本ドキュメントは Cypress (サイプレス) 製品に関する情報が記載されております。本ドキュメントには、「MB」から始まるシリーズ名，品名およびオーダ型格が記載されておりますが、これらはすべて「CY」から始まるシリーズ名，品名およびオーダ型格として、新規および既存のお客様に引き続き提供してまいります。

**オーダ型格の調べ方について**

1. www.cypress.com/pcnにアクセスしてください。
2. SEARCH PCNS フィールドに、オーダ型格などのキーワードを入力し、「Apply」をクリックしてください。
3. 該当するタイトル(Title)をクリックしてください。
4. 「Affected Parts List」ファイルを開いてください。
   当該ファイルに記載されている各種変更情報をご利用ください。

**詳しいお問い合わせ先**

Cypress 製品およびそのソリューションの詳細につきましては、お近くの営業所へお問い合わせください。

**サイプレスについて**

サイプレスは、世界で最も革新的な車載や産業機器、スマート家電、民生機器および医療機器製品向けに、最先端の組み込みシステム ソリューションを提供するリーディング カンパニーです。サイプレスのマイクロコントローラーや、アナログ IC、ワイヤレスおよび USB ベースのコネクティビティ ソリューション、高い信頼性と高性能を提供するメモリ製品は、各種機器メーカーの差異化製品の開発と早期市場参入を支援します。サイプレスは、ベストクラスのサポートと開発リソースをグローバルに提供することで、彼らが従来市場を破壊しまったく新しい製品カテゴリを歴史的なスピードで市場投入できるよう支援します。詳細はサイプレスのウェブサイト (japan.cypress.com) をご覧ください。

# AN203980

## FM4 S6E2Cx シリーズ Over the Air Update 32-Bit Microcontroller with Embedded Dual Flash

**Associated Part Family:** **S6E2Cx シリーズ**

本書では、Over The Air update with embedded Dual Flash Memory (以降、OTA-D) 用リファレンスソフトウェアについて解説します

## Contents


はじめに. . . . . 1
対象製品. . . . . 1
1 概要. . . . . 2
1.1 背景. . . . . 2
2 OTA-Dリファレンスソフトウェア . . . . . 3
2.1 基本機能. . . . . 3
2.2 OTA-D 部のサーバ・クライアント間動作シーケンス . 8
2.3 機能詳細. . . . . 9
3 OTA-Dリファレンスソフトウェアの準備 . . . . . 15
3.1 準備手順概略. . . . . 15
3.2 SK-FM4-216-ETHERNETの準備. . . . . 16
3.3 PCの準備. . . . . 16
3.4 ブートローダ. . . . . 19
3.5 OTA-Dライブラリ . . . . . 21
3.6 ユーザソフトウェア. . . . . 21
4 OTA-Dリファレンスソフトウェアの実行 . . . . . 27
5 OTA-D応用例 . . . . . 30
6 Document History. . . . . 31


## はじめに

**本書の構成**

本書は、「SK-FM4-216-ETHERNET Wi-Fi Sample Kit Setup Manual（英語版）」と併せてお読みください。

本書を読む前に、以下の参考文書をお読みいただくことを推奨します。

**参考文書**

フラッシュメモリのプログラミングマニュアル

cypress.com/downloads/s6e2cc_mn709-00007-j.pdf

S6E2Cxシリーズ データシート

cypress.com/downloads/s6e2cc_ds709-00009-j.pdf

## 対象製品

本ドキュメントに記載されている内容の対象製品は、下記のとおりです。

| シリーズ名 | 品種型格（パッケージサフィックスは除く） |
|---|---|
| S6E2Cx シリーズ | S6E2C1AH, S6E2C1AJ, S6E2C1AL<br>S6E2C2AH, S6E2C2AJ, S6E2C2A,<br>S6E2C3AH, S6E2C3AJ, S6E2C3AL<br>S6E2C4AH, S6E2C4AJ, S6E2C4AL<br>S6E2C5AH, S6E2C5AJ, S6E2C5AL<br>S6E2CCAH, S6E2CCAJ, S6E2CCAL |

# 1 概要

本書では、Over The Air update with embedded Dual Flash Memory（以降、OTA-D）用リファレンスソフトウェアについて解説します。このOTA-D 用リファレンスソフトウェアは、内蔵フラッシュメモリを2 面持つMCU を使用したシステム上で、無線通信用ソフトウェアとユーザソフトウェアを更新します。
本更新方法では、片側の内蔵フラッシュメモリで動作している無線用ソフトウェアおよびユーザソフトウェアが、最新版のソフトウェアをサーバからダウンロードし、他方の内蔵フラッシュメモリに保存します。その後、システムを再起動することによって2 面ある内蔵フラッシュメモリのどちらを実行するか判断し、最新版ソフトウェアを実行します。

**注意事項**

- リファレンスソフトウェアは、予告なしに変更することがあります。また、リファレンスソフトウェアは*SK-FM4-216-ETHERNET* での標準的な使い方を示したものです。このため、お客様のシステム上でご使用の際は、十分評価された上でご使用ください。
- 本書と併せて提供される *OTA-D* 用リファレンスソフトウェアは、参考用として用意したものです。このため、ソフトウェアアップデートシステムの自己開発の参考用途以外に使用しないでください。リファレンスソフトウェアの使用に起因し生じた損害については、当社は一切その責任を負いません。

## 1.1 背景

近年、様々なアプリケーションに対して様々なユーザソフトウェアが使用されています。近年までは、アプリケーションの市場への出荷後は、ユーザソフトウェア更新は行われず、更新の場合は製品を回収して部品を差し替えるのが一般的でした。しかし最近では、アプリケーションで使用されるユーザソフトウェアの高度化や多様化、また開発の短納期化に伴い、市場への出荷済み製品に組み込まれたユーザソフトウェアのアップデートが必要とされています。
そこで、通信ネットワークの高度化に伴い、製品を回収することなしに無線を経由してアプリケーションに内蔵されたソフトウェアの更新を行うOver The Air update（以降、OTA）という方法が提唱され始めております。

# 2 OTA-Dリファレンスソフトウェア

OTA-D リファレンスソフトウェアは、S6E2Cx シリーズに搭載されている※MCU内蔵デュアルバンクフラッシュメモリとリマップという機能を利用します。

※MCU 内蔵デュアルバンクフラッシュメモリとリマップ機能については冒頭に記載した参考文書を参照してください。

※本ドキュメントは、OTA-D リファレンスソフトウェアバージョン 1.0 を対象としています。

## 2.1 基本機能

OTA-D リファレンスソフトウェアは、①ブートローダ部，②インフォメーション領域，③OTA-D 部，④ユーザソフトウェア部の4 つから構成されるOTA-D 機能を実現するための参考ソフトウェアアプリケーションです。

ブートローダ部はシステム起動時に最初に実行するプログラムです。インフォメーション領域はフラッシュメモリに書き込んだOTA-D 部とユーザソフトウェア部の情報を保存する領域です。OTA-D 部は、ユーザソフトウェアに組み込んで使用するライブラリです。ユーザソフトウェアはお客様が作成するプログラムで、本書ではSK-FM4-216-ETHERNET のLCD の表示を行うプログラムとして説明しています。

"Wi-Fi module sample kit" (SK-FM4-216-ETHERNET, Wi-Fi board, SSL, TCP/IP, RTOS および関連するDriver 等)に合わせてOTA-D リファレンスソフトウェアを利用して、サーバから更新対象プログラムの最新版をダウンロードし、MCU 内蔵フラッシュメモリ内の旧版を更新することができます。OTA-D リファレンスソフトウェアの処理フローをFigure 2-1、フラッシュメモリへの書込みとリマップによるフラッシュメモリの切替えフロー(OTA-D リファレンスソフトウェアの物理メモリアドレスマップ) をFigure 3 に示します。

