AN4053

EZ-USB® FX2™ および FX2LP™のアイソクロナスまたはバルクエンドポイントを介したデータのストリーミング

著者: Rama Sai Krishna Vakkantula

関連プロジェクト: あり

関連製品ファミリー: CY7C68013A

ソフトウェアバージョン: なし

関連アプリケーションノート: AN65209, AN61345

本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://www.cypress.com/go/AN4053 をご覧ください。

オーディオやビデオなどのデータ型を使用する USB アプリケーションでは、連続した高速なデータの流れが必要です。このデータは途切れの無いストリームで流れるため、ストリーミングと呼ばれます。このアプリケーションノートでは、ストリーミングデータをサポートする USB の高帯域幅の配信メカニズムについて説明します。また、FX2LP の高帯域エンドポイントの実装と実行を行うためのコードを含みます。さまざまな転送タイプの選択や転送レートの測定のための、付属 PC アプリケーションが用意されています。このアプリケーションノートに添付されているサンプルコードは、サイプレス USB フレームワークを使用して、オルタネート USB 設定の実装方法と、代替となる転送レートのホストへの設定について紹介します。USB Hi-Speed コード例の全一覧については、http://www.cypress.com/?rID=101782 を参照してください。

目次

1 はじめに 1
2 関連プロジェクトファイル 2
3 ストリーミング アプリケーション 2
4 高帯域幅転送 4
4.1 データ PID シーケンシング (HS 操作) 4
5 Cypress SuiteUSB 5
6 ストリーミング ファームウェアの例 5
7 Cypress USB フレームワークの使用 6
7.1 fw.c 7
7.2 dscr.a51 7
7.3 CyStream.c 9
8 CyStreamer.hex の実行 11
9 パフォーマンス分析 13
10 FX2LP と イメージセンサのインターフェース 14
11 参考文献 14
12 概要 14
13 改訂履歴 16
セールス, ソリューションおよび法律情報 17

1 はじめに

USB は、「アイソクロナス」(ISO) と呼ばれる転送タイプのストリーミングデータを提供します。ラテン語では、iso は等価、chron は時間を意味します。ストリーミングデータを含むデータパケットのための各 USB タイムフレームには同じ時間が与えられます。

USB ISO 転送には 2 つの重要な特性があります。

1. CONTROL、BULK、または INTERRUPT 転送とは異なり ISO 転送では、ハンドシェイクや、破損または欠落したパケットのリトライは行いません。このようなオーバーヘッドは、タイムフレームごとに保証されたデータを配信するという重要な ISO 機能を妨げます。
2. USB ISO 転送メカニズムは保証されたデータバケットを提供しますが、途切れることのないデータのストリームを実現するためには、データのプロバイダとコンシューマはすばやくバケットを埋めて空にする必要があります。

破損しているか欠落している ISO パケットは再送されませんが、コンシューマ側はその発生を検出して是正処置を取れます。例えば、欠落しているデータチャンクは、補間、または以前のデータの繰り返しによって置き換えられます。しかし、ストリーミングアプリケーションをうまく設計するためには、データのプロバイダ（たとえばカメラ）とコンシューマ（たとえば PC）の両方がストリーミングデータの優先度を高くして、パケットの誤ロードやアンロードが起こらないようにする必要があります。

INTERRUPT エンドポイントは、ストリーマにも成れます。

USB 仕様バージョン 2.0 では、ハイスピード（480 Mbits/sec）動作のみの高帯域のエンドポイントを定義しています。高帯域幅のエンドポイントは、1 つのマイクロフレームにつき 1 つ以上、最大 3 パケットまでのパケットを送信または受信できます。したがって、ハイスピードのデバイスでは、INTERRUPT 転送を使用したデータのストリーミングもできます。フルスピードのデバイスでは、INTERRUPT エンドポイントの最大パケットサイズは、ISO の最大パケットサイズ 1023 バイトと比較すると 64 バイトであるため、INTERRUPT 転送は高帯域幅ストリーミングアプリケーションには適していません。

このアプリケーションノートでは、ハイスピード USB ISO ストリーミングデータ用に FX2LP をプログラミングする方法を紹介します。また、比較のためにフルスピード転送の例も記載しています。以前の FX2（LP ではない）デバイスもまた、少しの変更で使用できます（このドキュメントで説明しています）。帯域幅を比較する目的で、一組の BULK 転送も記載しています。

2 関連プロジェクトファイル

このアプリケーションノートに添付されている zip ファイル（AN4053.zip）には、次のフォルダがあります。

1. FX2LP ストリーマソースコード

Keil プロジェクトファイル CYStream.Uv2 は、`Firmware\CyStreamer` フォルダにあります。これをダブルクリックすると、連続してストリーミングデータを送信/受信する FX2LP ファームウェアを実装した Keil プロジェクトが起動します。Keil IDE の出力は *CyStream.hex* ファイルで、FX2LP 開発ボードに書き込みすることが出来ます。

2. VS_Control_Center

この Microsoft Visual Studio C#アプリケーションを使用して、FX2LP hex コード（*CyStream.hex*）を FX2LP 開発ボードにダウンロードします。実行可能ファイルは、`\bin` ディレクトリにあります。また、Microsoft Visual Studio で *.sln*（ソリューション）ファイルを読み込んで、ソースコードを確認/変更することができます。

3. VS_Streamer

この Microsoft Visual Studio C ++プロジェクトは、PC から FX2LP 開発ボードにデータを流す、CyStream コードを実行するホスト側のアプリケーションです。このアプリケーションは、FX2LP ストリーミングファームウェアのさまざまな帯域幅の設定をテストするために使用されます。実行可能ファイルは、`\bin` ディレクトリにあります。また、*.sln* ファイルをロードしてソースコードを確認/変更することができます。

このアプリケーションは、Cypress SuiteUSB をインストールすることで入手できます。このアプリケーションは、`Cypress Suite USB 3.4.7\CyAPI\examples\Streamer` にあります。

また、Cypress SuiteUSB のインストールとともに、C#ベースのストリーマアプリケーションも入手できます。C#のバージョンは、`Cypress Suite 3.4.7\CyUSB.NET\examples\Streamer` にあります。

