

EZ-USB® FX2LP™/FX3™ のバルク転送ホスト アプリケーションの設計

著者: Gayathri Vasudevan

関連製品ファミリ: CY7C6801XA, CYUSB3014

関連するリソース: すべての一覧を表示するには、[ここをクリック](#)してください。最新のアプリケーション ノート、または関連プロジェクトファイルを入手するには、[以下をご参照](#)ください。<http://www.cypress.com/AN70983>

サイプレスは、ホスト PC 向けのソフトウェア開発ツールを Visual Studio .NET ライブラリの形式で提供しています。このライブラリは Windows レベルで USB コーディングを簡素化します。本アプリケーション ノートは .NET クラス ライブラリを紹介し、FX2LP や FX3 開発キット (DVK) 上で実行されるサンプル「バルクループ」ファームウェアを使用してデータを送信および取得する Windows の例を作成する方法を示します。

目次

1. はじめに	1	6.2. FX3 Bulkloop ファームウェア	14
2. 機能の概要	2	7. トラブルシューティング	15
3. アプリケーション ノート フォルダのレイアウト	3	8. システム要件	15
3.1. 開発基板	3	8.1. ハードウェア	15
3.2. USB Control Center を用いてファームウェアを開 発基板にダウンロード	4	8.1.1. FX2LP	15
4. アプリケーションの概要	6	8.1.2. FX3	15
4.1. 特長	6	8.2. 開発ソフトウェア	15
4.2. 動作説明	7	9. 関連するリソース	16
5. サイプレスの .NET ライブラリの使用	8	10. まとめ	16
5.1. CyUSB.dll にリファレンスを追加	8	11. 用語集	17
5.2. ソース コード概要	10	A 付録: FX2LP/FX3 DVK ドライバーの Windows へのイン ストール	18
5.3. 非同期通信	11	改訂履歴	21
6. ファームウェア	12	ワールドワイド販売と設計サポート	22
6.1. FX2LP Bulkloop ファームウェア	12		

1. はじめに

本アプリケーション ノートは、Microsoft .NET 言語向けのサイプレス ライブラリを使用してサイプレスの FX2LP や FX3 デバイスと通信するためのホスト PC アプリケーションを実装する方法を示します。このライブラリを使用すると、Visual C#, Visual Basic または Visual C++ プログラムは、より高いレベルで FX2LP や FX3 ベースのデバイスと通信することができます。たとえば、USBDeviceList および CyUSBDevice クラス（「[ソース コードの概要](#)」節で説明）では、デバイスおよびエンドポイントはシンプルなクラス メンバーとして認識されます。この高レベル モデルを使った USB デバイスとの通信は、複数の低レベル Win32 API 呼び出しに比べると大きな改善です。たとえば、完全な USB デバイス ツリー ディスプレイを 15 行の C# コードだけでプログラムすることができます。

このアプリケーションノートでは Visual Studio ソリューションは Visual C#, Visual Basic および Visual C++ 言語でサンプル プロジェクトを提供しています。ホスト プロジェクトは Microsoft Visual Studio Express 2012 を用いてビルドおよびテストされます。ホスト アプリケーションは FX2LP または FX3 の BULK IN と BULK OUT エンドポイントに書き込みます。ライブラリ クラスは、サイプレスが提供している USB ドライバー CyUsb3.sys と通信してこれらのデバイスとエンドポイントにアクセスします。

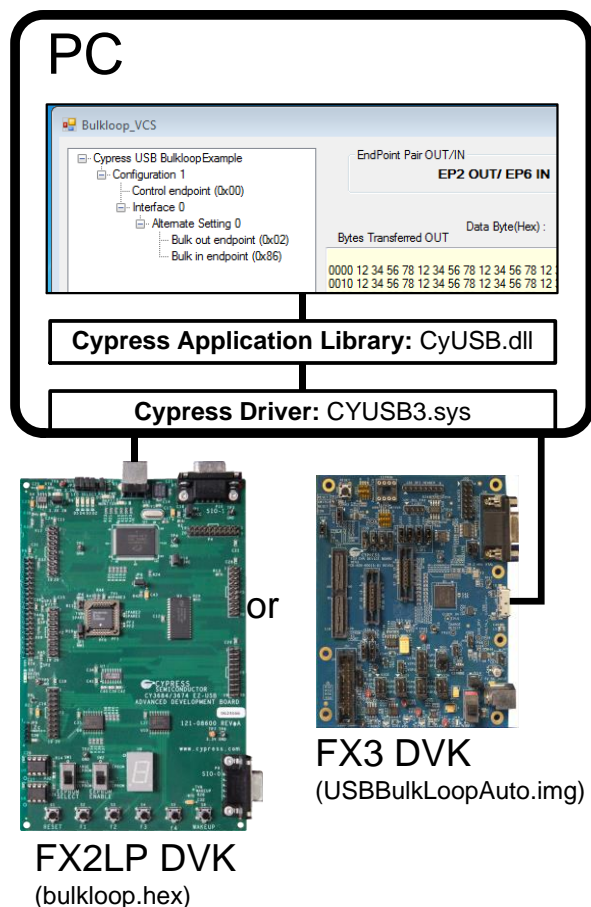
Windows Visual Studio コードを練習するために、このアプリケーションノートでは FX2LP および FX3 評価基板用に「バルクループ」というサンプル ファームウェアが提供されています。このファームウェアはホスト PC データを受け入れて、命令で同じデータを PC に送り返します。

このアプリケーション ノートでは USB 接続の Windows 側とデバイス側の概要を説明します。また、本アプリケーション ノートに関連するサンプル コードのソフトウェア レイアウトを説明し、FX2LP および FX3 評価基板をテスト ターゲットとして準備する方法を詳しく説明

します。さらに、コンパイル済みの Windows コードを実行する方法を示します。これにより、Visual Studio を使用せずに Windows アプリケーションを実行することが可能です。

2. 機能の概要

図 1. システム ブロック図



注: 図 1 の FX3 DVK は一例です。CYUSB3KIT-001 /CYUSB3KIT-003 を使用することができます。

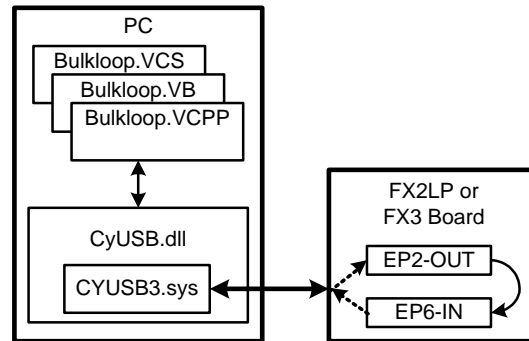
図 1 では、ホスト PC は USB ケーブルで 2 個のサイプレス ターゲット基板のどちらか 1 個に接続します。FX2LP または FX3 評価基板は、バルク データをエンドポイント 2-OUT で受け入れ、PC データ取得のためにエンドポイント 6-IN バッファにコピーします。ファームウェア ファイル名は図 1 の各基板の下に示されています。

注: USB の方向は、PC を中心としているため、OUT は PC からデバイスへ、IN はデバイスからホストへの方向です。

ホスト アプリケーションは CyUSB.dll ライブラリへの呼び出しを行って EP2-OUT にデータを送信し、その結果、サイプレスの USB ドライバー CYUSB3.sys と通信します。ホストが EP6 に IN 要求を発行すると、バルク データは CyUsb3.sys および CyUsb.dll ライブラリ呼び出しを介して送り返され、最後に表示されるためにホスト アプリケーションに到達します。送受信されたバルク データは転送されたバイト数と共に PC テキスト ボックスに表示されます。このデータフローは、図 2 に描かれています。

