

STM32F7x0 バリュースライン、STM32H750 バリュースライン、STM32H7B0 バリュースライン、および STM32H730 バリュースライン・マイコンでの外部メモリコードの実行

概要

新しい複雑な機能を提供できるアプリケーションへの要求が高まっており、その結果、より大きな Flash メモリ領域を備えたデバイスに対する要求が高まっています。

外部 Flash メモリを使用することで、同等の性能レベルでより大きな記憶容量を実現するとともに、増加する Flash メモリ領域のニーズに対応するコスト効率の高いソリューションを提供します。

STM32F7x0 バリュースライン、STM32H750 バリュースライン、STM32H7B0 バリュースライン、および STM32H730 バリュースラインのデバイスは、内部 Flash メモリ領域を低減して市場の要求に応えます。

このアプリケーションノートでは、これらのバリュースラインデバイスで外部メモリからコードを実行するアプリケーションを構築するために必要な手順について説明します。

内部 Flash メモリからブートし、外部メモリからのユーザアプリケーションの実行にジャンプする方法の詳細を説明します。

関連ドキュメント

ST マイクロエレクトロニクスの Web サイト(www.st.com)から入手可能:

- STM32F7 シリーズ用 HAL、下位層ドライバ、専用ミドルウェア搭載 STM32Cube MCU パッケージ データブリーフ (DB2601)
- STM32H7 シリーズ用 HAL および専用ミドルウェア搭載 STM32Cube MCU パッケージ データブリーフ (DB3259)
- STM32F75xxx および STM32F74xxx advanced Arm®-based 32-bit MCU リファレンスマニュアル (RM0385)
- STM32H743/753 advanced Arm®-based 32-bit MCU リファレンスマニュアル (RM0433)
- STM32H7A3/B3 および STM32H7B0 バリュースライン advanced Arm®-based 32-bit MCU リファレンスマニュアル (RM0455)
- STM32H723/733、STM32H725/735 および STM32H730 バリュースライン advanced Arm®-based 32-bit MCU リファレンスマニュアル (RM0468)
- STM32F7 シリーズのシステムアーキテクチャと性能 アプリケーションノート (AN4667)
- STM32 マイクロコントローラの Quad-SPI (QSPI) インタフェース アプリケーションノート (AN4760)
- STM32H7x3 ハードウェア開発スタートガイド アプリケーションノート (AN4938)
- STM32F7 シリーズ MCU ハードウェア開発スタートガイド アプリケーションノート (AN4661)
- STM32 マイクロコントローラの Octo-SPI (OSPI) インタフェース アプリケーションノート (AN5050)
- STM32CubeProgrammer ソフトウェア説明ユーザマニュアル (UM2337)
- STM32H7B3LI MCU 搭載評価ボード ユーザマニュアル (UM2662)
- STM32H7 シリーズ用 STM32CubeH7 スタートガイド ユーザマニュアル (UM2204)

1 一般情報

本書は、Arm[®] ベースのデバイスに適用されます。

注 Arm は、米国内およびその他の地域にある Arm Limited (またはその子会社) の登録商標です。



2 外部メモリコードの実行の概要

2.1 外部メモリコードの実行原理

STM32CubeF7 v1.12.0 および STM32CubeH7 v1.8.0 ファームウェアパッケージには、内部 Flash メモリからブートする方法および外部メモリを設定してユーザアプリケーション(外部メモリ上にある)にジャンプする方法を実証するアプリケーションがいくつか用意されています。XiP と BootROM の 2 つのユースケースが使用できます。

- XiP ユースケースは、外部 Flash メモリ(QSPI/OSPI または FMC-NOR Flash メモリ)からの「eXecute in Place (インプレース実行)」を目的としたものです。ユーザアプリケーションコードは、ターゲット実行メモリアドレス(外部 QSPI/OSPI または FMC-NOR Flash メモリ)とリンクさせる必要があります。
- BootROM ユースケースは、内部 Flash メモリからブートし、外部 RAM メモリ(SDRAM、SRAM、または OSPI-RAM)を設定し、コード・ストレージ領域(SDCARD または SPI-Flash メモリ)からユーザアプリケーションのバイナリを外部 SDRAM、外部 SRAM、または外部 OSPI-RAM にコピーしてから、ユーザアプリケーションへジャンプする方法を示すことを目的としています。ユーザアプリケーションコードは、ターゲット実行メモリアドレス(外部 SDRAM、SRAM、または OSPI-RAM)とリンクさせる必要があります。

下の表で説明されているアプリケーションは、以下のボードの \Applications\ExtMem_CodeExecution にあるファームウェアパッケージから入手できます。

- 32F723EDISCOVERY ボード(STM32F730 デバイス用)
- STM32756G-EVAL ボード(STM32F750 デバイス用)
- STM32H743I-EVAL ボード(STM32H750 デバイス用)
- STM32H7B3I_EVAL および STM32H7B3I-DK ボード(STM32H7B0 デバイス用)
- STM32H735G-DK ボード(STM32H730 デバイス用)

