PSoC 4 Lab 4: Mixed-Signal Design Lab Manual

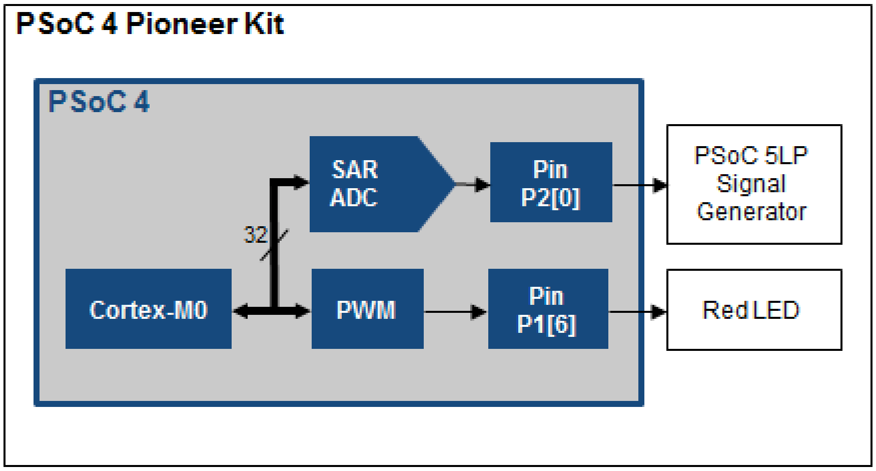
Objectives

1. Use the on-board PSoC 5LP as a signal generator for the PSoC 4
2. Measure the analog signal generator voltage using the PSoC 4 Sequencing SAR ADC Component
3. Modulate the red LED using a PWM controlled by the Sequencing SAR ADC output

Requirements	Details
Hardware	CY8CKIT-042 PSoC 4 Pioneer Kit
Software	PSoC Creator 3.0 SP1, PSoC Programmer, Bootloader Host Tool
Firmware	PSoC4Lab4
Components used	Sequencing SAR ADC, PWM, Pin Components

Block Diagram

Figure 1: Lab 4 Block Diagram



Theory

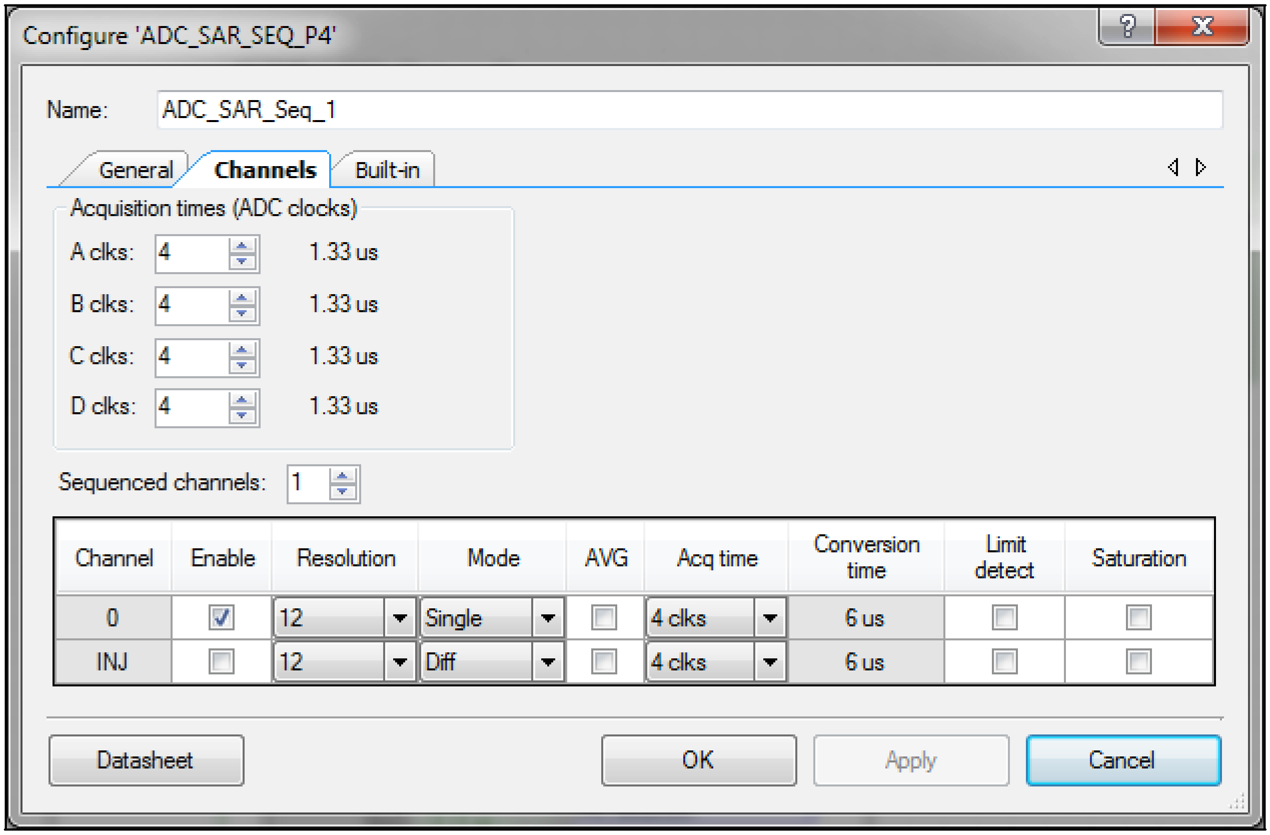
The goal of this lab is to learn how to use PSoC 4's Sequencing SAR ADC to measure an analog signal, and use the LEDs to display that measurement. PSoC 4's Sequencing SAR ADC is capable of making 12-bit analog measurements at a rate up to 1 Msps. The hardware sequencing engine makes it possible to sequence multiple inputs to the ADC without CPU intervention.

