

## Protocolo de resolución de problemas MeteoMex Aeria

<b>HARDWARE</b>	<b>2</b>
Revisión de componentes externos al dispositivo	2
Revisión de componentes del dispositivo	2
Reinicio del dispositivo	2
<b>SOFTWARE</b>	<b>3</b>
Instalación de Arduino IDE	3
Configuración inicial de Arduino IDE	5
Problemas para instalar librerías, acceso denegado.	5
¿Qué hacer si el puerto USB al que está conectado el dispositivo no aparece en Arduino IDE?	6
Agregar boards extra (ESP32)	7
Selección de placa en Arduino IDE	11
Gestión de bibliotecas	12
Agregar las dos librerías esenciales para utilizar MeteoMex Aeria	13
A) <i>BME280</i>	13
B) <i>SparkFun</i>	14
Error si no se instala alguna librería	16
Programación del dispositivo MeteoMex Aeria	16
Contraseña y nombre del internet	16
Verificar el token	17
Contraseña o nombre de red de internet incorrecta	19
Hard resetting via RTS pin	21
Los datos de lectura tienen un formato no legible	23
No se encuentra el sensor BME280	25

## HARDWARE

### Revisión de componentes externos al dispositivo

Para realizar un diagnóstico de una posible falla del dispositivo MeteoMex, se recomienda partir de la revisión de los componentes esenciales externos al dispositivo.

1. Verificar que hay luz eléctrica a la cual conectar el dispositivo.
2. Verificar que el cargador del dispositivo es funcional y que no esté dañado.
3. Verificar que el internet funciona y la señal alcance la zona donde se encuentra el dispositivo. Si es necesario, reiniciar el router de la red de internet.
4. Verificar que no haya cambios en la red de internet (nombre o contraseña)

### Revisión de componentes del dispositivo

Una vez verificados los puntos anteriores, si el dispositivo aún no realiza lecturas, se puede proceder a revisar que el dispositivo:

1. esté correctamente soldado
2. no presente daños en el puerto del cargador micro USB
3. verificar que el led del dispositivo se encienda al conectarlo a la luz eléctrica

### Reinicio del dispositivo

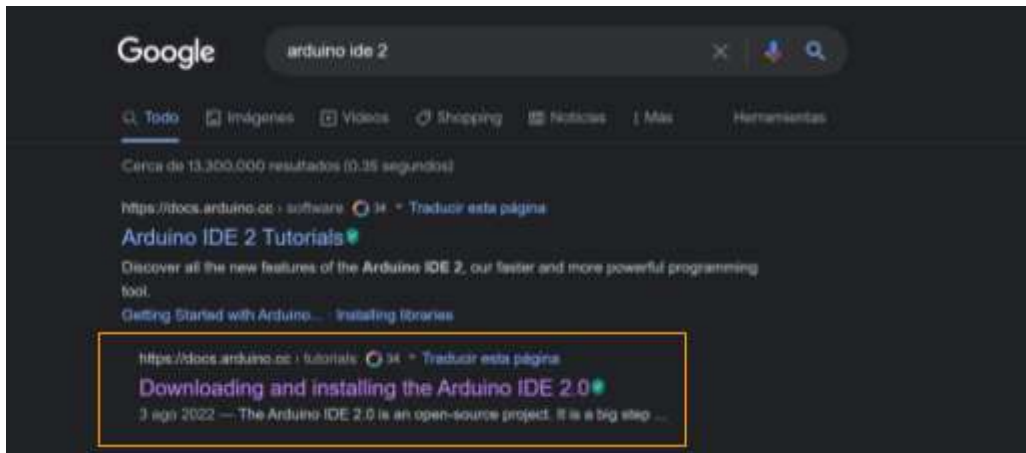
Si los aspectos anteriores se encuentran funcionales, se puede reiniciar el dispositivo de manera manual presionando el botón que se encuentra por encima del puerto de carga (resaltado en el cuadro naranja de la foto siguiente).



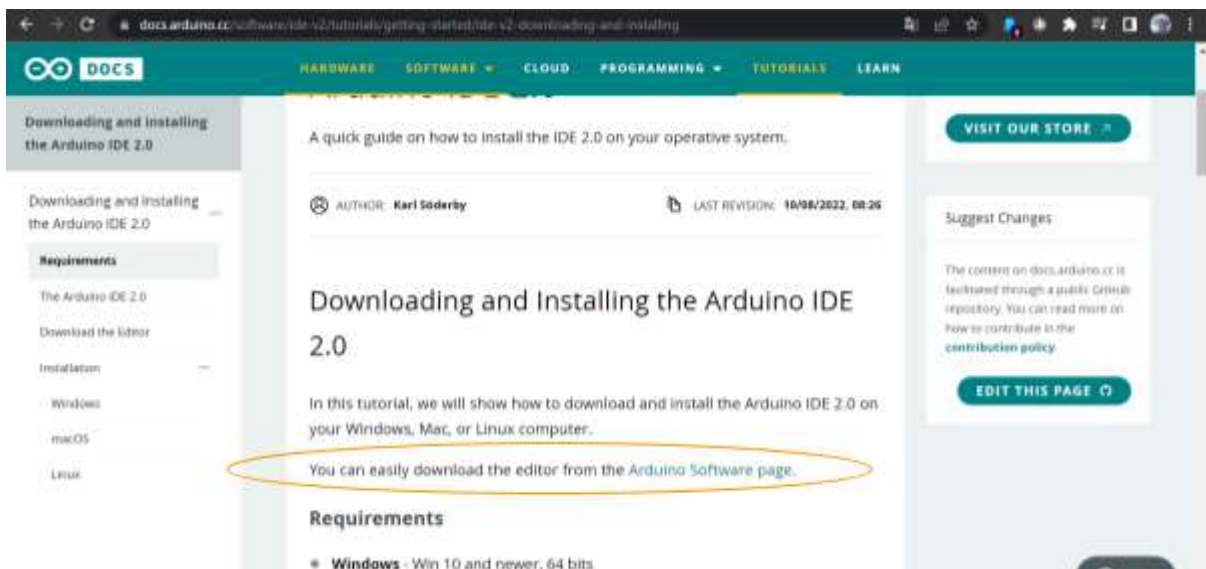
## SOFTWARE

### Instalación de Arduino IDE

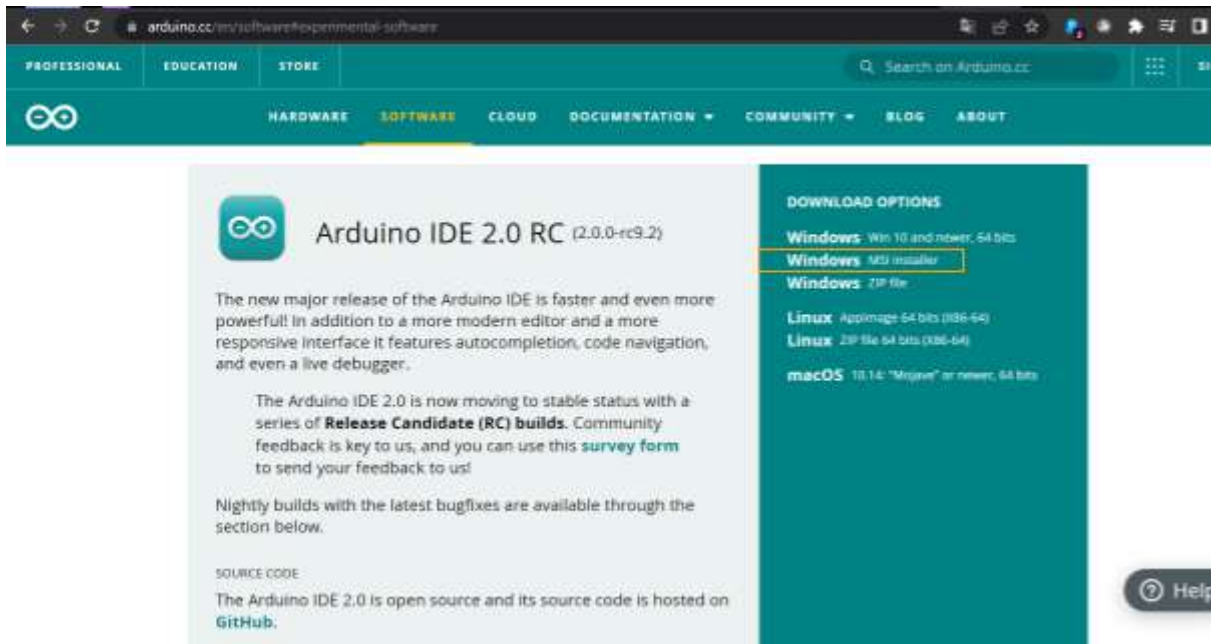
1. Realizar la búsqueda en el navegador e ir al sitio oficial de Arduino IDE. En este manual se hará referencia a la versión 2.0.0 de Arduino IDE usada en Windows 11.



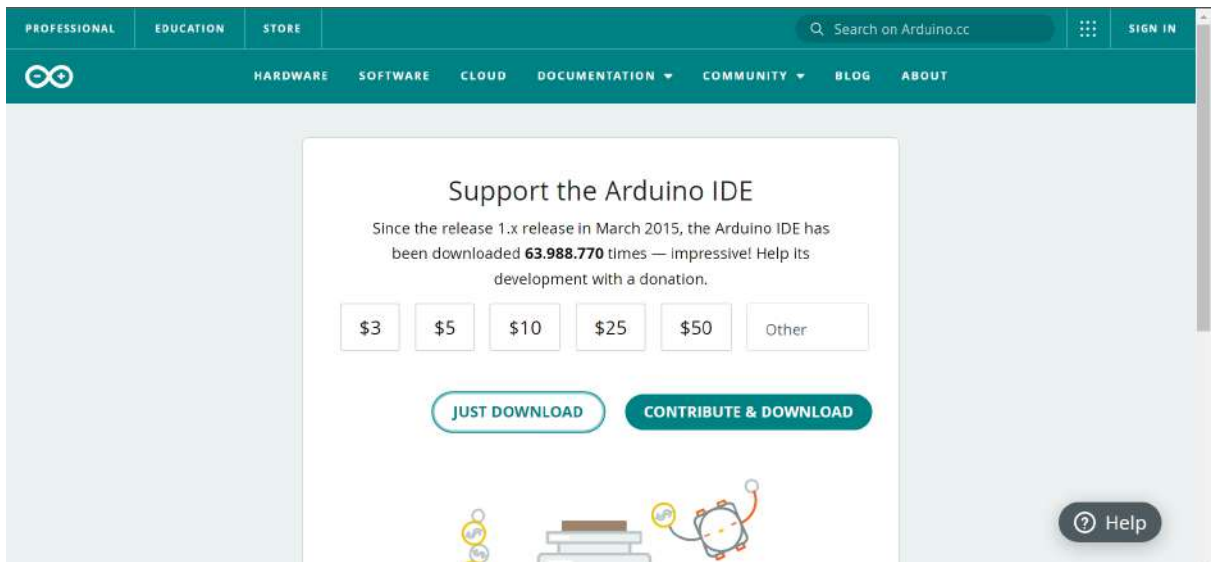
2. Ir a la página de software, dando click en el enlace <https://www.arduino.cc/en/software>



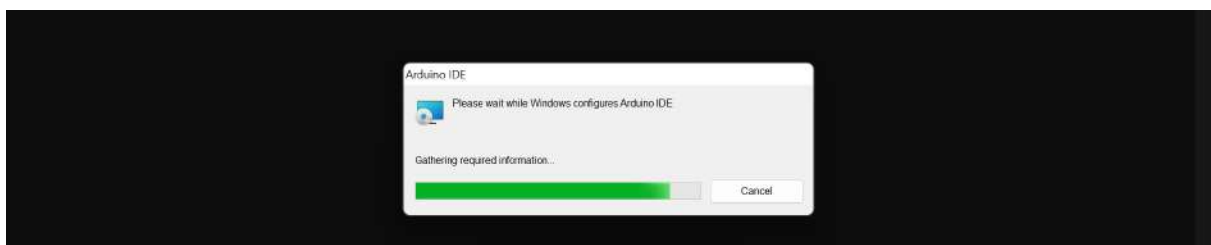
3. Seleccionar la opción para realizar la instalación que mejor nos convenga. En este ejemplo, se utilizará el instalador MSI para Windows.



