

Course on Internet of Things

Exercises Session 3: Switches

Introduction:

Switches exist in various incarnations. There are simple mechanical switches: on/off or push-button switches but there are also switches that turn on or off with various external conditions:

- temperature
- infrared radiation (e.g. the Passive Infrared Sensor or PIR sensor)
- Hall switches detecting magnetic field
- microphone switches which turn on when a certain noise level is detected

In this exercise we will use a mechanical push button switch and a PIR sensor

Mechanical Switch



PIR sensor



Exercise 1:

The push button switch is connected to GPIO 22 by a solder and connects onto the ESP32 board.

Write a function that polls the state of the switch every 100 milliseconds and prints state changes (only print out a message when the switch state has changed). Catch <ctrl + c> to smoothly exit the program.

Check <https://docs.micropython.org/en/latest/esp8266/tutorial/pins.html> and

<https://docs.micropython.org/en/latest/esp32/quickref.html> for information on how to accomplish this.

Exercise 2:

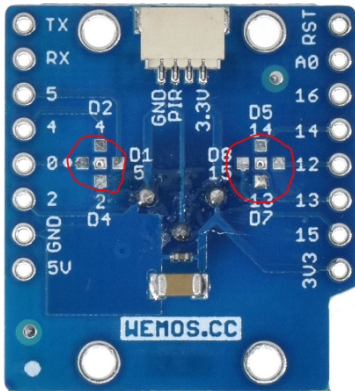
Do the same thing as in exercise 1 but use external interrupts and a callback routine to do the job.

Exercise 3:

Combine this exercise with the exercise on LEDs. Switch the built-in LED on when the button is pressed and off when it is released.

Exercise 4:

Write a program scanning the PIR sensor, indicating if a person is close. The PIR sensor, just like the push button, uses a single GPIO line as input. The boards allow you to choose which GPIO to use, look behind the board to identify the soldered pins if any.



Exercise 5:

Write a program (make use of functions) that turns on all the colors on the WS2812 addressable LED one at a time for each time the push button is pressed clockwise.

(red color, first address to the last address per button release then the next color green then blue). Turn all LED's off at the beginning and end

Exercise 6:

Do the same thing as in exercise 5 but counter clockwise (blue color, last address to the first address per button release then the next color green then red). Turn all LED's off at the beginning and end

Exercise 7:

Write a program that polls the state of the PIR sensor when motion is detected and raise an alarm (with the buzzer) every 3sec and sleeps 500ms until it is interrupted with a push button. Allow the program to exit on Ctrl + C. (make use of functions and use the Tripler Base)

