ESP 32



ESP01





ESP02

ESP03





ESP04

ESP05





ESP06

ESP07





ESP08

ESP09





ESP10

ESP11





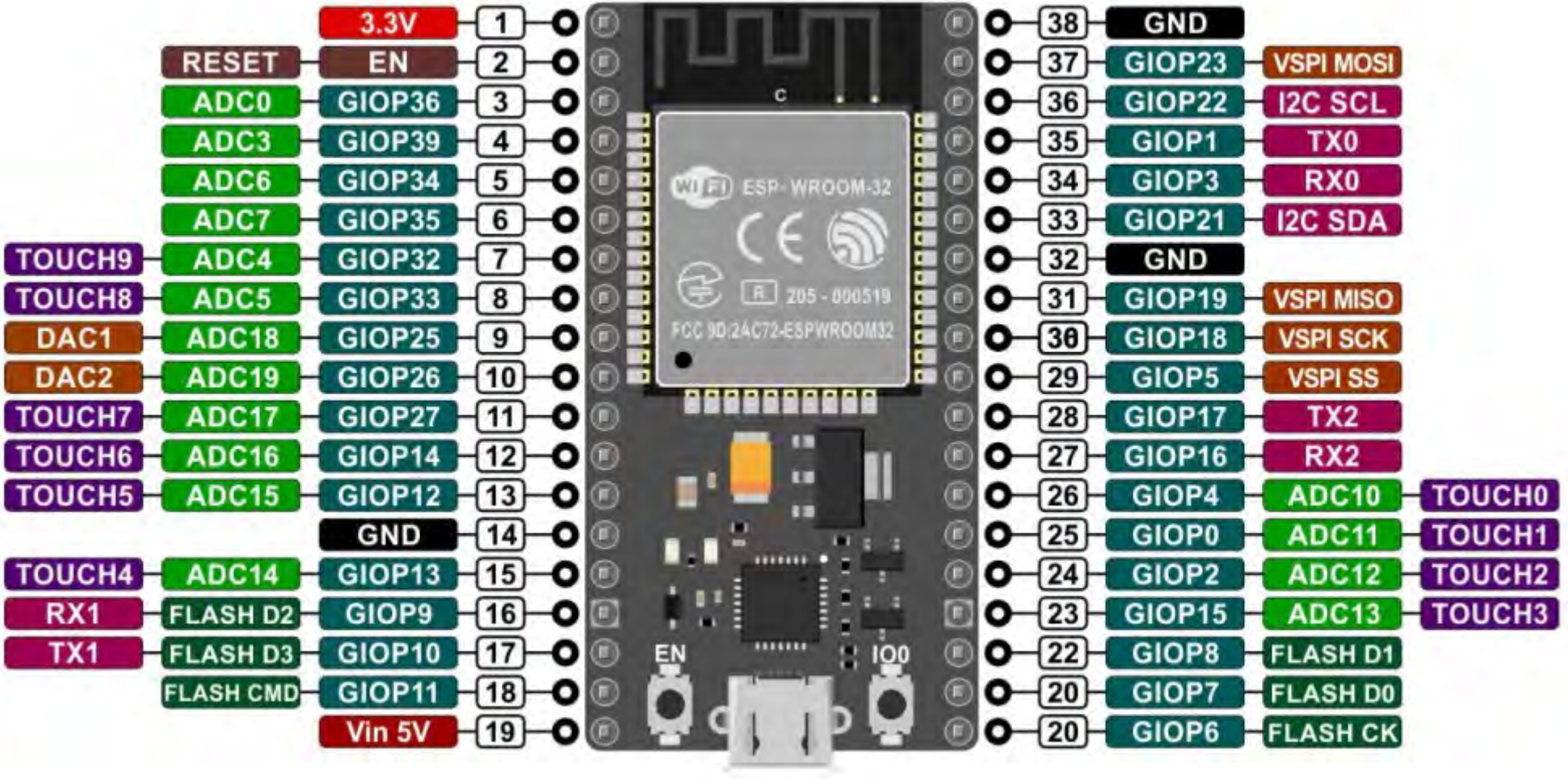
ESP12



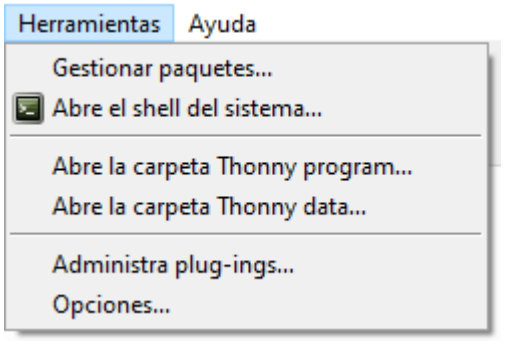
ESP32



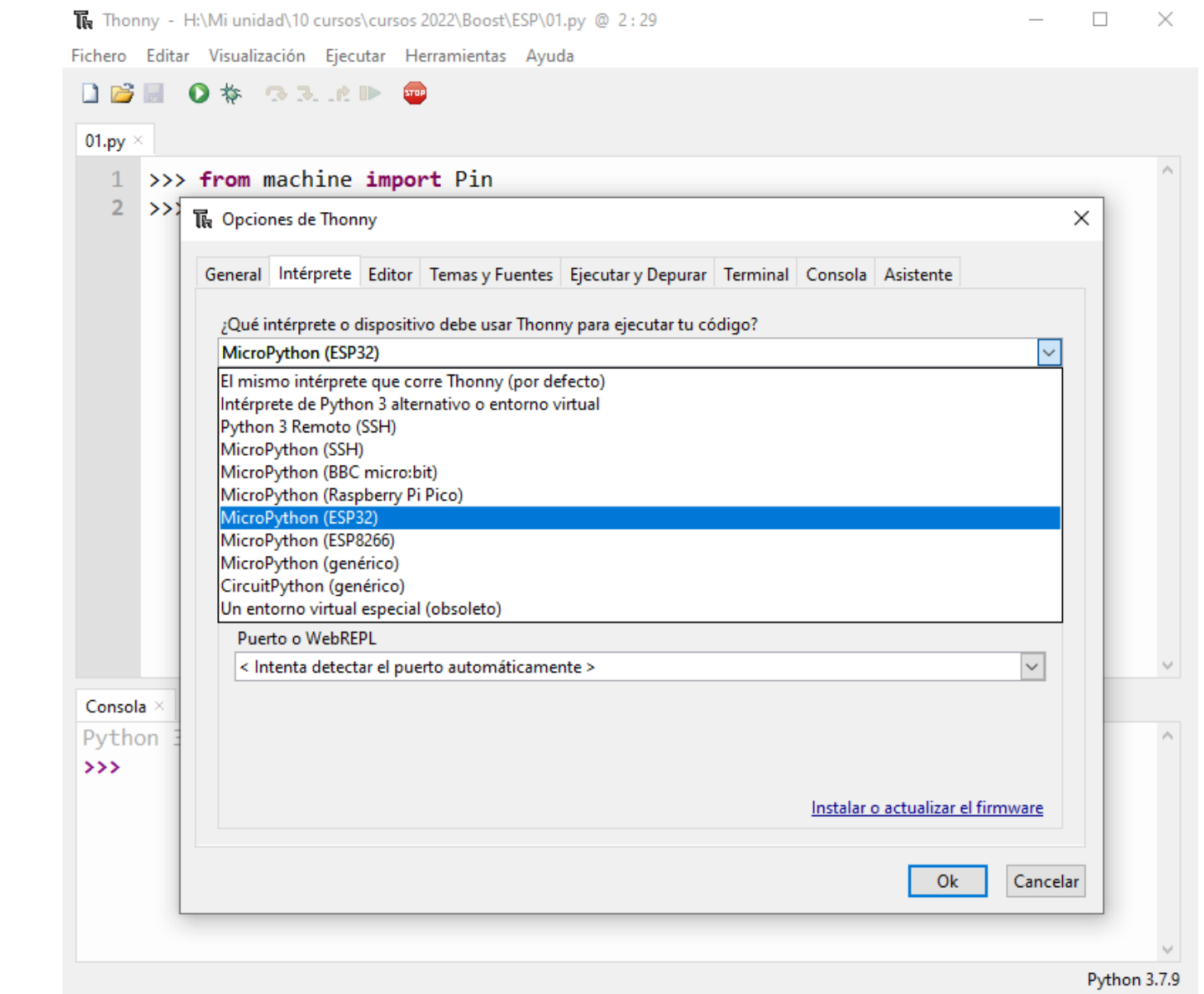
ESP32s


3.3V 1
RESET EN 2
ADC0 GIOP36 3
ADC3 GIOP39 4
ADC6 GIOP34 5
ADC7 GIOP35 6
TOUCH9 ADC4 GIOP32 7
TOUCH8 ADC5 GIOP33 8
DAC1 ADC18 GIOP25 9
DAC2 ADC19 GIOP26 10
TOUCH7 ADC17 GIOP27 11
TOUCH6 ADC16 GIOP14 12
TOUCH5 ADC15 GIOP12 13
GND 14
TOUCH4 ADC14 GIOP13 15
RX1 FLASH D2 GIOP9 16
TX1 FLASH D3 GIOP10 17
FLASH CMD GIOP11 18
Vin 5V 19
38 GND
37 GIOP23 VSPI MOSI
36 GIOP22 I2C SCL
35 GIOP1 TX0
34 GIOP3 RX0
33 GIOP21 I2C SDA
32 GND
31 GIOP19 VSPI MISO
30 GIOP18 VSPI SCK
29 GIOP5 VSPI SS
28 GIOP17 TX2
27 GIOP16 RX2
26 GIOP4 ADC10 TOUCH0
25 GIOP0 ADC11 TOUCH1
24 GIOP2 ADC12 TOUCH2
23 GIOP15 ADC13 TOUCH3
22 GIOP8 FLASH D1
20 GIOP7 FLASH D0
20 GIOP6 FLASH CK
ESP-WROOM-32
205 - 000519
FCC ID:2AC72-ESPWROOM32
EN
IO0

