



ACCESO POR MODBUS AL
INVERSOR
SAJ H1 S2
MEDIANTE
ESP32

V1.1



Índice de contenidos

• Agradecimientos.....	2
• Finalidad	3
• Materiales necesarios	3
• ESP32: USB to UART y localización del puerto COM	4
• Arduino IDE 1.8.18 Portable.....	4
• Instalación de librería y placas ESP32 en Arduino IDE	5
Instalación de las placas ESP32 (importante, la versión ha de ser la 1.0.6)	6
• Preparación del sketch y volcado al ESP32.....	8
Abrir el sketch	8
Modificaciones en el sketch.....	8
Selección de la Placa y ajustes previos a la compilación.....	9
Subir la APP a la ESP32	10
• Comprobación del funcionamiento de la APP	12
• Posibles errores y solución.....	13
Errores al compilar	13
Errores al monitorizar la ESP32	13

Agradecimientos

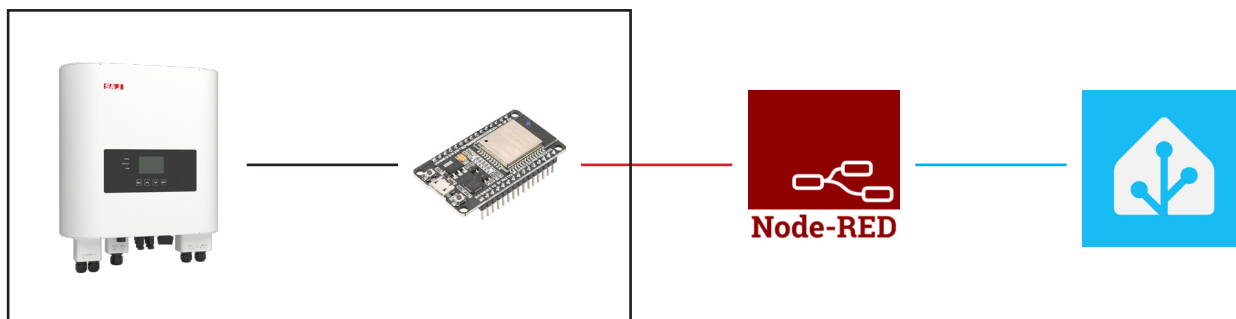
Este tutorial se ha realizado gracias a toda la información que se encuentra en el grupo de telegram.

SAJ H1 Inversor híbrido
https://t.me/saj_nooficialoriginal



Finalidad

La finalidad de este tutorial es la grabación de un esp32 para que se conecte por bluetooth al Dongle del inversor utilizando la librería blufi de espressif. A su vez, el esp32 levanta un servicio tcp que permite recibir llamadas con protocolo modbus tcp.



Materiales necesarios

Los materiales necesarios para realizar este tutorial son los siguientes:

- **Sketch h1_s2_modbus_esp32.ino**

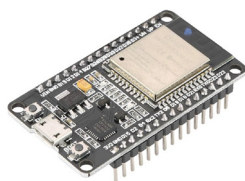
Descargar el sketch desde github a través del siguiente enlace:

https://github.com/sgsancho/saj_h1_s2_modbus_esp32/blob/main/h1_s2_modbus_esp32.ino

- **Placa de desarrollo ESP32**

En caso de no tener, hay que comprar una placa de desarrollo para poder grabar la aplicación en ella, por poner algún enlace de ejemplo (en Aliexpress cuestan la mitad):

<https://www.amazon.es/HiLetgo-ESP-wroom-32-Desarrollo-procesador-microcontrolador/dp/B0718T232Z/>



- **Arduino IDE 1.8.18 (en este tutorial se va a usar la versión portable)**

<https://downloads.arduino.cc/arduino-1.8.18-windows.zip>



- **Drivers USB to UART**

Según la placa que se adquiriera, es necesario que se instalen algunos drivers para que el ordenador reconozca la ESP al conectarla, para que reconozca el puerto COM:

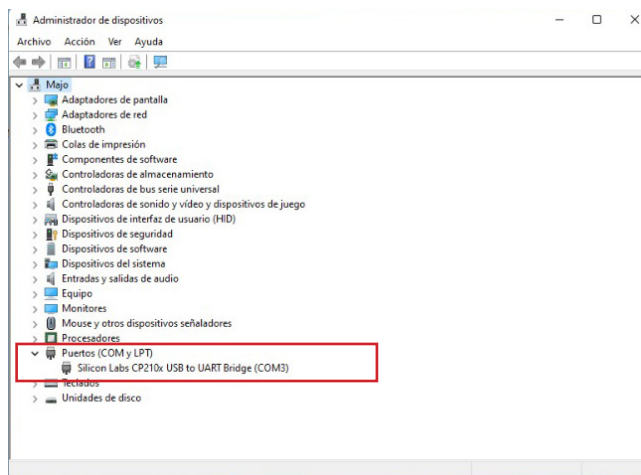
<https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads>



ESP32: USB to UART y localización del puerto COM

Comprobar si el PC reconoce la placa ESP32 al conectarla y localizar el puerto COM

Enchufar la ESP32 en el ordenador, ir a administrador de dispositivos, si el ordenador no nos reconoce la placa aparecerá en dispositivos no reconocidos y será necesario descargar e instalar los drivers que se han puesto en la página anterior para el USB to UART, en caso de estar instalados los drivers aparecerá en la sección “Puertos (COM y LPT)” como se aprecia en la siguiente imagen:



Como se aprecia en la imagen, la ESP32 conectada en este caso está en el puerto COM3, este dato es posible que debas recordarlo para seleccionarlo en Arduino IDE posteriormente.