Podemos seleccionar realizar una donación o descargarlo sin donar



- Una vez descargado el instalador, damos doble click en el archivo y esperamos a que se realice la instalación.



- El instalador se cerrará, pero veremos el acceso directo en nuestro escritorio. De este modo, tenemos instalado el Arduino IDE.



### **Configuración inicial de Arduino IDE**

Una vez descargado, podremos abrir Arduino IDE. En un inicio se descargan algunas librerías esenciales.

### **Problemas para instalar librerías, acceso denegado.**

Si al abrir Arduino IDE por primer vez, vemos un mensaje similar al siguiente:

**Could not install Arduino AVR platform, Access is denied**

Es muy probable que sea un problema con el antivirus.

**Solución:** Desactivar temporalmente antivirus.

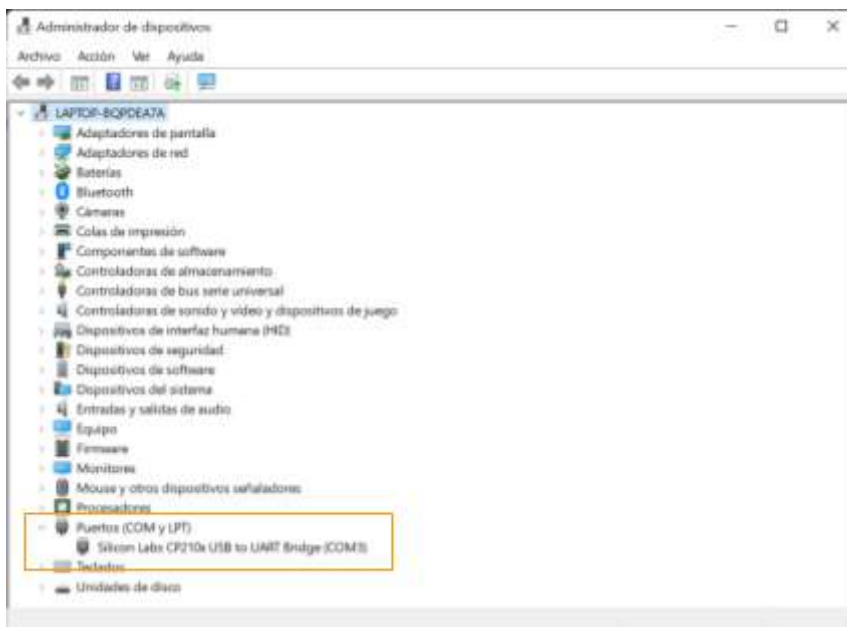
### ¿Qué hacer si el puerto USB al que está conectado el dispositivo no aparece en Arduino IDE?

Si el puerto USB en el que conectamos el dispositivo MeteoMex no aparece en la interfaz de Arduino IDE, en primer lugar podemos revisar que nuestro equipo de cómputo identifique el puerto.

Para verificar esto, en la barra de búsqueda de nuestro equipo colocaremos “Administrador de dispositivos”. Y verificaremos que al conectar el dispositivo, aparezca como “otros”.

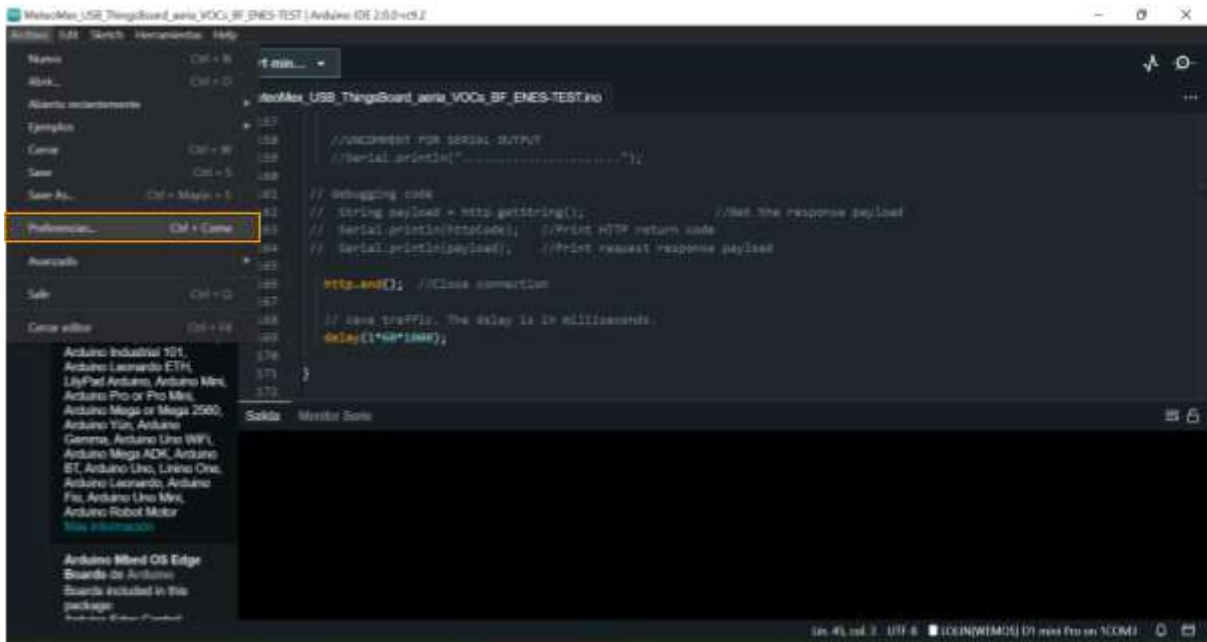
Si sí aparece, es necesario instalar drivers, particularmente: CP210x USB to UART Bridge VCP Drivers. Podemos encontrarlos en la página: <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads> en **Downloads > CP210x Windows Drivers**. Esto descargará un archivo zip. Seleccionaremos el archivo indicado para instalarlo (x64 o x86), dando clic en ejecutar, siguiente y finalizar.

Una vez instalados los drivers, al realizar la misma prueba, el puerto será reconocido como **Puertos (COM y LPT)**.

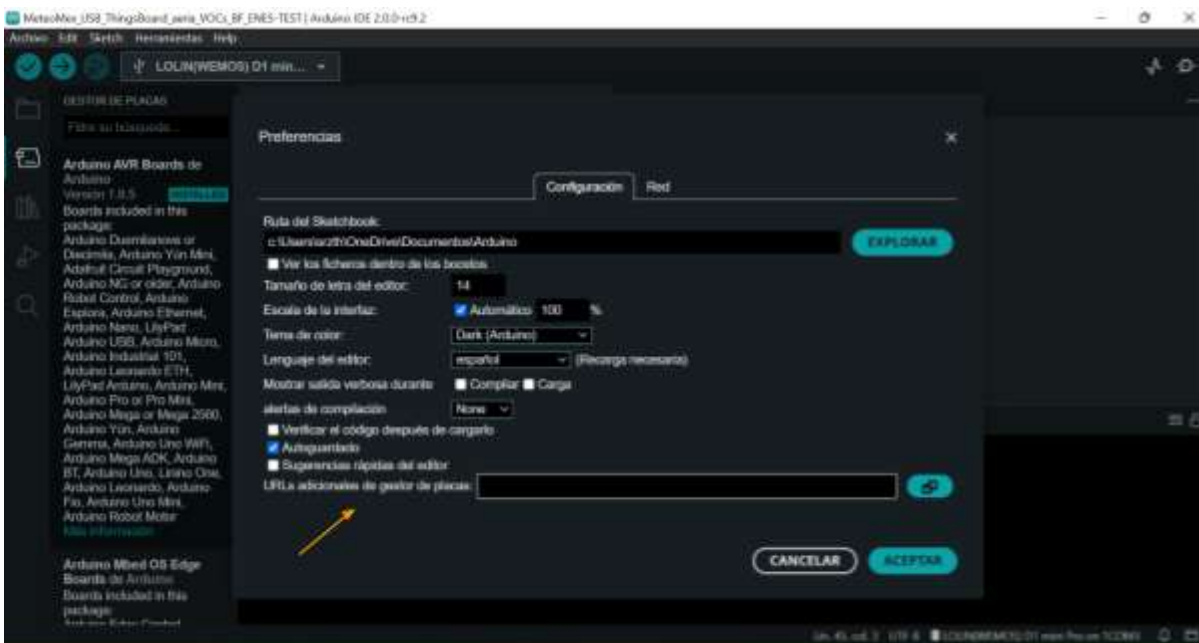


### Agregar boards extra (ESP32)

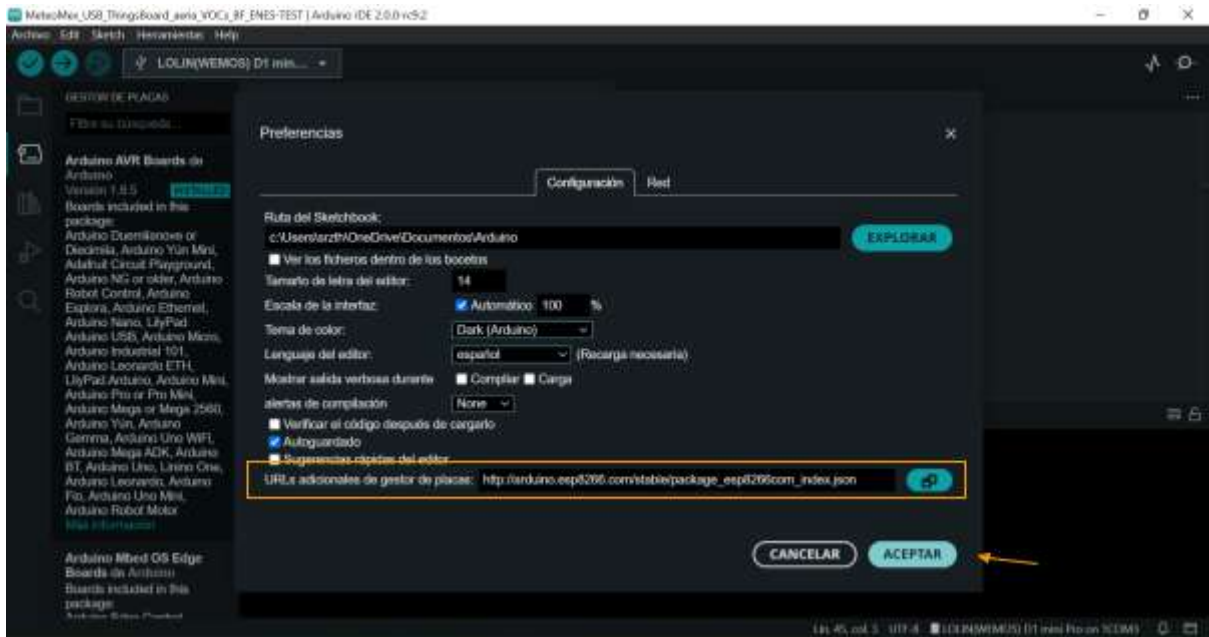
1. Una vez tengas instalado y abierto Arduino IDE, busca en la esquina superior izquierda el menú de archivo y dirígete a **Archivo > Preferencias**