表 1. アプリケーションの詳細

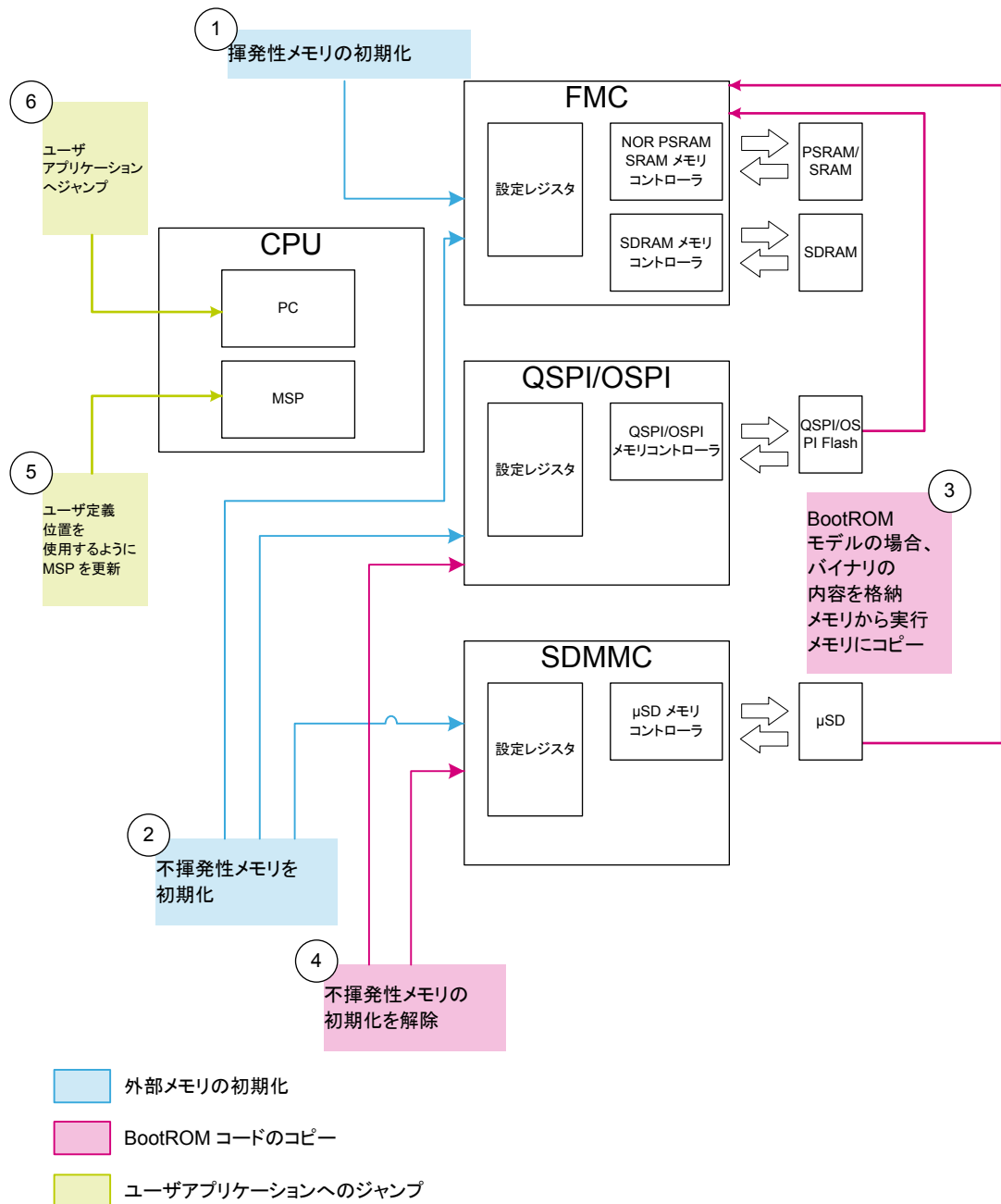
アプリケーション	説明
ExtMem_Boot	内部 Flash メモリからブートし、外部メモリを設定して、外部メモリ上にあるユーザアプリケーションにジャンプする方法を示します。 ユーザは、コード実行用に QSPI/OSPI Flash メモリ、FMC-NOR Flash メモリ、外部 SDRAM、外部 SRAM、または外部 OSPI-RAM を選択できます。
ExtMem_Application\LedToggling	外部 Flash メモリ(QSPI/OSPI Flash メモリまたは FMC-NOR Flash メモリ)、外部 SDRAM、外部 SRAM、または外部 OSPI-RAM から動作するサンプルアプリケーション
ExtMem_Application\FreeRTOS	外部 Flash メモリ(QSPI/OSPI Flash メモリまたは FMC-NOR Flash メモリ)、外部 SDRAM、外部 SRAM、または外部 OSPI-RAM から実行する FreeRTOS アプリケーションのサンプル

外部メモリブート アプリケーションでは、外部メモリを使用可能な状態にするために必要なリソースの初期化を行う必要があります。このアプリケーションは、ユーザの設定に従って必要なリソースを初期化します(セクション 3.3 設定を参照)。

外部メモリブート アプリケーションでは、メインスタックポインタをセットアップし、アプリケーションが外部メモリ上で実行されるように設定する必要があります。このタイプのブート方式によって、かなりのサイズのユーザアプリケーションに対応できます。

外部メモリブート アプリケーションでは、ユーザアプリケーションにジャンプする前に、セットアップ段階の後に不要になったリソースをリセットまたは解放するようにします。このブート方式の概要を次の図に示します。

