PSoC 4 Lab 3: CapSense UART Lab Manual

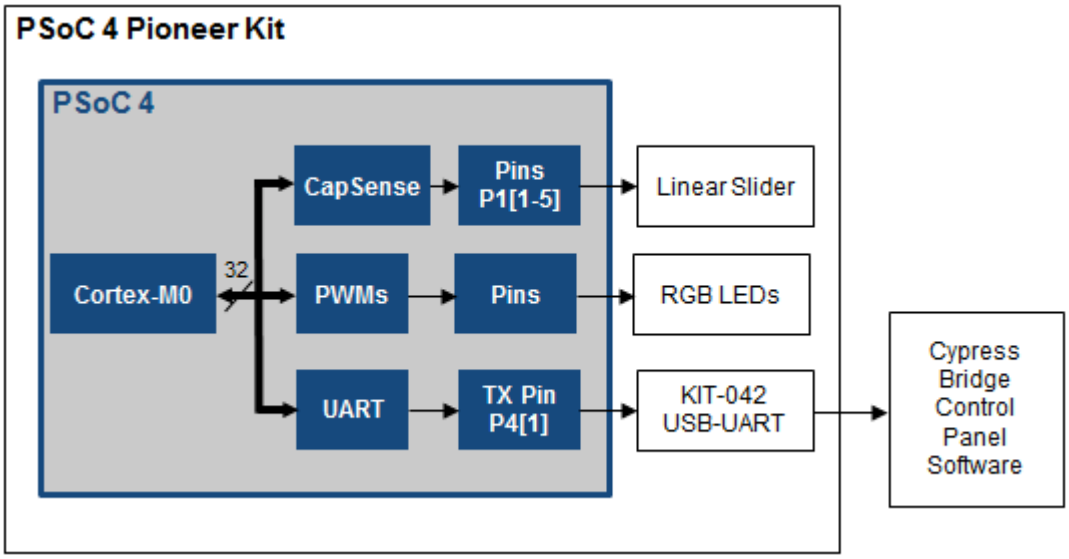
Objectives

1. Implement a multi-segment slider using the CapSense Component
2. Debug the design with the UART Component and the USB-UART bridge on the PSoC 4 Pioneer Kit

Requirements	Details
Hardware	CY8CKIT-042 PSoC 4 Pioneer Kit
Software	PSoC Creator 3.0 SP1, Bridge Control Panel 1.10 or higher
Firmware	PSoC4Lab3
Components used	CapSense, UART, PWM, Pins

