

Software Input Pins

My name is Alan Hawse and this is PSoC 101. In the last lesson I talked about output pins. Now let's read some input. Start by making a copy of your first project. You can do that by using the save-as feature which is available by right clicking the original project to bring up the context sensitive menu. This option is like save-as in any other document editor program – it creates a new copy of your project and makes it the focus of PSoC Creator. Another option is to use the copy-paste functionality – copy the project using the option in the context menu and paste it into the workspace. You get a copy of the original project, which you can then rename. Be careful as all the open files remain in the editor. The advantage of this approach is that both the original and the new projects are available in the workspace. That will be useful down the road if you wish to refer back to your older designs. However, make sure that you are editing the files in the active project. The active project-the one that PSoC Creator will build- is highlighted in bold in the Workspace explorer. Files that are not part of the active project are greyed out but can still be edited. To make a project active, right click on it and set it to active. Remember, in the editor the active project files have a white tab and inactive projects have grey tabs. To make sure I do not edit the wrong files I usually open the schematic file and then click “close all but this tab”

Before I talk about input pins I have to change something from the first lesson that has been giving me the heeby-jeebies. If you look at the schematic you will see three pins but there is no indication of what they do. If you showed this to a colleague they would have to rely on your pin naming to understand the intended functionality. Let's fix that by documenting the led circuit by using the library of off-chip components.

In the pin customizer I will expose the external terminal. Then I'll search for a resistor, drop it into the schematic and wire it to the pin. I do not want to show the specific name and value of the resistor in this case because I am just interested in the intent of my project, not the specifics of the Pioneer kit. So I will remove that from view in the customizer. Next I'll add an LED and give it a handy name. And finally I will add a power supply and hide its name. To wire these items up I use the wiring tool. Look how the cursor changes to show that you are over a terminal that accepts a connection. Just click to complete the wire. Once you have made the connection the wire will remain connected when you move things around. You can also create a wire by pressing the w key. I'll connect up the LED and power as well – notice how I can just touch the terminals to make a connection – I do not have to use a wire. OK, now I can repeat that with the other pins to make my schematic easier to understand. Remember to expose the external terminal on the blue and green LEDs too. All of the components and

wires that are BLUE are just for documentation and are not actually part of the PSoC. Don't get confused.

Now I am going to add that input pin. Search for pin again and drop it to the left of the output pins. I am going to connect it to a button on the kit so let's be good engineers and start with the documentation. In the pin customizer I will enable the external terminal. Then I'll add a switch and call it SW2, because that is the name from the Pioneer kit, and then I'll connect that to ground. Back in the pin customizer I have to make some other changes. First, I need a software pin, just like we did in lesson one. Next I have to change the drive mode of the pin. The switch is connected to ground. When I press it, the input signal will go low. However, when I am not pressing it there is nothing to force the signal high. So I will choose the resistive pull-up mode in the pin that connects an internal resistor to ensure that an un-driven pin reads high. One common mistake people make is to forget to set the drive mode and get frustrated that it cannot be read. I've done it myself and so I'll try to remind you about this from time to time.

Now I will assign the pin to the physical pin that is attached to the SW2 button on the kit. In C I will change the old code to simply copy the state of the input pin to one of the outputs. When I build and program this design I can control the LED from the switch.

As before, please take the time to reproduce this project for yourself. There is no substitute for hands-on learning. Now, try to modify the C code so the red and green LEDs alternate – when one turns off the other turns on, and vice versa.

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