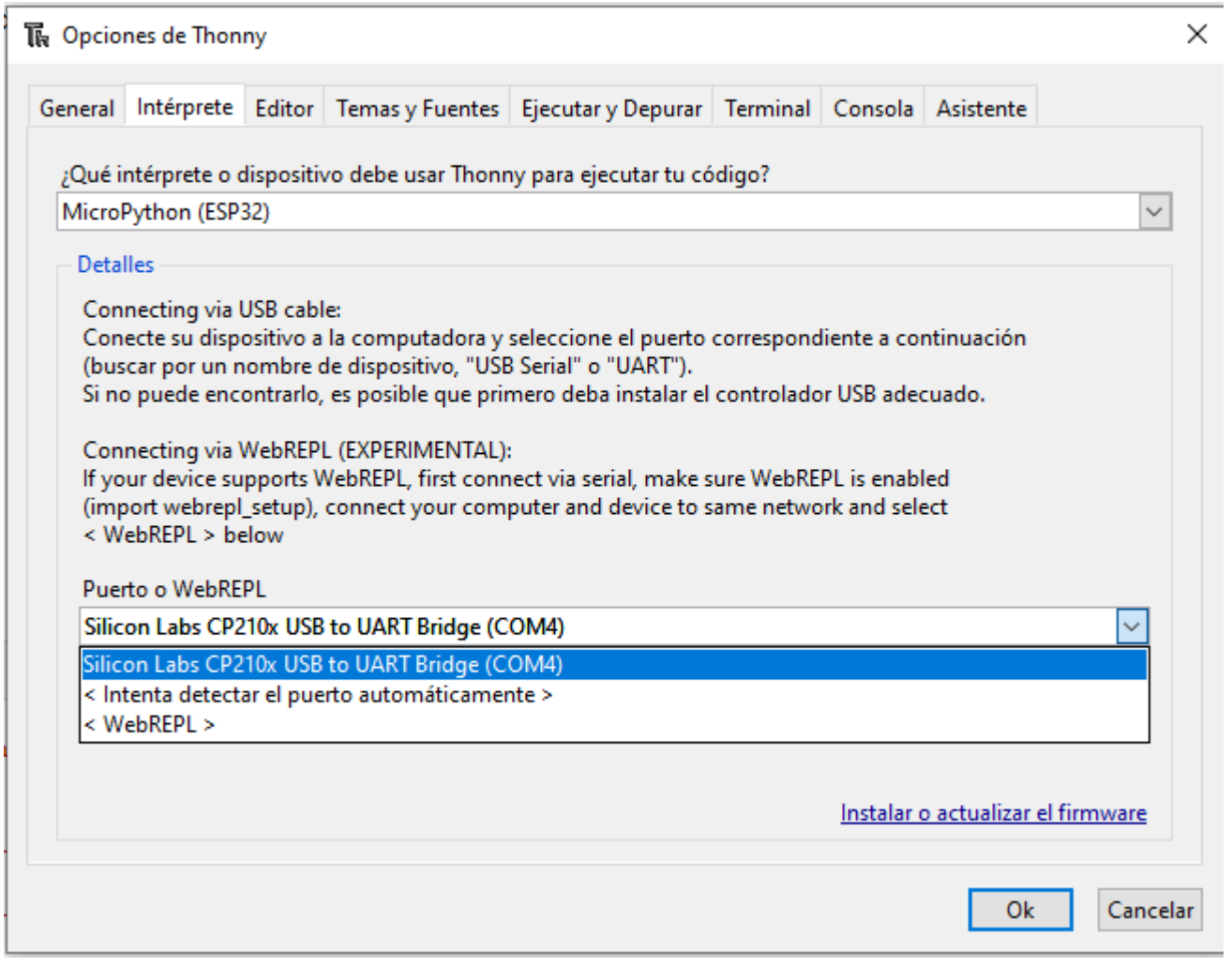
Con thonny



Herramientas/opciones



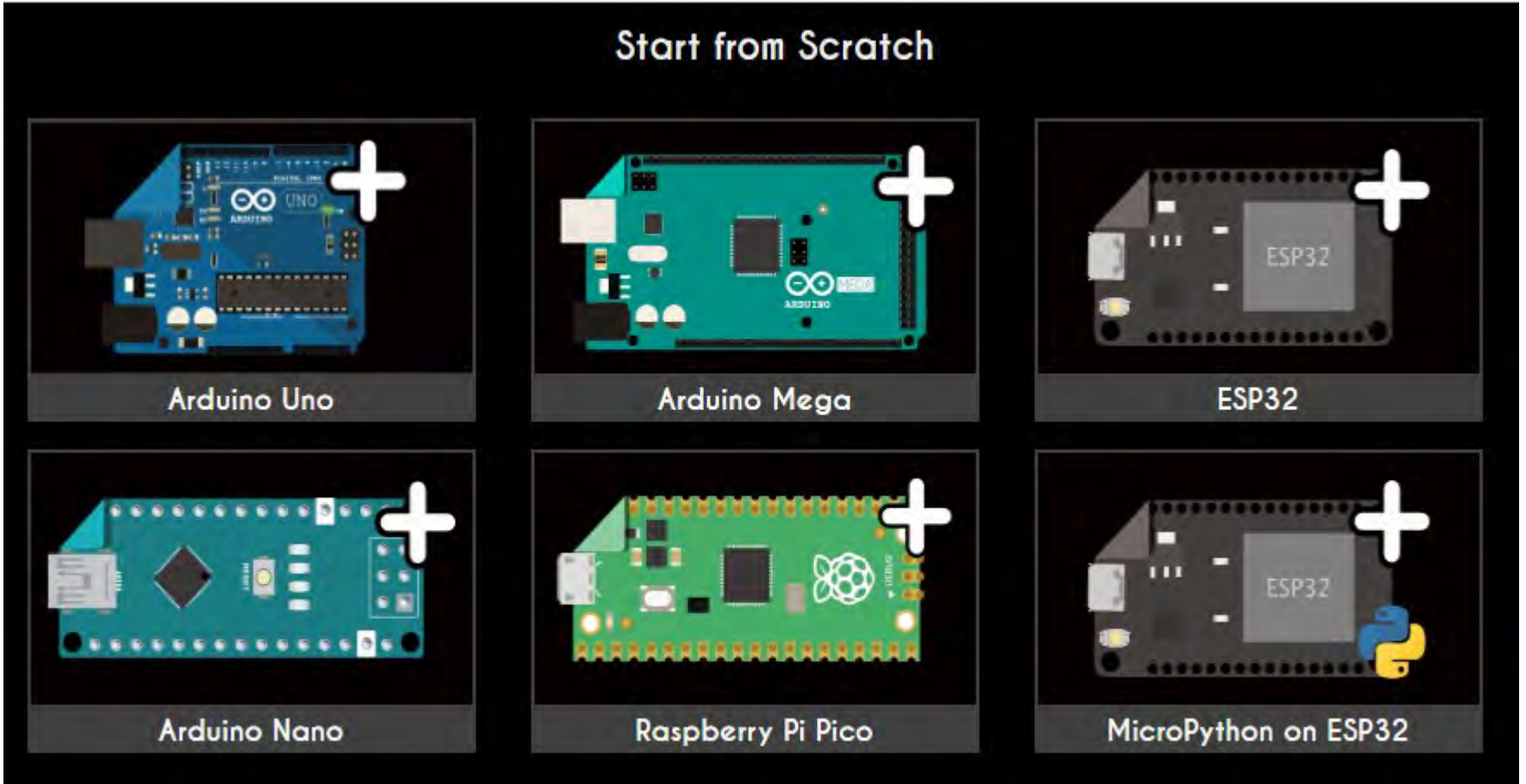
Interprete/MicroPython (ESP32)



