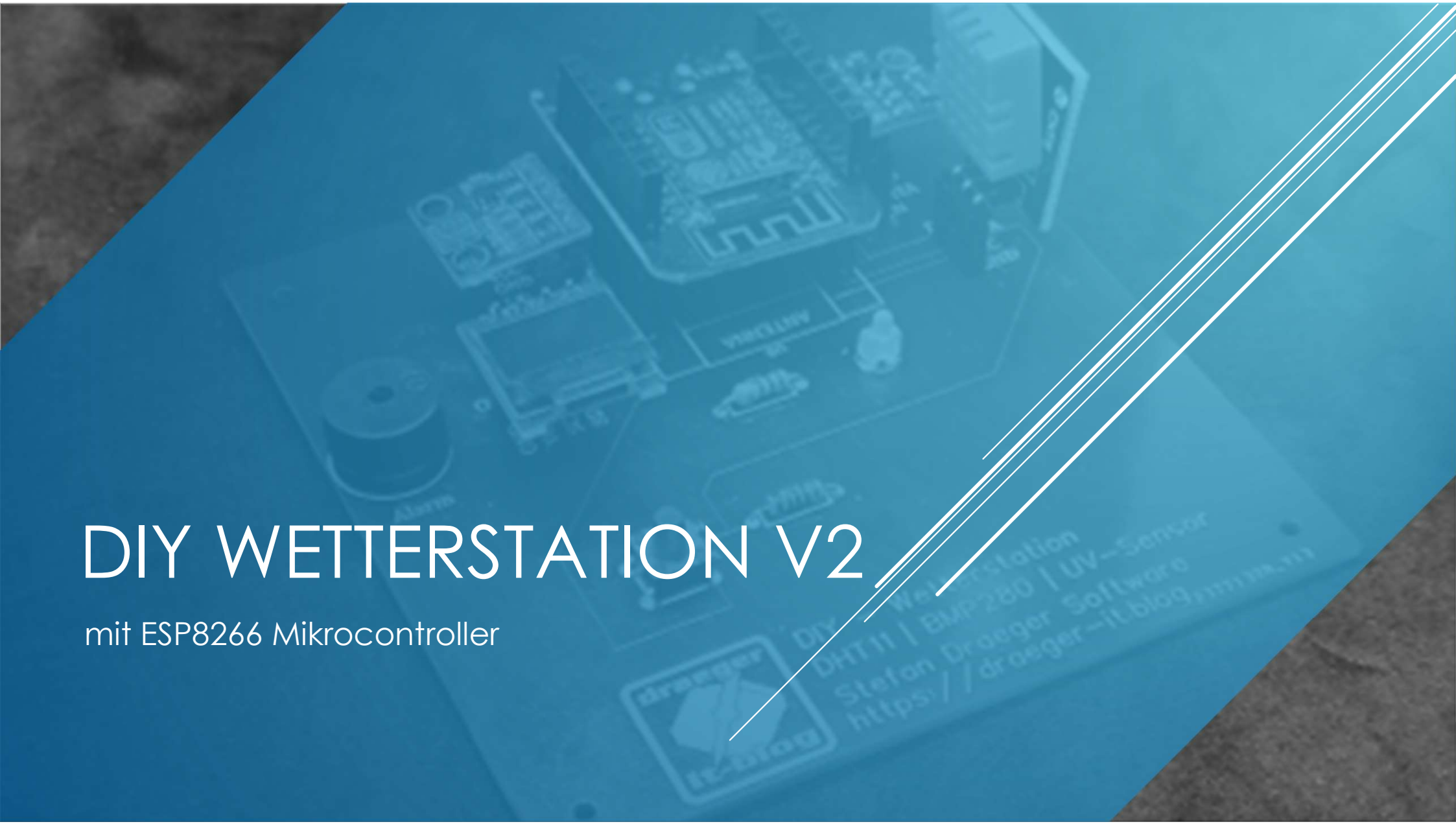


DIY WETTERSTATION V2

mit ESP8266 Mikrocontroller

AGENDA

- Vorstellungsrunde
- Was wollen wir machen?
- Einrichten der Entwicklungsumgebung
- Aufbau der Platine
- Programmieren der Sensoren & Aktoren
- Anbindung an ThingSpeak
- Programmieren einer Webseite mit Sensordaten

VORSTELLUNGSRUNDE

- Name: Stefan Draeger
- Alter: 42
- Familienstand: verheiratet, 2 Kinder
- Beruf: Softwareentwickler
- Hobbies: Mikrocontroller, Garten, Haus

WAS WOLLEN WIR MACHEN?

Einen Mikrocontroller programmieren

- Arduino IDE einrichten & benutzen
  - C/C++ kennenlernen
- Mikrocontroller Wemos D1 Mini
- Programmieren von Sensoren & Aktoren
- Aufbau einer WiFi Verbindung
  - Einrichten des Mikrocontrollers als AccessPoint

Einen IoT Dienst ansprechen

- Einrichten
  - eines Accounts
  - eines Dashboards
- Aufbau der Verbindung zu ThingSpeak

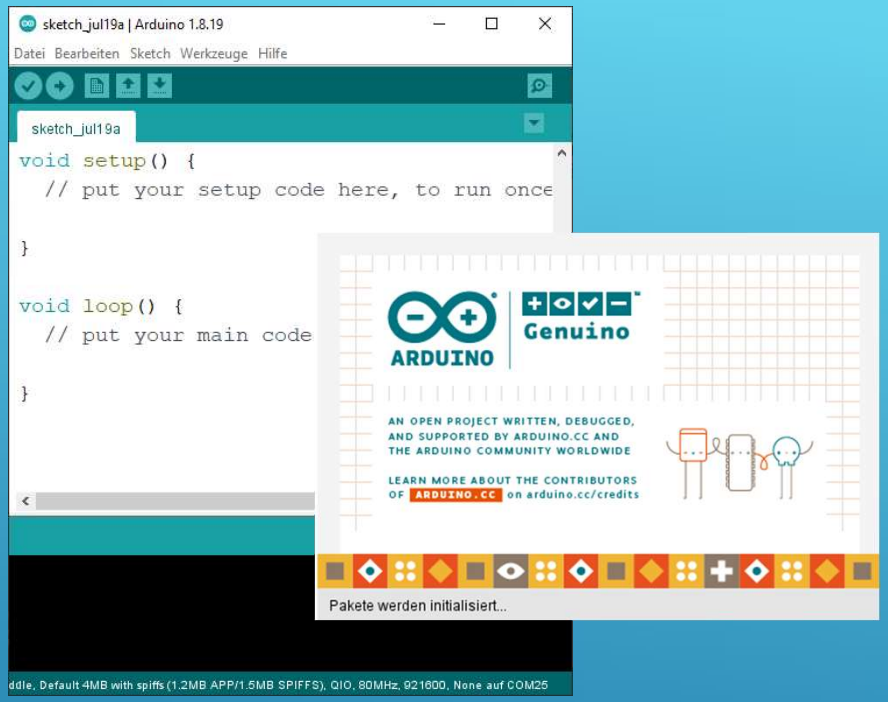
Eine Webseite ausliefern

- programmieren einer einfachen HTML Seite
- Einrichten des ESP8266 als Webserver

RESSOURCEN

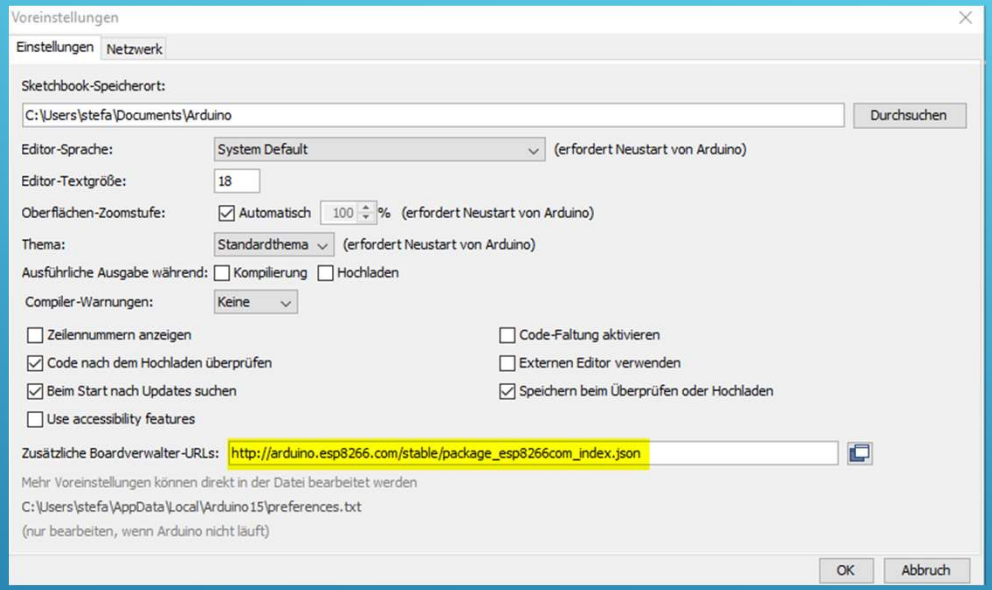
- Downloads & andere Ressourcen auf USB Stick bei mir ☺

- kostenloser Download unter https://arduino.cc
- verfügbar für Windows, macOS & Linux
- als EXE-Datei oder ZIP-Datei



EINRICHTEN DER ARDUINO IDE

Download & Installieren

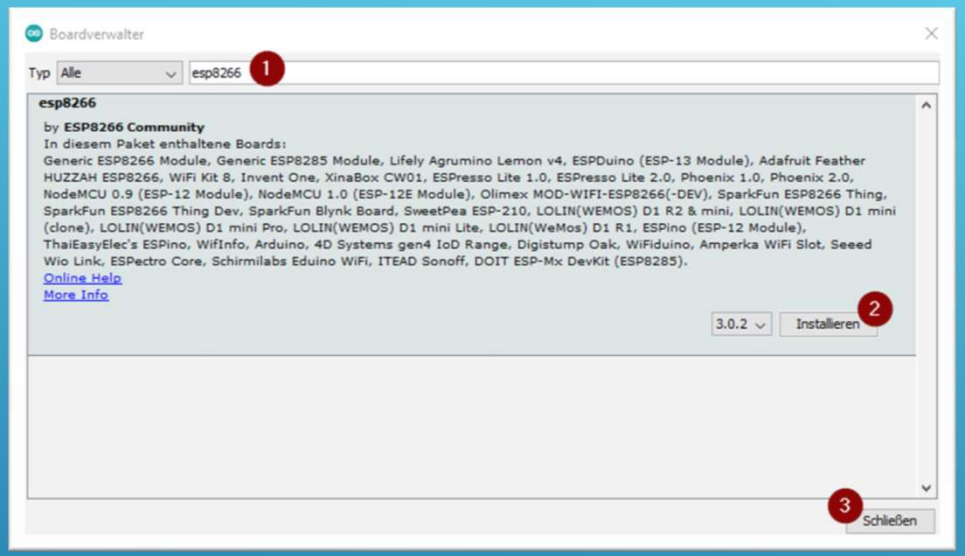


- zusätzliche URL für den Boardverwalter

http://arduino.esp8266.com/stable/package_esp8266com_index.json

EINRICHTEN DER ARDUINO IDE

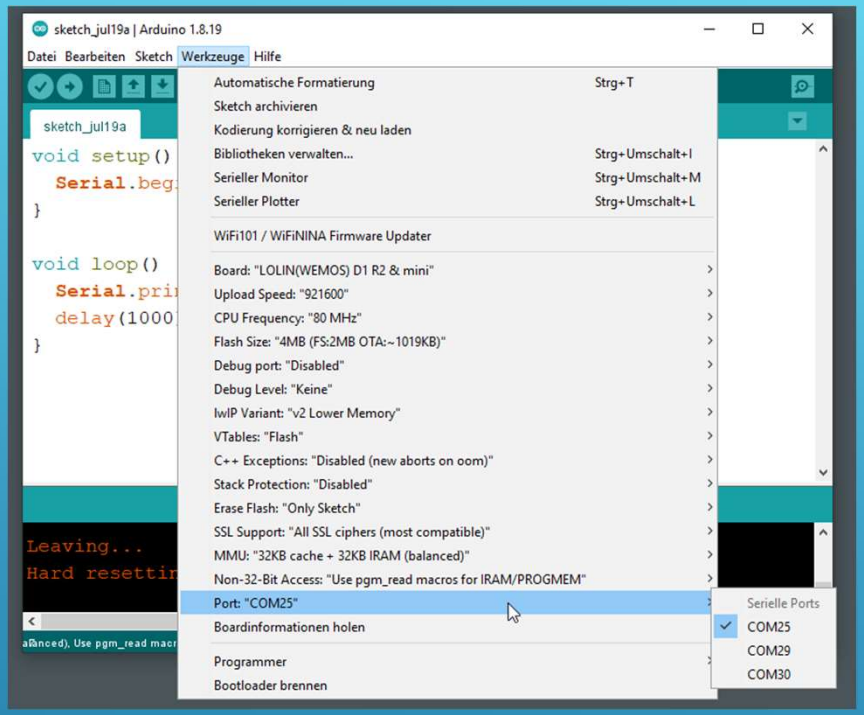
Einrichten des ESP8266

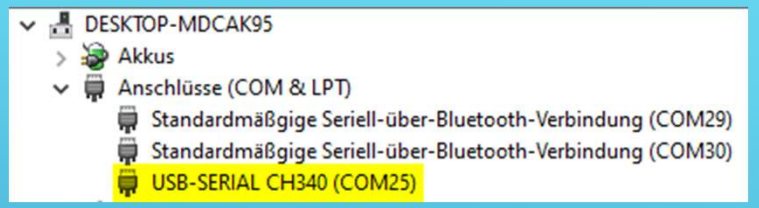


- Schritt 1 – suchen nach „esp8266“
- Schritt 2 – Schaltfläche „Installieren“ betätigen
- Schritt 3 – Schaltfläche „Schließen“ betätigen

EINRICHTEN DER ARDUINO IDE

Installieren des Boardtreibers





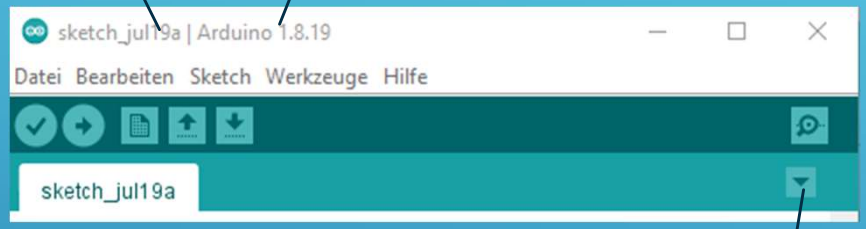
- Hauptmenü „Werkzeuge“ > „Board: xyz“
  > „ESP8266 Boards (3.0.2)“
  > „LOLIN(WEMOS) D1 R2 & mini“
- Port ggf. aus dem Geräte-Manager ablesen

EINRICHTEN DER ARDUINO IDE

Board wählen

Dateiname

Version der IDE



zusätzliche Datei anlegen

- Überprüfen
- Hochladen
- Neu
- Öffnen
- Speichern
- Serieller Monitor

unten rechts gewählter Mikrocontroller

ARDUINO IDE

Toolbar & Funktionen

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

- Funktion „setup“ wird einmalig ausgeführt wenn der Mikrocontroller gestartet wird.
- Funktion „loop“ wird dauerhaft ausgeführt.