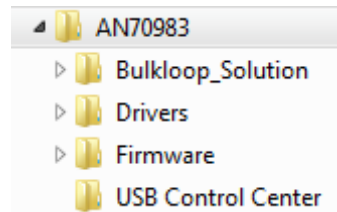
Microsoft Visual Studio ソリューションは Visual C# (図 1 に Bulkloop_VCS として示される)、Visual Basic および Visual C++ のプロジェクトをサポートしています。これらのプロジェクトは同じ結果を生成します。

図 2. Bulkloop データ フロー



3. アプリケーション ノート フォルダのレイアウト

図 3. アプリケーション ノート フォルダのレイアウト



サイプレスは、ファームウェア開発向けのソフトウェア ツール一式、および FX2LP および FX3 デバイスとの通信の .NET ベースのアプリケーションを提供しています。便宜上、本アプリケーション ノートで使用するファイルは付属の zip ファイルで提供されています。完全なツール一式をダウンロードするために、[サイプレス ウェブサイト](#)へアクセスしてください。

図 3 はアプリケーション ノート フォルダのレイアウトを示します。

- **Bulkloop_Solution:** これは Microsoft .NET ソリューションであり、Visual Basic, Visual C#, Visual C++ の 3 つの言語でバルク ループ テスターを実装します。
- **Drivers:** 初めて FX2LP や FX3 開発基板を差し込む時、Windows は適切なドライバーの場所を問い合わせます。このフォルダ は Windows XP (「w xp」), Windows Vista (「w l h」) および Windows 7 (「w i n 7」) のドライバーを格納しています。Vista と Windows 7 ドライバーは 32 ビット システム (「x 8 6」) および 64 ビット システム (「x 6 4」) 向けに用意されています。ドライバーは cyusb3.sys と名付けられ、FX3 と FX2LP 両方に対応しています。
- **注:** ezusb.sys と cyusb.sys は、FX2LP ファミリーデバイス用にサイプレスによって配布されたレガシードライバーです。新しいデザイン向けには ezusb.sys または cyusb.sys をお勧めしません。cyusb3.sys への移行の詳細については、[CyUSB.sys から CyUSB3.sys への EZ-USB FX2LP™ のドライバーの移行](#)を参照してください。
- **Firmware:** このフォルダは FX2LP および FX3 基板のバルク ループバック ファームウェアを格納しています。完全な開発スイートをダウンロードした後、コードの研究、変更、コンパイルを行えるようにソース コードが含まれています。
- **USB Control Center:** [EZ-USB FX3 ソフトウェア開発キット](#) (SDK) の Visual C#アプリケーションは、FX2LP および FX3 ファームウェアをそれぞれの対応する開発基板にダウンロードします。

3.1. 開発基板

FX2LP 開発基板は [FX2LP 開発キット](#)に同梱されています。FX3 基板は [FX3 開発キット](#)(CYUSB3KIT-001 または CYUSB3KIT-003)に同梱されています。Windows コードを実行する前に、図 2 に示した右側のループバック機能を実行するために開発基板を用意する必要があります。本アプリケーション ノート付属の.zip ファイルには 2 つのファームウェア ディレクトリがあり、1 つはサイプレス FX2LP 用 (Bulkloop_FX2LP)、もう 1 つはサイプレス FX3 用 (Bulkloop_FX3) です。

ループバック ファームウェアをロードする前に、FX2LP 基板のジャンパとスイッチが表 1 の通りに、FX3 のジャンパとスイッチが表 2/表 3 の通りにセットされていることを確認してください。

表 1. FX2LP 評価基板の設定

ジャンパ/スイッチ	用途	設定
JP1, JP10	EZ-USB チップの 3.3V 電源	IN, IN
JP2	Vbus から電源供給	IN
JP3	4 個の LED をアクティブにする	IN(x4)
JP5	3.3V 電流モニター	IN
JP6, JP7	FX2LP メモリ マップ (内部 + 外部 RAM)	OUT, OUT
JP8	二次ウェイクアップ	x
SW1	小さい/大きい EEPROM	x
SW2	EEPROM イネーブル	OFF (EEPROM なし)

表 2. FX3 評価基板 (CYUSB3KIT-001) の設定

ジャンパ/スイッチ	用途	設定
PMODE スイッチ	USB ダウンロードを選択	1, 2: OFF(下の方); 3, 4: x
J96, J97	USB ダウンロードを選択	上のピン 2 本 (2 と 3)
J98	USB ダウンロードを選択	未接続
J53	Vbus から電源供給	IN

表 3. FX3 評価基板 (CYUSB3KIT-003) の設定

ジャンパ/スイッチ	用途	設定
J3	開発基盤は USB3.0 VBUS から電源供給	ショート
J4	USB ダウンロードを選択	ショート
J5	外部 SRAM を選択しない	オープン

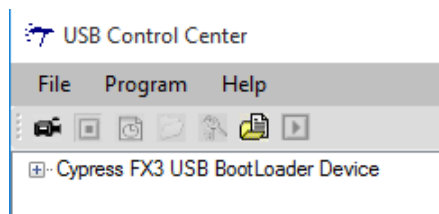
FX2LP および FX3 基板の SW2 は、FX2LP および FX3 チップが USB ポートに初めて接続する時ホストに対してどのように見えるかを制御します。基板が表の通りに設定されると、FX2LP および FX3 は内部ロジックを使用して USB ローダー デバイスとしてエミュレーションし、USB を介してユーザー ファームウェアのダウンロードの準備ができます。

3.2. USB Control Center を用いてファームウェアを開発基板にダウンロード

USB Control Center フォルダ内の CyControl.exe ファイルをダブルクリックすると、USB Control Center コンソールが開きます。図 4 は、FX2LP および FX3 基板が左パネルのデバイス リストにどのように表示されるかを示します。

添付の CyUSB3.inf ファイルが使用された場合、FX3 基板は「Cypress USB Bootloader」として (FX2LP 基盤は「FX2LP Default Device: NO EEPROM」として) 表示されます。

図 4. USB Control Center のデバイス リスト



左パネルに何も表示されていない場合、開発基板は正しく設定されない、または Windows ドライバーがインストールされていない可能性があります。詳細は、[付録](#)を参照してください。

注: 左パネルにすべての接続された USB デバイスを表示する USB Control Center のバージョンがあります。表示を簡素化するために、**Device Class Selection** タブをクリックして、**Devices served by the CyUSB.sys drive...** チェックボックスを除きすべてを選択解除します。

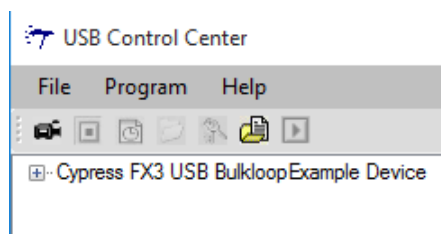
バルクループ ファームウェアを開発基板にダウンロードするために、お使いの基板を選択してください (図 4)。Program をクリックし、FX2 または FX3 を選択し、RAM を選択してファームウェアをオンチップ プログラム メモリにダウンロードします。次に、以下のように対応するファイルに移動します。