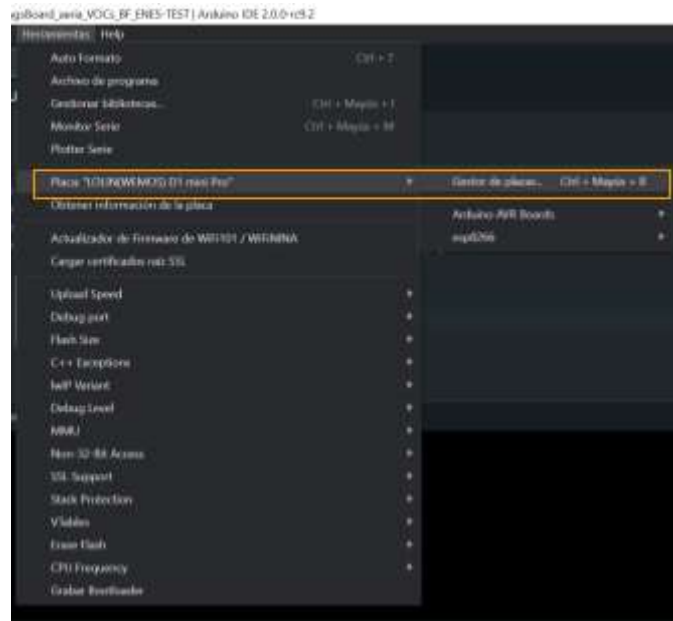
2. Se abrirá el cuadro de preferencias, que luce como el siguiente:



3. En el campo que dice "URLs adicionales de gestor de placas", copia la siguiente dirección: [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)  
Y presiona "Aceptar"

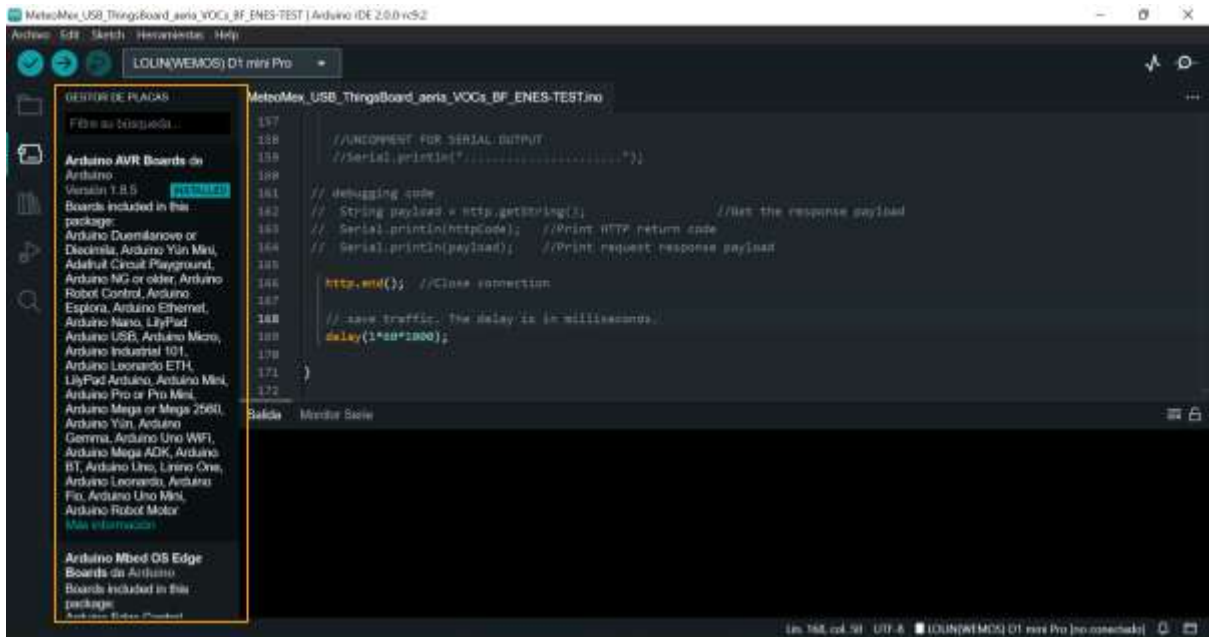


4. Ahora dirígete a **Herramientas > Placa > Gestor de placas**

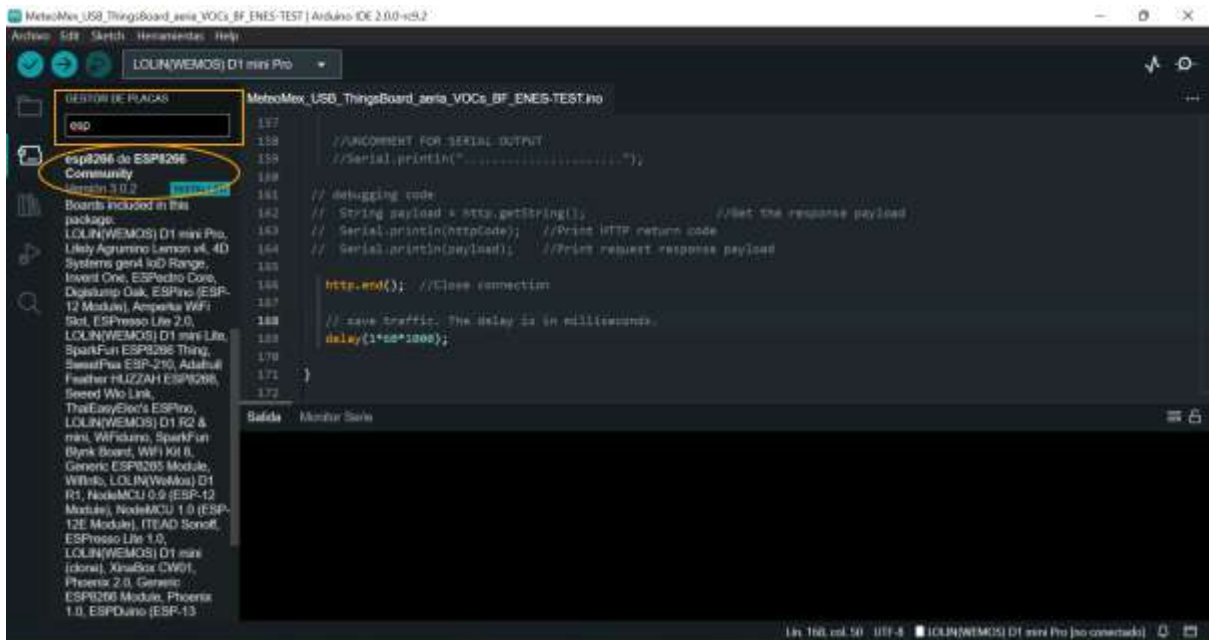


5. Esto abrirá un cuadro de búsqueda en el lateral izquierdo. Introduce la palabra clave “esp” en el cuadro de búsqueda. Selecciona esp8266 by ESP8266 y presiona “Instalar”.

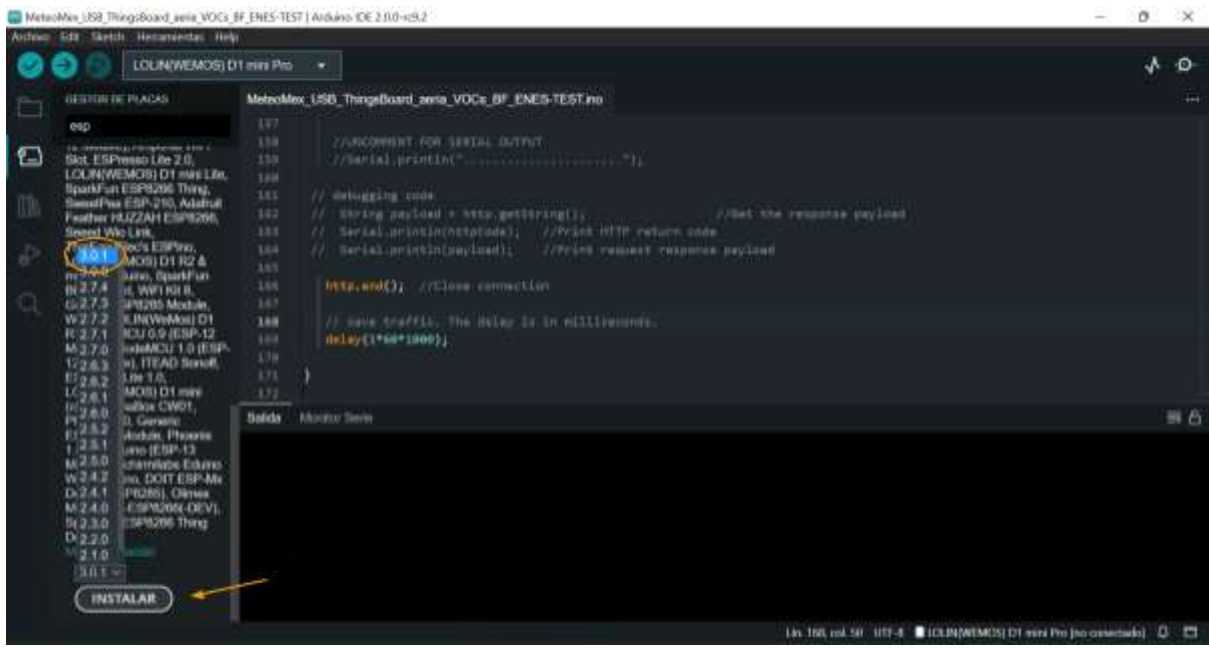




- Introduce la palabra clave “esp” en el cuadro de búsqueda.  
Busca la opción **esp8266** de **ESP8266**.



- Elege la Versión 3.0.2 y selecciona “Instalar”. (En el ejemplo se ve la selección de la versión 3.0.1).



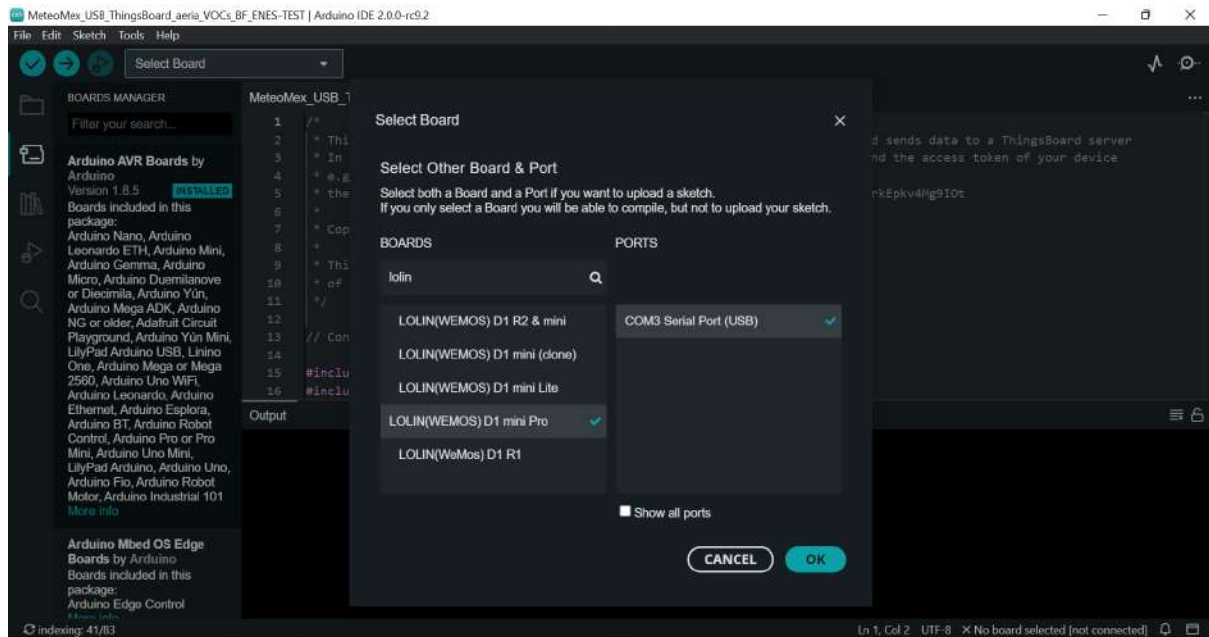
Una vez instalado el gestor de placas, podremos configurar el acceso a la placa del dispositivo MeteoMex.

