
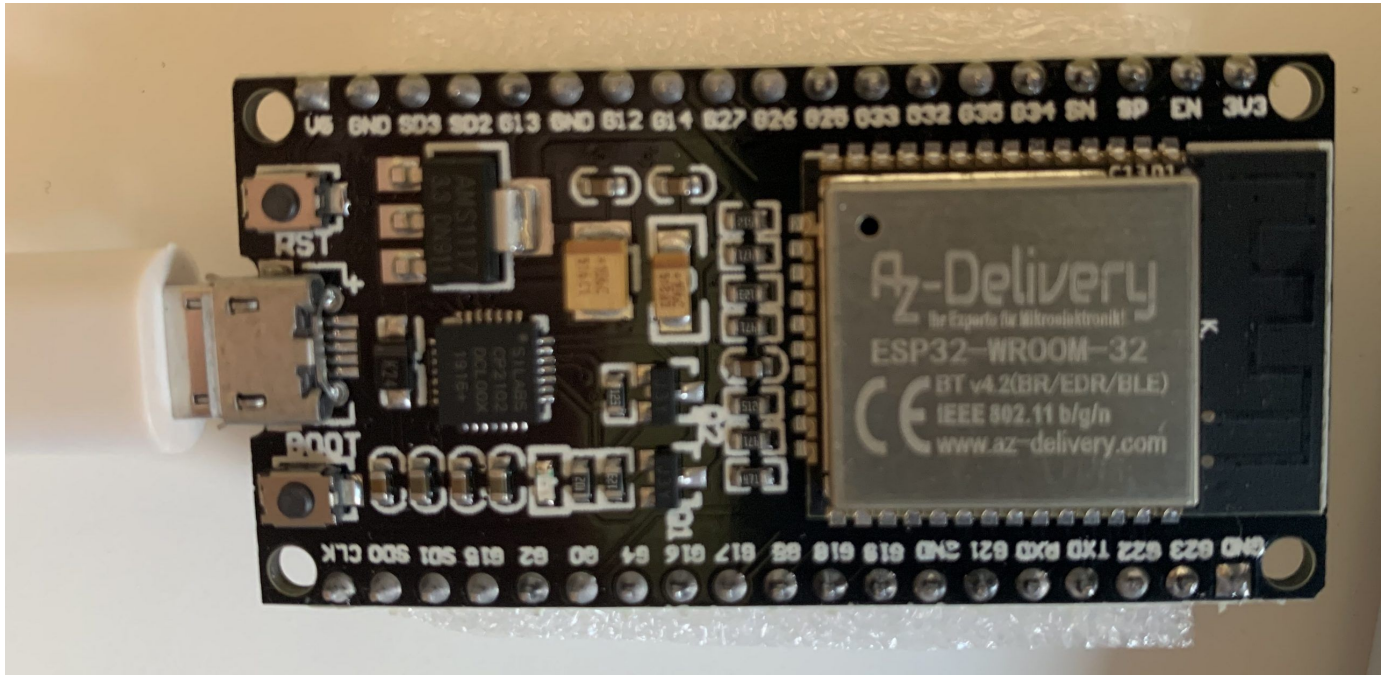




Arranque Del Microcontrolador ESP32

 Juanjo Beunza / 15 abril 2020



Compartir:

Acabo de recibir el microcontrolador ESP32 DevKitC V2 NodeMCU de AZ-Delivery. Es una placa de desarrollo de bajo coste (9€ en Amazon, transporte incluido, ha llegado en 24 horas, [link](#)), bajo voltaje (3,3V) y altas prestaciones:

- Wifi integrado
- Bluetooth dual integrado
- Microprocesador Tensilica Xtensa LX6 a 160 o 240 MHz

Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)



