

注意: この日本語版文書は参考資料としてご利用ください。 最新情報は必ずオリジナルの英語版をご参照願います。

# AN3767

# MPLAB Harmony v3を使った SAM E54 MCUのライブ アップデート アプリケーション

## はじめに

SAM E54 MCU(マイクロコントローラ)のデュアルバンク フラッシュを使うと、アプリケーションにライブ アップ デート機能を実装できます。アプリケーションにおけるライブ アップデート機能とは、内部フラッシュまたは NVM(不揮発性メモリ)内の非アクティブバンクにアプリケーション コード (ファームウェア)を書き込むものです。 ライブ アップデート機能を使うと、アクティブバンク上で動作中のアプリケーションに影響を及ぼす事なく、新し いファームウェアに更新する事ができます。ライブ アップデート機能が実行中も、現在実行中のアプリケーション の動作を継続できます。

ライブ アップデート アプリケーションの特長は以下の通りです。

- デュアルバンク フラッシュを備えたSAM E54 MCUはバンクAとバンクBに分かれており、それぞれにブート ローダおよびアプリケーション コードセクションがあります。
- 最初はUARTフェイルセーフ ブートローダがバンクAに書き込まれ、リセット後アプリケーションはバンクAから実行されます。その後、ライブ アップデート アプリケーションとUARTフェイルセーフ ブートローダがスクリプトでマージされ、UARTフェイルセーフ ブートローダを使ってバンクBに書き込まれます。
- ライブ アップデート アプリケーションはUART等のシリアル インターフェイス経由でホストプログラム(通常 はPC上で実行)と通信し、ファームウェアを受信します。
- ライブ アップデート機能を実装するコードはファームウェア更新が要求されている時のみ実行されます。それ 以外の時はアイドル状態で、他のアプリケーション機能を実行できます。

本書では、デュアルバンク フラッシュを備えたSAM E54 MCUにライブ アップデート機能を実装する方法を説明します。

# 目次

はし	こめに	1
1.	ハードウェアおよびソフトウェア要件	3
	1.1 SAM E54 Curiosity Ultra開発ボード	3
	1.2 MPLAB <sup>®</sup> X IDE(統合開発環境)とXCコンパイラ	3
	1.3 MPLAB Harmony v3	3
	1.4 Python	3
2.	設計	4
	2.1 UARTライブ アップデート プロトコル	5
	2.2 コマンドの説明	5
	2.3 通信タスク	7
	2.4 コマンド処理タスク	7
	2.5 プログラミング タスク	7
	2.6 メモリレイアウト	8
	2.7 実行フロー	9
3.	設定	
0.	3.1 ブートローダ リンカスクリプト	
	3.2 MHCの設定	14
	3.3 プロジェクトの設定	18
4.	アプリケーションの実行	
	4.1 ブートローダ アプリケーションの実行	
	4.2 ライブ アップデート アプリケーションの実行	23
5.	まとめ	26
0		
6.	<b></b>	
Mic	xrochip社ウェブサイト	28
製品	品変更通知サービス	28
おそ	客様サポート	28
Mic	rochip社のデバイスコード保護機能	
注分	<b>ま</b> との注意と	20
<i>ا</i> تتر	≠⊥♥ノノニ芯⊼	29
商机	票29	
品質	<b>寳管理システム</b>	30
各国	国の営業所とサービス	31

# 1. ハードウェアおよびソフトウェア要件

### 1.1 SAM E54 Curiosity Ultra開発ボード

SAM E54 Curiosity Ultra開発ボードはSAM E54 MCUを評価するための開発キットです。SAM E54は120 MHzで動作 するArm® Cortex®-M4コアを採用しています。SAM E54 Curiosity Ultra開発ボードはプログラマとデバッガを実装し ており、ハードウェアを追加しなくても開発を始める事ができます。MikroElectronika mikroBUS™ Click™アダプタ ボードによる機能拡張、Microchip社製PHYドータボードによるEthernet接続機能の追加、Microchip社製拡張ボード によるWi-Fi™接続機能の追加、Microchip社製オーディオ ドータボードによるオーディオ入出力機能の追加が可能で す。拡張ボードの有無に関わらず、SAM E54 Curiosity Ultra開発ボードを使うと、Bluetooth®オーディオ、CAN、 GUI(グラフィカル ユーザ インターフェイス)、IoT(モノのインターネット)、ロボット開発、コンセプト実証の設計 等、各種アプリケーションの自由な開発が可能です。

SAM E54 Curiosity Ultra開発ボードはMicrochip社オンラインストアでご購入になれます。

### 1.2 MPLAB<sup>®</sup> X IDE(統合開発環境)とXCコンパイラ

MPLAB X IDEは拡張可能で柔軟な設定が可能なソフトウェア プログラムです。このIDEはほとんど全てのMicrochip 社製MCUをサポートし、組み込み回路の研究、設定、開発、デバッグ、評価に役立つ強力なツールを備えています。 MPLAB X IDEはMicrochip社ウェブサイトからダウンロードできます。本書で説明するアプリケーションでは MPLAB X IDEバージョン5.40を使います。MPLAB XCコンパイラはMicrochip社ウェブサイトからダウンロードでき ます。本書で説明するアプリケーションではMPLAB XC32バージョン2.41を使います。

### 1.3 MPLAB Harmony v3

MPLAB Harmony v3は、柔軟かつ相互運用性に優れたソフトウェア モジュールを提供する統合型の組み込みソフト ウェア開発フレームワークです。これによりデバイスの細部、複雑なプロトコル、ライブラリの組み込み等に煩わ される事なく、32ビットPIC<sup>®</sup>およびSAMデバイス向けアプリケーションの開発に専念できます。

MPLAB Harmony v3にはMHC (MPLAB Harmony Configurator)が含まれます。このツールは使いやすいGUIを備えて おり、デバイスの設定、ライブラリの選択、アプリケーション コードの開発が容易に行えます。MPLAB Harmony MHCはMPLAB X IDEに統合されるプラグインとして提供されますが、他の開発環境でスタンドアロンとして使うた めのJava実行ファイルも別に備えています。

本書のアプリケーションでは、以下のMPLAB Harmony v3リポジトリを使います。これらのリポジトリはGitHubか ら、またはMHCM (MPLAB Harmony Content Manager)ツールを使ってダウンロードできます。

- ・ CSP(チップサポート パッケージ)
- DEV\_PACKS (Harmony 3製品データベース)
- MHC (Harmony 3 Configurator)
- bootloader(ブートローダ)
- bootloader\_apps\_uart (UARTブートローダ アプリケーション)
- MPLAB Harmony reference\_apps(MPLAB Harmony 3リファレンス アプリケーション) apps\sam\_e54\_ cult\same54\_uart\_live\_updateで最新版を確認できます。

#### 1.4 Python

このアプリケーションでは、バイナリ(UARTフェイルセーフ ブートローダとライブ アップデート アプリケーション)のマージと、マージしたバイナリをホストPCからSAM E54 Curiosity Ultra開発ボードに送信するためにPython v2.7のスクリプトを使います。