## Selección de placa en Arduino IDE

Posterior a la instalación de ESP8266, podrás configurar el acceso a la placa.

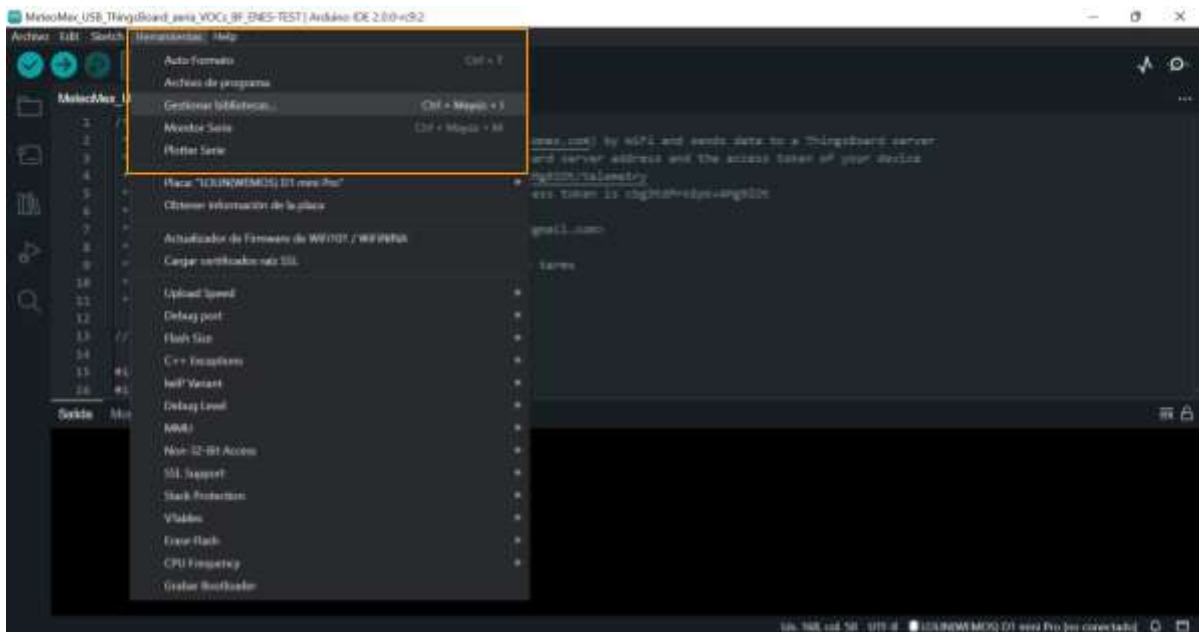
1. Buscaremos debajo del menú, en el área de accesos rápidos, una lista de selección titulada: **Selección de placa**. Al seleccionar la lista, se desplegará un cuadro de búsqueda que nos permitirá ingresar palabras clave.
2. Ingresé la palabra clave "lolin" y seleccione la placa **LOLIN(WEMOS) D1 mini Pro**. Una vez seleccionado el nombre de la placa, seleccione el puerto USB de la placa (en este caso COM3). Y seleccione "Aceptar".

De este modo, la placa quedará seleccionada y será identificada cada vez que sea conectada.

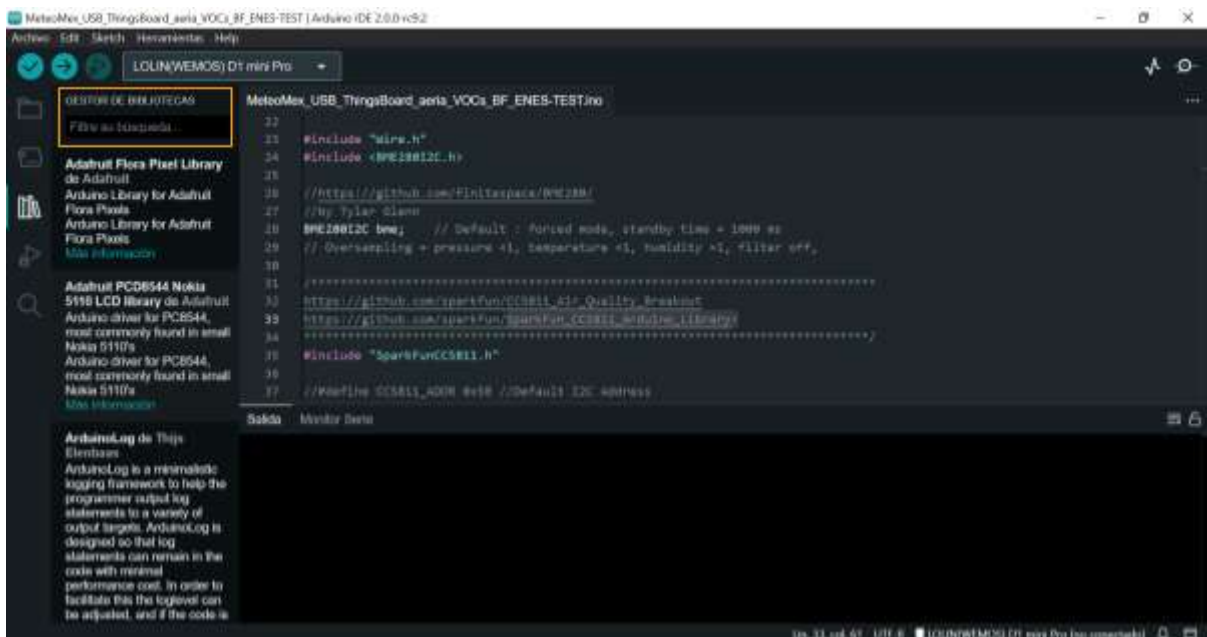


## Gestión de bibliotecas

1. Una vez tengas instalado y abierto Arduino IDE, dirígete a **Herramientas > Gestionar bibliotecas**



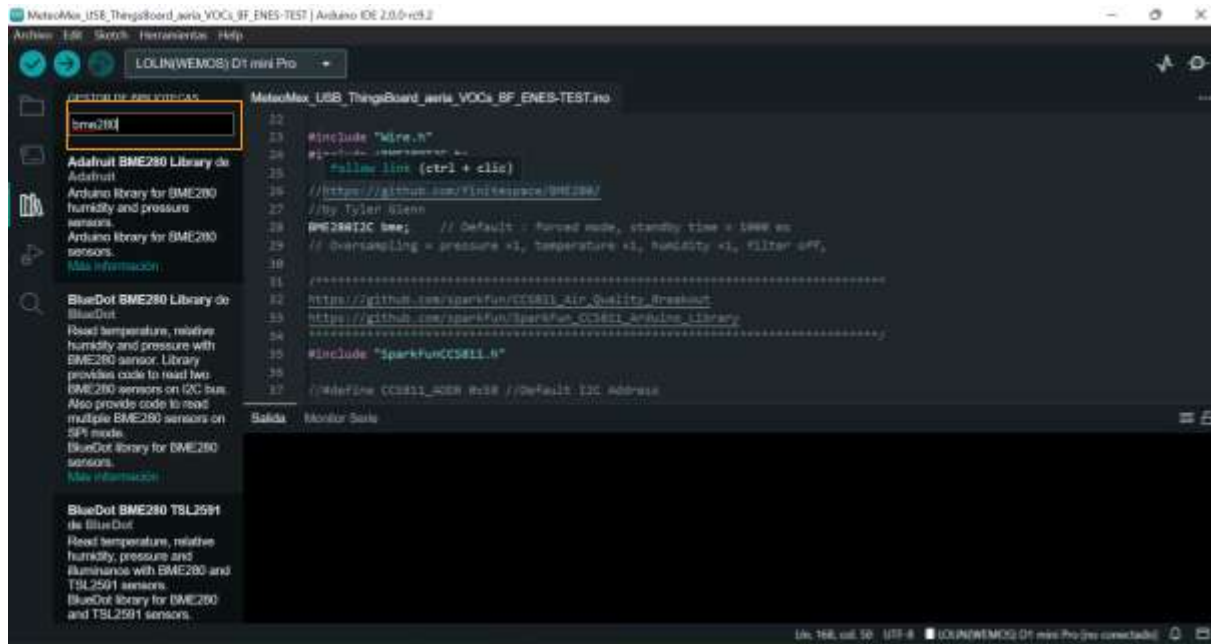
2. Se abrirá un menú de búsqueda en la parte lateral izquierda. En el cuadro de búsqueda ingrese la palabra clave de la biblioteca de interés.



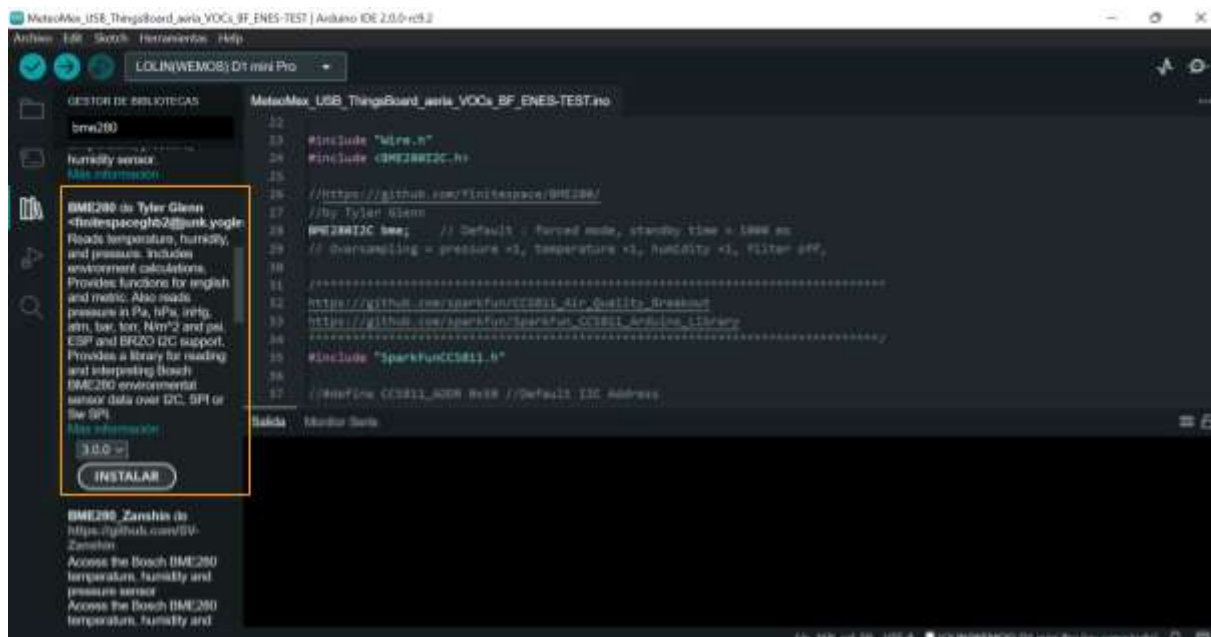
Agregar las dos librerías esenciales para utilizar MeteoMex Aeria

