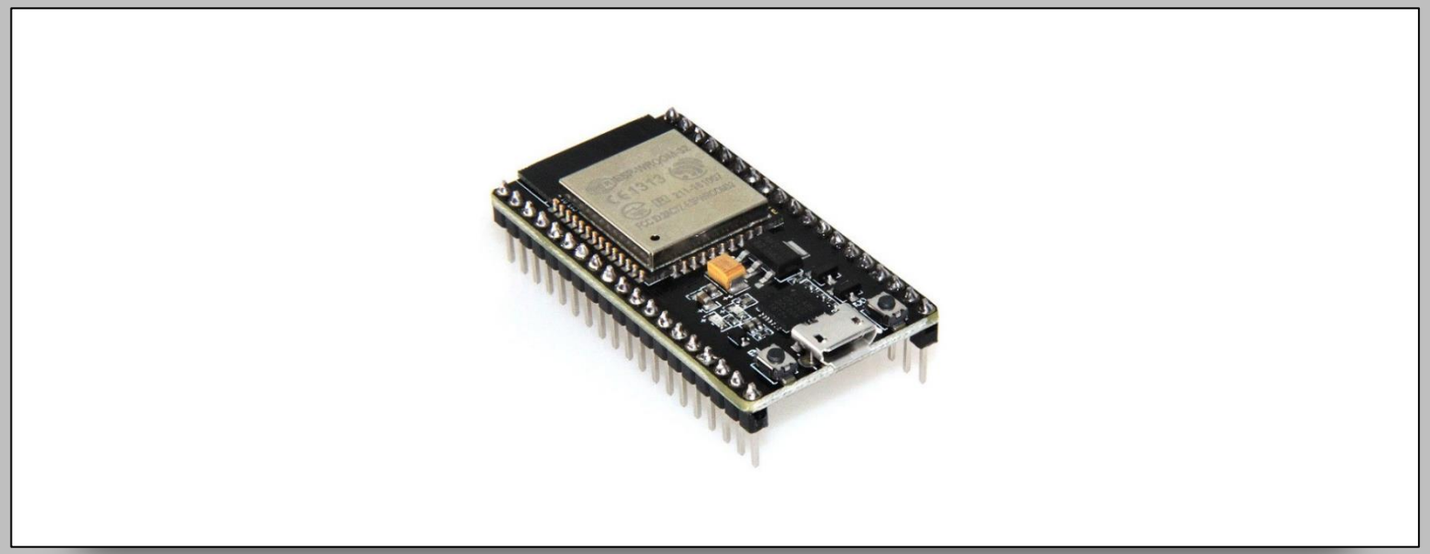
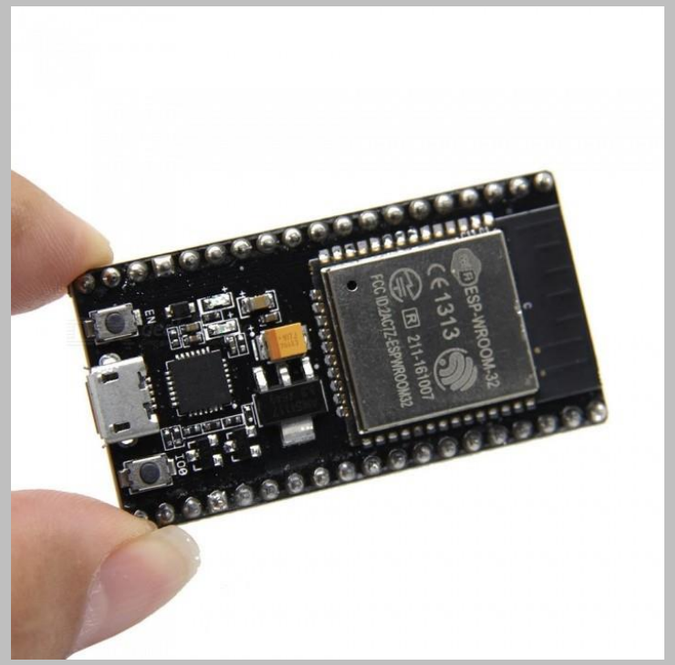
Introdução ao NodeMCU-32S ESP-WROOM-32



