



## CY15FRAMKIT-001

# シリアル F-RAM™ 開発キット ガイド

文書番号: 001-97580 Rev.\*\*

Cypress Semiconductor  
198 Champion Court  
San Jose, CA 95134 USA  
Phone (USA): +1.800.858.1810  
Phone (Intl): +1.408.943.2600  
[www.cypress.com](http://www.cypress.com)

## 著作権

© Cypress Semiconductor Corporation, 2015. 本文書に記載される情報は予告なく変更される場合があります。Cypress Semiconductor Corporation (サイプレス セミコンダクタ社) は、サイプレス製品に組み込まれた回路以外のいかなる回路を使用することに対して一切の責任を負いません。サイプレス セミコンダクタ社は、特許またはその他の権利に基づくライセンスを譲渡することも、または含意することはありません。サイプレス製品は、サイプレスとの書面による合意に基づくものでない限り、医療、生命維持、救命、重要な管理、または安全の用途のために使用することを保証するものではなく、また使用することを意図したものではありません。さらにサイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネントとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

## 商標

PSoC はサイプレス セミコンダクタ社の登録商標であり、PSoC Creator および F-RAM は同社の商標です。本書で言及するその他全ての商標または登録商標は、それぞれの所有者に帰属します。

## ソースコード

すべてのソースコード (ソフトウェアおよび/またはファームウェア) はサイプレス セミコンダクタ社 (以下「サイプレス」) が所有し、全世界の特許権保護 (米国およびその他の国)、米国の著作権法ならびに国際協定の条項により保護され、かつそれらに従います。サイプレスが本書面によりライセンシーに付与するライセンスは、個人的、非独占的かつ譲渡不能のライセンスであり、適用される契約で指定されたサイプレスの集積回路と併用されるライセンシーの製品のみをサポートするカスタム ソフトウェアおよび/またはカスタム ファームウェアを作成する目的に限って、サイプレスのソースコードの派生著作物をコピー、使用、変更そして作成するためのライセンス、ならびにサイプレスのソースコードおよび派生著作物をコンパイルするためのライセンスです。上記で指定された場合を除き、サイプレスの書面による明示的な許可なくして本ソースコードを複製、変更、変換、コンパイル、または表示することはすべて禁止します。

## 免責条項

サイプレスは、明示的または黙示的を問わず、本資料に関するいかなる種類の保証も行いません。これには、商品性または特定目的への適合性の黙示的な保証が含まれますが、これに限定されません。サイプレスは、本文書に記載される資料に対して今後予告なく変更を加える権利を留保します。サイプレスは、本文書に記載されるいかなる製品または回路を適用または使用したことによって生ずるいかなる責任も負いません。サイプレスは、誤動作や故障によって使用者に重大な傷害をもたらすことが合理的に予想される生命維持システムの重要なコンポーネントとしてサイプレス製品を使用することを許可していません。生命維持システムの用途にサイプレス製品を供することは、製造者がそのような使用におけるあらゆるリスクを負うことを意味し、その結果サイプレスはあらゆる責任を免除されることを意味します。

ソフトウェアの使用は、適用されるサイプレス ソフトウェア ライセンス契約によって制限され、かつ制約される場合があります。

# 目次



<b>安全情報</b> .....	<b>5</b>
法規制の順守 .....	5
一般的な安全上の注意 .....	6
<b>1. はじめに</b> .....	<b>7</b>
1.1 キットの内容 .....	7
1.2 PSoC Creator™ .....	8
1.3 はじめよう .....	8
1.4 追加の学習資料 .....	8
1.5 テクニカル サポート .....	8
1.6 本書の表記法 .....	8
<b>2. キットのインストール</b> .....	<b>9</b>
2.1 必須なソフトウェア .....	9
2.2 キットソフトウェアのインストール .....	9
2.3 ソフトウェアのアンインストール .....	11
<b>3. キットの概要</b> .....	<b>12</b>
3.1 CY15FRAMKIT-001 シリアル F-RAM 開発キット の概要 .....	12
3.2 キットの動作およびコンフィギュレーション ガイド .....	13
3.2.1 電源供給ジャンパ .....	13
3.2.2 F-RAM デバイス .....	13
3.2.3 I <sup>2</sup> C F-RAM デバイスのスレーブ アドレス選択用の DIP スイッチ .....	13
3.2.4 CY8CKIT-042/Arduino UNO R3 基板へのコネクタ .....	14
3.2.5 デバッグ ヘッダ .....	15
3.2.6 テスト ポイント .....	15
3.3 PSoC 4 Pioneer Kit に接続した CY15FRAMKIT-001 .....	15
3.4 Arduino UNO R3 キットに接続した CY15FRAMKIT-001 .....	16
<b>4. ハードウェア</b> .....	<b>17</b>
4.1 基板の詳細 .....	17
4.2 動作理論 .....	19
4.3 機能の説明 .....	19
4.3.1 SPI F-RAM デバイス (256K ビットの FM25W256) .....	19
4.3.2 I <sup>2</sup> C F-RAM デバイス (256K ビットの FM24W256) .....	22
<b>5. サンプル プロジェクト</b> .....	<b>25</b>

5.1	PSoC 4 Pioneer Kit のプログラミング .....	25
5.2	UART の設定 .....	28
5.2.1	ハイパーターミナルの設定 .....	30
5.2.2	PuTTY の設定 .....	32
5.3	プロジェクト: PSoC 4 F-RAM SPI .....	34
5.3.1	プロジェクトの説明 .....	34
5.3.2	ハードウェア接続 .....	34
5.3.3	ファームウェア フロー .....	35
5.3.4	出力の検証 .....	35
5.3.5	プロジェクトの変更 .....	36
5.3.6	API .....	38
5.4	プロジェクト: PSoC 4 F-RAM I <sup>2</sup> C .....	39
5.4.1	プロジェクトの説明 .....	39
5.4.2	ハードウェア接続 .....	39
5.4.3	ファームウェア フロー .....	40
5.4.4	出力の検証 .....	40
5.4.5	プロジェクトの変更 .....	41
5.4.6	API .....	42
5.5	プログラミング Arduino UNO キット .....	44
5.6	プロジェクト: Arduino F-RAM SPI .....	45
5.6.1	プロジェクトの説明 .....	45
5.6.2	ハードウェア接続 .....	45
5.6.3	ファームウェア フロー .....	46
5.6.4	出力の検証 .....	46
5.6.5	API .....	47
5.7	プロジェクト: Arduino F-RAM I <sup>2</sup> C .....	48
5.7.1	プロジェクトの説明 .....	48
5.7.2	ハードウェア接続 .....	48
5.7.3	ファームウェア フロー .....	49
5.7.4	出力の検証 .....	49
5.7.5	API .....	49
<b>付録 .....</b>		<b>50</b>
A.1	CY15FRAMKIT-001 回路図 .....	50
A.2	ピン割り当ての表 .....	53
A.3	デバッグ ヘッダ I/O .....	54
A.4	0Ω 抵抗と負荷なしの使用法 .....	55
A.5	30Ω 抵抗と負荷なしの使用法 .....	55
A.6	部品表 (BOM) .....	56

# 安全情報



## 法規制の順守

CY15FRAMKIT-001 シリアル F-RAM™開発キットは、実験室環境でハードウェアまたはソフトウェア用のシリアル メモリ開発プラットフォームとして使用されます。このボードは、保護カバーを含まないオープンシステム設計です。そのため、この基板は近くにある他の電子・電気機器と干渉する可能性があります。

この基板に追加の配線を取り付けるか、工場出荷時の設定から製品の動作を変更すると、性能に影響を及ぼし、すぐ近くにある他の装置に干渉する可能性があります。そのような干渉が検出された場合、適切な解決策を実行すべきです。

工場出荷時の CY15FRAMKIT-001 は、クラス A の製品として CE の要件を満たすことが確認されています。



CY15FRAMKIT-001 は ESD に弱いデバイスを含んでいます。静電気は人体やあらゆる装置に簡単に蓄積し、検出されず放電することがあります。高エネルギー放電にさらされたデバイスが完全に損傷する場合があります。適切な ESD 保護が、性能低下または機能喪失を防止するために推奨されます。CY15FRAMKIT-001 基板は、使用しない時には出荷時に使われていた保護パッケージに格納してください。



### 寿命／製品リサイクル

このキットの製品寿命は、箱の裏側に記載される製造日から 5 年です。キットの廃棄については、最寄りのリサイクル業者にお問い合わせください。

## 一般的な安全上の注意

### ESD 保護

ESD は基板および関連のコンポーネントを破損する可能性があります。サイプレスは、ユーザーが ESD ワークステーションでのみ ESD テスト手順を実行することを推奨します。ESD ワークステーションがない場合、部品を取り扱う時には、基板のシャーシグラウンド (あらゆる塗装されていない金属面) に接続されている静電気除去リストストラップを身に着けて、適切な ESD 保護を行ってください。

### 基板の取り扱い

CY15FRAMKIT-001 基板は ESD に弱いです。基板を持つ時は基板の縁だけを持ってください。箱から基板を取り出したら、接地され静電気のない面に置いてください。可能であれば導電性フォームパッドを使用してください。どんな表面であっても、基板を滑らせないでください。

# 1. はじめに



CY15FRAMKIT-001 シリアルF-RAM開発キット (DVK) にご興味をお持ちいただきありがとうございます。キット (シールド) は使いやすい、且つ安価な開発キットとして設計され、サイプレスのシリアル強誘電体ランダム アクセス メモリ (F-RAM) の機能を示します。柔軟性を持つように設計されたため、このキットは、CY8CKIT-042 PSoC® 4 Pioneer Kit、CY8CKIT-040、CY8CKIT-042 BLE、サードパーティ製のArduino™ UNO R3、および他のいくつかのArduinoキットとのフットプリント互換性を持っています。この基板は、256KビットのSPI F-RAM、256KビットのI<sup>2</sup>C F-RAM、I<sup>2</sup>C F-RAMデバイス選択ピンを制御するためのスリーポール DIPスイッチ、Arduino UNO互換ヘッダ、および2個のデバッグ ヘッダを含んでいます。このキットは5.0Vまたは3.3Vの電源電圧に対応します。

CY15FRAMKIT-001はサイプレスのシリアルF-RAMメモリ用の開発キットです。サイプレスのF-RAMは強誘電体技術に基づいて構築され、高性能RAMに不揮発性データ ストレージを組み合わせたメモリです。シリアルF-RAMはフル スピードのバスで高速書き込みを実現します。書き込み遅延がなくデータが瞬時に不揮発性になります。これは、アクティブ電流が300μA、スタンバイ電流が6μAという低消費電力なメモリです。書き込み速度が高速のため、アクティブになる短い時間のみで非常に低いエネルギーを消費します。事実上100兆回の読み出し／書き込みサイクルという無限の耐久性を提供しています。

## 1.1 キットの内容

CY15FRAMKIT-001 シリアル F-RAM DVK は 11-1 に示しているように、以下のものを含んでいます。

- CY15FRAMKIT-001 シリアル F-RAM DVK 基板
- クイックスタート ガイド

11-1. CY15FRAMKIT-001 シリアル F-RAM DVK の内容物



詳細については、[www.cypress.com/shop](http://www.cypress.com/shop) をご覧ください。キットの内容を確認し、部品が欠落している場合、最寄りのサイプレス販売代理店にお問い合わせください。

## 1.2 PSoC Creator™

PSoC Creator は最先端の、使いやすい統合設計環境 (IDE) です。これは、事前検証および事前に特性確認された PSoC Component ™ のライブラリに基づく、革命的なハードウェアとソフトウェアの協調設計を紹介します。

PSoC Creator により、以下のことが可能です。

- PSoC コンポーネントをドラッグ アンド ドロップして、カスタム設計の回路図を作成する
- コンポーネントを自動的に配置して配線し、GPIO を設定する、
- 備えているコンポーネント API でファームウェアを開発しデバッグする

また、PSoC Creator により、ユーザーは、PSoC デバイス用の結合コンパイラ チェーンやプロダクション プログラマを含む全体ツール体系を活用できます。詳細については、[www.cypress.com/Creator](http://www.cypress.com/Creator) をご覧ください。

## 1.3 はじめよう

このガイドはユーザーがシリアルF-RAM DVKに慣れるのに役立ちます。「キットのインストール」章はキット ソフトウェアのインストールについて述べます。「キットの概要」章はキットの機能について説明します。「ハードウェア」章はハードウェア動作を詳しく述べます。「サンプル プロジェクト」章はサンプル コードを記述します。付録の節では、回路図、ピンの割当て、0オームと30オーム抵抗の使用、および部品表 (BOM) を述べます。

## 1.4 追加の学習資料

データシートとアプリケーション ノートの追加学習資料は、[www.cypress.com/go/F-RAM](http://www.cypress.com/go/F-RAM) をご覧ください。ベースボードに関連するリソースは以下から入手できます。

- PSoC 4 Pioneer Kit (CY8CKIT-042)
- Arduino UNO R3

## 1.5 テクニカル サポート

支援が必要な場合、[www.cypress.com/support](http://www.cypress.com/support) のサポート ウェブページをご覧ください。+1(800)541-4736 内線 2 (米国) あるいは +1 (408) 943-2600 内線 2 (米国外) にお電話の上、カスタマ サポートにお問い合わせください。

## 1.6 本書の表記法

表 1-1. 本書の表記法

表記法	使用法
Courier New フォント	ファイルの場所、ユーザーが入力したテキストおよびソースコードを示す: C:\...cd\icc\
イタリック フォント	ファイル名および参考文献を示す: <i>PSoC Designer User Guide</i> にある <i>sourcefile.hex</i> ファイルを参照してください。
[角括弧、太字]	操作手順でキーボードに入力するコマンドを示す: [Enter] または [Ctrl] [C]
File > Open	メニュー パスを示す: File > Open > New Project
太字	操作手順でのコマンド、メニューパス、アイコン名を示す: <b>File</b> アイコンをクリックしてから、 <b>Open</b> をクリックします。
Times New Roman フォント	式を示す: $2 + 2 = 4$
灰色のボックス内のテキスト	製品の注意点や製品固有の機能を説明



## 2. キットのインストール



本節では CY15FRAMKIT-001 シリアル F-RAM DVK ソフトウェアのインストールおよび前提条件について述べます。

### 2.1 必須なソフトウェア

キットを使用する前に、PSoC Creator IDE または Arduino IDE をインストールする必要があります。サイプレスの全てのソフトウェアのインストールは管理者特権が要求されます。インストール後のソフトウェア実行は管理者特権を必要としません。サイプレスが提供するソフトウェアが実行中であれば、キット ソフトウェアをインストールする前に全て終了してください。

PSoC Creator IDE をダウンロードするには、[www.cypress.com/psoccreator](http://www.cypress.com/psoccreator) にアクセスしてください。

Arduino IDE をダウンロードするには、[arduino.cc/en/Main/Software](http://arduino.cc/en/Main/Software) にアクセスしてください。

### 2.2 キットソフトウェアのインストール

CY15FRAMKIT-001 シリアル F-RAM DVK ソフトウェアをインストールするには以下の手順に従ってください。

1. [www.cypress.com/go/CY15FRAMKIT-001](http://www.cypress.com/go/CY15FRAMKIT-001) から CY15FRAMKIT-001 DVK ソフトウェアをダウンロードします。CY15FRAMKIT-001 ソフトウェアが 3 つの異なるフォーマットで提供されます (図 2-1 を参照):

図 2-1. ダウンロード時のシリアル F-RAM DVK ソフトウェアの可能なフォーマット

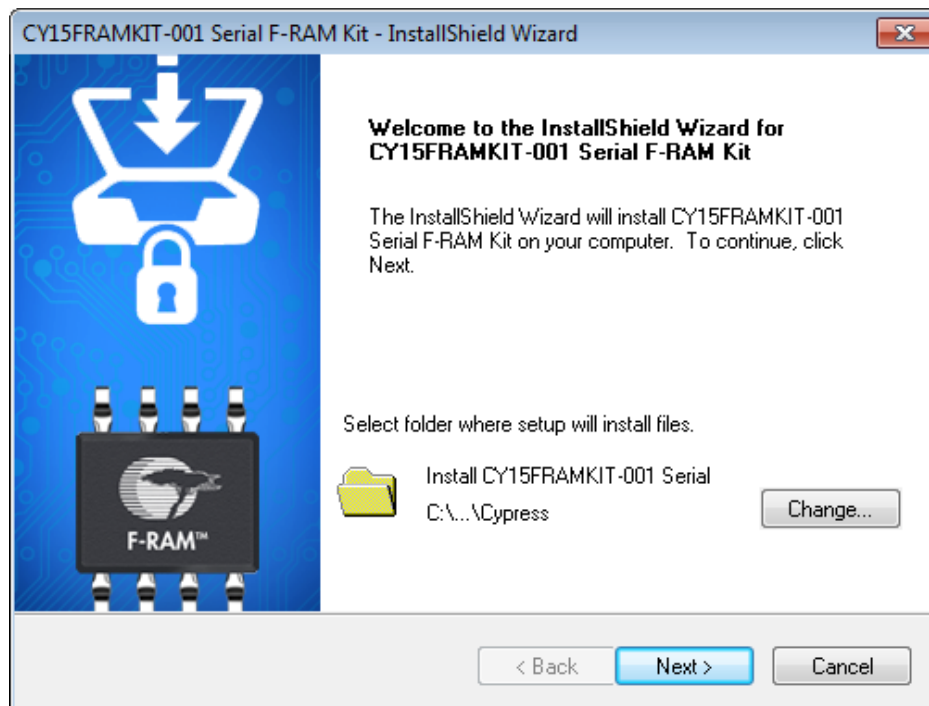


- **CY15FRAMKIT-001 Kit Setup:** この実行ファイルはキットのサンプル コード、ハードウェア ファイルおよびユーザー ドキュメントを含むキットの内容物のみをインストールします。
  - **CY15FRAMKIT-001 CD ISO:** このファイルは、CD を作成するか、または WinZip や WinRAR などの ISO 抽出プログラムで抽出するのに使用できる CD-ROM イメージ フォーマットで保存される完全なパッケージです。また、ファイルは、Virtual CloneDrive と MagicISO などの仮想ドライブ プログラムで仮想 CD のように取り付けることができます。このファイルは必要なファームウェア、ハードウェア ファイル、およびユーザー ドキュメントを全て含んでいます。
2. CD ISO ファイルを既にダウンロードした場合、仮想ドライブにロードしてください。仮想ドライブがない場合、ISO ファイルの内容を解凍してください。「CD/DVD から自動実行」が PC で有効にされていない場合、解凍した内容または取り付けた ISO のルート ディレクトリでの **cyautorun.exe** をダブルクリックします。図 2-2 に示すインストール ウィンドウは自動的に表示されます。

注: 「Kit Setup」ファイルを使用している場合、インストールするにはステップ 3 に進んでください。

3. 図 2-2 に示すように、キット起動画面が表示されます。インストールのデフォルト ロケーションは **Program Files** フォルダ下の **Cypress** フォルダです。

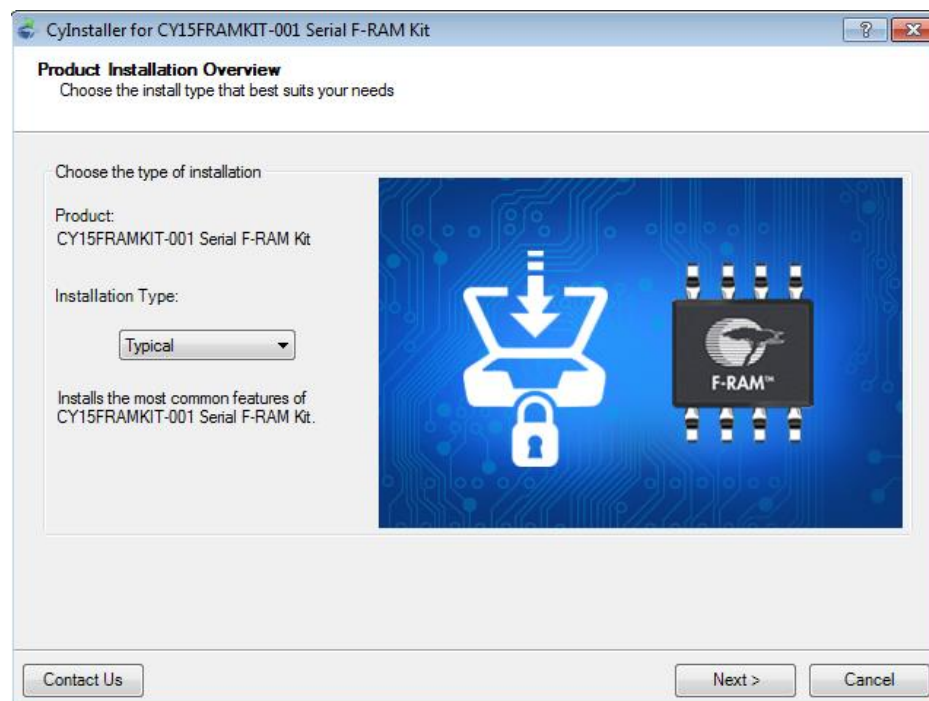
図 2-2. キット インストーラの起動画面



CY15FRAMKIT-001 キット関連のファイルをインストールしたいフォルダを選択します。ディレクトリを選択して **Next** をクリックします。

4. 図 2-3 に示すように、**Product Installation Overview** ウィンドウで **Typical installation** タイプを選択します。「Next」をクリックします。

図 2-3. 「Product Installation Overview」ウィンドウ



インストールが開始されたら、パッケージの一覧がインストール ページに表示されます。インストールが完了すると、緑色のチェックマークが各パッケージのそばに表示されます。

5. ユーザーが初めてインストールするサイプレスの製品であれば、ユーザーの連絡先情報を入力するか、または **Continue without Contact Information** オプションを選択します。**Finish** をクリックして CY15FRAMKIT-001 キットのインストールを完了します。