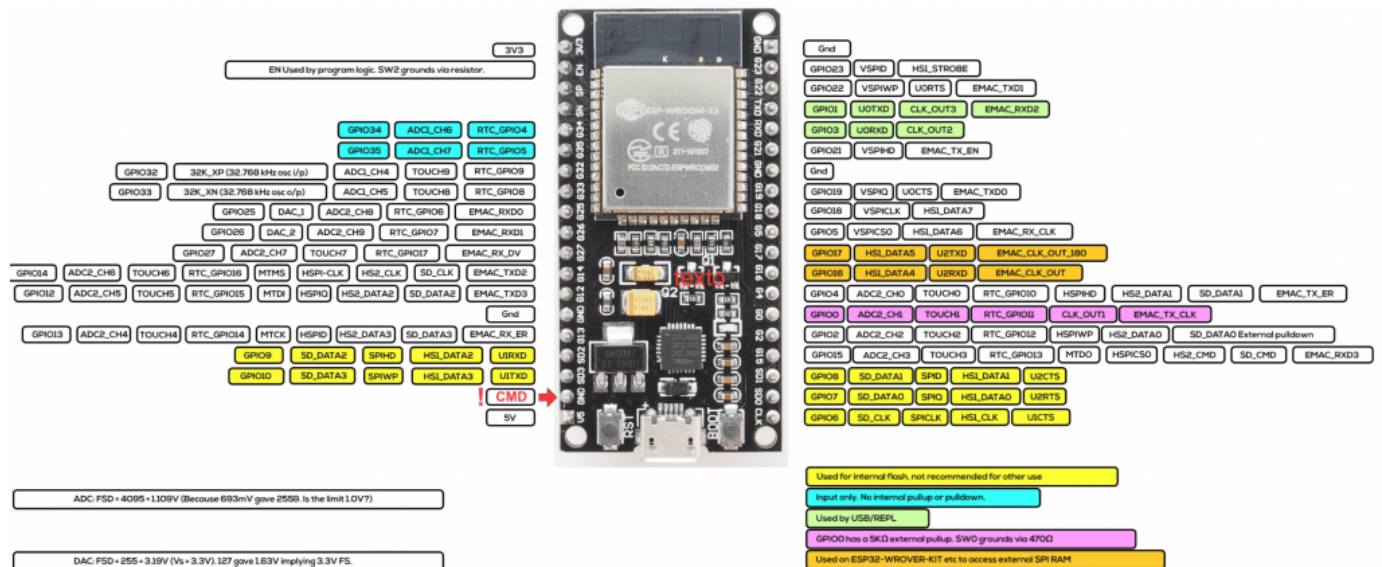
Frontal de la caja



Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)

Trasera de la caja

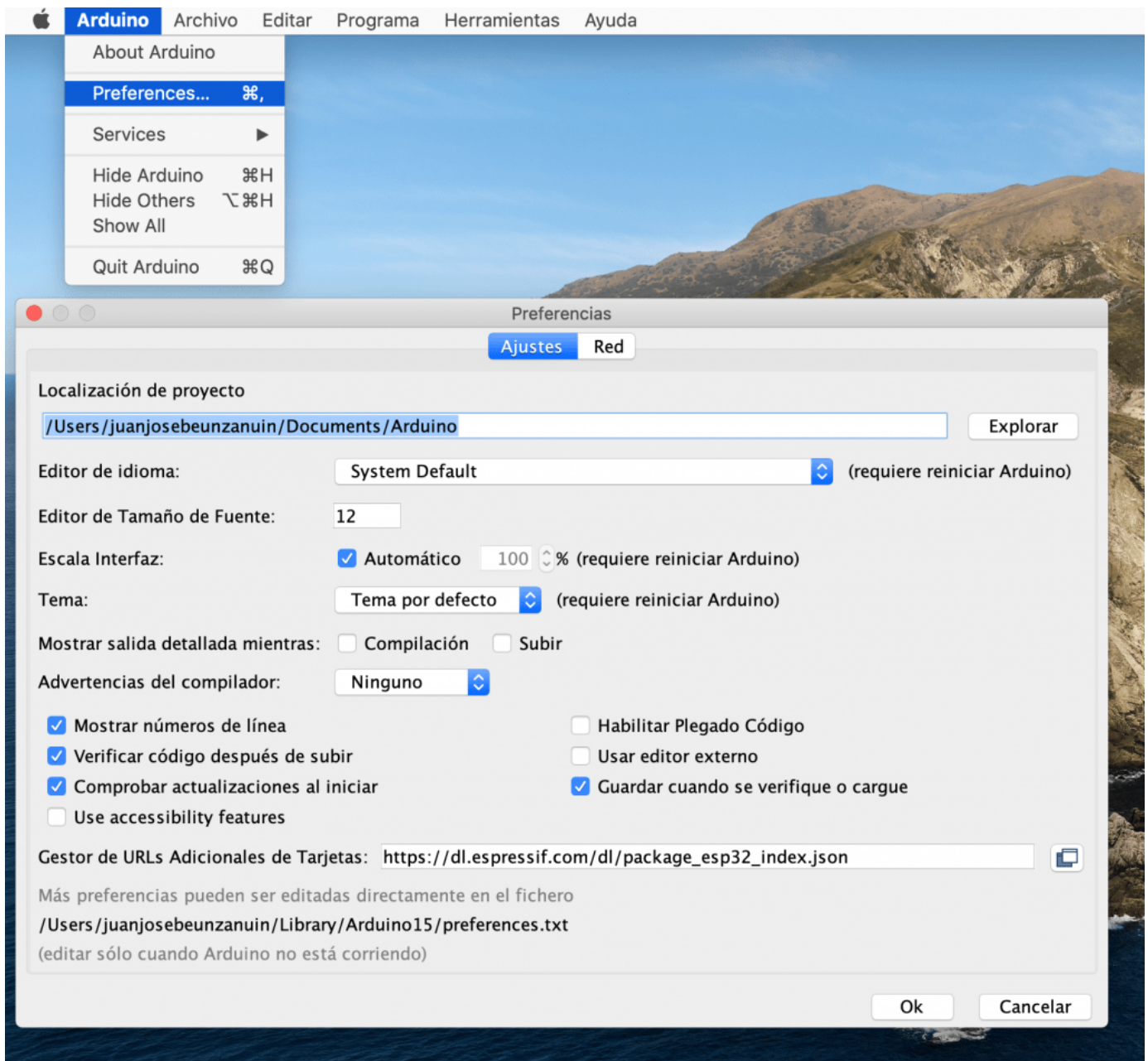
Está fabricado por Espressif Systems (China), fue lanzado en 2016 y es la evolución del ESP8266 al que sustituye. El fabricante final, en mi caso, fue AZ-Delivery de Alemania y viene con los pins ya soldados en la placa pero sin conector. Las etiquetas de identificación de los pines se leen mal (hecho en Alemania?), pero es fácil localizarlos con una foto guía de AZ-Delivery. Ojo con el pin CMD (en rojo) que es un GND (corregido en la placa, pero no en la guía).