以下のセクションでは、ISO ストリーミングの詳細と、FX2LP を使用して ISO のデータストリームを実装するために必要な情報を説明します。

3 ストリーミング アプリケーション

USB デバイスを PC に接続すると、PC はエニュメレーションというプロセスを使用して、デバイスの特性とその要求事項を知ります。エニュメレーション中、ホストは、使用するデバイスドライバ、およびデバイスに含まれるエンドポイント（データソースおよびシンク）の数と特性などの詳細を確認するためにデバイスに問合せを行います。ISO エンドポイントの場合、ホストはディスクリプタを読み取り、ISO 転送中にデバイスの要求事項を確認します。主要な ISO パラメータは、ISO エンドポイントディスクリプタの *wMaxPacketSize* フィールドで表される、必要帯域幅です。ISO 帯域幅の制限は次のとおりです。

- フルスピード転送の場合、ホストは 1 ミリ秒フレームごとに 1 つの ISO パケットを入れることができます。パケットサイズは最大 1023 バイトです。1 ミリ秒ごとに 1023 バイトのため、最大転送速度は 1.023 メガバイト/秒です。

- ハイスピード転送の場合、ホストは 125 マイクロ秒フレームごとに最大 3 つの ISO パケットを入れられます。パケットサイズは最大 1024 バイトです。1/8 ミリ秒ごとに 3072 バイトのため、最大転送速度は 24.576 メガバイト/秒です。

両端のプロバイダおよびコンシューマが、IN と OUT のエンドポイント FIFO にデータをすばやく移動させ、望ましい転送レートを達成できる場合にのみ、データは中断することなくストリーミングされます。FX2/FX2LP が持つ GPIF（General Programmable Interface）と呼ばれるハードウェアのデータ転送部は、このようなスピードでデータを処理するために必要です。

USB には、ホストが複数のストリーミング帯域幅を選択できるように、オルタネート設定と呼ばれる仕組みがあります。例として、USB カメラでのオルタネート設定を表 1 に示します。

表 1. USB カメラの 3 つの帯域幅の設定

オルタネート設定	wMaxPacketSize
1	1023
2	512
3	256

PC は、バス上にこの USB カメラ一つだけではない可能性があるため、接続されているすべてのデバイスにその USB 帯域幅を割り当てる必要があります。

この帯域幅のネゴシエーションプロセス中、1 フレームあたり 700 バイトしか USB カメラに保障できないと、ホストが判断します。カメラの ISO エンドポイントディスクリプタに 1023 バイトの設定 1 つしかない場合、ホストはカメラを設定（使用）せず、アプリケーションはフェイルします。しかし、カメラには他の選択肢があるため、ホストはオルタネート設定＃2 を選択できます。この場合、ホストは、要求するオルタネート設定を含む（SET_INTERFACE を使用して）メッセージをカメラに送信します。これによって、カメラのファームウェアに、おそらくフレームレートを下げることによって、帯域幅要件を減らすよう指示します。

ISO 帯域幅のオルタネート設定は 1 から始まります。デフォルト設定の 0 は ISO 帯域幅を消費してはならないためです。これらのパケットのサイズは最大値を使います。 デバイスは、任意のパケットで小さいバイト数を転送することができます。長さゼロの ISO パケットにも意味があります。これは、デバイスが「ジャストインタイム」の ISO IN 要求を満たすデータを供給できなかったことを示すために送信されます。

周辺機器の設計者は、スケジューリング効率のために、より大きなパケットサイズを使うことを選択できます。例えば、4 つの 256 バイトパケットの代わりに 1 つの 1024 バイトパケットを使用するケースです。この決定は、通常、ペリフェラル内のデータバッファの数とサイズに依存します。この変更に対応するために、USB は高帯域エンドポイント用の *bInterval* フィールドを定義します。この番号は、転送要求の間にスキップするマイクロフレームの数をホストに知らせます。このアプリケーションノートの記載を含む、ほとんどの高帯域幅アプリケーションでは、*bInterval* は 1 に設定され、「すべてのマイクロフレームで転送をスケジュールする」ことを示します。

BULK エンドポイントは、高帯域エンドポイントとはみなされませんが、BULK エンドポイントの(予定されない)バースト的な転送を処理するのに十分なバッファリングが提供されていれば、データをストリームするために使用できます。ただし、これは推奨しません。軽負荷の USB システムではカメラが動作する可能性がありますが、USB ディスクドライブなどのデバイスを増設すると、カメラへの BULK 転送が遅くなる可能性があるためです。

USB のストリーミングアプリケーションを設計する際には、下記を考慮してください。

- 転送スループット：求められる 1 秒あたりのバイト数
- FX2LP は、プログラミングによる（MCU 関連の）転送を使用してデータを転送するのに十分な速度があるか、あるいは、GPIF を使用して FIFO データを転送する必要があるか？
- システムのバッファリング要件： ISO エンドポイントは仕様によってはダブルバッファですが、FX2LP などの IC は必要に応じて最大 4 つのバッファを提供します。
- 必要な帯域幅とデータレートを満たす適切なエンドポイントタイプ
- マイクロフレームあたりに必要なパケット数
- 選択した高帯域幅転送の PC ドライバのサポート

4 高帯域幅転送

USB では、バスの時間をフレームと呼ばれる固定長セグメントに分割します。フルスピードデバイスの場合、ホストは 1 ミリ秒ごとにフレームマーカー(SOF、Start of Frame)を発行します。ハイスピードデバイスの場合、ホストは 125 マイクロ秒ごとにマイクロフレームを発行します。

マイクロフレームあたり 1024 バイト以上を必要とするハイスピードなエンドポイントは、高帯域幅エンドポイント呼ばれます。高帯域幅エンドポイントは、マイクロフレームごとに最大 3 つのパケットを転送できます。各パケットは最大 1024 バイトです。マイクロフレームあたりの転送回数は、エンドポイントディスクリプタの *wMaxPacketSize* フィールドで定義されます。マイクロフレームを使用しているため、ハイスピード動作の場合にのみ高帯域エンドポイントが可能です。

USB2.0 仕様では、ピリオディックエンドポイント(ISO または Interrupt)は、必要なバスアクセス周期を指定する必要があります。これは、エンドポイントディスクリプタの *bInterval* フィールドを設定して指定します。エンドポイントディスクリプタの *bInterval* フィールドは、エンドポイントがホストによってポーリングされる最大レートを定義します。これにより、ホストがエンドポイントにサービスを提供するレートを遅くするメカニズムを提供します。