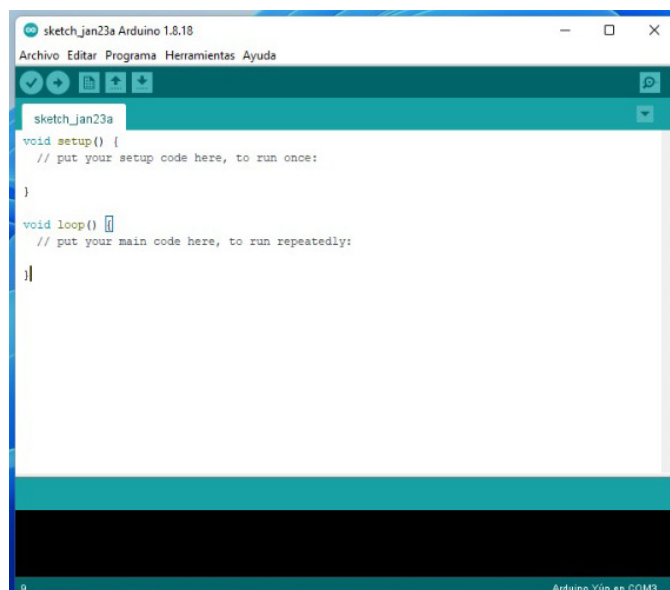
Arduino IDE 1.8.18 Portable

Como se ha comentado en la página de materiales, en este tutorial, vamos a utilizar la versión portable de Arduino IDE 1.8.18.

Si ya tienes descargado el zip de la página de materiales, lo único que hay que hacer es descomprimir el zip en una carpeta llamada “portable”.

Ya se puede ejecutar Arduino IDE

Accedemos a la carpeta y ejecutamos Arduino IDE, al cargar nos aparecerá la siguiente ventana:



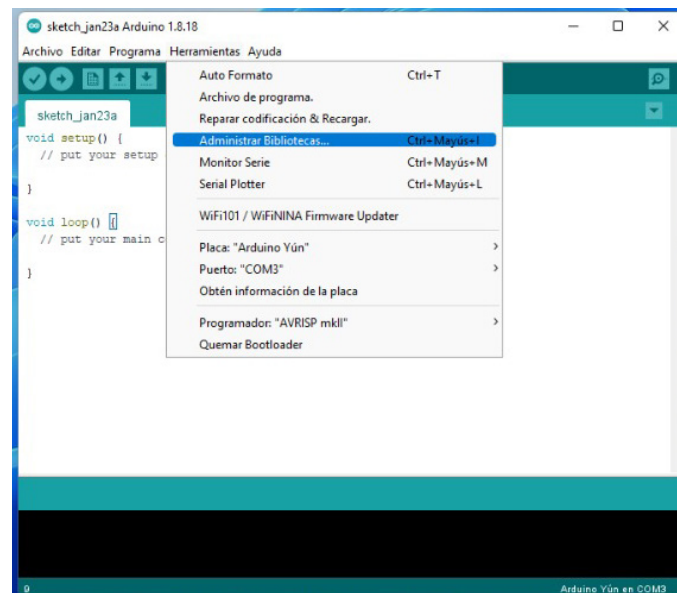


Instalación de librería y placas ESP32 en Arduino IDE

En este apartado nos centramos en instalar en Arduino tanto la librería de BLEdevice y las placas de ESP32 para que nos la reconozca Arduino IDE.

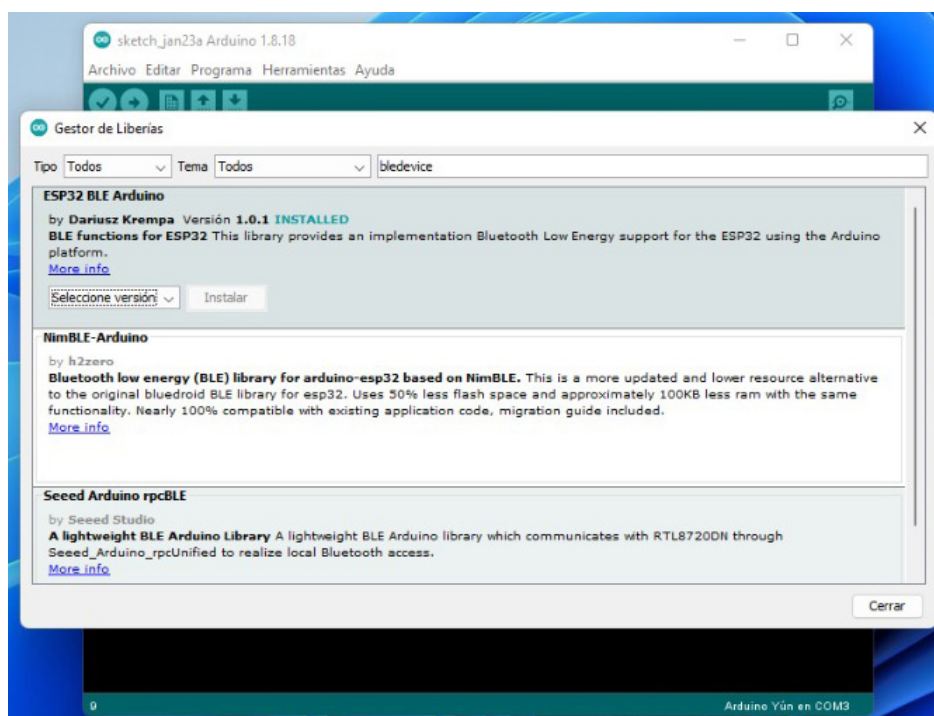
• Instalación de la librería BLEdevice

En la barra de herramientas superior desplegamos la pestaña de “Herramientas” y escogemos la opción “Administrar Bibliotecas...”:



Nos aparecerá la siguiente ventana, en el buscador de la parte superior escribiremos “bledevice”, tendremos que instalar la primera opción que aparece, “ESP32 BLE Arduino”, para ello escogeremos la versión “1.0.1” y presionamos “Instalar”.