図 1. 外部メモリコードのブート方式



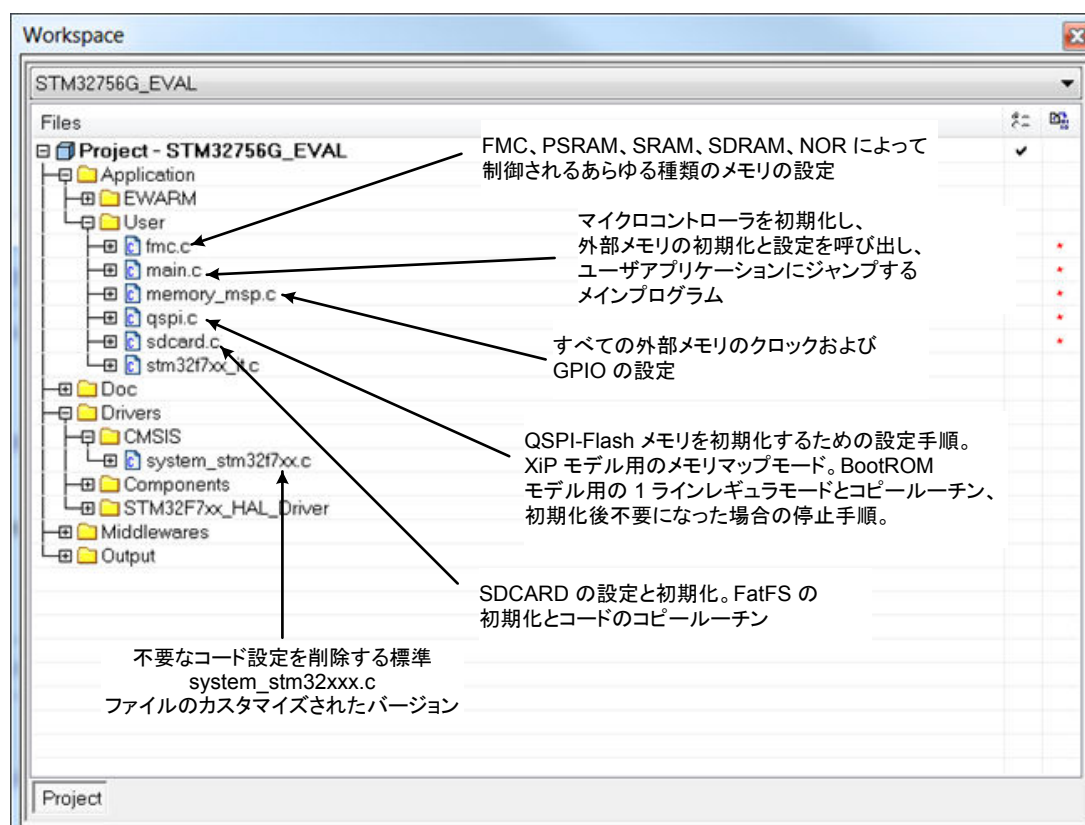
2.2 外部メモリブートアプリケーションの説明

外部メモリブート アプリケーションには、STM32CubeF7/H7 パッケージ上に、各ハードウェアプラットフォームでサポートされている設定に合わせて調整されたソースファイルのセットが含まれています。

STM32H7B0 バリュースラインデバイスおよび STM32H730 バリュースラインデバイスでは、外部メモリ用のクロックと GPIO の設定は、各メモリごとに BSP ドライバ内で行われます。qspi.c ファイルは ospi.c ファイルに置き換えられています。

次の図に、サポートされているすべての設定に対するすべてのファイルのスーパーセットの例を示します。

図 2. 外部メモリブートアプリケーションのソースファイルのスーパーセット



3 サポートされているブートモデル

アプリケーションは、以下の 2 種類の実行モデルをサポートしています。

- インプレース実行のサポート (XiP のサポート)
- BootROM のサポート

ユーザは、memory.h ヘッダファイルを調整して、ニーズに合わせた設定を選択する必要があります。

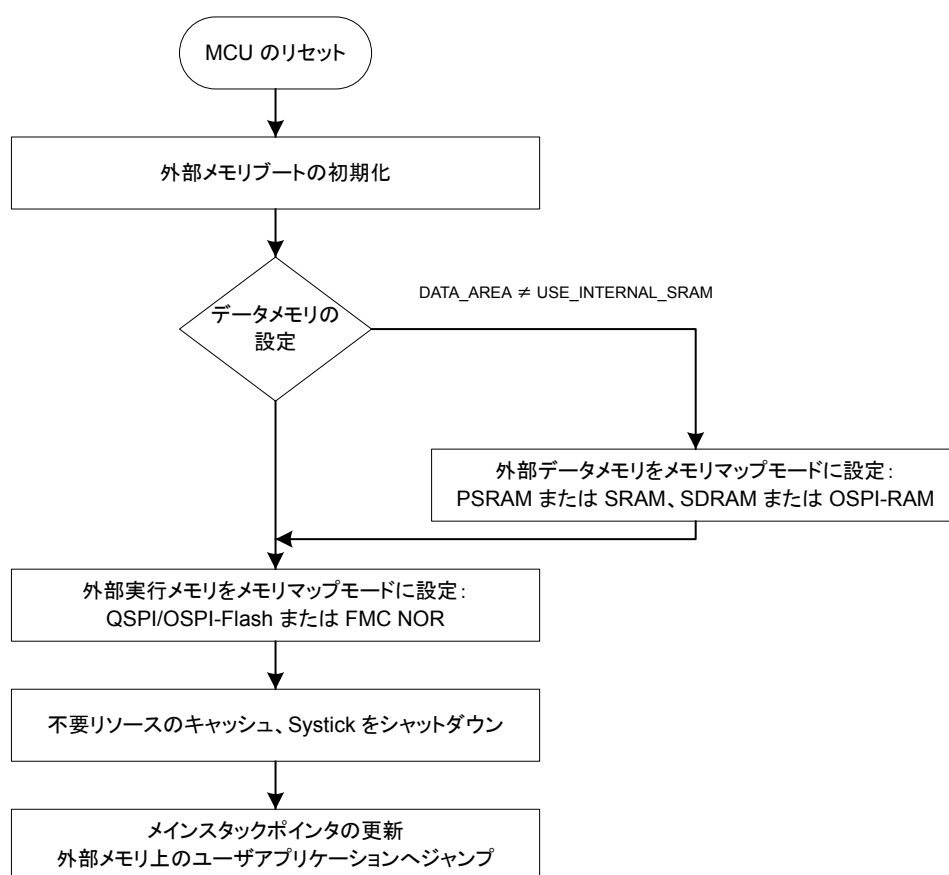
3.1 インプレース実行 (XiP) のサポート

XiP モデルは、コード・ストレージに使用される外部不揮発性メモリからの直接のコード実行に基づいています。この実行モデルでは、実行コード・ユーザアプリケーションへの直接アクセスを CPU に許すメモリマップのサポートが必要です。XiP モデルは、FMC/QSPI/OSPI インタフェースを介して外部 NOR/QSPI/OSPI Flash メモリで使用できます。

memory.h ファイルのユーザ設定に基づき、外部メモリブート アプリケーションが SDRAM、SRAM、PSRAM、OSPI-RAM、または内部 SRAM の揮発性メモリのうちの 1 つを設定します。このモデルでは、揮発性メモリはデータ用としてのみ使用されます。

