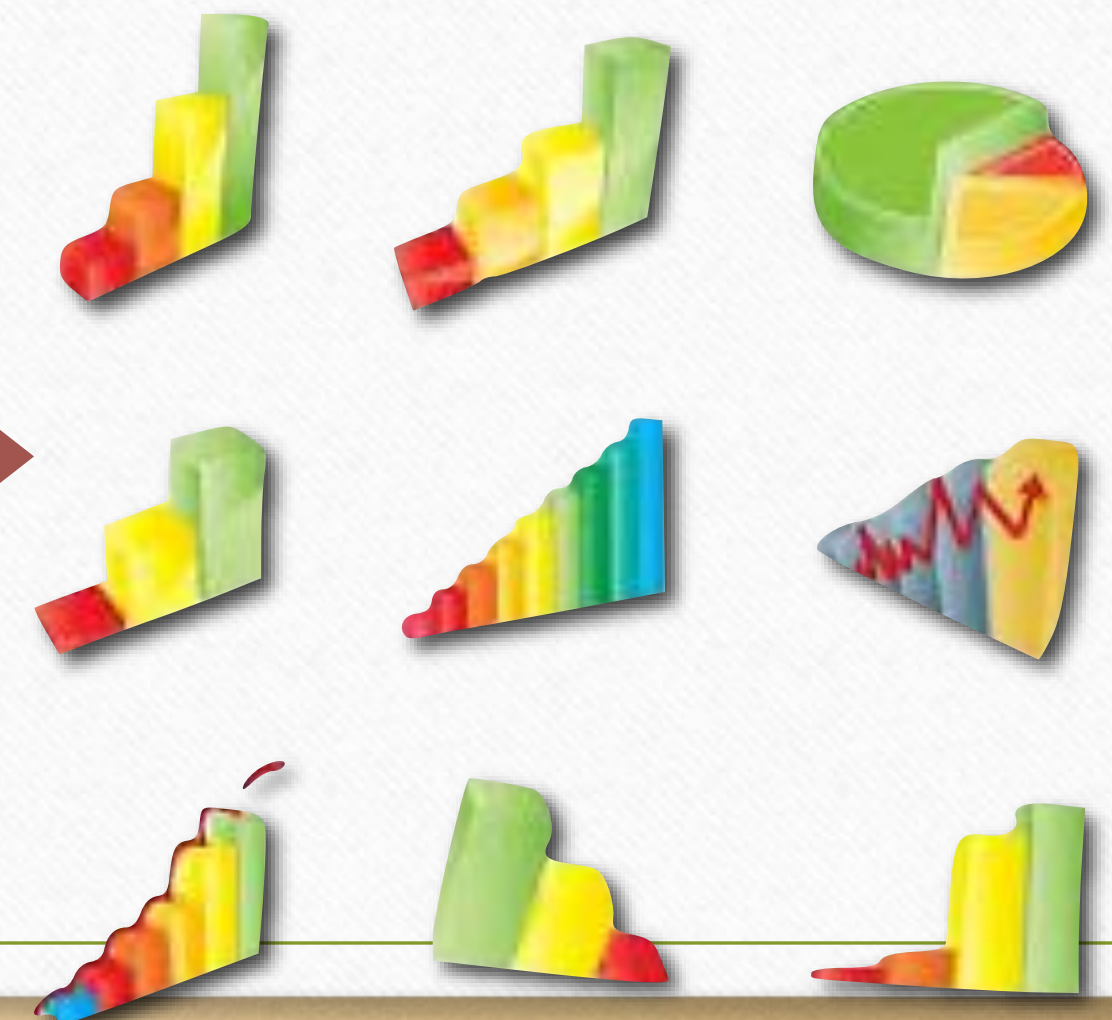
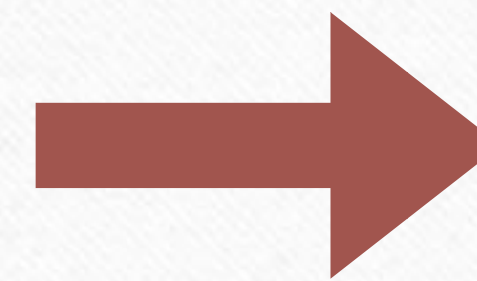
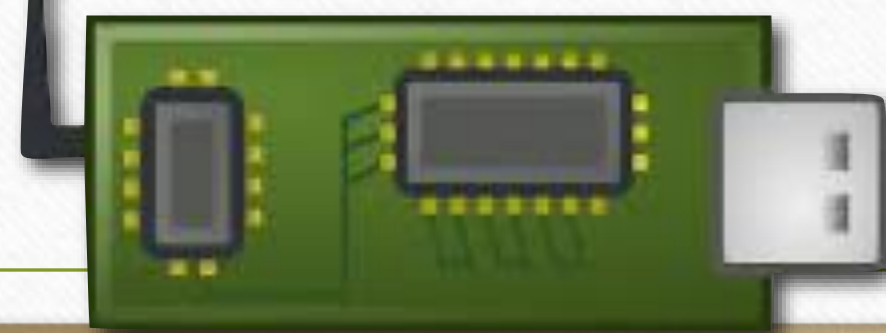
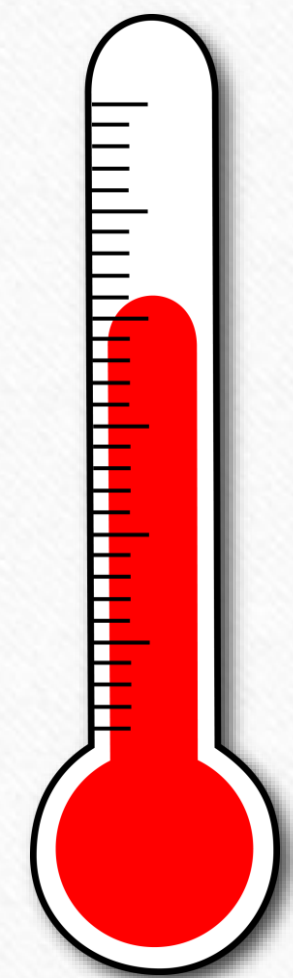


