# Working with ESP32

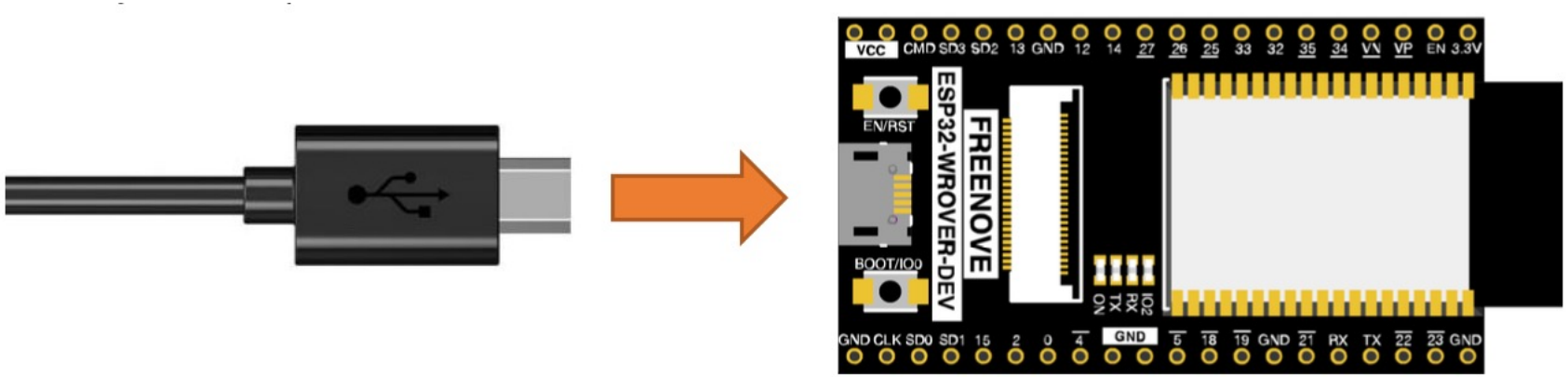
Haroon Lone

Feb 8, 2023

# Step1: Install Thonny

- Use following link to download Thonny
  - https://thonny.org/

# Step2: Connect ESP32 to computer with the USB cable

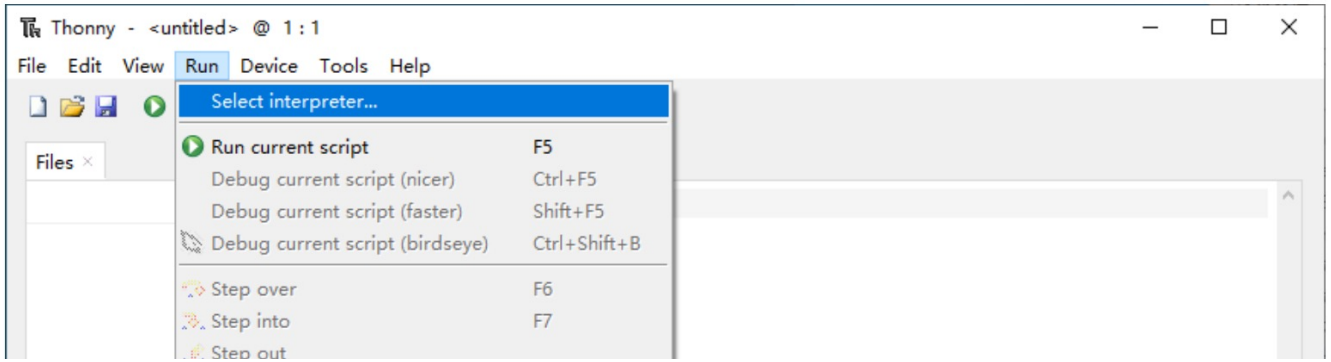


# Step 2.1 Install Driver (optional)

- If your computer doesn't recognize the board, install required driver by going through the following link
  - https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/establish-serial-connection.html#:~:text=Connect%20ESP32%20to%20PC,in%20internet%20and%20install%20them.

# Step 3: Install MicroPython on ESP32

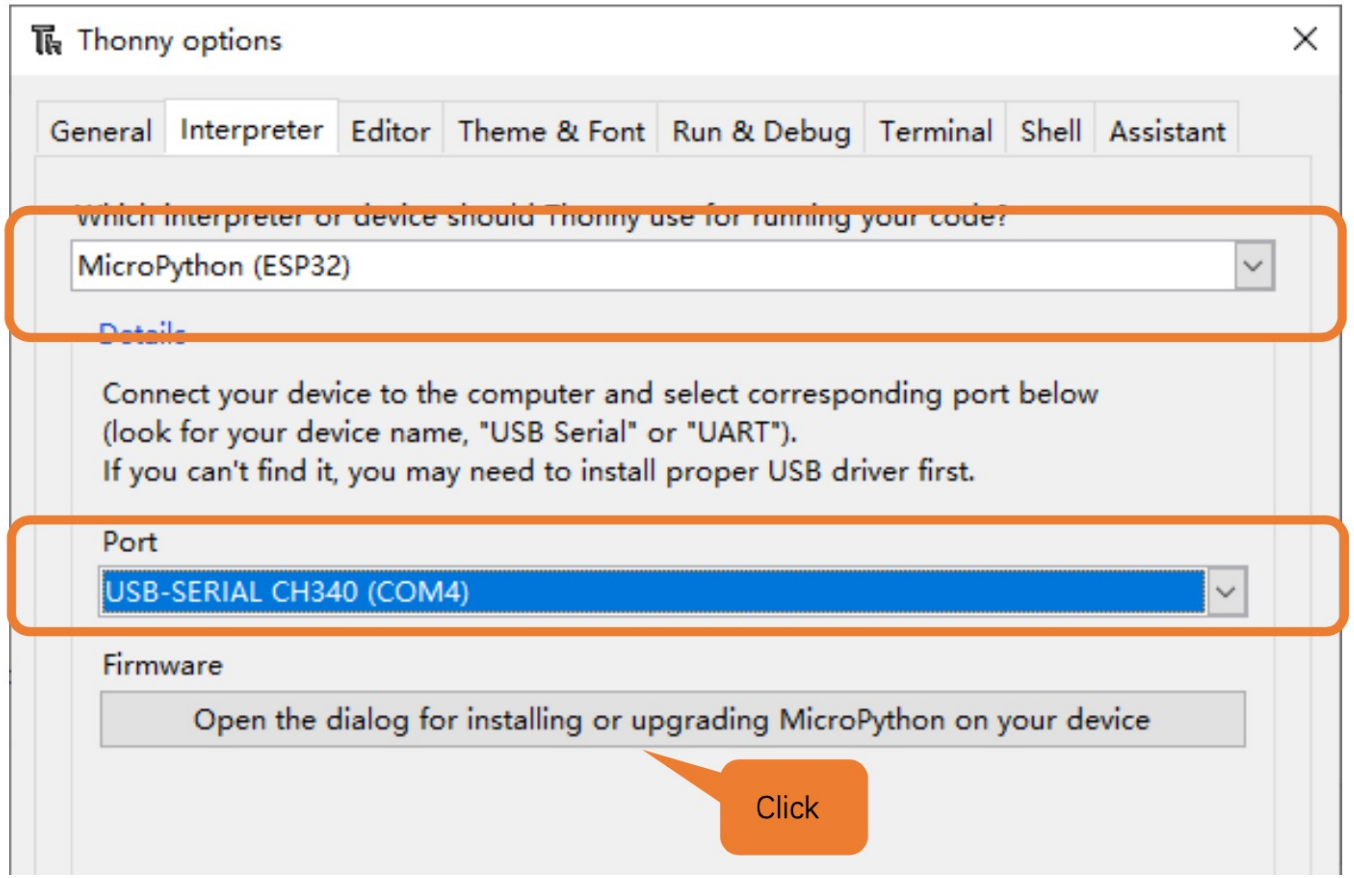
1. Connect ESP32 to computer via USB
2. Open Thonny and select “Select interpreter”




