EZ Loader Custom USB Firmware Loader Driver

Introduction

A unique feature of the Cypress EZ-USB®, EZ-USB FX™, and EZ-USB FX2™ family of microcontrollers is the ability to change device personality through firmware download and ReNumeration™. An EZ-USB-based device may have only enough nonvolatile storage in an EEPROM to store a unique Vendor ID (VID) and Product ID (PID). This VID/PID combination can be bound to a specific device driver on the host system. This VID/PID combination can also be bound to a driver whose only function is to download firmware to the device. This application note describes such a driver. The EZ Loader Driver discussed in this application note is a template that peripheral designers can use to create a custom driver that knows how to download their firmware to their specific device. The loader driver can be downloaded

Getting Started

During your development cycle you will use the EZ-USB Control Panel via the USB cable to download an Intel® Hex file or you will use the Keil tools to download your firmware via the serial cable. While these methods are fine for development, eventually you will want to automate the process of firmware download and ReNumeration. This is where the EZ Loader driver comes in.

The EZ Loader driver provided with the Cypress Semiconductor EZ-USB Xcelerator Development kits requires very little modification to support a specific device, so an extensive knowledge of Windows® driver programming is not required. However, you should be familiar with Windows Plug and Play, Windows INF files, the Windows Registry and USB.

The EZ Loader driver is a device driver and therefore requires a Windows Device Driver Kit (DDK). Before attempting to customize the EZ Loader driver, you should verify that you are able to successfully build a device driver using the DDK. Later in this application note you will verify that you can successfully build a device driver following step-by-step procedures.

EZ Loader Driver Files

The source files needed to build the EZ Loader driver are included with the EZ-USB Xcelerator Development kit and can be found in C:\Cypress\USB\Drivers\ezloader after the development tools have been installed. The driver files and a brief description follow.

1. ezloader.c – This is the main EZ Loader source file. Contains the DriverEntry() and other standard USB driver entry points along with the code to perform the firmware download.
2. ezloader.h – Header file for the EZ Loader driver.
3. firmware.c – Contains an array of Intel Hex records for the device firmware. In the step by step procedures described later, you will use the hex2c.exe utility to convert your firmware from Intel Hex Record format into C code that can be placed into this file. When the EZ Loader driver is compiled, your firmware is included in the driver image.
4. loader.c – Contains an array of Intel Hex Records for device firmware that enables download to external RAM. If your firmware extends into external RAM, this firmware is downloaded to the EZ-USB device first to enable download to external RAM. This file was created using the Cypress Hex2c utility from the a3load.hex file found in the C:\Cypress\USB\Examples\FX2\a3load directory. The a3load.hex file works for the AN2131, FX and FX2 parts.
5. ezloader.rc –Resource file. Contains driver version information.
6. makefile – Required by the DDK build utility.
7. sources – Required by the DDK build utility. You will customize this file in the “Step-by-Step Example” section to create a driver with a unique name.

Required Tools

Microsoft Visual C++ .NET

Microsoft® Visual C++ 5.0/6.0 must be installed prior to installing the DDK. The Microsoft C compiler is automatically invoked by the DDK build utility. Microsoft® Visual C++.NET™ will also work with the EZ Loader.

Windows Driver Development Kit (DDK)

A Windows Driver Development Kit is required to build the EZ Loader device driver. Please visit Microsoft at the following url for further information concerning Driver Development Kits: http://www.microsoft.com/whdc/ddk/winddk.mspx.

hex2c Utility—Intel Hex Record to C Translator

This Cypress-provided utility converts an input file of Intel Hex Records into C code that can be compiled and linked into the EZ-Loader driver. This utility creates a C file containing an array of INTEL_HEX_RECORD structures (defined in ezloader.h). It is a Win32 console application and is used as follows from a command prompt:

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

intel_hexfile_name is an input file, and is the hex file created when you compile your firmware using the Keil Tools (i.e., widget.hex).

c_filename is the output filename that is created/overwritten.

var_name is optional and is the name of the array in the C output file. By default var_name = firmware.

The hex2c utility is located in the C:Cypress\USB\bin directory. The source code for this utility is provided with the EZ-USB Xcelerator Kit in the C:\Cypress\USB\Util\Hex2c directory.

Building the EZ Loader Device Driver

Before modifying the EZ Loader driver, you should create a new directory which will be used to build your custom loader driver. This is to prevent you from inadvertently modifying the ezmon.sys driver provided in the development kit. Once a new directory has been created, and the EZ Loader source files have been copied into it, the ezloader driver can be compiled using the DDK build utility.

