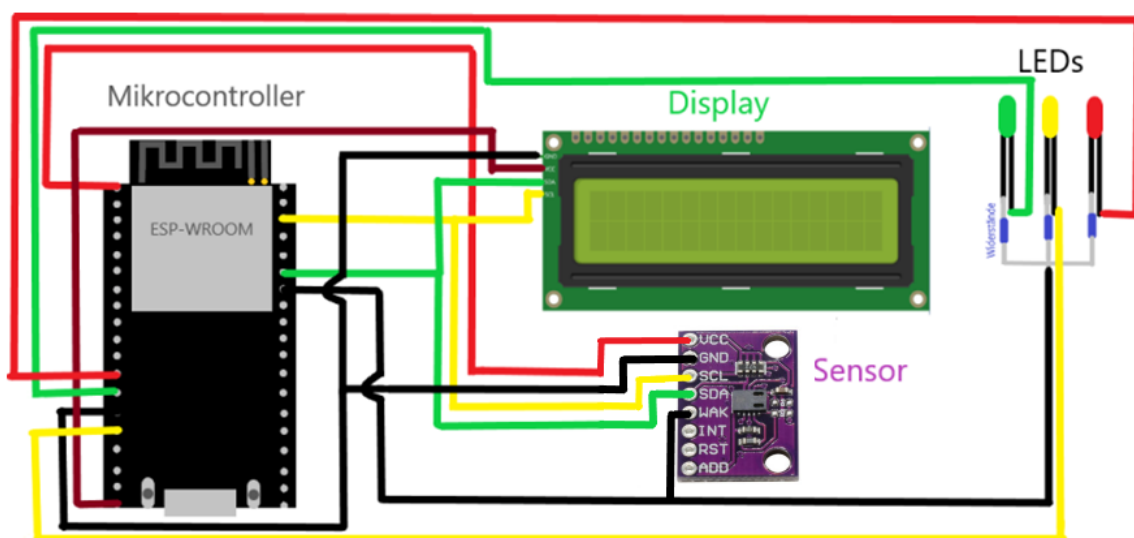


# CO<sub>2</sub> Ampel

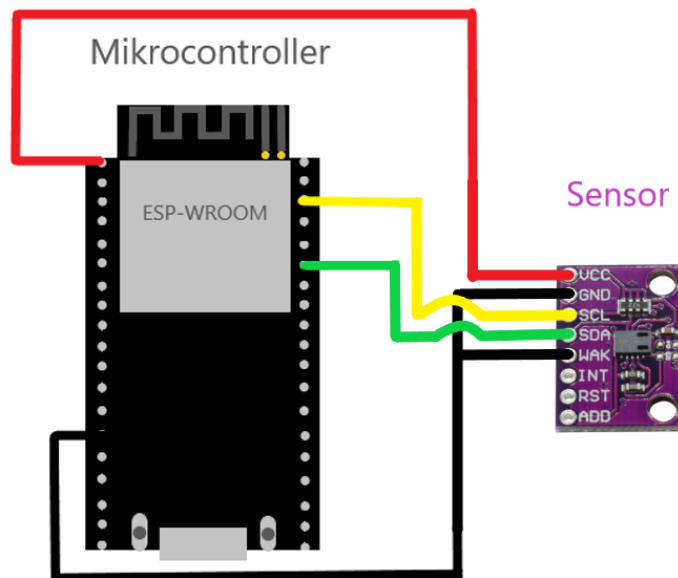
## *Anleitung Aufspielen Software*

Wir beginnen mit dem Schaltplan und dem zusammenstecken der Kabel. In der Übersicht ist der Schaltplan abgebildet. Entsprechend der folgenden Abbildungen folgen die Verbindung Mikrocontroller und Sensor, Mikrocontroller und Display sowie Mikrocontroller und LEDs. Bitte genau auf die Verbindungen achten!

### Übersicht

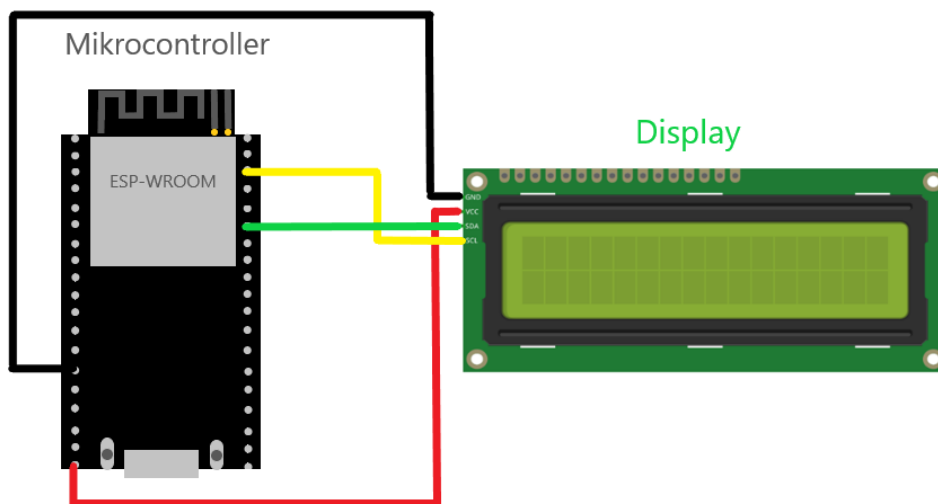


## Verbindungen Mikrocontroller & Sensor



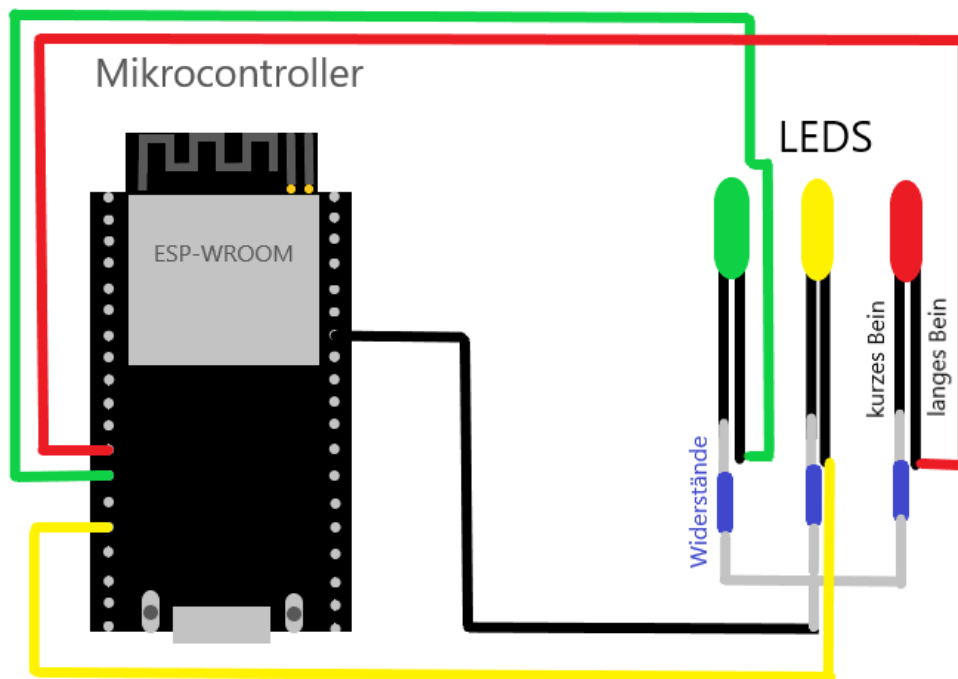
Kabelfarbe	Mikrocontroller	Sensor
Rot	3,3V	VCC
Schwarz	GND	GND & WAK
Grün	PIN 21	SDA
Gelb	PIN 22	SCL

