

概要

高精度なタイミングのデジタル信号を必要とする市場に対応できる STM32 マイクロコントローラが用意されています。このような市場として、デジタル電源、照明、無停電電源、ソーラ・インバータ、ワイヤレス・チャージャなどがあります。このような用途は、高分解能タイマ（HRTIM）ペリフェラルによって実現しています。これらのタイマは、最大で 12 種類の信号を生成し、制御、同期、保護の目的で多彩な入力信号を扱うことができます。モジュラ・アーキテクチャにより、変換のさまざまな接続形態と多彩なパラレル・コンバータに対応でき、実行中の再設定も可能にしています。

HRTIM は、表 1 に挙げた製品として用意されています。

このペリフェラルは、膨大な制御レジスタを備えていることから複雑に思われます。このドキュメントでは、リファレンス・マニュアルの詳しい説明の補足として、これらの製品をすぐに使用するための情報と数々のサンプルを取り上げています。

この説明書の最初のセクションでは、HRTIM プログラミングの簡潔性を示すことを目指しています。まず環境のセットアップについて説明し、続いて多数の簡単なサンプルを使用して理解を助けています。これらの基本的なケースでは、手順を追ってタイマの各種機能を紹介し、プログラミングのガイドラインを提供します。HRTIM の経験が浅い読者は、このセクションを十分に理解する必要があります。2 番目のセクションでは、新たな設計に着手する際にコンバータを設定する方法を取り上げています。このような設計では、このドキュメントにある既成のコード・サンプルを利用する場合もあれば、このドキュメントにはない接続形態を扱うときに、このドキュメントからアイデアと技巧を得る場合もあります。この説明書では、コンバータの設計（制御技法とコンポーネントの寸法設定）そのものは扱っていません。設計については、該当のアプリケーション・ノートを参照してください。

それぞれのサンプルには、コンバータの簡単な説明、制御波形、コードが付随しています。これらのコード（および STM32 の HAL ライブラリや LL ライブラリを使用して作成した同等のコード）は www.st.com からダウンロードできます。

表 1. 対象とする製品

タイプ	部品番号または製品ライン
マイクロ コントローラ	STM32F334C4、STM32F334C6、STM32F334C8、STM32F334K4、STM32F334K6、STM32F334K8、STM32F334R4、STM32F334R6、STM32F334R8
	STM32H742、STM32H743/753、STM32H745/755 および STM32H747/757 の各ライン、STM32H750IB、STM32H750VB、STM32H750XB、STM32H753AI、STM32H753BI、STM32H753II、STM32H753VI、STM32H753XI、STM32H753ZI、STM32H755BI、STM32H755II、STM32H755XI、STM32H755ZI、STM32H757AI、STM32H757BI、STM32H757II、STM32H757XI、STM32H757ZI
	STM32G474CB、STM32G474CC、STM32G474CE、STM32G474MB、STM32G474MC、STM32G474ME、STM32G474QB、STM32G474QC、STM32G474QE、STM32G474RB、STM32G474RC、STM32G474RE、STM32G474VB、STM32G474VC、STM32G474VE、STM32G484CE、STM32G484ME、STM32G484QE、STM32G484RE、STM32G484VE

目次

1	準備	6
1.1	前提条件	6
1.2	ハードウェアのセットアップ	6
1.3	ツールのセットアップ	7
1.4	HRTIM のバージョン	7
1.5	マイクロコントローラと HRTIM のセットアップ	8
1.5.1	システム・クロックの初期化	8
1.5.2	HRTIM の初期化	8
1.5.3	HRTIM の DLL 初期化	9
1.5.4	HRTIM の I/O 初期化	9
1.5.5	他のペリフェラルの初期化	10
1.5.6	HRTIM 機能の確認	11
2	HRTIM の基本的な動作原理	12
2.1	単独の PWM の生成	12
2.2	複数の PWM の生成	14
2.3	他のタイミング・ユニットとマスタ・タイマの使用による PWM の生成	17
2.4	任意の波形の生成	19
3	電圧モード・デュアルバックコンバータ	21
4	同期整流とフォールト保護を備えた電圧モードバックコンバータ	24
5	非反転バック - ブーストコンバータ	26
6	遷移モード力率コントローラ	29
7	多相降圧コンバータ	33
7.1	ADC の動作のデバッグ	38
8	デッドタイムが挿入されないサイクル単位の保護	42
9	他のサンプル	45

10	ソフトウェア・サンプルの入手先	46
付録 A	HRTIM v2.	48
A.1	機能	48
A.1.1	6 番目のタイミング・ユニット (タイマ F)	48
A.1.2	デュアル・チャネル DAC トリガ	48
A.1.3	外部イベントのカウンタ	48
A.1.4	拡張フォールト機能	48
A.1.5	プッシュブルの改善	48
A.1.6	クラシック PWM モード	49
A.1.7	アップ / ダウン・モード	49
A.1.8	ADC ポスト・スケーラ	49
A.1.9	ヌル・デューティ・サイクル・モード	49
A.1.10	新しいインターリーブ・モード	50
A.1.11	スワップ・モード	50
付録 B	ソフトウェアの移行.	51
B.1	HRTIM v1.0 から HRTIM v1.1 への移行	51
B.2	HRTIMv1 から HRTIMv2 への移行	51
付録 C	参考文献.	54
C.1	STM32F334x.	54
C.2	STM32H74x / STM32H75x	54
C.3	STM32G47x	54
C.4	その他	55
	改版履歴	56

表の一覧

表 1.	対象とする製品	1
表 2.	組込み先の製品に応じた HRTIM の各種機能	7
表 3.	HRTIM の入力周波数動作範囲	8
表 4.	タイマ分解能 / 最小 PWM 周波数 (fHRTIM = 144 MHz の場合)	14
表 5.	タイマ分解能 / 最小 PWM 周波数 (fHRTIM = 170 MHz の場合)	15
表 6.	サンプリング・トリガと Data[0..4] の RAM テーブルに保存される結果	37
表 7.	想定される変換結果	39
表 8.	サンプルの概要	46
表 9.	イベントの対応表	51
表 10.	タイマごとのウィンドウイング信号のマッピング (EEFLTR[3:0] = 1111)	52
表 11.	タイマごとのフィルタリング信号のマッピング	52
表 12.	HRTIM ADC のトリガ 1 レジスタとトリガ 3 レジスタ (HRTIM_ADC1R、HRTIM_ADC3R)	53
表 13.	HRTIM ADC のトリガ 2 レジスタとトリガ 4 レジスタ (HRTIM_ADC2R、HRTIM_ADC4R)	53
表 14.	文書改版履歴	56
表 15.	日本語版文書改版履歴	56

図の一覧

図 1.	基本的な PWM の生成.....	13
図 2.	基本的な PWM 信号を生成する HRTIM 設定.....	13
図 3.	複数の PWM 信号の生成.....	16
図 4.	マスタ・タイマによる PWM の生成.....	18
図 5.	任意の波形の生成.....	19
図 6.	電圧 - モードのバックコンバータ.....	21
図 7.	電圧モードの降圧波形、ADC のサンプリングと割込み.....	22
図 8.	PWM の割込みとレジスタの更新.....	22
図 9.	同期整流を備えた電圧モードバック.....	24
図 10.	FAULT に伴うバック動作.....	25
図 11.	非反転バック - ブーストコンバータ.....	26
図 12.	バック - ブースト動作モード.....	26
図 13.	降圧昇圧コンバータの動作波形.....	27
図 14.	遷移モード PFC.....	29
図 15.	過電流状態の Ton max における遷移モード PFC の動作.....	30
図 16.	Toff max と Toff min における遷移モード PFC の動作.....	31
図 17.	5 相インターリーブ降圧コンバータ.....	33
図 18.	5 相インターリーブ降圧コンバータの制御波形.....	34
図 19.	マスタ・タイマによる位相の無効化.....	35
図 20.	位相シェディングとバースト・モードによる低負荷管理.....	36
図 21.	低負荷降圧コンバータ.....	36
図 22.	アクティブな位相の数に応じたサンプリング・ポイントの配置.....	38
図 23.	ADC のサンプリング・ポイントの確実な正しい配置.....	39
図 24.	ADC のサンプリング (5 相設定).....	40
図 25.	ADC のサンプリング (3 相設定).....	40
図 26.	電流モードの降圧コンバータ.....	42
図 27.	過電流保護スキーム.....	42
図 28.	デッドタイムを使用しない過電流保護.....	43
図 29.	デュアル外部イベントによる複数サイクル過負荷保護.....	44
図 30.	単一外部イベントによる複数サイクル過負荷保護.....	44
図 31.	スロープ補正によるピーク電流モードの制御.....	48
図 32.	アップのみのモードとアップ / ダウン・モードの比較.....	49

1 準備

このアプリケーション・ノートで取り上げるマイクロコントローラは Arm[®] コア^(a) を基本としています。

このセクションでは、HRTIM プログラミングに集中できるように、その着手前に必要な準備作業について詳しく説明します。

HRTIM とこのドキュメントで使用する各種ペリフェラルに関連するドキュメントを [付録 C : 参考文献](#) に一覧で挙げています。製品のリファレンス・マニュアルで、HRTIM に関連するセクションに事前に目を通しておくことをお勧めします。

1.1 前提条件

本格的に HRTIM に取り組む前に、対応が必要な前提条件があります。読者には基本的な C プログラミングのスキル、マイクロコントローラと開発環境を扱った最小限の経験があることが前提となるほか、スイッチング・モード電源の理論的な知識が求められます。制御方針とコンポーネントの寸法設定の詳細はこのアプリケーション・ノートの取り扱い範囲外です。関連資料を参照してください。

この説明書では、簡潔な説明を目的として、電圧レベルに関係なく、マイクロコントローラで直接扱うことができる論理信号とアナログ電圧のみを考慮します。ただし、外部コンポーネントを扱う機能や電源のスイッチングによる副作用を扱う機能をタイマやマイクロコントローラが備えている場合は、これらに対する何らかの基準を設定します。

最後に、感電、火傷、死亡のリスクを排除するために、十分なスキルを持つ技術担当者が電力アプリケーションを運用すること、危険な電圧レベルで動作するアプリケーションではマイクロコントローラを使用することを厳守します。

1.2 ハードウェアのセットアップ

STM32F334 ディスカバリ・キット（注文コード STM32F3348-DISCO）は極めて手頃な価格のツールでありながら、HRTIM の実験的な使用を開始し、それを継続するうえで最適です。このキットはプログラミング・インタフェースを備えているので、チップのプログラミングとデバッグのために別途用意する必要がある部材は USB ケーブルのみです。すべての I/O は 2.54 mm 間隔のピンで用意されているので、各種ユニバーサル基板にも接続できます。また、電力コンバータとして、LED 駆動用の反転ブースト型バックコンバータと、独立した入出力を持つ低電圧バック/ブーストコンバータの 2 基を備えています。

オシロスコープは必須です。特に、4 チャネル以上の監視を必要とする設定の際にロジック・アナライザと組み合わせて使用します。わかりにくい高分解能の手順を観測するために、オシロスコープには 1 GS/s 以上のサンプリング・レートが必要なほか、高いタイミング精度を得るためにインターリーブ・サンプリングのオプションが求められます。



a. Arm は、米国内およびその他の地域における Arm Limited 社（またはその子会社）の登録商標です。

初期のデバッグ段階で電力コンバータからのフィードバック（論理パルスまたはアナログ信号）をエミュレートするために、1 台以上のファンクション・ジェネレータがあると効果的です。状況によっては、このジェネレータにトリガ入力が必要です。トリガ入力がない場合は、フリー・タイミング・ユニットを使用して HRTIM 自体でフィードバック信号をエミュレートすることもできます。ただし、この場合はコーディング作業（またはソフトウェア・サンプルの再使用）が多くなります。

1.3 ツールのセットアップ

インストール済みのコンパイラ（どのデモも 32 KB に収まっています）、および ST-LINK-V2/ST-LINK-V3 デバッグ・インタフェースに対応した IDE が必要です。

以降で紹介する各コードはコンパイラに依存しないので、どのようなツールチェーンでも汎用の HRTIM プロジェクト・テンプレートにコピーするだけで使用できます。

ソフトウェアのソースは、次のツールチェーンのワークスペースに用意されています。

- IAR™ (EWARM 7.10.3)
- Keil® (MDK-ARM 4.7)
- STM32 向け System Workbench (SW4STM32)

1.4 HRTIM のバージョン

HRTIM のペリフェラルは、その最初の導入以来、機能向上が進んでいます。現在、表 2 に挙げた製品に組み込まれています。必要に応じて、そのリビジョンは HRTIMv1 や HRTIMv2、または簡略化して v1 や v2 と表記されます。

表 2. 組み込み先の製品に応じた HRTIM の各種機能

品名	HRTIM のリビジョン	参照	主な変更点
STM32F334xx	V1.0	HRTIMv1	-
STM32H74xxx STM32H75xxx	V1.1		DLL なし
STM32G47xxx STM32G48xxx	V2.0	HRTIMv2	- 6 番目のタイマと 6 番目の FAULT 入力の追加 - 新しい動作モード 詳細については、付録 A : HRTIM v2 を参照してください。

付録 B : ソフトウェアの移行では他の製品に切り替える際に、HRTIM のリビジョンをどのように扱うかについて説明しています。

1.5 マイクロコントローラと HRTIM のセットアップ

1.5.1 システム・クロックの初期化

高い分解能を得るには、PLL の高周波出力を HRTIM に直接供給する必要があります。DLL があれば、その DLL でクロック周期を 32 分周して高分解能を実現できます。

これにより、HRTIM のクロック周波数が 32 倍に高くなったような効果が得られます。

クリスタルによるハイスピード外部（HSE）オシレータまたはハイスピード内部（HSI）オシレータを使用して 2 つのオプションを利用できます。

製品ごとの許容周波数範囲を表 3 に示します。

表 3. HRTIM の入力周波数動作範囲

品名	PLL 入力	入力周波数 (MHz)		HRTIM 周波数 (GHz)		分解能 (ps)	
		最小値	最大値	最小値	最大値	最小値	最大値
STM32F334xx	HSE	128	144	4.096	4.608	217	244
	HSI	128		4.096			
STM32H74xxx STM32H75xxx	任意	-	480	-	0.480	2.08 ns (DLL なし)	-
STM32G47xxx STM32G48xxx		100	170	3.20	5.44	184	312

HAL ライブラリを初期化 (HAL_Init) した後、ただちにメイン・ルーチンで *SystemClock_Config()* 関数を使用してクロックを初期化します。

CPU のクロックも、PLL の 2 分周によって得られるので、その最高周波数は PLL の出力周波数の 1/2 です。高分解能での動作を維持したまま、マイクロコントローラの使用量を削減するために CPU のクロック周波数を低くすることもできます。

メイン・ルーチンで *SystemCoreClockUpdate* 関数を実行して CPU の動作周波数を確認できます。この周波数で *SystemCoreClock* 変数の値が更新されます。

1.5.2 HRTIM の初期化

このセクションでは、HRTIM を初期化する方法を手順を追って詳しく説明するほか、個々の関数呼び出しも取り上げます。この初期化を実際に行うのは *HAL_HRTIM_Init* ルーチンと *HAL_HRTIM_MspInit* ルーチンです。

HRTIM のクロック初期化

マイクロコントローラが動作している状態で、プログラミングの前に HRTIM にクロックを供給する必要があります。RCC（リセットおよびクロック・コントローラ）ペリフェラルを使用して次の 2 つの手順で実行します。

1. HRTIM に使用するハイスピード PLL を RCC_CFGR3 レジスタで選択します。

```
__HAL_RCC_HRTIM1_CONFIG(RCC_HRTIM1CLK_PLLCLK);
```

2. APB2 バスにマッピングしたレジスタに対してクロックを有効にします。

```
__HRTIM1_CLK_ENABLE();
```


1.5.3 HRTIM の DLL 初期化

HRTIM の DLL（遅延ロック・ループ）はきめ細かいタイミングを提供し、高周波クロック周期を等間隔で 32 分周します。

高分解能を使用できるようにするには、この DLL を 1 回以上較正する必要があります。電圧や温度の条件が変化したときは、HRTIM が動作中でもソフトウェアで較正を透過的に再実行できます。ハードウェアによる周期的較正も可能です。

以下のコードは STM32F334 を対象とした較正の方法を示しています。DLLRDY フラグを設定済みであれば高分解能を使用できます。

```
/* DLL の較正：周期的較正を有効化、周期を 14 マイクロ秒に設定 */
HRTIM1->sCommonRegs.DLLCR = HRTIM_CALIBRATIONRATE_14 | HRTIM_DLLCR_CALEN;
/* DLL 較正終了フラグの確認 */
while (HRTIM1->sCommonRegs.ISR & HRTIM_IT_DLLRDY == RESET);
```

最小の較正周期 ($2048 \times t_{\text{HRTIM}}$) をデフォルトの条件として、周期的較正を有効にしておくことをお勧めします。

注： DLL がロックされていないと（多くは HSE オシレータを適切に設定していない場合）、このコードの実行が停止します。HAL ライブラリには較正を実行する関数が用意されています。この関数は、タイムアウト確認機能を備え、必要に応じてエラー・ハンドラを伴って実行をリダイレクトします。これは、HAL を使用するソフトウェア・サンプルで使用されている関数です。

1.5.4 HRTIM の I/O 初期化

HRTIM の入出力は標準の I/O ポートにマッピングされるので、他の任意の I/O ペリフェラルとしてプログラムする必要があります。HRTIM のオルタネート機能は次の 2 つに分けて考えることができます。

- STM32F334 製品と STM32G4 製品向けの AF3 と AF13 の各オルタネート機能
- STM32H7 製品向けの AF1、AF2、AF3 の各オルタネート機能

HRTIM の I/O 初期化は 2 段階で実行します。HRTIM の各レジスタを初期化する前に、まず HAL_HRTIM_MspInit 関数で HRTIM の入力を初期化します。

HRTIM の制御レジスタをプログラミングした後（ここのサンプルでは *GPIO_HRTIM_outputs_Config* 関数で実行）、カウンタを有効にしてから HRTIM の出力を初期化します。この手順は、GPIO 回路から HRTIM に制御を渡す前に、HRTIM 内部で出力の状態を確実に正しく定義しておくことを目的としています。

1.5.5 他のペリフェラルの初期化

HRTIM は、以下に挙げるような多数のマイクロコントローラのペリフェラルと相互作用します。HRTIM が動作するうえで、これらのすべてを初期化する必要はありません。後ほど説明するサンプルのいくつかには、以下のペリフェラルを初期化するコードが用意されています。

ネスト化されたベクタ割込みコントローラ (NVIC)

HRTIM の割込みリクエストは、7 つ (HRTIMv1 の場合) または 8 つ (HRTIMv2 の場合) の割込みベクタにグループ化されています。すべてのフォールトは、高い優先度を設定できるベクタにグループ化されます。

NVIC のうち、HRTIM に関連する部分は `HAL_HRTIM_MspInit` 関数にプログラミングされています。

DMA コントローラ

大半の割込みリクエストは DMA リクエストとして使用でき、6 つ (HRTIMv1 の場合) または 7 つ (HRTIMv2 の場合) の DMA チャンネルにグループ化されています (マスタ・タイマも含めたタイミング・ユニットごとに 1 チャンネル)。

DMA を使用する HRTIM 動作は、`HAL_HRTIM_WaveformCounterStart_DMA` などの固有の開始と停止の関数により、タイマの開始時に有効になります。

詳細については、[セクション 2](#) を参照してください。

コンパレータ

内蔵コンパレータを使用してアナログ信号の条件を設定できます。これらのコンパレータは、その出力を HRTIM に送る前に初期化が必要です。

この初期化では、アナログ入力のプログラミング、クロックの有効化、極性の設定などを実行します。

オペアンプ

内蔵オペアンプでは、ADC またはコンパレータに送る低電圧信号を増幅できます。コンパレータ同様、オペアンプも初期化する必要があります。

ADC

HRTIM では、あらゆる ADC をトリガできます。各 ADC は、そのレギュラ・シーケンス上またはインジェクト・シーケンス上で、外部トリガを受け取ることができるように初期化する必要があります。