The file 'sources' tells the DDK 'build' utility how to build the driver. It specifies the source files that comprise the driver and the name of the driver output file. As provided, the 'sources' file will create a driver called 'ezloader.sys.'

Depending on the build environment, your completed driver will be placed in the \lib\i386\free or checked directory. Verify that these directories have been created prior to building the driver or the 'build' utility will report an error. When customizing the EZ Loader driver, the driver output file name should be changed to something other than ezloader.sys. This is done by changing the "TARGETNAME=" field in the 'sources' file to a new unique name. The "Step-by-Step Example" section will illustrate how to change the 'sources' file.

Step-by-Step Example

This step-by-step example uses a hypothetical USB device, the Cypress Widget that you are building a firmware loader driver for.

Devices that use ReNumeration actually need two PIDs. One PID is for the device prior to firmware download, and the other is the PID for the fully functional device after firmware download. We will use PID 0x1004 for the pre-firmware Widget and PID 0x1005 for the fully functional device. The latter is the PID that is embedded in the Widget firmware and is usually designated in the dscr.a51 file.

Cypress's Vendor ID (VID) are 0x0547 and 0x04B4. For the purpose of this exercise, the 0x0547 Vendor ID will be used. Additionally, the Widget will use a Product ID (PID) of 0x1004 and 0x1005. You may use the Cypress VID for development purposes only, when your device is ready for production, you will need to obtain a unique VID. For information on obtaining a Vendor ID for your production devices, visit www.usb.org.

The device driver for the Widget is the EZ-USB General-Purpose Device Driver (ezusb.sys) that is provided with the EZ-USB Xcelerator Development Kits. This driver will be bound to VID = 0x547, PID = 0x1005.

This step-by-step example will create a new driver called wdgtldr.sys (Widget Loader). This driver will be bound to VID = 0x0547, PID = 0x1004.

Follow these steps when your firmware is complete enough that you want to automate the firmware download process by integrating the firmware into a firmware loader driver. For this example we will use the ep_pair.hex firmware found in C:\Cypress\USB\Examples\EzUsb\ep_pair. If you are using the FX2, follow the same procedures outlined below using the bulkloop.hex example. The FX2 bulkloop.hex example is found in the C:\Cypress\USB\Examples\FX2\bulkloop directory.

1. Create a new directory called Widget and copy the entire contents (files and subdirectories) of the C:\Cypress\USB\Drivers\ezloader directory into it.
2. Edit the 'sources' file with a text editor in the newly created directory and change the line TARGETNAME=ezloader to TARGETNAME=wdgtldr (*Figure 1*) You should use no more than eight characters for your loader name. A line in the sources file will also need to be uncommented, unless you are using the Windows 98 DDK. The following line needs to be uncommented: TARGETLIBS=$(DDK_LIB_PATH)\usbd.lib. To do this, simply remove the # symbol and also delete the space following the # symbol(*Figure 2*).
3. Using the Keil tools, open the ep_pair.Uv2 project file located in C:\Cypress\USB\Examples\EzUsb\ep_pair if using either the AN2131 or the CY7C64613 parts. If using the CY7C68013 FX2, open the bulkloop.Uv2 project file in C:\Cypress\USB\Examples\FX2\bulkloop. In the dscr.a51 file change the PID to 1005 as shown in *Figure 3*, and then recompile the project. Note that these bits are in little endian order as shown below. Place a copy of the ep_pair.hex (bulkloop.hex for FX2) file into the following directory: C:\Cypress\USB\Bin. While this is not required, doing so stops you from having to type in the full path to your file in the next step. Another option is to place a copy of hex2c.exe in your the ep_pair.hex or bulkloop.hex as appropriate, firmware directory.

```
#*********************************************
#*
#*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
```