Block Diagram

Figure 1: Lab 3 Block Diagram



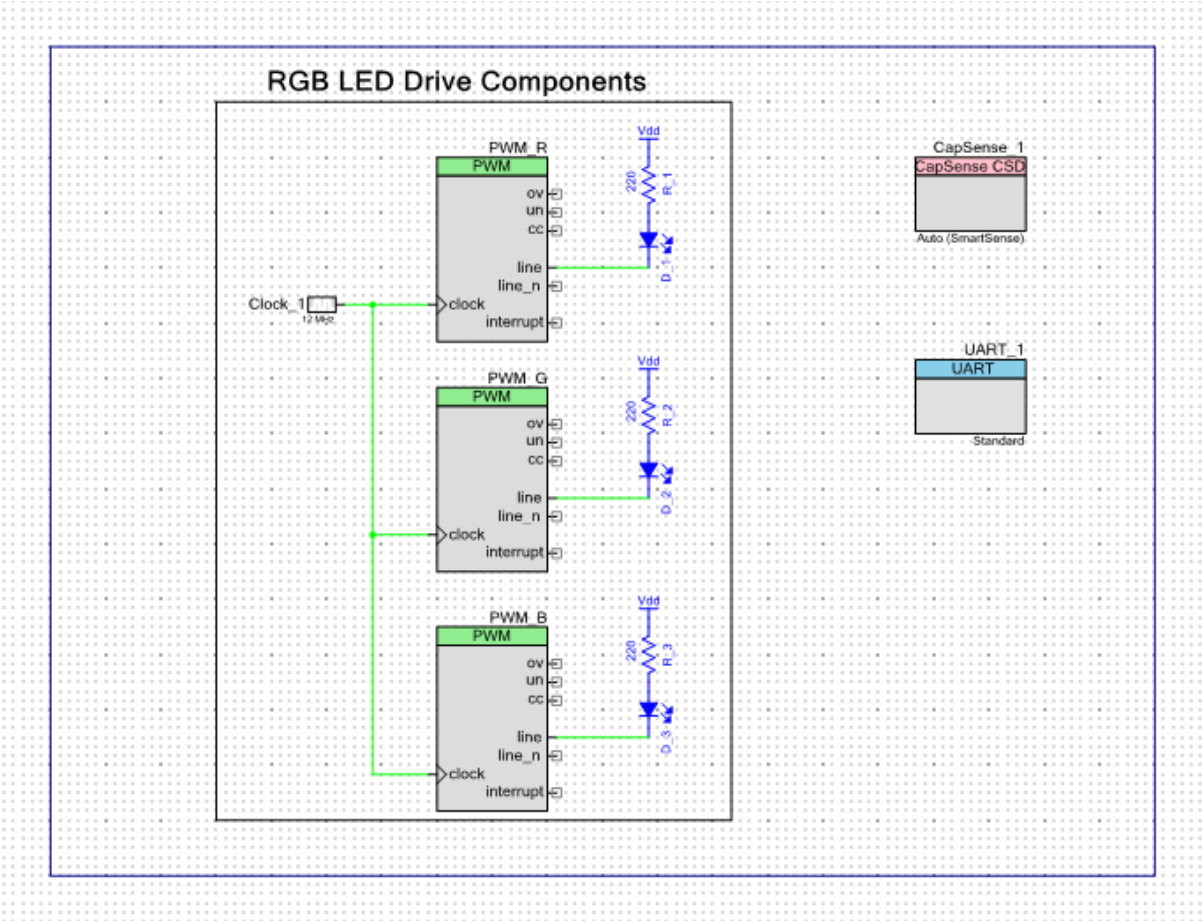
Theory

The goal of this lab is to learn how to use the PSoC 4 CapSense Component to sense the position of a finger on a linear slider, and use the LEDs to display that position. PSoC 4's dedicated CapSense hardware measures finger presence by measuring the change in capacitance when a finger is introduced to a capacitive sensor, normally a small copper shape on a PCB. The lab will also demonstrate how to use a UART to transmit the position to a PC by using the USB-UART functionality of the PSoC 4 Pioneer kit, and the Bridge Control Panel software that ships with PSoC Programmer. PSoC 4 contains two (2) Serial Communication Blocks (SCBs) that can implement UART, SPI, or I^2C interfaces. We will use the UART Component for this lab.

Procedure: Firmware

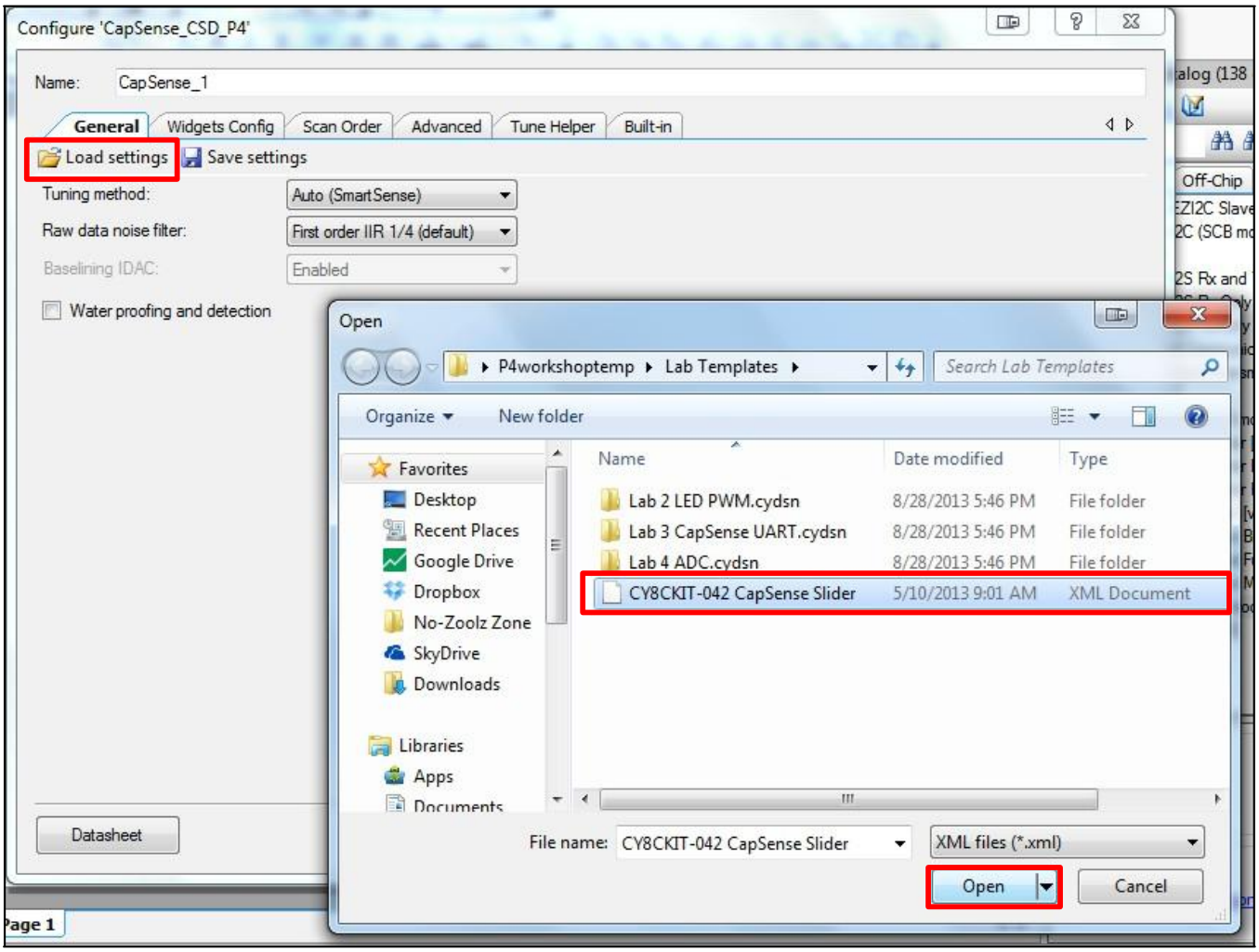
1. Close any open workspaces of files
2. Open the PSoC4Lab3CapSense.cywrk
3. Open the project's schematic by double-clicking on the "TopDesign.cysch" file in the Workspace Explorer. Note that in this schematic, we've included three PWMs and Pin Components, along with the "LED_RGB.c" firmware library to allow for easy driving of tri-colored LEDs.
4. In the Component Catalog, under the "CapSense" category, select the "CapSense CSD" Component, and drag it into the schematic.
5. In the Component Catalog, under the "Communications" category, select the "UART (SCB mode)" Component, and drag it into the schematic. The schematic should look like the one shown in Figure 3.

