

Bau dir einen mobilen Feinstaubsensor

piko & melzai

(Geekfem Hamburg, Haecksen, CCCHH)

Feinstaub

- Teilchen in der Luft, die nicht sofort zu Boden sinken
- Definition 1959, erste EU-Richtlinie 1980
- Früher einfach Gewicht angegeben
- *Particulate matter*

PM₁₀ und PM_{2,5}

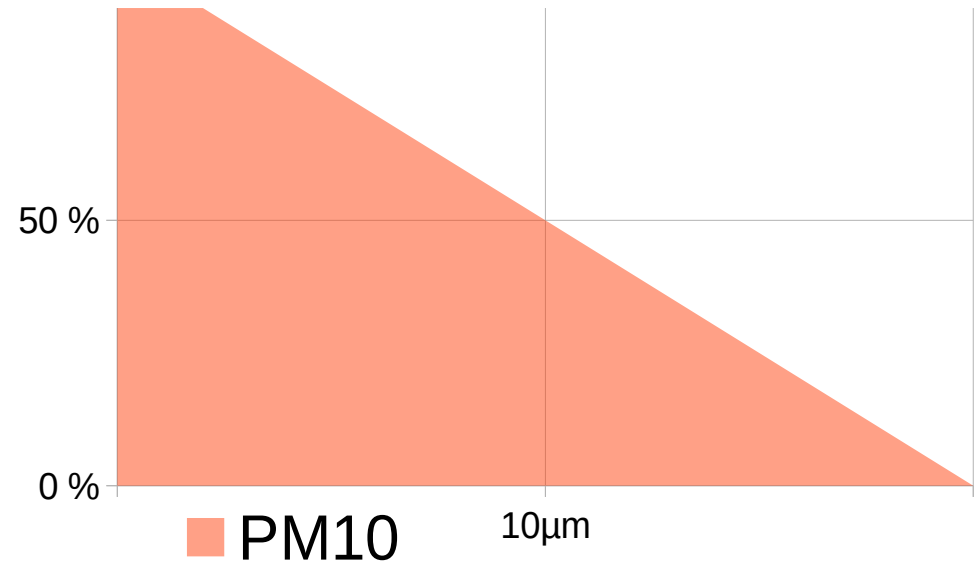
- Von *particulate matter*
- Standard der US-amerikanischen Umweltschutzbehörde EPA
- PM₁₀
- PM_{2,5}
- Ultrafeine Teilchen bis 0,1µm



<https://www.epa.gov/pm-pollution/particulate-matter-pm-basics>

PM₁₀ und PM_{2,5}

- Von *particulate matter*
- Standard der US-amerikanischen Umweltschutzbehörde EPA
- PM₁₀
- PM_{2,5}
- Ultrafeine Teilchen bis 0,1µm



Anteil der gewichteten Teilchen

Grenzwerte

	EU – PM ₁₀	EU – PM _{2,5}	WHO – PM ₁₀	WHO – PM _{2,5}
Jahresmittelwert in µg/m ³	40	25	20	10
Tagesmittelwert in µg/m ³	50		50	25
Tage mit Überschreitungen	35		keine	keine

Grenzwerte

	EU – PM ₁₀	EU – PM _{2,5}	WHO – PM ₁₀	WHO – PM _{2,5}
Jahresmittelwert in µg/m ³	40	25	20	10
Tagesmittelwert in µg/m ³	50		50	25
Tage mit Überschreitungen	35		keine	keine

Grenzwerte

	EU – PM ₁₀	EU – PM _{2,5}	WHO – PM ₁₀	WHO – PM _{2,5}
Jahresmittelwert in µg/m ³	40	25	20	10
Tagesmittelwert in µg/m ³	50		50	25
Tage mit Überschreitungen	35		keine	keine

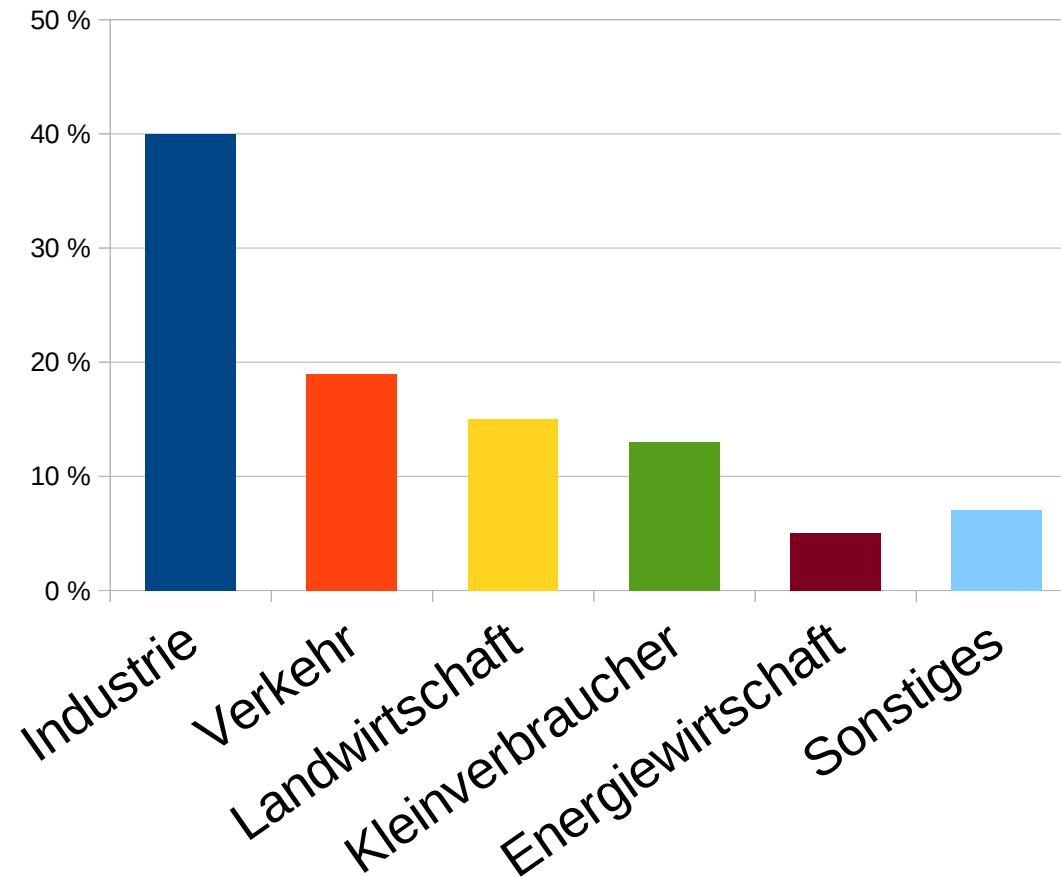
Grenzwerte

	EU – PM ₁₀	EU – PM _{2,5}	WHO – PM ₁₀	WHO – PM _{2,5}
Jahresmittelwert in µg/m ³	40	25	20	10
Tagesmittelwert in µg/m ³	50		50	25
Tage mit Überschreitungen	35		keine	keine

