**GPIOs – Part 1**

Welcome back to Cypress Academy, WICED WiFi 101. In the last video I showed you how to blink an LED using a GPIO configured as an output. In this video, we will monitor the state of a button on the kit using an input and then we will add an interrupt as well.

This time we will start by copying the project from the last video to a new name. The steps are:

1. Right click on the 02\_blinkled folder and select “copy”
2. Right click on the 02 chapter folder and select “paste”
3. Right click on the new folder name, the C source file and the make file one at a time and rename them to “04\_button”.
4. Open the file 04\_button.mk. Update the application name and the C source file name to match the new project.

Let’s also create a make target for this new project now. The easiest thing to do is right click on the previous make target, select New, and change 02\_blinkled to 03\_button. Don’t forget to delete “Copy of ” from the start of the name.

Now we need to modify the code. We’ll double click on 04\_button.c, remove the code to blink the LED and add in code to look at the state of the input. As with the LED, we don’t have to initialize the pin because the platform files already handle that for you, but if we did need to it would be configured as an INPUT\_PULL\_UP.

In the loop, we just look at the state of the button and either turn the LED on or off depending on the button.

Remember, if you are doing this on a kit without the shield, then WICED\_SH\_MB1 will need to be changed to WICED\_BUTTON1 and the LED needs to be WICED\_LED1.

The LEDs on the shield are active high while the LEDs on the base board are active low so the project will result in the LED being off when the button is pressed for the base board.

So that’s cool – the button is controlling the LED. What about if we wanted the button to toggle the LED? A simple way to do that is with an interrupt. First, we will copy this project to a new one called 05\_interrupt and we will make all of the necessary changes to the file names, make file parameters, and make target.

Then, we will update the code in 05\_interrupt to look like this…

In the initialization, we need to enable the interrupt for the pin that the button is connected to. We will make it a falling edge interrupt. Remember, if you right click on it and select “Open Declaration” you can see other options for the interrupt.

We need an interrupt service routine that will be called when an interrupt occurs – it just toggles the state of the LED.

Notice that the there is no loop inside application\_start because there is nothing for it to do. In this case the application\_start thread will just exit once the interrupt is started. We’ll talk more about threads in the RTOS videos coming up.

In the next video, I’ll show you how to use the built in UART to USB bridge to display information on a terminal window on your computer.

You can post your comments and question in our Wifi developer 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.