インストールが完了した後、キットの内容物は <Install\_Directory>\CY15FRAMKIT-001 Serial F-RAM Kit\1.0 に格納されます。デフォルトのロケーションは次のとおりです。

- Windows 7 (64 ビット) の場合: C:\Program Files (x86)\Cypress\CY15FRAMKIT-001 Serial F-RAM Kit\1.0
- Windows 7 (32 ビット) の場合: C:\Program Files\Cypress\CY15FRAMKIT-001 Serial F-RAM Kit\1.0

注: Windows 7/8/8.1 ユーザーに対しては、インストールしたファイルとフォルダは読み出し専用です。プロパティを変更するにはフォルダを右クリックして、**Properties** > **Attributes** を選択して **Read-only** オプションを無効にします。**Apply** と **OK** をクリックしてウィンドウを閉じます。

インストールが完了した後、以下のものがユーザーのコンピューターにインストールされます。

- キットの文書
  - クイック スタート ガイド
  - キット ガイド
  - リリース ノート
- ファームウェア
  - PSoC 4 F-RAM SPI サンプル プロジェクト
  - PSoC 4 F-RAM I2C サンプル プロジェクト
  - Arduino UNO F-RAM SPI サンプル プロジェクト
  - Arduino UNO F-RAM I2C サンプル プロジェクト
- ハードウェア
  - 回路図
  - レイアウト
  - ガーバー
  - PCB 組み立て図面
  - 部品表 (BOM)

## 2.3 ソフトウェアのアンインストール

ソフトウェアをアンインストールするには、以下の方法の 1 つを使用します。

- **Start > Control Panel > Programs and Features** に進んで、対象のソフトウェア パッケージを選択して **Uninstall** をクリックします。
- **Start > All Programs > Cypress > Cypress Update Manager > Cypress Update Manager** に進みます。
  - 「CY15FRAMKIT-001 Serial F-RAM Kit 1.0 Rev \*\*」を選択して、**Uninstall** をクリックします。**Product Installation Overview** ウィンドウで、**Installation Type** ドロップダウン メニューから **Remove** を選択します。指示に従ってアンインストールします。

注: この方法では、キット ソフトウェアのみがアンインストールされ、キット ソフトウェアとともにインストールされた他のソフトウェアはアンインストールされません。

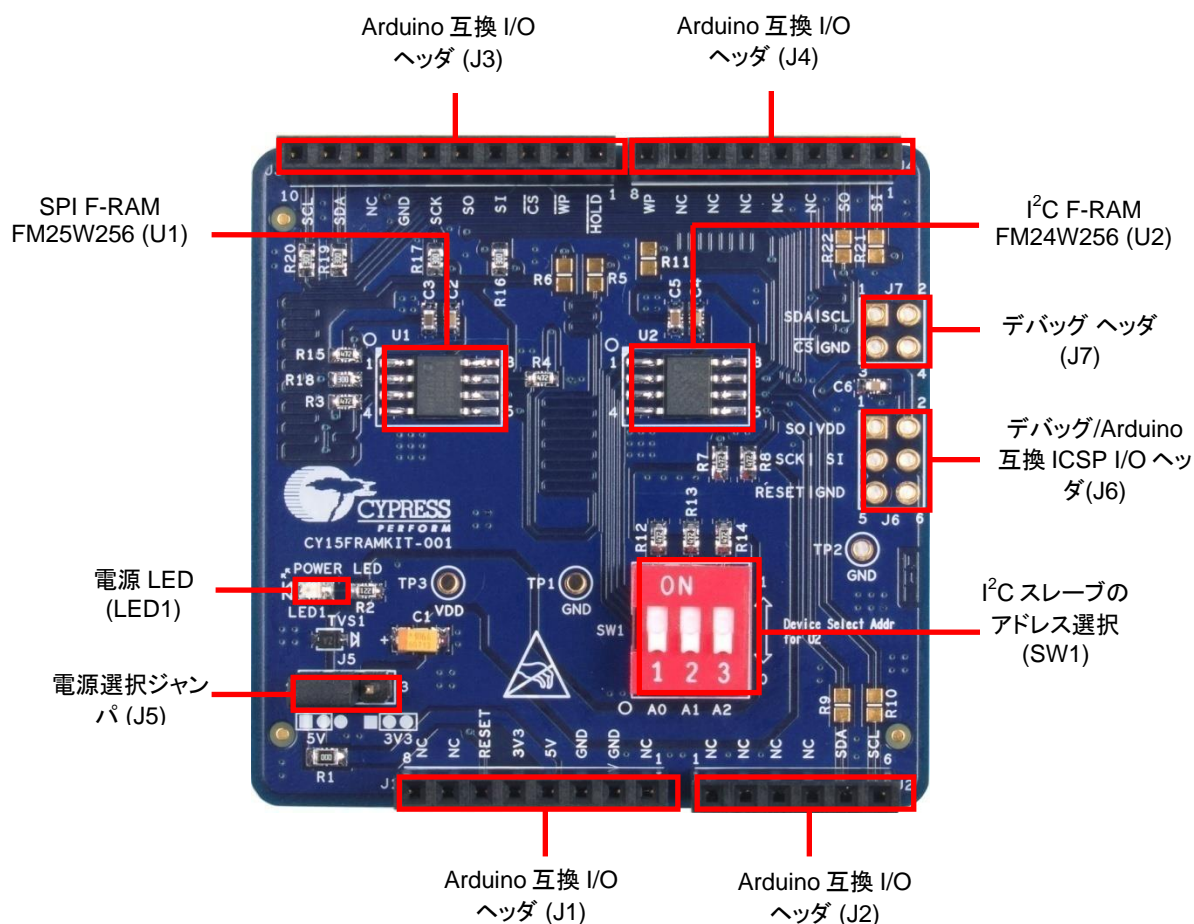
## 3. キットの概要



### 3.1 CY15FRAMKIT-001 シリアル F-RAM 開発キット の概要

CY15FRAMKIT-001 シリアル F-RAM DVK はシリアル F-RAM (SPI および I<sup>2</sup>C) の機能を理解するために使用することができます。DVK は 256K ビットの SPI F-RAM および 256K ビットの I<sup>2</sup>C F-RAM を搭載するシールド基板です。この基板には Arduino UNO と互換性がありサイプレスの PSoC 4 Pioneer Kit または Arduino UNO R3 基板のどちらかに接続するコネクタが 4 個あります。これらのコネクタはスタック可能ですので、Arduino と互換性のあるその他のシールドをコネクタにスタックすることができます。このキットは 3.3V または 5.0V で動作します。図 3-1 に DVK 基板およびその概要を示します。

図 3-1.DVK 基板の概要



「A.5 30Ω 抵抗と負荷なしの使用法」節を参照して、2 個の 30Ω 抵抗を取り付けて、CY15FRAMKIT-001 ハードウェアを Arduino UNO R2 と互換性のあるハードウェアにします。

## 3.2 キットの動作およびコンフィギュレーション ガイド

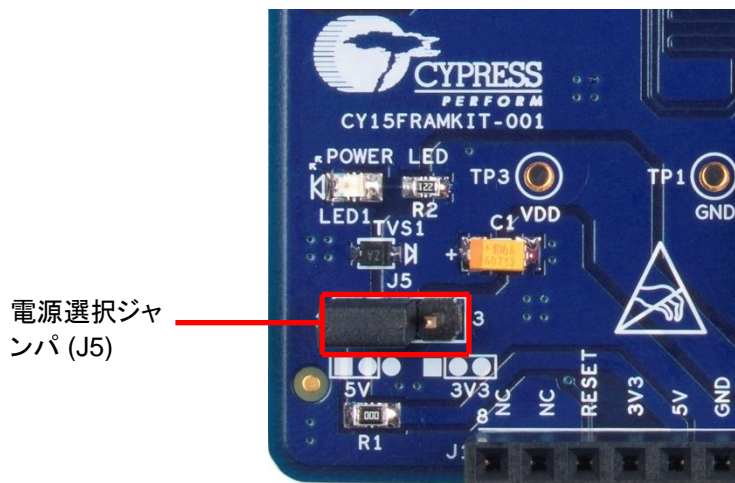
### 3.2.1 電源供給ジャンパ

図 3-2 に示すように、CY15FRAMKIT-001 はジャンパ J5 経由の選択で 3.3V と 5.0V のどちらかで動作できます。工場出荷時のジャンパのデフォルト設定は 5.0V です (ピン 1 とピン 2 を短絡)。Arduino UNO R3 基板は 5.0V で動作し、ジャンパ J5 のデフォルト設定を適用します。CY8CKIT-042 Pioneer Kit は 3.3V と 5.0V のどちらかで動作します。Pioneer Kit のデフォルト設定は 3.3V です。よって、親基板の電圧設定により、ジャンパ J5 を 5.0V (ピン 1 とピン 2 を短絡) または 3.3V (ピン 2 とピン 3 を短絡) に設定します。

#### 注意

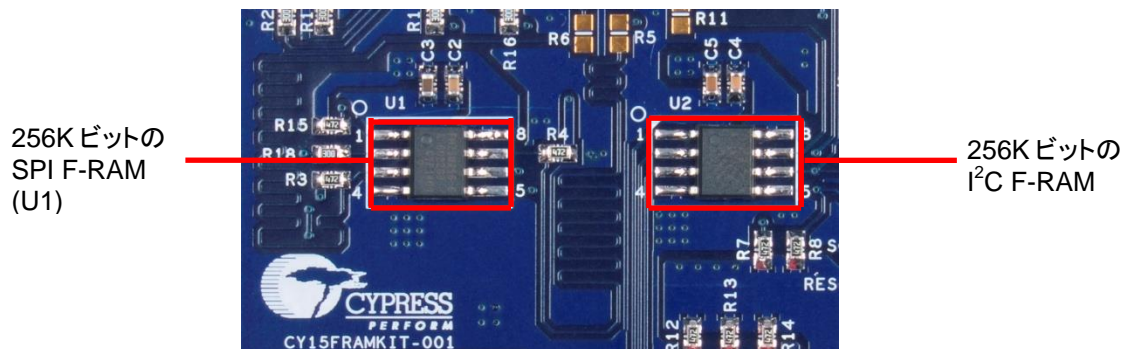
1. 外部電源から CY15FRAMKIT-001 基板に電源を供給しないでください。この基板は親基板から電源供給されるように設計されています。
2. CY15FRAMKIT-001 は 2.7V から 5.5V まで動作します。最大電圧制限 (5.5V) を超えると基板が破損する可能性があります。
3. 基板が正しく動作するために、親基板の動作電圧に一致するようにジャンパ J5 が設定されることを確保してください。

図 3-2. 電源選択ジャンパ



### 3.2.2 F-RAM デバイス

図 3-3. 基板上のサイプレスのシリアル F-RAM デバイス



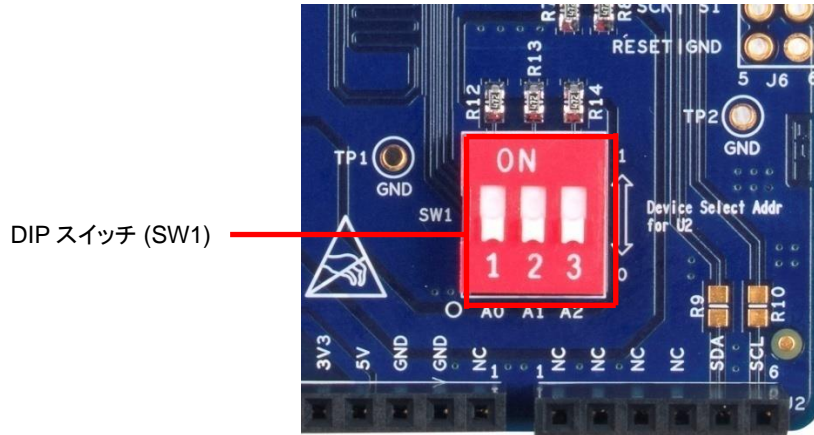
### 3.2.3 I²C F-RAM デバイスのスレーブ アドレス選択用の DIP スイッチ

図 3-4 に示すように、DVK は、個別のスレーブ選択アドレス ピン (A2~A0) を論理 LOW「0」または論理 HIGH「1」に割り当てることで I²C F-RAM 用の異なるスレーブ アドレスを設定するための DIP スイッチ オプションを提供します。DIP スイッチはオン位置



の場合、4.7kΩ のプルアップ抵抗を使って対応するスレーブ選択アドレス ピンを論理 HIGH に維持します。DIP スイッチはオン位置にない場合、対応するスレーブ選択アドレス ピンが F-RAM デバイスにより、内部弱プルダウン抵抗を介して論理 LOW に維持されます。工場出荷時の DIP スイッチのデフォルト設定はオン位置にないので、A2、A1 と A0 の値が論理 LOW「0」に設定されます。

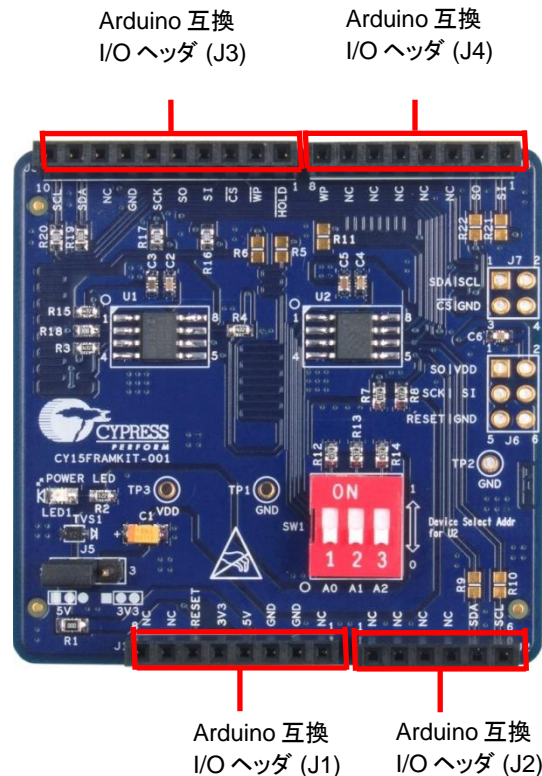
図 3-4. I<sup>2</sup>C F-RAM の A0、A1 と A2 設定用の DIP スイッチ



### 3.2.4 CY8CKIT-042/Arduino UNO R3 基板へのコネクタ

図 3-5 に、CY8CKIT-042/Arduino UNO R3 基板への Arduino-互換のコネクタ (J1、J2、J3 と J4) を示します。ユーザーはこれらのコネクタを介して DVK 基板を CY8CKIT-042/Arduino UNO R3 基板に接続することができます。回路図の詳細については、17 ページの「ハードウェア」章を参照してください。

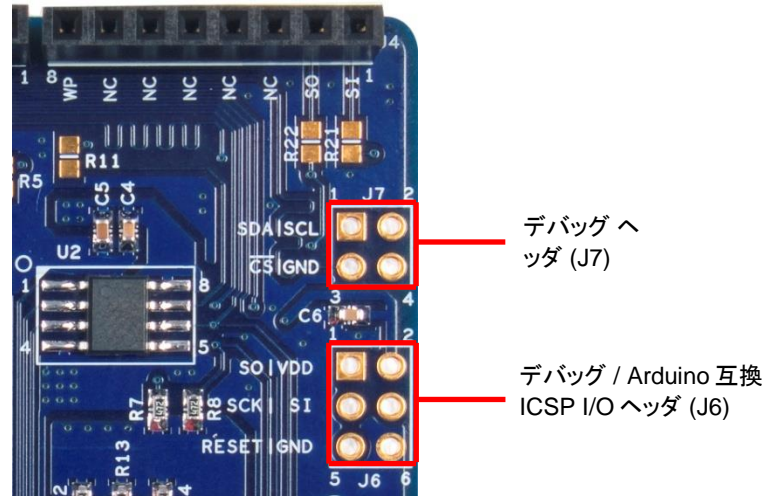
図 3-5. CY8CKIT-042/Arduino UNO R3 基板へのコネクタ



### 3.2.5 デバッグ ヘッド

デバッグ ヘッドは SPI と I<sup>2</sup>C 通信ピンへの簡単なアクセスのために提供されます。図 3-6 に示しているように、ジャンパ J6 はインサーキット シリアル プログラミング (ICSP) ヘッド ピンのレイアウトに従って、SPI 信号へのアクセスを提供します。ジャンパ J7 は SPI F-RAM デバイスのチップ選択信号および I<sup>2</sup>C 通信信号へのアクセスを可能にします。

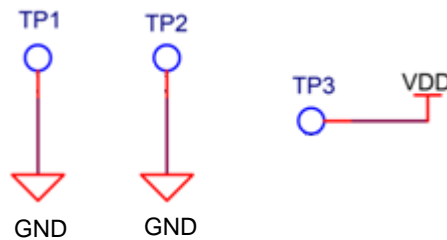
図 3-6. デバッグ ヘッド



### 3.2.6 テスト ポイント

図 3-7 に示すように、DVK 基板は V<sub>DD</sub> (TP3) および GND (TP1 と TP2) テスト ポイントを提供します。

図 3-7. テスト ポイント



#### 注意

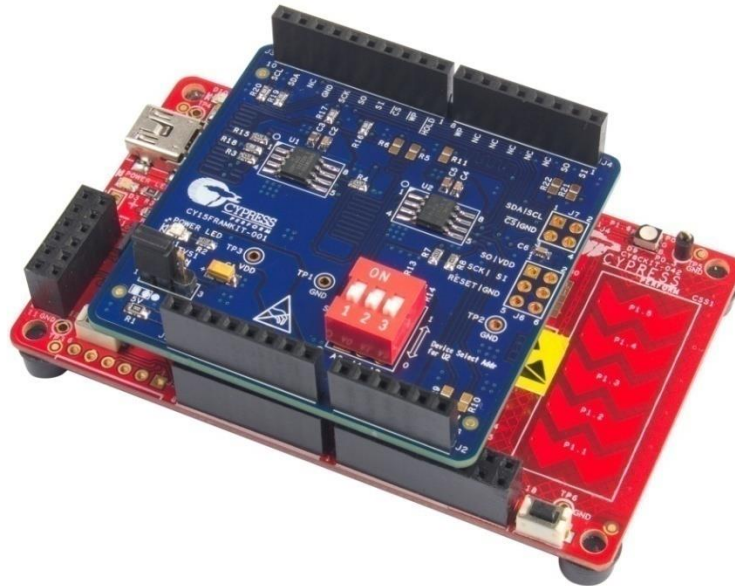
これらのテスト ポイントは調査と測定の目的に限り使用されます。基板が損傷しないように、これらのテスト ポイントを使ってキットに電源を供給しないでください。

## 3.3 PSoC 4 Pioneer Kit に接続した CY15FRAMKIT-001

CY15FRAMKIT-001 は J1、J2、J3 と J4 という 4 個のコネクタを介して PSoC 4 Pioneer Kit (CY8CKIT-042) に接続されます。シリアル F-RAM キットのデフォルトの動作電圧は 5.0V です。この動作電圧に一致させるために、PSoC 4 Pioneer Kit の電圧設定を 5.0V に変更します。ユーザーは PSoC 4 Pioneer Kit とシリアル F-RAM キットの両方で 3.3V の電源設定を選択することで、これらのキットを 3.3V で動作させることもできます。

**注:** CY15FRAMKIT-001 での J2 は 6 ピンの単列ジャンパです。PSoC 4 Pioneer Kit では、J2 は 9x2 のヘッドです。したがって、CY15FRAMKIT-001 の J2 は Pioneer 基板上の J2 ヘッドの外側列の左端にある 6 個のピンに差し込まれます。CY15FRAMKIT-001 が一方方向でのみ挿入可能なので、その事を簡単に認知できます。

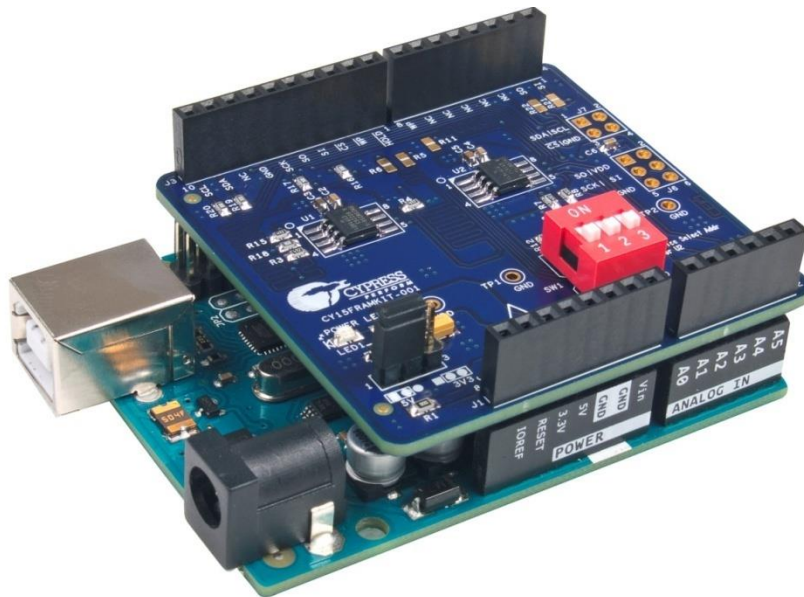
図 3-8. PSoC 4 Pioneer Kit に取り付けた CY15FRAMKIT-001



### 3.4 Arduino UNO R3 キットに接続した CY15FRAMKIT-001

CY15FRAMKIT-001 は J1、J2、J3 と J4 という 4 個のコネクタを介して Arduino UNO R3 キットに接続されます。これらのコネクタピンは 1 対 1 でマッピングされます。

図 3-9. Arduino UNO R3 キットに取り付けた CY15FRAMKIT-001





## 4. ハードウェア



### 4.1 基板の詳細

図 4-1 に示しているように、CY15FRAMKIT-001 シリアル F-RAM DVK は以下のブロックを含んでいます。

- 256K ビットの SPI F-RAM (U1)
- 256K ビットの I2C F-RAM (U2)
- 電源選択ジャンパ (J5)
- 電源 LED (LED1)
- デバイスのアドレス選択用の DIP スイッチ (SW1)
- SPI/I2C 信号用のデバッグ ヘッダ (J6, J7)
- Arduino 互換コネクタ (J1, J2, J3, J4)

