

FX2LP™ 入門

About this document

Scope and purpose

AN65209 では、EZ-USB® FX2LP™ USB 2.0 デバイス コントローラーを紹介します。本アプリケーション ノートは、FX2LP のプロジェクトを作成し、その様々な開発ツールを研究する手助けをします。また、FX2LP を迅速に深く理解できるように適切な文書へのアクセスを提供します。

関連プロジェクト

あり

関連製品ファミリ

CY7C68013A/14A/15A/16A/53

ソフトウェア バージョン

なし

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。

Table of contents

About this document	1
Table of contents	1
1 はじめに	3
1.1 Bulkloop_FX2LP プロジェクト	3
1.1.1 VS_Control Center	3
1.1.2 Bulkloop_VCS	3
1.1.3 Windows ドライバーとライブラリ	3
2 USB 2.0	5
3 FX2LP の概要	6
3.1 FX2LP の機能	6
3.1.1 USB インターフェース	6
3.1.2 パラレル インターフェース	7
3.1.2.1 スレーブ FIFO	7
3.1.2.2 汎用プログラマブル インターフェース	8
3.2 シリアル インターフェース	9
3.3 CPU とメモリ	9
3.3.1 8051	9

Table of contents

3.3.2	ブート オプション	9
3.3.2.1	パッケージ オプション	10
3.4	FX2LP のサンプル アプリケーション	11
3.4.1	スレーブ FIFO を使用して FPGA/ASIC と通信	11
3.4.2	参考資料	11
3.4.3	FPGA の FX2LP からのブート	11
4	サイプレスの設計リソース	13
5	FX2LP 開発ツール	15
5.1	FX2LP 開発基板	16
5.2	Bulkloop デモを実行	17
5.2.1	USB Control Center を使用	17
5.2.2	BULKLOOP_VCS の使用	20
5.3	FX2LP ファームウェア開発	21
5.3.1	Keil uVision2	21
5.3.2	サイプレス USB ファームウェア フレームワーク	21
5.3.2.1	TD_Init	23
5.3.2.2	TD_Poll	23
5.3.2.3	割り込みサービス ルーチン	24
5.3.2.4	USB の 2 つのスピードに対応	25
5.3.3	Bulkloop プロジェクトをビルド	25
5.4	GPIF Designer	26
6	Windows 側	29
6.1	サイプレスの USB ドライバー	29
6.1.1	ドライバ／ライブラリの代替オプション	30
6.2	サイプレス ライブラリ	31
7	まとめ	32
8	付録 A: FX2LP 開発キット (DVK)	33
8.1	ファームウェア サンプル プロジェクト	34
9	付録 B: サイプレスからの追加 USB ハイスピード デバイス	36
10	付録 C: サードパーティの開発キットと SDK	37
10.1	サードパーティの SDK	37
11	付録 D: アプリケーション ノートとリファレンス デザイン	38
11.1	アプリケーション ノート	38
12	付録 E: .inf ファイルへのカスタム VID と PID の追加	41
	改訂履歴	43

はじめに

1 はじめに

サイプレス EZ-USB FX2LP (以下、FX2LP) は、USB 2.0 の最大帯域幅に対応するために設計された柔軟な USB 2.0 ペリフェラル コントローラーです。USB 2.0 の 480 Mbps の信号速度を最大限に活用するために、FX2LP は、特殊化されたハードウェアを含んでおり、USB データをバッファリングし、MCU、ASIC、FPGA などのさまざまな高帯域幅の外部デバイスにシームレスに接続します。本アプリケーション ノートでは、USB 2.0 を簡単に紹介してから、FX2LP の高いスループットに寄与する機能を説明します。

いかなる IC でも、その優位性はそれをサポートするツールの優位性に依存します。よって、本アプリケーション ノートの残りの節では **FX2LP 開発キット** (DVK) および FX2LP ファームウェアとソフトウェア開発ツールを紹介します。添付 zip ファイルには、サンプル コードを格納しているフォルダと本アプリケーションで詳しく説明するアプリケーションがあります。以下で、それらを簡単に説明します。

1.1 Bulkloop_FX2LP プロジェクト

本アプリケーション ノートに添付された Bulkloop プロジェクトは、USB デバイスを作るために FX2LP ファームウェアを構成する方法を示します。ほとんどの USB 要求処理は、サイプレス製の USB ファームウェア フレームワークで行われ、ユーザー コードが特殊なアプリケーションの要求に応じてのみ行います。*bulkloop.c* ファイルには、カスタムアプリケーションの基本として使用できる完全な USB デバイス テンプレートがあります。

サイプレス ツールは、FX2LP CPU (8051) 用の無料の Keil uVision2 評価バージョンを含んでいます。Keil uVision2 統合開発環境 (IDE) は、ユーザー コードと USB ファームウェア フレームワークをコンパイルして HEX ファイルを生成します。この HEX ファイルは、提供された **USB Control Center** アプリケーションによりロードおよび提供された **Bulkloop_VCS** アプリケーションで実行する必要があります。

ファームウェアツールの説明の一部に、GPIF Designer プログラムが、どのようにグラフィカルな波形エントリを使用してカスタムインターフェースを作成するかを示す紹介がされています (**GPIF Designer** を参照)。

1.1.1 VS_Control Center

これは、**USB Control Center** というアプリケーションを作成する Visual C#で書かれている Microsoft Visual Studio ソリューションです。このアプリケーションは FX2LP コード (HEX ファイル) を FX2LP 開発キットにダウンロードするために使用されます。すぐに使用するための実行ファイルおよび参考用のソースコードが提供されています。サイプレス.NET ライブラリを使用し、その多くの高レベル関数の使い方を実演します。

1.1.2 Bulkloop_VCS

これは、Visual C#で書かれている簡単な Windows Visual Studio アプリケーションです。各種 USB コントローラー カードやコンピューターで USB 転送帯域幅を測定するのに使用できる連続データ ループとバイトカウントを備えています。USB Control Center と同様に、ソース コードと実行ファイルが提供されています。

1.1.3 Windows ドライバーとライブラリ

サイプレスは、FX2LP ベースの設計に対応する Windows ドライバーを提供しています。これは、すべての USB 転送タイプに対応し、顧客の製品で使用するための 2 進数形式で提供されています。

USB Control Center と Bulkloop_VCS アプリケーションは、サイプレス製の管理された Microsoft .NET クラス ライブラリ **CyUSB.dll** を使用しています。これは、Visual Basic.NET、Visual C#、Visual C++ (Win フォー

はじめに

ム)、Visual J#の言語と互換性があります。このライブラリは提供する高レベル関数によって、ドライバーの複雑さを出さずに、非常に簡単な USB プログラミング モデルを提供します。

USB 2.0

2 USB 2.0

USB はキーボード、マウス、プリンター、サムドライブ、ハード ディスク、ポータブル メディア プレーヤーなどのコンピューター周辺機器の接続を標準化します。USB は周辺機器に通信を提供し、電源を供給します。USB は PC や消費者機器の主要な接続ソリューションになってきました。その人気は、その標準化された強固な基礎構造で達成された使い易さに因るところが大きいです。

1996 年に公開された USB 1.0 仕様は、当時の一般的なデバイスに対応するための 2 種の転送速度を定義しました。ロースピードのデバイスは 1.5Mbps で動作し、キーボードやジョイスティックなどのデバイスに対応します。フルスピードのデバイスは 12Mbps で動作し、プリンターやディスクドライブなど高帯域幅のデバイスに対応します。1998 年に公開されたマイナー リビジョン 1.1 は、主にハブの問題に対応し、広く採用されている第一世代の仕様となりました。

USB 2.0 仕様は、2000 年に公開されました。これは、信号速度を 480Mbps に向上させ、Hi-Speed (ハイスピード) と名付けます。2.0 仕様は以前の転送速度と互換性があります。

2008 年に公開された USB 3.0 仕様は、さらに信号速度を 5Gbps に向上させ、SuperSpeed (スーパースピード) と呼びます。サイプレスの FX3 ファミリはこの速度に対応しています。

本アプリケーション ノートの対象は、ハイスピードとフルスピードで動作するサイプレス FX2LP です。**付録 B** には、サイプレスが提供している他のハイスピード機器の一覧を示します。サイプレスが提供しているすべての USB コントローラーは[こちら](#)に記載されています。USB 2.0 の詳細については、**USB 101: An Introduction to Universal Serial Bus 2.0** を参照してください。

FX2LP の概要

3 FX2LP の概要

FX2LP は単一のチップに USB 2.0 トランシーバ、スマートシリアルインターフェースエンジン (SIE)、大容量データバッファ、拡張された 8051 マイクロコントローラ、およびプログラム可能なペリフェラルインターフェースを備えています。FX2LP の簡略ブロック図を **Figure 1** に示します。

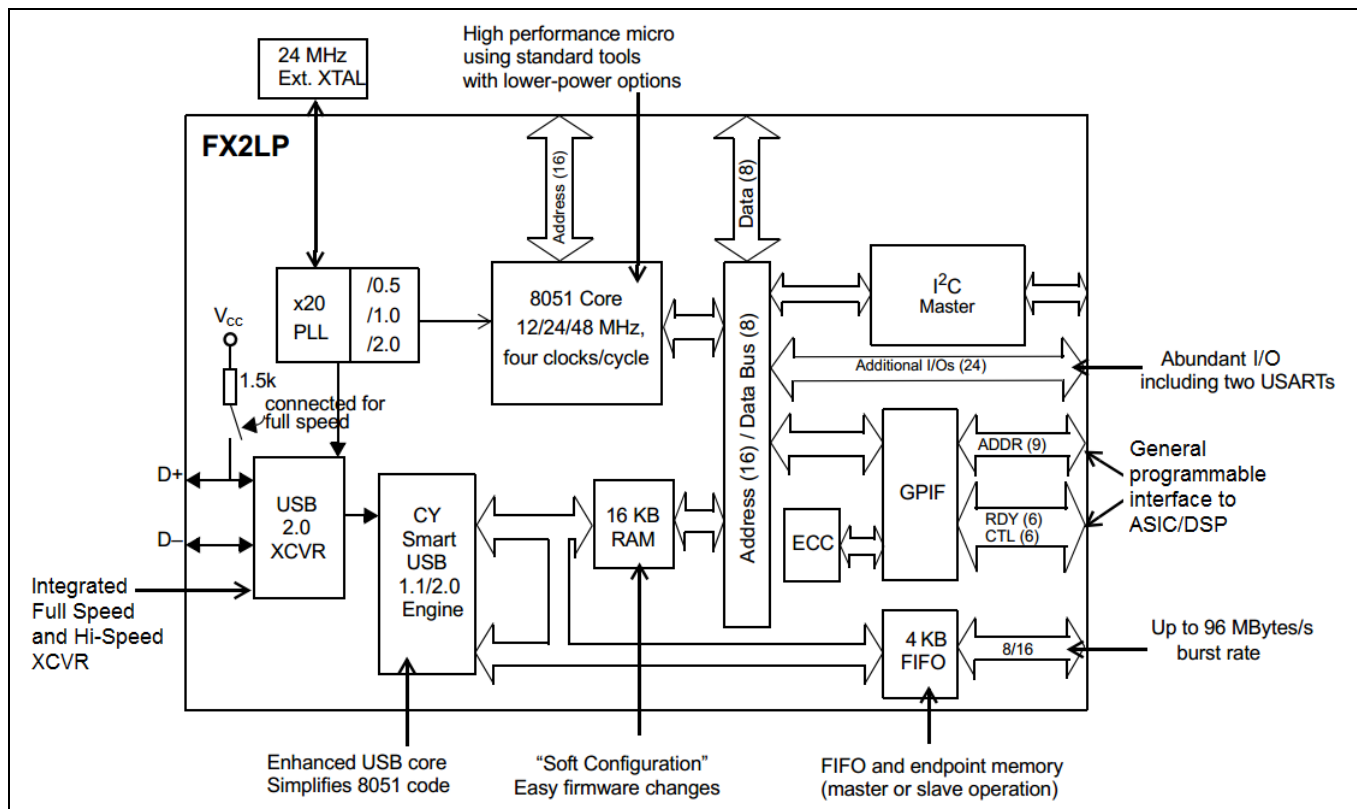


Figure 1 FX2LP ブロック図

FX2LP の機能は、USB ホストと周辺機器の間でデータを転送することです。FX2LP はパラレルとシリアルインターフェースを使用してこの接続機能を提供します。FX2LP の CPU が USB ホストに送信する前にデータを変更する必要がないシステムでは、FX2LP ファームウェアはそのハードウェア転送ユニットを初期化するだけで、高帯域幅 USB 転送を CPU 介入なしで実行します。

FX2LP は HID や大容量記憶装置やカスタム デバイスなどの標準 USB クラスに準拠するいかなる USB デバイス タイプとしてもエミュレートするようにプログラムすることができます。

3.1 FX2LP の機能

本節では、FX2LP の主な機能を簡単に説明します。

3.1.1 USB インターフェース

ハイスピード USB 転送は、パイプの両端をサポートするものと見なせます。**Figure 2** に示すように、一方の端でパイプは USB インターフェースを介して 480Mbps でデータを送受信します。パイプの他方の端は**パラレルインターフェース**セクションで説明します。

FX2LP の概要

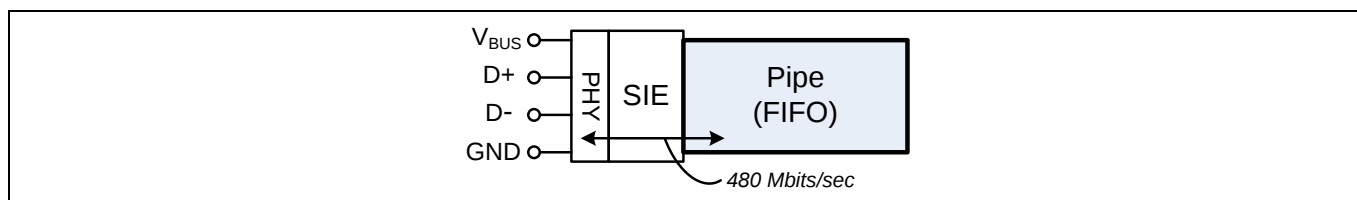


Figure 2 データパイプへの USB 接続

USB ホスト側に対応するために、FX2LP は以下のものを備えています。

- ハイスピードとフルスピード USB 転送に対応する USB 2.0 トランシーバを備えた物理バス インターフェース (PHY)
- スマートシリアルインターフェースエンジン (SIE)。SIE は PHY 信号をバイトに変換します。すべてのサイプレス USB SIE は、エラー訂正やパケット ID (PID) の同期化などの USB の低レベル要素を処理するロジックを追加しており、プログラマをそれらのタスクから解放します。このスマート SIE はエンドポイント FIFO メモリにバイトを転送します。またスマート SIE は、FX2LP を本格的な USB デバイスとしてエミュレートする論理も備えており、その内蔵 RAM にコードのロードもできます。
- すべての 4 つの USB 転送タイプ (制御、バルク、割り込み、アイソクロナス) に対応する 7 個のエンドポイント。その内、1 個は USB データのトランスミッタまたはレシーバとなります。
- 複数のバッファリングが可能な大容量エンドポイントバッファ (FIFO)。二重、三重、または四重バッファリングにより、USB 転送はペリフェラルに接続されたパイプラインを介して行われ、処理能力を向上させます。
- 低消費電力動作 (FX2LP の「LP」の意味)。USB VBUS ワイヤは 5V を供給し、周辺機器への制限された電源を提供します。FX2LP の低消費電力によって、バス電源供給 (バスパワー) 方式のアプリケーションを作成できます。例えば、FX2LP 開発キットはバスから電源供給されており、外部電源ユニットを不要にします。また FX2LP は、周辺機器がそれ自身に電源を供給するセルフパワー設計でも使用できます。

3.1.2 パラレルインターフェース

USB 2.0 仕様および豊富な FX2LP バッファリングが USB インターフェースを介してエンドポイント FIFO の内外へバイトを高速で転送することに役立ちますが、それは仕事の半分だけです。パイプの他方の端では、USB 転送レートに一致する速度で FIFO データをチップの内外へ転送する必要があります。FX2LP は、この目的に応じて特別に設計された 2 つのハードウェアインターフェースを備えています: スレーブ FIFO と汎用プログラマブルインターフェース (Figure 1 に示します)。

Note: 高速転送ロジック以外に、FX2LP 8051 は、USB と周辺機器の間で移動しているデータを翻訳または変更する必要のあるアプリケーションに対しエンドポイント FIFO へのアクセスを提供しています。

3.1.2.1 スレーブ FIFO

FX2LP は、MCU、FPGA や ASIC など FIFO コントローラーを内蔵した外部デバイスでの使用に対応してスレーブ FIFO インターフェースを備えています (Figure 3 を参照)。

