



EZ ロード カスタム USB ファームウェアロード・ドライバ

概要

サイプレスの EZ-USB、EZ-USB FXTM、および EZ-USB FX2TM マイクロコントローラファミリの特長は、デバイスの性質をファームウェアのダウンロードおよび ReNumerationTM を介して変更できることである。EZ-USB ベースの典型的なデバイスは、固有のベンダ ID (VID) とプロダクト ID (PID) を記憶する不揮発性記憶域を EEPROM 内に有している。この VID/PID の組合せは、ホストシステム上の特定のデバイスドライバと結合している。また、この VID/PID の組合せは、ファームウェアをデバイスにダウンロードする機能しか持っていないドライバに結合させることもできる。本アプリケーションノートでは、このドライバについて説明する。この EZ ロードドライバを使用すると、周辺装置の設計者は、特定のデバイスに対するファームウェアのダウンロードが可能なカスタムドライバを作成できる。

開始

設計者は、開発中に、EZ-USB コントロールパネルを使用し、USB ケーブルを介して、Intel 16 進ファイルをダウンロードすることができる。あるいは Keil ツールを使用することにより、シリアルケーブルを介して設計者のファームウェアをダウンロードすることもできる。これらの方法は開発用としては優れているが、設計者は最終的には、ファームウェアのダウンロードおよび ReNumeration プロセスを自動化したいはずである。そこで、EZ ロードドライバが登場する。

サイプレスセミコンダクタ EZ-USB Xcelerator 開発キットを備えた EZ ロードドライバの場合、特定のデバイスをサポートするために必要な改変がほとんどなく、したがって Windows ドライバプログラミングに関する広範囲の知識は不要であるが、Windows プラグ&プレイ、Windows INF ファイル、Windows レジストリおよび USB については、設計者は精通していなければならない。

EZ ロードドライバはデバイスドライバであるから、Windows 98/Windows 2000 用のデバイスドライバキット (DDK) が必要である。設計者は、EZ ロードドライバの個別化を図る前に、DDK を使用してデバイスドライバを正しく構築できることを検証しなければならない。本アプリケーションノートの後半を読めば、設計者は、ステップバイステップ手順に従ってデバイスドライバを正しく構築できる。

EZ ロードドライバファイル

EZ ロードドライバ構築に必要なソースファイルは、EZ-USB Xcelerator 開発キットに含まれており、インストールした開発キットツールのディレクトリ C:\Cypress\USB\Drivers\ezloader 内にある。

1. ezloader.c- これはメイン EZ ロードソースファイルで、ファームウェアのダウンロードを実行するためのコードと共に、DriverEntry()DriverEntry()およびその他の標準 USB ドライバエントリポイントが含まれている。
2. ezloader.h- EZ ロードドライバのヘッダファイル。
3. firmware.c- デバイスファームウェア用 Intel 16 進レコード・アレイが入っている。後述のステップバイステップ手順の中に、設計者のファームウェアを Intel 16 進レコードフォーマットから、このファイル内に置く C コードに変換するための hex2c.exe ユーティリティがある。EZ ロードドライバをコンパイルすると、ドライバイメージ中に設計者のファームウェアが含まれる。
4. loader.c- 外部 RAM へのダウンロードを可能にするデバイスファームウェア用 Intel 16 進レコード・アレイが入っている。設計者のファームウェアが外部 RAM 中へ拡張する場合、外部 RAM へのダウンロードを可能にするために、このファームウェアが最初に EZ-USB デバイスにダウンロードされる。
5. ezloader.rc- リソースファイル。ドライバのバージョン情報が含まれている。
6. makefile- DDK ビルドユーティリティに必要なファイル。
7. sources- DDK ビルドユーティリティに必要なファイル。設計者は、ステップバイステップ例の中でこのファイルをカスタマイズし、固有の名称を有するドライバを作成できる。

必要なツール

Microsoft Visual C++ (5.0/6.0)

DDK をインストールする前に MicrosoftTM Visual C++TM (5.0/6.0) をインストールしなければならない。Microsoft C コンパイラは、DDK ビルドユーティリティによって自動的に呼び出される。