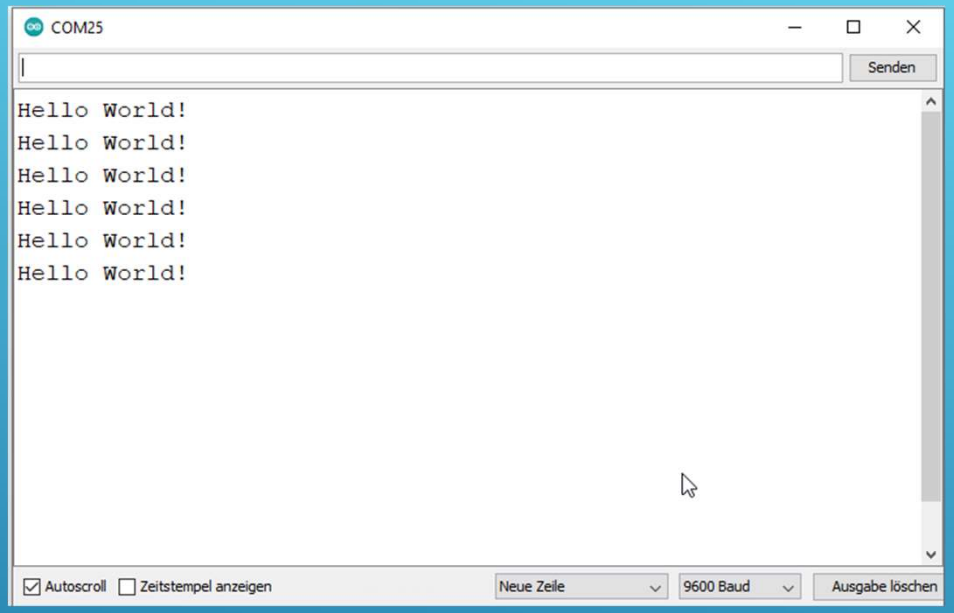
ARDUINO IDE

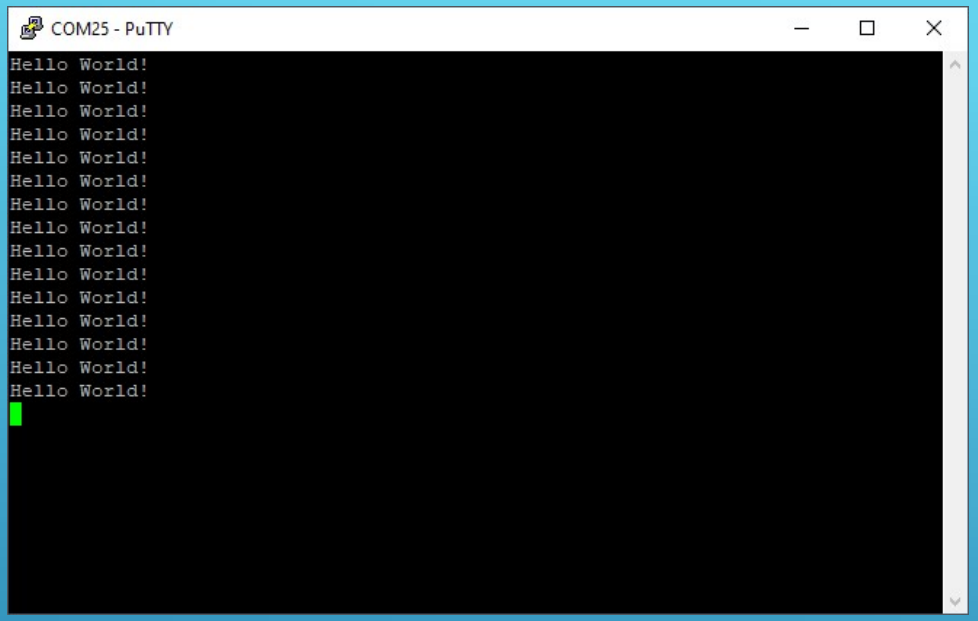
Funktionen setup & loop

ARDUINO IDE
Ein kleines Beispiel

```
void setup() {
  //beginn der seriellen Kommunikation mit
  //9600 baud
  Serial.begin(9600);
}

void loop() {
  //Ausgabe des Textes "Hello World!" auf
  //der seriellen Schnittstelle
  Serial.println("Hello World!");
  //eine kleine Pause von 1 Sekunde (1000 ms.)
  delay(1000);
}
```

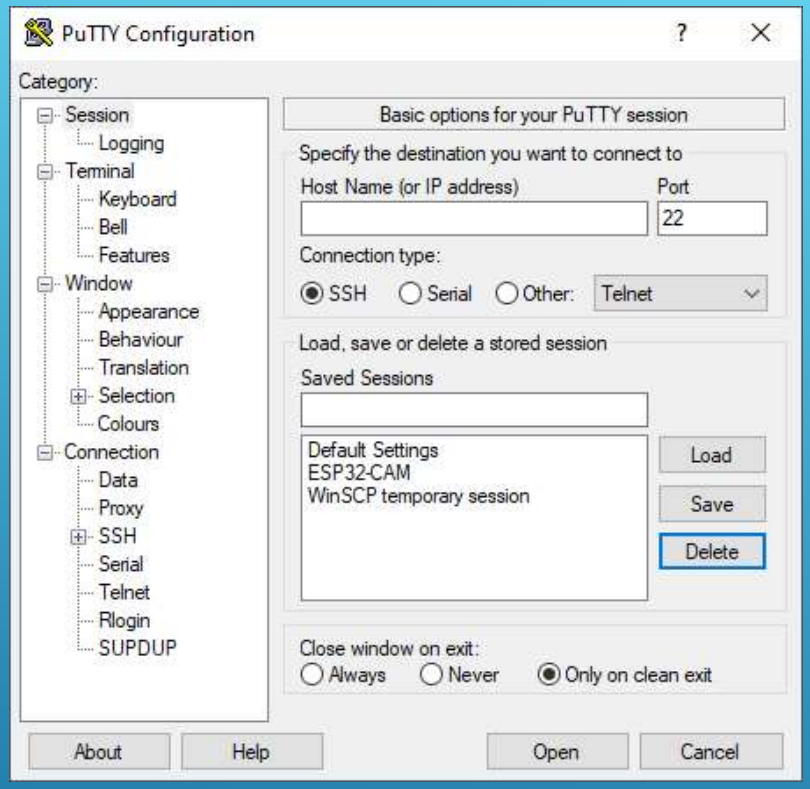
- es wird auf der seriellen Schnittstelle der Text “Hello World!” ausgeführt
- lesbar zbsp. im seriellen Monitor der Arduino IDE oder auch über Putty





ARDUINO IDE

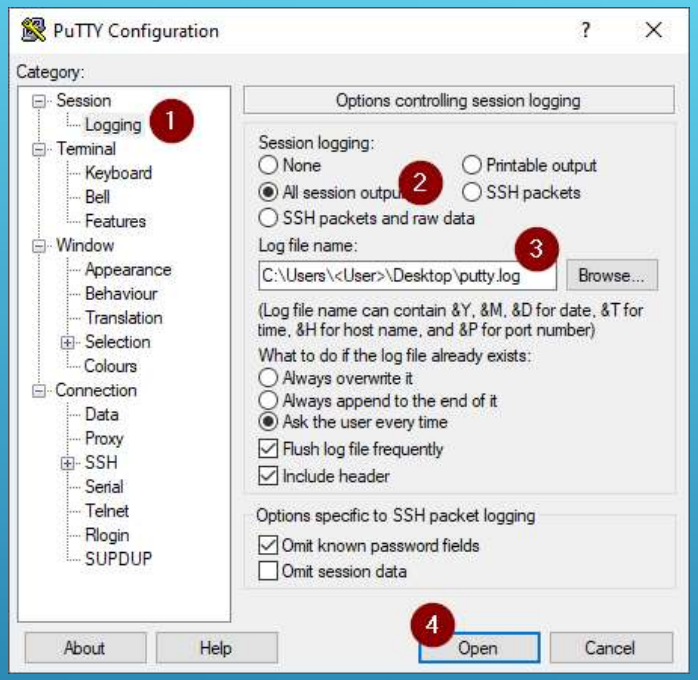
Ausgabe im seriellen Monitor & putty

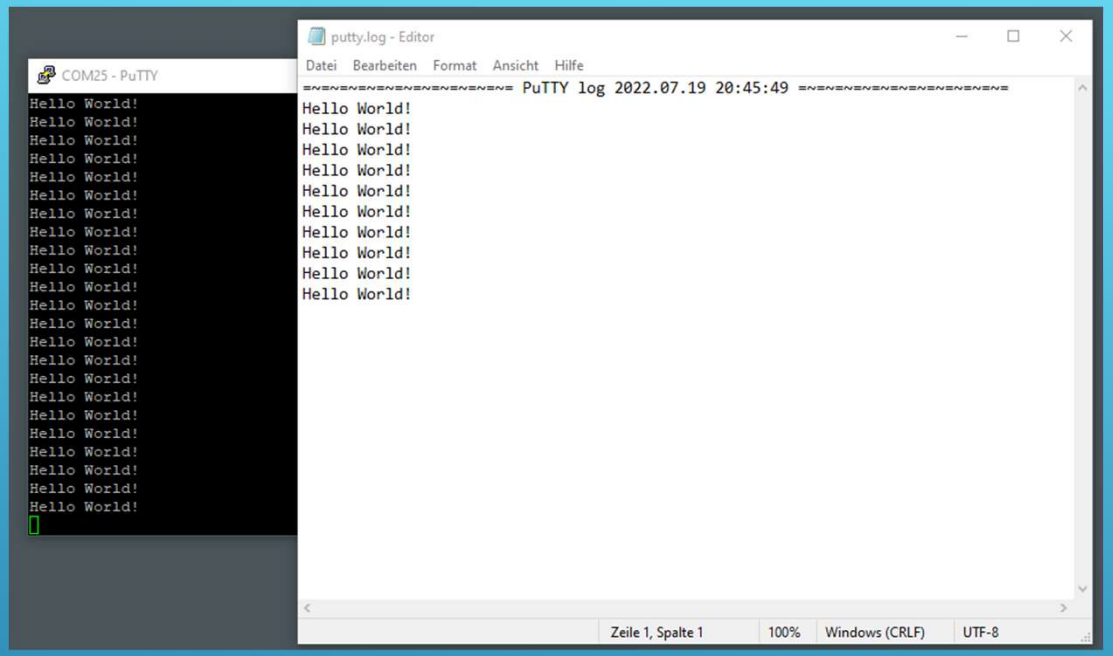


EXKURS PUTTY

Was ist putty?

- Download unter putty.org
- Programm zum Aufbau einer Verbindung zu einem entfernten Computer / Gerät per SSH, Serial oder anderen Schnittstellen
- Logging der Ausgaben in Datei möglich
- viele nützliche Features





EXKURS PUTTY

Loggen der Daten in eine Datei

```
//ein Kommentar

/*
   ein mehrzeiliger Kommentar
*/

//eine Ganze Zahl
int zahl0 = 1;
long zahl1 = 1L;

//eine Gleitkomma Zahl
float zahl2 = 1.2f;
double zahl3 = 1.2d;

//ein Array
int liste[3] = {1,2,3};
```

- Kommentare am Code helfen diesen zu verstehen
- speichern von Werte in Feldern, Variablen
- Ganze Zahlen als Integer, Long
- Gleitkommazahlen als Float oder Double
- eine Liste von Werten als Array

C/C++ PROGRAMMIEREN

kurze Einführung

```
void setup() {
  Serial.begin(9600);
}

void eineFunktion() {
  //do something
}

String eineWeitereFunktion() {
  return "Text";
}

String eineFunktionMitParameter(String name, String vorname) {
  return name + " " + vorname;
}
```

C/C++ PROGRAMMIEREN

Funktionen

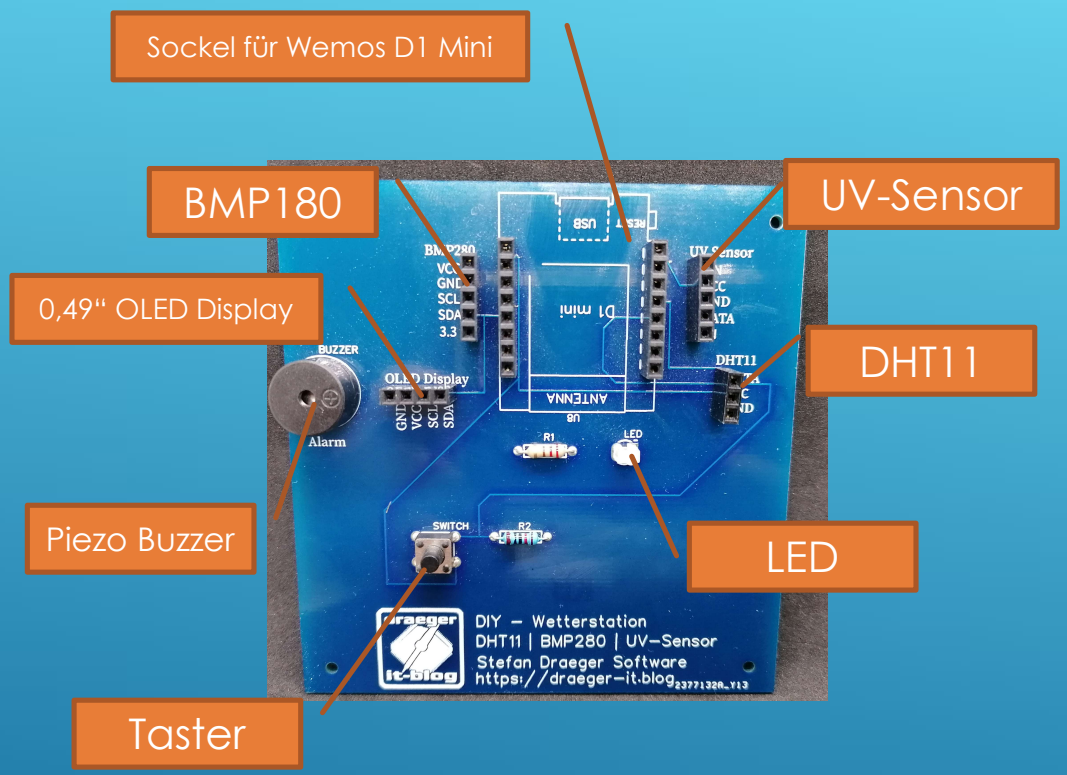
- Funktionen dienen zum strukturieren von Code
- redundanter Code entfällt

