# Introduction to ESP32

by Fernando Koyanagi

In this article we are going to talk about ESP32, which I consider to be an older brother of ESP8266. I really like this microcontroller because it has WiFi. Just so you have an idea, before ESP exists, if you needed an Arduino to have WiFi, you would have to spend between $ 200 and $ 300 to buy a Wifi adapter. The adapter for network cable is not that expensive, but for WiFi it has always been and still is expensive. But fortunately, Espressif Systems has launched ESP and is resolving our lives.

I like ESP32 with this format that has a USB port.

https://youtu.be/iZc8_X8JxYs

This NodeMCU scheme is easy to manipulate because it does not need any electronics. Just plug in the cable, power the device and program it. It works just like an Arduino.

Anyway, today we will talk about the general aspects of ESP32 and how to configure the Arduino IDE to program more devices of the type. Also we will make a program that searches the networks and shows which one is more powerful.

## Step 1: Key Features

Chip with built-in WiFi: standard 802.11 B / G / N, operating in the range of 2.4 to 2.5GHz

Modes of operation: Client, Access Point, Station + Access Point

Dual core microprocessor Tensilica Xtensa 32-bit LX6

Maximum current per pin is 12mA (it is recommended to use 6mA)

It has 36 GPIOs

GPIOs with PWM / I2C and SPI functions

Adjustable clock from 80MHz up to 240MHz

Operating voltage: 3.3 VDC

It has SRAM of 512KB

Features 448KB ROM

It has external flash memory of 32Mb (4 megabytes)

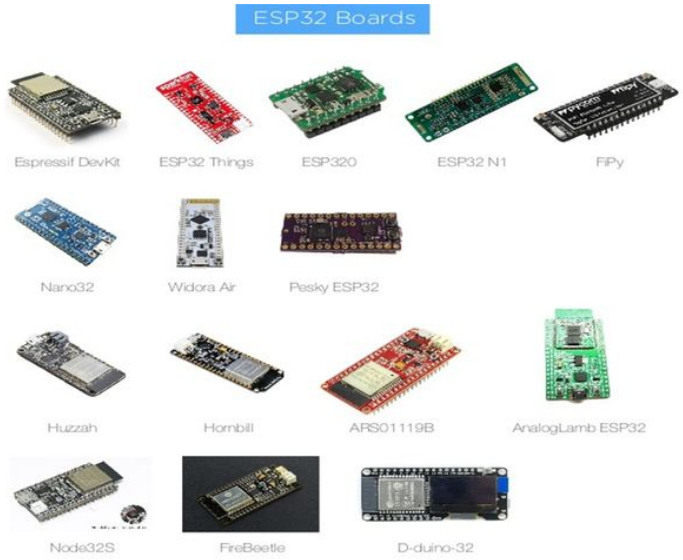
It has Bluetooth v4.2 BR / EDR and BLE (Bluetooth Low Energy)

## Step 2: Comparison Between ESP32, ESP8266 and 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 |

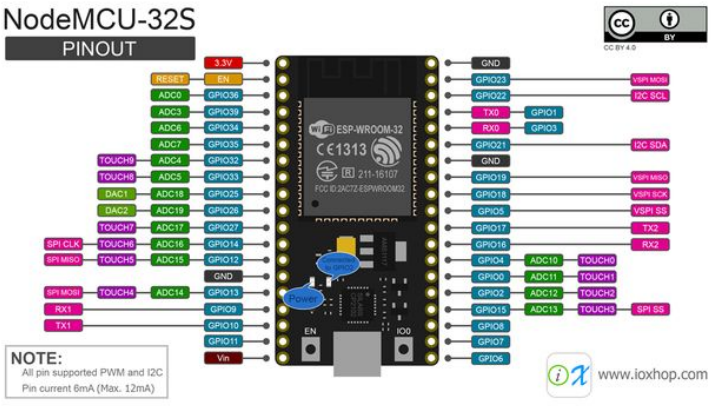
## Step 3: Types of ESP32

ESP32 was born with a lot of siblings. Today I'm using the first from the left, Espressif, but there are several brands and types, including Oled display built-in. However, the differences are all the same chip: the Tensilica LX6, 2 Core.




## Step 4: WiFi NodeMCU-32S ESP-WROOM-32

This is the diagram of ESP that we are using in our assembly. It is a chip that has lots of appeal and power. They are several pins you choose whether they want to work as digital analog, analog digital or even if that work the door as digital.



## Step 5: Configuring Arduino IDE (Windows)

Here's how to configure the Arduino IDE so we can compile for ESP32:

1. Download the files through the link: https://github.com/espressif/arduino-esp32

2. Unzip the file and copy the contents to the following path:

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

Note: If there is no directory "espressif" and "esp32", just create them normally.