Windows 98DDK / Windows 2000DDK

EZ ロードデバイスドライバを構築するには、Windows 98DDK または Windows 2000DDK のいずれかが必要である。DDK は、<http://www.microsoft.com/ddk/> の Microsoft ウェブサイトを介してダウンロードできる。Windows 98DDK を使用してコンパイルされたドライバは、Windows 2000 システムでも動作し、同様に Windows 2000DDK を使用してコンパイルされたドライバは、Windows 98 システムでも動作する。

hex2c ユーティリティ - Intel 16 進レコードから C 言語への翻訳プログラム

サイプレスセミコンダクタが提供するこのユーティリティは、Intel 16 進レコードの入力ファイルをコンパイルでき、かつ、EZ ロードドライバにリンクできる C コードに変換するユーティリティである。このユーティリティは、Win32 コンソールアプリケーションであり、コマンドプロンプトから、以下のように入力して使用する。

```
hex2c <intel_hexfile_name> <c_filename> <var_name>
```

このユーティリティは、INTEL_HEX_RECORD 構造 (ezloader.h の中で定義される) のアレイを含んだ C ファイルを作成する。intel_hexfile_name は入力ファイルで、設計者が設計者のファームウェアを Keil ツール (すなわち widget.hex) を使用してコンパイルした時点で作成される 16 進ファイル。

c_filename は、生成 / 重ね書きされる出力ファイル名。

var_name はオプションで、C 出力ファイル中のアレイの名称。デフォルトにより、var_name=ファームウェアとなる。

hex2c ユーティリティは、C:\Cypress\USB\bin ディレクトリにある。このユーティリティのソースコードは、C:\Cypress\USB\Util\Hex2c ディレクトリ内に、EZ-USB Xcelerator キットを持つ。

EZ ロードデバイスドライバの構築

設計者は、EZ ロードドライバを改変する前に、設計者のカスタムロードドライバを構築する新しいディレクトリを作成しなければならない。これにより、開発キットにある ezmon.sys ドライバに対する設計者による不注意な改変が防止される。新しいディレクトリを作成し、その中に EZ ロードソースファイルをコピーすれば、DDK ビルドユーティリティを使用して ezloader ドライバをコンパイルできる。

「ソース」ファイルがドライバの構築方法を DDK ビルドユーティリティに通知する。DDK ビルドユーティリティは、ドライバおよびドライバ出力ファイルの名称からなるソースファイルを指定する。指定したソースファ

イルが提供されると、「ソース」ファイルは、「ezloader.sys」と呼ばれるドライバを作成する。

完成したドライバは、構築環境に応じて、\lib\i386\free またはチェック済みディレクトリ中に置く。ドライバ構築に先立って、これらのディレクトリが作成済みであることを確認しなければならない。ディレクトリが作成されていないと、ビルドユーティリティは、エラーをレポートする。EZ ロードドライバをカスタマイズする場合、ドライバの出力ファイル名を、ezloader.sys 以外の別の名称に変更しなければならない。この名称の変更は、「ソース」ファイル中の "TARGETNAME=" 欄を、新しい固有の名称に変更することによって行う。設計者は、ステップバイステップ例の中で「ソース」ファイルを変更できる。

ステップバイステップ例

このステップバイステップ例は、仮想 USB デバイス、サイプレスウィジェットを使用した例である。

ReNumeration を使用するデバイスには、2 つの PID が必要となる。ここでは、プリファームウェア・ウィジェットの PID として 0x1004 を使用し、完全に機能するデバイスの PID として 0x1005 を使用する。後者の PID は、ウィジェットファームウェア中に埋め込まれており、通常、dscr.a51 ファイル内で指定される。

サイプレスのベンダ ID (VID) は、0x0547 である。ここで、ウィジェットはプロダクト ID (PID)、0x1004 および 0x1005 を使用する。設計者は、開発目的としてのみ、サイプレス VID を使用できるが、デバイスの製造準備完了後は、固有の VID を入手しなければならない。設計者が製造するデバイスのためのベンダ ID の獲得に関する情報は、www.usb.org にある。

ウィジェット用デバイスドライバは、EZ-USB Xcelerator 開発キットを備えた EZ-USB 汎用デバイスドライバ (ezusb.sys) である。このドライバは、VID = 0x0547、PID = 0x1005 に結合されている。

このステップバイステップ例によって作成されるドライバは、wdgtldr.sys (ウィジェットローダ) と呼ばれる新しいドライバである。このドライバは、VID = 0x0547、PID = 0x1004 に結合されている。

ファームウェアが完成し、設計者が、完成したファームウェアをファームウェアローダ・ドライバに統合することによってファームウェアダウンロード・プロセスを自動化したい場合、以下のステップに従う。この例では、キール 8051 開発ツールを使用して作成された、C:\Cypress\USB\Examples\EzUsb\ep_pair 内にある ep_pair.hex ファームウェアを使用している。