Grenzwerte

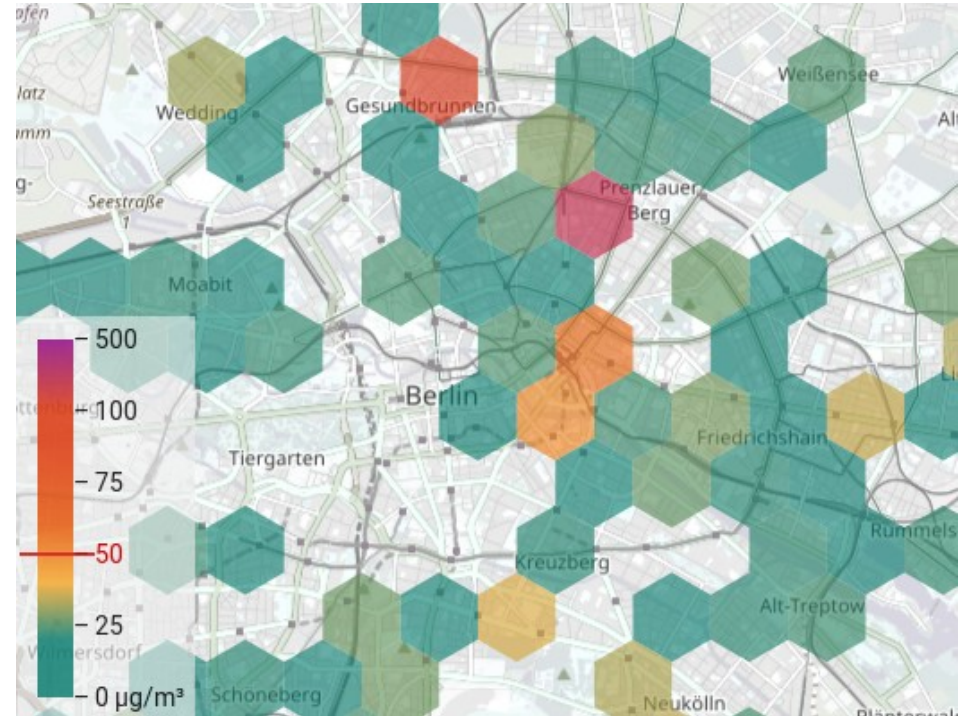
	EU – PM ₁₀	EU – PM _{2,5}	WHO – PM ₁₀	WHO – PM _{2,5}
Jahresmittelwert in µg/m ³	40	25	20	10
Tagesmittelwert in µg/m ³	50		50	25
Tage mit Über- schreitungen	35		keine	keine

Nicht-natürliche Feinstaubquellen



Luftdaten.info

- Stationärer Feinstaubsensor
- Projekt des OK Lab in Stuttgart
- Weitgehend gleiche Hardware
- Karte auf Website



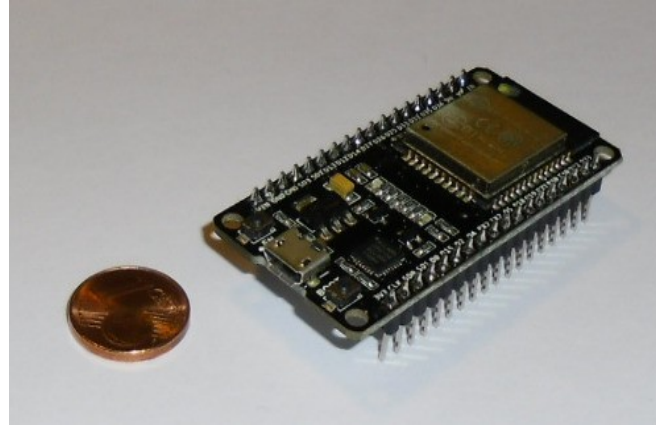
Teile

- Chip (Mikrocontroller)– ESP32
- Luftfeuchtigkeits- und Temperatursensor – DHT22
- Feinstaubsensor – SDS011
- Widerstand – 4,7k Ω

- Powerbank
- Smartphone

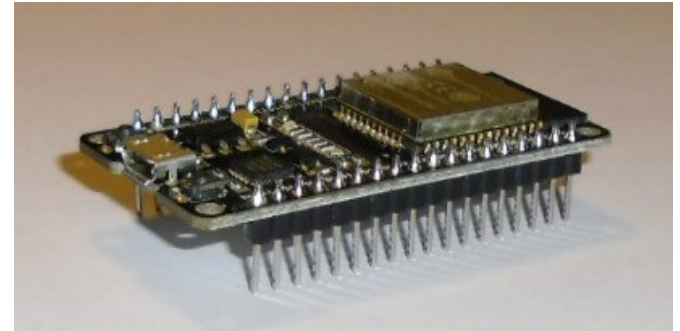
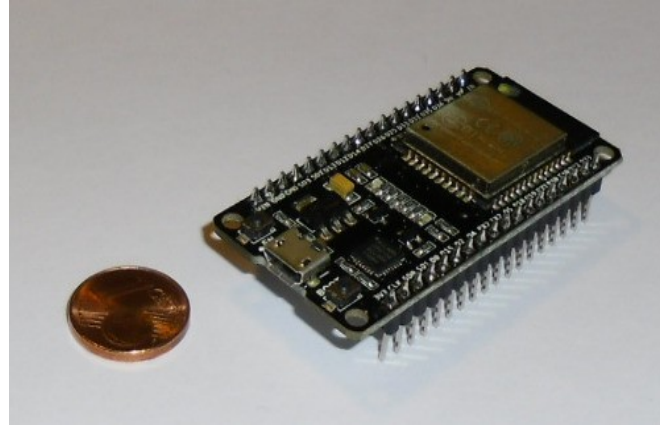
ESP32

