AN50963

EZ-USB FX1/FX2LP のブート オプション

著者: Prajith Cheerakkoda

関連プロジェクト: あり

関連製品ファミリ: CY7C68013A/14A/15A/16A、CY7C64713

ソフトウェア バージョン: CyConsole Version 1.7.0.2

関連アプリケーション ノート: AN65209

このアプリケーションノートの最新版または関連するプロジェクトファイルを入手するには http://cypress.com/an50963 をご覧ください。

更にサンプルコードをお求めでしょうか?以下を参照ください。

豊富な FX2LP サンプルコードにアクセスするには、USB High-Speed サンプルコードウェブページをご覧ください。

USB 3.0 ペリフェラルコントローラーをお探しでしょうか?

USB 3.0 製品ファミリにアクセスするには、USB 3.0 製品ファミリウェブページをご覧ください。

本アプリケーション ノート (AN50963) はサイプレスの EZ-USB® FX1™/FX2LP™ USB 2.0 ペリフェラル コントローラーで適用可能なブート オプションを詳細に説明します。

目次

1 はじめに 1
1.1 ブートローダの実行手順 2
2 I²C ブート 2
2.1 C0 ロード: I²C デバイスからの USB ID のロード 2
2.2 C2 ロード: I²C デバイスからのファームウェアのロード 3
2.3 EEPROM ハードウェア接続 4
2.4 I²C EEPROM のプログラミング 5
3 USB ブート 8
3.1 サンプル ホスト アプリケーション GUI 8
3.2 スクリプト ファイルを使用したファームウェア自動ダウンロード 11
4 外部パラレル メモリからのブート 13
4.1 ブートの要件 14
4.2 ファームウェア変更 15
4.3 ハードウェア変更 15
5 まとめ 15
改訂履歴 16
ワールドワイドな販売と設計サポート 17

1 はじめに

サイプレスの EZ-USB FX1/FX2LP は USB 2.0 ペリフェラル デバイス向けの高度な統合ソリューションを提供します。このソリューションは、業界標準 8051 マイクロコントローラーである高性能 CPU を内蔵しています。このソリューションのソフト (RAM ベース) アーキテクチャによりコンフィギュレーションやアップグレードは柔軟に実現できます。

FX1/FX2LP は、以下のブート オプションがあります。

- I²C ブート: I²C EEPROM を使って、ベンダーID (VID) / 製品 ID (PID) または VID/PID/デバイス ID (DID) 情報のみで完全なファームウェアを FX2LP の内蔵 RAM にダウンロードします。

注意事項: USB Implementers Forum (USB インペリメンターズ フォーラム) から受ける VID を使用する必要があります。テストの目的以外、サイプレス VID および特にサイプレスのデフォルト VID/PID は使用できません。このルールはファームウェア VID/PID および EEPROM VID/PID に適用されます。

- USB ブート: FX2LP 開発キット (DVK) に伴う Control Center アプリケーションのユーティリティを使用して、ファームウェア イメージを FX2LP の内蔵 RAM にダウンロードします。FX2LP デバイスを列挙した後で、USB ホストからファームウェアを FX2LP にダウンロードできます。デフォルトのサイプレス VID/PID または得意の VID/PID を使って FX2LP を列挙できます。VID/PID 情報のみを持っている I²C EEPROM を FX2LP に接続した場合、これらの VID/PID を使用できます。その後、ホスト アプリケーションを使ってファームウェアを FX2LP にダウンロードできます。
- 外部パラレル メモリからのブート: EZ-USB FX2LP/FX1 チップはアドレス／データ バスに接続された外部パラレル メモリからファームウェアを実行します。128 ピン EZ-USB デバイスの場合にのみ、オフ チップ データを加えたり、メモリをプログラムできます。100 ピンと 56 ピン EZ-USB チップは外部パラレル メモリをサポートしません。

EZ-USB FX2LP デバイス (CY7C68013A/14A/15A/16A) はフルスピード モードおよびハイスピード モードの両方に対応しますが EZ-USB FX1 デバイス (CY7C64713) はフルスピード モードにのみ対応します。FX1/FX2LP CPU はエンハンスド 8051 マイクロコントローラーであり、実行速度が高くて、機能が拡張されます。

注意事項: 本アプリケーション ノートでは、特記しない限り、説明する FX2LP の機能は、FX1 にも備えられています。

1.1 ブートローダの実行手順

リセット後、FX2LP ブートローダは以下のステップでブート オプションを選択します。

1. もし (I²C バス上またはパラレル アドレス／データ バス上の) オフ チップ メモリを EZ-USB に接続しない場合、配線で固定接続される内部ロジックが提供するディスクリプタと VID/PID/DID を使って FX2LP をデフォルト USB デバイスとして列挙します。
2. カスタム VID/PID/DID の値を含む EEPROM が EZ-USB I²C バスに接続される場合、EZ-USB はデフォルト USB デバイスとして列挙しますが、EEPROM からの VID/PID/DID 値を既存の値のかわりに用います。
3. EZ-USB ファームウェアを含む EEPROM が I²C バスに接続される場合、ファームウェアは自動的に EEPROM から EZ-USB オンチップ RAM にロードされます。その後 CPU はリセット状態から復帰して、ブートロードのコードを実行します。
4. EZ-USB は以下の 3 つの条件が起きたら、オフチップ メモリからのファームウェアを実行しはじめます。
   - フラッシュ, EEPROM, または他のメモリはアドレス/データ バス (128 ピン パッケージのみ) に接続されます。
   - 以前の条件を満たした正しくフォーマットされた EEPROM は使用されていません。
   - EA (外部アクセス) ピンは HIGH レベルに接続されます (EZ-USB がオフチップ メモリから 0x0000 でコード実行を開始することを示す)。

     再列挙とは FX2LP ブートに関する大切な概念です。デフォルト USB デバイスが列挙し、USB ホストがファームウェアとディスクリプタ表を EZ-USB にダウンロードしてから、ダウンロードされたコードを実行しはじめます。このコード実行は、USB デバイスからの物理的な切断／接続を電気的にシミュレーションし、これにより EZ-USB が 2 番目のデバイスとして再び列挙します。この時はダウンロードされたコードおよびディスクリプタにより定義された USB の特性を設定します。この特許化された二次列挙プロセスは「ReNumeration」(再列挙) と呼ばれます。

     EZ-USB Technical Reference Manual (TRM) の第 3 章「Enumeration and ReNumeration」(列挙および再列挙) では FX2LP ブートシーケンスについて、より詳しく説明します。

2 I²C ブート

I²C インターフェースを使って、ファームウェアまたは VID/PID/DID の情報を FX2LP にダウンロードします。もしファームウェアまたは VID/PID/DID の情報を格納している EEPROM が I²C バスに接続されたら、ファームウェアや VID/PID/DID 情報は自動的に EEPROM から FX2LP にロードされます。ファームウェアなら、オンチップ RAM にロードされます。その後 CPU はリセット状態から復帰して、ブートロードのコードを実行します。I²C バスを経由するロードは 2 つの形式があります。詳細な情報については、TRM の「3.2. EZ-USB Startup Modes」(EZ-USB スタートアップ モード) を参照ください。