3. Browse for *xyz.bin file [Micro-python] in the shared folder*

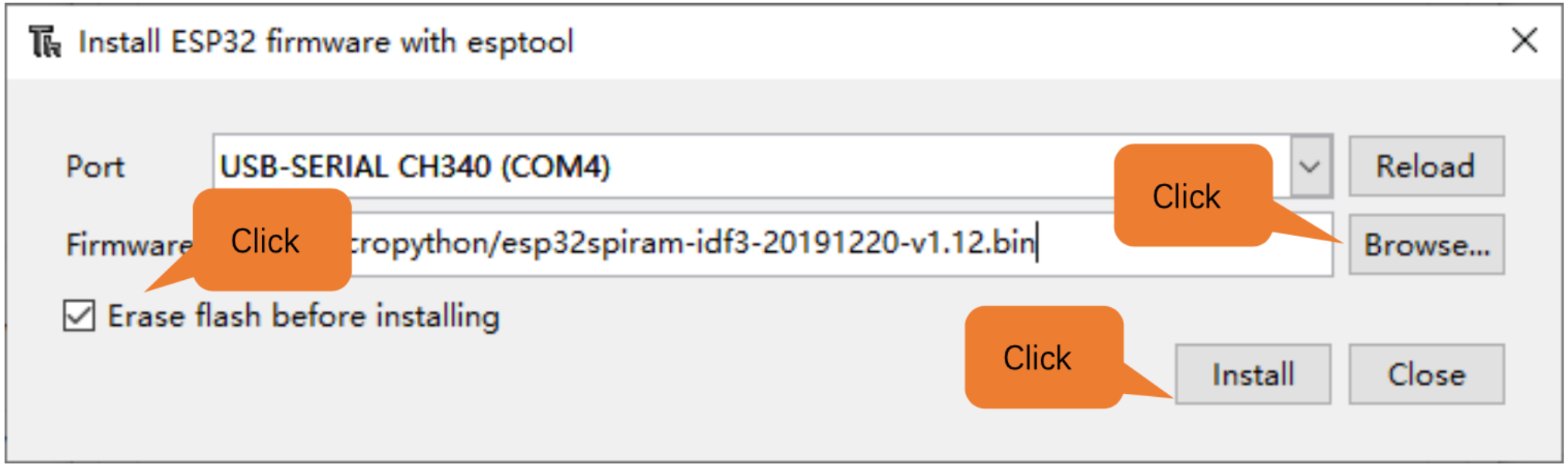
# Step 3.1: Install MicroPython on ESP32

1. Select options as shown in the following window

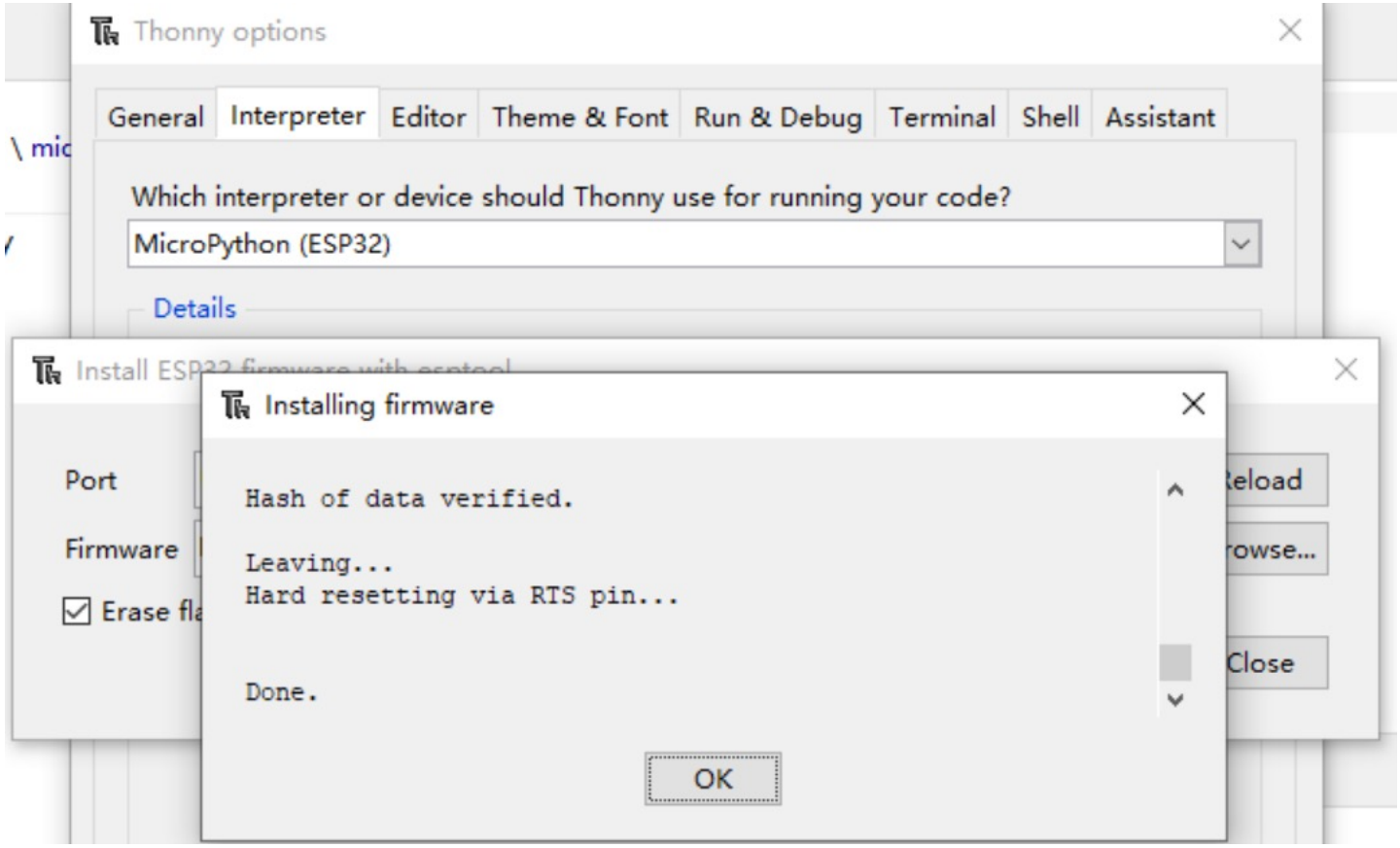


# Step 3.1: Install MicroPython on ESP32

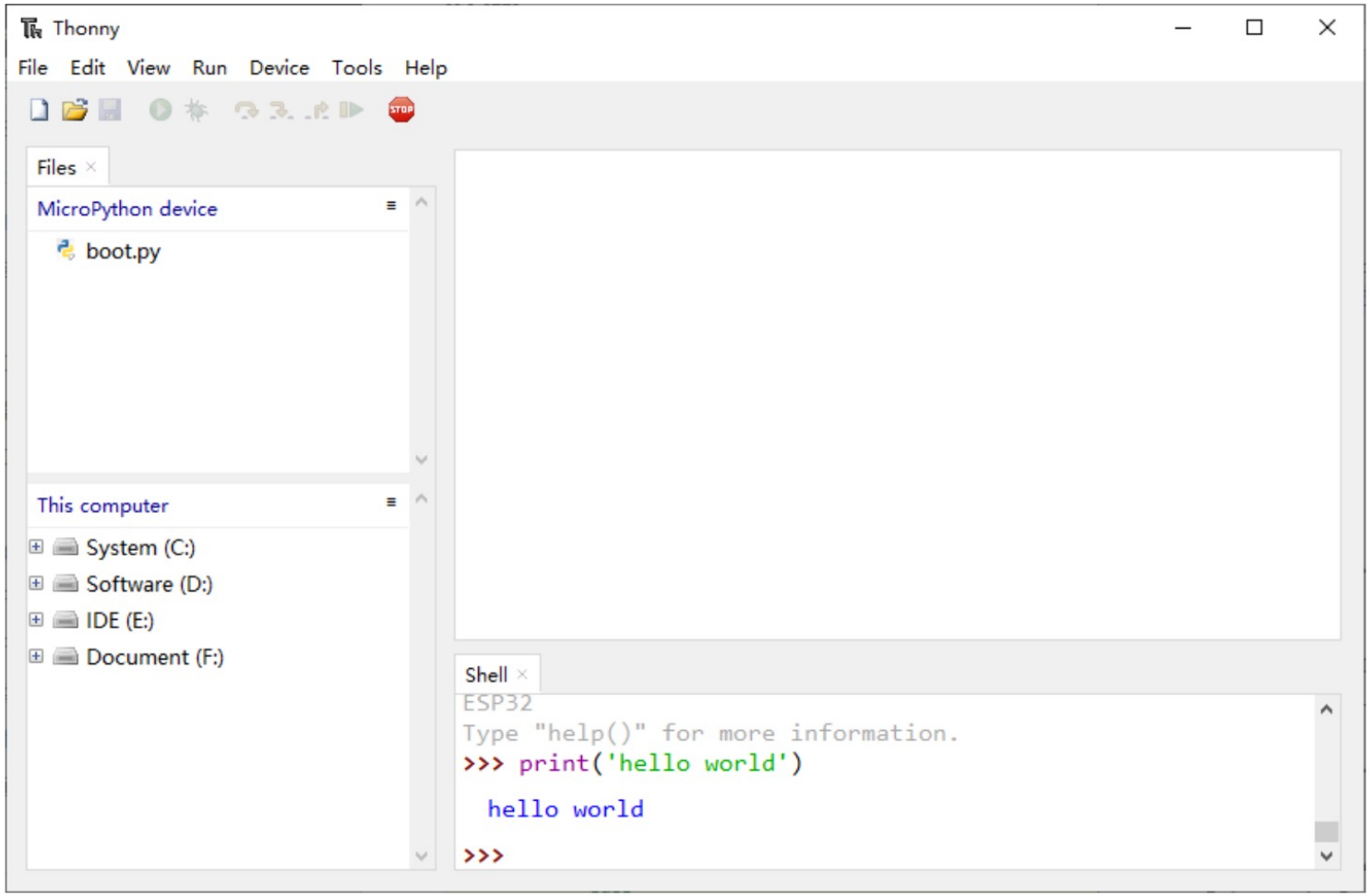
1. Select options as shown in the following window. The file can be found at *Lab-data/Python_Firmware/…*



