Internet of Things (IoT)

S. Benabid - sorore.benabid@ext.devinci.fr

Année universitaire 2023-2024

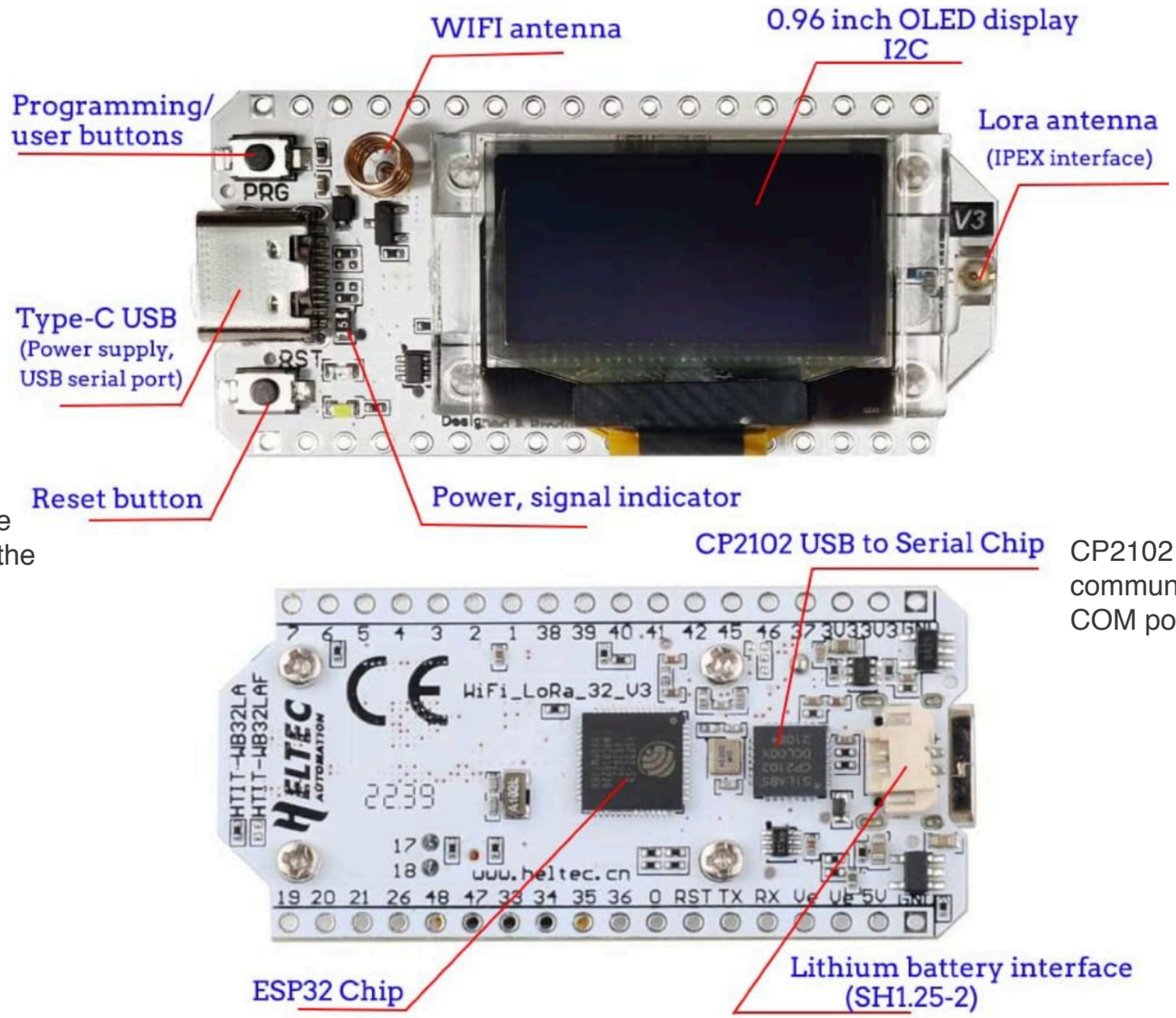
Lesson 2

- MCU Board: Heltec Wifi LoRa 32 V3 Board
- Sensors

Hardware development board

- **MCU Board: Heltec Wifi LoRa 32 V3 Board**

BOOT button to put the board in flashing mode (available to receive code).



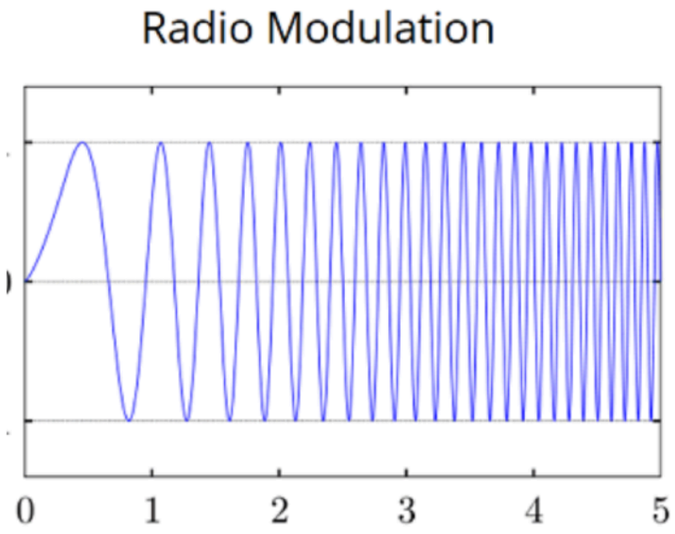
RESET button (may be labeled EN) to restart the board

CP2102 chip (USB to UART) to communicate with your computer via a COM port using a serial interface.

Hardware development board

- **LoRa (Long Range Radio)**

  is a wireless data communication technology that uses a radio modulation technique that can be generated by Semtech LoRa transceiver chips.



LoRa transceiver chips

This modulation technique allows:

Long distance communication

Small amounts of data (low bandwidth)

High immunity to interference

Low power consumption

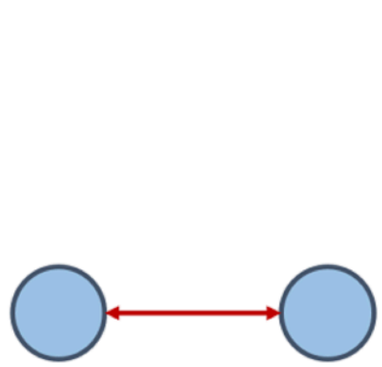
Hardware development board

- **LoRa (Long Range Radio)**

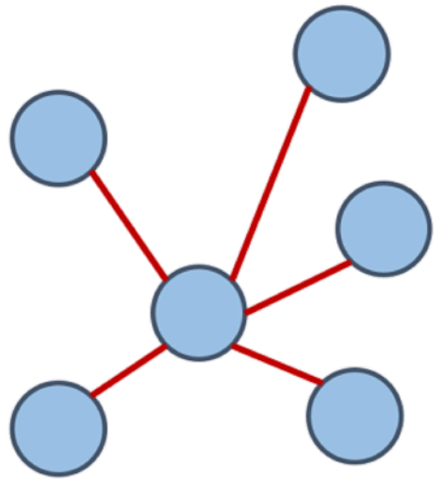
LoRa uses <u>unlicensed frequencies</u> that are available worldwide. anyone can freely use them without paying or having to get a license.

Hardware development board

- **LoRa (Long Range Radio)**



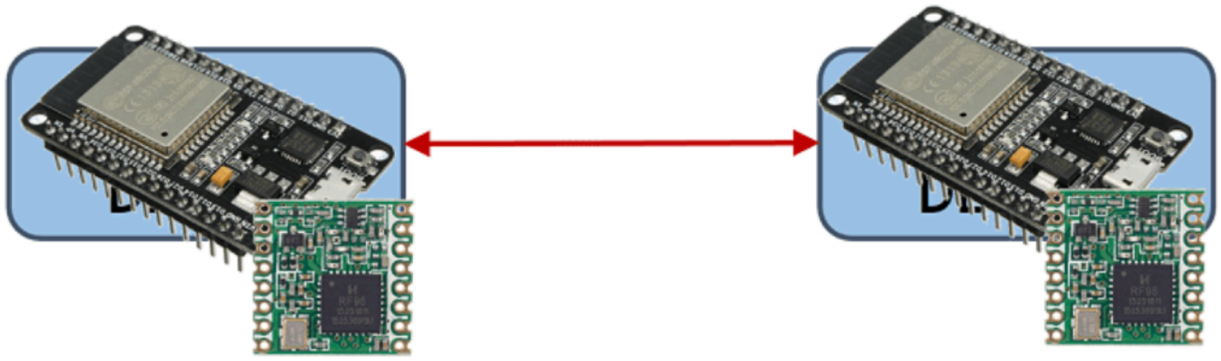
Point to point communication



Network

LoRaWAN™

In point to point communication, for example, to exchange data between two ESP32 boards equipped with LoRa transceiver chips that are relatively far from each other or in environments without Wi-Fi coverage.