Figure 1. TARGETNAME in Widget Loader "Sources" File

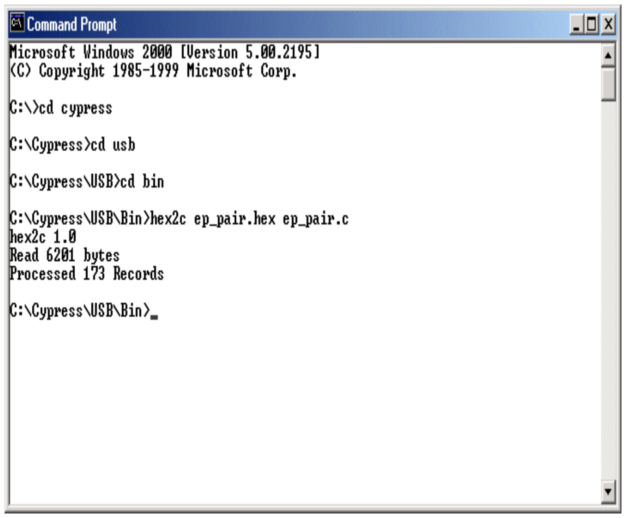
```
# to build this driver using the Windows 2000 DDK, uncomment the following line:
TARGETLIBS=$(DDK_LIB_PATH)\usbd.lib
```

Figure 2. "Sources" File modified for Win XP SP1 DDK

```
;-----------------------------------------------------------------------------
;; Global Variables
;;-----------------------------------------------------------------------------
;; Note: This segment must be located in on-chip memory.
        rseg DSCR        ;; locate the descriptor table anywhere below 8K
DeviceDscr:dbdeviceDscrEnd-DeviceDscr;; Descriptor length
        db    DSCR_DEVICE;; Decriptor 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:
```

Figure 3. Changing VID in dscr.a51

4. Use the ‘hex2c.exe’ utility to convert the Intel Hex Record format firmware to C code that can be linked into the loader driver. Open a command prompt window and navigate to C:\Cypress\USB\Bin or your firmware directory if you copied hex2c into it. At the command prompt type the following: hex2c ep_pair.hex ep_pair.c (*Figure 4*). This will take the ep_pair.hex file as input and generate ep_pair.c. This new C file now contains a large array of INTEL_HEX_RECORD type structures called firmware. See *Figure 5*. FX2 use hex2c bulkoop.hex bulkloop.c.
5. Place a copy of ep_pair.c or bulkloop.c into the Widget directory you created earlier. Open the firmware.c file for editing and replace its firmware array with the firmware array from ep_pair.c (*Figure 4*). After you have modified the firmware.c file, save it and then delete the ep_pair.c or bulkloop.c file from the directory.
6. Now you are ready to build the firmware loader driver. Open your DDK via the Window Start and Programs menus. When a command prompt is opened navigate to C:\Widget\ezloader. At the command prompt type build -c (*Figure 6*). Your driver will be located in C:\Widget\ezloader\lib\i386. Place the wdgtldr.sys file in your Windows driver directory.



```
Command Prompt
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>_
```

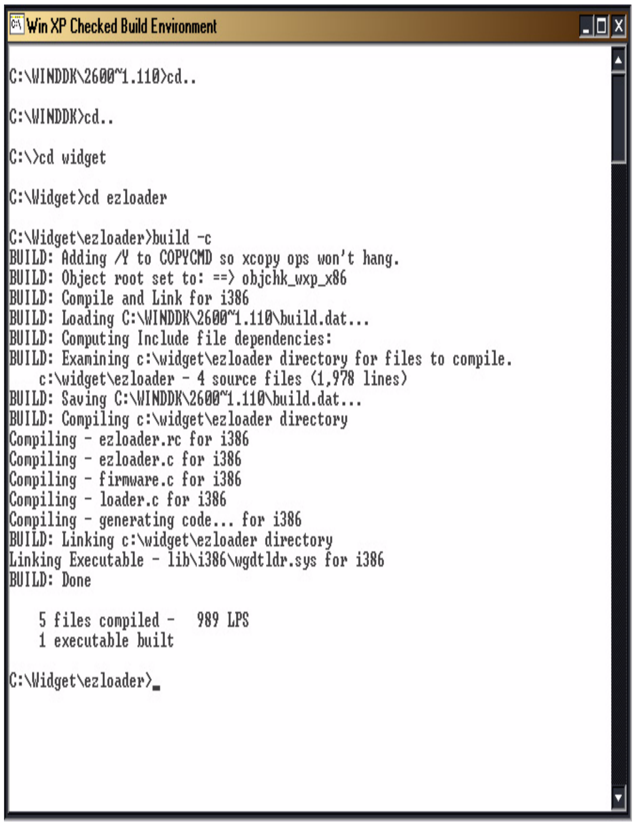
Figure 4. Converting ep_pair.hex Using the Hex2c Utility

```
INTEL_HEX_RECORD firmware[] = {
  16,
  0x42f6,
  0,
  {0xe4,0xf5,0x2c,0xf5,0x2b,0xf5,0x2a,0xf5,0x29,0xc2,0x03,0xc2,0x00,0xc2,0x02,0xc2},
  16,
  0x4306,
  0,

{0x01,0x12,0x46,0x15,0x7e,0x44,0x7f,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,
. . . .
```