- Mikrocontroller
- Mikro-USB
- 48 Pins
- klein und günstig



ESP32

- Mikrocontroller
- Mikro-USB
- 48 Pins
- klein und günstig



ESP32

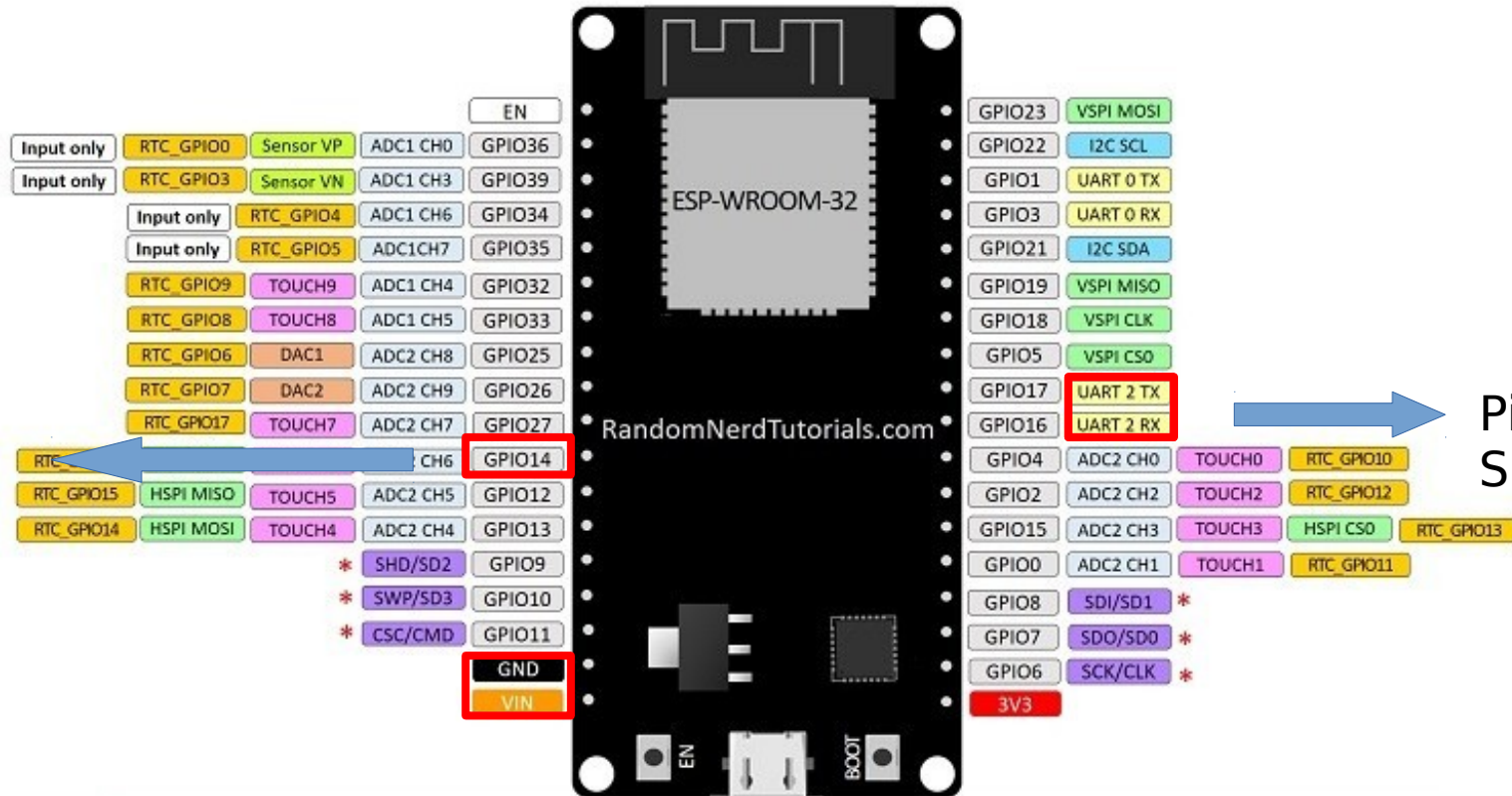
- Mikrocontroller
- Mikro-USB
- 48 Pins
- klein und günstig



ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs

Pin 14
DHT22

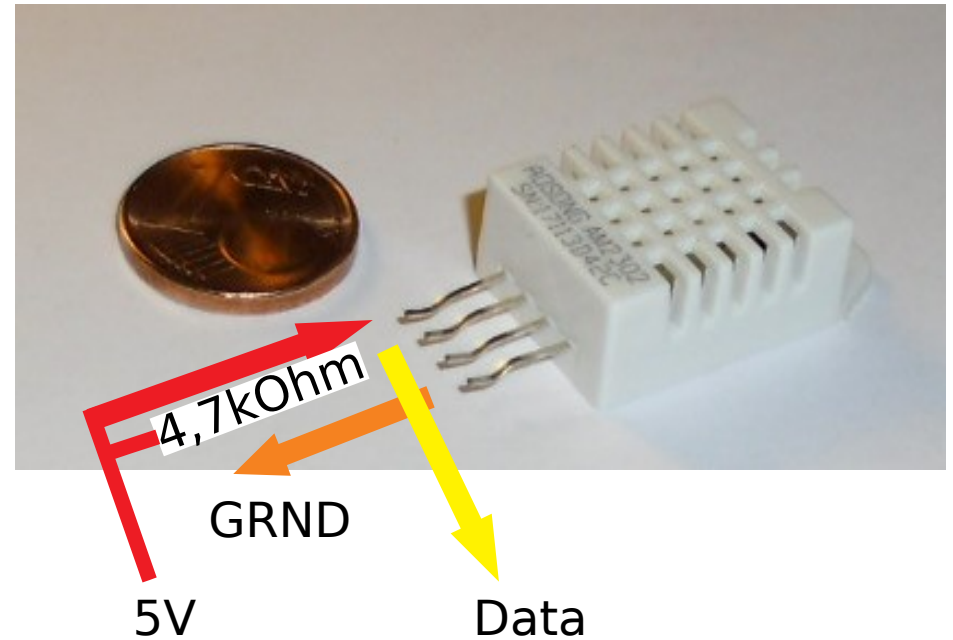


Pin 16, 17
SDS011

* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

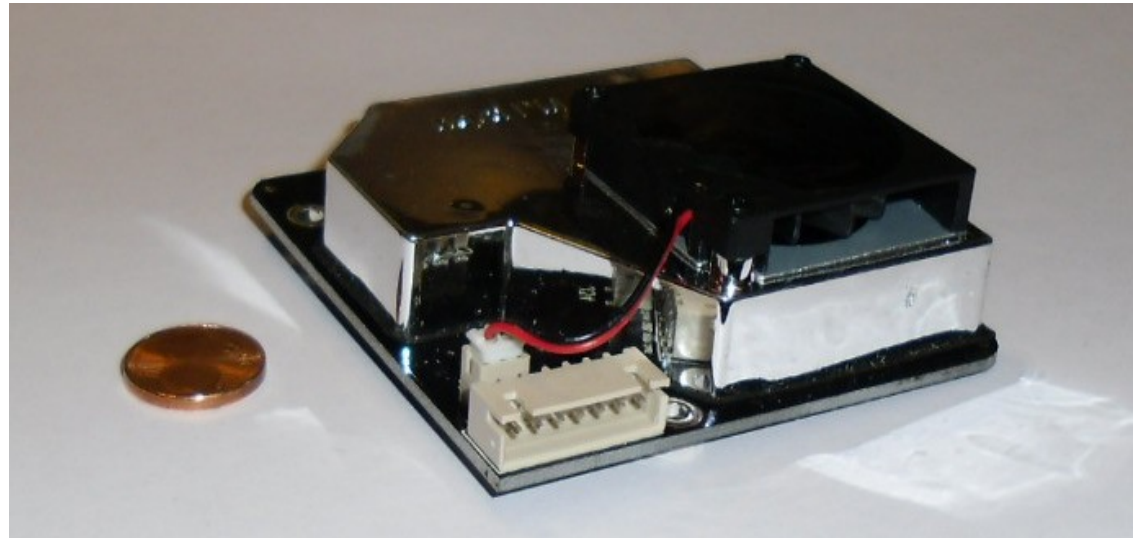
DHT22

Feuchtigkeits- und
Temperatursensor

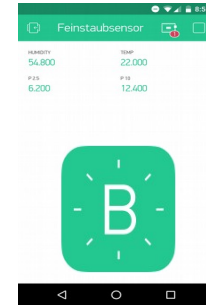
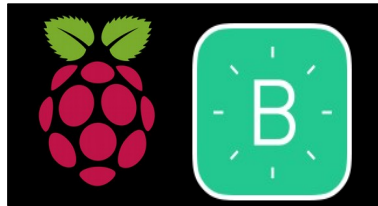
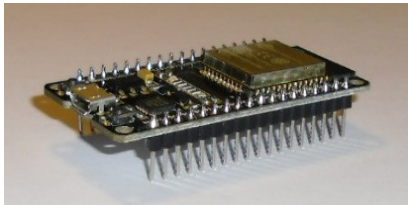
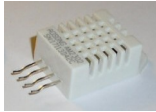
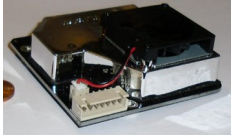


SDS011

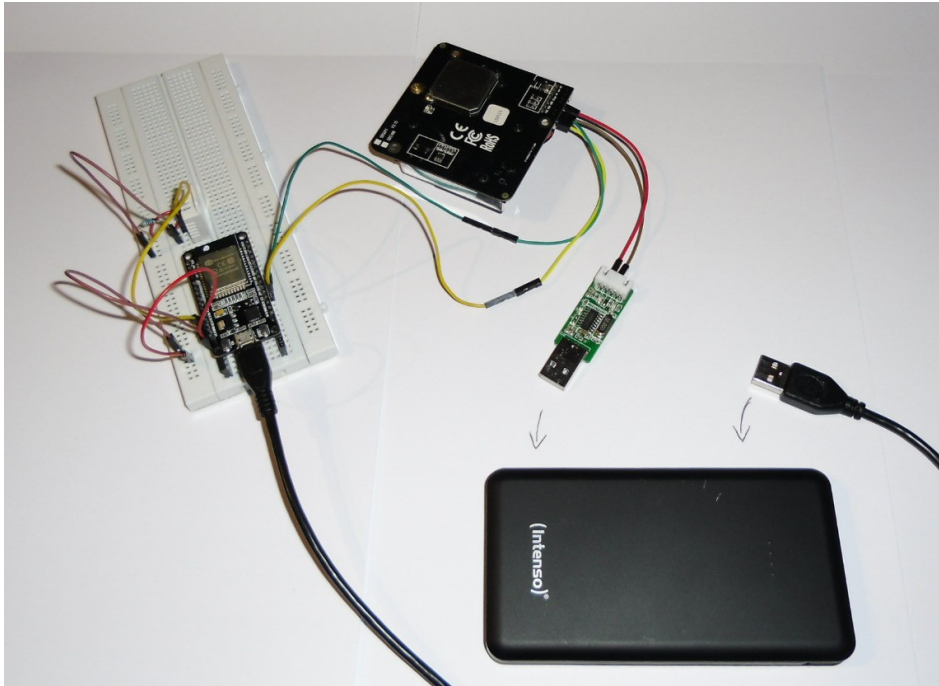
- Feinstaubsensor
- 0,3 bis 10 μm PM_{10} und $\text{PM}_{2,5}$
- Lüfter



Der Bauplan



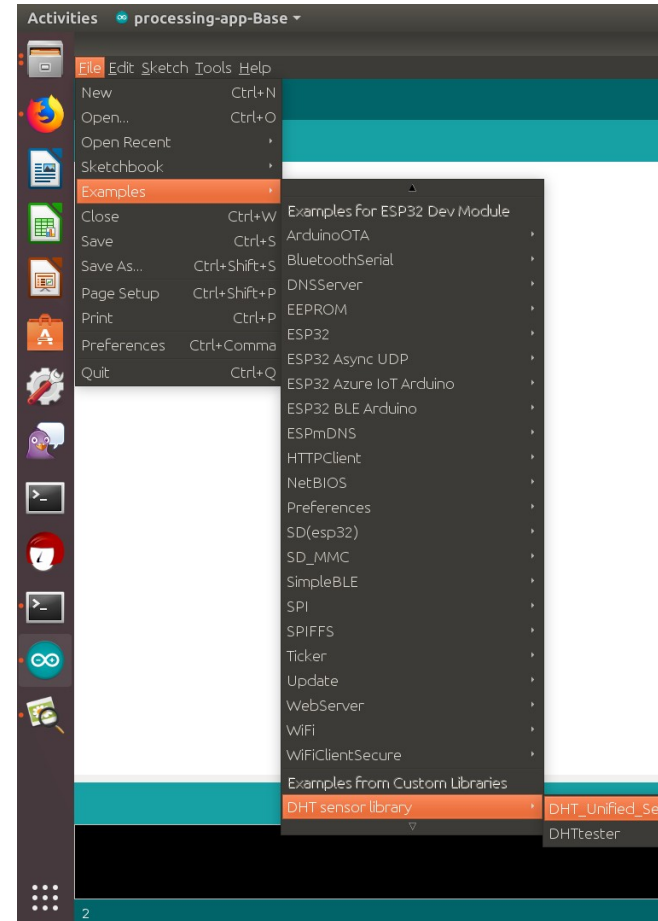
Der Bauplan



- SDS011 R → ESP32 T 17
- SDS011 T → ESP32 R 16
- DHT22 Data → ESP32 Pin 14
- DHT Strom, GRND → ESP32
VIN, GRND

Hardware needs Software

- Arduino IDE
<https://www.arduino.cc>
- Installiere ESP32 Kern
<https://github.com/espressif/arduino-esp32>
nach Arduino/hardware
- Schließe ESP32 an, lade zB. das Beispiel „Wifi Scan“ auf den Chip
- Seriellen Monitor öffnen, Scanergebnisse anschauen