次のフローチャートに、XiP モデルの動作フローを示します。

図 3. XiP モデルの動作フロー



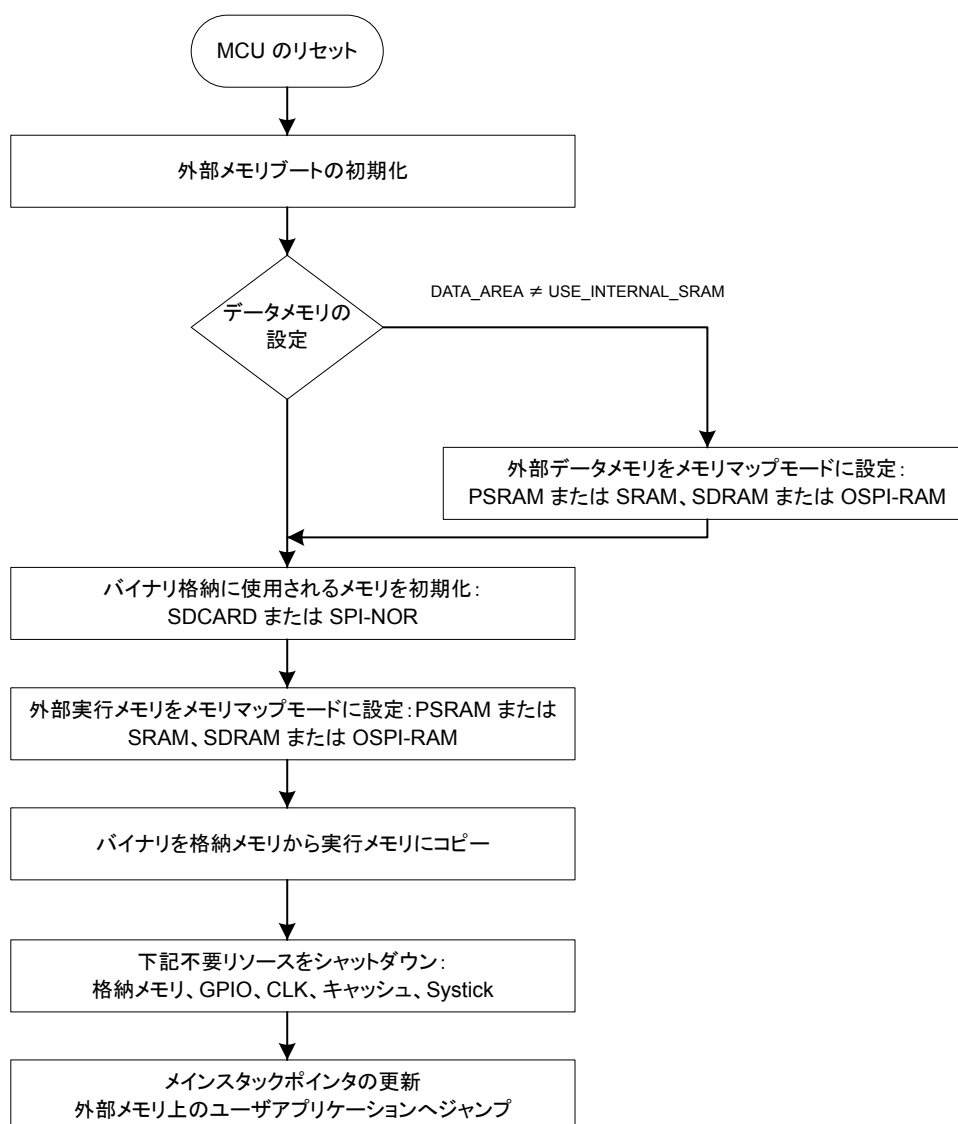
3.2 BootROM のサポート

BootROM モデルは、選択した揮発性メモリからのコード実行に基づいています。この実行モデルは、SDCARD のように、メモリマップインタフェースを持たないメモリにバイナリデータが格納される場合に適しています。このモデルは、SPI-NOR (1-ラインで QSPI/OSPI を使用してエミュレートされる) の場合など、バイナリデータが低スループットでメモリに格納される場合にも適切です。

memory.h ファイルのユーザ設定に基づき、外部メモリブート アプリケーションが SDRAM、SRAM、PSRAM、OSPI-RAM または内部 SRAM の揮発性メモリのうちの 2 つを設定します。このモデルでは、バイナリデータは、外部メモリブート アプリケーションによって実行される前に、不揮発性メモリから 1 つの揮発性メモリにコピーされます。2 番目の揮発性メモリはデータ用に使用されます。

次のフローチャートに、BootROM モデルの動作フローを示します。

図 4. BootROM モデルの動作フロー



3.3 設定

ユーザの設定は、次の定義によって定義されます。

- DATA_AREA: データの保持に使用する揮発性メモリの指定に使用します。サポートされているメモリ(使用するボードによる)は次のとおりです。
 - USE_EXTERNAL_SDRAM: データ保持には外部 SDRAM が使用されます。
 - USE_EXTERNAL_SRAM: データ保持には外部 SRAM が使用されます。
 - USE_EXTERNAL_PSRAM: データ保持には外部 PSRAM が使用されます。
 - USE_INTERNAL_SRAM: データ保持には内部 SRAM が使用されます。
 - USE_EXTERNAL_RAM: データ保持には外部 OSPI-RAM が使用されます。
- CODE_AREA: ユーザアプリケーションの実行位置の指定に使用します。この領域は、BootROM 方式の場合は揮発性メモリ、XiP 方式の場合は不揮発性メモリになります。サポートされているメモリは(使用するハードウェアによる)次のとおりです。
 - XiP モデル: BINARY_AREA は未定義にする必要があります。
 - USE_QSPI: コード実行には QSPI Flash が使用されます。
 - USE_OSPI: コード実行には OSPI Flash が使用されます。
 - USE_NOR: コード実行には FMC-NOR が使用されます。
 - BootROM モデル: BINARY_AREA を定義する必要があります。
 - USE_EXTERNAL_SDRAM: コード実行には外部 SDRAM が使用されます。
 - USE_EXTERNAL_SRAM: コード実行には外部 SRAM が使用されます。
 - USE_EXTERNAL_PSRAM: コード実行には外部 PSRAM が使用されます。
 - USE_INTERNAL_SRAM: コード実行に内部 SRAM が使用されます。
 - USE_EXTERNAL_RAM: コード実行には外部 OSPI-RAM が使用されます。
- BINARY_AREA: BootROM モデルでのみ定義します。ユーザアプリケーションを含むバイナリの場所を指定するために使用します。選択した設定によっては、追加の定義が必要です。サポートされているメモリ(使用するハードウェアによる):
 - USE_SPI_NOR: バイナリ格納には SPI NOR Flash が使用されます。
 - BINARY_BASE_OFFSET: SPI NOR Flash 内のバイナリのオフセット
 - BINARY_SIZE: バイナリイメージのサイズ
 - USE_SDCARD: バイナリス格納には SD Card が使用されます。
 - BINARY_FILENAME: 実行されるバイナリファイルの名前

ユーザは、少なくともユーザアプリケーションの適切な起動に必要なコードとデータが選択したメモリに含まれるようにする必要があります。その後、ユーザアプリケーションで、必要なその他のメモリを初期化できます。

3.4 外部メモリの品名一覧

次の表に、ボードおよびブートモデルによって使用される外部メモリの品名を示します。STM32F7x0 バリュールライン、STM32H750 バリュールライン、STM32H7B0 バリュールライン、および STM32H730 バリュールラインのデバイス専用のボードはないので、使用するボード(互換性のあるデバイス搭載)は以下のとおりです。