Figure 1. OTA-Dリファレンスソフトウェア処理フロー

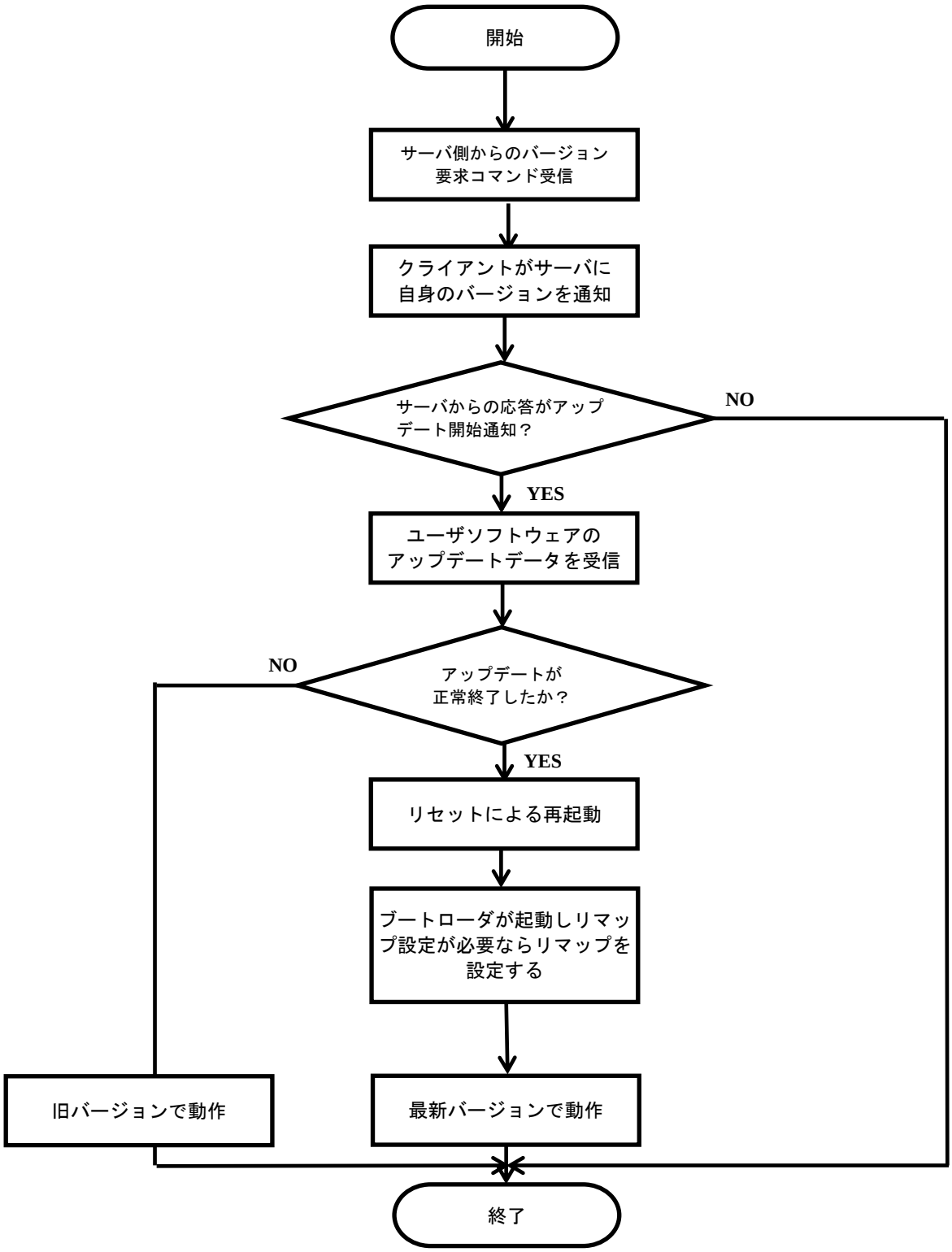


Figure 2. OTA-Dリファレンスソフトウェアの物理メモリアドレスマップ

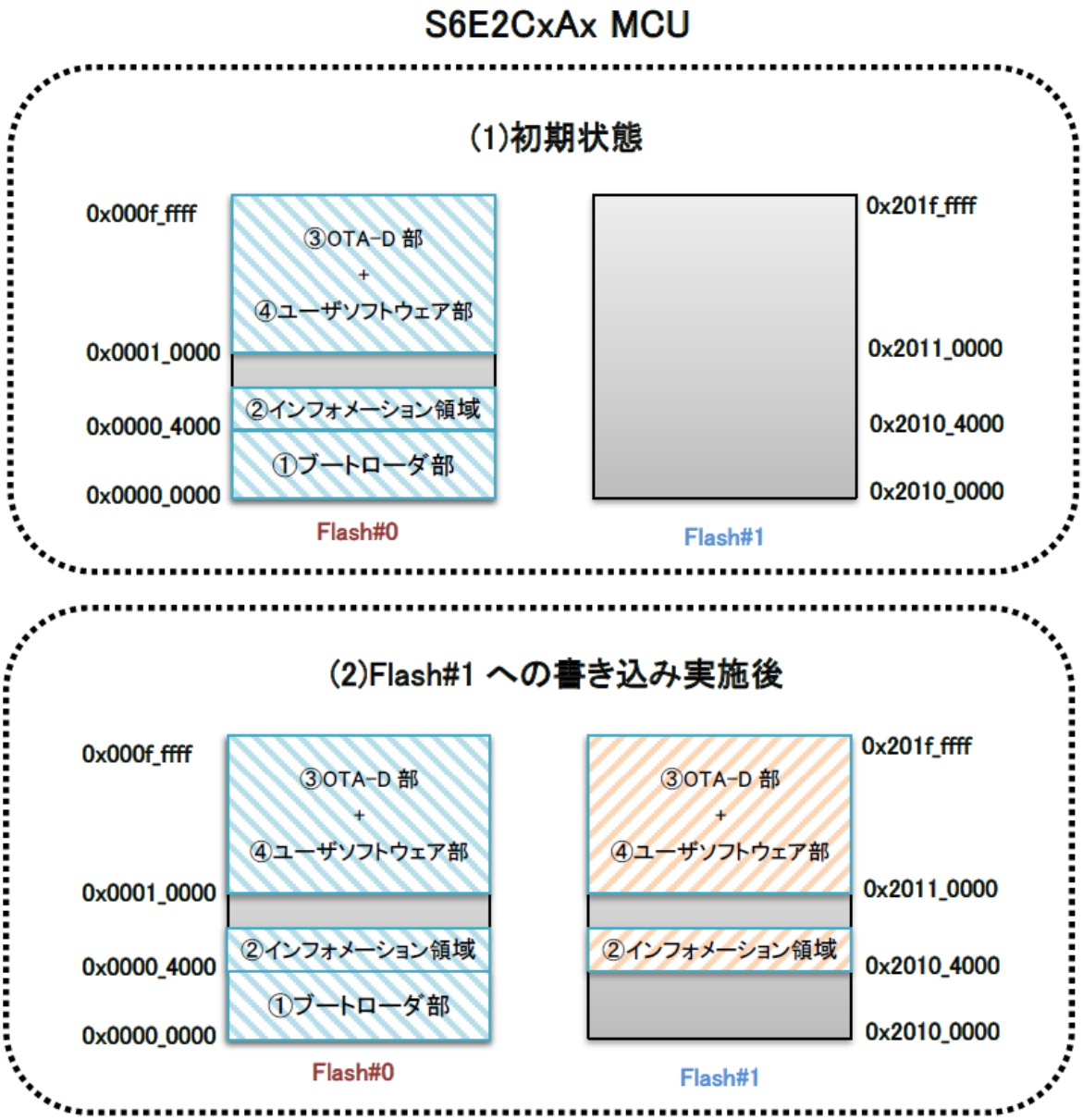
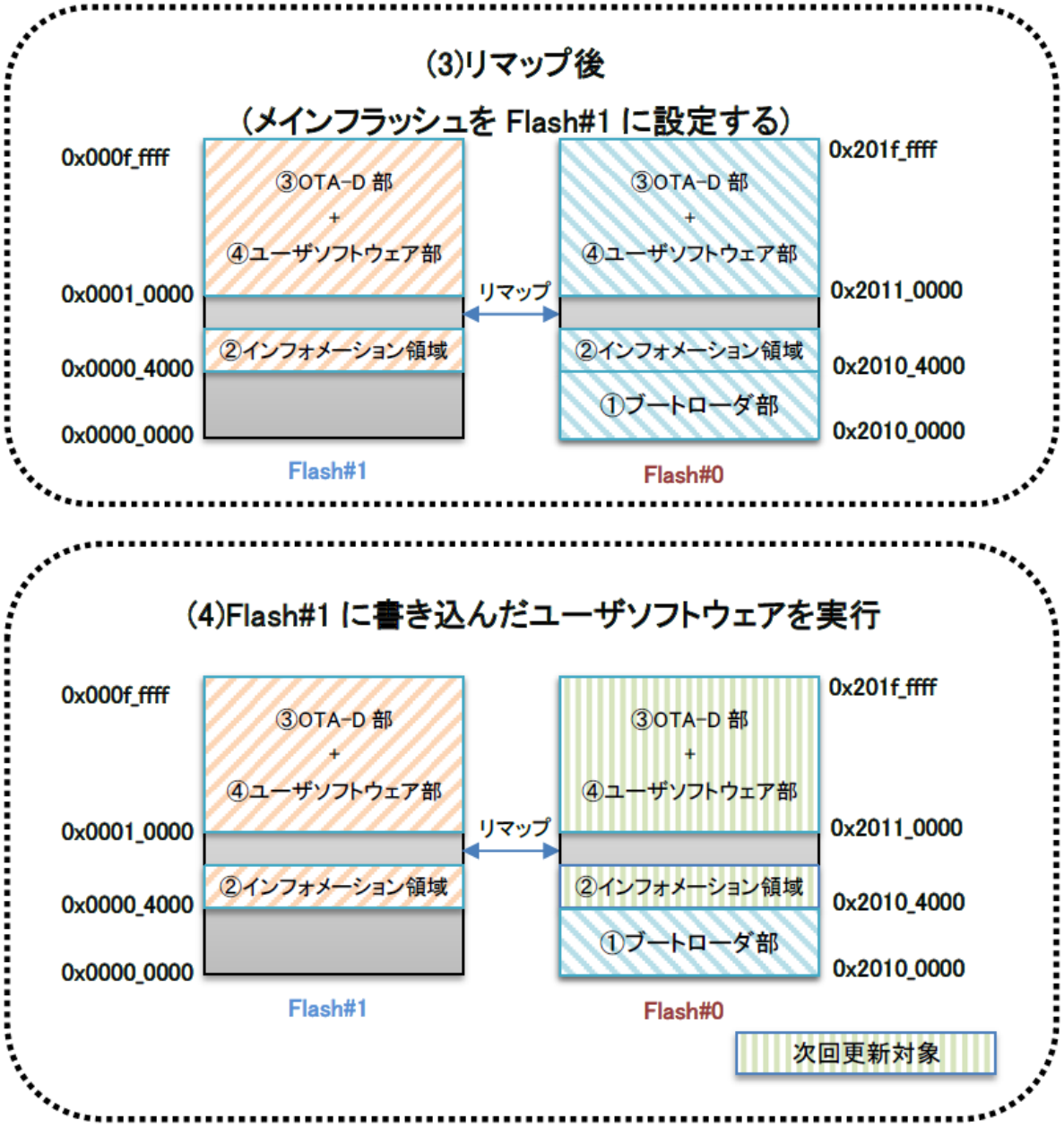


Figure 3. OTA-Dリファレンスソフトウェアの物理メモリアドレスマップ(続き)



### 2.1.1 ブートローダ部

ブートローダ部は常にFlash#0 バンクに常駐します。システム起動時に読みだされ、Flash#0 バンクのインフォメーション領域情報に基づいて、MCU 内OTA-D 部とユーザソフトウェア部の最新版が物理メモリアドレス上のFlash#0, Flash#1 のいずれにあるかを判断します。

デュアルフラッシュモードに設定後(*)、ユーザソフトウェアの最新版がFlash#0 にある場合はOTA-D 部とユーザソフトウェア部の先頭の物理メモリアドレス(0x0001_0000)にジャンプします。OTA-D 部とユーザソフトウェア部がFlash#1 にある場合はリマップ機能を使用し(*)、OTA-D 部とユーザソフトウェア部の先頭の物理メモリアドレス(0x0001_0000)にジャンプします。