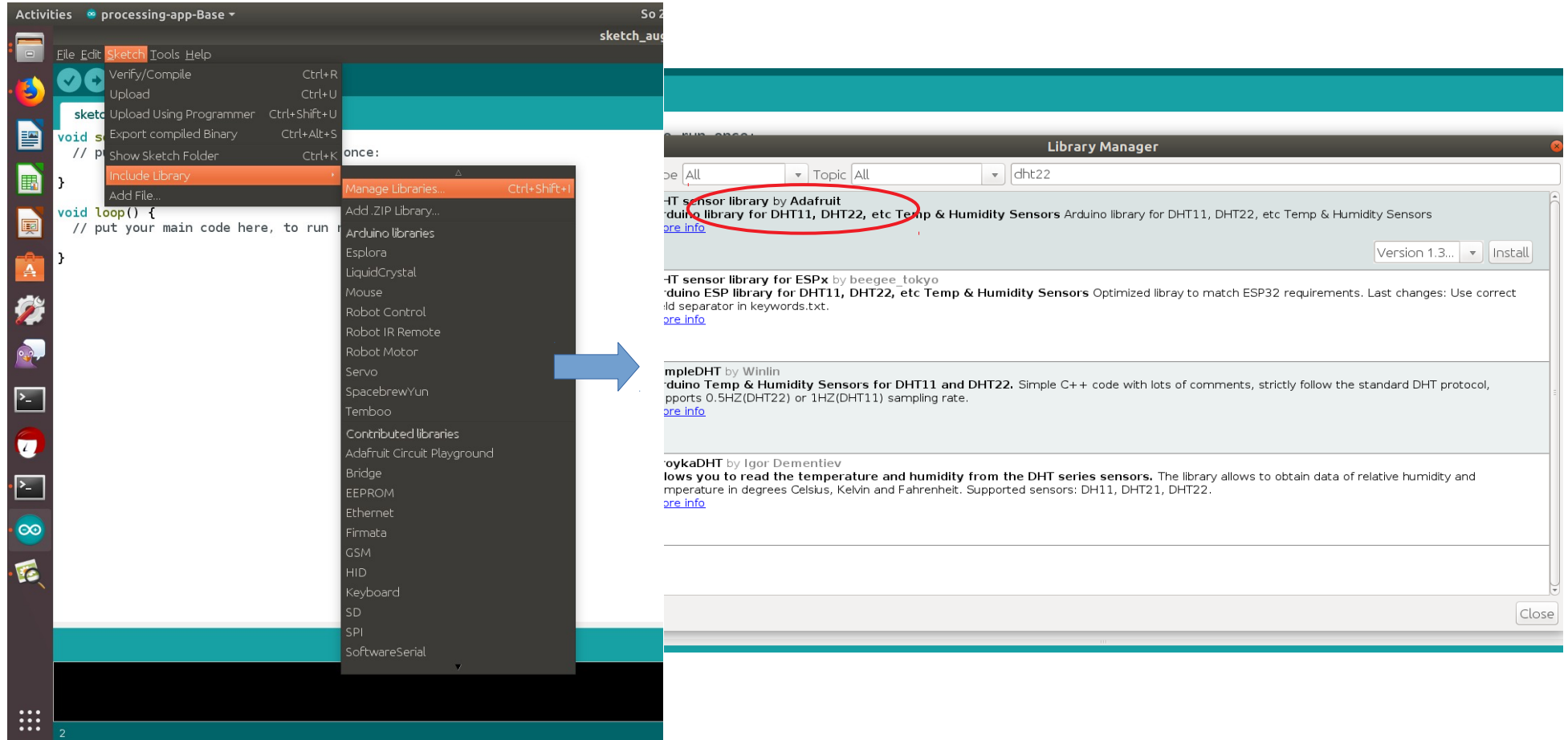
Bibliotheken, Variablen, Methoden

- Bibliotheken: Häuser mit Funktionen wie zB ein Krankenhaus
 - Eine Tür pro Station, eine kranke Person geht hinein, eine gesunde kommt (hoffentlich) heraus
- Variablen, Objekte: Baue ein echtes Krankenhaus, speichere eine Zahl
- Methoden: Tue Dinge, handle den Patienten mit einer Komresse