# 2. 設計

SAM E54 MCUのライブ アップデート アプリケーションの設計ではデュアルバンク フラッシュメモリ機能を使いま す。これらのバンクはバンクA、バンクBと呼びます。アプリケーションはコード実行中のバンクをアクティブバン ク、もう一方のバンクを非アクティブバンクと認識します。

デュアルバンク フラッシュにより、現在のファームウェアをアクティブバンクから実行中に新しいファームウェア を非アクティブバンクに書き込む事ができます。

ライブ アップデート アプリケーションは以下の2つのタスクに分かれています。

- ・ アプリケーション タスク
- ・ ライブ アップデート タスク

アプリケーション タスク - エンドユーザ アプリケーションを表します。デュアルバンク フラッシュのアクティブバ ンクから現在のエンドユーザ アプリケーションを実行しながら、非アクティブバンクに最新ファームウェアを書き込 みます。本書で説明するアプリケーション タスクは必要最低限のものです。アプリケーション タスクはNVM制御ス テータス レジスタをチェックし、最新ファームウェアが書き込まれているバンクを特定し、バンクAから実行してい る場合は500 msごとに、バンクBから実行している場合は1,000 msごとにLED1またはLED2を点滅させます。例えば、 LED1の点滅で始め、ライブ アップデート要求があると、非アクティブバンクにファームウェアを書き込んだ後で LED2の点滅を始めます。アプリケーション ファームウェアが更新されるたびにこのプロセスが繰り返されます。

ライブ アップデート タスク - あらかじめ定義された通信プロトコル(「UARTライブ アップデート プロトコル」参照)でPC上のホスト アプリケーションと通信し、ホストPCから要求があった時にファームウェアを更新します。要求がない時はアイドル状態で、アプリケーションの機能を実行できます。

ライブ アップデート タスクは以下のサブタスクに分かれています。

- 通信タスク
- コマンド処理タスク
- ・ プログラミング タスク

下図にライブ アップデート アプリケーションの構造を示します。

#### 図2-1. ライブ アップデート アプリケーションの構造



### 2.1 UARTライブ アップデート プロトコル

ライブ アップデート ファームウェアは、あらかじめ定義された通信プロトコルを使ってPC上のホスト アプリケー ションと通信します。このプロトコルの詳細は以下の通りです。

UARTライブ アップデート プロトコルは、下図に示す通りGuardバイト、データサイズ バイト、コマンドバイト、 データバイトで構成されています。

図2-2. UARTライブ アップデート プロトコル

GUARD	Data Size	Command	Data 0	 Data N
(4 Bytes)	(4 Bytes)	(1 Byte)	(4 Bytes)	(4 Bytes)

- Guard
  - Guardは定数値0x5048434Dです。
  - この値は不正なコマンドから保護するためのものです。
  - ライブ アップデート ファームウェアは、パケット受信開始時にGuard値を確認してから後続の処理へ進み ます。
- ・ データサイズ
  - 受信するデータバイトの数を示します。
  - この値はコマンドごとに異なります。
- ・ コマンド
  - 処理するコマンドを示します。各コマンドは1バイトです。
  - サポートしているコマンドは以下の通りです。
    - Unlock (0xA0)
    - Data (0xA1)
    - Verify (0xA2)
- ・データ
  - コマンドで処理するデータを格納しています。
  - 受信データ長はデータサイズ フィールドで示されています。
  - ブートローダはワード(4バイト)単位でデータを受信します。
  - データワードはリトル エンディアン(LSB first)で送信します。

応答コード

ライブ アップデートは各コマンドへの応答として1キャラクタの応答コードを送信します。この応答コードを受信 するか100 msのタイムアウトが経過するまで、ホストは次のコマンドを送信できません。

有効な応答コードは以下の通りです。

- OK (0x50) コマンドを受信して正常に処理した
- エラー(0x51)‐コマンド処理エラー
- 無効(0x52) 無効なコマンドを受信した
- CRC OK (0x53) CRC成功
- CRC Fail (0x54) CRC失敗

次のセクション「コマンドの説明」では、UARTライブ アップデート プロトコルで使われる要求と応答コードの詳 細を説明します。

#### 2.2 コマンドの説明

Unlockコマンド

下図に、Unlockコマンド シーケンスとその応答を示します。

図2-3. UARTライブ アップデートのUnlockコマンド



- UnlockコマンドはDataコマンドより先に発行する必要があります。
  - アプリケーションの開始アドレスと終了アドレスを求めるために使います。
  - この情報で、開始アドレスと終了アドレスがフラッシュメモリの範囲内にあるか検証します。
  - データパケット内のアドレスがUnlockコマンドで呼び出した領域内にあるか検証します。
- 受信するデータは8バイト(開始アドレス+イメージサイズ)です。
- 開始アドレス:
  - アプリケーションの開始アドレスです。
  - デバイスによって異なりますが、ブートローダの終了アドレス以降である事が必要です。
  - 消去単位サイズの境界に揃っている事が必要です。
  - ブートローダを更新するには、この値が「0」に設定する必要があります。
- イメージサイズは消去単位サイズ単位で指定します。この値はデバイスによって異なります。消去単位サイズの詳細はMPLAB Harmony Bootloader Helpの「Bootloader Sizing and Considerations」を参照してください。

Dataコマンド

下図に、Dataコマンド シーケンスとその応答を示します。

#### 図2-4. UARTライブ アップデートのDataコマンド



- Dataコマンドはイメージデータの送信に使います。
- データサイズはブロック開始アドレスの合計(4バイト)です。
- 消去サイズは、消去されるフラッシュのサイズです(デバイスに依存)。
- ブロック開始アドレスはUnlockコマンドでロック解除した領域内である事が必要です。
- ロック解除した領域外に書き込みを要求するとエラーが発生し、データは破棄されます。

#### Verifyコマンド

Verifyコマンド シーケンスと応答を下図に示します。

図2-5. UARTライブ アップデートのVerifyコマンド



- Verifyコマンドは送信されプログラミングされるイメージデータを検証するために使います。
- イメージCRCは0xEDB88320の多項式を持つ標準IEEE<sup>®</sup> CRC32です。
- 内部CRCはプログラミング後にフラッシュメモリから読み出す値を基に求めるものです。従って、チェーン全体を検証するものです。
- イメージCRCはロック解除した領域に対して求めます。

#### 2.3 通信タスク

通信タスクは、選択した通信インターフェイス経由でホストPCまたは組み込みホストから割り込みモードでデータ を受信します。ホストから受信したパケットをヘッダ情報に基づいて検証し、コマンド処理タスクへ渡します。

#### 2.4 コマンド処理タスク

コマンド処理タスクは通信タスクから受け取ったコマンドを処理、実行します。ホストPCへ応答を返し、コマンド がプログラミング コマンドであった場合、制御をプログラミング タスクへ渡します。

### 2.5 プログラミング タスク

プログラミング タスクは受信したデータパケットを内部フラッシュメモリに書き込みます。このタスクはNVMモ ジュール ライブラリを使ってロック解除、消去、書き込み動作を実行し、フラッシュ動作の完了を待機している間 に次のパケットを受信するために通信タスクを呼び出します。