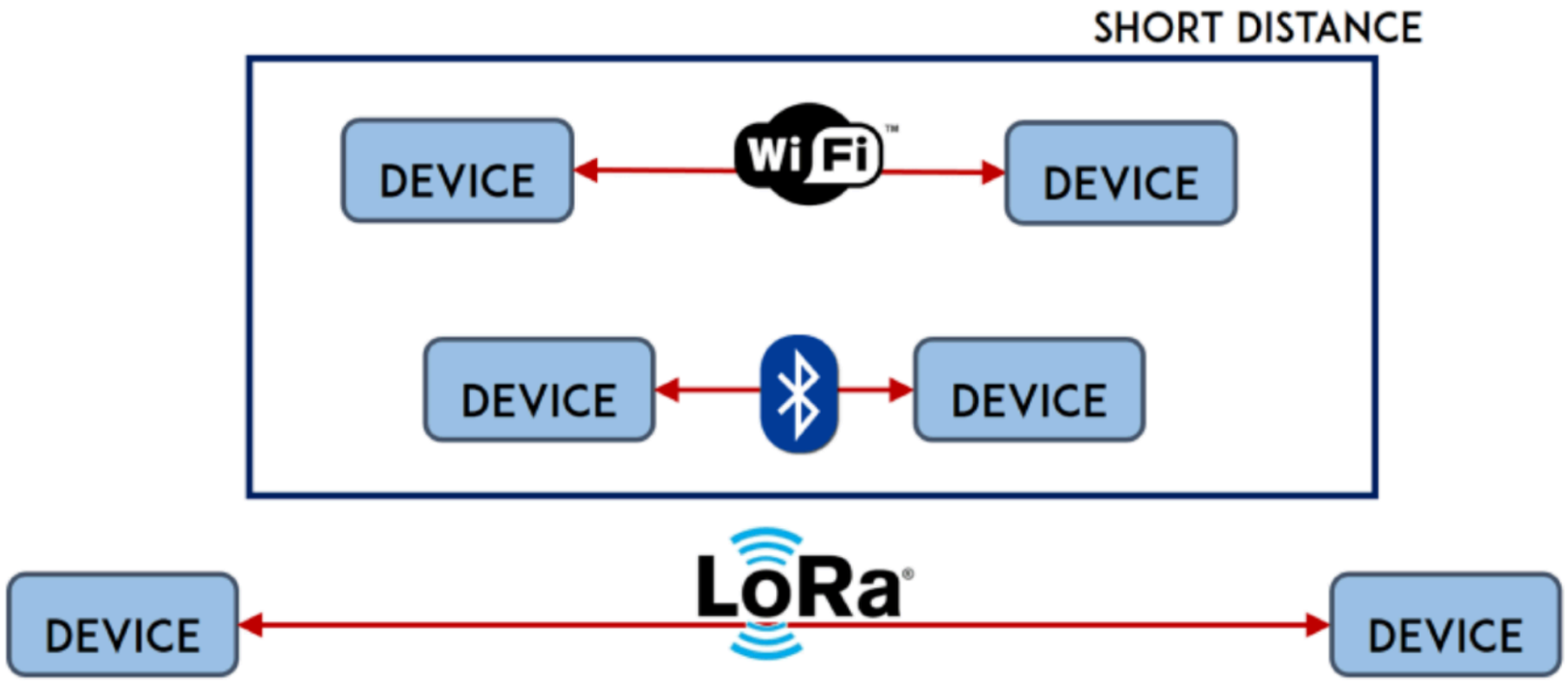


Hardware development board

- **LoRa: Bandwidth vs Range**

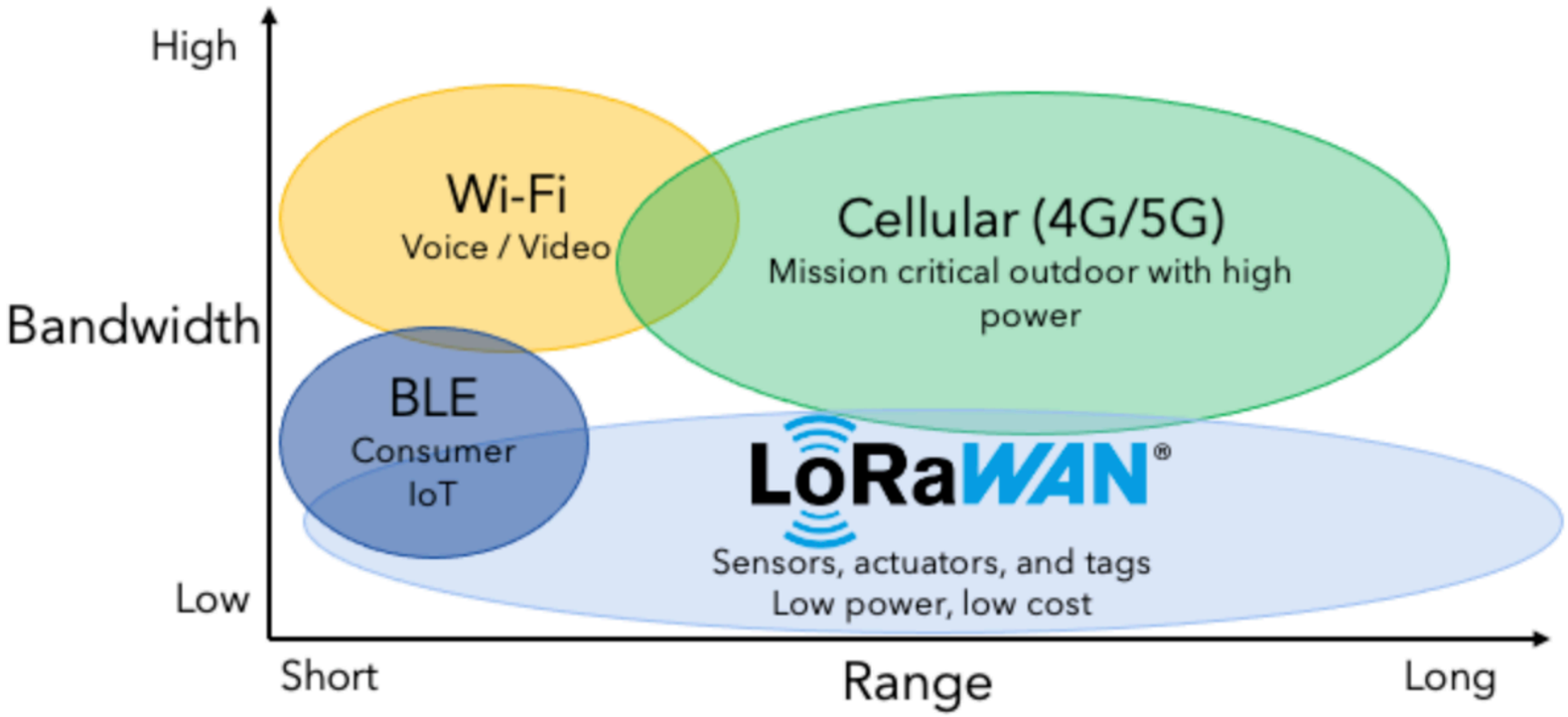
  Unlike Wi-Fi or Bluetooth that only support short distance communication, two LoRa devices with a proper antenna can exchange data over a long distance at more than 200 meters.There are also other LoRa solutions that easily have a range of more than 30Km