The lab will also demonstrate how to use the PSoC 5LP device (originally used as the programmer and debugger) on the PSoC 4 Pioneer Kit to implement additional behavior. In this case, we will use a WaveDAC Component on the PSoC 5LP to generate two waveforms that we will read with the PSoC 4 Sequencing SAR ADC.

Procedure: Firmware

1. Close any open workspaces or files.
2. Open the PSoC4Lab4.cywrk workspace.
3. Open the project's schematic by double-clicking on the "TopDesign.cysch" file in the Workspace Explorer. Note that in this schematic, we've included three PWMs and pins, along with the "LED_RGB.c" library to allow for easy driving of tri-colored LEDs.
4. In the Component Catalog, under the "Analog->ADC" category, select the "Sequencing SAR ADC" Component and drag it into the schematic.
5. Open the "ADC_SAR_Seq_1" Component Configuration tool by double-clicking on the Component. Click on the "Channels" tab to configure the ADC inputs. Set the "Sequenced channels" input to 1 to remove the unused inputs. Change the "Mode" of channel 0 to "Single." The configuration window should look like the one shown in Figure 2.

Figure 2: ADC SAR Sequencer Channel Configuration



Name: ADC_SAR_Seq_1

Acquisition times (ADC clocks)

A clks: 4 1.33 us

B clks: 4 1.33 us

C clks: 4 1.33 us