2.1 C0 ロード: I²C デバイスからの USB ID のロード

パワーオン リセットでは、FX2LP デバイスが、アドレス 0 の 0xC0 値で、I²C バスに接続された EEPROM を検出したら、EZ-USB は自動的に VID, PID, および DID を EEPROM から内部メモリにコピーします。表 1 は C0 ロードのデータのフォーマットを示します。

表 1. C0 ロード フォーマット

EEPROM アドレス	目次
0	0xC0
1	ベンダーID (VID) L
2	ベンダーID (VID) H
3	製品 ID (PID) L
4	製品 ID (PID) H
5	デバイス ID(DID) L
6	デバイス ID(DID) H
7	コンフィギュレーション バイト

EEPROM の第 8 バイト目は以下の対象を制御するコンフィギュレーション ビットを格納します。

- I^2C バス スピード: 100 kHz (デフォルト値)
- 切断状態: デフォルト値は、EZ-USB がリセット状態から復帰するとすぐに USB に接続されることを定義

TRM の「3.4.2. Serial EEPROM Present, First Byte is 0xC0」(シリアル EEPROM 接続、最初バイト: 0xC0) ではコンフィギュレーション ビットについて詳しく説明します。

FX2LP はその後、ホストの Get Descriptor-Device (デバイスのディスクリプタを取得) 要求への対応アクションとして、ホストに EEPROM バイトを送信します。 これらの 6 バイトはデフォルト USB デバイス ディスクリプタの VID/PID/DID バイトのみを置き換えます。これにより、EEPROM の VID/PID/DID の値と一致したホスト ドライバーがホスト OS でロードされます。

2.2 C2 ロード: I^2C デバイスからのファームウェアのロード

パワーオン リセットでは、FX2LP はアドレス 0 の 0xC2 値で I^2C バスに接続された EEPROM を検出したら、EEPROM からファームウェアをダウンロードします。表 2 に C2 ロード用の EEPROM イメージ フォーマットを示します。C0 ロードと比べて、C2 ロードを使用すると USB ホストからファームウェアをダウンロードすることは不要です。

注意事項: TRM の「3.4.3. Serial EEPROM Present, First Byte is 0xC0」(シリアル EEPROM 接続、最初バイト:0xC2) では、コンフィギュレーション ビットについて詳しく説明します。

サイプレスは CyConsole および Control Center というホスト アプリケーション ユーティリティを提供します。開発の際はこれらのいずれかを使用して、EEPROM をプログラムします。Linux 開発の場合、ホスト アプリケーションは Linux 用の FX3 SDK からダウンロードできます。

FX2LP は 表 2 に示しているフォーマットを変換します;FX2LP はフォーマットを復号化し、このフォーマットを適切なメモリ位置にダウンロードします。Hex2bix ユーティリティを使って C2 ロード用の EEPROM イメージを生成できます。Hex2bix ユーティリティの詳細情報については、 FX2LP DVK をインストールした後、*C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Utilities\Hex2Bix* での *readme.txt* ファイルを参照ください。

表 2. C2 ロード フォーマット

EEPROM アドレス	目次
0	0xC2
1	ベンダーID (VID) L
2	ベンダーID (VID) H
3	製品 ID (PID) L
4	製品 ID (PID) H
5	デバイス ID(DID) L
6	デバイス ID(DID) H

EEPROM アドレス	目次
7	コンフィギュレーション バイト
8	長さ H
9	長さ L
10	開始アドレス H
11	開始アドレス L
–	データ ブロック
–	
–	長さ H
–	長さ L
–	開始アドレス H
–	開始アドレス L
–	データ ブロック
–	
–	0x80
–	0x01
–	0XE6
–	0x00
最終	00000000

EZ-USB は、EEPROM アドレスピン A2, A1, A0 を使用して、小さい EEPROM または大きい EEPROM に接続されているかを判断します。シングルバイト・アドレス EEPROM(小さい EEPROM)は 000 番地に接続する必要がありますが、ダブルバイトアドレス EEPROM(大きい EEPROM)は 001 番地に接続する必要があります。一般に、C2 ロードにはダブルバイトアドレス EEPROM が使用されます。EZ-USB デバイスが小さいまたは大きい EEPROM に接続されているかどうかを判断する方法の詳細については、テクニカルリファレンスマニュアル(TRM)の「13.6. EEPROM ブートローダ」を参照ください。

注意事項: FX2LP は 64KB EEPROM から C2 ロードを実行可能ですが、コードはハードウェア ブートローダでオンチップ RAM (16KB) のみにダウンロードできます。ブートローダを使用すると、2 段階のブートロードによりファームウェアを外部 RAM にダウンロードできます。Hex2bix ユーティリティは組込みブートローダを持っており、-e オプションを使って呼び出せます。-e オプションは、IIC フォーマットで外部 RAM 用のファイルを作成するために使用されます。

DVK の図は、EEPROM の FX2LP への接続例を示します。大きい EEPROM および小さい EEPROM それぞれに使用するデバイスは 24LC128-I/P および 24LC00/P です。

2.3 EEPROM ハードウェア接続

図 1 および図 2 に C0 ロードおよび C2 ロードの接続図をそれぞれ示しています。

外部 プルアップ抵抗は SCL ラインと SDA ラインの両方に必要です。推奨値は 2.2 kΩ です。以下の手順に従って接続を設定します。

- プルアップ抵抗は接続の FX2LP 側に配置されることを確認
- スイッチまたはジャンパを 1 個搭載して SDA ピンを FX2LP から切断します。これに起因して EEPROM ブートが無効になるため、システム内で EEPROM プログラムが可能

C0 ロードの場合、EEPROM アドレス ピンの A2, A1, A0 を使って、EEPROM が単一バイトでアドレス付けられた(000) メモリであることを示します。

図 1. C0 ロード時の EEPROM 接続

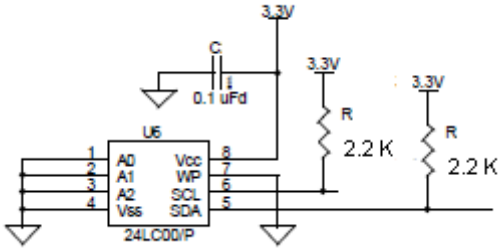
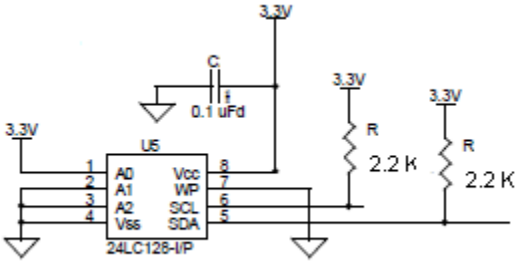