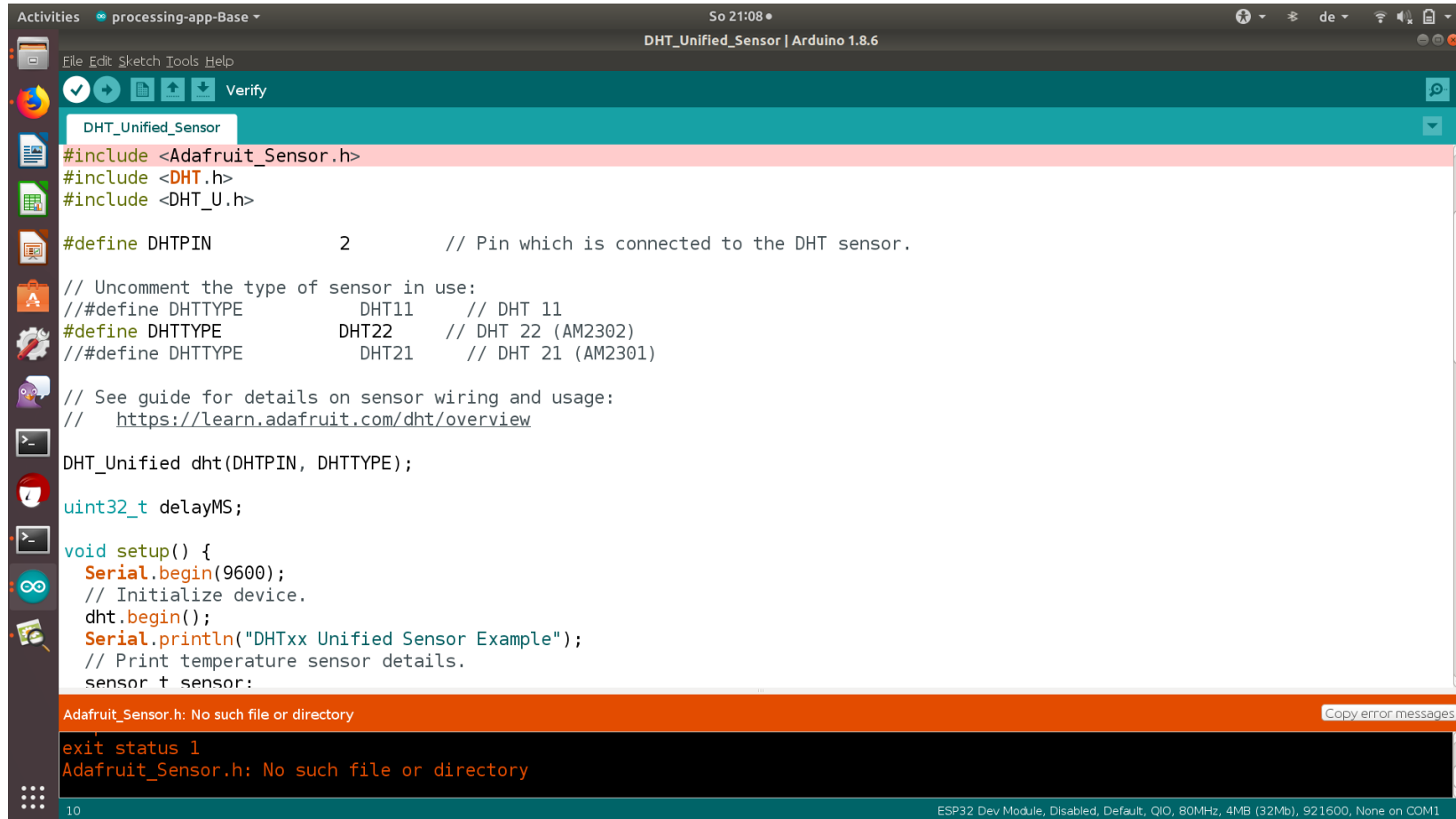
Bibliotheken

- Bibliotheken: Häuser mit Funktionen wie zB ein Krankenhaus
 - Eine Tür pro Station, eine kranke Person geht hinein, eine gesunde kommt (hoffentlich) heraus
- Eine Bibliothek pro Sensor, eine für WiFi, eine für Blynk
- Liste aller registrierten Bibliotheken für den ESP32 in der Arduino IDE
<https://www.arduino-libraries.info/architectures/esp32>
- Kein Blynk, kein SDS011??
<https://www.blynk.cc/>
<https://github.com/sleiner/SDS011>