FX2LP の概要

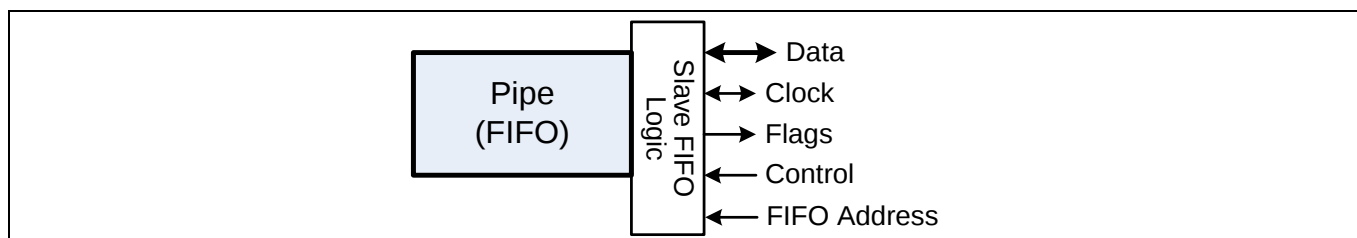


Figure 3 FX2LP スレーブ FIFO インターフェース

スレーブ FIFO インターフェースの対応を以下に示します。

- 8 または 16 ビット幅のデータバスを提供します。
- 非同期 (クロックを使用しない)、または同期して (クロックを使用) 動作します。
- (同期モードでは) 外部からクロックを受けるか、CLKOUT ピンを介して供給される内部 FX2LP 30/48MHz クロックを使用します。外部周辺機器は、CLKOUT ピンに供給されたクロックを使用して動作できるなら、水晶が不要です。
- 満杯や空等の FIFO 状態を示す出力フラグを有します。
- 制御入力 OE#、RD#、WR#を有します。
- 4 つの FX2LP FIFO の中から 1 つを選択するための 2 つのアドレスラインを有します。
- FIFO が満杯か空になると自動的に USB 転送を開始します。
- (FIFO にデータがあるが、満杯になっていない時) インターフェース デバイスが「ショート パケット」転送を開始するために使用する制御信号 (PKETND) を有します。ショート パケットは通常長い USB 転送の最後のパケットとして発生します。

FX2LP 内でのデータ処理を必要としないシステムでは、FX2LP スレーブ FIFO インターフェースは、スレーブ FIFO インターフェースを選択する、および FIFO の状態を示すフラグを設定するためだけ、最小限のファームウェアを必要とします。スレーブ FIFO インターフェースの波形とタイミング値については、[FX2LP データシート](#)を参照してください。

3.1.2.2 汎用プログラマブル インターフェース

すべての外部コントローラーが FIFO に接続するように設計されているとは限りません。このため、FX2LP は高速インターフェース GPIF ([Figure 4](#) に示す) を備えています。このインターフェースは、追加の外付け論理回路無しでディスクドライブ、FPGA、ASIC などの一般的なインターフェースへの直接接続を実現します。GPIF の中心部はステート マシンです。ユーザーは [GPIF Designer](#) を使用してステート マシンを制御する必要があります。GPIF は、すべての波形情報を格納するデータ構造体である 4 つの波形ディスクリプタのいずれかで制御します。GPIF Designer により、設計者はディスクリプタの形式を理解しなくても構いません。グラフィカルな波形エントリを使って、FX2LP プロジェクトに含めることができる C 言語のソース ファイルを作成します。

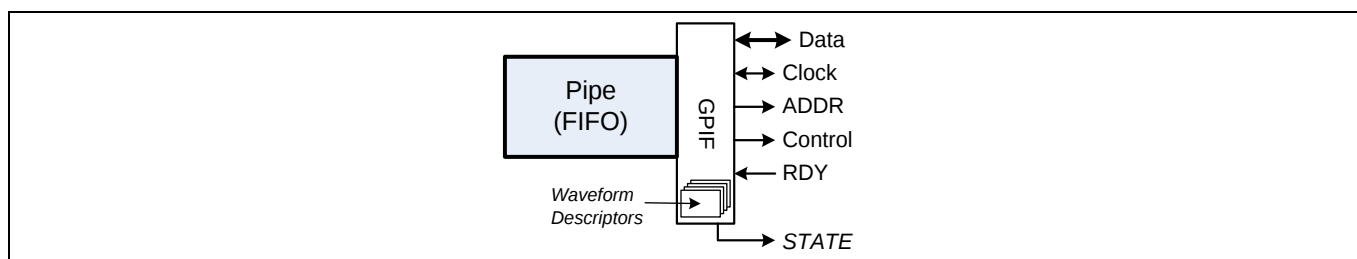


Figure 4 FX2LP GPIF Interface

FX2LP の概要

GPIF ([GPIF Designer](#) を参照) は:

- 8 または 16 ビット幅のデータバスを提供します。
- (クロックと)同期して動作します。
- 外部クロックを受け入れるか、内部 FX2LP 30/48 MHz クロック (同期モードで) を使用し、デバイスのピンに送信します。
- 9 本のアドレス出力を提供します。アドレスを初期化し、1 クロック毎にインクリメントすることができます。
- 6 本の制御信号 (CTL) とレディー信号 (RDY) を有します。6 本の制御出力も 1 クロック毎にプログラムできます。簡単な例として、これらの制御信号を使用して、期間と極性がプログラム可能な RD# と WR# ストロブを作ることができます。6 本のレディー入力は、GPIF ステートマシン内でテストして状態を読み出し、インターフェースの同期化を実行することができます。3 つのステート出力 (GSTATE [2:0]) は GPIF のステートを示し、ロジックアナライザを使ったデバッグ処理に役立ちます。

3.2 シリアル インターフェース

- USB およびペリフェラル インターフェースに加えて、FX2LP は以下のものを備えています。
- 100 または 400 kHz で動作する I²C マスター (専用)。
- 2 個の標準 8051 USART。これらは同期オプションが可能な標準 UART です。USART インターフェースピンは GPIO ピンに多重化されずに個別の I/O ピンとして使用できます。
- 最大 40 の GPIO。これらのピンは多目的であり、GPIO ピンまたはスレーブ FIFO や GPIF インターフェースに対応するピンとして機能します。[パッケージオプション](#)セクションではこれらのインターフェース オプションを示します。

3.3 CPU とメモリ

3.3.1 8051

- FX2LP は、2 個の USART、3 個のカウンター/タイマーおよび拡張割り込みシステムを備えた 8051 コアを搭載しています。コアは 48 MHz、24 MHz、または 12 MHz のクロックを使用します。CPU は内蔵コード/データ RAM の 16 KB でサポートされます。
- 拡張された割り込みシステムは「Autovector」メカニズムを使用して、サービスを必要とする USB 動作によって 27 の USB 割り込みサービスルーチン (ISR) のいずれかを自動的に呼び出します。ポインタハードウェアを自動的にインクリメントする (「Autopointers」) ことで、ブロック転送を加速させます。

3.3.2 ブート オプション

- FX2LP は RAM をプログラム記憶用に使用しています。FX2LP には、以下のブート オプションがあります。
 - USB ブート
 - I²C ブート
 - 外部パラレルメモリからのブート
- USB に接続された時、FX2LP のスマート SIE は、プログラムコードを FX2LP の内部 RAM にロードできる USB ブートローダとしてエミュレートします。コードがロードされた後、FX2LP はそれ自体を USB から電氣的に切り離して、ダウンロードされたコードで定義されたデバイスとして直ぐに再接続します。このプロセスは Re-Numeration™と呼ばれています。
- FX2LP のプログラム RAM は、電源投入時に外部シリアル EEPROM からロードすることもできます。ブート オプションの詳細については、[AN50963 - EZ-USB® FX1™/FX2LP™ Boot Options](#) を参照してください。

FX2LP の概要

3.3.2.1 パッケージ オプション

FX2LP には、**Figure 5** に示すように 3 つのパッケージ オプションがあります。

- 56 ピン SSOP、QFN、VFBGA
- 100 ピン TQFP パッケージ
- 128 ピン TQFP パッケージ

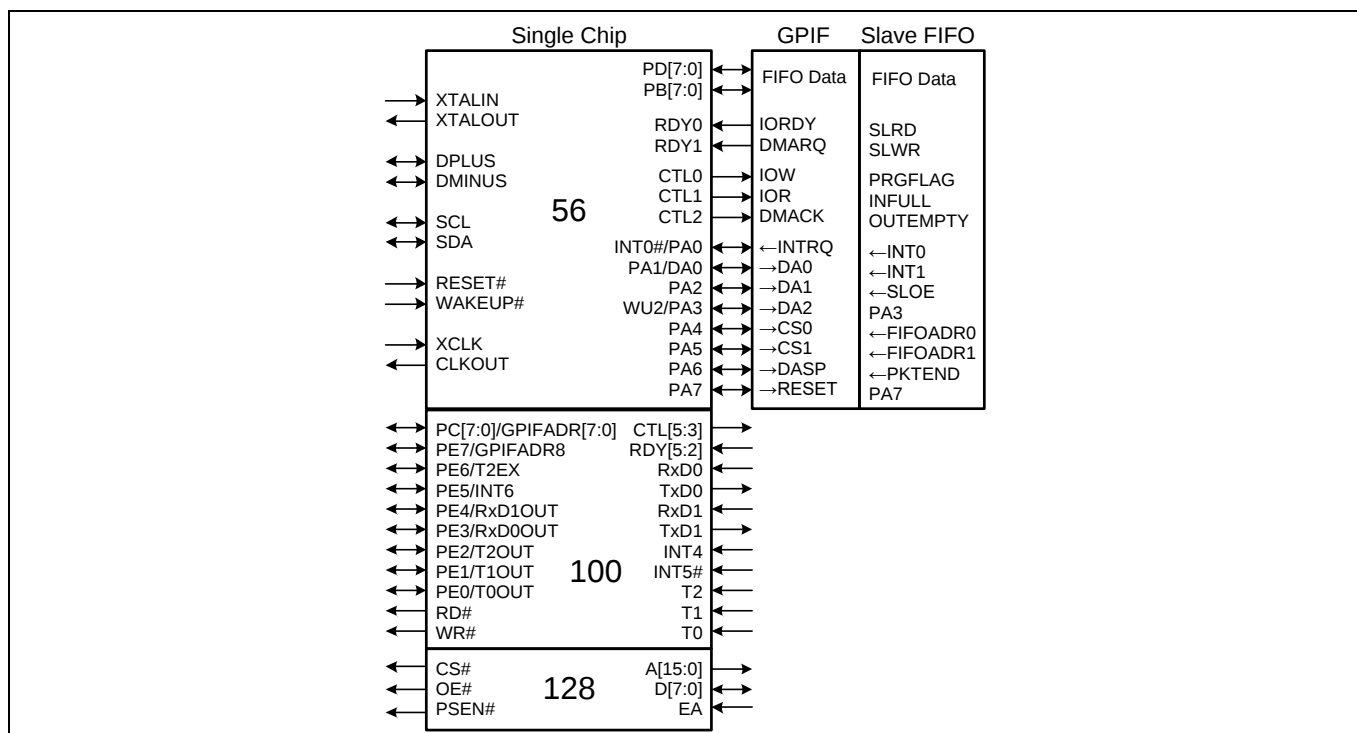


Figure 5 FX2LP の 3 つのパッケージ オプション

56 ピンパッケージは FX2LP の最も低コストのバージョンです。56 ピンブロックの左側 (**Figure 5**) はファミリのすべてのパッケージバージョンに共通しています。すべてのパッケージバージョンには、シングルチップ、GPIF、スレーブ FIFO の 3 つのモードがあります。これらのモードは 56 ピンブロックの右側に示す信号 (**Figure 5**) を定義します。8051 は、内部レジスタを使用してインターフェースモードを選択します。シングルチップモードは、電源投入時のデフォルト コンフィギュレーションです。

100 ピンパッケージは、44 ピンを 56 ピンパッケージに追加して機能を拡大します。

- 追加の 2 個の 8 ビット I/O ポート: PORTC と PORTE。
- 追加の 7 本の GPIF 制御信号 (CTL) とレディー信号 (RDY)。
- 9 本の非多重化制御信号 (2 本の UART、3 本のタイマー 入力、INT4、INT5#)。
- PORTE に多重化された追加の 8 本の制御信号。
- PORTC(8) と PORTE(1) に多重化された 9 本の GPIF アドレス ライン、
- PORTC の読み出しと書き込みストロープに使用できる RD#と WR#信号。

128 ピンパッケージは 8051 アドレス、データバスおよびそれらの制御信号を追加します。これらの追加のピンにより、FX2LP は外部 8051 メモリにアクセスできます。このパッケージは FX2LP 開発基板で使用されます。

FX2LP の概要

3.4 FX2LP のサンプル アプリケーション

3.4.1 スレーブ FIFO を使用して FPGA/ASIC と通信

Figure 6 に示す FPGA または ASIC は、FX2LP のスレーブ FIFO のピンに直接接続する FIFO コントローラを含んでいます。FPGA/ASIC は、データ ロガーやイメージ センサーなどアプリケーション固有のハードウェアにも接続しています。図に示すように同期 FIFO が IFCLK 信号を使用していますが、FX2LP は非同期 (クロックなし) FIFO インターフェースもサポートしています。

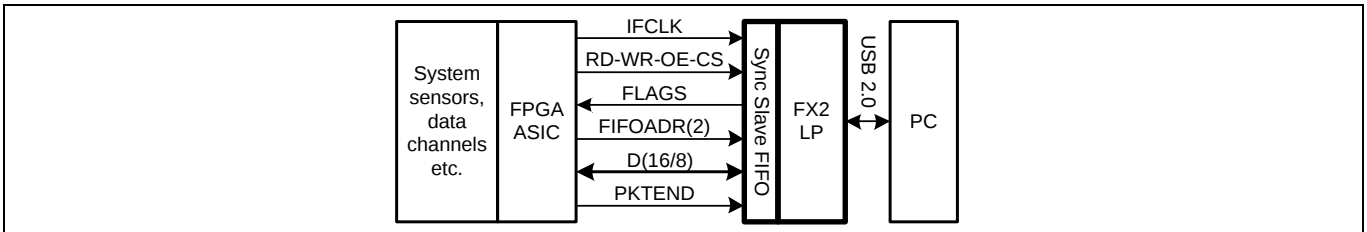


Figure 6 FPGA/ASIC は FX2LP を FIFO として取り扱う

参考資料

FX2LP のスレーブ FIFO インターフェースの詳細については、詳細な設計例を示す [AN61345 - Designing with EZ-USB FX2LP™ Slave FIFO Interface using FPGA](#) を参照してください。

FX2LP デバイスを使用するホスト PC や携帯電話向けイメージ センサーに対するインターフェースを持つ UVC フレームワーク実装のプロジェクト例については、[KBA222479 USB2.0 Camera Interface Using FX2LP™ and Lattice CrossLink FPGA](#) を参照してください。

必要な FIFO インターフェースの信号を変更せずに提供する幾つかのインターフェース チップがあります。例えば、MPEG デコーダは以下のようにその信号を FX2LP FIFO にマッピングできます：

Table 1 MPEG デコーダの FX2LP への接続

MPEG デコーダの信号	FX2LP の信号
MPEG_CLK	IFCLK
MPEG_SYNC	PKTEND#
MPEG_VALID	SLWR#
D[7:0]	FD[7:0]
外部チューナ制御	I ² C バス

3.4.2 参考資料

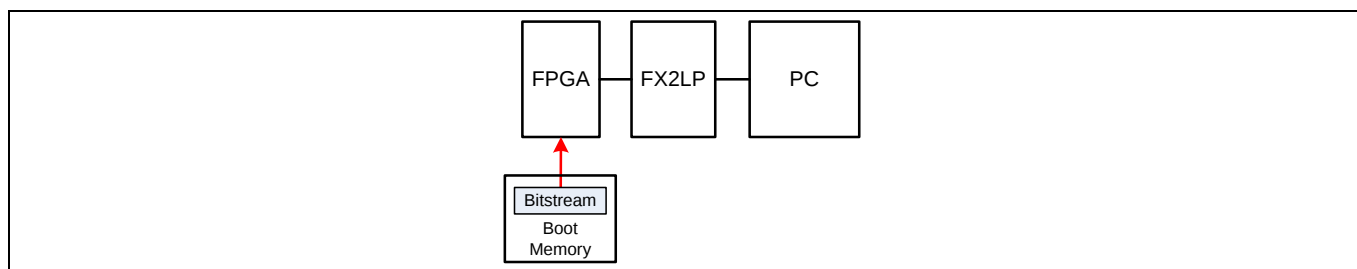
サイプレスは、[FX2LP DMB-T / H TV Dongle Reference Design](#) に記載されている「TV Dongle」の参考設計を提供しています。 <http://www.cypress.com/?rID=37775>

外部チップが **Table 1** のように FX2LP のスレーブ FIFO の信号に正確にマッピングしていない場合、FX2LP GPIF は追加の外部ロジックを必要としないで要求された信号に一致するようプログラムできます。

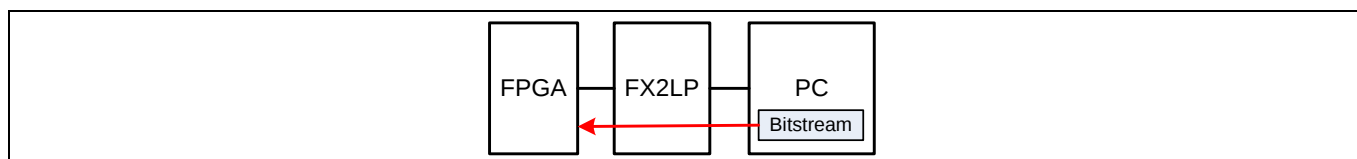
3.4.3 FPGA の FX2LP からのブート

FX2LP は、効率的なシステム統合には最も重要なものです。**Figure 7** と **Figure 8** は、FX2LP が外部ロジックの必要をなくす方法を示します。

FX2LP の概要

**Figure 7** FPGA が外部メモリから起動

電源投入時に、FPGA は外部メモリを使用してそのコンフィギュレーションビットストリームを起動します (**Figure 7**)。

**Figure 8** FPGA が FX2LP を介して起動

FX2LP は、**Figure 8** のように PC から FPGA ビットストリームをロードするよう設定することができます。これにより、ブートメモリが不要になり、PC からの FPGA の更新が可能になります。