(*)OTA-D 部の該当機能をSRAM に転送し実行していま

### 2.1.2 インフォメーション領域

インフォメーション領域は、フラッシュメモリに書き込んだOTA-D 部とユーザソフトウェア部の情報を保存する領域です。

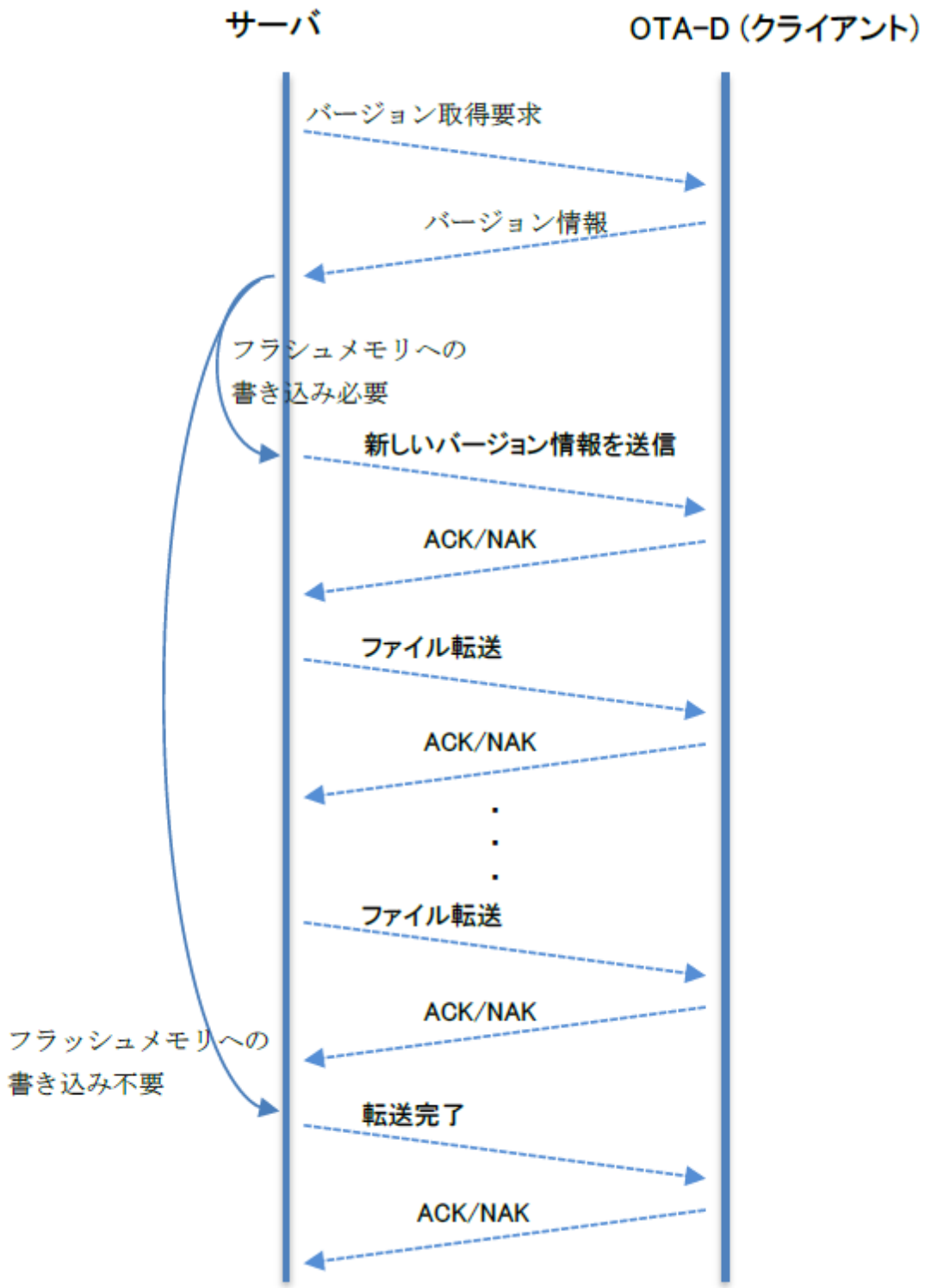
### 2.1.3 OTA-D部

OTA-D 部は、ユーザソフトウェアに組み込むライブラリです。OTA-D 部のサーバ・クライアント間動作シーケンスを「2.2 OTA-D 部のサーバ・クライアント間動作シーケンス」に示します。

### 2.1.4 ユーザソフトウェア部

ユーザソフトウェア部は、OTA-D ライブラリをリンクして使用します。OTA-D リファレンスソフトウェアは、ユーザソフトウェア部とWi-Fi Chip set 付属ソフトウェア群（RTOS 含む）と組み合わせて使用することで、同時にダウンロードしフラッシュメモリに書き込むことができます。

## 2.2 OTA-D 部のサーバ・クライアント間動作シーケンス



## 2.3 機能詳細

OTA-Dリファレンスソフトウェアの機能やインタフェースについて説明します。

以降、「ユーザソフトウェア」は、OTA-D部とユーザソフトウェア部を示します。

### 2.3.1 ブートローダ部

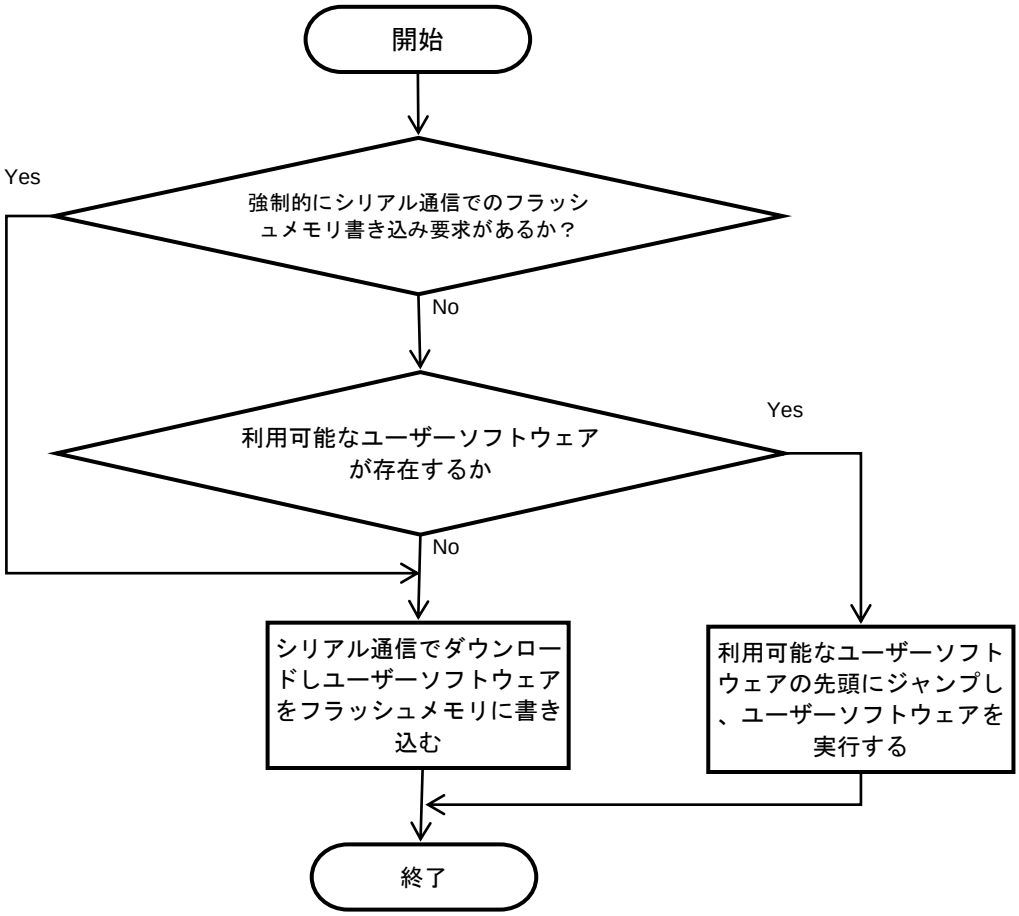
ブートローダ部は、MCUのリセット直後に動作します。ブートローダ部は、以下の機能を提供します。

- ユーザが、シリアル通信を使用して、強制的にフラッシュメモリに書き込む機能。
- インフォメーション領域の情報を比較して、2 面ある内蔵フラッシュメモリのどちらが最新のユーザソフトウェアかを判断する機能。
- 最新のユーザソフトウェアを選択し、ユーザソフトウェアの先頭の物理メモリアドレス(0x0001_0000)にジャンプし、ユーザソフトウェアを実行する機能。
- 利用可能なユーザソフトウェアが存在しないと判断した場合、ユーザソフトウェアをブートローダ部の機能として、シリアル通信を使用してダウンロードしフラッシュメモリに書き込む機能。

ブートローダ単体ではWi-Fi で通信できません。そのため、フラッシュメモリにまだユーザソフトウェアが存在しない場合は、シリアル通信を使って、ユーザソフトウェアをフラッシュメモリに書き込みます。詳細は「3.6.2 ユーザソフトウェアのシリアル通信によるフラッシュメモリへの書込み」を参照してください。

ブートローダ部の動作フローをFigure 4 に示します。

Figure 4. ブートローダ部の動作フロー



### 2.3.2 インフォメーション領域

ユーザソフトウェアの情報は、インフォメーション領域に保存します。インフォメーション領域に格納するデータは Table 1 に示します。

Table 1. インフォメーション領域に格納するデータ

| 0x0000_4000 / 0x2010_4000 | +3 | +2 | +1 | +0 |
|---|---|---|---|---|
| | app load count | | check sum | |

check sum：インフォメーション領域のチェックサム

ユーザソフトウェアのフラッシュメモリへの書込みが完了すると、チェックサムを計算し算出結果を書き込みます。

チェックサムの計算方法：check sum = ~SUM + 1 + MAGIC_NUM

(SUMとは、チェックサムを除き、ほかのインフォメーション領域（6バイト）の情報をsigned half-word形式で加算した結果です。MAGIC_NUMとはマジックナンバーであり、0x1C5Aと定義します)

app load count：ユーザソフトウェアのダウンロードカウント

ユーザソフトウェアのフラッシュへの書込みが完了したら、

0x0000_4002へ書き込む場合は0x2010_4002の値+1を書き込みます 。

0x2010_4002へ書き込む場合は0x0000_4002の値+1を書き込みます。

| 0x0000_4004 / 0x2010_4004 | +3 | +2 | +1 | +0 |
|---|---|---|---|---|
| | version_low | | version_high | |

version_high：ユーザソフトウェアのバージョン (上位16ビット) を格納します。

version low ：ユーザソフトウェアのバージョン (下位16ビット) を格納します。

「コマンドの種類」の「(2) バージョン情報の送信」で説明しているVersionをversion_highに、Levelをversion_lowに書込みます。

このリファレンスソフトウェアでのユーザソフトウェア更新履歴の判断基準は以下の通りです。

Flash#0、Flash#1についてはFigure 2を参照してください。
version_highはversion_lowの値に関わらず、Flash#0とFlash#1の数値の大きい方を選択します。

version_lowはversion_highが同じ場合、version_lowの数値の大きいFlash#0またはFlash#1を選択します。

version_highもversion_lowも同じ場合、app load countの数値の大きいFlash#0またはFlash#1を選択します。

例）版数2.08の管理

version_high=2、version_low=08

### 2.3.3 OTA-D部

**ota_version_section**

“ota_version_section”はコードセクションの名前です。このコードセクションにOTA-D部のバージョン情報を格納します。

インストールしたフォルダのproject\spansion_s6e2cx_8189em_with_ota\EWARM-RELEASE\configにあるEWARMプロジェクトのicfリンクファイルの下記の箇所で、ota_version_sectionコードセクションを配置するメモリ領域を指定しています。

```
define symbol __OTA_LIB_version__start__ = 0x00090000;
place at address mem: __OTA_LIB_version__start__ { section ota_version_section };
```

0x00090000はユーザが指定するメモリ領域の開始アドレスです。

**ota_section**

“ota_section”は、コードセクションの名前です。このコードセクションにユーザソフトウェアを格納します。

インストールしたフォルダのproject\spansion_s6e2cx_8189em_with_ota\EWARM-RELEASE\configにあるEWARMプロジェクトのicfリンクファイルの下記の箇所で、ota_sectionコードセクションを配置するメモリ領域を指定しています。

```
define symbol __OTA_LIB_start__ = 0x00090010;
place at address mem: __OTA_LIB_start__ { section ota_section };
```

0x00090010はユーザが指定するメモリ領域の開始アドレスです。

**ota_version**

OTA-D部のota_version_t型変数です。この変数は、OTA-D部のバージョン情報です。

Table 2. ota_version_t型の構造体一覧

| 型名 | メンバー | データ型 | 説明 |
|---|---|---|---|
| ota_version_t | version_high | uint16_t | OTA-Dライブラリのバージョン情報（上位16ビット） |
| | version_low | uint16_t | OTA-Dライブラリのバージョン情報（下位16ビット） |