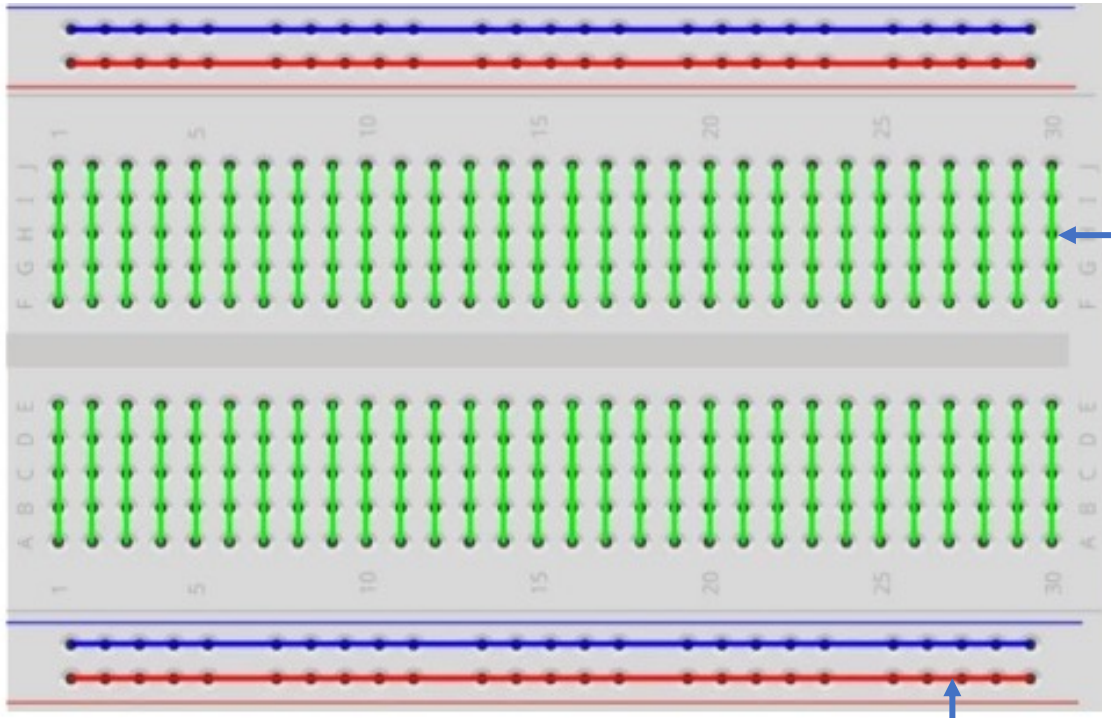
# Step 3: Confirmation



# Step 4: Print ``Hello world” on ESP32 shell



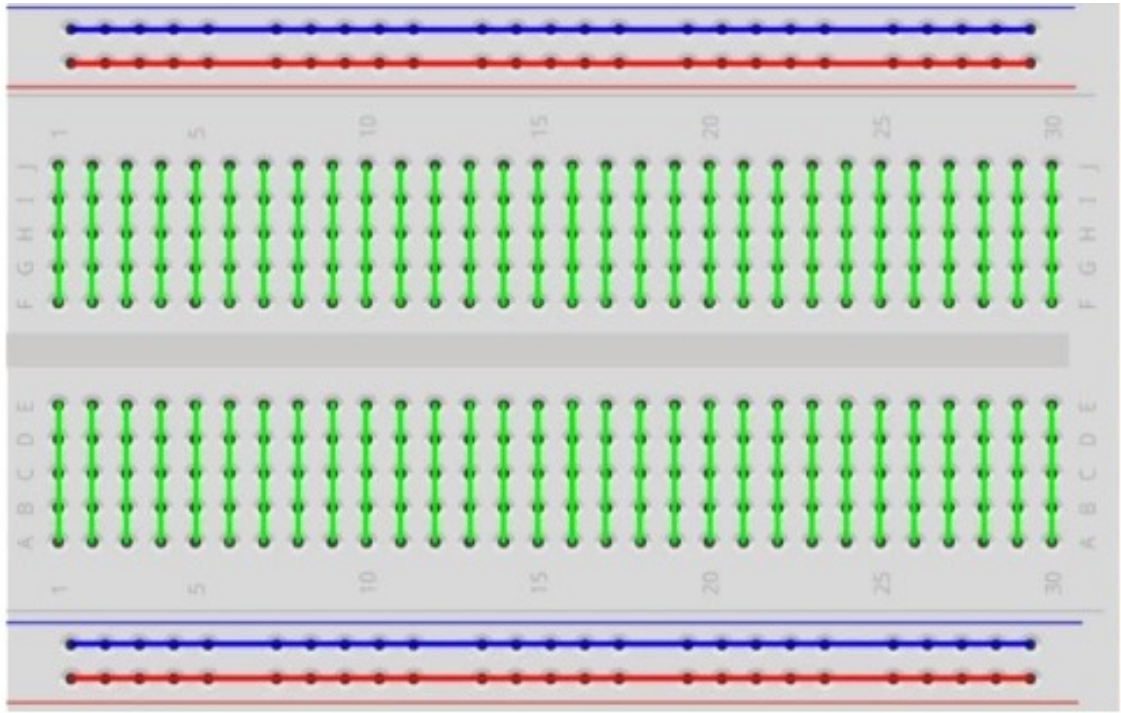
# Breadboard Internal connections

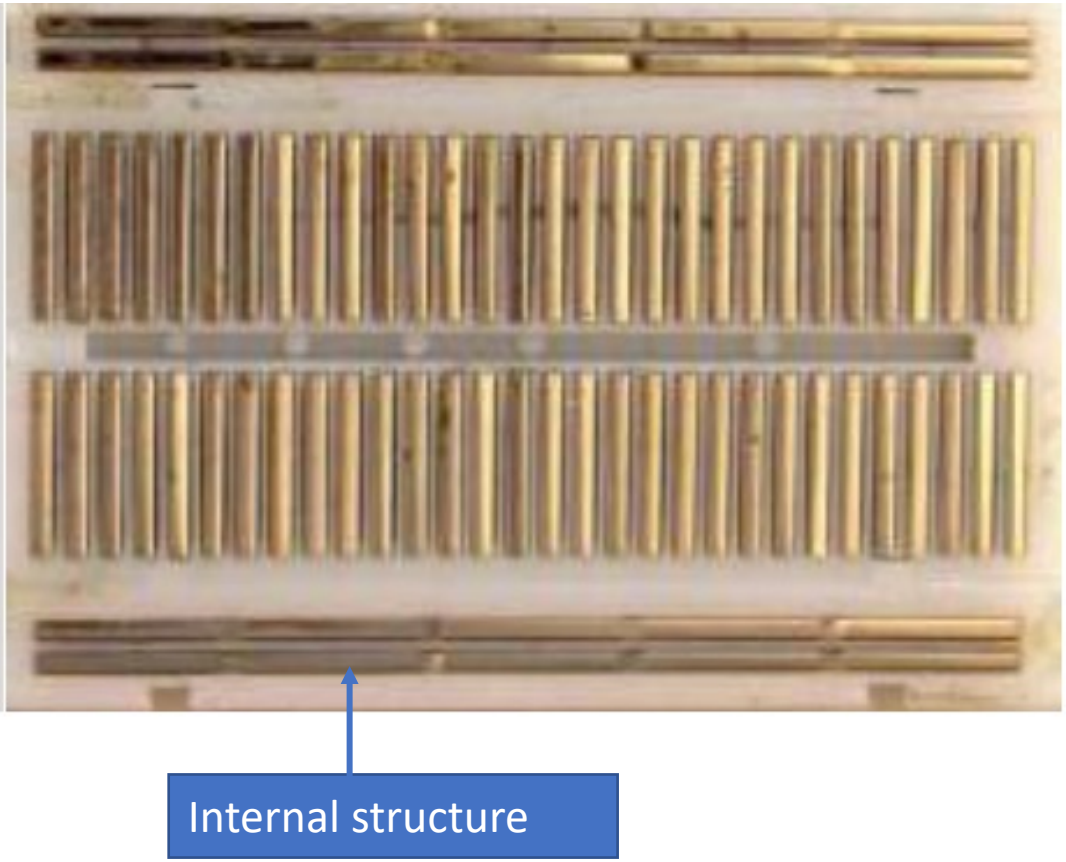


Vertical rows (green colored) connected internally

Horizontal rows (red/blue colored) connected internally

# Breadboard





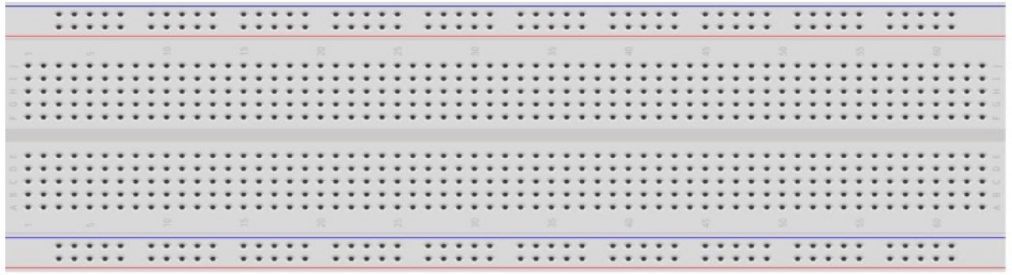
# Blinking LED

# Components required

- ESP32
- Breadboard
- Led (1)
- Resistor 220 ohm (1)
- Jumper wires (2)

