Cypress USB Console Users' Guide

© 2003 Cypress Semiconductor

Table of Contents

1   Main Window

CyConsole is used to communicate with USB devices that have matched to the CyUSB.sys device driver. (See Device Driver for information on making devices match to the driver.)

The console supports hot-plugging of USB devices. When a device that matches to the driver is plugged-in to the PC it will automatically appear in the Device List near the top of the CyConsole window. If you plug-in a device and it doesn't appear in the Device List it is probably matching to a different driver.

Descriptor information for the selected USB device is displayed on the Device Properties tab.
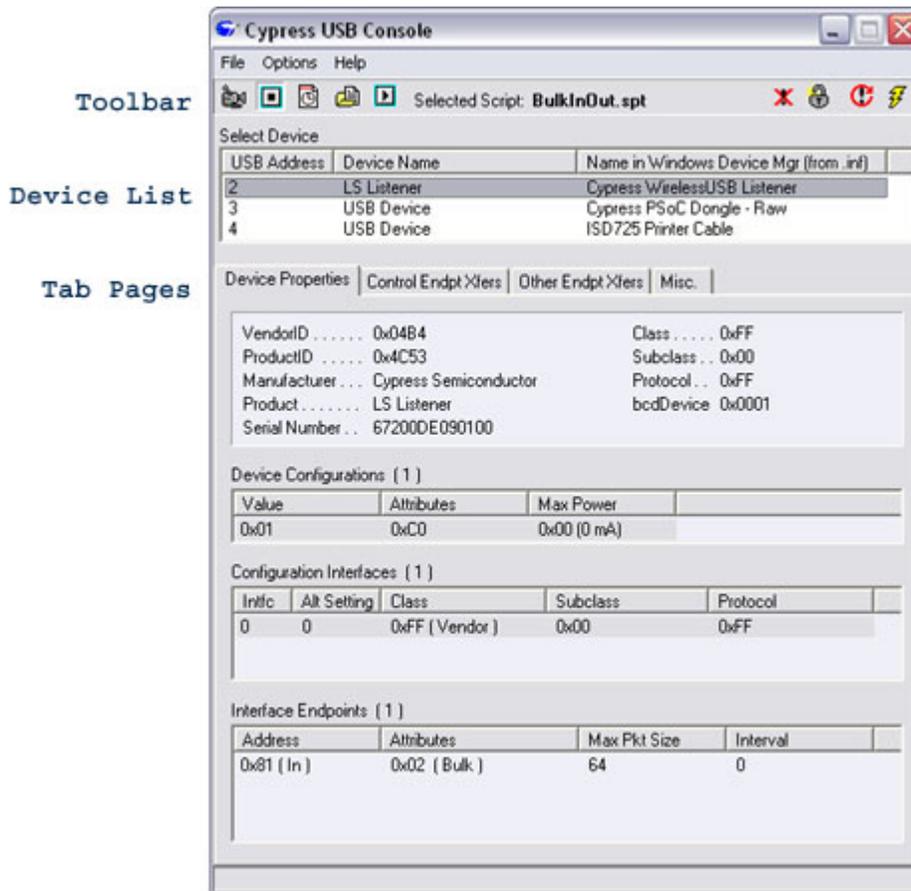
The **Configuration Interfaces** section lists all the interfaces (including alternate interfaces) exposed by the device. Clicking on an interface in the list automatically selects that interface and causes its endpoints to be displayed in the **Interface Endpoints** list.

Control transfers are setup and performed from the Control EndPt Xfers tab page.

Bulk, Interrupt and Isochronous transfers are setup and performed from the Other EndPt Xfers tab page.