図 4-1. CY15FRAMKIT-001 シリアル F-RAM DVK の詳細

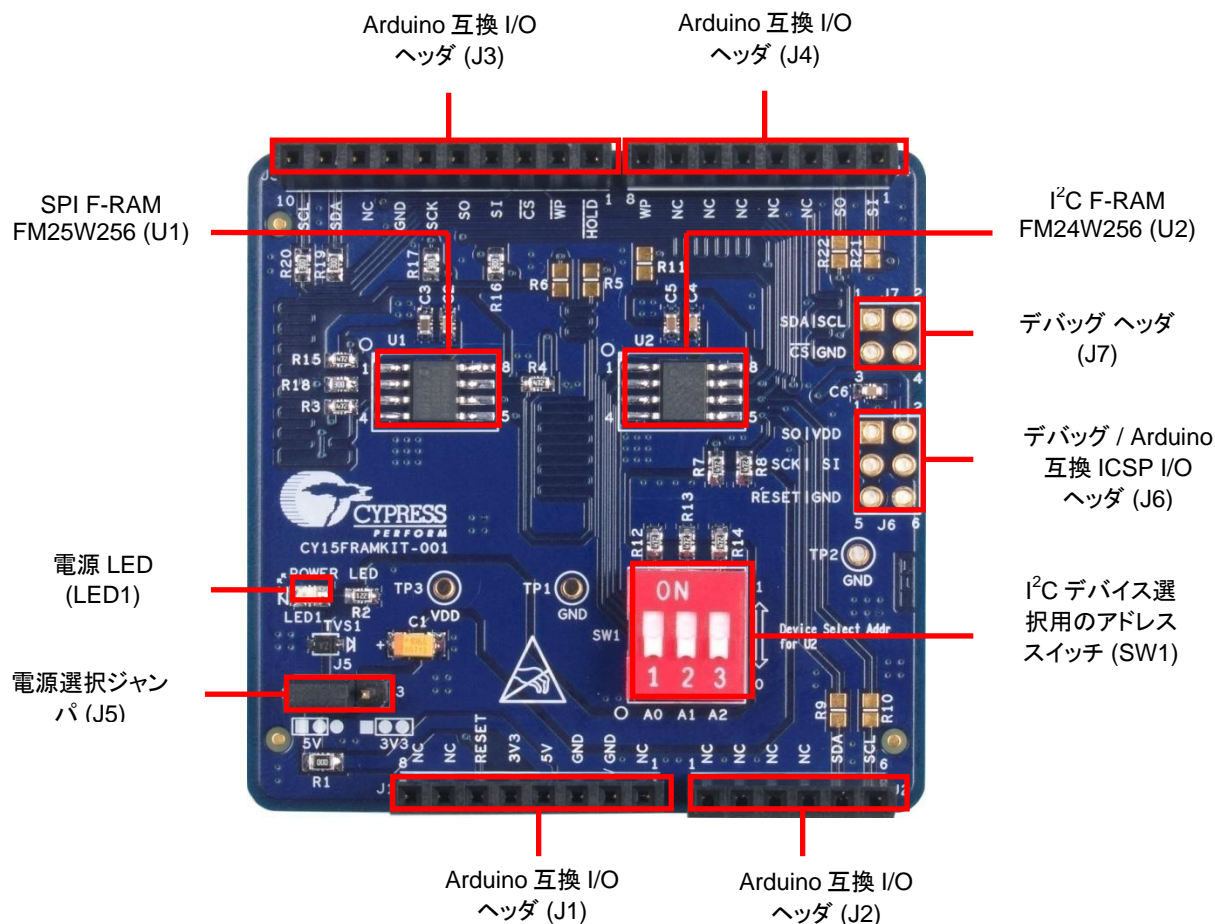
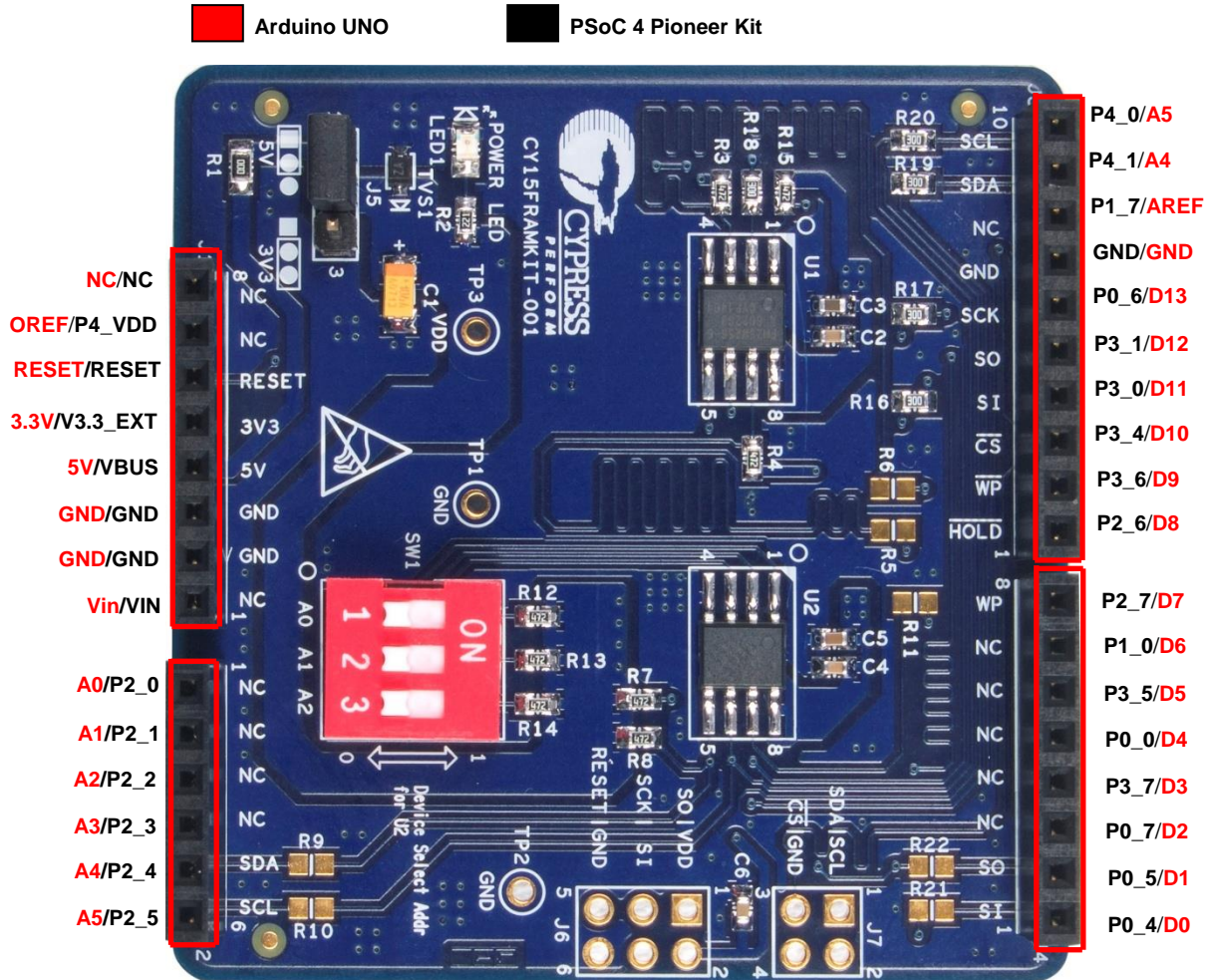


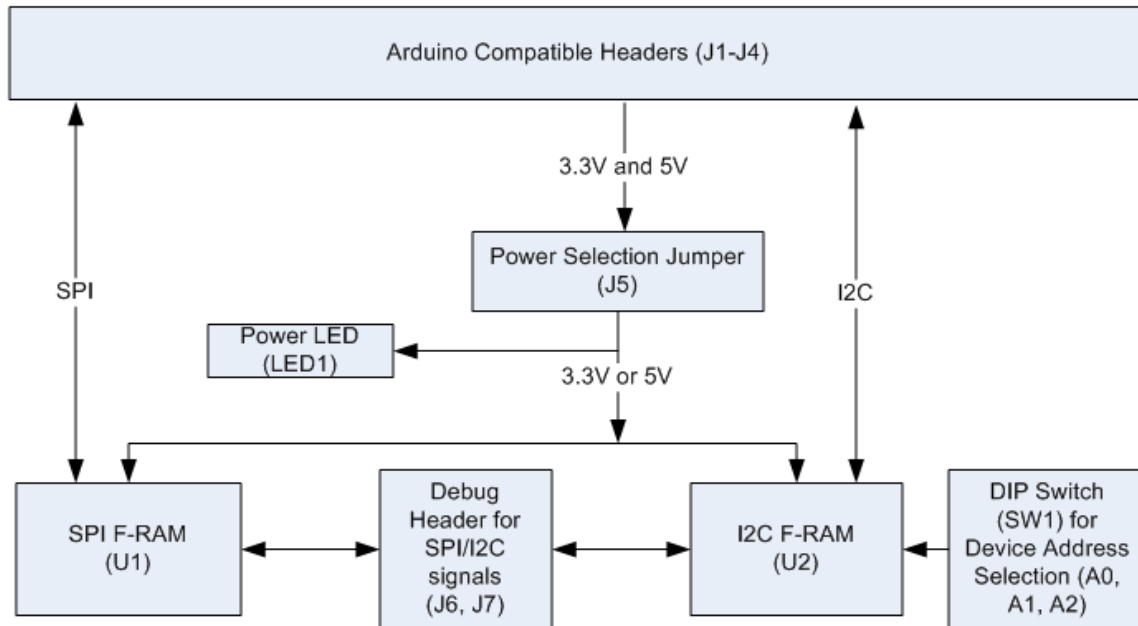
図 4-2. CY15FRAMKIT-001 シリアル F-RAM DVK と PSoC 4 Pioneer/Arduino UNO R3 キットのピン マッピング



## 4.2 動作理論

本節では、PSoC 4 Pioneer Kit をブロック レベルで説明します。

図 4-3. ブロック図



CY15FRAMKIT-001 シリアル F-RAM DVK は 256K ビットの SPI および 256K ビットの I<sup>2</sup>C F-RAM デバイスを提供します。DVK は PSoC 4 Pioneer Kit または Arduino UNO R3 キットに差し込むことができるシールド基板です。

CY15FRAMKIT-001 シリアル F-RAM DVK は電源オンの状態を示す電源 LED (LED1) を備えています。キットはジャンパ J5 経由の選択で 3.3V と 5.0V のどちらかで動作できます。また、このキットは I<sup>2</sup>C F-RAM のデバイス アドレスを変更できます。さらに、キットは SPI/I<sup>2</sup>C 信号用のデバッグ ヘッダを提供します。

## 4.3 機能の説明

### 4.3.1 SPI F-RAM デバイス (256K ビットの FM25W256)

CY15FRAMKIT-001 シリアル F-RAM DVK はサイプレスの 256K ビットの SPI F-RAM デバイスを使用します。Arduino 互換 ヘッダを介したデバイス ピンへの読み出しと書き込みアクセスが可能です。

#### 特長

- 32Kx8 構成の 256K ビット強誘電体 RAM (F-RAM)
  - 高いアクセス可能回数: 100 兆 ( $10^{14}$ ) 回の読み出し／書き込み
  - 151 年のデータ保持期間
  - NoDelay™書き込み
  - 信頼性の高い強誘電体プロセスを使用
- 高速シリアル ペリフェラル インターフェース (SPI)
  - 最大 20MHz の周波数

- シリアル フラッシュおよび EEPROM からの置き換え
- SPI モード 0 (0, 0) および SPI モード 3 (1, 1) をサポート
- 洗練された書き込み保護スキーム
  - 書き込み保護 ( $\overline{WP}$ ) ピンを使用したハードウェアによる保護
  - 書き込みディスエーブル命令を使用したソフトウェアによる保護
  - アレイの 1/4、1/2、または全体を対象としたソフトウェア ブロック保護
- 低消費電力
  - アクティブ モードの時、1MHz で 250 $\mu$ A の電流を消費
  - スタンバイ モードの時、15 $\mu$ A (標準) の電流を消費
- 広い範囲の動作電圧: VDD = 2.7V~5.5V
- 産業用途向け温度範囲: -40°C~+85°C
- 8ピン小型集積回路 (SOIC) パッケージ
- RoHS 準拠

### メモリの書き込み

SPI F-RAM への書き込みはすべて WREN (書き込みイネーブル) オペコード (0x06) から始まります。WRITE オペコード (0x02) の後に、メモリへ書き込む最初のデータ バイトを指定する 15 ビット アドレス (A14~A0) を含む 2 バイト アドレスが続きます。2 バイト アドレスの上位ビットは無視されます。アドレスの 2 バイトの後に続くバイトは順次に書き込まれるデータ バイトです。バス マスターがクロックを送り、 $\overline{CS}$  を LOW に維持している限り、アドレスは内部でインクリメントされます。7FFFh の最終アドレスに達すると、カウントは 0000h に戻ります。データは MSB から書き込みます。 $\overline{CS}$  の立ち上がりエッジで書き込み動作を終了します。SPI F-RAM の書き込み動作は下図に示します。

図 4-4. SPI F-RAM メモリ書き込み

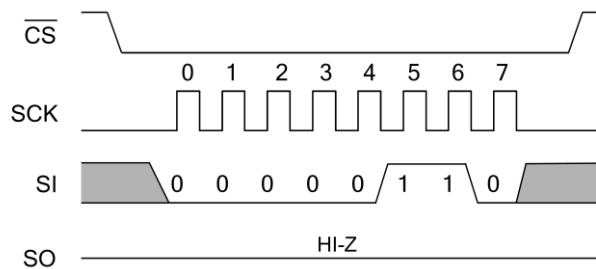
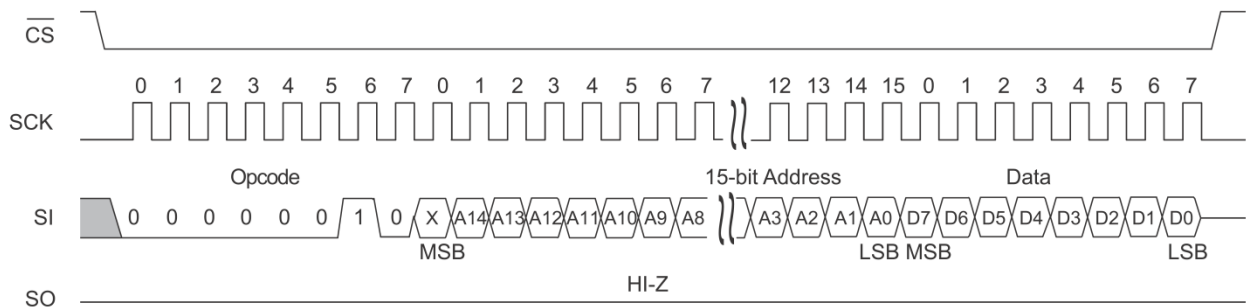


図 4-5. SPI F-RAM メモリ書き込み

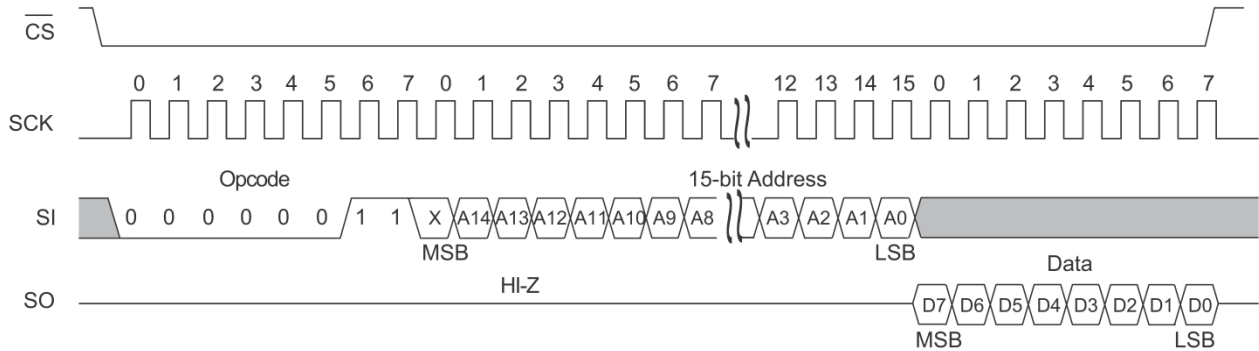


### メモリ読み出し

$\overline{CS}$  の立ち上がりエッジの後に、バス マスターは READ オペコード (0x03) を発行できます。READ コマンドの後には、読み出し動作の開始バイトを指定する 15 ビット アドレス (A14~A0) を含む 2 バイトのアドレスが続きます。アドレスの上位ビットは無視さ

れます。READ オペコードとアドレスの 2 バイトが発行された後、デバイスは SO ピンの次の 8 つのクロックで読み出しデータ バイトを出力します。SI 入力は読み出しデータ バイトの出力中には無視されます。後続のバイトは順次に読み出されるデータ バイトです。バス マスターがクロックを送り、 $\overline{CS}$  が LOW である限り、アドレスは内部でインクリメントされます。7FFFh の最終アドレスに達すると、カウントは 0000h に戻ります。データは MSB から読み出します。 $\overline{CS}$  の立ち上がりエッジで読み出し動作を停止し、SO ピンをトリステートにします。SPI F-RAM の読み出し動作は下図に示します。

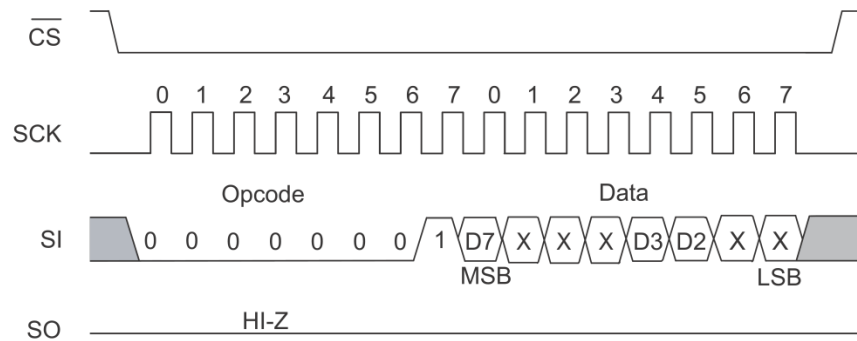
図 4-6. SPI F-RAM メモリ読み出し



### ステータス レジスタ書き込み

ステータス レジスタ書き込み WRSR コマンド (0x01) により、SPI バス マスターが SPI F-RAM のステータス レジスタに書き込むことができます。

図 4-7. SPI F-RAM ステータス レジスタ書き込み

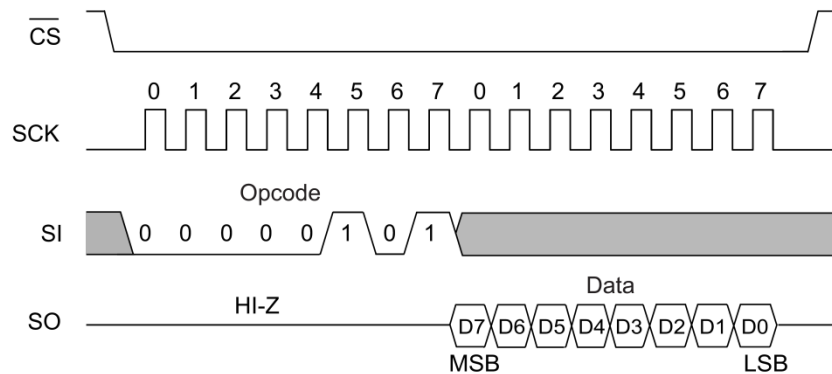


### ステータス レジスタ読み出し

ステータス レジスタ読み出し RDSR コマンド (0x05) により、バス マスターは SPI F-RAM のステータス レジスタの内容を検証することができます。ステータス レジスタを読み出すことで、書き込み保護機能の現時点の状態に関する情報を得ます。RDSR オペコードに続いて、SPI F-RAM デバイスはステータス レジスタの内容を持つ 1 バイトを返します。



図 4-8. SPI F-RAM ステータス レジスタ読み出し



F-RAM デバイスの詳細については、「[FM25W256 データシート](#)」を参照してください。

### 4.3.2 I<sup>2</sup>C F-RAM デバイス (256K ビットの FM24W256)

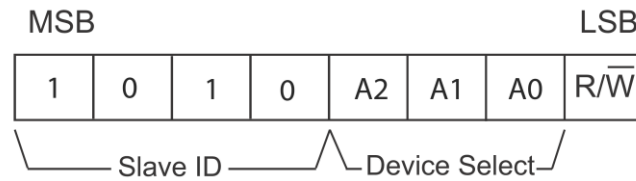
CY15FRAMKIT-001 シリアル F-RAM DVK は 256K ビットの I<sup>2</sup>C F-RAM デバイスを使用します。Arduino 互換ヘッダを介したデバイスへの読み出しと書き込みアクセスが可能です。

#### 特長

- 256K ビットの F-RAM は論理的に 32K × 8 として構成
  - 高いアクセス可能回数: 100 兆 (10<sup>14</sup>) 回の読み出し／書き込み
  - 151 年のデータ保持期間
  - NoDelay™ 書き込み
  - 信頼性の高い強誘電体プロセスを使用
- 高速 2 線式シリアル インターフェース (I<sup>2</sup>C)
  - 最大 1MHz の周波数
  - シリアル (I<sup>2</sup>C) EEPROM からの置き換え
  - 100kHz と 400kHz に応じた従来のタイミングに対応
- 低消費電力
  - アクティブ モードの時、100kHz で 100μA の電流を消費
  - スタンバイ モードの時、15μA (標準) の電流を消費
- 広い範囲の動作電圧: VDD = 2.7V~5.5V
- 産業用途向け温度範囲: -40°C~+85°C
- 8ピン小型集積回路 (SOIC) パッケージ
- RoHS 準拠

