CY3662

EZ-811HS Development Kit Guide

Doc. No. 001-69598 Rev. **

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

Phone (USA): 800.858.1810
Phone (Intnl): 408.943.2600
http://www.cypress.com

Copyrights

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

EZ-USB is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.>

Flash Code Protection

Cypress products meet the specifications contained in their particular Cypress PSoC Data Sheets. Cypress believes that its family of PSoC products is one of the most secure families of its kind on the market today, regardless of how they are used. There may be methods, unknown to Cypress, that can breach the code protection features. Any of these methods, to our knowledge, would be dishonest and possibly illegal. Neither Cypress nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

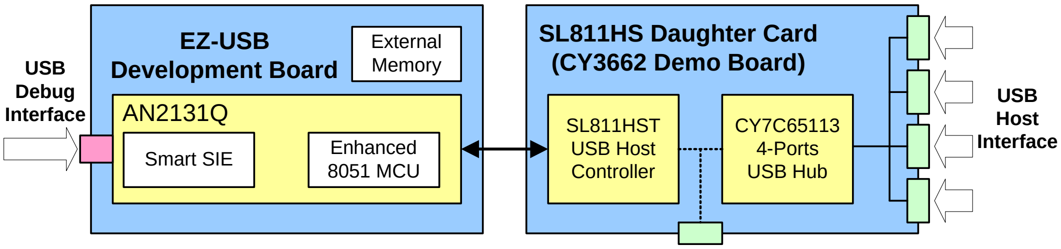
Cypress is willing to work with the customer who is concerned about the integrity of their code. Code protection is constantly evolving. We at Cypress are committed to continuously improving the code protection features of our products.

Contents

1. Introduction

The EZ-811HS Development Kit (DVK) hardware consists of an EZ-USB® development board and a SL811HS daughter card (CY3662 demo board). The SL811HS is an embedded USB Host and Slave Controller capable of communicating with either full-speed or low-speed peripherals. The SL811HS can interface to devices such as microprocessors, microcontrollers, and DSPs. The EZ-USB development board is the ideal starting platform for SL811HS development. It uses the internal 8051 core as the MCU interface to the SL811HS and the EZ-USB interface for debugging. Figure 1-1 shows the interconnect block diagram between the SL811HS daughter card and the EZ-USB development board.

Figure 1-1. Interfacing SL811HS Daughter Card with EZ-USB Development Board



1.1 Kit Contents

The EZ-811HS Development Kit includes the following components:

- SL811HS daughter card (CY3662 demo board)
- EZ-USB development board
- USB cable
- Serial cable
- Power supply
- Kit CD

1.1.1 CD Contents

Table 1-1 shows the directories that are installed for the EZ-811HS Development Kit.

Table 1-1. Installation Directories on the CD

Directories	Description
`C:\CYPRESS\USB\ez811\bin`	Binary files and firmware files
`C:\CYPRESS\USB\ez811\doc`	EZ-811HS related documentation - Getting Started and data sheets
`C:\CYPRESS\USB\ez811\EZ811_DK`	EZ-811 PC host application
`C:\CYPRESS\USB\ez811\firmware\Emb_Host`	Embedded host firmware

Table 1-1. Installation Directories on the CD

Directories	Description
`C:\CYPRESS\USB\ez811\firmware\slave`	Slave mode firmware
`C:\CYPRESS\USB\ez811\firmware\USBHost`	Simple embedded USB host application documentation and firmware
`C:\CYPRESS\USB\ez811\HARDWARE`	Hardware related information
`C:\CYPRESS\USB\ez811\Drivers\Linux`	Linux driver for 811HS

1.2 Design Support and Resources

- SL811HS Data Sheets and Application Notes

  The kit CD contains the SL811HS data sheet and application notes. You can also view the following application notes on the Cypress website.

  - AN1215 - Basic Embedded Host Using the SL811HS
    http://www.cypress.com/?rID=12903
  - AN1137- Interfacing an External Processor to the SL811HS/S
    http://www.cypress.com/?rID=12923

- Design with CY3662 (SL811HS)

  The EZ-811HS DVK User Guide on the CD provides the description of hardware and firmware.

- EZ-USB Technical Reference Manual (TRM)

  The EZ-USB TRM on the CD contains a detailed description of the features and functionality of the EZ-USB device. Learn about the various features of EZ-USB and how they are implemented.