1.1 Toolbar

The toolbar provides a set of speed buttons to perform utility tasks.
The group on the left is for recording and playing sequences of IO commands called scripts.
The group of buttons on the right are for device IO management.
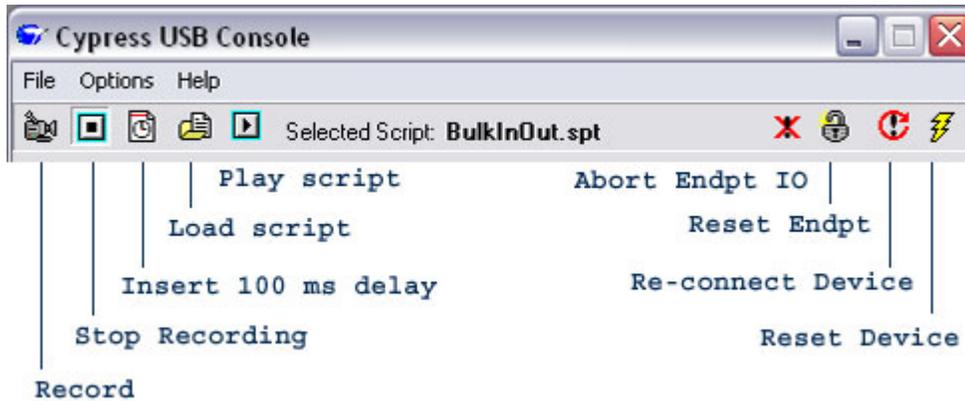
1.2 Scripts

CyConsole provides a mechanism for recording and re-playing a sequence of data transactions.
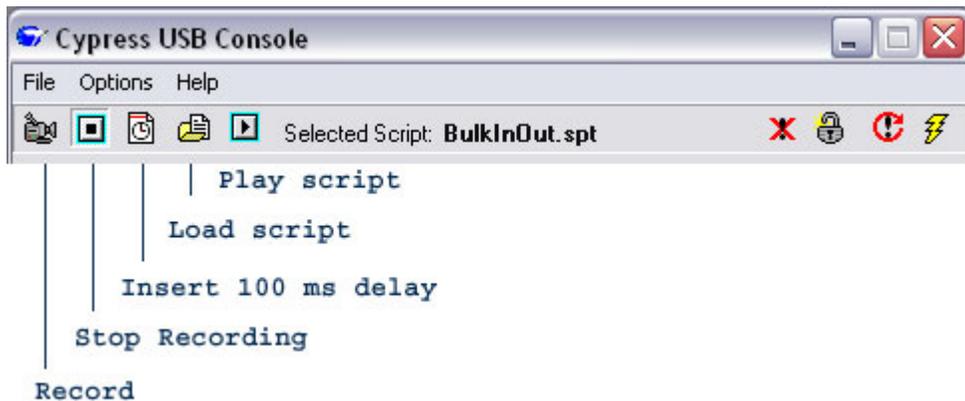
A recording session is started by clicking on the camera icon in the toolbar. The status bar at the bottom of the Console window will display "Recording . . ." in red.

While recording, any data transaction is saved to a temporary file. This includes simple data transfers and file transfers. Each transaction records the interface and endpoint address of the transfer, as well as the output data. It does *NOT* record the USB device. So, before playing-back a recorded script, the user MUST select the appropriate USB device to receive the transaction commands.

The recording session is ended by clicking on the 'Stop Recording' icon in the toolbar (next to the camera icon) or by pressing the Record button a second time. When a recording session is stopped, the user is prompted for a file name in which to save the recorded script of data transfers.

Delays of 100 ms can be added to the script while recording. Click on the clock icon in the toolbar to insert a delay.

To play-back a recorded script of transfer commands:

    1) Click on the Load Script icon in the toolbar.
    2) Select the script file to be loaded.
    3) Select the target device from the USB Device list
    4) Click the Play Script icon in the toolbar.

1.3   Device Properties

The Device Properties tab displays the descriptor values for the selected USB device. This tab also lists all the alternate interfaces reported for the device and all the endpoints reported for each alternate interface.

