

FM0+ファームウェア開発入門

著者: James Trudeau

関連部品 : すべての FM0+部品

関連アプリケーションノート&コード例 : FM0+ポートフォリオ リソースを参照してください

AN210985 は、Arm® Cortex®-M0+プロセッサコアをベースにした 32 ビット汎用マイクロコントローラ (以下 MCU) の FM0+ ポートフォリオを紹介し、超低電力設計に最適です。本文書は、ハードウェアの特長と機能、ファームウェアの開発、およびユーザが利用可能なテクニカル リソースをご提供します。

目次

1 FM0+ MCU の概要.....1	3.2 S6E1C3 キットでの FLASH USB DIRECT Programmer の使用.....7
2 ファームウェア開発.....3	3.3 FLASH MCU Programmer の使用.....9
2.1 ペリフェラルドライブライブラリ v2.x の概要...3	4 FM0+ポートフォリオ リソース.....10
2.2 ソフトウェア開発の概要.....4	改訂履歴..... Error! Bookmark not defined.
3 内蔵フラッシュメモリへの書き込み.....6	
3.1 始める前に.....7	

1 FM0+ MCU の概要

サイプレスの FM0+は、CM0+プロセッサをベースにした、32 ビットの汎用でエネルギー効率の高い MCU のポートフォリオです。FM0+ MCU は 40MHz で動作し、家電製品、センサー、メーター、HMI システム、電動工具、ネットワーク機器 (モノのインターネット) 電池式またはウェアラブル デバイス等に最適なさまざまなチップ搭載のペリフェラルに対応します。

FM0+ポートフォリオには 2 つのシリーズがあります。各シリーズは異なる機能を備えた複数のデバイス パッケージによって構成されます。表 1 は、各シリーズの代表的な機能における最大値を示します。

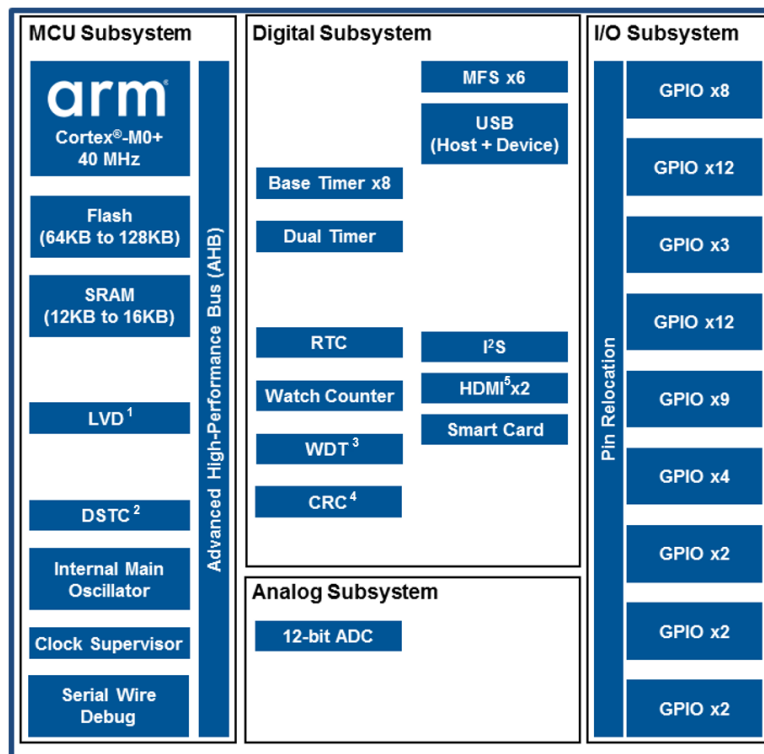
表 1. FM0+の概要

シリーズ	S16E1C	S6E1A
フラッシュ/SRAM (KB)	128/16	88/6
GPIO	54	37
ベース タイマ	8	4
マルチファンクション タイマ	-	1
直交カウンタ	-	1
マルチファンクションシリアル	6	3
USB	1	-
HDMI-CEC	2	-
DMA/DSTC	-/64	2/-
ADC 入力	8	8
CRC	Y	-
I2S	2	-
スマートカードインターフェース	1	-

