PSoC Creator 101: CY8CKIT-049 Prototyping Kit Reset and Return to Bootloader Project

My name is Alan Hawse; I'm Executive Vice President of Software at Cypress Semiconductor.

In this video, I will show you how to speed up your work on the 049 Prototyping Kit by adding a simple reset function. You will program the user button to cause the application to stop running and jump back into the bootloader. By doing this, you won't have to unplug and re-plug your kit to start the bootloader operation.

Normally, when you plug the board into a USB port, it starts in the bootloader application and then quickly jumps into the bootloadable application. But if you want to reprogram the board, you hold the SW1 button down as you insert it into the USB port. That forces the board to stay in the bootloader application. This works just fine, but if you're doing rapid iterations in your project, the procedure gets a little time consuming and is quite frankly a little bit annoying.

So I figured out a way to quickly reset the board and jump back into the bootloader from any application. I put this circuit into just about every design I do with this kit. The design uses a timer component that is controlled from the user button. When the timer expires, after two seconds, it triggers an interrupt called "Reset_ISR"; which then calls the bootloader API to reset the target and rerun the bootloader.

I'll breakdown the schematic, so you can understand what it does. The timer is clocked at 12 kHz and the switch is attached to the start, reload and stop inputs of the timer. I'll double click on the reset timer to open up its customizer dialogue. I've configured it as a one shot counter, with a period of 24,000. That means when it starts it will count down from 24,000 and stop when it reaches 0 and then assert the interrupt. The period of 24,000 equates to two seconds with the 12 kHz input clock. In the right-hand corner of the dialogue, you can see that I enable three of the five possible inputs to the timer component. They are all connected to the SW1, but with different modes. The start and reload inputs are active on a falling edge; whereas the stop is active on a rising edge. The 049 switch is active low; that means when it is pressed it will generate a low signal inside of the PSoC. When this happens, it causes the timer to reset the counter to 24,000 and start counting down again. If the switch remains pressed for two seconds, the terminal count is reached and the interrupt is asserted. If the button is released before the two seconds expires, then a rising edge occurs and the counter will stop. The interrupt does not fire and the application will continue to run as if nothing happens. But then, when I press the button again, the reload input ensures that the whole process restarts from 24,000.

You only need a tiny amount of coding in your application to make this work. It's one of the great things about PSoC; many, many problems you can solve with a hardware component that Cypress gives you and not have to go crazy writing a bunch of firmware. In this case, you will first start by writing the ISR; which just has to call the bootloadable load API to reenter the bootloader. Then, in main, just start the reset timer, install the ISR, and enable the interrupts.

After that you can write your application normally and the reset system will not interrupt or interfere with the rest of your application in any way. Add this to your applications and you'll be able to restart the bootloader in just a couple of seconds without the fuss of removing the kit from the USB port. Hit the reset button and jump right back into the bootloader. The bootloader host automatically reconnects to the bootloader each time you plug it back in and then you will be able to download your bootloadable application in no time at all.

There's one last thing to mention: this design does not in any way mean you can't use the button for other things. One of the great things about PSoC is the hardware can be wired where ever it is you want to use it. There's only one switch on the kit, so dedicating it to this one function would really just be wasteful. This is a PSoC; so you can route the SW1 input straight to the blue LED on P16 while still driving the timer. The switch is now driving the LED as well as controlling the reset timer. Now, don't forget if you hold it down for two seconds, it will bomb right back into the bootloader and your application will end.

If you want to give this a try, download the project from our website. I recommend that you play around with the timeout setting on the timer to find your optimum hold time before the reset is activated. And you should definitely verify that you can still use the switch in your application without rebooting the board. I'm very interested in hearing from all of you; you’re welcome to email me anytime you want; alan_hawse@cypress.com. You can win with PSoC and I’ll help you do it.