## Verbindungen Mikrocontroller & Display



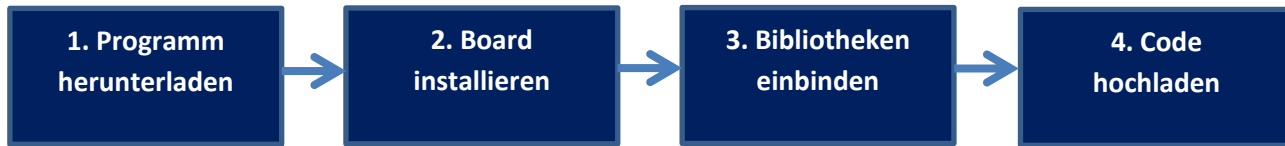
Kabelfarbe	Mikrocontroller	Display
Rot	5V	VCC
Schwarz	GND	GND
Grün	PIN 21	SDA
Gelb	PIN 22	SCL

## Verbindungen Mikrocontroller & LEDs



Kabelfarbe	Mikrocontroller	LEDs
Rot	14	Rote LED (langes Bein)
Grün	12	Grüne LED (langes Bein)
Gelb	13	Gelbe LED (langes Bein)
Schwarz	GND	Über die Widerstände an die kurzen Beine der LEDs

## Software



### 1. Programm herunterladen

Zunächst benötigt ihr das Programm Arduino IDE. Ihr könnt das Programm unter dem folgenden Link herunterladen und installieren:

#### Arduino - Software

Unter dem Link öffnet sich die Seite von Arduino. Wählt in dem dem türkisen Kasten die entsprechende Version für euren Rechner aus und installiert das Programm.



**ARDUINO 1.8.13**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board. Refer to the [Getting Started](#) page for installation instructions.

**Windows** Installer, for Windows 7 and up  
**Windows** ZIP file for non admin install

**Windows app** Requires Win 8.1 or 10  
[Get](#)

**Mac OS X** 10.10 or newer

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

[Release Notes](#)  
[Source Code](#)  
[Checksums \(sha512\)](#)

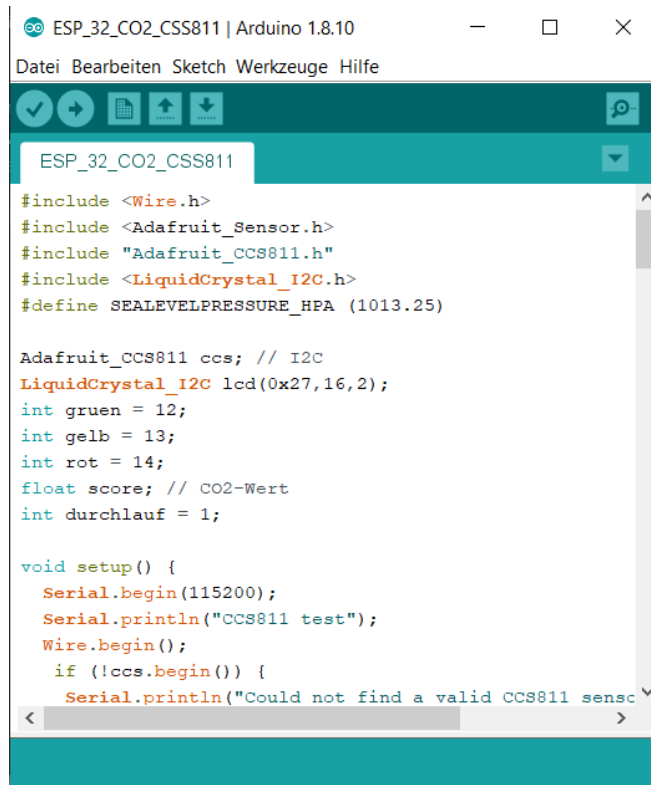
Hinweis: Klickt einfach auf „**just download**“

Falls ihr Hilfe bei der Installation benötigt, gibt es auch eine kurze Anleitung auf der Seite wie ihr das Programm für Windows installiert:

#### Arduino - Windows

## 2. Board installieren

Öffnet jetzt die Datei: „ESP\_CO2\_CCS811“ aus dem ZIP Ordner (Datei haben wir euch geschickt). Es öffnet sich ein Fenster mit der Frage, ob ihr die Datei in einen Sketch-Ordner verlegen möchtet- bestätigt mit OK. Danach sollte sich folgende Oberfläche öffnen.



```

ESP_32_CO2_CCS811 | Arduino 1.8.10
Datei Bearbeiten Sketch Werkzeuge Hilfe

ESP_32_CO2_CCS811

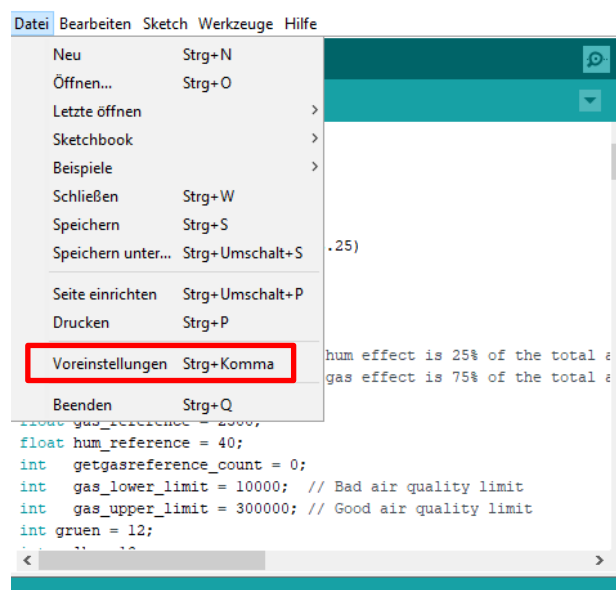
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include "Adafruit_CCS811.h"
#include <LiquidCrystal_I2C.h>
#define SEALEVELPRESSURE_HPA (1013.25)

Adafruit_CCS811 ccs; // I2C
LiquidCrystal_I2C lcd(0x27,16,2);
int gruen = 12;
int gelb = 13;
int rot = 14;
float score; // CO2-Wert
int durchlauf = 1;

void setup() {
  Serial.begin(115200);
  Serial.println("CCS811 test");
  Wire.begin();
  if (!ccs.begin()) {
    Serial.println("Could not find a valid CCS811 sensc

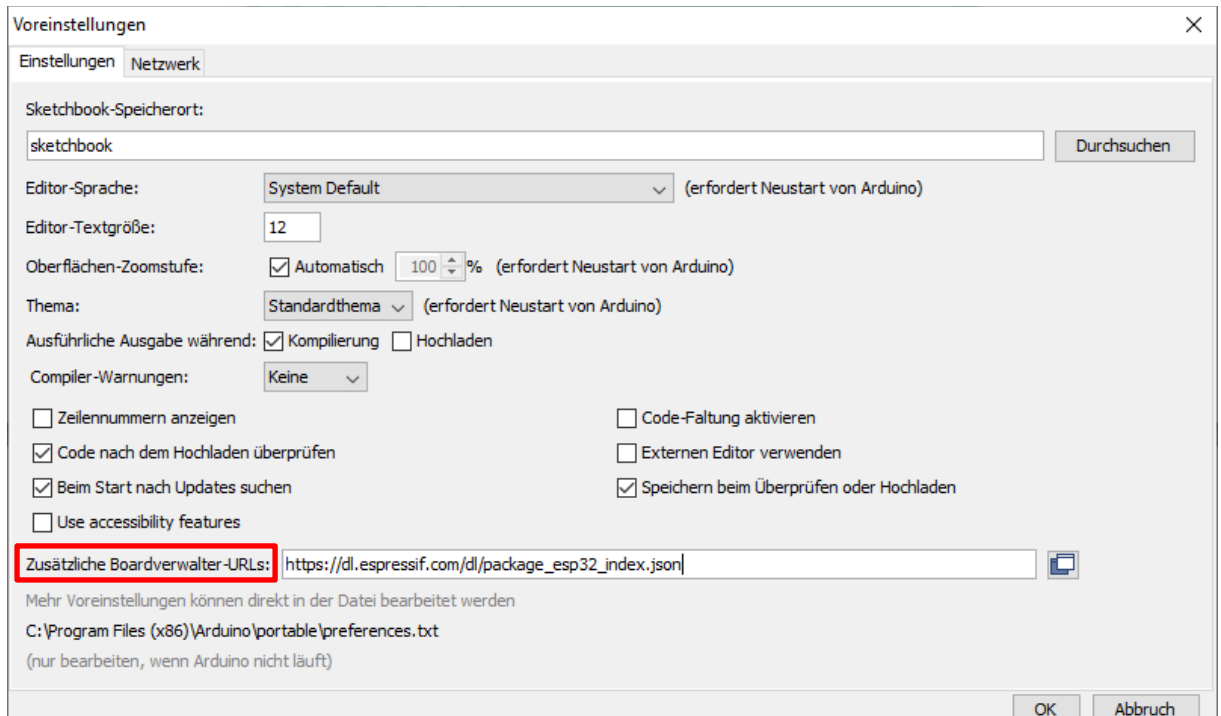
```