参考資料

Xilinx Spartan-3E FPGA を使用した実装例については、[AN63620 - Configuring a Xilinx Spartan-3E FPGA Over USB Using EZ-USB FX2LP™](#)を参照してください。

サイプレスの設計リソース

4 サイプレスの設計リソース

サイプレスの **FX2LP 設計リソース** は、データシート、アプリケーションノート、評価キット、参考設計、ファームウェアの例およびソフトウェアツールを含んでいます。リソースは **Table 2** でまとめます。

Table 2 FX2LP リソースのまとめ

設計	利用可能なリソース	リソースへのアクセス
ハードウェア	開発基板 - 回路図、基板ファイル、資料	開発キット (DVK) 回路図 基板ファイルは FX2LP DVK のインストールに含まれる DVK ユーザー ガイド DVK クイック スタート ガイド
	水晶振動子、電源供給用のデカップリングコンデンサ、PCB レイアウトなどの推奨事項が記載されているハードウェア設計ガイドライン	アプリケーションノート - AN15456
	IBIS モデル	http://www.cypress.com/?id=193&rtID=114
FX2LP のファームウェア	Keil IDE の無料バージョン (最大 4KB のコード サイズ)	FX2LPDVK のインストールに含まれる
	ファームウェア例	
	FPGA を持った完全な設計での同期スレーブ FIFO ファームウェア例	アプリケーションノート - AN61345
ファームウェアデバッグ	Keil デバッガ環境のセットアップ、使用、トラブルシューティング	アプリケーションノート - AN42499
	FX1/FX2LP ファームウェアのシリアル (UART) ポート デバッグ処理	アプリケーションノート - AN58009
ホスト PC のソフトウェア	USB2.0 ドライバー - cyusb.sys	Suite USB のインストールに含まれる。 Suite USB のインストール ファイル (.exe) は FX2LP DVK のインストールにも含まれる Mac OS 用の Suite USB も利用可能。 Linux 用の FX3 SDK を使用して Linux プラットフォーム用に Control Center と同様なホスト アプリケーションを取得
	ホスト アプリケーション例: Control Center とストリーマ アプリケーション	
GPIF インターフェイス設計	GPIF 波形を設計し、ファームウェアに統合するコードを生成するために使用する GPIF Designer ツール	GPIF Designer のインストールに含まれる GPIF Designer のインストール ファイル (.exe) は FX2LP DVK のインストールにも含まれる
	一般的に利用されている GPIF 実装例	アプリケーションノート: AN57322 - Interfacing SRAM with FX2LP over GPIF AN66806 - EZ-USB® FX2LP™ GPIF Design Guide AN63787 - EZ-USB® FX2LP™ GPIF and Slave FIFO Configuration Examples using 8-bit Asynchronous Interface

サイプレスの設計リソース

設計	利用可能なリソース	リソースへのアクセス
	GPIF およびツールの使用方法に関する資料	GPIF Designer ユーザー ガイド - GPIF Designer ツールに含まれる

その他の資料

FX2LP データシート	http://www.cypress.com/?rID=38801
FX2LP テクニカル リファレンス マニュアル	http://www.cypress.com/?rID=38232
アプリケーション ノート	http://www.cypress.com/?id=193&rtID=76
リファレンス デザイン	http://www.cypress.com/?id=193&rtID=201
知識ベース記事	http://www.cypress.com/?id=193&rtID=118
USB 2.0 に関する資料	http://www.beyondlogic.org/usbnutshell/usb1.shtml AN57294 - USB 101: An Introduction to Universal Serial Bus 2.0
サードパーティの開発キット	http://www.ztex.de/usb-fpga-1 http://www.opalkelly.com/products/xem6010/ これらの開発キットの特長は「 付録 C: サードパーティの開発キットと SDK 」に記載

FX2LP 開発ツール

5 FX2LP 開発ツール

FX2LP ファームウェアの開発とテストの手順は **Figure 9** に示します。

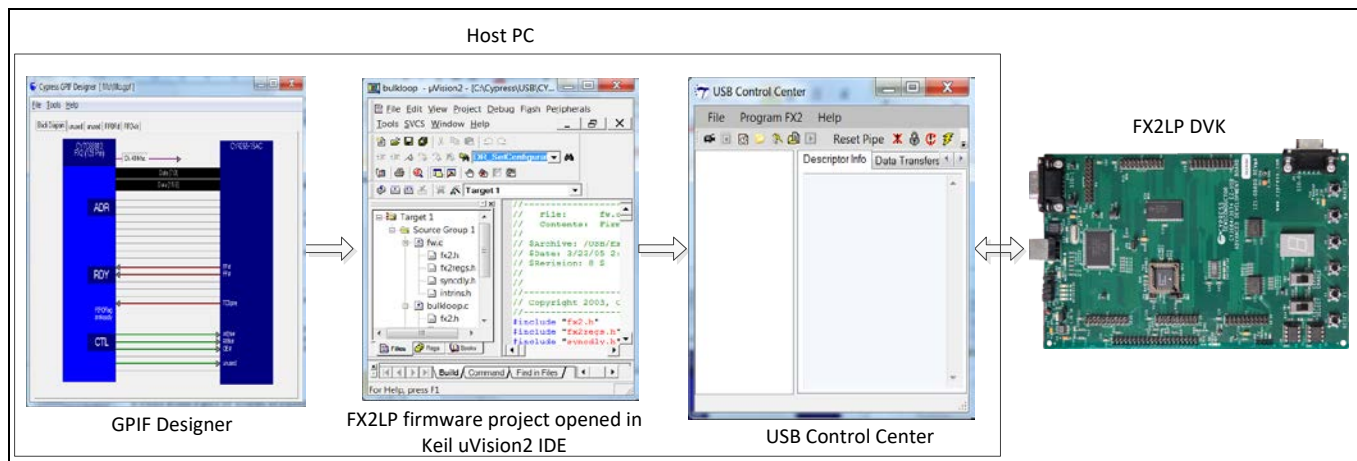


Figure 9 ファームウェアの開発とテスト手順

1. GPIF Designer を使用してインターフェース波形を開発して、FX2LP をその GPIF に接続された周辺機器と通信させます。.c ファイルをエクスポートし、.c ファイルを FX2LP ファームウェア プロジェクトに統合します。GPIF を使用しない場合、このステップは不要です。GPIF Designer の使用は、**GPIF Designer** で詳しく説明します。
2. サイプレスが提供するファームウェアフレームワークを使用して FX2LP ファームウェアを開発します。ステップ 1 でエクスポートされた.c ファイルを統合します。Keil uVision2 IDE を使用してファームウェアプロジェクトをビルドします。このステップでは、.hex と .iic ファイルが生成されます。.hex ファイルは FX2LP の RAM をプログラムするため、.iic ファイルは FX2LP に接続された EEPROM をプログラムするためです。FX2LP ファームウェアフレームワークおよび FX2LP ファームウェアの構築手順は、**FX2LP ファームウェア開発**で説明します。
3. Control Center アプリケーションを使用して、FX2LP の RAM または FX2LP に接続された EEPROM をプログラムします。Control Center アプリケーションの使用は、**USB Control Center を使用**で説明します。
4. FX2LP DVK を使用してファームウェアの機能を検証します。FX2LP DVK の詳細情報は、**5.1** および **付録 A: FX2LP 開発キット (DVK)**で説明します。

本アプリケーションノートに添付された zip ファイルは以下のディレクトリ構造を含んでいます。

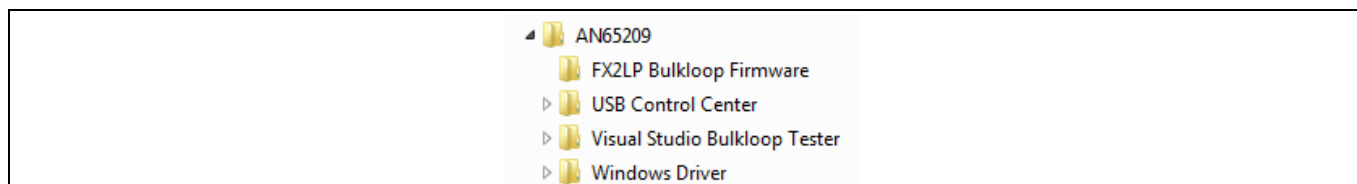


Figure 10 アプリケーションノートフォルダ

Note: 解凍したアプリケーションノート フォルダには、ダウンロードできるサイプレスの様々なサンプルコードとツールがあります。便宜上、すべてが1つのフォルダに収集されています。新しい情報については、**サイプレスのウェブサイト**を参照して最新のファイルバージョンを取得してください。

FX2LP 開発ツール

本節では、*bulkloop* という FX2LP ファームウェア例を作成し、FX2LP 開発キットにロードする説明を含んでいます。説明を進めながら、FX2LP を使用した USB 周辺機器設計を容易にする様々なツールを紹介します。FX2LP DVK を使用しない場合でも、使用可能な FX2LP チップとツールの概要を説明します。本節は、Windows アプリケーションを作成し、実行して *bulkloop* 設計をテストする方法を説明します。

5.1 FX2LP 開発基板

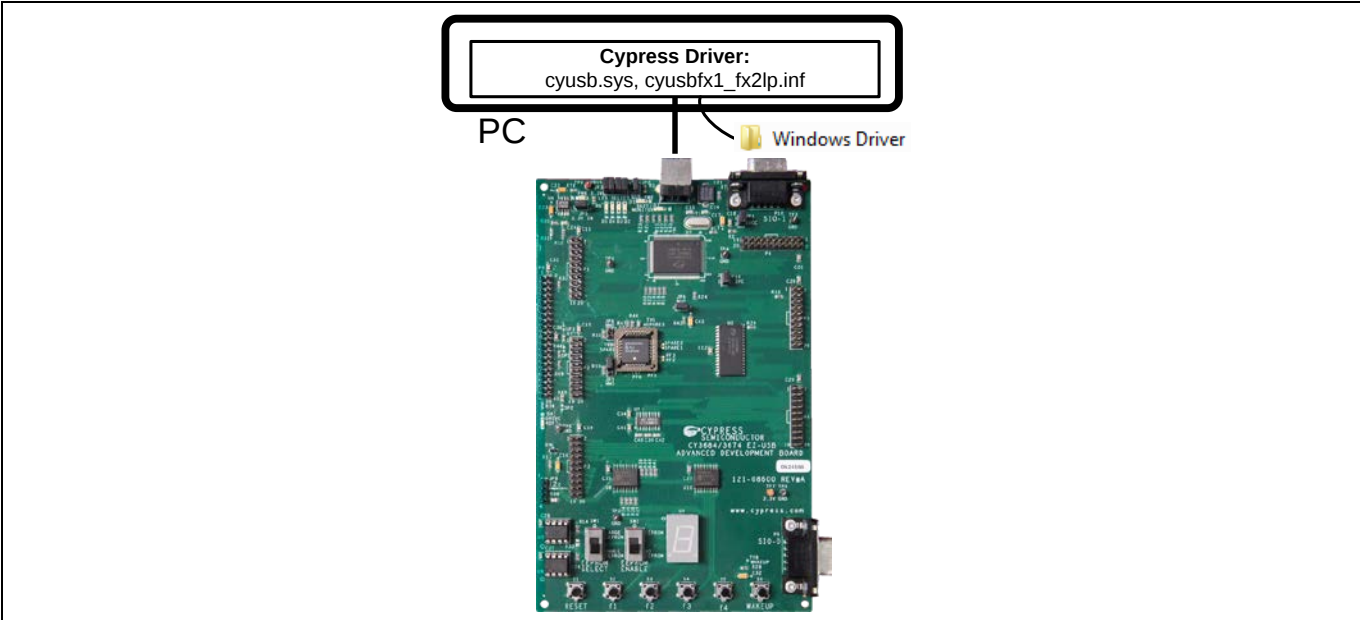


Figure 11 FX2LP 開発キットと Windows ドライバー

FX2LP 開発キットは <http://www.cypress.com/?rID=14321> にあります。

また、Keil uVision2 IDE (オブジェクト サイズが最大 4 KB に制限) の無料バージョンと C コンパイラを含む開発ソフトウェアパッケージはこのキット (無料でダウンロード) にも含まれています。DVK は付録 A で詳しく説明します。

基板を初めて PC に接続する手順は以下の通りです。

1. Table 3 に示すように FX2LP 基板のジャンパを用意します。

Table 3 EZ-USB FX2LP 基板のジャンパの設定

JP	ステート	用途
6、7	OUT	メモリを開発用に設定
2	IN	USB コネクタから基板へ電源を供給
1、5、10	IN	ローカル 3.3 V 電源
3	IN	すべての 4 個のジャンパが IN、4 個の LED (D2~D5) を起動
8	両方	未使用 (リモート ウェイクアップテスト用)

2. 基板の左下隅で、EEPROM ENABLE スライドを「NO EEPROM」の位置 (下) に切り替えます。これで、FX2LP チップはコード ロードアとしてエニユメレートします。別のスライド スイッチ (EEPROM SELECT) は両方の位置に切り替えられます。
3. FX2LP 基板を PC USB ポートに接続します。これが初めての場合、USB ドライバーをインストールするようポップアップメッセージが表示されます。アプリケーション ノート ドライバーのフォルダに移動して Windows OS に対応するサブフォルダを選択します。

FX2LP 開発ツール

Windows の Device Manager でドライバーが正常にインストールされたことを確認できます。

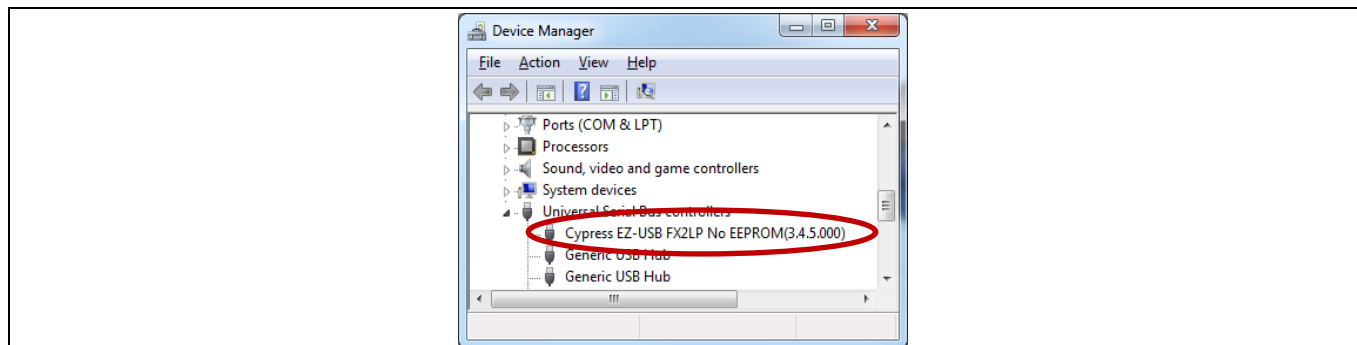


Figure 12 FX2LP 基板ドライバー (ブートローダ) が完了

ドライバーがインストールされた後、FX2LP 開発基板と通信するあらゆる Windows アプリケーションでこのドライバーを認識できます。

bulkloop.hex ファイルを作成する方法を説明する前に、次の 2 つの節には、2 つの Windows アプリケーションを使用して FX2LP ファームウェアをロードしテストする方法を示します。

5.2 Bulkloop デモを実行

5.2.1 USB Control Center を使用

Figure 13 に示すフォルダには、USB Control Center という Windows アプリケーションを作成するための Microsoft Visual C#ソリューションがあります。このアプリケーションは FX2LP ファームウェア (HEX ファイル) を FX2LP チップにダウンロードするために使用されます。ファームウェアがロードした後、FX2LP 基板は自動的にそれ自体を USB から切り離して、ロードされた HEX ファイルで定義されたデバイスとして再接続します (ReNumeration™)。ロードされた HEX ファイルがカスタムのベンダー ID (VID) と製品 ID (PID) を持っている場合、これらの値をサイプレスの .inf ファイルに追加する必要があります。

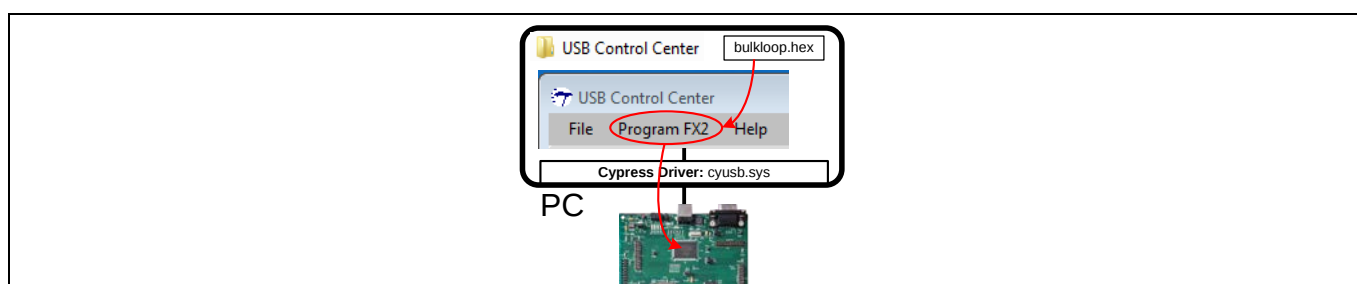


Figure 13 USB Control Center は HEX ファイルを FX2LP 開発基板にロード

参考資料

カスタム VID と PID を .inf ファイルに追加する詳細については、[付録 E: .inf ファイルへのカスタム VID と PID の追加](#)を参照してください。