# Components required

Breadboard x1



LED x1

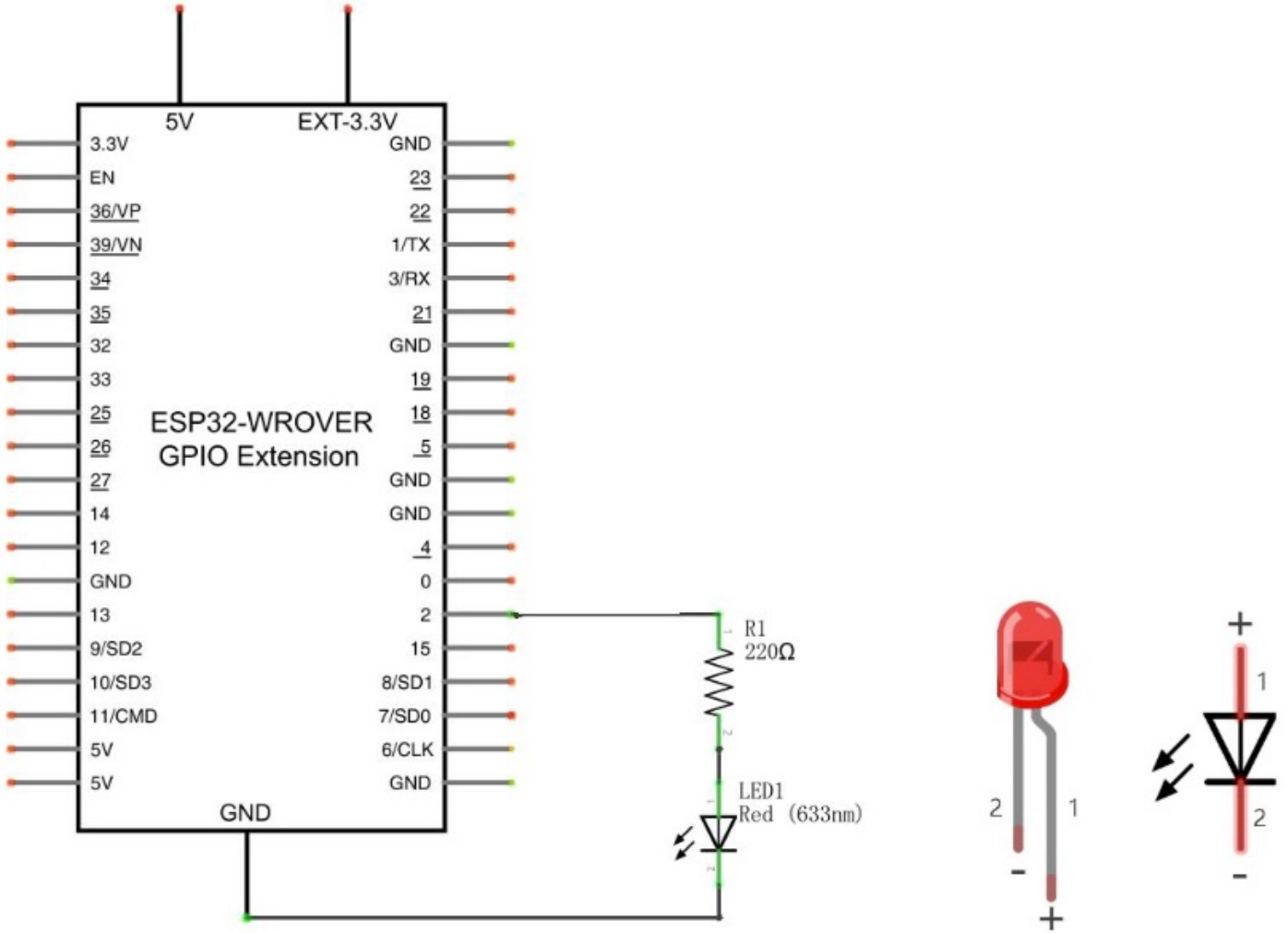
Resistor 220Ω x1

Jumper M/M x2

# LED

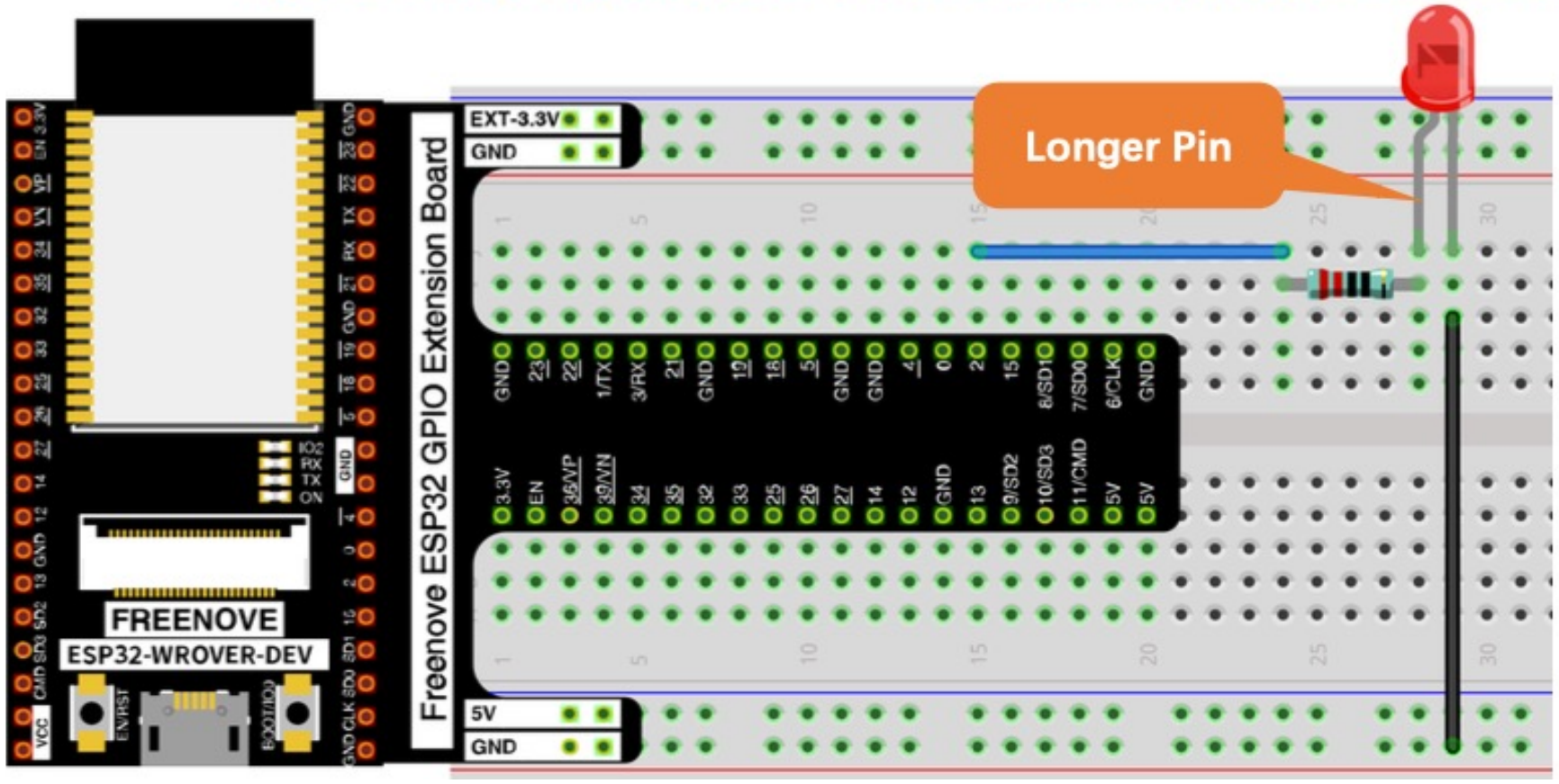
- It is a type of diode
- Longer leg (+ve), shorter leg (-ve).
- Negative leg, often called as GND (ground)
- LED’s can’t be connected to a power supply directly
- Often resistor is connected in series with a LEd