図 2. C2 ロード時の EEPROM 接続



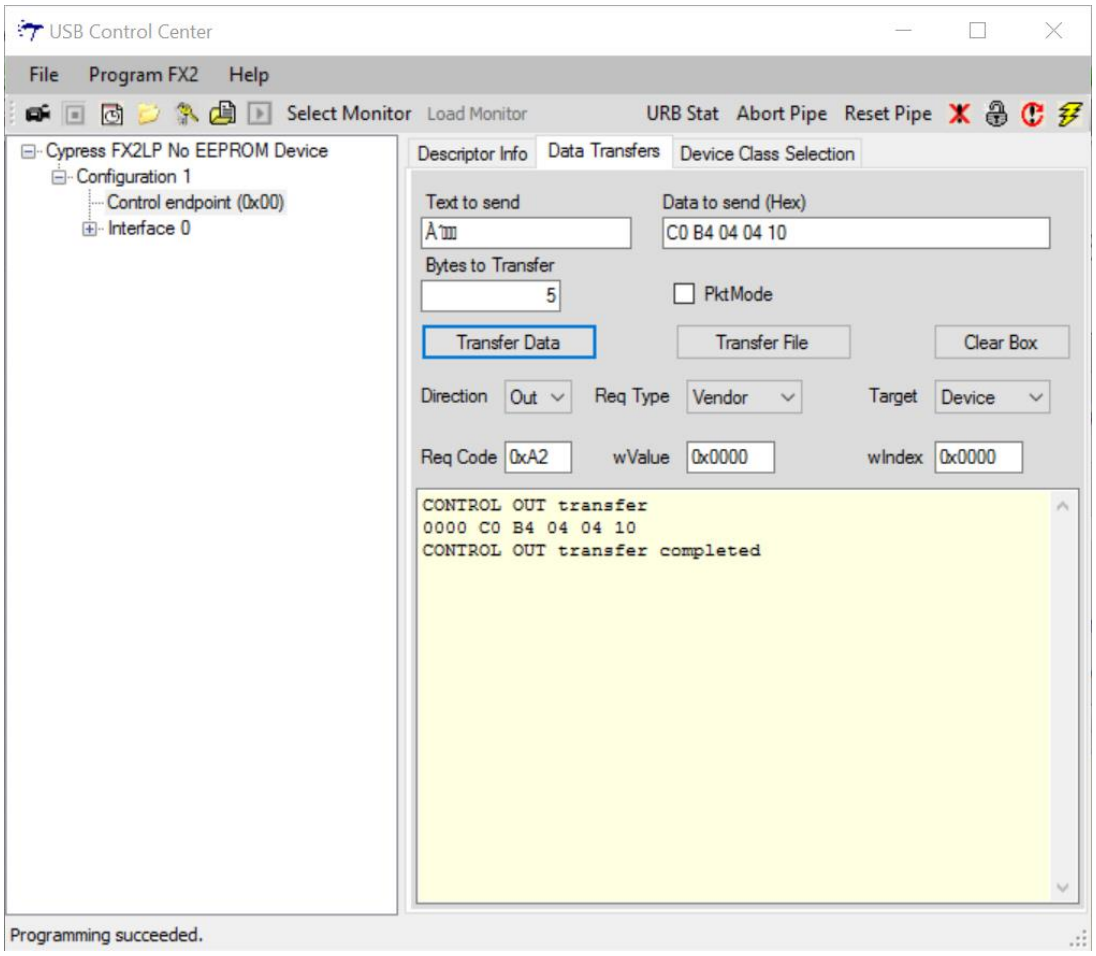
2.4 I²C EEPROM のプログラミング

EEPROM をプログラムする 2 つのオプションがあります。EEPROM を事前にプログラムしてから、システムにドロップするか、または Control Center により EEPROM をシステム内でプログラムするオプションです。以下は、Control Center を使って FX2LP DVK で C0 ロードを実行する手順です。

1. デバイスがデフォルトのサイプレス VID/PID で列挙できるよう EEPROM を接続しないで FX2LP DVK 基板に電源を供給します。デバイスは Control Center でアクセスするために *CyUSB3.sys* に関連付ける必要があります。CyUSB3.sys Programmer's Reference (CyUSB3.sys プログラマー用の参照) の「Matching Device to Driver」(ドライバーへのデバイスのマッチング) を参照ください。
2. 基板が列挙した後、DVK 上のスイッチ SW2 を「EEPROM」、SW1 を「小さい EEPROM」に設定します。
3. Control Center を開き、*C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Vend_ax* (インストールに応じて異なる場合もある) 内の DVK ファームウェア例の Vend_ax へナビゲートし()、「Open」をクリックします。これにより、*Vend_Ax.hex* ファイルが FX2LP の内蔵 RAM にダウンロードされます。Control Center を使って FX2LP をプログラミングする際は、C:*Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Windows Applications\Application Source files\c_sharp*(FX2LP DVK のインストール後に表示される) にある *CyControlCenter.pdf* ファイルを参照ください。
4. 「Control endpoint (0x00)」を選択し、**Req Code** フィールドに「0xA2」、**wValue** と **wIndex** フィールドに「0x0000」を入力し、**Direction** を「Out」, **Req type** を「Vendor」, **Target** を「Device」にします。最後に、表 3 で示しているように、VID/PID の組み合わせを **Data to send (Hex)** フィールドに入力します。図 3 では Control Center の **Data to send** フィールドに入れされたバイトを示します。
5. **Transfer Data** をクリックします。図 3 に EEPROM が正常にアップロードされた状態を示します。
6. プログラムされた VID/PID をチェックするために、デバイスのプラグを引き抜いてからまた差し込みます。今回、デバイスはプログラムされた VID/PID で列挙します。DVK が CyUSB3.sys ドライバーにバインドされている場合にのみ、コントロールセンターユーティリティで列挙されることに注意してください。例えば、'C0 B4 04 04 10' を渡すと、'Cypress FX2LP Sample Device' と列挙されます。

C0 ロードを実行する他の方法は、Control Center ユーティリティで**小さい EEPROM** オプションを使う方法です。ユーティリティを開き、ステップ 1 と 2 に進んで、**Program FX2** タブをクリックし、それから「小さい EEPROM」を選択します。VID/PID 情報を保存する IIC ファイルをブラウズして、**開きます**。IIC ファイル small.iic のサンプルは、アプリケーションノートと共に提供されています。