FX2: Firmware\Bulkloop_Fx2LP\bulkloop.hex

FX3: Firmware\Bulkloop_FX3\USBBulkLoopAuto\Debug\ USBBulkLoopAuto.img

ファイルをダブルクリックしてファームウェアをロードします。USB ローダー デバイスは、図 5 に示す USB ループバック デバイスに置き換えられます。

図 5. バルクループ デバイス



これは実行中の EZ-USB ReNumeration です。バルクループ ファームウェアがロードされた後、開発基板は自動的に USB から切断し、ダウンロードされたファームウェアによって実装されるデバイスとして再接続します。これで、FX2LP または FX3 開発基板は本アプリケーション ノートの PC サンプル コードの USB ループバック周辺機器として動作できるようになります。

4. アプリケーションの概要

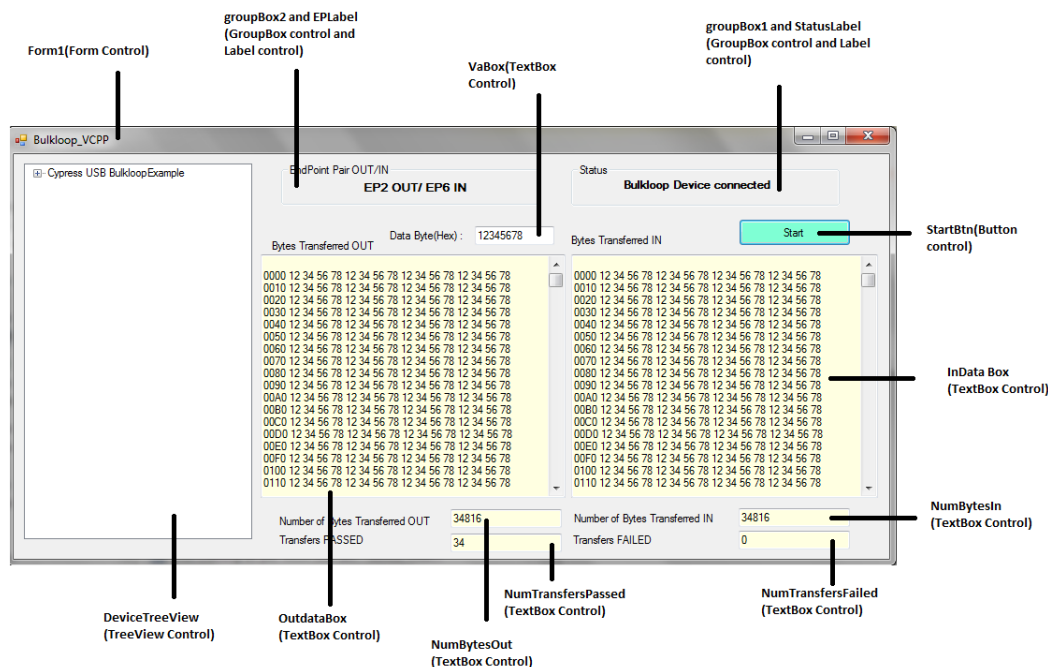
アプリケーションを試用するために、望ましい.NET 言語に対応するディレクトリ内の実行ファイルをダブルクリックします。

- VB_Application\Application\Bulkloop_VB.exe
- VC++_Application\Application\Bulkloop_VCPP.exe
- VCS_Application\Application\Bulkloop_VCS.exe

4.1. 特長

図 6 は Bulkloop アプリケーションのユーザー インターフェースを示します。

図 6. ユーザー インターフェース



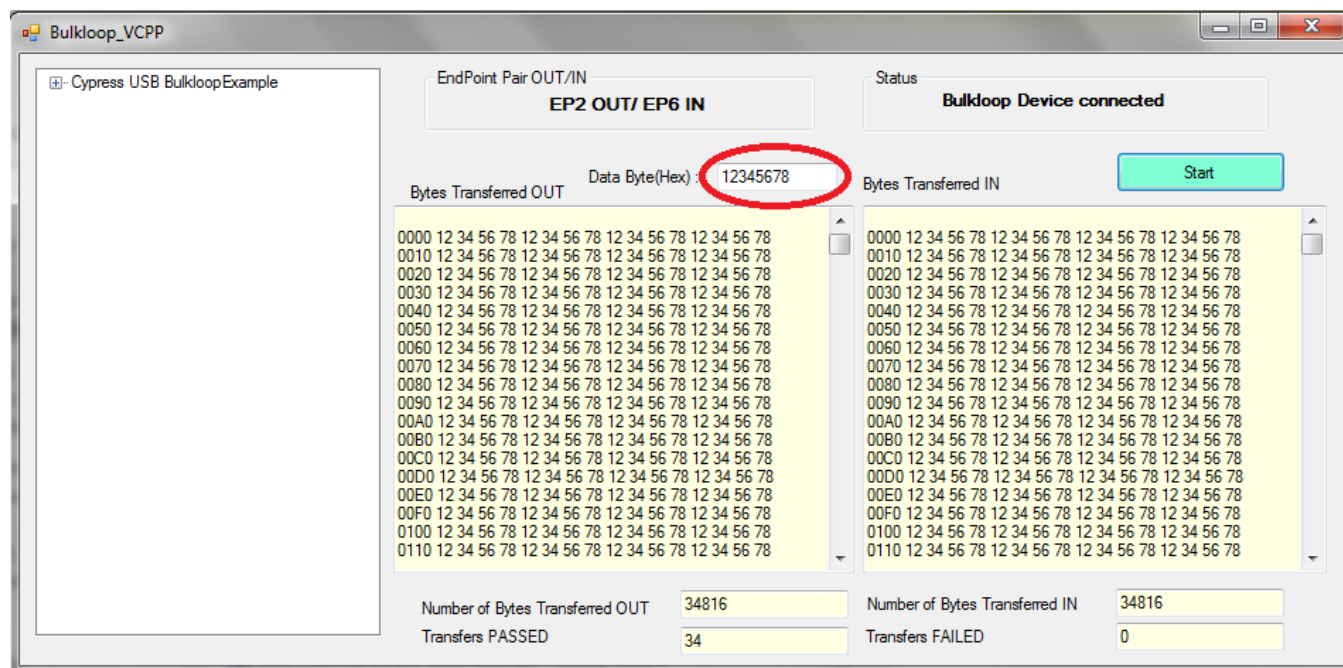
このアプリケーションのユーザー インターフェースは以下のものを示すために設計されています。

- **Status:** 現時点でターゲット デバイスが接続されているか切断されているかを報告します。
- **Endpoint Pair OUT/IN:** ターゲット デバイス上の一对のエンドポイントであり、現時点でターゲットとのデータ転送用の OUT と IN エンドポイントとして使用されます。これは情報提供の目的のみ (ユーザー変更不可) で、ソース コードの設定を反映します。
- **Data Bytes (Hex):** ユーザー選択可能なデータ パターン (最大許容長さは 8 ニブルまたは 4 バイト) で、バルクループ動作中にループバックの形式で転送されます。
- **Bytes Transferred OUT:** ホスト PC からターゲットへ転送されるバイトの記録です。各行は、行内のバイトの開始アドレスで始まります。
- **Bytes Transferred IN:** 転送されるバイトがターゲットからホスト PC までの IN バイトであるという点を除けば OUT と同じです。
- **Number of Bytes transferred OUT:** PC からターゲットに転送されるバイト数です。
- **Number of Bytes transferred IN:** ターゲットから PC に転送されるバイト数です。
- **Start/Stop:** Bulk-loop 動作のスタート/ストップ ボタンです。
- **DeviceTreeView:** Cyusb3.sys に接続されたすべてのデバイスを表示します。

