

Procedimiento de una Red Ad Hoc

Qué es Arduino

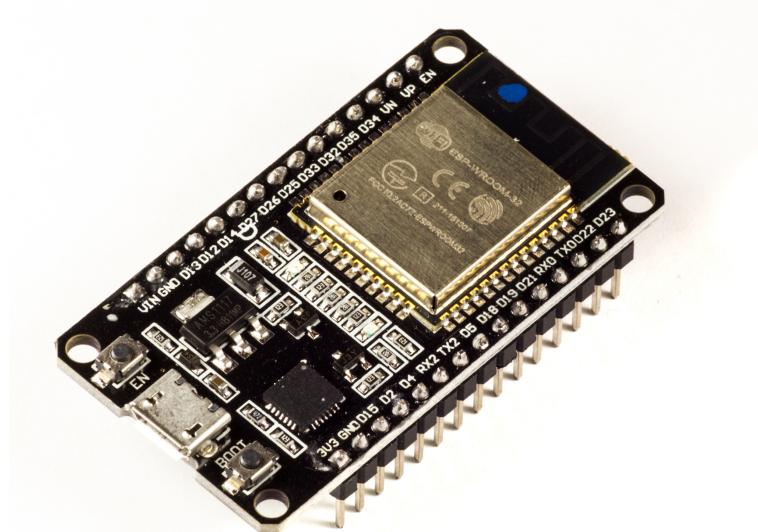
El Arduino es una plataforma de código abierto para la creación de proyectos electrónicos, esta se encuentra basada en un hardware y un software. El hardware cuenta con la capacidad de conectar periféricos en las entradas y salidas de un microcontrolador, mientras que el software puede trabajar con sistemas operativos de Windows, Mac OS y GNU/Linux libre, flexible y fácil de emplear. [1]



Arduino [1]

Qué es una ESP32

Este es un dispositivo similar al Arduino. La ESP32 es una placa fabricada por la empresa Espressif diseñado para aportar en la versatilidad, robustez y fiabilidad en un gran número de aplicaciones, está funciona como un microcontrolador de bajo costo y a la vez de bajo consumo. [2,3]



ESP32 [2]

Características de la ESP32:

- Microprocesador Tensilica de 240 MHz de doble núcleo con rendimiento de 600 DMIPS (Millones de Instrucciones Por Segundo de tipo Dhrystone) [3]
 - Tiene 520 KBy SRAM y una memoria flash de 16 MB[3]
 - Trabaja con una temperatura ambiente entre 2,2 a 3,6 V. El bajo consumo de energía es proporcionado por un procesador ULP (Ultra-Low-Power)[3]
 - El ESP32 consume 2,5 μ A [3]
 - Puede almacenar hasta 8 Kbytes de datos y programas en el SRAM, por lo que es posible acceder a temporizadores, interrupciones y periféricos. [3]
 - Tiene añadido el SoC (Sistem on a Chip) que incluye el componente WiFi/HT40 802 y la funcionalidad de Bluetooth [3]
 - Tiene una antena PCB integrada que se puede conectar a un componente externo a través de un conector IPEX [3]
 - Sensor Hall, botón táctil capacitivo un amplificador analógico de baja señal y un cristal de cuarzo de 32 kHz [3]

Qué es una ESP86

Instalación del Arduino IDE

1. Se tiene que ingresar al siguiente enlace <https://www.arduino.cc/en/software> y descargar una versión del software Arduino IDE que sea compatible con el sistema operativo que se esté utilizando. Para el presente proyecto se trabajará con la versión 2.1.1 para Windows, se recomienda utilizar siempre la última versión.



Arduino IDE 2.1.1

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

[SOURCE CODE](#)

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.14: "Mojave" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Al seleccionar una de las opciones de descarga aparecerá una página Web con la opción de contribuir monetariamente con los desarrolladores del Software, la donación es completamente opcional y se puede omitir presionando “JUST DOWNLOAD”

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **74,529,488** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other

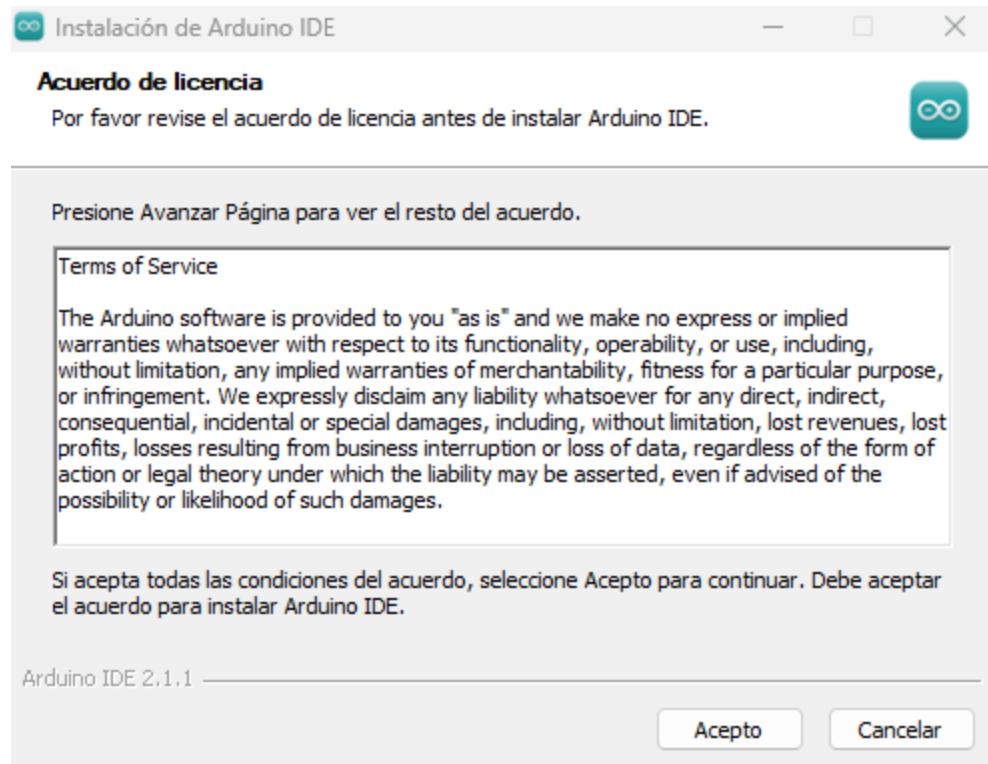
[JUST DOWNLOAD](#)

[CONTRIBUTE & DOWNLOAD](#)

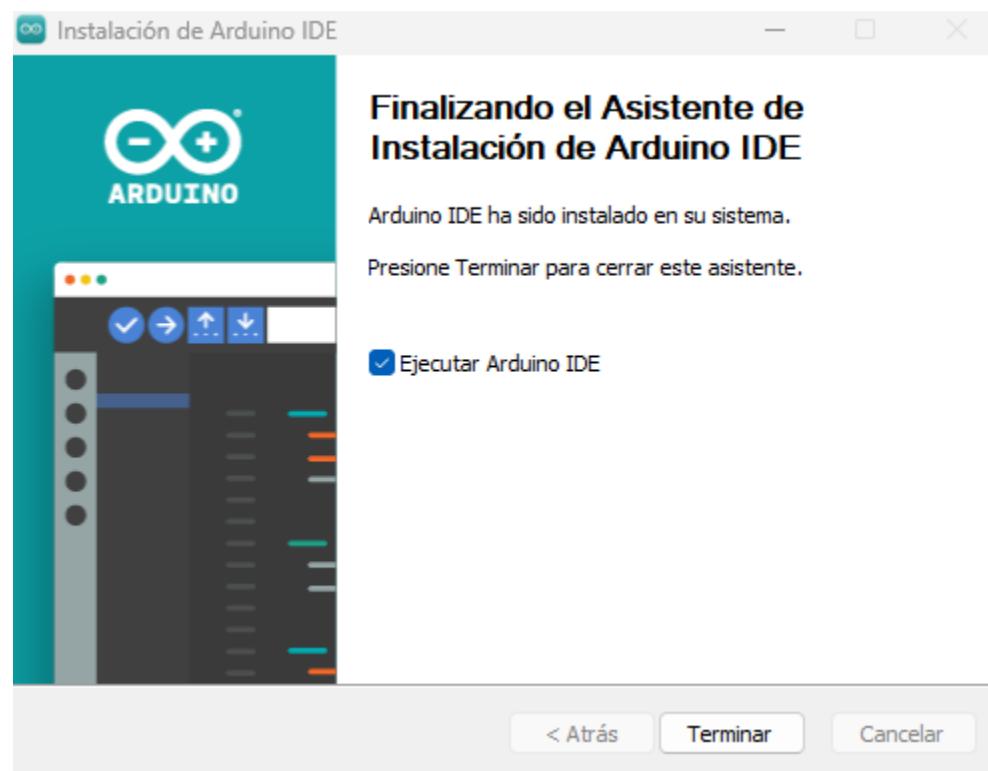


Learn more about [donating to Arduino](#).