Una vez descargado e instalado quedará de la siguiente manera:



Ya podemos cerrar la ventana del “Gestor de librerías”.

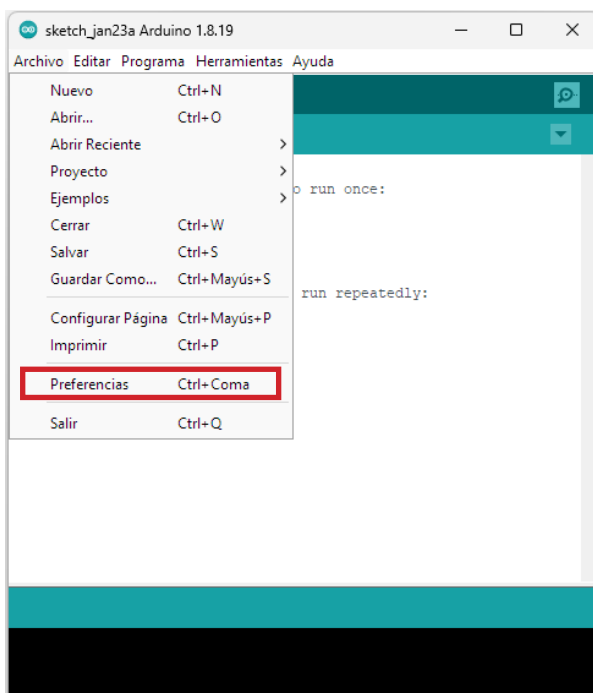


• Instalación de las placas ESP32 (importante, la versión ha de ser la 1.0.6)

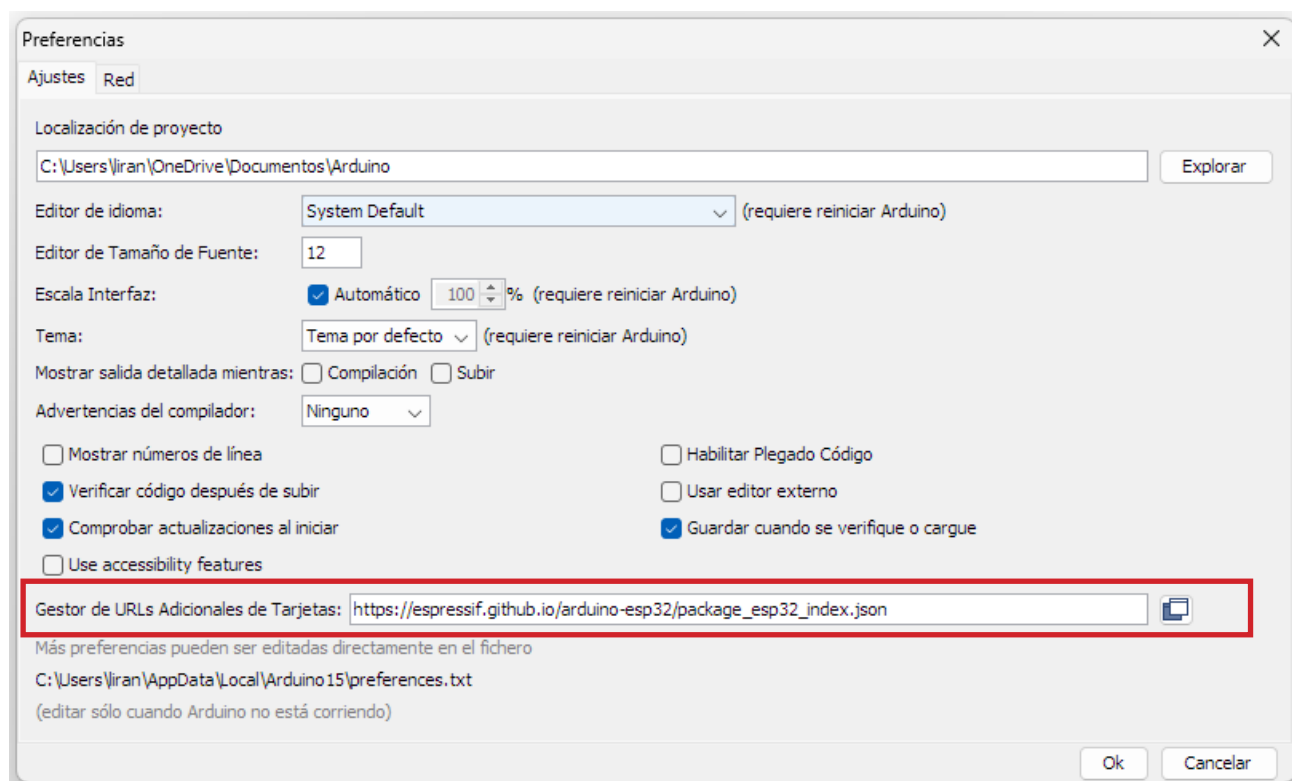
Antes de ir a instalar las placas hay que agregar el enlace de github de Espressif en las preferencias de Arduino IDE, este paso es necesario para poder descargar la versión 1.0.6, el enlace que hay que incluir es el siguiente:

https://espressif.github.io/arduino-esp32/package_esp32_index.json

Para ello desplegamos “Archivo” y abrimos las preferencias:

