Introduction

Scope
The primary objective of this document is to help developers to understand the SL811HS VxWorks USB host controller driver (HCD). This document covers the following topics:

- A general overview of the SL811HS board developed by Cypress Semiconductor, the VxWorks USB stack, and the SL811HS VxWorks HCD.
- Description of the software environment setup process.
- Description of the process to set up USB test devices.

Please note that some items are covered in more detail in references found at the end of this document.

Overview
The SL811HS USB Host embedded host controller is a single chip USB embedded host solution that can communicate with either full speed or low-speed USB peripherals. The SL811HS can interface to devices such as microprocessors, micro-controllers, DSPs or directly to a variety of buses such as ISA, PCMCIA and others. The SL811HS USB controller conforms to the low and full speed requirements of Universal Serial Bus Specification, Revision 1.1, September 23, 1998.

The VxWorks host controller driver conforms to the Universal Serial Bus Specification, Revision 1.1, September 23, 1998 and can be integrated into the Wind River Systems’ VxWorks USB host driver stack to provide a complete USB solution for the VxWorks environment.

Figure 1: VxWorks USB Host Driver Stack
At the bottom of the stack is the USB host controller (USB HC), the piece of hardware in the host system that controls each USB. For each type of host controller there is a single, hardware-dependent USB host controller driver (HCD). For example, the VxWorks USB HCD for SL811HS, usbHcdSi811hsLib.

The interface between the USB host driver (USBD) and the HCD allows for each HCD to control one or more underlying HCs. Also, Wind River’s USBD is capable of connecting to multiple USB HCDs simultaneously. These design features allow you to build a range of complex USB systems.

The USBD is the hardware-independent module above the HCD(s). The USBD manages each USB connected to the host and provides the path through which higher layers communicate with the USB. Among its responsibilities, the USBD handles USB power management and USB bandwidth management automatically. Also, unique to the Wind River architecture, the USBD manages USB hubs. Hub functionality is critical to the proper operation of the USB, so the designers of the Wind River USBD concluded that hub functionality should be handled transparently by the USBD. This means that the
USBD also handles the dynamic attachment and removal of USB hubs and devices.

Figure 1 shows the USB Client Module at the top of the stack. USB class drivers are typical examples of client modules. USB class drivers are responsible for managing individual types of devices that can be connected to the USB; they rely on the USBD to provide the communication path to the individual devices. Applications, diagnostics, and test programs are other examples of client modules that rely on the USBD to communicate with USB devices. For example, Wind River provides the test application/module usbTool which gives you interactive control of the USB and USB devices.

VxWorks USB HCD for SL811HS, usbHcdSl811hsLib, conforms to the VxWorks USB stack’s USBD/HCD interface to interface to upper layers of the VxWork’s USB stack while supporting SL811HS USB host controller to interface to the hardware layer. The HCD supports control, interrupt, isochronous, and bulk endpoints.

The HCD interfaces with the VxWorks USBD/HCD via a set of application interfaces (API). Table 1 describes these API functions.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usbHcdSl811hExec()</td>
<td>HCD_EXEC_FUNC entry point for SL811H HCD</td>
</tr>
<tr>
<td>IncAttach()</td>
<td>Initialize the HCD and attach to specified bus(es)</td>
</tr>
<tr>
<td>IncDetach()</td>
<td>Disconnect from specified bus(es) and shut down</td>
</tr>
<tr>
<td>IncSetBusState()</td>
<td>Sets bus suspend/resume state</td>
</tr>
<tr>
<td>IncSofIntervalGet()</td>
<td>Retrieves current SOF (start of frame) interval</td>
</tr>
<tr>
<td>IncSofIntervalSet()</td>
<td>Set new SOF interval</td>
</tr>
<tr>
<td>IncCurrentFrameGet()</td>
<td>Return current bus frame number</td>
</tr>
<tr>
<td>IncIrpSubmit()</td>
<td>Queue an IRP (Input/Output request packet) for execution</td>
</tr>
<tr>
<td>IncIrpCancel()</td>
<td>Attempts to cancel a pending IRP</td>
</tr>
<tr>
<td>IncPipeCreate()</td>
<td>Create pipe &amp; calculate / reserve bandwidth</td>
</tr>
<tr>
<td>IncPipeDestroy()</td>
<td>Destroy pipe and release associated</td>
</tr>
</tbody>
</table>