- Funktionen für
  - addition,
  - subtraktion,
  - multiplikation,
  - division

zweier Zahlen

C/C++ PROGRAMMIEREN

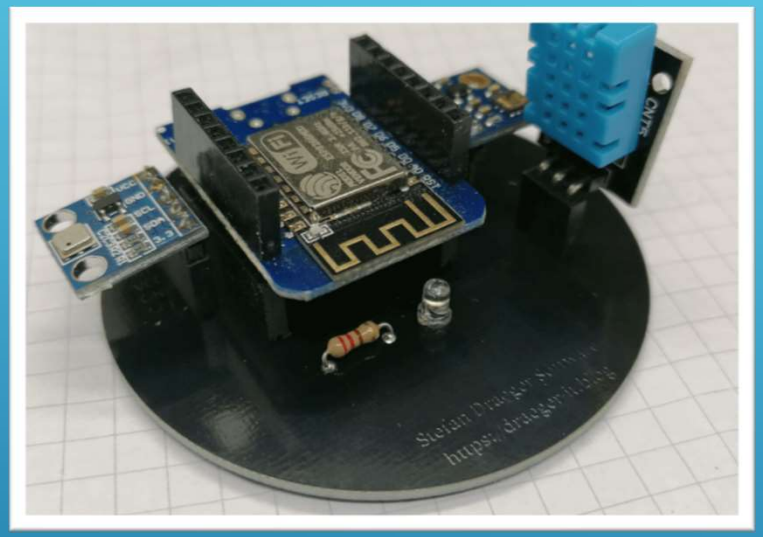
Einen kleinen Taschenrechner programmieren



- Platine verfügt über Buchsenleisten für die Sensoren
  - BMP180
  - UV-Sensor
  - DHT11
  - 0,42“ OLED Display
- Piezo Buzzer für Soundausgabe
- LED, und
- Taster

PLATINE „DIY WETTERSTATION V2“

Aufbau & Überblick



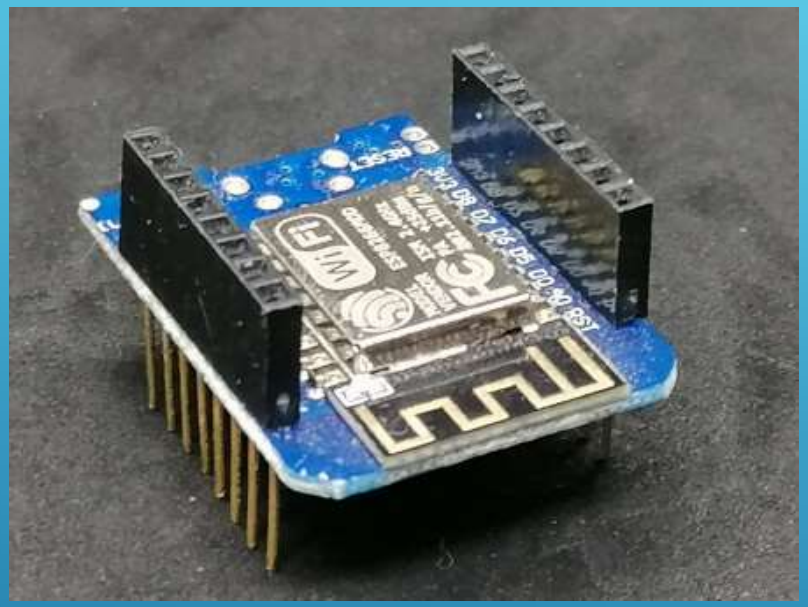
Version 1 der „DIY Wetterstation“

- Der WiFi Chip wird bei betrieb warm und erzeugt somit im Gehäuse in Klima welches die Werte der Sensoren verfälscht.
- Sensoren können besser platziert werden (zbsp. UV-Sensor an die Oberfläche)

PLATINE „DIY WETTERSTATION V2“

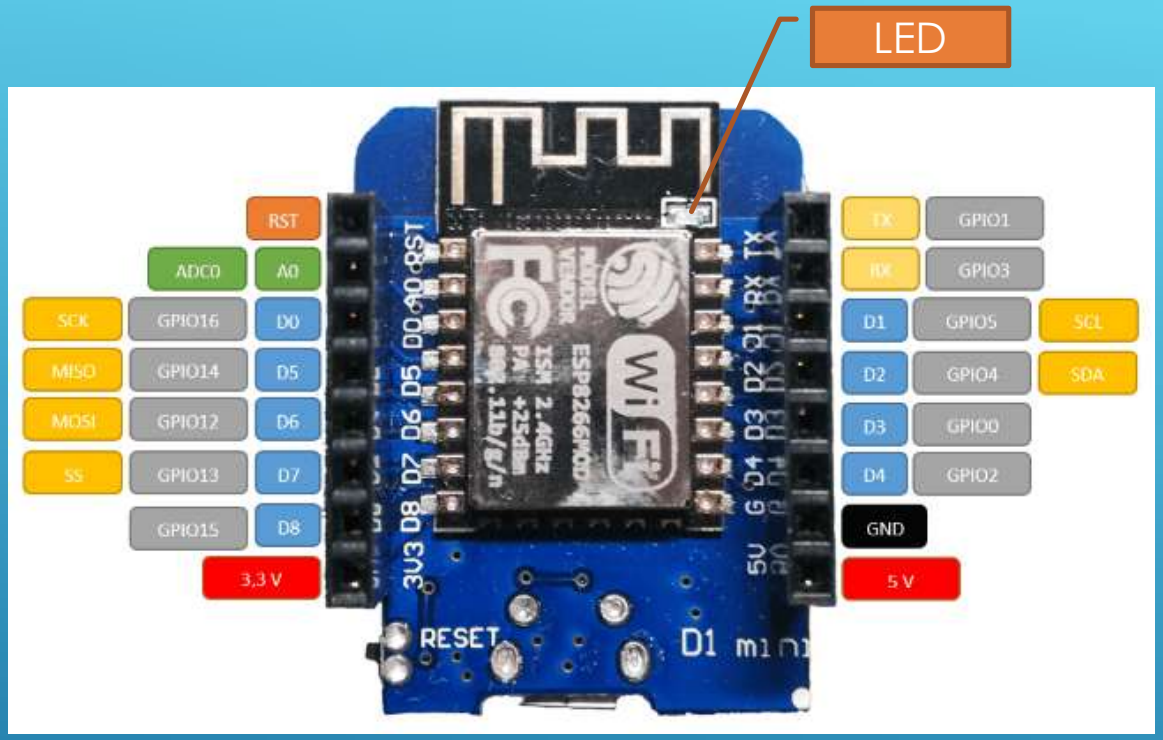
Warum die Sensoren auf Buchsenleisten?

Spezifikation



Länge	34,2 mm
Breite	25,6 mm
Gewicht	10 g
Flash Speicher	4 Mbyte
CPU Taktgeschwindigkeit	80 MHz / 160 MHz
Betriebsspannung	3.3 V
max. Stromstärke pro I/O Pin	20 mA
max. Stromstärke für die 3.3 V Versorgung	50 mA
digitale Eingänge / Ausgänge	11
digitale PWM Ausgänge	11
analoge Eingänge / Ausgänge	1

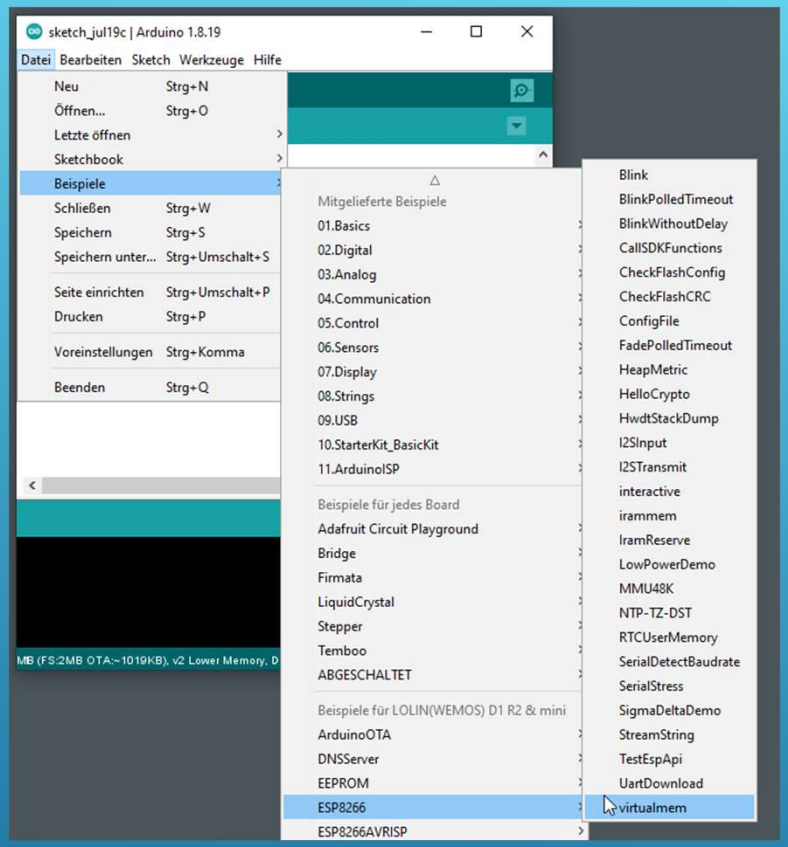
WEMOS D1 MINI (ESP8266)



- Pins für 3.3V & 5V
- Schnittstellen I²C, SPI, UART
- ein analoger Eingang
- 11 digitale Ein/Ausgänge
  - davon 10 als PWM

WEMOS D1 MINI (ESP8266)

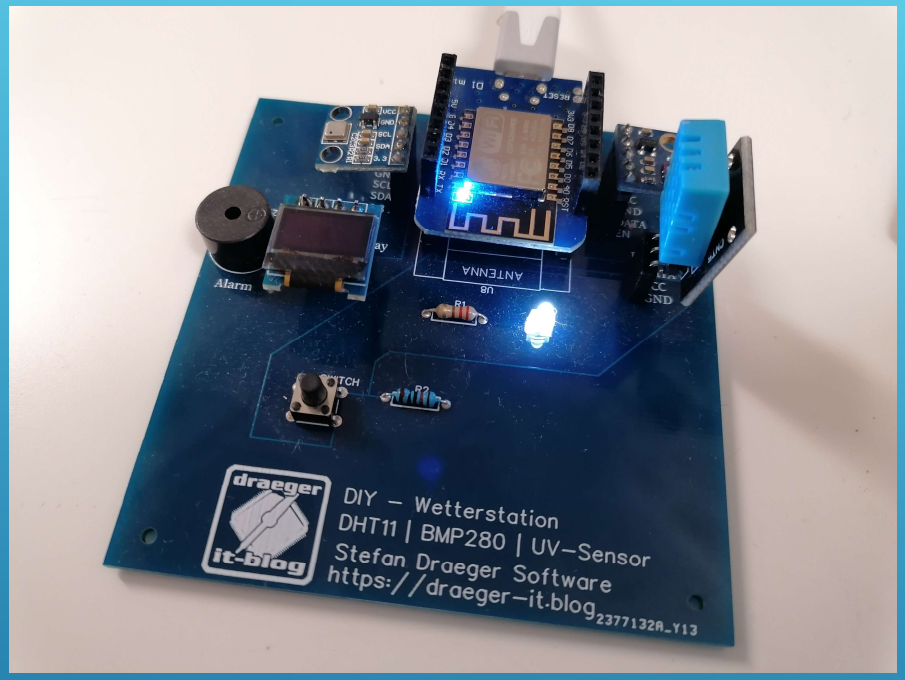
Pinout



- Beispiele unter „Datei“ > „Beispiele“ > „ESP8266“

WEMOS D1 MINI (ESP8266)

Beispiele in der Arduino IDE



LED blinken

LED Fade Effekt

WEMOS D1 MINI (ESP8266)

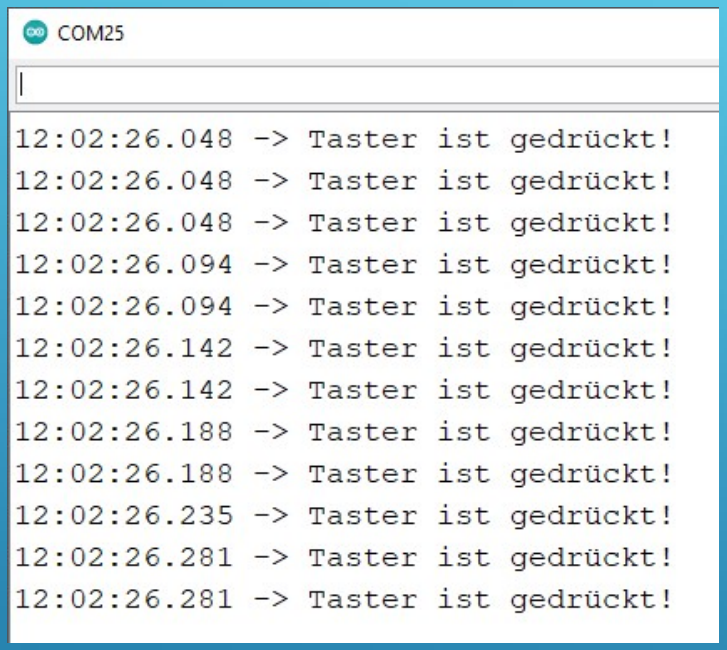
Beispiele

```
#define button D5

void setup() {
  Serial.begin(9600);
  pinMode(button, INPUT);
}

void loop() {
  int state = digitalRead(button);
  if(state == HIGH){
    Serial.println("Taster ist gedrückt!");
  }
}
```