ADC で考えられる用途として、外部イベント (出力のセット/リセットやカウンタのリセットなどを目的とするイベント) を HRTIM 上でトリガするアナログ・ウォッチドッグがあります。

DAC

DAC を使用してコンパレータの閾値を定義します。HRTIM の DAC トリガを使用することで、HRTIM の動作に同期して DAC を更新できます。

汎用タイマ

以下の用途で、他のオンチップ・タイマに HRTIM をリンクすることもできます。

- 外部イベント
- バースト・モード・トリガまたはクロック
- HRTIM レジスタを更新するトリガ

1.5.6 HRTIM 機能の確認

すべての初期化が完了すると、以下の簡単なコードを使用して、HRTIM の準備が整っているかどうかを確認できます。このサンプル・コード (HRTIM_BasicPWM のサンプル) は、HRTIM の TD1 出力を有効にして、その出力をソフトウェアで切り替えます。

```
/* HRTIM に PLLx2 クロックを使用 */
__HAL_RCC_HRTIM1_CONFIG(RCC_HRTIM1CLK_PLLCLK);
/* HRTIM のクロックを有効化 */
__HRTIM1_CLK_ENABLE();
/* DLL の較正：周期的較正を有効化、周期を 14 マイクロ秒に設定 */
HRTIM1->sCommonRegs.DLLCR = HRTIM_CALIBRATIONRATE_14 | HRTIM_DLLCR_CALEN;
/* DLL 較正終了フラグの確認 */
while(HRTIM1->sCommonRegs.ISR & HRTIM_IT_DLLRDY == RESET);

HRTIM1->sCommonRegs.OENR = HRTIM_OENR_TD1OEN; /* TD1 出力を有効化 */
GPIO_HRTIM_outputs_Config(); /* HRTIM 出力を初期化 */

while(1)
{
    /* ソフトウェアで TD1 をセットおよびリセット */
    HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx1R = HRTIM_SET1R_SST;
    HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx1R = HRTIM_RST1R_SRT;
}
```

ここに掲載した STM32F334 のコードは、HRTIM_Snippets のサンプルと HRTIM_BasicPWM のサンプルにあります。どちらの場合も、`#define HRTIM_CHECK` ステートメントでサンプルを選択する必要があります。

以降のこのドキュメントでは、クロックと DLL を初期化するコード部分は再掲せず、`HRTIM_Minimal_Config()` 関数の呼び出しで置き換えます。

2 HRTIM の基本的な動作原理

HRTIM には、多数の機能とそのモジュラ・アーキテクチャに起因する複雑さがあることは明らかです。しかし、基本的には、4 つの比較レジスタごとに 6 つまたは 7 つの 16 bit 自動再ロード・アップカウンタで構成されています。

周期と比較のプログラミング（144 MHz 入力クロックのサンプル）

高分解能プログラミングは全面的に透過になっています。これにより、HSE オシレータを使用したときに、4.6 GHz（144 MHz x 32）のクロックで動作するタイマのような操作性が得られます。独自の 16 bit レジスタに高分解能の精度でタイミング（周期と比較）を直接書き込むことができます。カウント周期は、式 $PER = T_{counting} / T_{High-res}$ を使用してプログラミングします。

たとえば、周期値レジスタを $10 \mu s / 217 ps = 46082d$ に設定することで $10 \mu s$ の時間周期が得られます。これは次のようにプログラミングします。

```
HRTIM1->TimerxRegs[HRTIM_TIMERINDEX_TIMER_D].PERxR = 0x0000B400;
```

この周期値の結果が 16 bit の範囲に収まらない場合は、16 bit の範囲に収まるように 217 ps の倍数で高分解能を調整できます。

セット/リセット・クロスバー

各タイミング・ユニットは、セット/リセット・クロスバーを通じて 2 つの出力を制御します。通常の固定された PWM モードでは、カウント周期の開始時点で出力がセットされ、指定の比較で一致があると出力がリセットされます。それと比べ、クロスバーでは出力のセットとリセットの定義がはるかに柔軟です。出力をセットまたはリセットする任意のタイマ・イベントを使用できます。

出力ステージ

セット/リセット・クロスバーで生成された波形は、最終的に出力ステージを通過して、次のような"後処理"に送られます。

- デッドタイムを伴う相補信号の生成
- 高周波変調の追加
- 信号極性の変更
- 保護を目的とした出力のシャット・ダウン

これらの特徴を念頭に置くと、最初の要素的な PWM 信号を作成できるようになります。

2.1 単独の PWM の生成

このセッションでは以下の点に重点を置きます。

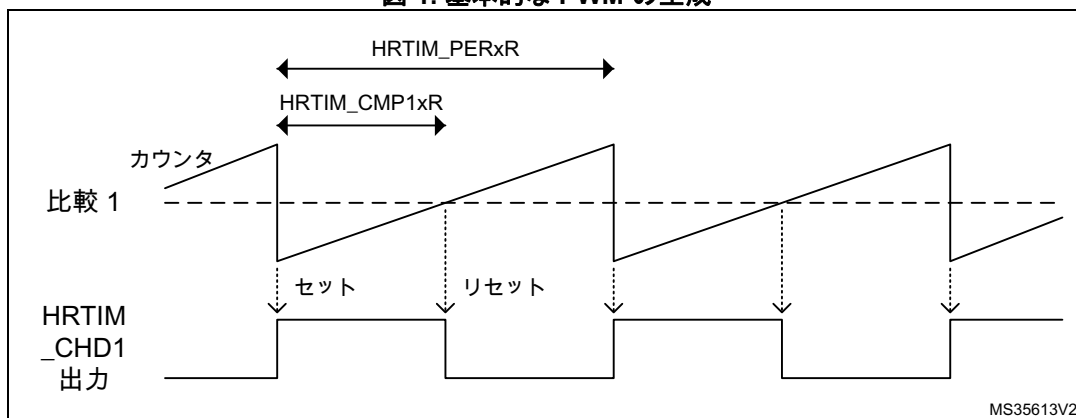
- タイマの連続モード
- 最も簡潔なクロスバーの設定
- 出力ステージの有効化

PWM 信号はほとんどの電力コンバータで基本的な構成要素であり、電動モータの駆動、圧電ブザーの駆動、DAC のエミュレーションなどの多くの目的で使われます。

以下のサンプルでは、限られた数の HRTIM レジスタをプログラミングすることにより、HRTIM を使用してこのような目的を極めて簡潔に達成できることを示します。

 1 のように、デューティ・サイクルが 50% の 100 kHz PWM 信号を HRTIM の CHA1 出力に生成するとします。

図 1. 基本的な PWM の生成



タイマ D を連続（フリーラン）モードに設定する必要があります。式 $PER = f_{HRCK} / f_{PWM}$ を使用して周期値レジスタ HRTIM_PERAR に PWM 周期をプログラミングします。

この式の結果は、 $144 \text{ MHz} \times 32 / 100 \text{ kHz} = 46080d$ (0xB400) となります。

この周期にデューティ・サイクルの 50% を乗算する式 $PER \times DC$ からパルス幅が得られます。

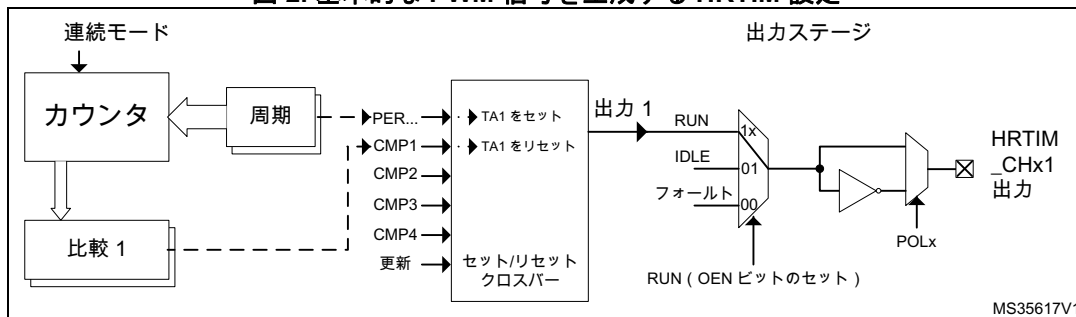
この式の結果は、 $0.5 \times 46080d = 23040d$ (0x5A00) となります。

PER ビットを設定した HRTIM_SETx1R レジスタと CMP1 ビットを設定した HRTIM_RSTx1R レジスタを使用して、セット/リセット・クロスバーで波形が作成されます。

最終的に HRTIM_OENR レジスタで出力が有効になります。

このシーケンスは、簡潔な PWM 生成に使用するタイマ機能の概要を紹介したものです。この設定を図 2 に示します。

図 2. 基本的な PWM 信号を生成する HRTIM 設定



サンプル・コードが HRTIM_BasicPWM のサンプルにあります。また、以下に示すコードが HRTIM_Snippets のサンプルにあります。どちらの場合も、`#define SINGLE_PWM` ステートメントでサンプルを選択する必要があります。

```

/* 連続モードで動作する TIMD カウンタ */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].TIMxCR = HRTIM_TIMCR_CONT;

/* 周期を 100 kHz、デューティ・サイクル (CMP1) を 50% に設定 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].PERxR = 0x0000B400;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP1xR = 0x00005A00;

/* TIMD 周期値で TD1 出力をセットし、TIMD CMP1 イベントで TD1 出力をリセット */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx1R = HRTIM_SET1R_PER;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx1R = HRTIM_RST1R_CMP1;

HRTIM1->sMasterRegs.MCR = HRTIM_MCR_TDCEN; /* タイマ D を開始 */
HRTIM1->sCommonRegs.OENR = HRTIM_OENR_TD1OEN; /* TD1 出力を有効化 */
GPIO_HRTIM_outputs_Config(); /* HRTIM GPIO 出力を初期化 */

```

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

2.2 複数の PWM の生成

このセクションでは以下の点に重点を置きます。

- 複数のタイミング・ユニットの使用
- レジスタのプリロード

HRTIM では、5 種類の周波数を使用して最大で 10 個の PWM 信号を生成できます（マスタ・タイマを使用する場合は 6 種類の周波数が可能。[セクション 2.3](#)を参照）。

以下のサンプルでは、2 種類のタイムベースを使用して 4 つの PWM 信号を生成しています。

デューティ・サイクルを 25% とした 2 つの 100 kHz 位相シフト信号を、HRTIM_CHD1 上と HRTIM_CHD2 上にタイマ D で生成します。条件は以下のとおりです。

- HRTIM_CHD1 : TD 周期値でセット、TD CMP1 でリセット
- HRTIM_CHD2 : TD CMP2 でセット、TD 周期値でリセット

デューティ・サイクルを 25% とした 2 つの 33.333 kHz 位相シフト信号を、HRTIM_CHA1 上と HRTIM_CHA2 上にタイマ A で生成します。条件は以下のとおりです。

- HRTIM_CHA1 : TA 周期値でセット、TA CMP1 でリセット
- HRTIM_CHA2 : TA CMP2 でセット、TA CMP3 でリセット

[表 4](#) と [表 5](#) に示すように、タイマ A の周期は、最高分解能で可能な最小周波数よりも短くなります。

表 4. タイマ分解能/最小 PWM 周波数 ($f_{HRTIM} = 144 \text{ MHz}$ の場合)

CKPSC[2:0]	プリスケール比	f_{HRCK} 等価周波数	分解能	最小 PWM 周波数
000	1	$144 \times 32 \text{ MHz} = 4.608 \text{ GHz}$	217 ps	70.3 kHz
001	2	$144 \times 16 \text{ MHz} = 2.304 \text{ GHz}$	434 ps	35.1 kHz
010	4	$144 \times 8 \text{ MHz} = 1.152 \text{ GHz}$	868 ps	17.6 kHz
011	8	$144 \times 4 \text{ MHz} = 576 \text{ MHz}$	1.73 ns	8.8 kHz

表 4. タイマ分解能/最小 PWM 周波数 ($f_{\text{HRTIM}} = 144 \text{ MHz}$ の場合) (続き)

CKPSC[2:0]	プリスケール比	f_{HRCK} 等価周波数	分解能	最小 PWM 周波数
100	16	$144 \times 2 \text{ MHz} = 288 \text{ MHz}$	3.47 ns	4.4 kHz
101	32	144 MHz	6.95 ns	2.2 kHz
110	64	$144/2 \text{ MHz} = 72 \text{ MHz}$	13.88 ns	1.1 kHz
111	128	$144/4 \text{ MHz} = 36 \text{ MHz}$	27.7 ns	550 Hz

表 5. タイマ分解能/最小 PWM 周波数 ($f_{\text{HRTIM}} = 170 \text{ MHz}$ の場合)

CKPSC[2:0]	プリスケール比	f_{HRCK} 等価周波数	分解能	最小 PWM 周波数
000	1	$170 \times 32 \text{ MHz} = 5.44 \text{ GHz}$	184 ps	83 kHz
001	2	$170 \times 16 \text{ MHz} = 2.72 \text{ GHz}$	368 ps	41.5 kHz
010	4	$170 \times 8 \text{ MHz} = 1.36 \text{ GHz}$	735 ps	20.86 kHz
011	8	$170 \times 4 \text{ MHz} = 680 \text{ MHz}$	1.47 ns	10.4 kHz
100	16	$170 \times 2 \text{ MHz} = 340 \text{ MHz}$	2.94 ns	5.194 kHz
101	32	170 MHz	5.88 ns	2.59 kHz
110	64	$170/2 \text{ MHz} = 85 \text{ MHz}$	11.76 ns	1.3 kHz
111	128	$170/4 \text{ MHz} = 42.5 \text{ MHz}$	23.53 ns	650 Hz

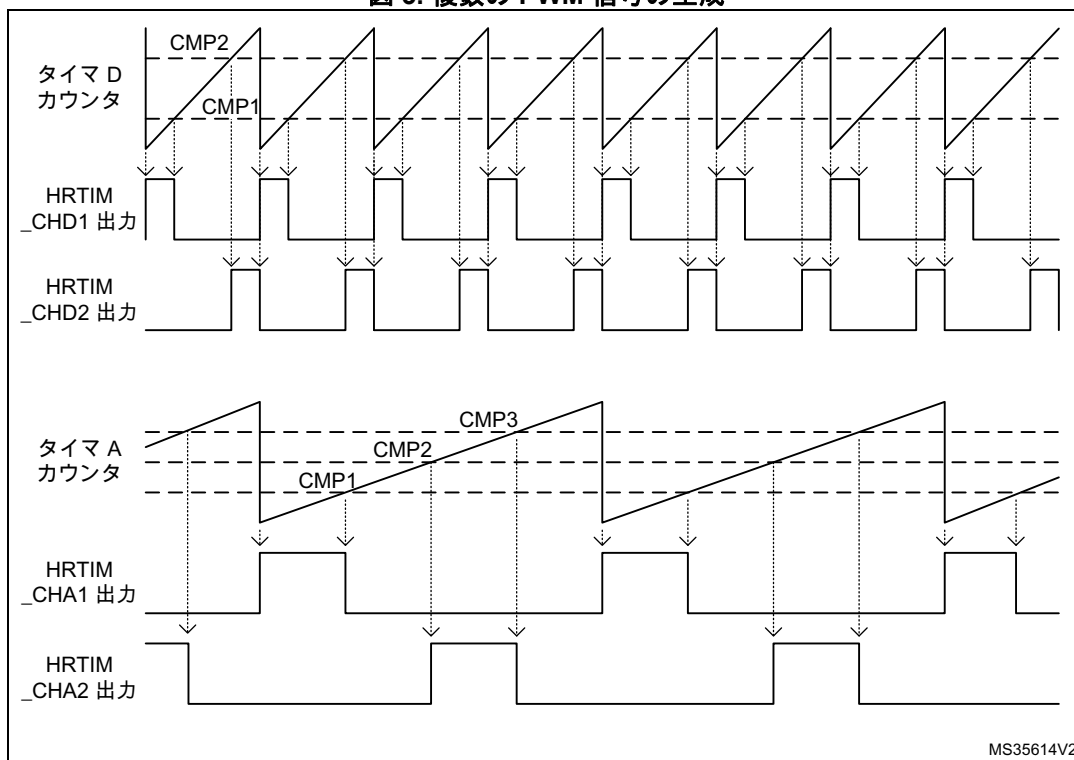
144 MHz の入力クロックでスイッチング周波数を 33.33 kHz にするには、周波数プリスケアラを 4 に設定し、周期値を $\text{PER} = f_{\text{HRCK}}/f_{\text{PWM}}$ で計算する必要があります。

33.333 kHz の周波数は、PER レジスタを $(144 \text{ MHz} \times 8)/33.333 \text{ kHz} = 34560d$ (0x8700) に設定することで得られます。

このサンプルではデューティ・サイクルが更新されませんが、現実的なユース・ケースに近い状態にするには、PREEN ビットを使用してレジスタのプリロード・メカニズムを有効にします。REP (繰り返し) イベントによって、すべての周期のそれぞれの開始時にプリロード・レジスタの転送がトリガされます。

注： HRTIM_CHA1/HRTIM_CHA2 波形と HRTIM_CHD1/HRTIM_CHD2 波形の間で PWM の開始時に遅延が見られることがあります。この遅延は正常であり、最初のカウント周期が経過した後でのみ最初の更新イベントが発生することに起因しています。この更新イベントが発生すると、比較レジスタはプログラムされた値を取得します、この遅延がないまま波形を開始すると、アクティブな比較レジスタすべての内容を一度に更新することを目的として、ソフトウェアによって強制的にレジスタが更新されることが考えられます (TASWU ビットと TDSWU ビットを使用)。

図 3. 複数の PWM 信号の生成



MS35614V2

サンプル・コードが HRTIM_BasicPWM のサンプルにあります。また、以下に示すコードが HRTIM_Snippets のサンプルにあります。どちらの場合も、`#define MULTIPLE_PWM` ステートメントでサンプルを選択する必要があります。

```

/* ----- タイマ D の初期化 ----- */
/* TIMD カウンタを連続モードで動作している状態にして、REP イベントでプリロードを有効化 */
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].TIMxCR =
HRTIM_TIMCR_CONT + HRTIM_TIMCR_PREEN + HRTIM_TIMCR_TREPU;

/* 周期を 100 kHz、CMP1 を周期の 25%、CMP2 を周期の 75% にそれぞれ設定 */
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].PERxR = _100KHz_PERIOD;
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP1xR = _100KHz_PERIOD/4;
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP2xR =
(3*_100KHz_PERIOD)/4;

/* TIMD 周期で TD1 出力をセットし、TIMD CMP1 イベントで TD1 出力をリセット */
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx1R = HRTIM_SET1R_PER;
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx1R = HRTIM_RST1R_CMP1;
/* TD2 出力を TIMD CMP2 でセットし、TIMD 周期イベントでリセット */
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx2R = HRTIM_SET2R_CMP2;
HRTIM1->stimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx2R = HRTIM_RST2R_PER;

```



```

/* ----- タイマ A の初期化 ----- */
/* プリスケアラを 010b (4 で除算) として、TIMA カウンタを連続モードで動作 */
/* REP イベントでプリロードを有効化 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].TIMxCR = HRTIM_TIMCR_CONT
+ HRTIM_TIMCR_PREEN + HRTIM_TIMCR_TREPU + HRTIM_TIMCR_CK_PSC_1;

/* 周期を 33 kHz、デューティ・サイクルを 25% に設定 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].PERxR = _33KHz_PERIOD;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].CMP1xR = _33KHz_PERIOD/4;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].CMP2xR = _33KHz_PERIOD/2;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].CMP3xR =
(3*_33KHz_PERIOD)/4;

/* TIMA 周期で TA1 出力をセットし、TIMA CMP1 イベントで TA1 出力をリセット */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].SETx1R = HRTIM_SET1R_PER;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].RSTx1R = HRTIM_RST1R_CMP1;
/* TA2 出力を TIMA CMP2 でセットし、TIMA 周期イベントでリセット */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].SETx2R = HRTIM_SET2R_CMP2;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_A].RSTx2R = HRTIM_RST2R_CMP3;

/* TA1、TA2、TD1、TD2 の各出力を有効化 */
HRTIM1->sCommonRegs.OENR = HRTIM_OENR_TA1OEN + HRTIM_OENR_TA2OEN +
HRTIM_OENR_TD1OEN + HRTIM_OENR_TD2OEN;
GPIO_HRTIM_outputs_Config(); /* HRTIM GPIO 出力を初期化 */

/* タイマ A とタイマ D を開始 */
HRTIM1->sMasterRegs.MCR = HRTIM_MCR_TACEN + HRTIM_MCR_TDCEN;