図 3. EEPROM プログラム



注意事項: 転送されるデータのフォーマットは、表 3 の C0 ロードフォーマットに従います。

表 3. C0 ロード時の入力バイト

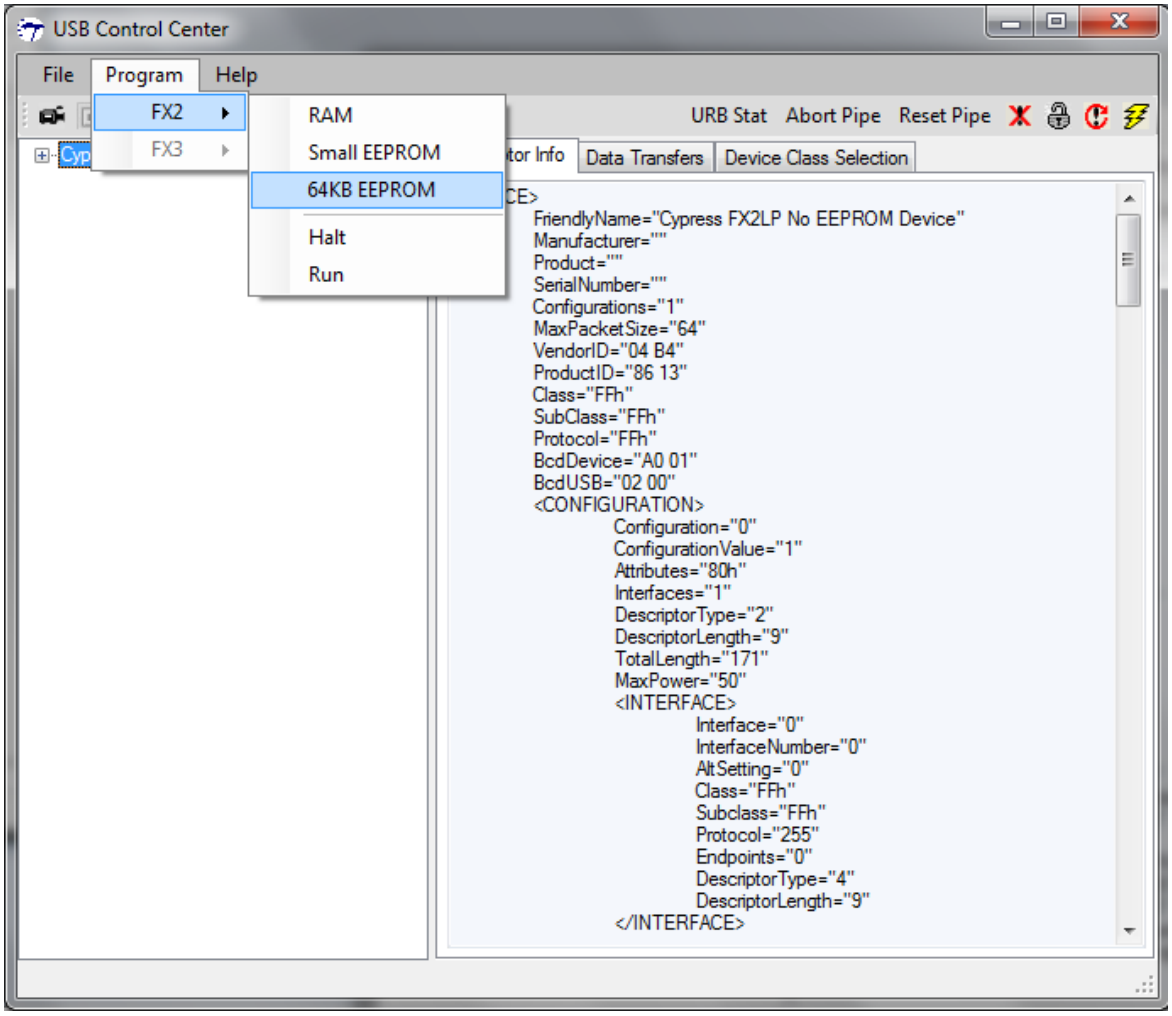
バイト数	パラメーター	注記
バイト 0	C0	可能な小さい EEPROM を表示
バイト 1	VID L	VID の下位バイト
バイト 2	VID H	VID の上位バイト
バイト 3	PID L	PID の下位バイト
バイト 4	PID H	PID の上位バイト

以下の手順は、Control Center を使って FX2LP DVK で C2 ロードを実行する方法を説明します。

1. FX2LP DVK 基板で、「SW2-NO EEPROM」を選択し、USB A-B ケーブルで基板上の J1 コネクタと Windows PC USB ホスト コントローラー ポートを接続します。EZ-USB デバイスは初期設定の VID/PID で列挙します。
2. EEPROM イメージ ファイル (*.iic*) をプログラミングする前に、スイッチ設定を「SW2-EEPROM」および「SW1-LARGE EEPROM」と選択して基板搭載の大きな EEPROM U5 を選択します。
3. Windows から Control Center を起動します: **Start** > **All Programs** > **Cypress** > **Cypress USBSuite** > **Control Center** (図 4 を参照ください)。EZ-USB FX2LP は「Cypress FX2LP No EEPROM Device」として表示されます。
4. 図 4 に示すように、EZ-USB インターフェース ウィンドウで **Program** > **FX2** > **64KB EEPROM** を選択してプロジェクト フォルダに移動します。*bulkloop.iic* イメージを *C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Bulkloop* または対応するインストール パスで選択します。
5. RESET ボタン S1 を押します。デバイスは新しい VID/PID で列挙し、ドライバーに通知します。バルク ループ動作のテストのために Control Center でデバイスにアクセスするには、デバイスを *CyUSB3.sys* に関連付ける必要があります。

   注意事項: サイプレスは FX1/FX2 のために *CyUSB.sys* を *CyUSB3.sys* に置き換えました。サイプレスは *CyUSB.sys* としてのサポートを停止しました。

図 4. C2 ロード



3 USB ブート

ホスト アプリケーションはベンダー コマンドを使ってファームウェアを FX2LP にダウンロードできます。USB エンドポイント 0 プロトコルはベンダー固有の要求にメカニズムを提供しています。bmRequestType フィールドのビット 6:5 はベンダー要求のために「10」にセットされます。

FX2LP は内部メモリを読み書きする 0xA0 ベンダー要求に対応しています。ホスト アプリケーションは 0xA0 ベンダー要求を使ってファームウェアを FX2LP にダウンロードできます。0xA0 ベンダー要求では wValue フィールドはアクセスするメモリ位置のアドレスを指定します。0xA0 ベンダー要求のデータ フェーズは、メモリに書き込むデータを送信するために使用されます。A0 ベンダー要求のフォーマットを表 4 に示します。詳細については TRM の「3.8. EZ-USB Vendor Request for Firmware Load」を参照ください。

表 4. A0 ベンダー要求フォーマット

Byte	Field	Value	Meaning	EZ-USB Response
0	bmRequest	**0x40**	Vendor Request, OUT	None required
1	bRequest	**0xA0**	'Firmware Load'	
2	wValueL	**AddrL**	Starting Address	
3	wValueH	**AddrH**		
4	wIndexL	0x00		
5	wIndexH	0x00		
6	wLenghtL	**LenL**	Number of Bytes	
7	wLengthH	**LenH**		

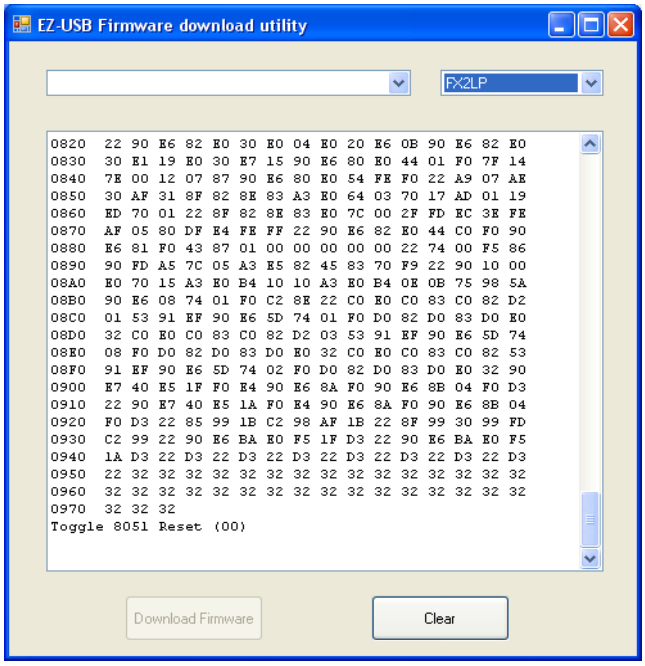
3.1 サンプル ホスト アプリケーション GUI

CyAPI.lib (VC++) ベースのホスト アプリケーション「EZ-USB ファームウェア ダウンロード ユーティリティ」は本アプリケーション ノートに同梱されており、Visual Studio 2008 を使って開発されました。0xA0 ベンダー要求を用いたファームウェアのダウンロードをデモします。CyAPI.lib は SuiteUSB 3.4 – Visual Studio 用 USB 開発ツールの一部です。CyAPI.lib の C#に相当する CyUSB.dll には、ファームウェアを FX2LP にダウンロードするための LoadRAM()API が含まれています。