Bibliotheken installieren

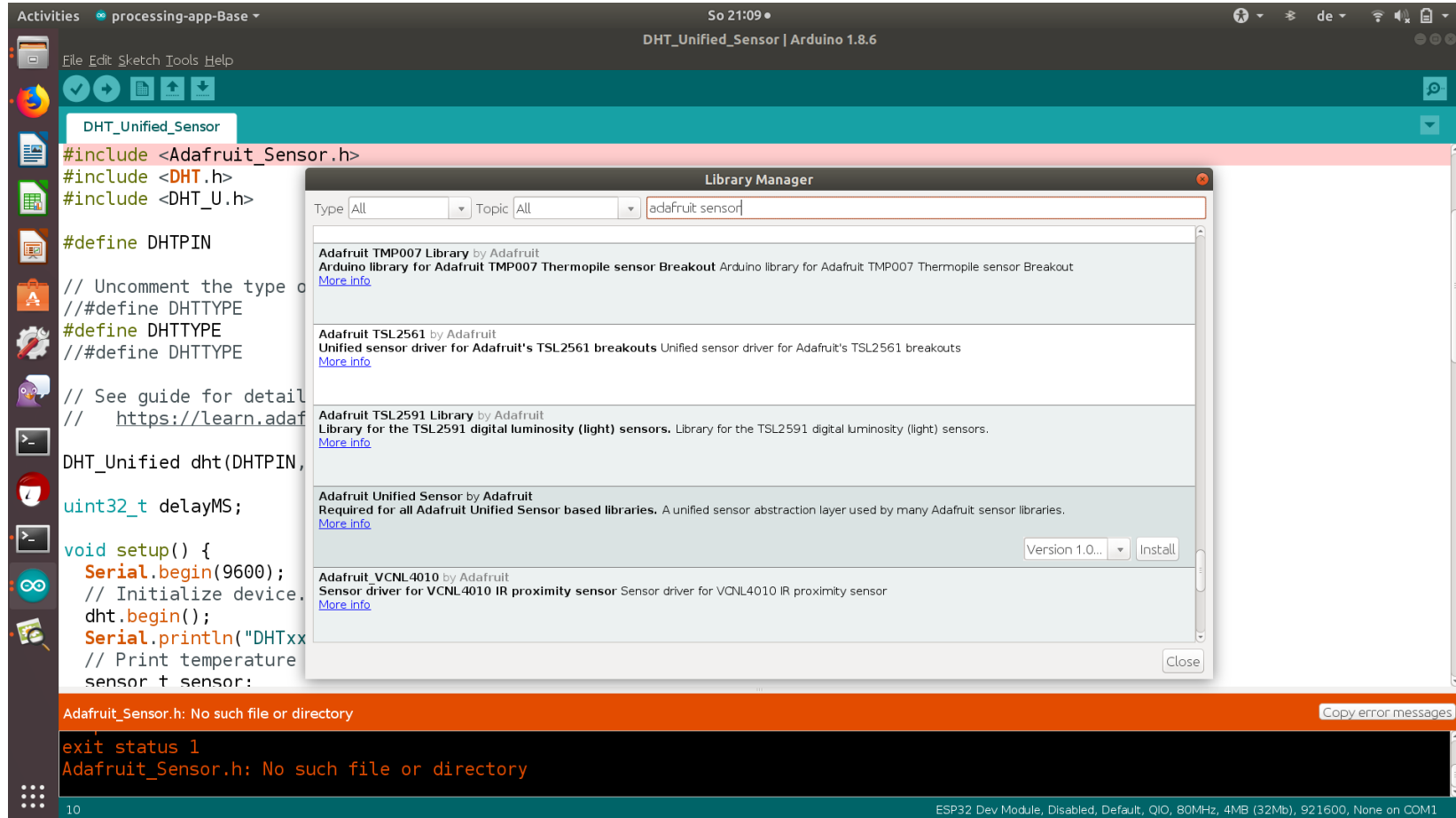


Übersetzen in Maschinensprache

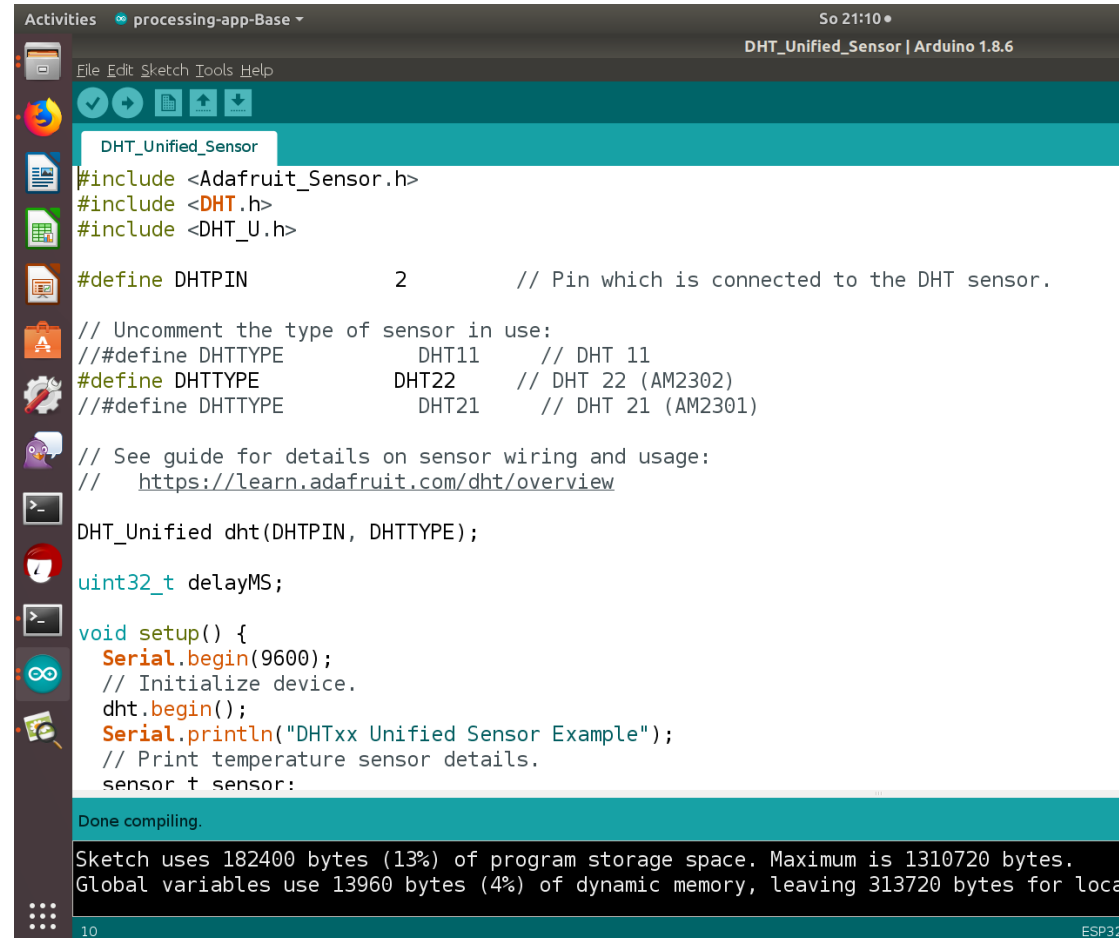


```
Activities  processing-app-Base  So 21:08 •  
DHT_Unified_Sensor | Arduino 1.8.6  
File Edit Sketch Tools Help  
Verify  
DHT_Unified_Sensor  
#include <Adafruit_Sensor.h>  
#include <DHT.h>  
#include <DHT_U.h>  
  
#define DHTPIN 2 // Pin which is connected to the DHT sensor.  
  
// Uncomment the type of sensor in use:  
// #define DHTTYPE DHT11 // DHT 11  
#define DHTTYPE DHT22 // DHT 22 (AM2302)  
// #define DHTTYPE DHT21 // DHT 21 (AM2301)  
  
// See guide for details on sensor wiring and usage:  
// https://learn.adafruit.com/dht/overview  
  
DHT_Unified dht(DHTPIN, DHTTYPE);  
  
uint32_t delayMS;  
  
void setup() {  
  Serial.begin(9600);  
  // Initialize device.  
  dht.begin();  
  Serial.println("DHTxx Unified Sensor Example");  
  // Print temperature sensor details.  
  sensor_t sensor;  
  
  Adafruit_Sensor.h: No such file or directory  
  exit status 1  
  Adafruit_Sensor.h: No such file or directory  
10  
ESP32 Dev Module, Disabled, Default, QIO, 80MHz, 4MB (32Mb), 921600, None on COM1
```

Ups, Adafruit Unified Sensor Lib nachinstallieren



Nochmal versuchen - Erfolg!



The screenshot shows the Arduino IDE interface with the 'DHT_Unified_Sensor' sketch loaded. The code is as follows:

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN 2 // Pin which is connected to the DHT sensor.

// Uncomment the type of sensor in use:
// #define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302)
// #define DHTTYPE DHT21 // DHT 21 (AM2301)

// See guide for details on sensor wiring and usage:
// https://learn.adafruit.com/dht/overview

DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;

void setup() {
  Serial.begin(9600);
  // Initialize device.
  dht.begin();
  Serial.println("DHTxx Unified Sensor Example");
  // Print temperature sensor details.
  sensor_t sensor;
```

Below the code editor, a status bar indicates 'Done compiling.' and provides memory usage details: 'Sketch uses 182400 bytes (13%) of program storage space. Maximum is 1310720 bytes. Global variables use 13960 bytes (4%) of dynamic memory, leaving 313720 bytes for local variables.' The bottom status bar shows '10' and 'ESP32'.

Bibliotheken einbinden

```
#include <WiFi.h>  
#include <WiFiClient.h>  
#include <BlynkSimpleEsp32.h>
```

```
#include <SDS011.h>
```

```
#include <DHT.h>
```

Objekte bauen, Pins deklarieren

```
#define BLYNK_PRINT Serial

// You should get Auth Token in
// the Blynk App.
char auth[] = " AAAAAAAAAAAAAA";

// Your WiFi credentials.
char ssid[] = "WIFI NAME";
char pass[] = "WIFI PWD";
BlynkTimer timer;
```

```
float p10, p25;

int errorflag;

HardwareSerial sdsSerial(2); //
use Serial 2: RX pin:16, TX pin:17

SDS011 my_sds;

#define DHTPIN 14
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
```

Methoden: Sensoren auslesen und senden

```
void readDHT()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read
            from DHT sensor!");
        return;
    }

    Blynk.virtualWrite(V5, h);
    Blynk.virtualWrite(V6, t);
}
```

```
void readSDS()
{
    errorflag = my_sds.read(&p25,
        &p10);
    if (errorflag) {
        Serial.println("Failed to read
            from SDS sensor!");
        return;
    }
    Serial.println("P2.5: " +
        String(p25));
    Serial.println("P10:  " +
        String(p10));
    Blynk.virtualWrite(V7, p25);
    Blynk.virtualWrite(V8, p10);
}
```

Woher weiß der Chip, was er zu tun hat?