FM0+ MCU の主な機能は次のとおりです。

- 性能と省エネルギー  
MCUは、現在利用可能な最もエネルギー効率の高い Arm プロセッサである CM0+コアに基づいています。FM0+ MCU の最適化された処理およびフラッシュアーキテクチャにより、業界で最もエネルギー効率の高い CM0+ MCU となり、業界をリードする 35  $\mu$ A/CoreMark<sup>®</sup>スコアを達成しました。
- 超低消費電力  
超低電力デバイスでは、動作電圧範囲は 1.65~3.6 V、CPU クロックの最大周波数は 40 MHz、アクティブモード電流は 40 $\mu$ A/MHz、そして RTC スタンバイモード電流は 0.6 $\mu$ A です。
- 高性能フラッシュメモリ  
メモリ容量は、フラッシュが 56 KB~512 KB、RAM が最大 64 KB です。通常この容量は、より大きな Cortex-M3/M4 コアを備えた MCU でのみ見られます。フラッシュメモリは、CPU の最大速度についてのゼロ ウェイト動作に対応しており、最大 20 年のデータ保持期間を実現します。
- 他の主な特長  
これらのデバイスは、アナログ周辺機器も備えています。簡素化されたバス マトリックスは消費電力を減少させます。これらのデバイスには、各ペリフェラル用のローカルクロックゲーティングも含まれ、CPU とペリフェラル用に独立したクロック分周器により、消費電力の微調整を可能にします。FM0+ MCU は、フルスピード USB2.0 ホストおよびデバイス機能を備え、複数のシリアル通信インターフェースと AES 暗号化を提供します。  
図 1 に例として S6E1C シリーズのブロック図を示します。FM0+ MCU は、A/D コンバーター、USB、構成可能な多機能シリアルインターフェース、リアルタイムクロックなどのさまざまな周辺機器を提供します。周辺機器のサポートはデバイスによって異なります。ピン数、パッケージオプション、動作電圧範囲、使用可能な周辺機器、フラッシュ/SRAM オプションなどの各シリーズの詳細については、[製品選択ガイド](#)とデータシートを確認してください。ハードウェア設計の考慮事項の詳細については、AN203277 をご覧ください。

図 1. FM0+ S6E1C シリーズのブロック図



<sup>1</sup> 低電圧検出

<sup>2</sup> 記述子システム転送コントローラー

<sup>3</sup> ウォッチドッグタイマー

<sup>4</sup> 巡回冗長検査

<sup>5</sup> HDMI 家電制御信号

## 2 ファームウェア開発

このセクションでは、サイプレス FM ペリフェラルドライブライブラリ (PDL) v2.x について説明します。PDL はすべての FM 製品のファームウェア開発の中心です。PDL により、広範囲なペリフェラルのソフトウェア開発が容易に行えます。レジスタおよびビットの構造を理解する必要性が軽減されます。このライブラリを要求機能仕様に合わせて設定することで、ペリフェラルの初期化と制御を API 関数を使用して行えます。FM0+プロセッサに加えて、PDL はサイプレス FM4 および FM3 プロセッサと周辺機器をサポートします。PDL を使用すると、ある製品から他の製品へコードを移植することがより簡単になります。

レジスタ レベルでの開発者であっても、PDL をインストールすることをお勧めします。PDL から、あらゆる FM0+ デバイスのデバイス固有のヘッダ ファイル、スタートアップ コード、コンフィギュレーション ファイル、および IDE プロジェクト ファイルを入手できます。これらのファイルを PDL と共に使用することも、PDL から独立して使用することも可能です。

PDL はソース コードとして用意されます。PDL ソース コードは、低レベルで MCU をプログラムするために必要な詳細な知識を得るのに役立ちます。また、適切なデータシートとペリフェラル マニュアルを併せて参照することで、ペリフェラルを使用するために必要な情報を取得できます。利用可能な広範な技術文書へのリンクについては、この文書の「[FLASH MCU Programmer の使用](#)」を参照してください。

PDL はすべての FM デバイスの中心であるため、ファームウェア開発に関する詳細な情報は PDL クイックスタートガイドに記載されています。これには、PDL サンプル コードをビルドおよび実行する手順が含まれます。PDL クイック スタート ガイドは PDL と共にインストールされます。[サイプレス PDL ソフトウェアアーカイブ](#)からも個別に入手できます。

### 2.1 ペリフェラルドライブライブラリ v2.x の概要

PDL はサポートされているデバイスのドライバを構築するためのスーパーセットのコードです。スーパーセット デザインとは：

- ペリフェラルを初期化、設定および使用するために必要なすべての API が提供されます。
- PDL は対象となるペリフェラルが選択されたデバイスに存在することを保証するためのエラーチェックを含みません

このスーパーセット デザインは、利用可能なペリフェラルにかかわらず、PDL がすべてのデバイスに役立つことを意味します。これにより、ペリフェラルが存在するプラットフォームにわたってコードの互換性が維持されます。指定したハードウェアでサポートされないペリフェラルを含むように PDL を設定した場合、プロジェクトは実行時ではなく、コンパイル時に失敗します。PDL コンフィギュレーション ロジックはターゲット プロセッサを認識し、サポートされないペリフェラルのペリフェラル レジスタ ヘッダをビルドから除去します。