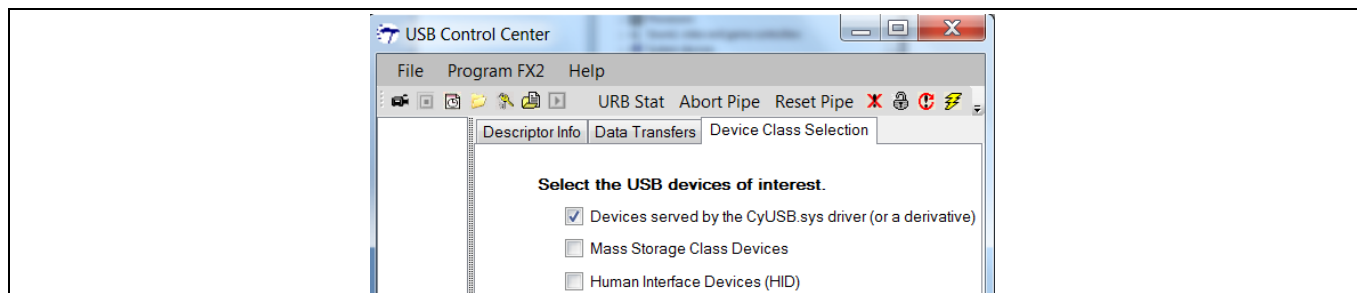
USB Control Center には、*bulkloop* アプリケーションやユーザーが書いたファームウェアをテストするために FX2LP DVK の内外への USB 転送を開始できるようにする「Data Transfer」タブがあります。

添付 zip ファイルには、Visual Studio ソリューション (**AN65209\USB Control Center**) およびその 2 進数にコンパイルされたバージョンがあります。試しに、以下の手順を行ってください。

FX2LP 開発ツール

1. デバッグまたはリリースのフォルダ (**AN65209\USB Control Center\bin**) の **CyControl.exe** ファイルをダブルクリックします。

FX2LP 基板および他の接続された USB デバイスが左パネルに表示されます。これを簡略化して FX2LP 基板だけを表示するために、右パネルの **Device Class Selection** タブをクリックし、**Devices served by the CyUSB.sys driver (or a derivative)** アイテムを除いてすべてのチェックボックスをオフにします。



そうすると、左パネルは **Figure 14** のようになります。

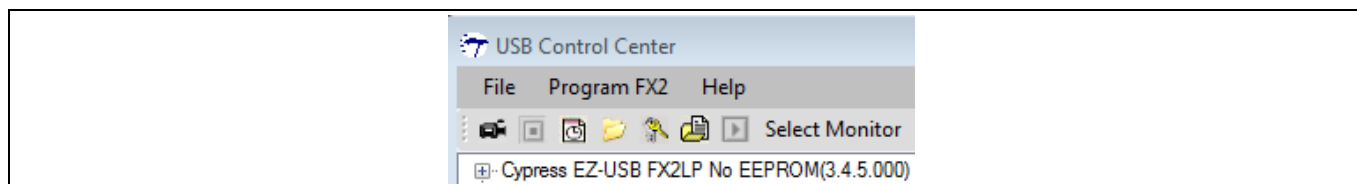


Figure 14 USB Control Center が FX2LP 基板を検知

2. これで、FX2LP ファームウェア **bulkloop.hex** (**AN65209\FX2LP Bulkloop Firmware**) を FX2LP DVK にロードする準備ができました。サイプレス デバイスをクリックしてハイライトし、**Program FX2 > RAM** を選択します。**FX2LP Bulkloop Firmware** フォルダを開き、**bulkloop.hex** ファイルを選択します。コードをロードした後、FX2LP 開発キットはそれ自体を USB から切り離して、ロードされたファームウェア **bulkloop.hex** で作成されたデバイスとして再接続します (**Figure 15**)。これで、サイプレス ReNumeration™を実演しました。FX2LP は 1 回目はコード ロードャとして、2 回目はロードされたデバイスとして、2 回エニユメレートしました。

LED D5 は、ハイスピード USB が接続した場合毎秒 8 回点滅し、フルスピード USB が接続した場合毎秒 1 回点滅します。FX2LP DVK 上の LED D5 のトグル周波数を観察すると、これが分かります。ハイスピード USB を接続した場合、LED が約 1 秒間ゆっくり点滅して、その後速く点滅し始めます。その理由は、USB デバイスが最初はフルスピードで接続して、その後ホストとハイスピード接続を折衝するためです。これは、**FX2LP ファームウェア開発**で説明するサイプレス USB フレームワークで処理する多くのファームウェア要素の一つです。FX2LP DVK 上の 7 セグメント表示器はホストに戻って転送される準備ができたパケット数を示します。これで、受信されて「ループバック」されたパケットがまだ無いため、「0」の数字で点灯します。

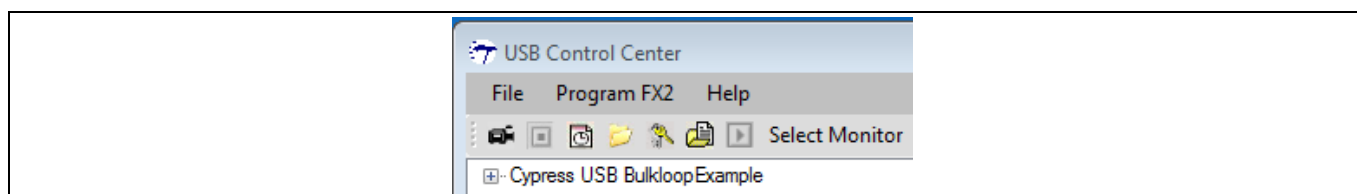


Figure 15 FX2LP 基板が再表示

FX2LP 開発ツール

Note: FX2LP DVK に HEX ファイルをロードする前に、まず「EEPROM ENABLE」スイッチを「NO EEPROM」の位置のままにし、リセットボタン(**Figure 11**、左下隅)を押してFX2LP をリセットし、それによってFX2LP ブートローダを有効にする必要があります。

Figure 16 には、USB Control Center と、**bulkloop.hex** ファイルをロードした FX2LP 開発基板との対話を示します。**bulkloop** ファームウェアは PC 接続に応じてフルスピードまたはハイスピード デバイスとしてエミュレートし、OUT エンドポイント 2 から受信した BULK データを PC に戻って送信するために IN エンドポイント 6 バッファにコピー (ループ) します。図に示すように、エンドポイントは 2 重バッファリングされています。この機能は本節で説明します。

Note: USB の方向はホストを中心にします: OUT はホストからデバイスへ方向を意味し、IN はデバイスからホストへ方向を意味します。

また **bulkloop** ファームウェアは、FX2LP 開発基板上の LED や 7 セグメント表示デバイスの制御やエンドポイント割り込みの使用などの便利な操作のコードも含んでいます。このコードは **FX2LP ファームウェア開発** で詳しく説明します。

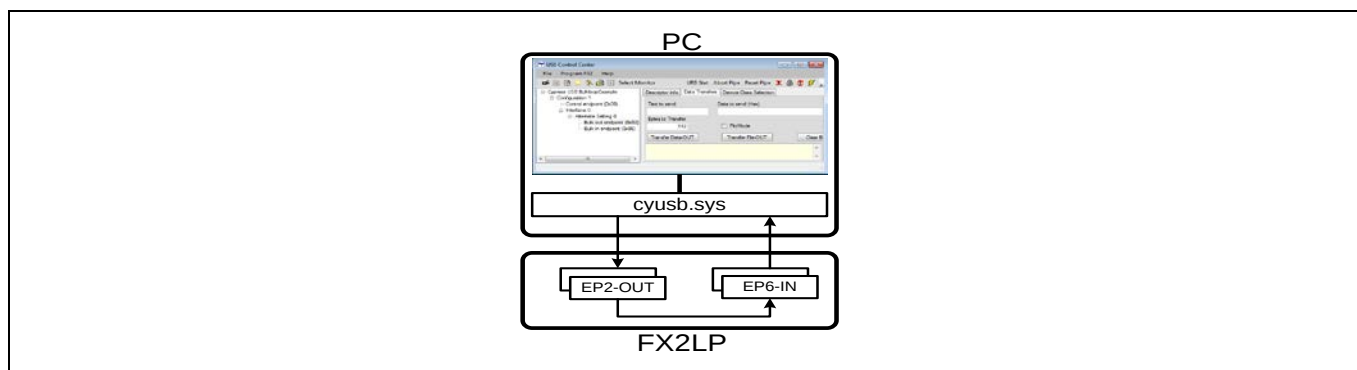


Figure 16 USB Control Center は Bulkloop ファームウェアをテスト

1. Control Center で Bulkloop 例のツリービューを拡大して、実施された BULK エンドポイントを表示します (**Figure 17**)。

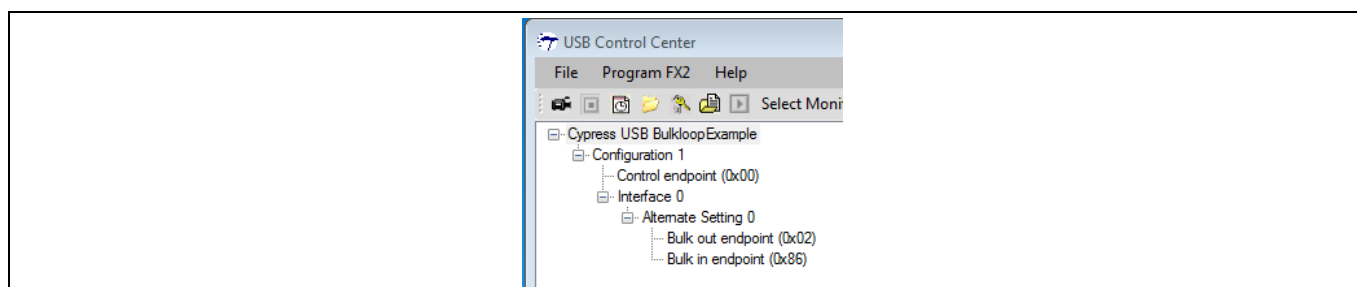


Figure 17 サンプル Bulkloop のデバイス内部

2. **Data Transfer** タブを選択します。左パネルで **Bulk out endpoint (0x02)** エントリをクリックし、転送データボタンが **Transfer Data-OUT** であると確認します。このボタンをクリックすると、
 - 512 バイト (初期設定値が 0) は PC から FX2LP DVK に転送されます。
 - LED D3 は点滅して OUT 転送を示します。
 - 7 セグメント表示器は 1 にインクリメントされ、1 つのパケットが EP2-OUT を介し受信されて EP6-IN エンドポイント FIFO にロードされ、ホストへの転送の準備ができたことを示します。

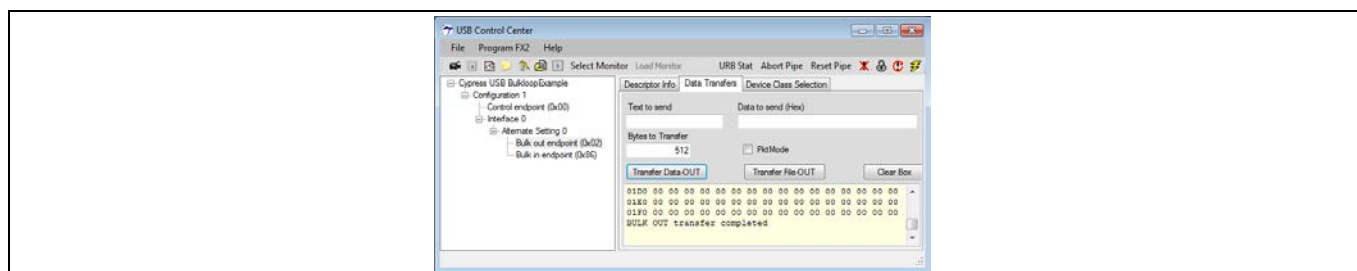


Figure 18 成功した BULK-OUT 転送

1. **Transfer Data-OUT** ボタンをもう一度クリックします。PC は 2 番目のパケットを FX2LP に転送し、7 セグメント表示器は 2 にインクリメントされます。
2. **Bulk in endpoint (0x86)** エントリをハイライトします。これで、転送ボタンは **-IN** を示します。このボタンをクリックすると、
 - 512 バイトが FX2LP からホストに転送され、16 進数値として表示されます。
 - LED D2 は点滅して IN 転送を示します。
 - 7 セグメント表示器は 1 にデクリメントされます。
3. **Transfer Data-IN** ボタンをもう一度クリックします。キューに入った 2 番目のパケットはホストに転送され、7 セグメント表示器は待機しているパケットが無いことを示します。このシーケンスでは、2 個のエンドポイントの二重バッファリング動作を確認します。
4. **Bulk out** エンドポイントをもう一度選択します。次に、マウスカーソルを **Text to send** ボックス内に置いて「1234」を入力します。**Data to send** ボックスと **Bytes to Transfer** ボックス内に表示されている 16 進数値は入力した数字毎にインクリメントされます。**Transfer Data-OUT** ボタンをクリックします。
5. **Text to send** ボックスで、「abcdefg」を入力して、**Transfer Data-OUT** ボタンをクリックします。
6. 「Bulk in endpoint」を選択し、**Bytes to Transfer** を初期設定値の 512 バイトのままにします。**Transfer Data-IN** ボタンを 2 回クリックして、転送された 2 つのデータパケットを読み出します。

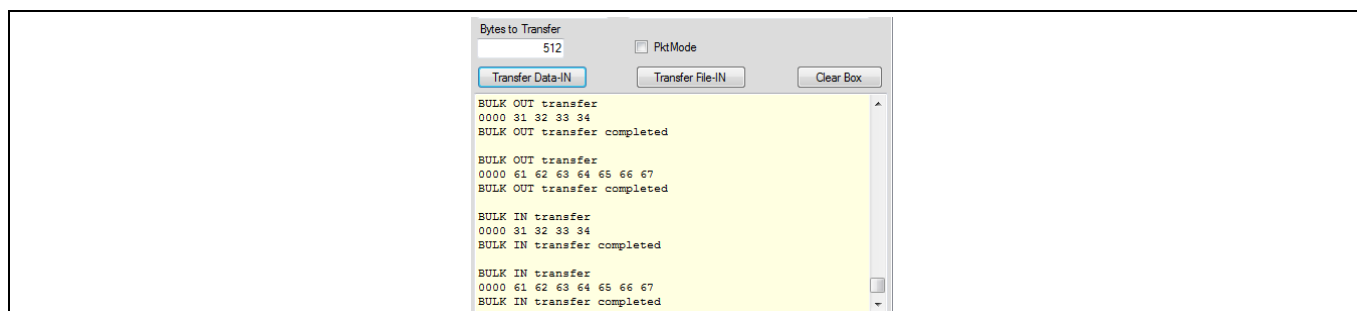


Figure 19 2 つの OUT パケット、2 つの IN パケット

Figure 19 に示すように、USB ペリフェラルが常に、要求されたバイト数(この場合、512)と使用可能なバイト数(4 または 7)の小さい方を送信するという USB 機能があります。

5.2.2 BULKLOOP_VCS の使用

Bulkloop_VCS という 2 番目の Visual Studio プロジェクトは、FX2LP 開発基板の内外へのスリム化された転送を実行します。転送されたバイトの累計を維持しながら BULK データを連続して送受信します。これは USB Control Center と同様、デバイスの特性情報を左パネルに表示します。USB Control Center と違う点は、**Start** ボタンが押されると、自動的に連続転送をセットアップして実行することです。

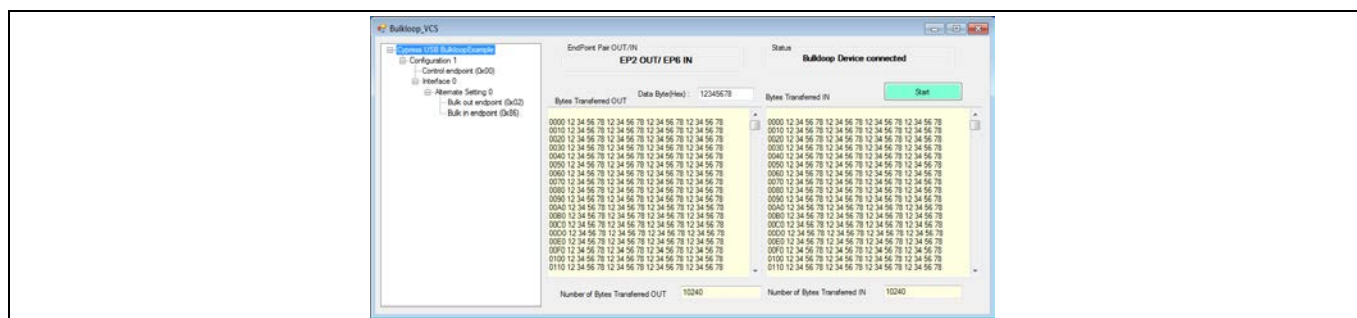


Figure 20 Bulkloop_VCS Window

このアプリケーションを使用する前に、前節で説明したように USB Control Center を使用して **bulkloop.hex** ファイルをロードします。「EEPROM ENABLE」スイッチを「NO EEPROM」の位置のままにし、FX2LP 開発キットの RESET ボタンを押して FX2LP ブートローダを有効にしたことを確認してください。

5.3 FX2LP ファームウェア開発

本節では、**bulkloop.hex** ファイルを生成する手順を説明します。

5.3.1 Keil uVision2

サイプレス FX2LP DVK は、8051 用の Keil uVision2 ツールの無料のデモバージョンを含んでいます。この IDE は十分な機能を有していますが、最大コードサイズが 4096 バイトに制限されます。**bulkloop.uv2** のようなサイプレス デモ プロジェクトはこの制限を満たしているため、コードを検討し、変更できます。より大きなプロジェクトの場合、Keil 社から全部の uVision2 ツールセットを購入する必要があります。