Sensordaten erfassen und weiterverarbeiten

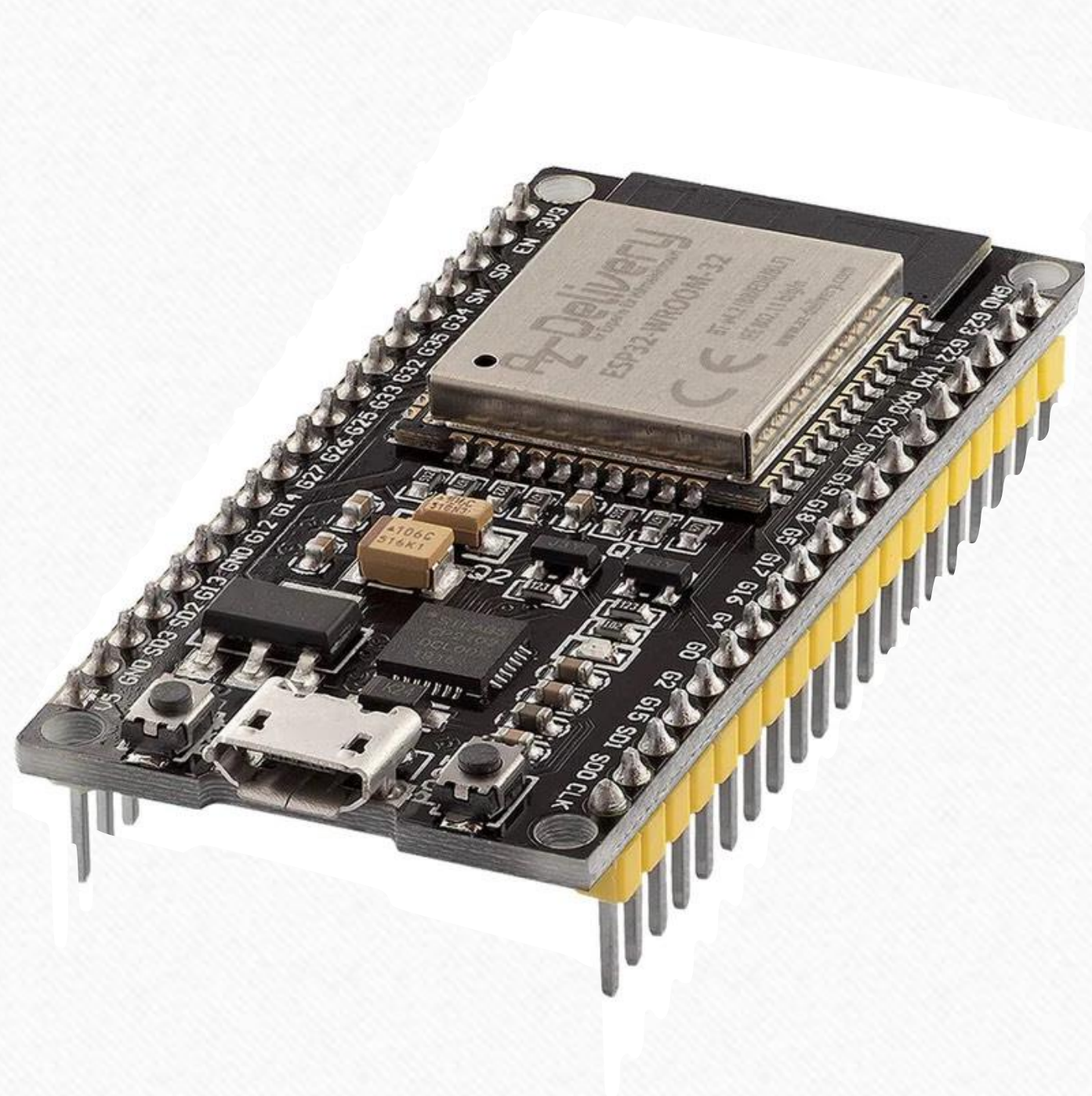
INSY/SYS

Heutige Aufgabe

- Sensordaten erfassen (SYS)
- In gemeinsamer Datenbank speichern (INSY & SYS)
- Daten weiterverarbeiten (INSY)



ESP32



TECHNISCHE DATEN

Stromversorgung

5V (USB)