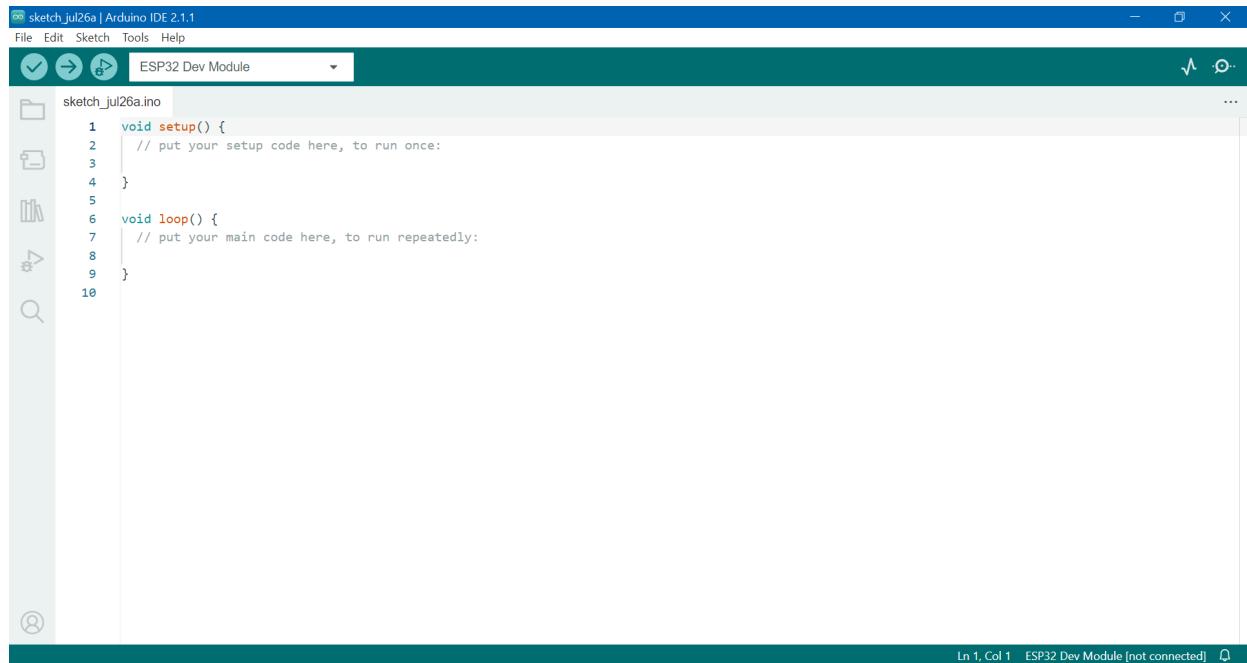
2. Al finalizar la descarga se debe presionar en el instalador y aceptar el acuerdo de licencia. Las demás configuraciones pueden ser a conveniencia del usuario.



Cuando se termine de configurar la instalación se mostrará la siguiente ventana, simbolizando que la instalación se procesó exitosamente:



Al ingresar al software se debería ver la siguiente ventana



The screenshot shows the Arduino IDE interface. The title bar reads "sketch_jul26a | Arduino IDE 2.1.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar has icons for file operations like Open, Save, and Run. A dropdown menu shows "ESP32 Dev Module". The left sidebar has icons for file, folder, and search. The main area displays the code for "sketch_jul26a.ino":

```
1 void setup() {
2     // put your setup code here, to run once:
3 }
4
5 void loop() {
6     // put your main code here, to run repeatedly:
7 }
8
9 }
10
```

In the status bar at the bottom right, it says "Ln 1, Col 1 ESP32 Dev Module [not connected]".

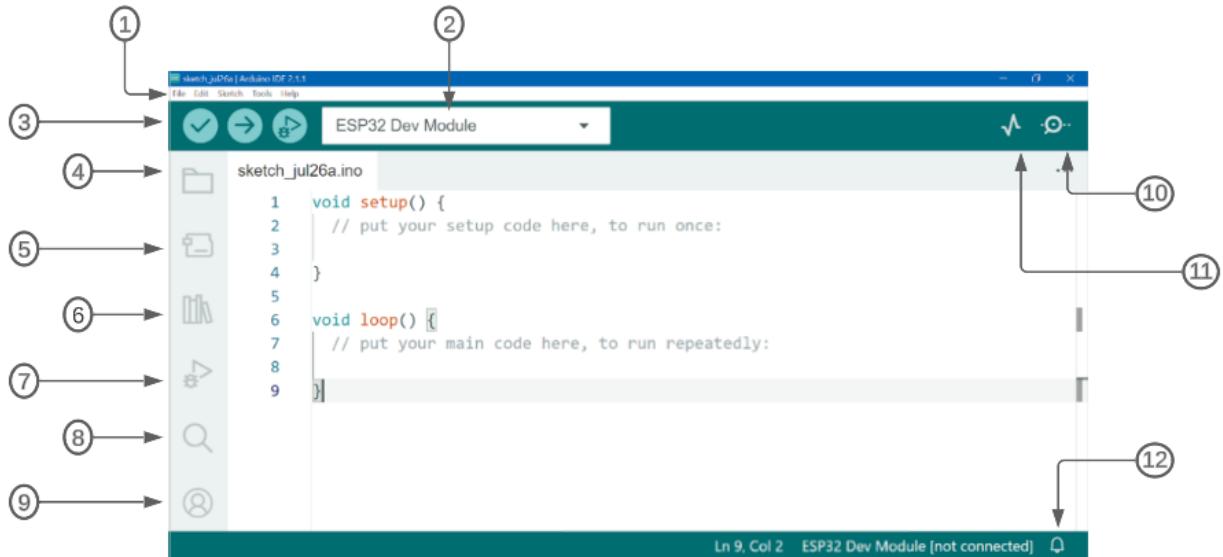
Plataforma del Arduino IDE

Arduino tiene su propia plataforma de desarrollo integrado (IDE), compuesto por un conjunto de herramientas de programación, empaquetado como un programa de aplicación. En otras palabras, este software cuenta con un compilador, un depurador, un editor de código, y un constructor de interfaz (GUI). [4]

Características del Arduino IDE

- Soporte de multiplataforma de Arduino [4]
- Detección automática de la placa conecta [4]
- Muestra memoria Flash y SRAM ocupada por un sketch o proyecto [4]
- Autoguardo al compilar y cargar sketch [4]
- Carga de sketch vía red (Wifi o Ethernet) para Arduino Yun [4]

Partes de Arduino IDE



1. **Barra de menú:** Área del software que indica las opciones de las herramientas de la aplicación en forma de menús desplegables, esta incluye (File, Edit, Sketch, Tools, Help)
2. **Selector del puerto y del microcontrolador:** Menú desplegable de los dispositivos que se encuentren instalados para programar junto con la selección del puerto al cual se encuentra conectado la placa del microcontrolador.
3. **Botones VAD:** Se puede encontrar los botones de Verificar, actualizar y debug.
 - a. **Verificar:** Botón que se encarga de hacer un procesamiento interno para verificar la sintaxis del programa.
 - b. **Actualizar:** Tiene la función de verificar la sintaxis y de enviar el código al microcontrolador.
 - c. **Debug:** Identifica y corrige errores en el código para que este pueda funcionar correctamente.
4. **Sketchbook:** Carpeta en donde se encuentran todos los proyectos y programas que se han creado con el software, cada proyecto tiene la nomenclatura de Sketch
5. **Board Manager:** Permite la instalación y administración de diferentes placas de microcontroladores.
6. **Library manager:** Permite buscar, instalar y administrar bibliotecas de código predefinidas.
7. **Debug:** Abre una sección para identificar un proceso y corregir errores que se presenten para que el programa pueda funcionar correctamente.