#### スレーブ デバイス アドレス

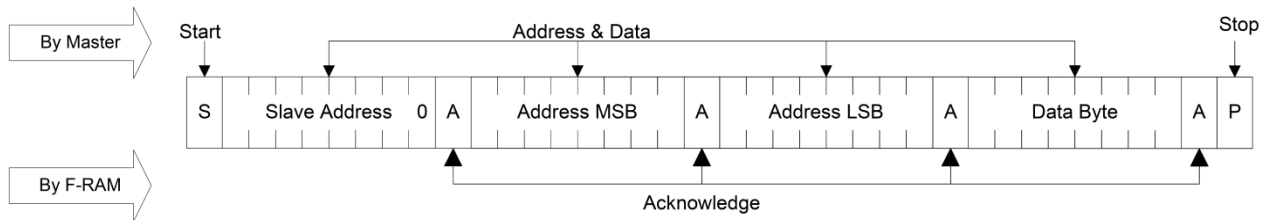
START 条件後に FM24W256 が待ち受けている最初のバイトはスレーブ アドレスです。図 4-9 に示すように、スレーブ アドレスはデバイス タイプまたはスレーブ ID、デバイス選択アドレス ビット、およびトランザクションが読み出しか書き込みかを指定するビットから成ります。ビット 7~4 はデバイス タイプ (スレーブ ID) であり、FM24W256 の場合は 1010b にセットする必要があります。これらのビットにより、他の機能タイプが同じアドレス範囲の I<sup>2</sup>C バス上に存在することができます。ビット 3~1 はデバイス選択アドレス ビットです。デバイスを選択するには外部アドレス ピンの値と一致させる必要があります。各デバイスに異なるアドレスを割り当てることで、最大 4 個の FM24W256 デバイスが同じ I<sup>2</sup>C バス上に存在できます。ビット 0 は読み出し／書き込みビット (R/W) です。R/W = 「1」は読み出し動作を、R/W = 「0」は書き込み動作を示します。

図 4-9. I<sup>2</sup>C F-RAM スレーブ デバイス アドレス レジスタ


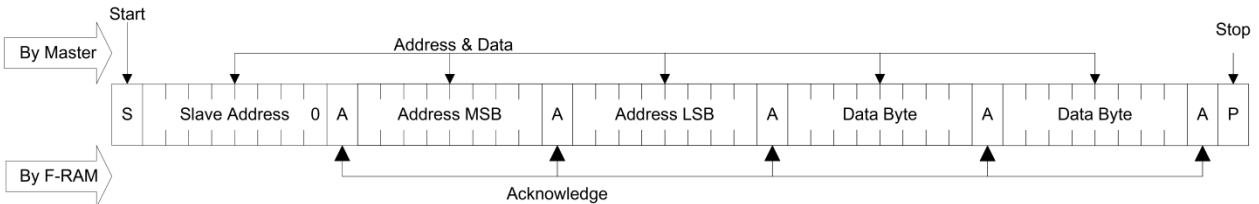
### メモリの書き込み

すべての書き込みはスレーブ アドレスに続いてメモリ アドレスの順で始まります。バス マスターはスレーブ アドレスの LSB ( $R/\overline{W}$  ビット) を「0」にセットすることで書き込み動作であることを示します。アドレス指定した後、バス マスターは 1 データ バイトずつメモリに送信して、メモリはアクノリッジ条件を生成します。連続バイトはいくつでも書き込めます。アドレス範囲の終わりに内部で達した場合、アドレス カウンタは 7FFFh から 0000h にラップアップします。

シングルバイト書き込み:

図 4-10. I<sup>2</sup>C F-RAM メモリのシングルバイト書き込み


マルチバイト書き込み:

図 4-11. I<sup>2</sup>C F-RAM マルチシステム メモリ書き込み


### メモリ読み出し

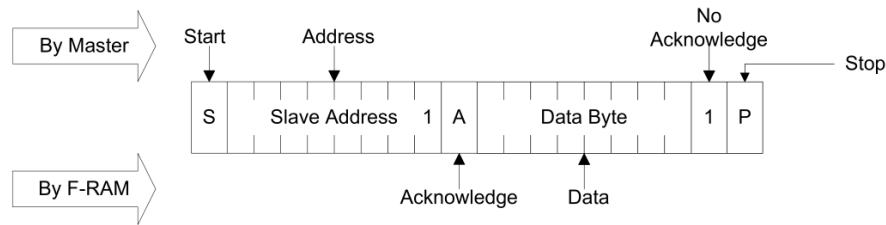
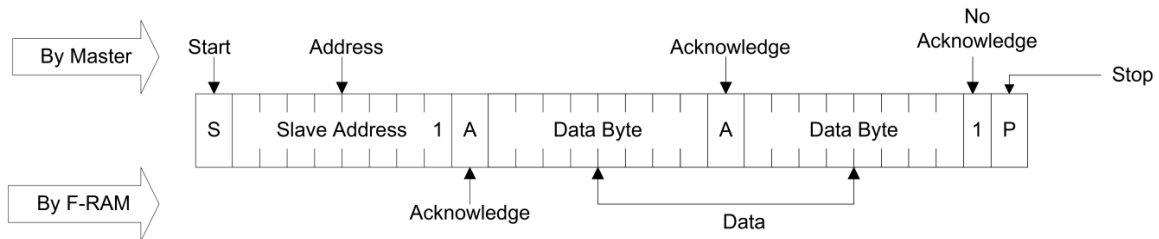
読み出し動作には、現時点のアドレス読み出しと選択的な (ランダム) アドレス読み出しという 2 つの基本的なタイプがあります。現時点のアドレス読み出しでは、FM24W256 は内部アドレス ラッチを使用してアドレスを提供します。選択的読み出しでは、ユーザーはアドレスを特定の値に設定する手順を行います。

現時点のアドレス読み出しと連続読み出し

FM24W256 は内部ラッチを使用して読み出しアドレスを指定します。現時点のアドレス読み出しは、アドレス ラッチ内の既存の値を読み出し動作の開始点として使用します。システムは、直近の動作のアドレス直後に続くアドレスから読み出します。

現時点のアドレス読み出しを行うために、バス マスターは LSB が「1」にセットされたスレーブ アドレスを提供します。これは読み出し動作が要求されることを示します。完全なスレーブ アドレスを受信した後、FM24W256 は次のクロックで現時点のアドレスからデータをシフトアウトし始めます。現時点のアドレスは内部アドレス ラッチで保持されている値です。

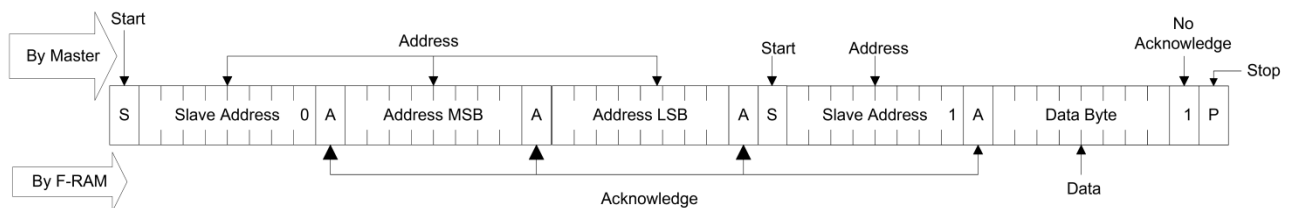
現時点のアドレスを冒頭に、バス マスターはバイトをいくつでも読み出せます。従って連続読み出しは単に複数バイトを転送する現時点のアドレス読み出しです。各バイト転送の後、内部アドレス カウンターはインクリメントされます。

図 4-12. I<sup>2</sup>C F-RAM シングル バイト現時点のアドレス読み出し

図 4-13. I<sup>2</sup>C F-RAM マルチシステム連続読み出し


### 選択的 (ランダム) 読み出し

ランダムなアドレスを読み出し動作の開始点に設定する簡単な方法があります。これは、書き込み動作の冒頭 3 バイトを使って内部アドレスをセットし、続いて読み出し動作を行うことで実現します。

選択的読み出しを行うために、バス マスターは LSB ( $R/\bar{W}$ ) が 0 にセットされたスレーブ アドレスを送信します。これは書き込み動作を示します。書き込みプロトコルに従って、バス マスターはそれに続き、内部アドレス ラッチにロードされるアドレス バイトを送信します。FM24W256 がアドレスをアクノリッジした後、バス マスターは START 条件を発行します。これにより、書き込み動作が中止されると同時に、LSB が「1」にセットされたスレーブ アドレスで読み出しコマンドが発行されます。これで、動作は現時点のアドレス読み出しとなります。

図 4-14. F-RAM I<sup>2</sup>C 選択的 (ランダム) 読み出し




## 5. サンプル プロジェクト



サイプレスが提供する CY15FRAMKIT-001 シリアル F-RAM DVK パッケージには 4 つのサンプル プロジェクトが組み込まれます。これらのサンプル プロジェクトにより、ユーザーはマイクロコントローラーとのシリアル F-RAM インターフェースを理解できます。サンプル プロジェクトは PSoC 4 Pioneer Kit と Arduino UNO R3 キットに基づきます。

[www.cypress.com/go/CY15FRAMKIT-001](http://www.cypress.com/go/CY15FRAMKIT-001) からサンプル プロジェクトをダウンロードします。インストーラはサンプル プロジェクトを<Install\_Directory>\CY15FRAMKIT-001 Serial F-RAM Kit\1.0\Firmware\フォルダに配置します。

以下のサンプル プロジェクトは PSoC 4 ベースのプロジェクトで、PSoC Creator ソフトウェアと PSoC 4 Pioneer Kit を必要とします。

- PSoC4\_FRAM\_SPI
- PSoC4\_FRAM\_I2C

以下のサンプル プロジェクトは Arduino ベースのプロジェクトで、Arduino IDE と Arduino UNO R3 キットを必要とします。

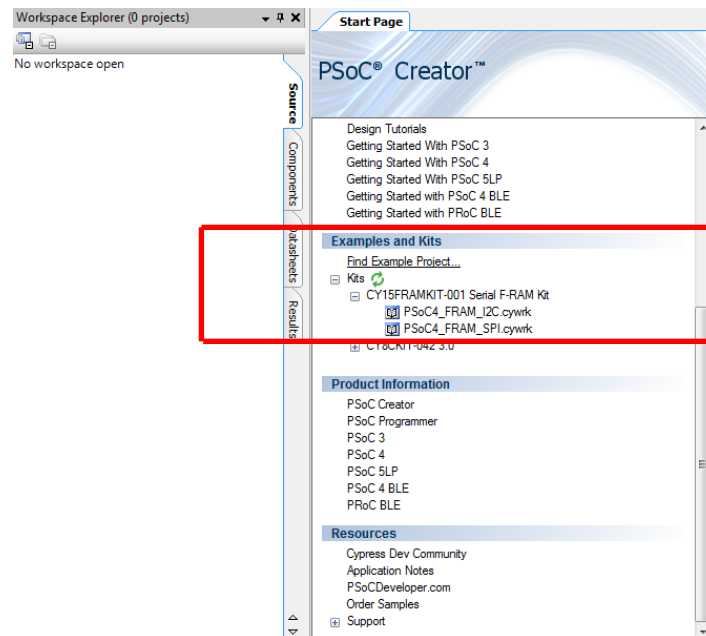
- Arduino\_FRAM\_SPI
- Arduino\_FRAM\_I2C

### 5.1 PSoC 4 Pioneer Kit のプログラミング

PSoC 4 サンプル プロジェクトをプログラムするには以下の手順に従ってください (この手順では PSoC4\_FRAM\_SPI プロジェクトが例として使われます)。

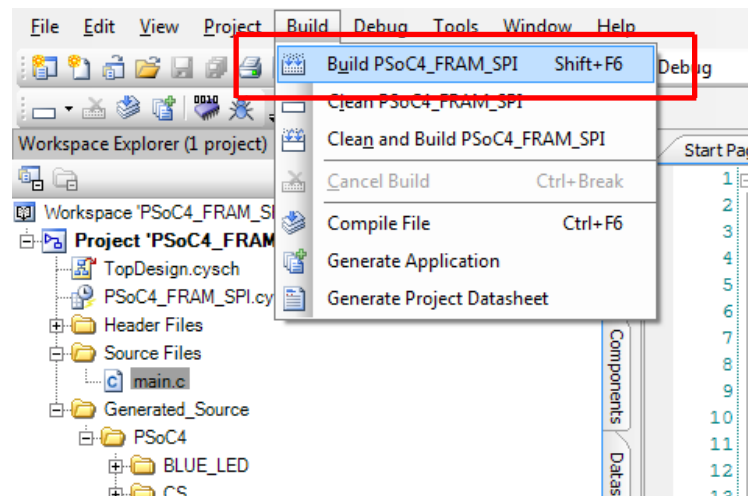
1. **Start > All Programs > Cypress > PSoC Creator 3.1 > PSoC Creator 3.1** から PSoC Creator を起動します。PSoC Creator がまだインストールされていない場合、[www.cypress.com/psoccreator](http://www.cypress.com/psoccreator) からダウンロードします。
2. 「Start Page」の **Examples and Kits** セクションで **Kits > CY15FRAMKIT-001** を選択します。図 5-1 に示しているように、サンプル プロジェクトの一覧が表示されます。所望のサンプル プロジェクトをクリックします。

図 5-1. PSoC Creator でサンプル プロジェクトを開く



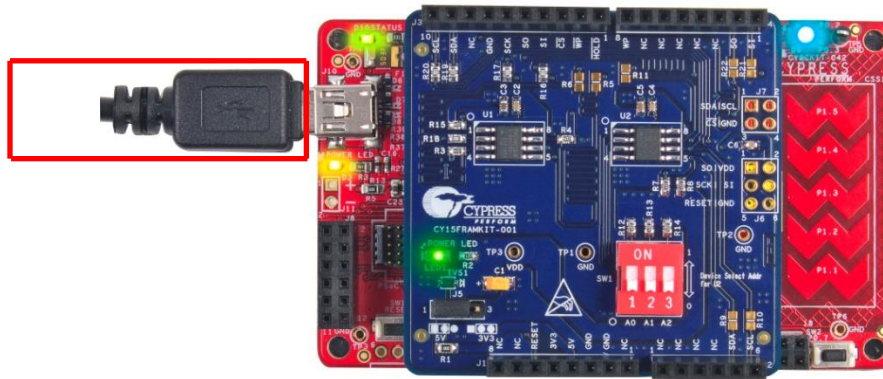
3. プロジェクトを保存したいフォルダを選択して **OK** をクリックします。
4. **Build > Build <プロジェクト名>** をクリックして、hex ファイルを生成することで、コード例をコンパイルします。

図 5-2. PSoC Creator でプロジェクトをコンパイルする



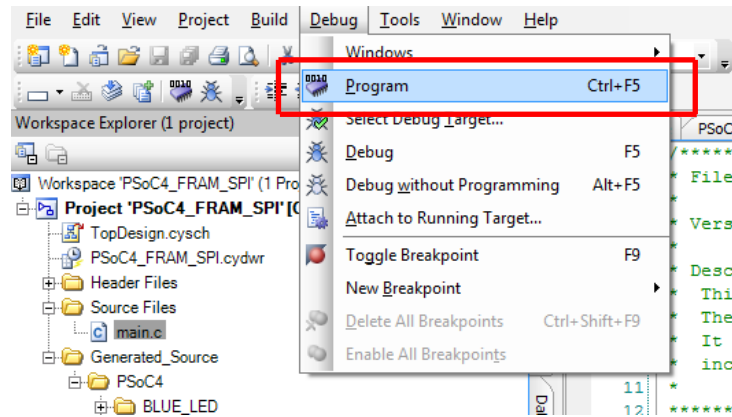
5. 動作電圧を選択します。CY15FRAMKIT-001 はジャンパ J5 経由の選択で 3.3V と 5.0V のどちらかで動作できます。工場出荷時のジャンパのデフォルト設定は 5.0V です (ピン 1 とピン 2 を短絡)。Arduino UNO R3 基板は 5.0V で動作し、ジャンパ J5 のデフォルト設定を適用します。CY8CKIT-042 Pioneer Kit は 3.3V と 5.0V のどちらかで動作します。Pioneer Kit のデフォルト設定は 3.3V です。よって、親基板の電圧設定により、ジャンパ J5 を 5.0V (ピン 1 とピン 2 を短絡) または 3.3V (ピン 2 とピン 3 を短絡) に設定します。PSoC 4 をプログラムするには、図 5-3 に示すように、USB ミニ B コネクタ J10 に接続した USB ケーブルを使って Pioneer 基板を PC に接続します。

図 5-3. USB を介して PSoC 4 Pioneer Kit を PC に接続



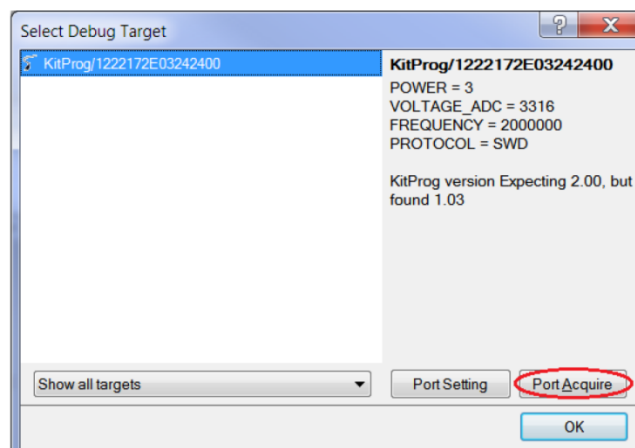
6. PSoC Creator で **Debug > Program** をクリックします。

図 5-4. PSoC Creator でデバイスをプログラム



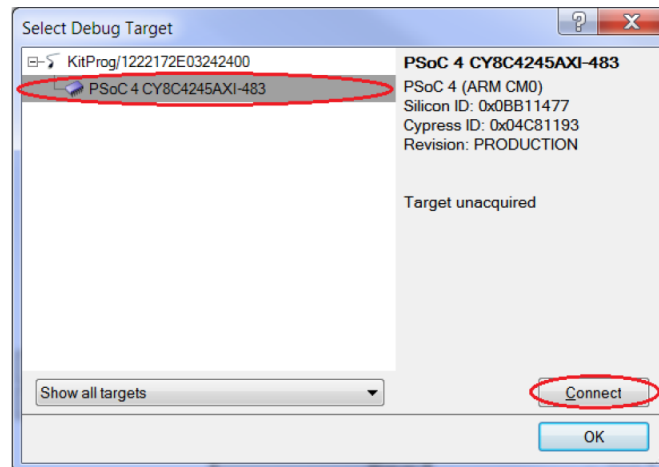
7. デバイスがまだ選択されていない場合は、PSoC Creator は Programming ウィンドウを開きます。KitProg を選択して **Port Acquire** ボタンをクリックします。

図 5-5. PSoC Creator でデバイスを選択



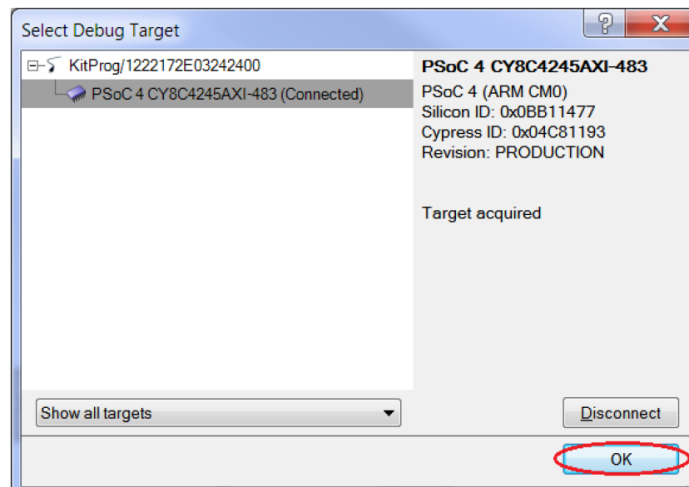
8. デバイスは、選択された後、KitProg の下に階層構造で表示されます。そして、**Connect** ボタンをクリックします。

図 5-6. PSoC Creator でデバイスに接続



9. **OK** をクリックして、ウィンドウを閉じると、書き込みが始まります。

図 5-7. PSoC Creator でデバイスをプログラム

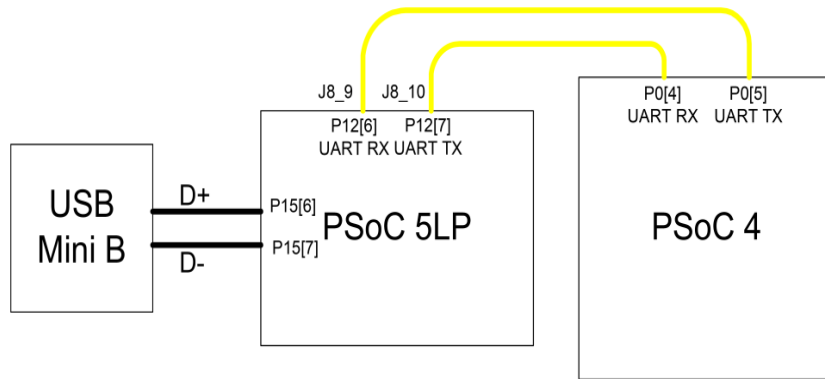


## 5.2 UART の設定

本節では、シリアル COM 接続用にハイパーターミナル、またはハイパーターミナルの代わりに PuTTY を使用して PSoC 4 サンプル プロジェクトの出力を PC に表示するための UART インターフェースの設定について説明します。