Figure 3: Schematic With CapSense and UART Placed



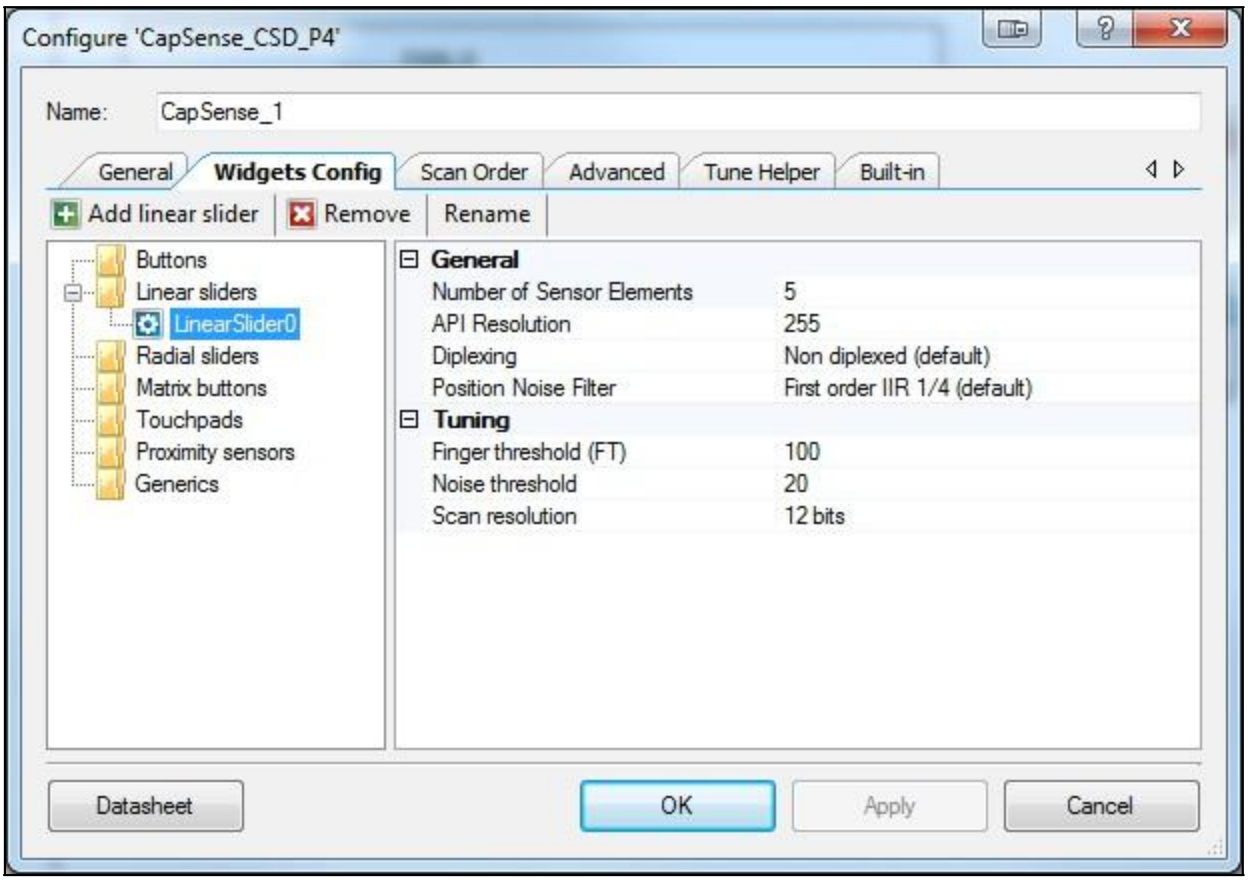
6. Open the CapSense Component Configuration tool by double-clicking on the Component. Click “Load settings” to automatically configure the Component for the kit’s CapSense slider. Select the file “CY8CKIT-042 CapSense Slider.xml” in the lab template workspace directory and press “Open.” This step is illustrated in Figure 4 below.