**ota_update_process**

- C言語インタフェース

  int errorcode = ota_update_process(ota_parameters_t *ota_parameters)

- パラメータ

  入力：ota_parameters　　ユーザソフトウェアから渡されたサーバ情報

  出力：errorcode　　OTAアップデート関数からの戻り値

Table 3. OTA_Parameters_t型の構造体一覧

| 型名 | 定義 | データ型 | 説明 |
|---|---|---|---|
| ota_parameters_t | ipaddress | unsigned char * | OTAサーバのIPアドレス |
| | port | unsigned short | OTAサーバのポート番号 |
| | ssl_flg | unsigned short | SSLの使用・未使用を決めるフラグ<br>1:SSLを使う<br>0:SSLを使わない |

Table 4. OTAアップデート関数の戻り値

| マクロ名 | 戻り値 | 説明 |
|---|---|---|
| OTA_UPDATE_SUCESS | 0 | OTAアップデートが成功 |
| OTA_NETWORK_ERROR | -1 | ネットワーク系エラーが発生 |
| OTA_VERSION_ERROR | -2 | すでに最新版であるため、アップデートは不要 |
| OTA_OTHER_ERROR | -3 | その他のエラー |

本関数は、ota_parametersパラメータを通してOTAサーバのIPアドレスとポート番号情報を取得し、ユーザソフトウェアをアップデートします。関数の戻り値によってOTAアップデートの実行結果を判断します。

**コマンドパケット**

各データの定義を以下に示します。() 内の数値は設定値を表します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| STX | 1 | (0x02)コマンドパケットの開始データを表します |
| LEN | 2 | コマンドパケットの長さ (IDX, CMD, DATAの総バイト数) |
| IDX | 2 | (0～65535)データパケット番号 |
| CMD | 1 | コマンドの種類<br>サーバ側 →クライアント側<br>(0x05)バージョン情報の取得:クライアントで動作しているユーザソフトウェアのバージョン情報を要求します<br>(0x01)バージョン情報の送信:サーバで管理しているユーザソフトウェアのバージョン情報を送信します<br>(0x02)ファイル転送:ファイルの転送を行います<br>(0x04)転送完了:ファイルの転送完了を通知します<br>クライアント側 → サーバ側<br>(0x05)クライアントで動作しているユーザソフトウェアのバージョン情報を送信します<br>(0x06)肯定応答を表します（エラーなし）<br>(0x15)否定応答を表します（エラー発生） |
| DATA | N | データエリア（具体的な形式は、CMDの種類によって変化します） |
| CS | 1 | チェックサム (IDX, CMD, DATAの2の補数：CS = ~(IDX+CMD+DATA) + 1) |
| ETX | 1 | (0x03)コマンドパケットの最終データを表します |

**コマンドの種類**

コマンドの種類について説明します。

**1. バージョン情報の取得(CMD=0x05)**

実行しているユーザソフトウェアのバージョン情報を取得します。

送信時にデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| - | 1 | 未使用 (0x00) |

OTA-D部が返信する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| version_high | 2 | クライアント側の最新ユーザソフトウェアのバージョン<br>（上位16ビット） |
| version_low | 2 | クライアント側の最新ユーザソフトウェアのバージョン<br>（下位16ビット） |

**2. バージョン情報の送信(CMD=0x01)**

これから書き込むユーザソフトウェアのバージョン情報を通知します。

送信時にデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| version_high | 2 | サーバ側のユーザソフトウェアのバージョン（上位16ビット） |
| version_low | 2 | サーバ側のユーザソフトウェアのバージョン（下位16ビット） |

OTA-D部が肯定応答する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| ErrorCode | 1 | 未使用 (Don’t care) |

OTA-D部が否定応答する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| ErrorCode | 1 | エラーコード<br>-1: データ受信エラー |

**3. ファイル転送(CMD=0x02）**

ユーザソフトウェアをフラッシュメモリに書き込みます。

送信時にデータエリア (DATA) に格納するデータを以下に示します。IDX は1 から始まりパケットごとに +1 します。65535 を超えたら0 に戻ります。

| 名前 | バイト数 | 説明 |
|---|---|---|
| StartAddress | 4 | メモリ上のソフトウェアデータの開始アドレス |
| ProgramData | 256 (Max) | プログラムデータ。最大256バイト |

OTA-D部が肯定応答する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| ErrorCode | 1 | 未使用 (Don’t care) |

OTA-D部が否定応答する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| ErrorCode | 1 | エラーコード<br>-1: データ受信エラー<br>-2: フラッシュへの書込み失敗 |

**4. 転送完了(CMD=0x04)**

ユーザソフトウェアの転送完了を通知します。

送信時にデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| - | 1 | 未使用 (0x00) |

OTA-D部が肯定応答する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| ErrorCode | 1 | 未使用 (Don't care) |

OTA-D部が否定応答する場合のデータエリア (DATA) に格納するデータを以下に示します。

| 名前 | バイト数 | 説明 |
|---|---|---|
| ErrorCode | 1 | エラーコード<br>-1: データ受信エラー<br>-2: フラッシュへ書込み失敗 |

### 2.3.4 ユーザソフトウェア

ユーザソフトウェアの具体的な構成例は、SK-FM4-216-ETHERNET Wi-Fi Sample Kit Setup Manual（英語版）の3.3 Software Block Diagram を参照してください。

# 3 OTA-Dリファレンスソフトウェアの準備

本章ではOTA-Dリファレンスソフトウェアを使用するための準備について説明します。

## 3.1 準備手順概略

1. ブートローダを作成しフラッシュメモリに書き込んでください。
   詳細は「3.4ブートローダ」を参照してください。
2. OTA-Dライブラリを作成してください。
   詳細は「3.5 OTA-Dライブラリ」を参照してください。
3. ユーザソフトウェアとOTA-Dライブラリをリンクしてください。
   詳細は「3.6 ユーザソフトウェア」を参照してください。
4. ユーザソフトウェアをフラッシュメモリに書き込んでください (1回目)。
   SK-FM4-216-ETHERNETボードとPCをシリアルケーブルで接続し、PC側でシリアル専用のTCPIP OTAサーバを起動してフラッシュメモリに書き込みます。
   詳細は「3.6.2 ユーザソフトウェアのシリアル通信によるフラッシュメモリへの書込み」を参照してください。

※なお、SK-FM4-216-ETHERNETとPCの事前準備については、「3.2 SK-FM4-216-ETHERNETの準備」と「3.3 PCの準備」に記載しています。

## 3.2 SK-FM4-216-ETHERNETの準備

SK-FM4-216-ETHERNETを準備するには、SK-FM4-216-ETHERNET Wi-Fi Sample Kit Setup Manual （英語版）の4.1 Hardware Setup を参照してください。

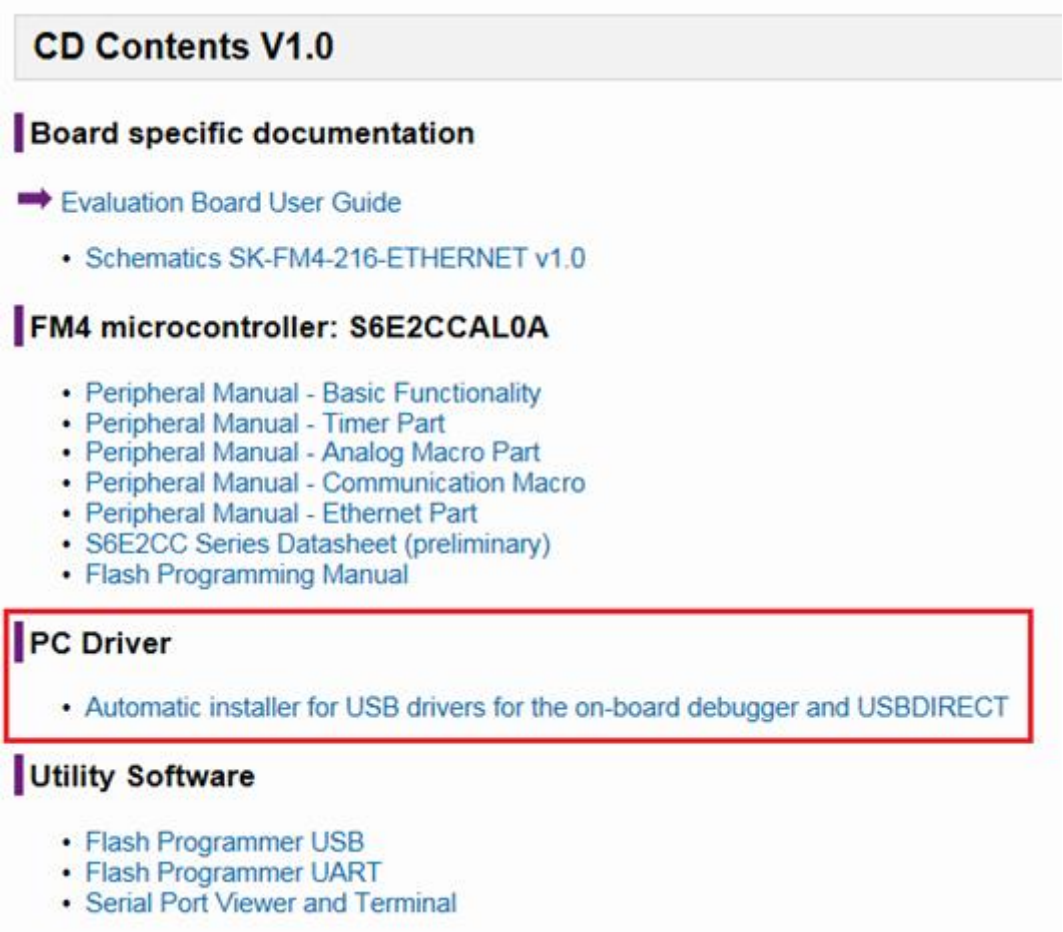
## 3.3 PCの準備

PCの推奨環境については、SK-FM4-216-ETHERNET Wi-Fi Sample Kit Setup Manual（英語版）の4.1.1 Operating Environment (for reference) を参照してください。用意したPCには、SK-FM4-216 ETHERNETとPC間のUSBケーブルを使用したシリアル通信のために専用ドライバをインストールしてください。また、サーバとSK-FM4-216-ETHERNET（クライアント）間で無線の暗号通信を行う場合には、PCにOpenSSLのインストールが必要になります。

### 3.3.1 USBドライバのインストール

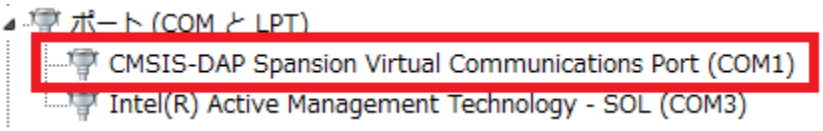
SK-FM4-216-ETHERNETに添付されているCD-ROMに入っているUSBドライバをPCにインストールしてください。インストールするUSBドライバはFigure 5 を参照してください。