4.2. 動作説明

ループバック デバイスが検出された時、「Status」ボックスに「connected」状態を示し、左パネルに USB デバイスの内部構造を示すツリーを表示します。ユーザーは転送するバイト パターンを入力するか、またはデフォルトの 0x12345678 のバイト パターンのままにすることができます。**Start** ボタンを押します。ループバック デバイスはパケットを連続的に送受信します。

図 7. Bulkloop アプリケーションからのデータ転送



Stop を押して、転送を終了します。バイト カウンターはこのアプリケーションで様々なホスト コントローラーおよび PC を使う USB 転送の帯域幅を測定するために使用できます。

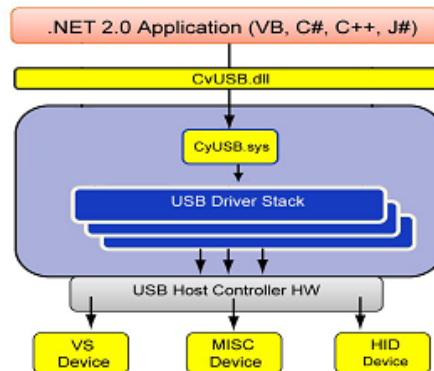
この文書ではこの後、.NET 言語のアプリケーションを詳細に説明します。

5. サイプレスの.NET ライブラリの使用

5.1. CyUSB.dll にリファレンスを追加

CyUSB.dll は、USB デバイスへハイレベルでのプログラミング インターフェースを提供するマネージド Microsoft .NET クラス ライブラリです。Visual Basic.NET, C#, Visual J#や Managed C++といったあらゆる Microsoft Visual Studio.NET の管理言語から CyUSB.dll のクラスとメソッドにアクセスできます。

図 8. CyUSB.dll を使用したアプリケーション開発



サイプレスのライブラリを使用するには、以下の手順を行ってください。

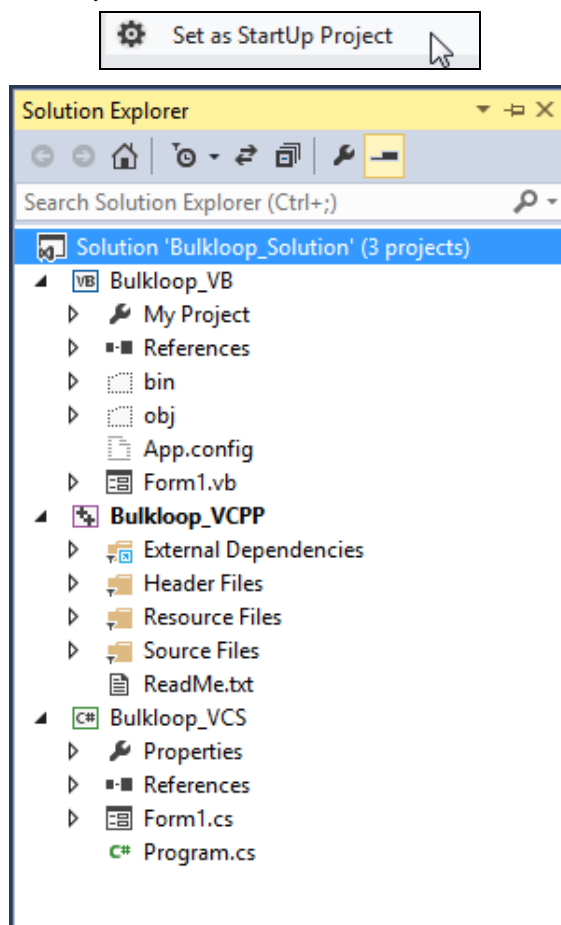
1. プロジェクト内に CyUSB.dll のライブラリ リファレンスを追加します。
2. ライブラリをプログラムに組み込むためにユーザーのソース ファイルに 1 行追加します。

図 9 に、Bulkloop_Exercise ソリューションでの Visual Basic, VC++および VC#プロジェクトにライブラリ リファレンスが表示されることを示します。リファレンス項目を展開した時に CyUSB が表示されない場合、プロジェクト名を右クリックして「reference」項目を選択して、CyUSB.dll を格納するフォルダを開覧します。本アプリケーション ノートとともに提供されるプロジェクトは「C:\AN70983\CyUSB.dll」パスを使用します。他の場所にアプリケーション ノート ファイルをインストールした場合、これらのリファレンスはファイルの場所に一致するように更新する必要があります。

ユーザーが **Build** ボタンをクリックすると、ソリューションでのすべてのプロジェクトがコンパイルされます。あらゆるエラー メッセージは、すべての 3 つのプロジェクトに適用されます。残りの 2 つのプロジェクトを「build」動作から除外することで、使用中のプロジェクトにエラー メッセージを制限できます。「Solution Explorer」パネルでソリューション名を右クリックして、**Properties** を選択して、ビルド動作から除外したいプロジェクトのチェックを外します。

複数のプロジェクトが「build」動作の対象である場合、プロジェクト名を右クリックして、スタートアップ プロジェクトとして設定することで言語バージョンを選択します。これは言語間で切り替えるための手早い方法です。

図 9. CyUSB.dll をすべての 3 つのプロジェクトに追加



言語によって、表 4 に従ってファイルの最上部にコードへのリファレンスを追加します。

表 4. NET 言語用のリファレンス構文

言語	ファイル	追加するライン
Visual Basic	form1.vb	Imports CyUSB
Visual C#	form1.cs	Using CyUSB
Visual C++	form1.h	#using "..\..\Application\CyUSB.dll" (またはユーザーのコンピューターの対応するパス)