D clks: 4 1.33 us

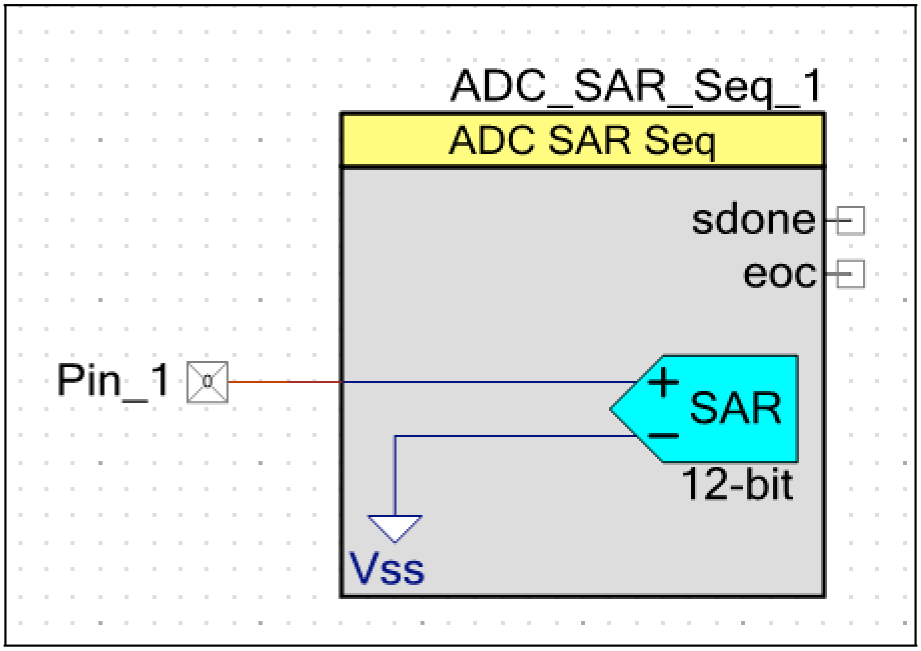
Sequenced channels: 1

Channel	Enable	Resolution	Mode	AVG	Acq time	Conversion time	Limit detect	Saturation
0	☑	12	Single	□	4 clks	6 us	□	□
INJ	□	12	Diff	□	4 clks	6 us	□	□

6. Press “OK” to close the configuration window. The ADC_SAR_Seq_1 Component should now look like the one shown in Figure 3.

7. In the Component Catalog, under the “Ports and Pins” category, select the “Analog Pin” Component, drag it into the schematic, and connect it to the positive input of the SAR ADC. The schematic should look like the one shown in Figure 3.

Figure 3: SAR ADC and Pin Appearance



8. Open the design wide resources by double-clicking the “Lab 4 ADC.cydwr” file. Navigate to the “Pins” tab. Using the “Port” dropdown, map the “ADC_SAR_Seq_1:Bypass” pin to P1[7], and “Pin_1” to P2[0].

9. Open the “main.c” file by double-clicking on it in the Workspace Explorer.

10. Replace the “Change1” line with the ADC start code, shown in Code 1.

Code 1: Lab 4 “Change1” ADC Start Code

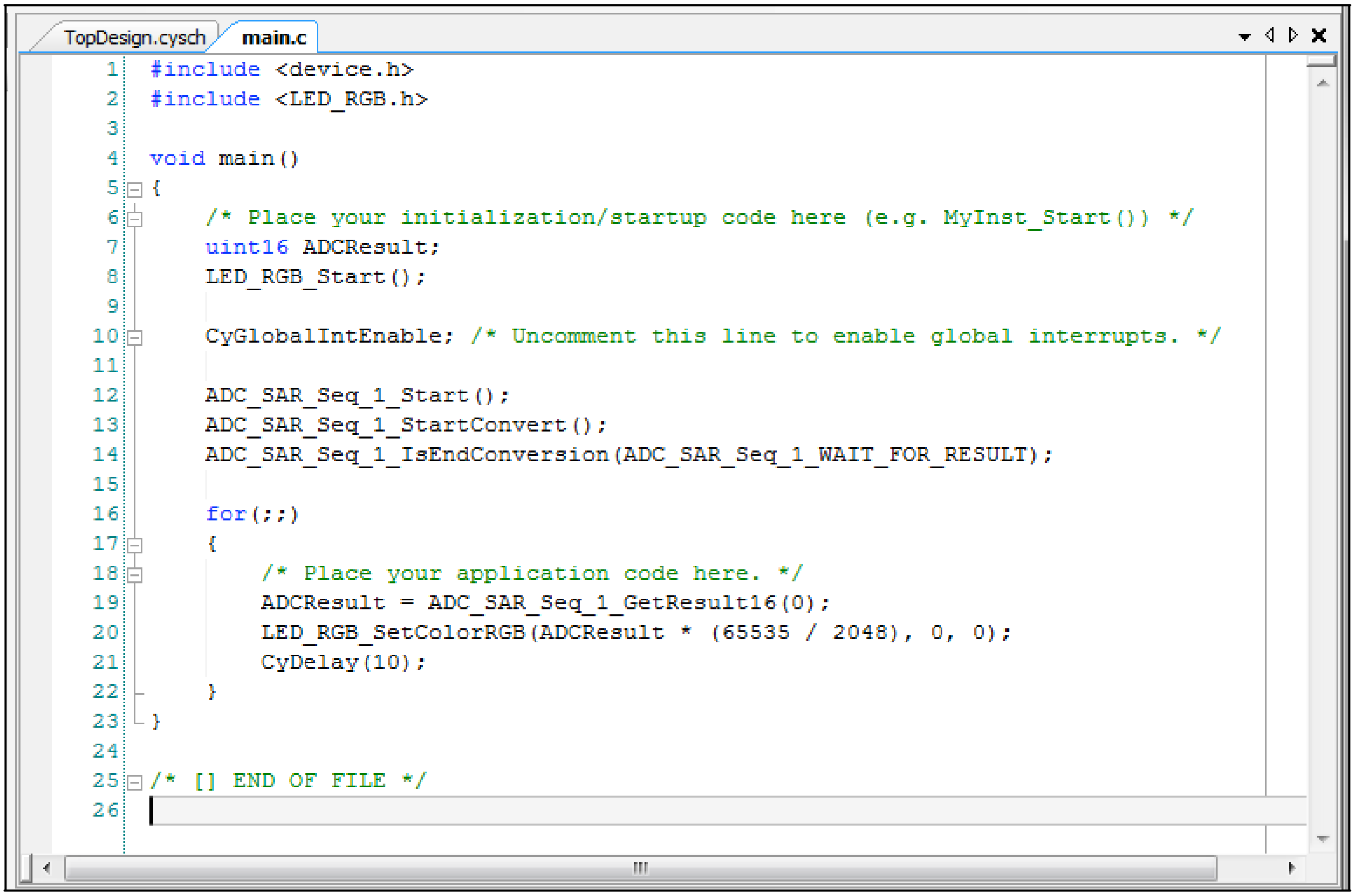
```
 ADC_SAR_Seq_1_Start(); // Start the ADC hardware block

 ADC_SAR_Seq_1_StartConvert(); // Start ADC conversion

 ADC_SAR_Seq_1_IsEndConversion(ADC_SAR_Seq_1_WAIT_FOR_RESULT);
// Wait until conversion is completed
```

