御レジスタの説明を参照してください）。

000 : スレーブモードは無効です。CEN = “1” の場合、プリスケアラは内部クロックによって直接クロック供給されます。

001 : エンコーダモード 1 - カウンタは、TI1FP2 のレベルに応じて、TI2FP1 のエッジでカウントアップ/ダウンします。

010 : エンコーダモード 2 - カウンタは、TI2FP1 のレベルに応じて、TI1FP2 のエッジでカウントアップ/ダウンします。

011 : エンコーダモード 3 - カウンタは、他の入力のレベルに応じて、TI1FP1 と TI2FP2 の両方のエッジでカウントアップ/ダウンします。

100 : リセットモード - 選択されたトリガ入力（TRGI）の立ち上がりエッジでカウンタを再初期化し、レジスタの更新を生成します。

101 : ゲートモード - カウンタクロックは、トリガ入力（TRGI）がハイのときに有効になります。トリガがローになると、カウンタは停止します（リセットはされません）。カウンタの開始と停止の両方が制御されます。

110 : トリガモード - カウンタは、トリガ TRGI の立ち上がりエッジで開始します（リセットはされません）。カウンタの開始のみが制御されます。

111 : 外部クロックモード 1 - 選択されたトリガ（TRGI）の立ち上がりエッジがカウンタのクロックとして供給されます。

注： トリガ入力として TI1F_ED が選択されている場合（TS=100）、ゲートモードを使用することはできません。TI1F_ED は TI1F の変化ごとに 1 パルスを出力しますが、ゲートモードはトリガ信号のレベルをチェックします。

表 63. TIMx 内部トリガ接続

スレーブ TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM15	TIM2	TIM3	TIM16_OC	TIM17_OC

19.5.4 TIM15 DMA／割り込み有効レジスタ（TIM15_DIER）

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	rw				rw	rw	rw	rw	rw	rw			rw	rw	rw

ビット 15 予約済み、常に 0 として読み出されます。

ビット 14 **TDE** : トリガ DMA リクエストイネーブル

0 : トリガ DMA リクエストは無効です。

1 : トリガ DMA リクエストは有効です。

ビット 13:11 予約済み、常に 0 として読み出されます。

ビット 10 **CC2DE** : キャプチャ／比較 2 DMA リクエストイネーブル

0 : CC2 DMA リクエストは無効です。

1 : CC2 DMA リクエストは有効です。



- ビット 9 **CC1DE** : キャプチャ／比較 1 DMA リクエストイネーブル
- 0 : CC1 DMA リクエストは無効です。
 - 1 : CC1 DMA リクエストは有効です。
- ビット 8 **UDE** : 更新 DMA リクエストイネーブル
- 0 : 更新 DMA リクエストは無効です。
 - 1 : 更新 DMA リクエストは有効です。
- ビット 7 **BIE** : ブレーク割り込みイネーブル
- 0 : ブレーク割り込みは無効です。
 - 1 : ブレーク割り込みは有効です。
- ビット 6 **TIE** : トリガ割り込みイネーブル
- 0 : トリガ割り込みは無効です。
 - 1 : トリガ割り込みは有効です。
- ビット 5 **COMIE** : COM 割り込みイネーブル
- 0 : COM 割り込みは無効です。
 - 1 : COM 割り込みは有効です。
- ビット 4:3 予約済み、常に 0 として読み出されます。
- ビット 2 **CC2IE** : キャプチャ／比較 2 割り込みイネーブル
- 0 : CC2 割り込みは無効です。
 - 1 : CC2 割り込みは有効です。
- ビット 1 **CC1IE** : キャプチャ／比較 1 割り込みイネーブル
- 0 : CC1 割り込みは無効です。
 - 1 : CC1 割り込みは有効です。
- ビット 0 **UIE** : 更新割り込みイネーブル
- 0 : 更新割り込みは無効です。
 - 1 : 更新割り込みは有効です。

19.5.5 TIM15 ステータスレジスタ (TIM15_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0				rc_w0	rc_w0	rc_w0

- ビット 15:11 予約済み、常に 0 として読み出されます。
- ビット 10 **CC2OF** : キャプチャ／比較 2 オーバーキャプチャフラグ
- CC1OF の説明を参照してください。
- ビット 9 **CC1OF** : キャプチャ／比較 1 オーバーキャプチャフラグ
- このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。
- 0 : オーバキャプチャは検出されていません。
 - 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。
- ビット 8 予約済み、常に 0 として読み出されます。



ビット 7 BIF : ブレーク割り込みフラグ

このフラグは、ブレーク入力アクティブになると、ハードウェアによってセットされます。ブレーク入力アクティブでない場合、ソフトウェアによってクリアできます。

0 : ブレークイベントは発生していません。

1 : ブレーク入力アクティブレベルが検出されました。

ビット 6 TIF : トリガ割り込みフラグ

このフラグは、トリガイメント時（スレーブモードコントローラがゲートモード以外のすべてのモードで有効なときに、TRGI 入力アクティブエッジが検出されたとき、またはゲートモードが選択されている場合は、両方のエッジが検出されたとき）にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : トリガイメントは発生していません。

1 : トリガ割り込みが保留中です。

ビット 5 COMIF : COM 割り込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます（キャプチャ/比較制御ビット - CCxE, CCxNE, OCxM - が更新されたとき）。ソフトウェアによってクリアされます。

0 : COM イベントは発生していません。

1 : COM 割り込みがペンディング中です。

ビット 5:3 予約済み、常に 0 として読み出されます。

ビット 2 CC2IF : キャプチャ/比較 2 割り込みフラグ

CC1IF の説明を参照してください。

ビット 1 CC1IF : キャプチャ/比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合 :

このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます（センターアラインモードでは、例外もあります。TIMx_CR1 レジスタの CMS ビットの説明を参照してください）。ソフトウェアによってクリアされます。

0 : 一致していません。

1 : カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ/ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。

CC1 チャンネルが入力として設定されている場合 :

このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0 : 入力キャプチャは発生していません。

1 : カウンタの値が TIMx_CCR1 レジスタにキャプチャされました（IC1 で、選択された極性に一致するエッジが検出されました）。

ビット 0 UIF : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

– 繰り返しカウンタ値に関するオーバーフロー（繰り返しカウンタ=0 の場合の更新）、および TIMx_CR1 レジスタの UDIS=0 の場合。

– TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

– TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、トリガイメントによって CNT が再初期化されたとき（[セクション 19.5.3 : TIM15 のスレーブモード制御レジスタ \(TIM15_SMCR\)](#) を参照）。

19.5.6 TIM15 のイベント生成レジスタ (TIM15_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			w	w	w

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7 **BG** : ブレーク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : ブレークイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 **TG** : トリガ生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : TIMx_SR レジスタの TIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。

注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4:3 予約済み、常に 0 として読み出されます。

ビット 2 **CC2G** : キャプチャ / 比較 2 イベント生成

CC1G の説明を参照してください。

ビット 1 **CC1G** : キャプチャ / 比較 1 イベント生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0 : 影響なし。

1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。

CC1 チャンネルが出力として設定されている場合 :

CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。

CC1 チャンネルが入力として設定されている場合 :

カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx_ARR) をとります。

19.5.7 TIM15 のキャプチャ／比較モードレジスタ 1（TIM15_CCMR1）

アドレスオフセット：0x18

リセット値：0x0000

チャンネルは、入力（キャプチャモード）または出力（比較モード）で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
				IC2F[3:0]								IC2PSC[1:0]	IC1F[3:0]		
	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w	r/w

出力比較モード：

ビット 15 予約済み、常に 0 として読み出されます。

ビット 14:12 **OC2M[2:0]**：出力比較 2 モード

ビット 11 **OC2PE**：出力比較 2 プリロードイネーブル

ビット 10 **OC2FE**：出力比較 2 高速イネーブル

ビット 9:8 **CC2S[1:0]**：キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

- 00：CC2 チャンネルは出力として設定されます。
- 01：CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。
- 10：CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。
- 11：CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット（TIMx_SMCR レジスタ）で内部トリガ入力を選択されている場合のみ機能します。

注： CC2S ビットは、チャンネルがオフ（TIMx_CCER レジスタの CC2E=0）のときにのみ書き込み可能です。

ビット 7 予約済み、常に 0 として読み出されます。

ビット 6:4 OC1M : 出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000 : 停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。

001 : 一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010 : 一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011 : 反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100 : 強制非アクティブレベル - OC1REF は強制的にローになります。

101 : 強制アクティブレベル - OC1REF は強制的にハイになります。

110 : PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111 : PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注 : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モード 1 または 2 では、比較結果が変化するとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OCFE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 **CC1S** : キャプチャ／比較 1 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

入力キャプチャモード

ビット 15:12 **IC2F** : 入力キャプチャ 2 フィルタ

ビット 11:10 **IC2PSC[1:0]** : 入力キャプチャ 2 プリスケアラ

ビット 9:8 **CC2S** : キャプチャ／比較 2 選択

このビットフィールドは、チャンネルの方向（入力／出力）と、使用される入力を定義します。

00 : CC2 チャンネルは出力として設定されます。

01 : CC2 チャンネルは入力として設定され、IC2 は TI2 に配置されます。

10 : CC2 チャンネルは入力として設定され、IC2 は TI1 に配置されます。

11 : CC2 チャンネルは入力として設定され、IC2 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC2S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC2E=0) のときにのみ書き込み可能です。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個の連続イベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$, N = 2

0010 : $f_{SAMPLING} = f_{CK\_INT}$, N = 4

0011 : $f_{SAMPLING} = f_{CK\_INT}$, N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$, N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$, N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$, N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$, N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$, N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$, N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$, N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$, N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$, N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$, N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$, N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$, N = 8

注 : **ICx F[3:0] = 1, 2, または 3 のときに、式の f_{DTS} が CK_INT に置き換えられるよう注意する必要があります。**

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。
プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

- 00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。
- 01 : キャプチャは、2 イベントごとに行われます。
- 10 : キャプチャは、4 イベントごとに行われます。
- 11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向 (入力 / 出力) と、使用される入力を定義します。

- 00 : CC1 チャンネルは出力として設定されます。
- 01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。
- 10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。
- 11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

19.5.8 TIM15 のキャプチャ / 比較有効レジスタ (TIM15_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rW		rW	rW	rW	rW	rW	rW

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7 **CC2NP** : キャプチャ / 比較 2 相補出力極性

CC1NP の説明を参照してください。

ビット 6 予約済み、常に 0 として読み出されます。

ビット 5 **CC2P** : キャプチャ / 比較 2 出力極性

CC1P の説明を参照してください。

ビット 4 **CC2E** : キャプチャ / 比較 2 出力イネーブル

CC1E の説明を参照してください。

ビット 3 **CC1NP** : キャプチャ / 比較 1 相補出力極性

0 : OC1N はアクティブハイです。

1 : OC1N はアクティブローです。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。

ビット 2 **CC1NE** : キャプチャ / 比較 1 相補出力イネーブル

0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。



ビット 1 **CC1P** : キャプチャ / 比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

- 0 : OC1 はアクティブハイです。
- 1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、トリガまたはキャプチャ動作に対する TI1FP1 と TI2FP1 の極性を選択します。

00 : 非反転/立ち上がりエッジ。この回路は TIxFP1 の立ち上がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。

01 : 反転/立ち下がりエッジ。この回路は TIxFP1 の立ち下がりエッジに反応し (キャプチャモード、リセットモードでのトリガ、またはトリガモード)、TIxFP1 は反転されます (ゲートモードでのトリガ)。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ。回路は TIxFP1 の立ち上がりエッジと立ち下がりエッジの両方に反応し (キャプチャモード、リセットモードでのトリガ、またはトリガモード)、TIxFP1 は反転されません (ゲートモードでのトリガ)。

注 : このビットは、**LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。**

ビット 0 **CC1E** : キャプチャ / 比較 1 出力イネーブル

CC1 チャンネルが出力として設定されている場合 :

- 0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。
- 1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

- 0 : キャプチャは無効です。
- 1 : キャプチャは有効です。

表 64. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能

制御ビット					出力状態 ⁽¹⁾	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	0	0	0	出力無効 (タイマによって駆動されない) OCx=0、OCx_EN=0	出力無効 (タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	0	1	出力無効 (タイマによって駆動されない) OCx=0、OCx_EN=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP、OCxN_EN = 1
		0	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	出力無効 (タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム OCxN_EN=1
		1	0	0	出力無効 (タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効 (タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
		1	0	1	オフ状態 (インアクティブ状態で出力有効) OCx=CCxP、OCx_EN=1	OCxREF + 極性 OCxN=OCxREF xor CCxNP、 OCxN_EN=1
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	オフ状態 (インアクティブ状態で出力有効) OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補 (OCREF ではなく) + 極性 + デッドタイム OCxN_EN=1
0	0	X	0	0	出力無効 (タイマによって駆動されない) 非同期: OCx=CCxP、OCx_EN=0、OCxN=CCxNP、OCxN_EN=0 クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	オフ状態 (インアクティブ状態で出力有効) 非同期: OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
	1		0	1		
	1		1	0		
	1		1	1		

1. チャンネルの両方の出力が使用されないとき (CCxE=CCxNE=0)、OISx、OISxN、CCxP、および CCxNP はクリアされたままでなければなりません。

注: 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OCx および OCxN チャンネルの状態と、GPIO および AFIO レジスタに依存します。

19.5.9 TIM15 のカウンタ (TIM15_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 CNT[15:0] : カウンタ値

19.5.10 TIM15 のプリスケアラ (TIM15_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 PSC[15:0] : プリスケアラ値

カウンタクロック周波数 (CK_CNT) は $f_{CK\_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

19.5.11 TIM15 の自動再ロードレジスタ (TIM15_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 ARR[15:0] : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 18.3.1 : タイムベースユニット \(445 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

19.5.12 TIM15 繰り返しカウンタレジスタ (TIM15_RCR)

アドレスオフセット : 0x30

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7:0 **REP[7:0]** : 繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウントをリスタートします。繰り返し更新イベント U_RC のみ、REP_CNT に REP 値がロードされるので、TIMx_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) はエッジアラインモードで PWM 周期の数を意味します。

19.5.13 TIM15 のキャプチャ／比較レジスタ 1 (TIM15_CCR1)

アドレスオフセット : 0x34

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CCR1[15:0]** : キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合 :

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

CC1 チャンネルが入力として設定されている場合 :

CCR1 は、最後の入力キャプチャ 1 イベント (IC1) によって転送されたカウンタ値です。

19.5.14 TIM15 のキャプチャ／比較レジスタ 2 (TIM15_CCR2)

アドレスオフセット : 0x38

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CCR2[15:0] : キャプチャ／比較 2 値

CC2 チャンネルが出力として設定されている場合 :

CCR2 は、実際のキャプチャ／比較 2 レジスタにロードされる値 (プリロード値) です。

TIMx_CCMR2 レジスタの OC2PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 2 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較され、OC2 出力に送信される値を含みます。

CC2 チャンネルが入力として設定されている場合 :

CCR2 は、最後の入力キャプチャ 2 イベント (IC2) によって転送されたカウンタ値です。

19.5.15 TIM15 ブレークおよびデッドタイムレジスタ (TIM15_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

注 : ビット AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] は、LOCK 設定に応じて書き込みがロックされるので、TIMx_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 15 MOE : メイン出力イネーブル

このビットは、ブレーク入力 that アクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャネルに対してのみ有効です。

0 : OC および OCN 出力は無効か、強制的にアイドル状態になります。

1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 19.5.8 : TIM15 のキャプチャ／比較有効レジスタ (TIM15_CCER) (507 ページ))。

ビット 14 AOE : 自動出力イネーブル

0 : MOE はソフトウェアによってのみセットできます。

1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力 that アクティブでない場合)。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 BKP : ブレーク極性

- 0 : ブレーク入力 BRK はアクティブローです。
- 1 : ブレーク入力 BRK はアクティブハイです。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 BKE : ブレークイネーブル

- 0 : ブレーク入力 (BRK および CCS クロック障害イベント) は無効です。
- 1 : ブレーク入力 (BRK および CCS クロック障害イベント) は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 19.5.8 : TIM15 のキャプチャ/比較有効レジスタ (TIM15_CCER) (507 ページ))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。
- 1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります。この場合、OC/OCN イネーブル出力信号 = 1 です。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください (セクション 19.5.8 : TIM15 のキャプチャ/比較有効レジスタ (TIM15_CCER) (507 ページ))。

- 0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。
- 1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、まず強制的にアイドルレベルになります (OC/OCN イネーブル出力信号 = 1)。

注 : このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 9:8 LOCK[1:0] : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

- 00 : LOCK オフ - どのビットも書き込み保護されません。
- 01 : LOCK レベル 1 = TIMx_BDTR レジスタの DTG ビット、TIMx_CR2 レジスタの OISx および OISxN ビット、および TIMx_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。
- 10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込めなくなります。
- 11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCMR レジスタの OCxM および OCxPE ビット) が書き込めなくなります。

注 : LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

ビット 7:0 **DTG[7:0]** : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。
DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}、ここで t_{dtg}=t_{DTS}
DTG[7:5]=10x => DT=(64+DTG[5:0])x t_{dtg}、ここで T_{dtg}=2x t_{DTS}
DTG[7:5]=110 => DT=(32+DTG[4:0])x t_{dtg}、ここで T_{dtg}=8x t_{DTS}
DTG[7:5]=111 => DT=(32+DTG[4:0])x t_{dtg}、ここで T_{dtg}=16x t_{DTS}
例 : T_{DTS}=125ns (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。
0 から 15875 ns (125 ns 単位)
16 μs から 31750 ns (250 ns 単位)
32 μs から 63 μs (1 μs 単位)
64 μs から 126 μs (2 μs 単位)

注 : このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

19.5.16 TIM15 DMA 制御レジスタ (TIM15_DCR)

アドレスオフセット : 0x48

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済み、常に 0 として読み出されます。

ビット 12:8 **DBL[4:0]** : DMA パースト長

この 5 ビットのベクタは、DMA 転送長 (タイマは、TIMx_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにパースト転送を認識します) を指定します。
00000 : 1 回転送
00001 : 2 回転送、
00010 : 3 回転送、
...
10001 : 18 回転送。

ビット 7:5 予約済み、常に 0 として読み出されます。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します (TIMx_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき)。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例 :

00000 : TIMx_CR1、
00001 : TIMx_CR2、
00010 : TIMx_SMCR
...



19.5.17 完全転送の TIM15 DMA アドレス (TIM15_DMAR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **DMAB[15:0]** : DMA パーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込み動作は、次のアドレスにあるレジスタへのアクセスとなります :

$$(\text{TIMx\_CR1 アドレス}) + (\text{DBA} + \text{DMA インデックス}) \times 4$$

ここで、TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスはDMA 転送によって自動的に制御され、範囲は 0 から DBL です (DBL はTIMx_DCR 内で設定)。

19.5.18 TIM15 レジスタマップ

TIM15 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 65. TIM15 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	TIM15_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.		OPM	URS	UDIS	CEN
	リセット値																							0	0	0				0	0	0	0	
0x04	TIM15_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	Res.	MMS[2:0]			CCDS	CCUS	Res.	CCPC	
	リセット値																						0	0	0		0	0	0	0	0		0	
0x08	TIM15_SMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]			
	リセット値																						0	0	0	0				0	0	0	0	
0x0C	TIM15_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE	
	リセット値																		0			0	0	0	0	0	0			0	0	0	0	
0x10	TIM15_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF	
	リセット値																					0	0	0	0	0	0			0	0	0	0	
0x14	TIM15_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG	
	リセット値																								0	0	0			0	0	0	0	
0x18	TIM15_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]	Res.	OC1M [2:0]		OC1PE	OC1FE	CC1S [1:0]				
	リセット値																			0	0	0	0	0		0	0	0	0	0	0	0		
	TIM15_CCMR1 入力キャプチャ モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]			IC2 PSC [1:0]	CC2S [1:0]	IC1F[3:0]			IC1 PSC [1:0]	CC1S [1:0]						
	リセット値																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x20	TIM15_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E	
	リセット値																									0		0	0	0	0	0	0	
0x24	TIM15_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x28	TIM15_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	TIM15_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x30	TIM15_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]								
	リセット値																									0	0	0	0	0	0	0	0	
0x34	TIM15_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x38	TIM15_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x44	TIM15_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]									
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	



表 65. TIM15 レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	TIM15_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
	リセット値																				0	0	0	0	0				0	0	0	0	0
0x4C	TIM15_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

19.6 TIM16 および TIM17 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

19.6.1 TIM16 および TIM17 制御レジスタ 1 (TIM16_CR1 および TIM17_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
						rw	rw	rw				rw	rw	rw	rw

ビット 15:10 予約済み、常に 0 として読み出されます。

ビット 9:8 **CKD[1:0]** : クロック分周

このビットフィールドは、タイマクロック (CK_INT) 周波数と、デッドタイムジェネレータとデジタルフィルタ (TlX) によって使用されるデッドタイムおよびサンプリングクロック (t_{DTS}) との間の分周比を示します。

00 : $t_{DTS}=t_{CK\_INT}$

01 : $t_{DTS}=2*t_{CK\_INT}$

10 : $t_{DTS}=4*t_{CK\_INT}$

11 : 予約済み - この値をプログラミングしないでください。

ビット 7 **ARPE** : 自動再ロードプリロードイネーブル

0 : TIMx_ARR レジスタはバッファされません。

1 : TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済み、常に 0 として読み出されます。

ビット 3 **OPM** : ワンパルスモード

0 : カウンタは更新イベントで停止しません。

1 : カウンタは次の更新イベントでカウントを停止します (CEN ビットをクリア)。

ビット 2 **URS** : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。
0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します（有効な場合）。これらのイベントは、次のとおりです。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー／アンダーフローのみが更新割り込みまたは DMA リクエストを生成します（有効な場合）。

ビット 1 **UDIS** : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント（UEV）は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC、CCR_x）は値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケールは再初期化されます。

ビット 0 **CEN** : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

19.6.2 TIM16 および TIM17 制御レジスタ 2 (TIM16_CR2 および TIM17_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						r/w	r/w					r/w	r/w		r/w

ビット 15:10 予約済み、常に 0 として読み出されます。

ビット 9 **OIS1N** : 出力アイドル状態 1 (OC1N 出力)

0 : MOE=0 のとき、デッドタイム後に OC1N=0

1 : MOE=1 のとき、デッドタイム後に OC1N=0

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIM_x_BKR レジスタの LOCK ビット)。

ビット 8 **OIS1** : 出力アイドル状態 1 (OC1 出力)

0 : MOE=0 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

1 : MOE=1 のとき、OC1=0 (OC1N が実装されている場合、デッドタイム後に)

注 : このビットは、LOCK レベル 1、2、または 3 がプログラムされているときには変更できません (TIM_x_BKR レジスタの LOCK ビット)。

ビット 7:4 予約済み、常に 0 として読み出されます。



- ビット 3 **CCDS** : キャプチャ/比較 DMA 選択
- 0 : CCx DMA リクエストは、CCx イベントが発生すると送信されます。
1 : CCx DMA リクエストは、更新イベントが発生すると送信されます。
- ビット 2 **CCUS** : キャプチャ/比較制御更新選択
- 0 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによってのみ更新されます。
1 : キャプチャ/比較制御ビットがプリロードされる時には (CCPC=1)、COMG ビットをセットすることによって、または TRGI の立ち上がりエッジで更新されます。
- 注：** このビットは、相補出力を持つチャンネルでのみ機能します。
- ビット 1 予約済み、常に 0 として読み出されます。
- ビット 0 **CCPC** : キャプチャ/比較プリロード制御
- 0 : CCxE、CCxNE、および OCxM ビットはプリロードされません。
1 : CCxE、CCxNE、および OCxM ビットがプリロードされます。書き込み後は、COM ビットがセットされた場合にのみ更新されます。
- 注：** このビットは、相補出力を持つチャンネルでのみ機能します。

19.6.3 TIM16 および TIM17 DMA/割り込み有効レジスタ (TIM16_DIER および TIM17_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
						rw	rw	rw		rw				rw	rw

- ビット 15 予約済み、常に 0 として読み出されます。
- ビット 14 予約済み、常に 0 として読み出されます。
- ビット 13:10 予約済み、常に 0 として読み出されます。
- ビット 9 **CC1DE** : キャプチャ/比較 1 DMA リクエストイネーブル
- 0 : CC1 DMA リクエストは無効です。
1 : CC1 DMA リクエストは有効です。
- ビット 8 **UDE** : 更新 DMA リクエストイネーブル
- 0 : 更新 DMA リクエストは無効です。
1 : 更新 DMA リクエストは有効です。
- ビット 7 **BIE** : ブレーク割り込みイネーブル
- 0 : ブレーク割り込みは無効です。
1 : ブレーク割り込みは有効です。
- ビット 6 予約済み、常に 0 として読み出されます。
- ビット 5 **COMIE** : COM 割り込みイネーブル
- 0 : COM 割り込みは無効です。
1 : COM 割り込みは有効です。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 4:2 予約済み、常に 0 として読み出されます。

ビット 1 **CC1IE** : キャプチャ/比較 1 割り込みイネーブル

- 0 : CC1 割り込みは無効です。
- 1 : CC1 割り込みは有効です。

ビット 0 **UIE** : 更新割り込みイネーブル

- 0 : 更新割り込みは無効です。
- 1 : 更新割り込みは有効です。

19.6.4 TIM16 および TIM17 ステータスレジスタ (TIM16_SR および TIM17_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

ビット 15:10 予約済み、常に 0 として読み出されます。

ビット 9 **CC1OF** : キャプチャ/比較 1 オーバークャプチャフラグ

このフラグは、対応するチャンネルが入力キャプチャモードに設定されているときのみ、ハードウェアによってセットされます。“0”を書き込むことによってソフトウェアによってクリアされます。

- 0 : オーバークャプチャは検出されていません。
- 1 : CC1IF フラグがすでにセットされているときに、カウンタの値が TIMx_CCR1 レジスタにキャプチャされました。

ビット 8 予約済み、常に 0 として読み出されます。

ビット 7 **BIF** : ブレーク割り込みフラグ

このフラグは、ブレーク入力がアクティブになると、ハードウェアによってセットされます。ブレーク入力にアクティブでない場合、ソフトウェアによってクリアできます。

- 0 : ブレークイベントは発生していません。
- 1 : ブレーク入力にアクティブレベルが検出されました。

ビット 6 予約済み、常に 0 として読み出されます。

ビット 5 **COMIF** : COM 割り込みフラグ

このフラグは、COM イベント時にハードウェアによってセットされます (キャプチャ/比較制御ビット - CCxE、CCxNE、OCxM - が更新されたとき)。ソフトウェアによってクリアされます。

- 0 : COM イベントは発生していません。
- 1 : COM 割り込みがペンディング中です。

ビット 4:2 予約済み、常に 0 として読み出されます。

ビット 1 **CC1IF**：キャプチャ／比較 1 割り込みフラグ

CC1 チャンネルが出力として設定されている場合：
このフラグは、カウンタが比較値と一致したときにハードウェアによってセットされます（センターアラインモードでは、例外もあります。TIMx_CR1 レジスタの CMS ビットの説明を参照してください）。ソフトウェアによってクリアされます。

0：一致していません。
1：カウンタ TIMx_CNT の内容が TIMx_CCR1 レジスタの内容と一致しました。TIMx_CCR1 の内容が TIMx_ARR の内容より大きいときには、カウンタオーバーフロー時（アップカウントおよびアップ／ダウンカウントモードの場合）、またはアンダーフロー時（ダウンカウントモードの場合）に CC1IF ビットはハイになります。

CC1 チャンネルが入力として設定されている場合：
このビットは、キャプチャ時にハードウェアによってセットされます。ソフトウェアによって、または TIMx_CCR1 レジスタを読み出すことによってクリアされます。

0：入力キャプチャは発生していません。
1：カウンタの値が TIMx_CCR1 レジスタにキャプチャされました（IC1 で、選択された極性に一致するエッジが検出されました）。

ビット 0 **UIF**：更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0：更新は発生していません。
1：更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

- 繰り返しカウンタ値に関するオーバーフロー（繰り返しカウンタ=0 の場合の更新）、および TIMx_CR1 レジスタの UDIS=0 の場合。
- TIMx_CR1 レジスタの URS=0 かつ UDIS=0 であり、TIMx_EGR レジスタの UG ビットを使用して、CNT がソフトウェアによって再初期化されたとき。

19.6.5 TIM16 および TIM17 のイベント生成レジスタ（TIM16_EGR および TIM17_EGR）

アドレスオフセット：0x14

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								w		w				w	w

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7 **BG**：ブレイク生成

このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。

0：影響はありません。
1：ブレイクイベントが生成されます。MOE ビットがクリアされ、BIF フラグがセットされます。有効な場合は、関連する割り込みまたは DMA 転送が発生します。

ビット 6 予約済み、常に 0 として読み出されます。

ビット 5 **COMG** : キャプチャ / 比較制御更新生成
このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。
0 : 影響なし。
1 : CCPC ビットがセットされているときには、CCxE、CCxNE、および OCxM ビットを更新できます。
注 : このビットは、相補出力を持つチャンネルでのみ機能します。

ビット 4:2 予約済み、常に 0 として読み出されます。

ビット 1 **CC1G** : キャプチャ / 比較 1 イベント生成
このビットは、イベントを生成するためにソフトウェアによってセットされ、ハードウェアによって自動的にクリアされます。
0 : 影響はありません。
1 : チャンネル 1 でキャプチャ / 比較イベントが生成されます。
CC1 チャンネルが出力として設定されている場合 :
CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。
CC1 チャンネルが入力として設定されている場合 :
カウンタの現在値が TIMx_CCR1 レジスタにキャプチャされます。CC1IF フラグがセットされ、対応する割り込みまたは DMA リクエストが送信されます (有効な場合)。CC1IF フラグがすでにハイの場合、CC1OF フラグがセットされます。

ビット 0 **UG** : 更新生成
このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。
0 : 影響はありません。
1 : カウンタを再初期化し、レジスタの更新を生成します。プリスケアラカウンタもクリアされます (プリスケアラ比は変化しません)。センターアラインモードが選択されている場合、または、DIR=0 (カウントアップ) の場合、カウンタはクリアされます。そうでない場合、DIR=1 (カウントダウン) であれば、自動再ロード値 (TIMx_ARR) をとります。

19.6.6 TIM16 および TIM17 のキャプチャ / 比較モードレジスタ 1 (TIM16_CCMR1 および TIM17_CCMR1)

アドレスオフセット : 0x18

リセット値 : 0x0000

チャンネルは、入力 (キャプチャモード) または出力 (比較モード) で使用できます。チャンネルの方向は対応する CCxS ビットで定まります。このレジスタの他のビットはすべて、入力モードと出力モードで異なる機能を持ちます。特定のビットについて、OCxx は、チャンネルが出力設定のときの機能を示し、ICxx は、チャンネルが入力設定のときの機能を記述します。したがって、同じビットが入力ステージと出力ステージで異なる意味を持つことに注意する必要があります。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]			
								rW	rW	rW	rW	rW	rW	rW	rW

出力比較モード：

ビット 15:7 予約済み、常に 0 として読み出されます。

ビット 6:4 **OC1M**：出力比較 1 モード

これらのビットは、OC1 および OC1N が導き出される出力基準信号 OC1REF の動作を定義します。OC1REF はアクティブハイですが、OC1 および OC1N のアクティブレベルは CC1P および CC1NP ビットに依存します。

000：停止 - 出力比較レジスタ TIMx_CCR1 とカウンタ TIMx_CNT との間の比較結果は出力に影響しません。

001：一致時にチャンネル 1 をアクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にハイになります。

010：一致時にチャンネル 1 を非アクティブレベルに設定します。OC1REF 信号は、カウンタ TIMx_CNT がキャプチャ/比較レジスタ 1 (TIMx_CCR1) と一致したときに、強制的にローになります。

011：反転 - TIMx_CNT = TIMx_CCR1 のとき、OC1REF は反転します。

100：強制非アクティブレベル - OC1REF は強制的にローになります。

101：強制アクティブレベル - OC1REF は強制的にハイになります。

110：PWM モード 1 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はインアクティブ (OC1REF="0") に、そうでない場合はアクティブ (OC1REF="1") になります。

111：PWM モード 2 - カウントアップ時、チャンネル 1 は、TIMx_CNT < TIMx_CCR1 の場合は非アクティブに、そうでない場合はアクティブになります。カウントダウン時、チャンネル 1 は、TIMx_CNT > TIMx_CCR1 の場合はアクティブに、そうでない場合は非アクティブになります。

注： 1: これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2: PWM モード 1 または 2 では、比較結果が変化したとき、または出力比較モードが停止モードから PWM モードに変更されたときにのみ、OCREF のレベルが変化します。

ビット 3 OC1PE : 出力比較 1 プリロードイネーブル

0 : TIMx_CCR1 のプリロードレジスタは無効です。TIMx_CCR1 は、いつでも書き込み可能であり、新しい値はただちに有効になります。

1 : TIMx_CCR1 のプリロードレジスタは有効です。読み書きはプリロードレジスタに対して行われます。TIMx_CCR1 プリロード値は、更新イベントのたびにアクティブレジスタにロードされます。

注 : 1 : これらのビットは、LOCK レベル 3 がプログラムされていて (TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力に設定) のときには、変更できません。

2 : PWM モードは、ワンパルスモード (TIMx_CR1 レジスタの OPM ビットがセットされている) のときのみ、プリロードレジスタを検証せずに使用できます。そうでない場合、動作は保証されません。

ビット 2 OC1FE : 出力比較 1 高速イネーブル

このビットは、CC 出力に対するトリガがイベントの効果を加速するために使用されます。

0 : CC1 の動作は、トリガがオンのときでも、通常、カウンタと CCR1 の値に依存します。トリガ入力エッジ発生から CC1 出力が有効になるまでの最小遅延は、5 クロックサイクルです。

1 : トリガ入力のアクティブエッジは、CC1 出力に対して、比較一致のように働きます。このため、OC は、比較結果には関係なく、比較レベルにセットされます。トリガ入力をサンプリングし、CC1 出力を有効にするまでの遅延は、3 クロックサイクルに短縮されます。OC1FE は、チャンネルが PWM1 または PWM2 モードに設定されている場合のみ機能します。

ビット 1:0 CC1S : キャプチャ/比較 1 選択

このビットフィールドは、チャンネルの方向 (入力/出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力を選択されている場合のみ機能します。

注 : CC1S ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときのみ書き込み可能です。

入力キャプチャモード

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7:4 **IC1F[3:0]** : 入力キャプチャ 1 フィルタ

このビットフィールドは、TI1 入力をサンプリングする周波数と、TI1 に適用されるデジタルフィルタの長さを定義します。デジタルフィルタはイベントカウンタでできていて、出力に有効な変化をもたらすには N 個のイベント発生が必要です。

0000 : フィルタなし、サンプリングは f_{DTS} で行われます。

0001 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 2

0010 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 4

0011 : $f_{SAMPLING} = f_{CK\_INT}$ 、N = 8

0100 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 6

0101 : $f_{SAMPLING} = f_{DTS}/2$ 、N = 8

0110 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 6

0111 : $f_{SAMPLING} = f_{DTS}/4$ 、N = 8

1000 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 6

1001 : $f_{SAMPLING} = f_{DTS}/8$ 、N = 8

1010 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 5

1011 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 6

1100 : $f_{SAMPLING} = f_{DTS}/16$ 、N = 8

1101 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 5

1110 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 6

1111 : $f_{SAMPLING} = f_{DTS}/32$ 、N = 8

注 : $ICx F[3:0] = 1, 2$, または 3 のときに、式の f_{DTS} が CK_INT に置き換えられるよう注意する必要があります。

ビット 3:2 **IC1PSC** : 入力キャプチャ 1 プリスケアラ

このビットフィールドは、CC1 入力 (IC1) に作用するプリスケアラの分周比を定義します。

プリスケアラは、CC1E = 0 (TIMx_CCER レジスタ) になるとリセットされます。

00 : プリスケアラなし。キャプチャは、キャプチャ入力のエッジが検出されるたびに行われます。

01 : キャプチャは、2 イベントごとに行われます。

10 : キャプチャは、4 イベントごとに行われます。

11 : キャプチャは、8 イベントごとに行われます。

ビット 1:0 **CC1S** : キャプチャ / 比較 1 選択

このビットフィールドは、チャンネルの方向 (入力 / 出力) と、使用される入力を定義します。

00 : CC1 チャンネルは出力として設定されます。

01 : CC1 チャンネルは入力として設定され、IC1 は TI1 に配置されます。

10 : CC1 チャンネルは入力として設定され、IC1 は TI2 に配置されます。

11 : CC1 チャンネルは入力として設定され、IC1 は TRC に配置されます。このモードは、TS ビット (TIMx_SMCR レジスタ) で内部トリガ入力が選択されている場合のみ機能します。

注 : **CC1S** ビットは、チャンネルがオフ (TIMx_CCER レジスタの CC1E=0) のときにのみ書き込み可能です。

19.6.7 TIM16 および TIM17 のキャプチャ／比較有効レジスタ (TIM16_CCER および TIM17_CCER)

アドレスオフセット : 0x20

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

ビット 15:4 予約済み、常に 0 として読み出されます。

ビット 3 **CC1NP** : キャプチャ / 比較 1 相補出力極性

0 : OC1N はアクティブハイです。

1 : OC1N はアクティブローです。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされ、(TIMx_BDTR レジスタの LOCK ビット)、かつ CC1S="00" (チャンネルは出力として設定) になった直後は書き込みできません。

ビット 2 **CC1NE** : キャプチャ / 比較 1 相補出力カインエーブル

0 : オフ - OC1N はアクティブではありません。OC1N のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1E ビットによって決まります。

1 : オン - OC1N 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1E ビットにより、対応する出力ピンに出力されます。

ビット 1 **CC1P** : キャプチャ / 比較 1 出力極性

CC1 チャンネルが出力として設定されている場合 :

0 : OC1 はアクティブハイです。

1 : OC1 はアクティブローです。

CC1 チャンネルが入力として設定されている場合 :

CC1NP/CC1P ビットは、キャプチャ動作に対する TI1FP1 と TI2FP1 の極性を選択します。

00 : 非反転/立ち上がりエッジ。この回路は TIxFP1 の立ち上がりエッジに反応し、TIxFP1 は反転されません。

01 : 反転/立ち下がりエッジ。この回路は TIxFP1 の立ち下がりエッジに反応し、TIxFP1 は反転されます。

10 : 予約済み。この設定は使用しないでください。

11 : 非反転/両エッジ。この回路は TIxFP1 の立ち上がりおよび立ち下がりエッジの両方に反応し、TIxFP1 は反転されません。

注 : このビットは、LOCK レベル 2 または 3 がプログラムされた直後は書き込みできません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 0 **CC1E** : キャプチャ / 比較 1 出力カインエーブル

CC1 チャンネルが出力として設定されている場合 :

0 : オフ - OC1 はアクティブではありません。OC1 のレベルは、MOE、OSSI、OSSR、OIS1、OIS1N および CC1NE ビットによって決まります。

1 : オン - OC1 信号は、MOE、OSSI、OSSR、OIS1、OS1N、および CC1NE ビットにより、対応する出力ピンに出力されます。

CC1 チャンネルが入力として設定されている場合 :

このビットによって、カウンタ値のキャプチャ / 比較レジスタ 1 (TIMx_CCR1) へのキャプチャが実際に行われるかどうかが決まります。

0 : キャプチャは無効です。

1 : キャプチャは有効です。



表 66. ブレーク機能を持つ相補 OCx および OCxN チャンネルの出力制御ビット機能

制御ビット					出力状態 ⁽¹⁾	
MOE ビット	OSSI ビット	OSSR ビット	CCxE ビット	CCxNE ビット	OCx 出力状態	OCxN 出力状態
1	X	0	0	0	出力無効(タイマによって駆動されない) OCx=0、OCx_EN=0	出力無効(タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	0	1	出力無効(タイマによって駆動されない) OCx=0、OCx_EN=0	OCxREF + 極性 OCxN = OCxREF xor CCxNP、OCxN_EN = 1
		0	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	出力無効(タイマによって駆動されない) OCxN=0、OCxN_EN=0
		0	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補 (OCREF で はなく) + 極性 + デッドタイム OCxN_EN=1
		1	0	0	出力無効(タイマによって駆動されない) OCx=CCxP、OCx_EN=0	出力無効(タイマによって駆動されない) OCxN=CCxNP、OCxN_EN=0
		1	0	1	オフ状態(インアクティブ状態で出力有効) OCx=CCxP、OCx_EN=1	OCxREF + 極性 OCxN=OCxREF xor CCxNP、 OCxN_EN=1
		1	1	0	OCxREF + 極性 OCx=OCxREF xor CCxP、 OCx_EN=1	オフ状態 (インアクティブ状態で出力有効) OCxN=CCxNP、OCxN_EN=1
		1	1	1	OCREF + 極性 + デッドタイム OCx_EN=1	OCREF に対する相補 (OCREF で はなく) + 極性 + デッドタイム OCxN_EN=1
0	0	X	0	0	出力無効 (タイマによって駆動されない) 非同期: OCx=CCxP、OCx_EN=0、OCxN=CCxNP、OCxN_EN=0 クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
	0		0	1		
	0		1	0		
	0		1	1		
	1		0	0	オフ状態 (インアクティブ状態で出力有効) 非同期: OCx=CCxP、OCx_EN=1、OCxN=CCxNP、OCxN_EN=1 クロックが存在する場合: デッドタイム後、OCx=OISx および OCxN=OISxN。ただし、OISx と OISxN は、アクティブ状態における OCX と OCxN の両方に対応しないことを前提とします。	
	1		0	1		
	1		1	0		
	1		1	1		

1. チャンネルの両方の出力が使用されないとき (CCxE=CCxNE=0)、OISx、OISxN、CCxP、および CCxNP はクリアされたままではなりません。

注: 相補 OCx および OCxN チャンネルに接続されている外部入出力ピンの状態は、OCx および OCxN チャンネルの状態と、GPIO および AFIO レジスタに依存します。

19.6.8 TIM16 および TIM17 のカウンタ (TIM16_CNT および TIM17_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CNT[15:0] : カウンタ値

19.6.9 TIM16 および TIM17 のプリスケアラ (TIM16_PSC および TIM17_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 PSC[15:0] : プリスケアラ値

カウンタクロック周波数 (CK_CNT) は $f_{CK\_PSC} / (PSC[15:0] + 1)$ に等しいです。

PSC は、更新イベントごとにアクティブプリスケアラレジスタにロードされる値を含みます (更新イベントには、TIMx_EGR レジスタの UG ビットを通じて、またはリセットモードに設定されているトリガコントローラを通じて、カウンタがクリアされる場合も含まれます)。

19.6.10 TIM16 および TIM17 自動再ロードレジスタ (TIM16_ARR および TIM17_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 ARR[15:0] : 自動再ロード値

ARR は、実際の自動再ロードレジスタにロードされる値です。

ARR の更新と動作の詳細については、[セクション 18.3.1 : タイムベースユニット \(445 ページ\)](#) を参照してください。

自動再ロード値が null のときには、カウンタはブロックされます。

19.6.11 TIM16 および TIM17 繰り返しカウンタレジスタ（TIM16_RCR および TIM17_RCR）

アドレスオフセット：0x30

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7:0 REP[7:0]：繰り返しカウンタ値

これらのビットによって、プリロードレジスタが有効なときの比較レジスタの更新レート（プリロードレジスタからアクティブレジスタへの定期的な転送）と、割り込みが有効な場合の更新割り込み生成の頻度をセットアップできます。

REP_CNT に関連するダウンカウンタがゼロに達するたびに、更新イベントが生成され、REP 値からカウンタをリスタートします。繰り返し更新イベント U_RC でのみ、REP_CNT に REP 値がロードされるので、TIMx_RCR レジスタへの書き込みは、次の繰り返し更新イベントまで有効になりません。

PWM モードでは、(REP+1) はエッジアラインモードで PWM 周期の数を意味します。

19.6.12 TIM16 および TIM17 のキャプチャ／比較レジスタ 1（TIM16_CCR1 および TIM17_CCR1）

アドレスオフセット：0x34

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 CCR1[15:0]：キャプチャ／比較 1 値

CC1 チャンネルが出力として設定されている場合：

CCR1 は、実際のキャプチャ／比較 1 レジスタにロードされる値（プリロード値）です。

TIMx_CCMR1 レジスタの OC1PE ビットでプリロード機能が選択されていない場合、この値は不変にロードされます。そうでない場合、プリロード値は、更新イベントが発生したときに、アクティブキャプチャ／比較 1 レジスタにコピーされます。

アクティブキャプチャ／比較レジスタは、カウンタ TIMx_CNT と比較されて、OC1 出力に送信される値を含みます。

CC1 チャンネルが入力として設定されている場合：

CCR1 は、最後の入力キャプチャ 1 イベント（IC1）によって転送されたカウンタ値です。

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

19.6.13 TIM16 および TIM17 ブレークおよびデッドタイムレジスタ (TIM16_BDTR および TIM17_BDTR)

アドレスオフセット : 0x44

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

注 : ビット AOE、BKP、BKE、OSSI、OSSR、および DTG[7:0] は、LOCK 設定に応じて書き込みがロックされるので、TIMx_BDTR レジスタへの最初のアクセス時に、これらすべてを設定しなければならないことがあります。

ビット 15 **MOE** : メイン出力イネーブル

このビットは、ブレーク入力アクティブとなると、ハードウェアによって非同期にクリアされます。ソフトウェアによって、または、AOE ビットに応じて自動的にセットされます。出力として設定されたチャネルに対してのみ有効です。

- 0 : OC および OCN 出力は無効か、強制的にアイドル状態になります。
 - 1 : OC および OCN 出力は、それぞれのイネーブルビット (TIMx_CCER レジスタの CCxE、CCxNE ビット) がセットされている場合は有効です。
- 詳細については、OC/OCN イネーブルの説明を参照してください (セクション 19.5.8 : TIM15 のキャプチャ/比較有効レジスタ (TIM15_CCER) (507 ページ))。

ビット 14 **AOE** : 自動出力イネーブル

- 0 : MOE はソフトウェアによってのみセットできます。
- 1 : MOE は、ソフトウェアによって、または次の更新イベント時に自動的にセットできます (ブレーク入力アクティブでない場合)。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

ビット 13 **BKP** : ブレーク極性

- 0 : ブレーク入力 BRK はアクティブラーです。
- 1 : ブレーク入力 BRK はアクティブハイです。

注 : このビットは、LOCK レベル 1 がプログラムされている場合 (TIMx_BDTR レジスタの LOCK ビット)、変更できません。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 12 **BKE** : ブレークイネーブル

- 0 : ブレーク入力 (BRK および CCS クロック障害イベント) は無効です。
- 1 : ブレーク入力 (BRK および CCS クロック障害イベント) は有効です。

注 : このビットは、LOCK レベル 1 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

注 : このビットへの書き込み操作では、書き込みが有効になるまでに 1 APB クロックサイクルの遅延が生じます。

ビット 11 OSSR : RUN モードのオフ状態の選択

このビットは、MOE=1 のとき、相補出力を持ち、出力として設定されているチャンネルで使用されます。OSSR は、相補出力がタイマに実装されていない場合には、実装されません。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 19.5.8 : TIM15 のキャプチャ/比較有効レジスタ \(TIM15_CCER\) \(507 ページ\)](#))。

0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、インアクティブレベルで有効になります。この場合、OC/OCN イネーブル出力信号 = 1 です。

注： このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 10 OSSI : アイドルモードのオフ状態の選択

このビットは、MOE=0 のとき、出力として設定されているチャンネルで使用されます。

詳細については、OC/OCN イネーブルの説明を参照してください ([セクション 19.5.8 : TIM15 のキャプチャ/比較有効レジスタ \(TIM15_CCER\) \(507 ページ\)](#))。

0 : インアクティブのとき、OC/OCN 出力は無効です (OC/OCN イネーブル出力信号 = 0)。

1 : インアクティブのとき、CCxE=1 または CCxNE=1 になると、OC/OCN 出力は、まず強制的にアイドルレベルになります (OC/OCN イネーブル出力信号 = 1)。

注： このビットは、LOCK レベル 2 がプログラムされているときには変更できません (TIMx_BDTR レジスタの LOCK ビット)。

ビット 9:8 LOCK[1:0] : ロック設定

これらのビットは、ソフトウェアエラーに対する書き込み保護を提供します。

00 : LOCK オフ - どのビットも書き込み保護されません。

01 : LOCK レベル 1 = TIMx_BDTR レジスタの DTG ビット、TIMx_CR2 レジスタの OISx および OISxN ビット、および TIMx_BDTR レジスタの BKE/BKP/AOE ビットは、書き込みができなくなります。

10 : LOCK レベル 2 - LOCK レベル 1 に加えて、CC 極性ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCER レジスタの CCxP/CCxNP ビット) と OSSR および OSSI ビットも書き込みできません。

11 : LOCK レベル 3 - LOCK レベル 2 に加えて、CC 制御ビット (関連するチャンネルが CCxS ビットを通じて出力に設定されている場合は、TIMx_CCMR レジスタの OCxM および OCxPE ビット) が書き込みできません。

注： LOCK ビットは、リセット後に一度だけ書き込みができます。いったん TIMx_BDTR レジスタに書き込みが行われると、その内容は次のリセットまで凍結されます。

ビット 7:0 DTG[7:0] : デッドタイムジェネレータのセットアップ

これらのビットでは、相補出力の間に挿入されるデッドタイムの長さを指定します。デッドタイムの時間 (DT) は、次の式で与えられます。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}, ここで t_{dtg}=t_{DTS}

DTG[7:5]=10x => DT=(64+DTG[5:0])x t_{dtg}, ここで T_{dtg}=2x t_{DTS}

DTG[7:5]=110 => DT=(32+DTG[4:0])x t_{dtg}, ここで T_{dtg}=8x t_{DTS}

DTG[7:5]=111 => DT=(32+DTG[4:0])x t_{dtg}, ここで T_{dtg}=16x t_{DTS}

例 : T_{DTS}=125ns (8MHz) の場合、可能なデッドタイムの値は、以下のとおりです。

0 から 15875 ns (125 ns 単位)

16 μs から 31750 ns (250 ns 単位)

32 μs から 63 μs (1 μs 単位)

64 μs から 126 μs (2 μs 単位)

注： このビットフィールドは、LOCK レベル 1、2、または 3 がプログラムされているとき、変更できません (TIMx_BDTR レジスタの LOCK ビット)。

19.6.14 TIM16 および TIM17 DMA 制御レジスタ（TIM16_DCR および TIM17_DCR）

アドレスオフセット：0x48

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

ビット 15:13 予約済み、常に 0 として読み出されます。

ビット 12:8 **DBL[4:0]** : DMA バースト長

この 5 ビットのベクタは、DMA 転送長（タイマは、TIMx_DMAR アドレスに対して読み出しまたは書き込みアクセスが行われるときにバースト転送を認識します）、つまり転送回数を指定します。転送は、ハーフワードまたはバイトです（以下の例を参照）。

00000 : 1 回転送
00001 : 2 回転送、
00010 : 3 回転送、
...
10001 : 18 回転送。

ビット 7:5 予約済み、常に 0 として読み出されます。

ビット 4:0 **DBA[4:0]** : DMA ベースアドレス

この 5 ビットのベクタは、DMA 転送のベースアドレスを指定します（TIMx_DMAR アドレスを通じて読み出し/書き込みアクセスが行われるとき）。DBA は、TIMx_CR1 レジスタのアドレスから始まるオフセットとして定義されます。

例：

00000 : TIMx_CR1、
00001 : TIMx_CR2、
00010 : TIMx_SMCR
...

例： 次の転送を考えます：DBL = 7 転送 かつ DBA = TIMx_CR1。この場合、転送は、TIMx_CR1 アドレスから始めて、7 つのレジスタに対して行われます。

19.6.15 完全転送の TIM16 および TIM17 DMA アドレス（TIM16_DMAR および TIM17_DMAR）

アドレスオフセット：0x4C

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:0 **DMAB[15:0]** : DMA パーストアクセスレジスタ

DMAR レジスタへの読み出しまたは書き込みアクセスは、次のアドレスにあるレジスタへのアクセスとなります。

" (TIMx_CR1 アドレス) + DBA + (DMA インデックス) "のとき :

TIMx_CR1 アドレスは制御レジスタ 1 のアドレスであり、DBA は TIMx_DCR レジスタで設定された DMA ベースアドレスであり、DMA インデックスは DMA 転送によって自動的に制御されたオフセットです (TIMx_DCR レジスタの転送 DBL の長さに応じて)。

DMA バースト機能の使用例

この例では、タイマ DMA バースト機能を使って CCRx レジスタ値を更新します (x = 2、3、4)。このとき、DMA は CCRx レジスタへハーフワードを転送します。

これは次のステップに従って行います。

1. 対応する DMA チャンネルを次のように設定します。
 - DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスとします。
 - DMA チャンネルメモリアドレスを、DMA によって CCRx レジスタに転送されるデータを格納する RAM 内のバッファアドレスとします。
 - 転送データ数 = 3 とします (下の注を参照)。
 - サーキュラモードは無効です。
2. DBA と DBL のビットフィールドを次のように設定することによって、DCR レジスタを設定します。
DBL = 3 転送、DBA = 0xE。
3. TIMx 更新 DMA リクエストを有効にします (DIER レジスタのUDE ビットをセット)。
4. TIMx を有効化
5. DMA チャンネルを有効化注 :

注 : この例は、各 CCRx レジスタが 1 回更新される場合です。たとえば、各 CCRx レジスタが 2 回更新される場合は、転送データ数は 6 になります。データ 1、データ 2、データ 3、データ 4、データ 5、データ 6 を格納する RAM のバッファを例にします。データは、CCRx レジスタに次のように転送されます。最初の更新 DMA リクエストでデータ 1 が CCR2 に転送され、データ 2 は CCR3 に、データ 3 は CCR4 にそれぞれ転送され、2 番目の更新 DMA リクエストでデータ 4 が CCR2 に、データ 5 が CCR3 に、データ 6 が CCR4 にそれぞれ転送されます。

19.6.16 TIM16 および TIM17 のレジスタマップ

TIM16 および TIM17 レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 67. TIM16 および TIM17 のレジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	TIM16_CR1 および TIM17_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	Res.	OPM	URS	UDIS	CEN					
	リセット値																							0	0	0				0	0	0	0						
0x04	TIM16_CR2 および TIM17_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC						
	リセット値																							0	0	Res.	Res.	Res.	Res.	0	0	Res.	0						
0x0C	TIM16_DIER および TIM17_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE						
	リセット値																		0					0	0	0	0	0	Res.	Res.	Res.	0	0						
0x10	TIM16_SR および TIM17_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF						
	リセット値																						0	Res.	0	0	0	0		Res.	Res.	0	0						
0x14	TIM16_EGR および TIM17_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG						
	リセット値																							Res.	0	0	0			Res.	Res.	0	0						
0x18	TIM16_CCMR1 およ び TIM17_CCMR1 出力比較モード	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]							
	リセット値																										0	0	0	0	0	0	0	0					
	TIM16_CCMR1 およ び TIM17_CCMR1 入力キャプチャモー ド	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]			IC1 PSC [1:0]		CC1S [1:0]									
	リセット値																										0	0	0	0	0	0	0	0					
0x20	TIM16_CCER およ び TIM17_CCER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E					
	リセット値																													0	0	0	0						
0x24	TIM16_CNT およ び TIM17_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x28	TIM16_PSC およ び TIM17_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x2C	TIM16_ARR およ び TIM17_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x30	TIM16_RCR およ び TIM17_RCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]													
	リセット値																									0	0	0	0	0	0	0	0	0					
0x34	TIM16_CCR1 およ び TIM17_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x44	TIM16_BDTR およ び TIM17_BDTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]	DT[7:0]														
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						



表 67. TIM16 および TIM17 のレジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x48	TIM16_DCR および TIM17_DCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]				Res.	Res.	Res.	DBA[4:0]					
	リセット値																				0	0	0	0	0				0	0	0	0	0
0x4C	TIM16_DMAR およ び TIM17_DMAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

20 基本タイマ (TIM6/TIM7)

このセクションは、STM32F05x、STM32F07x、および STM32F09x デバイスにのみ適用されます。TIM7 は、STM32F07x および STM32F09x デバイスでのみ使用できます。

20.1 TIM6/TIM7 の概要

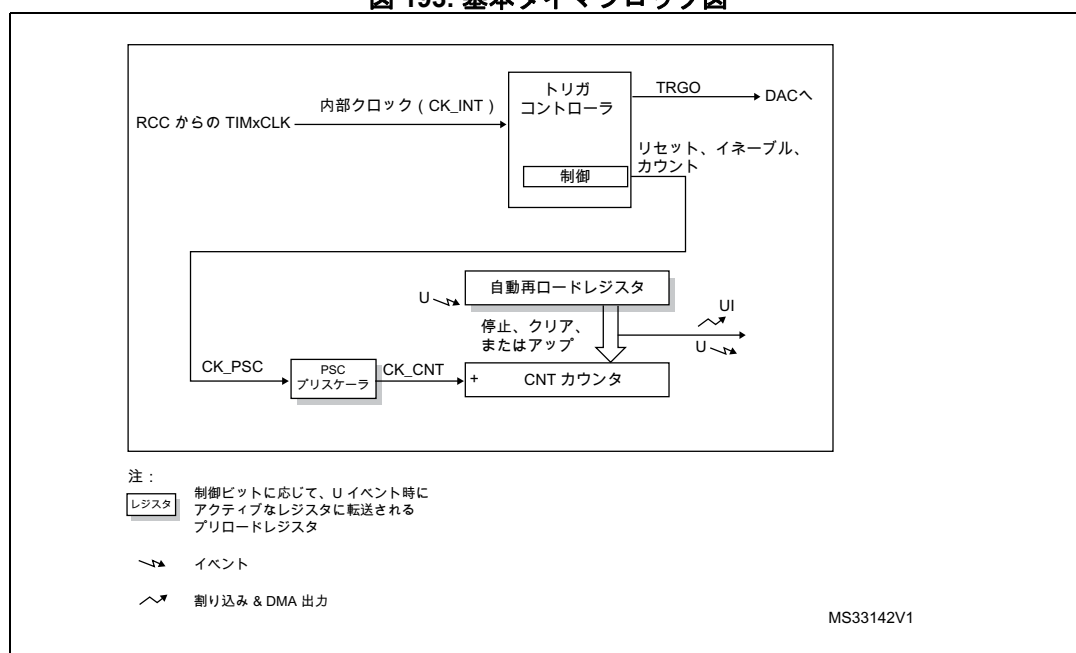
基本タイマ TIM6 は、プログラム可能なプリスケアラで駆動される 16 ビット自動再ロードカウンタで構成されます。

このタイマは、タイムベース生成を目的とした汎用タイマとして使用できますが、特にデジタルアナログコンバータ (DAC) の駆動にも使用されます。実際、TIM6 は内部で DAC に接続されており、トリガ出力を通じて DAC を駆動できます。

20.2 TIM6/TIM7 の主な機能

- 16 ビット自動再ロードアップカウンタ
- 16 ビットのプログラム可能なプリスケアラ (動作中も変更可能) で、カウンタクロック周波数を 1 から 65535 の間の値で分周可能。
- DAC をトリガする同期回路
- 更新イベント時の割り込み/DMA 生成: カウンタオーバーフロー

図 193. 基本タイマブロック図



20.3 TIM6/TIM7 の機能詳細

20.3.1 タイムベースユニット

プログラム可能なタイマのメインブロックは、自動再ロードレジスタを持つ 16 ビットアップカウンタです。カウンタのクロックは、プリスケアラによって分周できます。

カウンタ、自動再ロードレジスタ、およびプリスケアラレジスタは、ソフトウェアで読み書きができます。カウンタが動作中でも、読み書きが可能です。

タイムベースユニットには、次のレジスタで構成されます。

- カウンタレジスタ (TIMx_CNT)
- プリスケアラレジスタ (TIMx_PSC)
- 自動再ロードレジスタ (TIMx_ARR)

自動再ロードレジスタはプリロードされます。自動再ロードレジスタの読み書きはプリロードレジスタへのアクセスとなります。プリロードレジスタの内容は、TIMx_CR1 レジスタの自動再ロードプリロードイネーブルビット (ARPE) に応じて、常時または更新イベント UEV ごとに、シャドウレジスタに転送されます。TIMx_CR1 レジスタの UDIS ビットが 0 である場合、カウンタがオーバーフロー値に達すると、更新イベントが送信されます。また、ソフトウェアで生成することもできます。更新イベントの生成については、各設定の詳細が説明されています。

カウンタのクロックは、TIMx_CR1 レジスタのカウントイネーブルビット (CEN) がセットされているときのみ、プリスケアラ出力 CK_CNT から供給されます。

実際のカウンタイネーブル信号 CNT_EN は、CEN の 1 クロックサイクル後にセットされます。

プリスケアラの説明

プリスケアラは、カウンタクロック周波数を 1 から 65536 の間の値で分周することができます。16 ビットレジスタ (TIMx_PSC レジスタ) を使って制御される 16 ビットカウンタをベースとしています。TIMx_PSC 制御レジスタはバッファされているので、動作中に変更できます。新しいプリスケアラ比は、次の更新イベントで有効になります。

[図 194](#) と [図 195](#) に、プリスケアラ比を動作中に変更したときのカウンタの動作の例を示します。

図 194. プリスケアラ分周比が 1 から 2 に変化したときのカウンタのタイミング図

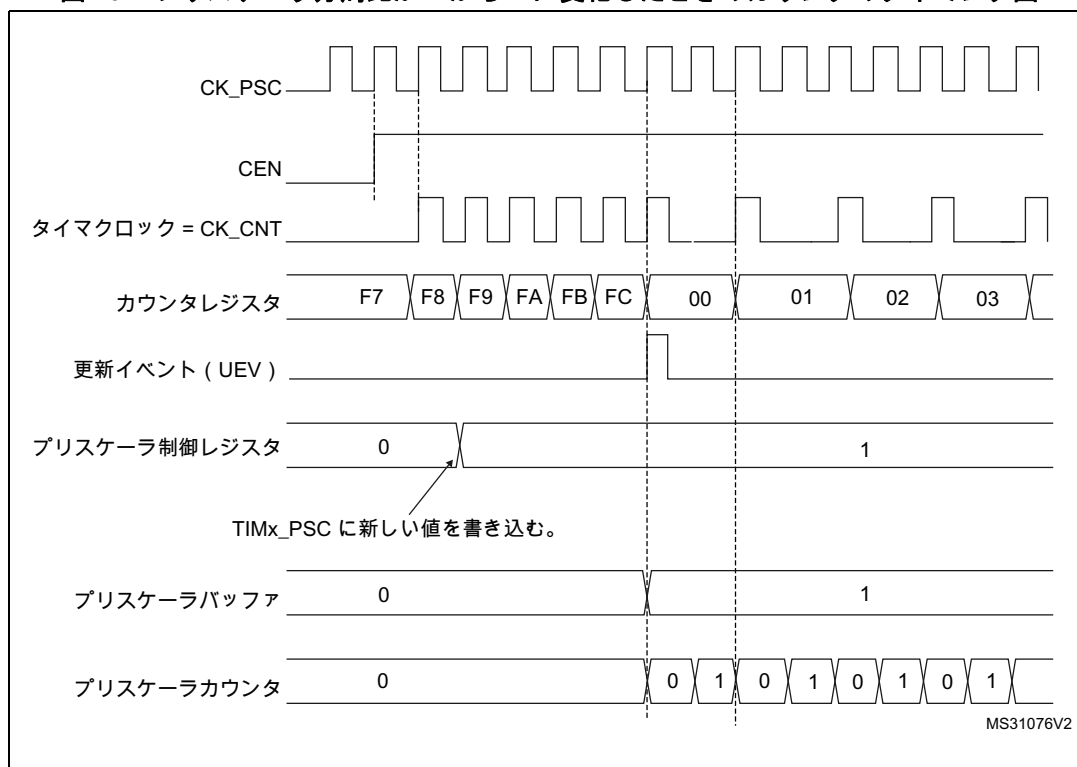
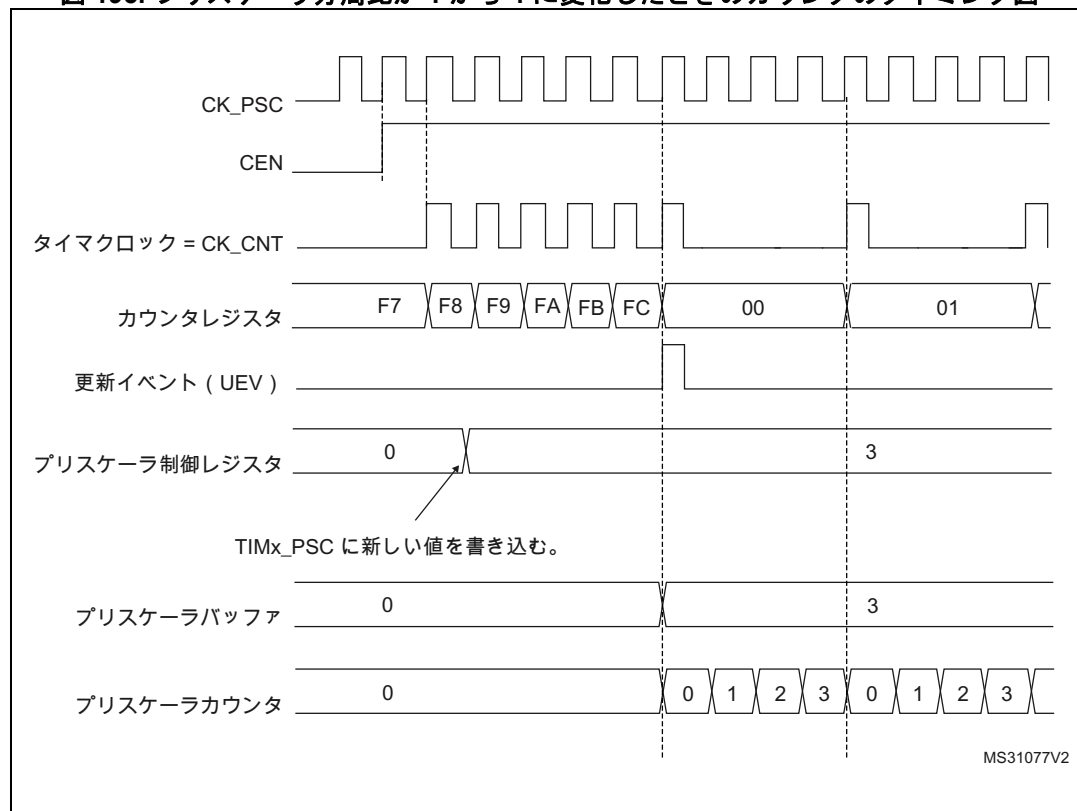


図 195. プリスケアラ分周比が 1 から 4 に変化したときのカウンタのタイミング図



20.3.2 カウンタモード

カウンタは、0 から自動再ロード値 (TIMx_ARR レジスタの内容) までカウントした後、0 からカウントをリスタートして、カウンタオーバーフローイベントを生成します。

更新イベントは、カウンタオーバーフローごとに、または、TIMx_EGR レジスタの UG ビットをセットすることによって (ソフトウェアで、または、スレーブモードコントローラを使用して) 生成できます。

UEV イベントは、ソフトウェアで TIMx_CR1 レジスタの UDIS ビットをセットすることによって無効にできます。これは、プリロードレジスタに新しい値を書き込んでいるときにシャドウレジスタが更新されるのを防ぐためです。このようにすると、UDIS ビットに 0 が書き込まれるまで更新イベントは発生しませんが、カウンタとプリスケアラカウンタは両方とも 0 からリスタートします (ただし、プリスケアラ比は変化しません)。さらに、TIMx_CR1 レジスタの URS (更新リクエスト選択) ビットがセットされている場合、UG ビットをセットすると更新イベント UEV が生成されますが、UIF フラグはセットされません (割り込みや DMA リクエストは送信されません)。

更新イベントが発生すると、すべてのレジスタが更新され、URS ビットの設定に応じて、更新フラグ (TIMx_SR レジスタの UIF ビット) がセットされます。

- プリスケアラのバッファにはプリロード値 (TIMx_PSC レジスタの内容) が再ロードされます。
- 自動再ロードシャドウレジスタは、プリロード値 (TIMx_ARR) で更新されます。

以下の図は、TIMx_ARR = 0x36 のときの、さまざまなクロック周波数におけるカウンタの動作の例を示します。

図 196. 内部クロック分周比が 1 の場合のカウンタのタイミング図

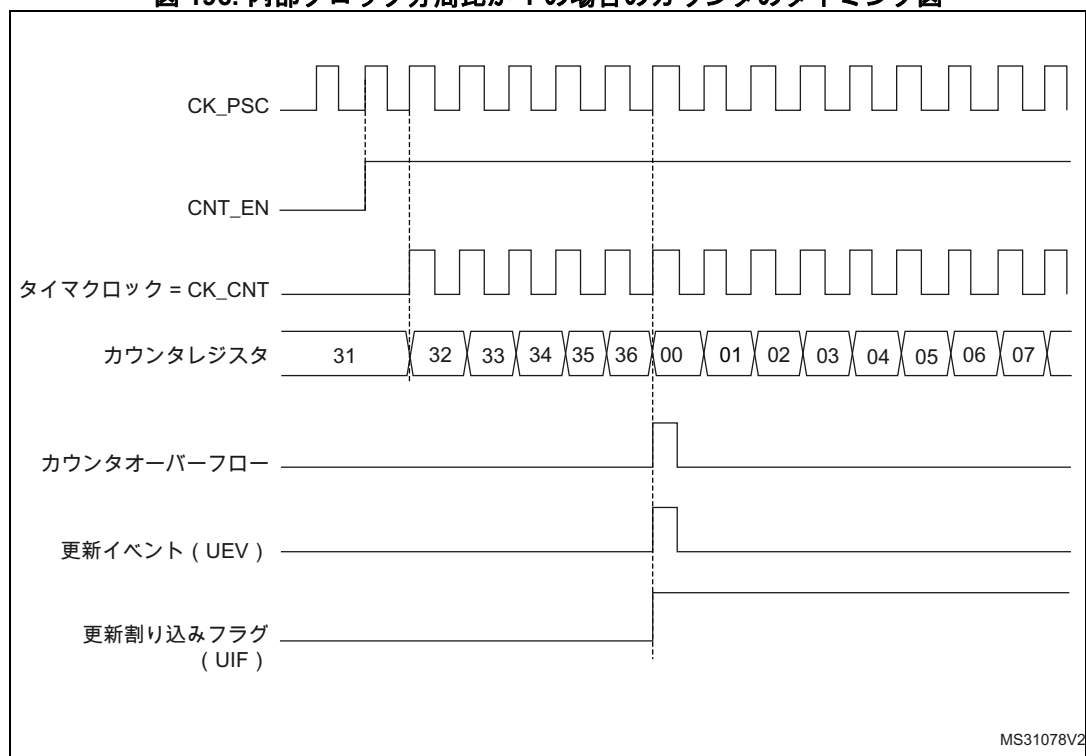


図 197. 内部クロック分周比が 2 の場合のカウンタのタイミング図

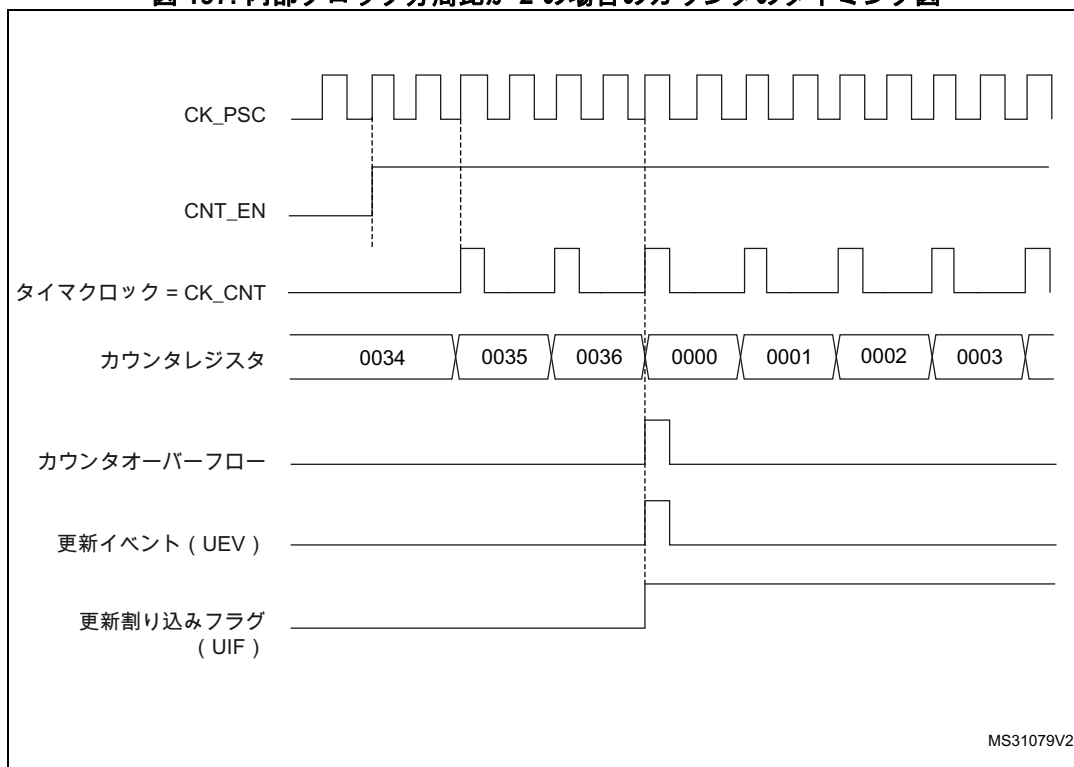


図 198. 内部クロック分周比が 4 の場合のカウンタのタイミング図

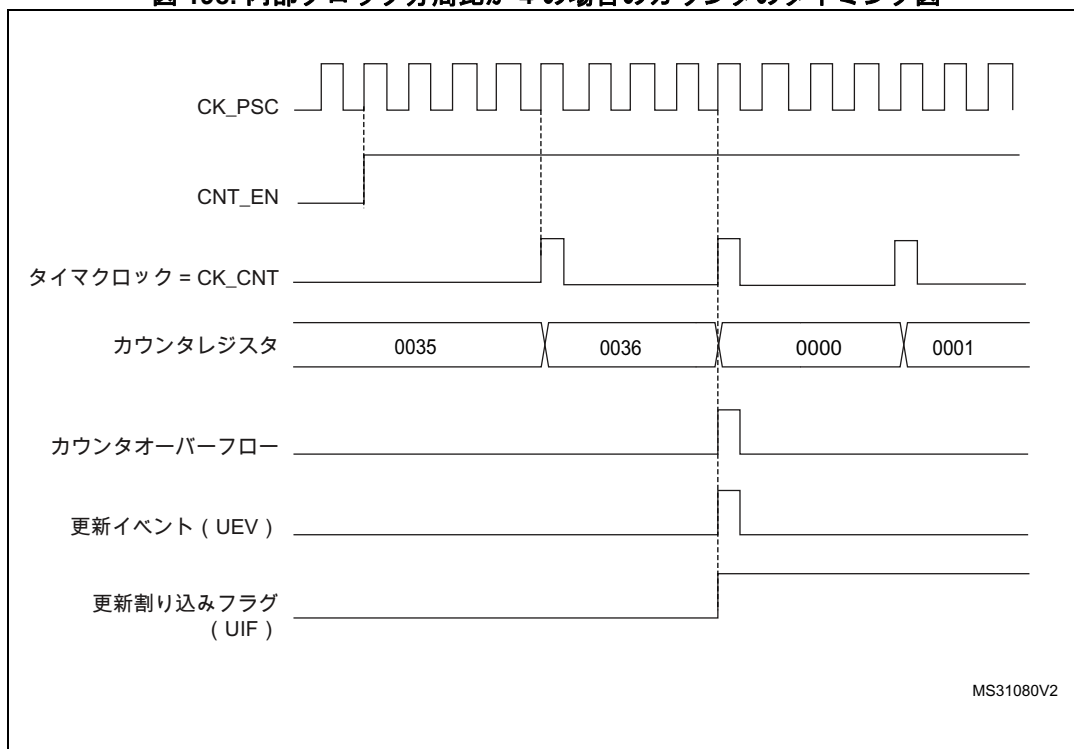


図 199. 内部クロック分周比が N の場合のカウンタのタイミング図

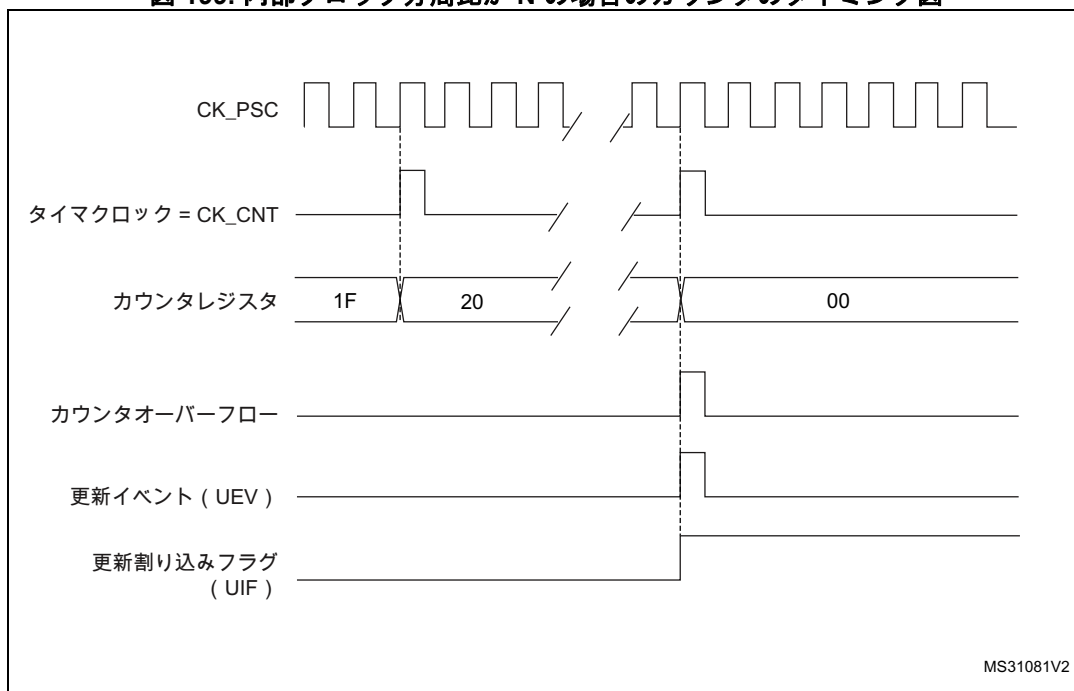


図 200. ARPE = 0 (TIMx_ARR はプリロードされない) の場合の更新イベント時のカウンタのタイミング図

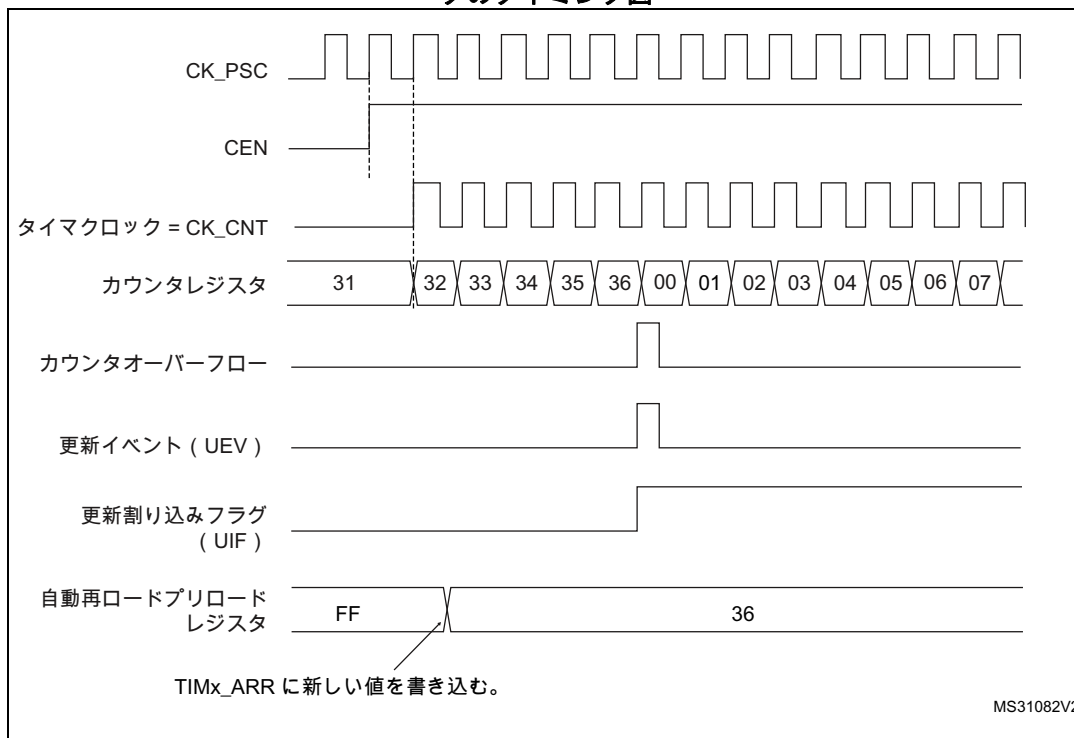
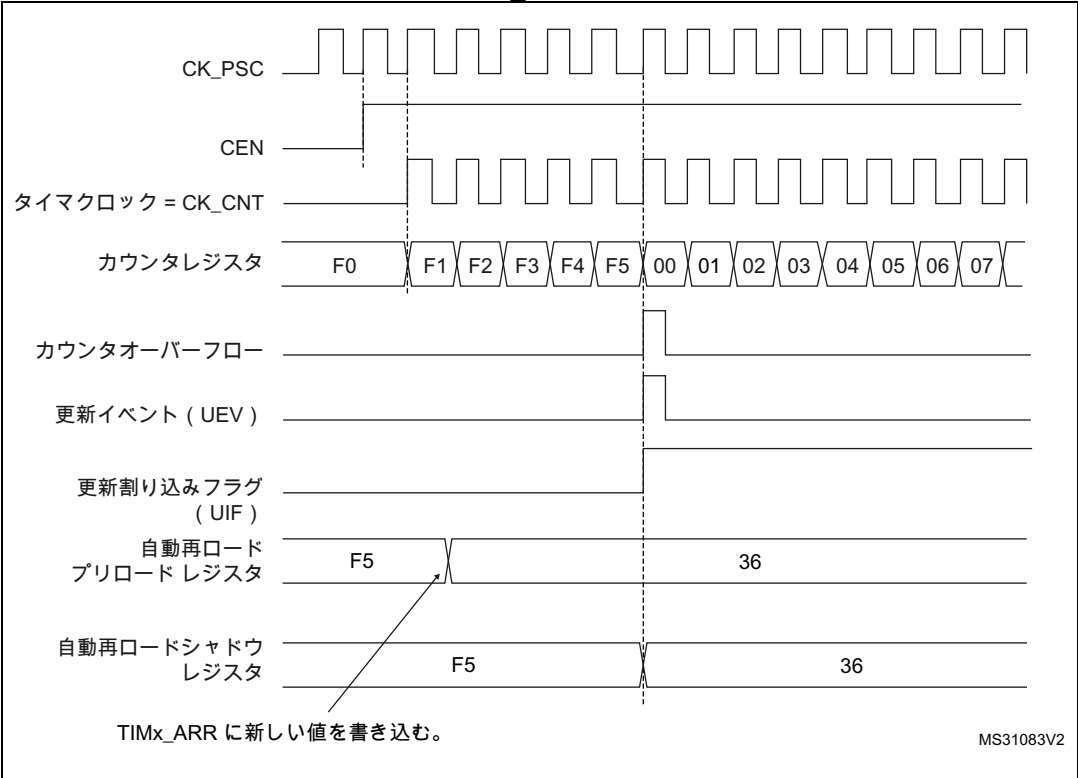


図 201. ARPE=1 (TIMx_ARR はプリロード) の場合の更新イベント時のカウンタのタイミング図 (TIMx_ARR はプリロード)



20.4 TIM6/TIM7 レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

20.4.1 TIM6/TIM7 制御レジスタ 1（TIMx_CR1）

アドレスオフセット：0x00

リセット値：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
								rw				rw	rw	rw	rw

ビット 15:8 予約済み、常に 0 として読み出されます。

ビット 7 **ARPE**：自動再ロードプリロードイネーブル

0：TIMx_ARR レジスタはバッファされません。

1：TIMx_ARR レジスタはバッファされます。

ビット 6:4 予約済み、常に 0 として読み出されます。

ビット 3 **OPM**：ワンパルスモード

0：カウンタは更新イベントで停止しません。

1：カウンタは次の更新イベントでカウントを停止します（CEN ビットをクリア）。

ビット 2 URS : 更新リクエストソース

このビットは、UEV イベントソースを選択するために、ソフトウェアによってセット／クリアされます。

0 : 次のイベントのいずれかが更新割り込みまたは DMA リクエストを生成します（有効な場合）。これらのイベントは、次のとおりです。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

1 : カウンタオーバーフロー／アンダーフローのみが更新割り込みまたは DMA リクエストを生成します（有効な場合）。

ビット 1 UDIS : 更新ディセーブル

このビットは、UEV イベント生成を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : UEV は有効です。更新イベント（UEV）は、次のいずれかのイベントによって生成されます。

- カウンタオーバーフロー／アンダーフロー
- UG ビットのセット
- スレーブモードコントローラからの更新生成

バッファを持つレジスタにはプリロード値がロードされます。

1 : UEV は無効です。更新イベントは生成されず、シャドウレジスタ（ARR、PSC）はそれぞれの値を維持します。ただし、UG ビットがセットされた場合や、スレーブモードコントローラからハードウェアリセットを受信した場合には、カウンタとプリスケアラは再初期化されます。

ビット 0 CEN : カウンタイネーブル

0 : カウンタは無効です。

1 : カウンタは有効です。

注： ゲートモードは、CEN ビットが事前にソフトウェアでセットされている場合にのみ動作します。ただし、トリガモードでは、ハードウェアによって自動的に CEN ビットをセットできます。

ワンパルスモードでは、更新イベントが発生すると、CEN は自動的にクリアされます。

20.4.2 TIM6/TIM7 制御レジスタ 2 (TIMx_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									rw	rw	rw				

ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **MMS** : マスタモード選択

これらのビットは、同期のためにマスタモードでスレーブタイマに送信される情報 (TRGO) を選択するために使用します。組み合わせは、次のとおりです。

000 : **リセット** - TIMx_EGR レジスタの UG ビットがトリガ出力 (TRGO) として使用されます。トリガ入力によってリセットが発生したとき (スレーブモードコントローラがリセットモードに設定されているとき) TRGO 信号は実際のリセットから遅れて発生します。

001 : **イネーブル** - カウンタイネーブル信号 CNT_EN がトリガ出力 (TRGO) として使用されます。これは、いくつかのタイマを同時に開始するときや、スレーブタイマが有効な時間枠を制御するときに役立ちます。カウンタイネーブル信号は、ゲートモードに設定されているとき、CEN 制御ビットとトリガ入力との論理和 (OR) によって生成されます。

カウンタイネーブル信号がトリガ入力によって制御されているとき、マスタ/スレーブモードが選択されている場合を除いて、TRGO に遅延が存在します (TIMx_SMCR レジスタの MSM ビットの説明を参照してください)。

010 : **更新** - 更新イベントがトリガ出力 (TRGO) として選択されます。たとえば、マスタタイマをスレーブタイマのプリスケアラとして使用できます。

ビット 3:0 予約済み、常に 0 として読み出されます。

20.4.3 TIM6/TIM7 DMA/割り込み有効レジスタ (TIMx_DIER)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **UDE** : 更新 DMA リクエストイネーブル

0 : 更新 DMA リクエストは無効です。

1 : 更新 DMA リクエストは有効です。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIE** : 更新割り込みイネーブル

0 : 更新割り込みは無効です。

1 : 更新割り込みは有効です。

20.4.4 TIM6/TIM7 のステータスレジスタ (TIMx_SR)

アドレスオフセット : 0x10

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UIF** : 更新割り込みフラグ

このビットは、更新イベント時にハードウェアによってセットされます。ソフトウェアによってクリアされます。

0 : 更新は発生していません。

1 : 更新割り込みが保留中です。このビットは、レジスタが以下で更新されたときにハードウェアによってセットされます。

–TIMx_CR1 レジスタの UDIS = 0 であり、繰り返しカウンタ値でオーバーフローまたはアンダーフローが発生したとき。

–TIMx_CR1 レジスタの URS = 0 かつ UDIS = 0 の場合に、TIMx_EGR レジスタの UG ビットを使用して、ソフトウェアで CNT が再初期化されたとき。

20.4.5 TIM6/TIM7 のイベント生成レジスタ (TIMx_EGR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

ビット 15:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **UG** : 更新生成

このビットは、ソフトウェアによってセットでき、ハードウェアによって自動的にクリアされます。

0 : 影響はありません。

1 : タイマカウンタを再初期化して、レジスタの更新を生成します。プリスケアラカウンタもクリアされます（プリスケアラ比は変化しません）。

20.4.6 TIM6/TIM7 のカウンタ (TIMx_CNT)

アドレスオフセット : 0x24

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **CNT[15:0]** : カウンタ値



20.4.7 TIM6/TIM7 プリスケーラ (TIMx_PSC)

アドレスオフセット : 0x28

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **PSC[15:0]** : プリスケーラ値
カウンタクロック周波数 CK_CNT は $f_{CK\_PSC} / (PSC[15:0] + 1)$ に等しいです。
PSC は、更新イベントごとにアクティブなプリスケーラレジスタにロードされる値を含みます。

20.4.8 TIM6/TIM7 自動再ロードレジスタ (TIMx_ARR)

アドレスオフセット : 0x2C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 15:0 **ARR[15:0]** : 自動再ロード値
ARR は、実際の自動再ロードレジスタにロードされる値です。
APR の更新と動作の詳細については、[セクション 20.3.1 : タイムベースユニット \(539 ページ\)](#) を参照してください。
自動再ロード値が null のときには、カウンタはブロックされます。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

20.4.9 TIM6/TIM7 レジスタマップ

TIMx レジスタは、次の表のように、16 ビットアドレス可能レジスタとして配置されます。

表 68. TIM6/TIM7 レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIMx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARPE	Res.	Res.	Res.		OPM	URS	UDIS	CEN
	リセット値																								0				0	0	0	0	
0x04	TIMx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		MMS[2:0]			Res.	Res.	Res.	Res.	
	リセット値																								0	0	0						
0x0C	TIMx_DIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																							0								0	
0x10	TIMx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																															0	
0x14	TIMx_EGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	
	リセット値																															0	
0x24	TIMx_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x28	TIMx_PSC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x2C	TIMx_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

21 赤外線インタフェース (IRTIM)

このデバイスではリモートコントロールで赤外線インタフェース (IRTIM) を使用できます。リモートコントロール機能を実行するために、赤外線 LED と併せて使用できます。

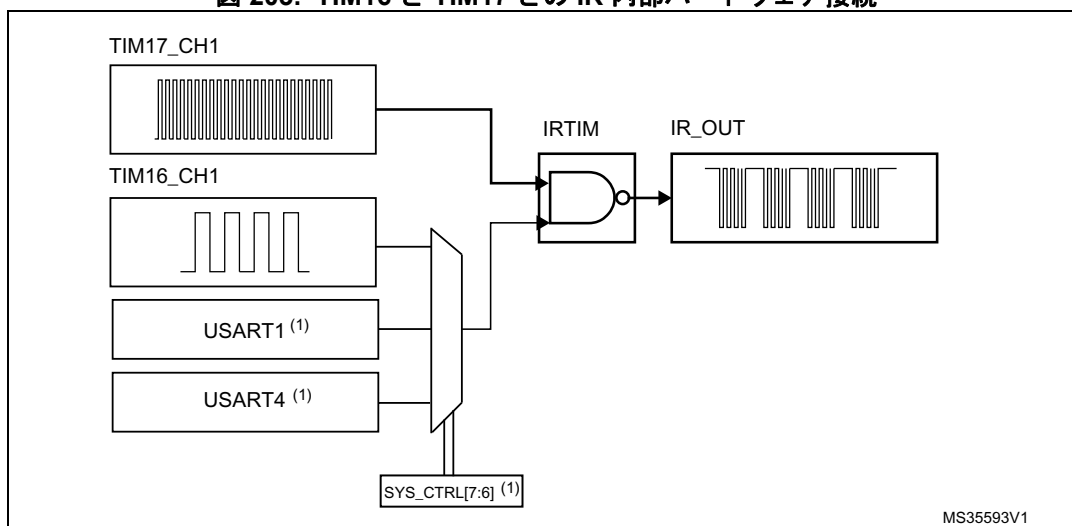
図 203 に示すように、USART1、USART4、TIM16、および TIM17 との間で内部接続を使用します。

赤外線リモートコントロール信号を生成するには、正しい波形を生成するために IR インタフェースを有効にし、TIM16 チャンネル 1 (TIM16_OC1) と TIM17 チャンネル 1 (TIM17_OC1) を適切に設定する必要があります。

赤外線レシーバは、基本的な入力キャプチャモードで簡単に実装できます。

STM32F09x デバイスでは、SYS_CTRL1[7:6] による選択に応じて、USART1 または USART4 の変化した波形を IR_OUT から送信できます。USARTx_TX 出力信号は、TIM17_CH1 によって生成されたキャリア周波数によって変化します。

図 203. TIM16 と TIM17 との IR 内部ハードウェア接続



1. USART1 と USART4 は STM32F09x デバイスでのみ IRTIM とリンクできます。

標準の IR パルス変調モードはすべて 2 つのタイマ出力比較チャンネルをプログラミングすることで取得できます。

TIM16 は包絡線変調を生成し、TIM17 は高周波キャリア信号を生成するために使用されます。

赤外線機能は IR_OUT ピンから出力されます。この機能の有効化は、GPIOx_AFRx レジスタを通じ、関連するオルタネート機能ビットを有効にすることで行われます。

大電流シンク LED ドライバ機能 (PB9 ピンでのみ使用可能) の有効化は SYSCFG_CFGR1 レジスタの I2C_PB9_FMP ビットを通じて行われ、赤外線 LED を直接制御するために必要な大電流を流すために使用されます。

コード例については、「付録」セクション [A.10.1 : TIM16 および TIM17 設定のコード例](#) を参照してください。

22 独立型ウォッチドッグ (IWDG)

22.1 概要

デバイスは、内蔵ウォッチドッグペリフェラルを搭載しており、使用上、高い安全レベル、タイミングの正確さ、および柔軟性を兼ね備えています。独立型ウォッチドッグペリフェラルは、ソフトウェア障害による誤動作を検出および解決し、カウンタが与えられたタイムアウト値に達すると、システムリセットをトリガします。

独立型ウォッチドッグ (IWDG) は、独自の低速クロック (LSI) によってクロック供給されるので、メインクロックに障害があってもアクティブなままです。

IWDG は、メインアプリケーションの外部で、完全に独立したプロセスとして実行するウォッチドッグが必要な場合に最適ですが、タイミング精度が低いという制約があります。ウィンドウ型ウォッチドッグの詳細については、[セクション 23 \(562 ページ\)](#) を参照してください。

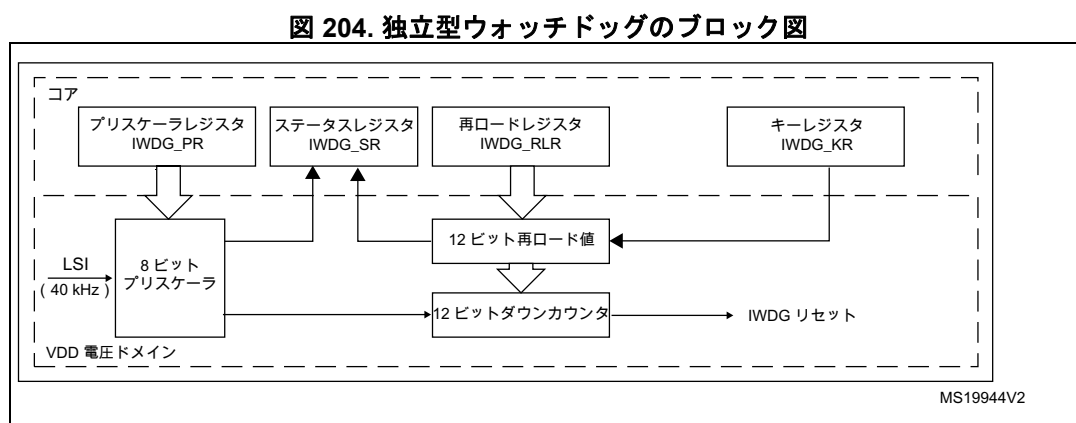
22.2 IWDG の主な機能

- フリーランニングダウンカウンタ
- 独立した RC オシレータからのクロック供給 (STANDBY および STOP モードで動作可能)
- 条件付きリセット
 - ダウンカウンタの値が 0x000 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
 - ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合)。

22.3 IWDG の機能説明

22.3.1 IWDG ブロック図

[図 204](#) に、独立型ウォッチドッグモジュールの機能ブロックを示します。



注： ウォッチドッグ機能は、STOP モードおよびSTANDBY モードでも機能する CORE 電圧ドメインに実装されています。

キーレジスタ (IWDG_KR) に値 0x0000 CCCC が書き込まれることによって独立型ウォッチドッグが開始すると、カウンタはリセット値 0xFFFF からカウントダウンを開始します。カウント値の終わり (0x000) に達すると、リセット信号が生成されます (IWDG_reset)。

IWDG_KR レジスタにキー値 0x0000 AAAA が書き込まれると、IWDG_RLR の値がカウンタに再ロードされ、ウォッチドッグのリセットが防止されます。

22.3.2 ウィンドウオプション

IWDG は、IWDG_WINR レジスタに適切なウェインドウをセットすることによって、ウィンドウ型ウォッチドッグとしても機能します。

カウンタがウィンドウレジスタ (IWDG_WINR) に格納された値より大きい間に再ロード操作が行われると、リセットが生成されます。

IWDG_WINR のデフォルト値は 0x0000 0FFF です。この値が更新されない場合は、ウィンドウオプションは無効にされます。

ウィンドウ値が変わるとすぐに再ロード操作が行われ、ダウンカウンタを IWDG_RLR 値にリセットし、次の再ロードを生成するためのサイクル数計算を容易にします。

ウィンドウオプションが有効な場合の IWDG の設定

1. IWDG_KR レジスタに 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. IWDG_KR レジスタに 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. IWDG_PR を 0 から 7 までプログラムすることによって、IWDG のプリスケラに書き込みを行います。
4. 再ロードレジスタ (IWDG_RLR) に書き込みます。
5. レジスタが更新されるのを待ちます (IWDG_SR = 0x0000 0000)。
6. ウィンドウレジスタ IWDG_WINR に書き込みます。これにより、カウンタ値 IWDG_RLR が自動的にリフレッシュされます。

注： **ウィンドウ値を書き込むことで、IWDG_SR が"0x0000 0000"にセットされた時点でカウンタ値をRLR でリフレッシュすることができます。**

コード例については、「付録」セクション [A.15.2：ウィンドウによる IWDG 設定のコード例](#) を参照してください。

ウィンドウオプションが無効な場合の IWDG の設定

ウィンドウオプションが使用されていない場合、IWDG は以下のように設定することができます。

1. IWDG_KR レジスタに 0x0000 CCCC を書き込むことによって、IWDG を有効にします。
2. IWDG_KR レジスタに 0x0000 5555 を書き込むことによって、レジスタのアクセスを有効にします。
3. IWDG_PR を 0 から 7 までプログラムすることによって、IWDG のプリスケラに書き込みを行います。
4. 再ロードレジスタ (IWDG_RLR) に書き込みます。
5. レジスタが更新されるのを待ちます (IWDG_SR = 0x0000 0000)。
6. カウンタ値を IWDG_RLR (IWDG_KR = 0x0000 AAAA) でリフレッシュします。

コード例については、「付録」セクション [A.15.1：IWDG 設定のコード例](#) を参照してください。

22.3.3 ハードウェアウォッチドッグ

デバイスのオプションビットを使って「ハードウェアウォッチドッグ」機能が有効化されると、ウォッチドッグは電源投入時に自動的に有効になり、カウンタがカウントの終わりに達する前にソフトウェアによってキーレジスタへ書き込まれない限り、またはダウンカウンタがウィンドウ内に再ロードされた場合は、リセットを生成します。

22.3.4 レジスタのアクセス保護

IWDG_PR、IWDG_RLR、および IWDG_WINR レジスタへの書き込みアクセスは保護されます。これらを変更するには、まず、IWDG_KR レジスタにコード 0x0000 5555 を書き込む必要があります。このレジスタに別の値を書き込むと、シーケンスがブレイクされ、レジスタへのアクセスが再び保護されます。これは、再ロード操作 (0x0000 AAAA の書き込み) であることを意味します。ステータスレジスタは、プリスケアラの更新、あるいはダウンカウンタ再ロード値やウィンドウ値の更新が行われていることを示すために使用されます。

コード例については、「付録」セクション [A.15.1 : IWDG 設定のコード例](#) を参照してください。

22.3.5 デバッグモード

マイクロコントローラがデバッグモードになると (コアは停止状態)、IWDG カウンタは、DBG モジュールの DBG_IWDG_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。

22.4 IWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード (16 ビット) またはワード (32 ビット) 単位でアクセスする必要があります。

22.4.1 キーレジスタ (IWDG_KR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされる)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:0 **KEY[15:0]** : キー値 (書き込み専用、読み出しでは 0x0000)

これらのビットには、ソフトウェアによって一定間隔でキー値 0xAAAA が書き込まれなければなりません。そうしないと、カウンタが 0 に達した時点でウォッチドッグがリセットを生成します。

キー値 0x5555 を書き込むことによって、IWDG_PR、IWDG_RLR、および IWDG_WINR レジスタへのアクセスが可能になります ([セクション 22.3.4 : レジスタのアクセス保護](#)を参照)。

キー値 CCCCh を書き込むと、ウォッチドッグが開始します (ハードウェアウォッチドッグオプションが選択されている場合を除く)。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

22.4.2 プリスケアラレジスタ (IWDG_PR)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
													rw	rw	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2:0 **PR[2:0]** : プリスケアラ分周回路

これらのビットは、書き込みアクセス保護されています（[セクション 22.3.4 : レジスタのアクセス保護](#)を参照）。カウンタクロックを供給するプリスケアラ分周回路を選択するようにソフトウェアで書き込まれます。プリスケアラ分周比を変更できるようにするには、IWDG_SR レジスタの PVU ビットをリセットする必要があります。

- 000 : 4 分周
- 001 : 8 分周
- 010 : 16 分周
- 011 : 32 分周
- 100 : 64 分周
- 101 : 128 分周
- 110 : 256 分周
- 111 : 256 分周

注 : このレジスタを読み出すと、VDD 電圧ドメインからプリスケアラ値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG_SR レジスタの PVU ビットがリセットされているときのみとなります。

22.4.3 再ロードレジスタ (IWDG_RLR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RL[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **RL[11:0]** : ウォッチドッグカウンタ再ロード値

これらのビットは、書き込みアクセス保護されています ([セクション 22.3.4](#)を参照)。IWDG_KR レジスタに値 0xAAAA が書き込まれるたびにウォッチドッグカウンタにロードされる値を定義するために、ソフトウェアで書き込まれます。ウォッチドッグカウンタは、この値からカウントダウンします。タイムアウトまでの時間は、この値とクロックプリスケールによって決まります。タイムアウトに関する詳細はデータシートを参照してください。

再ロード値を変更できるようにするには、IWDG_SR レジスタの RVU ビットをリセットする必要があります。

注 : このレジスタを読み出すと、VDD 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合、この値は最新でないか、有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG_SR レジスタの RVU ビットがリセットされているときのみとなります。

22.4.4 ステータスレジスタ (IWDG_SR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000 (STANDBY モードによりリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
													r	r	r

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **WVU** : ウォッチドッグカウンタウィンドウ値の更新

このビットは、ウィンドウ値の更新が進行中であることを示すために、ハードウェアによってセットされます。 V_{DD} 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

ウィンドウ値は、WVU ビットがリセットされているときのみ更新できます。

このビットは、一般のウィンドウが 1 の場合のみ生成されます。

ビット 1 **RVU** : ウォッチドッグカウンタ再ロード値の更新

このビットは、再ロード値の更新が進行中であることを示すために、ハードウェアによってセットされます。 V_{DD} 電圧ドメインで再ロード値の更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

再ロード値は、RVU ビットがリセットされているときのみ更新できます。

ビット 0 **PVU** : ウォッチドッグプリスケアラ値の更新

このビットは、プリスケアラ値の更新が進行中であることを示すために、ハードウェアによってセットされます。 V_{DD} 電圧ドメインでプリスケアラ更新操作が完了したときに、ハードウェアによってリセットされます (最大 5 RC 40 kHz サイクルかかる)。

プリスケアラ値は、PVU ビットがリセットされているときのみ更新できます。

注 : 複数の再ロード値、プリスケアラ値、またはウィンドウ値がアプリケーションで使用される場合は、それぞれ、再ロード値を変更する前に RVU ビットがリセットされるまで待つか、プリスケアラ値を変更する前に PVU ビットがリセットされるまで待つか、またはウィンドウ値を変更する前に WVU ビットがリセットされるまで待つ必要があります。ただし、プリスケアラ値、再ロード値、またはウィンドウ値を更新した後は、RVU、PVU、または WVU がリセットされるのを待たずに、コード実行を続けることができます (低電力モードに入った場合を除く)。

22.4.5 ウィンドウレジスタ (IWDG_WINR)

アドレスオフセット : 0x10
リセット値 : 0x0000 0FFF (STANDBY モードによりリセットされる)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WIN[11:0]											
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11:0 **WIN[11:0]** : ウォッチドッグカウンタウィンドウ値

これらのビットは、書き込みアクセス保護されています ([セクション 22.3.4](#)を参照)。これらのビットは、ダウンカウンタと比較されるウィンドウ値の上限を含みます。

リセットを防ぐには、カウンタの値がウィンドウレジスタの値よりも小さく、0x0 よりも大きい間にダウンカウンタを再ロードする必要があります。

再ロード値を変更できるようにするには、IWDG_SR レジスタの WVU ビットをリセットする必要があります。

注： このレジスタを読み出すと、V_{DD} 電圧ドメインから再ロード値が返されます。このレジスタへの書き込み操作が進行中の場合には、この値は有効でないことがあります。このため、このレジスタから読み出された値が有効なのは、IWDG_SR レジスタの WVU ビットがリセットされているときのみとなります。

22.4.6 IWDG レジスタマップ

次の表に、IWDG レジスタマップとリセット値を示します。

表 69. IWDG レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	IWDG_KR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY[15:0]															
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	IWDG_PR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PR[2:0]		
	リセット値																														0	0	0
0x08	IWDG_RLR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RL[11:0]											
	リセット値																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	IWDG_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	リセット値																														WVU	RVU	PVU
0x10	IWDG_WINR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WIN[11:0]											
	リセット値																					1	1	1	1	1	1	1	1	1	1	1	1

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

23 システムウィンドウ型ウォッチドッグ (WWDG)

23.1 概要

システムウィンドウ型ウォッチドッグ (WWDG) は、通常、外部の影響や予期しない論理条件などによって発生し、アプリケーションプログラムを正常なシーケンスから逸脱させるソフトウェア障害の発生を検出するために使用されます。ウォッチドッグ回路は、T6 ビットがクリアされる前にプログラムがダウンカウンタの内容をリフレッシュしない限り、プログラムされた時間の経過後に MCU リセットを生成します。MCU リセットは、ダウンカウンタがウィンドウレジスタ値に達する前に 7 ビットのダウンカウンタの値 (制御レジスタ内) がリフレッシュされた場合にも生成されます。このことは、限られた時間枠 (time-window) の間にカウンタがリフレッシュされなければならないことを意味します。

WWDG クロックは、APB クロックから分周され、また設定可能な時間枠 (time-window) があるので、これをプログラムしてアプリケーション動作の異常な進み・遅れを検出できます。

WWDG は、正確な時間枠内で反応するウォッチドッグが必要なアプリケーションに適しています。

23.2 WWDG の主な機能

- プログラム可能なフリーランニングダウンカウンタ
- 条件付きリセット
 - － ダウンカウンタの値が 0x40 より小さくなったときにリセット (ウォッチドッグが有効な場合)。
 - － ダウンカウンタがウィンドウ外で再ロードされた場合にリセット (ウォッチドッグが有効な場合) ([図 206](#)を参照)。
- 早期ウェイクアップ割り込み (EWI) : ダウンカウンタが 0x40 になったときにトリガ (有効であり、ウォッチドッグがアクティブな場合)

23.3 WWDG の機能説明

ウォッチドッグが有効 (WWDG_CR レジスタの WDGA ビットがセットされている) な場合、7 ビットのダウンカウンタ (T[6:0] ビット) が 0x40 に達して 0x3F に戻った (T6 がクリアされた) 時点で、リセットを開始します。カウンタがウィンドウレジスタに格納された値より大きい間にソフトウェアがカウンタを再ロードした場合にも、リセットが生成されます。

Figure 10-10 illustrates the internal logic of the Watchdog timer. The diagram shows the following components and their interconnections:

- リセット (Reset):** A signal input to a NAND gate and a comparator.
- Comparator:** Compares the current counter value (T6:0) with the configured value (W6:0). When $T6:0 > W6:0$, the output is 1.
- WWDG_CR (Watchdog Control Register):** Receives the reset signal and the comparator output. It controls the WWDG_CFR register and the WWDG_CNT register.
- WWDG_CFR (Watchdog Control Register):** A 7-bit register that controls the WWDG_CNT register.
- WWDG_CNT (7-bit Down Counter):** A 7-bit down counter that counts down from 7 to 0.
- WDG プリスケアラ (WDGTB):** A prescaler that divides the PCLK by 4096.
- PCLK (RCC クロックコントローラから):** The system clock provided by the RCC clock controller.

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などにつきましてはいかなる責任にも負いません。

アプリケーションによっては、EWI 割り込みを使用して、WWDG リセットを生成せずにソフトウェアのシステムチェックやシステム復旧／グレースフルデグラデーションを管理することができます。この場合、対応する割り込みサービスルーチン (ISR) で WWDG カウンタを再ロードし、WWDG リセットを回避してから必要な操作をトリガしてください。

EWI 割り込みは、WWDG_SR レジスタの EWIF ビットに“0”を書き込むことによってクリアされます。

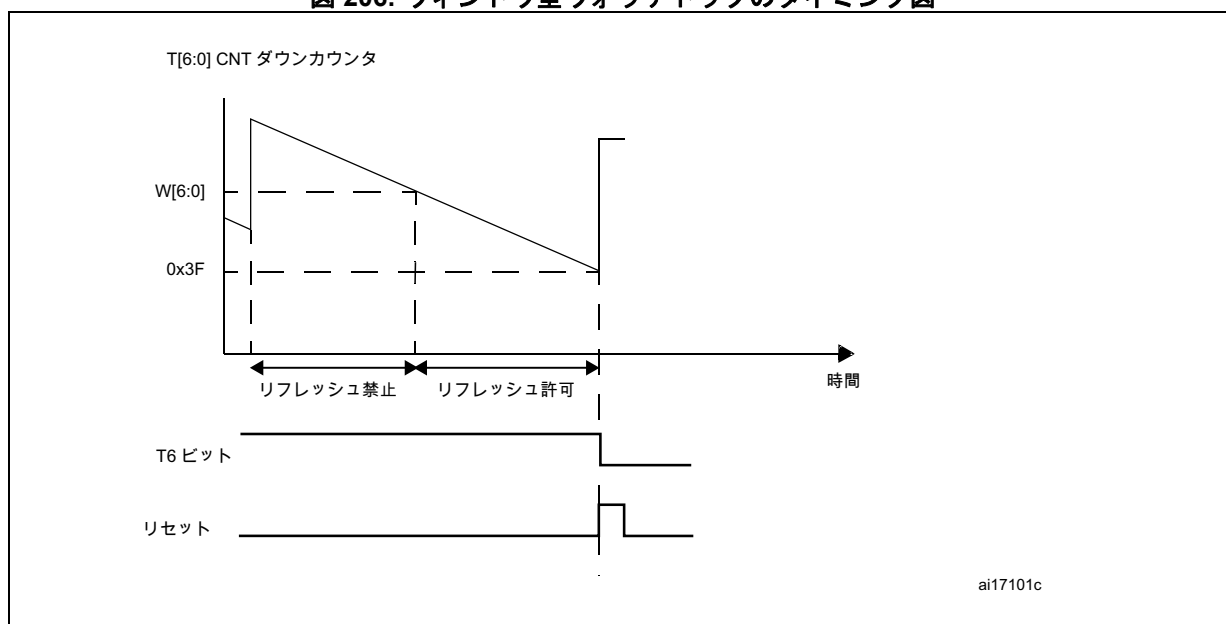
注： 優先順位の高いタスクにおけるシステムロックなどによって EWI 割り込みが使用できない場合、最終的には WWDG リセットが生成されます。

23.3.4 ウォッチドッグタイムアウトをプログラムする方法

図 206 の式を使用して、WWDG のタイムアウトを計算することができます。

警告： WWDG_CR レジスタに書き込むときには、ただちにリセットされるのを防ぐために、常に T6 ビットに 1 を書き込んでください。

図 206. ウィンドウ型ウォッチドッグのタイミング図



タイムアウト値は次の式で算出されます。

$$t_{\text{WWDG}} = t_{\text{PCLK}} \times 4096 \times 2^{\text{WDGTB}[1:0]} \times (\text{T}[5:0] + 1) \quad (\text{ms})$$

ここで、

t_{WWDG} : WWDG タイムアウト

t_{PCLK} : APB クロック周期の測定値 (ms)

4096 : 内部分周器に対応する値

たとえば、APB 周波数が 48 MHz と等しい場合、WDGTB[1:0] は 3 にセットされ、T[5:0] は 63 にセットされます。

$$t_{\text{WWDG}} = 1/48000 \times 4096 \times 2^3 \times (63 + 1) = 43.69 \text{ ms}$$

t_{WWDG} の最小値と最大値については、データシートを参照してください。

23.3.5 デバッグモード

マイクロコントローラがデバッグモードになると（Cortex®-M0 コアは停止状態）、WWDG カウンタは、DBG モジュールの DBG_WWDG_STOP 設定ビットに応じて、通常どおりに動作を続けるか、または停止します。詳細については、[セクション 32.9.2：タイマ、ウォッチドッグ、および I2C のデバッグサポート](#)を参照してください。

23.4 WWDG レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#)を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

23.4.1 制御レジスタ (WWDG_CR)

アドレスオフセット：0x00

リセット値：0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
								rs	rw						

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **WDGA**：有効化ビット

このビットは、ソフトウェアでセットされ、リセット後はハードウェアによってのみクリアされます。WDGA = 1 のとき、ウォッチドッグはリセットを生成できます。

- 0：ウォッチドッグは無効です。
- 1：ウォッチドッグは有効です。

ビット 6:0 **T[6:0]**：7 ビットカウンタ（MSB から LSB まで）

これらのビットは、ウォッチドッグカウンタの値を含みます。(4096 x 2^{WDGTB[1:0]} PCLK サイクルごとにデクリメントされます。0x40 に達して 0x3F に戻ると (T6 がクリアされると)、リセットが生成されます。



23.4.2 設定レジスタ (WWDG_CFR)

アドレスオフセット : 0x04

リセット値 : 0x0000 007F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB[1:0]		W[6:0]						
						rs	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **EWI** : 早期ウェイクアップ割り込み

このビットがセットされているときには、カウンタの値が 0x40 に達したときに割り込みが発生します。
この割り込みは、リセット後にハードウェアによってのみクリアされます。

ビット 8:7 **WDGTB[1:0]** : タイマーベース

プリスケアラのタイムベースは、次のように変更できます。

- 00 : CK カウンタクロック (PCLK/4096) 1 分周
- 01 : CK カウンタクロック (PCLK/4096) 2 分周
- 10 : CK カウンタクロック (PCLK/4096) 4 分周
- 11 : CK カウンタクロック (PCLK/4096) 8 分周

ビット 6:0 **W[6:0]** : 7 ビットウィンドウ値

これらのビットは、ダウンカウンタと比較されるウィンドウ値を含みます。

23.4.3 ステータスレジスタ (WWDG_SR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

ビット 31:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **EWIF** : 早期ウェイクアップ割り込みフラグ

このビットは、カウンタの値が 0x40 に達したときにハードウェアによってセットされます。“0”を書き込んでソフトウェアでクリアする必要があります。“1”を書き込んでも、ビットの値は変化しません。このビットは、割り込みが有効でない場合にもセットされます。

23.4.4 WWDG レジスタマップ

次の表に、WWDG のレジスタマップとリセット値を示します。

表 70. WWDG レジスタマップとリセット値

オフセット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	WWDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		WDGA	T[6:0]										
	リセット値																									0	1	1	1	1	1	1	1				
0x04	WWDG_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB1	WDGTB0	W[6:0]										
	リセット値																							0	0	0	1	1	1	1	1	1	1				
0x08	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF				
	リセット値																																0				

レジスタ境界アドレスについては、[セクション 2.2.2 : メモリマップとレジスタ境界アドレス](#) を参照してください。

24 リアルタイムクロック (RTC)

24.1 概要

RTC は、あらゆる低電力モードを管理する自動ウェイクアップ機能を提供します。

本リアルタイムクロック (RTC) は、独立した BCD タイマ／カウンタです。RTC は、プログラム可能なアラーム割り込み機能を備えた時刻／カレンダーを搭載しています。

また、割り込み機能を備えたプログラム可能な周期的ウェイクアップフラグも搭載しています。

2 つの 32 ビットレジスタには、2 進化 10 進数形式 (BCD) で表現した秒、分、時 (12 時間または 24 時間形式)、曜日、日、月、年が含まれています。サブセカンドの値もバイナリ形式で利用できます。

28 日、29 日 (うるう年)、30 日、31 日の補正は、自動的に行われます。サマータイム補正も行われます。

サブセカンド、秒、分、時、曜日、日付のプログラム可能なアラームを備えた 32 ビットレジスタが追加されています。

クリスタルオシレータ精度の偏差を補正するために、デジタル較正機能が利用可能です。

RTC ドメインリセット後、すべての RTC レジスタは、起こりうる不要な書き込みアクセスから保護されます。

供給電圧が動作範囲内にある間は、デバイスのステータス (実行モード、低電力モード、またはリセット中) に関係なく、RTC が停止することはありません。

24.2 RTC の主な機能

RTC ユニットの主な機能を以下に示します (図 207: STM32F03x、STM32F04x、および STM32F05x デバイスの RTC ブロック図および図 208: STM32F07x および STM32F09x デバイスの RTC ブロック図を参照)。

- サブセカンド、秒、分、時 (12 または 24 時間形式)、曜日、日、月、年に対応するカレンダー。
- ソフトウェアでプログラム可能なサマータイム補正。
- 割り込み機能を備えたプログラム可能なアラーム。アラームは、カレンダー項目のどの組み合わせでもトリガ可能。
- 自動ウェイクアップ割り込みをトリガする周期的なフラグを生成する自動ウェイクアップユニット。
- リファレンスクロック検出: より正確な秒のクロックソース (50 または 60 Hz) の使用で、カレンダーの精度を向上。
- サブセカンドシフト機能を使用する外部クロックとの正確な同期。
- デジタル較正回路 (周期的なカウンタ修正): 数秒の較正時間範囲で得られる 0.95 ppm の精度。
- イベントを保存するタイムスタンプ機能。
- 設定可能なフィルタおよび内部ブルアップのあるタンパ検出イベント。
- マスク可能な割り込み／イベント:
 - アラーム A
 - ウェイクアップ割り込み
 - タイムスタンプ
 - タンパ検出
- バックアップレジスタ。

24.3 RTC の実装

表 71. STM32F0xx RTC implementation⁽¹⁾

RTC の機能	STM32F03x STM32F04x STM32F05x	STM32F07x STM32F09x
周期的ウェイクアップタイマ	-	X
RTC_TAMP1	X	X
RTC_TAMP2	X	X
RTC_TAMP3	-	X
アラーム A	X	X

1. X : サポートされています。

24.4 RTC の機能説明

24.4.1 RTC ブロック図

図 207. STM32F03x、STM32F04x、および STM32F05x デバイスの RTC ブロック図

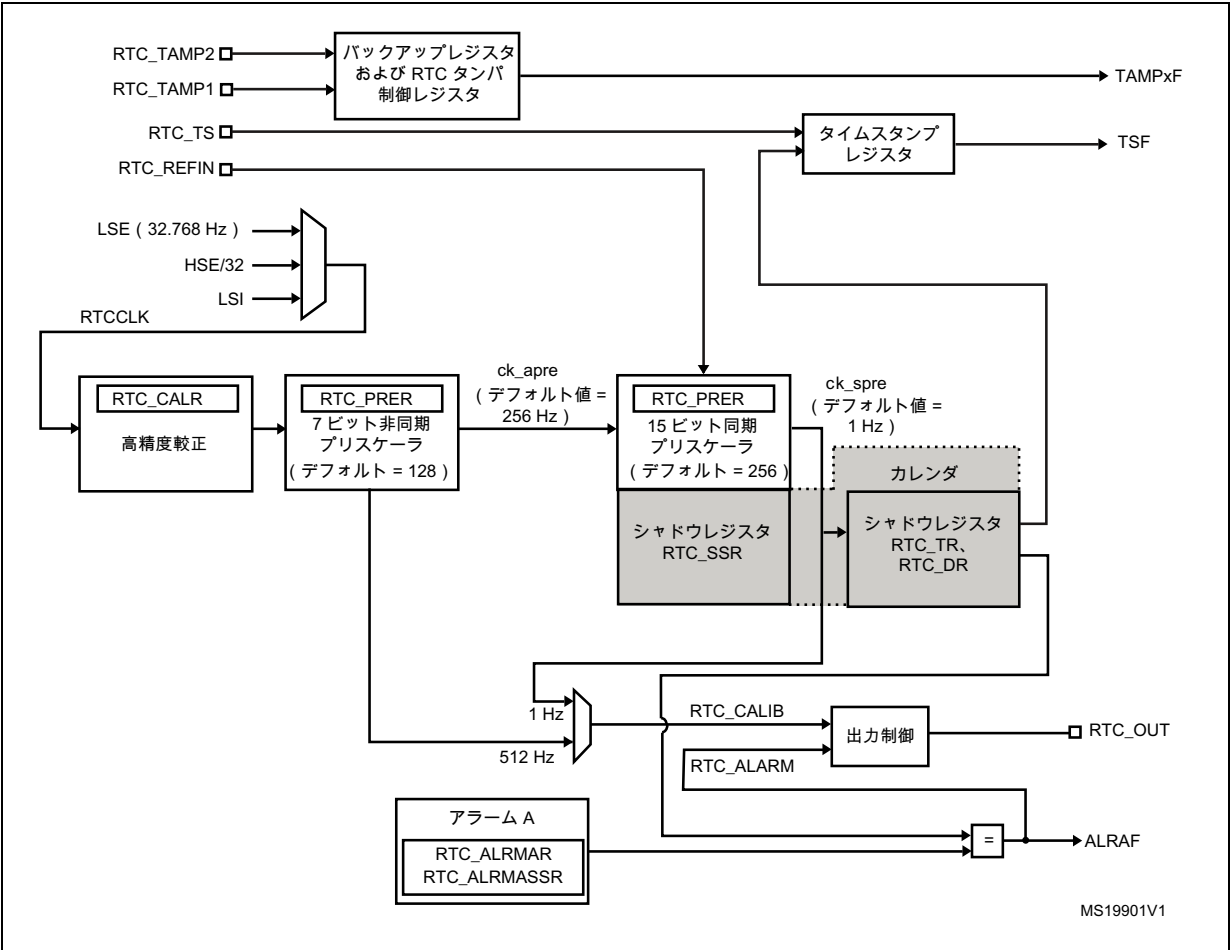
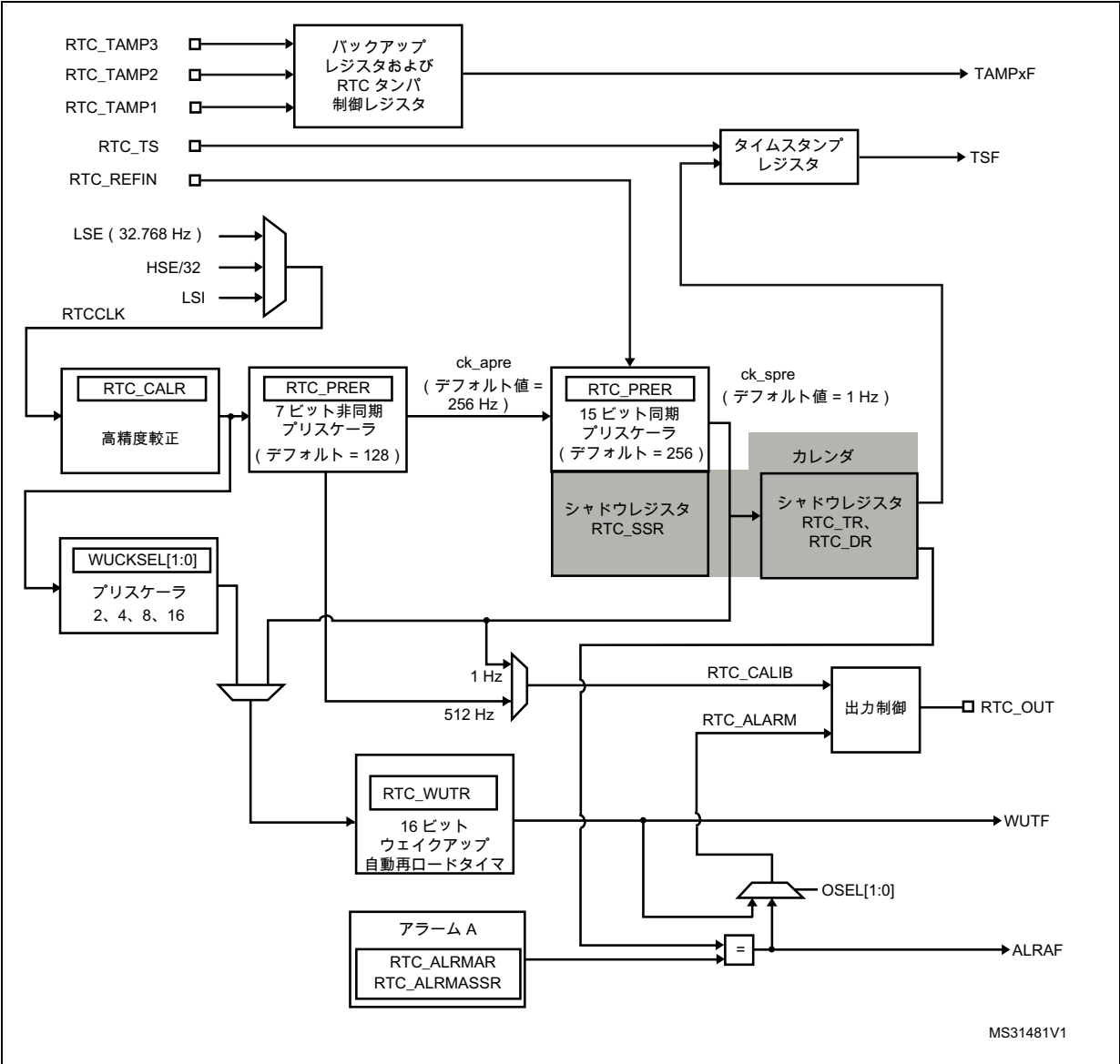


図 208. STM32F07x および STM32F09x デバイスの RTC ブロック図



RTC には次の要素が含まれます。

- 1つのアラーム
- 最大3つのタンパイイベント
- 532ビットバックアップレジスタ
 - バックアップレジスタ (RTC_BKPxR) は、V_{DD} 電源が遮断された場合に VBAT によって電源が供給される、RTC ドメインに搭載されています。
- オルタネート機能出力：次の2つの出力のうち1つを選択する RTC-OUTです。
 - RTC_CALIB：512 Hz または 1Hz のクロック出力 (LSE 周波数 32.768 kHz の場合)。この出力は、RTC_CR レジスタの COE ビットをセットして有効にします。
 - RTC_ALARM：アラーム A。この出力は、RTC_CR レジスタの OSEL[1:0] ビットを設定して選択します。
- オルタネート機能入力：
 - RTC_TS：タイムスタンプイベント
 - RTC_TAMP1：タンパ1イベント検出
 - RTC_TAMP2：タンパ2イベント検出
 - RTC_TAMP3：タンパ3イベント検出
 - RTC_REFIN：50 または 60 Hz のリファレンスクロック入力

24.4.2 RTC によって制御される GPIO

RTC_OUT、RTC_TS、および RTC_TAMP1 は、同一ピン (PC13) 上に配置されます。

RTC_ALARM 出力の選択は次のように RTC_TAFCR レジスタを通して行います。PC13VALUE ビットを使用して、RTC_ALARM 出力をプッシュプルモードとオープンドレインモードのどちらで設定するかを選択します。

PC13 が RTC オルタネート機能として使用されない場合、RTC_TAFCR の PC13MODE ビットを設定することで、出力を強制的にプッシュプルモードに設定できます。これにより、PC13VALUE ビットから出力データ値が与えられます。この場合、PC13 の出力のプッシュプル状態とデータ値は STANDBY モードでも維持されます。

出力形式は [表 72](#) に示す優先順位に従います。

PC14 と PC15 が LSE オシレータとして使用されない場合、RTC_TAFCR レジスタの PC14MODE ビットと PC15MODE ビットをそれぞれ設定することで、出力を強制的にプッシュプルモードに設定できます。これにより、PC14VALUE と PC15VALUE から出力データ値が与えられます。この場合、PC14 および PC15 の出力のプッシュプル状態とデータ値は STANDBY モードでも維持されます。

出力形式は [表 73](#) および [表 74](#) に示す優先順位に従います。

表 72. RTC ピン PC13 の設定⁽¹⁾

ピンの設定と機能	RTC_ALARM 出力有効	RTC_CALIB 出力有効	RTC_TAMP1 入力有効	RTC_TS 入力有効	PC13MODE ビット	PC13VALUE ビット
RTC_ALARM 出力 OD	1	無視	無視	無視	無視	0
RTC_ALARM 出力 PP	1	無視	無視	無視	無視	1
RTC_CALIB 出力 PP	0	1	無視	無視	無視	無視



表 72. RTC ピン PC13 の設定⁽¹⁾ (続き)

ピンの設定と機能	RTC_ALARM 出力有効	RTC_CALIB 出力有効	RTC_TAMP1 入力有効	RTC_TS 入力有効	PC13MODE ビット	PC13VALUE ビット
RTC_TAMP1 入力 フローティング	0	0	1	0	無視	無視
RTC_TS および RTC_TAMP1 入力 フローティング	0	0	1	1	無視	無視
RTC_TS 入力 フローティング	0	0	0	1	無視	無視
出力 PP に固定	0	0	0	0	1	PC13 出力データ値
ウェイクアップ ピンまたは標準 GPIO	0	0	0	0	0	無視

1. OD : オープンドレイン、PP : プッシュプル

表 73. LSE ピン PC14 の設定 ⁽¹⁾

ピンの設定と機能	RCC_BDCR レジスタの LSEON ビット	RCC_BDCR レジスタの LSEBYP ビット	PC14MODE ビット	PC14VALUE ビット
LSE オシレータ	1	0	無視	無視
LSE バイパス	1	1	無視	無視
出力 PP に固定	0	無視	1	PC14 出力データ値
標準 GPIO	0	無視	0	無視

1. OD : オープンドレイン、PP : プッシュプル

表 74. LSE ピン PC15 の設定 ⁽¹⁾

ピンの設定と機能	RCC_BDCR レジスタの LSEON ビット	RCC_BDCR レジスタの LSEBYP ビット	PC15MODE ビット	PC15VALUE ビット
LSE オシレータ	1	0	無視	無視
出力 PP に固定	1	1	1	PC15 出力データ値
	0	無視		
標準 GPIO	0	無視	0	無視

1. OD : オープンドレイン、PP : プッシュプル

24.4.3 クロックとプリスケアラ

RTC クロックソース (RTCCLK) は、LSE クロック、LSI オシレータクロック、HSE クロックのうちから、クロックコントローラを介して選択されます。RTC クロックソースの設定に関する詳細は、[セクション 7：リセットおよびクロック制御 \(RCC\)](#) を参照してください。

プログラム可能なプリスケアラステージで、カレンダーの更新に使用する 1 Hz のクロックを生成します。消費電力を最少に抑えるため、プリスケアラは以下に示す 2 つのプログラム可能なプリスケアラに分割されます ([図 207：STM32F03x、STM32F04x、および STM32F05x デバイスの RTC ブロック図](#) および [図 208：STM32F07x および STM32F09x デバイスの RTC ブロック図](#) を参照)。

- RTC_PRER レジスタの PREDIV_A ビットで設定される 7 ビットの非同期プリスケアラ
- RTC_PRER レジスタの PREDIV_S ビットで設定される 15 ビットの同期プリスケアラ

注： *両方のプリスケアラを使用する場合は、非同期プリスケアラを高い値に設定して消費を最低限に抑えることをお勧めします。*

LSE 周波数 32.768 kHz で 1 Hz (ck_spre) の内部クロック周波数を得るため、非同期プリスケアラ分周比は 128、同期プリスケアラの分周比は 256 に設定されます。

最低分周比は 1、最大分周比は 2^{22} です。

これは、約 4 MHz の最大入力周波数に相当します。

f_{ck_apre} は、次の式で与えられます。

$$f_{CK\_APRE} = \frac{f_{RTCCLK}}{PREDIV\_A + 1}$$

ck_apre クロックは、サブセカンドダウンカウンタであるバイナリ RTC_SSR にクロックを供給するために使用されます。値がゼロになると、RTC_SSR は、PREDIV_S の内容で再ロードされます。

f_{ck_apre} は、次の式で与えられます。

$$f_{CK\_SPRE} = \frac{f_{RTCCLK}}{(PREDIV\_S + 1) \times (PREDIV\_A + 1)}$$

ck_spre クロックは、カレンダーの更新に、または 16 ビットウェイクアップ自動再ロードタイマのタイムベースとして使用できます。短いタイムアウト期間を得るため、16 ビットウェイクアップ自動再ロードタイマを、プログラム可能な 4 ビット非同期プリスケアラで分周した RTCCLK で動作させることもできます (詳細は [セクション 24.4.6：周期的自動ウェイクアップ](#) を参照)。

24.4.4 リアルタイムクロックとカレンダー

RTC カレンダーの時刻および日付レジスタには、PCLK (APB クロック) と同期するシャドウレジスタからアクセスします。同期持続の待ち時間を避けるため、これらのレジスタに直接アクセスすることもできます。

- サブセカンド用 RTC_SSR
- 時刻用 RTC_TR
- 日付用 RTC_DR

RTCCLK 2 サイクルごとに現在のカレンダー値がシャドウレジスタにコピーされ、RTC_ISR レジスタの RSF ビットがセットされます ([セクション 24.7.4：RTC 初期化とステータスレジスタ \(RTC_ISR\)](#) を参照)。STOP モードおよび STANDBY モードでは、コピーは行われません。これらのモードが終了すると、RTCCLK 2 サイクル以内にシャドウレジスタが更新されます。



アプリケーションが、カレンダーレジスタを読み出す際、実際にはシャドウレジスタの内容にアクセスします。RTC_CR レジスタの BYPSHAD 制御ビットをセットすることにより、カレンダーレジスタに直接アクセスできます。デフォルトでは、このビットはクリアされており、ユーザはシャドウレジスタにアクセスします。

RTC_SSR、RTC_TR または RTC_DR レジスタを BYPSHAD = 0 の状態で読み出す際は、APB クロックの周波数 (f_{APB}) は、RTC クロック (f_{RTCCLK}) の周波数の 7 倍以上でなければなりません。

シャドウレジスタは、システムリセットによってリセットされます。

24.4.5 プログラム可能なアラーム

RTC ユニットは、以下に示すプログラム可能なアラーム、アラーム A。

プログラム可能なアラーム機能は、RTC_CR レジスタの ALRAE ビットを通じて有効にします。ALRAF は、カレンダーのサブセカンド、秒、分、時、日または曜日がそれぞれアラームレジスタ RTC_ALRMSSR および RTC_ALRMAR にプログラムされている値と一致する場合は 1 にセットされます。各カレンダー項目は、RTC_ALRMAR レジスタの MSKx ビットおよび RTC_ALRMSSR レジスタの MASKSSx ビットで個別に選択できます。アラームの割り込みは、RTC_CR レジスタの ALRAIE ビットを通じて有効にします。

注意： *秒の項目が選択されている (RTC_ALRMAR で MSK1 ビットがリセットされている) 場合、正しい動作を保証するため、RTC_PRER レジスタでセットされる同期プリスケアラの分周比は 3 以上でなければなりません。*

アラーム A (RTC_CR レジスタの OSEL[1:0] ビットで有効になっている場合) は、RTC_ALARM 出力に送ることができます。RTC_ALARM 出力の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

24.4.6 周期的自動ウェイクアップ

周期的ウェイクアップフラグは、16 ビットのプログラム可能な自動再ロードダウンカウンタによって生成されます。ウェイクアップタイマの範囲は 17 ビットまで拡張できます。

ウェイクアップ機能は、RTC_CR レジスタの WUTE ビットを通じて有効にします。

ウェイクアップタイマクロック入力には、次のものが使用できます。

- 2、4、8、または 16 分周した RTC クロック (RTCCLK)
RTCCLK が LSE (32.768kHz) である場合、最小分解能 61 μ s で、ウェイクアップ割り込み周期を 122 μ s から 32 s の範囲で設定できます。
- ck_spre (通常は 1 Hz の内部クロック)
ck_spre 周波数が 1 Hz の場合、1 秒の分解能でウェイクアップ時間を 1 秒 からおよそ 36 時間までの範囲で設定できます。このプログラム可能な広い時間範囲は、2 つの部分に分かれます。
 - WUCKSEL[2:1] = 10 の場合は 1 秒から 18 時間、
 - WUCKSEL[2:1] = 11 の場合は約 18 時間から 36 時間です。後者の場合、16 ビットカウンタの現在値に 216 が加算されます。初期化シーケンスが完了すると ([ウェイクアップタイマのプログラミング \(577 ページ\)](#) を参照)、タイマがカウントダウンを開始します。ウェイクアップ機能が有効な場合、低電力モードでもカウントダウンはアクティブのままとなります。さらに、カウンタがゼロに到達すると、RTC_ISR レジスタの WUTF フラグがセットされ、ウェイクアップカウンタが再ロード値 (RTC_WUTR レジスタ値) で自動的に再ロードされます。

その後、WUTF フラグはソフトウェアでクリアする必要があります。

RTC_CR2 レジスタの WUTIE ビットをセットして周期的ウェイクアップ割り込みを有効にすると、デバイスは低電力モードを終了できます。

周期的なウェイクアップフラグは、RTC_CR レジスタの OSEL[0:1] ビットを通じて有効になっている場合に限り、RTC_ALARM 出力に送ることができます。RTC_ALARM 出力の極性は、RTC_CR レジスタの POL ビットを通じて設定できます。

低電力モード (SLEEP、STOP、STANDBY) と同様に、システムリセットもウェイクアップタイムには影響しません。

24.4.7 RTC の初期化と設定

RTC レジスタアクセス

RTC レジスタは、32 ビットのレジスタです。APB インターフェイスは、RTC レジスタアクセスに 2 ウェイトステートを挿入します。ただし、BYP SHAD = 0 のときのカレンダーシャドウレジスタへの読み出しアクセスは除きます。

RTC レジスタ書き込み保護

システムリセット後、RTC レジスタは、PWR_CR レジスタの DBP ビットをクリアすることによって、意図しない書き込みアクセスから保護されます (電源制御のセクションを参照)。RTC レジスタ書き込みアクセスを可能にするには、DBP ビットをセットする必要があります。

RTC ドメインリセット後、すべての RTC レジスタは書き込み保護されます。RTC レジスタへの書き込みは、書き込み保護レジスタ RTC_WPR にキーを書き込むことにより有効になります。

RTC_TAFCR、RTC_BKPxR、および RTC_ISR[13:8] を除くすべての RTC レジスタの書き込み保護を解除するには、次のステップが必要です。

1. RTC_WPR レジスタに“0xCA”を書き込みます。
2. RTC_WPR レジスタに“0x53”を書き込みます。

誤ったキーを書き込むと、書き込み保護が再度アクティブになります。

保護メカニズムは、システムリセットの影響を受けません。

カレンダーの初期化と設定

時間形式やプリスケアラ設定を含むカレンダー時刻と日付の初期値をプログラムするには、次のシーケンスが必要です。

1. RTC_ISR レジスタで INIT ビットを 1 にセットして、初期化モードに入ります。このモードでは、カレンダーカウンタが停止し、その値を更新することができます。
2. RTC_ISR レジスタの INITF ビットをポーリングします。INITF が 1 にセットされると、初期化フェーズモードに入ります。これには RTCCLK クロック約 2 サイクルを必要とします（クロック同期のため）。
3. カレンダーカウンタのための 1 Hz クロックを生成するには、RTC_PRER レジスタで両方のプリスケアラ分周比をプログラムします。
4. シャドウレジスタ (RTC_TR および RTC_DR) に時刻と日付の初期値をロードし、RTC_CR レジスタの FMT ビットを介して時間形式（12 時間または 24 時間）を設定します。
5. INIT ビットをクリアして初期化モードを終了します。その後、カレンダーカウンタの実際の値が自動的にロードされ、4 RTCCLK クロックサイクル後にカウントが再開します。

初期化シーケンスが完了すると、カレンダーがカウントを開始します。

注： システムリセット後、アプリケーションは RTC_ISR レジスタの INITS フラグを読み出し、カレンダーが初期化されたか否かを確認できるようになります。このフラグが 0 であれば、カレンダーの年の項目が RTC ドメインリセットデフォルト値 (0x00) にセットされているため、初期化されていません。初期化後にカレンダーを読み出すには、まずソフトウェアで RTC_ISR レジスタの RSF フラグがセットされていることを確認する必要があります。

コード例については、「付録」セクション [A.16.1: RTC カレンダー設定のコード例](#) を参照してください。

サマータイム

サマータイム管理は、RTC_CR レジスタの SUB1H ビット、ADD1H ビット、BKP ビットを介して行われます。

SUB1H または ADD1H を使用すると、ソフトウェアは初期化手順を踏まずに 1 度の操作で、カレンダーから 1 時間引いたり足したりすることができます。

さらに、ソフトウェアは BKP ビットを使用してこの操作を記憶することができます。

アラームのプログラミング

プログラム可能なアラームをプログラムまたは更新するには、同様な手順を踏む必要があります。

1. RTC_CR の ALRAE をクリアしてアラーム A を無効にします。
2. アラーム A レジスタ (RTC_ALRMSSR/RTC_ALRMAR) をプログラムします。
3. RTC_CR レジスタで ALRAE をセットしてアラーム A を再び有効にします。

注： RTC_CR レジスタの各変更は、クロック同期のため RTCCLK クロック約 2 サイクル後に有効になりません。

コード例については、「付録」セクション [A.16.2: RTC アラーム設定のコード例](#) を参照してください。

ウェイクアップタイマのプログラミング

ウェイクアップタイマ自動再ロード値 (RTC_WUTR の WUT[15:0]) の設定または変更には、次の手順が必要です。

1. RTC_CR の WUTE をクリアしてウェイクアップタイマを無効にします。
2. RTC_ISR の WUTWF がセットされ、ウェイクアップ自動再ロードカウンタおよび WUCKSEL[2:0] ビットへのアクセスが許可されていることが確認されるまで WUTWF をポーリングします。これには RTCCLK クロック約 2 サイクルを必要とします (クロック同期のため)。
3. ウェイクアップ自動再ロード値 WUT[15:0] およびウェイクアップクロック選択 (RTC_CR の WUCKSEL[2:0] ビット) をプログラムします。RTC_CR で WUTE をセットしてタイマを再び有効にします。ウェイクアップタイマがカウントダウンを再開します。WUTWF ビットは、クロックの同期化により、WUTE クリア後、2 RTCCLK クロックサイクルまでクリアされます。

コード例については、「付録」セクション [A.16.3 : RTC WUT 設定のコード例](#) を参照してください。

24.4.8 カレンダの読み出し

RTC_CR レジスタの BYPSHAD 制御ビットがクリアされている場合

RTC カレンダレジスタ (RTC_SSR、RTC_TR、および RTC_DR) を正しく読み出すには、APB クロック周波数 (f_{PCLK}) が RTC クロック周波数 (f_{RTCCLK}) の 7 倍以上でなければなりません。これにより、同期メカニズムの安全な動作が保証されます。

APB クロック周波数が RTC クロック周波数の 7 倍未満である場合、ソフトウェアによってカレンダー時間と日付レジスタを 2 回読み出す必要があります。RTC_TR の 2 回目の読み出しが 1 回目の読み出しと同じ結果であれば、データが正しいことが保証されます。同じでない場合は、3 回目の読み出しアクセスを行う必要があります。どの場合も、APB クロック周波数は必ず RTC クロック周波数以上でなければなりません。

RTC_ISR レジスタの RSF ビットは、カレンダーレジスタが RTC_SSR、RTC_TR、および RTC_DR シャドウレジスタにコピーされるたびにセットされます。コピーは、2 RTCCLK サイクルごとに行われます。3 つの値における一貫性を保証するため、RTC_SSR または RTC_TR のどちらかを読み出すと、高次カレンダーシャドウレジスタの値は RTC_DR が読み出されるまでロックされます。ソフトウェアが 2 RTCCLK サイクル未満の間隔でカレンダーの読み出しアクセスを行う場合、最初のカレンダー読み出し後に RSF をソフトウェアでクリアする必要があり、その後ソフトウェアは、RSF ビットがセットされるまで待つてから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再読み出す必要があります。

低電力モード (STOP または STANDBY) からのウェイクアップ後は、RSF をソフトウェアでクリアする必要があります。その後、ソフトウェアは、いまだ RSF がセットされるまで待つてから、RTC_SSR、RTC_TR、および RTC_DR レジスタを再読み出す必要があります。

RSF ビットは、ウェイクアップ後にクリアする必要がありますが、低電力モードに入る前には、その必要はありません。

システムリセット後、ソフトウェアは RSF がセットされるまで待つてから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。実際、システムリセットがかかると、シャドウレジスタはデフォルト値にリセットされます。

初期化 ([カレンダーの初期化と設定 \(576 ページ\)](#)) を参照) 後、ソフトウェアは RSF がセットされるまで待つてから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。

同期 ([セクション 24.4.10 : RTC の同期](#)) を参照) 後、ソフトウェアは RSF がセットされるまで待つてから、RTC_SSR、RTC_TR、および RTC_DR レジスタを読み出す必要があります。

コード例については、「付録」セクション [A.16.4 : RTC 読み出しカレンダーのコード例](#) を参照してください。

RTC_CR レジスタ (バイパスシャドウレジスタ) の BYPSHAD 制御ビットがセットされている場合

カレンダーレジスタを読み出すと、カレンダーカウンタの値が直接与えられるため、RSF ビットがセットされるのを待つ必要がありません。シャドウレジスタは低電力モード (STOP または STANDBY) では更新されないため、低電力モード終了後に特にこのような読み出しが有用です。

BYPSHAD ビットが 1 にセットされている場合、レジスタへの 2 回の読み出しアクセス間で RTCCLK エッジが発生した場合は、さまざまなレジスタ間で互いに不整合が起きる場合があります。さらに、読み出し操作中に RTCCLK エッジが発生した場合、レジスタの 1 つが不正な値となる場合があります。ソフトウェアはすべてのレジスタを 2 回読み出し、その結果を比較してデータに整合性があり正しいことを確認する必要があります。その代わりに、ソフトウェアはカレンダーレジスタの最下位の数値を 2 回比較するだけで構いません。

注: *BYPSHAD = 1 の間、カレンダーレジスタの読み出し命令が完了するには 1 APB サイクルだけ余計に必要となります。*

24.4.9 RTC のリセット

カレンダーシャドウレジスタ (RTC_SSR、RTC_TR、RTC_DR) および RTC ステータスレジスタ (RTC_ISR) の一部のビットは、利用可能なすべてのシステムリセットリソースによってデフォルト値にリセットされます。

逆に、次のレジスタは RTC ドメインリセットによってそれぞれのデフォルト値にリセットされ、システムリセットの影響は受けません。RTC の現在のカレンダーレジスタ、RTC 制御レジスタ (RTC_CR)、プリスケアラレジスタ (RTC_PRER)、RTC 較正レジスタ (RTC_CALR)、RTC シフトレジスタ (RTC_SHIFTR)、RTC タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、および RTC_TSDR)、RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)、RTC バックアップレジスタ (RTC_BKPxR)、ウェイクアップタイマレジスタ (RTC_WUTR)、およびアラーム A レジスタ (RTC_ALRMASSR/RTC_ALRMAR)。

さらに、RTC はリセットソースが RTC ドメインリセットと異なる場合は、システムリセット中でも動作し続けます。RTC ドメインリセットが発生すると、RTC は停止し、すべての RTC レジスタがリセット値にセットされます。

24.4.10 RTC の同期

RTC は、高精度でリモートクロックと同期できます。サブセカンド項目 (RTC_SSR または RTC_TSSSR) を読み出すと、リモートクロックによって維持されている時刻と RTC 間の正確なオフセットが計算できます。その後、RTC_SHIFTR を使用してほんの一瞬クロックを「シフト」することによって RTC を調整し、このオフセットを取り除くことができます。

RTC_SSR には、同期プリスケアラのカウンタの値が入っています。これにより、RTC によって維持されている正確な時刻を $1/(\text{PREDIV}_S + 1)$ 秒の分解能まで計算することができます。その結果、同期プリスケアラ値 ($\text{PREDIV}_S[14:0]$) を増加させることにより分解能を改善できます。許可されている最大分解能 (32768 Hz クロックで 30.52 μs) は、 PREDIV_S を 0x7FFF にセットすることにより得られます。

ただし、 PREDIV_S を増加させるということは、同期プリスケアラの出力を 1 Hz に維持するため PREDIV_A を減らす必要があることを意味します。このように、非同期プリスケアラの出力周波数が増加すると、RTC の動的消費電力が増加する場合があります。

RTC は、RTC シフト制御レジスタ (RTC_SHIFTR) を使って微調整できます。RTC_SHIFTR に書き込むことにより、 $1/(\text{PREDIV}_S + 1)$ 秒の分解能で、クロックを最大 1 秒だけシフト (遅れ/進み) させることができます。このシフト操作の本質は、同期プリスケアラのカウンタ $\text{SS}[15:0]$ に $\text{SUBFS}[14:0]$ 値を加算することであり、この操作はクロックを遅らせることになります。同時に

ADD1S ビットがセットされた場合、1 秒追加すると同時に秒の小数部を差し引くことになるため、クロックを進めることになります。

注意： シフト操作を始める前に、ユーザは $SS[15] = 0$ であることを確認し、オーバーフローが発生しないようにする必要があります。

RTC_SHIFTR レジスタへの書き込みによってシフト操作が始まるとすぐに、シフト操作が保留中であることを示す SHPF フラグがハードウェアによってセットされます。このビットは、シフト操作が完了するとすぐに、ハードウェアによってクリアされます。

注意： この同期機能はリファレンスクロック検出機能とは両立できません。具体的には、 $REFCKON = 1$ のときにファームウェアから RTC_SHIFTR への書き込みはできません。

24.4.11 RTC リファレンスクロック検出

RTC カレンダの更新は、リファレンスクロックである RTC_REFIN に同期させることができます。通常は商用電源 (50 または 60 Hz) です。RTC_REFIN リファレンスクロックには、32.768 kHz LSE クロックより高い精度が必要です。RTC_REFIN 検出が有効になると (RTC_CR の REFCKON ビットが 1 にセット)、カレンダは引き続き LSE クロックによって駆動されますが、RTC_REFIN を使用してカレンダ更新周波数 (1 Hz) の不正確さを補正します。

各 1 Hz クロックエッジは、一番近い RTC_REFIN クロックエッジ (所与の時間枠内に見つかった場合) と比較されます。ほとんどの場合、2 つのクロックエッジは正しく整列しています。LSE クロックが不正確なために 1 Hz のクロックがずれた場合、RTC は 1 Hz のクロックを少しシフトさせ、その後の 1 Hz のクロックエッジが整列するようにします。このメカニズムのおかげで、カレンダはリファレンスクロックと同様に正確になります。

RTC は、32.768 kHz クォーツから生成される 256 Hz クロック (ck_apre) を使用して、リファレンスクロックソースがあるかどうかを検出します。検出は各カレンダ更新 (1 秒ごと) 程度の時間枠で行われます。最初のリファレンスクロックエッジを検出する際、この時間枠は ck_apre 7 周期に等しくなります。その後のカレンダ更新では、ck_apre 3 周期より短い時間枠が使用されます。

リファレンスクロックがこの時間枠内で検出されるたびに、ck_apre クロックを出力する非同期プリスケアラは強制的に再ロードされます。プリスケアラは同時に再ロードされるので、リファレンスクロックおよび 1 Hz のクロックが整列するタイミングには影響しません。クロックが整列していない場合、後の 1 Hz クロックエッジは、リファレンスクロックと整列するように再ロードによって少しシフトされます。

リファレンスクロックが停止した (ck_apre 3 周期の枠内でリファレンスクロックエッジが発生しない) 場合、カレンダは LSE クロックのみを基準にして更新が継続されます。その後 RTC は ck_spre エッジを中心として ck_apre 7 周期という広い検出時間枠でリファレンスクロックを待ちます。

基準クロック検出を有効にした場合、PREDIV_A および PREDIV_S を以下に示すそれぞれのデフォルト値にセットする必要があります。

- PREDIV_A = 0x007F
- PREDIV_S = 0x00FF

注： RTC_REFIN クロック検出は、STANDBY モードでは利用できません。



24.4.12 RTC の高精度デジタル校正

RTC 周波数の精度は、-487.1~+488.5 ppm の範囲で、分解能約 0.954 ppm でデジタル的に校正できます。周波数の修正は、一連の微調整（個々の RTCCLK パルスの追加や削除）によって行われます。このような調整は、短い期間で観測された場合でも RTC が十分に校正されるように、かなり広範に分散して行われます。

この高精度デジタル校正は、入力周波数が 32768 Hz の場合、RTCCLK 約 2^{20} パルスのサイクルまたは 32 秒の間で行われます。このサイクルは、RTCCLK によって駆動される 20 ビットカウンタ、cal_cnt[19:0] によって維持されます。

高精度校正レジスタ (RTC_CALR) によって、32 秒サイクル中にマスクされる RTCCLK クロックサイクル数を指定します。

- CALM[0] ビットを 1 にセットすると、32 秒サイクルの中でちょうど 1 パルスがマスクされます。
- CALM[1] ビットを 1 にセットすると、さらに 2 サイクルがマスクされます。
- CALM[2] ビットを 1 にセットすると、さらに 4 サイクルがマスクされます。
- CALM[8] ビットを 1 にセットするまで続けると、256 クロックがマスクされます。

注： CALM[8:0] (RTC_CALR) によって、32 秒サイクル中にマスクされる RTCCLK パルス数を指定します。CALM[0] ビットを 1 にセットすると、cal_cnt[19:0] = 0x80000 になった時点で、32 秒サイクル中でちょうど 1 パルスがマスクされます。CALM[1] = 1 では、さらに 2 サイクルがマスクされ (cal_cnt = 0x40000 および 0xC0000)、CALM[2] = 1 では、さらに 4 サイクルがマスクされ (cal_cnt = 0x20000/0x60000/0xA0000/0xE0000)、CALM[8] = 1 まで続けると、256 クロックがマスクされず (cal_cnt = 0xXX800)。

CALM では、細かい分解能で RTC 周波数を最大 487.1 ppm 負の方向に調整することができ、CALP ビットでは周波数を 488.5 ppm 正の方向に調整することができます。CALP を 1 にセットすることにより、実質上は、RTCCLK 2^{11} サイクルごとに、RTCCLK パルスが 1 パルス追加で挿入されます。すなわち、32 秒サイクルごとに 512 クロックが追加されることとなります。