同じマイクロフレームでのデータパケット転送を追跡するため、高帯域幅のアイソクロナス転送では、パケット ID(PID)シーケンスと呼ばれるメカニズムを使用しています。高帯域幅の ISO エンドポイントを設計するためには、以下で説明するデータ PID シーケンスを理解することが重要です。

4.1 データ PID シーケンシング (HS 操作)

フルスピード USB デバイスは、2 つのデータ PID、DATA0 と DATA1 を使用します。USB 2.0 仕様では、ハイスピード動作用の 2 つの PID、MDATA と DATA2 が追加されています。PID シーケンシングでは、マイクロフレーム内にて紛失または破損したパケットを検出します。表 2 は、異なるペイロードサイズの IN および OUT ISO 転送の PID シーケンス順序を示しています。

表 2. 高帯域幅 ISO 転送 PID シーケンス

利用可能パケット	方向	データ PID シーケンス		
		Pkt 1	Pkt 2	Pkt 3
3	IN	DATA2	DATA1	DATA0
2	IN	DATA1	DATA0	–
1	IN	DATA0	–	–
3	OUT	MDATA	MDATA	DATA2
2	OUT	MDATA	DATA1	–
1	OUT	DATA0	–	–

IN 転送では、ホストはエニュメレーション中に *wMaxPacketSize* フィールドによってデバイスから報告されたマイクロフレームあたりのパケット数まで受信することを期待します。デバイスによって送信された最初の DATA PID は、ホストに、マイクロフレームで発行すべき IN トークンの数を示します。 DATA2 はあと 2 つを意味し、DATA1 はあと 1 つを意味し、そして DATA0 はもう無いことを意味します。 デバイスにデータがない場合、デバイスは DATA0 PID を使用してゼロレングスパケット(ZLP)を送信します。

OUT 転送では、ホストは最後のパケットを除く全てのパケットに MDATA(More Data)PID を使用します。最後の PID は、ホストが送信した MDATA PID の数を示す DATA PID です。

4.1.1 FX2 (LP ではない) ISO IN 転送の検討

エニュメレーション中、ハイスピード ISO デバイスは、マイクロフレーム内で転送することが予想される追加パケットの数をホストに通知します。デバイスは、*wMaxPacketSize* というエンドポイントディスクリプタフィールドにこれを示します。ビット 12:11 は、マイクロフレームあたりパケットが、追加なし、1 つ追加、または 2 つ追加を指定します。これらは、USB 2.0 仕様ではトランザクション機会と呼ばれ、デバイスがマイクロフレームで使用する必要のないパケットの最大数を表します。

FX2 は、最初のホスト IN トークンがマイクロフレーム内に到着するまでに、実際にどれだけの IN FIFO バッファが埋められて、USB 転送用にコミットされたかに応じて、正しい DATA PID を送信します。コード例では、*wMaxPacketSize* フィールドにてマイクロフレーム内で 3 つの IN パケット(最大)が可能であると報告します。ただし、マイクロフレーム内で 2 つのパケットをコミットしただけの場合は、DATA1 PID(表 2)を使用して最初のパケットを送信する必要があります。

正しい ISO-IN PID のシーケンシングを自動的に実現するため、FX2LP は、EPnISOINPKTS と呼ばれる 4 つのレジスタを提供します(n = 2,4,6、または 8)。

b7	b6	b5	b4	b3	b2	b1	b0
AADJ	0	0	0	0	0	INPPF1	INPPF0
0	0	0	0	0	0	0	1

フレームあたりの IN パケット数(INPPF)は 1 にデフォルト設定されていますが、1,2 または 3 に設定されている場合があります。

自動 PID シーケンスを有効にするには、自動調整(AADJ)を 1 に設定する必要があります。

AADJ = 0 の場合、INPFF [1:0]で与えられる PID を使用して、無条件に ISO IN 転送を開始します。

初期の FX2 デバイスにはこの自動メカニズム(AADJ ビットなし)がないため、プログラマーは ISO IN 転送中に正しい Data PID シーケンスが送信されるようにする必要があります。これを行うには、次のいずれかの方法を実行します。

1. E1.設計上、すべての IN FIFO がすべてのマイクロフレームに合わせてロードされ、準備されていることを確認します。たとえば、INPPF フィールドが 2 に設定されている場合は、転送ごとに 2 つのバッファがロードされ、USB 用にコミットされていることを確認します。
2. 次のマイクロフレームのデータが不足していることがわかっている場合は、未使用の FIFO のいずれかをイネーブルにして ZLP を送信します。簡単な方法としては、FIFO のバイトカウントレジスタに値 0 をロードするだけです。
3. 外部ソースが USB FIFO をロードする場合、FIFOADR [1:0]で FIFO を選択し、PKTEND ピンをアサートすることによって、未使用の IN FIFO をアーミングして ZLP を送信できます。PKTEND は、部分的に埋められた(または空の)FIFO を USB 送信用にコミットするために使用されます。

ホストドライバは、USB PID シーケンスエラーを正常に処理するかどうかを決められます。クライアント・アプリケーションまで指示が伝わり、デバイスが誤った PID シーケンスを送信する状況を処理するホストドライバを、サイプレスは提供します。AADJ = 1 のときに正しい Data PID シーケンスを自動的に送信するため、FX2LP を使用したトリガーは禁止です。

アイソクロナス OUT エンドポイントを経由する転送のためにレジスタを設定する必要はありません。デバイスファームウェアは、エンドポイントディスクリプタ内のマイクロフレームごとに必要なパケット数のみを報告する必要があります。

5 Cypress SuiteUSB

このアプリケーションノートの記述とテストに必要なフォルダは、添付の zip ファイルに含まれています。より幅広いサポートのために、Cypress SuiteUSB、Visual Studio 用の USB 開発ツールセット、Keil ファームウェアプロジェクトのサンプルを提供しています。Cypress SuiteUSB を使用すると、すべての Cypress USB 2.0 ファミリの.NET Windows アプリケーションを作成できます。このスイートには、開発ツールと互換性のあるサイプレス Windows ドライバ *cyusb.sys* が含まれています。

Cypress SuiteUSB は http://www.cypress.com/?rID=34870 からダウンロードできます。Cypress SuiteUSB のデフォルトのインストールパスは、`C:\Cypress\Cypress Suite USB 3.4.7` です。