周辺機器を使用するコードを書く前に、シリーズまたはデバイスのデータシートを参照して、周辺機器のサポートを確認してください。

#### 2.1.1 PDL v2.x の入手とインストール

[サイプレス PDL ソフトウェアアーカイブ](#)から PDL インストーラをダウンロードします。インストーラを起動し、表示に従って設定してください。

#### 2.1.2 PDL の構造

PDL v2.x は、[表 2](#) に示すようにいくつかのフォルダに編成されます。

表 2. PDL フォルダの構造

パス\フォルダ	変更内容
cmsis	cmsis ヘッダ ファイル
devices	各デバイス パッケージ: 共通ヘッダ ファイル 各 IDE のコンフィギュレーション、スタートアップ、およびプロジェクト ファイル
doc	PDL ドキュメント
driver	ドライバのソース コードおよびヘッダ ファイル
examples	サポートされた各スターターキット上の各ペリフェラルのサンプル コード
utilities	さまざまなユーティリティ ファイル

PDL を使用する場合、通常、変更するファイルは `pdl_user.h` と `main.c` だけです。

## 2.2 ソフトウェア開発の概要

表 3. サポートされているツールチェーン

ベンダー	ツール	バージョン
IAR Systems	<a href="#">Embedded Workbench</a>	7.50.1 以降
ARM Keil	<a href="#">µVision</a>	5.17 以降
オープン ソース/ARM	<a href="#">GCC ARM Embedded</a>	4.9-d015-Q1-Update 以降
iSYSTEM	<a href="#">winIDEA</a>	9.12 以降

### 2.2.1 PDL サンプル コードの使用

PDL v2.1 には、特定のスターターキット用に構成されたコード例が含まれます。サンプル コードは、examples フォルダにスターターキット別に整理されています。各サンプル コードは特定のペリフェラルの基本的な初期化および設定を実演します。複数のサンプル コードがあるペリフェラルもあります。

**Note:** PDL プロジェクトをビルドおよび実行する手順については、PDL と共にインストールされた PDL クイック スタート ガイドを参照してください。[サイプレス PDL ソフトウェアアーカイブ](#)からも個別に入手できます。

一部のサイプレス FM0+スターターキットは、キットの一部として、あるバージョンの PDL をインストールします。それらのキットは PDL の旧バージョンをインストールする場合があります。スターターキットのサンプル コードは、キットが使用する PDL のバージョンにのみ対応しており、新しいバージョンを含む他のバージョンには対応していません。その他の PDL のバージョンには対応していません。

### 2.2.2 PDL を使用したコードを書く

カスタム プロジェクトの作成、PDL の設定、ペリフェラルの設定、ペリフェラルの使用などトピックの詳細情報については、PDL ディレクトリ内の doc フォルダにある PDL クイック スタート ガイドを参照してください。これは[サイプレス PDL ソフトウェアアーカイブ](#)からも個別に入手できます。

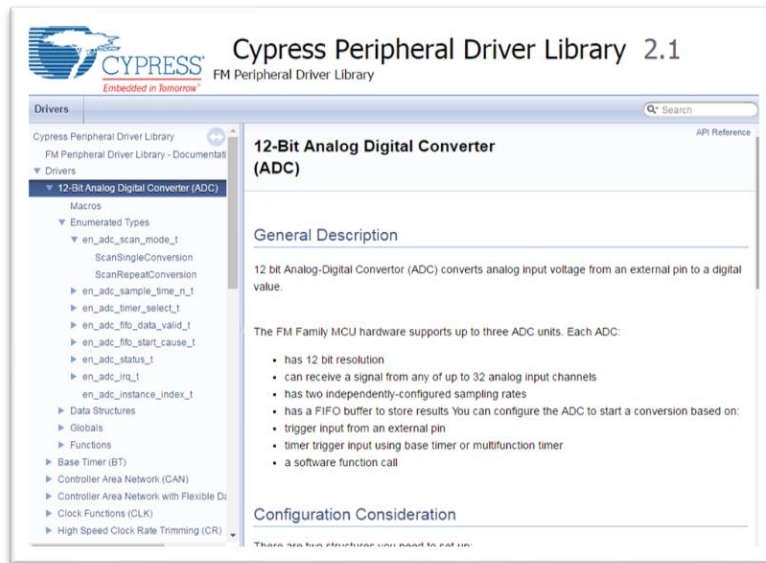
### 2.2.3 PDL API ドキュメント

PDL API ドキュメントは HTML ベースであり、ソース コードから生成されます。PDL インストーラは、ドキュメントをここに配置します: `<PDL ディレクトリ>\doc\pdl_api_reference_manual.html`。

ドキュメントを初めて開く時、簡単にアクセスできるようにブラウザでブックマークを作成してください。

このドキュメントでは、左側のナビゲーション メニューを使用して必要な情報を検索します。**Drivers** セクションには、特定の周辺機器のすべての情報が一覧表示されます。任意のドライバを展開して、マクロ、型、構造体、グローバル変数や API 関数を表示します。[図 2](#) は、ドキュメントのホームページを示します。