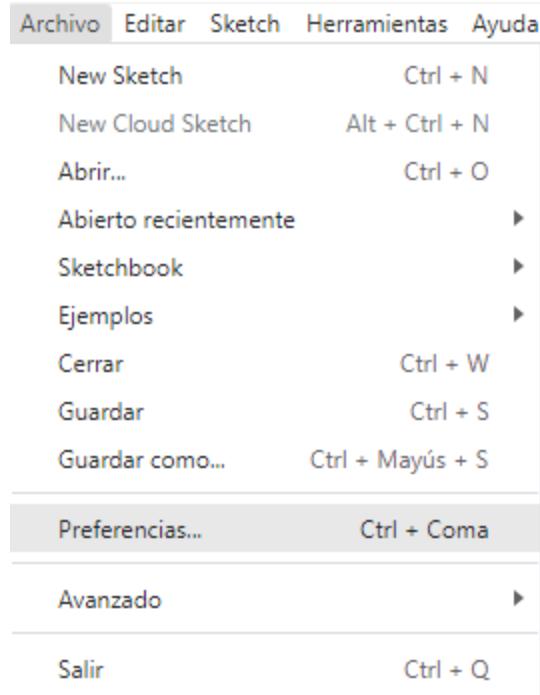
8. **Search:** Busca entre carpetas y archivos en los que se estén trabajando alguna
9. **Sign in:** Logeo para usuarios de Arduino IDE
10. **Serial monitor:** Herramienta de texto que muestra los datos en forma de texto en tiempo real
11. **Serial Plotter:** Herramienta que permite una visualización de datos enviados desde la placa del microcontrolador a través del puerto serie.
12. **Notification:** Alertas o mensajes que el sistema puede enviar para informar al usuario de eventos que se están presentando.

Monitor serié: Muestra los datos que son enviados y recibidos por el Arduino, es de notar que existen alternativas con el monitor serie, pero este suele ser más que suficiente

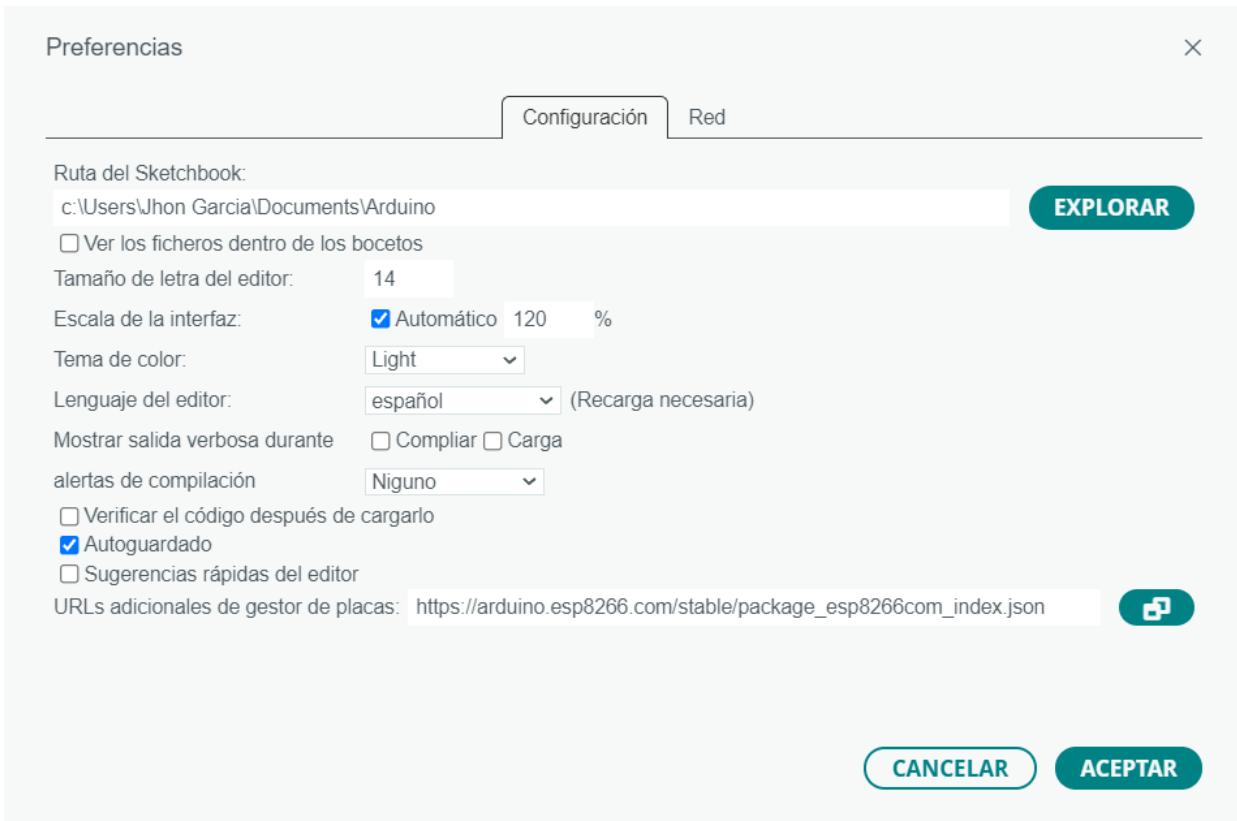
Instalación de la ESP32 en el Arduino IDE

Para empezar a utilizar la ESP32 en el entorno de Arduino es necesario instalar las dependencias que utiliza la placa del microcontrolador, para ello se necesitan seguir los siguientes pasos:

Ir a preferencias, el cual se encuentra en una de las opciones del menú desplegable del Archivo.



A continuación se abre la ventana de preferencias en el que se pondrá el siguiente texto para obtener las librerías de la ESP32 "https://dl.espressif.com/dl/package_esp32_index.json" en el apartado que dice "URLs adicionales de gestor de placas" [5]



Una vez hecho lo anterior se procede a instalar las librerías en el Arduino IDE, para hacerlo se necesita ir al **Board Manager**, aquí se busca la placa “esp32” y se instala “esp32 de Espressif Systems”. Para este proyecto se trabaja con la versión 2.0.11

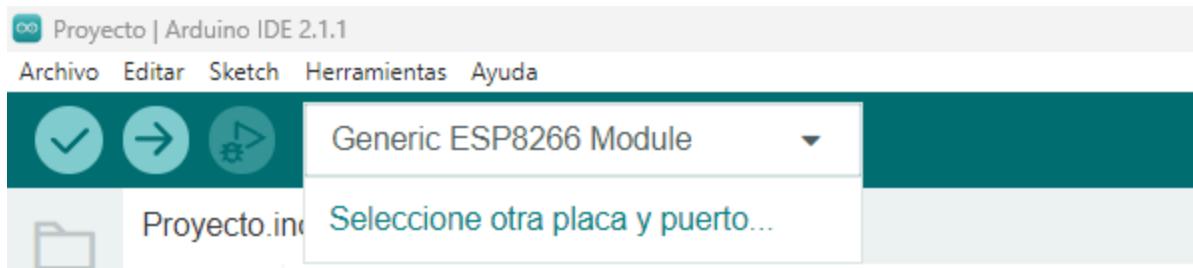


Al tener instalada la placa esp32 en el Arduino IDE, se procede a realizar el código. En esta sección del documento se enviará un archivo sin ningún código programado, puesto que solamente se quiere establecer una conexión entre el ordenador y el microcontrolador.