Figure 5. インストールするUSBドライバ




インストール後にデバイスマネージャを起動しUSBドライバが正常にインストールされたか確認してください。割り当てられるポートはPCによって変わります。

Figure 6. インストール後の確認




### 3.3.2 暗号化通信の準備

暗号化通信を行う場合の手順を示します。

1. 「3.3 PCの準備」で用意したPCにSSLをインストールしてください。

   暗号化通信を行う場合は、サーバ用のPCに下記のリンクからOpenSSLをインストールしてください。

   http://slproweb.com/products/Win32OpenSSL.html

推奨バージョンは32ビット版v1.0.1JとV1.0.1Lです。推奨するインストールディレクトリは c:\OpenSSL- Win32です。

**注意事項**

- 弊社ではOpenSSLの品質を保証していません。
- 推奨バージョン以外のOpenSSLをインストールした場合や、推奨ディレクトリ以外にインストールした場合、OTAサーバソフトウェアの再ビルドが必要です。

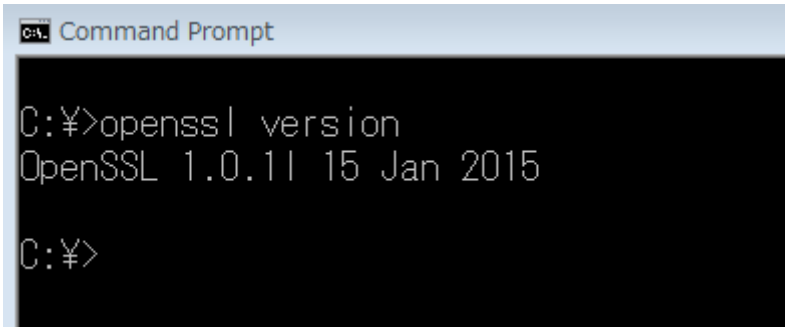
2. 環境変数“Path”の設定

   インストールが完了したら環境変数”Path”にc:\OpenSSL-Win32\binを追加してください。

3. インストール確認

   コマンドプロンプトでopenssl versionコマンドを実行しOpenSLLのバージョンが表示されればインストールは完了です。

Figure 7. OpenSSLバージョンの確認



4. サーバプログラムの再ビルド
   1. の＜注意事項＞で記載しているようにOpenSSLの推奨バージョンや推奨ディレクトリ以外にインストールした場合はサーバプログラムの再ビルドが必要です。サーバプログラムの作成手順について説明します。

   a. 「Visual Studio Express 2013 for Desktop」のインストール

   サーバプログラムを再ビルドするための推奨環境は「Visual Studio Express 2013 for Desktop」です。

   Visual Studio Communityシリーズでの動作は未確認です。

   b. プロジェクトファイルを開く

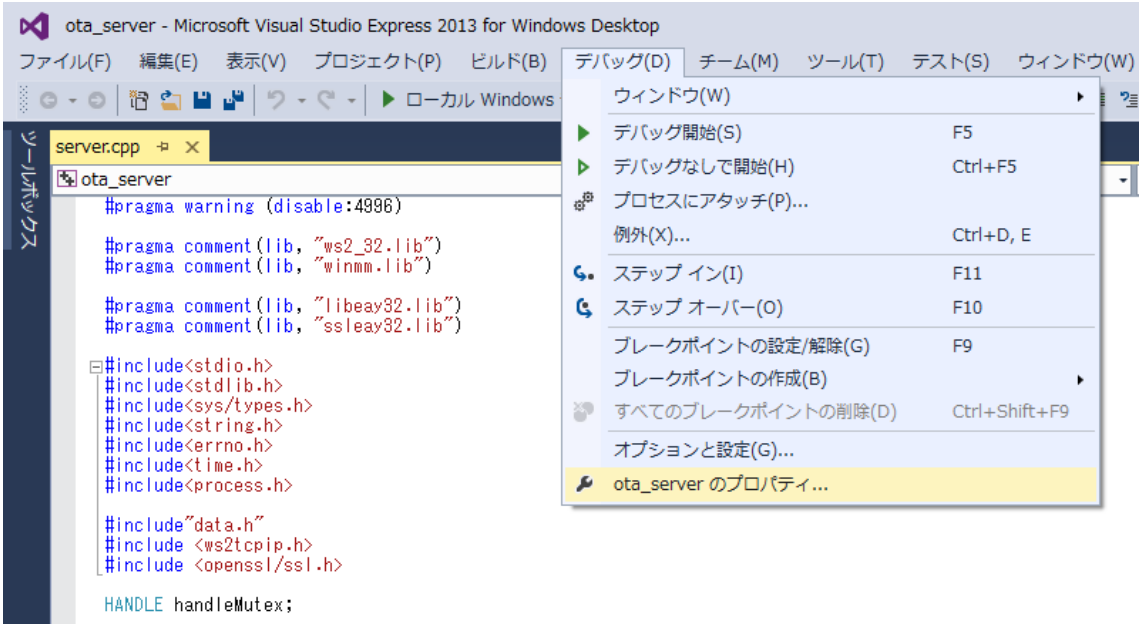
   次のプロジェクトファイルを開きます。

   “v02.5a_(SK-FM4-216-ETHERNET)\tools\ota_server_ssl_source\ota_server.sin”

   c. プロパティを変更します。

   メニューから“デバッグ”→“ota_serverのプロパティ”を選択します。

Figure 8. デバッグメニューのプロパティ

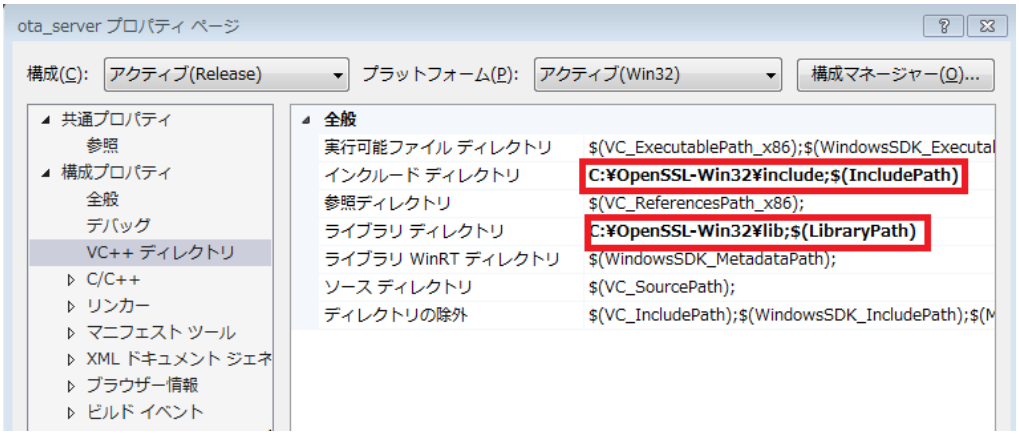



プロパティページが開いたらVC++ディレクトリを選択します。

インクルードディレクトリにはOpenSSLをインストールしたフォルダにあるincludeディレクトリを設定してください。

ライブラリディレクトリにはOpenSSLをインストールしたフォルダにあるlibディレクトリを設定してください。

Figure 9. プロパティの変更




d. 再ビルド

再ビルドしてください。

e. 実行ファイルの置き換え

再ビルドすると“02.5a_(SK-FM4-216-ETHERNET)\tools\ota_server_with_ssl_source\Release”に実行ファイル” ota_server.exe”が作成されます。この”. ota_server.exe”を“02.5a_(SK-FM4-216-ETHERNET)\tools\ota_server_ssl\tcpip”にコピーしてください。

## 3.4 ブートローダ

ブートローダの作成手順とフラッシュメモリへの書込み方法を説明します。

### 3.4.1 ブートローダの作成手順

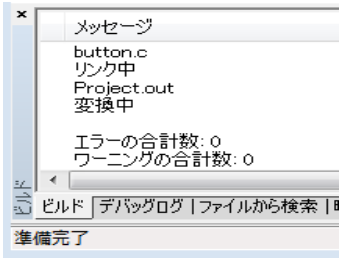
ブートローダのEWARMプロジェクトファイル (v02.5a_(SK-FM4-216- ETHERNET)\project\spansion_bootloader\EWARM-RELEASE\Project.eww) を開き、ツールバー上のmakeボタンをクリックし、プロジェクトをビルドします。

Figure 10. ブートローダのビルドの実施




ビルド出力情報を確認し、ビルドに成功したかをチェックします。

Figure 11. ブートローダのビルドの確認



### 3.4.2 ブートローダのフラッシュメモリへの書込み

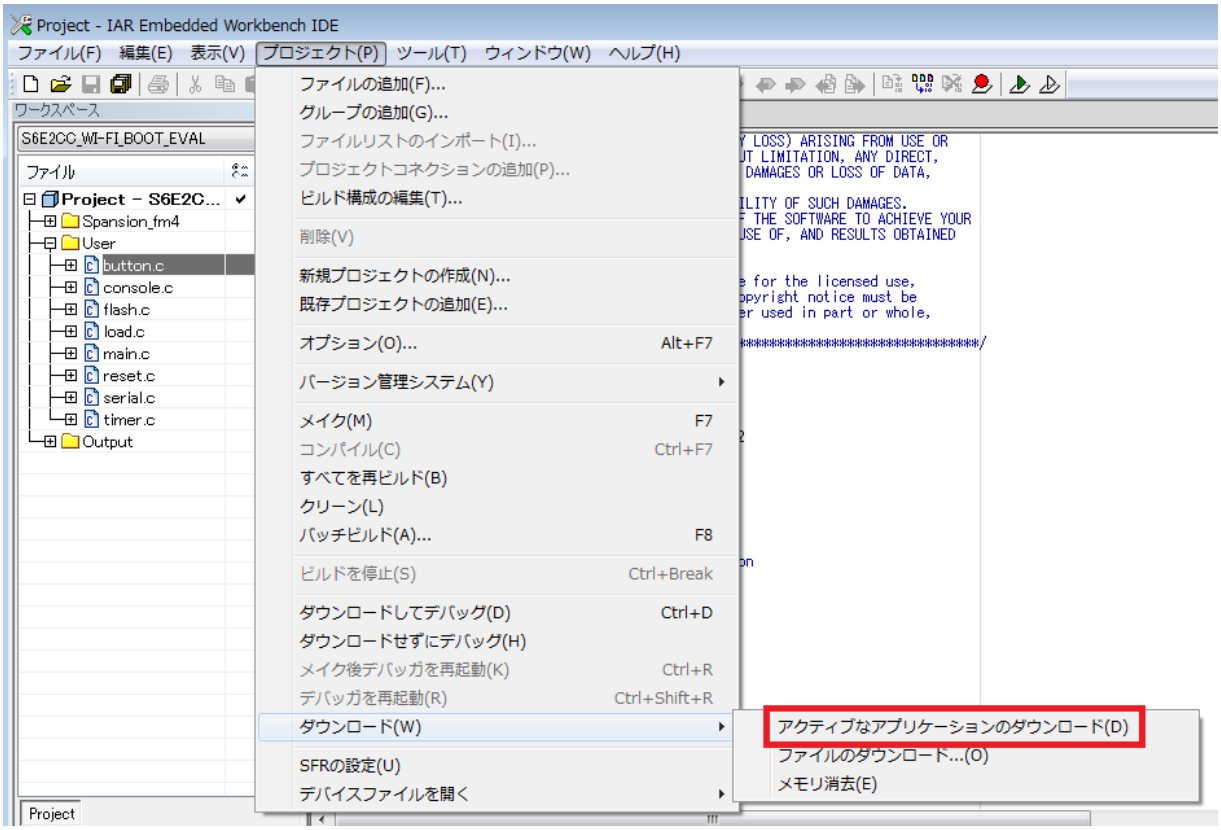
ブートローダをフラッシュメモリに書き込みます。

ICE※でPCをSK-FM4-216-ETHERNETのJTAGポートと接続してください。

ブートローダのEWARMプロジェクトのウィンドウで、“プロジェクト”→“ダウンロード”→“アクティブなアプリケーションのダウンロード”を実行し、ブートローダをSK-FM4-216-ETHERNETに書込みます。