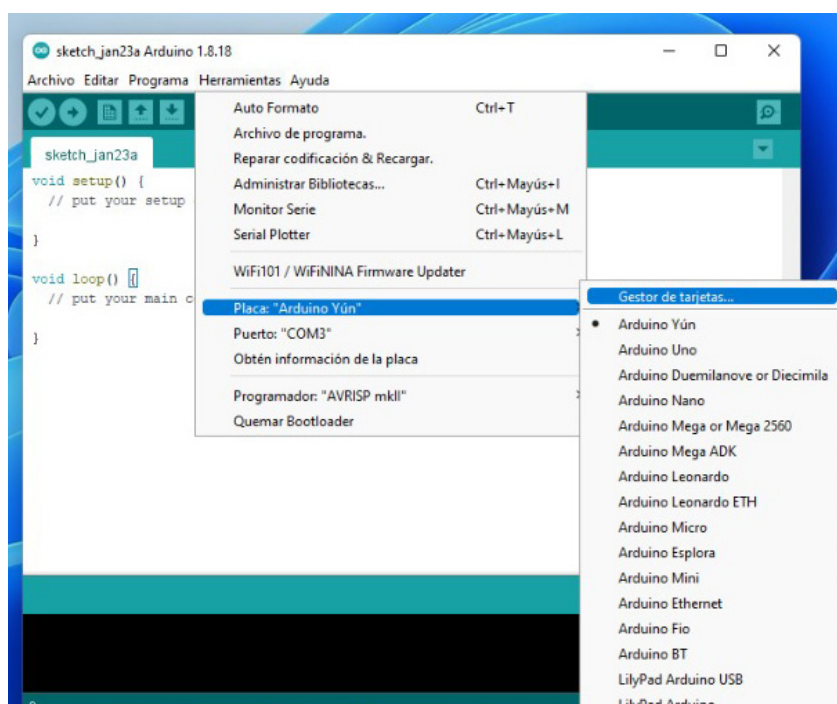


Dentro de las preferencias pegamos el enlace anterior en el apartado de “Gestor de URLs Adicionales de Tarjetas” y pulsamos “Ok”:

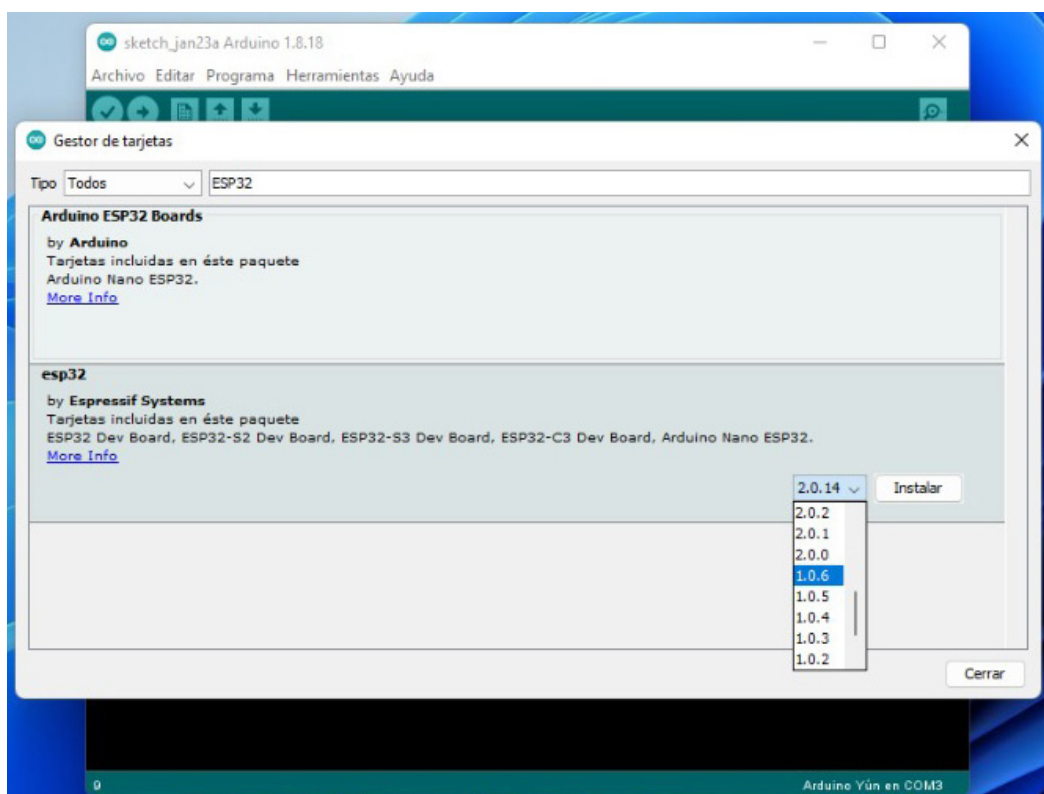




Ya podemos ir a instalar las placas, para ello abrimos la pestaña de “Herramientas” y bajamos hasta el apartado placas, se abrirá otro desplegable en el que podremos seleccionar la opción “Gestor de tarjetas...”:



Se abrirá una ventana similar a la del gestor de librerías de antes, en este caso en el buscador pondremos lo siguiente, **“ESP32”**, al cargar la búsqueda nos cargará dos resultado, en este caso nos quedaremos con la segunda opción, **“esp32 by Espressif Systems”**, antes de pulsar instalar hay que seleccionar la versión **“1.0.6”**



Una vez terminada la instalación ya podremos cerrar la ventana del “Gestor de tarjetas”, dentro del apartado placas de herramientas ya nos aparecerán las placas ESP32.