Escoger el puerto

wokwi

https://wokwi.com/



WOKWI GUARDAR COMPARTIR Docs

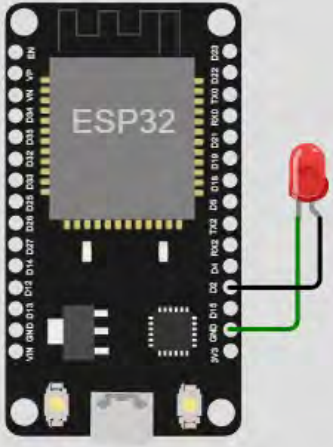
sketch.ino diagram.json Library Manager

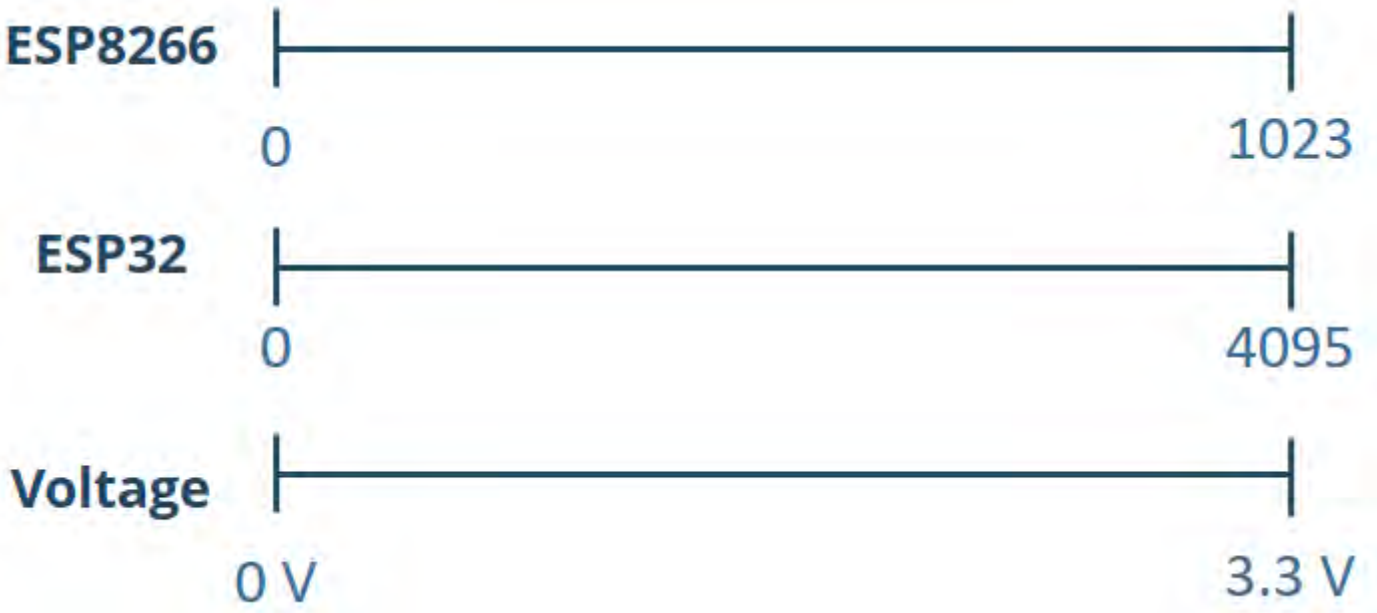
```ino
//Paso 1
int pin_led = 2;

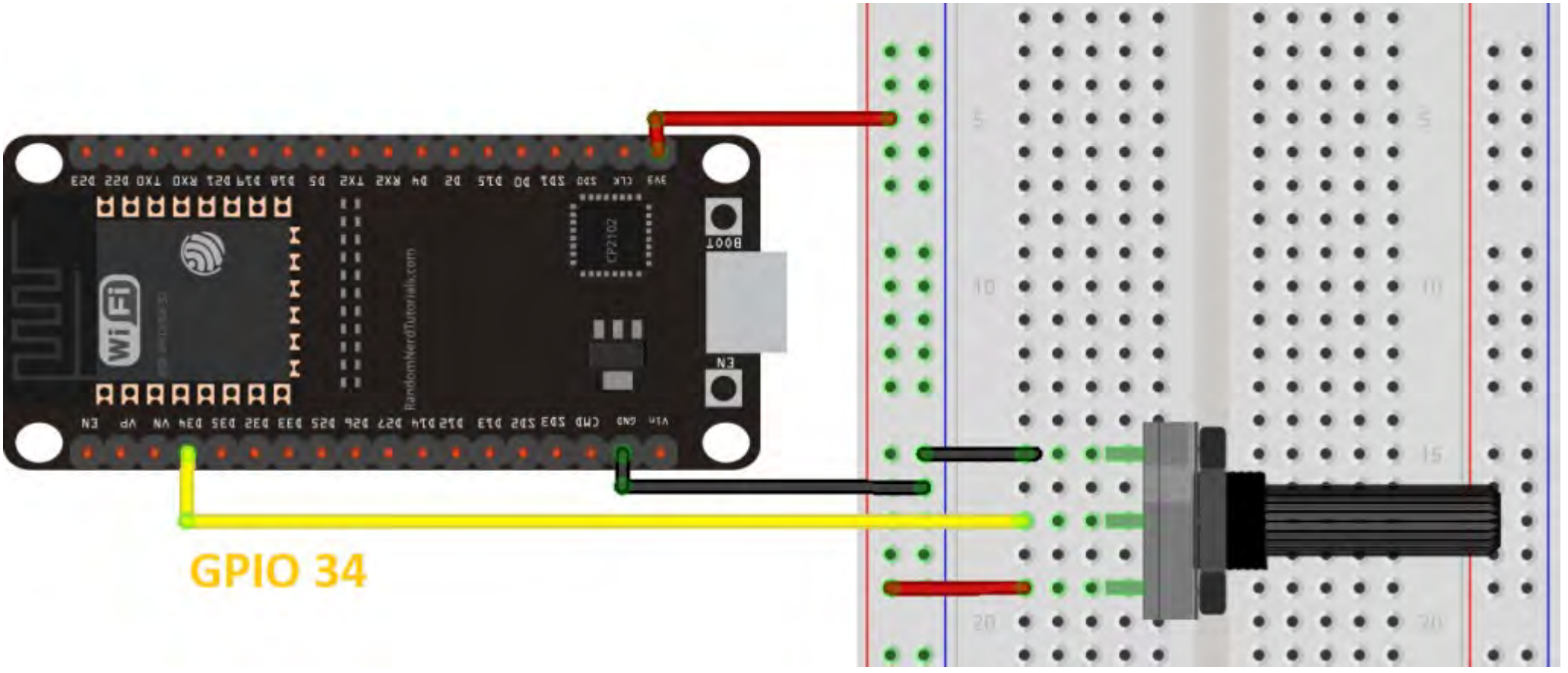
//Paso 2
void setup() {
    pinMode(pin_led, OUTPUT);
}

//Paso 3
void loop() {
    digitalWrite(pin_led, HIGH);
    delay(1000);
    digitalWrite(pin_led, LOW);
    delay(1000);
}
```