11. Replace the “Change2” line with the ADC GetResult API, shown in Code 2. The entire main.c should look like that shown in Figure 4.

Code 2: Lab 4 “Change 2” ADC Get Result API

```
ADCResult = ADC_SAR_Seq_1_GetResult16(0); // Get the ADC result
```

Figure 4: Lab 4 Solution “main.c”

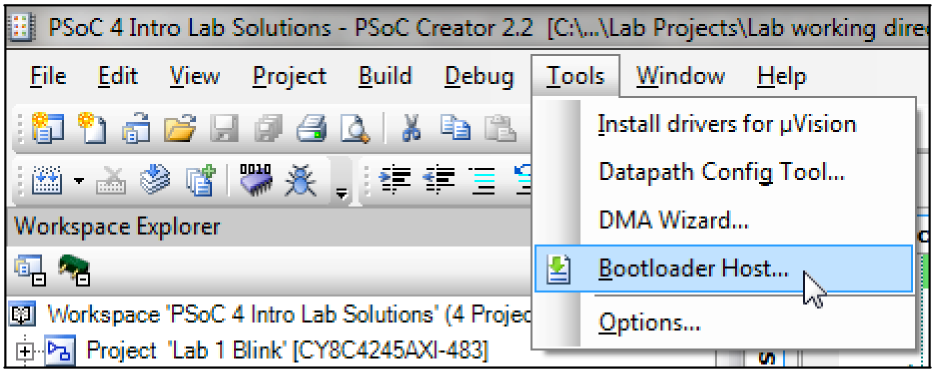


```
#include <device.h>
#include <LED_RGB.h>

void main()
{
    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    uint16 ADCResult;
    LED_RGB_Start();

    CyGlobalIntEnable; /* Uncomment this line to enable global interrupts. */

    ADC_SAR_Seq_1_Start();
    ADC_SAR_Seq_1_StartConvert();
    ADC_SAR_Seq_1_IsEndConversion(ADC_SAR_Seq_1_WAIT_FOR_RESULT);

    for(;;)
    {
        /* Place your application code here. */
        ADCResult = ADC_SAR_Seq_1_GetResult16(0);
        LED_RGB_SetColorRGB(ADCResult * (65535 / 2048), 0, 0);
        CyDelay(10);
    }
}

/* [] END OF FILE */
```

12. Press the “Program” button on the PSoC Creator toolbar to build the project and program your kit. After programming, The Red LED will stay lit at a constant intensity, because the voltage on the ADC input pin P2[0] is not varying. Next we will program the PSoC 5LP with a program to make it generate a waveform that will drive the PSoC 4 SAR ADC input.

13. **Bootloading the onboard PSoC 5LP device:** In PSoC Creator, under the “Tools” menu, select the “Bootloader Host…” option. This is shown in Figure 5.