Figure 4: Load CapSense Component Settings From the “.xml” File



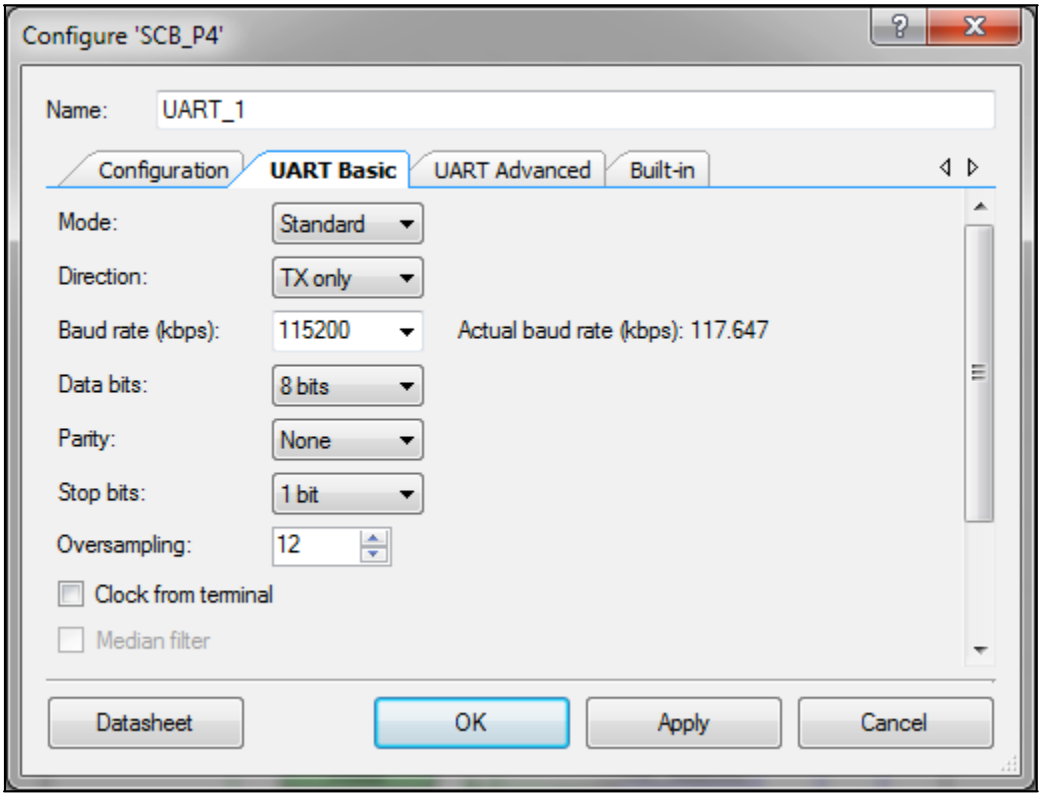
7. Navigate to the “Widgets Config” tab, and click on the “LinearSlider0” widget to examine its settings. These settings are used to configure the CapSense sensors, and are described in detail in the CapSense Component datasheet. Press the “OK” button when you are done looking at the settings. This window is shown in Figure 5.

Figure 5: CapSense Widget Configuration Tab



8. Open the UART Component Configuration tool by double-clicking on the UART Component. Navigate to the “UART Basic” tab to configure the UART settings. Change the value of the “Direction” dropdown to “TX only” to setup the UART for uni-directional communication. Press “OK” to close the window and apply changes. The UART Component correctly configured is shown in Figure 6.

Figure 6: UART Configuration Tool



9. Double-click on the "Lab 3 CapSense UART.cydwr" file to open the design wide resources interface.

10. On the "Pins" tab, assign the "UART_1:tx" pin to P4[1], the "CapSense_1:Cmod" pin to P4[2], and the "CapSense_1:Sns[0-4]" pins to P1[1] through P1[5], in either ascending or descending order. The completed pin assignment should look something like that shown in Figure 7.

Figure 7: CapSense and UART Pin Mapping

Alias	Name	Port	Pin	Lock
Cmod	\CapSense_1:Cmod\	P4[2] SCB0:SPI:SCLK	22	☑
LinearSlider0_e0__LS	\CapSense_1:Sns[0]\	P1[1] TCPWM2:N	38	☑
LinearSlider0_e1__LS	\CapSense_1:Sns[1]\	P1[2] TCPWM3:P	39	☑
LinearSlider0_e2__LS	\CapSense_1:Sns[2]\	P1[3] TCPWM3:N	40	☑
LinearSlider0_e3__LS	\CapSense_1:Sns[3]\	P1[4]	41	☑
LinearSlider0_e4__LS	\CapSense_1:Sns[4]\	P1[5]	42	☑
	\UART_1:tx\	P4[1] SCB0:I2C:SDA, SCB0:SPI:MISO, SCB0:UART:TX	21	☑
	Pin_BlueLED	P0[3]	27	☑
	Pin_GreenLED	P0[2] SCB0:SPI:SS3	26	☑
	Pin_RedLED	P1[6]	43	☑

