Tutorial ESP32-S3 I

using Arduino IDE software

Learning Outcomes

> Connecting hard circuit to soft circuits
> Introduction to Physical Computing + XIAO ESP32 S3

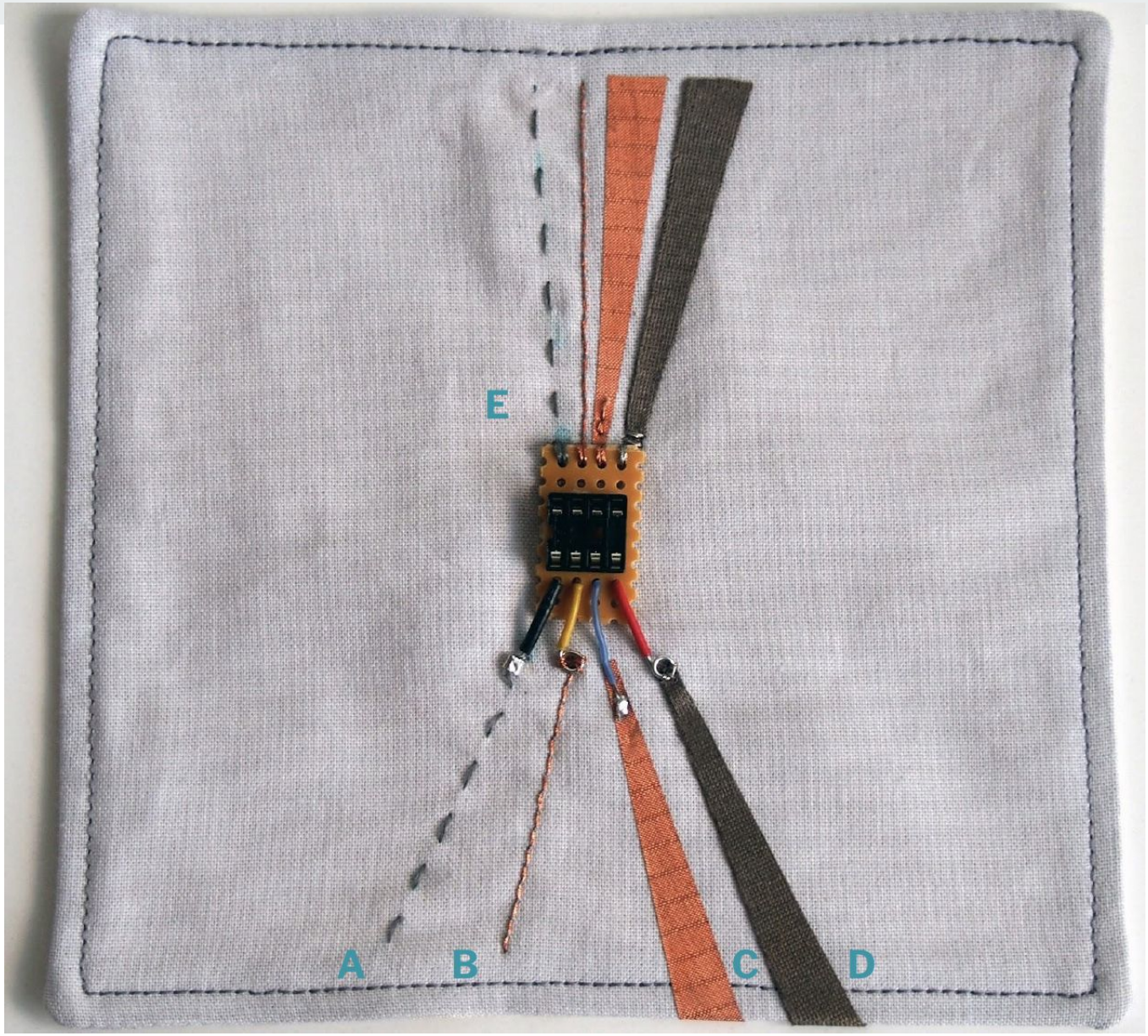
> Set up circuit to connect your sensor to XIAO ESP32 S3

> Write basic code, read sensor data

> Extra: connect two sensors, two data points in code

Hard to soft circuit.

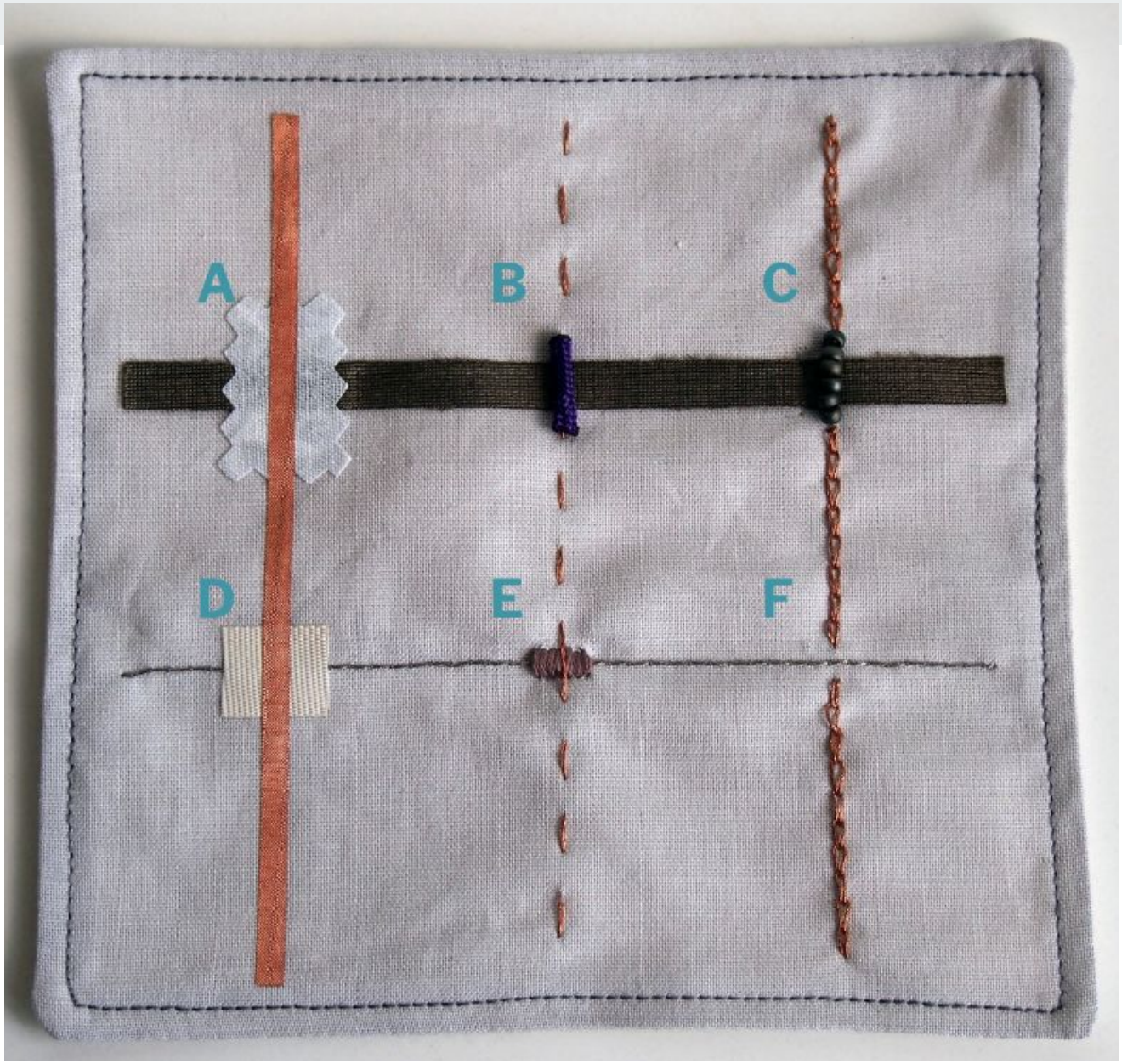
- A} Use a crimp bead to connect to the thread;
- B} Use pliers to create a sewable hole with a wire (same principle as we did with the LED);
- C} solder on fabric > Needs to be tested before hand and depending on the fabric if it works or not
- D} sew wire directly to fabric;
- E} Desoldering the pins: leaves you with metal loops which you can sew conductive thread in;
- F} Using items like snap-buttons or safety pins.



https://www.computationalcraft.io/about/

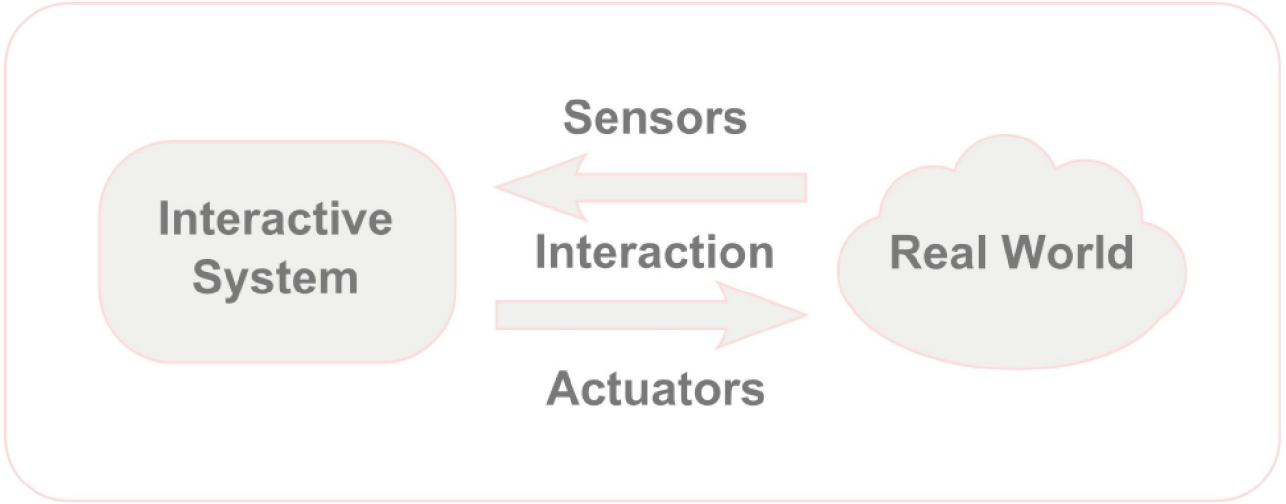
Hard to soft circuit: overlapping traces.

- A} Fabric (with iron-on adhesive);
- B} Paracord;
- C} Beads;
- D} Tape;
- E} Embroider;
- D} Sew under.

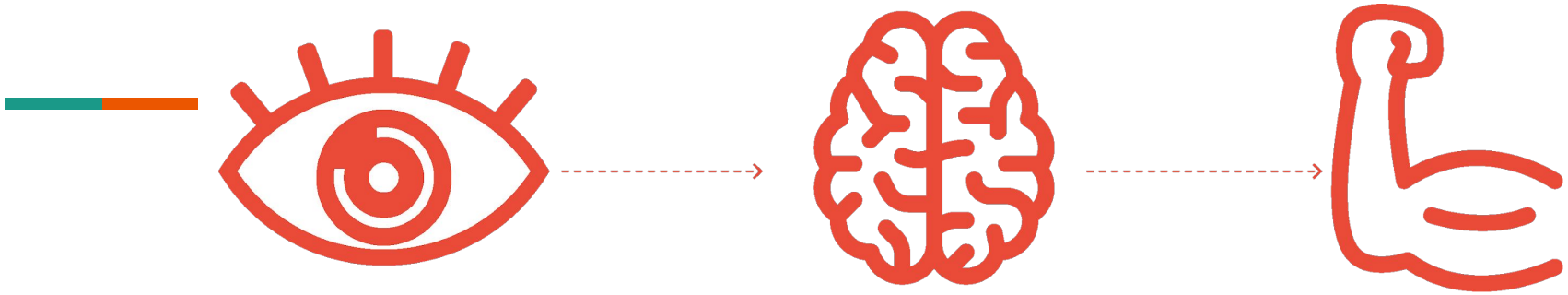


https://www.computationalcraft.io/about/

PHYSICAL COMPUTING



ELECTRIC SYSTEM



Senses

Brain

Action

INPUTS
Sensors

Take information

PROCESSING
Microcontrollers

Read and process the information

OUTPUTS
Actuators

React to the information

ELECTRIC SYSTEM:
Processors

HARDWARE

Processors (CPU)
Microcontrollers
Printed circuit board (PCB)