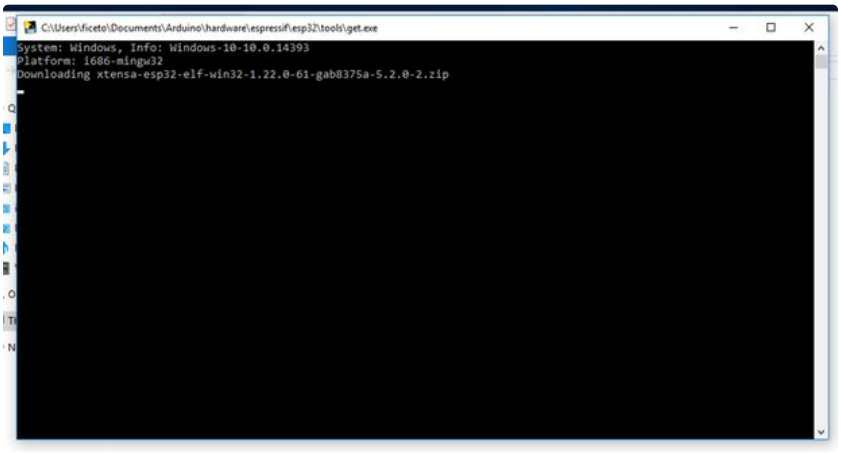
3. Open the directory

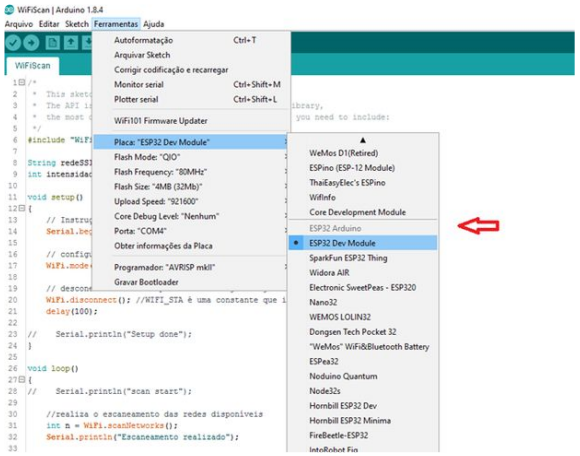
C: / Users / [YOUR_USER_NAME] / Documents / Arduino / hardware / espressif / esp32 / tools

Run the file "get.exe".

Ready, now just pick the ESP32 board in "tools >> board" and compile your code.

4. After the "get.exe" finishes, plug the ESP32, wait for the drivers to be installed (or install manually).




## Step 6: WiFi Scan

Here's an example of how to look for available WiFi networks near the ESP-32, as well as the signal strength of each of them. With each scan, we will also find out which network has the best signal strength.

## Step 7: Code

First let's include the library "WiFi.h", it will be necessary to allow us to work with the network card of our device.

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

Here are two variables that will be used to store the network's SSID (name) and signal strength.

```
String networkSSID = "";
 int strengthSignal = -9999;
```

---

## Step 8: Setup

In the setup () function, we will define the WiFi behavior mode of our device. In this case, since the goal is to search for available networks, we will configure our device to work as a "station".

```
void setup()
 {
   // Initialize Serial to log in Serial Monitor
   Serial.begin(115200);

       // configuring the mode of operation of WiFi as station
       WiFi.mode(WIFI_STA);//WIFI_STA is a constant indicating the station mode

       // disconnect from the access point if it is already connected
       WiFi.disconnect();
      delay(100);

//     Serial.println("Setup done");
 }
```

---

## Step 9: Loop

In the loop () function, we will search for the available networks and then print the log in the found networks. For each of these networks we will make the comparison to find the one with the highest signal strength.

```
void loop()
 {
//    Serial.println("scan start");

  // performs the scanning of available networks     int n = WiFi.scanNetworks();
    Serial.println("Scan performed");

     //check if you have found any network
     if (n == 0) {
        Serial.println("No network found");
    } else {
        networkSSID = "";
        strengthSignal= -9999;
        Serial.print(n);
        Serial.println(" networks found\n");
        for (int i = 0; i < n; ++i) {
       //print on serial monitor each of the networks found
          Serial.print("SSID: ");
          Serial.println(WiFi.SSID(i)); //network name (ssid)
          Serial.print("SIGNAL: ");
          Serial.print(WiFi.RSSI(i)); //signal strength
          Serial.print("\t\tCHANNEL: ");
          Serial.print((int)WiFi.channel(i));
          Serial.print("\t\tMAC: ");
          Serial.print(WiFi.BSSIDstr(i));
          Serial.println("\n\n");


          if(abs(WiFi.RSSI(i)) < abs(strengthSignal))
          {
             strengthSignal = WiFi.RSSI(i);
             networkSSID = WiFi.SSID(i);
             Serial.print("NETWORK WITH THE BEST SIGNAL FOUND: ( ");
             Serial.print(networkSSID);
             Serial.print(" ) - SIGNAL : ( ");
             Serial.print(strengthSignal );
             Serial.println(" )");
          }

          delay(10);
        }
    }
    Serial.println("\n---------------------------------------------------------------------------------\n");

 // interval of 5 seconds to perform a new scan
     delay(5000);
}
```

### "If (abs (WiFi.RSSI (i))"

Note that in the above statement we use abs (), this function takes the absolute value (ie not negative) of the number. In our case we did this to find the smallest of the values in the comparison, because the signal intensity is given as a negative number and the closer to zero the better the signal.

## Step 10: Files

Download all my files in: www.fernandok.com