ARDUINO IDE

Einen Taster programmieren

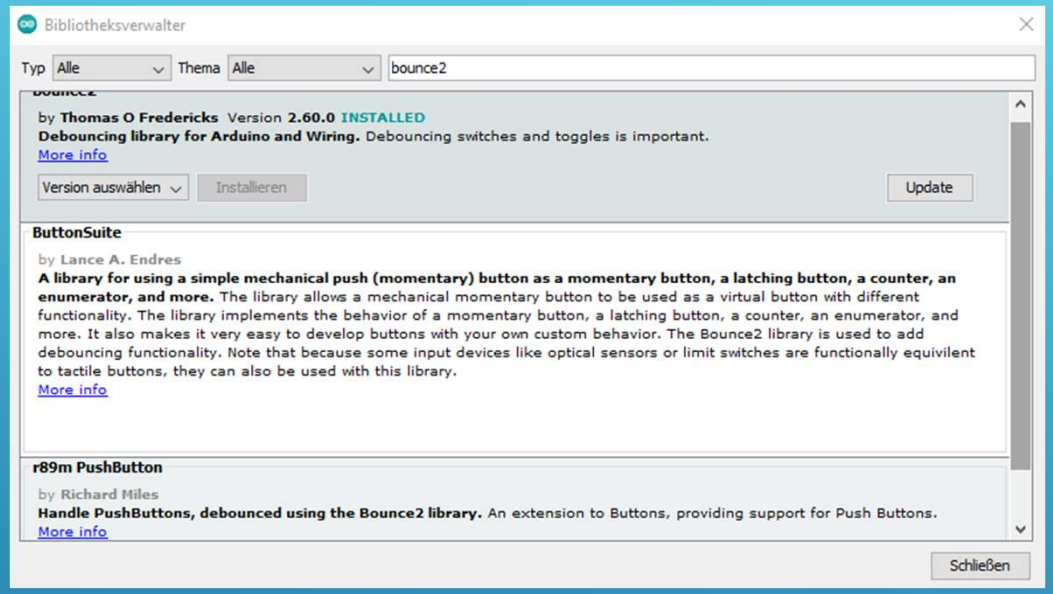
- digitaler Pin muss als Eingang definiert werden
- Status des Tasters auslesen mit digitalRead(<Pin>)
  - HIGH = Taster gedrückt
  - LOW = Taster ist nicht gedrückt



ARDUINO IDE

Taster prellt

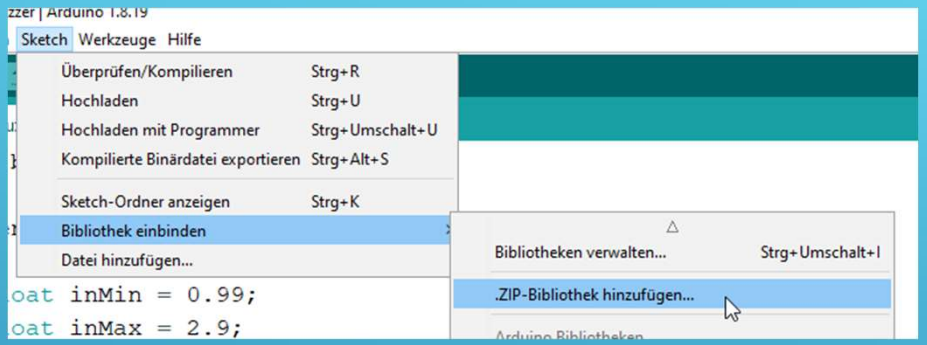
- Mikrocontroller arbeitet mit max. 160 MHz und kann die Operation viele Millionen mal pro Sekunde ausführen
- Status wird somit mehrfach gelesen obwohl der Taster nur kurz betätig wurde
- Wie entprellt man einen Taster?

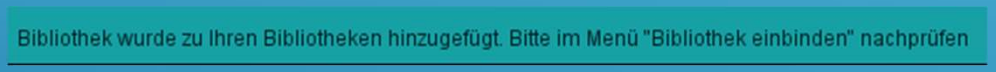


- Bibliotheken ermöglichen einfachen Zugriff auf komplexe Programmierung
- einfacher Zugriff auf Sensoren & Aktoren
- „Sketch“ > „Bibliothek einbinden“ > „Bibliotheken verwalten…“

ARDUINO IDE

Taster entprellen & Installieren einer Bibliothek

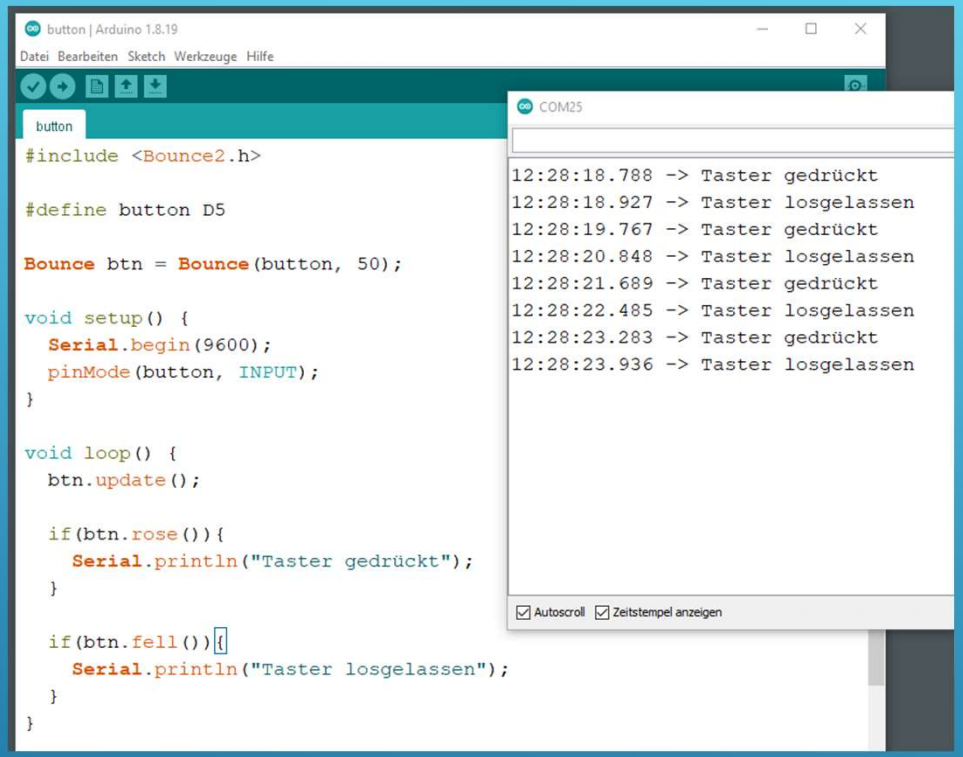




- Öffnen der ZIP-Datei
  - „Sketch“ > „Bibliothek einbinden“ > „.ZIP-Bibliothek hinzufügen…“
- Datei muss einem definierten Format entsprechen

ARDUINO IDE

alternative Installation einer ZIP-Datei



ARDUINO IDE

Taster entprellen mit bounce2

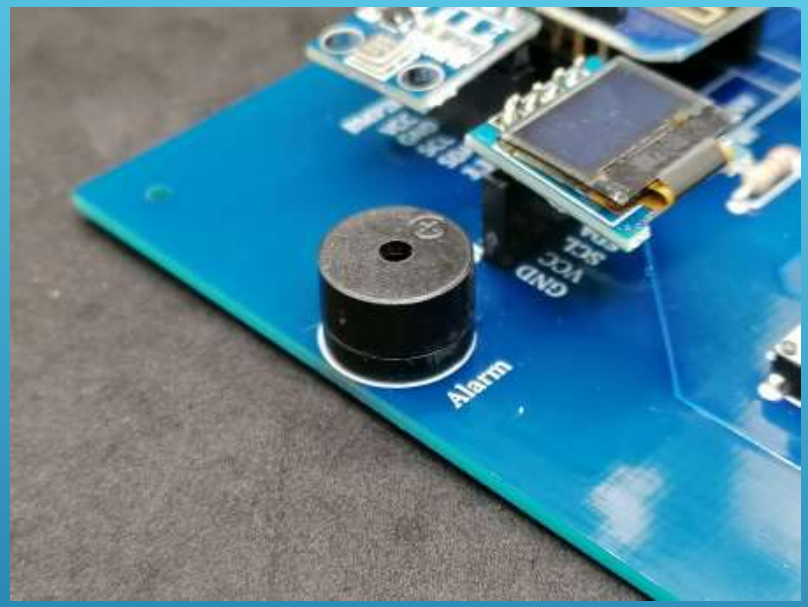
- Schritt 1 - einbinden der Bibliothek
- Schritt 2 – initialisieren eines Objektes vom Typ Bounce mit Parameter button und Millisekunden für die Pause,
- Schritt 3 – in der Loop, aktualisieren des Tasterobjektes,
- Schritt 4 – auslesen des Tasters mit
  - rose() – Taster gedrückt
  - fell() – Taster losgelassen

LED aktivieren mit dem Taster

LED bei Tastendruck aktivieren / deaktivieren

WEMOS D1 MINI (ESP8266)

Beispiele



- Ausgabe einfacher Töne & kleine Lieder
- kann als Alarm für einen Schwellwert eingesetzt werden

ARDUINO IDE

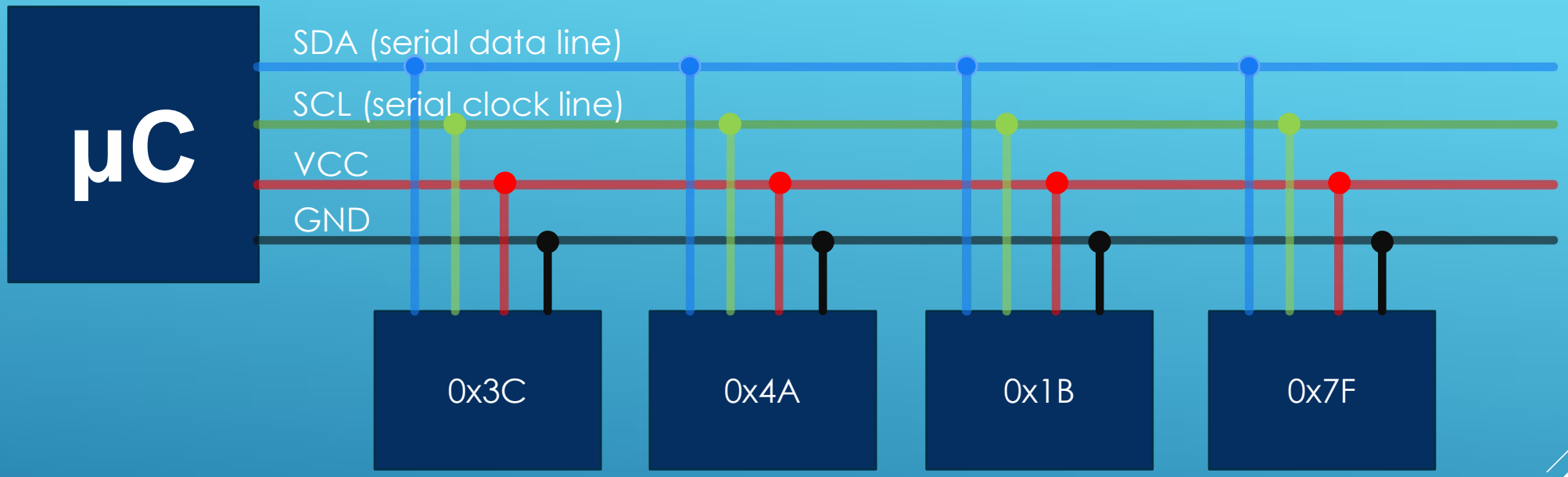
Soundausgabe

```
//Piezo Buzzer am
//digitalen Pin D8 angeschlossen
#define buzzer D8

void setup() {
  //Pin des Buzzers als Ausgang definieren
  pinMode(buzzer, OUTPUT);
}

void loop() {
  //Ausgabe von einem Ton mit 750 Hertz
  tone(buzzer, 750);
  //eine kleine Pause von 500ms.
  delay(500);
  //den Ton deaktivieren
  noTone(buzzer);
}
```

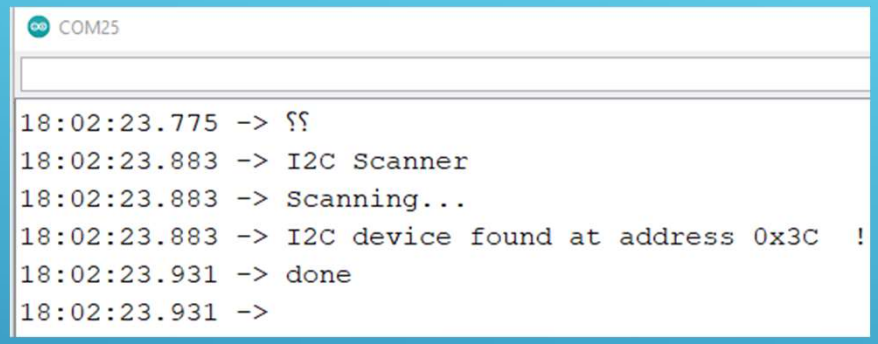
ARDUINO IDE

Ausgabe von Tönen

- Erzeugen eines Tones mit der Funktion
  - tone(<Pin>,<Frequenz>)
  - Tone(<Pin>,<Frequenz>, <Dauer>)



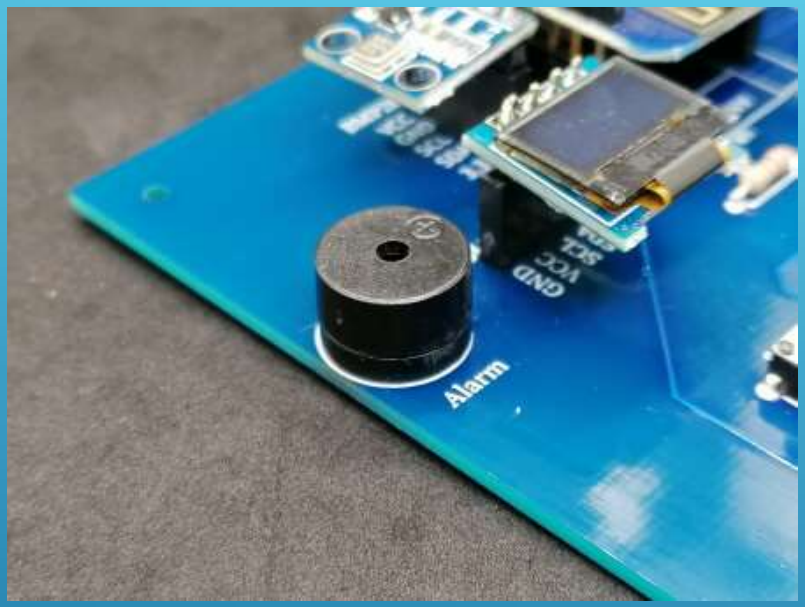
EXKURS I²C KOMMUNIKATION



- URL: https://playground.arduino.cc/Main/I2cScanner/
- ermöglicht die einfache suche nach Sensoren welche per I²C angebunden sind