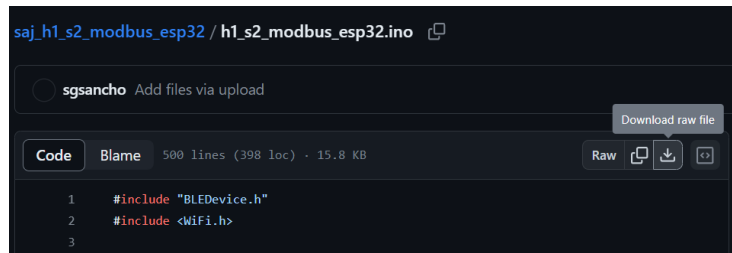


Preparación del sketch y volcado al ESP32

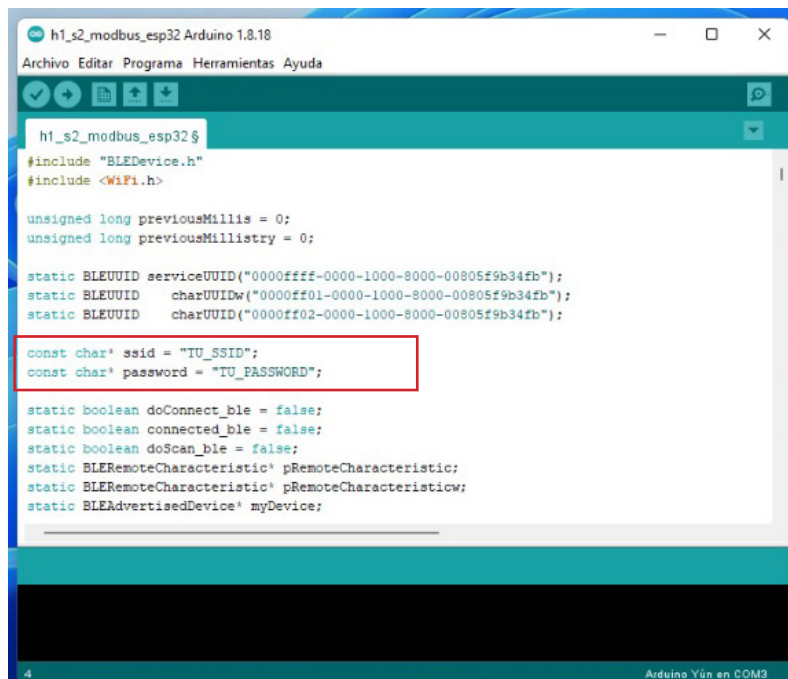
• Abrir el sketch

Con los anteriores pasos realizados, ya pasamos a cargar el sketch que tenemos que flashear en nuestro ESP32, si no lo has descargado al principio del tutorial, descarga el sketch desde github a través del siguiente enlace:

https://github.com/sgsancho/saj_h1_s2_modbus_esp32/blob/main/h1_s2_modbus_esp32.ino



Si ya lo tenías descargado, desde Arduino IDE lo abrimos, para ello desplegamos “Archivo” y seleccionamos la opción “Abrir...”, nos vamos a la ruta donde esté en **h1_s2_modbus_esp32.ino** y lo abrimos, nos saltará una ventana diciendo que ha de crear una carpeta para meter el sketch, aceptamos y se abrirá una nueva ventana de Arduino IDE con el sketch, podemos cerrar la otra ventana ya que no nos es necesaria:



• Modificaciones en el sketch

Los únicos cambios que hay que realizar en este sketch están en las líneas 11 y 12:

```
const char* ssid = "TU_SSID";
```

```
const char* password = "TU_PASSWORD";
```

En ambos casos dejaremos las comillas.

En ssid, pondremos el nombre del WiFi de casa (es necesario que el WiFi sea de 2.4GHz ya que casi todas las ESP32 solo van a esa banda)

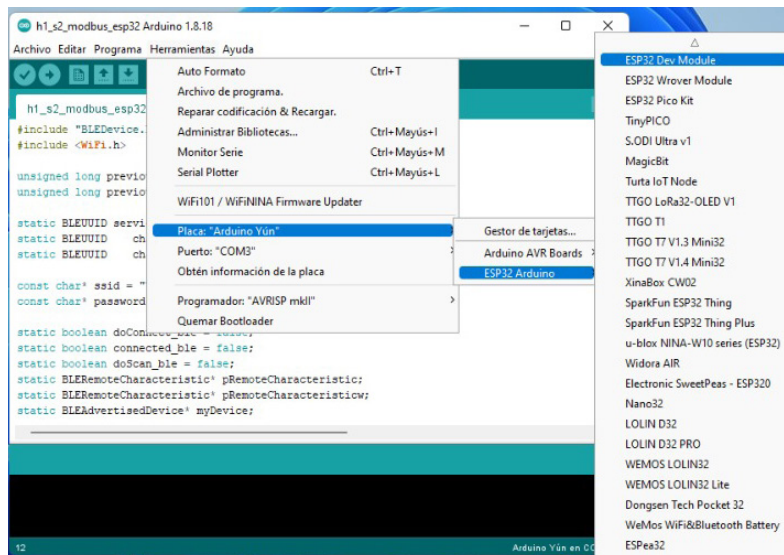
En password pondremos la clave del WiFi de casa.



• Selección de la Placa y ajustes previos a la compilación

Hecho todo lo anterior, solo nos queda seleccionar la placa y poner los ajustes para el volcado del sketch a la placa.