SOFTWARE

Code:
Java, C, C++, Pyhton, C#

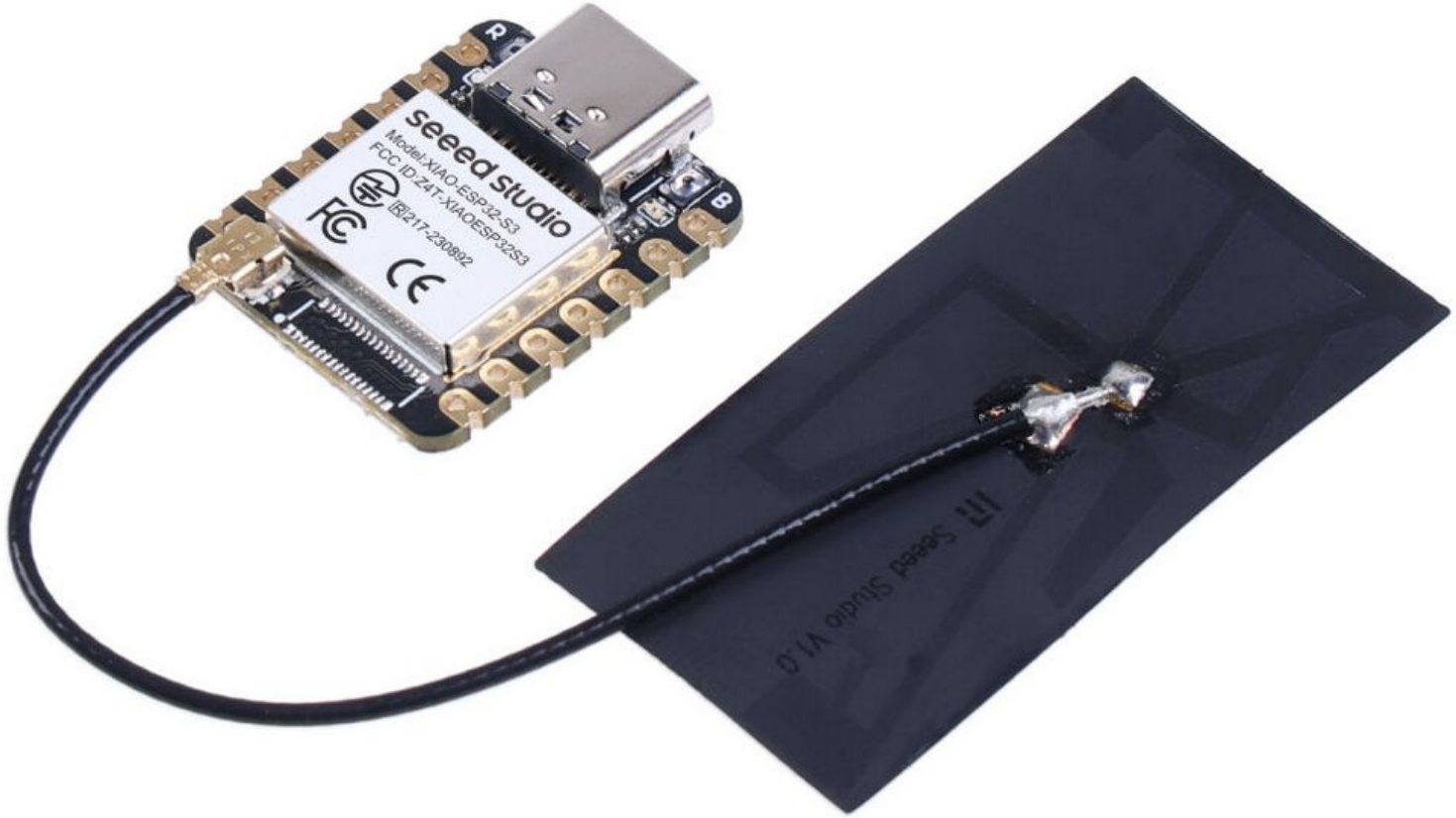
```
function execute(sessId, raw) {
    var tokens = raw.match(/(\w+|[+=();])/g);
    var peek = _ => tokens[0];
    var eat = _ => tokens.shift();
    var ate = x => peek() === x && eat();
    var want = x => {
        if (!ate(x)) throw 'Expected \"' + x + '\"
    };

    function statement() {
        scope[sessId][eat()] = [want('='), expr(),
    }

    function expr() {
        for (var v = term(); ate('+'); v = v + term
        return v;
    }

    function term() {
        for (var v = atom(); ate('('); v = v([expr(
        return v;
```

HARDWARE:

Wiki “Getting Started with Seeed Studio XIAO ESP32S3 Series”:
https://wiki.seeedstudio.com/xiao_esp32s3_getting_started/

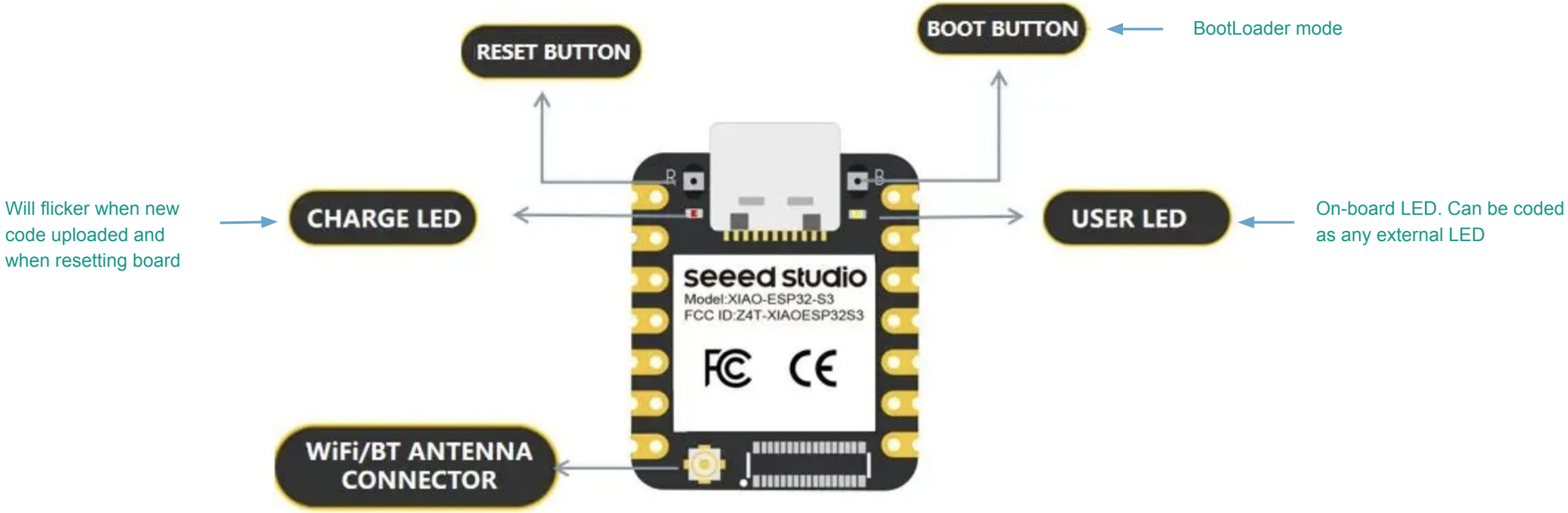
Seeed Studio XIAO ESP32-S3

- Arduino / MicroPython supported
- Supports 2.4GHz WiFi and BLE 5.0 dual wireless communication
- Supports 100m+ remote communication when connected with U.FL antenna



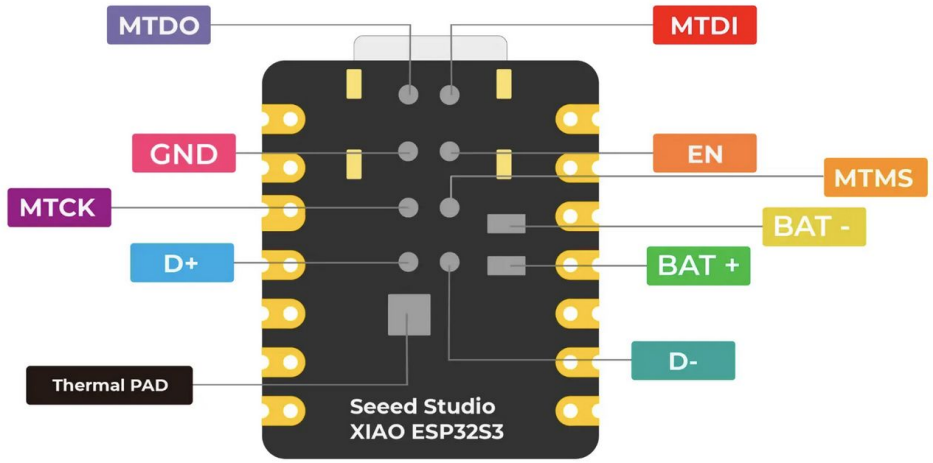
Read all technical specifications and data-sheet here:

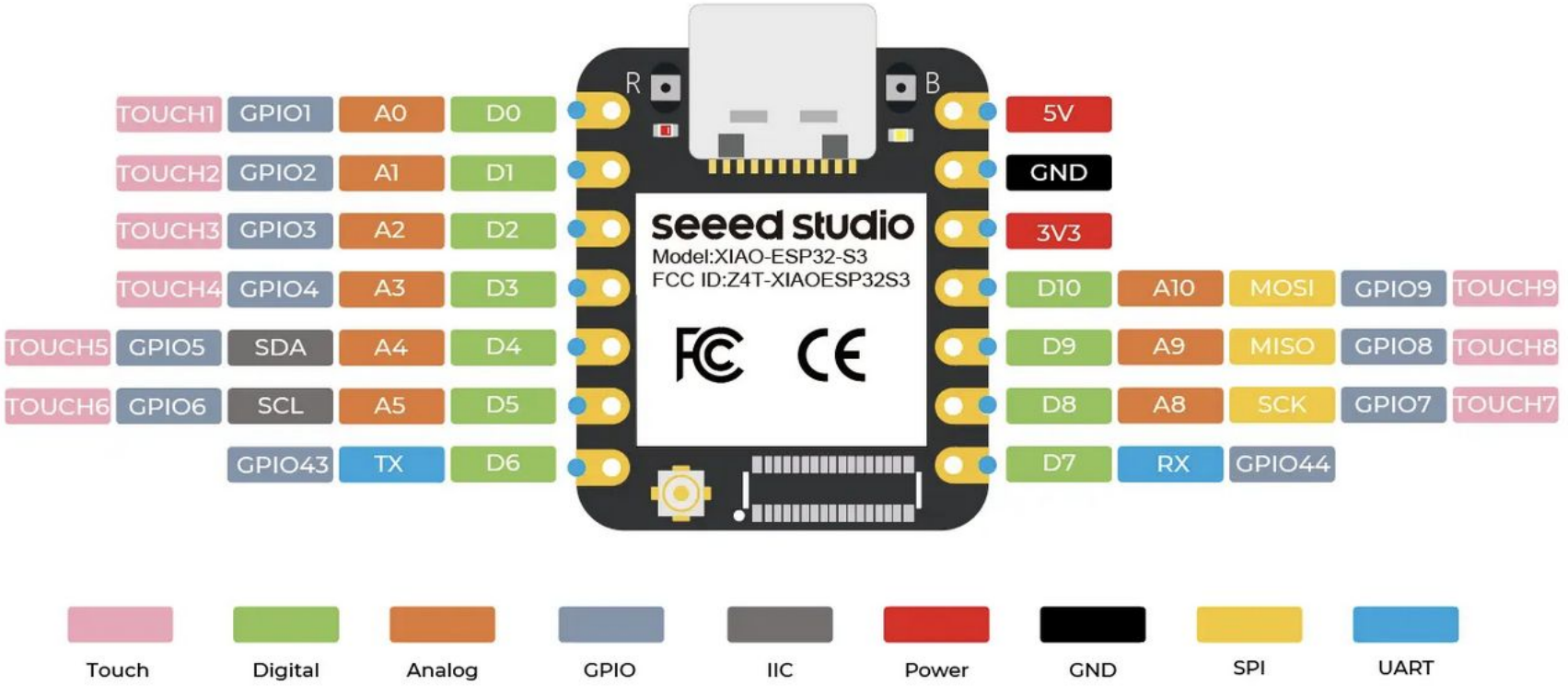
https://files.seeedstudio.com/wiki/SeeedStudio-XIAO-ESP32S3/res/esp32-s3_datasheet.pdf



OVERVIEW

XIAO ESP32 S3





PIN OUT

XIAO ESP32 S3

A pin is how **inputs** (buttons, etc) and **outputs** (LEDs, speakers, etc) communicate with the Microcontroller.