Bevor ihr den Code auf den Mikrocontroller laden könnt, müsst ihr zunächst das richtige Board einbinden. Dazu wählt ihr im Menü: Datei -> Voreinstellungen aus.

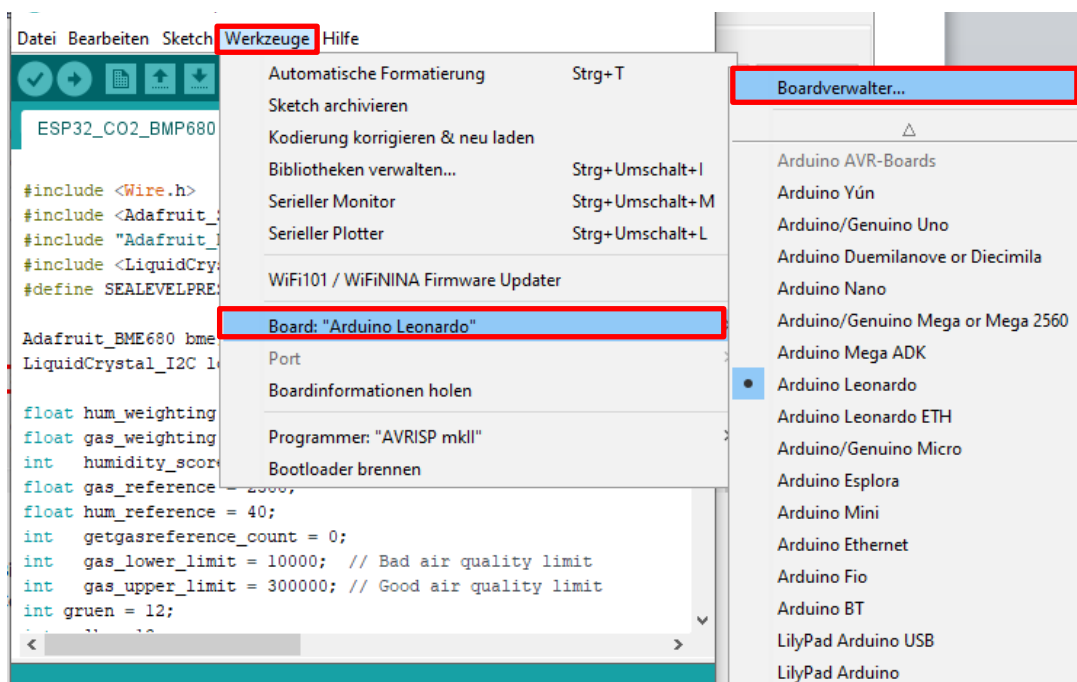


Es öffnet sich das folgende Fenster. Fügt in dem Feld „Zusätzliche Boardverwalter-URLs:“ den Link wie unten im Bild gezeigt ein und bestätigt mit OK.

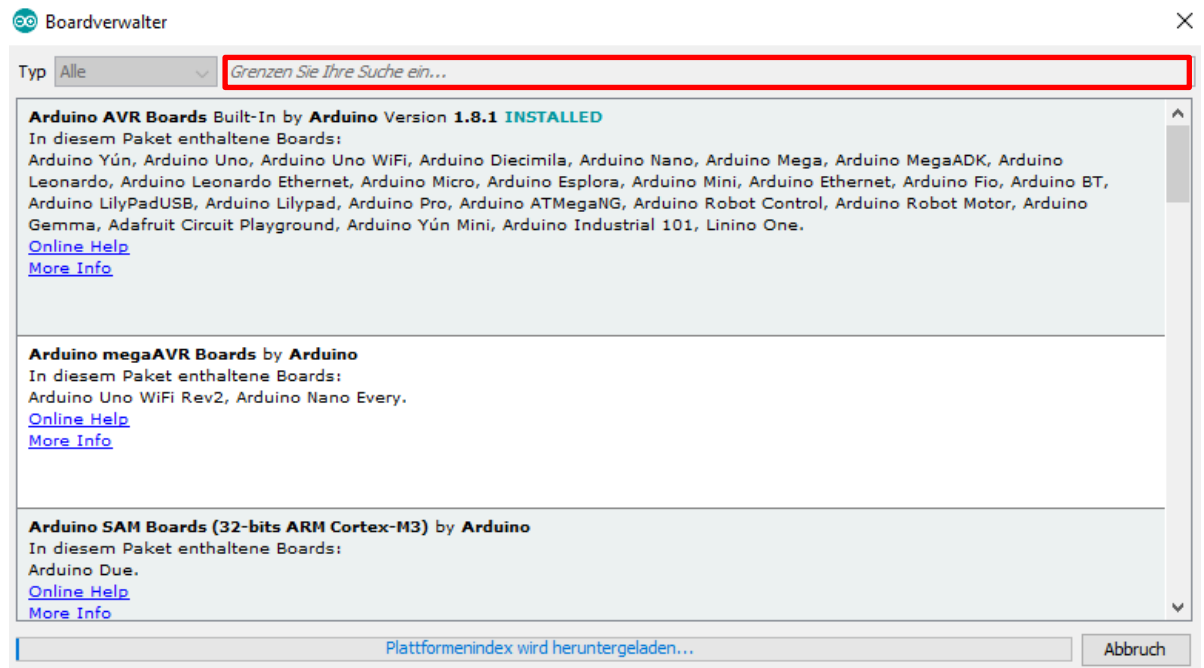
**[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)**



