My name is Alan Hawse and this is PSoC 101. Three years ago, I made this set of videos. Unfortunately, three years ago I made an error and we had to pull one of the videos off the internet. This is the replacement video. The other unfortunate thing was as I got ready to put this video up and to record the video I looked at the old one and I was a lot thinner back then. Oh well…here we go.

So, back to PSoC 101. Now I am going to show you how to toggle pins using the programmable digital hardware provided inside of the PSoC. Again, with no C code at all.

As usual, make a copy of the last project. Delete the wire to the red pin, then search for, and add in, a toggle flip-flop. Wire the clock input to the switch and wire the q output to the pin. This will toggle the output every time you release the switch because the low to high transition causes the flip-flop to toggle. Place a logic high on the ‘t’ input terminal to enable the flip-flop.

This design is a little more complex than the ones we’ve shown up until now and you may find that you forget to connect something or you connect something incorrectly. That’s not a problem. To fix it all you need to do is to make what we call a breaking move on the wire.

Normally, when you move a component around, the wire follows the terminal so that the connection is not broken. If you press the control key before clicking on the component it does a breaking move and leaves the wire behind. This is a great way to correct the wiring quickly. You can also select individual wire segments and delete them in order to re-draw the circuit correctly.

In this program the green LED follows the button, just like in the last lesson, but the red one only changes state when the button is released. So now we'll have four states – off, green, red and yellow. Yellow is of course the color that you see when both the red and the green LEDs are on at the same time.

If I want the red LED to change when I press the button instead of when I release it, I can just add a NOT gate to the clock input.

Now, when I press the switch, the high-low transition on the input pin is inverted, and the color progression is now red, green, off, then yellow.

How many MCUs do you know that can do all that with no firmware? The answer is basically none. This is definitely a good project to do for yourself because it gives you practice wiring things up and making more complex designs. Try making this design and adding in another flip-flop to control the blue LED and make the program cycle through another color progression. You can then use the rising edge of the button for one flip flop and the falling edge for the other. Don’t forget to enable the hardware connection in the blue LED or you'll drive yourself crazy trying to get it to toggle.

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