

Thingsboard ESP32 project

Ziel dieses Projekts ist es, Temperatur- und Feuchtigkeitsmesswerte, die von an einen ESP32 angeschlossenen Sensoren erfasst werden, an die Online-Plattform Thingsboard zu senden.

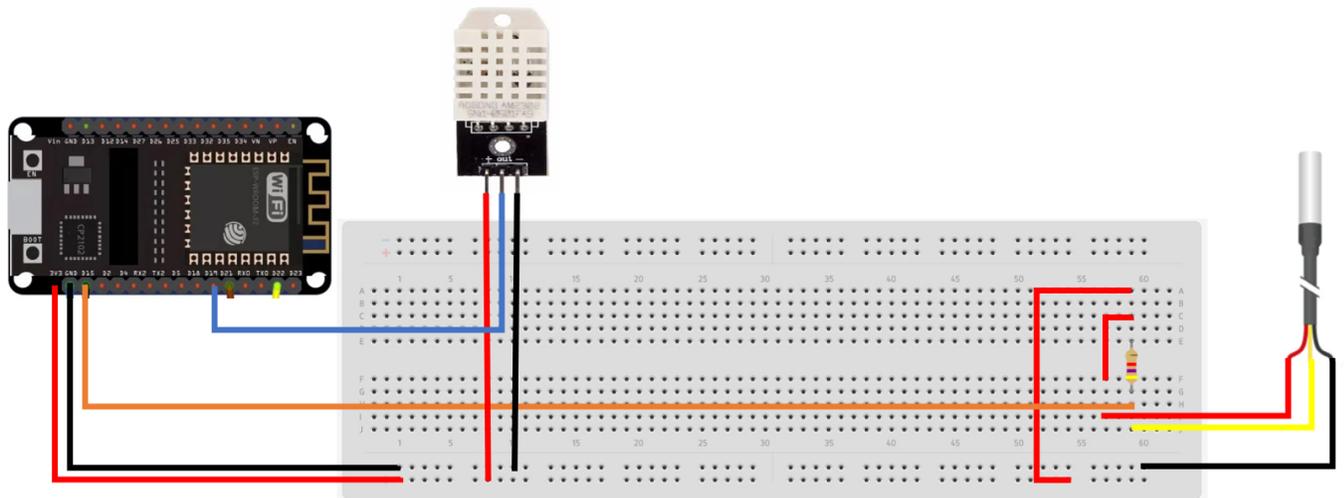
Anschließen der Sensoren an den ESP

Schaltkreis

Zur Herstellung der Schaltung werden folgende Komponenten benötigt:

- ❖ ESP32
- ❖ SEN-DHT22 Humidity Sensor
- ❖ DS18B20 Temperature Sensor
- ❖ 4.7 kΩ resistor
- ❖ Jumper wires
- ❖ Breadboard

Verbinden Sie die Komponenten wie in der Abbildung gezeigt:



Arduino Code

Erstellen Sie eine neue Arduino-Skizze und fügen Sie den folgenden Code ein:

```
#include <OneWire.h>
#include <Wire.h>
#include <config.h>
```

```
//-----Temperature sensor-----
#include <DallasTemperature.h>
```

```

const int SENSOR_PIN = 15; // Arduino pin connected to DS18B20 sensor's DQ pin
OneWire oneWire(SENSOR_PIN); // setup a oneWire instance
DallasTemperature tempSensor(&oneWire); //pass oneWire to DallasTemperature library
float tempCelsius; // temperature in Celsius
float temperature = 0;

//-----Humidity Sensor-----
#include <DHT.h>
#define DHTPIN 19
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE); //// Initialize DHT sensor for normal 16mhz Arduino
float humidity = 0;

void setup()
{
  Serial.begin(115200);

  dht.begin();
}

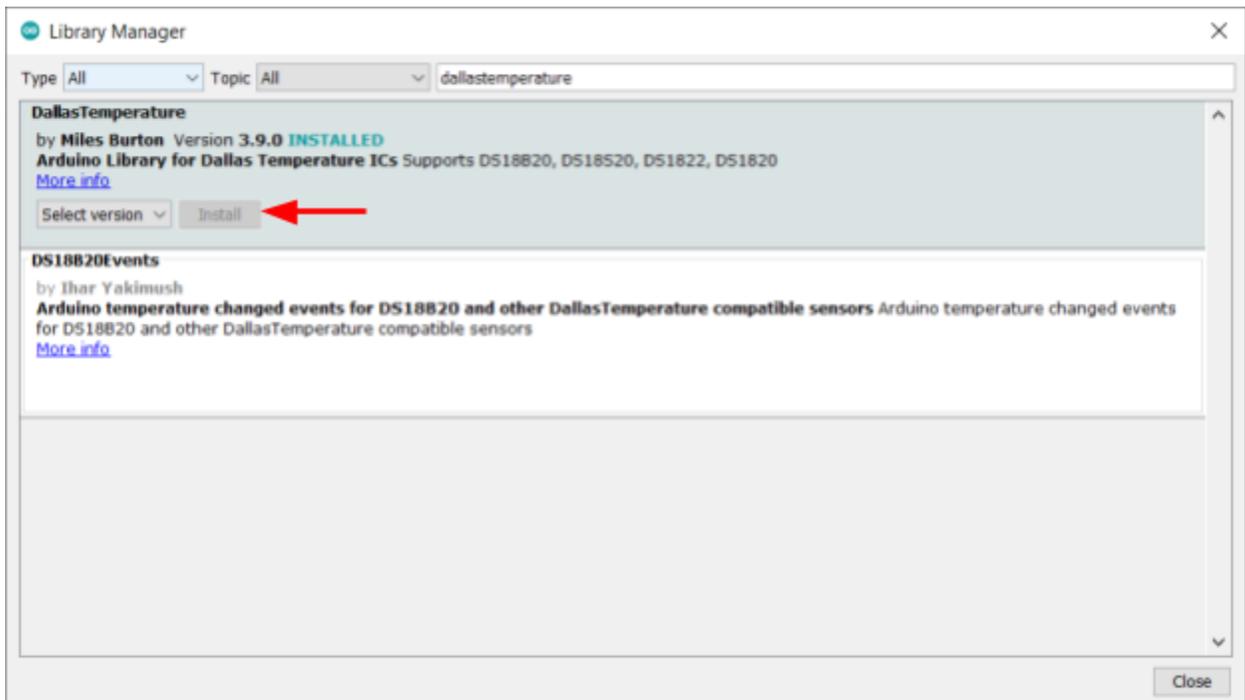
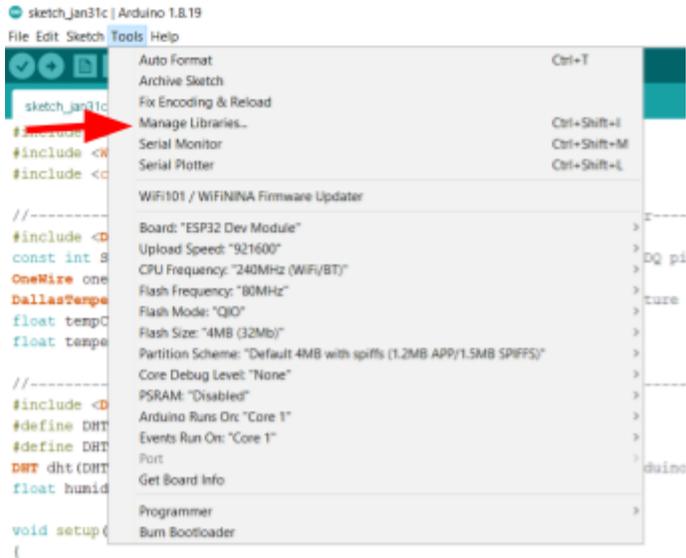