6 ストリーミング ファームウェアの例

このノートに含まれるアプリケーションである CyStream は、EZ-USB FX2/FX2LP 開発ボードを使用した USB アイソクロナスストリーミングと BULK 転送をデモします。このコード例は、Windows Streamer アプリケーションのデータストリームを提供し、コンシューマのテストデバイスを実装しています。

CyStream プロジェクトは、ハイスピード(HS)操作用とフルスピード(FS)操作用の 2 つの異なるコンフィグレーションディスクリプタを持つ単一のインターフェース(番号 0)を作成します。表 3 は 7 つの HS オルタネート設定を示し、表 4 は 4 つの FS オルタネート設定を示します。

表 3. FX2LP HS コンフィギュレーション ディスクリプタのインタフェース 0 のオルタネート設定

オルタネート設定	エンドポイント数	エンドポイント番号と方向(Type)	最大パケットサイズ(Bytes)
0	1	2 IN (Bulk)	512
1	1	2 OUT (Bulk)	512
2	2	2 IN (Bulk)	512
		6 OUT (Bulk)	512
3	1	2 IN (Isochronous)	3x1024
4	1	2 OUT (Isochronous)	3x1024
5	1	2 IN (Isochronous)	1x1024
6	2	2 IN (Isochronous)	1x1024
		6 OUT (Isochronous)	1x1024

表 4. FX2LP FS コンフィギュレーション ディスクリプタのインタフェース 0 のオルタネート設定

オルタネート設定	エンドポイント数	エンドポイント番号と方向(Type)	最大パケットサイズ(Bytes)
0	1	2 IN (Bulk)	64
1	1	2 OUT (Bulk)	64
2	1	2 IN (Isochronous)	1023
3	1	2 OUT (Isochronous)	1023

ホストアプリケーション(Streamer.exe)は、異なる転送パラメータ(オルタネート設定)を選択して、スループットレートを測定できます。達成可能な最大スループット・レートを測定するには、エンドポイント FIFO にバイトを出し入れする際に発生する遅延など、FX2LP 側で遅延が入らないようにすることが重要です。このため、このアプリケーションノートの FX2LP コードは最小のオーバーヘッドで IN および OUT 転送を処理します。

- IN 転送 — IN FIFO が使用可能になると、8051 は 1024 バイトの HS 転送または 1023 バイトの FS 転送のために IN FIFO を単純に再度受信可能にします。初期化中にロードされた FIFO データは毎回再送信されます。
- OUT 転送 — 8051 は OUT FIFO からバイトを転送しません。それはバイトカウントロー(EPxBCL)レジスタに任意のバイトカウントをロードすることによって次の OUT 転送のみを再度準備させます。

注意事項:実際のアプリケーションは FIFO データを管理するため、FX2LP は転送オーバヘッドを最小化するための 2 つのハードウェアメカニズムを提供します。

1. データ転送を、FX2LP エンドポイント FIFO への直接ハードウェア接続された外部コントローラーによって処理させます。これにより、8051 はデータパスの処理に関与しなくなります。FIFO 接続には、8 ビットまたは 16 ビットのデータバス、読み出しとび書き込みのストローブ、および FIFO のステータスフラグが含まれます。
2. FX2LP FIFO は自動モードで動作するように設定できます。自動モードでは、エンドポイント FIFO 内のバイト数がレジスタに設定されたレベルと一致した後、パケットが自動的に USB 転送されます。

このアプリケーション・ノートでは、エンドポイント FIFO の手動モード設定を使用しています。 8051 は、FIFO ステータスの確認とエンドポイント・データのコミットを担当します。

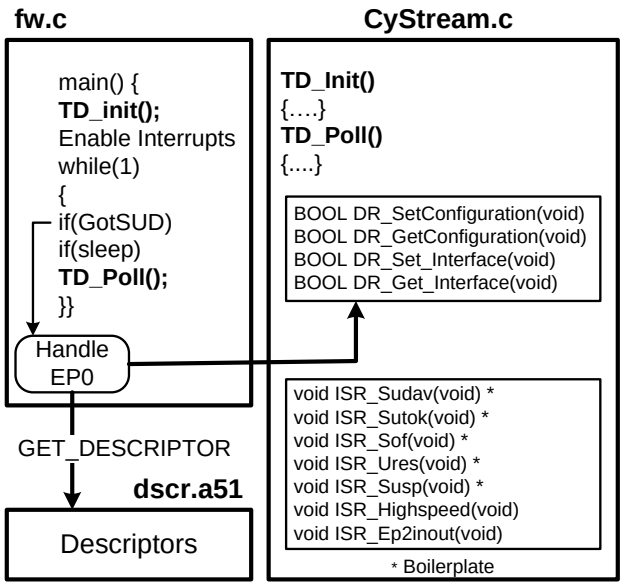
7 Cypress USB フレームワークの使用

このアプリケーションノートに含まれるコード例は、ストリーミングデータの FX2LP エンドポイントを設定する方法を示します。このセクションでは、コードの主要なルーチンについて詳しく説明します。この例には、次のソースファイルが含まれています。

1. Cypress フレームワーク ファイル, *fw.c*
2. ディスクリプタ ファイル, *dscr.a51*
3. CyStream.c ファイル

次の図に示すように、これらのコードモジュールは互いに機能します。この構成は、Cypress ファームウェアフレームワークに基づく FX2LP USB アプリケーションに適用されます。

図 1. CyStream コード モデル



7.1 fw.c

このサイプレスで書かれたファームウェア、フレームワーク は低レベルの USB の詳細を処理します。このファイルを変更する必要はありません。*fw.c* には、プロジェクトの main()関数が含まれています。この関数は、起動時に **TD_Init()**を 1 回呼び出してから、操作中に **TD_Poll()**を繰り返し呼び出します。このエンドレスループは、CONTROL エンドポイント(EP0)のセットアップパケットも処理します。GET_DESCRIPTOR 要求の場合は、dscr.a51 で指定したディスクリプタデータが使用されます。他のホストのリクエストについては、インターフェースのオルタネート設定を変更するなど、さまざまなアクションを処理するためにアプリケーションへ呼び出されます。エンドレスループは USB サスペンドとレジュームイベント(スリープ)も処理します。