Simulation




ESP8266
0
1023
ESP32
0
4095
Voltage
0 V
3.3 V


GPIO 34

```
from machine import Pin, ADC
from time import sleep

pot = ADC(Pin(34))
pot.atten(ADC.ATTN_11DB)       #Full range: 3.3v

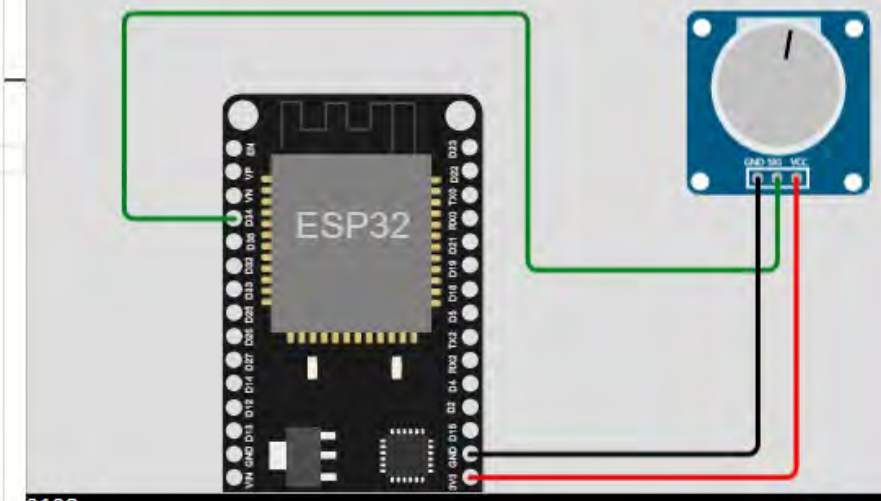
while True:
  pot_value = pot.read()
  print(pot_value)
  sleep(0.1)
```

WOKWI SAVE SHARE Docs

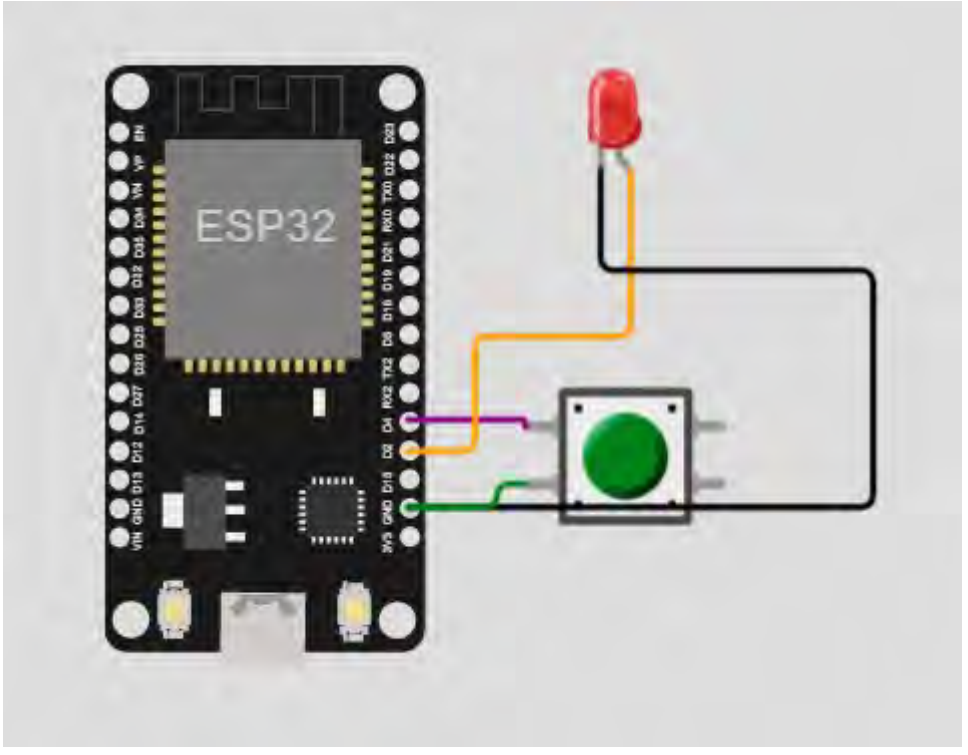
main.py diagram.json

```
1  from machine import Pin, ADC
2  from time import sleep
3
4  pot = ADC(Pin(34))
5  pot.atten(ADC.ATTN_11DB)       #Full range: 3.3v
6
7  while True:
8      pot_value = pot.read()
9      print(pot_value)
10     sleep(0.1)
```