```

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

2.3 他のタイミング・ユニットとマスタ・タイマの使用による PWM の生成

このセクションでは、指定のタイマに関連していない出力での信号生成に重点を置きます。

このサンプルでは、セット / リセット・クロスバーを使用することによって、用意されている他の任意のタイマによって出力に PWM 信号（または他の波形）を生成できることを示します。この方法は次の場合に効果的です。

- 次のサンプルのように、マスタ・タイマを使用して、PWM に依存しない 6 番目の周波数を生成する。
- ピン配置の制約に対処する。たとえば、HRTIM_CHE1 出力と HRTIM_CHE2 出力がない場合（ピン数が少ないパッケージに多く見られます）でも、タイマ E を使用して波形を生成します。

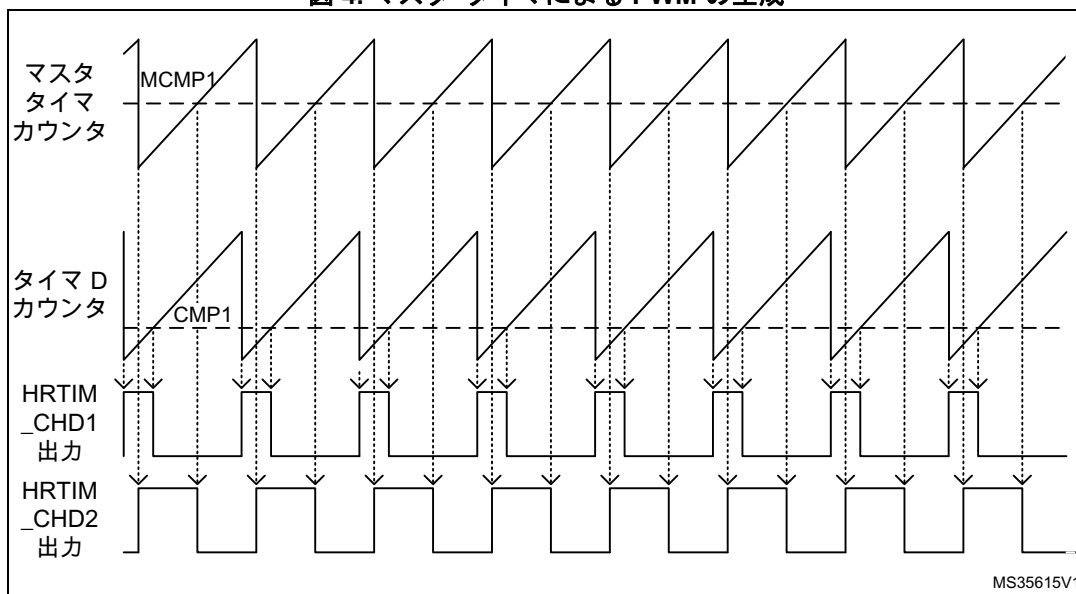
注： 同じリソースを使用しているすべてのタイマで同じプリスケアラ係数を使用する必要があります。たとえば、マスタ・タイマで HRTIM_CHA1 出力または HRTIM_CHA2 出力を制御している場合は、マスタ・タイマとタイマ A に同じ CKPSC[2:0] 値を使用します。

以下のサンプルでは、スイッチング周波数がわずかに異なる 2 つの PWM 信号を HRTIM_CHD1 出力と HRTIM_CHD2 出力に生成します。条件は次のとおりです。

- HRTIM_CHD1 : TD 周期値でセット、TD CMP1 でリセット
- HRTIM_CHD2 : マスタ周期値でセット、マスタ CMP1 でリセット

これらの信号の周波数は、わずかに異なる値にします（信号には比較的ゆっくりした位相シフトの変化が現れます）。これは、オシロスコープ上で波形を見やすくするため、また信号が全的に非同期であることを確認するためです。

図 4. マスタ・タイマによる PWM の生成



以下のコードは HRTIM_BasicPWM のサンプルです。また、以下に示すコードが HRTIM_Snippets のサンプルにあります。どちらの場合も、`#define PWM_MASTER` ステートメントでサンプルを選択する必要があります。

```
/* ----- マスタ・タイマの初期化 ----- */
/* マスタ・カウンタを連続モードで動作している状態にして、REP イベントでプリロードを有効化 */
HRTIM1->sMasterRegs.MCR = HRTIM_MCR_CONT + HRTIM_MCR_PREEN +
HRTIM_MCR_MREPU;

/* 周期を 101 kHz、デューティ・サイクルを 50% に設定 */
HRTIM1->sMasterRegs.MPER = _100KHz_Plus_PERIOD;
HRTIM1->sMasterRegs.MCMP1R = _100KHz_Plus_PERIOD/2;

/* ----- タイマ D の初期化 ----- */
/* TIMD カウンタを連続モードで動作している状態にして、REP イベントでプリロードを有効化 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].TIMxCR =
HRTIM_TIMCR_CONT + HRTIM_TIMCR_PREEN + HRTIM_TIMCR_TREPU;

/* 周期を 100 kHz、デューティ・サイクル (CMP1) を 50% に設定 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].PERxR = _100KHz_PERIOD;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP1xR = _100KHz_PERIOD/4;
```

```

/* TIMD 周期で TD1 出力をセットし、TIMD CMP1 イベントで TD1 出力をリセット */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx1R = HRTIM_SET1R_PER;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx1R = HRTIM_RST1R_CMP1;

/* TD2 出力を TIMD CMP2 でセットし、TIMD 周期イベントでリセット */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx2R =
HRTIM_SET2R_MSTPER;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx2R =
HRTIM_RST2R_MSTCMP1;

/* TD1 出力と TD2 出力を有効化 */
HRTIM1->sCommonRegs.OENR = HRTIM_OENR_TD1OEN + HRTIM_OENR_TD2OEN;
GPIO_HRTIM_outputs_Config(); /* HRTIM GPIO 出力を初期化 */

/* マスタ・タイマとタイマ D を開始 */
HRTIM1->sMasterRegs.MCR |= HRTIM_MCR_MCEN + HRTIM_MCR_TDCEN;

```

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

2.4 任意の波形の生成

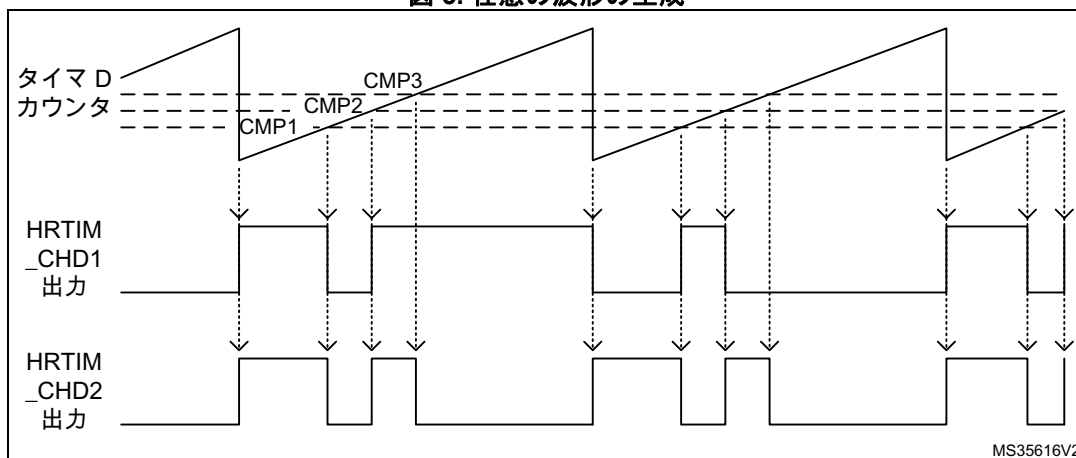
このセクションでは、1 周期あたりで複数のセット/リセット/トグルのリクエストを持つ波形に重点を置きます。

以下のサンプルでは、クロスバースのセット/リセット・ソースを同時に 32 個使用することにより、PWM 以外の波形を容易に生成できることを示します。

このサンプルでは、2 つの任意の波形を HRTIM_CHD1 出力と HRTIM_CHD2 出力に生成します。条件は次のとおりです。

- HRTIM_CHD1 : TD 周期値でトグル、TD CMP1 でトグル、TD CMP2 でトグル
- HRTIM_CHD2 : TD 周期値と TD CMP3 でセット、TD CMP2 と TD CMP3 でリセット

図 5. 任意の波形の生成



以下のコードは HRTIM_BasicPWM のサンプルです。また、以下に示すコードが HRTIM_Snippets のサンプルにあります。どちらの場合も、`#define ARBITRARY_WAVEFORM` ステートメントでサンプルを選択する必要があります。

```
/* ----- タイマ D の初期化 ----- */
/* TIMD カウンタを連続モードで動作している状態にして、REP イベントでプリロードを有効化 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].TIMxCR = HRTIM_TIMCR_CONT
+ HRTIM_TIMCR_PREEN + HRTIM_TIMCR_TREPU;

/* 周期を 100 kHz とエッジ・タイミングに設定 */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].PERxR = _100KHz_PERIOD;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP1xR = _100KHz_PERIOD/4;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP2xR =
(3*_100KHz_PERIOD)/8;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].CMP3xR = _100KHz_PERIOD/2;

/* TIMD 周期値、CMP1、CMP2 のイベントで TD1 をトグル */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx1R = HRTIM_SET1R_PER +
HRTIM_SET1R_CMP1 + HRTIM_SET1R_CMP2;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx1R = HRTIM_RST1R_PER +
HRTIM_RST1R_CMP1 + HRTIM_RST1R_CMP2;

/* TD2 出力を TIMD PER と CMP2 のイベントでセット、TIMD CMP1 と CMP3 のイベント
でリセット */
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].SETx2R = HRTIM_SET2R_PER +
HRTIM_SET2R_CMP2;
HRTIM1->sTimerxRegs[HRTIM_TIMERINDEX_TIMER_D].RSTx2R = HRTIM_RST2R_CMP1 +
HRTIM_RST2R_CMP3;

/* TD1 出力と TD2 出力を有効化 */
HRTIM1->sCommonRegs.OENR = HRTIM_OENR_TD1OEN + HRTIM_OENR_TD2OEN;
GPIO_HRTIM_outputs_Config(); /* HRTIM GPIO 出力を初期化 */

/* タイマ D を開始 */
HRTIM1->sMasterRegs.MCR = HRTIM_MCR_TDCEN;
```

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

3 電圧モード・デュアルバックコンバータ

このセクションでは以下の点に重点を置きます。

- 繰り返しカウンタの割込み
- ADC トリガ
- 繰り返しイベントでのレジスタの更新

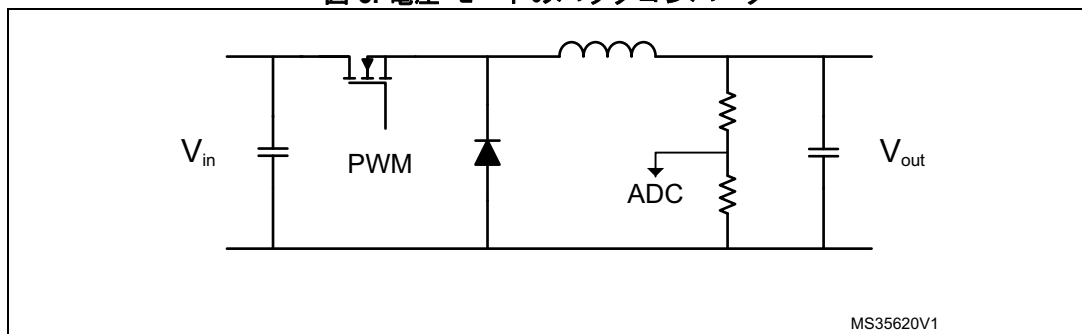
ほとんどの場合、降圧コンバータは電圧のステップダウン変換に使用します。一定の周波数で動作し、デューティ・サイクルは可変です。

理想的なコンバータでは、比率 V_{in}/V_{out} はデューティ・サイクル D のみで決まり、 $V_{out} = D \times V_{in}$ となります。

実際には、目的の出力電圧が維持されるように制御アルゴリズムでデューティ・サイクルを調整します。以下のサンプルでは、電圧モード降圧コンバータを考えます。コンバータ出力の電圧読み取りに基づいてデューティ・サイクルを制御します。入力電圧の読み取りを追加してフィードフォワード補正を実装できます。

電力ステージの接続形態を図 6 に示します。ここでは、電圧の安定化を目的として ADC で出力電圧を読み取っています。

図 6. 電圧-モードのバックコンバータ



ソフトウェア・パッケージ (HRTIM_DualBuck) とともに用意されているサンプルでは、並列動作する 2 つの降圧コンバータ (同じ周波数で動作し、オン時間が重ならないコンバータ) を制御するように HRTIM がプログラミングされています。

HRTIM を連続モード動作として、PWM 信号を次のように定義します。

- HRTIM_CHD1 : TD 周期値でセット、TD CMP1 でリセット
- HRTIM_CHD2 : TD CMP2 でセット、TD 周期値でリセット

HRTIM CHD2 は[セクション 2.2](#)の説明にあるように生成されるので、以下の図には示されていません。

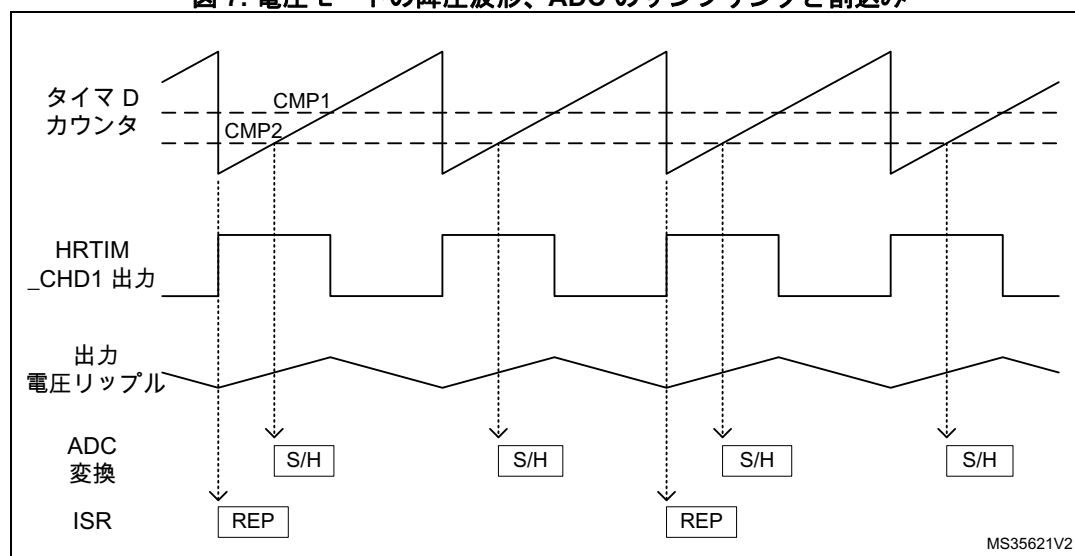
このサンプルでは、他の 2 つの比較イベントを使用して、ADC をトリガします。

HRTIM_CHD1 と HRTIM_CHD2 で、スイッチ・オン時間の中央で CMP3 と CMP4 が発生するようにそのタイミングが計算されます。これによって平均電圧を測定できるほか、電力スイッチングに起因するリンギングとスイッチング・ノイズが発生しないタイミングで確実に変換が実行されます。

カウンタの動作、生成された PWM 出力、出力電圧に発生したリップルの拡大表示を図 7 に示します。HRTIM_CHD1 チャンネルの ADC サンプルング・ポイントも表示されています。PWM 周波数が高いことから、繰り返しカウンタを使用して割込みサービス・ルーチンの数を削減しています。図 7 では、

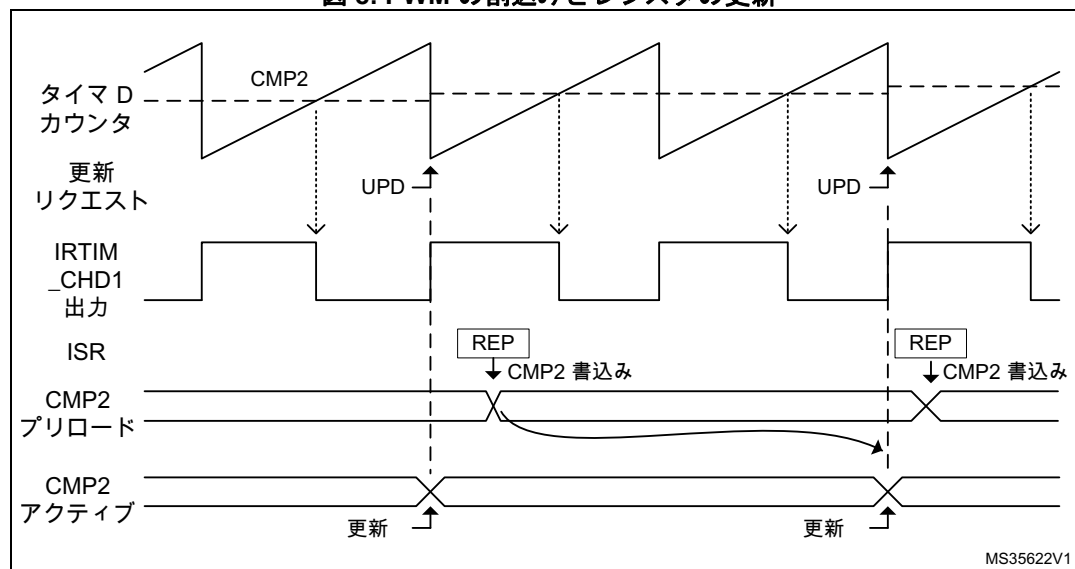
ISR レートが PWM 周波数の 1/2 になるように、繰り返しの自動再ロード・レートを 1 に設定しています。このサンプルでは、繰り返しレートを 127 に設定して変更レートを極めて低くしています。

図 7. 電圧モードの降圧波形、ADC のサンプリングと割込み



HRTIM_CHD1 におけるデューティ・サイクルの調整が割込みルーチンで完了した後のデューティ・サイクルの更新を図 8 に示します。繰り返しイベントのたびにレジスタが更新されるようにタイマがプログラミングされています。

図 8. PWM の割込みとレジスタの更新



デュアルバックデモの概要

HRTIM_CHD1 におけるデューティ・サイクルは、繰り返しイベントで生成される繰り返しサービス・ルーチンで絶えず変更されます。これによって、実際のコンバータ管理を模擬的に再現します。PWM のデューティ・サイクルを反映したローパス・フィルタ処理信号は、STM32F334 ディスカバリ・キットの TP3 テスト・ポイントで監視できます。

HRTIM_CHD2 におけるデューティ・サイクルは一定です。

各出力に現れるオン時間の中央で変換がトリガされるように ADC が設定されています。この 2 つの変換は同じ ADC インジェクト・トリガで実行されます（2 つの信号が重ならないので、このような実行が可能です）。ADC は不連続インジェクト・モードでプログラミングされています（一方の変換は CMP3 でトリガされ、他方の変換は CMP4 でトリガされます）。

この 2 つの変換は、同じ入力（PA4）に対して実行されます。HRTIM_CHD1 のローパス・フィルタ処理した PWM を PA4 に接続し、デバッグで ADC 変換を監視できます。実際のユース・ケースでは、出力電圧に降圧が発生するたびに変換が実行されるように ADC を容易に再プログラミングできます。

HRTIM_FLT1 入力は PA12（アクティブ・ロー）で有効になり、PWM がシャット・ダウンすることが示されます（詳細については[セクション 4](#)を参照）。フォールトがトリガされると（PA12 入力が GND に接続された状態）、HRTIM_CHD1 信号のみが停止します。ユーザ・ボタンを押すとシステムが再設定されます。