ホストアプリケーションには、デバイスリストとターゲットデバイスリストの 2 つのドロップダウンリストがあります。デバイスリストは、CyUSB3.sys に接続されている USB デバイスを確認して、ターゲットデバイスを選択するために使用されます。ターゲットデバイスリストは、ターゲットデバイスが FX2LP か FX/ANXX を選択するために使用されます。また、テキストボックスと 2 つのボタンも含まれています。テキストボックスには、デバイスにダウンロードされているデータが表示されます。2 つのボタンの 1 つは、ファームウェアのダウンロードに使用されます。もう一方はテキストボックスをクリアするために使用されます。デバイスにダウンロードされた値は、ホストアプリケーションのテキストボックスで更新されます。図 5 は、ファームウェアダウンロードホストアプリケーション GUI を示します。

注意事項: このユーティリティは Windows XP, Windows Vista, Windows 7, Windows 8.1, および Windows 10 オペレーティング システムで動作確認済みです。

図 5. ファームウェア ダウンロードのホスト アプリケーション



注意事項: このサンプル ホスト アプリケーションは *CyUSB3.sys* ドライバーにのみ対応しています。

3.1.1 EZ-USB ファームウェア ダウンロード ユーティリティの実装

本アプリケーション ノート同梱のホスト アプリケーションは USB ホストから FX2LP へのファームウェア ダウンロードをデモします。この方法により、ファームウェアを容量の大きい EEPROM に格納することが不要となり、容量の小さく、かつ安い EEPROM を使用することが可能です。

この場合、ホストは最初に CPU をリセットに保持し、EZ-USB のデフォルト USB デバイスをファームウェアのダウンロードに使用します。その後、ホストが CPU をリセットから復帰させて、CPU がダウンロードしたコードを実行できるようになります。FX2LP へのファームウェア ダウンロードは A0 のベンダー コマンドを使用することで実行されます。ホストは CPU をリセット状態に移行するか、またはリセット状態から復帰させるために CPUCS レジスタに書き込む必要があります。FX2LP の CPUCS レジスタの 8051RES ビット (ビット 0) を使用して 8051 をリセットするかまたは実行します。USB ホストは 0xA0 ベンダー要求を介してこのビットに書き込めます。ホストは 8051 をリセットするには「1」を書き込んで、8051 を実行するには「0」を書き込みます。

0xA0 ファームウェア ロード要求は再列挙の後でも使用されますが 8051 がリセット状態に維持される間にのみ有効です。

アプリケーションがベンダー固有の USB 要求を実装し、ファームウェア ロード機能が不要な場合、ユーザーのカスタム要求に bRequest 値 0xA0 を使用しないでください。将来の非互換性を発生させないようにベンダー要求 0xA0～0xAF はサイプレスによって予約済みです。

Keil µVision2 は FX2LP 用のファームウェア開発に使用されます。ファームウェアが Intel hex フォーマット (http://microsym.com/editor/assets/intelhex.pdf)で出力されます。このファームウェア ファイルは FX2LP にダウンロードされるファームウェア イメージを得るために解析されます。EZ-USB ファームウェア ダウンロード ユーティリティはファームウェア ファイルを解析してファームウェア イメージを FX2LP にダウンロードします。

注意事項: CPUCS のアドレスの違い (AN21xx, FX, FX1 の場合は 0x1F92 で、FX2LP の場合は 0xE600) と AN21xx, FX と FX1, および FX2LP シリーズのチップ間のレジスタ アドレスの違いによって、使用される Vend_ax ファームウェアはこれら 2 つの間で違います。ホスト アプリケーションは AN21xx と FX へのファームウェア ダウンロードを実行するだけでなく、簡単に使用可能な ArrayList フォーマットで Vend_ax ファームウェアを出します。「AN45471 – Create Your Own USB Vendor Commands Using FX2LP」はベンダー コマンド、およびいくつかのベンダー コマンドを定義する Vend_ax に含まれるコマンドをテストする方法について詳細に説明します。

3.1.2 アルゴリズム

0xA0 ベンダー要求は FX2LP の内部メモリのみにアクセスできます。ファームウェアの一部が外部メモリに保存される場合、外部メモリにアクセスする方法が必要です。DVK の一部である Vend_ax 例はこの目的に使用されます。DVK をインストールした後、この例を *C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.1\Firmware\Vend_ax* (インストールによってロケーションが異なる) から入手できます。このファームウェアは FX2LP 外部メモリの読出し／書込み用に 0xA3 ベンダー要求を実装します。このベンダー要求は内部メモリにもアクセスできます。

標準の Intel Hex フォーマット

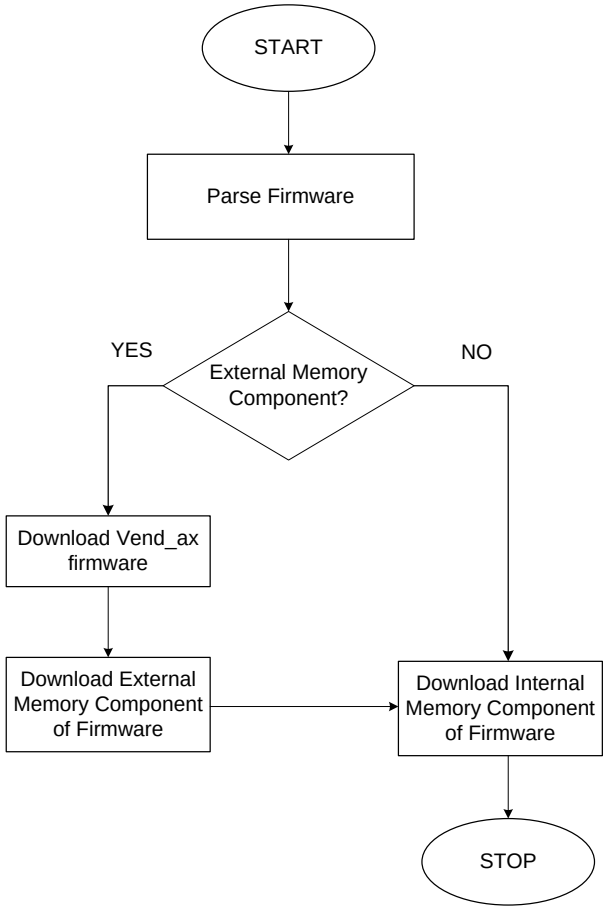
HEX ファイルの各行は基本構成が同じで、「レコード」と呼ばれています。各レコードは以下の 6 つの部分から成ります。

- **スタート コード**: 1 文字 (ASCII コロン「:」)
- **バイト カウント**: データ フィールドのバイト数 (16 進数のペア) である 16 進数の 2 桁
- **アドレス**: データのメモリ位置の開始の 16 ビット アドレスである 16 進数の 4 桁
- **レコード タイプ**: データ フィールドのタイプを定義する 16 進数の 2 桁(00～05)
- **データ**: 16 進数の 2*n* 桁で表されるデータの *n* バイトのシーケンス
- **チェックサム**: フィールド 1 とフィールド 6 (スタート コード「:」バイトとチェックサムの 16 進数の 2 桁) 以外のすべてのフィールドの合計値の 2 の補数の最下位バイトである 16 進数の 2 桁

