CapSense

My name is Alan Hawse and this is PSoC 101. One of the coolest things you can do with PSoC is to detect the presence of a finger by sensing capacitance. Put simply, PSoC offers the best capacitive sensing on the planet. And our CapSense component makes it really easy to use. All the Pioneer kits have five sensors which we can use individually as buttons or to create a slider. In this lesson we will show how CapSense provides a very sensitive slider with just these 5 sensors.

Make a copy of the UART project because we are going to report finger positions to the terminal emulator. Search for and add a CapSense component. Rename it and add a slider widget. Notice that, by default, it has the name LinearSlider0 and uses 5 sliders sensor elements. You can design your boards with more or less, depending upon the resolution you need. As you'll see, with 5 sensors we can easily deliver values from zero to one hundred.

Next we pick the pins in the DWR. You'll need to use the psoc101 web page for the pin selections for your Pioneer kit. First, select the CMod pin. CMOD is an external capacitor that stores charge for the CapSense block. Then the sensor pins.

In our C code we can delete everything except the UART Start function and the global interrupt enable macro. Like EZI2C the CapSense block fires interrupts while scanning to update internal data structures.

Start the CapSense component. It is important that this line of code comes after the interrupt enable macro. This is because CapSense defaults to using an auto-tuning feature, which relies on interrupts being enabled in order to set up the component correctly.

Next initialize the baselines, this zeros out the noise and the parasitic capacitance. Then run the ScanEnabledWidgets function to start the scanning process. This function returns immediately as the hardware block can do the scan in the background. In your code you must only read the scanning results when the scan is complete, which you can determine from the IsBusy function. In the main loop check if CapSense is busy. If it is not that means the scan is complete so you can act on the results.

The GetCentroidPos function takes an argument for the widget you are interested in. In our case there is just one slider but you could have configured several of them. The name of the widget is generated by the component and follows a formula. It starts with the component instance name, then the name of the slider from the customizer, which is LinearSlider0, but that has to be capitalized. Lastly you append the type of the widget, which is the string "LS" with

two underscores. The GetCentroidPos function returns a 16-bit value for the position of the finger. If there was no finger present then it returns 0xFFFF hex. You are going to send a message to the terminal about the finger position now but only if there was a finger on the slider and the position changed.

The UART has a UartPutString function that requires a conversion of the numerical result from GetCentroidPos into a string. Do that with the sprintf compiler library function. At the top of the file add an include of stdio.h and create a character buffer for the string. Then, make a call to sprintf in the main loop to copy the number into the string and append carriage-return and line feed characters. Once you have the string just print it to the UART.

Lastly, you need to update the CapSense baselines using the UpdateBaselines function to correct for changes in the environment and then restart the scanning process using ScanEnabledWidgets

In the terminal emulator you will see that you have very fine control over the value of the slider even though it only has 5 sensors. For your challenge I want you to replace the slider with a pair of buttons. Just configure the sensors at opposite ends of the slider to be CapSense buttons. In the C code detect presses and releases on both buttons and report them via the UART. Make sure you do not just stream data all the time – you are going to have to detect a change in the button state – up or down – and just report the event only once.

As always you are welcome to email me at alan_hawse@cypress.com.