下図に、ファームウェア更新のフローチャートを示します。



#### 図2-6. ファームウェア 更新のフローチャート

### 2.6 メモリレイアウト

ライブ アップデート アプリケーション機能はデュアルバンク フラッシュメモリを備えたMCUで使えます。以下の セクションでは、SAM E54デュアルバンク フラッシュのメモリレイアウト、ブートローダとライブ アップデート アプリケーションの配置と設定について説明します。

SAM E54のフラッシュメモリは2つのバンクで構成されています。バンクAとバンクBです。ブートローダは両バン ク開始位置に配置され、その後にアプリケーション イメージが配置されます。

既定値ではバンクAはアドレス0x00000000、バンクBは0x00080000に配置されます。0x00000000に配置された バンク(既定値ではバンクA)をアクティブバンク、0x00080000に配置されたバンクを非アクティブバンクと呼びま す。

**Note:** 0x0000000に配置されたバンクをアクティブバンクと呼ぶのは、Cortex-M CPUアーキテクチャがアドレス 0x0000000から最初の命令を実行するよう設計されているためです。従って、リセット時に実行する必要がある コードは0x0000000に配置する必要があります。

下図に、SAM E54デュアルバンク フラッシュにおけるライブ アップデート アプリケーションのメモリレイアウト を示します。 図2-7. デュアルバンク メモリ内のライブ アップデート アプリケーション



#### 2.7 実行フロー

初めて実行する場合、MPLAB<sup>®</sup> Harmony v3 UARTフェイルセーフ ブートローダを使ってライブ アップデート アプ リケーションをプログラムします。以下のトピックでは、MPLAB Harmony v3 UARTフェイルセーフ ブートローダ とライブ アップデート アプリケーションのシステムレベル実行フローについて説明します。

#### UARTフェイルセーフ ブートローダのシステムレベル実行フロー

UARTフェイルセーフ ブートローダ コードはデバイスリセット時に実行を開始します。ファームウェア更新モード への移行条件が成立していなければ、UARTフェイルセーフ ブートローダはユーザ アプリケーションの実行を開始 します。ファームウェア 更新モードに移行した場合、UARTフェイルセーフ ブートローダはフラッシュ、消去、プ ログラムを実行します。

#### トリガ手法

- トリガ手法はオンボードスイッチをブートローダトリガピンとして使い、デバイスリセット時にブートローダ へ移行します。
- トリガ手法は、RAMの先頭16バイトにブートローダ要求パターン(0x5048434D)があるかチェックし、デバイ スリセット時にブートローダへ移行します。

下図に、ライブ アップデート アプリケーションを初めてプログラムする場合のMPLAB Harmony v3 UARTフェイル セーフ ブートローダ システムレベル実行フローを示します。 図2-8. ライブ アップデート アプリケーションをプログラムする場合のMPLAB Harmony v3ブートローダ フロー



アクティブバンクから実行中のブートローダが、非アクティブバンクを更新するためのマージイメージ (ブートロー ダ + ライブ アップデート アプリケーション)をアドレス0x00080000から書き込みます。ブートローダは、更新に 成功するとホスト アプリケーションに通知します。ライブ アップデート アプリケーションは、ユーザがスイッチ2 を押してバンクスワップとシステムリセット(BKSWRST)を実行するまで待機します。スイッチ2を押すと以下が実 行されます。

- 非アクティブバンクをアクティブにし、アクティブバンクを非アクティブにします。バンクAは非アクティブになり、バンクBはアクティブになります。
- 更新されたアプリケーションを実行するために、ホストはリセットコマンドを発行します。

どちらのバンクがフラッシュ アドレス0x0000000に配置されているかを示す情報は、フラッシュメモリ内の特別 なヒューズビットに格納されています。これらのヒューズビットは個別に消去/書き込みできます。ホストから BKSWRSTコマンドを受信したブートローダは、フラッシュ(NVM)制御レジスタのBKSWRSTビットをセットします。 BKSWRSTビットがセットされると、フラッシュ(NVM)コントローラはバンクを切り換えし、ヒューズビット (STATUS.AFIRST)を直前のステータスに基づいて以下の通りに設定します。

- STATUS.AFIRST = 0: バンクBの開始アドレスが0x0000000に割り当てられる
- STATUS.AFIRST = 1: バンクAの開始アドレスが0x0000000に割り当てられる

リセット時、フラッシュ(NVM)コントローラはヒューズビット(STATUS.AFIRST)のステータスを確認し、アクティ ブメモリバンクへジャンプしてコードを実行します。

MPLAB Harmony v3 UART フェイルセーフ ブートローダの詳細は <Harmony フォルダ>/ bootloader\_ apps\_uart/apps/uart\_fail\_safe\_bootloader/bootloaderを参照してください。

#### ライブ アップデートのシステムレベル実行フロー

MPLAB Harmony v3 UARTフェイルセーフ ブートローダを通じてライブ アップデート アプリケーションのプログ ラミングに成功すると、ブートローダはライブ アップデート アプリケーションを読み込み、CPUはアプリケーショ ンの実行を開始します。

ライブ アップデート アプリケーションはホストから新しいアプリケーション イメージを受信します。更新が成功 すると、ライブ アップデート アプリケーションはアイドル状態に移行して次のライブ アップデートの要求を待ち ます。スイッチSW2を押すと、更新されたファームウェアが読み込まれます。スイッチを押すと、ライブ アップ デート アプリケーションは非アクティブバンクをアクティブに、アクティブバンクを非アクティブにします。バン クAは非アクティブになり、バンクBはアクティブになります。ライブ アップデート アプリケーションは更新した アプリケーションを実行するためリセットコマンドを発行します。

下図に、ライブ アップデート アプリケーションの実行フローを示します。

図2-9. ライブ アップデート アプリケーションの実行フロー



ライブ アップデート アプリケーションの実行シーケンスは以下の通りです。

 デバイスリセット時、ブートローダはデュアルバンク フラッシュメモリのアクティブバンクからライブ アッ プデート アプリケーションを読み込みます。

- ライブ アップデート アプリケーションはシステムを初期化し、アプリケーション タスクを呼び出した後で ライブ アップデート タスクを呼び出します。
- アプリケーション タスクはライブ アップデート アプリケーションを実行しているアクティブバンクを確認 し、バンクAから実行している場合は500 msごとにLED1を、バンクBから実行している場合は1000 msごと にLED2を点滅させます。
- ライブ アップデート タスクはファームウェア更新が要求されている時のみ実行されます。更新の要求がない 時はアイドル状態で、アプリケーション タスクの実行を継続します。
- 5. アプリケーションがファームウェア更新要求を受信すると、ライブ アップデート タスクは以下を行います。
  - ホストPCからUnlockコマンドを受信すると、パケットの受信を開始します。
  - ホストPCからDataコマンドを受信すると、ファームウェア イメージを受信して非アクティブバンクに プログラムします。
  - ホストPCからVerifyコマンドを受信すると、受信したファームウェア イメージを検証します。その後、 ユーザがスイッチ2を押してバンクをスワップし、デバイスをリセットして更新されたアプリケーション が実行されるのを待ちます。
- ステップ5でエラーが発生した場合、またはホスト アプリケーションがアクティブバンクの更新を要求し、 そのアドレスがアクティブバンクのシリアルセクタにある場合、ライブ アップデート タスクはエラー応答を 送信してプログラミングを中断し、スイッチ2でライブ アップデート再開のためにシステムリセットがトリ ガされるまでLED1とLED2を点灯させます。
- ライブ アップデートが成功すると、LED1が消灯してLED2が点滅します。スイッチ2を押してバンクをス ワップし、システムリセットをトリガして更新したアプリケーションを実行するまでこの状態は維持されま す。