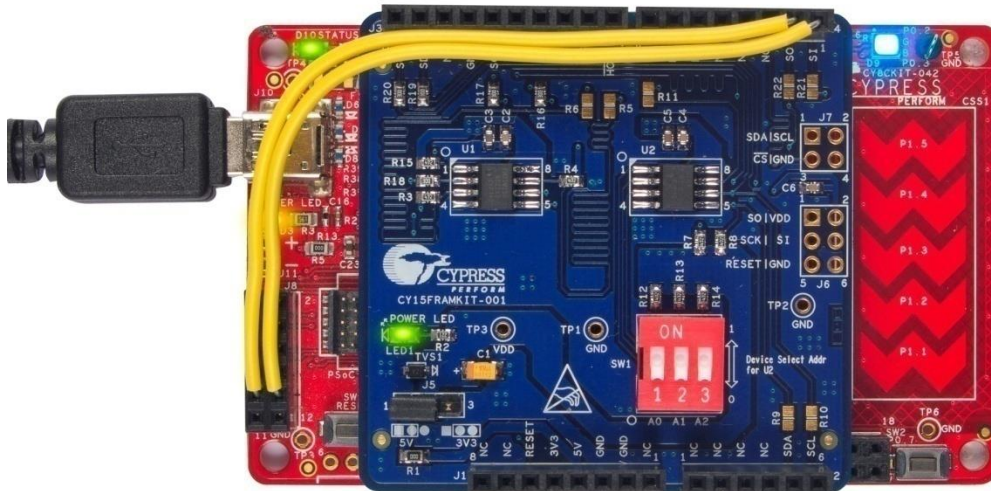
PSoC 4 Pioneer Kit の PSoC 5LP デバイスは図 5-8 に示すように USB-UART ブリッジとして設定できます。PSoC 4 サンプル プロジェクトの両方は USB ポートを介して結果を PC に送信するための UART 通信プロトコルを含んでいます。ハードウェア設定およびアーキテクチャの詳細については、PSoC 4 Pioneer Kit のウェブ ページ (<http://www.cypress.com/?rid=77780>) を参照してください。

図5-8. PSoC4とPSoC 5LP間のUART接続のブロック図



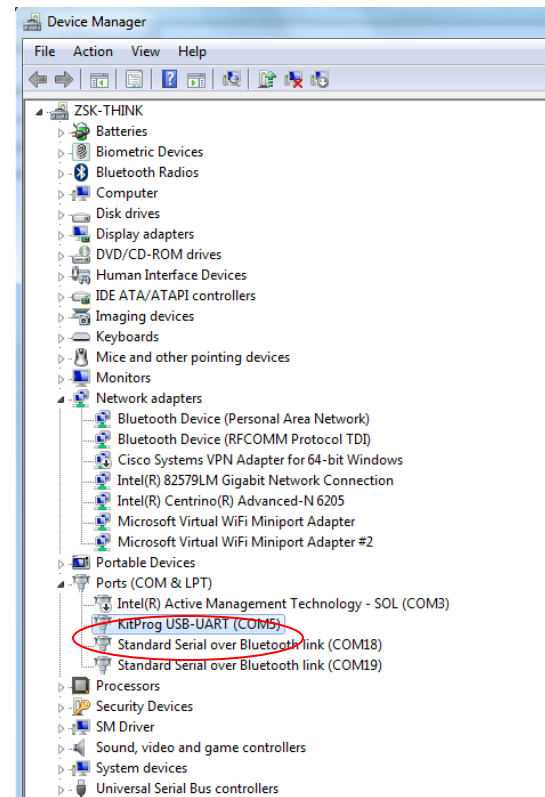
CY15FRAMKIT-001キット上のJ4\_1 (SI) を介してアクセス可能なPSoC 4のRXポートをJ8\_10 (TX) に接続します。同様に、CY15FRAMKIT-001キットのJ4\_2 (SO) を介してアクセス可能なPSoC 4のTXポートをJ8\_9に接続します (図5-9を参照)。

図 5-9. PSoC 4 Pioneer Kit 上の UART 接続



USB ミニ B を J10 に接続します。キットは **KitProg USB-UART** として認識され、**Device Manager > Ports (COM & LPT)** の下に表示します。KitProg USB-UART には 1 個の通信ポートが割り当てられます。

図5-10. Device ManagerでのKitProg USB-UART

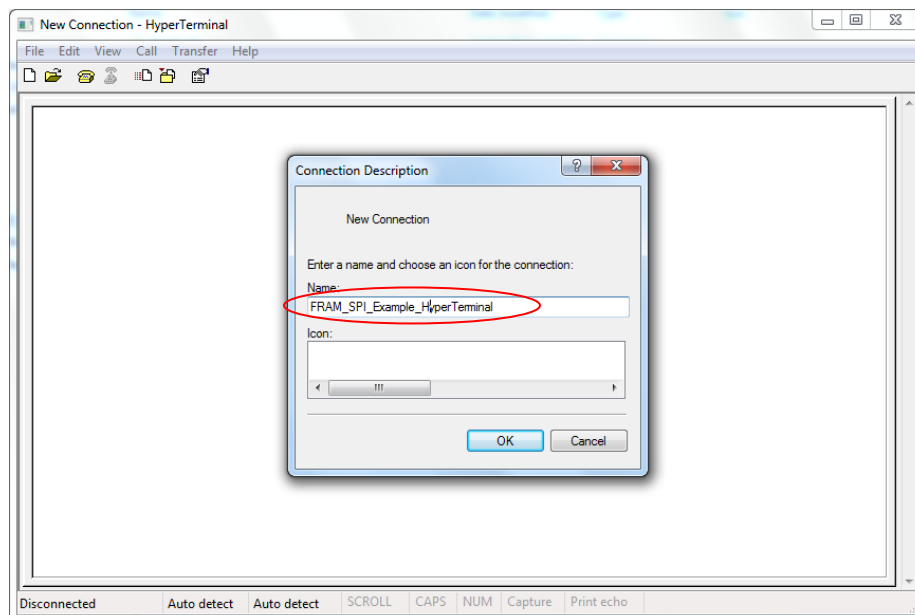


次の節では、PC 上のハイパーターミナルまたは PuTTY を PSoC 4 Pioneer Kit との通信用に使用してシリアル接続を設定する手順を説明します。

### 5.2.1 ハイパーターミナルの設定

ハイパーターミナルを開き、**File > New Connection** を選択し、新しい接続の「名前」を入力して、**OK** をクリックします。

図 5-11. 新しい接続を開く



1. 通信ポートを選択できる新しいウィンドウが開きます。

**Connect using** で適切な通信ポート「COMX」(または KitProg USB-UART に割り当てられた特有の通信ポート) を選択して、**OK** をクリックして、ハイパーターミナルの次の設定に進みます。

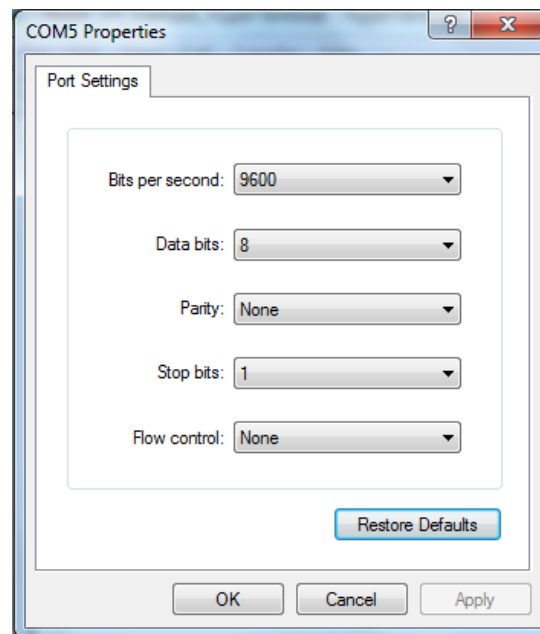
このサンプル コードでは **COM5** を使用します。ユーザーのセットアップ用の COM 設定を検証して適切な COMX を選択します。

図 5-12. 通信ポートを選択



2. **COMX Properties** ウィンドウが開きます。図 5-13 に示すように、このウィンドウの **Port Settings** タブで **Restore Defaults** をクリックして、「Bits per second」、「Data bits」、「Parity」、「Stop bits」および「Flow control」パラメーターを設定して、**OK** をクリックします。サンプル プロジェクトでの PSoC 4 UART ポート設定を変更すれば、それと同じ設定をハイパーターミナルで適用することを確認してください。**OK** をクリックして、通信を開始します。

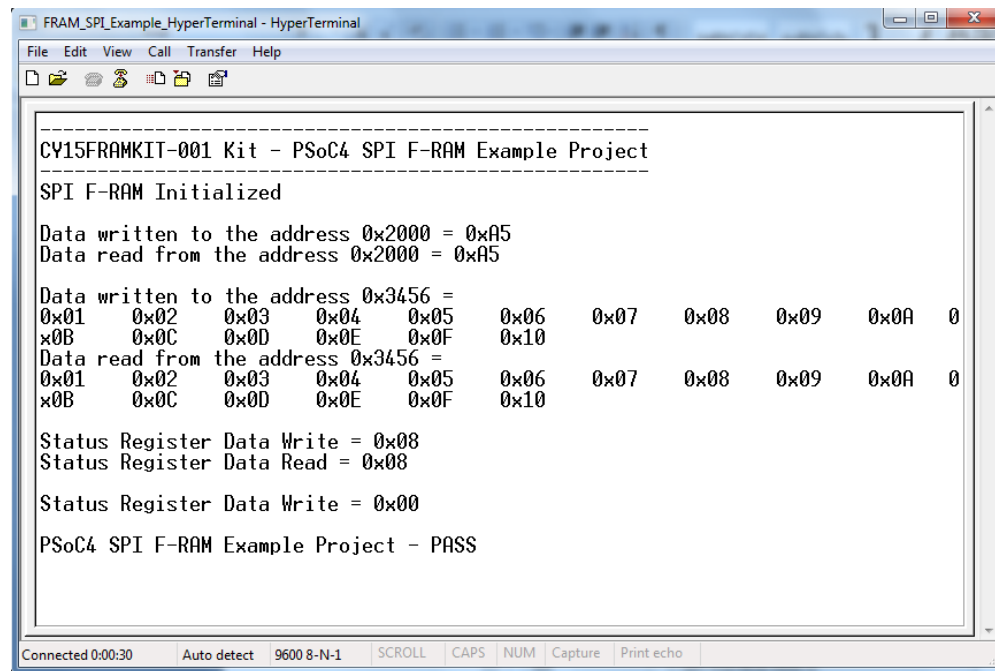
図 5-13. 通信ポートを設定





3. ハイパーターミナル ウィンドウが開き、PSoC 4 のサンプル プロジェクトの出力がこのウィンドウに表示されます。

図 5-14. 通信ポートを設定

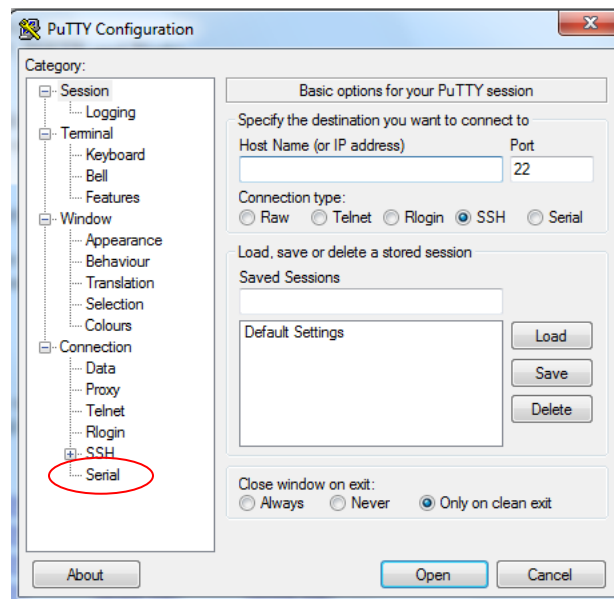


## 5.2.2 PuTTY の設定

PuTTY はウィンドウズ用のフリーの SSH と Telnet クライアントです。[www.putty.org/](http://www.putty.org/) から PuTTY をダウンロードできます。

1. PuTTY アイコンをダブルクリックして、**Connection** 下の **Serial** を選択します。

図 5-15. 新しい接続を開く

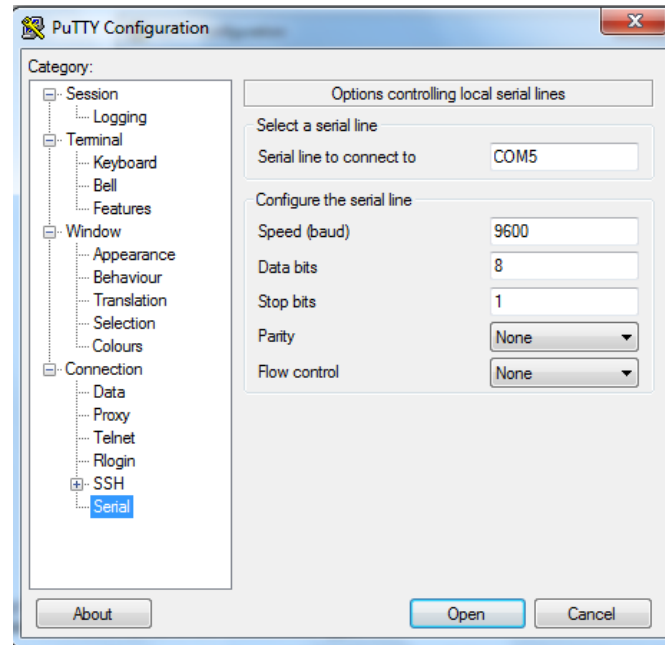




- 通信ポートを選択できる新しいウィンドウが開きます。**Serial line to connect to** フィールドに COMX を入力します。このサンプル コードでは **COM5** を使用します。ユーザーのセットアップ用の COM 設定を検証して適切な COMX を選択します。

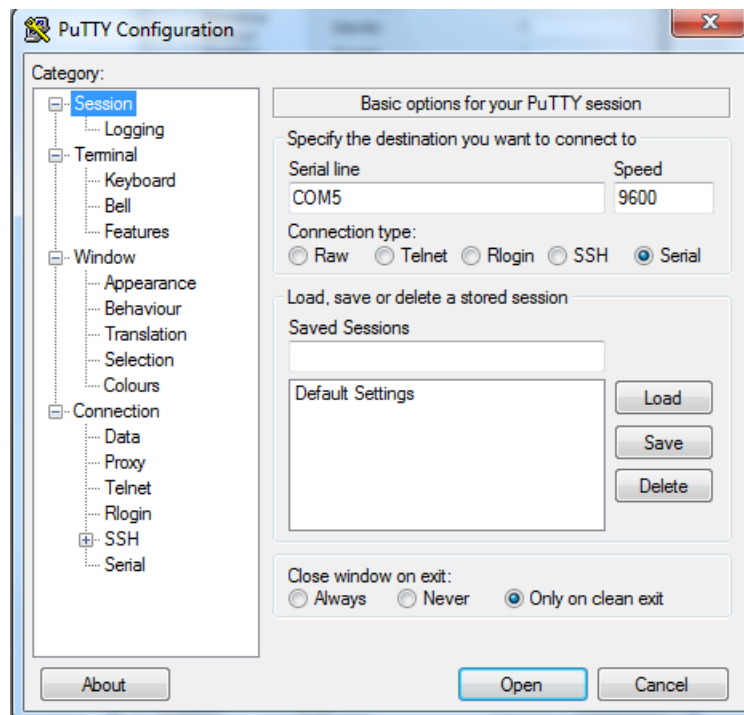
**Configure the serial line** 下の *Speed (baud)*、*Data bits*、*Stop bits*、*Parity* および *Flow control* を選択します。**Session** をクリックして、**Connection type** 下の **Serial** を選択します。

図 5-16. 通信ポートを選択・設定



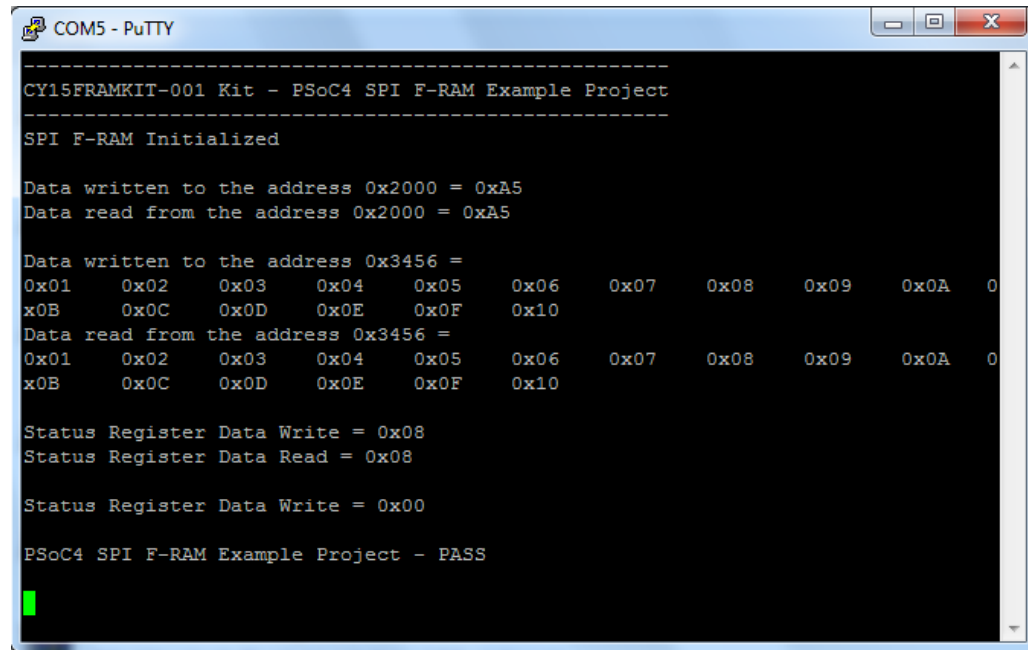
- Category** 下の **Session** を選択して、**Communication type:** を「Serial」に設定します。**Open** をクリックします。

図 5-17. PuTTY で通信タイプを選択



- COM 端末ソフトウェアは PSoC 4 からのデータを表示します。例えば、SPI サンプル プロジェクトの初期画面が図 5-18 に示されます。

図 5-18. PuTTY に表示するデータ



```

COM5 - PuTTY
-----
CY15FRAMKIT-001 Kit - PSoC4 SPI F-RAM Example Project
-----
SPI F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x01  0x02  0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0
0x0B  0x0C  0x0D  0x0E  0x0F  0x10
Data read from the address 0x3456 =
0x01  0x02  0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0
0x0B  0x0C  0x0D  0x0E  0x0F  0x10

Status Register Data Write = 0x08
Status Register Data Read = 0x08

Status Register Data Write = 0x00

PSoC4 SPI F-RAM Example Project - PASS

```

## 5.3 プロジェクト: PSoC 4 F-RAM SPI

### 5.3.1 プロジェクトの説明

このサンプル プロジェクトが CY8CKIT-042 PSoC 4 Pioneer Kit で実行され、SPI F-RAM コンポーネント、および F-RAM メモリ/ステータス レジスタの読み出しと書き込みをするための API を提供します。サンプル プロジェクトでは、F-RAM への書き込みと、F-RAM からの読み出しを行って、PSoC 4 での RGB LED (D9) を青色で点灯することで動作が成功したことを示します。出力も UART を介してハイパーターミナルに表示されます。

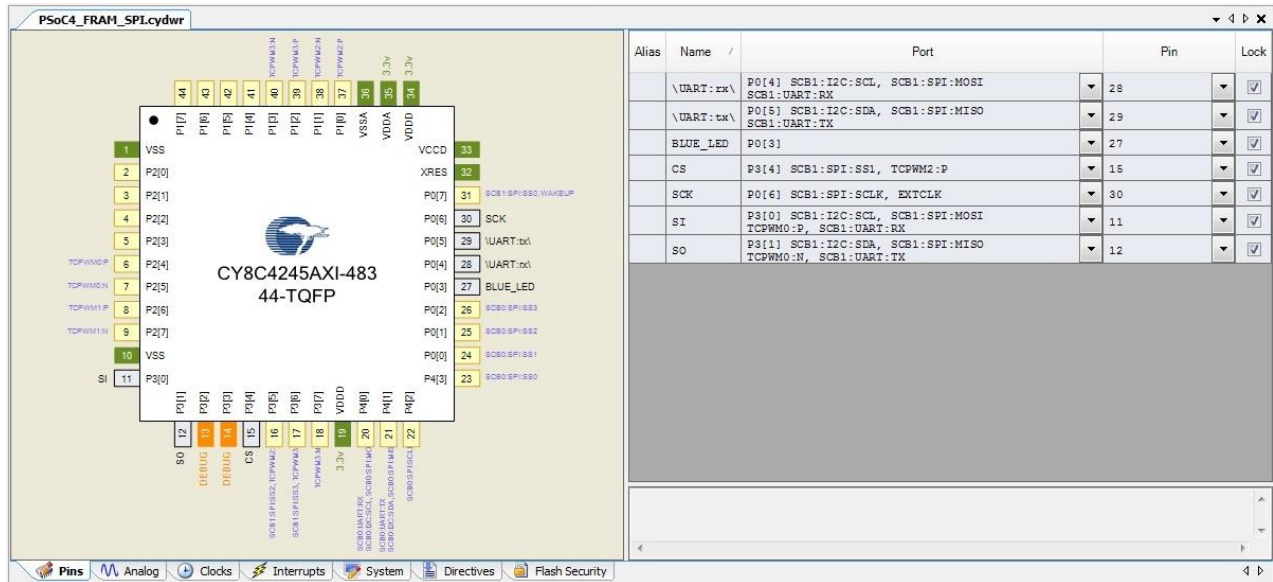
### 5.3.2 ハードウェア接続

UART 設定については、5.2 を参照してください。その他の接続はすべて基板上にハードワイヤ接続されます。ワークスペース エクスプローラで、*PSoC4\_FRAM\_SPI.cydwr* を開いて、適切なピンを選択します。