Each alternate interface setting for the device appears as a separate interface entry in the 'Configuration Interfaces' list. To change Alternalte Interface settings, simply select the appropriate line in the list.
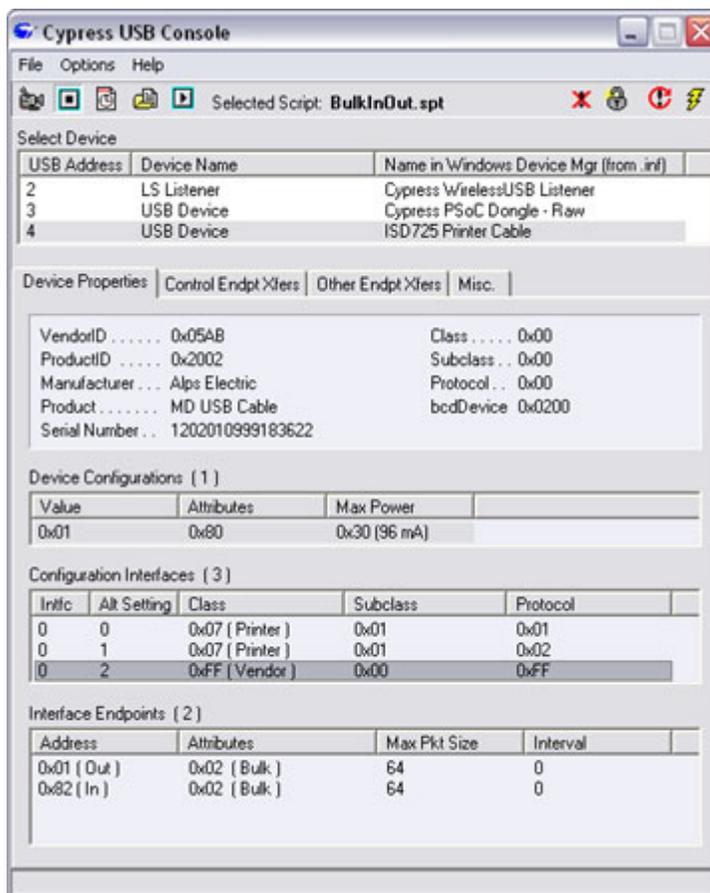
NOTE: A properly functioning Windows USB driver stack will report each primary interface as a separate USB device. So, USB devices that report multiple primary interfaces will actually show-up as multiple USB devices in the USB device list (one for each interface).

1.4 Control Endpt Xfers

To perform a control pipe transfer:

1) Select the device in the device list
2) Select the Control Endpt Xfers tab
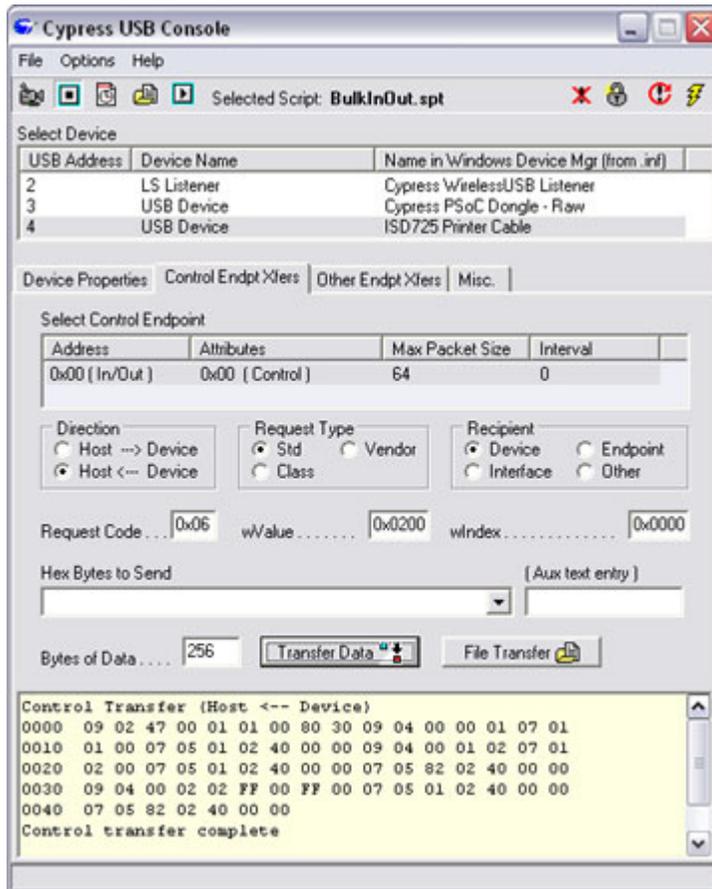
3) Select the desired control endpoint (most devices only expose 1 control endpoint)
4) Specify the direction of the transfer
5) Specify the type of transfer
6) Specify the recipient of the transfer
7) Specify the Request Code (in hexadecimal) for the transfer
8) Specify the wParam (in hexadecimal) for the transfer
9) Specify the wIndex (in hexadecimal) for the transfer
10) Enter any data bytes that should be transferred (these can be entered as hexadecimal or as text)
11) Enter the number of 'Bytes of Data' to transfer (decimal)
12) Click on the 'Transfer Data' or 'File Transfer' button

To clear the output area, right click on the output region and select 'Clear'.