- STM32F730 デバイスをエミュレートするために 32F723EDISCOVERY を使用します。
- STM32F750 デバイスをエミュレートするために STM32F756G-EVAL を使用します。
- STM32H750 デバイスをエミュレートするために STM32H743I-EVAL を使用します。
- STM32H7B0 デバイスをエミュレートするために STM32H7B3I-EVAL と STM32H7B3I-DK を使用します。
- STM32H730 デバイスをエミュレートするために STM32H735G-DK を使用します。

表 2. 各ボード上で使用されるブートモデル別の外部メモリ

ブートモデル	メモリ	STM32F730 デバイス搭載の 32F723EDISCOVERY	STM32F750 デバイス搭載の STM32F756G-EVAL	STM32H750 デバイス搭載の STM32H743I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-DK	STM32H730 デバイス搭載の STM32H735G-DK
XiP	QSPI:OSPI Flash メモリ	MX25L51245GZ2I-08G (バス幅:4 本ライン)	N25Q512A13GSF40E (バス幅:4 本ライン)	2 個の Quad-SPI Flash MT25QL512ABB8ESF-0SIT または ツイン Quad-SPI Flash MT25TL01GHBB8ESF-0SIT 1 個 (バス幅:8 本ライン)	Octo-SPI フラッシュ NOR MX25LM51245G (バス幅:8 本ライン)	Octo-SPI フラッシュ NOR MX25LM51245G (バス幅:8 本ライン)	Octo-SPI フラッシュ NOR MX25LM51245G (バス幅:8 本ライン)
	NOR Flash メモリ(FMC 上)	-	PC28F128M29E WL A (バス幅:16 ビット)	MT28EW128ABA1 L PC-OSIT	MT28EW128ABA1 LPC-OSIT	-	-
BootROM	SPI-NOR (QSPI 1 ラインでエミュレート)	-	N25Q512A13GSF40E (バス幅:2 本ライン)	2 個の Quad-SPI Flash MT25QL512ABB8ESF-0SIT または ツイン Quad-SPI Flash MT25TL01GHBB8ESF-0SIT 1 個 (バス幅:2 本ライン)	Octo-SPI フラッシュ NOR MX25LM51245G (バス幅:1 本ライン)	Octo-SPI フラッシュ NOR MX25LM51245G (バス幅:1 本ライン)	Octo-SPI フラッシュ NOR MX25LM51245G (バス幅:1 本ライン)
	SDCARD	-	ネイティブサポート	トランシーバ IP4856CX25/C_Module_REV	HSP051-4M10	HSP051-4M10	HSP051-4M10
揮発性メモリ	内部 SRAM	ネイティブサポート	ネイティブサポート	ネイティブサポート	ネイティブサポート	ネイティブサポート	ネイティブサポート
	外部 SRAM	-	IS61WV102416BL L -10ML I (バス幅:16 ビット)	IS61WV102416BL L -10ML I (バス幅:16 ビット)	IS42S32800J-6B LI (バス幅:32 ビット)	-	-
	外部 SDRAM	-	IS42S32800G-6BL I (バス幅:32 ビット)	IS42S32800G-6BL I (バス幅:32 ビット)	IS42S32800J-6B LI (バス幅:32 ビット)	IS42S32800J-6B LI (バス幅:16 ビット)	-
	外部 PSRAM	IS66WV51216EBLL-55BLI	-	-	-	-	-

ブートモデル	メモリ	STM32F730 デバイス搭載の 32F723EDISCOVERY	STM32F750 デバイス搭載の STM32756G-EVAL	STM32H750 デバイス搭載の STM32H743I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-DK	STM32H730 デバイス搭載の STM32H735G-DK
揮発性メモリ	外部 OSPI-RAM	(バス幅: 16 ビット) -	-	-	-	-	OSPI HyperRAM S70KL1281

4 考慮すべきリソースの制約

初期化後に不要になったリソース（割込み、進行中の転送、未使用ピン）は、ユーザアプリケーションにジャンプする前に解放する必要があります。余分な電力消費を防ぎ、ユーザアプリケーションへの干渉を制限するために、これを行う必要があります。特に BootROM モデルの場合、バイナリの格納に使用されたペリフェラルは不要なので、リセットする必要があります。

ユーザは、外部メモリブートアプリケーションによって使用されるリソースの量を考慮する必要があります。外部メモリインタフェースが起動して実行中のままであることを保証するためです。リソース制約は以下のものに関連しています。

- ピンの割り当てと設定
- インタフェースの設定（QSPI/OSPI IP レジスタは変更してはなりません。FMC IP レジスタは部分的に更新できます）
- 有害な形で IP リセットクロックを無効化することやクロック周波数／ソースを更新することを回避するための RCC の設定