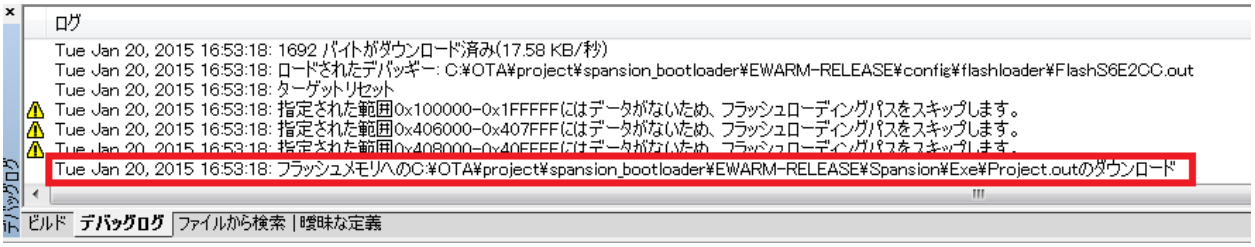
※ICEの種別設定は、“プロジェクト”→“オプション”でオプション画面を開き、“デバッガ”→“設定”→“ドライバ”の項で設定してください

Figure 12. ブートローダのフラッシュメモリへの書込み




デバッグのログ情報を確認し、フラッシュメモリへの書込みに成功したかをチェックします。

Figure 13. ブートローダのフラッシュメモリへの書込み結果の確認



## 3.5 OTA-Dライブラリ

### 3.5.1 OTA-D ライブラリの作成手順

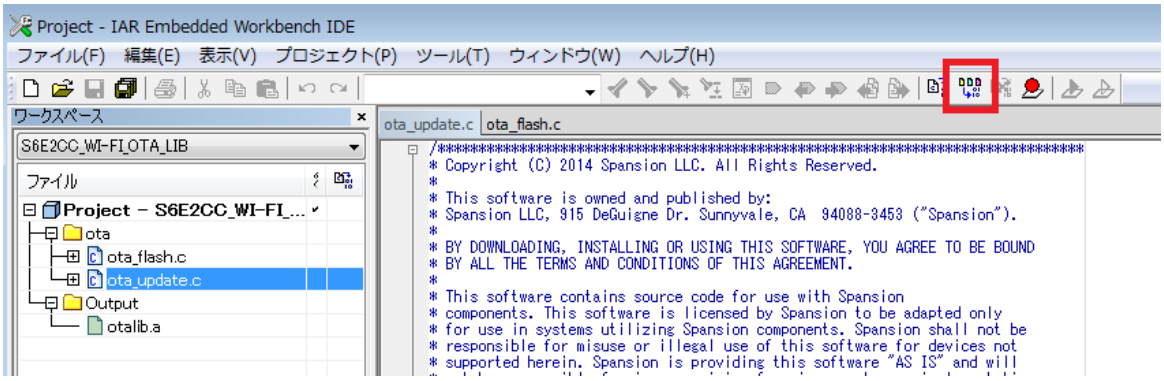
OTA-D ライブラリのEWARM プロジェクトファイル

(v02.5a_(SK-FM4-216-ETHERNET)¥project¥spansion_ota¥EWARM-RELEASE¥Project.eww)を開き、ツールバー上のmake ボタンをクリックしてプロジェクトをビルドするとOTA-D ライブラリファイル

v02.5a_(SK-FM4-216-ETHERNET)¥component¥common¥ota¥lib¥iar¥otalib.a が生成されます。

なお、このファイルはユーザソフトウェアにリンクされます。

Figure 14. OTA-Dライブラリのビルドの実施




## 3.6 ユーザソフトウェア

ユーザソフトウェアの作成手順とフラッシュメモリへの書込み方法を説明します。

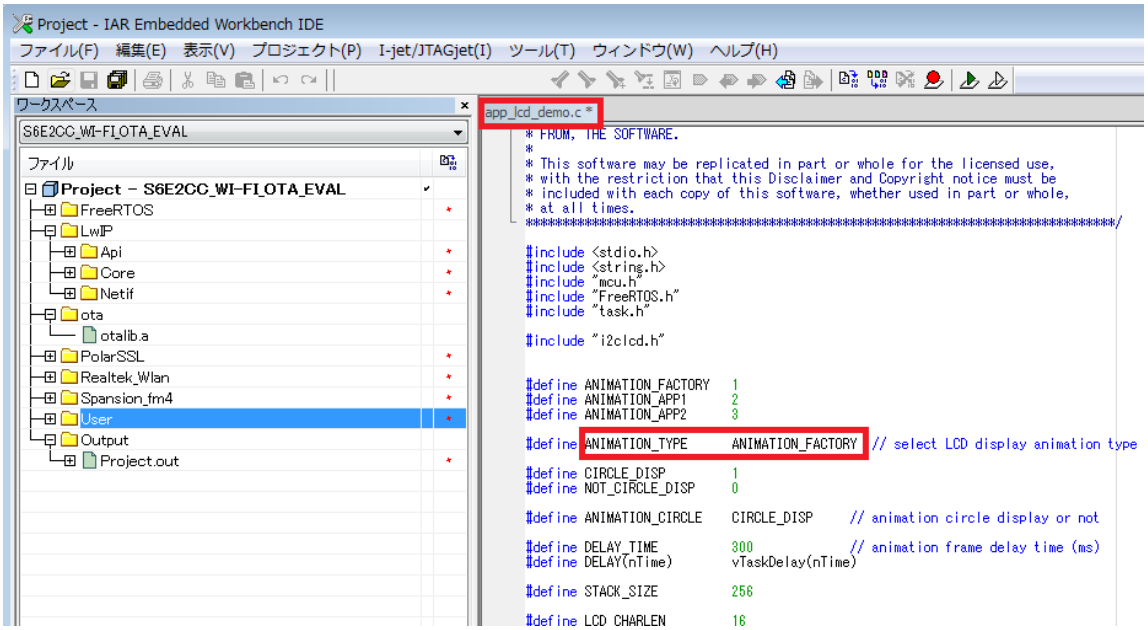
### 3.6.1 ユーザソフトウェアの作成手順

ユーザソフトウェアのEWARM プロジェクトファイル

(v02.5a_(SK-FM4-216-ETHERNET)¥project¥spansion_s6e2cx_8189em_with_ota¥EWARM-RELEASE¥Project.eww)を開きます。

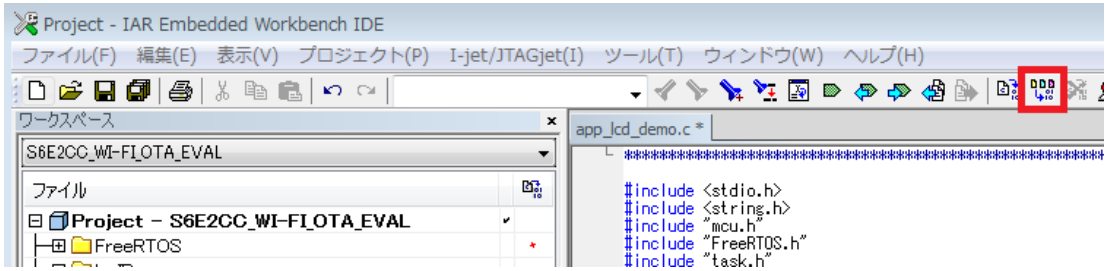
ソースファイル app_lcd_demo.c を編集し、ANIMATION_TYPE マクロをANIMATION_FACTORY に変更して保存します。

Figure 15. ユーザソフトウェアの作成（工場出荷版としてのHEXファイルの作成）



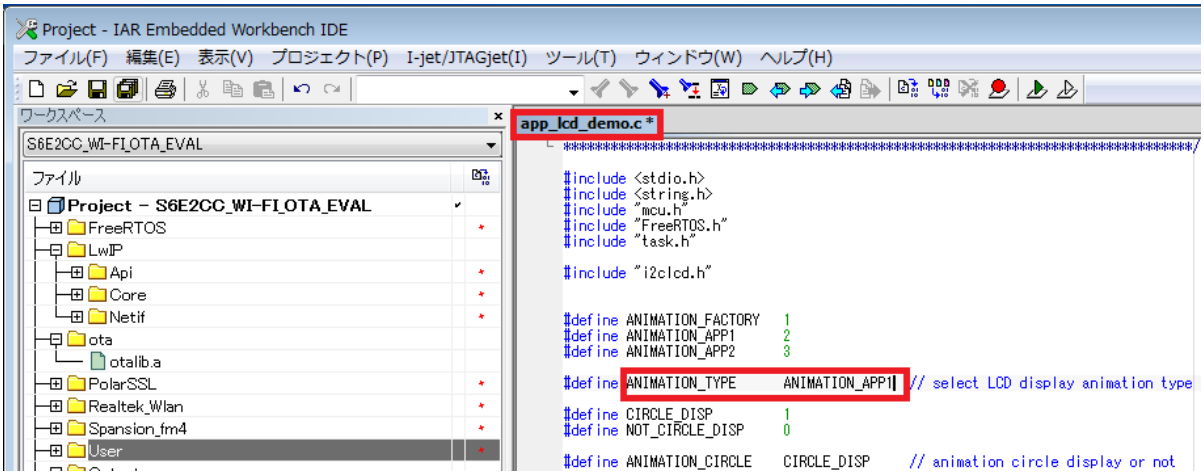

ツールバー上の make ボタンをクリックしてプロジェクトをビルドすると、hex 形式のユーザソフトウェア v02.5a_(SK-FM4-216-ETHERNET)¥project¥spansion_s6e2cx_8189em_with_ota¥EWARM-RELEASE¥Spansion¥Exe¥Project.hex が生成されます。このプログラムを製品出荷時のファイルとして、Project_factory_reset.hex にリネームします。このファイルは、シリアル通信によってブートローダに転送された後、ブートローダによってフラッシュメモリに書き込まれます。