Para ello, nos vamos a la pestaña de “**Herramientas**” y bajamos al desplegable de “**placa**” y en el apartado “**ESP32 Arduino**” seleccionaremos el tipo de placa ESP32 que tenemos, en el caso de la placa que he puesto de ejemplo, la placa a seleccionar sería la placa “**ESP32 Dev Module**”:

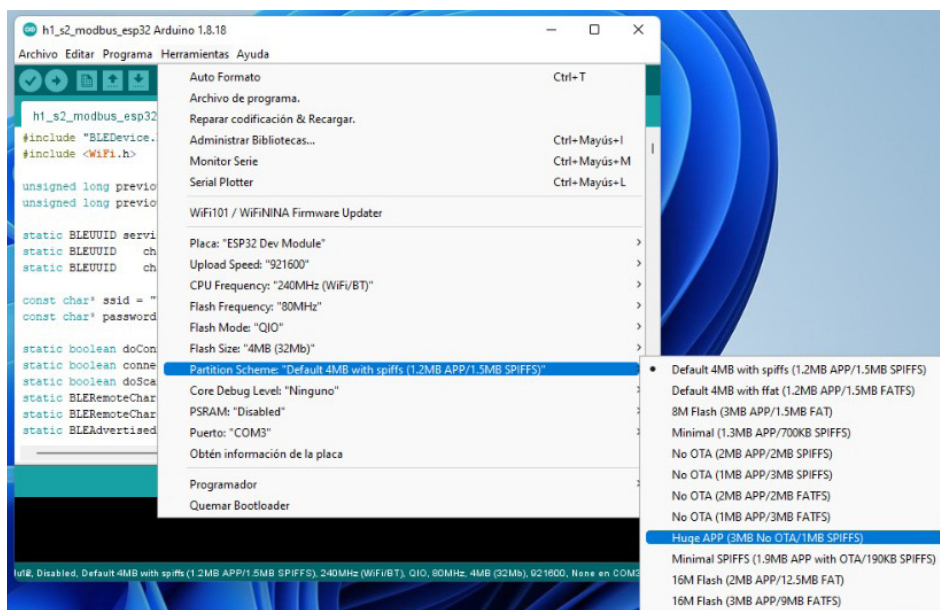


Ya seleccionada la placa, la pestaña de herramientas habrá cambiado y habrá más opciones asociadas a las placas ESP32.

Las únicas dos opciones que debemos tocar son los apartados “**Partition Scheme**” y “**Puerto**”:

Partition Scheme

En este caso y siguiendo con el ejemplo de la ESP32 que puse al principio, es necesario escoger la opción “**Huge APP (3MB No OTA/1MB SPIFFS)**”



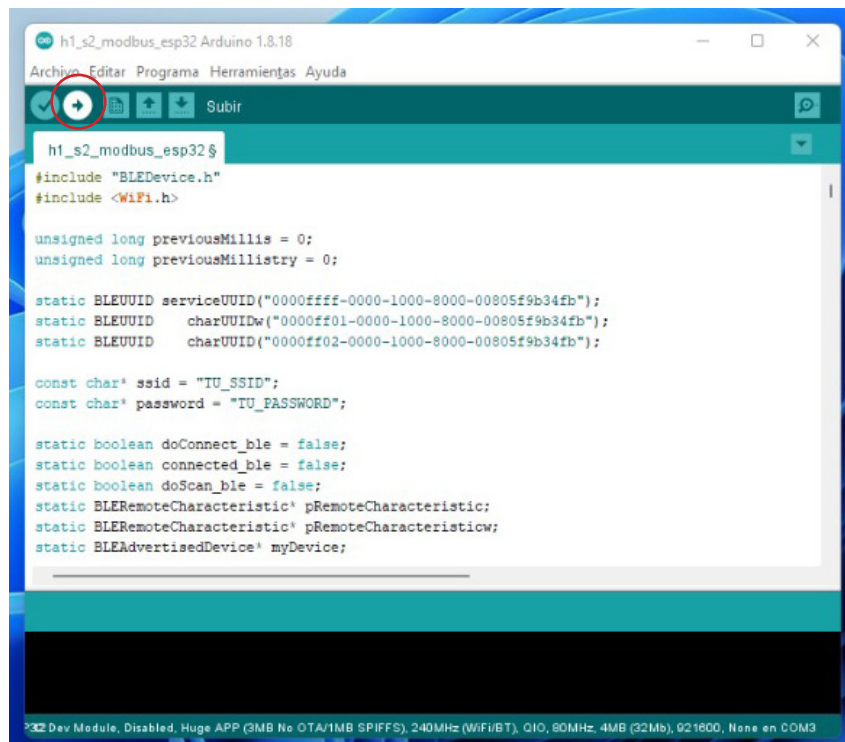
Puerto

En esta parte, lo único que tendremos que seleccionar el puerto COM de nuestra ESP32, en principio, debería estar seleccionada de forma automática, en el caso de no ser así, se elige y listo, en el caso del ejemplo el puerto COM era el 3 y ya está elegido en la imagen anterior.