図 2. PDL ドキュメント

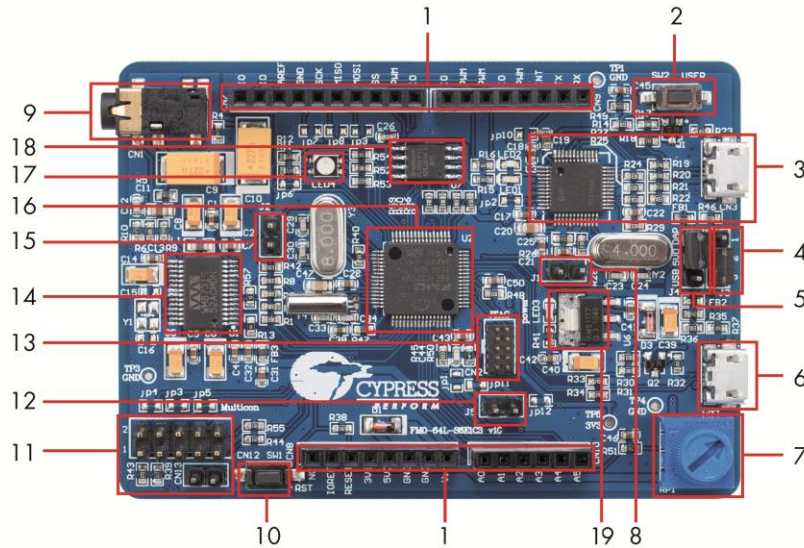


役立つ情報への追加リンクについては、このドキュメントの [FM0+ポータルリソース](#)のセクションを参照してください。

### 3 内蔵フラッシュメモリへの書き込み

FM MCU のファームウェア開発の詳細については、*PDL クイックスタートガイド*を参照してください。ほとんどの IDE は内蔵フラッシュへのプログラムが可能です。単体のフラッシュプログラマが好まれる、もしくはその方法しかない場合もあります。このセクションでは、S6E1C デバイスの組込みフラッシュをプログラムする方法を示します。これらの手順は、**FM0+ S6E1C3 MCU スターターキット**に含まれるボードを対象とします。図 3 にハードウェアボード上の主要コンポーネントを示します。

図 3. S6E1C3 MCU スターターキットボード



- |  |  |
|--|--|
| 1. Arduino interface (CN7-CN10)              | 10. Reset button                         |
| 2. User button                               | 11. Multicon connector (CN12, CN13)      |
| 3. Programmer and debugger (CMSIS-DAP)       | 12. Jumper for current measure (J5)      |
| 4. Series programming mode select (J3)       | 13. 10-pin JTAG connector                |
| 5. Power supply resource select (J4)         | 14. Stereo codec                         |
| 6. USB device connector (CN4)                | 15. Programming mode jumper of FM0+ (J2) |
| 7. Potentiometer                             | 16. Cypress FM0+ MCU S6E1C32D0A          |
| 8. Programming mode jumper of MB9AF312K (J1) | 17. RGB LED                              |
| 9. Headphone and microphone jack (CN1)       | 18. Cypress 4-Mb SRAM                    |
|  | 19. 3.3V Voltage Regulator               |

このキットを使用しない場合、特定のターゲット ハードウェアに対応するように命令を修正する必要があります。ジャンパの設定およびその他の詳細情報については、使用する基板で提供されたドキュメントを参照してください。

### 3.1 始める前に

サイプレスは、FM0+デバイスで使用する2つのフラッシュプログラマを提供します。

- [FM0+/FM3/FM4 向けの FLASH MCU Programmer](#)
- [FLASH USB Direct Programmer](#)

S6E1C3 キットは FLASH USB Direct Programmer をサポートします。

このキットは FLASH MCU Programmer をサポートしていません。キットの CMSIS-DAP インターフェースは、プログラマが必要とする SIN/SOTO\_0 ピンを使用します。ほとんどのユーザーデザインでは、カスタムボード上に CMSIS-DAP インターフェース (U3) を実装しないと考えられます。その場合、USB 接続のないデザインには FLASH MCU Programmer が最適です。

これらのフラッシュプログラマの使用方法には、シングルステップまたは自動プログラミング (完全動作) の2つがあります。チップ消去が必要な保護されたフラッシュデバイスでは、シングルステップのみが機能することに注意してください。

また、ダウンロードするファイルが必要です。ファイルフォーマットは Motorola S-Record または Intel-HEX でなければなりません。この手順では、キットに同梱された Motorola S-Record ファイルを使用します。

IDE でコードをビルドする時、S-Record または Intel-HEX フォーマットのファイルを生成できる場合があります。使用する IDE のドキュメントを参照してください。IAR Embedded Workbench では、**Project > Options > Output Converter** を使用します。Keil µVision では、**Project > Options for Target > Output** パネルを使用します。