注意事項: HEX ファイル フォーマットの詳細については、仕様書を参照ください。

図 6 にホスト アプリケーションによりファームウェアを FX2LP にダウンロードするために使用されるアルゴリズムを示します。

図 6. ファームウェア ダウンロード アルゴリズム



EZ-USB ファームウェア ダウンロード ユーティリティでの関数

以下はホスト アプリケーション内の関数です。

- **WndProc(message %m):** これのユーザー定義関数はウィンドウに送信されるメッセージを処理します。
- **CPU_Reset(bool ON):** この関数はブーリアン入力に基づいて FX2LP の CPU をリセットまたは実行します。入力が true の場合、CPU はリセットされます。入力が false の場合、CPU は実行できます。
- **GetDevice():** この関数は *CyUSB.sys* に関連付けられたデバイスを Device_List ドロップダウン リストに移入し、管理します。
- **Error(String^ err):** この関数はエラー メッセージをプリントするために使用します。
- **InitVendAX(bool FX2LP):** この関数は Vend_ax ファームウェアを含むアレイ (VendAX) を初期化するために使用します。この関数へのブーリアン入力は、ファームウェアに外部メモリ コンポーネントが含まれるかを指定します。VendAX アレイは Vend_ax ファームウェアを FX2LP にダウンロードするために使用します。
- **Firmware_Download(String^ file):** この関数は、解析用の入力としてファームウェア ファイルのパスを使用し、FX2LP にファームウェアをダウンロードします。「VendAX」が入力として渡された場合、この関数は FX2LP に Vend_ax ファームウェアをダウンロードします。ファームウェアが外部メモリ コンポーネントを持つ場合、この関数は Vend_ax ファームウェアをダウンロードするために再帰的に呼び出されます。FX2LP の内部 RAM は 16KB です。FX2LP の先行バージョン (AN21xx、FX など) の場合は 8 KB です。0xA3 が内部と外部メモリにアクセスすることができるので、アルゴリズムを単純化するために、ホスト アプリケーションは 0x2000 を外部メモリの開始位置として使用します。

3.2 スクリプト ファイルを使用したファームウェア 自動ダウンロード

Control Center は、自動的に FX2LP/ FX1 にファームウェアをダウンロードできるスクリプトを作成可能です。この手法を使用するには、C0 ロード: I2C デバイスのセクションで説明しているように、FX2LP/FX1 を「C0 Load」経由で VID / PID で起動します。INF ファイルには、ファームウェアをダウンロードするためのシーケンスの記録を含むスクリプト ファイルにこの VID/PID(BootloaderVID/PID)をバインドします。列挙すると、このファームウェアはスクリプト ファイルを使用してデバイスにダウンロードされます。このデバイスは、ダウンロードしたファームウェアに含まれている VID/PID で列挙します。この方法の利点は、ファームウェアの大きい EEPROM への格納が不要になることです。したがって、より小さくて安価な EEPROM を使えます。また、USB からのファームウェアのダウンロードを自動化します。スクリプト機能はサイプレスが提供する *CyUSB3.sys* ドライバーでサポートされています。

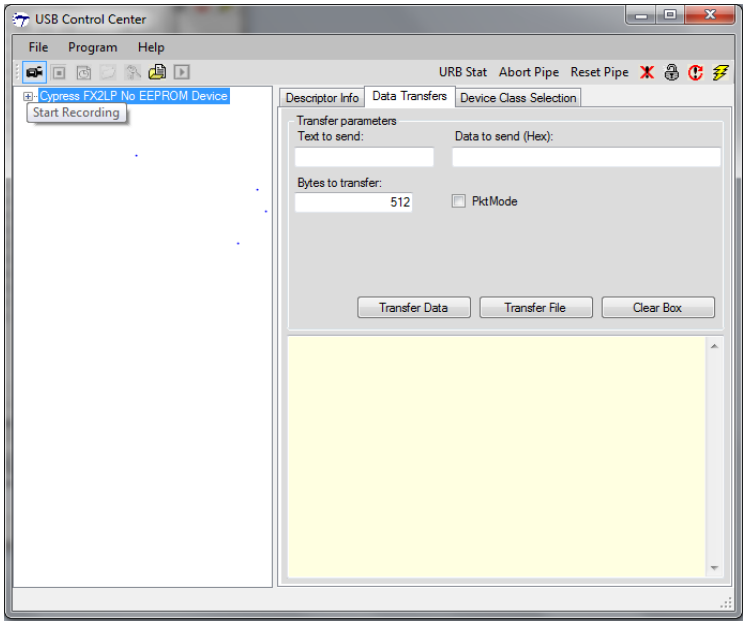
注意事項: ダウンロードされたファームウェアで使用される VID/PID 値は、ブートローダ VID/PID 値とは異なる必要があります。そうでない場合は、ファームウェアの再帰的なダウンロードが発生します。

3.2.1 スクリプト ファイルの作成

必要なファームウェアを手に入れた後、FX2LP の DVK を使用してファームウェアをダウンロードするためのスクリプトを作成します。

1. SDA ラインと FX2LP の接続を切ります (SW2 を「No EEPROM」位置に)。
2. USB ケーブルを使用して開発基板をコンピュータに接続します。
3. 基板がそのデフォルト設定で列挙することを許可します。Device Manager では、この設定は、FX2LP の場合「Cypress FX2LP No EEPROM Device」(VID/ PID0x04B4/0x8613) として表示され、FX1 の場合「Cypress EZ-USB FX1 No EEPROM Device」 (VID/ PID0x04B4/0x6473) として表示される必要があります。ドライバーをデバイスにバインドする手順は、DVK ドキュメントの一部である EZ-USB DVK ユーザー ガイドの 7.2.4 項「Binding Cypress USB Driver for the Downloaded Firmware Image」(ダウンロードしたファームウェア イメージ用のサイプレス USB ドライバーのバインド) に記載されています。
4. Control Center を開きます。
5. 図 7 に示すように、Control Center の左上部にある「Record」ボタンをクリックします。

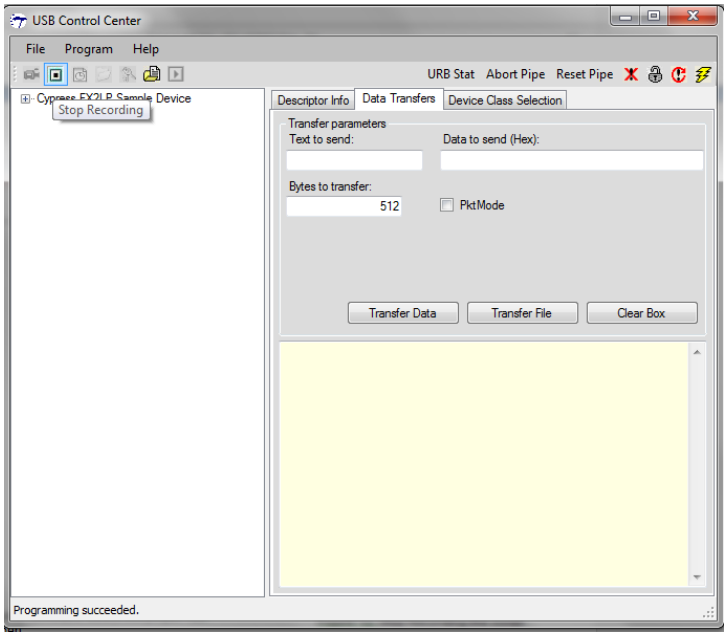
図 7. スクリプト記録開始



6. **Program** > **FX2** > **RAM** を選択して、スクリプトを生成されるファームウェア (hex ファイル) を選択します。
7. プログラムが正常に RAM にロードされると、図 8 に示すように「Start recording」ボタンの近くの「Stop recording」ボタンをクリックします。スクリプトを*.spt* 拡張子で保存します。
8. ファイルに「*MyDevice.spt*」という名前を付け、保存します。ここでスクリプトが作成されました。