EXKURS I²C KOMMUNIKATION

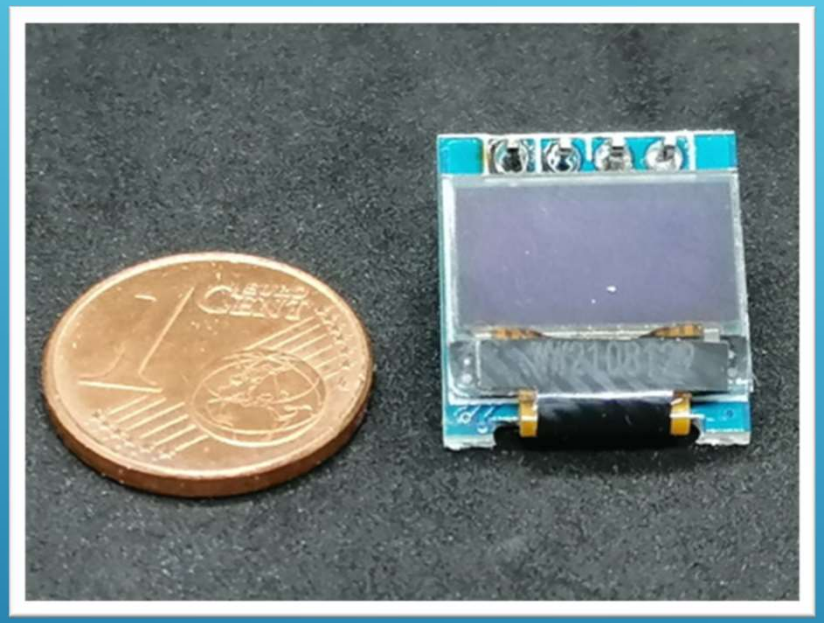
I²C Scanner



Abspielen von Tönen per Tastendruck

WEMOS D1 MINI (ESP8266)

Beispiele



- Auflösung 64 x 32 Punkte
- Schnittstelle I²C

ARDUINO IDE

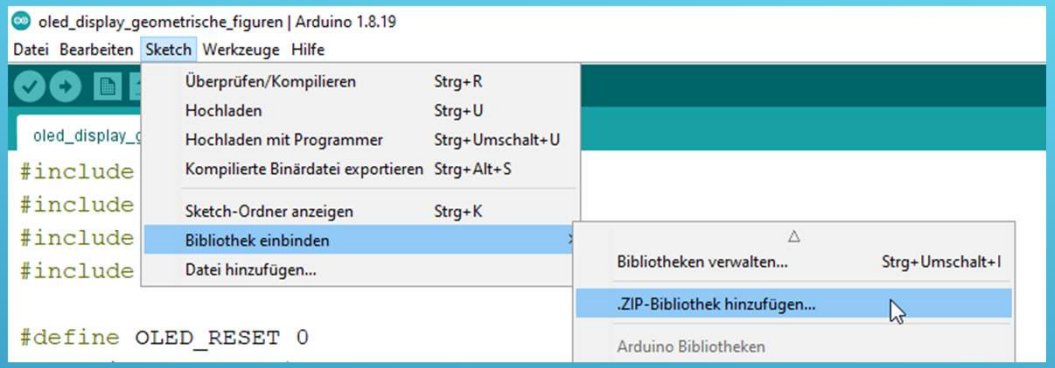
0,49“ OLED Display programmieren

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET 0
Adafruit_SSD1306 display(OLED_RESET);

void setup()   {
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.display();
  delay(2000);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(60, 15);
  display.println("Hello");
  display.display();
}