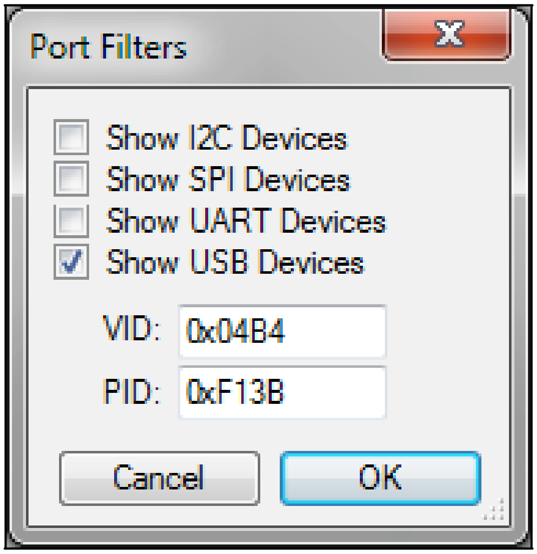
Figure 5: PSoC Creator Bootloader Host Menu Option


PSoC 4 Intro Lab Solutions - PSoC Creator 2.2 [C:\...\Lab Projects\Lab working dire
File
Edit
View
Project
Build
Debug
Tools
Window
Help
Install drivers for µVision
Datapath Config Tool...
DMA Wizard...
Bootloader Host...
Options...
Workspace Explorer
Workspace 'PSoC 4 Intro Lab Solutions' (4 Projec
Project 'Lab 1 Blink' [CY8C4245AXI-483]

14. Click the “Open File” button in the upper left of the Bootloader Host GUI. Select the file “PSoC5LP_WaveDAC.cyacd” in the project template directory.

15. In the upper right corner of the “Ports” field in the Bootloader Host window, press the “Filters…” button. Ensure that “Show USB Devices” is checked, and the VID and PID values are 0x04B4 and 0xF13B, respectively, as shown in Figure 6.. Close the window by pressing the “OK” button.