CALM を CALP と合わせて使用すると、32 秒サイクルの間に RTCCLK -511 から +512 サイクルまでのオフセットが追加でき、これは約 0.954 ppm の分解能で校正範囲 -487.1~+488.5 ppm に換算されます。

有効校正周波数 (FCAL) を入力周波数 (FRTCCLK) に対して求める計算式は次のとおりです。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (\text{CALP} \times 512 - \text{CALM}) / (2^{20} + \text{CALM} - \text{CALP} \times 512)]$$

PREDIV_A < 3 の場合の校正

非同期プリスケアラ値 (RTC_PRER レジスタの PREDIV_A ビット) が 3 未満の場合、CALP ビットを 1 にセットすることはできません。CALP がすでに 1 にセットされていて、PREDIV_A ビットが 3 未満の値にセットされた場合、CALP の設定値は無視され、CALP が 0 に設定された場合と同じように校正されます。

PREDIV_A が 3 未満の状態では校正を実施するには、各秒のカウントが 8 RTCCLK クロックサイクル分早められるように同期プリスケアラ値 (PREDIV_S) を小さくする必要があります。これは 32 秒毎に 256 クロックサイクル追加することに相当します。結果として、CALM ビットのみを使用して 32 秒サイクルの間に 255~256 クロックパルス (243.3~244.1 ppm の校正範囲に相当)を追加することができます。

公称 RTCCLK 周波数が 32768 Hz で、PREDIV_A が 1 (分周比 2) の場合、PREDIV_S を 16383 ではなく 16379 (4 少ない) にセットする必要があります。また、PREDIV_A が 0 の場合、PREDIV_S を 32767 ではなく 32759 (8 少ない) にセットする必要がありますので注意して下さい。

PREDIV_S をこのように減少させた場合、校正された入力クロックの有効周波数の式は次のようになります。

$$F_{\text{CAL}} = F_{\text{RTCCLK}} \times [1 + (256 - \text{CALM}) / (2^{20} + \text{CALM} - 256)]$$

この場合、RTCCLK が正確に 32768.00 Hz であれば、CALM[7:0] が 0x100 (CALM 設定範囲の中間値) と等しくなるのが正しい設定です。

RTC 較正值の確認

RTC の精度は、RTCCLK の正確な周波数を測定し、正しい CALM 値および CALP 値を計算することにより保証されます。オプションの 1 Hz 出力が搭載されており、アプリケーションによって RTC 精度の測定と確認を行うことができます。

ある時間間隔で RTC の周波数を精密に測定すると、デジタル校正サイクルを測定周期とどのように合わせているかにより、測定期間中に最大 2 RTCCLK クロックサイクルの測定誤差が生じます。

ただし、この測定誤差は、測定周期が校正サイクル周期と同じ長さであれば排除できます。この場合、観測される唯一の誤差はデジタル校正の分解能による誤差となります。

- デフォルトでは、校正サイクル周期は 32 秒です。

このモードを使用して正確に 32 秒で 1 Hz 出力の精度を測定すると、その精度は 0.477 ppm (校正分解能の制限により 32 秒で 0.5 RTCCLK サイクル) 以内となることが保証されます。

- RTC_CALR レジスタの CALW16 ビットを 1 にセットして、校正サイクル周期を強制的に 16 秒にすることができます。

この場合、RTC 精度は最大誤差 0.954 ppm (16 秒で 0.5 RTCCLK サイクル) で 16 秒間で測定できます。ただし、校正分解能が下がるため、長期的な RTC 精度もまた 0.954 ppm に下がります。CALW16 が 1 にセットされると、CALM[0] ビットは 0 のままとなります。

- RTC_CALR レジスタの CALW8 ビットを 1 にセットして、校正サイクル周期を強制的に 8 秒にすることができます。

この場合、RTC 精度は最大誤差 1.907 ppm (8 秒で 0.5 RTCCLK サイクル) で 8 秒で測定できます。長期的な RTC 精度もまた 1.907 ppm に下がります。CALW8 が 1 にセットされると、CALM[1:0] ビットは 00 のままとなります。

動作中の再校正

次の処理を実施することにより、RTC_ISR/INITF = 0 の間でも、校正レジスタ (RTC_CALR) を動作中に更新することができます。

1. RTC_ISR/RECALPF (再校正保留フラグ) をポーリングします。
2. このフラグが 0 にセットされている場合は、必要に応じて新しい値を RTC_CALR に書き込みます。すると、RECALPF が自動的に 1 にセットされます。
3. RTC_CALR への書き込み動作後 ck_apre 3 サイクル以内に、新しい校正設定が有効になります。

コード例については、「付録」セクション [A.16.5 : RTC 校正のコード例](#) を参照してください。

24.4.13 タイムスタンプ機能

タイムスタンプは、RTC_CR レジスタの TSE ビットを 1 にセットすることにより有効になります。

RTC_TS ピンでタイムスタンプイベントが検出されると、タイムスタンプレジスタ (RTC_TSSSR、RTC_TSTR、RTC_TSDR) にカレンダーが保存されます。

また、タイムスタンプイベントが発生すると、RTC_ISR レジスタのタイムスタンプフラグビット (TSF) がセットされます。

RTC_CR レジスタの TSIE ビットをセットすることにより、タイムスタンプイベントが発生したときに割り込みが生成されます。

タイムスタンプフラグ (TSF) がすでにセットされている間に新しいタイムスタンプイベントが検出された場合、タイムスタンプオーバーフローフラグ (TSOVF) がセットされ、タイムスタンプレジスタ (RTC_TSTR および RTC_TSDR) は、その前のイベントの結果を維持します。

注： *同期処理のため、TSF はタイムスタンプイベント発生から ck_apre 2 サイクル後にセットされます。一方、TSOVF のセットに遅延はありません。これは、2 つのタイムスタンプイベントの発生したタイミングが近い場合、TSF がまだ“0”であっても TSOVF が“1”と検出される可能性があることを意味します。よって、TSOVF のポーリングは TSF がセットされた後に実施することをお勧めします。*

注意： *TSF ビットがクリアされたと思われた直後にタイムスタンプイベントが発生した場合、TSF および TSOVF ビットの両方がセットされます。同時に発生するタイムスタンプイベントのマスキングを回避するため、TSF がすでに“1”と読み出されていない限りは、アプリケーションによる“0”クリアの処理を行ってはいけません。*

オプション機能として、タンパイベントによってタイムスタンプイベントを記録することもできます。TAMPTS 制御ビットの詳細については、[セクション 24.7.13 : RTC タイムスタンプサブセカンドレジスタ \(RTC_TSSSR\)](#) を参照してください。

24.4.14 タンパ検出

RTC_TAMPx 入力イベントは、フィルタを使用してエッジ検出またはレベル検出のどちらにも設定できます。

タンパ検出は次の目的で設定することができます。

- RTC バックアップレジスタを消去する。
- STOP モードおよび STANDBY モードからのウェイクアップが可能な割り込みを生成する。

RTC バックアップレジスタ

バックアップレジスタ (RTC_BKPxR) は、システムリセットや STANDBY モードからのウェイクアップではリセットされません。

バックアップレジスタは、タンパ検出イベント発生時にリセットされます ([セクション 24.7.17 : RTC バックアップレジスタ \(RTC_BKPxR\)](#) および [タンパ検出の初期化 \(582 ページ\)](#) を参照。またはフラッシュの読み出し保護がレベル 1 からレベル 0 に変わったとき)。

タンパ検出の初期化

各入力は、RTC_TAFCR レジスタにおいて該当する TAMPxE ビットを 1 にセットすることにより有効にできます。

各 RTC_TAMPx タンパ検出入力、RTC_ISR レジスタの TAMPxF フラグに関連付けられています。

TAMPxF フラグは、ピン上でタンパイイベントが発生した後にアサートされます。その際の遅延時間を以下に示します。

- TAMPFLT = 0x0 以外の場合、ck_apre 3 サイクル（フィルタを使ったレベル検出）
- TAMPTS = 1 の場合、ck_apre 3 サイクル（タンパイイベント時のタイムスタンプ）
- TAMPFLT = 0x0（エッジ検出）および TAMPTS = 0 の場合、遅延なし

TAMPxF がセットされている場合に、この周期中に同一ピンで発生した新たなタンパイイベントを検出することはできません。

RTC_TAFCR レジスタの TAMPIE ビットをセットすることにより、タンパ検出イベント発生時に割り込みが生成されます。

タンパイイベント時のタイムスタンプ

TAMPTS を“1”にセットすると、すべてのタンパイイベントがタイムスタンプイベントを発生させるようになります。この場合、通常のタイムスタンプイベント発生時と同様に TSF ビットまたは TSOVF ビットが RTC_ISR でセットされます。TSF または TSOVF がセットされるのと同時に、影響を受けるタンパフラグレジスタ、TAMPxF がセットされます。

タンパ入力でのエッジ検出

TAMPFLT ビットが“00”の場合、該当する TAMPxTRG ビットに応じて立ち上がりエッジまたは立ち下がりエッジが観測されると、RTC_TAMPx ピンがタンパ検出イベントを生成します。エッジ検出を選択すると、RTC_TAMPx 入力の内部プルアップ抵抗が無効になります。

注意： *タンパ検出イベントを確実に検出するため、RTC_TAMPx ピンが有効になる前にタンパ検出イベントが発生した場合でも検出できるよう、エッジ検出に使用される信号は対応する TAMPxE ビットと論理積がとられます。*

- TAMPxTRG = 0 の場合：タンパ検出が有効になる（TAMPxE ビットが 1 にセットされる）前に RTC_TAMPx オルタネート機能がすでにハイのとき、TAMPxE がセットされた後に RTC_TAMPx に立ち上がりエッジ入力がなくとも、RTC_TAMPx 入力が有効になるとすぐにタンパイイベントが検出されます。
- TAMPxTRG = 1 の場合：タンパ検出が有効になる前に RTC_TAMPx オルタネート機能がすでにローのとき、RTC_TAMPx 入力が有効になるとすぐに（TAMPxE がセットされた後に RTC_TAMPx に立ち下がりエッジ入力がなくとも）、タンパイイベントが検出されます。

タンパイイベントが検出されクリアされた後に、バックアップレジスタ（RTC_BKPxR）を再プログラムする場合には、事前に、RTC_TAMPx オルタネート機能を無効にしてから再度有効にする（TAMPxE を 1 にセット）必要があります。これによって、RTC_TAMPx 入力の値がタンパ検出を示している間に、アプリケーションがバックアップレジスタにデータを書き込むのを防ぎます。これは、RTC_TAMPx オルタネート機能の入力でのレベル検出に相当します。

注： *タンパ検出は、V_{DD} 電源がオフのときでも有効です。バックアップレジスタの不必要なリセットを避けるには、RTC_TAMPx オルタネート機能が設定されているピンを外部で適切な信号レベルに接続しておく必要があります。*

RTC_TAMPx 入力でのフィルタを使ったレベル検出

フィルタを使ったレベル検出は、TAMPFLT を 0 以外の値にセットすることにより行われます。タンパ検出イベントは、(TAMPFLT に応じて) 2、4 または 8 回のいずれかの連続したサンプルが TAMPxTRG ビットで指定するレベルで観測されたときに生成されます。

RTC_TAMPx 入力は、TAMPPUDIS が 1 にセットされて無効な状態になっていない限り、その状態がサンプリングされる前に I/O の内部プルアップ抵抗でプリチャージされています。プリチャージの継続時間は TAMPPRCH ビットによって決定され、RTC_TAMPx 入力ピンにおけるより大きな容量を持たせることができます。

タンパ検出の遅延時間と、プルアップによる電力消費との間のトレードオフは、TAMPFREQ を使用してレベル検出のサンプリング周波数を決定することにより、最適化できます。

注： プルアップ抵抗の電気的特性については、データシートを参照してください。

コード例については、「付録」セクション [A.16.6 : RTC タンパおよびタイムスタンプ設定のコード例](#) と [A.16.7 : RTC タンパおよびタイムスタンプのコード例](#) を参照してください。

24.4.15 較正クロック出力

RTC_CR レジスタで COE ビットが 1 にセットされると、RTC_CALIB デバイス出力にリファレンスクロックが供給されます。

RTC_CR レジスタの COSEL ビットがリセットされ、かつ PREDIV_A = 0x7F である場合、RTC_CALIB 周波数は $f_{\text{RTCCLK}}/64$ です。これは 32.768 kHz の RTCCLK 周波数に対する 512 Hz の較正出力に相当します。立ち下がリエッジには軽いジッタがあるため、RTC_CALIB のデューティサイクルは不規則になります。したがって、立ち上がりエッジの使用が推奨されます。

COSEL がセットされ、かつ “PREDIV_S+1” がゼロ以外の 256 の倍数である場合 (すなわち、PREDIV_S[7:0] = 0xFF)、RTC_CALIB 周波数は $f_{\text{RTCCLK}}/(256 * (\text{PREDIV\_A}+1))$ となります。これは、RTCCLK 周波数が 32.768 kHz で、プリスケアラデフォルト値 (PREDIV_A = 0x7F、PREDIV_S = 0xFF) に対する 1 Hz の較正出力に相当します。

注： **RTC_CALIB または RTC_ALARM 出力が選択されると、RTC_OUT ピンは自動的に出力オルタネート機能に設定されます。**

コード例については、「付録」セクション [A.16.8 : RTC クロック出力のコード例](#) を参照してください。

24.4.16 アラーム出力

RTC_CR レジスタの OSEL[1:0] 制御ビットを使用してアラームオルタネート機能出力、RTC_ALARM を有効にし、出力となる機能を選択します。これらの機能は、RTC_ISR レジスタの該当するフラグの内容を反映します。

RTC_CR の POL 制御ビットは、POL が 1 にセットされているときに選択されたフラグビットの逆が出力されるよう、出力の極性を決定します。

アラームオルタネート機能出力

RTC_ALARM ピンは、制御ビット ALARMOUTTYPE (RTC_TAFCR レジスタ) を使用して、出力オーブンドレインまたは出力プッシュプルに設定できます。

注： **RTC_ALARM 出力が有効になると、この設定は、RTC_CALIB (COE ビットは無視され、クリアされたまま) の設定よりも優先されます。**

RTC_CALIB または RTC_ALARM 出力が選択されると、RTC_OUT ピンは自動的に出力オルタネート機能に設定されます。

24.5 RTC 低電力モード

表 75. 低電力モードが RTC に与える影響

モード	説明
SLEEP	影響なし。 RTC 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスは STOP モードから復帰します。
STANDBY	RTC クロックソースが LSE または LSI の場合、RTC はアクティブのままです。RTC アラーム、RTC タンパイイベント、RTC タイムスタンプイベント、RTC ウェイクアップにより、デバイスは STANDBY モードから復帰します。

24.6 RTC 割り込み

すべての RTC 割り込みは、EXTI コントローラに接続されています。[セクション 12.2: 拡張割り込み / イベントコントローラ \(EXTI\) を参照してください。](#)

RTC 割り込みを有効にするには、次のシーケンスが必要です。

- 割り込みモードで RTC イベントに対応する EXTI ラインを設定して有効にし、立ち上がりエッジ感度を選択します。
- NVIC で RTC IRQ チャンネルを設定し、有効にします。
- RTCを、RTC 割り込みを生成するように設定します。

表 76. 割り込み制御ビット

割り込みイベント	イベントフラグ	有効化制御ビット	SLEEP モードの 終了	STOP モードの 終了	STANDBY モードの 終了
アラーム A	ALRAF	ALRAIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
RTC_TS 入力 (タイムスタンプ)	TSF	TSIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
RTC_TAMP1 入力検出	TAMP1F	TAMPIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
RTC_TAMP2 入力検出	TAMP2F	TAMPIE	あり	あり ⁽¹⁾	あり ⁽¹⁾
RTC_TAMP3 入力検出 ⁽²⁾	TAMP3F	TAMPIE	あり	あり ⁽¹⁾	あり ⁽¹⁾

- STOP モードおよび STANDBY モードからのウェイクアップは、RTC クロックソースが LSE または LSI のときのみ可能です。
- STM32F07x および STM32F09x デバイスで使用できます。

24.7 RTC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

24.7.1 RTC 時刻レジスタ (RTC_TR)

RTC_TR は、カレンダー時刻シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。 [カレンダーの初期化と設定 \(576 ページ\)](#) および [カレンダーの読み出し \(577 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、 [RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x00

RTC ドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]			HU[3:0]		
									rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rW	rW	rW	rW	rW	rW	rW		rW	rW	rW	rW	rW	rW	rW

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式

1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

24.7.2 RTC 日付レジスタ (RTC_DR)

RTC_DR は、カレンダー日付シャドウレジスタです。このレジスタは、必ず初期化モードで書き込む必要があります。[カレンダーの初期化と設定 \(576 ページ\)](#) および [カレンダーの読み出し \(577 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x04

RTC ドメインリセット値 : 0x0000 2101

システムリセット : BYPSHAD = 0 の場合、0x0000 2101 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]				MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値のままにしておかなければなりません。

ビット 23:20 **YT[3:0]** : BCD 形式での年の十の位

ビット 19:16 **YU[3:0]** : BCD 形式での年の一の位

ビット 15:13 **WDU[2:0]** : 曜日

000 : 禁止
001 : 月曜日
...
111 : 日曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値に保持する必要があります。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

24.7.3 RTC 制御レジスタ (RTC_CR)

アドレスオフセット : 0x08
RTC ドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
								rw	rw	rw	rw	rw	rw	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	Res.	ALRAIE	TSE	WUTE	Res.	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw		rw	rw	rw		rw		rw	rw	rw	rw	rw	rw	rw

- ビット 31:24 予約済みであり、リセット値に保持する必要があります。
- ビット 23 **COE** : 較正出力イネーブル
このビットは、RTC_CALIB 出力を有効にします。
0 : 較正出力は無効です。
1 : 較正出力は有効です。
- ビット 22:21 **OSEL[1:0]** : 出力選択
これらのビットは、RTC_ALARM 出力に送られるフラグの選択に使用します。
00 : 出力は無効です。
01 : アラーム A 出力は有効です。
10 : 予約済み
11 : ウェイクアップ出力は有効です。
- ビット 20 **POL** : 出力極性
このビットは、RTC_ALARM 出力の極性の設定に使用します。
0 : ALRAF/WUTF がアサートされると、このピンがハイになります (OSEL[1:0] に応じて)。
1 : ALRAF/WUTF がアサートされると、このピンがローになります (OSEL[1:0] に応じて)。
- ビット 19 **COSEL** : 較正出力選択
COE = 1 のとき、このビットによって RTC_CALIB に出力される信号を選択します。
0 : 較正出力は 512 Hz です。
1 : 較正出力は 1 Hz です。
これらの周波数は、RTCCLK が 32.768 kHz で、プリスケアラがデフォルト値 (PREDIV_A = 127 および PREDIV_S = 255) の場合に有効です。 [セクション 24.4.15 : 較正クロック出力](#) を参照してください。
- ビット 18 **BKP** : バックアップ
このビットは、サマータイムの変更を実施したか否かを記憶しておくため、ユーザが書き込むことができます。
- ビット 17 **SUB1H** : 1 時間差し引き (冬時間変更)
このビットを初期化モード以外のようにセットすると、現在時刻が 0 でない場合にカレンダー時刻から 1 時間を差し引きします。このビットは常に 0 として読み出されます。
現在時間が 0 のときにこのビットをセットしても、影響はありません。
0 : 影響なし。
1 : 現在時刻から 1 時間差し引きします。これは、冬時間変更で使用できます。

ビット 16 **ADD1H** : 1 時間加算 (サマータイム変更)

このビットを初期化モード以外のときにセットすると、カレンダー時刻に 1 時間加算します。このビットは常に 0 として読み出されます。

0 : 影響なし。

1 : 現在時刻に 1 時間加算します。これは、サマータイム変更に使用できます。

ビット 15 **TSIE** : タイムスタンプ割り込みイネーブル

0 : タイムスタンプ割り込みは無効です。

1 : タイムスタンプ割り込みは有効です。

ビット 14 **WUTIE** : ウェイクアップタイマ割り込みイネーブル

0 : ウェイクアップタイマ割り込みは無効です。

1 : ウェイクアップタイマ割り込みは有効です。

ビット 13 予約済みであり、リセット値のままにしておかなければなりません。

ビット 12 **ALRAIE** : アラーム A 割り込みイネーブル

0 : アラーム A 割り込みは無効です。

1 : アラーム A 割り込みは有効です。

ビット 11 **TSE** : タイムスタンバイイネーブル

0 : タイムスタンプは無効です。

1 : タイムスタンプは有効です。

ビット 10 **WUTE** : ウェイクアップタイマイネーブル

0 : ウェイクアップタイマは無効です。

1 : ウェイクアップタイマは有効です。

ビット 9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8 **ALRAE** : アラーム A イネーブル

0 : アラーム A は無効です。

1 : アラーム A は有効です。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FMT** : 時間形式

0 : 24 時間/日形式

1 : AM/PM 時間形式

ビット 5 **BYP SHAD** : シャドウレジスタをバイパスします。

0 : カレンダー値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、シャドウレジスタから取得され、これらは 2 RTCCLK サイクルごとに 1 回更新されます。

1 : カレンダー値は (RTC_SSR、RTC_TR、RTC_DR から読み出す場合)、カレンダーカウンタから直接取得されます。

注 : *APB クロックの周波数が RTCCLK の 7 倍未満である場合、BYP SHAD は“1”にセットする必要があります。*

ビット 4 **REFCKON** : RTC_REFIN リファレンスクロック検出イネーブル (50 または 60 Hz)

0 : RTC_REFIN 検出は無効です。

1 : RTC_REFIN 検出は有効です。

注 : *PREDIV_S は 0x00FF である必要があります。*

ビット 3 **TSEDGE** : タイムスタンプイベントアクティブエッジ

0 : RTC_TS 入力の立ち上がりエッジによってタイムスタンプイベントを生成します。

1 : RTC_TS 入力の立ち下がりエッジによってタイムスタンプイベントを生成します。

不要な TSF 設定を回避するため、TSEDGE が変化した場合には TSE をリセットする必要があります。

ビット 2:0 **WUCKSEL[2:0]** : ウェイクアップクロック選択

000 : RTC/16 クロックが選択されます。

001 : RTC/8 クロックが選択されます。

010 : RTC/4 クロックが選択されます。

011 : RTC/2 クロックが選択されます。

10x : ck_spre (通常は 1 Hz) クロックが選択されます。

11x : ck_spre (通常は 1 Hz) クロックが選択され、 2^{16} が WUT カウンタ値に加算されます (下記注を参照)。

注 : *初期化モード (RTC_ISR/INITF = 1) の場合のみ、このレジスタのビット 7、6、4 が書き込めます。*
WUT = ウェイクアップユニットカウンタ値 $WUT = (0x0000 \sim 0xFFFF) + 0x10000$ (WUCKSEL[2:1] = 11 の場合追加されます。)
このレジスタのビット 2~0 は、RTC_CR WUTE ビット = 0 かつ RTC_ISR WUTWF ビット = 1 の場合にのみ書き込めます。
カレンダーの時間項目のインクリメント中は時間を変更しないことが推奨されます。カレンダーの時間項目のインクリメントがマスクされる可能性があるためです。
ADD1H および SUB1H の変更は、次の秒から有効になります。
このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

24.7.4 RTC 初期化とステータスレジスタ (RTC_ISR)

このレジスタは、書き込み保護されています (RTC_ISR[13:8] ビットを除く)。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x0C

RTC ドメインリセット値 : 0x0000 0007

システムリセット : 0 にクリアされる INIT、INITF、RSF ビット以外は影響されません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RECALPF
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	Res.	ALRAF	INIT	INITF	RSF	INITs	SHPF	WUTWF	Res.	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0		rc_w0	rw	r	rc_w0	r	r	r		r

ビット 31:17 予約済みであり、リセット値のままにしておかなければなりません。

ビット 16 **RECALPF** : 再校正保留フラグ

ソフトウェアによって RTC_CALR レジスタに書き込みが行われると、RECALPF ステータスフラグが自動的に“1”にセットされ、RTC_CALR レジスタがブロックされたことを示します。新たな校正設定が認識されると、このビットは“0”に戻ります。[動作中の再校正](#)を参照してください。

ビット 15 **TAMP3F** : RTC_TAMP3 検出フラグ

このフラグは、RTC_TAMP3 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 14 **TAMP2F** : RTC_TAMP2 検出フラグ

このフラグは、RTC_TAMP2 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 13 **TAMP1F** : RTC_TAMP1 検出フラグ

このフラグは、RTC_TAMP1 入力にタンパ検出イベントが検出されたときに、ハードウェアによってセットされます。
ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 12 **TSOVF** : タイムスタンプオーバーフローフラグ

このフラグは、TSF が既にセットされている間にタイムスタンプイベントが発生したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。TSOVF は、TSF ビットをクリアした後、チェックしてからクリアすることが推奨されます。そうしないと、TSF ビットがクリアされる直前にタイムスタンプイベントが発生した場合、オーバーフローを見逃す可能性があります。

ビット 11 **TSF** : タイムスタンプフラグ

このフラグは、タイムスタンプイベントが発生したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 10 **WUTF** : ウェイクアップタイマフラグ

このフラグは、ウェイクアップ自動再ロードカウンタが 0 に到達したときに、ハードウェアによってセットされます。
このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。
このフラグは、WUTF が再び 1 にセットされる前、RTCCLK 1.5 周期以上前にソフトウェアでクリアする必要があります。

ビット 9 予約済みであり、リセット値に保持する必要があります。



ビット 8 ALRAF : アラーム A フラグ

このフラグは、時刻/日付レジスタ (RTC_TR および RTC_DR) がアラーム A レジスタ (RTC_ALRMAR) と一致したときにハードウェアによってセットされます。

このフラグは、ソフトウェアで 0 を書き込むことによってクリアされます。

ビット 7 INIT : 初期化モード

0 : フリーランニングモード

1 : 時刻と日付レジスタ (RTC_TR と RTC_DR)、およびプリスケアラレジスタ (RTC_PRER) のプログラムに使用する初期化モードです。INIT がリセットされると、カウンタは停止し、新しい値からカウントし始めます。

ビット 6 INITF : 初期化フラグ

このビットが 1 にセットされると、RTC は初期化状態となり、時刻、日付およびプリスケアラレジスタが更新できます。

0 : カレンダーレジスタを更新できません。

1 : カレンダーレジスタを更新できます。

ビット 5 RSF : レジスタ同期フラグ

このビットは、カレンダーレジスタがシャドウレジスタ (RTC_SSRx、RTC_TRx および RTC_DRx) にコピーされるたびにハードウェアによってセットされます。このビットは、シフト操作が保留中 (SHPF = 1) に初期化モードで、またはバイパスシャドウレジスタモード (BYPSSHAD = 1) で、ハードウェアによってクリアされます。このビットは、ソフトウェアでクリアすることもできます。

初期化モードでソフトウェアまたはハードウェアによってクリアされます。

0 : カレンダーシャドウレジスタはまだ同期していません。

1 : カレンダーシャドウレジスタは同期しています。

ビット 4 INITS : 初期化ステータスフラグ

このビットは、カレンダーの年の項目が 0 ではないとき (RTC ドメインリセット状態) にハードウェアによってセットされます。

0 : カレンダーは初期化されていません。

1 : カレンダーは初期化されています。

ビット 3 SHPF : シフト操作保留

0 : 保留中のシフト操作はありません。

1 : 保留中のシフト操作があります。

このフラグは、RTC_SHIFTR への書き込みによってシフト操作が開始された直後に、ハードウェアによってセットされます。該当するシフト操作が実行されると、ハードウェアによってクリアされます。SHPF ビットに書き込んでも影響はありません。

ビット 2 WUTWF : ウェイクアップタイマ書き込みフラグ

このビットは、ハードウェアによって WUTE ビットが RTC_CR で 0 にセットされた後で 2 RTCCLK サイクルまでセットされ、WUTE ビットが 1 にセットされた後で 2 RTCCLK サイクルまでクリアされます。ウェイクアップタイマ値は、WUTE ビットがクリアされ、WUTWF がセットされたときに変更されます。

0 : ウェイクアップタイマ設定は更新できません。

1 : ウェイクアップタイマ設定は更新できます。

ビット 1 予約済みであり、リセット値に保持する必要があります。**ビット 0 ALRAWF** : アラーム A 書き込みフラグ

このビットは、RTC_CR で ALRAE ビットが 0 にセットされた後、アラーム A 値が変更可能な時にハードウェアによってセットされます。

初期化モードでハードウェアによってクリアされます。

0 : アラーム A は更新できません。

1 : アラーム A は更新できます。

注 : **ALRAF、WUTF、および TSF ビットは、0 にプログラムされてから 2 APB クロックサイクル後にクリアされます。**

24.7.5 RTC プリスケアラレジスタ (RTC_PRER)

このレジスタは、必ず初期化モードで書き込む必要があります。初期化は、2 回の書き込みアクセスに分けて行う必要があります。[カレンダーの初期化と設定 \(576 ページ\)](#) を参照してください。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x10

RTC ドメインリセット値 : 0x007F 00FF

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]						
									r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREDIV_S[14:0]														
	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22:16 **PREDIV_A[6:0]** : 非同期プリスケアラ分周比
非同期分周比です。
 $ck\_apre \text{ 周波数} = RTCCLK \text{ 周波数} / (PREDIV\_A + 1)$

ビット 15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 **PREDIV_S[14:0]** : 同期プリスケアラ分周比
同期分周比です。
 $ck\_spre \text{ 周波数} = ck\_apre \text{ 周波数} / (PREDIV\_S + 1)$

24.7.6 RTC ウェイクアップタイマレジスタ（RTC_WUTR）

このレジスタは、RTC_ISR の WUTWF が 1 にセットされているときにのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット：0x14

RTC ドメインリセット値：0x0000 FFFF

システムリセット：影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WUT[15:0]															
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **WUT[15:0]**：ウェイクアップ自動再ロード値ビット

ウェイクアップタイマが有効（WUTE が 1 にセット）なとき、ck_wut の（WUT[15:0] + 1）サイクル毎に WUTF フラグがセットされます。ck_wut の周期は、RTC_CR レジスタの WUCKSEL[2:0] ビットで選択します。

WUCKSEL[2] = 1 のとき、ウェイクアップタイマは 17 ビットとなり、WUCKSEL[1] が事実上タイマに再ロードされる最上位ビットである WUT[16] となります。

WUTF の最初のアサートは、WUTE がセットされてからck_wut の（WUT+1）サイクル後に発生します。WUCKSEL[2:0] = 011 (RTCCLK/2) のときに WUT[15:0] を 0x0000 にセットすることはできません。



24.7.7 RTC アラーム A レジスタ (RTC_ALRMAR)

このレジスタは、RTC_ISR の ALRAWF が 1 にセットされた場合、または初期化モードの場合にのみ書き込みめます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x1C

RTC ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

- ビット 31 **MSK4** : アラーム A 日付マスク
- 0 : 日付／曜日が一致すると、アラーム A がセットされます。
 - 1 : アラーム A の比較では日付／曜日を無視します。

- ビット 30 **WDSEL** : 曜日選択
- 0 : DU[3:0] は日付の一の位を表します。
 - 1 : DU[3:0] は曜日を表します。DT[1:0] は無視されます。

- ビット 29:28 **DT[1:0]** : BCD 形式での日の十の位
- ビット 27:24 **DU[3:0]** : BCD 形式での日の一の位または曜日

- ビット 23 **MSK3** : アラーム A 時マスク
- 0 : 時が一致すると、アラーム A がセットされます。
 - 1 : アラーム A の比較では時を無視します。

- ビット 22 **PM** : AM/PM 表記
- 0 : AM または 24 時間形式
 - 1 : PM

- ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位
- ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

- ビット 15 **MSK2** : アラーム A 分マスク
- 0 : 分が一致すると、アラーム A がセットされます。
 - 1 : アラーム A の比較では分を無視します。

- ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位
- ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

- ビット 7 **MSK1** : アラーム A 秒マスク
- 0 : 秒が一致すると、アラーム A がセットされます。
 - 1 : アラーム A の比較では秒を無視します。

- ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位
- ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位



24.7.8 RTC 書き込み保護レジスタ (RTC_WPR)

アドレスオフセット : 0x24

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **KEY** : 書き込み保護キー

このバイトはソフトウェアで書き込まれます。

このバイトを読み出すと常に 0x00 が返されます。

RTC レジスタの書き込み保護解除方法については、[RTC レジスタ書き込み保護](#) を参照してください。

24.7.9 RTC サブセカンドレジスタ (RTC_SSR)

アドレスオフセット : 0x28

RTC ドメインリセット値 : 0x0000 0000

システムリセット : BYPSHAD = 0 の場合、0x0000 0000 です。BYPSHAD = 1 の場合、影響を受けません。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS** : サブセカンド値

SS[15:0] は、同期プリスケアラのカウンタ内の値です。秒の小数部は、下の式によって与えられます。

秒の小数部 = (PREDIV_S - SS) / (PREDIV_S + 1)

注 : **SS** は、シフト操作後に限り、PREDIV_S より大きな値となる場合があります。この場合、正確な時刻/日付は、RTC_TR/RTC_DR で示される値よりも 1 秒少ない値となります。

24.7.10 RTC シフト制御レジスタ (RTC_SHIFTR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x2C

RTC ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SUBFS[14:0]														
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

ビット 31 **ADD1S** : 1 秒加算

0 : 影響なし。

1 : 時計／カレンダーに 1 秒加算します。

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

この機能は、SUBFS (下記説明を参照) と共に使用されることを想定しており、不可分操作で、効果的に時計に秒の小数部を加算することを目的としています。

ビット 30:15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:0 **SUBFS** : 秒の小数部差し引き

このビットは書き込み専用であり、常に 0 として読み出されます。シフト操作が保留中 (RTC_ISR で SHPF = 1) の場合、このビットに書き込んでも影響はありません。

SUBFS に書き込まれた値は、同期ブリスケーラのカウンタに加算されます。このカウンタはカウントダウンして行くので、この操作によって、次の式で求める値が効果的にクロックから差し引き(遅延)されます。

遅れ (秒) = SUBFS / (PREDIV_S + 1)

ADD1S 機能が SUBFS と共に用いられた場合、秒の小数部を効果的にクロックに加算する (クロックを進める) ことができ、実際のクロックの進みは次の式のとおりとなります。

進み (秒) = (1 - (SUBFS / (PREDIV_S + 1)))

注 : **SUBFS に書き込むことにより RSF はクリアされます。その後、ソフトウェアが RSF = 1 まで待つことにより、シャドウレジスタがシフトされた時刻で更新されていることが確実にになります。**

24.7.11 RTC タイムスタンプ時刻レジスタ（RTC_TSTR）

このレジスタの内容は、RTC_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

アドレスオフセット：0x30

RTC ドメインリセット値：0x0000 0000

システムリセット：影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

ビット 31:23 予約済みであり、リセット値のままにしておかなければなりません。

ビット 22 **PM** : AM/PM 表記

0 : AM または 24 時間形式
1 : PM

ビット 21:20 **HT[1:0]** : BCD 形式での時の十の位

ビット 19:16 **HU[3:0]** : BCD 形式での時の一の位

ビット 15 予約済みであり、リセット値のままにしておかなければなりません。

ビット 14:12 **MNT[2:0]** : BCD 形式での分の十の位

ビット 11:8 **MNU[3:0]** : BCD 形式での分の一の位

ビット 7 予約済みであり、リセット値のままにしておかなければなりません。

ビット 6:4 **ST[2:0]** : BCD 形式での秒の十の位

ビット 3:0 **SU[3:0]** : BCD 形式での秒の一の位

24.7.12 RTC タイムスタンプ日付レジスタ (RTC_TSDR)

このレジスタの内容は、RTC_ISR で TSF が 1 にセットされている場合にのみ有効です。また、TSF ビットがリセットされるとクリアされます。

アドレスオフセット : 0x34

RTC ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:13 **WDU[1:0]** : 曜日

ビット 12 **MT** : BCD 形式での月の十の位

ビット 11:8 **MU[3:0]** : BCD 形式での月の一の位

ビット 7:6 予約済みであり、リセット値のままにしておかなければなりません。

ビット 5:4 **DT[1:0]** : BCD 形式での日の十の位

ビット 3:0 **DU[3:0]** : BCD 形式での日の一の位

24.7.13 RTC タイムスタンプサブセカンドレジスタ（RTC_TSSSR）

このレジスタの内容は、RTC_ISR/TSF がセットされている場合にのみ有効です。また、RTC_ISR/TSF ビットがリセットされるとクリアされます。

アドレスオフセット：0x38

RTC ドメインリセット値：0x0000 0000

システムリセット：影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:0 **SS**：サブセカンド値

SS[15:0] は、タイムスタンプイベントが発生したときの同期プリスケアラのカウンタの値です。



24.7.14 RTC 較正レジスタ (RTC_CALR)

このレジスタは書き込み保護されています。書き込みアクセスの手順は、[RTC レジスタ書き込み保護 \(575 ページ\)](#) を参照してください。

アドレスオフセット : 0x3C

RTC ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
r/w	r/w	r/w					r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15 **CALP** : RTC 周波数を 488.5 ppm 増加

0 : RTCCLK パルスは加えられません。

1 : RTCCLK の 2^{11} パルスごとに 1 パルス効果的に挿入されます (周波数が 488.5 ppm 増加)。

この機能は、CALM と共に使用されることを想定しており、カレンダーの周波数を高分解能で下げることができます。入力周波数が 32768 Hz の場合、32 秒枠の間に追加される RTCCLK パルスの数は次のように算出されます : $(512 * CALP) - CALM$

[セクション 24.4.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 14 **CALW8** : 8 秒較正サイクル周期の使用

CALW8 が“1”にセットされると、8 秒較正サイクル周期が選択されます。

注 : **CALW8 = 1 の場合、CALM[1:0] は“00”に固定されます。**[セクション 24.4.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 13 **CALW16** : 16 秒較正サイクル周期の使用

CALW16 が“1”にセットされると、16 秒較正サイクル周期が選択されます。CALW8 = 1 の場合、このビットを“1”にセットすることはできません。

注 : **CALW16 = 1 の場合、CALM[0] は“0”に固定されます。**[セクション 24.4.12 : RTC の高精度デジタル較正](#) を参照してください。

ビット 12:9 予約済みであり、リセット値のままにしておかなければなりません。

ビット 8:0 **CALM[8:0]** : 較正マイナス

RTCCLK 2^{20} パルス (入力周波数が 32768 Hz の場合 32 秒) 内の CALM をマスクすることによって、カレンダーの周波数が下げられます。この方法により、カレンダーの周波数を 0.9537 ppm の分解能で下げることができます。

カレンダーの周波数を上げるには、この機能を CALP と共に使用する必要があります。[セクション 24.4.12 : RTC の高精度デジタル較正 \(580 ページ\)](#) を参照してください。

24.7.15 RTC タンパおよびオルタネート機能設定レジスタ (RTC_TAFCR)

アドレスオフセット : 0x40
RTC ドメインリセット値 : 0x0000 0000
システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC15 MODE	PC15 VALUE	PC14 MODE	PC14 VALUE	PC13 MODE	PC13 VALUE	Res.	Res.
								rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMPP UDIS	TAMPPRCH [1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPTS	Res.	Res.	TAMP2T RG	TAMP2E	TAMPIE	TAMP1T RG	TAMP1 E
rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw

- ビット 31:24 予約済みであり、リセット値に保持する必要があります。
- ビット 23 **PC15MODE** : PC15 モード
0 : PC15 は、GPIO 設定レジスタによって制御されます。そのため、PC15 は STANDBY モードでフローティング状態になります。
1 : LSE が無効の場合、PC15 は強制的にプッシュプル出力に設定されます。
- ビット 22 **PC15VALUE** : PC15 値
LSE が無効で PC15MODE = 1 の場合、PC15VALUE が PC15 出力データを設定します。
- ビット 21 **PC14MODE** : PC14 モード
0 : PC14 は、GPIO 設定レジスタによって制御されます。そのため、PC14 は STANDBY モードでフローティング状態になります。
1 : LSE が無効の場合、PC14 は強制的にプッシュプル出力に設定されます。
- ビット 20 **PC14VALUE** : PC14 値
LSE が無効で PC14MODE = 1 の場合、PC14VALUE が PC14 出力データを設定します。
- ビット 19 **PC13MODE** : PC13 モード
0 : PC13 は、GPIO 設定レジスタによって制御されます。そのため、PC13 は STANDBY モードでフローティング状態になります。
1 : RTC オルタネート機能がすべて無効の場合、PC13 は強制的にプッシュプル出力に設定されます。
- ビット 18 **PC13VALUE** : RTC_ALARM 出力タイプ/PC13 値
PC13 を使用して RTC_ALARM を出力した場合、PC13VALUE が出力設定を設定します。
0 : RTC_ALARM はオープンドレイン出力になります。
1 : RTC_ALARM はプッシュプル出力になります。
RTC 代替機能がすべて無効で PC13MODE = 1 の場合、PC13VALUE が PC13 出力データを設定します。
- ビット 17:16 予約済みであり、リセット値に保持する必要があります。
- ビット 15 **TAMPPUDIS** : RTC_TAMPx ブルアップディセーブル
このビットにより、毎回のサンプリング前に各 RTC_TAMPx ピンをプリチャージするかどうか決定します。
0 : サンプリング前に RTC_TAMPx ピンをプリチャージします (内部ブルアップを有効化)。
1 : RTC_TAMPx ピンのプリチャージを無効化します。

ビット 14:13 **TAMPPRCH[1:0]** : RTC_TAMPx プリチャージ時間

これらのビットにより、各サンプリングの前にプルアップを有効化している時間を決定します。
TAMPPRCH は、各 RTC_TAMPx 入力に対して有効です。

0x0 : 1 RTCCLK サイクル

0x1 : 2 RTCCLK サイクル

0x2 : 4 RTCCLK サイクル

0x3 : 8 RTCCLK サイクル

ビット 12:11 **TAMPFLT[1:0]** : RTC_TAMPx フィルタカウント

これらのビットにより、タンパイブントをアクティブにするのに必要な指定のレベル (TAMP*TRG) での連続サンプリングの数を決定します。TAMPFLT は、各 RTC_TAMPx 入力に対して有効です。

0x0 : RTC_TAMPx 入力 that アクティブレベル (RTC_TAMPx 入力における内部プルアップ無し) に遷移するときのエッジで、タンパイブントがアクティブになります。

0x1 : アクティブレベルでの連続した 2 回のサンプリングの後、タンパイブントがアクティブになります。

0x2 : アクティブレベルでの連続した 4 回のサンプリングの後、タンパイブントがアクティブになります。

0x3 : アクティブレベルでの連続した 8 回のサンプリングの後、タンパイブントがアクティブになります。

ビット 10:8 **TAMPFREQ[2:0]** : タンパサンプリング周波数

各 RTC_TAMPx 入力 that サンプリングされる周波数を決定します。

0x0 : RTCCLK / 32768 (RTCCLK = 32768 Hz の場合 1 Hz)

0x1 : RTCCLK / 16384 (RTCCLK = 32768 Hz の場合 2 Hz)

0x2 : RTCCLK / 8192 (RTCCLK = 32768 Hz の場合 4 Hz)

0x3 : RTCCLK / 4096 (RTCCLK = 32768 Hz の場合 8 Hz)

0x4 : RTCCLK / 2048 (RTCCLK = 32768 Hz の場合 16 Hz)

0x5 : RTCCLK / 1024 (RTCCLK = 32768 Hz の場合 32 Hz)

0x6 : RTCCLK / 512 (RTCCLK = 32768 Hz の場合 64 Hz)

0x7 : RTCCLK / 256 (RTCCLK = 32768 Hz の場合 128 Hz)

ビット 7 **TAMPTS** : タンパ検出イベント時のタイムスタンプの有効化

0 : タンパ検出イベントがあっても、タイムスタンプは保存されません。

1 : タンパ検出イベント時、タイムスタンプが保存されます。

RTC_CR レジスタで TSE = 0 であっても TAMPTS は有効です。

ビット 6 **TAMP3TRG** : RTC_TAMP3 入力のアクティブレベル

TAMPFLT != 00 の場合

0 : RTC_TAMP3 入力 that ローのままのとき、タンパ検出イベント that トリガされます。

1 : RTC_TAMP3 入力 that ハイのままのとき、タンパ検出イベント that トリガされます。

TAMPFLT = 00 の場合

0 : RTC_TAMP3 入力の立ち上がりエッジ with タンパ検出イベント that トリガされます。

1 : RTC_TAMP3input の立ち下りエッジ with タンパ検出イベント that トリガされます。

ビット 5 **TAMP3E** : RTC_TAMP3 検出イネーブル

0 : RTC_TAMP3 入力検出は無効です。

1 : RTC_TAMP3 入力検出は有効です。

ビット 4 **TAMP2TRG** : RTC_TAMP2 入力のアクティブレベル

TAMPFLT != 00 の場合

0 : RTC_TAMP2 入力 that ローのままのとき、タンパ検出イベント that トリガされます。

1 : RTC_TAMP2 入力 that ハイのままのとき、タンパ検出イベント that トリガされます。

TAMPFLT = 00 の場合

0 : RTC_TAMP2 入力の立ち上がりエッジ with タンパ検出イベント that トリガされます。

1 : RTC_TAMP2 入力の立ち下りエッジ with タンパ検出イベント that トリガされます。

ビット 3 **TAMP2E** : RTC_TAMP2 入力検出イネーブル

- 0 : RTC_TAMP2 検出は無効です。
- 1 : RTC_TAMP2 検出は有効です。

ビット 2 **TAMPIE** : タンパ割り込みイネーブル

- 0 : タンパ割り込みは無効です。
- 1 : タンパ割り込みは有効です。

ビット 1 **TAMP1TRG** : RTC_TAMP1 入力のアクティブレベル

TAMPFLT != 00 の場合

- 0 : RTC_TAMP1 入力がローのままのとき、タンパ検出イベントがトリガされます。
- 1 : RTC_TAMP1 入力がハイのままのとき、タンパ検出イベントがトリガされます。

TAMPFLT = 00 の場合

- 0 : RTC_TAMP1 入力の立ち上がりエッジでタンパ検出イベントがトリガされます。
- 1 : RTC_TAMP1 入力の立ち下がりエッジでタンパ検出イベントがトリガされます。

ビット 0 **TAMP1E** : RTC_TAMP1 入力検出イネーブル

- 0 : RTC_TAMP1 検出は無効です。
- 1 : RTC_TAMP1 検出は有効です。

注意 : **TAMPFLT = 0 の場合、TAMP1E ビットは、TAMP1TRG が変更されたらリセットし、TAMP1F の不要なセットを回避する必要があります。**

24.7.16 RTC アラーム A サブセカンドレジスタ (RTC_ALRMASSR)

このレジスタは、RTC_CR レジスタの ALRAIE がリセットされた場合、または初期化モードの場合にのみ書き込みます。

このレジスタは書き込み保護されています。書き込みアクセスの手順は、RTC レジスタ書き込み保護を参照してください。[RTC レジスタ書き込み保護 \(575 ページ\)](#)

アドレスオフセット : 0x44

RTC ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MASKSS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rW	rW	rW	rW								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SS[14:0]														
	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:24 MASKSS[3:0] : このビットから始まる最上位ビットのマスク

- 0 : アラーム A に対してサブセカンドを比較しません。このアラームは秒の位がインクリメントされたときにセットされます (他の項目が一致していることを前提として)。
- 1 : アラーム A の比較では SS[14:1] を無視します。SS[0] のみ比較されます。
- 2 : アラーム A の比較では SS[14:2] を無視します。SS[1:0] のみ比較されます。
- 3 : アラーム A の比較では SS[14:3] を無視します。SS[2:0] のみ比較されます。
- ...
- 12 : アラーム A の比較では SS[14:12] を無視します。SS[11:0] が比較されます。
- 13 : アラーム A の比較では SS[14:13] を無視します。SS[12:0] が比較されます。
- 14 : アラーム A の比較では SS[14] を無視します。SS[13:0] が比較されます。
- 15 : アラームをアクティブにするには、15 の全ての SS ビットを比較し一致する必要があります。同期カウンタのオーバーフロービット (ビット 15) が比較されることはありません。このビットは、シフト操作後に限り、0 でなくなる場合があります。

ビット 23:15 予約済みであり、リセット値に保持する必要があります。

ビット 14:0 SS[14:0] : サブセカンド値

この値が同期プリスケアラのカウンタの内容と比較され、アラーム A をアクティブ化するかどうかを決定します。0~MASKSS-1 のビットだけが比較されます。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

24.7.17 RTC バックアップレジスタ (RTC_BKPxR)

アドレスオフセット : 0x50 から 0x60

RTC ドメインリセット値 : 0x0000 0000

システムリセット : 影響なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BKP[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw

ビット 31:0 BKP[31:0]

アプリケーションはこれらのレジスタに対してデータの読み書きをすることができます。
これらのレジスタは、V_{DD} がオフになった場合、V_{BAT} によって電源が供給されるため、システムリセットによりリセットされず、デバイスが低電力モードで動作する場合、レジスタの内容は有効なまま保持されます。
このレジスタは、TAMPx_F = 1 である限り、あるいはフラッシュ読み出し保護が無効なときは、タンパ検出イベント時にリセットされます。



24.7.18 RTC レジスタマップ

表 77. RTC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0x00	RTC_TR	Res	Res	Res	Res	Res	Res	Res	Res	Res	PM	HT [1:0]	HU[3:0]				Res	MNT[2:0]			MNU[3:0]			Res	ST[2:0]			SU[3:0]							
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	RTC_DR	Res	Res	Res	Res	Res	Res	Res	Res	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]			Res	Res	DT [1:0]		DU[3:0]						
	リセット値									0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	Res		0	0	0	0	0	1	
0x08	RTC_CR	Res	Res	Res	Res	Res	Res	Res	Res	COE	OSEL [1:0]	POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	Res	ALRAIE	TSE	WUTE	Res	ALRAE	Res	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]					
	リセット値									0	0	0	0		0	0	0	0		0	0	0		0	0	0	0	0	0	0	0	0	0		
0x0C	RTC_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RECALPF	TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	Res	ALRAF	INIT	INTF	RSF	INITS	SHPF	WUT WF	Res	ALRAWF				
	リセット値														0		0	0	0	0	0	0		0	0	0	0	0	0	1		1			
0x10	RTC_PRER	Res	Res	Res	Res	Res	Res	Res	Res	PREDIV_A[6:0]							PREDIV_S[14:0]																		
	リセット値										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1		
0x14	RTC_WUTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUT[15:0]																		
	リセット値																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x1C	RTC_ALRMAR	MSK4	WDSEL	DT [1:0]	DU[3:0]				MSK3	PM	HT [1:0]	HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]			MSK1	ST[2:0]			SU[3:0]								
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x24	RTC_WPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY										
	リセット値																								0	0	0	0	0	0	0	0	0		
0x28	RTC_SSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SS[15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x2C	RTC_SHIFTR	ADD1S	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SUBFS[14:0]																	
	リセット値	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x30	RTC_TSTR	Res	Res	Res	Res	Res	Res	Res	Res	Res	PM	HT [1:0]	HU[3:0]				Res	MNT[2:0]			MNU[3:0]			Res	ST[2:0]			SU[3:0]							
	リセット値										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0		
0x34	RTC_TSDR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDU[1:0]			MT	MU[3:0]			Res	Res	DT [1:0]		DU[3:0]							
	リセット値																0	0	0	0	0	0	0	0			0	0	0	0	0	0	0		
0x38	RTC_TSSSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SS[15:0]																		
	リセット値																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x3C	RTC_CALR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CALP	CALW8	CALW16	Res	Res	Res	Res	Res	CALM[8:0]										
	リセット値																0	0	0						0	0	0	0	0	0	0	0	0	0	

表 77. RTC レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x40	RTC_TAFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC15MODE	PC15MODE	PC14VALUE	PC14MODE	PC13VALUE	PC13VALUE	Res.	Res.	TAMPPUDIS	TAMPPRCH[1:0]	13	TAMPFLT[1:0]	11	10	TAMPFREQ[2:0]		7	TAMPTS	TAMP3-TRG	TAMP3E	TAMP2-TRG	TAMP2E	TAMP1E	TAMP1TRG	TAMP1E
	リセット値									0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x44	RTC_ALRMASSR	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]															
	リセット値					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x50 to 0x60	RTC_BKP0R	BKP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	to RTC_BKP4R	BKP[31:0]																																
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。



25 I2C (Inter-integrated circuit) インタフェース

25.1 概要

I²C (Inter-integrated circuit) バスインタフェースは、マイクロコントローラとシリアル I²C バス間の通信を処理します。マルチマスタ機能を備え、すべての I²C バス固有のシーケンシング、プロトコル、アービトレーション、およびタイミングを制御します。標準モード (Sm)、高速モード (Fm)、および高速モードプラス (Fm+) をサポートします。

また、SMBus (System Management Bus) および PMBus (Power Management Bus) と互換性があります。

DMA を使用して、CPU の負荷を軽減できます。

25.2 I2C の主な機能

- I²C バス仕様 rev03 との互換性 :
 - スレーブおよびマスタモード
 - マルチマスタ機能
 - 標準モード (最大 100 kHz)
 - 高速モード (最大 400 kHz)
 - 高速モードプラス (最大 1 MHz)
 - 7ビットおよび 10 ビットアドレッシングモード
 - 複数の 7 ビットスレーブアドレス (2 つのアドレス、1 つは設定可能なマスク付き)
 - すべての 7 ビットアドレス確認応答モード
 - 同報 (General call) コール
 - プログラム可能なセットアップおよびホールド時間
 - 使いやすいイベント管理
 - クロックストレッチオプション
 - ソフトウェアリセット

- DMA 機能付きの 1 バイトバッファ
- プログラム可能なアナログおよびデジタルノイズフィルタ

製品の実装によっては、次の追加機能も使用できます ([セクション 25.3 : I2C の実装](#) を参照) :

- SMBus 仕様 rev 2.0 との互換性 :
 - ハードウェア PEC (Packet Error Checking) の生成と ACK 制御による確認
 - コマンドおよびデータ確認応答制御
 - アドレス解決プロトコル (ARP) サポート
 - ホストおよびデバイスのサポート
 - SMBus アラート
 - タイムアウトおよびアイドル条件の検出
- PMBus rev 1.1標準との互換性
- 独立したクロック : 独立したクロックソースの選択により、I2C の通信速度は PCLK の再プログラミングから独立
- アドレス一致時に STOP モードからウェイクアップ

25.3 I2C の実装

このマニュアルでは、I2C1 で実装されている詳細な機能セットについて説明します。I2C2 は、より少ない機能セットをサポートしますが、それ以外は I2C1 と同じです。違いを次の表に示します。

表 78. STM32F0xx I2C 実装

I2C の機能 ⁽¹⁾	STM32F03x STM32F04x	STM32F05x		STM32F07x STM32F09x	
	I2C1	I2C1	I2C2	I2C1	I2C2
7 ビットアドレスモード	X	X	X	X	X
10 ビットアドレスモード	X	X	X	X	X
標準モード (最大 100 kbit/s)	X	X	X	X	X
高速モード (最大 400 kbit/s)	X	X	X	X	X
独立クロック	X	X	-	X	-
SMBus	X	X	-	X	-
STOP モードからのウェイクアップ	X	X	-	X	-
Fm+ モードのための 20 mA 出力駆動	X	X	-	X	X

1. X : サポートされています。

25.4 I2C の機能詳細

データの送受信に加えて、このインタフェースは、データをシリアル形式からパラレル形式（およびその逆）に変換します。割り込みは、ソフトウェアによって有効または無効にできます。このインタフェースは、データピン（SDA）とクロックピン（SCL）によって I2C バスに接続されます。標準（最大 100 kHz）、高速モード（最大 400 kHz）、または高速モードプラス（最大 1 MHz）の I²C バスで接続できます。

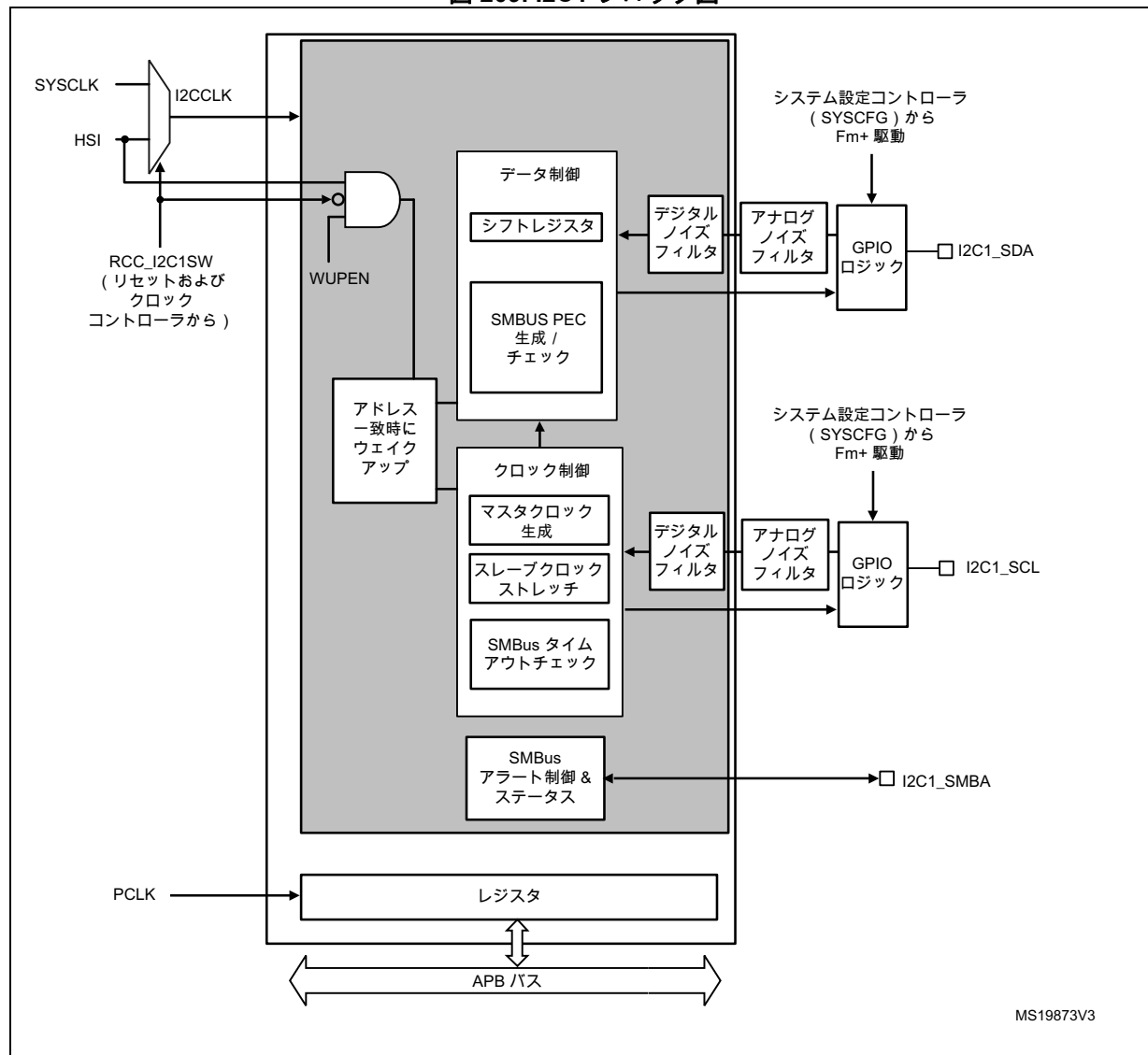
このインタフェースは、データピン（SDA）とクロックピン（SCL）によって SMBus に接続することもできます。

SMBus 機能がサポートされる場合、追加の SMBus アラートピン（SMBA）オプションも使用できます。



I2C1 インタフェースのブロック図を 209 に示します。

図 209. I2C1 ブロック図

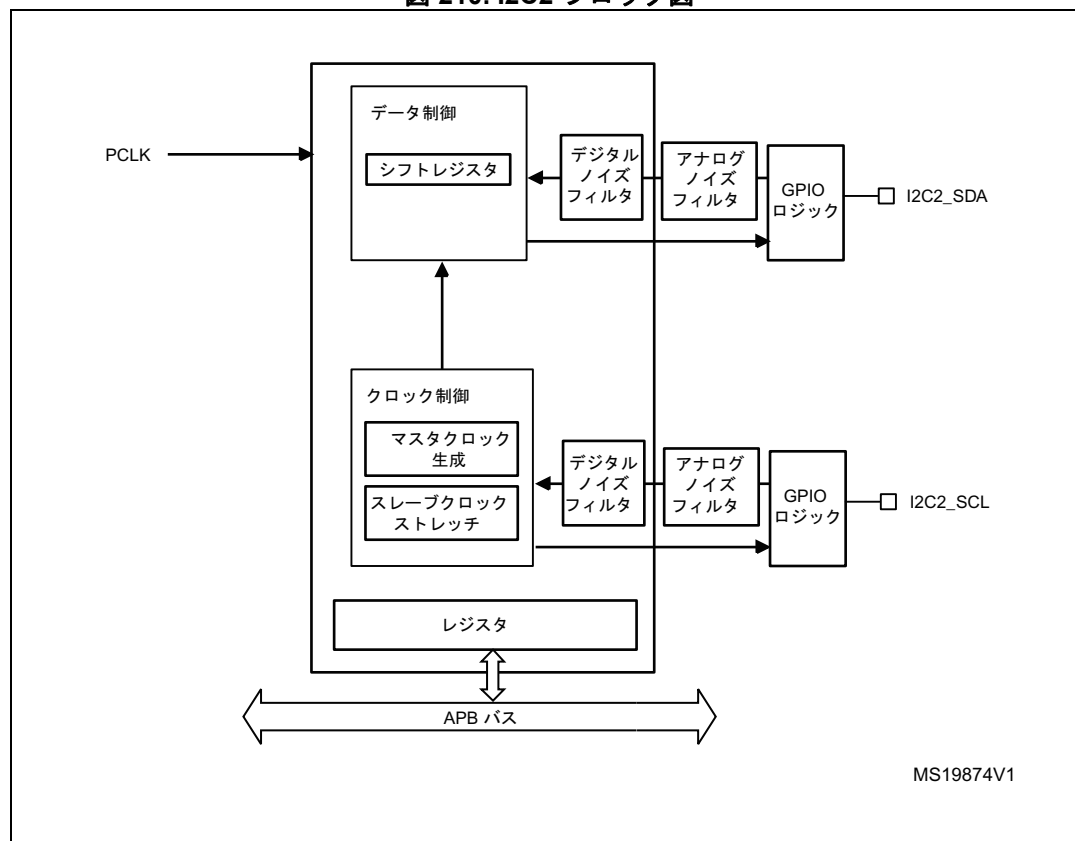


I2C1 I/O は、高速モードプラス動作のための 20 mA 出力電流駆動をサポートします。これを有効にするには、の SCL および SDA の駆動機能制御ビットをセットします。[セクション 10.1.1: SYSCFG 設定レジスタ 1 \(SYSCFG CFGR1\)](#)

25.4.2 I2C2 ブロック図

I2C2 インタフェースのブロック図を [図 210](#) に示します。

図 210. I2C2 ブロック図



MS19874V1

25.4.3 I2C クロックの要件

I2C カーネルは I2CCLK によってクロック供給されます。

I2CCLK の周期 t_{I2CCLK} は、次の条件を満たす必要があります。

$$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4 \text{ and } t_{I2CCLK} < t_{HIGH}$$

ここで：

t_{LOW} : SCL ロー時間、および t_{HIGH} : SCL ハイ時間

$t_{filters}$: 有効なときには、アナログフィルタとデジタルフィルタによる遅延の合計。

アナログフィルタの遅延は、最大 260 ns です。デジタルフィルタの遅延は、 $DNF \times t_{I2CCLK}$ です。

PCLK の周期 t_{PCLK} は、次の条件を満たす必要があります。

$$t_{PCLK} < 4/3 t_{SCL}$$

t_{SCL} : SCL 周期

注意： I2C カーネルが PCLK によってクロック供給されるとき、PCLK は t_{I2CCLK} の条件を満たす必要があります。

25.4.4 モード選択

このインタフェースは、次の 4 つのモードのいずれかで動作できます：

- スレーブトランスミッタ
- スレーブレシーバ
- マスタトランスミッタ
- マスタレシーバ

デフォルトでは、スレーブモードで動作します。このインタフェースは、START コンディションを生成したときにはスレーブからマスタへ、アービトレーションの喪失または STOP 生成が発生したときにはマスタからスレーブへ自動的に切り替わるため、マルチマスタ機能を使用できます。

通信の流れ

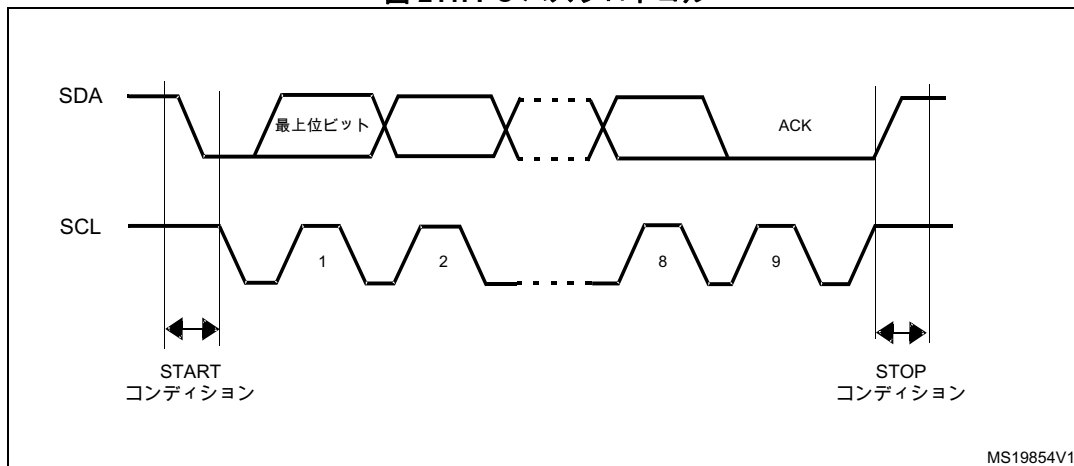
マスタモードでは、I2C インタフェースは、データ転送を開始し、クロック信号を生成します。シリアルデータ転送は、常に START コンディションで開始され、STOP コンディションで終わります。START および STOP コンディションは、マスタモードではソフトウェアによって生成されます。

スレーブモードでは、このインタフェースは、自己アドレス (7 または 10 ビット) と同報アドレスを認識できます。同報アドレスの検出は、ソフトウェアによって有効または無効にできます。予約済みの SMBus アドレスもソフトウェアによって有効にできます。

データとアドレスは、MSB ファーストの 8 ビットバイトとして転送されます。START コンディションの後に続く最初のバイト (7 ビットモードでは 1 バイト、10 ビットモードでは 2 バイト) にアドレスが含まれています。アドレスは、常にマスタモードで送信されます。

8 クロックサイクルのバイト転送の後には 9 番目のクロックパルスが続きます。その間に、レシーバはトランスミッタに確認応答ビットを送信する必要があります。次の図を参照してください。

図 211. I²C バスプロトコル



確認応答 (Acknowledge) は、ソフトウェアによって有効または無効にできます。I2C インタフェースのアドレスは、ソフトウェアによって選択できます。

25.4.5 I2C の初期化

ペリフェラルの有効化と無効化

I2C ペリフェラルクロックは、クロックコントローラで設定し、有効にする必要があります ([セクション 7: リセットおよびクロック制御 \(RCC\)](#) を参照)。

そして、I2C_CR1 レジスタの PE ビットをセットすることによって、I2C を有効にできます。

I2C が無効なときには (PE=0)、I²C はソフトウェアリセットを実行します。詳細については、[セクション 25.4.6: ソフトウェアリセット](#) を参照してください。

ノイズフィルタ

I2C_CR1 レジスタの PE ビットをセットすることによって I2C ペリフェラルを有効にする前に、必要な場合は、ノイズフィルタを設定する必要があります。デフォルトでは、SDA および SCL 入力にアナログノイズフィルタがあります。このアナログフィルタは I²C 仕様に準拠しており、高速モードおよび高速モードプラスで最大 50 ns のパルス幅を持つスパイクを抑制します。ANFOFF ビットをセットすることによって、このアナログフィルタを無効にし、I2C_CR1 レジスタの DNF[3:0] ビットを設定することによってデジタルフィルタを選択することができます。

デジタルフィルタが有効なときには、SCL または SDA ラインのレベルは、DNF x I2CCLK 周期より長く安定していた場合のみ、内部で変更されます。これにより、プログラム可能な 1 ~ 15 I2CCLK 周期の長さを持つスパイクを抑制できます。

表 79. アナログフィルタとデジタルフィルタの比較

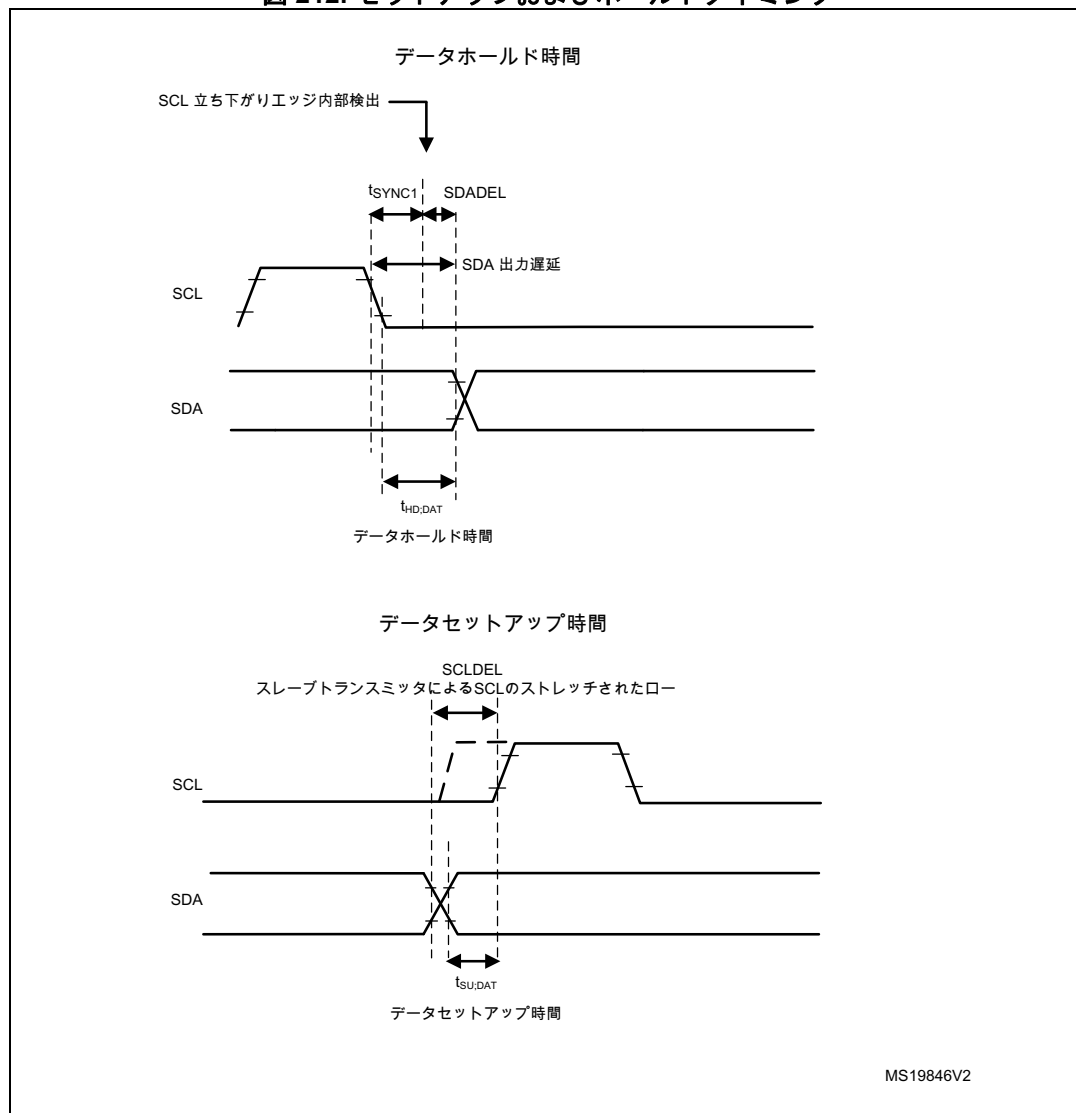
	アナログフィルタ	デジタルフィルタ
抑制されるスパイク のパルス幅	³ 50 ns	長さを 1 ~ 15 I2C ペリフェラルクロックにプログラム 可能
利点	STOP モードで使用可能	– プログラム可能な長さ : 追加のフィルタリング機能対 標準要件 – 安定した長さ
欠点	温度、電圧、 プロセスのばらつき	デジタルフィルタが有効なときには、アドレス一致時の STOP モードからのウェイクアップは使用できない

注意 : I2C が有効なときには、フィルタ構成の変更はできません。

I2C のタイミング

マスタおよびスレーブモードで正しいデータホールドおよびセットアップ時間が使用されるのを保証するためには、タイミングを設定する必要があります。これを行うには、I2C_TIMINGR レジスタの PRESC[3:0]、SCLDEL[3:0]、および SDADEL[3:0] ビットをプログラムします。

図 212. セットアップおよびホールドタイミング



- SCL 立ち下がりエッジが内部で検出されると、SDA 出力を送信する前に遅延が挿入されます。この遅延は、 $t_{SDA\,DEL} = SDADEL \times t_{PRESC} + t_{I2CCLK}$ であり $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。 $t_{SDA\,DEL}$ はホールド時間 $t_{HD,DAT}$ に影響を与えます。

SDA 出力遅延の合計は、次のとおりです：

$$t_{SYNC1} + \{[SDADEL \times (PRESC+1) + 1] \times t_{I2CCLK}\}$$

t_{SYNC1} の長さは、次のパラメータに依存します。

- SCL 立ち下がり傾斜
- アナログフィルタが有効なときの入力遅延: $t_{\text{AF(min)}} < t_{\text{AF}} < t_{\text{AF(max)}} \text{ ns}$
- デジタルフィルタが有効なときの入力遅延: $t_{\text{DNF}} = \text{DNF} \times t_{\text{I2CCLK}}$
- SCL と I2CCLK クロックの同期による遅延 (2 ~ 3 I2CCLK 周期)

SCL 立ち下がりエッジの未定義の領域をブリッジするためには、SDADEL を次のようにプログラムする必要があります:

$$\{t_r(\text{max}) + t_{\text{HD;DAT}}(\text{min}) - t_{\text{AF(min)}} - [(DNF+3) \times t_{\text{I2CCLK}}]\} / \{(PRESC+1) \times t_{\text{I2CCLK}}\} \leq \text{SDADEL}$$

$$\text{SDADEL} \leq \{t_{\text{HD;DAT}}(\text{max}) - t_{\text{AF(max)}} - [(DNF+4) \times t_{\text{I2CCLK}}]\} / \{(PRESC+1) \times t_{\text{I2CCLK}}\}$$

注: $t_{\text{AF(min)}} / t_{\text{AF(max)}}$ は、アナログフィルタが有効なときのみ、等式に含まれます。 t_{AF} の値については、デバイスのデータシートを参照してください。

最大 $t_{\text{HD;DAT}}$ は、標準モード、高速モード、および高速モードプラスで 3.45 μs 、0.9 μs 、および 0.45 μs ですが、遷移時間による $t_{\text{VD;DAT}}$ の最大値より短い必要があります。この最大値を満たす必要があるのは、デバイスが SCL 信号の LOW 周期 (t_{LOW}) をストレッチしない場合だけです。クロックが SCL をストレッチする場合、クロックをリリースする前に、データがセットアップ時間まで有効である必要があります。

SDA 立ち上がりエッジは、通常、最悪ケースであり、この場合、前の等式は次のようになります:

$$\text{SDADEL} \leq \{t_{\text{VD;DAT}}(\text{max}) - t_r(\text{max}) - 260 \text{ ns} - [(DNF+4) \times t_{\text{I2CCLK}}]\} / \{(PRESC+1) \times t_{\text{I2CCLK}}\}$$

注: **NOSTRETCH=0 のときには、SCLDEL の値に従って、デバイスはセットアップ時間を保証するために SCL ローをストレッチするので、この条件に違反することがあります。**

t_r 、 t_r 、 $t_{\text{HD;DAT}}$ 、および $t_{\text{VD;DAT}}$ の標準値については、[表 80: I2C-SMBUS 仕様のデータのセットアップおよびホールド時間](#)を参照してください。

- SDA 出力の送信後、SCL ラインはセットアップ時間中、ローレベルに保たれます。このセットアップ時間は、 $t_{\text{SCLDEL}} = (\text{SCLDEL}+1) \times t_{\text{PRESC}}$ であり $t_{\text{PRESC}} = (\text{PRESC}+1) \times t_{\text{I2CCLK}}$ です。

t_{SCLDEL} は、セットアップ時間 $t_{\text{SU;DAT}}$ に影響を与えます。

SDA 遷移 (立ち上がりエッジは通常、最悪のケース) の未定義の領域をブリッジするためには、SCLDEL を次のようにプログラムする必要があります:

$$\{[t_r(\text{max}) + t_{\text{SU;DAT}}(\text{min})] / [(\text{PRESC}+1)] \times t_{\text{I2CCLK}}\} - 1 \leq \text{SCLDEL}$$

t_r および $t_{\text{SU;DAT}}$ の標準値については、[表 80: I2C-SMBUS 仕様のデータのセットアップおよびホールド時間](#)を参照してください。

使用される SDA および SCL 遷移時間の値は、アプリケーションの値です。標準から最大値を使用すると、SDADEL と SCLDEL の計算の制約が増えますが、アプリケーションにかかわらず、この機能を使用できます。

表 80. I²C-SMBUS 仕様のデータのセットアップおよびホールド時間

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モードプラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
$t_{HD;DAT}$	データホールド時間	0	-	0	-	0	-	0.3	-	μs
$t_{VD;DAT}$	データ有効時間	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	データセットアップ時間	250	-	100		50		250		ns
t_r	SDA および SCL 信号の立ち上がり時間	-	1000		300	-	120	-	1000	
t_f	SDA および SCL 信号の立ち下がり時間	-	300		300	-	120	-	300	

また、マスタモードでは、I2C_TIMINGR レジスタの PRESC[3:0]、SCLH[7:0]、および SCLL[7:0] ビットをプログラムすることによって、SCL クロックのハイおよびローレベルを設定する必要があります。

- SCL 立ち下がりエッジが内部で検出されると、SCL 出力をリリースする前に遅延が挿入されます。この遅延は、 $t_{SCLL} = (SCLL+1) \times t_{PRESC}$ であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。
 t_{SCLL} は、SCL ロー時間 t_{LOW} に影響を与えます。
- SCL 立ち上がりエッジが内部で検出されると、SCL 出力を強制的にローレベルにする前に遅延が挿入されます。この遅延は、 $t_{SCLH} = (SCLH+1) \times t_{PRESC}$ であり、 $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$ です。
 t_{SCLH} は、SCLハイ時間 t_{HIGH} に影響を与えます。

詳細については、セクション [I2C マスタ初期化](#) を参照してください。

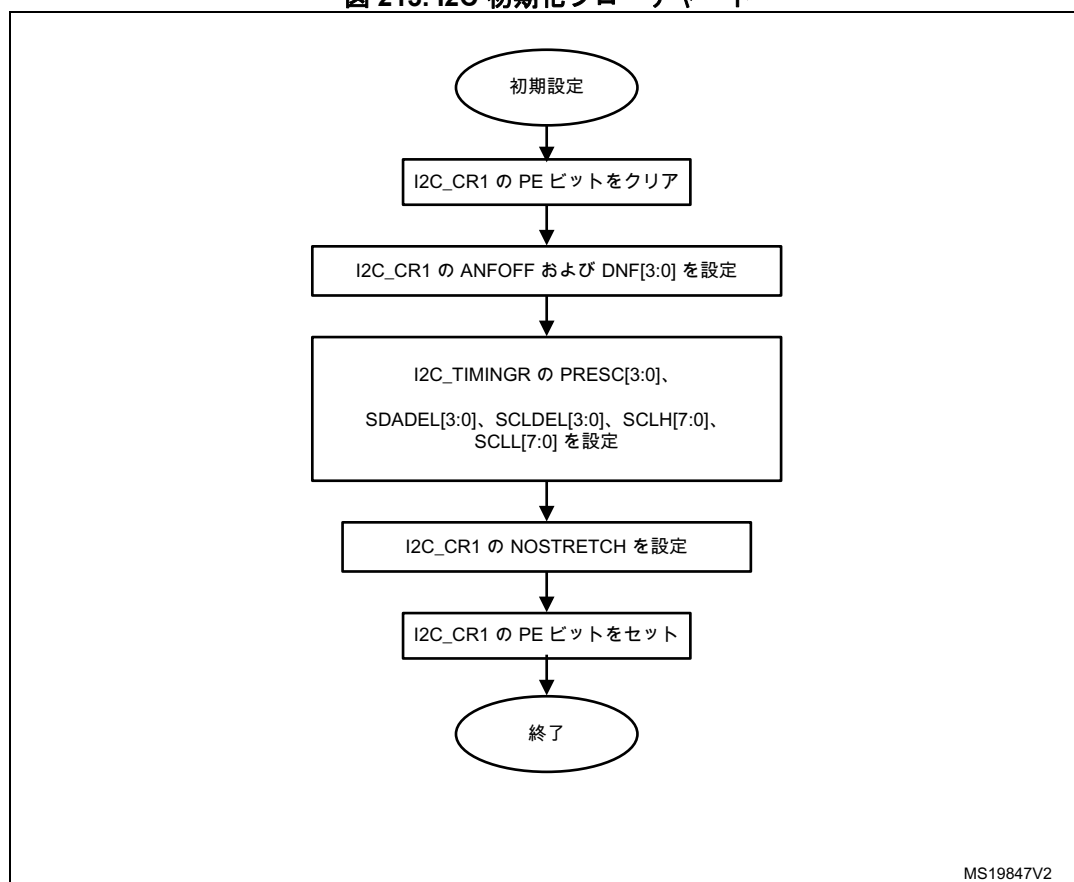
注意： I2C が有効なときには、タイミング構成の設定はできません。

ペリフェラルを有効にする前に、I2C スレーブ NOSTRETCH モードも設定する必要があります。詳細については、[I2C スレーブ初期化](#) を参照してください。

注意： I2C が有効なときには、NOSTRETCH 構成の変更はできません。



図 213. I2C 初期化フローチャート



25.4.6 ソフトウェアリセット

ソフトウェアリセットを行うには、I2C_CR1 レジスタの PE ビットをクリアします。その場合、I2C のライン SCL および SDA がリリースされます。内部状態マシンがリセットされ、通信制御ビットとステータスビットがリセット値に戻ります。構成レジスタは影響を受けません。

影響を受けるレジスタのビットは、以下のとおりです：

1. I2C_CR2 レジスタ：START、STOP、NACK
2. I2C_ISR レジスタ：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO、OVR

SMBus 機能がサポートされるときには、以下も影響を受けます：

1. I2C_CR2 レジスタ：PECBYTE
2. I2C_ISR レジスタ：PECERR、TIMEOUT、ALERT

ソフトウェアリセットを実行するためには、PE は少なくとも 3 APB クロックサイクルの間、ローに保たなければなりません。このためには、次のソフトウェアシーケンスを書き込みます：- PE=0 を書き込む - PE=0 を確認する - PE=1 を書き込む

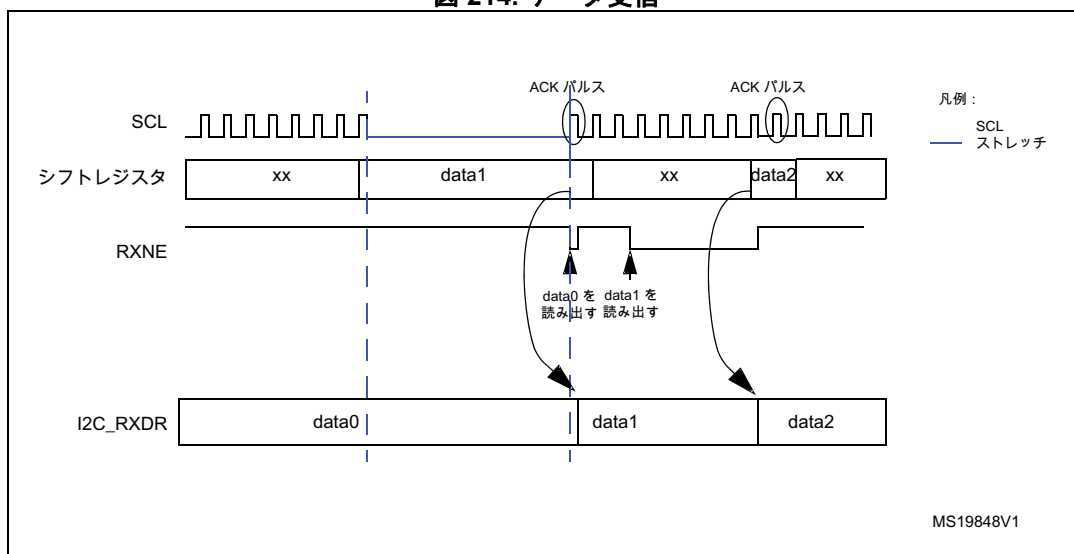
25.4.7 データ転送

データ転送は、送受信データレジスタとシフトレジスタを通じて管理されます。

受信

SDA 入力はシフトレジスタに送られます。8 番目の SCL パルスの後（完全なデータバイトの受信後）、シフトレジスタは、I2C_RXDR レジスタが空の場合（RXNE=0）、このレジスタにコピーされます。RXNE=1 の場合、すなわち、前に受信されたデータバイトがまだ読み出されていなかった場合、SCL ラインは I2C_RXDR が読み出されるまでストレッチされます。ストレッチは、8 番目と 9 番目の SCL パルスの間（確認応答パルスの前）に挿入されます。

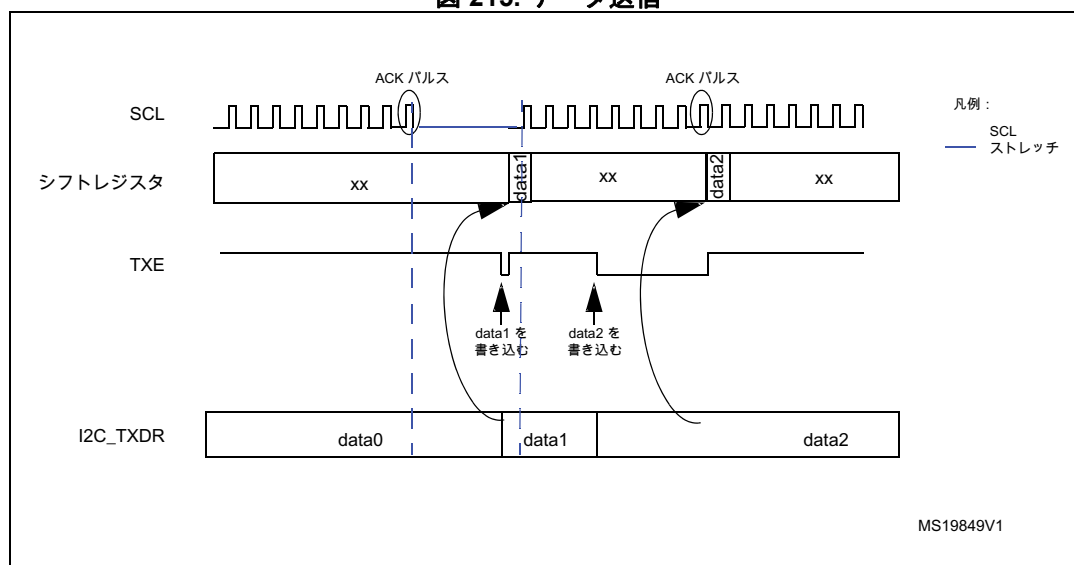
図 214. データ受信



送信

I2C_TXDR レジスタが空（TXE=0）でない場合、その内容が 9 番目の SCL パルス（確認応答パルス）の後、シフトレジスタにコピーされます。次に、シフトレジスタの内容が SDA ラインにシフトアウトされます。TXE=1 の場合、すなわち、I2C_TXDR にデータがまだ書き込まれていない場合、SCL ラインは I2C_TXDR に書き込まれるまでストレッチされます。ストレッチは、9 番目の SCL パルスの後で行われます。

図 215. データ送信



ハードウェア転送管理

次のようにさまざまなモードでバイト転送を管理し、通信をクローズするために、I2C にはハードウェアにバイトカウンタが組み込まれています。

- マスタモードでの NACK、STOP、および ReSTART 生成
- スレーブレシーバモードでの ACK 制御
- SMBus 機能がサポートされているときの PEC 生成/確認

バイトカウンタは、マスタモードでは常に使用されます。デフォルトでは、スレーブモードでは無効ですが、I2C_CR2 レジスタの SBC (スレーブバイト制御) ビットをセットすることによって、ソフトウェアにより有効にできます。

転送されるバイト数は、I2C_CR2 レジスタの NBYTES[7:0] ビットフィールドでプログラムされます。転送バイト数 (NBYTES) が 255 より大きい場合、またはレシーバが受信データバイトの確認応答値を制御したい場合には、I2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。このモードでは、NBYTES でプログラムされたバイト数が転送されると、TCR フラグがセットされ、TCIE がセットされている場合は割り込みが生成されます。SCL は、TCR フラグがセットされている間、ストレッチされます。TCR は、NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

NBYTES カウンタに最後のバイト数が再ロードされたときには、RELOAD ビットがクリアされる必要があります。

マスタモードで RELOAD=0 のときには、カウンタは 2 つのモードで使用できます：

- **自動終了モード** (I2C_CR2 レジスタの AUTOEND = "1")。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、マスタは STOP コンディションを自動的に送信します。
- **ソフトウェア終了モード** (I2C_CR2 レジスタの AUTOEND = 0)。このモードでは、NBYTES[7:0] ビットフィールドでプログラムされたバイト数が転送されると、ソフトウェアアクションが求められます。TC フラグがセットされ、TCIE ビットがセットされている場合は割り込みが生成されます。SCL 信号は、TC フラグがセットされている間、ストレッチされます。TC フラグは、I2C_CR2 レジスタの START または STOP ビットがセットされたときに、ソフトウェアによってクリアされます。マスタが RESTART コンディションを送信したいときには、このモードを使用する必要があります。

注意： *AUTOEND* ビットは、*RELOAD* ビットがセットされているときには効果がありません。

表 81. I2C 設定表

機能	SBC ビット	RELOAD ビット	AUTOEND ビット
マスタ Tx/Rx NBYTES + STOP	x	0	1
マスタ Tx/Rx + NBYTES + RESTART	x	0	0
スレーブ Tx/Rx すべての受信バイトに ACK	0	x	x
スレーブ Rx および ACK 制御	1	1	x

25.4.8 I2C スレーブモード

I2C スレーブ初期化

スレーブモードで動作するには、少なくとも 1 つのスレーブアドレスを有効にする必要があります。2 つのレジスタ I2C_OAR1 と I2C_OAR2 を使用して、スレーブ専用アドレス OA1 および OA2 をプログラムできます。

- OA1 は、I2C_OAR1 レジスタの OA1MODE ビットをセットすることによって、7 ビットモード（デフォルト）または 10 ビットアドレッシングモードに設定できます。

OA1 を有効にするには、I2C_OAR1 レジスタの OA1EN ビットをセットします。

- 追加のスレーブアドレスが必要な場合は、2 番目のスレーブアドレス OA2 を設定できます。I2C_OAR2 レジスタの OA2MSK[2:0] ビットを設定することによって、最大 7 つの OA2 LSB をマスクできます。したがって、OA2MSK が 1 から 6 まで設定された場合、OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6]、または OA2[7] のみが受信アドレスと比較されます。OA2MSK が 0 に等しくなくなるとすぐに、OA2 のアドレスコンパレータは、確認応答されない I2C 予約済みアドレス（0000 XXX および 1111 XXX）を除外します。OA2MSK=7 の場合、受信されたすべてのアドレスが確認応答されます（予約済みアドレスを除く）。OA2 は常に 7 ビットアドレスです。

これらの予約済みアドレスは、特定のイネーブルビットによって有効化された場合、I2C_OAR1 または I2C_OAR2 レジスタが OA2MSK=0 でプログラムされた場合、確認応答できます。

OA2 を有効にするには、I2C_OAR2 レジスタの OA2EN ビットをセットします。

- 同報アドレスは、I2C_CR1 レジスタの GCEN ビットをセットすることで有効になります。

I2C が有効アドレスの 1 つによって選択されると、ADDR 割り込みステータスフラグがセットされ、ADDRIE ビットがセットされている場合は割り込みが生成されます。

デフォルトでは、スレーブはクロックストレッチ機能を使用し、必要なときには、ソフトウェアアクションを実行するために、SCL 信号をローレベルでストレッチすることを意味します。マスタがクロックストレッチをサポートしない場合、I2C_CR1 レジスタの NOSTRETCH=1 で I2C を設定する必要があります。

ADDR 割り込みの受信後、いくつかのアドレスが有効な場合は、I2C_ISR レジスタの ADDCODE[6:0] ビットを読み出して、一致するアドレスを確認する必要があります。転送方向を知るために、DIR フラグも確認する必要があります。

スレーブクロックストレッチ (NOSTRETCH = 0)

デフォルトモードでは、I2C スレーブは次の状況で SCL クロックをストレッチします：

- ADDR フラグがセットされると：受信アドレスは有効なスレーブアドレスの 1 つと一致します。このストレッチは、ADDRCF ビットをセットすることによりソフトウェアによって ADDR フラグがクリアされたときにリリースされます。
- 送信時、前のデータ送信が完了し、新しいデータが I2C_TXDR レジスタに書き込まれなかった場合、または ADDR フラグがクリアされたときに (TXE=1)、最初のデータバイトが書き込まれていなかった場合。このストレッチは、データが I2C_TXDR レジスタに書き込まれたときにリリースされます。
- 受信時、I2C_RXDR レジスタがまだ読み出されておらず、新しいデータ受信が完了したとき。このストレッチは、I2C_RXDR が読み出されたときにリリースされます。
- スレーブバイト制御モードおよび再ロードモード (SBC=1 および RELOAD=1) で TCR = 1 のとき、すなわち、最後データバイトが転送されたとき。このストレッチは、NBYTES[7:0] フィールドにゼロ以外の値を書き込むことによって TCR がクリアされたときにリリースされます。
- SCL 立ち下がりエッジの検出後、I2C は、 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCCLK}$ の間、SCL ローをストレッチします。

クロックストレッチなしのスレーブ (NOSTRETCH = 1)

I2C_CR1 レジスタの NOSTRETCH = 1 のとき、I2C スレーブは SCL 信号をストレッチしません。

- ADDR フラグがセットされている間、SCL クロックはストレッチされません。
- 送信時、転送に対応する最初の SCL パルスが発生する前に、I2C_TXDR レジスタにデータが書き込まれる必要があります。そうでない場合、アンダーランが発生し、I2C_ISR レジスタで OVR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。OVR フラグは、最初のデータ送信が開始し、STOPF ビットがまだセットされている (クリアされていない) ときにもセットされます。したがって、次の転送で送信される最初のデータを書き込んだ後でのみ、前に転送の STOPF フラグをクリアすることによって、送信される最初のデータについても、OVR ステータスが提供されることを確実にできます。
- 受信時、次のデータバイトの 9 番目の SCL パルス (ACK パルス) が発生する前に、I2C_RXDR レジスタからデータが読み出される必要があります。そうでない場合、オーバーランが発生し、I2C_ISR レジスタの OVR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

スレーブバイト制御モード

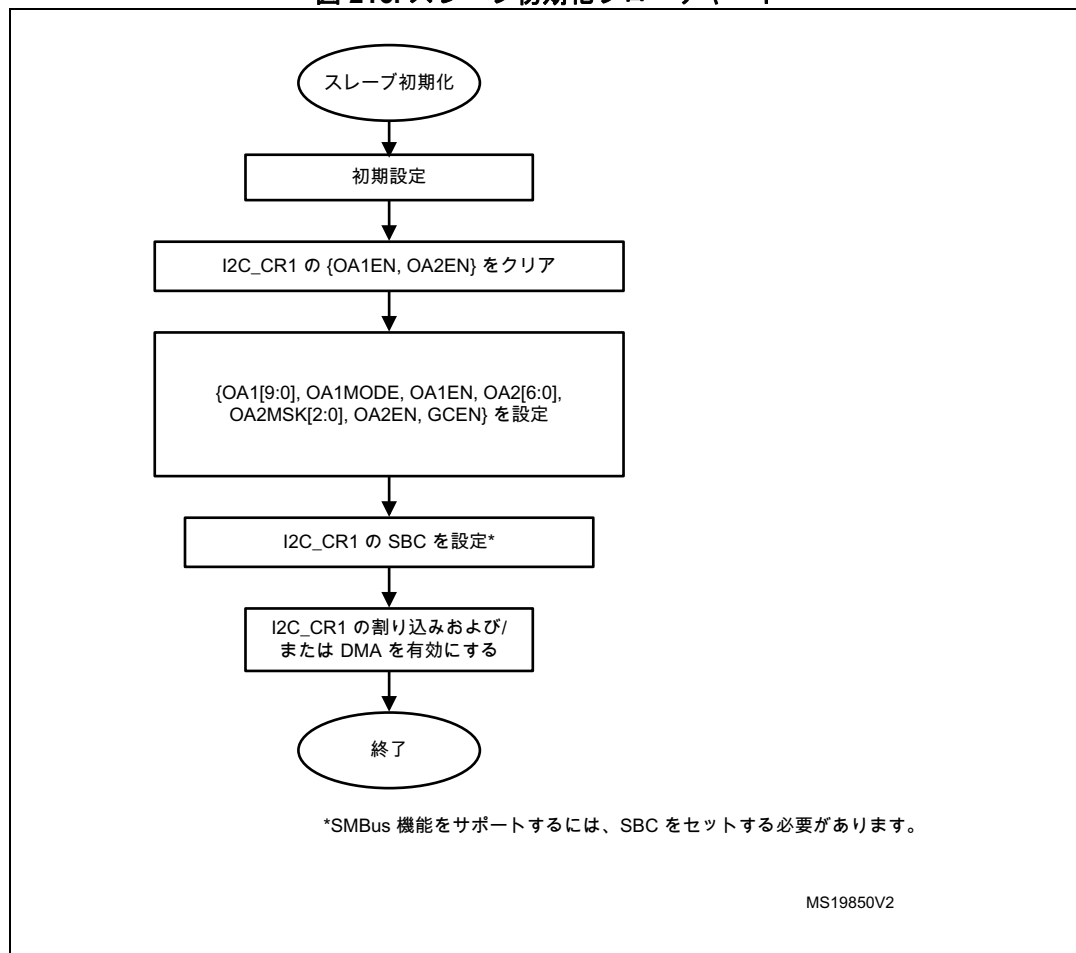
スレーブ受信モードでバイト ACK 制御を可能にするためには、I2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。これは、SMBus 標準に準拠する必要があります。

スレーブ受信モードでバイト ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD=1)。各バイトの制御を得るには、ADDR 割り込みサブルーチンで NBYTES を 0x1 に初期化し、各受信バイト後に 0x1 に再ロードする必要があります。バイトが受信されると、TCR ビットがセットされ、8 番目と 9 番目の SCL パルスの間で、SCL 信号ローをストレッチします。I2C_RXDR レジスタからデータを読み出すことができ、その後、I2C_CR2 レジスタの ACK ビットを設定することによって、確認応答するかどうかを決定できます。SCL ストレッチは、NBYTES をゼロ以外の値にプログラムすることによってリリースされ、確認応答または非確認応答が送信され、次のバイトを受信できます。

NBYTES に 0x1 より大きい値をロードでき、この場合、受信フローは NBYTES データ受信、継続します。

- 注：** SBC ビットは、I2C が無効なとき、またはスレーブがアドレス指定されていないとき、または ADDR=1 のときに設定する必要があります。
- RELOAD ビットの値は、ADDR=1 のとき、または TCR=1 のときに変更できます。**
- 注意：** スレーブバイト制御モードは、NOSTRETCH モードと互換性がありません。NOSTRETCH=1 のときに SBC をセットすることはできません。

図 216. スレーブ初期化フローチャート



コード例については、「付録」セクション [A.14.3：スレーブモードで設定された I2C のコード例](#) を参照してください。

スレーブトランスミッタ

I2C_TXDR レジスタが空になると、送信割り込みステータス (TXIS) が生成されます。I2C_CR1 レジスタの TXIE ビットがセットされている場合は、割り込みが生成されます。

TXIS ビットは、I2C_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

NACK が受信されると、I2C_ISR レジスタの NACKF ビットがセットされ、I2C_CR1 レジスタの NACKIE ビットがセットされていた場合は割り込みが生成されます。マスタが STOP または RESTART コンディションを実行できるように、スレーブは SCL および SDA ラインを自動的にリリースします。TXIS ビットは、NACK 受信時にはセットされません。

STOP が受信され、I2C_CR1 レジスタの STOPIE ビットがセットされると、I2C_ISR レジスタの STOPF フラグがセットされ、割り込みが生成されます。ほとんどのアプリケーションでは、SBC は通常、0 にプログラムされます。この場合、スレーブアドレスが受信されたときに (ADDR=1)、TXE = 0 であった場合、I2C_TXDR レジスタの内容を最初のデータバイトとして送信するか、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C_TXDR レジスタをフラッシュするかを選択できます。

スレーブバイト制御モード (SBC=1) では、送信バイト数をアドレス一致割り込みサブルーチンの NBYTES でプログラムする必要があります (ADDR=1)。この場合、転送中の TXIS イベントの数は、NBYTES でプログラムされた値に対応します。

注意： *NOSTRETCH=1 のとき、SCL クロックは ADDR フラグがセットされている間はストレッチされないため、最初のデータバイトをプログラムするために ADDR サブルーチンで I2C_TXDR レジスタの内容をフラッシュすることはできません。最初に送信されるデータバイトは、I2C_TXDR レジスタで前もってプログラムされている必要があります。*

- このデータは、前の送信メッセージの最後の TXIS イベントで書き込まれたデータでもかまいません。
- このデータバイトが送信データバイトでない場合、新しいデータバイトをプログラムするために TXE ビットをセットすることによって I2C_TXDR レジスタをフラッシュできます。アドレスの確認応答に続いて、最初のデータ送信が開始する前にこれらが実行されることを保証するためには、STOPF ビットのクリアは、これらのアクションの後でのみ行う必要があります。

最初のデータ送信が開始したときに STOPF がまだセットされていた場合、アンダーランエラーが生成されます (OVR フラグがセットされます)。

TXIS イベントが必要な場合 (送信割り込みまたは送信 DMA リクエスト)、TXIS イベントを生成するためには、TXE ビットに加えて TXIS ビットもセットする必要があります。

図 217. I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=0)

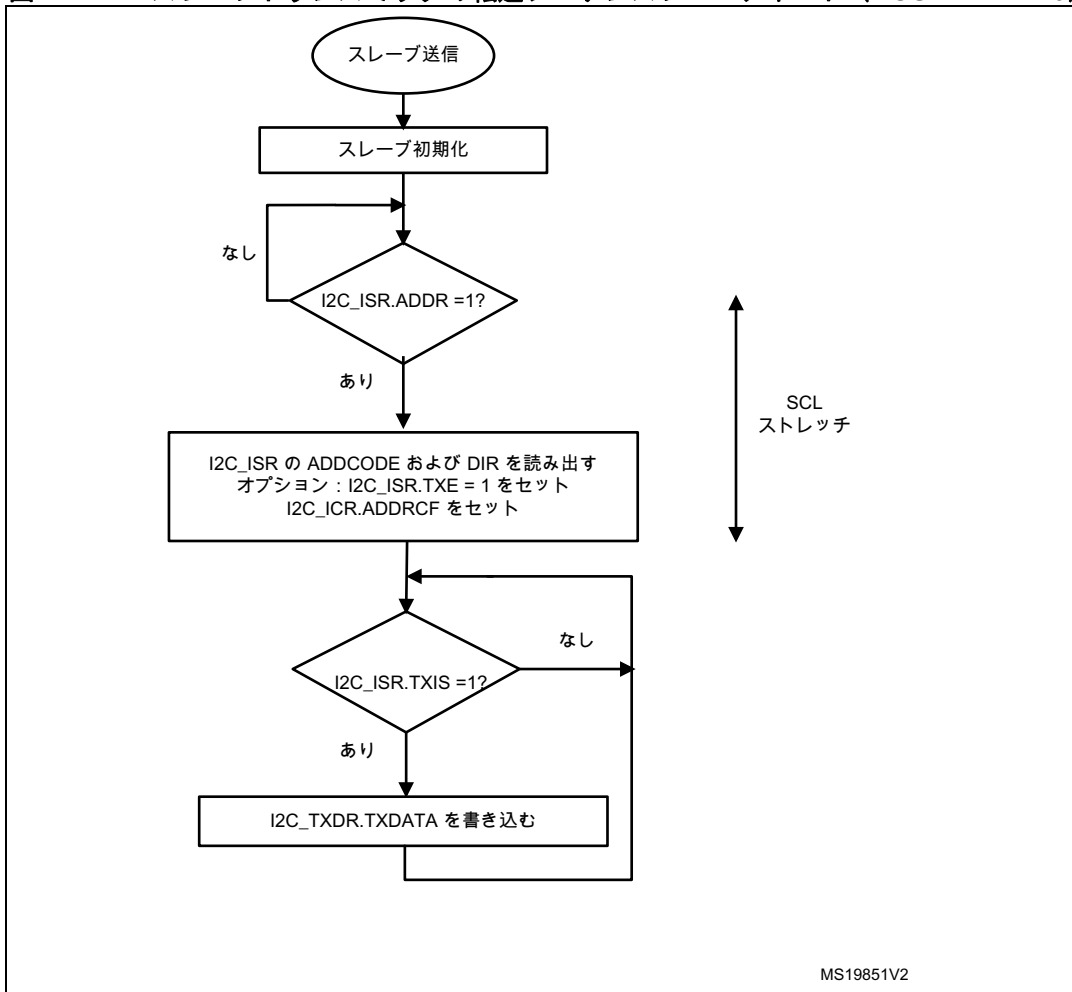


図 218. I2C スレーブトランスミッタの転送シーケンスフローチャート (NOSTRETCH=1)

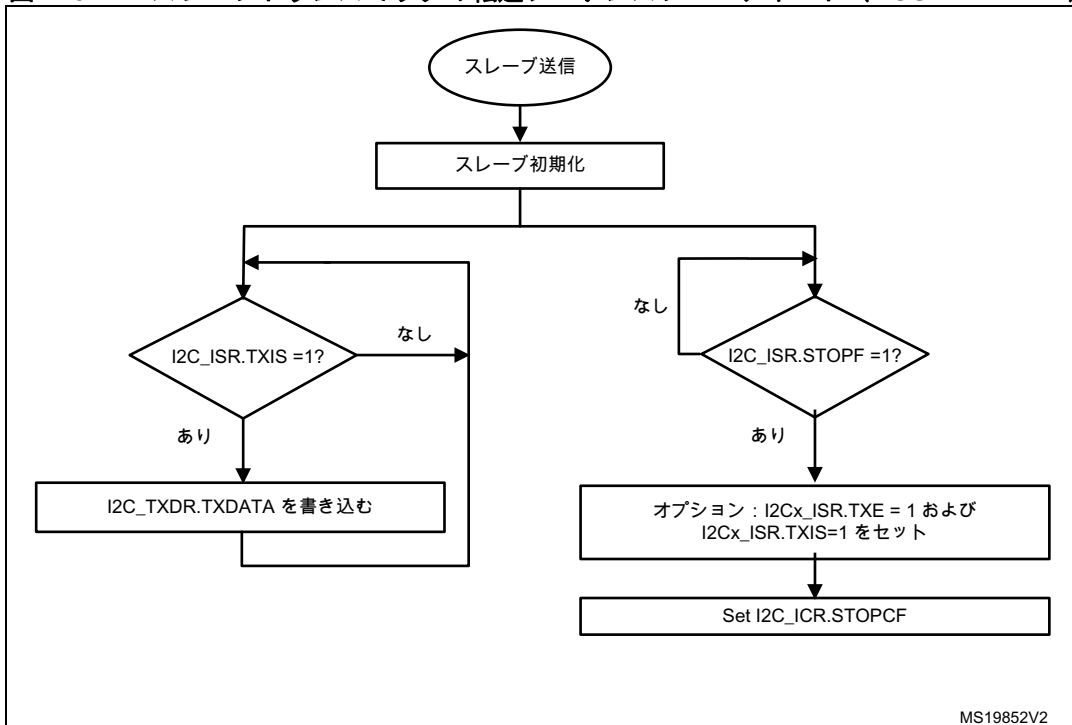
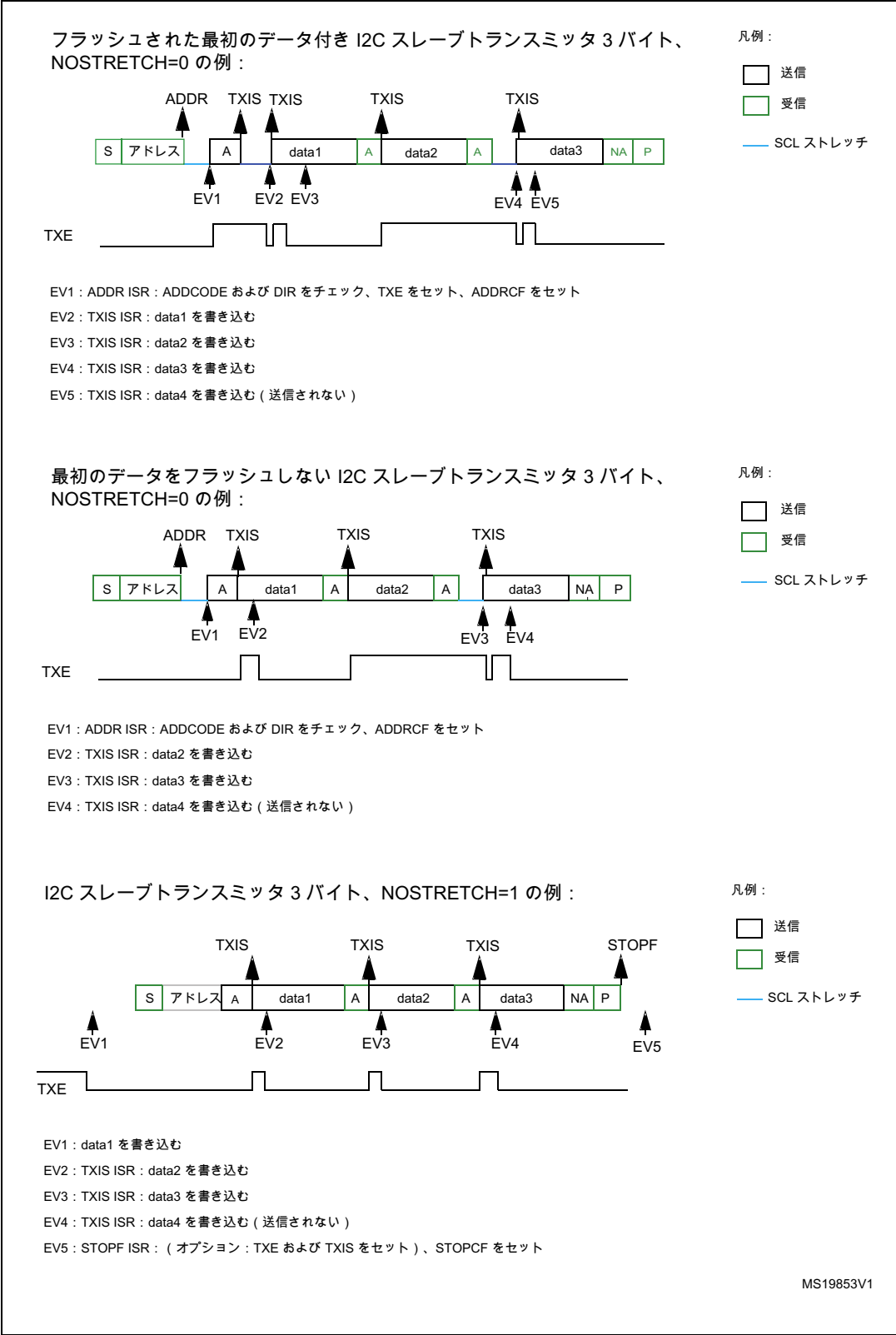


図 219. I2C スレーブトランスミッタの転送バス図



コード例については、「付録」セクション [A.14.6 : I2C スレーブトランスミッタのコード例](#) を参照してください。

スレーブレシーバ

I2C_RXDR がフルのときには、I2C_ISR の RXNE がセットされ、I2C_CR1 の RXIE がセットされている場合は割り込みが生成されます。RXNE は、I2C_RXDR が読み出されたときにクリアされます。

STOP が受信され、I2C_CR1 レジスタの STOPIE ビットがセットされると、I2C_ISR の STOPF がセットされ、割り込みが生成されます。

図 220. スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=0)

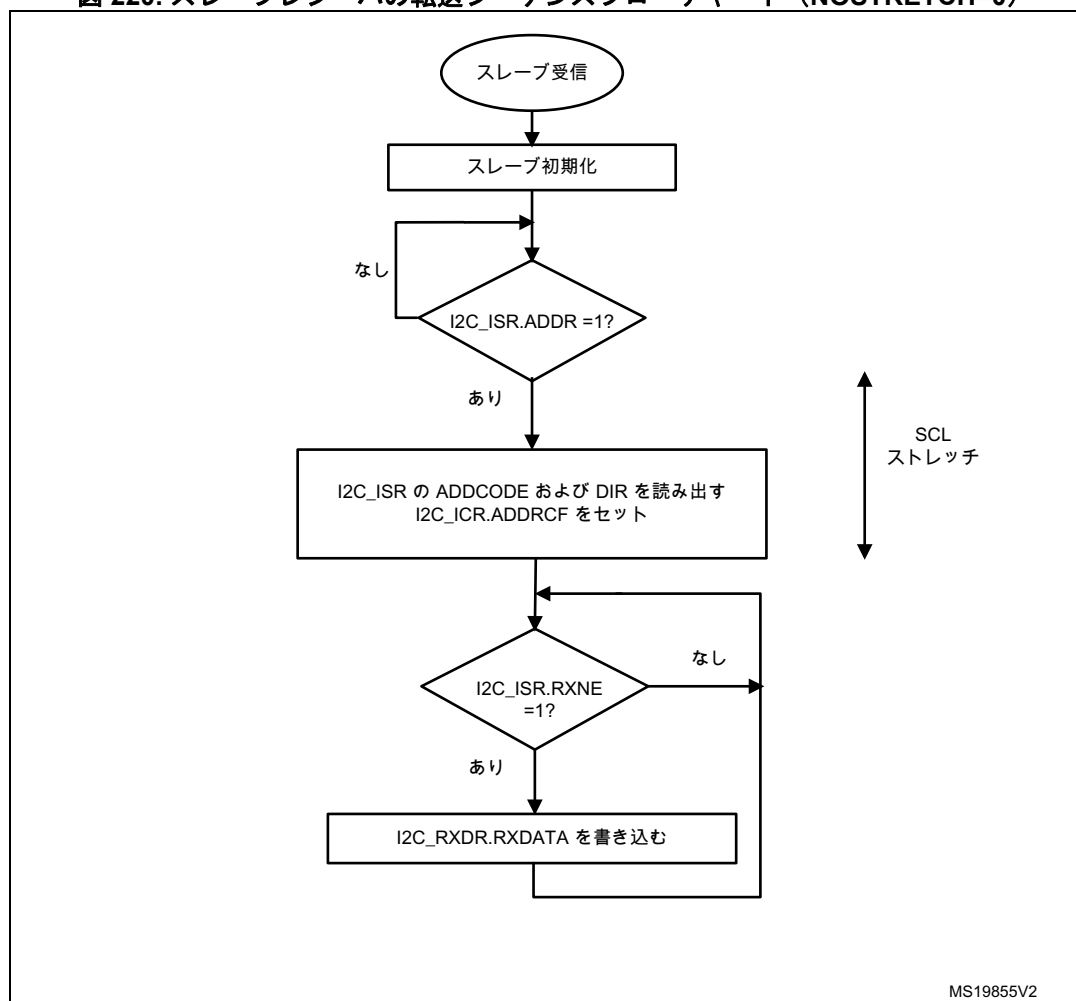


図 221. スレーブレシーバの転送シーケンスフローチャート (NOSTRETCH=1)

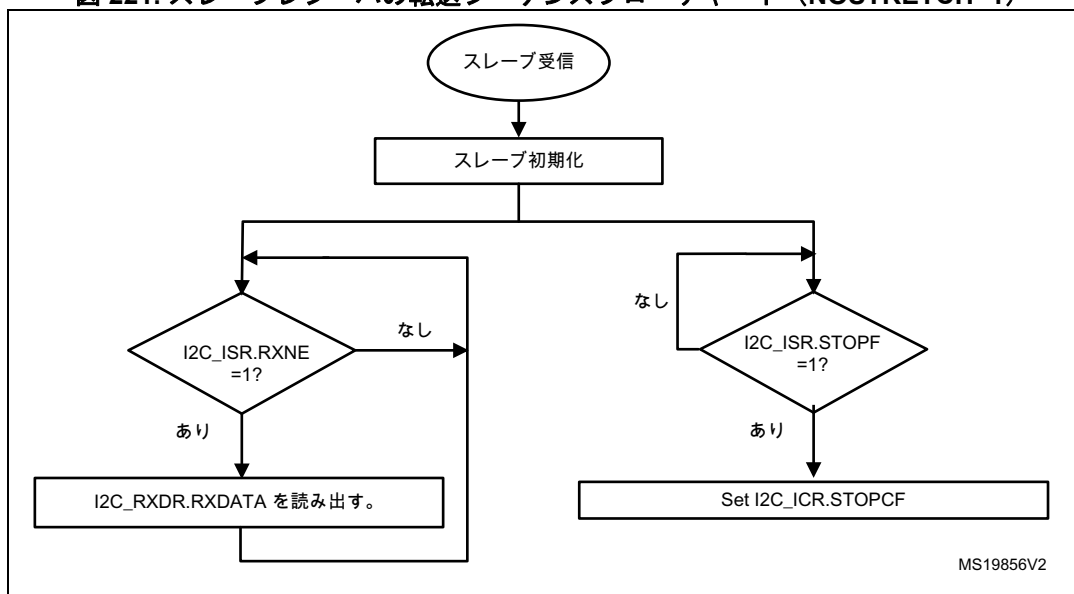
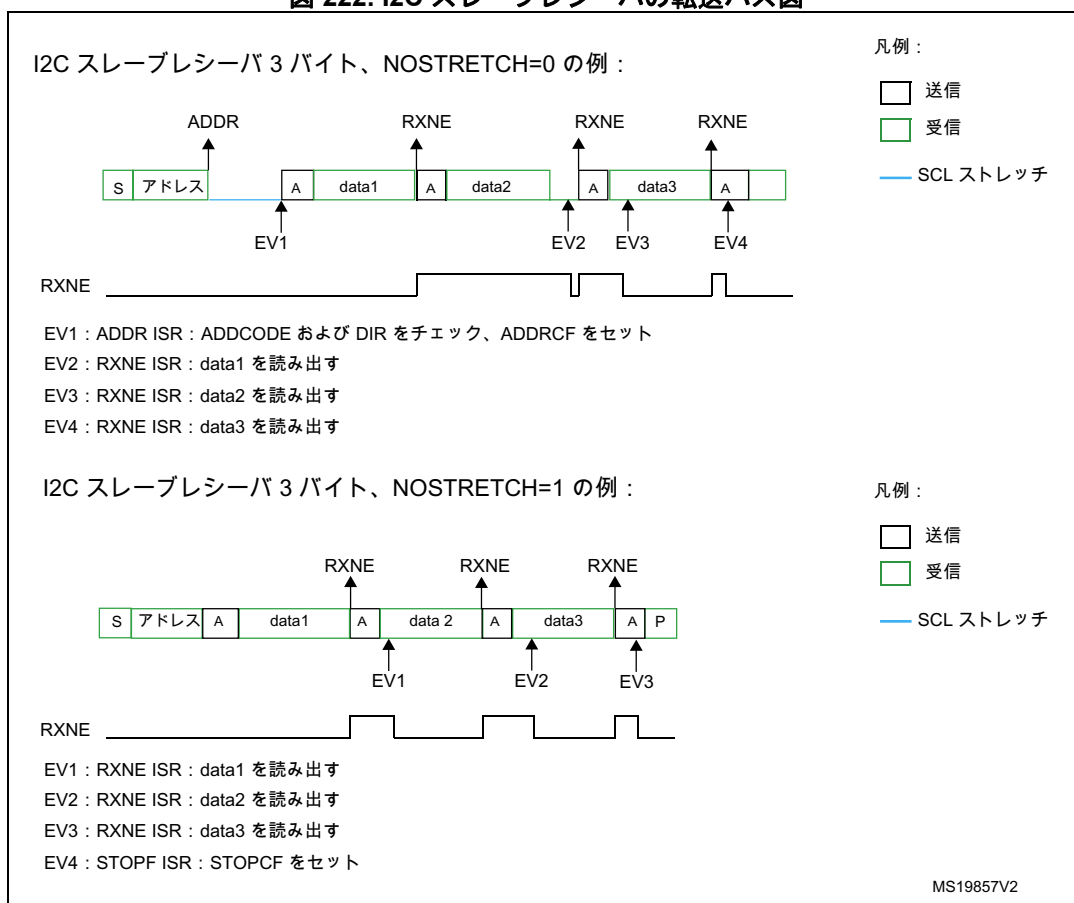


図 222. I2C スレーブレシーバの転送バス図



コード例については、「付録」セクション [A.14.7 : I2C スレーブレシーバのコード例](#) を参照してください。

25.4.9 I2C マスタモード

I2C マスタ初期化

ペリフェラルを有効にする前に、I2C_TIMINGR レジスタの SCLH および SCLL ビットをセットすることによって、I2C マスタクロックを設定する必要があります。

マルチマスタ環境とスレーブクロックストレッチをサポートするために、クロック同期メカニズムが実装されています。

クロック同期を可能にするために：

- クロックのローレベルは SCLL カウンタを使用してカウントされ、SCL ローレベル内部検出から開始されます。
- クロックのハイレベルは SCLH カウンタを使用してカウントされ、SCL ハイレベル内部検出から開始されます。

I2C は、SCL 立ち下がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延 t_{SYNC1} の後に SCL ローレベルを検出します。SCLL カウンタが I2C_TIMINGR レジスタの SCLL[7:0] ビットでプログラムされた値に達すると、I2C は SCL をハイレベルにリリースします。

I2C は、SCL 立ち上がりエッジ、SCL 入力ノイズフィルタ（アナログ + デジタル）、および I2CxCLK クロックとの SCL 同期に応じた遅延 t_{SYNC2} の後に SCL ハイレベルを検出します。SCLH カウンタが I2C_TIMINGR レジスタの SCLH[7:0] ビットでプログラムされた値に達すると、I2C は SCL をローレベルにします。

結果として、マスタクロック周期は次のとおりです：

$$t_{\text{SCL}} = t_{\text{SYNC1}} + t_{\text{SYNC2}} + \{[(\text{SCLH}+1) + (\text{SCLL}+1)] \times (\text{PRESC}+1) \times t_{\text{I2CCLK}}\}$$

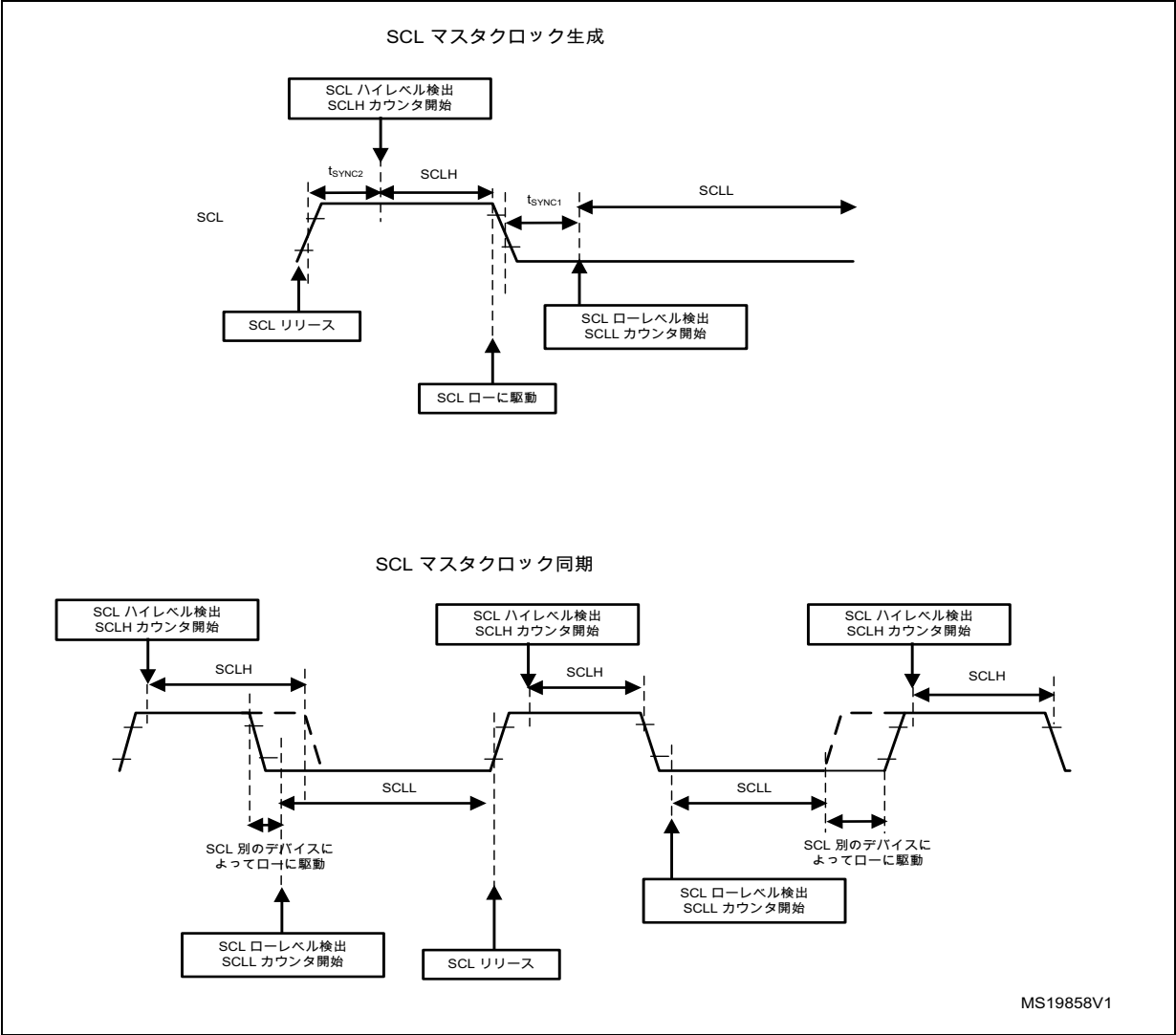
t_{SYNC1} の長さは、次のパラメータに依存します：

- SCL 立ち下がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延：DNF $\times t_{\text{I2CCLK}}$
- I2CCLK クロックとの SCL 同期による遅延（2 ～ 3 I2CCLK 周期）

t_{SYNC2} の長さは、次のパラメータに依存します：

- SCL 立ち上がり傾斜
- 有効な場合、アナログフィルタによる入力遅延。
- 有効な場合、デジタルフィルタによる入力遅延：DNF $\times t_{\text{I2CCLK}}$
- I2CCLK クロックとの SCL 同期による遅延（2 ～ 3 I2CCLK 周期）

図 223. マスタクロック生成



注意： I²C または SMBus 準拠のためには、マスタクロックは次のタイミングを満たす必要があります：

表 82. I²C-SMBUS 仕様のクロックタイミング

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モード プラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
f _{SCL}	SCL クロック周波数		100		400		1000		100	kHz
t _{HD:STA}	(反復) START コンディションの ホールド時間	4.0	-	0.6		0.26	-	4.0	-	μs
t _{SU:STA}	反復 START コンディションの セットアップ時間	4.7	-	0.6		0.26	-	4.7	-	μs
t _{SU:STO}	STOP コンディションのセット アップ時間	4.0	-	0.6		0.26	-	4.0	-	μs

表 82. I²C-SMBUS 仕様のクロックタイミング (続き)

記号	パラメータ	標準モード (Sm)		高速モード (Fm)		高速モード プラス (Fm+)		SMBUS		単位
		最小値	最大値	最小値	最大値	最小値	最大値	最小値	最大値	
t _{BUF}	STOP コンディションと START コンディションの間のバスフリー時間	4.7	-	1.3		0.5	-	4.7	-	μs
t _{LOW}	SCL クロックのロー周期	4.7	-	1.3		0.5	-	4.7	-	μs
t _{HIGH}	SCL クロックの周期	4.0	-	0.6		0.26	-	4.0	50	μs
t _r	SDA および SCL 信号の立ち上がり時間	-	1000	-	300		120	-	1000	ns
t _f	SDA および SCL 信号の立ち下がり時間	-	300	-	300		120	-	300	ns

注 : SCLL は、t_{BUF} および t_{SU:STA} タイミングの生成にも使用されます。
SCLH は、t_{HD:STA} および t_{SU:STO} タイミングの生成にも使用されます。
I2C_TIMINGR 設定と I2CCLK 周波数の例については、[セクション 25.4.10 : I2C_TIMINGR レジスタの設定例](#)を参照してください。

マスタ通信の初期化 (アドレスフェーズ)

通信を初期化するためには、I2C_CR2 レジスタでアドレス指定されたスレーブについて次のパラメータをプログラムする必要があります。

- アドレッシングモード (7 ビットまたは 10 ビット) : ADD10
- 送信されるスレーブアドレス : SADD[9:0]
- 転送方向 : RD_WRN
- 10 ビットアドレスが読み出される場合 : HEAD10R ビット。HEAD10R を設定して、完全なアドレスシーケンスが送信されなければならないか、ヘッダのみ (方向の変更の場合) かを示す必要があります。
- 転送されるバイト数 : NBYTES[7:0]。バイト数が 255 バイト以上の場合は、NBYTES[7:0] に 0xFF を書き込む必要があります。

次に、I2C_CR2 レジスタの START ビットをセットする必要があります。START ビットがセットされているとき、上記のすべてのビットを変更することはできません。

その場合、マスタは、バスがフリーである (BUSY = 0) ことを検出すると、t_{BUF} の遅延後に、自動的に START コンディションとスレーブアドレスを送信します。

アービトレーション喪失の場合、マスタはスレーブモードに自動的に切り替えて、スレーブとしてアドレス指定された場合は専用アドレスを確認応答できます。

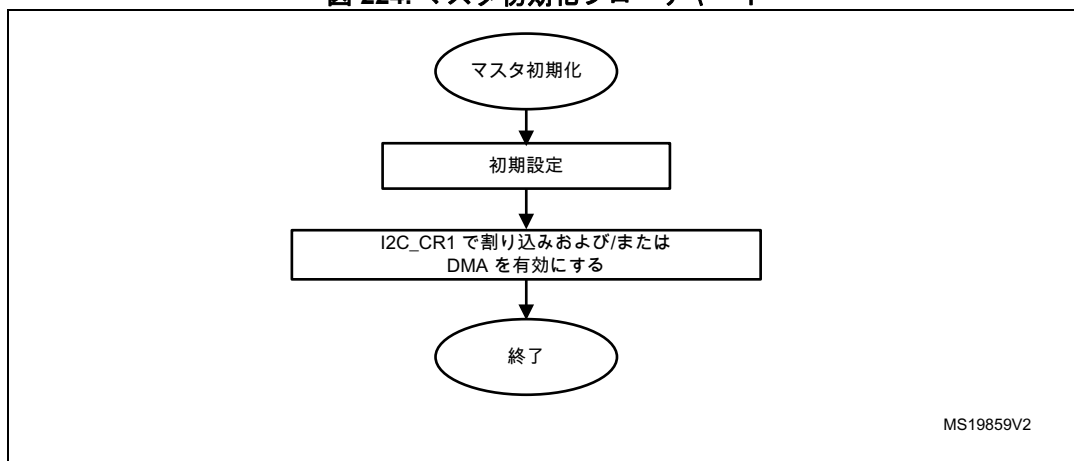
注 : START ビットは、スレーブアドレスがバスに送信されたとき、受信した確認応答値にかかわらず、ハードウェアによってリセットされます。START ビットは、アービトレーション喪失が発生した場合にも、ハードウェアによってリセットされます。START ビットがセットされているときに、I2C がスレーブとしてアドレス指定された場合 (ADDR=1)、I2C はスレーブモードに切り替わり、ADDRCF ビットがセットされたときに START ビットがクリアされます。

注 : 反復スタートコンディションにも同じ手順が適用されます。この場合、BUSY=1 です。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

図 224. マスタ初期化フローチャート

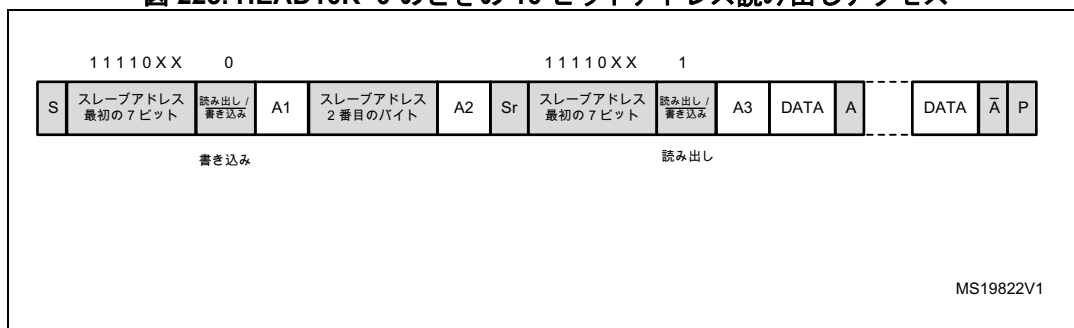


コード例については、「付録」セクション [A.14.1: 受信するためにマスタモードに設定された I2C のコード例](#)と「付録」セクション [A.14.2: 送信するためにマスタモードに設定された I2C のコード例](#)を参照してください。

10 ビットアドレススレーブをアドレス指定するマスタレシーバの初期化

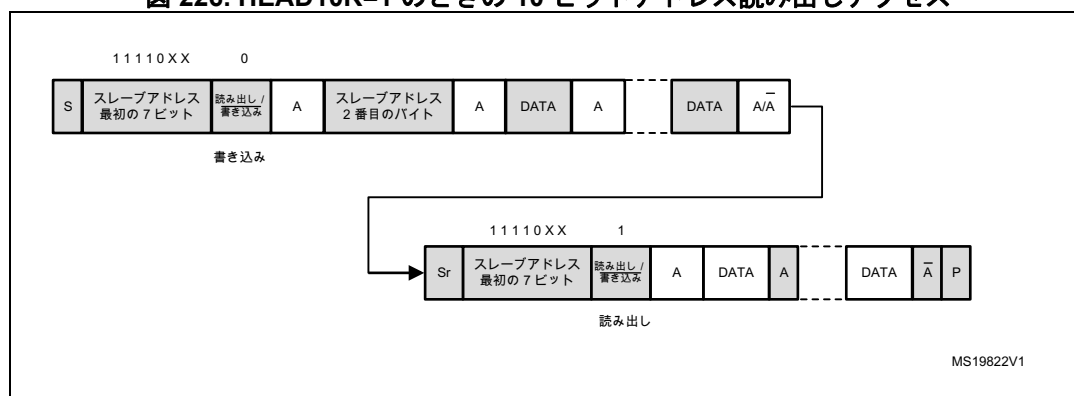
- スレーブアドレスが 10 ビット形式の場合、I2C_CR2 レジスタの HEAD10R ビットをクリアすることによって、完全な読み出しシーケンスを送信することができます。この場合、マスタは、START ビットがセットされた後、次のような完全なシーケンスを自動的に送信します：(Re) START + スレーブアドレス 10 ビットヘッダ書き込み + スレーブアドレスの 2 番目のバイト + RStart + スレーブアドレス 10 ビットヘッダ読み出し。

図 225. HEAD10R=0 のときの 10 ビットアドレス読み出しアクセス



- マスタが 10 ビットアドレススレーブをアドレス指定して、このスレーブアドレスにデータを送信した後、同じスレーブからデータを読み出す場合には、まず、マスタ送信フローが行われる必要があります。その場合、反復開始が、HEAD10R=1 で設定された 10 ビットスレーブアドレスでセットされます。この場合、マスタは次のシーケンスを送信します：ReStart + スレーブアドレス 10 ビットヘッダ読み出し

図 226. HEAD10R=1 のときの 10 ビットアドレス読み出しアクセス



マスタートランスマッタ

書き込み転送の場合、ACK が受信されたときの9 番目の SCL パルス後、各バイトの送信後にTXIS フラグがセットされます。

I2C_CR1 レジスタの TXIE ビットがセットされている場合、TXIS イベント時に割り込みが生成されます。このフラグは、I2C_TXDR レジスタに次に送信されるデータバイトが書き込まれると、クリアされます。

転送中の TXIS イベントの数は、NBYTES[7:0] でプログラムされた値に対応します。送信されるデータバイト数の合計が 255 より大きい場合、I2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

TXIS フラグは、NACK 受信時にはセットされません。

- RELOAD=0 でNBYTES データが転送されたとき：
 - － 自動終了モード (AUTOEND=1) では、STOP が自動的に送信されます。
 - － ソフトウェア終了モード (AUTOEND=0) では、TC フラグがセットされ、ソフトウェアアクションを実行するために SCL ラインがローでストレッチされます：

正しいスレーブアドレス設定と転送バイト数で I2C_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションがバスに送信されます。

I2C_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。
- NACK が受信された場合：TXIS フラグはセットされず、NACK 受信後、自動的に STOP コンディションが送信されます。I2C_ISR レジスタの NACKF フラグがセットされ、NACKIE ビットがセットされていた場合は割り込みが生成されます。

図 227. N≤255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート

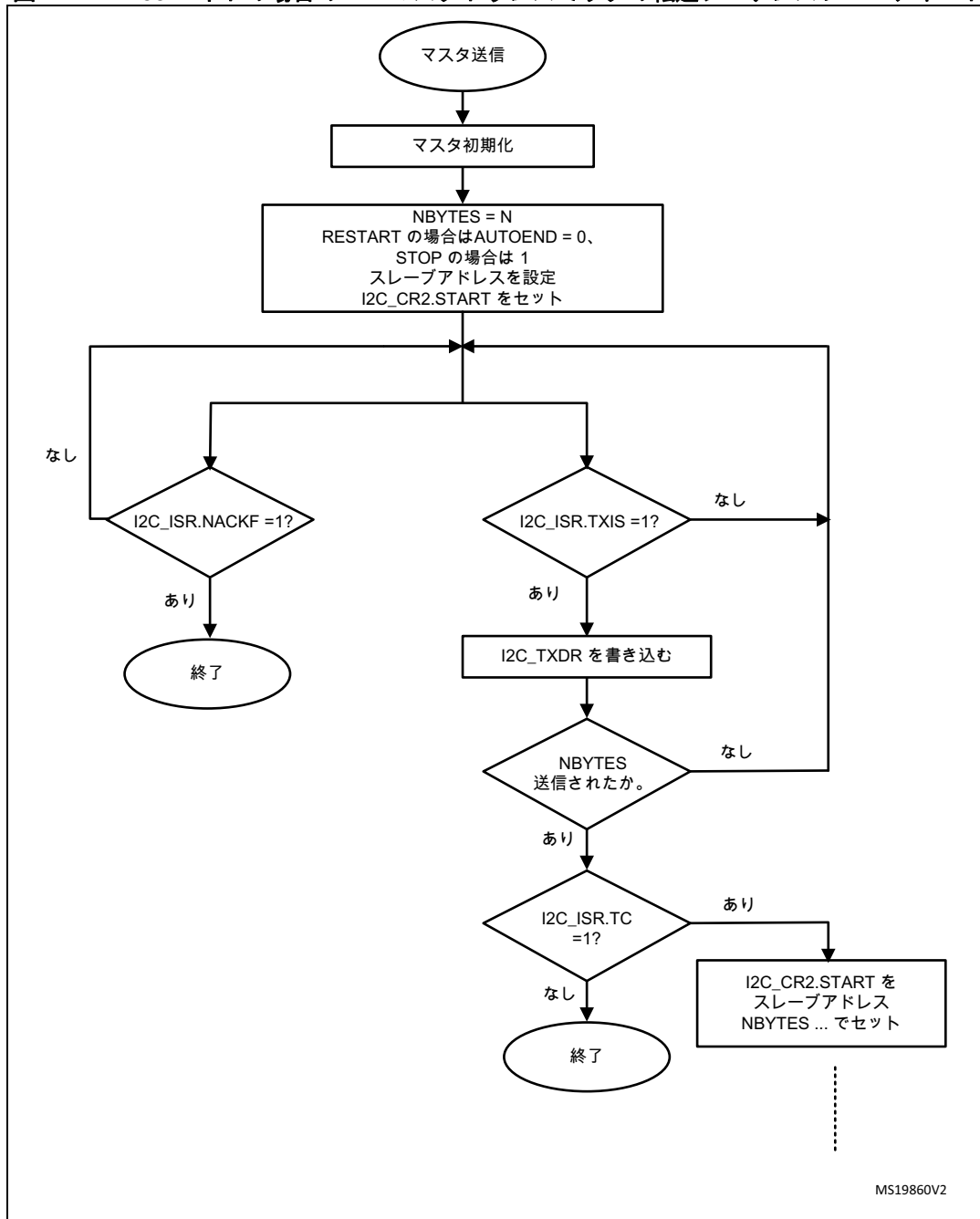


図 228. N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスフローチャート

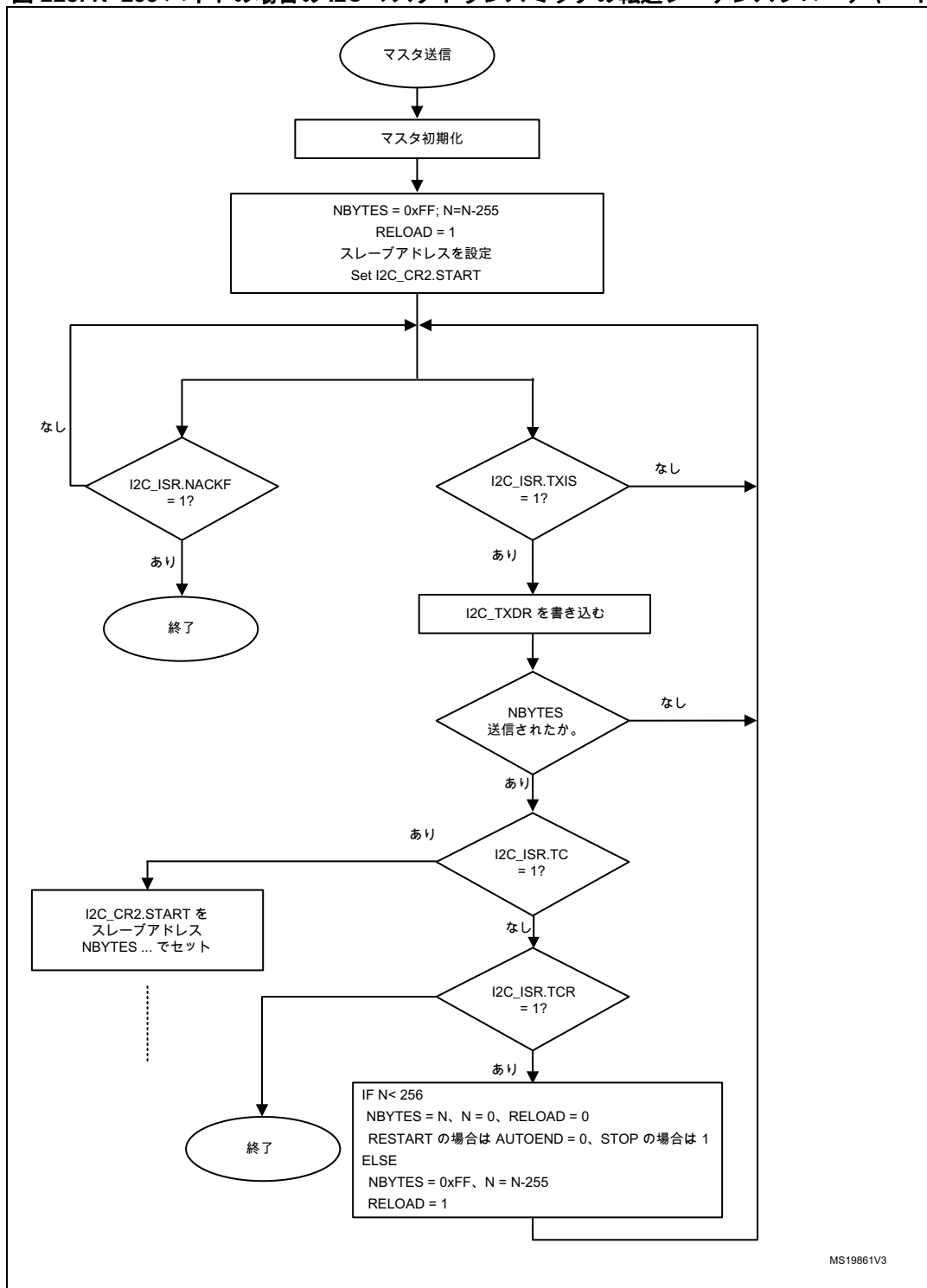
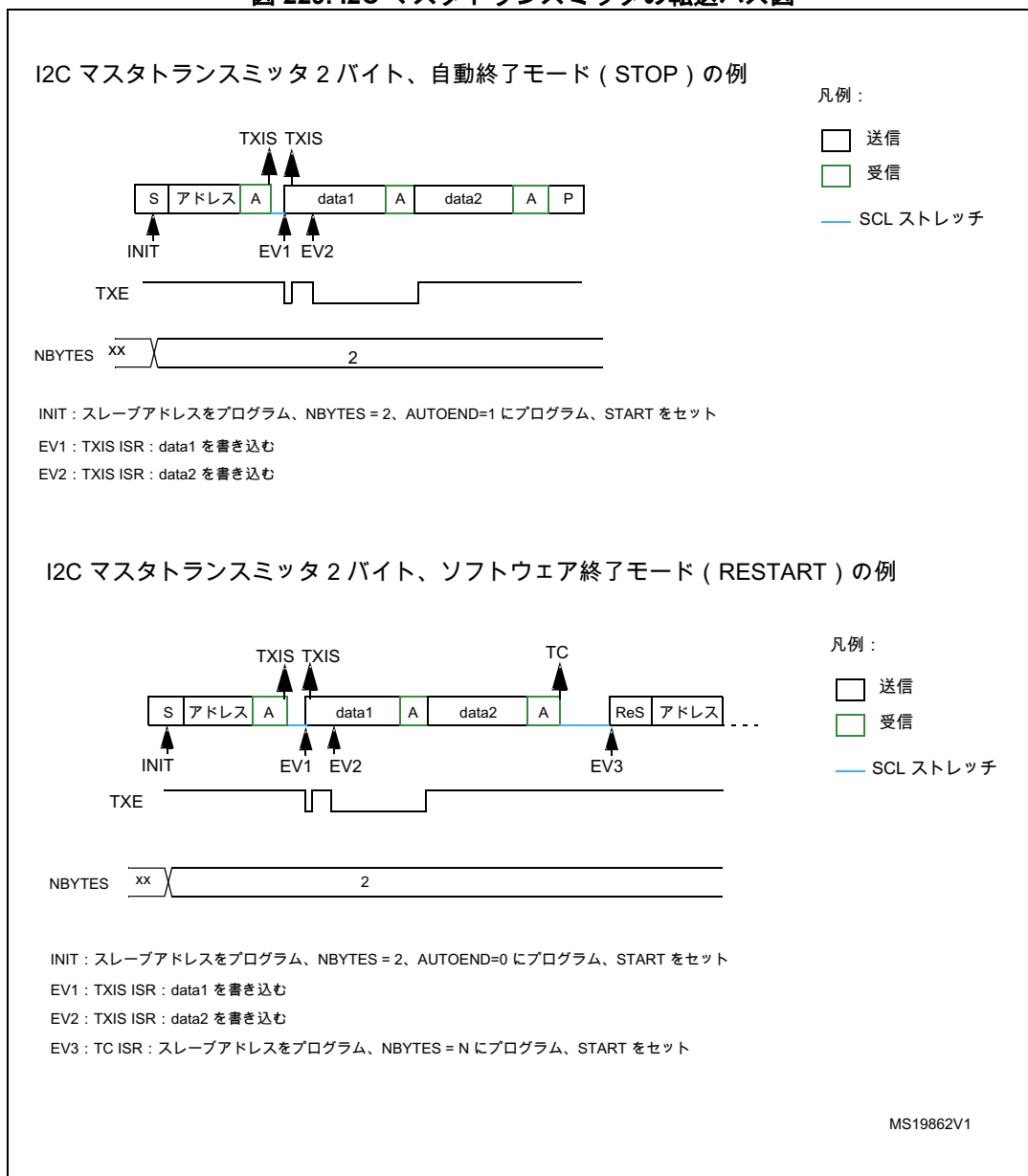


図 229. I2C マスタトランスミッタの転送バス図



コード例については、「付録」セクション [A.14.4 : I2C マスタトランスミッタのコード例](#)を参照してください。

マスタレシーバ

読み出し転送の場合、各バイトの受信後や8 番目の SCL パルス後に RXNE フラグがセットされます。I2C_CR1 レジスタの RXIE ビットがセットされている場合、RXNE イベント時に割り込みが生成されます。このフラグは、I2C_RXDR が読み出されたときにクリアされます。

受信されるデータバイト数の合計が 255 より大きい場合、I2C_CR2 レジスタの RELOAD ビットをセットすることによって、再ロードモードを選択する必要があります。この場合、NBYTES[7:0] データが転送されると、TCR フラグがセットされ、NBYTES[7:0] にゼロ以外の値が書き込まれるまで、SCL ラインはローでストレッチされます。

- RELOAD=0 でNBYTES[7:0] データが転送されたとき：
 - 自動終了モード (AUTOEND=1) では、最後の受信バイト後に NACK および STOP が自動的に送信されます。
 - ソフトウェア終了モード (AUTOEND=0) では、最後の受信バイト後に NACK が自動的に送信され、TC フラグがセットされ、ソフトウェアアクションを実行できるように、SCL ラインがローでストレッチされます。

正しいスレーブアドレス設定と転送バイト数で I2C_CR2 レジスタの START ビットをセットすることによって、RESTART コンディションをリクエストできます。START ビットをセットすると、TC フラグがクリアされ、START コンディションとスレーブアドレスがバスに送信されます。

I2C_CR2 レジスタの STOP ビットをセットすることによって、STOP コンディションをリクエストできます。STOP ビットをセットすると、TC フラグがクリアされ、STOP コンディションがバスに送信されます。