# Schematic


5V
EXT-3.3V
3.3V
EN
36/VP
39/VN
34
35
32
33
25
26
27
14
12
GND
13
9/SD2
10/SD3
11/CMD
5V
5V
GND
GND
23
22
1/TX
3/RX
21
GND
19
18
5
GND
GND
4
0
2
15
8/SD1
7/SD0
6/CLK
GND
ESP32-WROVER
GPIO Extension
R1
220Ω
LED1
Red (633nm)


2
1
-
+
+
1
2
-

# Hardware layout



# Program

```
from time import sleep_ms
from machine import Pin

led=Pin(2,Pin.OUT) #create LED object from pin2,Set Pin2 to outpu
try:
    while True:
        led.value(1)          #Set led turn on
        sleep_ms(1000)
        led.value(0)          #Set led turn off
        sleep_ms(1000)
except:
    pass
```

# How to blink LED?

1. Connect components as shown in Hardware layout slide. Make sure connections are correct
2. Open Thonny on your computer
3. Connect ESP32 to computer via USB
4. Upload boot.py first and then your program to ESP32 from Thonny
5. Run the program file. Now LED should blink

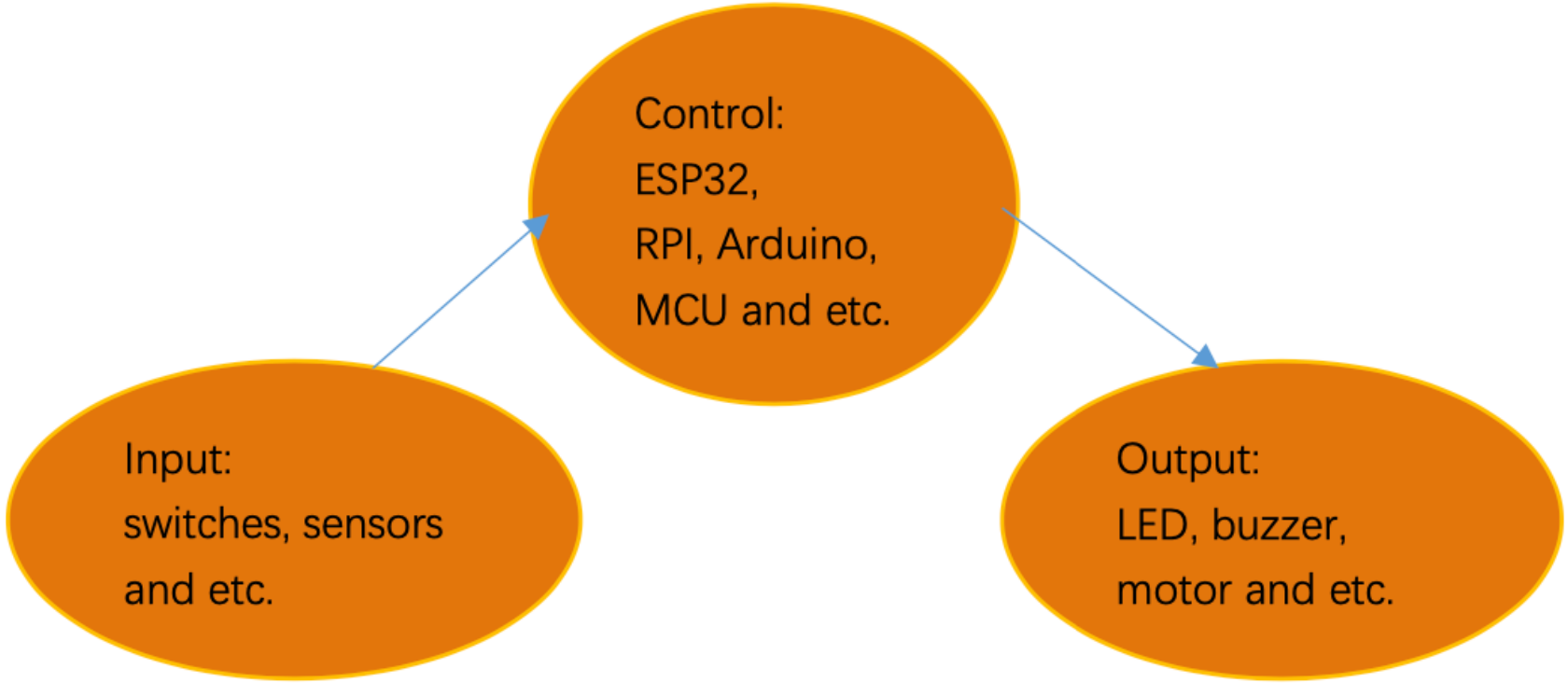
# General Instructions

- Use **ctrl + C (**or ctrl + ]**)** to gain control of ESP32. Sometimes it gets busy
- Run the program after uploading to ESP32
- Run program by pressing reset button on ESP32

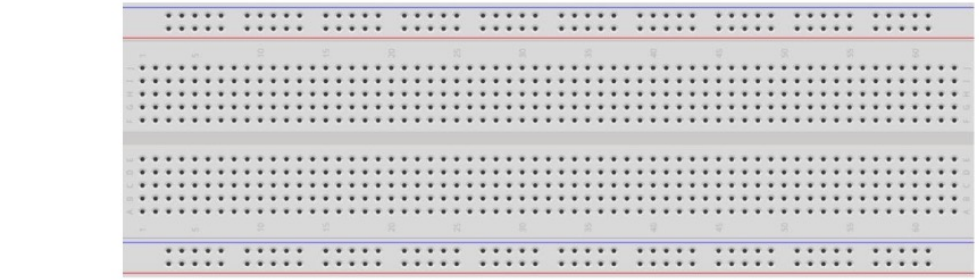
Don’t put any other component in the board other than the mentioned components.

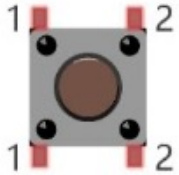
# Controlling LED with a button

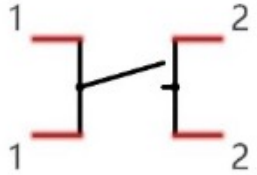
# Input and Output



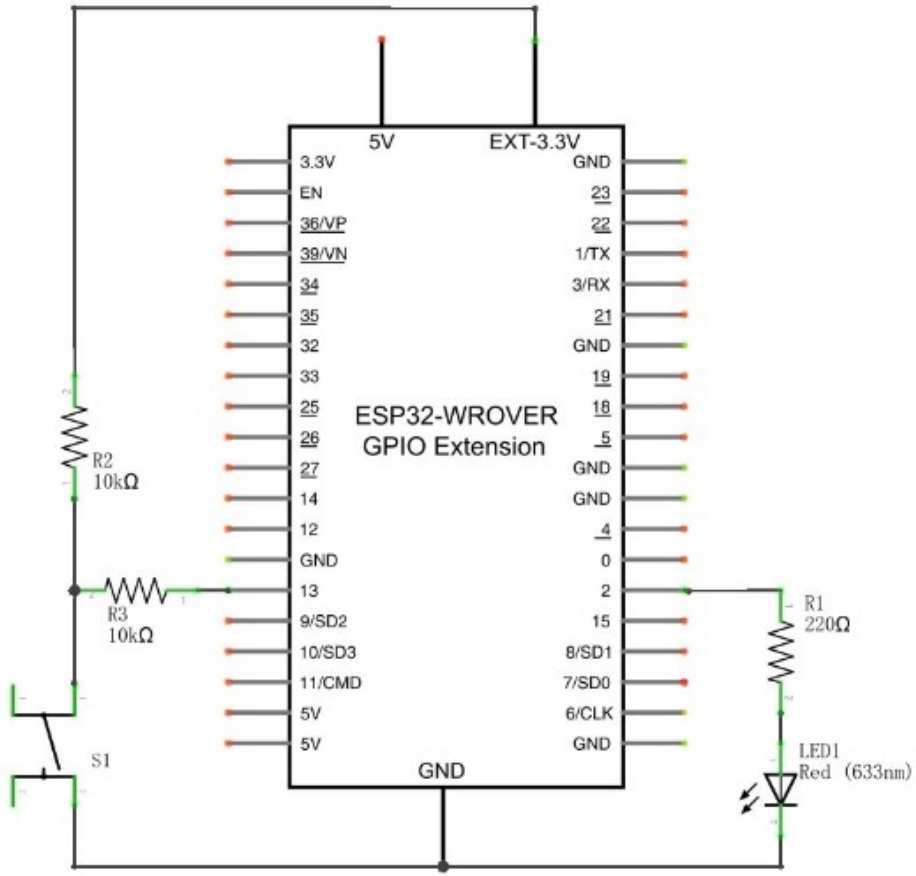
# Components

| | | | | |
|---|---|---|---|---|
| Breadboard x1 | | | | |
|  | | | | |
| Jumper M/M x4 | LED x1 | Resistor 220Ω x1 | Resistor 10kΩ x2 | Push button x1 |
|  |  |  |  |  |