Arbeitsspannung

3.3V

Schnittstellen

SPI, I2C, I2S,
CAN, UART

Wi-Fi Protokolle

802.11 b/g/n

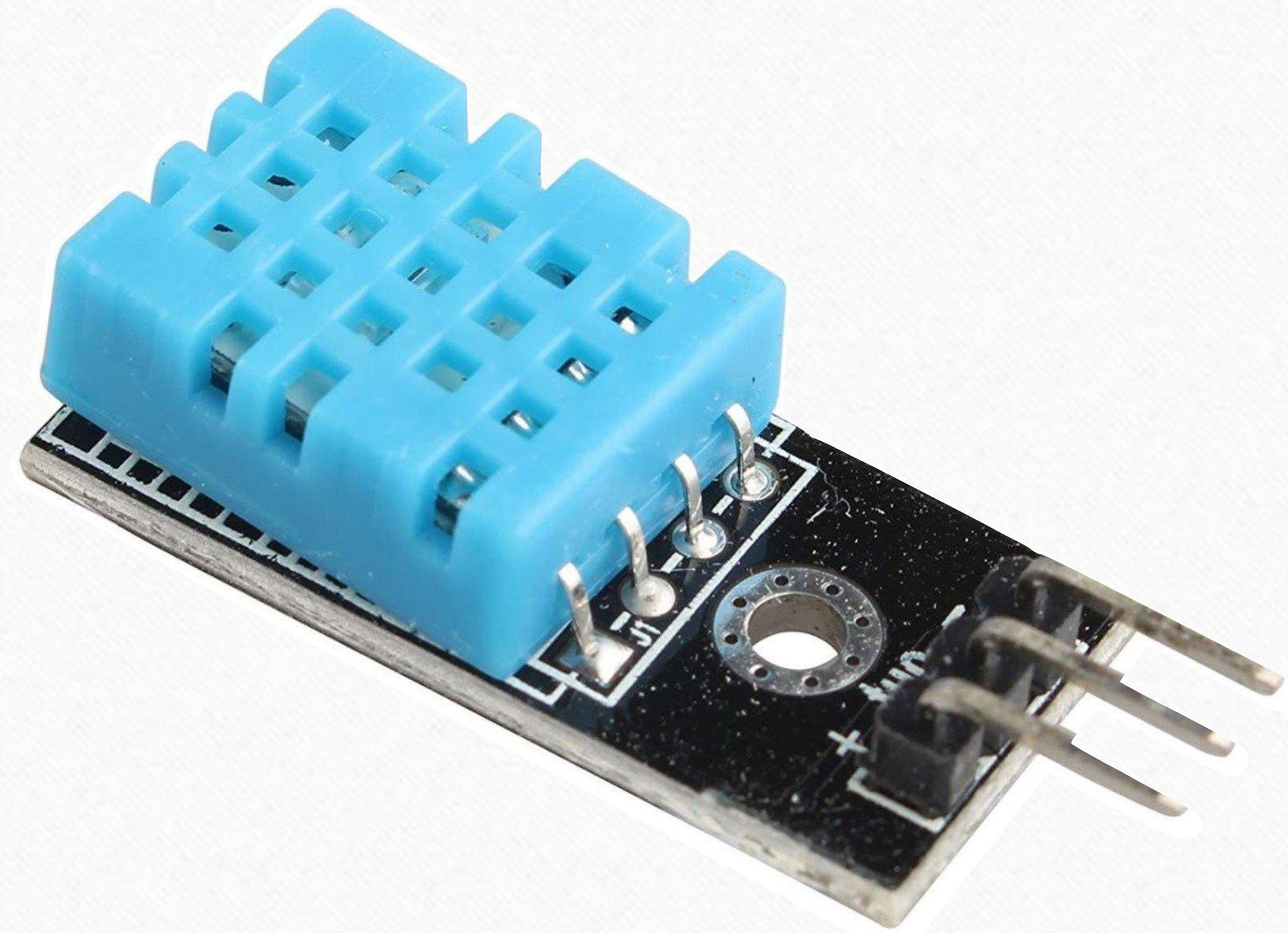
Bluetooth

V4.2

Temperatursensor - DHT11

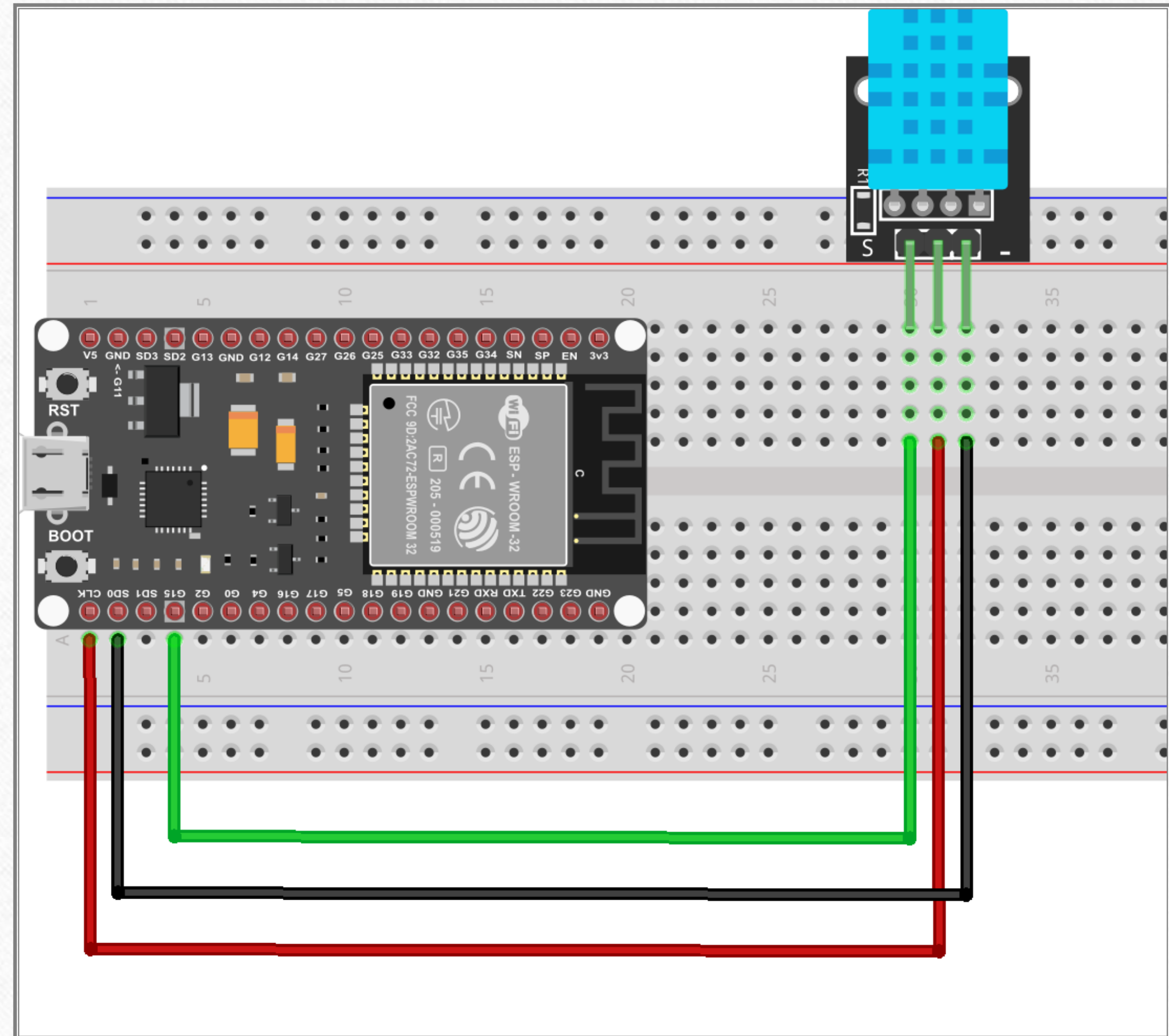
TECHNISCHE DATEN

Temperaturbereich	0 to 50 °C +/-2 °C
Luftfeuchtebereich	20 to 90% +/-5%
Spannung	3 – 5.5 V DC
Preis	\$1 to \$5



Hardwareaufbau

- ESP 32
 - Spannungsversorgung USB
- DHT11
 - 2x Versorgungsleitungen 3,3V
 - Datenleitung - D2