1.5 Other Endpt Xfers

The direction of the data transfer on non-control endpoints is implied by the address of the endpoint. (Addresses with the high-order bit set are IN endpoints. . . usually noted by an address of 0x8_ .)

By selecting the desired endpoint, the transfer direction is also defined.

To perform a non-control pipe data transfer:

1)   Select the device in the device list
2)   Select the desired Alt Interface on the Device Properties tab
3)   Select the Other Endpt Xfers tab
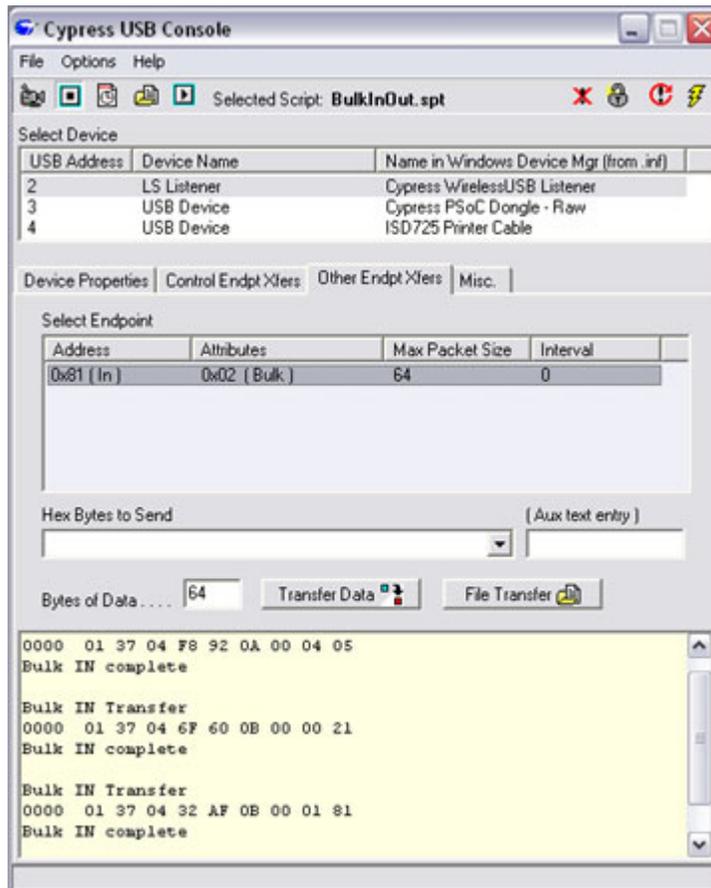
4)   Select the desired endpoint (Note that whenever a different endpoint is selected, the Bytes of Data is reset to the Max Packet Size for that endpoint.)
5)   Enter any data bytes that should be transferred (these can be entered as hexadecimal or as text)
6)   Click on the 'Transfer Data' or 'File Transfer' button

To clear the output area, right click on the output region and select 'Clear'.

2   EZ-USB Interface

The EZ-USB Interface is accessed from the Tools menu item of the main menu. The interface consists of a dialog box that mimics the behavior of a previous Cypress control panel application.