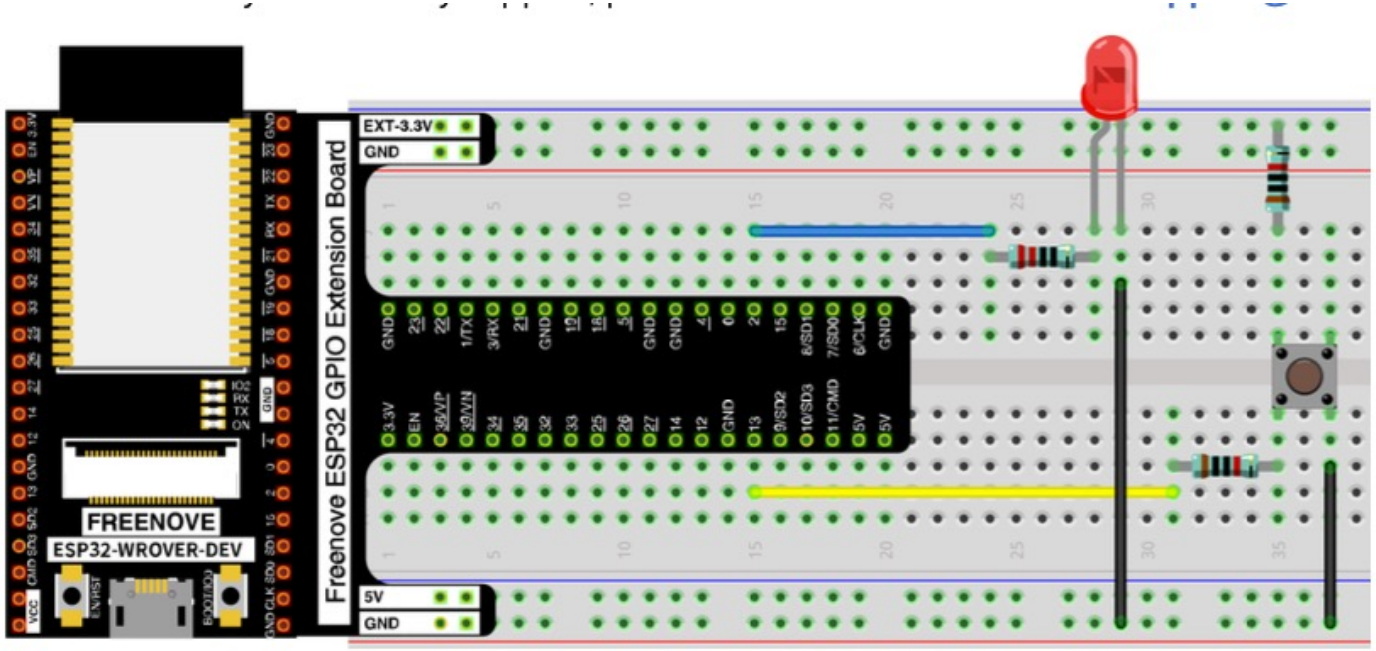


# Schematic

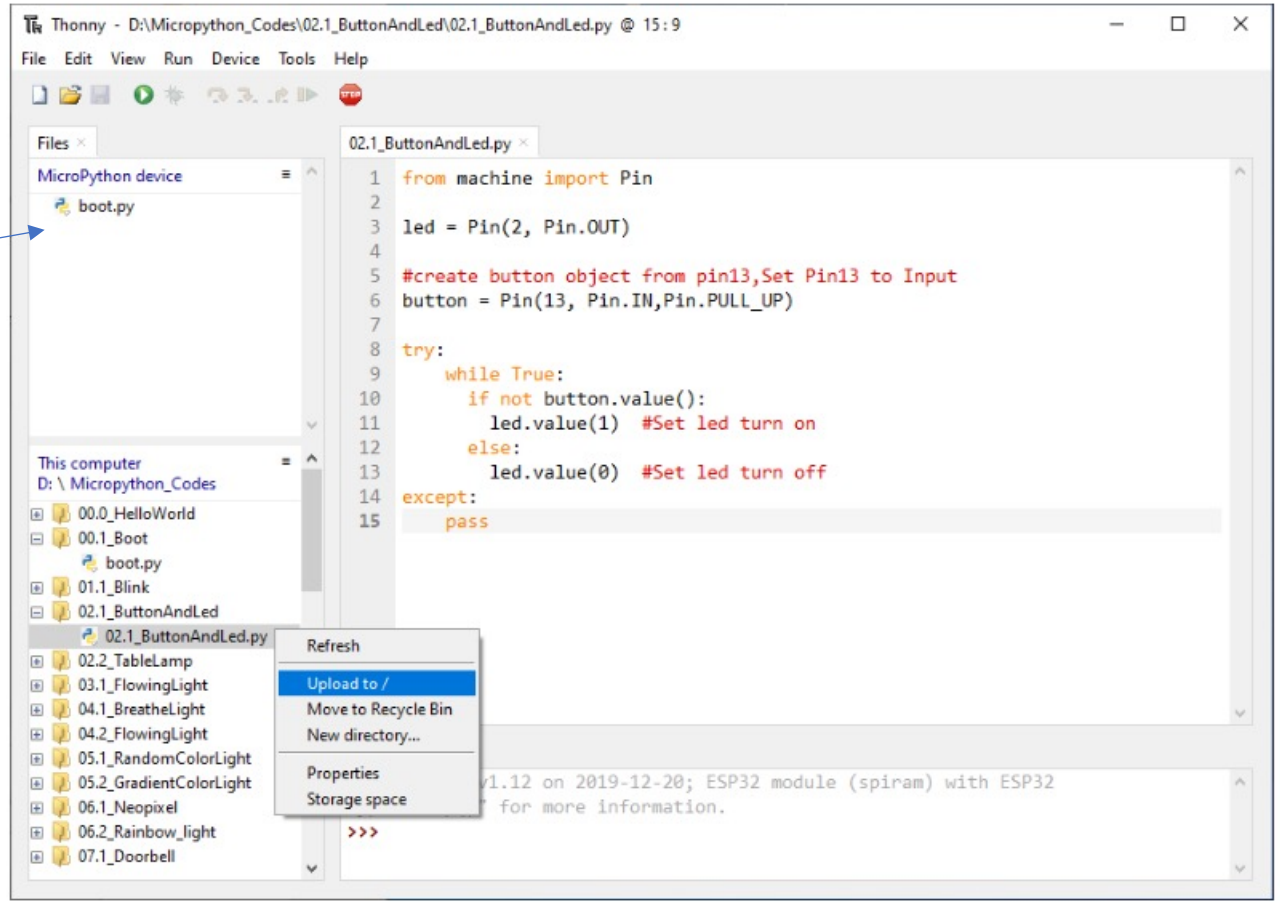

5V
EXT-3.3V
3.3V
EN
36/VP
39/VN
34
35
32
33
25
26
27
14
12
GND
13
9/SD2
10/SD3
11/CMD
5V
5V
GND
23
22
1/TX
3/RX
21
GND
19
18
5
GND
GND
4
0
2
15
8/SD1
7/SD0
6/CLK
GND
ESP32-WROVER
GPIO Extension
R2
10kΩ
R3
10kΩ
S1
R1
220Ω
LED1
Red (633nm)

# Connections



# Code

Both files should show here




Make sure that both boot.py and program file are on ESP32

# How to run your code?

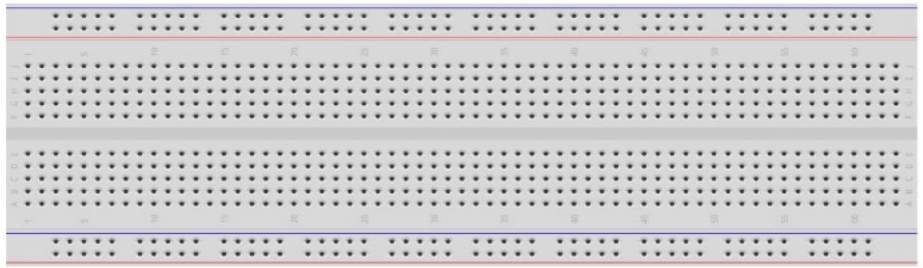
- Connect components to ESP32 (Double check the order)
- Powerup ESP32 by connecting to laptop
- Open Thonny and upload *boot.py and program file* to ESP32. Delete remaining files on the ESP32
- Reset ESP32 (by clicking a small button near the USB input)
- Now your project should work. Test by pressing and releasing the button on the breadboard

# Infrared Motion Detection

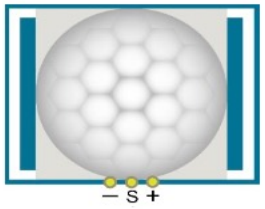
Use heat waves (IR) generated by a body to turn on the LED.