  LoRa **Data rate** of **300bps** to **50kbps**



Hardware development board

- **LoRa: Bandwidth vs Range**



LoRaWAN Bandwidth vs Range

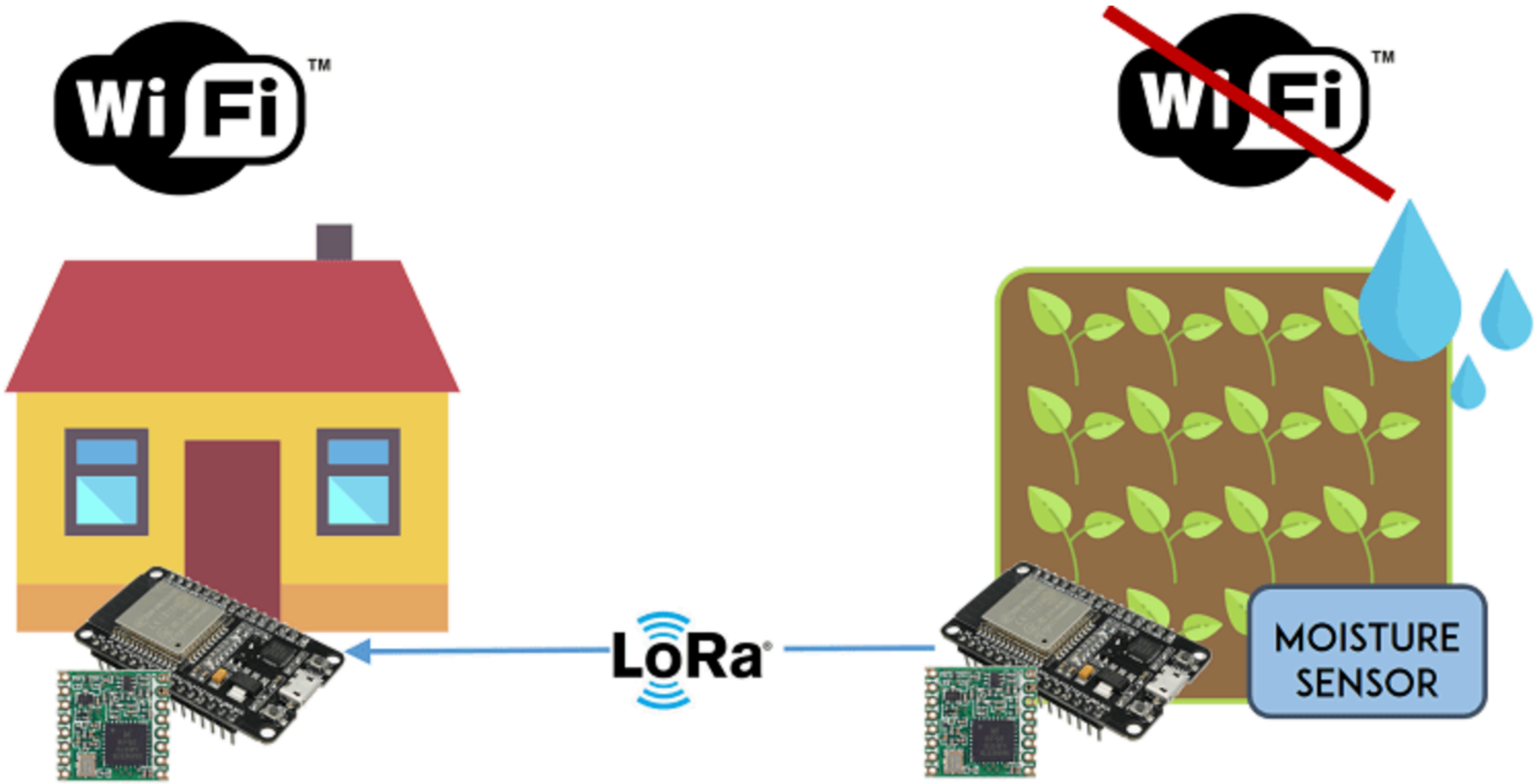
LoRa is not suitable for projects that

- Require high data-rate transmission
- Need very frequent transmissions

Hardware development board

- **LoRa: practical application**

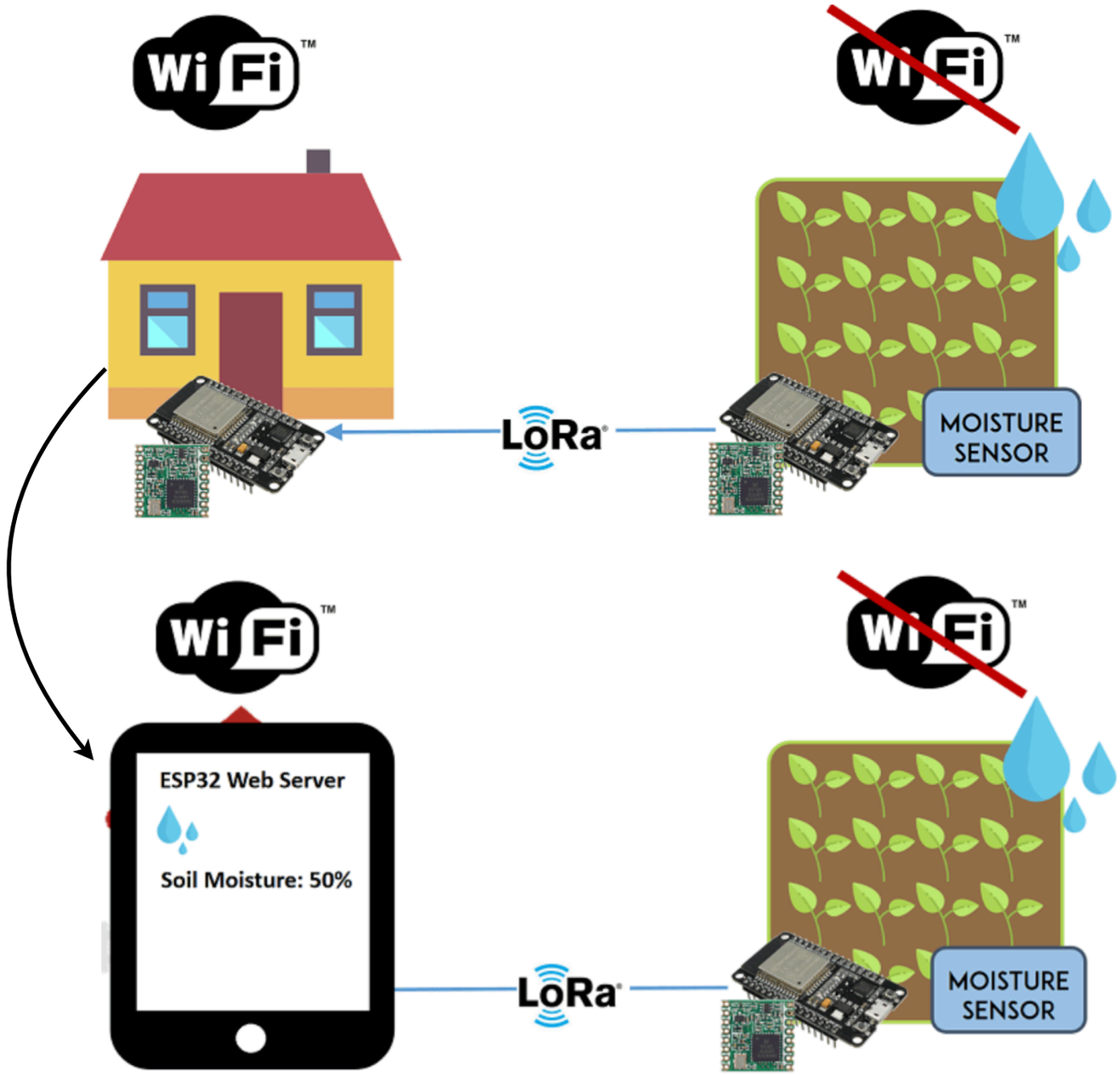
  You want to measure the moisture in your field. Although, it is not far from your house, it probably doesn't have Wi-Fi coverage. So, you can build a sensor node with an ESP32 and a moisture sensor, that sends the moisture readings once or twice a day to another ESP32 using LoRa.



Hardware development board

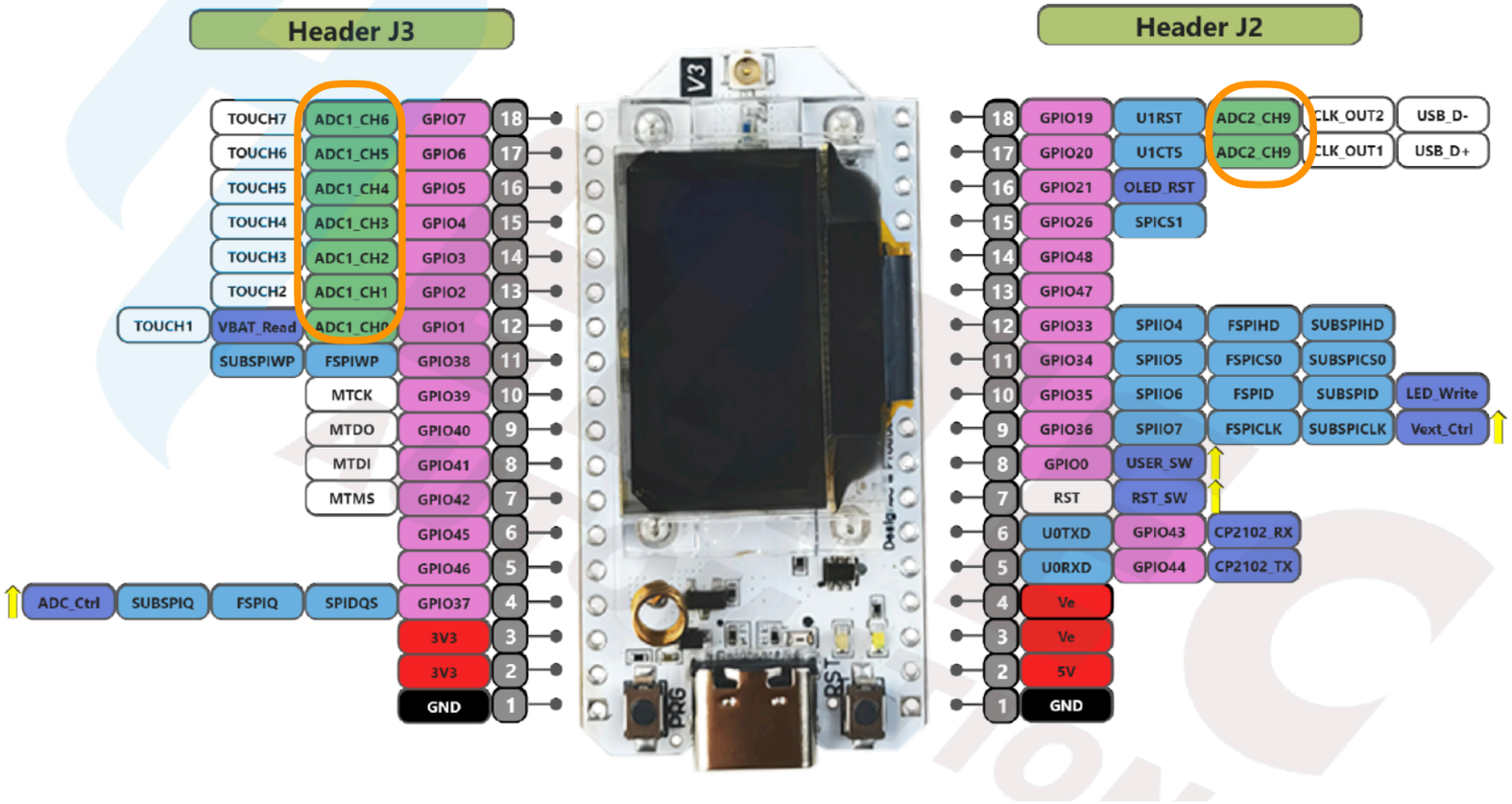
• LoRa: practical application

The first ESP32 has access to Wi-Fi, and it can run a web server that displays the moisture readings.



Hardware development board

- **ESP32-S3 peripherals**
  - 02 12-bits ADC channel
  - 2 SPI interfaces
  - 3 UART interfaces
  - 2 I2C interfaces
  - …etc



The ADC features are assigned to specific static pins. However, you can decide which pins are UART, I2C, SPI, etc – you just need to assign them in the code.