Por Fernando Koyanagi

Características principais



- ✓ Chip com Wi-Fi embutido : padrão 802.11 B/G/N, operando na faixa de 2.4 a 2.5GHz
- ✓ Modos de operação : Client, Access Point, Station+Access Point
- ✓ Microprocessador dual core Tensilica Xtensa 32-bit LX6
- ✓ Clock ajustável de 80MHz até 240MHz
- ✓ Tensão de operação : 3.3 VDC
- ✓ Possui SRAM de 512KB
- ✓ Possui ROM de 448KB
- ✓ Possui memória flash externa de 32Mb (4 megabytes)
- ✓ Corrente máxima por pino é de 12mA (recomenda-se usar 6mA)
- ✓ Possui 36 GPIOs
- ✓ GPIOs com função PWM / I2C e SPI
- ✓ Possui Bluetooth v4.2 BR / EDR e BLE (Bluetooth Low Energy)

Em www.fernandok.com

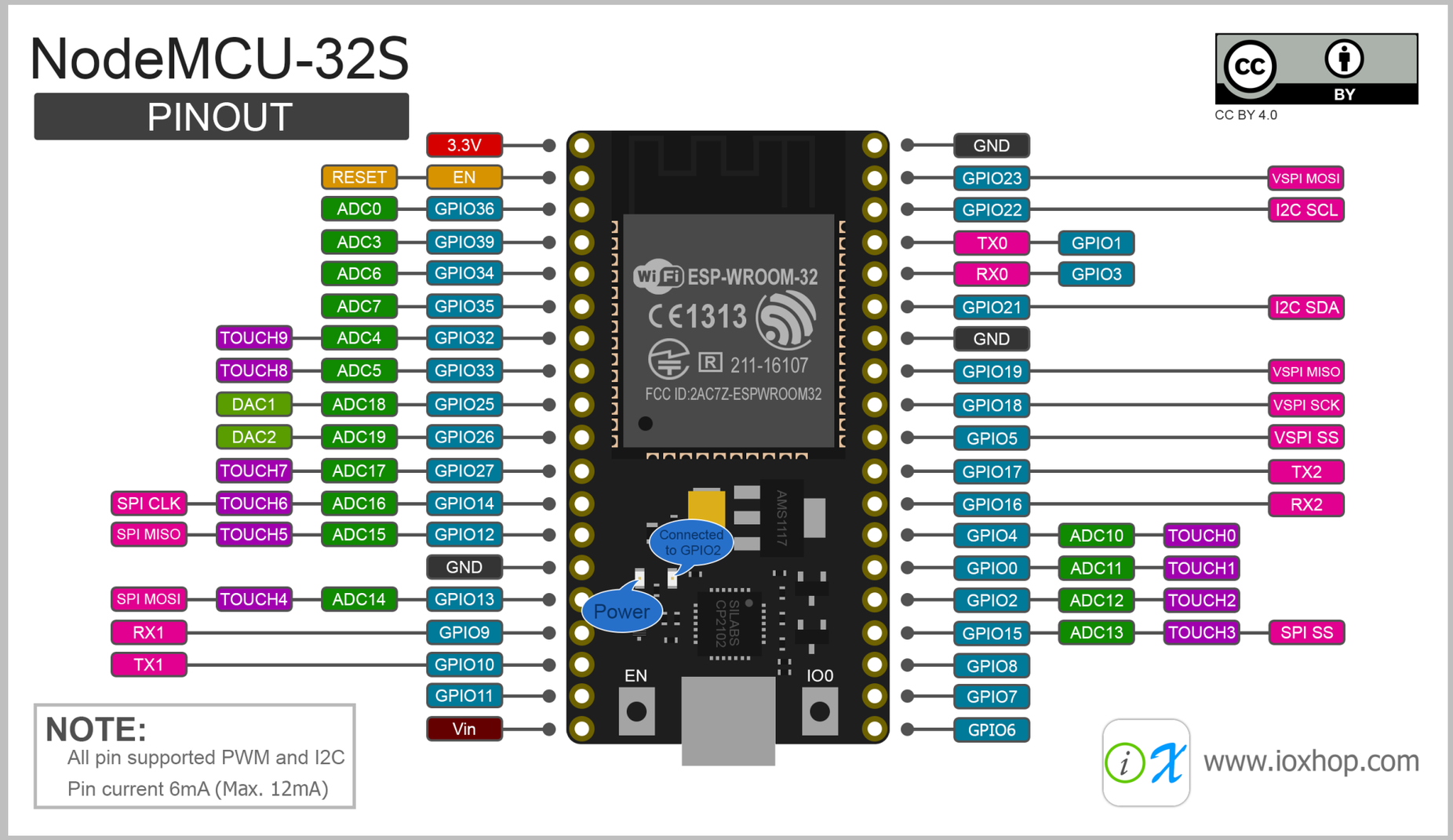
Download arquivo PDF dos diagramas
Download arquivo INO do código fonte

