




POLYTECH
NICE SOPHIA

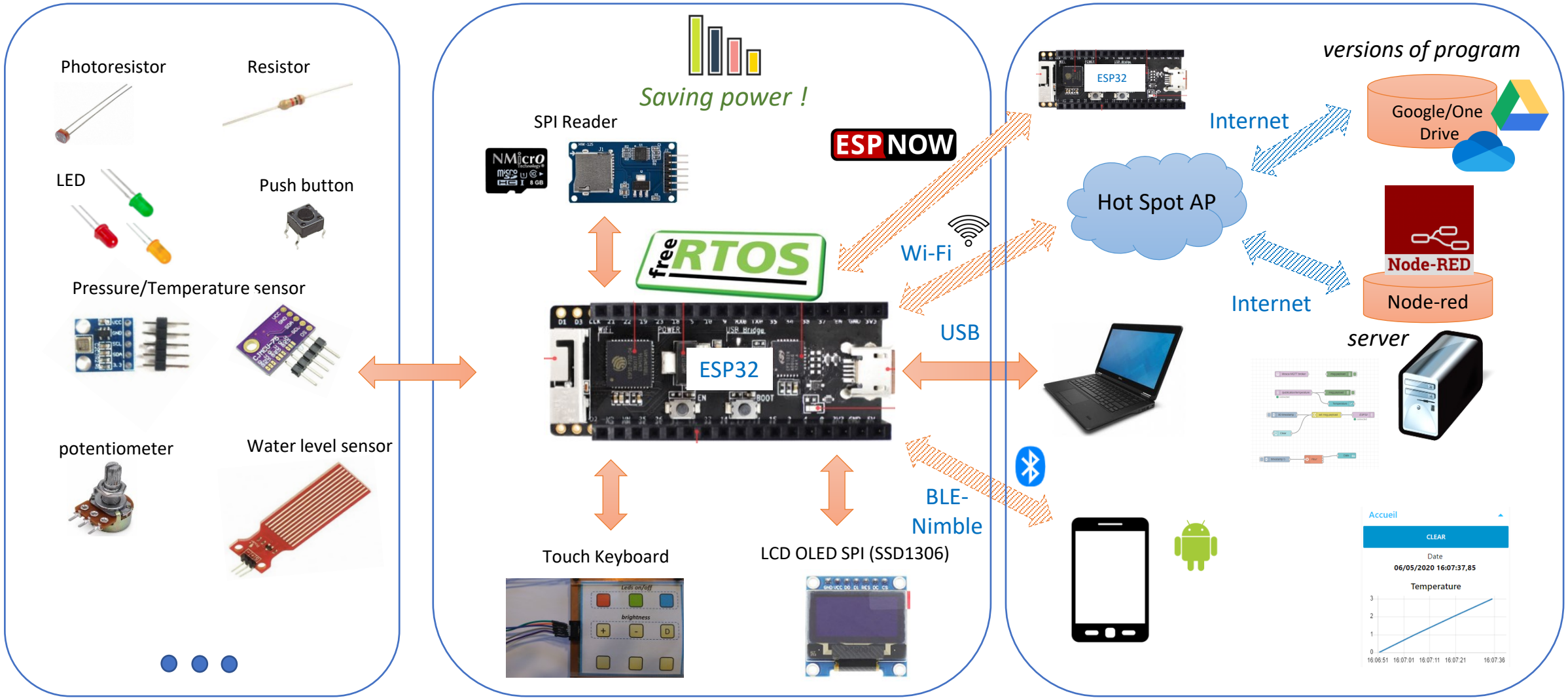
# IoT Design course

Fabrice MULLER

Fabrice.Muller@univ-cotedazur.fr

2021 - 2022

# What we will design … an IoT system



# What we will use … Software & Communication

## Development Tool

Visual Studio Code

Node-Red

Microsoft GitHub

git

ESPRESSIF

CMake

## Network/Cloud

Postman

MQTTBox

One Drive

Google Drive

## Operating System

FreeRTOS

Ubuntu

## Languages

THE C PROGRAMMING LANGUAGE

Java Script

CSS

HTML

C++

JSON

Doxygen

## Android Tool

Beacon Scanner

BLE Scanner

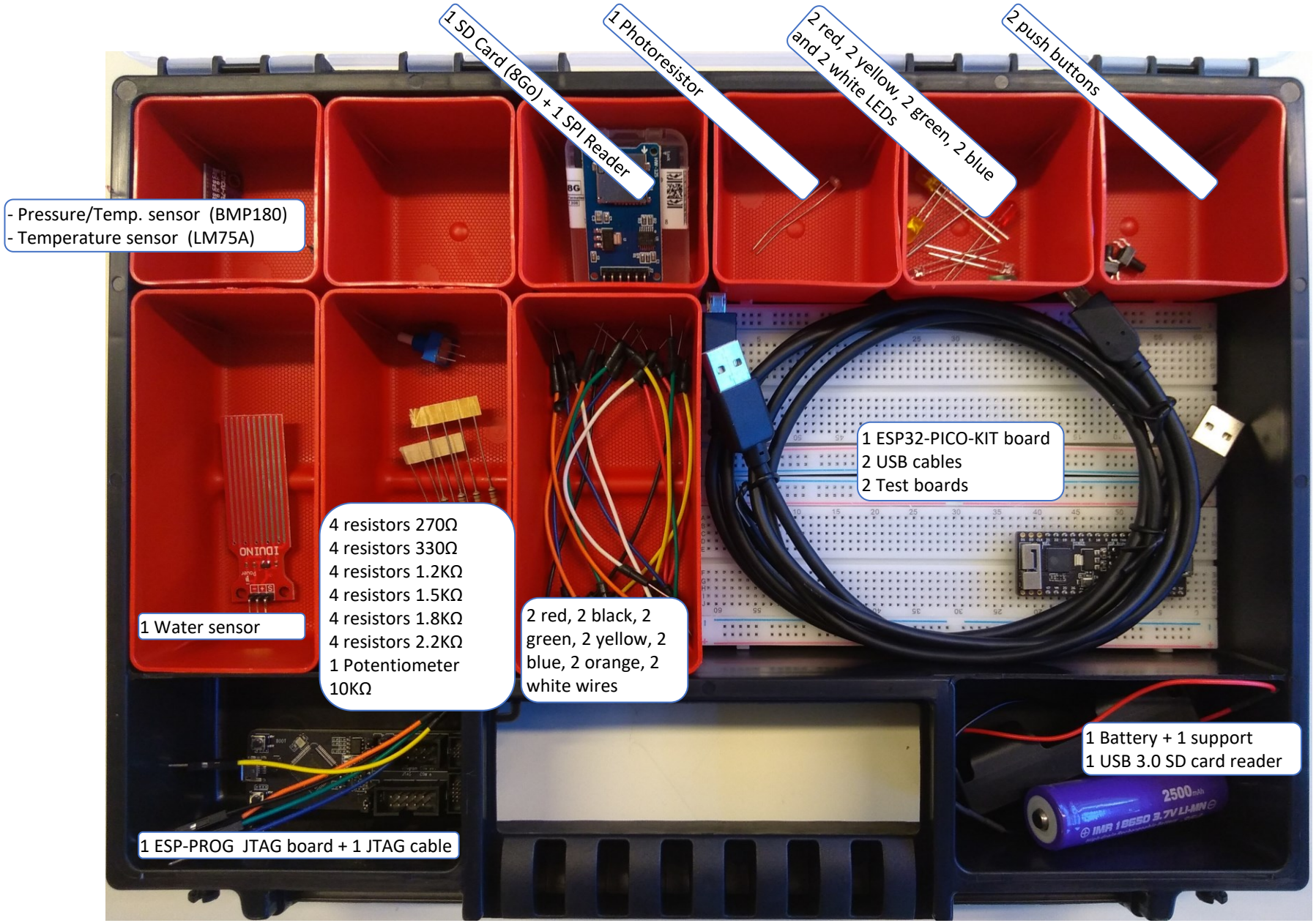
LightBlue

## Wireless Communication

Bluetooth LE - Nimble

WI-FI

ESP NOW ESP Now


1 SD Card (8Go) + 1 SPI Reader
1 Photoresistor
2 red, 2 yellow, 2 green, 2 blue and 2 white LEDs
2 push buttons
- Pressure/Temp. sensor (BMP180)
- Temperature sensor (LM75A)
1 ESP32-PICO-KIT board
2 USB cables
2 Test boards
1 Water sensor
4 resistors 270Ω
4 resistors 330Ω
4 resistors 1.2KΩ
4 resistors 1.5KΩ
4 resistors 1.8KΩ
4 resistors 2.2KΩ
1 Potentiometer 10KΩ
