|  |  |  |  |
| --- | --- | --- | --- |
|  | ESP8266 | ESP-01 | ESP32 |
| File | <http://arduino.esp8266.com/stable/package_esp8266com_index.json> | <http://arduino.esp8266.com/stable/package_esp8266com_index.json> | <https://dl.espressif.com/dl/package_esp32_index.json> |
| Agregue libreria | ESP8266 | ESP8266 | ESP32 |
| PLACA | NodeMCU 1.0 ESP-12E Module | Generic ESP Module | DOIT ESP32 DEVKIT V1 |
| Speed |  | 11520 |  |

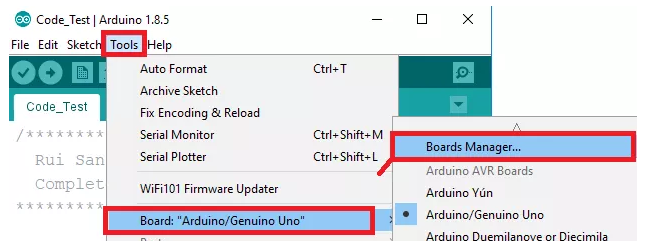
|  |  |
| --- | --- |
|  | Placa: NodeMCU 1.0 ESP-12E Module  az-delivery-devkit-v4 o FireBeetle-ESP32  En la parte plateada dice el modelo de microcontrolador  Es de tipo ESP8266 |

|  |  |
| --- | --- |
|  | En la parte de atras de placa  En la placa escoja :   * **ESP32 Dev Module / ESP32 DOIT DEVKIT V1** * **ESP32 Dev Module /** NodeMCU 1.0 ESP-12E Module |

|  |  |
| --- | --- |
| **DELANTE** | **ATRAS** |
| DOIT  MODEL ESP8266MOD | 9600 BAUD  DRIVER CH3406 |

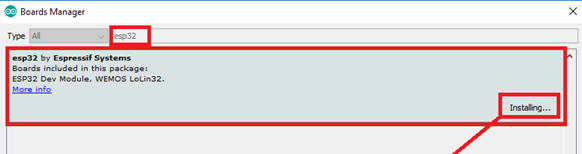
|  |  |
| --- | --- |
| File / Preferences |  |

**https://dl.espressif.com/dl/package\_esp32\_index.json, http://arduino.esp8266.com/stable/package\_esp8266com\_index.json**



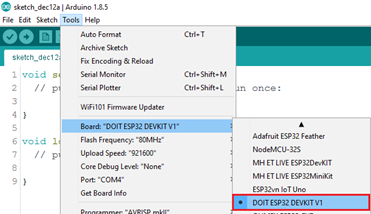
Search for **ESP32 / ESP8266**

and press install button for the “**ESP32 by Espressif Systems**“:



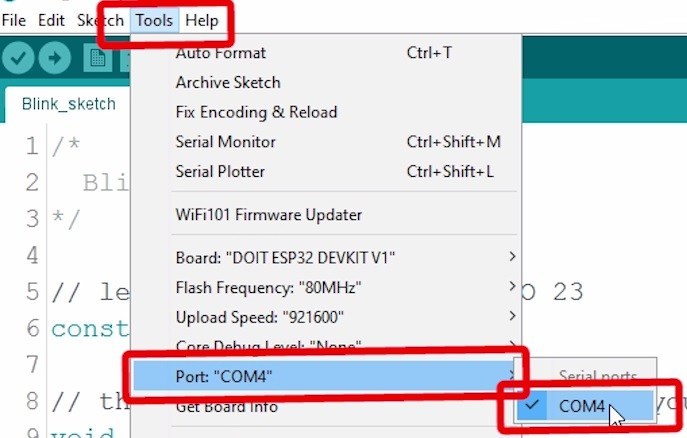
Select your Board in **Tools** > **Board** menu

* **ESP32 Dev Module / ESP32 DOIT DEVKIT V1**
* **ESP32 Dev Module /** NodeMCU 1.0 ESP-12E Module

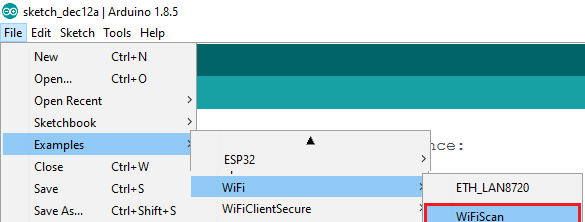


Select the Port (if you don’t see the COM Port in your Arduino IDE, you need to install the [CP210x USB to UART Bridge VCP Drivers](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)):