Table 1: HCD USBD/HCD interface functions

### Vxworks Development Environment

#### System Requirements

The SL811HS HCD has been developed and tested with the development environment below (developed on Windows 2000 host); however, it can be ported to other architectures supported by VxWorks. The system development environment is composed of:

- Wind River Systems SBC405GP single board computer
- Wind River Systems SBC405GP board support package (BSP)/reference design (can be obtained from Wind River Systems)
- Cypress Semiconductor SL811HS development daughter card for SBC405GP single board computer
- Host computer supported by Wind River Systems Tornado integrated development environment. Host systems must have:
  - For Windows, an Intel 80486 or better; Intel Pentium 90 or better is recommended:
    - 64MB RAM
    - 315MB disk space for typical installation
    - A CD-ROM drive for installation
    - A browser for displaying documentation
- Wind River Systems Tornado 2.0 for PPC405 (can be obtained from Wind River Systems: part# TDK-13877-ZC-0)
- Wind River Systems Tornado 2.0 Cumulative Patch 3 (can be obtained from Wind River Systems: part# SPR-T2CP3)
- Wind River Systems Tornado 2.0 Windows 2000 Patch (can be obtained from Wind River Systems: part# SPR-62517), only required if installing Tornado on a Windows 2000 host.
• Wind River Systems USB 1.1 for Tornado 2.0 with source. Source is required on PPC405 platform, because PPC405 binaries are not included in the CD distribution. (Can be obtained from Wind River Systems: part# TDK-14145-ZC-00)

• usbHcdSl811hsLib USB HCD for SL811HS. The source files include:
  o usbHcdSl811hsLib.h
  o usbHcdSl811hsLib.c
  o usbPciStub.c
  o usbSl811hs.lib
  o Makefile (replacement for version in Wind River Systems SBC405GP board support package)
  o config.h (replacement for version in Wind River Systems SBC405GP board support package)
  o ppc405GP.h (replacement for version in Wind River Systems SBC405GP board support package)
  o sysLib.c (replacement for version in Wind River Systems SBC405GP board support package)
  o wrSbc405gp.h (replacement for version in Wind River Systems SBC405GP board support package)
  o usrUsbTool.c (replacement for version in Wind River Systems USB 1.1 for Tornado 2.0)
  o usbBulkDevLib.c (replacement for version in Wind River Systems USB 1.1 for Tornado 2.0)

### Installation

Please note that %WIND_BASE% refers to the top level Tornado directory name in the following sections.

### Hardware

Connect Cypress Semiconductor SL811HS development daughter card for SBC405GP to Wind River Systems SBC405GP single board computer (Please refer to the documentation that came with these products). The connections are illustrated in Table 2.

<table>
<thead>
<tr>
<th>SL811HS Signal</th>
<th>SBC405GP Signal</th>
<th>SBC 405GP SBC Connector, Pin No</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>BP_ADDR8</td>
<td>JX12, 45</td>
</tr>
<tr>
<td>nRD</td>
<td>BP_OE_N</td>
<td>JX14, 56</td>
</tr>
<tr>
<td>nWR</td>
<td>BP_WE_N</td>
<td>JX14, 58</td>
</tr>
<tr>
<td>NCS</td>
<td>BP_CS_N2</td>
<td>JX12, 19</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>JX14, 64</td>
</tr>
<tr>
<td>nRST</td>
<td>BDP_P_PRESET_N</td>
<td>JX14, 32</td>
</tr>
<tr>
<td>INTRQ</td>
<td>IRQ0</td>
<td>JX14, 50</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>JX11, 51</td>
</tr>
<tr>
<td>D0</td>
<td>BP_DATA0</td>
<td>JX11, 61</td>
</tr>
<tr>
<td>D1</td>
<td>BP_DATA1</td>
<td>JX11, 64</td>
</tr>
<tr>
<td>D2</td>
<td>BP_DATA2</td>
<td>JX11, 59</td>
</tr>
<tr>
<td>D3</td>
<td>BP_DATA3</td>
<td>JX11, 60</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>JX11, 56</td>
</tr>
<tr>
<td>D4</td>
<td>BP_DATA4</td>
<td>JX11, 55</td>
</tr>
<tr>
<td>D5</td>
<td>BP_DATA5</td>
<td>JX11, 58</td>
</tr>
<tr>
<td>D6</td>
<td>BP_DATA6</td>
<td>JX11, 53</td>
</tr>
<tr>
<td>D7</td>
<td>BP_DATA7</td>
<td>JX11, 54</td>
</tr>
<tr>
<td>GND</td>
<td>GND</td>
<td>JX11, 63</td>
</tr>
<tr>
<td>+5V</td>
<td>+5V</td>
<td>JX11, 8</td>
</tr>
</tbody>
</table>