Comparativo entre ESP32, ESP8266 e Arduino R3

	ESP32	ESP8266	ARDUINO UNO R3
Cores	2	1	1
Arquitetura	32 bits	32 bits	8 bits
Clock	160MHz	80MHz	16MHz
WiFi	Sim	Sim	Não
Bluetooth	Sim	Não	Não
RAM	512KB	160KB	2KB
FLASH	16Mb	16Mb	32KB
GPIO	36	17	14
Interfaces	SPI / I2C / UART / I2S / CAN	SPI / I2C / UART / I2S	SPI / I2C / UART
ADC	18	1	6
DAC	2	0	0

Tipos de ESP32

WiFi NodeMCU-32S ESP-WROOM-32



Configurando IDE do Arduino (windows)

Vejamos a seguir como configurar a IDE do Arduino para podermos compilar para o ESP-32.

1. Faça o download dos arquivos através do link

2. Descompacte o arquivo e copie o conteúdo para o seguinte caminho:

C:/Users/[YOUR_USER_NAME]/Documents/Arduino/hardware/espressif/esp32

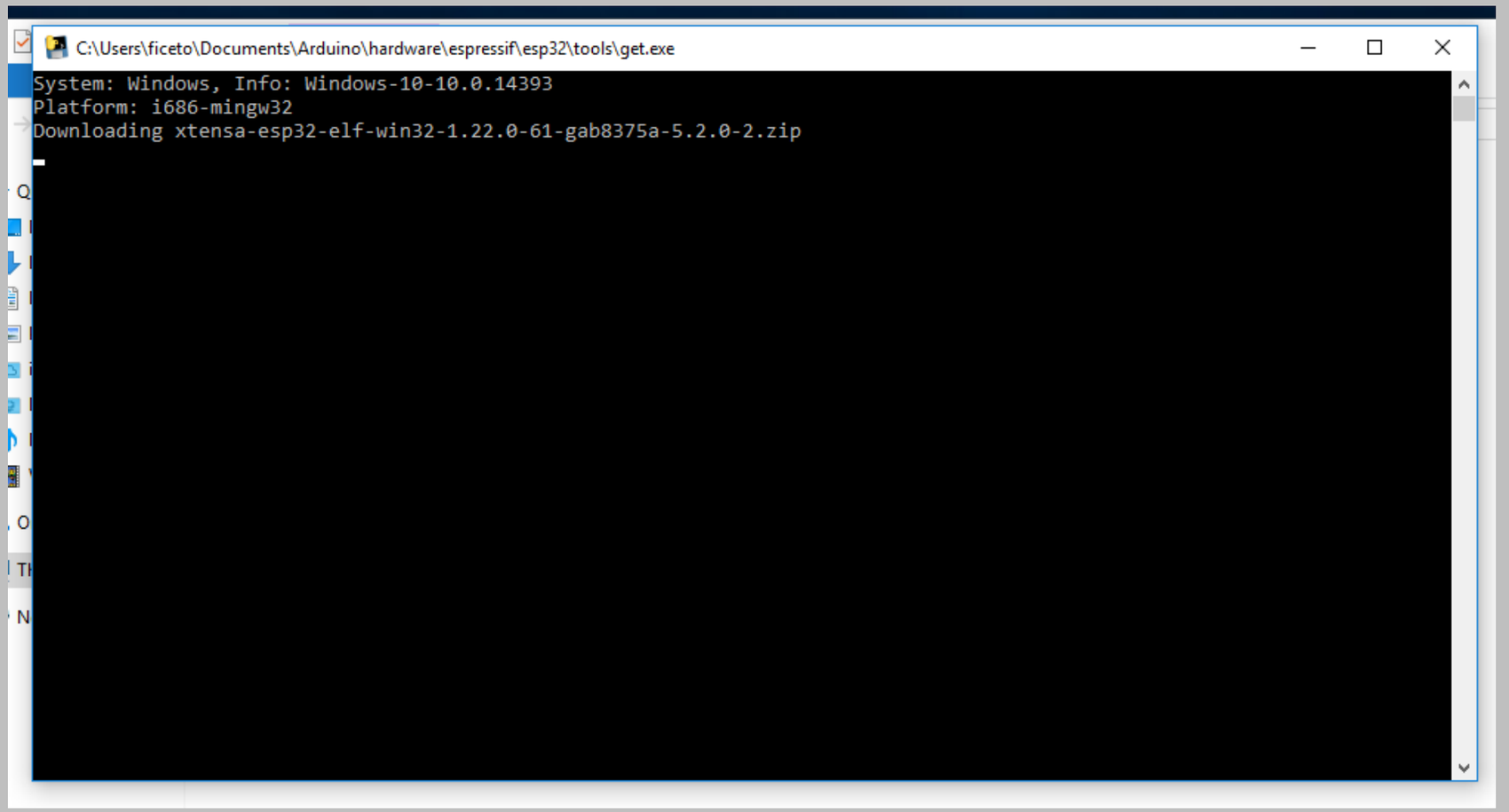
Obs: caso não exista o diretório “espressif” e “esp32”, basta criá-los normalmente.