2 red, 2 black, 2 green, 2 yellow, 2 blue, 2 orange, 2 white wires
1 Battery + 1 support
1 USB 3.0 SD card reader
1 ESP-PROG JTAG board + 1 JTAG cable

# Part 1 - IoT Framework

3h30

Lecture : 1h

## Lab 1 : Framework

1h

- Working on Linux
- Espressif IoT Development Framework
- FreeRTOS
- Visual Code Studio
- GitHub
- Doxygen

## Lab 2 : ESP32 Debug

0.5h

- Debugging ESP32 program with JTAG

## Lab 3 : Working with C and IDF framework

1h30

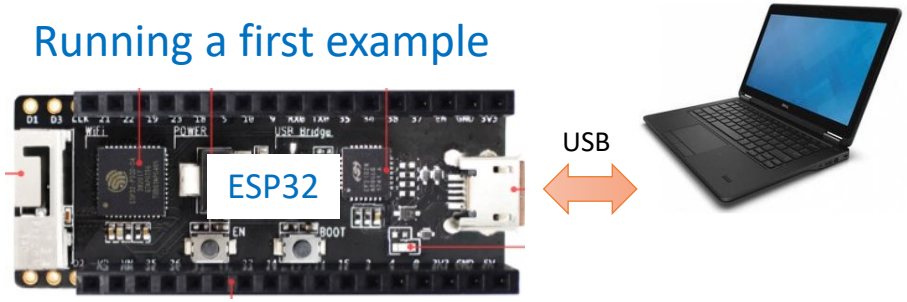
- Macro, header file
- Pointer, memory allocation, linked list
- Doxygen documentation

## Lab 4 : IDF Components & Configuration

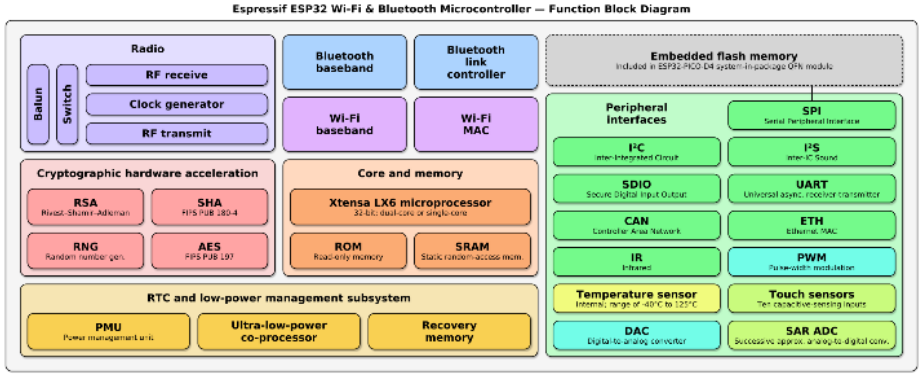
0.5h

- Components for ESP32
- Custom menu configuration
- Default configuration

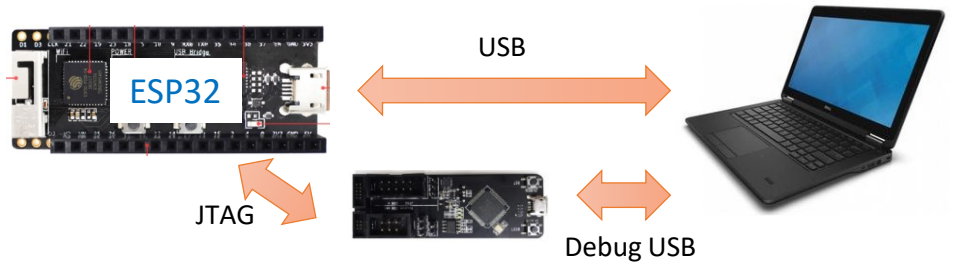
### Running a first example




### ESP32 Architecture




### Debug example



# 4h30 Part 2 - IoT Architecture

Lesson : 30mn

## Lab 1 : ESP32 Architecture

0.5h

- Main characteristics
- Real-Time Clock

## Lab 2 : ESP32 Memory

1h00

- IRAM, DRAM
- Non-Volatile SRAM (NVS)
- SPI flash file system (SPIFFS)