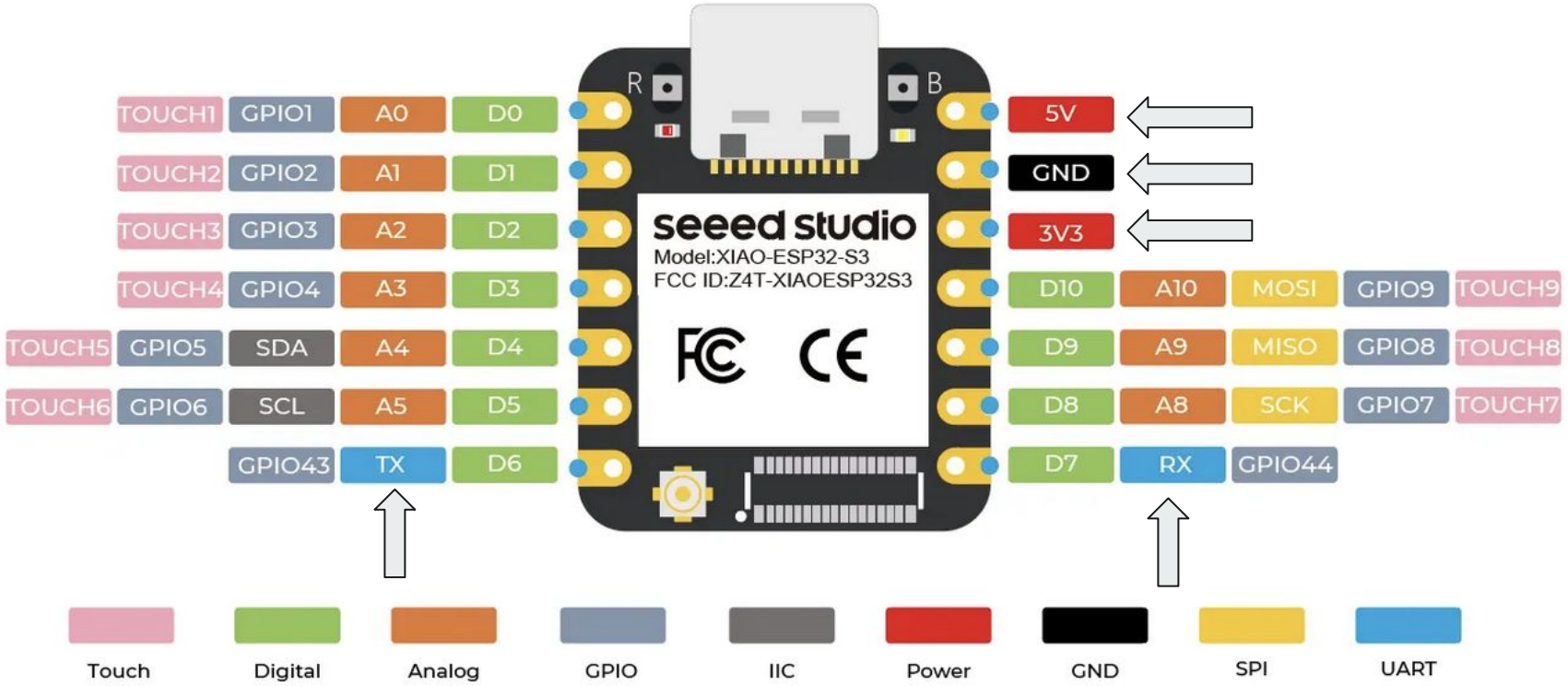
TX/RX

//serial - transmit/receive

1 ground pin

2 power pins

// 5V, 3.3V



POWER PINS

XIAO ESP32 S3

10 Digital I/O Pins

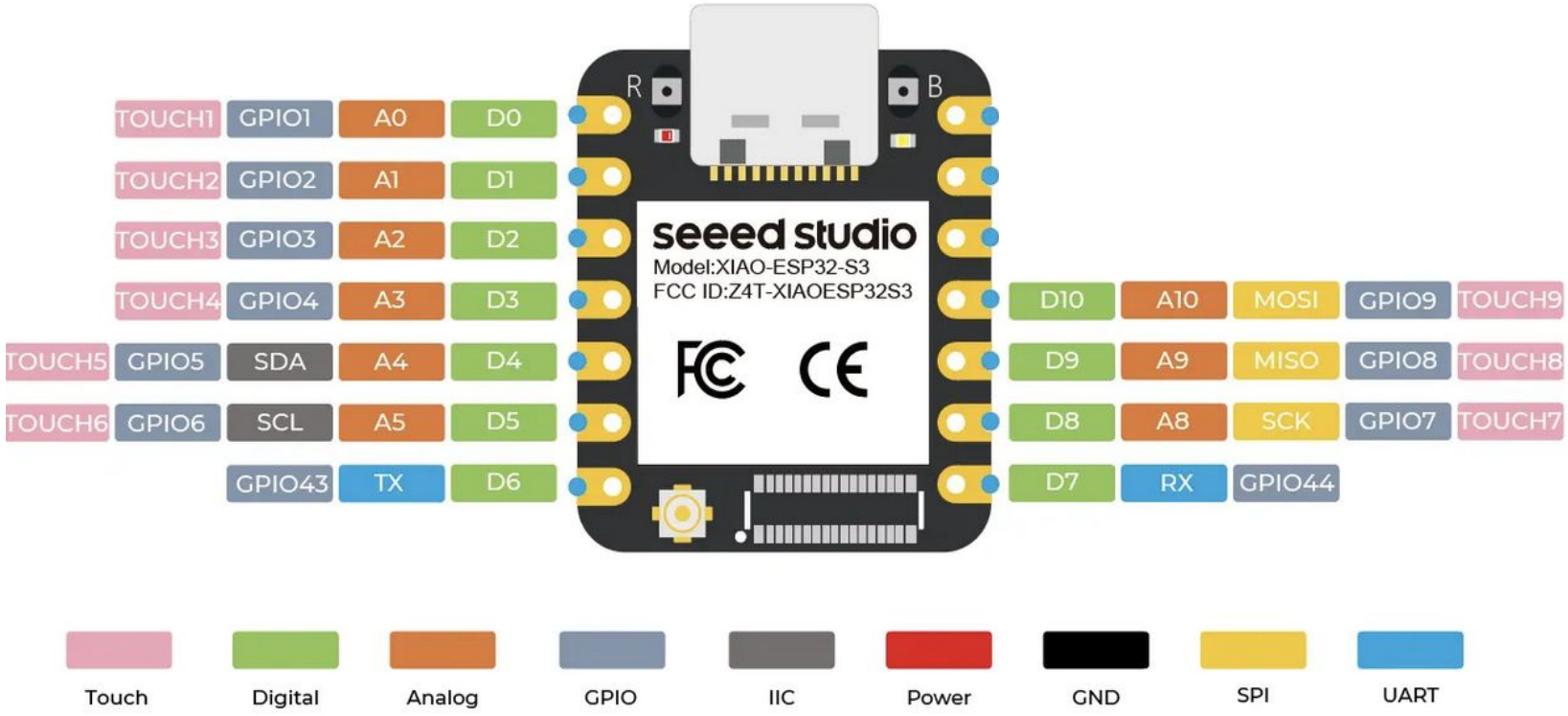
On Off

High Low

10 Analog Input pins

i2C communication SDA/ SCL

manage all data exchange with I2C-compatible components



DATA PINS

XIAO ESP32 S3

SOFTWARE: Arduino IDE

Open-source electronic prototyping platform enabling users to create interactive electronic objects.

Alternative software: Visual Studio Code with PlatformIO extension

SENSORS

Detect events or changes in its environment and send the information to other electronics.

*Can be digital and analog, but most **commonly are analog (range of values)***

Touch
Light
Pressure
T°
etc..

ACTUATORS

Component of a machine that is responsible for moving and controlling a mechanism or system:

Movement
Sound
Light
Signals
etc..

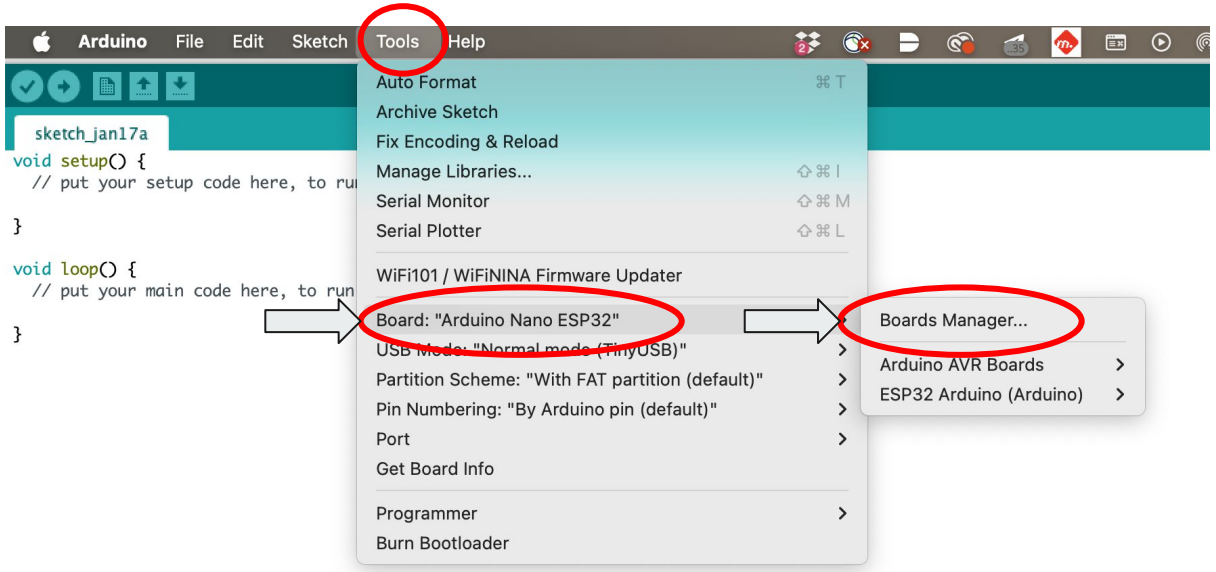
*Can be digital and analog, but **most commonly are digital (0 or 1**)*

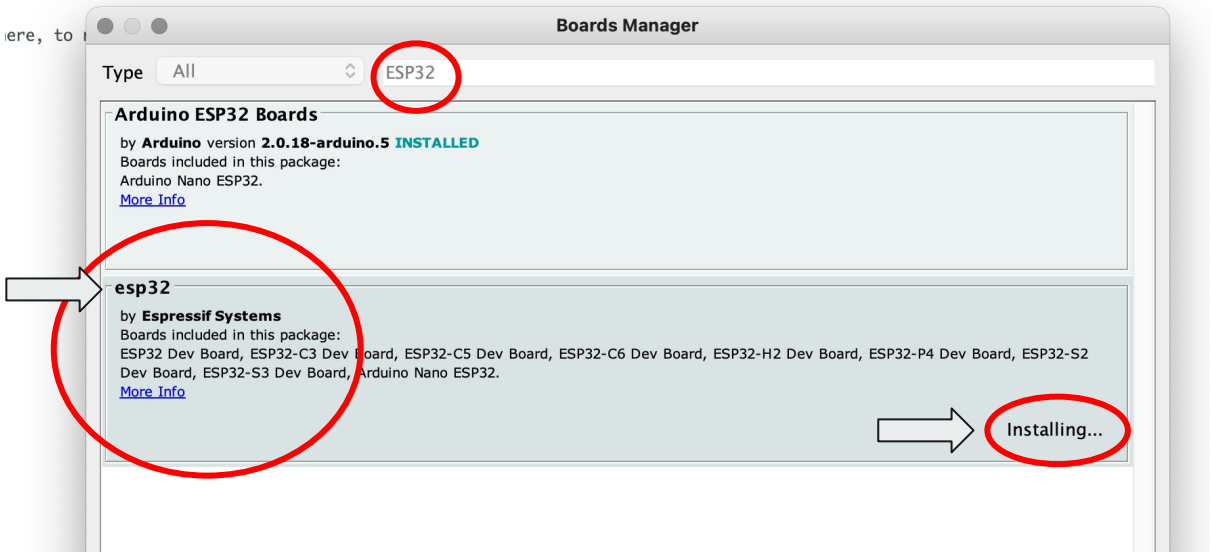
GETTING STARTED WITH THE XIAO ESP32 S3

Step 1. Download and Install the stable version of Arduino IDE according to your operating system.
https://www.arduino.cc/en/software/

Step 2. Launch the Arduino application.

Step 3. Add ESP32 board package to Arduino IDE.