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>



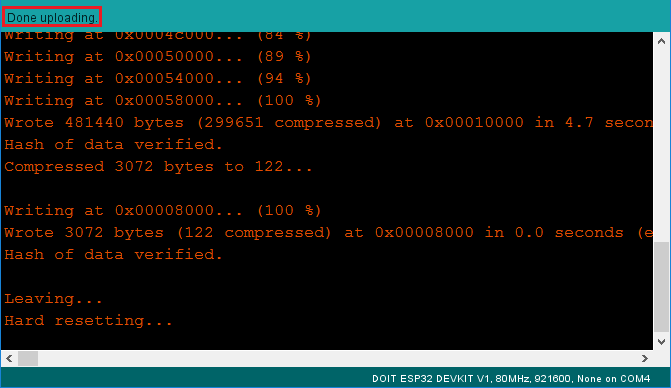
3. Open the following example under **File** > **Examples** > **WiFi (ESP32)** > **WiFiScan**



4. Press the **Upload** button in the Arduino IDE. Wait a few seconds while the code compiles and uploads to your board.

Arduino IDE upload WiFiScan sketch to ESP32

6. If everything went as expected, you should see a “**Done uploading.**” message.



7. Open the Arduino IDE Serial Monitor at a baud rate of 115200:

Open Arduino IDE Serial Monitor at baud rate 115200

|  |  |
| --- | --- |
| 8. Press the ESP32 on-board **Enable** button and you should see the networks available near your ESP32: | ESP32 Testing Add-on installation in Windows PC, Max OS X, and Linux computer |

## Troubleshooting

**Port not enabled**

En menu escoge Tools y el Puerto aparece en gris, no aperece COM1, COM2, o COM3,

**Solucion:**

1. Verifique en Control Panel / Devices que esta instalado el Driver
2. Instale el driver de link [CP210x USB to UART Bridge VCP Drivers](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers)):
3. DOWNLOAD ch340 driver generic

**Time Out**

If you try to upload a new sketch to your ESP32 and you get this error message

“A fatal error occurred: Failed to connect to ESP32: Timed out… Connecting…“.

It means that your ESP32 is not in flashing/uploading mode.

Having the right board name and COM por selected, follow these steps:

|  |  |
| --- | --- |
| Hold-down the “**BOOT**” button in your ESP32 board |  |
| Press the “**Upload**” button in the Arduino IDE to upload your sketch: | Arduino IDE uploading new sketch to ESP32 |
| After you see the  “**Connecting….**” message in your Arduino IDE, release the finger from the “**BOOT**” button |  |
| After that, you should see the “**Done uploading**” message.  Now press **Enable-Reset** Button to restart ESP32 |  |

You’ll also have to repeat that button sequence every time you want to upload a new sketch. But if you want to solve this issue once for all without the need to press the **BOOT**button, follow the suggestions in the next guide:

* [[SOLVED] Failed to connect to ESP32: Timed out waiting for packet header](https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/)

If you experience any problems or issues with your ESP32, take a look at our in-depth [ESP32 Troubleshooting Guide](https://randomnerdtutorials.com/esp32-troubleshooting-guide/).

## Wrapping Up

Take a look at the [ESP32 troubleshooting guide](https://randomnerdtutorials.com/esp32-troubleshooting-guide/).

* [**Learn ESP32 with Arduino IDE [eBook + Video Course]**](https://randomnerdtutorials.com/learn-esp32-with-arduino-ide/)
* [ESP32 vs ESP8266 – Pros and Cons](https://makeradvisor.com/esp32-vs-esp8266/)
* [Free ESP32 Projects and Tutorials](https://randomnerdtutorials.com/projects-esp32/)
* [Build an ESP32 Web Server with Arduino IDE](https://randomnerdtutorials.com/esp32-web-server-arduino-ide/)
* [ESP32 DHT11/DHT22 Web Server with Arduino ID](https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-web-server-arduino-ide/)

**My personal experience:**in 2021, I use almost exclusively the ESP32 for IoT projects. It is more versatile, and it comes with much more functionalities like Bluetooth, [different wake-up sources](https://randomnerdtutorials.com/esp32-deep-sleep-arduino-ide-wake-up-sources/), many peripherals,

Once you move to the ESP32, you won’t want to go back to the ESP8266.

* The ESP32 is faster than the ESP8266;
* The ESP32 comes with more GPIOs with multiple functions;
* The ESP32 supports analog measurements on 18 channels (analog-enabled pins) versus just one 10-bit ADC pin on the ESP8266;
* The ESP32 supports Bluetooth while the ESP8266 doesn’t;
* The ESP32 is dual-core, and the ESP8266 is single core;
* The ESP8266 is cheaper than the ESP32;
* The ESP8266 has a wider community (although we don’t think that at this point, the difference is that big);
* For many IoT and Wi-Fi projects, the ESP8266 can do the job for a lower price;
* Both boards support MicroPython firmware.