- SL811HS Firmware Examples

  The SL811 firmware examples on the CD contains both host and slave mode firmware examples that help you to develop your own firmware.

- Online Technical Support

  For knowledge base articles, customer forums, and online application support, visit http://www.cypress.com/support.

1.3 Document Revision History

Table 1-2. Revision History

Revision	PDF Creation Date	Origin of Change	Description of Change
**	27/06/2011	VSO	Initial version of kit guide

1.4 Documentation Conventions

Table 1-3. Document Conventions for Guides

Convention	Usage
`Courier New`	Displays file locations, user entered text, and source code: `C:\ ...cd\icc\`
Italics	Displays file names and reference documentation: Read about the *sourcefile.hex* file in the *PSoC Designer User Guide*.
[**Bracketed, Bold**]	Displays keyboard commands in procedures: [**Enter**] or [**Ctrl**] [**C**]
File > Open	Represents menu paths: File > Open > New Project
Bold	Displays commands, menu paths, and icon names in procedures: Click the **File** icon and then click **Open**.
Times New Roman	Displays an equation: $2 + 2 = 4$
Text in gray boxes	Describes Cautions or unique functionality of the product.

2. Hardware Configuration

The EZ-811HS DVK includes a hardware daughter card that has the SL811HST and the CY7C65113 4-port hub. This daughter card is designed to be inserted directly into the header sockets available on the EZ-USB development kit, without any wire connections. The 4-port USB hub extends the number of downstream ports available to a maximum of four. You have a choice of either one USB port (directly from SL811HST) or four downstream ports (through CY7C65113). Figure 2-1 is a photograph of the EZ-811HS development board. There are several jumpers, which enable you to select different configuration modes.

JP1 – Master/Slave Mode Select Jumper

JP2 – Slave Speed Select Jumper

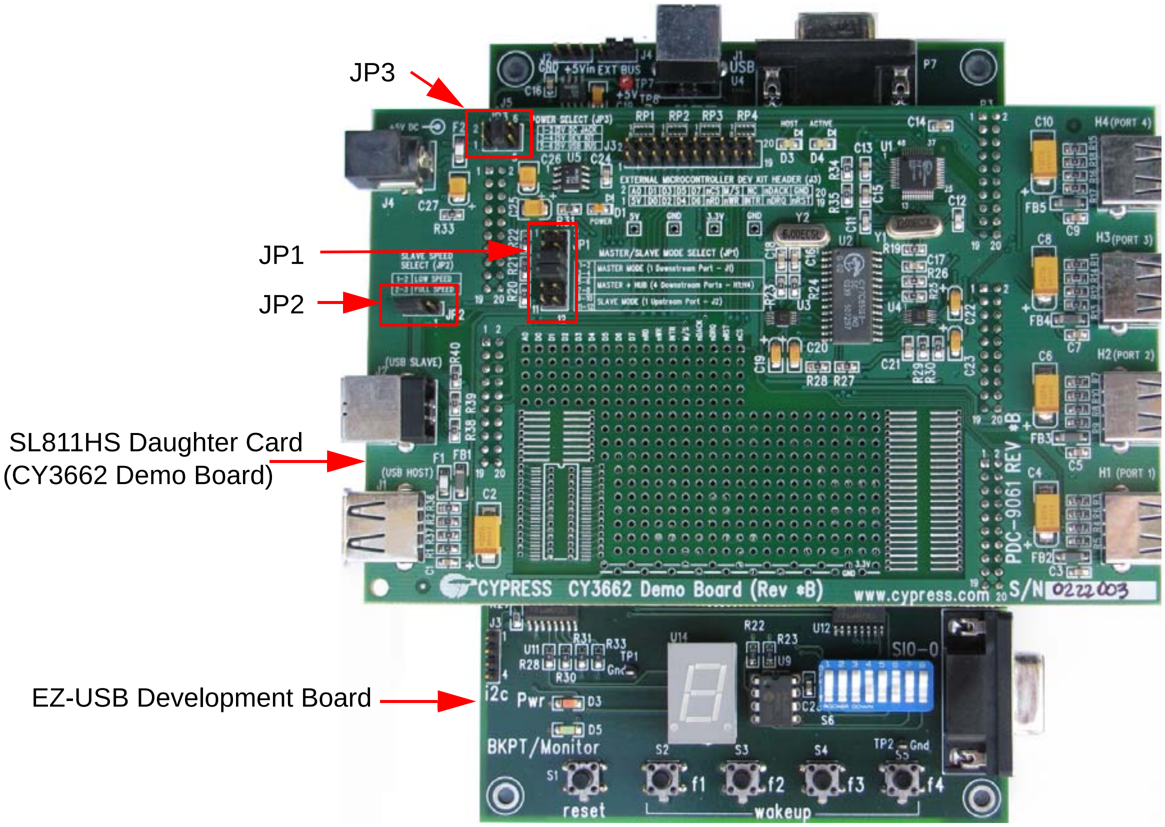
JP3 – Power Select Jumper

The master/slave mode select jumper provides the following options:

- Choose between the SL811HST USB host only, with one downstream port (J1)
- Choose to automatically allow the SL811HST to enumerate with the on-board hub to extend the downstream ports to four (H1~H4)
- Choose slave operation when SL811HST is configured as a slave USB controller. This slave mode is configured together with the slave speed select jumper

The power select source (JP3 on the EZ-811HS DVK) can also be selected from an external DC power adapter, EZ-USB development kit, or from the USB bus during slave mode operation. Data and control signals are also available on the header to ease connection of the SL811HS to other types of CPU interfaces.

Figure 2-1. SL811HS Daughter Card and EZ-USB Development Board



Configuring the EZ-USB DVK requires only one DIP switch setting to be made - DIP switch SW4 of S6 must be in the ON position to disable the lower 32 K external memory. This allows the SL811HST address and data registers to be mapped to this area. The upper 32 K of external memory is allocated to firmware code and variables. Note that when the SL811HST demo board is inserted on to the headers (P1~P5) of the EZ-USB DVK and the EZ-USB enumerates, “Cypress EZ-USB Development Board” should be displayed. This indicates that the monitor code for debugging has loaded successfully (green LED or D5 on the board turns on) and the Keil Debugger can be used to download and debug code.

Code and variable spaces must be carefully placed to avoid disruption to the monitor code space. Firmware code can be downloaded in two ways - using USB or serial connection. Use the Control Panel software for the EZ-USB DVK to download code through the USB into the 8051 core. You can also use Keil uVision 2 to download code through the serial port. On successful download, the host LED (D3 on the EZ-811 board) turns on, indicating that SL811HST is in host mode, and the ACTIVE LED blinks indicating that it is waiting for any slave devices that are attached to the SL811HST to enumerate. It is recommended that LEDs also be connected to indicate successful enumeration of the hub’s downstream devices. Port A bits 7 to 4 are configured in firmware to indicate which downstream Port 1 to 4, respectively, has successfully completed the enumeration process.

3. Host Mode Firmware Operation

3.1 Host Firmware

The firmware includes detection of any USB slave devices, enumeration of standard devices, and support for HID and hub devices. The firmware is limited to supporting a maximum of five simultaneously connected devices. This consists of a 4-port hub connected directly to the SL811HS followed by four non-hub devices downstream. The firmware does not support a multiple hub tier structure, multiple interface devices, or ISO-transfers. After successful enumeration of a hub device, any downstream device is also detected and goes through the process of enumeration. Dynamic USB address allocation of slave device is used; this means that the next available USB address is assigned to the current USB device plugged in. Any device plugged directly into the SL811HST always starts with USB address 1.

3.2 Firmware Files Functional Description

The firmware is based on the EZ-USB framework, allowing modular code development. Table 3-1 lists the firmware files and their description.

Table 3-1. Firmware Files with Description

Firmware Filename	Description
fw.c	EZ-USB framework handles enumeration of the EZ-USB development kit and provides the EZ-USB interface to the PC or laptop.
host2131.c	Handles the interface to the EZ-USB development board. This includes sending bulk command data to the 8051 for the SL811HST and returning data captured from the slave device back to the PC. It handles the IN and OUT transfers to and from EZ-USB's host PC.
dscr.a51	Defines the EZ-USB interface and endpoint pipes.
host_811.c	This is the main file that implements all functions to communicate with the SL811HST. You can easily extract the code required to implement just the host controller portion to interface to others microprocessor or MCUs.

The firmware file of host_811.c provides all functions to communicate with SL811HST. Table 3-2 describes the function of the major functions routines inhost_811.c.