### 3.2 S6E1C3 キットでの FLASH USB DIRECT Programmer の使用

これらの手順では、スターターキットファイルをダウンロードしてインストールし、必要な S-Record ファイルにアクセスできることが前提です。そうでない場合、別の S-Record または Intel-HEX ファイルを探して、それを代わりに使用します。USB 接続は、ターゲット デバイスでの USB サポートが必要です。

#### 1. ジャンパの設定

ボードのジャンパが表 4 に従って配置されていることを確認してください。

表 4. FLASH USB DIRECT Programmer による S6E1C3 プログラミングのジャンパ設定

ジャンパ	初期値	USB によるフラッシュ書き込み	用途
J1	開放	開放	MB9AF312K (CMSIS-DAP) を実行モードに設定します。
J2	開放	クローズ	S6E2GM をプログラミングモードに設定します。
J3	ピン 2 とピン 3	ピン 2 とピン 3	USB プログラミングモードに設定します。
J4	ピン 1 とピン 2	ピン 2 とピン 3	USB コネクタ (CN4) から電力を取得

#### 2. 基板への電源供給

USB ケーブルを CN4 コネクタに接続してください。電源 LED (LED3) は点灯します (緑色)。正しいコネクタの位置については、[図 3](#) を参照してください。

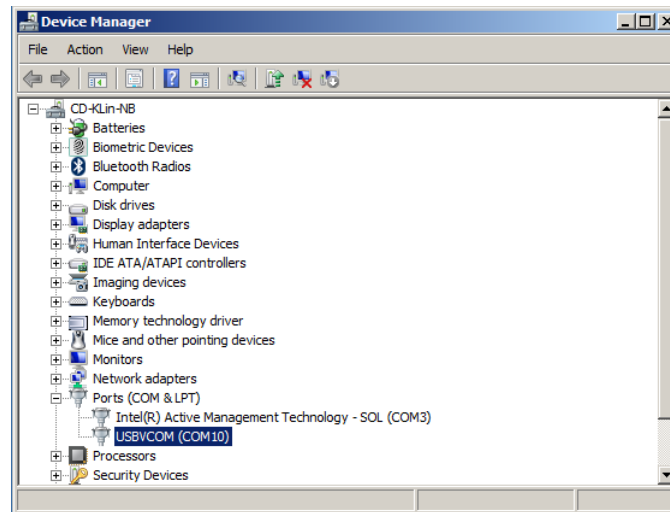
#### 3. 使用する COM ポートを識別

使用する基板がどの COM ポートに接続されているかを知る必要があります。この番号を使用して Flash Programmer を設定します。

[サイプレスシリアルポートビューアーおよびターミナルツール](#) (キットと共にインストールされる) を使用している場合ボード接続時に、この情報を含むポップアップ通知が表示されます。COM ポートがわからない場合は、デバイスマネージャーを開き、**Ports (COM & LPT)** を探します。USBVCOM のエントリが表示されます。[図 4](#) に示すように、COM ポートはそのエントリの最後に表示されます。

番号を控えてください。

図 4.1 使用中の COM ポートを特定する



#### 4. FLASH USB DIRECT Programmer の起動

デフォルトのインストールでは、プログラマは以下の場所です。

`C:\Program Files (x86)\Cypress\FLASH USB Direct Programmer`

#### 5. プログラマを構成します。

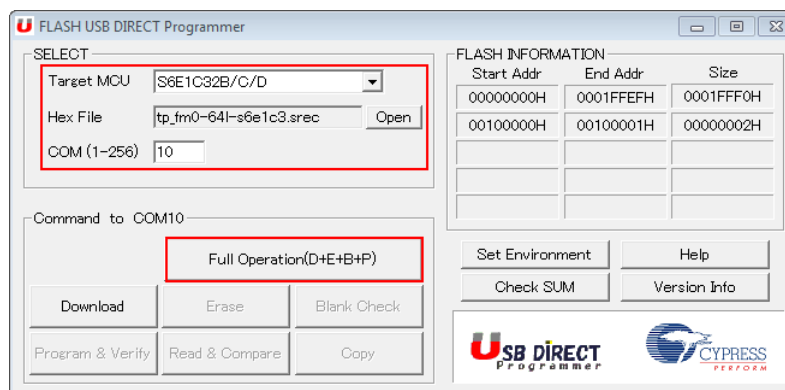
このステップでは、ターゲットデバイスのプログラマをセットアップします。図 5 を参照してください。

- A. Target MCU を S6E1C32B/C/D に設定します。
- B. Hex File を、ボードにフラッシュするファイルに設定します。

この例では、`tp_fm0-64l-s6e1c3.srec` を使用します。このファイルはスターターキット基板をその初期状態に復元します。S-Record ファイルは<キットディレクトリ>\Firmware\Demo Projects\Test\_Demo\_Codeにあります。