Simulation




```
2198
2198
2198
2198
2198
2198
2198
2198
2198
2198
```


ESP32

```
from machine import Pin

pin_02 = Pin(2, Pin.OUT, value=0)  # El GPIO_02 funciona como salida para encender el LED
pin_04 = Pin(4, Pin.IN)            # El GPIO_04 funciona como entrada para detectar el pulsador

while True:
    if pin_04.value() == 1:        # si se pulsa el pulsador...
        pin_02.on()                # ...se encenderá el LED
    pin_02.off()                   # si no detecta nada el LED se permanecerá apagado
```

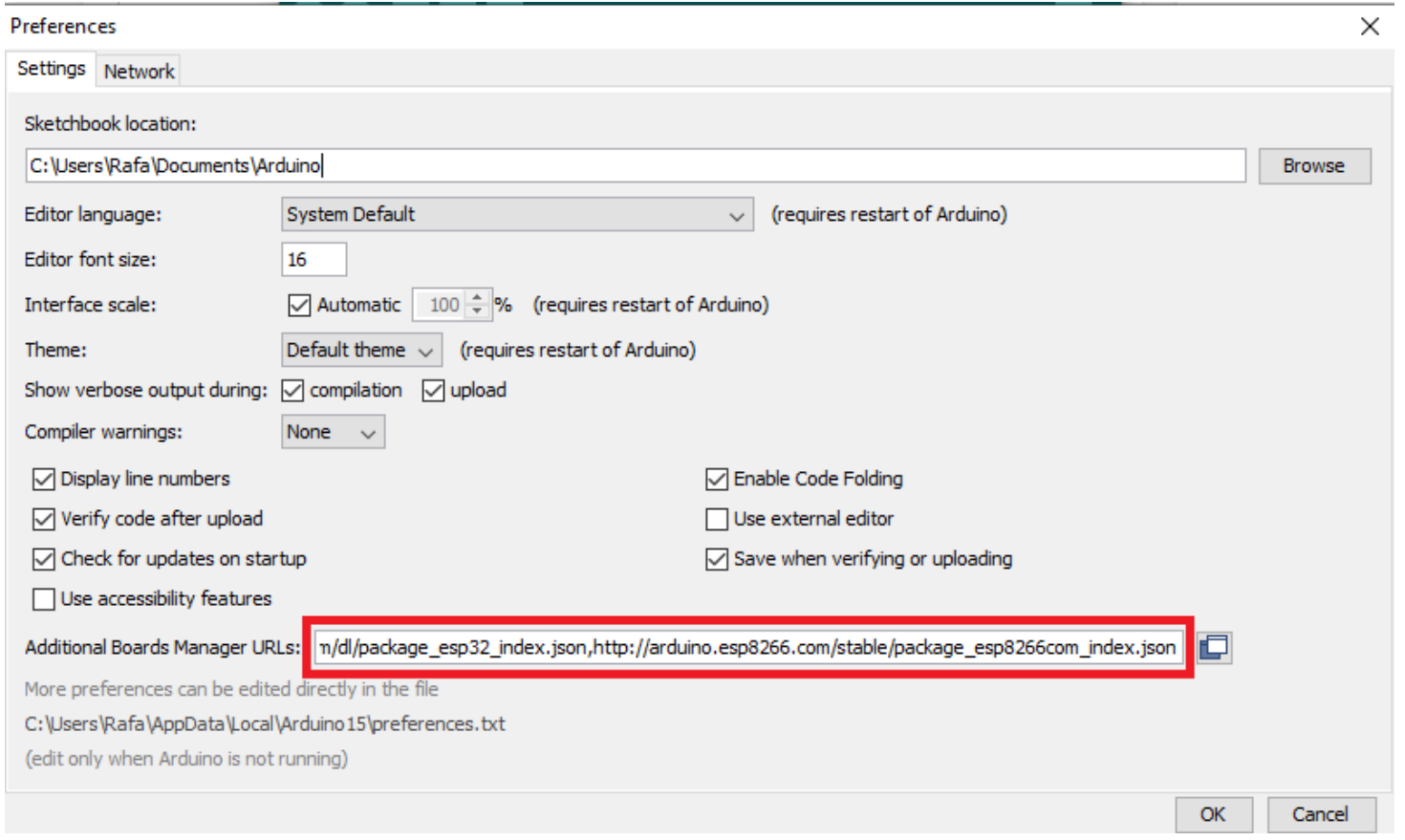
```
from machine import Pin

pin_02 = Pin(2, Pin.OUT, value=0)
pin_04 = Pin(4, Pin.IN)

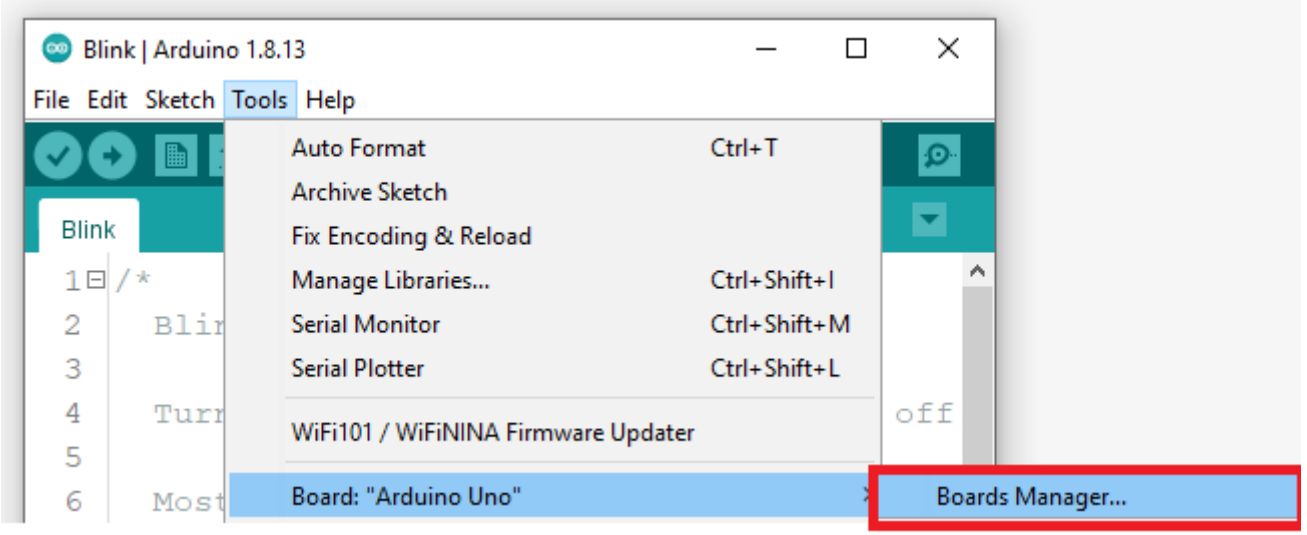
while True:
    if pin_04.value() == 1:
        pin_02.on()
    pin_02.off()
```