I- ARRANQUE

Lo primero que hay que hacer es instalar los drivers de la ESP32 en el IDE de Arduino (si es esa tu elección). Para ello hay que añadir la siguiente dirección (https://dl.espressif.com/dl/package_esp32_index.json) a:

- Arduino / Preferencias / Gestor de URLs Adicionales de Tarjetas.

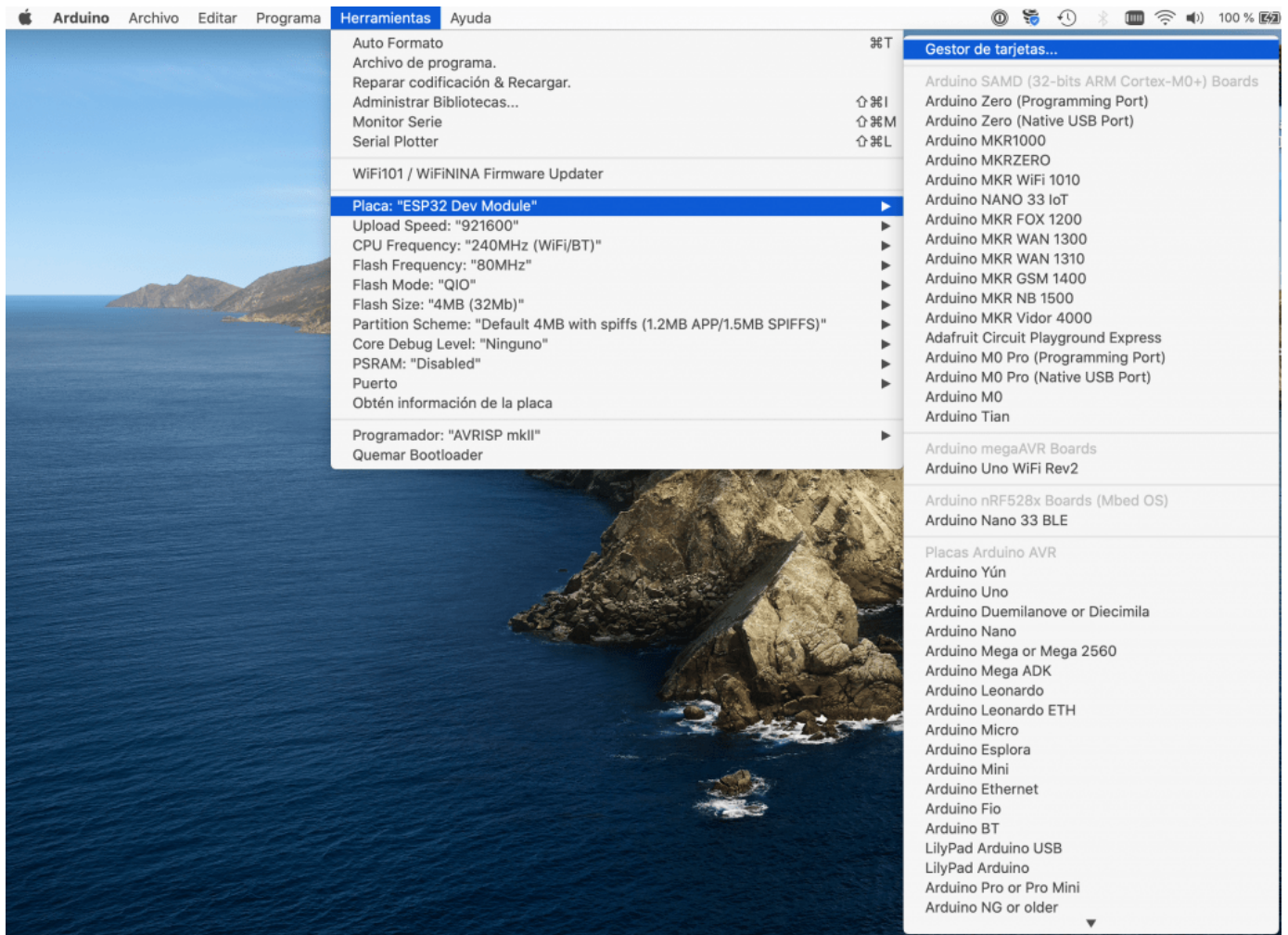


Abrir Preferencias del IDE de Arduino y pegar URL en Gestor de Tarjetas

Una vez añadida la URL para buscar el driver, acudir a:

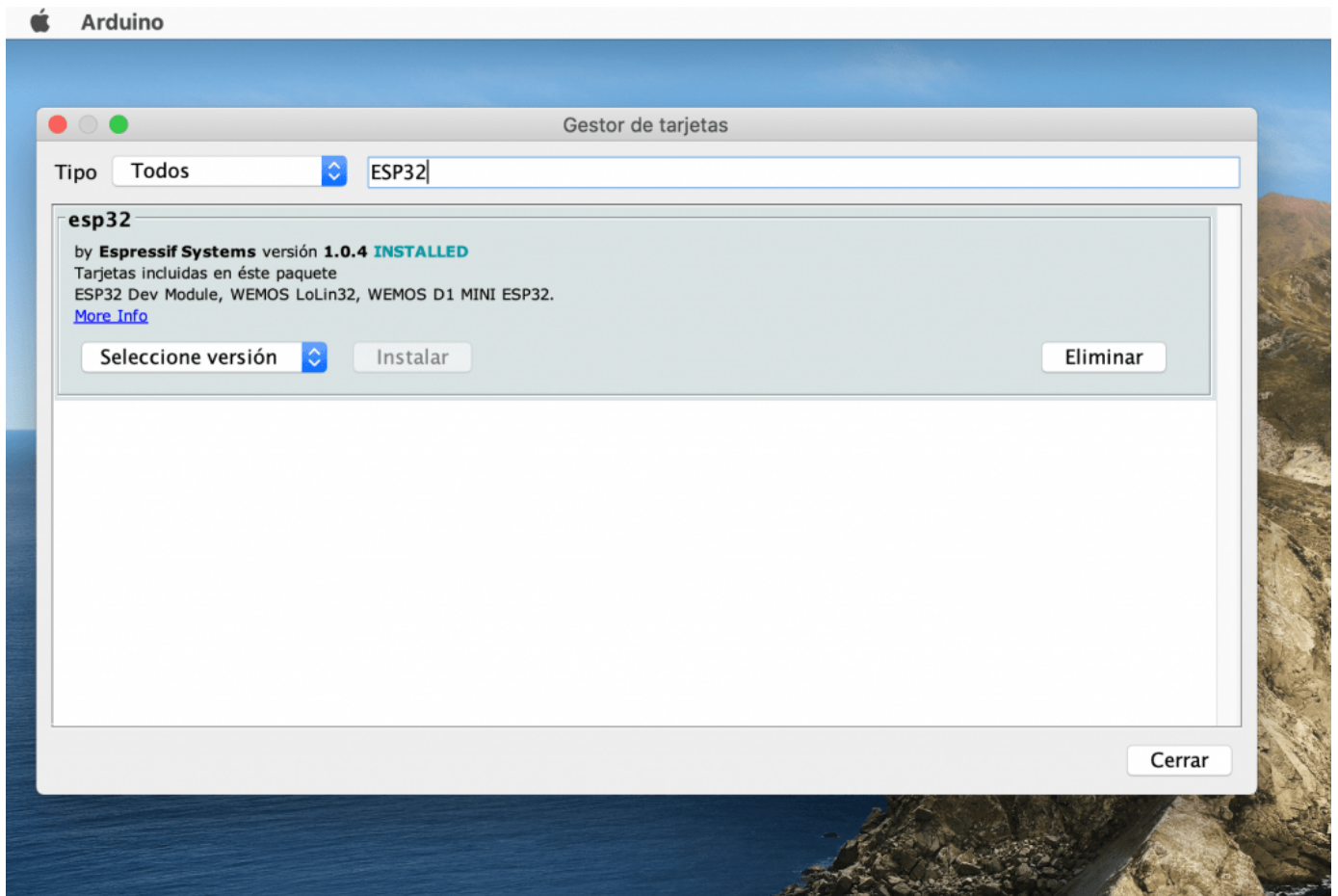
- Herramientas / Placa / Gestor de Tarjetas

Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)



Abrir Gestor de Tarjetas

Y escribir en la pestaña de búsqueda ESP32. Darle a buscar. Nos saldrá:



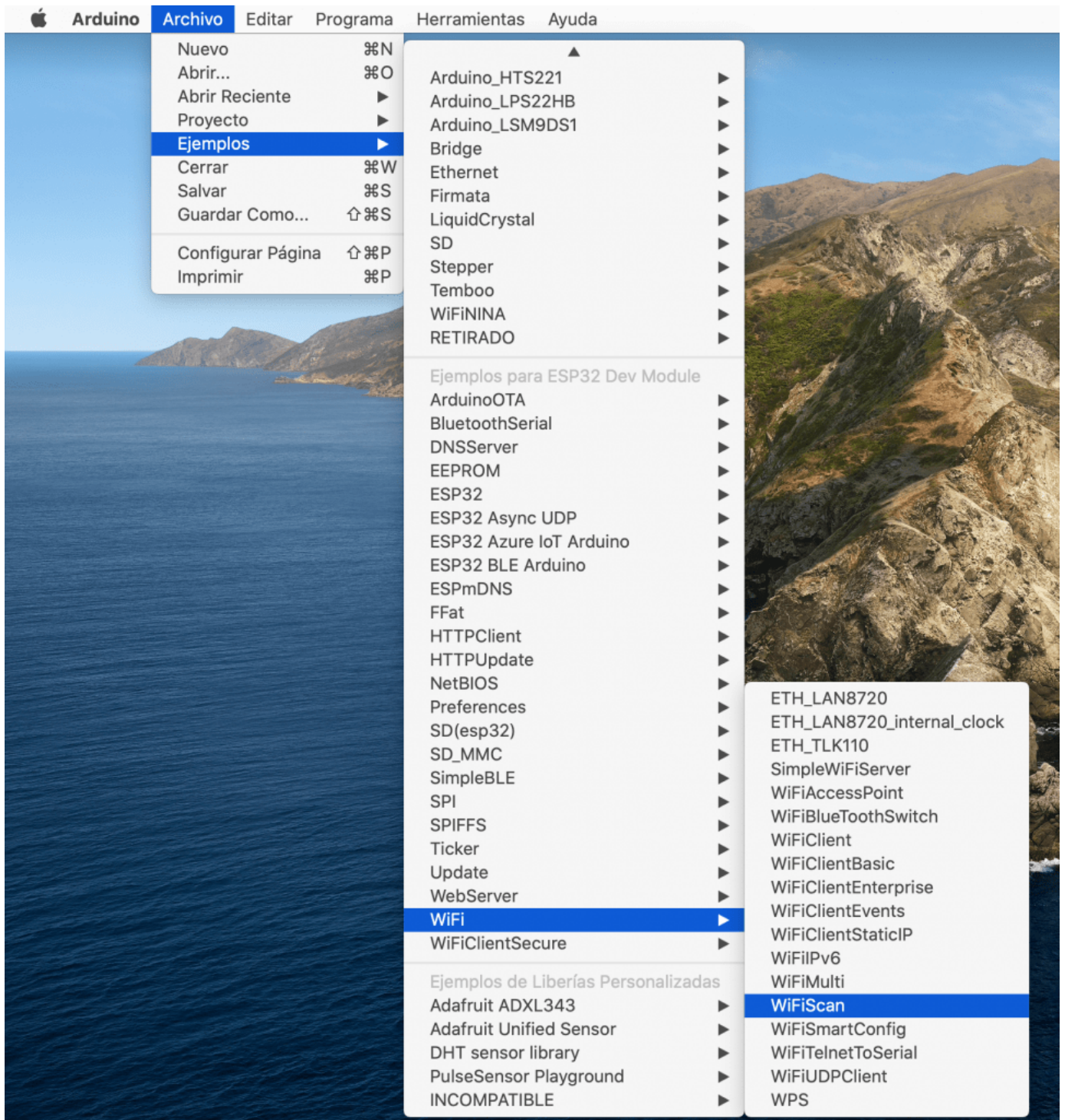
Paquete por Espressif Systems para ESP32

A mi ya me sale instalado, pero tú tendrás que instalarlo. A mi me costó 15 minutos bajar el archivo. No debería tardar más de unos segundos (en teoría), pero bueno. A mi me funcionó bien.