## 3. 設定

ライブ アップデート アプリケーションは以下のアプリケーションで構成されています。

- UARTフェイルセーフ ブートローダ(uart\_fail\_safe\_bootloader\_sam\_e54\_xpro)このアプリケーション はライブ アップデート アプリケーションを初めてプログラムする時に使うブートローダを備えています。
- ライブ アップデート アプリケーション(uart\_same54\_uart\_live\_update)このアプリケーションはユーザ アプリケーション タスクとライブ アップデート機能の実装を備えています。

Note: MPLAB Harmony v3ブートローダ プロジェクトはMPLAB Harmony v3ブートローダUARTアプリケーション リポジトリにあり、以下で提供しています: <Harmonyフォルダ>/bootloader\_apps\_uart/apps/uart\_fail\_safe\_ bootloader/bootloader

ブートローダとUARTフェイルセーフ ブートローダの設定に関する追加情報は以下で提供しています。 <Harmonyフォルダ>/bootloader/doc/help\_bootloader.chmと<Harmonyフォルダ>/bootloader\_apps\_uart/docs/index.html

## 3.1 ブートローダ リンカスクリプト

ブートローダ ライブラリはMHCで生成したカスタムリンカ(btl.ld)スクリプトを使います。MHCは以下を生成します。

- 指定したブートローダサイズ
- コードと読み出し専用データの開始アドレス、ROM(読み出し専用メモリ) セクションのアドレス
- 読み書きデータの開始アドレス、RAM(ランダムアクセス メモリ) セクションのアドレス(下図参照)

リンカスクリプト内の値はMHC設定(ブートローダ設定)のブートローダコンポーネントに基づきます。

ブートローダをRAMから実行してフラッシュメモリへの書き込みと次のデータブロックの受信を同時に実行できる ようにリンカスクリプトを設定します。

外部トリガを使わず、起動時にアプリケーションからブートローダをトリガする場合、アプリケーションはRAMの 先頭から16バイトの領域内にブートローダ要求パターンを保存する必要があります。

SAM E54の場合、-O1最適化オプションを指定した場合のブートローダのサイズは1,672バイトですが、ブートローダ サイズは最も近い消去単位サイズ(8,192バイト)に丸められます。これにより、ブートローダに機能を追加してア プリケーションへのオーバーラップを防ぐ事ができます。

#### 図3-1. ブートローダ カスタム リンカスクリプト



Note: ユーザ アプリケーションとブートローダのメモリ領域が重複しないように注意する必要があります。

#### 3.2 MHCの設定

ライブ アップデート アプリケーションの実装に必要なMHCを設定する手順は以下の通りです。

1. SAM E54 Curiosity Ultra BSPをスイッチとLEDの設定に追加します。

Start Page × Project Graph Configuration Options Help - + 🗙 🗔 🔳 💽 🔜 📟 🛖 💷 View: Root □-SAM E54 Curiosity Ultra BSP Device Family Pack (DFP) CMSIS Pack ⊞-BSP Pin Types SAM E54 Curiosity Ultra BSP System EVSYS Core Peripheral Library Harmony Core Service O RTOS NVMCTRL Core Service O Peripheral Library MEMORY DSU Peripheral Library SERCOM1 Peripheral Library **12C** RTC SPI Peripheral Library TMR UART

図3-2. ライブ アップデート アプリケーションのBSP設定

- デバイスヒューズはブートローダ プロジェクトによって更新されるため、カスタム リンカスクリプトからデ バイスフューズの設定を削除します。
  - 図3-3. ライブ アップデート アプリケーションのヒューズ設定の無効化



 ブートローダ プロジェクトのアプリケーション開始アドレスと一致するライブ アップデート アプリケー ションの開始アドレスを設定します。



図3-4. ライブ アップデート アプリケーションの開始アドレスの設定

 MPLAB Harmony Coreコンポーネントを追加してアプリケーション タスクとライブ アップデート タスクを 設定します。これによりサブタスクを追加するためのtasks.cやユーザ設定を追加するためのuser.h等の ソースファイルが生成されます。





- 5. CRC計算用にDSUモジュールを追加します。
- 6. ホストからのパケットを100 msのターンアラウンド時間で受信するために、SysTickを100 msの周期で設定 します。



図3-6. ライブ アップデート アプリケーションのSysTick設定

7. 異なるレートでLEDを点滅させるためにRTCモジュールを追加します。
 図3-7. ライブ アップデート アプリケーションのRTC設定



 UARTモジュールを追加し、実行時にホストPCからライブ アップデート アプリケーションを受信するために リングバッファを有効にします。

図3-8. ライブ アップデート アプリケーションのUART設定

- 9. [MHC] > [Tools] > [Pin Configurations]からPin Configurationsプラグインを起動します。
- 10. ホストPCからデータを受信するようにUARTピンを設定します。 図3-9. ライブ アップデート アプリケーションのUARTピン設定

Project Graph Pin Diagram x Pin Table x Pin Settings x									
Order: Pins  V Table View  Easy View									
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
111	PC27	SERCOM1_PAD0	SERCOM1_P $\lor$	Digital	High Impedance $\lor$	n/a			Normal $\sim$
112	PC28	SERCOM1_PAD1	SERCOM1_P ~	Digital	High Impedance $\lor$	n/a			Normal $\sim$

Note: SERCOM1はライブ アップデート タスク用に予約済みです。それ以外の目的には使ってはいけません。

### 3.3 プロジェクトの設定

- ・ プリプロセッサ マクロの定義:
  - ROM-ORIGINとROM\_LENGTHはXC32リンカ変数であり、MHCで指定した値で上書きされます。「ライ ブアップデートアプリケーションの開始アドレスの設定」を参照してください。
  - MHCで指定したアプリケーション開始アドレスの値はリンカスクリプトに自動的に格納されます。再生成後に「ブートローダ カスタム リンカスクリプト」を参照してください。
- ビルド後にコマンドラインを実行:
  - 以下のビルドオプションを使うと、ビルドが完了した時に.hexからバイナリファイルを生成できます。

```
${MP_CC_DIR}/xc32-objcopy -I ihex -O binary ${DISTDIR}/${PROJECTNAME}.$
{IMAGE_TYPE}.hex
${DISTDIR}/${PROJECTNAME}.${IMAGE_TYPE}.bin
```