以下のピン割り当て表は参考用ですが、使用するボードに応じたピンの選択に役立ちます。使用可能なオルタネート機能に基づいて、他のピン選択を使用することもできます。

表 3. 各メモリのボードごとのピン配置

メモリ/ボード	STM32F730 デバイス搭載の 32F723EDISCOVERY	STM32F750 デバイス搭載の STM32756G-EVAL	STM32H750 デバイス搭載の STM32H743I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-DK	STM32H730 デバイス搭載の STM32H735G-DK
QSPI/OSPI Flash メモリ	PB(2,6)、PC(9,10)、PD13、PE2	PB(2,6)、PF(6,8,9)	PB2、PC11、PF(6,7,8,9)、PG(6,9,14)、PH(2,3)	PB2、PC(1,5)、PD6、PF(6,7,8,9)、PG(6,14)、PH3	PG(6,9)、PB2、PC(1,5)、PD(11,7)、PF(6,7,9)、PH3	PB2、PD(4,5,7,11,12,13)、PE2、PG(6,9)、PF10
NOR Flash メモリ (FMC 上)	-	PD(0,1,4,5,6,7,8,9,10,11,12,13,14,15) PE(2,3,4,5,6,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,12,13,14,15) PG(0,1,2,3,4,5)	PD(0,1,4,5,6,7,8,9,10,11,12,13,14,15) PE(2,3,4,5,6,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,12,13,14,15) PG(0,1,2,3,4,5)	PC(6,7)、PD(0,1,4,5,8,9,10,11,12,13,14,15)、PE(0,1,7,8,9,10,11,12,13,14,15)、PF(0,1,2,3,4,5,12,13,14,15)、PG(0,1,2,3,4,5)	-	-
外部 SRAM	-	PD(0,1,3,4,5,8,9,10,11,12,13,14,15) PE(0,1,3,4,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,12,13,14,15) PG(0,1,2,3,4,5,10)	PD(0,1,3,4,5,8,9,10,11,12,13,14,15) PE(0,1,3,4,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,12,13,14,15) PG(0,1,2,3,4,5,6,9,10,12,13,14)	PD(0,1,4,5,8,9,10,11,12,13,14,15)、PE(0,1,7,8,9,10,11,12,13,14,15)、PF(0,1,2,3,4,5,12,13,14,15)、PG(0,1,2,3,4,5,10)	-	-
外部 SDRAM	-	PD(0,1,8,9,10,14,15) PE(0,1,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,11,12,13,14,15) PG(0,1,4,5,8,15) PH(2,3,5,8,9,10,11,12,13,14,15) PI(0,1,2,3,4,5,6,7,9,10)	PD(0,1,8,9,10,14,15) PE(0,1,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,11,12,13,14,15) PG(0,1,2,3,4,5,8,15) PH(5,6,7,8,9,10,11,12,13,14,15) PI(0,1,2,3,4,5,6,7,9,10)	PA7、PD(0,1,8,9,10,14,15)、PE(0,1,7,8,9,10,11,12,13,14,15)、PF(0,1,2,3,4,5,11,12,13,14,15)、PG(0,1,4,5,8,15)、PH(6,7)	PD(0,1,8,9,10,14,15)、PE(0,1,7,8,9,10,11,12,13,14,15)、PF(0,1,2,3,4,5,11,12,13,14,15)、PG(0,1,4,5,8,15)、PH(5,6,7)	-
外部 PSRAM	PD(0,1,4,5,7,8,9,10,11,12,14,15) PE(0,1,7,8,9,10,11,12,13,14,15) PF(0,1,2,3,4,5,12,13,14,15) PG(0,1,2,3,4,5)	-	-	-	-	-
外部 OSPI-RAM	-	-	-	-	-	PF(0,1,3,4,12)、PG(0,1,10,11,12)

次の表に、変更しないで保持する必要があるリソースを示します。外部ストレージが使用不可になることを避けるために変更してはいけないペリフェラル(またはペリフェラルの一部)の一覧を記載しています。記載のペリフェラルは、リセットやクロックを無効にしたり、動作を変更するように再設定したりしてはなりません。

注 いくつかの要素は、選択したボード用に選択された外部メモリブートアプリケーションの設定およびプラットフォームのハードウェアに応じて変わることがあります。

注 外部ローダでは、外部メモリがすべてツインクワッド SPI Flash である限り、シングル QSPI モードはサポートされません。

表 4. メモリタイプごとに必要なペリフェラル

	STM32F730 デバイス搭載の 32F723EDISCOVERY	STM32F750 デバイス搭載の STM32756G-EVAL	STM32H750 デバイス搭載の STM32H743I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-EVAL	STM32H7B0 デバイス搭載の STM32H7B3I-DK	STM32H730 デバイス搭載の STM32H735G-DK
QSPI/OSPI Flash メモリ	QSPI1 (0x9000000)。	QSPI1 (0x9000000)。	デュアル QSPI モード QSPI は、このアプリケーションではサポートされていません	OSPI (8 進モード)	OSPI (8 進モード)	OSPI (8 進モード)
NOR 型 Flash (FMC 上)	-	FMC-NOR (FMC-PSRAM / FMC-SRAM なし)	FMC-NOR (FMC-PSRAM / FMC-SRAM なし)	FMC-NOR (FMC-SRAM なし)	-	-
外部 SRAM	-	FMC-SRAM (FMC-PSRAM / FMC-NOR なし)	FMC-SRAM (FMC-PSRAM / FMC-NOR なし)	FMC-SRAM (FMC-NOR なし)	-	-
外部 SDRAM	-	FMC-SDRAM (最大システム周波数 200 MHz)	FMC-SDRAM	FMC-SDRAM	FMC-SDRAM	-
外部 PSRAM	FMC-SRAM (FMC-SRAM / FMC-NOR なし)	-	-	-	-	-
外部 OSPI-RAM	-	-	-	-	-	OSPI (リニアバースト)

5 外部メモリユーザアプリケーションの説明

5.1 必要な更新

外部メモリアプリケーションは標準とは異なる特定のブート方式に基づいており、オンチップアプリケーションからオフチップアプリケーションへのスムーズな移行をサポートしています。

アプリケーションの場所が変更されたので、ユーザが行わなければならない 2 つの更新があります。

- 選択したブートオプションに対応するメモリ配置で、必要なリンクファイルが使用されるようにしてください。
- 正しいアドレスを使用するように VTOR の設定を更新してください。

5.2 ロードとデバッグ

5 つのボード 32F723EDISCOVERY、STM32756G-EVAL、STM32H743I-EVAL、STM32H7B3I-EVAL、STM32H7B3I-DK、および STM32H735G-DK には、外部不揮発性メモリ用のローダがあります。これらのローダは、STM32CubeF7/H7 内で次の形で提供されます。

- EWARM IDE 用のパッチ (EWARM v8.50.6 以降を使用する場合は不要)
- MDK-ARM IDE 用の専用バック

XiP モデルでは、内部 Flash デバッグと同様のシームレスなロードおよびデバッグ体験が得られます。SW4STM32 IDE および STM32Cube IDE の場合、外部 Flash メモリにアプリケーションをロードするには、STM32CubeProgrammer を使用する必要があります。

BootROM モデルでは、アプリケーションは以下の外部揮発性メモリから実行するためにコンパイル・リンクされます。

- 外部 SDRAM: STM32H750 バリユーラインおよび STM32H7B0 バリユーラインのリンクアドレスは 0xD0000000、STM32F7x0 バリユーラインのリンクアドレスは 0xC0000000
- 外部 SRAM: STM32H750 バリユーライン、STM32H7B0 バリユーライン、および STM32F7x0 バリユーラインのリンクアドレスは 0x68000000
- 外部 OSPI-RAM: STM32H730 バリユーラインのリンクアドレスは 0x70000000

