|  |  |
| --- | --- |
|  | *Internet of Things Lab for Air quality Monitoring (IoT4AQ)*  *International Workshop – Air Quality & IoT-based Air Sensors*  *14-15 March 2024, Alioune Diop University, Senegal* |

IoT4AQ workshop exercises

## Exercises on GPIO

**G**eneral **P**urpose **I**nput **O**utput (GPIO) are digital signal lines that can be programmed as input or output.

The ESP32 has an LED that is user programmable and very often a program blinking this LED is used to test the CPU hardware.

**Exercise 1:**

Blink: Write a program that blinks the LED on the CPU card at a frequency of 1 Hz (500ms on, 500 ms off)

**Exercise 2:**

SOS: In the marine world the morse code “SOS” was used to alert security crews that a boat was in difficulties and needed to be rescued. The code consists on 3 short pulses (“S” in morse code), 3 long pulses (“O” in morse code) followed by another 3 short pulses.  
Implement a program that sends the SOS signal on the user LED. Use an interval of 200 ms between light pulses, 200ms for the short light pulse and 700 ms for the long light pulse.  
You may use a function sending a light pulse that takes its length as a parameter.

**Exercise 3:**

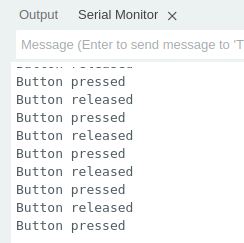
The CPU card has 2 push buttons. The first one (EN) is the reset button and the second one, boot, connected to GPIO 0 is used to put the CPU into flash mode. (You must press the boot button, keep it pressed while you press and release the reset button, and then release it).  
In case you do not use it to put the CPU into boot mode, you can use this button like any push button connected to any other GPIO line.

**Exercise 4:**

Write a program that reads the boot button every 100 ms and prints its state.



* Write a program that reads the push button every 100 ms but indicates its state only when it changes.

****

**Too easy?**

* Write a program increasing the light intensity of the user LED in a linear way and the decreasing it linearly again. Hint: Look up how **P**ulse **W**idth **M**odulation (PWM) works and how it is used on the Arduino SDK.