LED によって以下の状態が示されます。

- 青：正常動作中に点滅
- 赤：FAULT がトリガされると点滅
- オレンジ：PWM の更新 ISR の発生と期間を指示

デモ・コード

HRTIM_DualBuck のサンプルにサンプル・コードが用意されています。

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

4 同期整流とフォールト保護を備えた電圧モードバックコンバータ

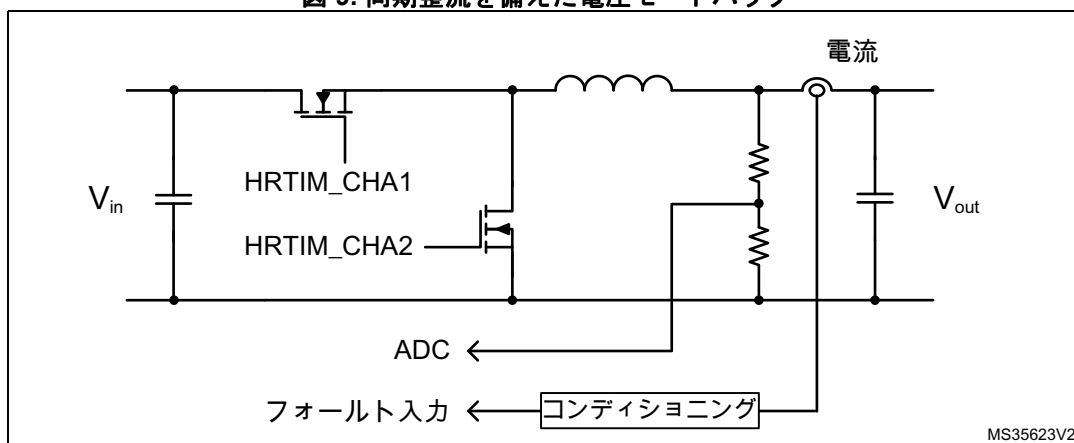
このセクションでは以下の点に重点を置きます。

- デッド・タイム
- デジタル FAULT 保護
- バースト・モード・コントローラ

以下のサンプルでは、同期整流を実現するためにバイパス・ダイオードを MOSFET に置き換えた電圧モードバックコンバータを取り上げます。出力コンデンサでインダクタンスを消磁するときに損失が低減されることで、コンバータの効率向上が可能です。

電力ステージの接続形態を図 9 に示します。このステージは、出力電圧の読み取りと、FAULT 入力を使用した過電流保護を備えています。プログラミング可能な閾値を超える電流が発生するとコンバータがシャット・ダウンします。簡潔にするために、電流センサとコンディショニング回路については説明しません。HRTIM_FLT1 入力に想定される FAULT フィードバックは、PA12 入力のデジタル信号です。

図 9. 同期整流を備えた電圧モードバック



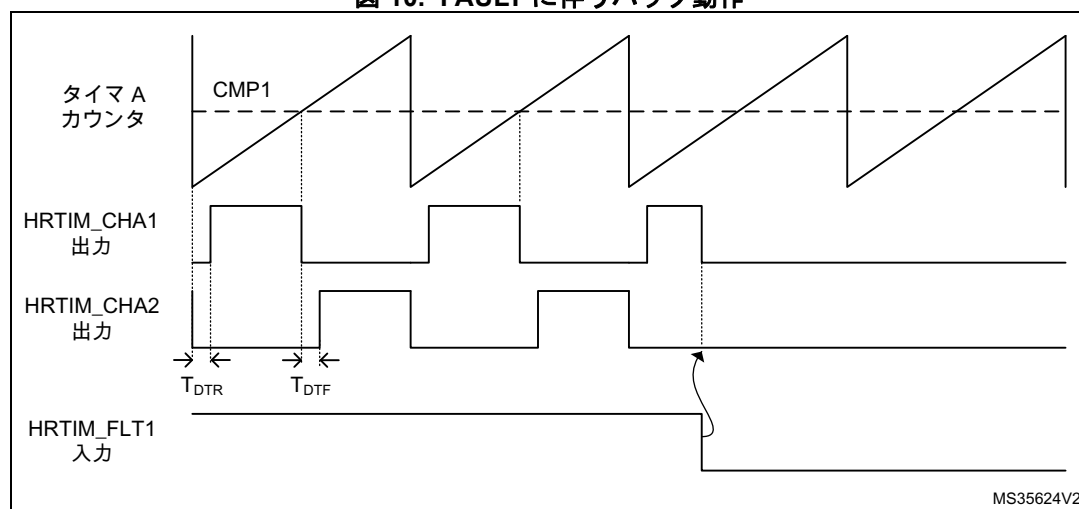
HRTIM を連続モード動作にして、PWM 信号を次のように定義します。

- HRTIM_CHA1 : TA 周期値でセット、TA CMP2 でリセット
- HRTIM_CHA2 : デッド・タイム・ジェネレータを備え、HRTIM_CHA1 に対して相補的に動作（立ち上がりエッジと立ち下がりエッジのデッド・タイムが同じ）

このコンバータは、HRTIM_FLT1 デジタル入力によってロー・レベル検知で過電流保護されています。正出力極性と非アクティブ・フォールト状態のプログラミングにより、HRTIM_CHA1 出力と HRTIM_CHA2 出力は HRTIM_FLT1 イベントで強制的にロー・レベルになります。

HRTIM_CHA1 出力と HRTIM_CHA2 出力の降圧制御波形を図 10 に示します。フォールトが発生した状況も示しています。

図 10. FAULT に伴うバック動作



同期整流を備えたバックのデモンストレーションの概要

HRTIM_CHA1 におけるデューティ・サイクルは、繰り返しイベントで生成される繰り返しサービス・ルーチンで絶えず変更されます。これによって、実際のコンバータ管理を模擬的に再現します。HRTIM_FLT1 入力は PA12 (アクティブ・ロー) で有効になり、HRTIM_CHA1 と HRTIM_CHA2 の両方で PWM がシャット・ダウンすることが示されます (ロー・レベル検知)。

フォールトがトリガされると (PA12 入力が GND に接続された状態)、HRTIM_CHA1 信号と HRTIM_CHA2 信号がシャット・ダウンされます。ユーザ・ボタンを押すとシステムが再設定されます。

LED によって以下の状態が示されます。

- 青：正常動作中に点滅
- 赤：FAULT がトリガされると点滅
- オレンジ：PWM 更新 ISR の発生と期間を指示

PA1 (V_{in}) 入力と PA3 (V_{out}) 入みに現れるコンバータ・オン時間の中央で変換がトリガされるように ADC が設定されています。このデモを実行するには、バック・ブーストコンバータの V_{out} 入力を 5V_O 電源に接続する必要があります。得られた電圧は V_{out} ピンに出力されます。

注： ディスカバリ・キットのブーストステージを省略するには、PA11 ピンを強制的に 1 にする必要があります (T6 PMOS をオンに切り替えます)。

デモ・コード

HRTIM_BuckSyncRect のサンプルにサンプル・コードが用意されています。

ターゲットのボードとファームウェアへのアクセス・パスについては [セクション 10](#) を参照してください。

5 非反転バック-ブーストコンバータ

このセクションでは以下の点に重点を置きます。

- 4つのスイッチによるコンバータ駆動
- 0%と100%のPWM生成

このサンプルでは、非反転バック-ブーストコンバータを駆動するように HRTIM を設定し、動作している状態で降圧モードから昇圧モードに切り替える方法を示します。この設定では、従来の反転バック-ブーストコンバータよりも多くのスイッチが必要になりますが、グランド基準の正出力電圧が得られる利点があります。

STM32F334 ディスカバリ・キットに実装した電力ステージの接続形態を図 11 に示します。

図 11. 非反転バック-ブーストコンバータ

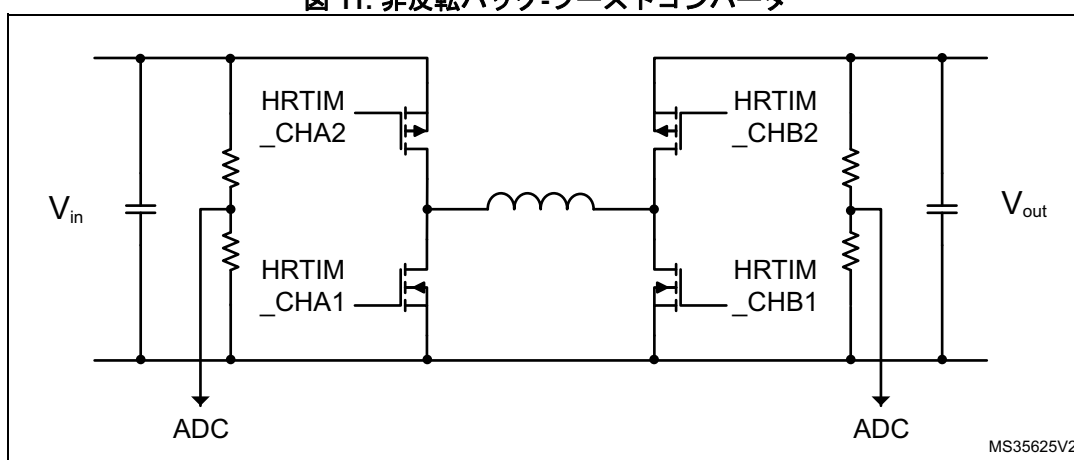
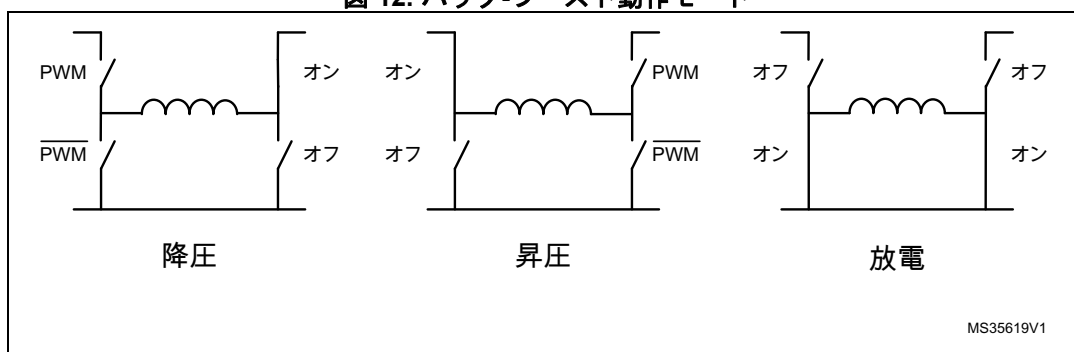


図 12 に、このデモの 3 種類の動作モードを簡略化した回路図で示します（MOSFET のアンチパラレル・ダイオードは省略しています）。バックモードとブーストモードの両方に同期整流方式を適用します（MOSFET の内部ダイオードによる導電損失を低減するために相補型 MOSFET を駆動しています）。コンバータがブーストモードからバックモードに切り替わる時に電流が入力ソースに逆流しないように、3 番目のモードとして "放電" が必要です。出力コンデンサが放電してその電圧が入力電圧を下回るまで、マイクロコントローラによってこのモードが維持されます。

ハーフ・ブリッジを駆動する相補出力を強制的に 0% または 100% の PWM にすることで、一定のオン/オフ状態になります。

図 12. バック-ブースト動作モード



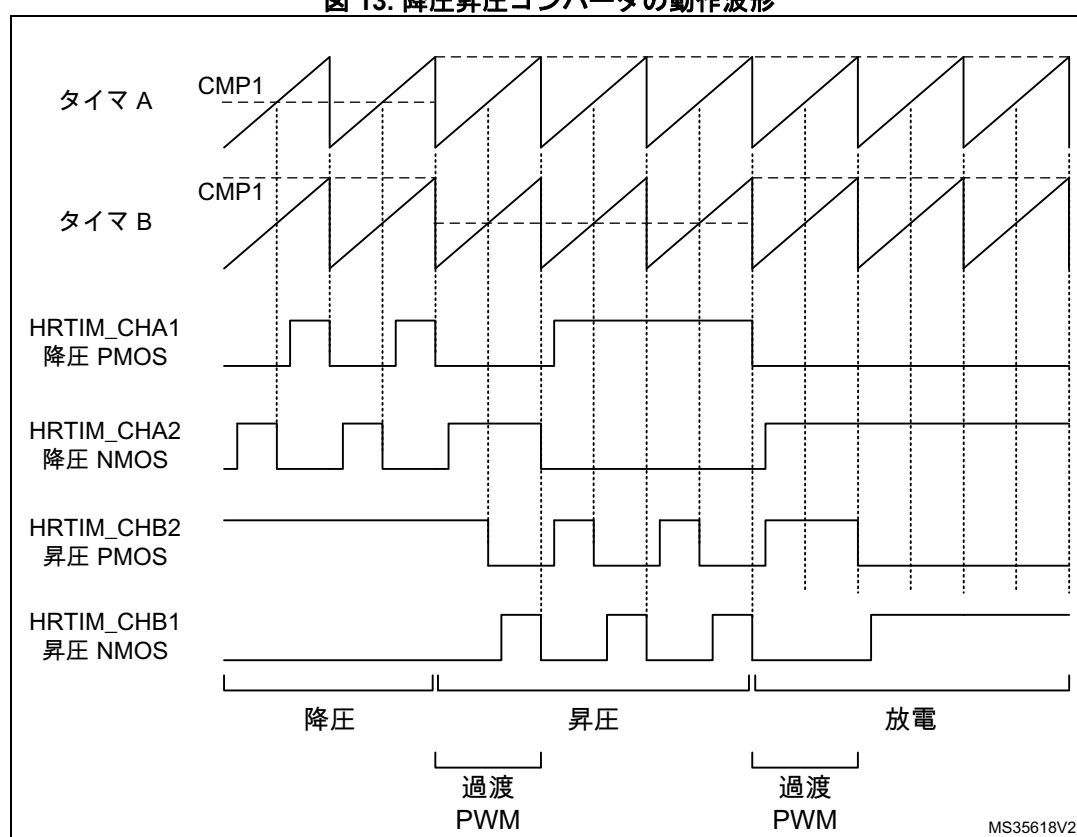
HRTIM を連続モード動作にして、PWM 信号を次のように定義します。

- HRTIM_CHA1 : TA CMP1 でセット、TA 周期値でリセット
- HRTIM_CHA2 : デッド・タイム・ジェネレータを備え、HRTIM_CHA1 に対して相補的に動作（立ち上がりエッジと立ち下がりエッジのデッド・タイムが同じ）
- HRTIM_CHB1 : TB CMP1 でセット、TB 周期値でリセット
- HRTIM_CHB2 : デッド・タイム・ジェネレータを備え、HRTIM_CHB1 に対して相補的に動作（立ち上がりエッジと立ち下がりエッジのデッド・タイムが同じ）

このコンバータは、HRTIM_FLT1 デジタル入力によってロー・レベル検知で保護されています。正出力極性と非アクティブ・フォールト状態のプログラミングにより、FAULT1 イベントですべての出力が強制的にロー・レベルになります。

図 13 は、3 種類の動作モードで 4 つの出力に現れる降圧昇圧制御波形を示しています。

図 13. 降圧昇圧コンバータの動作波形



比較レジスタの値を 0 または 100 にプログラミングすることで、それぞれ 0% または 100% のデューティ・サイクルが得られ、2 つの相補出力がオン/オフに維持されます。

- CMP1 の値が PER の値に等しい場合、デューティ・サイクルは 100% になります (CMP1 のセット・イベントが PER のリセット・イベントより優先されます。リファレンス・マニュアルでハードウェアの優先順の考え方を参照してください。2 つのイベントが同時に発生した場合の優先順位は、高い順に CMP4 → CMP3 → CMP2 → MP1 → PER です)。
- CMP1 の値が PER の値よりも大きい場合、デューティ・サイクルは 0% になります (セット・イベントが生成されなくなります)。

降圧昇圧のデモ概要

このデモを実行するには、降圧昇圧コンバータの V_{in} 入力ピンを、STM32F334 ディスカバリ・キット電源の 5V_O 出力ピンに接続する必要があります。得られた電圧は V_{out} ピンに出力されます。

このデモは降圧モードで始まり、 V_{in} 未満の範囲で V_{out} が連続的に変化するように、TIMA_IRQ ハンドラでデューティ・サイクルがゆっくり調整されます。プッシュボタンを押して電圧が 5 V 未満になると、昇圧モードが有効になります（電圧チェックによって、キットの最大出力電圧を超えないように制御されます）。デューティ・サイクルが一定の状態では電圧が V_{in} の値を超えます。もう一度プッシュボタンを押すと、出力コンデンサが放電してその電圧が 4.5 V になり、再び降圧モードが有効になります。

PA1 (V_{in}) 入力と PA3 (V_{out}) 入力に現れるコンバータ・オン時間の中央で変換がトリガされるように ADC が設定されています。これらの値はメイン・ルーチンで mV 単位に変換され、デモの間、デバッグで電圧が直接読み取られます（実行時に更新されるウォッチが使用されます）。

HRTIM_FLT1 入力は PA12（アクティブ・ロー）で有効になり、すべての出力で PWM がシャット・ダウンすることが示されます（ロー・レベル検知）。フォールトがトリガされると（PA12 入力が GND に接続された状態）、HRTIM_CHA1、HRTIM_CHA2、HRTIM_CHB1、HRTIM_CHB2 の各信号がシャットダウンします。ユーザ・ボタンを押すとシステムが再設定されます。

LED によって以下の状態が示されます。

- 緑：降圧動作中に点滅
- 青：昇圧動作中に点滅
- 赤：FAULT がトリガされると点滅
- オレンジ：PWM 更新 ISR の発生と期間を指示

デモ・コード

HRTIM_BuckBoost のサンプルにサンプル・コードが用意されています。

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

6 遷移モード力率コントローラ

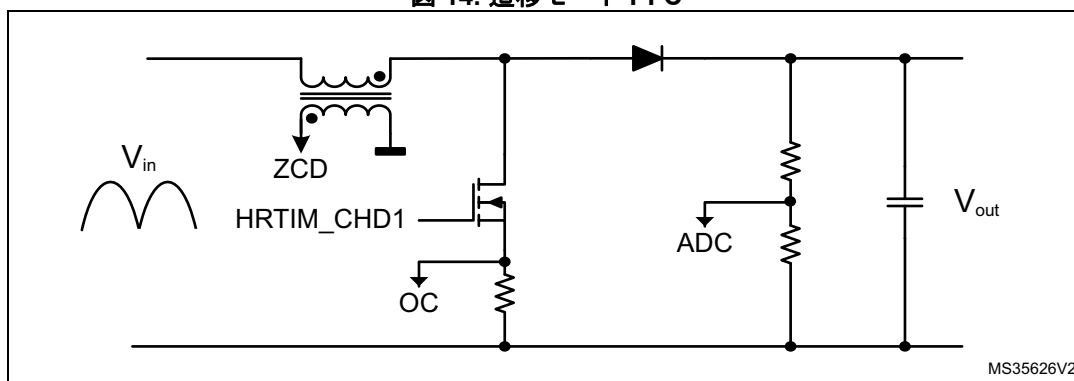
このセクションでは以下の点に重点を置きます。

- 外部イベントの条件設定とフィルタ処理
- タイマ・カウンタのリセット
- イベント・ブランキング・ウィンドウ
- キャプチャ・ユニット
- 一定 T_{on} 時間コンバータ

以下のサンプルでは、遷移モード（境界通電モードともいいます）力率コントローラ（PFC）を取り上げます。

その接続形態を図 14 に示します。これは昇圧コンバータに類似しています。出力電圧読み取り、および ZCD（ゼロ電流検出）用メイン・インダクタと組み合わせた消磁検出巻線を備えています。

図 14. 遷移モード PFC



基本的な動作原理は、固定の T_{on} 時間中に電流をインダクタに蓄積することです。

この電流は T_{off} 時間中に減少し、null になると周期が再開されます。メイン・インダクタ上に設けた補助巻線による ZCD 回路を使用して、この状態が検出されます。 T_{on} が一定であることから、インダクタ内のピーク電流値は整流された AC 入力電圧に正比例し、これによって力率が補正されます。