```
void setup() {  
.... // Initialisiere hier alle Objekte, verbinde zum WLAN  
}  
  
void loop() {  
... // Lese wiederholt Sensoren aus und schicke Daten per WLAN  
}
```

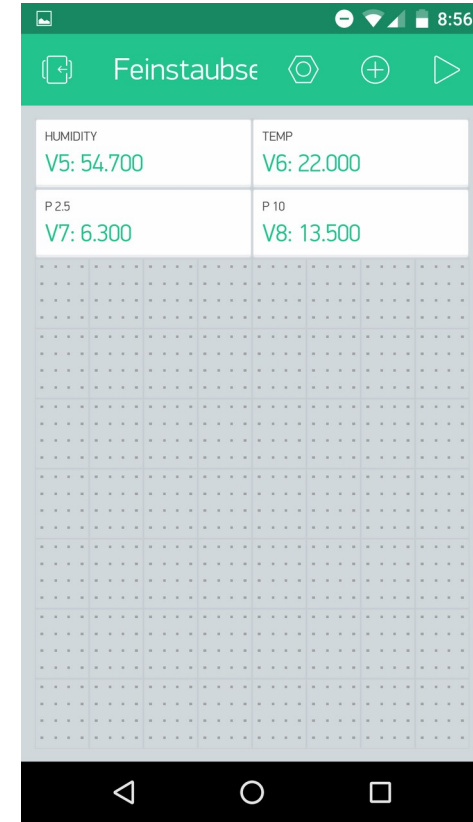

Set it up and loop through it

```
void setup() {  
    // Serial Monitor  
    Serial.begin(115200);  
    Blynk.begin(auth, ssid, pass,  
                IPAddress(192,168,178,36),8080);  
  
    timer.setInterval(1000L,sendSensor);  
  
    sdsSerial.begin(9600);  
    my_sds.begin(&sdsSerial);  
  
    dht.begin();  
}
```

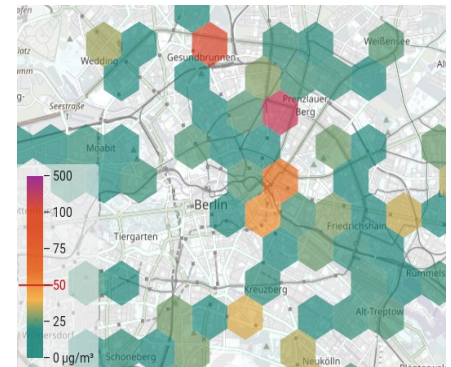
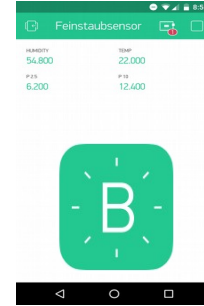
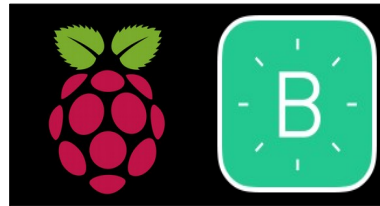
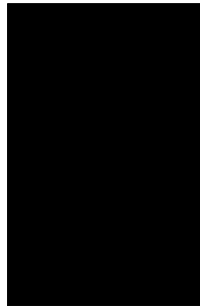
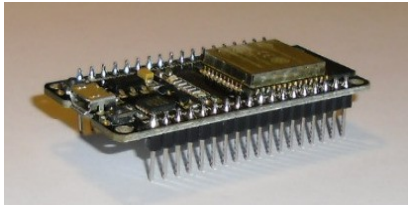
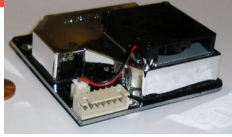
```
void loop() {  
    Blynk.run();  
    timer.run();  
}  
void sendSensor()  
{  
    readDHT();  
    readSDS();  
}
```

Blynk App

- Blynk Server Jar installieren
- IP-Adresse in Blynk-App setzen
- Chip und Pins in App definieren (ca. 5 min)



Die Zukunft



Quellen

<https://www.umweltbundesamt.de/themen/luft/luftschadstoffe/feinstaub>

https://www.umweltbundesamt.de/sites/default/files/medien/384/bilder/dateien/3_tab_emi-ausgew-luftschadst_2018_0.pdf

<https://earthobservatory.nasa.gov/Features/Aerosols>

<http://www.umweltbundesamt.at/pm10/>

http://www.euro.who.int/__data/assets/pdf_file/0005/78638/E90038.pdf

<https://deutschland.maps.luftdaten.info/>

ESP32, DHT22, SDS011 datasheets

DHT22

```
void readDHT()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature();

    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read
from DHT          sensor!");
        return;
    }
    Serial.println(t);

    Blynk.virtualWrite(V5, h);
    Blynk.virtualWrite(V6, t);
}
```

```
#include <DHT.h>
#define DHTPIN 14
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
void setup() {
    dht.begin();
}
```

SDS011

```
void readSDS()
{
    errorflag = my_sds.read(&p25, &p10);
    if (errorflag) {
        Serial.println("Failed to read
from SDS sensor!");
        return;
    }
    Serial.println("P2.5: " +
String(p25));
    Serial.println("P10:  " +
String(p10));
    Blynk.virtualWrite(V7, p25);
    Blynk.virtualWrite(V8, p10);
}
```

```
#include <SDS011.h>
```

```
float p10, p25;
```

```
int errorflag;
```

```
HardwareSerial sdsSerial(2); // use
Serial 2: RX pin:16, TX pin:17
```

```
SDS011 my_sds;
```

```
void setup() {
```

```
    sdsSerial.begin(9600);
```

```
    my_sds.begin(&sdsSerial);
```

```
}
```

Blynk

```
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// You should get Auth Token in the Blynk App.
char auth[] = "AAAAAAAAAAAAAA";

// Your WiFi credentials.
char ssid[] = "WIFI NAME";
char pass[] = "WIFI PWD";
BlynkTimer timer;

void sendSensor()
{
  readDHT();
  readSDS();
}
```

```
void setup()
{
  // Debug console
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass,
  IPAddress(192,168,178,36),8080);

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```