Figure 5. ep_pair.c INTEL_HEX_RECORD Structure



```
Win XP Checked Build Environment
C:\WINDDK\2600~1.110>cd..

C:\WINDDK>cd..

C:\>cd widget

C:\Widget>cd ezloader

C:\Widget\ezloader>build -c
BUILD: Adding /Y to COPYCMD so xcopy ops won't hang.
BUILD: Object root set to: ==> objchk_wxp_x86
BUILD: Compile and Link for i386
BUILD: Loading C:\WINDDK\2600~1.110\build.dat...
BUILD: Computing Include file dependencies:
BUILD: Examining c:\widget\ezloader directory for files to compile.
    c:\widget\ezloader - 4 source files (1,978 lines)
BUILD: Saving C:\WINDDK\2600~1.110\build.dat...
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
Compiling - generating code... for i386
BUILD: Linking c:\widget\ezloader directory
Linking Executable - lib\i386\wgdtldr.sys for i386
BUILD: Done

    5 files compiled -   989 LPS
    1 executable built

C:\Widget\ezloader>_
```

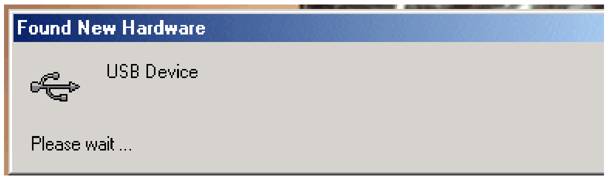
Figure 6. Building wgdtldr with Windows XP SP1 DDK

7. To bind the new driver with the widget device you will need to create an INF file. A sample INF file for the Cypress widget used in this example is provided at the end of this application note. The completed INF file should be placed in the same directory where other inf files are located in your system. This example was written for Windows 2000 systems and is compatible with Windows 98. Additional information on creating INF files can be found at: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/setupapi/setupapiovr_6xf7.asp. While the order you place your sections inside the INF file should not matter, the order of the items inside of a section can be critical.
8. Before you can use your new firmware driver you will need to bind it with your device. This is done by loading the serial EEPROM with the VID/PID combination defined in the INF file you create for your custom USB device. This is the first of two VID/PID combinations your device will use. When the VID/PID in the EEPROM is reported to the host, the host will invoke the wdgtldr.sys which loads your firmware. To load your VID/PID combination in the serial EEPROM open the EZ-USB Control Panel, ensure the Target Field matches your target device, and download the Vend_ax.hex example. Do not use the following Cypress PIDs: 0080, 0081, 1002, 2122, 2125, 2126, 2131, 2136, 2225, 2226, 2235, 2236 or 8613. Using these PID values can cause conflicts in the Windows Registry and result in the wrong driver being loaded.
   — **EZ-USB**: Change the Dir field to 0 OUT and the Hex Bytes field to B0 47 05 04 10 01 00 and press the Vend Req button. Next, change the Dir field to 1 IN and press the Vend Req button and verify the same values are reported back to you (*Figure 7*).
   — **EZ-USB FX**: Change the Dir field to 0 OUT and the Hex Bytes field to B4 47 05 04 10 01 00 and press the Vend Req button. Next, change the Dir field to 1 IN and press the Vend Req button and verify the same values are reported back to you.
   — **EZ-USB FX2**: Change the Dir field to 0 OUT and the Hex Bytes field to C0 47 05 04 10 01 00 00 and press the Vend Req button. Next, change the Dir field to 1 IN and press the Vend Req button and verify the same values are reported back to you. See section 3.5 of the *FX2 Technical Reference Manual* for additional EE-PROM configuration byte (byte 8) information.

When you disconnect and reconnect your device the Windows New Hardware Wizard will appear (*Figure 8*) and install your device. The first VID/PID combination causes the host to load your firmware via the wdgtldr.sys firmware driver you have created (*Figure 9*). The device then ReNumerates to the VID/PID combination in the ep_pair.hex or bulkllop.hex example created in step 3, loading the ezusb.sys device driver (*Figure 10*).