5.2. ソースコード概要

下表に、ホスト アプリケーション コードの機能と重要なステップを説明します。

表 5. コード概要

動作／機能	説明	CyUSB.dll API
オブジェクトの初期化	<ul style="list-style-type: none"> • USBDeviceList, CyUSBDevice, CyBulkEndPoint クラスのインスタンスおよび他の必要な変数を作成・初期化 • 「usbDevices」は、クラス ライブラリを介してアクセスできる USB デバイスの動的なリストを表す。デバイスのセレクト マスク (「DEVICES_CYUSB」) により提供されて、Cyusb3.sys によって対応されるすべての USB デバイスを表す USBDevice オブジェクトをその「usbDevices」に挿入 	<ul style="list-style-type: none"> • USBDeviceList は、クラス ライブラリを介してアクセスできる USB デバイスの動的なリストを表す USBDeviceList のインスタンスが作成されると、指定されたデバイスのセレクト マスクによって提供されるすべての USB デバイスを表す USBDevice オブジェクトをそのインスタンスに挿入する。これらの USBDevice オブジェクトは、適切に初期化されており、使用準備ができています • CyUSBDevice クラスは、Cyusb3.sys デバイス ドライバーに接続された USB デバイスを表す • CyBulkEndPoint は CyUSBEndPoint 抽象クラスのサブクラスの 1 つ
デバイスの接続と切断の検出	<ul style="list-style-type: none"> • ホストとターゲットのハードウェア間の接続が確立された時、event_handler の usbDevices_DeviceAttached() 関数が呼び出される • ターゲットのデバイスが切断された時、event_handler の usbDevices_DeviceRemoved() 関数が呼び出される • これらの両方のイベント ハンドラーは順々に RefreshDevice() 関数を呼び出す 	
RefreshDevice()	<ul style="list-style-type: none"> • 対象のデバイスの VID および PID に基づいて対象のデバイスを「myDevice」にインスタンス化。この例での bulkloop ファームウェアは 0x04B4/0x00F0 の VID/PID を使用 • この VID/PID を持つデバイスが複数あれば、「myDevice」は最初に接続したデバイスにインスタンスとハンドルを生成 • ステータス ラベルおよびスタート/ストップ ボタンを有効か無効にする • この例での bulkloop ファームウェアが同じエンドポイントを使用するため、対象のデバイスが見つかった場合、「BulkOutEndpt」と「BulkInEndpt」をそれぞれエンドポイント EP2 と EP6 にインスタンス化 • ツリー表示をクリアして、Cyusb3.sys に接続したデバイスの最新のリストで更新 	<ul style="list-style-type: none"> • USBDeviceList[int VID, int PID]: このインデックスオペレーターは、リストにある USBDevice オブジェクトの VendorID および ProductID プロパティに基づいて、USBDeviceList のエレメントへのアクセスを提供 • EndpointOf(byte addr): アドレス プロパティが addr に等しい CyUSBEndPoint オブジェクトを返す。アドレスが addr に等しいエンドポイントがない場合、「null」を返す
データ転送動作	<ul style="list-style-type: none"> • StartBtn_Click() イベント ハンドラーは Start ボタンをクリックする時に呼び出される • この関数は最初にアプリケーションのユーザー インターフェースの「Data Bytes(Hex)」フィールドからユーザーの値を取得して、ボタンのテキストを「Stop」に変更して、データの送受信が実際に行われる「tXfers」スレッドを開始 • 他の親スレッドはアプリケーションのユーザー インターフェース イベントとその他の OS イベントのみを処理 • スレッドが既に動作中であれば、このイベント ハンドラーはスレッドを中止して、ボタンのテキストを「Start」に戻す 	

動作／機能	説明	CyUSB.dll API
TransfersThread()	<ul style="list-style-type: none"> •SetOutputData() を呼び出して、OUT 転送のためにデータをセットアップ •スレッドが開始された後、OUT および IN 方向のデータ転送は XferData() API で開始される 	XferData(ref byte[] buf, ref int len): XferData メソッドは buf から len バイトを受信／送信。これはデータ転送用の、ライブラリの主な I/O メソッド。このメソッドは同期 (ブロッキング) I/O 動作を実行し、トランザクションが完了するか、またはエンドポイントの TimeOut 期間が経過した時まで結果を返さない。TimeOut 期間が経過する前にトランザクションが正常に完了した場合、「true」を返す。実際に転送したバイト数は len に戻される
SetOutputData()	<ul style="list-style-type: none"> •この関数はアプリケーションのユーザー インターフェースの「Data Bytes(Hex)」フィールド内のユーザー指定のデータを取得して、OUT 転送で転送されるこれらのバイトを「outData[]」バッファに入れる 	
StatusUpdate()	<ul style="list-style-type: none"> •これは UI 更新用のコールバック関数で、TransfersThread から呼び出される。この関数は転送されるバイトと転送されるバイト数で UI のテキストボックスを更新 	

5.3. 非同期通信

このアプリケーションは同期通信方式を使ってホストとターゲット間でデータを送受信します。この方式では、「XferData」データ転送関数は完全なデータが転送されるまで結果を返しません。言い換えれば、アプリケーションは完全なデータ転送が完了するまでブロックされます。その後、次のデータ転送動作は開始します。この方式は殆どのアプリケーションに適しています。しかし、転送に使用可能な完全なデータ帯域幅を必要とするアプリケーションでは、より高度な非同期通信方式が使用されます。非同期通信方式はより複雑で上級ユーザーに適しています。サイプレスの USBSuite とともに提供される「Streamer」例では非同期通信方式を使用する方法が示されています。

6. ファームウェア

ファームウェアはバルク データのループバック動作を行うためにエンドポイントとインターフェースを設定します。2 つのエンドポイントはバルク転送を処理するために設定されます。それらは OUT エンドポイントと IN エンドポイントです。ホストから送信されたデータは OUT エンドポイント バッファに格納されます。このデータは IN エンドポイントに転送された後、要求されると、ホストに戻されます。

6.1. FX2LP Bulkloop ファームウェア

以下のコード スニペットは FX2LP bulkloop ファームウェアの重要なコード機能を説明します。

	説明	コード スニペット	関数名	ファイル名
ディスクリプタ の情報	<ul style="list-style-type: none"> • VID: 0x04B4 • PID: 0x00F0 • デバイス クラス: ペンダー • コンフィギュレーション数: 1 • インターフェース数: 1 • エンドポイント数: 2 • 対応するエンドポイント: <ul style="list-style-type: none"> ◦ EP2 BULK OUT, 512 バイトの最大パケット サイズ ◦ EP6 BULK IN, 512 バイトの最大パケット サイズ 			dscr.a51
初期化	<ul style="list-style-type: none"> • FX2LP ベースのターゲットの異なるコンポーネントの初期化は TD_Init() 関数で行われる。この関数は、初期化のために main() から呼び出される。 		TD_Init()	bulkloop.c
	<ul style="list-style-type: none"> • CPU クロックを 48MHz に設定 	CPUCS = ((CPUCS & ~bmCLKSPD) bmCLKSPD1);		
	<ul style="list-style-type: none"> • FX2LP のインターフェースをコンフィギュレーション • FX2LP は、スレーブ FIFO モードでコンフィギュレーション • インターフェース クロックは、内部 48MHz のクロックとして選択 	IFCONFIG = 0x40;		
	エンドポイントが以下のようにコンフィギュレーションされる: <ul style="list-style-type: none"> • エンドポイント 2 – OUT, バルク, ダブルバッファ, 512 バイト • エンドポイント 6 – IN, バルク, ダブルバッファ, 512 バイト 	EP2CFG = 0xA2; SYNCDELAY; EP6CFG = 0xE2; SYNCDELAY;		
	<ul style="list-style-type: none"> • OUT エンドポイントは、コンフィギュレーション後、ホストからのパケットを受信するように装備する必要がある。エンドポイントがダブルバッファであるため、エンドポイントを装備するために 2 つのパケットをスキップする必要がある。装備の本質は、バッファを解放し、ホストからパケットを受信できるようにすること • バイト カウント レジスタのビット 7 に「1」を書き込むことで、パケットがスキップされる 	EP2BCL = 0x80; SYNCDELAY; EP2BCL = 0x80; SYNCDELAY;		
	<ul style="list-style-type: none"> • TD_Poll 関数内のデータ転送用の AUTO ポインタを有効にする 	AUTOPTTRSETUP = 0x01;		