1. ウィジェットと呼ぶ新しいディレクトリを作成し、その中に、C:\Cypress\USB\Drivers\ezloader ディレクトリのすべての内容（ファイルおよびサブディレクトリ）をコピーする。
2. 新しく作成したディレクトリ中のテキストエディタを使用して、「ソース」ファイルを編集し、TARGETNAME=ezloader の行を TARGETNAME=wdgtldr に変更する（図 1）。ローダ名として使用できる文字数は、8 文字以内。設計者が、Windows 2000DDK を使用してファームウェアローダ・ドライバを構築している場合、同時に、行 TARGETLIBS=\$(DDK_LIB_PATH)\usbd.lib を非注釈にしなければならないが、これは#を削除することによって非注釈にすることができる（図 2）。

```
*****
;*
;*File:Sources
;*build
;*information file for the WDM build.exe
utility
;*
;*Date: February 12, 1999
;*Version:1.00.00
;*
;*Notes:
;*Copyright (c) 1997,1998,1999 Anchor Chips,
;*Inc. May not be reproduced without
;*permission. See the license agreement for
;*more details.
;*
;*Environment:
;* WDM Build utility information file
;*
;*Revision History:
;*
*****
TARGETNAME=wdgtldr
TARGETTYPE=DRIVER
TARGETPATH=.\LIB
DRIVERTYPE=WDM
```

図 1. ウィジェットローダ「ソース」ファイル中の
TARGETNAME

```
# to build this driver using the Windows 2000 DDK, uncomment the
following line:
TARGETLIBS=$(DDK_LIB_PATH)\usbd.lib
```

図 2. Windows2000DDK 用に改変された
「ソース」ファイル

3. Keil ツールを使用して、C:\Cypress\USB\Examples\EzUsb\ep_pair 内にある ep_pair プロジェクトファイルを開く。図 3 のように、dscr.a51 ファイル内の PID を 1005 に変更し、次にプロジェクトを再コンパイルする。これらのビットは、後述のように、リトルエンディアンの順になっていることに注意。ep_pair.hex ファイルのコピーを、ディレクトリ C:\Cypress\USB\Bin 内に置く。これは必須ではないが、これにより、次ステップにおいてファイルへのフルパスの入力が不要になる。他のオプションは、hex2c.exe のコピーを、設計者の ep_pair.hex ファームウェアディレクトリ中に置くことである。
4. 「hex2c.exe」ユーティリティを使用して、Intel 16 進レコードフォーマット・ファームウェアを、ローダドライバにリンクできる C コードに変換する。コマンドプロンプトウィンドウを開き、C:\Cypress\USB\Bin へ移動するか、あるいは設計者が hex2c を C:\Cypress\USB\Bin にコピーしている場合は、設計者のファームウェアディレクトリへ移動する。コマンドプロンプトに、hex2c ep_pair.hex ep_pair.c と入力する（図 4）。これにより、入力として ep_pair.hex ファイルを取り、ep_pair.c が生成される。この新しい C ファイルには、ファームウェアと呼ばれる INTEL_HEX_RECORD タイプの構造の大きなアレイが含まれている。図 5 参照。
5. ep_pair.c のコピーを、作成済みのウィジェットディレクトリに置く。firmware.c ファイルを開いて編集し、そのファームウェアアレイを ep_pair.c からのファームウェアアレイに置換する（図 4）。設計者は、firmware.c ファイルを修正後、修正ファイルを保存し、ディレクトリから ep_pair.c ファイルを削除する。

```
-----
;; Global Variables
;;
-----
;; Note: This segment must be located in on-part memory.
rseg DSCR ;; locate the descriptor table anywhere below 8K
DeviceDscr:dbdeviceDscrEnd-DeviceDscr;; Descriptor length
db DSCR_DEVICE;; Descriptor type
dw 0001H ;; Specification Version (BCD)
db 00H ;; Device class
db 00H ;; Device sub-class
db 00H ;; Device sub-sub-class
db 64 ;; Maximum packet size
dw 4705H ;; Vendor ID
dw 0510H ;; Product ID - changed for EZ-Loader example ID
dw 0100H ;; Product version ID
db 0 ;; Manufacturer string index
db 0 ;; Product string index
db 0 ;; Serial number string index
db 1 ;; Number of configurations

deviceDscrEnd:
```