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

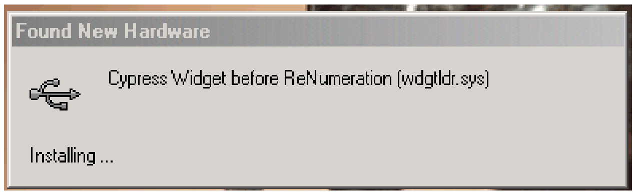
Figure 7. Programming the EEPROM with EZ-USB Control Panel


Found New Hardware

USB Device

Please wait ...

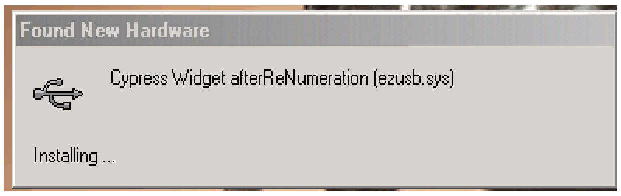
Figure 8. Windows New Hardware Wizard Detects Your Device


Found New Hardware

Cypress Widget before ReNumeration (wdgtldr.sys)

Installing ...

Figure 9. wdgtldr.sys Will Load Your Firmware


Found New Hardware

Cypress Widget afterReNumeration (ezusb.sys)

Installing ...

Figure 10. Device Now ReNumerates and Uses ezusb.sys

Verifying the wdgtldr.sys

To verify the wdgtldr.sys, open the EZ-USB Control Panel. Press the GetDevice button (*Figure 11*). Press the GetPipes button (*Figure 12*). If using the FX2, you will see addtional pipes declared (*Figure 13*). Finally, pressing the bulkloop button demonstrates the ep_pair example (*Figure 14*) The device has been configured without any further user interaction.

```
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
```

Figure 11. Pressing GetDevice report

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

Figure 12. Pressing GetPipes report

```
Get PipeInfo
Pipe:  0  Type: BLK Endpoint:  2 OUT  MaxPktSize: 0x200
Pipe:  1  Type: BLK Endpoint:  4 OUT  MaxPktSize: 0x200
Pipe:  2  Type: BLK Endpoint:  6 IN   MaxPktSize: 0x200
Pipe:  3  Type: BLK Endpoint:  8 IN   MaxPktSize: 0x200
```

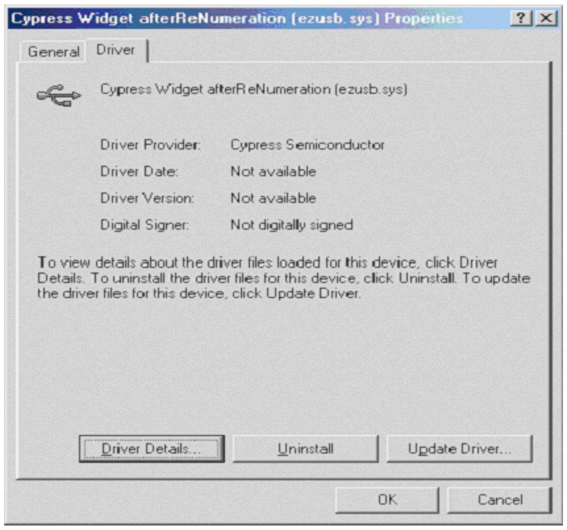
Figure 13. Pressing GetPipes report - FX2

```
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
```

Figure 14. Pressing BulkLoop Report

Updating the EZ Loader

Providing firmware and/or device driver updates can be accomplished by creating an Install Shield program, which is beyond the scope of this application note, or by using the update INF example file shown at the end of this application note. *Figures 15* through *18* show a summary of the update process from selecting Update Driver in the Windows Device Manager to a EZ-USB Control Panel display of the updated firmware (the bulktest example was used in creating the updated wdgtldr.sys. to replace the wdgtldr.sys built using ep_pair.) The update procedure shown is from a Windows 2000 system but this process works equally well with Windows XP..


Cypress Widget afterReNumeration (ezusb.sys) Properties

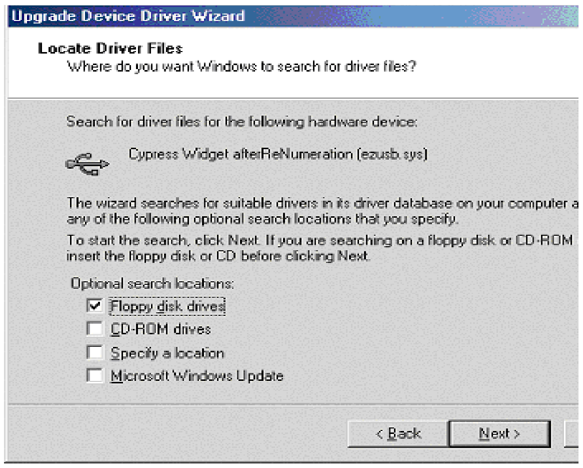
General | Driver

Cypress Widget afterReNumeration (ezusb.sys)

Driver Provider: Cypress Semiconductor
Driver Date: Not available
Driver Version: Not available
Digital Signer: Not digitally signed

To view details about the driver files loaded for this device, click Driver Details. To uninstall the driver files for this device, click Uninstall. To update the driver files for this device, click Update Driver.