Table 3-2. Functional Description of Major Routines in host_811.c

Routine Name	Description
slave_detect()	This is the main loop that is called repeatedly by the TD_Poll() routine in host2131.c. If there is a slave device detect, it updates the data structures containing information about the host controller and downstream ports.
speed_detect()	This function performs three basic setups: detects new device attachment, identifies the speed of the device attached, and sets up various control registers for proper operation at the correct speed.
EnumUsbDev()	This routine supports the entire enumeration process of a slave USB device. It provides the top-level routines to perform each individual USB request required to enumerate a device. Including USB reset, get descriptors (device, configuration, string and class-specific), set address, and set configuration. Note that the current firmware only supports a single interface configuration. At the same time, to facilitate EZ-USB's host software device attachment update, necessary slave device information is stored during the enumeration process. Examples are VIDs, PIDs, EP0 maximum packet size, and each data endpoint's attributes such as endpoint address, direction, and payload size. This routine only stores up to a maximum of four data endpoints of information, which can be increased as required.
ep0Xfer()	This function handles control endpoint 0 transactions during enumeration and any other USB host requests. It calls *usbXfer()* to set up and initiate any host requests by passing information such as USB address, payload size, SETUP request types, and buffer location of returned data (if any). A USB request consists of three stages - setup stage (SETUP token with request DATA0), data stage (IN/OUT token with DATAx), and status stage (IN/OUT token with null DATA0). Actual USB traffic is initiated by the routine *usbXfer()*.
usbXfer()	This is the core of all USB data and control transactions process. It writes to appropriate registers of SL811HST to initiate a USB transaction as required, such as a write (SETUP/OUT+data) or a read (IN). It also handles low-speed transaction through a hub by appending a PREamble token for any request that goes down all the way to a low-speed device attached to a hub. After each host request is sent, it waits for an acknowledgement from the slave device by means of an interrupt. It then determines the type of response from the slave device and terminates as necessary. If there is a request for multiple data from a device that is greater than the maximum endpoint zero payload size, it must re-arm the SL811HST to grab the next set of data from device until all are received. To send data to the device, store data into the buffer, give SL811HST the start address of the buffer, set data length, and arm the SL811HST to start USB's SETUP/OUT data transaction.
DataRW()	This function is similar to *ep0Xfer()*, except that it is used only for data transfer. By specifying the USB device address, endpoint address, maximum payload size, data length, and buffer address, you can initiate any data transaction, including IN to or OUT of, the slave device. Note that you must write to the buffer if you are doing an OUT data transfer.
HubPortEnum()	Upon successful enumeration of a hub device, endpoint 1 of the hub is used to determine any port change if there is a downstream device attachment or detachment. During a change, this routine detects which port has changes and whether it is a device connect or disconnect. During an attachment, it also performs reset to the slave device, determines the device speed, and calls *EnumUsbDev()* to enumerate the attached USB device. When a device is disconnected, it clears the connection state change-bit of the hub.

As mentioned earlier, take care in organizing the code and data memory of the firmware. The code and data spaces are allocated as follows:

- Internal 8 K EZ-USB Code/Xdata Memory:
  Both USBJmpTb.a51 and Dscr.a51 code are located in the EZ-USB internal 8-K memory
- Lower 32 K External Memory:

  SL811H_ADDR address at 0x4000
  SL811H_DATA address at 0x4001
- Upper 32 K External Memory:

  Program's Code/Xdata Memory Allocation
  - Xdata Space: 0x2000 to 0x23FF, using EZ-USB's ISO buffer
    (size: 0x0400, 1 Kbytes)
  - Code Space1: 0x8000 to 0x9EFF, for EZ-USB's Fw.c and host2131.c
    (size: 0x1F00, 7 Kbytes)
  - Code Space2: 0xA000 to 0xDFFF, for host_811.c
    (size: 0x4000, 16 Kbytes)
- Monitor's Code/Xdata Memory Allocation (Do Not Use)
  - Monitor xdata Space: 0x9F00 ~ 0x9FFF
    (size: 0x0100, 256 bytes)
  - Monitor Code Space: 0xE000 ~ 0xFFFF
    (size: 0x2000, 8 Kbytes)

These demonstrations operate at +10 dBm of RF output power. They are limited to +10 dBm because of the RF power restrictions imposed in Europe and Japan. The power can be increased to +20 dBm in the United States and Canada only. The process is explained in detail on pages 47 and 64 of the CY3271 PSoC FirstTouch Starter Kit Guide.

3.3 PC Application Software

The PC application is used to demonstrate and exercise the EZ-811HS kit in host mode.