11. In the "Workspace Explorer", double-click the "main.c" file to open it in the code editor.

12. Replace the "Change1" line with the initialization code, shown in Code 1.

Code 1: Lab 3 "Change1" Initialization Code

```
CapSense_1_Start();  // Starts the CapSense hardware block
CapSense_1_InitializeAllBaselines(); // Initializes the CapSense baselines
UART_1_Start(); // Starts the Serial Communication hardware block
```

13. Replace the "Change2" line with the API to read the CapSense slider position, shown in Code 2.

Code 2: Lab 3 "Change2" CapSense Slider Position Read API

```
CapSensePosition = CapSense_1_GetCentroidPos(
CapSense_1_LINEARSLIDER0___LS);  // Reads the CapSense Slider
position
```

14. Replace the “Change3” line with the API to read the CapSense slider position, shown in Code 3. The entire “main.c” should look like that shown in Figure 8.

Code 3: Lab 3 “Change3” UART Put Char API

```
UART_1_UartPutChar(CapSensePosition); // Sends the CapSense slider position to the UART
```

Figure 8: Lab 3 Solution “main.c”

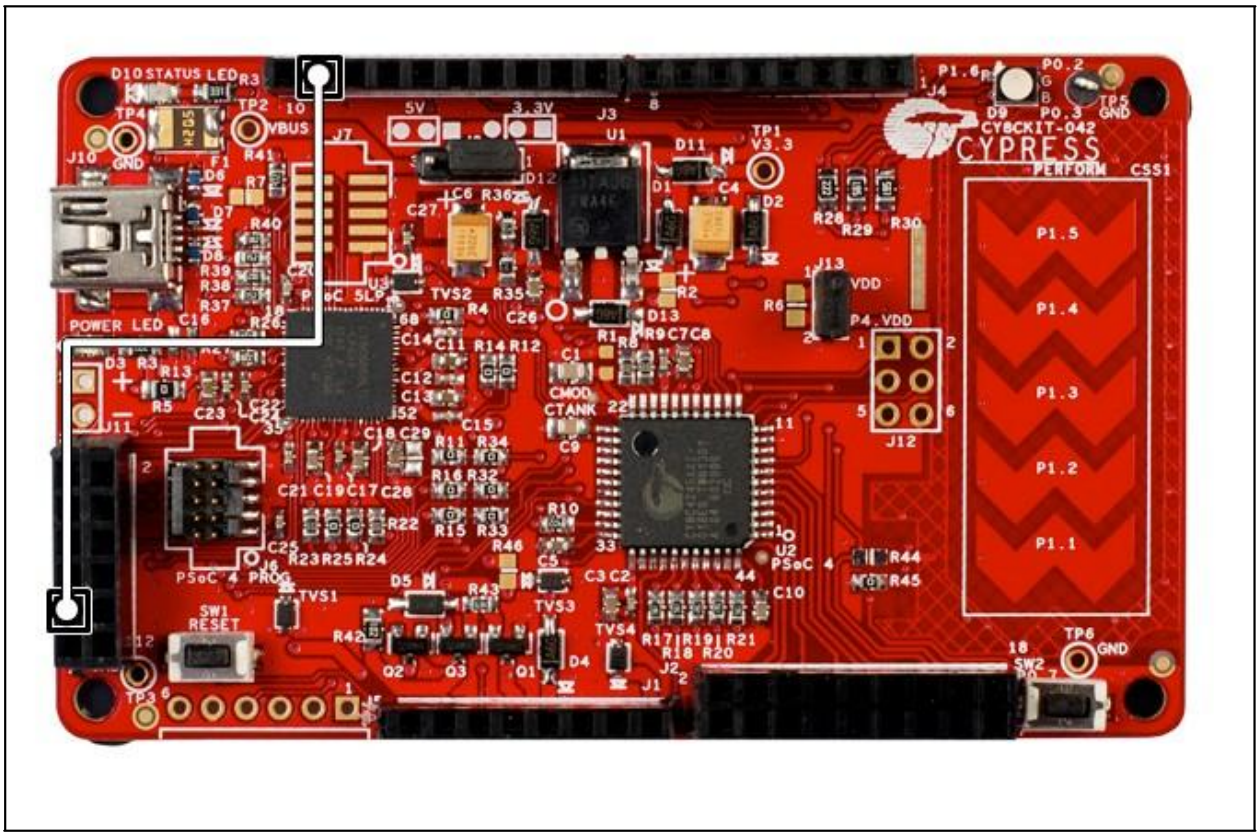
main.c | TopDesign.cysch | Lab 3 CapSense UART.cydwr