Figure 15. Updating Firmware or Device Driver


Upgrade Device Driver Wizard

Locate Driver Files
Where do you want Windows to search for driver files?

Search for driver files for the following hardware device:

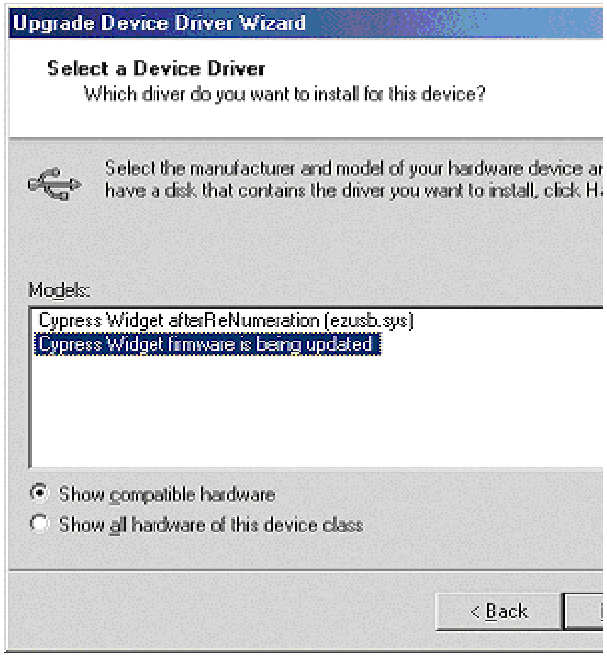
Cypress Widget afterReNumeration (ezusb.sys)

The wizard searches for suitable drivers in its driver database on your computer a
any of the following optional search locations that you specify.

To start the search, click Next. If you are searching on a floppy disk or CD-ROM
insert the floppy disk or CD before clicking Next.

Optional search locations:
☑ Floppy disk drives
☐ CD-ROM drives
☐ Specify a location
☐ Microsoft Windows Update

Figure 16. Updating Firmware (continued)


Upgrade Device Driver Wizard

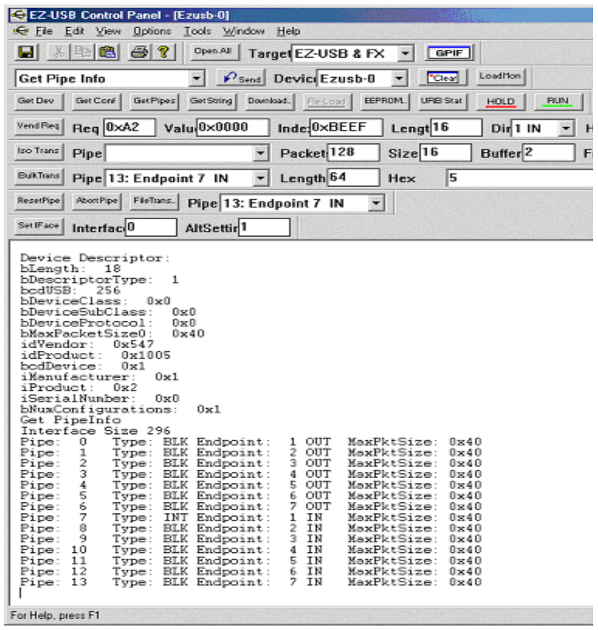
Select a Device Driver
Which driver do you want to install for this device?