void loop() {}
```

- I2C Adresse des Displays 0x3C
- Textgröße 1
- Textfarbe – weiß
- Cursor Position
  - Spalte 60
  - Zeile 15

ARDUINO IDE

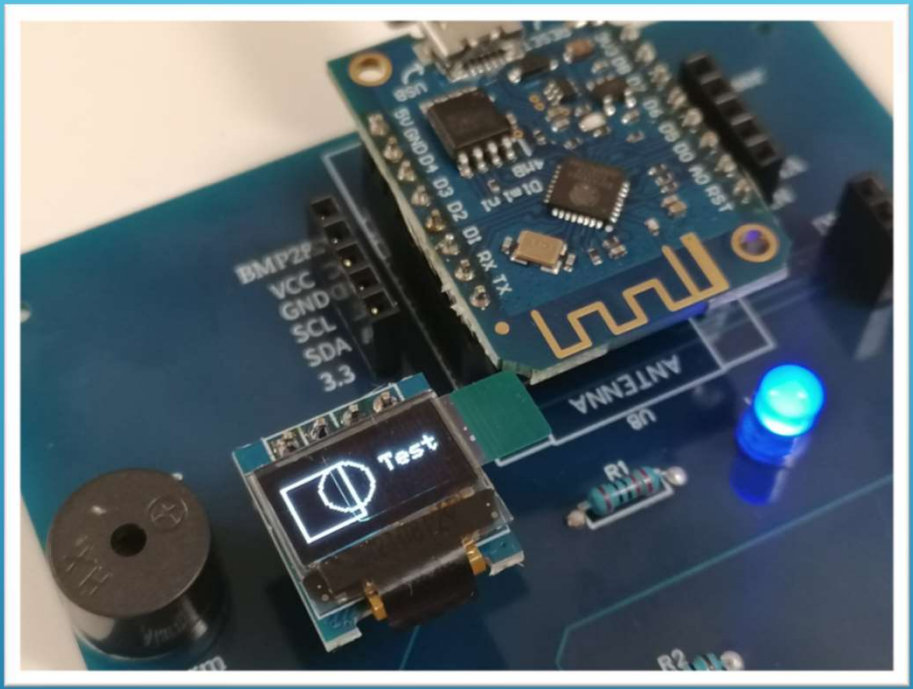
Anzeigen des Textes „Hello“



- Benötigte Bibliotheken:
  - Adafruit_BusIO
  - Adafruit_SSD1306_Wemos_OLED
  - Adafruit-GFX-Library

ARDUINO IDE

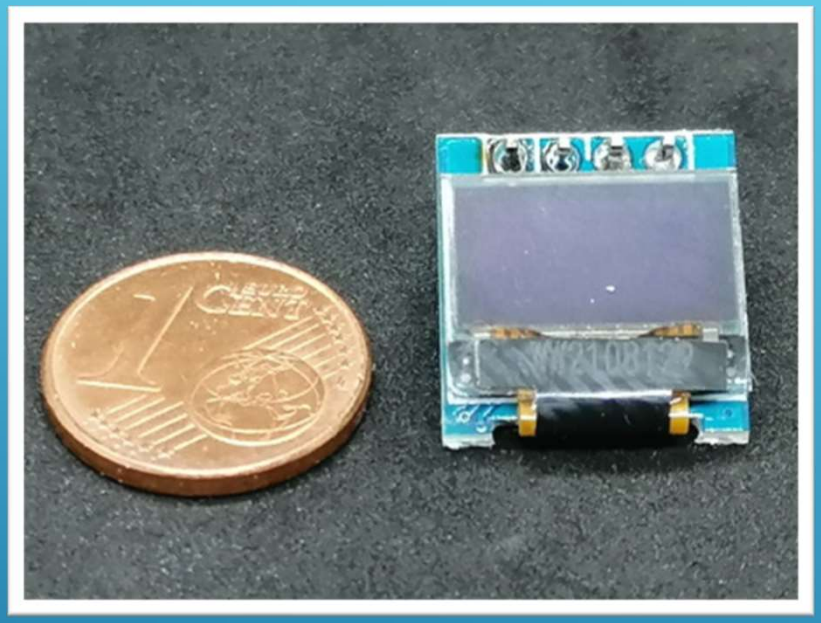
Installieren der benötigten Bibliotheken



- gedacht für große OLED Displays
- viele nützliche Funktionen
  - Schriftart, Schriftgröße, Farbe
  - Zeichnen von einfachen geometrischen Figuren
  - Zeichnen von XBMP Bilder
- weitere Informationen unter https://adafruit.github.io/Adafruit_SSD1306/html/class_adafruit___s_s_d1306.html

ARDUINO IDE

Besonderheiten bei der Adafruit Bibliothek

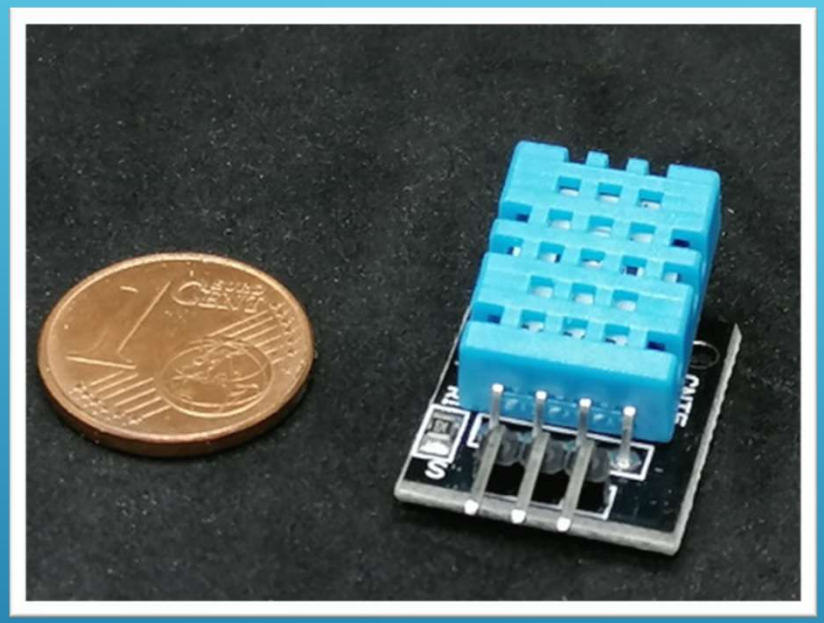


Anzeigen von Text

Anzeigen von geometrischen Figuren

WEMOS D1 MINI (ESP8266)

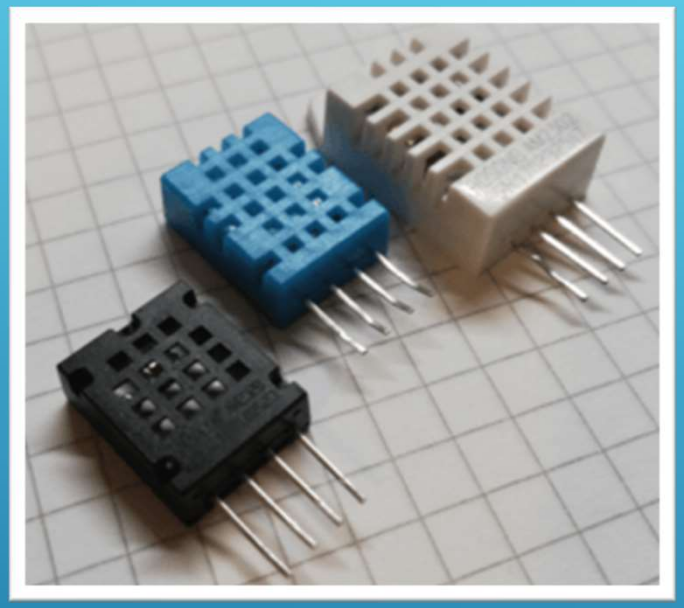
Beispiele



- Technische Daten:
  - rel. Luftfeuchtigkeit
    - Messbereich 20 % bis 90 %
    - Toleranz ±5 %
  - Temperatur
    - Messbereich 0 bis 60 °C
    - Toleranz ±2 °C
  - Betriebsspannung 5 V

ARDUINO IDE

Sensor DHT11



- DHT11 – blau
- DHT22 – weiß
  - deutlich genauer,
  - teurer
- AM2320 – schwarz
  - deutlich genauer,
  - anderes Pinout,
  - teurer

ARDUINO IDE

Alternativen zum DHT11

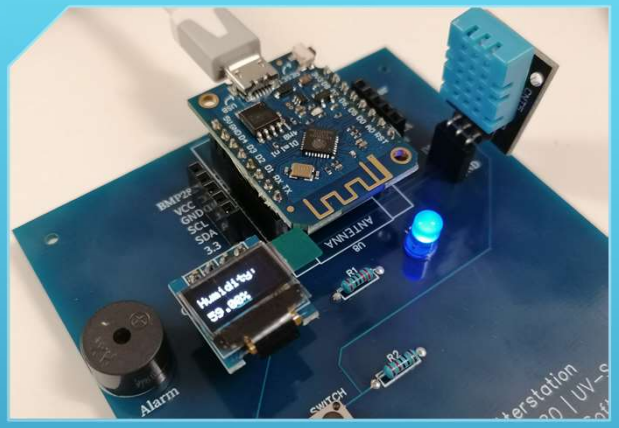
- Bibliothek DHTesp
- Beispiele für ESP32, ESP8266

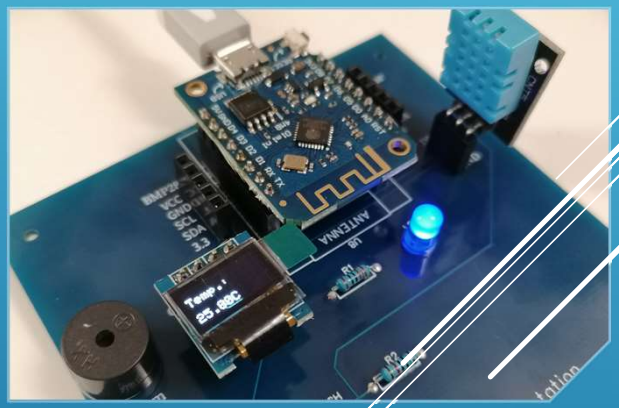
ARDUINO IDE

Installieren der benötigten bibliothek

Ausgabe auf dem seriellen Monitor

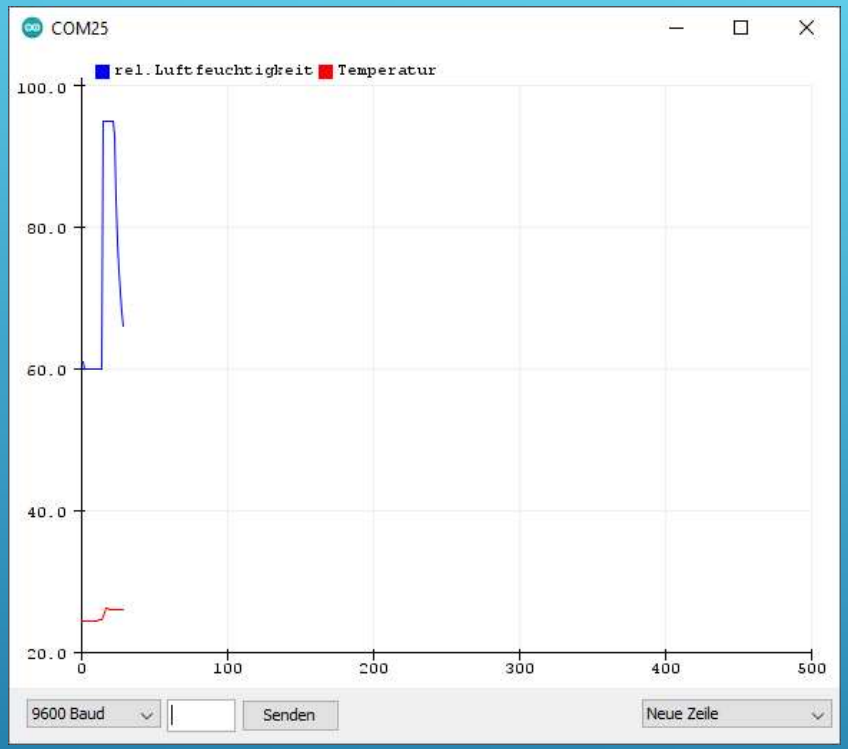
Anzeigen auf dem OLED Display





WEMOS D1 MINI (ESP8266)

Beispiele

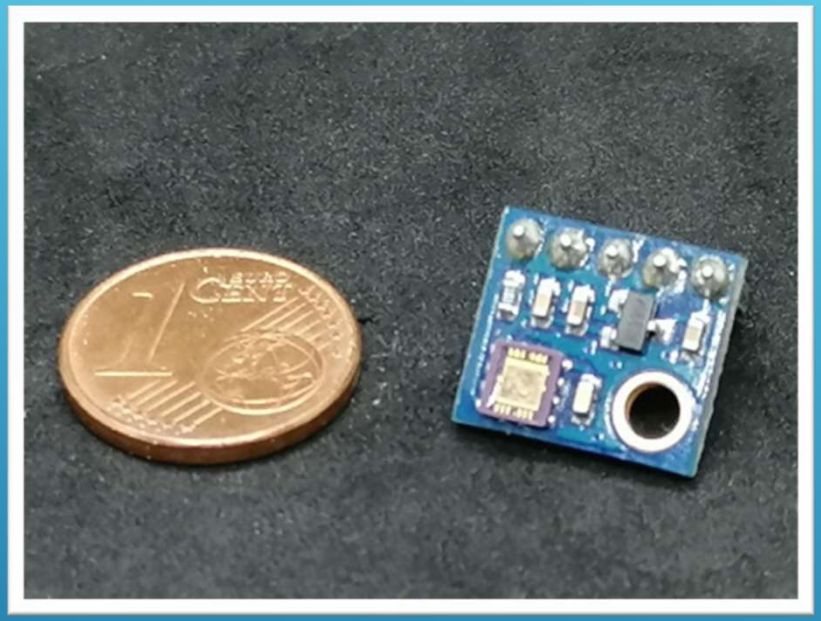


ARDUINO IDE

serieller Plotter

- visualisieren von Daten eines oder mehrerer Sensoren

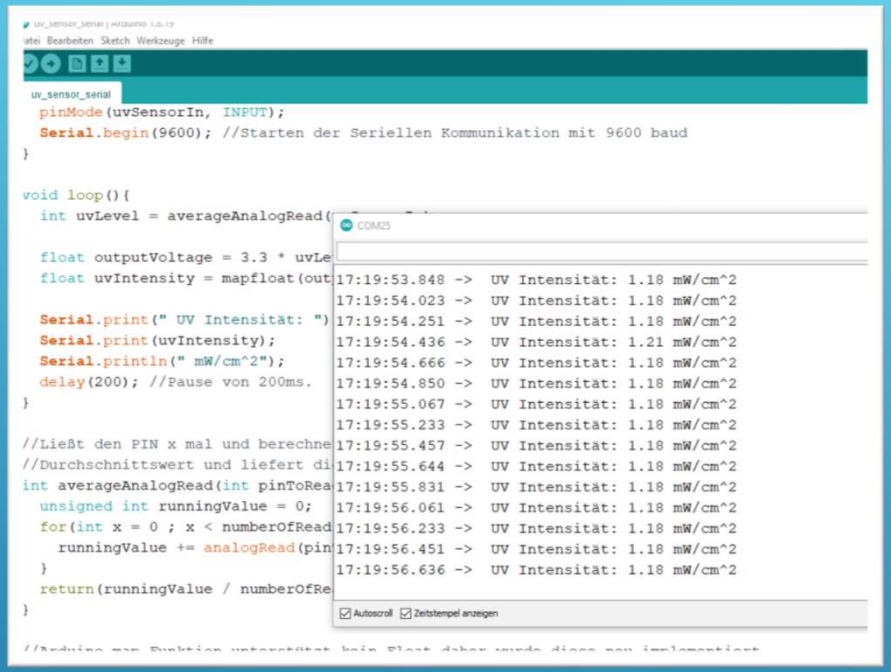
```
String line1 = "rel.Luftfeuchtigkeit Temperatur";
String line2 = String(luftfeuchtigkeit, DEC)+ " " + String(temperatur, DEC);
Serial.println(line1);
Serial.println(line2);
```



- Technische Daten
  - Betriebsspannung 3,3 V
  - maximale Stromaufname 0,3 mA
  - Ruhestrom 0,1 µA

ARDUINO IDE

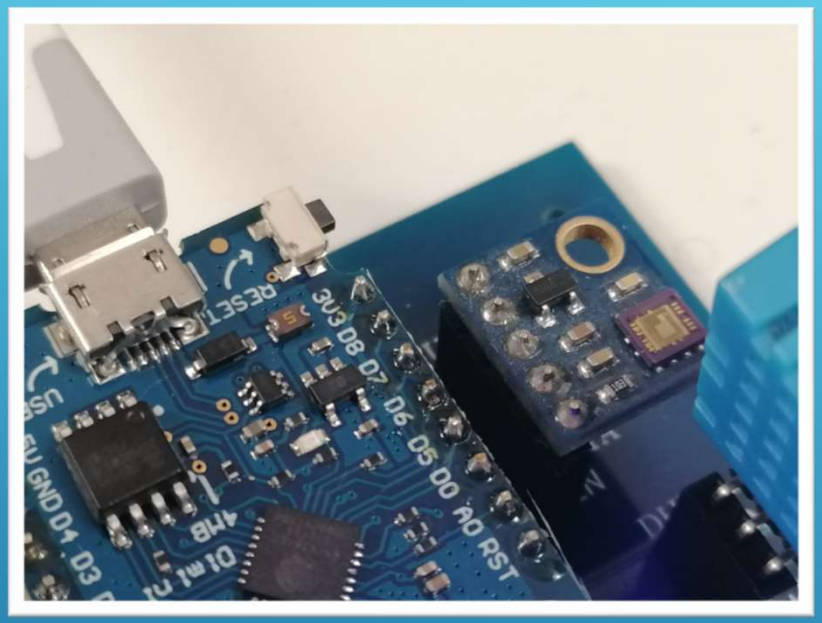
analoger UV-Sensor



- Komplizierte Berechnung aus einem analogen Wert (0..1023) zu einem UV Wert

ARDUINO DIE

Berechnung des UV-Wertes

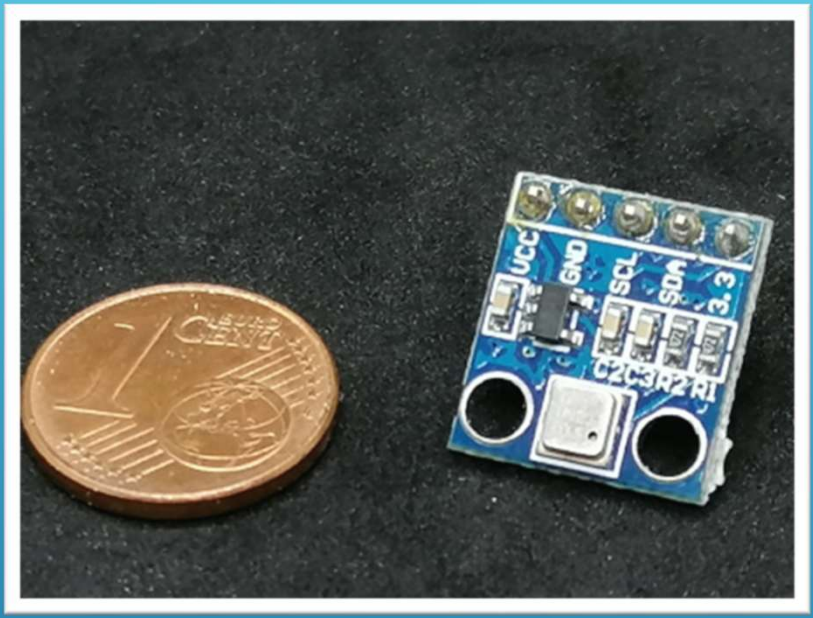


Anzeigen auf dem OLED Display

Ausgabe von Sound bei erreichen eines Schwellwertes

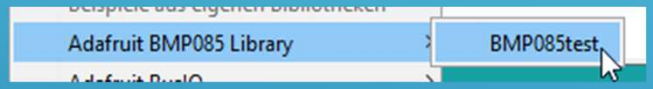
WEMOS D1 MINI (ESP8266)

Beispiele



- Technische Daten
  - Betriebsspannung 1.8 V bis 3.6 V
  - Leistungsaufnahme 0,5µA bei 1Hz
  - Genauigkeit 0.02 hPa
  - Messbereich 300 hPa bis 1100 hPa

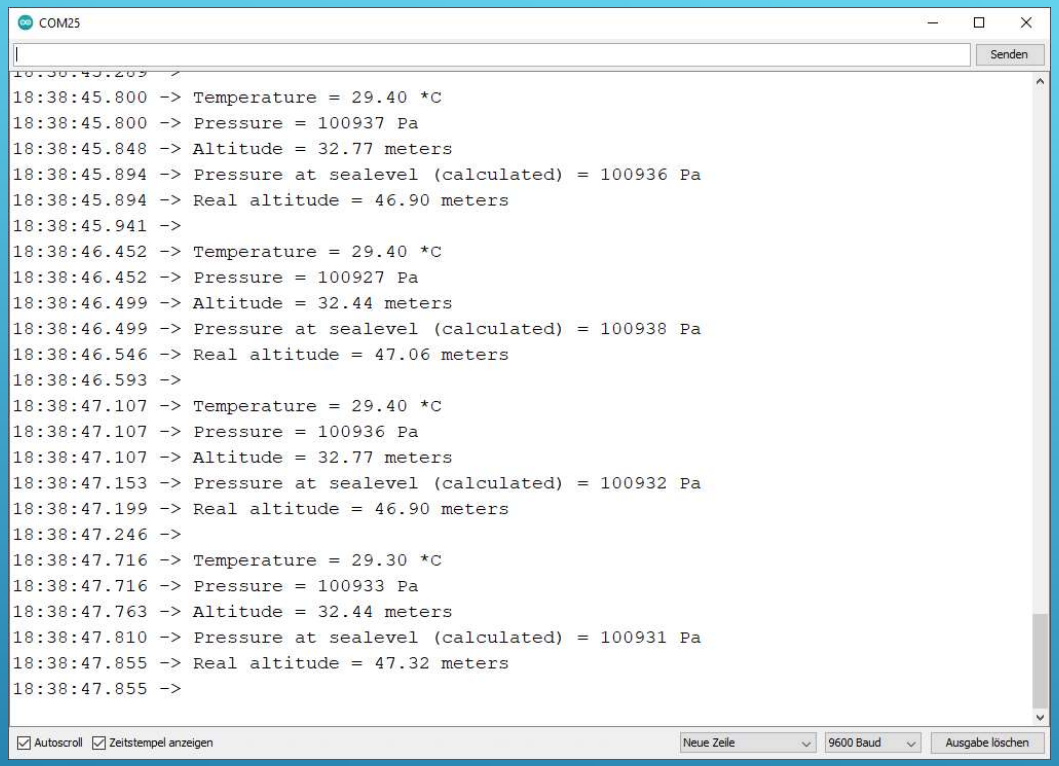
ARDUINO IDE BMP180



- Bibliothek Adafruit-BMP085-Library

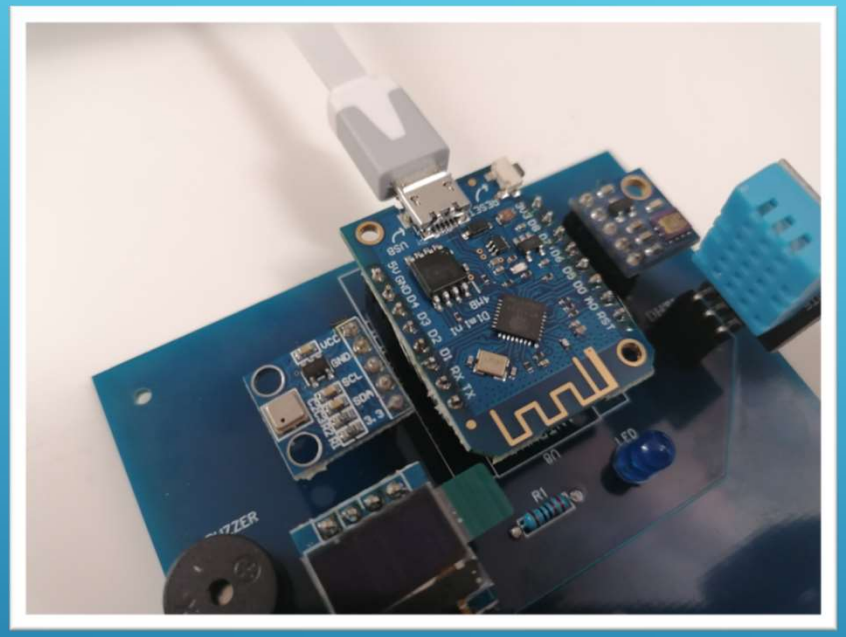
ARDUINO IDE

Installieren der benötigen Bibliothek



- Sensor liefert Werte für
  - Temperatur,
  - Luftdruck,
  - Höhe über den Meeresspiegel

ARDUINO DIE
Sensorwerte

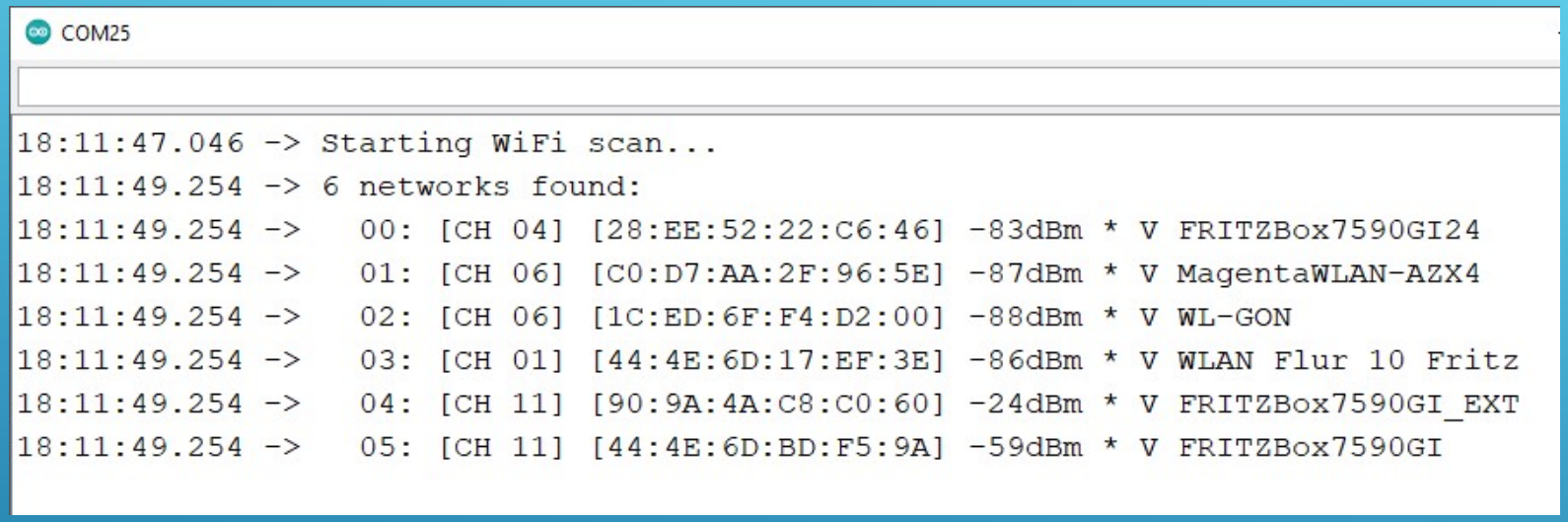


Anzeigen auf dem seriellen Monitor

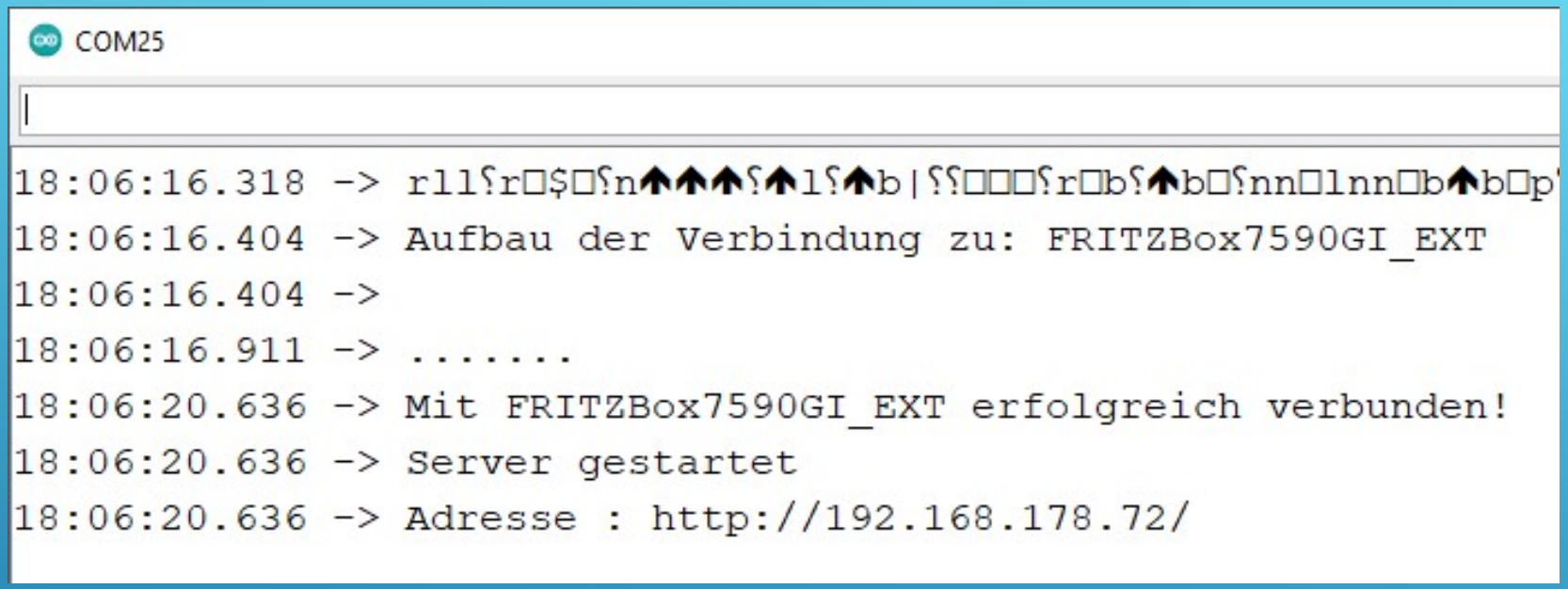
Anzeigen auf dem OLED Display

WEMOS D1 MINI (ESP8266)

Beispiele

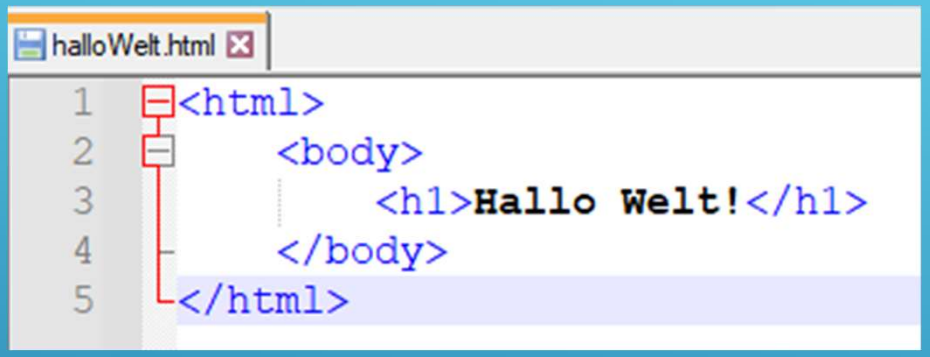


WIFI SCANNER



AUFBAU EINER WI-FI VERBINDUNG

ERSTELLEN EINER WEBSEITE IN HTML



- HTML ist eine Beschreibungssprache keine Programmiersprache!
- Webseiten können
  - dynamischen Code mit Php, JavaScript,
  - Bilder, Videos

  enthalten

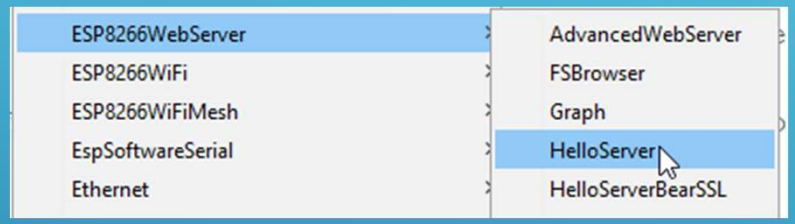
HTML KOMPRIMIEREN