The EZ-USB Interface presents several functions on its Device and Descriptors Toolbars that are specific to the Cypress FX2 family of USB devices. The remaining toolbars, Vend Req, Iso Trans, Bulk Trans, Pipes and IFace, are all simply alternative ways to access functionality that is available from the console's Main Window.
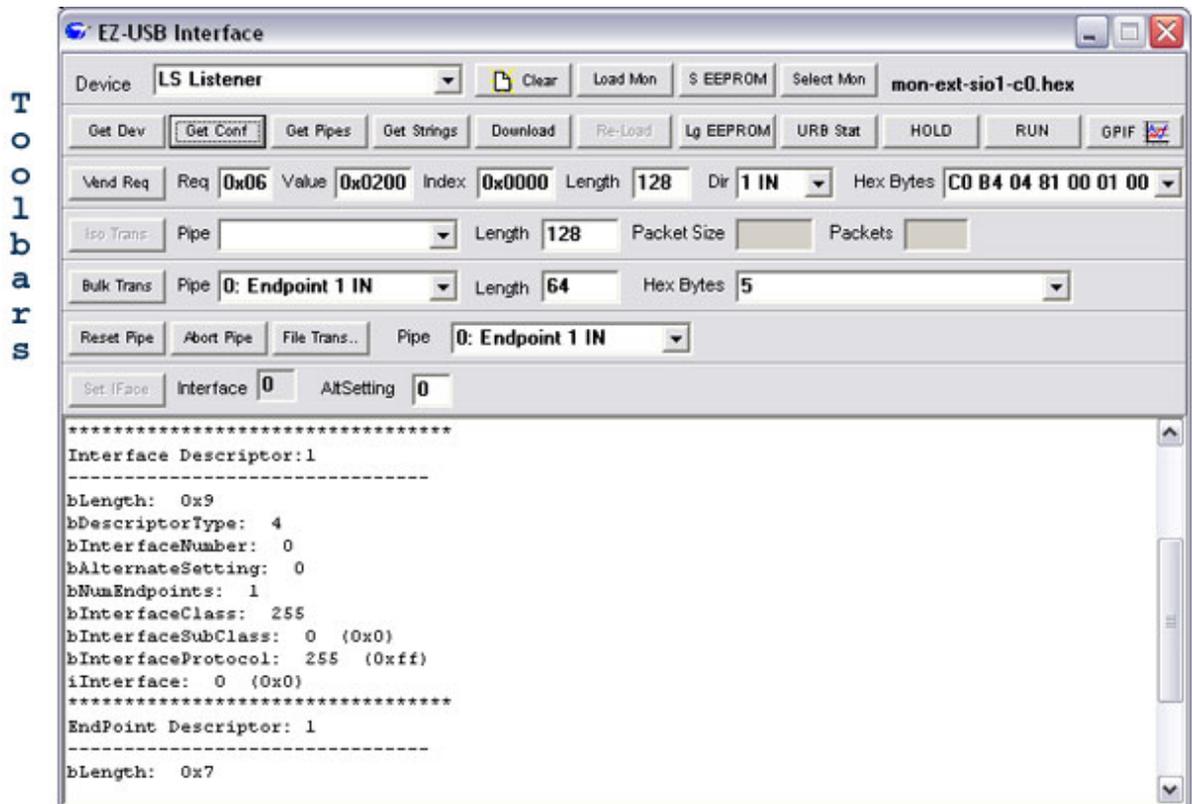
You can cause the EZ-USB Interface to pop-up automatically each time the console application launches by checking the "**Show EZ UI on Startup**" menu item in the main console window's **Options** menu.

2.1　Device Toolbar

The Device Toolbar presents a collection of high-level, fairly unrelated functions.

The **Device Selector** combo box is used to select the USB device with which to communicate.

The **Clear** button erases the contents of the output area.

The **LoadMon** button uses the <u>Download button</u> functionality to send the selected monitor file to an attached FX2's RAM.

The **S EEPROM** button is used to program 256-byte EEPROMs. When this button is clicked, the user is prompted to select a (.iic) file to load into the EEPROM. (This button first downloads the Vend_Ax.bix file, then sends the 0xA2 vendor request to program the EEPROM.)

The **Select Mon** button is used to change the file that will be downloaded when the **LoadMon** button is clicked. The name of the monitor file that is currently specified is displayed to the right of the Select Mon button.

The **Monitor File Name** displays the file that will be downloaded when the **LoadMon** button is

clicked. Click on the file name itself or the **Select Mon** button to change the selected file.

2.2   Descriptors Toolbar

The Descriptors Toolbar (of the EZ-USB Interface) provides buttons for displaying different USB descriptors, returned by the selected USB device.

In addition, some Cypress FX2-specific functions are presented on this toolbar.

The **Get Dev** button causes the selected device's Device Descriptor information to be displayed in the output area.

The **Get Conf** button causes the selected device's Configuration Descriptor information to be displayed in the output area.

The **Get Pipes** button displays information about the selected Alternate Interface's endpoints.

The **Get Strings** button causes the selected device's String Descriptors to be displayed in the output area.

The **Download** button prompts the user to select a file to load into an attached FX2's RAM.

The **Re-Load** button downloads the last file selected into an attached FX2's RAM.   (Performs the Download function without allowing the user to select a file.)

The **Lg EEPROM** button is used to program 64-Kbyte EEPROMs. When this button is clicked, the user is prompted to select a (.iic) file to load into the EEPROM. (This button first downloads the Vend_Ax.bix file, then sends the 0xA9 vendor request to program the EEPROM.)