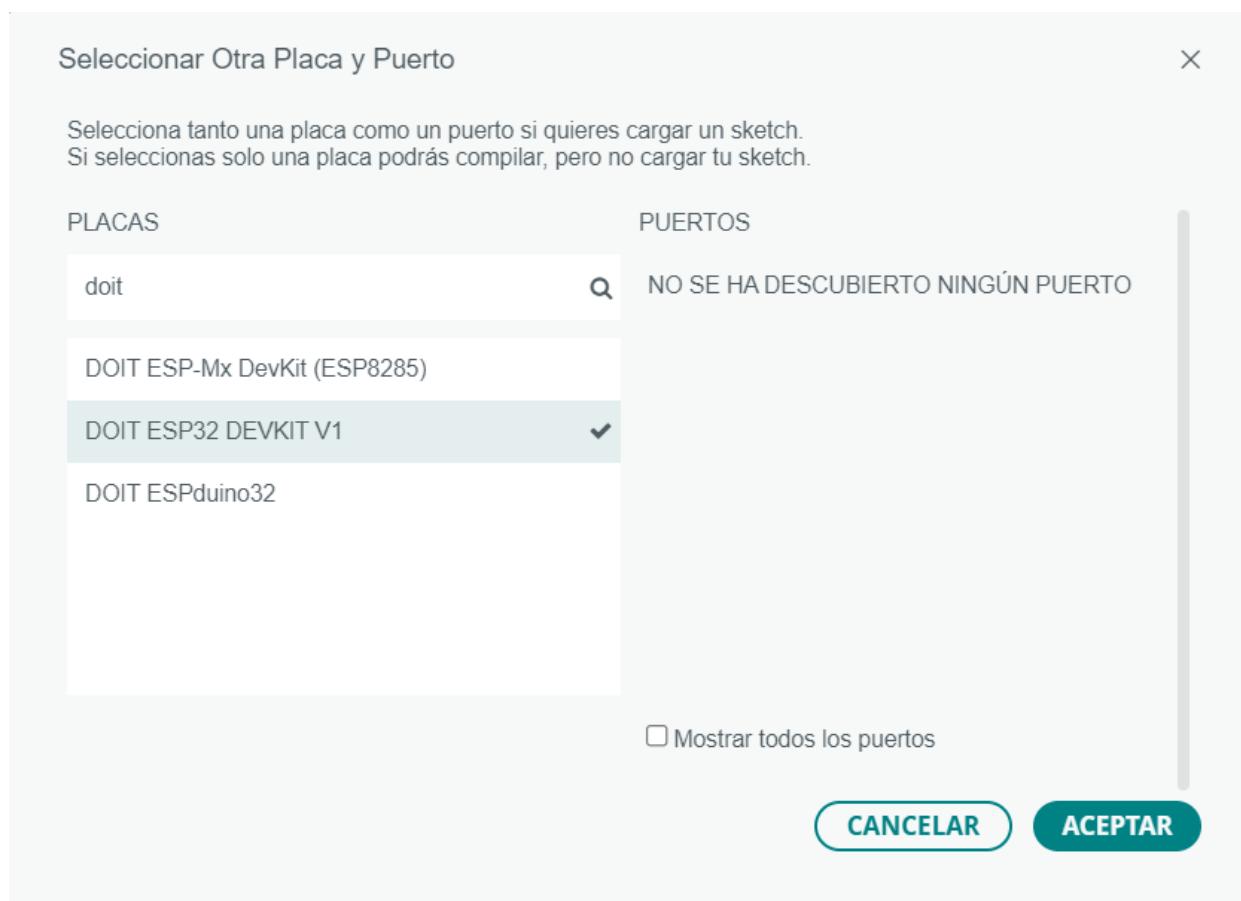
```
Proyecto.ino
```

```
1 void setup() {
2     // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7     // put your main code here
8 }
9
```

Para seleccionar la placa que se desea utilizar y el puerto con el cual se está conectando al computador, se abre el menú desplegable del **Selector del puerto y del microcontrolador**:

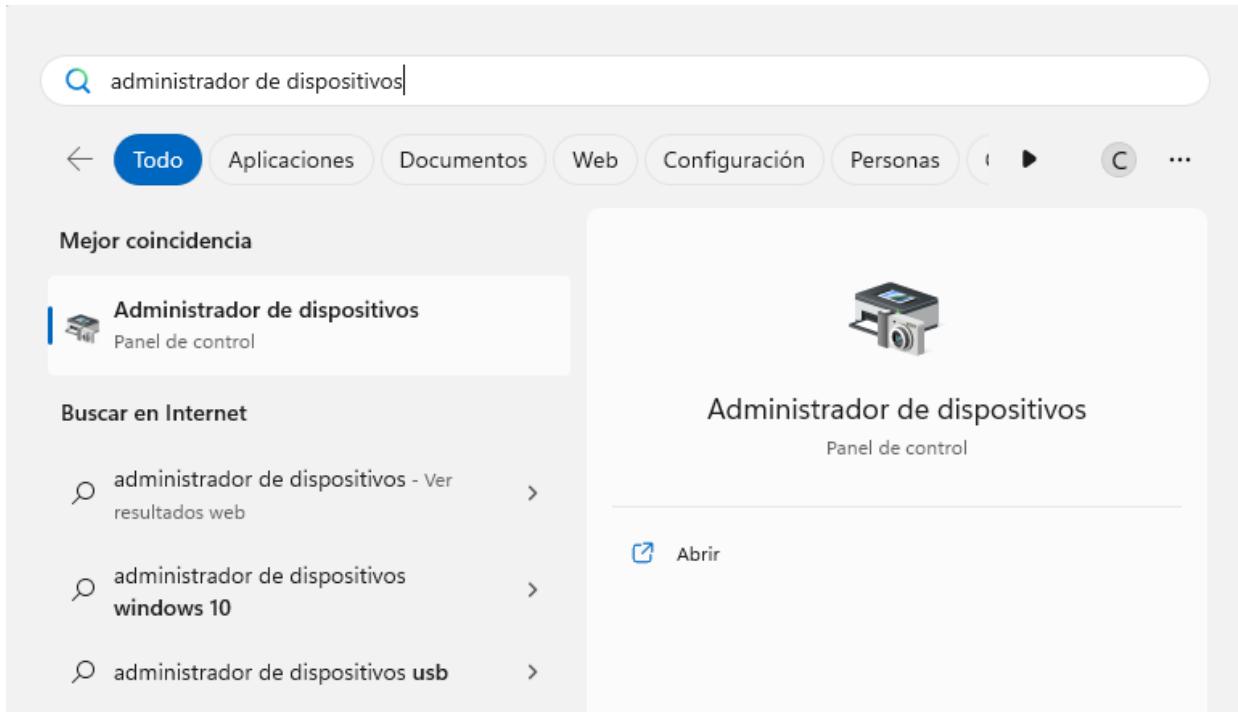


Se abre la siguiente ventana, esta se encuentra conformada por dos solicitudes de configuración: La placa y el Puerto. La placa con la que se trabajará en el proyecto se llama “DOIT ESP32 DEVKIT V1”. No obstante, no se puede visualizar que un puerto se encuentre habilitado a pesar de que la ESP32 se encuentre conectado. Esto puede ser debido a que el cable no es ideal para la transmisión de datos o que no se encuentre un Driver apropiado para este sistema embebido.

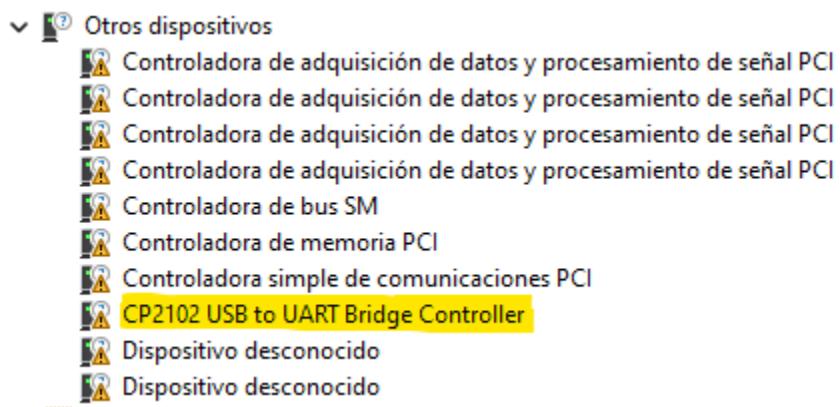


Problemas de Reconocimiento del puerto con Windows 10-11

Para verificar que el microcontrolador es reconocido por el computador, se verifica con la interfaz de **Administrador de dispositivos** de Windows [6]



Al tener la ESP32 conectada sin los drivers instalados se puede apreciar el equipo de la siguiente manera, se puede comprobar conectando y desconectando.



Para instalar los drivers se debe ir al siguiente enlace <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers> y descargar el adaptador de Windows, para este proyecto el que se descargó fue "CP210x Universal Windows Driver"

Download and Install VCP Drivers

Downloads for Windows, Macintosh, Linux and Android below.

*Note: The Linux 3.x.x and 4.x.x version of the driver is maintained in the current Linux 3.x.x and 4.x.x tree at www.kernel.org.