Table 2: SL811HS Daughter Card Connections to SBC405GP

The DIP switches on the SBC405GP single board computer has been tested with the following settings:

<table>
<thead>
<tr>
<th>SW1-1:On</th>
<th>SW1-2:On</th>
<th>SW1-3:On</th>
<th>SW1-4:On</th>
<th>SW1-5:On</th>
<th>SW1-6:On</th>
<th>SW1-7:On</th>
<th>SW1-8:On</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW3-1:On</td>
<td>SW3-2:On</td>
<td>SW3-3:On</td>
<td>SW3-4:On</td>
<td>SW3-5:On</td>
<td>SW3-6:On</td>
<td>SW3-7:On</td>
<td>SW3-8:On</td>
</tr>
<tr>
<td>SW4-1:On</td>
<td>SW4-2:On</td>
<td>SW4-3:On</td>
<td>SW4-4:On</td>
<td>SW4-5:On</td>
<td>SW4-6:On</td>
<td>SW4-7:On</td>
<td>SW4-8:On</td>
</tr>
<tr>
<td>SW5-1:On</td>
<td>SW5-2:On</td>
<td>SW5-3:On</td>
<td>SW5-4:On</td>
<td>SW5-5:On</td>
<td>SW5-6:On</td>
<td>SW5-7:On</td>
<td>SW5-8:On</td>
</tr>
</tbody>
</table>

Table 3: SBC405GP Dip Switch Settings

- Connect network cable, serial cable, and USB cable as appropriate between host and single board computers
Software

1) Install Wind River Systems Tornado 2.0 for PPC405 (TDK-13877-ZC-0) software on host. Please refer to documentation included with product.

2) Install Wind River Systems SBC405GP reference design/board support package (BSP) software on host. The BSP installs under %WIND_BASE%\target\config\wrSbc405gp.

3) Install Wind River Systems Tornado 2.0 Cumulative Patch 3 (SPR-T2CP3)

4) If working on Windows 2000 host, install Wind River Systems Tornado 2.0 Windows 2000 Patch (SPR-62517)

5) Install Wind River Systems USB 1.1 for Tornado 2.0 (TDK-14145-ZC-00) with source.

6) Make a copy of %WIND_BASE%\target\config\wrSbc405gp directory as %WIND_BASE%\target\config\CypressUSB-ppc405gp. This new directory will be the working directory for our test environment.

7) Install usbHcdSl811hsLib USB HCD for SL811HS source files in %WIND_BASE%\target\config\CypressUSB-ppc405gp. Some of the existing files will be replaced.

8) Make a backup of existing %WIND_BASE%\target\src\drv\usb\usbBulkDevLib.c and replace it with %WIND_BASE%\target\config\CypressUSB-ppc405gp\usbBulkDevLib.c

9) Open a DOS command window (command prompt window) and execute the following commands. This will also build the USB libraries for PPC405:
   a. Setup proper VxWorks environment variables:
      
      %WIND_BASE%\host\x86-win32\bin\torVars.bat
   b. cd %WIND_BASE%\target\src
   c. make CPU=PPC405 TOOL=gnu

10) Please follow the instructions in %WIND_BASE%\target\config\wrSbc405gp\target.nr to program the bootrom code to SBC405GP Flash.

Configuration and Build

This section describes the steps required to configure and build a VxWorks image with USB support.

BSP

- %WIND_BASE%\target\config\CypressUSB-ppc405gp\config.h defines
  SL811H_IO_ADDR (0x70000000) and
  SL811H_IO_ADDR_DATA (0x70800000) for SL811HS memory mapped I/O.
  SL811H_IO_ADDR is for the SL811HS address register, and
  SL811H_IO_ADDR_DATA is for the data register. The interrupt used is also defined in this header file. These defines are
  SL811H_INT_LVL (25) and
  SL811H_INT_VEC (25)

- %WIND_BASE%\target\config\CypressUSB-ppc405gp\wrSbc405gp.h defines
  SL811H_MEMORY_START (0x70000000) and
  SL811H_MEMSIZE (0x10000000) for memory mapped I/O address region

- %WIND_BASE%\target\config\CypressUSB-ppc405gp\sysLib.c activates the memory mapped I/O region.