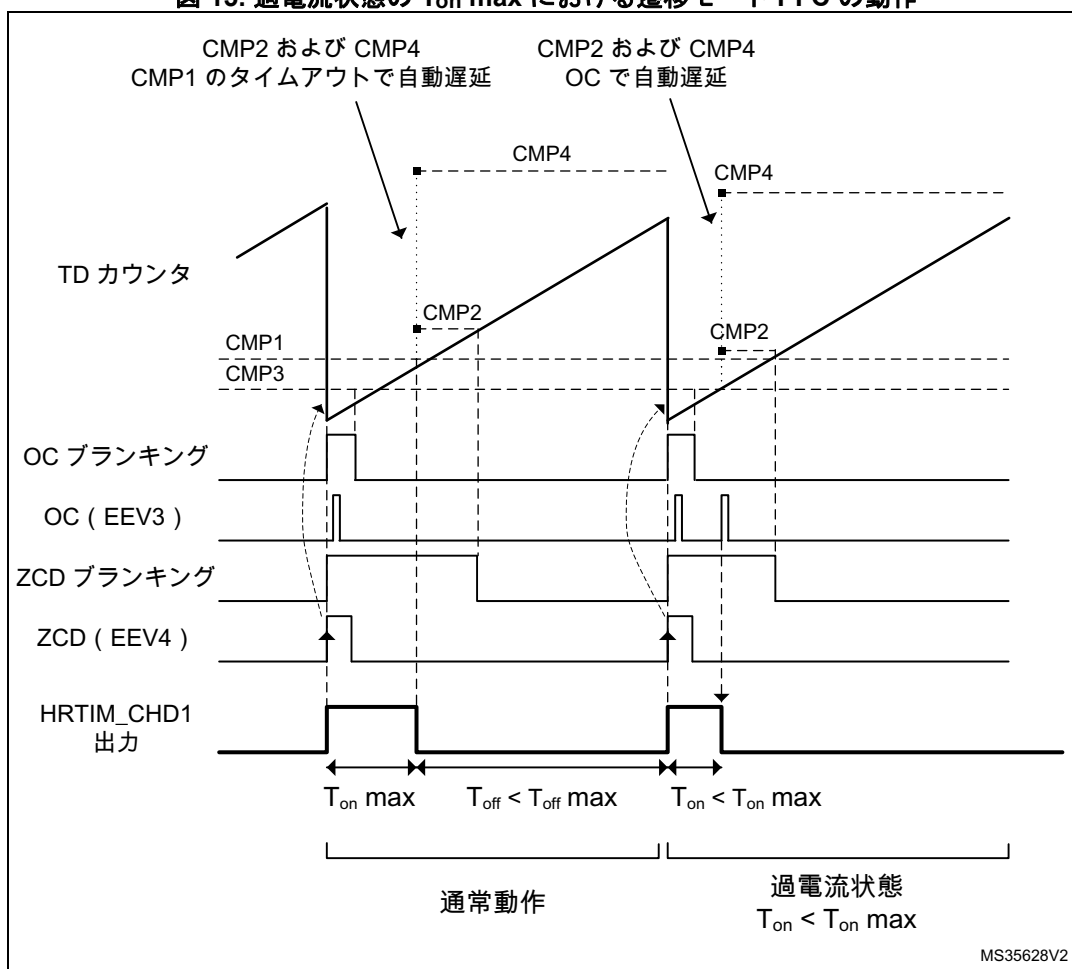
このコンバータは、一定の T_{on} と、 T_{off} の変動（入力電圧に依存）によって変化する周波数で動作します。また、ゼロ交差が検出されない場合に動作する機能や、過電流（OC）の条件下で T_{on} を制限する機能も備えている必要があります。OC フィードバックは通常、内蔵コンパレータで調整され、外部イベント・チャンネルに配送されます。

注： 以下の動作原理は、オフ時間が一定のコンバータにも適用できます。

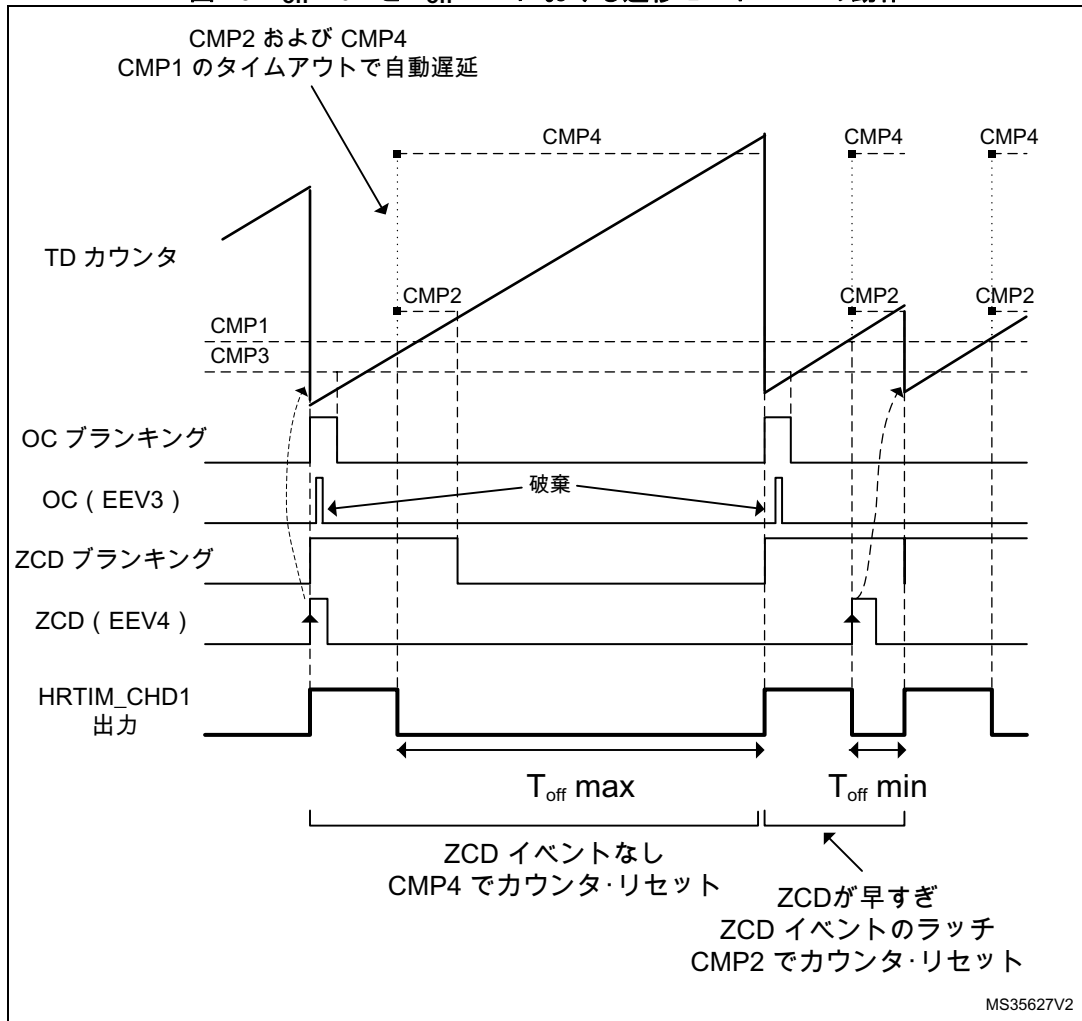
さまざまな動作モードにおける波形を図 15 と図 16 に示します。ここでは以下のパラメータを設定しています。

- $T_{on\ min}$: この期間では、誤った過電流が破棄されます（多くの場合、このような電流はバイパス・ダイオードのリカバリ電流です）。これは OC ブランキングで表され、CMP3 でプログラミングします。
- $T_{on\ max}$: 事実上、コンバータのセットポイントであり、CMP1 で定義します。
- $T_{off\ min}$: 電流制限がゼロに近いとき（消磁は非常に速い）、周波数を制限します。自動遅延モードで CMP2 によって定義されます。
- $T_{off\ max}$: ZCD が働かない場合に、システムが動かなくなることを防ぎます。自動遅延モードで CMP4 によって定義されます。

図 15. 過電流状態の $T_{on\ max}$ における遷移モード PFC の動作



MS35628V2

図 16. $T_{off\ max}$ と $T_{off\ min}$ における遷移モード PFC の動作

HRTIM を連続モード動作として、HRTIM_CHD1 信号を次のように定義します。

- TD CMP4 または $ZCD_{latched}$.TDCMP2 でセット
- TD CMP1 または OC.TDCMP3 でリセット

$ZCD_{latched}$.TDCMP2 は、タイマ D カウンタのリセットから TD CMP2 一致による終了までのブランキング・ウィンドウで ZCD イベントがフィルタ処理されていることを示しています。この ZCD イベントはラッチされます。ブランキング・ウィンドウの範囲で発生したこのイベントは破棄されず、ブランキング期間の終了時点で有効です。ZCD 信号が外部イベント 4 に適用されます。

OC.TDCMP3 は、タイマ D カウンタのリセットから TD CMP3 一致による終了までのブランキング・ウィンドウで OC イベントがフィルタ処理されていることを示しています。この OC イベントはラッチされません。ブランキング・ウィンドウの範囲で発生したこのイベントは破棄されます。OC 信号が外部イベント 3 に適用されます。

CMP2 と CMP4 に基づく 2 つの T_{off} 値はいずれも自動的に遅延します。これらのタイミングは、出力信号の立ち上がりエッジ (CMP1 一致または OC イベントで発生するエッジ) を基準とする必要があります。CMP2 イベントと CMP4 イベントは、それぞれキャプチャ 1 イベントとキャプチャ 2 イベントとして扱われます。

イベントに基づき、CMP1 のタイムアウトに伴う自動遅延モードで生成されます。この 2 つのキャプチャ・ユニットは OC 信号 (EEV3) でトリガされます。

タイマ D カウンタは、ZCD イベントまたは CMP4 一致 (タイムアウト条件の場合) でリセットされます。

このコンバータは、HRTIM_FLT1 デジタル入力によってロー・レベル検知で保護されています。正出力極性と非アクティブ・フォールト状態のプログラミングにより、FAULT1 イベントで HRTIM_CHD1 出力は強制的にロー・レベルになります。

遷移モード PFC のデモの概要

このデモを実行してすべての動作モードをテストするには、次の 2 つの入力信号のフィードバックをファンクション・ジェネレータでシミュレートする必要があります。

- 過電流 (OC、EEV3/PB7)
- ゼロ交差検出 (ZCD、EEV4/PB6)

次のように各種動作モードをテストできます。

- T_{on} 時間で OC 信号が生成された場合はパルスが短くなります。
- ZCD 信号でタイマ・カウンタがリセットされ、これに応じてスイッチング周波数変化します。

FAULT1 入力は PA12 (アクティブ・ロー) で有効になり、PWM がシャット・ダウンすることが示されます (ロー・レベル検知)。フォールトがトリガされると (PA12 入力が GND に接続された状態)、HRTIM_CHD1 信号が停止します。ユーザ・ボタンを押すとシステムが再設定されます。

LED によって以下の状態が示されます。

- 緑 : 正常動作中に点滅;
- オレンジ : FAULT がトリガされると点滅

デモ・コード

HRTIM_TM_PFC のサンプルにサンプル・コードが用意されています。

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

7 多相降圧コンバータ

このセクションでは以下の点に重点を置きます。

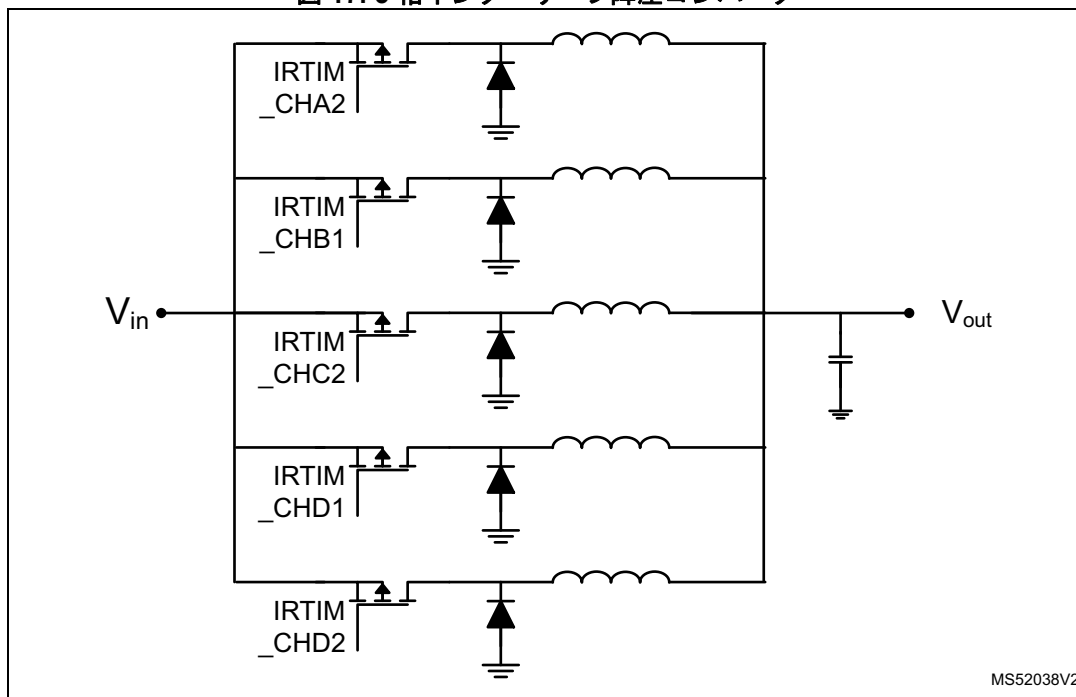
- マスタ・タイマ
- クロス・タイマのリセット
- 複数タイマの同期
- バースト・モード・コントローラ

多相技術は複数の電力変換接続形態（降圧、昇圧）に適用できます。これらの主要な利点は以下のとおりです。

- 入力コンデンサと出力コンデンサでの電流リップルの削減
- EMI の低減
- 位相の数を動的に変えること（位相シェディング）による軽負荷での高効率

HRTIM で駆動する 5 相インターリーブ降圧コンバータを図 17 に示します。

図 17. 5 相インターリーブ降圧コンバータ



マスタ・タイマは位相管理を受け持ちます。スレーブ・タイマを周期的にリセットすることによって複数の降圧コンバータ間の位相関係を定義します。相対位相シフトは 360° を位相の数で除算した値で、このサンプルでは 72° です。

マスタ・タイマは連続モードで動作し、スレーブ・タイマはシングルショット再トリガ可能モードで動作します。スレーブ・タイマ・カウンタはマスタ・タイマ・イベントでリセットされます（タイマ A はマスタ周期でリセットされ、タイマ B、C、D はそれぞれマスタ比較 1、2、3 でリセットされます）。

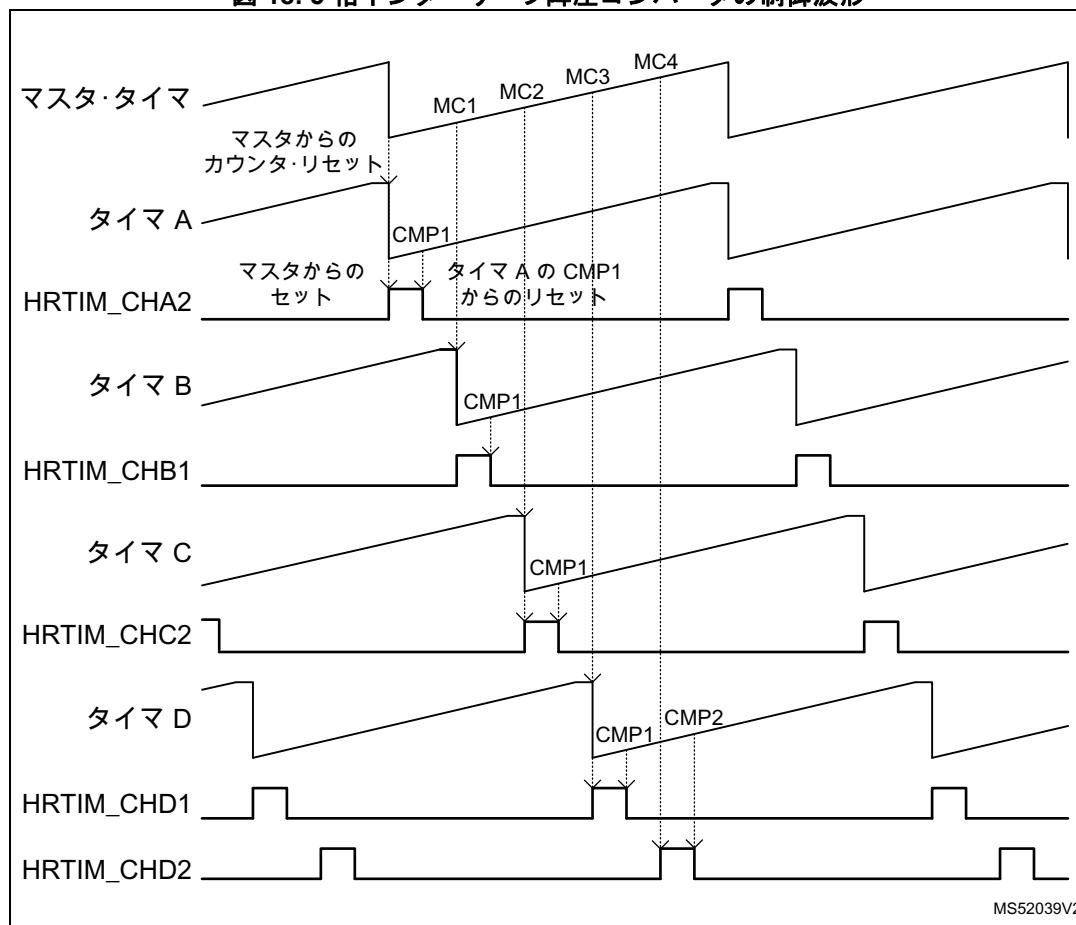
デューティ・サイクルは、位相シフトからは独立して各タイマにプログラミングされ、動作中に変化することがあります。出力波形は次のように定義します。

- HRTIM_CHA2 はマスタ・タイマ周期で:セット、TACMP1 でリセット
- HRTIM_CHB1 はマスタ・タイマの MCMP1 でセット、TBCMP1 でリセット
- HRTIM_CHC2 はマスタタイマの MCMP2 でセット、TCCMP1 でリセット
- HRTIM_CHD1 はマスタタイマの MCMP3 でセット、TDCMP1 でリセット
- HRTIM_CHD2 はマスタタイマの MCMP4 でセット、TDCMP2 でリセット

波形と主なイベント（出力のセット/リセットとスレーブ・タイマ・カウンタのリセット）を図 18 に示します。MCx はマスタ・タイマの CMPx、Cx はスレーブ・タイマの CMPx を表します。マスタ・タイマは、同じイベントでカウンタのリセットと出力のセットの両方を管理します。スレーブ・タイマからの比較イベントを使用してデューティ・サイクルをプログラミングします。これにより、出力電圧の安定化を目的とするデューティ・サイクル管理から、効率と EMI（アクティブな位相、位相シフトの変動、拡散スペクトル周波数のディザリングの削減）を目的とする位相管理を切り離すことができます。

注：ここで挙げるサンプルは、STM32F334 ディスカバリ・ハードウェアによるテストを意図しているので、出力マッピングといくつかの技巧を採用しています。通常の実装では、HRTIM_CHA1、HRTIM_CHB1、HRTIM_CHC1、HRTIM_CHD1、HRTIM_CHE1 の各出力を使用し、HRTIM_CHA2、HRTIM_CHB2、HRTIM_CHC2、HRTIM_CHD2、HRTIM_CHE2 の各相補出力に対する同期整流のオプションを用意します。

図 18. 5 相インターリーブ降圧コンバータの制御波形



マスタ・タイマは、その繰り返しイベントで更新されます（N 回の PWM 周期ごとに 1 回の更新で、スイッチング周波数から CPU による負荷を切り離し。N は 1 ~ 256 の整数）。この更新イベントによって、プリロードからアクティブなレジスタへの転送がトリガされるので、グリッチのリスクが発生することなく、変更されたすべての値が同時に考慮されます。