図 3. dscr.a51 内の VID を変更する

```

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\>cd cypress

C:\Cypress>cd usb

C:\Cypress\USB>cd bin

C:\Cypress\USB\Bin>hex2c ep_pair.hex ep_pair.c
hex2c 1.0
Read 6201 bytes
Processed 173 Records

C:\Cypress\USB\Bin>_

```

図 4. Hex2c ユーティリティを使用して
ep_pair.hex を変換する

6. 設計者は、この段階でファームウェアローダ・ドライバを構築することができる。Windows [スタート] および [プログラム] メニューを介して設計者の DDK を開く。コマンドプロンプトを開いて、C:\Widget\ezloader へ移動する。コマンドプロンプトでは、build-c と入力する (図 6)。設計者のドライバは、C:\Widget\ezloader\lib\i386 の中にある。wdgtldr.sys ファイルを設計者の Windows ドライバディレクトリ内に置く。
7. 新しいドライバとウィジェットデバイスを関連させるためには、設計者は、INF ファイルを作成しなければならない。この例で使用されているサイプレス ウィジェットのサンプル INF ファイルは、本アプリケーションノートの最後にある。完成した INF ファイルは、設計者のシステムにおいて、他の inf ファイルがあるディレクトリと同じディレクトリの中に置かなければならない。この例は、Windows 2000 システム用に作成されたものであり、Windows 98 とコンパチブルである。INF ファイル作成に関するその他の情報については、
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/setupapi/setupapiovr_6xf7.aspにある。設計者がセクションを INF ファイル内に置く順序は重要ではないが、セクション内の項目の順序は重要である。

```

INTEL_HEX_RECORD firmware[] = {
    16,
    0x4216,
    0,
    {0xe4,0xf5,0x2c,0xf5,0x2b,0xf5,0x2a,0xf5,0x29,0xc2,0xc3,0xc2,0xc0,0xc2,0xc2,0xc2},
    16,
    0x4306,
    0,
    {0x01,0x12,0x46,0x15,0x7e,0x44,0x71,0x50,0x8e,0x08,0x8f,0x09,0x75,0x0a,0x44,0x75},
    16,
    0x4316,
    0,
    {0x0b,0x62,0x75,0x0c,0x44,0x75,0x0d,0x82,0xee,0x54,0xe0,0x70,0x03,0x02,0x44,0x02},
    16,
    0x4326,
    0,
    {0x75,0x2d,0x00,0x75,0x2e,0x80,0x8e,0x2f,0x8f,0x30,0xc3,0x74,0xbc,0x9f,0xff,0x74},
    16,
    0x4336,
    0,
    {0x44,0x9e,0xcf,0x24,0x02,0xcf,0x34,0x00,0xfe,0xe4,0x8f,0x28,0x8e,0x27,0xf5,0x26},
    16,
    0x4346,
    0,
    . . .
}

```

図 5. ep_pair.c INTEL_HEX_RECORD 構造

```

Checked Build Environment

New or updated MSVC detected. Updating DDK environment....

Setting environment for using Microsoft Visual C++ tools.
Starting dirs creation...Completed.

C:\NTDDK>cd..

C:\>cd Widget

C:\Widget>cd ezloader

C:\Widget\ezloader>build -c
BUILD: Object root set to: ==> objchk
BUILD: /i switch ignored
BUILD: Compile and Link for i386
BUILD: Loading c:\NTDDK\build.dat...
BUILD: Computing Include file dependencies:
BUILD: Examining c:\widget\ezloader directory for files to compile.
      c:\widget\ezloader - 4 source files (2,066 lines)
BUILD: Compiling c:\widget\ezloader directory
Compiling - ezloader.rc for i386
Compiling - ezloader.c for i386
Compiling - firmware.c for i386
Compiling - loader.c for i386
BUILD: Linking c:\widget\ezloader directory
Linking Executable - lib\i386\wdgtldr.sys for i386
BUILD: Done

      4 files compiled - 1033 LPS
      1 executable built

C:\Widget\ezloader>_