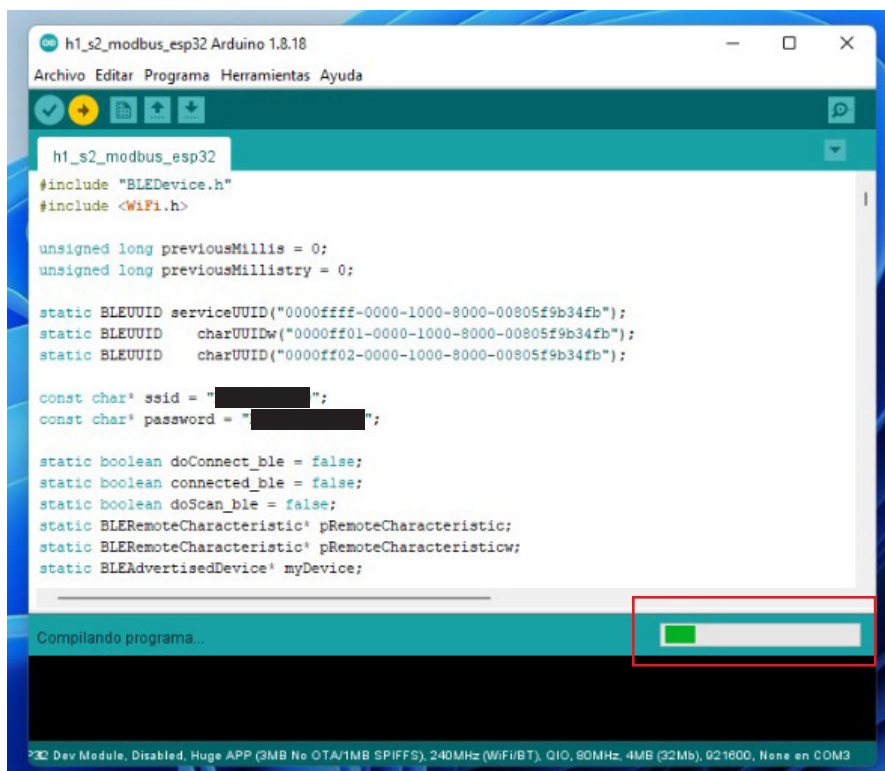


• Subir la APP a la ESP32

Ya solo queda compilar y subir la APP a la ESP32, para esto solo es necesario pulsar el botón de la flecha en la parte superior izquierda de Arduino IDE:



Una vez apretado dicho botón Arduino empezará a compilar la APP para volcarla en la ESP32, aparece en la parte inferior derecha una barra verde que se irá llenando:

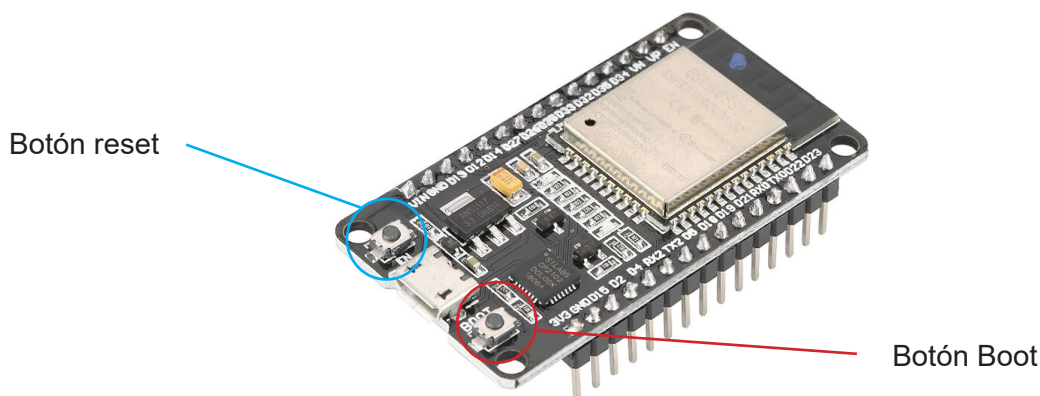


Una vez completada la compilación en la parte negra de abajo empezará a aparecer texto en blanco y naranja.



```
Subiendo...
El Sketch usa 1506694 bytes (47%) del espacio de almacenamiento de programa. El máximo es 314572
Las variables Globales usan 55240 bytes (16%) de la memoria dinámica, dejando 272440 bytes para
esptool.py v3.0-dev
Serial port COM3
Connecting....._
```

En ese momento empezará el volcado, cuando aparece el texto “**Connecting....._**” será necesario apretar y mantener pulsado el botón “**Boot**” en la ESP32 (hay dos botones en la propia placa, en un lado tienes el botón reset “**RST**” y en el otro está el botón “**BOOT**”).



Cuando salte el punto de “Connecting” empezará a flashear la ESP32, y empezarán a aparecer el porcentaje de grabación de la misma:

```
Subido
Writing at 0x000c0000... (81 %)
Writing at 0x000c4000... (83 %)
Writing at 0x000c8000... (85 %)
Writing at 0x000cc000... (87 %)
Writing at 0x000d0000... (89 %)
Writing at 0x000d4000... (90 %)
Writing at 0x000d8000... (92 %)
Writing at 0x000dc000... (94 %)
Writing at 0x000e0000... (96 %)
Writing at 0x000e4000... (98 %)
Writing at 0x000e8000... (100 %)
```

Una vez completada la grabación nos aparecerá el siguiente texto:

```
Subido
Writing at 0x000e8000... (100 %)
Wrote 1506816 bytes (896837 compressed) at 0x00010000 in 14.9 seconds (effective 810.9 kbit/s)
Hash of data verified.
Compressed 3072 bytes to 119...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (119 compressed) at 0x00008000 in 0.1 seconds (effective 384.0 kbit/s)
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
```

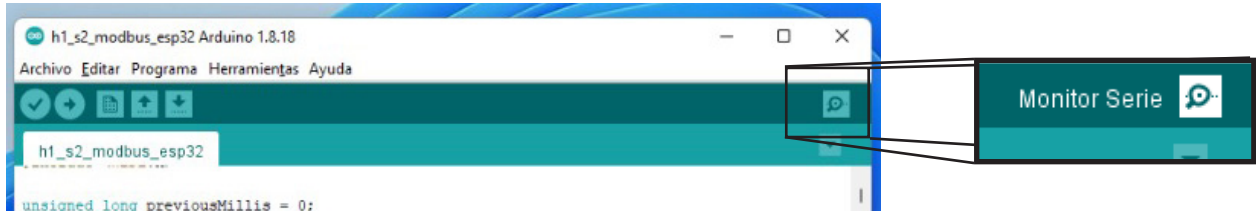
Antes de pulsar el botón reset de la ESP32, vamos a comprobar que todo sea correcto en el siguiente paso.