5.3.2 サイプレス USB ファームウェアフレームワーク

サイプレスは、Keil 社のツールと共に低レベルの USB 要素を処理するために USB ファームウェアフレームワークのファイル群を提供しています。これにより、開発期間を特定のアプリケーションコードに集中させられます。

bulkloop.uv2 プロジェクト内には、以下のプロジェクト ファイルがあります。

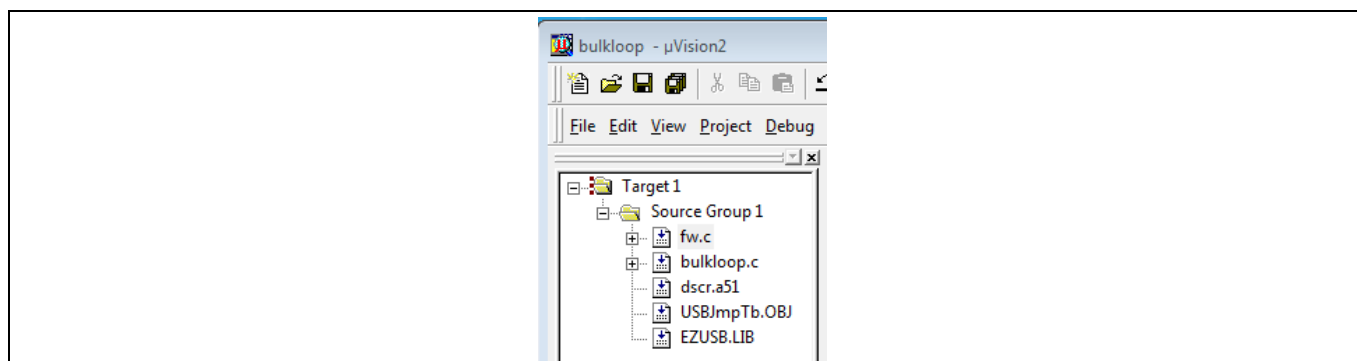


Figure 21 bulkloop.uv2 プロジェクト ファイル

- **fw.c** はサイプレス USB ファームウェアフレームワークです。

FX2LP 開発ツール

ファームウェアフレームワークの詳細については、[FX2LP Development Kit User Guide.pdf](#) の第 5 章を参照してください。

- **bulkloop.c** はアプリケーションです。他のサイプレスの例では、このコード モジュールを **peripheral.c** と呼ぶことがあります。プロジェクト名に応じてこのファイル名を変更することが勧められます。
- **dscr.a51** は USB デバイスをエミュレートするのに必要なデータを格納するアセンブリ言語ファイルです。このファイルは **.db** (define byte - バイト定義) ステートメント形式のテーブル データから構成されています。このファイルを編集して VID/PID や記述文字列などをカスタマイズします。
- **USBJmpTb.OBJ** は FX2LP アーキテクチャに必要なジャンプ テーブルであり、変更を全く必要としません。
- **EZUSB.LIB** は主に I²C 通信に使用する FX2LP 関数のライブラリを格納しています。ソース コードは FX2LP DVK に含まれており、**Target\Lib\LP** サブフォルダ内にあります。

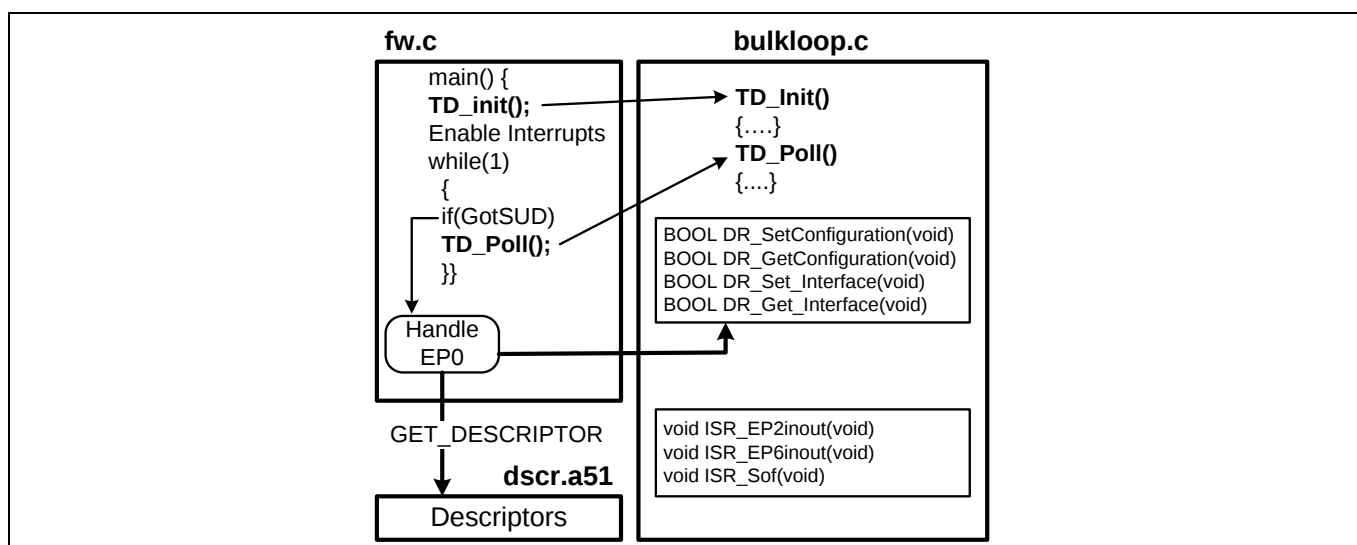


Figure 22 USB ファームウェア フレームワーク構造

Figure 22 は、コード モジュール間のやり取りを示します。

Fw.c は **main** 関数を含んでいます。エミュレーションなどの USB メンテナンスの多くを実行し、カスタマイズが必要な度にアプリケーション コード (**bulkloop.c**) 内の具体的に名付けられた外部関数を呼び出します。**Fw.c** はほとんどユーザーによる変更を必要としません。色々な準備手順を整えた後、**bulkloop.c** に用意してある **TD_init** という外部関数を呼び出します。(接頭辞 TD は「タスク ディスパッチャ」の略です。)

次に、CONTROL エンドポイント 0 を介して SETUP パケットの着信を確認する無限ループに移行します。ループは USB サスペンド イベントも確認しますが、**tbulkloop** アプリケーションでは使用されません。ループを通り抜ける度に、**bulkloop.c** に用意してある外部 **TD_Poll** 関数を呼び出します。このアプリケーションでは、**TD_Poll** 関数を使用して PC からエンドポイント データを FX2LP を介してループします。

すべての USB ペリフェラルはその CONTROL エンドポイントを介してエミュレーションとオペレーショナルという 2 タイプの要求を受信します。

FX2LP 開発ツール

エミュレーション

USB デバイスが接続されると、ホスト PC は複数の **GET_DESCRIPTOR** 要求を送信して、エミュレーションと呼ばれるプロセスの一部としてデバイスの種類とその要件を見つけます。**fw.c** コードはこれらの要求を捉えて、**dscr.a51** ファイルに格納された値を使用してそれら进行处理します。

USB フレームワークを使用する利点は、コードがテスト済みであり、USB の「第 9 章」の要件を満たすことが確認されていることです。第 9 章は、デバイスの (EP0 を介した) 要求とそれらの適切な応答を説明する USB 仕様の章です。

オペレーショナル

ユーザー コードが必要になると、**fw.c** は、**bulkloop.c** ファイル内に用意している DR 接頭辞 (デバイス リクエスト) 付きの具体的に名付けられた外部関数を呼び出します。*bulkloop* のような簡単なアプリケーションには、1 つのコンフィギュレーションと 1 つのインターフェースのみがあります。それで、**Figure 22** の 2 つの **DR_Set-Get** 関数のペアは単にホストが送信した SET 値を格納し、ホストが GET 要求を発行するとそれらを一時的に保存します。より複雑なコンフィギュレーションの場合は、これらの DR 呼び出し (「フック」) を使用してカメラの解像度を変更したり、2 つの異なるインターフェースに要求を送信したりできます。

bulkloop.c には任意の USB デバイス用に完全なテンプレートがあるため、それをカスタム実装の基本として使用できます。本節の残りの部分は、*bulkloop* アプリケーションを実施するためにユーザー コードを必要とするファイルの 3 つの部分について説明します。

5.3.2.1 TD_Init

この関数は次のことを行います:

- 8051 クロックを 48 MHz に設定します。
- 開発基板の 4 個の LED を消灯します。LED は、特定のメモリ領域を読み出してオン/オフにします。この方法は、I/O ピンを使用せずに LED を制御できます。
- EP2 を BULK-OUT エンドポイントとして、EP6 を BULK-IN エンドポイントとして設定します。両方は二重バッファリングされており、512 バイト FIFO を使用します (**Figure 16**)。
- 2 つのパケットを受信するために EP2-OUT を有効にします。OUT エンドポイントは、MSB がセットされたそのバイト カウント レジスタに任意の値を書き込んで有効にします。MSB (「SKIP ビット」と呼ばれる) をセットすると、パケットがスキップされます。
- FX2LP の 2 つの自動ポインタを有効にします。これらのハードウェア ポインタは、EP2-OUT バッファから EP6-IN バッファへの効率的なメモリ間のバイト転送向けに自動的にインクリメントされます。
- 3 つの割り込みを有効にします: SOF、EP2、EP6 エンドポイント割り込み。

5.3.2.2 TD_Poll

TD_Poll は **fw.c** 内にある無限ループで呼び出されます (**Figure 22**)。 *bulkloop* アプリケーションには 2 つのタスクのみが必要です。

1. ホストへの転送用に待機しているパケット数で 7 セグメント表示器を更新します。FX2LP の EP6CS (エンドポイント 6 制御とステータス) レジスタはビット 6~4 でこの数を提供します。

参考資料

レジスタの詳細情報については、**FX2LP テクニカル リファレンス マニュアル** の第 15 章を参照してください。

FX2LP 開発ツール

2. エンドポイント FIFO フラグを確認して、OUT パケットを IN バッファへ転送する時点を判断します。その時点では、自動ポインタを使用して EP2-OUT バッファからパケット データを EP6-IN バッファに転送します

転送の第 2 決定をする方法を理解するには、FX2LP のエンドポイント FIFO フラグに関する 2 つのポイントを理解するのが重要です。

- a) 複数のバッファリングが使用された場合、FULL と EMPTY フラグ (EP2468STAT レジスタ) は 1 つだけでなくすべてのバッファを反映します。したがって、この二重バッファリングの場合、1 つの OUT パケットが受信された時、2 番目のバッファがまだ OUT 転送が可能なので、FULL フラグはアサートされていません。FULL フラグは、2 番目のパケットが着信された時にのみアサートします。同様に、IN エンドポイント EMPTY フラグは、両方のバッファが空であり、8051 によって新しいデータで一杯にされる準備ができていない時にのみアサートします。
- b) FX2LP は、USB パケットの正常な受信／送信の後にのみ、(EP2468STAT レジスタの) FIFO フラグを更新します。

このため、以下の 2 つの条件が同時に満たされると、ループ転送が行われます。

- EP2-OUT が空でないこと
- EP6-IN が満杯でないこと

つまり、EP2 にはパケットがあり、EP6 にはパケット受信の余地がある場合です。このようなテスト方法は、いかなるパケット サイズにも対応でき、二重バッファリングを考慮に入れます。

5.3.2.3 割り込みサービス ルーチン

bulkloop.c ファイルは、すべての USB 割り込みソースに対応した ISR 関数が含まれます。幾つかの ISR は **fw.c** コード向けにフラグをセットし、他は USB エニュメレーション関数を実行するものです。ユーザーはアプリケーションに使用する ISR のコードを記入するだけで済みます。

以下の ISR は FX2LP USB 割り込み要求をクリアする方法を示します。

```
// Setup Token Interrupt Handler
voidISR_Sutok(void) interrupt 0
{
    EZUSB_IRQ_CLEAR();
    USBIRQ = bmSUTOK;      // Clear SUTOK IRQ
}
```

割り込みキーワードの後の「0」はすべての USB 割り込み要求の ID です。2 つの割り込み要求フラグは、まず一般 USB 割り込みフラグ、次に個別 USB ソース フラグ (この例では、「Setup Token Arrived」フラグ) という既定の順番でクリアされます。この ISR は「コード フック」の一例です。コードをこの ISR に記入することで、SETUP パケット着信時にあらゆる動作ができます。

bulkloop アプリケーションは、以下の 4 つのカスタマイズされた ISR を必要とします。

Set_Configuration ISR

ホストは、FX2LP コンフィギュレーションをエニュメレーション過程の最後のステップとします。この時点はアプリケーション ハードウェアを初期化する良いタイミングです。7 セグメント表示器を駆動する I²C ユニットはここで初期化されます。

FX2LP 開発ツール

EP2INOUT/EP6INOUT ISR

これらの IRQ、パケットの EP6-IN からの送信時、または EP2-OUT への到着時に実行されます。ISR のコードは FX2LP 開発基板上の LED をオンにし、の LED の点灯時間を制御するために、(**bulkloop.c** 内で) **inblink** (EP6-IN) または **outblink** (EP2-OUT) 変数を設定します。

SOF ISR

SOF ISR は便利なタイマーとされており、フルスピードでは 1 ミリ秒ごとに、ハイスピードでは 125 マイクロ秒ごとに実行されます。ISR コードは ISR を介して FX2LP 開発基板上の LED を 500 回ごとに (フルスピードでは 1 回/秒、ハイスピードでは 8 回/秒) トグルします。また、ISR のコードは、IN と OUT パケットが着信した時に設定された **inblink** と **outblink** 変数をデクリメントし、カウンタが 0 に達するとインジケータ LED をオフにします。

5.3.2.4 USB の 2 つのスピードに対応

USB 2.0 デバイスは、リセット後にフルスピードで動作します。ホストは、低レベルバスの信号方式を使用してデバイスとネゴシエートして動作速度を決定します。FX2LP は、ホストとデバイスのスピードネゴシエーションの結果としてハイスピード動作に切り替わったことを示す割り込みを発生させます。

USB 2.0 デバイスはフルスピードとハイスピードで動作する必要があります。ハイスピード USB デバイスは、2 つのディスクリプター式(それぞれの一式が 1 種のスピードに対応する)を提供しています。2 つの ISR は、スピードに適切なディスクリプタを指定することを担当します。

- **ISR_Ures** (USB リセット) コードは、フルスピードディスクリプタを「現時点のスピード」ディスクリプタとして指定し、ハイスピードディスクリプタを「他のスピード」ディスクリプタとして指定します。ハイスピード USB デバイスがフルスピードポートに差し込まれている場合、さらなる動作は不要です。
- **ISR_Highspeed** 割り込みサービスコードは現時点/他のディスクリプタの指定を交換します。ハイスピードは「現時点のスピード」となり、フルスピードは「他のスピード」となります。この指定方法により、Windows はハイスピードデバイスがフルスピードポートに差し込まれていることを検出すると、下図のメッセージを表示します。

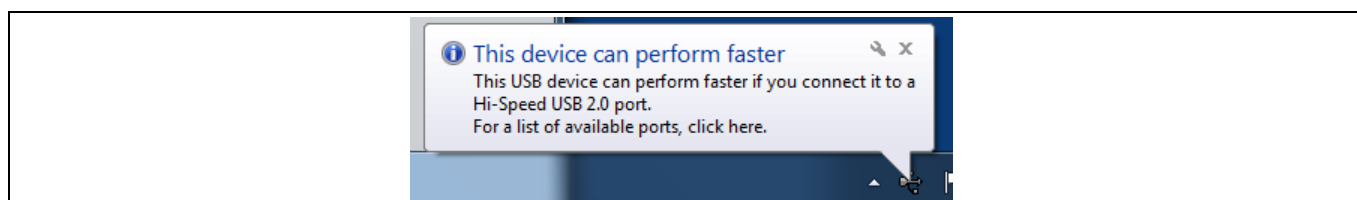


Figure 23 Windows はハイスピード デバイスが性能を向上できる時を認識する

この重要な USB の準備過程は、ファームウェア フレームワークがユーザーの代わりに行う作業の一例です。2 つのディスクリプタおよびディスクリプタ交換コードは既にかかれてあります。ユーザーは、アプリケーション固有のディスクリプタ フィールドに記入するだけです。

5.3.3 Bulkloop プロジェクトをビルド

FX2LP Bulkloop ファームウェア フォルダで、**bulkloop.uv2** ファイルをダブルクリックします。そうすると、uVision2 IDE が開かれ、bulkloop がロードされます。プロジェクトをコンパイルしリンクさせるために、**Rebuild All Target Files** ボタンをクリックします。