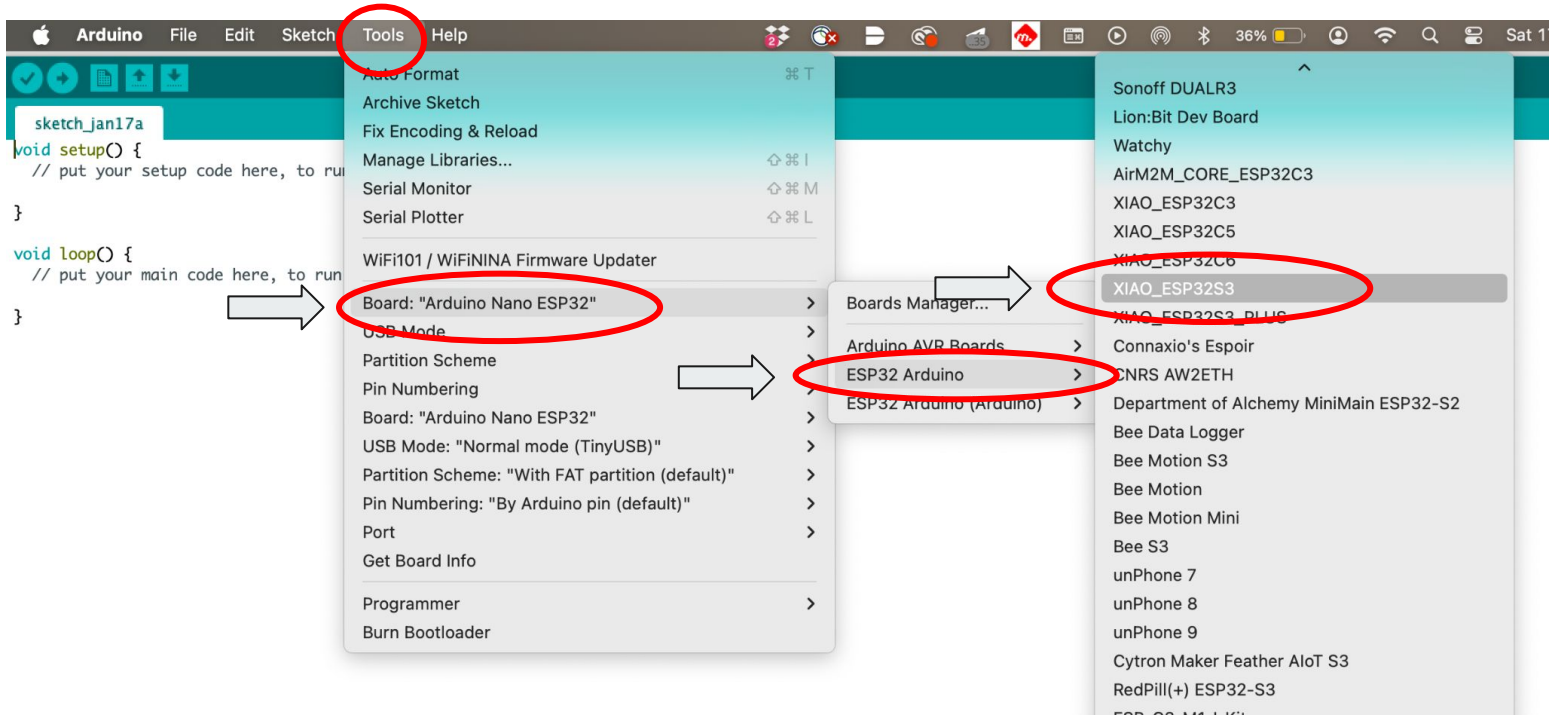


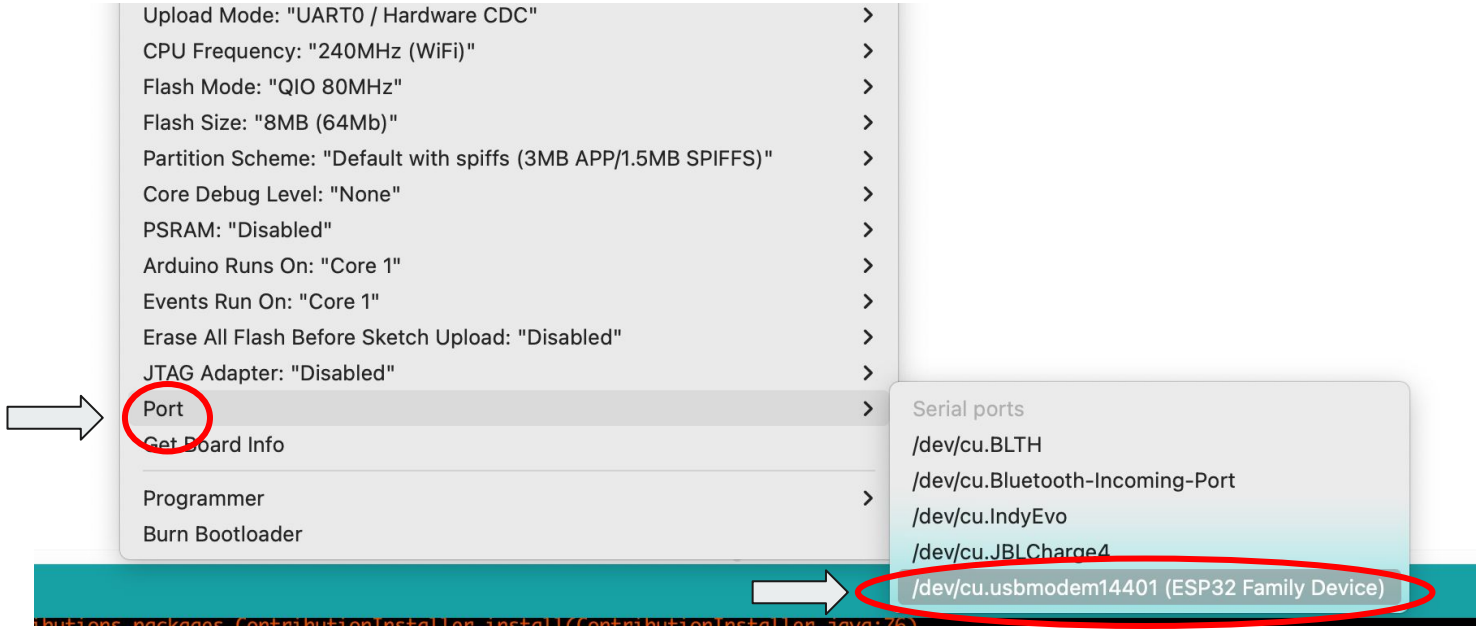


Step 4. Select your board and port.
At the top of the Arduino IDE you can directly select the port. This will probably be the one with "usbmodem" or "usbserial" in the name.

If you're unsure, unplug and plug again to see which port is missing.

Select **XIAO_ESP32S3**.





General problems on first use:

- The XIAO is connected to the computer, but *no port number* is found.
- The XIAO is connected, and a port number appears, but the *program upload fails*.

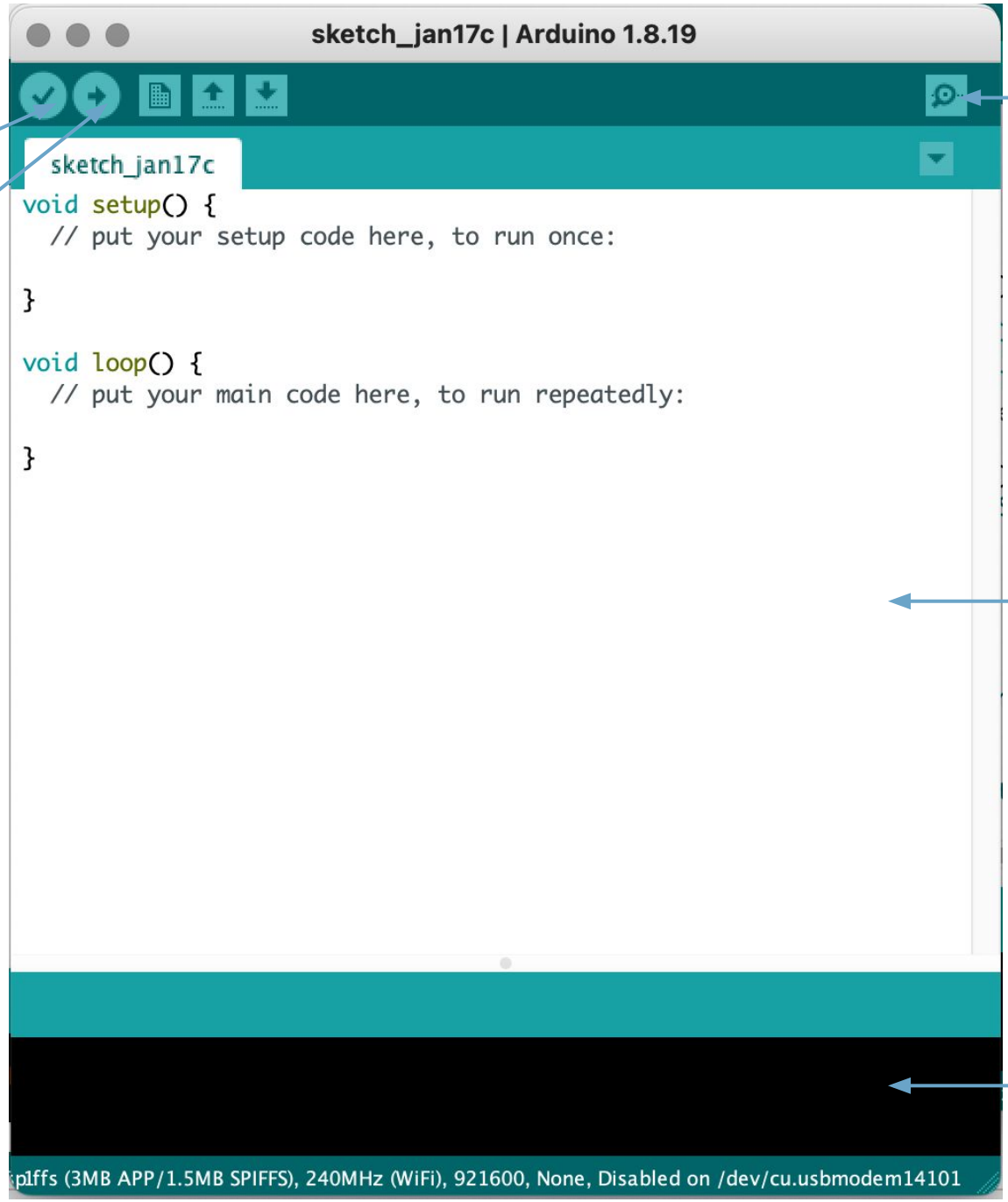
Use BootLoader Mode:
https://wiki.seeedstudio.com/xiao_esp32s3_getting_started/#bootloader-mode

- **Step 1**. Press and hold the B00T button on the XIAO ESP32S3 without releasing it.
- **Step 2**. Keep the B00T button pressed and then connect to the computer via the data cable. Release the B00T button after connecting to the computer.
- **Step 3**. Upload the **File > Examples > 01.Basics > Blink** program to check the operation of the XIAO ESP32S3.

¿How to write code in Arduino?