### A) BME280

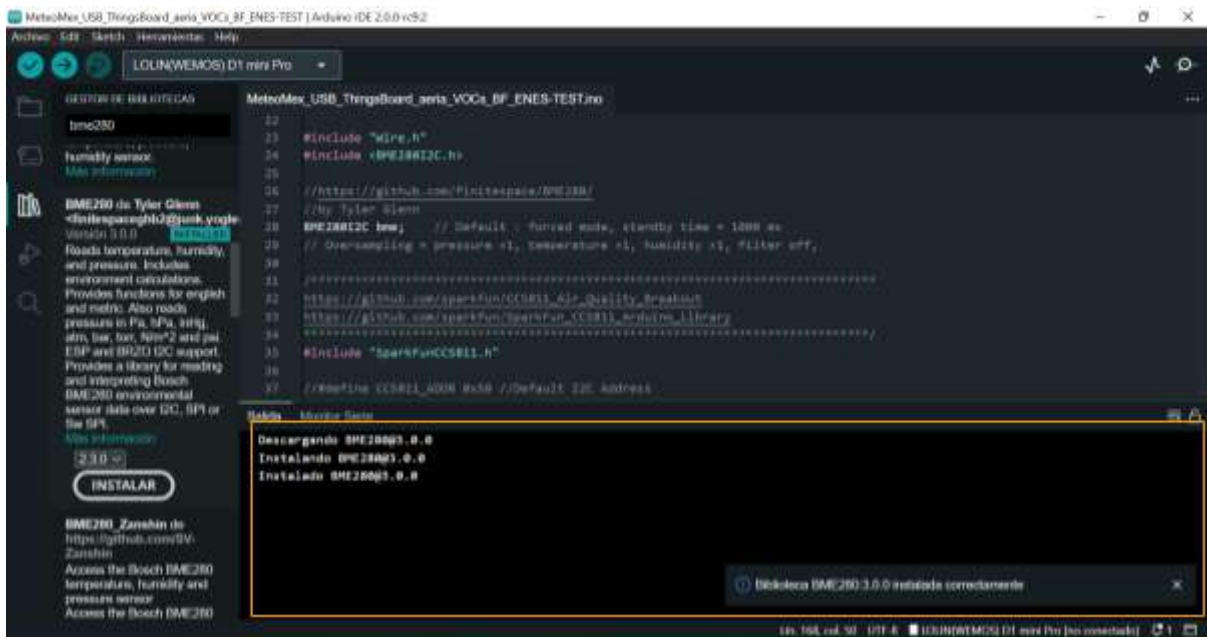
1. En el cuadro de búsqueda ingrese la palabra clave “BME280”.



2. Busque entre los resultados **BME280** de **Tyler Glenn**. Versión 3.0.0 y seleccione “Instalar”

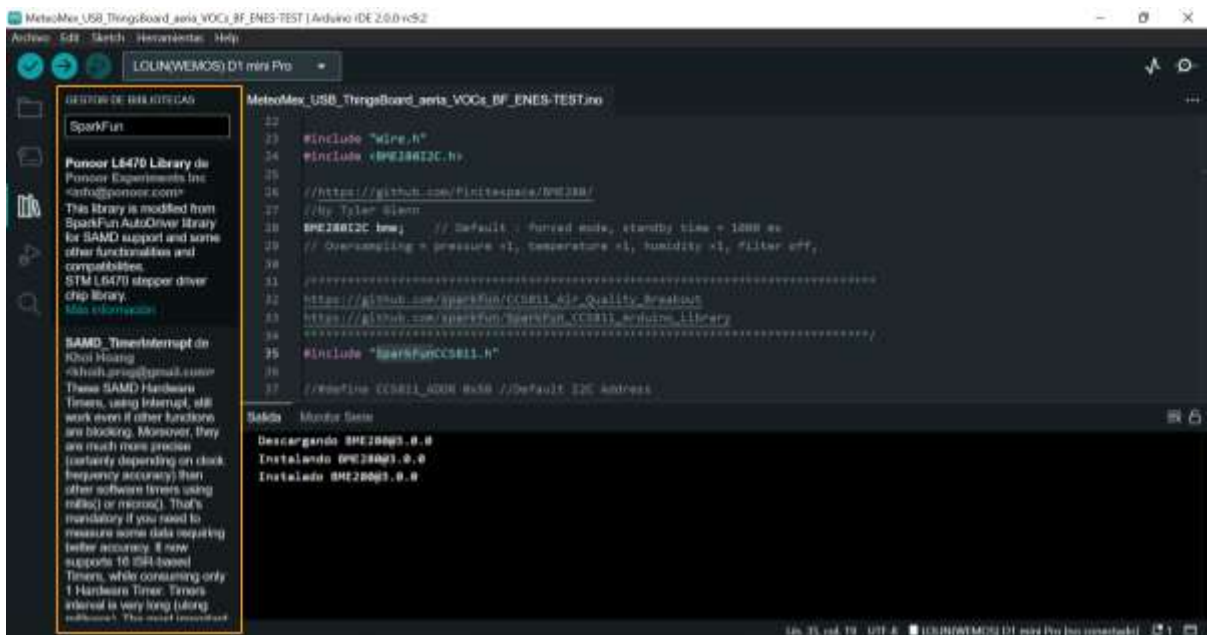


3. Una vez terminado el proceso y si no se presentaron errores, nos aparecerá en el área de salida un mensaje como el siguiente:



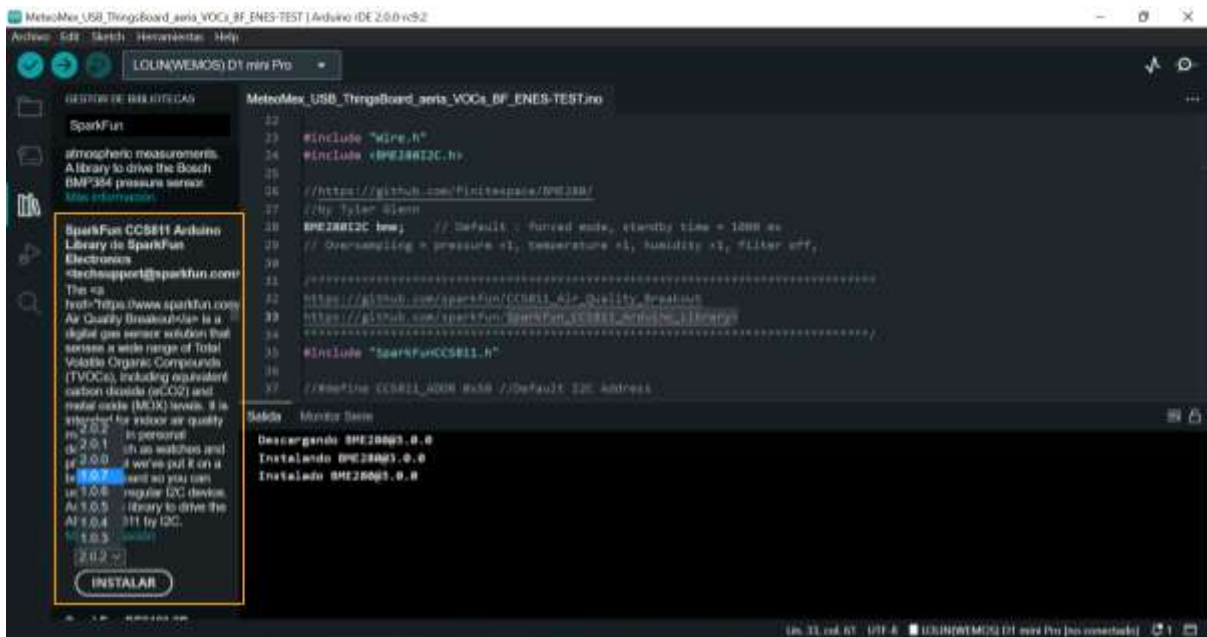
## B) SparkFun

1. En el cuadro de búsqueda ingrese la palabra clave "SparkFun".

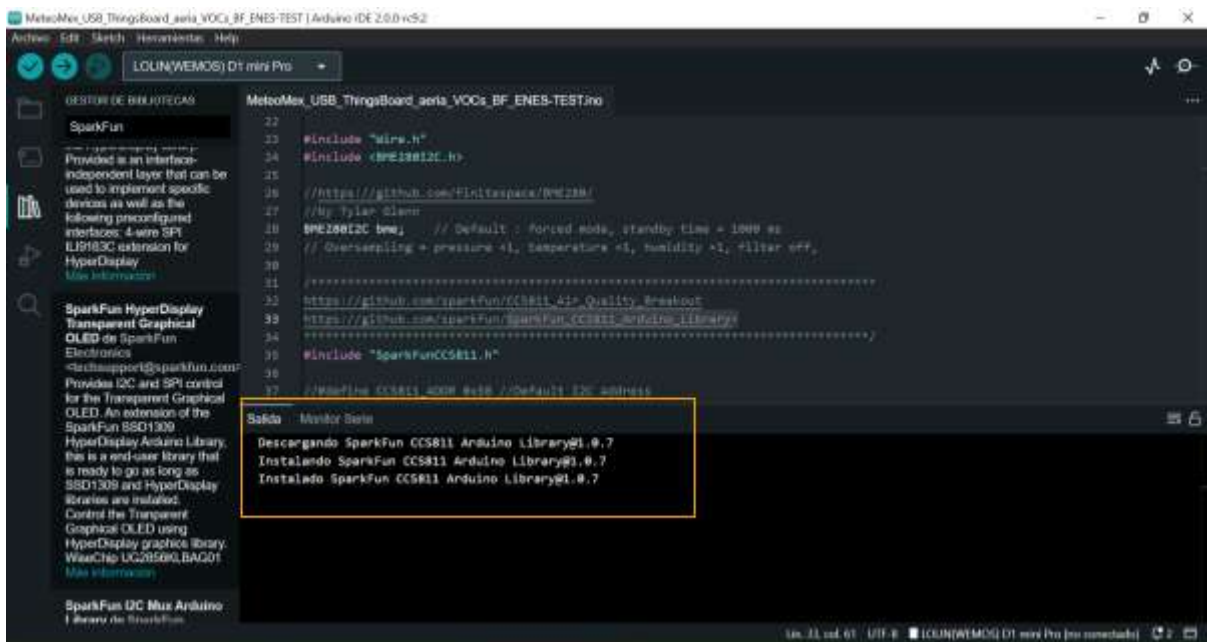


2. Busque entre los resultados **SparkFun CCS811 Arduino Library** de **SparkFun Electronics**. Versión 1.0.7 y seleccione "Instalar". Es muy importante instalar la versión 1.0.7 y NO la más reciente.