COM (1-256)をデバイスマネージャーに表示された値に設定してください。

図 5. プログラマの設定



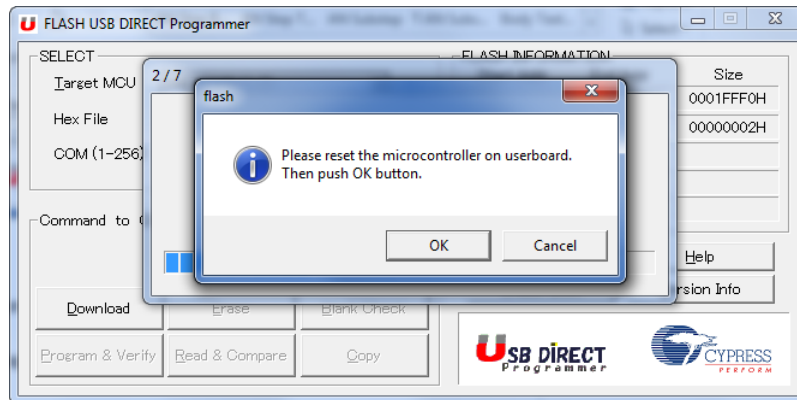


## 6. フラッシュへの書き込み

- A. **Full Operation(D+E+B+P)** ボタンをクリックしてプログラミングを開始してください。(注：フル操作は、セキュアされたフラッシュでは機能しません。シングル ステップを使用してください。) フラッシュ書き込みプロセスが開始されます。
- B. 基板をリセットしてください。図 6 に示すように、ボードのリセットスイッチ (SW1) を押し、**OK** をクリックしてください。

プロセス中にこれを 2 回以上行う必要がある場合があります。

図 62. MCU をリセットする



## 7. 基板を通常動作に復元

完了したら、ジャンパを元の構成に戻すか、表 4 に示すようにデフォルト値に戻してください。

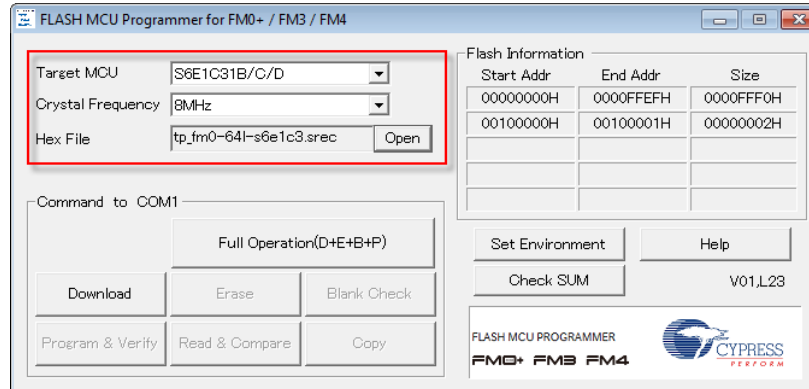
成功を確認するために、Serial Port Viewer and Terminal ツール(キットと共にインストールされます)を使用して基板に接続してデモ コードを実行します。詳しい手順は、スターターキットガイドに記載されています。

### 3.3 FLASH MCU Programmer の使用

セクション 3.1 始める前にで説明したように、S6E1C3 キットはこのプログラマをサポートしていません。したがって、このセクションでは、キットで FLASH MCU Programmer を使用する方法については説明しません。ただし、このプログラマは FM0+デバイスをサポートしているため、カスタム設計では有効な代替手段です。すべての FM0+デバイスはシリアル接続でプログラムできます。

FLASH USB DIRECT Programmer と同じようにプログラマを設定します。図 7 に示すように、ターゲット MCU と Crystal Frequency を設定し、hex ファイルを指定します。

図 7. プログラムの設定



**Set Environment** をクリックして、正しい COM ポートを指定します。詳細については、FLASH MCU Programmer のドキュメントを参照してください。

## 4 FM0+ポートフォリオ リソース

サイプレスは FM0+製品を理解し、生産性を向上させるために多くのリソースを提供します。表 5 を使用して、設計プロセスのどの段階にいるかに基づいて、必要なリソースを識別して選択します。