GPIO: General Purpose Input Output

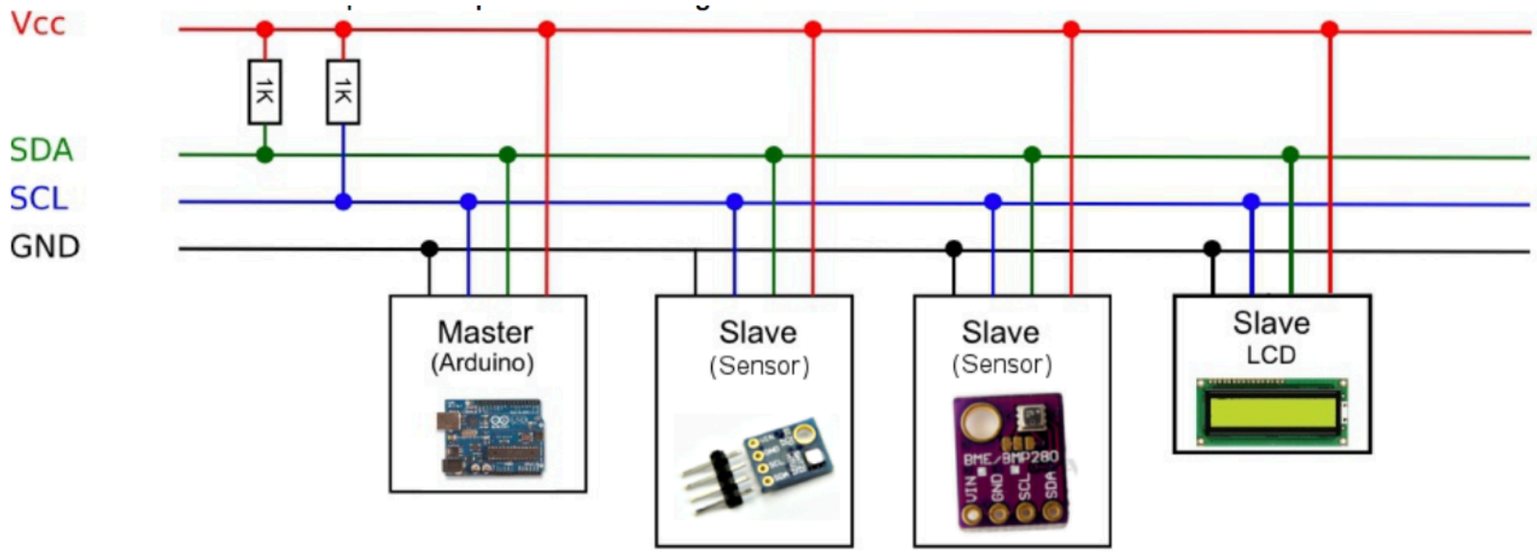
Hardware development board

- **Serial communication protocols**

I2C (Inter-Integrated Circuit): is two-wire serial communication protocol. In which one wire is used for the data (**SDA**) and other wire is used for the clock (**SCL**). Is based on **Half-duplex** communication protocol