図3-10. ライブ アップデート アプリケーションのバイナリ生
----------------------------------

ategories	
General	Configuration type: application
<ul> <li>File Inclusion/Exclusion</li> <li>Conf: [sam e54 cult]</li> </ul>	Pre and post step operations: Note: commands are run from the project directory (ProjectDir macro below)
<ul> <li>Loading</li> </ul>	Execute this line before build
···· · Libraries ··· · · Building	Macro Value
XC32 (Global Options)	Insert Macro Device ATSAME54P20A
• xc32-gcc	ProjectDir "D:\MassMarket\reference_apps\apps\sam_e54_cult\same54_ua ProjectName uart live update sam e54 cult
	ConfName sam_e54_cult
• xc32-ar	
Code Coverage	Execute this line after build     If the context of the conte
	Jy 1 mex -0 binally storage integration of the s
	Insert Marro Device ATSAME54P20A
	ProjectDir "D:\MassMarket\reference_apps\apps\sam_e54_cult\same54_ua
	ConfName sam_e54_cult
	<pre></pre>
	Options affecting hex file:
	Insert unprotected checksum in user ID memory Normalize hex file
Manage Configurations	
	OK Cancel Apply Unlock Help

# 4. アプリケーションの実行

## 4.1 ブートローダ アプリケーションの実行

MPLAB Harmony v3 UARTフェイルセーフ ブートローダを使ってライブ アップデート アプリケーションをプログ ラムする手順は以下の通りです。

- 1. MPLAB Harmony v3ブートローダ パッケージをダウンロードします。
- 2. MPLAB Harmony v3 UARTブートローダ アプリケーション パッケージをダウンロードします。
- 3. SAM E54 Curiosity Ultra開発ボードのデバッグポートにmicro-USBケーブルを接続します。
- MPLAB X IDEを使ってUARTフェイルセーフ ブートローダを開きます(パス: <Harmonyフォルダ >/bootloader\_apps\_uart/apps/uart\_fail\_safe\_bootloader/bootloader/firmware/sam\_e54\_xpro.X)。
   Note: SAM E54 Curiosity Ultra開発ボードで機能するようにプロジェクトを再設定する必要があります。
- 5. MHCを起動し、以下の手順でSAM E54 Curiosity Ultra開発ボード用にプロジェクトを再設定します。
  - 5.1. SERCOM2を削除します。

図4-1. ブートローダMHC再設定 - SERCOM2モジュールの削除



5.2. SERCOM1をプロジェクト グラフに追加します。



図4-2. ブートローダMHC再設定 - SERCOM1モジュールの追加

- 5.3. [MHC] > [Tools] > [Pin Configurations]からPin Configurationsプラグインを起動します。
- 5.4. SERCOM2のピン設定を削除します。

#### 図4-3. ブートローダMHC再設定 - SERCOM2のピン設定の削除

Project Graph*	Pin Diagra	m 🗴 Pin Table 🗴 Pi	n Settings 🗙						
Order: Ports	Order: Ports 🗸 Table View 🗹 Easy View								
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
100	PB24	SERCOM2_PAD1	SERCOM2_P V	Digital	High Impedance $\lor$	n/a			NORMAL $\sim$
101	PB25	SERCOM2_PAD0	Available	Digital	High Impedance $\lor$	n/a			Normal $\sim$
102	PB26		AC_CMP0	Digital	High Impedance $\lor$	Low			Normal $\sim$
103	PB27		GPIO	Digital	High Impedance $$	Low			Normal $\sim$
104	PB28		PDEC_QDI1	Digital	High Impedance $$	Low			Normal $\sim$
105	PB29		SERCOM0_PAD0 SERCOM2_PAD1	Digital	High Impedance $$	Low			Normal $\sim$
After Pin 🕂 Reconfigurations									
100	PB24		Available 🗸	Digital	High Impedance 🗸	Low			NORMAL V

Available V Digital High Impedance V

5.5. SERCOM1のピンを設定します。

PB25

101

#### 図4-4. ブートローダMHC再設定 - SERCOM1のピンの設定

Project Graph Pin Diagram x Pin Table x Pin Settings x									
Order: Pins View Table View Easy View									
Pin Number	Pin ID	Custom Name	Function	Mode	Direction	Latch	Pull Up	Pull Down	Drive Strength
111	PC27	SERCOM1_PAD0	SERCOM1_P ~	Digital	High Impedance $\lor$	n/a			Normal $\sim$
112	PC28	SERCOM1_PAD1	SERCOM1_P ~	Digital	High Impedance $\lor$	n/a			Normal $\sim$

NORMAL

#### Note:

- SAM E54 Curiosity Ultra開発ボードはEDBGにSERCOM1モジュールを使います。従って、 SERCOMモジュールとそのピンを再設定する必要があります。
- ・ プロジェクトがリビルドされるため、MHCを閉じないでおきます。
- 6. プロジェクトを再生成します。MPLAB X IDEを使ってUARTフェイルセーフ ブートローダをビルドしてプロ グラムします。
- MPLAB X IDEを使ってUARTフェイルセーフ ブートローダをリビルドする手順は以下の通りです。このス テップではブートローダ バイナリイメージを作成します。このイメージをライブ アップデート アプリケー ションとマージして一つのバイナリイメージを作成します(ステップ9参照)。
  - デバイスヒューズはUARTフェイルセーフ ブートローダ プロジェクトによって更新されるため、カスタム リンカスクリプトからデバイスヒューズ設定を削除します。
    - 図4-5. ブートローダMHC再設定 ヒューズ設定の削除



7.2. [Project Properties]ウィンドウで[Building]を選択し、[Execute this line after build]を選択します(下図 参照)。

図4-6. ブートローダMHC再設定 - ビルドオプションの有効化



- 8. MPLAB X IDEを使ってUARTフェイルセーフ ブートローダ アプリケーションを再ビルドします。ただし、プ ログラムしないでおきます。
- 9. ライブ アップデート アプリケーションをダウンロードし、MPLAB X IDEを使ってビルドします(パス: <ライ ブ アップデート アプリケーション フォルダ>/firmware/sam\_e54\_cult.X)。ただし、プログラムしないでおき ます。

10. コマンド プロンプトからbtl\_app\_merge\_bin.pyスクリプトを実行し、生成したブートローダ バイナリと ライブ アップデート アプリケーション バイナリをマージします。

python <Harmony folder>/bootloader/tools\_archive/btl\_app\_merge\_bin.py -0 0x2000 -b
<Harmony
folder>/bootloader\_apps\_uart/apps/uart\_fail\_safe\_bootloader/bootloader/firmware/
sam\_e54\_xpro.X/dist/<Config\_Name>/production/<Project\_Name>.production.bin
-a <Live Update application
folder>/firmware/sam\_e54\_cult.X/dist/<Config\_Name>/production/<Project
Name>.production.bin

10.1. コマンド プロンプトに以下が表示されます。 図4-7. ブートローダ バイナリのマージ結果

'\r\n##### Merged Bootloader and Application binaries to ', 'btl\_app\_merged.bin', '#####')