FX2LP 開発ツール

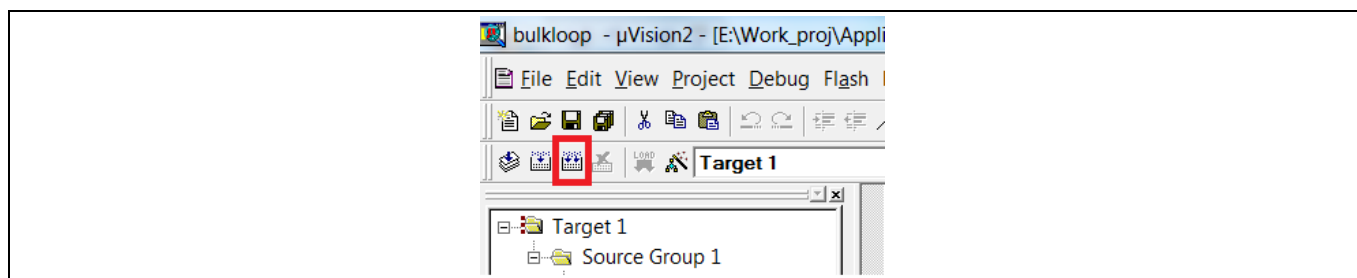


Figure 24 uVision2 IDE

これにより、前節のダウンロード対象の *bulkloop.hex* ファイルが作成されます。

Keil uVision2 をインストールする際に以下のことに注意してください。

- ウェブダウンロードの zip ファイルを解凍した後、読み出し専用ファイルとしてインストールすることがあります。この場合、フォルダを右クリックし、「Properties」を選択し、「Read-only」ボックスのチェックを (入れている場合) チェックを外します。
- Keil ビルドの最後のステップとして、Keil 出力を iic ファイルに変換する **hex2bix.exe** プログラムが自動的に実行されます。このファイルの保存場所はプロジェクトにコード化されているため、Keil ツールを移動またはインストールするとパスを破損することがあります。簡単な修正方法として、**hex2bix.exe** ファイルを Keil プロジェクトフォルダに入れ、パスを指定せずにこのユーティリティを起動します。(これは、本アプリケーションノートに添付されたコードで行われます。) Keil プロジェクトウィンドウのトップファイル (デフォルト ファイル名「Target 1」) を右クリックし、「Options...」を選択し、「Output」タブを選択することでパスを検索できます。パスは以下のように見えます。

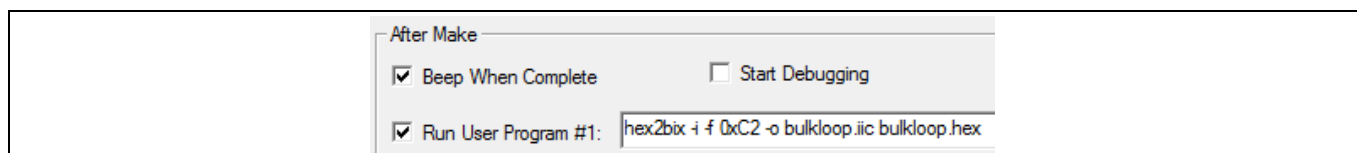


Figure 25 hex2bix ユーティリティへのパス

5.4 GPIF Designer

bulkloop の例では、すべてはチップ内で処理されます。8051 はフラグの確認、エンドポイント FIFO 間のバイト移動、開発キット上のステータス インジケータ (LED と 7 セグメント表示器) の更新を行います。ほとんどの FX2LP 設計は、それより複雑であり、FX2LP とディスクドライブやイメージセンサーなどの外部ペリフェラルとの間でデータを転送します。このために、GPIF は外部システムとの高速でフレキシブルなインターフェースを提供します。本節は、サイプレス GPIF Designer ツールの使用に対応して順を追って命令を説明している [AN66806 - Getting Started with EZ-USB FX2LP GPIF](#) の概要を述べます。

GPIF 設計の最初のステップとしては、[Figure 26](#) に示すようにブロック図にインターフェース信号を描きます。

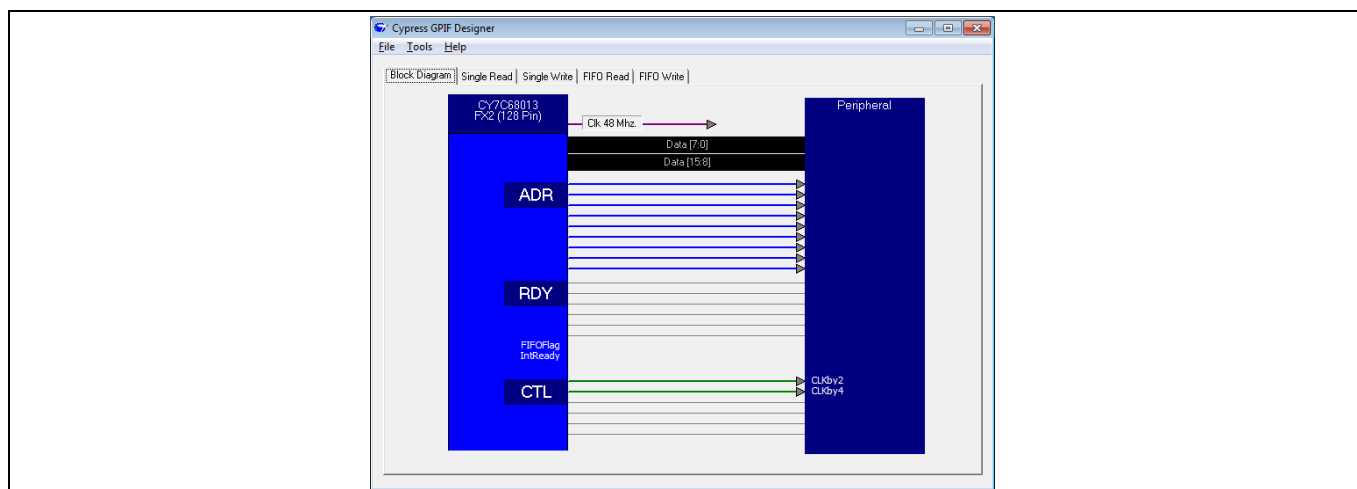


Figure 26 GPIF Designer の「Block Diagram」タブ

GPIF は、6 本の制御 (CTL) 出力と 6 本のレディー (RDY) 入力を持っています。これらの信号は、ブロック図で RDY または CTL のテキスト ブロックを右クリックして名前を変更できます。新しい名前は GPIF Designer で伝え、ステートマシンの決定点 (decision point) と波形名として選択できます。

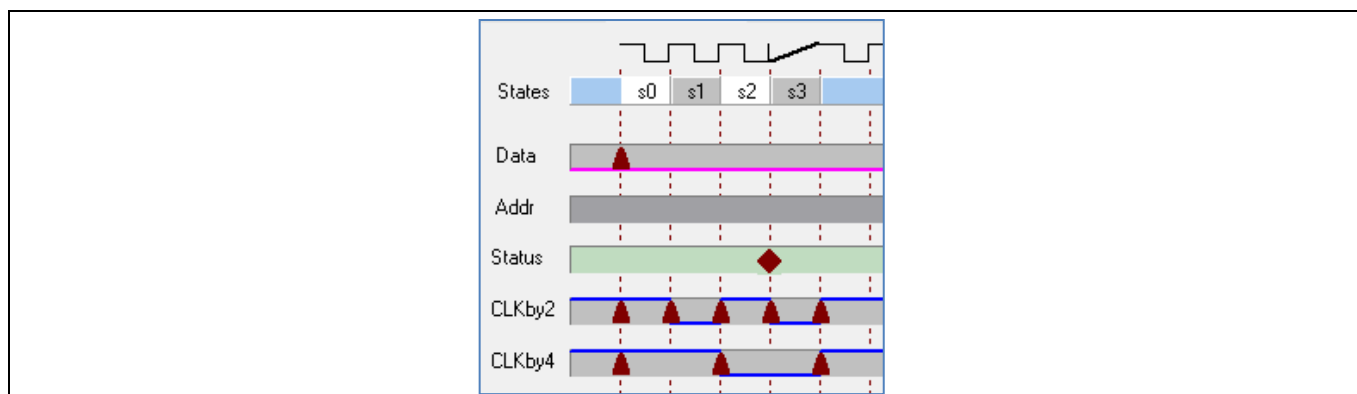


Figure 27 GPIF の波形エントリがステートを作成

Figure 27 には、GPIF Designer の波形エントリ画面を示します。この簡単な例では、内部 48 MHz クロックは 2 と 4 で分周され、名前が変更された CTL0 (CLKby2) と CTL1 (CLKby4) ピンに出力されます。縦の点線は遷移が発生するクロックエッジを示します。これらは立ち上がりエッジまたは立ち下がりエッジで設定できます。遷移を作り出すには、クロックエッジで所望の波形をクリックします。そうすると、波形がトグルされ、遷移点を示す赤い三角形が挿入されます。これらの三角形を削除したり、別のクロックエッジにドラッグしたりでき、マウスクリックで論理レベルを 1 か 0 に直接セットできます。

波形の変化を定義する際、GPIF Designer はトップのラインでステートを自動的に作成します (この場合は S0～S3 の 4 つのステート)。この波形はクロック出力を生成するために設計されているため、無限に繰り返します。これを実現するために、ステート 3 からステート 0 に分岐します。ステートへの分岐は、決定ステート (decision state) の始まりの点線をクリックして、ステータスラインに決定点を挿入することで行われます。GPIF Designer は決定ステートを示すダイヤモンド画像を挿入します。このダイヤモンド画像を右クリックすると、2 つの変数と 3 つの論理演算子 (AND、OR、XOR) で分岐条件用の論理方程式を作成するためのダイアログボックスが表示されます。無条件分岐は、同じステートを *if* 条件の確認と *else* 条件のデスティネーションに選択することで作成されます。

FX2LP 開発ツール

参考資料

GPIF の詳細情報については、[FX2LP テクニカル リファレンス マニュアル](#)の第 10 章を参照してください。

RAM と FX2LP の GPIF との通信に関する詳細情報については、[Interfacing SRAM with FX2LP over GPIF - AN57322](#) を参照してください。

Windows 側

6 Windows 側

Bulkloop デモを実行では、**USB Control Center** と **BULKLOOP_VCS** の Windows アプリケーションを使用して、Keil ツールで作成した bulkloop HEX ファイルをテストする方法を説明しました。本節では、FX2LP ベースのデバイスと通信する Windows アプリケーションを作成する用のサイプレスおよび他社のツールの概要を説明します。

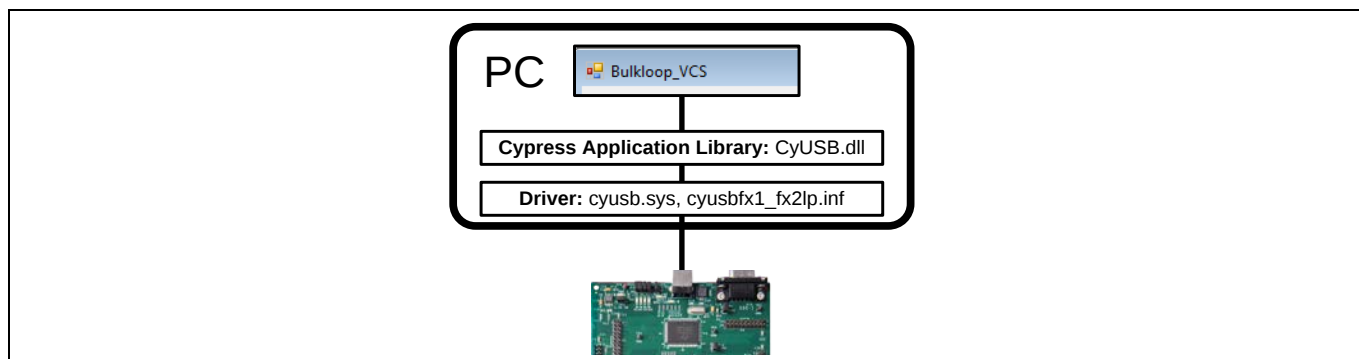


Figure 28 Windows .NET アプリケーションはサイプレスのドライバーとライブラリを使用して FX2LP と通信

6.1 サイプレスの USB ドライバー

サイプレス USB ドライバー (**CyUSB.sys**) はホストと USB デバイスとの通信に使用される強固な高性能 Windows ドライバーです。このドライバーは EZ-USB FX2LP DVK インストールの一部としてホスト PC 上にインストールされ、**Windows Driver** フォルダ内の zip 形式の添付ファイルに保存されています。ドライバーはバイナリ形式で提供され、FX2LP ベースのデバイスとともに配布することができます。

- **CyUSB.sys** は USB 2.0 標準に準拠したどのデバイスにも通信できるカーネル モードの USB 機能ドライバーです。このドライバーは汎用で、基本的な USB コマンドと互換性がありますが、USB デバイスクラス固有のより高レベルのコマンドを実装していません。
- ドライバーは、カスタム USB アプリケーションからベンダー固有のデバイスと通信するか、または、実験や診断アプリケーション用に低レベル USB 要求を任意の USB デバイスへ送信するための理想的な手段です。
- ドライバーがサポートするオペレーティング システム/プラットフォームは 32/64 ビットの Windows XP、Windows Vista および Windows 7 です。

例えば、ユーザーの会社の VID または PID、あるいはその両方を挿入する、またはカスタム記述テキスト文字列を挿入して、*.inf* ファイルを変更する際、Microsoft の「WHQL」(Windows Hardware Quality Labs) 認証を取得することをお勧めします。この Windows ロゴ認証を得ることにより、ドライバーをインストールする際に、警告メッセージが出ません。[付録 E: .inf ファイルへのカスタム VID と PID の追加](#)で、*inf* ファイルにカスタム VID/PID を追加する方法を説明します。

参考資料

CyUSB.sys の詳細については、**Cypress Suite USB\Driver** フォルダにある *CyUSB.pdf* ファイルを参照してください。(このフォルダはサイプレススイート USB のインストール後に表示されます)。

Windows 側

6.1.1 ドライバー／ライブラリの代替オプション

他の Windows ドライバーとライブラリが利用できます。Microsoft 製の WinUSB (詳細については [msdn](#) サイトを参照) は、カーネル モード ドライバー (**Winusb.sys**) と **WinUSB 関数** を含むユーザー モード ダイナミック リンク ライブラリ (**Winusb.dll**) で構成されています。これら高レベルの関数を使うと、ユーザーはユーザー モードのソフトウェアで USB デバイスを管理できます。**Winusb.sys** は Windows XP、Windows Vista、Windows 7 および Windows 8 でサポートされます。サイプレスのドライバーと同様に、インストールするには添付の *.inf* ファイルを必要とします。

Note: **WinUSB.sys** は **ISOCHRONOUS** 転送をサポートしませんが、サイプレスのドライバーはすべての USB 転送タイプをサポートします。

Note: Linux と Mac のユーザーは、**LIBUSB** というオープンソースのユーザー モード USB ルーチンの一式を調査できます。詳細については、[Libusb-1.0 API Reference](#) を参照してください。

Linux マシンでは、(FX3 SDK for Linux の一部として提供される) CyUSB Suite for Linux ソフトウェアは、FX2/FX3 デバイスファームウェアイメージをダウンロードし、デバイス上のさまざまなインターフェースをテストするのに使用できます。**FX3 SDK for Linux** の

fx3_sdk_v1.3_linux\cyusb_linux_1.0.4\cyusb_linux_1.0.4\docs というインストール フォルダにある *cyusb_linux_user_guide.pdf* を参照してください。この資料では、CyUSB Suite for Linux ソフトウェアをインストールする方法、ファームウェアを FX2/FX3 にダウンロードする方法、およびベンダー拡張機能、BULK OUT/IN 転送と ISOCHRONOUS OUT/IN 転送を試験する方法について説明します。CyUSB Suite for Linux - プログラマ リファレンスマニュアル (同じフォルダに保存される *cyusb_linux_programmers_guide.pdf* ファイル) は、Linux 用の cyusb ライブラリとライブラリでのユーザーがプログラムするアプリケーションをビルドおよび統合する方法を説明します。

FX2LP はヒューマン インターフェース デバイス (HID)、通信 (COMM) デバイスや大容量記憶 (MSC) デバイスなどの標準 Windows クラスに準拠するようにもプログラムできます。サイプレスのウェブサイトには、それらの共通クラスのサンプル FX2LP プログラムが搭載されています。Windows クラスに準拠する利点はデバイスが初めて挿入される時にドライバーのインストールが不要です。つまり、ドライバーは Windows の一部です。

Windows 側

6.2 サイプレス ライブラリ

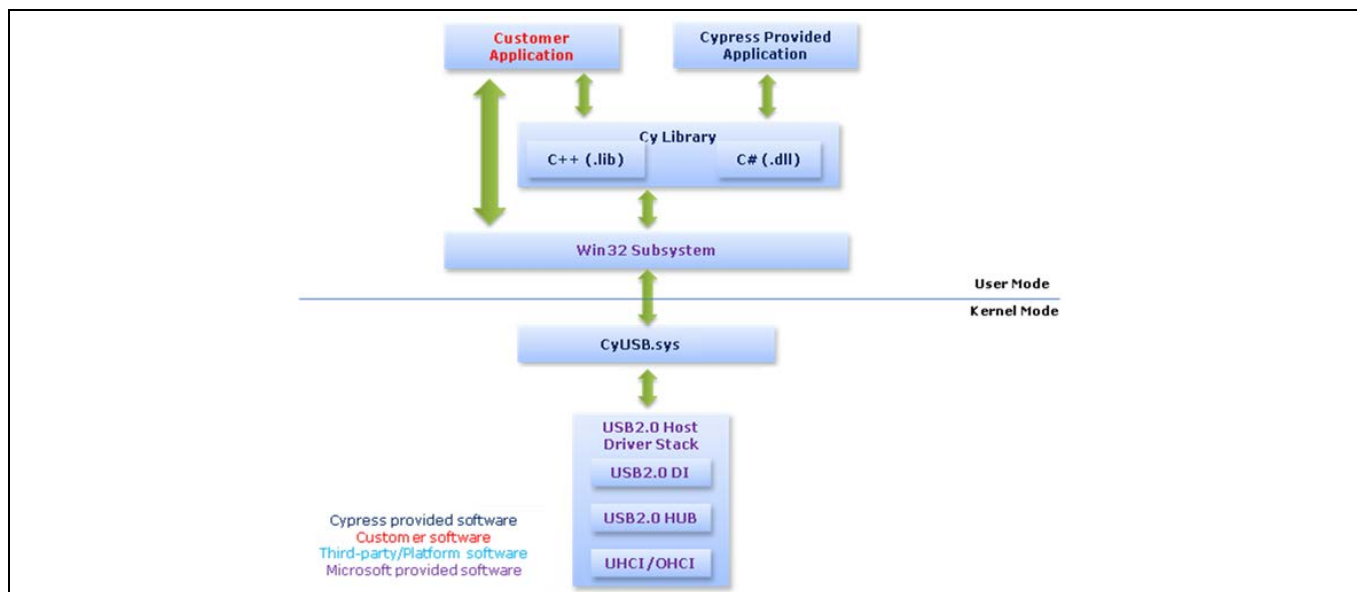


Figure 29 アプリケーションを備えたサイプレス ライブラリ通信の詳細図

Bulkloop_VCS アプリケーションはサイプレスの.NET ライブラリを使用した任意の Windows と FX2LP の通信に使用されるソフトウェア アーキテクチャの一例です。C#言語のプログラムは、サイプレスのドライバにアクセスするライブラリ呼び出しをして FX2LP と通信します (Figure 28 を参照)。**CyUSB.dll** ライブラリは Microsoft .NET 言語をサポートします。サイプレスは **CyAPI.lib** という C++クラス ライブラリも提供しています。両方のライブラリとも、**Cypress Suite USB** をインストールすると利用できます。