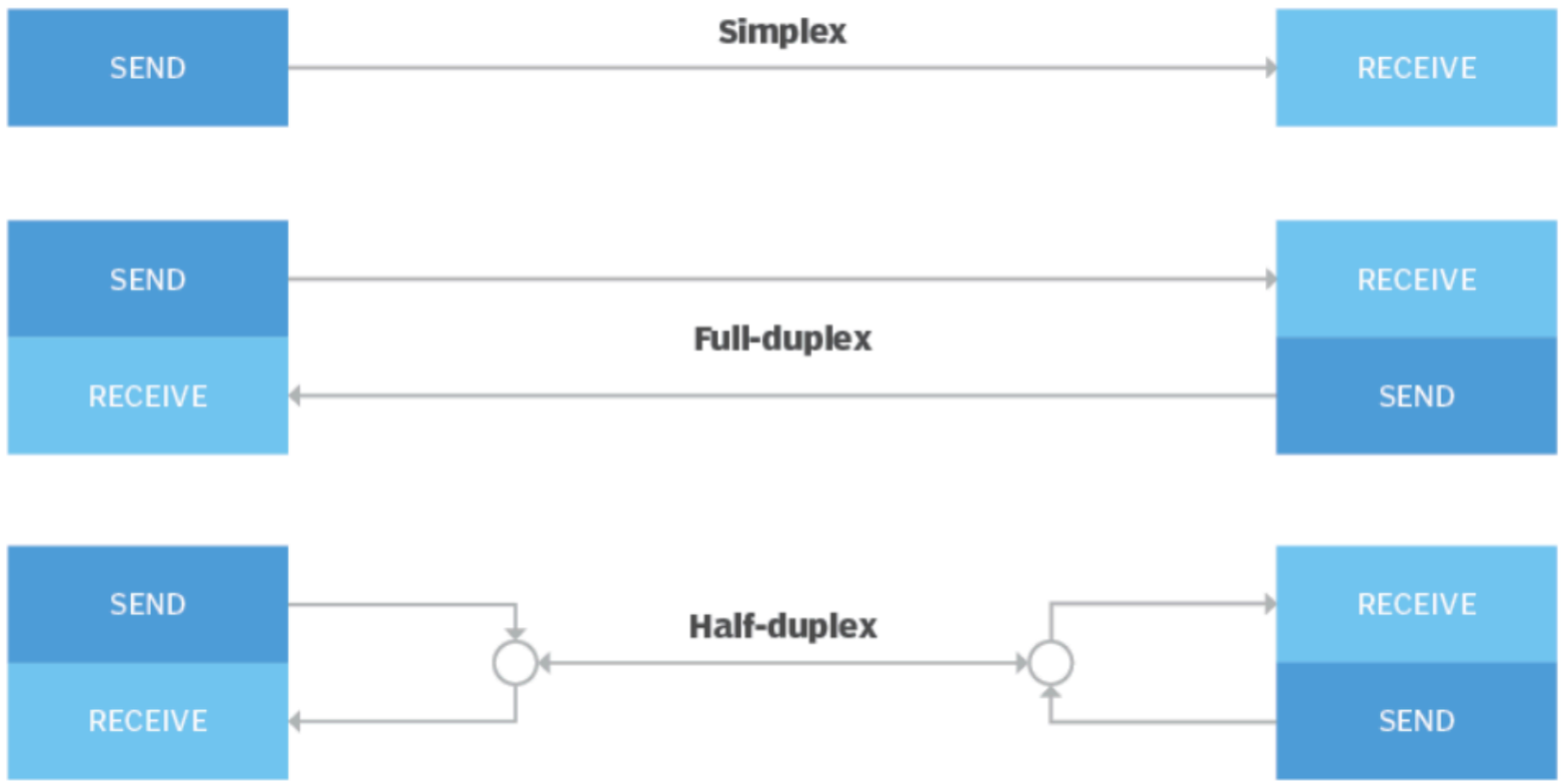
SDA: Serial Data

SCL: Serial Clock



Hardware development board

- **Serial communication protocols**

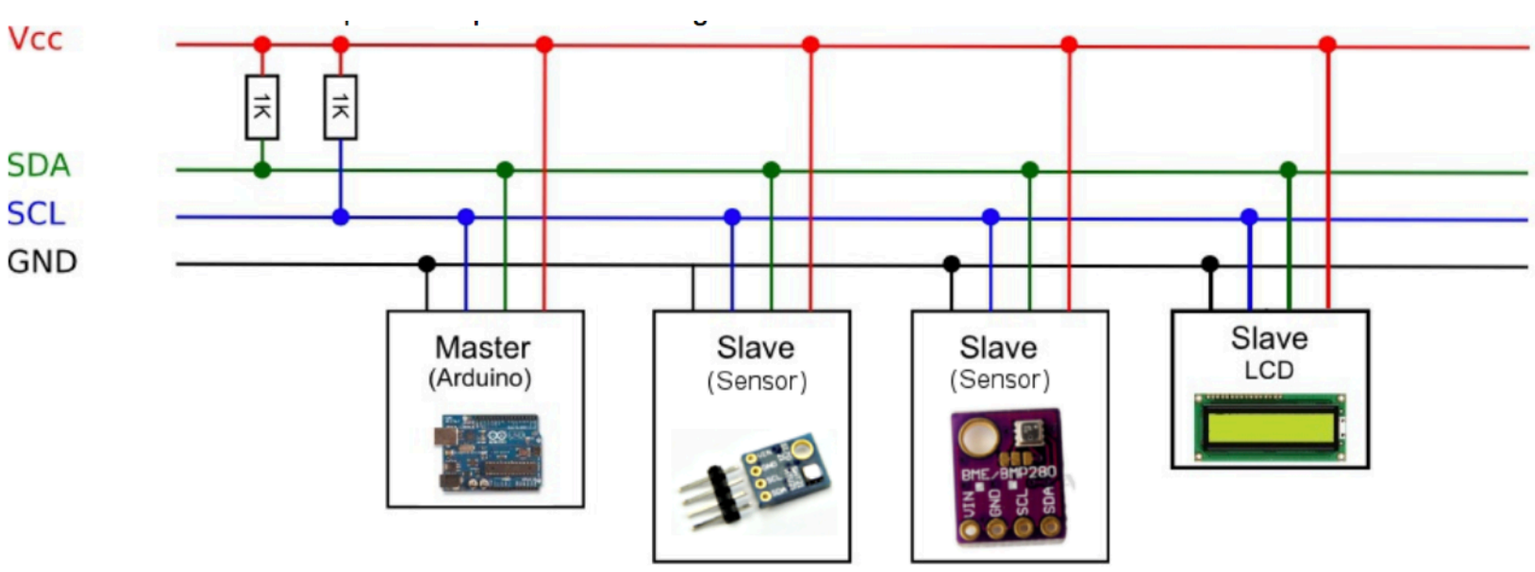


Hardware development board

- **Serial communication protocols**

Advantages of I2C

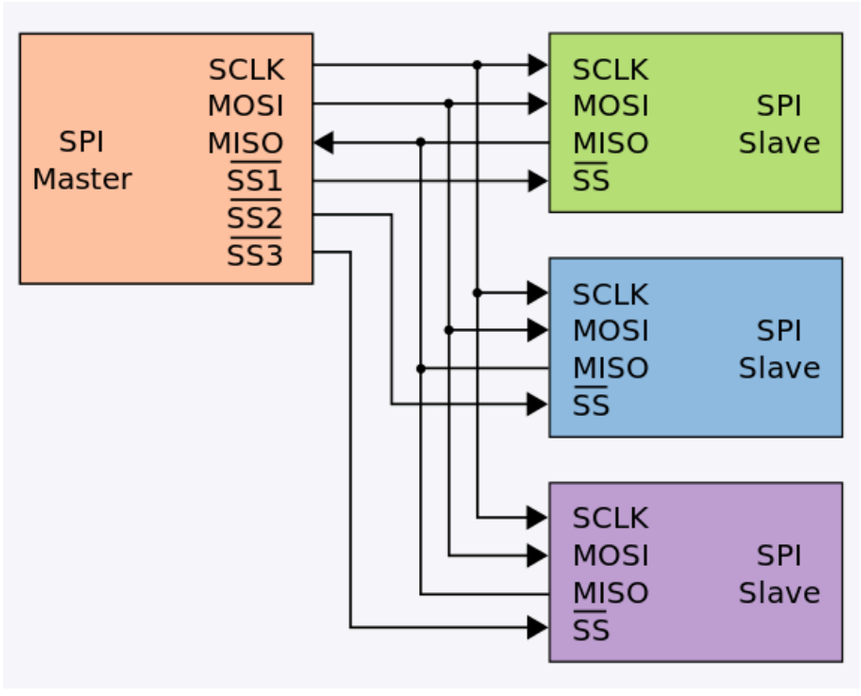
- It requires only two-wire, one wire for the data (SDA), and other wire for the clock (SCL).
- It provides the flexibility to the user to select the transmission rate as per the requirements.
- In I2C Bus, each device on the bus is independently addressable.
- It has the capability to handle multiple masters and multiple slaves on the I2C Bus.
- I2C provides ACK/NACK (acknowledgment/ Not-acknowledgement) features that provide help in error handling.



Hardware development board

- **Serial communication protocols**

  SPI (Serial Peripheral Interface): is four wire-based **full-duplex** communication protocol:
  - **MOSI** Master Out Slave In,
  - **MISO** Master In Slave Out,
  - **SCLK** Serial CLock which produces by the master,
  - **SS** Slave Select line which use to select specific slave during the communication.



Advantages of SPI

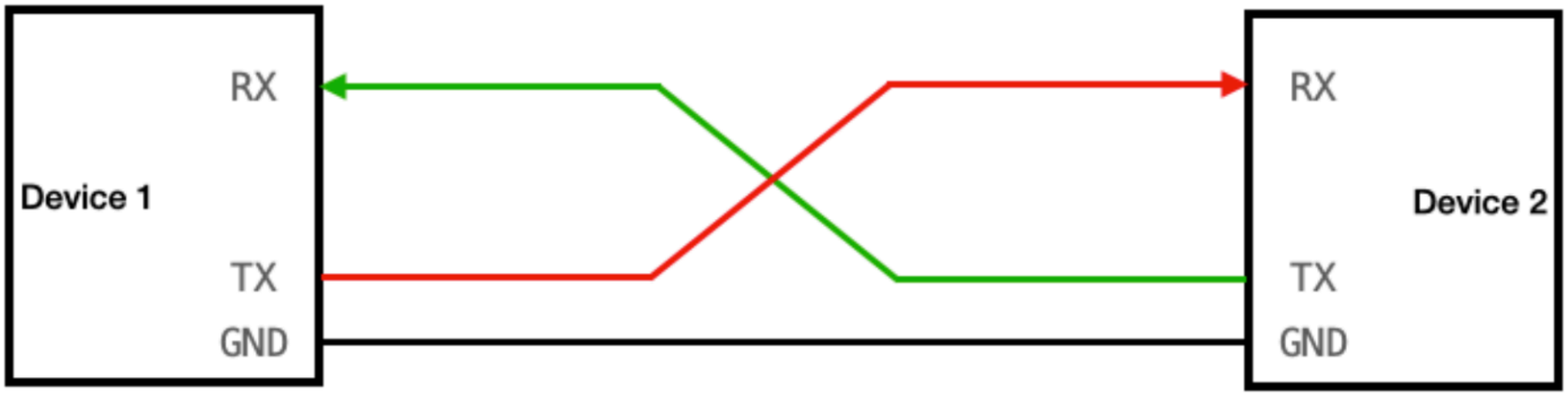
- There is no start and stop bits, so the data can be streamed continuously without interruption.
- No complicated slave addressing system like I2C.
- Higher data transfer rate than I2C (almost twice as fast).
- Separate MISO and MOSI lines, so data can be sent and received at the same time.
- Simple software implementation.

Hardware development board

- **Serial communication protocols**

UART (Universal Asynchronous Receiver-Transmitter):

- Is device-to-device communication protocol.
- Serial, asynchronous (the clock line is not required), full-duplex communication protocol.
- Has two unidirectional wires (RX and TX)



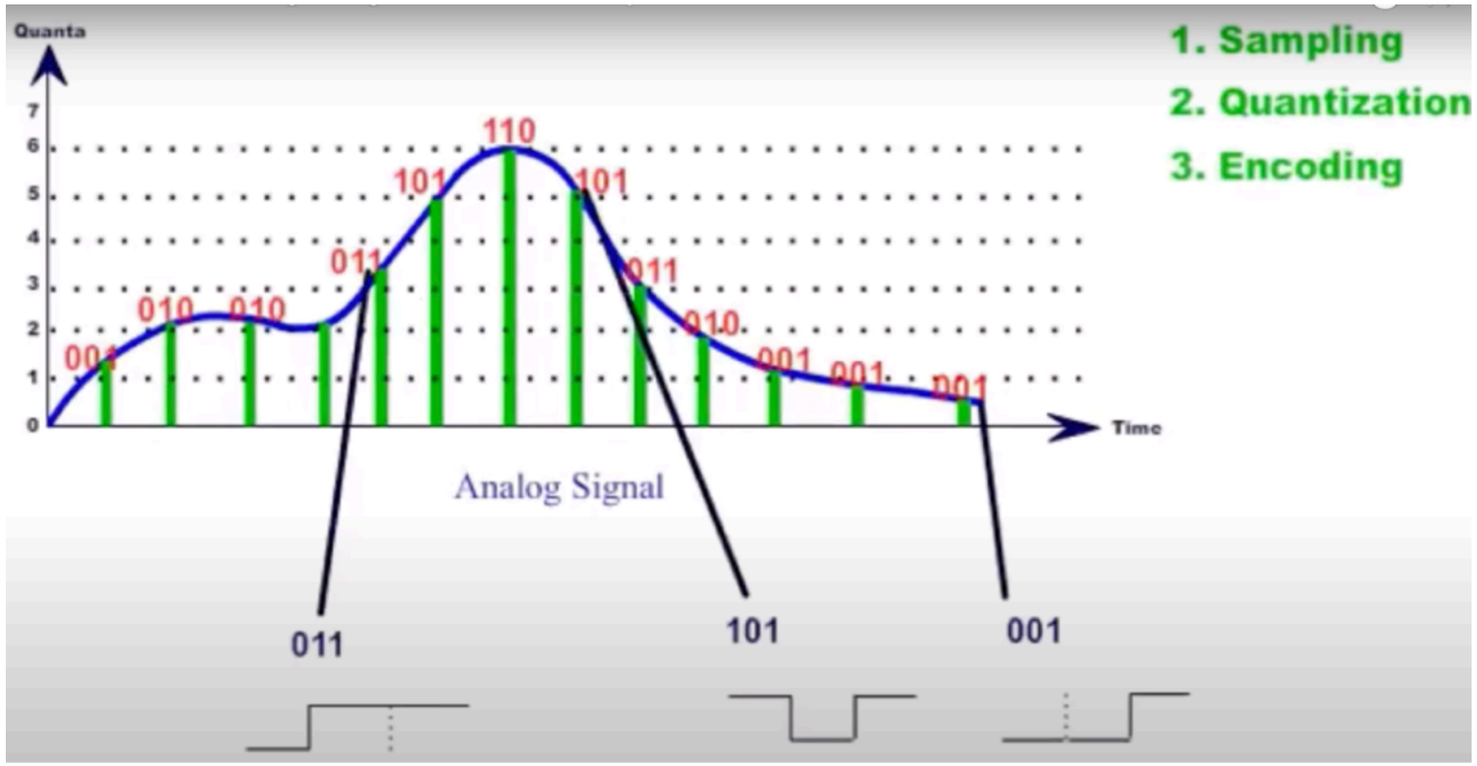
Hardware development board

- **Serial communication protocols comparison**

Protocol	UART	SPI	I2C
Complexity	Simple	Complex as device increases	Easy to chain multiple devices
Speed	Slowest	Fastest	Faster than UART
Number of devices	Up to 2 devices	Many, but gets complex	Up to 127, but gets complex
Number of wires	2	4	2
Duplex	Full Duplex	Full Duplex	Half Duplex
No. of masters and slaves	Single to Single	1 master, multiple slaves	Multiple slaves and masters

Hardware development board

- **ADC (Analog to Digital Converter)**

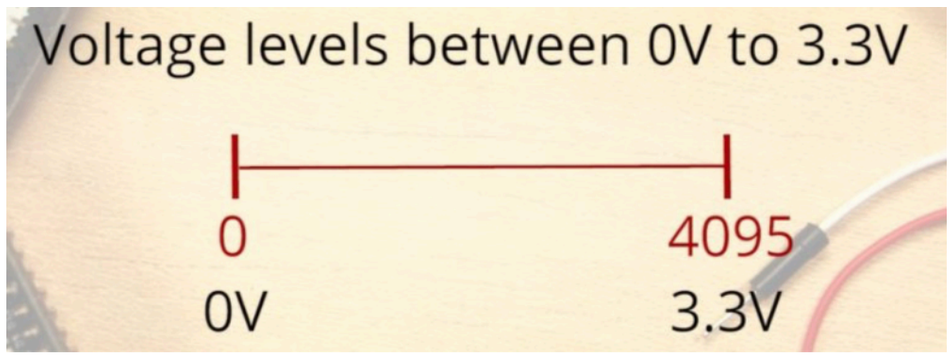


Hardware development board

- **ADC (Analog to Digital Converter)**

Reading an analog value with the ESP32 => measure varying voltage levels between 0 V and 3.3 V