その後、アプリケーションバイナリを SPI_NOR Flash メモリまたは SDCARD に格納する必要があります。ユーザアプリケーションを格納領域から実行 RAM 領域にコピーするのはブートアプリケーションの仕事です。

結果として、アプリケーションのロード方式を IDE (MDK-ARM または EWARM) の外部メモリ Flash ローダで処理することはできません (アプリケーションのリンクアドレスと格納アドレスが異なるため)。

BINARY_AREA 定義 (ブートアプリケーションの "memory.h" ファイルで指定) に応じて、このモデルでは、以下の 2 つの異なるロード方式を使用する必要があります。

- SPI_NOR
ユーザアプリケーションは、SPI-NOR Flash メモリのアドレス 0x90000000 に格納する必要があります。これは、STM32CubeProgrammer を使用して行う必要があります。SPI-Flash アドレスなどの異なるロードアドレスを指定できるようにするには、アプリケーションの出力はバイナリ形式である必要があります。詳細については、以下の図を参照してください。
- SDCARD
ユーザは、ビルドの出力であるバイナリファイルを、ユーザアプリケーションの格納に使用する SDCARD に手動でコピーして、SDCARD を評価ボードに挿入する必要があります。

以下の図に、ロードとデバッグの手順を示します。

5. STM32CubeProgrammer

1. Connect

2. Download

3. External loaders

4. File path: \\Applications\\ExtMem_CodeExecution\\ExtMem_Application\\LedToggling\\EWARM\\XIP_OSPI_InternalSRAM\\Exe\\Project.bin

5. Start address: 0x90000000

6. Start Programming

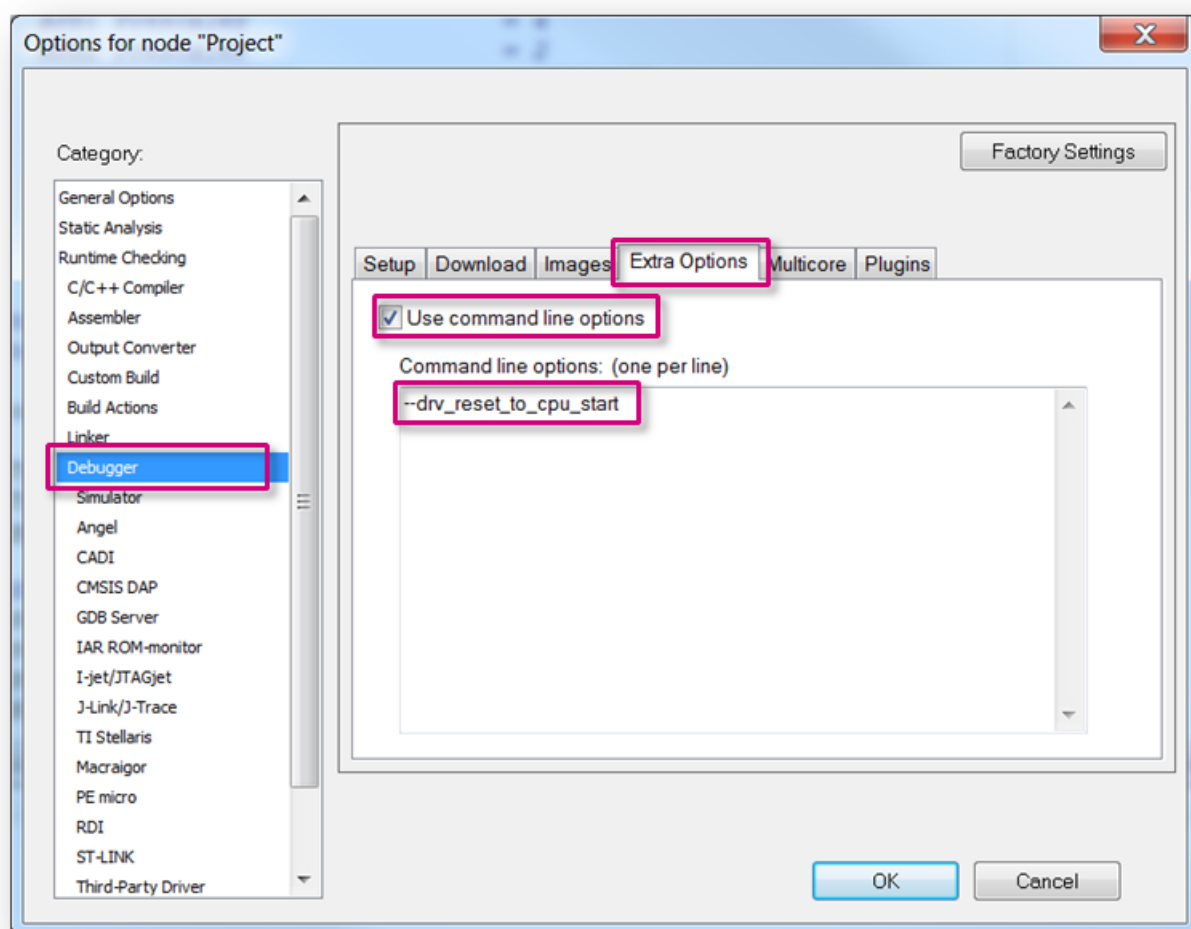
External loaders	Device	Memory	Start Address	Size	Access	Flash
<input checked="" type="checkbox"/>	MT25T101G_STM32H7xxI-EVAL	STM32H7xxI-EVAL	0x90000000	128M	0x100	NOR_FLASH
<input checked="" type="checkbox"/>	MT25QL512A_STM32756G-EVAL	STM32756G-EVAL	0x90000000	64M	0x10	NOR_FLASH
<input checked="" type="checkbox"/>	MX25LM51245G_STM32H7B3I-EVAL	STM32H7B3I-EVAL	0x90000000	64M	0x1000	NOR_FLASH
<input checked="" type="checkbox"/>	MX25LM51245G_STM32H7B3I_DISCO	STM32H7B3I	0x90000000	64M	0x1000	NOR_FLASH
<input checked="" type="checkbox"/>	MX25LM51245G_STM32H735G-DK	STM32H735G-DK	0x90000000	64M	0x1000	NOR_FLASH

5.3 EWARM IDE を使用したデバッグ

EWARM IDE で、外部メモリから実行中のユーザアプリケーションをデバッグする場合、特に注意が必要です。EWARM は、PC (プログラムカウンタ) のデフォルトの CPU リセット値を、ユーザアプリケーションで指定された値 (外部実行メモリ内のアドレス値) で上書きします。