C++ *

ARDUINO IDE



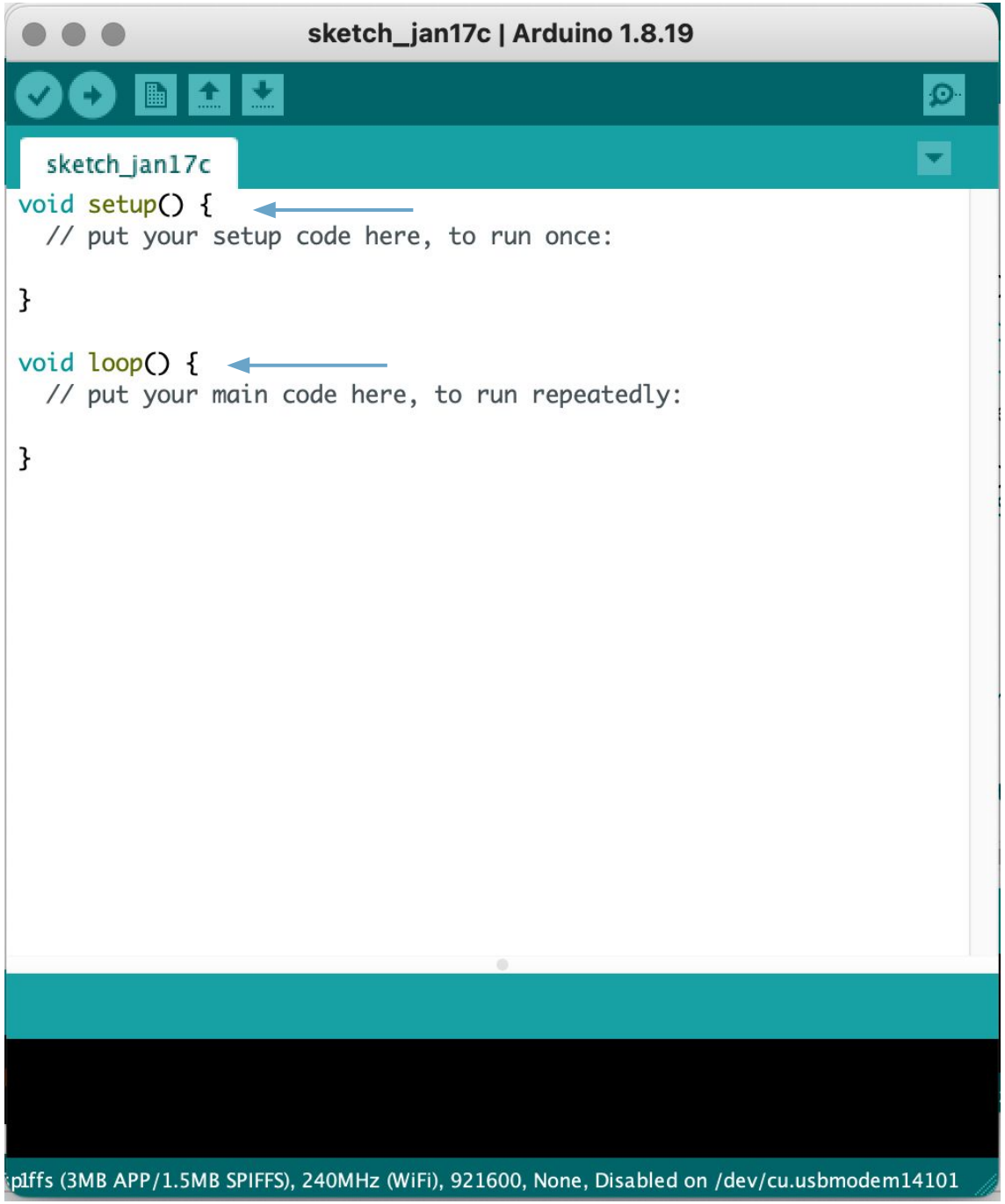
Compile code

Upload code

Serial Monitor

Code lines:
C++

Errors



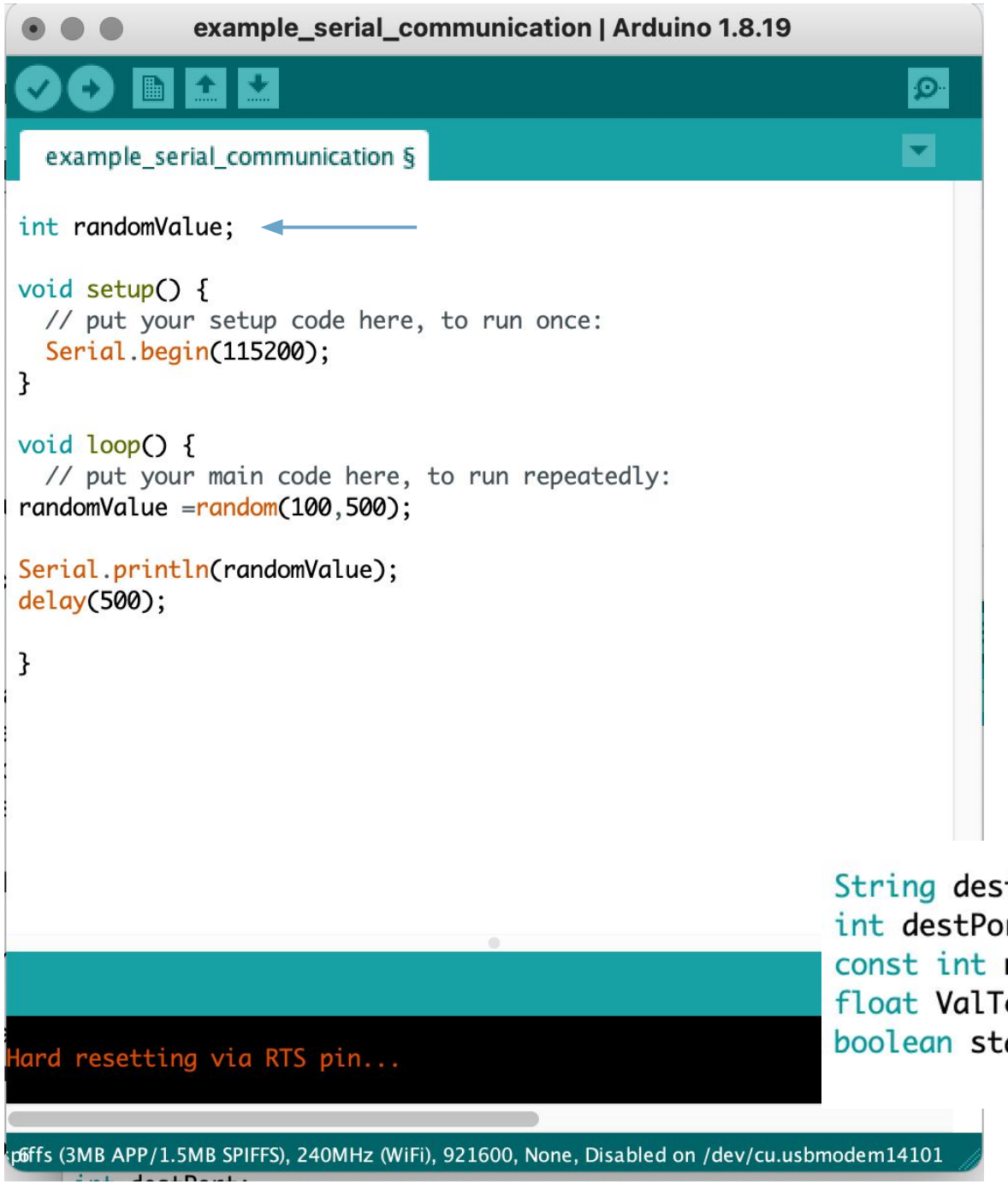
Basic functions of the code which control the structure:

void setup() {}

Executes just **1** time, **after powering up the microcontroller or after resetting it**.

void loop() {}

Executes **infinite** times

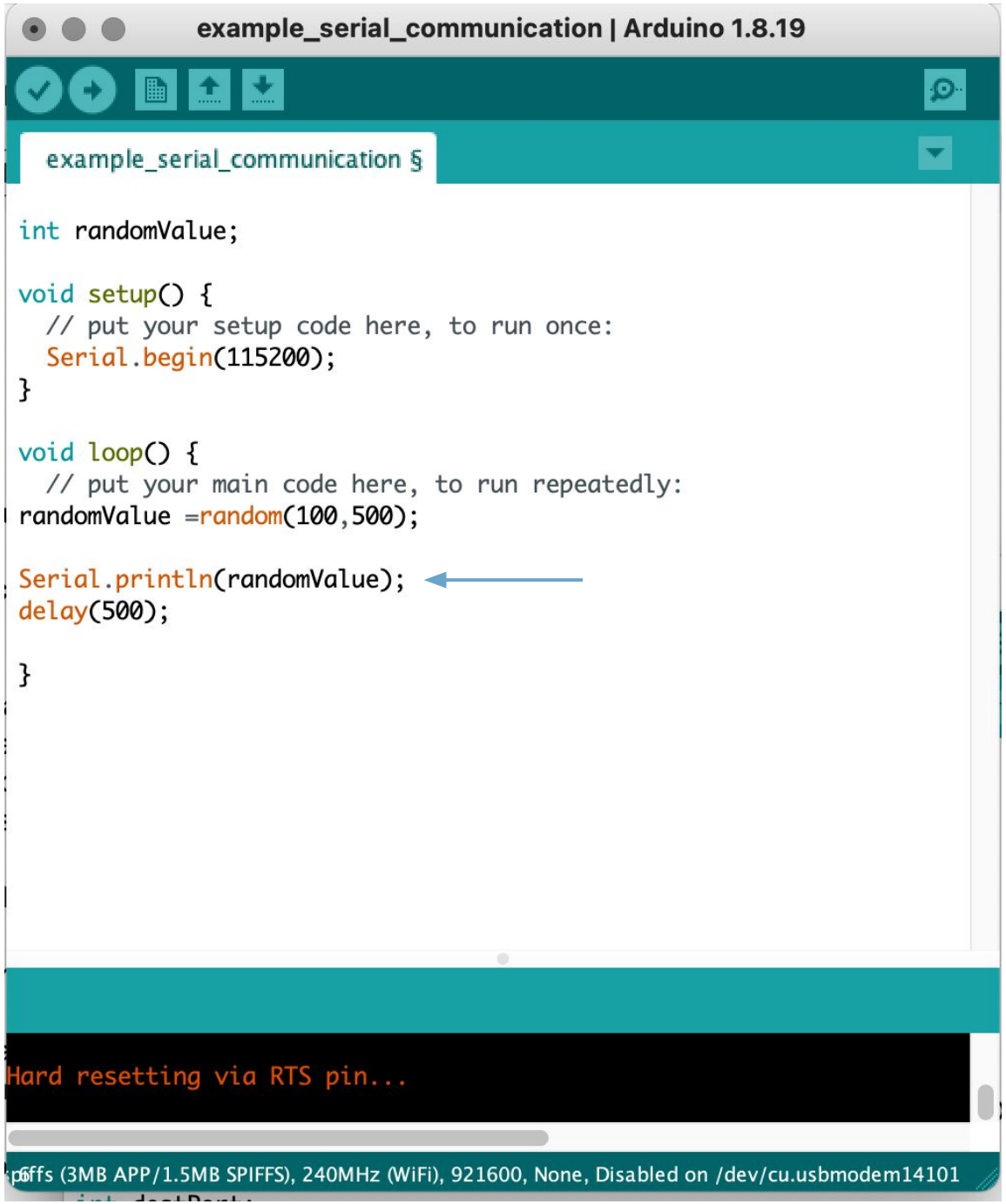


Above setup is where you can declare **global** (scope) variables.

A variable is a way of naming and storing a value for later use by the program, such as data from a sensor or an intermediate value used in a calculation.

Changes to light blue.
int is the most common

```
String destIp;                //multiple characters e.g. IP address
int destPort;                 //integer number, can be replaced by new values
const int numReadings = 10; //constant integer number, can't be replaced
float ValTemp;                //float number, number with decimals
boolean state = true;         //0 or 1 variable, true or false
```



Segmenting code into functions allows a programmer to create modular pieces of code that perform a defined task and then return to the area of code from which the function was "called".

Changes to colour orange.
Always follow by ()

1) Start serial communication
Serial.begin (speed); // e.g. 9600bps;

2) Assign component to PIN number and decide if it is input or output
pinMode (pin number, OUTPUT or INPUT);

3) Give it some time before it executes the next line of code
delay (time in miliseconds); //1000 = 1 second

4) Apply 5V to a particular Pin
digitalWrite (pin number, HIGH or LOW); // if actuator is digital
analogWrite (pin number, 0 to 255); // if actuator is analog

5) Read incoming data from digital or analog sensor
digitalRead (pin number);
analogRead (pin number);

6) Write to Serial Monitor
Serial.println ("message");
Serial.println (variable);

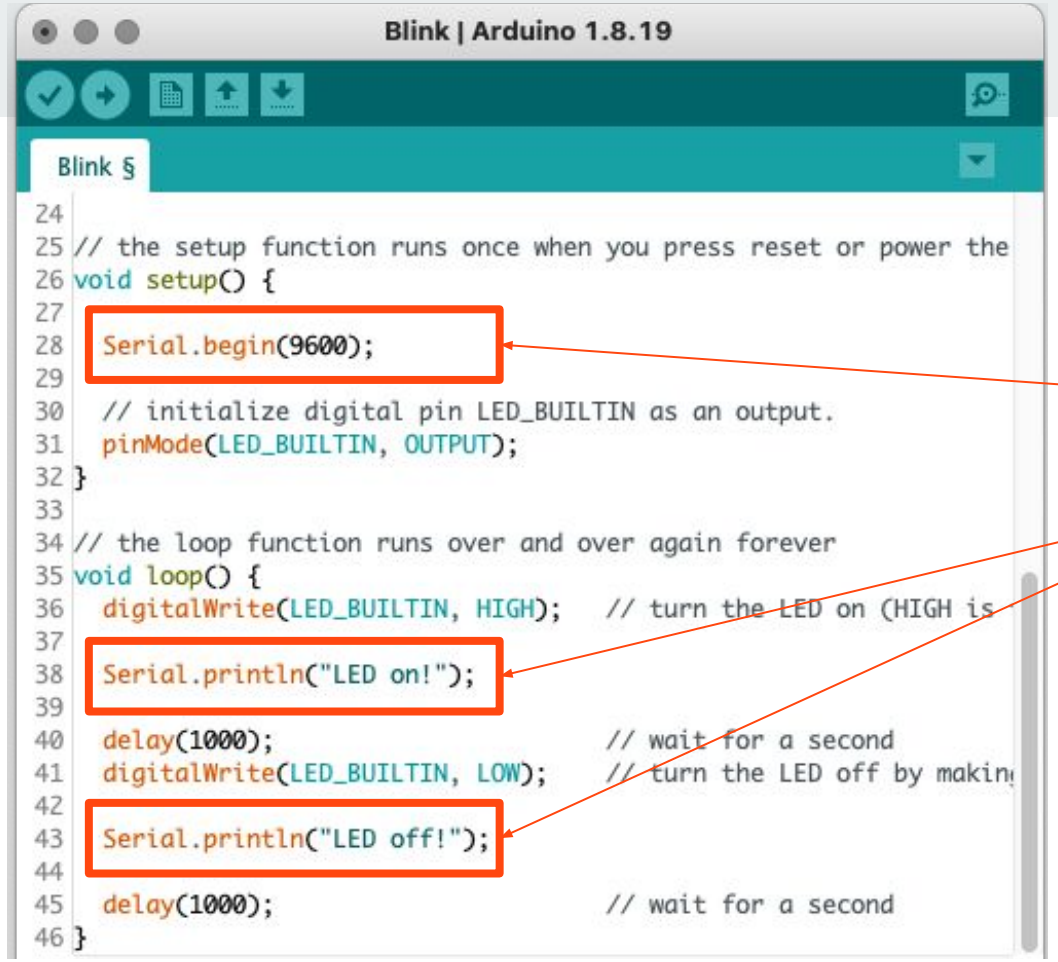
7) Map incoming values from one range to another
map (value, min original, max original, min new, max new);