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

void main()
{
    /* Place your initialization/startup code here (e.g. MyInst_Start()) */
    uint16 CapSensePosition;
    LED_RGB_Start();
    CyGlobalIntEnable;
    CapSense_1_Start();
    CapSense_1_InitializeAllBaselines();
    UART_1_Start();

    /* CyGlobalIntEnable; */ /* Uncomment this line to enable global interrupts. */
    for(;;)
    {
        /* Place your application code here. */
        CapSense_1_UpdateEnabledBaselines();
        CapSense_1_ScanEnabledWidgets();
        while(CapSense_1_IsBusy() != 0);
        CapSensePosition = CapSense_1_GetCentroidPos(CapSense_1_LINEARSLIDER0__LS);
        LED_RGB_SetColorCircle(255 * CapSensePosition);
        UART_1_UartPutChar(CapSensePosition);
        CyDelay(10);
    }
}

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

15. Press the “Program” button on the PSoC Creator toolbar to build the project and program your kit. After programming, you should be able to change the color of the tri-color LED by moving your finger up and down along the CapSense Slider.

16. **PSoC 4 Pioneer Kit Hardware Setup:** Using one of the wires shipped with the kit, connect PSoC 4 pin P4[1] (connector [J3] pin 9) with PSoC 5LP pin P12[6] (connector J8 pin 9). This is shown in Figure 9.

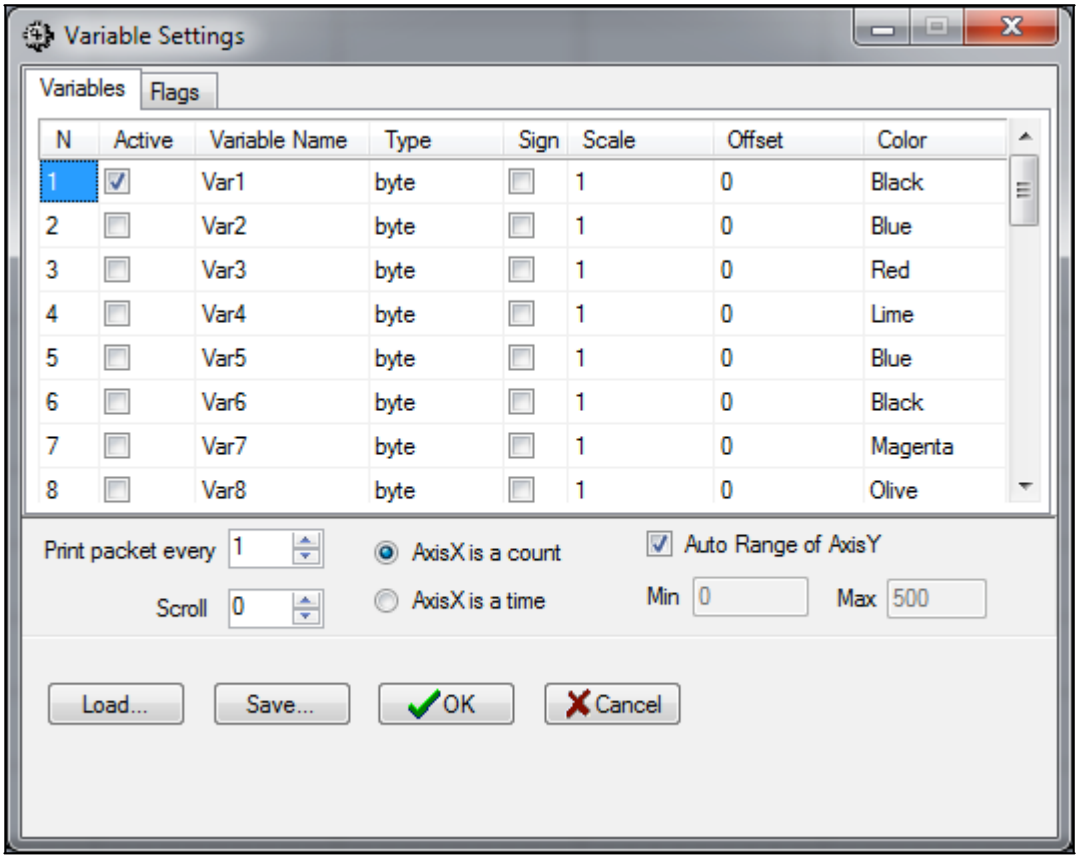
Figure 9: PSoC 4 Pioneer Kit UART TX Connection



17. Open the Bridge Control Panel software, which is in the Windows Start menu under “Cypress -> Bridge Control Panel.”

18. Click on the “Chart” menu, and select “Variable Settings.” Ensure that the first variable is named “Var1”, its “Active” checkbox is checked, the “Type” is “byte” and the “Sign” checkbox is unchecked. Press “OK” to close the window. This is shown in Figure 10.