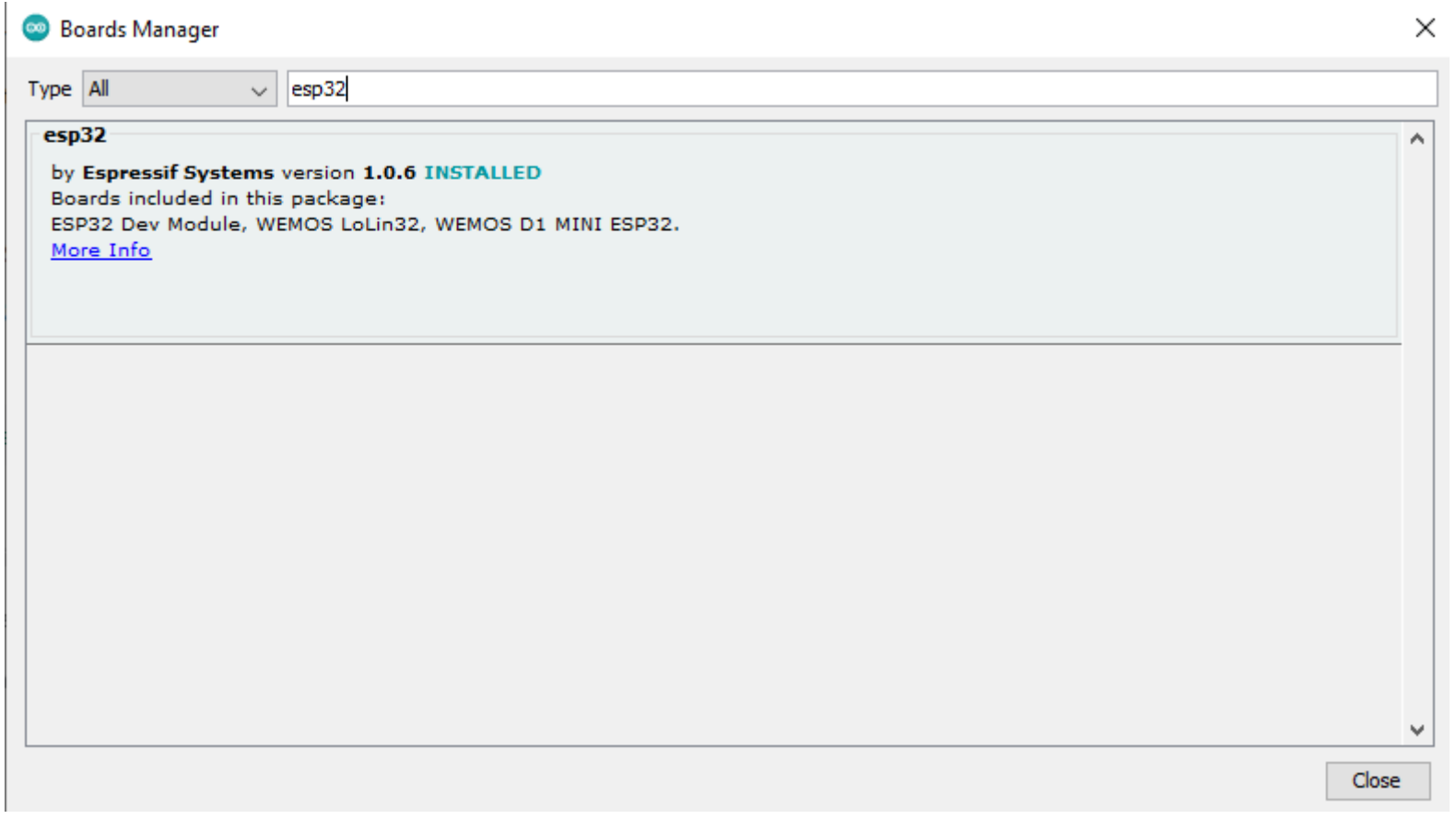
Con IDE de Arduino

archivo->Preferencias



https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

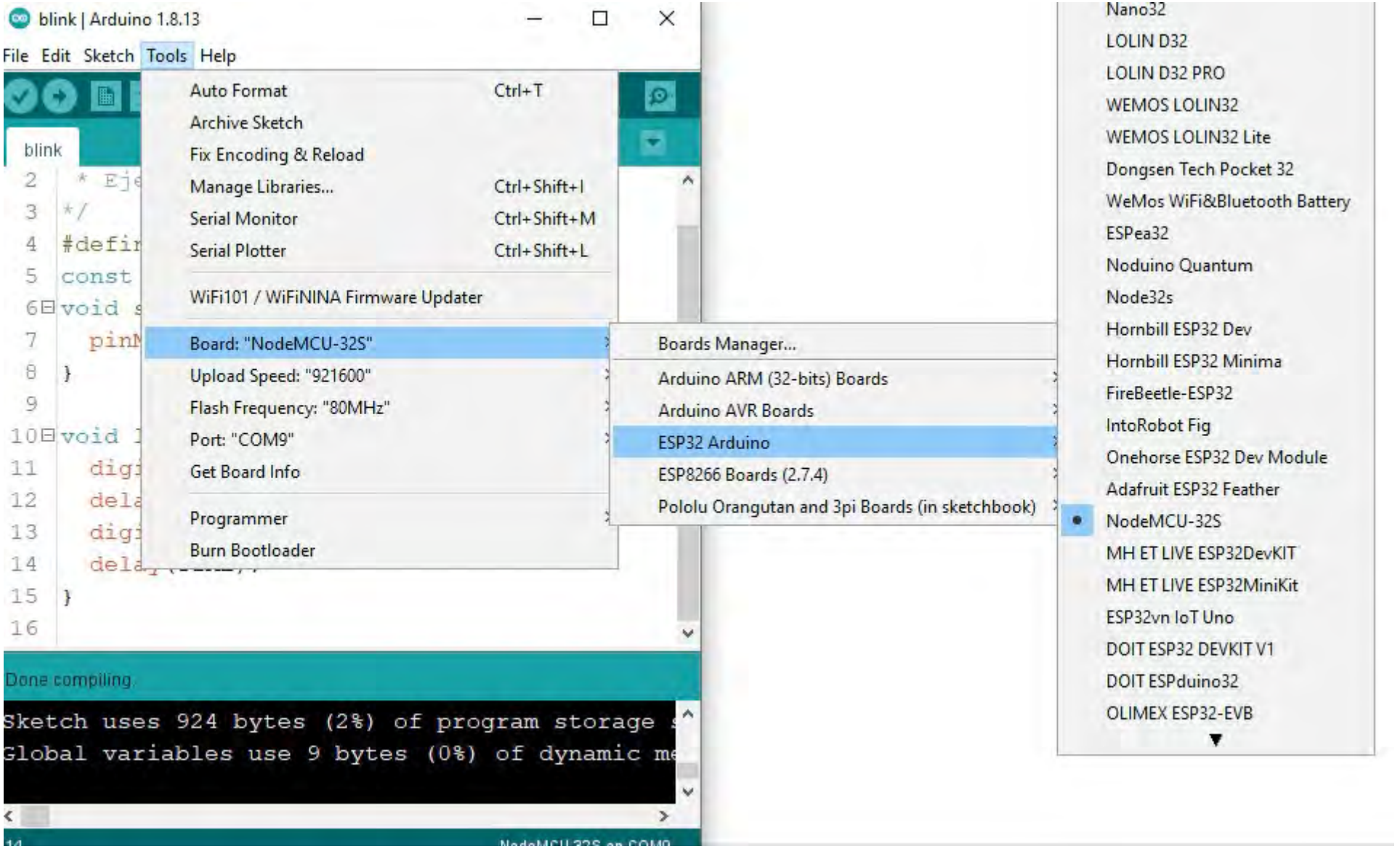

Blink | Arduino 1.8.13
File Edit Sketch Tools Help
Blink
1 /*
2 Blir
3
4 Turr
5
6 Most
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Uno"
Boards Manager...
off


Boards Manager
Type
All
esp32
esp32
by Espressif Systems version 1.0.6 INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info
Close

La tarjeta Esp32 cuanta con un led interno conectado en el GPIO2 por lo que no es necesario agregar un led.

```
/*
 * Ejemplo blink ESP32s
*/
#define Led 2
const int TIME = 1000;
void setup() {
  pinMode(Led, OUTPUT);
}

void loop() {
  digitalWrite(Led, HIGH);
  delay(TIME);
  digitalWrite(Led, LOW);
  delay(TIME);
}
```


blink | Arduino 1.8.13
File Edit Sketch Tools Help
blink
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "NodeMCU-32S"
Upload Speed: "921600"