## Lab 3 : GPIO

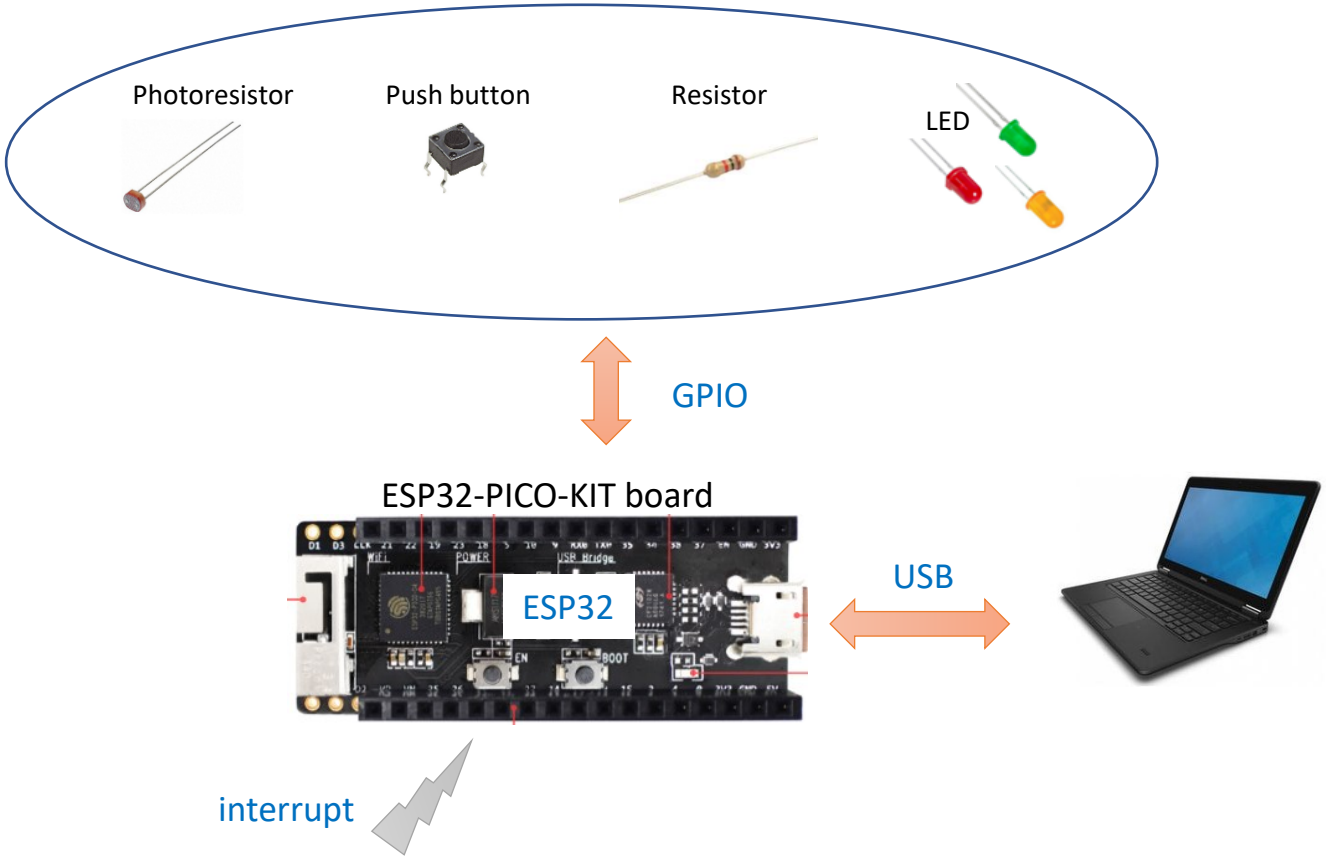
1h30

- Configuration I/O (push button, Led)
- Advanced GPIO configurations

## Lab 4 : Interrupt

1h

- GPIO interrupt
- Push button example



# Part 3 - ESP-IDF FreeRTOS

8h00

Lesson 2h

## Lab 1 : Task & Scheduling

1h15

- One & two cores scheduling
- Idle Task

## Lab 2 : Message Queue & Interrupt

1h15

- Single Message Queue, Timeout & Blocking queue
- Interrupt
- APP: De-bouncing interrupt

## Lab 3 : Semaphore & Mutex

1h00

- Semaphore : binary, counter
- Mutex

## Lab 4 : Timer, Task notification & Event group

1h00

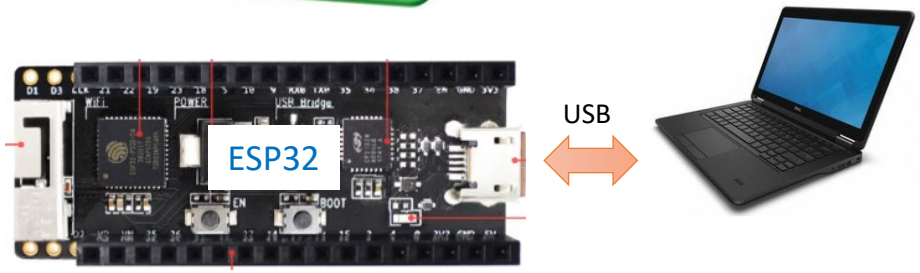
- Software Timer
- Task notification, Event group

## Lab 5 : Full application

Optional

1h30

- APP: application using FreeRTOS functionalities and using keyboard terminal



# 3h30 Part 4 – Inputs/Outputs

Lesson : 30mn

## Lab 1 : DAC / ADC

1h

- Digital to Analog Converter
- Analog to Digital Converter
- APP: color selection of LED = F(potentiometer)