サイプレスの USB ファームウェアフレームワークがコード開発を FX2LP レベルで単純化すると同じように、これらのライブラリはコーディングを Windows レベルで簡略化します。(CyUSB.dll に基づいた) ソフトウェア モデルは以下の 3 つの要素を使用しています。

- **USBDeviceList** クラスは差し込まれている USB デバイスの一覧を作成します。コンストラクタは、HID や MSC 等のシングル デバイス クラスの一覧またはサイプレスドライバに対応されるデバイスの一覧をフィルターすることができます。
- デバイス一覧は、**CyUSBDevice** クラスのインスタンス一覧として表示されます。
- **CyUSBDevice** クラスは 1 つ以上の **CyUSBEndPoints** を提供し、それらを介してデータ転送が実行されます。

この高レベル モデルを使った USB デバイスとの通信は **DeviceIOControl** などの複数の Win32 API 呼び出しに比べると大きな改善です。

サイプレス ライブラリに慣れる最良の方法は Microsoft Visual C# Express edition を使って **Bulkloop_VCS** ソリューションを開いて、コードを調べることです。

参考資料

ホスト アプリケーション開発の詳細については、アプリケーション ノート [AN70983 - Designing a Bulk Transfer Host Application for EZ-USB® FX2LP™/FX3™](#)を参照してください。

.NET dll ライブラリの詳細については、**Cypress Suite USB\CyUSB.NET** フォルダ (サイプレスの Suite USB インストール後に表示される) にある **CyUSB.NET.pdf** を参照してください。**CyAPI.lib** の詳細情報については、**Cypress Suite USB\CyAPI** フォルダにある **CyAPI.pdf** をご覧ください。

まとめ

7 まとめ

サイプレスの FX2LP は USB 2.0 ハイスピード設計要件を満たすように設計されています。サイプレスはユーザーの設計サイクルの各ステップを支援するための幅広いサポート資料一式を提供します。このアプリケーションノートは最初に FX2LP を紹介し、次に BULK 転送例の設計、作成、テストの手順に従って実行することでサイプレスのハードウェア、ファームウェアおよびソフトウェアツールを紹介します。この例はユーザーのカスタム アプリケーションのベースとして使用できます。

著者について

氏名: Rama Sai Krishna

役職: シニア アプリケーション エンジニア

連絡先: rskv@cypress.com

付録 A: FX2LP 開発キット (DVK)

8 付録 A: FX2LP 開発キット (DVK)

CY3684EZ-USB FX2LP 開発キットは完全な開発リソースです。このキットはカスタムプロジェクトの開発とテスト用のプラットフォームを提供します。開発キットは設計のファームウェア、ハードウェアとソフトウェア面の関連資料を含んでいます。**Figure 30** は FX2LP DVK 開発基板のコンポーネントを示しています。

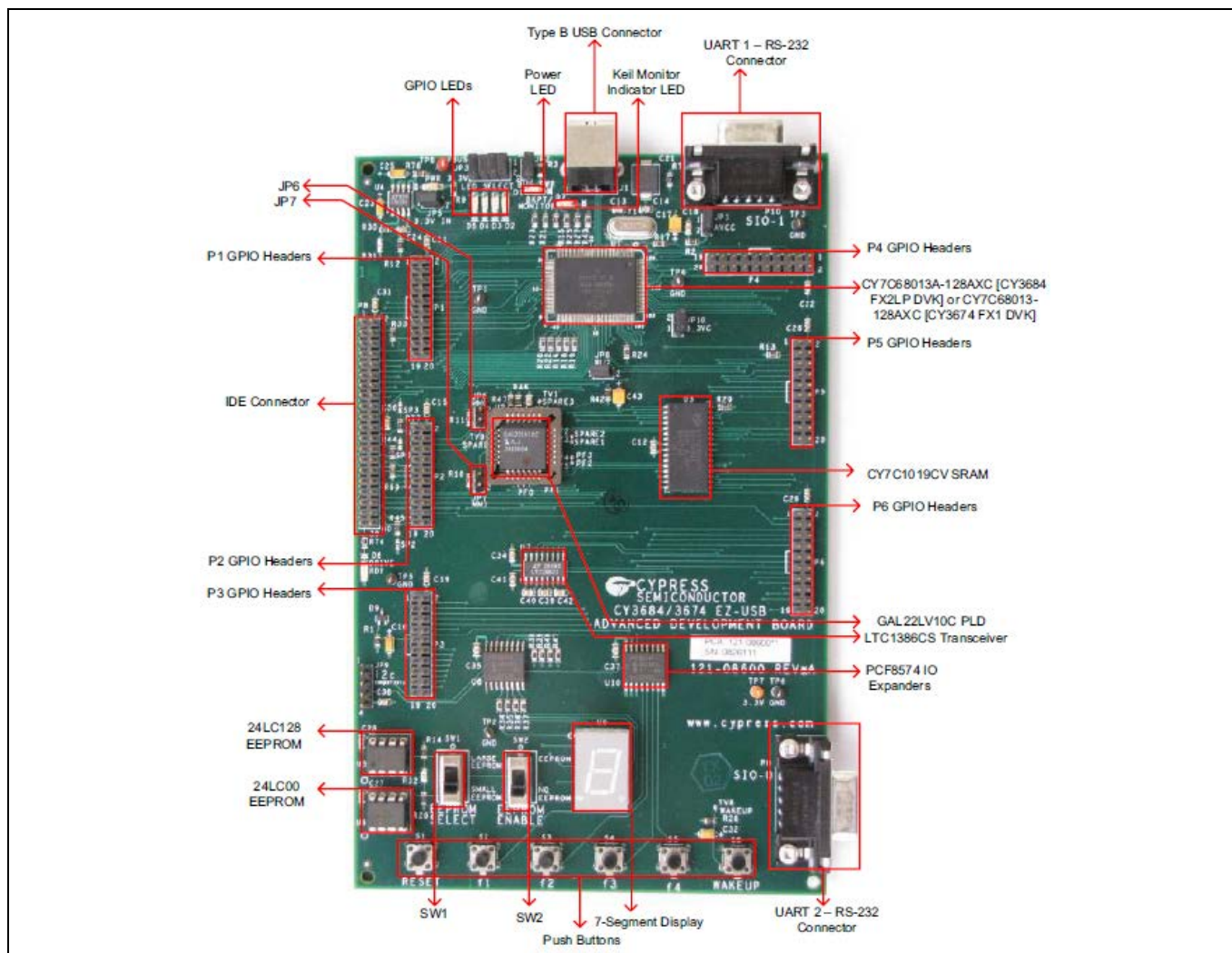


Figure 30 FX2LP DVK 開発基板のコンポーネント

FX2LP DVK は FX2LP 用の優れたデバッグプラットフォームです。(本アプリケーションノートで記述するように) 基板が USB からコードイメージをロードするか、またはコードを基板搭載 EEPROM にプログラムできます。

USB 2.0 ハイスピード デバイス用のハードウェアを開発する際に、基板のレイアウトと設計はプロジェクトの成功に極めて重要です。開発者が一般的なレイアウトエラーをしないために、サイプレスは新規の基板開発を支援するいくつかのリソースを提供しています。[AN1168](#) と [AN15456](#) アプリケーションノートは FX2LP の設計ガイドラインを提供します。また、FX2LP DVK インストールのハードウェアサブディレクトリは FX2LP 開発キットの回路図、開発基板用の BOM、および開発基板レイアウトと設計ファイルを含んでいます。

付録 A: FX2LP 開発キット (DVK)

8.1 ファームウェア サンプル プロジェクト

FX2LP ファームウェア開発で説明したように、FX2LP DVK は、ハイスピード デバイスに対応した USB 2.0 仕様の第 9 章に記載されている準拠の要件を満たす USB ファームウェア フレームワーク ライブラリを提供します。USB フレームワークは、FX2LP チップの初期化、USB 標準デバイスの要求処理や USB での一時停止と再開の電源管理などの一般的な動作にサイプレスのコードを使用してカスタム ファームウェア開発を簡略化して加速させます。USB ファームウェア フレームワークは関数「hook」とファームウェア例も提供します。それによって、ファームウェア開発プロセスは簡単になります。ユーザーは、USB ディスクリプタ テーブルとコードを書いて、低レベルの USB の細部を気にせずに希望の機能を実装できます。

CY3684EZ-USB FX2LP 開発キットのインストール後、ファームウェア ディレクトリは **Table 4** に示される例を含んでいます。これらの例はカスタム FX2LP ベースの製品用に参照または基礎として使用できます。

Table 4 FX2LP ファームウェア例の詳細

S.No	ファームウェア例	説明
1	hid_kb	DVK 基板上のボタンと 7 セグメント ディスプレイを使用して HID クラス キーボードをエミュレート。
2	Bulkloop	EZ-USB バルク エンドポイントを演習するバルク ループバック テスト。これは EP2OUT を EP6IN に、EP4OUT を EP8IN にループバックし、開発基板のインジケータを更新。
3	Bulkext	外部 RAM を使用するバルク ループバック テスト。データが OUT エンドポイント バッファから DVK 基板上の外部 RAM にコピーされ、そして IN エンドポイント バッファに戻される。これは EP2OUT を EP6IN に、EP4OUT を EP8IN にループバック。
4	Bulksrc	テストに対応したバルク データの制限の無いプロバイダとコンシューマ。これは CyConsole または CyBulk を使って駆動。EP2OUT と EP4OUT はすべてのバルク パケットを受け入れる。EP6IN はハイスピードで動作する時、常に 512 バイトのパケットを返し、フルスピードで動作する時、常に 64 バイトのパケットを返す。EP8IN のバッファの使用可能性に基づいて、EP4OUT の最新のパケットが EP8IN に書き込まれる。
5	dev_io	簡単な開発基板 I/O サンプルを構築するためのソース ファイル。このソフトウェアは EZ-USB 開発キット上でボタンと LED をどのように使用するかを示す。
6	EP_Interrupts	エンドポイント割り込みを使用するバルク ループバックのファームウェア。
7	extr_intr	INT0、INT1、INT4、INT5 および INT6 を使用する外部割り込み処理。
8	ibn	IBN (In Bulk NAK) 割り込みを使って転送を開始する、EP2OUT から EP6IN への、EP4OUT を EP8IN へのバルク ループバック。
9	LEDCycle	DVK 上の汎用インジケータ LED (D2、D3、D4、D5) の使用。
10	Pingnak	PING NAK 割り込みを使って転送を開始する、EP2OUT から EP6IN へ、および EP4OUT から EP8IN へのバルク ループバック。
11	iMemtest	内蔵 RAM をテスト。
12	vend_ax	ベンダー固有のコマンドを実装する方法を示す。

USB ファームウェア フレームワークは多くのファームウェア プロジェクトに共通している関数を実装する EZ-USB ライブラリ (EZUSB.LIB) を使用します。これらの関数は変更する必要がありません。したがって、ライブラリの形で提供されます。ただし、キットはライブラリ用のソースコードを含んでいます。

付録 A: FX2LP 開発キット (DVK)

す。それによって、ユーザーはある関数を変更したいか、または何かがどのように実行されるかを知りたい時、このソースコードを使用できます。EZ-USB ライブラリ (5.4 節) とファームウェアフレームワーク (第 5 章) の詳細な情報は [CY3684 DVK Kit_Guide](#) に記述されています。このライブラリは

「Target\Lib\LP」フォルダの下にある DVK に含まれています。顧客がサイプレスの広い範囲のハイスピード製品を使用して高速に自分のアプリケーションを設計できるように、すべての [USB ハイスピード サンプルコード](#) の包括的な一覧を提供しています。他の USB ハイスピード デバイス、キット、SDK、アプリケーション ノートおよびリファレンス デザインの詳細は、以下に記載します。

Note: これらのソフトウェアリソースは継続して改善されています。そのため、サイプレスは [サイプレスのウェブサイト](#) から最新のソフトウェアをダウンロードすることを推奨します。

付録 B: サイプレスからの追加 USB ハイスピード デバイス

9 付録 B: サイプレスからの追加 USB ハイスピード デバイス

サイプレスの USB ハイスピード ペリフェラル ファミリは以下の追加デバイスを含んでいます。

- **AT2LP:** サイプレスの EZ-USB® AT2LP™ (CY7C68300C/301C/320C) は 1 個の USB ポートと 1 個か 2 個の ATA/ATAPI ベースの大容量ストレージ デバイス ポート間の固定機能ブリッジを実装しています。AT2LP 上の PATA インターフェースはユーザーの設計にハード ディスク ドライブ (HDD)、コンパクト フラッシュおよびソリッド ステート ドライブ (SSD) の使用を有効にします。AT2LP は大容量ストレージ タイプのアプリケーションに最適であり、カスタム ファームウェア無しで商品の市場投入までの時間を短縮します。AT2LP はすべての ATA/ATAPI-6 準拠の大容量ストレージ デバイスをサポートします。
- **NX2LP-Flex:** サイプレスの EZ-USB NX2LP-Flex™ (CY7C68033/34) は低消費電力プログラム可能な USB-SLC NAND コントローラーです。このコントローラーはプログラム可能であるため、設計者は NAND デバイスのサポートにより、特別な機能を含むようにコントローラーをプログラムできます。これは固定コントローラーより優れている利点です。NX2LP-Flex にあるハードウェア ECC エンジン は 1 ビットのエラー訂正と 2 ビットのエラー検出をサポートします。
- **SX2:** The EZ-USB SX2 (**CY7C68001**) は標準マイクロプロセッサ、DSP、ASIC や FPGA などの任意の外部 マスターと連動して、任意のペリフェラル設計に対する USB 2.0 サポートを可能にするように設計されているプログラム可能なデバイスです。SX2 は USB データを送受信するための内蔵 USB トランシーバ、シリアル インターフェース エンジン (SIE) および コマンド デコーダを備えています。このコントローラーは最大の柔軟性とスループットのために 1 つの 4KB FIFO 空間を共有する 4 つのエンドポイントを持っています。SX2 はコマンドとデータ入出力用の 3 つのアドレス ピンと選択可能な 8 ビット/16 ビットのデータバスを備えています。
- **FX2LP18:** MoBL-USB FX2LP18 (CY7C68053) は 1.8V で動作し、低消費電力携帯機器での使用に適しています。AN6076 は FX2LP と FX2LP18 の相違点をリストアップします。

付録 C: サードパーティの開発キットと SDK

10 付録 C: サードパーティの開発キットと SDK

1. Opal Kelly 社からの FPGA + FX2LP 基板:

この基板の詳細な情報は以下のロケーションにあります。

<http://www.opalkelly.com/products/xem6010/>

特長:

- ダウンロードと制御用のハイスピード USB 2.0 インターフェース (サイプレス FX2LP - CY68013A)
- Xilinx Spartan-6 (XC6SLX45-2FGG または XC6SLX150-2FGG)
- 32 Mi ビットのシリアルフラッシュ (Numonyx M32P25)
- 128 Mi バイトの DDR2 (Micron MT47H64M16HR)
- 小型フォームファクタ – クレジットカードより小さい (75 mm x 50 mm x 15.9 mm (2.95" x 1.97" x 0.63"))
- 外部 DC ソースから自己電源供給
- マルチ PLL、マルチ出力クロックジェネレータ (サイプレス CY22393)。

2. ZTEX 社からの FPGA + FX2LP 基板:

これらの基板の詳細な情報は以下のロケーションにあります。

<http://www.ztex.de/usb-fpga-1/>

ZTEX 社からこのような基板の特長:

- サイプレス CY7C68013A EZ-USB FX2LP マイクロコントローラー
- ハイスピード (480Mbps) USB インターフェース
- Xilinx Spartan 3 XC3S400 FPGA
- 60 の汎用 I/O (GPIO)
- 20 の特殊 I/O (SIO)
- 128 kb の EEPROM (例えば、ファームウェア向け)
- フラッシュメモリ (任意)

10.1 サードパーティの SDK

ZTEX 社は FX2LP ベースの基板と連動する SDK を提供します。また、ホストソフトウェアの開発を支援する JAVA ベースの API も提供します。詳細については、<http://www.ztex.de/firmware-kit/> をご覧ください。