Figure 16. ユーザソフトウェアのビルドの実施




次にソースファイルapp_lcd_demo.cを編集し、ANIMATION_TYPEマクロをANIMATION_APP1に変更して保存します。

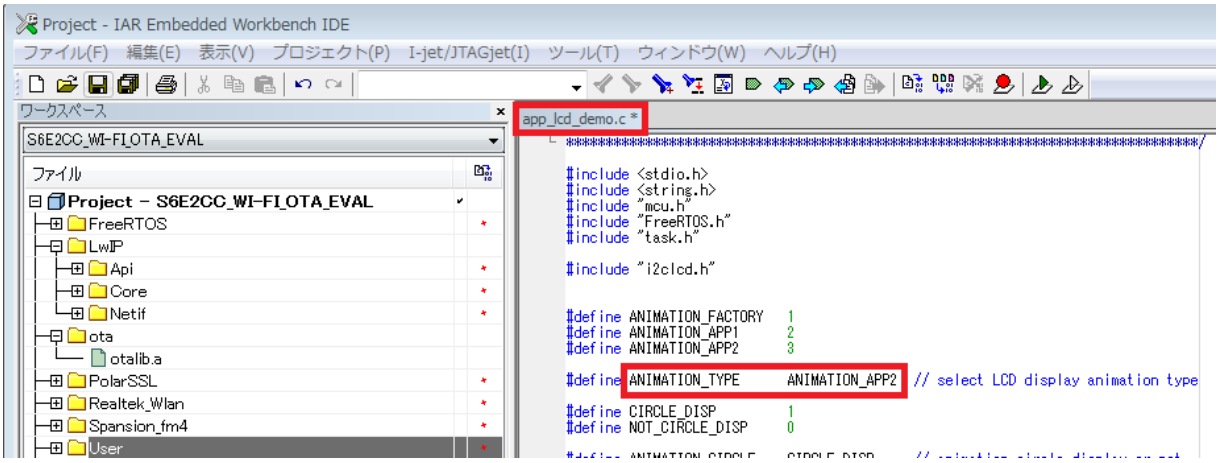
Figure 17. ユーザソフトウェアの作成（アプリケーション1としてのHEXファイルを作成）




プロジェクトをビルドすると、hex 形式のユーザソフトウェア (v02.5a_(SK-FM4-216-ETHERNET)¥project¥spansion_s6e2cx_8189em_with_ota¥EWARM-RELEASE¥Spansion¥Exe¥Project.hex) が生成されます。このプログラムをVERSION=1, LEVEL=1 のユーザソフトウェアとして、Project_app1.hex にリネームします。このファイルはTCPIP OTA サーバで使用します。

最後にソースファイルapp_lcd_demo.cを編集し、ANIMATION_TYPEマクロをANIMATION_APP2に変更して保存します。

Figure 18. ユーザソフトウェアの作成（アプリケーション2としてのHEXファイルを作成）




プロジェクトをビルドすると、hex 形式のユーザソフトウェア (v02.5a_(SK-FM4-216-ETHERNET)¥project¥spansion_s6e2cx_8189em_with_ota¥EWARM-RELEASE¥Spansion¥Exe¥Project.hex) が生成されます。このプログラムをVERSION=1, LEVEL=2 のユーザソフトウェアとして、Project_app2.hex にリネームします。このファイルはTCPIP OTA サーバで使用します。Project_app2.hex と Project_app1.hex はLCD の表示が違います。

### 3.6.2 ユーザソフトウェアのシリアル通信によるフラッシュメモリへの書込み

SK-FM4-216-ETHERNET のフラッシュメモリ上に利用可能なユーザソフトウェアがない場合、SERIAL OTA サーバの実行と同時にユーザソフトウェアをシリアル通信でダウンロードしフラッシュメモリに書き込みます。

既にフラッシュメモリに書き込まれているユーザソフトウェアをシリアル通信にてアップデートする場合は、下記の手順に従ってSK-FM4-216-ETHERNETを操作してください。

ステップ1：ユーザボタン (SW4) を押したままにします。

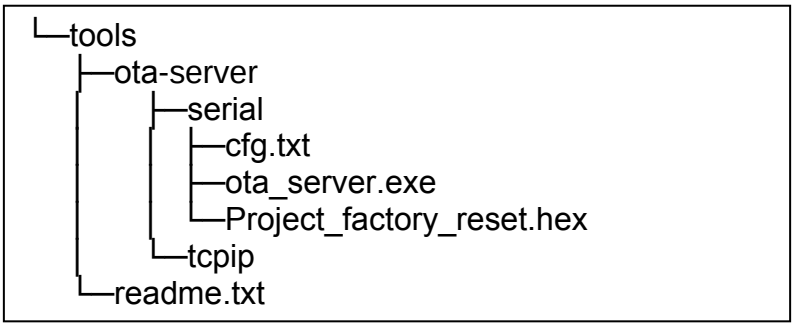
ステップ2：リセットボタン (RST) をクリックし、SK-FM4-216-ETHERNETをリセットします。

ステップ3：ユーザボタン (SW4) を5秒以上押し続けます。

上記の操作が完了すると、ブートローダはシリアル通信によりユーザソフトウェアのダウンロードが可能となります。

シリアルでダウンロードするには、USB ケーブルでPC とSK-FM4-216-ETHERNET のUSB ポートを接続してください。ユーザソフトウェア(Project_factory_reset.hex を例にします)をSERIAL OTA サーバプログラムの所属フォルダv02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥serial 配下にコピーしてください。

Figure 19. SERIAL OTAサーバのディレクトリ構成

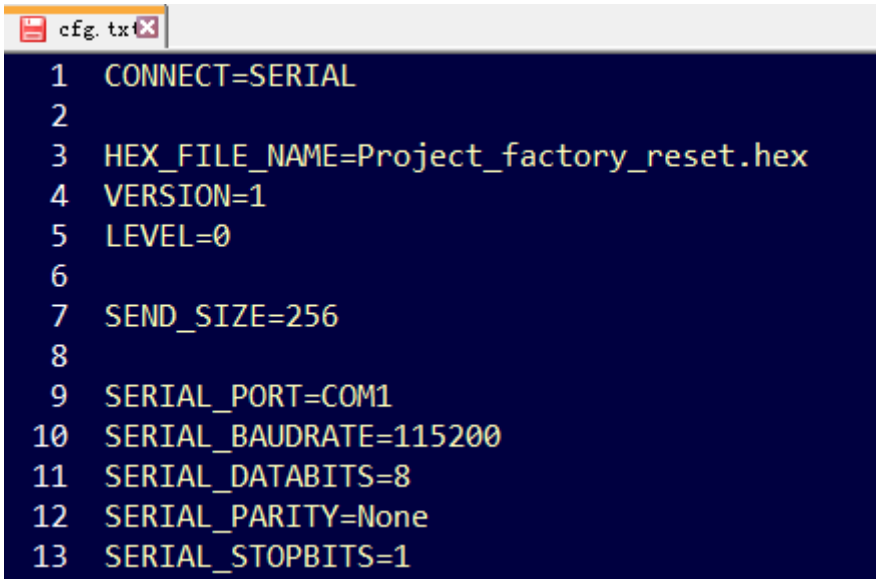



**注意事項**

- SERIAL OTAサーバプログラムはインテルHEX形式以外のファイルを解析できません。

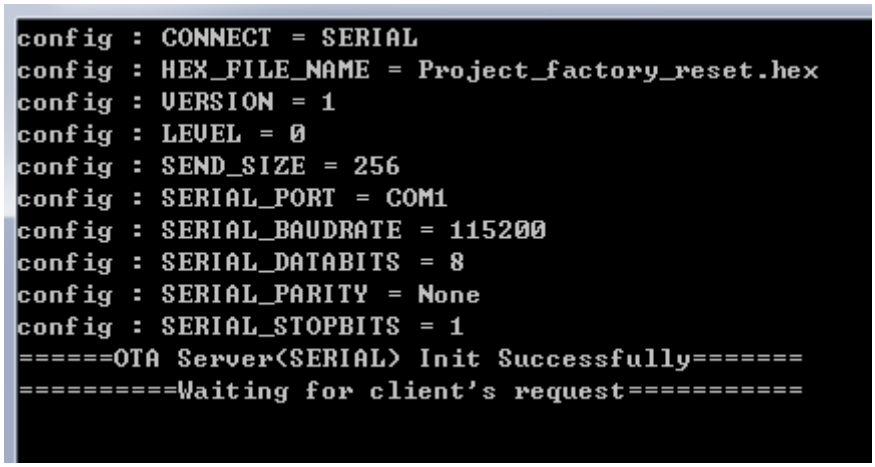
SERIAL OTA サーバのコンフィグレーションファイル v02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥serial¥cfg.txt をFigure 20 ように設定して保存します（設定ファイル中の設定項目の説明はv02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥readme.txt を参照してください）。

Figure 20. SERIAL OTAサーバ用コンフィグレーションファイルの内容




v02.5a_(SK-FM4-216-ETHERNET)\tools\ota_server\serial\ota_server.exeをダブルクリックして、SERIAL OTAサーバを起動し、正常に実行されることを確認してください。

Figure 21. SERIAL OTAサーバの起動画面



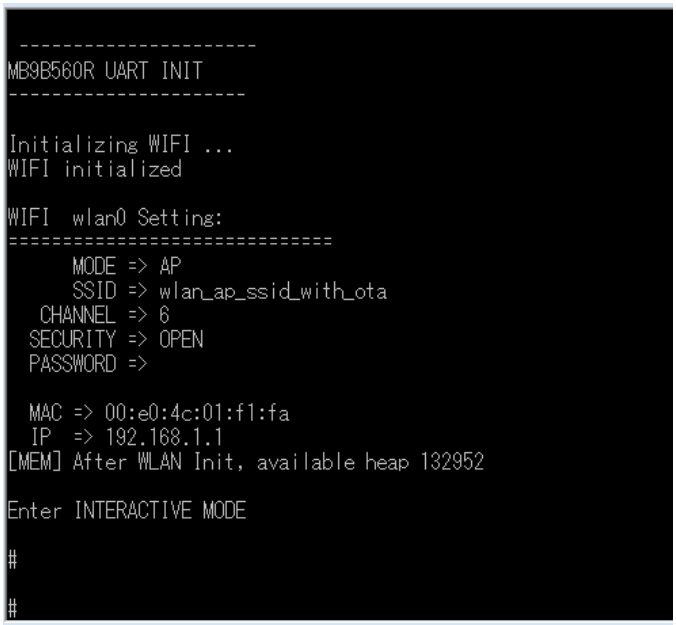
シリアル通信でのフラッシュメモリの書込み動作はFigure 22 に示します。

Figure 22. SERIAL OTAサーバの実行画面




フラッシュメモリへの書込みが完了するとFigure 23 の様にWi-Fiの情報が表示されます。

Figure 23. SERIAL OTAサーバでのフラッシュメモリ書込み終了後のWi-Fi情報の表示

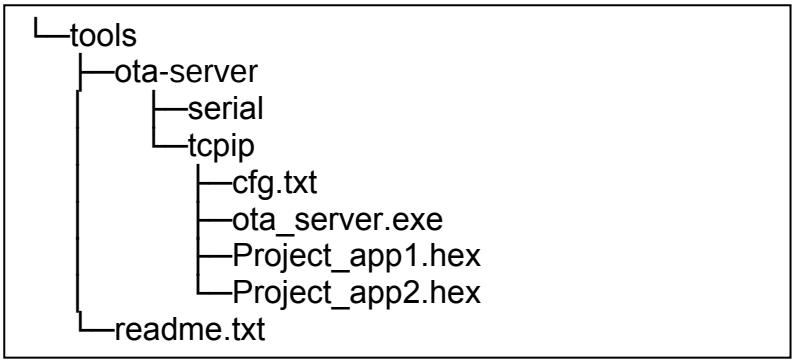


# 4 OTA-Dリファレンスソフトウェアの実行

本章では OTA-D リファレンスソフトウェアの実行手順について説明します。

1. PC の無線LAN を「wlan_ap_ssid_with_ota」無線ネットワークに接続し、PC の無線LAN のIP アドレスを192.168.1.4 に設定します。ここではIP アドレスが192.168.1.4 である場合を例として説明します。固定アドレスであれば、他のアドレスでも使用可能です。
2. 「3.6.1 ユーザソフトウェアの作成手順」で生成したProject_app1.hex とProject_app2.hex を TCPIP OTA サーバプログラムの所属フォルダ v02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥tcpip にコピーしてください。