The ADC resolution= 12bits => 2^{12} = 4096 values => from 0 to 4095

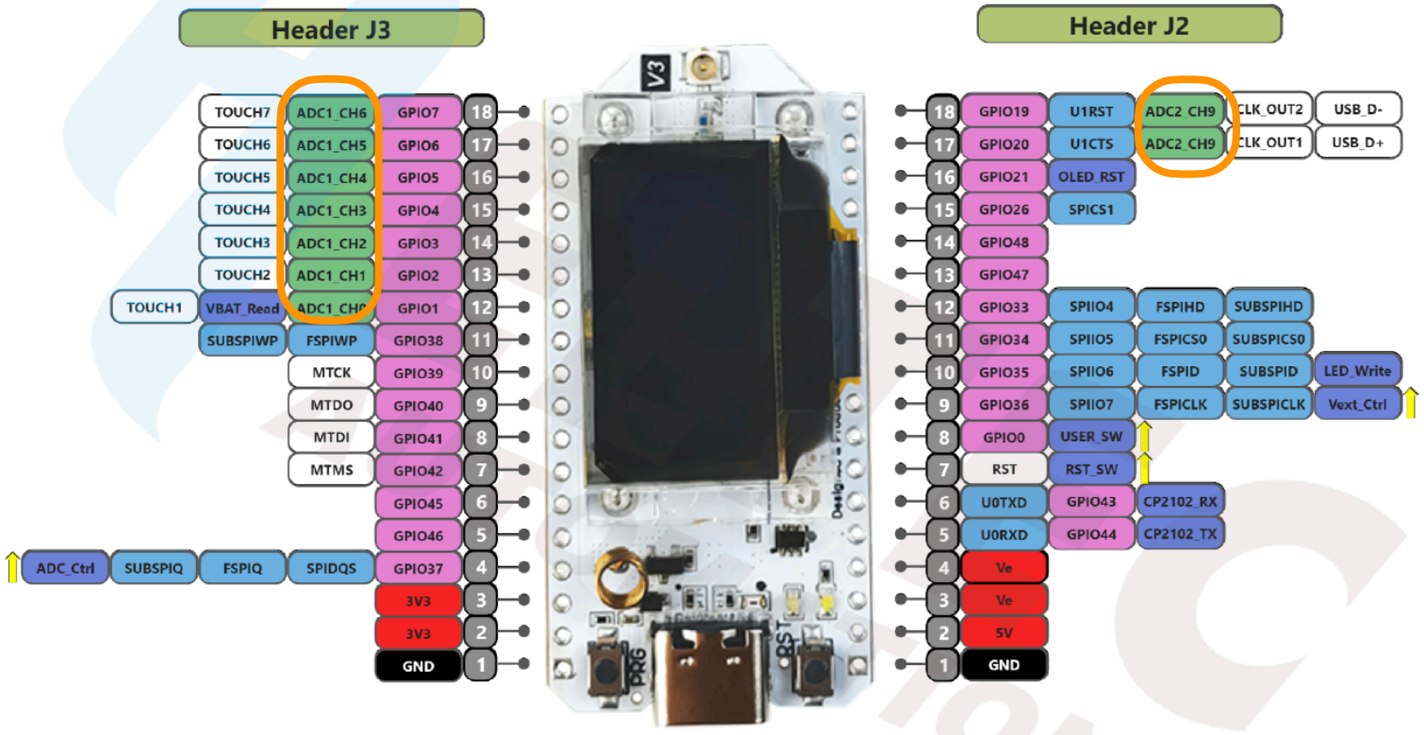


Hardware development board

- **ADC (Analog to Digital Converter)**

The ESP32 S3 integrates two 12-bit ADCs, supporting a total of 09 measurement channels (analog enabled pins). The ADC driver supports ADC1 (7 channels, attached to GPIOs 1 - 7), and ADC2 (02 channels, attached to GPIOs 19-20)

Reading an analog input with the ESP32 S3 using the Arduino IDE is as simple as using the analogRead() function. It accepts as argument, the GPIO you want to read: analogRead(GPIO);



Hardware development board

- **ADC (Analog to Digital Converter)**

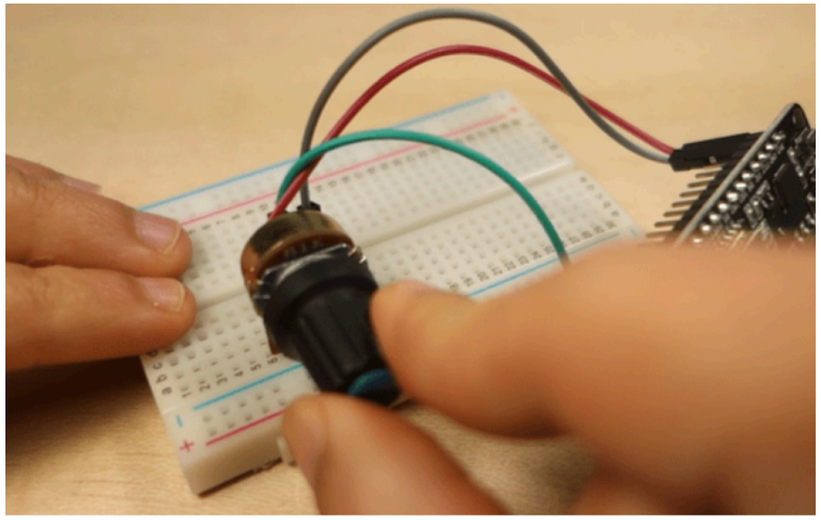
Example: Read Analog Values from a Potentiometer with ESP32

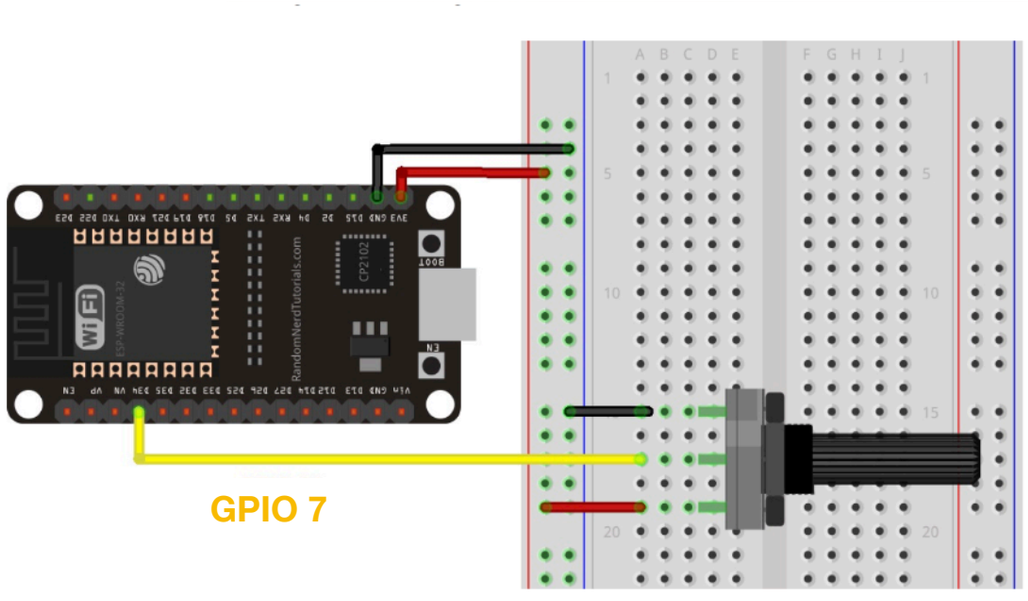
```
// Potentiometer is connected to GPIO 7 (Analog ADC1_CH6)
const int potPin = 7;

// variable for storing the potentiometer value
int potValue = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);
}

void loop() {
  // Reading potentiometer value
  potValue = analogRead(potPin);
  Serial.println(potValue);
  delay(500);
}
```