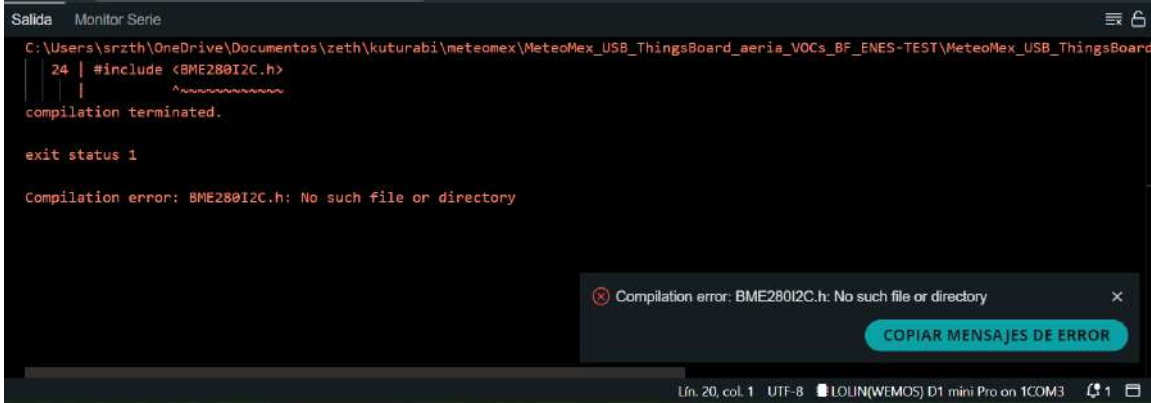


- Una vez terminado el proceso y si no se presentaron errores, nos aparecerá en el área de salida un mensaje como el siguiente:



## Error si no se instala alguna librería

En caso de no contar con alguna librería instalada, nos aparecerá un error similar al siguiente:



```
Salida Monitor Serie
C:\Users\srrzth\OneDrive\Documents\zeth\katurabi\meteoMex\MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST\MeteoMex_USB_ThingsBoard
24 | #include <BME280I2C.h>
    | ^~~~~~
compilation terminated.

exit status 1

Compilation error: BME280I2C.h: No such file or directory

Compilation error: BME280I2C.h: No such file or directory
COPIAR MENSAJES DE ERROR
Lín. 20, col. 1 UTF-8 LOLIN(WEMOS) D1 mini Pro on 1COM3
```

Para solucionarlo, es necesario seguir los pasos anteriormente mencionados para instalar BME280 o SparkFun.

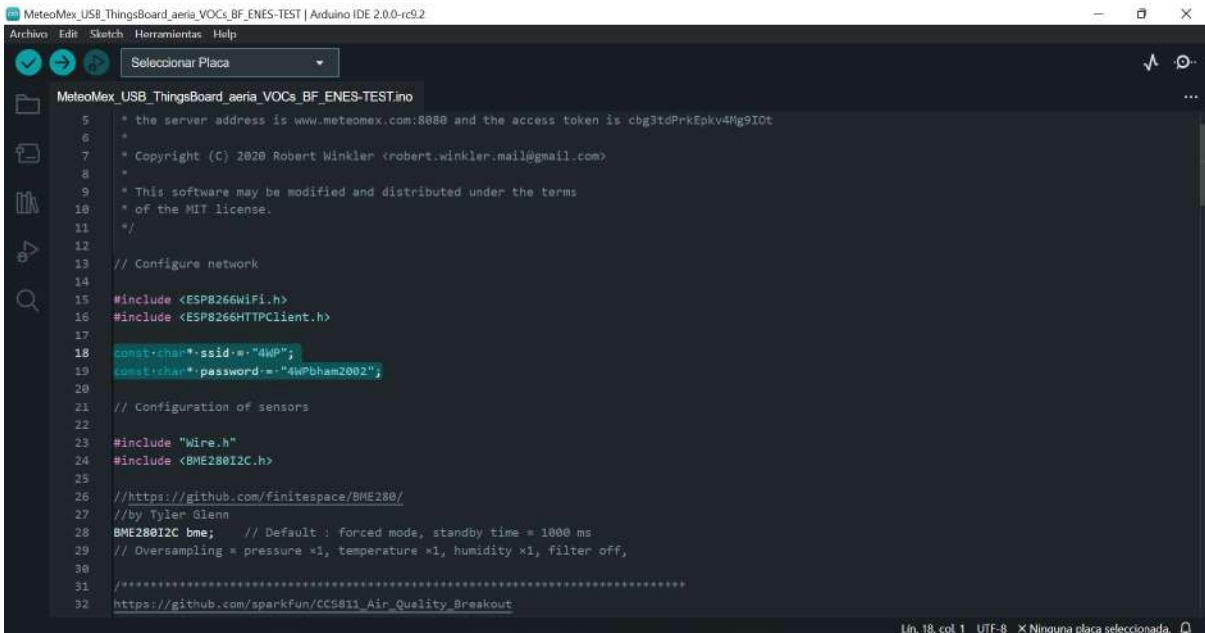
## Programación del dispositivo MeteoMex Aeria

Accede al archivo de programación adecuado para tu dispositivo, como el que se encuentra aquí:

[https://github.com/robert-winkler/MeteoMex/tree/master/MeteoMex\\_USB\\_ThingsBoard\\_aeria\\_VOCs](https://github.com/robert-winkler/MeteoMex/tree/master/MeteoMex_USB_ThingsBoard_aeria_VOCs)

### Contraseña y nombre del internet

Una vez que has abierto el archivo de código con Arduino IDE. El primer paso es cambiar los datos de conexión. Frente a **ssid** coloca el nombre de la red y frente a **password** la contraseña de la red de internet.



```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Editar Sketch Herramientas Help
Seleccionar Placa
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
5 * the server address is www.meteomex.com:8080 and the access token is cbg3tdPrkEpkv4Hg9I0t
6 *
7 * Copyright (C) 2020 Robert Winkler <robert.winkler.mail@gmail.com>
8 *
9 * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
17
18 const char* ssid = "4WP";
19 const char* password = "4WPbham2002";
20
21 // Configuration of sensors
22
23 #include "Wire.h"
24 #include <BME280I2C.h>
25
26 //https://github.com/finitespace/BME280/
27 //by Tyler Glenn
28 BME280I2C bme; // Default : forced mode, standby time = 1000 ms
29 // Oversampling = pressure *1, temperature *1, humidity *1, filter off,
30
31 /*****
32 https://github.com/sparkfun/CCS811_Air_Quality_Breakout
Lín. 18, col. 1 UTF-8 X Ninguna placa seleccionada.
```

En este ejemplo, la red se llama Red-Nt y la contraseña es 120822NtLvE



```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Edit Sketch Herramientas Help
LOLIN(WEMOS) D1 mini...
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
17
18 const char* ssid = "Red-It";
19 const char* password = "120812NtLvE";
20
21 // Configuration of sensors
22
23 #include "Wire.h"
24 #include <BME280I2C.h>
25
26 //https://github.com/Finitespace/BME280/
27 //by Tyler Glenn
28 BME280I2C bme; // Default : forced mode, standby time = 1000 ms
29 // Oversampling = pressure *1, temperature *1, humidity *1, filter off,
30
31 //*****
32 https://github.com/sparkfun/CCS811_Air_Quality_Breakout
33 https://github.com/sparkfun/SparkFun_CCS811_Arduino_Library
34 //*****
35 #include "SparkFunCCS811.h"
36
37 //define CCS811_ADDR 0x5B //Default I2C Address
38 #define CCS811_ADDR 0x5A //Alternate I2C Address
39
Lin 18, col 1 UTF-8 LOLIN(WEMOS) D1 mini Pro on 1COM3
```

### Verificar el token

El token le permite al dispositivo enviar la información al servidor. Este token es único para cada dispositivo y por ello es importante verificar que sea el correcto en los 4 lugares donde se invoca. Para corroborarlo, es necesario tener los datos de acceso proporcionados por MeteoMex. Dentro del código de programación del dispositivo, el token se encuentra en las líneas 4, 5, 93 y 98.

```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Edit Sketch Herramientas Help
LOLIN(WEMOS) D1 mini Pro
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
1 /*
2  * This program connects a MeteoMex aeria (http://www.meteomex.com) by WiFi and sends data to a ThingsBoard server
3  * In the ThingsBoard strings you have to use the ThingsBoard server address and the access token of your device
4  * e.g. http://www.meteomex.com:8080/api/v1/cbg3tdPrkEpkv4Mg0Y0R/telemetry
5  * the server address is www.meteomex.com:8080 and the access token is cbg3tdPrkEpkv4Mg0Y0R
6  *
7  * Copyright (C) 2020 Robert Winkler <robert.winkler.mail@gmail.com>
8  *
9  * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
17
```

```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Edit Sketch Herramientas Help
LOLIN(WEMOS) D1 mini Pro
SEARCH: BUSCAR
cbg3tdPrkEp Ae ab .*
Replace
4 resultados en 1 archivo ...
MeteoMex_USB_T...
... meteomex.com.80...
... and the access tok...
... meteomex.com.80...
... meteomex.com.80...
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
84 //start wifi connection
85 WiFi.begin(ssid, password);
86 while (WiFi.status() != WL_CONNECTED) {
87   delay(1000);
88   Serial.println("Connecting..");
89 }
90 Serial.println(WiFi.localIP());
91
92 //To check the connection and posting of data from your terminal
93 //curl -v -X POST -d '{"temperature":25}' http://www.meteomex.com:8080/api/v1/cbg3tdPrkEpky4Hg9IDt/telemetry --header "C
94
95 WiFiClient client;
96 HTTPClient http; //Declare object of class HTTPClient
97
98 http.begin(client, "http://www.meteomex.com:8080/api/v1/cbg3tdPrkEpky4Hg9IDt/telemetry"); //Specify request destinat
99 http.addHeader("Content-Type", "application/json"); //Specify content-type header
100
101 // Read BME280 data
102
103 float temp(NAN), hum(NAN), pres(NAN);
104
105 BME280::TempUnit tempUnit(BME280::TempUnit_Celsius);
106 BME280::PresUnit presUnit(BME280::PresUnit_Pa);
107
108 bme.read(pres, temp, hum, tempUnit, presUnit);
109
110 //UNCOMMENT FOR SERIAL OUTPUT
111 Serial.print("Temp: ");
112 Serial.println(temp);
113
```



MeteoMex\_USB\_ThingsBoard\_aeria\_VOCs\_BF\_ENES-TEST | Arduino IDE 2.0.0-rc9.2

Archivo Edit Sketch Herramientas Help

LOLIN(WEMOS) D1 mini...

```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
1  /*
2  * This program connects a MeteoMex aeria (http://www.meteomex.com) by WiFi and sends data to a ThingsBoard server
3  * In the ThingsBoard strings you have to use the ThingsBoard server address and the access token of your device
4  * e.g. http://www.meteomex.com:8080/api/v1/cbg3tdPrkEpkv4Mg9I0t/telemetry
5  * the server address is www.meteomex.com:8080 and the access token is cbg3tdPrkEpkv4Mg9I0t
6  *
7  * Copyright (C) 2020 Robert Winkler <robert.winkler.mail@gmail.com>
8  *
9  * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
```