Figure 6: Bootloader Host GUI Port Filter Settings



☐ Show I2C Devices
☐ Show SPI Devices
☐ Show UART Devices
☑ Show USB Devices

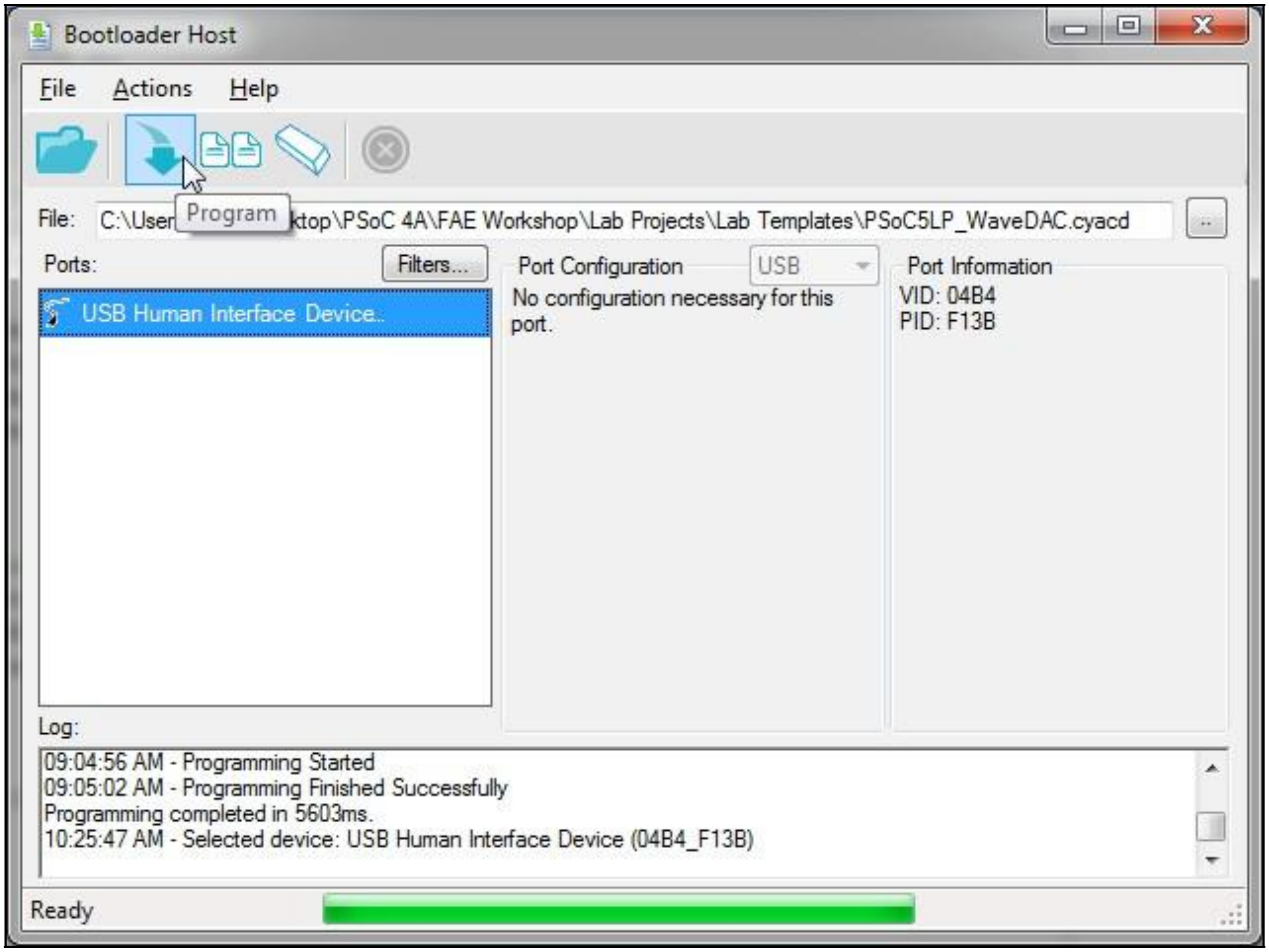
VID: 0x04B4
PID: 0xF13B

16. Unplug the PSoC 4 Pioneer Kit from USB. Hold down the reset switch, SW1, which is in the lower left corner of the board, while plugging the USB cable back in. Release the

button when the status LED D1 in the upper left corner of the board starts blinking. This indicates that the board is ready to be bootloaded.

17. At this point, the kit should appear in the “Ports” field in the Bootloader Host. Select it, and press the “Program” button, as shown in Figure 7.