7.2 dscr.a51

これは、特定の USB デバイス用のディスクリプタデータを含む 8051 アセンブリ言語モジュールです。このファイルには、ディスクリプタテーブルのデータを次の順序でリストするための.db(define byte)文が含まれています。

1. デバイスディスクリプタ
   a. USB Spec Version (2.0)
   b. Device Standard Class (none)
   c. EP0 MaxPacketSize (64)
   d. Vendor ID = 0x04B4 = Cypress
   e. Product ID = 0x1003 = “EZ-USB Example Streamer Device”.
   f. Manufacturer and Product String Indices
   g. Number of configurations = 1

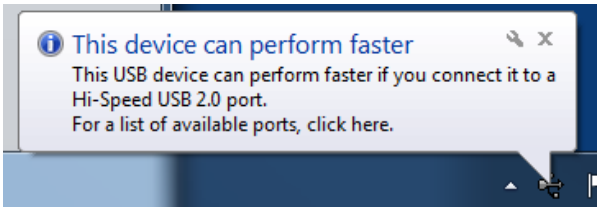
2. Device Qualifier(デバイス修飾子):フルスピードまたはハイスピードで動作できるデバイスの場合、'other' (non-current) スピードを設定します。
3. HS コンフィギュレーション
   a. Number of Interfaces = 1.
   b. Index to select this Configuration = 1
   c. Seven Interface-Endpoint Descriptor pairs that lay out the seven alternate setting parameters in 表 3.
4. FS コンフィギュレーション
   a. Number of Interfaces = 1.
   b. Index to select this Configuration = 1
   c. Four Interface-Endpoint Descriptor pairs that lay out the four alternate setting parameters in 図 4.
5. 2 つの文字列ディスクリプタ

7.2.1 FS/HS の操作について

USB 周辺機器がホストに接続されると、ホストの内部 USB ハブ(または外部にあるハブ)が周辺機器とのネゴシエーションを行って動作速度を決定します。ホストは決して実際の動作速度を選択しません。代わりに、ハブによって報告された速度を使用します。デバイスが接続されたままであれば、ハブが報告した動作速度は決して変化しません。デバイスの取り外しと再接続だけがスピード検出プロセスを再開します。

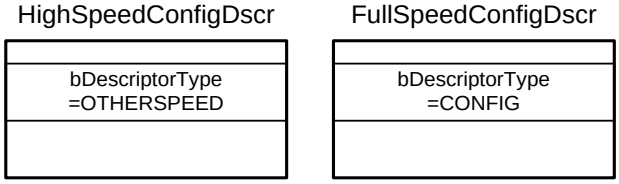
FS または HS で動作できるデバイスの場合、ハブによって報告される速度は「現在の」速度であり、動作可能な速度は他の速度です。2 つのコンフィギュレーションディスクリプタ(各動作速度に 1 つずつ)を持つことで、ホストは、デバイスが別のポートに接続された場合、デバイスが別の速度をサポートしていると判断できます。これにより、HS デバイスを FS ポートに接続すると、Windows は図 2 のメッセージを表示します。

図 2. FS ポートに接続された HS デバイスを検出した Windows



現在の動作速度を検出することと、現在使用されているコンフィギュレーションディスクリプタ(FS または HS)と、選択されていない(他の)速度に対応するものとを指定することは、FX2LP ファームウェアの役割です。ファームウェアは、正しい *bDescriptorType* 値、CONFIG、または OTHERSPEED を各コンフィギュレーションディスクリプタの 2 番目のバイトに挿入することによって、これを行います。

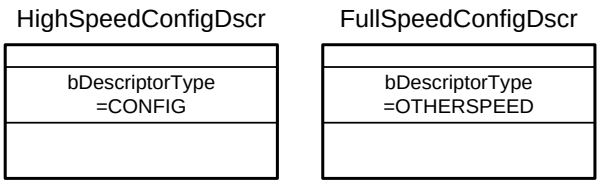
図 3. リセット後のディスクリプタ タイプ



USB バス RESET に続けて、デバイスは FS 動作に戻ります。したがって、USB リセット割り込みサービスルーチンでは、ファームウェアがディスクリプタをマークします(図 3)。エニュメレーション中に使用するために、各ディスクリプタへのポインタ(*pConfigDscr* と *pOtherConfigDscr*)も設定します。デバイスがフルスピードポートに接続されている場合、この指定は残ります。

FX2LP は、USB コアがハイスピード動作のネゴシエーションに成功するたびに割り込み（ISR_Highspeed）を受け取ります。この ISR では、図 4 に示すように、ファームウェアが指定を入れ替え、それに応じてディスクリプタポインタを設定します。

図 4. HS 検出後のディスクリプタ タイプ



7.2.2 wMaxPacketSize について

フォーマットが明らかでないアイテムの 1 つは、高帯域幅のエンドポイント用の *wMaxPacketSize* エントリです。3×1024 または 3072 バイトの最大パケットサイズを持つ場合、マイクロフレームごとに 3 つの 1024 バイトパケットを処理できるエンドポイントと報告します。ただし、物理パケットの最大サイズは 1024 バイトであるため、これは USB 2.0 仕様では定義されていません。この仕様では、ビット 10-0 が最大サイズ 1024 の *maxPacketSize* として、ビット 12-11 がマイクロフレームあたりの追加パケットとして：00 はなし、01 は 1、10 は 2 定義されます。したがって、表 5 の値を使用して、高帯域幅のエンドポイントディスクリプタを実装します。下位バイトファーストです。

表 5. 高帯域幅エンドポイントの許容 maxPacketSize 値

マイクロフレームごとのパケット	wMaxPacketSize
1	0x0400
2	0x0C00
3	0x1400

7.3 CyStream.c

これはストリーミングアプリケーションです。 アプリケーションコードは、USB フレームワークを使ってこのモジュールに書き込まれます。

1. ご使用のアプリケーションに合わせて TD_Init()と TD_Poll 関数を記述します。
2. Fw.c は、エンドポイント 0 を介してさまざまなデバイス要求を入力するときに、コード内の特定の名前が付いた関数を呼び出します。サイプレスコードテンプレート（peripheral.c）は、すべての関数スタブを含むコードスケルトンを提供することによって、関数の作成作業を削減します。ご使用のアプリケーションが使用する関数だけを記述します。
3. アプリケーションが使用する割り込みを処理する割り込みサービスルーチンを提供します。 これらの大部分（アスタリスクで示されている）は、単純な確認のハウスキーピングです。 それ以上のアクションが必要ない場合は、変更する必要はありません。