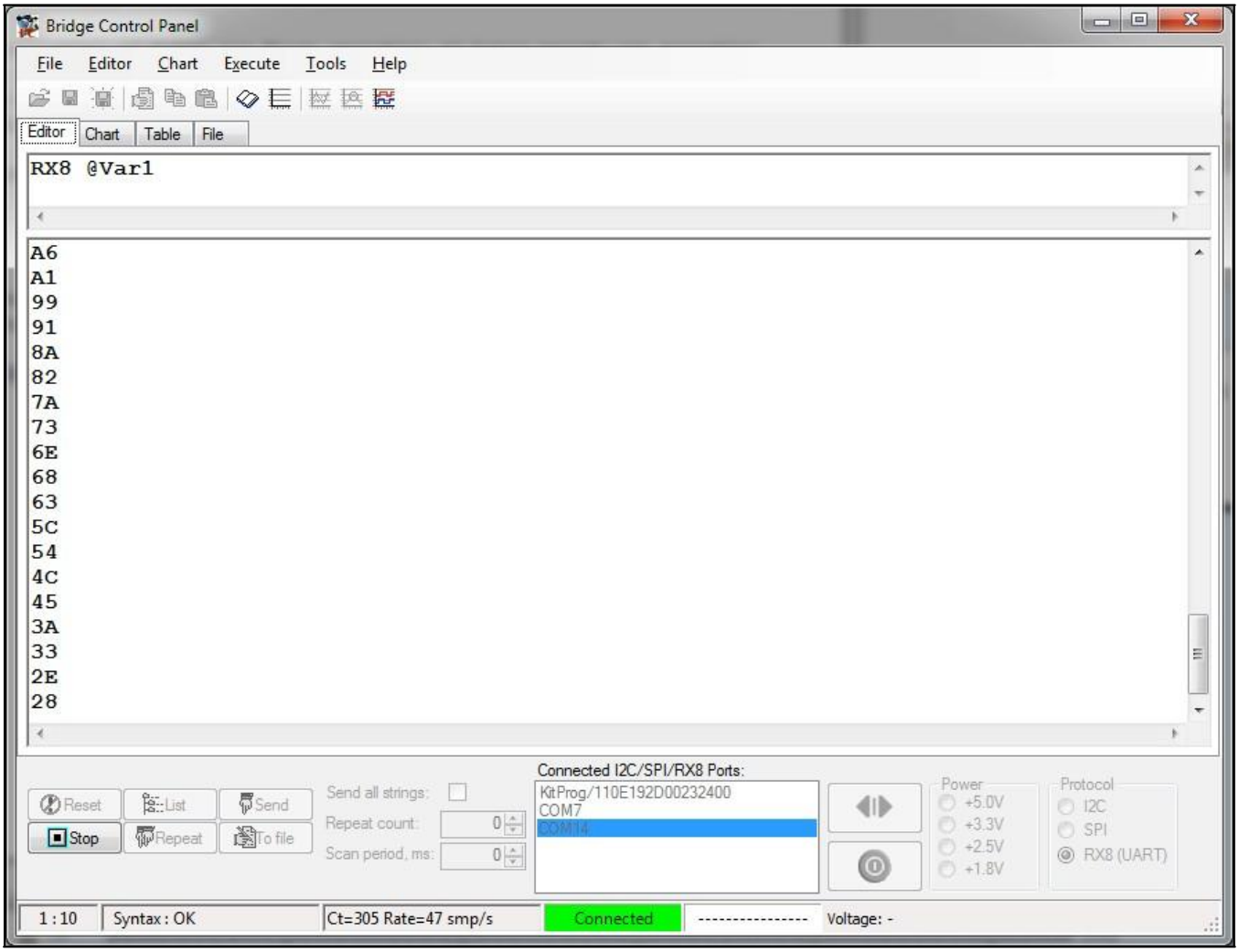
Figure 10: Bridge Control Panel Variable Settings Configuration



19. In the editor tab, select the PSoC 4 Pioneer Kit's UART-bridge COM port in the "Connected I^2C /SPI/RX8 Ports:" list at the bottom of the window. It is typically the highest numbered COM port in the list. The box below the ports list should turn green and say "Connected."

