

Q1.

The screenshot displays a C++ program and its execution output. The program, 'probe_PCI_bus', is a simple test for the PCI bus. The output shows a list of PCI devices, including the Intel Core i7-920M processor and the Intel HM77 Express Chipset. The development environment includes a code editor with the source code, an outline pane, and a console window showing the program's execution.

```

VxWorks6x_172.21.74.31@RUTVIJ-HP - Host Shell
Version 3.2

Copyright (c) 1995-2009 Wind River Systems, Inc.

C++ Constructors/Destructors Strategy is AUTOMATIC

-> probe_PCI_bus

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

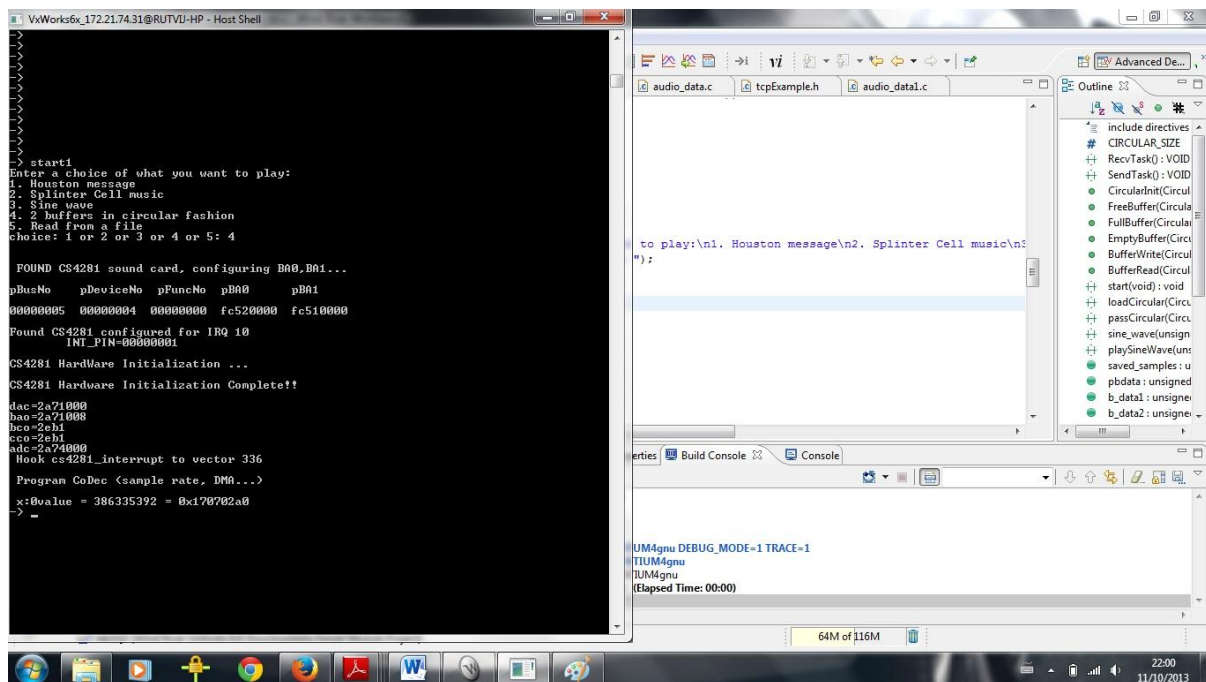
vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x06000000
0x00000086 0x00002412 0x00000000 0x00000011 0x00000000 0x06000000
0x00000086 0x0000241c 0x00000000 0x0000001c 0x00000000 0x06000000

vendorID, deviceID, Bus #, Device#, Func #, u.classCode
0x00000086 0x00002570 0x00000000 0x00000000 0x00000000 0x06000000
0x00000086 0x00002572 0x00000000 0x00000002 0x00000000 0x0
```

The Cirrus Logic CS4281 Sound card is attached to bus number 5 as shown in the above screenshot.

Q2.



The screenshot displays a VxWorks development environment. On the left, a terminal window titled 'VxWorks6x_172.21.74.31@RUTVJ-HP - Host Shell' shows the execution of a program. The program prompts the user to 'start!' and then 'Enter a choice of what you want to play:'. The user has selected option 1, 'Houston message'. The program output shows the configuration of a CS4281 sound card, including bus, device, and function numbers, and the initialization of hardware. The program then prints the sample rate and DMA address. On the right, a code editor shows the source files 'audio_data.c', 'tcpExample.h', and 'audio_data1.c'. The 'audio_data.c' file contains the implementation of the 'playSound()' function, which is called internally for the first time. The console window at the bottom shows the output of the program, including the sample rate and DMA address.

playSound() is the API which I created to pass buffers to the driver. It is called internally for the first time. The buffers are passed in the form of choices earlier displayed on running the program. The API playSound() can be called dynamically to change the audio being played.

Q3.

Circular buffer is implemented to pass more than one buffer in a circular fashion. I successfully applied another approach to this problem in which the data to be played is extracted from a text file stored in the /tgtsvr/filename.txt location and the file accessed and played in a circular fashion.

Q4.

The API playSineWave () can be called dynamically to play the particular frequency sine wave. The change in the frequency occurs dynamically without interruption with an instantaneous change in the audible frequency.

LAB 3 Sign Off Sheet ECEN 4/5623

Name: RUTVIJ KARKHANIS

4623 / 5623 (Circle One)

Question 3:

Probing the PCI bus to find all devices

☐ Average ☐ Good ☒ Excellent

Question 4:

a) Sound driver can play the sample sound correctly

☐ Average ☐ Good ☒ Excellent

b) API implementation

☐ Average ☐ Good ☒ Excellent

Question 5: Playing multiple sounds in a circular fashion

☐ Average ☐ Good ☒ Excellent

Grad Question:

a) Able to play 'sine' waveform correctly

☐ Average ☒ Good ☐ Excellent

b) Allow to change the frequency of the sine wave dynamically

☐ Average ☐ Good ☒ Excellent

Signature and Date:-

Soyth 11/10/13

Comments:

Used circular buffer for question 5