Figure 7: Bootloader Host Tool Ready to Bootload the PSoC 5LP Device

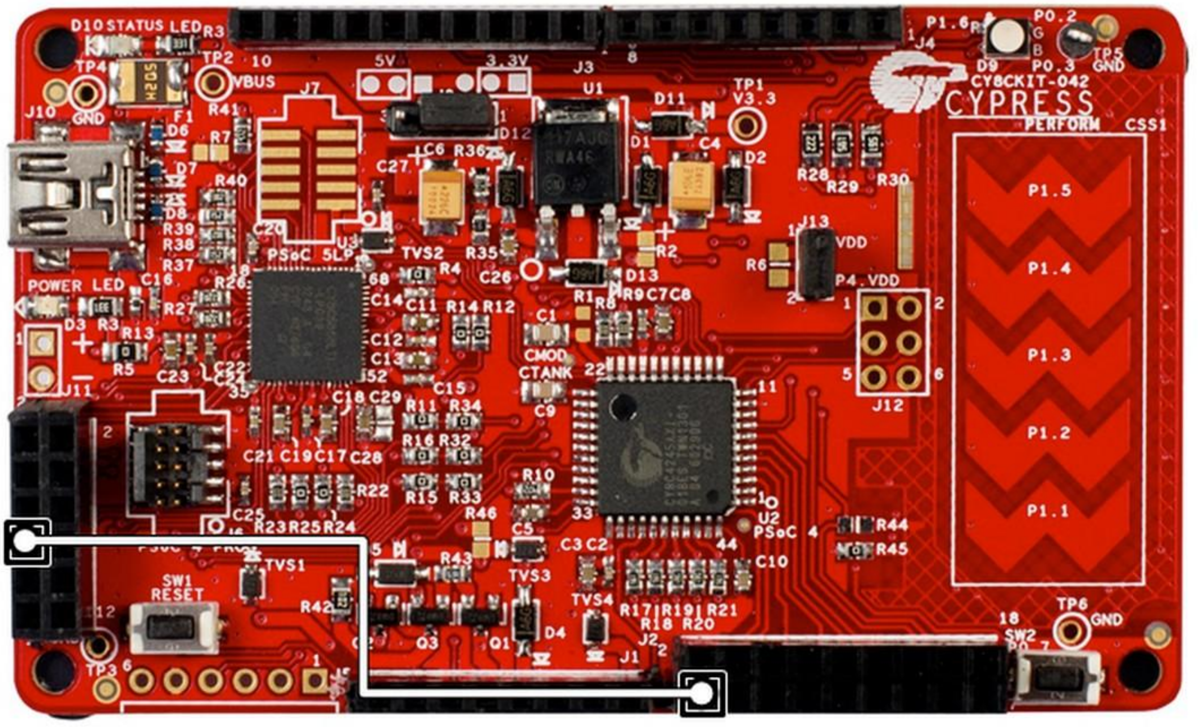

Bootloader Host
File Actions Help
Program
File: C:\User ktop\PSoC 4A\FAE Workshop\Lab Projects\Lab Templates\PSoC5LP_WaveDAC.cyacd
Ports:
Filters...
USB Human Interface Device..
Port Configuration
USB
No configuration necessary for this port.
Port Information
VID: 04B4
PID: F13B
Log:
Ready

```
09:04:56 AM - Programming Started
09:05:02 AM - Programming Finished Successfully
Programming completed in 5603ms.
10:25:47 AM - Selected device: USB Human Interface Device (04B4_F13B)
```

18. The bootloader results will appear in the “Log” at the bottom of the window. The PSoC 5LP should now be programmed to generate an analog waveform. All that remains is to connect it to our PSoC 4 ADC input. Before moving on to the next step, ensure to close the Bootloader Host tool.

19. **PSoC 4 Pioneer Kit Hardware Setup:** Using one of the wires shipped with the kit, connect PSoC 4 pin P2[0] (connector J2 pin 1) with PSoC 5LP pin P3[6] (connector J8 pin 7). This is shown in Figure 8.