Select the manufacturer and model of your hardware device ar
have a disk that contains the driver you want to install, click H

Models:
Cypress Widget afterReNumeration (ezusb.sys)
Cypress Widget firmware is being updated

● Show compatible hardware
○ Show all hardware of this device class

Figure 17. Selecting Updated Driver



Target EZ-USB & FX
Get Pipe Info
Devic Ezusb-0
Req 0xA2 Valu 0x0000 Inde 0xBEEF Lengt 16 Dir 1 IN
Pipe Packet 128 Size 16 Buffer 2
Pipe 13: Endpoint 7 IN Length 64 Hex 5
Pipe 13: Endpoint 7 IN
Interfac 0 AltSettir 1

```
Device Descriptor:
bLength:  18
bDescriptorType:  1
bcdUSB:  256
bDeviceClass:  0x0
bDeviceSubClass:  0x0
bDeviceProtocol:  0x0
bMaxPacketSize0:  0x40
idVendor:  0x547
idProduct:  0x1005
bcdDevice:  0x1
iManufacturer:  0x1
iProduct:  0x2
iSerialNumber:  0x0
bNumConfigurations:  0x1
Get PipeInfo
Interface Size 296
Pipe:  0   Type: BLK Endpoint:  1 OUT  MaxPktSize: 0x40
Pipe:  1   Type: BLK Endpoint:  2 OUT  MaxPktSize: 0x40
Pipe:  2   Type: BLK Endpoint:  3 OUT  MaxPktSize: 0x40
Pipe:  3   Type: BLK Endpoint:  4 OUT  MaxPktSize: 0x40
Pipe:  4   Type: BLK Endpoint:  5 OUT  MaxPktSize: 0x40
Pipe:  5   Type: BLK Endpoint:  6 OUT  MaxPktSize: 0x40
Pipe:  6   Type: BLK Endpoint:  7 OUT  MaxPktSize: 0x40
Pipe:  7   Type: INT Endpoint:  1 IN   MaxPktSize: 0x40
Pipe:  8   Type: BLK Endpoint:  2 IN   MaxPktSize: 0x40
Pipe:  9   Type: BLK Endpoint:  3 IN   MaxPktSize: 0x40
Pipe: 10   Type: BLK Endpoint:  4 IN   MaxPktSize: 0x40
Pipe: 11   Type: BLK Endpoint:  5 IN   MaxPktSize: 0x40
Pipe: 12   Type: BLK Endpoint:  6 IN   MaxPktSize: 0x40
Pipe: 13   Type: BLK Endpoint:  7 IN   MaxPktSize: 0x40
```

Figure 18. Updated Firmware Load in Control Panel

Conclusion

In this application note the Cypress General-Purpose USB driver, ezusb.sys was used as the device driver after renumeration. Developers should note that this driver is provided as a starting point for driver development and it is not intended to serve as a production device driver. A list of Cypress recommended consultants to assist you with your driver development is available at: http://www.cypress.com/support/cypros.cfm

The fw.c file included in the ep_pair example and other examples provided in our development kits, contains code that ensures descriptor and set-up information is loaded internal to the chip. Do not comment out this code when developing your firmware, unless you are positive that the descriptors will reside in on-chip RAM.

As you are creating your custom INF file and you need to make changes during the development process it is a good idea to clean the Windows Registry of the VID/PID combinations for your widget. When you clean the Window Registry you should also delete the wgdtldr PNF file that will be located in the Windows/INF directory. We recommend making a back up copy of your Windows Registry prior to making any modifications.

After following the process outlined in this application note you are now able to create a firmware loader driver to automate the firmware download for your custom USB device. You have also learned of one method available for issuing firmware loader driver updates for your custom USB device.

Widget Troubleshooting

If after building the Widget you experience problems, check the following:

- Ensure you do not have a space in the directory name where your Driver Development Kit is located.
- If using any Driver Development Kit other than the Windows 98 DDK, ensure you have uncommented the line and removed the leading space as noted in step 2 of the "Step-by-Step Example" instructions.
- If you are using Windows 98 or your end users may be using Windows 98, ensure your driver name is no more than 8 characters.
- If you loose communications with the development board procedures for regaining communications for each development board can be found in a Knowledge Base article available on our web site, search for Corrupted EEPROM.
- If you need to clean the Windows Registry, procedures for different Operating Systems can be found in a Knowledge Base article available on our web site, search for Windows Registry.

Additional support is available via our web-based technical support system at www.cypress.com.