# Components

Breadboard x1



HC SR501 x1



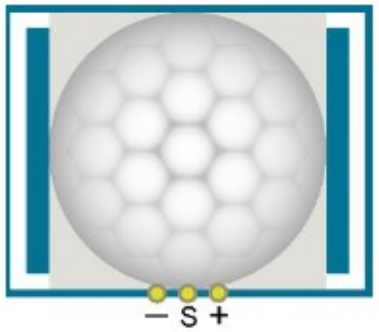
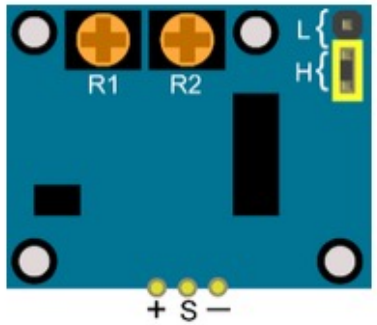
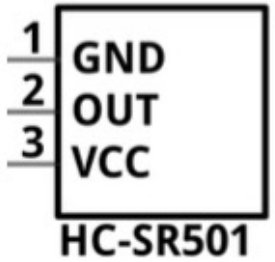

LED x1

Resistor 220Ω x1

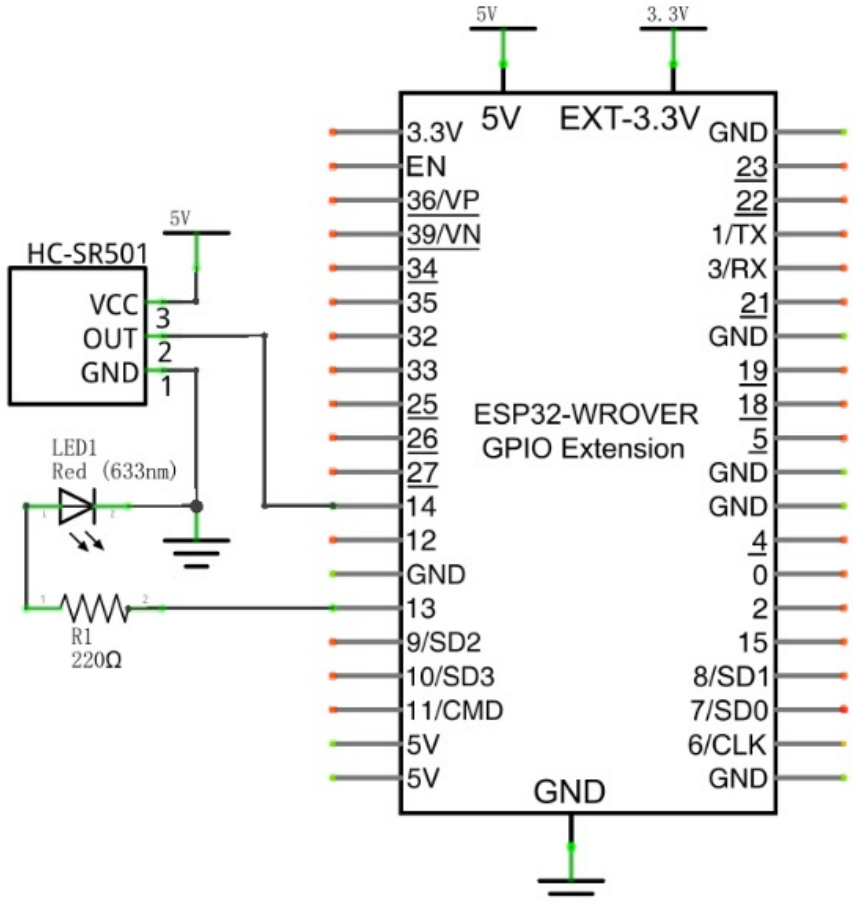
Jumper F/M x3

Jumper M/M x2

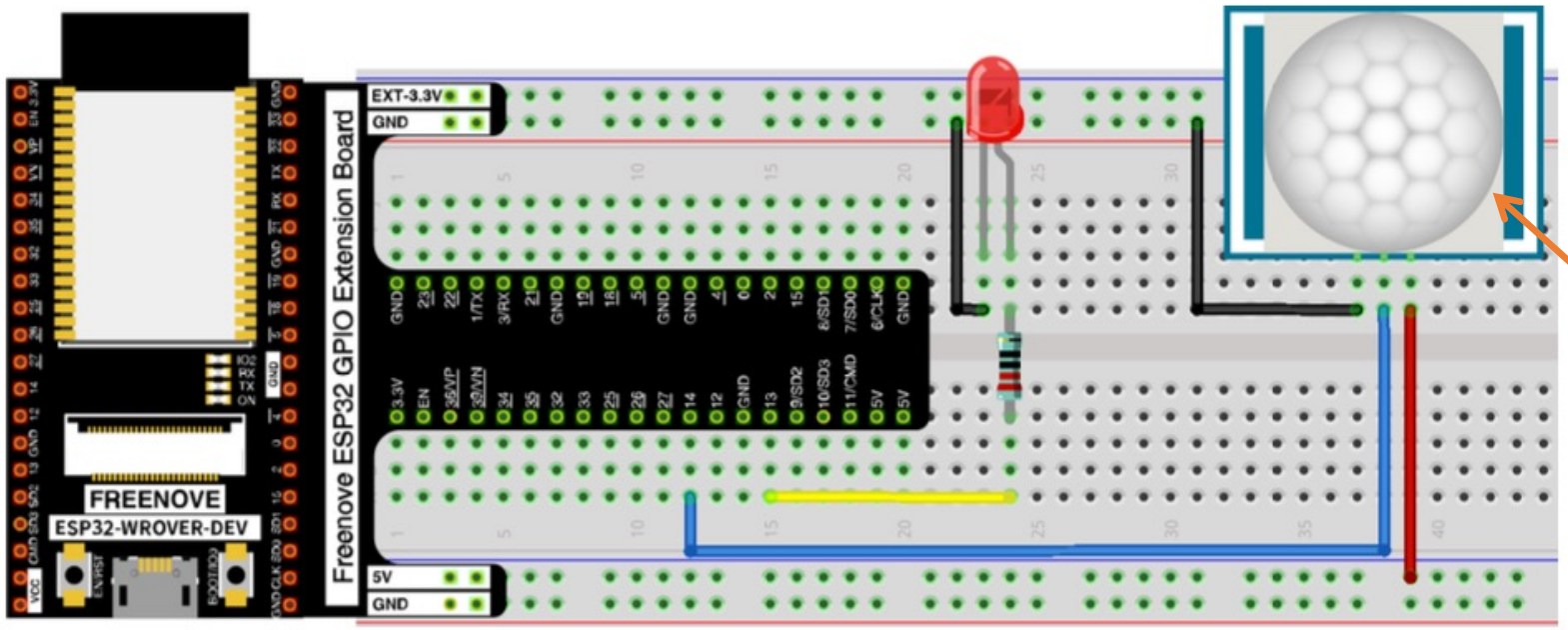
# PIR sensor board

| Top | Bottom | Schematic |
| --- | --- | --- |
|   |   |   |

# Schematic


5V
3.3V
5V
EXT-3.3V
3.3V
EN
36/VP
39/VN
34
35
32
33
25
26
27
14
12
GND
13
9/SD2
10/SD3
11/CMD
5V
5V
GND
GND
23
22
1/TX
3/RX
21
GND
19
18
5
GND
GND
4
0
2
15
8/SD1
7/SD0
6/CLK
GND
ESP32-WROVER
GPIO Extension
5V
HC-SR501
VCC
OUT
GND
3
2
1
LED1
Red (633nm)
R1
220Ω

# Connections




The PIR sensor will not mount on the board. Connect it via Male-to-female connectors/wires

# Code

```
from machine import Pin
import time

sensorPin=Pin(14,Pin.IN)
ledPin=Pin(13,Pin.OUT)

try:
    while True:
      if not sensorPin.value():
        ledPin.value(1)  #Set led turn on
      else:
        ledPin.value(0)  #Set led turn off
except:
    pass
```

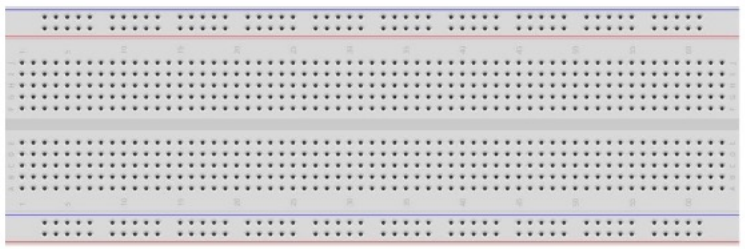
# How to run your code?