表 5-1. PSoC 4 F-RAM SPI I/O ピン配置

F-RAM I/O	PSoC 4 のピン
CS#	P3_4
SI	P3_0
SO	P3_1
SCK	P0_6

図 5-19. PSoC 4 F-RAM SPI サンプル プロジェクトのピン設定



### 5.3.3 ファームウェア フロー

以下の手順はサンプル プロジェクトのファームウェア フローを提供します。

- F-RAM SPI および USB-UART ブロックを初期化します。
- ディスプレイ (ハイパーターミナル/PuTTY) からの「Enter」キーを待機します。
- EXAMPLE\_DATA\_BYTE (0xA5) を EXAMPLE\_ADDR\_1 (0x2000) アドレスに書き込みます。
- EXAMPLE\_ADDR\_1 (0x2000) アドレスから 1 バイトを読み出して、fram\_rd\_byte に格納します。
- EXAMPLE\_DATA\_BYTE (0xA5) を EXAMPLE\_ADDR\_1 (0x2000) アドレスに書き込みます。
- EXAMPLE\_ADDR\_1 (0x2000) アドレスから 1 バイトを読み出して、fram\_rd\_byte 変数に格納します。
- data\_bytes アレイ (1、2、3、...、16) からのデータの 16 バイトを EXAMPLE\_ADDR\_2 (0x3456) アドレスに書き込みます。
- 読み出し関数で EXAMPLE\_ADDR\_2 (0x3456) アドレスから 16 バイトのデータを読み出します。
- EXAMPLE\_STS\_REG\_VALUE (0x08) をステータス レジスタに書き込みます。
- ステータス レジスタを読み出して、その値を「statug\_reg」変数に格納します。
- 読み出したデータを書き込んだデータと比較します。比較値が一致した場合、青色 LED が点灯します。
- ステータス レジスタをクリアします。

### 5.3.4 出力の検証

データ出力は図 5-20 および図 5-22 に示されます。

図 5-20. PSoC 4 F-RAM SPI サンプル プロジェクトの出力 - ユーザーが押した Enter Key の待機中

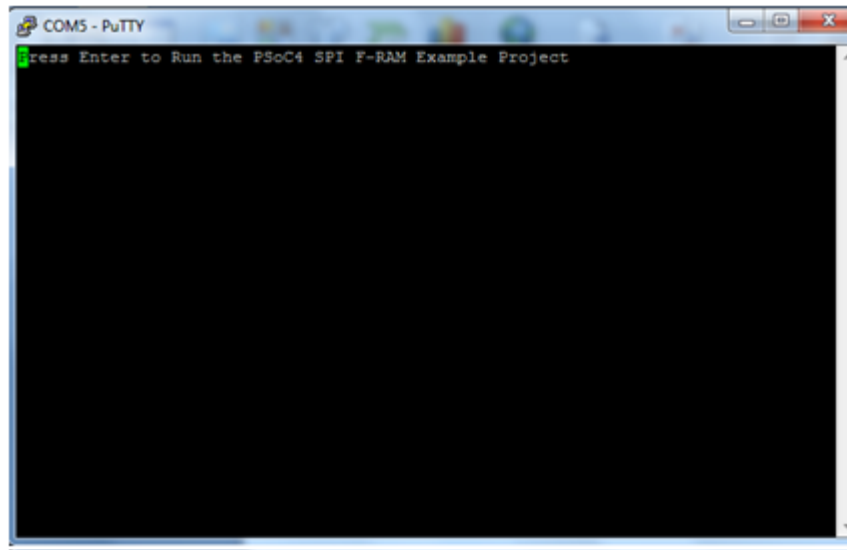
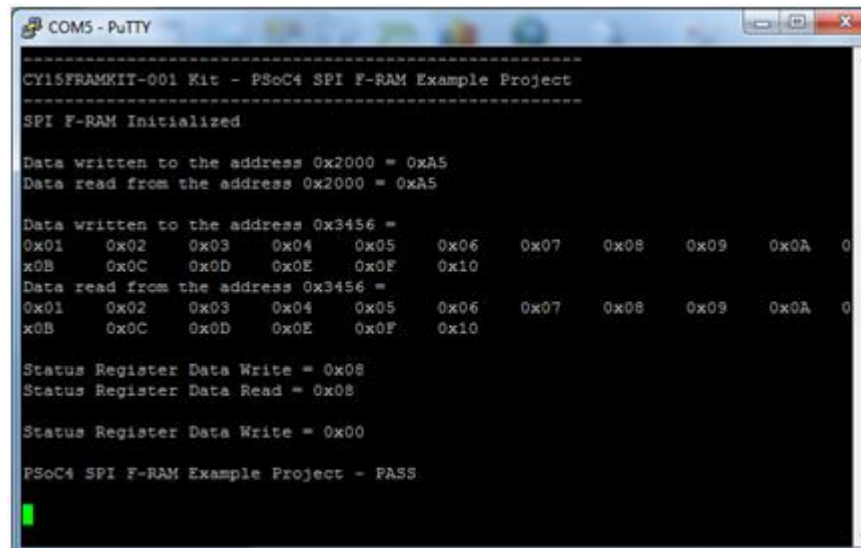


図 5-21. PSoC 4 F-RAM SPI サンプル プロジェクトの出力データ



### 5.3.5 プロジェクトの変更

プロジェクトの **Source Files** フォルダ下の **main.c** ファイルを開いて、図 5-22 に示す **EXAMPLE\_ADDR\_1**、**EXAMPLE\_ADDR\_2** と **EXAMPLE\_DATA\_BYTE** 変数を設定することでアドレスとデータ バイト値を変更します。例えば、**0x2000** を **0x1200** に、**0x3456** を **0x1600** に、**0xA5** を **0x5A** に変更します。

図 5-22. main.c ソース ファイルのスナップショット

```

PSoC4_FRAM_SPI.cydwr  main.c
20  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
21  *****
22  #include <device.h>
23  #include <stdio.h>
24  #include "FRAM_SPI.h"
25
26  // Size of data buffer
27  #define BUFFER_SIZE      (16u)
28
29  #define STRING_SIZE      (50u)
30
31  // Example F-RAM Address
32  #define EXAMPLE_ADDR_1   (0x2000)
33
34  // Example F-RAM Address
35  #define EXAMPLE_ADDR_2   (0x3456)
36
37  // Example F-RAM Data
38  #define EXAMPLE_DATA_BYTE (0xa5)
39

```

また、ユーザーは図 5-23 に示すバースト書き込みデータを変更することもできます。例えば、83 行目での **i+1** を **i+3** に変更すると、0x03~0x13 の入力データが初期化されます。

図 5-23. 入力データ バッファ初期化用の main.c ソース ファイルのスナップショット

```

79  // Initialization of variables
80  for(i = 0; i < BUFFER_SIZE; i++)
81  {
82      fram_rd data[i] = 0;
83      data_bytes[i] = i+1;
84  }

```

上記の変更をした後のプロジェクトをビルドして、Pioneer Kit をプログラムします。図 5-24 に示しているように、**0x1200** アドレスに書き込んだ **0x5a** データ、および **0x1600** アドレスから書き込まれた **0x03 – 0x13** データが出力に表示されます。次の節に説明する API は異なるアプリケーションをビルドするために使用できます。

図 5-24. 変更した SPI F-RAM プロジェクトの出力

```

COM5 - PuTTY
CY15FRAMKIT-001 Kit - PSoC4 SPI F-RAM Example Project
SPI F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
0xD  0x0E  0x0F  0x10  0x11  0x12
Data read from the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
0xD  0x0E  0x0F  0x10  0x11  0x12

Status Register Data Write = 0x08
Status Register Data Read = 0x08

Status Register Data Write = 0x00
PSoC4 SPI F-RAM Example Project - PASS

```

### 5.3.6 API

#### 5.3.6.1 void FRAM\_SPI\_Init(void)

説明: SPI ブロックを初期化する。

パラメーター: なし

返回值: なし

注意事項: なし

#### 5.3.6.2 uint8 FRAM\_SPI\_Write(uint32 addr、uint8 data\_write)

説明: 1 バイトのデータを F-RAM に書き込む。

パラメーター: uint32 addr: 書き込み用の 32 ビット F-RAM アドレス  
uint8 data\_write: 書き込まれるデータ バイト

返回值: uint8: 0 - 通信エラー  
1 - 成功

注意事項: なし

#### 5.3.6.3 uint8 FRAM\_SPI\_Read(uint32 addr、uint8 \*data\_read)

説明: F-RAM から 1 バイトのデータを呼び出す。

パラメーター: uint32 addr: 読み出し用の 32 ビット F-RAM アドレス  
uint8 \*data\_read: データを格納するための 8 ビットのポインタ変数

返回值: uint8: 0 - 通信エラー  
1 - 成功

注意事項: なし

#### 5.3.6.4 uint8 FRAM\_SPI\_BurstWrite(uint32 addr 、 uint8 \*data\_write\_ptr 、 uint32 total\_data\_count)

説明: F-RAM にデータの total\_data\_count 数を書き込む。

パラメーター: uint32 addr: 書き込み用の 32 ビット F-RAM アドレス  
uint8 \*data\_write\_ptr: 書き込みデータ バイトのアレイへのポインタ  
uint32 total\_data\_count: 書き込みデータ バイト数

返回值: uint8: 0 - 通信エラー  
1 - 成功

注意事項: なし

#### 5.3.6.5 uint8 FRAM\_SPI\_BurstRead(uint32 addr 、 uint8 \*data\_read\_ptr 、 uint32 total\_data\_count )

説明: F-RAM からデータの total\_data\_count 数を読み出す。

パラメーター: uint32 addr: 読み出し用の 32 ビット F-RAM アドレス  
uint8 \*data\_read\_ptr: データ バイトを格納するのに使用するアレイへのポインタ

uint32 total\_data\_count: 書き込みデータ バイト数

戻り値: uint8: 0 – 通信エラー  
1 – 成功

注意事項: なし

### 5.3.6.6 uint8 FRAM\_SPI\_Status\_Reg\_Write (uint8 data\_byte)

説明: F-RAM ステータス レジスタ書き込み。

パラメーター: uint8 data\_byte -> 書き込まれる 1 バイトのステータス レジスタ データ

戻り値: uint8: 0 – 通信エラー  
1 – 成功

注意事項: なし

### 5.3.6.7 uint8 FRAM\_SPI\_Status\_Reg\_Read (uint8 \* status)

説明: F-RAM ステータス レジスタ読み出し。

パラメーター: uint8 \* status : ステータスレジスタに格納する値へのポインタ

戻り値: uint8: 0 – 通信エラー  
1 – 成功

注意事項: なし

## 5.4 プロジェクト: PSoC 4 F-RAM I<sup>2</sup>C

### 5.4.1 プロジェクトの説明

このサンプル プロジェクトが CY8CKIT-042 PSoC 4 Pioneer Kit で実行され、I<sup>2</sup>C F-RAM コンポーネントおよび、F-RAM メモリの読み出しと書き込みをするための API を提供します。サンプル プロジェクトでは、F-RAM への書き込みと、F-RAM からの読み出しを行って、PSoC 4 での RGB LED (D9) を青色で点灯することで動作が成功したことを示します。出力も UART を介してハイパーターミナルに表示されます。

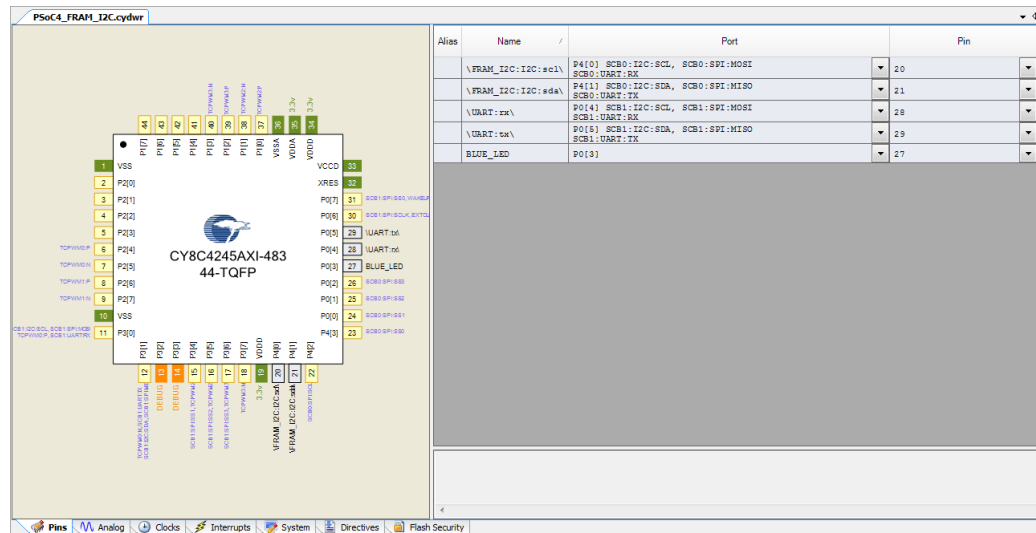
### 5.4.2 ハードウェア接続

UART 設定については、[5.2](#) を参照してください。その他の接続はすべて基板上にハードワイヤ接続されます。ワークスペースエクスプローラで、*PSoC4\_FRAM\_I2C.cydwr*を開いて、適切なピンを選択します。

表 5-2. PSoC 4 F-RAM I<sup>2</sup>C I/O ピン配置

F-RAM I/O	PSoC 4 のピン
SDA	P4_1
SCL	P4_0

図 5-25. PSoC 4 F-RAM I<sup>2</sup>C サンプル プロジェクトのピン設定



### 5.4.3 ファームウェア フロー

以下の手順はサンプル プロジェクトのファームウェア フローを提供します。

- F-RAM I<sup>2</sup>C および USB-UART ブロックを初期化します。
- EXAMPLE\_DATA\_BYTE (0xA5) を EXAMPLE\_ADDR\_1 (0x2000) アドレスに書き込みます。
- EXAMPLE\_ADDR\_1 (0x2000) アドレスから 1 バイトを読み出して、fram\_rd\_byte 変数に格納します。
- data\_bytes アレイ (1、2、3、...、16) からの 16 バイトのデータを EXAMPLE\_ADDR\_2 (0x3456) アドレスに書き込みます。
- ランダム読み出し関数で EXAMPLE\_ADDR\_2 (0x3456) アドレスからデータの 15 バイトを読み出します。
- 現在の読み出し関数で 16 バイト目を読み出します。
- 読み出しデータを書き込みデータと比較し、読み出しに成功すれば、青色 LED が点灯します。

### 5.4.4 出力の検証

読み出しが成功した場合、RGB LED は青色で点灯します。読み出しに失敗した場合、RGB LED は点灯しません。データ出力は図 5-26 および図 5-27 に示されます。

図 5-26. PSoC 4 F-RAM I<sup>2</sup>C サンプル プロジェクトの出力 - ユーザーが押した **Enter Key** の待機中

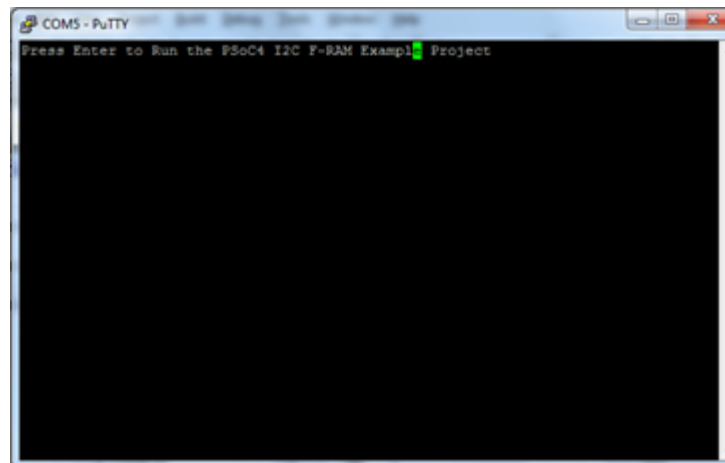
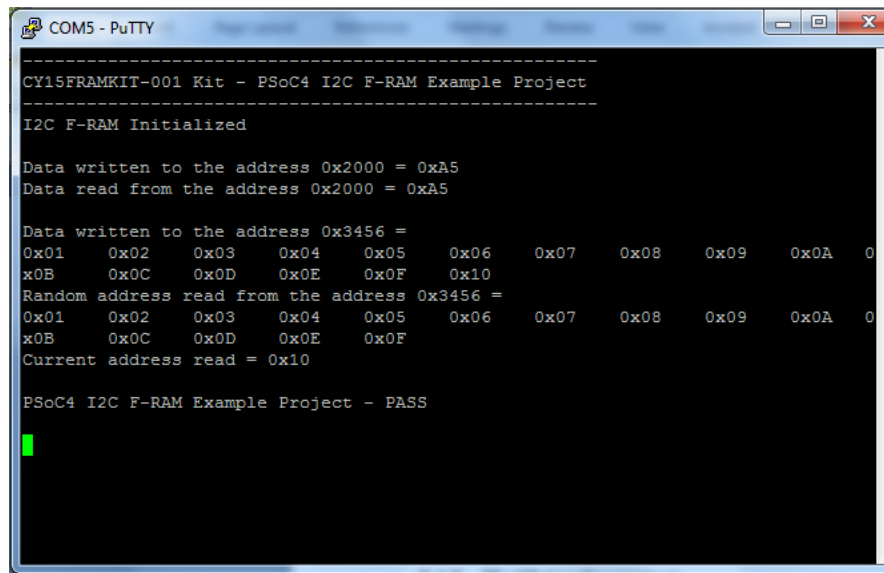




図 5-27. PSoC 4 F-RAM I<sup>2</sup>C サンプル プロジェクトの出力データ



```

COM5 - PuTTY
-----
CY15FRAMKIT-001 Kit - PSoC4 I2C F-RAM Example Project
-----
I2C F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0
x0B 0x0C 0x0D 0x0E 0x0F 0x10
Random address read from the address 0x3456 =
0x01 0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09 0x0A 0
x0B 0x0C 0x0D 0x0E 0x0F
Current address read = 0x10

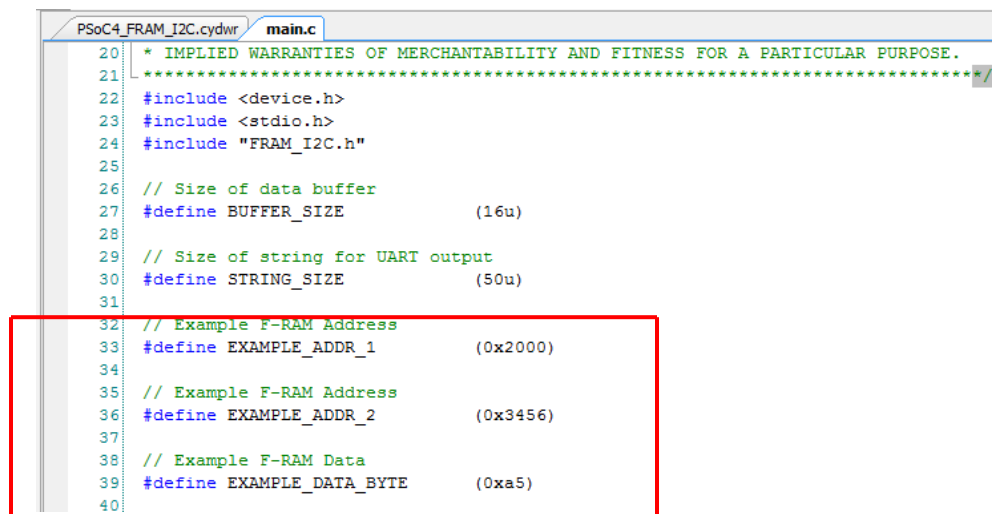
PSoC4 I2C F-RAM Example Project - PASS

```

### 5.4.5 プロジェクトの変更

プロジェクトの **Source Files** フォルダ下の **main.c** ファイルを開いて、図 5-28 に示す **EXAMPLE\_ADDR\_1**、**EXAMPLE\_ADDR\_2** および **EXAMPLE\_DATA\_BYTE** の値を変更します。例えば、**0x2000** を **0x1200** に、**0x3456** を **0x1600** に、**0xa5** を **0x5a** に変更します。

図 5-28. main.c ソース ファイルのスナップショット



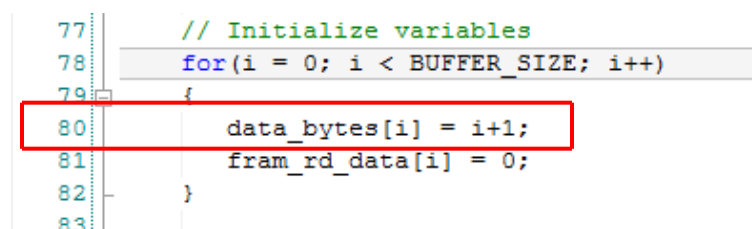
```

PSoC4_FRAM_I2C.cydwr  main.c
20  * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
21  *****
22  #include <device.h>
23  #include <stdio.h>
24  #include "FRAM_I2C.h"
25
26  // Size of data buffer
27  #define BUFFER_SIZE          (16u)
28
29  // Size of string for UART output
30  #define STRING_SIZE          (50u)
31
32  // Example F-RAM Address
33  #define EXAMPLE_ADDR_1      (0x2000)
34
35  // Example F-RAM Address
36  #define EXAMPLE_ADDR_2      (0x3456)
37
38  // Example F-RAM Data
39  #define EXAMPLE_DATA_BYTE   (0xa5)
40

```

また、ユーザーは図 5-29 に示すバースト書き込みデータを変更することもできます。例えば、80 行目での **i+1** を **i+3** に変更すると、0x03~0x13 の入力データが初期化されます。