   注意事項: スクリプトの生成が Control Center によって引き起こされたエラーによって失敗することがあります。この場合には、Control Center を閉じてアドミニストレータ モードでそれを再開し、各ステップを繰り返し、スクリプトを作成します。

図 8. スクリプト記録停止



3.2.2 INF ファイル作成

以前に作成したスクリプトを用いてファームウェアをダウンロードするためには、INF ファイルを作成する必要があります。このファイルはデバイスの小さな EEPROM にプログラムされた VID/PID を識別します。次に、スクリプト ファイルを用いてファームウェアをデバイスにロードします。デモ用のサンプルの INF ファイルは、添付するアプリケーション ノート プロジェクトの一部として含まれています。スクリプトに基づいてファームウェアをダウンロードするための INF ファイルの作成ステップは CyUSB3.sys Programmer's Reference の第 2 項に記載されています。

1. このアプリケーション ノートに搭載しているサンプルの INF ファイル *cyusb3.inf* から実行します。スクリプトをできる限り容易に実行できるためにこのファイルにコード コメントを入れて構成されます。
2. ノートパッドなどのテキスト エディタで *cyusb3.inf* ファイルを開きます。
3. *cyusb3.inf* ファイルで、FX2LP の標準ブートローダの VID/PID である 04B4/8613 の VID/PID を小さな EEPROM にダウンロードされた VID/PID に変換します。
4. バルク ループ ファームウェアの VID/PID である 04B4/1004 VID/PID をユーザーのファームウェアからの VID/PID に変換します。
5. INF ファイルの下位にある「strings」部分に示されている各 VID/PID がスクリプトダウンロード前の VID / PID のエントリーに一致し、ファームウェア ダウンロード後デバイスが列挙する VID/PID に一致することを確実にしてください。クォーテーション マークの中に意味のあるテキストを入れます。
6. 新しい INF ファイルを保存します。このアプリケーション ノートに添付されている INF ファイルにはデジタル署名が含まれています。INF ファイルを変更するとカタログ ファイル内のデジタル署名が無効になることに注意してください。INF ファイルの認定を取得する必要があります。また、署名なしにそれを使用すると、Windows 8, Windows 8.1, および Windows 10 ではドライバー署名強制機能を無効化する必要があります。Windows 7, Windows Vista, および Windows XP では、警告が表示しますが、ドライバー署名を無効化せずに接続できます。

3.2.3 スクリプト ファイルによるファームウェア ダウンロード

このアプリケーションノートに付属のファイルは、スクリプトを使用してバルクループファームウェア（VID 04B4 および PID 1004）を FX2LP にダウンロードします。C0 ロードを通して小さな EEPROM にダウンロードされた VID/PID 値は 04B4/8613 です。したがって、付属の INF ファイルにはこれらの ID が指定されます。これらのファイルは、このアプリケーションノートとともに、OS および CPU アーキテクチャにより別々のフォルダにあります。異なる VID/PID 値を試したい場合は、次を変更してください：小さい EEPROM にダウンロースために使用された IIC ファイル，RAM にダウンロードされるファームウェア，スクリプトファイル, INF ファイル。

これらのファイルを試すために使用されるデフォルトの VID/PID は 0x04B4/0x8613 です。ファームウェアの VID/PID は 0x04B4/0x1004 です。

1. スクリプト ファイル，INF ファイル，および基板が用意されると、スクリプトを用いて FX2LP デバイスにファームウェアをダウンロードします。
2. スクリプト, ドライバー ファイル, および INF ファイルは同じフォルダに格納されていることを確実にしてください。
3. 基板をホストに接続し、添付するアプリケーション ノートの「Drivers」フォルダにあるドライバーを手動で結びつけます。
4. このアプリケーション ノートの「Drivers」フォルダから Windows XP, Windows Vista, Windows 7, Windows 8, および Windows 8.1 (32 ビットと 64 ビット) のために異なる INF ファイルが用意されています。ユーザーの OS に適合する INF ファイルを選択してください。

4 外部パラレル メモリからのブート

EZ-USB アドレス／データ バスを通してパラレル メモリを外部で接続できます。ファームウェアは EZ-USB の内部 RAM にロードされずに、外部のパラレル メモリから直接実行します。このメモリはフラッシュ，EEPROM，または他の任意のパラレルメモリです。

注意事項: 128 ピン パッケージのみは外部アドレス／データ バスを持っています。そのため、このアプリケーションは、CY7C68013A-128AXC，CY7C68014A-128AXC，CY7C64713-128AXC，および CY7C64714-128AXC の製品のみに制限されます。

このモードのこのような要求を満たすために、ファームウェアと開発基板に対して若干の修正が必要です。

4.1 ブートの要件

1. このモードのコード メモリは外部パラレル メモリでなければなりません。電源投入時に EA 入力ピンが HIGH 論理に設定されると、外部 64KB メモリが全部コード空間としてマッピングされすべての内部メモリ (主な RAM) がデータ空間として使用されるように、EZ-USB FX2LP のメモリ マップは修正されます。

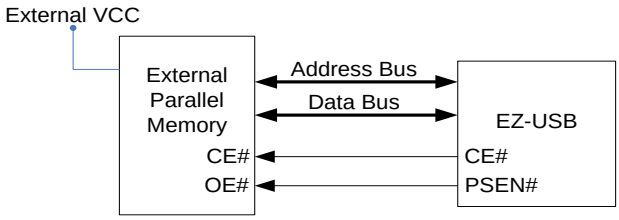
   注意事項: EZ-USB の他のフォン ノイマン アーキテクチャ構成と対照的に、EA=1 を設定すると EZ-USB はハーバード アーキテクチャ構成になります。

2. EZ-USB は最初に I^2C バスを通してシリアル EEPROM から起動しようとします。0xC0 と 0xC2 を最初のバイトとして確認して起動方法を決定します。シリアル EEPROM ブートロードを避けるために、I^2C バスにシリアル EEPROM がないことが必要です。外部シリアル EEPROM が接続されると、最初のバイトの値は 0xC0 または 0xC2 を持ってはいけません。

3. これらの 2 つの要求を満たすと、FX2LP DVK のジャンパ設定が MM0=IN と MM1=OUT の場合、電源投入時に EZ-USB がアドレス／データバスを通して外部メモリからファームウェアを検索して 0x0000 のアドレスからコードを実行し始めることを確保できます。FX2LP DVK のメモリ マップの詳細については EZ-USB DVK User Guide の「4.8 Memory Maps」を参照ください。