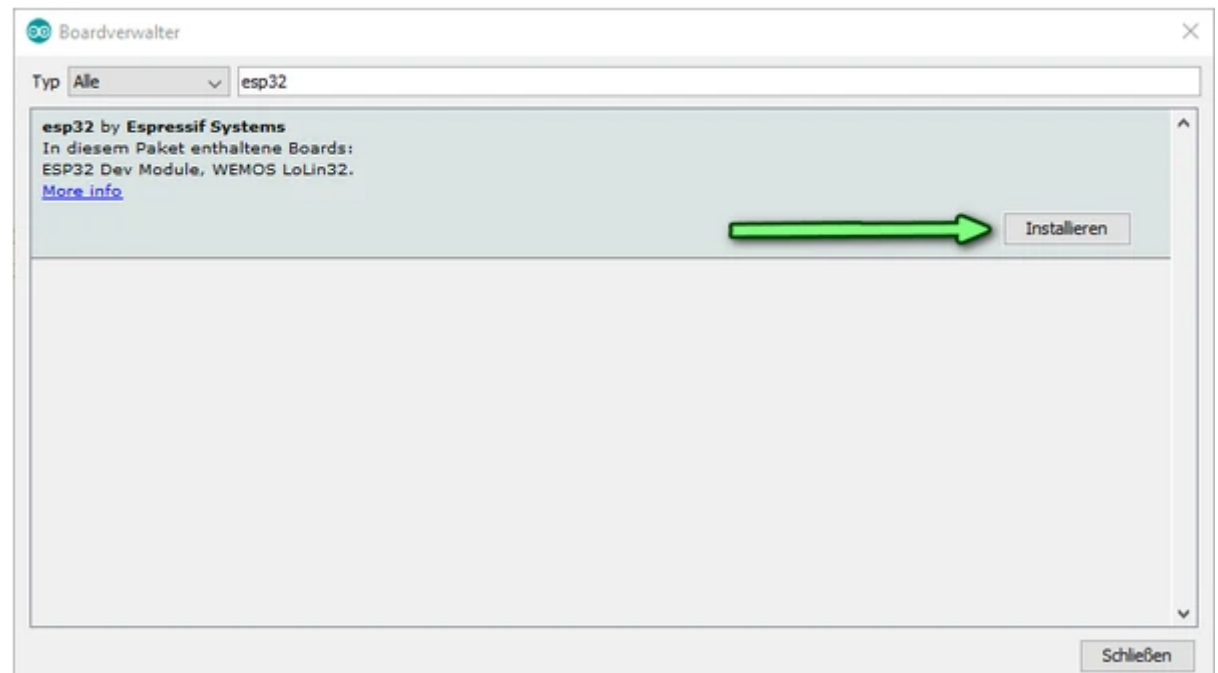
Im nächsten Schritt öffnet ihr unter Werkzeuge -> Board -> Boardverwalter.



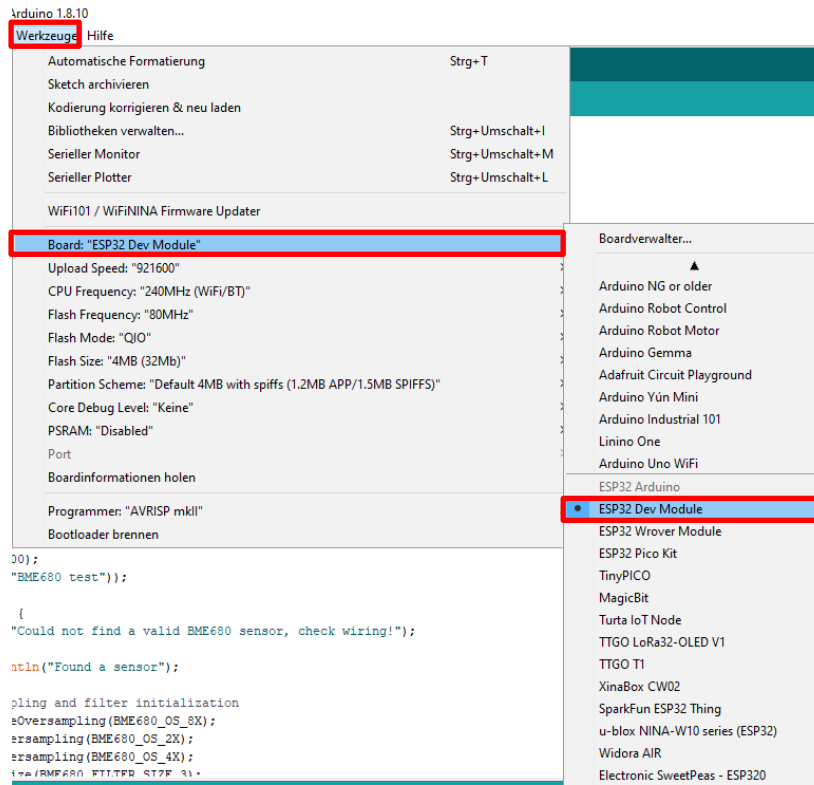
Es öffnet sich das Fenster vom Boardverwalter. In der Suchleiste gebt ihr den Begriff: „ESP32“ ein.



In den Suchergebnissen erscheint das Paket vom ESP32. Zum Bestätigen klickt ihr auf „Installieren“. Dies könnte einen Moment dauern.

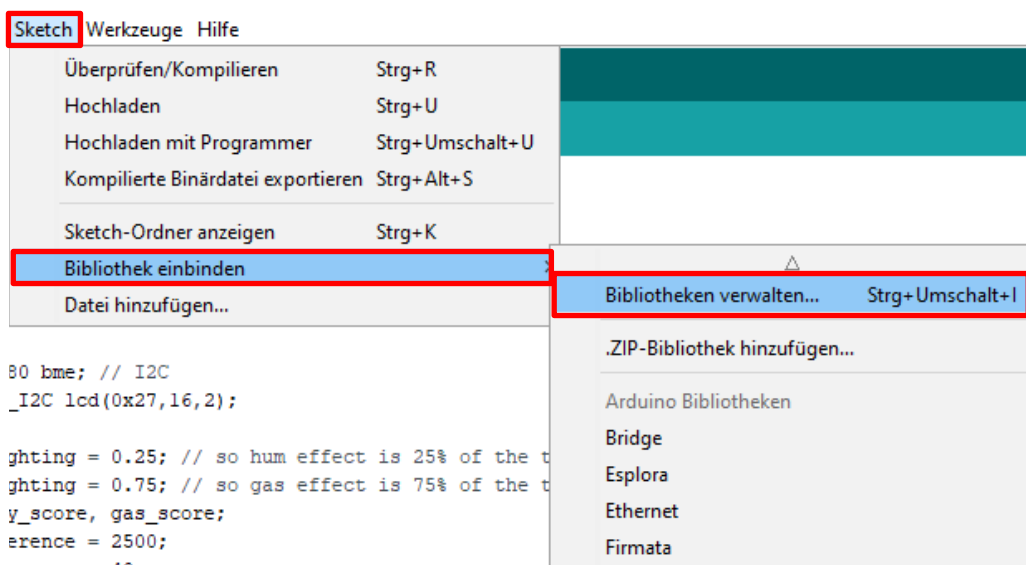


Danach öffnet ihr erneut unter Werkzeuge -> Board und sucht nach dem „ESP32 Dev Modul“. Wählt dieses Board aus und kehrt zur Oberfläche zurück.