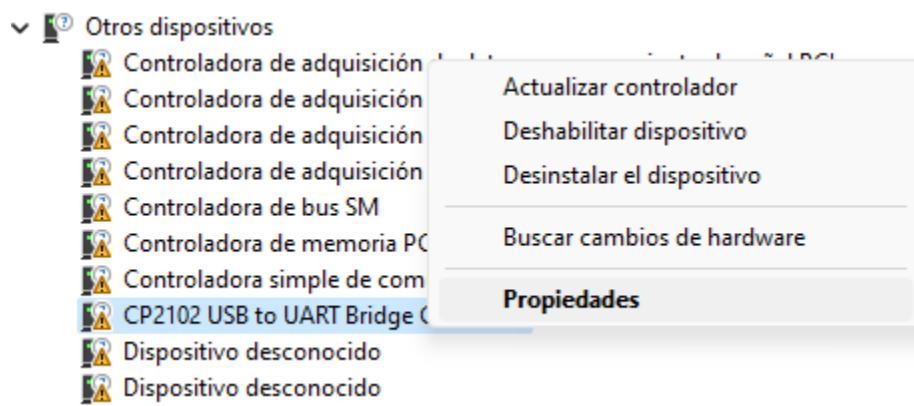
Software Downloads

[Software \(11\)](#)[Software - 11](#)

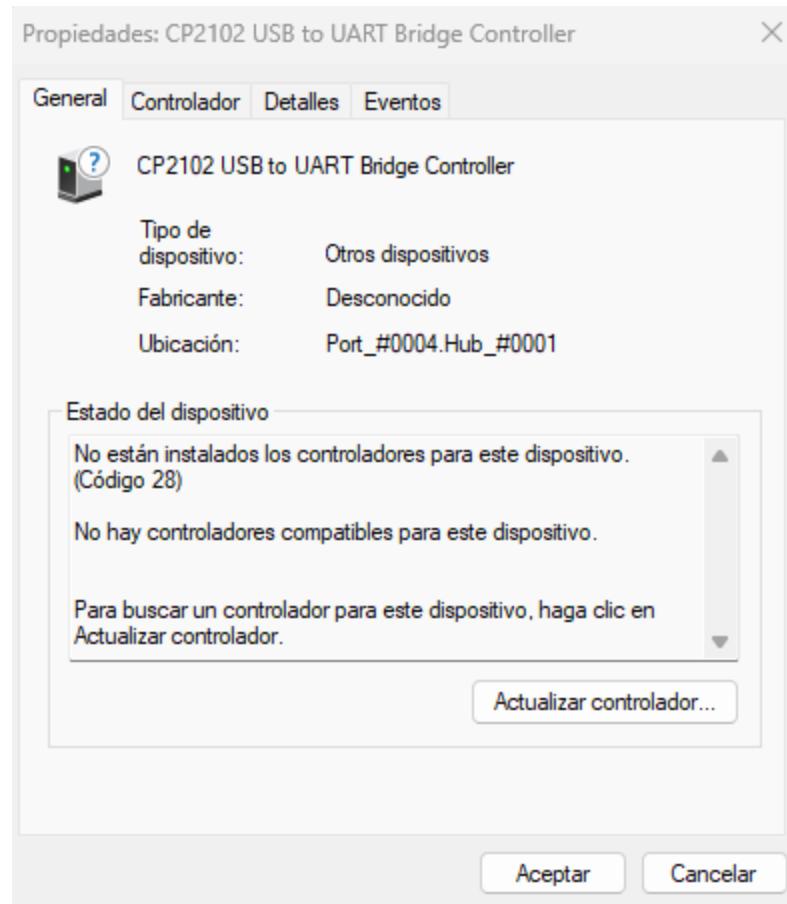
CP210x Universal Windows Driver	v11.3.0 6/24/2023
CP210x VCP Mac OSX Driver	v6.0.2 10/26/2021
CP210x VCP Windows	v6.7 9/3/2020
CP210x Windows Drivers	v6.7.6 9/3/2020
CP210x Windows Drivers with Serial Enumerator	v6.7.6 9/3/2020

[Show 6 more Software](#)

Para instalar el driver, se necesita descomprimir el archivo que se descargó; ir nuevamente al **Administrador de dispositivos** e ir a las propiedades del esp32



En la ventana emergente de las propiedades del ESP32 se encuentra una opción en la pestaña "General" llamada "Actualizar controlador", para configurar el driver se debe hacer click en este botón,



El sistema ofrece dos opciones para actualizar el controlador, el primero es “Buscar controladore automáticamente”, esta a pesar de ser una opción fiable, el sistema la utiliza por defecto al detectar cualquier equipo; por lo que si no se detecta de una forma instantanea es recomendable la segunda opción “Examinar mi PC en busca de controladores”. Esta es una opción completamente recomendable y la que se empleará en el presente proyecto.

X

←  Actualizar controladores - CP2102 USB to UART Bridge Controller

¿Cómo quieres buscar controladores?

→ Buscar controladores automáticamente

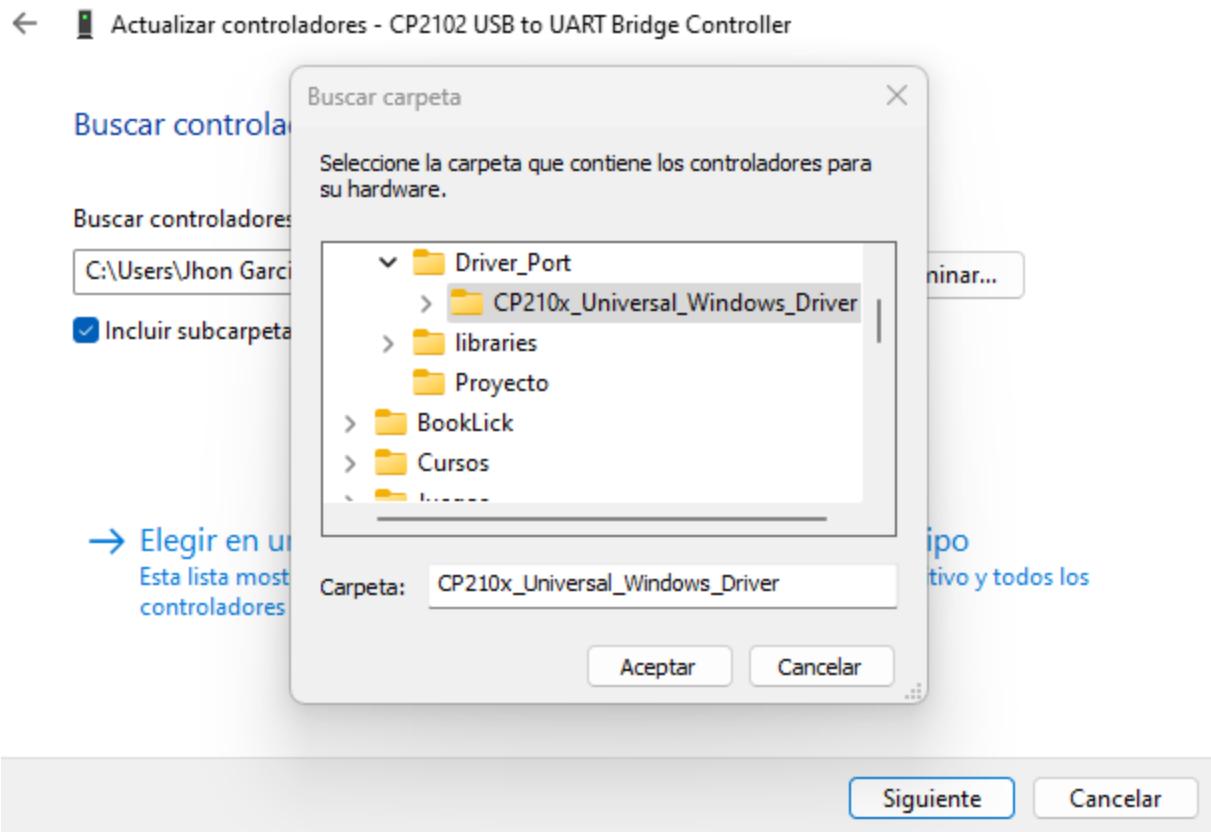
Windows buscará el mejor controlador disponible en el equipo y lo instalará en el dispositivo.

→ Examinar mi PC en busca de controladores

Busca e instala un controlador manualmente.

[Cancelar](#)

Al presionar “Examinar mi PC en busca de controladores”, se abrirá un explorador de archivos en el que se buscara el archivo descargado.



Una vez seleccionado el archivo se instalará en el equipo de manera que si se detecta nuevamente un microcontrolador “ESP32” el sistema lo reconocerá por defecto.

X

←  Actualizar controladores - Silicon Labs CP210x USB to UART Bridge (COM10)

Windows actualizó correctamente los controladores

Windows finalizó la instalación de los controladores para este dispositivo:



Silicon Labs CP210x USB to UART Bridge

Cerrar

Finalmente para comprobar que el driver se instalo correctamente es necesario volver a conectar nuevamente el “ESP32” y comprobar si la sección de Puertos (COM y LPT) se encuentra el equipo. Es de notar que si no se encuentra conectado ningún equipo la sección de Puertos COM y LPT no aparecerá.