4. 実験用の WinBond W27C512-45Z パラレル EEPROM のアクセス時間は 45 ns です。W27C512-45Z の詳細については「W27C512 データシート」を参照ください。置換可能なデバイスは Atmel AT27C512R です。詳細については、AT27C512R データシートを参照ください。図 9 に接続の詳細を示します。チップは十分な電源で安定させてください; チップによっては DVK がサポートするチップより高消費電力が必要です。[1]

   図 9. 外部パラレル メモリの接続詳細



   外部パラレル メモリは以下の特性が必要です。

   - アクセス時間は速くて EZ-USB がコードを読み出すことを可能にします。アクセス時間 tACC1 の詳細については、FX2LP データシートを参照ください。
   - ファームウェアの生のバイナリ イメージは、シリアル EEPROM へロードする IIC ファイル イメージと違い、スタート バイトなしで位置 0x0000 から外部パラレル メモリにロードする必要があります。Keil ソフトウェアが生成する hex ファイルはインテル hex フォーマットです。このファイルは直接にロードできません。プログラマー ユーティリティでインテル hex フォーマットが生のバイナリ ファイルのフォーマットに自動的に変換されない限り外部パラレル メモリにロードする前に、生のバイナリ ファイルのフォーマットに変換してください。ALL-100 プログラマーを使って FX2LP DVK で WinBond W27C512-45Z パラレル EEPROM をプログラミングする試験は確認済みです。このプログラマーのソフトウェア パッケージは、インテル hex ファイルを生の hex ファイルに変換するユーティリティを持っています。Hex2bix ユーティリティが生成する IIC ファイルはスタート バイトがファームウェア イメージに挿入されるため、外部パラレル メモリへロードしてはいけません。

5. セットアップ データ ポインターは、内部メモリ位置 0x0000～0x3FFF、0xE000～0xE1FF だけアクセスできます。列挙時にファームウェアが SUDPTRH:L を使ってディスクリプタをポイントする場合、ディスクリプタは内部データ空間に格納しなければなりません。このブート方法では、ディスクリプタを EZ-USB の外部にあるコード メモリにファームウェアと格納します。したがって、起動の際は外部メモリから内部メモリに無条件でディスクリプタをコピーする必要があります。

[1] サイプレスは Atmel AT27C512R に対してテストを行っていません。データシートによると AT27C512R は代替可能な部品です。これは OTP デバイスです。より遅いアクセス時間で再プログラム可能なデバイスは 24 MHz で動作する FX2LP に対応します。

4.2 ファームウェア変更

外部パラレル メモリから動作するために、EZ-USB フレームワークに基づいて任意ファームウェアを CY3684 DVK の部品として変更できます。ブートの要件の節に説明されたようにディスクリプタを EZ-USB 内部メモリ (主な RAM) にコピーしなければなりません。EZ-USB フレームワークの変更手順は以下のとおりです。

1. *fw.c* には、main 関数の TD_Init() の次にディスクリプタ セットアップに関するセクションがあります。このセクションでは、「if ((WORD)&DeviceDscr & 0xC000)」の IF 条件を取り除き、そして、この IF 条件に存在するすべてのコード (主にディスクリプタのコピー) は無条件に実行してください。これにより、標準 USB ディスクリプタ (デバイス/コンフィギュレーション ディスクリプタ) の値は常に内部メモリ (主な RAM) にコピーされます。
2. これらの標準 USB ディスクリプタを外部メモリから内部メモリにコピーするラインを変更して外部メモリに保存するディスクリプタが確かにコード メモリにあることをコンパイラに通知します: 「*((BYTE xdata *)IntDescrAddr+i) = *((BYTE xdata *)ExtDescrAddr+i);」を「*((BYTE xdata *)IntDescrAddr+i) = *((BYTE code *)ExtDescrAddr+i);」に変更。
3. クラス固有ディスクリプタが定義されたすべてのアプリケーションに対してクラス固有ディスクリプタを外部メモリから内部メモリへコピーします。HID KB 事例の場合、pHIDDscr, pReportDscr, および pReportDscrEnd の 3 つのディスクリプタ ポインターは main 関数で更新されます。これらのディスクリプタ ポインターのオフセットも main 関数で更新されます。DR_GetDescriptor 関数 (*periph.c* での関数) 内のディスクリプタ ポインターの再評価を削除します。

4.3 ハードウェア変更

ハードウェア変更で一番大切なのは、EA ピンをロジック 1 にセットすることです。しかし、DVK 基板を使う際は、特定のハードウェア接続を変更して、外部パラレル メモリから EZ-USB を実行する必要があります。これらの変更は CY3686 DVK および CY4611B DVK の両方に適用します。

4.3.1 基板の変更

1. DVK 基板はアドレス／データ バスには SRAM (CY3686 の U3 および CY4611B の U4) が搭載されます。バスの競合を防止するために、このチップを除去するまたは無効にします。SRAM チップ イネーブル (ピン 5、RAMCE#) をロジック 1 にセットしてチップを無効にします。
2. SRAM へのコントロール信号を GAL22LV10C チップ (U2) を通して配線します。GAL22LV10C チップ (U2) は SRAM に信号を送る前に外部パラレル メモリ アクセス信号を変換します。これらの信号がチップを経由しないように、ソケットから GAL22LV10C を除去する、または再プログラムします。
3. I^2C バスのシリアル EEPROM を未接続にするまたは無効にしてください。シリアル EEPROM を無効にするには、CY3686 では SW2 を「No EEPROM」にセット、そして、CY4611B 上のジャンパ J22 を除去します。

5 まとめ

本資料では、FX1/FX2LP で適用可能なブート オプション, それらのオプションの重要性, メリット, およびデメリットについて説明しました。この情報により、適用するアプリケーションに対して最適なブート オプションを決められます。

著者について

氏名: Prajith Cheerakkoda

役職: 上級アプリケーション エンジニア

改訂履歴

文書名: AN50963 - EZ-USB FX1/FX2LP のブート オプション

文書番号: 001-86997

版	ECN	変更者	発行日	変更内容
**	3954738	HZEN	04/04/2013	これは英語版 001-50963 Rev *F を翻訳した日本語版 001-86997 Rev. **です。
*A	4885098	HZEN	08/19/2015	これは英語版 001-50963 Rev *J を翻訳した日本語版 001-86997 Rev. *A です。
*B	5412347	LIP	08/23/2016	新しいテンプレートに更新しました。サンセットレビューを完了する。
*C	6300422	HIKA	09/10/2018	これは英語版 001-50963 Rev *L を翻訳した日本語版 001-86997 Rev. *C です。

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持っています。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	cypress.com/arm
車載用	cypress.com/automotive
クロック&バッファ	cypress.com/clocks
インターフェース	cypress.com/interface
IoT (モノのインターネット)	cypress.com/iot
メモリ	cypress.com/memory
マイクロコントローラ	cypress.com/mcu
PSoC	cypress.com/psoc
電源用 IC	cypress.com/pmic
タッチ センシング	cypress.com/touch
USB コントローラー	cypress.com/usb
ワイヤレス	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 | 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

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

© Cypress Semiconductor Corporation, 2009-2018. 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で) 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ことはない。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, エラッタ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, 参照目的のためのみ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。) のためには設計, 意図又は承認されていない。重要な構成部分とは, それ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 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。