Because the SL811HST daughter card (CY3662 demo board) works on the EZ-USB development board platform, the EZ-USB's general purpose driver (Cypress EZ-USB sample device) is used. This driver performs basic control of the SL811HST in the form of a Windows-based software application program; see Figure 3-1. With the general-purpose driver, you can communicate with the EZ-USB and request information from slave devices that are attached on the SL811HST.

The software can update the devices and display USB device information such as USB address, speed type, class type, VID, PID, control, and data endpoint's attributes. By changing the USB address field, you can communicate directly with that device, requesting its device descriptor, configuration descriptor, string descriptor, or class descriptor. By selecting the appropriate endpoint number and data length, the software is able to initiate data transactions to that endpoint. The slave USB device in turn responds with data to the host controller, the SL811HST, which is then transferred back to the EZ-USB's host software. You can easily add functions in the software to help it ensure that the embedded host development is working as expected.

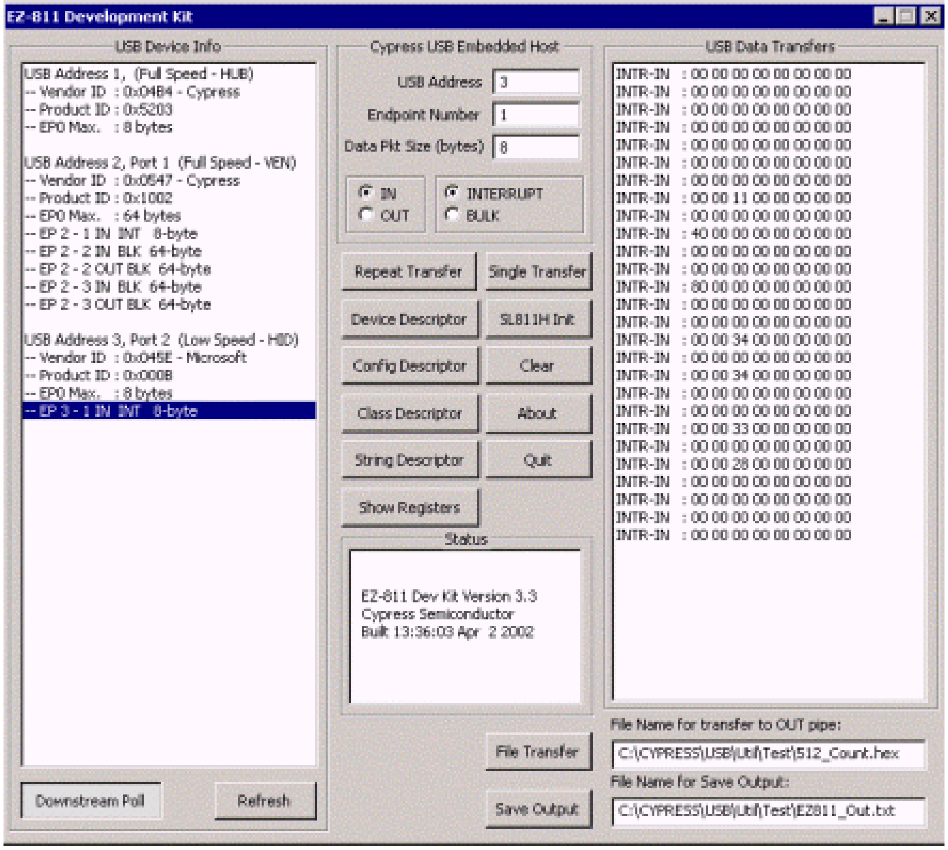
3.3.1 PC Application Software Operation

Figure 3-1 shows the EZ-811HS Host Demo Software (application software). The operation procedure is as follows:

1. To use **Master+Hub** mode, set jumpers 5 – 6 and 7 – 8 on the SL811 demo board.
2. Connect the USB cable from the PC to the EZ-USB development board.