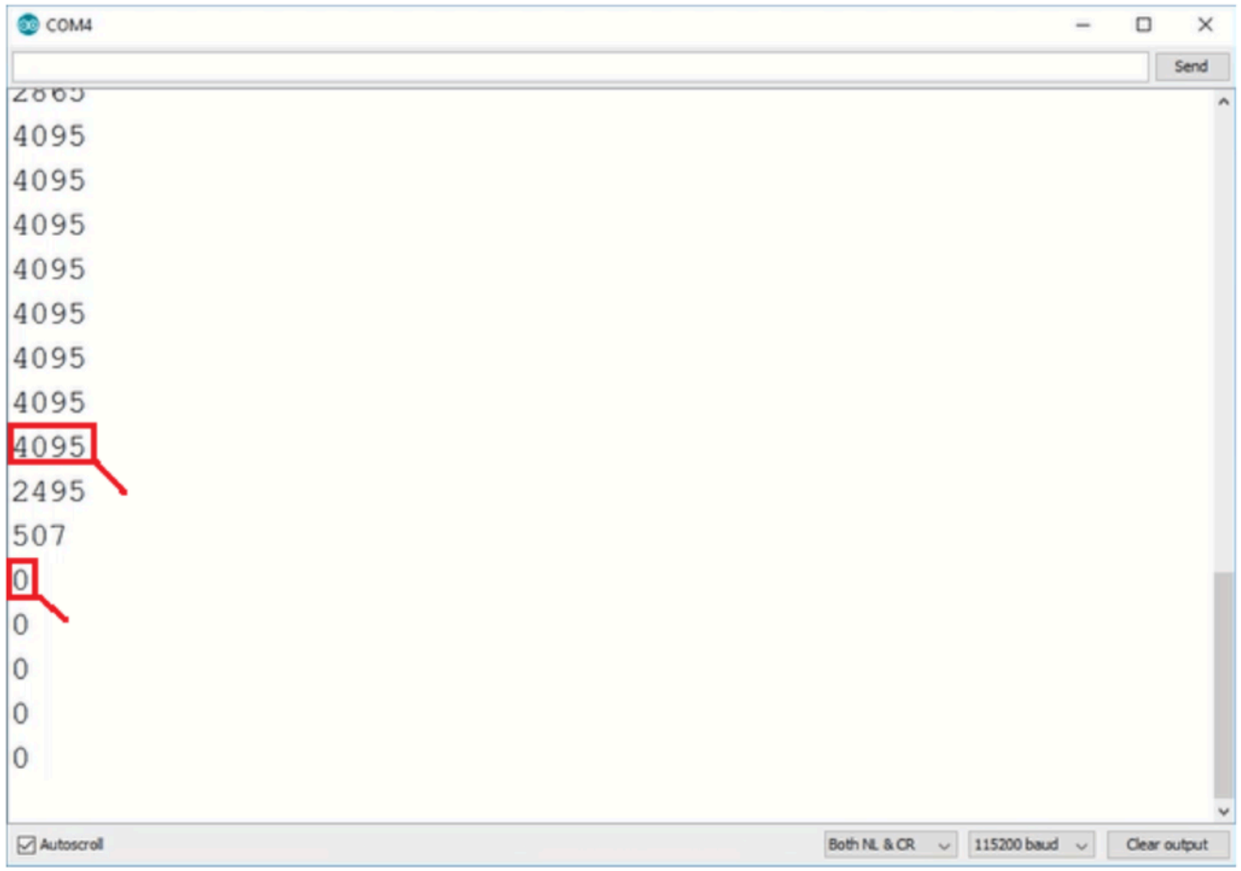


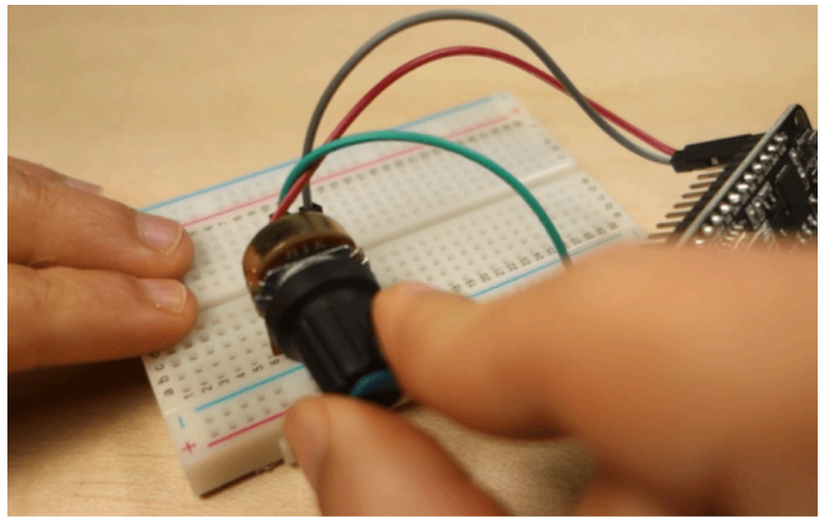
Hardware development board

- **ADC (Analog to Digital Converter)**

  Example: Read Analog Values from a Potentiometer with ESP32

The maximum value you'll get is 4095 and the minimum value is 0.



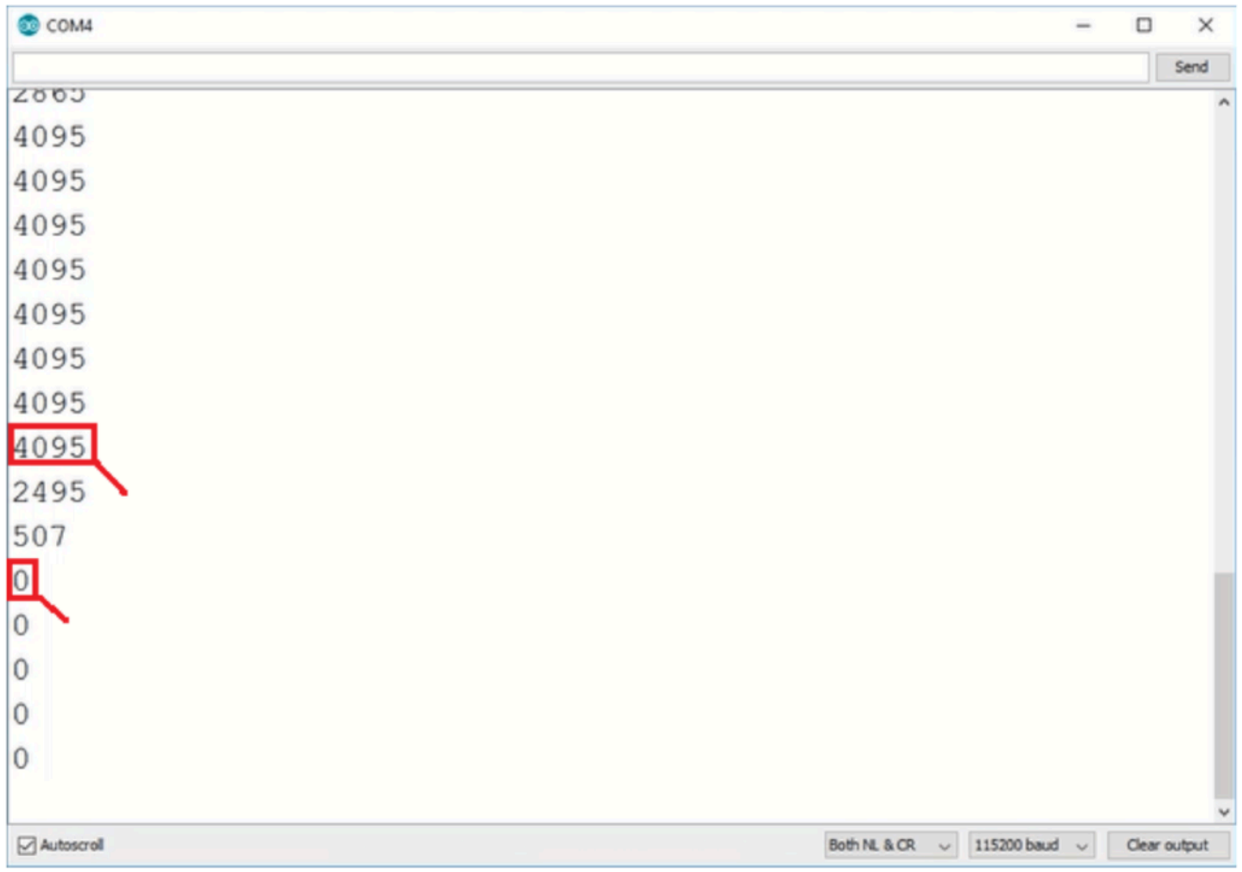


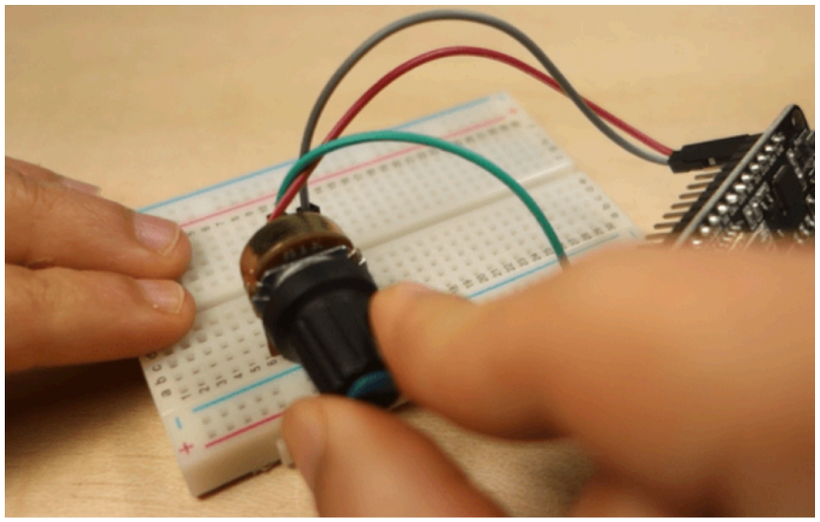
Hardware development board

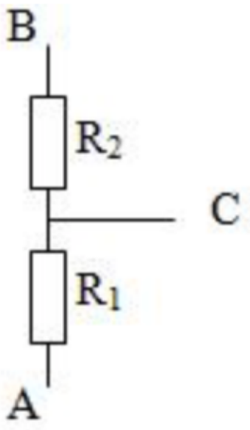
- **ADC (Analog to Digital Converter)**

Example: Read Analog Values from a Potentiometer with ESP32

The maximum value you'll get is 4095 and the minimum value is 0.