 コマンド プロンプトからbtl\_host.pyを実行し、マージしたバイナリをデュアルバンク フラッシュの非ア クティブバンクへ書き込みます。マージしたバイナリbtl\_app\_merged.binがbtl\_app\_merge\_bin.pyを 呼び出したパス内に生成されます。

python <Harmony folder>/bootloader/tools\_archive/btl\_host.py -v -s -i <COM PORT> -d
same5x -a <Address> -f btl\_app\_merged.bin

Note:

- ブートローダのHostスクリプトの詳細は、以下のヘルプのHostスクリプト設定方法を参照してください。
   <Harmonyフォルダ>/bootloader/doc/help\_bootloader.chmと<Harmonyフォルダ>/bootloader\_apps\_uart/ docs/index.html
- 12. 下図に、成功したアプリケーション バイナリのプログラミングを示します。
  - 図4-8. アプリケーション バイナリ プログラミングのブートローダHostスクリプトログ

Unlocking					
Uploading 2 bl	locks at	address	524288	(0x80000)	
block 1 of	F 2				
block 2 of	F 2				
Verification					
success					
Swapping Bank	And Rebo	oting			
Reboot Done					

# 4.2 ライブ アップデート アプリケーションの実行

- 1. 「ブートローダ アプリケーションの実行」の手順を実行し、ライブ アップデート アプリケーションをダウン ロードします。
- 2. 上記のステップが成功した場合、SAM E54 Curiosity Ultra開発ボードのLED1が点滅を開始するはずです。
- <ライブ アップデート アプリケーション フォルダ>/scriptsにあるHostスクリプトlive\_update.pyをコマン ド プロンプトから実行し、現行バージョンのアプリケーションをアクティブパネルから実行している間に新 しいバージョンのファームウェアを非アクティブパネルにプログラムします。

python <Live Update application folder>/scripts/live\_update.py -v -i <COM PORT> -d same5x -a <Address> -f btl\_app\_merged.bin.

- 3.1. マージしたバイナリbtl\_app\_merged.binがbtl\_app\_merge\_bin.pyを呼び出したパス内に生成されます。
- 3.2. 下図にライブ アップデート スクリプトのヘルプを示します。

図4-9. ライブ アップデート アプリケーションHostスクリプトのヘルプウィンドウ

Usage: live_update.py [	options]
Options:	
-h,help	show this help message and exit
-v,verbose	enable verbose output
-r BAUD,baud=BAUD	UART baudrate
-t,tune	auto-tune UART baudrate
<ul> <li>-i PATH,interface=</li> </ul>	РАТН
	communication interface
-f FILE,file=FILE	binary file to program
-a ADDR,address=AD	DR
	destination address
<ul> <li>-p SectSize,sector</li> </ul>	Size=SectSize
	Device Sector Size in Bytes
-b,boot	enable write to the bootloader area
-d DEV,device=DEV	<pre>target device (samc2x/samd1x/samd2x/samd5x/samda1/same</pre>
	7x/same5x/samg5x/saml2x/samha1/pic32mk/pic32mx/pic32mz

Command: python <python script> -v -s -i <COM PORT> -d <Device Name> -a <Address> -f <live\_update\_application\_image> <python script>: live\_update.py <COM PORT>: Serial Communication Port <Device Name>: same5x <Address>: Application start address 0x9D100000 Example: python < Live update application folder \scripts\ live\_update.py -v -s i COM6 -d same5x -a 0x80000 -f sam\_e54\_cult.X.production.bin Note: Running the help command provides a brief overview of options available as shown below. Command: python <cript> --help <python script>: live\_update.py

**Note:** ブートローダのHostスクリプトの詳細は、以下のヘルプのHostスクリプト設定方法を参照してください。<Harmonyフォルダ>/bootloader/doc/help bootloader.chm

7. 下図に、ライブ アップデートのファームウェア更新時の出力を示します。
 図4-10. ライブ アップデート アプリケーションのファームウェア更新時のコンソール出力

```
Live Update request started.

Unlocking

Uploading 2 blocks at address 524288 (0x80000)

... block 1 of 2

... block 2 of 2

Verification

... success

Live Update is success.

Press Switch (SW2) to reset the device for the programmed application firmware to run.
```

- LED1が点滅を停止し、LED2が点滅を始めます。これはアプリケーションのプログラミングが成功した事を 示します。
  - 5.1. LED1が点滅を停止し、LED2が500 msごとに点滅していればアプリケーションはバンクAから実行 されています。
  - 5.2. LED1が点滅を停止し、LED2が1,000 msごとに点滅していればアプリケーションはバンクBから実 行されています。
- スイッチSW2を押してバンクを切り換え、プログラムしたアプリケーション ファームウェアを実行するよう にデバイスをリセットします。
  - 6.1. LED2が点滅を停止し、LED1が500 msごとに点滅していればアプリケーションはバンクAから実行 されています。

- 6.2. LED2が点滅を停止し、LED1が1000 msごとに点滅していればアプリケーションはバンクBから実行 されています。
- 7. ライブ アップデート中にエラーが発生するとLED1とLED2が点灯します。以下の手順でエラーから復帰できます。
  - 7.1. スイッチSW2を長押ししてシステムをリセットした後、ステップ4でライブ アップデート アプリケーションをプログラムします。 または
  - 7.2. デバイスをリセットするか電源を一度切断してステップ4から再実行してライブ アップデート アプ リケーションをプログラムします。デバイスをリセットするか電源を一度切断してステップ4から再 実行してライブ アップデート アプリケーションをプログラムします。

### 5. まとめ

本書ではライブ アップデート アプリケーションの実装を説明しました。ライブ アップデート アプリケーションを 使うと、既存のアプリケーション ソフトウェアを実行しながらソフトウェアを更新し、簡単に新しい機能を導入で きます。これによりデバイスをより長く使い、製品の保守コストを低減できます。定期的な更新もニーズに応じた 更新も可能で、製品の展開が容易になります。

本書は、ユーザ アプリケーションにライブ アップデート機能を実装するための参考資料として利用できます。

ライブ アップデート タスクは実行時にファームウェアを更新するために割り込みモードで実行されます。従って、 ファームウェア更新要求が存在する時にアプリケーション タスクが連続的にポーリングを行わないようにする必要 があります。そうしないと、ホストPCからファームウェアを受信中にデータ損失またはタイムアウトが発生します。