Configurando IDE do Arduino (windows)

3. **Abra o diretório**

C:/Users/[YOUR_USER_NAME]/Documents/Arduino/hardware/espressif/esp32/tools

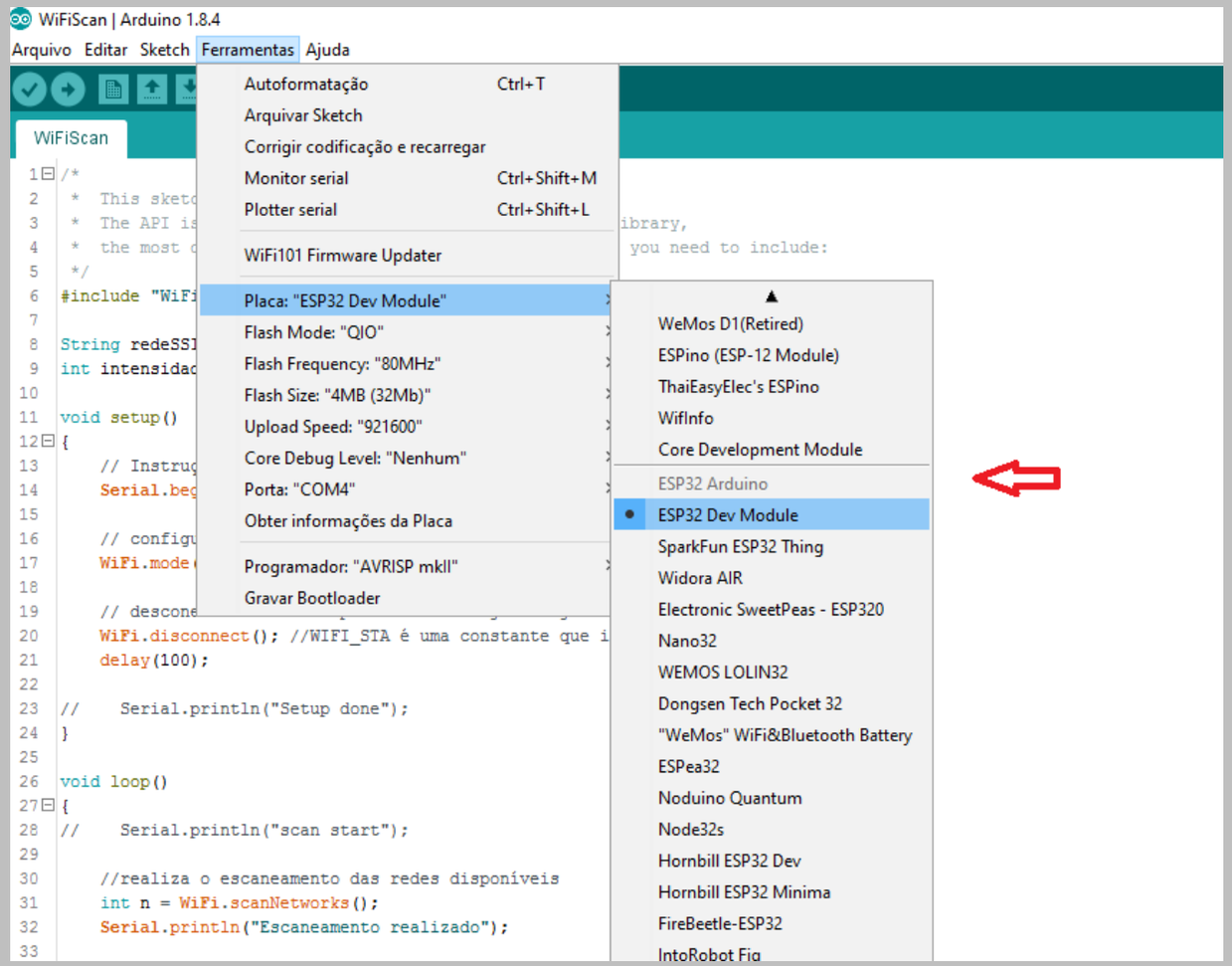
Execute o arquivo “get.exe”.