3. Click **Start > Programs > Cypress > USB > EZ-USB Control Panel.** The EZ-USB Control Panel opens with a single device showing "Ezusb-0".
4. Select the **Download** button and select the following file: `C:\CYPRESS\USB\ez811\bin\Emb_Host.hex` on the SL811HST demo board. First the **HOST** LED turns on; then, the **ACTIVE** LED turns on.
5. Start the PC application software (EZ811_DK.exe) in `C:\CYPRESS\USB\ez811\bin`.
6. Plug in a peripheral device such as a mouse to an SL811HS demo board port (Port1-Port4).
7. The **Downstream Poll** button controls the downstream polling function. Turn it off to allow easier analysis of USB bus traces of downstream devices. Select is to allow the SL811HS to detect downstream device attach and detach. Use the **Refresh** button to update the Device Info window.
8. Select an interrupt endpoint (as on a mouse or keyboard) by selecting the endpoint directly in the USB Device Info window; enter the **Address**, **EP**, and **Type**.
9. Select **Repeat Transfer** and then move the mouse, or select a mouse button; see the output from the device in the USB Data Transfers window.
10. Deselect **Repeat Transfer** to stop transfers.
11. Press **Quit** to exit the host application.

Figure 3-1. EZ-811HS Host Demo Software



```
USB Address 1,  (Full Speed - HUB)
-- Vendor ID  : 0x04B4 - Cypress
-- Product ID : 0x5203
-- EP0 Max.   : 8 bytes

USB Address 2, Port 1  (Full Speed - VEN)
-- Vendor ID  : 0x0547 - Cypress
-- Product ID : 0x1002
-- EP0 Max.   : 64 bytes
-- EP 2 - 1 IN  INT   8-byte
-- EP 2 - 2 IN  BLK  64-byte
-- EP 2 - 2 OUT BLK  64-byte
-- EP 2 - 3 IN  BLK  64-byte
-- EP 2 - 3 OUT BLK  64-byte

USB Address 3, Port 2  (Low Speed - HID)
-- Vendor ID  : 0x045E - Microsoft
-- Product ID : 0x0008
-- EP0 Max.   : 8 bytes
-- EP 3 - 1 IN  INT   8-byte
```

USB Address: 3

Endpoint Number: 1

Data Pkt Size (bytes): 8

● IN ○ OUT

● INTERRUPT ○ BULK

```
EZ-811 Dev Kit Version 3.3
Cypress Semiconductor
Built 13:36:03 Apr  2 2002
```

```
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 11 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 40 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 80 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 34 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 34 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 33 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 28 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
INTR-IN   : 00 00 00 00 00 00 00 00
```

File Name for transfer to OUT pipe: C:\CYPRESS\USB\Util\Test\512_Count.hex

File Name for Save Output: C:\CYPRESS\USB\Util\Test\EZ811_Out.txt

4. Slave Mode Firmware Operation

4.1 Slave Mode Firmware

In EZ-811HS DVK installation directory, `C:\CYPRESS\USB\ez811\firmware\slave` directory contains demo code that enumerates the SL811HS (in slave mode) as a simple HID consumer control device. The firmware can be downloaded to the EZ-USB development board using the EZ-USB Control Panel, as described in PC Application Software Operation on page 13. This configures the SL811HS daughter card (CY3662 demo board) appropriately. Three consumer buttons are implemented: audio mute, volume up, and volume down.

4.1.1 SL811HS Daughter Card Setup

The demo code performs the necessary USB and HID-Class requests for successful enumeration of the slave device.

To set up the SL811HS daughter card (CY3662 demo board) for this example, follow these steps:

1. Insert the SL811HST daughter card on to the EZ-USB development board.
2. On the EZ-USB development board, turn on bit 4 of S6.
3. On the SL811HST daughter card, set JP1 to slave mode (set jumpers 9–10 and 11–12), JP2 to full speed (jumper 2–3), and JP3 to 5 V development kit.