Salida Monitor Serie x

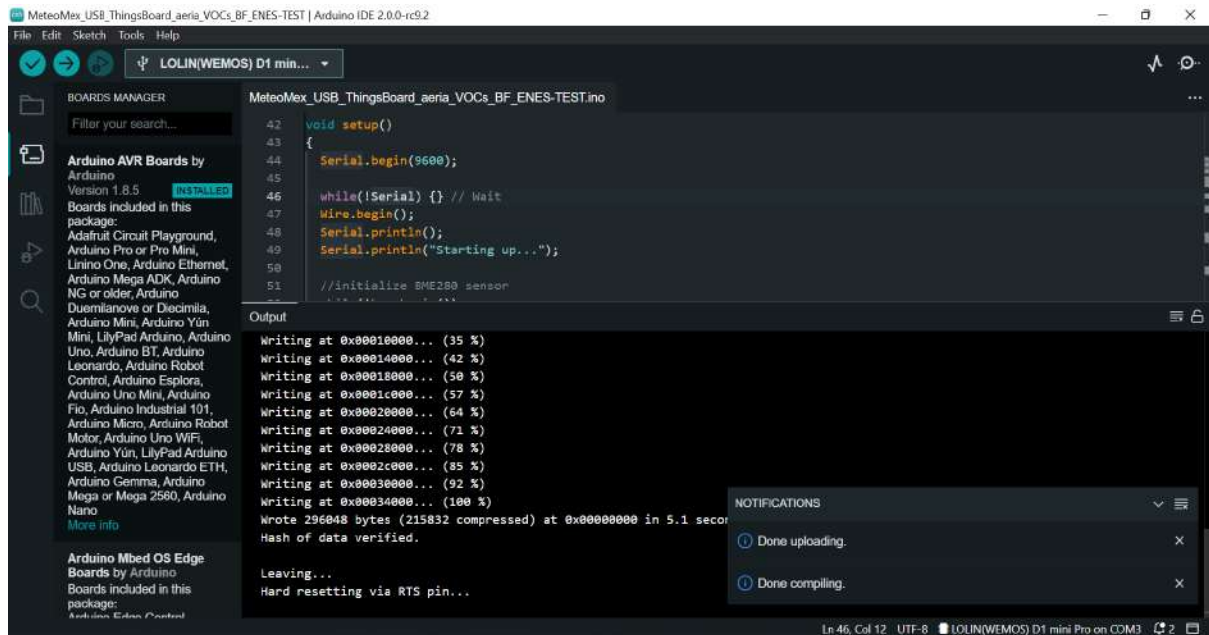
Mensaje (Ctrl + Enter para enviar el mensaje a 'LOLIN(WEMOS) D1 mini Pro' en 'COM3') Ambos NL & CR 115200 baud

```
13:3:38.028 -> Connecting..
13:3:39.018 -> Connecting..
13:3:40.751 -> Connecting..
13:3:41.744 -> Connecting..
13:3:42.751 -> Connecting..
13:3:42.751 -> 192.168.10.110
13:3:42.751 -> Temp: 31.09 °C Humidity: 27.06% RH Pressure: 83326.41 Pa
13:3:43.575 -> eCO2: 425 ppm tVOC:3 ppb
```

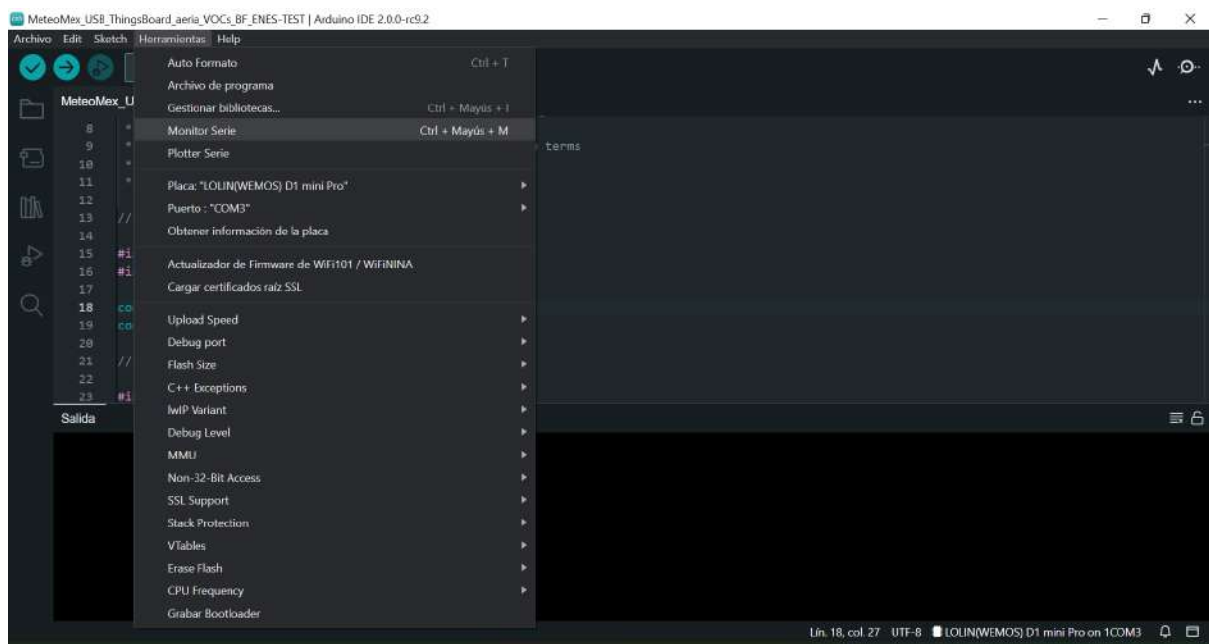
Lin. 19, col. 40 UTF-8 LOLIN(WEMOS) D1 mini Pro on 1COM3

## Hard resetting via RTS pin

Si durante la carga y compilación del código al dispositivo, aparece que el dispositivo se reseteara y no pasa nada más, no es necesario hacer nada. El dispositivo está listo para usarse.



## Abra Herramientas > Monitor Serie

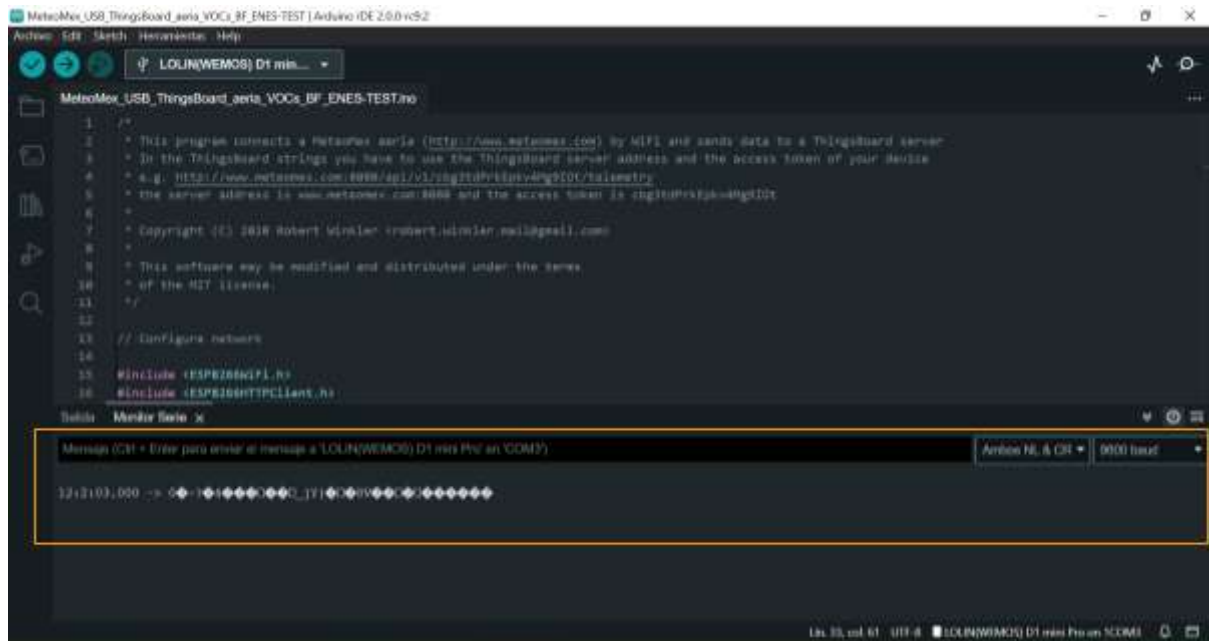


Y podrá observar que el dispositivo ya se encuentra en funcionamiento.

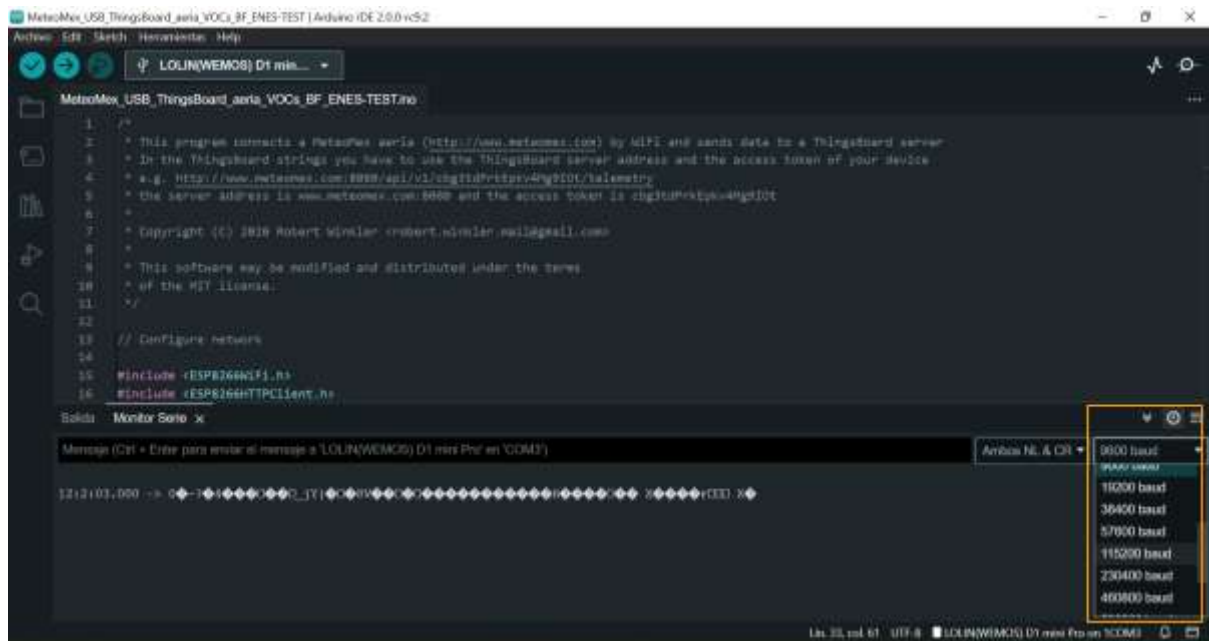
```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Edit Sketch Herramientas Help
ψ LOLIN(WEMOS) D1 mini...
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
8
9 * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
17
18 const char* ssid = "RED-NA";
19 const char* password = "ElMasPerronAqui";
20
21 // Configuration of sensors
22
23 #include "Wire.h"
Salida Monitor Serie x
Mensaje (Ctrl + Enter para enviar el mensaje a 'LOLIN(WEMOS) D1 mini Pro' en 'COM3') Ambos NL & CR 115200 baud
13:10:57.682 -> Connecting..
13:10:58.673 -> Connecting..
13:10:59.697 -> Connecting..
13:11:00.693 -> Connecting..
13:11:01.694 -> Connecting..
13:11:02.975 -> Connecting..
13:11:03.684 -> Connecting..
13:11:04.679 -> Connecting..
indexing: 8/84 Lln. 18, col. 27 UTF-8 LOLIN(WEMOS) D1 mini Pro on 1COM3
```

### Los datos de lectura tienen un formato no legible

Si al conectar el dispositivo de monitoreo MeteoMex Aeria, en el **Monitor Serie** no se visualizan los datos de manera correcta, es necesario cambiar la velocidad de transmisión. Este valor puede encontrarse por defecto en 9600.