-  Procesadores
-  Puertos (COM y LPT)
 -  Silicon Labs CP210x USB to UART Bridge (COM10)
-  Teclados

Secciones de Código

El código que se utilizo en el presente proyecto se pueden encontrar en el siguiente enlace url, este se encuentra dividio por las siguientes secciones:

- **Sección 1: Descripción general y bibliotecas utilizadas**

En esta sección, se describe el propósito del código y la utilización de la biblioteca "painlessMesh" para implementar una red ad hoc en dispositivos ESP32.

Descripción general:

En este código, se implementa una red ad hoc utilizando la biblioteca "painlessMesh" en dispositivos ESP32. La red de malla permitirá que los dispositivos se comuniquen entre sí de forma inalámbrica.

Bibliotecas utilizadas:

Se incluye la biblioteca "painlessMesh.h" para habilitar funciones y herramientas relacionadas con la configuración y manejo de redes de malla en el código.

Nota: si la librería que otorga Arduino no funciona descargue directamente de la página oficial de PainlessMesh.

- **Sección 2: Definición de parámetros de la red ad hoc**

En esta sección, se explican los parámetros utilizados para configurar la red ad hoc, como el nombre de la red, la contraseña y el puerto de comunicación.

Definición de parámetros de la red ad hoc:

- MESH_PREFIX: Se define el nombre de la red como "Red Ad Hoc".
- MESH_PASSWORD: Se asigna una contraseña para acceder a la red, en este caso, "clave1234".
- MESH_PORT: Se asigna el puerto de comunicación para la red de malla, que será el puerto 5555.

- **Sección 3: Variables globales y prototipos de funciones**

En esta sección, se explican las variables globales utilizadas en el código y se presenta el prototipo de la función "receivedCallback".

Variables globales:

- lastGreetingTime: Se declara una variable de tipo unsigned long llamada "lastGreetingTime" e inicializa con el valor 0. Esta variable se utiliza para controlar el envío periódico de mensajes de saludo en la red de malla.
- greetingInterval: Se define la constante "greetingInterval" con un valor de 5000 milisegundos, que representa el intervalo de tiempo entre los saludos.

Prototipo de función:

- receivedCallback: Esta función se encarga de manejar los mensajes recibidos en la red de malla. Toma como parámetros el identificador del nodo remitente ("from") y el contenido del mensaje ("msg").

- **Sección 4: Configuración inicial y loop principal**

En esta sección, se explica la configuración inicial del dispositivo ESP32 y el bucle principal "loop()", que se encarga de actualizar el estado de la red de malla y enviar mensajes de saludo a los nodos.

Configuración inicial (setup):

- Se inicializa la comunicación serial con una velocidad de 115200 baudios para la depuración.
- Se configura la salida de mensajes de depuración para mostrar errores, inicio y conexión de la red en el monitor serial.
- Se asigna manualmente el número de nodo al dispositivo ESP32.
- Se inicializa la red de malla con los parámetros definidos anteriormente y se configura la función "receivedCallback" para manejar los mensajes recibidos.

Bucle principal (loop):

- Se actualiza el estado de la red de malla en cada iteración del bucle.
- Se obtiene el tiempo actual en milisegundos.
- Si ha transcurrido el intervalo de saludo y la lista de nodos conocidos no está vacía, se envía un mensaje de saludo ("Hola desde el Nodo [número de nodo]") a todos los nodos de la red utilizando la función "sendBroadcast".
- Se actualiza el tiempo del último saludo enviado para el siguiente ciclo del bucle.

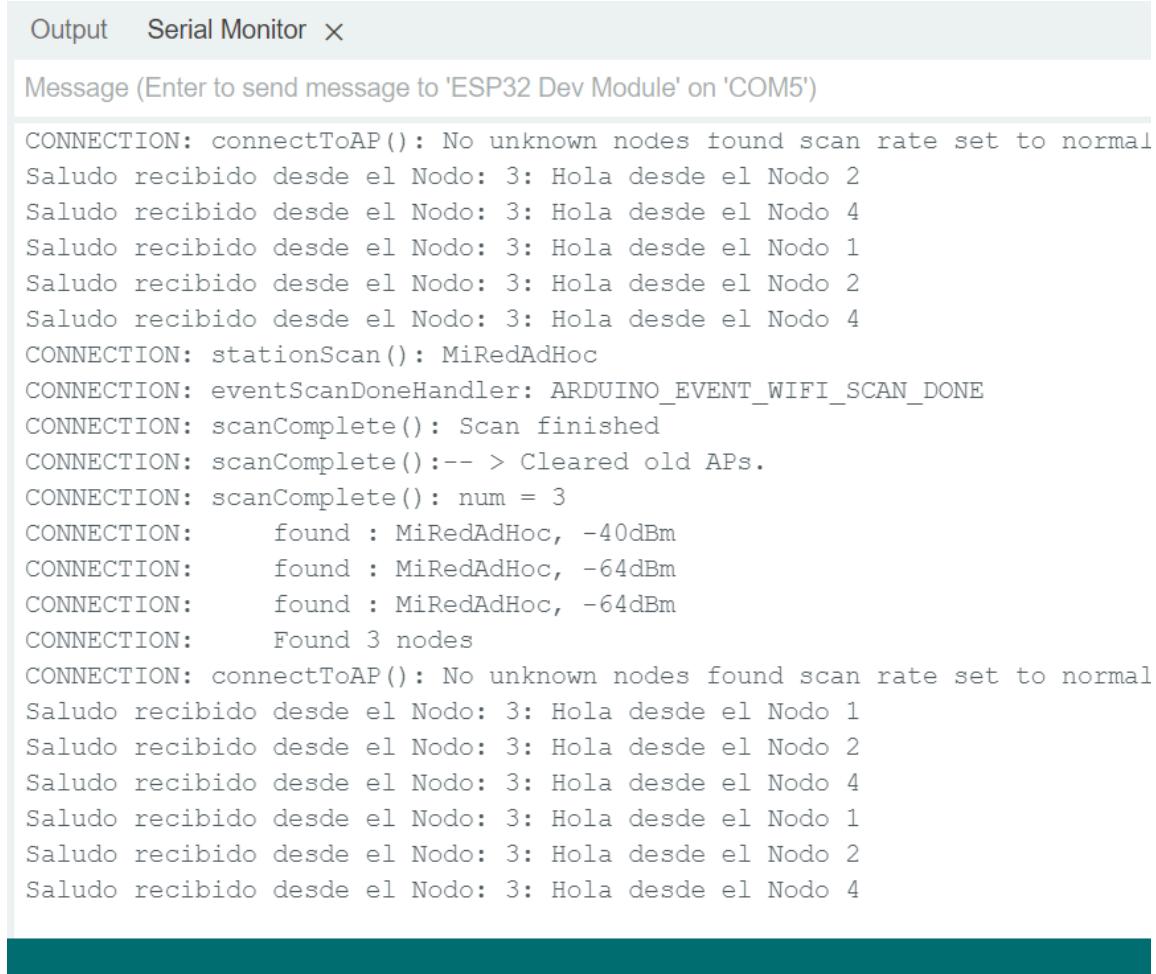
- **Sección 5: Función "receivedCallback"**

En esta sección, se explica el funcionamiento de la función "receivedCallback", que se encarga de mostrar los mensajes recibidos en el monitor serial.

Función receivedCallback:

- Cuando se recibe un mensaje en la red de malla, esta función se ejecuta automáticamente con los parámetros "from" (identificador del nodo remitente) y "msg" (contenido del mensaje).
- Muestra en el monitor serial el número de nodo del remitente y el contenido del mensaje recibido, lo que permitirá verificar la comunicación entre los nodos de la red.

Resultados



The screenshot shows the Arduino Serial Monitor interface. At the top, there are tabs for 'Output' and 'Serial Monitor' with an 'X' button. Below the tabs, the text 'Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')' is displayed. The main area contains the following text output:

```
MESSAGE: connectToAP(): No unknown nodes found scan rate set to normal
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 2
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 4
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 1
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 2
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 4
MESSAGE: stationScan(): MiRedAdHoc
MESSAGE: eventScanDoneHandler: ARDUINO_EVENT_WIFI_SCAN_DONE
MESSAGE: scanComplete(): Scan finished
MESSAGE: scanComplete():-- > Cleared old APs.
MESSAGE: scanComplete(): num = 3
MESSAGE:     found : MiRedAdHoc, -40dBm
MESSAGE:     found : MiRedAdHoc, -64dBm
MESSAGE:     found : MiRedAdHoc, -64dBm
MESSAGE:     Found 3 nodes
MESSAGE: connectToAP(): No unknown nodes found scan rate set to normal
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 1
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 2
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 4
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 1
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 2
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 4
```

En el monitor serial, se mostrarán los mensajes de saludo recibidos desde otros nodos de la red, junto con el número de nodo asignado manualmente a este dispositivo ESP32. El dispositivo enviará un mensaje de saludo a todos los nodos conocidos en un intervalo de tiempo determinado.

```

CONNECTION:      Found 3 nodes
CONNECTION: connectToAP(): No unknown nodes found scan rate set to normal
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 1
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 2
Saludo recibido desde el Nodo: 3: Hola desde el Nodo 4

```

Envío y creación de un archivo .txt.