このイベントはすべてのスレーブ・レジスタにも伝播し、コンバータ全体で新たな動作セットポイント（位相シフトとデューティ・サイクル）が同期して更新されます。MCR レジスタの MPREPU ビットおよび TIMxCR レジスタの MSTU ビットを設定することで、この動作が実現します（TIMx はマスタの更新イベントで更新されます）。

次の更新の前に最大限のソフトウェア実行時間を確保できるように、コンバータはその実行時に単一の割り込みサービス・ルーチン（マスタの繰り返しイベント）で制御されます。

多相コンバータでは、出力負荷に応じて位相の数を動的に調整できるので、損失が減少し、効率が向上します。ある位相が無効になった場合は、残った各位相の位相シフトを調整する必要があります。4 つのマスタ・タイマの比較レジスタを調整することで、このような位相シフト調整を実現できます。マスタの比較値をマスタ・タイマ周期より大きい値に設定すると、スレーブ・タイマとその出力のセットは再トリガされなくなり、図 19 のように、該当の位相はオフになります。

図 19. マスタ・タイマによる位相の無効化

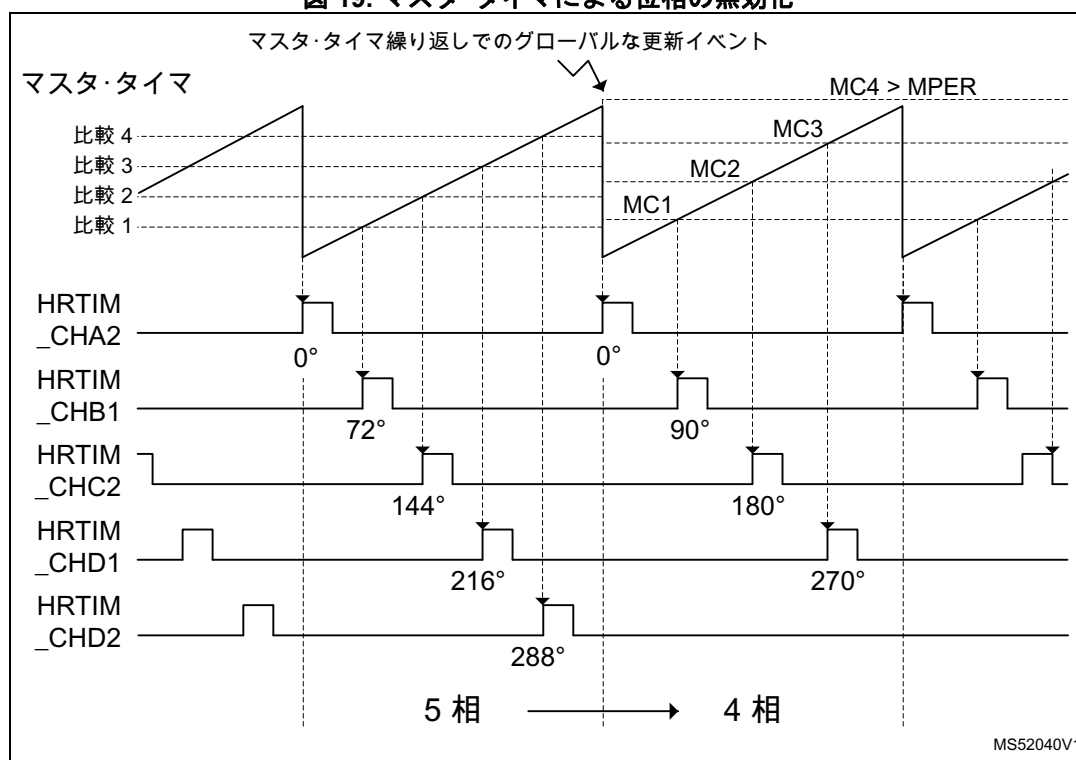
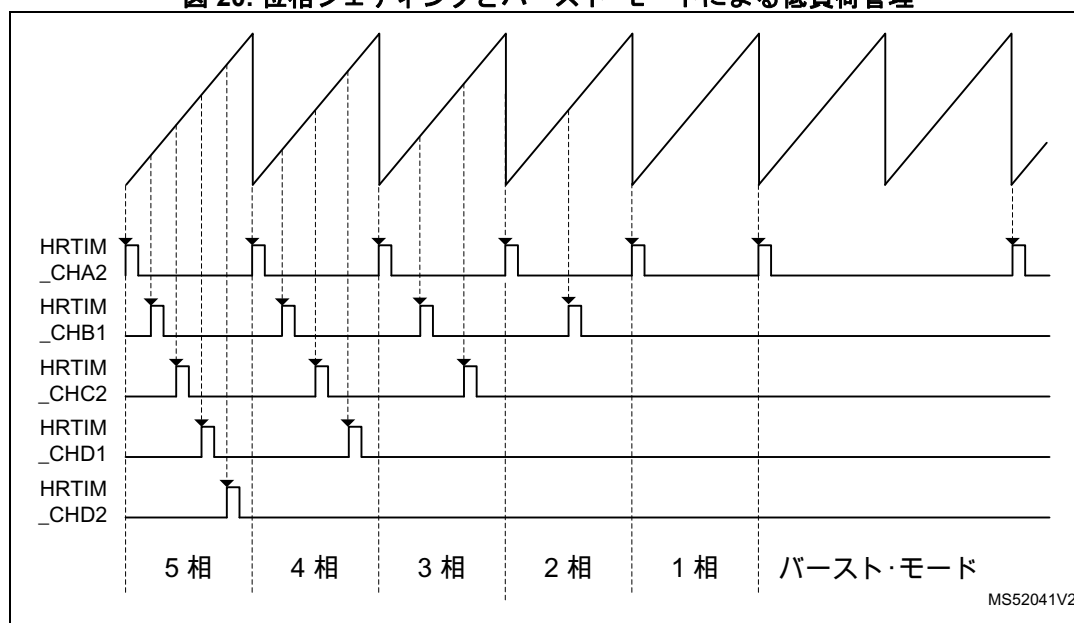


図 20 は、考えられるさまざまな位相對応を示しています。5 相モードから 1 相モードまでの 5 種類と、位相スキップのためのバースト・モードの合計 6 種類です。

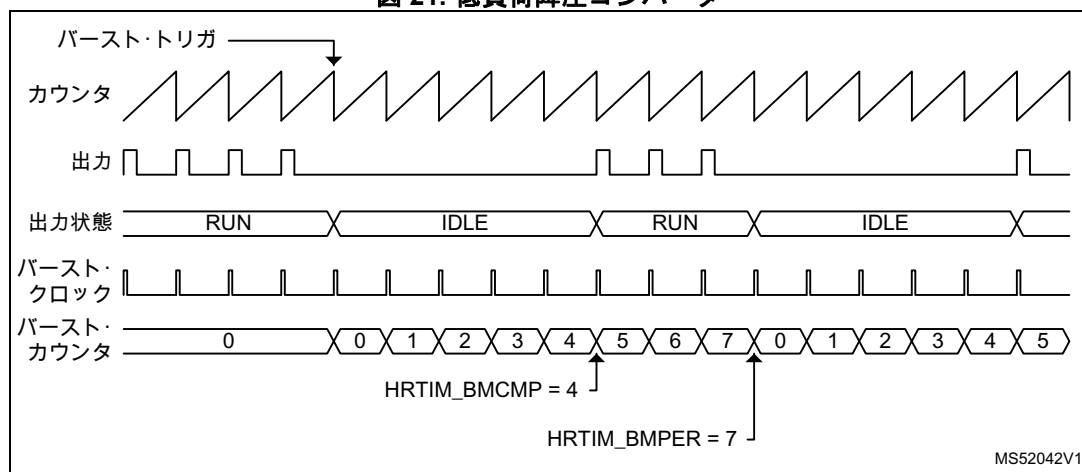
図 20. 位相シェディングとバースト・モードによる低負荷管理



バースト・モード・コントローラを使用すると、PWM 出力が定期的に無効になることにより、低負荷動作時に降圧コントローラの効率を改善できます。シングル・コンバータと多相コンバータのどちらにも、この手法を適用できます。

実行/アイドル比が 3/7 の場合の PWM 波形を図 21 に示します。この場合のバースト・モードはマスタ・タイマ・カウンタのロールオーバー・イベントでトリガされ、連続動作になります。全負荷動作に戻るときに、このモードをソフトウェアで終了します。

図 21. 低負荷降圧コンバータ



スペアの TxCMPy 比較イベントによって ADC トリガが生成されます。ここに用意されたサンプルでは、各位相のオン時間の中央で変換が実行されるように ADC が設定されています。どの ADC トリガソースでも位相がシフトしているので、それらすべてを 1 つの ADC トリガに結合して ADC リソースを節約できます（多相コンバータ全体を扱うには ADC の通常チャンネルが 1 つあれば十分です）。

ADC は非連続モードで動作します。シーケンサで一度にスキャンされる変換は 1 つなので、トリガごとに異なる変換が始まります。変換結果は、サーキュラ・モードでプログラミングされた DMA コントローラによって、5 つの場所を持つ RAM テーブルに保存されます。この RAM テーブルを大きくして（モジュール 5）、任意の深さで変換履歴を保存することもできます。

注： ADCx_CFGR レジスタの DISCNUM ビットフィールドをプログラミングすることによって、トリガごとに後続の複数の変換を定義することもできます（一般的には 1 つの電圧と 1 つの電流）。

サンプリング方法は、アクティブな位相の数によって異なります。停止しているスレーブ・タイマがあることが考えられるので、ADC のトリガ・ポイントに合わせる必要があります。ここに提示された実装では、位相データが確実に RAM の中の同じ場所で読み取られるようにダミーの変換を生成できます。表 6 に、アクティブな位相の数に応じたアクティブな ADC トリガを示します。

表 6. サンプリング・トリガと Data[0..4] の RAM テーブルに保存される結果

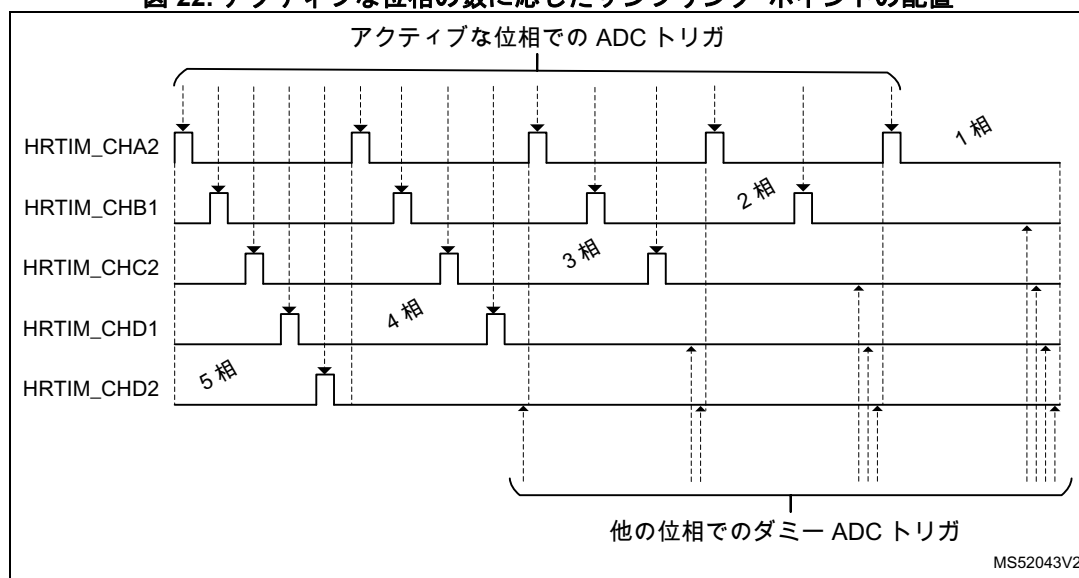
トリガ		5 相	4 相	3 相	2 相	1 相
1	ソース	TACMP2				
	結果 : Data[0]	Phase1_value				
2	ソース	TBCMP2				TACMP3
	結果 : Data[1]	Phase1_value				ダミー
3	ソース	TCCMP2			TBCMP3	TACMP4
	結果 : Data[2]	Phase3_value			ダミー	
4	ソース	TDCMP3		TCCMP3	TBCMP4	TAPER
	結果 : Data[3]	Phase4_value		ダミー		
5	ソース	TDCMP4			TDCMP3	MCMP4
	結果 : Data[4]	Phase5_value	ダミー			

位相が無効な場合は、残りのタイマ・リソースを使用して引き続きダミー・トリガが生成されます。これによって、指定の位相読み取りが必ず同じ位置で実行されるので、コンバータの動作中に ADC や DMA コントローラを再プログラミングしてシーケンスや DMA テーブル長を変更する必要がありません。

DMA がサーキュラ・モードで動作していて、Data[0..4] の RAM テーブルに変換結果が連続的に保存されているとします。表 6 は、Phase 3_value が必ず Data[2] の場所に保存されることを示しています。2 相の設定と 1 相の設定では、それぞれ TBCMP3 トリガと TACMP4 トリガを使用して Data[2] の場所にダミーの変換結果が保存されます（タイマ B は、1 相の設定では停止するので、トリガには使用できません）。

ダミーのサンプリング・ポイント（図 22 を参照）は、多相コンバータ周期の終了に近い位置に並べて配置します。これにより、関連するサンプリング・ポイントの配置に最大限の自由度が得られます。

図 22. アクティブな位相の数に応じたサンプリング・ポイントの配置



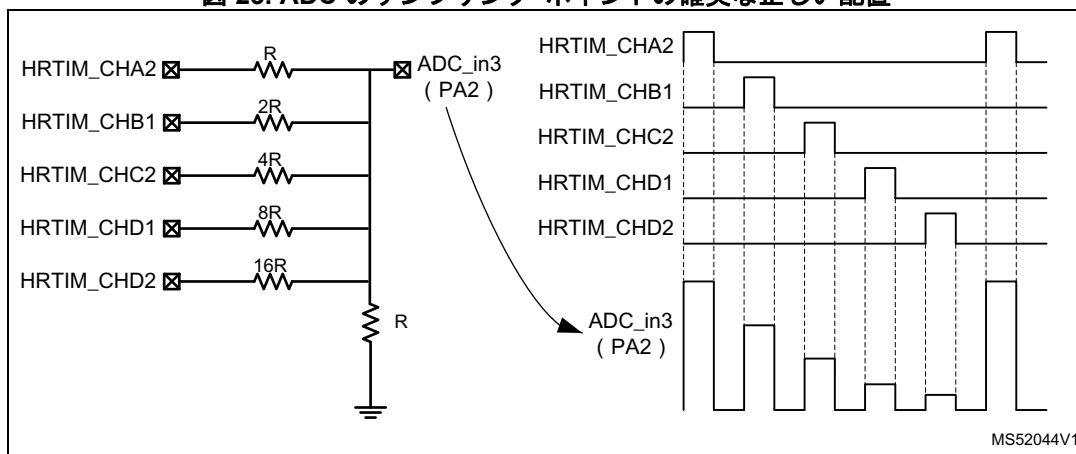
このサンプルではサンプリング・ポイントを 5 つにとどめていますが、これよりも多くのサンプリング・イベントを使用して他の入力を変換できます。ADC のインスタンスをもう 1 つ使用して、1 つまたは複数の独立した別のコンバータの動作を監視することもできます。

7.1 ADC の動作のデバッグ

このセクションでは、ADC の動作をデバッグして、サンプリング・ポイントの配置を簡単に確認するための技巧を紹介します。この技巧は、[図 23](#) のように、コンバータの PWM 出力を備えた簡潔な DAC の使用を基本としています。ADC のサンプリングはコンバータの T_{on} 時間の中央に配置されるので、同時に以下の動作を容易に確認できます。

- サンプリング・ポイントがオン時間の中で発生する（そうでない場合は null 値が得られる）。
- RAM テーブルには正しい順序で値が保存される。1 番目の位相のサンプリングでは $V_{DD}/2$ が返され、2 番目では $V_{DD}/3$ が返され、以下同様に続く。
- 位相シェディングでサンプリング・ポイントが失われることがない。サンプリングが失われても、テーブルの内部で位相データがローテーションされる。

図 23. ADC のサンプリング・ポイントの確実な正しい配置



さまざまな設定で想定される値を表 7 に示します。バースト・モードでは 1 相の設定に近い設定になります。

表 7. 想定される変換結果

トリガ		5 相	4 相	3 相	2 相	1 相
1	結果 : Data[0]	V _{REF+} /2				
2	結果 : Data[1]	V _{REF+} /3				0
3	結果 : Data[2]	V _{REF+} /5			0	
4	結果 : Data[3]	V _{REF+} /9		0		
5	結果 : Data[4]	V _{REF+} /17	0			

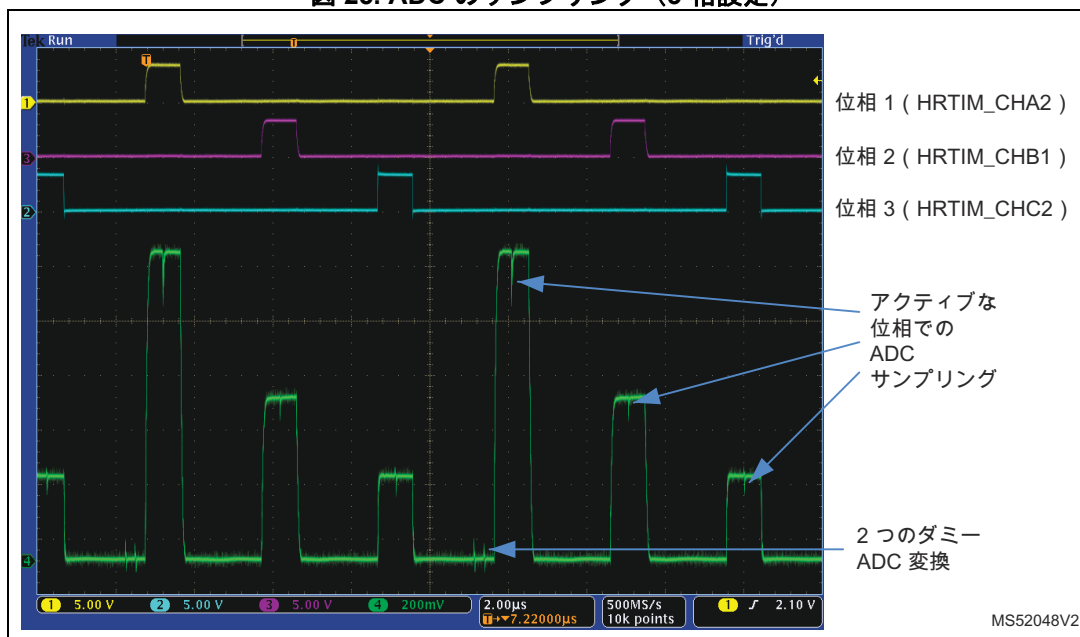
高い精度で ADC の動作を監視する方法としてオシロスコープの使用があります。STM32 マイクロコントローラで最高の ADC 精度を実現する方法 (AN2834) の説明にあるように、SAR ADC では、サンプリングとホールドの切り替えに起因して、位相サンプリングの開始時に署名が記録されます。この署名は、図 24 のように小さい負のスパイクとして観測されます (アナログ信号源のインピーダンスが過剰に低くないことが条件です)。

図 24. ADC のサンプリング (5 相設定)



ダミー変換の発生も確認できます。図 25 の最下部の波形を参照してください。

図 25. ADC のサンプリング (3 相設定)



注意： タイマ・リセット関数を使用する場合は、設定を保持している TIMxCR レジスタを更新してからアクティブにする必要があります。更新ソースがリセットなので、最初の更新を強制的にソフトウェアで実行してからカウンタを有効にして、コンバータの動作を開始する必要があります。この操作には、TIMx レジスタの TxSWU ビットを使用します。

多相のデモ概要

このデモでは、ハードウェアのサポートがない場合、5 つの HRTIM PWM チャンネルの監視のみが可能です（アクティブな PWM 出力は、フリーであるか、LED にもオンボードの降圧昇圧コンバータにも効果を及ぼさないかのどちらかです）。ここで必要な接続は以下のとおりです。