## Lab 2 : PWM

1h

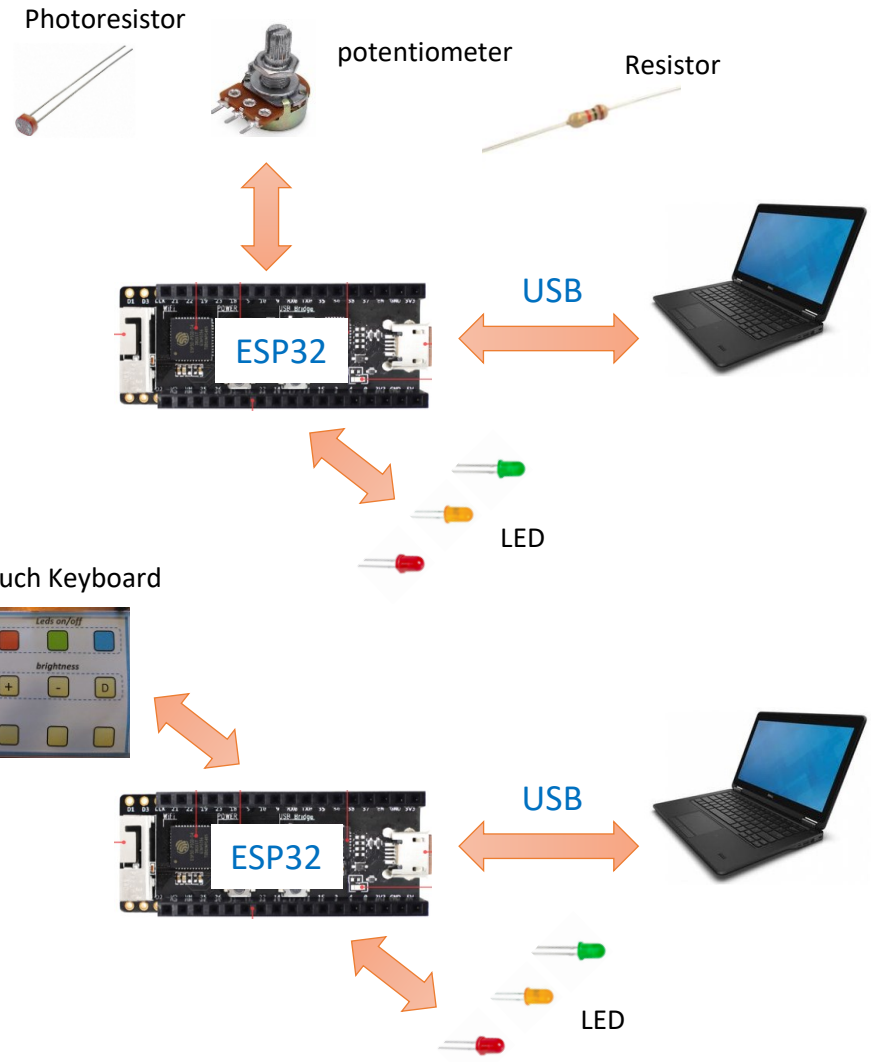
- LEDC
- APP: light intensity of LED = F(photoresistor) with PWM

## Lab 3 : Touch & Hall Sensor

Optional

1h

- Hall sensor
- Touch sensor
- APP: Touch Keyboard management



# 3h30 Part 5 – I2C communication

Lesson : 30mn

## Lab 1 : I2C Master/Slave

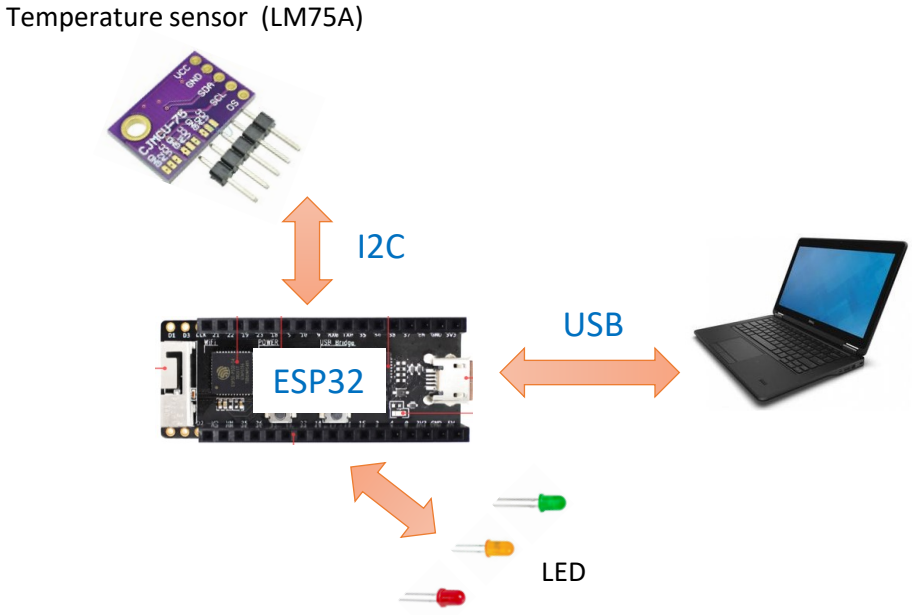
1h

- Memory access master/slave

## Lab 2 : Slave sensor with I2C

1h

- Temperature sensor (LM75A)