Habilitar el Plugin:

Verificar que se tenga una versión inferior a la 2.0 de Arduino IDE debido a fallos en diferentes librerías, revisar que se haya instalado correctamente la plataforma ESP32 utilizando el Administrador de Tarjetas.

Este plugin debe ir en la parte en la que se guardan los proyectos de Arduino en una carpeta llamada tools.

C:\Users\je_\Documents\Arduino				
	Nombre	Fecha de modificación	Tipo	Tamaño
★ Acceso rápido	22072023_Chat_Prueba_Exitoso	22/07/2023 2:11 p. m.	Carpeta de archivos	
OneDrive - Personal	CHATSEMUESTRANTOELQUEENVIACOMOEL...	22/07/2023 5:41 p. m.	Carpeta de archivos	
Este equipo	enviando_mensajes_entre_nodos_con_numeros...	13/07/2023 4:14 p. m.	Carpeta de archivos	
↓ Descargas	ENviararchivos	13/08/2023 11:00 a. m.	Carpeta de archivos	
□ Documentos	Hola_mundo	15/08/2023 10:25 p. m.	Carpeta de archivos	
Escritorio	libraries	14/08/2023 12:32 a. m.	Carpeta de archivos	
Imágenes	PARAGUARDARENGLAESTALISTOYDOCUMEN...	19/07/2023 1:30 p. m.	Carpeta de archivos	
Música	ProyectoWS	6/07/2023 9:02 p. m.	Carpeta de archivos	
Objetos 3D	ProyectoWS1FDocumentado	6/07/2023 10:17 p. m.	Carpeta de archivos	
Videos	REDMESH	11/07/2023 7:25 p. m.	Carpeta de archivos	
Disco local (C:)	sketch_aug15a	17/08/2023 1:04 a. m.	Carpeta de archivos	
Red	sketch_aug16a	16/08/2023 11:57 p. m.	Carpeta de archivos	
	sketch_oct23a	7/07/2023 12:03 p. m.	Carpeta de archivos	
	ssd1306_128x64_i2c_0x3c_ejemplo_de_la_panta...	12/07/2023 11:05 p. m.	Carpeta de archivos	
	tools	14/08/2023 12:26 a. m.	Carpeta de archivos	8 KB
	Ultima_version_Ned_ad_hoc	14/07/2023 1:12 p. m.	Carpeta de archivos	
	WifiStap2p	7/07/2023 2:12 a. m.	Carpeta de archivos	
	WSconbotonesOFFyON	6/07/2023 11:31 p. m.	Carpeta de archivos	
	WSconbotonesOFFyONyEstado	6/07/2023 11:53 p. m.	Carpeta de archivos	
	toolszip	17/08/2023 1:19 p. m.	Carpeta comprimida ...	

El comando "ESP32 Sketch Data Upload" en el entorno de desarrollo Arduino IDE se utiliza para cargar archivos de datos en la memoria flash del módulo ESP32.

sketch.aug15a Arduino 1.8.19

Archivo Editar Programa Herramientas Ayuda

```

12
13
14 // Intervalo de
15 const unsigned i
16
17
18 // Prototipos de
19 void receivedCal
20
21
22 // Instancia de
23 painlessMesh mes
24
25
26 unsigned long p
27
28 void setup() {
29   Serial.begin(1
30
Writing at 0x000aac
Writing at 0x000b4c
Writing at 0x000b5c
Writing at 0x000bf0
Writing at 0x000c54
Writing at 0x000ca4
Wrote 786944 bytes
Hash of data verifi
31
Leaving...
Hard resetting via RTS pin...

```

Auto Formato Ctrl+T

Archivo de programa. Reparar codificación & Recargar. Administrar Bibliotecas... Monitor Serie Serial Plotter WiFi101 / WiFiNINA Firmware Updater

ESP32 Sketch Data Upload

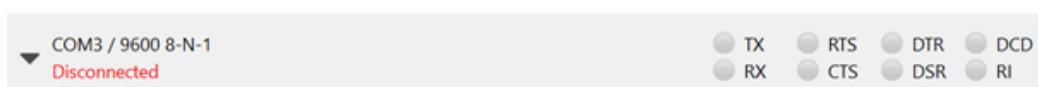
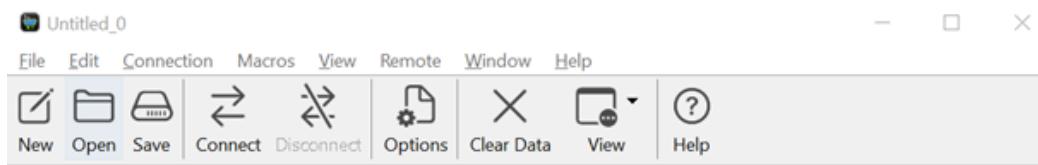
Placa: "ESP32 Dev Module" Upload Speed: "115200" CPU Frequency: "240MHz (WiFi/BT)" Flash Frequency: "80MHz" Flash Mode: "QIO" Flash Size: "4MB (32Mb)" Partition Schema: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)" Core Debug Level: "Ninguno" PSRAM: "Disabled" Arduino Runs On: "Core 1" Events Run On: "Core 1" Erase All Flash Before Sketch Upload: "Disabled" JTAG Adapter: "Disabled" Puerto: "COM3" Obtén información de la placa Programador Quemar Bootloader

bit/s)...

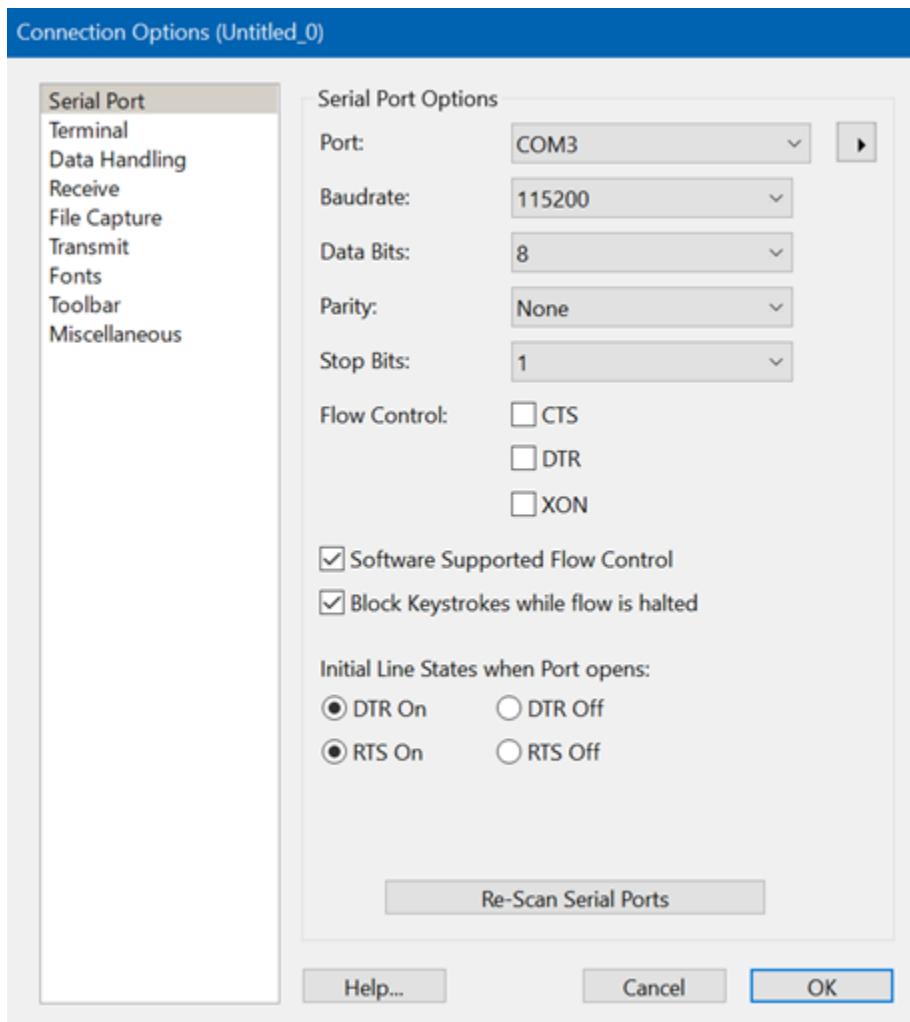
84

ESP32 Dev Module: Disabled. Disabled. Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 115200, Core 1, Core 1, None, Disabled en COM3

Para la obtención de los mensajes enviados en monitor abrimos CoolTerm:



Ingresar en la opción Connect.



