P2-4b Kill Switch

Welcome back to Cypress Academy, PSoC 6 101. In the last video I showed you how to use the PSoC digital logic to build a hardware kill switch.

Now let’s add that same kill switch functionality to our BLE-controlled arm project. In order to do this I'm going to take advantage of the fact that PSoC Creator will let you have two projects open in the same workspace at the same time. I'm sure that as soon as I said this my friend Greg had the hair stand up on the back of his neck. He hates having multiple projects open at the same time because it is really, really easy to edit in the wrong place or program the wrong thing.

Let me show you how this works … if you look in the workspace explorer you will see that I have a bunch of projects in this workspace. And you may notice that one of these projects is in BOLD… that means that it is the currently active project. When you click the build or program button this is the project that will be built and programmed

Many times, I have seen people - including myself - edit in a project and then click program… only to find out that they were editing the wrong project.

Here let me show you.

First, right click on the main controller project and pick “set as active project”. Notice that after a second or two that project becomes bolded. That means this is the project that will be programmed.

Now open the BasicKillSiwtch schematic. Then open the schematic for your main controller. Notice the tab for the basic kill switch is somewhat grey. That means it is NOT - I repeat NOT - the active project. In fact, any tab that is grey is not the active project… and any tab that is white is part of the active project.

Notice that if I change the BasicKillSwitch project to be active that the two tabs switch… one goes to white and the other goes to grey.

The best way to not be confused is to close all of the tabs except for the ones you are using. You can close a tab by right clicking and choosing close… or by right clicking and selecting close all but this to close all other tabs.

Now, open the BasicKillSwitch schematic and the main controller schematic. Then right click on the main controller project and select set as active. Now click on the tab for the kill switch schematic… select all of your kill switch hardware and type ctrl-c to copy it to the clip board.

Then click on the main controller schematic and type ctrl-v to paste and click where you want it to go in the schematic.

Now close the basic kill switch project so you don't accidentally edit it.

It turns out that one of the features of the TCPWM is that you can start and stop the PWM counter using hardware. Moreover, both the start and stop inputs can be configured as rising or falling edges.

To enable a hardware kill and start functionality on the PWMs, we need to double click and configure the PWMs. Then I’ll click on the advanced tab, and set the kill input to falling edge and start input to rising edge. Back to the schematic, I can now connect the output of the flip-flop to the kill and start inputs of both PWMs.

Once the schematic is done we need one small change in the main\_cm4.c file - specifically to start the LEDBlink PWM… like so.

Another thing that people tend to forget is to set the pins. So I'll go back to the DWR, set the pins… then hit the program button.

Now let's try it out… notice that when I reset the controller… that the LED is blinking red…. When I open up the terminal notice that the keys don’t change the position of the robot… excellent. Then when I click on the hardware switch the blinking red turns off and the green turns on… AND… I can control robot arm with the keys… nice.

Now we have our Kill Switch, aka Nicholas Safety Switch, in place. In the next video I will walk through how to use the capacitive sensing interface on the PSoC 6 BLE Pioneer Kit to control the robotic arm with our CapSense technology.

You can post your comments and questions in our PSoC 6 community or as always you are welcome to email me at alan\_hawse@cypress.com or tweet me at @askioexpert with your comments, suggestions, criticisms and questions.