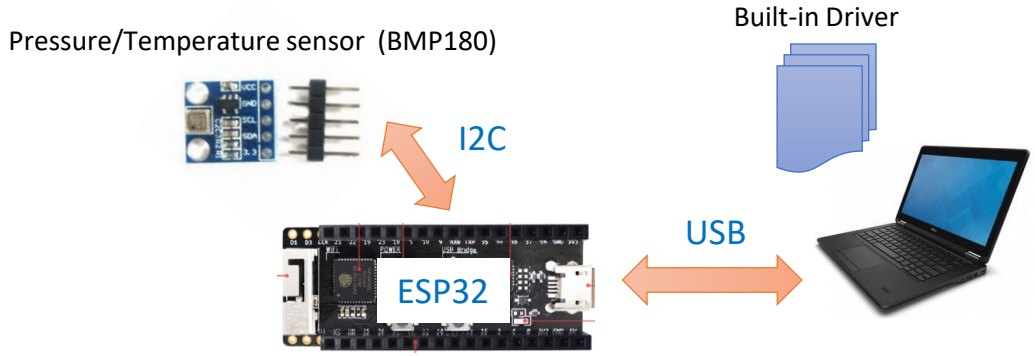



## Lab 3 : BMP180 with I2C

Optional

1h

- APP: adaption of built-in driver



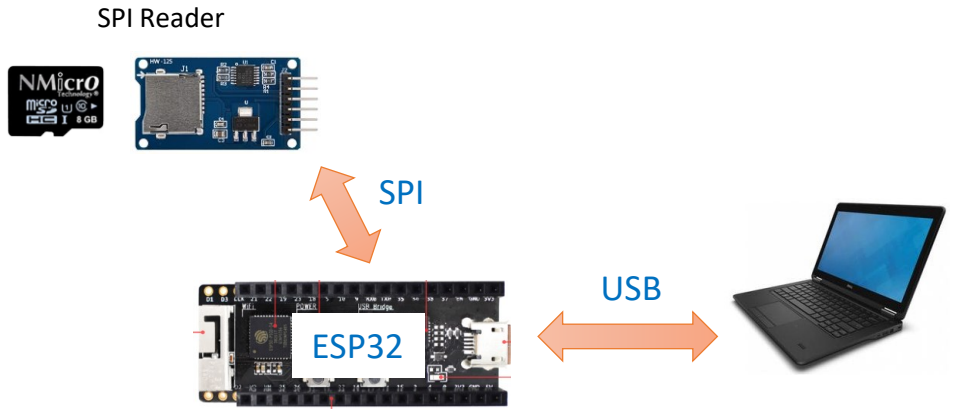
# 2h30 Part 6 – SPI communication

Lesson : 30mn

## Lab 1 : SSD storage with SPI

1h

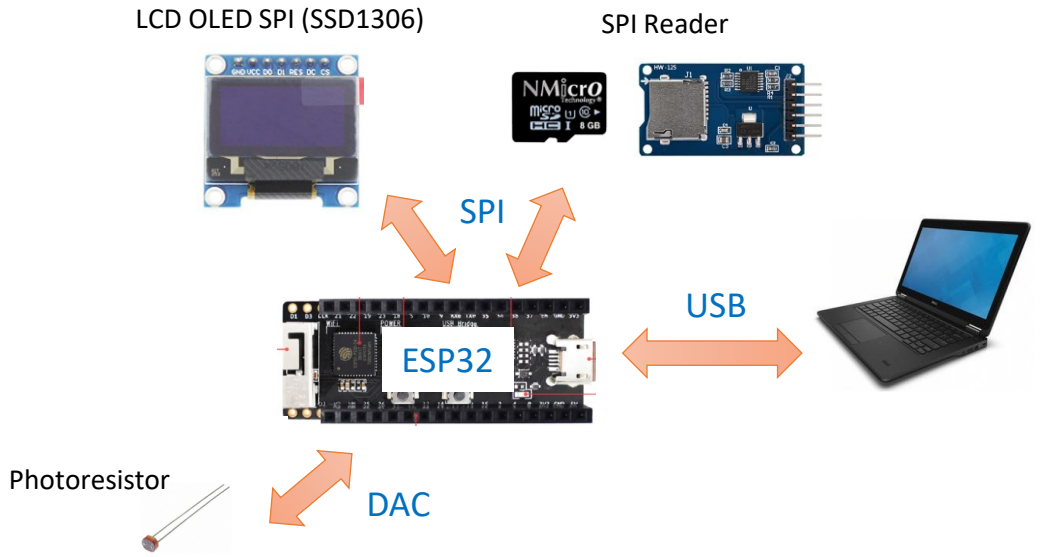
- Using API Unix standard interface




## Lab 2 : Screen with SPI

1h

- LCD OLED (SSD1306)
- APP: Display and save of sensor information



# 1h30 Part 7 – Low power modes

Lesson : 10mn

## Lab 1 : Sleep mode

0.5h

- Timer
- GPIO

## Lab 2 : Deep Sleep mode

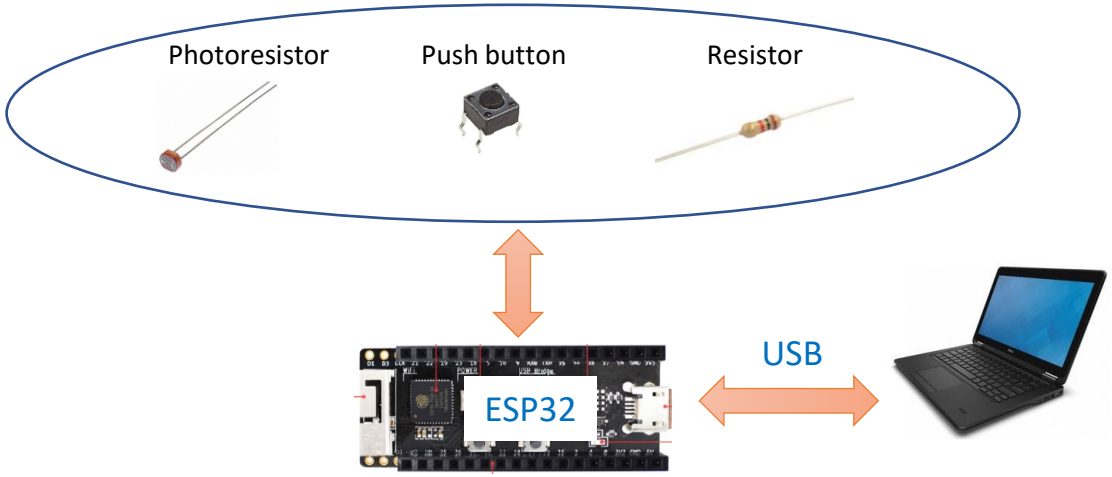
0.5h

- Timer
- EXT0 and EXT1

## Lab 3 : hibernation mode

0.3h

- Timer



# 4h00 Part 8 – Wi-Fi communication

Lesson : 30mn

## Lab 1 : Internet connection

0.5h

- Simple/Advanced connection
- Scanner

WIFI "classroom"

## Lab 2 : REST Client

1h15

- Getting data & memory allocation
- JSON format & parsing
- Advanced application: SMS

WIFI Hot Spot

## Lab 3 : Internet server

1h15

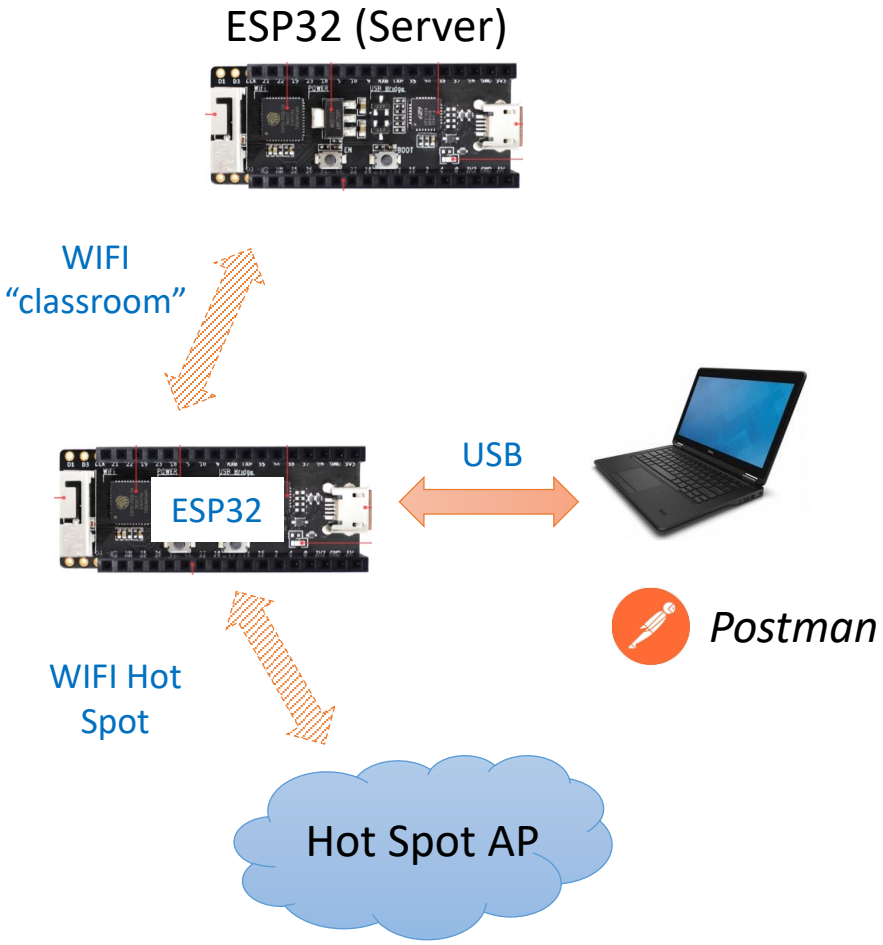
- Create & implement an End points
- Serve & create a simple Web site