図 230. N ≤ 255 バイトの場合の I2C マスタレシーバの転送シーケンスフローチャート

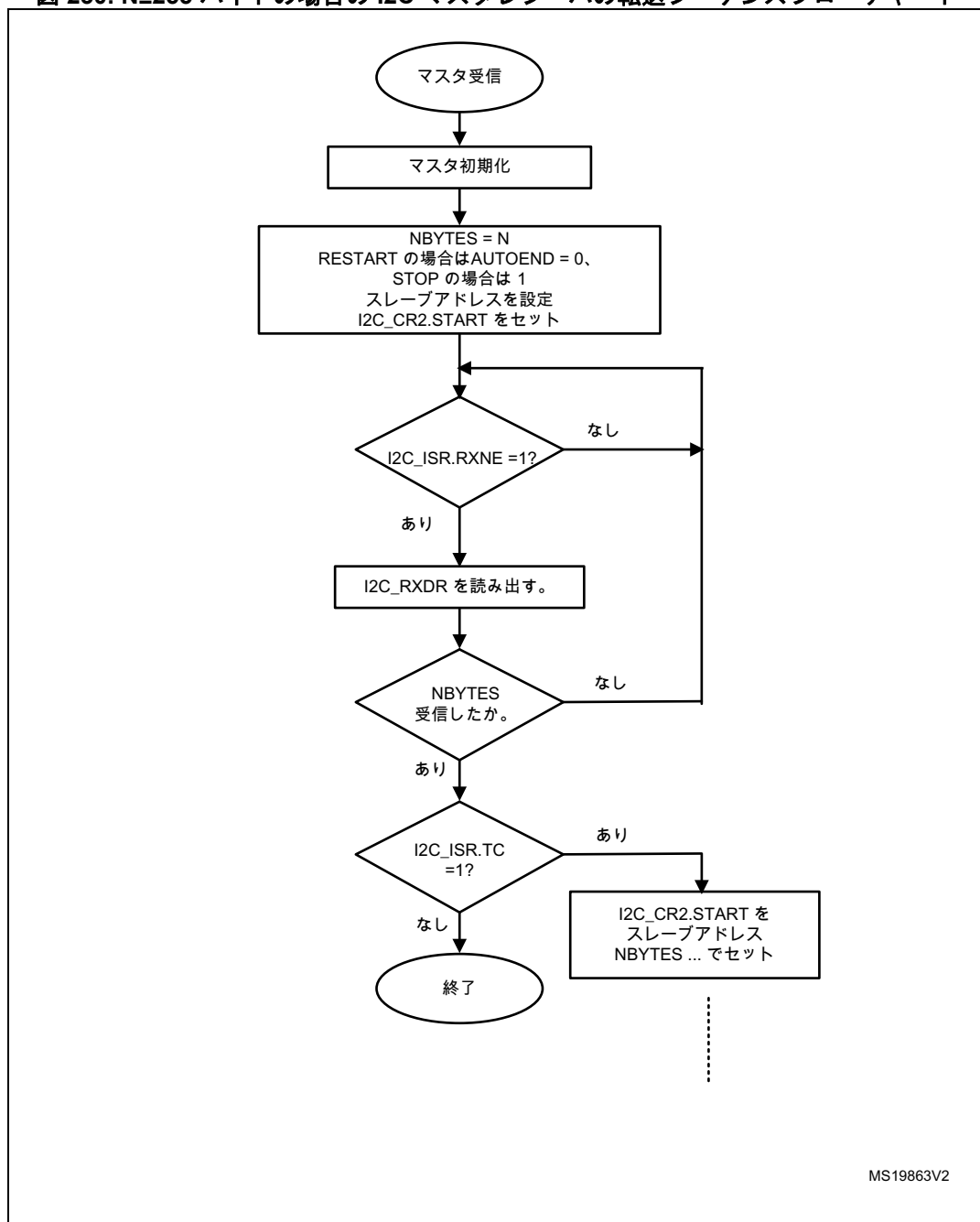


図 231. N>255 バイトの場合の I2C マスタレーバの転送シーケンスフローチャート

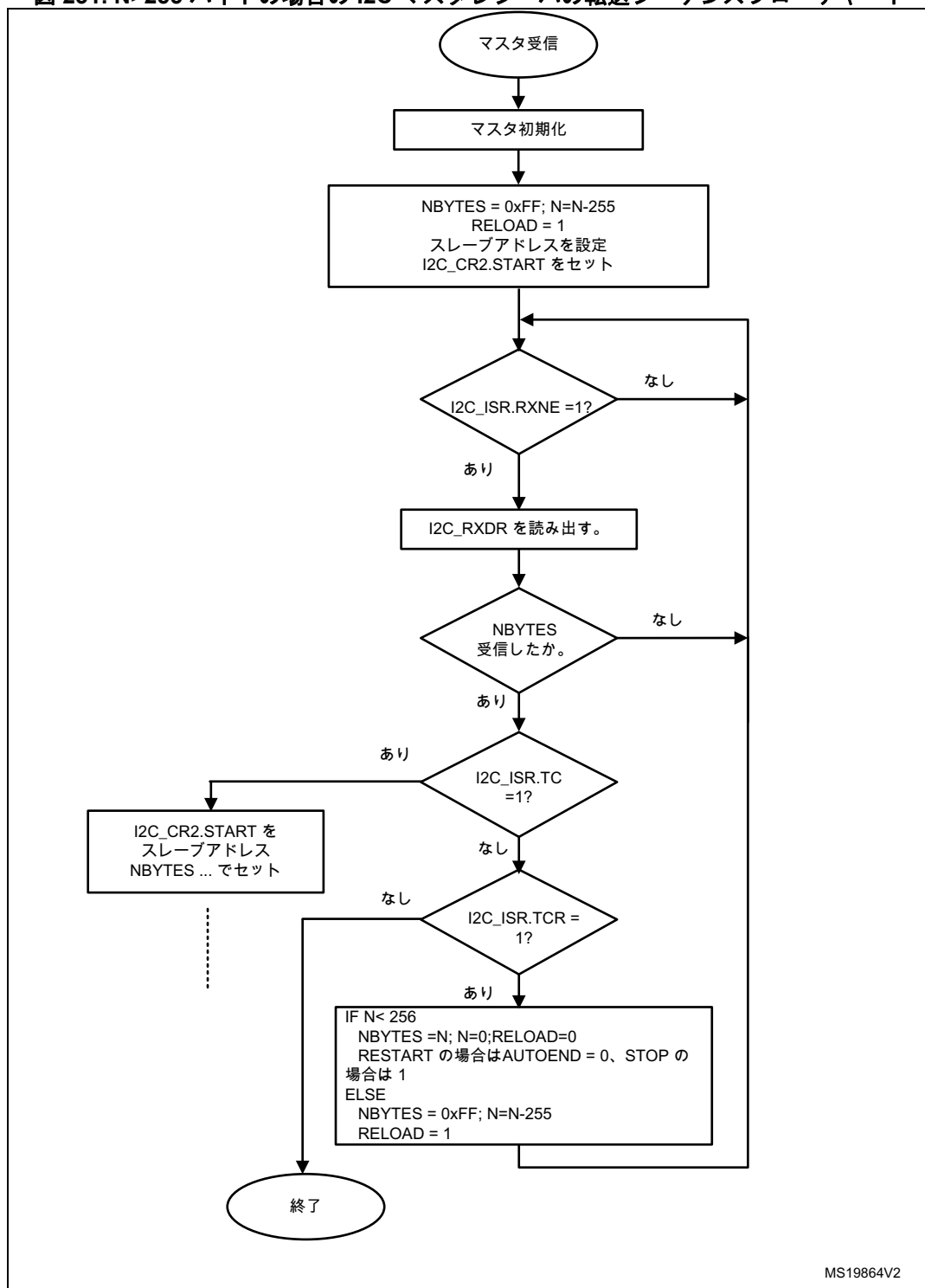
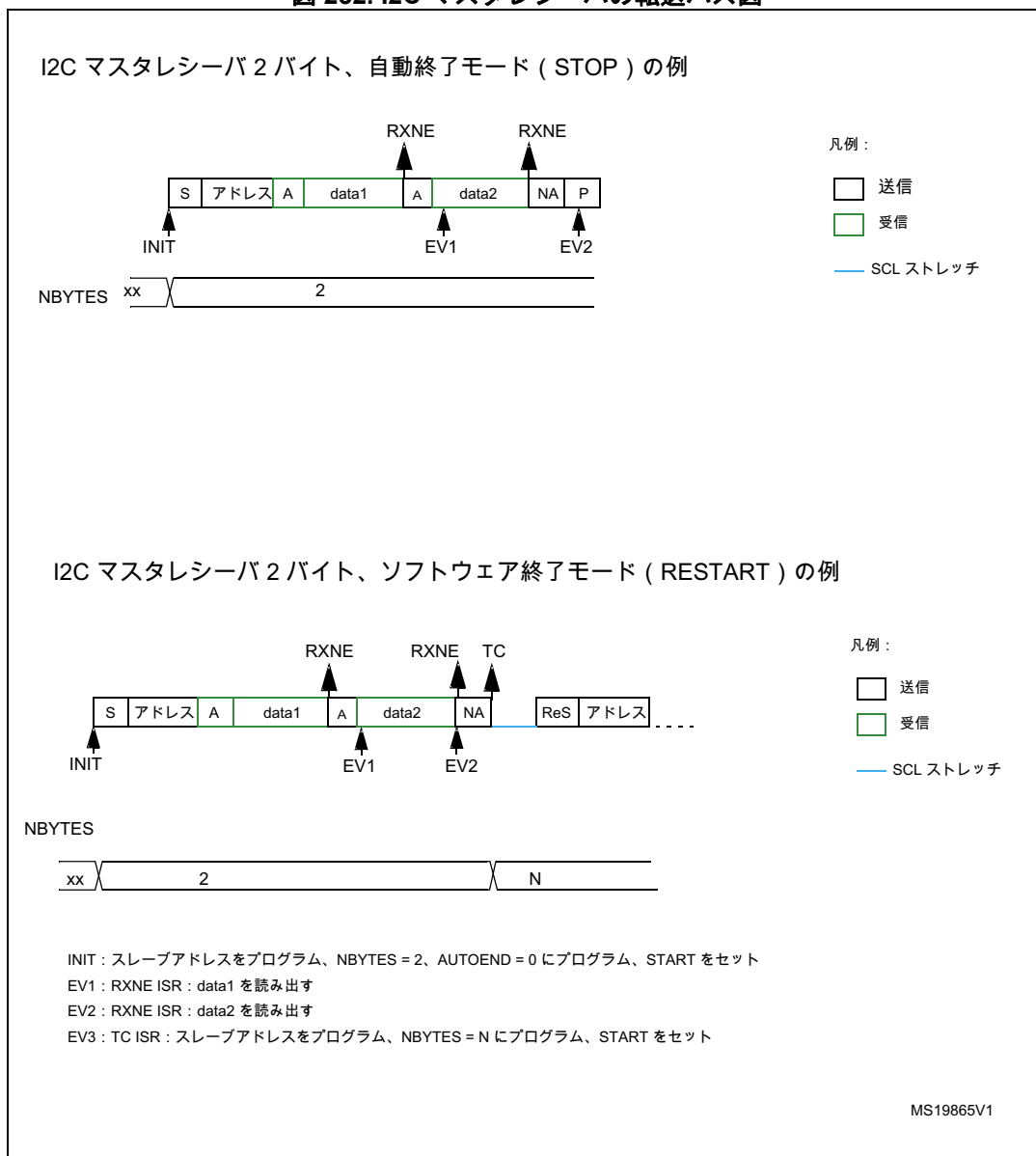


図 232. I2C マスタレシーバの転送バス図



コード例については、「付録」セクション [A.14.5: I2C マスタレシーバのコード例](#) を参照してください。

25.4.10 I2C_TIMINGR レジスタの設定例

下の表に、I²C 仕様に準拠したタイミングを得るための I2C_TIMINGR をプログラムする方法の例を示します。より正確な設定値を得るには、アプリケーションノート I²C タイミング設定ツール (AN4235) および該当するソフトウェア STSW-STM32126 を参照してください。

表 83. f_{I2CCLK} = 8 MHz でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t _{SCLL}	200x250 ns = 50 μs	20x250 ns = 5.0 μs	10x125 ns = 1250 ns	7x125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t _{SCLH}	196x250 ns = 49 μs	16x250 ns = 4.0 μs	4x125ns = 500ns	4x125 ns = 500 ns
t _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~2000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x1	0x0
t _{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t _{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

1. SCL 周期 t_{SCL} は、SCL 内部検出遅延のため、t_{SCLL} + t_{SCLH} より大きくなります。t_{SCL} として示されている値は例にすぎません。
2. t_{SYNC1} + t_{SYNC2} 最小値は、4 x t_{I2CCLK} = 500 ns です。例：t_{SYNC1} + t_{SYNC2} = 1000 ns
3. t_{SYNC1} + t_{SYNC2} 最小値は、4 x t_{I2CCLK} = 500 ns です。例：t_{SYNC1} + t_{SYNC2} = 750 ns
4. t_{SYNC1} + t_{SYNC2} 最小値は、4 x t_{I2CCLK} = 500 ns です。例：t_{SYNC1} + t_{SYNC2} = 655 ns

表 84. f_{I2CCLK} = 16 MHz でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t _{SCLL}	200 x 250 ns = 50 μs	20 x 250 ns = 5.0 μs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t _{SCLH}	196 x 250 ns = 49 μs	16 x 250 ns = 4.0 μs	4 x 125ns = 500 ns	3 x 62.5 ns = 187.5 ns
t _{SCL} ⁽¹⁾	~100 μs ⁽²⁾	~10 μs ⁽²⁾	~2500 ns ⁽³⁾	~1000 ns ⁽⁴⁾
SDADEL	0x2	0x2	0x2	0x0
t _{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t _{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. SCL 周期 t_{SCL} は、SCL 内部検出遅延のため、t_{SCLL} + t_{SCLH} より大きくなります。t_{SCL} として示されている値は例にすぎません。
2. t_{SYNC1} + t_{SYNC2} 最小値は、4 x t_{I2CCLK} = 250 ns です。例：t_{SYNC1} + t_{SYNC2} = 1000 ns
3. t_{SYNC1} + t_{SYNC2} 最小値は、4 x t_{I2CCLK} = 250 ns です。例：t_{SYNC1} + t_{SYNC2} = 750 ns
4. t_{SYNC1} + t_{SYNC2} 最小値は、4 x t_{I2CCLK} = 250 ns です。例：t_{SYNC1} + t_{SYNC2} = 500 ns

表 85. $f_{I2CCLK} = 48 \text{ MHz}$ でのタイミング設定の例

パラメータ	標準モード (Sm)		高速モード (Fm)	高速モードプラス (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	$200 \times 250 \text{ ns} = 50 \mu\text{s}$	$20 \times 250 \text{ ns} = 5.0 \mu\text{s}$	$10 \times 125 \text{ ns} = 1250 \text{ ns}$	$4 \times 125 \text{ ns} = 500 \text{ ns}$
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	$196 \times 250 \text{ ns} = 49 \mu\text{s}$	$16 \times 250 \text{ ns} = 4.0 \mu\text{s}$	$4 \times 125 \text{ ns} = 500 \text{ ns}$	$2 \times 125 \text{ ns} = 250 \text{ ns}$
$t_{SCL}^{(1)}$	$\sim 100 \mu\text{s}^{(2)}$	$\sim 10 \mu\text{s}^{(2)}$	$\sim 2500 \text{ ns}^{(3)}$	$\sim 875 \text{ ns}^{(4)}$
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	$2 \times 250 \text{ ns} = 500 \text{ ns}$	$2 \times 250 \text{ ns} = 500 \text{ ns}$	$3 \times 125 \text{ ns} = 375 \text{ ns}$	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	$5 \times 250 \text{ ns} = 1250 \text{ ns}$	$5 \times 250 \text{ ns} = 1250 \text{ ns}$	$4 \times 125 \text{ ns} = 500 \text{ ns}$	$2 \times 125 \text{ ns} = 250 \text{ ns}$

1. SCL 周期 t_{SCL} は、SCL 内部検出遅延のため、 $t_{SCLL} + t_{SCLH}$ より大きくなります。 t_{SCL} として示されている値は例にすぎません。
2. $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 83.3 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 1000 \text{ ns}$
3. $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 83.3 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 750 \text{ ns}$
4. $t_{SYNC1} + t_{SYNC2}$ 最小値は、 $4 \times t_{I2CCLK} = 83.3 \text{ ns}$ です。例: $t_{SYNC1} + t_{SYNC2} = 250 \text{ ns}$

25.4.11 SMBus 固有の機能

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#)を参照してください。

概要

システム管理バス (SMBus) は、さまざまなデバイスが互いに通信したり、残りのシステム部分と通信したりできる 2 線インタフェースです。I²C の動作原理に基づきます。SMBus により、システムおよびパワーマネジメント関連のタスク向けの制御バスが実現できます。

このペリフェラルは、SMBUS 仕様 rev 2.0 (<http://smbus.org>) と互換性があります。

システム管理バス仕様では、3 種類のデバイスを規定しています。

- スレーブとは、コマンドを受信したり、コマンドに応答したりするデバイスです。
- マスタとは、コマンドを発行し、クロックを生成し、転送を終了させるデバイスです。
- ホストとは、システムの CPU にメインインタフェースを提供する特殊なマスタです。ホストは、マスタ/スレーブとすることができ、SMBus ホスト通知プロトコルをサポートする必要があります。システム内では、ただ 1 つのホストが許容されます。

このペリフェラルは、マスタまたはスレーブデバイスとして、また、ホストとしても設定できます。



SMBUS は、I²C 仕様 rev 2.1 に基づきます。

バスプロトコル

特定のデバイスについて、11 の可能なコマンドプロトコルがあります。デバイスは、11 のプロトコルの一部または全部を使用して通信できます。プロトコルは、Quick Command、Send Byte、Receive Byte、Write Byte、Write Word、Read Byte、Read Word、Process Call、Block Read、Block Write、および Block Write-Block Read Process Call です。これらのプロトコルは、ユーザのソフトウェアによって実装してください。

これらのプロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

アドレス解決プロトコル (ARP)

SMBus スレーブアドレスの競合は、各スレーブデバイスに新しいユニークなアドレスを動的に割り当てることによって解決できます。アドレス割り当てを目的とする各デバイスを分離する仕組みを提供するために、各デバイスは一意デバイス識別子 (UDID) を実装する必要があります。128 ビットの番号がソフトウェアによって実装されます。

このペリフェラルは、アドレス解決プロトコル (ARP) をサポートします。SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。ARP コマンドは、ユーザのソフトウェアによって実装してください。

ARP サポートのために、スレーブモードでアービトレーションも行われます。

SMBus アドレス解決プロトコルの詳細については、SMBus 仕様バージョン 2.0 (<http://smbus.org>) を参照してください。

受信コマンドおよびデータ確認応答制御

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、セクション [スレーブバイト制御モード \(622 ページ\)](#) を参照してください。

Host Notify プロトコル

このペリフェラルは、I2C_CR1 レジスタの SMBHEN ビットをセットすることによって、Host Notify (ホスト通知) プロトコルをサポートします。この場合、ホストは SMBus ホストアドレス (0b0001 000) を確認応答します。

このプロトコルが使用されると、デバイスはマスタとして動作し、ホストはスレーブとして動作します。

SMBus アラート

SMBus ALERT オプション信号がサポートされます。スレーブ専用デバイスは、通信したいホストの SMBALERT# ピンを通じてホストに信号を送信します。ホストは、割り込みを処理し、アラート応答アドレス (0b0001 100) を通じて全 SMBALERT# デバイスに同時にアクセスします。SMBALERT# をローに引き下げたデバイスのみが、アラート応答アドレスを確認応答します。

スレーブデバイスとして設定されたとき (SMBHEN=0)、I2C_CR1 レジスタの ALERTEN ビットをセットすることによって、SMBA ピンはローに引き下げられます。同時に、アラート応答アドレスが有効になります。

ホストとして設定されたとき (SMBHEN=1)、SMBA ピンで立ち下がりエッジが検出され、ALERTEN=1 のとき、I2C_ISR レジスタの ALERT フラグがセットされます。I2C_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。ALERTEN=0 のときには、外部 SMBA ピンがローの場合でも、ALERT ラインはハイとみなされます。

SMBus ALERT ピンが不要な場合には、ALERTEN=0 の場合、SMBA ピンを標準 GPIO として使用できます。

パケットエラーチェック

信頼性と通信の堅牢性を向上させるために、SMBus 仕様にパケットエラーチェックメカニズムが導入されました。パケットエラーチェックは、各メッセージ転送の終わりにパケットエラーコード (PEC) を付加することによって実装されます。PEC は、すべてのメッセージバイト (アドレスと読み出し/書き込みビットを含む) に対して $C(x) = x_8 + x^2 + x + 1$ CRC-8 多項式を使用して計算されます。

ペリフェラルはハードウェア PEC 計算機が組み込まれ、受信バイトがハードウェアによって計算された PEC に一致しないときには自動的に非確認応答を送信できます。

タイムアウト

このペリフェラルは、SMBus 仕様バージョン 2.0 で定義された 3 つのタイムアウトに準拠するために、ハードウェアタイマが組み込まれています。

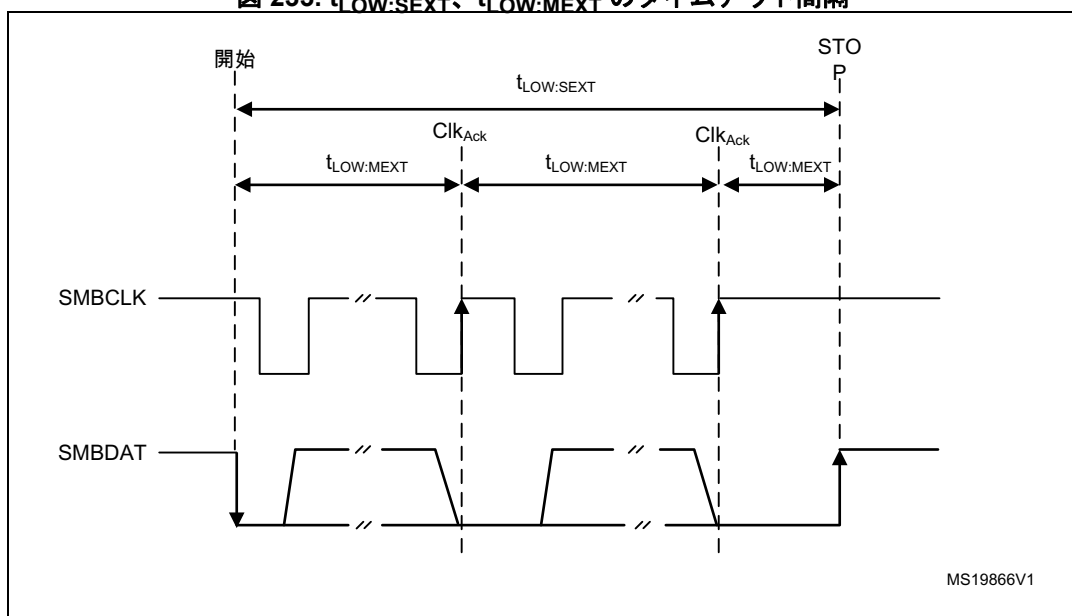
表 86. SMBus タイムアウト仕様

記号	パラメータ	リミット		単位
		最小値	最大値	
t_{TIMEOUT}	クロックロータイムアウト検出	25	35	ms
$t_{\text{LOW:SEXT}}^{(1)}$	累積クロックロー延長時間 (スレーブデバイス)	-	25	ms
$t_{\text{LOW:MEXT}}^{(2)}$	累積クロックロー延長時間 (マスタデバイス)	-	10	ms

1. $t_{\text{LOW:SEXT}}$ は、特定のスレーブデバイスが初めの START から STOP までの 1 つのメッセージのクロックサイクルを延長できる累積時間です。別のスレーブデバイスまたはマスタもクロックを延長して、合計のクロックロー延長時間が $t_{\text{LOW:SEXT}}$ より大きくなる場合があります。したがって、このパラメータは、スレーブデバイスをフルスピードのマスタの単独のターゲットとして測定されます。
2. $t_{\text{LOW:MEXT}}$ は、マスタがメッセージの各バイト内のクロックサイクルを START-to-ACK、ACK-to-ACK、または ACK-to-STOP から定義に従って延長できる累積時間です。スレーブデバイスまたは別のマスタもクロックを延長して、合計のクロックロー時間が $t_{\text{LOW:MEXT}}$ より大きくなる場合があります。したがって、このパラメータは、フルスピードスレーブデバイスをマスタの単独のターゲットとして測定されます。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

図 233. $t_{\text{LOW:SEXT}}$ 、 $t_{\text{LOW:MEXT}}$ のタイムアウト間隔



バスアイドル検出

マスタは、クロックおよびデータ信号が $t_{\text{HIGH,MAX}}$ より大きい t_{IDLE} の間ハイであった場合、バスはフリーであるとみなすことができます (表 82 : I2C-SMBUS 仕様のクロックタイミングを参照してください)

このタイミングパラメータは、マスタがバスに動的に追加し、SMBCLK または SMBDAT ラインで状態遷移を検出できなかった可能性のあるコンディションをカバーします。この場合、マスタは十分に長い時間待って、転送が進行中でないことを確認する必要があります。このペリフェラルは、ハードウェアバスアイドル検出をサポートします。

25.4.12 SMBus 初期化

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。セクション 25.3 : I2C の実装を参照してください。

SMBus 通信を行うためには、I2C 初期化に加えて、他にも特定の初期化を行う必要があります。

受信コマンドおよびデータ確認応答制御 (スレーブモード)

SMBus レシーバは、受信した各コマンドまたはデータを NACK できなければなりません。スレーブモードで ACK 制御を可能にするためには、I2C_CR1 レジスタの SBC ビットをセットすることによって、スレーブバイト制御モードを有効にする必要があります。詳細については、スレーブバイト制御モード (622 ページ) を参照してください。

特定アドレス (スレーブモード)

必要な場合は、特定の SMBus アドレスを有効にしてください。詳細については、[バスアイドル検出 \(646 ページ\)](#) を参照してください。

- SMBus デバイスのデフォルトアドレス (0b1100 001) は、I2C_CR1 レジスタの SMBDEN ビットをセットすることによって有効になります。
- SMBus ホストアドレス (0b0001 000) は、I2C_CR1 レジスタの SMBHEN ビットをセットすることによって有効になります。
- アラート応答アドレス (0b0001100) は、I2C_CR1 レジスタの ALERTEN ビットをセットすることによって有効になります。

パケットエラーチェック

PEC 計算を有効にするには、I2C_CR1 レジスタの PECEN ビットをセットします。その場合、PEC 転送はハードウェアバイトカウンタ (I2C_CR2 レジスタの NBYTES[7:0]) を使用して管理されます。PECEN ビットは、I2C を有効にする前に設定する必要があります。

PEC 転送はハードウェアバイトカウンタによって管理されるので、スレーブモードで SMBus とインタフェースするときには SBC ビットをセットする必要があります。PEC は、PECBYTE ビットがセットされ、RELOAD ビットがクリアされたとき、NBYTES-1 データの転送後に転送されます。RELOAD がセットされた場合、PECBYTE は効果がありません。

注意： I2C が有効なときには、PECEN 設定の変更はできません。

表 87. SMBUS の PEC 設定

モード	SBC ビット	RELOAD ビット	AUTOEND ビット	PECBYTE ビット
マスタ Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
マスタ Tx/Rx NBYTES + PEC + RESTART	x	0	0	1
スレーブ Tx/Rx と PEC	1	0	x	1

タイムアウト検出

タイムアウト検出は、I2C_TIMEOUTR レジスタの TIMOUTEN および TEXTEN ビットをセットすることによって有効になります。SMBus 仕様バージョン 2.0 で指定された最大時間の前にタイムアウトを検出するようにタイマをプログラムする必要があります。

- t_{TIMEOUT} チェック
 t_{TIMEOUT} チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] ビットを t_{TIMEOUT} パラメータをチェックするためにタイマ再ロード値でプログラムする必要があります。SCL ローレベルタイムアウトを検出するためには、TIDLE ビットを 0 に設定する必要があります。
その場合、タイマは、I2C_TIMEOUTR レジスタの TIMOUTEN をセットすることによって有効になります。
 $\text{SCL が } (\text{TIMEOUTA}+1) \times 2048 \times t_{\text{I2CCLK}}$ より長い時間、ローに設定された場合、I2C_ISR レジスタの TIMEOUT フラグがセットされます。
[表 88：さまざまな I2CCLK 周波数での TIMEOUTA の設定例 \(最大値 \$t_{\text{TIMEOUT}} = 25 \text{ ms}\$ \)](#) を参照してください。

注意： TIMOUTEN ビットがセットされているときには、TIMEOUTA[11:0] ビットおよび TIDLE ビットの設定変更はできません。

- $t_{\text{LOW:SEXT}}$ および $t_{\text{LOW:MEXT}}$ チェック
ペリフェラルがマスタとして設定されているか、スレーブとして設定されているかに応じて、12 ビットの TIMEOUTB タイマは、スレーブの場合は $t_{\text{LOW:SEXT}}$ をチェックするために、マスタの場合は $t_{\text{LOW:MEXT}}$ をチェックするために、設定する必要があります。標準では最大値のみが規定されているので、両方について同じ値を選ぶことができます。
その場合、タイマは、I2C_TIMEOUTR レジスタの TEXTEN ビットをセットすることによって有効になります。
SMBus ペリフェラルが、 $(\text{TIMEOUTB}+1) \times 2048 \times t_{\text{I2CCLK}}$ より長い時間および [バスアイドル検出 \(646 ページ\)](#) セクションで述べられているタイムアウト間隔で、累積 SCL ストレッチを実行した場合、I2C_ISR レジスタの TIMEOUT フラグがセットされます。
[表 89 : さまざまな I2CCLK 周波数での TIMEOUTB の設定例](#) を参照してください。

注意 : TEXTEN ビットがセットされているときには、TIMEOUTB 設定変更はできません。

バスアイドル検出

t_{IDLE} チェックを有効にするためには、12 ビットの TIMEOUTA[11:0] フィールドを t_{IDLE} パラメータを得るためにタイマ再ロード値でプログラムする必要があります。SCL および SDA ハイレベルタイムアウトを検出するためには、TIDLE ビットを 1 に設定する必要があります。
その場合、タイマは、I2C_TIMEOUTR レジスタの TIMEOUTEN ビットをセットすることによって有効になります。
SCL および SDA の両方のラインが $(\text{TIMEOUTA}+1) \times 4 \times t_{\text{I2CCLK}}$ より長い間ハイのままであった場合、I2C_ISR レジスタの TIMEOUT フラグがセットされます。
[表 90 : さまざまな I2CCLK 周波数での TIMEOUTA の設定例 \(最大値 \$t_{\text{IDLE}} = 50 \mu\text{s}\$ \)](#) を参照してください。

注意 : TIMEOUTEN がセットされているときに、TIMEOUTA および TIDLE 設定を変更することはできません。

25.4.13 SMBus : I2C_TIMEOUTR レジスタの設定例

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#) を参照してください。
● t_{TIMEOUT} の最大時間を 25 ms に設定 :

表 88. さまざまな I2CCLK 周波数での TIMEOUTA の設定例
(最大値 $t_{\text{TIMEOUT}} = 25 \text{ ms}$)

f_{I2CCLK}	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	t_{TIMEOUT}
8 MHz	0x61	0	1	$98 \times 2048 \times 125 \text{ ns} = 25 \text{ ms}$
16 MHz	0xC3	0	1	$196 \times 2048 \times 62.5 \text{ ns} = 25 \text{ ms}$
48 MHz	0x249	0	1	$586 \times 2048 \times 20.08 \text{ ns} = 25 \text{ ms}$

- $t_{\text{LOW:SEXT}}$ および $t_{\text{LOW:MEXT}}$ の最大時間を 8 ms に設定 :



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 89. さまざまな I2CCLK 周波数での TIMEOUTB の設定例

f_{I2CCLK}	TIMEOUTB[11:0] ビット	TEXTEN ビット	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

- t_{IDLE} の最大時間を 50 μ s に設定

表 90. さまざまな I2CCLK 周波数での TIMEOUTA の設定例 (最大値 $t_{IDLE} = 50 \mu$ s)

f_{I2CCLK}	TIMEOUTA[11:0] ビット	TIDLE ビット	TIMEOUTEN ビット	t_{IDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 μ s
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 μ s
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 μ s

25.4.14 SMBus スレーブモード

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#) を参照してください。

2C スレーブ転送管理 ([セクション 25.4.8 : I2C スレーブモード](#) を参照) に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

SMBus スレーブトランスミッタ

IP が SMBus で使用されるときには、SBCは、プログラムされたデータバイト数の終わりの PEC 送信を可能にするため、1 にプログラムする必要があります。PECBYTE ビットがセットされているときには、NBYTES[7:0] でプログラムされたバイト数には PEC 送信が含まれます。その場合、TXIS 割り込みの合計数は NBYTES-1 であり、NBYTES-1 データ転送後にマスタが追加のバイトをリクエストした場合、I2C_PECR レジスタの内容が自動的に送信されます。

注意 : *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 234. N バイト + PEC の場合の SMBus スレーブトランスミッタの転送シーケンスフローチャート

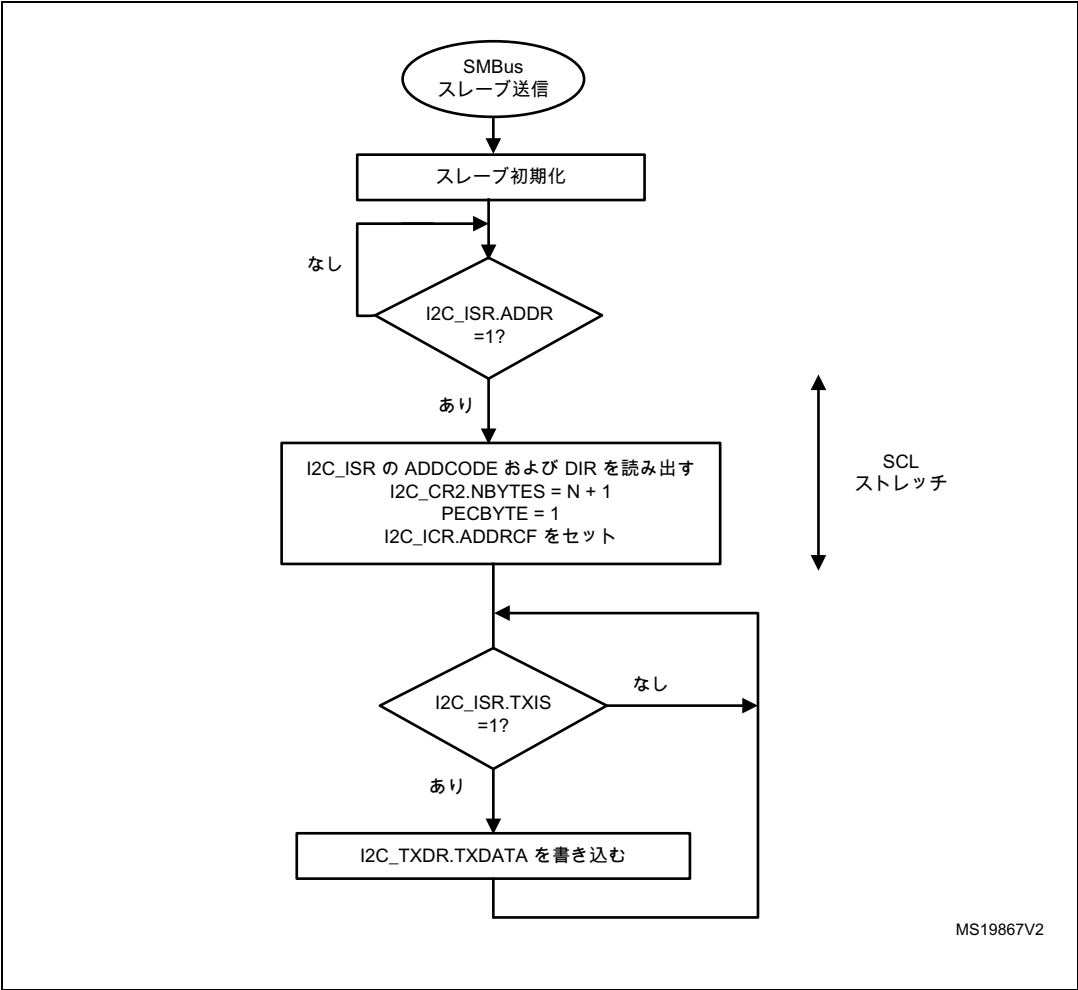
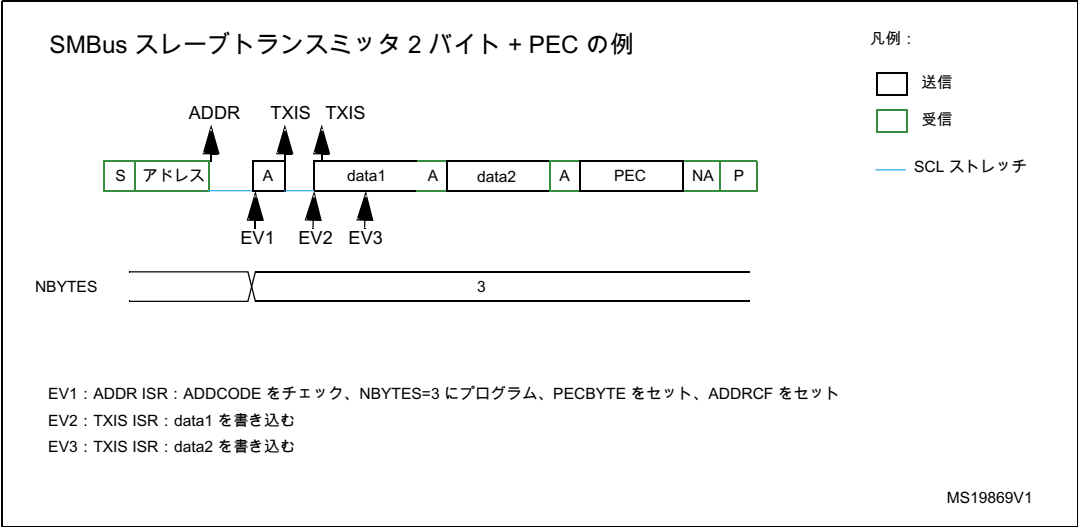


図 235. SMBus スレーブトランスミッタの転送バス図 (SBC=1)



SMBus スレーブレシーバ

I2C が SMBus モードで使用されるときには、SBC は、プログラムされたデータバイト数の終わりの PEC チェックを可能にするため、1 にプログラムする必要があります。各バイトの ACK 制御を可能にするためには、再ロードモードを選択する必要があります (RELOAD=1)。詳細については、[スレーブバイト制御モード \(622 ページ\)](#) を参照してください。

PEC バイトをチェックするためには、RELOAD ビットをクリアして、PECBYTE ビットをセットする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが内部 I2C_PECR レジスタの内容と比較されます。ACK ビットの値にかかわらず、比較が一致しなかった場合は NACK が自動的に生成され、比較が一致した場合は ACK が自動的に生成されます。PEC バイトが受信されると、他のデータと同様に I2C_RXDR レジスタにコピーされ、RXNE フラグがセットされます。

PEC 不一致の場合、PECERR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

ACK ソフトウェア制御が不要な場合は、PECBYTE=1 をプログラムし、同じ書き込み操作で NBYTES を連続フローで受信するバイト数にプログラムします。NBYTES-1 が受信された後、次の受信バイトが PEC であるかどうかチェックされます。

注意： *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 236. N バイト + PEC の場合の SMBus スレーブレシーバの転送シーケンスフローチャート

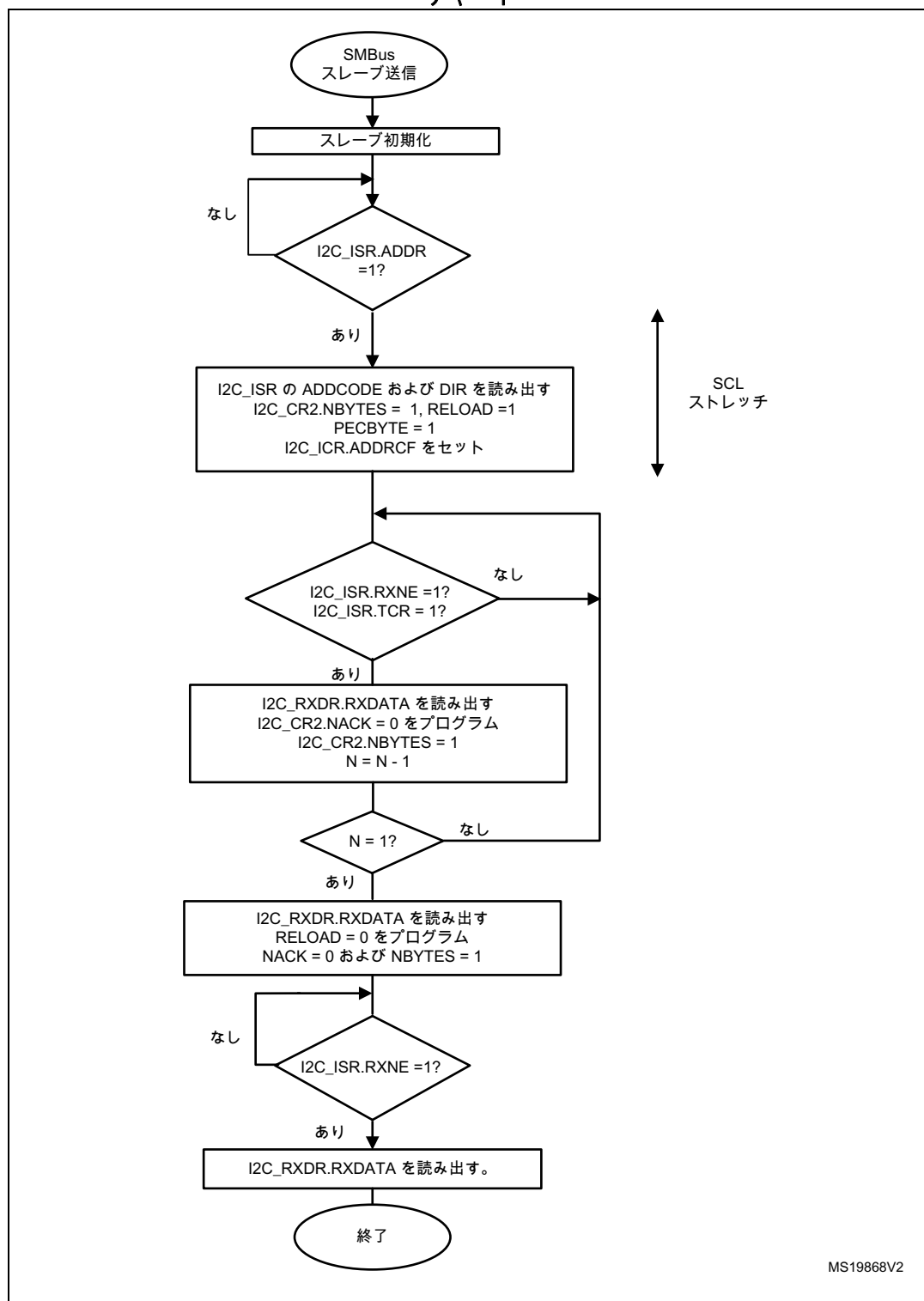
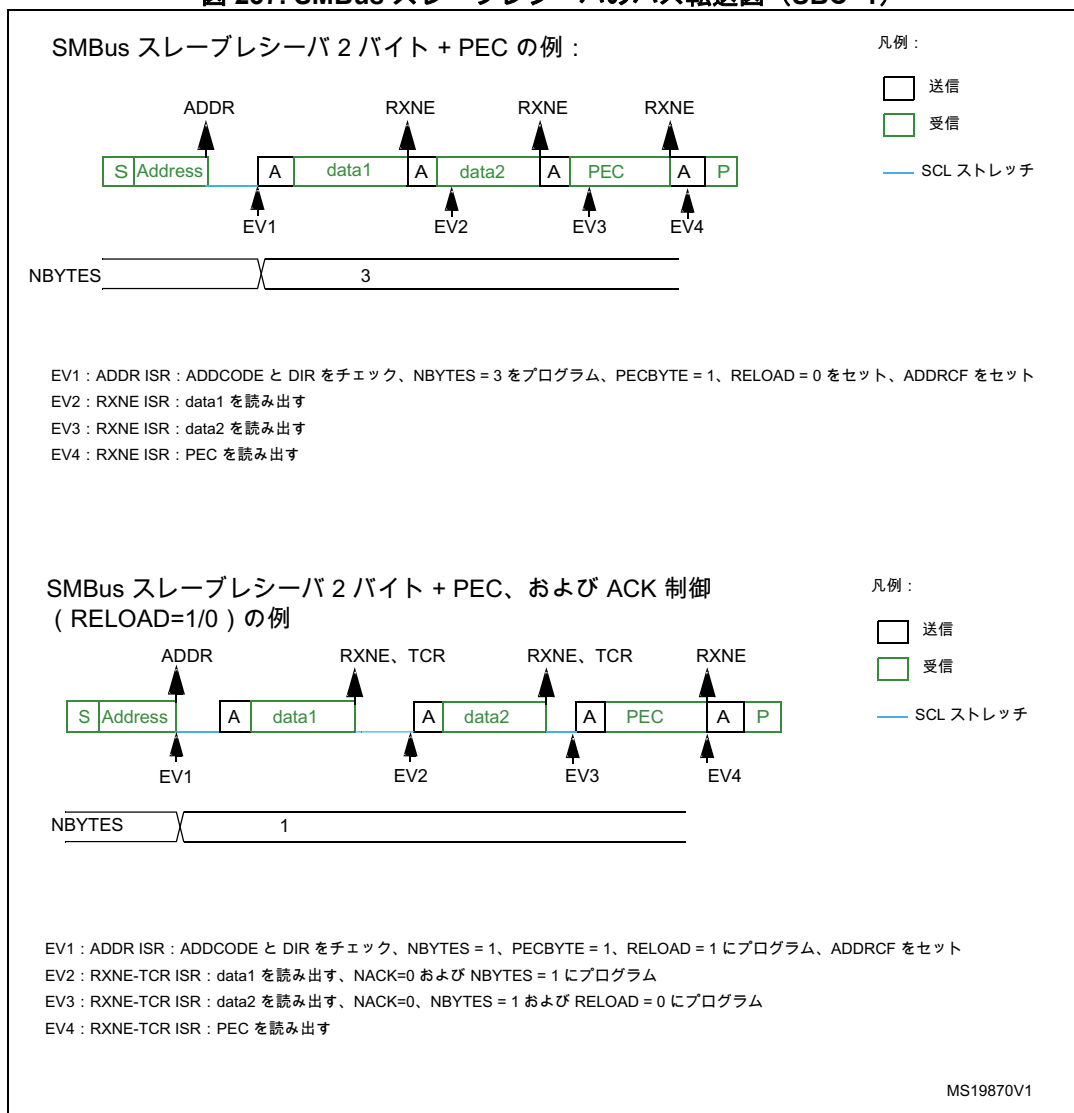


図 237. SMBus スレーブレシーバのバス転送図 (SBC=1)



このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#) を参照してください。

I2C マスタ転送管理 ([セクション 25.4.9 : I2C マスタモード](#) を参照) に加えて、SMBus をサポートするために、いくつか追加のソフトウェアフローチャートが用意されています。

SMBus マスタトランスミッタ

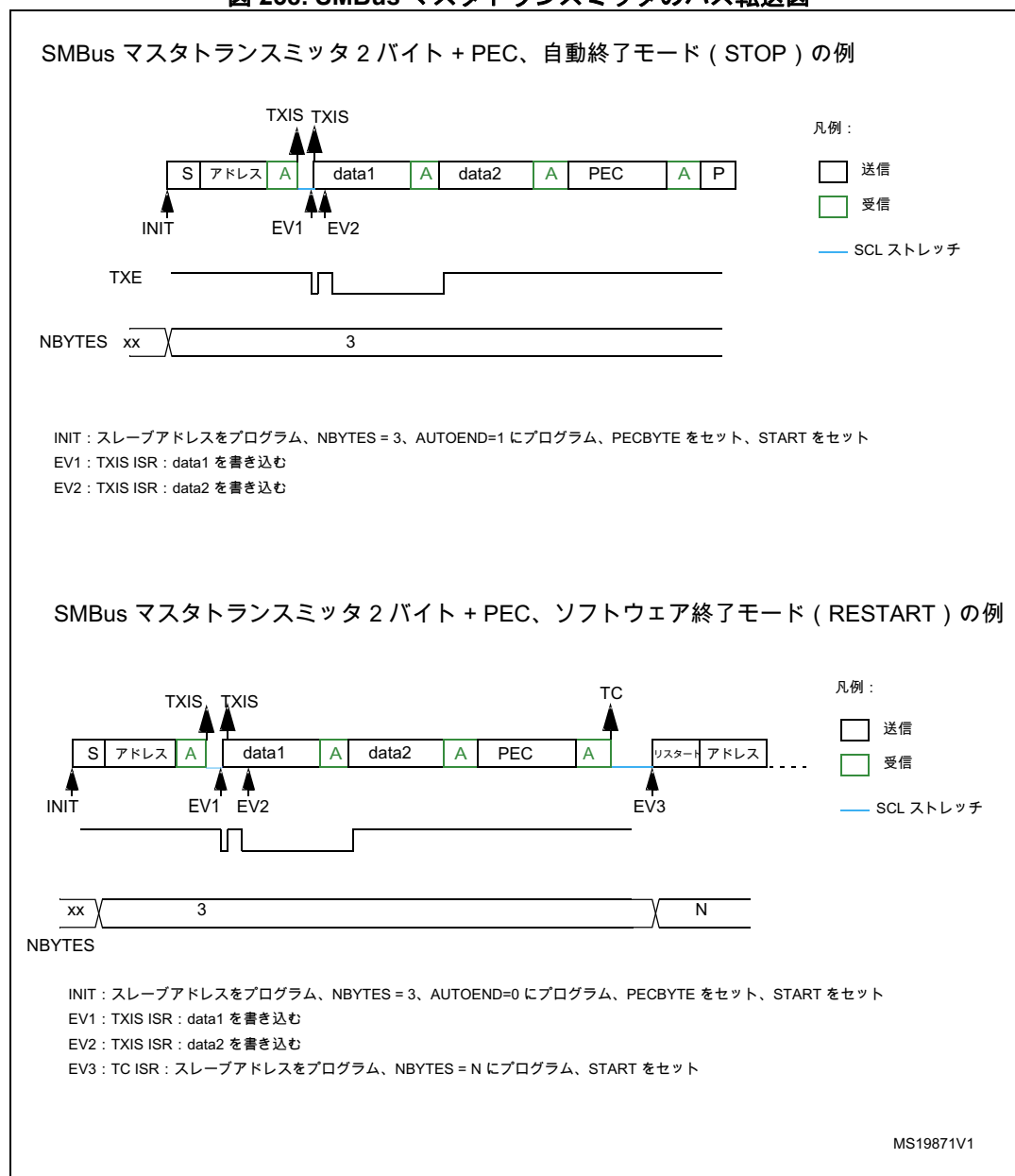
SMBus マスタが PEC を送信したいときには、START ビットをセットする前に、PECBYTE ビットをセットする必要があります。バイト数を NBYTES[7:0] フィールドでプログラムする必要があります。この場合、TXIS 割り込みの合計数は NBYTES-1 になります。したがって、NBYTES=0x1 のときに PECBYTE ビットがセットされた場合、I2C_PECR レジスタの内容が自動的に送信されます。

SMBus マスタが PEC 後に STOP コンディションを送信したい場合は、自動終了モードを選択してください (AUTOEND=1)。この場合、PEC 送信に続いて、STOP コンディションが自動的に送信されます。

SMBus マスタが PEC 後に RESTART コンディションを送信したい場合は、ソフトウェアモードを選択してください (AUTOEND=0)。この場合、NBYTES-1 が送信されると、PEC 送信後に I2C_PECR レジスタの内容が送信され、TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションを TC 割り込みサブルーチンでプログラムする必要があります。

注意： **PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。**

図 238. SMBus マスタトランスミッタのバス転送図



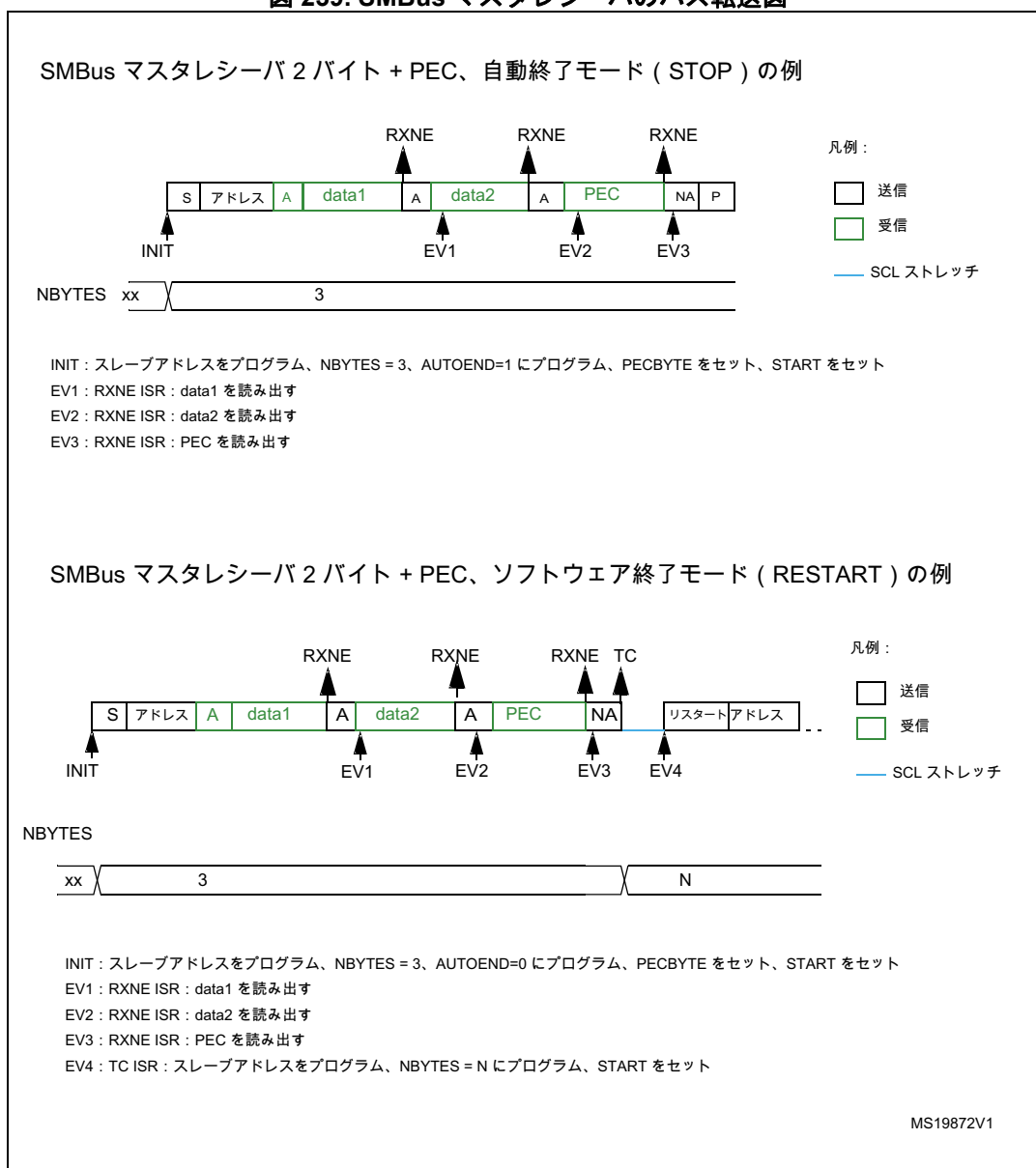
SMBus マスタレシーバ

SMBus マスタが転送終了時に PEC を受信してから STOP を受信したいときには、自動終了モードを選択できます (AUTOEND=1)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが I2C_PECR レジスタの内容と自動的に照合されます。PEC バイトに対して NACK 応答が与えられた後、STOP コンディションが送信されます。

SMBus マスタが転送終了時に PEC バイトを受信してから RESTART を受信したいときには、ソフトウェアモードを選択する必要があります (AUTOEND=0)。START ビットをセットする前に、PECBYTE ビットをセットする必要があります。スレーブアドレスをプログラムする必要があります。この場合、NBYTES-1 データが受信された後、次の受信バイトが I2C_PECR レジスタの内容と自動的に照合されます。PEC バイト受信後に TC フラグがセットされ、SCL ラインローをストレッチします。RESTART コンディションは、TC 割り込みサブルーチンでプログラムできます。

注意： *PECBYTE ビットは、RELOAD ビットがセットされているときには効果がありません。*

図 239. SMBus マスタレシーバのバス転送図



25.4.15 アドレス一致時に STOP モードからウェイクアップ

このセクションは、STOP モードからのウェイクアップ機能がサポートされるときにのみ適用されます。[セクション 25.3 : I2C の実装](#) を参照してください。

I2C は、アドレス指定されたとき、MCU を STOP モードからウェイクアップできます (APB クロックはオフ)。すべてのアドレッシングモードがサポートされます。

STOP モードからのウェイクアップを有効にするには、I2C_CR1 レジスタの WUPEN ビットをセットします。STOP モードからのウェイクアップを可能にするには、HSI オシレータを I2CCLK のクロックソースとして選択する必要があります。

STOP モード中、HSI はオフです。START が検出されると、I2C インタフェースは HSI をオンに切り替えて、HSI がウェイクアップするまで SCL ローをストレッチします。

HSI は、アドレス受信に使用されます。

アドレス一致の場合、I2C は、MCU のウェイクアップ時間の間、SCLローをストレッチします。ストレッチは、ADDR フラグがソフトウェアによってクリアされたときにリリースされ、転送は通常通りに続行されます。

アドレスが一致しなかった場合、HSI は再びオフになり、MCU はウェイクアップしません。

注： I2C クロックがシステムクロックの場合、または WUPEN = 0 の場合、START 受信後も HSI オシレータはオンになりません。

ADDR 割り込みによってのみ、MCU をウェイクアップできます。したがって、I2C がマスタとして、または ADDR フラグのセット後にアドレス指定されたスレーブとして転送を行っているときには、STOP モードに入らないでください。これを管理するには、ADDR 割り込みルーチンで SLEEPDEEP ビットをクリアして、STOPF フラグのセット後にのみ再びオンにセットします。

注意： デジタルフィルタは、STOP モードからのウェイクアップ機能と互換性がありません。DNF ビットが 0 でない場合、WUPEN ビットをセットしても効果はありません。

注意： この機能は、I2C クロックソースが HSI オシレータのときのみ使用できます。

注意： STOP モードからのウェイクアップ機能の正しい動作を保証するには、クロックストレッチを有効にする必要があります (NOSTRETCH=0)。

注意： STOP モードからのウェイクアップが無効な場合 (WUPEN=0)、STOP モードに入る前に、I2C ペリフェラルを無効にする必要があります (PE=0)。

25.4.16 エラー条件

以下は、通信エラーを引き起こす可能性のあるエラー条件です。

バスエラー (BERR)

バスエラーは、START または STOP コンディションが検出され、複数の 9 SCL クロックパルス後になかったときに検出されます。START または STOP コンディションは、SCL がハイと有的时候に SDA エッジが発生した場合に検出されます。

バスエラーフラグは、I2C がマスタまたはアドレス指定されたスレーブとして転送に関与する場合にのみ (すなわち、スレーブモードのアドレスフェーズでないとき)、セットされます。

スレーブモードで START または RESTART の誤配置が検出された場合、I2C は、正しい START コンディションの場合と同様に、アドレス認識状態に入ります。

バスエラーが検出されると、I2C_ISR レジスタの BERR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

アービトレーション喪失 (ARLO)

アービトレーション喪失は、SDA ラインでハイレベルが送信されたが、SCL 立ち上がりエッジでローレベルがサンプリングされたときに検出されます。

- マスタモードでは、アービトレーション喪失は、アドレスフェーズ、データフェーズ、およびデータ確認応答フェーズで検出されます。この場合、SDA および SCL ラインはリリースされ、START 制御ビットがハードウェアによってクリアされ、マスタは自動的にスレーブモードに切り替わります。
- スレーブモードでは、アービトレーション喪失は、データフェーズとデータ確認応答フェーズで検出されます。この場合、転送は中止され、SCL および SDA ラインがリリースされます。

アービトレーション喪失が検出されると、I2C_ISR レジスタの ARLO フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。



オーバーラン／アンダーランエラー (OVR)

オーバーランまたはアンダーラインエラーは、スレーブモードで NOSTRETCH=1 のとき、および次のときに検出されます：

- 受信時、新しいバイトが受信され、RXDR レジスタがまだ読み出されていないとき。新しい受信バイトは失われ、新しいバイトへの応答として NACK が自動的に送信されます。
- 送信時：
 - － STOPF=1 のときには、最初のデータバイトが送信されなければなりません。TXE=0、0xFF の場合、I2C_TXDR レジスタの内容が送信され、そうでない場合は送信されません。
 - － 新しいバイトが送信されるべきときに、I2C_TXDR レジスタがまだ書き込まれていなかった場合、0xFF が送信されます。

オーバーランまたはアンダーランエラーが検出されると、I2C_ISR レジスタの OVR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

パケットエラーチェックエラー (PECERR)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#) を参照してください。

PEC エラーは、受信した PEC バイトが I2C_PECR レジスタの内容と一致しなかったときに検出されます。正しくない PEC の受信後、NACK が自動的に送信されます。

PEC エラーが検出されると、I2C_ISR レジスタの PECERR フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

タイムアウトエラー (TIMEOUT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#) を参照してください。

タイムアウトエラーは、次のような条件で発生します：

- TIDLE=0 であり、SCL が TIMEOUTA[11:0] ビットで定義された時間だけローのままであった場合：これは SMBus タイムアウトの検出に使用されます。
- TIDLE=1 であり、SDA および SCL が TIMEOUTA[11:0] ビットで定義された時間だけハイのままであった場合：これはバスアイドル状態の検出に使用されます。
- マスタ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合 (SMBus $t_{\text{LOW:MEXT}}$ パラメータ)。
- スレーブ累積クロックロー延長時間が TIMEOUTB[11:0] ビットで定義された時間に達した場合 (SMBus $t_{\text{LOW:SEXT}}$ パラメータ)。

マスタモードでタイムアウト違反が検出されると、STOP コンディションが自動的に送信されます。

スレーブモードでタイムアウト違反が検出されると、SDA および SCL ラインが自動的にリリースされます。

タイムアウトエラーが検出されると、I2C_ISR レジスタの TIMEOUT フラグがセットされ、I2C_CR1 レジスタの ERRIE ビットがセットされていた場合は割り込みが生成されます。

アラート (ALERT)

このセクションは、SMBus 機能がサポートされるときにのみ適用されます。 [セクション 25.3 : I2C の実装](#) を参照してください。

ALERT フラグは、I2C インタフェースがホストとして設定され (SMBHEN=1)、アラートピン検出が有効であり (ALERTEN=1)、SMBA ピンで立ち下がりがエッジが検出されたときにセットされます。I2C_CR1 レジスタの ERRIE ビットがセットされている場合は、割り込みが生成されます。

25.4.17 DMA リクエスト

DMA を使用した送信

送信について DMA (Direct Memory Access) を有効にするには、I2C_CR1 レジスタの TXDMAEN ビットをセットします。TXIS ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 11 : ダイレクトメモリアクセスコントローラ \(DMA\) \(197 ページ\)](#)) を参照) を使用して設定された SRAM 領域から I2C_TXDR レジスタにロードされます。

データのみが DMA で転送されます。

- マスタモード : 初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます (送信されたスレーブアドレスを DMA で転送することはありません)。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。 [マスタトランスミッタ \(634 ページ\)](#) を参照してください。

コード例については、「付録」セクション [A.14.8 : DMA で送信するためにマスタモードに設定された I2C のコード例](#) を参照してください。

- スレーブモードでは :
 - NOSTRETCH=0 では、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。
 - NOSTRETCH=1 では、アドレス一致イベントの前に DMA を初期化する必要があります。
- SMBus をサポートする場合 : PEC 転送は NBYTES カウンタによって管理されます。 [SMBus スレーブトランスミッタ \(649 ページ\)](#) および [SMBus マスタトランスミッタ \(653 ページ\)](#) を参照してください。

注 : DMA が送信に使用される場合、TXIE ビットが有効である必要はありません。

DMA を使用した受信

受信について DMA (Direct Memory Access) を有効にするには、I2C_CR1 レジスタの RXDMAEN ビットをセットします。RXNE ビットがセットされているときには、データは、I2C_RXDR レジスタから DMA ペリフェラル ([セクション 11 : ダイレクトメモリアクセスコントローラ \(DMA\) \(197 ページ\)](#)) を参照) を使用して設定された SRAM 領域にロードされます。データのみ (PEC を含む) が DMA で転送されます。

- マスタモード、初期化、スレーブアドレス、方向、バイト数、および START ビットはソフトウェアによってプログラムされます。すべてのデータが DMA を使用して転送されるときには、START ビットをセットする前に、DMA を初期化する必要があります。転送の終了は、NBYTES カウンタによって管理されます。
- NOSTRETCH=0 のスレーブモードでは、すべてのデータが DMA を使用して転送されるときには、アドレス一致イベントの前、または ADDR 割り込みサブルーチンで、ADDR をクリアする前に DMA を初期化する必要があります。



- SMBus がサポートされる場合（[セクション 25.3 : I2C の実装](#) を参照）：PEC 転送は NBYTES カウンタによって管理されます。[SMBus スレーブレシーバ \(651 ページ\)](#) および [SMBus マスタレシーバ \(655 ページ\)](#) を参照してください。

注：DMA が受信に使用される場合、RXIE ビットが有効である必要はありません。

コード例については、「付録」セクション [A.14.9 : DMA で受信するためにスレーブモードに設定された I2C のコード例](#) を参照してください。

25.4.18 デバッグモード

マイクロコントローラがデバッグモードに入ると（コア停止）、DBG モジュールの DBG_I2Cx_SMBUS_TIMEOUT 設定ビットに応じて、SMBus タイムアウトは、通常の動作を続行するか、あるいは停止します。

25.5 I2C 低電力モード

表 91. 低電力モード

モード	説明
SLEEP	影響なし。 I2C 割り込みによって、デバイスは SLEEP モードから復帰します。
STOP	I2C レジスタの内容は保たれます。WUPEN=1 の場合：アドレス認識が機能します。I2C アドレス一致条件によって、デバイスは STOP モードから復帰します。WUPEN=0 の場合：STOP モードに入る前に I2C を無効にする必要があります。
STANDBY	I2C ペリフェラルはパワーダウンされ、STANDBY 終了後に再初期化する必要があります。

25.6 I2C 割り込み

次の表に、I2C 割り込みリクエストの一覧を示します。

表 92. I2C 割り込みリクエスト

割り込みイベント	イベントフラグ	イベントフラグ/ 割り込みのクリア方法	割り込み有効制御 ビット
受信バッファノットエンプティ	RXNE	I2C_RXDR レジスタを読み出す	RXIE
送信バッファ割り込みステータス	TXIS	I2C_TXDR レジスタに書き込む	TXIE
STOP 検出割り込みフラグ	STOPF	STOPCF=1 を書き込む	STOPIE
転送完了再ロード	TCR	I2C_CR2 の NBYTES[7:0] に 0 以外を書き込む	TCIE
転送完了	TC	START=1 または STOP=1 を 書き込む	
アドレス一致	ADDR	ADDRCF=1 を書き込む	ADDRIE
NACK 受信	NACKF	NACKCF=1 を書き込む	NACKIE



表 92. I2C 割り込みリクエスト (続き)

割り込みイベント	イベントフラグ	イベントフラグ/ 割り込みのクリア方法	割り込み有効制御 ビット
バスエラー	BERR	BERRCF=1 を書き込む	ERRIE
アービトレーション喪失	ARLO	ARLOCF=1 を書き込む	
オーバーラン/アンダーラン	OVR	OVRCF=1 を書き込む	
PEC エラー	PECERR	PECERRCF=1 を書き込む	
タイムアウト/ t_{LOW} エラー	TIMEOUT	TIMEOUTCF=1 を書き込む	
SMBus アラート	ALERT	ALERTCF=1 を書き込む	

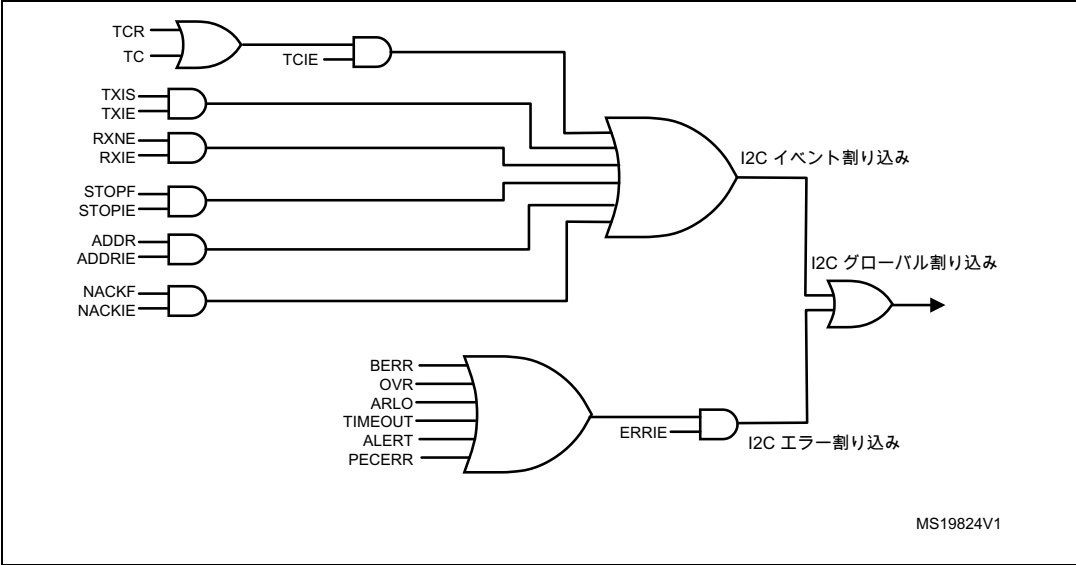
製品実装に応じて、これらの割り込みイベントのすべてで同じ割り込みベクトルを共有するか (I2C グローバル割り込み)、2 つの割り込みベクトル (I2C イベント割り込みと I2C エラー割り込み) にグループ化することができます。詳細については、表 37 : ベクタテーブルを参照してください。

I2C 割り込みを有効にするには、次のシーケンスが必要です。

1. NVIC で I2C IRQ チャンネルを設定し、有効にします。
2. 割り込みを生成するように I2C を設定します。

I2C ウェイクアップイベントは EXTI コントローラに接続されます (セクション 12.2 : 拡張割り込み / イベントコントローラ (EXTI) を参照)。

図 240. I2C 割り込みマッピング図



MS19824V1

25.7 I2C レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタは、ワード（32 ビット）単位でアクセスされます。

25.7.1 制御レジスタ 1 (I2C_CR1)

アドレスオフセット：0x00

リセット値：0x0000 0000

アクセス：このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERT EN	SMBD EN	SMBH EN	GCEN	WUP EN	NOST RETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	Res.	ANF オフ	DNF				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 **PECEN** : PEC イネーブル

0 : PEC 計算は無効です。

1 : PEC 計算は有効です。

注： **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。**

ビット 22 **ALERTEN**: SMBus アラート有効

デバイスモード (SMBHEN=0) :

0 : SMBA ピンをハイにリリリースし、アラート応答アドレスヘッダを無効にします。0001100x の後に NACK が続きます。

1 : SMBA ピンをローに駆動し、アラート応答アドレスヘッダを有効にします : 0001100x の後に ACK が続きます。

ホストモード (SMBHEN=1) :

0 : SMBus アラートピン (SMBA) はサポートされません。

1 : SMBus アラートピン (SMBA) はサポートされます。

注： **ALERTEN=0 のときには、SMBA ピンを標準 GPIO として使用できます。**

SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。

ビット 21 **SMBDEN** : SMBus デバイスデフォルトアドレス有効

0 : デバイスデフォルトアドレス無効。アドレス 0b1100001x は NACK されます。

1 : デバイスデフォルトアドレス有効。アドレス 0b1100001x は ACK されます。

注： **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。**

ビット 20 **SMBHEN** : SMBus ホストアドレス有効

0 : ホストアドレス無効。アドレス 0b0001000x は NACK されます。

1 : ホストアドレス有効。アドレス 0b0001000x は ACK されます。

注 : **SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。**

ビット 19 **GCEN** : 同報イネーブル

0 : 同報は無効です。アドレス 0b00000000 は NACK されます。

1 : 同報は有効です。アドレス 0b00000000 は ACK されます。

ビット 18 **WUPEN** : STOP モードからのウェイクアップ有効

0 : STOP モードからのウェイクアップ無効。

1 : STOP モードからのウェイクアップ有効。

注 : **STOP モードからのウェイクアップ機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。**

注 : **WUPEN は、DNF = 0000 のときのみセットできます。**

ビット 17 **NOSTRETCH** : クロックストレッチ無効

このビットは、スレーブモードでのクロックストレッチを無効にするために使用されます。マスタモードではクリアされたままでなければなりません。

0 : クロックストレッチ有効

1 : クロックストレッチ無効

注 : **このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。**

ビット 16 **SBC** : スレーブバイト制御

このビットは、スレーブモードでのハードウェアバイト制御を有効にするために使用されます。

0 : スレーブバイト制御無効

1 : スレーブバイト制御有効

ビット 15 **RXDMAEN** : DMA 受信リクエスト有効

0 : DMA モードは受信に無効

1 : DMA モードは受信に有効

ビット 14 **TXDMAEN** : DMA 送信リクエスト有効

0 : DMA モードは送信に無効

1 : DMA モードは送信に有効

ビット 13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **ANFOFF** : アナログノイズフィルタ OFF

0 : アナログノイズフィルタ有効

1 : アナログノイズフィルタ無効

注 : **このビットは、I2C が無効 (PE = 0) のときのみプログラムできます。**

ビット 11:8 **DNF[3:0]** : デジタルノイズフィルタ

これらのビットは、SDA および SCL 入力のデジタルノイズフィルタを設定するために使用されます。デジタルフィルタは、最大 $DNF[3:0] * t_{I2CCLK}$ の長さのスパイクを除去します。

0000 : デジタルフィルタ無効

0001 : デジタルフィルタは有効であり、最大 $1 t_{I2CCLK}$ の除去能力を持ちます。

...

1111 : デジタルフィルタは有効であり、最大 $15 t_{I2CCLK}$ の除去能力を持ちます。

注 : **アナログフィルタも有効化した場合、デジタルフィルタがアナログフィルタに追加されます。このフィルタは、I2C が無効 (PE = 0) のときのみプログラムできます。**

ビット 7 **ERRIE** : エラー割り込み有効

0 : エラー検出割り込み無効

1 : エラー検出割り込み有効

注 : 次のようなエラーが発生すると、割り込みが生成されます :

アービトレーション喪失 (ARLO)

バスエラー検出 (BERR)

オーバーラン/アンダーラン (OVR)

タイムアウト検出 (TIMEOUT)

PEC エラー検出 (PECERR)

アラートピンイベント検出 (ALERT)

ビット 6 **TCIE** : 転送完了割り込み有効

0 : 転送完了割り込み無効

1 : 転送完了割り込み有効

注 : 次のようなイベントが発生すると、割り込みが生成されます :

転送完了 (TC)

転送完了再ロード (TCR)

ビット 5 **STOPIE** : STOP 検出割り込み有効

0 : STOP 検出 (STOPF) 割り込み無効

1 : STOP 検出 (STOPF) 割り込み有効

ビット 4 **NACKIE** : 非確認応答受信割り込み有効

0 : 非確認応答 (NACKF) 受信割り込み無効

1 : 非確認応答 (NACKF) 受信割り込み有効

ビット 3 **ADDRIE** : アドレス一致割り込み有効 (スレーブのみ)

0 : アドレス一致 (ADDR) 割り込み無効

1 : アドレス一致 (ADDR) 割り込み有効

ビット 2 **RXIE** : RX 割り込み有効

0 : 受信 (RXNE) 割り込み無効

1 : 受信 (RXNE) 割り込み有効

ビット 1 **TXIE** : TX 割り込み有効

0 : 送信 (TXIS) 割り込み無効

1 : 送信 (TXIS) 割り込み有効

ビット 0 **PE** : ペリフェラルは有効です。

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

注 : **PE=0** のとき、**I2C SCL** および **SDA** ラインはリリースされます。内部ステートマシンおよびステータスビットはリセット値に戻されます。クリアされたときには、**PE** は少なくとも 3 APB クロックサイクルの間、ローに保たれる必要があります。

25.7.2 制御レジスタ 2 (I2C_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTOE ND	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_W RN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw									

ビット 31:27 予約済みであり、リセット値に保持する必要があります。

ビット 26 **PECBYTE** : パケットエラーチェックバイト

このビットはソフトウェアによってセットされ、PEC が転送されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、また、PE=0 のとき、ハードウェアによってクリアされます。

0 : PEC 転送なし。

1 : PEC 送信／受信がリクエストされます。

注 : このビットに 0 を書き込んでも、効果はありません。

このビットは、RELOAD がセットされているときには効果がありません。

このビットは、SBC=0 のとき、スレーブモードでは効果がありません。

SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。

ビット 25 **AUTOEND** : 自動終了モード (マスタモード)

このビットは、ソフトウェアによってセット／クリアされます。

0 : ソフトウェア終了モード : NBYTES データが転送されると TC フラグがセットされ、SCL ローをストレッチします。

1 : 自動終了モード : NBYTES データが転送されると、STOP コンディションが自動的に送信されます。

注 : このビットは、スレーブモードまたは RELOAD ビットがセットされているときには効果がありません。

ビット 24 **RELOAD** : NBYTES 再ロードモード

このビットは、ソフトウェアによってセット／クリアされます。

0 : 転送は、NBYTES データの転送後、完了します (STOP または RESTART が続きます)。

1 : 転送は、NBYTES データの転送後に完了しません (NBYTES が再ロードされます)。NBYTES データが転送されると TCR フラグがセットされ、SCL ローをストレッチします。

ビット 23:16 **NBYTES[7:0]** : バイト数

送受信されるバイト数は、ここでプログラムされます。このフィールドは、SBC=0 のスレーブモードでは効果がありません。

注 : START ビットがセットされているときに、これらのビットを変更することはできません。

ビット 15 NACK : NACK 生成 (スレーブモード)

このビットはソフトウェアによってセットされ、NACK が送信されたとき、または STOP コンディションあるいはアドレス一致を受信したとき、または PE=0 のとき、ハードウェアによってクリアされます。

0 : 現在の受信バイト後に ACK が送信されます。

1 : 現在の受信バイト後に NACK が送信されます。

注 : このビットに 0 を書き込んでも、効果はありません。

このビットは、スレーブモードでのみ使用されます : マスタレシーバモードでは、NACK ビットの値にかかわらず、STOP または RESTART コンディション前の最後のバイト後に NACK が自動的に生成されます。

スレーブレシーバ NOSTRETCH モードでオーバーランが発生すると、NACK ビットの値にかかわらず、NACK が自動的に生成されます。

ハードウェア PEC チェックが有効なとき (PECBYTE=1)、PEC 確認応答値は NACK 値に依存しません。

ビット 14 STOP : STOP 生成 (マスタモード)

このビットはソフトウェアによってセットされ、STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。

マスタモード :

0 : STOP 生成なし。

1 : 現在のバイト転送後の STOP 生成。

注 : このビットに 0 を書き込んでも、効果はありません。

ビット 13 START : START 生成

このビットはソフトウェアによってセットされ、START とアドレスシーケンスが送信された後、アービトラクション喪失によって、タイムアウトエラー検出によって、または PE=0 のときに、ハードウェアによってクリアされます。I2C_ICR レジスタの ADDRCF ビットに "1" を書き込むことによって、ソフトウェアによってクリアすることもできます。

0 : START 生成なし。

1 : RESTART/START 生成 :

– I2C がすでにマスタモードであり、AUTOEND=0 の場合、このビットをセットすると、RELOAD=0 のとき、NBYTES 転送の終了後に REPEATED START コンディションが生成されます。

– そうでない場合、このビットをセットすると、バスがフリーになると、START コンディションが生成されます。

注 : このビットに 0 を書き込んでも、効果はありません。

バスが BUSY の場合、または I2C がスレーブモードの場合でも、START ビットをセットできません。

このビットは、RELOAD がセットされているときには効果がありません。

ビット 12 HEAD10R : 読み出し方向のみの 10 ビットアドレスヘッダ (マスタレシーバモード)

0 : マスタは完全な 10 ビットスレーブアドレス読み出しシーケンスを送信します : START + 2 バイトの書き込み方向の 10 ビットアドレス + RESTART + 読み出し方向の 10 ビットアドレスの最初の 7 ビット。

1 : マスタは 10 ビットアドレスの最初の 7 ビットのみを送信し、その後読み出し方向を送信します。

注 : START ビットがセットされているときに、このビットを変更することはできません。

ビット 11 ADD10 : 10 ビットアドレッシングモード (マスタモード)

0 : マスタは 7 ビットアドレッシングモードで動作します。

1 : マスタは 10 ビットアドレッシングモードで動作します。

注 : START ビットがセットされているときに、このビットを変更することはできません。

ビット 10 RD_WRN : 転送方向 (マスタモード)

0 : マスタは書き込み転送をリクエストします。

1 : マスタは読み出し転送をリクエストします。

注 : START ビットがセットされているときに、このビットを変更することはできません。

ビット 9:8 **SADD[9:8]** : スレーブアドレスビット 9:8 (マスタモード)

7 ビットアドレッシングモード (ADD10=0) :

これらのビットは無視されます。

10 ビットアドレッシングモード (ADD10=1) :

これらのビットには、送信されるスレーブアドレスのビット 9:8 を書き込みます。

注: *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 7:1 **SADD[7:1]** : スレーブアドレスビット 7:1 (マスタモード)

7 ビットアドレッシングモード (ADD10=0) :

これらのビットには、送信される 7 ビットのスレーブアドレスを書き込みます。

10 ビットアドレッシングモード (ADD10=1) :

これらのビットには、送信されるスレーブアドレスのビット 7:1 を書き込みます。

注: *START ビットがセットされているときに、これらのビットを変更することはできません。*

ビット 0 **SADD0** : スレーブアドレスビット 0 (マスタモード)

7 ビットアドレッシングモード (ADD10=0) :

このビットは無視されます。

10 ビットアドレッシングモード (ADD10=1) :

このビットには、送信されるスレーブアドレスのビット 0 を書き込みます。

注: *START ビットがセットされているときに、これらのビットを変更することはできません。*

25.7.3 Own Address 1 レジスタ (I2C_OAR1)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]							OA1[0]
rw					rw	rw		rw							rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA1EN** : Own Address 1 有効

- 0 : Own Address 1 無効。受信されたスレーブアドレス OA1 は NACK されます。
- 1 : Own Address 1 有効。受信されたスレーブアドレス OA1 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10 **OA1MODE** Own Address 1 10 ビットモード

- 0 : Own Address 1は 7 ビットアドレスです。
- 1 : Own Address 1は 10 ビットアドレスです。

注 : このビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 9:8 **OA1[9:8]** : インタフェースアドレス

- 7 ビットアドレッシングモード : 無視されます。
- 10 ビットアドレッシングモード : アドレスのビット 9:8。

注 : これらのビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 7:1 **OA1[7:1]** : インタフェースアドレス

- アドレスのビット 7:1
- 注 :** これらのビットは、OA1EN=0 のときのみ書き込むことができます。

ビット 0 **OA1[0]** : インタフェースアドレス

- 7 ビットアドレッシングモード : 無視されます。
- 10 ビットアドレッシングモード : アドレスのビット 0。

注 : このビットは、OA1EN=0 のときのみ書き込むことができます。

25.7.4 Own Address 2 レジスタ (I2C_OAR2)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]				OA2[7:1]						
rw					rw				rw						

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **OA2EN** : Own Address 2 有効

- 0 : Own Address 2 無効。受信されたスレーブアドレス OA2 は NACK されます。
- 1 : Own Address 2 有効。受信されたスレーブアドレス OA2 は ACK されます。

ビット 14:11 予約済みであり、リセット値に保持する必要があります。

ビット 10:8 **OA2MSK[2:0]** : Own Address 2 マスク

- 000 : マスクなし。
- 001 : OA2[1] はマスクされ、無視されます。OA2[7:2] のみ比較されます。
- 010 : OA2[2:1] はマスクされ、無視されます。OA2[7:3] のみ比較されます。
- 011 : OA2[3:1] はマスクされ、無視されます。OA2[7:4] のみ比較されます。
- 100 : OA2[4:1] はマスクされ、無視されます。OA2[7:5] のみ比較されます。
- 101 : OA2[5:1] はマスクされ、無視されます。OA2[7:6] のみ比較されます。
- 110 : OA2[6:1] はマスクされ、無視されます。OA2[7] のみ比較されます。
- 111 : OA2[7:1] はマスクされ、無視されます。比較は行われず、すべての（予約済みを除く）7 ビット受信アドレスが確認応答されます。

注 : これらのビットは、OA2EN=0 のときのみ書き込むことができます。
OA2MSK が 0 でなくなると、予約済み I2C アドレス (0b0000xxx および 0b1111xxx) は、比較が一致した場合でも確認応答されません。

ビット 7:1 **OA2[7:1]** : インタフェースアドレス

アドレスのビット 7:1

注 : これらのビットは、OA2EN=0 のときのみ書き込むことができます。

ビット 0 予約済みであり、リセット値に保持する必要があります。



25.7.5 タイミングレジスタ (I2C_TIMINGR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
rw								rw				rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCLH[7:0]								SCLL[7:0]							
rw								rw							

ビット 31:28 **PRESC[3:0]** : タイミングプリスケアラ

このフィールドは、データのセットアップおよびホールドカウンタ ([I2C のタイミング \(615 ページ\)](#)) を参照) と SCL ハイおよびローレベルカウンタ ([I2C マスタ初期化 \(630 ページ\)](#)) を参照) に使用されるクロック周期 t_{PRESC} を生成して、I2CCLK をプリスケールするために使用されます。

$$t_{\text{PRESC}} = (\text{PRESC} + 1) \times t_{\text{I2CCLK}}$$

ビット 27:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:20 **SCLDEL[3:0]** : データセットアップ時間

このフィールドは、送信モードの SDA エッジと SCL 立ち上がりエッジの間に遅延 t_{SCLDEL} を生成するために使用されます。

$$t_{\text{SCLDEL}} = (\text{SCLDEL} + 1) \times t_{\text{PRESC}}$$

注 : t_{SCLDEL} は、 $t_{\text{SU:DAT}}$ タイミングを生成するために使用されます。

ビット 19:16 **SDADEL[3:0]** : データホールド時間

このフィールドは、送信モードの SCL 立ち下がりエッジと SDA エッジの間に遅延 t_{SDADEL} を生成するために使用されます。

$$t_{\text{SDADEL}} = \text{SDADEL} \times t_{\text{PRESC}}$$

注 : t_{SDADEL} は、 $t_{\text{HD:DAT}}$ タイミングを生成するために使用されます。

ビット 15:8 **SCLH[7:0]** : SCL ハイ周期 (マスタモード)

このフィールドは、マスタモードで SCL ハイ周期を生成するために使用されます。

$$t_{\text{SCLH}} = (\text{SCLH} + 1) \times t_{\text{PRESC}}$$

注 : t_{SCLH} は、 $t_{\text{SU:STO}}$ および $t_{\text{HD:STA}}$ タイミングを生成するためにも使用されます。

ビット 7:0 **SCLL[7:0]** : SCL ロー周期 (マスタモード)

このフィールドは、マスタモードで SCL ロー周期を生成するために使用されます。

$$t_{\text{SCLL}} = (\text{SCLL} + 1) \times t_{\text{PRESC}}$$

注 : t_{SCLL} は、 t_{BUF} および $t_{\text{SU:STA}}$ タイミングを生成するためにも使用されます。

注 : このレジスタは、I2C が無効 ($\text{PE}=0$) のときに設定する必要があります。

25.7.6 タイムアウトレジスタ (I2C_TIMEOUTR)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

アクセス : このレジスタへの書き込みアクセスが進行中のときに書き込みアクセスが発生した場合を除き、ウェイト状態なし。この場合、前の書き込みアクセスが完了するまで、2 番目の書き込みアクセスにウェイト状態が挿入されます。2 番目の書き込みアクセスの遅延は、最大 $2 \times \text{PCLK1} + 6 \times \text{I2CCLK}$ です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB [11:0]											
rw				rw											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA [11:0]											
rw			rw	rw											

ビット 31 **TEXTEN** : 拡張クロックタイムアウト有効

- 0 : 拡張クロックタイムアウト検出は無効です。
- 1 : 拡張クロックタイムアウト検出は有効です。 $t_{\text{LOW:EXT}}$ を超える累積 SCL ストレッチが I2C インタフェースによって行われると、タイムアウトエラーが検出されます (TIMEOUT=1)。

ビット 30:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:16 **TIMEOUTB[11:0]** : バスタイムアウト B

このフィールドは、累積クロック拡張タイムアウトを設定するために使用されます :
マスタモードでは、マスタ累積クロックロー拡張時間 ($t_{\text{LOW:MEXT}}$) が検出されます。
スレーブモードでは、スレーブ累積クロックロー拡張時間 ($t_{\text{LOW:SEXT}}$) が検出されます。
 $t_{\text{LOW:EXT}} = (\text{TIMEOUTB} + 1) \times 2048 \times t_{\text{I2CCLK}}$

注 : これらのビットは、**TEXTEN=0** のときのみ書き込むことができます。

ビット 15 **TIMOUTEN** : クロックタイムアウト有効

- 0 : SCL タイムアウト検出は無効です。
- 1 : SCL タイムアウト検出は有効です。SCL が t_{TIMEOUT} ($\text{TIDLE}=0$) を超えてローであるか、 t_{IDLE} ($\text{TIDLE}=1$) を超えてハイであった場合、タイムアウトエラーが検出されます (TIMEOUT=1)。

ビット 14:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TIDLE** : アイドルクロックタイムアウト検出

- 0 : TIMEOUTA は、SCL ロータイムアウトの検出に使用されます。
 - 1 : TIMEOUTA は、SCL と SDA の両方のハイタイムアウト (バスアイドル条件) の検出に使用されます。
- 注 :** このビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

ビット 11:0 **TIMEOUTA[11:0]** : バスタイムアウト A

このフィールドは、以下を設定するために使用されます :
– $\text{TIDLE}=0$ のときの SCL ロータイムアウト条件 t_{TIMEOUT}
 $t_{\text{TIMEOUT}} = (\text{TIMEOUTA} + 1) \times 2048 \times t_{\text{I2CCLK}}$
– $\text{TIDLE}=1$ のときのバスアイドル条件 (SCL と SDA の両方のハイ)
 $t_{\text{IDLE}} = (\text{TIMEOUTA} + 1) \times 4 \times t_{\text{I2CCLK}}$

注 : これらのビットは、**TIMOUTEN=0** のときのみ書き込むことができます。

注 : **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 25.3 : I2C の実装を参照してください。**



25.7.7 割り込みおよびステータスレジスタ (I2C_ISR)

アドレスオフセット : 0x18
リセット値 : 0x0000 0001
アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r							r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

- ビット 31:24 予約済みであり、リセット値に保持する必要があります。
- ビット 23:17 **ADDCODE[6:0]** : アドレス一致コード (スレーブモード)
これらのビットは、アドレス一致イベントが発生したときに (ADDR = 1)、受信したアドレスで更新されます。
10 ビットアドレスの場合、ADDCODE は 10 ビットのヘッダとその後のアドレスの 2 つの MSB を示します。
- ビット 16 **DIR** : 転送方向 (スレーブモード)
このフラグは、アドレス一致イベントが発生したときに (ADDR=1)、更新されます。
0 : 書き込み転送、スレーブはレシーバモードになります。
1 : 読み出し転送、スレーブはトランスミッタモードになります。
- ビット 15 **BUSY** : バスビジー
このフラグは、バスで通信が進行中であることを示します。START コンディションが検出されたときに、ハードウェアによってセットされます。STOP コンディションが検出されたとき、または PE=0 のときにハードウェアによってクリアされます。
- ビット 14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **ALERT** : SMBus アラート
このフラグは、SMBHEN=1 (SMBus ホスト設定)、ALERTEN=1、および SMBALERT イベント (立ち下がリエッジ) が SMBA ピンで検出されたときに、ハードウェアによってセットされます。ALERTCF ビットをセットすることによって、ソフトウェアによってクリアされます。
注: このビットは、PE=0 のとき、ハードウェアによってクリアされます。
SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。
- ビット 12 **TIMEOUT** : タイムアウトまたは t_{LOW} 検出フラグ
このフラグは、タイムアウトまたは拡張クロックタイムアウトが発生したときに、ハードウェアによってセットされます。TIMEOUTCF ビットをセットすることによって、ソフトウェアによってクリアされます。
注: このビットは、PE=0 のとき、ハードウェアによってクリアされます。
SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。



ビット 11 PECERR : 受信時の PEC エラー

このフラグは、受信した PEC が PEC レジスタの内容に一致しないときに、ハードウェアによってセットされます。正しくない PEC の受信後、NACK が自動的に送信されます。PECCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。

ビット 10 OVR : オーバーラン/アンダーラン (スレーブモード)

このフラグは、NOSTRETCH=1 のスレーブモードで、オーバーラン/アンダーランエラーが発生したときに、ハードウェアによってセットされます。OVRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

ビット 9 ARLO : アービトレーション喪失

このフラグは、アービトレーション喪失の場合に、ハードウェアによってセットされます。ARLOCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

ビット 8 BERR : バスエラー

このフラグは、ペリフェラルが転送に関与しているので、START または STOP コンディションの誤配置が検出されたときに、ハードウェアによってセットされます。このフラグは、スレーブモードのアドレスフェーズではセットされません。BERRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

ビット 7 TCR : 転送完了再ロード

このフラグは、RELOAD=1 および NBYTES データが転送されたときに、ハードウェアによってセットされます。NBYTES にゼロ以外の値が書き込まれたときにソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

このフラグは、マスタモード、または SBC ビットがセットされているときのスレーブモードのみ使用されます。

ビット 6 TC : 転送完了 (マスタモード)

このフラグは、RELOAD=0、AUTOEND=0、および NBYTES データが転送されたときに、ハードウェアによってセットされます。START ビットまたは STOP ビットがセットされたときに、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

ビット 5 STOPF : STOP 検出フラグ

このフラグは、バス上で STOP コンディションが検出され、ペリフェラルがこの転送に関与しているときに、ハードウェアによってセットされます:

– マスタとして。ただし、STOP コンディションがペリフェラルによって生成される場合。

– または、スレーブとして。ただし、ペリフェラルがこの転送中にアドレス指定されていた場合。

STOPCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

ビット 4 NACKF : 非確認応答受信フラグ

このフラグは、バイト送信後に NACK を受信したときに、ハードウェアによってセットされます。NACKCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。

ビット 3 ADDR : アドレス一致 (スレーブモード)

このビットは、受信したスレーブアドレスが有効なスレーブアドレスの 1 つに一致したときに、ハードウェアによってセットされます。ADDRCF ビットをセットすることによって、ソフトウェアによってクリアされます。

注: このビットは、 $PE=0$ のとき、ハードウェアによってクリアされます。



- ビット 2 **RXNE** : 受信データレジスタノットエンプティ (レシーバ)
このビットは、受信データが I2C_RXDR レジスタにコピーされ、読み出す準備ができたときに、ハードウェアによってセットされます。I2C_RXDR が読み出されたときにクリアされます。
注: このビットは、PE=0 のとき、ハードウェアによってクリアされます。
- ビット 1 **TXIS** : 送信割り込みステータス (トランスミッタ)
このビットは、I2C_TXDR レジスタが空であり、送信データを I2C_TXDR レジスタに書き込む必要があるときに、ハードウェアによってセットされます。次の送信データが I2C_TXDR レジスタに書き込まれたときにクリアされます。
このビットは、NOSTRETCH=1 のときのみ、ソフトウェアによって 1 を書き込んで、TXIS イベントを生成することができます (TXIE=1 の場合に割り込み、または TXDMAEN=1 の場合に DMA リクエスト)。
注: このビットは、PE=0 のとき、ハードウェアによってクリアされます。
- ビット 0 **TXE** : 送信データレジスタエンプティ (トランスミッタ)
このビットは、I2C_TXDR レジスタが空のときに、ハードウェアによってセットされます。次の送信データが I2C_TXDR レジスタに書き込まれたときにクリアされます。
このビットは、ソフトウェアによって 1 を書き込んで、送信データレジスタ I2C_TXDR をフラッシュできます。
注: このビットは、PE=0 のとき、ハードウェアによってセットされます。

25.7.8 割り込みクリアレジスタ (I2C_ICR)

アドレスオフセット : 0x1C
リセット値 : 0x0000 0000
アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERTC F	TIM OUTCF	PECCF	OVRCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDRC F	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

- ビット 31:14 予約済みであり、リセット値に保持する必要があります。
- ビット 13 **ALERTCF** : アラートフラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの ALERT フラグがクリアされます。
注: SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。
- ビット 12 **TIMOUTCF** : タイムアウト検出フラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの TIMEOUT フラグがクリアされます。
注: SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。
- ビット 11 **PECCF** : PEC エラーフラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの PECERR フラグがクリアされます。
注: SMBus 機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 25.3 : I2C の実装を参照してください。
- ビット 10 **OVRCF** : オーバーラン／アンダーランフラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの OVR フラグがクリアされます。



- ビット 9 **ARLOCF** : アービトレーション喪失フラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの ARLO フラグがクリアされます。
- ビット 8 **BERRCF** : バスエラーフラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの BERRF フラグがクリアされます。
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **STOPCF** : STOP 検出フラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの STOPF フラグがクリアされます。
- ビット 4 **NACKCF** : 非確認応答フラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの ACKF フラグがクリアされます。
- ビット 3 **ADDRCF** : アドレス一致フラグクリア
このビットに 1 を書き込むと、I2C_ISR レジスタの ADDR フラグがクリアされます。このビットに 1 を書き込むと、I2C_CR2 レジスタの START ビットもクリアされます。
- ビット 2:0 予約済みであり、リセット値に保持する必要があります。

25.7.9 PEC レジスタ (I2C_PECR)

アドレスオフセット : 0x20
リセット値 : 0x0000 0000
アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
								r							

- ビット 31:8 予約済みであり、リセット値に保持する必要があります。
- ビット 7:0 **PEC[7:0]** パケットエラーチェックレジスタ
PECEN=1 のとき、このフィールドは内部 PEC を含みます。
PEC は、PE=0 のとき、ハードウェアによってクリアされます。

注 : **SMBus 機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって強制的に 0x00000000 に設定されます。セクション 25.3 : I2C の実装を参照してください。**

25.7.10 受信データレジスタ (I2C_RXDR)

アドレスオフセット : 0x24
リセット値 : 0x0000 0000
アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r							

ビット 31:8 予約済みであり、リセット値に保持する必要があります。
ビット 7:0 **RXDATA[7:0]** 8 ビット受信データ
I²C バスから受信したデータバイト。

25.7.11 送信データレジスタ (I2C_TXDR)

アドレスオフセット : 0x28
リセット値 : 0x0000 0000
アクセス : ウェイト状態なし

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
								rw							

ビット 31:8 予約済みであり、リセット値に保持する必要があります。
ビット 7:0 **TXDATA[7:0]** 8 ビット送信データ
I²C バスに送信されるデータバイト。
注: これらのビットは、TXE=1 のときのみ書き込むことができます。

25.7.12 I2C レジスタマップ

次の表に、I2C のレジスタマップとリセット値を示します。

表 93. I2C レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x0	I2C_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE				TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	リセット値									0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x4	I2C_CR2	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]							NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]													
	リセット値						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x8	I2C_OAR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]												
	リセット値																	0					0	0	0	0	0	0	0	0	0	0	0			
0xC	I2C_OAR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]					Res.							
	リセット値																	0					0	0	0	0	0	0	0	0	0	0				
0x10	I2C_TIMINGR	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]				SCLH[7:0]					SCLL[7:0]													
	リセット値	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	I2C_TIMEOUTR	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]										TIMEOUTEN	Res.	TIDLE	TIMEOUTA[11:0]																	
	リセット値	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	I2C_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE			
	リセット値									0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1		
0x1C	I2C_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMEOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.			
	リセット値																			0	0	0	0	0	0			0	0	0						
0x20	I2C_PECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]											
	リセット値																								0	0	0	0	0	0	0	0	0	0		
0x24	I2C_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]											
	リセット値																								0	0	0	0	0	0	0	0	0	0		
0x28	I2C_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]												
	リセット値																								0	0	0	0	0	0	0	0	0	0		

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

26 USART (Universal synchronous asynchronous receiver transmitter)

26.1 概要

USART (Universal Synchronous Asynchronous Receiver Transmitter) を使用すると、業界標準の NRZ 非同期シリアルデータフォーマットを必要とする外部機器と柔軟に全二重データ交換を行うことができます。USART は、プログラム可能なボーレートジェネレータを使用して、非常に広範囲のボーレートを提供します。

USART は、マルチプロセッサ通信に加え、同期式単方向通信と半二重単線通信をサポートします。また、LIN (Local Interconnect Network)、スマートカードプロトコル、および IrDA (Infrared Data Association) SIR ENDEC 仕様、およびモデム操作 (CTS/RTS) もサポートします。

マルチバッファ設定で DMA (直接メモリアクセス) を使用することによって、高速データ通信が可能です。

26.2 USART の主な機能

- 全二重非同期通信
- NRZ 標準フォーマット (マーク/スペース)
- 16 倍または 8 倍に設定可能なオーバーサンプリング方式によって、速度とクロック誤差の間の柔軟性を実現
- 最大 6 Mbit/s (クロック周波数が 48 MHz のとき) まで共通にプログラム可能な送信および受信ボーレートと 8 倍のオーバーサンプリング
- 以下を許可するデュアルクロックドメイン:
 - USART 機能と STOP モードからのウェイクアップ
 - PCLK 再プログラミングから独立した便利なボーレートプログラミング
- 自動ボーレート検出
- プログラム可能なデータワード長 (7 または 8 または 9 ビット)
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 設定可能なストップビット (1 または 2 個のストップビット)
- 同期通信のための同期モードとクロック出力
- 単線半二重通信
- DMA を使用した連続通信
- 送受信バイトは集中型 DMA を使用して専用 SRAM にバッファリング
- トランスマッタとレシーバ用に個別の有効ビット
- 送信と受信の信号極性を個別に制御
- スワップ可能な Tx/Rx ピン設定
- モデムと RS-485 トランシーバのハードウェアフロー制御

- 通信制御／エラー検出フラグ
- パリティ制御：
 - － パリティビットの送信
 - － 受信したデータバイトのパリティ検査
- 14 の割り込みソース（フラグ付き）
- マルチプロセッサ通信
アドレスが一致しない場合、USART はミュートモードに入ります。
- ミュートモードからのウェイクアップ（アイドルライン検出またはアドレスマーク検出による）

26.3 USART の拡張機能

- LIN マスタの同期ブレーク送信機能と LIN スレーブのブレーク検出機能
 - － USART が LIN 用にハードウェア設定されている場合、13 ビットのブレーク生成と 10/11 ビットのブレーク検出
- 通常モードで 3/16 ビット期間をサポートする IrDA SIR エンコーダデコーダ
- スマートカードモード
 - － ISO/IEC 7816-3 標準で定義されているスマートカードの T=0 および T=1 非同期プロトコルをサポート
 - － スマートカード動作に 0.5 個および 1.5 個のストップビット
- Modbus 通信のサポート
 - － タイムアウト機能
 - － CR/LF キャラクタ認識

26.4 USART の実装

表 94. STM32F0xx USART 実装

USART モード/ 機能 ⁽¹⁾	STM32F03x	STM32F05x		STM32F04x		STM32F07x		STM32F09x		
	USART1	USART1	USART2	USART1	USART2	USART1/USART2	USART3/USART4	USART1/USART2/ USART3	USART4	USART5/USART6/ USART7/USART8
モデムのハードウェア フロー制御	X	X	X	X	X	X	X	X	X	-
DMA を使用した連続通信	X	X	X	X	X	X	X	X	X	X
マルチプロセッサ通信	X	X	X	X	X	X	X	X	X	X
同期モード	X	X	X	X	X	X	X	X	X	X
スマートカードモード	X ⁽²⁾	X ⁽²⁾	-	X ⁽³⁾	-	X ⁽³⁾	-	X ⁽³⁾	-	-
単線半二重通信	X	X	X	X	X	X	X	X	X	X
Ir SIR ENDEC ブロック	X	X	-	X	-	X	-	X	-	-
LIN モード	X	X	-	X	-	X	-	X	-	-
デュアルクロックドメイン と STOP モードからの ウェイクアップ	X	X	-	X	-	X	-	X	-	-
レシーバタイムアウト 割り込み	X	X	-	X	-	X	-	X	-	-
Modbus 通信	X	X	-	X	-	X	-	X	-	-
自動ボーレート検出 (サポートされている モード)	2 (モード 0/1)	2 (モード 0/1)	-	4	-	4	-	4	-	-
ドライバインネブル	X	X	X	X	X	X	X	X	X	X
USART データ長	8 および 9 ビット			7、8、および 9 ビット						

1. X: サポートされています。
2. UE ビット = 0 のとき、SCLK 出力は無効です。
3. UE ビット値に関係なく、CLKEN = 1 の場合に SCLK は常に使用可能です。

26.5 USART の機能詳細

USART の双方向通信には、少なくとも 2 本のピンが必要です。すなわち、受信データ入力 (RX) と送信データ出力 (TX) です。

- **RX** : 受信データ入力。
これはシリアルデータ入力です。データ復旧にはオーバーサンプリング技術が使用され、有効な入力データとノイズを区別しています。
- **TX** : 送信データ出力。
トランスミッタが無効なときは、出力ピンは入出力ポート設定に戻ります。トランスミッタが有効で、送信すべきデータがないとき、TX ピンはハイレベルになります。単線 およびスマートカードモードでは、この I/O はデータの送受信に使用されます。

シリアルデータは、通常の USART モードでは、これらのピンを通じて送受信されます。フレームは、以下で構成されます。

- 送受信前のアイドルライン
- スタートビット
- データワード (7、8、または 9 ビット) LSBファースト
- フレームの完了を示す 0.5 個、1 個、1.5 個、2 個のストップビット
- USART インタフェースはボーレートジェネレータを使用
- ステータスレジスタ (USARTx_ISR)
- 受信および送信データレジスタ (USARTx_RDR、USARTx_TDR)
- ボーレートレジスタ (USARTx_BRR)
- ガードタイムレジスタ (USARTx_GTPR) - スマートカードモードの場合

各ビットの定義については、[セクション 26.8 : USART レジスタ \(722 ページ\)](#) を参照してください。

同期モードおよびスマートカードモードでインタフェースするには、以下のピンが必要です。

- **SCLK** : クロック出力。このピンは、SPI マスタモードに対応する同期送信用のトランスミッタデータクロックを出力します (スタートビットとストップビットのクロックパルスはなく、ソフトウェアオプションで最後のデータビットのクロックパルスを送信します)。これと並行して、RX でデータを同期受信できます。これを使用して、シフトレジスタを持つペリフェラル (例 : LCD ドライバ) を制御できます。クロックの位相と極性は、ソフトウェアでプログラム可能です。スマートカードモードでは、SCLK 出力はスマートカードにクロックを供給できます。

RS232 ハードウェアフロー制御モードでは、以下のピンが必要です。

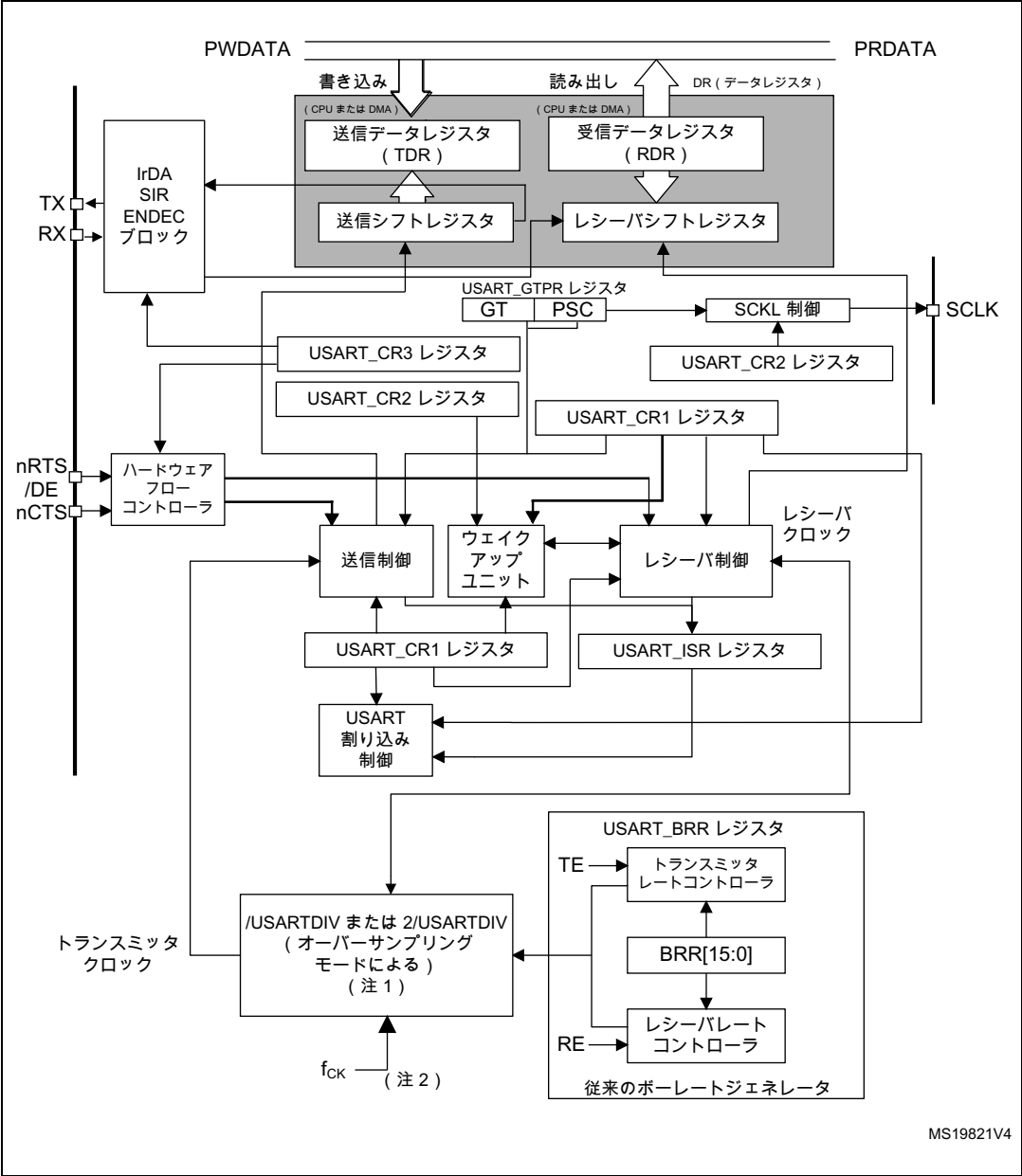
- **nCTS** : Clear To Send は、ハイレベルのとき、現在の転送の終わりにデータ送信をブロックします。
- **nRTS** : Request To Send は、ローレベルのとき、USART がデータを受信する準備ができたことを示します。

RS485 ハードウェアフロー制御モードでは、以下のピンが必要です。

- **DE** : Driver Enable は、外部トランシーバの送信モードを有効にします。

注 : **DE と nRTS は同じピンを共有します。**

図 241. USART のブロック図



MS19821V4

1. USARTx_BRR レジスタの USARTDIV のコーディングの詳細については、[セクション 26.5.4: ボーレート生成](#)を参照してください。
2. f_{CK} は、 f_{LSE} 、 f_{HSI} 、 f_{PCLK} 、 f_{SYS} のいずれかです。

26.5.1 USART キャラクタの説明

ワード長は、USARTx_CR1 レジスタの M[1:0] ビットをプログラムすることによって、7、8、または 9 ビットとして選択できます（図 242 を参照）。

- 7 ビットのキャラクタ長 : M[1:0] = 10
- 8 ビットのキャラクタ長 : M[1:0] = 00
- 9 ビットのキャラクタ長 : M[1:0] = 01

注 : 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ボーレート (0x7F および 0x55 フレーム検出) はサポートされません。7 ビットモードは、一部の USART でのみサポートされます。

デフォルト設定では、信号 (TX または RX) はスタートビットの処理中ではロー状態です。また、ストップビットの処理中にはハイ状態です。

これらの値は、極性設定制御により、各信号について個別に反転できます。

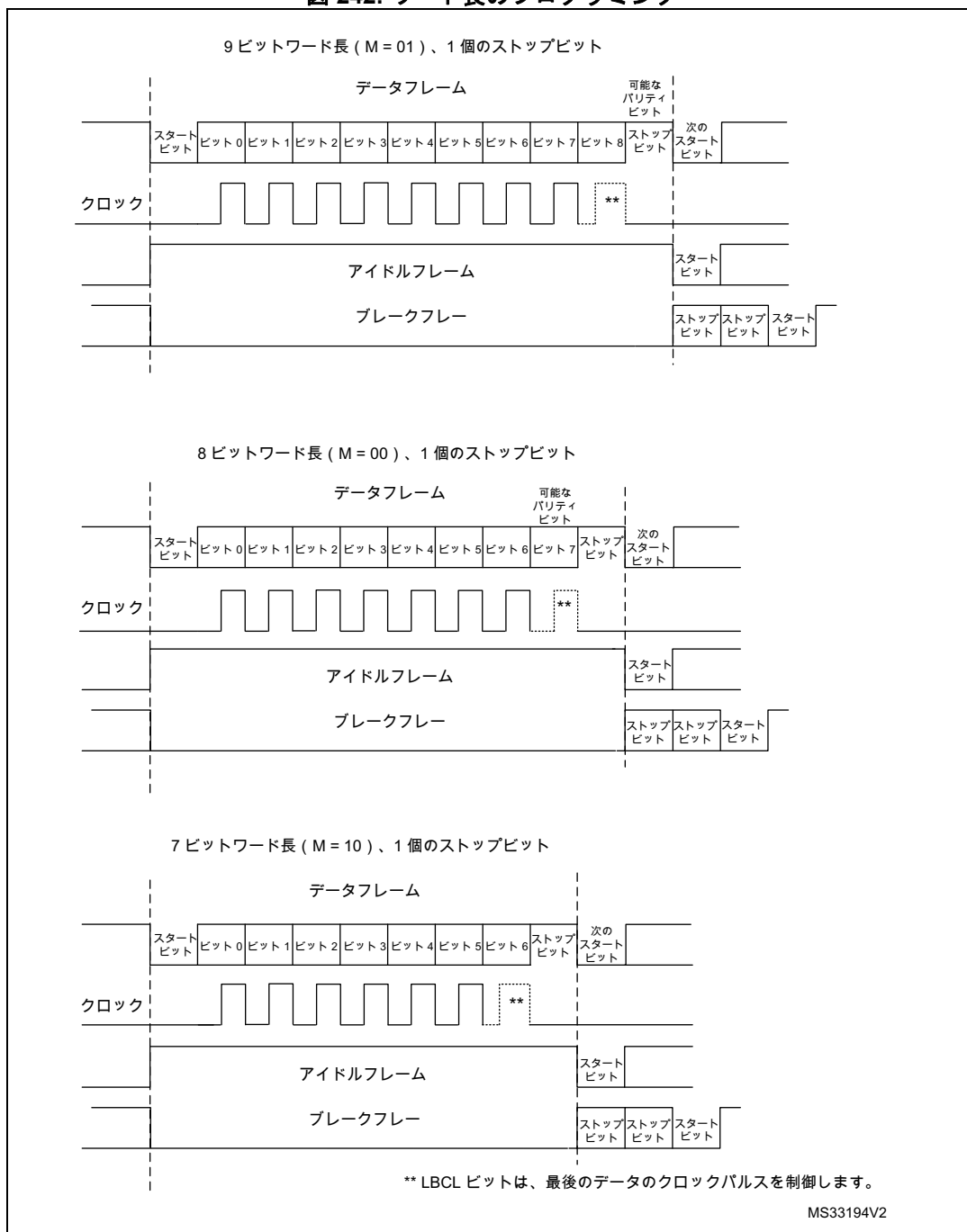
アイドルキャラクタは、すべてが「1」のフレームとして解釈されます（「1」の数にはストップビットの数が含まれます）。

ブレイクキャラクタは、フレーム周期中に「0」を受信することと解釈されます。ブレイクフレームの終了時、トランスミッタは 2 個のストップビットを挿入します。

送信と受信は、共通のボーレートジェネレータによって駆動され、それぞれのクロックは、トランスミッタとレシーバの有効ビットがそれぞれセットされたときに生成されます。

各ブロックの詳細を次に示します。

図 242. ワード長のプログラミング



26.5.2 トランスミッタ

トランスミッタは、M ビットのステータスに応じて、7、8、または 9 ビットのデータワードを送信できます。トランスミッタ機能を有効にするには、送信イネーブルビット (TE) をセットする必要があります。送信シフトレジスタ内のデータは TX ピンで出力され、対応するクロックパルスは SCLK ピンで出力されます。

キャラクタ送信

USART 送信時、データはLSBファースト（デフォルト設定）で TX ピンにシフトアウトされます。このモードでは、USARTx_TDR レジスタは、内部バスと送信シフトレジスタの間のバッファ（TDR）で構成されます（図 241 を参照）。

各キャラクタの前には、スタートビット（1 ビット周期、論理レベルロー）があります。キャラクタは、設定可能な数のストップビットで終端されます。

USART では、0.5、1、1.5、および 2 個のストップビットがサポートされます。

注： 送信データを USARTx_TDR に書き込む前に、TE ビットをセットする必要があります。

データの送信中に TE ビットをリセットしないでください。送信中に TE ビットをリセットすると、ボーレートカウンタが停止されるため、TX ピンのデータが破壊されます。送信中のデータは失われます。

TE ビットが有効になると、アイドルフレームが送信されます。

設定可能なストップビット

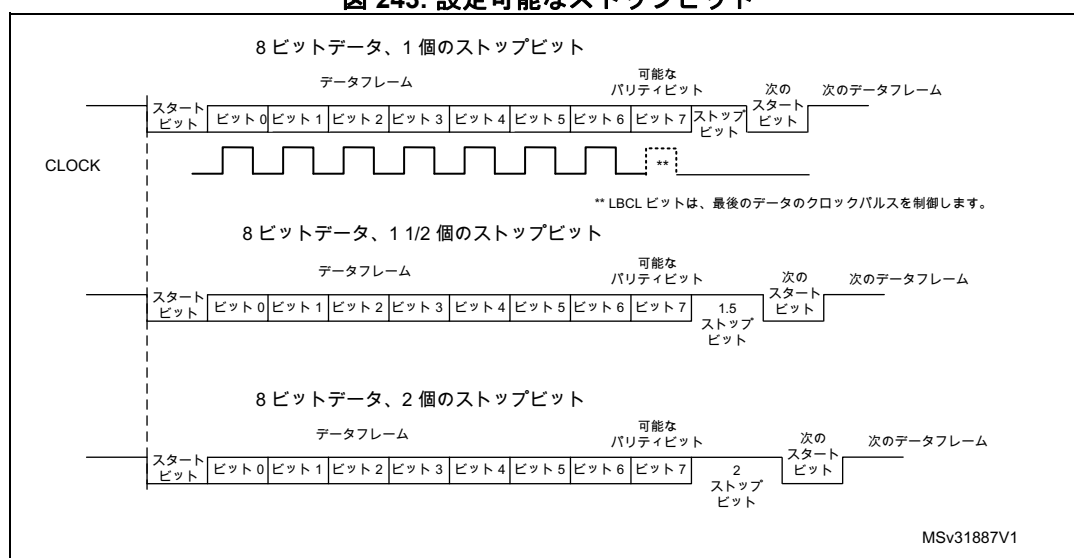
各キャラクタとともに送信されるストップビットの数は、制御レジスタ 2 のビット 13、12 でプログラミングできます。

- **1 個のストップビット：** ストップビット数のデフォルト値です。
- **2 個のストップビット：** 通常の USART モード、単線モード、およびモデムモードでサポートされます。
- **ストップビット 1, 5 個：** スマートカードモードで使用されます。
- **0.5 個のストップビット：** スマートカードモードでのデータの受信に使用されます。

アイドルフレームの送信にはストップビットが含まれます。

ブ레이크送信は、10 個のロービット（M[1:0] = 00 のとき）または 11 個のロービット（M[1:0] = 01 のとき）または 9 個のロービット（M[1:0] = 10 のとき）の後に 2 個のストップビットが続きます（図 243 を参照）。長いブ레이크（9/10/11 個のロービットを超える長さのブ레이크）を送信することはできません。

図 243. 設定可能なストップビット



キャラクタ送信手順

1. ワード長を定義するには、USARTx_CR1 の M ビットをプログラムします。
2. USARTx_BRR レジスタを使用して、目的のボーレートを選択します。
3. USARTx_CR2 レジスタでストップビットの数をプログラミングします。
4. USARTx_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USARTx_CR3 レジスタの DMA 有効 (DMAT) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
6. USARTx_CR1 の TE ビットをセットして、最初の送信としてアイドルフレームを送信します。
7. 送信するデータを USARTx_TDR レジスタに書き込みます (これによって TXE ビットがクリアされます)。シングルバッファの場合、送信される各データにこれを繰り返します。
8. USARTx_TDR レジスタに最後のデータを書き込んだ後、TC=1 になるまで待ちます。これは、最後のフレームの送信が完了したことを示します。これは、たとえば最後の送信内容の破壊を避けるために、USART が無効になったり HALT モードに入ったりする場合に必要です。

コード例については、「付録」セクション [A.19.1 : USART トランスミッタ設定のコード例](#) を参照してください。

1 バイト通信

TXE ビットは、常に送信データレジスタへの書き込みによってクリアされます。

TXE ビットは、ハードウェアによってセットされ、次を示します。

- データは USARTx_TDR レジスタからシフトレジスタへ移動され、データ送信が開始しています。
- USARTx_TDR レジスタは空です。
- 次のデータを、前のデータに上書きせずに、USARTx_TDR レジスタに書き込みます。

コード例については、「付録」セクション [A.19.2 : USART 送信バイトのコード例](#) を参照してください。

TXEIE ビットがセットされている場合、このフラグは割り込みを生成します。

送信が行われている場合、USARTx_TDR レジスタへの書き込み命令によってデータが TDR レジスタに格納され、次に、データが現在進行中の送信の最後にシフトレジスタにコピーされます。

送信が行われていないときには、USARTx_TDR レジスタへの書き込み命令によってデータがシフトレジスタに格納され、データ送信が開始され、TXE ビットがセットされます。

フレームが送信され (ストップビットの後)、TXE ビットがセットされると、TC ビットはハイレベルになります。USARTx_CR1 レジスタの TCIE ビットがセットされると、割り込みが生成されます。

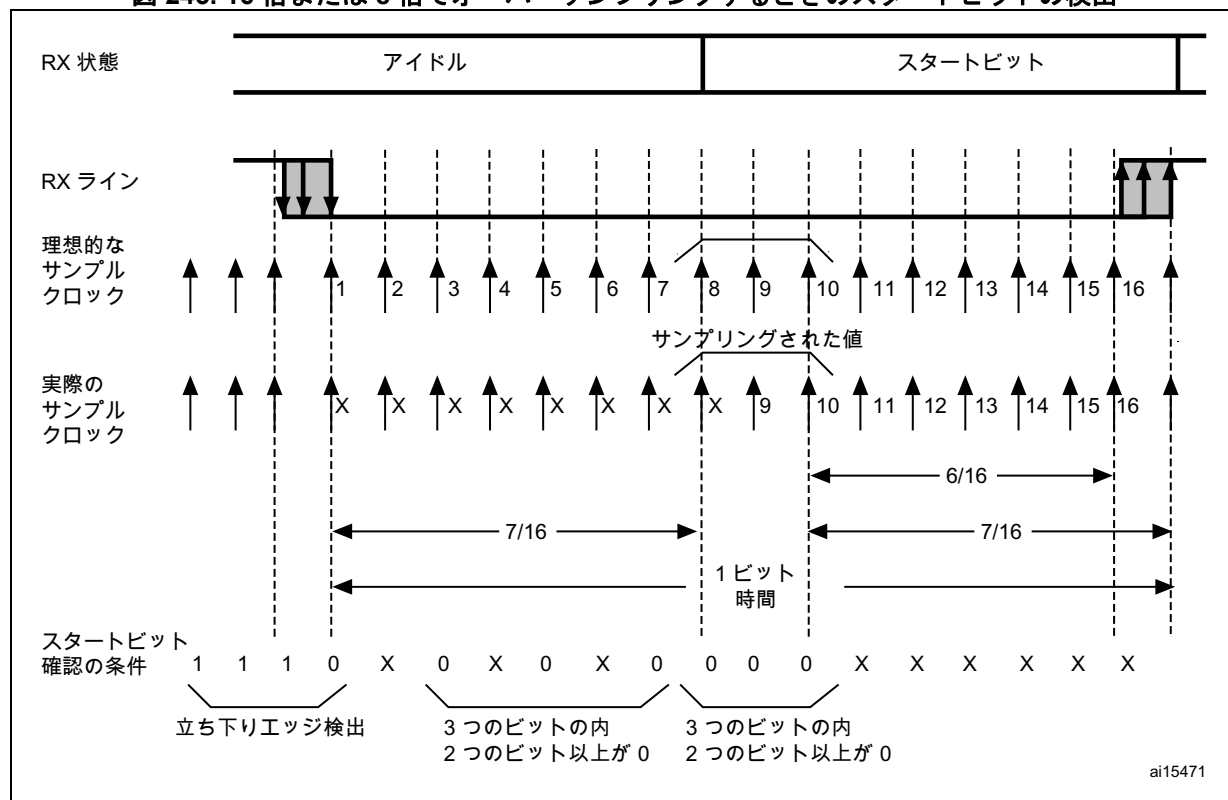
USARTx_TDR レジスタに最後のデータを書き込んだ後は、USART を無効にしたり、マイクロコントローラを低電力モードにする前に TC=1 になるまで待つ必要があります ([図 244 : 送信時の TC/TXE の動作](#) を参照)。

The diagram illustrates the USART TXE operation across three frames. The TX Line shows the transmission of data frames. The TXE Flag indicates when the transmit buffer is empty. The USART_DR shows the data being transmitted. The TC Flag indicates when the transmission is complete. The diagram shows that the TXE flag is set by hardware and cleared by software. The TC flag is set by hardware and cleared by software. The TXE flag is set by hardware and cleared by software. The TC flag is set by hardware and cleared by software.

Figure 2-44: USART TXE Operation

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や損害などに付きましては如何なる責任にも負いません。

図 245. 16 倍または 8 倍でオーバーサンプリングするときのスタートビットの検出



注： シーケンスが完了していない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻って（フラグはセットされません）、立ち下がりエッジを待ちます。

サンプリングされた 3 つのビットが 0 の場合（第 3、第 5、および第 7 ビットでの最初のサンプリングで 3 つのビットが 0 であり、第 8、第 9、および第 10 ビットでの 2 回目のサンプリングでも 3 つのビットが 0）、スタートビットが確認されます（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）。

次の場合、スタートビットは確認されますが（RXNE フラグがセットされ、RXNEIE=1 の場合は割り込みが生成されます）、NF ノイズフラグがセットされます。

a. 両方のサンプリングについて、サンプリングされた 3 ビットのうちの 2 ビットが 0 の場合（第 3、第 5、および第 7 ビットのサンプリングと、第 8、第 9、および第 10 ビットのサンプリング）。

または

b. いずれか一方のサンプリングで（第 3、第 5、および第 7 ビットでのサンプリングまたは第 8、第 9、および第 10 ビットでのサンプリング）、3 つのビットのうち 2 つが 0 の場合。

a. または b. のいずれの条件も満たされない場合、スタートビットの検出は中止され、レシーバはアイドル状態に戻ります（フラグはセットされません）。

キャラクタの受信

USART の受信時には、データは RX ピンを通じて LSB ファーストでシフトインされます。このモードでは、USARTx_RDR レジスタは、内部バスと受信シフトレジスタの間のバッファ (RDR) で構成されます。

キャラクタ受信手順

1. ワード長を定義するには、USARTx_CR1 の M ビットをプログラムします。
2. ボーレートレジスタ USARTx_BRR を使用して、目的のボーレートを選択します。
3. USARTx_CR2 レジスタでストップビットの数をプログラミングします。
4. USARTx_CR1 レジスタの UE ビットに 1 を書き込んで、USART を有効にします。
5. マルチバッファ通信を行う場合には、USARTx_CR3 レジスタの DMA 有効 (DMAR) を選択します。マルチバッファ通信の説明に基づいて、DMA レジスタを設定します。
6. USARTx_CR1 レジスタの RE ビットをセットします。これによってレシーバが有効になり、スタートビットの検索を開始します。

コード例については、「付録」セクション [A.19.4 : USART レシーバ設定のコード例](#) を参照してください。

キャラクタが受信されると、

- RXNE ビットは、シフトレジスタの内容が RDR レジスタに転送されたことを示すためにセットされます。言い換えると、データは受信され、読み出し可能です (関連するエラーフラグも同様です)。
- RXNEIE ビットがセットされていた場合、割り込みが生成されます。
- 受信中にフレームエラー、ノイズまたはオーバーランエラーが検出された場合、エラーフラグをセットできます。RXNE によって PE フラグもセットできます。
- マルチバッファでは、RXNE はバイト受信ごとにセットされ、受信データレジスタの DMA 読み出しによってクリアされます。
- シングルバッファモードでは、RXNE ビットのクリアは、ソフトウェアによる USARTx_RDR レジスタの読み出しによって行われます。RXNE フラグは、USARTx_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。オーバーランエラーを避けるには、次のキャラクタの受信が終了する前に、RXNE ビットをクリアする必要があります。

コード例については、「付録」セクション [A.19.5 : USART 受信バイトのコード例](#) を参照してください。

ブレークキャラクタ

ブレークキャラクタを受信すると、USART はブレークキャラクタをフレーミングエラーとして処理します。

アイドルキャラクタ

アイドルフレームが検出されると、受信データキャラクタの場合と同じ手順が行われ、IDLEIE ビットがセットされていた場合は割り込みが生成されます。

オーバーランエラー

RXNE ビットがリセットされていないときにキャラクタを受信すると、オーバーランエラーが発生します。RXNE ビットがクリアされない限り、データをシフトレジスタから RDR レジスタに転送することはできません。

RXNE フラグは、バイトを受信するたびにセットされます。次のデータを受信したときに RXNE フラグがセットされていた場合、または前回の DMA リクエストがまだ処理されていない場合、オーバーランエラーが発生します。オーバーランエラーが発生すると、

- ORE ビットがセットされます。
- RDR の内容は失われません。USARTx_RDR の読み出しが行われると、前のデータが使用できません。
- シフトレジスタは上書きされます。その後、オーバーラン中に受信されたデータは失われます。
- RXNEIE ビットがセットされているか、EIE ビットがセットされている場合、割り込みが生成されます。
- ORE ビットは、ICR レジスタの ORECF ビットをセットすることによってリセットされます。

注： *ORE ビットがセットされた場合、少なくとも 1 個のデータが失われています。この場合、次の 2 つの可能性がありえます。*

- RXNE=1 の場合、最後の有効なデータが受信レジスタ RDR に格納され、読み出すことができます。

- RXNE=0 の場合、最後の有効なデータはすでに読み出されたので、RDR には読み出すべきものがないことを意味します。このケースは、有効な最後のデータが RDR で読み出されると同時に新しい（そして失われた）データが受信されると発生します。

クロックソースと適切なオーバーサンプリング方式の選択

クロックソースの選択は、クロック制御システムを通じて行われます（リセットおよびクロック制御（RCC）のセクションを参照）。クロックソースは、USART を有効にする（UE ビットをセットする）前に選ぶ必要があります。

クロックソースの選択は、2 つの基準に従って行われる必要があります。

- USART を低電力モードで使用できること
- 通信速度

クロックソース周波数は、 f_{CK} です。

デュアルクロックドメインと STOP モードからのウェイクアップがサポートされるときには、クロックソースは次のソースの 1 つにできます。PCLK（デフォルト）、LSE、HSI、または SYSCLK。そうでない場合、USART のクロックソースは PCLK です。

LSE または HSI をクロックソースとして選択すると、USART は MCU が低電力モードのときにデータを受信できます。受信データとウェイクアップモードの選択に応じて、USART より必要なときに MCU をウェイクアップし、ソフトウェアが USARTx_RDR レジスタを読み出すことによって、または DMA によって受信データを転送します。

他のクロックソースの場合、USART 通信を可能にするためには、システムを有効にする必要があります。

通信速度の範囲（特に最大通信速度）もクロックソースによって決まります。

レシーバは有効な受信データとノイズを区別して、データを復旧するユーザ設定可能なさまざまなオーバーサンプリング技術を実装しています。これにより、最大通信速度とノイズ／クロック精度の耐性の間でトレードオフができます。

オーバーサンプリング方式は、USARTx_CR1 レジスタの OVER8 ビットをプログラムすることによって選択でき、ボーレートクロックの 16 倍または 8 倍にすることができます (図 246 および図 247)。

アプリケーションに応じて、

- 高速 (最大 $f_{CK}/8$) を達成するには、8 倍 (OVER8=1) のオーバーサンプリングを選択します。この場合、クロック偏差に対するレシーバの最大許容誤差は軽減されます (セクション 26.5.5: クロック偏差に対する USART レシーバの許容誤差 (697 ページ) を参照)。
- クロック偏差に対するレシーバの許容誤差を増やすには、16 倍 (OVER8=0) のオーバーサンプリングを選択します。この場合、最大速度は $f_{CK}/16$ に制限されます。 f_{CK} はクロックソース周波数です。

論理レベルの評価方法を選択するには、USARTx_CR3 レジスタの ONEBIT ビットをプログラミングします。これには 2 つのオプションがあります。

- 受信されたビットの中央にある 3 つのサンプルの多数決。この場合、多数決に使用された 3 つのサンプルが等しくないとき、NF ビットがセットされます。
- 受信されたビットの中央にある 1 つのサンプル。

アプリケーションに応じて、

- ノイズの多い環境で操作するときは 3 つのサンプルの多数決方式 (ONEBIT=0) を選択します。そしてノイズが検出された場合 (図 95 を参照) は、サンプリング中にグリッチが発生していることとなりますので、そのデータを除去します。
- ラインがノイズフリーであるときは、1 つのサンプルによる方式 (ONEBIT=1) を選択し、クロック偏差に対するレシーバの許容誤差を増やします (セクション 26.5.5: クロック偏差に対する USART レシーバの許容誤差 (697 ページ) を参照)。この場合、NF ビットはセットされません。

フレーム内でノイズが検出された場合：

- RXNE ビットの立ち上がりエッジで、NF ビットがセットされます。
- 無効なデータがシフトレジスタから USARTx_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USARTx_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

NF ビットは、ICR レジスタの NCFE ビットをセットすることによってリセットされます。

注： 8 倍のオーバーサンプリングは、LIN、スマートカード、および IrDA の各モードでは使用できません。これらのモードでは、OVER8 ビットはハードウェアによって 0 に固定されています。

図 246. データサンプリング (16 倍のオーバーサンプリング)

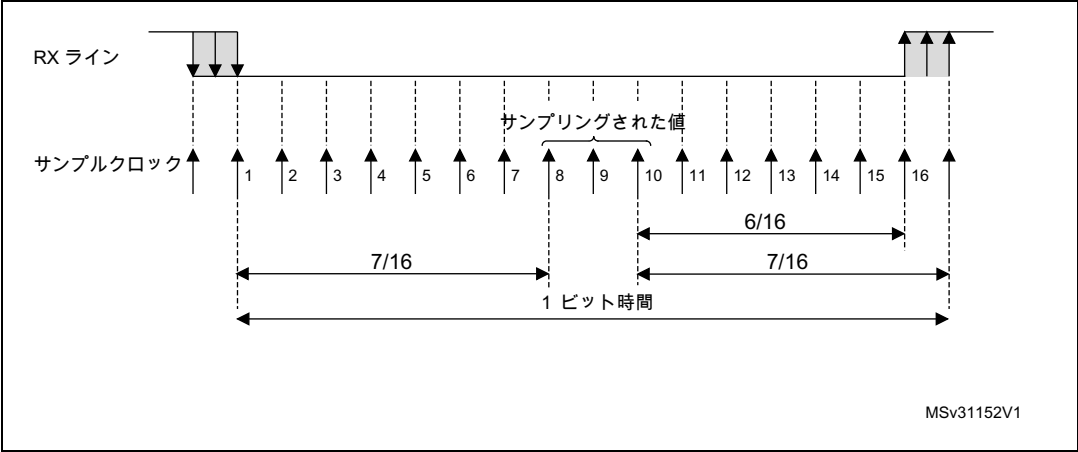


図 247. データサンプリング (8 倍のオーバーサンプリング)

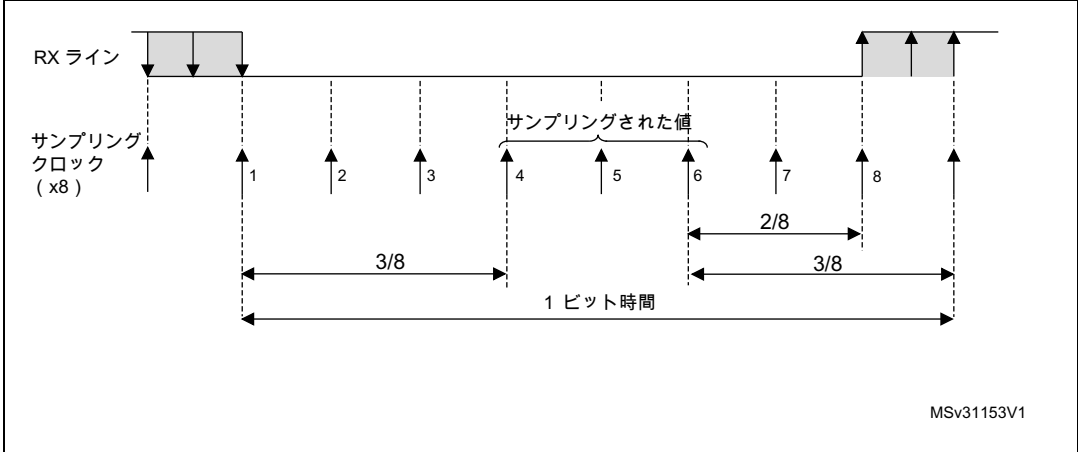


表 95. サンプリングされたデータからのノイズ検出

サンプリングされた値	NE ステータス	受信ビットの値
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

フレーミングエラー

フレーミングエラーは、次の場合に検出されます。

非同期化または過剰なノイズのため、受信時に予想されたタイミングでストップビットが認識されない場合。

フレーミングエラーが検出された場合：

- FE ビットがハードウェアによってセットされます。
- 無効なデータがシフトレジスタから USARTx_RDR レジスタへ転送されます。
- シングルバイト通信の場合、割り込みは生成されません。ただし、このビットは、割り込みを生成する RXNE ビットと同時に立ち上がります。マルチバッファ通信の場合、USARTx_CR3 レジスタの EIE ビットがセットされている場合に割り込みが発行されます。

USARTx_ICR レジスタの FECF に 1 を書き込むことによって、FE ビットがリセットされます。

受信時の設定可能なストップビット

受信するストップビット数は、制御レジスタ 2 の制御ビットを通じて設定でき、通常モードでは 1 または 2、スマートカードモードでは 0.5 または 1.5 にできます。

- **0.5 個のストップビット (スマートカードモードでの受信) :** 0.5 個のストップビットでは、サンプリングは行われません。したがって、0.5 個のストップビットが選択されている場合、フレーミングエラーやブレイクフレームは検出されません。
- **1 個のストップビット :** ストップビット 1 個のサンプリングは、8 番目、9 番目、および 10 番目のサンプルで行われます。
- **1.5 個のストップビット (スマートカードモード) :** スマートカードモードでの送信時は、デバイスは、データが正しく送信されたことをチェックする必要があります。したがって、レシーバブロックを有効にし (USARTx_CR1 レジスタの RE=1)、ストップビットをチェックして、スマートカードがパリティエラーを検出したかどうかをテストする必要があります。パリティエラーが発生した場合、スマートカードはサンプリング時のデータ信号を強制的にローレベルにします (これは、フレーミングエラーとしてフラグされる NACK 信号です)。その後、1.5 個のストップビットの最後に、RXNE ビットとともに FE フラグがセットされます。ストップビット 1.5 個のサンプリングは、16 番目、17 番目、および 18 番目のサンプルで (ストップビットの開始から 1 ボークロック周期後に) 行われます。1.5 個のストップビットは、2 つの部分に分解できます。すなわち、何も起こらない 0.5 ボークロック周期と、途中でサンプリングが行われる通常の 1 ストップビット周期です。詳細については、[セクション 26.5.13 : スマートカードモード \(708 ページ\)](#) を参照してください。
- **2 個のストップビット :** ストップビット 2 個のサンプリングは、最初のストップビットの 8 番目、9 番目、および 10 番目のサンプルで行われます。最初のストップビットでフレーミングエラーが検出された場合、フレーミングエラーフラグがセットされます。2 番目のストップビットでは、フレーミングエラーの検査は行われません。RXNE フラグは、最初のストップビットの最後でセットされます。

26.5.4 ボーレート生成

レシーバとトランスミッタ (Rx と Tx) のボーレートは、USARTx_BRR レジスタでプログラムされたものと同じ値に設定されます。

Equation 1 : 標準 USART のボーレート (SPI モードを含む) (OVER8 = 0 または 1)

16 倍のオーバーサンプリングの場合、等式は次のとおりです。

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{\text{USARTDIV}}$$

8 倍のオーバーサンプリングの場合、等式は次のとおりです。

$$\text{Tx/Rx ボー} = \frac{2 \cdot f_{\text{CK}}}{\text{USARTDIV}}$$

Equation 2 : スマートカード、LIN、および IrDA モードのボーレート (OVER8 = 0)

スマートカード、LIN、および IrDA モード。16 倍のオーバーサンプリングのみをサポート :

$$\text{Tx/Rx ボー} = \frac{f_{\text{CK}}}{\text{USARTDIV}}$$

USARTDIV は、符号なしの固定小数点数であり、USARTx_BRR レジスタでコード化されます。

- OVER8 = 0 のとき、BRR = USARTDIV です。
- OVER8 = 1 のとき、
 - BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。
 - BRR[3] は、クリアされたままにする必要があります。
 - BRR[15:4] = USARTDIV[15:4]

注 : USARTx_BRR レジスタへの書き込み操作の後、ボーカウンタはボーレジスタの新しい値に更新されます。したがって、通信中はボーレートレジスタの値を変更しないでください。

16 倍または 8 倍のオーバーサンプリングの場合、USARTDIV は 16d 以上である必要があります。

USARTx_BRR レジスタの値から USARTDIV を得る方法

Example 1

$f_{\text{CK}} = 8 \text{ MHz}$ で 9600 ボーを得るには、

- 16 倍のオーバーサンプリングの場合 :
 $\text{USARTDIV} = 8\,000\,000/9600$
 $\text{BRR} = \text{USARTDIV} = 833\text{d} = 0341\text{h}$
- 8 倍のオーバーサンプリングの場合 :
 $\text{USARTDIV} = 2 \cdot 8\,000\,000/9600$
 $\text{USARTDIV} = 1666,66 \text{ (1667d = 683h)}$
 $\text{BRR}[3:0] = 3\text{h} \ll 1 = 1\text{h}$
 $\text{BRR} = 0\text{x}681$

Example 2

- $f_{CK} = 48 \text{ MHz}$ で 921.6 kbaud を得るには、
- 16 倍のオーバーサンプリングの場合 :
USARTDIV = 48 000 000/921 600
BRR = USARTDIV = 52d = 34h
 - 8 倍のオーバーサンプリングの場合 :
USARTDIV = 2 * 48 000 000/921 600
USARTDIV = 104 (104d = 68h)
BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h
BRR = 0x64

表 96. 16 倍または 8 倍のオーバーサンプリングの両方の場合の $f_{CK} = 48 \text{ MHz}$ でのプログラムされたボーレートの誤差計算⁽¹⁾

ボーレート		16 倍のオーバーサンプリング (OVER8=0)			8 倍のオーバーサンプリング (OVER8=1)		
S.No	目標	実際	BRR	誤差 (%) = (計算値 - 目標値) ボーレート / 目標ボーレート	実際	BRR	誤差
2	2.4 KBps	2.4 KBps	0x4E20	0	2.4 KBps	0x9C40	0
3	9.6 KBps	9.6 KBps	0x1388	0	9.6 KBps	0x2710	0
4	19.2 KBps	19.2 KBps	0x9C4	0	19.2 KBps	0x1384	0
5	38.4 KBps	38.4 KBps	0x4E2	0	38.4 KBps	0x9C2	0
6	57.6 KBps	57.62 KBps	0x341	0.03	57.59 KBps	0x681	0.02
7	115.2 KBps	115.11 KBps	0x1A1	0.08	115.25 KBps	0x340	0.04
8	230.4 KBps	230.76 KBps	0xD0	0.16	230.21 KBps	0x1A0	0.08
9	460.8 KBps	461.54 KBps	0x68	0.16	461.54 KBps	0xD0	0.16
10	921.6 KBps	923.07 KBps	0x34	0.16	923.07 KBps	0x64	0.16
11	2 MBps	2 MBps	0x18	0	2 MBps	0x30	0
12	3 MBps	3 MBps	0x10	0	3 MBps	0x20	0
13	4 MBps	N/A	N/A	N/A	4 MBps	0x14	0
14	5 MBps	N/A	N/A	N/A	5052.63 KBps	0x11	1.05
15	6 MBps	N/A	N/A	N/A	6 MBps	0x10	0

1. CPU クロック周波数が低いほど、ボーレートの精度も低下しますボーレートの達成可能な上限は、これらのデータによって決定できます。



26.5.5 クロック偏差に対する USART レシーバの許容誤差

USART の非同期レシーバは、クロックシステムの合計偏差が USART レシーバの許容誤差未満の場合のみ、正しく動作します。合計偏差の要因は、次のとおりです。

- DTRA : トランスミッタの誤差に起因する偏差 (トランスミッタのローカルオシレータの偏差も含みます)
- DQUANT : レシーバのボーレート量子化に起因する誤差
- DREC : レシーバのローカルオシレータの偏差
- DTCL : 送信ラインに起因する偏差 (一般には、ローからハイへの遷移タイミングとハイからローへの遷移タイミングの間に非対称性をもたらす可能性のあるトランシーバに起因)

$$DTRA + DQUANT + DREC + DTCL + DWU < \text{USART receiver's 許容誤差}$$

ここで、

DWU は、STOP モードからのウェイクアップが使用されたときのサンプリングポイントの偏差によるエラーです。

M[1:0] = 01 の場合 :

$$DWU = \frac{t_{WUSTOP}}{11 \times T_{bit}}$$

M[1:0] = 00 の場合 :

$$DWU = \frac{t_{WUSTOP}}{10 \times T_{bit}}$$

M[1:0] = 10 の場合 :

$$DWU = \frac{t_{WUSTOP}}{9 \times T_{bit}}$$

t_{WUSTOP} は、STOP モードからのウェイクアップ時間であり、製品のデータシートで指定されています。

USART レシーバは、次の選択に応じて、表 97 および 表 98 で指定された最大許容偏差まで、データを正しく受信できます。

- USARTx_CR1 レジスタの M ビットで定義された 9、10、または 11 ビットキャラクタ長
- USARTx_CR1 レジスタの OVER8 ビットによって定義された 8 倍または 16 倍のオーバーサンプリング
- USARTx_BRR レジスタのビット BRR[3:0] が 0000 に等しいかどうか。
- データのサンプリングに 1 ビットを使用するか 3 ビットを使用するか (USARTx_CR3 レジスタの ONEBIT ビットの値に依存)。

表 97. BRR [3:0] = 0000 のときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 98. BRR[3:0] が 0000 でないときの USART レシーバの許容誤差

M ビット	OVER8 ビット = 0		OVER8 ビット = 1	
	ONEBIT = 0	ONEBIT = 1	ONEBIT = 0	ONEBIT = 1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注：表 97 および表 98 で指定されたデータは、M ビット = 00 のとき、受信フレームに正確に 10 ビット持続時間のアイドルフレームが含まれる特殊なケースで、若干異なることがあります (M ビット = 01 のときには 11 ビット持続時間、または M ビット = 10 のときには 9 ビット持続時間)。

26.5.6 自動ボーレート検出

USART は、1 キャラクタの受信に基づいて、USARTx_BRR レジスタ値を検出し、自動的にセットすることができます。自動ボーレート検出は、2 つの状況で便利です。

- システムの通信速度が事前に分かっていないとき。
- システムが比較的低い精度のクロックソースを使用している場合、このメカニズムによって、クロック偏差を測定しなくても、正しいボーレートを求めることができます。

クロックソース周波数は、期待される通信速度と互換性がなければなりません (モード 0 および 1 のみがサポートされている場合、16 倍のオーバーサンプリングを選択する必要があり、ボーレートは $f_{CK}/65535$ と $f_{CK}/16$ までの範囲内です。また、4 つのモードがサポートされている場合、16 倍のオーバーサンプリングのとき、ボーレートは $f_{CK}/65535$ から $f_{CK}/16$ までの範囲内です。8 倍のオーバーサンプリングのとき、ボーレートは $f_{CK}/65535$ から $f_{CK}/8$ までの範囲内です)。

自動ボーレート検出を有効にする前に、自動ボーレート検出モードを選択する必要があります。キャラクタパターンに基づいて、いくつかのモードがあります。

これらのモードは、USARTx_CR2 レジスタの ABRMOD[1:0] フィールドで選択できます。これらの自動ボーレートモードでは、同期データ受信中にボーレートが数回測定され、各測定値が前回の測定値と比較されます。

以下のモードがあります。

- **モード 0** : 1 ビットで始まるキャラクタ。この場合、USART はスタートビットの時間を測定します (立ち下がりエッジから立ち上がりエッジまで)。
- **モード 1** : 10xx ビットパターンで始まるキャラクタ。この場合、USART はスタートと最初のデータビットの時間を測定します。低速な信号傾斜の場合の精度を高めるために、測定は立ち下がりエッジから立ち下がりエッジまでで行われます。

- **モード 2:** 0x7F キャラクタフレーム (LSB ファーストモードでは 0x7F キャラクタ、または MSB ファーストモードでは 0xFE)。この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 6 の終了時に更新されます (立ち下がリエッジから立ち下がリエッジまで行われた測定に基づいて、BR6)。ビット 0 からビット 6 までは BRs でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。
 - **モード 3:** 0x55 キャラクタフレーム。この場合、ボーレートは、最初はスタートビット (BRs) の終了時に更新され、次にビット 0 の終了時に更新され (立ち下がリエッジから立ち下がリエッジまで行われた測定に基づいて BR0)、最後にビット 6 (BR6) の終了時に更新されます。ビット 0 は BRs でサンプリングされ、ビット 1 からビット 6 までは BR0 でサンプリングされ、キャラクタの残りのビットは BR6 でサンプリングされます。
- 並行して、RX ラインの中間遷移ごとに別のチェックが行われます。RX の遷移がレシーバと十分に同期していない場合はエラーが生成されます (レシーバは、ビット 0 で計算されたボーレートに基づきます)。

自動ボーレート検出を有効にする前に、ゼロ以外のボーレート値を書き込むことによって、USARTx_BRR レジスタを初期化する必要があります。

自動ボーレート検出を有効にするには、USARTx_CR2 レジスタの ABREN ビットをセットします。USART は、RX ラインの最初のキャラクタを待ちます。自動ボーレート動作の完了は、USARTx_ISR レジスタの ABRF フラグのセットによって示されます。ラインにノイズが多い場合、正しいボーレート検出を保証できません。この場合、BRR 値が破損して、ABRE エラーフラグがセットされることがあります。また、通信速度が自動ボーレート検出の範囲と互換性がない場合にも発生します (ビット時間が 16 から 65536 までのクロック周期でなく (16 倍のオーバーサンプリング)、8 から 65536 までのクロック周期でない (8 倍のオーバーサンプリング))。

RXNE 割り込みは、操作の終了を知らせます。

その後いつでも、ABRF フラグをリセットすることによって (0 を書き込むことによって)、自動ボーレート検出を再開できます。

注: *自動ボーレート操作中に USART が無効化された場合 (UE=0)、BRR 値が破損することがあります。*

26.5.7 マルチプロセッサ通信

マルチプロセッサ通信では、次のビットをクリアされた状態に保つ必要があります。

- USART_CR2 レジスタの LINEN ビット
- USART_CR3 レジスタの HDSEL、IREN、および SCEN ビット

USART とのマルチプロセッサ通信が可能です (ネットワーク内で複数の USART を接続して)。たとえば、1 つの USART をマスタとして、その TX 出力を別の USART の RX 入力に接続することができます。別の USART はスレーブであり、それぞれの TX 出力の論理積をとった上でマスタの RX 入力に接続します。

マルチプロセッサ設定では、多くの場合、メッセージの本来の受信者のみがメッセージ内容の全体を能動的に受信することが望ましく、これによって対象外の受信者に対する USART サービスの余分なオーバーヘッドを減らすことができます。

対象外のデバイスは、ミュート機能によってミュートモードにできます。ミュートモード機能を使用するためには、USARTx_CR1 レジスタの MME ビットをセットする必要があります。

ミュートモードでは、

- 受信ステータスビットはセットできません。
- 受信割り込みはすべて禁止されます。
- USARTx_ISR レジスタの RWU ビットは 1 にセットされます。特定の条件下では、USARTx_RQR レジスタの MMRQ ビットを通じて、RWU をハードウェアまたはソフトウェアによって自動的に制御できます。

USART は、USARTx_CR1 レジスタの WAKE ビットの設定に応じて、次のいずれかの方法でミュートモードに入ったり終了したりできます。

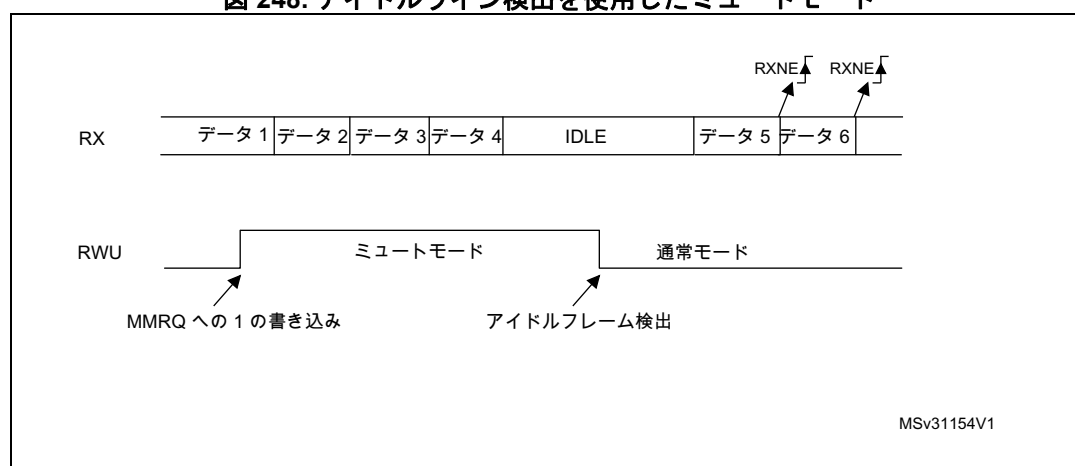
- WAKE ビットがリセットされている場合は、アイドルライン検出
- WAKE ビットがセットされている場合は、アドレスマーク検出

アイドルライン検出 (WAKE=0)

MMRQ ビットに 1 が書き込まれ、RWU が自動的にセットされたときには、USART はミュートモードに入ります。

USART は、アイドルフレームを検出するとウェイクアップします。その後、RWU ビットはハードウェアによってクリアされますが、USARTx_ISR レジスタの IDLE ビットはセットされません。アイドルライン検出を使用したミュートモードの動作例を [図 248](#) に示します。

図 248. アイドルライン検出を使用したミュートモード



注： IDLE キャラクタがすでに経過しているときに MMRQ がセットされた場合は、ミュートモードに入りません (RWU はセットされません)。

ラインが IDLE のときに USART が有効にされた場合、1 IDLE フレーム後にアイドル状態が検出されます (1 キャラクタフレームの受信後だけでなく)。

4 ビット／7 ビットアドレスマーク検出 (WAKE=1)

このモードでは、MSB が 1 の場合、バイトはアドレスとして認識され、そうでない場合はデータとみなされます。アドレスバイトのうち、ターゲットレシーバのアドレスは 4 または 7 LSB です。7 または 4 ビットアドレス検出の選択は、ADD7M7 ビットを使用して行われます。この 4 ビット／7 ビットワードは、レシーバによって、USARTx_CR2 レジスタの ADD ビットでプログラムされたレシーバの自己アドレスと比較されます。

注： 7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) で行われます。

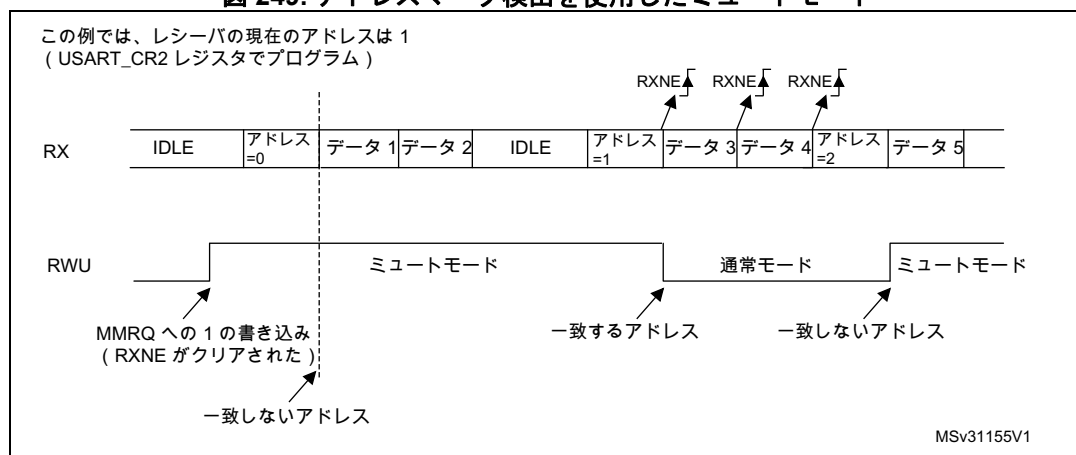
プログラミングされたアドレスと一致しないアドレスキャラクタを受信すると、USART はミュートモードに入ります。この場合、RWU ビットはハードウェアによってセットされます。USART がミュートモードに入ったときには、このアドレスバイトに対して RXNE フラグはセットされず、割り込みも DMA リクエストも発行されません。

MMRQ ビットに 1 が書き込まれたときにも、USART はミュートモードに入ります。この場合、RWU ビットも自動的にセットされます。

プログラムされたアドレスに一致するアドレスキャラクタを受信すると、USART はミュートモードを終了します。続いて RWU ビットがクリアされ、それ以降のバイトは通常どおりに受信されます。RWU ビットはクリアされているので、アドレスキャラクタに対して RXNE ビットがセットされます。

アドレスマーク検出を使用したミュートモードの動作例を [図 249](#) に示します。

図 249. アドレスマーク検出を使用したミュートモード



26.5.8 Modbus 通信

USART は、Modbus/RTU および Modbus/ASCII プロトコルの実装に対する基本的サポートを備えています。Modbus/RTU は、半二重のブロック転送プロトコルです。プロトコルの制御部分（アドレス認識、ブロック整合性制御、およびコマンド解釈）は、ソフトウェアで実装する必要があります。

USART は、ソフトウェアに負荷をかけず、他のリソースを使用せずに、ブロック検出の終了に対する基本的サポートを提供します。

Modbus/RTU

このモードでは、1 つのブロックの終了は 2 キャラクタ時間を超える「サイレンス」（アイドルライン）によって認識されます。この機能は、プログラム可能なタイムアウト機能を通じて実装されます。

タイムアウト機能と割り込みは、USARTx_CR2 レジスタの RTOEN ビットと、USARTx_CR1 レジスタの RTOIE を通じて有効にする必要があります。2 キャラクタ時間のタイムアウトに対応する値（たとえば、22 x ビット時間）を RTO レジスタでプログラムする必要があります。最後のストップビットの受信後、この時間にわたって受信ラインがアイドルのときには、割り込みが生成されて、現在のブロック受信が完了したことをソフトウェアに知らせます。