- As illustrated in the USB Exercise Tool, the SL811HC HCD is initialized by the following code. usbHcdSl811hExec is the entry point to the USB HCD:

  SL811_IO_CFG sl811IOCfg = {
  SL811H_IO_ADDR,
  SL811H_IO_ADDR_DATA,
  SL811H_INT_VEC,
  SL811H_INT_LVL
  };
  GENERIC_HANDLE *pToken;
  usbdHcdAttach (usbHcdSl811hExec, &sl811IOCfg, pToken);

  The following detaches the HCD:
  usbdHcdDetach (*pToken);

- The SL811HS development board is connected to the PPC405GP platform via external bus controller. The external bus
control registers must be configured before the SL811HS is functional. This is done in `usbHcdSl811hsLib.c`:

```c
LOCAL void sl811DcrInit()
{
    int regValue = 0;
    printf("Enter sl811DcrInit\n");

    /* IRQ0 from SL811HS daughter card is edge sensitive */
    regValue = sysDcrUictrGet() | 0x00000040;
    sysDcrUictrSet(regValue); /* set int trigger levels */

    /* Configure the DCR CR0 register */
    /* Disable GPIO 17 to enable IRQ0 as an interrupt */
    regValue = sysDcrCr0Get() & ~CR0_GPIO_17_EN;
    sysDcrCr0Set(regValue);

    /* Configure the DCR CR1 register to set the enable pin */
    regValue = sysDcrCr1Get() | CR1_PCI_PW_EN;
    sysDcrCr1Set(regValue);

    /* Configure the EBC Configuration Register EBC0_CFG */
    sysDcrEbcSet(EBC_CFG, 0xb84c0000);

    /* Configure the Peripheral Bank 2 Configuration Reg */
    sysDcrEbcSet(EBC_B2CR, 0x700f8000);

    /* Configure the Peripheral Bank 2 Access Reg */
    sysDcrEbcSet(EBC_B2AP, 0x090ffe00);
}
```

### Tornado Project

- Next, create project directories for Tornado GUI. The project will be created in `%WIND_BASE%\target\proj\CypressUSB-ppc405_vx`. Open a DOS command window (command prompt window) and execute the following commands:
  - Setup proper VxWorks environment variables:
    ```bash
    %WIND_BASE%\host\x86-win32\bin\torVars.bat
    cd %WIND_BASE%\target\config\CypressUSB-ppc405
    make bsp2prj
    ```
- Start Tornado GUI and open the project file `%WIND_BASE%\target\proj\CypressUSB-ppc405_vx\CypressUSB-ppc405gp_vx.wpj`
- If you are planning to use the included USB exerciser tool, add `%WIND_BASE%\target\config\CypressUSB-ppc405\usrUsbTool.c` to your Tornado workspace by selecting the “Files” tab in the workspace window. Then, right clicking on CypressUSB-ppc405_vx and adding the desired file.
- USB components can be further configured from the “VxWorks” tab in the workspace window. These entries are under “CypressUSB-ppc405gp_vx->hardware->bus services->USB Hosts” and “CypressUSB-ppc405gp_vx->hardware->peripherals->USB Devices”
- Compile and download to SBC405GP target.

### USB Excerciser Tool

Please refer to `%WIND_BASE%\target\config\CypressUSB-ppc405\usrUsbTool.c` for examples on the usage of the USB HCD. This file is a customized version of the version included with the VxWorks USB Developer’s Kit 1.1 User’s Guide. It includes support for the SL811HS HCD. The tool is further described in VxWorks USB Developer’s Kit 1.1 User’s Guide. In this customized version, `usbInit` command is called automatically upon `usbTool` activation, whereas the in the original it is not.

Sample session to test the different USB devices is shown below. Mass storage device is installed as “od:” by `usbTool`:
Cypress SL811HS VxWorks Host Control Driver User's Guide

Copyright (c) 2001, Wind River Systems, Inc.
usbTool: USB exerciser, version 01e
-> usbTool
-usbTool: USB exerciser, version 01e

Help/? Displays list of supported commands.
path Displays current default path

port 2 is device 0x17 = MULTITRIP/Dynalog
port 2 is device 0x2a = YAMAHA/YAMAHA TST-4250 USB Speaker
port 4 not connected
port 5 not connected
port 6 not connected
usb>
usb>
usb>

usb>usbnumtext
Press esc to terminate polling.
Press [enter] to terminate polling.

usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>usb>us
Porting to Other Development Platforms
The SL811HS HCD driver can be ported to other development platforms. The following steps explain the porting process.