Figure 8: PSoC 4 Pioneer Kit PSoC 5LP WaveDAC Connection

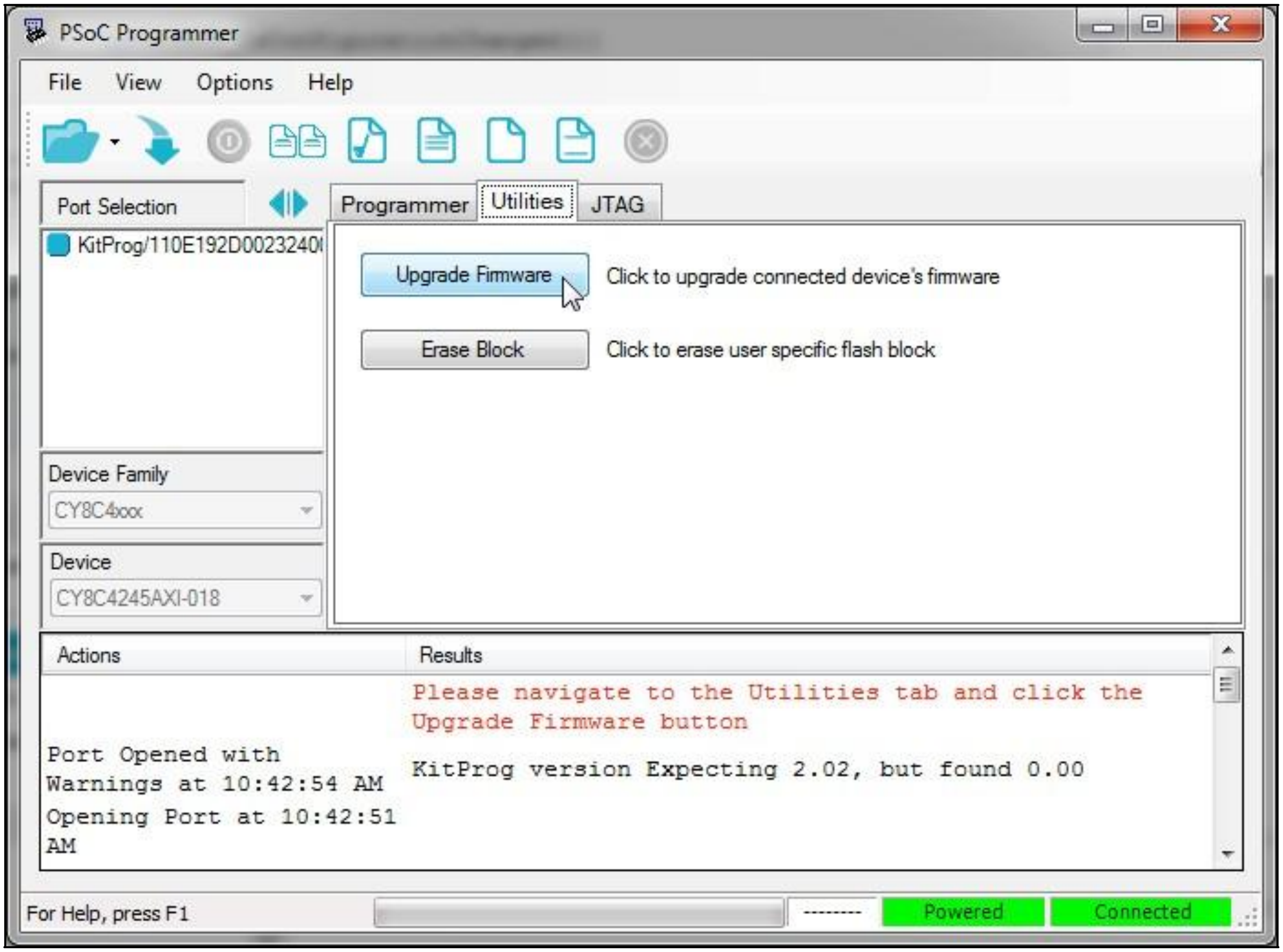


20. At this point, the red LED should be pulsing in and out in a sine wave pattern at a frequency of 1 Hz.

21. Briefly press SW1 to change the WaveDAC waveform from sine wave to sawtooth wave. The red LED should now vary as a sawtooth wave. Pressing the button again will toggle between waveforms.

22. To restore the factory firmware to the PSoC 5LP, open PSoC Programmer, which is located in the “Cypress->PSoC Programmer 3.18” location in the Start menu.
23. When PSoC Programmer is opened, it will detect the firmware installed on the PSoC 5LP and report it. It should show version “0.00” with our custom WaveDAC firmware. It will also instruct you to upgrade the firmware to get to the most recent official version. This is shown in Figure 9.
24. To restore the factory firmware, navigate to the “Utilities” tab and press the “Upgrade Firmware” button. This is shown in Figure 9. The PSoC 5LP will now be programmed with the factory firmware, and will no longer produce the WaveDAC outputs. To bootload the custom firmware again, repeat steps 13. through 18.

Figure 9: PSoC Programmer Firmware Update Interface



KitProg/110E192D0023240

Click to upgrade connected device's firmware

Click to erase user specific flash block

Device Family: CY8C4xxx

Device: CY8C4245AXI-018

```
Port Opened with            KitProg version Expecting 2.02, but found 0.00
Warnings at 10:42:54 AM
Opening Port at 10:42:51
AM
```

Please navigate to the Utilities tab and click the Upgrade Firmware button

Conclusion

You have successfully implemented a Sequencing SAR ADC in PSoC 4, reading analog values generated by the PSoC 5LP on the PSoC 4 Pioneer Kit.

Stretch Goals

1. Modulate another LED characteristic instead of red intensity
    a. In this lab, we varied the red LED intensity based on the input to make the two waveform shapes more distinct.
    b. You could use the other LED_RGB functions to vary different LED characteristics, like hue, saturation or value. The other useful LED_RGB module functions are shown in Code 3.

Code 3: LED_RGB Module Functions

```
void LED_RGB_SetColorRGB(uint16 redIntensity,
     uint16 greenIntensity, uint16 blueIntensity); // Sets the intensity of
     each individual color in the RGB LED array.

void LED_RGB_SetColorCircle(uint16 hue); // Sets the color of the RGB
     LEDs with full brightness

void LED_RGB_SetColorCircleHSV(uint16 hue,
     uint16 saturation, uint16 value); // Sets the hue, saturation and value
     of the RGB LEDs allowing full white (all on) and full black (all off)
```

2. Transmit the ADC readings from PSoC 4 back to the PC via UART
    a. In this lab, we demonstrated the ADC readings by driving the red LED at different intensities.
    b. We could integrate some of the functionality demonstrated in lab 3 and use the UART to transmit readings back to the PC.
    c. The custom PSoC 5LP WaveDAC firmware still implements a USB-UART bridge.

Document Revision History

Revision	By	Description
**	MAXK	First Release
*A	GUL	Updated formatting and other minor corrections
*B	PKX	Updated to be stand alone lab