SOME IMPORTANT FUNCTIONS

SERIAL MONITOR

YOUR FIRST SKETCH

Use the serial monitor to send and receive messages from Arduino. For example, we can use it to read sensor values and debug.




```
Serial.begin(baud rate); in setup
Serial.println(“String”); in loop
Serial.println(variable); in loop
```

SERIAL MONITOR

YOUR FIRST SKETCH

PROGRAMMING WITH ARDUINO IDE / ESP32 S3 (C/C++)

INSTRUCTIONS

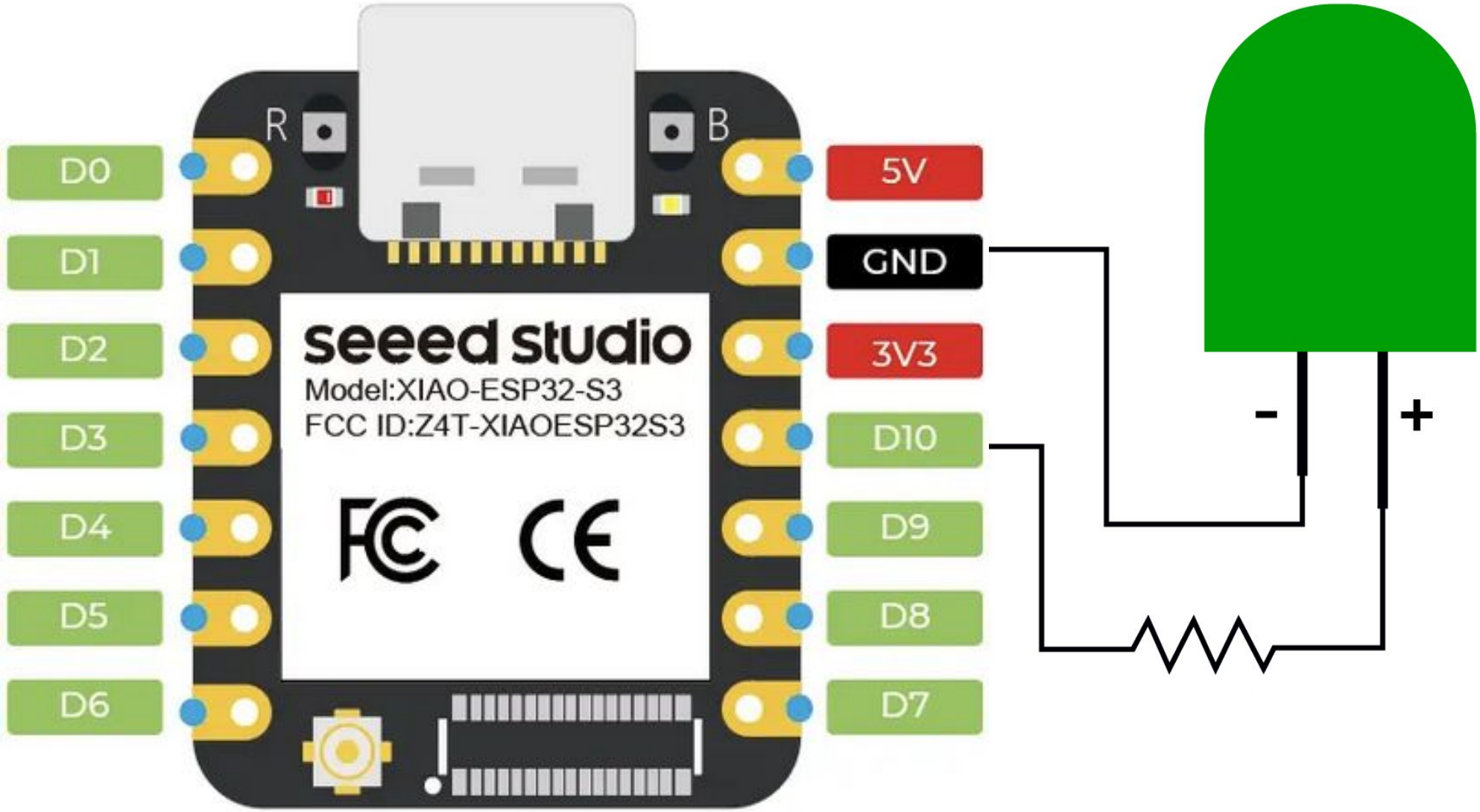
IF sensor is pressed/ stretched
TURN ON LED
ELSE
TURN OFF LED

ELSE..IF

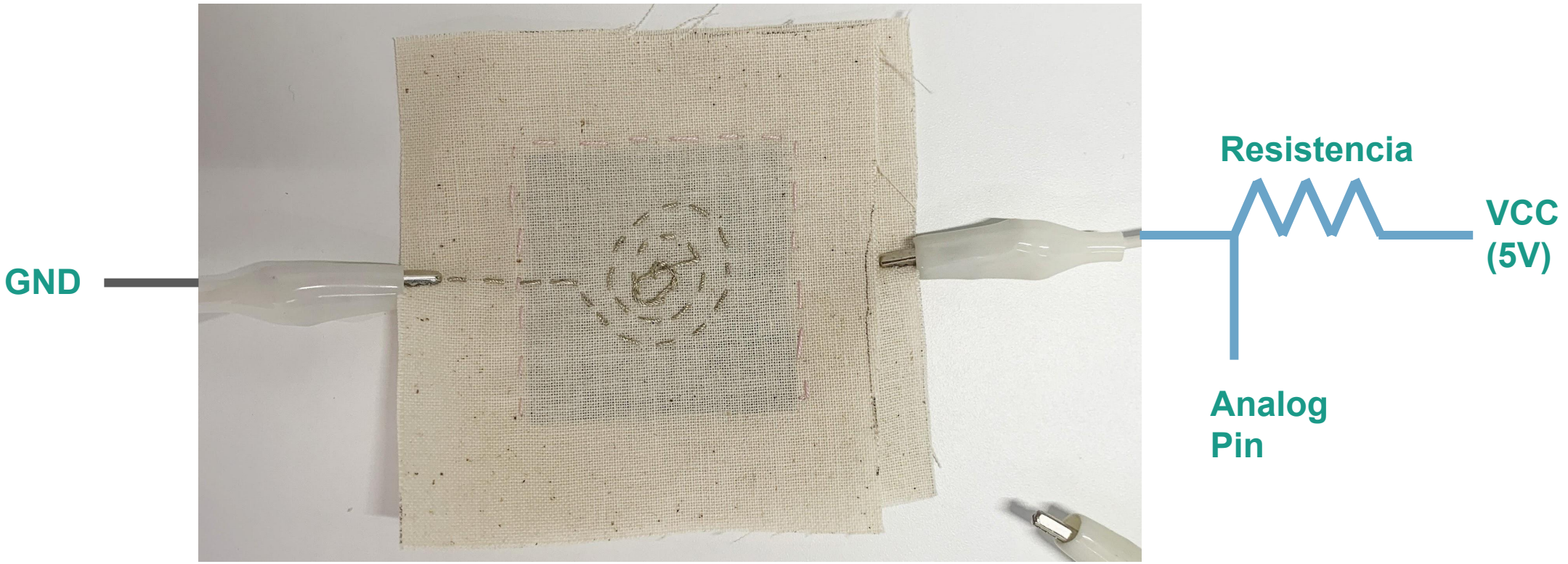
IF (x) is true, then (y) occurs, otherwise (z) occurs.
meaning...

```
if (condition x = true) {

//action A happends

} else {
//action B happens
}
```