Figure 24. TCPIP OTAサーバのディレクトリ構成



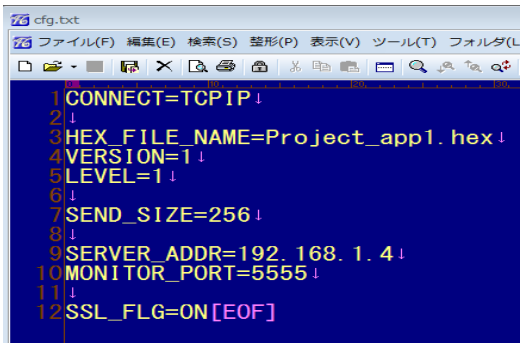

**注意事項**

TCPIP OTAサーバプログラムはインテルHEX形式以外のファイルを解析できません。

3. Figure 25 を参考にしてTCPIP OTA サーバの設定ファイル v02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥tcpip¥cfg.txt 編集して保存します（設定ファイル中の設定項目の説明はv02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥readme.txt を参照してください）。SSL を使用しない場合はSSL_FLG=OFF に書き換えてください。

Figure 25. TCPIP OTAサーバ用コンフィグレーションファイルの内容

(Project_app1.hexをフラッシュメモリに書き込む場合)

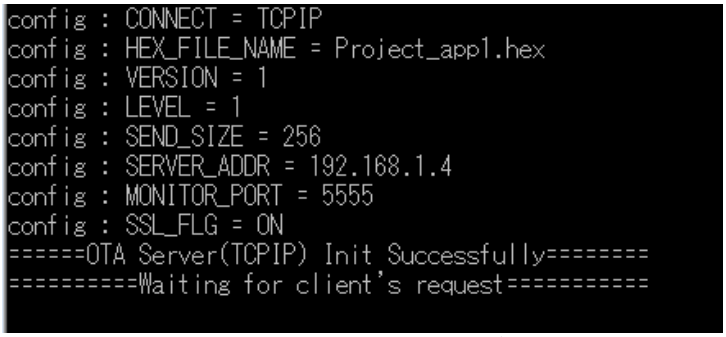


4. v02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥tcpip¥ota_server.exe をダブルクリックして TCPIP OTA サーバを起動し、正常に起動されたかを確認してください。

**注意事項**

- プログラムがWindowsのファイアウォールに遮断されないように、ファイアウォールの設定を一時的に変更してください。

Figure 26. TCPIP OTAサーバの起動画面



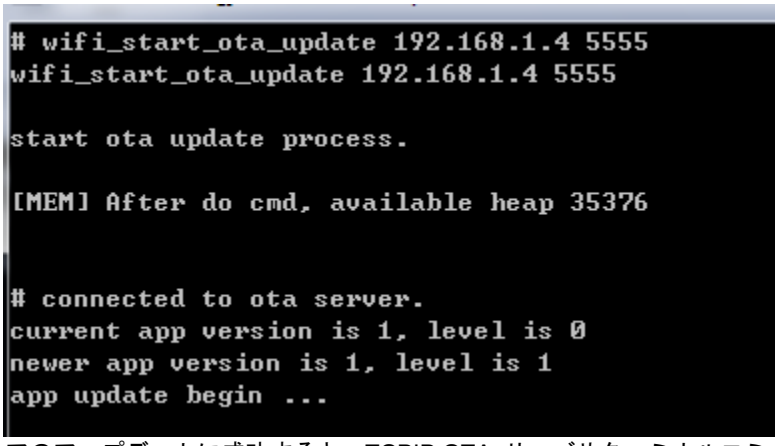

5. ユーザソフトウェアが起動するとシリアル通信によってユーザソフトウェアとの対話が可能になります。インタフェースには「3.6.2 ユーザソフトウェアのシリアル通信によるフラッシュメモリへの書込み」で説明したSERIAL OTA サーバやTeraterm などのターミナルエミュレータが使用できます。上記インタフェース上で「help」を入力すると、コマンド一覧を参照できます。一覧にある「wifi_start_ota_update」コマンドはOTA-D 機能のコマンドです。

Figure 27. Wi-Fiのコマンド一覧



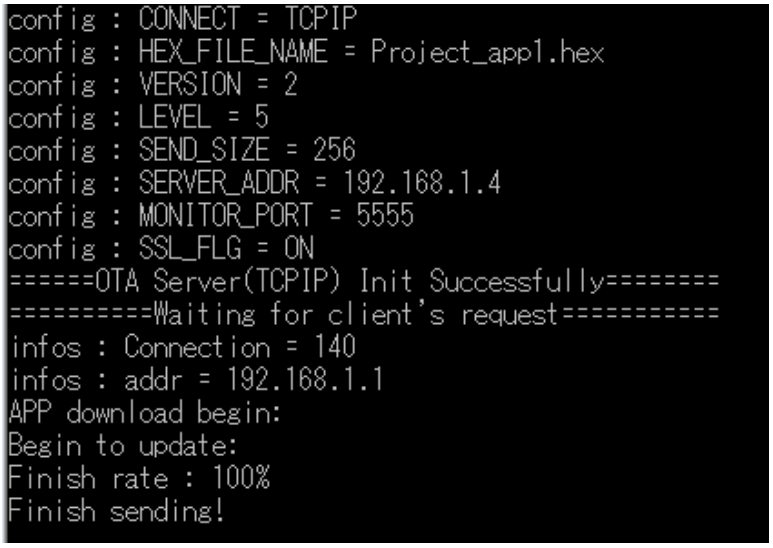
6. 「wifi_start_ota_update 192.168.1.4 5555 ssl」コマンドでユーザソフトウェアのアップデートを開始します。「192.168.1.4」はTCPIP OTA サーバのIP アドレスです。「5555」はTCPIP OTA サーバのポート番号です。SSL を使用する場合はポート番号の後ろに空白を入れてから小文字でssl と入力してください。SSL を使用しない場合はIP アドレスとポート番号だけ指定してください。IP アドレスとポート番号はTCPIP OTA サーバの設定ファイル中の設定値と一致させてください。

Figure 28. TCPIP OTA-Dのコマンド画面




ユーザソフトウェアのアップデートに成功すると、TCPIP OTA サーバやターミナルエミュレータの実行画面でFinish sending!と表示されればフラッシュメモリへの書込みが完了しました。

Figure 29. TCPIP OTAサーバの実行画面

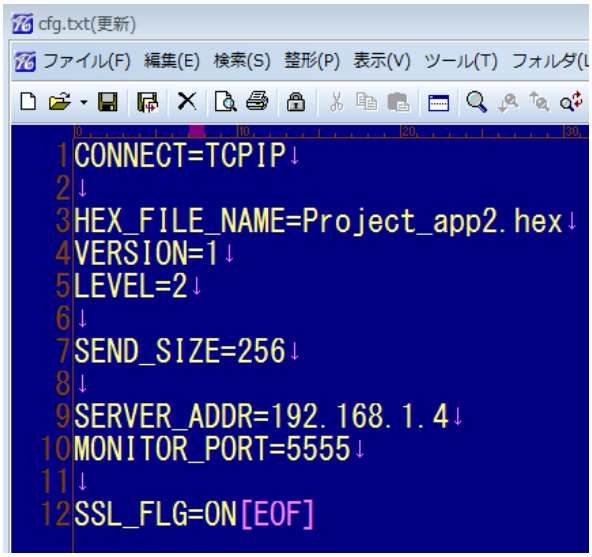


7. フラッシュメモリに新しく書き込むユーザソフトウェアをTCPIP OTA サーバ側のアップデート対象ファイルとする方法を説明します。まず、新しく書き込むユーザソフトウェアのhex ファイル（例えば前述のProject_app2.hex）をOTA サーバプログラムの所在フォルダv02.5a_(SK-FM4-216-ETHERNET)¥tools¥ota_server¥tcpip にコピーします。次に、OTA サーバ設定ファイルcfg.txt 中の「HEX_FILE_NAME」，「VERSION」と「LEVEL」を変更して設定ファイルを保存します。最後に、TCPIP OTA サーバを再起動してください。

**注意事項**

- *TCPIP OTA* サーバ設定ファイル*cfg.txt* を編集したあと、新規設定を有効にするには、*TCPIP OTA* サーバ*ota_server.exe* を再起動する必要があります。

Figure 30. TCPIP OTAサーバ用コンフィグレーションファイルの内容




# 5 OTA-D応用例

本製品の応用例は「S6E2CxシリーズOver The Air update with embedded Dual Flash Memory応用編（仮称）」にて提供する予定です。

# 6 Document History

Document Title: AN203980 - FM4 S6E2Cx シリーズ Over the Air Update 32-Bit Microcontroller with Embedded Dual Flash

Document Number: 002-03981

| Revision | ECN | Orig. of Change | Submission Date | Description of Change |
|---|---|---|---|---|
| ** | – | SXKA | 06/17/2015 | Initial release |
| *A | 5151153 | SXKA | 03/03/2016 | Migrated Spansion Application Note from S6E2Cx_AN709-00017-1v0-J to Cypress format |

## Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

## Products

| | |
|---|---|
| ARM® Cortex® Microcontrollers | cypress.com/arm |
| Automotive | cypress.com/automotive |
| Clocks & Buffers | cypress.com/clocks |
| Interface | cypress.com/interface |
| Lighting & Power Control | cypress.com/powerpsoc |
| Memory | cypress.com/memory |
| PSoC | cypress.com/psoc |
| Touch Sensing | cypress.com/touch |
| USB Controllers | cypress.com/usb |
| Wireless/RF | cypress.com/wireless |

## PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

## Cypress Developer Community

Community | Forums | Blogs | Video | Training

## Technical Support

cypress.com/go/support

PSoC is a registered trademark and PSoC Creator is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com

© Cypress Semiconductor Corporation, 2015-2016. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you under its copyright rights in the Software, a personal, non-exclusive, nontransferable license (without the right to sublicense) (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units. Cypress also grants you a personal, non-exclusive, nontransferable, license (without the right to sublicense) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely to the minimum extent that is necessary for you to exercise your rights under the copyright license granted in the previous sentence. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and Company shall and hereby does release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. Company shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.