表 5. FM0+ポートフォリオリソースナビゲーター

目的	リソース
FM0+の評価	本書を利用してください。 サイプレスのウェブサイトでの <a href="#">FM0+製品ページ</a> を参照してください。 FM0+スターターキットを購入してください。 <a href="#">FM0+ S6E1A1 MCU 評価基板</a> <a href="#">FM0+ S6E1C3 シリーズ スターターキット</a> <a href="#">FM0+データシート</a> を参照してください。 <a href="#">AN202487- Differences Among FM0+, FM3, and FM4 Families</a> を参照してください。
FM 製品の選択	<a href="#">製品選択ガイド</a> をダウンロードして確認してください。 <a href="#">AN202487- Differences Among FM0+, FM3, and FM4 Families</a> を参照してください。
ハードウェア設計の理解	<a href="#">AN203277-- FM 32-bit Microcontroller Family Hardware Design Considerations</a> を参照してください。
用意されているソフトウェア開発ツールの理解	<a href="#">IAR Embedded Workbench</a> <a href="#">Keil μVision IDE</a> <a href="#">iSYSTEM winIDEA</a> <a href="#">GCC ARM Embedded</a>
ペリフェラルドライバライブラリの理解	<a href="#">FM0+ S6E1C3 シリーズスターターキット</a> を購入してください。 <a href="#">PDL</a> をダウンロードして、 <a href="#">PDL クイックスタートガイド</a> を参照してください。 <a href="#">PDL クイックスタートガイド</a> の「 <a href="#">PDL プロジェクトの構築と実行</a> 」セクションを参照してください。 PDL と共にインストールされた PDL サンプルコードを参照してください。
特定の FM0+ ペリフェラルの理解	<a href="#">FM0+ 関連のアプリケーションノート</a> を検索してください。例: <a href="#">AN202483 - FM0+ S6E1A1 Series MCU Low-Voltage 3-Phase BLDC and PMSM Control</a> <a href="#">AN204389 - FM0+ Family 3-Phase ACIM Scalar Control</a> <a href="#">AN99231- FM0+ファミリの S6E1C3 シリーズにおける割込みの使用について</a>

目的	リソース
FM0+ の低レベルソフトウェアの開発	<p>PDL クイックスタートガイドの「カスタムプロジェクトの作成」セクションを参照してください。</p> <p>PDL デバイスフォルダーのプロジェクトファイルとスタートアップコードを使用してください。</p> <p>レジスタレベルのプログラミング技術の習得のために PDL ソース コードを使用してください。</p> <p><a href="#">FM0+ データシート</a>を参照してください。</p> <p><a href="#">FM0+ ペリフェラルマニュアル</a>をテクニカルリファレンスとして使用してください。</p>
フラッシュ プログラミングの理解	<p>フラッシュプログラマを入手</p> <ul style="list-style-type: none"> <li><a href="#">FM0+/FM3/FM4 向けの FLASH MCU Programmer</a></li> <li><a href="#">FLASH USB DIRECT Programmer</a></li> </ul> <p>本ドキュメントの「<a href="#">内蔵フラッシュメモリへの書き込み</a>」セクションを参照してください。</p> <p>特定の FM0+ シリーズのフラッシュ プログラミング マニュアルを参照してください。</p> <ul style="list-style-type: none"> <li><a href="#">S6E1Ax</a>    <a href="#">S6E1Cx</a></li> </ul> <p><a href="#">AN204438 - How to Setup Flash Security for FM0+, FM3 and FM4 Families</a> を参照してください。</p>

## 著者について

名前: James Trudeau

役職: Senior Principal Application Engineer

経歴: James Trudeau はサイプレスのシニアプリンシパルアプリケーションエンジニアです。

## 改訂履歴

文書名: AN210985 – FM0+ファームウェア開発入門

文書番号: 02-

版	変更内容
**	本版は英語版 002-10985 Rev. *D について、CYPRESS DEVELOPER COMMUNITY の参加者によって日本語に翻訳されたドキュメントです。

## セールス、ソリューションおよび法律情報

### ワールドワイドな販売と設計サポート

サイプレスは、事業所、ソリューションセンター、メーカー代理店、および販売代理店の世界的なネットワークを保持しています。お客様の最寄りのオフィスについては、[サイプレスのロケーションページ](#)をご覧ください。

#### 製品

Arm® Cortex® Microcontrollers	<a href="http://cypress.com/arm">cypress.com/arm</a>
車載用	<a href="http://cypress.com/automotive">cypress.com/automotive</a>
クロック&バッファ	<a href="http://cypress.com/clocks">cypress.com/clocks</a>
インターフェース	<a href="http://cypress.com/interface">cypress.com/interface</a>
IoT (モノのインターネット)	<a href="http://cypress.com/iot">cypress.com/iot</a>
メモリ	<a href="http://cypress.com/memory">cypress.com/memory</a>
マイクロコントローラ	<a href="http://cypress.com/mcu">cypress.com/mcu</a>
PSoC	<a href="http://cypress.com/psoc">cypress.com/psoc</a>
電源用 IC	<a href="http://cypress.com/pm/c">cypress.com/pm/c</a>
タッチセンシング	<a href="http://cypress.com/touch">cypress.com/touch</a>
USB コントローラー	<a href="http://cypress.com/usb">cypress.com/usb</a>

ワイヤレス [cypress.com/wireless](http://cypress.com/wireless)

#### PSoC®ソリューション

[PSoC 1](#) | [PSoC 3](#) | [PSoC 4](#) | [PSoC 5LP](#) | [PSoC 6 MCU](#)

#### サイプレス開発者コミュニティ

[コミュニティ](#) | [サンプルコード](#) | [Projects](#) | [ビデオ](#) | [ブログ](#) | [トレーニング](#) | [Components](#)