Optional

## Lab 4 : Switching Wi-Fi modes

0.5h

- AP & STA mode
- APP: connect a card without hardcoded identifier



# 1h30 Part 9 – Message Queuing Telemetry Transport

Lesson : 30mn

## Lab 1 : Mosquitto Broker

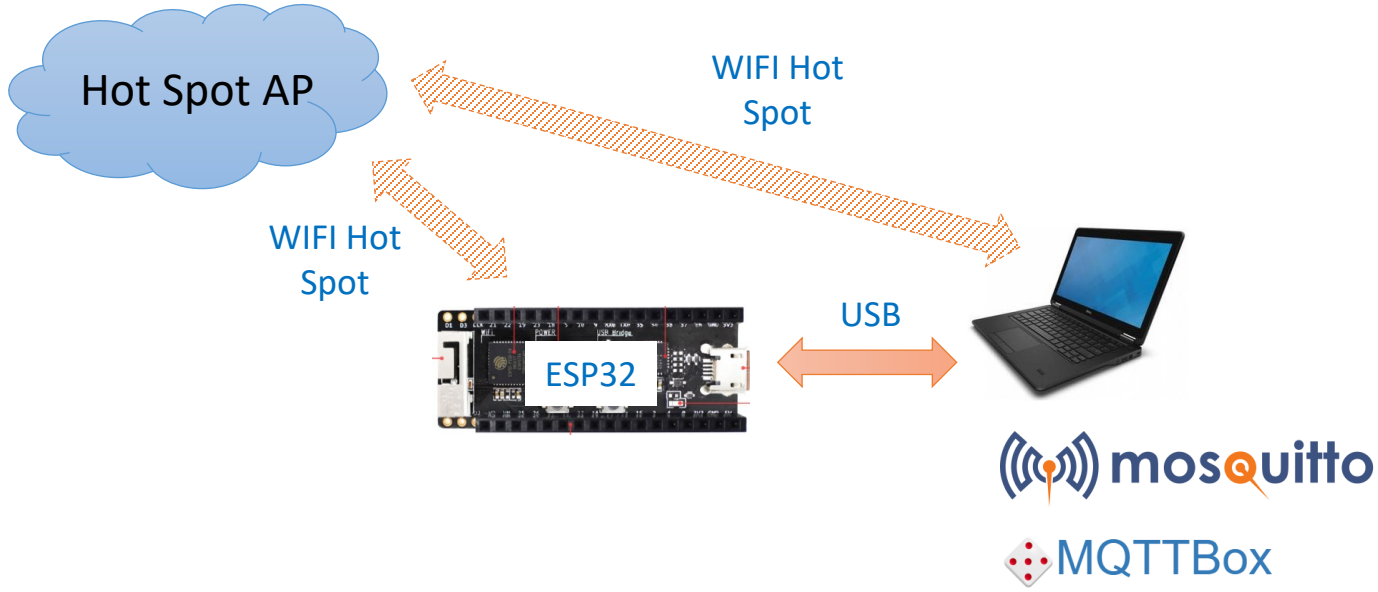
1h

- Install Mosquitto
- Using Mosquitto

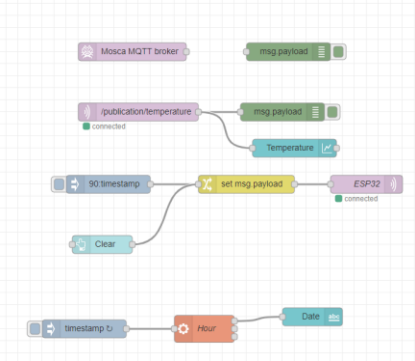
## Lab 2 : MQTT services

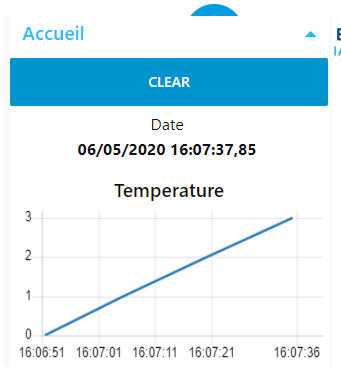
1h

- IoT workflow with MQTT Box
- SUBSCRIBED service
- PUBLISHED service



# 3h30 Part 10 – Node-Red




Lesson : 30mn

