My name is Alan Hawse and this is PSoC 101 – even though it's PSoC 101 3 years later than I originally did it.

In the last project we used a toggle flip-flop to control pins directly from the hardware. Now we are going to use a basic counter to make the operation even more complex even though the schematic will remain quite simple. This will be a great way to introduce you to the concept of wire busses.

Make a copy of the flip-flop project and delete all of the flip-flops. We only want the pins and the inverter. Search for and add a BasicCounter. Set it to be 2-bits wide and place a logic high component on the enable, so it will just run, and a logic low on the reset. Lastly, connect the inverted switch signal to the clock input of the Basic Counter so that it will count button presses. Now we need to wire our 2-bit output to the red and green LEDs. You'll immediately notice that the wire is really thick. This is because the BasicCounter output is a bus – it's actually a pair of signals that are shown drawn together. When you wire the bus to the LED pins you get errors about mismatched widths. To address that right click on the wire and select "Edit Name and Width". In the dialog, we want to manually name the wire so un-check the first box. Then un-check the name box because we still want the tool to choose the wire name. All we want to do is tell PSoC Creator that we want to pick one of the signals from the bus. Click on the Bit radio button and it chooses bit zero. Close the dialog and repeat for the wire that goes to the green LED. This time change the bit number to one to grab the other signal on the bus. That clears all of the errors and adds the bit number labels to the wires. You can move the labels around relative to the wires in order to make things look right and nice and neat. If you get errors about having no input on Pin_BLUE it is because you previously enabled the hardware connection on that pin and you're not driving it in this design. Simply open up the pin customizer and un-check the HW connection box to clear the error.

When I program the kit it now toggles through the 4 colors that are made by combining the two LEDs – it will go: off, red, green, and then yellow.

Re-create this design for yourself and extend your design into a 3-bit counter that drives all three of the LEDs. This will give you 8 different color combinations – off, red, green yellow, blue, magenta, cyan, and white. Once again, be sure to make the blue LED pin hardware-accessible.

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