Para modificarlo, en la parte inferior derecha, desplegamos la lista de selección de los baud. Y seleccionamos 115200



Posteriormente, conectamos el dispositivo de monitoreo MeteoMex Aeria y podremos ver como se enciende y conecta al servidor. Una vez conectado, nos aparecerá el IP.



```
MetroMox_USB_ThingsBoard_aria_VOCs_BF_ENES-TEST[Ardkoo IDE 2.0.0-rc2]
Arduio IDE: Sketch: Herencia2020 - Help
LOLH[WEMOS] 01 min...
MetroMox_USB_ThingsBoard_aria_VOCs_BF_ENES-TEST.ino
1 //
2 * This program connects a MetroMox ariia (http://www.metro-mox.com) by WiFi and sends data to a ThingsBoard server
3 * In the ThingsBoard strings you have to use the ThingsBoard server address and the access token of your device
4 * e.g. http://www.metro-mox.com:8080/api/v1/ing3tdhr4tspvdyng5D01/telemetry
5 * the server address is www.metro-mox.com:8080 and the access token is ing3tdhr4tspvdyng5D01
6 *
7 * Copyright (C) 2020 Robert Winkler (robert.winkler.mail@gmail.com)
8 *
9 * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
Monitor Serie x
Message (Ctrl + Enter) para enviar el mensaje a LOLH[WEMOS] 01 min (Por an 5,000)
Arduio NL & Ctrl 115200 baud
12:41:40.917 -> Starting up...
12:41:40.917 -> Found IMU260 sensor! Success.
12:41:41.902 -> Connecting...
12:41:42.903 -> Connected!
12:41:44.532 -> Connecting...
12:41:45.531 -> Connecting...
12:41:46.502 -> Connecting...
12:41:46.502 -> 192.168.10.110
```

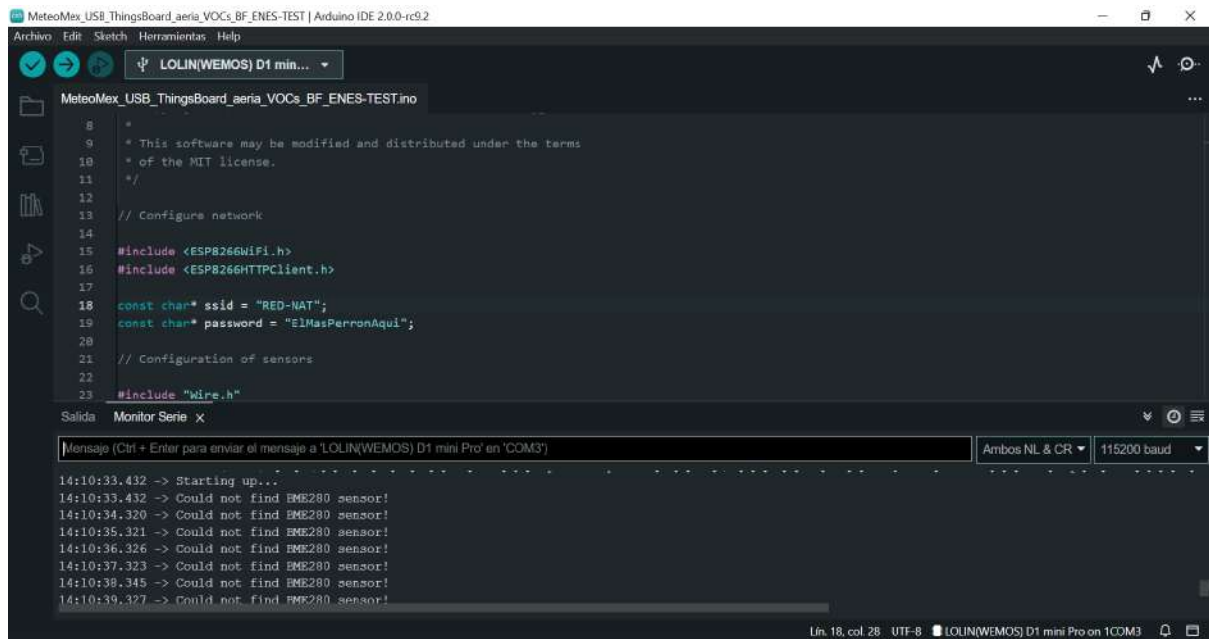
Asimismo nos mostrará las lecturas más recientes. Esto significa que nuestro dispositivo está funcionando correctamente.

```
MetroMox_USB_ThingsBoard_aria_VOCs_BF_ENES-TEST[Ardkoo IDE 2.0.0-rc2]
Arduio IDE: Sketch: Herencia2020 - Help
LOLH[WEMOS] 01 min...
MetroMox_USB_ThingsBoard_aria_VOCs_BF_ENES-TEST.ino
1 //
2 * This program connects a MetroMox ariia (http://www.metro-mox.com) by WiFi and sends data to a ThingsBoard server
3 * In the ThingsBoard strings you have to use the ThingsBoard server address and the access token of your device
4 * e.g. http://www.metro-mox.com:8080/api/v1/ing3tdhr4tspvdyng5D01/telemetry
5 * the server address is www.metro-mox.com:8080 and the access token is ing3tdhr4tspvdyng5D01
6 *
7 * Copyright (C) 2020 Robert Winkler (robert.winkler.mail@gmail.com)
8 *
9 * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
Monitor Serie x
Message (Ctrl + Enter) para enviar el mensaje a LOLH[WEMOS] 01 min (Por an 5,000)
Arduio NL & Ctrl 115200 baud
12:41:47.754 -> 192.168.10.110
12:41:47.799 -> Temp: 26.74 °C Humidity: 34.60% RH Pressure: 83364.71 Pa
12:41:52.277 -> CO2: 466 ppb TVOC:10 ppb
12:41:52.277 ->
12:41:52.821 -> 192.168.10.110
12:41:52.821 -> Temp: 26.70 °C Humidity: 35.72% RH Pressure: 83364.27 Pa
12:41:57.316 -> CO2: 805 ppb TVOC:73 ppb
12:41:57.316 ->
```



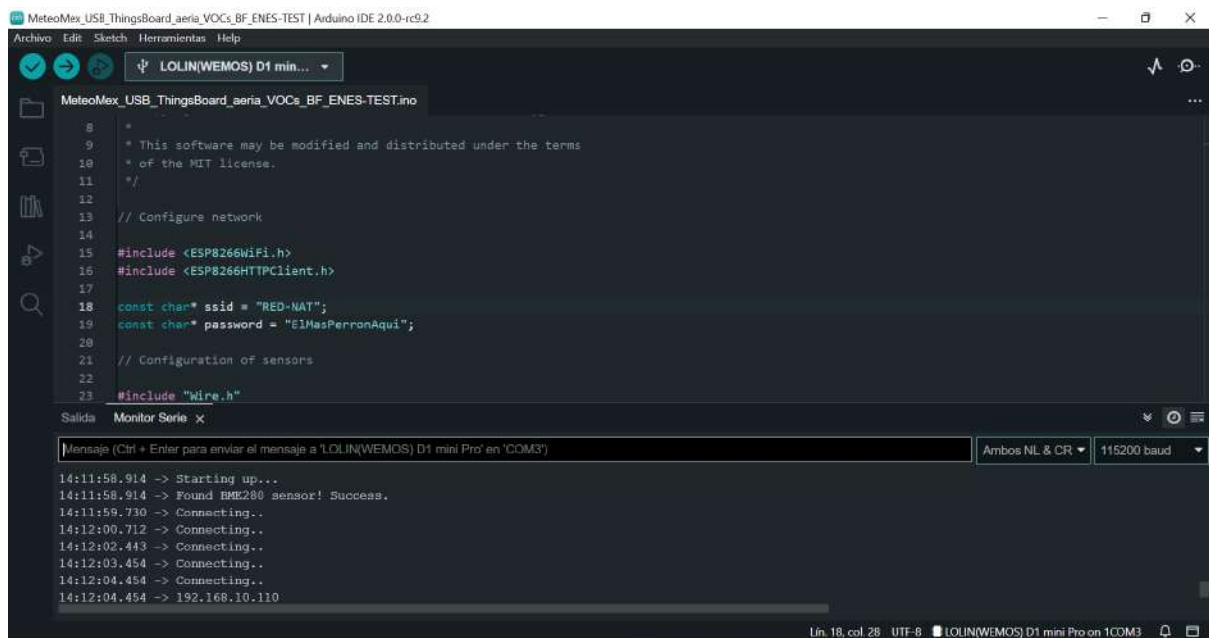
## No se encuentra el sensor BME280

Si el dispositivo Meteomex Aeria se encuentra conectado pero sin el shield con los sensores, nos arrojará el mensaje de que no se puede encontrar el sensor.



```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Edit Sketch Herramientas Help
LOLIN(WEMOS) D1 mini...
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
8 *
9 * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
17
18 const char* ssid = "RED-NAT";
19 const char* password = "ElMasPerronAqui";
20
21 // Configuration of sensors
22
23 #include "Wire.h"
Salida Monitor Serie x
Mensaje (Ctrl + Enter para enviar el mensaje a 'LOLIN(WEMOS) D1 mini Pro' en 'COM3') Ambos NL & CR 115200 baud
14:10:33.432 -> Starting up...
14:10:33.432 -> Could not find BME280 sensor!
14:10:34.320 -> Could not find BME280 sensor!
14:10:35.321 -> Could not find BME280 sensor!
14:10:36.326 -> Could not find BME280 sensor!
14:10:37.323 -> Could not find BME280 sensor!
14:10:38.345 -> Could not find BME280 sensor!
14:10:39.327 -> Could not find BME280 sensor!
Lín. 18, col. 28 UTF-8 LOLIN(WEMOS) D1 mini Pro on 1COM3
```

La solución es desconectar el dispositivo del puerto USB, verificar que el shield con los sensores se encuentra colocado correctamente y volver a conectar el dispositivo. Posteriormente nos arrojará el mensaje de que se encontró el sensor exitosamente.



```
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST | Arduino IDE 2.0.0-rc9.2
Archivo Edit Sketch Herramientas Help
LOLIN(WEMOS) D1 mini...
MeteoMex_USB_ThingsBoard_aeria_VOCs_BF_ENES-TEST.ino
8 *
9 * This software may be modified and distributed under the terms
10 * of the MIT license.
11 */
12
13 // Configure network
14
15 #include <ESP8266WiFi.h>
16 #include <ESP8266HTTPClient.h>
17
18 const char* ssid = "RED-NAT";
19 const char* password = "ElMasPerronAqui";
20
21 // Configuration of sensors
22
23 #include "Wire.h"
Salida Monitor Serie x
Mensaje (Ctrl + Enter para enviar el mensaje a 'LOLIN(WEMOS) D1 mini Pro' en 'COM3') Ambos NL & CR 115200 baud
14:11:58.914 -> Starting up...
14:11:58.914 -> Found BME280 sensor! Success.
14:11:59.730 -> Connecting..
14:12:00.712 -> Connecting..
14:12:02.443 -> Connecting..
14:12:03.454 -> Connecting..
14:12:04.454 -> Connecting..
14:12:04.454 -> 192.168.10.110
Lín. 18, col. 28 UTF-8 LOLIN(WEMOS) D1 mini Pro on 1COM3
```