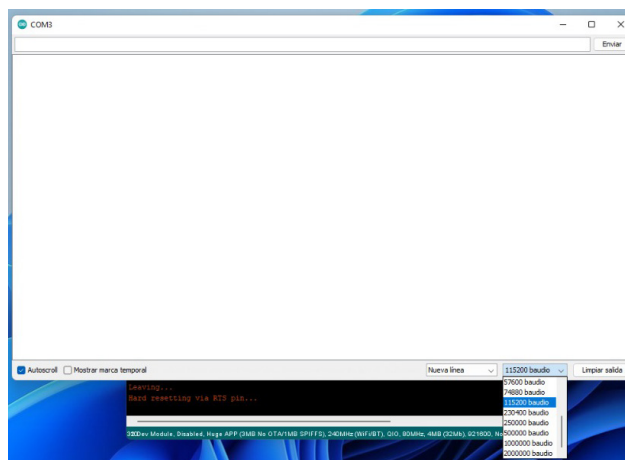
Comprobación del funcionamiento de la APP

NOTA: Es necesario estar cerca del inversor y del router WiFi para que no hayan errores de conexión.

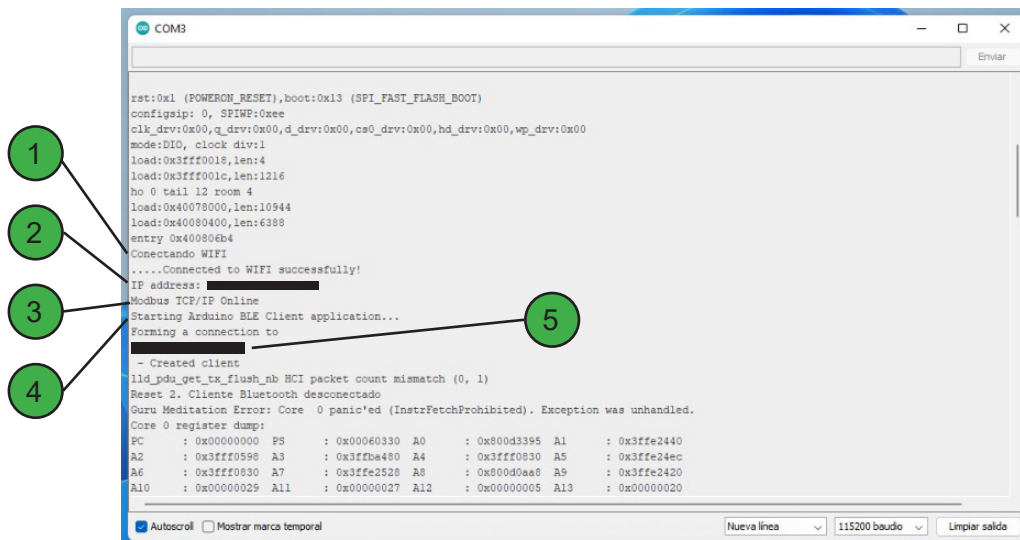
Para comprobar el funcionamiento de la APP en la ESP32, antes de pulsar el botón reset “**RST**” vamos a abrir el “**Monitor Serie**”



Nos aparecerá una ventana nueva, en la que debemos seleccionar “**115200 baudio**” en la parte inferior derecha:



Ahora ya pulsamos el botón reset de la ESP32 y nos empezará a monitorizar el arranque de la placa, si todo está correcto, nos aparecerá una pantalla como la siguiente:



Los números hacen referencia a:

1. ESP32 intenta la conexión al WiFi,
2. Una vez conectada, aparecerá la IP que tiene la ESP32 en el router, es necesario hacerla fija en el portal de router.
3. Apertura del servicio Modbus TCP
4. Abriendo el cliente BLE
5. Conexión con el dongle del inversor, aquí aparece la MAC del dongle.

YA HAS CONSEGUIDO CONECTAR LA ESP32 AL DONGLE DEL INVERSOR



Posibles errores y solución

Hay dos tipos de errores que se pueden dar, por un lado están los errores al compilar, y por otro “errores” con la placa ya grabada:

• Errores al compilar

Al intentar compilar la APP me aparece un error y no llega a rellenarse la barra verde, aparece el siguiente texto o similar:

Compilation error: ‘union arduino_event_info_t’ has no member named ‘disconnected’; did you mean ‘eth_connected’?

O

Ringbuffer(size_t length, ringbuf_type_t type = RINGBUF_TYPE_NOSPLIT);

También pueden aparecer errores de **duplicidades en la librería BLEdevice.h**, por lo que se para la compilación.

SOLUCIÓN:

Estos errores se producen por no haber usado las placas ESP32 correctas, lo más probable es que estés en una versión más reciente de la que hay que utilizar, elimina las placas en el programa y vuelve a instalar la versión 1.0.6, en caso de seguir con el error, habría que borrar todo posible rastro de Arduino y empezar de cero el tutorial.

• Errores al monitorizar la ESP32

Estos errores se producen una vez ya compilada la ESP32:

En el monitoreo serial, nunca llega a conectarse al WiFi.

SOLUCIÓN:

Este error puede deberse a dos cosas, el SSID y/o el Password puestos en el sketch no es correcto, por lo que hay que revisar que esté correcto y en el caso de no estar bien habría que corregirlo y volver a compilar y grabar la ESP32.

El WiFi solo está en la banda de 5GHz y por esto no se puede conectar, activar la banda 2.4GHz.

La ESP32 se conecta a internet, pero no me aparece la IP, a parte tampoco carga el Modbus TCP, salta directamente a abrir el cliente BLE.

SOLUCIÓN:

Has conseguido grabar la ESP32 sin usar las placas correctas y sin la librería BLEdevice, esto se debe a que la versión de placas ESP32 2.0.10 en adelante ya incluyen la librería BLEdevice.h, y se puede llegar a grabar la ESP32, pero el sketch no es compatible con estas librerías y no hace bien la función.

Se queda congelado intentado abrir el cliente BLE.

SOLUCIÓN:

Lo más probable es que estés demasiado lejos del inversor, por lo que no llega a encontrar nunca el dongle y no salta desde este punto, acércate al inversor lo máximo posible.

**En caso de encontrar otros errores no contemplados aquí
coméntalo en el grupo de Telegram.**