```

図 6. Windows2000DDK を使用して
wdgtldr を構築する

8. 設計者は、新しいファームウェアドライバを使用する前に、新しいファームウェアドライバと設計者のデバイスを関連付けなければならない。この関連付けは、カスタム USB デバイス用として設計者が作成する INF ファイル中で定義される VID/PID の組合せを、シリアル EEPROM にロードすることによって行う。これは、設計者のデバイスが使用する 2 つの VID/PID の組合せのうちの第 1 の組合せである。EEPROM 内の VID/PID ホストにレポートされると、ホストは、設計者のファームウェアをロードする wdgtdr.sys を呼び出す。シリアル EEPROM に設計者の VID/PID の組合せをロードするには、[EZ-USB コントロールパネル] を開き、目標欄と設計者の対象デバイスが一致していることを確認し、Vend_ax.hex 例をダウンロードする。以下に列記するサイプレス PID を使用してはならない。0080、0081、1002、2122、2125、2126、2131、2136、2225、2226、2235、2236 または 8613。

- **EZ-USB** : Dir 欄を 0 OUT に変更し、かつ、Hex Bytes 欄を B0 47 05 04 10 01 00 に変更し、Vend Req ボタンを押す。次に Dir 欄を 1 IN に変更して Vend Req ボタンを押す、同じ値がレポートバックされることを確認する (図 7)。
- **EZ-USB FX** : Dir 欄を 0 OUT に変更し、かつ、Hex Bytes 欄を B4 47 05 04 10 01 00 に変更し、Vend Req ボタンを押す。次に Dir 欄を 1 IN に変更して Vend Req ボタンを押す、同じ値がレポートバックされることを確認する。
- **EZ-USB FX2** : Dir 欄を 0 OUT に変更し、かつ、Hex Bytes 欄を C0 47 05 04 10 01 00 に変更し、Vend Req ボタンを押す。次に Dir 欄を 1 IN に変更して Vend Req ボタンを押す、同じ値がレポートバックされることを確認する。

```
Vendor Request
0000 B0 47 05 04 10 01 00
Vendor Request
0000 B0 47 05 04 10 01 00
```

図 7. [EZ-USB コントロールパネル] を使用して EEPROM をプログラムする

設計者がデバイスをいったん切離し後、再接続すると、Windows ニューハードウェアウィザードが現れ (図 8)、設計者のデバイスをインストールする。第 1 の VID/PID の組合せにより、ホストは、設計者が作成した wdgtdr.sys ファームウェアドライバを介して、設計者のファームウェアをロードする (図 9)。次に、ステップ 3 で作成された ep_pair.hex 例の中の VID/PID の組合せに対してデバイスが ReEnumerate し、ezusb.sys デバイスドライバをロードする (図 10)。

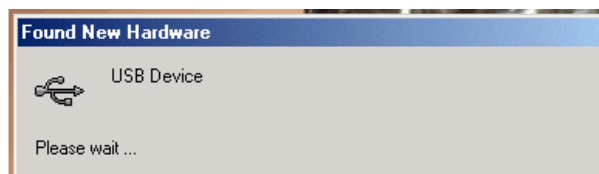


図 8. Windows ニューハードウェアウィザードが設計者のデバイスを検出する

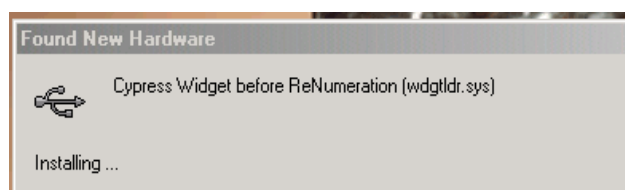


図 9. wdgtdr.sys が設計者のファームウェアをロードする

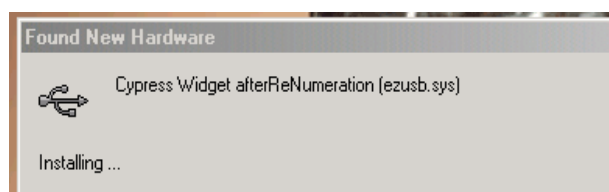


図 10. デバイスが ReEnumerate し、ezusb.sys を使用する

wdgtdr.sys の確認

[EZ-USB コントローパネル] を開いて wdgtdr.sys を確認する。[GetDevice] ボタンを押す (図 11)。[GetPipes] ボタンを押す (図 12)。最後に [bulkloop] ボタンを押すと、ep_pair 例が表示される (図 13)。デバイスは、以上の操作以外のユーザとの対話が不要な構成になっている。

```
Device Descriptor:
bLength: 18
bDescriptorType: 1
bcdUSB: 256
bDeviceClass: 0x0
bDeviceSubClass: 0x0
bDeviceProtocol: 0x0
bMaxPacketSize0: 0x40
idVendor: 0x547
idProduct: 0x1005
bcdDevice: 0x1
iManufacturer: 0x0
iProduct: 0x0
iSerialNumber: 0x0
bNumConfigurations: 0x1
```