II- PROBAR LA WIFI

Tras conectar el ESP32 con un cable microUSB-USB al comp, arrancamos el ejemplo WifiScan

Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)



Abrimos el Ejemplo - WiFi - WiFiScan

Código de WiFiScan:

```
/*
```

Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)

```
*/  
  
include "WiFi.h"  
  
void setup()  
{  
  Serial.begin(115200);  
  
  // Set WiFi to station mode and disconnect from an AP if it was previously  
  connected  
  WiFi.mode(WIFI_STA);  
  WiFi.disconnect();  
  delay(100);  
  Serial.println("Setup done");  
}  
  
void loop()  
{  
  Serial.println("scan start");  
  
  // WiFi.scanNetworks will return the number of networks found  
  int n = WiFi.scanNetworks();  
  Serial.println("scan done");  
  if (n == 0) {  
    Serial.println("no networks found");  
  } else {  
    Serial.print(n);  
    Serial.println(" networks found");  
    for (int i = 0; i < n; ++i) {  
      // Print SSID and RSSI for each network found  
      Serial.print(i + 1);  
      Serial.print(": ");  
      Serial.print(WiFi.SSID(i));
```



```

Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)? " ":"*");

delay(10);

}

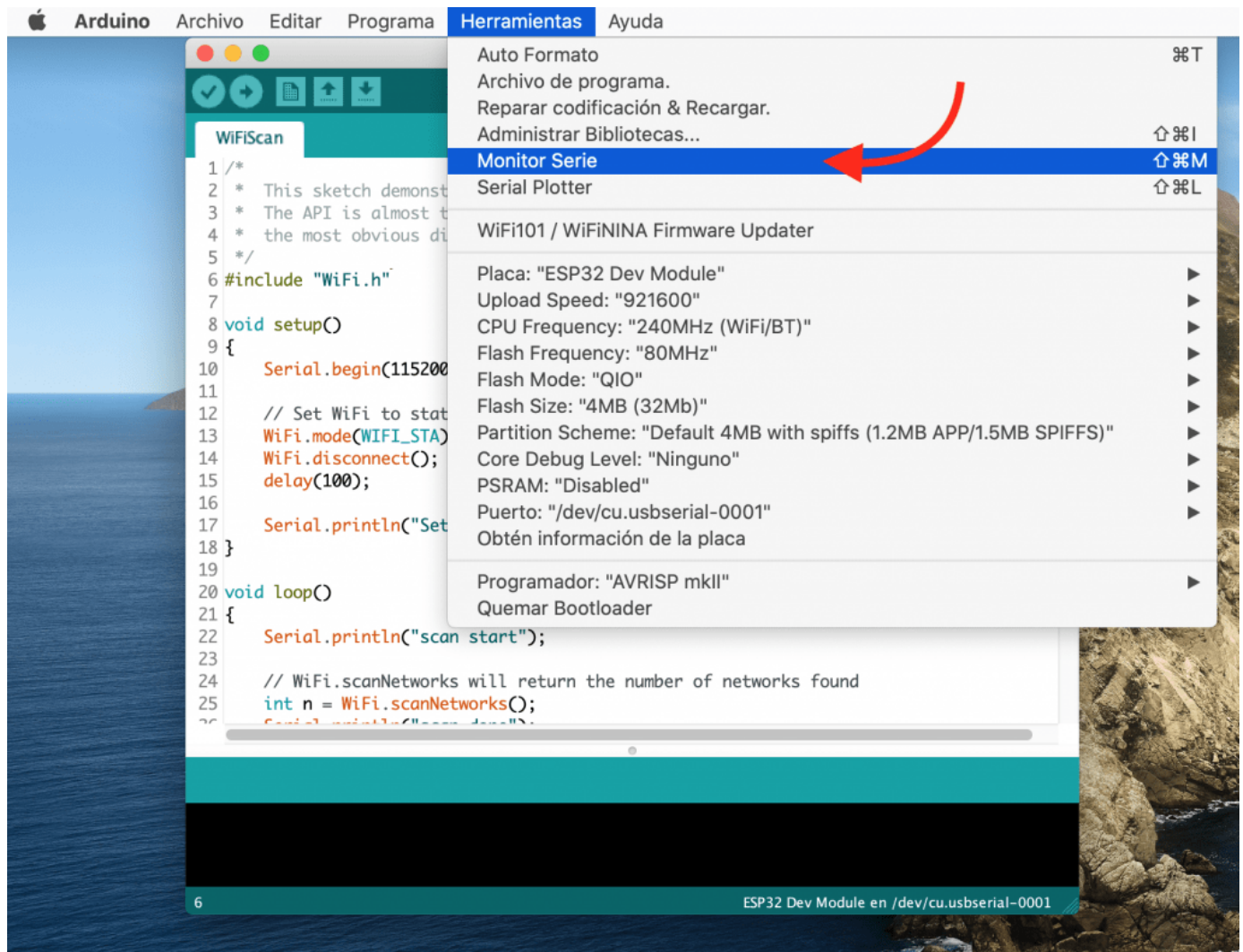
}

Serial.println("");
// Wait a bit before scanning again
delay(5000);
}

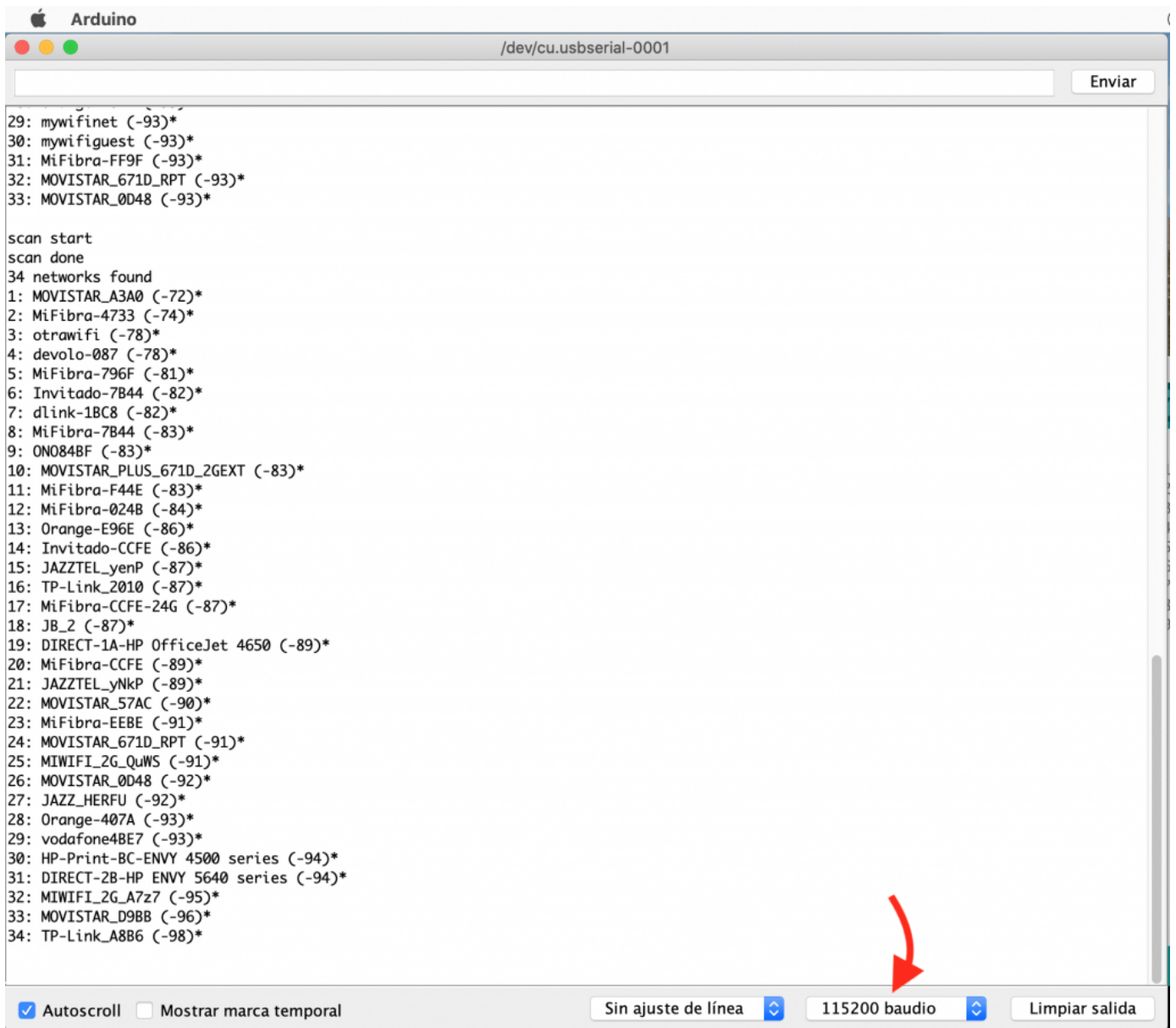
```

Seleccionamos la placa (ESP32 Dev Module) el puerto (usb), y compilamos + subimos el programa.