The **URB Stat** button displays the error messages from the most recently completed USB transaction.

The **HOLD** button performs an FX2 CPU Reset, setting the CPUCS.0 bit to 1. (Equivalent to sending a vendor request of A0 with wValue = 0xE600, wIndex = 0x0000 and data byte of 0x01.)

The **RUN** button releases the FX2 CPU Reset, setting the CPUCS.0 bit to 0. (Equivalent to sending a vendor request of A0 with wValue = 0xE600, wIndex = 0x0000 and data byte of 0x00.)

If the Cypress GPIF Designer application has been installed at "..\..\GPIF Designer", relative to the CyConsole.exe file (true if both CyConsole and GPIF Designer were installed to their default destinations), the **GPIF** button will launch **GPIF Designer** . If **GPIF Designer** is not found at this location, the button will be hidden.

2.3   Vend Req Toolbar

The Vend Req Toolbar is used to send vendor-specific control transfers to a device. To send other types of control requests (Standard, or Class requests) or to send requests to a different recipient (Interface, Endpoint, or Other) use the CyConsole Control Endpt Xfers tab.

Enter the byte representing your vendor request command code in the **Req** field.

Enter the two-byte hexadecimal value for your vendor request's wValue parameter in the **Value** field.

Enter the two-byte hexadecimal value for your vendor request's wIndex parameter in the **Index** field.

Indicate whether the data is to be sent to the device (OUT) or read from the USB device (IN) using the **Dir** drop-down list.

Type-in the data bytes to send into the **Hex Bytes** combo box. Separate each byte by a non-hexadecimal character (i.e. a space).

Enter the number of bytes to transfer in the **Length** field. To send multiple copies of the same byte, select OUT as the Direction for the transfer. Enter the single byte to send in the Hex Bytes field. Enter the number of copies to send in the Length field.

Initiate the data transfer by clicking the **Vend Req** button.

2.4   Iso Trans Toolbar

The Iso Trans Toolbar is used to perform isochronous data transfers over isochronous endpoints.

Once the Pipe and Length of transfer have been specified, the **Iso Trans** button causes the transfer to be performed.

The **Pipe** combo box is used to select the endpoint on which to perform the transfer. Only isochronous endpoints (Attributes = 0x01) are listed in the box.

Specify the number of bytes to transfer in the **Length field.** Note that Iso transfers on IN endpoints should always be an exact multiple of the **Packet Size**. Otherwise, the transfer will likely fail.

The **Packet Size** is a read-only value displayed here for convenience. All Iso transfers performed by the application use the wMaxPacketSize of the endpoint. This is the value displayed in the **Packet Size** field.

The **Packets** field is updated on the fly as the Length field is changed. This field displays the number of packets that are required for the transfer, based on the **Length** and the **Packet Size** (Length / Packet Size).

2.5   Bulk Trans Toolbar

The Bulk Trans Toolbar is used to perform data transfers through bulk pipes.

Use the **Pipe** combo box to select the pipe through which you want to transfer the data.

Type-in the data bytes to send into the **Hex Bytes** combo box. Separate each byte by a non-hexadecimal character (i.e. a space).

Enter the number of bytes to transfer in the **Length** field. To send multiple copies of the same byte, select an OUT pipe. Enter the single byte to send in the Hex Bytes field. Enter the number of copies to send in the Length field.

Click the **Bulk Trans** button to perform the specified data transfer.

If the bulkloop.hex file has been downloaded to an FX2 device, the **Bulk Loop** button will act as a toggle to start/stop the iterative transmission of the specified Hex Bytes over Pipe 0 (Endpoint address 0x02) and read them back over Pipe 2 (Endpoint address 0x86).

2.6   Pipes Toolbar

The Pipes Toolbar allows endpoint-specific operations on any of the non-control endpoints.
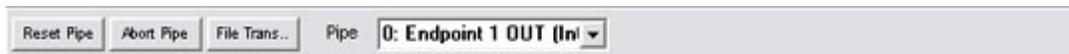
The **Pipe** combo box presents a list of all the non-Control endpoints exposed by the current Alternate Interface. The current selection changes to reflect the most recent selection from the Bulk Pipes or the Iso Pipes combo boxes.

The **Reset Pipe** button causes the selected endpoint to be reset.

To clear pending data transfers on the endpoint, click the **Abort Pipe** button.