VC= VCC*R1/(R1+R2)
Voltage divider
RTotal=R1+R2

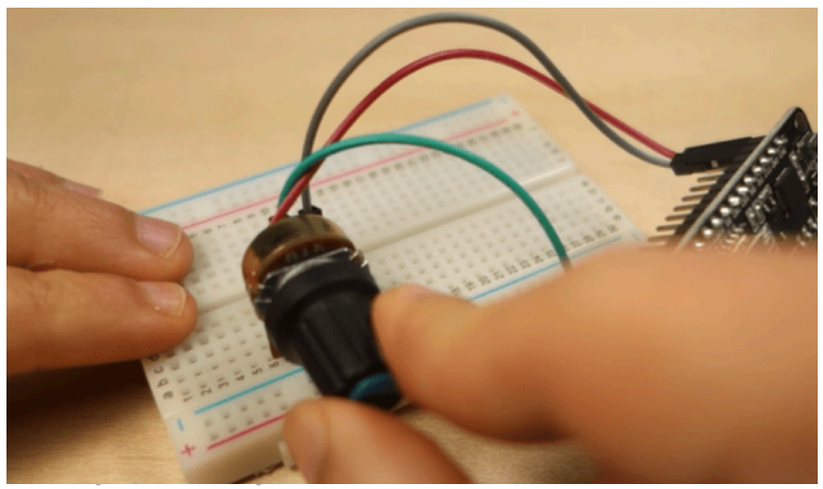
Potentiometer equivalent schematic

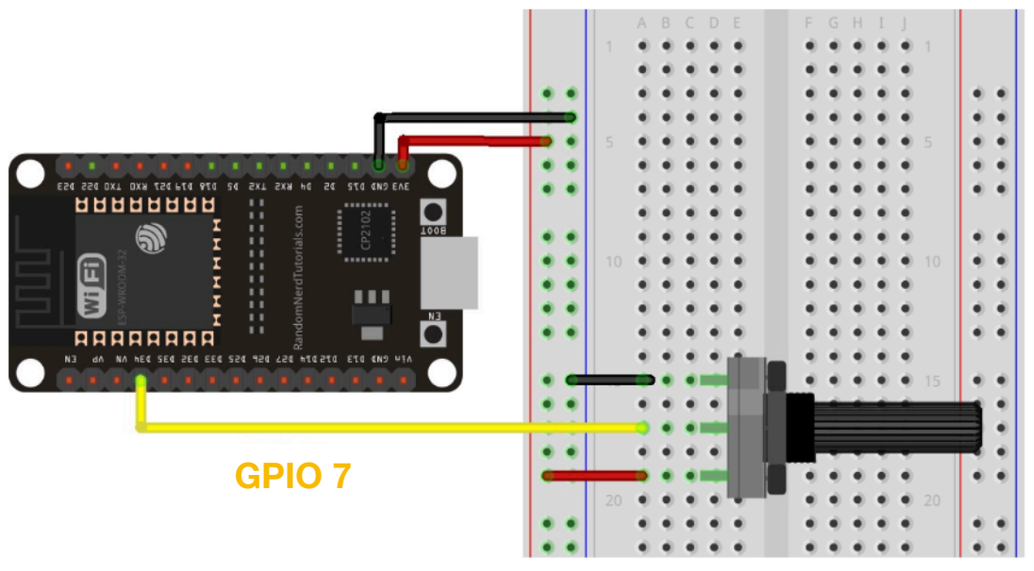
Hardware development board

- **ADC (Analog to Digital Converter)**

Example: Read Analog Values from a Potentiometer with ESP32 and display the resistance value

```
/* Potentiometer reading program */
// Constants
#define VIN 3.3 // V power voltage
#define R 10000 //ohm resistance value
// Parameters
const int potPin = 7; // Pin connected to sensor
//Variables
int potValue; // Analog value from the sensor
float res; //resistance value
void setup(void) {
  Serial.begin(115200);
}
void loop(void) {
  potVal = analogRead(potPin);
  res=sensorRawToPhys(potVal);
  Serial.print(F("Raw value from sensor= "));
  Serial.println(potVal); // the analog reading
  Serial.print(F("Physical value from sensor = "));
  Serial.print(res); // the analog reading
  Serial.println(F(" ohm")); // the analog reading
  delay(1000);
}
float sensorRawToPhys(int raw){
  // Conversion rule
  float Vout = float(raw) * (VIN / float(4095));// Conversion analog to voltage
  float phys = R *((Vout))/VIN; // Conversion voltage to resistance between GND and
  return phys;                  signal
}
```

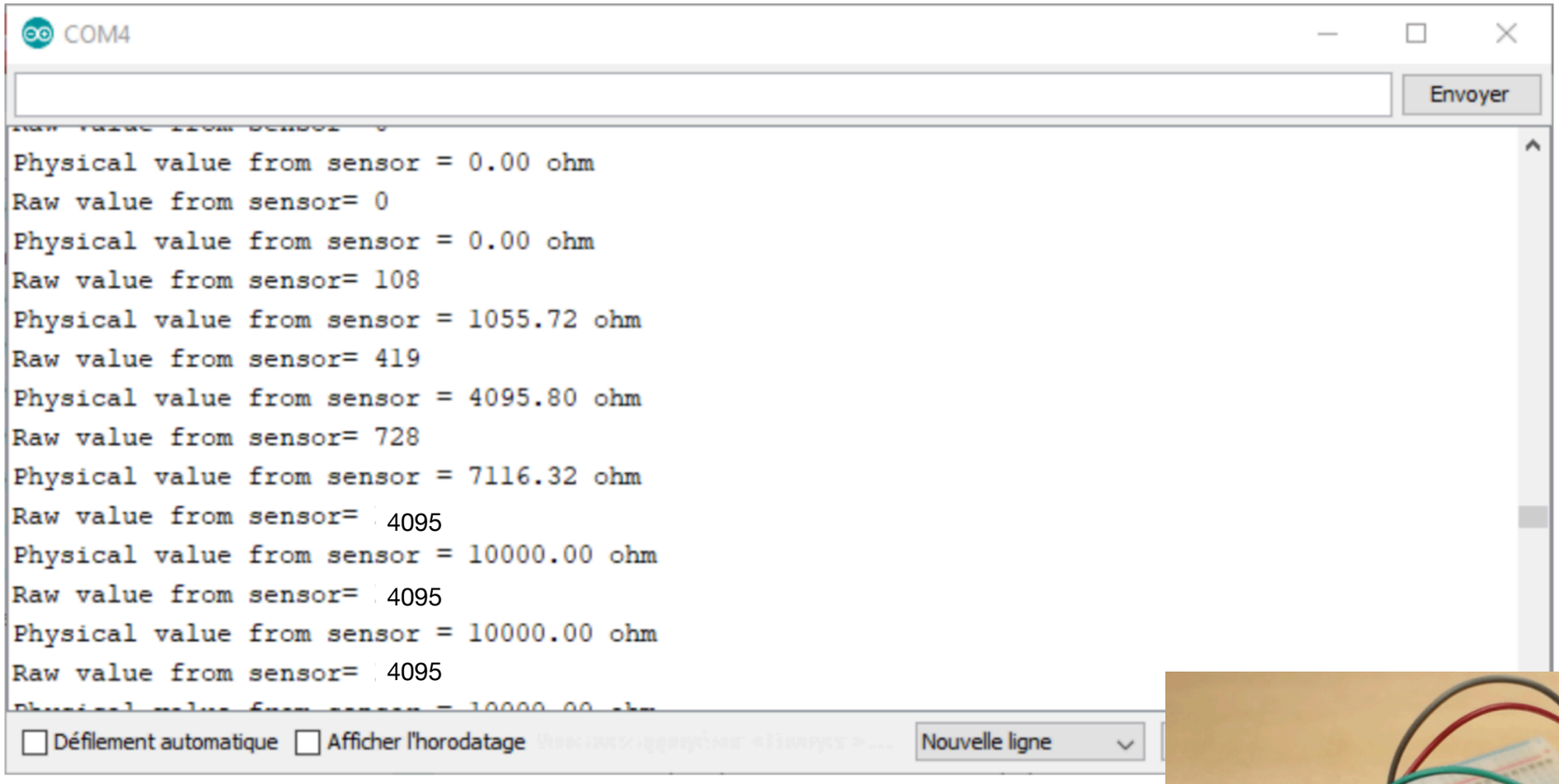


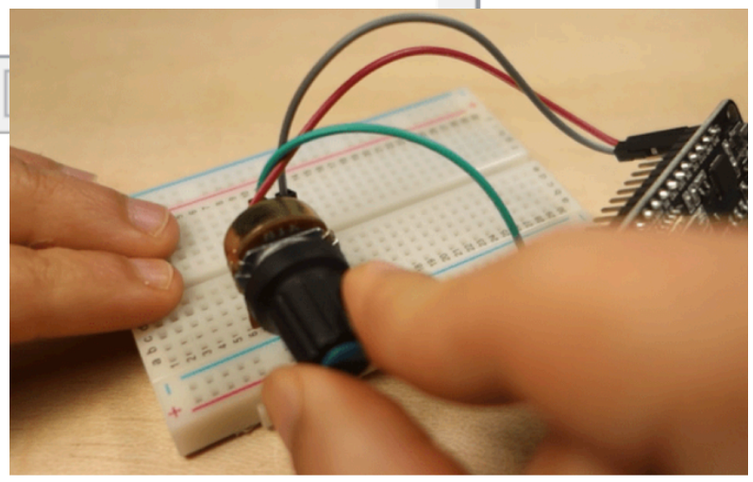


Hardware development board

- **ADC (Analog to Digital Converter)**

Example: Read Analog Values from a Potentiometer with ESP32 and display the resistance value





Survey

- **The fastest serial communication protocol is:**

1. SPI
2. I2C
3. UART

- **The ADC resolution in Wifi Lora 32 Board is**

1. 8 bits
2. 10 bits
3. 12 bits
4. 64 bits

Survey

- **In SPI communication protocol, MOSI line is used for:**

    1. Clock
    2. Data
    3. Both clock and data

- **A wireless data communication covered the highest range distance is:**

    1. WiFi
    2. BLE
    3. LoRaWan