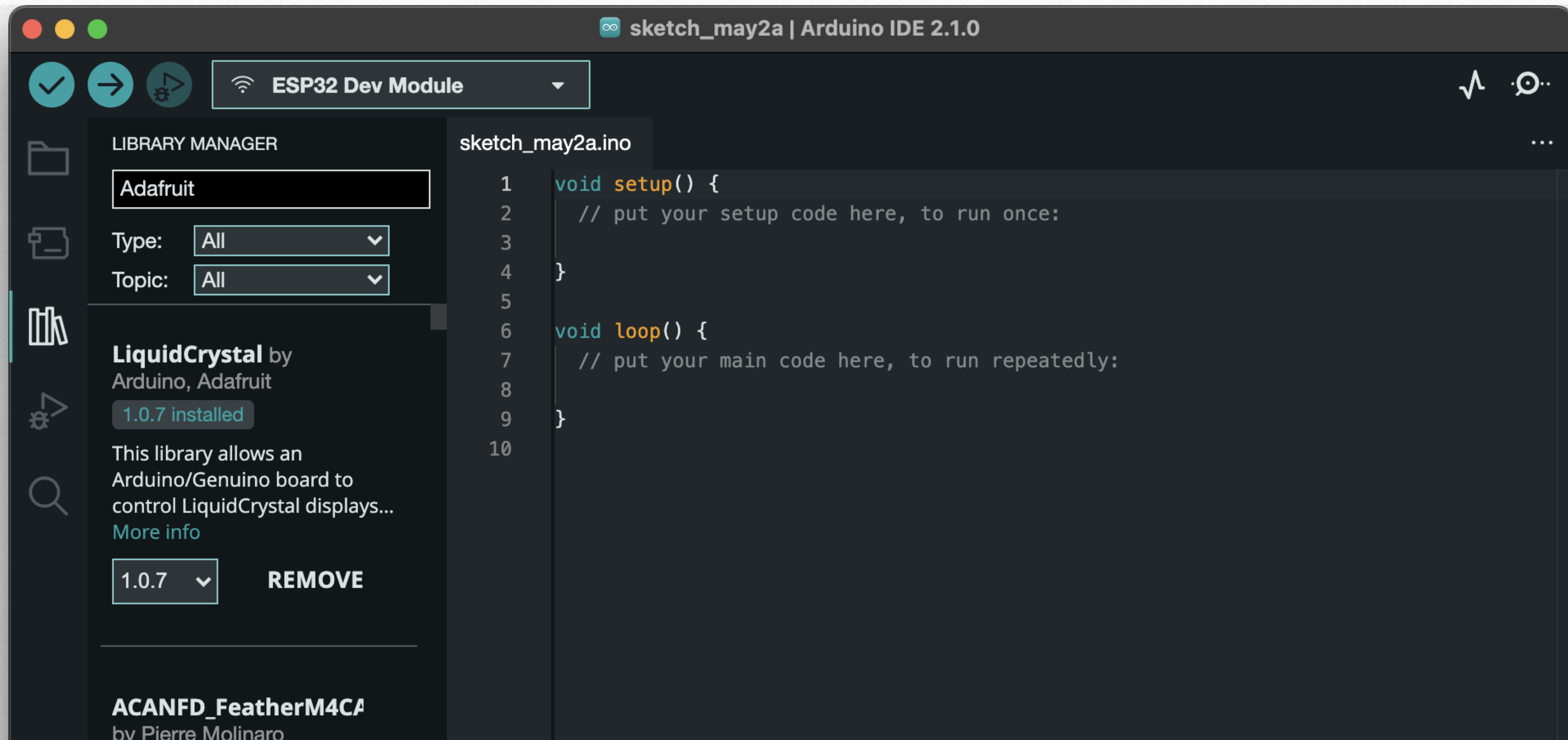
Herangehensweise

Sensordaten lesen

WiFi-Verbindung herstellen

Datenbankverbindung herstellen

Programmteile zusammenführen



ESP32 Dev Module

LIBRARY MANAGER

dht11

Type: All

Topic: All

1.0.3

INSTALL

DHT sensor library by
Adafruit

1.4.4 installed

Arduino library for DHT11,
DHT22, etc Temp & Humidity
Sensors Arduino library for...

[More info](#)

1.4.4

REMOVE

**DHT sensor library for
ESPx** by beegee_tokyo

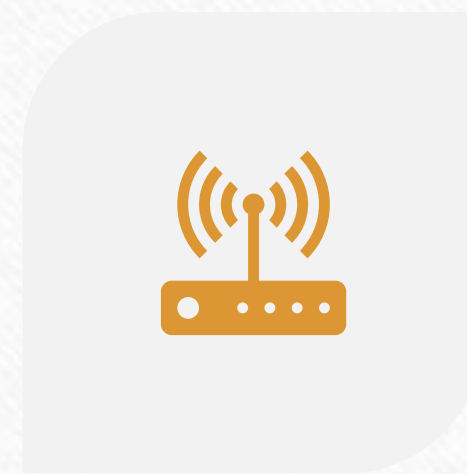
DHTtester.ino

```
1 // Example testing sketch for various DHT humidity/temperature sensors
2 // Written by ladyada, public domain
3
4 // REQUIRES the following Arduino libraries:
5 // - DHT Sensor Library: https://github.com/adafruit/DHT-sensor-library
6 // - Adafruit Unified Sensor Lib: https://github.com/adafruit/Adafruit\_Sensor
7
8 #include "DHT.h"
9
10 #define DHTPIN 2 // Digital pin connected to the DHT sensor
11 // Feather HUZZAH ESP8266 note: use pins 3, 4, 5, 12, 13 or 14 --
12 // Pin 15 can work but DHT must be disconnected during program upload.
13
14 // Uncomment whatever type you're using!
15 // #define DHTTYPE DHT11 // DHT 11
16 #define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
17 // #define DHTTYPE DHT21 // DHT 21 (AM2301)
18
19 // Connect pin 1 (on the left) of the sensor to +5V
20 // NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
21 // to 3.3V instead of 5V!
22 // Connect pin 2 of the sensor to whatever your DHTPIN is
23 // Connect pin 3 (on the right) of the sensor to GROUND (if your sensor has 3 pins)
24 // Connect pin 4 (on the right) of the sensor to GROUND and leave the pin 3 EMPTY (if you
```