Modbus/ASCII

このモードでは、ブロックの終了は特定の（CR/LF）キャラクタシーケンスによって認識されます。USART は、キャラクター一致機能を使用して、このメカニズムを管理します。

ADD[7:0] フィールドで LF ASCII コードをプログラムし、キャラクター一致割り込みを有効にすることによって (CMIE=1)、LF が受信されたときにソフトウェアに通知し、ソフトウェアは DMA バッファの CR/LF をチェックできます。

26.5.9 パリティ制御

パリティ制御 (送信中のパリティビット生成と受信中のパリティ検査) を有効にするには、USARTx_CR1 レジスタの PCE ビットをセットします。M ビットによって定義されたフレーム長に応じて、可能な USART フレームフォーマットを表 99 に示します。

表 99. フレームフォーマット

M ビット	PCE ビット	USART フレーム ⁽¹⁾
00	0	SB 8 ビットデータ STB
00	1	SB 7 ビットデータ PB STB
01	0	SB 9 ビットデータ STB
01	1	SB 8 ビットデータ PB STB
10	0	SB 7 ビットデータ STB
10	1	SB 6 ビットデータ PB STB

1. 凡例: SB: スタートビット、STB: ストップビット、PB: パリティビット。データレジスタでは、PB は常に MSB 位置を取ります (M ビットの値に応じて、9、8、または 7 番目)。

偶数パリティ

パリティビットは、6、7、または 8 LSB ビット (M ビットの 値に応じて) とパリティビットから構成されるフレーム内で「1」の数が偶数になるように計算されます。

たとえば、データ = 00110101 であり、4 ビットがセットされた場合、偶数パリティが選択された場合 (USARTx_CR1 の PS ビット = 0)、パリティビットは 0 になります。

奇数パリティ

パリティビットは、6、7、または 8 LSB ビット (M ビットの 値に応じて) とパリティビットから構成されるフレーム内で「1」の数が奇数になるように計算されます。

たとえば、データ = 00110101 であり、4 ビットがセットされた場合、奇数パリティが選択された場合 (USARTx_CR1 の PS ビット = 1)、パリティビットは 1 になります。

受信中のパリティチェック

パリティチェックに失敗した場合、USARTx_ISR レジスタの PE フラグがセットされ、USARTx_CR1 レジスタの PEIE ビットがセットされている場合は割り込みが生成されます。PE フラグは、USARTx_ICR レジスタの PECF に 1 を書き込むことによってクリアされます。

送信中のパリティ生成

USARTx_CR1 の PCE ビットがセットされている場合、データレジスタに書き込まれたデータの MSB ビットは送信されますが、パリティビットによって変更されます (偶数パリティが選択された場合 (PS=0) は偶数個の「1」、奇数パリティが選択された場合 (PS=1) は奇数個の「1」)。



26.5.10 LIN (Local Interconnection Network) モード

このセクションは、LIN モードがサポートされるときにのみ適用されます。[セクション 26.4: USART の実装 \(681 ページ\)](#) を参照してください。

LIN モードを選択するには、USARTx_CR2 レジスタの LINEN ビットをセットします。LIN モードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx_CR2 レジスタの CLKEN
- USARTx_CR3 レジスタの STOP[1:0]、SCEN、HDSEL、および IREN

コード例については、「付録」セクション [A.19.6: USART LIN モードのコード例](#) を参照してください。

LIN 送信

LIN マスタ送信の場合、[セクション 26.5.2: トランスミッタ](#) に説明されている手順を適用する必要があります。通常の USART 送信と同じですが、次のような違いがあります。

- 8 ビットのワード長を設定するには M ビットをクリアします。
- LIN モードに入るには、LINEN ビットをセットします。この場合、SBKRQ ビットをセットすると、13 個の “0” ビットがブレークキャラクタとして送信されます。その後、値 “1” の 2 ビットが送信され、次の START 検出が可能になります。

LIN 受信

LIN モードが有効になると、ブレーク検出回路が有効になります。この検出は、通常の USART レシーバとは完全に独立しています。アイドル状態時やフレームの処理中には、発生のたびにブレークが検出できます。

レシーバが有効になると (USARTx_CR1 レジスタの RE=1)、RX 入力の START 信号を探します。スタートビットの検出方法は、ブレークキャラクタやデータの検索方法と同じです。スタートビットが検出された後、データの場合とまったく同様に次のビットがサンプリングされます (8、9、および 10 番目のサンプル)。10 個 (USARTx_CR2 レジスタの LBDL=0) または 11 個 (USARTx_CR2 レジスタの LBDL=1) の連続したビットが “0” として検出され、その後にデリミタキャラクタが続く場合、USARTx_ISR レジスタの LBDF フラグがセットされます。LBDIE ビットが 1 の場合、割り込みが生成されます。ブレークを確認する前に、RX ラインがハイレベルに戻ったことを知らせるデリミタが検査されます。

この 10 または 11 が発生する前に “1” がサンプリングされた場合、ブレーク検出回路は現在の検出をキャンセルし、再びスタートビットを検索します。

LIN モードが無効にされた場合 (LINEN=0)、レシーバは、ブレーク検出を考慮することなく、通常の USART として機能し続けます。

LIN モードが有効にされた場合 (LINEN=1)、フレーミングエラーが発生 (つまり、ブレークフレームの場合と同様に、“0” の位置でストップビットを検出) すると、レシーバは停止し、ブレーク検出回路が “1” (ブレークワードが完全でなかった場合) またはデリミタキャラクタ (ブレークが検出された場合) を受信するまで停止状態を維持します。

ブレーク検出回路ステートマシンの動作とブレークフラグを [図 250: LIN モードでのブレーク検出 \(11 ビットブレーク長、LBDL=1\) \(704 ページ\)](#) に示します。

ブレークフレームの例を [図 251: LIN モードでのブレーク検出とフレーミングエラー検出 \(705 ページ\)](#) に示します。

図 250. LIN モードでのブレーク検出（11 ビットブレーク長、LBDF=1）

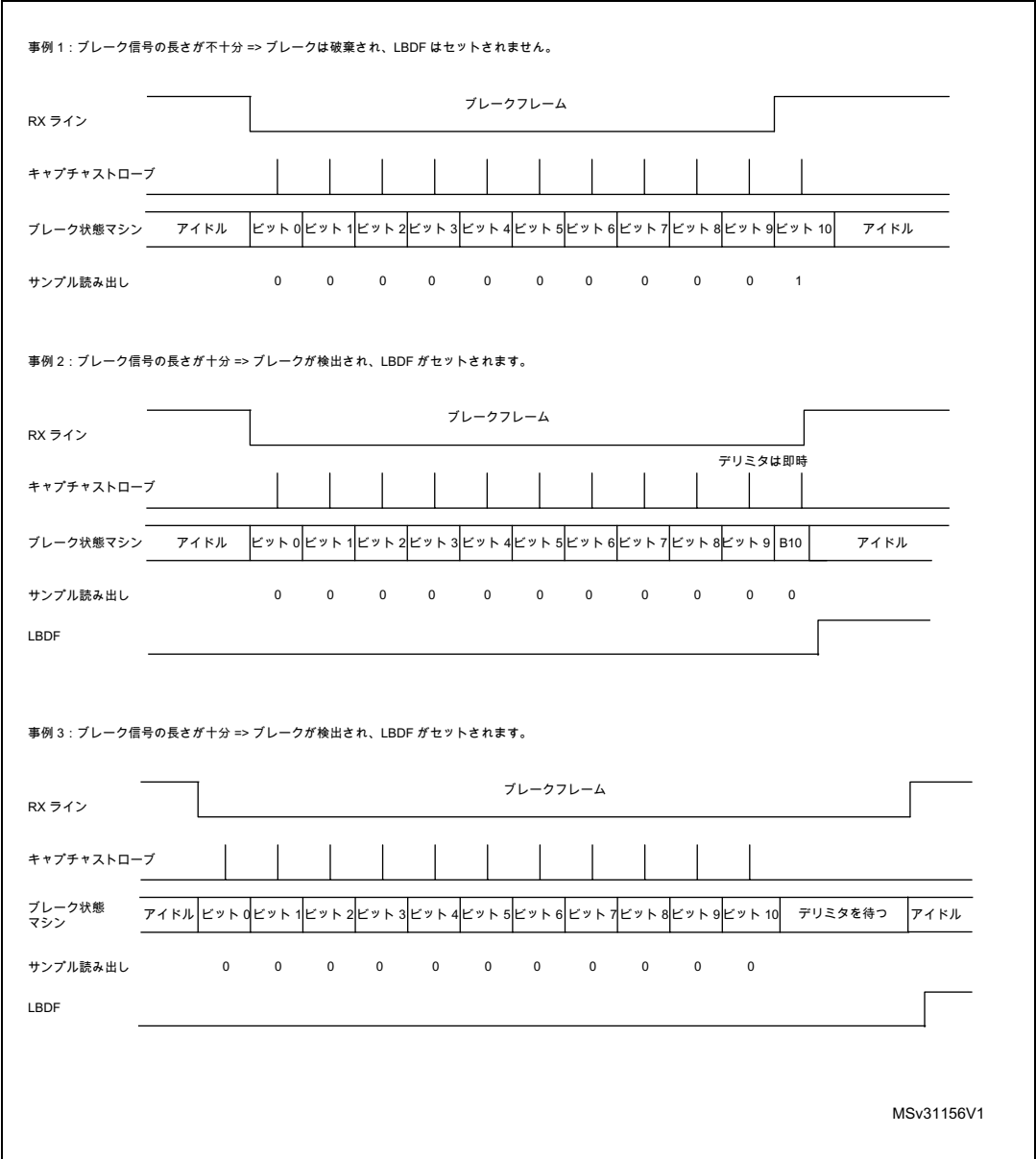
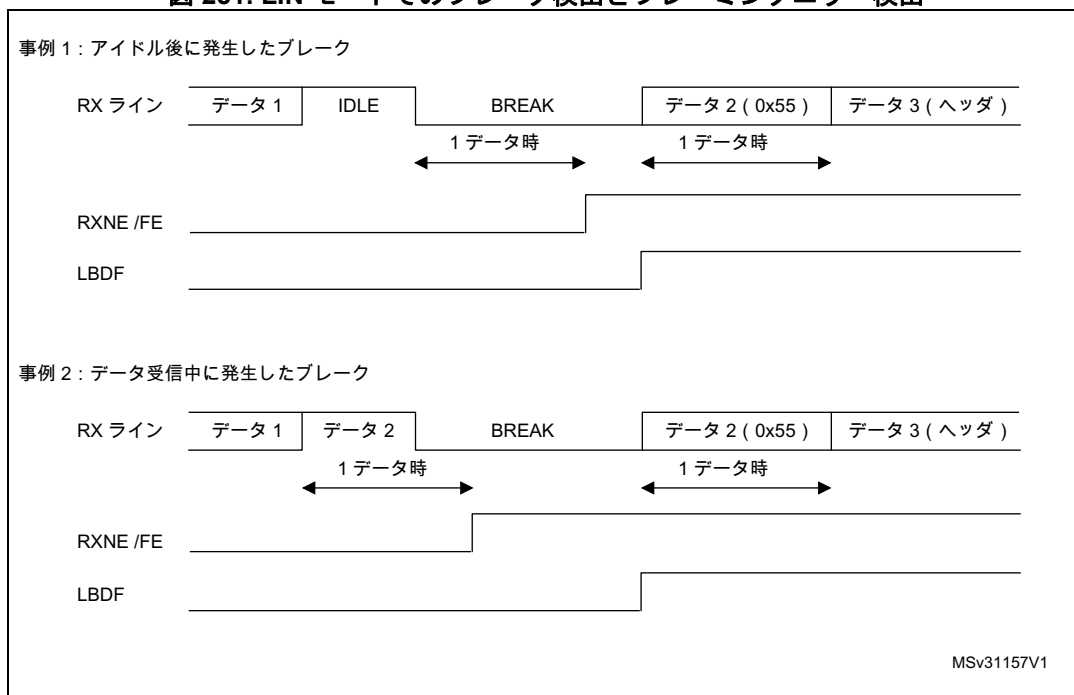


図 251. LIN モードでのブレーク検出とフレーミングエラー検出



26.5.11 USART 同期モード

同期モードを選択するには、USARTx_CR2 レジスタの CLKEN ビットに 1 を書き込みます。同期モードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx_CR2 レジスタの LINEN ビット
- USARTx_CR3 レジスタの SCEN、HDSEL、および IREN ビット

このモードでは、USART を使用して、双方向同期シリアル通信をマスタモードで制御できます。SCLK ピンは USART トランスミッタクロックの出力です。スタートビットとストップビットの処理中には、SCLK ピンにクロックパルスは送信されません。USARTx_CR2 レジスタの LBCL ビットの状態によっては、有効な最後のデータビット（アドレスマーク）の処理中にクロックパルスが生成されることもあります。USARTx_CR2 レジスタの CPOL ビットは、クロック極性を選択するために使用され、USARTx_CR2 レジスタの CPHA ビットは、外部クロックの位相を選択するために使用されます（図 252、図 253、および図 254 を参照）。

アイドル時、プリアンブル処理時、およびブレーク送信時には、外部 SCLK クロックは起動されません。

同期モードでは、USART トランスミッタは非同期モードの場合とまったく同じように機能します。しかし、CPOL と CPHA に基づいて SCLK が TX と同期すると、TX 上のデータが同期します。

このモードでは、USART レシーバは非同期モードの場合とは異なる動作をします。RE=1 の場合、データはオーバーサンプリングなしで、SCLK (CPOL と CPHA に応じて立上りまたは立下りエッジ) でサンプリングされます。ボーレート (1/16 ビット持続時間) に応じたセットアップ時間とホールド時間を守る必要があります。

注： SCLK ピンはTX ピンと連携して動作します。したがって、クロックが供給されるのは、トランスミッタが有効であり (TE=1)、データが送信中 (データレジスタ USARTx_TDR への書き込み) の場合に限られます。つまり、データ送信を行わずに同期データを受信することはできません。

USART が無効にされたときには (UE=0)、クロックパルスが正常に機能するように、LBCL、CPOL、および CPHA ビットを選択する必要があります。

注： コード例については、「付録」セクション A.19.7: USART 同期モードのコード例を参照してください。

図 252. USART の同期送信の例

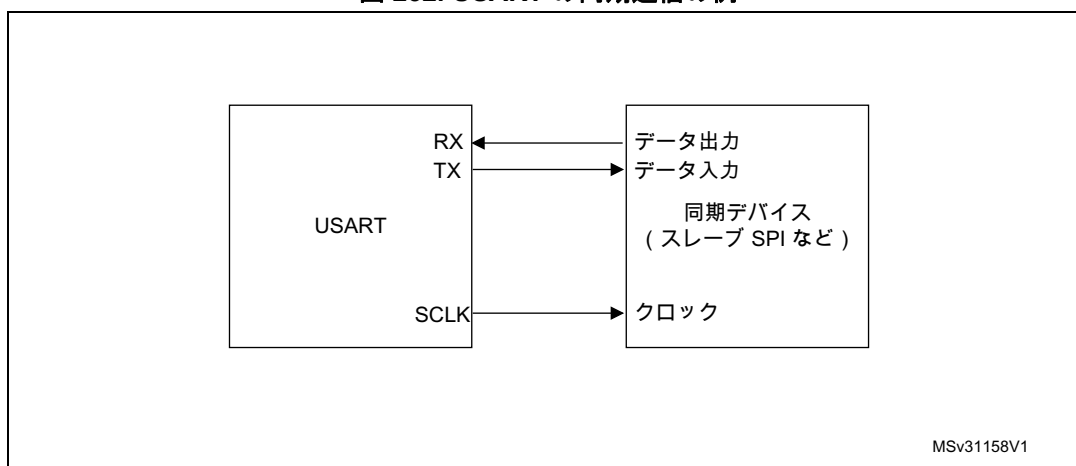


図 253. USART データクロックタイミング図 (M ビット = 00)

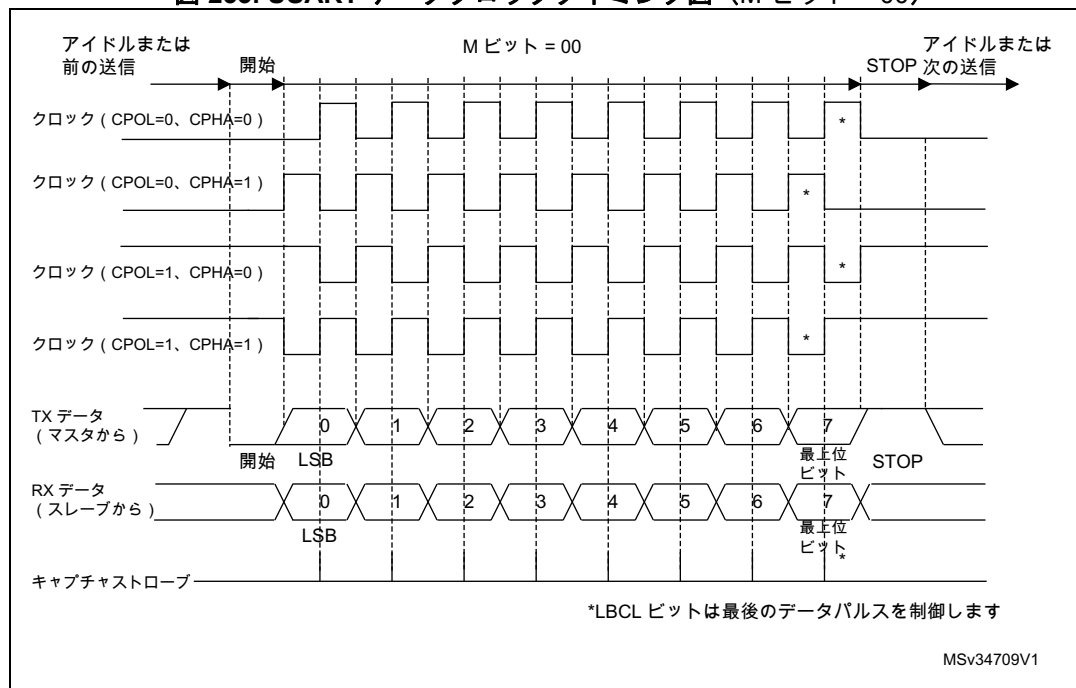


図 254. USART データクロックタイミング図 (M ビット = 01)

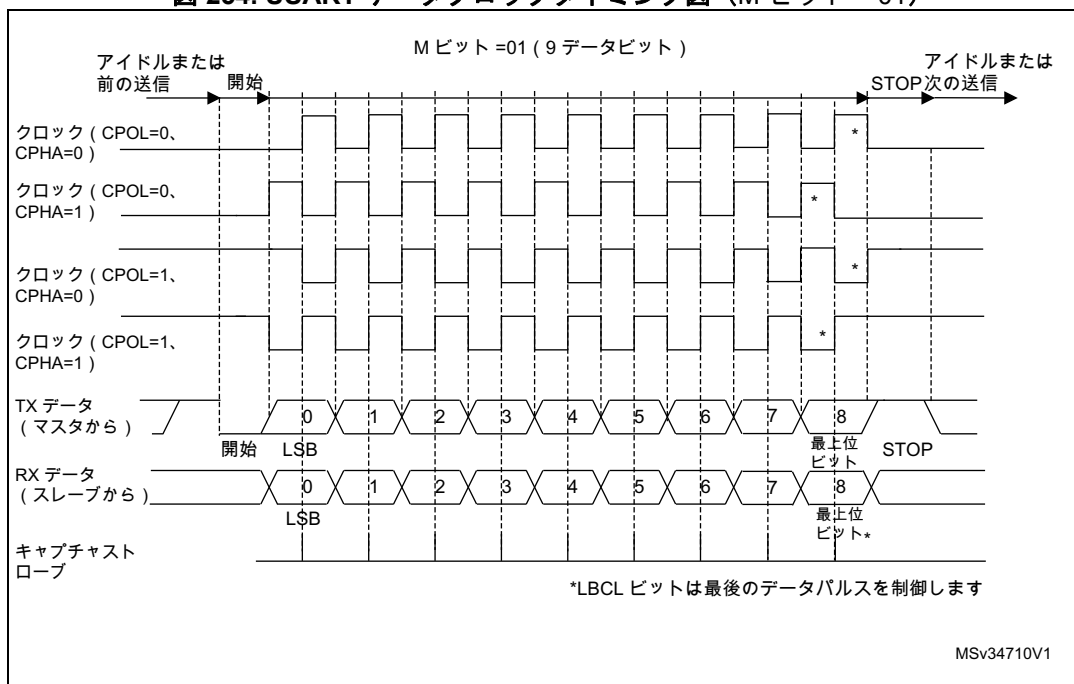
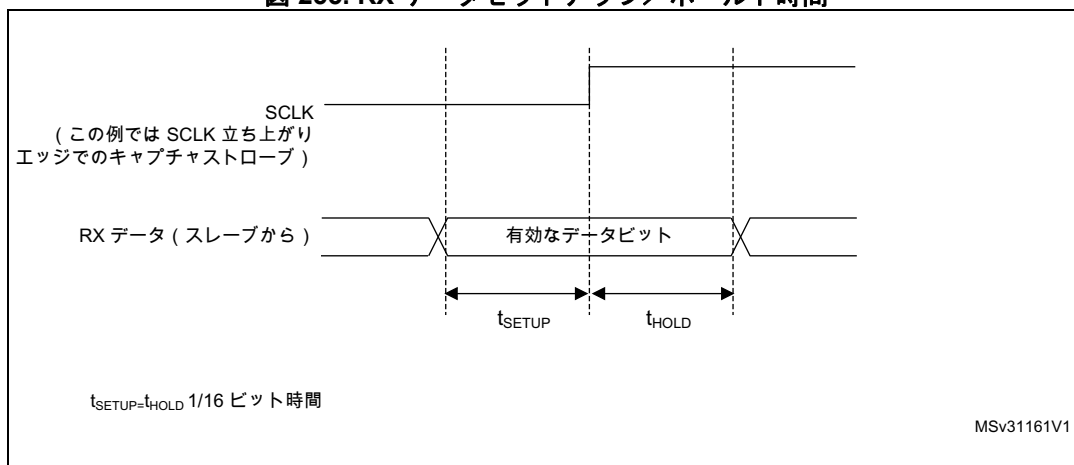


図 255. RX データセットアップ/ホールド時間



注：スマートカードモードでは SCLK の機能が異なります。詳細については、[セクション 26.5.13：スマートカードモード](#)を参照してください。

26.5.12 単線半二重通信

単線半二重モードを選択するには、USARTx_CR3 レジスタの HDSEL ビットをセットします。このモードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx_CR2 レジスタの LINEN および CLKEN ビット
- USARTx_CR3 レジスタの SCEN および IREN ビット

USART は、単線半二重のプロトコルに従うように設定できます。この場合、TX ラインと RX ラインは内部接続されます。半二重通信と全二重通信の選択は、USART_CR3 レジスタの制御ビット HDSELで行います。

HDSEL ビットに 1 が書き込まれると、

- TX ラインと RX ラインが内部接続されます。
- RX ピンは使用されなくなります。
- データが送信されないときには、TX ピンは常に解放されます。したがって、アイドル時や受信時には標準入出力として機能します。つまり、TX が外部プルアップ付きの代替機能オープンドレインとして設定されるように、I/O を設定する必要があります。

この点を除くと、通信プロトコルは通常の USART モードと同じです。ラインの競合はソフトウェアによって管理する必要があります（たとえば、集中型アービタを使用して）。特に、送信がハードウェアによってブロックされることはなく、TE ビットがセットされている間は、データレジスタにデータが書き込まれるとすぐに、送信が続行されます。

コード例については、「付録」セクション [A.19.8: USART 単線半二重のコード例](#) を参照してください。

26.5.13 スマートカードモード

このセクションは、スマートカードモードがサポートされるときにのみ適用されます。 [セクション 26.4: USART の実装 \(681 ページ\)](#) を参照してください。

スマートカードモードを選択するには、USARTx_CR3 レジスタの SCEN ビットをセットします。スマートカードモードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx_CR2 レジスタの LINEN ビット
- USARTx_CR3 レジスタの HDSEL および IREN ビット

さらに、スマートカードにクロックを供給するために CLKEN ビットをセットすることもできます。

スマートカードインタフェースは、ISO 7816-3 標準で定義されたスマートカードの非同期プロトコルをサポートするように設計されています。T=0（キャラクタモード）と T=1（ブロックモード）の両方がサポートされます。

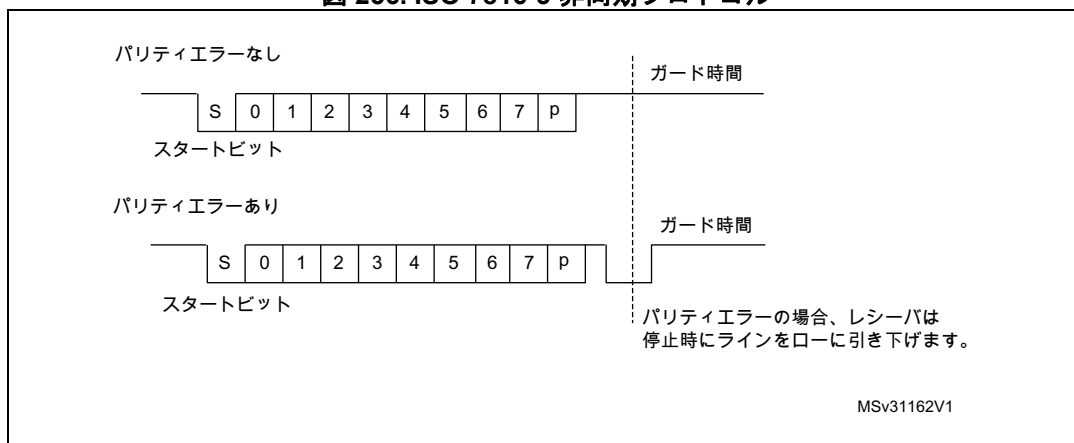
USART は次のように設定してください。

- 8 ビット+パリティ：ワード長が 8 ビットにセットされ、および USARTx_CR1 レジスタの PCE=1。
- 1.5 個のストップビット：USARTx_CR2 レジスタの STOP=11。受信にはストップビット 0.5 個を選択することも可能です。

T=0（キャラクタ）モードでは、パリティエラーはガードタイム中の各キャラクタの終わりに示されます。

[図 256](#) に、パリティエラーの有無によるデータラインの状況の変化の例を示します。

図 256. ISO 7816-3 非同期プロトコル



スマートカードと接続されると、USART の TX 出力は、スマートカードによっても駆動される双方向ラインを駆動します。TX ピンは、オープンドレインとして設定される必要があります。

スマートカードモードは、単線半二重通信プロトコルを実装します。

- 送信シフトレジスタからのデータの送信は、少なくとも 1/2 ボークロックの遅れが保証されます。通常動作では、満杯の送信シフトレジスタは、次のボークロックエッジでシフト動作を開始します。スマートカードモードでは、この送信は、保証された 1/2 ボークロック分だけさらに遅れます。
- 送信時、スマートカードがパリティエラーを検出した場合には、ラインをローに駆動することによって (NACK)、この状態を USART に知らせます。この NACK 信号 (1 ボークロックの間、送信ラインをローに引き下げ) は、1.5 個のストップビットが組み込まれたトランスミッタ側にフレーミングエラーを引き起こします。USART は、プロトコルに従って、データの自動再送信を処理できます。再試行回数は、SCARCNT ビットフィールドでプログラムされます。プログラムされた再試行回数後も USART が NACK を受信し続けた場合は、送信を停止して、エラーをフレーミングエラーとして通知します。TXE ビットは、USARTx_RQR レジスタの TXFRQ ビットを使用してセットできます。
- 送信時のスマートカード自動再試行: USART による NACK の検出と反復キャラクタのスタートビットの間に 2.5 ボー周期の遅延が挿入されます。最後の反復キャラクタの受信終了時、ただちに TC ビットがセットされます (ガードタイムはありません)。ソフトウェアで再び繰り返したい場合は、規格によって指定されている 2 ボー周期以上を確保する必要があります。
- 1.5 個のストップビット周期でプログラムされたフレームの受信時にパリティエラーが検出された場合、受信フレームの完了後 1 ボークロック周期間、送信ラインがローに引き下げられます。これは、USART に送信されたデータが正しく受信されなかったことをスマートカードに知らせるためです。NACK 制御ビットがセットされている場合、パリティエラーはレシーバによって "NACK" され、そうでない場合、NACK は送信されません (T=1 モードで使用されます)。受信したキャラクタにエラーがあった場合、RXNE/受信 DMA リクエストは有効になりません。プロトコルの仕様に従って、スマートカードは同じキャラクタを再送信する必要があります。SCARCNT ビットフィールドで指定された最大試行回数後も、受信したキャラクタにエラーがあった場合、USART は NACK の送信を停止して、エラーをパリティエラーとして通知します。
- 受信時のスマートカード自動再試行: USART がカードを NACK したが、カードがキャラクタを繰り返さなかった場合、BUSY フラグはセットされたままです。
- 送信時、USART は 2 つの連続するキャラクタの間にガードタイム (ガードタイムレジスタでプログラム) を挿入します。ガードタイムは前のキャラクタのストップビット後に測定されるので、GT[7:0] レジスタを目的の CGT (7816-3 仕様で定義されている Character Guard Time) から 12 (1 キャラクタの時間) を引いた値にプログラムする必要があります。

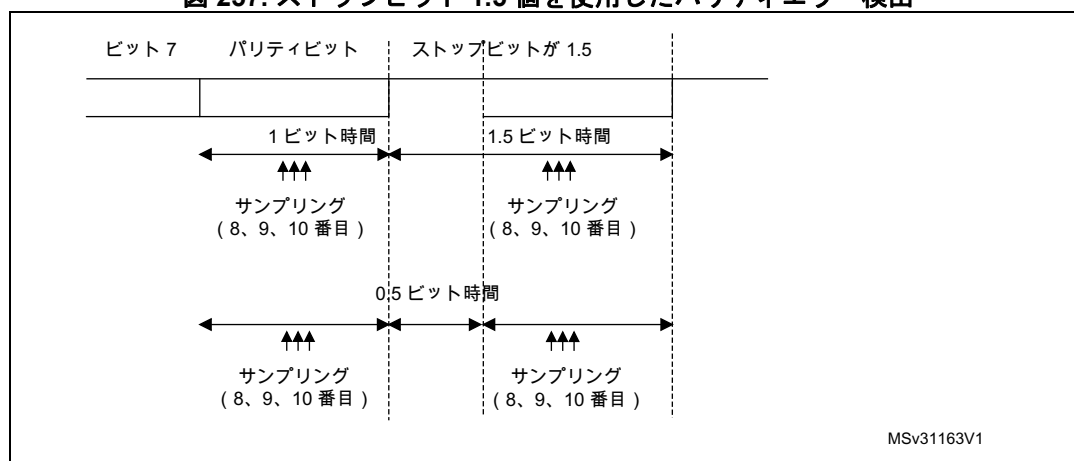
- TC フラグのアサーションは、ガードタイムレジスタをプログラムすることによって遅らせることができます。通常動作では、TC がアサートされるのは、送信シフトレジスタが空であり、他に未処理の送信リクエストがない場合です。スマートカードモードでは、空の送信シフトレジスタは、ガードタイムカウンタをトリガして、ガードタイムレジスタにプログラミングされた値までカウントアップします。この間、TC は強制的にローレベルに保たれます。ガードタイムカウンタがプログラミングされた値に達すると、TC がハイにアサートされます。
- TC フラグのネゲートは、スマートカードモードの影響を受けません。
- レシーバからの NACK によってトランスミッタ端でフレーミングエラーが検出された場合、トランスミッタの受信ブロックは、この NACK をスタートビットとして検出しません。ISO プロトコルによれば、受信される NACK の期間は 1 または 2 ボークロック周期です。
- レシーバ側では、パリティエラーが検出されて NACK が送信された場合、レシーバはこの NACK をスタートビットとして検出しません。

注： スマートカードモードでは、ブレイクキャラクタは意味を持ちません。フレーミングエラー発生時のデータ 0x00 は、ブレイクではなくデータとして処理されます。

TE ビットをトグルするとき、アイドルフレームは送信されません。アイドルフレームは、他の設定では定義されますが、ISO プロトコルでは定義されていません。

図 257 に、USART による NACK 信号のサンプリング方法を示します。この例では、USART はデータを送信中であり、ストップビットが 1.5 個組み込まれています。データと NACK 信号の整合性を検査するために、USART のレシーバ部が有効にされます。

図 257. ストップビット 1.5 個を使用したパリティエラー検出



USART は、SCLK 出力を通じてスマートカードにクロックを供給できます。スマートカードモードでは、SCLK は通信に関係せず、5 ビットのプリスケアラを通じて単に内部のペリフェラル入力クロックから取得されます。この分周比は、プリスケアラレジスタ USARTx_ で設定されます。SCLK 周波数は、 $f_{CK}/2$ から $f_{CK}/62$ までの範囲でプログラムでき、 f_{CK} はペリフェラル入力クロックです。

ブロックモード (T=1)

T=1 (ブロック) モードでは、パリティエラー送信は、USART_CR3 レジスタの NACK ビットをクリアすることによって無効化されます。

ブロックモードでスマートカードからの読み出しをリクエストするときには、ソフトウェアは USART_CR2 レジスタの RTOEN ビットをセットすることによってレシーバタイムアウト機能を有効にし、RTOR レジスタの RTO ビットフィールドを BWT (ブロックウェイトタイム) - 11 の値にプログラムする必要があります。この期間が終了する前にカードからの応答が受信されなかった場合、RTOF フラグがセットされ、タイムアウト割り込みが生成されます (USART_CR1 レジスタの RTOIE ビットがセットされていた場合)。この期間が終了する前に最初のキャラクタが受信された場合は、RXNE 割り込みによって示されます。

注： *ブロックモードのスマートカードからの読み出しに USART を DMA モードで使用するときでも、RXNE 割り込みを有効にする必要があります。並行して、DMA は最初の受信バイトの後でのみ有効にする必要があります。*

2 つの連続するキャラクタの間で最大ウェイトタイムの自動チェックを行うには、最初のキャラクタの受信後 (RXNE 割り込み)、RTOR レジスタの RTO ビットフィールドを CWT (キャラクタウェイトタイム) - 11 の値にプログラムする必要があります。この時間は、ボータイム単位で表されます。スマートカードが前のキャラクタの終了後、CWT 未満の時間内に新しいキャラクタを送信しなかった場合、USART は RTOF フラグと割り込み (RTOIE ビットがセットされているとき) によって、これをソフトウェアに通知します。

注： *RTO カウンタはカウントを開始します。*

- STOP = 00 の場合、ストップビットの最後から

- STOP = 10 の場合、2 番目のストップビットの最後から

- STOP = 11 の場合、ストップビットの最初から 1 ビット時間後

スマートカードプロトコルの定義にあるように、BWT/CWT 値は最後のキャラクタの開始 (スタートビット) から定義されます。RTO レジスタは、最後のキャラクタ自体の長さを考慮して、それぞれ BWT - 11 または CWT - 11 にプログラムする必要があります。

ブロック長カウンタは、USART が受信するすべてのキャラクタをカウントするために使用されます。このカウンタは、USART の送信時 (TXE=0) にリセットされます。ブロックの長さは、スマートカードによってブロックの 3 番目のバイト (プロローグフィールド) で伝えられます。この値を USARTx_RTOR レジスタの BLEN フィールドでプログラムする必要があります。DMA モードを使用するときには、ブロックの開始前に、このレジスタフィールドを最小値 (0x0) にプログラムする必要があります。この値では、4 番目の受信キャラクタの後に割り込みが生成されます。ソフトウェアは LEN フィールド (3 番目のバイト) を読み出す必要があり、その値は受信バッファから読み出される必要があります。

割り込み駆動受信モードでは、ブロックの長さはソフトウェアによって、または BLEN 値をプログラムすることによってチェックできます。ただし、ブロックの開始前に、BLEN の最大値 (0xFF) をプログラムすることができます。実際の値は、3 番目のキャラクタの受信後にプログラムされます。

ブロックが LRC 水平冗長検査 (1 エピローグバイト) を使用している場合は、BLEN=LEN です。ブロックが CRC メカニズム (2 エピローグバイト) を使用している場合は、BLEN=LEN+1 をプログラムする必要があります。合計ブロック長 (プロローグ、エピローグ、および情報フィールドを含む) は、BLEN+4 に等しくなります。ブロックの終わりは EOBFF フラグと割り込み (EOBIE ビットがセットされているとき) によってソフトウェアに通知されます。

ブロック長エラーの場合、ブロックの終わりは RTO 割り込みによって通知されます (キャラクタウェイトタイムオーバーフロー)。

注： *エラーチェックコード (LRC/CRC) は、ソフトウェアによって計算/確認されなければなりません。*



ダイレクトおよびインバースコンベンション

スマートカードプロトコルは、ダイレクトとインバースの 2 つコンベンションを定義しています。

ダイレクトコンベンションは、LSB ファースト、論理ビットの値 1 がラインの H 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=0、DATAINV=0（デフォルト値）をプログラムする必要があります。

インバースコンベンションは、MSB ファースト、論理ビットの値 1 が単線の L 状態に対応、および偶数パリティとして定義されています。このコンベンションを使用するためには、制御ビット MSBFIRST=1、DATAINV=1 をプログラムする必要があります。

注： **論理データ値が反転されると (0=H、1=L)、パリティビットも同じように反転されます。**

カードのコンベンションを認識するために、カードは最初のキャラクタ TS を ATR (Answer To Reset) フレームの最初のキャラクタとして送信します。TS には、LHHL LLL LLH と LHHL HHH LLH の 2 つのパターンが使えます。

- (H) LHHL LLL LLH は、インバースコンベンションをセットアップします。状態 L が値 1 にエンコードされ、モーメント 2 は最上位ビットを含みます (MSB ファースト)。インバースコンベンションによってデコードされると、送受信されたバイトは 3F に等しくなります。
- (H) LHHL HHH LLH は、ダイレクトコンベンションをセットアップします。状態 H が値 1 にエンコードされ、モーメント 2 は最下位ビットを含みます (LSB ファースト)。ダイレクトコンベンションによってデコードされると、送受信されたバイトは 3B に等しくなります。

キャラクタパリティは、2 から 10 までの 9 個のモーメントに 1 にセットされた偶数個のビットがあったときに正しいとみなされます。

USART はカードが使用するコンベンションを知らないで、いずれのパターンであるかを認識して、それに応じて動作できる必要があります。パターン認識はハードウェアでは行われず、ソフトウェアシーケンスによって行われます。さらに、USART がダイレクトコンベンション（デフォルト）で設定され、カードがインバースコンベンションで応答した場合、TS = LHHL LLL LLH になり、USART が受信したキャラクタは 03 になり、パリティは奇数です。

したがって、TS パターン認識には 2 つの方式を使用できます。

方式 1

USART は、標準スマートカードモード／ダイレクトコンベンションでプログラムされます。この場合、TS パターンの受信によってパリティエラー割り込みと、カードに対するエラー信号が生成されます。

- パリティエラー割り込みは、カードがダイレクトコンベンションで正しく応答しなかったことをソフトウェアに知らせます。ソフトウェアは、USART をインバースコンベンションで再プログラムします。
- エラー信号に対して、カードは同じ TS キャラクタを再試行し、再プログラムされた USART によって今度は正しく受信されます。

または、パリティエラー割り込みに応答して、ソフトウェアは USART を再プログラムし、カードに対して新しいリセットコマンドを生成してから、TS を再び待つこともできます。

方式 2

USART は 9 ビット／パリティなしモード、ビット反転なしでプログラムされます。このモードでは、次のように 2 つの TS パターンのいずれかを受信します。

(H) LHHL LLL LLH = 0x103 -> インバースコンベンションを選択

(H) LHHL LLL LLH = 0x103 -> ダイレクトコンベンションを選択

ソフトウェアは受信されたキャラクタをこの 2 つのパターンと照合して、いずれかに一致した場合は、それに応じて、次のキャラクタ受信に備えて USART をプログラムします。

2 つのうちのどちらも認識されなかった場合、ネゴシエーションを再開するためにカードリセットが生成されます。

26.5.14 Ir SIR ENDEC ブロック

このセクションは、IrDA モードがサポートされるときにのみ適用されます。[セクション 26.4:USART の実装 \(681 ページ\)](#) を参照してください。

IrDA モードを選択するには、USARTx_CR3 レジスタの IREN ビットをセットします。IrDA モードでは、次のビットをクリアされた状態に保つ必要があります。

- USARTx_CR2 レジスタの LINEN、STOP、および CLKEN ビット
- USARTx_CR3 レジスタの SCEN および HDSEL ビット

IrDA SIR 物理層は、ロジック 0 を赤外光パルスとして表現する RZI (Return to Zero, Inverted) 変調方式の使用を指定します ([図 258](#) を参照)。

SIR 送信エンコーダは、USART からの NRZ (Non Return to Zero) 送信ビットストリーム出力を変調します。出力パルスストリームは、外部の出力ドライバと赤外線 LED に送信されます。SIR ENDEC の場合、USART は最大 115.2 Kbps のビットレートしかサポートしません。通常モードでは、送信されるパルス幅は、ビット周期の 3/16 と指定されます。

SIR 受信デコーダは、赤外線検出回路からの RZ (Return to Zero) ビットストリームを復調し、受信した NRZ シリアルビットストリームを USART に出力します。デコーダの入力は、アイドル状態のノーマルハイレベル (マーク状態) です。送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。デコーダ入力がローレベルのとき、スタートビットが検出されます。

- IrDA は半二重通信プロトコルです。トランスミッタがビジーである場合 (USART が IrDA エンコーダにデータを送信しているとき)、IrDA デコーダは IrDA 受信ライン上にあるすべてのデータを無視します。レシーバがビジーである (USART が IrDA デコーダからデコードされたデータを受信している) 場合、IrDA は USART から IrDA への TX 上のデータをエンコードしません。データの受信中は、送信データの破壊を防ぐために、送信を避けてください。
- 0 はハイパルスとして送信され、1 は 0 として送信されます。通常モードでは、パルスの幅は、選択されたビット周期の 3/16 と規定されます ([図 259](#) を参照)。
- SIR デコーダは、IrDA 準拠の受信信号を USART 用のビットストリームに変換します。
- SIR 受信ロジックは、ハイ状態を論理値 1 とみなし、ローパルスを論理値 0 とみなします。
- 送信エンコーダの出力は、デコーダ入力とは逆の極性になっています。SIR 出力は、アイドル時にロー状態になります。
- IrDA仕様では、1.41 us より大きなパルスを受け入れる必要があります。受け入れられるパルス幅は、プログラム可能です。レシーバ端のグリッチ検出回路は、PSC 2 周期 (PSC は USARTx_GTPR でプログラムされたプリスケール値) より小さな幅のパルスをフィルタします。PSC 1 周期より小さな幅のパルスは常に拒否されますが、1 周期以上 2 周期未満の幅のパルスは受け入れられることも、拒否されることもあります。2 周期より大きな幅のパルスは、パルスとして受け入れられます。PSC=0 のとき、IrDA エンコーダ/デコーダは機能しません。
- レシーバは、低電力トランスミッタと通信できます。
- IrDA モードでは、USARTx_CR2 レジスタのストップビットを「1 ストップビット」に設定する必要があります。

IrDA 低電力モード

トランスミッタ

低電力モードでは、パルス幅はビット周期の 3/16 に維持されません。代わりに、パルス幅は低電力ボーレート（最小で 1.42 MHz）の 3 倍となります。

一般に、この値は 1.8432 MHz ($1.42 \text{ MHz} < \text{PSC} < 2.12 \text{ MHz}$) です。低電力モードのプログラム可能な分周器は、この値を得るためにシステムクロックを分周します。

レシーバ

低電力モードでの受信は、通常モードでの受信と同様です。グリッチ検出の場合、USART は 1 PSC よりも短いパルスを破棄する必要があります。有効なローレベルは、その期間が IrDA 低電力ボーレート（USARTx_GTPR の PSC 値）の 2 周期分を超える場合にのみ受け入れられます。

注： **PSC 2 周期未満 1 周期以上の幅のパルスは、拒否されることも、拒否されないこともあります。**
レシーバのセットアップ時間は、ソフトウェアで管理してください。IrDA 物理層仕様では、送信と受信の間に最小 10 ms の遅延を指定しています (IrDA は半二重プロトコルです)。

図 258. IrDA SIR ENDEC- ブロック図

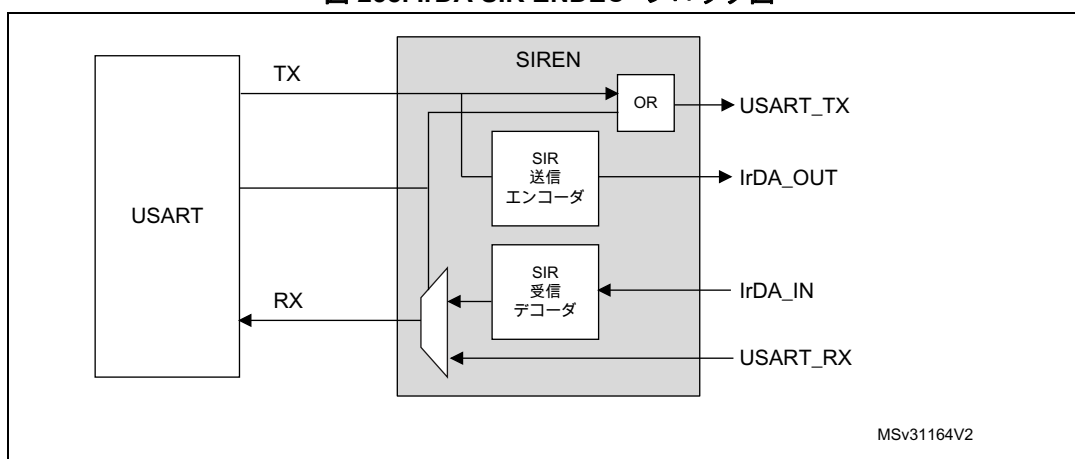
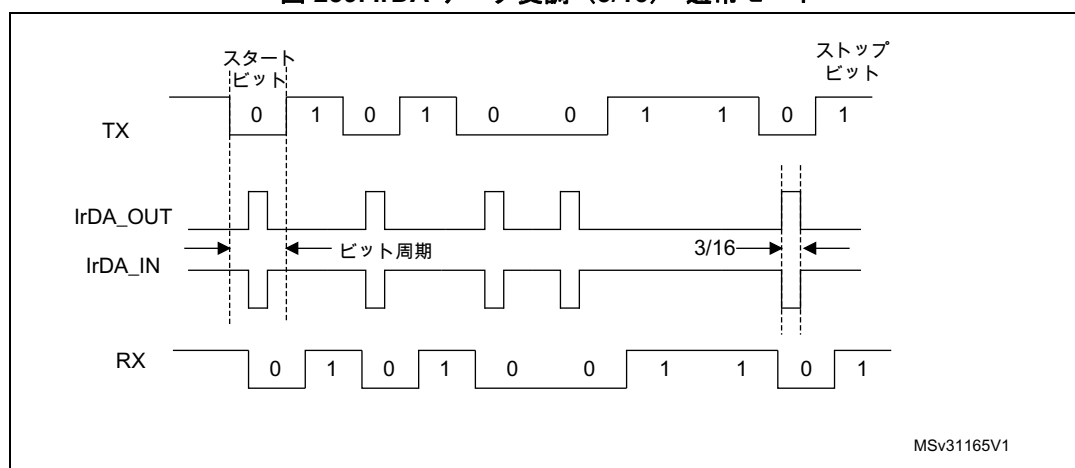


図 259. IrDA データ変調 (3/16) - 通常モード



26.5.15 DMA を使用した連続通信

USART は、DMA を使用して連続通信を行うことができます。Rx バッファと Tx バッファに対する DMA リクエストは、それぞれ独立して生成できます。

注： *DMA モードがサポートされるかどうかについては、[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。DMA がサポートされない場合は、[セクション 26.5.2 : トランスミッタ](#) または [セクション 26.5.3 : レシーバ](#) での説明に従って USART を使用してください。連続通信を行うには、USARTx_ISR レジスタの TXE/ RXNE フラグをクリアします。*

コード例については、「付録」セクション [A.19.11 : USART DMA のコード例](#) を参照してください。

DMA を使用した送信

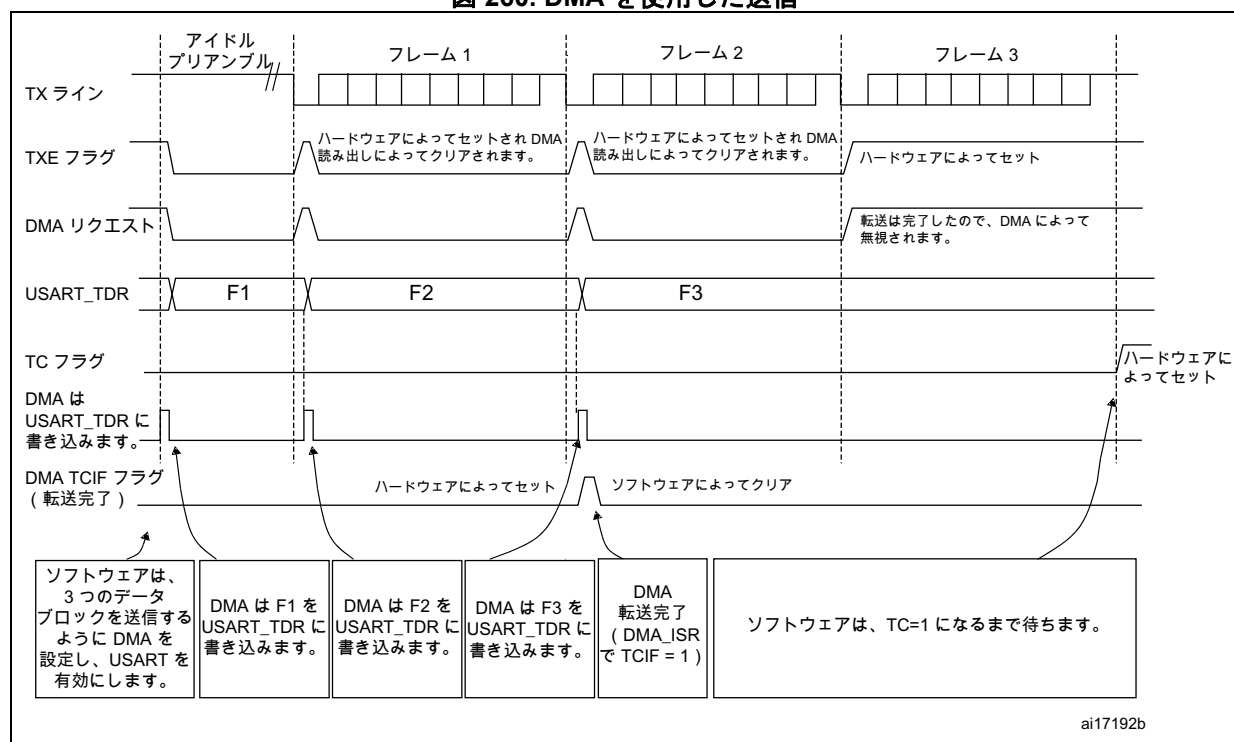
DMA モードでの送信を有効にするには、USARTx_CR3 レジスタの DMAT ビットをセットします。TXE ビットがセットされるたびに、データは、DMA ペリフェラル ([セクション 11 : ダイレクトメモリアクセスコントローラ \(DMA\) \(197 ページ\)](#)) を参照) を使用して設定された SRAM 領域から USARTx_TDR レジスタにロードされます。DMA チャンネルを USART 送信用に割り付けるには、次の手順を実行します (x はチャンネル番号を示します)

1. DMA 制御レジスタに USARTx_TDR レジスタのアドレスを書き込み、これを転送先として設定します。データは、各 TXE イベント後にメモリからこのアドレスに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送元として設定します。データは、各 TXE イベント後に、このメモリ領域から USARTx_TDR レジスタにロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の DMA 割り込み生成を設定します。
6. USARTx_ICR レジスタの TCCF ビットをセットすることによって、USARTx_ISR レジスタの TC フラグをクリアします。
7. DMA レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

送信モードでは、送信すべきすべてのデータを DMA が書き込むと (DMA_ISR レジスタの TCIF フラグがセットされます)、TC フラグを観察して USART 通信の完了を確認することができます。これは、USART を無効にしたり STOP モードに入ったりする前に、最後の送信が壊れないようにするために必要です。ソフトウェアは、TC=1 になるまで待つ必要があります。TC フラグは、すべてのデータ転送中、クリアされたままであり、最後のフレームの送信終了時にハードウェアによってセットされます。

図 260. DMA を使用した送信



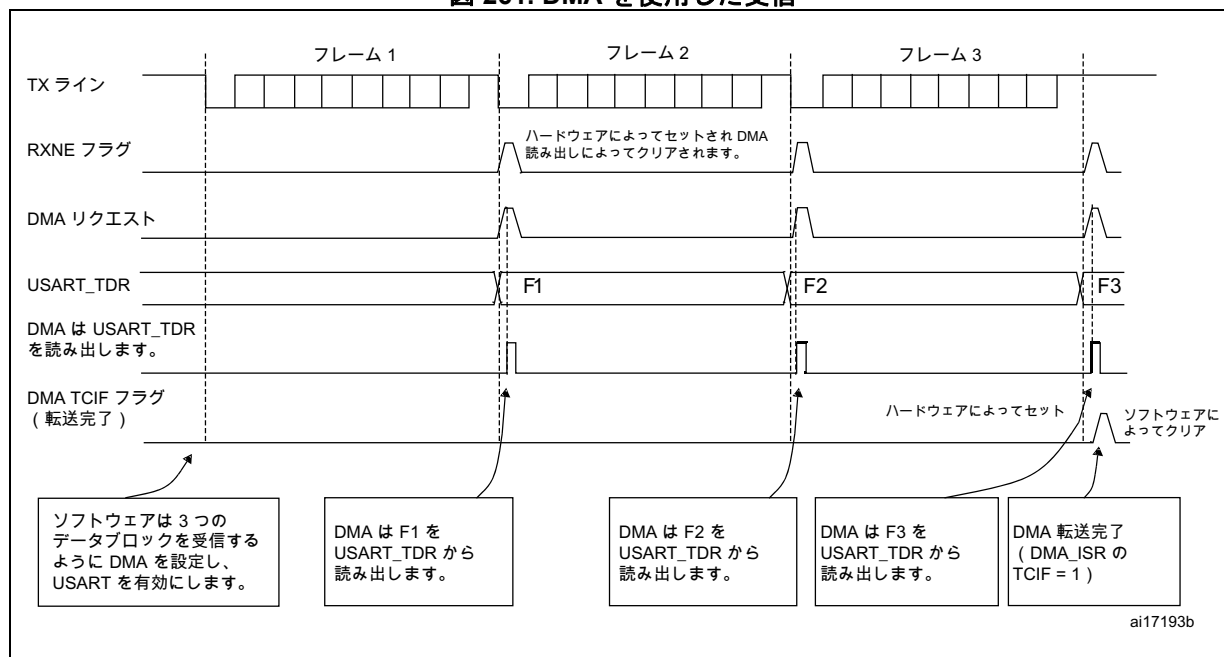
DMA を使用した受信

DMA モードでの受信を有効にするには、USARTx_CR3 レジスタの DMAR ビットをセットします。データは、データバイトが受信されると、USARTx_RDR レジスタから、DMA ペリフェラル ([セクション 11: ダイレクトメモリアクセスコントローラ \(DMA\) \(197 ページ\)](#)) を参照) を使用して設定された SRAM 領域にロードされます。DMA チャンネルを USART 受信用に割り付けるには、次の手順を実行します。

1. DMA 制御レジスタに USARTx_RDR レジスタのアドレスを書き込み、これを転送元として設定します。データは、各 RXNE イベント後に、このアドレスからメモリに移動されます。
2. DMA 制御レジスタにメモリアドレスを書き込み、これを転送先として設定します。データは、各 RXNE イベント後に、USARTx_RDR からこのメモリ領域にロードされます。
3. 転送すべきバイト総数を DMA 制御レジスタに設定します。
4. チャンネル優先順位を DMA 制御レジスタで設定します。
5. アプリケーションで必要とされる 1/2 転送終了、転送完了後の割り込み生成を設定します。
6. DMA 制御レジスタのチャンネルを有効にします。

DMA コントローラにプログラミングされたデータ転送数に達すると、DMA コントローラは、DMA チャンネルの割り込みベクタに基づいて割り込みを生成します。

図 261. DMA を使用した受信



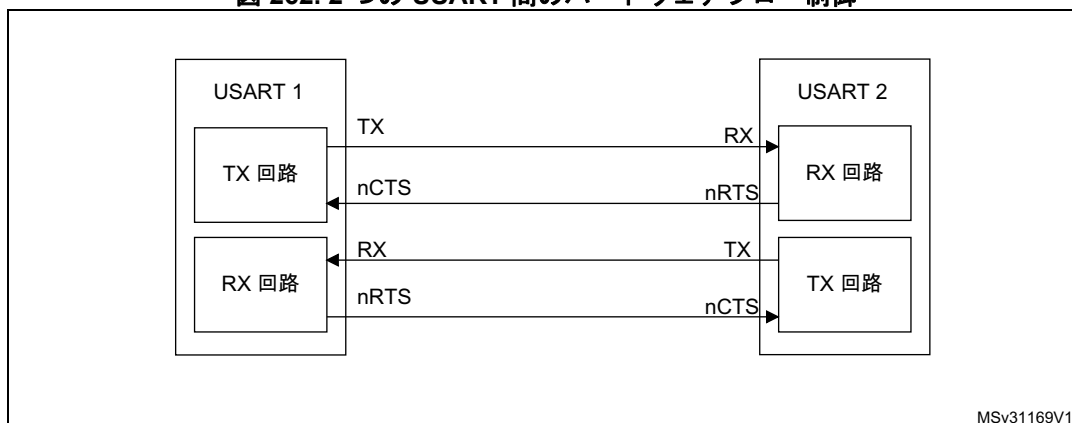
マルチバッファ通信における割り込み生成とエラーフラグ

マルチバッファ通信でトランザクション中にエラーが発生した場合、現在のバイトの後でエラーフラグがアサートされます。割り込み有効フラグがセットされている場合、割り込みが生成されます。1バイト受信においてRXNEとともにアサートされるフレーミングエラー、オーバーランエラー、およびノイズフラグに関しては、別のエラーフラグ割り込み有効ビット (USARTx_CR3レジスタのEIEビット)があり、これがセットされている場合、いずれかのエラーが発生すると、現在のバイトの後で割り込みが有効になります。

26.5.16 RS232 ハードウェアフロー制御およびRS485 ドライバ有効

nCTS入力とnRTS出力を使用すると、2つのデバイス間でシリアルデータフローを制御できます。[図 262](#)に、このモードで2つのデバイスを接続する方法を示します。

図 262. 2つのUSART間のハードウェアフロー制御

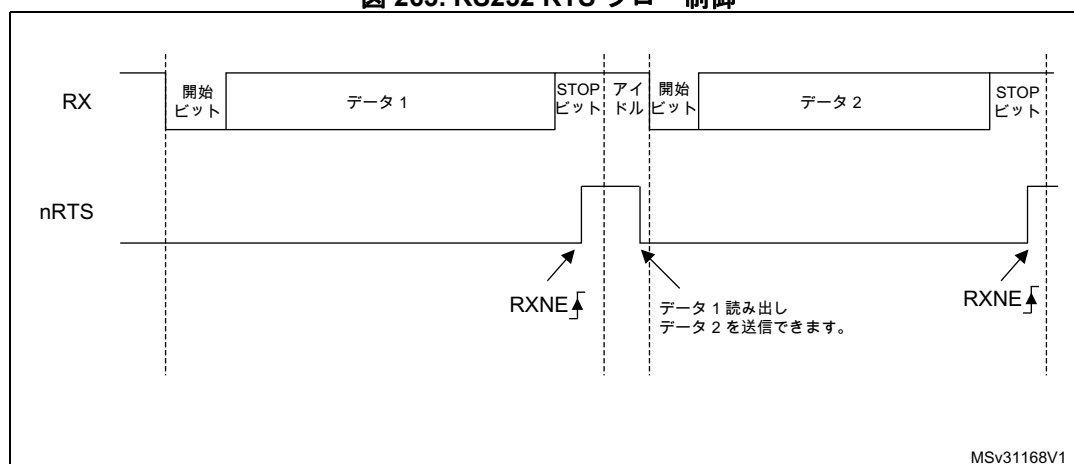


RS232 RTSとCTSのフロー制御は、USARTx_CR3レジスタのRTSEビットとCTSEビットにそれぞれ1を書き込むことによって、個別に有効にできます。

RS232 RTS フロー制御

RTS フロー制御が有効な場合 (RTSE=1)、USART レシーバが新しいデータを受信可能である限り、nRTS がアサートされます (ローレベル接続)。受信レジスタが満杯になると nRTS がネゲートされ、現在のフレームの終わりに送信が停止する予定であることを示します。図 263 に、RTS フロー制御が有効な場合の通信例を示します。

図 263. RS232 RTS フロー制御

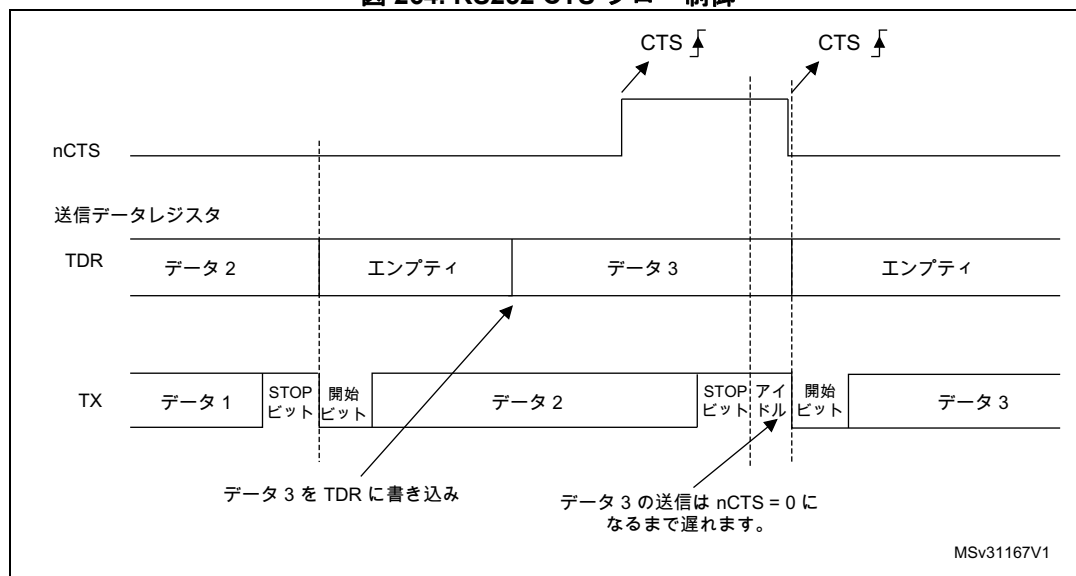


RS232 CTS フロー制御

CTS フロー制御が有効な場合 (CTSE=1)、トランスミッタは、nCTS 入力を検査してから、次のフレームを送信します。nCTS がアサートされた場合 (ローレベル接続)、次のデータが送信されます (データが送信されると想定、つまり TXE=0 の場合)。そうでない場合、送信は行われません。送信中に nCTS がネゲートされると、現在の送信が完了してから、トランスミッタが停止します。

CTSE=1 の場合、nCTS 入力がトグルすると、CTSIF ステータスビットはハードウェアによって自動的にセットされます。このビットは、レシーバの通信準備ができているかどうかを示します。USARTx_CR3 レジスタの CTSIE ビットがセットされている場合、割り込みが生成されます。図 264 に、CTS フロー制御が有効な場合の通信例を示します。

図 264. RS232 CTS フロー制御



注： 正しい動作のために、nCTS は、現在のキャラクタの終了の少なくとも 3 USART クロックソース周期前にアサートする必要があります。さらに、2 x PCLK 周期より短いパルスでは CTSCF フラグがセットされない場合があることに注意してください。

コード例については、「付録」セクション [A.19.12 : USART ハードウェアフロー制御のコード例](#) を参照してください。

RS485 ドライバ有効

ドライバ有効機能を有効にするには、USARTx_CR3 制御レジスタのビット DEM をセットします。これにより、DE (Driver Enable) 信号によって外部トランシーバ制御を有効にできます。アサーション時間は、DE 信号の有効化から START ビットの開始までの時間です。USARTx_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。ネゲート時間は、送信メッセージの最後のストップビットの終了から DE 信号の無効化までの時間です。USARTx_CR1 制御レジスタの DEAT [4:0] ビットフィールドを使用してプログラムされます。DE 信号の極性は、USARTx_CR3 制御レジスタの DEP ビットを使用して設定できます。

USART では、DEAT および DEDT はサンプル時間単位（オーバーサンプリングレートに応じて 1/8 または 1/16 ビット持続時間）で表されます。

26.5.17 STOP モードからのウェイクアップ

USART は、UESM ビットがセットされ、USART クロックが HSI または LSE にセットされているとき、MCU を STOP モードからウェイクアップできます（リセットおよびクロック制御 (RCC) のセクションを参照してください）。

STOP モードからの MCU のウェイクアップ は、標準の RXNE 割り込みを使用して行うことができます。この場合、STOP モードに入る前に RXNEIE ビットをセットする必要があります。

または、WUS ビットフィールドによって、特定の割り込みを選択することもできます。

MCU を STOP モードからウェイクアップできるようにするためには、STOP モードに入る前に、USARTx_CR1 制御レジスタの UESM ビットをセットする必要があります。

ウェイクアップイベントが検出されると、ハードウェアによって WUF フラグがセットされ、WUFIE ビットがセットされていた場合はウェイクアップ割り込みが生成されます。

注： STOP モードに入る前に、ユーザは USART が転送を行っていないことを確認する必要があります。
BUSY フラグでは、受信中に STOP モードに入らないことを保証できません。
WUF フラグは、MCU が STOP モードか、アクティブモードかに関係なく、ウェイクアップイベントが検出されたときにセットされます。
初期化とレシーバの有効化の直後に STOP モードに入るときには、REACK ビットをチェックして、USART が実際に有効であることを確認する必要があります。
受信に DMA が使用されるときには、STOP モードに入る前に無効化し、STOP モードの終了時に再び有効にする必要があります。
STOP モードからのウェイクアップは、すべてのモードで使用できるわけではありません。たとえば、SPI はマスタモードでのみ動作するので、SPI モードでは機能しません。

STOP モードでのミュートモードの使用

- USART は、STOP モードに入る前にミュートモードになります。
- アイドル検出は STOP モードでは機能しないので、アイドル検出時にミュートモードからウェイクアップすることはできません。
 - アドレス一致によるミュートモードからのウェイクアップが使用される場合、STOP モードからのウェイクアップのソースもアドレス一致でなければなりません。STOP モードに入るときに RXNE フラグがセットされる場合、アドレス一致によって STOP からウェイクアップしても、インタフェースはミュートモードのままです。
 - USART が START ビット検出時に MCU を STOP モードからウェイクアップするように設定された場合、WUF フラグはセットされますが、RXNE フラグはセットされません。

26.6 USART 低電力モード

表 100. 低電力モードが USART に与える影響

モード	説明
SLEEP	影響はありません。USART 割り込みによって、デバイスは SLEEP モードを終了します。
STOP	USART は、UESM ビットがセットされ、USART クロックが HSI または LSE にセットされているとき、MCU を STOP モードからウェイクアップできます。 STOP モードからの MCU のウェイクアップは、標準の RXNE または WUF 割り込みを使用して行うことができます。
STANDBY	USART はパワーダウンされ、デバイスが STANDBY モードを終了したときに再初期化する必要があります。

26.7 USART 割り込み

表 101. USART 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
送信データレジスタエンプティ	TXE	TXEIE
CTS 割り込み	CTSIF	CTSIE
送信完了	TC	TCIE



この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 101. USART 割り込みリクエスト (続き)

割り込みイベント	イベントフラグ	イネーブル制御ビット
受信データレジスタノットエンプティ (データの読み出し可能)	RXNE	RXNEIE
オーバーランエラー検出	ORE	
アイドルライン検出	IDLE	IDLEIE
パリティエラー	PE	PEIE
LIN ブレーク	LBDF	LBDIE
マルチバッファ通信におけるノイズフラグ、オーバーランエラー、およびフレーミングエラー。	NF または ORE または FE	EIE
キャラクター致	CMF	CMIE
レシーバタイムアウト	RTOF	RTOIE
ブロックの終了	EOBF	EOBIE
STOP モードからのウェイクアップ	WUF ⁽¹⁾	WUFIE

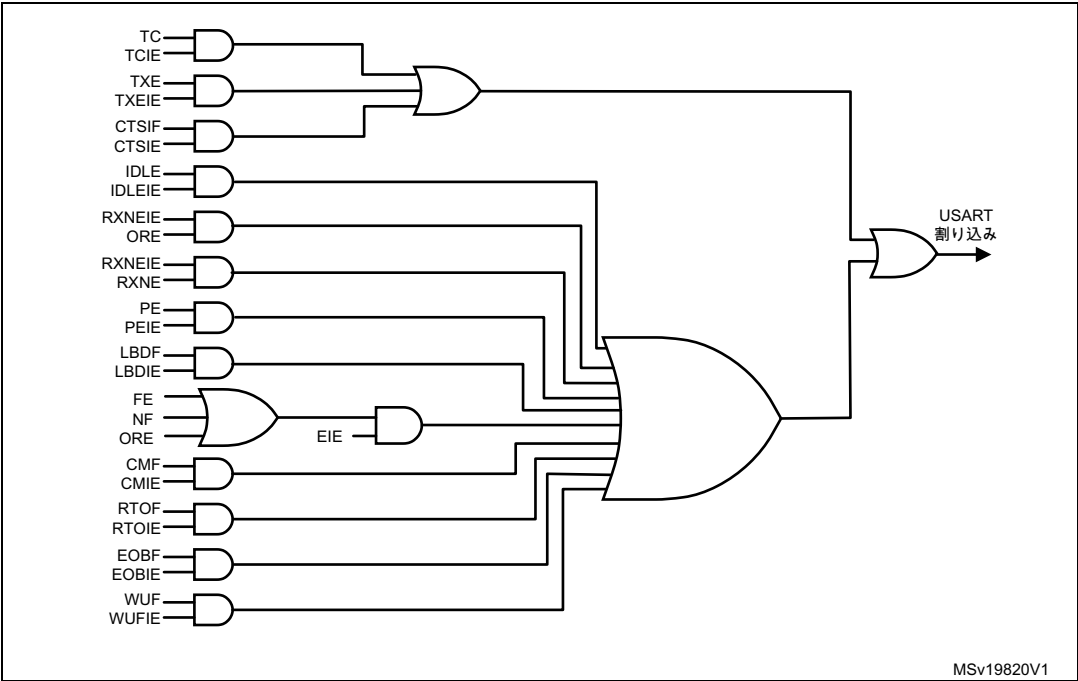
1. WUF 割り込みは、STOP モードでのみアクティブです。

USART 割り込みイベントは、同じ割り込みベクトルに接続されます (図 265 を参照)。

- 送信時: 送信完了、Clear to Send、送信データレジスタエンプティ、またはフレーミングエラー (スマートカードモード) 割り込み。
- 受信時: アイドルライン検出、オーバーランエラー、受信データレジスタノットエンプティ、パリティエラー、LIN ブレーク検出、ノイズフラグ、フレーミングエラー、キャラクター致など。

これらのイベントは、対応する有効制御ビットがセットされている場合に割り込みを生成します。

図 265. USART 割り込みマッピング図



MSv19820V1

26.8 USART レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

26.8.1 制御レジスタ 1 (USARTx_CR1)

アドレスオフセット : 0x00
リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:29 予約済みであり、リセット値のままにしておかなければなりません。

ビット 28 **M1** : ワード長

このビットとビット 12 (M0) によって、ワード長が決まります。ソフトウェアによってセット/クリアされます。
M[1:0] = 00: スタートビット 1 個、データビット 8 個、ストップビット n 個
M[1:0] = 01: スタートビット 1 個、データビット 9 個、ストップビット n 個
M[1:0] = 10: スタートビット 1 個、データビット 7 個、ストップビット n 個
このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注: 7 ビットデータ長モードでは、スマートカードモード、LIN マスタモード、および自動ボーレート (0x7F および 0x55 フレーム検出) はサポートされません。

ビット 27 **EOBIE** : ブロック終了割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。
0 : 割り込みは禁止されています。
1 : USARTx_ISR レジスタの EOBIF フラグがセットされると、USART 割り込みが生成されます。

注: USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。

ビット 26 **RTOIE** : レシーバタイムアウト割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。
0 : 割り込みは禁止されています。
1 : USARTx_ISR レジスタの RTOIF フラグがセットされると、USART 割り込みが生成されます。

注: USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#)。

ビット 25:21 **DEAT[4:0]** : ドライバ有効アサーション時間

この 5 ビット値は、DE (Driver Enable) 信号の有効化からスタートビットの開始までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット持続時間) で表されます。
このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注: ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 20:16 **DEDT[4:0]** : ドライバ有効ネゲート時間

この 5 ビット値は、送信メッセージの最後のストップビットの終了から DE (Driver Enable) 信号の無効化までの時間を定義します。サンプル時間単位 (オーバーサンプリングレートに応じて、1/8 または 1/16 ビット持続時間) で表されます。

DEDT 時間中に USARTx_TDR レジスタに書き込みが行われた場合、DEDT 時間と DEAT 時間の両方が経過するまで、新しいデータは送信されません。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注: **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 15 **OVER8** : オーバーサンプリングモード

0 : 16 倍のオーバーサンプリング

1 : 8 倍のオーバーサンプリング

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注: **LIN、IrDA、およびモードでは、このビットは常にクリア状態に保つ必要があります。**

ビット 14 **CMIE** : キャラクター致割り込み有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの CMF ビットがセットされると、USART 割り込みが生成されます。

ビット 13 **MME** : ミュートモード有効

このビットは、USART のミュートモード機能を有効にします。セットされると、USART は、WAKE ビットの定義に従って、アクティブモードとミュートモードを切り替えることができます。ソフトウェアによってセット/クリアされます。

0 : レシーバは永続的にアクティブモードです。

1 : レシーバはミュートモードとアクティブモードを切り替えることができます。

ビット 12 **M0** : ワード長

このビットとビット 28 (M1) によって、ワード長が決まります。ソフトウェアによってセット/クリアされます。ビット 28 (M1) の説明を参照してください。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 11 **WAKE** : レシーバウェイクアップ方式

このビットによって、ミュートモードからの USART のウェイクアップ方式が決まります。ソフトウェアによってセット/クリアされます。

0 : アイドルライン

1 : アドレスマーク

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 10 **PCE** : パリティ制御有効

このビットは、ハードウェアのパリティ制御 (生成と検出) を選択します。パリティ制御が有効なとき、算出されたパリティは MSB 位置 (M=1 の場合はビット 9、M=0 の場合はビット 8) に挿入され、受信されたデータではパリティが検査されます。このビットは、ソフトウェアによってセット/クリアされます。このビットがセットされると、送受信において現在のバイトの後で PCE が有効になります。

0 : パリティ制御は無効です。

1 : パリティ制御は有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 9 **PS** : パリティ選択

このビットは、パリティの生成/検出が有効である (PCE ビットがセットされている) とき、奇数パリティ/偶数パリティを選択します。ソフトウェアによってセット/クリアされます。パリティは、現在のバイトの後で選択されます。

0 : 偶数パリティ

1 : 奇数パリティ

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 8 PEIE : PE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの PE=1 のときには、USART 割り込みが生成されます。

ビット 7 TXEIE : 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの TXE=1 のときには、USART 割り込みが生成されます。

ビット 6 TCIE : 転送完了割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの TC=1 のときには、USART 割り込みが生成されます。

ビット 5 RXNEIE : RXNE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの ORE=1 または RXNE=1 のときには、USART 割り込みが生成されます。

ビット 4 IDLEIE : IDLE 割り込み有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの IDLE=1 のときには、USART 割り込みが生成されます。

ビット 3 TE : トランスミッタ有効

このビットは、トランスミッタを有効にします。ソフトウェアによってセット／クリアされます。

0 : トランスミッタは無効です。

1 : トランスミッタは有効です。

注 : 送信時、TE ビットに 0 パルス (0 の後に 1) を与えると、現在のワードの後にプリアンブル (アイドルライン) を送信します (スマートカードモードの場合を除きます)。アイドルキャラクタを生成するためには、すぐには TE に 1 を書き込まないでください。必要な時間を確保するために、ソフトウェアは USARTx_ISR レジスタの TEACK ビットをポーリングできます。

スマートカードモードでは、TE がセットされると、送信が開始されるまでに 1 ビット時間の遅れが生じます。

- ビット 2 RE : レシーバ有効

このビットは、レシーバを有効にします。ソフトウェアによってセット／クリアされます。

0 : レシーバは無効です。

1 : レシーバは有効であり、スタートビットの検索が開始されます。
- ビット 1 UESM : STOP モードでの USART 有効

このビットがクリアされると、USART は MCU を STOP モードからウェイクアップできません。

このビットがセットされると、USART は MCU を STOP モードからウェイクアップできますが、USART クロック選択が RCC にて HSI または LSE であることが条件です。

このビットは、ソフトウェアによってセット／クリアされます。

0 : USART は STOP モードから MCU をウェイクアップできません。

1 : USART は STOP モードから MCU をウェイクアップできます。この機能がアクティブなとき、USART のクロックソースは HSI または LSE でなければなりません（リセットおよびクロック制御（RCC）を参照）。

注：

STOP モードに入る直前に UESM ビットをセットし、STOP モードの終了時にクリアすることが推奨されます。

USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。
- ビット 0 UE : USART 有効

このビットがクリアされると、USART ブリスケーラと出力はただちに停止され、現在の操作は破棄されます。USART の設定は保たれますが、USARTx_ISR のステータスフラグはすべてそれぞれのデフォルト値にセットされます。このビットは、ソフトウェアによってセット／クリアされます。

0 : USART ブリスケーラと出力は無効であり、低電力モードです。

1 : USART は有効です。

注：

ラインにエラーを生成せずに低電力モードに入るためには、TE ビットをリセットする必要があり、ソフトウェアは USARTx_ISR の TC ビットがセットされるのを待ってから、UE ビットをリセットする必要があります。

UE=0 のときには DMA リクエストもリセットされるので、UE ビットをリセットする前に DMA チャンネルを無効にする必要があります。

26.8.2 制御レジスタ 2 (USARTx_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw				

ビット 31:28 ADD[7:4] : USART ノードのアドレス

このビットフィールドは、認識される USART ノードのアドレスまたはキャラクタコードを指定します。

これは、マルチプロセッサ通信において、7 ビットアドレスマーク検出によるウェイクアップのために、ミュータモードまたは STOP モード時に使用されます。トランスミッタによって送信されるキャラクタの MSB は 1 でなければなりません。通常の受信時、ミュータモードが無効なときに（たとえば、Modbus プロトコルのブロック終了検出）、キャラクタ検出のために使用することができます。この場合、受信されたキャラクタ全体（8 ビット）が ADD[7:0] 値と比較され、一致した場合は CMF フラグがセットされます。

このビットフィールドは、受信が無効のとき（RE=0）または USART が無効のとき（UE=0）のみ、書き込むことができます。

ビット 27:24 ADD[3:0] : USART ノードのアドレス

このビットフィールドは、認識される USART ノードのアドレスまたはキャラクタコードを指定します。

これは、マルチプロセッサ通信において、アドレスマーク検出によるウェイクアップのために、ミュータモードまたは STOP モード時に使用されます。

このビットフィールドは、受信が無効のとき（RE=0）または USART が無効のとき（UE=0）のみ、書き込むことができます。

ビット 23 RTOEN : レシーバタイムアウト有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : レシーバタイムアウト機能は無効です。

1 : レシーバタイムアウト機能は有効です。

この機能が有効なとき、RTOR（レシーバタイムアウトレジスタ）でプログラムされた時間にわたって RX ラインがアイドル（受信なし）であった場合、USARTx_ISR レジスタの RTOF フラグがセットされます。

注： USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。

ビット 22:21 ABRMOD[1:0] : 自動ボーレートモード

これらのビットは、ソフトウェアによってセット／クリアされます。

00 : スタートビットの測定がボーレートの検出に使用されます。

01 : 立ち下がりがエッジから立ち下がりがエッジまでの測定。（受信されたフレームはシングルビット = 1 で始まらなければならない、その場合、フレーム = Start10xxxxxx）

10 : 0x7F フレーム検出

11 : 0x55 フレーム検出

このビットフィールドは、ABREN=0 または USART が無効（UE=0）のときのみ書き込むことができます。

注： DATAINV=1 および／または MSBFIRST=1 の場合、パターンはライン上で同じである必要があります（たとえば、MSBFIRST の場合は 0xAA）。

USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。

ビット 20 ABREN : 自動ボーレート有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : 自動ボーレート検出は無効です。

1 : 自動ボーレート検出は有効です。

注： USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。

ビット 19 MSBFIRST : 最上位ビットファースト

このビットは、ソフトウェアによってセット／クリアされます。

0 : スタートビットに続いて、データはデータビット 0 から順に送受信されます。

1 : スタートビットに続いて、データは MSB（ビット 7/8/9）から順に送受信されます。

このビットフィールドは、USART が無効（UE=0）のときのみ書き込むことができます。

ビット 18 DATAINV : バイナリデータ反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : データレジスタからの論理データは正／ダイレクトロジックで送受信されます。(1=H、0=L)

1 : データレジスタからの論理データは、負／インバースロジックで送受信されます。(1=L、0=H) パリティビットも反転されます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 17 TXINV : TX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : TX ピン信号は標準ロジックレベルを使用して機能します ($V_{DD}=1$ /アイドル、Gnd=0/マーク)。

1 : TX ピン信号値は反転されます。($V_{DD}=0$ /マーク、Gnd=1/アイドル)。

これにより、TX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 16 RXINV : RX ピンアクティブレベル反転

このビットは、ソフトウェアによってセット／クリアされます。

0 : RX ピン信号は標準ロジックレベルを使用して機能します ($V_{DD}=1$ /アイドル、Gnd=0/マーク)。

1 : RX ピン信号値は反転されます。($V_{DD}=0$ /マーク、Gnd=1/アイドル)。

これにより、RX ラインで外部インバータを使用できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 15 SWAP : TX/RX ピンのスワップ

このビットは、ソフトウェアによってセット／クリアされます。

0 : TX/RX ピンは標準ピンアウトでの定義に従って使用されます。

1 : TX および RX ピンの機能はスワップされます。これにより、別の USART へのクロスワイヤ接続の場合に動作できます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 14 LINEN : LIN モード有効

このビットは、ソフトウェアによってセット／クリアされます。

0 : LIN モードは無効です。

1 : LIN モードは有効です。

LIN モードでは、USARTx_RQR レジスタの SBKRQ ビットを使用して LIN 同期ブレイク (下位 13 ビット) を送信し、LIN 同期ブレイクを検出することができます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **USART が LIN モードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 13:12 STOP[1:0] : STOP ビット

このビットは、ストップビットのプログラミングに使用します。

00 : 1 個のストップビット

01 : 0.5 個のストップビット

10 : 2 個のストップビット

11 : 1.5 個のストップビット

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 11 CLKEN : クロック有効

このビットによって、SCLK ピンを有効にできます。

0 : SCLK ピンは無効です。

1 : SCLK ピンは有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **同期モードまたはスマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

- 注: UE ビット値に関係なく、CLKEN = 1 で SCLK が常に使用可能な場合にスマートカードに SCLK クロックを正しく供給するには、次のステップを順守する必要があります。
- UE = 0
 - SCEN = 1
 - GTPR 設定 (PSC を設定する必要がある場合は、USARTx_ GTPR レジスタへのシングルアクセスで PSC と GT を設定することを推奨)
 - CLKEN = 1
 - UE = 1

ビット 10 CPOL : クロック極性

このビットによって、同期モードにおける SCLK ピンのクロック出力の極性を選択できます。CPHA ビットと連携して動作し、希望するクロック/データ関係になるようにします。

0 : 送信ウィンドウの外で、SCLK ピンはローレベルを維持します。

1 : 送信ウィンドウの外で、SCLK ピンはハイレベルを維持します。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

- 注: 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 9 CPHA : クロック位相

このビットは、同期モードでの SCLK ピンのクロック出力の位相を選択するために使用されます。CPOL ビットと連携して動作し、希望するクロック/データ関係になるようにします (図 253 および 図 254 を参照)。

0 : 最初のクロック遷移が最初のデータキャプチャエッジです。

1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

- 注: 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 8 LBCL : 最終ビットのクロックパルス

このビットは、同期モードで送信される最終データビット (MSB) に関連するクロックパルスを、SCLK ピンに出力する必要があるかどうかを選択するために使用されます。

0 : 最終データビットのクロックパルスは、SCLK ピンに出力されません。

1 : 最終データビットのクロックパルスは、SCLK ピンに出力されます。

- 注意: 最終ビットは、USARTx_CR1 レジスタの M ビットによって選択された 7 または 8 または 9 ビットフォーマットに応じて送信された 7 番目または 8 番目または 9 番目のデータビットです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

- 注: 同期モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 LBDIE : LIN ブレーク検出割込み有効

ブレーク割込みマスクです (ブレークデリミタを使用したブレーク検出)。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの LBDF=1 のときには、割り込みが生成されます。

- 注: LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 5 **LBDL** : LIN ブレーク検出長
このビットでは、10 ビットと 11 ビットのブレーク検出を選択します。
0 : 10 ビットブレーク検出
1 : 11 ビットブレーク検出
このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。
注 : **LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 4 **ADDM7** : 7 ビットアドレス検出 / 4 ビットアドレス検出
このビットは、4 ビットアドレス検出と 7 ビットアドレス検出の選択に使用されます。
0 : 4 ビットアドレス検出
1 : 7 ビットアドレス検出 (8 ビットデータモード)
このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。
注 : **7 ビットおよび 9 ビットデータモードでは、アドレス検出は、それぞれ 6 ビットおよび 8 ビットアドレス (ADD[5:0] および ADD[7:0]) に対して行われます。**

ビット 3:0 予約済みであり、リセット値に保持する必要があります。

注 : **トランスミッタが有効なときには、3 つのビット (CPOL、CPHA、LBCL) に書き込まないでください。**

26.8.3 制御レジスタ 3 (USARTx_CR3)

アドレスオフセット : 0x08
リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFIE	WUS		SCARCNT2:0]			Res.
									rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	v	v	rw	rw	rw	rw

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22 **WUFIE** : STOP モードからのウェイクアップ割り込み有効
このビットは、ソフトウェアによってセット / クリアされます。
0 : 割り込みは禁止されています。
1 : USARTx_ISR レジスタの WUF=1 のときには、USART 割り込みが生成されます。
注 : **WUFIE は、STOP モードに入る前にセットする必要があります。**
WUF 割り込みは、STOP モードでのみアクティブです。
USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 21:20 **WUS[1:0]** : STOP モードからのウェイクアップ割り込みフラグ選択

このビットフィールドは、WUF を有効にするイベントを指定します (STOP モードからのウェイクアップフラグ)。

00 : WUF はアドレス一致時に有効になります (ADD[7:0] および ADDM7 による定義に従って)。

01 : 予約済み

10 : WuF はスタートビット検出時に有効になります。

11 : WUF は RXNE 時に有効になります。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。**

ビット 19:17 **SCARCNT[2:0]** : スマートカード自動再試行カウンタ

このビットフィールドは、スマートカードにおける送受信の再試行回数を指定します。

送信モードでは、送信エラーが生成されるまでの送信の自動再試行回数を指定します (FE ビットをセット)。

受信モードでは、受信エラーが生成されるまでの受信の試行エラー回数を指定します (RXNE および PE ビットをセット)。

このビットフィールドは、USART が無効 (UE=0) のときのみプログラムする必要があります。

USART が有効になると (UE=1)、このビットフィールドは再送信を停止するために 0x0 にのみ書き込み可能です。

0x0 : 再送信無効 - 送信モードでの自動再送信禁止

0x1 から 0x7 : 自動再送信試行回数 (信号エラーの生成前)

注 : **スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 16 予約済みであり、リセット値に保持する必要があります。

ビット 15 **DEP** : ドライバ有効極性選択

0 : DE 信号はアクティブハイです。

1 : DE 信号はアクティブローです。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 14 **DEM** : ドライバ有効モード

このビットにより、DE 信号によって外部トランシーバ制御を有効にできます。

0 : DE 機能は無効です。

1 : DE 機能は有効です。DE 信号は RTS ピンで出力されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **ドライバ有効機能がサポートされない場合、このビットは予約済みであり、クリア状態に保つ必要があります。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 13 **DDRE** : 受信エラー時 DMA 無効

0 : 受信エラーの場合、DMA は無効になります。対応するエラーフラグはセットされますが、RXNE は 0 に保たれ、オーバーランを防ぎます。結果として、DMA リクエストはアサートされないため、エラーのあるデータは転送されず (DMA リクエストなし)、次の正しい受信データが転送されます (スマートカードモードで使用する)。

1 : 受信エラーの後、DMA は無効化されます。対応するエラーフラグと RXNE がセットされます。エラーフラグがクリアされるまで、DMA リクエストはマスクされます。つまり、ソフトウェアは、まず、DMA リクエストを無効にするか (DMAR=0)、RXNE をクリアしてから、エラーフラグをクリアする必要があります。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **受信エラーは、パリティエラー、フレーミングエラー、またはノイズエラーです。**

ビット 12 OVRDIS : オーバーラン無効

このビットは、受信オーバーラン検出を無効にするために使用されます。

0 : オーバーランエラーフラグ、ORE は、受信データが読み出される前に新しいデータを受信したときにセットされます。

1 : オーバーラン機能は無効です。RXNE フラグがまだセットされている間に新しいデータを受信した場合、

ORE フラグはセットされず、新しく受信されたデータが USARTx_RDR レジスタの前の内容に上書きされます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： この制御ビットにより、データを読み出さずに通信フローをチェックできます。

ビット 11 ONEBIT : 1 サンプルビット方式有効

このビットによって、サンプル方式を選択できます。1 サンプルビット方式が選択されると、ノイズ検出フラグ (NF) が無効になります。

0 : 3 サンプルビット方式

1 : 1 サンプルビット方式

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 10 CTSIE : CTS 割り込み有効

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタの CTSIF=1 のときには、割り込みが生成されます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 9 CTSE : CTS 有効

0 : CTS ハードウェアフロー制御が無効です。

1 : CTS モードが有効です。データは nCTS 入力のアサート (0 に関係) されている場合にのみ転送されます。データの送信中に nCTS 入力 negate された場合、送信は停止前に完了します。nCTS が negate されている間にデータがデータレジスタに書き込まれた場合、nCTS がアサートされるまで送信は延期されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 8 RTSE : RTS 有効

0 : RTS ハードウェアフロー制御が無効です。

1 : RTS 出力は有効であり、レシーババッファにスペースがあるときにのみ、データがリクエストされます。現在の文字が転送された後、データの転送は停止すると期待されます。データを受信できるとき、nRTS 出力がアサートされます (0 にプルされます)。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 7 DMAT : DMA 有効トランスミッタ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードは送信に有効

0 : DMA モードは送信に無効です。

ビット 6 DMAR : DMA 有効レシーバ

このビットは、ソフトウェアでセット / クリアされます。

1 : DMA モードが受信に有効です。

0 : DMA モードが受信に無効です。

ビット 5 SCEN : スマートカードモード有効

このビットはスマートカードモードを有効にするために使用します。

0 : スマートカードモードが無効です。

1 : スマートカードモードが有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : *USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。*

ビット 4 NACK : スマートカード NACK 有効

0 : パリティエラーの際の NACK 転送が無効です。

1 : パリティエラー時の NACK 転送が有効です。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : *USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。*

ビット 3 HDSEL : 半二重選択

単線半二重モードの選択です。

0 : 半二重モードは選択されません。

1 : 半二重モードが選択されます。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

ビット 2 IRLP : IrDA 低電力

このビットは、通常と低電力の IrDA モードの選択に使用されます。

0 : 通常モード

1 : 低電力モード

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : *IrDA モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。*

ビット 1 IREN : IrDA モード有効

このビットは、ソフトウェアによってセット/クリアされます。

0 : IrDA は無効です。

1 : IrDA は有効です。

このビットは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : *IrDA モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。*

ビット 0 EIE : エラー割り込みイネーブル

エラー割り込み有効ビットは、フレーミングエラー、オーバーランエラー、またはノイズフラグ (USARTx_ISR レジスタの FE=1 または ORE=1 または NF=1) の場合に割り込み生成を有効にするために必要です。

0 : 割り込みは禁止されています。

1 : USARTx_ISR レジスタで FE=1、ORE=1、または NF=1 になると、割り込みが生成されます。

26.8.4 ボーレートレジスタ (USARTx_BRR)

このレジスタは、USART が無効 (UE=0) のときのみ書き込むことができます。自動ボーレート検出モードでハードウェアによって自動的に更新されます。

アドレスオフセット : 0x0C

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BRR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:16 予約済みであり、リセット値に保持する必要があります。

ビット 15:4 **BRR[15:4]**

BRR[15:4] = USARTDIV[15:4]

ビット 3:0 **BRR[3:0]**

OVER8 = 0 のとき、BRR[3:0] = USARTDIV[3:0]。

OVER8 = 1 のとき、

BRR[2:0] = USARTDIV[3:0] であり、右に 1 ビットシフトされます。

BRR[3] は、クリアされたままにする必要があります。

26.8.5 ガード時間およびプリスケアラレジスタ (USARTx_GTPR)

アドレスオフセット : 0x10

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GT[7:0]								PSC[7:0]							
rw								rw							

ビット 31:16 予約済みであり、リセット値のままにしておかなければなりません。

ビット 15:8 **GT[7:0]** : ガード時間値

このビットフィールドは、ガード時間値をボークロック周期数でプログラムするために使用します。これはスマートカードモードで使用します。このガード時間値の後は転送完了フラグがセットされます。

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 7:0 **PSC[7:0]** : プリスケアラ値

IrDA 低電力および IrDA 通常モード :

PSC[7:0] = IrDA 通常および低電力ボーレート

USART クロックソースを分周して低電力周波数を得るためのプリスケアラのプログラミングに使用します。

ソースクロックは、レジスタに与えられた値 (上位 8 ビット) で分周されます。

00000000 : 予約済み - この値はプログラミングしないでください。

00000001 : クロックソースは 1 で分周されます。

00000010 : クロックソースは 2 で分周されます。

...

スマートカードモード :

PSC[4:0] : プリスケアラ値

USART クロックソースを分周してスマートカードのクロックを提供するプリスケアラのプログラミングに使用します。

レジスタで指定された値 (上位 5 ビット) を 2 倍して、ソースクロック周波数の分周比を求めます。

00000 : 予約済み - この値はプログラミングしないでください。

00001 : クロックソースは 2 で分周されます。

00010 : クロックソースは 4 で分周されます。

00011 : クロックソースは 6 で分周されます。

...

このビットフィールドは、USART が無効 (UE=0) のときのみ書き込むことができます。

注 : **スマートカードモードが使用される場合、ビット [7:5] はクリアされたままにする必要があります。**

スマートカードモードや IrDA モードがサポートされない場合、このビットフィールドは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

26.8.6 レシーバタイムアウトレジスタ (USARTx_RTOR)

アドレスオフセット : 0x14

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BLEN[7:0]								RTO[23:16]							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 31:24 BLEN[7:0] : ブロック長

このビットフィールドは、受信時のスマートカード T=1 のブロック長を指定します。この値は、情報文字の数 + エピローグフィールドの長さ (1-LEC/2-CRC) - 1 と等しくなります。

例 :

BLEN = 0 -> 0 情報文字 + LEC

BLEN = 1 -> 0 情報文字 + CRC

BLEN = 255 -> 254 情報文字 + CRC (合計 256 文字)

スマートカードモードでは、TXE=0 のときにブロック長カウンタがリセットされます。

このビットフィールドは、他のモードでも使用できます。この場合、RE=0 (受信無効) のとき、および/または EOBCF ビットが 1 に書き込まれたときに、ブロック長カウンタがリセットされます。

注 : この値は、ブロック受信の開始後にプログラムできます (プロローグフィールドの LEN 文字のデータを使用)。必ず受信したブロックにつき一度だけプログラムするようにしてください。

ビット 23:0 RTO[23:0] : レシーバタイムアウト値

このビットフィールドは、レシーバタイムアウト値をビット持続時間の数で指定します。

標準モードでは、最後の受信キャラクタの後、RTO 値を超える間、新しいスタートビットが検出されなかった場合、RTOF フラグがセットされます。

スマートカードモードでは、この値は CWT および BWT を実装するために使用されます。詳細については、スマートカードの章を参照してください。

この場合、タイムアウト測定は最後の受信キャラクタのスタートビットから始めて行われます。

注 : この値は、受信キャラクタごとにプログラムされる必要があります。

注 : RTOR は、動作中に書き込むことができます。新しい値がカウンタ以下の場合、RTOF フラグがセットされます。

レシーバタイムアウト機能がサポートされない場合、このレジスタは予約済みであり、ハードウェアによって 0x00000000 に強制的に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

26.8.7 リクエストレジスタ (USARTx_RQR)

アドレスオフセット : 0x18

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w	w	w	w	w

ビット 31:5 予約済みであり、リセット値のままにしておかなければなりません。

ビット 4 **TXFRQ** : 送信データー掃リクエスト

このビットに 1 を書き込むと、TXE フラグがセットされます。

これにより、送信データを破棄できます。このビットは、エラー (NACK) によりデータが送信されなかった場合、および USARTx_ISR レジスタで FE フラグがアクティブである場合に、スマートカードモードでのみ使用する必要があります。

USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。

ビット 3 **RXFRQ** : 受信データー掃リクエスト

このビットに 1 を書き込むと、RXNE フラグがクリアされます。

これにより、受信したデータを読み出さずに破棄して、オーバーラン条件を避けることができます。

ビット 2 **MMRQ** : ミュートモードリクエスト

このビットに 1 を書き込むと、USART はミュートモードになり、RWU フラグがセットされます。

ビット 1 **SBKRQ** : ブレーク送信リクエスト

このビットに 1 を書き込むと、SBKF フラグがセットされ、送信マシンが使用可能になるとすぐに、ラインで BREAK を送信するリクエストが発行されます。

注: **アプリケーションが、まだ送信されていないものも含めて、以前に挿入されたすべてのデータに続いてブレークキャラクタを送信する必要がある場合、ソフトウェアは SBKRQ ビットをセットする前に、TXE フラグのアサートを持つ必要があります。**

ビット 0 **ABRRQ** : 自動ボーレートリクエスト

このビットに 1 を書き込むと、USARTx_ISR の ABRF フラグがリセットされ、次の受信データフレームでの自動ボーレート測定をリクエストします。

注: **USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。[セクション 26.4 : USART の実装 \(681 ページ\)](#) を参照してください。**

26.8.8 割り込みおよびステータスレジスタ (USARTx_ISR)

アドレスオフセット : 0x1C

リセット値 : 0x0200 00C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
									r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	Res.	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE
r	r		r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:25 予約済みであり、リセット値に保持する必要があります。

ビット 24:23 予約済みであり、リセット値に保持する必要があります。

ビット 22 REACK : 受信有効確認応答フラグ

このビットは、受信有効値が USART によって考慮されるときに、ハードウェアによってセット/リセットされます。

STOP モードからのウェイクアップがサポートされるとき、STOP モードに入る前に USART が受信できる状態であることを REACK フラグを使用して確認できます。

ビット 21 TEACK : 送信有効確認応答フラグ

このビットは、送信有効値が USART によって考慮されるときに、ハードウェアによってセット/リセットされます

USARTx_CR1 レジスタで TE=0 を書き込んだ後、TE=1 を書き込むことによってアイドルフレームリクエストが生成されるとき、TE=0 の最小周期を満たすために使用できます。

ビット 20 WUF : STOP モードからのウェイクアップフラグ

このビットは、ウェイクアップイベントが検出されたときに、ハードウェアによってセットされます。イベントは、WUS ビットフィールドによって定義されます。USARTx_ICR レジスタの WUCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR3 レジスタの WUFIE=1 である場合、割り込みが生成されます。

注： *UESM がクリアされると、WUF フラグもクリアされます。*

WUF 割り込みは、STOP モードでのみアクティブです。

USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 19 RWU : レシーバのミュートモードからのウェイクアップ

このビットは、USART がミュートモードかどうかを示します。ウェイクアップ/ミュートシーケンスが認識されたときに、ハードウェアによってクリア/セットされます。ミュートモード制御シーケンス (アドレスまたは IDLE) は、USARTx_CR1 レジスタの WAKE ビットによって選択されます。

IDLE モードでのウェイクアップが選択されたとき、このビットは USARTx_RQR レジスタの MMRQ ビットに 1 を書き込むことによって、ソフトウェアによってのみセットできます。

0 : レシーバはアクティブモードです。

1 : レシーバはミュートモードです。

ビット 18 SBKF : ブレーク送信フラグ

このビットは、ブレークキャラクタ送信がリクエストされたことを示します。USARTx_RQR レジスタの SBKRQ ビットに 1 を書き込むことによって、ソフトウェアによってセットされます。ブレーク送信のストップビット時に、ハードウェアによって自動的にリセットされます。

0 : ブレークキャラクタは送信されません。

1 : ブレークキャラクタは送信されます。

ビット 17 CMF : キャラクター致フラグ

このビットは、ADD[7:0] によって定義されたキャラクタが受信されたときに、ハードウェアによってセットされます。USARTx_ICR レジスタの CMCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR1 レジスタの CMIE=1 の場合、割り込みが生成されます。

0 : キャラクター致は検出されていません。

1 : キャラクター致が検出されました。

ビット 16 BUSY : ビジーフラグ

このビットは、ハードウェアによってセット/リセットされます。RX ラインで通信中 (スタートビットの検出時) はアクティブです。成否にかかわらず、受信終了時にリセットされます。

0 : USART はアイドルです (受信なし)。

1 : 受信中です。

ビット 15 ABRF : 自動ボーレートフラグ

このビットは、自動ボーレートがセットされたときにハードウェアによってセットされ (RXNE もセットされ、RXNEIE=1 の場合は割り込みが生成されます)、または、自動ボーレート操作が成功せずに完了したときにセットされます (ABRE=1) (この場合、ABRE、RXNE、および FE もセットされます)。新しい自動ボーレート検出をリクエストするために、USARTx_RQR レジスタの ABRRQ に 1 を書き込むことによって、ソフトウェアによってクリアされます。

注: USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 14 ABRE : 自動ボーレートエラー

このビットは、ボーレート測定が失敗した場合に、ハードウェアによってセットされます (範囲外のボーレートまたはキャラクタ比較の失敗)。

USARTx_CR3 レジスタの ABRRQ ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

注: USART が自動ボーレート機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 13 予約済みであり、リセット値に保持する必要があります。**ビット 12 EOBF** : ブロック終了フラグ

このビットは、完全なブロックが受信されたときに、ハードウェアによってセットされます (たとえば、T=1 スマートカードモード)。検出は、受信バイト数が BLEN + 4 以上である場合に行われます (ブロックの開始時から、プロローグを含む)。

USARTx_CR2 レジスタの EOBIE=1 である場合、割り込みが生成されます。

USARTx_ICR レジスタの EOBCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ブロック終了に達していません。

1 : ブロック終了 (文字数) に達しました。

注: スマートカードモードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 11 RTOF : レシーバタイムアウト

このビットは、RTOR レジスタでプログラムされたタイムアウト値が通信なしで経過したときに、ハードウェアによってセットされます。USARTx_ICR レジスタの RTOCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR2 レジスタの RTOIE=1 の場合、割り込みが生成されます。

スマートカードモードでは、タイムアウトは CWT または BWT タイミングに対応します。

0 : タイムアウト値に達していません。

1 : データを受信せずにタイムアウト値に達しました。

注: 時間が RTOR レジスタでプログラムされた値に等しい場合、2 つのキャラクタが分離され、RTOF はセットされません。この時間がこの値に 2 サンプル時間 (オーバーサンプリング方式によって 2/16 または 2/8) を加えた値を超える場合、RTOF フラグがセットされます。

カウンタは RE=0 の場合でもカウントしますが、RTOF は RE=1 のときのみセットされます。RE がセットされたときにタイムアウトがすでに経過していた場合、RTOF はセットされます。

USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 10 CTS : CTS フラグ

このビットは、ハードウェアによってセット/リセットされます。nCTS 入力ピンのステータスの反転コピーです。

0 : nCTS ラインはセットされました。

1 : nCTS ラインはリセットされました。

注: ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 9 CTSIF : CTS 割り込みフラグ

このビットは、CTSE ビットがセットされていた場合、nCTS 入力がトグルしたときにハードウェアによってセットされます。USARTx_ICR レジスタの CTSCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

また、USARTx_CR3 レジスタで CTSIE=1 であれば、割り込みが生成されます。

0 : nCTS ステータスラインでの変更はありません。

1 : nCTS ステータスラインで変更がありました。

注： ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。

ビット 8 LBDF : LIN ブレーク検出フラグ

このビットは、LIN ブレークが検出されると、ハードウェアによってセットされます。USARTx_ICR レジスタの LBDCF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR2 レジスタの LBDIE=1 である場合、割り込みが生成されます。

0 : LIN ブレークは検出されませんでした。

1 : LIN ブレークが検出されました。

注： USART が LIN モードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。

ビット 7 TXE : 送信データレジスタエンプティ

このビットは、USARTx_TDR レジスタの内容がシフトレジスタに転送されると、ハードウェアによってセットされます。USARTx_TDR レジスタへの書き込みによってクリアされます。

TXE フラグは、USARTx_RQR レジスタの TXFRQ に 1 を書き込んでクリアし、データを破棄することもできます (スマートカード T=0 モードでの送信失敗の場合のみ)。

USARTx_CR1 レジスタの TXEIE ビット=1 の場合、割り込みが生成されます。

0 : データはシフトレジスタに転送されません。

1 : データはシフトレジスタに転送されます。

注： このビットは、シングルバッファ送信時に使用されます。

ビット 6 TC : 送信完了

データを含むフレームの送信が完了し、TXE がセットされている場合、このビットはハードウェアによってセットされます。USARTx_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。USARTx_ICR レジスタの TCCF に 1 を書き込むことによって、または USARTx_TDR レジスタに書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR1 レジスタの TCIE=1 である場合、割り込みが生成されます。

0 : 送信は完了していません。

1 : 送信は完了しています。

注： TE ビットがリセットされ、送信中でなかった場合、TC ビットはただちにセットされます。

ビット 5 RXNE : 読み出しデータレジスタノットエンプティ

このビットは、RDR シフトレジスタの内容が USARTx_RDR レジスタに転送されると、ハードウェアによってセットされます。USARTx_RDR レジスタの読み出しによってクリアされます。RXNE フラグは、USARTx_RQR レジスタの RXFRQ に 1 を書き込むことによってクリアすることもできます。

USARTx_CR1 レジスタの RXNEIE=1 の場合、割り込みが生成されます。

0 : データは受信されていません。

1 : 受信データを読み出すことができます。

ビット 4 IDLE : アイドルライン検出

このビットは、アイドルラインが検出されたときに、ハードウェアによってセットされます。USARTx_CR1 レジスタの IDLEIE=1 である場合、割り込みが生成されます。USARTx_ICR レジスタの IDLECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : アイドルラインは検出されていません。

1 : アイドルラインが検出されました。

注 : *RXNE ビットがセットされるまで (新しいアイドルラインが発生するまで)、IDLE ビットは再びセットされません。*

ミュートモードが有効な場合 (MME=1)、USART がミュートでない場合 (RWU=0)、WAKE ビットによって選択されたミュートモードに関係なく、IDLE はセットされます。RWU=1 の場合、IDLE はセットされません。

ビット 3 ORE : オーバーランエラー

このビットは、RXNE=1 のときに、シフトレジスタで現在受信中のデータを RDR レジスタに転送する準備ができたときに、ハードウェアによってセットされます。USARTx_ICR レジスタの ORECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR1 レジスタの RXNEIE=1 または EIE=1 の場合、割り込みが生成されます。

0 : オーバーランエラーはありません。

1 : オーバーランエラーが検出されました。

注 : *このビットがセットされると、RDR レジスタの内容は失われませんが、シフトレジスタは上書きされます。EIE ビットがセットされている場合、マルチバッファ通信中に ORE フラグがセットされた場合、割り込みが生成されます。*

USARTx_CR3 レジスタの OVRDIS ビットがセットされると、このビットは永続的に 0 に強制設定されます (オーバーラン検出なし)。

ビット 2 NF : START ビットノイズ検出フラグ

このビットは、受信フレームでノイズが検出されるとハードウェアによってセットされます。USARTx_ICR レジスタの NFCF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

0 : ノイズは検出されていません。

1 : ノイズが検出されました。

注 : *このビットは、割り込みを生成する RXNE ビットと同時に出現するため、割り込みを生成しません。EIE ビットがセットされている場合、マルチバッファ通信中に NF フラグがセットされた場合、割り込みが生成されます。*

注 : *ラインがノイズフリーであるとき、NF フラグを無効にして、ONEBIT ビットに 1 をプログラミングして偏差に対する USART の許容誤差を増加させることができます (セクション 26.5.5 : クロック偏差に対する USART レシーバの許容誤差 (697 ページ) を参照)。*

ビット 1 FE : フレーミングエラー

このビットは、非同期化、過度なノイズ、またはブレークキャラクタが検出されたときに、ハードウェアによってセットされます。USARTx_ICR レジスタの FECF ビットに 1 を書き込むことによって、ソフトウェアによってクリアされます。

スマートカードモードでは、送信時、成功せずに (カードがデータフレームを NACK) 最大送信試行回数に達すると、このビットがセットされます。

USARTx_CR1 レジスタの EIE=1 の場合、割り込みが生成されます。

0 : フレーミングエラーは検出されていません。

1 : フレーミングエラーまたはブレークキャラクタが検出されました。

ビット 0 PE : パリティエラー

このビットは、レシーバモードでパリティエラーが発生したときに、ハードウェアによってセットされます。USARTx_ICR レジスタの PECF に 1 を書き込むことによって、ソフトウェアによってクリアされます。

USARTx_CR1 レジスタの PEIE=1 の場合、割り込みが生成されます。

0 : パリティエラーはありません。

1 : パリティエラー

26.8.9 割り込みフラグクリアレジスタ (USARTx_ICR)

アドレスオフセット : 0x20

リセット値 : 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											W			W	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EOBCF	RTOCF	Res.	CTSCF	LBDCF	Res.	TCCF	Res.	IDLECF	ORECF	NCF	FECF	PECF
			W	W		W	W		W		W	W	W	W	W

ビット 31:21 予約済みであり、リセット値に保持する必要があります。

ビット 20 **WUCF** : STOP モードからのウェイクアップフラグクリア

このビットに 1 を書き込むと、USARTx_ISR レジスタの WUF フラグがクリアされます。

注: **USART が STOP モードからのウェイクアップ機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。**

ビット 19:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **CMCF** : キャラクター致フラグクリア

このビットに 1 を書き込むと、USARTx_ISR レジスタの CMF フラグがクリアされます。

ビット 16:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **EOBCF** : ブロック終了クリアフラグ

このビットに 1 を書き込むと、USARTx_ISR レジスタの EOBF フラグがクリアされます。

注: **USART がスマートカードモードをサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 11 **RTOCF** : レシーバタイムアウトフラグクリア

このビットに 1 を書き込むと、USARTx_ISR レジスタの RTOF フラグがクリアされます。

注: **USART がレシーバタイムアウト機能をサポートしない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 10 予約済みであり、リセット値に保持する必要があります。

ビット 9 **CTSCF** : CTS フラグクリア

このビットに 1 を書き込むと、USARTx_ISR レジスタの CTSIF フラグがクリアされます。

注: **ハードウェアフロー制御機能がサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 8 **LBDCF** : LIN ブレーク検出クリアフラグ

このビットに 1 を書き込むと、USARTx_ISR レジスタの LBDF フラグがクリアされます。

注: **LIN モードがサポートされない場合、このビットは予約済みであり、ハードウェアによって強制的に 0 に設定されます。セクション 26.4 : USART の実装 (681 ページ) を参照してください。**

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **TCCF** : 送信完了フラグクリア

このビットに 1 を書き込むと、USARTx_ISR レジスタの TC フラグがクリアされます。

ビット 5 予約済みであり、リセット値に保持する必要があります。



- ビット 4 **IDLECF** : アイドルライン検出フラグクリア
このビットに 1 を書き込むと、USARTx_ISR レジスタの IDLE フラグがクリアされます。
- ビット 3 **ORECF** : オーバーランエラーフラグクリア
このビットに 1 を書き込むと、USARTx_ISR レジスタの ORE フラグがクリアされます。
- ビット 2 **NCF** : ノイズ検出フラグクリア
このビットに 1 を書き込むと、USARTx_ISR レジスタの NF フラグがクリアされます。
- ビット 1 **FECF** : フレーミングエラーフラグクリア
このビットに 1 を書き込むと、USARTx_ISR レジスタの FE フラグがクリアされます。
- ビット 0 **PECF** : パリティエラーフラグクリア
このビットに 1 を書き込むと、USARTx_ISR レジスタの PE フラグがクリアされます。

26.8.10 受信データレジスタ (USARTx_RDR)

アドレスオフセット : 0x24
リセット値 : 未定義

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

- ビット 31:9 予約済みであり、リセット値に保持する必要があります。
- ビット 8:0 **RDR[8:0]** : 受信データ値
受信データキャラクタを含みます。
RDR レジスタは、入カシフトレジスタと内部バスとの間にパラレルインタフェースを提供します (図 241 を参照)。
パリティを有効にして受信する場合、MSB ビットで読み出される値が受信したパリティビットです。

26.8.11 送信データレジスタ (USARTx_TDR)

アドレスオフセット : 0x28
リセット値 : 未定義

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
							rW	rW	rW	rW	rW	rW	rW	rW	rW



ビット 31:9 予約済みであり、リセット値に保持する必要があります。

ビット 8:0 **TDR[8:0]** : 送信データ値

送信されるデータキャラクタを含みます。

TDR レジスタは、内部バスと出力シフトレジスタとの間にパラレルインタフェースを提供します (図 241 を参照)。

パリティを有効にして (USARTx_CR1 レジスタの PCE ビットに 1 をセット) 送信しているとき、MSB (データ長に応じてビット 7 または 8) に書き込まれた値は、パリティによって置き換えられるため、無効です。

注: このレジスタは、TXE=1 のときのみ書き込むことができます。

26.8.12 USART レジスタマップ

次の表に、USART のレジスタマップとリセット値を示します。

表 102. USART レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x00	USARTx_CR1	Res	Res	Res	M1	EOBIE	RTOIE	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE						
	リセット値				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x04	USARTx_CR2	ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD1	ABRMOD0	ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP[1:0]		CKEN	CPOL	CPHA	LBCL	Res	LBIDIE	LBIDL	ADDM7	Res	Res	Res	Res						
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0										
0x08	USARTx_CR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUFIE	WUS		SCARCNT[2:0]			Res	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE						
	リセット値											0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x0C	USARTx_BRR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BRR[15:0]																					
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0x10	USARTx_GTPR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	GT[7:0]					PSC[7:0]																
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x14	USARTx_RTOR	BLEN[7:0]								RTO[23:0]																													
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x18	USARTx_RQR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ							
	リセット値																											0	0	0	0	0	0	0					
0x1C	USARTx_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	Res	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NF	FE	PE						
	リセット値										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0					
0x20	USARTx_ICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WUCF	Res	Res	CMCF	Res	Res	Res	Res	EOBCF	RTOCF	Res	CTSCF	LBDCF	Res	TCOF	Res	IDLECF	ORECF	NCF	FECF	PECF						
	リセット値												0			0					0	0		0	0		0		0	0	0	0	0	0					

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 102. USART レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
0x24	USARTx_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]														
	リセット値																								X	X	X	X	X	X	X	X	X						
0x28	USARTx_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]														
	リセット値																								X	X	X	X	X	X	X	X	X						

レジスタ境界アドレスについては、[セクション 2.2 \(46 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

27 シリアルペリフェラルインタフェース/I2S (SPI/I2S)

27.1 概要

SPI/I²S インタフェースを使用して、SPI プロトコルまたは I²S オーディオプロトコルに基づき外部デバイスと通信することができます。SPI または I²S モードはソフトウェアによって選択可能です。デバイスのリセット後は、デフォルトで SPI モトローラモードが選択されます。

SPI (シリアルペリフェラルインタフェース) プロトコルは、外部デバイスとの半二重、全二重、および単方向の同期シリアル通信をサポートしています。このインタフェースはマスタとして設定することも可能で、その場合、外部スレーブデバイスに通信クロック (SCK) を供給します。このインタフェースは、マルチマスタ設定で動作することもできます。

I²S プロトコルも、同期シリアル通信インタフェースです。スレーブまたはマスタモードで、全二重および半二重通信として動作することができます。フィリップス I²S 規格、MSB/LSB 詰め規格、PCM 規格など、4 つのオーディオ規格に対応できます。

27.2 SPI の主な機能

- マスタまたはスレーブ動作
- 3 本のラインでの全二重同期転送
- 2 本のラインでの半二重同期転送 (双方向データライン有り)
- 2 本のラインでの単方向同期転送 (単方向データライン有り)
- 4 ~ 16 ビットのデータサイズ選択
- マルチマスタモード機能
- 8 個のマスタモードポーレートプリスケアラ (最大周波数 $f_{PCLK}/2$)
- スレーブモード周波数 (最大周波数 $f_{PCLK}/2$)
- マスタとスレーブの両方に対するハードウェア/ソフトウェアによる NSS 管理 : マスタ/スレーブ動作の動的切り替え
- クロックの極性と位相をプログラム可能
- データ順序をプログラム可能 (MSB ファースト/LSB ファーストのシフト)
- 専用の送受信フラグ (割り込み機能付き)
- SPI バスビジステータスフラグ
- SPI モトローラモードをサポート
- ハードウェア CRC 機能による信頼性の高い通信 :
 - Tx モードでは CRC 値を最終バイトとして送信可能
 - 最終受信バイトに対する CRC エラーの自動チェック
- マスタモードの障害、オーバーランの各フラグ (割り込み機能付き)
- CRC エラーフラグ
- DMA 機能付きの 2 つの 32 ビット内蔵 Rx および Tx FIFO
- SPI TI モードをサポート

27.3 I2S の主な機能

- 半二重通信（トランスミッタまたはレシーバのみ）
- マスタまたはスレーブ動作
- 正確なオーディオサンプリング周波数（8～192 kHz）を実現するプログラム可能な 8 ビットのリニアプリスケアラ
- 16、24、または 32 ビットのデータフォーマット
- パケットフレームはオーディオチャネルによって 16 ビット（16 ビットデータフレーム）または 32 ビット（16、24、32 ビットデータフレーム）に固定。
- プログラム可能なクロック極性（定常状態）
- スレーブ送信モードのアンダーランフラグ、受信モード（マスタおよびスレーブ）のオーバーランフラグ、受信モードと送信モード（スレーブの場合のみ）のフレームエラーフラグ
- 送受信用の 16 ビットレジスタ（両チャネルサイドに対して 1 個のデータレジスタ）
- 以下の I²S プロトコルをサポート。
 - － フィリップス I²S 規格
 - － MSB 詰め規格（左詰め）
 - － LSB 詰め規格（右詰め）
 - － PCM 規格（16 ビットチャネルフレーム、または 32 ビットチャネルフレームに拡張された 16 ビットデータフレームでの、ショートおよびロングフレーム同期付き）
- データ方向は常に MSB ファースト。
- 送受信（16 ビット幅）用の DMA 機能
- 外部オーディオコンポーネントを駆動するためのマスタクロックを出力可能。周波数比は、 $256 \times F$ （ F_S はオーディオサンプリング周波数）に固定。

27.4 SPI/I2S の実装

表 103 このマニュアルでは、STM32F0xx デバイスでの SPI/I2S 実装を示します。

表 103. STM32F0xx SPI の実装⁽¹⁾

SPI の機能	STM32F03x	STM32F04x STM32F05x		STM32F07x STM32F09x	
	SPI1	SPI1	SPI2	SPI1	SPI2
ハードウェア CRC 計算	X	X	X	X	X
Rx/Tx FIFO	X	X	X	X	X
NSS パルスモード	X	X	X	X	X
I2S モード	X	X	-	X	X
TI モード	X	X	X	X	X

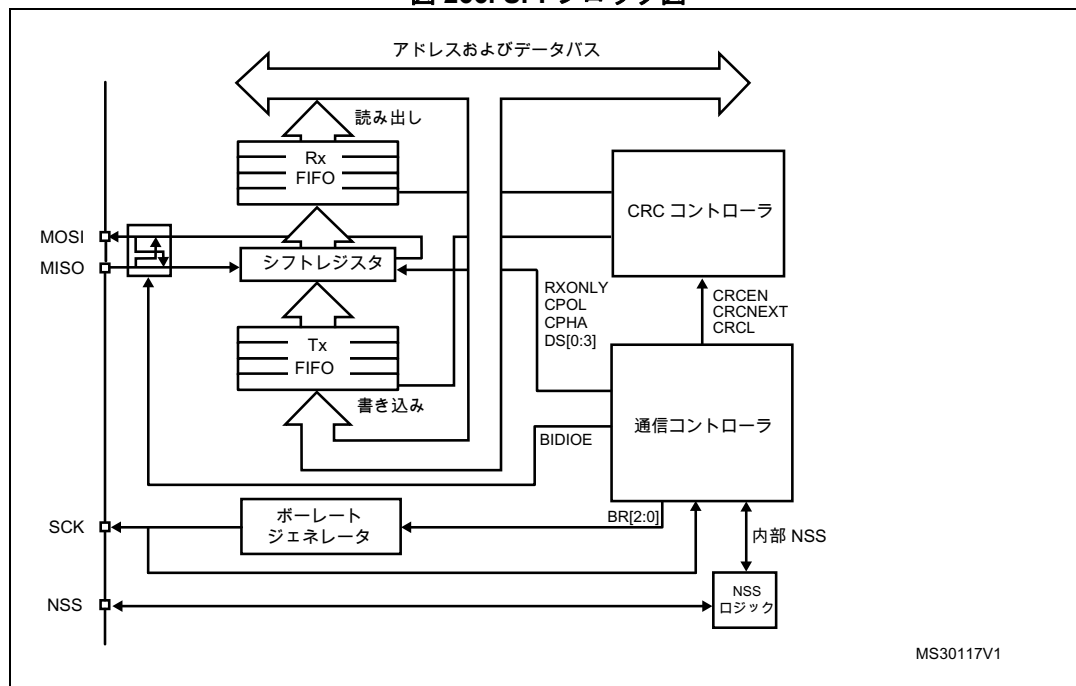
1. X：サポートされています。

27.5 SPI の機能説明

27.5.1 概要

SPI では、MCU と外部デバイス間の同期シリアル通信が可能です。アプリケーションソフトウェアは、ステータスフラグをポーリングするか、または専用の SPI 割り込みを使用することで、通信を管理することができます。SPI の主要要素およびそれらの相互作用を以下のブロック図 (図 266) に示します。

図 266. SPI ブロック図



4 本の I/O ピンが外部デバイスとの SPI 通信専用に使われます。

- **MISO** : マスターイン/スレーブアウトデータ。一般に、このピンは、スレーブモードではデータの送信に、マスタモードではデータの受信に使われます。
- **MOSI** : マスターアウト/スレーブインデータ。一般に、このピンは、マスタモードではデータの送信に、スレーブモードではデータの受信に使われます。
- **SCK** : SPI マスタではシリアルクロックの出力に、SPI スレーブでは入力に使われます。
- **NSS** : スレーブ選択用のピンです。このピンは、SPI および NSS の設定に応じて、以下のいずれかに使用できます。
 - － 個々の通信用スレーブデバイスを選択する
 - － データフレームを同期させる
 - － 複数のマスタ間での競合を検出する

詳細は、[セクション 27.5.4 : スレーブ選択 \(NSS\) ピンの管理](#) を参照してください。

SPI バスを使用することで、1 つのマスタデバイスと 1 つ以上のスレーブデバイスとの間で通信することができます。バスは 2 本以上の線から成り、1 本はクロック信号用、その他はデータの同期転送用です。SPI ノード間でのデータ交換とそれらのスレーブ選択信号管理に応じて、その他の信号を追加することができます。

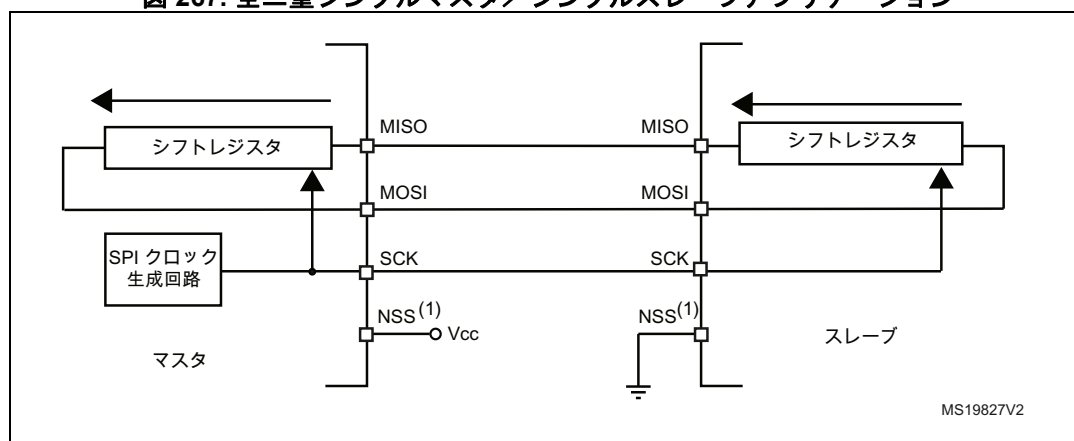
27.5.2 マスタとスレーブの 1 対 1 の通信

SPI を使用することで、MCU は対象となるデバイスやアプリケーション要件に応じたさまざまな設定で通信ができます。これらの設定には、2 または 3 本の線（ソフトウェア NSS 管理あり）、あるいは 3 または 4 本の線（ハードウェア NSS 管理あり）が使われます。通信は常にマスタによって開始されます。

全二重通信

SPI は、デフォルトで全二重通信に設定されます。この設定では、マスタおよびスレーブのシフトレジスタは、MOSI ピンと MISO ピンの間に 2 本の単方向ラインを介してリンクされます。SPI 通信の間、データはマスタから供給される SCK クロックのエッジに同期してシフトされます。マスタは、送信すべきデータを MOSI ライン経由でスレーブに送信し、MISO ライン経由でスレーブからデータを受信します。データフレーム転送が完了した（すべてのビットがシフトされた）時点で、マスタとスレーブの間で情報が交換されます。

図 267. 全二重シングルマスタ/シングルスレーブアプリケーション

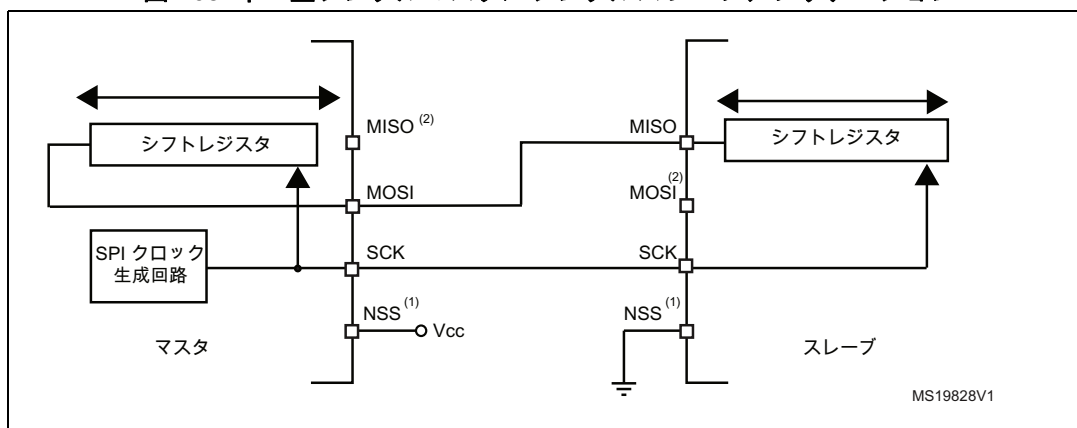


1. この場合、NSS ピンは入力として設定されます。

半二重通信

SPIx_CR1 レジスタの BIDIMODE ビットをセットすることで、SPI は半二重モードで通信できます。この設定では、1 本の交差接続ラインを使用して、マスタとスレーブのシフトレジスタを互いにリンクさせます。この通信中に、データは SCK クロックのエッジに同期して、シフトレジスタ間でシフトされます。シフトの方向は、マスタとスレーブの両方が SPIx_CR1 レジスタの BDIOE ビットを使用して相互に選択された転送方向となります。この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、他のアプリケーションで使用でき、GPIO として機能します。

図 268. 半二重シングルマスタ/シングルスレーブアプリケーション



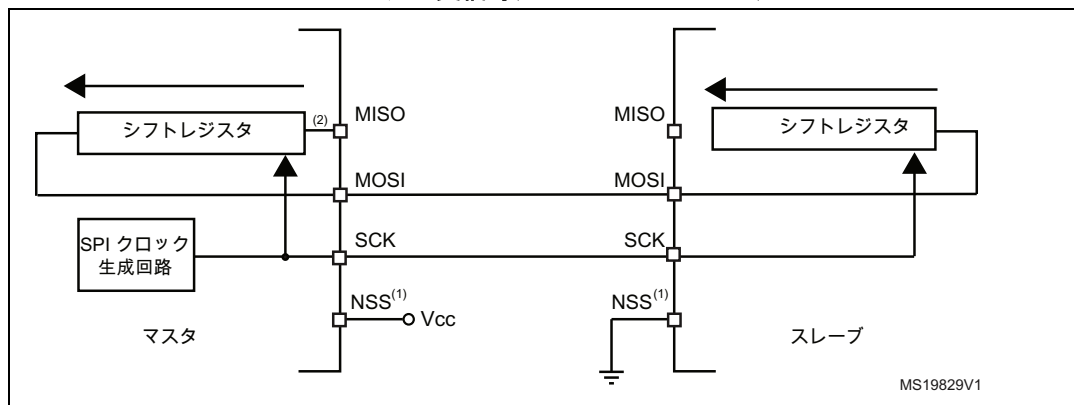
1. この場合、NSS ピンは入力として設定されます。
2. この設定では、マスタの MISO ピンとスレーブの MOSI ピンは、GPIO として使用できます。

単方向通信

SPI は、SPIx_CR2 レジスタの RXONLY ビットを使用して送信専用または受信専用に設定することにより、単方向モードで通信できます。この設定では、マスタとスレーブのシフトレジスタ間の転送に使用するのは 1 ラインのみです。残りの MISO ピンと MOSI ピンのペアは通信には使用されず、標準の GPIO として使用できます。

- **送信専用モード (RXONLY = 0) の場合**：設定は全二重の場合と同じです。アプリケーションは、未使用の入力ピンでキャプチャされた情報を無視する必要があります。このピンは標準の GPIO として使用できます。
- **受信専用モード (RXONLY = 1) の場合**：アプリケーションにて、RXONLY ビットをセットすることによって、SPI 出力機能を無効にできます。スレーブ設定では、MISO 出力が無効化され、ピンを GPIO として使用することができます。スレーブ選択信号がアクティブな間は、スレーブは MOSI ピンからデータを受信し続けます（[27.5.4 : スレーブ選択 \(NSS\) ピンの管理](#)を参照）。データバッファの設定に応じて、受信データイベントが出現します。マスタ設定では、MOSI 出力が無効化され、ピンを GPIO として使用することができます。SPI が有効である間はクロック信号が生成され続けます。クロックを停止させる唯一の方法は、クロックの設定に応じて、RXONLY ビットまたは SPE ビットをクリアし、MISO ピンからの受信パターンが終了し、データバッファ構造への書き込みが行われるまで待つことです。

図 269. 単方向シングルマスタ/シングルスレーブアプリケーション (送信専用モードのマスタ/受信専用モードのスレーブ)



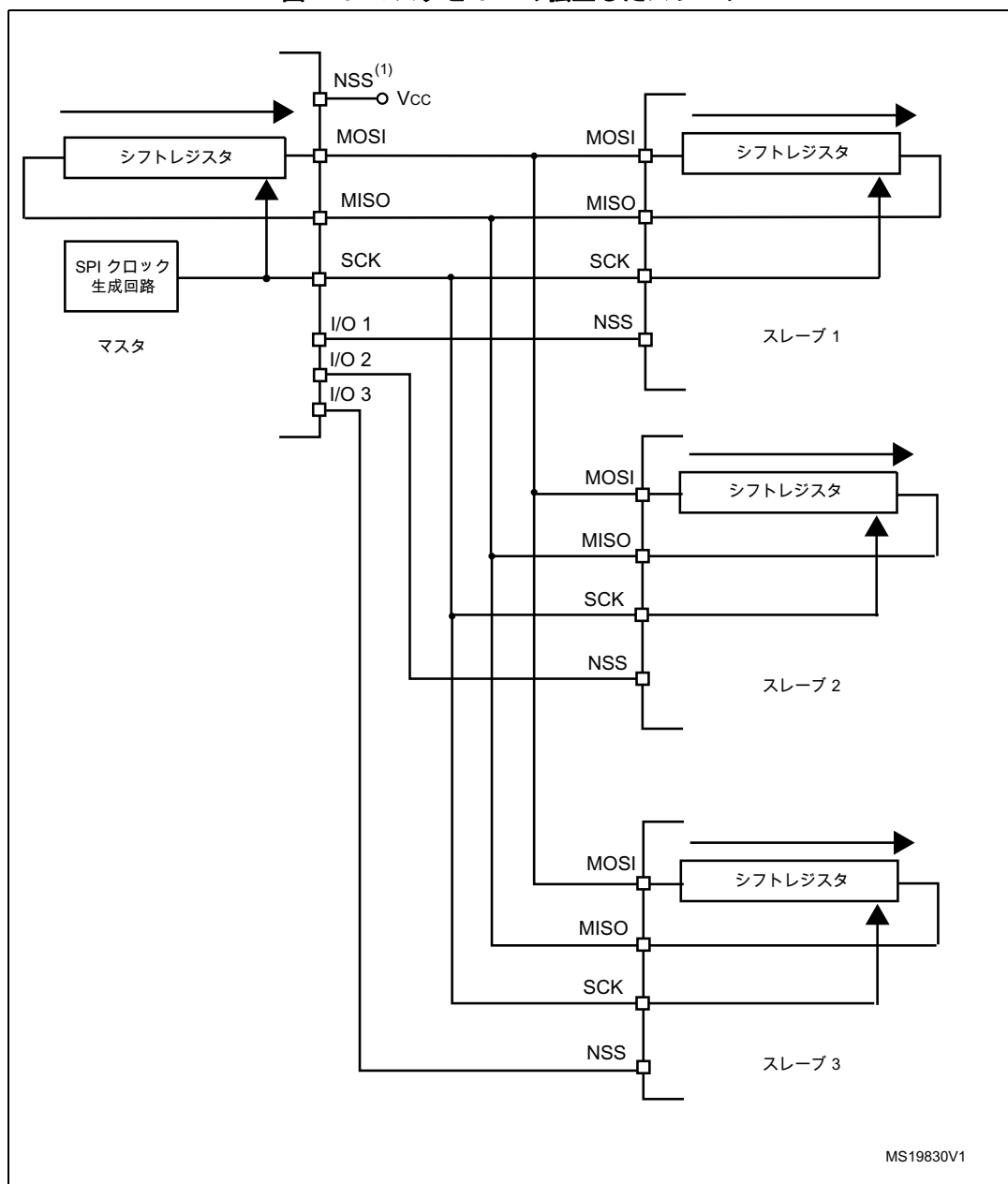
1. この場合、NSS ピンは入力として設定されます。
2. 入力情報はシフトレジスタでキャプチャされますが、標準の送信専用モード (たとえば、OVF フラグ) では無視する必要があります。
3. この設定では、両方のMISO ピンを GPIO として使用できます。

注： **すべての単方向通信は、トランザクション方向の設定を固定して (双方向モードはBDIO ビットが変化しない限り有効)、別の半二重通信に置き換えることができます。**

27.5.3 標準マルチスレーブ通信

2 つ以上の独立したスレーブがある設定の場合、マスタは GPIO ピンを使用して、各スレーブのチップセレクトラインを管理します (図 270 を参照)。マスタは、スレーブの NSS 入力に接続されている GPIO をローレベルにプルダウンすることによって、スレーブの 1 つを選択する必要があります。これを行うことにより、標準マスタと専用スレーブの通信が確立します。

図 270. マスタと 3 つの独立したスレーブ



1. スレーブの MISO ピンは相互接続されているので、すべてのスレーブにおいて、その MISO ピンの GPIO 設定をオルタネート機能オープンドレインとしてセットする必要があります ([セクション 9.3.7 : I/O オルタネート機能の入力/出力 \(161 ページ\)](#) を参照)。

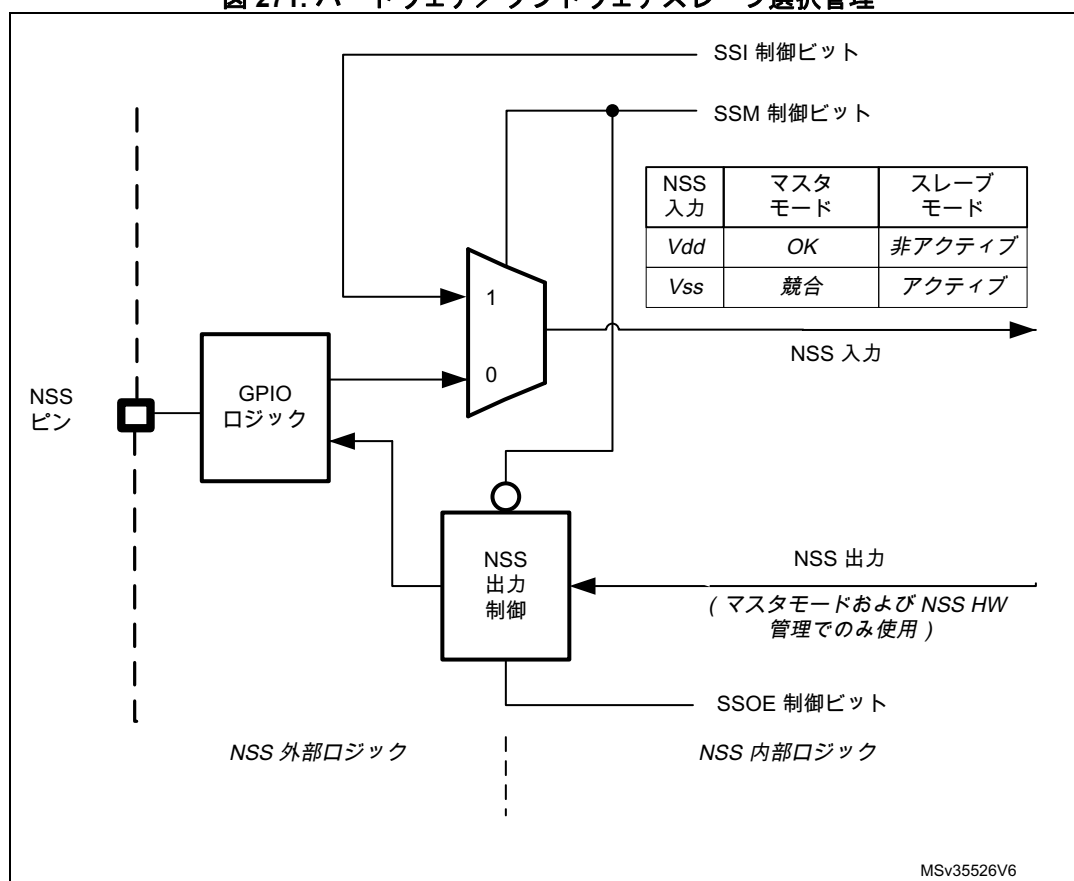
27.5.4 スレーブ選択 (NSS) ピンの管理

スレーブモードでは、NSS は標準の「チップセレクト」入力として機能し、スレーブをマスタと通信させます。マスタモードでは、NSS は出力としても入力としても使用できます。入力として使用する場合は、NSS はマルチマスタのバスの衝突を未然に防ぎ、出力として使用する場合は 1 つのスレーブのスレーブ選択信号を駆動させることができます。

ハードウェアまたはソフトウェアのスレーブ選択管理は、SPIx_CR1 レジスタの SSM ビットを使用して、以下のようにセットすることができます。

- **ソフトウェア NSS 管理 (SSM = 1)** : この設定では、スレーブ選択情報は SPIx_CR1 レジスタの SSI ビットの値によって内部で駆動されます。外部 NSS ピンは他のアプリケーションで使用できます。
- **ハードウェア NSS 管理 (SSM = 0)** : この場合、2 通りの設定が可能です。次のどちらの設定を使用するかは、NSS 出力設定 (SPIx_CR1 レジスタの SSOE ビット) によって決まります。
 - **NSS 出力が有効な場合 (SSM = 0, SSOE = 1)** : この設定は、MCU がマスタとしてセットされている場合にのみ使用します。NSS ピンはハードウェアによって管理されます。NSS 信号は、SPI がマスタモードで有効になる (SPE = 1) とすぐにローレベルに駆動され、SPI が無効化される (SPE = 0) までローレベルに保たれます。パルスは、NSS パルスモードが有効になると連続通信間で生成できます (NSSP = 1)。この NSS 設定では、SPI はマルチマスタ設定で機能させることはできません。
 - **NSS 出力が無効な場合 (SSM = 0, SSOE = 0)** : マイクロコントローラがバスでマスタとして機能している場合、この設定によりマルチマスタ機能が可能になります。このモードで、NSS ピンがローレベルにプルダウンされた場合、SPI はマスタモードのフォールト状態に入り、デバイスは自動的にスレーブモードに再設定されます。スレーブモードでは、NSS ピンは標準の「チップセレクト」入力として機能し、NSS ラインがローレベルの間はスレーブが選択されます。

図 271. ハードウェア/ソフトウェアスレーブ選択管理



27.5.5 通信フォーマット

SPI 通信中は受信と送信の操作が同時に行われます。シリアルクロック (SCK) は、データライン上で行われる情報のシフトとサンプリングを同期させます。通信フォーマットは、クロック位相、クロック極性、およびデータフレームフォーマットに応じて決定されます。マスタデバイスとスレーブデバイスの通信を可能にするには、双方が同じ通信フォーマットに従う必要があります。

クロックの位相および極性の制御

SPIx_CR1 レジスタの CPOL ビットと CPHA ビットを使用することによって、考えられる4つのタイミングの関係をソフトウェアで選択できます。CPOL (クロック極性) ビットは、データが転送されていないときのクロックのアイドル状態の値を制御します。このビットは、マスタモードとスレーブモードの両方に影響を与えます。CPOL がリセットされると、SCK ピンはローレベルのアイドル状態になります。CPOL がセットされると、SCK ピンはハイレベルのアイドル状態になります。

CPHA ビットがセットされると、SCK ピンの2番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがリセットされていれば立ち下がりエッジ、CPOL ビットがセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。CPHA ビットがリセットされている場合、SCK ピンの1番目のエッジがトランザクションの最初のデータビットをキャプチャします (CPOL ビットがセットされていれば立ち下がりエッジ、CPOL ビットがリセットされていれば立ち上がりエッジ)。データは、この種のクロック遷移が発生するたびにラッチされます。

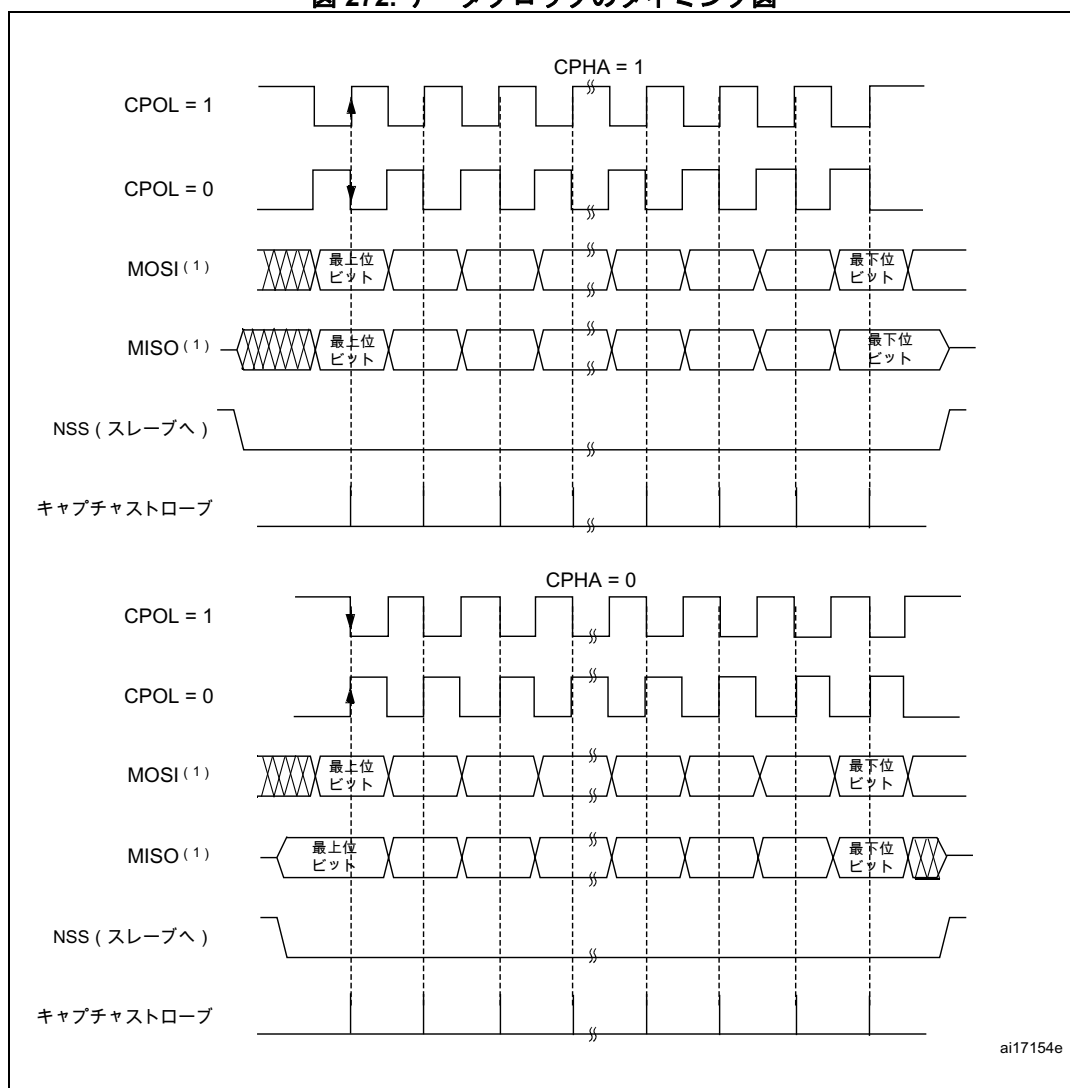
CPOL (クロック極性) ビットと CPHA (クロック位相) ビットの組み合わせによって、データキャプチャのクロックエッジを選択できます。

 272は、CPHA ビットと CPOL ビットの4つの組み合わせによる SPI 全二重転送を示しています。

注： *CPOL または CPHA ビットを変更する前に、SPE ビットをリセットすることによって、SPI を無効にする必要があります。*

SCK のアイドル状態は、SPIx_CR1 レジスタで (CPOL = 1 なら SCK のプルアップ、CPOL = 0 なら SCK のプルダウンによって) 選択された極性に一致する必要があります。

図 272. データクロックのタイミング図

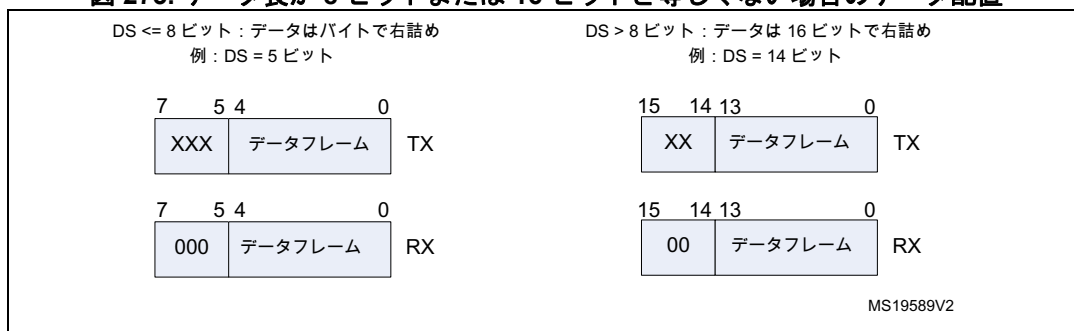


1. データビットの順序は LSBFIRST ビットの設定値に依存します。

データフレームフォーマット

LSBFIRST ビットの値に応じて、SPI シフトレジスタを設定することで、シフトをMSB ファーストまたは LSB ファーストに設定することができます。データフレームサイズは、DS ビットを使用して選択できます。4 ビットから 16 ビット長までの間にセットでき、この設定は送信と受信の両方に適用できます。選択されたデータフレームサイズにかかわらず、FIFO への読み出しアクセスは FRXTH レベルに整列されている必要があります。SPIx_DR レジスタにアクセスすると、データフレームはバイト（データがバイトに適合する場合）またはハーフワード（図 273 を参照）のいずれかに常に右詰めされます。通信中、データフレーム内のビットのみがクロック供給され、送信されます。

図 273. データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置



注： 最小データ長は 4 ビットです。4 ビット未満のデータ長が選択された場合は、8 ビットのデータフレームサイズに固定されます。

27.5.6 SPI の設定

設定手順は、マスタとスレーブではほぼ同じです。特定のモードの設定については、それぞれのモードに関する章を参照してください。標準通信を初期化する必要があるときは、以下の手順を実行します。

- 適切な GPIO レジスタに書き込みを行います。MOSI ピン、MISO ピン、SCK ピンの GPIO 設定を行います。
- SPI_CR1 レジスタに書き込みを行います。
 - BR[2:0] ビットを使用して、シリアルクロックボーレートを設定します（注：4 を参照）。
 - CPOL ビットと CPHA ビットの組み合わせを設定して、データ転送とシリアルクロックの 4 つの関係のうちの 1 つを定義します（CPHA は NSSP モードでクリアする必要があります）。（注：2 を参照）。
 - RXONLY または BIDIMODE、および BIDIOE を設定することによって（RXONLY と BIDIMODE は同時にセットできません）、単方向または半二重モードを選択します。
 - LSBFIRST ビットを設定して、フレームフォーマットを定義します（注：2 を参照）。
 - CRC が必要な場合は（SCK クロック信号がアイドル状態のとき）、CRCL と CRCEN ビットを設定します。
 - SSM と SSI を設定します（注：2 および 3 を参照）。
 - MSTR ビットを設定します（マルチマスタ NSS 設定では、MODF エラーを防ぐためにマスタが設定されている場合、NSS での競合を避けること）。
- 以下のように、SPI_CR2 レジスタに書き込みを行います。
 - 転送のデータ長を選択するために DS[3:0] ビットを選択します。
 - SSOE を設定します（注：1 および 2 および 3 を参照）。
 - TI プロトコルが必要な場合は FRF ビットをセットします（TI モードで NSSP ビットをクリアされたままにする）。
 - 2 つのデータユニット間で NSS パルスモードが必要な場合は NSSP ビットをセットします（NSSP モードで CHPA と TI ビットをクリアされたままにする）。
 - FRXTH ビットを設定します。RXFIFO 閾値は、SPIx_DR レジスタの読み出しアクセスサイズに揃える必要があります。
 - DMA がパックモードで使用される場合は、LDMA_TX および LDMA_RX ビットを初期化します。
- SPI_CRCPR レジスタに書き込みを行います。必要に応じて CRC 多項式を設定します。
- 適切な DMA レジスタに書き込みを行います。DMA ストリームが使用されている場合は、DMA レジスタに SPI Tx および Rx 専用の DMA ストリームを設定します。

- 注：
- (1) このステップはスレーブモードでは必要ありません。
 - (2) このステップはTI モードでは必要ありません。
 - (3) このステップはNSSP モードでは必要ありません。
 - (4) このステップは、スレーブモードにてTI モードで動作している場合を除き、スレーブモードでは必要ありません。

コード例については、「付録」セクション [A.17.1 : SPI マスタ設定のコード例](#) と [A.17.2 : SPI スレーブ設定のコード例](#) を参照してください。

27.5.7 SPI を有効にする手順

マスタがクロックを送信する前に、SPI スレーブを有効にすることを推奨します。さもなければ、望ましくないデータ送信が発生することがあります。スレーブのデータレジスタは、マスタとの通信を開始する前に、送信データをすでに格納していなければなりません（通信クロックの 1 番目のエッジに、またはクロック信号が連続的なときは現在の通信の最後の前に）。SPI スレーブが有効になる前に、SCK 信号を選択された極性に対応するアイドル状態のレベルに安定させる必要があります。

全二重（または送信専用モード）のマスタは、SPI が有効で TXFIFO がエンプティでない場合、または次の TXFIFO への書き込み時に通信を開始します。

あらゆるマスタ受信専用モードにおいて（RXONLY = 1 または BIDIMODE = 1、および BIDIOE = 0）、SPI が有効になるとすぐに、マスタは通信を開始し、クロックは動作を開始します。

DMA を処理するには、該当する章を参照してください。

27.5.8 データの送受信手順

RXFIFO および TXFIFO

すべての SPI データトランザクションは 32 ビット埋め込み FIFO を通過します。これにより、SPI は連続フローで動作できるようになり、またデータフレームサイズが短い場合にオーバーランを防ぐことができます。各方向にはそれぞれ TXFIFO と RXFIFO と呼ばれる固有の FIFO があります。これらの FIFO は、CRC 計算を有効にした状態で、受信専用モード（スレーブまたはマスタ）を除くすべての SPI モードで使用されます（[セクション 27.5.13 : CRC 計算](#)を参照）。

FIFO の扱いはデータ変換モード（二重、単方向）、データフレームフォーマット（フレーム内のビット数）、FIFO データレジスタで実行されるアクセスサイズ（8 ビットまたは 16 ビット）、および FIFO アクセス時にデータパッキングが使用されるかどうかに従います（[セクション 27.5.12 : TI モード](#)を参照）。

SPiX_DR レジスタへの読み出しアクセスからは、RXFIFO に保管された、まだ読み出されていない一番古い値が返されます。SPiX_DR への書き込みアクセスでは、送信キューの最後に TXFIFO に書き込まれるデータを保管します。読み出しアクセスは、必ず SPiX_CR2 レジスタの FRXTH ビットによって設定された RXFIFO 閾値に揃える必要があります。FTLVL[1:0] および FRLVL[1:0] ビットは、両方の FIFO について現在の占有レベルを示します。

SPiX_DR レジスタへの読み出しアクセスは RXNE イベントで管理する必要があります。このイベントは、データが RXFIFO に保管され、閾値（FRXTH ビットで定義される）に達した場合にトリガされます。RXNE がクリアされると、RXFIFO はエンプティであるとみなされます。同様に、送信するデータフレームの書き込みアクセスは TXE イベントで管理されます。このイベントは、TXFIFO レベルが容量の半分以下である場合にトリガされます。そうでない場合、TXE はクリアされ、TXFIFO がフルであるとみなされます。このように、データフレームフォーマットが 8 ビットを超えない場合、TXFIFO では最大 3 つのデータフレームのみを保管できるのに対し、RXFIFO では最大 4 つを保管することができます。この差異は、ソフトウェアが 16 ビットモードで TXFIFO により多くのデータの書き込みを試みた場合に、すでに TXFIFO に保管されている 3x 8 ビットデータフレームが破損する

可能性を防ぎます。TXE イベントと RXNE イベントの両方を割り込みによってポーリングまたは処理できます。[図 275](#) から [図 278](#) までを参照してください。