図 5-29. 入力データ バッファ初期化用の main.c ソース ファイルのスナップショット



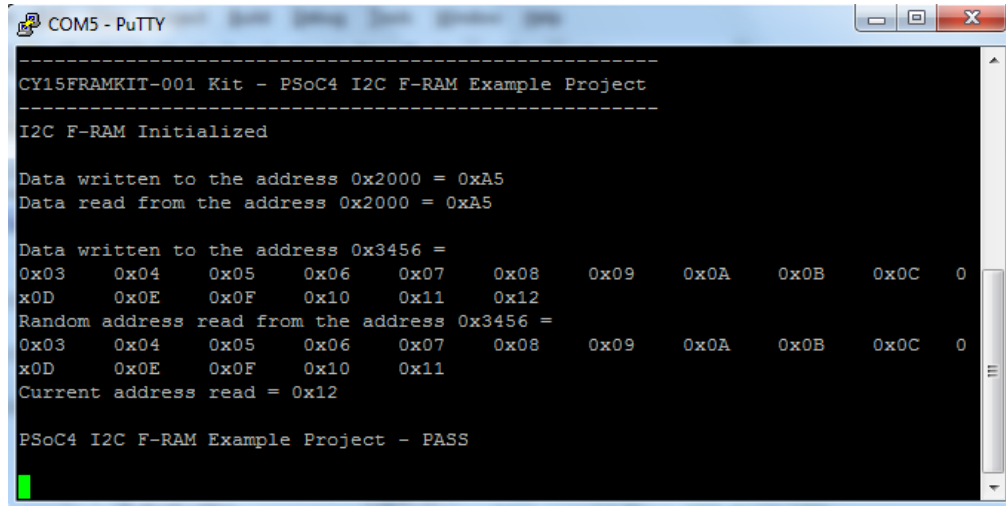
```

77  // Initialize variables
78  for(i = 0; i < BUFFER_SIZE; i++)
79  {
80  data_bytes[i] = i+1;
81  fram_rd_data[i] = 0;
82  }
83

```

上記の変更をした後、プロジェクトをビルドしてキットをプログラムします。図 5-30 に示しているように、**0x1200** アドレスに書き込んだ **0x5a** データ、および **0x1600** アドレスから書き込まれた **0x03 – 0x13** データが出力に表示されます。次の節に説明する API は異なるアプリケーションをビルドするために使用できます。

図 5-30. 変更した I<sup>2</sup>C F-RAM プロジェクトの出力



```

COM5 - PuTTY
-----
CY15FRAMKIT-001 Kit - PSoC4 I2C F-RAM Example Project
-----
I2C F-RAM Initialized

Data written to the address 0x2000 = 0xA5
Data read from the address 0x2000 = 0xA5

Data written to the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
x0D  0x0E  0x0F  0x10  0x11  0x12
Random address read from the address 0x3456 =
0x03  0x04  0x05  0x06  0x07  0x08  0x09  0x0A  0x0B  0x0C  0
x0D  0x0E  0x0F  0x10  0x11
Current address read = 0x12

PSoC4 I2C F-RAM Example Project - PASS

```

## 5.4.6 API

以下の API は PSoC 4 F-RAM I<sup>2</sup>C 用に定義されます。これらの API は Arduino 用のものと同じです。

### 5.4.6.1 void FRAM\_I2C\_Init(void)

説明: I2C ブロックを初期化する。

パラメーター: なし

返り値: なし

注意事項: なし

### 5.4.6.2 uint32 FRAM\_I2C\_Write (uint8 slave\_id、uint32 addr、uint8 \*data\_write\_ptr、uint32 total\_data\_count)

説明: slave\_id に指定される F-RAM にデータの total\_data\_count 数を書き込む。

パラメーター: uint8: 7 ビットのスレーブ ID

uint32 addr: 書き込み用の 32 ビット F-RAM アドレス

uint8 \*data\_write\_ptr: 書き込みデータ バイトのアレイへのポインタ

uint32 total\_data\_count: 書き込みデータ バイト数

返り値: PSoC 4 の場合、

uint32: エラー ステータス

FRAM\_I2C\_MSTR\_NO\_ERROR -- (0x00u)

FRAM\_I2C\_MSTR\_ERR\_ARB\_LOST -- (0x01u)

FRAM\_I2C\_MSTR\_ERR\_LB\_NAK -- (0x02u)

FRAM\_I2C\_MSTR\_NOT\_READY -- (0x04u)

```

FRAM_I2C_MSTR_BUS_BUSY      -- (0x08u)
FRAM_I2C_MSTR_ERR_ABORT_START -- (0x10u)
FRAM_I2C_MSTR_ERR_BUS_ERR    -- (0x100u)
FRAM_I2C_MSTR_ERR            -- (0xFFu)

```

Arduino UNO の場合、

uint32: エラー ステータス

成功 -- (0x00u)

データが長くて送信バッファに入り切らない -- (0x01u)

アドレスの送信時に NACK を受信 -- (0x02u)

データの送信時に NACK を受信 -- (0x03u)

その他のエラー -- (0x04u)

注意事項: なし

#### 5.4.6.3 uint32 FRAM\_I2C\_Current\_Read (uint8 slave\_id, uint8 \*data\_read\_ptr, uint32 total\_data\_count)

説明: slave\_id に指定される F-RAM の現在のアドレスからデータの total\_data\_count 数を読み出す。

パラメーター: uint8: 7 ビットのスレーブ ID

uint8 \*data\_read\_ptr: データ バイトを格納するのに使用するアレイへのポインタ

uint32 total\_data\_count: 書き込みデータ バイト数

返り値: PSoC 4 の場合、

uint32: エラー ステータス

```

FRAM_I2C_MSTR_NO_ERROR      -- (0x00u)
FRAM_I2C_MSTR_ERR_ARB_LOST  -- (0x01u)
FRAM_I2C_MSTR_ERR_LB_NAK    -- (0x02u)
FRAM_I2C_MSTR_NOT_READY     -- (0x04u)
FRAM_I2C_MSTR_BUS_BUSY      -- (0x08u)
FRAM_I2C_MSTR_ERR_ABORT_START -- (0x10u)
FRAM_I2C_MSTR_ERR_BUS_ERR    -- (0x100u)
FRAM_I2C_MSTR_ERR            -- (0xFFu)

```

Arduino UNO の場合、読み出されたバイト数を返す。

注意事項: なし

#### 5.4.6.4 uint32 FRAM\_I2C\_Random\_Read (uint8 slave\_id, uint32 addr, uint8 \*data\_read\_ptr, uint32 total\_data\_count)

説明: slave\_id に指定される F-RAM からデータの total\_data\_count 数を読み出す。

パラメーター: uint8: 7 ビットのスレーブ ID

uint32 addr: 読み出し用の 32 ビット F-RAM アドレス

uint8 \*data\_read\_ptr: データ バイトを格納するのに使用するアレイへのポインタ

uint32 total\_data\_count: 書き込みデータ バイト数

戻り値: PSoC 4 の場合、

uint32: エラー ステータス

FRAM_I2C_MSTR_NO_ERROR	-- (0x00u)
FRAM_I2C_MSTR_ERR_ARB_LOST	-- (0x01u)
FRAM_I2C_MSTR_ERR_LB_NAK	-- (0x02u)
FRAM_I2C_MSTR_NOT_READY	-- (0x04u)
FRAM_I2C_MSTR_BUS_BUSY	-- (0x08u)
FRAM_I2C_MSTR_ERR_ABORT_START	-- (0x10u)
FRAM_I2C_MSTR_ERR_BUS_ERR	-- (0x100u)
FRAM_I2C_MSTR_ERR	-- (0xFFu)

Arduino UNO の場合、読み出されたバイト数を返す。

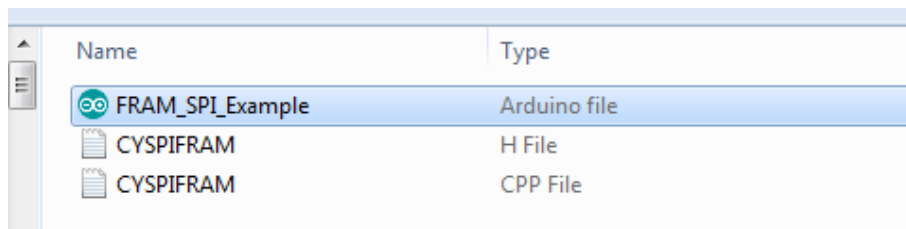
注意事項: なし

## 5.5 プログラミング Arduino UNO キット

Arduino サンプル プロジェクトをプログラムするには以下の手順に従ってください (この手順では Arduino\_FRAM\_SPI プロジェクトが例として使われます)。

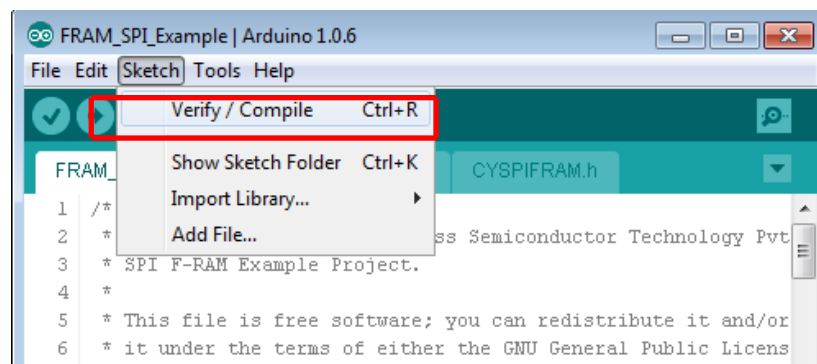
1. Arduino IDE をインストールします。Arduino のウェブページから Arduino IDE ソフトウェアをダウンロードできます:  
<http://arduino.cc/en/Main/Software>。
2. 「Arduino\_FRAM\_SPI」プロジェクト フォルダを開き、**FRAM\_SPI\_Example.ino** ファイルをダブルクリックします。

図 5-31. Arduino プロジェクト ファイルを開く



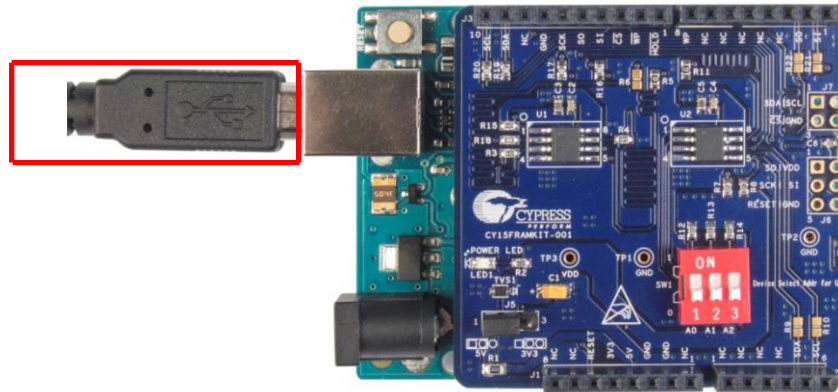
3. コード例をコンパイルします: **Sketch > Verify / Compile**。正常なコンパイルの終了時に、IDE は「Done compiling.」ステータスを表示し、2 進数のスケッチを生成します。

図 5-32. Arduino プロジェクトをコンパイル



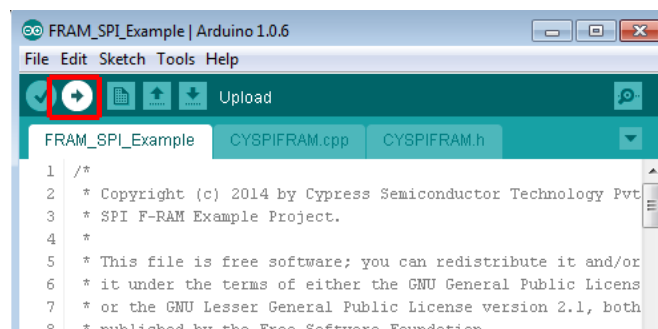
4. Arduino UNO キットをプログラムするには、図 5-33 に示すように、Arduino UNO 互換の USB ケーブルを使用して USB ポートを介して PC を Arduino 基板に接続します。Arduino UNO キットが 5V で動作するため、5V の電源を選択するために CY15FRAMKIT-001 上の電源選択ジャンパ (J5) をピン 1 とピン 2 に配置することを確実に行ってください。

図 5-33. Arduino UNO R3 キットに取り付けた CY15FRAMKIT-001



5. Upload ボタンをクリックして 2 進数のスケッチ ファイルをアップロードします。

図 5-34. Arduino UNO R3 キットのプログラミング



## 5.6 プロジェクト: Arduino F-RAM SPI

### 5.6.1 プロジェクトの説明

このサンプル プロジェクトは Arduino UNO 基板で実行され、F-RAM メモリ/ステータス レジスタを読み出すおよび書き込むための SPI API を提供します。サンプル プロジェクトは F-RAM に書き込み、F-RAM から読み出し、結果を直列出力ウィンドウに表示します。

### 5.6.2 ハードウェア接続

すべての接続は基板上に配線がつながため、このプロジェクトには追加のハードウェア接続が不要です。Arduino SPI ライブラリはデフォルトで次のピンを使用します。

表 5-3. Arduino F-RAM SPI I/O ピン配置

F-RAM I/O	Arduino ピン
CS#	D10
SI	D11
SO	D12
SCK	D13

### 5.6.3 ファームウェア フロー

以下の手順はサンプル プロジェクトのファームウェア フローを提供します。

- F-RAM SPIを初期化します。
- EXAMPLE\_DATA\_BYTE (0xA5) を EXAMPLE\_ADDR\_1 (0x2000) アドレスに書き込みます。直列出力に書き込んだデータを表示します。
- EXAMPLE\_ADDR\_1 アドレスから 1 バイトを読み出します。F-RAM SPIを初期化します。
- EXAMPLE\_DATA\_BYTE (0xA5) を EXAMPLE\_ADDR\_1 (0x2000) アドレスに書き込みます。直列出力に書き込んだデータを表示します。
- EXAMPLE\_ADDR\_1 (0x2000) アドレスから 1 バイトを読み出して、fram\_rd\_byte 変数に格納します。直列出力に読み出したデータを表示します。
- 書き込んだデータと読み出したデータを比較して、結果 (PASS/FAIL) を直列出力に表示します。
- data\_bytes アレイ (1、2、3、...、16) からの 16 バイトのデータを EXAMPLE\_ADDR\_2 (0x3456) アドレスに書き込みます。
- 読み出し関数で EXAMPLE\_ADDR\_2 (0x3456) アドレスから 16 バイトのデータを読み出します。
- 書き込んだデータと読み出したデータを比較して、結果 (PASS/FAIL) を直列出力に表示します。
- EXAMPLE\_STS\_REG\_VALUE (0x08) をステータス レジスタに書き込みます。
- ステータス レジスタを読み出して、その値を「statug\_reg」変数に格納します。
- 書き込んだデータと読み出したデータを比較して、結果 (PASS/FAIL) を直列出力に表示します。
- ステータス レジスタをクリアします。

### 5.6.4 出力の検証

Tools >Serial Port を開き、図 5-35 に示すように、通信に適切な COM ポートを選択します。

図 5-35. Arduino IDE 内の直列ポート

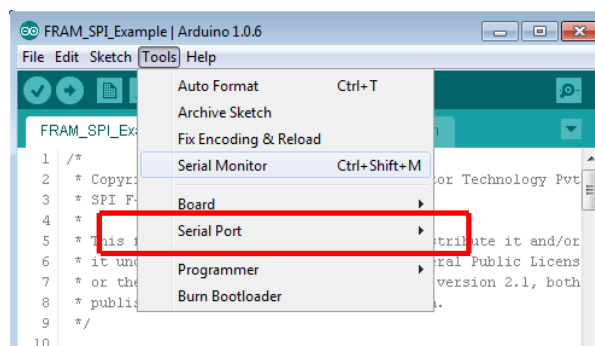


図 5-36 と図 5-37 に示すように、**Tools > Serial Monitor** を開き、直列出力を表示させます。

図 5-36. Arduino IDE 内の直列モニター

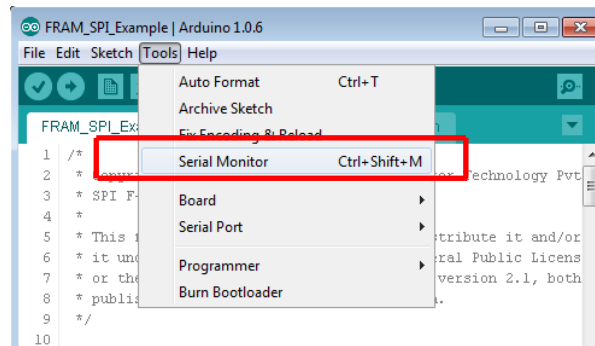
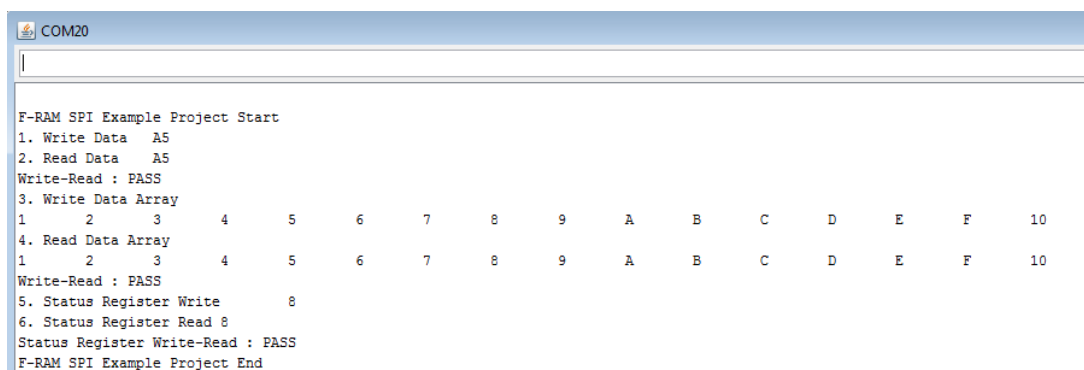


図 5-37. SPI F-RAM プロジェクト用のモニター出力



## 5.6.5 API

### 5.6.5.1 void FRAM\_SPI\_Init()

説明: SPI ライブラリを初期化する。

パラメーター: なし

返回值: なし

注意事項: なし

### 5.6.5.2 void FRAM\_SPI\_Write (uint32 addr、uint8 data\_write)

説明: 1 バイトのデータを F-RAM に書き込む。

パラメーター: uint32 addr: 書き込み用の 32 ビット F-RAM アドレス  
uint8 data\_write: 書き込まれるデータ バイト

返回值: なし

注意事項: なし

### 5.6.5.3 uint8 FRAM\_SPI\_Read(uint32 addr)

説明: F-RAM から 1 バイトのデータを呼び出す。

パラメーター: uint32 addr: 読み出し用の 32 ビット F-RAM アドレス

返回值: uint8: 読み出されるデータ

注意事項: なし



#### 5.6.5.4 void FRAM\_SPI\_BurstWrite(uint32 addr, uint8 \*data\_write\_ptr, uint32 total\_data\_count)

説明: F-RAM にデータの total\_data\_count 数を書き込む。  
パラメーター: uint32 addr: 書き込み用の 32 ビット F-RAM アドレス  
uint8 \*data\_write\_ptr: 書き込まれるデータ バイトのアレイへのポインタ  
uint32 total\_data\_count: 書き込まれるデータ バイト数  
返り値: なし  
注意事項: なし

#### 5.6.5.5 void FRAM\_SPI\_BurstRead(uint32 addr, uint8 \*data\_read\_ptr, uint32 total\_data\_count )

説明: F-RAM からデータの total\_data\_count 数を読み出す。  
パラメーター: uint32 addr : 読み出し用の 32 ビット F-RAM アドレス  
uint8 \*data\_read\_ptr: データ バイトを格納するのに使用するアレイへのポインタ  
uint32 total\_data\_count: 書き込みデータ バイト数  
返り値: なし  
注意事項: なし

#### 5.6.5.6 void FRAM\_SPI\_Status\_Reg\_Write (uint8 data\_byte)

説明: F-RAM ステータス レジスタ書き込み。  
パラメーター: uint8 data\_byte -> 書き込まれる 1 バイトのステータス レジスタ データ  
返り値: なし  
注意事項: なし

#### 5.6.5.7 uint8 FRAM\_SPI\_Status\_Reg\_Read (void)

説明: F-RAM ステータス レジスタ読み出し。  
パラメーター: なし  
返り値: uint8: ステータス レジスタ データ 1 バイト  
注意事項: なし

## 5.7 プロジェクト: Arduino F-RAM I<sup>2</sup>C

### 5.7.1 プロジェクトの説明

このサンプル プロジェクトは Arduino UNO 基板で実行され、F-RAM メモリを読み出すおよび書き込むための I<sup>2</sup>C API を提供します。サンプル プロジェクトは F-RAM に書き込み、F-RAM から読み出し、結果を直列出力ウィンドウに表示します。

### 5.7.2 ハードウェア接続

すべての接続は基板上に配線がつながため、このプロジェクトには追加のハードウェア接続が不要です。Arduino WIRE ライブラリ (I<sup>2</sup>C) はデフォルトで次のピンを使用します。