```
1  <html>
2      <body>
3          <h1>Hallo Welt!</h1>
4      </body>
5  </html>
6
7  <html><body><h1>Hallo Welt!</h1></body></html>
```

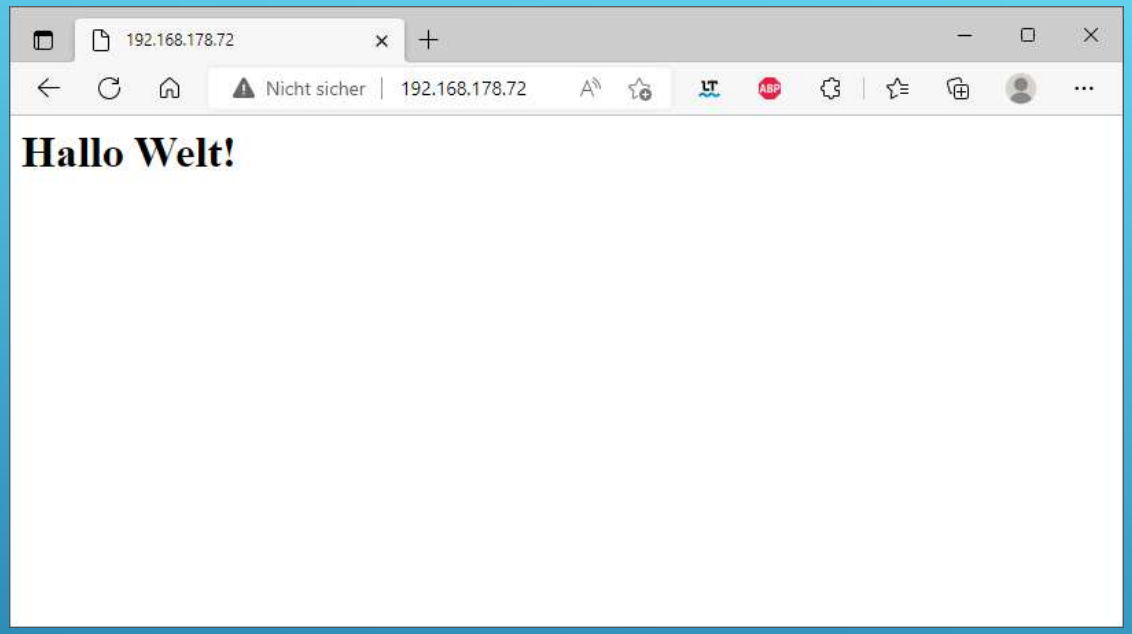
- Reduziert Ladezeiten,
- spart auf dem Mikrocontroller Speicher,
- Internet Service
  - Google.de > „html komprimieren“

ERSTELLEN EINES WEBSERVERS

- „Datei“ > „Beispiele“ > „ESP8266WebServer“ > „HelloServer“

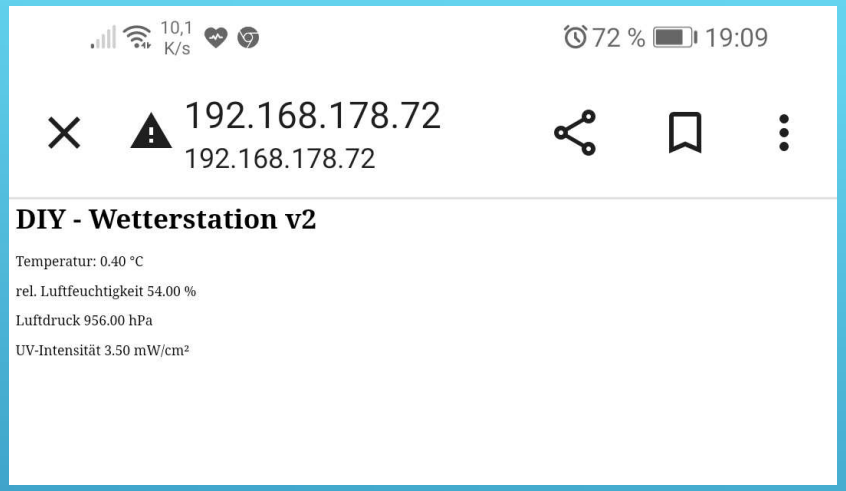