このセクションの残りの部分では、CyStream.c コードについて説明します。

7.3.1 TD_Init()

初期化プログラムは、以下を実行します。

1. IFCONFIG レジスタを 0x40 に設定して 8051 の 48MHz クロックレートを選択し、I/O ピンのグループをポート（GPIO）ピンとして設定します（これらのピンはスレーブ FIFO または GPIF インタフェースピンとしても機能します）。
2. 4 つの開発ボードの LED を消灯します。ボードの LED は、あらかじめ定義されたメモリ位置を読み取ることによってオン/オフされます。
3. EP2 を有効にし、他のすべてを無効にします。その後、ホストが EP6-OUT を使用するオルタネート設定に変更すると、コードは設定が変更された時点でこのエンドポイントを有効にします。
4. フレーム開始（SOF）割り込みを有効にします。SOF ISR は LED 点滅タイミングを処理します。

5. EP2-IN FIFO を 1024 データバイトで埋めて、最初の IN 転送に使用します。FIFO に特定のデータを持たせることは、USB バスアナライザでパケットを表示するのに便利ですが、データはこのアプリケーションには関係ありません。したがって、この手順はオプションです。

7.3.2 TD_Poll()

USB フレームワークが無限ループで呼び出す TD_Poll()はとても簡単です。エンドポイント FIFO がいつ再度有効化する必要があるかを判断するために FIFO フラグをチェックする必要があります。これは 2 つの同様のコードセクションで行います。1 つはハイスピード用、もう 1 つはフルスピード操作用です。各セクションでは次のことが行われます。

1. **EP2-IN FIFO が FULL 状態でないことを確認します。**これは、ホストが FIFO IN データを正常に読み取ったことを示し、FX2LP ロジックが FIFO 制御を 8051 に渡したことを示します。フルでは無い場合:
   a. IN LED をオンにし、SOF 割り込みサービスルーチン(ISR)の時定数(inblink)を設定してオフにします。

   現在のオルタネート設定と一致するサイズの EP2-IN バイトカウントレジスタをロードして、次の IN 転送を再度準備します。これらの値は、表 3(HS)および表 4(FS)の一番右の列に対応します。
2. **EP2-OUT FIFO の EMPTY 状態をチェックします。** これは、ホストが OUT データを EP2-OUT FIFO に正常に送信し、FX2LP ロジックが FIFO 制御を 8051 に渡したことを示します。エンプティではない場合は下記のとおりです。
   a. OUT LED を点灯し、SOF ISR の時定数(アウトブリンク)を設定して消灯します。
   b. EP2BCL レジスタにバイトカウントをロードすることによって、次の OUT 転送を再度準備します。実際のバイトカウント値は関係ありませんが、SKIP ビットを設定するにはビット 7 を設定する必要があります。 SKIP ビットは、ちょうど受信されたパケットが無視されることを可能にし、そのバッファはすぐに次の OUT パケットのために利用可能にされます。
3. **(HS のみ)EP6-OUT FIFO が空でないことを確認します。**これは、ホストが OUT データを EP6-OUT FIFO に正常に送信し、FX2LP ロジックが FIFO 制御を 8051 に渡したことを示します。空でない場合、ステップ 2a および 2b を実行しますが、代わりに EP6BCL レジスタを使用します。

7.3.3 デバイス リクエストの処理と IRQs

ほとんどの単純なデバイス要求は、ローカル変数をコンフィギュレーションインデックスに設定する DR_SetConfiguration や、インデックスを返す DR_GetConfiguration など、わかりやすいものです。単純な ISR は、IRQ フラグをクリアする以外の何もしません。

7.3.4 CyStream.c 関数

CyStream.c は、4 つの関数で動作します。

- ISR_Ures
- ISR_Highspeed
- DR_Set_Interface
- DR_SOF

7.3.4.1 ISR_Ures

この ISR は、USB バス RESET の完了時にトリガします。フルスピードモードではハイスピードデバイスがリセットから外れるので、ISR は FS コンフィギュレーションディスクリプタをメインに設定し、HS コンフィギュレーションディスクリプタを OTHERSPEED として設定します(図 3)。その後、ISR は HS LED(EVKIT ボード上の D2)をオフにし、アクティビティ LED D5 に 'blinkmask'を使用して 2 秒の時間を設定します。

7.3.4.2 ISR_Highspeed

この ISR は、FX2LP がハイスピードモードに入るとトリガします。コンフィギュレーションディスクリプタがスワップされ(図 4)、ハイスピード LED2 がオンになり、デフォルトの EP2-IN が初期化されます。変数 'blinkmask'は、高速で 1 秒の D2 ブリンク期間に設定されます。

7.3.4.3 DR_SetInterface

USB ホストは Set_Interface 要求を使用して、インターフェイスのオルタネート設定を変更します。この設計では 1 つのインタフェースしか使用されないため、インタフェースインデックスのチェックは不要です。コードは、要求されたオルタネート設定を表す SETUP パケット(SETUPDAT [2])の 3 番目のバイトのみを取り出す必要があります。

2 つのスイッチステートメントは、ハイスピード動作のための表 3 パラメータとフルスピード動作のための表 4 パラメータを使用して、オルタネートセッティングに従ってエンドポイントをコンフィギュレーションします。case 文は次のように処理します。

- 設定で使用されているエンドポイントのみを有効にする
- IN エンドポイントの場合、失効したデータをフラッシュするために FIFOS をリセットする BULK IN エンドポイントの場合、エンドポイントのデータ・トグルを 0 にリセットする
- OUT エンドポイントの場合、必要な数の FIFO を準備する
- 7 セグメント表示を更新して、新しいオルタネート設定 0-6 を表示する

7.3.4.4 DR_SOF

SOF 割り込みは、LED 点滅速度などのタイミングイベントに役立ちます。SOF 割り込みは、フルスピードでは 1mSec、ハイスピードでは 125 マイクロ秒ごとに発生します。SOF ISR は時間をカウントし、必要に応じてアクティビティ LED D5、IN LED D4、および OUT LED D3 をオフにします。