- Connect components to ESP32 (Double check the order)
- Power up ESP32 by connecting to laptop
- Open Thonny and make sure only *boot.py* is on ESP32. Delete remaining files
- Upload your program file
- Reset ESP32 (by clicking a small button near the USB input)
- Now your project should work.

# Ultrasonic Ranging

# Components

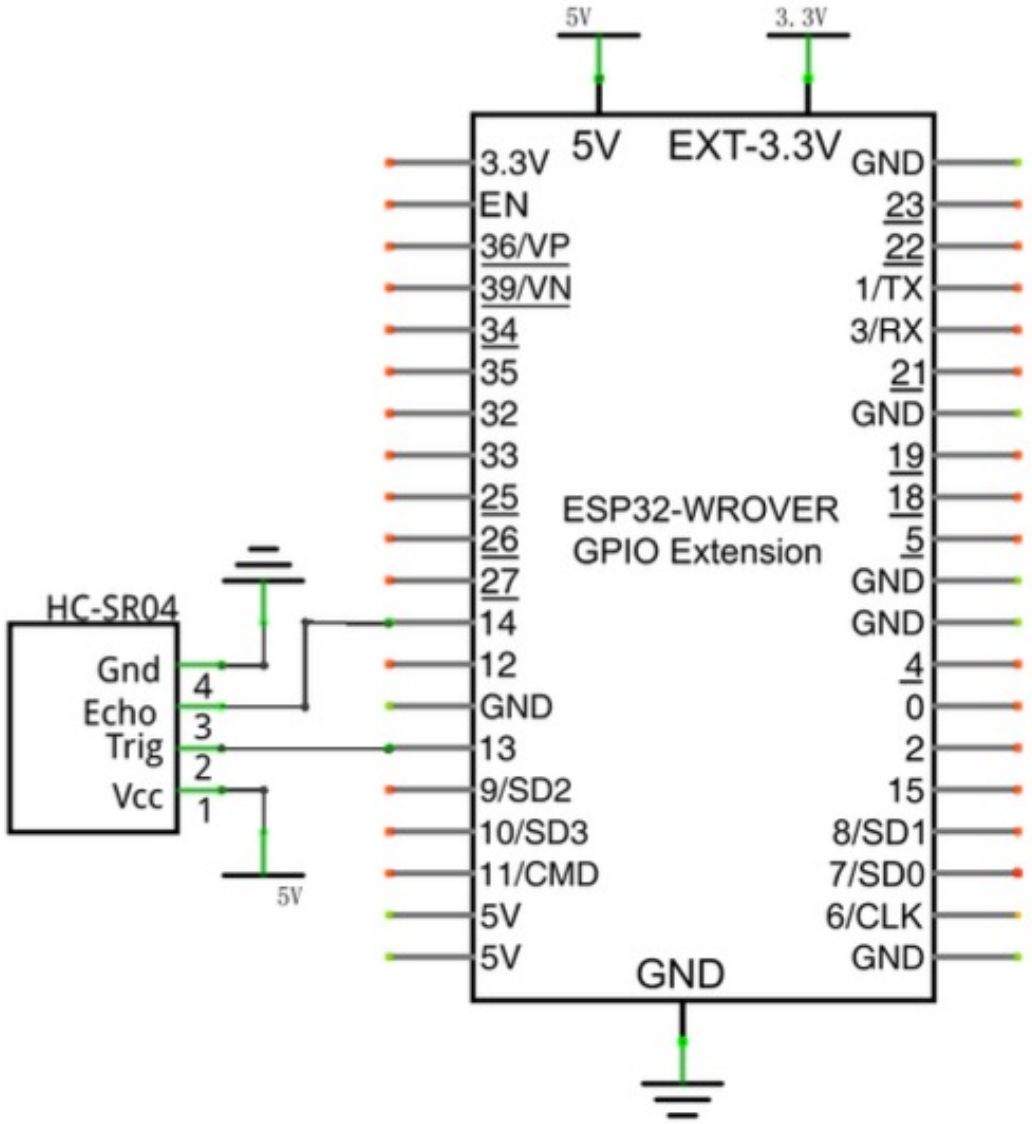
Breadboard x1



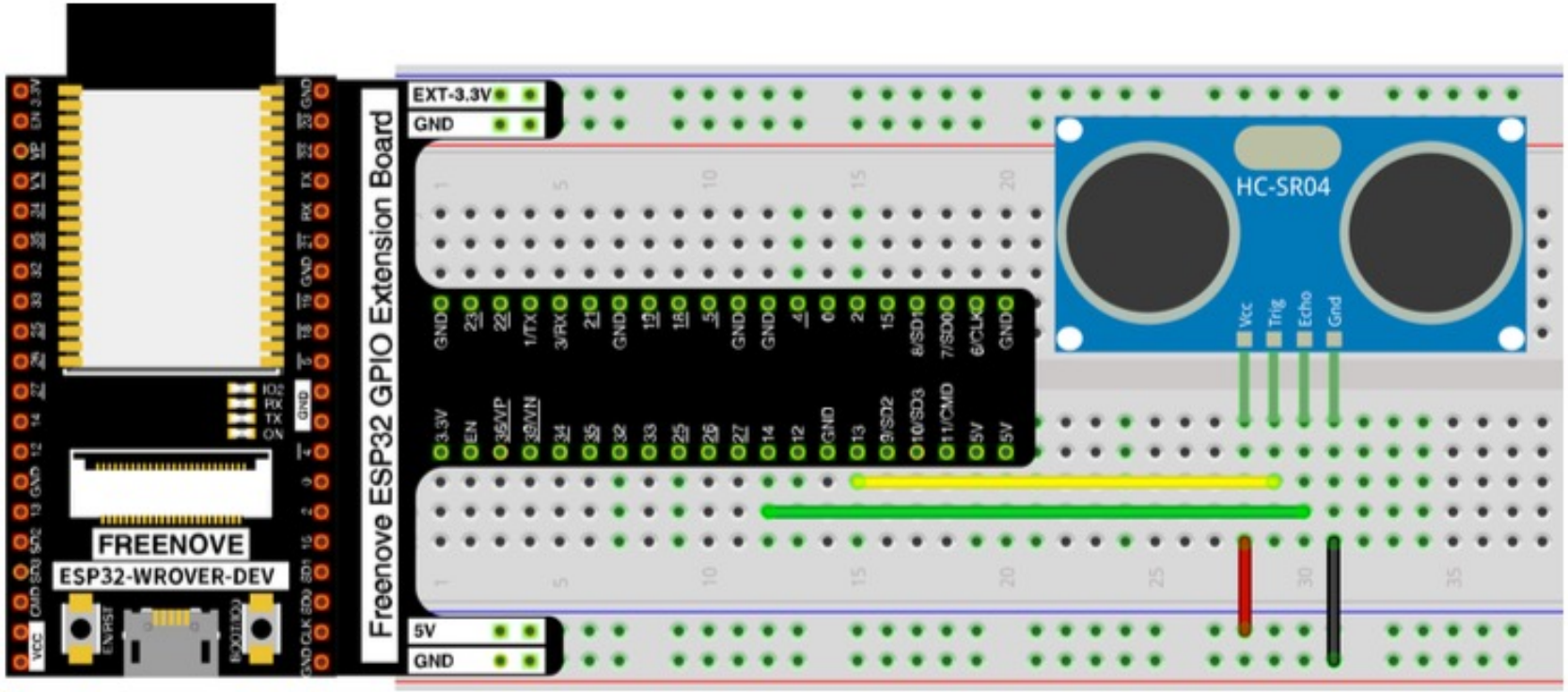
Jumper F/M x4

HC SR04 x1

# Schematic


5V
3. 3V
5V
EXT-3.3V
3.3V
EN
36/VP
39/VN
34
35
32
33
25
26
27
14
12
GND
13
9/SD2
10/SD3
11/CMD
5V
5V
GND
GND
23
22
1/TX
3/RX
21
GND
19
18
5
GND
GND
4
0
2
15
8/SD1
7/SD0
6/CLK
GND
ESP32-WROVER
GPIO Extension
HC-SR04
Gnd
Echo
Trig
Vcc
4
3
2
1
5V

# Connections



# Code

```
soundVelocity=340
distance=0

def getSonar():
    trigPin.value(1)
    time.sleep_us(10)
    trigPin.value(0)
    while not echoPin.value():
        pass
    pingStart=time.ticks_us()
    while echoPin.value():
        pass
    pingStop=time.ticks_us()
    pingTime=time.ticks_diff(pingStop,pingStart)
    distance=pingTime*soundVelocity//2//10000
    return int(distance)

time.sleep_ms(2000)
while True:
    time.sleep_ms(500)
    print('Distance: ',getSonar(),'cm' )
```

# How to run your code?

- Connect components to ESP32 (Double check the order)
- Powerup ESP32 by connecting to laptop
- Open Thonny and make sure only *boot.py* is on ESP32. Delete remaining files
- Reset ESP32 (by clicking a small button near the USB input)
- Now your project should work.