```
server.send(200, "text/plain", "hello from esp8266!\r\n");
server.send(200, "text/html", " <html><body><h1>Hallo Welt!</h1></body></html> ");
```



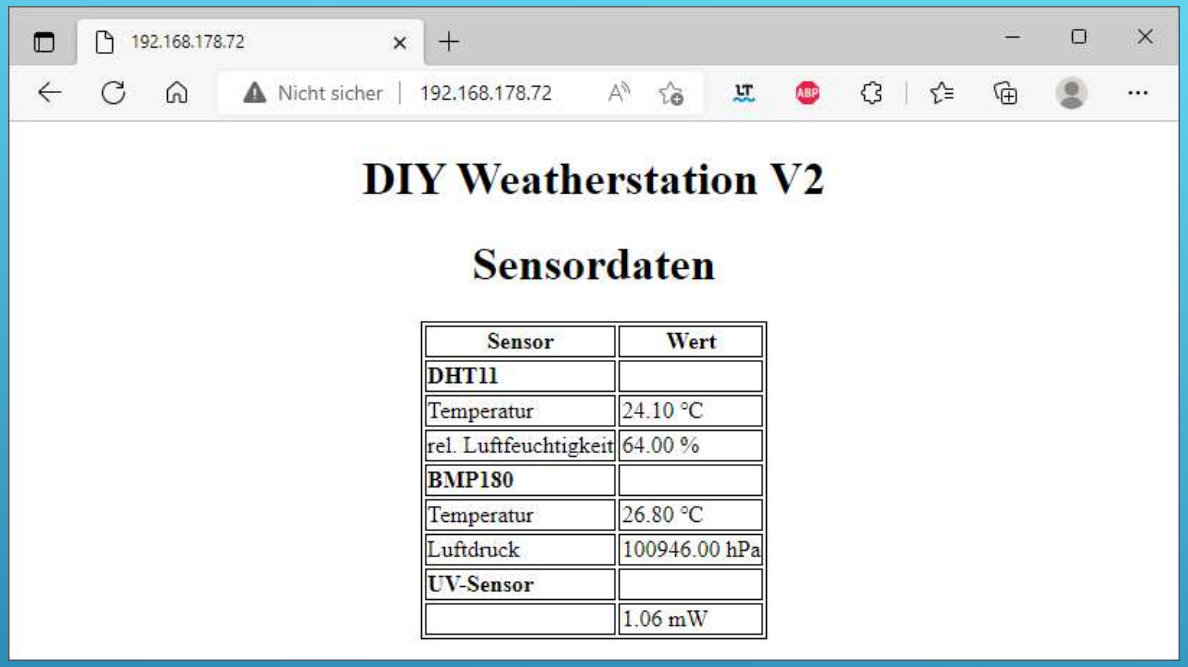
- öffnen der IP-Adresse im Browser
- Anzeige des Textes „Hallo Welt!“

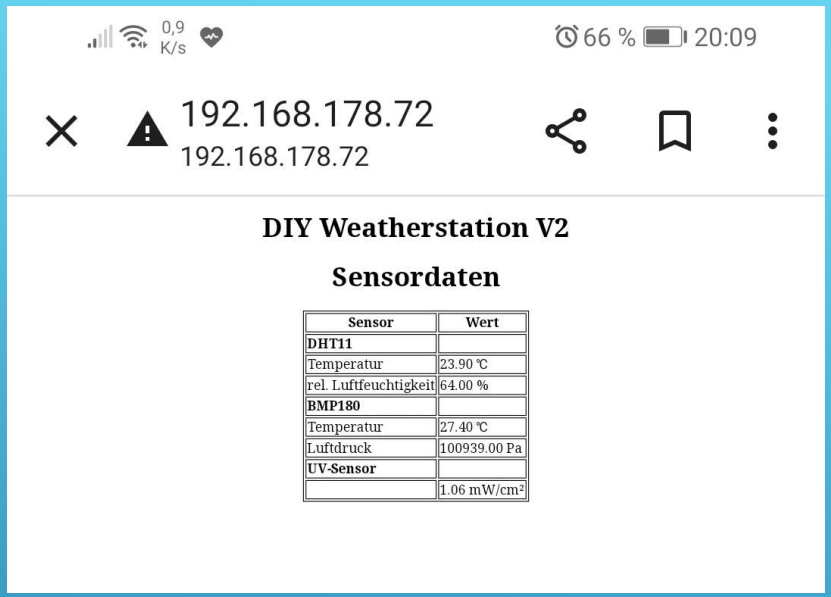
ANZEIGE DER WEBSEITE IM BROWSER



- Adresse im Browser aufrufen Handy & Computer

ERSTELLEN EINER HTML SEITE FÜR SENSORWERTE

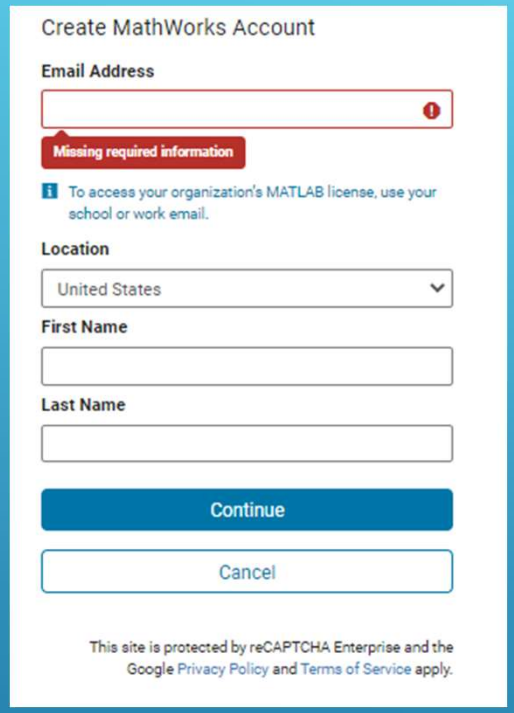




AUSLESEN DER SENSORWERTE UND EINFÜGEN IN DIE SEITE

- Adresse: https://thingspeak.com/

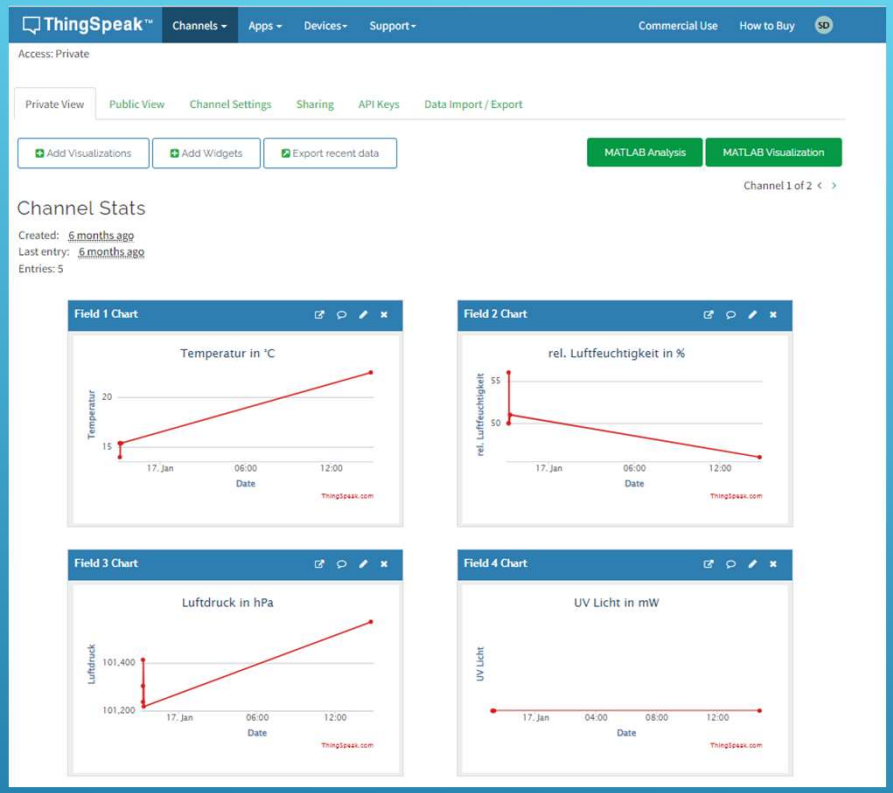
THINGSSPEAK



- Email Address – eine echte E-Mail Adresse
- Location – Germany
- First Name – Vorname
- Last Name - Nachname

THINGSSPEAK

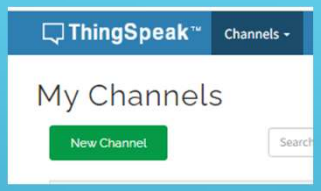
Benutzer anlegen

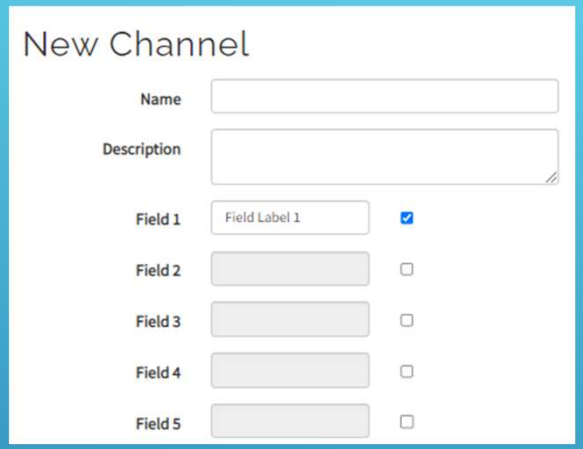


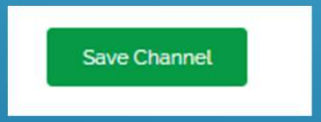
THINGSPEAK

Dashboard

- Liniendiagramme für Sensorwerte
- Export als CSV-Datei
- Einfaches hinzufügen per HTTP-Request (ohne Login)



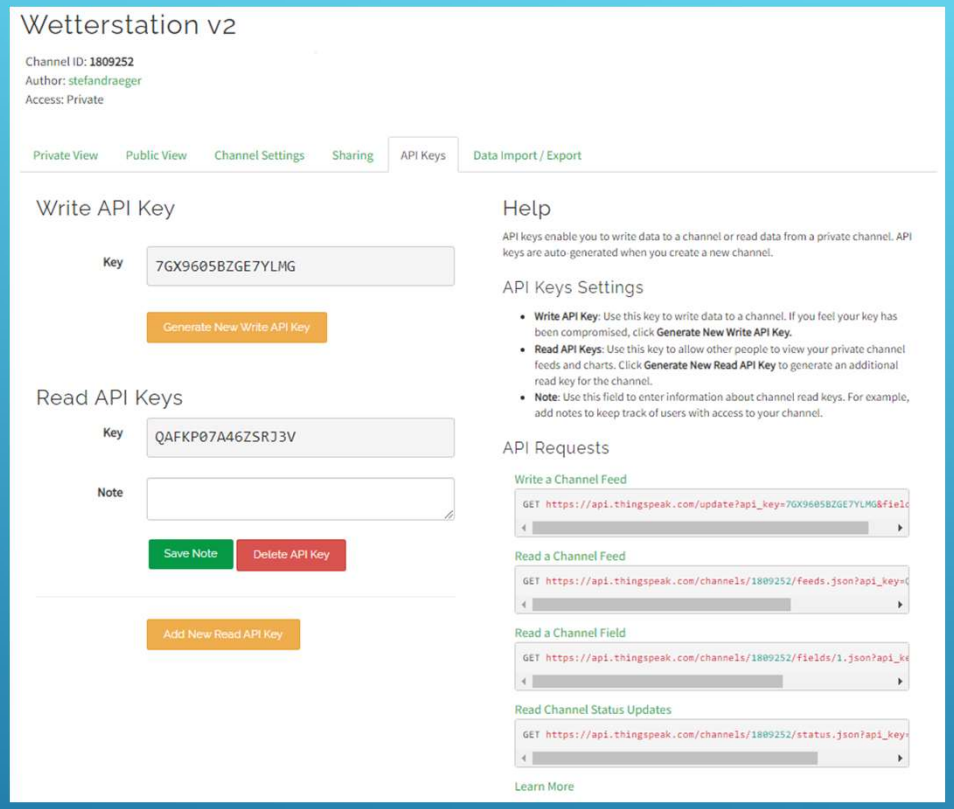




- neuen Channel über „New Channel“ erstellen
- Name – Name des neuen Dashboards,
- Description – Beschreibung (optional)
- Field1 – Temperatur
- Field2 – rel. Luftfeuchtigkeit
- Field3 – Luftdruck
- Field4 - uvIntensitaet

THINGSPEAK

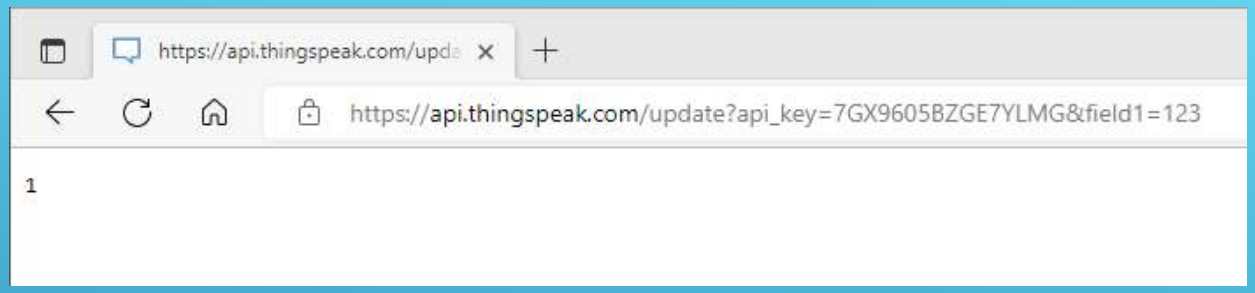
neues Dashboard anlegen

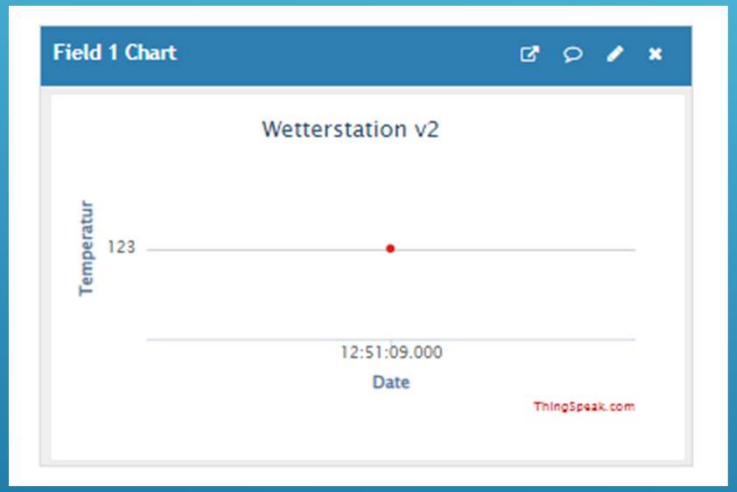


THINGSPEAK

API Key für den Zugriff

- API Key wird benötigt zum validieren des Benutzers
- Daten lesen & schreiben nur mit Key möglich

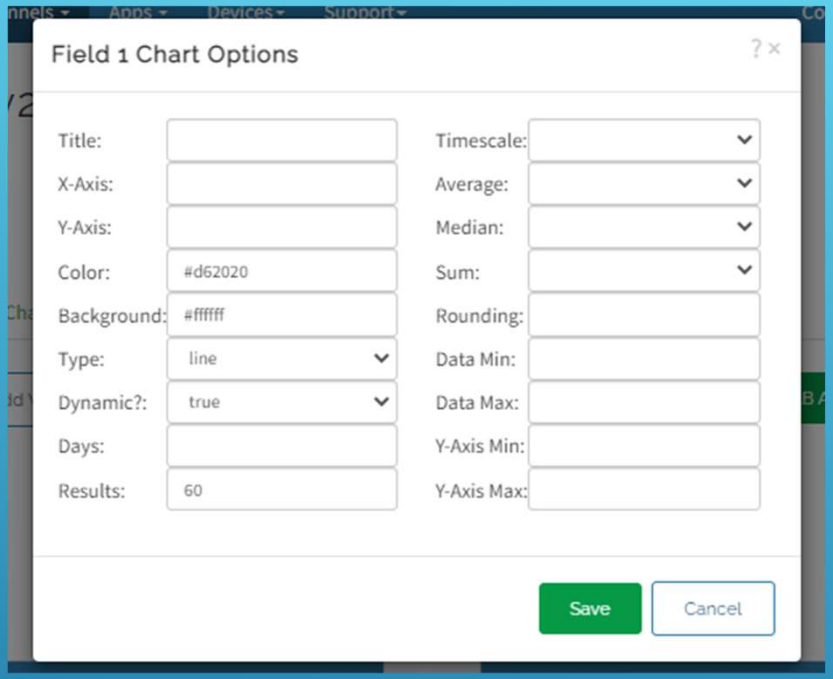




- API Key wird benötigt zum validieren des Benutzers
- Daten lesen & schreiben nur mit Key möglich

BROWSER

Beispiel URL zum einfügen von Daten



- Title – Titel des Diagramms
- X-Axis – Text auf der X-Achse
- Y-Axis – Text auf der Y-Achse
- Color – HTML HEX-Color Wert für die Linie
- Background - HTML HEX-Color Wert für den Hintergrund
- Type: Art des Diagramms
- Data Min: minimaler Wert
  - kann vom Sensor entnommen werden
- Data Max: maximaler Wert
  - kann vom Sensor entnommen werden

THINGSPEAK

Konfiguration der Diagramme

```
String API_KEY = "";
String thingSpeakAPIUrl = "http://api.thingspeak.com/update?api_key="+API_KEY+"&";
```

```
void loop() {
  if ((WiFiMulti.run() == WL_CONNECTED)) {
    String url = thingSpeakAPIUrl;
    url = url + "field1=" + getDht11TempValue();
    url = url + "&field2=" + getDht11HumidityValue();
    url = url + "&field3=" + getBmp180Pressure();
    url = url + "&field4=" + getUvSensorValue();
    http.begin(client, url);

    int httpCode = http.GET();
    Serial.println("HTTP Code:"+httpCode);
    Serial.println("URL:"+url);

    delay(PAUSE);
  }
```

- HTTP Request = Anfrage
- HTTP Response = Antwort
- Adresse wird zusammengefügt aus der Basis und den Sensordaten

ARDUINO IDE

Absenden eines Requests

VIELEN DANK!

info@draeger-it.blog