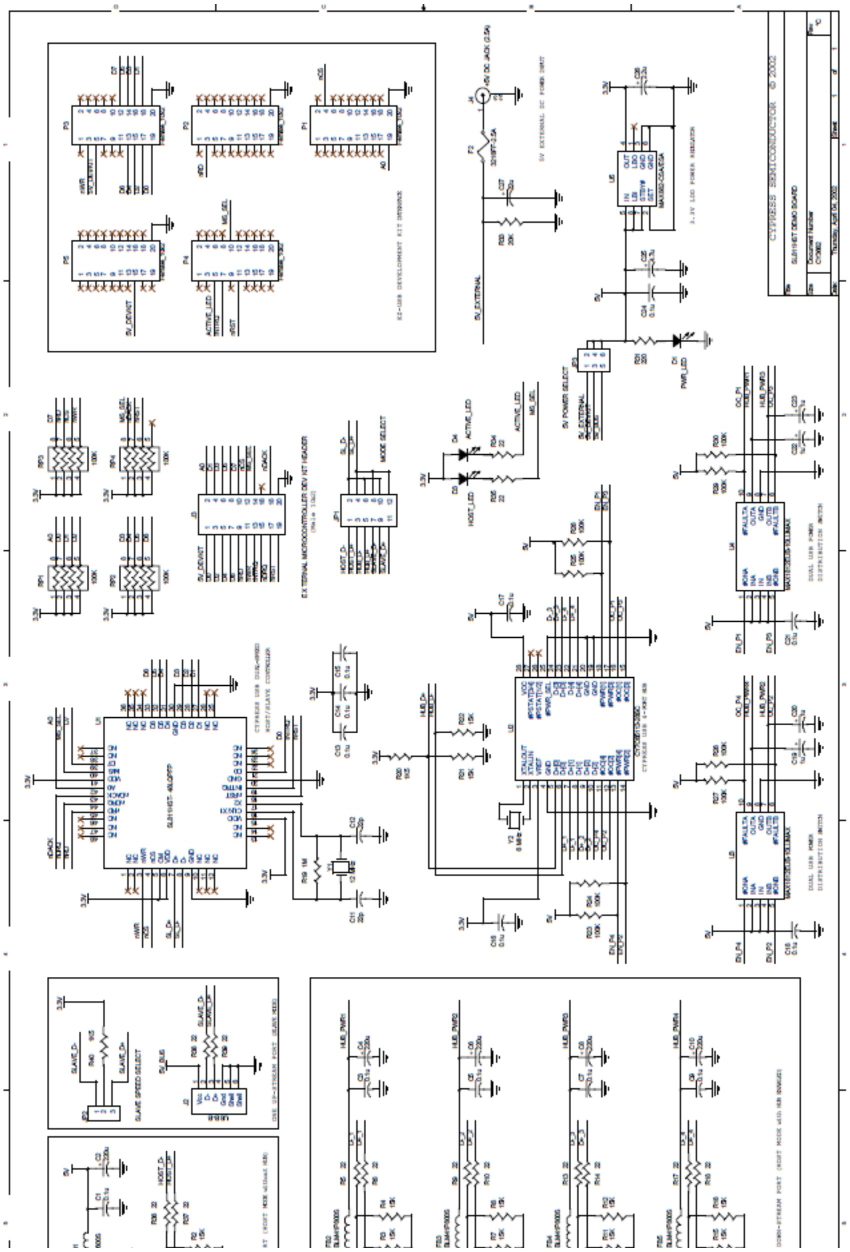
   Note JP2 = full speed means the left-most jumpers are connected, because pin 1 is on the right.
4. Optionally, add three tact switches on Port C - bit 0 to bit 2.
   a. To connect switches, pull up P5–9 to 3.3 V with a 100-K resistor and connect P5-9 with an SPST switch to ground.
   b. Repeat this setup using P5–10 and then again using P5–11. Three 100-K resistors and three SPST switches are required.
   c. On the SL811, P5–9 is bit 2, P5–10 is bit 1, and P5–11 is bit 0. Refer to the schematic for the EZ-USB Development Board.
   d. The Mute, Vol Up, and Vol Down buttons are active low.

   Note To only view enumeration, omit Step 4.
5. Insert the USB cable to connect the EZ-USB board. Use the EZ-USB Control Panel to download `ez811\bin\HIDSlave.hex`.
6. When this is done, turn the HOST LED (D3) off and ACTIVE LED (D4) off.
7. Using another USB cable and USB port on your PC, connect J2 (USB connector) to the PC. D4 on the SL811HST daughter card turns on.

   If necessary, select `C:\windows\inf` as the location containing the driver file.
8. If successful, you will see two new entries in the Device Manager:
   a. HID-Compliant Consumer Control Device
   b. USB Human Interface Device
9. Try the three buttons to control the mute, volume up, and volume down of your sound card.

A. Appendix

A.1 SL811HS Daughter Card (CY3662 Demo Board) Schematic


CYPRESS SEMICONDUCTOR © 2002
SL811HST DEMO BOARD
CY3662
5V DC JACK (2.5A)
5V EXTERNAL DC POWER INPUT
3.3V LDO POWER REGULATOR
EZ-USB DEVELOPMENT KIT INTERFACE
EXTERNAL MICROCONTROLLER DEV KIT HEADER
MODE SELECT
5V POWER SELECT
SL811HST-48LQFP
SLAVE SPEED SELECT