Verificar que los datos sean iguales a los establecidos en Arduino.

Untitled_0 *

File Edit Connection Macros View Remote Window Help

New Open Save Connect Disconnect Options Clear Data View Help

ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)

configSip: 0, SPIWP:0xee

clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00

mode:DIO, clock div:1

load:0x3fff0030,len:1344

load:0x40078000,len:13964

load:0x40080400,len:3600

entry 0x400805f0

setLogLevel: ERROR |

Bienvenido al chat del Nodo 2

COM8 / 115200 8-N-1
Connected 00:00:06, 358 / 0 bytes

TX RX RTS CTS DTR DSR DCD RI

Se verifica la conexión del ESP32.

Untitled_0 *

File Edit Connection Macros View Remote Window Help

Disconnect Ctrl+K

Options... Ctrl+I

Reset Port

Send Serial Break Ctrl+B

Flush Serial Port Ctrl+Shift+F

Toggle RTS Ctrl+Shift+Alt+R

Toggle DTR Ctrl+Shift+Alt+D

Toggle Break State Ctrl+Shift+Alt+B

Mensaje Envi gAAAAABk3i

XgCPPTKa6

Send String... Ctrl+T

Send Text/Binary File... Ctrl+Shift+T

Llton2cvLk Capture to Text/Binary File Start... Ctrl+R

65XKMvraelQmIPSNhmlU2SuwxOH-XtVwBwp0-OhY46oly5EYlxo2RcsX ZC1DY1qNn4adoXAlWHIDTWAd1Ug5DjbfIErTwplWR2AM7H4rE p5n1d2am750mal.8dxz-0Tp2d3PLSubjx3htfORUaVzelom0Q..REQD- Z3cCtuUW0L_VHnxMhVn984tb3H2XbjkZHDf_mh1R2LmbjpdEKhk-pizwGP098-6BVOrnb8lgpbKshlx3kg1lyglp519 _bjdJ_fTeAs5pctvZtluewF0RDfVfe-5 _zvJH7HD005lg5nEcxF01gDxRULy9bsXP3HT6Ax34etar9p03E5hroSmazw0vOQWtEjsqVaYePt dacDnB3xjDIHp_HSqx2k- qLhs5AStrO-Y2znKXWGwlwqu2un7wClfuEUswM_2vRtibpYPqnQHR04sfIBxDZseee8Q5dSSK6ok- Jd0Fd5b4LR0gabxHbT90lm9Mc6fQ30UGlpGb04wPTPM0INL-58HPAVH1W5-mS6p226MLyC1zrUK_QwZRYvNjMk9q- eVU4JImtX0ynH1p Número de nodos conectados: 1 Número de nodos conectados: 1 Número de nodos conectados: 1

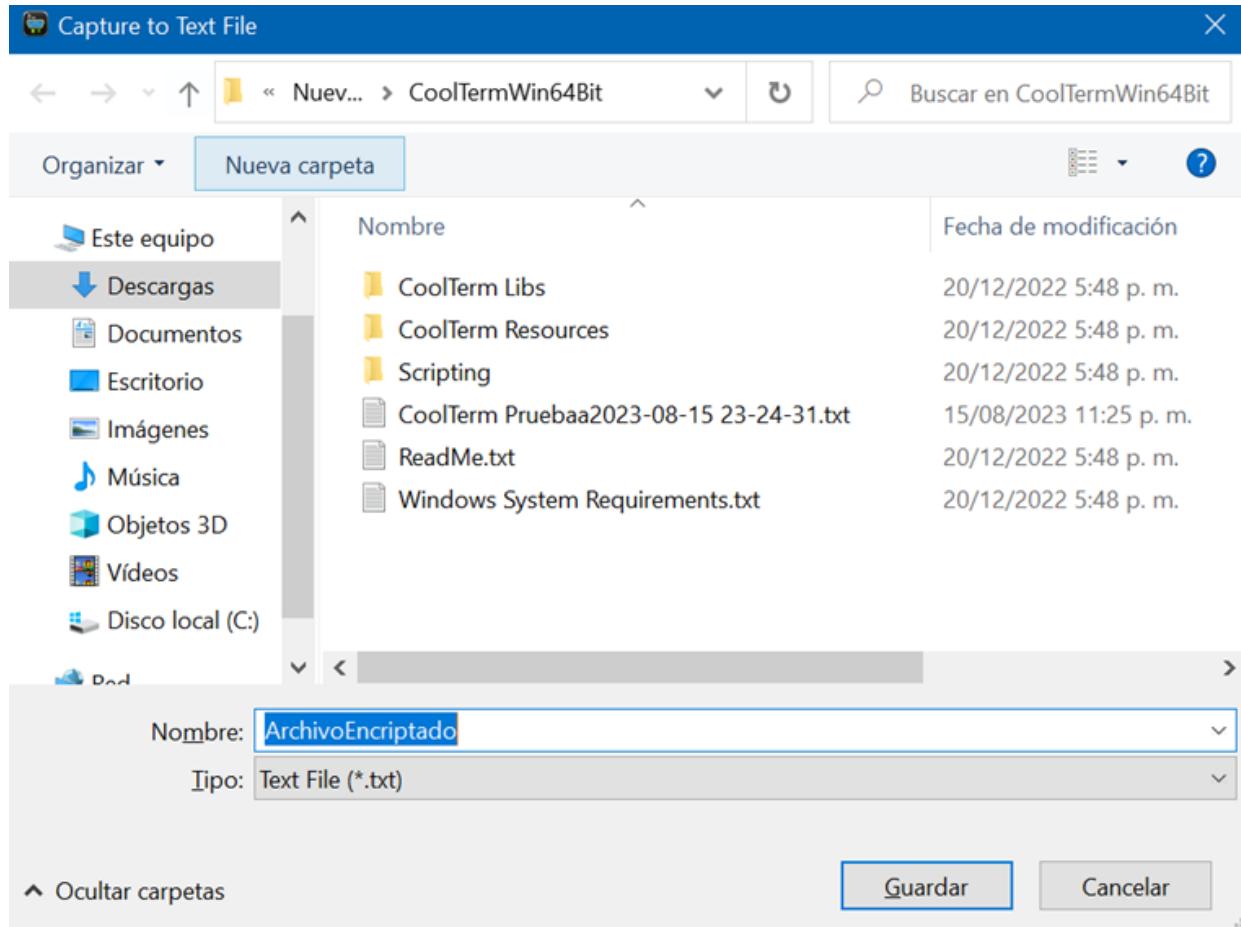
COM8 / 115200 8-N-1
Connected 00:02:41, 1.618 / 0 bytes

TX RX RTS CTS DTR DSR DCD RI

qJ5AStrO-Y2znKXWGwlwqu2un7wClfuEUswM_2vRtibpYPqnQHR04sfIBxDZseee8Q5dSSK6ok- Jd0Fd5b4LR0gabxHbT90lm9Mc6fQ30UGlpGb04wPTPM0INL-58HPAVH1W5-mS6p226MLyC1zrUK_QwZRYvNjMk9q- eVU4JImtX0ynH1n

Buscar

Luego de esto se inicia la recepción de datos para luego de esto guardarlos en un .txt y elegir una carpeta destino.



Referencias:

- [1] <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- [2] <https://lobodarobotica.com/blog/o-que-e-esp32-pra-que-serve-quando-usar/>
- [3] <https://deinfo.uepg.br/~alunoso/2019/SO/ESP32/HARDWARE/>
- [4] <https://aprendiendoarduino.wordpress.com/2016/12/11/ide-arduino/>
- [5] <https://www.programadornovato.com/installar-esp32-en-arduino-ide/>
- [6] <https://www.youtube.com/watch?v=rGjYxXve5s&t=1s>
- [7] <https://eclassevirtual.com/que-es-un-protocolo-de-enrutamiento-y-como-funciona/>

