

## Lesson 2-4 GPIOs, LEDs, and Buttons – Part 2

Welcome back to Cypress Academy, WICED WiFi 101. In the last video I showed you how to blink an LED using a GPIO that was configured as an output. In this video, I will monitor the state of a button on the kit using it as an input. I will also connect the input to an interrupt.

This time, instead of creating a project from scratch, I will start by just copying the project from the last video to a new name. The steps to do this are as follows:

1. Right click on the 02\_blinkled folder and select “copy”
2. Then, right click on the 02 chapter folder and select “paste”
3. Then right click on the new folder name, the C source file, and the makefile one at a time and rename them to “04\_button”.
4. Then, 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 the “Copy of ” from the start of the name.

Now I need to modify the code. I will 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, I don’t have to initialize this pin because the platform files already handled that for you. If I did need to, I would configure it as an INPUT\_PULL\_UP.

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

[exercise hints:

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.

end exercise hints.]

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

Then, I will update the code in 05\_interrupt to look like this...

In the initialization section, I need to enable the interrupt for the pin that the button is connected to. I 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.

I need an interrupt service routine that will be called when an interrupt occurs. All it will do is just toggles the state of the LED.

Notice that there is no loop inside of `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 Chapter 3.

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

Remember, you can post your comments and questions in our WiFi developer community, or, as always, you are welcome to email me at [alan\\_hawse@cypress.com](mailto:alan_hawse@cypress.com) or tweet me at [@askiotexpert](https://twitter.com/askiotexpert). Thank you.