データ交換を管理するもう一つの方法は、DMA を使用することです（[セクション：DMA の主な機能を参照](#)）。

RXFIFO がフルのときに次のデータを受信した場合、オーバーランイベントが発生します（[セクション 27.5.9：SPI ステータスフラグの OVR フラグの説明を参照](#)）。オーバーランイベントは割り込みによってポーリングまたは処理できます。

セットされる BSY ビットは、現在のデータフレームの進行中のトランザクションを示します。クロック信号が流れ続けているときは、BSY フラグはマスタ側のデータフレーム間でセットされたままになりますが、スレーブの各データフレーム転送間の 1 つの SPI クロックの最小時間においてローレベルになります。

シーケンス処理

いくつかのデータフレームを単一シーケンスに渡してメッセージを完成させることができます。送信が有効な場合、シーケンスはマスタ側の TXFIFO に何らかのデータが存在する場合に開始し、その間続行します。TXFIFO がエンプティになるまで、マスタによってクロック信号が供給され続け、その後追加のデータを待つことを停止します。

半二重（BIDIMODE=1、BIDIOE=0）または単方向（BIDIMODE=0、RXONLY=1）の受信専用モードでは、SPI が有効化され、受信専用モードがアクティブ化されると、直ちにマスタによってシーケンスが開始されます。マスタによってクロック信号が供給されますが、この信号はマスタが SPI または受信専用モードを無効にするまで停止しません。マスタは、クロック信号が停止するまでデータフレームを受信し続けます。

マスタはあらゆるトランザクションを連続モードで供給できる（SCK 信号は連続的）一方で、データフローおよびその内容をいつでも処理できるスレーブ機能を優先する必要があります。必要に応じて、マスタは通信速度を下げ、より低速のクロックか、または十分な遅延を含む個別のフレーム／データセッションを供給する必要があります。次の 2 点に注意してください。SPI モードのマスタまたはスレーブに対するアンダーフローエラー信号はありません。また、スレーブからのデータは、たとえスレーブがそれらのデータを時間内に適切に準備できない場合でも、常にマスタによってトランザクション処理されます。スレーブが DMA を使用することが特に望ましいのは、データフレームが短く、バスが高速の場合です。

マルチスレーブシステムにおいて通信用のスレーブを 1 つだけ選択するには、各シーケンスを NSS パルス内に収める必要があります。単一のスレーブシステムでは NSS によってスレーブを制御する必要はありませんが、スレーブを各データ転送シーケンスの開始と同期させるために、ここにもパルスを供給することをお勧めします。NSS は、ソフトウェアとハードウェアの両方で管理できます（[セクション 27.5.4：スレーブ選択 \(NSS\) ピンの管理を参照](#)）。

セットされた BSY ビットは、進行中のデータフレームトランザクションを示します。専用のフレームトランザクションが終了すると、RXNE フラグが立てられます。最後のビットは単にサンプリングされ、すべてのデータフレームが RXFIFO に保管されます。

SPI を無効にする手順

SPI を無効にする場合は、本項に記載されている無効化手順に従ってください。この手順は、ペリフェラルクロックが停止し、システムが低電力モードに入る前に行うことが重要です。この場合、進行中のトランザクションが破壊されることがあります。モードによっては、この無効化手順が連続通信を停止させる唯一の方法です。

全二重または送信専用モードでは、マスタは、転送するデータの供給を停止した時点でいかなるトランザクションも終了することができます。この場合、クロックは最後のデータトランザクション後に停止します。奇数のデータフレームがダミーバイトの変換を防ぐためにトランザクション処理された場合、パッキングモードでは特に注意する必要があります（[データパッキング](#)セクションを参照）。これらのモードで SPI を無効化する前に、ユーザは標準的な無効化手順に従う必要があります。フレームトランザクションの進行中または次のデータフレームを TXFIFO に保管しているときにマスタトランスミッタで SPI を無効化した場合、SPI の動作は保証されません。

マスタがいずれかの受信専用モードに設定されている場合、連続クロックを停止する唯一の方法は SPE=0 にしてペリフェラルを無効にすることです。これは、最初のビットのサンプリング時間の間、および最後のビット転送が開始される前における最後のデータフレームトランザクション内の特定の時間枠内で発生する必要があります（予測されるデータフレームの総数を受信し、最後の有効データフレーム後に追加の「ダミー」データの読み出しを防ぐため）。このモードで SPI を無効にするには、特定の手順に従う必要があります。

SPI が無効の場合、受信済みで読み出されていないデータは RXFIFO に保管されたままになり、次に SPI を有効にする際に、新しいシーケンスを開始する前に処理する必要があります。未読のデータを防ぐには、SPI を無効にする際に RXFIFO がエンプティであることを確認してください。これは、正しい無効化手順を使用して、またはペリフェラルリセット専用の固有レジスタを制御してソフトウェアリセットを行い、すべての SPI レジスタを初期化することで確認できます (RCC_APB1RSTR レジスタの SPIIRST ビットを参照)。

標準的な無効化手順は、送信セッションが完全に終わったかどうかをチェックするために、FTLV[1:0] および BSY フラグの状態をプルして行われます。このチェックは、たとえば以下に示すように、進行中のトランザクションの終わりを識別する必要があるような特別な場合にも行うことができます。

- NSS 信号がソフトウェアによって管理されており、マスタはスレーブに適切な NSS パルスの終わりを提供する必要がある場合
- 最後のデータフレームまたは CRC フレームのトランザクションがまだペリフェラルバスで進行している間に、DMA または FIFO からのトランザクションのストリームが完了した場合

正しい無効化手順を以下に示します（受信専用モードが使用されている場合を除く）。

1. FTLVL[1:0] = 00（送信するデータがなくなる）まで待ちます。
2. BSY = 0（最後のデータフレームが処理される）まで待ちます。
3. SPI を無効にします（SPE = 0）。
4. FRLVL[1:0] = 00（受信したデータをすべて読み出す）までデータを読み出します。

受信専用モードの場合の正しい無効化手順を以下に示します。

1. 最後のデータフレームの進行中に特定の時間枠内で SPI を無効（SPE = 0）にすることにより、受信フローへの割り込みを行います。
2. BSY = 0（最後のデータフレームが処理される）まで待ちます。
3. FRLVL[1:0] = 00（受信したデータをすべて読み出す）までデータを読み出します。

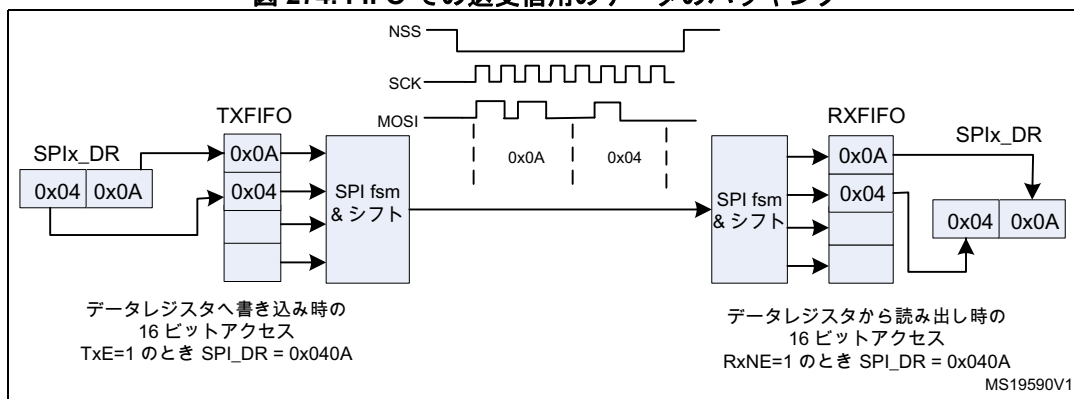
注： *パッキングモードが使用され、8 ビット以下 (1 バイトに適合) のフォーマットの奇数のデータフレームを受信する必要がある場合は、FRLVL[1:0] = 01 のときに FRXTH をセットしなければなりません。これにより、最後の奇数データフレームを読み出し、良好な FIFO ポインタの整列を維持するための RXNE イベントが生成されます。*

データパッキング

データフレームサイズが 1 バイト (8 ビット以下) に適合する場合、SPIx_DR レジスタで 16 ビットの読み出しまたは書き込みアクセスが実行されたときに、自動的にデータパッキングが使用されます。この場合、二重データフレームパターンが並行して処理されます。最初に、SPI はアクセスしたワードの LSB に保管されたパターンを使用して動作します。次に、MSB に保管された残り半分を使用して動作します。図 274 に、データパッキングモードのシーケンス処理の例を示します。2 つのデータフレームは、トランスミッタの SPIx_DR レジスタに単一の 16 ビットがアクセスした後で送信されます。このシーケンスでは、RXFIFO 閾値が 16 ビット (FRXTH=0) にセットされている場合に、レシーバで 1 つの RXNE イベントのみを生成できます。この場合、この単一の RXNE イベントへのレスポンスとして、レシーバは SPIx_DR の単一の 16 ビット読み出しにより、両方のデータフレームにアクセスする必要があります。RXFIFO 閾値設定および以下の読み出しアクセスは、常にレシーバ側に整理させる必要があります。整理されていない場合は、データを失う可能性があります。

これらの奇数の「1 バイトに適合」するデータフレームを処理する必要がある場合、特定の問題が発生します。トランスミッタ側では、SPIx_DR への 8 ビットアクセスを持つ任意の奇数シーケンスの最後のデータフレームを書き込むだけで十分です。レシーバでは、RXNE イベントを生成するために、フレームの奇数シーケンスで受信した最後のデータフレームの Rx_FIFO 閾値レベルを変更する必要があります。

図 274. FIFO での送受信のデータのパッキング



DMA (ダイレクトメモリアクセス) を使用する通信

最高速度で動作し、オーバーランを回避するために必要なデータレジスタの読み出し／書き込み処理を容易にするために、SPI は簡単なリクエスト／確認応答プロトコルを実現する DMA 機能を備えています。

SPIx_CR2 レジスタの TXE または RXNE イネーブルビットをセットすると、DMA アクセスがリクエストされます。TxバッファとRxバッファには、別々のリクエストを発行する必要があります。

- 送信では、TXE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx_DR レジスタに書き込みます。
- 受信では、RXNE が 1 にセットされるたびに DMA リクエストが発行されます。その後、DMA は SPIx_DR レジスタを読み出します。

図 275 から 図 278 までを参照してください。

SPI がデータの送信にのみ使用される場合、SPI Tx DMA チャンネルのみを有効にすることができます。この場合、受信したデータは読み出されないため、OVR フラグがセットされます。SPI がデータの受信にのみ使用される場合、SPI Rx DMA チャンネルのみを有効にすることができます。

送信モードで、DMA がすべての送信データを書き込んだとき (DMA_ISR レジスタの TCIF フラグがセットされます)、BSY フラグを監視することで SPI 通信の完了を確認できます。最後の送信内容の破壊を避けるために、SPI を無効にする前、または STOP モードに入る前にこの操作を行う必要があります。ソフトウェアは、まず FTLVL[1:0] = 00 になるまで待つてから、BSY=0 になるまで待つ必要があります。

DMA を使用して通信を開始する場合、DMA チャンネルの管理によるエラーイベントを防ぐために、以下の手順を順に実行する必要があります。

1. DMA Rx が使用される場合、SPI_CR2 レジスタの RXDMAEN ビットの DMA Rx バッファを有効にします。
2. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを有効にします。
3. DMA Tx が使用される場合、SPI_CR2 レジスタの TXDMAEN ビットにて DMA Tx バッファを有効にします。
4. SPE ビットをセットして、SPI を有効にします。

コード例については、「付録」セクション [A.17.5 : DMA による SPI マスタ設定のコード例](#) と [A.17.6 : DMA による SPI スレーブ設定のコード例](#) を参照してください。

通信を終了するには、以下の手順を順に実行する必要があります。

1. ストリームが使用される場合、DMA レジスタにて Tx および Rx の DMA ストリームを無効にします。
2. SPI 無効化手順に従って SPI を無効にします。
3. DMA Tx および (または) DMA Rx が使用されている場合、SPI_CR2 レジスタの TXDMAEN ビットおよび RXDMAEN ビットをクリアすることにより、DMA Tx バッファおよび Rx バッファを無効にします。

DMA によるパッキング

DMA (SPIx_CR2 レジスタでセットされた TXDMAEN および RXDMAEN) で転送が管理される場合、パッキングモードは SPI TX および SPI RX DMA チャンネルで設定される PSIZE 値によって、自動的に有効/無効になります。DMA チャンネルの PSIZE 値が 16 ビットと等しい場合、SPI データサイズは 8 ビット以下で、パッキングモードは有効になります。その後、DMA は SPIx_DR レジスタへの書き込み動作を管理します。

データパッキングモードを使用する場合で転送データ数が 2 の倍数でない場合は、LDMA_TX/LDMA_RX ビットをセットする必要があります。SPI では、最後の DMA 転送の送受信に 1 つのデータのみを考慮します (詳細については、[データパッキング \(759 ページ\)](#) を参照)。

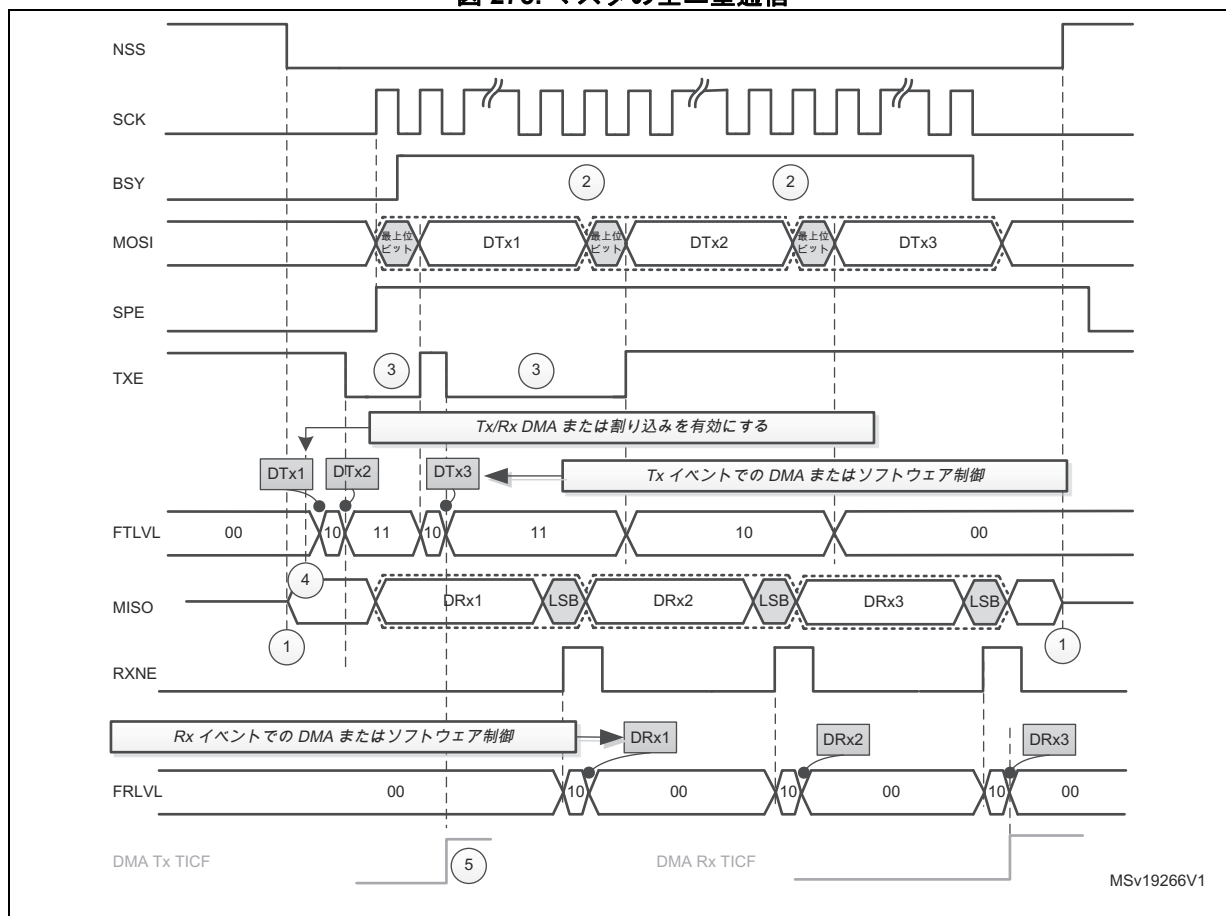
通信図

このセクションでは、一部の標準的なタイミング構成について説明します。これらの構成は、SPI イベントの処理方法（ポーリング、割り込み、または DMA）にかかわらず有効です。単純にするには、ここでは LSBFIRST=0、CPOL=0、および CPHA=1 設定を共通の前提として使用します。DMA ストリームの完全な設定は提供されません。

以下の番号付された注は、[図 275 \(762 ページ\)](#) から [図 278 \(765 ページ\)](#) に共通です。

1. スレープは、NSS がアクティブで SPI が有効になると MISO ラインを制御し始め、そのいずれかがリリースされるとラインから切断されます。事前にマスタ専用のデータを準備するには、トランザクションを開始する前にスレープに十分な時間を与える必要があります。
マスタでは、SPI が有効な場合のみ、SPI ペリフェラルが MOSI および SCK 信号で（また、ときに NSS 信号でも）制御できるようになります。SPI が無効の場合、SPI ペリフェラルは GPIO ロジックから切断され、これによりこれらのラインのレベルは GPIO 設定にのみ依存します。
2. マスタでは、通信（クロック信号）が連続的な場合に、BSY はフレーム間でアクティブなままとなります。スレープでは、BSY 信号は必ずフレーム間で最低でも 1 クロックサイクルは無効になります。
3. TXE 信号は TXFIFO がフルの場合にのみクリアされます。
4. DMA アービトレーションプロセスは TXDMAEN ビットのセット後すぐに開始します。TXE 割り込みは TXEIE のセット後すぐに生成されます。TXE 信号がアクティブレベルにあると、TxFIFO へのデータ転送が開始され、TxFIFO がフルまたは DMA 転送が完了するまで続きます。
5. 送信するデータをすべて TxFIFO に適合することができる場合、SPI バスでの通信が開始される前に DMA Tx TCIF フラグを立てることができます。このフラグは、常に SPI トランザクションが完了する前に立てられます。
6. パッケージの CRC 値は、SPIx_TxCRCR および SPIx_RxCRCR レジスタで、フレームごとに連続的に計算されます。CRC 情報はすべてのデータパッケージが完了した後で処理されます。これは、自動的に DMA で（必ず Tx チャンネルを処理するフレーム数にセットすること）、またはソフトウェアで（必ず最後のデータフレーム処理中に CRCNEXT を処理すること）行われます。SPIx_TxCRCR で計算された CRC 値がトランスミッタによって簡単に送信されるのに対し、受信した CRC 情報は RxFIFO に読み込まれ、SPIx_RxCRCR レジスタの内容と比較されます（差異がある場合は、ここで CRC エラーフラグが立てられます）。このため、ユーザは注意してこの情報を FIFO から一掃する必要があります。これには、RxFIFO のすべての保管された内容を読み出すソフトウェアか、Rx チャンネルの適切なデータフレーム数がプリセットされている場合は DMA を使用します（データフレーム数 + CRC フレーム数）（前提条件の例の設定を参照）。
7. データがパックされたモードで、TxE および RxNE イベントはペアにされ、データフレーム数が偶数になるまで FIFO への各読み出し／書き込みアクセスは 16 ビット幅となります。TxFIFO が $\frac{3}{4}$ フルの場合、FTLVL ステータスは FIFO フルレベルのままです。このため、最後の奇数のデータフレームを TxFIFO が $\frac{1}{2}$ フルになる前に格納することはできません。このフレームは 8 ビットアクセスを持つ TxFIFO に格納されます。これは、ソフトウェアで、または LDMA_TX 制御がセットされている場合は自動的に DMA で行われます。
8. パックされたモードで最後の奇数のデータフレームを受信するには、最後のデータフレームが処理された時に Rx 閾値を 8 ビットに変更する必要があります。これは、ソフトウェアで FRXTH=1 に設定するか、LDMA_RX がセットされている場合は自動的に DMA 内部信号で行われます。

図 275. マスタの全二重通信



マスタの全二重通信の前提条件の例：

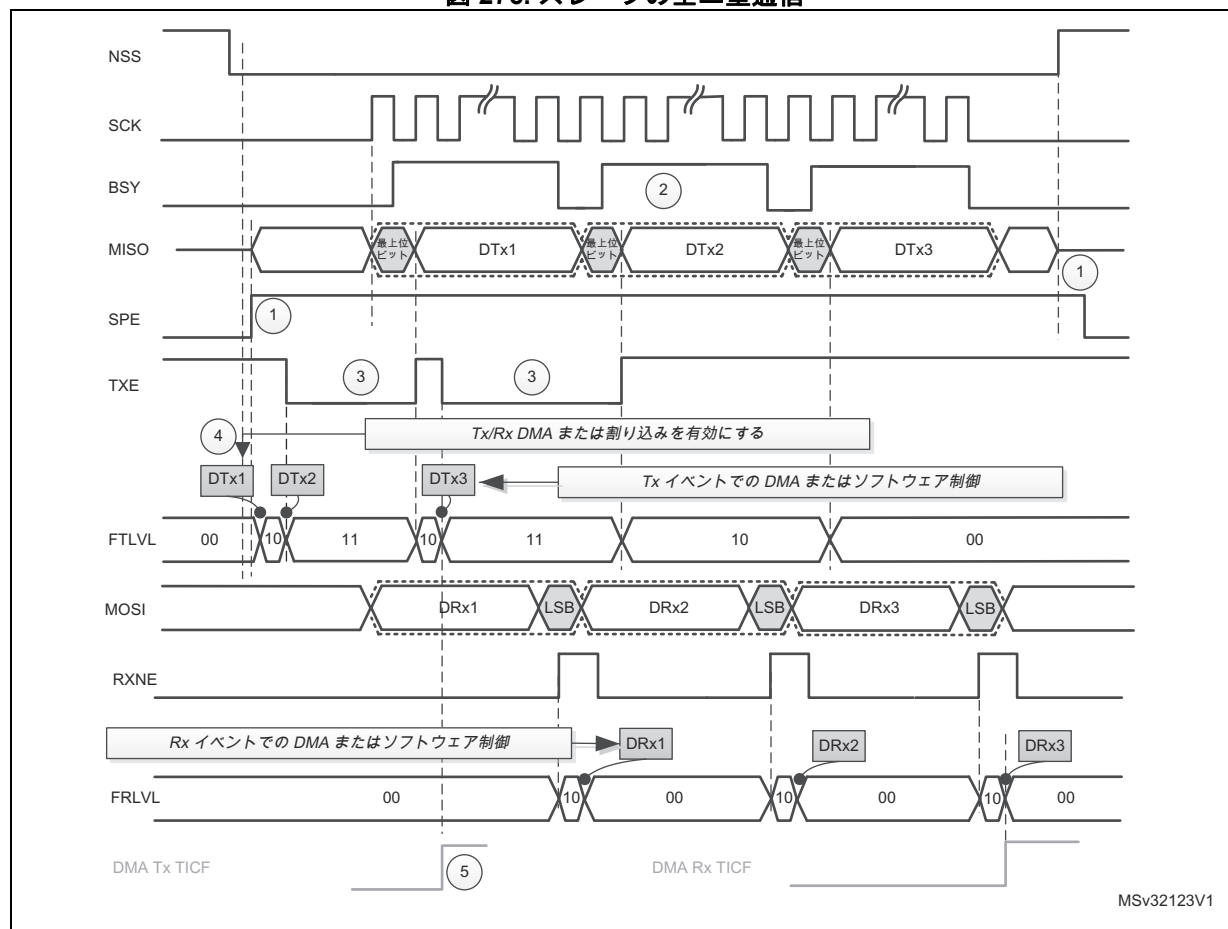
- データサイズ > 8 ビット

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(761 ページ\)](#) を参照してください。

図 276. スレーブの全二重通信



スレーブの全二重通信の前提条件の例：

- データサイズ > 8 ビット

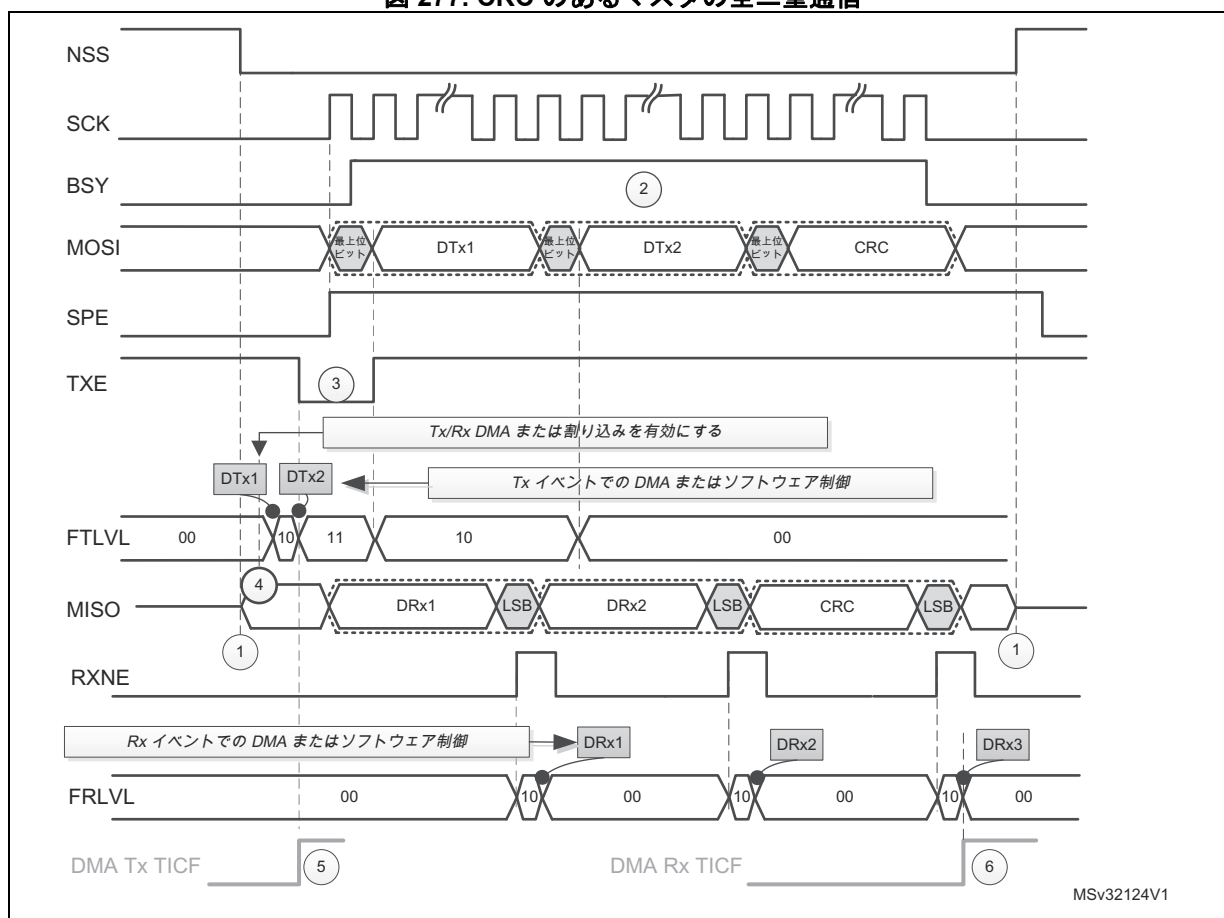
DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(761 ページ\)](#) を参照してください。

コード例については、「付録」セクション [A.17.3 : SPI 全二重通信のコード例](#) を参照してください。

図 277. CRC のあるマスタの全二重通信



CRC のあるマスタの全二重通信の前提条件の例：

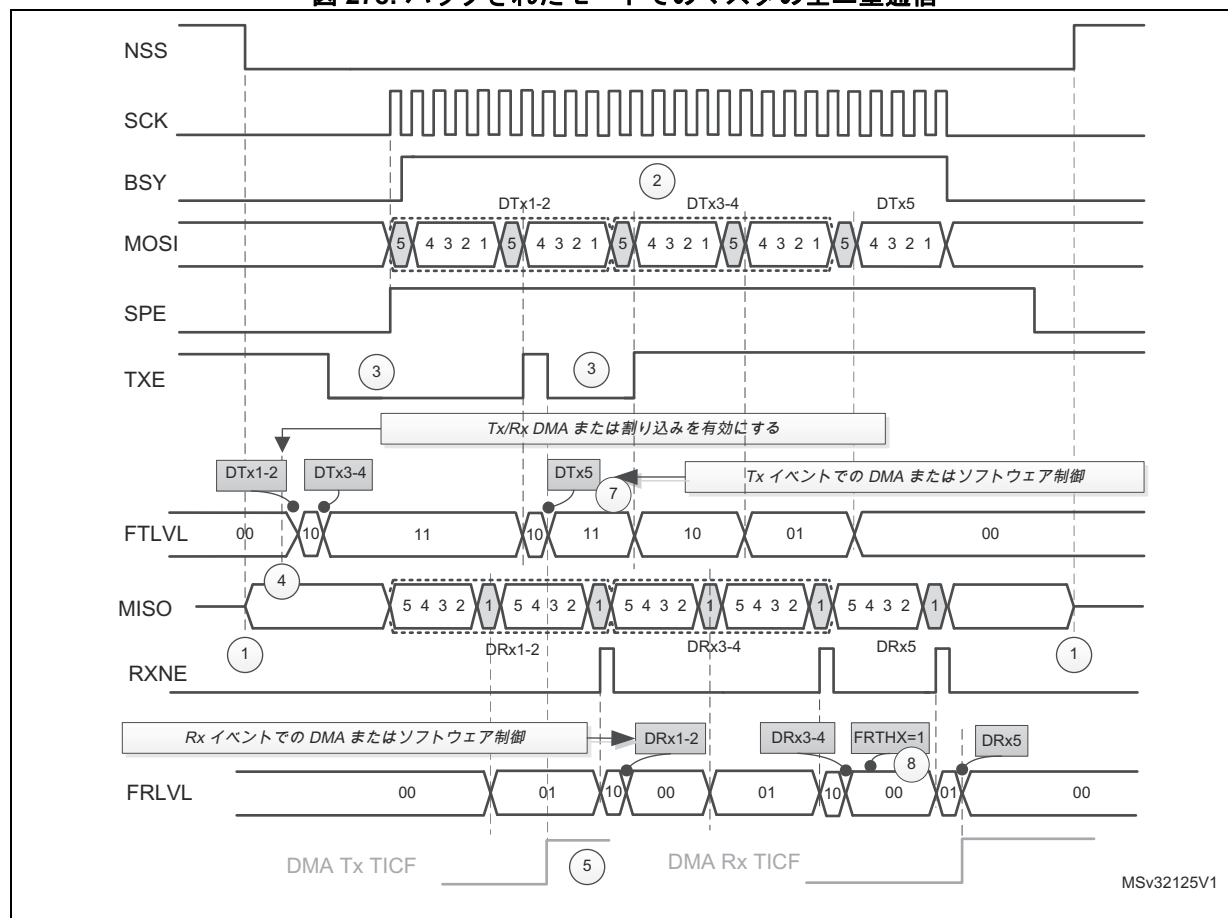
- データサイズ = 16 ビット
- CRC は有効

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 2 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット

共通の前提条件および注の詳細については、[通信図 \(761 ページ\)](#) を参照してください。

図 278. パックされたモードでのマスタの全二重通信



パックされたモードでのマスタの全二重通信の前提条件の例：

- データサイズ = 5 ビット
- 読み出し／書き込み FIFO は、ほとんど 16 ビットアクセスで実行されます。
- FRXTH=0

DMA を使用する場合：

- DMA でトランザクションされる Tx フレーム数を 3 にセット
- DMA でトランザクションされる Rx フレーム数を 3 にセット
- Tx と Rx DMA チャンルの両方の PSIZE を 16 ビットにセット
- LDMA_TX=1 および LDMA_RX=1

共通の前提条件および注の詳細については、[通信図 \(761 ページ\)](#) を参照してください。

27.5.9 SPI ステータスフラグ

アプリケーションが SPI バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

Tx バッファエンプティフラグ (TXE)

TXE フラグは、送信 TXFIFO に送信データを格納するための十分なスペースがある場合にセットされます。TXE フラグは TXFIFO レベルにリンクされます。フラグはハイになり、TXFIFO レベルが FIFO の深さの 1/2 以下になってもハイを維持します。SPIx_CR2 レジスタの TXEIE ビットがセットされている場合は、割り込みを生成できます。このビットは、TXFIFO レベルが 1/2 を超えると自動的にクリアされます。

Rx バッファノットエンプティ (RXNE)

SPIx_CR2 レジスタの FRXTH ビット値によって、RXNE フラグがセットされます。

- FRXTH をセットすると RXNE はハイになり、RXFIFO レベルが 1/4 (8 ビット) 以上になってもハイを維持します。
- FRXTH をクリアすると RXNE はハイになり、RXFIFO レベルが 1/2 (16 ビット) 以上になってもハイを維持します。

SPIx_CR2 レジスタの RXNEIE ビットがセットされている場合は、割り込みを生成できます。

RXNE は、上記の条件が真ではなくなった場合にハードウェアによって自動的にクリアされます。

ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます (このフラグへの書き込みは無効)。

BSY フラグがセットされると、SPI 上でデータ転送が進行中であることを示します (SPI バスはビジー)。

BSY は、転送終了を検出するために特定のモードで使用できます。これにより、低電力モード (ペリフェラルにクロックを供給しない) に入る前に、ソフトウェアで SPI やそのペリフェラルクロックを無効にすることができます。これによって、最後の転送データの破壊を回避します。

BSY フラグは、マルチマスタシステムでの書き込み衝突の回避にも役立ちます。

BSY フラグは次のいずれかの条件下でクリアされます。

- SPI が正常に無効にされたとき
- マスタモードで、障害が検出 (MODF ビットが 1 にセットされます) されたとき
- マスタモードで、データ送信が終了し、送信準備ができていない新しいデータがないとき
- スレーブモードで、各データ転送間で少なくとも SPI の 1 クロックサイクルの間、BSY フラグが 0 にセットされているとき

注： 次の送信がすぐにマスタで処理される場合 (マスタが受信専用モードに設定されているか、その送信 FIFO がエンプティでない場合)、通信は連続的で、BSY フラグはマスタ側での送信の間、“1”にセットされたままになります。スレーブとは異なりますが、必ず (BSY フラグを使用する代わりに) TXE フラグと RXNE フラグを使用して、データの送受信の処理を行うことを推奨します。

27.5.10 SPI エラーフラグ

次のいずれかのエラーフラグがセットされていて、ERRIE ビットをセットすることにより割り込みが有効になっている場合、SPI 割り込みが生成されます。

オーバーランフラグ (OVR)

マスタまたはスレーブでデータを受信し、RXFIFO に受信データを格納するための十分なスペースがない場合に、オーバーラン状態が発生します。これは、ソフトウェアまたは DMA が RXFIFO に格納された前の受信データを読み出すために必要な時間が十分になかったか、データストレージのための空間が制限されている場合に発生する可能性があります。たとえば、RXFIFO は CRC が受信専用モードでのみ有効な場合は使用できないため、この場合、受信バッファは単一データフレームバッファに制限されます（[セクション 27.5.13 : CRC 計算](#)を参照）。

オーバーラン条件が発生すると、RXFIFO にある前の値を新しく受信した値で上書きすることはできません。新しく受信した値は破棄され、それ以降に送信されたすべてのデータは失われます。OVR ビットをクリアするには、SPI_DR レジスタを読み出し、続けて SPI_SR レジスタに読み出しアクセスを行います。

モードフォールト (MODF)

モードフォールトは、マスタデバイスが内部 NSS 信号（NSS ハードウェアモードでは NSS ピン、NSS ソフトウェアモードでは SSI ビット）をローレベルにプルダウンしたときに発生します。これにより、MODF ビットが自動的にセットされます。マスタモードフォールトは、SPI インタフェースに次のような影響を与えます。

- MODF ビットがセットされ、ERRIE ビットがセットされている場合は SPI 割り込みが生成されます。
- SPE ビットがクリアされます。これによって、デバイスからのすべての出力がブロックされ、SPI インタフェースが無効になります。
- MSTR ビットがクリアされ、デバイスは強制的にスレーブモードになります。

MODF ビットをクリアするには、次のソフトウェアシーケンスを実行します。

1. MODF ビットがセットされている間、SPIx_SR レジスタに読み出し／書き込みアクセスを行います。
2. 次に、SPIx_CR1 レジスタに書き込みを行います。

複数の MCU で構成されるシステムでスレーブ間の競合を避けるには、MODF ビットをクリアするシーケンス中、NSS ピンをハイレベルにプルアップする必要があります。このクリアシーケンスの後、SPE ビットと MSTR ビットは、元の状態に戻すことができます。安全のため、MODF ビットがセットされている間、ハードウェアは SPE ビットと MSTR ビットのセットを許可しません。スレーブデバイスでは、MODF ビットはセットできません。ただし、前回のマルチマスタ競合の結果としてセットする場合は例外です。

CRC エラー (CRCERR)

このフラグを使用して、SPIx_CR1 レジスタの CRCEN ビットがセットされているときに受信された値の有効性を検証します。シフトレジスタに受信された値が、レシーバである SPIx_RXCRCR の値と一致しなかった場合、SPIx_SR レジスタの CRCERR フラグがセットされます。フラグはソフトウェアによってクリアされます。

TI モードフレームフォーマットエラー (FRE)

SPI がスレーブモードで動作し、かつ TI モードプロトコルに準拠した設定となっている場合、通信の進行中に NSS パルスが発生すると、TI モードフレームフォーマットエラーが検出されます。このエラーが発生すると、SPIx_SR レジスタの FRE フラグがセットされます。エラー発生時には SPI は無効にされず、この NSS パルスは無視されます。SPI は次の NSS パルスを待ってから新規の転送を開始します。このエラーの検出により 2 バイトのデータが失われるため、データは破壊される可能性があります。

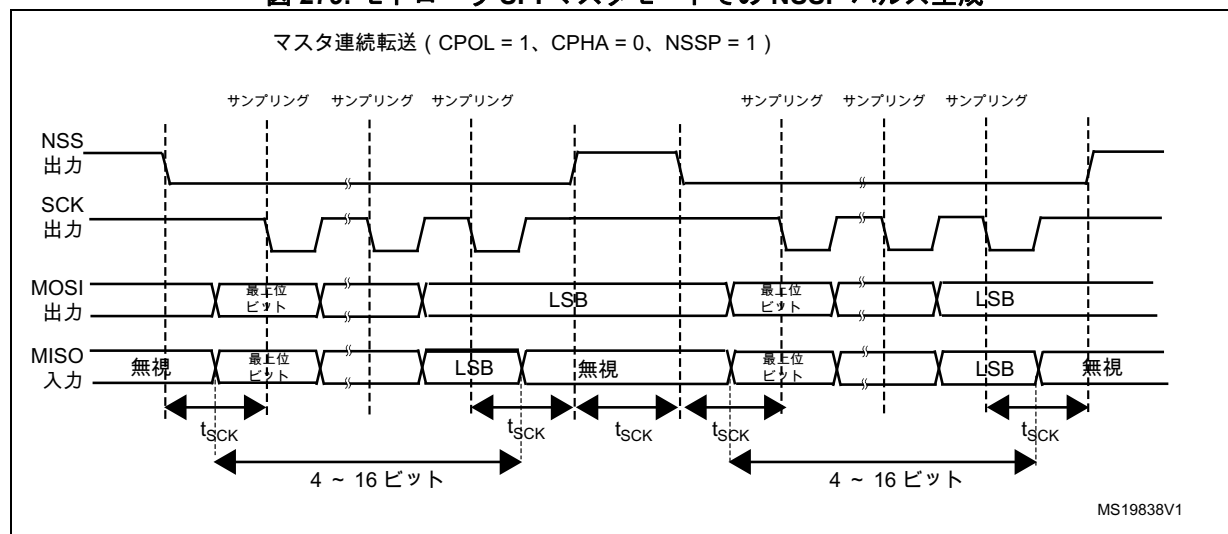
SPIx_SR レジスタを読み出すと、FRE フラグがクリアされます。ERRIE ビットがセットされている場合、NSS エラー検出時に割り込みが生成されます。この場合、データの一貫性が保証されなくなるため、SPI を無効にする必要があり、またスレーブ SPI が再び有効化された場合は、マスタによって通信を再起動する必要があります。

27.5.11 NSS パルスモード

このモードは SPIx_CR2 レジスタの NSSP ビットで有効化され、SPI インタフェースが 1 番目のエッジにキャプチャのある モトローラ SPI マスタ (FRF=0) として設定されている場合にのみ有効になります (SPIx_CR1 CPHA = 0、CPOL 設定は無視されます)。有効にすると、最低 1 つのクロック周期の間、NSS がハイレベルを維持する場合に、2 つの連続したデータフレーム転送の間で NSS パルスが生成されます。このモードでは、スレーブでデータをラッチできます。NSSP パルスモードは、シングルマスタ-スレーブペアのアプリケーション用に設計されています。

図 279 に、NSSP パルスモードが有効な場合の NSS ピンの管理の図を示します。

図 279. モトローラ SPI マスタモードでの NSSP パルス生成



注: CPOL = 0 の場合に同様の動作が発生します。この場合、サンプリングエッジは SCK の立ち上がりエッジで、NSS のアサートおよびネゲートはこのサンプリングエッジを参照します。

27.5.12 TI モード

マスタモードでの TI プロトコル

SPI インタフェースは TI プロトコルと互換性があります。SPIx_CR2 レジスタの FRF ビットを使って、SPI をこのプロトコルに準拠させるように設定することができます。

SPIx_CR1 レジスタにセットされる値によらず、クロックの極性と位相は TI プロトコル要件に必ず適合します。NSS 管理も TI プロトコルに固有なものになります。これにより、この場合の SPIx_CR1 レジスタと SPIx_CR2 レジスタによる NSS 管理の設定 (SSM、SSI、SSOE) ができなくなります。

スレーブモードでは、SPI ポーレートプリスケラを使用して、現在のトランザクションが終了した時点で、MISO ピンの状態がハイインピーダンスに変化するタイミングを制御します (図 280 を参照)。任意のポーレートが使用できるため、このタイミングを非常に柔軟に決定することができます。ただし、ポーレートは外部マスタクロックポーレートに設定されるのが一般的です。MISO 信号がハイインピーダンス (t_{release}) になるまでの遅延は、内部再同期と SPIx_CR1 レジスタの BR[2:0] ビットで設定されたポーレート値によって変わります。この値は次式で求められます：

$$\frac{t_{\text{baud\_rate}}}{2} + 4 \times t_{\text{pclk}} < t_{\text{release}} < \frac{t_{\text{baud\_rate}}}{2} + 6 \times t_{\text{pclk}}$$

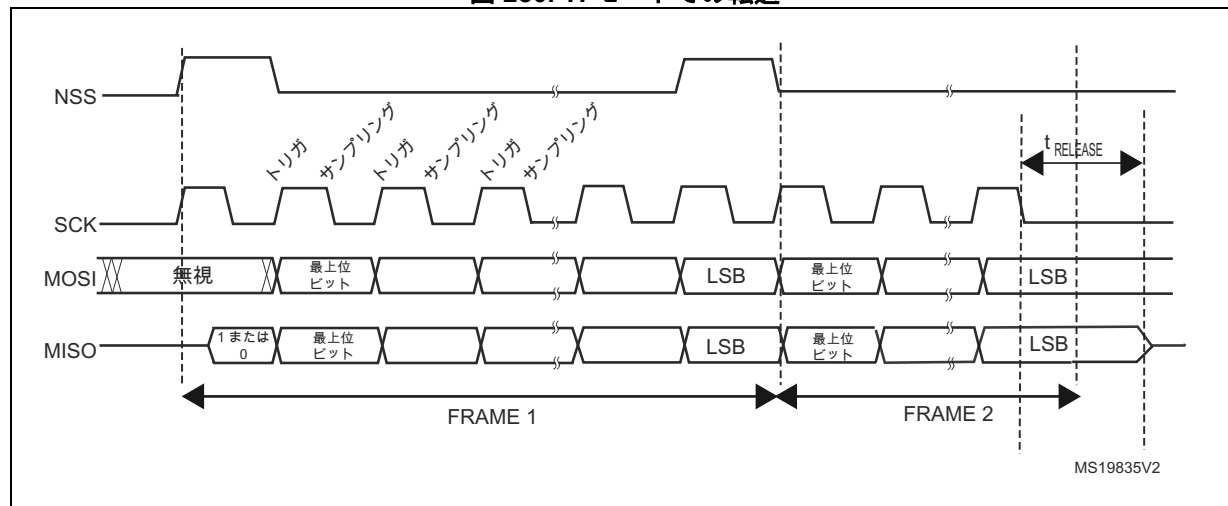
スレーブがデータフレームトランザクション中に NSS パルスの誤配置を検出すると、TIFRE フラグがセットされます。

データサイズが 4 ビットまたは 5 ビットである場合、全二重モードまたは送信専用モードのマスタは、最下位ビット後に追加された 1 つまたは複数のダミーデータビットを持つプロトコルを使用します。TI NSS パルスは、各周期の最下位ビットではなく、このダミービットクロックサイクル上に生成されます。

この機能はモトローラの SPI 通信には使用できません (FRF ビットを 0 に設定)。

図 280 : TI モードでの転送 TI モードが選択されているときの SPI の通信波形を示します。

図 280. TI モードでの転送



27.5.13 CRC 計算

2 つの CRC 計算機が、送信データおよび受信データの信頼性をチェックするために実装されています。SPI は、フレームデータ長とは別に、8 ビットまたは 16 ビットに固定される CRC8 または CRC16 を計算します。その他のデータフレーム長では、CRC は使用できません。

CRC の原理

CRC 計算は、SPI が有効 (SPE = 1) になる前に、SPIx_CR1 レジスタの CRCEN ビットをセットすることによって有効になります。CRC 値は、各ビットに対して奇数のプログラム可能な多項式の値を使用して計算されます。計算は、SPIx_CR1 レジスタの CPHA ビットと CPOL ビットによって定義されるサンプリングクロックエッジで行われます。計算された CRC 値は、データブロックの最後のみならず、CPU または DMA によって管理される転送に関しても、自動的にチェックされます。受信データをもとに内部で計算された CRC 値とトランスミッタが送信した CRC 値の間に不一致が検出された場合、データ破壊エラーを示すために CRCERR フラグがセットされます。CRC 計算を処理する正しい手順は、SPI の設定および選択された転送管理によって変わります。

注： *多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。*

CPU によって管理される CRC 転送

通信が開始され、最後のデータフレームが SPIx_DR レジスタで送信または受信されるまで正常に続きます。次に、CRC フレームトランザクションが現在処理中のデータフレームトランザクションの後に行われることを示すために、SPIx_CR1 レジスタの CRCNEXT ビットをセットする必要があります。CRCNEXT ビットは、最後のデータフレームトランザクションの終了前にセットしてください。CRC のトランザクション中は CRC 計算は行われません。

受信した CRC は、データバイトまたはワードのように RXFIFO に格納されます。このため、CRC モードでは、受信バッファを一度に 1 データフレームずつ受信するために使用する単一の 16 ビットバッファとみなす必要があります。

CRC フォーマットトランザクションは、通常データシーケンスの最後に通信を行うためにデータフレームをもう 1 つ受け取ります。ただし、16 ビット CRC によってチェックされた 8 ビットデータフレームを設定する場合、CRC 全体を送信するにはフレームがあと 2 つ必要です。

最後の CRC データを受信すると、受信した値と SPIx_RXCRC レジスタの値を比較する自動チェックが行われます。ソフトウェアは、SPIx_SR レジスタの CRCERR フラグをチェックして、データ転送の内容が破壊されているか否かを判断する必要があります。ソフトウェアは、CRCERR フラグに "0" を書き込んでクリアします。

CRC 受信後、CRC 値は RXFIFO に格納され、RXNE フラグをクリアするために SPIx_DR レジスタを読み出す必要があります。

DMA によって管理される CRC 転送

SPI 通信が CRC 通信と DMA モードで有効化される場合、CRC の送受信は通信の最後に自動で行われます (ただし、受信専用モードで CRC データを読み出す場合を除く)。CRCNEXT ビットはソフトウェアで処理する必要はありません。SPI 送信用 DMA チャンネルのカウンタは、転送するデータフレーム数から CRC フレームを除いてセットする必要があります。レシーバ側では、受信した CRC 値はトランザクションの終了時に DMA によって自動的に処理されますが、RXFIFO から受信した CRC 情報は常にレシーバにロードされますので、ユーザは注意してそれを一掃してください。全二重モードでは、受信 DMA チャンネルのカウンタを、受信するデータフレームの数 (CRC を含む) にセットすることができます。つまり、たとえば 16 ビット CRC でチェックされた 8 ビットデータフレームの特定の例を、次の式で表すことができます。

$$\text{DMA\_RX} = \text{Numb\_of\_data} + 2$$

受信専用モードでは、DMA 受信チャネルカウンタには転送されたデータ量のみが含まれます (CRC 計算を除く)。次に、DMA からの全体の転送に基づき、すべての CRC 値はこのモードでは単一のバッファとして動作するため、ソフトウェアによって FIFO から読み戻す必要があります。

転送中に内容の破壊が生じた場合、データと CRC の転送の最後に SPIx_SR レジスタの CRCERR フラグがセットされます。

パッキングモードが使用され、データ数が奇数の場合は、LDMA_RX ビットを管理する必要があります。

SPIx_TXCRC および SPIx_RXCRC の値のリセット

CRC フェーズのあとで新しいデータがサンプリングされた場合、SPIx_TXCRC 値および SPIx_RXCRC 値は自動的にクリアされます。これにより、データを中断することなく転送するために、DMA サーキュラモードが使用可能になります (受信専用モードでは使用不可)。いくつかのデータブロックは中間の CRC チェックフェーズによってカバーされます。

通信中に SPI が無効化された場合は、次のシーケンスに従う必要があります。

- 1. SPI を無効にします。
- 2. CRCEN ビットをクリアします。
- 3. CRCEN ビットを有効にします。
- 4. SPI を有効にします。

注：SPI がスレーブモードのとき、SPE ビットの値に関係なく、CRCEN がセットされるとすぐに、CRC 計算機は SCK スレーブ入カクロックの影響を受けます。CRC 計算の間違いを回避するために、ソフトウェアは、クロックが安定している (定常状態にある) ときのみ、CRC 計算を有効にする必要があります。SPI インタフェースがスレーブとして設定されている場合、データフェーズと CRC フェーズの間で NSS 内部信号をローレベルに維持する必要があります。

27.6 SPI 割り込み

SPI 通信中、割り込みは次のイベントによって生成できます。

- ロードする準備ができていない TXFIFO の送信
- 受信 RXFIFO で受信したデータ
- マスタモードフォールト
- オーバーランエラー
- TI フレームフォーマットエラー
- CRC プロトコルエラー

割り込みは個別に有効または無効にできます。

表 104. SPI 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
ロードする準備ができていない TXFIFO の送信	TXE	TXEIE
RXFIFO で受信したデータ	RXNE	RXNEIE
マスタモードフォールトイベント	MODF	ERRIE
オーバーランエラー	OVR	
TI フレームフォーマットエラー	FRE	
CRC プロトコルエラー	CRCERR	

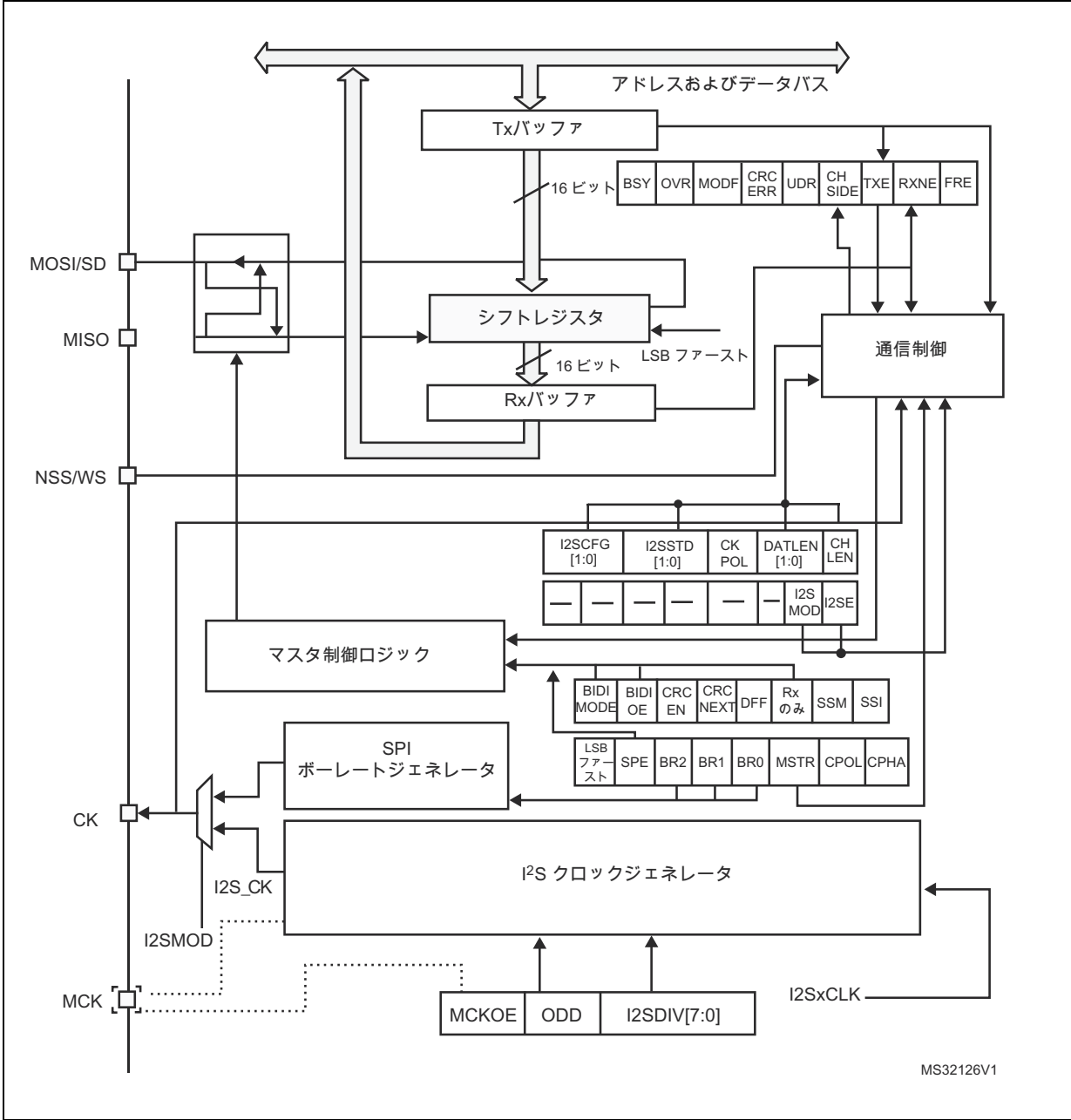
コード例については、「付録」セクション [A.17.4 : SPI 割り込みのコード例](#) を参照してください。

27.7 I²S の機能説明

27.7.1 I²S の概要

I²S のブロック図を [図 281](#)に示します。

図 281. I²S ブロック図



1. MCK は MISO ピンに配置されます。

SPI はオーディオ I²S インタフェースとして機能することができます。それには、SPIx_I2SCFGR レジスタの I2SMOD ビットをセットして I²S 機能を有効にしてください。このインタフェースは、主に SPI と同じピン、フラグ、および割り込みを使用します。

I²S と SPI は、以下の3 つのピンを共用します。

- SD (シリアルデータ) : MOSI ピンに配置され、2 つの時間多重化データチャネルを送受信します (半二重モードのみ)。
- WS (ワードセレクト) : NSS ピンに配置され、マスタモードではデータ制御信号の出力、スレーブモードでは入力です。
- CK : SCK ピンに配置され、マスタモードではシリアルクロック出力、スレーブモードではシリアルクロック入力です。

外部オーディオデバイスにマスタクロック出力が必要な場合、追加のピンを使用できます。

- MCK : 別々に配置され、I²S がマスタモードに設定されている (かつ、SPIx_I2SPR レジスタの MCKOE ビットがセットされている) とき、 $256 \times f_s$ と等しい (f_s はオーディオサンプリング周波数) 設定済みの周波数で生成されたこの追加クロックを出力するために使用されます。

I²S は、マスタモードに設定されているとき、専用のクロックジェネレータを使用して通信クロックを生成します。このクロックジェネレータは、マスタクロック出力のソースでもあります。I²S モードでは、2 つの追加レジスタを使用できます。1 つはクロックジェネレータ設定 SPIx_I2SPR にリンクされ、もう 1 つは汎用 I²S 設定レジスタ SPIx_I2SCFGR (オーディオ規格、スレーブ/マスタモード、データフォーマット、パケットフレーム、クロック極性など) です。

SPIx_CR1 レジスタとすべての CRC レジスタは、I²S モードでは使用されません。同様に、SPIx_CR2 レジスタの SSOE ビットと SPIx_SR レジスタの MODF および CRCERR ビットも使用されません。

I²S は、16 ビット幅モードでのデータ転送に同じ SPI レジスタ (SPIx_DR) を使用します。

27.7.2 サポートされるオーディオプロトコル

3 線バスでは、一般に 2 つのチャネル (右チャネルと左チャネル) で時間多重化されたオーディオデータのみを処理する必要があります。しかしながら、送受信用には 1 つの 16 ビットレジスタしかありません。したがって、各チャネルサイドに対応する適切な値をデータレジスタに書き込んだり、データレジスタからデータを読み出して SPIx_SR レジスタの CHSIDE ビットをチェックして対応するチャネルを識別したりすることは、ソフトウェアの責任です。左チャネルは常に最初に送信され、その後で右チャネルが送信されます (CHSIDE は PCM プロトコルには無関係です)。

4 つのデータおよびパケットフレームを使用できます。データは次のフォーマットで送信されます。

- 16 ビットフレームにパックされた 16 ビットデータ
- 16 ビットフレームにパックされた 32 ビットデータ
- 24 ビットフレームにパックされた 32 ビットデータ
- 32 ビットフレームにパックされた 32 ビットデータ

32 ビットパケットに拡張された 16 ビットデータを使用するとき、最初の 16 ビット (MSB) は上位ビットであり、LSB の 16 ビットは、ソフトウェア動作や DMA リクエストを必要とせずに (1 回の読み込み/書き込み動作のみで) 強制的に 0 にされます。

24 ビットと 32 ビットのデータフレームは、SPIx_DR レジスタとの間で 2 回の CPU 読み出しまたは書き込み動作 (あるいはアプリケーションにとって DMA が望ましい場合は 2 回の DMA 動作) を必要とします。特に 24 ビットのデータフレームの場合、8 つの下位ビットはハードウェアによって 0 ビットで 32 ビットに拡張されます。

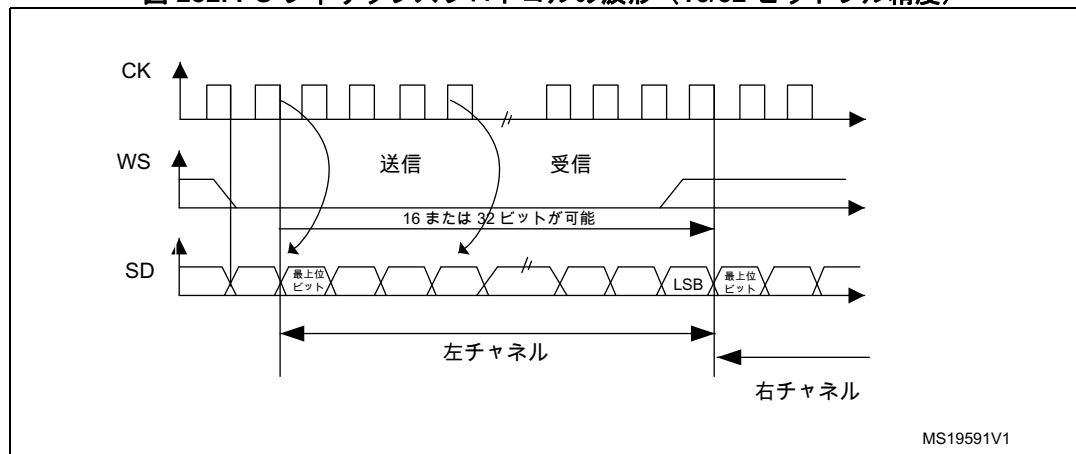
すべてのデータフォーマットと通信規格に対して、最上位ビットは常に最初に送信されます (MSB ファースト)。

I²S インタフェースは、SPIx_I2SCFGR レジスタの I2SSSTD[1:0] と PCMSYNC ビットを使用して設定可能な 4 つのオーディオ規格をサポートします。

フィリップス I²S 規格

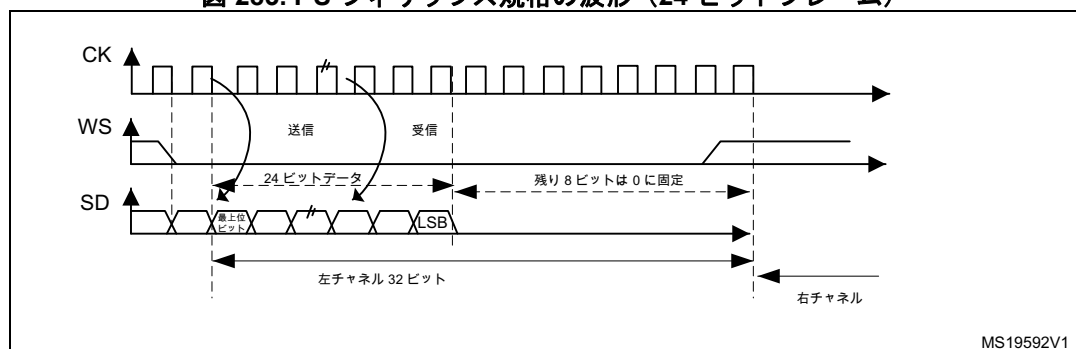
この規格では、どのチャンネルが送信されているかを示すために WS 信号を使用します。この信号が有効になってから 1 CK クロックサイクル後に最初のビット (MSB) が使用可能になります。

図 282. I²S フィリップスプロトコルの波形 (16/32 ビットフル精度)



データは、CK の立ち下がりエッジでラッチされ (トランスミッタの場合)、立ち上がりエッジで読み出されます (レシーバの場合)。WS 信号も CK の立ち下がりエッジでラッチされます。

図 283. I²S フィリップス規格の波形 (24 ビットフレーム)



このモードでは、SPIx_DR レジスタに対して 2 回の書き込みまたは読み出し動作が必要です。

- 送信モードの場合

0x8EAA33 を送信する必要がある場合 (24 ビット) :

図 284. 0x8EAA33 の送信



- 受信モードの場合
データ 0x8EAA33 が受信される場合：

図 285. 0x8EAA33 の受信

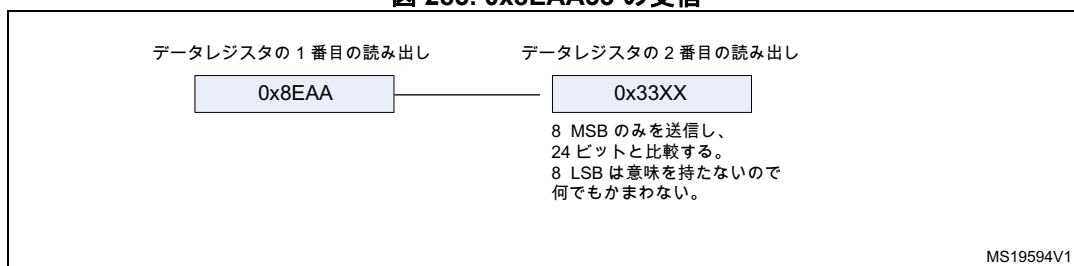
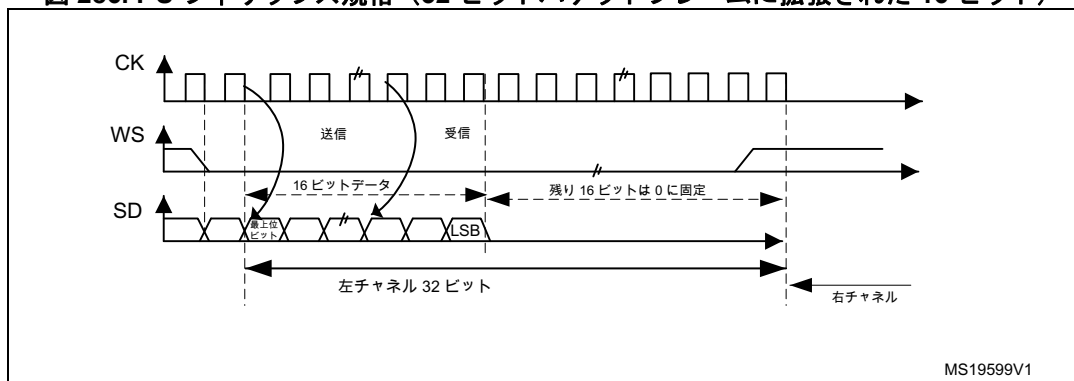


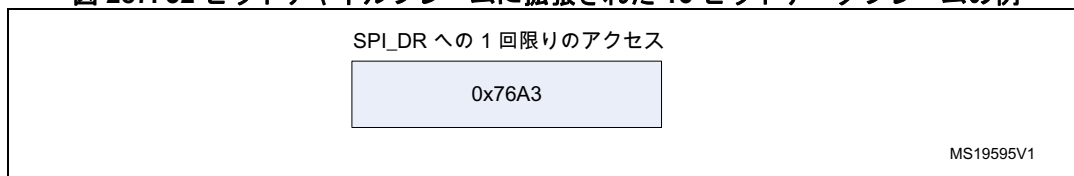
図 286. I²S フィリップス規格 (32 ビットパケットフレームに拡張された 16 ビット)



I²S 設定フェーズで、32 ビットチャネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x76A30000) であるとき、[図 287](#) に示す動作が要求されます。

図 287. 32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例



送信では、SPIx_DR に MSB が書き込まれるたびに、TXE フラグがセットされ、可能ならば、SPIx_DR レジスタに新しい送信値をロードするために割り込みが生成されます。これはハードウェアによって行われるため、0x0000 がまだ送信されていなくても行われます。

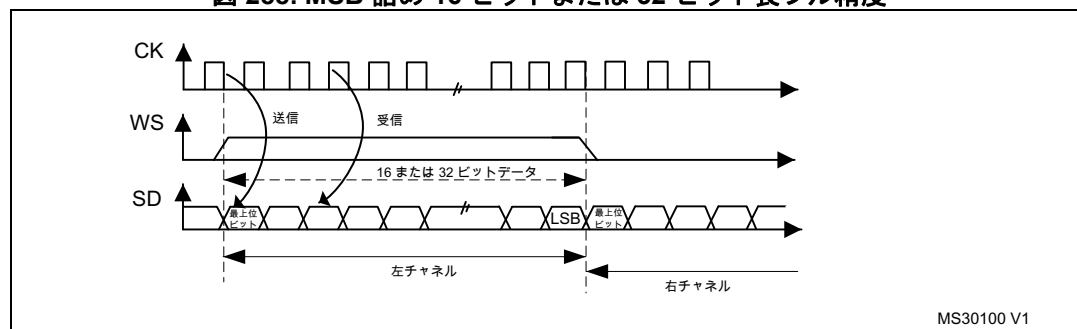
受信では、RXNE フラグがセットされ、可能ならば、最初の 16 MSB ハーフワードの受信時にその割り込みが生成されます。

このように、2 回の書き込みまたは読み出し動作の間にはより多くの時間が設けられるため、アンダーランまたはオーバーラン状態（データ転送の方向に依存）を避けられます。

MSB 詰め規格

この規格では、WS 信号は最初のデータビット（最上位ビット）と同時に生成されます。

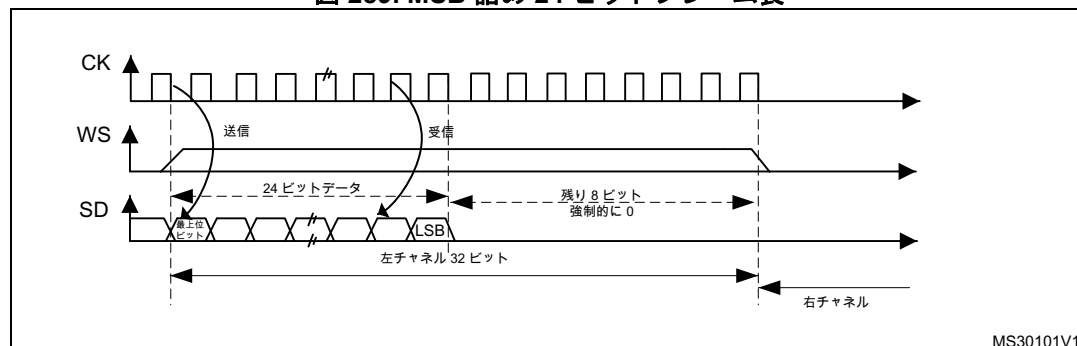
図 288. MSB 詰め 16 ビットまたは 32 ビット長フル精度



MS30100 V1

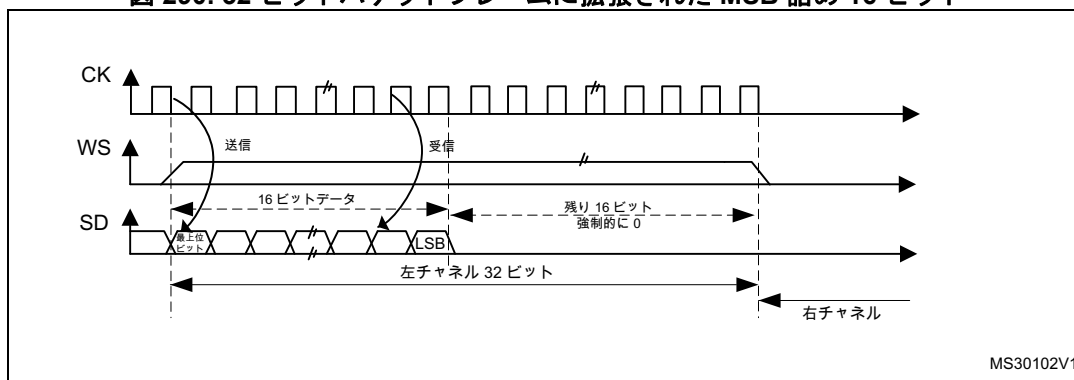
データは、CK の立ち下がリエッジでラッチされ（トランスミッタの場合）、立ち上がりエッジで読み出されます（レシーバの場合）。

図 289. MSB 詰め 24 ビットフレーム長



MS30101V1

図 290. 32 ビットパケットフレームに拡張された MSB 詰め 16 ビット



LSB 詰め規格

この規格は、MSB 詰め規格と似ています（16 ビットと 32 ビットのフル精度フレームフォーマットに違いはありません）。

入出力信号のサンプリングは I2S フィリップス規格と同じです。

図 291. LSB 詰め 16 ビットまたは 32 ビットフル精度

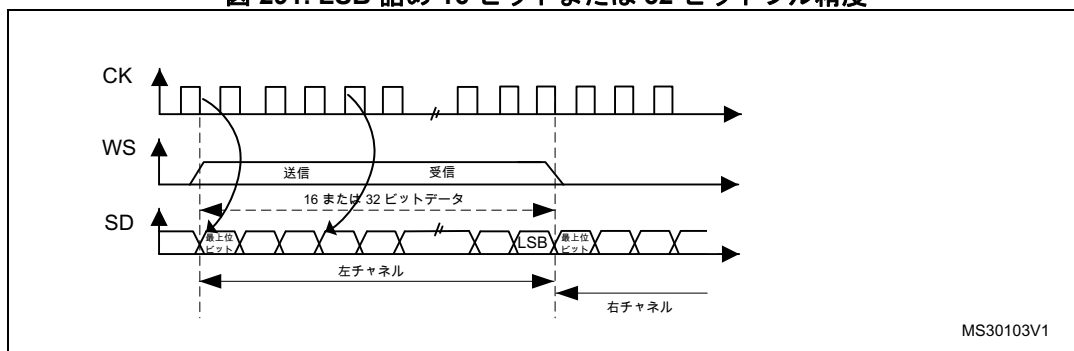
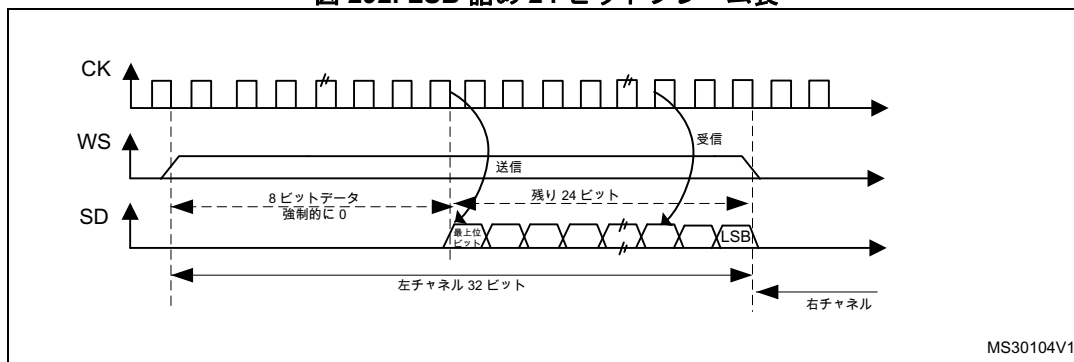


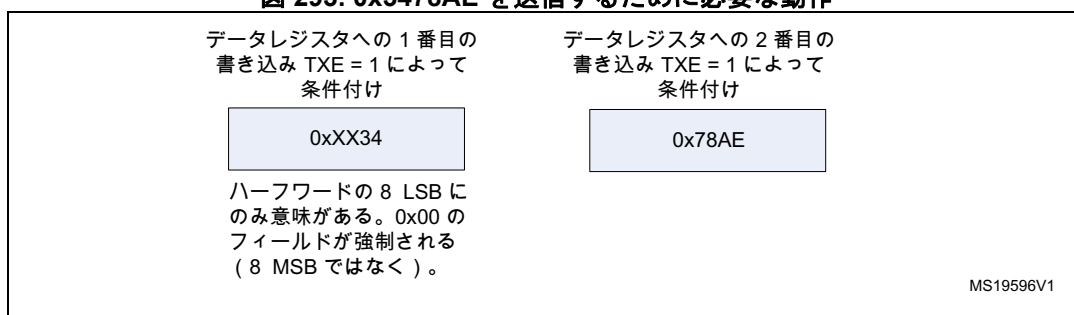
図 292. LSB 詰め 24 ビットフレーム長



● 送信モードの場合

データ 0x3478AE を送信する必要がある場合、ソフトウェアまたは DMA によって SPIx_DR レジスタへの 2 回の書き込み動作が必要です。この動作を次に示します。

図 293. 0x3478AE を送信するために必要な動作



- 受信モードの場合
データ 0x3478AE が受信される場合、RXNE イベントごとに、SPIx_DR レジスタから連続する 2 回の読み出し動作が必要です。

図 294. 0x3478AE の受信に必要な動作

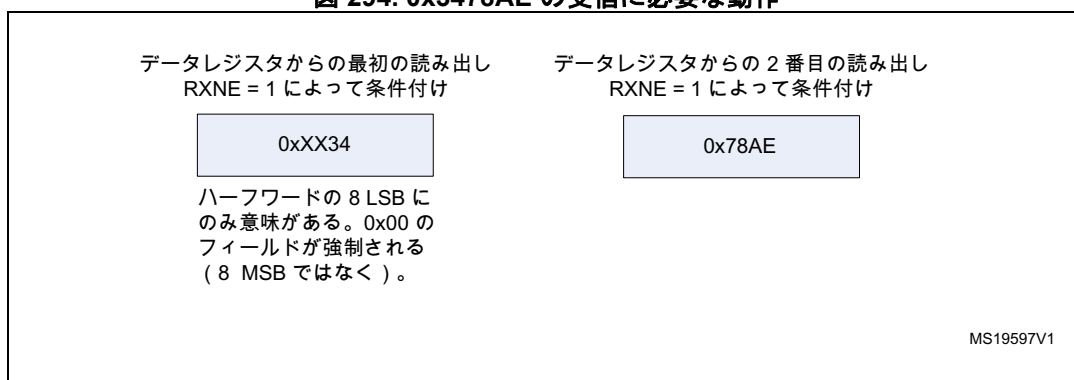
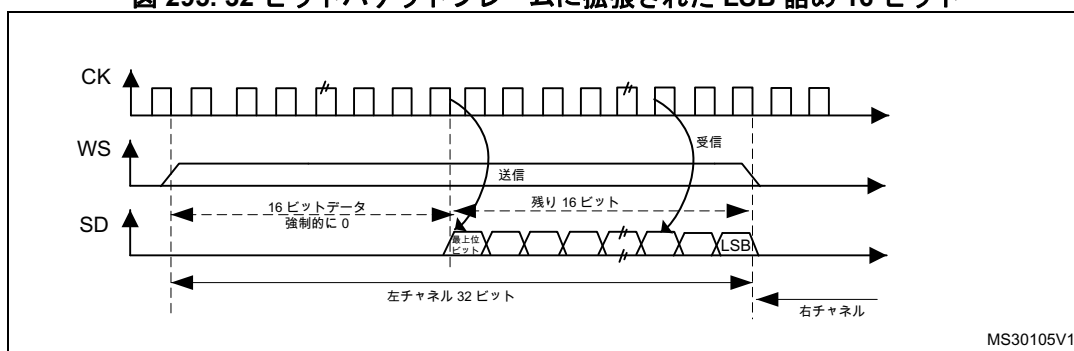


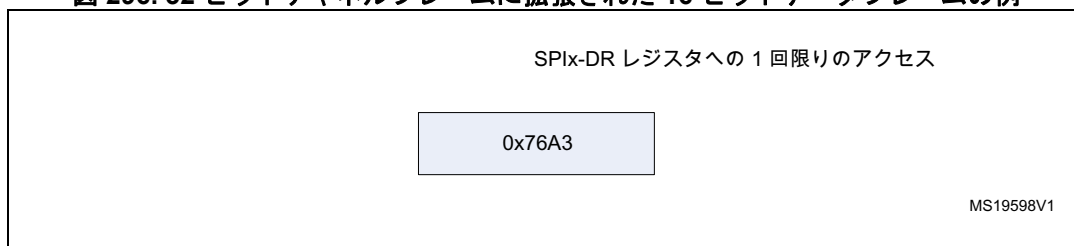
図 295. 32 ビットパケットフレームに拡張された LSB 詰め 16 ビット



I²S 設定フェーズで、32 ビットチャンネルフレームに拡張された 16 ビットデータフレームが選択されたとき、SPIx_DR レジスタへのアクセスは 1 回のみ必要です。残りの 16 ビットは、データを 32 ビットフォーマットに拡張するために、ハードウェアによって強制的に 0x0000 にされます。この場合、それはハーフワード MSB に相当します。

送信するデータまたは受信したデータが 0x76A3 (32 ビットに拡張された 0x0000 76A3) であるとき、[図 296](#) に示す動作が要求されます。

図 296. 32 ビットチャネルフレームに拡張された 16 ビットデータフレームの例



送信モードでは、TXE イベントが発生すると、アプリケーションは送信するデータ（この場合は 0x76A3）を書き込む必要があります。0x000 フィールドが最初に送信されます（32 ビット拡張部）。有効なデータ（0x76A3）が SD に送信されると、すぐに TXE フラグが再びセットされます。

受信モードでは、0x0000 フィールドではない有効なハーフワードが受信されると、すぐに RXNE がアサートされます。

このように、アンダーランやオーバーランの状態を防ぐために、2 回の書き込みまたは読み出し動作の間には、より多くの時間が設けられています。

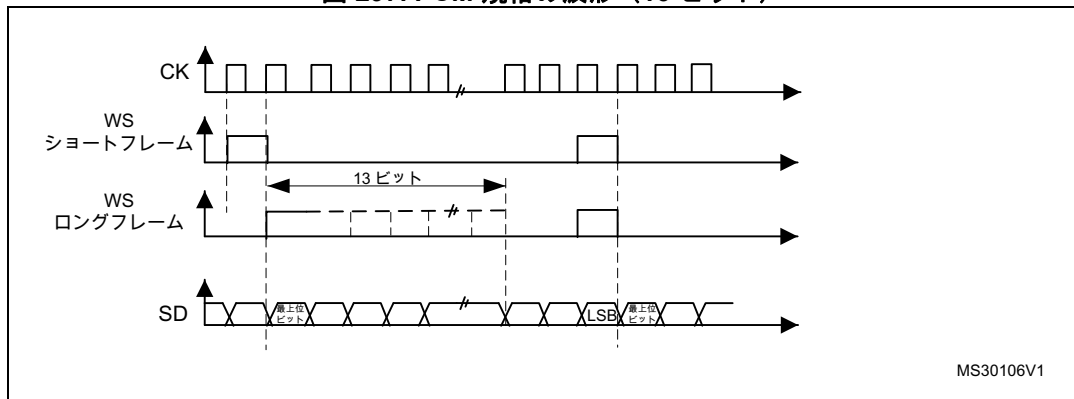
PCM 規格

PCM 規格では、チャネルサイド情報を使用する必要はありません。SPIx_I2SCFGR レジスタの PCMSYNC ビットを使用して、2 つの PCM モード（ショートおよびロングフレーム）の使用および設定が可能です。

PCM モードで、出力信号（WS、SD）は CK 信号の立ち上がりエッジでサンプリングされます。入力信号（WS、SD）は立ち下がりエッジでキャプチャされます。

CK および WS は、マスタモードで出力として設定されることに注意してください。

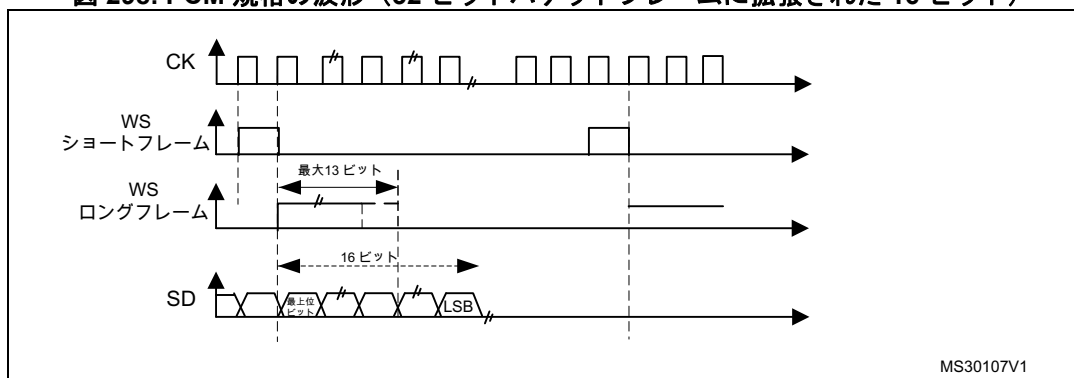
図 297. PCM 規格の波形（16 ビット）



ロングフレーム同期では、WS 信号のアサクション時間はマスタモードで 13 ビットに固定されています。

ショートフレーム同期では、WS 同期信号の長さは、わずか 1 サイクルです。

図 298. PCM 規格の波形 (32 ビットパケットフレームに拡張された 16 ビット)

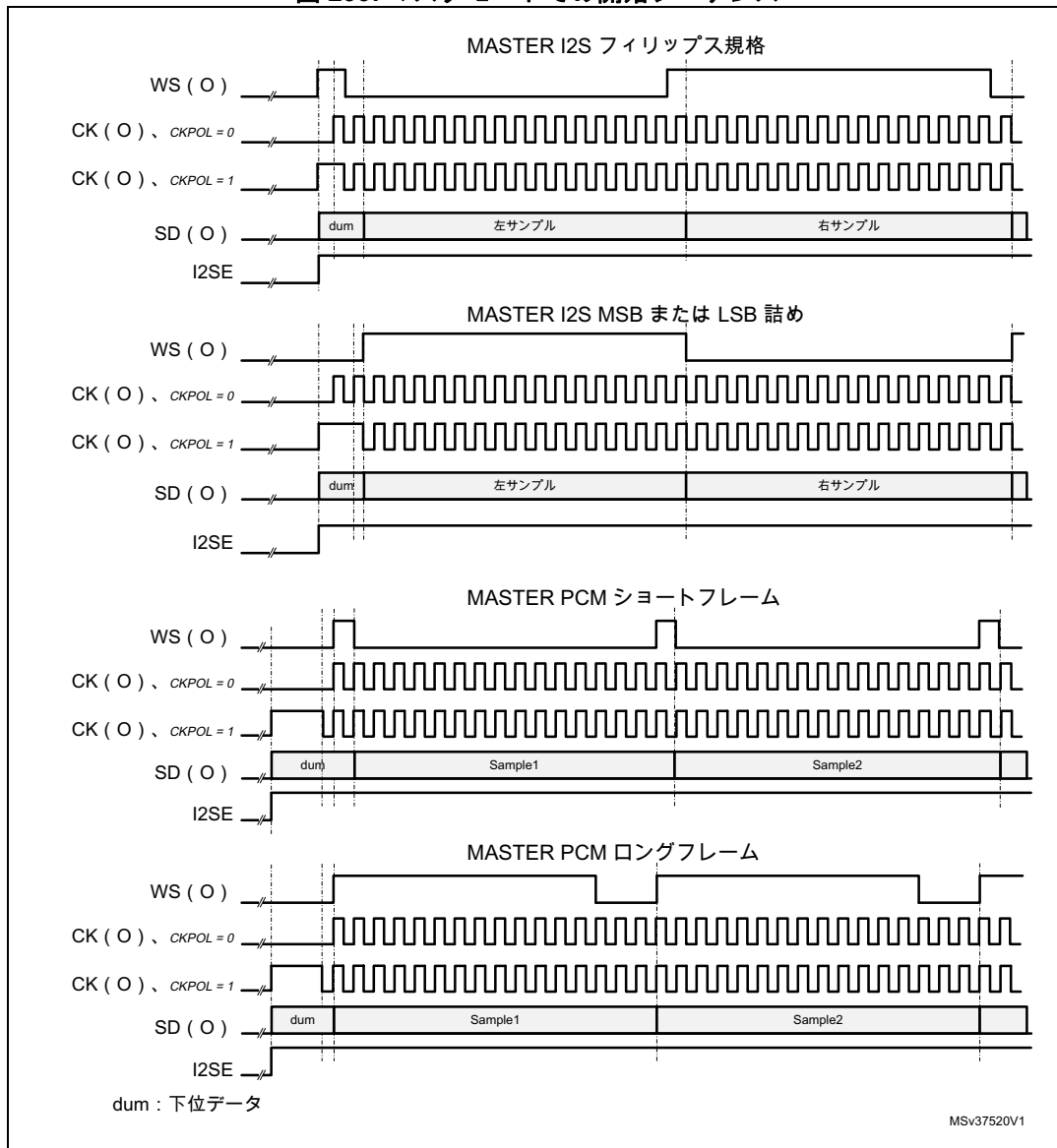


注： 2 つのモード (マスタとスレーブ) と 2 つの同期 (ショートとロング) に関しては、スレーブモードでも、連続した 2 つのデータ (したがって 2 つの同期信号) 間のビット数を (SPIx_I2SCFGR レジスタの DATLEN および CHLEN ビットで) 指定する必要があります。

27.7.3 起動に関する説明

図 299 は、SPI/I2S が有効化された場合 (I2SE ビット) の MASTER モードでのシリアルインタフェースの処理について示しています。生成された信号での CKPOL の影響についても示します。

図 299. マスタモードでの開始シーケンス



スレーブモードの場合、ユーザは WS がアクティブになる前にオーディオインタフェースを有効にする必要があります。これは、I2S フィリップス規格が WS = 1、またはその他の規格が WS = 0 の場合に、I2SE ビットを 1 にセットする必要があることを意味しています。

27.7.4 クロックジェネレータ

I²S ビットレートにより、I²S データライン上のデータフローと I²S クロック信号周波数が決まります。

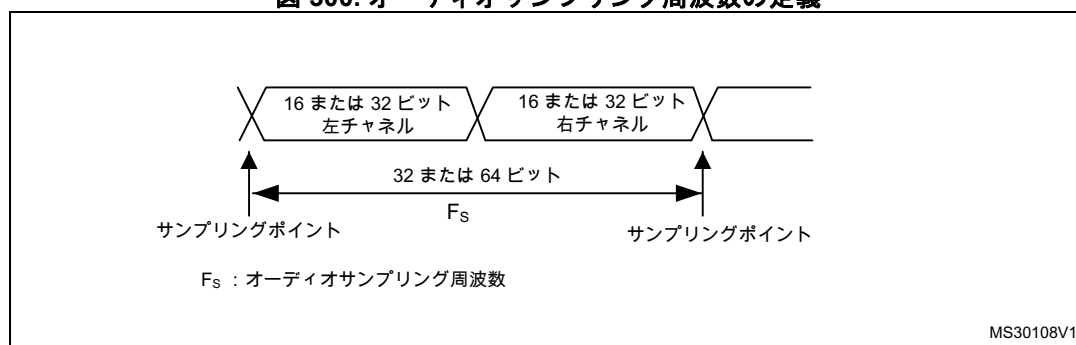
I²S ビットレート = チャンネルあたりのビット数 × チャンネル数 × オーディオサンプリング周波数

16 ビットオーディオ、左チャンネルおよび右チャンネルの場合、I²S ビットレートは次のように算出されます。

$$I^2S \text{ ビットレート} = 16 \times 2 \times f_s$$

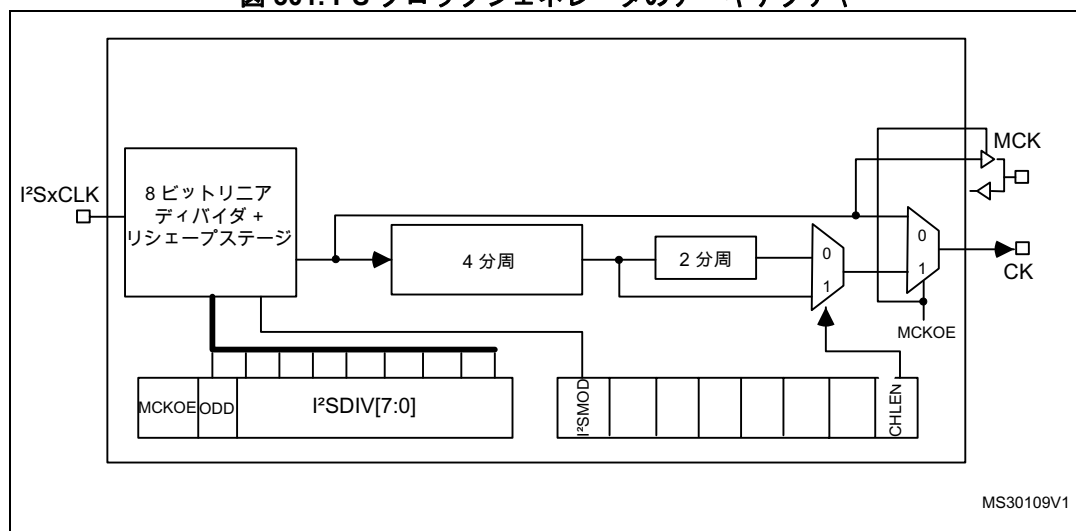
パケット長が 32 ビットの場合、I²S ビットレート = $32 \times 2 \times f_s$ となります。

図 300. オーディオサンプリング周波数の定義



マスタモードが設定された場合、希望するオーディオ周波数で通信するために、特定の措置を講じてリニアディバイダを適切にプログラムする必要があります。

図 301. I²S クロックジェネレータのアーキテクチャ



1. x は 2 または 3 です。

[図 301](#) に、通信クロックのアーキテクチャを示します。I2Sx クロックは常にシステムクロックです。

オーディオサンプリング周波数の取り得る値は、192 kHz、96 kHz、48 kHz、44.1 kHz、32 kHz、22.05 kHz、16 kHz、11.025 kHz、8 kHz（またはその他のこの範囲内にある値）。希望する周波数を達成するには、次の式に基づいてリニアディバイダをプログラムする必要があります。

マスタクロックが生成される (SPIx_I2SPR レジスタの MCKOE ビットをセットされる) 場合

$$f_S = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD) \times 8] \dots \text{チャンネルフレームが 16 ビット幅の場合}$$
$$f_S = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD) \times 4] \dots \text{チャンネルフレームが 32 ビット幅の場合}$$

マスタクロックが無効にされた場合 (MCKOE ビットをクリア)

$$f_S = I2SxCLK / [(16 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャンネルフレームが 16 ビット幅の場合}$$
$$f_S = I2SxCLK / [(32 \times 2) \times ((2 \times I2SDIV) + ODD)] \dots \text{チャンネルフレームが 32 ビット幅の場合}$$

表 105 に、さまざまなクロック設定に対する精度値の例を示します。

注：最適クロック精度を可能にするその他の設定も可能です。

表 105. 標準 8 MHz HSE を使用した場合のオーディオ周波数精度⁽¹⁾

SYSCLK (MHz)	データ長	I2SDIV	I2SODD	MCLK	ターゲット fs (Hz)	実際の fs (Hz)	誤差
48	16	8	0	なし	96000	93750	2.3438%
48	32	4	0	なし	96000	93750	2.3438%
48	16	15	1	なし	48000	48387.0968	0.8065%
48	32	8	0	なし	48000	46875	2.3438%
48	16	17	0	なし	44100	44117.647	0.0400%
48	32	8	1	なし	44100	44117.647	0.0400%
48	16	23	1	なし	32000	31914.8936	0.2660%
48	32	11	1	なし	32000	32608.696	1.9022%
48	16	34	0	なし	22050	22058.8235	0.0400%
48	32	17	0	なし	22050	22058.8235	0.0400%
48	16	47	0	なし	16000	15957.4468	0.2660%
48	32	23	1	なし	16000	15957.447	0.2660%
48	16	68	0	なし	11025	11029.4118	0.0400%
48	32	34	0	なし	11025	11029.412	0.0400%
48	16	94	0	なし	8000	7978.7234	0.2660%
48	32	47	0	なし	8000	7978.7234	0.2660%
48	16	2	0	あり	48000	46875	2.3430%
48	32	2	0	あり	48000	46875	2.3430%
48	16	2	0	あり	44100	46875	6.2925%
48	32	2	0	あり	44100	46875	6.2925%
48	16	3	0	あり	32000	31250	2.3438%
48	32	3	0	あり	32000	31250	2.3438%
48	16	4	1	あり	22050	20833.333	5.5178%
48	32	4	1	あり	22050	20833.333	5.5178%
48	16	6	0	あり	16000	15625	2.3438%
48	32	6	0	あり	16000	15625	2.3438%

表 105. 標準 8 MHz HSE を使用した場合のオーディオ周波数精度⁽¹⁾ (続き)

SYSCCLK (MHz)	データ長	I2SDIV	I2SODD	MCLK	ターゲット fs (Hz)	実際の fs (Hz)	誤差
48	16	8	1	あり	11025	11029.4118	0.0400%
48	32	8	1	あり	11025	11029.4118	0.0400%
48	16	11	1	あり	8000	8152.17391	1.9022%
48	32	11	1	あり	8000	8152.17391	1.9022%

1. この表の値は、さまざまなクロック設定に対するほんの一例です。最適クロック精度を可能にするその他の設定も可能です。

27.7.5 I²S マスタモード

I²S はマスタモードで設定できます。つまり、シリアルクロックは、ワードセレクト信号 WS だけでなく、CK ピン上でも生成されます。マスタクロック (MCK) は、SPIx_I2SPR レジスタの MCKOE ビットで、出力するかしないかを制御できます。

手順

- 適切なオーディオサンプリング周波数に到達するシリアルクロックボーレートを定義するため、SPIx_I2SPR レジスタの I2SDIV[7:0] ビットを選択します。SPIx_I2SPR レジスタの ODD ビットも定義する必要があります。
- 通信クロックの一定したレベルを定義するために、CKPOL ビットを選択します。外部の DAC/ADC オーディオコンポーネントにマスタクロック MCK を供給する必要がある場合、SPIx_I2SPR レジスタの MCKOE ビットをセットします (I2SDIV と ODD の値は、MCK 出力の状態に応じて計算する必要があります。詳細については、[セクション 27.7.4 : クロックジェネレータ](#)を参照)。
- SPIx_I2SCFGR レジスタの I2SMOD ビットをセットして I²S 機能を有効にし、I2SSTD[1:0] と PCMSYNC ビットにより I²S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してチャンネルあたりのビット数を、それぞれ選択します。SPIx_I2SCFGR レジスタの I2SCFG[1:0] ビットにより I²S マスタモードと方向 (トランスミッタまたはレシーバ) も選択します。
- 必要な場合は、SPIx_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
- SPIx_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

WS と CK は出力モードに設定されます。SPIx_I2SPR の MCKOE ビットがセットされている場合、MCK も出力です。

送信シーケンス

送信シーケンスは、Txバッファにハーフワードが書き込まれたときに開始されます。

Txバッファに書き込まれる最初のデータは、左チャンネルのデータに対応すると想定します。データが Txバッファからシフトレジスタに転送されると、TXE がセットされ、右チャンネルに対応するデータを Txバッファに書き込む必要があります。CHSIDE フラグは、どのチャンネルが送信されるかを示します。CHSIDE フラグは TXE がハイレベルになったときに更新されるため、このフラグは TXE フラグがセットされた場合に意味を持ちます。

完全なフレームとは、左チャンネルのデータ送信と、それに続く右チャンネルのデータ送信であるとみなす必要があります。左チャンネルのみが送信される部分的フレームは実現できません。

データハーフワードは、最初のビット送信時に 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

選択された I²S 規格モードに応じた書き込み動作の詳細については、[セクション 27.7.2: サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx_DR レジスタに書きこむ必要があります。

I2SE をクリアすることによって I²S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

受信シーケンス

動作モードは、ポイント 3 を除いて、送信モードの場合と同じです（[セクション 27.7.5: I2S マスタモード](#)に示す手順を参照）。ポイント 3 では、I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャネルの長さに関係なく、オーディオデータは 16 ビットの packets によって受信されます。つまり、Rxバッファが満杯になるたびに RXNE フラグがセットされ、SPIx_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャネルまたは左チャネルに受信されることがあります。

RXNE ビットは、SPIx_DR レジスタの読み出しによってクリアされます。

CHSIDE は毎回の受信後に更新されます。CHSIDE は、I²S セルによって生成される WS 信号に反応します。

選択された I²S 規格モードに応じた読み出し動作の詳細については、[セクション 27.7.2: サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx_CR2 レジスタの ERRIE ビットがセットされている場合、割り込みが生成されてエラーを知らせます。

I²S をスイッチオフするには、I²S が新しいデータ転送を開始することなく転送サイクルを適切に完了できるように、特定の動作が要求されます。そのシーケンスは、データ長とチャンネル長の設定、および選択したオーディオプロトコルモードに依存します。ケース別の説明

- 32 ビットチャンネル長に拡張された 16 ビットデータ長 (DATLEN = 00、CHLEN = 1)、LSB 詰めモードを使用 (I2SSTD = 10)
 - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
 - b) 次に I²S の 17 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。
- 32 ビットチャンネル長に拡張された 16 ビットデータ長 (DATLEN = 00 および CHLEN = 1)、MSB 詰め、I²S または PCM モード (それぞれ、I2SSTD = 00、I2SSTD = 01、または I2SSTD = 11)
 - a) 最後の RXNE を待ちます。
 - b) 次に I²S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。
- DATLEN と CHLEN のその他すべての組み合わせについては、I2SSTD ビットを通じて選択したオーディオモードが何であれ、次のシーケンスを実行して I²S をスイッチオフします。
 - a) 最後から 2 番目の RXNE = 1 (n - 1) を待ちます。
 - b) 次に I²S の 1 クロックサイクルを待ちます (ソフトウェアループを使用)。
 - c) I²S を無効にします (I2SE = 0)。

注： 転送時、BSY フラグはローレベルに保持されます。

27.7.6 I²S スレーブモード

スレーブ設定の場合、I²S は送信または受信モードに設定できます。動作モードは、主として I²S マスタ設定に関して述べたものと同じ規則に従います。スレーブモードでは、I²S インタフェースによって生成されるクロックはありません。クロックと WS 信号は、I²S インタフェースに接続された外部マスタから入力されます。したがって、ユーザがクロックを設定する必要はありません。

設定ステップを次に示します。

1. SPIx_I2SCFGR レジスタの I2SMOD ビットをセットして I²S モードを選択し、I2SSTD[1:0] ビットにより I²S 規格を、DATLEN[1:0] ビットによりデータ長を、CHLEN ビットを設定してフレームのチャンネルあたりのビット数を、それぞれ選択します。SPIx_I2SCFGR レジスタの I2SCFG[1:0] ビットにより、スレーブのモード (送信または受信) も選択します。
2. 必要な場合は、SPIx_CR2 レジスタに書き込むことによって、可能性のあるすべての割り込みソースと DMA 機能を選択します。
3. SPIx_I2SCFGR レジスタの I2SE ビットをセットする必要があります。

送信シーケンス

送信シーケンスは、外部マスタデバイスがクロックを送信したときと、NSS_WS 信号がデータの転送を要求したときに開始されます。スレーブを有効にしなければ、外部マスタは通信を開始しません。I²S データレジスタは、マスタが通信を開始する前に、ロードする必要があります。

I²S、MSB 詰め、および LSB 詰めモードの場合、データレジスタに書き込まれる最初のデータは、左チャンネルのデータに対応します。通信が開始されると、データは Tx バッファからシフトレジスタに転送されます。次に、右チャンネルのデータを I²S データレジスタに書き込むように要求するために、TXE フラグがセットされます。

CHSIDE フラグは、どのチャンネルが送信されるかを示します。スレーブモードでは、マスタ送信モードに比べて、CHSIDE は外部マスタからの WS 信号に反応します。つまり、スレーブが最初のデータの送信準備をしなければ、マスタはクロックを生成できません。WS アサーションは、最初に送信される左チャンネルに対応します。

注： **I2SE は、マスタの最初のクロックが CK ラインに到達するよりも、少なくとも 2 PCLK サイクル前に書き込まれる必要があります。**

データハーフワードは、最初のビット送信時に内部バスから 16 ビットシフトレジスタに同時にロードされてから、MOSI/SD ピンに MSB ファーストで連続的にシフトアウトされます。TXE フラグは、Txバッファからシフトレジスタへの毎回の転送後にセットされ、SPIx_CR2 レジスタの TXEIE ビットがセットされている場合は割り込みが生成されます。

なお、Txバッファへの書き込みの前に、TXE フラグが 1 であることを確認する必要があります。

選択された I²S 規格モードに応じた書き込み動作の詳細については、[セクション 27.7.2：サポートされるオーディオプロトコル](#)を参照してください。

連続したオーディオデータ送信を行うには、現在の送信が終了する前に、次の送信データを SPIx_DR レジスタに書きこむ必要があります。次のデータ通信の最初のクロックエッジよりも前にデータが SPIx_DR レジスタに書き込まれない場合、アンダーランフラグがセットされ、割り込みが生成されることがあります。これによって、転送データに誤りがあることがソフトウェアに知らされます。SPIx_CR2 レジスタの ERRIE ビットがセットされた場合、SPIx_SR レジスタの UDR フラグがハイレベルになると割り込みが生成されます。この場合、I²S をスイッチオフし、左チャンネルからデータ転送をリスタートする必要があります。

I2SE をクリアすることによって I²S をスイッチオフするには、TXE = 1 および BSY = 0 になるまで待つ必要があります。

受信シーケンス

動作モードは、ポイント 1 を除いて、送信モードの場合と同じです（[セクション 27.7.6：I2S スレーブモード](#)に示す手順を参照）。ポイント 1 では、SPIx_I2SCFGR レジスタの I2SCFG[1:0] ビットを通じてマスタ受信モードを設定する必要があります。

データやチャンネルの長さに関係なく、オーディオデータは 16 ビットのパケットによって受信されます。つまり、Rxバッファがフルになるたびに SPIx_SR レジスタの RXNE フラグがセットされ、SPIx_CR2 レジスタの RXNEIE ビットがセットされている場合は割り込みが生成されます。データ長とチャンネル長の設定にもよりますが、Rxバッファへの 1 回または 2 回の受信によって、オーディオ値が右チャンネルまたは左チャンネルに受信されることがあります。

CHSIDE フラグは、SPIx_DR レジスタから読み出されるデータが受信されるたびに更新されます。このフラグは、外部マスタコンポーネントによって管理される外部 WS ラインに反応します。

RXNE ビットは、SPIx_DR レジスタの読み出しによってクリアされます。

選択された I²S 規格モードに応じた読み出し動作の詳細については、[セクション 27.7.2：サポートされるオーディオプロトコル](#)を参照してください。

前の受信データがまだ読み出されていない間にデータが受信された場合、オーバーランが生成され、OVR フラグがセットされます。SPIx_CR2 レジスタの ERRIE ビットがセットされた場合、エラーを示すために割り込みが生成されます。

受信モードで I²S をスイッチオフするには、最後の RXNE = 1 を受信した直後に I2SE をクリアする必要があります。

注： **外部マスタコンポーネントには、オーディオチャネルを介して 16 ビットまたは 32 ビットのパケットでデータを送受信する機能が必要です。**



27.7.7 I²S ステータスフラグ

アプリケーションが I²S バスの状態を完全に監視できるように、3 つのステータスフラグが用意されています。

ビジーフラグ (BSY)

BSY フラグは、ハードウェアによってセット/クリアされます（このフラグへの書き込みは無効）。このフラグは I²S の通信層の状態を示します。

BSY がセットされると、I²S が通信中でビジーであることを示します。マスタ受信モード (I2SCFG=11) には 1 つの例外があり、BSY フラグは受信時にローレベルに保持されます。

ソフトウェアが I²S を無効にする必要がある場合、BSY フラグは転送の終わりを検出するために役立ちます。これによって、最後の転送データの破壊を回避します。このため、下記の手順を厳守する必要があります。

BSY フラグは転送が開始されるとセットされます。ただし、I²S がマスタ受信モードにある場合を除きます。

BSY フラグをクリアするタイミング

- 転送が完了したとき（ただし、通信が連続的だと思われるマスタ送信モードの場合を除く）
- I²S が無効化されたとき

通信が連続的な場合

- マスタ送信モードでは、BSY フラグはすべての転送期間を通じてハイレベルに保持されます。
- スレーブモードでは、BSY フラグは、各転送間で I²S の 1 クロックサイクルの間ローレベルになります。

注： 各データの送受信の処理には BSY フラグを使用しないでください。代わりに、TXE フラグと RXNE フラグを使用することをお勧めします。

Txバッファエンプティフラグ (TXE)

このフラグがセットされると、Txバッファはエンプティ（空）であり、次に送信するデータをバッファにロードできることを示します。送信されるデータがすでにTxバッファに格納されているとき、TXE フラグはリセットされます。TXE フラグは、I²S が無効にされている (I2SE ビットがリセット) ときにもリセットされます。

Rxバッファノットエンプティ (RXNE)

このフラグがセットされると、Rxバッファに有効な受信データがあることを示します。このフラグは、SPIx_DR レジスタが読み出されるとリセットされます。

チャネルサイドフラグ (CHSIDE)

送信モードでは、このフラグは TXE がハイレベルになるとリフレッシュされます。このフラグは、SD 上の転送データが属するチャネルサイドを示します。スレーブ送信モードでアンダーランエラーイベントが発生した場合、このフラグは信頼できないため、通信を再開する前に、I²S をスイッチオフし、さらにスイッチオンする必要があります。

受信モードでは、このフラグは SPIx_DR にデータが受信されるとリフレッシュされます。このフラグは、どちらのチャネルサイドからデータが受信されたかを示します。なお、エラー (OVR など) が発生した場合、このフラグは無意味になるため、I²S を無効にし、さらに有効にすることによってリセットする必要があります（変更が必要な場合は設定する）。

このフラグは、PCM 規格では意味を持ちません（ショートとロングフレームの両モード）。

SPIx_SR の OVR または UDR フラグがセットされ、SPIx_CR2 の ERRIE ビットもセットされると、割り込みが生成されます。この割り込みをクリアするには、割り込みソースをいったんクリアした後で、SPIx_SR ステータスレジスタを読み出します。

27.7.8 I²S エラーフラグ

I²S セルには 3つのエラーフラグがあります。

アンダーランフラグ (UDR)

スレーブ送信モードでは、ソフトウェアが SPIx_DR にまだ値をロードしていない間に、データ送信用の最初のクロックが現われると、このフラグがセットされます。このフラグは、SPIx_I2SCFGR レジスタの I2SMOD ビットがセットされると使用できます。SPIx_CR2 レジスタの ERRIE ビットがセットされている場合は、割り込みを生成できます。

UDR ビットは、SPIx_SR レジスタの読み出し動作によってクリアされます。

オーバーランフラグ (OVR)

このフラグがセットされるのは、データが受信され、前のデータが SPIx_DR レジスタからまだ読み出されていないときです。結果として、受信データは失われます。SPIx_CR2 の ERRIE ビットがセットされている場合、割り込みが生成されることがあります。

この場合、受信バッファの内容は、送信側のデバイスからの新しい受信データによって更新されません。SPIx_DR レジスタへの読み出し動作によって、前に正しく受信されたデータが返されます。それ以降に送信されたすべてのハーフワードは失われます。

OVR ビットをクリアするには、SPIx_DR レジスタを読み出し、続けて SPIx_SR レジスタに読み出しアクセスを行います。

フレームエラーフラグ (FRE)

このフラグは、I²S がスレーブモードに設定された場合にのみハードウェアによってセットすることができます。このフラグは、スレーブが WS ラインの変化を想定していないときに外部マスタが WS ラインを変化させようとしている場合にセットされます。同期が失われた場合、この状態から回復し、外部マスタデバイスと I²S スレーブデバイスを再同期させるためには、次のステップに従います。

1. I²S を無効にします。
2. 正しいレベルが WS ラインで検出されたとき (WS ラインは I²S モードではハイレベル、MSB 詰めまたは LSB 詰めあるいは PCM モードではローレベル)、再度有効にします。

マスタデバイスとスレーブデバイスの間の同期外れは、CK 通信クロック上または WS フレーム同期ライン上のノイズの多い環境に起因する可能性があります。ERRIE ビットをセットすれば、エラー割り込みを生成させることができます。同期外れフラグ (FRE) は、ステータスレジスタを読み出すときに、ソフトウェアでクリアされます。

27.7.9 DMA の機能

I²S モードでは、DMA は SPI モードとまったく同じように機能します。異なる点は、I²S モードではデータ転送保護システムがないため、CRC 機能を使用できない点のみです。

27.8 I²S 割り込み

表 106 に I²S 割り込みのリストを示します。

表 106. I²S 割り込みリクエスト

割り込みイベント	イベントフラグ	イネーブル制御ビット
送信バッファエンプティフラグ	TXE	TXEIE
受信バッファノットエンプティフラグ	RXNE	RXNEIE
オーバーランエラー	OVR	ERRIE
アンダーランエラー	UDR	
フレームエラーフラグ	FRE	

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

27.9 SPI および I²S レジスタ

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。さらに、SPI_DR へは8 ビット単位でアクセスできます。

27.9.1 SPI 制御レジスタ 1 (SPIx_CR1)

アドレスオフセット : 0x00

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDI MODE	BIDI OE	CRC EN	CRC NEXT	CRCL	RX ONLY	SSM	SSI	LSB FIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

ビット 15 **BIDIMODE** : 双方向データモードイネーブル。このビットは、共通の双方向データラインを 1 本使用して、半二重通信を有効にします。双方向モードがアクティブのときは、RXONLY ビットをクリアされたままにします。

0 : 2 線単方向データモードを選択します。
1 : 1 線双方向データモードを選択します。

注 : このビットは I²S モードでは使用しません。

ビット 14 **BIDIOE** : 双方向モードでの出力イネーブル
双方向モードでの転送方向は、このビットと BIDIMODE ビットを組み合わせで選択します。
0 : 出力は無効です (受信専用モード)。
1 : 出力は有効です (送信専用モード)。

注 : マスタモードでは MOSI ピンが使用され、スレーブモードでは MISO ピンが使用されます。
このビットは I²S モードでは使用しません。

ビット 13 **CRCEN** : ハードウェア CRC 計算イネーブル
0 : CRC 計算は無効です。
1 : CRC 計算は有効です。
注 : 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。
このビットは I²S モードでは使用しません。

ビット 12 **CRCNEXT** : 送信 CRC Next
0 : 次の送信値は Tx バッファから送信されます。
1 : 次の送信値は Tx CRC レジスタから送信されます。
注 : このビットは、最後のデータが SPIx_DR レジスタに書き込まれた直後に書き込む必要があります。
このビットは I²S モードでは使用しません。

ビット 11 **CRCL** : CRC 長
このビットは、CRC 長をセットするために、ソフトウェアによってセット/クリアされます。
0 : 8 ビットの CRC 長
1 : 16 ビットの CRC 長
注 : 正しい動作のためには、このビットへの書き込みは、SPI が無効 (SPE = 0) のときにのみ行います。
このビットは I²S モードでは使用しません。



ビット 10 RXONLY : 受信専用モードイネーブル

このビットは、データ受信専用の単方向ラインを 1 本使用して、単方向通信を有効にします。受信専用モードがアクティブのときは、BIDIMODE ビットをクリアされたままにします。このビットはマルチスレーブシステムでも役立ちます。そのシステムでは、この特定のスレーブはアクセスされず、アクセスされたスレーブからの出力は破壊されません。

0 : 全二重 (送受信)

1 : 出力は無効です (受信専用モード)。

注： このビットは I²S モードでは使用しません。

ビット 9 SSM : ソフトウェアスレーブ管理

SSM ビットがセットされているとき、NSS ピンの入力は SSI ビットからの値に置き換えられます。

0 : ソフトウェアスレーブ管理は無効です。

1 : ソフトウェアスレーブ管理は有効です。

注： このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 8 SSI : 内部スレーブ選択

このビットは、SSM ビットがセットされているときにのみ有効です。このビットの値は強制的に NSS ピンに設定され、NSS ピンの I/O 値は無視されます。

注： このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 7 LSBFIRST : フレームフォーマット

0 : データは MSB ファーストとともに送信/受信されます。

1 : データは LSB ファーストとともに送信/受信されます。

注： 1.このビットは、通信中には変更しないでください。

2.このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 6 SPE : SPI イネーブル

0 : ペリフェラルは無効です。

1 : ペリフェラルは有効です。

注： SPI を無効にするときは、SPI を無効にする手順 (758 ページ) に記載されている手順に従ってください。

このビットは I²S モードでは使用しません。

ビット 5:3 BR[2:0] : ボーレート制御

000 : $f_{HCLK} / 2$

001 : $f_{HCLK} / 4$

010 : $f_{HCLK} / 8$

011 : $f_{HCLK} / 16$

100 : $f_{HCLK} / 32$

101 : $f_{HCLK} / 64$

110 : $f_{HCLK} / 128$

111 : $f_{HCLK} / 256$

注： これらのビットは、通信中には変更しないでください。

このビットは I²S モードでは使用しません。

- ビット 2 **MSTR** : マスタ選択
- 0 : スレーブ設定
 - 1 : マスタ設定
- 注 :** このビットは、通信中には変更しないでください。
このビットは I²S モードでは使用しません。
- ビット 1 **CPOL** : クロック極性
- 0 : アイドル時に CK は 0 となります。
 - 1 : アイドル時に CK は 1 となります。
- 注 :** このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。
- ビット 0 **CPHA** : クロック位相
- 0 : 最初のクロック遷移が最初のデータキャプチャエッジです。
 - 1 : 2 番目のクロック遷移が最初のデータキャプチャエッジです。
- 注 :** このビットは、通信中には変更しないでください。
このビットは I²S モードおよび SPI TI モードでは使用しません。

27.9.2 SPI 制御レジスタ 2 (SPIx_CR2)

アドレスオフセット : 0x04

リセット値 : 0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXTH	DS [3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 15 予約済みであり、リセット値に保持する必要があります。
- ビット 14 **LDMA_TX** : 送信の最後の DMA 転送
- このビットは、DMA で送信するデータの合計数が奇数であるか偶数であるかを定義するために、データバッキングモードで使用されます。これは、SPIx_CR2 レジスタの TXDMAEN ビットがセットされている場合で、バッキングモードが使用されている場合にのみ意味を持ちます (データ長 ≤ 8 ビット、SPIx_DR への書き込みアクセスは 16 ビット幅)。SPI が無効化された場合に書き込む必要があります (SPIx_CR1 レジスタで SPE = 0)。
- 0 : 転送データ項目の数は偶数です。
 - 1 : 転送データ項目の数は奇数です。
- 注 :** **CRCEN** ビットがセットされている場合、**SPI を無効にする手順 (758 ページ)** を参照してください。
このビットは I²S モードでは使用しません。
- ビット 13 **LDMA_RX** : 受信の最後の DMA 転送
- このビットは、DMA で受信するデータの合計数が奇数であるか偶数であるかを定義するために、データバッキングモードで使用されます。これは、SPIx_CR2 レジスタの RXDMAEN ビットがセットされている場合で、バッキングモードが使用されている場合にのみ意味を持ちます (データ長 ≤ 8 ビット、SPIx_DR への書き込みアクセスは 16 ビット幅)。SPI が無効化された場合に書き込む必要があります (SPIx_CR1 レジスタで SPE = 0)。
- 0 : 転送データ項目の数は偶数です。
 - 1 : 転送データ項目の数は奇数です。
- 注 :** **CRCEN** ビットがセットされている場合、**SPI を無効にする手順 (758 ページ)** を参照してください。
このビットは I²S モードでは使用しません。

ビット 12 FRXTH : FIFO 受信閾値

このビットは、RXNE イベントをトリガする RXFIFO の閾値をセットするために使用されます。

0 : FIFO レベルが 1/2 (16 ビット) 以上である場合に RXNE イベントが生成されます。

1 : FIFO レベルが 1/4 (8 ビット) 以上である場合に RXNE イベントが生成されます。

注： このビットは I²S モードでは使用しません。

ビット 11:8 DS [3:0] : データサイズ

以下のビットは SPI 転送のデータ長を設定します。

0000 : 未使用

0001 : 未使用

0010 : 未使用

0011 : 4 ビット

0100 : 5 ビット

0101 : 6 ビット

0110 : 7 ビット

0111 : 8 ビット

1000 : 9 ビット

1001 : 10 ビット

1010 : 11 ビット

1011 : 12 ビット

1100 : 13 ビット

1101 : 14 ビット

1110 : 15 ビット

1111 : 16 ビット

ソフトウェアが「未使用」値のいずれかの書き込みを試みた場合、値は強制的に「0111」(8 ビット)になります。

注： このビットは I²S モードでは使用しません。

ビット 7 TXEIE : Txバッファエンプティ割り込みイネーブル

0 : TXE 割り込みはマスクされます。

1 : TXE 割り込みはマスクされません。TXE フラグがセットされたとき、割り込みリクエストの生成に使用されます。

ビット 6 RXNEIE : Rxバッファノットエンプティ割り込みイネーブル

0 : RXNE 割り込みはマスクされます。

1 : RXNE 割り込みはマスクされません。RXNE フラグがセットされたとき、割り込みリクエストの生成に使用されます。

ビット 5 ERRIE : エラー割り込みイネーブル

このビットは、エラー状態が発生したとき (SPI モードでは CRCERR、OVR、MODF ; TI モードでは FRE ; I²S モードでは UDR、OVR、FRE)、割り込みの生成を制御します。

0 : エラー割り込みはマスクされます。

1 : エラー割り込みは有効です。

ビット 4 FRF : フレームフォーマット

0 : SPI モトローラモード

1 : SPI TI モード

注： このビットは、SPI が無効 (SPE=0) のときにのみ書き込む必要があります。

このビットは I²S モードでは使用しません。

ビット 3 NSSP : NSS パルス管理

このビットは、マスタモードでのみ使用されます。これにより、SPI は連続転送中に 2 つの連続したデータ間で NSS パルスを生成できます。単一のデータ転送の場合、転送後、NSS ピンは強制的にハイレベルになります。

CPHA = '1' の場合も FRF = '1' の場合も意味を持ちません。

0 : NSS パルスなし

1 : NSS パルス発生回路

注： 1.このビットは、SPI が無効 (SPE=0) のときにのみ書き込む必要があります。
2.このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 2 SSOE : SS 出力イネーブル

0 : マスタモードで SS 出力は無効にされ、SPI インタフェースはマルチマスタ設定で機能できます。

1 : SPI インタフェースが有効であるとき、マスタモードで SS 出力は有効です。SPI インタフェースはマルチマスタ環境では機能できません。

注： このビットは I²S モードおよび SPI TI モードでは使用しません。

ビット 1 TXDMAEN : Txバッファ DMA イネーブル

このビットがセットされると、TXE フラグがセットされるたびに DMA リクエストが生成されます。

0 : Txバッファ DMA は無効です。

1 : Txバッファ DMA は有効です。

ビット 0 RXDMAEN : Rxバッファ DMA イネーブル

このビットがセットされると、RXNE フラグがセットされるたびに DMA リクエストが生成されます。

0 : Rxバッファ DMA は無効です。

1 : Rxバッファ DMA は有効です。

27.9.3 SPI ステータスレジスタ (SPIx_SR)

アドレスオフセット : 0x08

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTLV[1:0]		FRLVL[2:0]		FRE	BSY	OVR	MODF	CRC ERR	UDR	CHSIDE	TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0	r	r	r	r

ビット 15:13 予約済みであり、リセット値に保持する必要があります。

ビット 12:11 **FTLV[1:0]** : FIFO 送信レベル

これらのビットは、ハードウェアによってセット/クリアされます。

- 00 : FIFO エンプティ
- 01 : 1/4 FIFO
- 10 : 1/2 FIFO
- 11 : FIFO フル (FIFO 閾値が 1/2 より大きい場合、FULL とみなす)

注 : これらのビットは I²S モードでは使用されません。

ビット 10:9 **FRLVL[1:0]** : FIFO 受信レベル

これらのビットは、ハードウェアによってセット/クリアされます。

- 00 : FIFO エンプティ
- 01 : 1/4 FIFO
- 10 : 1/2 FIFO
- 11 : FIFO フル

注 : これらのビットは、CRC 計算が有効なときは I²S モードや SPI 受信専用モードでは使用されません。

ビット 8 **FRE** : フレームフォーマットエラー

このフラグは、TI スレーブモードと I²S スレーブモードの SPI で使用されます。[セクション 27.5.10 : SPI エラーフラグ](#)および[セクション 27.7.8 : I2S エラーフラグ](#)を参照してください。

このフラグは、ハードウェアによってセットされ、SPIx_SR がソフトウェアによって読み出されるとリセットされます。

- 0 : フレームフォーマットエラーはありません。
- 1 : フレームフォーマットエラーが発生しました。

ビット 7 **BSY** : ビジーフラグ

- 0 : SPI (または I2S) はビジー状態ではありません。
 - 1 : SPI (または I2S) が通信ビジー状態であるか、または Tx バッファが空ではありません。
- このフラグはハードウェアによってセット/クリアされます。

注 : BSY フラグを使用する際は注意が必要です。[セクション 27.5.9 : SPI ステータスフラグおよび SPI を無効にする手順 \(758 ページ\)](#)を参照してください。

ビット 6 **OVR** : オーバーランフラグ

- 0 : オーバーランは発生していません。
 - 1 : オーバーランが発生しました。
- このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[I2S エラーフラグ \(789 ページ\)](#)を参照してください。

ビット 5 **MODF** : モードフォールト

- 0 : モードフォールトは発生していません。
 - 1 : モードフォールトが発生しました。
- このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[セクション : モードフォールト \(MODF\) \(767 ページ\)](#)を参照してください。

注： このビットはI²S モードでは使用しません。

ビット 4 **CRCERR** : CRC エラーフラグ

- 0 : 受信した CRC 値が SPIx_RXCRCR 値と一致します。
- 1 : 受信した CRC 値が SPIx_RXCRCR 値と一致しません。

このフラグは、ハードウェアによってセットされ、ソフトウェアによって 0 を書き込むことでクリアされます。

注： このビットはI²S モードでは使用しません。

ビット 3 **UDR** : アンダーランフラグ

- 0 : アンダーランは発生していません。
- 1 : アンダーランが発生しました。

このフラグは、ハードウェアによってセットされ、ソフトウェアシーケンスによってリセットされます。ソフトウェアシーケンスについては、[I2S エラーフラグ \(789 ページ\)](#) を参照してください。

注： このビットはSPI モードでは使用しません。

ビット 2 **CHSIDE** : チャネルサイド

- 0 : 左チャネルを送信する必要があるか、または受信が行われました。
- 1 : 右チャネルを送信する必要があるか、または受信が行われました。

注： このビットはSPI モードでは使用しません。PCM モードでは意味を持ちません。

ビット 1 **TXE** : 送信バッファエンプティ

- 0 : Txバッファは空ではありません。
- 1 : Txバッファは空です。

ビット 0 **RXNE** : 受信バッファノットエンプティ

- 0 : Rxバッファは空です。
- 1 : Rxバッファは空ではありません。

27.9.4 SPI データレジスタ (SPIx_DR)

アドレスオフセット : 0x0C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **DR[15:0]** : データレジスタ

受信したデータまたは送信されるデータ

このデータレジスタは、Rx および Tx FIFO 間のインタフェースとして使用できます。データレジスタが読み出されると、データレジスタへの書き込みが TxFIFO にアクセスしている間に、Rx FIFO にアクセスされます ([セクション 27.5.8 : データの送受信手順](#)を参照)。

注： データは常に右詰めです。未使用のビットは、レジスタへの書き込み時に無視され、レジスタの読み出し時にゼロとして読み出されます。Rx 閾値設定は、常に現在使用中の読み出しアクセスに対応している必要があります。

27.9.5 SPI CRC 多項式レジスタ (SPIx_CRCPR)

アドレスオフセット : 0x10

リセット値 : 0x0007

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:0 **CRCPOLY[15:0]** : CRC 多項式レジスタ

このレジスタは、CRC 計算用の多項式を格納します。

CRC 多項式 (0007h) は、このレジスタのリセット値です。必要に応じて、別の多項式を設定することができます。

注 : 多項式の値は必ず奇数でなければなりません。偶数の値はサポートされていません。

27.9.6 SPI Rx CRC レジスタ (SPIx_RXCRCR)

アドレスオフセット : 0x14

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **RxCRC[15:0]** : Rx CRC レジスタ

CRC 計算が有効なとき、RxCRC[15:0] ビットには、その後に受信したバイトから算出された CRC 値が格納されています。このレジスタは、SPIx_CR1 レジスタの CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPIx_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPIx_CR1 の CRCL ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPIx_CR1 レジスタの CRCL ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

注 : BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。

これらのビットは I²S モードでは使用されません。

27.9.7 SPI Tx CRC レジスタ (SPIx_TXCRCR)

アドレスオフセット : 0x18

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxCRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **TxCRC[15:0]** : Tx CRC レジスタ

CRC 計算が有効なとき、TxCRC[7:0] ビットには、その後に送信されたバイトから算出された CRC 値が格納されます。このレジスタは、SPIx_CR1 の CRCEN ビットに 1 が書き込まれたときにリセットされます。CRC は、SPIx_CRCPR レジスタにプログラムされた多項式を使用して連続的に計算されます。

データフレームフォーマットが 8 ビットデータに設定された場合 (SPIx_CR1 の CRCL ビットがクリアされている)、8 つの LSB ビットのみが考慮されます。CRC 計算は、任意の CRC8 規格に基づいて行われます。

16 ビットデータフレームフォーマットが選択された場合 (SPIx_CR1 レジスタの CRCL ビットがセットされている)、このレジスタの 16 ビット全体が考慮されます。CRC 計算は、任意の CRC16 規格に基づいて行われます。

注: *BSY フラグがセットされているときにこのレジスタを読み出すと、誤った値が返されることがあります。*

これらのビットは I²S モードでは使用されません。

27.9.8 SPIx_I2S 設定レジスタ (SPIx_I2SCFGR)

アドレスオフセット : 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	I2SMOD	I2SE	I2SCFG		PCMSYNC	Res.	I2SSTD		CKPOL	DATLEN		CHLEN
				rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

ビット 15:12 予約済み。ハードウェアによって 0 に固定されています。

ビット 11 **I2SMOD** : I2S モード選択

0 : SPI モードが選択されます。

1 : I2S モードが選択されます。

注: *このビットは、SPI または I²S が無効なときに設定してください。*

ビット 10 **I2SE** : I2S イネーブル

0 : I²S ペリフェラルは無効です。

1 : I²S ペリフェラルは有効です。

注: *このビットは SPI モードでは使用しません。*

ビット 9:8 **I2SCFG** : I2S 設定モード

00 : スレーブ - 送信

01 : スレーブ - 受信

10 : マスタ - 送信

11 : マスタ - 受信

注: *これらのビットは、I²S が無効なときに設定してください。*

これらは SPI モードでは使用しません。



ビット 7 **PCMSYNC** : PCM フレーム同期

0 : ショートフレーム同期

1 : ロングフレーム同期

注 : このビットは、I²SSTD = 11 (PCM 規格使用) の場合にのみ意味を持ちます。
SPI モードでは使用しません。

ビット 6 予約済みであり、ハードウェアによって 0 に固定されています。

ビット 5:4 **I²SSTD** : I²S 規格選択

00 : フィリップス I²S 規格

01 : MSB 詰め規格 (左詰め)

10 : LSB 詰め規格 (右詰め)

11 : PCM 規格

I²S 規格に関する詳細は、[セクション 27.7.2 \(773 ページ\)](#) を参照してください。

注 : 正しい動作のためには、これらのビットは、I²S が無効のときに設定してください。
これらは SPI モードでは使用しません。

ビット 3 **CKPOL** : インアクティブ状態のクロック極性

0 : I²S クロックのインアクティブ状態はローレベルです。

1 : I²S クロックのインアクティブ状態はハイレベルです。

注 : 正しい動作のためには、このビットは、I²S が無効のときに設定してください。
SPI モードでは使用しません。

CKPOL ビットは、SD 信号および WS 信号を受信または送信するために使用される CK エッジ感度には影響を与えません。

ビット 2:1 **DATLEN** : 転送されるデータ長

00 : 16 ビットデータ長

01 : 24 ビットデータ長

10 : 32 ビットデータ長

11 : 設定禁止

注 : 正しい動作のためには、これらのビットは、I²S が無効のときに設定してください。
これらは SPI モードでは使用しません。

ビット 0 **CHLEN** : チャネル長 (オーディオチャネルごとのビット数)

0 : 16 ビット幅

1 : 32 ビット幅

ビット書き込み動作は、DATLEN=00 のときにのみ意味を持ちます。そうでない場合、書き込まれた値とは無関係に、チャネル長はハードウェアによって 32 ビットに固定されます。

注 : 正しい動作のためには、このビットは、I²S が無効のときに設定してください。
SPI モードでは使用しません。

27.9.9 SPIx_I2S プリスケアラレジスタ (SPIx_I2SPR)

アドレスオフセット : 0x20

リセット値 : 0x0002

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV							
						rw	rw	rw							

ビット 15:10 予約済み。ハードウェアによって 0 に固定されています。

ビット 9 **MCKOE** : マスタクロック出カインエーブル

0 : マスタクロック出力は無効です。

1 : マスタクロック出力は有効です。

注 : このビットは、I2S が無効なときに設定してください。これらのビットは、I2S がマスタモードのときのみ使用します。

SPI モードでは使用しません。

ビット 8 **ODD** : プリスケアラの奇数分周比

0 : 実際のディバイダ値 = I2SDIV * 2

1 : 実際のディバイダ値 = (I2SDIV * 2) + 1

[セクション 27.7.3 \(781 ページ\)](#) を参照してください。

注 : このビットは、I2S が無効なときに設定してください。これらのビットは、I2S がマスタモードのときのみ使用します。

SPI モードでは使用しません。

ビット 7:0 **I2SDIV** : I2S リニアプリスケアラ

I2SDIV [7:0] = 0 または I2SDIV [7:0] = 1 は禁止されている値です。

[セクション 27.7.3 \(781 ページ\)](#) を参照してください。

注 : これらのビットは、I2S が無効なときに設定してください。これらのビットは、I2S がマスタモードのときのみ使用します。

これらは SPI モードでは使用しません。

27.9.10 SPI/I2S レジスタマップ

表 107 に、SPI/I2S レジスタマップとリセット値を示します。

表 107. SPI レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	SPIx_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BIDMODE	BIDOE	CRCE	CRCEXT	CRCL	RXONLY	SSM	SSI	LSBFIRST	SPE	BR [2:0]			MSTR	CPOL	CPHA				
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x04	SPIx_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]			TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSEO	TXDMAEN	RXDMAEN					
	リセット値																		0	0	0	0	1	1	1	0	0	0	0	0	0	0					
0x08	SPIx_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTLV[1:0]	FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCE	UDR	CHSIDE	TXE	RXNE						
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	1	0					
0x0C	SPIx_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x10	SPIx_CRCPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCPOLY[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1				
0x14	SPIx_RXCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RxCRC[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x18	SPIx_TXCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TxCRC[15:0]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x1C	SPIx_I2SCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2SMOD	I2SE	I2SCFG	PCMSYNC	Res.	I2STD	CKPOL	DATLEN	CHLEN								
	リセット値																					0	0	0	0		0	0	0	0	0	0					
0x20	SPIx_I2SPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV											
	リセット値																							0	0	0	0	0	0	0	0	1	0				

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。



28 タッチセンシングコントローラ (TSC)

このセクションは、STM32F05x、STM32F04x、STM32F07x、および STM32F09x デバイスにのみ適用されます。

28.1 概要

タッチセンシングコントローラを使用することで、どんなアプリケーションにも簡単に静電容量検出機能を追加することができます。静電容量検出技術とは、誘電材料（ガラス、プラスチックなど）により直接触れることがないよう保護されている電極の近くに近づけた指を検出することができる技術です。指（または何らかの導電性物体）によって発生する静電容量の変化は、表面電荷移動取得原理に基づき、実証済みの方法で測定されます。

タッチセンシングコントローラは、STMTouch タッチ検出ファームウェアライブラリによってフルサポートされています。この自由に利用できるライブラリにより、タッチ検出機能がエンドアプリケーションに確実に実装されます。

28.2 TSC の主な機能

タッチセンシングコントローラは、以下のような主な機能を備えています。

- 実績のある安定した表面電荷移動取得原理を採用。
- 最大 24 本の静電容量検出チャンネルに対応。
- 最大 8 本の静電容量検出チャンネルの同時取得により、非常に優れた応答時間を実現。
- スペクトル拡散機能により、ノイズの多い環境でのシステムの安定性が向上。
- 電荷移動取得シーケンスのハードウェア完全管理
- プログラム可能な電荷移動周波数
- プログラム可能なサンプリングコンデンサの I/O ピン
- プログラム可能なチャンネルの I/O ピン
- 最大カウント値をプログラムできるので、チャンネル障害時の取得時間の延長を回避。
- 割り込み機能を持つ専用の取得完了フラグおよび最大カウントエラーフラグ
- 最高 3 本の静電容量検出チャンネルに対して 1 個のサンプリングコンデンサとすることで、システム構成部品の個数を削減。
- 近接、タッチキー、リニア、およびロータリタイプのタッチセンサに対応。
- STMTouch タッチ検出ファームウェアライブラリで動作。

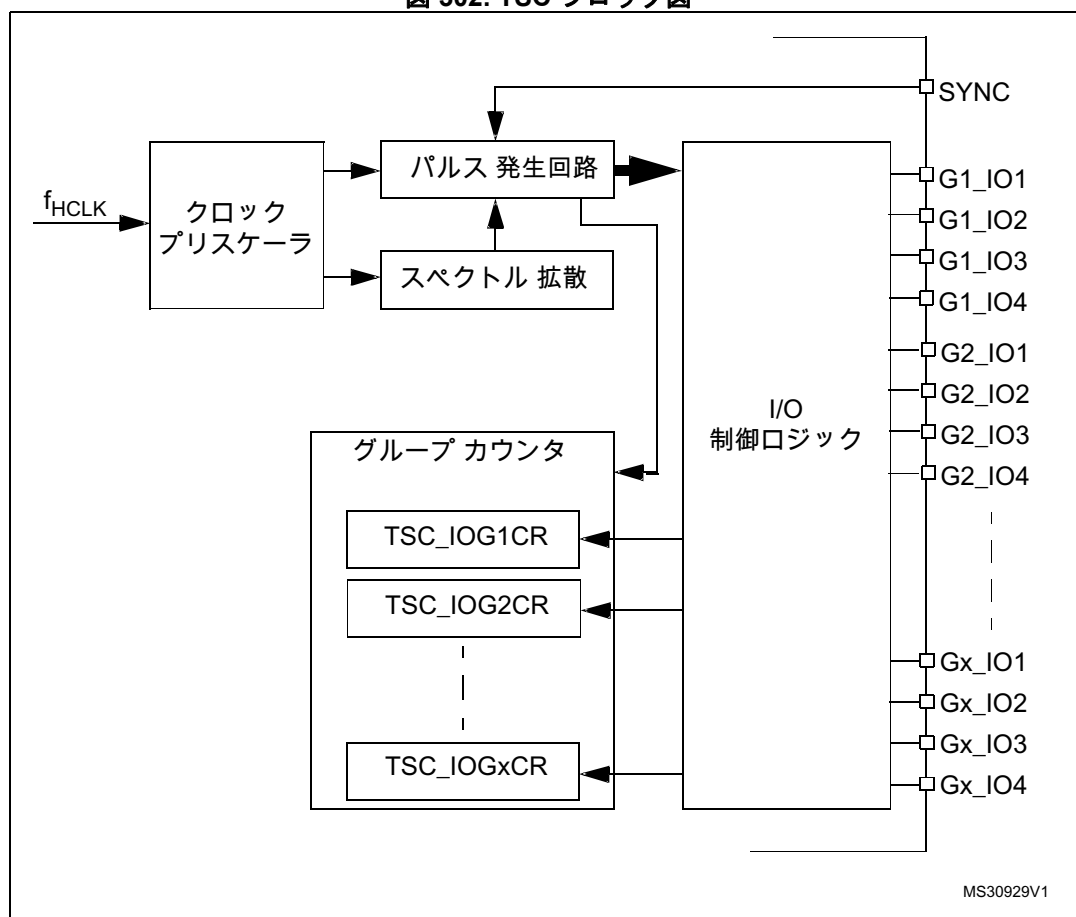
注： 静電容量検出チャンネルの本数は、パッケージのサイズ、および使用可能な I/O があるかどうかによって決まります。

28.3 TSC の機能説明

28.3.1 TSC ブロック図

タッチセンシングコントローラのブロック図を [図 302 : TSC ブロック図](#) に示します。

図 302. TSC ブロック図



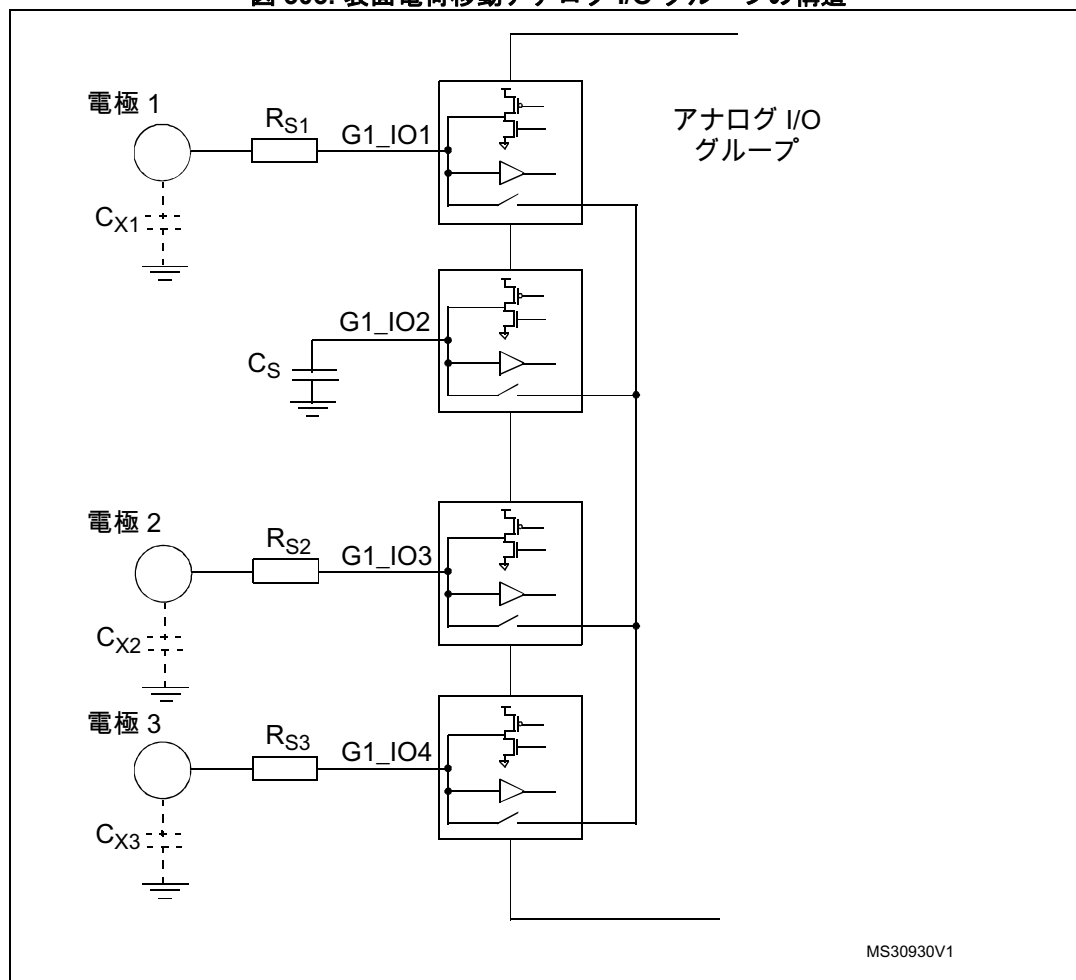
28.3.2 表面電荷移動取得の概要

表面電荷移動取得は、静電容量を計測するための実績のある、安定した効果的な方法です。この方法では、シングルエンド電極型で動作するために最低限必要な数の外部コンポーネントが使用されます。この取得は、4 つの GPIO から成るアナログ I/O グループを中心に設計されています ([図 303](#) を参照)。いくつかのアナログ I/O グループを使用することができ、それにより複数の静電容量検出チャネルを同時に取得し、サポートできる静電容量検出チャネルの数を増やすことができます。1 つのアナログ I/O グループ内で行われる静電容量検出チャネルの取得はシーケンシャルです。

GPIO のうちの 1 つは、サンプリングコンデンサ C_S 専用です。1 つのアナログ I/O グループでは、一度に 1 つのサンプリングコンデンサ I/O だけを有効にしてください。

残りの GPIO は電極に使用され、一般にチャネルと呼ばれます。特定のニーズ（近接検出など）については、1 つのアナログ I/O グループで同時に 1 つ以上のチャネルを有効にすることができます。

図 303. 表面電荷移動アナログ I/O グループの構造



MS30930V1

注： Gx_IOy の x はアナログ I/O グループの番号、 y は選択したグループ内での GPIO の番号を表します。

表面電荷移動取得の基本動作は、電極の静電容量 (C_X) を充電することと、蓄積された電荷の一部をサンプリングコンデンサ (C_S) へ移動することです。このシーケンスは、 C_S の電圧が所定の閾値（弊社の場合、 V_{IH} ）に達するまで繰り返されます。閾値に達するために必要な電荷移動の回数は、電極の静電容量の大きさで直接表されます。

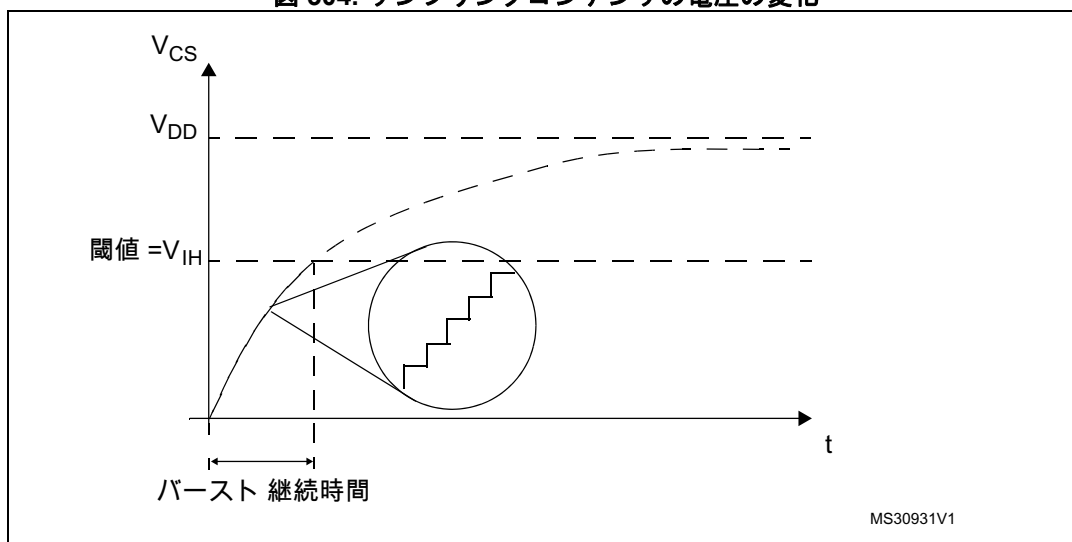
表 108 に、静電容量検出チャネル 1 の電荷移動取得シーケンスの詳細を示します。 C_S の電圧が所定の閾値に達するまで、状態 3～7 が繰り返されます。その他のチャネルの取得にも同じシーケンスが適用されます。電極のシリアルレジスタ R_S によって、この方法の ESD（静電気放電）イミュニティが向上します。

表 108. 取得シーケンスの概要

状態	G1_IO1 (電極)	G1_IO2 (サンプリング)	G1_IO3 (電極)	G1_IO4 (電極)	状態の説明
1	入力 フローティング (アナログ スイッチ閉)	出力オープン ドレインロー (アナログ スイッチ閉)	入力フローティング (アナログスイッチ閉)		すべての C _X および C _S を放電。
2	入力フローティング				デッドタイム
3	出力プッシュ プルハイ	入力フローティング			C _{X1} を充電。
4	入力フローティング				デッドタイム
5	入力フローティング (アナログスイッチ閉)		入力フローティング		C _{X1} から C _S への 電荷移動
6	入力フローティング				デッドタイム
7	入力フローティング				C _S の電圧を測定。

サンプリングコンデンサ C_S の電圧の経時変化を以下に示します。

図 304. サンプリングコンデンサの電圧の変化



28.3.3 リセットおよびクロック

TSC クロックソースは AHB クロック (HCLK) です。2 個のプログラム可能なプリスケアラを使用して、パルス発生回路およびスペクトル拡散内部クロックを生成します。

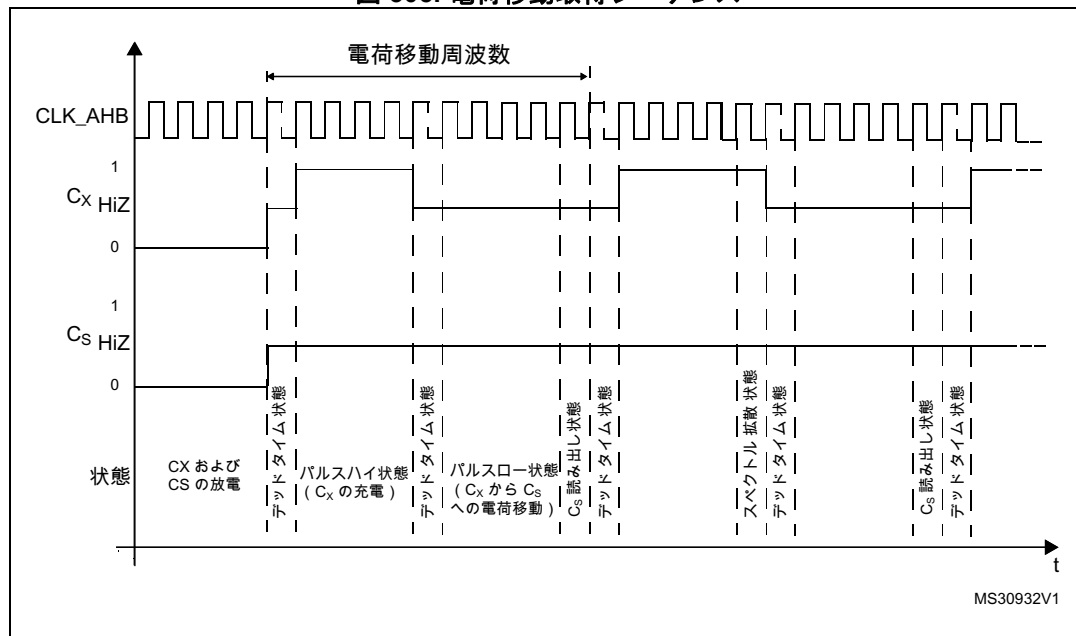
- パルス発生回路クロック (PGCLK) は、TSC_CR レジスタの PGPSC[2:0] ビットを使用して定義されます。
- スペクトル拡散クロック (SSCLK) は、TSC_CR レジスタの SSPSC ビットを使用して定義されます。

リセットおよびクロックコントローラ (RCC) は、タッチセンシングコントローラのクロックを有効にし、このペリフェラルをリセットするための専用ビットを提供します。詳細については、[セクション 7: リセットおよびクロック制御 \(RCC\)](#) を参照してください。

28.3.4 電荷移動取得シーケンス

電荷移動取得シーケンスの例を [図 305](#) に示します。

図 305. 電荷移動取得シーケンス



より高い柔軟性を得るために、電荷移動周波数は完全に設定可能です。パルスのハイ状態 (C_X の充電) およびパルスのロー状態 (C_X から C_S への電荷の移動) のどちらも、継続時間は TSC_CR レジスタの CTPH[3:0] ビットおよび CTPL[3:0] ビットを使用して定義することができます。パルスのハイ状態およびロー状態の継続時間の標準範囲は、500 ns ~ 2 μ s です。電極の静電容量を正確に測定できるようにするには、 C_X が常にフル充電となるように、パルスのハイ状態の継続時間を設定する必要があります。

電荷移動取得シーケンスを最適化するため、パルスのハイ状態とロー状態の間にデッドタイムが挿入されます。デッドタイムとは、サンプリングコンデンサの I/O とチャネルの I/O の両方が入力フローティング状態にある時間のことです。この状態の継続時間は HCLK の 2 周期です。

スペクトル拡散機能が有効であれば、パルスのハイ状態の最後に SSCLK クロックの周期の変数が追加されます。

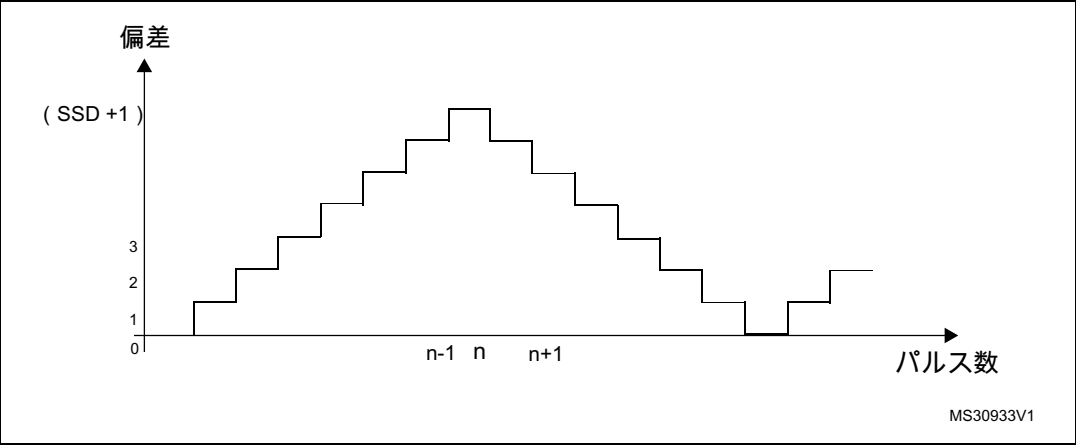
サンプリングコンデンサの I/O の読み出しは、 C_S の電圧が所定の閾値に達したかどうかを確認するために、パルスのロー状態の最後に行われ、その継続時間は HCLK の 1 周期分です。

28.3.5 スペクトル拡散機能

スペクトル拡散機能により、電荷移動周波数の変動を生成することができます。これは、ノイズの多い環境において電荷移動取得の安定性を向上させ、さらに誘導放出を低減するために行われます。周波数の最大変動幅は、公称電荷移動周期の 10~50% です。たとえば、公称電荷移動周波数が 250 kHz (4 μ s) の場合、典型的なスペクトル拡散偏差は 10% (400 ns) であり、この値から最小電荷移動周波数 (227 kHz 以下) が導き出されます。

実際には、スペクトル拡散とは、以下に示す原理を用いて、SSCLK 周期の変数を パルスのハイ状態に追加することを言います。

図 306. スペクトル拡散の変動の原理



次の表に、異なるHCLK 設定での最大周波数偏差を示します。

表 109. スペクトル拡散偏差と AHB クロック周波数

f_{HCLK}	スペクトル拡散の刻み	スペクトル拡散最大偏差
24 MHz	41.6 ns	10666.6 ns
48 MHz	20.8 ns	5333.3 ns

スペクトル拡散機能は、TSC_CR レジスタの SSE ビットを使用して有効／無効にできます。周波数偏差は、TSC_CR レジスタの SSPSC および SSD[6:0] ビットを介して、デバイス HCLK クロック周波数および選択された電荷移動周波数を収容できるように設定することも可能です。

28.3.6 最大カウントエラー

最大カウントエラーは、静電容量検出チャネルの障害により取得時間が長くなるのを防ぎます。アナログ I/O グループのカウンタに対して最大カウント値を指定するのがこのエラーです。この最大カウント値は TSC_CR レジスタの MCV[2:0] ビットを使用して指定されます。取得グループのカウンタがこの最大値に到達するとすぐに、処理中の取得は停止し、取得完了 (EOAF ビット) および最大カウントエラー (MCEF ビット) の両方のフラグがセットされます。対応する取得完了 (EOAIE ビット) や最大カウントエラー (MCEIE ビット) の割り込みイネーブルビットがセットされている場合は、割り込みも生成することができます。

28.3.7 サンプリングコンデンサ I/O および チャネル I/O のモード選択

タッチセンシングコントローラによる GPIO の制御を可能にするには、標準の GPIO レジスタおよび GPIOxAFR レジスタを介して、対応するオルタネート機能を有効にする必要があります。

TSC によって制御される GPIO モードは、TSC_IOSCR および TSC_IOCCR レジスタを使用して定義されます。

処理中の取得がない場合、タッチセンシングコントローラによって制御されるすべての I/O はデフォルト状態です。取得の処理中は、未使用 I/O（サンプリングコンデンサ I/O としても、チャネル I/O としても定義されない）のみがデフォルト状態です。TSC_CR レジスタの IODEF ビットは、デフォルト状態にある I/O の設定を定義します。次の表にモードに応じた I/O の設定を示します。

表 110. モードおよび IODEF ビットの値に応じた I/O の状態

IODEF ビット	取得ステータス	未使用 I/O モード	電極 I/O モード	サンプリング コンデンサ I/O モード
0 (出力プッシュプル ロー)	なし	出力プッシュプル ロー	出力プッシュプル ロー	出力プッシュプル ロー
0 (出力プッシュプル ロー)	処理中	出力プッシュプル ロー	-	-
1 (入力 フローティング)	なし	入力フローティング	入力フローティング	入力フローティング
1 (入力 フローティング)	処理中	入力フローティング	-	-

未使用 I/O モード

未使用 I/O は、TSC ペリフェラルによって制御される GPIO に対応しますが、電極 I/O としても、サンプリングコンデンサ I/O としても定義されません。

サンプリングコンデンサ I/O モード

TSC ペリフェラルによるサンプリングコンデンサ I/O の制御を可能にするには、まず対応する GPIO をオルタネート出力オープンドレインモードにセットし、次に TSC_IOSCR レジスタ の対応する Gx_IOy ビットをセットする必要があります。

1 つのアナログ I/O グループでは、一度に 1 つのサンプリングコンデンサのみを有効にしてください。

チャネル I/O モード

TSC ペリフェラルによるチャネル I/O の制御を可能にするには、まず対応する GPIO をオルタネート出力プッシュプルモードにセットし、次に TSC_IOCCR レジスタ の対応する Gx_IOy ビットをセットする必要があります。

より高い等価電極表面が要求される、あるいは取得プロセスを高速化する近接検出では、同じアナログ I/O グループに属するいくつかのチャネルを有効にし、同時に取得することが可能です。

注： **取得フェーズの間、および TSC ペリフェラルのオルタネート機能が有効でない場合でも、TSC_IOSCR または TSC_IOCCR ビットがセットされたらすぐに、タッチセンシングコントローラによって、対応する GPIO アナログスイッチが自動的に制御されます。**

28.3.8 取得モード

タッチセンシングコントローラには 2 種類の取得モードがあります。

- 通常の取得モード: TSC_CR レジスタの START ビットがセットされるとすぐに取得を開始します。
- 同期取得モード: TSC_CR レジスタの START ビットをセットすることで取得が有効になりますが、取得の開始は、立ち下がり／立ち上がりエッジおよび SYNC 入力ピンのハイレベルが検出されたときのみです。このモードは、CPU 負荷を追加することなく、静電容量検出チャンネルの取得を外部信号に同期させる場合に役立ちます。

TSC_ILOGCSR レジスタの GxE ビットで、どの I/O グループを有効 (対応するカウンタがカウントされる状態) にするかを指定します。無効なアナログ I/O グループの C_S 電圧は監視されず、このグループは取得完了フラグのトリガに加わりません。ただし、無効なアナログ I/O グループにいくつかのチャンネルが含まれる場合は、それらはパルスを出力します。

有効なアナログ I/O グループの C_S 電圧が所定の閾値に達すると、TSC_ILOGCSR レジスタの対応する GxS ビットがセットされます。すべての有効なアナログ I/O グループの取得が完了した (すべての有効なアナログ I/O グループのすべての GxS ビットがセットされた) 時点で、TSC_ISR レジスタの EOAF フラグがセットされます。TSC_IER レジスタの EOAIIE ビットがセットされている場合、割り込みリクエストが生成されます。

最大カウントエラーが検出された場合、処理中の取得は停止し、TSC_ISR レジスタの EOAF フラグと MCEIF フラグの両方がセットされます。対応するビット (TSCIER レジスタの EOAIIE および MCEIE ビット) がセットされていれば、両方のイベントに対して割り込みリクエストを生成することができます。最大カウントエラーが検出された場合、有効なアナログ I/O グループの残りの GxS ビットはセットされませんので注意してください。

割り込みフラグをクリアするには、TSC_ICR レジスタの対応する EOAIIC ビットと MCEIC ビットをセットする必要があります。

アナログ I/O グループのカウントは、新しい取得が開始された時点でクリアされます。カウンタは、取得完了と同時に、対応するチャンネルで生成された電荷移動サイクルの数に更新されます。

コード例については、「付録」セクション [A.18.1 : TSC 設定のコード例](#) を参照してください。

28.3.9 I/O ヒステリシスおよびアナログスイッチの制御

より高い柔軟性を提供するために、タッチセンシングコントローラを使用することで、各 Gx_IOy のシュミットトリガヒステリシスおよびアナログスイッチを制御することもできます。タッチセンシングコントローラが有効であると仮定すれば、I/O 制御モードの状態 (標準の GPIO レジスタまたはその他のペリフェラルによって制御されている) にかかわらず、この制御は実行可能です。この制御は、異なる取得シーケンスを実行したり、他の目的のために実行する際に役立つことがあります。

システムの誤検出を防ぐ目的で、TSC_IOHCR レジスタの対応する Gx_IOy ビットをリセットすることによって、TSC によって制御される GPIO のシュミットトリガヒステリシスを無効にする必要があります。

28.3.10 静電容量検出 GPIO

次の表に、静電容量検出 GPIO の概要を示します。

表 111. 静電容量検出 GPIO

グループ	静電容量検出グループ名	ピン名	グループ	静電容量検出グループ名	ピン名
1	TSC_G1_IO1	PA0	5	TSC_G5_IO1	PB3
	TSC_G1_IO2	PA1		TSC_G5_IO2	PB4
	TSC_G1_IO3	PA2		TSC_G5_IO3	PB6
	TSC_G1_IO4	PA3		TSC_G5_IO4	PB7
2	TSC_G2_IO1	PA4	6	TSC_G6_IO1	PB11
	TSC_G2_IO2	PA5		TSC_G6_IO2	PB12
	TSC_G2_IO3	PA6		TSC_G6_IO3	PB13
	TSC_G2_IO4	PA7		TSC_G6_IO4	PB14
3	TSC_G3_IO1	PC5	7	TSC_G7_IO1	PE2
	TSC_G3_IO2	PB0		TSC_G7_IO2	PE3
	TSC_G3_IO3	PB1		TSC_G7_IO3	PE4
	TSC_G3_IO4	PB2		TSC_G7_IO4	PE5
4	TSC_G4_IO1	PA9	8	TSC_G8_IO1	PD12
	TSC_G4_IO2	PA10		TSC_G8_IO2	PD13
	TSC_G4_IO3	PA13		TSC_G8_IO3	PD14
	TSC_G4_IO4	PA14		TSC_G8_IO4	PD15

28.4 TSC 低電力モード

表 112. 低電力モードが TSC に与える影響

モード	説明
SLEEP	影響なし。 TSC 割り込みによって、デバイスは SLEEP モードを終了します。
STOP	TSC レジスタは停止状態です。
STANDBY	STOP または STANDBY モードを終了するまで、TSC は動作を停止します。

28.5 TSC 割り込み

表 113. 割り込み制御ビット

割り込みイベント	有効化 制御ビット	イベント フラグ	フラッグ ビットのクリア	SLEEP モードの終了	STOP モードの終了	STANDBY モードの終了
取得完了	EOAIE	EOAIF	EOAIC	あり	なし	なし
最大カウントエラー	MCEIE	MCEIF	MCEIC	あり	なし	なし

コード例については、「付録」セクション [A.18.2 : TSC 割り込みのコード例](#)を参照してください。



28.6 TSC レジスタ

レジスタの説明で使用されている略語のリストについては、リファレンスマニュアルの [セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスすることができます。

28.6.1 TSC 制御レジスタ (TSC_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTPH[3:0]				CTPL[3:0]				SSD[6:0]						SSE	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSPSC		PGPSC[2:0]		Res.	Res.	Res.	Res.	MCV[2:0]			IODEF	SYNC POL	AM	START	TSCE
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:28 **CTPH[3:0]** : 電荷移動パルスハイ

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、電荷移動パルスのハイ状態 (C_Xの充電) の継続時間を定義します。

0000 : 1x t_{PGCLK}

0001 : 2x t_{PGCLK}

...