8 CyStreamer.hex の実行

すべての Cypress EZ-USB チップと同様に、FX2LP はプログラム記憶に RAM を使用します。このため、さまざまなソースからプログラムコードを RAM に移動できる USB ローダーを電源投入で起動します。これは接続された EEPROM を使用することも、PC からコードをロードできる USB デバイスとして使用することもできます。このセクションの手順は、*CyStream.hex* ファイルを USB 経由でダウンロードして実行する方法を示しています。

1. 表 6 に従って FX2LP ボードジャンパを準備します。

表 6. EZ-USB FX2LP ボード ジャンパー設定

JP	状態	目的
6, 7	OUT	Memory config for development
2	IN	Power the board from its USB connector
1, 5, 10	IN	Local 3.3-V source
Block 3	IN	All 4 jumpers IN — activate 4 LEDs D2-D5
8	Either	Not used (for Remote Wakeup testing)

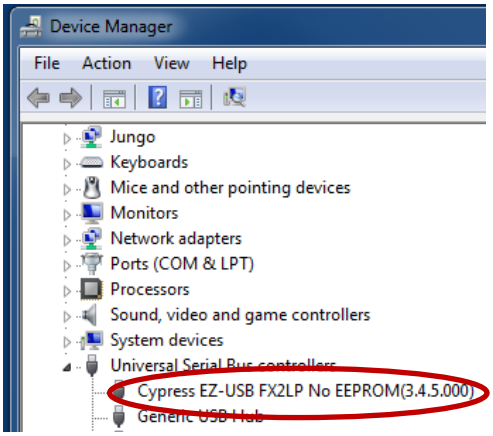
2. ボードの左下隅にある EEPROM ENABLE スライドスイッチ(SW2)を **NO EEPROM**(下)の位置に移動します。これにより、FX2LP チップは、USB デバイスをロードするコードとしてエニュメレーションすることができます。もう一方のスライドスイッチ(SW1、EEPROM SELECT)はどちらの位置にもあります。
3. FX2LP ボードを PC の USB ポートに接続します。初めての場合は、USB ドライバのインストールを促すポップアップメッセージが表示されます。下記の場所に移動します。

```
C:\Cypress\USB\
CY3684_EZ-USB_FX2LP_DVK\
1.0\Drivers\cyusbfx1_fx2lp
```

お使いの Windows OS に対応するフォルダを選択します。

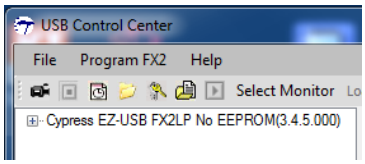
デバイスマネージャーを表示して、ドライバーのインストールが正常に完了したことを確認できます(次の図を参照)。

図 5. 正しくインストールされたドライバ

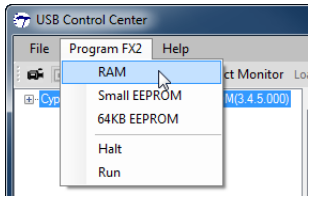


4. AN4053 フォルダにある USB コントロールセンターを起動します。実行ファイルは bin フォルダにあります。
5. 図 6 に示すように、FX2LP ボードが左パネルに表示されます。

図 6. USB コントロールセンターに表示されるデバイス

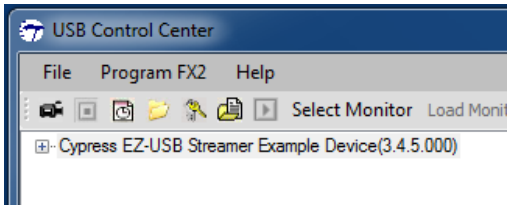


6. Cypress デバイスの項目をクリックして強調表示し、**Program FX2> RAM** を選択します。



7. `AN4053\FX2LP streamer source code\firmwareCyStreamer` に移動し、*CyStream.hex* ファイルをダブルクリックします。
8. USB コントロールセンターの左パネルを見ます。PC サウンドデバイスが有効になっている場合は、すぐに 'USB 接続'サウンドが続き、新しい USB デバイスが表示されます。

図 7. 新しい USB デバイス



これは Cypress ReNumeration プロセスです。図 6 に示す初期デバイスは、FX2LP デバイスにハードワイヤードされた USB ローダーです。コードが FX2LP RAM にロードされた後、FX2LP は図 7 の新しい USB デバイスと電気的に接続を解除し、再接続します。

CyStream.hex の新しいバージョンを作成し、FX2LP ボードにダウンロードしたい場合は、まずボードの RESET ボタンを押す必要があります。これにより、ボードが USB から切り離されます。RESET ボタンを離すと、FX2LP は USB ローダーとして PC に再接続します。Windows に切断再接続のイベントを認識させるのに十分な時間を与えるには、RESET ボタンを約 2 秒間押し続けることをお勧めします。

ストリーマーサンプルデバイスは、以下の手順によって確認されて実行を開始します。

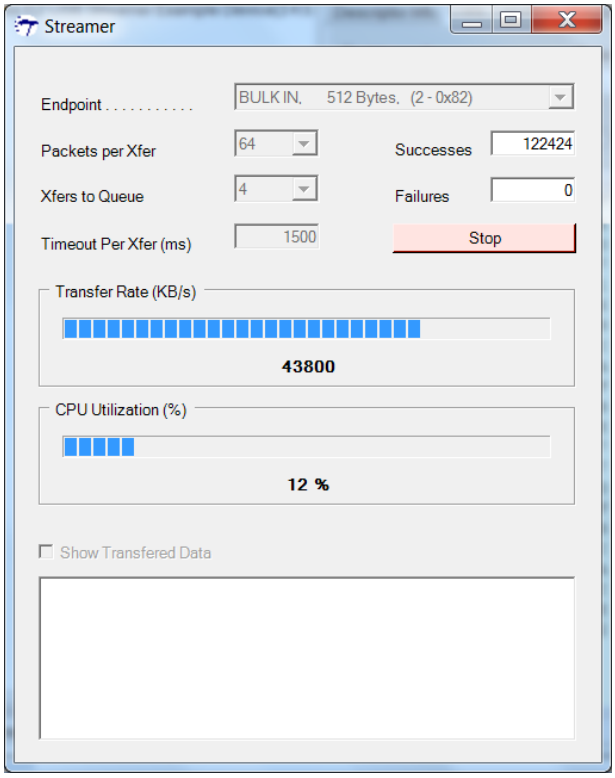
- 7 セグメントの読み取り値は、現在のオルタネート設定を示します。デフォルトの(パワーオン)設定は 0 です。
- LED D5 は、HS 接続の場合は 1 秒に 1 回、FS 接続の場合は 2 秒に 1 回点滅します。
- IN FIFO が作動すると LED D4 が点灯します。
- OUT FIFO が準備されると LED D3 がオンになります。
- LED D2 は、HS ではオン、FS 動作ではオフです。