The **File Trans . .** button will prompt for a source or destination file to be used in transferring data over the selected Pipe. (Source or Destination is implied by the direction of the endpoint.)

2.7   Set IFace Toolbar

The Set IFace Toolbar allows selection of the Alternate Interfaces of the device.
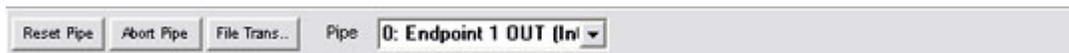
The **Set IFace** button selects the alternate interface indictated by the **AltSetting** value.

The alternate interface can also be selected by typing a valid **AltSetting** and pressing the Enter key.

Note that, because Windows always exposes a single primary interface for a device, the **Interface** value is fixed at zero.

3   Device Driver

The CyConsole communicates with USB devices via the CyUSB.sys device driver. Only USB devices serviced by the CyUSB.sys driver can be accessed from the CyConsole application.

Usually, matching of a USB device to the CyUSB.sys driver will need to be manually configured. This configuration consists of two steps.

Step 1 : Add the device's VendorID and ProductID to the CyUSB.inf file.

After installation of the SuiteUSB files,the driver is located in a Driver subdirectory of the install directory. (Default is C:\Program Files\Cypress\DevStudio\Driver.)

Open the file CyUSB.inf with a text editor (notepad.exe, for instance)

Locate the **[Cypress]** section and duplicate the line
; %VID_**VVVV** &PID_**PPPP** .DeviceDesc%=CyUsb, USB\VID_**VVVV** &PID_**PPPP**

Remove the semicolon from the duplicate line

Change the **VVVV** to contain the hexadecimal value of the VendorID for the device

Change the **PPPP** to contain the hexadecimal value of the ProductID for the device

For example, a device with vendorID 0x04B4 and productID 0xDE01 would have a new entry in the [Cypress] section like the following
%VID_04B4&PID_DE01.DeviceDesc%=CyUsb, USB\VID_04B4&PID_DE01

Now, move to the bottom of the CyUSB.inf file and locate the **[Strings]** section and duplicate the line.
; VID_**VVVV** &PID_**PPPP** .DeviceDesc="**My Device Description** "

Remove the semicolon from the duplicate line

Change the **VVVV** to contain the hexadecimal value of the VendorID for the device

Change the **PPPP** to contain the hexadecimal value of the ProductID for the device

For example, a device with vendorID 0x04B4 and productID 0xDE01 would have a new entry in the [Strings] section like   the following
VID_04B4&PID_DE01.DeviceDesc="Cypress OTG DE1 DevBoard"

Save the file.

Step 2 : Force Windows to use the CyUSB.sys driver with the device.

Connect the device to the PC

If Windows prompts for a driver or indicates that it needs a driver, direct the PC to use the CyUSB.sys   driver by steering it to the CyUSB.inf file in the [InstallDir]\Driver   directory.

If Windows does not prompt for a driver, it has already matched the device to a driver itself. In this case, you will need to manually instruct Windows to use the CyUSB.sys   driver.

Right-click **My Computer** and select the Manage menu item.

In the **Computer Management** window, select **Device Manager**

In the right window pane, click the + icon next to **Universal Serial Bus controllers**

Locate your device in the list and double click on it

Select the **Driver** tab in the Properties dialog that comes up

Select **Update Driver**

Click Next

Select **Display a list of the known drivers for this device**

Click Next

Click **Have Disk**

Click **Browse**

Navigate to the directory containing CyUSB.sys

CyUSB.inf should be automatically placed in the File name field

Click Open

Click OK

Click Next

Click Next

Don't re-boot your system when Windows suggests that you must.

4   Using Custom Drivers

Device drivers can be accessed / identified in the Windows operating system by a unique number called their Global Unique IDentifier (or GUID). Two drivers with the same GUID would not play well together on a system. (The *Unique* part of GUID would have been violated.)

The CyUSB.sys driver provides a mechanism to allow customers to register the driver with a new GUID that they create.   Any application that desires to access devices via that new driver GUID would also need to know the GUID.

CyConsole was designed with provision for connecting to other CyUSB.sys drivers that have different GUIDs. On the **Misc.** tab sheet you will notice a drop-down list labelled **Access devices attached to this driver** . The drop-down list is populated with the file names of all copies of CyUSB.sys that are installed in the system. (It gathers the list from the Windows registry.)