1111 : 16x t_{PGCLK}

注 : 取得処理中はこれらのビットを変更しないでください。

ビット 27:24 **CTPL[3:0]** : 電荷移動パルスロー

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、電荷移動パルスのロー状態 (C_X から C_Sへの電荷の移動) の継続時間を定義します。

0000 : 1x t_{PGCLK}

0001 : 2x t_{PGCLK}

...

1111 : 16x t_{PGCLK}

注 : 取得処理中はこれらのビットを変更しないでください。

ビット 23:17 **SSD[6:0]** : スペクトル拡散偏差

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、SSCLK クロックの周期の変数を電荷移動パルスのハイ状態に追加する際のスペクトル拡散偏差を定義します。

0000000 : 1x t_{SSCLK}

0000001 : 2x t_{SSCLK}

...

1111111 : 128x t_{SSCLK}

注 : 取得処理中はこれらのビットを変更しないでください。

ビット 16 **SSE** : スペクトル拡散イネーブル

このビットは、スペクトル拡散機能を有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : スペクトル拡散機能は無効です。

1 : スペクトル拡散機能は有効です。

注 : 取得処理中はこのビットを変更しないでください。

ビット 15 SSPSC : スペクトル拡散プリスケアラ

このビットは、ソフトウェアによってセット／クリアされます。このビットは、スペクトル拡散クロック (SSCLK) の生成に使用される AHB クロック分周器を選択します。

- 0 : f_{HCLK}
- 1 : $f_{HCLK} / 2$

注 : 取得処理中はこのビットを変更しないでください。

ビット 14:12 PGPSC[2:0] : パルス生成回路プリスケアラ

これらのビットはソフトウェアによってセット／クリアされます。これらのビットは、パルス発生回路クロック (PGCLK) の生成に使用される AHB クロック分周器を選択します。

- 000 : f_{HCLK}
- 001 : $f_{HCLK} / 2$
- 010 : $f_{HCLK} / 4$
- 011 : $f_{HCLK} / 8$
- 100 : $f_{HCLK} / 16$
- 101 : $f_{HCLK} / 32$
- 110 : $f_{HCLK} / 64$
- 111 : $f_{HCLK} / 128$

注 : 取得処理中はこれらのビットを変更しないでください。

ビット 11:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:5 MCV[2:0] : 最大カウント値

これらのビットは、ソフトウェアによってセット／クリアされます。これらのビットは、最大カウントエラーが生成される前に生成され得る電荷移動パルスの最大数を定義します。

- 000 : 255
- 001 : 511
- 010 : 1023
- 011 : 2047
- 100 : 4095
- 101 : 8191
- 110 : 16383
- 111 : 予約済み

注 : 取得処理中はこれらのビットを変更しないでください。

ビット 4 IODEF : I/O デフォルトモード

このビットは、ソフトウェアによってセット／クリアされます。このビットは、処理中の取得がない場合に、すべての TSC I/O の設定を定義します。処理中の取得がある場合、このビットは、すべての未使用 I/O (サンプリングコンデンサ I/O やチャネル I/O として定義されていない) の設定を定義します。

- 0 : I/O は強制的に出力プッシュプルローにされます。
- 1 : I/O は入力フローティング状態です。

注 : 取得処理中はこのビットを変更しないでください。

ビット 3 SYNCPOL : 同期ピンの極性

このビットは、同期入力ピンの極性を選択するために、ソフトウェアによってセット／クリアされます。

- 0 : 立ち下がりエッジのみ
- 1 : 立ち上がりエッジおよびハイレベル

ビット 2 AM : 取得モード

このビットは、取得モードを選択するために、ソフトウェアによってセット／クリアされます。

0 : 通常の取得モード (START ビットがセットされた直後に取得を開始)

1 : 同期取得モード (START ビットがセットされ、かつ選択された信号が SYNC 入力ピンで検出された場合に、取得を開始)

注 : 取得処理中はこのビットを変更しないでください。

ビット 1 START : 新しい取得を開始します。

このビットは、新しい取得を開始するために、ソフトウェアによってセットされます。このビットは、取得が完了した直後にハードウェアによって、または処理中の取得をキャンセルするためにソフトウェアによってクリアされます。

0 : 取得は開始されていません。

1 : 新しい取得を開始します。

ビット 0 TSCE : タッチセンシングコントローライネーブル

このビットは、タッチセンシングコントローラを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : タッチセンシングコントローラは無効です。

1 : タッチセンシングコントローラは有効です。

注 : タッチセンシングコントローラが無効の場合、TSC レジスタの設定は無効です。

28.6.2 TSC 割り込み有効レジスタ (TSC_IER)

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **MCEIE** : 最大カウントエラー割り込みイネーブル

このビットは、最大カウントエラーの割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 最大カウントエラー割り込みは無効です。

1 : 最大カウントエラー割り込みは有効です。

ビット 0 **EOAIE** : 取得完了割り込みイネーブル

このビットは、取得完了割り込みを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : 取得完了割り込みは無効です。

1 : 取得完了割り込みは有効です。

28.6.3 TSC 割り込みクリアレジスタ (TSC_ICR)

アドレスオフセット : 0x08

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **MCEIC** : 最大カウントエラー割り込みクリア

このビットは、最大カウントエラーフラグをクリアするためにソフトウェアによってセットされ、フラグがリセットされた時点でハードウェアによってクリアされます。“0”を書き込んでも、ビットの値は変化しません。

0 : 影響なし。

1 : TSC_ISR レジスタの対応する MCEF をクリアします。

ビット 0 **EOAIC** : 取得完了割り込みクリア

このビットは、取得完了フラグをクリアするためにソフトウェアによってセットされ、フラグがリセットされた時点でハードウェアによってクリアされます。“0”を書き込んでも、ビットの値は変化しません。

0 : 影響なし。

1 : TSC_ISR レジスタの対応する EOAF をクリアします。

28.6.4 TSC 割り込みステータスレジスタ (TSC_ISR)

アドレスオフセット : 0x0C

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEF	EOAF
														rw	rw

ビット 31:2 予約済みであり、リセット値に保持する必要があります。

ビット 1 **MCEF** : 最大カウントエラーフラグ

このビットは、アナログ I/O グループのカウントが指定された最大カウント値に達すると、ハードウェアによってセットされます。このビットは、ソフトウェアで TSC_ICR レジスタの MCEIC ビットに“1”を書き込むことによってクリアされます。

- 0 : 最大カウントエラー (MCE) は検出されていません。
- 1 : 最大カウントエラー (MCE) が検出されました。

ビット 0 **EOAF** : 取得完了フラグ

このビットは、すべての有効なグループの取得が完了した (すべての有効なアナログ I/O グループのすべての GxS ビットがセットされた、または最大カウントエラーが検出された) 時点で、ハードウェアによってセットされます。このビットは、ソフトウェアで TSC_ICR レジスタの EOAIIC ビットに“1”を書き込むことによってクリアされます。

- 0 : 取得は処理中または開始されていません。
- 1 : 取得は完了しました。

28.6.5 TSC I/O ヒステリシス制御レジスタ (TSC_IOHCR)

アドレスオフセット : 0x10

リセット値 : 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **Gx_IOy** : Gx_IOy シュミットトリガヒステリシスモード

これらのビットは、Gx_IOy シュミットトリガヒステリシスを有効/無効にするために、ソフトウェアによってセット/クリアされます。

- 0 : Gx_IOy シュミットトリガヒステリシスは無効です。
- 1 : Gx_IOy シュミットトリガヒステリシスは有効です。

注 : これらのビットは、I/O 制御モードの状態にかかわらず (標準の GPIO レジスタによって制御されていても)、I/O シュミットトリガヒステリシスを制御します。

28.6.6 TSC I/O アナログスイッチ制御レジスタ (TSC_IOASCR)

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **Gx_IOy** : Gx_IOy アナログスイッチイネーブル

これらのビットは、Gx_IOy アナログスイッチを有効／無効にするために、ソフトウェアによってセット／クリアされます。

0 : Gx_IOy アナログスイッチは無効（開）です。

1 : Gx_IOy アナログスイッチは有効（閉）です。

注 : これらのビットは、I/O 制御モードの状態にかかわらず（標準の GPIO レジスタによって制御されていても）、I/O アナログスイッチを制御します。

28.6.7 TSC I/O サンプリング制御レジスタ (TSC_IOSCR)

アドレスオフセット : 0x20

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **Gx_IOy** : Gx_IOy サンプリングモード

これらのビットは、Gx_IOy をサンプリングコンデンサ I/O として設定するために、ソフトウェアによってセット／クリアされます。1 つのアナログ I/O グループでは、1 つの I/O だけをサンプリングコンデンサとして定義してください。

0 : Gx_IOy は未使用です。

1 : Gx_IOy はサンプリングコンデンサとして使用されます。

注 : 取得処理中はこれらのビットを変更しないでください。

取得フェーズの間、および TSC ペリフェラルのオルタネート機能が有効でない場合でも、TSC_IOSCR ビットがセットされたらすぐに、タッチセンシングコントローラによって、対応する GPIO アナログスイッチが自動的に制御されます。

28.6.8 TSC I/O チャンネル制御レジスタ (TSC_IOC CRTSC_IOC CR)

アドレスオフセット : 0x28

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31:0 **Gx_IOy** : Gx_IOy チャンネルモード

これらのビットは、Gx_IOy をチャンネル I/O として設定するために、ソフトウェアによってセット／クリアされます。

0 : Gx_IOy は未使用です。

1 : Gx_IOy はチャンネルとして使用されます。

注： 取得処理中はこれらのビットを変更しないでください。

取得フェーズの間、およびTSC ペリフェラルのオルタネート機能が有効でない場合でも、TSC_IOC CR ビットがセットされたらすぐに、タッチセンシングコントローラによって、対応する GPIO アナログスイッチが自動的に制御されます。

28.6.9 TSC I/O グループ制御ステータスレジスタ (TSC_ILOGCSR)

アドレスオフセット : 0x30

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8S	G7S	G6S	G5S	G4S	G3S	G2S	G1S
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8E	G7E	G6E	G5E	G4E	G3E	G2E	G1E
								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:24 予約済みであり、リセット値に保持する必要があります。

ビット 23:16 **GxS** : アナログ I/O グループ x ステータス

これらのビットは、対応する有効なアナログ I/O グループ x で取得が完了した時点で、ハードウェアによってセットされます。これらのビットは、新しい取得が開始した時点で、ハードウェアによってクリアされます。

0 : アナログ I/O グループ x での取得は、処理中または開始されていません。

1 : アナログ I/O グループ x での取得は完了しました。

注 : 最大カウントエラーが検出された場合、有効なアナログ I/O グループの残りの GxS ビットはセットされません。

ビット 15:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **GxE** : アナログ I/O グループ x イネーブル

これらのビットは、対応するアナログ I/O グループ x での取得を有効 (カウンタがカウントされる状態) にするために、ソフトウェアによってセット/クリアされます。

0 : アナログ I/O グループ x での取得は無効です。

1 : アナログ I/O グループ x での取得は有効です。

28.6.10 TSC I/O グループ x カウンタレジスタ (TSC_ILOGxCR) (x = 1~8)

アドレスオフセット : 0x30 + 0x04 x アナログ I/O グループ番号

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CNT[13:0]													
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:0 **CNT[13:0]** : カウンタ値

これらのビットは、アナログ I/O グループ x の取得を完了させる (C_S の電圧が閾値に達する) ために、そこで生成された電荷移動サイクルの数を表します。

28.6.11 TSC レジスタマップ

表 114. TSC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x0000	TSC_CR	CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE	SSPSC	PGPSC[2:0]			Res	Res	Res	Res	MCV [2:0]			IODEF	SYNCPOL	AM	START	TSCE	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x0004	TSC_IER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MCEIE	EOAIE	
	リセット値																															0	0	
0x0008	TSC_ICR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MCEIC	EOAIC	
	リセット値																															0	0	
0x000C	TSC_ISR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MCEIF	EOAIF	
	リセット値																															0	0	
0x0010	TSC_IOHCR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1	
	リセット値	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x0014	予約済み																																	
0x0018	TSC_IOASCR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x001C	予約済み																																	
0x0020	TSC_IOSCR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0024	予約済み																																	
0x0028	TSC_IOCRR	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1	
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x002C	予約済み																																	
0x0030	TSC_IQGCSR	Res	Res	Res	Res	Res	Res	Res	Res	G8S	G7S	G6S	G5S	G4S	G3S	G2S	G1S	Res	Res	Res	Res	Res	Res	Res	Res	G8E	G7E	G6E	G5E	G4E	G3E	G2E	G1E	
	リセット値									0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0	
0x0034	TSC_IQG1CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[13:0]														
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0038	TSC_IQG2CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[13:0]														
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x003C	TSC_IQG3CR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[13:0]														
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0	

この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 114. TSC レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0040	TSC_I0G4CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	TSC_I0G5CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	TSC_I0G6CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	TSC_I0G7CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	TSC_I0G8CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	リセット値																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。



29 コントローラエリアネットワーク (bxCAN)

このセクションは、STM32F04x、STM32F072、STM32F078、および STM32F09x デバイスにのみ適用されます。

29.1 概要

Basic Extended CAN ペリフェラル (略称 **bxCAN**) は CAN ネットワークのインタフェースです。CAN プロトコルバージョン 2.0A および B をサポートしています。小さな CPU 負荷で多くの受信メッセージを効率よく取り扱えるように設計されています。また、送信メッセージの優先順位に関する要件も満たしています。

安全性が重視されるアプリケーションの場合、CAN コントローラは、CAN タイムトリガ通信オプションをサポートするすべてのハードウェア機能を提供します。

29.2 bxCAN の主な機能

- CAN プロトコルバージョン 2.0 A、B Active のサポート
- ビットレート：最大 1 Mbit/s
- タイムトリガ通信オプションのサポート

送信

- 3 つの送信メールボックス
- 設定可能な送信優先順位
- SOF 送信時のタイムスタンプ

受信

- 2 つの 3 ステージ受信 FIFO
- スケーラブルフィルタバンク：
 - 14 個のフィルタバンク
- 識別子リスト機能
- 設定可能な FIFO オーバーラン
- SOF 受信時のタイムスタンプ

タイムトリガ通信オプション

- 自動再送信モードの無効化
- 16 ビットのフリーランニングタイマ
- タイムスタンプを最後の 2 データバイトで送信

管理

- マスク可能な割り込み
- 固有のアドレス空間でのソフトウェア効率の高いメールボックス配置

29.3 bxCAN の概要

今日の CAN アプリケーションでは、ネットワークのノード数が増加し、複数のネットワークがゲートウェイを介してリンクされることもあります。一般に、システム内の（そして各ノードが扱う）メッセージ数は大幅に増加しています。アプリケーションメッセージに加えて、ネットワーク管理や診断のメッセージも導入されています。

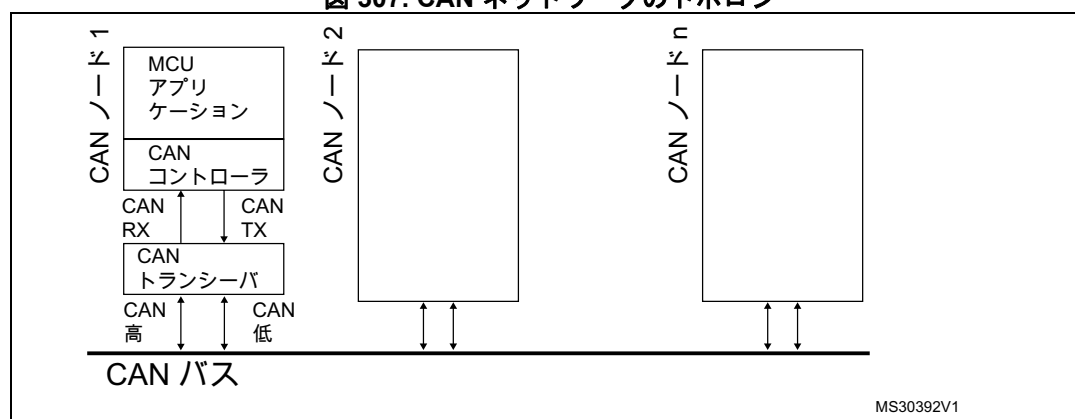
- 各種のメッセージを扱うには、高度なフィルタ機構が要求されます。

さらに、アプリケーションタスクはより多くの CPU 時間を必要とするため、メッセージ受信に起因するリアルタイム制約を減らす必要があります。

- 受信 FIFO 方式では、CPU は、メッセージを失うことなく長時間にわたってアプリケーションタスクに専念できます。

標準の CAN ドライバに基づく標準 HLP (Higher Layer Protocol : 上位層プロトコル) は、CAN コントローラとの効率的なインタフェースを必要とします。

図 307. CAN ネットワークのトポロジ



29.3.1 CAN 2.0B アクティブコア

bxCAN モジュールは、CAN メッセージの送受信を完全に自律的に処理します。標準識別子 (11 ビット) と拡張識別子 (29 ビット) は、ハードウェアによって完全にサポートされています。

29.3.2 制御レジスタ、ステータスレジスタ、設定レジスタ

アプリケーションは、これらのレジスタを以下の目的で使用します。

- CAN パラメータ (ボーレートなど) の設定
- 送信リクエスト
- 受信処理
- 割り込み管理
- 診断情報の入手

29.3.3 送信メールボックス

ソフトウェアがメッセージをセットアップするために、3 つの送信メールボックスが用意されています。最初に送信されるメールボックスは、送信スケジューラによって決定されます。

29.3.4 受信フィルタ

bxCAN は、受信メッセージからソフトウェアが必要なものを選び出し、他を破棄するために、28 個のスケラブルで設定可能な識別子フィルタバンクを備えています。

受信 FIFO

ハードウェアは、受信メッセージを格納するために 2 つの受信 FIFO を使用します。各 FIFO には、3 つの完全なメッセージを格納できます。FIFO の管理はすべてハードウェアによって行われます。

29.4 bxCAN 動作モード

bxCAN には、**初期化**、**通常**、および**スリープ**という 3 つの主な動作モードがあります。ハードウェアリセットの後、bxCAN は消費電力を低減するためにスリープモードに入り、CANTX では内部プルアップがアクティブとなります。ソフトウェアは、CAN_MCR レジスタの INRQ または SLEEP ビットをセットすることによって、bxCAN に**初期化**または**スリープモード**に入るように要求します。いずれかのモードに入ると、bxCAN は、CAN_MSR レジスタの INAK または SLAK ビットをセットすることによってそれを確定し、内部プルアップは無効になります。INAK も SLAK もセットされていない場合、bxCAN は**通常モード**です。**通常モード**に入る前に、bxCAN は必ず CAN バスで**同期**する必要があります。同期のために、bxCAN は、CAN バスがアイドル状態になる（つまり、CANRX 上で 11 個の連続したリセツシブビットが観測される）のを待ちます。

29.4.1 初期化モード

ハードウェアが初期化モードにある間に、ソフトウェアの初期化も行うことができます。このモードに入るには、ソフトウェアは CAN_MCR レジスタの INRQ ビットをセットし、ハードウェアが CAN_MSR レジスタの INAK ビットをセットしてこのリクエストを確認するのを待ちます。

初期化モードから出るには、ソフトウェアで INRQ ビットをクリアします。INAK ビットがハードウェアによってクリアされると、bxCAN は初期化モードを終了します。

初期化モードでは、CAN バスとの間のすべてのメッセージ転送は停止し、CAN バス出力 CANTX のステータスはリセツシブ（ハイ）になります。

初期化モードに入っても、設定レジスタの内容は一切変化しません。

CAN コントローラを初期化するには、ソフトウェアでビットタイミング (CAN_BTR) および CAN オプション (CAN_MCR) レジスタの設定を行う必要があります。

CAN フィルタバンクに関連付けられたレジスタ (モード、スケール、FIFO 割当て、アクティベーション、およびフィルタ値) を初期化するには、ソフトウェアが FINIT ビット (CAN_FMR) をセットする必要があります。フィルタの初期化は、初期化モード以外でも行うことができます。

注： **FINIT=1 のとき、CAN 受信は無効化されます。**

フィルタの値は、CAN_FA1R レジスタの関連するフィルタアクティベーションビットを無効にすることによっても変更できます。

フィルタバンクを使用しない場合は、これを非アクティブのままにしておく（対応する FACT ビットをクリアしておく）ことを推奨します。

コード例については、「付録」セクション [A.11.1 : bxCAN 初期化モードのコード例](#) を参照してください。

29.4.2 通常モード

初期化が終了すると、ソフトウェアはハードウェアに対し、CAN バスで同期して送受信を開始できるように通常モードに入るように要求する必要があります。

この通常モードに入るリクエストは、CAN_MCR レジスタの INRQ ビットをクリアすることによって行われます。bxCAN は通常モードに入り、11 個の連続したリセッショビットの発生（バスアイドル状態）を待って CAN バス上のデータ転送と同期すると、バス動作への参加準備完了となります。通常モードへの切り替えは、CAN_MSR レジスタの INAK ビットをクリアすることによって、ハードウェアによって確認されます。

フィルタ値の初期化は初期化モードとは別に行いますが、フィルタがアクティブでない（対応する FACTx ビットがクリアされている）ときに行う必要があります。フィルタのスケールとモードの設定は、通常モードに入る前に行う必要があります。

29.4.3 スリープモード（低電力）

消費電力を低減するため、bxCAN にはスリープモードと呼ばれる低電力モードがあります。ソフトウェアのリクエストで CAN_MCR レジスタの SLEEP ビットをセットすると、このモードに入ります。このモードでは bxCAN のクロックは停止しますが、ソフトウェアから bxCAN のメールボックスにアクセスすることは可能です。

bxCAN がスリープモードのときにソフトウェアが INRQ ビットをセットして初期化モードに入るように要求する場合には、SLEEP ビットもクリアする必要があります。

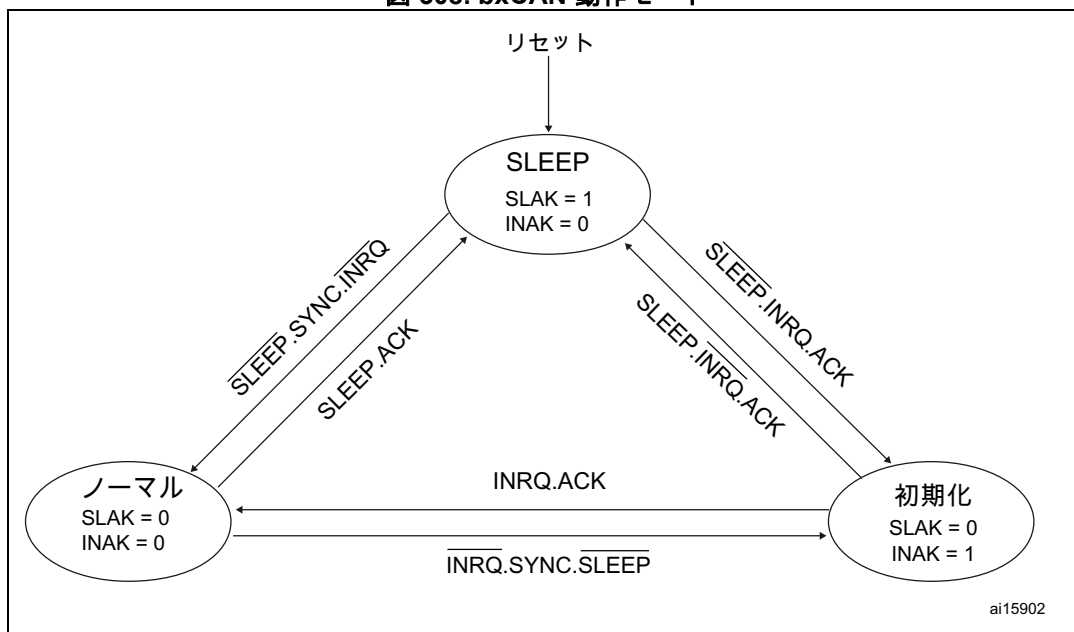
bxCAN がスリープモードからウェイクアップするには、ソフトウェアで SLEEP ビットをクリアするか、CAN バス動作を検出する必要があります。

CAN バス動作を検出すると、ハードウェアは、SLEEP ビットをクリアしてウェイクアップシーケンスを自動的に実行します（CAN_MCR レジスタの AWUM ビットがセットされている場合）。AWUM ビットがクリアされている場合、スリープモードから出るにはウェイクアップ割込み発生時にソフトウェアが SLEEP ビットをクリアする必要があります。

注： **ウェイクアップ割込みが有効である（CAN_IER レジスタの WKUIE ビットがセットされている）場合には、bxCAN がウェイクアップシーケンスを自動的に実行する場合でも、CAN バス動作の検出時にウェイクアップ割込みが生成されます。**

SLEEP ビットがクリアされた後、bxCAN が CAN バスに同期するとスリープモードを終了します（[図 308：bxCAN 動作モード](#)を参照）。SLAK ビットがハードウェアによってクリアされると、スリープモードを終了します。

図 308. bxCAN 動作モード



1. ACK = ハードウェアが CAN_MSR レジスタの INAK または SLAK ビットをセットすることによってリクエストを確認する待ち状態。
2. SYNC = CAN バスがアイドル状態になる（つまり、11 個の連続したリセッシーブビットが CANRX で観測される）までの bxCAN の待ち状態。

29.5 テストモード

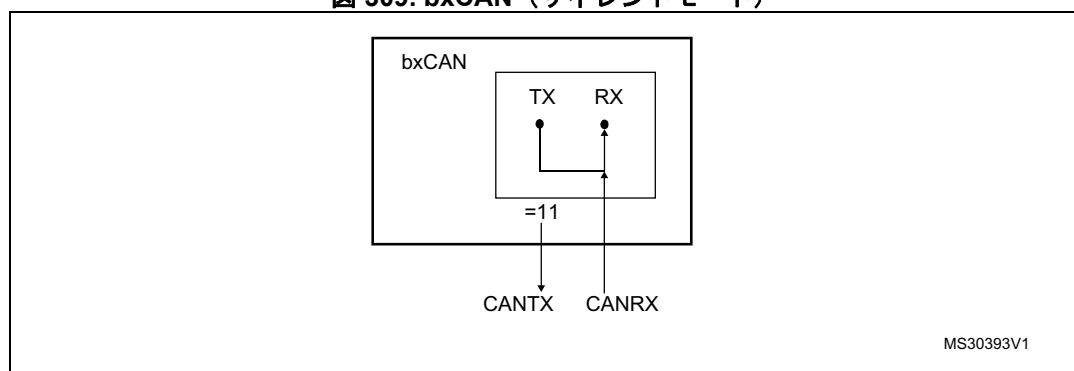
テストモードは、CAN_BTR レジスタの SILM ビットと LBKM ビットによって選択できます。これらのビットは、bxCAN が初期化モードにある間に設定する必要があります。テストモードが選択された後で通常モードに戻るには、CAN_MCR レジスタの INRQ ビットをリセットする必要があります。

29.5.1 サイレントモード

bxCAN をサイレントモードにするには、CAN_BTR レジスタの SILM ビットをセットします。

サイレントモードでは、bxCAN は有効なデータフレームやリモートフレームを受信できますが、CAN バスではリセッシーブビットのみを送信するため、送信を開始することはできません。bxCAN がドミナントビット（ACK ビット、オーバーロードフラグ、アクティブエラーフラグ）を送信する必要がある場合、CAN バスがリセッシーブ状態にとどまることがあっても CAN コアがこのドミナントビットを監視できるように、このビットは内部でルート変更されます。サイレントモードを使用すると、ドミナントビット（ACK ビット、エラーフレーム）の送信による影響を受けずに、CAN バス上のトラフィックを解析できます。

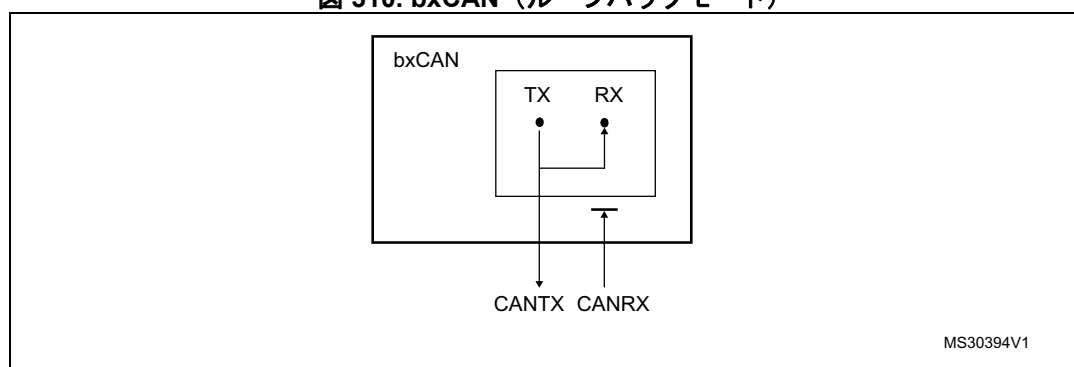
図 309. bxCAN (サイレントモード)



29.5.2 ループバックモード

bxCAN をループバックモードに設定するには、CAN_BTR レジスタの LBKM ビットをセットします。ループバックモードでは、bxCAN は自分が送信したメッセージを受信メッセージとして扱い、それらを受信メールボックスに格納します（受信フィルタを通過した場合）。

図 310. bxCAN (ループバックモード)

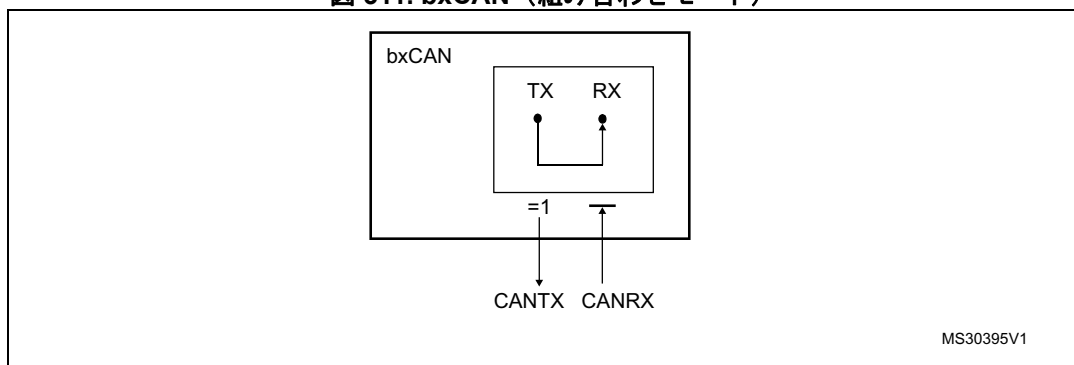


このモードは、セルフテスト機能のために用意されています。外部イベントから独立させるために、ループバックモードでは、CAN コアは ACK エラーを無視します（データ/リモートフレームの ACK スロットではドミナントビットはサンプリングされません）。このモードでは、bxCAN は Tx 出力から Rx 入力へ内部フィードバックを行います。bxCAN は、CANRX 入力ピンの実際の値を無視します。送信メッセージは CANTX ピンで監視できます。

29.5.3 サイレントモードとループバックモードの組み合わせ

CAN_BTR レジスタの LBKM ビットと SILM ビットをセットすることによって、ループバックモードとサイレントモードを組み合わせることもできます。このモードは「ホットセルフテスト」に使用できます。つまり、CANTX および CANRX ピンに接続されている実行中の CAN システムに影響を与えずに、ループバックモードと同様に bxCAN をテストできます。このモードでは、CANRX ピンは bxCAN から切断され、CANTX ピンはリセツピに保持されます。

図 311. bxCAN (組み合わせモード)



29.6 デバッグモードでの挙動

マイクロコントローラがデバッグモードになると (Cortex®-M0 コア停止)、次の状態に応じて、bxCAN は通常どおりに動作を続けるか、または停止します。

- DBG モジュールで CAN1 の場合は DBG_CAN1_STOP ビット、CAN2 の場合は DBG_CAN2_STOP ビット。
- CAN_MCR の DBF ビット。詳細については、[セクション 29.9.2 : CAN 制御/ステータスレジスタ](#)を参照してください。

29.7 bxCAN 機能の説明

29.7.1 送信処理

メッセージを送信するには、アプリケーションは、**EMPTY** 状態の送信メールボックスを 1 つ選択し、識別子、データ長コード (DLC)、およびデータを設定してから、CAN_TxR レジスタの対応する TXRQ ビットをセットすることによって、送信を要求する必要があります。メールボックスが **EMPTY** 状態でなくなると、ソフトウェアはこのメールボックスレジスタへの書き込みができなくなります。TXRQ ビットがセットされると、メールボックスはただちに **PENDING** 状態になり、最高優先順位のメールボックスになるまで待機します。[送信優先順位](#)を参照してください。メールボックスの優先順位が最高になると、メールボックスは送信の **SCHEDULED** 状態になります。スケジュール済みメールボックスのメッセージの送信は、CAN バスがアイドル状態になると開始されます (**TRANSMIT** 状態になります)。メールボックスの内容が正常に送信されると、そのメールボックスは再び **EMPTY** 状態になります。ハードウェアは、CAN_TSR レジスタの RQCP ビットと TXOK ビットをセットして、送信が正常に終了したことを示します。

送信が失敗した場合、その原因がアービトレーションロストである場合は CAN_TSR レジスタの ALST ビットによって、または、送信エラー検出である場合は TERR ビットによって示されます。

コード例については、「付録」セクション [A.11.2 : bxCAN 送信のコード例](#)を参照してください。

送信優先順位

識別子による優先順位

複数の送信メールボックスが保留中のときには、送信順序は、メールボックスに格納されているメッセージの識別子によって決まります。CAN プロトコルのアービトレーションに従って、識別子の値が最小のメッセージの優先順位が最高になります。識別子の値が同じ場合には、メールボックス番号の小さい方が優先的にスケジュールされます。

送信リクエストの順序による優先順位

送信メールボックスは、CAN_MCR レジスタの TXFP ビットをセットすることによって、送信 FIFO として設定できます。このモードでは、優先順位は送信リクエストの順序によって決まります。

このモードは分割送信の場合に非常に役立ちます。

アポート

ユーザは、CAN_TSR レジスタの ABRQ ビットをセットすることによって、送信リクエストをアポートできます。**PENDING** または **SCHEDULED** 状態では、メールボックスはただちにアポートされます。メールボックスが **TRANSMIT** 状態のときにアポートリクエストが出されると、次の 2 つの結果につながります。メールボックスの送信が正常に行われた場合、CAN_TSR レジスタの TXOK ビットがセットされ、メールボックスは **EMPTY** 状態になります。送信に失敗した場合、メールボックスは **SCHEDULED** 状態になり、送信がアポートされ、TXOK ビットがクリアされて、メールボックスは **EMPTY** 状態になります。いずれの場合にも、現在の送信が終了するとメールボックスは最低でも再び **EMPTY** 状態になります。

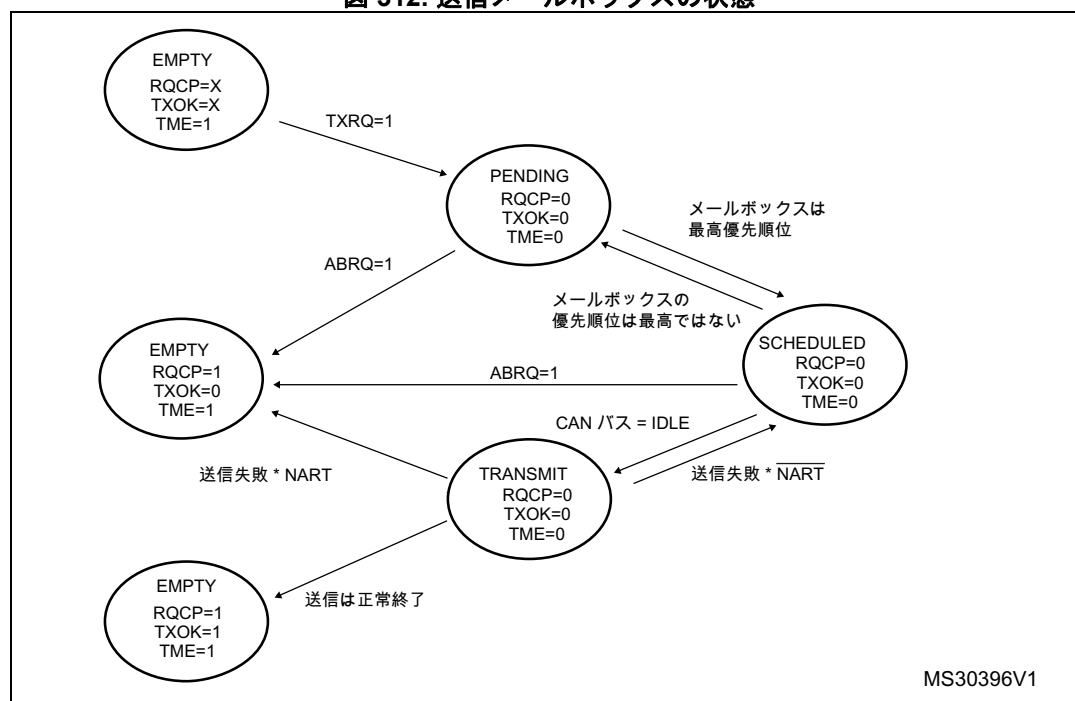
自動再送信禁止モード

このモードは、CAN 規格のタイムトリガ通信オプションの要件を満たすために実装されています。ハードウェアをこのモードに設定するには、CAN_MCR レジスタの NART ビットをセットする必要があります。

このモードでは、個々の送信は一度だけ開始されます。最初の試みがアービトレーションロスやエラーのために失敗した場合、ハードウェアはメッセージ送信を自動的にリスタートしません。

最初の送信試行の終わりに、ハードウェアはそのリクエストが完了したものとみなし、CAN_TSR レジスタの RQCP ビットをセットします。送信の結果は、CAN_TSR レジスタの TXOK、ALST、および TERR ビットによって示されます。

図 312. 送信メールボックスの状態



29.7.2 タイムトリガ通信モード

このモードでは、CAN ハードウェアの内部カウンタがアクティブになり、このカウンタを使用して受信／送信メールボックスの CAN_RDTxR／CAN_TDTxR レジスタに格納されるタイムスタンプ値を生成します。内部カウンタは、CAN ビット時間ごとにインクリメントされます（[セクション 29.7.7: ビットタイミング](#)を参照）。内部カウンタは、送受信ともに、フレーム開始（SOF）ビットのサンプルポイントでキャプチャされます。

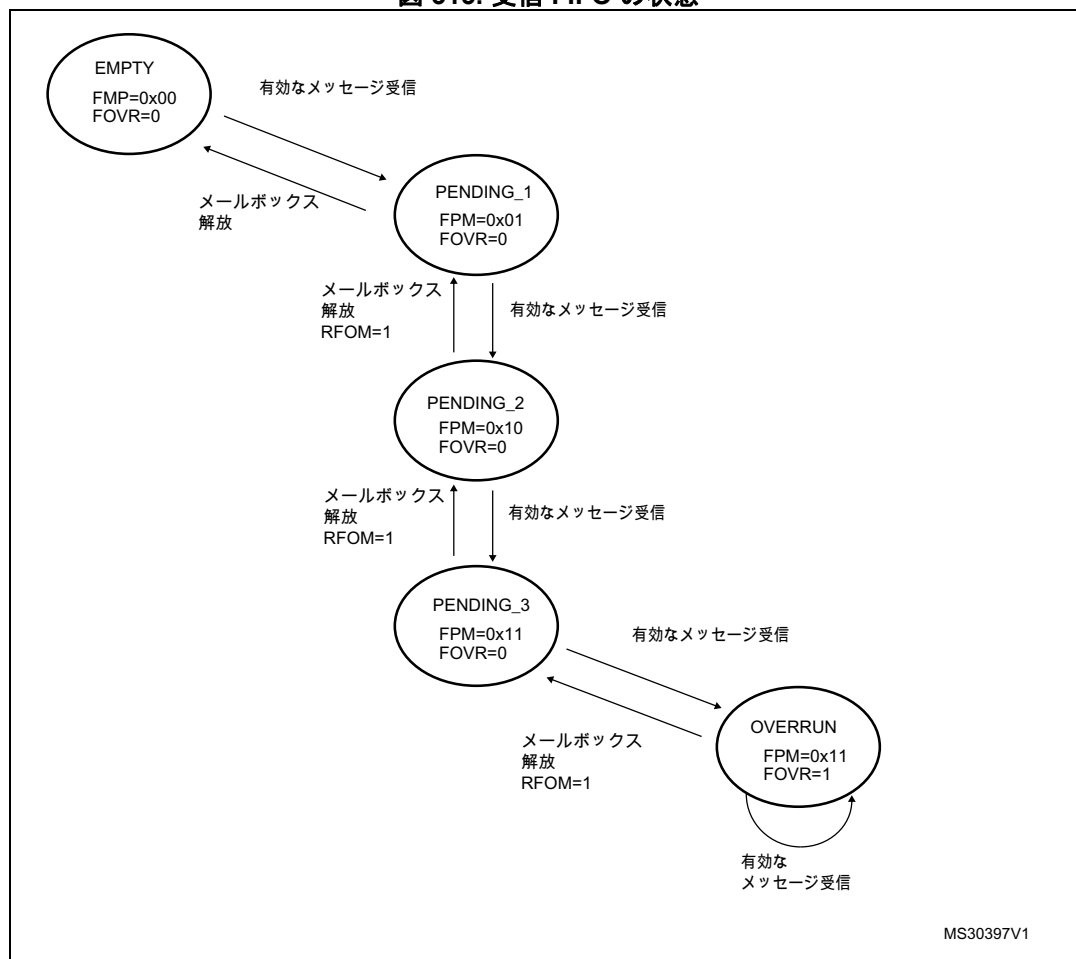
29.7.3 受信処理

CAN メッセージの受信には、3 つのメールボックスが FIFO として用意されています。CPU 負荷を軽減し、ソフトウェアを単純化し、データの一貫性を保証するために、FIFO の管理はすべてハードウェアによって行われます。アプリケーションは、FIFO 出力メールボックスを通じて、FIFO に格納されたメッセージにアクセスします。

有効なメッセージ

受信メッセージは、CAN プロトコルに従って正しく受信され（EOF フィールドの最後から 2 番目のビットまでエラーなし）、かつ、識別子フィルタリングを正常に通過した場合に、有効であるとみなされます（[セクション 29.7.4: 識別子フィルタリング](#)を参照）。

図 313. 受信 FIFO の状態



FIFO 管理

EMPTY 状態から始まり、受信した最初の有効なメッセージが FIFO に格納されると、FIFO は **PENDING_1** 状態になります。ハードウェアは、CAN_RFR レジスタの FMP[1:0] ビットを値 01b に設定して、このイベントを通知します。このメッセージは FIFO 出力メールボックスからアクセスできます。ソフトウェアは、このメールボックスの内容を読み出し、CAN_RFR レジスタの RFOM ビットをセットすることによってメールボックスを解放します。FIFO は再び **EMPTY** 状態になります。その間に新しい有効なメッセージを受信すると、FIFO は **PENDING_1** 状態にとどまり、出力メールボックスから新しいメッセージにアクセスできます。

コード例については、「付録」セクション [A.11.3 : bxCAN 受信のコード例](#) を参照してください。

アプリケーションがメールボックスを解放しない場合には、次の有効なメッセージが FIFO に格納され、FIFO は **PENDING_2** 状態 (FMP[1:0] = 10b) になります。この格納プロセスは、次の有効なメッセージに対しても繰り返され、FIFO は **PENDING_3** 状態 (FMP[1:0] = 11b) になります。この時点で、メールボックスが次の有効なメッセージを格納できるように、ソフトウェアは、RFOM ビットをセットして、出力メールボックスを解放する必要があります。解放しないと、受信した次の有効なメッセージが失われます。

[セクション 29.7.5 : メッセージストレージ](#) も参照してください。

オーバーラン

FIFO が **PENDING_3** 状態 (すなわち、3 個のメールボックスが満杯) になると、次の有効なメッセージの受信で **OVERRUN** 状態となり、メッセージが失われます。ハードウェアは、CAN_RFR レジスタの FOVR ビットをセットすることによって、オーバーラン状態を通知します。どのメッセージが失われるかは、FIFO の設定に依存します。

- FIFO ロック機能が無効 (CAN_MCR レジスタの RFLM ビットがクリア) である場合、最後に FIFO に格納されたメッセージが新しい受信メッセージによって上書きされます。この場合、アプリケーションは常に最新のメッセージにアクセスできます。
- FIFO ロック機能が有効 (CAN_MCR レジスタの RFLM ビットがセット) である場合、最新のメッセージが破棄され、ソフトウェアは FIFO 内にある 3 個の最も古いメッセージにアクセスできます。

受信関連の割込み

メッセージが FIFO に格納されると、FMP[1:0] ビットが更新され、割込みリクエストが生成されます (CAN_IER レジスタの FMPIE ビットがセットされている場合)。

FIFO が満杯になると (3 番目のメッセージが格納されると)、CAN_RFR レジスタの FULL ビットがセットされ、割込みが生成されます (CAN_IER レジスタの FFIE ビットがセットされている場合)。

オーバーラン状態になると、FOVR ビットがセットされ、割込みが生成されます (CAN_IER レジスタの FOVIE ビットがセットされている場合)。

29.7.4 識別子フィルタリング

CAN プロトコルでは、メッセージの識別子は、ノードのアドレスには関連付けられておらず、メッセージの内容に関連したものとなっています。したがって、トランスミッタは、そのメッセージをすべてのレシーバにブロードキャスト (一斉送信) します。メッセージを受信すると、受信ノードは、識別子の値に基づいてこのメッセージがソフトウェアに必要なものかどうかを判断します。このメッセージが必要な場合は、SRAM にコピーされます。そうでない場合、このメッセージは、ソフトウェアによる介入なしに破棄される必要があります。

この要件を満たすため、bxCAN コントローラは 28 個の設定可能でスケーラブルなフィルタバンク (27-0) をアプリケーションに提供します。それ以外のデバイスでは、ソフトウェアに必要なメッセージのみを受信できるように、bxCAN コントローラは 14 個の設定可能でスケーラブルなフィルタバンク (13-0) をアプリケーションに提供します。このハードウェアフィルタリングによって、ソフトウェアでフィルタリングを行う場合に必要となる CPU リソースを節約できます。各フィルタバンク x は、2 個の 32 ビットレジスタ (CAN_FxR0 と CAN_FxR1) で構成されています。

スケーラブルな幅

フィルタをアプリケーションニーズに合わせて最適化/適応するため、各フィルタバンクは個別に拡大/縮小できます。フィルタのスケールに応じて、フィルタバンクは以下のフィルタを提供します。

- 1 個の 32 ビットフィルタ : STDID[10:0]、EXTID[17:0]、IDE、および RTR ビット用。
- 2 個の 16 ビットフィルタ : STDID[10:0]、RTR、IDE、および EXTID[17:15] ビット用。

 [314](#)を参照してください。

さらに、これらのフィルタはマスクモードまたは識別子リストモードに設定できます。

マスクモード

マスクモードでは、識別子レジスタは、識別子のどのビットを「一致が必要」として扱い、どのビットを「無視」として扱うかを指定するマスクレジスタに関連付けられます。

識別子リストモード

識別子リストモードでは、マスクレジスタが識別子レジスタとして使用されます。このように、識別子とマスクを定義する代わりに、2 つの識別子を指定することで、単一識別子の数を倍にします。受信識別子のすべてのビットが、フィルタレジスタで指定されたビットと一致する必要があります。

フィルタバンクのスケールとモードの設定

フィルタバンクは、対応する CAN_FMR レジスタによって設定されます。フィルタバンクを設定するには、CAN_FAR レジスタの FACT ビットをクリアしてフィルタバンクを無効にする必要があります。フィルタのスケールは、CAN_FS1R レジスタの対応する FSCx ビットによって設定されます ( [314](#) を参照)。対応するマスク/識別子レジスタの**識別子リスト**または**識別子マスクモード**は、CAN_FMR レジスタの FBMx ビットによって設定されます。

識別子のグループをフィルタするには、マスクモードでマスク/識別子レジスタを設定します。

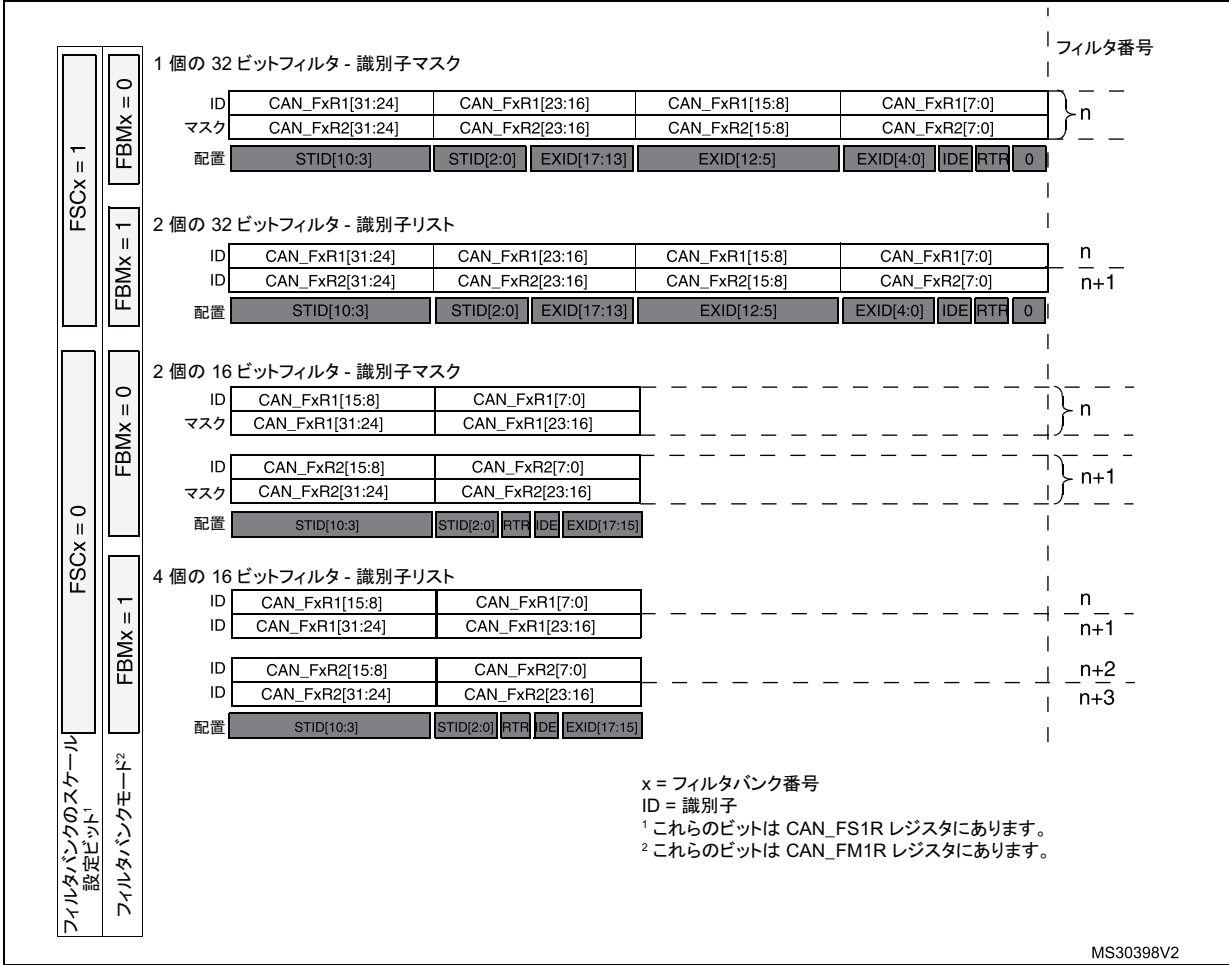
単一識別子を選択するには、識別子リストモードでマスク/識別子レジスタを設定します。

アプリケーションで使用されないフィルタは、非アクティブのままにしておきます。

フィルタバンク内の各フィルタには、各フィルタバンクのモードとスケールに応じて 0 から最大値までの番号 (フィルタ番号) が付けられます。

フィルタの設定については  [314](#) を参照してください。

図 314. フィルタバンクのスケール設定 - レジスタ構造



該当フィルタインデックス

メッセージは、FIFO で受信するとアプリケーションからアクセス可能になります。通常、アプリケーションのデータは SRAM のいずれかの位置にコピーされます。データを正しい位置にコピーするには、アプリケーションが識別子によってデータを識別する必要があります。この操作を省き、SRAM へのアクセスを容易にするために、CAN コントローラは該当フィルタインデックス (Filter Match Index) を搭載しています。

このインデックスは、フィルタの優先順位規則に従って、メッセージとともにメールボックスに格納されます。したがって、受信した各メッセージには関連付けられた該当フィルタインデックスがあります。

該当フィルタインデックスは、次の 2 つの方法で使用できます。

- 該当フィルタインデックスと予期される値のリストを比較します。
- 該当フィルタインデックスを配列のインデックスとして使用し、データの格納先にアクセスします。

マスクされていないフィルタについては、ソフトウェアが識別子を比較する必要はありません。

フィルタがマスクされている場合、ソフトウェアは、マスクされたビットのみを比較します。

フィルタ番号のインデックス値では、フィルタバンクのアクティブ状態を考慮に入れません。さらに、FIFO ごとに 1 つずつ、2 つの独立した番号付け方式を使用します。図 315 の例を参照してください。

図 315. フィルタ番号の例

フィルタ バンク	FIFO0	フィルタ 番号	フィルタ バンク	FIFO1	フィルタ 番号
0	ID リスト (32 ビット)	0 1	2	ID マスク (16 ビット)	0 1
1	ID マスク (32 ビット)	2	4	ID リスト (32 ビット)	2 3
3	ID リスト (16 ビット)	3 4 5 6	7	非アクティブ ID リスト (16 ビット)	4 5
5	非アクティブ ID リスト (32 ビット)	7 8	8	ID マスク (16 ビット)	6 7
6	ID マスク (16 ビット)	9 10	10	非アクティブ ID リスト (16 ビット)	8 9 10 11
9	ID リスト (32 ビット)	11 12	11	ID リスト (32 ビット)	12 13
13	ID マスク (32 ビット)	13	12	ID マスク (32 ビット)	14

ID = 識別子

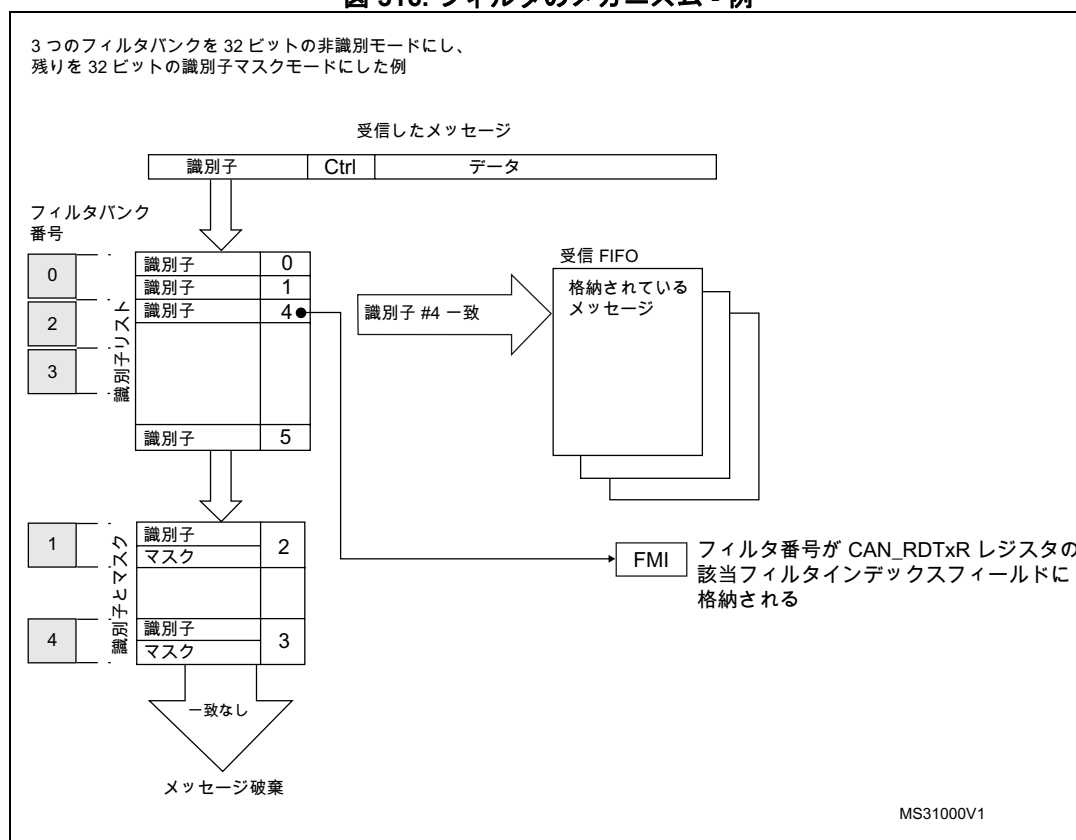
MS30399V1

フィルタの優先順位規則

フィルタの組み合わせによっては、1 つの識別子が複数のフィルタを正常に通過することがあります。この場合、受信メールボックスに格納される該当フィルタ値は、次の優先順位規則に基づいて選択されます。

- 32 ビットフィルタは 16 ビットフィルタよりも優先されます。
- 同じスケールのフィルタでは、識別子マスクモードよりも識別子リストモードが優先されます。
- スケールとモードが同じフィルタでは、フィルタ番号によって優先順位が与えられます（番号が小さいほど優先順位が高くなります）。

図 316. フィルタのメカニズム - 例



上の例は bxCAN のフィルタリングの原理を示しています。メッセージを受信すると、その識別子は、まず識別子リストモードで設定されたフィルタと比較されます。一致する項目があれば、メッセージは関連する FIFO に格納され、一致したフィルタのインデックスが該当フィルタインデックスに格納されます。この例では、識別子は識別子 #4 と一致するため、メッセージの内容と FMI 2 が FIFO に格納されます。

一致する項目がなければ、受信メッセージの識別子は、次にマスクモードで設定されたフィルタと比較されます。

この識別子がフィルタ内のどの識別子とも一致しない場合、メッセージはハードウェアによって破棄され、ソフトウェアが介入することはありません。

29.7.5 メッセージストレージ

CAN メッセージに関するソフトウェアとハードウェアのインタフェースは、メールボックスによって実装されます。メールボックスには、識別子、データ、制御、ステータス、タイムスタンプ情報など、メッセージに関するすべての情報が格納されます。

送信メールボックス

ソフトウェアは、送信しようとするメッセージを空の送信メールボックスにセットアップします。送信のステータスは、ハードウェアによって CAN_TSR レジスタに示されます。

表 115. 送信メールボックスの配置

送信メールボックスのベースアドレスのオフセット	レジスタ名
0	CAN_TlRxR
4	CAN_TDTxR
8	CAN_TDLxR
12	CAN_TDHxR

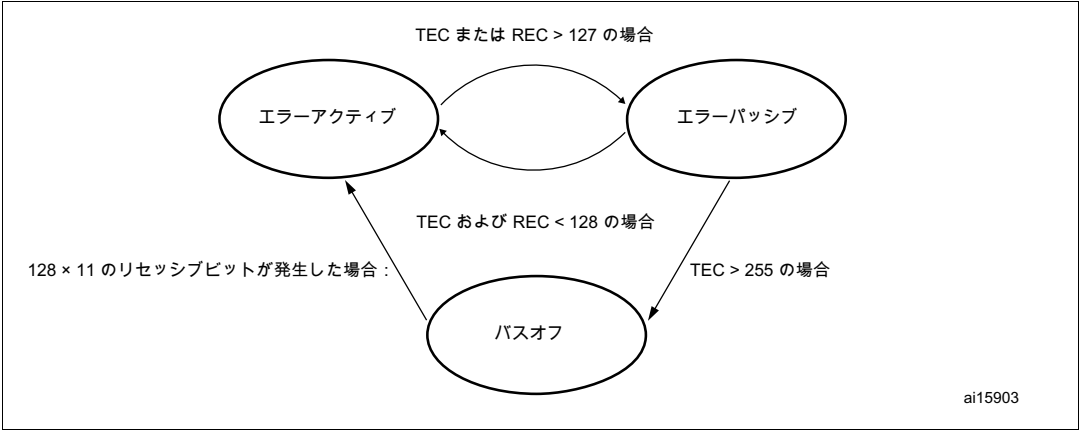
受信メールボックス

メッセージを受信すると、ソフトウェアは FIFO 出力メールボックスを通してそのメッセージにアクセスできます。ソフトウェアが読み出しなどのメッセージの処理を終えると、ソフトウェアは、CAN_RFR レジスタの RFOM ビットによって FIFO 出力メールボックスを解放し、次の受信メッセージにアクセスできるようにする必要があります。該当フィルタインデックスは、CAN_RDTxR レジスタの MFMI フィールドに格納されます。16 ビットタイムスタンプの値は、CAN_RDTxR レジスタの TIME[15:0] フィールドに格納されます。

表 116. 受信メールボックスの配置

受信メールボックスのベースアドレスのオフセット (バイト)	レジスタ名
0	CAN_RlRxR
4	CAN_RDTxR
8	CAN_RDLxR
12	CAN_RDHxR

図 317. CAN エラー状態図



29.7.6 エラー管理

CAN プロトコルに記述されているエラー管理は、エラー条件によってインクリメント／デクリメントされる送信エラーカウンタ (CAN_ESR レジスタの TEC 値) と受信エラーカウンタ (CAN_ESR レジスタの REC 値) を使用して、すべてハードウェアによって処理されます。TEC および REC の管理の詳細については、CAN 規格を参照してください。

ソフトウェアは、この 2 つのカウンタを読み出してネットワークの安定性を判断できます。さらに、CAN ハードウェアは、現在のエラーステータスに関する詳細な情報を CAN_ESR レジスタで提供します。CAN_IER レジスタ (ERRIE ビットなど) によって、ソフトウェアは、エラー検出時の割込み生成をきわめて柔軟に設定できます。

バスオフからの復帰

TEC が 255 を超えるとバスオフ状態になります。この状態は CAN_ESR レジスタの BOFF ビットによって示されます。バスオフ状態では、bxCAN はメッセージの送受信を行うことができません。

CAN_MCR レジスタの ABOM ビットに応じて、bxCAN は、自動的に、またはソフトウェアのリクエストに基づいて、バスオフ状態から復帰します (再びエラーアクティブになります)。しかし、いずれの場合も、bxCAN は、少なくとも CAN 規格で定められた復帰シーケンス (CANRX で観測される 11 個の連続したリセッスビットの発生 128 回) の間は待機する必要があります。

ABOM がセットされると、bxCAN は、バスオフ状態に入った後で自動的に復帰シーケンスを開始します。

ABOM がクリアされている場合、ソフトウェアは、bxCAN に対して初期化モードに入って出るようにリクエストして、復帰シーケンスを開始する必要があります。

注： *初期化モードでは、bxCAN は CANRX 信号を監視しないため、復帰シーケンスを完了できません。復帰するには、bxCAN は通常モードでなければなりません。*

29.7.7 ビットタイミング

ビットタイミング回路は、シリアルバスラインを監視し、スタートビットエッジでの同期と後続エッジでの再同期によって、サンプリングとサンプリングポイントの調整を行います。

この動作は、公称ビット時間を次のように 3 つのセグメントに分割するとわかりやすく説明できます。

- **同期セグメント (SYNC_SEG) :** この時間セグメント内にビット変化が生じることが想定されます。この時間の長さは 1 タイムクオンタム ($1 \times t_q$) に固定されています。
- **ビットセグメント 1 (BS1) :** サンプルポイントの位置を定義します。これには、CAN 規格の PROP_SEG と PHASE_SEG1 が含まれます。セグメントの時間は 1~16 タイムクオンタムの間でプログラミング可能ですが、ネットワークのさまざまなノード間の周波数の差に起因する正の位相ドリフトを補正するために、自動的に延長されることがあります。
- **ビットセグメント 2 (BS2) :** 送信ポイントの位置を定義します。これは、CAN 規格の PHASE_SEG2 に相当します。セグメントの時間は 1~8 タイムクオンタムの間でプログラミング可能ですが、負の位相ドリフトを補正するために、自動的に短縮されることもあります。

再同期ジャンプ幅 (SJW) は、ビットセグメントの延長／短縮量の上限を定め、1~4 タイムクオンタムの間でプログラミング可能です。

有効なエッジとは、コントローラ自体はリセッスビットを送信しないという条件で、あるビット時間において、ドミナントからリセッスへのバスレベルの最初の遷移と定義されます。

有効なエッジが SYNC_SEG ではなく BS1 で検出された場合、サンプルポイントが遅くなるように、BS1 は最大で SJW 延長されます。

逆に、有効なエッジが SYNC_SEG ではなく BS2 で検出された場合、送信ポイントが早くなるように、BS2 は最大で SJW 短縮されます。

プログラミングエラーに対する安全対策として、ビットタイミングレジスタ (CAN_BTR) は、デバイスが STANDBY モードの場合にのみ設定可能です。

注： CAN ビットタイミングと再同期メカニズムの詳細については、ISO11898 規格を参照してください。

図 318. ビットタイミング

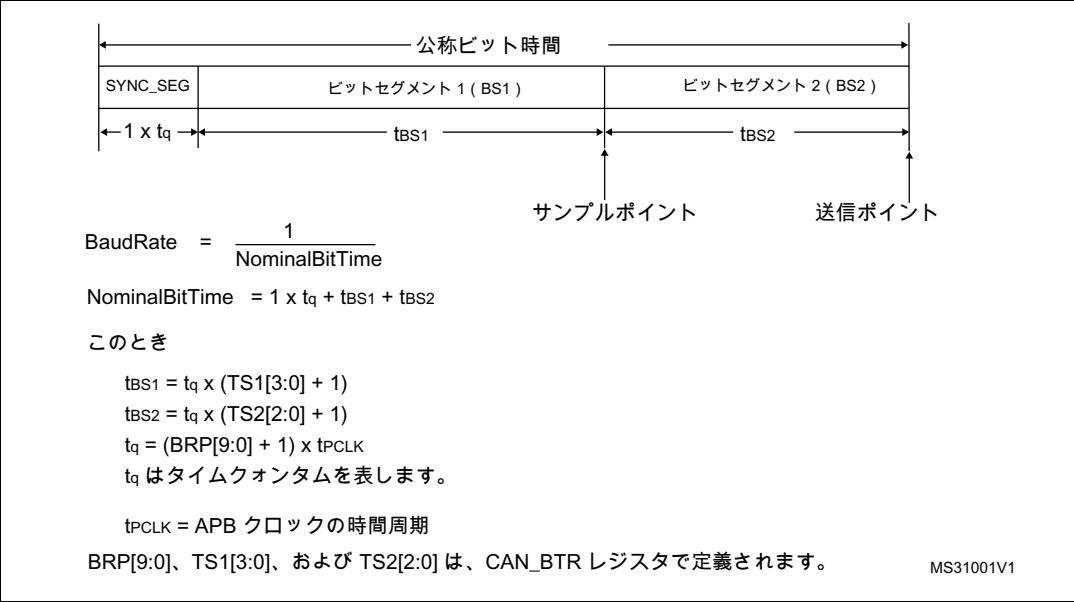
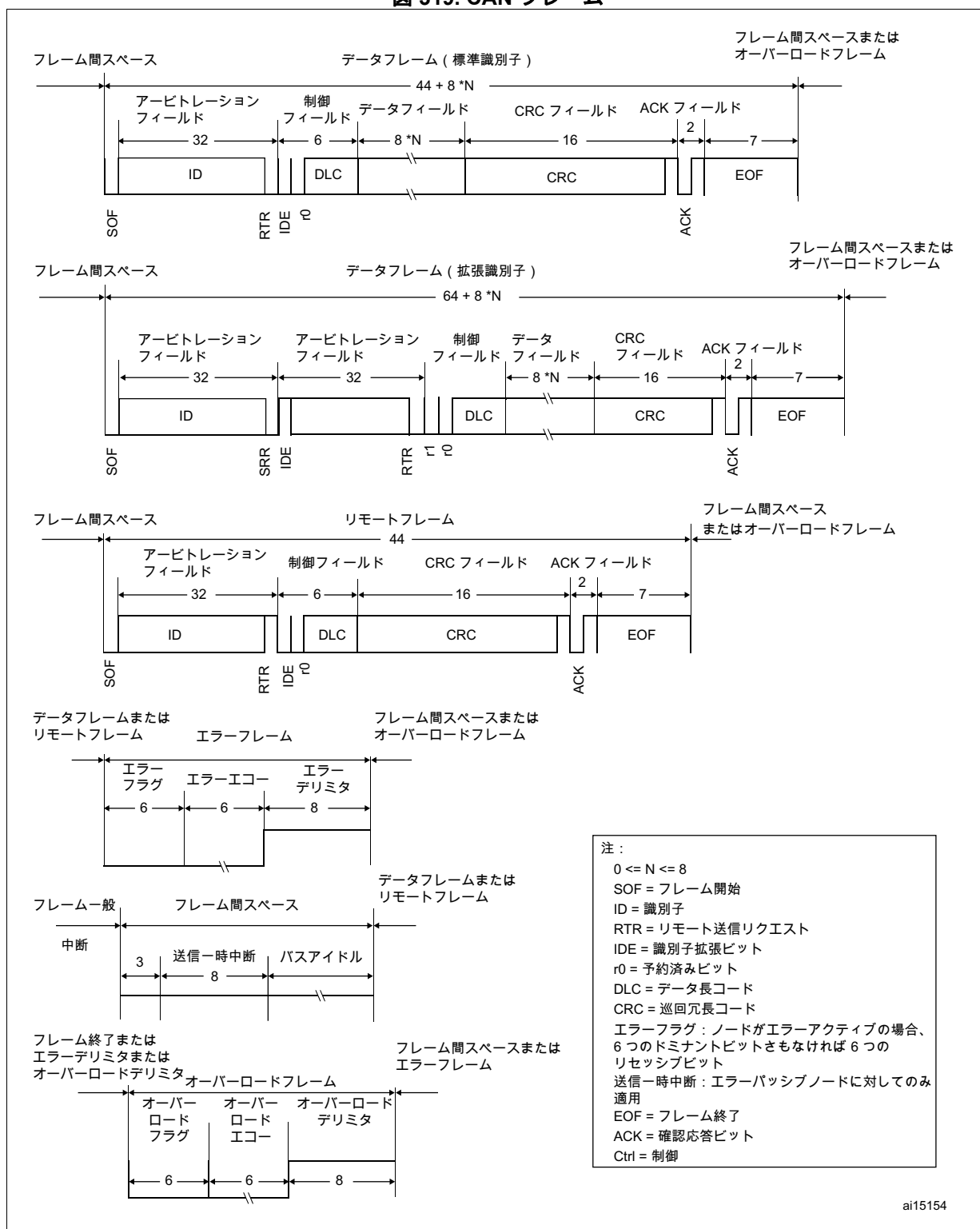


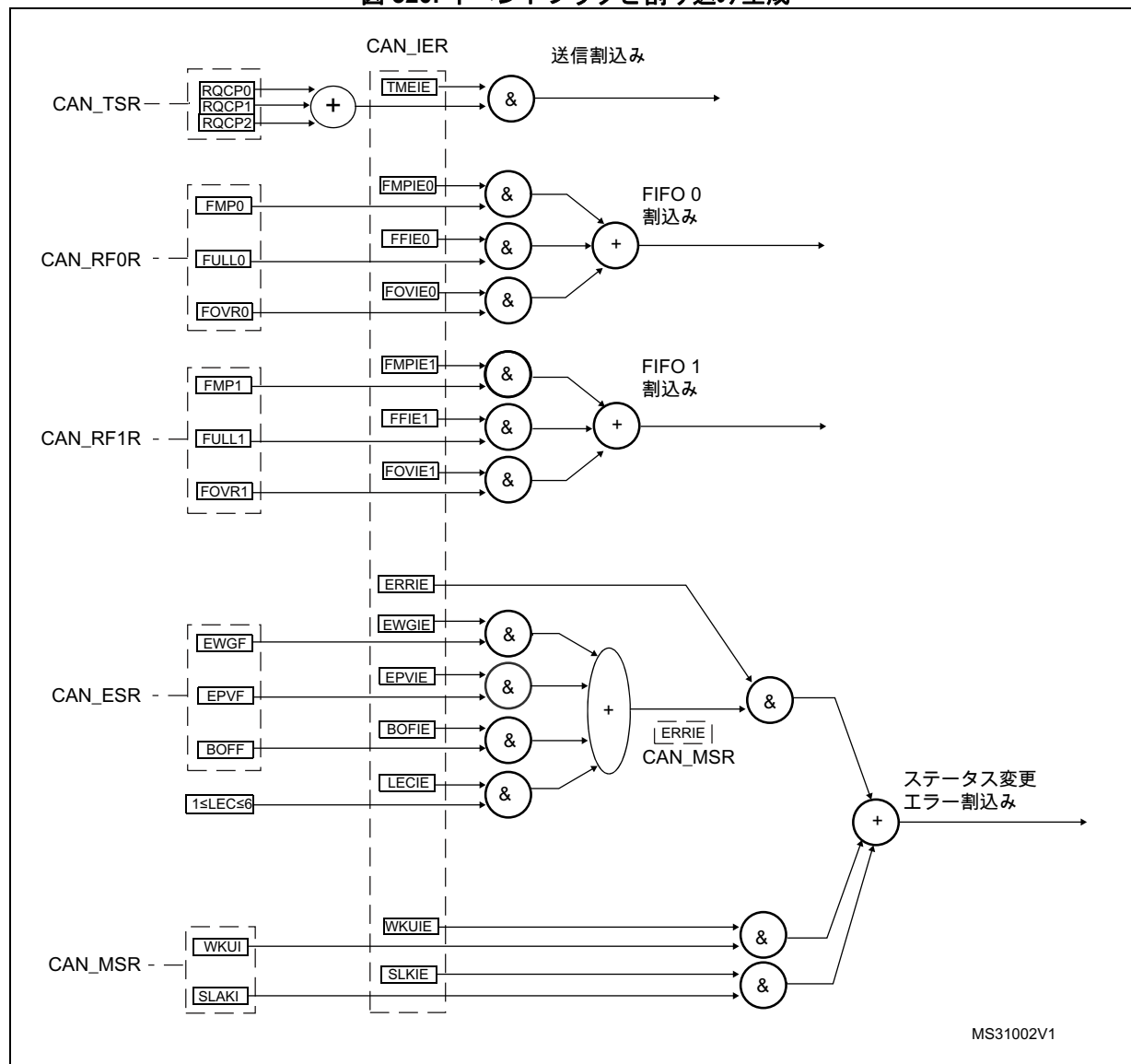
図 319. CAN フレーム



29.8 bxCAN の割込み

bxCAN には 4 つの割込みベクタが確保されています。各割込みソースは、CAN 割込み有効レジスタ (CAN_IER) によって個別に有効/無効にできます。

図 320. イベントフラグと割り込み生成



- **送信割り込み**は次のイベントによって生成できます。
 - 送信メールボックス 0 が空になり、CAN_TSR レジスタの RQCP0 ビットがセットされます。
 - 送信メールボックス 1 が空になり、CAN_TSR レジスタの RQCP1 ビットがセットされます。
 - 送信メールボックス 2 が空になり、CAN_TSR レジスタの RQCP2 ビットがセットされます。
- **FIFO 0 割り込み**は次のイベントによって生成できます。
 - 新しいメッセージの受信、CAN_RF0R レジスタの FMP0 ビットが“00”ではない場合。
 - FIFO0 の満杯状態、CAN_RF0R レジスタの FULL0 ビットがセットされる場合。
 - FIFO0 のオーバーラン状態、CAN_RF0R レジスタの FOVR0 ビットがセットされる場合。
- **FIFO 1 割り込み**は次のイベントによって生成できます。
 - 新しいメッセージの受信、CAN_RF1R レジスタの FMP1 ビットが“00”ではない場合。
 - FIFO1 の満杯状態、CAN_RF1R レジスタの FULL1 ビットがセットされる場合。
 - FIFO1 のオーバーラン状態、CAN_RF1R レジスタの FOVR1 ビットがセットされる場合。
- **エラー/ステータス変更割り込み**は次のイベントによって生成できます。
 - エラー条件：エラー条件の詳細については、CAN エラーステータスレジスタ (CAN_ESR) を参照してください。
 - ウェイクアップ条件：CAN Rx 信号で SOF が観測されたとき。
 - スリープモードへの遷移。

29.9 CAN レジスタ

ペリフェラルレジスタには、ワード (32 ビット) 単位でアクセスする必要があります。

29.9.1 レジスタのアクセス保護

特定の設定レジスタに誤ったアクセスを行うと、ハードウェアが CAN ネットワーク全体の動作を一時的に妨害することがあります。このため、ソフトウェアが CAN_BTR レジスタを変更できるのは、CAN ハードウェアが初期化モードのときに限られます。

誤ったデータを送信しても CAN ネットワークレベルでは問題は生じませんが、アプリケーションには重大な影響を与える可能性があります。ソフトウェアが送信メールボックスを変更できるのは、メールボックスが EMPTY 状態のときに限られます。図 312: 送信メールボックスの状態を参照してください。

フィルタ値を変更するには、関連するフィルタバンクを非アクティブにするか、FINIT ビットをセットします。さらに、CAN_FMR、CAN_FSR、および CAN_FFR レジスタにあるフィルタ設定 (スケール、モード、および FIFO 割当て) の変更は、CAN_FMR レジスタでフィルタ初期化モードがセットされている (FINIT = 1) 場合に限られます。

29.9.2 CAN 制御／ステータスレジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1](#) を参照してください。

CAN マスタ制御レジスタ (CAN_MCR)

アドレスオフセット : 0x00

リセット値 : 0x0001 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBF
															rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
リセット	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ
rS								rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:17 予約済みであり、リセット値に保持する必要があります。

ビット 16 **DBF** : デバッグ凍結

- 0 : デバッグ時に CAN が機能します。
- 1 : デバッグ時に CAN の送受信機能が凍結されます。受信 FIFO は、まだ正常にアクセス／制御できます。

ビット 15 **RESET** : bxCAN ソフトウェアマスタリセット

- 0 : 通常動作。
- 1 : bxCAN のマスタリセットを強制します。リセット後は、スリープモードがアクティブになります (FMP ビットと CAN_MCR レジスタはリセット値に初期化されます)。このビットは自動的に 0 にリセットされます。

ビット 14:8 予約済みであり、リセット値に保持する必要があります。

ビット 7 **TTCM** : タイムトリガ通信モード

- 0 : タイムトリガ通信モードは無効です。
- 1 : タイムトリガ通信モードは有効です。

注 : [タイムトリガ通信モードの詳細については、セクション 29.7.2 : タイムトリガ通信モードを参照してください。](#)

ビット 6 **ABOM** : 自動バスオフ管理

- このビットは、バスオフ状態から出るときの CAN ハードウェアの動作を制御します。
 - 0 : 11 個のリセツピットが 128 回観測され、ソフトウェアが CAN_MCR レジスタの INRQ ビットをまずセットし、クリアすると、ソフトウェアリクエストでバスオフ状態から出ます。
 - 1 : 11 個のリセツピットが 128 回観測されると、ハードウェアによって自動的にバスオフ状態から出ます。
- バスオフ状態の詳細については、[セクション 29.7.6 : エラー管理](#)を参照してください。

ビット 5 **AWUM** : 自動ウェイクアップモード

- このビットは、スリープモードでのメッセージ受信時の CAN ハードウェアの動作を制御します。
 - 0 : ソフトウェアリクエストで CAN_MCR レジスタの SLEEP ビットをクリアしてスリープモードから出ます。
 - 1 : CAN メッセージの検出時にハードウェアによって自動的にスリープモードから出ます。
- CAN_MCR レジスタの SLEEP ビットと CAN_MSR レジスタの SLAK ビットはハードウェアによってクリアされます。

- ビット 4 **NART** : 自動再送信禁止
- 0 : CAN ハードウェアは、CAN 規格に基づいてメッセージを正常に送信するまで、メッセージを自動的に再送信します。
- 1 : 送信結果 (正常、エラー、アービトレーションロスト) にかかわらず、メッセージは 1 回だけ送信されます。
- ビット 3 **RFLM** : 受信 FIFO ロックモード
- 0 : 受信 FIFO はオーバーラン時にロックされません。受信 FIFO が満杯になると、次の受信メッセージは前のメッセージに上書きされます。
- 1 : 受信 FIFO はオーバーランに対してロックされます。受信 FIFO が満杯になると、次の受信メッセージは破棄されます。
- ビット 2 **TXFP** : 送信 FIFO 優先順位
- このビットは、複数のメールボックスが同時にペンディング状態のときの送信順序を制御します。
- 0 : 優先順位はメッセージの識別子によって決まります。
- 1 : 優先順位はリクエスト順に (時系列に) 決まります。
- ビット 1 **SLEEP** : スリープモードリクエスト
- このビットは、CAN ハードウェアがスリープモードに入るように要求するために、ソフトウェアによってセットされます。現在の CAN 動作 (CAN フレームの送受信) が完了すると、すぐにスリープモードに入ります。
- ソフトウェアによってこのビットがクリアされると、スリープモードを終了します。
- AWUM ビットがセットされていて、CAN Rx 信号で SOF ビットが検出されると、このビットはハードウェアによってクリアされます。
- このビットはリセット後にセットされます。CAN はスリープモードで開始します。
- ビット 0 **INRQ** : 初期化リクエスト
- ソフトウェアによってこのビットがクリアされると、ハードウェアが通常モードに切り替わります。Rx 信号で 11 個の連続したリセシブビットが観測されると、CAN ハードウェアが同期し、送受信の準備ができます。ハードウェアは、CAN_MSR レジスタの INAK ビットをクリアすることによって、このイベントを通知します。
- このビットは、CAN ハードウェアに初期化モードに入るようにリクエストするために、ソフトウェアによってセットされます。ソフトウェアが INRQ ビットをセットすると、CAN ハードウェアは現在の CAN 動作 (送受信) が完了するのを待ってから、初期化モードに入ります。ハードウェアは、CAN_MSR レジスタの INAK ビットをセットすることによって、このイベントを通知します。

CAN マスタステータスレジスタ (CAN_MSR)

アドレスオフセット : 0x04
リセット値 : 0x0000 0C02

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	SLAKI	WKUI	ERRI	SLAK	INAK
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r



ビット 31:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **RX** : CAN Rx 信号

CAN_RX ピンの実際の値を監視します。

ビット 10 **SAMP** : 最終サンプルポイント

最後のサンプルポイントにおける RX の値（受信した現在のビット値）です。

ビット 9 **RXM** : 受信モード

CAN ハードウェアは、現在はレシーバです。

ビット 8 **TXM** : 送信モード

CAN ハードウェアは、現在はトランスミッタです。

ビット 7:5 予約済みであり、リセット値に保持する必要があります。

ビット 4 **SLAKI** : スリープ確認割込み

SLKIE=1 の場合、このビットは、bxCAN がスリープモードに入ったことを通知するために、ハードウェアによってセットされます。CAN_IER レジスタの SLKIE ビットがセットされている場合にこのビットがセットされると、ステータス変更割込みが生成されます。

SLAK がクリアされると、このビットはソフトウェアまたはハードウェアによってクリアされます。

注： *SLKIE=0 の場合、SLAKI ビットのポーリングはできません。この場合、SLAK ビットはポーリングできます。*

ビット 3 **WKUI** : ウェイクアップ割り込み

このビットは、CAN ハードウェアがスリープモードのときに SOF ビットが検出されたことを通知するために、ハードウェアによってセットされます。CAN_IER レジスタの WKUIE ビットがセットされている場合、このビットをセットするとステータス変更割込みが生成されます。

このビットはソフトウェアによってクリアされます。

ビット 2 **ERRI** : エラー割込み

エラー検出時に CAN_ESR のビットがセットされており、CAN_IER で対応する割込みが有効になっている場合、このビットはハードウェアによってセットされます。CAN_IER レジスタの ERRIE ビットがセットされている場合、このビットをセットするとステータス変更割込みが生成されます。

このビットはソフトウェアによってクリアされます。

ビット 1 **SLAK** : スリープ確認応答

このビットは、ハードウェアによってセットされ、CAN ハードウェアがスリープモードにあることをソフトウェアに知らせます。このビットは、ソフトウェアからのスリープモードリクエスト (CAN_MCR レジスタの SLEEP ビットをセット) に対する確認応答になります。

このビットは、CAN ハードウェアがスリープモードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセシブビットを観測する必要があります。

注： *CAN_MCR レジスタの SLEEP ビットがクリアされると、スリープモードから出るプロセスがトリガされます。SLEEP ビットのクリアについては、CAN_MCR レジスタの AWUM ビットの説明を参照してください。*

ビット 0 **INAK** : 初期化確認応答

このビットは、CAN ハードウェアが初期化モードにあることをソフトウェアに知らせるために、ハードウェアによってセットされます。このビットは、ソフトウェアからの初期化リクエスト (CAN_MCR レジスタの INRQ ビットをセット) に対する確認応答になります。

このビットは、CAN ハードウェアが初期化モードから出た (CAN バスで同期する) ときに、ハードウェアによってクリアされます。同期をとるため、ハードウェアは、CAN RX 信号で 11 個の連続したリセシブビットを観測する必要があります。

CAN 送信ステータスレジスタ (CAN_TSR)

アドレスオフセット: 0x08
リセット値: 0x1C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2
r	r	r	r	r	r	r	r	rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	TERR0	ALST0	TXOK0	RQCP0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

- ビット 31 **LOW2**: メールボックス 2 最低優先順位フラグ
複数のメールボックスが送信待ち状態であり、メールボックス 2 が最低優先順位の時、このビットがハードウェアによってセットされます。
- ビット 30 **LOW1**: メールボックス 1 最低優先順位フラグ
複数のメールボックスが送信待ち状態であり、メールボックス 1 が最低優先順位の時、このビットがハードウェアによってセットされます。
- ビット 29 **LOW0**: メールボックス 0 最低優先順位フラグ
複数のメールボックスが送信待ち状態であり、メールボックス 0 が最低優先順位の時、このビットがハードウェアによってセットされます。
注: ペンディング状態のメールボックスが1 つしかないとき、LOW[2:0] ビットは 0 にセットされません。
- ビット 28 **TME2**: 送信メールボックス 2 エンプティ
このビットは、メールボックス 2 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。
- ビット 27 **TME1**: 送信メールボックス 1 エンプティ
このビットは、メールボックス 1 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。
- ビット 26 **TME0**: 送信メールボックス 0 エンプティ
このビットは、メールボックス 0 に待ち状態となる送信リクエストがないとき、ハードウェアによってセットされます。
- ビット 25:24 **CODE[1:0]**: メールボックスコード
少なくとも 1 個の送信メールボックスが空き状態の場合、このコード値は、次の空き状態の送信メールボックスの番号になります。
すべての送信メールボックスが待ち状態の場合、このコード値は、最低優先順位の送信メールボックスの番号になります。
- ビット 23 **ABRQ2**: メールボックス 2 アポートリクエスト
対応するメールボックスへの送信リクエストをアポートするために、ソフトウェアによってセットされます。
メールボックスが空になると、ハードウェアによってクリアされます。
メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。
- ビット 22:20 予約済みであり、リセット値に保持する必要があります。
- ビット 19 **TERR2**: メールボックス 2 送信エラー
このビットは、前の送信 (TX) がエラーによって失敗したときにセットされます。
- ビット 18 **ALST2**: メールボックス 2 アービトレーションロスト
このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。



ビット 17 TXOK2：メールボックス 2 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0：前回の送信は失敗しました。

1：前回の送信は成功しました。

このビットは、メールボックス 2 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 312 を参照してください。

ビット 16 RQCP2：メールボックス 2 リクエスト完了

最後のリクエスト（送信またはアボート）が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます（CAN_TMD2R レジスタの TXRQ2 ビットをセット）。

このビットをクリアするとメールボックス 2 のすべてのステータスビット（TXOK2、ALST2、および TERR2）がクリアされます。

ビット 15 ABRQ1：メールボックス 1 アボートリクエスト

対応するメールボックスへの送信リクエストをアボートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 TERR1：メールボックス 1 送信エラー

このビットは、前の送信（TX）がエラーによって失敗したときにセットされます。

ビット 10 ALST1：メールボックス 1 アービトレーションロスト

このビットは、前の送信（TX）がアービトレーションロストによって失敗したときにセットされます。

ビット 9 TXOK1：メールボックス 1 送信 OK

ハードウェアは、送信を試みた後で毎回このビットを更新します。

0：前回の送信は失敗しました。

1：前回の送信は成功しました。

このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。図 312 を参照してください。

ビット 8 RQCP1：メールボックス 1 リクエスト完了

最後のリクエスト（送信またはアボート）が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます（CAN_T11R レジスタの TXRQ1 ビットをセット）。

このビットをクリアするとメールボックス 1 のすべてのステータスビット（TXOK1、ALST1、および TERR1）がクリアされます。

ビット 7 ABRQ0：メールボックス 0 アボートリクエスト

対応するメールボックスへの送信リクエストをアボートするために、ソフトウェアによってセットされます。

メールボックスが空になると、ハードウェアによってクリアされます。

メールボックスが送信待ち状態にない場合は、このビットをセットしても何も影響しません。

ビット 6:4 予約済みであり、リセット値に保持する必要があります。

ビット 3 TERR0：メールボックス 0 送信エラー

このビットは、前の送信（TX）がエラーによって失敗したときにセットされます。

- ビット 2 **ALST0** : メールボックス 0 アービトレーションロスト
このビットは、前の送信 (TX) がアービトレーションロストによって失敗したときにセットされます。
- ビット 1 **TXOK0** : メールボックス 0 送信OK
ハードウェアは、送信を試みた後で毎回このビットを更新します。
0 : 前回の送信は失敗しました。
1 : 前回の送信は成功しました。
このビットは、メールボックス 1 での送信リクエストが正常に完了したとき、ハードウェアによってセットされます。 [図 312](#)を参照してください。
- ビット 0 **RQCP0** : メールボックス 0 リクエスト完了
最後のリクエスト (送信またはアボート) が実行されたとき、ハードウェアによってセットされます。ソフトウェアによって「1」を書き込むことで、または送信リクエスト時にハードウェアによってクリアされます (CAN_TIOER レジスタの TXRQ0 ビットをセット)。
このビットをクリアするとメールボックス 0 のすべてのステータスビット (TXOK0、ALST0、および TERR0) がクリアされます。

CAN 受信 FIFO 0 レジスタ (CAN_RF0R)

アドレスオフセット : 0x0C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]	
										rs	rc_w1	rc_w1		r	r

- ビット 31:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **RFOM0** : FIFO 0 出力メールボックスの解放
FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。
このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。
- ビット 4 **FOVR0** : FIFO 0 オーバーラン
このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。
このビットはソフトウェアによってクリアされます。
- ビット 3 **FULL0** : FIFO 0 フル
FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。
このビットはソフトウェアによってクリアされます。
- ビット 2 予約済みであり、リセット値に保持する必要があります。
- ビット 1:0 **FMP0[1:0]** : FIFO 0 メッセージペンディング
これらのビットは、受信 FIFO 内でペンディング状態になっているメッセージの数を示します。
FMP は、ハードウェアが FIFO に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM0 ビットをセットして出力メールボックスを解放するたびに減少します。

CAN 受信 FIFO 1 レジスタ (CAN_RF1R)

アドレスオフセット : 0x10
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]	
										rs	rc_w1	rc_w1		r	r

ビット 31:6 予約済みであり、リセット値に保持する必要があります。

ビット 5 **RFOM1** : FIFO 1 出力メールボックスの解放
FIFO の出力メールボックスを解放するために、ソフトウェアによってセットされます。出力メールボックスの解放は、FIFO 内で少なくとも 1 個のメッセージがペンディング状態のときにのみ可能です。FIFO が空のときにビットをセットしても何も起きません。FIFO 内で複数のメッセージがペンディング状態にある場合、ソフトウェアが次のメッセージにアクセスするには、出力メールボックスを解放する必要があります。
このビットは、出力メールボックスが解放されると、ハードウェアによってクリアされます。

ビット 4 **FOVR1** : FIFO 1 オーバーラン
このビットは、FIFO が満杯のときに新しいメッセージが受信され、フィルタを通過したときに、ハードウェアによってセットされます。
このビットはソフトウェアによってクリアされます。

ビット 3 **FULL1** : FIFO 1 フル
FIFO に 3 個のメッセージが格納されたときに、ハードウェアによってセットされます。
このビットはソフトウェアによってクリアされます。

ビット 2 予約済みであり、リセット値に保持する必要があります。

ビット 1:0 **FMP1[1:0]** : FIFO 1 メッセージペンディング
これらのビットは、受信 FIFO1 内でペンディング状態になっているメッセージの数を示します。
FMP1 は、ハードウェアが FIFO1 に新しいメッセージを格納するたびに増加します。FMP は、ソフトウェアが RFOM1 ビットをセットして出力メールボックスを解放するたびに減少します。

CAN 割り込み有効レジスタ (CAN_IER)

アドレスオフセット : 0x14
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Res.	Res.	Res.	LEC IE	BOF IE	EPV IE	EWG IE	Res.	FOV IE1	FF IE1	FMP IE1	FOV IE0	FF IE0	FMP IE0	TME IE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

ビット 31:18 予約済みであり、リセット値に保持する必要があります。

ビット 17 **SLKIE** : スリープ割り込み有効
0 : SLAKI ビットがセットされると、割り込みは生成されません。
1 : SLAKI ビットがセットされると、割り込みが生成されます。

ビット 16 **WKUIE** : ウェイクアップ割り込み有効
0 : WKUI ビットがセットされると、割り込みは生成されません。
1 : WKUI ビットがセットされると、割り込みが生成されます。

ビット 15 **ERRIE** : エラー割り込みイネーブル
0 : CAN_ESR レジスタでエラー条件がペンディング状態のとき、割り込みは生成されません。
1 : CAN_ESR レジスタでエラー条件がペンディング状態のとき、割り込みが生成されます。

ビット 14:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **LECIE** : 最終エラーコード割り込み有効
0 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされるとき、ERRI ビットはセットされません。
1 : エラー検出時にハードウェアによって LEC[2:0] にエラーコードがセットされると、ERRI ビットがセットされます。

ビット 10 **BOFIE** : バスオフ割り込み有効
0 : BOFF がセットされるとき、ERRI ビットはセットされません。
1 : BOFF がセットされると、ERRI ビットがセットされます。

ビット 9 **EPVIE** : エラーパッシブ割り込み有効
0 : EPVF がセットされるとき、ERRI ビットはセットされません。
1 : EPVF がセットされると、ERRI ビットがセットされます。

ビット 8 **EWGIE** : エラー警告割り込み有効
0 : EWGF がセットされるとき、ERRI ビットはセットされません。
1 : EWGF がセットされると、ERRI ビットがセットされます。

ビット 7 予約済みであり、リセット値に保持する必要があります。

ビット 6 **FOVIE1** : FIFO オーバーラン割り込み有効
0 : FOVR がセットされるとき、割り込みは生成されません。
1 : FOVR がセットされると、割り込みが生成されます。

ビット 5 **FFIE1** : FIFO フル割り込み有効
0 : FULL ビットがセットされるとき、割り込みは生成されません。
1 : FULL ビットがセットされると、割り込みが生成されます。



- ビット 4 **FMPIE1** : FIFO メッセージペンディング割込み有効
- 0 : FMP[1:0] ビットの状態が 00b でないとき、割込みは生成されません。
- 1 : FMP[1:0] ビットの状態が 00b でないとき、割込みが生成されます。
- ビット 3 **FOVIE0** : FIFO オーバーラン割込み有効
- 0 : FOVR ビットがセットされるとき、割込みは生成されません。
- 1 : FOVR ビットがセットされると、割込みが生成されます。
- ビット 2 **FFIE0** : FIFO フル割込み有効
- 0 : FULL ビットがセットされるとき、割込みは生成されません。
- 1 : FULL ビットがセットされると、割込みが生成されます。
- ビット 1 **FMPIE0** : FIFO メッセージペンディング割込み有効
- 0 : FMP[1:0] ビットの状態が 00b でないとき、割込みは生成されません。
- 1 : FMP[1:0] ビットの状態が 00b でないとき、割込みが生成されます。
- ビット 0 **TMEIE** : 送信メールボックスエンプティ割込み有効
- 0 : RQCPx ビットがセットされるとき、割込みは生成されません。
- 1 : RQCPx ビットがセットされると、割込みが生成されます。
- 注: [セクション 29.8 : bxCAN の割込みを参照してください。](#)

CAN エラーステータスレジスタ（CAN_ESR）

アドレスオフセット : 0x18

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REC[7:0]								TEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEC[2:0]			Res.	BOFF	EPVF	EWGF
									rw	rw	rw		r	r	r

- ビット 31:24 **REC[7:0]** : 受信エラーカウンタ
- CAN プロトコルの障害隔離メカニズムの実装部です。受信時にエラーがあった場合、このカウンタは、CAN 規格に定められたエラー条件に応じて、1 または 8 だけインクリメントされます。受信が正常に行われるたびに、このカウンタは 1 だけデクリメントされます（カウンタの値が 128 を超えた場合は、120 にリセットされます）。カウンタ値が 127 を超えると、CAN コントローラはエラーパッシブ状態に入ります。
- ビット 23:16 **TEC[7:0]** : 9 ビット送信エラーカウンタの最下位バイト
- CAN プロトコルの障害隔離メカニズムの実装部です。
- ビット 15:7 予約済みであり、リセット値に保持する必要があります。

ビット 6:4 **LEC[2:0]** : 最終エラーコード

このフィールドはハードウェアによってセットされ、最後に CAN バスで検出されたエラーのエラー条件を示すコードを保持しています。メッセージがエラーなしで転送（受信または送信）されている場合には、このフィールドは“0”にクリアされます。
LEC[2:0] ビットは、ソフトウェアによって値 0b111 にセットできます。ビットはハードウェアによって更新され、現在の通信ステータスを示します。
000 : エラーなし
001 : スタッエラー
010 : フォームエラー
011 : ACK エラー
100 : ビットリセッショエラー
101 : ビットドミナントエラー
110 : CRC エラー
111 : ソフトウェアによってセットされます

ビット 3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **BOFF** : バスオフフラグ

このビットは、バスオフ状態に入るとハードウェアによってセットされます。バスオフ状態に入るのは、TEC オーバーフローのとき（255 を超えたとき）です。[セクション 29.7.6 \(838 ページ\)](#) を参照してください。

ビット 1 **EPVF** : エラーパッシブフラグ

このビットは、エラーパッシブ限界に達したときに（受信エラーカウンタまたは送信エラーカウンタ > 127）、ハードウェアによってセットされます。

ビット 0 **EWGF** : エラー警告フラグ

このビットは、警告限界に達したときに、ハードウェアによってセットされます（受信エラーカウンタまたは送信エラーカウンタ ≥ 96）。

CAN ビットタイミングレジスタ (CAN_BTR)

アドレスオフセット : 0x1C
リセット値 : 0x0123 0000

このレジスタは、CAN ハードウェアが初期化モードのときに、ソフトウェアによってのみアクセスできます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILM	LBKM	Res.	Res.	Res.	Res.	SJW[1:0]		Res.	TS2[2:0]			TS1[3:0]			
rw	rw					rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	BRP[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

ビット 31 **SILM** : サイレントモード（デバッグ用）

- 0 : 通常動作
- 1 : サイレントモード

ビット 30 **LBKM** : ループバックモード（デバッグ用）

- 0 : ループバックモードは無効です。
- 1 : ループバックモードは有効です。

ビット 29:26 予約済みであり、リセット値に保持する必要があります。



この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス株式が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス株式は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 25:24 **SJW[1:0]** : 再同期ジャンプ幅

これらのビットは、再同期を行うために CAN ハードウェアがビットを伸縮できる最大タイムクオンタム数を定義します。

$$t_{RJW} = t_q \times (SJW[1:0] + 1)$$

ビット 23 予約済みであり、リセット値に保持する必要があります。

ビット 22:20 **TS2[2:0]** : 時間セグメント 2

これらのビットは、時間セグメント 2 に含まれるタイムクオンタム数を定義します。

$$t_{BS2} = t_q \times (TS2[2:0] + 1)$$

ビット 19:16 **TS1[3:0]** : 時間セグメント 1

これらのビットは、時間セグメント 1 に含まれるタイムクオンタム数を定義します。

$$t_{BS1} = t_q \times (TS1[3:0] + 1)$$

ビットタイミングの詳細については、[セクション 29.7.7 : ビットタイミング \(838 ページ\)](#) を参照してください。

ビット 15:10 予約済みであり、リセット値に保持する必要があります。

ビット 9:0 **BRP[9:0]** : ボーレートプリスケアラ

これらのビットは、タイムクオンタムの長さを定義します。

$$t_q = (BRP[9:0] + 1) \times t_{PCLK}$$

29.9.3 CAN メールボックスレジスタ

ここでは、送受信メールボックスのレジスタについて説明します。詳細なレジスタ配置については、[セクション 29.7.5 : メッセージストレージ \(836 ページ\)](#) を参照してください。

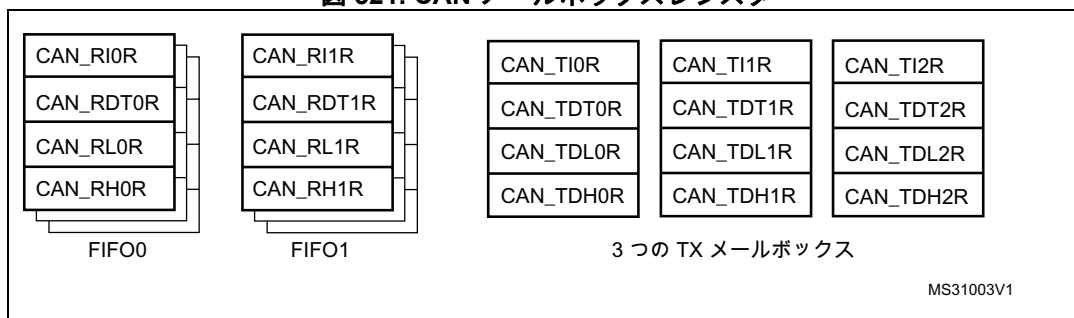
送信メールボックスと受信メールボックスのレジスタは同じですが、次の点が異なります。

- CAN_RDTxR レジスタの FMI フィールド。
- 受信メールボックスは常に書き込み保護されます。
- 送信メールボックスは空のとき (CAN_TSR レジスタの対応する TME ビットがセットされているとき) のみ、書き込み可能です。

送信メールボックスは 3 つ、受信メールボックスは 2 つあります。各受信メールボックスでは、3 種類の深さのレベルの FIFO にアクセスでき、FIFO 内の最も古い受信メッセージにのみアクセスできます。

各メールボックスは 4 つのレジスタから構成されています。

図 321. CAN メールボックスレジスタ



CAN 送信メールボックス識別子レジスタ (CAN_TlRxR) (x=0..2)

アドレスオフセット : 0x180、0x190、0x1A0
リセット値 : 0xFFFF XXXX (ただし、ビット 0、TXRQ = 0)

メールボックスが送信待ち(TMEx リセット)のとき、すべての送信レジスタは書き込み保護されます。
このレジスタは、送信リクエスト制御 (ビット 0、リセット値 0) も実装します。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	TXRQ
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子
IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。
- ビット 20:3 **EXID[17:0]** : 拡張識別子
拡張識別子の最下位ビットです。
- ビット 2 **IDE** : 識別子拡張
このビットは、メールボックス内のメッセージの識別子のタイプを定義します。
0 : 標準識別子。
1 : 拡張識別子。
- ビット 1 **RTR** : リモート送信リクエスト
0 : データフレーム
1 : リモートフレーム
- ビット 0 **TXRQ** : 送信メールボックスリクエスト
対応するメールボックスに送信を要求するために、ソフトウェアによってセットされます。
メールボックスが空になると、ハードウェアによってクリアされます。

CAN メールボックスデータ長制御／タイムスタンプレジスタ (CAN_TDTxR)
(x = 0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x184、0x194、0x1A4
リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIME[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLC[3:0]			
												rw	rw	rw	rw

ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ
このフィールドは、SOF 送信時にキャプチャされた 16 ビットタイマの値を含みます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 **TGT** : 送信グローバルタイム
このビットは、ハードウェアがタイムトリガ通信モードにある (CAN_MCR レジスタの TTCM ビットがセットされている) ときにのみアクティブとなります。
0 : タイムスタンプ TIME[15:0] は送信されません。
1 : タイムスタンプ TIME[15:0] の値は、8 バイトメッセージの最後の 2 データバイトで送信されます。このとき、データバイト 7 の TIME[7:0] とデータバイト 6 の TIME[15:8] は、CAN_TDHxR[31:16] レジスタ (DATA6[7:0]、DATA7[7:0]) に書き込まれたデータを置換します。これらの 2 バイトを CAN バス経由で送信するには、DLC を 8 としてプログラミングする必要があります。

ビット 7:4 予約済みであり、リセット値に保持する必要があります。

ビット 3:0 **DLC[3:0]** : データ長コード
このフィールドは、データフレーム内のデータバイト数、またはリモートフレームリクエストのデータバイト数を定義します。
メッセージは、DLC フィールドの値に応じて、0~8 個のデータバイトを含むことができます。

CAN メールボックスデータ下位レジスタ (CAN_TDLxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x188、0x198、0x1A8
リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:24 **DATA3[7:0]** : データバイト 3
メッセージのデータバイト 3 です。
- ビット 23:16 **DATA2[7:0]** : データバイト 2
メッセージのデータバイト 2 です。
- ビット 15:8 **DATA1[7:0]** : データバイト 1
メッセージのデータバイト 1 です。
- ビット 7:0 **DATA0[7:0]** : データバイト 0
メッセージのデータバイト 0 です。
メッセージは、0~8 個のデータバイトを含むことができ、バイト 0 から始まります。

CAN メールボックスデータ上位レジスタ (CAN_TDHxR) (x=0..2)

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット : 0x18C、0x19C、0x1AC
リセット値 : 0xFFFF XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

- ビット 31:24 **DATA7[7:0]** : データバイト 7
メッセージのデータバイト 7 です。
注: このメッセージの TGT と TTCM がアクティブな場合、DATA7 と DATA6 はタイムスタンプの値で置換されます。
- ビット 23:16 **DATA6[7:0]** : データバイト 6
メッセージのデータバイト 6 です。
- ビット 15:8 **DATA5[7:0]** : データバイト 5
メッセージのデータバイト 5 です。
- ビット 7:0 **DATA4[7:0]** : データバイト 4
メッセージのデータバイト 4 です。

CAN 受信 FIFO メールボックス識別子レジスタ (CAN_RlRxR) (x=0..1)

アドレスオフセット : 0x1B0、0x1C0
リセット値 : 0xFFFF XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXID[12:0]													IDE	RTR	Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

- ビット 31:21 **STID[10:0]/EXID[28:18]** : 標準識別子または拡張識別子
IDE ビットの値に応じて、標準識別子、または拡張識別子の最上位ビットです。
- ビット 20:3 **EXID[17:0]** : 拡張識別子
拡張識別子の最下位ビットです。
- ビット 2 **IDE** : 識別子拡張
このビットは、メールボックス内のメッセージの識別子のタイプを定義します。
0 : 標準識別子。
1 : 拡張識別子。
- ビット 1 **RTR** : リモート送信リクエスト
0 : データフレーム
1 : リモートフレーム
- ビット 0 予約済みであり、リセット値に保持する必要があります。

CAN 受信 FIFO メールボックスデータ長制御/タイムスタンプレジスタ (CAN_RDTxR) (x = 0..1)

アドレスオフセット : 0x1B4、0x1C4
リセット値 : 0xxxxx xxxxx

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIME[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMI[7:0]								Res.	Res.	Res.	Res.	DLC[3:0]			
r	r	r	r	r	r	r	r					r	r	r	r

- ビット 31:16 **TIME[15:0]** : メッセージタイムスタンプ
このフィールドは、SOF 検出時にキャプチャされた 16 ビットタイマの値を含みます。
- ビット 15:8 **FMI[7:0]** : 該当フィルタインデックス
このレジスタは、メールボックスに格納されたメッセージが通過したフィルタの索引を含みます。識別子フィルタリングの詳細については、[セクション 29.7.4 : 識別子フィルタリング \(832 ページ\)](#) の該当フィルタインデックス項を参照してください。
- ビット 7:4 予約済みであり、リセット値に保持する必要があります。
- ビット 3:0 **DLC[3:0]** : データ長コード
このフィールドは、データフレームに格納されるデータバイト数 (0~8) を定義します。リモートフレームリクエストの場合、このフィールドは 0 です。



この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

CAN 受信 FIFO メールボックスデータ下位レジスタ（CAN_RDLxR）（x=0..1）

メールボックスが空の状態でない場合、このレジスタのすべてのビットは書き込み保護されます。

アドレスオフセット：0x1B8、0x1C8

リセット値：0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA3[7:0]**：データバイト 3
メッセージのデータバイト 3 です。

ビット 23:16 **DATA2[7:0]**：データバイト 2
メッセージのデータバイト 2 です。

ビット 15:8 **DATA1[7:0]**：データバイト 1
メッセージのデータバイト 1 です。

ビット 7:0 **DATA0[7:0]**：データバイト 0
メッセージのデータバイト 0 です。
メッセージは、0～8 個のデータバイトを含むことができ、バイト 0 から始まります。

CAN 受信 FIFO メールボックスデータ上位レジスタ（CAN_RDHxR）（x=0..1）

アドレスオフセット：0x1BC、0x1CC

リセット値：0XXXXX XXXX

すべての受信レジスタは書き込み保護されます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:24 **DATA7[7:0]**：データバイト 7
メッセージのデータバイト 3 です。

ビット 23:16 **DATA6[7:0]** : データバイト 6
メッセージのデータバイト 2 です。

ビット 15:8 **DATA5[7:0]** : データバイト 5
メッセージのデータバイト 1 です。

ビット 7:0 **DATA4[7:0]** : データバイト 4
メッセージのデータバイト 0 です。

29.9.4 CAN フィルタレジスタ

CAN フィルタマスタレジスタ (CAN_FMR)

アドレスオフセット : 0x200
リセット値 : 0x2A1C 0E01

このレジスタのすべてのビットは、ソフトウェアによってセット/クリアされます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CAN2SB[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	FINIT
		rw	rw	rw	rw	rw	rw								rw

ビット 31:14 予約済みであり、リセット値に保持する必要があります。

ビット 13:8 **CAN2SB[5:0]** : CAN2 開始バンク
これらのビットは、ソフトウェアによってセット/クリアされます。CAN2 インタフェース (スレーブ) の開始バンクを 1 から 27 の範囲で定義します。

ビット 7:1 予約済みであり、リセット値に保持する必要があります。

ビット 0 **FINIT** : フィルタ初期化モード
フィルタバンクの初期化モード
0 : アクティブフィルタモード。
1 : フィルタの初期化モード。

CAN フィルタモードレジスタ (CAN_FM1R)

アドレスオフセット : 0x204
リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

注： [図 314：フィルタバンクのスケール設定- レジスタ構造 \(834 ページ\)](#) を参照してください。

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FBMx**：フィルタモード

- フィルタ x のレジスタのモードです。
- 0：フィルタバンク x の 2 つの 32 ビットレジスタは識別子マスクモードになっています。
 - 1：フィルタバンク x の 2 つの 32 ビットレジスタは識別子リストモードになっています。

注： **ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。**

CAN フィルタスケールレジスタ (CAN_FS1R)

アドレスオフセット : 0x20C
リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FSCx**：フィルタスケール設定

- これらのビットは、フィルタ 13-0 のスケール設定を定義します。
- 0：デュアル 16 ビットスケール設定
 - 1：シングル 32 ビットスケール設定

注： **ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。**

注： [図 314：フィルタバンクのスケール設定- レジスタ構造 \(834 ページ\)](#) を参照してください。

CAN フィルタ FIFO 割当てレジスタ (CAN_FFA1R)

アドレスオフセット : 0x214
リセット値 : 0x0000 0000

このレジスタへの書き込みはフィルタ初期化モード (CAN_FMR レジスタの FINIT = 1) のときのみ可能です。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FFA27	FFA26	FFA25	FFA24	FFA23	FFA22	FFA21	FFA20	FFA19	FFA18	FFA17	FFA16
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FFAx** : フィルタ x のフィルタ FIFO 割り当て
このフィルタを通過するメッセージは、特定の FIFO に保存されます。
0 : フィルタが FIFO 0 に割り当てられます。
1 : フィルタが FIFO 1 に割り当てられます。

注 : ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

CAN フィルタアクティベーションレジスタ (CAN_FA1R)

アドレスオフセット : 0x21C
リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FACT27	FACT26	FACT25	FACT24	FACT23	FACT22	FACT21	FACT20	FACT19	FACT18	FACT17	FACT16
				rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FACT15	FACT14	FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW

ビット 31:28 予約済みであり、リセット値に保持する必要があります。

ビット 27:0 **FACTx** : アクティブなフィルタ
ソフトウェアがこのビットをセットしてフィルタ x をアクティブにします。フィルタ x レジスタ (CAN_FxR[0:7]) を変更するには、FACTx ビットをクリアするか、CAN_FMR レジスタの FINIT ビットをセットする必要があります。
0 : フィルタ x はアクティブではありません。
1 : フィルタ x はアクティブです。

注 : ビット 27:14 はコネクティビティ・ライン・デバイスでのみ使用可能で、それ以外の場合は予約済みです。

CAN フィルタバンク i レジスタ x (CAN_FiRx) (i = 0..27、x = 1、2)

アドレスオフセット : 0x240 から 0x31C
リセット値 : 0xFFFF XXXX

28 個のフィルタバンク (i= 0 ~27) が存在します。各フィルタバンク i は、2 個の 32 ビットレジスタ (CAN_FiR[2:1]) で構成されています。

このレジスタを変更できるのは、CAN_FAxR レジスタの FACTx ビットがクリアされている場合か、CAN_FMR レジスタの FINIT ビットがセットされている場合に限られます。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

すべての設定において :

ビット 31:0 **FB[31:0]** : フィルタビット

識別子

レジスタの各ビットは、予期される識別子の対応するビットのレベルを指定します。
0 : ドミナントビットが予期されます。
1 : リセシブビットが予期されます。

マスク

レジスタの各ビットは、関連する識別子レジスタのビットが、予期される識別子の対応するビットに一致する必要があるかどうかを指定します。
0 : 無視 (このビットは比較に使用されません)
1 : 一致が必要 (受信識別子のビットのレベルは、フィルタの対応する識別子レジスタで指定されたレベルと同じでなければなりません)

注 : **フィルタのスケールとモードの設定に応じて、各レジスタの機能は異なることがあります。フィルタ配置、機能の説明、およびマスクレジスタの関連については、[セクション 29.7.4 : 識別子フィルタリング \(832 ページ\)](#) を参照してください。**

マスクモードにおけるマスク／識別子レジスタのビット配置は、**識別子リストモード**の場合と同じです。

フィルタバンクのレジスタ配置／アドレスについては、[表 117 \(864 ページ\)](#) を参照してください。

29.9.5 bxCAN レジスタマップ

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

表 117. bxCAN レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x000	CAN_MCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBF	リセット	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																1	0		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
0x004	CAN_MSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
	リセット値																					1	1	0	0												
0x008	CAN_TSR	LOW[2:0]				TIME[2:0]				CODE[1:0]				ABRQ2				Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2	ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0				
	リセット値	0	0	0	1	1	1	0	0	0					0	0	0	0	0		Res.		0	0	0	0	0	0									
0x00C	CAN_RF0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				
	リセット値																																	FMP0[1:0]			
0x010	CAN_RF1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMP1[1:0]			
	リセット値																																				
0x014	CAN_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE	ERRIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値																0	0	0																		
0x018	CAN_ESR	REC[7:0]								TEC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																			
0x01C	CAN_BTR	SILM	LBKM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
	リセット値	0	0							0	0			0	1	0	0	0	1	1																	
0x020-0x17F		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			
0x180	CAN_TI0R	STID[10:0]/EXID[28:18]										EXID[17:0]																IDE			RTR	TXRQ					
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0				
0x184	CAN_TDT0R	TIME[15:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLC[3:0]		
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x														x	x	x	x		
0x188	CAN_TDL0R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]											
	リセット値	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			

表 117. bxCAN レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x18C	CAN_TDH0R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x190	CAN_TI1R	STID[10:0]/EXID[28:18]												EXID[17:0]																IDE	RTR	TXRQ	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
0x194	CAN_TDT1R	TIME[15:0]																Res	Res	Res	Res	Res	Res	Res	TGT	Res	Res	Res	Res	DLC[3:0]			
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								X					X	X	X	X
0x198	CAN_TDL1R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x19C	CAN_TDH1R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1A0	CAN_TI2R	STID[10:0]/EXID[28:18]												EXID[17:0]																IDE	RTR	TXRQ	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0
0x1A4	CAN_TDT2R	TIME[15:0]																Res	Res	Res	Res	Res	Res	Res	TGT	Res	Res	Res	Res	DLC[3:0]			
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X								X					X	X	X	X
0x1A8	CAN_TDL2R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1AC	CAN_TDH2R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1B0	CAN_RI0R	STID[10:0]/EXID[28:18]												EXID[17:0]																IDE	RTR	Res	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x1B4	CAN_RDT0R	TIME[15:0]																FMI[7:0]								Res	Res	Res	Res	DLC[3:0]			
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	
0x1B8	CAN_RDL0R	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1BC	CAN_RDH0R	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x1C0	CAN_RI1R	STID[10:0]/EXID[28:18]												EXID[17:0]																IDE	RTR	Res	
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

表 117. bxCAN レジスタマップとリセット値（続き）

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x248	CAN_F1R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x24C	CAN_F1R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
.																																	
0x318	CAN_F27R1	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x31C	CAN_F27R2	FB[31:0]																															
	リセット値	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

30 ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)

このセクションは、STM32F04x、STM32F072、および STM32F078 デバイスにのみ適用されます。

30.1 概要

USB ペリフェラルは、フルスピードの USB 2.0 バスと APB バスの間のインタフェースを実装します。

USB サスペンド/レジュームがサポートされるため、節電のためにデバイスのクロックを停止できます。

30.2 USB の主な機能

- USB 仕様バージョン 2.0 フルスピード準拠
- 1 から 8 まで設定可能なエンドポイント数
- 最大 1024 バイトの専用パケットバッファメモリ SRAM (残りの 256 バイトは CAN ペリフェラルとのみ共有される)
- 巡回冗長検査 (CRC) の生成/チェック、NRZI (Non-return-to-zero Inverted) 方式のエンコード/デコード、およびビットスタッフィング
- アイソクロナス転送サポート
- ダブルバッファバルク/アイソクロナスエンドポイントサポート
- USB サスペンド/レジューム操作
- フレームロッククロックパルス生成
- USB 2.0 リンク電源管理サポート
- バッテリ充電仕様 Revision 1.2 サポート
- USB 接続/切断機能 (USB DP ライン上の制御可能な組み込みプルアップレジスタ)

表 118に、デバイスでの USB 実装を示します。

USB の機能⁽¹⁾

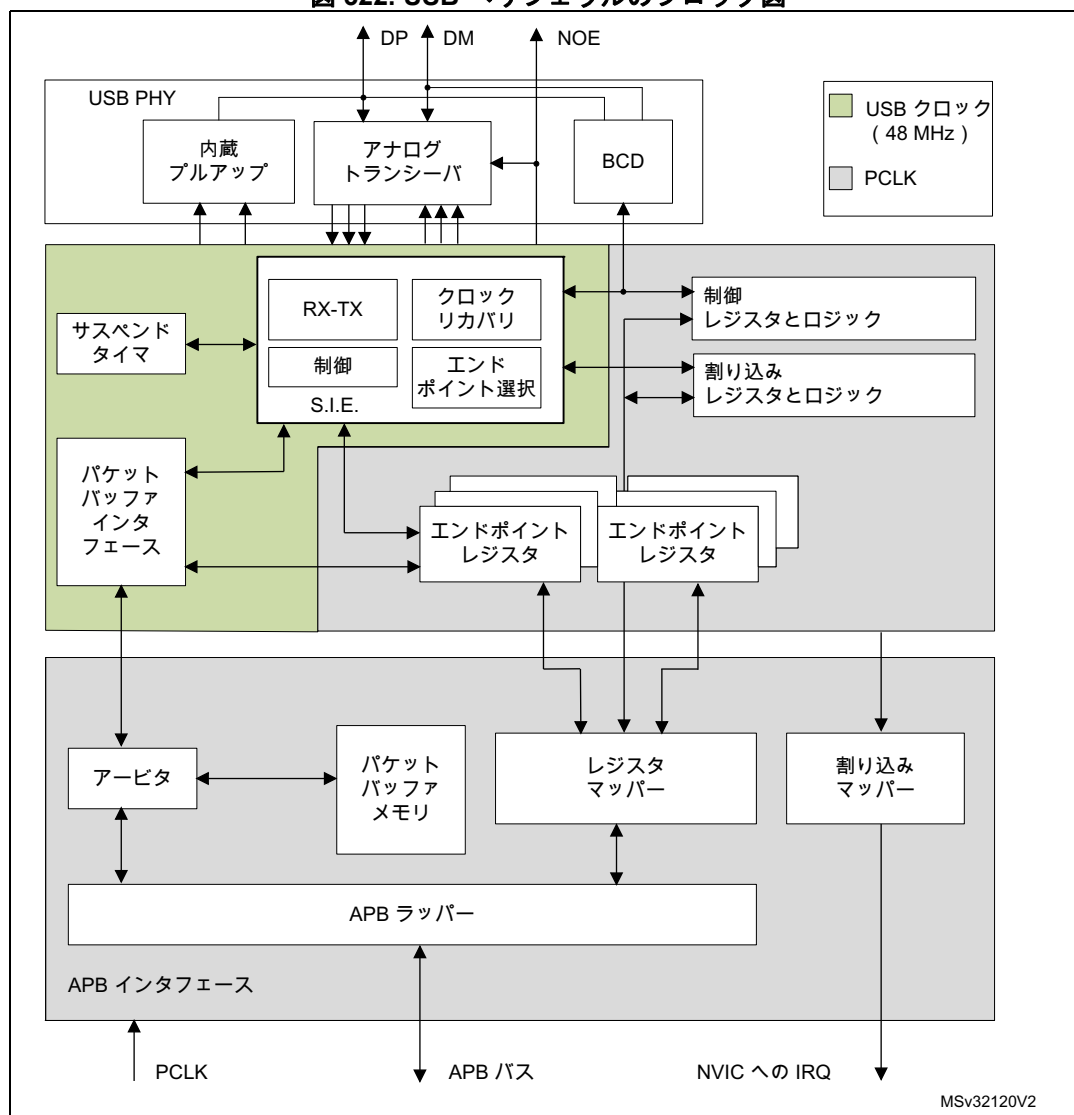
USB の機能 ⁽¹⁾	STM32F04x、STM32F072、STM32F078
	USB
エンドポイント数	8
専用パケットバッファメモリ SRAM のサイズ	1024 バイト ⁽²⁾
専用パケットバッファメモリ SRAM のアクセス方式	2 x 16 ビット/ワード
USB 2.0 リンク電源管理 (LPM) のサポート	X
バッテリー充電検出 (BCD) のサポート	X
USB_DP ライン上の組み込みプルアップレジスタ	X

1. X = サポート。
2. RCC_APB1ENR レジスタで CAN ペリフェラルクロックが有効なときには、最初の 768 のみが USB で使用可能であり、残りの 256 バイトは CAN によって使用されます。

30.4 USART の機能詳細

図 322 に、USB ペリフェラルのブロック図を示します。

図 322. USB ペリフェラルのブロック図



USB ペリフェラルは、ホスト PC とマイクロコントローラによって実装される機能の間に USB 準拠接続を提供します。ホスト PC とシステムメモリ間のデータ転送は、USB ペリフェラルによって直接アクセスされる専用バッファメモリを通じて行われます。この専用メモリのサイズは最大 1024 バイトであり、最大 16 の単方向または 8 つの双方向エンドポイントを使用できます。USB ペリフェラルは USB ホストとインタフェースして、トークンパケットを検出し、データの送受信を処理し、USB 標準の要求に従ってハンドシェイクパケットを処理します。トランザクションのフォーマット化は、CRC の生成とチェックも含めて、ハードウェアによって行われます。

各エンドポイントは、エンドポイント関連のメモリ領域の位置、その大きさ、または送信バイト数を示すバッファ記述ブロックに関連付けられます。有効な機能／エンドポイントのペアのトークンが USB ペリフェラルによって検出されると、関連するデータ転送が行われます（必要であり、エンドポイントが設定されている場合）。USB ペリフェラルによってバッファされたデータは、内部 16 ビットレジスタに配置され、専用バッファへのメモリアccessが行われます。すべてのデータが転送され

ると、必要な場合には、転送方向に従って、適切なハンドシェイクパケットが USB 経由で生成され、または転送の方向によって期待されます。

トランザクションの終了時には、エンドポイント固有の割り込みが生成され、ステータスレジスタを読み出すか、別の割り込み応答ルーチンを使用します。マイクロコントローラは、以下を決めることができます。

- 処理するエンドポイント
- エラーが発生した場合に行うトランザクションのタイプ（ビットスタッフィング、フォーマット、CRC、プロトコル、ACK の欠如、オーバーラン／アンダーランなど）

アイソクロナス転送と高スループットバルク転送に対する特別なサポートが提供され、ダブルバッファ私用が実装されることにより、USB ペリフェラルが使用できるバッファが常にあり、マイクロコントローラ用のバッファが別にあります。

必要なときには、制御レジスタに書き込むことによって、ユニットを低電力モード（SUSPEND モード）にできます。このとき、すべての静的消費電流が回避され、USB クロックをスローダウンまたは停止できます。低電力モード中に USB 入力でアクティビティが検出されると、デバイスは非同期でウェイクアップします。特殊な割り込みソースをウェイクアップラインに直接接続することにより、システムは通常のクロック生成をただちに再開し、直接クロックの開始／停止をサポートできます。

30.4.1 USB ブロックの説明

USB ペリフェラルは USB インタフェースに関するすべての機能を実装し、これには以下のブロックが含まれます。

- USB 物理インタフェース (USB PHY) : このブロックは、外部 USB ホストへの電気インタフェースを維持します。ディファレンシャルアナログトランシーバ本体、制御可能な組み込みプルアップレジスタ (USB_DP ラインに接続)、およびバッテリー充電検出 (BCD) のサポート、同じ USB_DP および USB_DM ラインでの多重化を含みます。アナログトランシーバの出力有効制御信号 (アクティブロー) は、USB_NOE により外部から供給されます。これを使用して、アクティビティ LED を駆動したり、実際の通信方向に関する情報を他の回路に提供することができます。
- シリアルインタフェースエンジン (SIE) : このブロックの機能には、同期パターン認識、ビットスタッフィング、CRC 生成およびチェック、PID 確認／生成、およびハンドシェイク評価が含まれます。USB トランシーバとインタフェースし、ローカルデータストレージ用のパケットバッファインタフェースによって提供される仮想バッファを使用する必要があります。このユニットは、フレーム開始 (SOF)、USB_Reset、データエラーなどの USB ペリフェラルイベントと、送信終了、パケットの正しい受信などのエンドポイント関連イベントに従って信号を生成し、これらの信号は割り込みの生成に使用されます。
- タイマ : このブロックは、フレーム開始ロッククロックパルスを生成し、トラフィックが 3 ms の間、受信されなかったときにはグローバルサスペンドを (ホストから) 検出します。
- パケットバッファインタフェース : このブロックは、ローカルメモリを管理して、送信と受信の両方のためのバッファのセットを柔軟に実装します。SIE から着信したリクエストに従って適切なバッファを選び、エンドポイントレジスタによって示されるメモリアドレスに配置します。パケットの終わりまで、各バイトの交換後にアドレスをインクリメントして、交換バイト数を追跡し、バッファが最大容量までオーバーランするのを防ぎます。
- エンドポイント関連レジスタ : 各エンドポイントには、エンドポイントのタイプと現在のステータスを含んだレジスタが関連付けられます。単方向／シングルバッファエンドポイントの場合、1 つのレジスタで 2 つのエンドポイントを実装できます。レジスタの数は 8 つであり、最大 16 の単方向／シングルバッファまたは最大 7 つのダブルバッファエンドポイントを任意の組み合わせで使用できます。たとえば、USB ペリフェラルをプログラムして、4 つのダブルバッファエンドポイントと 8 つのシングルバッファ／単方向エンドポイントを使用できます。

- 制御レジスタ：これらは USB ペリフェラル全体のステータスに関する情報を含むレジスタであり、レジュームやパワーダウンなどの USB イベントを強制するために使用されます。
- 割り込みレジスタ：これらは割り込みマスクとイベントのレコードを含みます。割り込み理由、割り込みステータスの問い合わせや、保留中の割り込みのステータスをクリアするために使用できます。

注： * エンドポイント 0 は、常にシングルバッファモードでの転送の制御に使用されます。

USB ペリフェラルは APB バスに APB インタフェースを通じて接続され、以下のブロックを含みます。

- パケットメモリ：これは、パケットバッファを物理的に含むローカルメモリです。パケットバッファインタフェースによって使用して、データ構造を作成し、アプリケーションソフトウェアから直接アクセスできます。パケットメモリのサイズは、最大 1024 バイトであり、512 個のハーフワードと 16 ビットで構成されます。
- アービター：このブロックは、APB バスおよび USB インタフェースから着信するリクエストを受け入れます。APB アクセスに優先順位を付けることによってコンフリクトを解決するとともに、常にメモリ帯域幅の半分をすべての USB 転送を完了するために確保しています。このタイムデュプレックス方式によって仮想デュアルポート SRAM が実装され、USB トランザクション中のメモリアccessを可能にします。この方式により、任意の長さのマルチワード APB 転送も可能です。
- レジスタマッパー：このブロックは、USB ペリフェラルのさまざまなバイト長およびビット長のレジスタを APB によってアドレス指定された構造化された 16 ビット長のハーフワードセットに集めます。
- APB ラッパー：これは、メモリとレジスタに APB へのインタフェースを提供します。また、USB ペリフェラル全体を APB アドレス空間にマップします。
- 割り込みマッパー：このブロックは、USB イベントによって割り込みを生成する方法を選択し、それらを NVIC にマップするために使用されます。

30.5 プログラミングに関する考慮事項

以下のセクションでは、アプリケーションソフトウェア開発を容易にするために、USB ペリフェラルとアプリケーションプログラムの間の予期される相互作用について説明します。

30.5.1 汎用 USB デバイスのプログラミング

このパートでは、USB 準拠動作を得るためにアプリケーションソフトウェアに必要な主なタスクについて説明します。ほとんどの汎用 USB イベントに関連するアクションが考慮に入れられており、ダブルバッファエンドポイントとアイソクロナス転送という特殊ケースについて、それぞれのパラグラフで説明します。システムリセットを別にする、アクションは常に USB ペリフェラルによって開始され、以下に述べる USB イベントの 1 つによって駆動されます。

30.5.2 システムリセットとパワーオンリセット

システムリセットおよびパワーオンリセットの際、アプリケーションソフトウェアが最初に行うべき操作は、必要なクロック信号のすべてを USB ペリフェラルに与えることであり、その後、リセット信号をネゲートして、レジスタにアクセスできるようにしなければなりません。初期化シーケンス全体は、以下のとおりです。

最初のステップとして、アプリケーションソフトウェアは、デバースクロック管理ロジックによって提供される関連の制御ビットを使用して、レジスタマクロセルクロックを有効にし、マクロセル固有のリセット信号をネゲートする必要があります。

その後、CNTR レジスタの PDWN ビットを使用して、USB トランシーバに関連するデバイスのアナログ部分をオンにする必要があります、これには特別な処理が必要です。このビットは、ポートトランシーバに供給される内部電圧基準をオンにすることを目的としています。この回路にはスタートアップ時間が定義されており（データシートで指定されている $t_{STARTUP}$ ）、この間の USB トランシーバの動作は定義されていません。したがって、CNTR レジスタの PDWN ビットをセットした後、この時間だけ待つ必要があります、その後、USB 部分のリセット条件を解除します（CNTR レジスタの FRES ビットをクリアすることによって）。ISTR レジスタをクリアすると、他のマクロセル動作が有効になる前に、誤って保留中の割り込みが削除されます。

システムリセット時には、マイクロコントローラは必要なすべてのレジスタとパケットバッファ記述テーブルを初期化して、USB ペリフェラルが割り込みとデータ転送を正しく生成できるようにする必要があります。エンドポイント固有ではないレジスタはすべて、アプリケーションソフトウェアの必要に応じて初期化する必要があります（有効にする割り込みの選択、パケットバッファのアドレスの選択など）。USB リセットの場合、その後、プロセスが続行します（以下のパラグラフを参照）。

USB リセット (RESET 割り込み)

このイベントが発生すると、USB ペリフェラルは、前のパラグラフで説明した初期化後のシステムリセットと同じ条件になります。すなわち、すべてのエンドポイントレジスタで通信が無効になります（USB ペリフェラルはパケットに応答しません）。USB リセットイベントに対して、USB 機能を有効にして、USB アドレスを 0 にし、デフォルトの制御エンドポイントのみを実装します（エンドポイントアドレスも 0）。これは、USB_DADDR レジスタの機能有効 (EF) ビットをセットし、EP0R レジスタに関連するパケットバッファを初期化することによって行われます。USB エン्यूメレーションプロセス時、ホストは一意なアドレスをこのデバイスに割り当て、これが USB_DADDR レジスタの ADD[6:0] ビットに書き込まれなければならない、他に必要なエンドポイントを設定する必要があります。

RESET 割り込みが受信されると、アプリケーションソフトウェアは、割り込みをトリガしたリセットシーケンスの終了から 10 ms 以内に、USB 機能 0 のデフォルトのエンドポイントを再び有効にする必要があります。

パケットバッファの構造と用途

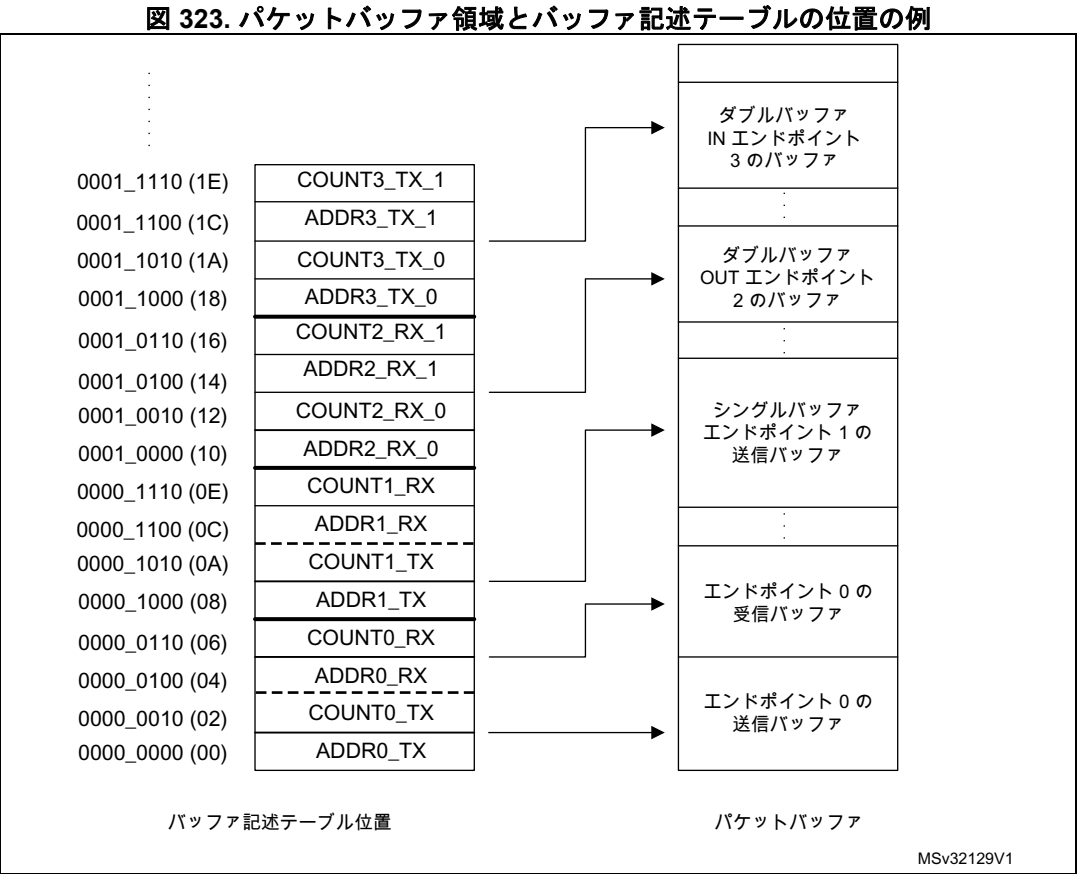
各双方向エンドポイントは、ホストとの間でデータを送受信できます。受信データは、そのエンドポイント用に予約された専用のメモリバッファに格納され、別のメモリバッファにはエンドポイントによって送信されるデータが格納されます。このメモリへのアクセスは、パケットバッファインタフェースブロックによって行われ、これはメモリアクセスリクエストを配信し、確認応答を待ちます。マイクロコントローラもパケットバッファメモリにアクセスする必要があるため、アクセスコンフリクトを避けるためにアービトレーションロジックが使用され、APB サイクルの半分がマイクロコントローラのアクセスに使用され、残りの半分が USB ペリフェラルのアクセスに使用されます。このように、パケットメモリがデュアルポート SRAM であるかのように両方のエージェントが動作でき、マイクロコントローラが連続アクセスを実行中でも、コンフリクトを気にする必要がありません。USB ペリフェラルロジックは、専用クロックを使用します。この専用クロックの周波数は、USB 標準の要件により 48 MHz に固定されており、これは APB バスへのインタフェースに使用されるクロックと異なる場合があります。APB クロック周波数が USB ペリフェラルのクロック周波数より高い、または低い場合、さまざまなクロック設定が可能です。

注： **USB データレートとパケットメモリインタフェース要件により、APB クロックは、データのオーバーラン/アンダーラン問題を回避するために、少なくとも 10 MHz 以上の周波数でなければなりません。**

各エンドポイントには 2 つのパケットバッファが関連付けられます（通常は、送信用に 1 つ、受信用に 1 つ）。バッファの位置とサイズはバッファ記述テーブルで指定されるので、バッファはパケットメモリ内の任意の場所に配置でき、このテーブルも USB_BTABLE レジスタによって示されるパケットメモリ内のアドレスに配置されます。各テーブルエントリがエンドポイントレジスタに関連付けら



れ、4 つの 16 ビットハードワードで構成されるため、テーブル開始アドレスは常に 8 バイトの境界に揃えられます (USB_BTABLE レジスタの下位 3 ビットは常に 000)。バッファディスクリプタテーブルのエントリについては、[セクション 30.6.2 : バッファディスクリプタテーブル](#)で説明します。エンドポイントが単方向であり、アイソクロナスでもダブルバッファバルクでもない場合、必要なパケットバッファは 1 つだけです (サポートされる転送方向に応じた 1 つのバッファ)。サポートされない転送方向または未使用のエンドポイントに関連するテーブル位置は、ユーザが使用できます。アイソクロナスおよびダブルバッファバルクエンドポイントには、パケットバッファの特別な取り扱いがあります (それぞれ、[セクション 30.5.4 : アイソクロナス転送](#) および [セクション 30.5.3 : ダブルバッファエンドポイント](#)を参照)。バッファ記述テーブルのエントリとパケットバッファ領域の関係を [図 323](#) に示します。



各パケットバッファは、始めから終わりまで、受信または送信のいずれかに使用されます。USB ペリフェラルは、割り当てられたメモリバッファに隣接するメモリ位置の内容を変えることはありません。割り当てられたバッファ長を超えるパケットが受信された場合 (バッファオーバーラン条件)、使用可能な最期の位置までしか、データはメモリにコピーされません。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

エンドポイントの初期化

エンドポイントを初期化するための最初のステップは、ADDRn_TX/ADDRn_RX レジスタに適切な値を書き込んで、送信データがすでに使用可能になったことと、受信データをバッファできることを USB ペリフェラルに知らせることです。USB_EPnR レジスタの EP_TYPE ビットをエンドポイントのタイプに応じてセットし、最終的には EP_KIND ビットを使用して、特に必要な機能を有効にする必要があります。送信側では、USB_EPnR レジスタの STAT_TX ビットを使用してエンドポイントを有効にする必要があります。受信の場合、STAT_RX ビットをセットして受信を有効にする必要があります。BL_SIZE および NUM_BLOCK フィールドを使用して、割り当てられたバッファサイズを COUNTn_RX に書き込む必要があります。単方向エンドポイントは、アイソクロナスおよびダブルバッファバルクエンドポイントを除き、サポートされる方向に関するビットとレジスタのみを初期化する必要があります。送信または受信が有効になると、レジスタ USB_EPnR と位置 ADDRn_TX/ADDRn_RX、COUNTn_TX/COUNTn_RX（それぞれ）をアプリケーションソフトウェアで変更しないでください。これらの値はハードウェアが直接変更できます。データ転送操作が完了し、CTR 割り込みイベントによって通知されると、再びアクセスできるようになり、新しい操作を再び有効にできます。

IN パケット（データ送信）

IN トークンパケットを受信するとき、受信したアドレスが設定済みの有効なエンドポイントと一致した場合、USB ペリフェラルは、アドレス指定されたエンドポイントに関連するバッファディスクリプタテーブルエントリ内の ADDRn_TX および COUNTn_TX 位置にアクセスします。これらの位置の内容は、内部 16 ビットレジスタ ADDR および COUNT に格納されます（ソフトウェアからはアクセスできません）。パケットメモリが再びアクセスされ、最初の送信バイトを読み出し（[パケットバッファの構造と用途 \(873 ページ\)](#)）を参照）、USB_EPnR のビット DTOG_TX に従って、DATA0 または DATA1 PID の送信を開始します。PID が完了すると、バッファメモリから読み出された最初のバイトが出力シフトレジスタにロードされ、USB バスで送信されます。最後のデータバイトが送信された後、計算された CRC が送信されます。アドレス指定されたエンドポイントが有効でなかった場合は、USB_EPnR レジスタの STAT_TX ビットに従って、NAK または STALL ハンドシェイクパケットがデータパケットの代わりに送信されます。

ADDR 内部レジスタは、現在のバッファメモリ位置を指すポインタとして使用され、COUNT は、残りの送信バイト数をカウントするために使用されます。パケットバッファメモリから読み出された各ハーフワードは、最下位バイトから順に USB バスで送信されます。送信バッファメモリは、COUNTn_TX/2 ハーフワードの場合、ADDRn_TX によって示されるアドレスから順に読み出されます。送信パケットが奇数個のバイトで構成される場合、最後にアクセスされたハーフワードの下位半分のみが使用されます。

ホストが ACK を受信した場合、USB_EPnR レジスタは次のように更新されます。DTOG_TX ビットがトグルされ、STAT_TX=10 (NAK) をセットすることによってエンドポイントが無効化され、ビット CTR_TX がセットされます。アプリケーションソフトウェアは、まず、USB_ISTR レジスタの EP_ID および DIR ビットを調べることによって、マイクロコントローラの処理を要求しているエンドポイントを識別する必要があります。CTR_TX イベントの処理は、割り込みビットをクリアすることから始まります。次に、アプリケーションソフトウェアは、送信データでいっぱい別のバッファを準備し、COUNTn_TX テーブル位置を次の転送時に送信されるバイト数で更新し、最後に、STAT_TX を 11 (VALID) にセットして、送信を再び有効にします。STAT_TX ビットが 10 (NAK) に等しいときには、そのエンドポイントにアドレス指定された IN リクエストは NAK され、フロー制御条件を示します。USB ホストは、成功するまで、トランザクションを再試行します。CTR 割り込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された 2 番目の IN トランザクションの通知が失われるのを避けるためには、一連の操作を上記の順序で実行する必要があります。

OUT および SETUP パケット (データ受信)

この 2 つのトークンは、USB ペリフェラルによっておおむね同じように扱われます。SETUP パケットの取り扱いにおける違いについては、制御転送に関する以下のパラグラフで詳しく述べます。OUT/SETUP PID を受信するとき、アドレスが有効なエンドポイントと一致した場合、USB ペリフェラルは、アドレス指定されたエンドポイントに関連するバッファディスクリプタテーブルエントリ内の ADDRn_RX および COUNTn_RX 位置の内容にアクセスします。ADDRn_RX の内容は、内部レジスタ ADDR に直接格納されます。COUNT がリセットされ、BL_SIZE および NUM_BLOCK ビットフィールドの値が COUNTn_RX 内に読み出されると、それを使用して BUF_COUNT が初期化され、これは、バッファオーバーラン条件のチェックに使用される内部 16 ビットカウンタです (これらの内部レジスタはすべて、ソフトウェアからはアクセスできません)。その後、USB ペリフェラルによって受信されるデータバイトは、ハーフワード単位でパケット化され (最初の受信バイトが最下位バイトとして格納されます)、内部 ADDR レジスタに含まれるアドレスから順にパケットバッファに転送され、各バイト転送により、BUF_COUNT はデクリメントされ、COUNT はインクリメントされます。DATA パケットの終わりが検出されると、受信された CRC の正しさがテストされ、受信時にエラーが発生しなかった場合のみ、ACK ハンドシェイクパケットが送信元ホストに送り返されます。

CRC が正しくなかった場合、または他の種類のエラー (ビットスタフ違反、フレームエラーなど) が発生した場合、データバイトは、少なくともエラー検出ポイントまではパケットメモリにコピーされますが、ACK パケットは送信されず、USB_ISTR レジスタの ERR ビットがセットされます。ただし、この場合、通常はソフトウェアのアクションは不要です。USB ペリフェラルは受信エラーから回復し、次のトランザクションを処理する準備ができた状態を保ちます。アドレス指定されたエンドポイントが有効でなかった場合は、USB_EPnR レジスタの STAT_RX ビットに従って、NAK または STALL ハンドシェイクパケットがデータパケットの代わりに送信され、データは受信メモリバッファに書き込まれません。

受信メモリバッファの位置は、ADDRn_RX に含まれるアドレスから、CRC を含めた受信データパケット長に対応するバイト数 (データペイロード長 + 2) だけ、または、BL_SIZE および NUM_BLOCK によって定義された割り当て済みの最後のメモリ位置のいずれか先に来る方まで書き込まれます。このように、USB ペリフェラルは、割り当てられた受信メモリバッファ領域の終わりを越えて書き込むことはありません。データパケットペイロードの長さ (アプリケーションによって使用された実際のバイト数) が割り当てられたバッファより大きい場合、USB ペリフェラルはバッファオーバーラン条件を検出します。この場合、通常の ACK の代わりに STALL ハンドシェイクが送信されて、問題をホストに通知し、割り込みは生成されず、トランザクションは失敗したとみなされます。

ACK ハンドシェイクパケットを送信することにより、トランザクションが正常に完了すると、内部 COUNT レジスタがバッファ記述テーブルエントリ内の COUNTn_RX 位置にコピーされ、影響を受けなかった BL_SIZE および NUM_BLOCK フィールドはそのままに保たれ (通常は再書き込みの必要はありません)、USB_EPnR レジスタが次のように更新されます。DTOG_RX ビットがトグルされ、STAT_RX=10 (NAK) をセットすることによってエンドポイントが無効化され、ビット CTR_RX がセットされます。エラーまたはバッファオーバーランによってトランザクションが失敗した場合、上記のアクションはいずれも実行されません。アプリケーションソフトウェアは、まず、USB_ISTR レジスタの EP_ID および DIR ビットを調べることによって、マイクロコントローラの処理を要求しているエンドポイントを識別する必要があります。CTR_RX イベントの処理は、まず、トランザクションのタイプを決定することによって行われます (USB_EPnR レジスタの SETUP ビット)。アプリケーションソフトウェアは、割り込みフラグビットをクリアして、処理されるエンドポイントに関連するバッファ記述テーブルエントリ内の COUNTn_RX 位置を読み出して、受信バイト数を取得する必要があります。受信データが処理された後、アプリケーションは USB_EPnR の STAT_RX ビットを 11 (有効) にセットして、次のトランザクションを有効にする必要があります。STAT_RX ビットが 10 (NAK) に等しいときには、そのエンドポイントにアドレス指定された OUT リクエストは NAK され、フロー制御条件を示します。USB ホストは、成功するまで、トランザクションを再試行します。CTR 割り込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された 2 番目の OUT トランザクションの通知が失われるのを避けるためには、一連の操作を上記の順序で実行する必要があります。

制御転送

制御転送は、SETUP トランザクション、すべて同じ方向のゼロ個以上のデータステージ、およびステータスステージ（反対方向のゼロバイトの転送）で構成されます。SETUP トランザクションは制御エンドポイントのみによって処理され、OUT トランザクション（データ受信）と非常に似ていますが、制御転送を初期化するために、アドレス指定されたエンドポイントレジスタの DTOG_TX および DTOG_RX ビットの値がそれぞれ 1 と 0 にセットされる点と、ソフトウェアが後続のトランザクションが IN か OUT かを SETUP の内容に応じて決定できるように、STAT_TX と STAT_RX の両方が 10 (NAK) にセットされる点が異なります。制御エンドポイントは CTR_RX イベントごとに USB_EPnR レジスタの SETUP ビットをチェックして、通常の OUT トランザクションと SETIP トランザクションを区別する必要があります。USB デバイスは、SETUP ステージで転送されたデータを解釈することによって、データステージの数と方向を判断でき、エラーの場合にはトランザクションを STALL する必要があります。そのためには、最後以外のすべてのデータステージで、未使用の方向を STALL にセットして、ホストが転送方向を反転するのが速過ぎた場合にはステータスステージとして STALL されるようにする必要があります。