9 パフォーマンス分析

`AN4053\VS_Streamer` の Streamer.exe は、USB ストリーミングアプリケーションのパフォーマンスを評価するために使用されます。このコードでは *CyUsb.sys*(AN4053\Driver)ドライバを使用して、CYStream ファームウェアを実行している FX2LP 開発ボード(CY3684)とインターフェイスを持ちます。

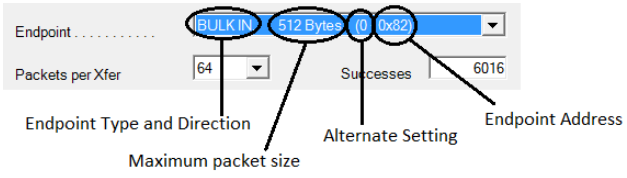
次に、Streamer ホストアプリケーションウィンドウのスクリーンキャプチャと、図 8 のウィンドウに表示されるフィールドの説明を示します。

図 8. Streamer.exe アプリケーションウィンドウ



エンドポイント:このドロップダウンリストは、オルタネート設定を示します。ハイスピードの場合は 0～6、フルスピードの場合は 0～3 です。これにより、異なる転送タイプとエンドポイントバッファーの数を選択できます。

図 9. エンドポイント ドロップダウンリストの説明



このセクションには、[エンドポイント]ドロップダウンリストに表示されるテキストの説明が表示されます。ドロップダウンリスト（図 9 を参照）の最初のものを取ってみましょう。設定：BULK IN, 512Bytes, (0-0x82)

BULK IN – このフィールドには、エンドポイントのタイプと方向が示されます。この例では BULK エンドポイントを使用して、IN 転送を実行できます。

512Bytes – エンドポイントの最大パケットサイズ。この場合、512 バイトです。

0 – このフィールドには、オルタネートセッティング番号が含まれます。この場合、オルタネートセッティング – 0 です。

0x82 – エンドポイントアドレス。この場合、エンドポイント番号 0x02 および方向 IN です。

Packets per Xfer ： 転送とは、1 つのデータセットのパケットの集合です。転送ごとのパケット数が増えると、USB オーバーヘッドが減少し、データ転送速度が向上します。

Xfers to Queue：この設定は、複数の転送を開始し、それらをタスクキューに追加するために使います。これにより、ホストアプリケーション側での連続転送間の待ち時間が短縮されます。したがって、より多くの転送をキューイングすると、より高いデータ転送速度が得られます。

Successes：ストリーミングテスト中に正常に転送されたパケットの総数が表示され、インクリメントされます。

Failures：バッファの転送でエラーが報告されるたびにインクリメントされます。可能性のある障害メカニズムは、ISO IN PID シーケンシングを適切に処理しなかった FX2（LP ではない）アプリケーションです。

Transfer Rate：選択されたエンドポイントを介した USB バスおよび EZ-USB FX2LP の現在のスループット性能を、リアルタイムで更新します

CPU Utilization：USB 経由でストリーミングする際に、コンピュータの CPU 使用率を視覚的に表示します。

このアプリケーションは、Windows®7 を実行する Intel IHC9 USB 2.0 ホストコントローラを搭載したシステムで実行されました。その結果、BULK 転送で達成できる平均スループットは 43.8 MBps で、ISO 転送で達成できる最大スループットは 24 MBps でした。FX2LP に外部ペリフェラルを接続することなく、上記の性能番号をテストすることに注意してください。データは FX2LP 内部で生成されます。外部ペリフェラルからデータを生成している場合、これらのパフォーマンス番号は、その外部ペリフェラルがデータをどのくらい高速に転送できるかにも依存します。

特定のシステムでは、BULK 転送速度が ISO 転送速度を超える可能性があります。これは、軽負荷のバスでは、ストリーマデバイスが利用可能な USB BULK 帯域幅割り当てのほとんどまたはすべてを使用するためです。ただし、ビデオを再生する USB ディスクドライブなど、他の USB デバイスが起動すると、ISO エンドポイントがネゴシエートされ帯域が保証されると、利用可能な BULK 帯域幅は減少します。

10 FX2LP と イメージセンサのインターフェース

FX2LP は、ベンダークラスドライバを使用して画像をストリーミングするためにイメージセンサにインターフェイスすることができます。KBA95736 - Interfacing FX2LP™ with Image Sensor.を参照してください。

11 参考文献

USB 2.0 Specification – (http://www.usb.org/developers/docs/usb_20_070113.zip)

FX2LP Technical Reference Manual(TRM)

12 概要

このアプリケーションノートでは、オーディオおよびビデオストリーミングなどのアプリケーションのストリーミングデータをサポートする USB 高帯域幅配信メカニズムについて説明します。関連するファームウェアプロジェクト "CYStream.Uv2"は、ハイスピード USB ISO 転送用に FX2LP をプログラミングする方法を示しています。帯域幅の比較のために、一組の BULK 転送も実装されています。また、Cypress USB フレームワークを使用して代替 USB 設定を実装し、ホストが異なる転送レートを選択できるようにする方法も示します。さまざまな転送タイプを選択し、転送レートを測定するために、付属の PC アプリケーション「Streamer.exe」が提供されています。

About the Author

Name: Rama Sai Krishna Vakkantula.

Title: Applications Engineer Staff

13 改訂履歴

文書タイトル: AN4053 - EZ-USB® FX2™および FX2LP™ のアイソクロナスまたはバルクエンドポイントを介したデータのストリーミング

文書番号: 001-75997

版	ECN	変更者	提出日	変更点
**	3520393	NITA	02/28/2012	英語版データシート001-15289を日本語に翻訳
*A	4635826	JNAN	01/22/2015	No Change ECN
*B	6118779	HIKA	04/02/2018	これは英語版 001-15289 Rev. *I を翻訳した日本語版 001-75997 Rev. *B です。

セールス, ソリューションおよび法律情報

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	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, 2008-2018. 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。