	内容	コード スニペット	関数名	ファイル名
データ ループバック	<ul style="list-style-type: none"> • bulkloop 実装は TD_Poll()関数で行われる。この関数はデバイスがアイドルの時に繰り返し呼び出される 		TD_Poll()	bulkloop.c
	<ul style="list-style-type: none"> • エンドポイント 2 がデータを含むかを確認する。これはエンドポイント ステータス レジスタ (EP2468STAT) 内のエンドポイント 2 のエンプティ ビットを読み出すことで行われる • エンドポイント 2 が (ホストから送信された)データがある場合は、エンドポイント 6 のデータ受信能力を確認。これはエンドポイント ステータス レジスタ内のフル ビットで示すエンドポイント 6「フル」フラグを読み出すことで行われる • エンドポイント 6 がフルでなければ、データが転送される 	<pre>if(!(EP2468STAT & bmEP2EMPTY)) { if(!(EP2468STAT & bmEP6FULL)) { } }</pre>		
	<ul style="list-style-type: none"> • データを転送するために、データ ポインタが対応するバッファに初期化される。1 番目の自動ポインタがエンドポイント 2 の FIFO バッファの 1 番目のバイトに初期化される 	<pre>APTR1H = MSB(&EP2FIFOBUF); APTR1L = LSB(&EP2FIFOBUF);</pre>		
	<ul style="list-style-type: none"> • 2 番目の自動ポインタがエンドポイント 6 の FIFO バッファの 1 番目のバイトに初期化される 	<pre>AUTOPTRH2 = MSB(&EP6FIFOBUF); AUTOPTRL2 = LSB(&EP6FIFOBUF);</pre>		
	<ul style="list-style-type: none"> • 転送されるバイト数はエンドポイント 2 のバイト カウント レジスタから読み出される。EP2BCL と EP2BCH レジスタは、ホストによって FIFO バッファに書き込まれるバイト数を含む。データがペリフェラル側にコミットされていない限り、これら 2 つのレジスタは FIFO IN と OUT トランザクションに転送されたデータのバイト数を含む 	<pre>count = (EP2BCH << 8) + EP2BCL;</pre>		
	<ul style="list-style-type: none"> • 自動ポインタが有効なので、ポインタが自動的にインクリメントし、EXTAUTODAT2= EXTAUTODAT1 のステートメントはエンドポイント 2 からエンドポイント 6 に転送される。このステートメントを実行するたびに、自動ポインタがインクリメントされる。このステートメントは、各バイトをエンドポイント 2 からエンドポイント 6 に転送するために繰り返し実行される 	<pre>for(i = 0x0000; i < count; i++) { EXTAUTODAT2 = EXTAUTODAT1; }</pre>		
	<ul style="list-style-type: none"> • データが転送された後、エンドポイント 2 は、ホストから新しいパケットを受信するように再装備する必要がある。エンドポイント 6 は「コミット」される、つまり FIFO バッファはホストがエンドポイント 6 からデータを読み出せるようにされる 	<pre>EP6BCH = EP2BCH; SYNCDelay; EP6BCL = EP2BCL; SYNCDelay; EP2BCL = 0x80;</pre>		

6.2. FX3 Bulkloop ファームウェア

以下のコード スニペットは FX3 bulkloop ファームウェアのコード機能を説明します。

	内容	FX3 API	関数名	ファイル名
ディスクリプタ の情報	<ul style="list-style-type: none"> • VID: 0x04B4 • PID: 0x00F0 • デバイス クラス: ペンダー • コンフィギュレーション数: 1 • インターフェース数: 1 • エンドポイント数: 2 • 対応するエンドポイント: <ul style="list-style-type: none"> ▫ EP2 BULK OUT, SS の場合は最大パケット サイズが 1024 バイト, HS の場合は 512 バイト ▫ EP6 BULK IN, SS の場合は最大パケット サイズが 1024 バイト, HS の場合は 512 バイト 			cyfxbulkpdsc r.c
初期化	<ul style="list-style-type: none"> • FX3 デバイスが初期化される • FX3 に内蔵する 61 本の I/O ピンの機能モードを設定 • RTOS Kernel は、main()の次に呼び出されるこれは、非返し呼び出しで、Threadx kernel を設定。この関数は、main() 関数からの最後の関数として呼び出す必要がある 	CyU3PDeviceInit() CyU3PDeviceConfigureIOMatrix() CyU3PKernelEntry()	main()	cyfxbulkpau t.o.c
	<ul style="list-style-type: none"> • FX3 アプリケーション定義機能は、デバイスとすべてのモジュールが初期化された後、システム モジュールから呼び出される。アプリケーション スレッドとその他の必要な OS プリミティブは、ここで作成できる。スレッドを少なくとも 1 つ作成する必要がある。この関数は、直接呼び出してはならない • BulkLpAppThread_Entry は作成したアプリケーション スレッド 	CyU3PThreadCreate()	CyFxApplicationD efine()	
	<ul style="list-style-type: none"> • CyFxBulkLpApplnDebugInit() はデバッグ モジュールを初期化 • CyFxBulkLpApplnInit()はバルクループ アプリケーションを初期化 		BulkLpAppThread _Entry()	
	<ul style="list-style-type: none"> • USB 3.0 のプラットフォーム内の USB デバイス モード ドライバーを起動し、コントロール エンドポイントに必要な DMA チャネルを作成 • USB ドライバーで USB セットアップ要求ハンドラーを登録 • USB ドライバーで USB イベント コールバック関数を登録。CyFxBulkLpApplnUSBEventCB はこのアプリケーションで登録される • USB 3.0 LPM 要求ハンドラー コールバックを登録 • USB ドライバーで USB ディスクリプタを登録。ドライバーは、サポートされている異なるタイプごとの 1 つのディスクリプタを記憶でき、最大 8 つの異なる文字列のディスクリプタを記憶できる。ドライバーは、この関数に渡されるディスクリプタポイントを格納するだけで、ディスクリプタのコピーを作らない。そのため、USB ドライバーがアクティブの間に、呼び出し側はこれらのディスクリプタバッファを解放してはならない • USB 接続を有効化/無効化。この関数は、USB 3.0 プラッ 	<ul style="list-style-type: none"> • CyU3PUsbStart() • CyU3PUsbRegisterSetup Callback() • CyU3PUsbRegisterEvent Callback() • CyU3PUsbRegisterLPMR equestCallback() • CyU3PUsbSetDesc() • CyU3PConnectState() 	CyFxBulkLpApplnI nit()	

	内容	FX3 API	関数名	ファイル名
	<p>トフォーム上の USB PHY を有効／無効にし、そのように USB ホストへの接続を制御するために使用される</p> <ul style="list-style-type: none"> CyFxBulkLpApplnStart ()が USB エンドポイントのプロパティをコンフィギュレーションする Set Configuration イベントの時に呼び出される 	<ul style="list-style-type: none"> CyU3PSetEpConfig() 	CyFxBulkLpApplnStart ()	
データ ループバック	<ul style="list-style-type: none"> DMA チャンネルを作成 DMA チャンネルは、プロデューサー ソケット、コンシューマー ソケット、バッファ式およびコールバック関数を必要とする。この例では、自動チャンネルは、プロデューサー ソケット EP2 とコンシューマー ソケット EP6 の間で作成される。EP2 は、データを受信するたびに、自動的に EP6 にコミットされる DMA チャンネルで転送を設定。この関数は、選択した DMA チャンネルでトランザクションを開始 	<ul style="list-style-type: none"> CyU3PDmaChannelCreate() CyU3PDmaChannelSetXfer() 	CyFxBulkLpApplnStart ()	