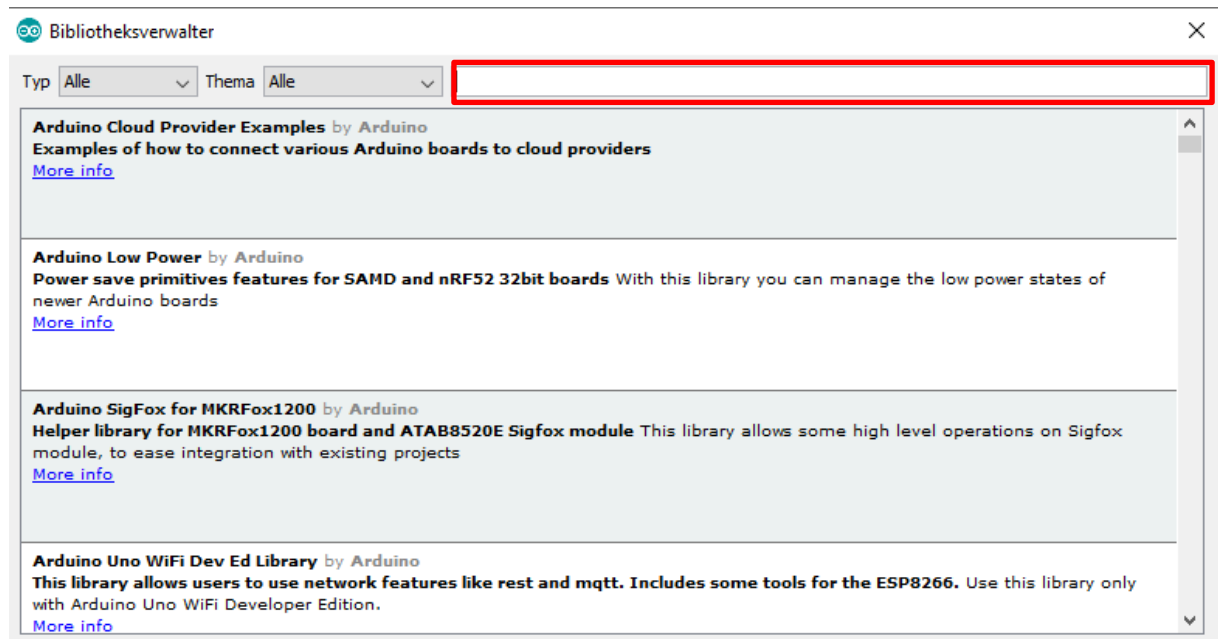
### 3. Bibliotheken einbinden

Bevor ihr den Code auf den Mikrocontroller (dem ESP32) implementieren könnt, ist die Installation von zwei Bibliotheken notwendig. Wählt dazu im Menü: Sketch -> Bibliotheken einbinden -> Bibliotheken verwalten aus.

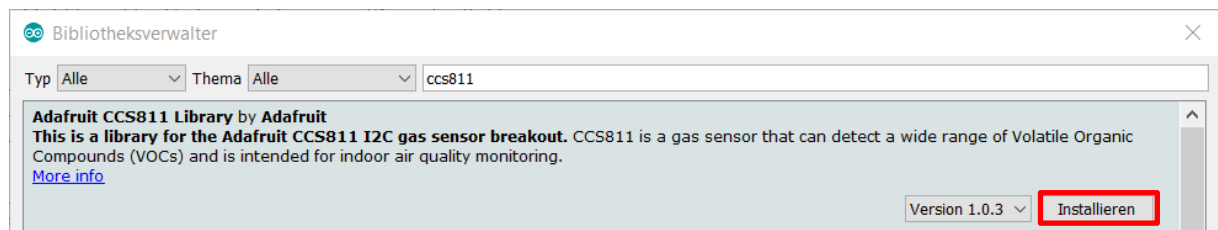




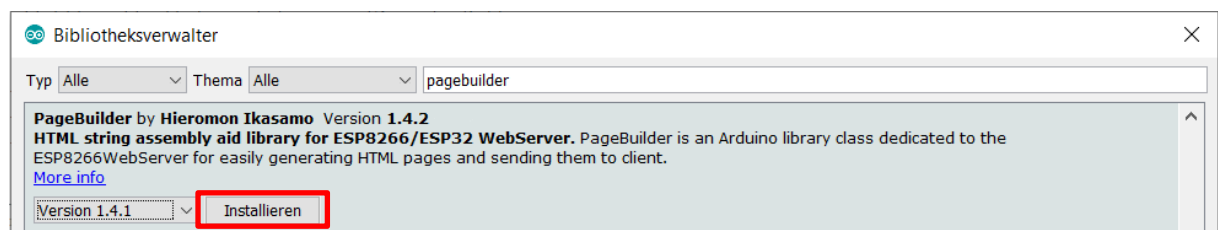
Es öffnet sich der Bibliotheksverwalter. In der Suchleiste gebt ihr den Namen für die Bibliothek vom Sensor ein. Gebt folgendes in der Suchleiste ein: „ccs811“



Bei der Suche nach „ccs811“, sollte dieses Ergebnis erscheinen. Bestätigt den Vorgang, indem ihr auf „Installieren“ klickt.

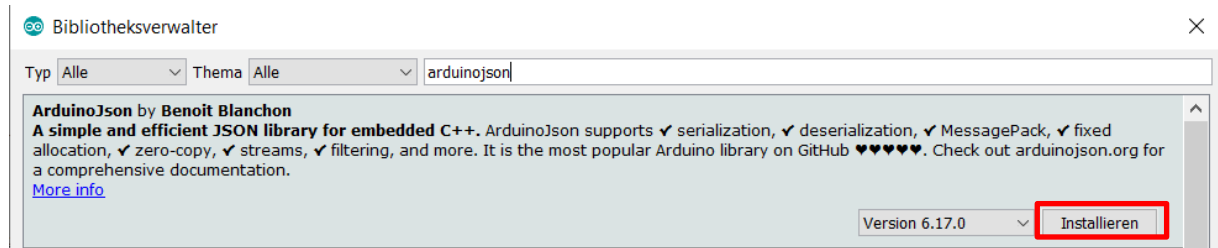


Die nächste Bibliothek die in die Suchleiste eingeben werden muss, ist „Pagebuilder“.



Bei der Suche nach „Pagebuilder“, sollte dieses Ergebnis erscheinen. Bestätigt den Vorgang erneut, indem ihr auf „Installieren“ klickt.

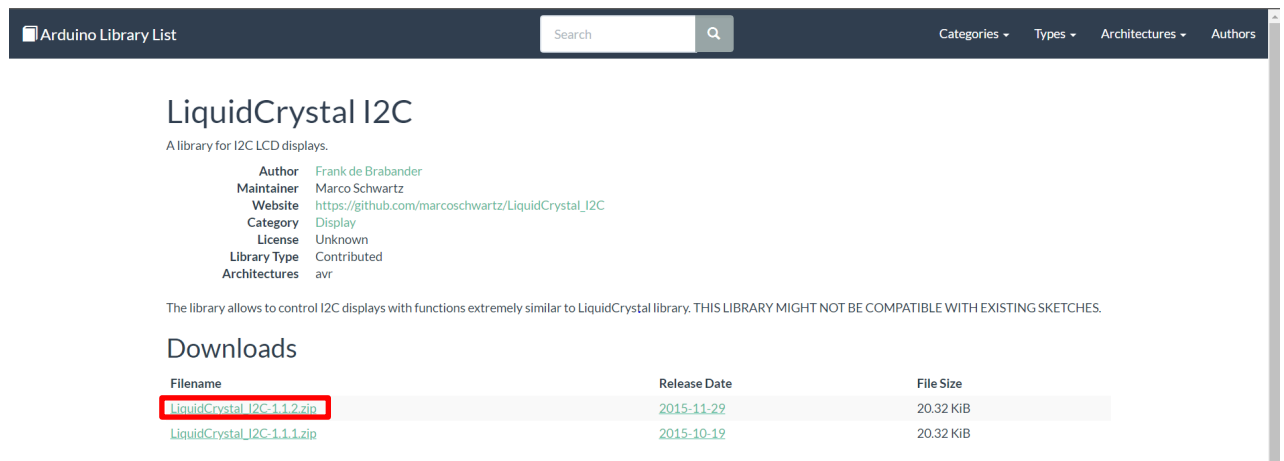
Der nächste Begriff für die Suchleiste ist: „ArduinoJson“. Bestätigt auch hier die Auswahl, indem ihr auf „Installieren“ klickt.