図 11. [GetDevice] レポートを押す


```
Pipe: 0 Type: BLK Endpoint: 2 IN MaxPktSize: 0x40
Pipe: 1 Type: BLK Endpoint: 2 OUT MaxPktSize: 0x40
```

図 12. [GetPipes] レポートを押す

```
Write IOCTL passed
0000 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
0010 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
0020 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
0030 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
Read IOCTL passed
0000 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
0010 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
0020 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
0030 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05 05
```

図 13. [BulkLoop] レポートを押す

EZ ロードの更新

ファームウェアおよび/またはデバイスドライバの更新は、本アプリケーションノートの範囲外であるが、インストールシールドプログラムを作成することによって、あるいは本アプリケーションノートの最後にある更新 INF ファイル例を使用することによって可能である。図 14～17 は、Windows デバイスマネージャ内の更新ドライバの選択から、更新済みファームウェアの EZ-USB コントロールパネル表示までの更新プロセスの概要を示したものである（バルクテスト例は、更新済み wdgtdr.sys の作成に使用されたものである）。図に示す更新手順は、Windows 2000 システムのものであるが、この手順は、Windows XP に対しても同様に利用できる。

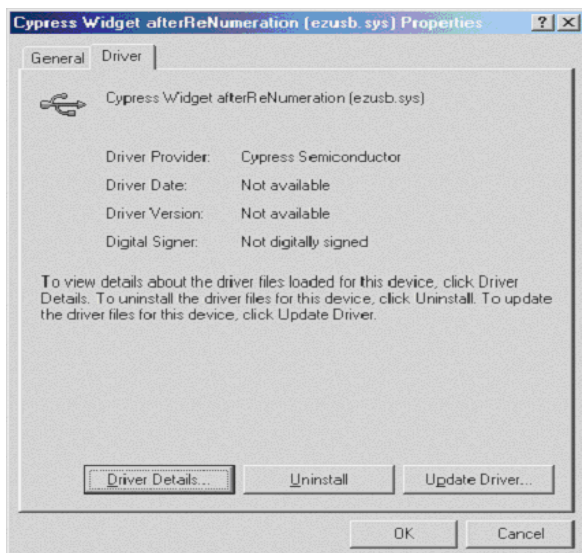


図 14. ファームウェアまたはデバイスドライバを更新する

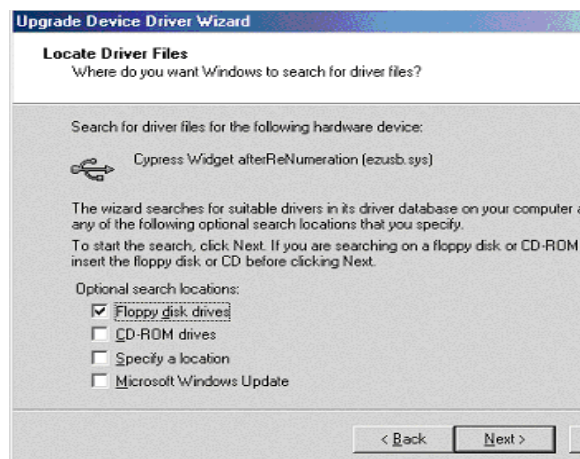


図 15. ファームウェアの更新（続き）

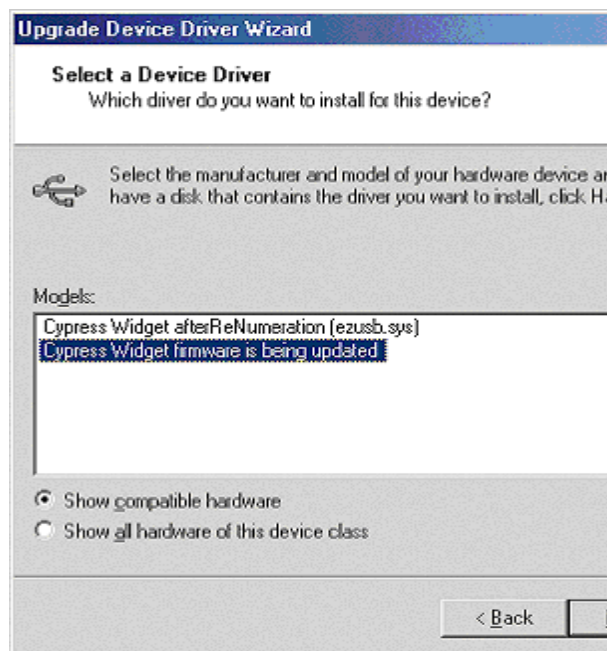


図 16. 更新済みドライバを選択する

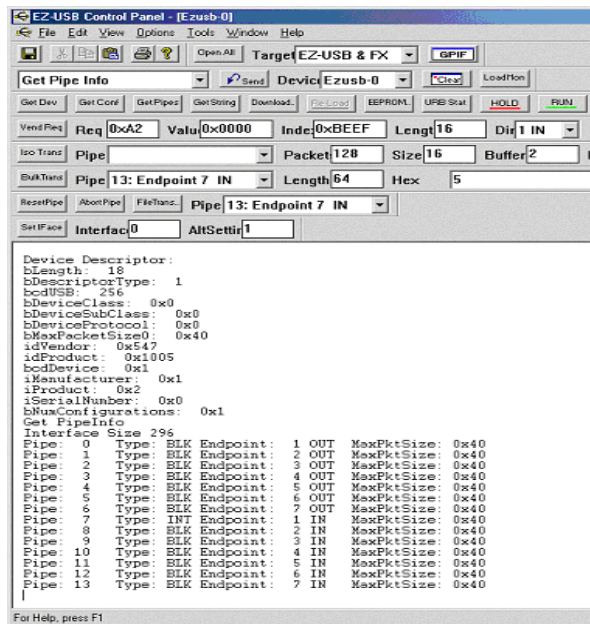


図 17. [コントロールパネル] への更新済み
ファームウェアのロード

まとめ

本アプリケーションノートでは、サイプレス汎用 USB ドライバ、ezusb.sys が使用されている。開発者は、このドライバがドライバ開発の出発点として提供されるものであり、実使用デバイスドライバとして機能することを目的としたものではないことに留意しなければならない。サイプレス推奨のドライバ開発を支援するコンサルタントのリストは、<http://www.cypress.com/support/cypros.cfm>にある。

当社の開発キットにある ep_pair 例およびその他の例に含まれている fw.c ファイルには、記述子およびセットアップ情報の内部チップへのロードを保証するコードが含まれている。ファームウェアの開発に当たり、このコードをコメントとして削除してはならない。

設計者ウィジェット用 VID/PID 組合せの Windows レジストリを、カスタム INF ファイル作成中、または、開発プロセスの間に変更しなければならない場合にクリーンにすることは良い考えである。Windows レジストリをクリーンにする場合、設計者は、Windows/INF ディレクトリ内にある wgdldr PNF ファイルも削除しなければならない。修正を実施する場合には、その前に設計者の Windows レジストリのバックアップコピーを取っておくことが推奨される。

本アプリケーションノートで要約したプロセスに従うことにより、設計者は、ファームウェアのダウンロードを自動実行できる、カスタム USB デバイス用ファームウェアローダ・ドライバを作成できる。また、本アプリケーションノートにより、設計者は、カスタム USB デバイスを更新するためにファームウェアローダ・ドライバ作成方法の 1 つを習得できる。