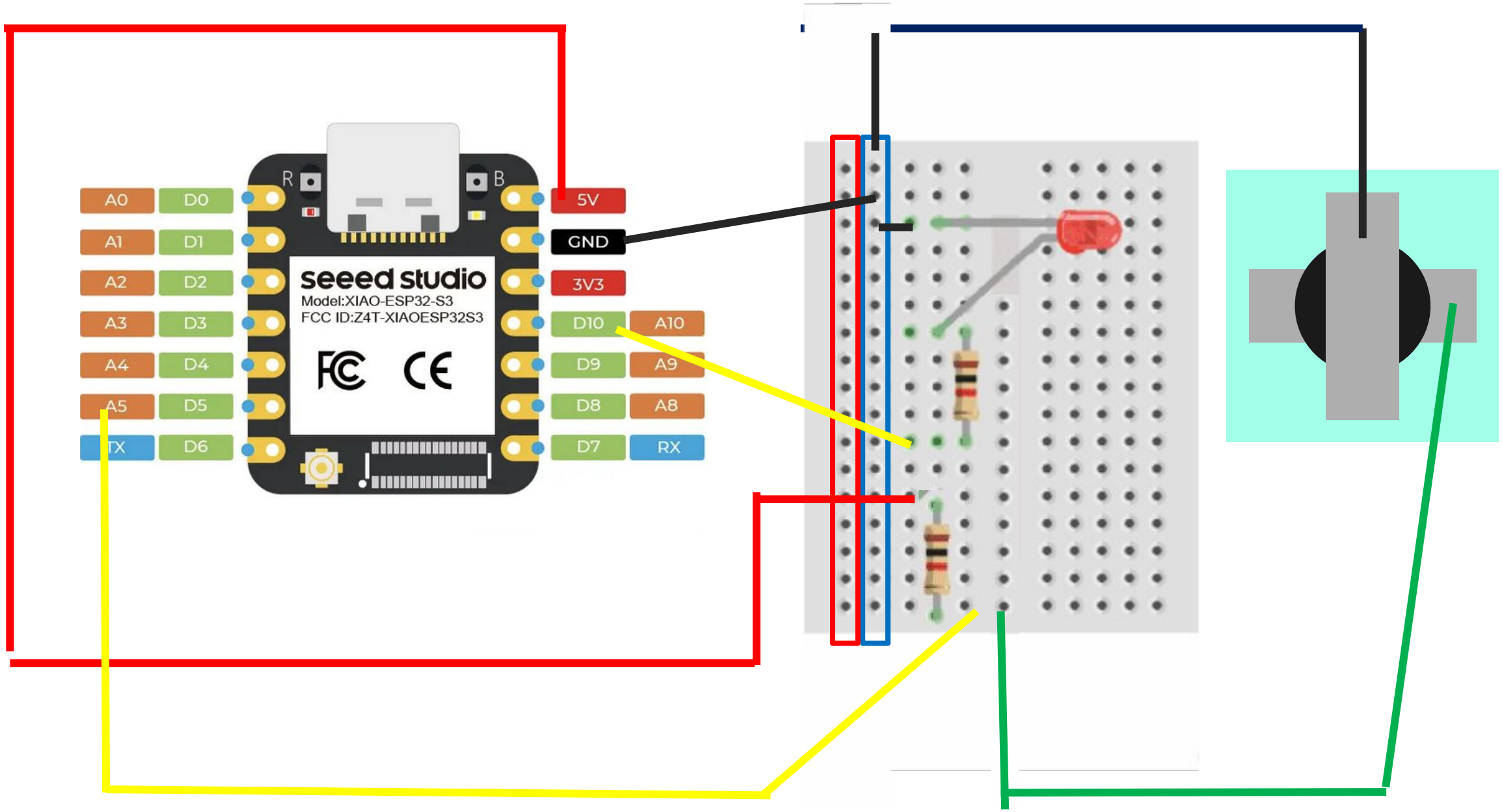
Connecting an LED



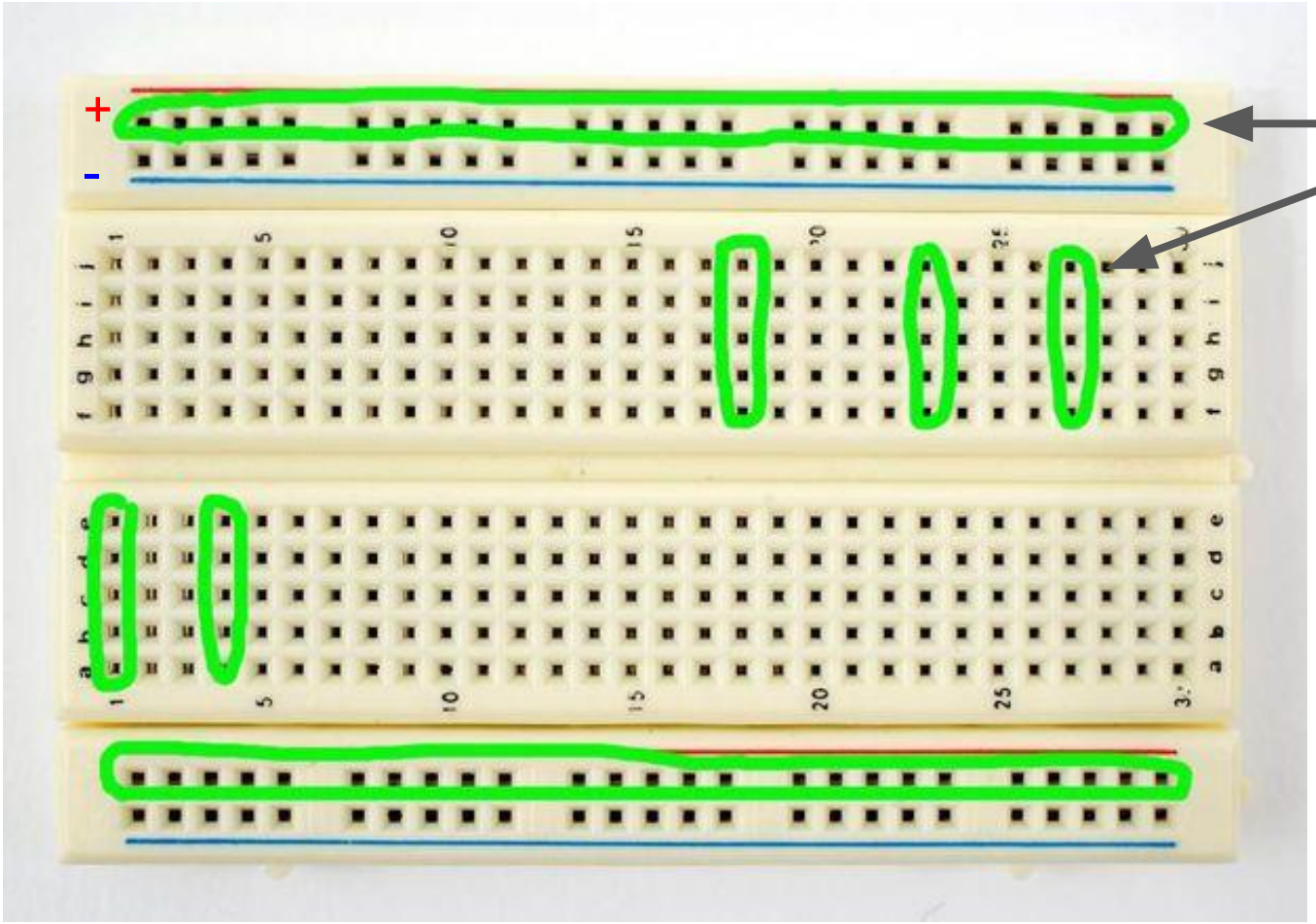
RESISTIVE SENSOR



RESISTIVE SENSOR + LED



Bread board



Let's code

1. Install Arduino IDE + ESP32 board package
2. Choose correct board
3. Choose correct port
4. Find the example code for 'blink'
5. Upload code and test

6. Connect one sensor and LED
7. Copy and write code (if-statement)
8. Upload code and test

IF button is pressed
TURN ON LED
ELSE
TURN OFF LED

Extra: pair up with someone who is ready and set up two sensors. **(same wiring principle as one sensor)**
>What do you need to add to the Arduino code to get two data points?

>Can you add other conditions for your if-statement?

For next week

During next tutorial we will learn how to collect data by WiFi (require Arduino IDE and python environment)

Link: https://realpython.com/installing-python/#reader-comments
Video: https://www.youtube.com/watch?v=QhukcScB9W0

IDEs: PyCharm, Visual Studio Code ……

Tutorial ESP32-S3 II

using Arduino IDE software

About me

Yao Zhang

2nd year phd student in Smart Wearable Lab, focusing on AI about human motion understanding

Supervised by Prof. Xiao Yu

Course assistant for II part of smart wearable course

yao.1.zhang@aalto.fi

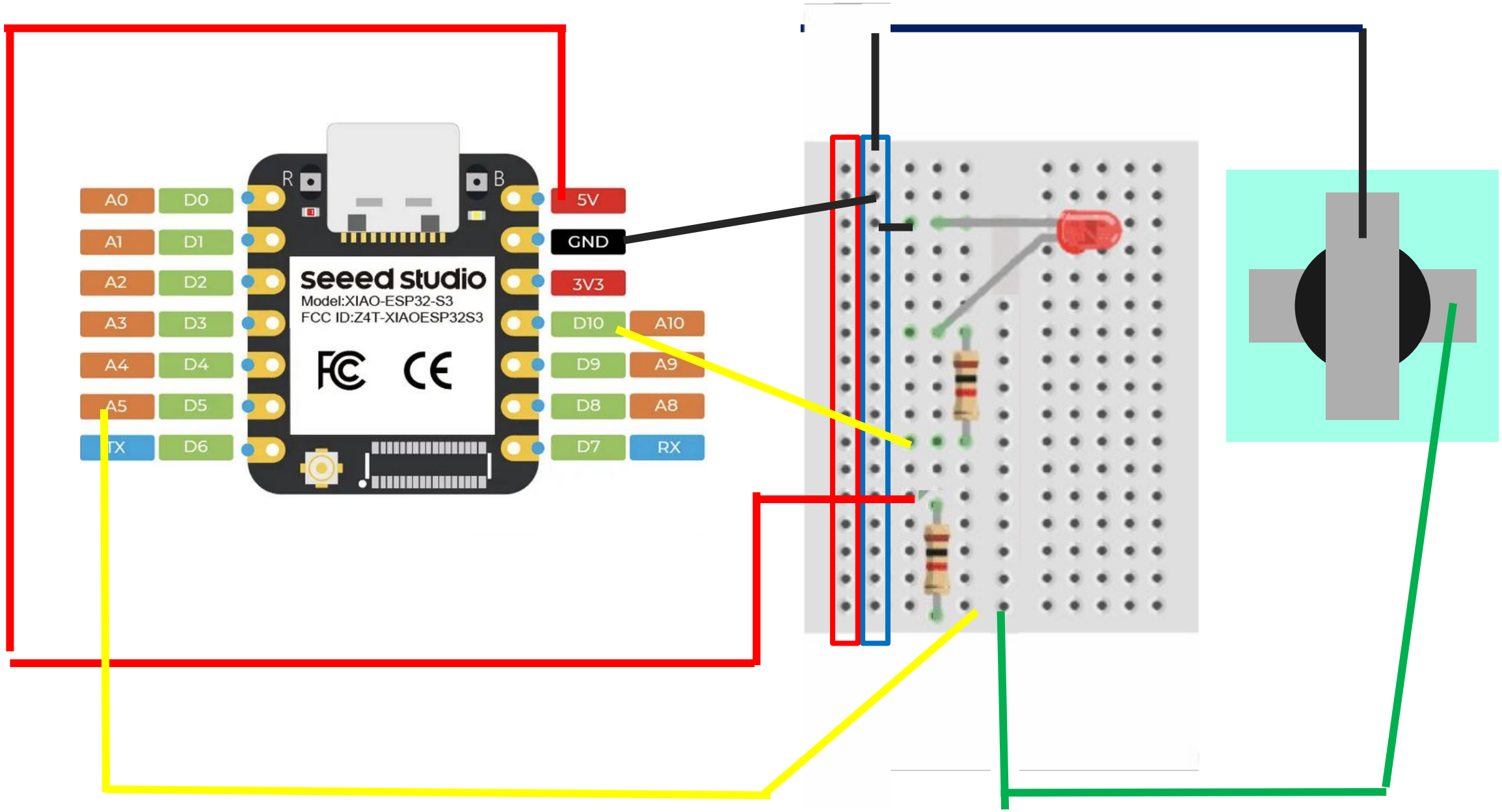
Learning Outcomes

> Introduction to sensor data collection

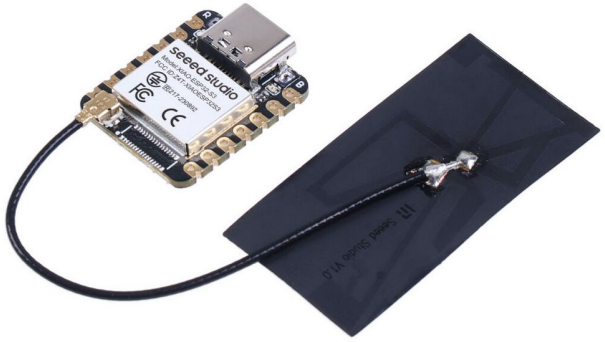
> Basics about python environment

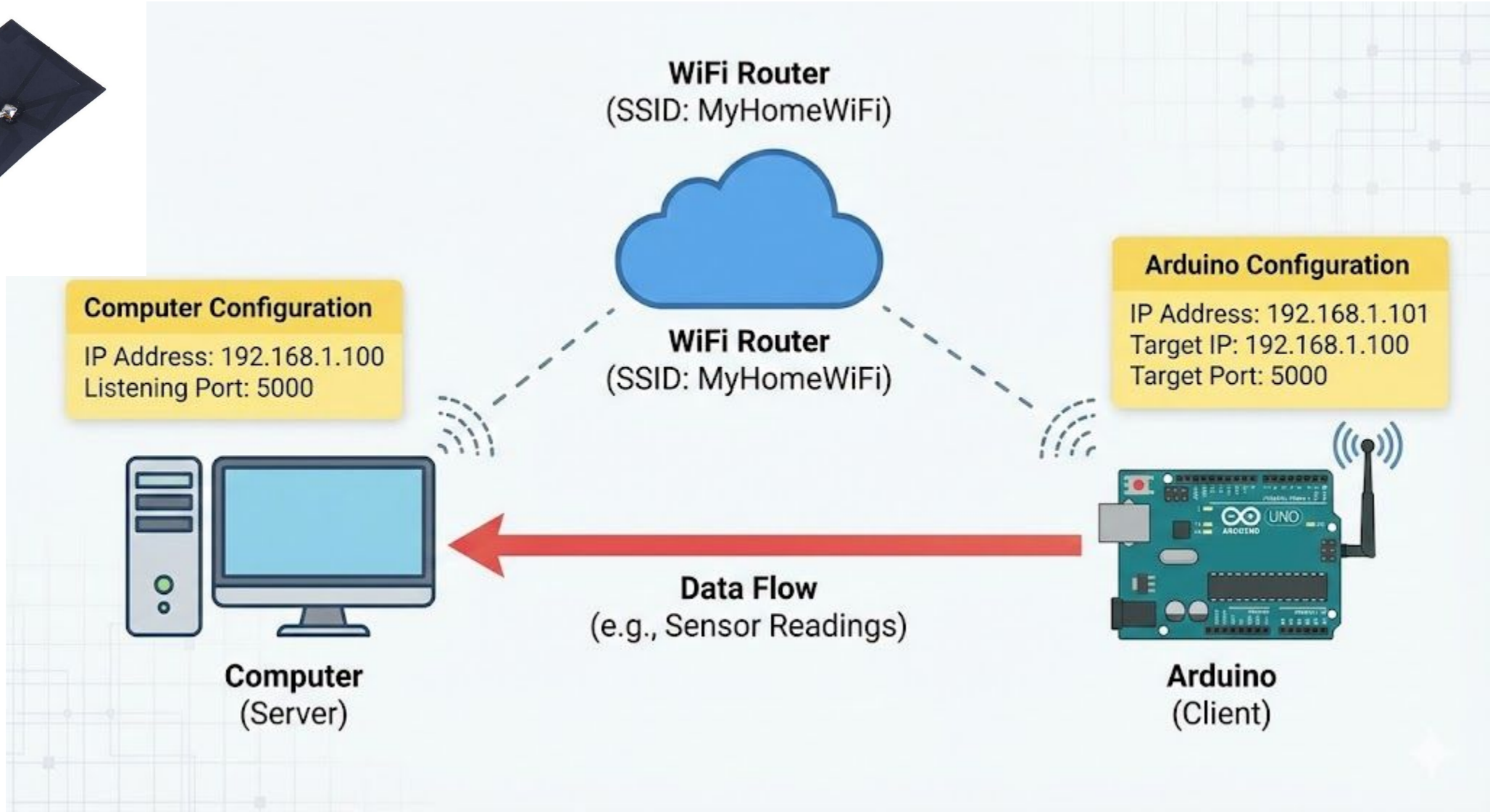
> How to collect data through WiFi

Last week: resistive sensor + LED



WiFi module of ESP32





How to use WiFi module

Important things:

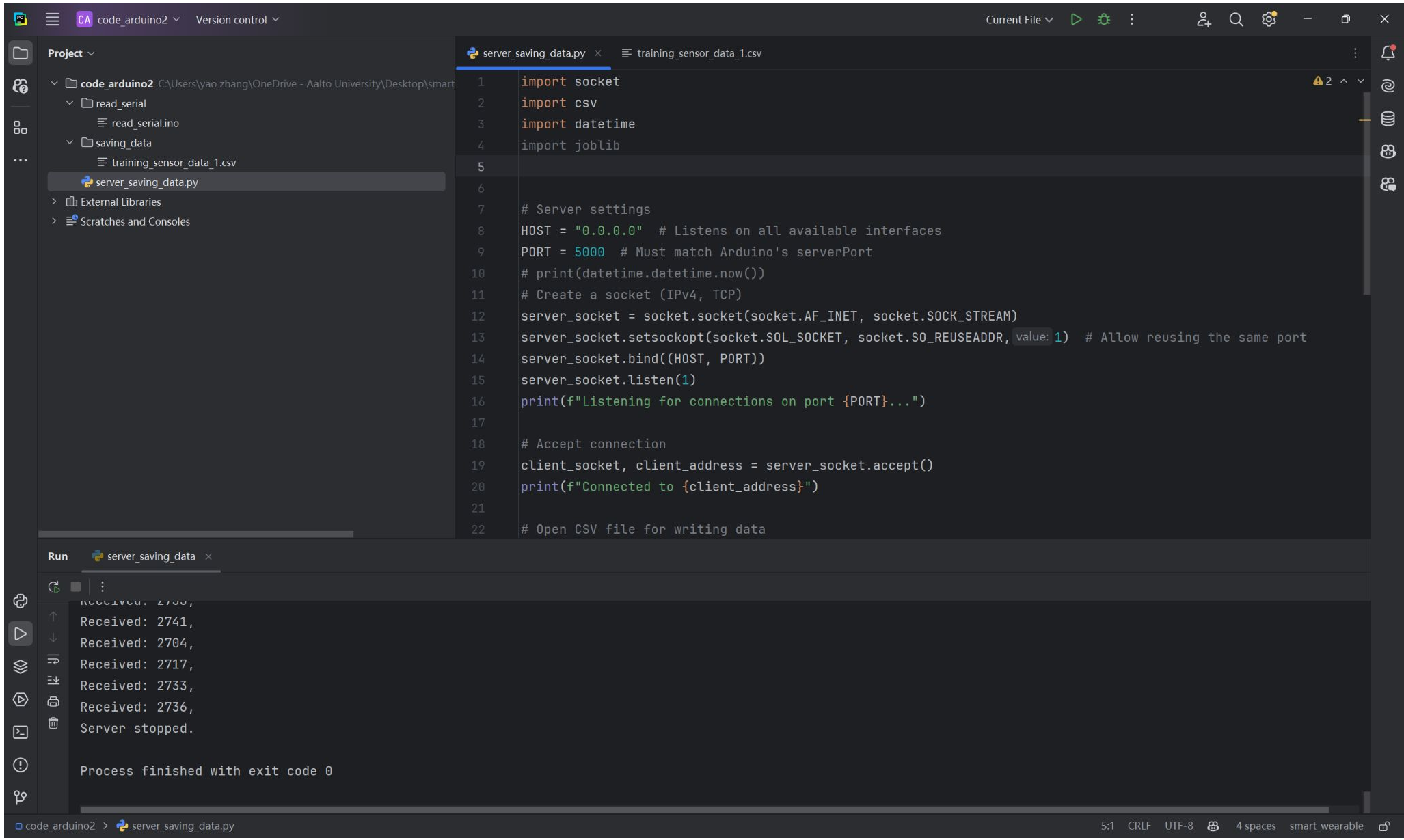
1. connect Arduino(sender) and your computer(server) to same WiFi(like: your phone hotspot).
2. changing your Arduino code with your own WiFi id and password.
3. get server IP after you connect your computer to the same WiFi.
4. keep Port number as same for both sender and server part.

```
// WiFi credentials
const char* ssid = "zyyyyy";     // Replace with your WiFi SSID
const char* password = "zy561526"; // Replace with your WiFi password

// Server details (your computer's IP and port)
const char* serverIP = "172.20.10.14";  // Replace with your computer's local IP
const int serverPort = 5000;  // Must match the Python script
```

Server (our computer)

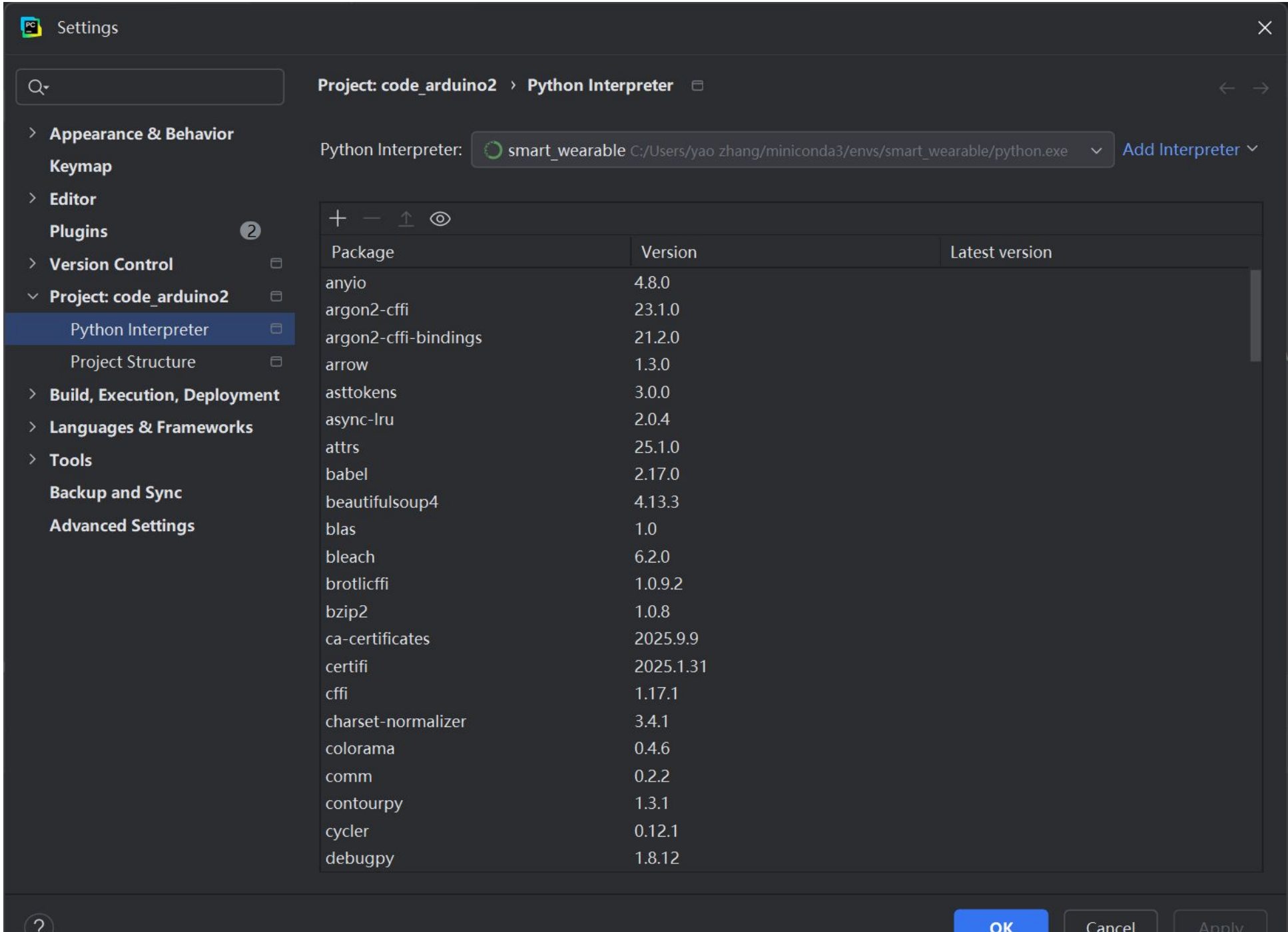
IDEs:

PyCharm
VS code
...



PyCharm

Python Interpreter(environment)

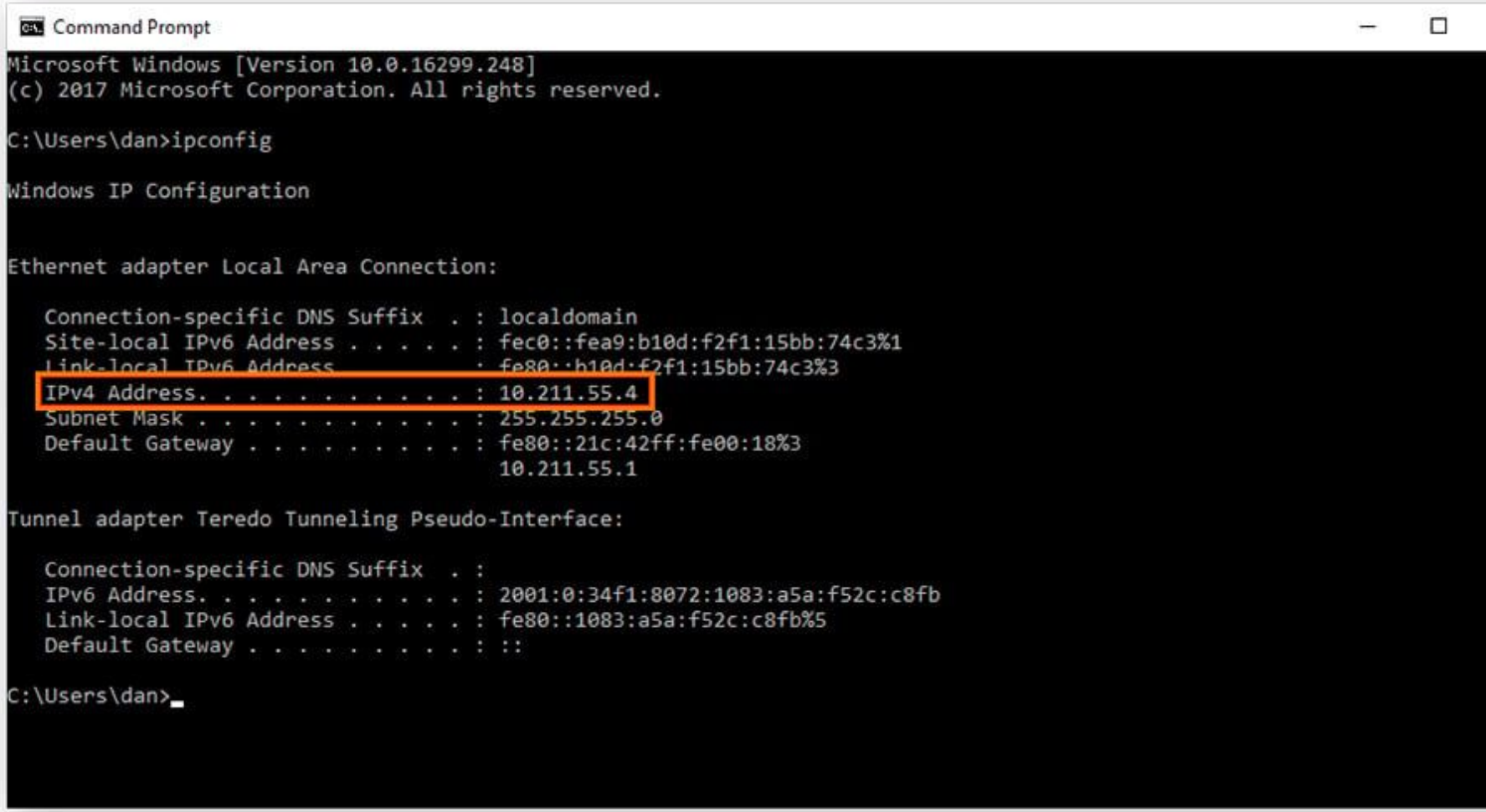


How to get IP

https://www.avast.com/c-how-to-find-ip-address

For Windows: ipconfig

For Mac: ifconfig | grep "inet"



Collecting data

```
#include <WiFi.h>

// WiFi credentials
const char* ssid = "zyyyyy";     // Replace with your WiFi SSID
const char* password = "zy561526"; // Replace with your WiFi password

// Server details (your computer's IP and port)
const char* serverIP = "172.20.10.6";  // Replace with your computer's local IP the
const int serverPort = 5000;  // Must match the server Python script

int sensorPin1 = A5;
int sensorValue1 = 0;
//const int buttonPin = 2;

WiFiClient client;

void setup() {
    Serial.begin(115200);
    // comment when no serial
    // while (!Serial);

    // 1. start connection
    Serial.print("Connecting to WiFi...");
    WiFi.begin(ssid, password);

    // 2. check status
    while (WiFi.status() != WL_CONNECTED) {
```

```
Listening for connections on po
Connected to ('172.20.10.2', 50
Received: 15
Received: 16
Received: 15
Received: 15
Received: 16
Received: 16
Received: 15
Received: 15
Received: 14
Received: 16
Received: 16
Received: 14
Received: 16
```

```
# Server settings
HOST = "0.0.0.0"  # Listens on all available interfaces
PORT = 5000  # Must match Arduino's serverPort
# print(datetime.datetime.now())
# Create a socket (IPv4, TCP)
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, value: 1)  # Allow
server_socket.bind((HOST, PORT))
server_socket.listen(1)
print(f"Listening for connections on port {PORT}...")

# Accept connection
client_socket, client_address = server_socket.accept()
print(f"Connected to {client_address}")

# Open CSV file for writing data
csv_filename = "./saving_data/training_sensor_data_1.csv"
with open(csv_filename, mode="w", newline="") as file:
    writer = csv.writer(file)
    writer.writerow(["Timestamp", "Sensor Value", "label"])  # Write header

    try:
```