Flash Frequency: "80MHz"
Port: "COM9"
Get Board Info
Programmer
Burn Bootloader
Boards Manager...
Arduino ARM (32-bits) Boards
Arduino AVR Boards
ESP32 Arduino
ESP8266 Boards (2.7.4)
Pololu Orangutan and 3pi Boards (in sketchbook)
Nano32
LOLIN D32
LOLIN D32 PRO
WEMOS LOLIN32
WEMOS LOLIN32 Lite
Dongsen Tech Pocket 32
WeMos WiFi&Bluetooth Battery
ESPea32
Noduino Quantum
Node32s
Hornbill ESP32 Dev
Hornbill ESP32 Minima
FireBeetle-ESP32
IntoRobot Fig
Onehorse ESP32 Dev Module
Adafruit ESP32 Feather
NodeMCU-32S
MH ET LIVE ESP32DevKIT
MH ET LIVE ESP32MiniKit
ESP32vn IoT Uno
DOIT ESP32 DEVKIT V1
DOIT ESPduino32
OLIMEX ESP32-EVB
Done compiling.
Sketch uses 924 bytes (2%) of program storage
Global variables use 9 bytes (0%) of dynamic me

Seleccionar el puerto y subimos el programa

Es importante antes de presionar subir debemos presionar el botón "BOOT" y cuando aparezca la leyenda subiendo en el IDE de arduino soltar el botón.

El botón de BOOT es el siguiente