# 6. 参考資料

- MPLAB<sup>®</sup> Harmony v3: www.microchip.com/mplab/mplab-harmony
- ・ SAM E54 Curiosity Ultra開発ボード向けのライブアップデート アプリケーション: microchiptech.github.io/MPLAB-Harmony-Reference-Apps/apps/sam\_e54\_cult/same54\_uart\_live\_update/ readme.html
- ・ SAM E54 Curiosity Ultra開発ボード + maXTouch<sup>®</sup> Curiosity Proボード向けSDカードUSBオーディオ プレーヤ (Legato Graphicsグラフィック使用): microchiptech.github.io/MPLAB-Harmony-Reference-Apps/apps/sam\_e54\_cult/ same54\_sdcard\_usb\_audio\_player/readme.html
- MPLAB Harmony v3ブートローダのヘルプ:
   <Harmonyフレームワーク ダウンロード フォルダ>\bootloader\doc\help\_bootloader.chm
- ・ MPLAB Harmony v3 Coreのヘルプ: microchip-mplab-harmony.github.io/core/frames.html?frmname=topic&frmfile=index.html
- Getting Started with MPLAB Harmony v3 Peripheral Libraries on SAM D5x/E5x MCUs: microchipdeveloper.com/harmony3:same54-getting-started-training-module
- MPLAB Harmony v3を使ったSAM E54 MCUのデュアルバンク ブートローダの構築: ww1.microchip.com/downloads/en/Appnotes/Dual-Bank-Bootloader-on-SAME54-Microcontroller-Using-MPLAB-Harmonyv3-DS00003508A.pdf
- MPLAB<sup>®</sup> Harmony v3リファレンス アプリケーション: github.com/MicrochipTech/MPLAB-Harmony-Reference-Apps
- MPLAB<sup>®</sup> Harmony v3の使い方に関する記事とその他の文書: www.microchip.com/mplab/mplab-harmony/mplab-harmony-articles-and-documentation
- MPLAB® Harmony v3 Configurator Overview: microchipdeveloper.com/harmony3:mhc-overview
- Create your first Motor Control Application using MPLAB<sup>®</sup> Harmony v3: microchipdeveloper.com/harmony3:motor-control-getting-started-training-module
- How to Build an Application by Adding a New PLIB, Driver, or Middleware to an Existing MPLAB Harmony v3 Project:

ww1.microchip.com/downloads/en/DeviceDoc/How\_to\_Build\_Application\_Adding\_PLIB\_%20Driver\_or\_Middlew are%20\_to\_MPLAB\_Harmony\_v3Project\_DS90003253A.pdf

- MPLAB® Harmony v3 SD Card Audio Player/Reader Tutorial: microchipdeveloper.com/harmony3:audio-player
- Graphics Quick Start Applications for PIC32MZ and SAM MCUs: github.com/Microchip-MPLAB-Harmony/gfx/wiki/Application-QuickStart
- MPLAB Harmony v3 USB Stack: microchip-mplab-harmony.github.io/usb/frames.html?frmname=topic&frmfile=index.html
- Create Your First USB Device CDC Single Application: github.com/Microchip-MPLAB-Harmony/usb/wiki/Create-your-first-usb-device-cdc-single-application
- Create Your First USB Host MSD Application: github.com/Microchip-MPLAB-Harmony/usb/wiki/Create-your-first-usb-host-msd-application
- MPLAB Harmony v3 TCP/IP Help: microchip-mplab-harmony.github.io/net/frames.html?frmname=topic&frmfile=index.html
- Create Your First TCP/IP Application: github.com/Microchip-MPLAB-Harmony/net/wiki/Create-your-first-tcpip-application
- MPLAB Harmony v3 Application Development Guide for MPLAB Harmony v2 Users: ww1.microchip.com/downloads/en/Appnotes/MPLAB\_Harmonyv3\_Application\_Development\_%20Guide\_for\_% 20MPLAB\_Harmonyv2\_Users\_DS00003388A.pdf

# Microchip社ウェブサイト

Microchip社はウェブサイト(www.microchip.com)を通してオンライン サポートを提供しています。当ウェブサイト では、お客様に役立つ情報やファイルを簡単に見つけ出せます。以下を含む各種の情報がご覧になれます。

- 製品サポート データシートとエラッタ、アプリケーション ノートとサンプル プログラム、設計リソース、 ユーザガイドとハードウェア サポート文書、最新のソフトウェアと過去のソフトウェア
- 技術サポート FAQ(よく寄せられる質問)、技術サポートのご依頼、オンライン ディスカッション グループ、 Microchip社のデザイン パートナー プログラムおよびメンバーリスト
- **ご注文とお問い合わせ** 製品セレクタと注文ガイド、最新プレスリリース、セミナー/イベントの一覧、お問い 合わせ先(営業所/正規代理店)の一覧

# 製品変更通知サービス

Microchip社の製品変更通知サービスは、お客様にMicrochip社製品の最新情報をお届けする配信サービスです。ご興味のある製品ファミリまたは開発ツールに関する変更、更新、リビジョン、エラッタ情報をいち早くメールにてお知らせします。

http://www.microchip.com/pcnにアクセスし、登録手続きをしてください。

# お客様サポート

Microchip社製品をお使いのお客様は、以下のチャンネルからサポートをご利用になれます。

- 正規代理店
- 技術サポート

サポートは正規代理店にお問い合わせください。各地の営業所もご利用になれます。本書の最後のページに各国の 営業所の一覧を記載しています。

技術サポートは以下のウェブページからもご利用頂けます。www.microchip.com/support

# Microchip社のデバイスコード保護機能

Microchip 社製品のコード保護機能について以下の点にご注意ください。

- Microchip社製品は、該当するMicrochip 社データシートに記載の仕様を満たしています。
- Microchip社では、通常の条件ならびに動作仕様書の仕様に従って使った場合、Microchip 社製品のセキュリティレベルは、現在市場に流通している同種製品の中でも最も高度であると考えています。
- Microchip社はその知的財産権を重視し、積極的に保護しています。Microchip 社製品のコード保護機能の侵害は 固く禁じられており、デジタル ミレニアム著作権法に違反します。
- Microchip社を含む全ての半導体メーカーで、自社のコードのセキュリティを完全に保証できる企業はありません。 コード保護機能とは、Microchip 社が製品を「解読不能」として保証するものではありません。コード保護機能は 常に進化しています。Microchip 社では、常に製品のコード保護機能の改善に取り組んでいます。

# 法律上の注意点

本書および本書に記載されている情報は、Microchip 社製品を設計、テスト、お客様のアプリケーションと統合する 目的を含め、Microchip 社製品に対してのみ使う事ができます。それ以外の方法でこの情報を使う事はこれらの条項 に違反します。デバイス アプリケーションの情報は、ユーザの便宜のためにのみ提供されるものであり、更新に よって変更となる事があります。お客様のアプリケーションが仕様を満たす事を保証する責任は、お客様にあります。 その他のサポートはMicrochip 社正規代理店にお問い合わせ頂くか、https://www.microchip.com/en-us/support/ design-help/client-supportservicesをご覧ください。

Microchip 社は本書の情報を「現状のまま」で提供しています。Microchip 社は明示的、暗黙的、書面、口頭、法定の いずれであるかを問わず、本書に記載されている情報に関して、非侵害性、商品性、特定目的への適合性の暗黙的保 証、または状態、品質、性能に関する保証をはじめとするいかなる類の表明も保証も行いません。