## Lab 1 : Working with Node-Red

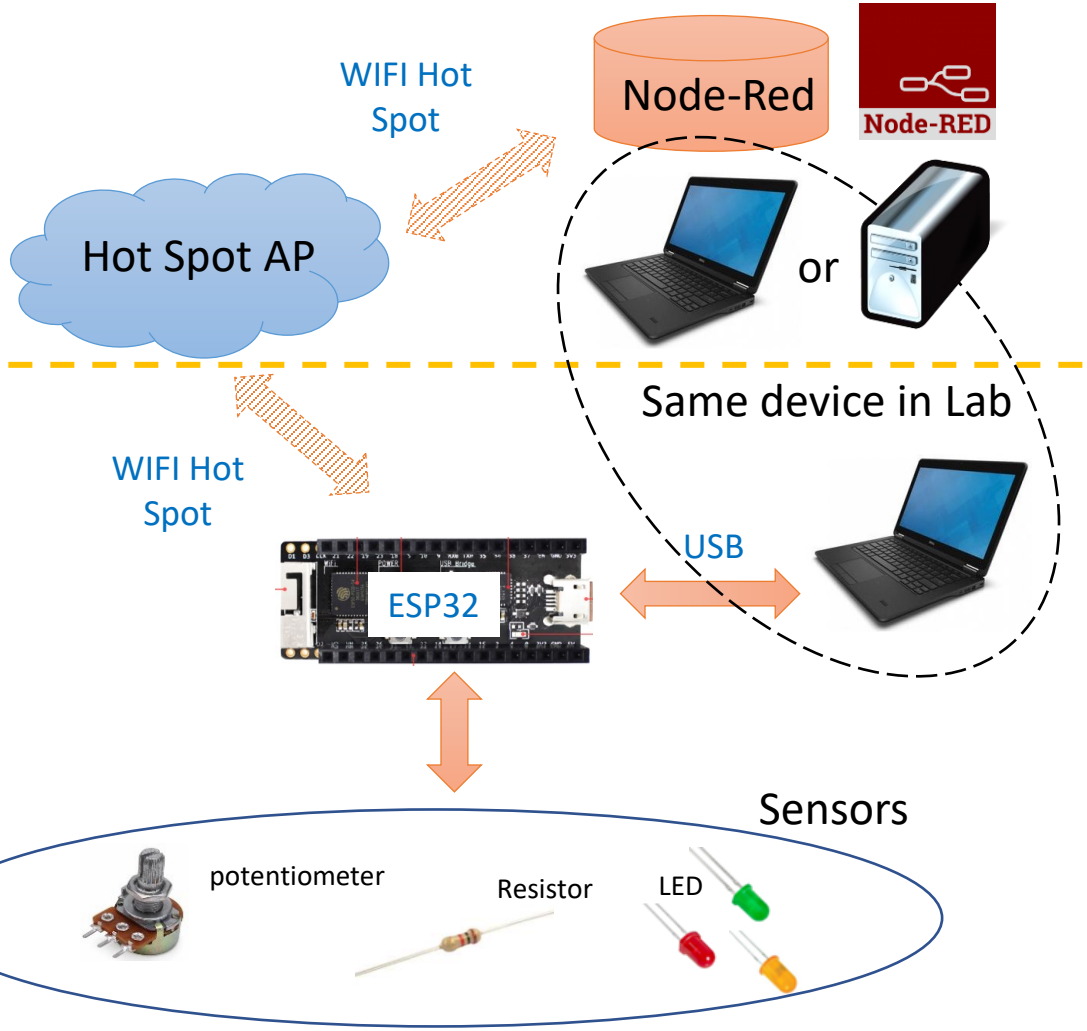
2h

- First use of Node-Red
- Internet connection
- Design a simple application

## Lab 2 : Node-Red with MQTT

1h

- APP: Design a complete IoT workflow
  - Sensor ⇔ ESP32 ⇔ Broker



1h30

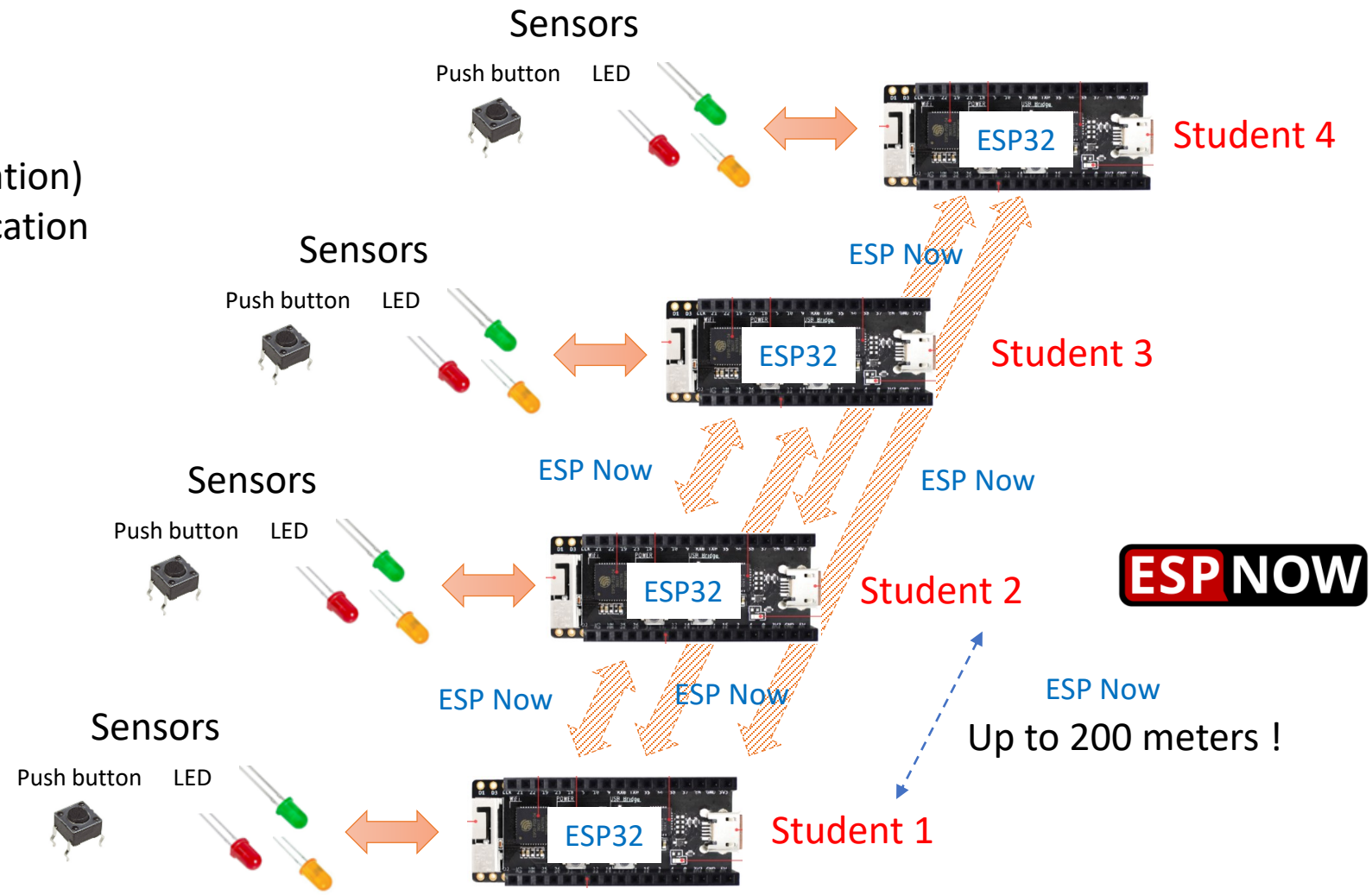
# Part 11 – Wi-Fi & « ESP Now » communication

Lesson : 15mn

Lab 1 : ESP Now

1h15

- Basic concept
- Advanced ESP Now (auto registration)
- APP: Inter-Communication application