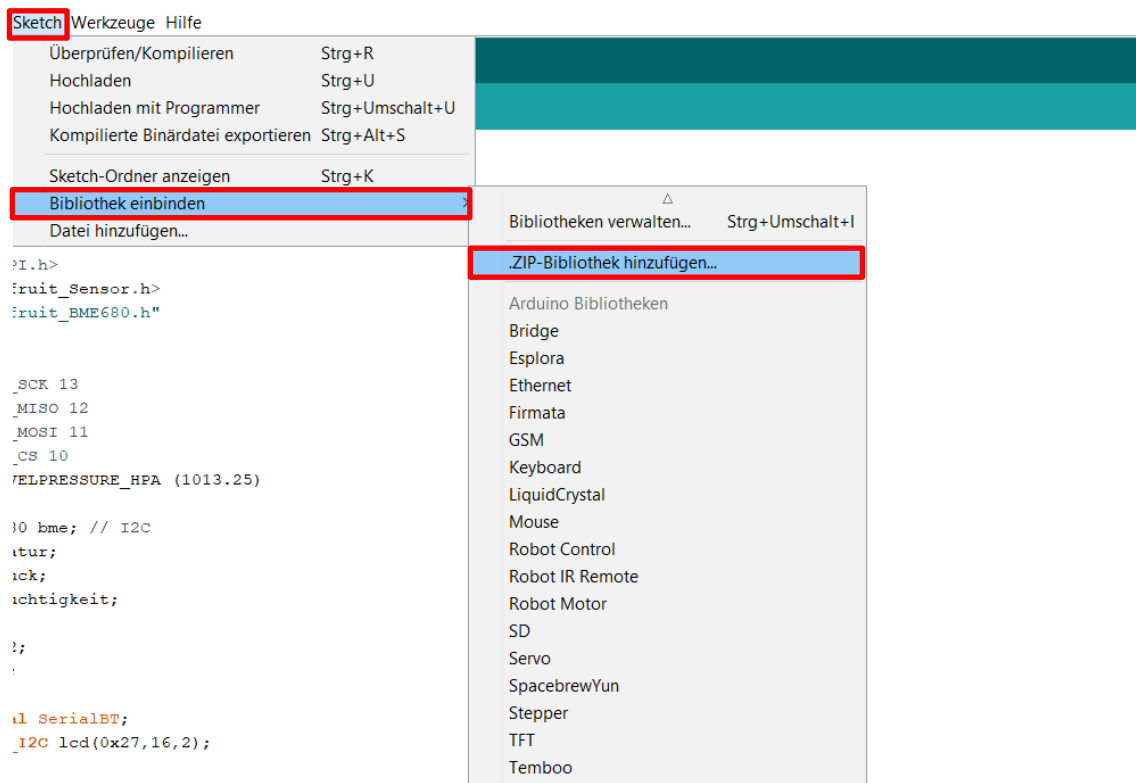
Für die Installation der LiquidCrystal Bibliothek ruft ihr folgenden Link auf:

**<https://www.arduinelibraries.info/libraries/liquid-crystal-i2-c>**

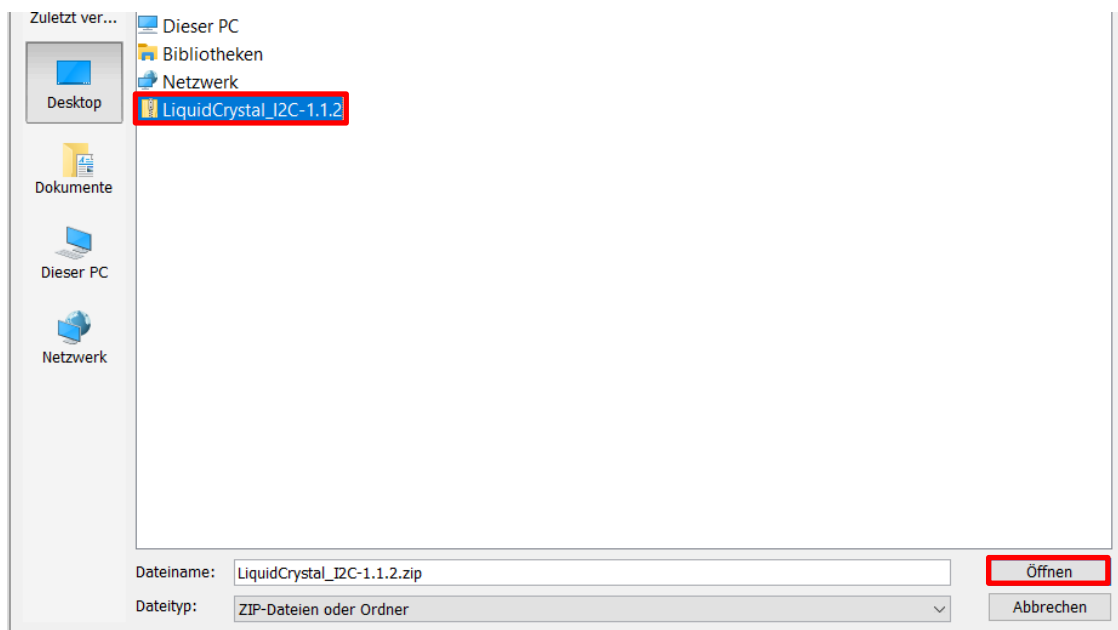
Auf der Seite ladet ihr die markierte ZIP-Datei herunter und speichert diese auf dem Desktop eures PCs ab.



Öffnet nun im Programm: Sketch -> Bibliotheken einbinden -> .ZIP Bibliothek hinzufügen aus.