7. トラブルシューティング

- ホストアプリケーションがコンパイルに失敗し、CyUSB.dll が正しく追加されず、参照のために組み込まれていない場合、CyUSB.dll の API にアクセスすることはできません。「[CyUSB.dll にリファレンスを追加](#)」の指示に従ってください。
- CyUSB の API が失敗した場合、それぞれの API のリターンコードを確認してください。リターンコードの値とその意味は、[SuiteUSB](#) をインストールした後、"<InstallDirectory>\Cypress Suite USB 3.4.x\CyUSB.NET\CyUSB.NET"にある CyUSB.dll のマニュアルに記載されています。
- XferData/BeginDataXfer API が失敗した場合、XferData または BeginDataXfer API の最後の呼び出し時のエラーコードを取得するには、"UsbdStatus"または "UsbdStatusString()"を使用します。
- API が、デバイスにアクセスすることができない場合は、デバイスハンドルが適切に割り当てられていることを確認してください。
- プロジェクトが Visual Studio 2012(VS 2010)で開けない場合は、VS 2010 で新しいプロジェクトを作成し、ソースファイルを移動して、CyUSB.dll ライブラリをプロジェクトに追加します。全ての既存の機能は VS 2010 以降のバージョンで動作します。

8. システム要件

8.1. ハードウェア

8.1.1. FX2LP

FX2LP DVK (CY3684) 基板は、この FX2LP の例では開発およびテスト プラットフォームとして使用されます。FX2LP DVK のインストール後、DVK の詳細な回路図がハードウェアのフォルダ内に用意されています。基板の詳細は、[FX2LP DVK](#) と共に提供されている [EZ-USB_Getting Started](#) 資料の「EZ-USB Advanced Development Board」に記載されています。

8.1.2. FX3

FX3 DVK (CYUSB3KIT-001 または CYUSB3KIT-001) 基板は、この FX3 の例では開発およびテスト プラットフォームとして使用することができます。(適切な FX3 DVK のインストール後) DVK の詳細な回路図がハードウェアのフォルダ内に用意されています。

8.2. 開発ソフトウェア

- Microsoft .NET framework 4.5
- ControlCenter - [サイプレスの USB スイート](#)をインストールすることで使用可能です。
- Keil uVision 2 – 4K バイトの評価バージョンが CY3684 DVK で提供されます。完全版については、Keil 社にお問い合わせください。
- Eclipse IDE – [FX3 SDK](#) をインストールすることで使用可能です。
- Microsoft Visual Studio 2010 または 2012

9. 関連するリソース

- [UsbSuiteQuickStartGuide](#) は Visual C#での CyUSB.dll を使用した開発、および Visual C++での CyAPI.lib を使用した開発の初心者役に立ちます。
- プログラマー参考用の C#ライブラリ (CyUSB.NET.pdf) は、[サイプレスの USB スイート](#)と共に提供されています。
- [サイプレスの USB スイート](#)や [FX3 SDK](#) と共に複数のアプリケーションの例も提供されています:
 - Streamer: データ転送のテストに使用できます。
 - Control Center: CyUSB3.sys によって USB デバイスと通信する用のユーザー インターフェースです。
- 「[EZ-USB FX2LP/FX3 Developing Bulk-Loop Example on Linux](#)」では、libusb を使用して、サイプレスの EZ-USB FX2LP/ FX3 製品用の Linux ベースのオペレーティング システムで、USB ホスト アプリケーションを開発する方法を説明します。このアプリケーションノートは参考のために提供されています。サイプレスはまた Linux 用の SDK も提供します。それは、[FX3 SDK](#) のウェブページから参照できます。
- 「[EZ-USB FX2LP - Developing USB Application on MAC OS X using LIBUSB](#)」では、libusb-1.0 を使用して、サイプレスの EZ-USB FX2LP 製品用の USB ホスト アプリケーション (Cocoa プリケーション) を Mac OS X 10.6/10.7 で開発する方法を説明します。このアプリケーションノートは参考のために提供されています。サイプレスはまた Linux 用の SDK も提供します。それは、[FX3 SDK](#) のウェブページから参照できます。
- USB SuperSpeed コード例のすべての一覧については、<http://www.cypress.com/101781> を参照してください。また、USB Hi-Speed コード例のすべての一覧については、<http://www.cypress.com/101782> を参照してください。

10. まとめ

このアプリケーションノートでは、サイプレスの FX2LP や FX3 デバイスと通信する実装のための、Microsoft.NET 言語用のサイプレス ライブラリを使用して、ホスト PC のアプリケーションを構築するためのいろいろな側面について説明します。

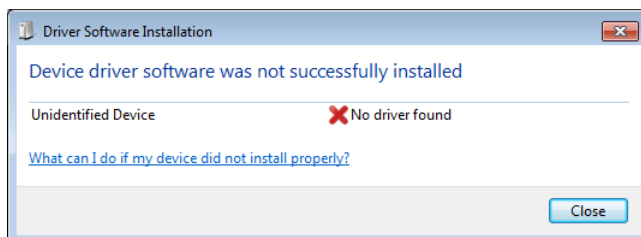
11. 用語集

用語	意味
Microsoft Visual Studio	Microsoft Visual Studio は、Microsoft 製の 統合開発環境 (IDE)。Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework, Microsoft Silverlight にサポートされるすべてのプラットフォームに、Windows Forms や WPF アプリケーションと共に、コンソールおよびグラフィカル ユーザー インターフェイス アプリケーションをネイティブ コードとマネージ コード両方で開発するために使用
Microsoft Visual C#	マネージ環境向けに設計された Microsoft バージョンの C#言語
Microsoft Visual C++	C, C++および C++/CLI プログラミング言語用の、Microsoft の商業用の IDE 製品
Microsoft Visual Basic	Visual Basic .NET (VB.NET) は、.NET Framework に実装される、従来の Visual Basic (VB) の進化と考えられる オブジェクト指向コンピューター プログラミング言語。Visual Basic は、Microsoft 社の COM プログラミングモデル用の、Microsoft 製第 3 世代のイベント駆動型プログラミング言語および IDE
CY3684 DVK	EZ-USB の FX2LP ファミリー用のサイプレス開発キット。詳細については、 サイプレスのウェブサイト を参照
CYUSB3KIT- 001 / CYUSB3KIT- 003 (EZ-USB ® FX3™ SuperSpeed Explorer Kit)	EZ-USB の FX3LP ファミリー用のサイプレス開発キット。詳細については、 CYUSB3KIT-001/ CYUSB3KIT-003 を参照
EZ-USB	FX2LP 等、サイプレスのすべてのハイスピード USB 製品の総称。
FX2LP	サイプレスのハイスピード USB コントローラー。詳細については、 CYUSB3KIT-001/ CYUSB3KIT-003 を参照
FX3	サイプレスの EZ-USB FX3 は、次世代の USB3.0 ペリフェラル コントローラーであり、統合性と柔軟性のある機能を提供しています。詳細については、 サイプレスのウェブサイト を参照。
サイプレスの USB スイート	Visual Studio 用のサイプレスの USB 開発ツール。詳細については、 サイプレスのウェブサイト を参照
CyUSB.dll	.NET 技術を使用したソフトウェア開発用のサイプレス ライブラリ。詳細については、サイプレスの USB スイートと共に提供するプログラマー参考用の C#ライブラリ (CyUSB.NET.pdf) を参照
Cyusb3.sys	USB 通信用のサイプレス Windows ドライバー。詳細については、「 Cypress CyUsb3.sys Programmers Reference 」を参照
USB Control Center	デバイスの識別とコードのダウンロードなどの汎用 USB 機能用のサイプレス アプリケーション。このアプリケーションは サイプレスの USB スイート を通じて提供される
Keil uVision	4K バイトの評価バージョンが CY3684 DVK で提供される。完全版については、 Keil 社にお問い合わせください。
Eclipse IDE	FX3 SDK は、開発ツールの一部として、C/C++開発者用 Eclipse IDE が提供される。Eclipse ベースのプラットフォームと CPP 機能が含まれる。このファームウェアの開発環境は、FX3 のデバッグ ファームウェアアプリケーションの開発、ビルドおよびデバッグに有用