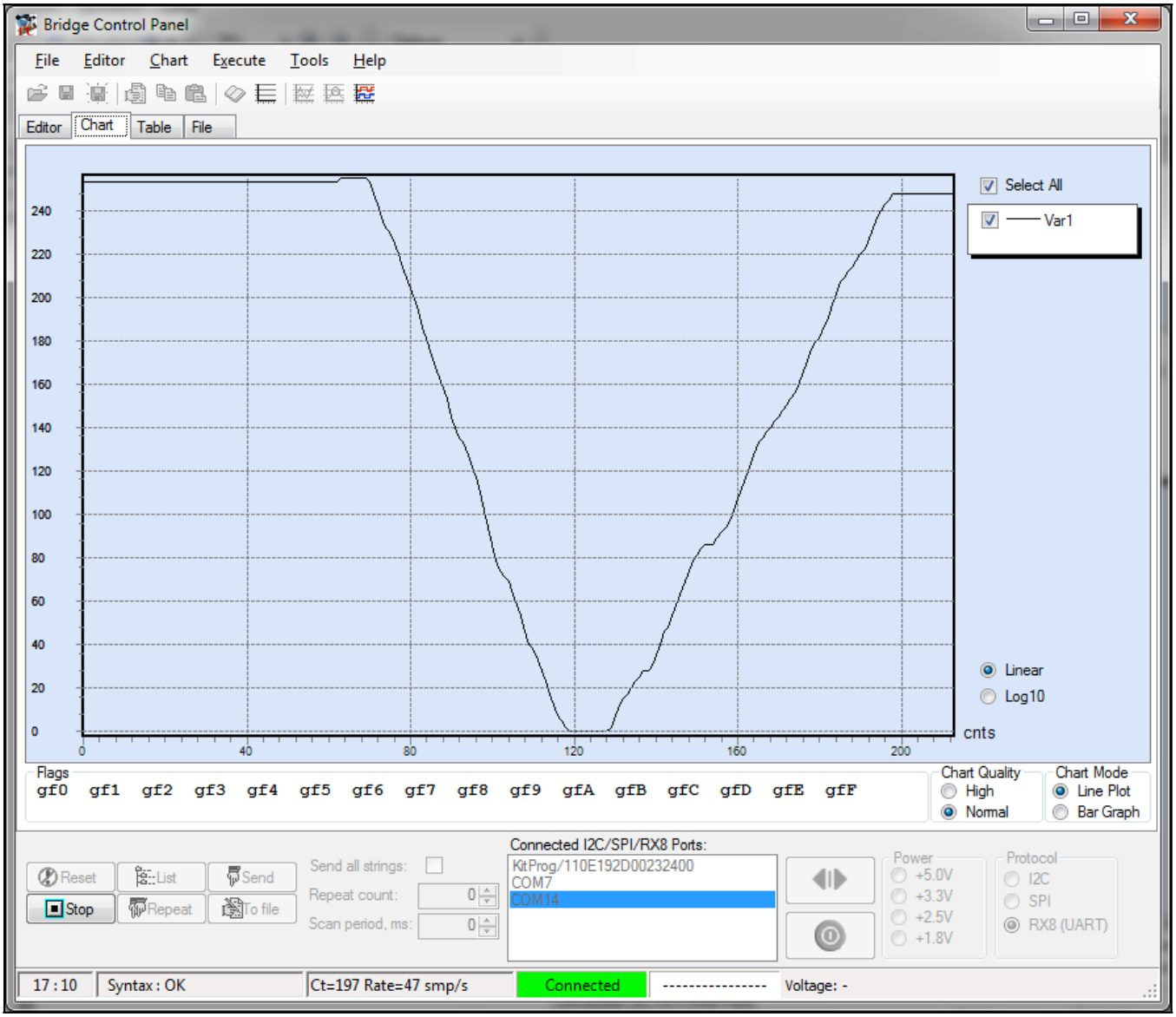
20. In the editor text box, enter the RX8 packet instruction "RX8 @Var1." With the cursor still on the line that you entered, press the "Repeat" button. Hex bytes should begin to appear in the console below the editor. Move your finger across the CapSense slider and watch the values change. It should look like the image shown in Figure 11.

Figure 11: Bridge Control Panel Data Transmission



21. Click on the “Chart” tab, and observe the values being charted over time. If you want to clear the graph and start over, Press the “Stop” button followed by the “Repeat” button. When sweeping your finger across the slider, the chart should similar to the one shown in Figure 12.

Figure 2: Bridge Control Panel Chart Interface



Conclusion

You have successfully implemented a CapSense slider with UART communication on your PSoC 4 Pioneer Kit. You have also learned how to use the Bridge Control Panel to communicate with the Pioneer kit, and graph the data.

Stretch Goals

1. Retain the last color on the LED when the finger is removed from the CapSense slider.
    a. In this lab, we don't examine the CapSense slider position for returning a "no finger" code, so the slider turns red when that code is returned.
    b. The "CapSense_1_GetCentroidPos()" API returns '0xffff' when no finger is detected. If this value is detected by firmware, the LED color can be left alone, and the last detected position's color will remain on the LEDs, even when the finger is removed.
2. Transmit the whole 16-bit variable "color."
    a. In this lab, we shifted the 16 bit variable down by 8 bits and transmitted only the most significant byte (MSB).
    b. We could transmit both bytes by masking them individually and using a control character to indicate coherency.
    c. In the Bridge Control Panel software, you will need to configure your variable to be an "int" instead of a "byte", receive both bytes of data, and interpret the packet including the control character. An RX8 command for this would look something like "RX8 [h=43] @1intColor @0intColor" if the control character was hex 0x43 (character 'C').
3. Implement bidirectional communications
    a. In this lab, we only implemented single direction communication from the PSoC 4 Pioneer Kit to the PC.
    b. Try communicating from the PC to the Pioneer Kit, perhaps setting the LED color using the Bridge Control Pane software.

Document Revision History

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