Abrimos el Monitor Serie del Arduino IDE:



Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)



Arduino
/dev/cu.usbserial-0001

Enviar

```
29: mywifinet (-93)*  
30: mywifiguest (-93)*  
31: MiFibra-FF9F (-93)*  
32: MOVISTAR_671D_RPT (-93)*  
33: MOVISTAR_0D48 (-93)*  
  
scan start  
scan done  
34 networks found  
1: MOVISTAR_A3A0 (-72)*  
2: MiFibra-4733 (-74)*  
3: otrawifi (-78)*  
4: devolo-087 (-78)*  
5: MiFibra-796F (-81)*  
6: Invitado-7B44 (-82)*  
7: dlink-1BC8 (-82)*  
8: MiFibra-7B44 (-83)*  
9: ON084BF (-83)*  
10: MOVISTAR_PLUS_671D_2GEXT (-83)*  
11: MiFibra-F44E (-83)*  
12: MiFibra-024B (-84)*  
13: Orange-E96E (-86)*  
14: Invitado-CCFE (-86)*  
15: JAZZTEL_yenP (-87)*  
16: TP-Link_2010 (-87)*  
17: MiFibra-CCFE-24G (-87)*  
18: JB_2 (-87)*  
19: DIRECT-1A-HP OfficeJet 4650 (-89)*  
20: MiFibra-CCFE (-89)*  
21: JAZZTEL_yNkP (-89)*  
22: MOVISTAR_57AC (-90)*  
23: MiFibra-EEBE (-91)*  
24: MOVISTAR_671D_RPT (-91)*  
25: MIWIFI_2G_QuWS (-91)*  
26: MOVISTAR_0D48 (-92)*  
27: JAZZ_HERFU (-92)*  
28: Orange-407A (-93)*  
29: vodafone4BE7 (-93)*  
30: HP-Print-BC-ENVY 4500 series (-94)*  
31: DIRECT-2B-HP ENVY 5640 series (-94)*  
32: MIWIFI_2G_A7z7 (-95)*  
33: MOVISTAR_D9BB (-96)*  
34: TP-Link_A8B6 (-98)*
```

☒ Autoscroll ☐ Mostrar marca temporal Sin ajuste de línea 115200 baudio Limpiar salida

No te olvides de definir la frecuencia a 115200 baudios (lo especificado en el programa WiFiScan). De otro modo no verás nada en el Serial Monitor.

ESP32 WifiScan



III-

TROUBLESHOOTING Hay dos problemas frecuentes con los que te puedes encontrar: 1. Error: "Failed to connect to ESP32: Timed out waiting for packet header" al intentar subir código a la placa. Hay dos modos de resolverlo, uno temporal y transitorio, el otro permanente. Para el primero (transitorio), hay que presionar permanentemente el botón de "BOOT" en el microcontrolador antes y durante la subida del programa compilado. De no ser así, el ESP32 entra en un modo no receptivo. Hay que realizar este proceso cada vez que subes código a la placa, lo que puede llegar a ser un poco engorroso. La solución permanente pasa por soldar un capacitor electrolítico de 10uF entre el pin E y el GND ([link](#)), como explica Randomnerdtutorials (foto tomada de su web).



2. El segundo problema con que nos hemos encontrado es que a veces hay que presionar el botón de "RESET" de la placa tras subir el programa, para que empiece a ejecutarse. Parece ser que este problema es mas frecuente en versiones antiguas del IDE Arduino. Con la versión 1.8.12 no hemos vuelto a ver ese problema, aunque nos de el aviso la pantalla del IDE Arduino.

En resumen: la ESP32 es una placa barata pero muy potente, orientada a desarrolladores, que requiere un poquito más de trabajo en su arranque que una placa Arduino, pero sin grandes problemas. Daremos más información según la vayamos probando.

Contributors:



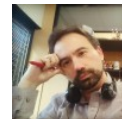
Juanjo Beunza
Administrator

✉
juanjo@juanjobeunza.com



Borja Rodríguez Vila
Editor

✉
borja.rodriguez@universidadeuropea.es



Enrique Puertas
Editor

✉
enrique.puertas@universidadeuropea.es

Compartir:

Etiquetas:

[Bluetooth](#)[ESP32](#)[IoT](#)[Wifi](#)

Entradas recientes

[Insertar datos de sensores en firestore desde pubsub \(Google Cloud\)](#)

[Conexion M5Stack ESP32 a Wifi](#)

Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)

Juanjo Beunza | Inteligencia Artificial, IoT y Salud

Este sitio web utiliza las cookies estrictamente necesarias para el funcionamiento de la web. Estas cookies no guardan ninguna información personal. [Cookies](#) [ACEPTO](#)