Configurando IDE do Arduino (windows)

4. Após a finalização do “get.exe”, plugue o ESP32, aguarde os drivers serem instalados (ou instale manualmente).

Pronto, agora basta escolher a placa do ESP32 em “ferramentas >> placa” e compilar seu código.



WiFi Scan

Vejamos a seguir um exemplo de como buscar as redes WiFi disponíveis próximas ao ESP-32, assim como a intensidade do sinal de cada uma delas. A cada escaneamento, também vamos descobrir qual a rede com a melhor intensidade de sinal.

Vamos ao código

Primeiramente vamos incluir a biblioteca “WiFi.h”, ela será necessária para nos permitir trabalhar com a placa de rede do nosso dispositivo.

```
#include "WiFi.h"
```

A seguir, vamos declarar duas variáveis que serão utilizadas para guardar o SSID (nome) da rede e a intensidade do sinal.

```
String redeSSID = "";
int intensidadeSinal = -9999;
```

Setup

Na função *setup()*, definiremos o modo de comportamento WiFi do nosso dispositivo. Nesse caso, como o objetivo é procurar por redes disponíveis, vamos configurar nosso dispositivo para trabalhar como “estação”.

```
void setup()
{
   // Instrução para inicializar o Serial, utilizaremos apenas para log no monitor.
   Serial.begin(115200);

   // configurando o modo de operação do WiFi como estação
   WiFi.mode(WIFI_STA); //WIFI_STA é uma constante que indica o modo estação

   // desconecta do access point caso ele já esteja conectado
   WiFi.disconnect();

}
```

Loop

Na função *loop()*, vamos fazer a busca pelas redes disponíveis e em seguida imprimir no log as redes encontradas. Para cada uma dessas redes faremos a comparação para encontrar a com maior intensidade de sinal.

```
void loop()
{
   //realiza o escaneamento das redes disponíveis
   int n = WiFi.scanNetworks();
   Serial.println("Escaneamento realizado");

   //verifica se encontrou alguma rede
   if (n == 0) {
     Serial.println("Nenhuma rede encontrada");
   }
   else {
    ...
  }
}
```

Loop

Continuando a função loop

```
void loop()
{
   ...
   else {
        redeSSID = "";
        intensidadeSinal= -9999;
        Serial.print(n);
        Serial.println(" redes encontradas");
        for (int i = 0; i < n; ++i) {
         // imprime no log cada uma das redes encontradas
            Serial.print(WiFi.SSID(i)); //nome da rede
            Serial.print(": ");
            Serial.print(WiFi.RSSI(i)); //intensidade do sinal
            ...
        }
   }
```

Loop

Continuando a função loop

```
void loop()
{
   ...
  else {
       ...
      // faz a comparação para saber se a rede encontrada tem melhor sinal do que a melhor rede até o momento.
       if(abs(WiFi.RSSI(i)) < abs(intensidadeSinal))
       {
          intensidadeSinal = WiFi.RSSI(i);
          redeSSID = WiFi.SSID(i);
          Serial.print("REDE COM MELHOR SINAL ENCONTRADA: ");
          Serial.print(redeSSID);
          Serial.print(" - SINAL : ");
          Serial.println(intensidadeSinal );
       }
     }//for
  }
  delay(5000);  // deixa um intervalo de 5 segundos para fazer um novo escaneamento
}
```

Loop

“if(abs(WiFi.RSSI(i)) < abs(intensidadeSinal))”

Repare que na instrução acima, utilizamos *abs()*, essa função pega o valor absoluto (ou seja, não negativo) do número. No nosso caso fizemos isso para achar o menor entre os valores na comparação, pois, a intensidade do sinal é dada como um número negativo, e quanto mais próxima de zero, melhor o sinal.

Link dos arquivos de configuração

https://github.com/espressif/arduino-esp32