- PA12 をグラウンドに接続する、FAULT 入力のテスト接続
- ADC の動作を監視するためのテスト接続

以下の各出力に PWM 信号が得られます。

- 位相 1 : HRTIM_CHA2 (PA9)
- 位相 2 : HRTIM_CHB1 (PA10)
- 位相 3 : HRTIM_CHC2 (PB13)
- 位相 4 : HRTIM_CHD1 (PB14)
- 位相 5 : HRTIM_CHD2 (PB15)

このデモは 5 相モードで始まります。プッシュボタンを押すと、デモ・モードが変わり、図 20 のように 5 相モードから 1 相モードまで、最後にバースト・モードの順にすべての位相シェディング・オプションがスキャンされます。

5 つの位相それぞれで、コンバータのオン時間の中央で変換がトリガされるように ADC が設定されています。このサンプルの場合は PA2 入力です（通常は、5 つの入力での 5 つの変換の順序です）。

FAULT1 入力は PA12（アクティブ・ロー）で有効になり、すべての出力で PWM がシャット・ダウンすることが示されます（ロー・レベル検知）。フォールトがトリガされると（PA12 入力が GND に接続された状態）、HRTIM_CHA2、HRTIM_CHB1、HRTIM_CHC2、HRTIM_CHD1、HRTIM_CHD2 の各信号がシャットダウンします。ユーザ・ボタンを押すとシステムが再設定されます。

LED によって以下の状態が示されます。

- LED5 の緑色点滅 : 5 相動作
- LED6 の青色点滅 : 4 相動作
- LED5 の緑色連続点灯 : 3 相動作
- LED6 の青色連続点灯 : 2 相動作
- 青色 LED と緑色 LED の連続点灯 : 1 相動作
- 青色 LED と緑色 LED の点滅 : バースト・モード動作
- 赤色の LED3 : FAULT がトリガされると点滅
- オレンジ色の LED4 : PWM 更新 ISR の発生と期間を指示

デモ・コード

HRTIM_Multiphase のサンプルにサンプル・コードが用意されています。サイズに制約のあるアプリケーションでは、#define SNIPPET ステートメントが HAL ベースのデモの代わりになります。

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

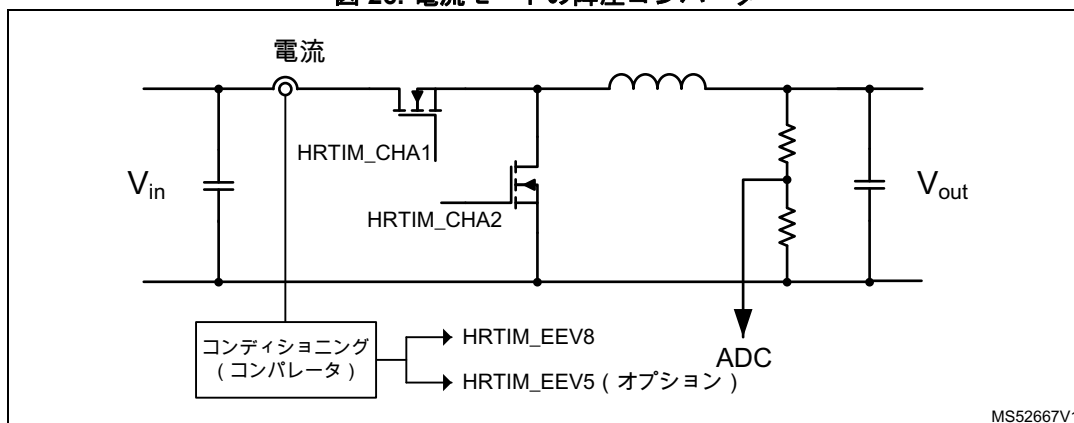
8 デッドタイムが挿入されないサイクル単位の保護

このセクションでは以下の点に重点を置きます。

- 過電流時のデッドタイム管理
- デッドタイムを挿入するための自動遅延モード

過電流保護された HRTIM 駆動降圧コンバータを図 26 に示します。

図 26. 電流モードの降圧コンバータ

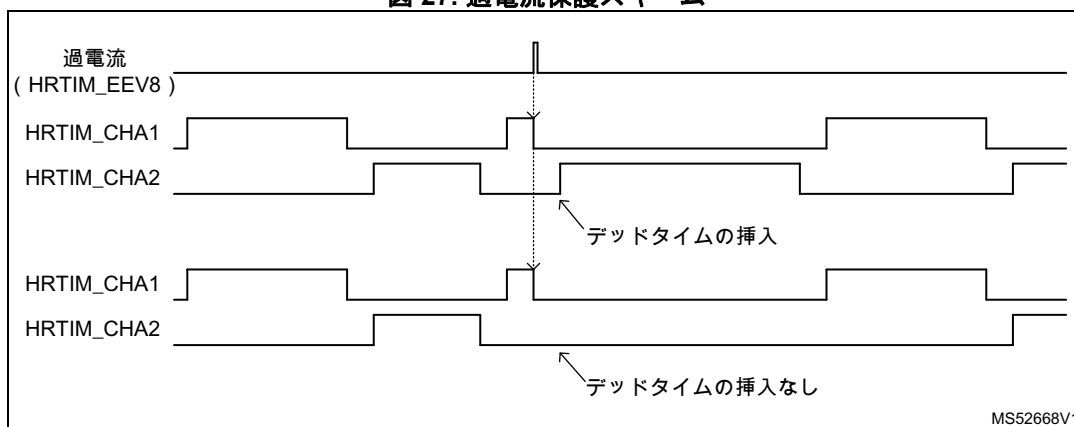


HRTIM_CHA1 で駆動されるハイサイドの MOSFET がメインの電源スイッチとして機能し、固定の PWM 周波数による出力電圧の制御とデューティ・サイクルの調整を実行します。HRTIM_CHA2 で制御されるローサイドの MOSFET は同期整流を実行します。これら 2 つのトランジスタは相補性 PWM 信号とクロス導通防止のためのデッドタイム挿入で制御されます。

セクション 4 の説明にあるように、デッドタイム生成ユニットでデッドタイムを挿入できます。デッドタイム生成ユニットを使用するということは、ハイサイドの MOSFET コマンドをリセットする過電流保護が機能していても、信号が必ず相補性であることを意味します。

過電流保護に続くスイッチング周期で両方の MOSFET がオフになる場合に備え、代わりとなる保護スキームの用意が必要になることが考えられます。図 27 に 2 種類の動作を示します。

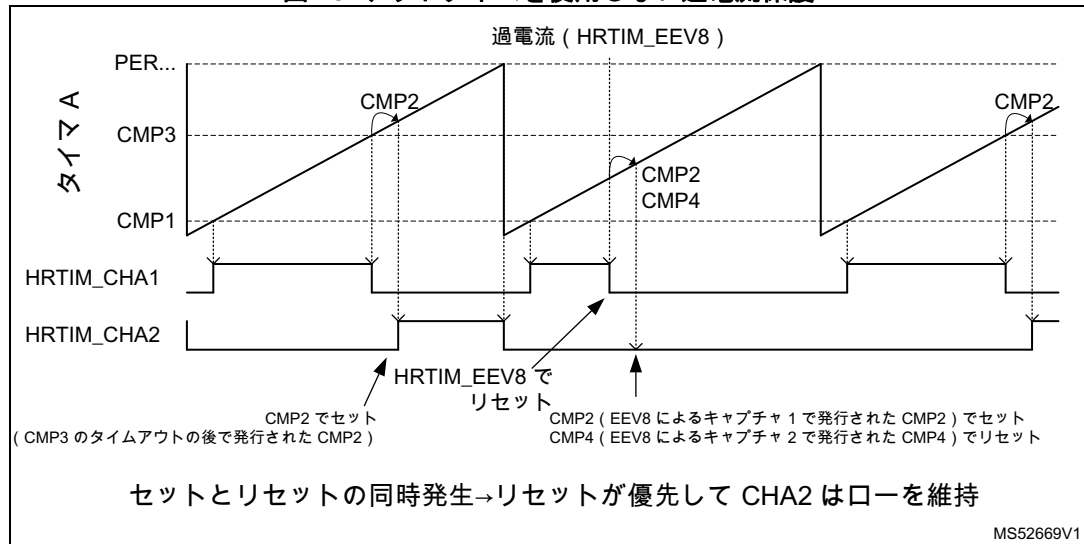
図 27. 過電流保護スキーム



デッドタイム生成ユニットを使用しない保護スキームを実装するには、CMP1 を使用して立ち上がりエッジにデッドタイムを生成します（タイマ A 周期で HRTIM_CHA2 をリセットし、タイマ A の CMP1 で HRTIM_CHA1 をセットします）。立ち下がりエッジにデッドタイムを生成するには、CMP3 のタイムアウトに伴って HRTIM_EEV8 入力でトリガされる CMP2 自動遅延モードを使用します。この手順は次のとおりです。

- 通常の動作条件下（過電流ではない状態）における HRTIM_CHA1 でのデューティ・サイクルを CMP3 レジスタで定義します。
- 過電流状態でなければ CMP3 のタイムアウトで自動遅延モードがトリガされ、CMP2 レジスタの値で定義されているデッドタイムの後で HRTIM_CHA2 出力がセットされます。
- 過電流状態であれば、HRTIM_EEV8 入力によって自動遅延モードがトリガされ、CMP2 で定義されているデッドタイムの後で、HRTIM_CHA2 出力に対するセット・リクエストがこのモードによって発行されます。このセット・リクエストを取り消すには、自動遅延モードで CMP4 レジスタを使用して、同時リセット・リクエストも発行します。リセット・リクエストはセット・リクエストよりも優先されるので、図 28 のように、HRTIM_CHA2 出力はローの状態を維持します。

図 28. デッドタイムを使用しない過電流保護



自動遅延の CMP2（このサンプルでは HRTIM_EEV8）と同じソースから、同じ比較値を指定して、自動遅延の CMP4 をトリガする必要があります。タイムアウト・モードを無効にする必要があります。過電流状態がないと、CMP4 イベントが生成されないからです。

自動遅延の CMP2 は、HRTIM_EEV8 の立ち上がりエッジと立ち下がりエッジでのキャプチャ 1 イベントでトリガされます。また、自動遅延の CMP4 は、同じく HRTIM_EEV8 の立ち上がりエッジと立ち下がりエッジでのキャプチャ 2 イベントでトリガされます。

CMP1 がオンになる前に過電流パルスが発生した場合、HRTIM_EEV8 はラッチされ、CMP1 イベントの時点まで遅延します。これにより、過電流情報が確認され、両方の出力が早いタイミングで発生しても、それらの出力がゼロに維持されます。

このスキームは、スイッチング周期の中で、短時間の過電流パルスに対して機能します。このような状態は、短時間の過負荷で発生することが普通です。

PWM の複数周期（複数サイクルの過負荷状態）のような長期間にわたって PWM のシャットダウンを必要とするアプリケーションでは、HRTIM_EEV8 と並列に接続した別の HRTIM_EEVx 入力（このサンプルでは HRTIM_EEV5）で上記のスキームを完了する必要があります。

HRTIM_EEV5 イベントはアクティブでオン・レベルになるようにプログラミングされているので、過電流状態が存在する限り、2 つの出力はローに維持されます（図 29 を参照）。HRTIM_EEV8 イベントのみがアクティブになる場合の波形を図 30 に示します。

図 29. デュアル外部イベントによる複数サイクル過負荷保護

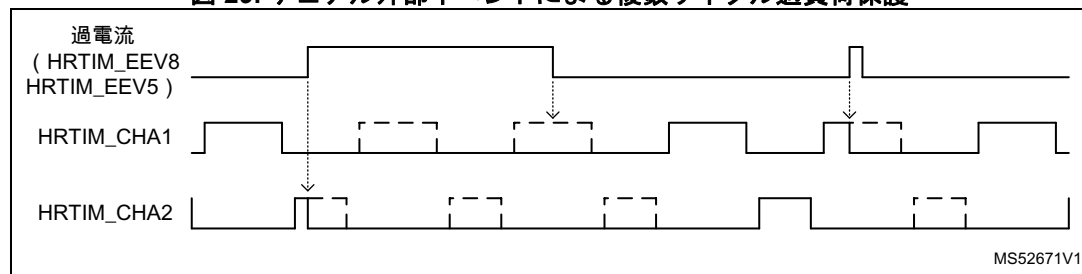
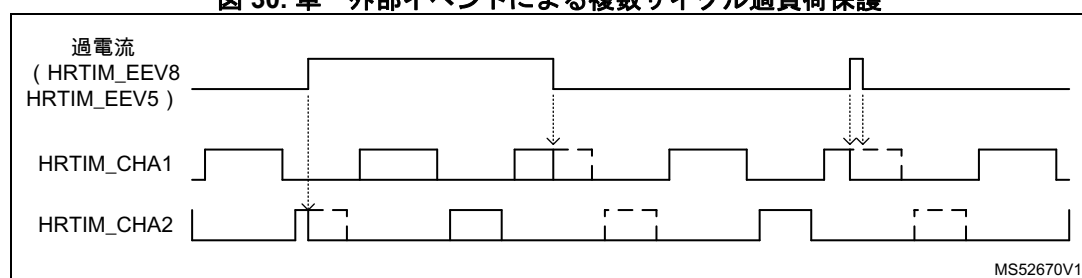


図 30. 単一外部イベントによる複数サイクル過負荷保護



まとめると、出力設定（ブラケット間の複数サイクル過負荷オプション）は以下のようになります。

- HRTIM_CHA1 のセット : CMP1
- HRTIM_CHA1 のリセット : CMP3 あるいは HRTIM_EEV8 のエッジ（または HRTIM_EEV5 のアクティブ化）
- HRTIM_CHA2 のセット : CMP2（自動遅延）
- HRTIM_CHA2 のリセット : タイマ A 周期あるいは CMP4 の自動遅延（または HRTIM_EEV5 のアクティブ化）

デモ・コード

HRTIM_CBC_Deadtime のサンプルにサンプル・コードが用意されています。

ターゲットのボードとファームウェアへのアクセス・パスについては[セクション 10](#)を参照してください。

9 他のサンプル

この説明書は今後充実してゆく予定です。新たなサンプルについては、www.st.com で STM32 製品に関するページを参照してください。

10 ソフトウェア・サンプルの入手先

このセクションでは、各ボードで実行できるサンプルと、最新の CubeFW 拡張パッケージでこれらのサンプルが置かれている場所をまとめています。

表 8 のアイコンは、STM32CubeMX を使用してプロジェクトを再生成できるかどうかを示しています。再生成できる場合は、同じファームウェアの場所に該当の ioc ファイルがあります。

表 8. サンプルの概要

サンプル	ボード (互換ボード)	ファームウェアの場所	STM32 CubeMX
HRTIM の基本的な 動作原理	32F3348DISCOVERY (NUCLEO-F334R8)	STM32Cube_FW_F3_V1.10.0\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_BasicPWM	-
		STM32Cube_FW_F3_V1.10.0\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_Snippets	-
	NUCLEO-G474RE	STM32Cube_FW_G4_Vx.y.z\Projects\ NUCLEO-G474RE\Examples_LL\HRTIM\ HRTIM_Basic_Single_PWM HRTIM_Basic_PWM_Master HRTIM_Basic_Multiple_PWM HRTIM_Basic_Arbitrary_Waveform	
		STM32Cube_FW_G4_Vx.y.z\Projects\ NUCLEO-G474RE\Examples\HRTIM\ HRTIM_Basic_Single_PWM HRTIM_Basic_PWM_Master HRTIM_Basic_Multiple_PWM HRTIM_Basic_Arbitrary_Waveform	
電圧モード・デュアル バックコンバータ	32F3348DISCOVERY (NUCLEO-F334R8)	STM32Cube_FW_F3_Vx.y.z\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_DualBuck	-
	B-G474E-DPOW1 (NUCLEO-G474RE)	STM32Cube_FW_G4_Vx.y.z\Projects\ B-G474E-DPOW1\Examples_MIX\ HRTIM\HRTIM_Dual_Buck	
同期整流とフォールト 保護を備えた電圧モードバック コンバータ	32F3348DISCOVERY	STM32Cube_FW_F3_Vx.y.z\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_BuckSyncRect	-
	B-G474E-DPOW1	STM32Cube_FW_G4_Vx.y.z\Projects\ B-G474E-DPOW1\Examples_MIX\ HRTIM\HRTIM_Buck_Sync_Rect	

表 8. サンプルの概要（続き）

サンプル	ボード (互換ボード)	ファームウェアの場所	STM32 CubeMX
非反転バック-ブーストコンバータ	32F3348DISCOVERY	STM32Cube_FW_F3_Vx.y.z\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_BuckBoost	-
		STM32Cube_FW_F3_Vx.y.z\Projects\ STM32F3348-Discovery\Examples_LL\ HRTIM\HRTIM_BuckBoost	-
	B-G474E-DPOW1	STM32Cube_FW_G4_Vx.y.z\Projects\ B-G474E-DPOW1\Examples_MIX\ HRTIM\HRTIM_Buck_Boost	
遷移モード力率コントローラ	32F3348DISCOVERY (NUCLEO-F334R8)	STM32Cube_FW_F3_Vx.y.z\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_TM_PFC	-
多相降圧コンバータ	32F3348DISCOVERY (NUCLEO-F334R8)	STM32Cube_FW_F3_Vx.y.z\Projects\ STM32F3348-Discovery\Examples\ HRTIM\HRTIM_Multiphase	-
デッドタイムが挿入されないサイクル単位の保護	NUCLEO-G474RE	STM32Cube_FW_G4_Vx.y.z\Projects\ NUCLEO-G474RE\Examples_LL\ HRTIM\HRTIM_CBC_Deadtime	

付録 A HRTIM v2

A.1 機能

このセクションでは、HRTIM v2.0 の主な改善点をまとめています。詳細についてはリファレンス・マニュアルを参照してください。

A.1.1 6 番目のタイミング・ユニット (タイマ F)

現在の HRTIM は 12 個の出力を備え、大規模なインターリーブ接続形態に対応しています。このような接続形態として、トリプルインターリーブ・ハーフブリッジ LLC (3 x 4 出力が必要。主にハーフブリッジの制御に 2 つ、次いで同期整流に 2 つ)、デュアルインターリーブ・フルブリッジ LLC (2 x 6 出力)、デュアル位相シフト・フルブリッジ・コンバータ (2 x 6 出力) などがあります。

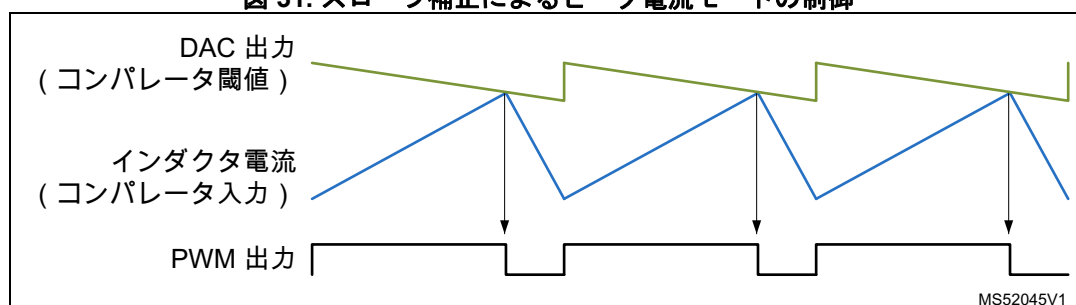
6 番目の FAULT 入力回路を備えています。

A.1.2 デュアル・チャネル DAC トリガ

この機能をハイスピード DAC と組み合わせて使用すると、PWM 信号に同期したのこぎり波を生成できます。図 31 に矢印で示されているように、コンパレータ入力 (青色) が閾値 (緑色) に達するたびに PWM 出力がリセットされます。

この技法はスロープ補正として知られ、ピーク電流モード・コンバータの安定性を確保するうえで必要です。

図 31. スロープ補正によるピーク電流モードの制御



A.1.3 外部イベントのカウンタ

HRTIM v2 はイベント駆動のフィルタ機能を備えています。外部イベントが有効と見なされるには、所定の回数でそのイベントが発生する必要があります。フライバック・コンバータでバレー (谷) スキップ・モードを実装する場合に、このような扱いが求められます。