;WIDGET INF File

```
[Version]
Signature="$CHICAGO$"
Class=USB
provider=%Cypress%
LayoutFile=layout.inf

[Manufacturer]
%Cypress%=Cypress

[Cypress]
; Entry point for the widget before firmware download and renumeration
; This VID/PID combination will call the EZ-Loader driver and download
; your firmware.
%USB\VID_0547&PID_1004.DeviceDesc%=EZUSB.Dev, USB\VID_0547&PID_1004

; Entry point for the widget after firmware download and renumeration
; Your firmware has been download, the device has ReNumerated. Now use
; the Cypress EZ-USB General Purpose Device Driver.
%USB\VID_0547&PID_1005.DeviceDesc%=EZUSB.Dev, USB\VID_0547&PID_1005

[PreCopySection]
HKR,,NoSetupUI,,1

[DestinationDirs]
EZUSB.Files.Ext = 10,System32\Drivers
EZUSB.Files.Inf = 10,INF
EZUSBDEV.Files.Ext = 10,System32\Drivers
EZUSBDEV.Files.Inf = 10,INF

[EZUSB.Dev]
CopyFiles=EZUSB.Files.Ext, EZUSB.Files.Inf
AddReg=EZUSB.AddReg

[EZUSB.Dev.NT]
; copyfiles commented out for Win2K to avoid user intervention during install
; CopyFiles=EZUSB.Files.Ext, EZUSB.Files.Inf
AddReg=EZUSB.AddReg

[EZUSB.Dev.NT.Services]
Addservice = EZUSB, 0x00000002, EZUSB.AddService

[EZUSB.AddService]
DisplayName = %EZUSB.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 2 ; SERVICE_AUTO_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %10%\System32\Drivers\ezusb.sys
LoadOrderGroup= Base

[EZUSB.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,ezusb.sys

[EZUSB.Files.Ext]
ezusb.sys

[EZUSB.Files.Inf]
widget.inf

[EZUSBDEV.Dev]
CopyFiles=EZUSBDEV.Files.Ext, EZUSBDEV.Files.Inf
AddReg=EZUSBDEV.AddReg
```



```
[EZUSBDEV.Dev.NT]
; copyfiles commented out for Win2K to avoid user intervention during install
; CopyFiles=EZUSBDEV.Files.Ext, EZUSBDEV.Files.Inf
AddReg=EZUSBDEV.AddReg

[EZUSBDEV.Dev.NT.Services]
AddService = EZUSBDEV, 0x00000002, EZUSBDEV.AddService

[EZUSBDEV.AddService]
DisplayName = %EZUSBDEV.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 2 ; SERVICE_AUTO_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %10%\System32\Drivers\wdgtldr.sys
LoadOrderGroup= Base

[EZUSBDEV.AddReg]
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,wdgtldr.sys

[EZUSBDEV.Files.Ext]
wdgtldr.sys

[EZUSBDEV.Files.Inf]
widget.Inf

;-----;

[Strings]
Cypress="Cypress Semiconductor"
USB\VID_0547&PID_1004.DeviceDesc="Cypress Widget before ReNumeration (wdgtldr.sys)"
USB\VID_0547&PID_1005.DeviceDesc="Cypress Widget after ReNumeration (ezusb.sys)"

EZUSB.SvcDesc="Cypress General Purpose USB Driver (ezusb.sys)"
EZUSBDEV.SvcDesc="Cypress General Purpose USB Driver w/ Widget Loader(wdgtldr.sys)"
;end of widget inf file
```

;Widget Firmware/Device Driver Update INF File

```
[Version]
Signature="$CHICAGO$"
Class=USB
provider=%Cypress%
LayoutFile=layout.inf

[Manufacturer]
%Cypress%=Cypress

[Cypress]
; Windows will only recognize the active VID/PID
; for updates so we will use the already renumbered device as our update entry point.
%USB\VID_0547&PID_1005.DeviceDesc%=EZUSB.Dev, USB\VID_0547&PID_1005

[PreCopySection]
HKR,,NoSetupUI,,1

[DestinationDirs]
EZUSB.Files.Ext = 10,System32\Drivers

[EZUSB.Dev]
AddReg=EZUSB.AddReg

[EZUSB.Dev.NT]
; copyfiles commented out for Win2K to avoid user intervention during install
CopyFiles=EZUSB.Files.Ext
AddReg=EZUSB.AddReg

[EZUSB.Dev.NT.Services]
Addservice = EZUSB, 0x00000002, EZUSB.AddService

[EZUSB.AddService]
DisplayName = %EZUSB.SvcDesc%
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
StartType = 2 ; SERVICE_AUTO_START
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
ServiceBinary = %10%\System32\Drivers\ezusb.sys
LoadOrderGroup= Base

[EZUSB.Files.Ext]
;This section will cause the wdgtdr to be updated, which in turn will call for the ezusb.sys
;These files should be located on the install medium. You must include ezusb.sys or the system
;will continue to renumerate, even if the device driver does not have an update.
wdgtldr.sys
ezusb.sys

;-----;

[Strings]
Cypress="Cypress Semiconductor"
USB\VID_0547&PID_1005.DeviceDesc="Cypress Widget firmware is being updated."
EZUSB.SvcDesc="Cypress General Purpose USB Driver (ezusb.sys)"
;end of firmware/device driver update inf file
```

EZ-USB は登録商標であり、EZ-USB FX、EZ-USB FX2 および ReNumeration は、サイプレスセミコンダクタ社の商標である。Intel は、Intel 社の登録商標である。Windows、Microsoft および Visual C++、および / または本アプリケーションノートで参照した他の Microsoft 製品は、登録商標または米国および / または他の国の Microsoft 社の商標のいずれかである。本文献において言及したすべての製品名および社名は、それぞれ所有者の商標である。

approved dsg 6/10/02