Workshop on IoT session 2

Hardware access and Micropython

## Exercise 1:

Connect to Micropython using a terminal emulator.

Import the machine package:

*import machine*  
Make Micropython known that our built-in LED is connected to GPIO pin 2 and set this pin to output mode.

Switch the LED on.

Switch it off again

Take <https://docs.micropython.org/en/latest/esp8266/tutorial/pins.html> as reference

Try the *help* function to get information about the machine package and the Pin module  
( help(machine), help(machine.Pin) )

Instead of importing *machine* import machine.Pin:

*from machine import Pin*

How do you access the GPIO now?

## Exercise 2:

Write a script to blink the built-in LED. Use the *value* method of machine.Pin to accomplish this.

Run this script using thonny or uPyCraft

Write a script to switch the LED off

## Exercise 3:

Change the brightness of the LED using PWM.

<https://docs.micropython.org/en/latest/esp8266/tutorial/pwm.html>

Write a script similar to the one shown during the lectures cycling the brightness of the LED. Instead of using a sin function as shown during the lectures, use a triangular function.

## Exercise 4:

Connect the WS2812 rgb LED to the WeMos D1 mini CPU. Just stack it onto the CPU card.

**Before powering the sandwich, please put a piece of paper on top of the LED. The LED is extremely bright and looking straight into it might damage your eyes!**

Write a script to switch all 3 colour components off.

Write a script that powers only the red, then only the green, then only the blue LED. Have the individual LEDs on for 1s and cycle through the colours 5 times. Then switch off the LED.

## Exercise 5:

Cycle through the colours showing the full potential of the WS2812.

## Exercise 6:

Disconnect the WS2812 and connect the SHT30 temperature and humidity sensor instead. Have a look at the description of the I2C class:

<https://docs.micropython.org/en/latest/library/machine.I2C.html>

Have a look at the code of the i2cScan.py script and make sure you understand it. Run it. On which I2C address can your SHT30 be addressed?

## Exercise 7:

Have a look at the SHT30 driver sht30.py. You can find it at:

<https://github.com/uraich/MicroPython_IoTDemos/tree/master/drivers/sht30>

Which methods would you use?

Write a script to measure and read out temperature and relative humidity every 2 s and print the results.