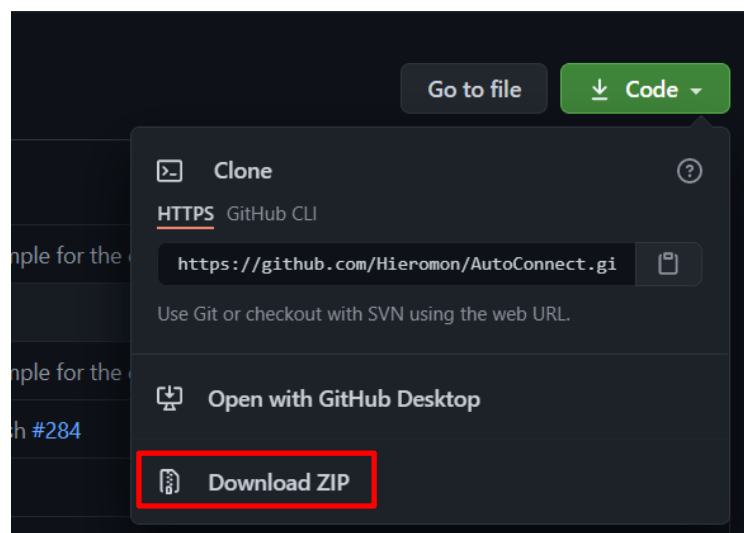
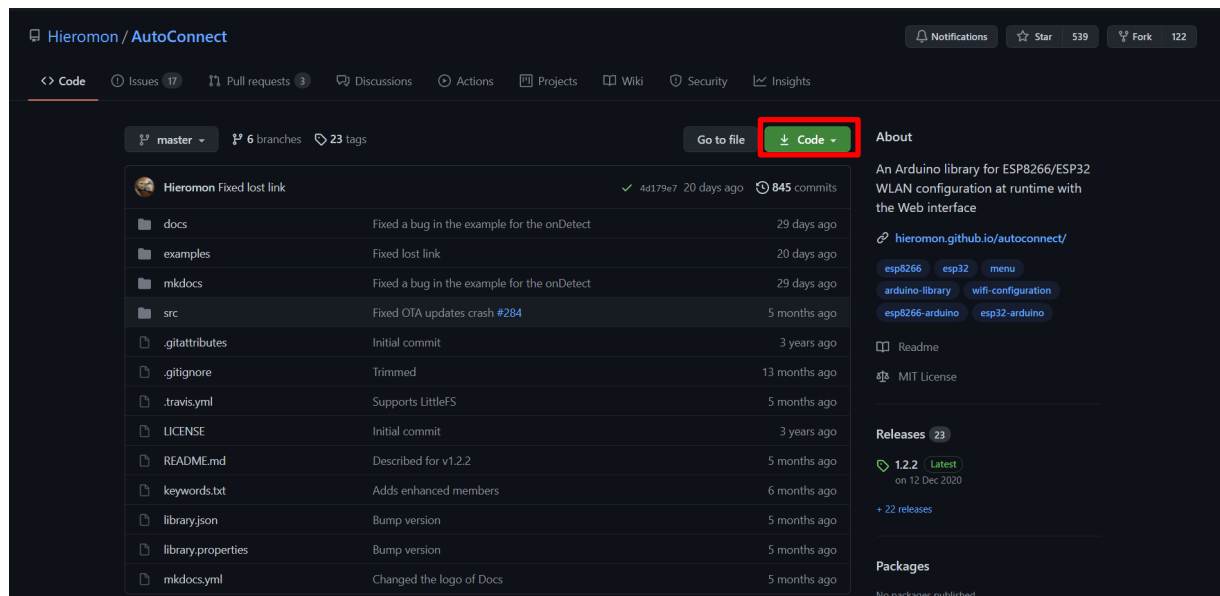
Es sollte sich folgendes Fenster öffnen, in diesem sucht ihr den Ort auf dem Desktop an dem ihr im vorherigen Schritt die Bibliothek abgespeichert habt. Dann klickt ihr auf den ZIP-Ordner und bestätigt mit „öffnen“.



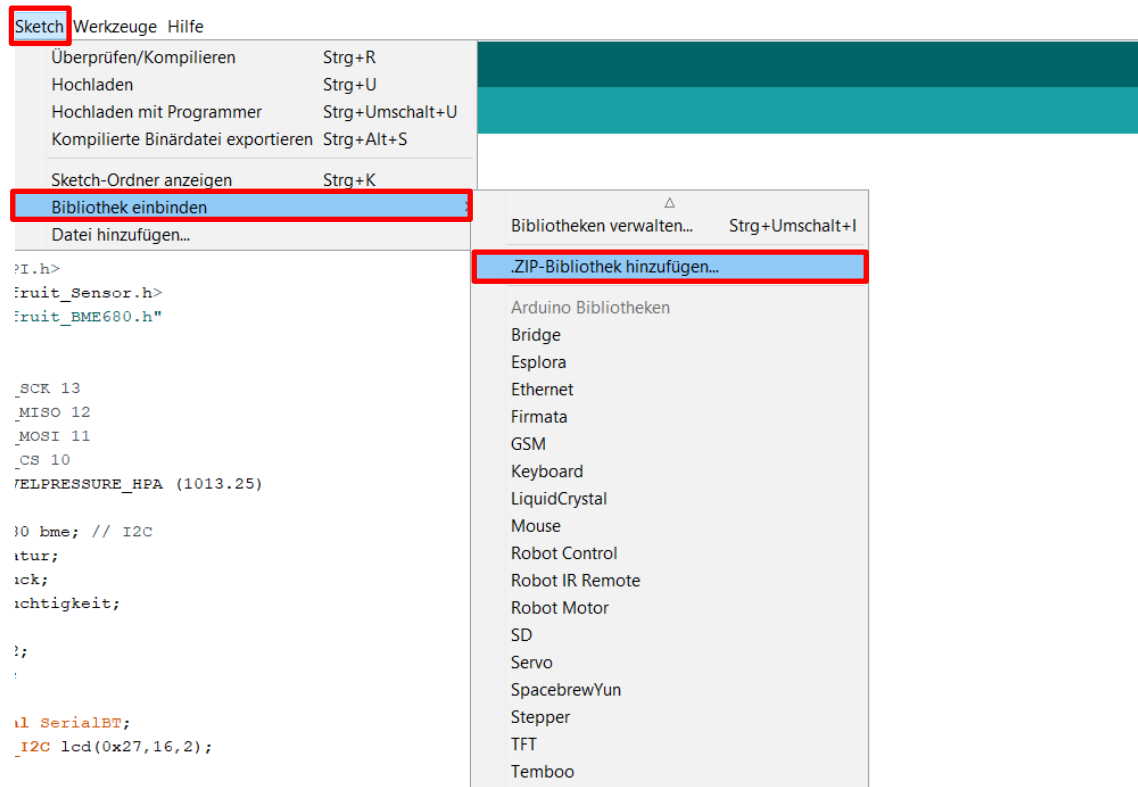
Für die letzte Bibliothek gehen wir ähnlich vor wie bei der LiquidCrystal Bibliothek. Ruft als erstes den folgenden Link auf:

**<https://github.com/Hieromon/AutoConnect>**

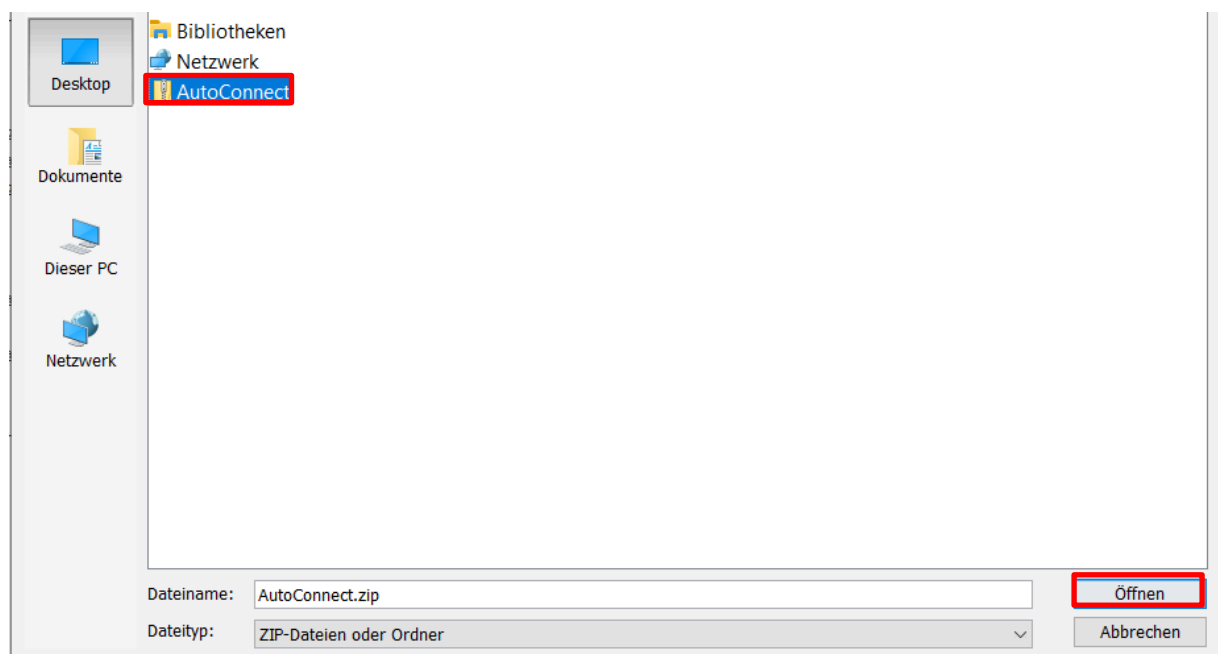
Auf der Seite klickt ihr auf Code, ladet ihr die markierte ZIP-Datei herunter und speichert diese auf dem Desktop eures PCs ab.