1) The driver initializes the hardware platform and configures the access parameters to the SL811HS. For example, if the SL811HS is connected to an expansion bus, then the driver must set up the bus access timing, address range which bus is being accessed, the banks size, etc. Currently, the sl811DcrInit() function in the “usbHcdSl811hsLib.c” setups the expansion bus access profile. This function can be modified for accommodating a specific processor and platform.

2) Physical to virtual memory mapping
The physical to virtual memory mapping is defined in the sysPhysMemDesc[] structure in the sysLib.c file. The following code segment shows the memory mapping

```c
PHYS_MEM_DESC sysPhysMemDesc[] = {
    {void *) LOCAL_MEM_LOCAL_ADRS,
     (void *) LOCAL_MEM_LOCAL_ADRS,
     LOCAL_MEM_SIZE,
     VM_STATE_MASK_VALID | VM_STATE_MASK_WRITABLE |
     VM_STATE_MASK_CACHEABLE,
     VM_STATE_VALID | VM_STATE_WRITABLE |
     VM_STATE_CACHEABLE },
    .
    .
    .
    #ifdef INCLUDE_SL811H
    ,
    {void *) SL811H_MEMORY_START,
     (void *) SL811H_MEMORY_START,
     SL811H_MEMSIZE,
     VM_STATE_MASK_VALID | VM_STATE_MASK_WRITABLE |
     VM_STATE_MASK_CACHEABLE | VM_STATE_MASK_GUARDED,
     VM_STATE_VALID | VM_STATE_WRITABLE |
     VM_STATE_CACHEABLE_NOT | VM_STATE_GUARDED
    }
    #endif /* INCLUDE_SL811H */
};
```

Currently, the physical memory and the virtual memory of SL811HS has mapped to the same location, SL811H_MEMORY_START (defined in the file wrSbc405gp.h and it has a value of 0x7000000). This can be changed to accommodate different development platforms.

3) Pass the configuration (address and interrupt) information into the driver entry function
To configure the address and the interrupt of the SL811HS driver, you must fill the configuration data structure SL811_IO_CFG. After the data structure has been filled, it would pass to the usbdHcdAttach() routine as shown in the following code segment:

```c
SL811 IO CFG sl811IOCf = {
    SL811H_IO_ADDR,
    SL811H_IO_ADDR_DATA, SL811H_INT_VEC,
    SL811H_INT_LVL};
GENERIC_HANDLE *pToken;
usbdHcdAttach (usbHcdSl811hExec, &sl811IOCf, pToken);
```

Tested devices
• USB keyboard
• USB keyboard with built in hub
• USB mouse
• USB printer
• ISD315 USB mass storage device
• YAMAHA YST-MS35D USB Speaker
• USB hub

Performance
A performance test has been conducted using the ISD315 USB mass storage device. The driver has achieved a throughput of 410 Kilobytes per second.

Limitations
Please refer to the SL811HS VxWorks Driver Errata for detail information.
Definitions

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSP</td>
<td>VxWorks board support package</td>
</tr>
<tr>
<td>HC</td>
<td>USB host controller</td>
</tr>
<tr>
<td>HCD</td>
<td>USB host controller driver</td>
</tr>
<tr>
<td>IRP</td>
<td>USB IRP Input/Output request packet</td>
</tr>
<tr>
<td>SBC</td>
<td>Single board computer</td>
</tr>
<tr>
<td>SL811HS</td>
<td>SL811HS Cypress USB 1.1 host controller chip</td>
</tr>
<tr>
<td>SOF</td>
<td>USB start of frame</td>
</tr>
<tr>
<td>URB</td>
<td>USB request block</td>
</tr>
<tr>
<td>USBD</td>
<td>USB host driver</td>
</tr>
<tr>
<td>usbHcdSl811hsLib</td>
<td>USB HCD for SL811HS</td>
</tr>
</tbody>
</table>

References

- VxWorks USB Developer’s Kit 1.1 Release Notes (available at http://www.windriver.com/windsurf)
- SL811HS Application Notes (available at http://www.cypress.com)
- SL811HS VxWorks Driver Errata (available at http://www.cypress.com)
## Document Revision History

<table>
<thead>
<tr>
<th>Revision #</th>
<th>Date</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>4/18/02</td>
<td>Initial version</td>
</tr>
<tr>
<td>1.1</td>
<td>6/20/02</td>
<td>Format the document into two columns</td>
</tr>
</tbody>
</table>