最後のデータステージを有効にするときには、反対方向を NAK にセットして、ホストが（ステータスステージを実行するために）転送方向をすぐに反転した場合、制御操作の完了を待つようにする必要があります。制御操作が正常に完了した場合、ソフトウェアは NAK を VALID に変更し、そうでない場合は STALL に変更します。同時に、ステータスステージが OUT の場合、STATUS_OUT (USB_EPnR レジスタの EP_KIND) ビットをセットして、ステータストランザクションが非ゼロのデータで実行された場合はエラーが生成されるようにする必要があります。ステータストランザクションが処理されると、アプリケーションは STATUS_OUT ビットをクリアし、STAT_RX を VALID にセットし（新しいコマンドを受け入れるため）、STAT_TX を NAK にセットします（次のセットアップの直後のステータスステージを遅延するため）。

USB 仕様では、SETUP パケットには ACK 以外のハンドシェイクでは応答できず、結果として、以前に発行されたコマンドをアボートして新しいコマンドを開始すると規定されているため、USB ロジックでは、制御エンドポイントはホストから受信した SETUP トークンに対して NAK または STALL パケットで応答することはできません。

STAT_RX ビットが 01 (STALL) または 10 (NAK) にセットされ、SETUP トークンが受信されると、USB はデータを受け入れ、要求されたデータ転送を実行し、ACK ハンドシェイクを返送します。そのエンドポイントに以前に発行された CTR_RX リクエストがあり、アプリケーションによってまだ確認応答されていない場合（すなわち、CTR_RX ビットが以前に完了した受信からセットされたままになっている場合）、USB は SETUP トランザクションを破棄し、状態にかかわらずハンドシェイクパケットで応答せず、受信エラーをシミュレートして、ホストが SETIP トークンを再び送信しなければならないようにします。これは、CTR_RX 割り込みをトリガしたトランザクションの直後に、同じエンドポイントにアドレス指定された SETUP トランザクションの通知が失われるのを避けるために行われます。

30.5.3 ダブルバッファエンドポイント

USB 標準によって定義されているすべてのエンドポイントタイプは、さまざまなトラフィックモデルを表し、各種のデータ転送操作の典型的な要件を記述しています。データの大部分をホスト PC と USB 機能の間で転送する必要があるときには、バルクエンドポイントタイプが最適のモデルです。ホストはバルクトランザクションをスケジュールして、フレーム内の使用可能な帯域幅のすべてをいっぱいにして、アドレス指定された USB 機能がバルクトランザクションを処理できる限り、実際の転送速度を最大化します。次のトランザクションが着信したときに、USB 機能が前のトランザクションでビジーであった場合は、NAK ハンドシェイクで応答し、ホスト PC は、USB 機能が処理できるようになるまで、同じトランザクションを再び発行するので、再送信によって帯域幅が占有されるため、実際の転送速度は低下します。このため、バルクエンドポイントでは、「ダブルバッファリング」と呼ばれる専用機能を使用できます。



「ダブルバッファリング」が有効なときには、データトグルシーケンシングが使用されて、USB ペリフェラルが要求されたデータ転送を実行するために使用するバッファが選択され、送信と受信の両方のパケットメモリ領域を使用して、トランザクションの成功ごとのバッファスワッピングを管理するため、アプリケーションは常に完全なバッファを使用でき、USB ペリフェラルは他方のバッファをいっぱいにします。たとえば、「受信」ダブルバッファバルクエンドポイント宛ての OUT トランザクション時、1 つのバッファが USB ホストから着信する新しいデータで満たされている間、もう 1 つのバッファはマイクロコントローラソフトウェアが使用できます（「送信」ダブルバッファバルクエンドポイントと IN トランザクションでも同様です）。

スワップバッファ管理には、アドレスポイントと割り当て済みメモリバッファの長さをホストする 4 つすべてのバッファ記述テーブル位置を使用する必要があるため、ダブルバッファバルクエンドポイントの実装に使用される USB_EPnR レジスタは、強制的に単方向として使用されます。したがって、1 つの STAT ビットのみを 00（無効）以外の値にセットする必要があります。ダブルバッファバルクエンドポイントが受信について有効な場合は STAT_RX、ダブルバッファバルクエンドポイントが送信について有効な場合は STAT_TX。ダブルバッファバルクエンドポイントを受信と送信の両方について有効にする必要がある場合は、2 つの USB_EPnR レジスタを使用する必要があります。

ダブルバッファリング機能を利用して、可能な限り高い転送速度を達成するには、前の章で説明したエンドポイントフロー制御構造を変更して、トランザクションの正常官僚の旅ではなく、USB ペリフェラルとアプリケーションの間でバッファコンフリクトが発生したときのみ、エンドポイントのステータスを NAK に切り替える必要があります。USB ペリフェラルによって現在使用されているメモリバッファは、エンドポイント方向に関連する DTOG ビットによって定義されます。すなわち、「受信」ダブルバッファバルクエンドポイントの場合は DTOG_RX（USB_EPnR レジスタのビット 14）、または「送信」ダブルバッファバルクエンドポイントの場合は DTOG_TX（USB_EPnR レジスタのビット 6）です。新しいフロー制御方式を実装するには、USB ペリフェラルはアプリケーションソフトウェアによって使用されているパケットバッファを知り、コンフリクトに注意する必要があります。USB_EPnR レジスタには 2 つの DTOG ビットがありますが、データおよびバッファシーケンシングのために USB ペリフェラルによって使用されるのは 1 つだけであり（ダブルバッファリング機能に必要な単方向成約のため）、もう 1 つはアプリケーションソフトウェアが、使用中のバッファを示すために使用できます。この新しいバッファフラグは、SW_BUF と呼ばれます。次の表で、「送信」および「受信」ダブルバッファバルクエンドポイントの場合の USB_EPnR レジスタのビットと DTOG/SW_BUF 定義の対応を説明します。

表 119. ダブルバッファリングバッファフラグの定義

バッファフラグ	「送信」エンドポイント	「受信」エンドポイント
DTOG	DTOG_TX (USB_EPnR ビット 6)	DTOG_RX (USB_EPnR ビット 14)
SW_BUF	USB_EPnR ビット 14	USB_EPnR ビット 6

USB ペリフェラルによって現在使用されているメモリバッファは、DTOG バッファフラグによって定義され、アプリケーションソフトウェアによって現在使用されているバッファは、SW_BUF バッファフラグによって示されます。バッファフラグの値と使用されているパケットバッファの関係は、どちらの場合も同じであり、次の表に示されています。

表 120. バルクダブルバッファリングメモリバッファの用途

エンド ポイント タイプ	DTOG	SW_BUF	USB ペリフェラルによって 使用されるパケットバッファ	アプリケーションソフトウェアに よって使用されるパケットバッファ
IN	0	1	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル位置。
	1	0	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル位置	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。
	0	0	なし ⁽¹⁾	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。
	1	1	なし ⁽¹⁾	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル位置。
OUT	0	1	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置。	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置。
	1	0	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置。	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置。
	0	0	なし ⁽¹⁾	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル位置。
	1	1	なし ⁽¹⁾	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル位置。

1. NAK ステータスのエンドポイント。

バルクエンドポイントのダブルバッファリング機能は、次のようにして有効化されます。

- USB_EPnR レジスタの EP_TYPE ビットフィールドに 00 を書き込み、エンドポイントをバルクとして定義します。
- 同じレジスタの EP_KIND ビットを 1 (DBL_BUF) にセットします。

アプリケーションソフトウェアは、最初に使用されるバッファに応じて、DTOG および SW_BUF ビットを初期化する責任があります。これは、この 2 つのビットが持つ特殊なトグル専用プロパティを考慮して行う必要があります。DBL_BUF のセット後に発生した初めてのトランザクションの終了により、ダブルバッファバルクエンドポイントの特殊なフロー制御がトリガされ、DBL_BUF がセットされたままである限り、このエンドポイントにアドレス指定された他のすべてのトランザクションについて、これが使用されます。各トランザクションの終了時、有効化された方向に応じて、アドレス指定されたエンドポイント USB_EPnR レジスタの CTR_RX または CTR_TX ビットがセットされます。同時に、USB_EPnR レジスタの影響を受けた DTOG ビットがハードウェアによってトグルされ、USB ペリフェラルのバッファスワッピングを完全にソフトウェアから独立させます。通常のトランザクション、および DBL_BUF のセット後の初めてのトランザクションと違って、STAT ビットペアはトランザクション終了の影響を受けず、値は 11 (有効) のままです。ただし、新しいトランザクションのトークンパケットが受信されると、実際のエンドポイントステータスは、USB ペリフェラルとアプリケーションソフトウェアの間のバッファコンフリクトが検出されたときには 10 (NAK) としてマスクされます (この条件は DTOG と SW_BUF が同じ値を持つことによって示されます。表 120 (879 ページ) を参照)。アプリケーションソフトウェアは、CTR イベント通知にตอบสนองして、割り込みフラグをクリアし、完了したトランザクションの要求された処理を開始します。アプリケーションパケットバッファの仕様が終了すると、ソフトウェアは SW_BUF ビットに 1 を書き込んでトグルし、そのバッファが使用可能になったことを USB ペリフェラルに通知します。このように、NAK されるトランザクションの数は、トランザクションデータのアプリケーション得ラボレーション時間によってのみ制限されます。得ラボレーション時間が USB バスでのトランザクションの完了に必要な時間

より短かった場合、フロー制御による再送信は行われず、実際の転送速度はホスト PC によってのみ制限されます。

アプリケーションソフトウェアは、11 (有効) とは異なる明示的なステータスを、関連する USB_EPnR レジスタの STAT ビットペアに書き込むことによって、ダブルバッファバULKエンドポイントのために実装される特殊なフロー制御をオーバーライドできます。この場合、USB ペリフェラルは、バッファ使用条件に関係なく、プログラムされたエンドポイントステータスを常に使用します。

30.5.4 アイソクロナス転送

USB 標準は、固定された正確なデータ生成/消費周波数を必要とするフルスピードペリフェラルをサポートし、この種類のトラフィックを「アイソクロナス」として定義しています。このデータの典型的な例は、オーディオサンプル、圧縮ビデオストリーム、および一般に、供給される周波数の正確さについて厳密な要件を持つ、あらゆる種類のサンプルデータです。エニュメレーションフェーズでエンドポイントが「アイソクロナス」として定義されると、ホストは必要な帯域幅をフレームに割り当て、エンドポイントの方向に応じて、各フレームに正確に 1 つの IN または OUT パケットを供給します。帯域幅要件を制限するために、アイソクロナストラフィックについては、失敗したトランザクションの再送信はできません。このため、アイソクロナストランザクションにはハンドシェイクフェーズはなく、ACK パケットは予期されず、データパケット後に送信されません。同じ理由で、アイソクロナス転送はデータグルーピングをサポートせず、常に DATA0 PID を使用してデータパケットを開始します。

エンドポイントのアイソクロナス動作は、USB_EPnR レジスタの EP_TYPE ビットを 10 にセットすることによって選択されます。ハンドシェイクフェーズがないため、STAT_RX/STAT_TX ビットの唯一の正当な値は 00 (無効) と 11 (有効) であり、他の値の場合は、USB 標準に準拠しない結果になります。アイソクロナスエンドポイントはダブルバッファリングを実装して、アプリケーションソフトウェア開発を容易にし、「送信」と「受信」の両方のパケットメモリ領域を使用して、トランザクションの成功のたびにバッファスワッピングを管理し、アプリケーションは常に完全なバッファを使用でき、USB ペリフェラルはもう 1 つのバッファを満たします。

USB ペリフェラルによって現在使用されているメモリバッファは、エンドポイント方向に関する DTOG ビットによって、表 121 に従って定義されます (「受信」アイソクロナスエンドポイントの場合は DTOG_RX、「送信」アイソクロナスエンドポイントの場合は DTOG_TX であり、両方とも、関連する USB_EPnR レジスタ)。

表 121. アイソクロナスメモリバッファの使用

エンド ポイント タイプ	DTOG ビットの値	USB ペリフェラルによって使用される パケットバッファ	アプリケーションソフトウェアによって 使用されるパケットバッファ
IN	0	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル 位置	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル 位置
	1	ADDRn_TX_1 / COUNTn_TX_1 バッファ記述テーブル 位置	ADDRn_TX_0 / COUNTn_TX_0 バッファ記述テーブル 位置
OUT	0	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル 位置	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル 位置
	1	ADDRn_RX_1 / COUNTn_RX_1 バッファ記述テーブル 位置	ADDRn_RX_0 / COUNTn_RX_0 バッファ記述テーブル 位置

ダブルバッファアイソクロナスエンドポイントで発生すると、アイソクロナスエンドポイントの実装に使用される USB_EPnR レジスタは強制的に単方向として使用されます。アイソクロナスエンドポイントを受信と送信の両方について有効にする必要がある場合は、2 つの USB_EPnR レジスタを使用する必要があります。

アプリケーションソフトウェアは、最初に使用されるバッファに応じて、DTOG ビットを初期化する責任があります。これは、この 2 つのビットが持つ特殊なトグル専用プロパティを考慮して行う必要があります。各トランザクションの終了時、有効化された方向に応じて、アドレス指定されたエンドポイント USB_EPnR レジスタの CTR_RX または CTR_TX ビットがセットされます。同時に、USB_EPnR レジスタの影響を受けた DTOG ビットがハードウェアによってトグルされ、バッファスワッピングを完全にソフトウェアから独立させます。STAT ビットペアはトランザクションの完了に影響を受けません。アイソクロナス転送ではハンドシェイクフェーズがないため、フロー制御は不可能であり、エンドポイントは常に 11 (有効) のままです。CRC エラーまたはバッファオーバーラン条件がアイソクロナス OUT 転送時に発生した場合、常に正しいトランザクションとみなされ、常に CTR_RX イベントをトリガします。ただし、CRC エラーが発生すると、常に USB_ISTR レジスタの ERR ビットがセットされて、データ破損の可能性をソフトウェアに通知します。

30.5.5 サスペンド／レジュームイベント

USB 標準では、SUSPEND と呼ばれる特殊なペリフェラル状態が定義され、USB バスからの平均電流は 2.5 mA より大きくなってはなりません。この要件はバスパワードデバイスにとって基本的な重要性を持ちますが、セルフパワードデバイスは、この厳格な消費電力成約を順守する必要はありません。サスペンドモードでは、ホスト PC は 3 ms を超えて USB バスにトラフィックを送信しないことによって通知を送信します。SOF パケットは、通常動作では 1 ms ごとに送信されなければならないので、USB ペリフェラルは 3 回連続の SOF パケットの欠落をホスト PC からのサスペンドリクエストとして検出し、USB_ISTR レジスタの SUSP ビットを 1 にセットし、有効な場合は割り込みが生成されます。デバイスがサスペンドされると、いわゆる RESUME シーケンスによって通常操作が復元され、これはホスト PC から、またはペリフェラルそのものから直接開始できますが、常にホスト PC によって終了されます。サスペンドされた USB ペリフェラルは、常に RESET シーケンスを検出して、通常の USB リセットイベントとして、このイベントに対応できなければなりません。

USB ペリフェラルをサスペンドするために使用される実際の手順は、デバイスの構成によって合計消費量を削減するために必要なアクションが異なるため、デバイスに依存します。

USB ペリフェラルの SUSP 通知に応答するアプリケーションソフトウェアルーチンの USB 関連の局面を中心にして、典型的なサスペンド手順の簡単な説明を以下に示します。

1. USB_CNTR レジスタの FSUSP ビットを 1 にセットします。このアクションによって、USB ペリフェラル内のサスペンドモードが有効になります。サスペンドモードが有効になると、USB がサスペンドされている間に SUSP 割り込みが発行されるのを回避するために、SOF 受信時のチェックが無効になります。
2. USB ペリフェラルとは異なるブロックの静的消費電力を除去または削減します。
3. USB_CNTR レジスタの LP_MODE ビットを 1 にセットして、アナログ USB トランシーバの静的消費電力を除去しつつ、レジュームアクティビティを検出できるままにしておきます。
4. オプションで、外部オシレータとデバイス PLL をオフにして、デバイス内部のアクティビティを停止します。

デバイスが SUSPEND モード中に USB イベントが発生したときには、RESUME 手順が呼び出されて、通常のチェックと復元し、通常の USB 動作を復元する必要があります。ウェイクイベントが USB リセットシーケンスである時には、このプロセスが 10 ms 以内で終了するように特に注意が必要です (詳細については、「ユニバーサルシリアルバス仕様」を参照してください)。USB ペリフェラルのサスペンド中にレジュームまたはリセットシーケンスが開始されると、USB_CNTR レジスタの LP_MODE ビットが非同期でクリアされます。このイベントによって WKUP 割り込みがトリガされた場合でも、システムクロックの再開による長い遅延のため、割り込み応答ルーチンの使用は慎重に評価する必要があります。公称クロックの再有効化の前の遅延を短くするには、サスペンド手順の終了直後にレジューム手順を行って、システムクロックが再開したらすぐにコードが実行されるようにします。システムをウェイクアップする際の ESD 放電またはその他の種類のノイズを防止するために (サスペンドモードの終了は、非同期イベントです)、サスペンド中にデータラインステータスの適切なアナログフィルタを有効にします。フィルタ幅は約 70 ns です。

以下は、レジューム手順で行うべきアクションのリストです。

- 1. オプションで、外部オシレータおよびデバイス PLL をオンにします。
- 2. USB_CNTR レジスタの FSUSP ビットをクリアします。
- 3. レジュームをトリガしたイベントを識別する必要がある場合は、USB_FNR レジスタのビット RXDP および RXDM を 表 122 に従って使用でき、この表には、あらゆる場合のソフトウェアのアクションもリストされています。必要な場合には、上記のビットのステータスを監視することによって、レジュームまたはリセットシーケンスの終了を検出でき、これらが「10」設定 (アイドルバス状態を表します) に達したことをチェックします。さらに、リセットシーケンスの終了時には、USB_ISTR レジスタの RESET ビットが 1 にセットされ、有効な場合は割り込みが発行され、通常通りに処理されます。

表 122. レジュームイベント検出

[RXDP,RXDM] ステータス	ウェイクアップイベント	必要なレジュームソフトウェアアクション
"00"	ルートのリセット	なし
"10"	なし (バス上のノイズ)	サスペンドモードに戻る
"01"	ルートのレジューム	なし
"11"	不可 (バス上のノイズ)	サスペンドモードに戻る

デバイスは、USB プロトコルに直接関係しない特定のイベントに対する応答として、サスペンドモード化を終了しなければならない場合があります (たとえば、マウスの移動はシステム全体をウェイクアップします)。この場合、USB_CNTR レジスタの RESUME ビットを 1 にセットし、1 ms から 15 ms までのインターバル後に 0 にリセットすることによって、レジュームシーケンスを開始できます (このインターバルは、システムクロックが公称周波数で動作しているときには 1 ms 周期で発生する ESOF 割り込みによって計測できます)。RESUME ビットがクリアされると、レジュームシーケンスはホスト PC によって完了され、その終了は、USB_FNR レジスタの RXDP および RXDM ビットを使用して監視できます。

注: RESUME ビットは、USB ペリフェラルがサスペンドモードにされた (USB_CNTR レジスタの FSUSP ビットを 1 にセットした) 後でのみ使用する必要があります。



30.6 USBレジスタ

USB ペリフェラルレジスタは、以下のグループに分けることができます。

- 共通レジスタ：割り込みおよび制御レジスタ。
- エンドポイントレジスタ：エンドポイント設定およびステータス
- バッファディスクリプタテーブル：データバッファの配置に使用されたパケットメモリの位置

すべてのレジスタアドレスは、USB ペリフェラルレジスタのベースアドレス 0x4000 5C00 に対するオフセットとして表されますが、バッファディスクリプタテーブル位置は USB_BTABLE レジスタによって指定されたアドレスから始まります。

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

ペリフェラルレジスタには、ハーフワード（16 ビット）またはワード（32 ビット）単位でアクセスする必要があります。

30.6.1 共通レジスタ

これらのレジスタは USB ペリフェラルの全般的動作に影響し、動作モード、割り込み処理、デバイスアドレス、およびホスト PC によって更新される現在のフレーム番号へのアクセス付与を定義します。

USB 制御レジスタ (USB_CNTR)

アドレスオフセット：0x40

リセット値：0x0003

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	L1REQM	Res	L1RESUME	RESUME	FSUSP	LP_MODE	PDWN	FRES
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w

ビット 15 **CTRM**：正しい転送割り込みマスク

- 0：正しい転送（CTR）割り込みは無効です。
- 1：CTR 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 14 **PMAOVRM**：パケットメモリ領域のオーバー／アンダーラン割り込みマスク

- 0：PMAOVR 割り込みは無効です。
- 1：PMAOVR 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 13 **ERRM**：エラー割り込みマスク

- 0：ERR 割り込みは無効です。
- 1：ERR 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 12 **WKUPM**：ウェイクアップ割り込みマスク

- 0：WKUP 割り込みは無効です。
- 1：WKUP 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 11 SUSPM : サスペンドモード割り込みマスク

- 0 : サスペンドモードリクエスト (SUSP) 割り込みは無効です。
- 1 : SUSP 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 10 RESETM : USB リセット割り込みマスク

- 0 : RESET 割り込みは無効です。
- 1 : RESET 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 9 SOFM : フレーム開始割り込みマスク

- 0 : SOF 割り込みは無効です。
- 1 : SOF 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 8 ESOFM : 予期されるフレーム開始割り込みマスク

- 0 : 予期されるフレーム開始 (ESOF) 割り込みは無効です。
- 1 : ESOF 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 7 L1REQM : LPM L1 状態リクエスト割り込みマスク

- 0 : LPM L1 状態リクエスト (L1REQ) 割り込みマスク無効
- 1 : L1REQ 割り込みは有効であり、USB_ISTR レジスタの対応するビットがセットされたときには割り込みリクエストが生成されます。

ビット 6 予約済み。**ビット 5 L1RESUME** : LPM L1 レジュームリクエスト

マイクロコントローラはこのビットをセットして、ホストに LPM L1 レジューム信号を送信できます。このビットは、信号の送信終了後にハードウェアによってクリアされます。

ビット 4 RESUME : レジュームリクエスト

マイクロコントローラはこのビットをセットして、ホストにレジューム信号を送信できます。USB 仕様に従って、ホスト PC がレジュームシーケンスを終了まで駆動する準備ができてから 1 ms 以上 15 ms 以下で有効にする必要があります。

ビット 3 FSUSP : 強制サスペンド

- ソフトウェアは、SUSP 割り込みが受信されたときにこのビットをセットする必要があるため、USB ペリフェラルが 3 ms 間、トラフィックを受信しなかったときに発行されます。
- 0 : 影響はありません。
- 1 : サスペンドモードに入ります。アナログトランシーバのクロックと静的消費電流は影響を受けません。サスペンド消費電力が要件の場合 (バスパワーデバイス)、アプリケーションソフトウェアは FSUSP の後、以下の説明に従って LP_MODE ビットをセットする必要があります。

ビット 2 **LP_MODE** : 低電力モード

このモードは、サスペンドモードの電力制約から、外部プルアップレジスタに供給される電力を除き、すべての静的消費電流を回避されるときに使用されます。USB サスペンド条件の消費電力要件を満たすために、アプリケーションがすべてのシステムクロックを停止するか、周波数を削減する準備ができたときに、この条件に入るべきです。サスペンドモード中に USB アクティビティ (WKUP イベント) が発生すると、このビットは非同期でリセットされます (ソフトウェアによってリセットすることもできます)。

0 : 低電力モードではありません。

1 : 低電力モードに入ります。

ビット 1 **PDWN** : パワーダウ

このビットは、何らかの理由で USB ペリフェラルを完全に無効にする必要がある場合、USB 関連のすべてのアナログ部品を完全にオフにするために使用されます。このビットがセットされると、USB ペリフェラルはトランシーバから切断され、使用できなくなります。

0 : パワーダウを終了します。

1 : パワーダウモードに入ります。

ビット 0 **FRES** : 強制的 USB リセット

0 : USB リセットをクリアします。

1 : USB に RESET 信号が送られたときと同様に、USB ペリフェラルを強制的にリセットします。ソフトウェアがこのビットをクリアするまで、USB ペリフェラルは RESET 状態に保たれます。有効な場合は、「USB-RESET」割り込みが生成されます。

USB 割り込みステータスレジスタ (USB_ISTR)

アドレスオフセット : 0x44

リセット値 : 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR	PMA OVR	ERR	WKUP	SUSP	リセット	SOF	ESOF	L1REQ	Res.	Res.	DIR	EP_ID[3:0]			
r	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0			r	r	r	r	r

このレジスタはすべての割り込みソースのステータスを含むため、アプリケーションソフトウェアは割り込みリクエストの原因になったイベントを判断できます。

このレジスタの上位部分はシングルビットを含み、そのそれぞれが特定のイベントを表します。これらのビットは、関連するイベントが発生したときにハードウェアによってセットされます。USB_CNTR レジスタの対応するビットがセットされている場合、汎用割り込みリクエストが生成されます。各ビットを調べる割り込みルーチンは、必要なすべてのアクションを実行し、最後に処理済みビットをクリアします。そのいずれかがクリアされていない場合、割り込みはまだ保留中であるとみなされ、割り込みラインは再びハイに保たれます。複数のビットが同時にセットされた場合、1 つの割り込みだけが生成されます。

エンドポイントトランザクションの完了は、割り込み応答遅延を軽減するために、さまざまな方法で処理できます。CTR ビットは、エンドポイントがトランザクションを正常に完了するとすぐにハードウェアによってセットされ、USB_CNTR の対応するビットがセットされていた場合は汎用割り込みリクエストが生成されます。エンドポイント専用割り込み条件は、USB_CNTR レジスタの CTRM ビットとは無関係に有効化されます。どちらの割り込み条件も、ソフトウェアが対応する USB_EPnR レジスタの保留中のビットをクリアするまでアクティブなままです (CTR ビットは実際には読み出し専用ビットです)。エンドポイント関連の割り込みの場合、ソフトウェアは、トランザクション方向 (DIR) および EP_ID 読み出し専用ビットを使用して、最後の割り込みリクエストを発行して、対応する割り込みサービスルーチンを呼び出したエンドポイントを特定できます。

ユーザは、ソフトウェアが割り込みサービ斯拉ーチンで USB_ISTR ビットをチェックする順序を指定することによって、同時に保留中の USB_ISTR ビットの相対的優先順位を選ぶことができます。イベントに関連し、処理済みのビットだけがクリアされます。サービ斯拉ーチンの終了時、残りの条件を処理するために、別の割り込みがリクエストされます。

ビットの誤ったクリアを避けるために、ロード命令付きでクリアすることが推奨され、その場合、変更してはならないすべてのビットに 1 が書き込まれ、クリアすべきすべてのビットに 0 が書き込まれます（これらのビットはソフトウェアによってのみクリアできます）。読み出し-変更-書き込みのサイクルは避けるべきです。読み出し操作と書き込み操作の間に、別のビットがハードウェアによってセットされる可能性があり、マイクロプロセッサにイベントを処理する時間ができる前に、次の書き込みによってクリアされる可能性があるためです。

以下に、各ビットについて詳しく説明します。

ビット 15 CTR : 正しい転送

このビットは、エンドポイントがトランザクションを正常に完了したことを示すために、ハードウェアによってセットされます。DIR および EP_ID ビットを使用して、ソフトウェアは割り込みをリクエストしたエンドポイントを判断できます。このビットは読み出し専用です。

ビット 14 PMAOVR : パケットメモリ領域のオーバー/アンダーラン

このビットは、マイクロコントローラに USB メモリリクエストに対応する時間がなくなった場合にセットされます。USB ペリフェラルは、このイベントを次のように処理します。受信時、ACK ハンドシェイクパケットは送信されず、送信時、送信ストリームのビットスタッフエラーが強制的に発行されます。いずれの場合も、ホストはトランザクションを再試行します。通常操作では、PMAOVR 割り込みは発生しません。失敗したトランザクションはホストによって再試行されるので、アプリケーションソフトウェアには、この割り込み処理中にデバイス操作をスピードアップして、次のトランザクションの差異施行に備える機会があります。ただし、アイソクロナス転送では（アイソクロナストランザクションは再試行されません）、データの喪失につながるため、これは行われません。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 13 ERR : 誤差

このフラグは、下記にリストされているエラーのいずれかが発生したときにセットされます。
NANS : 応答なし (No Answer)。ホスト応答がタイムアウトになりました。
CRC : 巡回冗長検査 (Cyclic Redundancy Check) エラー。トークンまたはデータで受信された CRC の 1 つが正しくありませんでした。
BST : ビットスタッフエラー。ビットスタッフエラーが、PID、データ、または CRC で検出されました。
FVIO : フレーミングフォーマット違反。非標準のフレームが受信されました（正しい位置にない EOP、正しくないトークンシーケンスなど）。
USB ペリフェラルと PC ホストは、エラーの場合、完全にトランスパレントな方法で再送信を管理するため、USB ソフトウェアは、通常、エラーを無視できます。この割り込みは、ソフトウェア開発フェーズで、または USB バスでの送信品質の監視に役立ち、問題の可能性をユーザに知らせることができます（コネクタの緩み、ノイズが多い環境、USB ケーブルの導体の破損など）。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 12 WKUP : ウェイクアップ

このビットは、サスペンドモード中に、USB ペリフェラルをウェイクアップするアクティビティが検出されたときにハードウェアによって 1 にセットされます。このイベントは CTRL レジスタの LP_MODE ビットを非同期でクリアし、USB_WAKEUP ラインを有効にし、レジュームプロセスの開始に関して残りのデバイス（ウェイクアップユニットなど）に通知するために使用されます。このビットは読み出し/書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 11 SUSP : サスペンドモードリクエスト

このビットは、3 ms にわたってトラフィックが受信されず、USB バスからのサスペンドモードリクエストを示したときに、ハードウェアによってセットされます。サスペンド条件チェックは、USB リセットの直後に有効化され、サスペンドモードがアクティブなとき (FSUSP=1)、レジュームシーケンスの終了まで、ハードウェアによって無効にされます。このビットは読み出し／書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 10 RESET : USB リセットリクエスト

USB ペリフェラルが入力でアクティブな USB RESET 信号を検出したときにセットされます。USB ペリフェラルは、RESET に対応して、内部プロトコル状態マシンをリセットし、USB_CNTR レジスタの RESETM 有効ビットがセットされている場合、割り込みが生成されます。RESET ビットがクリアされるまで、受信と送信は無効です。すべての設定レジスタがリセットされるわけではありません。マイクロコントローラはこれらのレジスタを明示的にクリアする必要があります (これは、RESET 割り込みの安全な配信を確保し、RESET の直前のトランザクションを完了できるようにするためです)。機能アドレスおよびエンドポイントレジスタは、USB リセットイベントによってリセットされます。このビットは読み出し／書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 9 SOF : フレーム開始

このビットは、新しい USB フレームの開始を知らせ、SOF パケットが USB バス経由で着信したときにセットされます。割り込みサービスルーチンによって SOF イベントを監視して、USB ホストへの 1 ms の同期イベントを実行し、SOF パケット受信時に更新される USB_FNR レジスタを安全に読み出すことができます (これはアイソクロナスアプリケーションに便利です)。このビットは読み出し／書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 8 ESOF : 予期されたフレーム開始

このビットは、予期された SOF パケットが受信されなかったときにハードウェアによってセットされます。ホストは 1 ms ごとに SOF パケットを送信しますが、ハブが正しく受信しなかった場合、サスペンドタイマがこの割り込みを発行します。3 回連続して ESOF 割り込みが生成され (すなわち、3 つの SOF パケットが失われ)、その間にトラフィックが発生しなかった場合、SUSP 割り込みが生成されます。このビットは、サスペンドタイマがまだロックされていないときに SOF パケットの喪失が発生したときにセットされます。このビットは読み出し／書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 7 L1REQ : LPM L1 状態リクエスト

このビットは、L1 状態を入力する LPM コマンドが正常に受信され、確認応答された場合に、ハードウェアによってセットされます。このビットは読み出し／書き込みですが、0 のみを書き込むことができ、1 を書き込んでも効果はありません。

ビット 6:5 予約済み。

ビット 4 **DIR** : トランザクションの方向

このビットは、割り込みリクエストを生成した成功したトランザクションの方向に従って、ハードウェアによって書き込まれます。

DIR ビット =0 の場合、割り込みをかけたエンドポイントに関連する USB_EPnR レジスタの CTR_TX ビットがセットされます。割り込みの原因となったトランザクションは、IN タイプです (USB ペリフェラルからホスト PC へのデータ送信)。

DIR ビット =1 の場合、割り込みをかけたエンドポイントに関連する USB_EPnR レジスタの CTR_RX ビットまたは CTR_TX と CTR_RX の両方がセットされます。割り込みの原因となったトランザクションは OUT タイプであるか (ホスト PC からのデータを USB ペリフェラルが受信)、または 2 つの保留中のトランザクションが処理を待っています。

これは割り込みを保留中にした方向を表すので、アプリケーションソフトウェアはこの情報を使用して、トリガしたトランザクションに関連する USB_EPnR ビットにアクセスできます。このビットは読み出し専用です。

ビット 3:0 **EP_ID[3:0]** : エンドポイント識別子

これらのビットは、割り込みリクエストを生成したエンドポイント番号に従って、ハードウェアによって書き込まれます。複数のエンドポイントトランザクションが保留中の場合、ハードウェアは、次のように定義される最も高い優先順位を持つエンドポイントに関連するエンドポイント識別子を書き込みます。優先順位の順に、2 つのエンドポイントセットが定義されます。アイソクロナスおよびダブルバッファバルクエンドポイントが最初に考慮され、次に、その他のエンドポイントが調べられます。同じセットの複数のエンドポイントが割り込みをリクエストしている場合、最下位リクエストエンドポイントレジスタに従って、USB_ISTR レジスタの EP_ID ビットが割り当てられ、EP0R が最も高い優先順位を持ち、その後に EP1R が続き、以下同様です。アプリケーションソフトウェアは、この優先順位方式に従って、レジスタを各エンドポイントに割り当てて、同時発生したエンドポイントリクエストを適切に順序づけることができます。これらのビットは読み出し専用です。

USB フレーム番号レジスタ（USB_FNR）

アドレスオフセット : 0x48

リセット値 : 0x0XXX (X は未定義)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDP	RXDM	LCK	LSOF[1:0]		FN[10:0]										
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15 **RXDP** : 受信データ + ラインステータス

このビットを使用して、受信データプラスアップストリームポートデータラインのステータスを観測できます。サスペンド終了ルーチン時に使用して、ウェイクアップイベントの決定を容易にできます。

ビット 14 **RXDM** : 受信データ - ラインステータス

このビットを使用して、受信データマイナスアップストリームポートデータラインのステータスを観測できます。サスペンド終了ルーチン時に使用して、ウェイクアップイベントの決定を容易にできます。

ビット 13 **LCK** : ロック

このビットは、USB リセット条件の終了後、または USB レジュームシーケンスの終了後に、少なくとも 2 つ連続した SOF パケットが受信されたときに、ハードウェアによってセットされます。ロックされると、USB リセットまたは USB サスペンドイベントが発生するまで、フレームタイムはこの状態に保たれます。

ビット 12:11 **LSOF[1:0]** : SOF の喪失

これらのビットは、ESOF 割り込みが生成されたときに、ハードウェアによって書き込まれ、失われた連続 SOF パケットの数をカウントします。SOF パケットの受信時、これらのビットはクリアされます。

ビット 10:0 **FN[10:0]** : フレーム番号

このビットフィールドは、最後に受信された SOF パケットに含まれる 11 ビットのフレーム番号を含みます。フレーム番号は、ホストによって送信されたフレームごとにインクリメントされ、アイソクロナス転送の場合に便利です。このビットフィールドは、SOF 割り込みの生成時に更新されます。

USB デバイスアドレス (USB_DADDR)

アドレスオフセット : 0x4C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD6	ADD5	ADD4	ADD3	ADD2	ADD1	ADD0
								rw	rw	rw	rw	rw	rw	rw	rw

ビット 15:8 予約済み

ビット 7 **EF** : 機能有効

このビットは、USB デバイスを有効にするために、ソフトウェアによってセットされます。このデバイスのアドレスは、次の ADD[6:0] ビットに含まれます。このビットが 0 の場合、USB_EPnR レジスタの設定にかかわらず、トランザクションは処理されません。

ビット 6:0 **ADD[6:0]** : デバイスアドレス

これらのビットは、エニュメレーションプロセス時にホスト PC によって割り当てられた USB 機能を含みます。必要なエンドポイントへのトランザクションを処理するためには、関連する USB_EPnR レジスタのこのフィールドとエンドポイントアドレス (EA) フィールドの両方とも、USB トークンに含まれる情報に一致する必要があります。

バッファテーブルアドレス (USB_BTABLE)

アドレスオフセット : 0x50

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BTABLE[15:3]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

ビット 15:3 **BTABLE[15:3]** : バッファテーブル

これらのビットは、専用パケットメモリ内のバッファ割り当てテーブルの開始アドレスを含みます。このテーブルは、各エンドポイントバッファ位置とサイズを記述し、8 バイトの境界に揃えられなければなりません（3 つの最下位ビットが常に 0）。このデバイスにアドレス指定されたトランザクションの開始ごとに、USB ペリフェラルはアドレス指定されたエンドポイントに関連このテーブルの要素を読み出して、バッファ開始位置とバッファサイズを取得します（[パケットバッファの構造と用途 \(873 ページ\)](#)を参照）。

ビット 2:0 予約済み、ハードウェアによって 0 に固定されています。

LPM 制御およびステータスレジスタ（USB_LPMCSR）

アドレスオフセット : 0x54

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]				REM WAKE	Res.	LPM ACK	LPM EN
								r				r		rw	rw

ビット 15:8 予約済み。

ビット 7:4 **BESL[3:0]** : BESL 値

これらのビットは、最後に ACK された LPM トークンとともに受信した BESL 値を含みます。

ビット 3 **REM WAKE** : bRemoteWake 値

このビットは、最後に ACK された LPM トークンとともに受信した bRemoteWake 値を含みます。

ビット 2 予約済み

ビット 1 **LPMACK** : LPM トークン確認応答イネーブル

0 : 有効な LPM トークンは NYET です。

1 : 有効な LPM トークンは ACK です。

NYET/ACK は、LPM トランザクションが成功した場合にのみ返されます。

EXT トークンと LPM トークンの両方にエラーがない場合（それ以外の場合は ERROR）

有効な bLinkState = 0001B（L1）を受信した場合（それ以外の場合は STALL）

ビット 0 **LPMEN** : LPM サポートイネーブル

このビットは、USB デバイス内の LPM サポートを有効にするために、ソフトウェアによってセットされます。このビットが 0 の場合は、LPM トランザクションは処理されません。

バッテリー充電検出回路(USB_BCDR)

アドレスオフセット : 0x58

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPPU	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2 DET	SDET	PDET	DC DET	SDEN	PDEN	DCD EN	BCD EN
rw								r	r	r	r	rw	rw	rw	rw

ビット 15 DPPU : DP プルアップ制御

このビットは、DP ライン上の組み込みプルアップを有効にするために、ソフトウェアによってセットされます。0 にクリアすると、ユーザのソフトウェアが必要とするときに、ホストとの接続が切断されたことを通知します。

ビット 14:8 予約済み。**ビット 7 PS2DET** : DM プルアップ検出ステータス

このビットは、PD 中のみアクティブになって DM 電圧レベルと V_{LGC} 閾値の比較の結果を示します。通常の状況では、DM レベルは閾値を下回る必要があります。閾値を上回る場合は、DM が外部的にハイレベルにプルアップされることを意味します。これは、PS2 ポート (DP および DM ラインの両方をプルアップする) への接続、または BCD 仕様に準拠しない一部の独自の充電器への接続によって発生します。

0 : 通常のポートが検出されました (SDP、ACA、CDP、または DCP に接続されたポート)。

1 : PS2 ポートまたは独自の充電器が検出されました。

ビット 6 SDET : 2 次検出 (SD) ステータス

このビットは、SD の結果を示します。

0 : CDP が検出されました。

1 : DCP が検出されました。

ビット 5 PDET : 1 次検出 (PD) ステータス

このビットは、PD の結果を示します。

0 : BCD サポートは検出されませんでした (SDP または独自デバイスに接続)。

1 : BCD のサポートが検出されました (ACA、CDP、または DCP に接続)。

ビット 4 DCDDET : データ接触検出 (DCD) ステータス

このビットは、DCD の結果を示します。

0 : データラインの接触は検出されませんでした。

1 : データラインの接触が検出されました。

ビット 3 SDEN : 2 次検出 (SD) モードイネーブル

このビットは、BCD を SD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 2 PDEN : 1 次検出 (PD) モードイネーブル

このビットは、BCD を PD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 1 DCDEN : データ接触検出 (DCD) モードイネーブル

このビットは、BCD を DCD モードにするために、ソフトウェアによってセットされます。正しく動作するには、検出モード (DCD、PD、SD、または OFF) を 1 つだけ選択する必要があります。

ビット 0 BCDEN : バッテリ充電検出 (BCD) イネーブル

このビットは、USB デバイス内の BCD サポートを有効にするために、ソフトウェアによってセットされます。有効にすると、USB PHY は BCD によって完全に制御され、通常の通信では使用できなくなります。BCD の検出完了後は、通常の USB 動作を可能にするために、このビットを 0 にクリアして BCD を OFF モードにします。

エンドポイント固有レジスタ

これらのレジスタの数は、USB ペリフェラルが処理する設計になっているエンドポイント数に応じて変化します。USB ペリフェラルは、最大 8 つの双方向エンドポイントをサポートします。各 USB デバイスは、制御エンドポイントをサポートしなければならず、そのアドレス (EA ビット) は 0 にセットされなければなりません。複数のエンドポイントが有効であり、同じエンドポイント番号値を持つ場合、USB ペリフェラルの動作は未定義です。各エンドポイントについて、USB_EPnR レジスタはエンドポイント固有の情報を格納できます。

USB エンドポイント n レジスタ (USB_EPnR)、n=[0..7]

アドレスオフセット : 0x00 から 0x1C

リセット値 : 0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR_RX	DTOG_RX	STAT_RX[1:0]			SETUP	EP_TYPE[1:0]		EP_KIND	CTR_TX	DTOG_TX	STAT_TX[1:0]		EA[3:0]		
rc_w0	t	t	t	r	rw	rw	rw	rc_w0	t	t	t	rw	rw	rw	rw

USB リセットが USB バスから受信されたとき、または CTRL レジスタのビット FRES を通じて強制されたときにもリセットされますが、USB リセットイベントの直前の正しいパケット通知の喪失を回避するために、CTR_RX および CTR_TX ビットは変更されません。各エンドポイントに USB_EPnR レジスタがあり、n はエンドポイント識別子です。

これらのレジスタでは読み出し-変更-書き込みのサイクルは避けるべきです。読み出し操作と書き込み操作の間に、いくつかのビットがハードウェアによってセットされる可能性があり、CPU に変更を検出する時間ができる前に、次の書き込みによって変更される可能性があるためです。この目的のために、この問題の影響を受けるすべてのビットが「不変」の値を持ち、変更が不要なときに使用される必要があります。これらのレジスタはロード命令付きで変更することが推奨され、この場合、ハードウェアによって変更できるすべてのビットには「不変」値が書き込まれます。

ビット 15 CTR_RX : 受信の場合の正しい転送

このビットは、このエンドポイントで OUT/SETUP トランザクションが正常に完了したときに、ハードウェアによってセットされます。ソフトウェアのみがこのビットをクリアできます。USB_CNTR レジスタの CTRM ビットがセットされた場合、エンドポイント関連の割り込み条件（常に有効化されます）とともに汎用割り込み条件が生成されます。発生したトランザクションのタイプ（OUT または SETUP）は、下記の SETUP ビットから決めることができます。

NAK または STALL ハンドシェイクで終了したトランザクションでは、プロトコルエラーやデータトグル不一致の場合のように、実際にはデータは転送されていないため、このビットはセットされません。このビットは読み出し／書き込みですが、0 のみを書き込むことができ、1 を書き込んで効果はありません。

ビット 14 DTOG_RX : 受信転送の場合のデータトグル

エンドポイントがアイソクロナスでない場合、このビットは次の受信データパケットのデータトグルビット（0=DATA0、1=DATA1）の予期される値を含みます。データ PID 値が一致するデータパケット受信後、ACK ハンドシェイクが USB ホストに送信されたときに、ハードウェアはこのビットをトグルします。エンドポイントが制御エンドポイントとして定義された場合、ハードウェアはこのエンドポイントにアドレス指定された SETUP PID の受信時に、このビットをクリアします。

エンドポイントがダブルバッファリングを使用している場合、このビットはパケットバッファスワッピングをサポートするためにも使用されます（[セクション 30.5.3 : ダブルバッファエンドポイント](#)を参照）。

エンドポイントがアイソクロナスの場合、この種類のエンドポイントではデータトグルは使用されず、DATA0 パケットのみが送信されるため、このビットはパケットバッファスワッピングをサポートするためにのみ使用されます（[セクション 30.5.4 : アイソクロナス転送](#)を参照）。アイソクロナス転送ではハンドシェイクは使用されないため、ハードウェアはデータパケット受信の終了直後に、このビットをトグルします。

このビットは、値を初期化するため（エンドポイントが制御エンドポイントでないときは必須）、または特定のデータトグル／パケットバッファ使用のために、ソフトウェアによってトグルすることもできます。アプリケーションソフトウェアが 0 を書き込むと、DTOG_RX の値は変更されませんが、1 を書き込むと、ビット値がトグルされます。このビットは読み出し／書き込みですが、1 を書き込むことによってのみ、トグルが可能です。

ビット 13:12 **STAT_RX[1:0]** : 受信転送の場合のステータスビット

これらのビットは、[表 123 : 受信ステータスエンコード \(895 ページ\)](#) にリストされているエンドポイントステータスに関する情報を含みます。これらのビットは、ソフトウェアによってトグルして、値を初期化することができます。アプリケーションソフトウェアが 0 を書き込むと、値は変更されませんが、1 を書き込むと、ビット値がトグルされます。ハードウェアは、このエンドポイントにアドレス指定された OUT または SETUP (制御のみ) トランザクションに対応する正しい転送が発生したとき (CTR_RX=1)、STAT_RX ビットを NAK にセットするので、ソフトウェアは新しいトランザクションに確認応答する前に受信データを処理する時間があります。

ダブルバッファバルクエンドポイントは、バッファ可用性条件に基づいてステータスを制御する特殊なトランザクションフロー制御を実装します ([セクション 30.5.3 : ダブルバッファエンドポイント](#) を参照)。

エンドポイントがアイソクロナスとして定義された場合、そのステータスは「VALID」または「DISABLED」のみなので、ハードウェアはトランザクションの成功後にエンドポイントのステータスを変更できません。ソフトウェアがアイソクロナスエンドポイントの STAT_RX ビットを STALL または NAK にセットした場合、USB ペリフェラルの動作は未定義です。これらのビットは読み出し/書き込みですが、1 を書き込むことによるのみ、トグルが可能です。

ビット 11 **SETUP** : セットアップトランザクション完了

このビットは読み出し専用であり、最後に完了したトランザクションが SETUP のときにハードウェアによってセットされます。このビットは制御エンドポイントについてのみ値を変更します。成功した受信トランザクションの場合 (CTR_RX イベント)、これを調べて、発生したトランザクションのタイプを判断する必要があります。割り込みサービスルーチンを次の着信トークンによる SETUP ビットの変更から保護するために、このビットは CTR_RX ビットが 1 の間は停止され、CTR_RX が 0 のときに状態が変更されます。このビットは読み出し専用です。

ビット 10:9 **EP_TYPE[1:0]** : エンドポイントタイプ

これらのビットは、[表 124 : エンドポイントタイプエンコード \(895 ページ\)](#) に記載されているように、このエンドポイントの動作を設定します。エンドポイント 0 は常に制御エンドポイントでなければならず、各 USB 機能にはアドレス 0 を持つ制御エンドポイントが少なくとも 1 つ必要ですが、必要な場合は、他の制御エンドポイントを使用することもできます。制御エンドポイントだけが SETUP トランザクションを処理し、他の種類のエンドポイントによって無視されます。SETUP トランザクションは NAK または STALL で応答することはできません。制御エンドポイントが NAK として定義された場合、SETUP トランザクションが受信されたときには、受信方向では USB ペリフェラルは応答せず、受信エラーをシミュレートします。制御エンドポイントが受信方向で STALL として定義された場合、SETUP パケットは受け入れられ、データを送信し、CTR 割り込みを発行します。エンドポイントが制御エンドポイントの場合でも、OUT トランザクションの受信は通常通りに処理されます。

バルクおよび割り込みエンドポイントの動作はほぼ同様ですが、EP_KIND 設定ビットを使用して使用可能な特殊な機能のみが異なります。

アイソクロナスエンドポイントの使用については、[セクション 30.5.4 : アイソクロナス転送](#) で説明します。

ビット 8 **EP_KIND** : エンドポイントの種類

このビットの意味は、EP_TYPE ビットによって設定されたエンドポイントのタイプに依存します。[表 125](#) はさまざまな意味を要約します。

DBL_BUF : このビットは、このバルクエンドポイントのダブルバッファリング機能を有効にするために、ソフトウェアによってセットされます。ダブルバッファバルクエンドポイントの使用については、[セクション 30.5.3 : ダブルバッファエンドポイント](#) で説明します。

STATUS_OUT : このビットは、ステータスアウトトランザクションが予期されることを示すためにソフトウェアによってセットされます。この場合、1 個以上のデータバイトを含んでいるすべての OUT トランザクションは、ACK ではなく STALL で応答されます。このビットを使用して、制御転送時のプロトコルエラーに対するアプリケーションの堅牢性を向上させることができ、その使用は制御エンドポイントのみを対象としています。STATUS_OUT がリセットされると、OUT トランザクションは必要に応じて任意の数のバイトを持つことができます。

ビット 7 CTR_TX : 送信の場合の正しい転送

このビットは、このエンドポイントで IN トランザクションが正常に完了したときに、ハードウェアによってセットされます。ソフトウェアのみがこのビットをクリアできます。USB_CNTR レジスタの CTRM ビットがセットされた場合、エンドポイント関連の割り込み条件（常に有効化されます）とともに汎用割り込み条件が生成されます。

NAK または STALL ハンドシェイクで終了したトランザクションでは、プロトコルエラーやデータトグル不一致の場合のように、実際にはデータは転送されていないため、このビットはセットされません。このビットは読み出し／書き込みですが、0 のみを書き込むことができます。

ビット 6 DTOG_TX : 送信転送の場合のデータトグル

エンドポイントがアイソクロナスでない場合、このビットは次の送信データパケットのデータトグルビット (0=DATA0、1=DATA1) の必要な値を含みます。データパケットの送信後、USB ホストから ACK ハンドシェイクが受信されると、ハードウェアはこのビットをトグルします。エンドポイントが制御エンドポイントとして定義された場合、このエンドポイントにアドレス指定された SETUP PID の受信時に、ハードウェアはこのビットを 1 にセットします。

エンドポイントがダブルバッファ機能を使用している場合、このビットはパケットバッファスワッピングをサポートするためにも使用されます（[セクション 30.5.3 : ダブルバッファエンドポイント](#)を参照）。

エンドポイントがアイソクロナスの場合、この種類のエンドポイントではデータトグルは使用されず、DATA0 パケットのみが送信されるため、このビットはパケットバッファスワッピングをサポートするためにのみ使用されます（[セクション 30.5.4 : アイソクロナス転送](#)を参照）。アイソクロナス転送ではハンドシェイクは使用されないため、ハードウェアはデータパケット送信の終了直後に、このビットをトグルします。

このビットは、値を初期化するため（エンドポイントが制御エンドポイントでないときは必須）、または特定のデータトグル／パケットバッファ使用のために、ソフトウェアによってトグルすることもできます。アプリケーションソフトウェアが 0 を書き込むと、DTOG_TX の値は変更されませんが、1 を書き込むと、ビット値がトグルされます。このビットは読み出し／書き込みですが、1 を書き込むことによってのみ、トグルが可能です。

ビット 5:4 STAT_TX[1:0] : 送信転送の場合のステータスビット

これらのビットは、[表 126](#) にリストされているエンドポイントのステータスに関する情報を含みます。これらのビットは、ソフトウェアによってトグルして、値を初期化できます。アプリケーションソフトウェアが 0 を書き込むと、値は変更されませんが、1 を書き込むと、ビット値がトグルされます。ハードウェアは、このエンドポイントにアドレス指定された IN または SETUP (制御のみ) トランザクションに対応する正しい転送が発生したとき (CTR_TX=1)、STAT_TX ビットを NAK にセットします。次に、ソフトウェアが次の送信データセットを準備するのを待ちます。

ダブルバッファバルクエンドポイントは、バッファ可用性条件に基づいてステータスを制御する特殊なトランザクションフロー制御を実装します（[セクション 30.5.3 : ダブルバッファエンドポイント](#)を参照）。

エンドポイントがアイソクロナスとして定義された場合、そのステータスは「VALID」または「DISABLED」のみです。したがって、ハードウェアは、トランザクションの成功後にエンドポイントのステータスを変更することはできません。ソフトウェアがアイソクロナスエンドポイントの STAT_TX ビットを STALL または NAK にセットした場合、USB ペリフェラルの動作は未定義です。これらのビットは読み出し／書き込みですが、1 を書き込むことによってのみ、トグルが可能です。

ビット 3:0 EA[3:0] : エンドポイントアドレス

ソフトウェアは、このエンドポイント宛でのトランザクションを識別するために使用される 4 ビットのアドレスをこのフィールドに書き込む必要があります。値は、対応するエンドポイントを有効にする前に書き込まれる必要があります。

表 123. 受信ステータスエンコード

STAT_RX[1:0]	意味
00	DISABLED : このエンドポイントにアドレス指定されたすべての受信リクエストは無視されます。
01	STALL : エンドポイントは停止していて、すべての受信リクエストは STALL ハンドシェイクで終了します。
10	NAK : エンドポイントは NAK され、すべての受信リクエストは NAK ハンドシェイクで終了します。
11	VALID : このエンドポイントは受信に有効です。

表 124. エンドポイントタイプエンコード

EP_TYPE[1:0]	意味
00	BULK
01	CONTROL
10	ISO
11	インタラプト

表 125. エンドポイントの種類の意味

EP_TYPE[1:0]		EP_KIND の意味
00	BULK	DBL_BUF
01	CONTROL	STATUS_OUT
10	ISO	未使用
11	インタラプト	未使用

表 126. 送信ステータスエンコード

STAT_TX[1:0]	意味
00	DISABLED : このエンドポイントにアドレス指定されたすべての送信リクエストは無視されます。
01	STALL : エンドポイントは停止していて、すべての送信リクエストは STALL ハンドシェイクで終了します。
10	NAK : エンドポイントは NAK され、すべての送信リクエストは NAK ハンドシェイクで終了します。
11	VALID : このエンドポイントは送信に有効です。

30.6.2 バッファディスクリプタテーブル

バッファディスクリプタテーブルはパケットバッファメモリ内にありますが、そのエントリは、USB マクロセルとデバイスの間でデータを交換するためのパケットバッファの位置とサイズを設定する追加レジスタとみなすことができます。

最初のパケットメモリ位置は、0x4000 6000 にあります。USB_EPnR レジスタに関連するバッファディスクリプタテーブルのエントリについて、以下に説明します。パケットメモリへのアクセスは、バイト (8 ビット) またはハーフワード (16 ビット) アクセスによってのみ行う必要があります。ワード (32 ビット) アクセスは許されません。

パケットバッファとバッファディスクリプタテーブルの使用に関する詳細については、[パケットバッファの構造と用途 \(873 ページ\)](#) を参照してください。

送信バッファアドレス n (USB_ADDRn_TX)

アドレスオフセット : [USB_BTABLE] + n*8

注 : *IN 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_ADDRn_TX_0 と呼ばれます。*
OUT 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_ADDRn_RX_0 と呼ばれます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn_TX[15:1]															-
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	-

ビット 15:1 **ADDRn_TX[15:1]** : 送信バッファアドレス
これらのビットは、USB_EPnR レジスタに関連するエンドポイントによって、次の IN トークンで送信されるデータを含んでいるパケットバッファの開始アドレスを指します。

ビット 0 パケットメモリはハーフワード長であり、すべてのパケットバッファがハーフワードで揃えられている必要があるため、常に 0 が書き込まれる必要があります。

送信バイトカウント n (USB_COUNTn_TX)

アドレスオフセット : [USB_BTABLE] + n*8 + 2

注 : *IN 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_COUNTn_TX_0 と呼ばれます。*
OUT 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_COUNTn_RX_0 と呼ばれます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	COUNTn_TX[9:0]									
						r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

ビット 15:10 USB 仕様によりパケットサイズは 1023 バイトに制限されるので、これらのビットは使用されません。これらの値は、USB ペリフェラルによって考慮されません。

ビット 9:0 **COUNTn_TX[9:0]** : 送信バイトカウント
これらのビットは、USB_EPhR レジスタに関連するエンドポイントによって、次の IN トークンで送信されるバイト数を含みます。

受信バッファアドレス n (USB_ADDRn_RX)

アドレスオフセット : [USB_BTABLE] + n*8 + 4

注 : **OUT 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_ADDRn_RX_1 と呼ばれます。**
IN 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_ADDRn_TX_1 と呼ばれます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRn_RX[15:1]															-
rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	rW	-

ビット 15:1 **ADDRn_RX[15:1]** : 送信バッファアドレス
これらのビットは、USB_EPhR レジスタに関連するエンドポイントによって、次の OUT/SETUP トークンで受信されるデータを含んでいるパケットバッファの開始アドレスを指します。
ビット 0 パケットメモリはハーフワード長であり、すべてのパケットバッファがハーフワードで揃えられている必要があるため、常に 0 が書き込まれる必要があります。

受信バイトカウント n (USB_COUNTn_RX)

アドレスオフセット : [USB_BTABLE] + n*8 + 6

注 : **OUT 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_COUNTn_RX_1 と呼ばれます。**
IN 方向のダブルバッファまたはアイソクロナスエンドポイントの場合、このアドレス位置は USB_COUNTn_TX_1 と呼ばれます。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BLSIZE	NUM_BLOCK[4:0]					COUNTn_RX[9:0]									
rW	rW	rW	rW	rW	rW	r	r	r	r	r	r	r	r	r	r

このテーブル位置は、両方ともパケット受信時に必要とされる 2 つの値を格納するために使用されます。最上位ビットは割り当てられるバッファサイズの定義を含み、バッファオーバーフロー検出を可能にし、この位置の最下位部分には、受信終了時に USB ペリフェラルによって実際の受信バイト数を書き込まれます。使用可能なビット数が制限されているため、バッファサイズは割り当てられたメモリブロック数を使用して表され、細かい粒度／小さいバッファと粗い粒度／大きなバッファの間のトレードオフにより、ブロックサイズを選択できます。割り当てられたバッファのサイズは、エンドポイントディスクリプタの一部であり、通常、エnumレーションプロセス時に maxPacketSize パラメータ値にしたって定義されます (「ユニバーサルシリアルバス仕様」を参照)。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

ビット 15 **BL_SIZE** : ブロックサイズ

このビットは、割り当てられたバッファ領域を定義するために使用されるメモリブロックのサイズを選択します。

- BL_SIZE=0 の場合、メモリブロックは 2 バイト長であり、これは、ハーフワード長のメモリで可能な最小ブロックです。このブロックサイズでは、割り当てられるバッファサイズは 2 ～ 62 バイトです。
- BL_SIZE=1 の場合、メモリブロックは 32 バイト以上であり、USB 仕様によって定義された最大パケット長に達することができます。このブロックサイズでは、割り当てられるバッファサイズは、理論上、32 ～ 1024 バイトであり、USB 標準仕様で可能な最長のパケットサイズです。ただし、適用可能なサイズは、使用可能なバッファメモリによって制限されます。

ビット 14:10 **NUM_BLOCK[4:0]** : ブロック数

これらのビットは、このパケットバッファに割り当てられるメモリブロック数を定義します。実際の割り当てメモリの量は、表 127 に示されているように、BL_SIZE の値に依存します。

ビット 9:0 **COUNTn_RX[9:0]** : 受信バイトカウント

これらのビットは、USB_EPnR レジスタに関連するエンドポイントによって、最後の OUT/SETUP トランザクションで受信されたバイト数を含みます。

表 127. 割り当てられるバッファメモリの定義

NUM_BLOCK[4:0] の値	BL_SIZE=0 のとき割り当てられるメモリ	BL_SIZE=1 のとき割り当てられるメモリ
0 ('00000)	設定禁止	32 バイト
1 ('00001)	2 バイト	64 バイト
2 ('00010)	4 バイト	96 バイト
3 ('00011)	6 バイト	128 バイト
...	...	...
14 ('01110)	28 バイト	480 バイト
15 ('01111)	30 バイト	512 バイト
16 ('10000)	32 バイト	544 バイト
...	...	...
29 ('11101)	58 バイト	960 バイト
30 ('11110)	60 バイト	992 バイト
31 ('11111)	62 バイト	N/A

30.6.3 USB レジスタマップ

次の表に、USB レジスタマップとリセット値を示します。

表 128. USB レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	USB_EP0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x04	USB_EP1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x08	USB_EP2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x0C	USB_EP3R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	USB_EP4R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x14	USB_EP5R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x18	USB_EP6R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	USB_EP7R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR_RX	DTOG_RX	STAT_RX [1:0]	SETUP	EP_TYPE [1:0]	EP_KIND	CTR_TX	DTOG_TX	STAT_TX [1:0]	EA[3:0]										
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x20- 0x3F	予約済み																																				
0x40	USB_CNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTRM	PMAOVRM	ERRM	WKUPM	SUSPM	RESETM	SOFM	ESOFM	L1REQM	Res.	L1RESUME	RESUME	FSUSP	LPMODE	PDWN	FRES				
	リセット値																	0	0	0	0	0	0	0	0			0	0	0	0	1	1				
0x44	USB_ISTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTR	PMAOVR	ERR	WKUP	SUSP	リセット	SOF	ESOF	L1REQ	Res.	Res.	DIR	EP_ID[3:0]							
	リセット値																	0	0	0	0	0	0	0	0			0	0	0	0	0	0				
0x48	USB_FNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDP	RXDM	LCK	LSOF [1:0]	FN[10:0]															
	リセット値																	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x				
0x4C	USB_DADDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EF	ADD[6:0]											
	リセット値																									0	0	0	0	0	0	0	0	0			
0x50	USB_BTABLER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BTABLER[15:3]																			
	リセット値																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

表 128. USB レジスタマップとリセット値 (続き)

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x54	USB_LPMCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BESL[3:0]				REMWAKE		Res.	LPMACK	LPMEN
	リセット値																									0	0	0	0	0		0	0
0x58	USB_BCDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DPPU	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2DET	SDET	PDET	DCDET	SDEN	PDEN	DCDEN	BCDEN
	リセット値																	0								0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

31 HDMI-CEC コントローラ (HDMI-CEC)

このセクションは、STM32F05x、STM32F04x、STM32F07x、および STM32F09x デバイスにのみ適用されます。

31.1 概要

Consumer Electronics Control (CEC) は HDMI (高解像度マルチメディアインタフェース) 規格の一部です (付録 1)。

ユーザ環境ですべての多様な視聴覚製品に高品位な制御機能を提供するプロトコルで構成されています。ロースピードでの最小限の処理とメモリのオーバーヘッドで、動作させるよう策定されています。

HDMI-CEC コントローラでは、このプロトコルに対するハードウェアサポートを提供しています。

31.2 HDMI-CEC コントローラの主な機能

- HDMI-CEC v1.4 仕様に準拠
- 2つのクロックソースオプションを用いた 32 kHz CEC カーネル
 - 固定プリスケラを用いた HSI RC オシレータ (HSI/244)
 - LSE オシレータ
- STOP モードの超低電力アプリケーションで動作
- 送信開始前に設定可能な信号フリータイム
 - CEC 状態と送信履歴に従って、ハードウェアによって自動的に実行される
 - ソフトウェアによって固定化 (7つのタイミングオプション)
- 設定可能なペリフェラルアドレス (OAR)
- リスسنモードをサポート
 - CEC ラインに干渉することなく、OAR とは異なる転送先アドレスに送信された CEC メッセージの受信を有効化
- 設定可能な Rx 許容誤差マージン
 - 標準の許容誤差
 - 拡張された許容誤差
- 受信 – エラー検出
 - ビット立ち上がりエラー (BRE)、オプションによる受信の停止 (BRESTOP)
 - ショートビット周期エラー (SBPE)
 - ロングビット周期エラー (LBPE)
- 設定可能なエラービット生成
 - BRE 検出 (BREGEN)
 - LBPE 検出 (LBPEGEN)
 - 常に SBPE 検出で生成される
- 送信エラー検出 (TXERR)
- アービトレーション喪失検出 (ARBLST)
 - 自動送信を再試行
- 送信アンダーラン検出 (TXUDR)
- 受信オーバーラン検出 (RXOVR)

31.3 HDMI-CEC の機能詳細

31.3.1 HDMI-CEC ピン

CEC バスは、デバイスからデータを送受信するために使用される 1 本の双方向ラインで構成されています。このバスは、27 kΩ プルアップレジスタを介して、+3.3 V 電源電圧に接続されています。ワイヤード AND 接続を使用するには、デバイスの出力段はオープンドレインまたはオープンコレクタである必要があります。

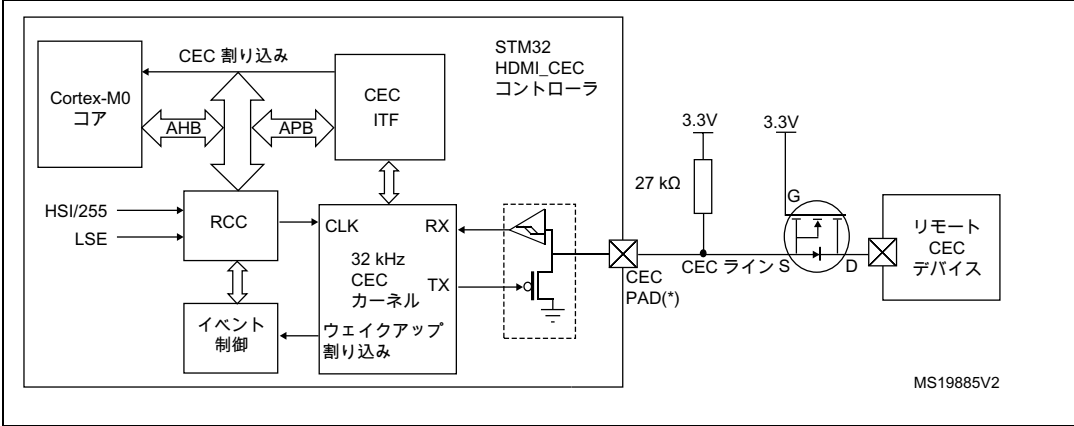
HDMI-CEC コントローラでは、CEC 双方向ラインをオルタネート機能オープンドレインとして設定されたものとみなし、標準 GPIO のオルタネート機能として管理します。27 kΩ プルアップを外部的に STM32 に追加する必要があります。

アプリケーションの電源をオフにした際に CEC バスと干渉しないよう、このような条件ではバスから CEC ピンを離す必要があります。これは、[図 324](#) に示すような MOS トランジスタを使用して実行できます。

表 129. HDMI ピン

名前	信号タイプ	説明
CEC	双方向	2 つの状態 : 1 = ハイインピーダンス 0 = ローインピーダンス 27 kΩ を外部的に追加する必要があります。

図 324. ブロック図



- 出力オープンドレインのオルタネート機能として設定された GPIO。
- 出力オープンドレインのオルタネート機能として設定される場合、シュミットトリガは有効なままとなります。

31.3.2 メッセージの説明

CEC ライン上のすべてのトランザクションは、イニシエータと 1 つまたは複数のフォローで構成されています。イニシエータは、メッセージの構造とデータを送信します。フォローは、すべてのデータを受信し、すべての確認応答ビットを設定します。

メッセージは単一フレームで送信されます。このフレームは、開始ビットとそれに続くヘッダブロックによって構成され、これにオプションとして OP コードとオペランドブロックの変数を含めることもできます。

これらのすべてのブロックは 8 ビットペイロードによって作成され（最上位ビットが最初に送信される）、続けてメッセージの終了 (EOM) ビット、および確認応答 (ACK) ビットによって作成されます。

EOM ビットはメッセージの最後のブロックにセットされ、その他のブロックではリセットされます。EOM が示された後でメッセージに追加のブロックが含まれている場合、このブロックは無視されます。EOM ビットは、他のデバイスがアクティブであることを確認するために、ヘッダブロックにセットして他のデバイスを「ping」することができます。

確認応答ビットは、イニシエータによって常にハイインピーダンスにセットされています。これは、ヘッダの自己アドレスを読み出したフォロー、またはブロードキャストメッセージを拒否する必要があるフォローが、ローに駆動できるようにするためです。

ヘッダは、ソース論理アドレスフィールドおよび転送先論理アドレスフィールドで構成されています。ブロードキャストメッセージには、特殊アドレス 0xF が使用されている点に注意してください。

図 325. メッセージの構造

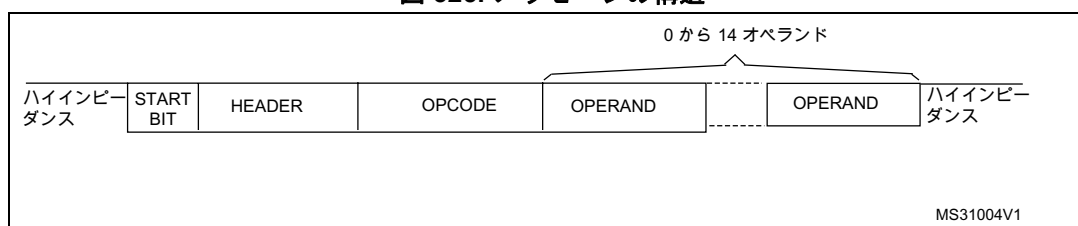


図 326. ブロック

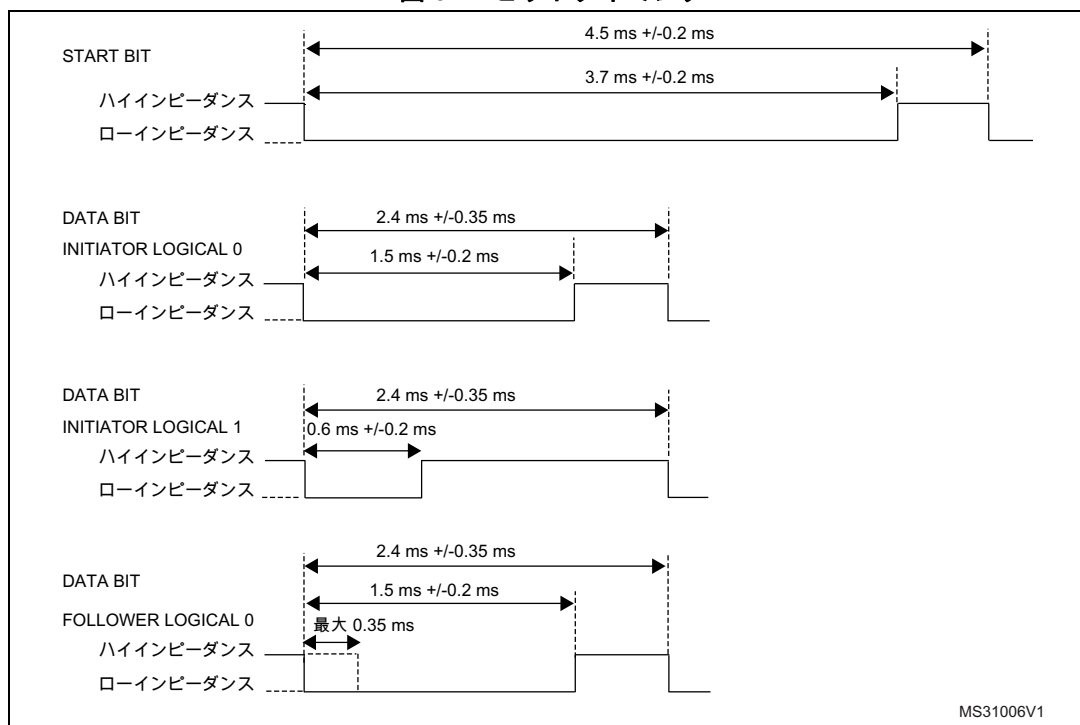


31.3.3 ビットタイミング

開始ビットの形式は一意で、メッセージの始まりを識別します。ロー時間および合計時間で検証されます。

メッセージの開始ビット後の残りのすべてのデータビットのタイミングは一定です。データビットの終わりのハイからローへの遷移は、CEC ラインがハイのままとなる最終ビットを除き、次のデータビットの始まりです。

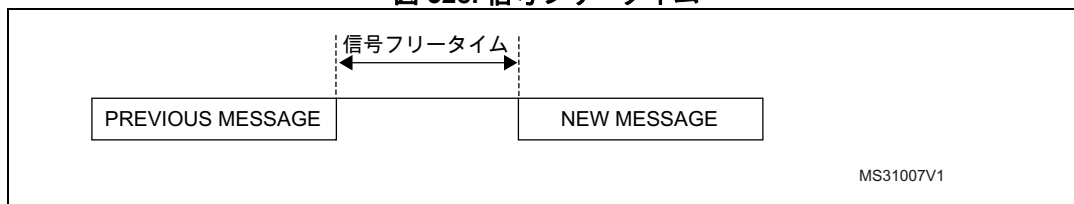
図 327. ビットタイミング



31.4 アービトレーション

CEC ラインにメッセージを送信または再送信する必要のあるすべてのデバイスは、いくつかのビット周期において無効であったことを確認する必要があります。信号フリータイムは、前のフレームの最終ビットからの開始時間として定義され、開始するデバイスと次の図に示す現在のステータスに依存します。

図 328. 信号フリータイム



1 度に使用できるイニシエータは 1 つのみであるため、同時に複数のイニシエータが送信を開始する場合に競合を避けるために、アービトレーション機構が提供されています。

CEC ラインアービトレーションは、開始ビットの先端から開始し、ヘッダブロック内のイニシエータアドレスビットの最後まで続けられます。この周期中、イニシエータは CEC ラインを監視し、このラインをハイインピーダンス状態に駆動する際に 0 に読み戻すと、アービトレーションを喪失したとみなして送信を停止し、フォロワになります。

図 329. アービトレーションフェーズ

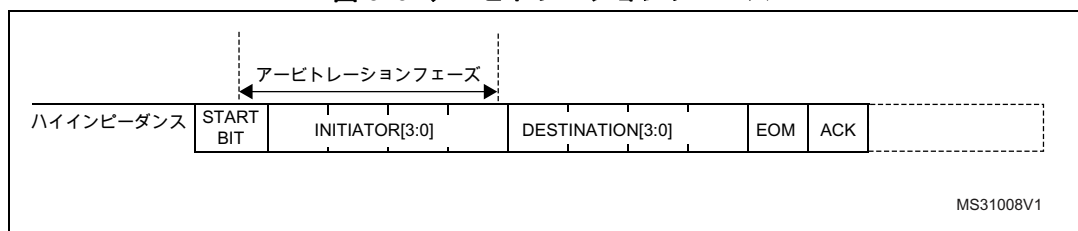
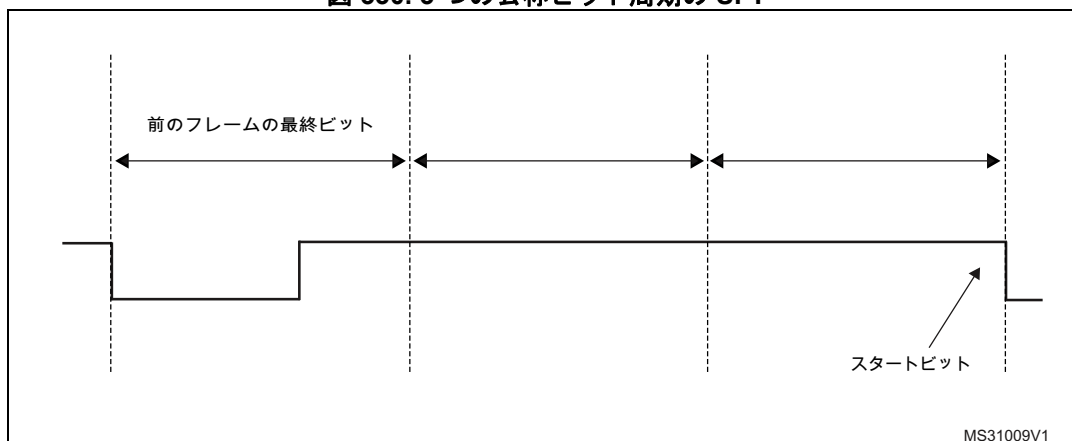


図 330 は、3 つの公称ビット周期の SFT の例を表します。

図 330. 3 つの公称ビット周期の SFT



送信を開始する前に、設定可能な時間枠がカウントされます。

SFT=0x0 設定では、HDMI-CEC デバイスが自動 SFT 計算を実行し、HDMI-CEC 規格に準拠していることを保証します。

- CEC が、送信が失敗した最後のバスイニシエータである場合、2.5 データビット周期
- CEC が新しいバスイニシエータである場合、4 データビット周期
- CEC が、送信が成功した最後のバスイニシエータである場合、6 データビット周期

これは、失敗した送信に最上位の優先順位、成功した送信の最後のイニシエータに最下位の優先順位を保証するために実行されます。

それ以外の場合は、固定されたタイミング値をカウントするために、SFT ビットを設定できる場合があります。可能な値は、0.5、1.5、2.5、3.5、4.5、5.5、6.5 データビット周期です。

31.4.1 SFT オプションビット

SFTOPT=0 の設定の場合、ソフトウェアによって送信開始コマンドがセットされると SFT のカウントが開始されます (TXSOM=1)。

SFTOPT=1 の場合、バスアイドル条件またはラインエラー条件が検出されると HDMI-CEC デバイスによって自動的に SFT のカウントが開始されます。TXSOM コマンドのセット時に SFT タイマが完了すると、送信は遅延なくただちに開始されます。SFT タイマが実行されたままである場合、システムは送信が開始される前にタイマが経過するまで待ちます。

SFTOPT=1 の場合、SFT タイマを開始させるバスイベント条件が、以下の場合に検出されます。

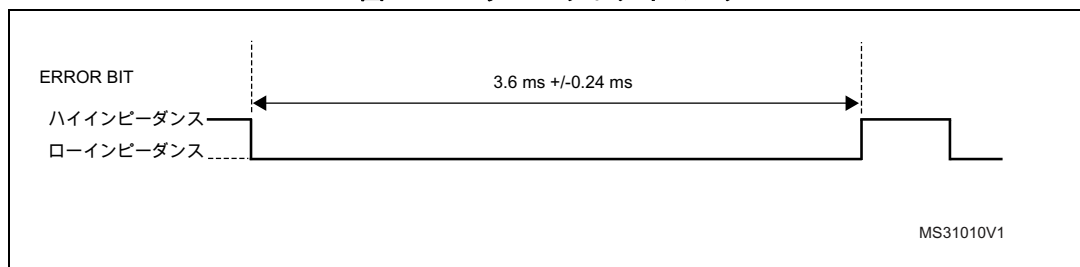
- 通常の送信／受信終了時に、TXEND/RXEND ビットはメッセージの最後のビット (ACK ビット) の最小公称データビット時間にセットされます。
- 送信エラーが検出された場合、TXERR 送信エラー検出時に SFT タイマが開始されます (TXERR=1)。
- CEC フォロワからの確認応答を喪失した場合、TXACK ビットがセットされたとき、つまり ACK ビットの公称サンプリング時間に、SFT タイマが開始されます。
- 送信アンダーランエラーが発生した場合、ACK ビットの終了時に TXUDR ビットをセットすると SFT タイマが開始されます。
- 受信のアボートを意味する受信エラーが検証された場合、SFT タイマはエラー検出と同じ時間に開始されます。エラービットが生成された場合、エラービットの終了時に SFT のカウントが開始されます。
- 開始ビットに誤りがある場合や、アイドル状態からコード化されていないローインピーダンスバス状態のいずれかである場合、SFT タイマはバスがハインピーダンスアイドル状態から戻るとすぐに再開されます。

31.5 エラー処理

31.5.1 ビットエラー

開始ビットを除くデータビットが無効であるとみなされる場合は、フォロワによってそのようなエラーが通知されることが期待されます。これには、公称データビット周期の 1.4 から 1.6 倍の CEC ライン上でロービット周期 (つまり、公称 3.6 ms) を生成して実行されます。

図 331. エラービットタイミング



31.5.2 メッセージエラー

メッセージは喪失したものとみなされ、以下の条件により再送信することができます。

- メッセージは、直接アドレス指定されたメッセージでは確認応答されません。
- メッセージは、ブロードキャストメッセージでは否定応答されます。
- 予期しないローインピーダンスが、CEC ライン上で検出されました (ラインエラー)。

CEC インタフェースがデータビットを受信する際に、3 種類のエラーフラグを検出できます。

31.5.3 ビット立ち上がりエラー (BRE)

ビット立ち上がりエラー (BRE) : 想定したウィンドウの外にビット立ち上がりエッジが検出されるとセットされます (図 332 を参照)。BRE フラグは BREIE=1 の場合にも CEC 割り込みを生成します。

BRE 検出の場合、BRESTP ビット値によってメッセージの受信を停止することができます。BREGEN ビットをセットすると、エラービットが生成されます。

BRESTP=1 のブロードキャストメッセージで BRE が検出されると、BREGEN=0 の場合でもエラービットが生成され、イニシエータによって失敗した送信の再試行が強制されます。エラービット生成は、BREGEN=0、BRDNOGEN=1 を設定することで無効化できます。

31.5.4 ショートビット周期エラー (SBPE)

SBPE は、ビット立ち下がりエッジが予想よりも早く検出された場合にセットされます (図 332 を参照)。SBPE フラグは SBPEIE=1 の場合にも CEC 割り込みを生成します。

エラービットは、SBPE エラー検出時に常にライン上に生成されます。エラービットは、リッスンモードのみがセットされ (LSTN=1)、以下の条件を満たしている場合には SBPE 検出時に生成されません。

- 直接アドレス指定されたメッセージは、SBPE とともに受信されます。
- ブロードキャストメッセージは、SBPE AND BRDNOGEN=1 とともに受信されます。

31.5.5 ロングビット周期エラー (LBPE)

LBPE は、有効なウィンドウでビット立ち下がりエッジが検出されない場合にセットされます (図 332 を参照)。LBPE フラグは LBPEIE=1 の場合にも CEC 割り込みを生成します。

LBPE は常に受信を停止し、LBPEGEN ビットがセットされるとライン上にエラービットが生成されます。

ブロードキャストメッセージで LBPE が検出されると、LBPEGEN=0 の場合でもエラービットが生成され、イニシエータによって失敗した送信の再試行が強制されます。エラービット生成は、LBPEGEN=0、BRDNOGEN=1 を設定することで無効化できます。

注 : **BREGEN=1、BRESTP=0 設定を避ける必要があります。**

図 332. エラー処理

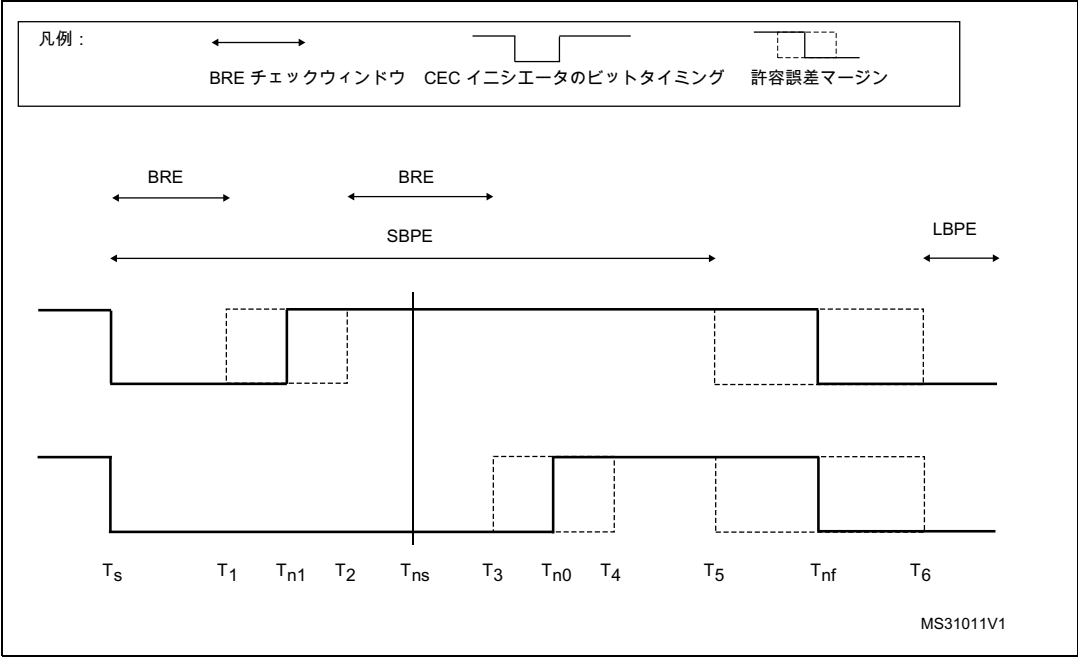


表 130. エラー処理タイミングパラメータ

時間	RXTOL	ms	説明
T_s	x	0	ビット開始イベント。
T_1	1	0.3	論理 1 を示すためにローからハイに遷移する一番早い時間。
	0	0.4	
T_{n1}	x	0.6	論理 1 を示すためにローからハイに遷移する公称時間。
T_2	0	0.8	論理 1 を示すためにローからハイに遷移する一番遅い時間。
	1	0.9	
T_{ns}	x	1.05	公称サンプリング時間。
T_3	1	1.2	ハイレインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番早い時間。
	0	1.3	
T_{n0}	x	1.5	ハイレインピーダンス状態 (論理 0) に戻るためにデバイスに許される公称時間。
T_4	0	1.7	ハイレインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番遅い時間。
	1	1.8	
T_5	1	1.85	後続のビットが開始される一番早い時間。
	0	2.05	
T_{nf}	x	2.4	公称データビット周期。
T_6	0	2.75	後続のビットが開始される一番遅い時間。
	1	2.95	

31.5.6 送信エラー検出 (TXERR)

CEC イニシエータでは、ハイインピーダンスの送信中でフォロウがビットをアサートしないと想定される場合に、CEC ラインでローインピーダンスを検出すると、TXERR フラグをセットします。TXERR フラグは TXERRIE=1 の場合にも CEC 割り込みを生成します。

TXERR のアサートによってメッセージの送信が停止されます。アプリケーションでは、失敗した送信を最大 5 回まで再試行することができます。

TXERR のチェックは、さまざまな CEC ラインの状態または RX 許容誤差設定によって、異なる方法で実行されます。

図 333. TXERR 検出

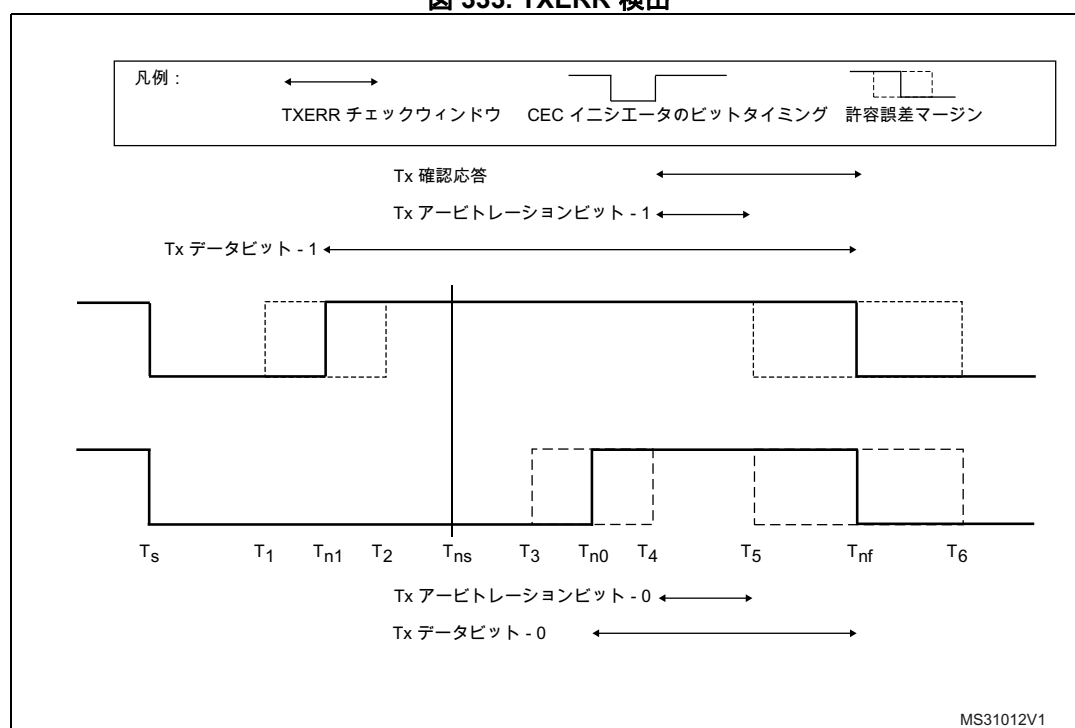


表 131. TXERR タイミングパラメータ

時間	RXTOL	ms	説明
T_s	x	0	ビット開始イベント。
T_1	1	0.3	論理 1 を示すためにローからハイに移移する一番早い時間。
	0	0.4	
T_{n1}	x	0.6	論理 1 を示すためにローからハイに移移する公称時間。
T_2	0	0.8	論理 1 を示すためにローからハイに移移する一番遅い時間。
	1	0.9	
T_{ns}	x	1.05	公称サンプリング時間。
T_3	1	1.2	ハイインピーダンス状態（論理 0）に戻るためにデバイスに許される一番早い時間。
	0	1.3	

表 131. TXERR タイミングパラメータ (続き)

時間	RXTOL	ms	説明
T_{n0}	x	1.5	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される公称時間。
T_4	0	1.7	ハイインピーダンス状態 (論理 0) に戻るためにデバイスに許される一番遅い時間。
	1	1.8	
T_5	1	1.85	後続のビットが開始される一番早い時間。
	0	2.05	
T_{nf}	x	2.4	公称データビット周期。
T_6	0	2.75	後続のビットが開始される一番遅い時間。
	1	2.95	

31.6 HDMI-CEC 割り込み

次の場合に割り込みを生成できます。

- 受信ブロック転送が完了または受信エラーが発生した場合の受信時。
- 送信ブロック転送が完了または送信エラーが発生した場合の送信時。

表 132. HDMI-CEC 割り込み

割り込みイベント	イベントフラグ	イネーブル制御ビット
Rx バイト受信	RXBR	RXBRIE
受信終了	RXEND	RXENDIE
Rx オーバーラン	RXOVR	RXOVRIE
Rx ビット立ち上がりエラー	BRE	BREIE
Rx ショートビット周期エラー	SBPE	SBPEIE
Rx ロングビット周期エラー	LBPE	LBPEIE
Rx 喪失確認応答エラー	RXACKE	RXACKIE
アービトレーション喪失	ARBLST	ARBLSTIE
Tx バイトリクエスト	TXBR	TXBRIE
送信終了	TXEND	TXENDIE
Tx バッファアンダーラン	TXUDR	TXUDRIE
Tx エラー	TXERR	TXERRIE
Tx 喪失確認応答エラー	TXACKE	TXACKIE

コード例については、「付録」セクション [CEC を設定する HDMI-CEC のコード例](#)、[割り込みが有効な HDMI-CEC 送信のコード例](#)、および [HDMI-CEC の割り込み管理のコード例](#)を参照してください。

31.7 HDMI-CEC レジスタ

レジスタの説明で使用されている略語のリストについては、[セクション 1.1 \(42 ページ\)](#) を参照してください。

31.7.1 CEC 制御レジスタ (CEC_CR)

アドレスオフセット : 0x00

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX EOM	TX SOM	CEC EN
													rs	rs	rw

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **TXEOM** : Tx メッセージの終了

TXEOM ビットは、CEC メッセージの最後のバイトの送信を指示するためにソフトウェアによってセットされます。

TXEOM は、TXSOM の場合と同時に、同じ条件下でハードウェアによってクリアされます。

0 : TXDR データバイトは EOM=0 で送信されます。

1 : TXDR データバイトは EOM=1 で送信されます。

注 : **TXEOM は、CECEN=1 のときにセットする必要があります。**

TXEOM は、TXDR に送信データを書き込む前にセットする必要があります。

TXSOM=0 のときに TXEOM がセットされると、送信メッセージは 1 バイト (HEADER) のみで構成されます (PING メッセージ)。

ビット 1 **TXSOM** : Tx メッセージの開始

TXSOM は、CEC メッセージの最初のバイトの送信を指示するためにソフトウェアによってセットされます。CEC メッセージが 1 バイトのみで構成される場合、TXEOM は TXSOM の前にセットする必要があります。

スタートビットは、SFT のカウント後に CEC ライン上で効果的に開始されます。メッセージの受信中に TXSOM がセットされると、送信は受信終了後に開始されます。

TXSOM は、送信アンダーラン (TXUDR=1)、否定応答 (TXACK=1)、および送信エラー (TXERR=1) が発生したときに、メッセージの最後のバイトが肯定応答 (TXEND=1) とともに送信された後で、ハードウェアによってクリアされます。また、CECEN=0 でもクリアされます。アービトレーション喪失 (ARBLST=1) の場合はクリアされず、送信は自動的に再試行されます。

TXSOM は、保留中や実行中の送信リクエストがあるかどうかをアプリケーションに知らせる、ステータスビットとしても使用できます。アプリケーションは、CECEN ビットをクリアすることで送信リクエストをいつでもアポートできます。

- 0 : 実行中の CEC 送信はありません。
- 1 : CEC 送信コマンド。

注： TXSOM は、CECEN=1 のときにセットする必要があります。
TXSOM は、送信データを TXDR で使用できる場合にセットする必要があります。
自己のペリフェラルアドレスを含む HEADER の最初の 4 つのビットは、受信用としてのみ使用される CEC_CFGR.OAR ではなく、TXDR[7:4] から取得されます。

ビット 0 CECEN : CEC 有効

CECEN ビットは、ソフトウェアによってセット／クリアされます。CECEN=1 の場合にメッセージの受信が開始され、TXSOM 制御が有効化されます。CECEN=0 の場合に CEC ペリフェラルが無効化され、CEC_CR レジスタのすべてのビットがクリアされ、実行中のすべての送受信がアポートされます。

- 0 : CEC ペリフェラルはオフです。
- 1 : CEC ペリフェラルはオンです。

31.7.2 CEC 設定レジスタ (CEC_CFGR)

このレジスタは HDMI-CEC コントローラの設定に使用されます。

アドレスオフセット : 0x04

リセット値 : 0x0000 0000

注意： CECEN=0 の場合のみ、CEC_CFGR に書き込む必要があります。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSTN	OAR[14:0]														
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SFT OPT	BRDNO GEN	LBPE GEN	BRE GEN	BRE STP	RX TOL	SFT[2:0]		
							rw	rw	rw	rw	rw	rw	rw		

ビット 31 LSTN : リスンモード

LSTN ビットは、ソフトウェアによってセット／クリアされます。

0 : CEC ペリフェラルでは、自己アドレス (OAR) に宛てられたメッセージのみを受信します。他の転送先に宛てられたメッセージは無視されます。ブロードキャストメッセージは常に受信されます。

1 : CEC ペリフェラルでは、肯定応答とともに自己アドレス (OAR) に宛てられたメッセージを受信します。他の転送先に宛てられたメッセージは受信されますが、CEC バスに干渉することはありません。確認応答は送信されません。

ビット 30:16 OAR : 自己アドレス設定

OAR ビットは、受信モードで考慮する必要がある転送先論理アドレスを選択するために、ソフトウェアによってセットされます。各ビットをセットすると、特定のビット位置によって指定された CEC 論理アドレスが有効になります。

HEADER 受信終了時に、受信した転送先アドレスが有効化されたアドレスと比較されます。アドレスが一致する場合は、受信メッセージが確認応答され、受信されます。アドレスが一致しない場合は、受信メッセージはリスンモード (LSTN=1) でのみ受信され、確認応答は送信されません。ブロードキャストメッセージは常に受信されます。

例 :

OAR = 0b000 0000 0010 0001 は、アドレス 0x0 および 0x5 が CEC で確認応答されることを意味します。結果として、これらのアドレスのいずれかにそれぞれ宛てられたメッセージが受信されます。

ビット 15:9 予約済みであり、リセット値に保持する必要があります。

ビット 8 SFTOP : SFT オプションビット

SFTOPT ビットは、ソフトウェアによってセット／クリアされます。

0 : ソフトウェアによって TXSOM がセットされると、SFT タイマが開始されます。

1 : メッセージの送受信の終了時に SFT タイマが自動的に開始されます。

ビット 7 BRDNOGEN : ブロードキャストでの生成エラービットの回避

BRDNOGEN ビットは、ソフトウェアによってセット／クリアされます。

0 : BRESTP=1 および BREGEN=0 のブロードキャストメッセージで BRE が検出されると、CEC ライン上にエラービットが生成されます。LBPEGEN=0 のブロードキャストメッセージで LBPE が検出されると、CEC ライン上にエラービットが生成されます。

1 : エラービットは、上記と同じ条件では生成されません。リスンモードがセットされている場合、ブロードキャストメッセージで SBPE が検出されてもエラービットは生成されません。

ビット 6 LBPEGEN : ロングビット周期エラーでのエラービットの生成

LBPEGEN ビットは、ソフトウェアによってセット／クリアされます。

0 : LBPE 検出で、CEC ライン上にエラービットは生成されません。

1 : LBPE 検出で、CEC ライン上にエラービットが生成されます。

注 : **BRDNOGEN=0 の場合、LBPEGEN=0 の場合でもブロードキャストでは LBPE 検出時にエラービットが生成されます。**

ビット 5 BREGEN : ビット立ち上がりエラーでのエラービットの生成

BREGEN ビットは、ソフトウェアによってセット／クリアされます。

0 : BRE 検出で、CEC ライン上にエラービットは生成されません。

1 : BRE 検出で、CEC ライン上にエラービットが生成されます (BRESTP がセットされている場合)。

注 : **BRDNOGEN=0 の場合、BREGEN=0 の場合でもブロードキャストで BRESTP=1 の BRE 検出時にエラービットが生成されます。**

ビット 4 BRESTOP : ビット立ち上がりエラーでの Rx 停止

BRESTOP ビットは、ソフトウェアによってセット／クリアされます。

0 : BRE 検出で CEC メッセージの受信が停止されることはありません。データビットは 1.05 ms でサンプリングされます。

1 : BRE 検出でメッセージの受信が停止されます。

ビット 3 RXTOL : Rx 許容誤差

RXTOL ビットは、ソフトウェアによってセット／クリアされます。

0 : 標準許容誤差マージン

- スタートビット : +/- 200 μ s 立ち上がり、+/- 200 μ s 立ち下がり
- データビット : +/- 200 μ s 立ち上がり、+/- 350 μ s 立ち下がり

1 : 拡張された許容誤差

- スタートビット : +/- 400 μ s 立ち上がり、+/- 400 μ s 立ち下がり
- データビット : +/- 300 μ s 立ち上がり、+/- 500 μ s 立ち下がり

ビット 2:0 SFT : 信号フリータイム

SFT ビットは、ソフトウェアによってセットされます。SFT=0x0 設定では、送信前に待った公称データビット周期数は、送信履歴に従って、ハードウェアによって支配されます。その他の設定では、SFT 番号はソフトウェアによって決定されます。

- 0x0
 - CEC が、送信が失敗した最後のバスイニシエータである場合、2.5 データビット周期 (ARBLST=1、TXERR=1、TXUDR=1、または TXACK= 1)
 - CEC が新しいバスイニシエータである場合、4 データビット周期
 - CEC が、送信が成功した最後のバスイニシエータである場合、6 データビット周期 (TXEOM=1)
- 0x1 : 0.5 公称データビット周期
- 0x2 : 1.5 公称データビット周期
- 0x3 : 2.5 公称データビット周期
- 0x4 : 3.5 公称データビット周期
- 0x5 : 4.5 公称データビット周期
- 0x6 : 5.5 公称データビット周期
- 0x7 : 6.5 公称データビット周期

31.7.3 CEC Tx データレジスタ (CEC_TXDR)

アドレスオフセット : 0x8

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]							
								w	w	w	w	w	w	w	w

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **TXD[7:0]** : Tx データレジスタ

TXD は、送信されるデータバイトを含む書き込み専用のレジスタです。

注 : TXD は、TXSTART=1 のときに書き込む必要があります。

31.7.4 CEC Rx データレジスタ (CEC_RXDR)

アドレスオフセット : 0xC

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]							
								r	r	r	r	r	r	r	r

ビット 31:8 予約済みであり、リセット値に保持する必要があります。

ビット 7:0 **RXD[7:0]** : Rx データレジスタ

RXD は読み出し専用で、CEC ラインから受信した最後のデータバイトを含みます。

31.7.5 CEC 割り込みおよびステータスレジスタ (CEC_ISR)

アドレスオフセット : 0x10

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TX ACKE	TX ERR	TX UDR	TX END	TXBR	ARB LST	RX ACKE	LBPE	SBPE	BRE	RX OVR	RX END	RXBR
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

ビット 31:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **TXACK** : Tx 喪失確認応答エラー

送信モードでは、TXACK はハードウェアによってセットされ、アプリケーションに確認応答を受信していないことを知らせます。ブロードキャスト送信の場合、TXACK はアプリケーションに否定応答を受信したことを知らせます。TXACK はメッセージの送信をアボートし、TXSOM および TXEOM 制御をクリアします。

TXACK はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 11 **TXERR** : Tx エラー

送信モードでは、CEC イニシエータがリリース中に CEC ラインでローインピーダンスを検出した場合に、ハードウェアによって TXERR がセットされます。TXERR はメッセージの送信をアボートし、TXSOM および TXEOM 制御をクリアします。

TXERR はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 10 **TXUDR** : Tx バッファアンダーラン

送信モードでは、次のバイト送信前の TXDR のロードにアプリケーションが間に合わなかった場合に、ハードウェアによって TXUDR がセットされます。TXUDR はメッセージの送信をアボートし、TXSOM および TXEOM 制御ビットをクリアします。

TXUDR はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 9 **TXEND** : 送信終了

TXEND は、CEC メッセージの最後のバイトが正常に送信されたことをアプリケーションに知らせるために、ハードウェアによってセットされます。TXEND は、TXSOM および TXEOM 制御ビットをクリアします。

TXEND はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 8 **TXBR** : Tx バイトリクエスト

TXBR は、次の送信データを TXDR に書き込む必要があることをアプリケーションに知らせるために、ハードウェアによってセットされます。TXBR は、現在送信されたバイトの 4 番目のビットが送信されたときにセットされます。アプリケーションでは、送信アンダーランエラー (TXUDR) が発生する前に、6 公称データビット周期内の TXDR に次のバイトを書き込む必要があります。

TXBR はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 7 **ARBLST** : アービトレーション喪失

ARBLST は、TXSOM コマンドに続くアービトレーション喪失イベントによって CEC デバイスが受信に切り換わることをアプリケーションに知らせるために、ハードウェアによってセットされます。ARBLST の原因は、先に開始される、またはより高い HEADER 優先順位で同時に開始される、競合するいずれかの CEC デバイスです。ARBLST のアサート後、TXSOM ビットでは次の送信試行を保留のままとします。

ARBLST はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 6 **RXACK** : Rx 喪失確認応答

受信モードでは、RXACK はハードウェアによってセットされ、CEC ラインで確認応答を確認できなかったことをアプリケーションに知らせます。RXACK はブロードキャストメッセージにのみ適用でき、またリスンモードでは直接アドレス指定されていないメッセージにも適用できます (転送先アドレスは OAR では無効)。RXACK はメッセージの受信をアボートします。

RXACK はソフトウェアで 1 を書き込むことによってクリアされます。

ビット 5 **LBPE** : Rx ロングビット周期エラー

ロングビット周期エラーのデータビット波形が検出された場合、LBPE はハードウェアによってセットされます。立ち下がりエッジが存在している場合、LBPE は、RXTOL が許容する最大ビット拡張許容誤差の最後にセットされます。LBPE は、常に CEC メッセージの受信を停止します。LBPE で、LBPEGEN=1 のときに CEC ライン上にエラービットが生成されます。ブロードキャストの場合は、LBPEGEN=0 の場合でもエラービットが生成されます。

LBPE はソフトウェアで 1 を書き込むことによってクリアされます。

- ビット 4 **SBPE** : Rx ショートビット周期エラー
- ショートビット周期エラーのデータビット波形が検出された場合、SBPE はハードウェアによってセットされます。SBPE は、想定された立ち上がりエッジの発生時にセットされます。SBPE で、CEC ライン上にエラービットが生成されます。
- SBPE はソフトウェアで 1 を書き込むことによってクリアされます。
- ビット 3 **BRE** : Rx ビット立ち上がりエラー
- ビット立ち上がりエラーのデータビット波形が検出された場合、BRE はハードウェアによってセットされます。立ち上がりエッジが存在している場合、BRE は、立ち上がりエッジの誤配置が発生したとき、または RXTOL が許容する最大 BRE 許容誤差の最後にセットされます。BRESTP=1 のとき、BRE ではメッセージの受信を停止します。BRE で、BREGEN=1 のときに CEC ライン上にエラービットが生成されます。
- BRE はソフトウェアで 1 を書き込むことによってクリアされます。
- ビット 2 **RXOVR** : Rx オーバーラン
- 新しいバイトの受信時に RXBR がクリアされていない場合や RXD に保存されていない場合、RXOVR はハードウェアによってセットされます。RXOVR アサートではメッセージの受信を停止するため、確認応答は送信されません。ブロードキャストの場合、否定応答が送信されます。
- RXOVR はソフトウェアで 1 を書き込むことによってクリアされます。
- ビット 1 **RXEND** : 受信終了
- RXEND は、CEC ラインから CEC メッセージの最後のバイトを受信し、RXD バッファに格納したことをアプリケーションに知らせるために、ハードウェアによってセットされます。RXEND は、RXBR と同じ時間でセットされます。
- RXEND はソフトウェアで 1 を書き込むことによってクリアされます。
- ビット 0 **RXBR** : Rx バイト受信
- RXBR ビットは、CEC ラインから新しいバイトを受信し、RXD バッファに格納したことをアプリケーションに知らせるために、ハードウェアによってセットされます。
- RXBR はソフトウェアで 1 を書き込むことによってクリアされます。

31.7.6 CEC 割り込み有効レジスタ (CEC_IER)

アドレスオフセット : 0x14

リセット値 : 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TXACK IE	TXERR IE	TX UDRIE	TXEND IE	TXBR IE	ARBLST IE	RXACK IE	LBPE IE	SBPE IE	BREIE	RXOVR IE	RXEND IE	RXBR IE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- ビット 31:13 予約済みであり、リセット値に保持する必要があります。
- ビット 12 **TXACKIE** : Tx 喪失確認応答エラー割り込み有効
- TXACKEIE ビットは、ソフトウェアによってセット／クリアされます。
- 0 : TXACKE 割り込み無効
- 1 : TXACKE 割り込み有効
- ビット 11 **TXERRIE** : Tx エラー割り込み有効
- TXERRIE ビットは、ソフトウェアによってセット／クリアされます。
- 0 : TXERR 割り込み無効
- 1 : TXERR 割り込み有効



- ビット 10 **TXUDRIE** : Tx アンダーラン割り込み有効
TXUDRIE ビットは、ソフトウェアによってセット／クリアされます。
0 : TXUDR 割り込み無効
1 : TXUDR 割り込み有効
- ビット 9 **TXENDIE** : Tx メッセージの終了割り込み有効
TXENDIE ビットは、ソフトウェアによってセット／クリアされます。
0 : TXEND 割り込み無効
1 : TXEND 割り込み有効
- ビット 8 **TXBRIE** : Tx バイトリクエスト割り込み有効
TXBRIE ビットは、ソフトウェアによってセット／クリアされます。
0 : TXBR 割り込み無効
1 : TXBR 割り込み有効
- ビット 7 **ARBLSTIE** : アービトレーション喪失割り込み有効
ARBLSTIE ビットは、ソフトウェアによってセット／クリアされます。
0 : ARBLST 割り込み無効
1 : ARBLST 割り込み有効
- ビット 6 **RXACKIE** : Rx 喪失確認応答エラー割り込み有効
RXACKIE ビットは、ソフトウェアによってセット／クリアされます。
0 : RXACHE 割り込み無効
1 : RXACHE 割り込み有効
- ビット 5 **LBPEIE** : ロングビット周期エラー割り込み有効
LBPEIE ビットは、ソフトウェアによってセット／クリアされます。
0 : LBPE 割り込み無効
1 : LBPE 割り込み有効
- ビット 4 **SBPEIE** : ショートビット周期エラー割り込み有効
SBPEIE ビットは、ソフトウェアによってセット／クリアされます。
0 : SBPE 割り込み無効
1 : SBPE 割り込み有効
- ビット 3 **BREIE** : ビット立ち上がりエラー割り込み有効
BREIE ビットは、ソフトウェアによってセット／クリアされます。
0 : BRE 割り込み無効
1 : BRE 割り込み有効
- ビット 2 **RXOVRIE** : Rx バッファオーバーラン割り込み有効
RXOVRIE ビットは、ソフトウェアによってセット／クリアされます。
0 : RXOVR 割り込み無効
1 : RXOVR 割り込み有効
- ビット 1 **RXENDIE** : 受信終了割り込み有効
RXENDIE ビットは、ソフトウェアによってセット／クリアされます。
0 : RXEND 割り込み無効
1 : RXEND 割り込み有効
- ビット 0 **RXBRIE** : Rx バイト受信割り込み有効
RXBRIE ビットは、ソフトウェアによってセット／クリアされます。
0 : RXBR 割り込み無効
1 : RXBR 割り込み有効

注意 : (*) **CECEN=0 の場合は、CEC_IER に書き込む必要があります。**



31.7.7 HDMI-CEC レジスタマップ

次の表に HDMI-CEC レジスタの一覧を示します。

表 133. HDMI-CEC レジスタマップとリセット値

オフ セット	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	CEC_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEOM	TXSOM	CECEN
	リセット値																													0	0	0	
0x04	CEC_CFGR	LSTN	OAR[14:0]														Res.	Res.	Res.	Res.	Res.	Res.	SFTOPT	BRDNOGEN	LBPEGEN	BREGEN	BRESTP	RXTOL	SFT[2:0]				
	リセット値	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						Res.	Res.	Res.		0	0	0	0	0	0	0	0
0x08	CEC_TXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]								
	リセット値																									0	0	0	0	0	0	0	0
0x0C	CEC_RXDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]								
	リセット値																																
0x10	CEC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXACKIE	TXERRIE	TXUDRIE	TXENDIE	TXBRIE	ARBLSSTIE	RXACKIE	LBPEIE	SBPEIE	BREIE	RXOVRIE	RXENDIE	RXBRIE
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	CEC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXACKIE	TXERRIE	TXUDRIE	TXENDIE	TXBRIE	ARBLSSTIE	RXACKIE	LBPEIE	SBPEIE	BREIE	RXOVRIE	RXENDIE	RXBRIE
	リセット値																				0	0	0	0	0	0	0	0	0	0	0	0	0

レジスタ境界アドレスについては、[セクション 2.2.2 \(47 ページ\)](#) を参照してください。

32 デバッグサポート (DBG)

32.1 概要

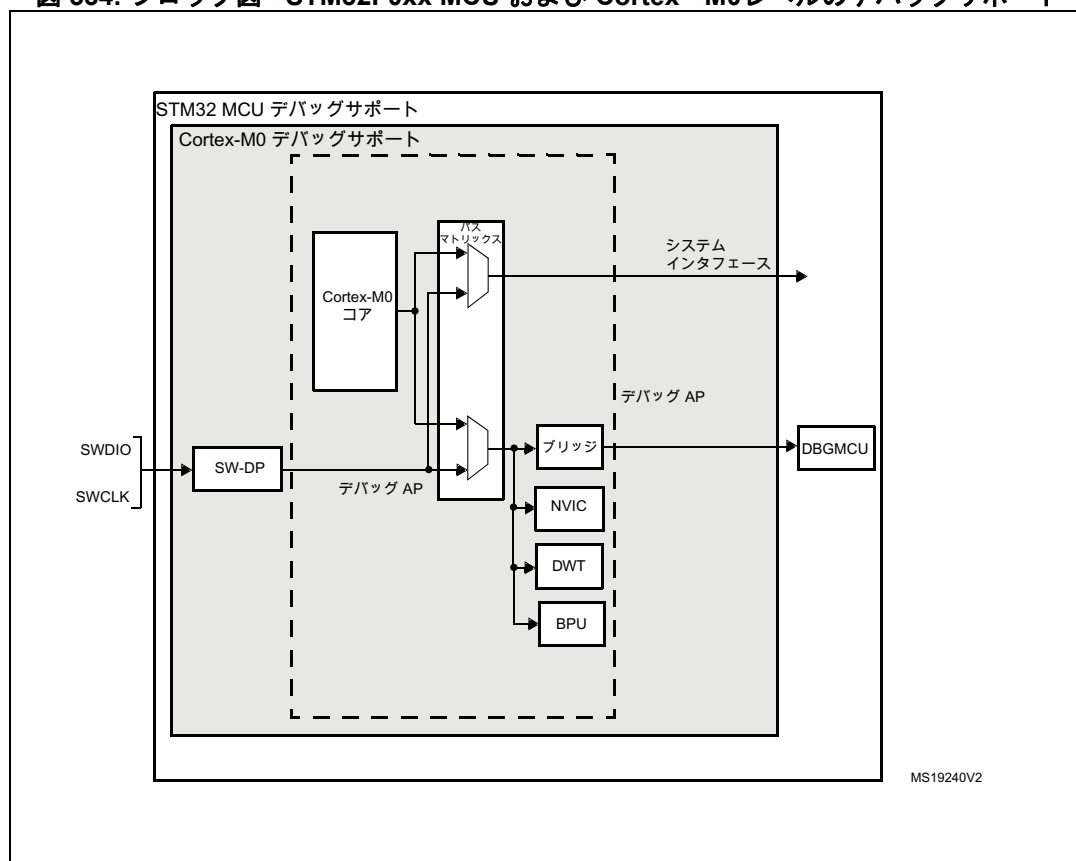
STM32F0xx は、Cortex®-M0 コアを中心にして構築されており、高度なデバッグ機能のためのハードウェア拡張機能を含んでいます。デバッグ拡張機能によって、コアは、特定の命令フェッチ時（ブレークポイント）またはデータアクセス時（ウォッチポイント）に停止することができます。停止したとき、コアの内部状態とシステムの外部状態を調査することができます。調査が終わったら、コアとシステムを復元して、プログラム実行を再開することができます。

デバッグ機能は、STM32F0xx MCU への接続時とデバッグ時にデバッガによって使用されます。

デバッグ用に、次のインターフェースを使用できます。

- シリアルワイヤ

図 334. ブロック図 - STM32F0xx MCU および Cortex®-M0レベルのデバッグサポート



1. Cortex®-M0 コアに内蔵されているデバッグ機能は、ARM CoreSight Design Kit のサブセットです。

ARM Cortex®-M0 コアは、統合的なオンチップデバッグサポート機能を提供します。この機能は以下の要素で構成されます。

- SW-DP : シリアルワイヤ
- BPU : ブレークポイントユニット
- DWT : データウォッチポイントトリガ

また、STM32F0xx専用の以下のデバッグ機能も内蔵されています。

- 柔軟性の高いデバッグピンの割り当て
- MCU デバッグボックス（低電力モードのサポート、ペリフェラルクロックの制御など）

注： **ARM Cortex®-M0 コアがサポートするデバッグ機能の詳細については、Cortex®-M0 Technical Reference Manual を参照してください（セクション 32.2：ARM リファレンス資料）。**

32.2 ARM リファレンス資料

- Cortex®-M0 Technical Reference Manual (TRM)
以下から入手可能です。
<http://infocenter.arm.com>
- ARM Debug Interface V5
- ARM CoreSight Design Kit revision r1p1 Technical Reference Manual

32.3 ピン名とデバッグポートピン

STM32F0xx MCUは、使用できるピン数の異なるさまざまなパッケージに組み込まれています。

32.3.1 SWD ポートピン

汎用入出力のオルタネート機能として、2 個のピンが SW-DP 用の出力に使用されます。これらのピンはすべてのパッケージで使用できます。

表 134. SW デバッグポートピン

SW-DP ピン名	SW デバッグポート		ピン割り当て
	タイプ	デバッグ割り当て	
SWDIO	IO	シリアルワイヤデータ入出力	PA13
SWCLK	I	シリアルワイヤクロック	PA14

32.3.2 SW-DP ピンの割り当て

リセット後（SYSRESETn または PORESETn）、SW-DP 用に使用されるピンは、デバッグホストによってすぐに使用可能な専用ピンとして割り当てられます。

ただし、MCU は SWD ポートを無効にすることができますので、汎用 IO（GPIO）に使用する関連ピンを解放する可能性もあります。SW-DPポートピンを無効にする方法の詳細については、[セクション 9.3.2：I/O ピンオルタネート機能マルチプレクサと配置（159 ページ）](#)を参照してください。

32.3.3 SWD ピンでの内部プルアップ／プルダウン

SW 入出力がユーザソフトウェアによって解放されると、GPIO コントローラがこれらのピンを制御します。GPIO 制御レジスタがリセット状態にあるとき、入出力も同じ状態になります。

- SWDIO：入力プルアップ
- SWCLK：入力プルダウン

プルアップおよびプルダウン抵抗を内蔵しているため、外部抵抗を追加する必要がありません。

32.4 ID コードとロック機構

MCU 内部には、いくつかの ID コードがあります。ST 社がツール設計者（たとえば、Keil 社、IAR 社、Raisonance 社）に強く推奨するのは、アドレス 0x40015800 に配置されている MCU デバイス ID コードを使用してデバッグをロックすることです。

デバッグ/プログラマツールでは、DEV_ID[15:0] のみを識別に使用してください（リビジョン ID を考慮しません）。

32.4.1 MCU デバイス ID コード

STM32F0xx 製品には MCU ID コードが内蔵されています。この ID は、ST 社製 MCU の部品番号とダイのリビジョンを識別します。

このコードは、ソフトウェアデバッグポート（2 本のピン）、またはユーザソフトウェアによってアクセスできます。

コード例については、「付録」セクション [A.12.1 : DBG 読み出しデバイス ID のコード例](#)を参照してください。

DBGMCU_IDCODE

アドレス : 0x40015800

32 ビットアクセスのみサポートされます。読み出し専用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID											
					r	r	r	r	r	r	r	r	r	r	r

ビット 31:16 **REV_ID(15:0)** リビジョン識別子

このフィールドは、デバイスのリビジョンを示します。リビジョン 1.0 では 0x1000 が、リビジョン 2.0 では 0x2000 が読み出されます。

ビット 15:12 予約済み。0b0110 を読み出します。

ビット 11:0 **DEV_ID(11:0)** : デバイス識別子

このフィールドは、デバイス ID を示します。

STM32F03x デバイスでは、デバイス ID は 0x444 です。

STM32F04x デバイスでは、デバイス ID は 0x445 です。

STM32F05x デバイスでは、デバイス ID は 0x440 です。

STM32F07x デバイスでは、デバイス ID は 0x448 です。

STM32F09x デバイスでは、デバイス ID は 0x442 です。



32.5 SWD ポート

32.5.1 SWD プロトコルの概要

この同期式シリアルプロトコルでは、次の 2 個のピンを使用します。

- SWCLK : ホストからターゲットへのクロック
- SWDIO : 双方向

このプロトコルでは、2 バンクのレジスタ (DPACC レジスタと APACC レジスタ) の読み出し／書き込みが可能です。

ビットは、ワイヤ上を LSB ファーストで転送されます。

SWDIO を双方向管理するには、ボード上でラインがプルアップされている必要があります (ARM の推奨値は 100 kΩ)。

プロトコルで SWDIO の方向が変化するたびに、ラインがホストからもターゲットからも駆動されないターンアラウンド時間が挿入されます。このターンアラウンド時間のデフォルトは 1 ビット時間ですが、SWCLK 周波数の設定によって調整できます。

32.5.2 SWD プロトコルシーケンス

各シーケンスは 3 つのフェーズで構成されます。

1. ホストによって送信されるパケットリクエスト (8 ビット)
2. ターゲットによって送信される確認応答 (3 ビット)
3. ホストまたはターゲットによって送信されるデータ転送フェーズ (33 ビット)

表 135. パケットリクエスト (8 ビット)

ビット	名前	説明
0	START	"1"である必要があります。
1	APnDP	0 : DP アクセス 1 : AP アクセス
2	RnW	0 : 書き込みリクエスト 1 : 読み出しリクエスト
4:3	A[3:2]	DP/AP レジスタのアドレスフィールド (表 139 (928 ページ) を参照)
5	Parity	先行するビットの 1 ビットパリティ
6	STOP	0
7	Park	ホストによって駆動されません。プルアップの効果で、ターゲットには「1」と読み出される必要があります。

DPACC レジスタと APACC レジスタの詳細については、Cortex®-M0 TRM を参照してください。

パケットリクエストの後には、必ずホストもターゲットもラインを駆動しないターンアラウンド時間 (デフォルトでは 1 ビット) が続きます。

表 136. ACK 応答 (3 ビット)

ビット	名前	説明
0~2	ACK	001 : FAULT 010 : WAIT 100 : OK

読み出しトランザクションの場合や、受信した ACK 応答が"WAIT" または"FAULT" の場合にのみ、ACK応答の後にターンアラウンド時間が続く必要があります。

表 137. データ転送 (33 ビット)

ビット	名前	説明
0~31	WDATA または RDATA	書き込み／読み出しデータ
32	Parity	32 データビットの 1 ビットパリティ

読み出しトランザクションの場合にのみ、データ転送の後にターンアラウンド時間が続く必要があります。

32.5.3 SW-DP ステートマシン (リセット、アイドル状態、ID コード)

SW-DP のステートマシンには、SW-DP を識別する内部 ID コードがあります。これは JEP-106 規格に準じています。この ID コードは、ARM のデフォルトであり、**0x0BB11477** (Cortex[®]-M0に対応) がセットされています。

注： **SW-DP ステートマシンは、ターゲットがこの ID コードを読み出すまで非アクティブであることに注意してください。**

- パワーオンリセット後、またはラインが 50 サイクルより長い間ハイレベルにあったあとでは、SW-DP ステートマシンはリセット状態になります。
- リセット状態のあと、ラインが 2 サイクル以上の間ローレベルであれば、SW-DP ステートマシンはアイドル状態になります。
- リセット状態のあとは、まずアイドル状態に入り、次に DP-SW ID CODE レジスタの読み出しアクセスを行う **必要があります**。そうしないと、ターゲットは、他のトランザクションに対して ACK 応答の"FAULT"を発行します。

SW-DP ステートマシンの詳細については、Cortex[®]-M0 TRM および CoreSight Design Kit r1p0 TRM を参照してください。

32.5.4 DP と AP の読み出し／書き込みアクセス

- DP への読み出しアクセスはポストされません。つまり、ターゲットは、ACK 応答が"OK"の場合はただちに応答し、ACK 応答が"WAIT"の場合は遅れて応答します。
- AP への読み出しアクセスはポストされます。つまり、アクセスの結果は次の転送時に返されます。次のアクセスが AP アクセスでない場合、結果を得るには DP-RDBUFF レジスタを読み出す必要があります。
AP 読み出しアクセスが成功したかどうかを判断するため、DP-CTRL/STAT レジスタの READOK フラグは、AP 読み出しアクセスまたは RDBUFF 読み出しリクエストのたびに更新されます。
- SW-DP は、DP と AP の両方の書き込みに使用できる書き込みバッファを実装しているため、たとえ他のトランザクションが未処理であっても、書き込み動作を受け付けることができます。書き込みバッファがフルのとき、ターゲットの ACK 応答は"WAIT"です。例外として、IDCODE 読み出し、CTRL/STAT 読み出し、または ABORT 書き込みは、書き込みバッファがフルであっても受け付けられます。
- 非同期クロックドメイン SWCLK と HCLK によって、書き込みを内部的に有効にするには、書き込みトランザクション後（パリティビット後）に SWCLK の 2 サイクルが余分に必要となります。これらのサイクルは、ラインをローレベルに駆動している間（アイドル状態）に適用してください。
これは、パワーアップリクエストのために CTRL/STAT の書き込みを行う際に特に重要です。パワーアップを必要とする次のトランザクションがただちに発生すると、そのトランザクションは失敗します。

32.5.5 SW-DP レジスタ

これらのレジスタへのアクセスは、APnDP = 0 のときに開始されます。

表 138. SW-DP レジスタ

A[3:2]	読み出し／ 書き込み	SELECT レジスタの CTRLSEL ビット	レジスタ	注
00	読み出し		IDCODE	製造者コードは、Cortex-M0 のデフォルトの ARM コードに設定されます。 0x0BB11477 (SW-DP を識別)。
00	書き込み		ABORT	
01	読み出し／ 書き込み	0	DP CTRL/STAT	用途は以下のとおりです。 – システム／デバッグパワーアップのリクエスト – AP アクセス用の転送動作の設定 – プッシュ比較／プッシュ検証動作の制御 – ステータスフラグ（オーバーラン、パワーアップ の確認応答）の読み出し
01	読み出し／ 書き込み	1	WIRE CONTROL	物理的なシリアルポートプロトコルの設定（ターン アラウンド時間など）を行います。
10	読み出し		READ RESEND	元の AP 転送を反復しなくても、破壊されたデバッ グ転送からの読み出しデータの復旧を可能にしま す。
10	書き込み		SELECT	現在のアクセスポートとアクティブな 4 ワードレ ジスタウィンドウを選択します。
11	読み出し／ 書き込み		読 み 出 し バ ッ フ	AP アクセスはポストされるため、この読み出し バッファは効果的です (AP 読み出しリクエストの 結果は、次の AP トランザクションで取得できる)。 この読み出しバッファは、新しいトランザクショ ンを開始することなく、前回の読み出しの結果と して AP から出力されるデータをキャプチャしま す。

32.5.6 SW-AP レジスタ

これらのレジスタへのアクセスは、APnDP = 1 のときに開始されます。

次の項目の組み合わせで、多くの AP レジスタをアドレス指定します。

- シフトされた値 A[3:2]
- DP SELECT レジスタの現在値

表 139. シフトされた値 A[3:2] によってアドレス指定される 32 ビットデバッグポートレジスタ

アドレス	A[3:2] の値	説明
0x0	00	予約済みであり、リセット値に保持する必要があります。
0x4	01	DP CTRL/STAT レジスタ：次の目的で使用されます。 – システム/デバッグパワーアップのリクエスト – AP アクセス用の転送動作の設定 – プッシュ比較/プッシュ検証動作の制御 – ステータスフラグ（オーバーラン、パワーアップの確認応答）の読み出し
0x8	10	DP SELECT レジスタ：現在のアクセスポートとアクティブな 4 ワードレジスタウィンドウの選択に使用されます。 – ビット 31:24 (APSEL)：現在の AP を選択します。 – ビット 23:8：予約済み – ビット 7:4 (APBANKSEL)：現在の AP でアクティブな 4 ワードレジスタウィンドウを選択します。 – ビット 3:0：予約済み
0xC	11	DP RDBUFF レジスタ：一連の動作のあとで（新しい JTAG-DP 動作をリクエストせずに）、デバッグが最終結果を得られるようにします。

32.6 コアデバッグ

コアデバッグはコアデバッグレジスタを通じてアクセスされます。これらのレジスタへのデバッグアクセスには、デバッグアクセスポートを使用します。コアデバッグは 4 個のレジスタで構成されています。

表 140. コアデバッグレジスタ

レジスタ	説明
DHCSR	32 ビットのデバッグ停止制御/ステータスレジスタ： レジスタプロセッサの状態についてのステータス情報を提供し、コアデバッグを有効にし、プロセッサの停止とステップ実行を行います。
DCRSR	17 ビットのデバッグコアレジスタセクタレジスタ： データの転送先または転送元となるプロセッサレジスタを選択します。
DCRDR	32 ビットのデバッグコアレジスタデータレジスタ： DCRSR（セクタ）レジスタによって選択されたプロセッサとの間でレジスタの読み出しおよび書き込みに使用するデータを保持します。
DEMCR	32 ビットのデバッグ例外/モニタ制御レジスタ： ベクタキャッチとデバッグモニタの制御を行います。

これらのレジスタは、システムリセットによってはリセットされません。パワーオンリセットによってのみリセットされます。詳細については、Cortex®-M0 TRM を参照してください。

リセット時に停止させるには、以下の手順が必要です。

- デバッグ例外／モニタ制御レジスタのビット 0 (VC_CORRESET) を有効にします。
- デバッグホールド制御／ステータスレジスタのビット 0 (C_DEBUGEN) を有効にします。

32.7 BPU (ブレイクポイントユニット)

Cortex-M0 BPU を実装することで、4 つのブレイクポイントレジスタが提供されます。BPU は、ARMv7-M (Cortex-M3 および Cortex-M4) で使用可能なフラッシュバッチブレイクポイント (FPB) ブロックのサブセットです。

32.7.1 BPU の機能

プロセッサのブレイクポイントは PC ベースのブレイクポイント機能を実装しています。

BPU CoreSight 識別レジスタ、およびそれらのアドレスやアクセスタイプに関する詳細は、ARMv6-M ARM および ARM CoreSight Components Technical Reference Manual を参照してください。

32.8 DWT (データウォッチポイント)

Cortex-M0 DWT を実装することで、2 つのウォッチポイントレジスタが提供されます。

32.8.1 DWT の機能

プロセッサのウォッチポイントは、データアドレスと PC ベースのウォッチポイント機能を果たす PC サンプリングレジスタの両方を実装し、コンパレータのアドレスマスキングをサポートしています (ARMv6-M ARM を参照)。

32.8.2 DWT プログラムカウンタサンプルレジスタ

データウォッチポイントユニットを実装しているプロセッサは、ARMv6-M のオプション品である DWT プログラムカウンタサンプルレジスタ (DWT_PCSR) も実装しています。このレジスタによって、デバッグは、プロセッサを停止せずに定期的に PC をサンプリングすることができます。これにより、粒度の荒いプロファイリングが提供されます。詳細については、ARMv6-M ARM を参照してください。

Cortex-M0 DWT_PCSR は、条件コードに成功した命令と失敗した命令の両方を記録します。

32.9 MCU デバッグコンポーネント (DBGMCU)

MCU デバッグコンポーネントは、デバッグによる以下のサポート機能を支援します。

- 低電力モード
- ブレークポイントにおける、タイマ、ウォッチドッグ、および I2C のクロック制御

32.9.1 低電力モードのデバッグサポート

低電力モードに入るには、WFI または WFE 命令を実行する必要があります。

MCU はいくつかの低電力モードを実装しており、CPU クロックを無効にしたり、CPU の消費電力を低減したりすることができます。

デバッグセッション中には、コアは FCLK や HCLK をオフにすることはできません。デバッグ時のデバッグ接続に必要なため、これらをアクティブな状態に保つ必要があります。MCU は、ユーザが低電力モードでソフトウェアをデバッグするための特殊な手段を備えています。

このため、デバッグホストは、最初にいくつかのデバッグ設定レジスタをセットして、低電力モード動作を変更する必要があります。

- SLEEP モードの場合、FCLK および HCLK は有効のままです。したがって、このモードは標準的なデバッグ機能になんら制限を加えないことが分かります。
- STOP および STANDBY モードの場合、DBG_STOP ビットをデバッグによって事前にセットする必要があります。

これによって、内部 RC オシレータクロックが、STOP モードで FCLK と HCLK を供給できます。

コード例については、「付録」セクション [A.12.2 : 低電力モードの DBG デバッグのコード例](#) を参照してください。

32.9.2 タイマ、ウォッチドッグ、および I²C のデバッグサポート

ブレークポイントにおいては、以下に示すタイマのカウンタやウォッチドッグの動作方法を選択する必要があります。

- ブレークポイントの中でもカウントを継続できます。この動作は、たとえば、PWM がモータを制御しているときに一般的に必要です。
- ブレークポイントの中でカウントを停止できます。この動作はウォッチドッグ用に必要です。

I²C の場合、ブレークポイントにおいては SMBUS タイムアウトをブロックするように選択できます。

32.9.3 デバッグ MCU 設定レジスタ (DBGMCU_CR)

このレジスタを使用して、デバッグ中に MCU を設定できます。次の設定が可能です。

- 低電力モードのサポート

この DBGMCU_CR はアドレス 0x4001 5804 に配置されます。

このレジスタは PORESET によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

デバッグホストがこれらの機能をサポートしない場合でも、ユーザソフトウェアによってこれらのレジスタへの書き込みが可能です。

アドレス : 0x40015804

32 ビットアクセスのみサポートされます。

POR リセット : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_STANDBY	DBG_STOP	Res.
													rw	rw	

ビット 31:3 予約済みであり、リセット値に保持する必要があります。

ビット 2 **DBG_STANDBY** : デバッグ STANDBY モード

0 : (FCLK=オフ、HCLK=オフ) デジタル部全体が電源オフになります。

ソフトウェアから見て、STANDBY モードから抜けることは、リセットベクタのフェッチと同じになります (ただし、いくつかのステータスビットは MCU が STANDBY モードから再開していることを示す)。

1 : (FCLK=オン、HCLK=オン) この場合、デジタル部は電源オフ状態ではなく、FCLK と HCLK は引き続きアクティブ状態の内部 RC オシレータから供給されます。さらに、MCU は STANDBY モード中にシステムリセットを生成するため、STANDBY モードから抜けることはリセットからのフェッチと同じになります。

ビット 1 **DBG_STOP** : デバッグ STOP モード

0 : (FCLK=オフ、HCLK=オフ) STOP モードでは、クロックコントローラがすべてのクロック (HCLK と FCLK を含む) を無効にします。STOP モードから抜けると、クロック設定はリセット後の場合と同じになります (CPU は 8 MHz の内部 RC オシレータ (HSI) から供給されます)。したがって、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります。

1 : (FCLK=オン、HCLK=オン) この場合、STOP モードに入ると、FCLK と HCLK は STOP モードでもアクティブ状態の内部 RC オシレータから供給されます。STOP モードから抜けるとき、ソフトウェアは、クロックコントローラを再プログラムして、PLL やクリスタル発振子などを有効にする必要があります (DBG_STOP=0 の場合と同様です)。

32.9.4 デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ)

DBGMCU_APB1_FZ レジスタは、デバッグにおいて MCU の設定に使用します。以下に示す APB ペリフェラルに関する設定が可能です。

- タイマクロックカウンタの凍結
- I2C SMBUS タイムアウトの凍結
- システムウィンドウ型ウォッチドッグおよび独立型ウォッチドッグのカウンタの凍結サポート

この DBGMCU_APB1_FZ はアドレス 0x4001 5808 に配置されます。

このレジスタは POR によって非同期にリセットされます (システムリセットではリセットされません)。システムリセット中に、デバッグによって書き込みが可能です。

アドレスオフセット : 0x08

32 ビットアクセスのみサポートされます。

パワーオンリセット (POR) : 0x0000 0000 (システムリセットではリセットされません)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	DBG_CAN_STOP	Res.	Res.	Res.	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.
										rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	DBG_TIM14_STOP	Res.	Res.	DBG_TIM7_STOP	DBG_TIM6_STOP	Res.	Res.	DBG_TIM3_STOP	DBG_TIM2_STOP
			rw	rw	rw		rw			rw	rw			rw	rw

ビット 31:26 予約済みであり、リセット値に保持する必要があります。

ビット 25 **DBG_CAN_STOP** : コア停止時は CAN 停止

- 0 : 通常モードと同じ動作です。
- 1 : CAN 受信レジスタは停止状態です。

ビット 22:24 予約済みであり、リセット値に保持する必要があります。

ビット 21 **DBG_I2C1_SMBUS_TIMEOUT** : コア停止時に SMBUS タイムアウトモードは停止

- 0 : 通常モードと同じ動作です。
- 1 : SMBUS タイムアウトは凍結されます。

ビット 20:13 予約済みであり、リセット値に保持する必要があります。

ビット 12 **DBG_IWDG_STOP** : コア停止時にデバッグ独立型ウォッチドッグは停止

- 0 : コアが停止しても独立型ウォッチドッグカウンタのクロックは継続されます。
- 1 : コア停止時に独立型ウォッチドッグカウンタのクロックは停止します。

ビット 11 **DBG_WWDG_STOP** : コア停止時にデバッグウィンドウ型ウォッチドッグは停止

- 0 : コアが停止してもウィンドウ型ウォッチドッグカウンタのクロックは継続されます。
- 1 : コア停止時にウィンドウ型ウォッチドッグカウンタのクロックは停止します。

- ビット 10 **DBG_RTC_STOP** : コア停止時にデバッグ RTC は停止
0 : コアが停止しても当該 RTC カウンタのクロックは供給されます。
1 : コア停止時に当該 RTC カウンタのクロックは停止します。
- ビット 9 予約済みであり、リセット値に保持する必要があります。
- ビット 8 **DBG_TIM14_STOP** : コア停止時に TIM14 カウンタは停止
0 : コアが停止しても TIM14 カウンタのクロックは供給されます。
1 : コア停止時に TIM14 カウンタのクロックは停止します。
- ビット 7:6 予約済みであり、リセット値に保持する必要があります。
- ビット 5 **DBG_TIM7_STOP** : コア停止時に TIM7 カウンタは停止
0 : コアが停止しても TIM7 カウンタのクロックは供給されます。
1 : コア停止時に TIM7 カウンタのクロックは停止します。
- ビット 4 **DBG_TIM6_STOP** : コア停止時に TIM6 カウンタは停止
0 : コアが停止しても TIM6 カウンタのクロックは供給されます。
1 : コア停止時に TIM6 カウンタのクロックは停止します。
- ビット 3:2 予約済みであり、リセット値に保持する必要があります。
- ビット 1 **DBG_TIM3_STOP** : コア停止時に TIM3 カウンタは停止
0 : コアが停止しても TIM3 カウンタのクロックは供給されます。
1 : コア停止時に TIM3 カウンタのクロックは停止します。
- ビット 0 **DBG_TIM2_STOP** : コア停止時に TIM2 カウンタは停止
0 : コアが停止しても TIM2 カウンタのクロックは供給されます。
1 : コア停止時に TIM2 カウンタのクロックは停止します。

32.9.5 デバッグ MCU APB2 凍結レジスタ (DBGMCU_APB2_FZ)

DBGMCU_APB2_FZ レジスタは、デバッグにおいて MCU の設定に使用します。以下に示す APB ペリフェラルに関する設定が可能です。

- タイマクロックカウンタの凍結

このレジスタはアドレス 0x4001580C に配置されます。

このレジスタは POR によって非同期にリセットされます（システムリセットではリセットされません）。システムリセット中に、デバッグによって書き込みが可能です。

アドレスオフセット：0x0C

32 ビットアクセスのみサポートされます。

POR：0x0000 0000（システムリセットではリセットされません）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw											

ビット 31:19 予約済みであり、リセット値に保持する必要があります。

ビット 18 **DBG_TIM17_STOP**：コア停止時に TIM17 カウンタは停止

- 0：コアが停止しても TIM17 カウンタのクロックは供給されます。
- 1：コア停止時に TIM17 カウンタのクロックは停止します。

ビット 17 **DBG_TIM16_STOP**：コア停止時に TIM16 カウンタは停止

- 0：コアが停止しても TIM16 カウンタのクロックは供給されます。
- 1：コア停止時に TIM16 カウンタのクロックは停止します。

ビット 16 **DBG_TIM15_STOP**：コア停止時に TIM15 カウンタは停止

- 0：コアが停止しても TIM15 カウンタのクロックは供給されます。
- 1：コア停止時に TIM15 カウンタのクロックは停止します。

ビット 15:12 予約済みであり、リセット値に保持する必要があります。

ビット 11 **DBG_TIM1_STOP**：コア停止時に TIM1 カウンタは停止

- 0：コアが停止しても TIM 1 カウンタのクロックは供給されます。
- 1：コア停止時に TIM 1 カウンタのクロックは停止します。

ビット 0:10 予約済みであり、リセット値に保持する必要があります。

32.9.6 DBG レジスタマップ

次の表にデバッグレジスタの一覧を示します。

表 141. DBG レジスタマップとリセット値

アド レス	レジスタ	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40015800	DBGMCU_IDCODE	REV_ID																Res.	Res.	Res.	Res.	DEV_ID											
	リセット値 ⁽¹⁾	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	
0x40015804	DBGMCU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_STANDBY	DBG_STOP	
	リセット値																													0	0		
0x40015808	DBGMCU_APB1_FZ	Res.	Res.	Res.	Res.	Res.	Res.	DBG_CAN_STOP	Res.	Res.	Res.	DBG_I2C1_SMBUS_TIMEOUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	DBG_TIM14_STOP	Res.	Res.	DBG_TIM7_STOP	DBG_TIM6_STOP	Res.	DBG_TIM3_STOP	DBG_TIM2_STOP	
	リセット値						0					0									0	0	0		0			0	0			0	0
0x4001580C	DBGMCU_APB2_FZ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP	Res.	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	リセット値													0	0	0					0												

1. リセット値は製品によって異なります。詳細については、[セクション 32.4.1 : MCU デバイス ID コード](#)を参照してください。

33 デバイス電子署名

デバイス電子署名は、フラッシュメモリモジュールのシステムメモリ領域に格納され、デバッグインタフェースまたは CPU を使用して読み出すことができます。電子署名に含まれる出荷時にプログラムされた識別データおよび較正データを使用すれば、ユーザファームウェアやその他の外部デバイスは、STM32F0xx マイクロコントローラの特性に自動的に整合させることができます。

33.1 ユニークデバイス ID レジスタ（96 ビット）

このユニークデバイス識別子は、以下の用途に最適です。

- シリアル番号（例：USB 文字列シリアル番号やその他のエンドアプリケーション）として使用
- 内部フラッシュメモリをプログラムする前に、このユニーク ID をソフトウェア暗号プリミティブやプロトコルと組み合わせて使用する際に、フラッシュメモリ内のコードのセキュリティを高めるためのセキュリティキーの一部として使用
- セキュアなブートプロセスなどの起動に使用

96 ビットのユニークデバイス識別子は、状況やデバイスの違いとは無関係にユニークなリファレンス番号を提供します。ユーザは、これらのビットを変更できません。

ベースアドレス：0x1FFF F7AC

アドレスオフセット：0x00

読み出し専用 = 0xXXXX XXXX（X は出荷時にプログラムされます）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **UID[31:0]**：BCD 形式で表されたウェーハ上の X および Y 座標



アドレスオフセット : 0x04

読み出し専用 = 0xxxxx xxxx (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:8 **UID[63:40]** : LOT_NUM[23:0]
ロット番号 (ASCII で符号化)
ビット 7:0 **UID[39:32]** : WAF_NUM[7:0]
ウェーハ番号 (8 ビットの符号なし番号)

アドレスオフセット : 0x08

読み出し専用 = 0xxxxx xxxx (X は出荷時にプログラムされます)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 31:0 **UID[95:64]** : LOT_NUM[55:24]
ロット番号 (ASCII で符号化)

33.2 メモリサイズデータレジスタ

33.2.1 フラッシュサイズデータレジスタ

ベースアドレス : 0x1FFF F7CC

アドレスオフセット : 0x00

読み出し専用 = 0xxxxx (X は出荷時にプログラムされます)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLASH_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

ビット 15:0 **FLASH_SIZE[15:0]** : フラッシュメモリサイズ
このビットフィールドは、デバイスのフラッシュメモリサイズを KB 単位で示します。
たとえば、0x040 は 64 KB に対応します。

付録 A コード例

A.1 概要

この付録は、このリファレンスマニュアルで説明しているシーケンスのコード例を示します。

これらのコード例は、www.st.com から入手可能な STM32F0xx スニペットファームウェアパッケージ **STM32SnippetsF0** から抽出したものです。

これらのコード例では、CMSIS ヘッダファイル (stm32f0xx.h) のペリフェラルビットやレジスタの説明を使用しています。

特定のレジスタを変更した場合は、「//」で始まるコードのコメントを解除してください。

A.2 フラッシュ操作のコード例

A.2.1 フラッシュメモリアンロックシーケンスコード

```
/* (1) すべての操作が終了するまで待ちます */
/* (2) フラッシュがロックされていないことをチェックします */
/* (3) アンロックシーケンスを実行します */
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (1) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
if ((FLASH->CR & FLASH_CR_LOCK) != 0) /* (2) */
{
    FLASH->KEYR = FLASH_FKEY1; /* (3) */
    FLASH->KEYR = FLASH_FKEY2;
}
```

A.2.2 メインフラッシュプログラミングシーケンスのコード例

```
/* (1) プログラミングを有効にするために、FLASH_CR レジスタの PG ビットをセットします */
/* (2) 任意のアドレスでデータ書き込み（ハーフワード）を実施します */
/* (3) FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます */
/* (4) FLASH_SR レジスタの EOP フラグをチェックします */
/* (5) 1 を書き込むことによって、ソフトウェアによってクリアします */
/* (6) プログラミングを無効にするために、PG ビットをリセットします */
FLASH->CR |= FLASH_CR_PG; /* (1) */
*(__IO uint16_t*)(flash_addr) = data; /* (2) */
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (3) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
if ((FLASH->SR & FLASH_SR_EOP) != 0) /* (4) */
{
    FLASH->SR |= FLASH_SR_EOP; /* (5) */
}
else
{
}
```

```
/* エラーケースを管理します */  
}  
FLASH->CR &= ~FLASH_CR_PG; /* (6) */
```

A.2.3 ページ消去シーケンスのコード例

```
/* (1) ページの消去を有効にするために、FLASH_CR レジスタの PER ビットをセットします */  
/* (2) 消去するページを選択するために、FLASH_AR レジスタをプログラムします */  
/* (3) 消去を開始するために、FLASH_CR レジスタの STRT ビットをセットします */  
/* (4) FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます */  
/* (5) FLASH_SR レジスタの EOP フラグをチェックします */  
/* (6) EOP に 1 を書き込むことによって、ソフトウェアによって EOP フラグをクリアします */  
/* (7) ページの消去を無効にするために、PER ビットをリセットします */  
FLASH->CR |= FLASH_CR_PER; /* (1) */  
FLASH->AR = page_addr; /* (2) */  
FLASH->CR |= FLASH_CR_STRT; /* (3) */  
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (4) */  
{  
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */  
}  
if ((FLASH->SR & FLASH_SR_EOP) != 0) /* (5) */  
{  
    FLASH->SR |= FLASH_SR_EOP; /* (6) */  
}  
else  
{  
    /* エラーケースを管理します */  
}  
FLASH->CR &= ~FLASH_CR_PER; /* (7) */
```

A.2.4 全体消去シーケンスのコード例

```
/* (1) 全体消去を有効にするために、FLASH_CR レジスタの MER ビットをセットします */  
/* (2) 消去を開始するために、FLASH_CR レジスタの STRT ビットをセットします */  
/* (3) FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます */  
/* (4) FLASH_SR レジスタの EOP フラグをチェックします */  
/* (5) EOP に 1 を書き込むことによって、ソフトウェアによって EOP フラグをクリアします */  
/* (6) 全体消去を無効にするために、PER ビットをリセットします */  
FLASH->CR |= FLASH_CR_MER; /* (1) */  
FLASH->CR |= FLASH_CR_STRT; /* (2) */  
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (3) */  
{  
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */  
}  
  
if ((FLASH->SR & FLASH_SR_EOP) != 0) /* (4) */  
{  
    FLASH->SR |= FLASH_SR_EOP; /* (5) */
```

```
}  
else  
{  
    /* エラーケースを管理します */  
}  
FLASH->CR &= ~FLASH_CR_MER; /* (6) */
```

A.2.5 オプションバイトのアンロックシーケンスのコード例

```
/* (1) すべての操作が終了するまで待ちます */  
/* (2) フラッシュがロックされていないことをチェックします */  
/* (3) フラッシュのアンロックシーケンスを実行します */  
/* (4) オプションバイトがロックされていないことをチェックします */  
/* (5) オプションバイトのアンロックシーケンスを実行します */  
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (1) */  
{  
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */  
}  
if ((FLASH->CR & FLASH_CR_LOCK) != 0) /* (2) */  
{  
    FLASH->KEYR = FLASH_FKEY1; /* (3) */  
    FLASH->KEYR = FLASH_FKEY2;  
}  
if ((FLASH->CR & FLASH_CR_OPTWRE) == 0) /* (4) */  
{  
    FLASH->OPTKEYR = FLASH_OPTKEY1; /* (5) */  
    FLASH->OPTKEYR = FLASH_OPTKEY2;  
}
```

A.2.6 オプションバイトのプログラミングシーケンスのコード例

```
/* (1) プログラミングを有効にするために、FLASH_CR レジスタの PG ビットをセットします */  
/* (2) データ書き込みを実施します */  
/* (3) FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます */  
/* (4) FLASH_SR レジスタの EOP フラグをチェックします */  
/* (5) EOP に 1 を書き込むことによって、ソフトウェアによって EOP フラグをクリアします */  
/* (6) プログラミングを無効にするために、PG ビットをリセットします */  
FLASH->CR |= FLASH_CR_OPTPG; /* (1) */  
*opt_addr = data; /* (2) */  
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (3) */  
{  
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */  
}  
if ((FLASH->SR & FLASH_SR_EOP) != 0) /* (4) */  
{  
    FLASH->SR |= FLASH_SR_EOP; /* (5) */  
}  
else
```

```
{  
    /* エラーケースを管理します */  
}  
FLASH->CR &= ~FLASH_CR_OPTPG; /* (6) */
```

A.2.7 オプションバイトの消去シーケンスのコード例

```
/* (1) オプションバイトの消去を有効にするために、FLASH_CR レジスタの OPTER ビットをセットします */  
/* (2) 消去を開始するために、FLASH_CR レジスタの STRT ビットをセットします */  
/* (3) FLASH_SR レジスタの BSY ビットがリセットされるまで待ちます */  
/* (4) FLASH_SR レジスタの EOP フラグをチェックします */  
/* (5) EOP に 1 を書き込むことによって、ソフトウェアによって EOP フラグをクリアします */  
/* (6) ページの消去を無効にするために、PER ビットをリセットします */  
FLASH->CR |= FLASH_CR_OPTER; /* (1) */  
FLASH->CR |= FLASH_CR_STRT; /* (2) */  
while ((FLASH->SR & FLASH_SR_BSY) != 0) /* (3) */  
{  
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */  
}  
if ((FLASH->SR & FLASH_SR_EOP) != 0) /* (4) */  
{  
    FLASH->SR |= FLASH_SR_EOP; /* (5) */  
}  
else  
{  
    /* エラーケースを管理します */  
}  
FLASH->CR &= ~FLASH_CR_OPTER; /* (6) */
```

A.3 クロックコントローラ

A.3.1 HSE 起動シーケンスのコード例

```
/**
 * 説明：この機能では、HSE レディ割り込みを有効にし、HSE を外部クロックとして起動します。
 */
__INLINE void StartHSE(void)
{
    /* RCC 用に NVIC を設定します */
    /* (1) RCC の割り込みを有効にします */
    /* (2) RCC の優先順位をセットします */
    NVIC_EnableIRQ(RCC_CR_S_IRQn); /* (1) */
    NVIC_SetPriority(RCC_CR_S_IRQn, 0); /* (2) */

    /* (1) HSE レディの割り込みを有効にします */
    /* (2) CSS を有効にします
        HSE を有効にし、オシレータではなく外部クロックを使用するよう HSEBYP をセットします
        HSE を有効にします */
    /* 注：クロックは RCC_CR_S_IRQHandler ISR の HSE に切り替えられます */
    RCC->CIR |= RCC_CR_HSERDYIE; /* (1) */
    RCC->CR |= RCC_CR_CSSON | RCC_CR_HSEBYP | RCC_CR_HSEON; /* (2) */
}

/**
 * 説明：この機能では、RCC 割り込みリクエストを処理し、システムクロックを HSE に切り替えます。
 */
void RCC_CR_S_IRQHandler(void)
{
    /* (1) フラグ HSE レディをチェックします */
    /* (2) フラグ HSE レディをクリアします */
    /* (3) システムクロックを HSE に切り替えます */

    if ((RCC->CIR & RCC_CR_HSERDYF) != 0) /* (1) */
    {
        RCC->CIR |= RCC_CR_HSERDYC; /* (2) */
        RCC->CFGR = ((RCC->CFGR & (~RCC_CFGR_SW)) | RCC_CFGR_SW_0); /* (3) */
    }
    else
    {
        /* エラーを報告します */
    }
}
```


A.3.2 PLL 設定変更のコード例

```
/* (1) PLL がシステムクロックとして使用されているかどうかをテストします */
/* (2) HSI をシステムクロックとして選択します */
/* (3) HSI が切り替わるまで待ちます */
/* (4) PLL を無効にします */
/* (5) PLLRDY がクリアされるまで待ちます */
/* (6) PLL マルチプライヤを 6 にセットします */
/* (7) PLL を有効にします */
/* (8) PLLRDY がセットされるまで待ちます */
/* (9) PLL をシステムクロックとして選択します */
/* (10) PLL がオンになるまで待ちます */
if ((RCC->CFGR & RCC_CFGR_SWS) == RCC_CFGR_SWS_PLL) /* (1) */
{
    RCC->CFGR &= (uint32_t) (~RCC_CFGR_SW); /* (2) */
    while ((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_HSI) /* (3) */
    {
        /* 実装を安定させるため、ここにタイムアウト管理を追加します */
    }
}
RCC->CR &= (uint32_t) (~RCC_CR_PLLON); /* (4) */
while((RCC->CR & RCC_CR_PLLRDY) != 0) /* (5) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
RCC->CFGR = RCC->CFGR & (~RCC_CFGR_PLLMUL) | (RCC_CFGR_PLLMUL6); /* (6) */
RCC->CR |= RCC_CR_PLLON; /* (7) */
while((RCC->CR & RCC_CR_PLLRDY) == 0) /* (8) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
RCC->CFGR |= (uint32_t) (RCC_CFGR_SW_PLL); /* (9) */
while ((RCC->CFGR & RCC_CFGR_SWS) != RCC_CFGR_SWS_PLL) /* (10) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
```

A.3.3 MCO 選択のコード例

```
/* プリスケーラを使用しないで MCO に出力するシステムクロックを選択します */
RCC->CFGR |= RCC_CFGR_MCO_SYSCCLK;
```

A.3.4 TIM14 によるクロックの測定設定のコード例

```
/**
 * 説明：この機能では、TIM14 を入力キャプチャとして設定し、TIM14 の割り込みを有効にします
 */
__INLINE void ConfigureTIM14asInputCapture(void)
{
    /* (1) タイマ 14 のペリフェラルクロックを有効にします */
    /* (2) アクティブ入力 TI1 を選択し、入力フィルタとプリスケアラをプログラムします */
    /* (3) キャプチャ／比較の割り込みを有効にします */
    RCC->APB1ENR |= RCC_APB1ENR_TIM14EN; /* (1) */
    TIM14->CCMR1 |= TIM_CCMR1_IC1F_0 | TIM_CCMR1_IC1F_1 \
                  | TIM_CCMR1_CC1S_0 | TIM_CCMR1_IC1PSC_1; /* (2) */
    TIM14->DIER |= TIM_DIER_CC1IE; /* (3) */

    /* TIM14 用に NVIC を設定します */
    /* (4) TIM14 の割り込みを有効にします */
    /* (5) TIM14 の優先順位をセットします */
    NVIC_EnableIRQ(TIM14_IRQn); /* (4) */
    NVIC_SetPriority(TIM14_IRQn, 0); /* (5) */

    /* (6) TI1 の入力として HSE/32 を選択します */
    /* (7) カウンタを有効にします */
    /* (8) キャプチャを有効にします */
    TIM14->OR |= TIM14_OR_TI1_RMP_1; /* (6) */
    TIM14->CR1 |= TIM_CR1_CEN; /* (7) */
    TIM14->CCER |= TIM_CCER_CC1E; /* (8) */
}
```