A.1.4 拡張フォールト機能

プログラミング可能なブランキング・ウィンドウを使用して、FAULT イベントをフィルタ処理できます。各チャネルにはフォールト・カウンタもあるので、所定回数の PWM 周期の範囲で発生したフォールトのみがアクティブになります。

A.1.5 プッシュプルの改善

デッド・タイムの挿入によってプッシュプル・モードを使用できるようになりました。また、シングルショット動作モードでも使用できます (このモードは HRTIM v1 では使用できません)。

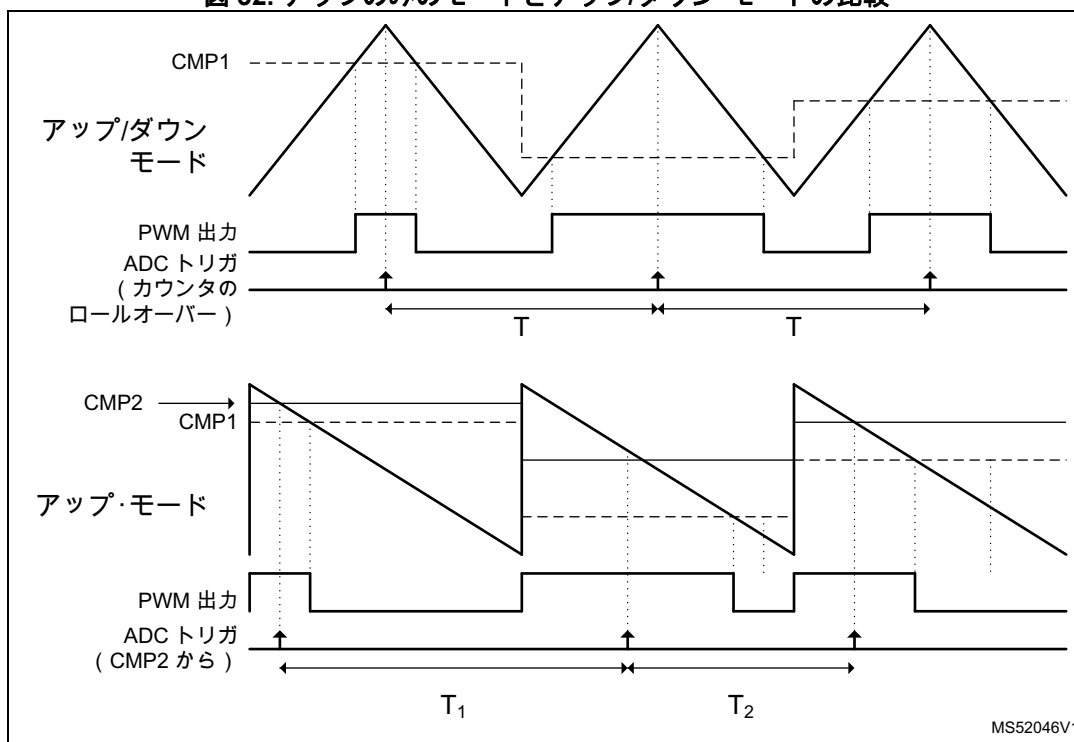
A.1.6 クラシック PWM モード

このモードでは、シャドウレジスタとプリロード・メカニズムを使用せずに HRTIM を動作状態にすることができ、ソフトウェアによって比較レジスタが書き込まれるとただちに出力に所定の状態を強制できます。これによって、早期ターンオフと遅延ターンオンが可能になり、コンバータの位相マージンをわずかに改善できる余地が生まれます。

A.1.7 アップ/ダウン・モード

このカウンタ動作モードは、モータ制御アプリケーションで広く使用されています。電力コンバータでも利点が得られます。図 32 に示すように、一定のサンプリング周波数とパルスの中央でのサンプリングが必要な場合に、このモードで ADC のサンプリングを簡素化できます。

図 32. アップのみのモードとアップ/ダウン・モードの比較



A.1.8 ADC ポスト・スケーラ

スイッチング周波数が高いアプリケーションでは、ADC ポスト・スケーラを使用すると ADC のトリガ・レートを低減できます。各 ADC トリガを個々に調整して、32 回の PWM 周期中に発生するトリガを最小で 1 回にすることができます。

A.1.9 ヌル・デューティ・サイクル・モード

HRTIMv2 では、Compare1 レジスタまたは Compare3 レジスタに null 値を書き込むことでヌル・デューティ・サイクルを強制できます。

A.1.10 新しいインターリーブ・モード

トリプル・インターリーブ・コンバータ、クワッド・インターリーブ・コンバータ、多相コンバータ向けに 2 つのインターリーブ・モードが新たに追加されました。

A.1.11 スワップ・モード

出力クロスバーの再プログラミングを必要とせずに、レジスタへの 1 回のアクセスで 2 つの出力をスワップできます。複数の PWM ペアを同時にスワップするために必要なすべての制御ビットが、同じレジスタに配置されています。

付録 B ソフトウェアの移行

B.1 HRTIM v1.0 から HRTIM v1.1 への移行

バージョン v1 と v1.1 との唯一の相違点は、HRTIM v1.1 では DLL が削除されていることです。

HRTIM_DLLCR レジスタは保持されています。コードを移植する際は、HRTIM_DLLCR へのすべてのアクセスと DLL の較正手順を削除する必要があります。

その結果、以下のようになります。

- 高分解能を使用した、すべてのクロックのプリスケアラ比が保持されます（HRTIM_MCR レジスタの CKPSC[2:0] = 000、001、010、011、100）。
- 可能な最高分解能が 2.5 ナノ秒であることを考慮してタイミングを調整する必要があります。

これらの点を除くと、すべての機能と動作モードは同じです。レジスタとビット・マッピングにはバージョン間で厳密な互換性があります。

B.2 HRTIMv1 から HRTIMv2 への移行

HRTIMv2 には、6 番目のタイミング生成ユニットといくつかの追加機能があります。

可能な範囲で互換性を維持するために、HRTIMv1 の予約済み領域に新たな制御ビットとステータス・ビットが追加されていますが、新しい 2 つのタイマにはバイナリ上の互換性はありません。

HRTIMv1 ではレジスタに空きがないことから、すべてのビットを 1 つの 32 bit レジスタに維持することにしたので互換性が失われています。ビット位置を変更した場合、HAL ライブラリと LL ライブラリを使用していれば、その変更が透過的に管理されます。これらのライブラリを使用していない場合は、HRTIMv2 に対応できるようにソース・コードを修正する必要があります。

以下の表では、v1 と v2 とのバイナリ以外の互換性の相違点を、グレー背景のセルに示しています。

表 9. イベントの対応表

移行先とバージョン		移行元																							
		タイマ A				タイマ B				タイマ C				タイマ D				タイマ E				タイマ F			
		CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4
タイマ A	v1	-	-	-	-	1	2	-	3	-	4	5	-	6	7	-	-	-	-	8	9	該当なし			
	v2	-	-	-	-	1	2	-	-	-	3	4	-	5	6	-	-	-	-	7	8	-	-	-	9
タイマ B	v1	1	2	-	3	-	-	-	-	-	-	4	5	-	-	6	7	8	9	-	-	該当なし			
	v2	1	2	-	-	-	-	-	-	-	-	3	4	-	-	5	6	7	8	-	-	-	-	9	-
タイマ C	v1	-	1	2	-	-	3	4	-	-	-	-	-	-	5	-	6	-	7	8	9	該当なし			
	v2	-	1	2	-	-	3	4	-	-	-	-	-	-	5	-	6	-	-	7	8	-	9	-	-
タイマ D	v1	1	-	-	2	-	3	-	4	5	-	6	7	-	-	-	-	8	-	-	9	該当なし			
	v2	1	-	-	2	-	3	-	4	-	-	-	5	-	-	-	-	6	-	-	7	8	-	9	-
タイマ E	v1	-	-	1	2	-	-	3	4	5	6	-	-	7	8	-	9	-	-	-	-	-	-	-	-
	v2	-	-	-	1	-	-	2	3	4	5	-	-	6	7	-	-	-	-	-	-	-	-	8	9

表 9. イベントの対応表 (続き)

移行先とバージョン		移行元																							
		タイマ A				タイマ B				タイマ C				タイマ D				タイマ E				タイマ F			
		CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4
タイマ F	v1	該当なし																							
	v2	-	-	1	-	2	-	-	3	4	-	-	5	-	-	6	7	-	8	9	-	-	-	-	-

表 10. タイマごとのウィンドウイング信号のマッピング (EEFLTR[3:0] = 1111)

HRTIM バージョン	移行先	タイマ A	タイマ B	タイマ C	タイマ D	タイマ E	タイマ F
v1	TIMWIN (ソース)	タイマ B CMP2	タイマ A CMP2	タイマ D CMP2	タイマ C CMP2	タイマ D CMP2	該当なし
v2		タイマ B CMP2	タイマ A CMP2	タイマ D CMP2	タイマ C CMP2	タイマ F CMP2	タイマ E CMP2

表 11. タイマごとのフィルタリング信号のマッピング

移行先とバージョン		移行元																							
		タイマ A				タイマ B				タイマ C				タイマ D				タイマ E				タイマ F			
		CMP1	CMP2	CMP4	HRTIM_CHA2	CMP1	CMP2	CMP4	HRTIM_CHB2	CMP1	CMP2	CMP4	HRTIM_CHC2	CMP1	CMP2	CMP4	HRTIM_CHD2	CMP1	CMP2	CMP4	HRTIM_CHE2	CMP1	CMP2	CMP4	HRTIM_CHF2
タイマ A	v1	-	-	-	-	1	-	2	3	4	-	5	6	7	-	-	-	-	8	-	-	該当なし			
	v2	-	-	-	-	1	-	2	3	4	-	5	-	7	-	-	-	-	8	-	-	6	-	-	-
タイマ B	v1	1	-	2	3	-	-	-	-	4	5	-	6	-	7	-	-	8	-	-	-	該当なし			
	v2	1	-	2	3	-	-	-	-	4	5	-	-	-	7	-	-	8	-	-	-	-	6	-	-
タイマ C	v1	-	1	-	-	2	-	3	4	-	-	-	-	5	-	6	7	-	-	8	-	該当なし			
	v2	-	1	-	-	2	-	3	-	-	-	-	-	5	-	6	7	-	-	8	-	4	-	-	-
タイマ D	v1	1	-	-	-	-	2	-	-	3	4	-	5	-	-	-	-	6	-	7	8	該当なし			
	v2	1	-	-	-	-	2	-	-	3	4	-	5	-	-	-	-	6	-	7	-	-	-	8	-
タイマ E	v1	-	1	-	-	2	-	-	-	3	-	4	5	6	-	7	8	-	-	-	-	該当なし			
	v2	-	1	-	-	2	-	-	-	3	-	-	-	6	-	7	8	-	-	-	-	-	-	4	5
タイマ F	v1	該当なし																							
	v2	-	-	1	-	-	2	-	-	-	-	3	-	-	4	5	-	6	-	7	8	-	-	-	-

表 12. HRTIM ADC のトリガ 1 レジスタとトリガ 3 レジスタ (HRTIM_ADC1R、HRTIM_ADC3R)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
v1	ADC1 TEPER	ADC1 TEC4	ADC1 TEC3	ADC1 TEC2	ADC1 TDPER	ADC1 TDC4	ADC1 TDC3	ADC1 TDC2	ADC1 TCPER	ADC1 TCC4	ADC1 TCC3	ADC1 TCC2	ADC1 TBRST	ADC1 TBPER	ADC1 TBC4	ADC1 TBC3
v2	ADC1 TEPER	ADC1 TEC4	ADC1 TEC3	ADC1 TFRST	ADC1 TDPER	ADC1 TDC4	ADC1 TDC3	ADC1 TFPER	ADC1 TCPER	ADC1 TCC4	ADC1 TCC3	ADC1 TFC4	ADC1 TBRST	ADC1 TBPER	ADC1 TBC4	ADC1 TBC3
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v1	ADC1 TBC2	ADC1 TARST	ADC1 TAPER	ADC1 TAC4	ADC1 TAC3	ADC1 TAC2	ADC1 EEV5	ADC1 EEV4	ADC1 EEV3	ADC1 EEV2	ADC1 EEV1	ADC1 MPER	ADC1 MC4	ADC1 MC3	ADC1 MC2	ADC1 MC1
v2	ADC1 TFC3	ADC1 TARST	ADC1 TAPER	ADC1 TAC4	ADC1 TAC3	ADC1 TFC2	ADC1 EEV5	ADC1 EEV4	ADC1 EEV3	ADC1 EEV2	ADC1 EEV1	ADC1 MPER	ADC1 MC4	ADC1 MC3	ADC1 MC2	ADC1 MC1

表 13. HRTIM ADC のトリガ 2 レジスタとトリガ 4 レジスタ (HRTIM_ADC2R、HRTIM_ADC4R)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
v1	ADC2 TERST	ADC2 TEC4	ADC2 TEC3	ADC2 TEC2	ADC2 TDRST	ADC2 TDPER	ADC2 TDC4	ADC2 TDPER	ADC2 TDC2	ADC2 TCRST	ADC2 TCPER	ADC2 TCC4	ADC2 TCC3	ADC2 TCC2	ADC2 TBPER	ADC2 TBC4
v2	ADC2 TERST	ADC2 TEC4	ADC2 TEC3	ADC2 TEC2	ADC2 TDRST	ADC2 TDPER	ADC2 TDC4	ADC2 TFPER	ADC2 TDC2	ADC2 TCRST	ADC2 TCPER	ADC2 TCC4	ADC2 TFC4	ADC2 TCC2	ADC2 TBPER	ADC2 TBC4
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
v1	ADC2 TBC3	ADC2 TBC2	ADC2 TAPER	ADC2 TAC4	ADC2 TAC3	ADC2 TAC2	ADC2 EEV10	ADC2 EEV9	ADC2 EEV8	ADC2 EEV7	ADC2 EEV6	ADC2 MPER	ADC2 MC4	ADC2 MC3	ADC2 MC2	ADC2 MC1
v2	ADC2 TFC3	ADC2 TBC2	ADC2 TAPER	ADC2 TAC4	ADC2 TFC2	ADC2 TAC2	ADC2 EEV10	ADC2 EEV9	ADC2 EEV8	ADC2 EEV7	ADC2 EEV6	ADC2 MPER	ADC2 MC4	ADC2 MC3	ADC2 MC2	ADC2 MC1

付録 C 参照文献

このセクションでは、HRTIM 関連のドキュメントを製品別に挙げています。これらは www.st.com で入手できます。

アプリケーション・ノートのように、あらゆる製品に該当する情報と概念を収めたドキュメントもあります。

C.1 STM32F334x

STM32F334x4/x6/x8 データシート

RM0364 *STM32F334xx advanced Arm[®]-based 32-bit MCUs*

UM1733 *Getting started with STM32F334 Discovery kit*

UM1735 *Discovery kit for STM32F3 series with STM32F334C8 MCU*

UM1736 *Getting started with STM32F334 discovery kit software development tools*

AN4885 *High brightness LED dimming using the STM32F3348 Discovery kit*

AN4449 *Buck-boost converter using the STM32F334 Discovery kit*

AN4296 *Use STM32F3/STM32G4 CCM SRAM with IAR[™] EWARM, Keil[®] MDK-ARM and GNU-based toolchains*

C.2 STM32H74x / STM32H75x

STM32H743/753xI データシート

STM32H750xB データシート

RM0364 *STM32H743/753 and STM32H750 advanced Arm[®]-based 32-bit MCUs*

C.3 STM32G47x

STM32G471/3/4 データシート

STM32G484 データシート

RM0440 *STM32G4xx advanced Arm[®]-based 32-bit MCUs*

AN5094 *Migrating between STM32F334/303 lines and STM32G474xx/G431xx microcontrollers*

AN4767 *On-the-fly firmware update for dual bank STM32 microcontrollers / Software expansion for STM32Cube*

STM32G4 オンライン・トレーニング

C.4 その他

- AN4232 *Getting started with analog comparators for STM32F3 Series and STM32G4 Series devices*
- AN4013 *STM32 cross-series timer overview*
- AN4856 *STEVAL-ISA172V2 : 2 kW fully digital AC - DC power supply (D-SMPS) evaluation board*
- AN4930 *500 W fully digital AC-DC power supply (D-SMPS) evaluation board*
- UM2348 *Getting started with the STEVAL-DPSLLCK1 evaluation kit for the 3 kW full bridge LLC digital power supply*
- STEVAL-DPSLLCK1 : 3 kW Full Bridge LLC resonant digital power supply evaluation kit

改版履歴

表 14. 文書改版履歴

日付	版	変更内容
2014 年 6 月 19 日	1	初版発行
2019 年 3 月 26 日	2	概要、セクション 1: 準備、セクション 1.2: ハードウェアのセットアップ、セクション 1.3: ツールのセットアップ、セクション 1.5.1: システム・クロックの初期化、セクション 1.5.4: HRTIM の I/O 初期化、セクション 1.5.5: 他のペリフェラルの初期化、セクション 2: HRTIM の基本的な動作原理、セクション 2.2: 複数の PWM の生成、およびセクション 9: 他のサンプルを更新。セクション 1.4: HRTIM のバージョン、セクション 7: 多相降圧コンバータ、付録 A: HRTIM v2、付録 B: ソフトウェアの移行、付録 C: 参考文献、およびそのサブセクションを追加。 表 1: 対象とする製品を更新。 表 2: 組込み先の製品に応じた HRTIM の各種機能、表 3: HRTIM の入力周波数動作範囲、および表 5: タイマ分解能/最小 PWM 周波数 (fHRTIM = 170 MHz の場合) を追加。 ドキュメント全体で文章を軽微に編集。
2019 年 7 月 16 日	3	概要、セクション 2.2: 複数の PWM の生成、セクション 2.3: 他のタイミング・ユニットとマスタ・タイマの使用による PWM の生成、セクション 2.4: 任意の波形の生成、セクション 3: 電圧モード・デュアルバックコンバータ、セクション 4: 同期整流とフォールト保護を備えた電圧モードバックコンバータ、セクション 5: 非反転バック・ブーストコンバータ、セクション 6: 遷移モード力率コントローラ、セクション 7: 多相降圧コンバータ、および多相のデモ概要を更新。 セクション 8: デッドタイムが挿入されないサイクル単位の保護およびセクション 10: ソフトウェア・サンプルの入手先を追加。 表 3: HRTIM の入力周波数動作範囲を更新。 図 1: 基本的な PWM の生成、図 2: 基本的な PWM 信号を生成する HRTIM 設定、図 3: 複数の PWM 信号の生成、図 5: 任意の波形の生成、図 7: 電圧モードの降圧波形、ADC のサンプリングと割込み、図 8: PWM の割込みとレジスタの更新、図 9: 同期整流を備えた電圧モードバック、図 10: FAULT に伴うバック動作、図 11: 非反転バック・ブーストコンバータ、図 13: 降圧昇圧コンバータの動作波形、図 14: 遷移モード PFC、図 15: 過電流状態の Ton max における遷移モード PFC の動作、図 16: Toff max と Toff min における遷移モード PFC の動作、図 17: 5 相インターリーブ降圧コンバータ、図 18: 5 相インターリーブ降圧コンバータの制御波形、図 19: マスタ・タイマによる位相の無効化、図 20: 位相シェディングとバースト・モードによる低負荷管理、図 22: アクティブな位相の数に応じたサンプリング・ポイントの配置、図 23: ADC のサンプリング・ポイントの確実な正しい配置、図 24: ADC のサンプリング (5 相設定)、および図 25: ADC のサンプリング (3 相設定) を更新。
2020 年 1 月 09 日	4	表 1: 対象とする製品および表 8: サンプルの概要を更新。 ドキュメント全体で文章を軽微に編集。

表 15. 日本語版文書改版履歴

日付	版	変更内容
2022 年 5 月	1	日本語版初版発行

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。STの登録商標についてはSTウェブサイトをご覧ください。www.st.com/trademarks 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。

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。

© 2022 STMicroelectronics - All rights reserved