#### テクニカルサポート

[cypress.com/support](http://cypress.com/support)



Cypress Semiconductor  
An Infineon Technologies Company  
198 Champion Court  
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2016-2020. 本書面は、Cypress Semiconductor Corporation 及び Spansion LLC を含むその子会社 (以下「Cypress」という。) に帰属する財産である。本書面 (本書面に含まれ又は言及されているあらゆるソフトウェア若しくはファームウェア (以下「本ソフトウェア」という。)) を含むは、アメリカ合衆国及び世界のその他の国における知的財産法令及び条約に基づき Cypress が所有する。Cypress はこれらの法令及び条約に基づく全ての権利を留保し、本段落で特に記載されているものを除き、その特許権、著作権、商標権又はその他の知的財産権のライセンスを一切許諾しない。本ソフトウェアにライセンス契約書が伴っておらず、かつ Cypress との間で別途本ソフトウェアの使用方法を定める書面による合意がない場合、Cypress は、(1) 本ソフトウェアの著作権に基づき、(a) ソースコード形式で提供されている本ソフトウェアについて、Cypress ハードウェア製品と共に用いるためにのみ、かつ組織内部でのみ、本ソフトウェアの修正及び複製を行うこと、並びに (b) Cypress のハードウェア製品ユニットに用いるためにのみ、(直接又は再販売者及び販売代理店を介して間接のいずれか) 本ソフトウェアをバイナリーコード形式で外部エンドユーザーに配布すること、並びに (2) 本ソフトウェア (Cypress により提供され、修正がなされていないもの) が抵触する Cypress の特許権のクレームに基づき、Cypress ハードウェア製品と共に用いるためにのみ、本ソフトウェアの作成、利用、配布及び輸入を行うことについての非独占的で譲渡不能な一身専属的ライセンス (サブライセンスの権利を除く) を付与する。本ソフトウェアのその他の使用、複製、修正、変換又はコンパイルを禁止する。

適用される法律により許される範囲内で、Cypress は、本書面又はいかなる本ソフトウェア若しくはこれに伴うハードウェアに関しても、明示又は黙示をとわず、いかなる保証 (商品性及び特定の目的への適合性の黙示の保証を含むがこれらに限られない) も行わない。いかなるコンピューティングデバイスも絶対に安全というわけではない。従って、Cypress のハードウェアまたはソフトウェア製品に講じられたセキュリティ対策にもかかわらず、Cypress は、Cypress 製品への権限のないアクセスまたは使用といったセキュリティ違反から生じる一切の責任を負わない。加えて、本書面に記載された製品には、エラーと呼ばれる設計上の欠陥またはエラーが含まれている可能性があり、公表された仕様とは異なる動作をする場合がある。適用される法律により許される範囲内で、Cypress は、別途通知することなく、本書面を変更する権利を留保する。Cypress は、本書面に記載のある、いかなる製品若しくは回路の適用又は使用から生じる一切の責任を負わない。本書面で提供されたあらゆる情報 (あらゆるサンプルデザイン情報又はプログラムコードを含む) は、参照目的のために提供されたものである。この情報で構成するあらゆるアプリケーション及びその結果としてのあらゆる製品の機能性及び安全性を適切に設計、プログラム、かつテストすることは、本書面のユーザーの責任において行われるものとする。Cypress 製品は、兵器、兵器システム、原子力施設、生命維持装置若しくは生命維持システム、蘇生用の設備及び外科的移植を含むその他の医療機器若しくは医療システム、汚染管理若しくは有害物質管理の運用のために設計され若しくは意図されたシステムの重要な構成部分としての使用、又は装置若しくはシステムの不具合が人身傷害、死亡若しくは物的損害を生じさせるようなその他の使用 (以下「本目的外使用」という。) ためには設計、意図又は承認されていない。重要な構成部分とは、その不具合が装置若しくはシステムの不具合を生じさせるか又はその安全性若しくは実効性に影響すると合理的に予想できるような装置若しくはシステムのあらゆる構成部分をいう。Cypress 製品のあらゆる本目的外使用から生じ、若しくは本目的外使用に関連するいかなる請求、損害又はその他の責任についても、Cypress はその全部又は一部をとわず一切の責任を負わず、かつ Cypress はそれら一切から本書により免除される。Cypress は Cypress 製品の本来目的外使用から生じ又は本目的外使用に関連するあらゆる請求、費用、損害及びその他の責任 (人身傷害又は死亡に基づく請求を含む) から免責補償される。

Cypress, Cypress のロゴ, Spansion, Spansion のロゴ及びこれらの組み合わせ, WICED, PSoC, Capsense, EZ-USB, F-RAM, 及び Traveo は、米国及びその他の国における Cypress の商標又は登録商標である。Cypress のより完全な商標のリストは、[cypress.com](http://cypress.com) を参照すること。その他の名称及びブランドは、それぞれの権利者の財産として権利主張がなされている可能性がある。