このブート方式では、ユーザアプリケーションの PC アドレスは、外部メモリブートアプリケーションが実行される (つまり、外部メモリの準備が整い、FMC または QSPI/OSPI を介してメモリがマップされる) まで、アクセスできないままです。EWARM がユーザアプリケーションの開始ポイントに直接ジャンプすると、ハードフォールトが生成されます。ハードフォールトを回避するには、下の図に示すように、ユーザはデバッグオプションに「--drv_reset_to_cpu_start」コマンドラインを追加する必要があります。この設定により、EWARM が PC を強制的に設定するのを防ぎ、ユーザアプリケーションにジャンプする前に、外部メモリブートアプリケーションに外部メモリを設定するように要求します。

図 6. デバッガのコマンドラインオプション



6 性能の特性評価

外部メモリから実行した場合、外部 Flash メモリの遅延と、命令／データパスが長くなることにより、性能に影響が出ます。STM32F7x0 バリユーライン、STM32H750 バリユーライン、および STM32H7B0 バリユーラインのデバイスを使用すると、Cortex-M7 L1 キャッシュのおかげでこの影響は軽減されます。

次の表に、ROM/RAM の組み合わせごとに達成された EEMBC® CoreMark® スコアの概要を示します。内部 Flash メモリから実行したとき、最高の性能が得られます。それでも、外部メモリから実行する場合の損失は大幅に低減されています。

これらの図は、外部メモリから動作させた場合の CPU 性能への影響を示しています。内部 Flash 構成のスコアは、参考用として提供されています。

表 5. 構成別の EEMBC® CoreMark® スコア

ROM メモリ	RAM メモリ	STM32F730 デバイス搭載の 32F723EDISCOVERY (I/D キャッシュ 8K/8K)	STM32F750 デバイス搭載の STM32756G-EVAL (I/D キャッシュ 4K/4K)	STM32H750 デバイス搭載の STM32H743I-EVAL (I/D キャッシュ 16K/16K)	STM32H7B0 デバイス搭載の STM32H7B3I-EVAL (I/D キャッシュ 16K/16K)	STM32H730 デバイス搭載の STM32H735G-DK (I/D キャッシュ 16K/16K)
QSPI/OSPI Flash メモリ	内部 SRAM	1089	948	2020	1414	2777
	外部 SRAM	-	940	1972	1380	-
	外部 SDRAM	-	871	1972	1380	-
	外部 PSRAM	1079	-	-	-	-
	外部 RAM	-	-	-	-	2713
NOR Flash メモリ	内部 SRAM	-	906	2020	1414	-
	外部 SRAM	-	899	1972	1380	-
	外部 SDRAM	-	833	1972	1380	-
外部 SRAM	内部 SRAM	-	1016	2020	1414	-
外部 SDRAM	内部 SRAM	-	989	2020	1414	-
外部 RAM	内部 SRAM	-	-	-	-	2777
内部 Flash	内部 SRAM	1092	1082	2020	1414	2777

改版履歴

表 6. 文書改版履歴

日付	版	変更内容
2018 年 7 月 11 日	1	初版発行
2020 年 6 月 16 日	2	<p>ドキュメント全体に STM32H7B0 バリユーラインを追加。 表紙を更新。 セクション 2.1 外部メモリコードの実行原理を更新。 セクション 2.2 外部メモリブートアプリケーションの説明を更新。 セクション 3.1 インプレース実行 (XiP) のサポートを更新。 セクション 3.3 設定を更新。</p> <p>下記に、STM32H7B0 デバイス搭載 STM32H7B3I-EVAL および STM32H7B3I-DK を追加。</p> <ul style="list-style-type: none"> 表 2. 各ボード上で使用されるブートモデル別の外部メモリを参照してください。 表 3. 各メモリのボードごとのピン配置を参照してください。 表 4. メモリタイプごとに必要なペリフェラルを参照してください。 セクション 5.2 ロードとデバッグを更新。 表 5. 構成別の EEMBC® CoreMark® スコアを参照してください。
2020 年 9 月 3 日	3	<p>ドキュメント全体に STM32H730 バリユーライン、外部 OSPI-RAM、STM32H735G-DK を追加：</p> <ul style="list-style-type: none"> 表紙を更新。 セクション 2.1 外部メモリコードの実行原理を更新。 セクション 2.2 外部メモリブートアプリケーションの説明を更新。 セクション 3.1 インプレース実行 (XiP) のサポートを更新。 セクション 3.2 BootROM のサポートを更新。 セクション 3.3 設定を更新。 表 2. 各ボード上で使用されるブートモデル別の外部メモリを更新。 表 3. 各メモリのボードごとのピン配置を更新。 表 4. メモリタイプごとに必要なペリフェラルを更新。 セクション 5.2 ロードとデバッグを更新。 表 5. 構成別の EEMBC® CoreMark® スコアを更新。

目次

1	一般情報	2
2	外部メモリコードの実行の概要	3
2.1	外部メモリコードの実行原理	3
2.2	外部メモリブートアプリケーションの説明	5
3	サポートされているブートモデル	6
3.1	インプレース実行 (XiP) のサポート	6
3.2	BootROM のサポート	6
3.3	設定	8
3.4	外部メモリの品名一覧	9
4	考慮すべきリソースの制約	11
5	外部メモリユーザアプリケーションの説明	14
5.1	必要な更新	14
5.2	ロードとデバッグ	14
5.3	EWARM IDE を使用したデバッグ	15
6	性能の特性評価	17
	改版履歴	18

表一覧

表 1.	アプリケーションの詳細	3
表 2.	各ボード上で使用されるブートモデル別の外部メモリ	9
表 3.	各メモリのボードごとのピン配置	12
表 4.	メモリタイプごとに必要なペリフェラル	13
表 5.	構成別の EEMBC® CoreMark® スコア	17
表 6.	文書改版履歴	18

図一覧

図 1.	外部メモリコードのブート方式	4
図 2.	外部メモリブートアプリケーションのソースファイルのスーパーセット	5
図 3.	XiP モデルの動作フロー	6
図 4.	BootROM モデルの動作フロー	7
図 5.	STM32CubeProgrammer	15
図 6.	デバッガのコマンドラインオプション	16

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。

© 2023 STMicroelectronics – All rights reserved