11 付録 D: アプリケーションノートとリファレンス デザイン

11.1 アプリケーションノート

- **AN15456 - Guide to Successful EZ-USB® FX2LP™ and EZ-USB FX1™ Hardware Design and Debug**

このアプリケーションノートでは、特にハイスピードで動作している時、起こり得る USB ハードウェア設計の問題を特定します。また、基板を造る前に容易に潜在的な問題を見つけ出せるようにし、基板の取得と実行時にデバッグを支援します。

- **AN50963 - EZ-USB® FX1™/FX2LP™ Boot Options**

このアプリケーションノートは、ファームウェアを FX1/FX2LP にダウンロードする様々な方法を記述しています。

- **AN66806 - EZ-USB® FX2LP™ GPIF Design Guide**

このアプリケーションノートは、GPIF Designer を使って GPIF 波形を開発するのに必要な手順を説明しています。

- **AN61345 - Implementing an FX2LP™- FPGA Interface**

このアプリケーションノートは、FX2LP を FPGA にインターフェースするサンプルプロジェクトを提供します。インターフェースでは、データ収集、産業用の制御と監視、画像処理などの FPGA ベースのアプリケーション用に高速 USB 接続を実装しています。FX2LP はスレーブ FIFO モードで動作し、FPGA はマスターとして動作します。このアプリケーションノートは、スレーブ FIFO 実装用に FX2LP サンプルファームウェアおよび FPGA 実装用に VHDL と Verilog サンプルプロジェクトも用意しています。

- **AN57322 - Interfacing SRAM with FX2LP over GPIF**

このアプリケーションノートは汎用プログラマブルインターフェース (GPIF) を使用してサイプレスの CY7C1399B SRAM を FX2LP に接続する方法について説明します。これは GPIF Designer を使って読み出し波形と書き込み波形を作成する方法を記述します。アプリケーションノートは FX2LP を他の SRAM に接続するための参考資料としても有用です。

- **AN58009 - Serial (UART) Port Debugging of FX1/FX2LP Firmware**

このアプリケーションノートはシリアルデバッグ処理用の FX2LP ファームウェアに必要なコードを記述しています。このコードにより、開発者は PC 端末プログラム内にデバッグメッセージと変数のリアルタイム値を印字するか、FX2LP の UART エンジンを使ってファイル内のデータを取り込むことができます。

- **AN42499 - Setting Up, Using, and Troubleshooting the Keil Debugger Environment**

このアプリケーションノートは、Keil 社のデバッガを使用する初心者向けの段階的なガイドです。このガイドは PC から SIO-1/0 へのシリアルケーブル接続、モニタコードのダウンロードおよび必要なプロジェクト設定を取り扱っています。また、デバッグセッションの開始と停止、ブレークポイントの設定、コード実行のステップスルー、潜在的な問題解決のためのガイドラインを提供します。

- **AN4053 - Streaming Data through Isochronous/Bulk Endpoints on EZ-USB® FX2 and EZUSB FX2LP**

このアプリケーションノートは EZ-USB FX2 または EZ-USB FX2LP を使ったストリーミングアプリケーションの基礎的な情報を提供します。高帯域幅アプリケーションで FX2/FX2LP を使用する際に考慮すべき設計問題をはじめ、バルクエンドポイント、アイソクロナスエンドポイント、および高帯域幅のアイソクロナスエンドポイントを介したデータのストリーミングに関する情報を提供します。

付録 D: アプリケーション ノートとリファレンス デザイン

- **AN58069 - Implementing an 8-Bit Parallel MPEG2-TS Interface Using Slave FIFO Mode in FX2LP**

このアプリケーション ノートはスレーブ FIFO モードを使用して 8 ビットのパラレル MPEG2-TS インターフェースを実装する方法を説明します。サンプル コードはデータ ストリーム用に、EZ-USB FX2LP をレシーバとして、データ ジェネレータをソースとして使用します。ハードウェア接続とサンプル コードが含まれています。

- **AN58170 - Code/Memory Banking Using EZ-USB**

EZ-USBFX2 のチップ ファミリは 8051 コアを搭載しています。8051 コアは 16 ビットのアドレス ラインを備えており、メモリの 64 KB にアクセスできます。ただし、いくつかのアプリケーションは 64 KB を超えるメモリ容量を必要とします。このアプリケーション ノートはこの 64 KB 境界を超えてアクセスする方法を記述します。

- **AN1193 - Using Timer Interrupt in Cypress EZ-USB FX2LP Based Applications**

このアプリケーション ノートは EZ-USBR FX2LP ファームウェア開発者がアプリケーションにタイマー割り込みを使用するのを助けます。

- **AN63787 - EZ-USB® FX2LP™ GPIF and Slave FIFO Configuration Examples using 8-bit Asynchronous Interface**

このアプリケーション ノートは 8 ビットの非同期パラレル インターフェースを実装するために、EZ-USB FX2LP で汎用プログラマブル インターフェース (GPIF) とスレーブ FIFO を手動モードと自動モードの両方で設定する方法を説明します。このアプリケーション ノートは連結して接続された 2 つの FX2LP 開発キットでテストされます。1 つはマスター モードで動作し、もう 1 つはスレーブ モードで動作します。

- **AN61244 - Firmware Optimization in EZ-USB**

このアプリケーション ノートは EZ-USB でのファームウェア最適化の方法を記載しています。これらの方法の一部はどのプロセッサにも共通していますが、その他は EZ-USB FX2LP の 8051 コアに固有のものであります。

- **AN74505 - EZ USB FX2LP - Developing USB Application on MAC OS X using LIBUSB**

このアプリケーション ノートは libusb を使用する MAC OS プラットフォームで構築されたホスト アプリケーションについて説明しています。ホスト アプリケーション (Cocoa アプリケーション) は libusb の API によって提供されたインターフェースを介して FX2LP の BULK IN と BULK OUT エンドポイントと通信します。このホスト アプリケーションは特定の VID/PID (=0x04B4/0x1004) 識別を送信するデバイスとの通信を実行します。

- **AN58764 - Implementing a Virtual COM Port in FX2LP**

このアプリケーション ノートは FX2LP で標準 Windows ドライバーを使用して仮想 COM ポート デバイスを実装する方法を説明します。この情報は UART から USB に移行するのに役立ちます。

- **AN45471 - Vendor Command Design Guide for FX2LP**

このアプリケーション ノートは特定の製品を実行するために USB ベンダー コマンドをコーディングする方法を実演します。また、このノートはベンダー コマンドを発行するために、サイプレスの CyConsole ユーティリティを使用する方法を説明しています。

付録 D: アプリケーション ノートとリファレンス デザイン

リファレンス デザイン

よく利用されているアプリケーション用の FX2LP のいくつかの参考設計が用意されています。参考設計は設計用のデモのソースコード、参考回路図と BOM (必要に応じて) を含んでいます。

以下はサイプレスのウェブサイトにある参考設計です。

- **CY4661 - External USB Hard Disk Drives (HDD) with Fingerprint Authentication Security**

サイプレスと UPEK 社からの CY4661 リファレンス デザイン キットはデータの保護と認証用に指紋認証とセキュリティを備えている外部 USB ハード ディスク ドライブ (HDD) 用のターンキー ソリューションを提供します。参考設計は UPEK 社のタッチストリップ指紋認証ソリューション (TCS3 スワイプ指紋センサーと TCD42 セキュリティ ASIC) を使用します。

- **FX2LP DMB-T/H TV Dongle reference design**

このリファレンス デザイン キットはサイプレスの FX2LP と Legend Silicon 社のチップセットに基づいたものです。Legend Silicon 社のチップセットによって MPEG2 TS ストリームに変換された取り込みと復調済みの RF 信号は FX2LP を介して PC に送信されます。PC はメディア プレーヤーを使ってこれらのストリームを再生します。これは必要なファイルをすべて含んでいる完全な設計です。

付録 E: .inf ファイルへのカスタム VID と PID の追加

12 付録 E: .inf ファイルへのカスタム VID と PID の追加

Bulkloop ファームウェア例は VID 0x4B4 と PID 0x1004 を使用します。これらの値はこのアプリケーションノートに添付された **cyusbf1_fx2lp.inf** ファイルに含まれています。FX2LP ファームウェアプロジェクトでは VID と PID 値を .inf ファイルに未だリストアップされていない値に変更したい場合、ユーザーのデバイスが USB Control Center などのサイプレス開発ツールによって認識されるようにこれらの値を追加する必要があります。カスタム VID と PID 値を **cyusbf1_fx2lp.inf** に追加するには、以下の手順に従ってください。

cyusbf1_fx2lp.inf ファイルを開きます。これはテキストファイルなので、WordPad などの任意のテキストエディタを使用できます。**Figure 31** と **Figure 32** に示すようにカスタム VID と PID を追加します。これらの図に示される手順では、カスタム VID が 0x4B4 で、カスタム PID が 0x1005 であると仮定しています。これらの手順が Windows XP (32 ビットのプラットフォーム) 用に表示されていることに注意してください。

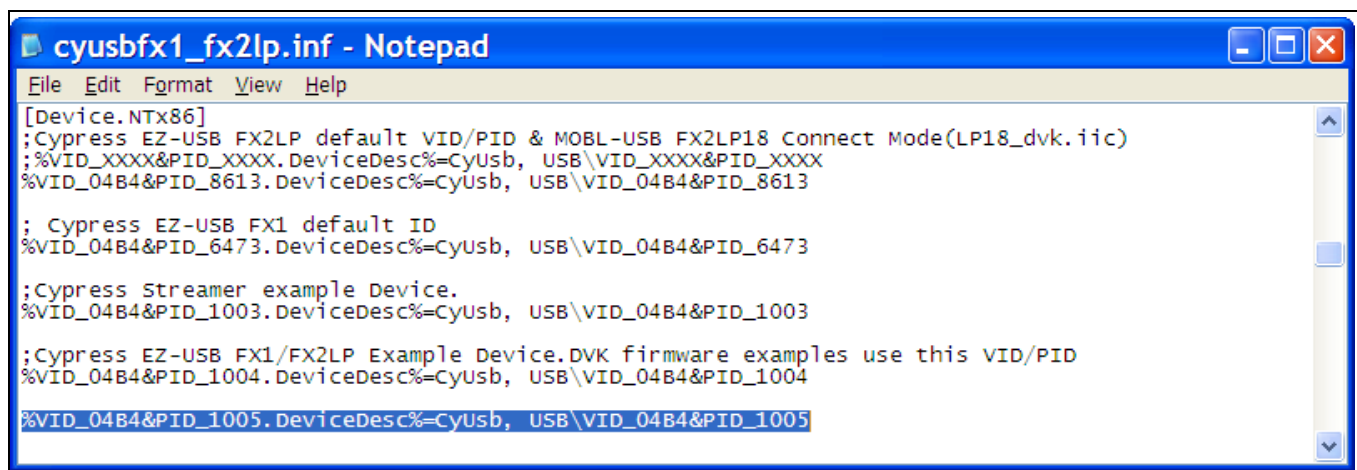


Figure 31 カスタムの VID と PID を Cyusb.inf に追加

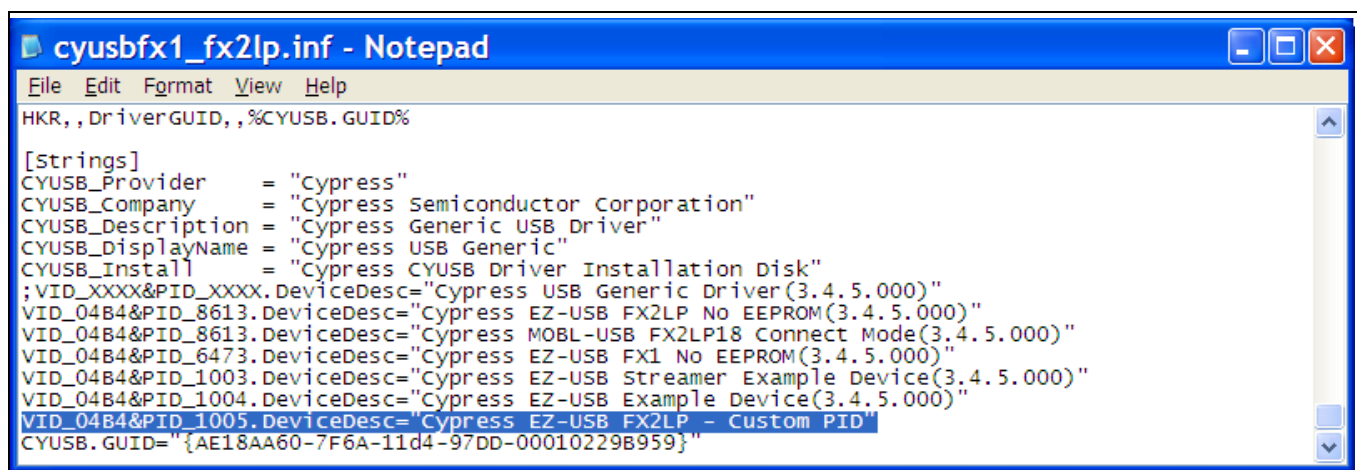


Figure 32 カスタムの VID と PID を Cyusb.inf に追加

これらの変更を行った後、FX2LP 開発基板を PC に接続し、VID 0x4B4 と PID 0x1005 を持つユーザーのファームウェアイメージ(.hex ファイル)をダウンロードします。ドライバーインストーラがドライバーの場所を問い合わせる時、ユーザーは変更された **cyusbf1_fx2lp.inf** ファイルを指して **CyUSB.sys** に結び付けます。FX2LP は **CyUSB.sys** に結び付けられた後、**Figure 33** に示すように、Control Center の左パネ

付録 E: .inf ファイルへのカスタム VID と PID の追加

ルに表示されます。ここで、Control Center でカスタムの VID と PID を使ってデータの転送を実行できます。

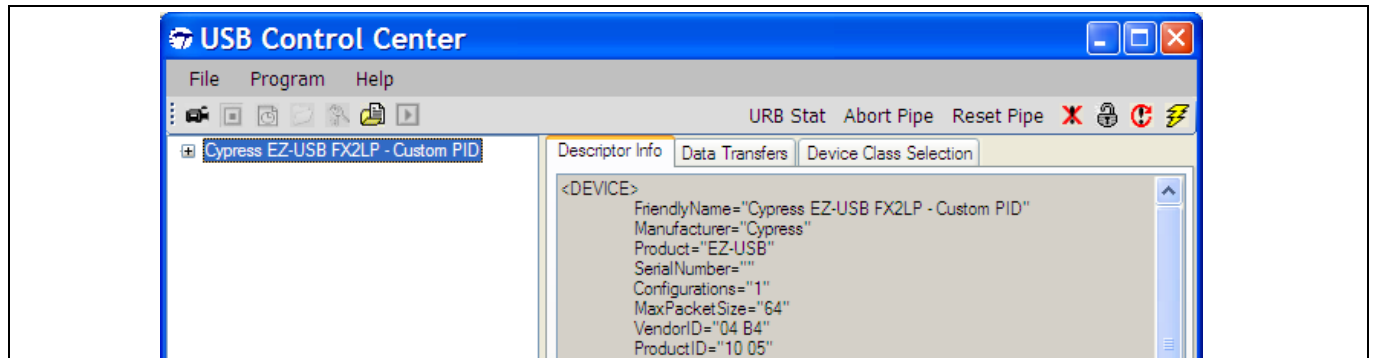


Figure 33 USB Control Center にリストアップされるカスタム FX2LP デバイス

改訂履歴

改訂履歴

Document version	Date of release	Description of changes
**	2015-03-19	これは英語版 001-65209 Rev. *B を翻訳した日本語版 001-86527 Rev. **です。
*A	2015-07-30	これは英語版 001-65209 Rev. *E を翻訳した日本語版 001-86527 Rev. *A です。
*B	2016-04-18	日本語版 001-86527 Rev. *A から変更はありません。
*C	2017-07-04	更新されたロゴと著作権。
*D	2019-05-09	これは英語版 001-65209 Rev. *H を翻訳した日本語版 001-86527 Rev. *D です。
*E	2021-04-26	テンプレートの変更を実施。 これは英語版 001-65209 Rev. *I を翻訳した日本語版 Rev. *E です。

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2021-04-26

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2021 Infineon Technologies AG.

All Rights Reserved.

Do you have a question about this document?

Go to www.cypress.com/support

Document reference

001-86527 Rev. *E

重要事項

本文書に記載された情報は、いかなる場合も、条件または特性の保証とみなされるものではありません（「品質の保証」）。本文に記された一切の事例、手引き、もしくは一般的価値、および／または本製品の用途に関する一切の情報に関し、インフィニオンテクノロジーズ（以下、「インフィニオン」）はここに、第三者の知的所有権の不侵害の保証を含むがこれに限らず、あらゆる種類の一切の保証および責任を否定いたします。

さらに、本文書に記載された一切の情報は、お客様の用途におけるお客様の製品およびインフィニオン製品の一切の使用に関し、本文書に記載された義務ならびに一切の関連する法的要件、規範、および基準をお客様が遵守することを条件としています。

本文書に含まれるデータは、技術的訓練を受けた従業員のみを対象としています。本製品の対象用途への適合性、およびこれら用途に関連して本文書に記載された製品情報の完全性についての評価は、お客様の技術部門の責任にて実施してください。

本製品、技術、納品条件、および価格についての詳しい情報は、インフィニオンの最寄りの営業所までお問い合わせください (www.infineon.com)。

警告事項

技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。

インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないこと予めご了承ください。