A 付録: FX2LP/FX3 DVK ドライバーの Windows へのインストール

Windows コンピューターに FX2LP/FX3 DVK キットをインストールしていない場合、コンピューターに DVK を初めて接続する際に次のメッセージが表示されます。

図 10. ドライバー ソフトウェア インストール



メッセージ ボックスを閉じて、Windows Device Manager を開きます。これを行うには **Start** ボタンをクリックし、右パネルの **Computer** を右クリックし、**Properties** を選択し、「System Information」を表示させます。左パネルの「**Device Manager**」をクリックします。Device Manager では、FX2LP は「Unknown Device」として表示し、FX3 は「WestBridge」として表示します。

図 11. 「Unknown Device」として表示される FX2LP

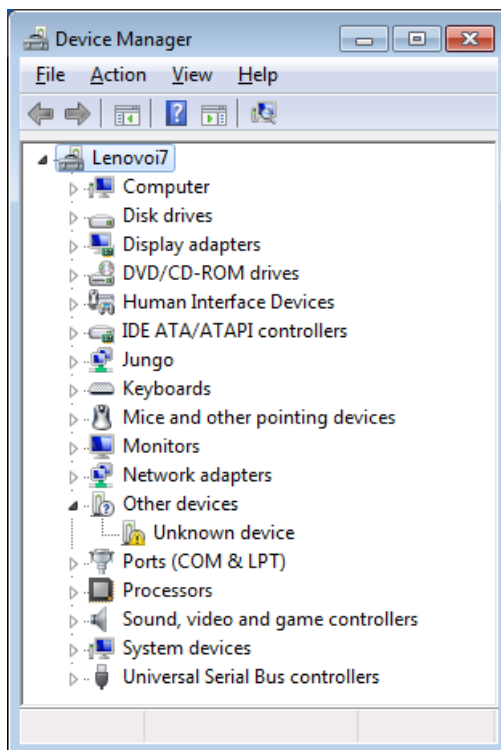


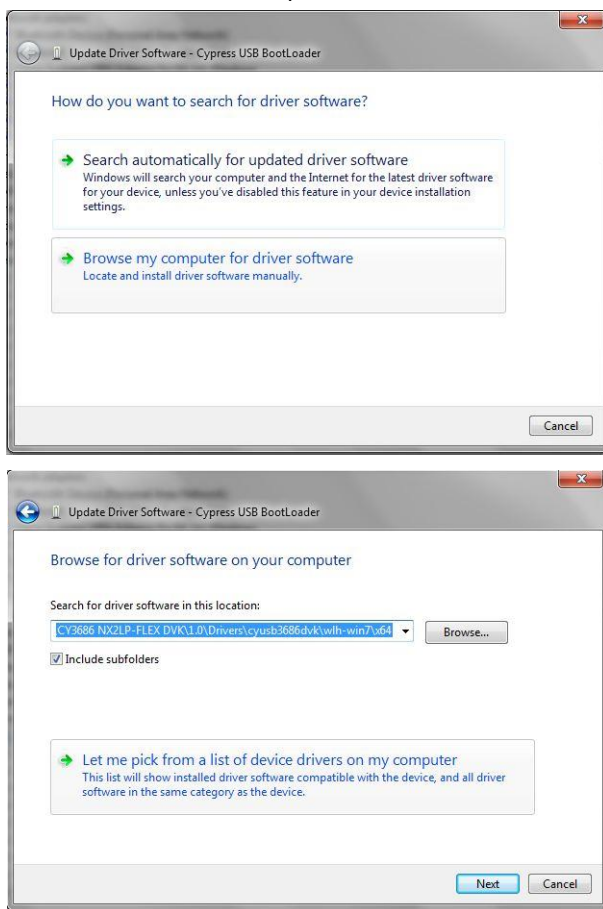
図 12. 「WestBridge」として表示される FX3



デバイス(感嘆符がある) を右クリックし、「Update Driver」を選択します。

「Browse my computer for driver software」を選択します。ドライバー ソフトウェア フォルダを参照し、「Next」をクリックします。

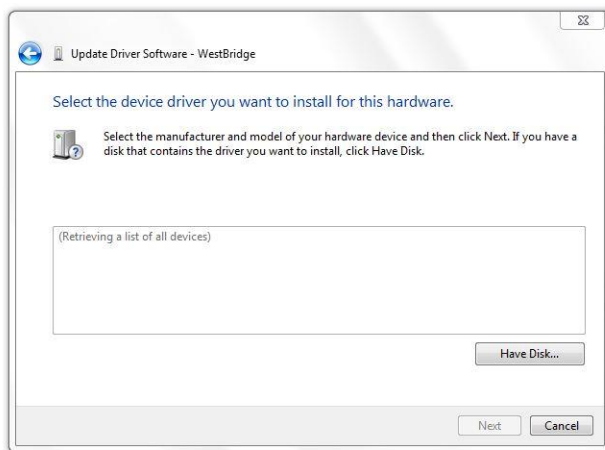
図 13. Update Driver



「Have Disk」ボタンをクリックし、「AN70983_Project\Drivers」を開き、「CyUSB3.inf」を」選択し、「OK」をクリックします。それぞれのオペレーティング システムの次の表に示すように、(AN70983_Project\Drivers) 内の inf フォルダを使用してください。

オペレーティング システム	フォルダ パス
Windows XP 32 ビット	wxp\x86
Windows 7 32 ビット	win7\x86
Windows 7 64 ビット	win7\x64
Windows Vista 32 ビット	wlh\x86
Windows Vista 64 ビット	wlh\x64

図 14. 「Have Disk」をクリックして「CyUSB3.inf」を選択



Next をクリックしてインストールを続行します。ドライバーをインストールした後、Device Manager ウィンドウは、黄色の感嘆符付きのエントリを削除し、正常にドライバーにバインドされた、認識されたデバイスを示します。

図 15. ドライバーをバインドした後のサイプレス デバイス



改訂履歴

文書名: AN70983 – EZ-USB® FX2LP™/FX3™のバルク転送ホスト アプリケーションの設計

文書番号: 001-98030

版	ECN	改版者	提出日	変更内容
**	4802512	HZEN	09/09/2015	これは英語版 001-70983 Rev. *C を翻訳した日本語版 001-98030 Rev. **です。
*A	5478178	HIKA	10/17/2016	これは英語版 001-70983 Rev. *D を翻訳した日本語版 001-98030 Rev. *A です。 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。
*B	5822982	AESATP12	07/18/2017	更新されたロゴと著作権。
*C	5934997	HIKA	10/19/2017	これは英語版 001-70983 Rev. *F を翻訳した日本語版 001-98030 Rev. *C です。

ワールドワイド販売と設計サポート

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

製品

ARM® Cortex® Microcontrollers	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/RF	cypress.com/wireless

PSoC® ソリューション

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

サイプレス開発者コミュニティ

[フォーラム](#) | [WICED IOT Forums](#) | [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

All other trademarks or registered trademarks referenced herein are the property of their respective owners.



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

© Cypress Semiconductor Corporation, 2015-2017. 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付属せず、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、明示又は黙示を問わず、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。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という。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問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。