Öffnet nun im Programm: Sketch -> Bibliotheken einbinden -> .ZIP Bibliothek hinzufügen aus.

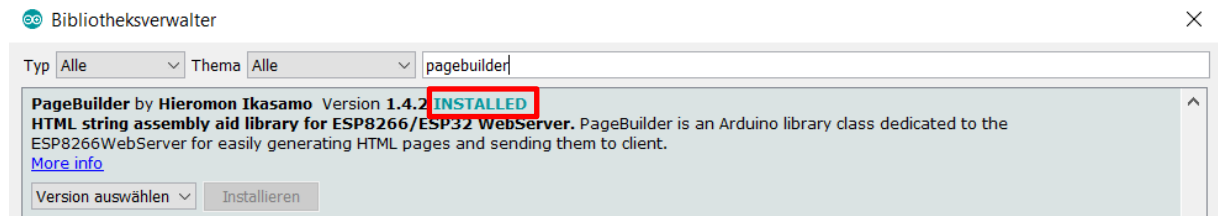


Es sollte sich erneut folgendes Fenster öffnen, in diesem sucht ihr den Ort auf dem Desktop an dem ihr im vorherigen Schritt die Bibliothek abgespeichert habt. Dann klickt ihr auf den ZIP-Ordner und bestätigt mit „öffnen“.



Wenn ihr nun erneut nach den Bibliotheken im Menü unter:

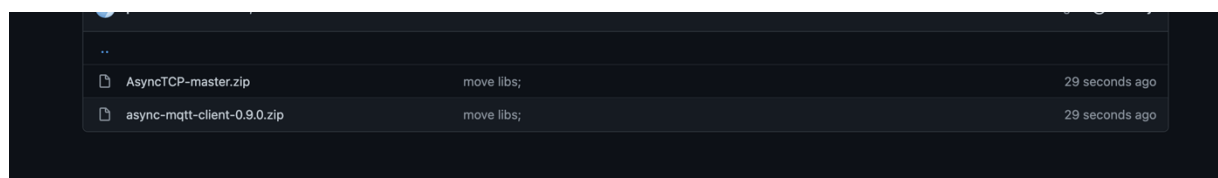
Sketch -> Bibliothek einbinden -> Bibliothek verwalten sucht, sollten diese neben dem Namen den Zusatz „Installed“ aufführen.



Als weitere Library „Adafruit Unified Sensor“ installieren:



Desweiteren wird eine MQTT Library benötigt, dazu findet man im Repository einen Ordner „arduino-libraries“, darin liegen zwei ZIP Dateien:

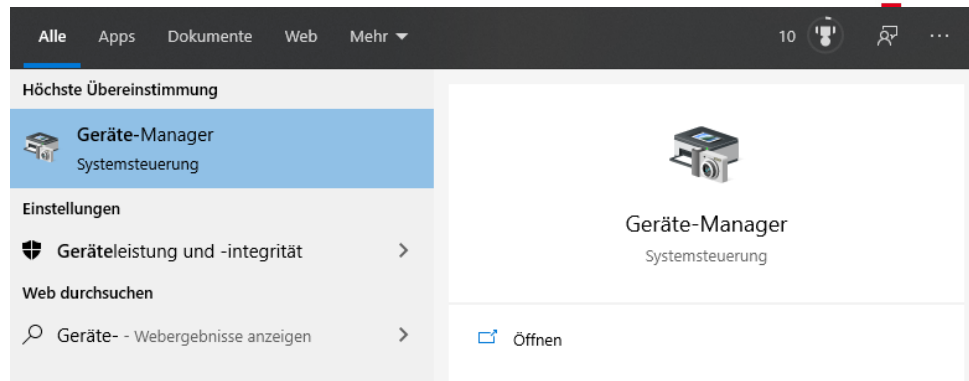


Diese ZIP-Dateien jeweils herunterladen, aber nicht entpacken.

Nun in der Arduino Software Sketch -> Bibliothek einbinden -> .ZIP-Bibliothek hinzufügen und dann jeweils zu den ZIP-Dateien navigieren und diese öffnen.

## 4. Code hochladen

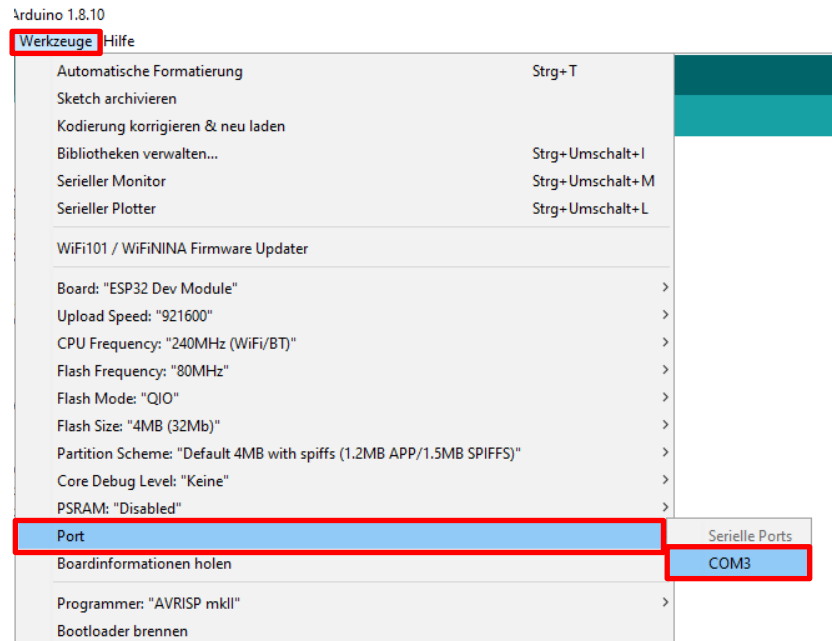
Als letzten Schritt verbindet ihr den Mikrocontroller mit eurem PC. Sucht auf eurem PC nach dem Geräte-Manager und öffnet die Anwendung.



Im Geräte-Manager könnt ihr unter Anschlüsse nachvollziehen, an welchem Port der Mikrocontroller angeschlossen ist.



Öffnet wieder das Programm Arduino IDE und wählt unter Werkzeuge -> Port -> den Port aus, der beim Geräte-Manager angezeigt wurde.



Nun drückt ihr auf den Pfeil für das Hochladen des Codes. Nach dem Vorgang ist der Mikrocontroller bereit für den Einsatz!