# 2h30 Part 12 – Bluetooth LE - Nimble

Lesson : 15mn

## Lab 1 : BLE iBeacon

1h15

- BLE Sequence
- BLE iBeacon
- Eddy Stone format

App

Beacon Scanner

## Lab 2 : GATT & GAP

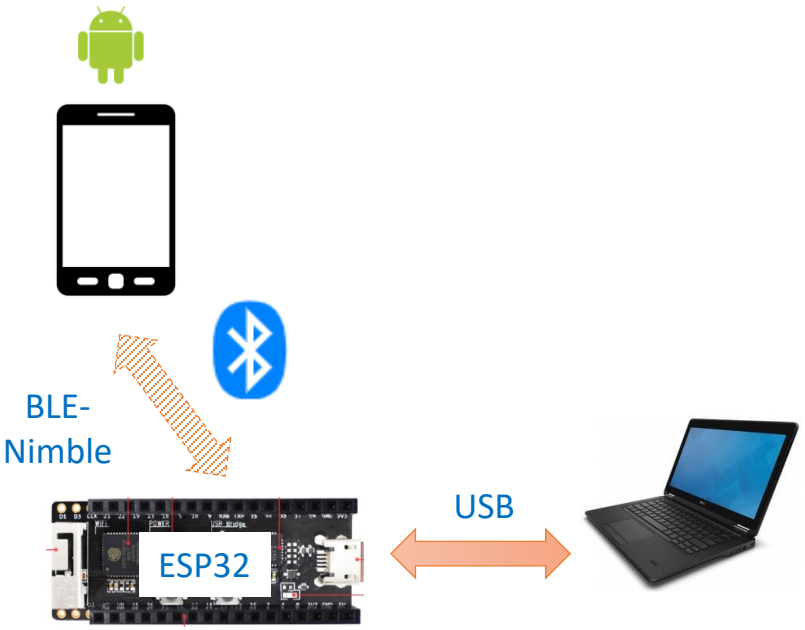
1h15

- Generic Attribute (GATT)
- Generic Access Profile (GAP)

App

BLE Scanner

LightBlue



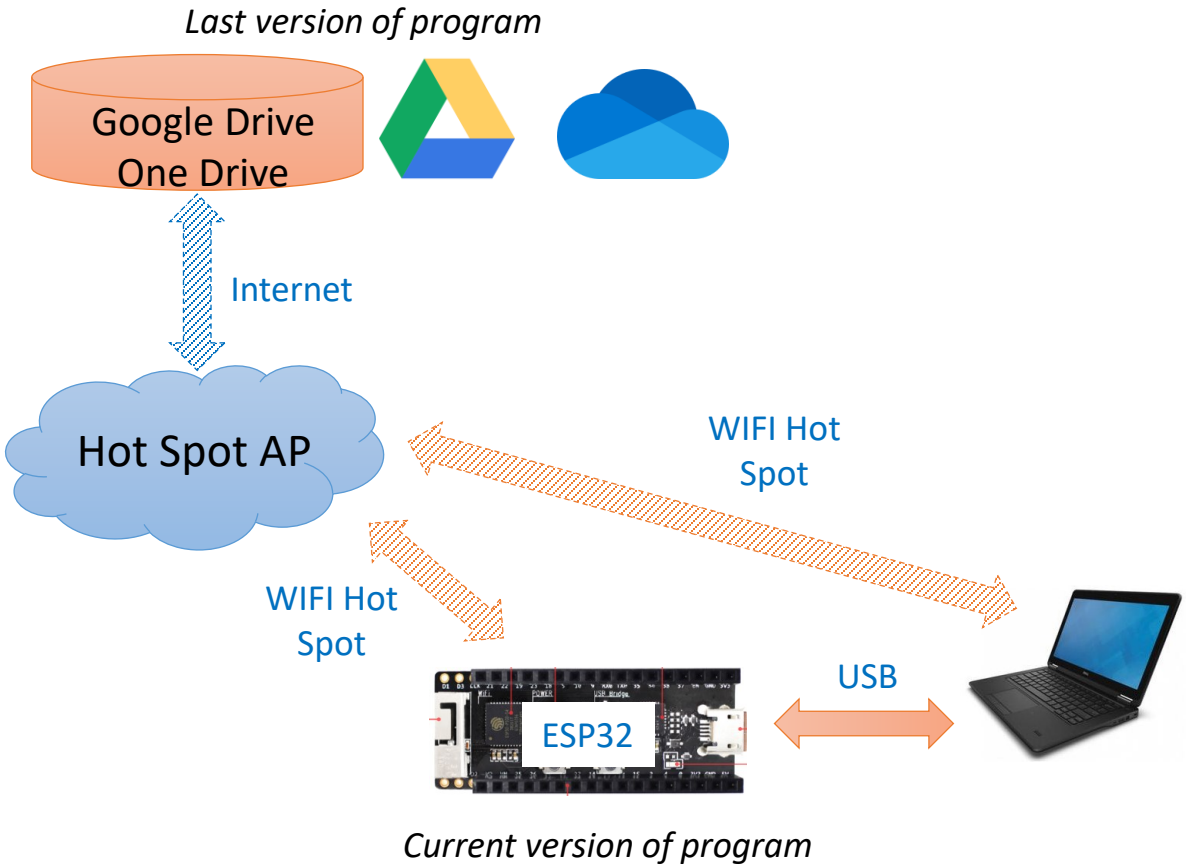
# 1h15 Part 13 – On The Air (OTA) program update

Lesson : 15mn

Lab 1 : Flashing program

1h

- Flashing program with OTA from Google/One Drive
- Versioning with GIT
- Advanced OTA



10h30

# Part 15 – Imagine and design your IoT Application !

Photoresistor

Resistor

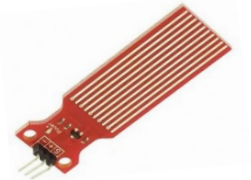
LED

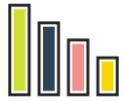
Push button

Pressure/Temperature sensor

potentiometer

Water level sensor





*Saving power !*

SPI Reader

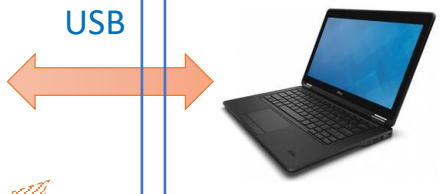
ESP NOW

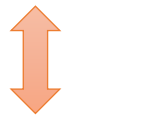
freeRTOS



Wi-Fi

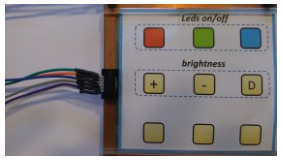
USB





BLE-Nimble

Touch Keyboard



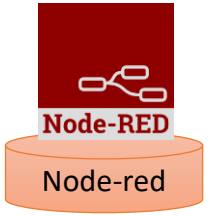

LCD OLED SPI (SSD1306)


*versions of program*

Internet


Hot Spot AP

Internet




*server*