Approved AN041_B 7/9/04 kkv

;WIDGET INF File

```
; Place your Copyright information here



[Version]
Signature="$CHICAGO$"
Class=USB
ClassGuid={36FC9E60-C465-11CF-8056-444553540000}
provider=%Cypress%
;CatalogFile=ezusb.cat
;The CatalogFile entry above is an example and should be uncommented and modified for your driver
;DriverVer=06/18/2003, 1.0.0.0
;The DriverVer entry above is an example and should be uncommented and modified for your driver


[Manufacturer]
%Cypress%=Cypress

;[SourceDisksNames]
;1=%strWidgetSourceDiskName%,,,
;The above section is for example purposes
;Uncomment and modify this section for your driver

;[SourceDisksFiles]
;wdgtldr.sys=1
;ezusb.sys=1
;The above section is for example purposes
;Uncomment and modify this section for your driver

[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.  Your finished product should use your own unique VID
; see www.usb.org for additional information
%USB\VID_0547&PID_1004.DeviceDesc%=WIDGET.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 we
; want to use the EZ-USB General Purpose Device Driver.  Your finished
; product should use your own unique VID - see www.usb.org
%USB\VID_0547&PID_1005.DeviceDesc%=EZUSB.Dev, USB\VID_0547&PID_1005


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

[EZUSB.Dev]
CopyFiles=EZUSB.Files.Ext, EZUSB.Files.Inf
AddReg=EZUSB.AddReg
;This section is for Windows 98, if you do not wish to support Windows 98
;delete or comment out above section

[EZUSB.Dev.NT]
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      = 3                  ; SERVICE_DEMAND_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

[WIDGET.Dev]
CopyFiles=WIDGET.Files.Ext, WIDGET.Files.Inf
AddReg=WIDGET.AddReg
;This section is for Windows 98, if you do not wish to support Windows 98
;delete or comment out above section

[WIDGET.Dev.NT]
CopyFiles=WIDGET.Files.Ext
AddReg=WIDGET.AddReg

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

[WIDGET.AddService]
DisplayName    = %WIDGET.SvcDesc%
ServiceType    = 1                  ; SERVICE_KERNEL_DRIVER
StartType      = 3                  ; SERVICE_DEMAND_START
ErrorControl   = 1                  ; SERVICE_ERROR_NORMAL
ServiceBinary  = %10%\System32\Drivers\wdgtldr.sys
LoadOrderGroup = Base

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

[WIDGET.Files.Ext]
wdgtldr.sys


;---------------------------------------------------------------;

[Strings]
;strWidgetSourceDiskName = "Cypress USB Drivers"
;The string above and the strings below should be modified to meet your needs
Cypress="Cypress Semiconductor"
USB\VID_0547&PID_1004.DeviceDesc="Using widget.inf for EZ-Loader before ReNumeration (wdgtldr.sys)"
USB\VID_0547&PID_1005.DeviceDesc="Cypress Widget afterReNumeration (ezusb.sys)"

EZUSB.SvcDesc="Cypress General Purpose USB Driver (ezusb.sys)"
WIDGET.SvcDesc="Cypress General Purpose USB Driver w/ Widget Loader (wdgtldr.sys)"

;The Windows DDK provides a chkinf tool that can be used to check your inf for errors.
;The chkinf tools requires Perl which can be downloaded, at the time of publication,
;at http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl

;end of widget inf file
```

;Widget Firmware/Device Driver Update INF File

```
; Place your Copyright information here

[Version]
Signature="$CHICAGO$"
Class=USB
ClassGuid={36FC9E60-C465-11CF-8056-444553540000}
provider=%Cypress%
;CatalogFile=ezusb.cat
;The CatalogFile entry above is an example and should be uncommented and modified for your driver
;DriverVer=06/18/2003, 2.0.0.0
;The DriverVer entry above is an example and should be uncommented and modified for your driver


[Manufacturer]
%Cypress%=Cypress

;[SourceDisksNames]
;1=%strWidgetSourceDiskName%,,,
;The above section is for example purposes
;Uncomment and modify this section for your driver

;[SourceDisksFiles]
;wdgtldr.sys=1
;ezusb.sys=1
;The above section is for example purposes
;Uncomment and modify this section for your driver


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


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

[EZUSB.Dev]
AddReg=EZUSB.AddReg, EZUSB.Files.Inf
;This section is for Windows 98, if you do not wish to
;support Windows 98, delete or comment out the above section.

[EZUSB.Dev.NT]
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      = 3                  ; SERVICE_DEMAND_START
ErrorControl   = 1                  ; SERVICE_ERROR_NORMAL
ServiceBinary  = %10%\System32\Drivers\ezusb.sys
LoadOrderGroup = Base

[EZUSB.Files.Ext]
;This section will cause the wdgtldr 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)"

;The Windows DDK provides a chkinf tool that can be used to check your inf for errors.
;The chkinf tools requires Perl which can be downloaded, at the time of publication,
;at http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl

;end of firmware/device driver update inf file
```