いかなる場合もMicrochip 社は、本情報またはその使用に関連する間接的、特殊的、懲罰的、偶発的または必然的損 失、損害、費用、経費のいかんにかかわらず、またMicrochip 社がそのような損害が生じる可能性について報告を受 けていた場合あるいは損害が予測可能であった場合でも、一切の責任を負いません。法律で認められる最大限の範囲 を適用しようとも、本情報またはその使用に関連する一切の申し立てに対するMicrochip 社の責任限度額は、使用者 が当該情報に関連してMicrochip 社に直接支払った額を超えません。

Microchip 社の明示的な書面による承認なしに、生命維持装置あるいは生命安全用途にMicrochip社の製品を使う事は 全て購入者のリスクとし、また購入者はこれによって発生したあらゆる損害、クレーム、訴訟、費用に関して、 Microchip 社は擁護され、免責され、損害をうけない事に同意するものとします。特に明記しない場合、暗黙的ある いは明示的を問わず、Microchip社が知的財産権を保有しているライセンスは一切譲渡されません。

# 商標

Microchip 社の名称とロゴ、Microchip ロゴ、Adaptec、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、 CryptoMemory、CryptoRF、dsPIC、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、 maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi ロゴ、MOST、MOST ロゴ、MPLAB、 OptoLyzer、PIC、picoPower、PICSTART、PIC32 ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、 SenGenuity、SpyNIC、SST、SST ロゴ、SuperFlash、Symmetricom、SyncServer、Tachyon、TimeSource、tinyAVR、 UNI/O、Vectron、XMEGA は米国とその他の国におけるMicrochip TechnologyIncorporated の登録商標です。

AgileSwitch、APT、ClockWorks、The Embedded Control SolutionsCompany、EtherSynch、Flashtec、Hyper Speed Control、HyperLightLoad、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、 ProASIC Plus、ProASIC Plus ロゴ、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、 TimePictra、TimeProvider、TrueTime、ZL は米国におけるMicrochip Technology Incorporated の登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、Augmented Switching、BlueSky、BodyCom、Clockstudio、CodeGuard、CryptoAuthentication、CryptoAutomotive、 CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、 Espresso T1S、EtherGREEN、GridTime、IdealBridge、In-Circuit Serial Programming、ICSP、INICnet、Intelligent Paralleling、IntelliMOS、Inter-Chip Connectivity、JitterBlocker、Knob-on-Display、KoD、maxCrypto、maxView、 memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified ロゴ、MPLIB、MPLINK、MultiTRAK、NetDetach、 Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、 REAL ICE、RippleBlocker、RTAX、RTG4、SAM-ICE、Serial Quad I/O、simpleMAP、SimpliPHY、SmartBuffer、 SmartHLS、SMART-I.S.、storClad、SQI、SuperSwitcher、SuperSwitcher II、Switchtec、SynchroPHY、 TotalEndurance、Trusted Time、TSHARC、USBCheck、VariSense、VectorBlox、VeriPHY、ViewSpan、 WiperLock、XpressConnect、ZENAは米国とその他の国におけるMicrochip Technology Incorporated の商標です。

SQTP は米国におけるMicrochip Technology Incorporated のサービスマークです。

Adaptec ロゴ、Frequency on Demand、Silicon Storage Technology、Symmcom はその他の国におけるMicrochip Technology Incorporatedの登録商標です。

GestIC は、その他の国におけるMicrochip Technology Germany II GmbH & Co. KG (Microchip Technology Incorporated の子会社)の登録商標です。

その他の商標は各社に帰属します。 © 2022, Microchip Technology Incorporated and its subsidiaries. All Rights Reserved. ISBN: 978-1-5224-9116-3

# 品質管理システム

Microchip社の品質管理システムについてはwww.microchip.com/qualityをご覧ください。



# 各国の営業所とサービス

#### 南北アメリカ

#### 本社

2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術サポート: http://www.microchip.com/support URL:

#### www.microchip.com

**アトランタ** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455

**オースティン、TX** Tel: 512-257-3370

**ポストン** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

**シカゴ** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

ダラス Addison, TX

Tel: 972-818-7423 Fax: 972-818-2924

**デトロイト** Novi, MI Tel: 248-848-4000

**ヒューストン、TX** Tel: 281-894-5983

インディアナポリス Noblesville, IN

Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380

#### ロサンゼルス

Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800

**ローリー、NC** Tel: 919-844-7510

ニューヨーク、NY Tel: 631-435-6000 サンノゼ、CA

Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980 Fax: 905-695-2078

# アジア/太平洋

オーストラリア - シドニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10 -8569-7000

**中国 - 成都** Tel: 86-28-8665-5511 **中国 - 重慶** 

Tel: 86-23-8980-9588 **中国 - 東莞** Tel: 86-769-8702-9880

**中国 - 広州** Tel: 86-20-8755-8029

**中国 - 杭州** Tel: 86-571-8792-8115 **中国 - 香港SAR** 

Tel: 852-2943-5100 **中国 - 南京** 

Tel: 86-25-8473-2460 中国 - 音島

中国 - **肖** - **日** Tel: 86-532-8502-7355 中国 - 上海

Tel: 86-21-3326-8000 **中国 - 瀋陽** Tel: 86-24-2334-2829

**中国 - 深圳** Tel: 86-755-8864-2200

**中国 - 蘇州** Tel: 86-186-6233-1526

**中国 - 武漢** Tel: 86-27-5980-5300 **中国 - 西安** 

Tel: 86-29-8833-7252 中国 - 厦門

Tel: 86-592-2388138 **中国 - 珠海** Tel: 86-756-3210040

### アジア/太平洋

インド - バンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - プネ

Tel: 91-20-4121-0141 **日本 - 大阪** Tel: 81-6-6152-7160

**日本 - 東京** Tel: 81-3-6880-3770

**韓国 - 大邱** Tel: 82-53-744-4301

韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ

Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870

**台湾 - 新竹** Tel: 886-3-577-8366 **台湾 - 高雄** 

Tel: 886-7-213-7830 **台湾 - 台北** 

Tel: 886-2-2508-8600 タイ - パンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100 欧州 オーストリア - ヴェルス Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4485-5910 Fax: 45-4485-2829

フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

**ドイツ - ガーヒンク** Tel: 49-8931-9700 **ドイツ - ハーン** Tel: 49-2129-3766400

**ドイツ - ハイルブロン** Tel: 49-7131-72400

**ドイツ - カールスルーエ** Tel: 49-721-625370 **ドイツ - ミュンヘン** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44

**ドイツ - ローゼンハイム** Tel: 49-8031-354-560

**イスラエル - ラーナナ** Tel: 972-9-744-7705

イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781

**イタリア - パドヴァ** Tel: 39-049-7625286

**オランダ - ドリューネン** Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-7288-4388

**ポーランド - ワルシャワ** Tel: 48-22-3325737 **ルーマニア - ブカレスト** Tel: 40-21-407-87-50

スペイン - マドリッド Tel: 34-91-708-08-90 Fax: 34-91-708-08-91

スウェーデン - ヨーテボリ Tel: 46-31-704-60-40 スウェーデン - ストックホルム

Tel: 46-8-5090-4654 イギリス - ウォーキンガム Tel: 44-118-921-5800 Fax: 44-118-921-5820