注： 測定は、TIM14 割り込みサブルーチンで行われます

A.4 GPIO

A.4.1 ロックシーケンスのコード例

```
/**
 * 説明：この機能では、ポート A 設定のターゲットピンをロックします
 *       この機能は、簡単にポート B をロックするように変更できます。
 * パラメータ：ロックには、ロックするポートピンマスクが含まれます。
 */
void LockGPIOA(uint16_t lock)
{
    /* (1) LCKK ビットに 1 を書き込み、ロックするピンのビットをセットします */
    /* (2) LCKK ビットに 0 を書き込み、ロックするピンのビットをセットします */
    /* (3) LCKK ビットに 1 を書き込み、ロックするピンのビットをセットします */
    /* (4) ロックレジスタを読み出します */
    /* (5) ロックレジスタをチェックします (オプション) */
    GPIOA->LCKR = GPIO_LCKR_LCKK + lock; /* (1) */
    GPIOA->LCKR = lock; /* (2) */
    GPIOA->LCKR = GPIO_LCKR_LCKK + lock; /* (3) */
    GPIOA->LCKR; /* (4) */
    if ((GPIOA->LCKR & GPIO_LCKR_LCKK) == 0) /* (5) */
    {
        /* エラーを管理します */
    }
}
```

A.4.2 オルタナート機能の選択シーケンスのコード例

```
/* このシーケンスでは GPIOA5、8、および 9 に AF2 を選択します。すべての GPIOA リファレンスを
別の GPIO ポートに変更することで、別のポートに簡単に応用できます。最後の 2 つのコード行の
ターゲットオルタナート機能で各ピンの 0x02 を置き換えると、オルタナート機能番号を変更できます。*/
/* (1) GPIOA のペリフェラルクロックを有効にします */
/* (2) GPIOA ピン 5、8、および 9 でオルタナート機能モードを選択します */
/* (3) TIM2_CH1 について、AFRL の PA5 で AF2 を選択します */
/* (4) TIM1_CH1 および TIM1_CH2 について、AFRH の PA8 および PA9 で AF2 を選択します */
RCC->AHBENR |= RCC_AHBENR_GPIOAEN; /* (1) */
GPIOA->MODER = (GPIOA->MODER & ~(GPIO_MODER_MODER5 | GPIO_MODER_MODER8
                                | GPIO_MODER_MODER9)) | GPIO_MODER_MODER5_1
               | GPIO_MODER_MODER8_1 | GPIO_MODER_MODER9_1; /* (2) */
GPIOA->AFR[0] |= 0x02 << (5 * 4); /* (3) */
GPIOA->AFR[1] |= 0x02 | (0x02 << ((9 - 8) * 4)); /* (4) */
```

A.4.3 アナログ GPIO 設定のコード例

```
/* (1) GPIOA、GPIOB、および GPIOC のペリフェラルクロックを有効にします */
/* (2) PA1 のアナログモードを選択します */
/* (3) PB1 のアナログモードを選択します */
/* (4) PC0 のアナログモードを選択します */
RCC->AHBENR |= RCC_AHBENR_GPIOAEN | RCC_AHBENR_GPIOBEN
              | RCC_AHBENR_GPIOCEN; /* (1) */
GPIOA->MODER |= GPIO_MODER_MODER1; /* (2) */
GPIOB->MODER |= GPIO_MODER_MODER1; /* (3) */
GPIOC->MODER |= GPIO_MODER_MODER0; /* (4) */
```

A.5 DMA

A.5.1 DMA チャネル設定シーケンスのコード例

```
/* 以下は ADC の例です。ペリフェラルに関連付けられたチャネルを持つ DMA 転送をサポートする任意の
   ペリフェラルに簡単にポートすることができます。これは、データシートで確認する必要があります。*/
/* (1) DMA のペリフェラルクロックを有効にします */
/* (2) ADC の DMA 転送を有効にします */
/* (3) ペリフェラルデータレジスタアドレスを設定します */
/* (4) メモリアドレスを設定します */
/* (5) チャネル 1 で行う DMA 転送回数を設定します */
/* (6) インクリメント、サイズ、および割り込みを設定します */
/* (7) DMA チャネル 1 を有効にします */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_DMAEN; /* (2) */
DMA1_Channel1->CPAR = (uint32_t) (&(ADC1->DR)); /* (3) */
DMA1_Channel1->CMAR = (uint32_t) (ADC_array); /* (4) */
DMA1_Channel1->CNDTR = 3; /* (5) */
DMA1_Channel1->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
                    | DMA_CCR_TEIE | DMA_CCR_TCIE ; /* (6) */
DMA1_Channel1->CCR |= DMA_CCR_EN; /* (7) */
/* DMA 用に NVIC を設定します */
/* (1) DMA チャネル 1 の割り込みを有効にします */
/* (2) DMA チャネル 1 の優先順位をセットします */
NVIC_EnableIRQ(DMA1_Channel1_IRQn); /* (1) */
NVIC_SetPriority(DMA1_Channel1_IRQn, 0); /* (2) */
```

A.6 割り込みとイベント

A.6.1 NVIC の初期化の例

```
/* (1) ADC の割り込みを有効にします */  
/* (2) ADC の優先順位を 2 にセットします */  
NVIC_EnableIRQ(ADC1_COMP_IRQn); /* (1) */  
NVIC_SetPriority(ADC1_COMP_IRQn,2); /* (2) */
```

A.6.2 外部割り込み選択のコード例

```
/* (1) GPIOA のペリフェラルクロックを有効にします */  
/* (2) EXTI0 に 0000 (リセット値) を書き込むことによって、ピン 0 の外部割り込みにポート A を  
    選択します */  
/* (3) EXTI_IMR レジスタの対応するマスクビットを設定します */  
/* (4) 立ち上がりエッジで割り込みラインのトリガ選択ビットを設定します */  
/* (5) 立ち下がりエッジで割り込みラインのトリガ選択ビットを設定します */  
RCC->AHBENR |= RCC_AHBENR_GPIOAEN; /* (1) */  
//SYSCFG->EXTICR[1] &= (uint16_t)~SYSCFG_EXTICR1_EXTI0_PA; /* (2) */  
EXTI->IMR = 0x0001; /* (3) */  
EXTI->RTSR = 0x0001; /* (4) */  
EXTI->FTSR = 0x0001; /* (5) */  
/* 外部割り込み用に NVIC を設定します */  
/* (1) EXTI0_1 の割り込みを有効にします */  
/* (2) EXTI0_1 の優先順位をセットします */  
NVIC_EnableIRQ(EXTI0_1_IRQn); /* (1) */  
NVIC_SetPriority(EXTI0_1_IRQn,0); /* (2) */
```

A.7 ADC

A.7.1 ADC 較正のコード例

```
/* (1) ADEN = 0 であることを確認します */
/* (2) ADEN をクリアします */
/* (3) ADCAL を設定して較正を始めます */
/* (4) ADCAL=0 になるまで待ちます */
if ((ADC1->CR & ADC_CR_ADEN) != 0) /* (1) */
{
    ADC1->CR &= (uint32_t)(~ADC_CR_ADEN); /* (2) */
}
ADC1->CR |= ADC_CR_ADCAL; /* (3) */
while ((ADC1->CR & ADC_CR_ADCAL) != 0) /* (4) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
```

A.7.2 ADC 有効化シーケンスのコード例

```
/* (1) ADC を有効にします */
/* (2) ADC レディになるまで待ちます */
ADC1->CR |= ADC_CR_ADEN; /* (1) */
while ((ADC1->ISR & ADC_ISR_ADRDY) == 0) /* (2) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
```

A.7.3 ADC 無効化シーケンスのコード例

```
/* (1) 実行中の変換がないことを確認します */
/* (2) 実行中の変換を停止します */
/* (3) ADSTP がハードウェアによってリセットされるまで（つまり変換が停止するまで）待ちます */
/* (4) ADC を無効にします */
/* (5) ADC が完全に無効になるまで待ちます */
if ((ADC1->CR & ADC_CR_ADSTART) != 0) /* (1) */
{
    ADC1->CR |= ADC_CR_ADSTP; /* (2) */
}
while ((ADC1->CR & ADC_CR_ADSTP) != 0) /* (3) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
ADC1->CR |= ADC_CR_ADDIS; /* (4) */
while ((ADC1->CR & ADC_CR_ADEN) != 0) /* (5) */
{
}
```

```
/* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
```

A.7.4 ADC クロック選択のコード例

```
/* このコードは、クロックソースとして HSI14 を選択します。*/
/* (1) ADC のペリフェラルクロックを有効にします */
/* (2) HSI14 RC オシレータを起動します */
/* (3) HSI14 がレディになるまで待ちます */
/* (4) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
RCC->APB2ENR |= RCC_APB2ENR_ADC1EN; /* (1) */
RCC->CR2 |= RCC_CR2_HSI14ON; /* (2) */
while ((RCC->CR2 & RCC_CR2_HSI14RDY) == 0) /* (3) */
{
    /* 実装を安定させるため、ここにタイムアウト管理を追加します */
}
//ADC1->CFGR2 &= (~ADC_CFGR2_CKMODE); /* (4) */
```

A.7.5 シングル変換シーケンスのコード例 - ソフトウェアトリガ

```
/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) VRefInt に CHSEL0、CHSEL9、CHSEL10、および CHSEL17 を選択します */
/* (3) サンプリングモード 111 (つまり 239.5 ADC クロック) が 17.1 us より大きくなるように
    選択します */
/* (4) VREFINT をウェイクアップします (VBAT、温度センサ、および VRefInt の場合のみ) */
//ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CHSELR = ADC_CHSELR_CHSEL0 | ADC_CHSELR_CHSEL9
               | ADC_CHSELR_CHSEL10 | ADC_CHSELR_CHSEL17; /* (2) */
ADC1->SMPR |= ADC_SMPR_SMP_0 | ADC_SMPR_SMP_1 | ADC_SMPR_SMP_2; /* (3) */
ADC->CCR |= ADC_CCR_VREFEN; /* (4) */
while (1)
{
    /* AD 変換を実行します */
    ADC1->CR |= ADC_CR_ADSTART; /* AD 変換を開始します */
    for (i=0; i < 4; i++)
    {
        while ((ADC1->ISR & ADC_ISR_EOC) == 0) /* 変換の終了を待ちます */
        {
            /* 実装を安定させるため、ここにタイムアウト管理を追加します */
        }
        ADC_Result[i] = ADC1->DR; /* ADC の変換結果を格納します */
    }
    ADC1->CFGR1 ^= ADC_CFGR1_SCANDIR; /* スキャン方向を反転します */
}
```


A.7.6 連続変換シーケンスのコード例 - ソフトウェアトリガ

```
/* このコード例は、連続モードおよび後方スキャンで AD 変換を設定します。
   割り込みも有効にします。*/

/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) 連続モードとスキャン方向を選択します */
/* (3) CHSEL1、CHSEL9、CHSEL10、および CHSEL17 を選択します */
/* (4) サンプリングモード 111 (つまり 239.5 ADC クロック) が 17.1 us より大きくなるように
   選択します */
/* (5) EOC、EOSEQ、およびオーバーランの割り込みを有効にします */
/* (6) VREFINT をウェイクアップします (VBAT、温度センサ、および VRefInt の場合のみ) */
//ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_CONT | ADC_CFGR1_SCANDIR; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL1 | ADC_CHSELR_CHSEL9
               | ADC_CHSELR_CHSEL10 | ADC_CHSELR_CHSEL17; /* (3) */
ADC1->SMPR |= ADC_SMPR_SMP_0 | ADC_SMPR_SMP_1 | ADC_SMPR_SMP_2; /* (4) */
ADC1->IER = ADC_IER_EOCIE | ADC_IER_EOSEQIE | ADC_IER_OVR1E; /* (5) */
ADC->CCR |= ADC_CCR_VREFEN; /* (6) */

/* ADC 用に NVIC を設定します */
/* (7) ADC の割り込みを有効にします */
/* (8) ADC の優先順位をセットします */
NVIC_EnableIRQ(ADC1_COMP_IRQn); /* (7) */
NVIC_SetPriority(ADC1_COMP_IRQn, 0); /* (8) */
```

A.7.7 シングル変換シーケンスのコード例 - ハードウェアトリガ

```
/* ADC とクロックを事前に有効にして、ADC を設定します。
   */
/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) 立ち下がりエッジと TIM15_TRGO で外部トリガを選択します */

/* (3) CHSEL0、1、2、および 3 を選択します */
//ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_EXTEN_0 | ADC_CFGR1_EXTSEL_2; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL0 | ADC_CHSELR_CHSEL1
               | ADC_CHSELR_CHSEL2 | ADC_CHSELR_CHSEL3; /* (3) */
```

A.7.8 連続変換シーケンスのコード例 - ハードウェアトリガ

```
/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) TIM15_TRGO (EXTSEL = 100)、立ち下がりエッジ (EXTEN = 10)、連続モード (CONT = 1) で
   外部トリガを選択します */
/* (3) CHSEL0/1/2/3 を選択します */
/* (4) EOC、EOSEQ、およびオーバーランの割り込みを有効にします */
//ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_EXTEN_1 | ADC_CFGR1_EXTSEL_2
               | ADC_CFGR1_CONT; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL0 | ADC_CHSELR_CHSEL1
```

```

        | ADC_CHSELR_CHSEL2 | ADC_CHSELR_CHSEL3; /* (3) */
ADC1->IER = ADC_IER_EOCIE | ADC_IER_EOSEQIE | ADC_IER_OVR1E; /* (4) */

/* ADC 用に NVIC を設定します */
/* (1) ADC の割り込みを有効にします */
/* (2) ADC の優先順位をセットします */
NVIC_EnableIRQ(ADC1_COMP_IRQn); /* (1) */
NVIC_SetPriority(ADC1_COMP_IRQn, 0); /* (2) */

```

A.7.9 DMA ワンショットモードシーケンスのコード例

```

/* (1) DMA のペリフェラルクロックを有効にします */
/* (2) ADC で DMA 転送を有効にします - ワンショットモードで、DMACFG は 0 に保たれます */
/* (3) ペリフェラルデータレジスタアドレスを設定します */
/* (4) メモリアドレスを設定します */
/* (5) DMA チャンネル 1 で行う DMA 転送回数を設定します */
/* (6) インクリメント、サイズ、および割り込みを設定します */
/* (7) DMA チャンネル 1 を有効にします */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_DMAEN; /* (2) */
DMA1_Channel1->CPAR = (uint32_t) (&(ADC1->DR)); /* (3) */
DMA1_Channel1->CMAR = (uint32_t) (ADC_array); /* (4) */
DMA1_Channel1->CNDTR = NUMBER_OF_ADC_CHANNEL; /* (5) */
DMA1_Channel1->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
                    | DMA_CCR_TEIE | DMA_CCR_TCIE; /* (6) */
DMA1_Channel1->CCR |= DMA_CCR_EN; /* (7) */

```

A.7.10 DMA サーキュラモードシーケンスのコード例

```

/* (1) DMA のペリフェラルクロックを有効にします */
/* (2) ADC とサーキュラモードの DMA 転送を有効にします */
/* (3) ペリフェラルデータレジスタアドレスを設定します */
/* (4) メモリアドレスを設定します */
/* (5) DMA チャンネル 1 で行う DMA 転送回数を設定します */
/* (6) インクリメント、サイズ、割り込み、およびサーキュラモードを設定します */
/* (7) DMA チャンネル 1 を有効にします */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_DMAEN | ADC_CFGR1_DMCFG; /* (2) */
DMA1_Channel1->CPAR = (uint32_t) (&(ADC1->DR)); /* (3) */
DMA1_Channel1->CMAR = (uint32_t) (ADC_array); /* (4) */
DMA1_Channel1->CNDTR = NUMBER_OF_ADC_CHANNEL; /* (5) */
DMA1_Channel1->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
                    | DMA_CCR_TEIE | DMA_CCR_CIRC; /* (6) */
DMA1_Channel1->CCR |= DMA_CCR_EN; /* (7) */

```

A.7.11 ウェイトモードシーケンスのコード例

```
/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) 連続モードとウェイトモードを選択します */
/* (3) CHSEL1/2/3 を選択します */
ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_CONT | ADC_CFGR1_WAIT; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL1 | ADC_CHSELR_CHSEL2
              | ADC_CHSELR_CHSEL3; /* (3) */
ADC1->CR |= ADC_CR_ADSTART; /* AD 変換を開始します */
```

A.7.12 オートオフおよびノーウェイトモードシーケンスのコード例

```
/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) TIM15_TRGO、立ち上がりエッジ、オートオフで外部トリガを選択します */

/* (3) CHSEL1/2/3/4 を選択します */
/* (4) EOC、EOSEQ、およびオーバーランの割り込みを有効にします */
ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_EXTEN_1 | ADC_CFGR1_EXTSEL_2
              | ADC_CFGR1_AUTOFF; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL1 | ADC_CHSELR_CHSEL2
              | ADC_CHSELR_CHSEL3 | ADC_CHSELR_CHSEL4; /* (3) */
ADC1->IER = ADC_IER_EOCIE | ADC_IER_EOSEQIE | ADC_IER_OVRIE; /* (4) */
```

A.7.13 オートオフおよびウェイトモードシーケンスのコード例

```
/* (1) CKMODE に 00 (リセット値) を書き込むことによって HSI14 を選択します */
/* (2) TIM15_TRGO、立ち下がりエッジ、連続モード、スキャン方向、オートオフで外部トリガを選択します */
/* (3) CHSEL1、CHSEL9、CHSEL10、および CHSEL17 を選択します */
/* (4) EOC、EOSEQ、およびオーバーランの割り込みを有効にします */
ADC1->CFGR2 &= ~ADC_CFGR2_CKMODE; /* (1) */
ADC1->CFGR1 |= ADC_CFGR1_EXTEN_0 | ADC_CFGR1_EXTSEL_2
              | ADC_CFGR1_SCANDIR | ADC_CFGR1_AUTOFF; /* (2) */
ADC1->CHSELR = ADC_CHSELR_CHSEL1 | ADC_CHSELR_CHSEL2
              | ADC_CHSELR_CHSEL3 | ADC_CHSELR_CHSEL4; /* (3) */
ADC1->IER = ADC_IER_EOCIE | ADC_IER_EOSEQIE | ADC_IER_OVRIE; /* (4) */
```

A.7.14 アナログウォッチドッグのコード例

```
/* (1) 連続モードを選択し、CH17 のみを監視するようにアナログウォッチドッグを設定します */
/* (2) アナログウォッチドッグの範囲を定義します。上限は 16b-MSW、下限は 16b-LSW です */
/* (3) アナログウォッチドッグの割り込みを有効にします */
ADC1->CFGR1 |= ADC_CFGR1_CONT
              | (17 << 26) | ADC_CFGR1_AWDEN | ADC_CFGR1_AWDSGL; /* (1) */
ADC1->TR = (vrefint_high << 16) + vrefint_low; /* (2) */
ADC1->IER = ADC_IER_AWDIE; /* (3) */
```

A.7.15 温度設定のコード例

```
/* (1) 温度センサに CHSEL16 を選択します */
/* (2) サンプリングモード 111 (つまり 239.5 ADC クロック) が 17.1 us より大きくなるように
    選択します */
/* (3) 温度センサをウェイクアップします (VBAT、温度センサ、および VRefInt の場合のみ) */
ADC1->CHSELR = ADC_CHSELR_CHSEL16; /* (1) */
ADC1->SMPR |= ADC_SMPR_SMP_0 | ADC_SMPR_SMP_1 | ADC_SMPR_SMP_2; /* (2) */
ADC->CCR |= ADC_CCR_TSEN; /* (3) */
```

A.7.16 温度計算のコード例

```
/* 温度センサ校正値アドレス */
#define TEMP110_CAL_ADDR ((uint16_t*) ((uint32_t) 0x1FFFF7C2))
#define TEMP30_CAL_ADDR ((uint16_t*) ((uint32_t) 0x1FFFF7B8))
#define VDD_CALIB ((uint16_t) (330))
#define VDD_APPLI ((uint16_t) (300))
int32_t temperature; /* 摂氏単位の温度を含みます */
temperature = (((int32_t) ADC1->DR * VDD_APPLI / VDD_CALIB)
               - (int32_t) *TEMP30_CAL_ADDR);
temperature = temperature * (int32_t) (110 - 30);
temperature = temperature / (int32_t) (*TEMP110_CAL_ADDR
                                       - *TEMP30_CAL_ADDR);
temperature = temperature + 30;
```

A.8 DAC

A.8.1 波形生成なしの独立トリガのコード例

```
/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) DAC ch1 および ch2 の DMA 転送を有効にし、DAC ch1 および ch2 の DMA アンダーランの
    割り込みを有効にし、DAC ch1 および ch2 を有効にし、TSEL1 に 000 を保持して TIM6 を
    トリガとして選択し、TSEL2 に 010 を書き込んで TIM7 をトリガとして選択します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_TSEL2_1 | DAC_CR_DMAUDRIE2 | DAC_CR_DMAEN2
        | DAC_CR_TEN2 | DAC_CR_EN2
        | DAC_CR_DMAUDRIE1 | DAC_CR_DMAEN1 | DAC_CR_BOFF1
        | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* ch1 で DAC 値を初期化します */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* ch2 で DAC 値を初期化します */
```

A.8.2 1 つの LFSR 生成による独立トリガのコード例

```
/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) 511 ビット振幅で WAVEx を 01 に、LFSR マスク振幅 (MAMPx) を 1000 に設定し、
    DAC ch1 および ch2 を有効にし、ch1 および ch2 のバッファを無効にし、TSEL2 に 010 を
    書き込んで TIM7 をトリガとして選択し、TSEL1 に 000 を保持して TIM6 をトリガとして
    選択します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_MAMP1_3
        | DAC_CR_MAMP2_3 | DAC_CR_WAVE2_0
        | DAC_CR_TSEL2_1 | DAC_CR_BOFF2
        | DAC_CR_TEN2 | DAC_CR_EN2
        | DAC_CR_BOFF1 | DAC_CR_TEN1
        | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* DAC 出力値を初期化します */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* DAC 出力値を初期化します */
```

A.8.3 異なる LFSR 生成による独立トリガのコード例

```
/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) 511 ビット振幅で WAVEx を 01 に、LFSR マスク振幅 (MAMPx) を 1000 に、
    LFSR マスク振幅 (MAMP2) を 0111 (つまり、ch2 の 255 ビット振幅) に設定し、
    DAC ch1 および ch2 を有効にし、ch1 および ch2 のバッファを無効にし、
    TSEL2 に 010 を書き込んで TIM7 をトリガとして選択し、TSEL1 に 000 を保持して
    TIM6 をトリガとして選択します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_WAVE2_0 | DAC_CR_MAMP1_3
        | DAC_CR_MAMP2_2 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
        | DAC_CR_TSEL2_1 | DAC_CR_BOFF2 | DAC_CR_TEN2 | DAC_CR_EN2
        | DAC_CR_BOFF1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* DAC 出力値を初期化します */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* DAC 出力値を初期化します */
```

A.8.4 1つの三角波生成による独立トリガのコード例

```

/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) 1023 ビット振幅で WAVEx を 10 に、LFSR マスク振幅 (MAMPx) を 1001 に設定し、
    DAC ch1 および ch2 を有効にし、ch1 および ch2 のバッファを無効にし、
    TSEL2 に 010 を書き込んで TIM7 をトリガとして選択し、TSEL1 に 000 を保持して
    TIM6 をトリガとして選択します */
/* (3) チャンネル 1 の三角波の低値を定義します */
/* (4) チャンネル 2 の三角波の低値を定義します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
          | DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
          | DAC_CR_MAMP2_3 | DAC_CR_MAMP2_0
          | DAC_CR_TSEL2_1 | DAC_CR_BOFF2 | DAC_CR_TEN2 | DAC_CR_EN2
          | DAC_CR_BOFF1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* (3) */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* (4) */

```

A.8.5 異なる三角波生成による独立トリガのコード例

```

/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) 1023 ビット振幅で WAVEx を 10 に、ch1 のマスク振幅 (MAMP1) を 1001 に、
    4095 ビット振幅でch2 のマスク振幅 (MAMP2) を 1011 に設定し、DAC ch1 および
    ch2 を有効にし、ch1 および ch2 のバッファを無効にし、TSEL2 に 010 を書き込んで
    TIM7 をトリガとして選択し、TSEL1 に 000 を保持して TIM6 をトリガとして選択します */
/* (3) チャンネル 1 の三角波の低値を定義します */
/* (4) チャンネル 2 の三角波の低値を定義します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
          | DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
          | DAC_CR_MAMP2_3 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
          | DAC_CR_TSEL2_1 | DAC_CR_BOFF2 | DAC_CR_TEN2 | DAC_CR_EN2
          | DAC_CR_BOFF1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
DAC->DHR12R1 = DAC_OUT1_VALUE; /* (3) */
DAC->DHR12R2 = DAC_OUT2_VALUE; /* (4) */

```

A.8.6 同時ソフトウェア開始のコード例

```

/* 目的の DHR レジスタに、デュアル DAC チャンネルデータをロードします */
DAC->DHR12RD = (uint32_t)((signal1[x] << 16) + signal2[x]);

```

A.8.7 波形生成なしの同時トリガのコード例

```

/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) 両チャンネルについて DAC ch1 の DMA 転送を有効にし、DAC ch1 および ch2 を有効にし、
    TSEL1 および TSEL2 に 010 を書き込んで TIM7 をトリガとして選択します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_TSEL1_1 | DAC_CR_TEN2 | DAC_CR_EN2

```



```
        | DAC_CR_TSEL2_1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */  
/* デュアル DAC 値を初期化します */  
DAC->DHR12RD = (uint32_t)((2048 << 16) + 2048);
```

A.8.8 1つのLFSR生成による同時トリガのコード例

```
/* (1) DAC のペリフェラルクロックを有効にします */  
/* (2) 511 ビット振幅で WAVEx を 01 に、LFSR マスク振幅 (MAMPx) を 1000 に設定し、  
DAC ch1 および ch2 を有効にし、ch1 および ch2 のパッファを無効にし、  
TSEL1 および TSEL2 に 010 を書き込んで TIM7 をトリガとして選択します */  
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */  
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_WAVE2_0  
        | DAC_CR_MAMP1_3 | DAC_CR_MAMP2_3  
        | DAC_CR_TSEL2_1 | DAC_CR_BOFF2  
        | DAC_CR_TEN2 | DAC_CR_EN2  
        | DAC_CR_TSEL1_1 | DAC_CR_BOFF1  
        | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */  
/* デュアルレジスタを初期化します */  
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);
```

A.8.9 異なるLFSR生成による同時トリガのコード例

```
/* (1) DAC のペリフェラルクロックを有効にします */  
/* (2) 511 ビット振幅で WAVEx を 01 に、LFSR マスク振幅 (MAMP1) を 1000 に設定し、  
LFSR マスク振幅 (MAMP2) を 0111 (つまり、ch2 の 255 ビット振幅) に設定し、  
DAC ch1 および ch2 を有効にし、ch1 および ch2 のパッファを無効にし、TSEL1 および  
TSEL2 に 010 を書き込んで TIM7 をトリガとして選択します */  
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */  
DAC->CR |= DAC_CR_WAVE1_0 | DAC_CR_WAVE2_0 | DAC_CR_MAMP1_3  
        | DAC_CR_MAMP2_2 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0  
        | DAC_CR_TSEL2_1 | DAC_CR_BOFF2  
        | DAC_CR_TEN2 | DAC_CR_EN2  
        | DAC_CR_TSEL1_1 | DAC_CR_BOFF1  
        | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */  
/* デュアルレジスタを初期化します */  
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);
```

A.8.10 1つの三角波生成による同時トリガのコード例

```
/* (1) DAC のペリフェラルクロックを有効にします */  
/* (2) 1023 ビット振幅で WAVEx を 10 に、LFSR マスク振幅 (MAMPx) を 1001 に設定し、  
DAC ch1 および ch2 を有効にし、ch1 および ch2 のパッファを無効にし、  
TSEL1 および TSEL2 に 010 を書き込んで TIM7 をトリガとして選択します */  
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */  
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1  
        | DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0  
        | DAC_CR_MAMP2_3 | DAC_CR_MAMP2_0  
        | DAC_CR_TSEL2_1 | DAC_CR_BOFF2  
        | DAC_CR_TEN2 | DAC_CR_EN2  
        | DAC_CR_TSEL1_1 | DAC_CR_BOFF1
```



```

        | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
/* デュアルレジスタを初期化します */
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);

```

A.8.11 異なる三角波生成による同時トリガのコード例

```

/* (1) DAC のペリフェラルクロックを有効にします */
/* (2) WAVEex を 10 に、1023 ビット振幅で ch1 のマスク振幅 (MAMP1) を 1001 に、
    4095 ビット振幅で ch2 のマスク振幅 (MAMP1) を 1011 に設定し、DAC ch1 および
    ch2 を有効にし、TSEL1 および TSEL2 に 010 を書き込んで TIM7 をトリガとして選択します */
RCC->APB1ENR |= RCC_APB1ENR_DACEN; /* (1) */
DAC->CR |= DAC_CR_WAVE1_1 | DAC_CR_WAVE2_1
        | DAC_CR_MAMP1_3 | DAC_CR_MAMP1_0
        | DAC_CR_MAMP2_3 | DAC_CR_MAMP2_1 | DAC_CR_MAMP2_0
        | DAC_CR_TSEL2_1 | DAC_CR_TEN2 | DAC_CR_EN2
        | DAC_CR_TSEL1_1 | DAC_CR_TEN1 | DAC_CR_EN1; /* (2) */
/* デュアルレジスタを初期化します */
DAC->DHR12RD = (uint32_t)((DAC_OUT2_VALUE << 16) + DAC_OUT1_VALUE);

```

A.8.12 DMA 初期化のコード例

```

/* (1) 両チャンネルについて DAC ch1 の DMA 転送を有効にし、DAC の DMA アンダーランの割り込みを
    有効にし、DAC ch1 および ch2 を有効にし、TSEL1 および TSEL2 に 010 を書き込んで
    TIM7 をトリガとして選択します */
DAC->CR |= DAC_CR_TSEL1_1 | DAC_CR_TEN2 | DAC_CR_EN2
        | DAC_CR_TSEL2_1 | DAC_CR_DMAUDRIE1 | DAC_CR_DMAEN1
        | DAC_CR_TEN1 | DAC_CR_EN1; /* (1) */
/* (1) DMA のペリフェラルクロックを有効にします */
/* (2) ペリフェラルデータレジスタアドレスを設定します */
/* (3) メモリアドレスを設定します */
/* (4) チャンネル 3 で行う DMA 転送回数を設定します */
/* (5) インクリメント、サイズ (32 ビット)、割り込み、メモリからペリフェラルへの転送、
    およびサーキュラモードを設定します */
/* (6) DMA チャンネル 3 を有効にします */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
DMA1_Channel3->CPAR = (uint32_t)(&(DAC->DHR12RD)); /* (2) */
DMA1_Channel3->CMAR = (uint32_t)signal_data; /* (3) */
DMA1_Channel3->CNDTR = SIGNAL_ARRAY_SIZE; /* (4) */
DMA1_Channel3->CCR |= DMA_CCR_MINC | DMA_CCR_MSIZE_1 | DMA_CCR_PSIZE_1
        | DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel3->CCR |= DMA_CCR_EN; /* (6) */

```

A.9 タイマ

A.9.1 TI2 の立ち上がりエッジのアップカウンタのコード例

```
/* (1) タイマ 1 のペリフェラルクロックを有効にします */
/* (2) GPIOA のペリフェラルクロックを有効にします */
/* (3) GPIOA ピン 9 のオルタネート機能モード (10) を選択します */
/* (4) GPIOA AFRH レジスタのピン 9 で AF2 を有効にすることによって PA9 の TIM1_CH2 を
    選択します */
RCC->APB2ENR |= RCC_APB2ENR_TIM1EN; /* (1) */
RCC->AHBENR |= RCC_AHBENR_GPIOAEN; /* (2) */
GPIOA->MODER = (GPIOA->MODER & ~(GPIO_MODER_MODER9))
    | (GPIO_MODER_MODER9_1); /* (3) */
GPIOA->AFR[1] |= 0x2 << ((9-8)*4); /* (4) */
/* (1) CC2S = '01' を書き込むことによって、TI2 入力の立ち上がりエッジを検出するようにチャネル 2 を
    設定し、TIMx_CCMR1 レジスタの IC2F[3:0] ビットに書き込むことによって、
    入力フィルタ時間を設定します (フィルタを使用しない場合は、IC2F=0000 にしておきます)。*/
/* (2) TIMx_CCER レジスタで CC2P=0 (リセット値) を書き込むことによって立ち上がりエッジ極性を
    選択します。
    */
/* (3) SMS=111 を書き込むことによって、タイマを外部クロックモード 1 に設定します
    TIMx_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を
    選択します。*/
/* (4) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
TIMx->CCMR1 |= TIM_CCMR1_IC2F_0 | TIM_CCMR1_IC2F_1
    | TIM_CCMR1_CC2S_0; /* (1) */
TIMx->CCER &= (uint16_t) (~TIM_CCER_CC2P); /* (2) */
TIMx->SMCR |= TIM_SMCR_SMS | TIM_SMCR_TS_2 | TIM_SMCR_TS_1; /* (3) */
TIMx->CR1 |= TIM_CR1_CEN; /* (4) */
```

A.9.2 2 回の ETR の立ち上がりエッジごとのアップカウンタのコード例

```
/* (1) タイマ 1 のペリフェラルクロックを有効にします */
/* (2) GPIOA のペリフェラルクロックを有効にします */
/* (3) GPIOA ピン 12 のオルタネート機能モード (10) を選択します */
/* (4) GPIOA AFRH レジスタのピン 12 で AF2 を有効にすることによって PA12 の TIM1_ETR を
    選択します */
RCC->APB2ENR |= RCC_APB2ENR_TIM1EN; /* (1) */
RCC->AHBENR |= RCC_AHBENR_GPIOAEN; /* (2) */
GPIOA->MODER = (GPIOA->MODER & ~(GPIO_MODER_MODER12))
    | (GPIO_MODER_MODER12_1); /* (3) */
GPIOA->AFR[1] |= 0x2 << ((12-8)*4); /* (4) */
/* (1) この例ではフィルタは不要なので、TIMx_SMCR レジスタの ETF[3:0] に 0000 を書き込みます。
    リセット値を保持します。
    TIMx_SMCR レジスタに ETPS[1:0]=01 を書き込むことによって、プリスケアラを設定します。
    TIMx_SMCR レジスタに ETP=0 を書き込むことによって、ETR ピンの立ち上がりエッジ検出を
    選択します。
    リセット値を保持します。
    TIMx_SMCR レジスタに ECE=1 を書き込むことによって、外部クロックモード 2 を有効にします。
    */
```

```
/* (2) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
TIMx->SMCR |= TIM_SMCER_ETPS_0 | TIM_SMCER_ECE; /* (1) */
TIMx->CR1 |= TIM_CR1_CEN; /* (2) */
```

A.9.3 入力キャプチャ設定のコード例

```
/* (1) アクティブ入力 TI1 (CC1S = 01) を選択し、8 クロックサイクルの入力フィルタ
   (IC1F = 0011) をプログラムし、CC1 の立ち上がりエッジ (CC1P = 0、リセット値) と
   それぞれの有効な遷移のプリスケアラ (IC1PS = 00、リセット値) を選択します。
/* (2) CC1E をセットすることによって、キャプチャを有効にします */
/* (3) キャプチャ/比較の割り込みを有効にします */
/* (4) カウンタを有効にします */
TIMx->CCMR1 |= TIM_CCMR1_CC1S_0
               | TIM_CCMR1_IC1F_0 | TIM_CCMR1_IC1F_1; /* (1) */
TIMx->CCER |= TIM_CCER_CC1E; /* (2) */
TIMx->DIER |= TIM_DIER_CC1IE; /* (3) */
TIMx->CR1 |= TIM_CR1_CEN; /* (4) */
```

A.9.4 入力キャプチャのデータ管理のコード例

このコードはタイマ割り込みサブルーチンに挿入する必要があります。

```
if ((TIMx->SR & TIM_SR_CC1IF) != 0)
{
    if ((TIMx->SR & TIM_SR_CC1OF) != 0) /* オーバーフローをチェックします */
    {
        /* オーバーフローエラー管理 */
        gap = 0; /* 計算ラップを再初期化します */
        TIMx->SR &= ~(TIM_SR_CC1OF | TIM_SR_CC1IF); /* フラグをクリアします */
        return;
    }
    if (gap == 0) /* 最初の立ち上がりエッジかどうかをテストします */
    {
        counter0 = TIMx->CCR1; /* CC1ICF をクリアするキャプチャカウンタを読み出します */

        gap = 1; /* 最初の立ち上がりエッジが検出されたことを示します */
    }
    else
    {
        counter1 = TIMx->CCR1; /* CC1ICF をクリアするキャプチャカウンタを読み出します */

        if (counter1 > counter0) /* キャプチャカウンタオーバーフローをチェックします */
        {
            Counter = counter1 - counter0;
        }
        else
        {
            Counter = counter1 + 0xFFFF - counter0 + 1;
        }
        counter0 = counter1;
    }
}
```

```
else
{
    /* 予期しない割り込み */
    /* アプリケーションの安定性のためにエラーを管理します */
}
```

注： このコードでは、単一のカウンタオーバーフローのみを管理します。多数のカウンタオーバーフローを管理するには、更新割り込みを有効にし (UIE = 1)、適切に管理する必要があります。

A.9.5 PWM 入力設定のコード例

```
/* (1) TIMx_CCR1 (CC1S = 01) にアクティブ入力 TI1を選択し、TIMx_CCR2 (CC2S = 10) に
    アクティブ入力 TI1 を選択します */
/* (2) TI1FP1 (TS = 101) を有効なトリガ入力として選択し、スレーブモードをリセットモード
    (SMS = 100) に設定します */
/* (3) CC1E と CC2E をセットすることによってキャプチャを有効にし、CC1 と CC1N の立ち上がり
    エッジを選択し (CC1P = 0 および CC1NP = 0、リセット値)、CC2 の立ち下がりエッジを
    選択します (CC2P = 1)。
```

```
    */
/* (4) キャプチャ/比較 1 の割り込みを有効にします */
/* (5) カウンタを有効にします */
TIMx->CCMR1 |= TIM_CCMR1_CC1S_0 | TIM_CCMR1_CC2S_1; /* (1) */
TIMx->SMCR |= TIM_SMCR_TS_2 | TIM_SMCR_TS_0
              | TIM_SMCR_SMS_2; /* (2) */
TIMx->CCER |= TIM_CCER_CC1E | TIM_CCER_CC2E | TIM_CCER_CC2P; /* (3) */
TIMx->DIER |= TIM_DIER_CC1IE; /* (4) */
TIMx->CR1 |= TIM_CR1_CEN; /* (5) */
```

A.9.6 DMA 設定による PWM 入力のコード例

```
/* (1) DMA のペリフェラルクロックを有効にします */
/* (2) DMA チャンネル x のペリフェラルデータレジスタアドレスを設定します */
/* (3) DMA チャンネル x のメモリアドレスを設定します */
/* (4) DMA チャンネル x で行う DMA 転送回数を設定します */
/* (5) インクリメント無し (リセット値)、サイズ (16 ビット)、割り込み、ペリフェラルからメモリへの
    転送、および DMA チャンネル x のサーキュラモードを設定します */
/* (6) DMA チャンネル x を有効にします */
RCC->AHBENR |= RCC_AHBENR_DMA1EN; /* (1) */
DMA1_Channel2->CPAR = (uint32_t) (&(TIM1->CCR1)); /* (2) */
DMA1_Channel2->CMAR = (uint32_t) (&Period); /* (3) */
DMA1_Channel2->CNDTR = 1; /* (4) */
DMA1_Channel2->CCR |= DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
                    | DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel2->CCR |= DMA_CCR_EN; /* (6) */
/* チャンネル 3 に対して (2) から (6) を繰り返します */
DMA1_Channel3->CPAR = (uint32_t) (&(TIM1->CCR2)); /* (2) */
DMA1_Channel3->CMAR = (uint32_t) (&DutyCycle); /* (3) */
DMA1_Channel3->CNDTR = 1; /* (4) */
DMA1_Channel3->CCR |= DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_0
                    | DMA_CCR_TEIE | DMA_CCR_CIRC; /* (5) */
DMA1_Channel3->CCR |= DMA_CCR_EN; /* (6) */
```

```
/* DMA 用に NVIC を設定します */
/* (7) DMA チャンネル x の割り込みを有効にします */
/* (8) DMA チャンネル x の優先順位をセットします */
NVIC_EnableIRQ(DMA1_Channel2_3_IRQn); /* (7) */
NVIC_SetPriority(DMA1_Channel2_3_IRQn, 3); /* (8) */
```

A.9.7 入力比較設定のコード例

```
/* (1) APBCLK/4 (つまり 12 MHz) になるよう、プリスケアラを 3 にセットします */
/* (2) ARR = 12000 - 1 にセットします */
/* (3) タイマクロックが 12 MHz のときに CCRx = ARR にセットすると、1 ms ごとにイベントが発生します */
/* (4) OC1 のトグルモードを選択し (OC1M = 011)、OC1 のプリロードレジスタを無効にします (OC1PE = 0、リセット値) */
/* (5) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします (CC1E = 1) */
/* (6) 出力を有効にします (MOE = 1) */
/* (7) カウンタを有効にします */
TIMx->PSC |= 3; /* (1) */
TIMx->ARR = 12000 - 1; /* (2) */
TIMx->CCR1 = 12000 - 1; /* (3) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_0 | TIM_CCMR1_OC1M_1; /* (4) */
TIMx->CCER |= TIM_CCER_CC1E; /* (5) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (6) */
TIMx->CR1 |= TIM_CR1_CEN; /* (7) */
```

A.9.8 エッジアライン PWM の設定例

```
/* (1) APBCLK/48 (つまり 1MHz) になるよう、プリスケアラを 47 にセットします */
/* (2) タイマクロックが 1 MHz のときに ARR = 8 にセットすると、周期が 9 us になります */
/* (3) CCRx = 4 をセットすると、4 us の間は信号がハイになります */
/* (4) OC1 の PWM モード 1 を選択し (OC1M = 110)、OC1 のプリロードレジスタを有効にします (OC1PE = 1) */
/* (5) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします (CC1E = 1) */
/* (6) 出力を有効にします (MOE = 1) */
/* (7) カウンタを有効にし (CEN = 1)、エッジアラインモードを選択し (CMS = 00、リセット値)、アップカウンタとして方向を選択します (DIR = 0、リセット値) */
/* (8) 更新生成を強制します (UG = 1) */
TIMx->PSC = 47; /* (1) */
TIMx->ARR = 8; /* (2) */
TIMx->CCR1 = 4; /* (3) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1PE; /* (4) */
TIMx->CCER |= TIM_CCER_CC1E; /* (5) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (6) */
TIMx->CR1 |= TIM_CR1_CEN; /* (7) */
TIMx->EGR |= TIM_EGR_UG; /* (8) */
```



A.9.9 センターアライン PWM の設定例

```
/* (1) APBCLK/48 (つまり 1MHz) になるよう、プリスケアラを 47 にセットします */
/* (2) タイマクロックが 1 MHz でカウント方法がセンターアラインのときに ARR = 8 にセットすると、
    周期が 16 us になります */
/* (3) CCRx = 7 をセットすると、14 us の間は信号がハイになります */
/* (4) OC1 の PWM モード 1 を選択し (OC1M = 110)、OC1 のプリロードレジスタを有効にします
    (OC1PE = 1、リセット値) */
/* (5) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
    (CC1E = 1) */
/* (6) 出力を有効にします (MOE = 1) */
/* (7) カウンタを有効にし (CEN = 1)、センターアラインモード 1 を選択します (CMS = 01) */
/* (8) 更新生成を強制します (UG = 1) */
TIMx->PSC = 47; /* (1) */
TIMx->ARR = 8; /* (2) */
TIMx->CCR1 = 7; /* (3) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1
              | TIM_CCMR1_OC1PE; /* (4) */
TIMx->CCER |= TIM_CCER_CC1E; /* (5) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (6) */
TIMx->CR1 |= TIM_CR1_CMS_0 | TIM_CR1_CEN; /* (7) */
TIMx->EGR |= TIM_EGR_UG; /* (8) */
```

A.9.10 OCxREF をクリアするための ETR 設定のコード例

```
/* このコードはエッジアライン PWM 設定と同じですが、CCMR1 (5) の ETR クリア (OC1CE = 1) のために
    OC1 のクリアを有効にします。ETR は SMCR (7) で設定されます。*/
/* (1) APBCLK/48 (つまり 1MHz) になるよう、プリスケアラを 47 にセットします */
/* (2) タイマクロックが 1 MHz のときに ARR = 8 にセットすると、周期が 9 us になります */
/* (3) CCRx = 4 をセットすると、4 us の間は信号がハイになります */
/* (4) OC1 の PWM モード 1 を選択し (OC1M = 110)、OC1 のプリロードレジスタを有効にし
    (OC1PE = 1)、ETR クリア (OC1CE = 1) のために OC1 のクリアを有効にします */
/* (5) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
    (CC1E = 1) */
/* (6) 出力を有効にします (MOE = 1) */
/* (7) OCREF クリアソースとして ETR を選択し (OCCS = 1)、外部トリガプリスケアラをオフにし
    (ETPS = 00、リセット値)、外部クロックモード 2 を無効にし (ECE = 0、リセット値)、
    ハイレベルでアクティブを選択します (ETP = 0、リセット値) */
/* (8) カウンタを有効にし (CEN = 1)、エッジアラインモードを選択し (CMS = 00、リセット値)、
    アップカウンタとして方向を選択します (DIR = 0、リセット値) */
/* (9) 更新生成を強制します (UG = 1) */
TIMx->PSC = 47; /* (1) */
TIMx->ARR = 8; /* (2) */
TIMx->CCR1 = 4; /* (3) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1PE
              | TIM_CCMR1_OC1CE; /* (4) */
TIMx->CCER |= TIM_CCER_CC1E; /* (5) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (6) */
TIMx->SMCR |= TIM_SMCR_OCCS; /* (7) */
TIMx->CR1 |= TIM_CR1_CEN; /* (8) */
TIMx->EGR |= TIM_EGR_UG; /* (9) */
```


A.9.11 エンコーダインタフェースのコード例

```
/* (1) TI1 の TI1FP1 を設定し (CC1S = 01)、TI2 の TI1FP2 を設定します (CC2S = 01) */
/* (2) 非反転の TI1FP1 と TI1FP2 を設定します (CC1P = CC2P = 0、リセット値) */
/* (3) 立ち上がり立ち下がり両エッジで両方の入力をアクティブに設定します (SMS = 011) */
/* (4) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
TIMx->CCMR1 |= TIM_CCMR1_CC1S_0 | TIM_CCMR1_CC2S_0; /* (1) */
TIMx->CCER &= (uint16_t)(~(TIM_CCER_CC21 | TIM_CCER_CC2P)); /* (2) */
TIMx->SMCR |= TIM_SMCR_SMS_0 | TIM_SMCR_SMS_1; /* (3) */
TIMx->CR1 |= TIM_CR1_CEN; /* (4) */
```

A.9.12 リセットモードのコード例

```
/* (1) CC1S = '01'を書き込むことによって、TI1 入力の立ち上がりエッジを検出するように
   チャンネル 1 を設定し、TIMx_CCMR1 レジスタの IC1F[3:0] ビットに書き込むことによって、
   入力フィルタ時間を設定します (フィルタを使用しない場合は、IC1F=0000 にしておきます)。*/
/* (2) TIMx_CCER レジスタで CC1P=0を書き込むことによって立ち上がりエッジ極性を選択します。
   レジスタ
   リセット値は保持されるため、必要ありません。*/
/* (3) SMS=100 を書き込むことによって、タイマをリセットモードに設定します
   TIMx_SMCR レジスタに TS=101 を書き込むことによって、トリガ入力ソースとして TI1 を
   選択します。*/
/* (4) 1 ms ごとにインクリメントするために、プリスケアラを 48000-1 にセットします */
/* (5) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
TIMx->CCMR1 |= TIM_CCMR1_CC1S_0; /* (1) */
TIMx->CCER &= (uint16_t)(~TIM_CCER_CC1P); /* (2) */
TIMx->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_TS_2 | TIM_SMCR_TS_0; /* (3) */
TIM1->PSC = 47999; /* (4) */
TIMx->CR1 |= TIM_CR1_CEN; /* (5) */
```

A.9.13 ゲートモードのコード例

```
/* (1) CC1S = '01'を書き込むことによって、TI1 入力のローレベルを検出するようにチャンネル 1 を
   設定し、TIMx_CCMR1 レジスタの IC1F[3:0] ビットに書き込むことによって、
   入力フィルタ時間を設定します (フィルタを使用しない場合は、IC1F=0000 にしておきます)。*/
/* (2) TIMx_CCER レジスタの CC1P=1 を書き込んで、極性を選択します */
/* (3) SMS=101 を書き込むことによって、タイマをゲートモードに設定します
   TIMx_SMCR レジスタに TS=101 を書き込むことによって、トリガ入力ソースとして TI1 を
   選択します。*/
/* (4) 250 us ごとにインクリメントするために、プリスケアラを 12000-1 にセットします */
/* (5) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
TIMx->CCMR1 |= TIM_CCMR1_CC1S_0; /* (1) */
TIMx->CCER |= TIM_CCER_CC1P; /* (2) */
TIMx->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_0
            | TIM_SMCR_TS_2 | TIM_SMCR_TS_0; /* (3) */
TIMx->PSC = 11999; /* (4) */
```




```
TIMx->CR1 |= TIM_CR1_CEN; /* (5) */
```

A.9.14 トリガモードのコード例

```
/* (1) CC2S = '01' を書き込むことによって、TI2 入力の立ち上がりエッジを検出するように  
   チャンネル 2 を設定し、TIMx_CCMR1 レジスタの IC1F[3:0] ビットに書き込むことによって、  
   入力フィルタ時間を設定します（フィルタを使用しない場合は、IC1F=0000 にしておきます）。*/  
/* (2) TIMx_CCER レジスタに CC2P=0 （リセット値）を書き込むことによって極性を選択します。  
/* (3) SMS=110 を書き込むことによって、タイマをトリガモードに設定します  
   TIMx_SMCR レジスタに TS=110 を書き込むことによって、トリガ入力ソースとして TI2 を  
   選択します。*/  
/* (4) 250 us ごとにインクリメントするために、プリスケアラを 12000-1 にセットします */  
TIMx->CCMR1 |= TIM_CCMR1_CC2S_0; /* (1) */  
TIMx->CCER &= ~TIM_CCER_CC2P; /* (2) */  
TIMx->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_1  
           | TIM_SMCR_TS_2 | TIM_SMCR_TS_1; /* (3) */  
TIM1->PSC = 11999; /* (4) */
```

A.9.15 外部クロックモード 2 + トリガモードのコード例

```
/* (1) 入力フィルタは設定せず（ETF=0000、リセット値）、プリスケアラを無効に設定し  
   （ETPS = 0、リセット値）、ETR の立ち上がりエッジ検出を選択し（ETP = 0、リセット値）、  
   外部クロックモード 2 を有効にします（ECE = 1）*/  
/* (2) 入力フィルタは設定せず（IC1F=0000、リセット値）、TI1 の入力キャプチャソースを選択します  
   （CC1S = 01）*/  
/* (3) TIMx_CCER レジスタに CC1P=0 （リセット値）を書き込むことによって極性を選択します */  
/* (4) SMS=110 を書き込むことによって、タイマをトリガモードに設定します  
   TIMx_SMCR レジスタに TS=101 を書き込むことによって、トリガ入力ソースとして TI1 を  
   選択します。*/  
TIMx->SMCR |= TIM_SMCR_ECE; /* (1) */  
TIMx->CCMR1 |= TIM_CCMR1_CC1S_0; /* (2) */  
TIMx->CCER &= ~TIM_CCER_CC1P; /* (3) */  
TIMx->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_1  
           | TIM_SMCR_TS_2 | TIM_SMCR_TS_0; /* (4) */  
/* TI2FP2 をトリガ 1 として使用します */  
/* (1) TIMx_CCMR1 レジスタに CC2S=01 を書き込むことによって、TI2 にTI2FP2 を 配置します */  
/* (2) TI2FP2 は、立ち上がりエッジを検出して、TIMx_CCER レジスタに CC2P= 0 と CC2NP = 0 を  
   書き込みます（リセット値を保持します）*/  
/* (3) TI2FP2 をスレーブモードコントローラのトリガ（TRGI）として設定するために、  
   TIMx_SMCR レジスタに TS=110 を書き込み、TI2FP2 を使用してカウンタを開始するために、  
   TIMx_SMCR レジスタの SMS に'110'（トリガモード）を書き込みます */  
TIMx->CCMR1 |= TIM_CCMR1_CC2S_0; /* (1) */  
//TIMx->CCER &= ~(TIM_CCER_CC2P | TIM_CCER_CC2NP); /* (2) */  
TIMx->SMCR |= TIM_SMCR_TS_2 | TIM_SMCR_TS_1  
           | TIM_SMCR_SMS_2 | TIM_SMCR_SMS_1; /* (3) */
```

A.9.16 ワンパルスモードのコード例

```

/* OPM 波形は、比較レジスタを書き込むことによって定義されます */
/* (1) APBCLK/48 (つまり 1MHz) になるよう、プリスケアラを 47 にセットします */
/* (2) タイマクロックが 1 MHz のときに ARR = 7 にセットすると、周期が 8 us になります */
/* (3) CCRx = 5 をセットすると、5 us (> 0 である必要がある) のパースト遅延が生じます */
/* (4) OC1 の PWM モード 2 を選択し (OC1M = 111)、OC1 のプリロードレジスタを有効にし
    (OC1PE = 1、リセット値)、PULSE_WITHOUT_DELAY がセットされている場合に高速インネブル
    (遅延なし) を有効にします */
/* (5) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
    (CC1E = 1) */
/* (6) 出力を有効にします (MOE = 1) */
/* (7) TIMx_CR1 レジスタの OPM ビットに "1" を書き込むことによって、次の更新イベント時に
    カウンタを停止し (OPM = 1)、自動再ロードレジスタを有効にします (ARPE = 1) */
TIMx->PSC = 47; /* (1) */
TIMx->ARR = 7; /* (2) */
TIMx->CCR1 = 5; /* (3) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1M_0
              | TIM_CCMR1_OC1PE
#ifdef PULSE_WITHOUT_DELAY > 0
              | TIM_CCMR1_OC1FE
#endif
              ; /* (4) */
TIMx->CCER |= TIM_CCER_CC1E; /* (5) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (6) */
TIMx->CR1 |= TIM_CR1_OPM | TIM_CR1_ARPE; /* (7) */

```

A.9.17 別のタイマをプリスケールリングするタイマのコード例

```

/* TIMy は TIMx のスレーブです */
/* (1) TIMx_CR2 に MMS = 010 を書き込むことによって、更新イベントをトリガ出力 (TRG0) として
    選択します。*/
/* (2) TIMy_SMCR に TS=000 (リセット値) を書き込むことによって、内部トリガとして ITR1 を
    使用して、スレーブモードで TIMy を設定します
    TIMx_SMCR レジスタに SMS=111 を書き込むことによって、TIMy を外部クロックモード 1 に
    設定します。*/
/* (3) 1 ms ごとにインクリメントするために、TIMx プリスケアラを 47999 にセットします */
/* (4) 1 秒ごとにオーバーフロー (つまり UEV) するために、TIMx 自動再ロードを 999 にセットします */
/* (5) 24 時間ごとにオーバーフローするために、TIMx 自動再ロードを 24*3600-1 にセットします */
/* (6) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
/* (7) TIMy_CR1 レジスタに CEN=1 を書き込むことによって、カウンタを有効にします。*/
TIMx->CR2 |= TIM_CR2_MMS_1; /* (1) */
TIMy->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_1 | TIM_SMCR_SMS_0; /* (2) */
TIMx->PSC = 47999; /* (3) */
TIMx->ARR = 999; /* (4) */
TIMy->ARR = (24 * 3600) - 1; /* (5) */
TIMx->CR1 |= TIM_CR1_CEN; /* (6) */
TIMy->CR1 |= TIM_CR1_CEN; /* (7) */

```

A.9.18 別のタイマを有効にするタイマのコード例

```
/* TIMy は TIMx のスレーブです */
/* (1) タイマ x をマスタモードに設定し、その出力比較 1 基準 (OC1REF) 信号をトリガ出力として
    送信します (TIM1_CR2 レジスタの MMS=001)。*/
/* (2) タイマ x の OC1REF 波形 (TIM1_CCMR1 レジスタ) を設定します。
    チャンネル 1 は、カウンタがキャプチャ/比較レジスタよりも小さい場合に PWM モード 1 になります
    (OC1M = 110 と書き込みます)。*/
/* (3) TIMy_SMCR に TS=000 (リセット値) を書き込むことによって、内部トリガとして ITR1 を
    使用して、スレーブモードで TIMy を設定します
    TIMy_SMCR レジスタに SMS=101 を書き込むことによって、TIMy をゲートモードに設定します。*/
/* (4) TIMx プリスケールを 2 にセットします */
/* (5) TIMy プリスケールを 2 にセットします */
/* (6) 100 ms ごとにオーバーフロー (つまり UEV) するために、TIMx 自動再ロードを 999 に
    セットします */
/* (7) キャプチャ比較レジスタを 0 と 999 の間の値にセットします */
TIMx->CR2 |= TIM_CR2_MMS_2; /* (1) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1; /* (2) */
TIMy->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_0; /* (3) */
TIMx->PSC = 2; /* (4) */
TIMy->PSC = 2; /* (5) */
TIMx->ARR = 999; /* (6) */
TIMx->CCR1 = 700; /* (7) */
/* カウントごとにトグルを生成するためにスレーブタイマを設定します */
/* (1) PWM モード 1 の TIMy を設定します (OC1M = 110 を書き込みます) */
/* (2) TIMy 自動再ロードを 1 にセットします */
/* (3) キャプチャ比較レジスタを 1 にセットします */
TIMy->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1; /* (1) */
TIMy->ARR = 1; /* (2) */
TIMy->CCR1 = 1; /* (3) */
/* TIMx OC1 の出力を有効にします */
/* (1) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
    (CC1E = 1) */
/* (2) 出力を有効にします (MOE = 1) */
TIMx->CCER |= TIM_CCER_CC1E; /* (1) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (2) */
/* TIMy OC1 の出力を有効にします */
/* (1) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
    (CC1E = 1) */
/* (2) 出力を有効にします (MOE = 1) */
TIMy->CCER |= TIM_CCER_CC1E; /* (1) */
TIMy->BDTR |= TIM_BDTR_MOE; /* (2) */
/* (1) 最初に TIMy_CR1 レジスタに CEN=1 を書き込むことによって、スレーブカウンタを有効にします。
    */
/* (2) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、マスタカウンタを有効にします。*/
TIMy->CR1 |= TIM_CR1_CEN; /* (1) */
TIMx->CR1 |= TIM_CR1_CEN; /* (2) */
```

A.9.19 マスタとスレーブの同期のコード例

```

/* (1) タイマ x をマスタモードに設定し、イネーブル信号をトリガ出力として送信します
   (TIM1_CR2 レジスタの MMS=001)。*/
/* (2) PWM モード 1 のタイマ x チャンネル 1 波形 (TIM1_CCMR1 レジスタ) を設定します
   OC1M = 110 を書き込みます) */
/* (3) TIMy_SMCR に TS=000 (リセット値) を書き込むことによって、内部トリガとして ITR1 を
   使用して、スレーブモードで TIMy を設定します
   TIMy_SMCR レジスタに SMS=101 を書き込むことによって、TIMy をゲートモードに設定します。*/
/* (4) TIMx プリスケールを 2 にセットします */
/* (5) TIMy プリスケールを 2 にセットします */
/* (6) 10 ms ごとにオーバーフロー(つまり UEV)するために、TIMx 自動再ロードを 99 にセットします */
/* (7) キャプチャ比較レジスタを 0 と 99 の間の値にセットします */
TIMx->CR2 |= TIM_CR2_MMS_0; /* (1) */
TIMx->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1; /* (2) */
TIMy->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_0; /* (3) */
TIMx->PSC = 2; /* (4) */
TIMy->PSC = 2; /* (5) */
TIMx->ARR = 99; /* (6) */
TIMx->CCR1 = 25; /* (7) */

/* 同期性を示すタイマとして示すために、スレーブタイマチャンネル 1 を PWM に設定します */
/* (1) PWM モード 1 の TIMy を設定します (OC1M = 110 を書き込みます) */
/* (2) TIMy 自動再ロードを 99 にセットします */
/* (3) キャプチャ比較レジスタを 25 にセットします */
TIMy->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1; /* (1) */
TIMy->ARR = 99; /* (2) */
TIMy->CCR1 = 25; /* (3) */

/* TIMx OC1 の出力を有効にします */
/* (1) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
   (CC1E = 1) */
/* (2) 出力を有効にします (MOE = 1) */
TIMx->CCER |= TIM_CCER_CC1E; /* (1) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (2) */

/* TIMy OC1 の出力を有効にします */
/* (1) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
   (CC1E = 1) */
/* (2) 出力を有効にします (MOE = 1) */
TIMy->CCER |= TIM_CCER_CC1E; /* (1) */
TIMy->BDTR |= TIM_BDTR_MOE; /* (2) */

/* (1) UG ビット (TIMx_EGR レジスタ) に "1" を書き込むことによって、タイマ x をリセットします */
/* (2) UG ビット (TIMy_EGR レジスタ) に "1" を書き込むことによって、タイマ y をリセットします */
TIMx->EGR |= TIM_EGR_UG; /* (1) */
TIMy->EGR |= TIM_EGR_UG; /* (2) */

/* (1) 最初に TIMy_CR1 レジスタに CEN=1 を書き込むことによって、スレーブカウンタを有効にします。
   TIMy はマスタタイマと同時に開始します */
/* (2) TIMx_CR1 レジスタに CEN=1 を書き込むことによって、マスタカウンタを開始にします。*/
TIMy->CR1 |= TIM_CR1_CEN; /* (1) */
TIMx->CR1 |= TIM_CR1_CEN; /* (2) */

```

A.9.20 外部トリガによって同期された 2 つのタイマのコード例

```
/* (1) TIMx をマスタモードに設定し、イネーブル信号をトリガ出力として送信します
   (TIM1_CR2 レジスタの MMS=001)。*/
/* (2) TI1 から入力トリガを取得するために、TIMx_SMCR に TS = 100 を書き込むことによって、
   TIMx をスレーブモードで設定します
   TIMx_SMCR レジスタに SMS=110 を書き込むことによって、TIMx をトリガモードに設定します。
   TIMx_SMCR に MSM = 1 を書き込むことによって、TIMx をマスタ/スレーブモードに設定します。
/* (3) タイマ 1 から入力トリガを取得するために、TIMy_SMCR に TS = 000 (リセット値) を
   書き込むことによって、TIMy をスレーブモードで設定します
   TIMy_SMCR レジスタに SMS=110 を書き込むことによって、TIMy をトリガモードに設定します。*/
/* (4) UG ビット (TIMx_EGR レジスタ) に '1' を書き込むことによって、タイマ x カウンタを
   リセットします */
/* (5) UG ビット (TIMy_EGR レジスタ) に '1' を書き込むことによって、タイマ y カウンタを
   リセットします */
TIMx->CR2 |= TIM_CR2_MMS_0; /* (1) */
TIMx->SMCR |= TIM_SMCR_TS_2 | TIM_SMCR_SMS_2 | TIM_SMCR_SMS_1
             | TIM_SMCR_MSM; /* (2) */
TIMy->SMCR |= TIM_SMCR_SMS_2 | TIM_SMCR_SMS_1; /* (3) */
TIMx->EGR |= TIM_EGR_UG; /* (4) */
TIMy->EGR |= TIM_EGR_UG; /* (5) */
/* タイマチャンネル 2 を PWM に設定します */
/* (1) PWM モード 1 のタイマ x チャンネル 2 波形 (TIM1_CCMR1 レジスタ) を設定します
   (OC2M = 110 を書き込みます) */
/* (2) TIMx プリスケアラを 2 にセットします */
/* (3) 10 ms ごとにオーバーフロー(つまり UEV)するために、TIMx 自動再ロードを 99 にセットします */
/* (4) キャプチャ比較レジスタを 0 と 99 の間の値にセットします */
TIMx->CCMR1 |= TIM_CCMR1_OC2M_2 | TIM_CCMR1_OC2M_1; /* (1) */
TIMx->PSC = 2; /* (2) */
TIMx->ARR = 99; /* (3) */
TIMx->CCR2 = 25; /* (4) */
/* 同期性を示すタイマとして示すために、スレーブタイマチャンネル 1 を PWM に設定します */
/* (1) PWM モード 1 の TIMy を設定します (OC1M = 110 を書き込みます) */
/* (2) TIMy プリスケアラを 2 にセットします */
/* (3) TIMx 自動再ロードを 99 にセットします */
/* (4) キャプチャ比較レジスタを 25 にセットします */
TIMy->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1; /* (1) */
TIMy->PSC = 2; /* (2) */
TIMy->ARR = 99; /* (3) */
TIMy->CCR1 = 25; /* (4) */
/* TIMx OC1 の出力を有効にします */
/* (1) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
   (CC1E = 1) */
/* (2) 出力を有効にします (MOE = 1) */
TIMx->CCER |= TIM_CCER_CC2E; /* (1) */
TIMx->BDTR |= TIM_BDTR_MOE; /* (2) */
/* TIMy OC1 の出力を有効にします */
/* (1) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
   (CC1E = 1) */
/* (2) 出力を有効にします (MOE = 1) */
TIMy->CCER |= TIM_CCER_CC1E; /* (1) */
TIMy->BDTR |= TIM_BDTR_MOE; /* (2) */
```

A.9.21 DMA バースト機能のコード例

```
/* この例では、TIMx は PWM センターアラインで事前に設定されています*/

/* DMA バースト機能を設定します */
/* 対応する DMA チャンネルを設定します */
/* (1) DMA チャンネルペリフェラルアドレスを、DMAR レジスタアドレスにセットします */
/* (2) DMA によって CCRx レジスタに転送されるデータを含む RAM のバッファのアドレスに、
DMA チャンネルメモリアドレスをセットします */
/* (3) データの転送数を (Duty_Cycle_Table) のサイズにセットします */
/* (4) CCR レジスタの DMA 転送を設定し、CIRC ビットをセットしてサーキュラモードを有効にし
(オプション)、メモリサイズを 16_bits MSIZE = 01 にセットし、ペリフェラルサイズを
32_bits PSIZE = 10 にセットし、MINC をセットしてメモリインクリメントモードを有効にし、
DIR をセットしてメモリから読み出したデータ転送方向をセットします */
/* (5) DBL = 3 転送および DBA = (@TIMx->CCR2 - @TIMx->CR1) >> 2 = 0xE で
TIMx_DCR レジスタを設定します */
/* (6) DIER レジスタの UDE ビットをセットして TIMx 更新 DMA リクエストを有効にします */
/* (7) TIMx を有効にします */
/* (8) DMA チャンネルを有効にします */
DMA1_Channel2->CPAR = (uint32_t)(&(TIMx->DMAR)); /* (1) */
DMA1_Channel2->CMAR = (uint32_t)(Duty_Cycle_Table); /* (2) */
DMA1_Channel2->CNDTR = 10*3; /* (3) */
DMA1_Channel2->CCR |= DMA_CCR_CIRC | DMA_CCR_MSIZE_0 | DMA_CCR_PSIZE_1
| DMA_CCR_MINC | DMA_CCR_DIR; /* (4) */
TIMx->DCR = (3 << 8)
+ (((uint32_t)(&TIM2->CCR2))
- ((uint32_t)(&TIM2->CR1))) >> 2); /* (5) */
TIMx->DIER |= TIM_DIER_UDE; /* (6) */
TIMx->CR1 |= TIM_CR1_CEN; /* (7) */
DMA1_Channel2->CCR |= DMA_CCR_EN; /* (8) */
```


A.10 IRTIM のコード例

A.10.1 TIM16 および TIM17 設定のコード例

```
/* 以下は RC5 規格の設定です */
/* TIM17 がキャリアとして使用されている間、TIM16 はエンベロープに使用されます */
#define TIM_ENV TIM16
#define TIM_CAR TIM17
/* (1) タイマ 16、17、および SYSCFG のペリフェラルクロックを有効にします */
/* (2) GPIOB のペリフェラルクロックを有効にします */
/* (3) GPIOB ピン 9 のオルタネート機能モードを選択します */
/* (4) IR_OUT について、AFRH の PB9 で AF0 (リセット値)を選択します */
/* (5) SYSCFG_CFGR1 の I2C_PB9_FMP+ ビットをセットすることで高シンクドライバ機能を有効にします */
RCC->APB2ENR |= RCC_APB2ENR_TIM16EN | RCC_APB2ENR_TIM17EN
               | RCC_APB2ENR_SYSCFGEN; /* (1) */
RCC->AHBENR |= RCC_AHBENR_GPIOBEN; /* (2) */
GPIOB->MODER = (GPIOB->MODER & ~GPIO_MODER_MODER9)
               | GPIO_MODER_MODER9_1; /* (3) */
GPIOB->AFR[1] &= ~(0x0F << ((9 - 8) * 4)); /* (4) */
SYSCFG->CFGR1 |= SYSCFG_CFGR1_I2C_FMP_PB9; /* (5) */
/* キャリア信号として TIM_CAR を設定します */
/* (1) APBCLK (つまり 48 MHz) になるよう、プリスケアラを 1 にセットします */
/* (2) タイマクロックが 48 MHz のときに ARR = 1333 にセットすると、周期が 36 kHz になります */
/* (3) CCRx = 1333/4 をセットすると、信号は 25% デューティサイクルを持ちます */
/* (4) OC1 の PWM モード 1 を選択し (OC1M = 110)、OC1 のプリロードレジスタを有効にします
(OC1PE = 1) */
/* (5) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
(CC1E = 1) */
/* (6) 出力を有効にします (MOE = 1) */
TIM_CAR->PSC = v; /* (1) */
TIM_CAR->ARR = 1333; /* (2) */
TIM_CAR->CCR1 = (uint16_t)(1333 / 4); /* (3) */
TIM_CAR->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1
                | TIM_CCMR1_OC1PE; /* (4) */
TIM_CAR->CCER |= TIM_CCER_CC1E; /* (5) */
TIM_CAR->BDTR |= TIM_BDTR_MOE; /* (6) */
/* TIM_ENV を変調エンベロープとして設定します */
/* (1) APBCLK (つまり 48 MHz) になるよう、プリスケアラを 1 にセットします */
/* (2) タイマクロックが 48 MHz のときに ARR = 42627 にセットすると、周期が 888 us になります */
/* (3) OC1 の強制インアクティブを選択します (OC1M = 100) */
/* (4) OC1 のアクティブハイ極性を選択し (CC1P = 0、リセット値)、OC1 の出力を有効にします
(CC1E = 1) */
/* (5) 出力を有効にします (MOE = 1) */
/* (6) 更新割り込みを有効にします (UIE = 1) */
TIM_ENV->PSC = 0; /* (1) */
TIM_ENV->ARR = 42627; /* (2) */
TIM_ENV->CCMR1 |= TIM_CCMR1_OC1M_2; /* (3) */
TIM_ENV->CCER |= TIM_CCER_CC1E; /* (4) */
TIM_ENV->BDTR |= TIM_BDTR_MOE; /* (5) */
TIM_ENV->DIER |= TIM_DIER_UIE; /* (6) */
/* TIM_CAR のみを有効にし、リセットします */
/* (1) カウンタを有効にし (CEN = 1)、エッジアラインモードを選択し (CMS = 00、リセット値)、
```



```

        アップカウンタとして方向を選択します (DIR = 0、リセット値) */
/* (2) 更新生成を強制します (UG = 1) */
TIM_CAR->CR1 |= TIM_CR1_CEN; /* (1) */
TIM_CAR->EGR |= TIM_EGR_UG; /* (2) */
/* TIM_ENV 割り込みを設定します */
/* (1) TIM_ENV の割り込みを有効にします */
/* (2) TIM_ENV の優先順位をセットします */
NVIC_EnableIRQ(TIM_ENV_IRQn); /* (1) */
NVIC_SetPriority(TIM_ENV_IRQn, 0); /* (2) */

```

A.10.2 IRTIM の IRQHandler のコード例

```

/**
 * 説明: この機能では、TIM_16 割り込みリクエストを処理します。
 *       この割り込みサブルーチンでは、TIMIC の 2 つの立ち上がりエッジ間のラップを計算します。
 *       このラップは「カウンタ」変数に格納されます。
 */
void TIM16_IRQHandler(void)
{
    uint8_t bit_msg = 0;

    if ((SendOperationReady == 1)
        && (BitsSentCounter < (RC5_GlobalFrameLength * 2)))
    {
        if (BitsSentCounter < 32)
        {
            SendOperationCompleted = 0x00;
            bit_msg = (uint8_t)((ManchesterCodedMsg >> BitsSentCounter) & 1);

            if (bit_msg == 1)
            {
                /* 強制アクティブレベル - OC1REF は強制的にハイになります */
                TIM_ENV->CCMR1 |= TIM_CCMR1_OC1M_0;
            }
            else
            {
                /* 強制インアクティブレベル - OC1REF は強制的にローになります */
                TIM_ENV->CCMR1 &= (uint16_t)(~TIM_CCMR1_OC1M_0);
            }
        }
        BitsSentCounter++;
    }
    else
    {
        SendOperationCompleted = 0x01;
        SendOperationReady = 0;
        BitsSentCounter = 0;
    }
    /* TIM_ENV 更新割り込みをクリアします */
    TIM_ENV->SR &= (uint16_t)(~TIM_SR_UIF);
}

```

A.11 bxCAN のコード例

A.11.1 bxCAN 初期化モードのコード例

```
/* (1) 設定を書き込むために CAN 初期化モードに移行します */
/* (2) 初期化モードの移行を待ちます */
/* (3) SLEEP モードを終了します */
/* (4) ループバックモード、タイミングを 1 Mb/s にセットします (BS1 = 4、BS2 = 3、
    プリスケアラ = 6) */
/* (5) 初期化モードを終了します */
/* (6) 初期化モードの終了を待ちます */
/* (7) フィルタ初期化モードに移行します (16 ビット + マスク、FIFO 0 のときフィルタ 0) */
/* (8) フィルタ 0 を有効にします */
/* (9) ID とマスクをセットします (標準 ID ケアのすべてのビット) */
/* (10) フィルタの初期化を終了します */
/* (11) IT イネーブルをペンディングする FIFO0 メッセージをセットします */
CAN->MCR |= CAN_MCR_INRQ; /* (1) */
while ((CAN->MSR & CAN_MSR_INAK) != CAN_MSR_INAK) /* (2) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
CAN->MCR &=~ CAN_MCR_SLEEP; /* (3) */
CAN->BTR |= CAN_BTR_LBKM | 2 << 20 | 3 << 16 | 5 << 0; /* (4) */
CAN->MCR &=~ CAN_MCR_INRQ; /* (5) */
while ((CAN->MSR & CAN_MSR_INAK) == CAN_MSR_INAK) /* (6) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
CAN->FMR |= CAN_FMR_FINIT; /* (7) */
CAN->FA1R |= CAN_FA1R_FACT0; /* (8) */
CAN->sFilterRegister[0].FR1 = CAN_ID << 5 | 0xFF70U << 16; /* (9) */
CAN->FMR &=~ CAN_FMR_FINIT; /* (10) */
CAN->IER |= CAN_IER_FMPIE0; /* (11) */
```

A.11.2 bxCAN 送信のコード例

```
/* (1) メールボックス 0 がエンプティであることをチェックします */
/* (2) データ長さに 1 を書き込みます */
/* (3) 8 ビットデータを書き込みます */
/* (4) ID フィールドに書き込んで、送信をリクエストします */
if ((CAN->TSR & CAN_TSR_TME0) == CAN_TSR_TME0) /* (1) */
{
    CAN->sTxMailBox[0].TDTR = 1; /* (2) */
    CAN->sTxMailBox[0].TDLR = CMD; /* (3) */
    CAN->sTxMailBox[0].TIR = (uint32_t)(CAN_ID << 21
        | CAN_TI0R_TXRQ); /* (4) */
}
```

A.11.3 bxCAN 受信のコード例

```
/* メッセージが FIFO 0 によってフィルタリングされ、受信されたかをチェックします */  
if ((CAN->RF0R & CAN_RF0R_FMP0) != 0)  
{  
    CAN_ReceiveMessage = CAN->sFIFOMailBox[0].RDLR; /* データを読み出します */  
    CAN->RF0R |= CAN_RF0R_RFOM0; /* FIFO を解放します */  
    if ((CAN_ReceiveMessage & 0xFF) == CMD)  
    {  
        /* 処理します */  
    }  
}
```

A.12 DBG のコード例

A.12.1 DBG 読み出しデバイス ID のコード例

```
/* MCU ID を 32 ビットアクセスで読み出します */  
MCU_Id = DBGMCU->IDCODE;
```

A.12.2 低電力モードの DBG デバッグのコード例

```
/* STOP モードでデバッグできるようにします */  
DBGMCU->CR |= DBGMCU_CR_DBG_STOP;
```

A.13 HDMI-CEC のコード例

A.13.1 CEC を設定する HDMI-CEC のコード例

```
/* (1) OAR = 0x0001 => OA = 0x0 */
/* (2) 受信バイト割り込み有効、受信終了割り込み有効 */
/* (3) CEC 有効 */
CEC->CFGR = (0x001<<16); /* (1) */
CEC->IER = CEC_IER_RXBRIE|CEC_IER_RXENDIE; /* (2) */
CEC->CR = CEC_CR_CECEN; /* (3) */
```

A.13.2 割り込みが有効な HDMI-CEC 送信のコード例

```
/* (1) 送信 IT をセットします */
/* (2) ニブルを使用して送信データレジスタを書き込みます（イニシエータ 0x0、転送先 0x1）*/

/* (3) メッセージビットの開始をセットします */
CEC->IER |= CEC_IER_TXBRIE; /* (1) */
CEC->TXDR = (uint32_t)(ADDRESS_INITIATOR << 4
                      | ADDRESS_DESTINATION); /* (2) */
CEC->CR |= CEC_CR_TXSOM; /* (3) */
```

A.13.3 HDMI-CEC の割り込み管理のコード例

```
if ((CEC->ISR & CEC_ISR_RXEND) == CEC_ISR_RXEND)
{
    CEC->ISR = CEC_ISR_RXBR | CEC_ISR_RXEND; /* フラグをリセットします */
    Received_Data = CEC->RXDR;
    if (Received_Data == CMD)
    {
        /* 処理します */
    }
}
else if ((CEC->ISR & CEC_ISR_RXBR) == CEC_ISR_RXBR)
{
    CEC->ISR = CEC_ISR_RXBR; /* フラグをリセットします */
    Received_Data = CEC->RXDR;
    /* 処理します */
}
else if ((CEC->ISR & CEC_ISR_TXBR) == CEC_ISR_TXBR)
{
    CEC->IER &= ~CEC_IER_TXBRIE; /* Tx IT をリセットします */
    CEC->CR |= CEC_CR_TXBOM; /* これが最後のバイトです */
    CEC->TXDR = CMD;
}
```

A.14 I2C のコード例

A.14.1 受信するためにマスタモードに設定された I2C のコード例

```
/* (1) タイミングレジスタ値は AN4235 xls ファイルで計算され、高速モードは I2CCLK が 48 MHz の  
    とき @400 kHz、立ち上がり時間は 140 ns、立ち下がり時間は 40 ns です */  
/* (2) ペリフェラル有効、受信割り込み有効 */  
/* (3) スレーブアドレス = 0x5A、読み出し転送、受信するための 1 バイト、自動終了 */  
I2C2->TIMINGR = (uint32_t)0x00B01A4B; /* (1) */  
I2C2->CR1 = I2C_CR1_PE | I2C_CR1_RXIE; /* (2) */  
I2C2->CR2 = I2C_CR2_AUTOEND | (1<<16) | I2C_CR2_RD_WRN  
            | (I2C1_OWN_ADDRESS << 1); /* (3) */
```

A.14.2 送信するためにマスタモードに設定された I2C のコード例

```
/* (1) タイミングレジスタ値は AN4235 xls ファイルで計算され、高速モードは I2CCLK が  
    48 MHz のとき @400 kHz、立ち上がり時間は 140 ns、立ち下がり時間は 40 ns です */  
/* (2) ペリフェラル有効 */  
/* (3) スレーブアドレス = 0x5A、書き込み転送、送信するための 1 バイト、自動終了 */  
I2C2->TIMINGR = (uint32_t)0x00B01A4B; /* (1) */  
I2C2->CR1 = I2C_CR1_PE; /* (2) */  
I2C2->CR2 = I2C_CR2_AUTOEND | (1 << 16) | (I2C1_OWN_ADDRESS << 1); /* (3) */
```

A.14.3 スレーブモードで設定された I2C のコード例

```
/* (1) タイミングレジスタ値は AN4235 xls ファイルで計算され、高速モードは I2CCLK が  
    48 MHz のとき @400 kHz、立ち上がり時間は 140 ns、立ち下がり時間は 40 ns です */  
/* (2) ペリフェラル有効、アドレス一致割り込み有効 */  
/* (3) 7 ビットアドレス = 0x5A */  
/* (4) 自己アドレス 1 を有効にします */  
I2C1->TIMINGR = (uint32_t)0x00B00000; /* (1) */  
I2C1->CR1 = I2C_CR1_PE | I2C_CR1_ADDRIE; /* (2) */  
I2C1->OAR1 |= (uint32_t)(I2C1_OWN_ADDRESS << 1); /* (3) */  
I2C1->OAR1 |= I2C_OAR1_OA1EN; /* (4) */
```

A.14.4 I2C マスタトランスミッタのコード例

```
/* Tx エンプティをチェックします */
if ((I2C2->ISR & I2C_ISR_TXE) == I2C_ISR_TXE)
{
    I2C2->TXDR = I2C_BYTE_TO_SEND; /* 送信するバイトです */
    I2C2->CR2 |= I2C_CR2_START; /* 送信します */
}
```

A.14.5 I2C マスタレシーバのコード例

```
if ((I2C2->ISR & I2C_ISR_RXNE) == I2C_ISR_RXNE)
{
    /* 受信レジスタを読み出すと、RXNE フラグがクリアされます */
    if (I2C2->RXDR == I2C_BYTE_TO_SEND)
    {
        /* 処理します */
    }
}
```

A.14.6 I2C スレーブトランスミッタのコード例

```
uint32_t I2C_InterruptStatus = I2C1->ISR; /* 割り込みステータスを取得します */
/* アドレス一致をチェックします */
if ((I2C_InterruptStatus & I2C_ISR_ADDR) == I2C_ISR_ADDR)
{
    I2C1->ICR |= I2C_ICR_ADDRCF; /* アドレス一致フラグをクリアします */
    /* 転送方向が読み出されたかどうかをチェックします (スレーブトランスミッタ) */
    if ((I2C1->ISR & I2C_ISR_DIR) == I2C_ISR_DIR)
    {
        I2C1->CR1 |= I2C_CR1_TXIE; /* 送信 IT をセットします */
    }
}
else if ((I2C_InterruptStatus & I2C_ISR_TXIS) == I2C_ISR_TXIS)
{
    I2C1->CR1 &=~ I2C_CR1_TXIE; /* 送信 IT を無効にします */
    I2C1->TXDR = I2C_BYTE_TO_SEND; /* 送信するバイトです */
}
```

A.14.7 I2C スレーブレシーバのコード例

```
uint32_t I2C_InterruptStatus = I2C1->ISR; /* 割り込みステータスを取得します */
if ((I2C_InterruptStatus & I2C_ISR_ADDR) == I2C_ISR_ADDR)
{
    I2C1->ICR |= I2C_ICR_ADDRCF; /* アドレス一致イベント */
}
else if ((I2C_InterruptStatus & I2C_ISR_RXNE) == I2C_ISR_RXNE)
{
    /* 受信レジスタを読み出すと、RXNE フラグがクリアされます */
    if (I2C1->RXDR == I2C_BYTE_TO_SEND)
    {
        /* 処理します */
    }
}
```

A.14.8 DMA で送信するためにマスタモードに設定された I2C のコード例

```
/* (1) タイミングレジスタ値は AN4235 xls ファイルで計算され、高速モードは I2CCCLK が
    48 MHz のとき @400 kHz、立ち上がり時間は 140 ns、立ち下がり時間は 40 ns です */
/* (2) ペリフェラル有効 */
/* (3) スレーブアドレス = 0x5A、書き込み転送、送信するための 2 バイト、自動終了 */
I2C2->TIMINGR = (uint32_t)0x00B01A4B; /* (1) */
I2C2->CR1 = I2C_CR1_PE | I2C_CR1_TXDMAEN; /* (2) */
I2C2->CR2 = I2C_CR2_AUTOEND | (SIZE_OF_DATA << 16)
            | (I2C1_OWN_ADDRESS << 1); /* (3) */
```

A.14.9 DMA で受信するためにスレーブモードに設定された I2C のコード例

```
/* (1) タイミングレジスタ値は AN4235 xls ファイルで計算され、高速モードは I2CCCLK が
    48 MHz のとき @400 kHz、立ち上がり時間は 140 ns、立ち下がり時間は 40 ns です */
/* (2) ペリフェラル有効、受信 DMA 有効 */
/* (3) 7 ビットアドレス = 0x5A */
/* (4) 自己アドレス 1 を有効にします */
I2C1->TIMINGR = (uint32_t)0x00B00000; /* (1) */
I2C1->CR1 = I2C_CR1_PE | I2C_CR1_RXDMAEN | I2C_CR1_ADDR1E; /* (2) */
I2C1->OAR1 |= (uint32_t)(I2C1_OWN_ADDRESS << 1); /* (3) */
I2C1->OAR1 |= I2C_OAR1_OA1EN; /* (4) */
```


A.15 IWDG のコード例

A.15.1 IWDG 設定のコード例

```
/* (1) IWDG を有効にします (オプションバイトで作業する場合は不要) */
/* (2) IWDG レジスタへの書き込みアクセスを有効にします */
/* (3) プリスケアラを 8 にセットします */
/* (4) ロールオーバーがそれぞれ 100 ms となるような再ロード値をセットします */
/* (5) フラグがリセットされているかどうかをチェックします */
/* (6) カウンタをリフレッシュします */
IWDG->KR = IWDG_START; /* (1) */
IWDG->KR = IWDG_WRITE_ACCESS; /* (2) */
IWDG->PR = IWDG_PR_PR_0; /* (3) */
IWDG->RLR = IWDG_RELOAD; /* (4) */
while (IWDG->SR) /* (5) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
IWDG->KR = IWDG_REFRESH; /* (6) */
```

A.15.2 ウィンドウによる IWDG 設定のコード例

```
/* (1) IWDG を有効にします (オプションバイトで作業する場合は不要) */
/* (2) IWDG レジスタへの書き込みアクセスを有効にします */
/* (3) プリスケアラを 8 にセットします */
/* (4) ロールオーバーがそれぞれ 100 ms となるような再ロード値をセットします */
/* (5) フラグがリセットされているかどうかをチェックします */
/* (6) 50 ms ウィンドウをセットして IWDG をリフレッシュします */
IWDG->KR = IWDG_START; /* (1) */
IWDG->KR = IWDG_WRITE_ACCESS; /* (2) */
IWDG->PR = IWDG_PR_PR_0; /* (3) */
IWDG->RLR = IWDG_RELOAD; /* (4) */
while (IWDG->SR) /* (5) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
IWDG->WINR = IWDG_RELOAD >> 1; /* (6) */
```

A.16 RTC のコード例

A.16.1 RTC カレンダー設定のコード例

```
/* (1) RTC レジスタへのアクセスを書き込みます */
/* (2) 初期化フェーズを有効にします */
/* (3) RTC レジスタ値を修正できるようになるまで待ちます */
/* (4) 40kHz/128 => 312 Hz、312Hz/312 => 1Hz になるようにプリスケアラをセットします */
/* (5) TR の新しい時間 */
/* (6) 初期化フェーズを無効にします */
/* (7) RTC レジスタへの書き込みアクセスを無効にします */
RTC->WPR = 0xCA; /* (1) */
RTC->WPR = 0x53; /* (1) */
RTC->ISR |= RTC_ISR_INIT; /* (2) */
while ((RTC->ISR & RTC_ISR_INITF) != RTC_ISR_INITF) /* (3) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
RTC->PRER = 0x007F0137; /* (4) */
RTC->TR = RTC_TR_PM | Time; /* (5) */
RTC->ISR &= ~RTC_ISR_INIT; /* (6) */
RTC->WPR = 0xFE; /* (7) */
RTC->WPR = 0x64; /* (7) */
```

A.16.2 RTC アラーム設定のコード例

```
/* (1) RTC レジスタへのアクセスを書き込みます */
/* (2) 修正するためにアラーム A を無効にします */
/* (3) アラーム A 値を修正できるようになるまで待ちます */
/* (4) 1 Hz ごとに割り込みが生じるようにアラーム A マスクを修正します */
/* (5) アラーム A とアラーム A 割り込みを有効にします */
/* (6) 書き込みアクセスを無効にします */
RTC->WPR = 0xCA; /* (1) */
RTC->WPR = 0x53; /* (1) */
RTC->CR &= ~RTC_CR_ALRAE; /* (2) */
while ((RTC->ISR & RTC_ISR_ALRAWF) != RTC_ISR_ALRAWF) /* (3) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
RTC->ALRMAR = RTC_ALRMAR_MSK4 | RTC_ALRMAR_MSK3
             | RTC_ALRMAR_MSK2 | RTC_ALRMAR_MSK1; /* (4) */
RTC->CR = RTC_CR_ALRAIE | RTC_CR_ALRAE; /* (5) */
RTC->WPR = 0xFE; /* (6) */
RTC->WPR = 0x64; /* (6) */
```

A.16.3 RTC WUT 設定のコード例

```
/* (1) RTC レジスタへのアクセスを書き込みます */
/* (2) 修正するためにウェイクアップタイマを無効にします */
/* (3) ウェイクアップ再ロード値を修正できるようになるまで待ちます */
/* (4) 1 Hz ごとにウェイクアップが生じるようにウェイクアップ値の再ロードカウンタを修正します */
/* (5) ウェイクアップカウンタとウェイクアップ割り込みを有効にします */
/* (6) 書き込みアクセスを無効にします */
RTC->WPR = 0xCA; /* (1) */
RTC->WPR = 0x53; /* (1) */
RTC->CR &= ~RTC_CR_WUTE; /* (2) */
while ((RTC->ISR & RTC_ISR_WUTWF) != RTC_ISR_WUTWF) /* (3) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
RTC->WUTR = 0x9C0; /* (4) */
RTC->CR = RTC_CR_WUTE | RTC_CR_WUTIE; /* (5) */
RTC->WPR = 0xFE; /* (6) */
RTC->WPR = 0x64; /* (6) */
```

A.16.4 RTC 読み出しカレンダーのコード例

```
if((RTC->ISR & RTC_ISR_RSF) == RTC_ISR_RSF)
{
    TimeToCompute = RTC->TR; /* 時間を取得します */
    DateToCompute = RTC->DR; /* 日付も読み出す必要があります */
}
```

A.16.5 RTC 較正のコード例

```
/* (1) RTC レジスタへのアクセスを書き込みます */
/* (2) 初期化フェーズを有効にします */
/* (3) RTC レジスタ値を修正できるようになるまで待ちます */
/* (4) 40kHz/125 => 320 Hz、320Hz/320 => 1Hz になるようにプリスケアラをセットします */
/* (5) TR の新しい時間 */
/* (6) 初期化フェーズを無効にします */
/* (7) 較正レジスタを修正できるようになるまで待ちます */
/* (8) 標準値が @25°C の較正を約 +20 ppm にセットします */
/* 注: RTC クロックに LSE を選択した場合に、較正に適用されます */
/* (9) RTC レジスタへの書き込みアクセスを無効にします */
RTC->WPR = 0xCA; /* (1) */
RTC->WPR = 0x53; /* (1) */
RTC->ISR |= RTC_ISR_INIT; /* (2) */
while ((RTC->ISR & RTC_ISR_INITF) != RTC_ISR_INITF) /* (3) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
RTC->PRER = (124<<16) | 319; /* (4) */
RTC->TR = RTC_TR_PM | Time; /* (5) */
```

```
RTC->ISR &=~ RTC_ISR_INIT; /* (6) */
while((RTC->ISR & RTC_ISR_RECALPF) == RTC_ISR_RECALPF) /* (7) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
RTC->CALR = RTC_CALR_CALP | 482; /* (8) */
RTC->WPR = 0xFE; /* (9) */
RTC->WPR = 0x64; /* (9) */
```

A.16.6 RTC タンパおよびタイムスタンプ設定のコード例

```
/* タンパ設定 :
- プリチャージ (PU) の無効化
- RTCCLK/256 タンパサンプリング周波数
- タンパ検出時のタイムスタンプの有効化
- RTC_TAMP2 (PA0) への立ち上がりエッジのトリガ検出の入力
- タンパ割り込みイネーブル */
RTC->TAFCR = RTC_TAFCR_TAMPUDIS | RTC_TAFCR_TAMPFREQ | RTC_TAFCR_TAMPTS
            | RTC_TAFCR_TAMP2E | RTC_TAFCR_TAMPIE;
```

A.16.7 RTC タンパおよびタイムスタンプのコード例

```
/* タンパとタイムスタンプフラグをチェックします */
if (((RTC->ISR & (RTC_ISR_TAMP2F)) == (RTC_ISR_TAMP2F))
    && ((RTC->ISR & (RTC_ISR_TSF)) == (RTC_ISR_TSF)))
{
    RTC->ISR &= ~RTC_ISR_TAMP2F; /* タンパフラグをクリアします */
    EXTI->PR |= EXTI_PR_PR19; /* EXTI ライン 19 フラグをクリアします */
    TimeToCompute = RTC->TSTR; /* タイムスタンプレジスタのタンパタイムを取得します */
    RTC->ISR &= ~RTC_ISR_TSF; /* タイムスタンプフラグをクリアします */
}
```

A.16.8 RTC クロック出力のコード例

```
/* (1) RTC レジスタへのアクセスを書き込みます */
/* (2) 修正するためにアラーム A を無効にします */
/* (3) アラーム A 値を修正できるようになるまで待ちます */
/* (4) 1 Hz ごとに割り込みが生じるようにアラーム A マスクを修正します */
/* (5) アラーム A とアラーム A 割り込みを有効にし、較正出力 (1 Hz) を有効にします */
/* (6) 書き込みアクセスを無効にします */
RTC->WPR = 0xCA; /* (1) */
RTC->WPR = 0x53; /* (1) */
RTC->CR &=~ RTC_CR_ALRAE; /* (2) */
while ((RTC->ISR & RTC_ISR_ALRAWF) != RTC_ISR_ALRAWF) /* (3) */
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
RTC->ALRMAR = RTC_ALRMAR_MSK4 | RTC_ALRMAR_MSK3
```

```
        | RTC_ALRMAR_MSK2 | RTC_ALRMAR_MSK1; /* (4) */
RTC->CR = RTC_CR_ALRAIE | RTC_CR_ALRAE | RTC_CR_COE
        | RTC_CR_COSEL; /* (5) */
RTC->WPR = 0xFE; /* (6) */
RTC->WPR = 0x64; /* (6) */
```

A.17 SPI のコード例

A.17.1 SPI マスタ設定のコード例

```
/* (1) マスタ選択、BR: Fpclk/256 (ボード上の C27 により、SPI_CLK は最小値にセットされる)、
    CPOL および CPHA は 0 (最初の立ち上がりエッジ) */
/* (2) スレーブ選択出力有効、RXNE IT、8 ビット Rx fifo */
/* (3) SPI1 を有効にします */
SPI1->CR1 = SPI_CR1_MSTR | SPI_CR1_BR; /* (1) */
SPI1->CR2 = SPI_CR2_SSOE | SPI_CR2_RXNEIE | SPI_CR2_FRXTH
        | SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; /* (2) */
SPI1->CR1 |= SPI_CR1_SPE; /* (3) */
```

A.17.2 SPI スレーブ設定のコード例

```
/* nSS ハード、スレーブ、CPOL および CPHA は 0 (最初の立ち上がりエッジ) */
/* (1) RXNE IT、8 ビット Rx fifo */
/* (2) SPI2 を有効にします */
SPI2->CR2 = SPI_CR2_RXNEIE | SPI_CR2_FRXTH
        | SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; /* (1) */
SPI2->CR1 |= SPI_CR1_SPE; /* (2) */
```

A.17.3 SPI 全二重通信のコード例

```
if ((SPI1->SR & SPI_SR_TXE) == SPI_SR_TXE) /* Tx エンプティをテストします */
{
    /* TXE の場合に 8 ビット通信を初期化します */
    *(uint8_t *)&(SPI1->DR) = SPI1_DATA;
}
```

A.17.4 SPI 割り込みのコード例

```
if ((SPI1->SR & SPI_SR_RXNE) == SPI_SR_RXNE)
{
    SPI1_Data = (uint8_t)SPI1->DR; /* データを受信し、フラグをクリアします */
    /* 処理します */
}
```

A.17.5 DMA による SPI マスタ設定のコード例

```
/* (1) マスタ選択、BR: Fpclk/256 (ボード上の C27 により、SPI_CLK は最小値にセットされる)、
    CPOL および CPHA は 0 (最初の立ち上がりエッジ) */
/* (2) DMA による TX および RX、スレーブ選択出力を有効にし、RXNE 割り込みを有効にし、
    8 ビット Rx fifo を選択します */
/* (3) SPI1 を有効にします */
SPI1->CR1 = SPI_CR1_MSTR | SPI_CR1_BR; /* (1) */
SPI1->CR2 = SPI_CR2_TXDMAEN | SPI_CR2_RXDMAEN | SPI_CR2_SSOE
            | SPI_CR2_RXNEIE | SPI_CR2_FRXTH
            | SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; /* (2) */
SPI1->CR1 |= SPI_CR1_SPE; /* (3) */
```

A.17.6 DMA による SPI スレーブ設定のコード例

```
/* nSS ハード、スレーブ、CPOL および CPHA は 0 (最初の立ち上がりエッジ) */
/* (1) DMA による TX および RX、RXNE 割り込みを有効にし、8 ビット Rx fifo を選択します */
/* (2) SPI2 を有効にします */
SPI2->CR2 = SPI_CR2_TXDMAEN | SPI_CR2_RXDMAEN
            | SPI_CR2_RXNEIE | SPI_CR2_FRXTH
            | SPI_CR2_DS_2 | SPI_CR2_DS_1 | SPI_CR2_DS_0; /* (1) */
SPI2->CR1 |= SPI_CR1_SPE; /* (2) */
```

A.18 TSC のコード例

A.18.1 TSC 設定のコード例

```
/* TCS を設定します */
/* 約 2.5  $\mu$ s の電荷移動を使用します */
/* (1) fPGCLK = fHCLK/32 を選択します
    パルスハイを 2xtPGCLK マスターにセットします
    パルスローを 2xtPGCLK にセットします
    最大カウント値を 16383 パルスにセットします
    TSC を有効にします */
/* (2) ヒステリシスを無効にします */
/* (3) 取得完了 IT を有効にします */
/* (4) サンプリングを有効にします (G2IO4) */
/* (5) チャンネルを有効にします (G2IO3) */
/* (6) グループを有効にします (G2) */
TSC->CR = TSC_CR_PGPSC_2 | TSC_CR_PGPSC_0 | TSC_CR_CTPH_0 | TSC_CR_CTPL_0
          | TSC_CR_MCV_2 | TSC_CR_MCV_1 | TSC_CR_TSCE; /* (1) */
TSC->IOHCR &= (uint32_t)(~(TSC_IOHCR_G2_IO4 | TSC_IOHCR_G2_IO3)); /* (2) */
TSC->IER = TSC_IER_EOAIE; /* (3) */
TSC->IOSCR = TSC_IOSCR_G2_IO4; /* (4) */
TSC->IOCCR = TSC_IOCCR_G2_IO3; /* (5) */
TSC->IOGCSR |= TSC_IOGCSR_G2E; /* (5) */
```

A.18.2 TSC 割り込みのコード例

```
/* 取得完了フラグ */
if ((TSC->ISR & TSC_ISR_EOAF) == TSC_ISR_EOAF)
{
    TSC->ICR = TSC_ICR_EOAIC; /* フラグをクリアします */
    AcquisitionValue = TSC->IOGXCR[1]; /* G2 カウンタ値を取得します */
}
```


A.19 USART のコード例

A.19.1 USART トランスミッタ設定のコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 8 データビット、1 スタートビット、1 ストップビット、パリティなし */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR1 = USART_CR1_TE | USART_CR1_UE; /* (2) */
```

A.19.2 USART 送信バイトのコード例

```
/* USART 送信を開始します */
USART1->TDR = stringtosend[send++]; /* TXE がセットされている場合に TC を初期化します */
```

A.19.3 USART 転送完了のコード例

```
if ((USART1->ISR & USART_ISR_TC) == USART_ISR_TC)
{
    if (send == sizeof(stringtosend))
    {
        send=0;
        USART1->ICR |= USART_ICR_TCCF; /* 転送完了フラグをクリアします */
    }
    else
    {
        /* 転送完了フラグをクリアし、TDR に新しい文字を書き込みます */
        USART1->TDR = stringtosend[send++];
    }
}
```

A.19.4 USART レシーバ設定のコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 8 データビット、1 スタートビット、1 ストップビット、パリティなし、受信モード */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR1 = USART_CR1_RXNEIE | USART_CR1_RE | USART_CR1_UE; /* (2) */
```

A.19.5 USART 受信バイトのコード例

```
if ((USART1->ISR & USART_ISR_RXNE) == USART_ISR_RXNE)
{
    chartoreceive = (uint8_t)(USART1->RDR); /* データを受信し、フラグをクリアします */
}
```



A.19.6 USART LIN モードのコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) LIN モード */
/* (3) 8 データビット、1 スタートビット、1 ストップビット、パリティなし、送受信を有効化 */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR2 = USART_CR2_LINEN | USART_CR2_LBDIE; /* (2) */
USART1->CR1 = USART_CR1_TE | USART_CR1_RXNEIE
             | USART_CR1_RE | USART_CR1_UE; /* (3) */
/* アイドルフレームの送信をポーリングします */
while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.19.7 USART 同期モードのコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 同期モード
    CPOL および CPHA = 0 => 最初の立ち上がりエッジ
    最終ビットのクロックパルス
    送信／受信の最上位ビットファースト */
/* (3) 8 データビット、1 スタートビット、1 ストップビット、パリティなし、送信を有効化、受信を有効化 */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR2 = USART_CR2_MSBFIRST | USART_CR2_CLKEN
             | USART_CR2_LBCL; /* (2) */
USART1->CR1 = USART_CR1_TE | USART_CR1_RXNEIE
             | USART_CR1_RE | USART_CR1_UE; /* (3) */
/* クロックなしでアイドルフレームの送信をポーリングします */
while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.19.8 USART 単線半二重のコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 単線半二重モード */
/* (3) 8 データビット、1 スタートビット、1 ストップビット、パリティなし、送受信モード有効 */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR3 = USART_CR3_HDSEL; /* (2) */
USART1->CR1 = USART_CR1_TE | USART_CR1_RXNEIE
             | USART_CR1_RE | USART_CR1_UE; /* (3) */
/* アイドルフレームの送信をポーリングします */
while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.19.9 USART スマートカードモードのコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 16 分周されるクロックは、3 MHz になります */
/* (3) スマートカードモード有効 */
/* (4) 1.5 ストップビット、クロック有効 */
/* (5) 8 データビット + パリティ、1 スタートビット */
USART1->BRR = 480000 / 96; /* (1) */
USART1->GTPR = 16 >> 1; /* (2) */
USART1->CR3 = USART_CR3_SCEN; /* (3) */
USART1->CR2 = USART_CR2_STOP_1 | USART_CR2_STOP_0
             | USART_CR2_CLKEN; /* (4) */
USART1->CR1 = USART_CR1_M | USART_CR1_PCE
             | USART_CR1_TE | USART_CR1_UE; /* (5) */
/* アイドルフレームの送信の転送完了をポーリングします (このフレームは送信されない) */

while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.19.10 USART IrDA モードのコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 低電力周波数を得るために、24 分周します */
/* (3) IrDA を有効にします */
/* (4) 8 データビット、1 スタートビット、1 ストップビット、パリティなし */
USART1->BRR = 480000 / 96; /* (1) */
USART1->GTPR = 24; /* (2) */
USART1->CR3 = USART_CR3_IREN; /* (3) */
USART1->CR1 = USART_CR1_TE | USART_CR1_UE; /* (4) */
/* アイドルフレームの送信をポーリングします */
while((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.19.11 USART DMA のコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) 送受信で DMA を有効にします */
/* (3) 8 データビット、1 スタートビット、1 ストップビット、パリティなし、送受信モード有効 */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR3 = USART_CR3_DMAT | USART_CR3_DMAR; /* (2) */
USART1->CR1 = USART_CR1_TE | USART_CR1_RE | USART_CR1_UE; /* (3) */
/* アイドルフレームの送信をポーリングします */
while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.19.12 USART ハードウェアフロー制御のコード例

```
/* (1) 16 倍のオーバーサンプリング (9600 ボー) */
/* (2) RTS と CTS 有効 */
/* (3) 8 データビット、1 スタートビット、1 ストップビット、パリティなし、送受信モード有効 */
USART1->BRR = 480000 / 96; /* (1) */
USART1->CR3 = USART_CR3_RTSE | USART_CR3_CTSE; /* (2) */
USART1->CR1 = USART_CR1_TE | USART_CR1_RXNEIE
            | USART_CR1_RE | USART_CR1_UE; /* (3) */
/* アイドルフレームの送信をポーリングします */
while ((USART1->ISR & USART_ISR_TC) != USART_ISR_TC)
{
    /* 安定したアプリケーションの場合はここにタイムアウトを追加します */
}
USART1->ICR |= USART_ICR_TCCF; /* TC フラグをクリアします */
```

```
USART1->CR1 |= USART_CR1_TCIE; /* TC 割り込みを有効にします */
```

A.20 WWDG のコード例

A.20.1 WWDG 設定のコード例

```
/* (1) それぞれ約 5.5 ms ごとにロールオーバーするようにプリスケアラをセットし、  
   ウィンドウ値をセットします (約 2.25 ms) */  
/* (2) 有効にする前に WWDG をリフレッシュします */  
/* (3) WWDG を有効にします */  
WWDG->CFR = 0x60; /* (1) */  
WWDG->CR = WWDG_REFRESH; /* (2) */  
WWDG->CR |= WWDG_CR_WDGA; /* (3) */
```

改版履歴

表 142. 文書改版履歴

日付	版	変更内容
2012 年 4 月 6 日	1	初版リリース
2012 年 8 月 8 日	2	このマニュアルにおける表記の規則 セクション 1.1 : レジスタに関する略記に「常に ~ として読み出されます」を追加 メモリマップ : 表 1 : STM32F05xxx メモリマップとペリフェラルレジスタ境界アドレスにブート設定のリンクを追加 表 3 : ブートモードに脚注を追加 セクション : DMA バスで「CPU」の後ろの「DCode」を削除 セクション 2.5 : ブート設定の文章を置き換え、物理的な再割り当ての段落を追加 セクション 1.2.2 : 読み出し操作でプリフェッチバッファを訂正し、段落を追加 セクション 1.5.1 : フラッシュアクセス制御レジスタ (FLASH_ACR) から「最初の〜後にセットします」という文章を削除し、注を削除し、ビット 5 の説明を追加 CRC : セクション 6.4.3 : 制御レジスタ (CRC_CR) でビット 4:3 と 2:1 のリファレンスを置換 PWR : 図 2 および 図 3 で、「POR」、「PDR」、および「PVD」を、「VPOR」、「VPDR」、および「VPVD」と置換 セクション 1.1.1 : 独立した A/D および D/A コンバータ用電源と基準電圧の「VDDA が VDD と異なる場合〜」の段落を変更 図 2 : パワーオンリセット／パワーダウンリセット波形の矢印を移動 セクション 1.4.2 : 電源制御／ステータスレジスタ (PWR_CSR) で WKUP1 を訂正 RCC : セクション 1.1.2 : システムリセットに電源リセットを追加 セクション 1.2.2 : HSI クロックに「〜の測定方法の詳細については〜」の段落を追加 セクション 1.2.12 : クロック信号出力に箇条書き項目「LSE」と「LSI」を追加 図 2 : クロックツリー (STM32F03x および STM32F05x デバイス)に「/1」、「LSI」、および「LSE」を追加 セクション 1.2.2 : HSI クロックに「HSI クロックを〜に駆動することができます」という文章を追加 セクション 1.2.11 : 独立型ウォッチドッグクロックの見出しに「独立型」を追加 図 5 : キャプチャモードにおける TIM14 を使用した周波数測定の後ろの最初の文章を置換 セクション : タイマ 14 の入力キャプチャチャネルは GPIO ラインまたは MCU の内部クロックです。この選択は、TIM14_OR レジスタの TI1_RMP [1:0] ビットで行います。以下を使用できます。に「システムが STOP モードで〜」という文章を追加。 セクション 1.2.13 : TIM14 を使用した内部／外部クロックの測定と セクション 1.3 : 低電力モードを追加

この資料は、STMicroelectronics NV 並びにその子会社 (以下 ST) が英文で記述した資料 (以下、「正規英語版資料」) を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

表 142. 文書改版履歴（続き）

日付	版	変更内容
2012 年 8 月 8 日	2 (続き)	セクション 1.4.1 : クロック制御レジスタ (RCC_CR) でビット 19 と ビット15:8 の説明を変更 セクション 1.4.2 : クロック設定レジスタ (RCC_CFGR) の「PLLXTPRE」以降のテキストを置換し、ビット 26:24 の説明を修正 セクション 1.4.4 および セクション 1.4.5 : APB ペリフェラルリセットレジスタ 1 (RCC_APB1RSTR) のレジスタのフォーマットを変更 セクション 1.4.7 : APB ペリフェラルクロック有効レジスタ 2 (RCC_APB2ENR) のレジスタのビット 0 の見出しを変更 セクション 1.4.10 : 制御／ステータスレジスタ (RCC_CSR) で「IWWDGSTF」を「IWDGRSTF」に置換し、ビット 23 に PORV18RSTDF POR/PDR を追加 セクション 1.4.4 : APB ペリフェラルリセットレジスタ 2 (RCC_APB2RSTR) から セクション 1.4.8 : APB ペリフェラルクロック有効レジスタ 1 (RCC_APB1ENR) の見出しを変更。 GPIO : セクション 9.3.1 : 汎用 I/O (GPIO) で SWDAT を置換 セクション 9.4.3 : GPIO ポート出力スピードレジスタ (GPIOx_OSPEEDR) ($x = A..F$) に特定のリセット値を追加 DMA : 図 21 : DMA ブロック図に「I2C1」、「TIM15」、「TIM16」、および「TIM17」を追加 ADC : セクション 13.12.5 : ADC 設定レジスタ 2 (ADC_CFGR2) と セクション 13.12.11 : ADC レジスタマップの「JITOFF_D2」と「JITOFF_D4」を「JITOFF_DIV4」と「JITOFF_DIV2」に変更 セクション 13.12.11 : ADC レジスタマップで「SMPR」を「SMP」に置換 DAC : セクション 14.2 : DAC1 の主な機能の注を置換 COMP : 図 92 : コンパレータ 1 および 2 のブロック図で「bandgap」を「V_{REFINT}」に置換し、COMP2_OUT に PA12 を追加 セクション 15.3.3 : COMP のリセットおよびクロックに「リセットおよびクロックイネーブルビットは～」という文章を追加 セクション 15.6.1 : COMP 制御およびステータスレジスタ (COMP_CSR) の COMP2MODE と COMP1MODE ビットの説明を変更 TIM : セクション 1.4.3 : TIM2 および TIM3 のスレーブモード制御レジスタ (TIM2_SMCR および TIM3_SMCR) と セクション 1.4.3 : TIM1 のスレーブモード制御レジスタ (TIM1_SMCR) に OCCS ビットを追加 セクション 1.4.10 : TIM14 のキャプチャ／比較レジスタ 1 (TIM14_CCR1) でビット 0 と 1 に「TI1_RMP」を割り当て 赤外線 赤外線機能の出力を IR_OUT に変更 IWDG : 「IWWDG」を全て「IWDG」に置換 RTC : セクション 25.6.19 : RTC バックアップレジスタ (RTC_BKPxR) の V_{DD} を削除 RTC ISR レジスタと TAFCR レジスタのタンパ 3 を削除

表 142. 文書改版履歴（続き）

日付	版	変更内容
2012 年 8 月 8 日	2 (続き)	I2C : セクション 24.4.5 : I2C の初期化とセクション 24.4.6 : ソフトウェアリセットを変更 セクション 24.7.1 : 制御レジスタ 1 (I2Cx_CR1) とセクション 24.7.7 : 割り込みおよびステータスレジスタ (I2Cx_ISR) で「SWRST」を削除 USART : セクション 25.5:USART の機能詳細で「12 ビットの整数部と 4 ビットの小数部」を削除 セクション 25.5.5 : クロック偏差に対する USART レシーバの許容誤差の等式を変更 セクション 25.7 : USART レジスタでリセット値の長さを変更 表 90 : USART レジスタマップとリセット値で「DEP」、「TC」、「RDR」、および「TDR」のリセット値を訂正 DEBUG : セクション 1.9 : MCU デバッグコンポーネント (DBGMCU) の箇条書き項目「TRACE ピンの割当て制御」を削除 セクション 1.5.3 : SW-DP ステートマシン (リセット、アイドル状態、ID コード) とセクション 1.5.5 : SW-DP レジスタの IDCODE 値を置換 セクション 1.9.2 : タイマ、ウォッチドッグ、および I²C のデバッグサポートで「APB 低速」を APB1 に置換 図 1 : STM32F0xx MCU と Cortex[®]M0 レベルのデバッグサポートのブロック図を訂正 セクション 1.3.1 : SWD ポートピンの見出しを変更 セクション 1.3.2: SW-DP ピンの割り当てとセクション 1.3.3 : SWD ピンでの内部プルアップ/プルダウンを変更 セクション 1.5 : SWD ポートの見出しを変更し、「SWDAT」を置換 セクション 1.9.3: デバッグ MCU 設定レジスタ (DBGMCU_CR) の「DBG SLEEP」を削除 セクション 1.9.4 : デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ) のウィンドウ型ウォッチドッグに「システム」を追加 セクション 1.9.6 : DBG レジスタマップのビット 0、5、6、および 7 を予約済みとしてマーク ユニーク ID : セクション 32.1 : ユニークデバイス ID レジスタ (96 ビット) の「ベースアドレス」の太字を解除



表 142. 文書改版履歴（続き）

日付	版	変更内容
2013 年 5 月 10 日	3	ドキュメント全体に部品番号 STM32F06x を追加。 ドキュメントタイトルを更新。 このマニュアルにおける表記の規則 セクション 1.1 : レジスタに関する略記に「(w_r0)」を追加。 関連ドキュメントに PM0215 の参考文献を追加。 用語の更新。 システムおよびメモリの概要 : バスマトリックスと AHB-APB ブリッジのセクションを変更。 セクション 2.3 : 内蔵 SRAMに 注 : を追加。 内蔵フラッシュメモリ : 表 4 : アクセスステータス対保護レベルと実行モードを更新。 セクション 1.3.1 : 読み出し保護の レベル 2 : デバッグなしの段落の表現を変更。 セクション 1.2.1 : フラッシュメモリの構成に情報ブロックの説明を追加。 「FKEYR」を「FKEY」に、「OPTKEYR」を「OPTKEY」に名称変更。 セクション 1.5.7 : フラッシュオプションバイトレジスタ (FLASH_OBR)を名称変更し、修正。 セクション 1.5.2 : フラッシュキーレジスタ (FLASH_KEYR)、セクション 1.5.3 : フラッシュオプションキーレジスタ (FLASH_OPTKEYR)、セクション 1.5.7 : フラッシュオプションバイトレジスタ (FLASH_OBR)、および セクション 1.5.8 : 書き込み保護レジスタ (FLASH_WRPFR) のリセット値に関する情報を変更。 セクション フラッシュメモリのロック解除、メインフラッシュメモリプログラムの項目 5、ページ消去と全体消去セクションの手順を変更。 ページ消去と全体消去に 注 : を追加。 表 6 : フラッシュインタフェース - レジスタマップとリセット値を更新。 オプションバイト : セクション 2 : オプションバイトを改訂。 PWR : セクション 1.1 : 電源の概要を変更。 セクション 1.1.4 : 電圧レギュレータの箇条書き項目に「STM32F06x デバイスでは～」を追加。 セクション 外部 NPOR 信号を追加。 セクション 1.3 : 低電力モードと セクション 1.3.5 : STANDBY モードに 警告 : を追加。

表 142. 文書改版履歴（続き）

日付	版	変更内容
2013 年 5 月 10 日	3 (続き)	RCC : セクション 1.1.1 : 電源リセットとセクション 1.1.2 : システムリセットの順番を反転。 外部クリスタル／セラミック発振子 (HSE クリスタル) に警告 : を、またセクション 1.2.5 : LSE クロックに警告 : を追加。 セクション 1.2 : クロックに箇条書き項目「タイマクロック周波数は、ハードウェアによって自動的に固定されます。」を追加。 セクション 1.1.1 : 電源リセット、セクション 1.1.2 : システムリセット、セクション 1.1.3 : RTC ドメインリセット、HSI14 の較正、セクション 1.2.12 : クロック信号出力、セクション 1.2.13 : TIM14 を使用した内部／外部クロックの測定、セクション 1.4.2 : クロック設定レジスタ (RCC_CFGR)、セクション 1.4.4 : APB ペリフェラルリセットレジスタ 2 (RCC_APB2RSTR)、セクション 1.4.12 : クロック設定レジスタ 2 (RCC_CFGR2)、セクション 1.4.13 : クロック設定レジスタ 3 (RCC_CFGR3) を更新。 図 1 : 簡略化されたリセット回路図と図 2 : クロックツリー (STM32F03x および STM32F05x デバイス) を変更。 オプションバイトロードリセットの FORCE_OBL を OBL_LAUNCH に置換。 セクション 7.2.13 : TIM14 のクロック独立型システムクロックソースを削除。 表 1 : RCC レジスタマップとリセット値に USART3EN、UART4EN、および UART5EN を追加。 表 1 : RCC レジスタマップとリセット値の RCC_CSR レジスタのリセット値を変更。 セクション 1.4.10 : 制御／ステータスレジスタ (RCC_CSR) のビット 23 のリセット値と説明を変更。 セクション 1.4.13 : クロック設定レジスタ 3 (RCC_CFGR3) のビット 8 を廃止に指定。 GPIO : セクション 9.1 : 概要を更新。 図 17 : 5 V トレラント I/O ポートビットの基本構成の「アナログ」の矢印を削除。 セクション 9.4.11 : GPIO ポートビットリセットレジスタ (GPIOx_BRR) (x = A ~ F) を名称変更。 SYSCFG : レジスタビット 23、22、および 20 の値を変更。 割り込みとイベント : 表 1 : ベクタテーブルに優先順位番号を追加し、EXTI4_15 の説明を変更。 ADC : セクション 13.4.1 : 較正 (ADCAL) の一部の表現を変更。 13.5.5 : タイミング図の例 (シングル／連続モード ハードウェア／ソフトウェアトリガ) と 13.12.4 : ADC 設定レジスタ 1 (ADC_CFGR1) のセクションで AUTDLY を WAIT に置換。 図 25 : ADC ブロック図を更新。 表 44 : ADC レジスタマップとリセット値で「0x44...0x2FC」を「0x44...0x304」に置換。



表 142. 文書改版履歴（続き）

日付	版	変更内容
2013 年 5 月 10 日	3 (続き)	DAC : <i>表 66 : 外部トリガ</i>の名前を訂正。 「DAC1」をすべて「DAC」に置換。 COMP : <i>セクション 15.6.1:COMP 制御およびステータスレジスタ (COMP_CSR)</i>の「COMP1SW1: コンパレータイネーブル」を「COMP1SW1 : コンパレータ 1 非反転入力 DAC スイッチ」に置換。 TIM2 および TIM3 : <i>図 106 : 高機能制御タイマのブロック図</i>を訂正。 TIM14 : <i>図 17 : TIM16 および TIM17 ブロック図</i>を更新。 TIM15/16/17 : <i>セクション 2.1 : TIM15/16/17 の概要、セクション 2.3 : TIM16 および TIM17 の主な機能、およびセクション 2.4.4 : クロックソース</i>を更新。 <i>セクション 2.6.1 : TIM16 および TIM17 制御レジスタ 1 (TIM16_CR1 および TIM17_CR1)</i>の注を削除。 「予約済み」としてマーク: <i>セクション 2.6.3 : TIM16 および TIM17 DMA／割り込み有効レジスタ (TIM16_DIER および TIM17_DIER)</i>のビット 14 および 6、<i>セクション 2.6.4 : TIM16 および TIM17 ステータスレジスタ (TIM16_SR および TIM17_SR)</i>のビット 6 および 0、<i>セクション 2.6.5 : TIM16 および TIM17 イベント生成レジスタ (TIM16_EGR および TIM17_EGR)</i>のビット 6。 <i>表 7 : TIM16 および TIM17 のレジスタマップとリセット値</i>を更新。 RTC : <i>セクション 25 : リアルタイムクロック (RTC)</i>で、全体的に「パワーオンリセット」を「バックアップドメインリセット」に置換。 <i>セクション 25.2 : RTC の主な機能</i>の「バックアップレジスタは、タンパ検出イベント発生時にリセットされます」を削除。 <i>RTC バックアップレジスタ</i>を更新。 <i>セクション 25.3.9:RTC のリセット</i>および<i>セクション 25.6.4:RTC 初期化およびステータスレジスタ (RTC_ISR)</i>の RTC_ISR レジスタのリセット値を更新。 I2C : <i>表 81: アナログフィルタとデジタルフィルタの比較</i>のデジタルフィルタが有効な場合を更新。 <i>セクション 26.7.1 : 制御レジスタ 1 (I2Cx_CR1)</i>の WUPEN に注を追加。 <i>図 275 : N>255 バイトの場合の I2C マスタトランスミッタの転送シーケンスのフローチャート</i>を訂正。 <i>表 90 : I2C-SMBUS 仕様のデータのセットアップおよびホールド時間</i>でパラメータ「データホールド時間」の最大値を削除し、行「データ有効時間」を追加。 <i>I2C のタイミングとスレーブクロックストレッチ (NOSTRETCH = 0)</i>のサブセクションを更新。

表 142. 文書改版履歴（続き）

日付	版	変更内容
2013 年 5 月 10 日	3 (続き)	I2C（続き）： 図 257：I2C ブロック図と図 259：セットアップおよびホールドタイミングを更新。 セクション「I2C レジスタマップ」を 2.8 から 26.7.12 に再分類。 セクション 26.4.14：アドレス一致時に STOP モードからウェイクアップに「警告：STOP モードからのウェイクアップが無効な場合」を追加。 セクション 26.5：I2C 低電力モードを追加。 セクション 24.7：I2C デバッグモードを 26.4.17 に移動し、デバッグモードに名称変更。 表 93：fI2CCLK = 8 MHz でのタイミング設定の例を更新。 USART： セクション 25.7.8：割り込みおよびステータスレジスタ (USART_ISR) のビット 19 (RWU) の注を削除。 SPI/I2S： SPI 初期化とシーケンス処理のセクションを再構成。 SPI 無効化手順と DMA 処理手順を訂正して更新。 NSSP モードと TI モードの説明を更新。 DMA CRC 管理を訂正。 CRC 割り込み機能を削除。 RXONLY と BIDIMODE ビットの説明を変更。 次のセクションを変更：単方向通信 (828 ページ)とデータフレームフォーマット (833 ページ)、28.5.7：SPI を有効にする手順 (835 ページ)、図 318：ハードウェア／ソフトウェアスレープ選択管理、図 319：データクロックのタイミング図、図 320：データ長が 8 ビットまたは 16 ビットと等しくない場合のデータ配置、図 327：TI モードでの転送。 セクション 28.9.9：SPIx_I2S プリスケアラレジスタ (SPIx_I2SPR) の SPIx_I2SPR のリセット値を変更。 図 322：マスタの全二重通信、図 323：スレーブの全二重通信、図 324：CRC のあるマスタの全二重通信、図 325：パックされたモードでのマスタの全二重通信を追加。 TSC： セクション 17.6.2：TSC 割り込み有効レジスタ (TSC_IER) の「パワーオンリセット値」を「リセット値」に置換。 DEBUG： セクション 1.3.2：SW-DP ピンの割り当てを名称変更。 セクション 1.4.1：MCU デバイス ID コードのビット 11：0 の説明を変更。 表 5：SW-DP レジスタの IDCODE レジスタに関する注を変更。



表 142. 文書改版履歴（続き）

日付	版	変更内容
2013 年 5 月 21 日	4	ADC : <i>セクション 13.5.2 : プログラム可能な分解能 (RES) - 高速変換モード</i>に注を追加。 RES[1 : 0] は ADEN=0 の場合にのみ変更可能。 ビット CKMODE を追加。 <i>セクション 13.4.2 : ADC オン/オフ制御 (ADEN、ADDIS、ADRDY)</i> および <i>セクション 13.4.3 : ADC クロック (CKMODE)</i> を書き換え <i>図 28 : ADC クロック構成</i>を追加。 CKMODE ビットと JITOFF ビットに関する問題を修正。ここで、JITOFF_DIV4/DIV2 ビットを、同じ位置で、v1のコードと同じようCKMODE[1:0] ビットに置換。 <i>セクション 13.4.2 : ADC オン/オフ制御 (ADEN、ADDIS、ADRDY)</i> で ADRDY の注を変更。 <i>セクション 13.12.8 : ADC チャンネル選択レジスタ (ADC_CHSELR)</i> を拡張（チャンネル 18 を追加）。 <i>セクション 13.9 : 温度センサと内部基準電圧</i>の TSVREFE ビットへの参照を削除し、代わりに TSEN/VREFEN を参照。
2014 年 1 月 9 日	5	表紙の部品番号とページ 1 の概要を変更。 低容量を STM32F03x に名称変更 中容量を STM32F05x に名称変更 STM32F06x を STM32F0x8 に名称変更 STM32F07x マイクロコントローラ (STM32F071xB および STM32F072xx) を追加。フラッシュメモリの範囲を 128 KB に拡張。 <i>セクション 8 : クロックリカバリシステム (CRS)</i>、<i>セクション 29 : コントローラエリアネットワーク (bxCAN)</i>、および <i>セクション 30 : ユニバーサルシリアルバスフルスピードデバイスインタフェース (USB)</i> を追加。 このマニュアルにおける表記の規則 <i>セクション 1.2 : 用語</i>を更新。 システムおよびメモリの概要 : <i>図 1 : システムアーキテクチャ (全機能)</i>に「TIM7、USART3、USART4、CAN、および USB」と「7 チャンネル DMA」を追加。 <i>表 1 : STM32F0xx ペリフェラルレジスタ境界アドレス</i>に「CRS、TIM7、USART3、USART4、CAN、および USB」を追加。 <i>表 1 : STM32F0xx ペリフェラルレジスタ境界アドレス</i>を更新。 <i>表 2 : STM32F0xx メモリ境界アドレス</i>を追加。 <i>セクション 2.3 : 内蔵 SRAM</i>で STM32F07x デバイスの 16 KB-RAM を追加。 <i>内蔵ブートローダ</i>に USB を追加。 内蔵フラッシュメモリ : <i>表 2 : フラッシュメモリの構成 (STM32F07x デバイス)</i>を追加。 <i>セクション 1.5.8 : 書き込み保護レジスタ (FLASH_WRP)</i> および <i>表 6 : フラッシュインタフェース - レジスタマップとリセット値</i>に「WRP[31 : 16]」を追加。 オプションバイト : <i>表 8 : オプションバイトの構成</i>に行「0x1FFF F80C」を追加。 <i>セクション 2.1.3 : 書き込み保護オプションバイト</i>を更新。 <i>表 9 : オプションバイトマップおよび ST 出荷時の値</i>を更新。

表 142. 文書改版履歴 (続き)

日付	版	変更内容
2014 年 1 月 9 日	5 (続き)	CRC : セクション 6.2 : CRC の主な機能、セクション 6.4.3 : 制御レジスタ (CRC_CR)、およびセクション 6.4.6 : CRC レジスタマップにプログラム可能な多項式機能を導入。 セクション : 多項式のプログラミング可能性およびセクション 6.4.5 : CRC 多項式 (CRC_POL) を追加。 PWR : STM32F07x デバイス機能を追加 図 1 : 電源の概要を更新 セクション 1.1.2 : 独立 I/O 電源レールを追加 セクション 1.4.2 : 電源制御/ステータスレジスタ (PWR_CSR) にビット 10 から 15 を追加 RCC : 「バックアップドメイン」を「RTC ドメイン」に名称変更。 STM32F07x デバイスの HSI48 48 MHz RC オシレータクロックおよび USART2 機能を追加。 セクション 1.2 : クロックおよびセクション 1.2.2 : HSI クロック、セクション 1.2.4 : PLL、セクション 1.2.7 : システムクロック (SYSCLK) の選択、セクション 1.2.12 : クロック信号出力、セクション 1.3 : 低電力モードを変更。 図 3 : クロックツリー (STM32F04x および STM32F07x デバイス) およびセクション 1.2.3 : HSI48 クロックを追加。 セクション 1.4.2 : クロック設定レジスタ (RCC_CFGR)、セクション 1.4.3 : クロック割り込みレジスタ (RCC_CIR)、セクション 1.4.5 : APB ペリフェラルリセットレジスタ 1 (RCC_APB1RSTR)、セクション 1.4.6 : AHB ペリフェラルクロック有効レジスタ (RCC_AHBENR)、セクション 1.4.8 : APB ペリフェラルクロック有効レジスタ 1 (RCC_APB1ENR)、セクション 1.4.11 : AHB ペリフェラルリセットレジスタ (RCC_AHBSTR)、セクション 1.4.13 : クロック設定レジスタ 3 (RCC_CFGR3)、およびセクション 1.4.14 : クロック制御レジスタ 2 (RCC_CR2) の STM32F07x デバイスのレジスタの説明を更新し、表 1 : RCC レジスタマップとリセット値を更新。 GPIO : LCKR を除くすべてのレジスタの範囲にポート E を追加。 セクション 9.1 : 概要を変更。 図 17 : I/O ポートビットの基本構造、図 18 : 入力フローティング/プルアップ/プルダウン設定、図 19 : 出力設定、図 20 : オルタネート機能設定、および図 21 : ハイインピーダンスアナログ設定で V_{DD} を V_{DDIO} に置換。 SYSCFG : セクション 1.1.1 : SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1) で STM32F07x デバイスのビット [30 : 24] と 21 を追加。 セクション 1.1.2 : SYSCFG 外部割り込み設定レジスタ 1 (SYSCFG_EXTICR1) のビット [15 : 0] にピン PE[x] を、セクション 1.1.3 : SYSCFG 外部割り込み設定レジスタ 2 (SYSCFG_EXTICR2) のビット [15 : 0] にピン PD[x] と PE[x] を、セクション 1.1.4 : SYSCFG 外部割り込み設定レジスタ 3 (SYSCFG_EXTICR3) およびセクション 1.1.5 : SYSCFG 外部割り込み設定レジスタ 4 (SYSCFG_EXTICR4) のビット [15 : 0] にピン PD[x]、PE[x]、および PF[x] を追加。 表 1 : SYSCFG レジスタマップとリセット値を更新。

表 142. 文書改版履歴（続き）

日付	版	変更内容
2014 年 1 月 9 日	5 (続き)	DMA 5 チャンネルを最大 7 チャンネルに変更 図 21 : DMA ブロック図を更新 図 23 : DMA リクエストマッピングを更新 表 29 : STM32F03x、STM32F04x、および STM32F05x デバイスでのチャンネルごとの DMA リクエストの概要および 表 30 : STM32F07x デバイスでのチャンネルごとの DMA リクエストの概要を追加 セクション 11.5.1 : DMA 割り込みステータスレジスタ (DMA_ISR) および セクション 11.5.2 : DMA 割り込みフラグクリアレジスタ (DMA_IFCR) にビット 27 から 20 を追加 割り込みとイベント 表 1 : ベクタテーブルを更新 ADC セクション 13.9 : 温度センサと内部基準電圧を更新 DAC 2 番目のチャンネル DAC_OUT2、ノイズ波生成、三角波生成、および STM32F07x デバイスのデュアルモードを追加 セクション 14.2 : DAC1 の主な機能を更新 セクション 14.4 : デュアルモードの機能説明 (STM32F07x デバイス) を追加 セクション 14.4 : ノイズ生成を追加 セクション 14.5 : DMA リクエストを追加 セクション 14.6.1 : DAC 制御レジスタ (DAC_CR) にビット 29 から 16 およびビット 11 から 6 を追加 セクション 14.6.2 : DAC ソフトウェアトリガレジスタ (DAC_SWTRIGR) にビット 1 「SWTRIG2」を追加 セクション 14.8.6 : DAC チャンネル 2 の 12 ビット右詰めデータ保持レジスタ (DAC_DHR12R2)、セクション 14.8.7 : DAC チャンネル 2 の 12 ビット左詰めデータ保持レジスタ (DAC_DHR12L2)、セクション 14.8.8 : DAC チャンネル 2 の 8 ビット右詰めデータ保持レジスタ (DAC_DHR8R2)、セクション 14.8.9 : デュアル DAC 12 ビット右詰めデータ保持レジスタ (DAC_DHR12RD)、セクション 14.8.10 : デュアル DAC 12 ビット左詰めデータ保持レジスタ (DAC_DHR12LD)、セクション 14.8.11 : デュアル DAC 8 ビット右詰めデータ保持レジスタ (DAC_DHR8RD)、セクション 14.8.13 : DAC チャンネル 2 データ出力レジスタ (DAC_DOR2) を追加 セクション 14.6.7 : DAC ステータスレジスタ (DAC_SR) にビット 29 を追加 基本タイマ (TIM6/TIM7) セクション 1 : 基本タイマ (TIM6/TIM7) に STM32F07x デバイスの TIM7 を追加 RTC : 「バックアップドメイン」を「RTC ドメイン」に名称変更。 STM32F07x デバイスの RTC_TAMP3 およびウェイクアップ割り込みを追加 表 67 : STM32F0xx RTC の実装を追加 セクション 25.3.6 : 周期的自動ウェイクアップ、セクション : ウェイクアップタイマのプログラミングを追加 セクション 25.6.4 : RTC 初期化とステータスレジスタ (RTC_ISR) でビット SHPF の読み出しパラメータとクリアパラメータを訂正 I2C : 表 74 : STM32F0xx I2C 実装を追加 表 90 : I2C-SMBUS 仕様のデータのセットアップおよびホールド時間を更新 I2C のタイミングセクションで 50 ns を tAF(min) に、260 ns を tAF(max) に置換



表 142. 文書改版履歴（続き）

日付	版	変更内容
2014 年 1 月 9 日	5 (続き)	USART : 表 90 : STM32F0xx USART 実装に自動ボーレート検出モードの数を追加。 図 288 : USART のブロック図およびセクション 27.7.4 : ボーレートレジスタ (USARTx_BRR) から整数部と小数部に関する記述を削除。 RTS フロー制御セクションおよび CTS フロー制御セクションを RS232 RTS フロー制御および RS232 CTS フロー制御に名称変更。セクション 27.5.1 : USART キャラクタの説明にアイドルキャラクタの場合を追加。セクション : キャラクタ送信にデータ送信中の TE リセットの推奨を追加。セクション : ブレークキャラクタを更新。 セクション : 1 バイト通信で USART 無効化手順に関する注を削除。 セクション 27.5.6 : 自動ボーレート検出にオーバーサンプリングが 8 倍の場合を追加。 セクション 27.5.16 RS232 ハードウェアフロー制御および RS485 ドライバ有効を名称変更。 セクション 27.5.17 : STOP モードからのウェイクアップに、STOP モードに移行する前に進行中の転送がないことを確認するチェックを追加。セクション : STOP モードでのミュートモードの使用に RXNE がセットされる場合を追加。 表 111 : USART レジスタマップとリセット値を更新。 表 90 : STM32F0xx USART 実装を変更。 セクション 27.7.1 : 制御レジスタ 1 (USARTx_CR1) にスマートカードモードに関する注を追加。 セクション 27.7.2 : 制御レジスタ 2 (SARTx_CR2) の CLKEN の説明に、SCLK クロックをスマートカードへ送るシーケンスを追加。 セクション 27.7.5 : ガード時間およびプリスケアラレジスタ (USARTx_GTPR) に PSC 設定に関する注を追加。 7 ビットのデータワード長のサポートを追加 : セクション 27.5.1 : USART キャラクタの説明、セクション 27.5.9 : パリティ制御、セクション : 偶数パリティおよびセクション : 奇数パリティ、セクション 27.5.5 : クロック偏差に対する USART レシーバの許容誤差に 6 LSB ビットを追加し、セクション 27.7.1 : 制御レジスタ 1 (USARTx_CR1) の M ビット M0 を名称変更。 2 つの自動ボーレート検出モードを追加 : セクション 27.5.5 : クロック偏差に対する USART レシーバの許容誤差およびセクション 27.5.6 : 自動ボーレート検出にモード 2 および 3 を追加。 セクション 25.7.7 : リクエストレジスタ (USART_RQR) で、ビット 2 MMRQ の「～はRWU フラグをリセットします」を「RWU フラグをセットします」に置換。 セクション 25.5.4 : ボーレート生成に「スマートカード、LIN、および IrDA モードでは、16 倍のオーバーサンプリングのみをサポート」を追加 図 198 : ワード長のプログラミングでストップビットを訂正して更新 SPI : 表 98 : STM32F0xx SPI 実装を変更。 TSC : セクション 17 : タッチセンシングコントローラ (TSC) でグループ 7 および 8 およびチャネル 19 から 24 を追加。表 106 : 静電容量検出 GPIO を追加 DEBUG : セクション 1.4.1:MCU デバイス ID コードのビット DEV_IDx に STM32F07x ID を追加 セクション 1.9.4 : デバッグ MCU APB1 フリーズレジスタ (DBGMCU_APB1_FZ) にビット 5 「DBG_TIM7_STOP」およびビット 25 「DBG_CAN_STOP」を追加

表 142. 文書改版履歴（続き）

日付	版	変更内容
2014 年 5 月 6 日	6	表紙：STM32F042x4/x6 および STM32F048x6 デバイスを追加して更新。 MMAP： 表 2：STM32F0xx メモリ境界アドレスを更新 セクション 2.5：ブート設定を更新 内蔵フラッシュメモリ： セクション 1.2.1：フラッシュメモリの構成、表 4：アクセスステータス対保護レベルと実行モード、およびセクション 1.5：フラッシュレジスタの説明を更新 オプションバイト：セクション 2.1：オプションバイトの説明を更新 RCC： STM32F04x デバイスで更新：セクション 1.2：クロック および図 3：クロックツリー (STM32F04x および STM32F07x デバイス)。 CRS： STM32F04x デバイスを追加して更新。 GPIO：図 17：I/O ポートビットの基本構造を更新 SYSCFG：SYSCFG レジスタを更新 DMA：更新 IRTIM：図 252：TIM16 と TIM17 との IR 内部ハードウェア接続を更新。 IWDG： セクション 24.3：IWDG の機能説明を更新。手順 6 を手順 1 に移動。 セクション 24.4：IWDG レジスタを更新。注を IWDG_WINR から IWDG_SR レジスタに移動。 STM32F04x を追加： RTC： 図 256：RTC ブロック図の見出しを更新。 表 67：STM32F0xx RTC の実装を更新。 25.3 RTC の機能説明を更新。 I2C 表 74：STM32F0xx I2C 実装に STM32F04x を追加 USART： 表 90：STM32F0xx USART 実装、27.5：USART の機能説明、および 27.7：USART レジスタに STM32F04x を追加して変更。 USART IP の表 109：フレームフォーマットを更新 表 98：STM32F0xx SPI 実装に STM32F04x を追加して変更。 STM32F04x を追加して TSC IP を更新。 USB： STM32F04x および STM32F078 を追加して USB IP を変更。 USB の主な機能と USB ブロックの説明を変更。 セクション 32：デバイス電子署名を更新 セクション 1.3.10：ワンパルスモードおよびセクション 1.3.15：タイマの同期を更新。



表 142. 文書改版履歴（続き）

日付	版	変更内容
2014 年 10 月 29 日	7	適用範囲を STM32F091xB/xC デバイスおよび STM32F098xC デバイスまで拡張。 以下のセクションを更新： - 表紙 システムおよびメモリの概要 - 図 1：システムアーキテクチャ - 表 1：STM32F0xx ペリフェラルレジスタ境界アドレス - セクション 2.3：内蔵 SRAM - セクション 2.5：ブート設定 内蔵フラッシュメモリ - セクション 3.1：フラッシュの主な機能 - セクション 3.2：フラッシュメモリの機能詳細 - セクション 4.1.3：書き込み保護オプションバイト 巡回冗長検査計算ユニット（CRC） - 概要 電源制御（PWR） - 電源の概要図に関する図の脚注 1 - セクション 6.1.2：独立 I/O 電源レール - セクション 6.4.2：電源制御／ステータスレジスタ（PWR_CSR） リセットおよびクロック制御（RCC） - セクション 7.2：クロック - 図 12：クロックツリー（STM32F04x、STM32F07x、および STM32F09x デバイス） - セクション 7.2.3：HSI48 クロック - セクション 7.2.7：システムクロック（SYSCLK）の選択 - セクション 7.2.12：クロック信号出力 - セクション 7.3：低電力モード - セクション 7.4.2：クロック設定レジスタ（RCC_CFGR） - セクション 7.4.4：APB ペリフェラルリセットレジスタ 2（RCC_APB2RSTR） - セクション 7.4.5：APB ペリフェラルリセットレジスタ 1（RCC_APB1RSTR） - セクション 7.4.6：AHB ペリフェラルクロック有効レジスタ（RCC_AHBENR） - セクション 7.4.7：APB ペリフェラルクロック有効レジスタ 2（RCC_APB2ENR） - セクション 7.4.8：APB ペリフェラルクロック有効レジスタ 1（RCC_APB1ENR） - セクション 7.4.13：クロック設定レジスタ 3（RCC_CFGR3） - セクション 7.4.13：クロック設定レジスタ 3（RCC_CFGR3） クロックリカバリシステム（CRS） - 概要 汎用 I/O（GPIO） - 概要

表 142. 文書改版履歴 (続き)

日付	版	変更内容
2014 年 10 月 29 日	7 (続き)	システム設定コントローラ (SYSCFG) – セクション 10.1.1 : SYSCFG 設定レジスタ 1 (SYSCFG_CFGR1) – セクション 10.1.7 : SYSCFG 割り込みライン 0 ステータスレジスタ (SYSCFG_ITLINE0) をセクション 10.1.37 : SYSCFG 割り込みライン 30 ステータスレジスタ (SYSCFG_ITLINE30) に追加 – 表 27 : STM32F09x デバイスの SYSCFG レジスタマップとリセット値を追加 ダイレクトメモリアクセスコントローラ (DMA) – 表 31 : STM32F07x デバイスでのチャネルごとの DMA リクエストの概要 – 表 35 : DMA レジスタマップとリセット値 (レジスタは STM32F07x および STM32F09x デバイスでのみ使用可能) 割り込みとイベント – 表 37 : ベクタテーブル – セクション 12.2.5 : 外部および内部の割り込み/イベントラインの配置 – セクション 12.3.1 : 割り込みマスクレジスタ (EXTI_IMR) D/A コンバータ (DAC) – 概要 – セクション 14.2 : DAC の主な機能 – セクション 14.3.4 : DAC チャネル変換 – セクション 14.4 : デュアルモードの機能詳細 (STM32F07x および STM32F09x デバイス) – セクション 14.5 : ノイズ生成 (STM32F07x および STM32F09x デバイス) – セクション 14.6 : 三角波生成 (STM32F07x および STM32F09x デバイス) – セクション 14.8.14 : DAC ステータスレジスタ (DAC_SR) コンパレータ (COMP) – 概要 – セクション 15.1 : 概要 汎用タイマ (TIM15/16) – セクション 19.4.14 : TIM15 外部 トリガ同期 – セクション 19.4.15 : タイマ同期 (TIM15) 基本タイマ (TIM6/TIM7) – 概要 赤外線インタフェース (IRTIM) – 図 203 : TIM16 と TIM17 との IR 内部ハードウェア接続

表 142. 文書改版履歴（続き）

日付	版	変更内容
2014 年 10 月 29 日	7 (続き)	リアルタイムクロック (RTC) – 表 71 : STM32F0xx RTC implementation – 図 208 : STM32F07x および STM32F09x デバイスの RTC ブロック図 – 表 76 : 割り込み制御ビット I2C (Inter-integrated circuit) インタフェース 表 78 : STM32F0xx I2C 実装 USART (Universal synchronous asynchronous receiver transmitter) – 表 94 : STM32F0xx USART 実装 シリアルペリフェラルインタフェース/I2S (SPI/I2S) – 表 103 : STM32F0xx SPI の実装 タッチセンシングコントローラ (TSC) – 概要 コントローラエリアネットワーク (bxCAN) – 概要 HDMI-CEC コントローラ (HDMI-CEC) – 概要 デバッグサポート (DBG) セクション 32.4.1 : MCU デバイス ID コード
2015 年 7 月 29 日	8	システムおよびメモリの概要 – セクション 2.5 : ブート設定で STM32F09x デバイスの元のメモリ空間値を訂正。 CRS – セクション 8.6.2 : CRS 設定レジスタ (CRS_CFGR) で CRS 設定レジスタ (CRS_CFGR) の SYNCSRC[1:0] の説明に注を追加。 割り込みとイベント – セクション 12.2.5: 外部および内部の割り込み/イベントラインの配置で EXTI ライン 31 の定義を訂正 付録 A のコード例 – セクション 付録 A : コード例のグローバル更新 : 色で表示。

表 143. 日本語版改版履歴

日付	版	変更内容
2016 年 6 月 30 日	1	日本語版 初版リリース

索引

A

ADC_CCR	267
ADC_CFGR1	260
ADC_CFGR2	264
ADC_CHSELR	266
ADC_CR	258
ADC_DR	266
ADC_IER	256
ADC_ISR	255
ADC_SMPR	264
ADC_TR	265

C

CAN_BTR	852
CAN_ESR	851
CAN_FA1R	862
CAN_FFA1R	862
CAN_FiRx	863
CAN_FM1R	861
CAN_FMR	860
CAN_FS1R	861
CAN_IER	850
CAN_MCR	843
CAN_MSR	844
CAN_RDHxR	859
CAN_RDLxR	859
CAN_RDTxR	858
CAN_RF0R	848
CAN_RF1R	849
CAN_RlRx	857
CAN_TDHxR	856
CAN_TDLxR	856
CAN_TDTxR	855
CAN_TlRx	854
CAN_TSR	846
CEC_CFGR	913
CEC_CR	912
CEC_IER	918
CEC_ISR	916
CEC_RXDR	916
CEC_TXDR	916
COMP_CSR	300
CRC_CR	83
CRC_DR	82
CRC_IDR	83
CRC_INIT	84
CRC_POL	84

CRS_CFGR	151
CRS_CR	149
CRS_ICR	154
CRS_ISR	152

D

DAC_CR	284
DAC_DHR12L1	289
DAC_DHR12L2	290
DAC_DHR12LD	291
DAC_DHR12R1	288
DAC_DHR12R2	289
DAC_DHR12RD	291
DAC_DHR8R1	289
DAC_DHR8R2	290
DAC_DHR8RD	292
DAC_DOR1	292
DAC_DOR2	293
DAC_SR	293
DAC_SWTRIGR	288
DBGMCU_APB1_FZ	932
DBGMCU_APB2_FZ	934
DBGMCU_CR	931
DBGMCU_IDCODE	923
DMA_CCRx	210
DMA_CMARx	213
DMA_CNDTRx	212
DMA_CPARx	212
DMA_IFCR	209
DMA_ISR	208
DMA2_CCRx	210
DMA2_CMARx	213
DMA2_CNDTRx	212
DMA2_CPARx	212
DMA2_CSELR	214
DMA2_IFCR	209
DMA2_ISR	208
DMAx_CSELR	214

E

EXTI_EMR	224
EXTI_FTSR	225
EXTI_IMR	223
EXTI_PR	227
EXTI_RTSR	224
EXTI_SWIER	226



F

FLASH_ACR	68
FLASH_CR	70
FLASH_KEYR	69
FLASH_OPTKEYR	69
FLASH_SR	70
FMPI2C_ISR	672

G

GPIOx_AFRH	170
GPIOx_AFR_L	169
GPIOx_BRR	170
GPIOx_BSRR	168
GPIOx_IDR	167
GPIOx_LCKR	168
GPIOx_MODER	165
GPIOx_ODR	167
GPIOx_OSPEEDR	166
GPIOx_OTYPER	165
GPIOx_PUPDR	166

I

I2C_CR1	662
I2C_CR2	665
I2C_ICR	674
I2C_ISR	672
I2C_OAR1	668
I2C_OAR2	669
I2C_PECR	675
I2C_RXDR	676
I2C_TIMEOUTR	671
I2C_TIMINGR	670
I2C_TXDR	676
I2Cx_CR2	665
IWDG_KR	556
IWDG_PR	557
IWDG_RLR	558
IWDG_SR	559
IWDG_WINR	560

P

PWR_CR	97
PWR_CSR	98

R

RCC_AHBENR	127
RCC_AHBSTR	136
RCC_APB1ENR	130

RCC_APB1RSTR	124
RCC_APB2ENR	128
RCC_APB2RSTR	123
RCC_BDCR	133
RCC_CFGR	117
RCC_CFGR2	138
RCC_CFGR3	139
RCC_CIR	120
RCC_CR	115
RCC_CR2	140
RCC_CSR	135
RTC_ALRMAR	595
RTC_BKPxR	606
RTC_CALR	601
RTC_CR	588
RTC_DR	587
RTC_ISR	591
RTC_PRER	593
RTC_SHIFTR	597
RTC_SSR	596
RTC_TAFCR	602
RTC_TR	586
RTC_TSDR	599
RTC_TSSSR	600
RTC_TSTR	598
RTC_WPR	596
RTC_WUTR	594

S

SPIx_CR1	791
SPIx_CR2	793
SPIx_CRCPR	798
SPIx_DR	797
SPIx_I2SCFGR	799
SPIx_I2SPR	801
SPIx_RXCR	798
SPIx_SR	796
SPIx_TXCR	799
SYSCFG_CFGR1	173
SYSCFG_CFGR2	180
SYSCFG_EXTICR1	177
SYSCFG_EXTICR2	177
SYSCFG_EXTICR3	178
SYSCFG_EXTICR4	179
SYSCFG_ITLINE0	181
SYSCFG_ITLINE1	181
SYSCFG_ITLINE10	185
SYSCFG_ITLINE11	186
SYSCFG_ITLINE12	186
SYSCFG_ITLINE13	187
SYSCFG_ITLINE14	187

SYSCFG_ITLINE15	188	TIMx_CCR2	370, 438
SYSCFG_ITLINE16	188	TIMx_CCR3	370, 438
SYSCFG_ITLINE17	188	TIMx_CCR4	371, 439
SYSCFG_ITLINE18	189	TIMx_CNT	368, 436, 463, 529, 549
SYSCFG_ITLINE19	189	TIMx_CR1	350, 420, 456, 518, 546
SYSCFG_ITLINE2	182	TIMx_CR2	351, 422, 519
SYSCFG_ITLINE20	189	TIMx_DCR	373, 440, 533
SYSCFG_ITLINE21	190	TIMx_DIER	355, 426, 457, 520, 548
SYSCFG_ITLINE22	190	TIMx_DMAR	374, 441, 533
SYSCFG_ITLINE23	190	TIMx_EGR	358, 429, 459, 522, 549
SYSCFG_ITLINE24	191	TIMx_PSC	368, 436, 463, 529, 550
SYSCFG_ITLINE25	191	TIMx_RCR	369, 530
SYSCFG_ITLINE26	191	TIMx_SMCR	353, 423
SYSCFG_ITLINE27	192	TIMx_SR	357, 427, 457, 521, 549
SYSCFG_ITLINE28	192	TSC_CR	812
SYSCFG_ITLINE29	193	TSC_ICR	815
SYSCFG_ITLINE3	182	TSC_IER	815
SYSCFG_ITLINE30	193	TSC_IOASCR	818
SYSCFG_ITLINE4	183	TSC_IOCRR	819
SYSCFG_ITLINE5	183	TSC_IOPCSR	820
SYSCFG_ITLINE6	183	TSC_IOPxCR	820
SYSCFG_ITLINE7	184	TSC_IOPHCR	817
SYSCFG_ITLINE8	184	TSC_IOSCR	818
SYSCFG_ITLINE9	185	TSC_ISR	817

T

TIM15_ARR	510
TIM15_BDTR	512
TIM15_CCER	507
TIM15_CCMR1	504
TIM15_CCR1	511
TIM15_CCR2	512
TIM15_CNT	510
TIM15_CR1	496
TIM15_CR2	497
TIM15_DCR	514
TIM15_DIER	500
TIM15_DMAR	515
TIM15_EGR	503
TIM15_PSC	510
TIM15_RCR	511
TIM15_SMCR	499
TIM15_SR	501
TIM5_OR	464
TIM6_CR2	548
TIMx_ARR	368, 437, 463, 529, 550
TIMx_BDTR	371, 531
TIMx_CCER	364, 434, 462, 527
TIMx_CCMR1	360, 430, 460, 523
TIMx_CCMR2	363, 433
TIMx_CCR1	369, 437, 464, 530

U

USARTx_BRR	733
USARTx_CR1	722
USARTx_CR2	725
USARTx_CR3	729
USARTx_GTPR	733
USARTx_ICR	741
USARTx_ISR	736
USARTx_RDR	742
USARTx_RQR	735
USARTx_RTOR	734
USARTx_TDR	742
USB_ADDRn_RX	897
USB_ADDRn_TX	896
USB_BCDR	890
USB_BTABE	889
USB_CNTR	883
USB_COUNTn_RX	897
USB_COUNTn_TX	896
USB_DADDR	889
USB_EPnR	892
USB_FNR	888
USB_ISTR	885
USB_LPMCSR	890



W

WWDG_CFR 566

WWDG_CR 565

WWDG_SR 566



この資料は、STMicroelectronics NV 並びにその子会社（以下 ST）が英文で記述した資料（以下、「正規英語版資料」）を、皆様のご理解の一助として頂くために ST マイクロエレクトロニクス㈱が英文から和文へ翻訳して作成したものです。この資料は現行の正規英語版資料の近時の更新に対応していない場合があります。この資料は、あくまでも正規英語版資料をご理解頂くための補助的参考資料のみにご利用下さい。この資料で説明される製品のご検討及びご採用にあたりましては、必ず最新の正規英語版資料を事前にご確認下さい。ST 及び ST マイクロエレクトロニクス㈱は、現行の正規英語版資料の更新により製品に関する最新の情報を提供しているにも関わらず、当該英語版資料に対応した更新がなされていないこの資料の情報に基づいて発生した問題や障害などにつきましては如何なる責任も負いません。

重要なお知らせ（よくお読み下さい）

STMicroelectronics NV およびその子会社（以下、ST）は、ST製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前にST製品に関する最新の関連情報を必ず入手してください。ST 製品は、注文請書発行時点で有効なST の販売条件に従って販売されます。

ST製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関してST は一切の責任を負いません。

明示又は黙示を問わず、STは本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件でST 製品が再販された場合、その製品についてSTが与えたいかなる保証も無効となります。

ST およびST ロゴはSTMicroelectronics の商標です。その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

© 2016 STMicroelectronics - All rights reserved