表 5-4. Arduino F-RAM I<sup>2</sup>C I/O ピン配置

F-RAM I/O	Arduino ピン
SDA	A4
SCL	A5

### 5.7.3 ファームウェア フロー

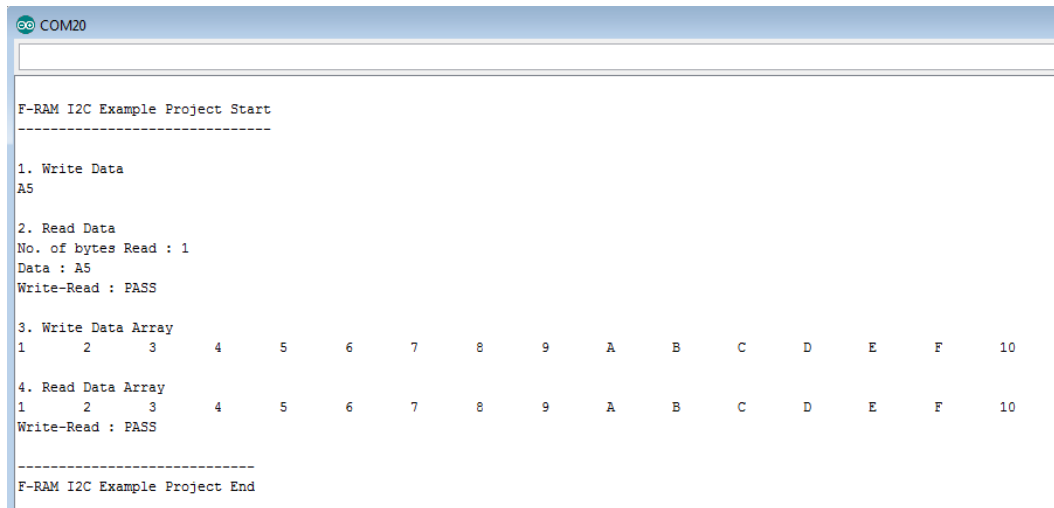
以下の手順はサンプル プロジェクトのファームウェア フローを提供します。

- F-RAM I<sup>2</sup>C を初期化して起動遅延を提供します。
- EXAMPLE\_DATA\_BYTE (0xA5) を EXAMPLE\_ADDR\_1 (0x2000) アドレスに書き込みます。
- EXAMPLE\_ADDR\_1 (0x2000) アドレスから 1 バイトを読み出して、fram\_rd\_byte 変数に格納します。直列出力に読み出したデータを表示します。
- 書き込んだデータと読み出したデータを比較して、結果 (PASS/FAIL) を直列出力に表示します。
- data\_bytes アレイ (1、2、3、...、16) からの 16 バイトのデータを EXAMPLE\_ADDR\_2 (0x3456) アドレスに書き込みます。
- ランダム読み出し関数で EXAMPLE\_ADDR\_2 (0x3456) アドレスからデータの 15 バイトを読み出します。
- 現在の読み出し関数で 16 バイト目を読み出します。
- 書き込んだデータと読み出したデータを比較して、結果 (PASS/FAIL) を直列出力に表示します。

### 5.7.4 出力の検証

図 5-38 に示すように、**Tools→Serial Monitor** を開き、直列出力を表示させます。

図 5-38. I<sup>2</sup>C F-RAM プロジェクト用のモニター出力



```

COM20

F-RAM I2C Example Project Start
-----
1. Write Data
A5

2. Read Data
No. of bytes Read : 1
Data : A5
Write-Read : PASS

3. Write Data Array
1      2      3      4      5      6      7      8      9      A      B      C      D      E      F      10
1      2      3      4      5      6      7      8      9      A      B      C      D      E      F      10

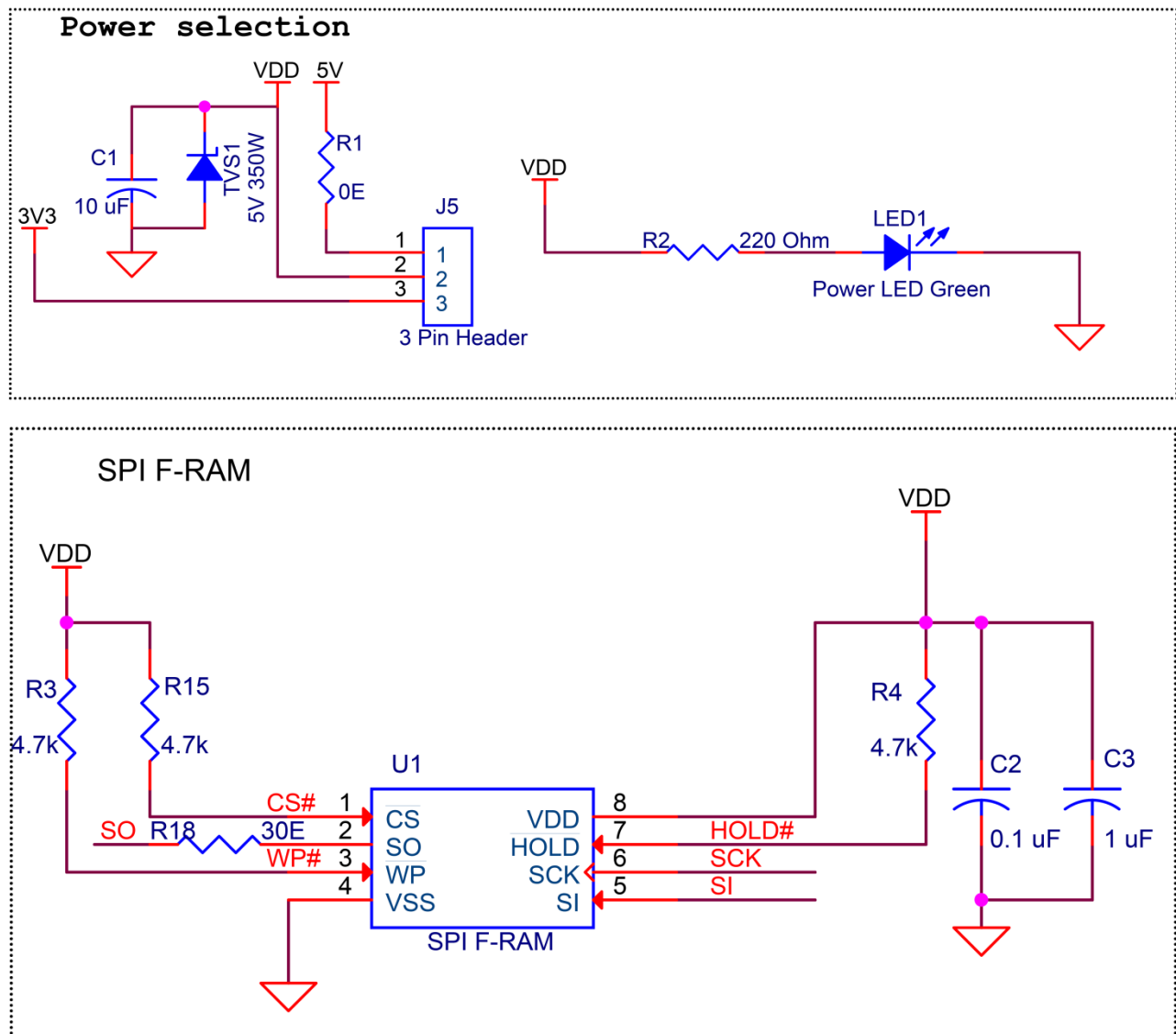
4. Read Data Array
1      2      3      4      5      6      7      8      9      A      B      C      D      E      F      10
Write-Read : PASS
-----
F-RAM I2C Example Project End

```

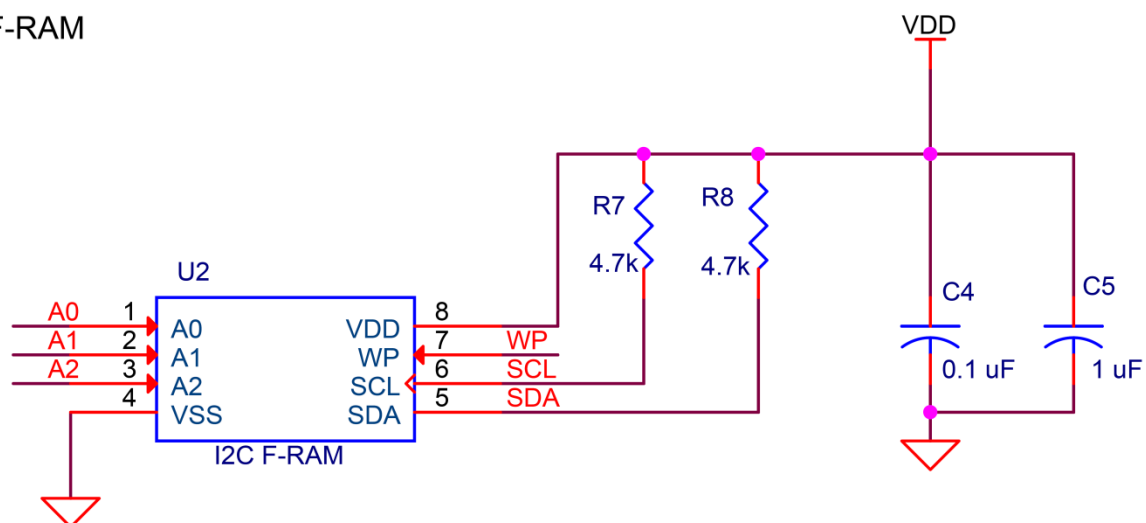
### 5.7.5 API

PSoC 4 I2C F-RAM の API 説明と同様です。5.4.6 項を参照してください。

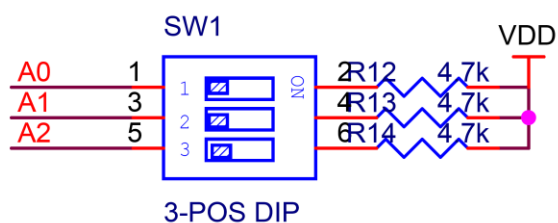
## A.1 CY15FRAMKIT-001 回路図



## I2C F-RAM

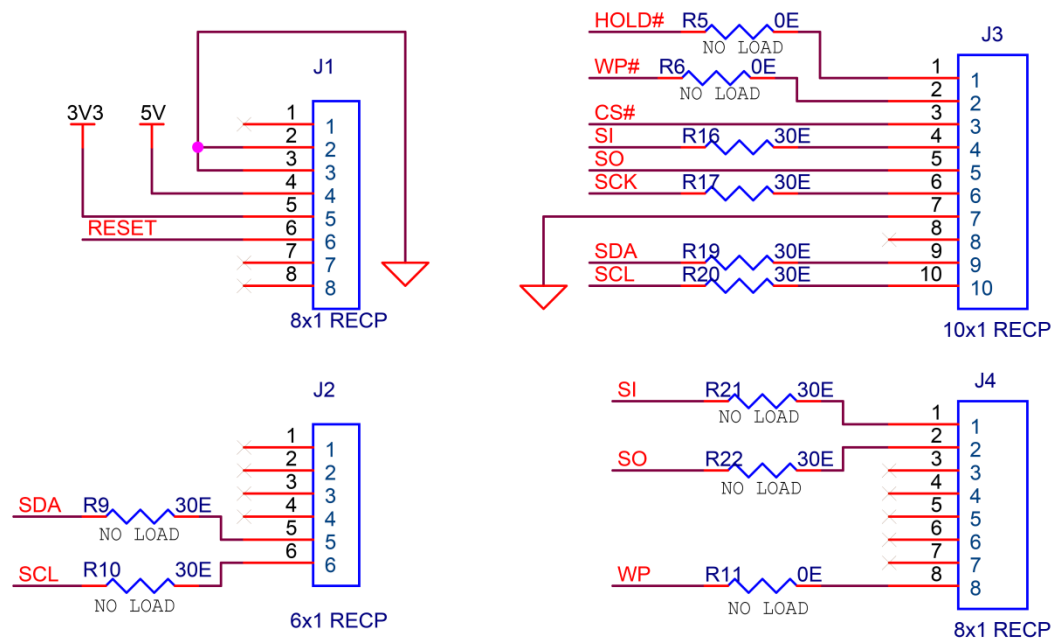


## I2C F-RAM Device Select Address

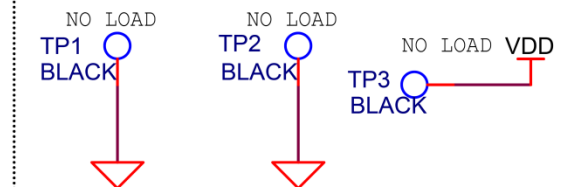
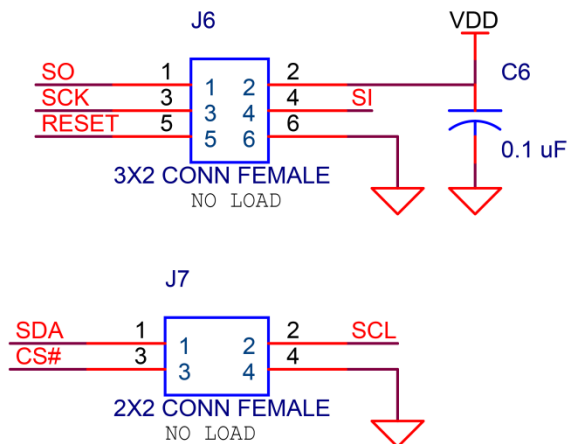


By default, A0, A1, A2 are pulled down internally

### (J1-J4) Arduino Compatible Headers



### Debug Header



## A.2 ピン割り当ての表

本節では、ヘッダのピン マップとそれらの使用方法を記述します。

### Arduino 互換のヘッダ (J1, J2, J3, J4)

J1			
ピン	Arduino 信号	PSoC 4 Pioneer Kit 信号	CY15FRAMKIT-001 キット信号
J1_1	VIN	VIN	NC
J1_2	GND	GND	GND
J1_3	GND	GND	GND
J1_4	5V	V5.0	5V
J1_5	3.3V	V3.3	3V3
J1_6	RESET#	RESET#	RESET#
J1_7	IOREF	P4_VDD	NC
J1_8	NC	NC	NC

J2			
ピン	Arduino 信号	PSoC 4 Pioneer Kit 信号	CY15FRAMKIT-001 キット信号
J2_1	A0	P2[0]	NC
J2_2	A1	P2[1]	NC
J2_3	A2	P2[2]	NC
J2_4	A3	P2[3]	NC
J2_5	A4/SDA (I <sup>2</sup> C)	P2[4]	SDA
J2_6	A5/SCL (I <sup>2</sup> C)	P2[5]	SCL

J3			
ピン	Arduino 信号	PSoC 4 Pioneer Kit 信号	CY15FRAMKIT-001 キット信号
J3_1	D8	P2[6]	HOLD# (SPI F-RAM)
J3_2	D9 (PWM)	P3[6]	WP# (SPI F-RAM)
J3_3	D10 (PWM/SS)	P3[4] / CS (SCB1)	CS# (SPI F-RAM)
J3_4	D11 (PWM / MOSI)	P3[0] / MOSI (SCB1)	SI
J3_5	D12 (MISO)	P3[1] / MISO (SCB1)	SO
J3_6	D13 (SCK)	P0[6] / SCK (SCB1)	SCK
J3_7	GND	GND	GND
J3_8	AREF	P1[7]	NC
J3_9	SDA	P4[1] / SDA (SCB0)	SDA
J3_10	SCL	P4[0] / SCL (SCB0)	SCL



J4			
ピン	Arduino 信号	PSoC 4 Pioneer Kit 信号	CY15FRAMKIT-001 キット信号
J4_1	D0 (UART RX)	P0[4] / SCL (SCB1)	NC
J4_2	D1 (UART TX)	P0[5] / SDA (SCB1)	NC
J4_3	D2	P0[7]	NC
J4_4	D3 (PWM)	P3[7]	NC
J4_5	D4	P0[0]	NC
J4_6	D5 (PWM)	P3[5]	NC
J4_7	D6 (PWM)	P1[0]	NC
J4_8	D7	P2[7]	WP (I <sup>2</sup> C F-RAM)

### A.3 デバッグ ヘッダ I/O

J6	
ピン	デバッグ信号
J6_1	SO
J6_2	VDD
J6_3	SCK
J6_4	SI
J6_5	RESET#
J6_6	GND

J7	
ピン	デバッグ信号
J7_1	SDA
J7_2	SCL
J7_3	CS#
J7_4	GND

## A.4 0Ω 抵抗と負荷なしの使用法

単位	抵抗	使用法
SPI F-RAM デバイス	R5	コントローラーから SPI HOLD#ピンの制御用に 0Ω 抵抗をはんだ付けする
SPI F-RAM デバイス	R6	コントローラーから SPI HOLD#ピンの制御用に 0Ω 抵抗をはんだ付けする
I <sup>2</sup> C F-RAM デバイス	R11	コントローラーから I <sup>2</sup> C WP#ピンを制御する用に 0Ω 抵抗をはんだ付けする

## A.5 30Ω 抵抗と負荷なしの使用法

単位	抵抗	使用法
I <sup>2</sup> C 通信	R9	Arduino UNO R2 基板上の I <sup>2</sup> C 通信用に SDA アクセスするために、30Ω 抵抗をはんだ付けする
I <sup>2</sup> C 通信	R10	Arduino UNO R2 基板上の I <sup>2</sup> C 通信用に SDA アクセスするために、30Ω 抵抗をはんだ付けする
SPI 通信	R21	J4_1 を SPI SI ピンとして使用するために 30Ω 抵抗をはんだ付けする。これにより、J3_4 上の SI が開放され、CY8CKIT-042 Pioneer Kit での I <sup>2</sup> C-USB ブリッジ通信に使用されることが可能になる
SPI 通信	R22	J4_2 を SPI SO ピンとして使用するために 30Ω 抵抗をはんだ付けする。これにより、J3_5 上の SO が開放され、CY8CKIT-042 Pioneer Kit での I <sup>2</sup> C-USB ブリッジ通信に使用されることが可能になる

## A.6 部品表 (BOM)

No.	数量	リファレンス	値	説明	メーカー	メーカー部品番号
1				プリント基板、52.07mm X 53.59mm、2 層、ENIG めっき、高 Tg、青色はんだ付けマスク、白色シルクスクリーン	サイプレス	
2	1	C1	10uF	CAP TANT 10UF 10V 20% 1206	AVX Corporation	TPSA106M010R1800
3	3	C2、C4、C6	0.1uF	CAP CER 0.1UF 50V Y5V 0603	Murata Electronics	GRM188R71C104KA01D
4	2	C3、C5	1uF	CAP CERAMIC 1.0UF 25V X5R 0603 10%	太陽誘電	TMK107BJ105KA-T
5	2	J1、J4	8x1 RECP	CONN RCPT .100" 8PS R/A SGL TIN	Samtec Inc	SSQ-108-03-T-S
6	1	J2	6x1 RECP	CONN RCPT .100" 6POS SNGL TIN	Samtec Inc	SSQ-106-03-T-S
7	1	J3	10x1 RECP	CONN RCPT .100" 10POS SNGL TIN	Samtec Inc	SSQ-110-03-T-S
8	1	J5	3ピン ヘッダ	CONN HEADER VERT SGL 3POS GOLD	3M	961103-6404-AR
9	1	LED1	電動 LED 青色	LED GREEN CLEAR 0805 SMD	Chicago Miniature	CMD17-21VGC/TR8
10	1	R1	0E	RES 0.0 OHM 1/8W JUMP 0805 SMD	Panasonic - ECG	ERJ-6GEY0R00V
11	1	R2	220 オーム	RES 220 OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ221V
12	8	R3、R4、R7、R8、R12、R13、R14、R15	4.7k	RES 4.7K OHM 1/10W 5% 0603 SMD	Panasonic - ECG	ERJ-3GEYJ472V
13	5	R16、R17、R18、R19、R20	30 オーム	RES SMD 30 OHM 5% 1/10W 0603	Panasonic - ECG	ERJ-3GEYJ300V
14	1	SW1	3-POS DIP	SW DIP SLIDE SPST 3 POS UNSEALD	E-Switch	KAS1103E
15	1	TVS1	5V 350W	TVS DIODE 5VWM 14.5VC SOD323	Diodes Incorporated	SD05-7
16	1	U1	SPI F-RAM	IC FRAM 256-KBIT 40MHZ 8SOIC	サイプレス	FM25W256-G
17	1	U2	I2C F-RAM	IC FRAM 256-KBIT 3.4MHZ 8SOIC	サイプレス	FM24W256-G
18	1	該当なし	ジャンパ	ジャンパ	3M	969102-0000-DA
負荷なしコンポーネント						
19	1	J6	3X2 CONN FEMALE	CONN HEADER VERT DUAL 6POS GOLD	3M	961206-6404-AR
20	1	J7	2X2 CONN FEMALE	CONN HEADER VERT DUAL 4POS GOLD	3M	961204-6404-AR
21	3	R5、R6、R11	0E	RES 0.0 OHM 1/10W 0603 SMD	Panasonic - ECG	ERJ-3GEYJ302V
22	4	R9、R10、R21、R22	30E	RES SMD 30 OHM 5% 1/10W 0603	Panasonic - ECG	ERJ-3GEYJ300V

No.	数量	リファレンス	値	説明	メーカー	メーカー部品番号
23	3	TP1、TP2、TP3	黒色	TEST POINT PC MINI .040"D Black	Keystone Electronics	5001
<b>特別ジャンパのインストールの指示</b>						
24	1	J5	ジャンパをピン 1 とピン 2 の間に取り付ける	ヘッダ&ワイヤ ハウジング 2.54MM シャント	3M	969102-0000-DA
<b>ラベル</b>						
25	1	該当なし	該当なし	LBL、PCA ラベル、ペンダコード、日付コード、シリアル番号 121-60201-01 REV 01 (YYWWVVXXXXX)	サイプレス	
26	1	該当なし	該当なし	LBL、CY15FRAMKIT-001 QR コード、12mm X 12mm	サイプレス	
27	1	該当なし	該当なし	LBL、静電気防止警告、ULINE PN S-6516、5/8" x 2"、"Attention Observe Precautions"	サイプレス	

# 改訂履歴



## 改訂履歴

文書名: CY15FRAMKIT-001 シリアル F-RAM™開発キット ガイド

文書番号: 001-97580

版	ECN#	変更者	発行日	変更内容
**	4769237	HZEN	06/02/2015	これは英語版 001-95689 Rev. *A を翻訳した日本語版 001-97580 Rev. **です。