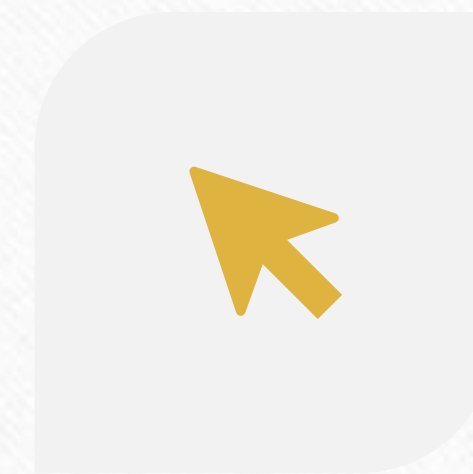

Bibliotheken/Libraries



DHT SENSOR LIBRARY - BY
ADAFRUIT



WIFI CLIENT



MYSQL CONNECTOR ARDUINO
- BY DR. CHARLES BELL

Hilfestellung in Github

<https://github.com/RolandStrohmer/datalogger>

DATEN AUSLESEN & Verarbeiten

MIT NODE.JS & JAVASCRIPT & SQL

1. Verbindung zur Datenbank mittels Node.js herstellen



2. Verarbeiten der erhaltenen Daten mittels SQL & JavaScript, um wertvolle Informationen zu generieren:



Node.js To-Do's:

1. Aktuelle LTS (Long-Term-Support) Version von **Node.js** auf dem PC installieren.
2. NPM (Node-Package-Manager) mitinstallieren
3. Überprüfen ob **NPM** und **Node.js** installiert sind:
 - Hierfür in der CMD **node -v** und **npm -v** eintippen
 - Node sollte **v18.16.0** sein und NPM **v9.5.1**
4. Ordner auf Windows anlegen, und den **Ordnerpfad in der CMD öffnen**
5. NPM in dem Ordner mittels CMD initialisieren -> **npm init**



Connection zur Datenbank mit mySQL:

1. **MySQL** mittels **NPM** im selben Ordner wie zuvor über CMD/Terminal installieren:
 - Befehl: **npm install mysql**
 - Jetzt sollten die Dateien **node_modules**, **package.json** und **package-lock.json** im Ordner ersichtlich sein
2. Ordner im Editor (zB Visual Studio Code) öffnen
3. Neues **JavaScript** File im Ordner **erstellen**
 - Name: **database.js**

Connection zur Datenbank mit Node.js:

1. Als nächstes die Funktion "createPool" von MySQL im JS-File importieren
2. Dach "pool" im File erstellen & Parameter für die Datenbank anpassen:

```
JS database.js ● TS index.d.ts
JS database.js > ...
1  const {createPool} = require('mysql');
2
3  const pool = createPool ({
4    host: "95.128.203.13",
5    user: "web204",
6    password: "t0pUUqS9",
7    database: "usr_web204_4",
8    connectionLimit: 10
9  });
```


Connection zur Datenbank mit Node.js:

1. nun eine Abfrage (query) erstellen die die ganze Datenbank in der Console ausgibt. Dafür den Namen der Datenbank anpassen

```
pool.query('select * from Datensammlung', (err, res, fields) => {  
  if (err) {  
    return console.log(err);  
  }  
  return console.log(res);  
});
```


Übungsaufgaben:

1. Erstelle eine SQL-Abfrage die die ganze Datenbank ausgibt. (GK)
2. Recherchiere wofür die Parameter "err, res, fields" stehen (GK)
3. Erstelle eine SQL-Abfrage, die den Sensor mit dem höchsten Temperaturwert ausgibt (GK)
4. Erstelle eine SQL-Abfrage, die die Durchschnittstemperatur pro Minute ausgibt (EK)
5. Information: Ihr dürft Hilfsmittel aus dem Internet verwenden

SCHREIBE DIE ANWORTEN SCHÖN
DOKUMENTIERT IN DEINEN CODE

HILFESTELLUNG:

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=EIJBSH
3IMB8&T=220S](https://www.youtube.com/watch?v=EIJBSH3IMB8&T=220S)

Jetzt geht es los:

- Schiebt bitte zwei Tische aneinander und bildet somit 7 "Lernstationen"
- Die Gruppeneinteilung erfolgt von uns Lehrern
- Die ersten zwei Personen im Alphabet pro Gruppe beginnen mit der SYT-Aufgabe
- Die zweiten zwei Personen im Alphabet pro Gruppe beginnen mit der INSY-Aufgabe
- Führt bitte die Aufgaben innerhalb der nächsten 30 Minuten durch
- Nach 30 Minuten wechselt ihr bitte von eurer bisherigen Aufgabe zur Anderen. Also von SYT -> INSY oder umgekehrt :)

Jetzt geht es los:

- Viel Spaß bei der heutigen Unternehmung :)
- Bei Fragen könnt ihr euch immer an uns Lehrer wenden.