void loop()
{
  humidity = dht.readHumidity();
  temperature = readTemperature();
  Serial.print("Temperature(Celsius):");
  Serial.print(temperature); Serial.print(", Humidity:"); Serial.println(humidity);
  delay(1000);
}

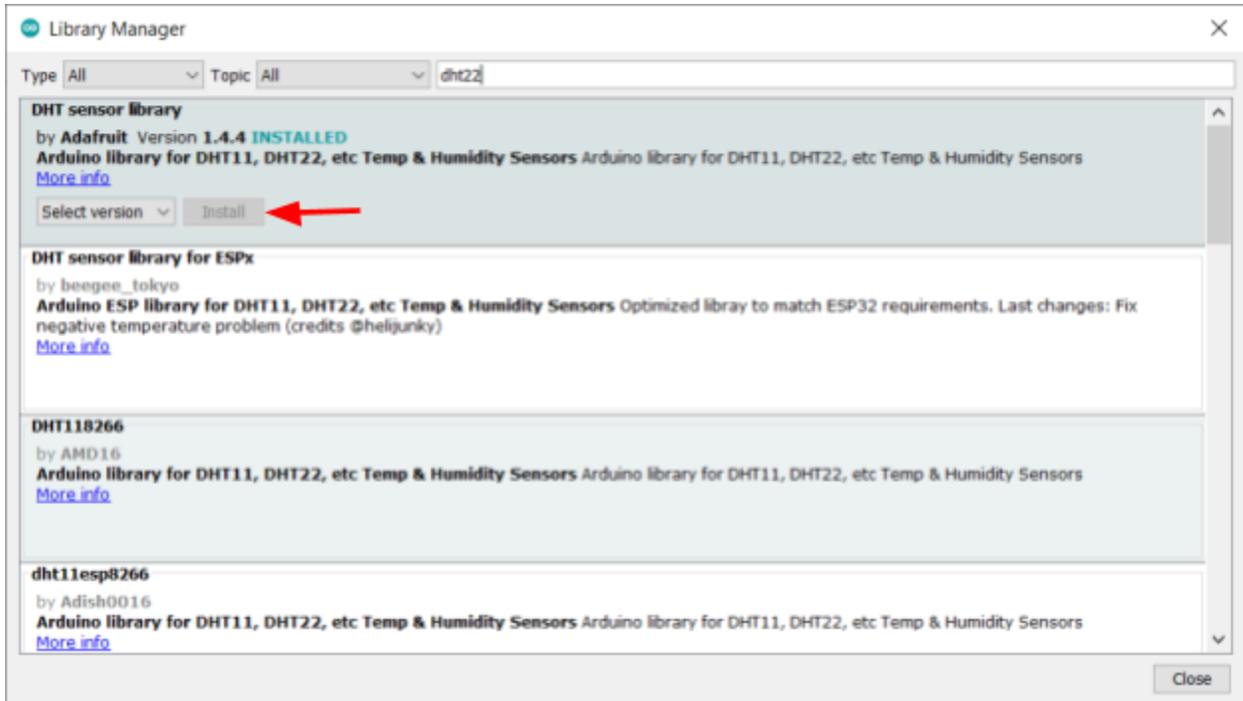
float readTemperature()
{
  //add your code here to get the temperature from your temperature sensor
  tempSensor.requestTemperatures(); // send the command to get temperatures
  tempCelsius = tempSensor.getTempCByIndex(0); // read temperature in Celsius

  return tempCelsius;
}

```

Fügen Sie die erforderlichen Bibliotheken hinzu, indem Sie auf Extras > Bibliotheken verwalten klicken, dann „Dallas Temperature“ in die Suchleiste eingeben und dann auf „Installieren“ klicken. Fügen Sie auf dieselbe Weise die DHT-Sensorbibliothek und alle anderen fehlenden Bibliotheken hinzu.

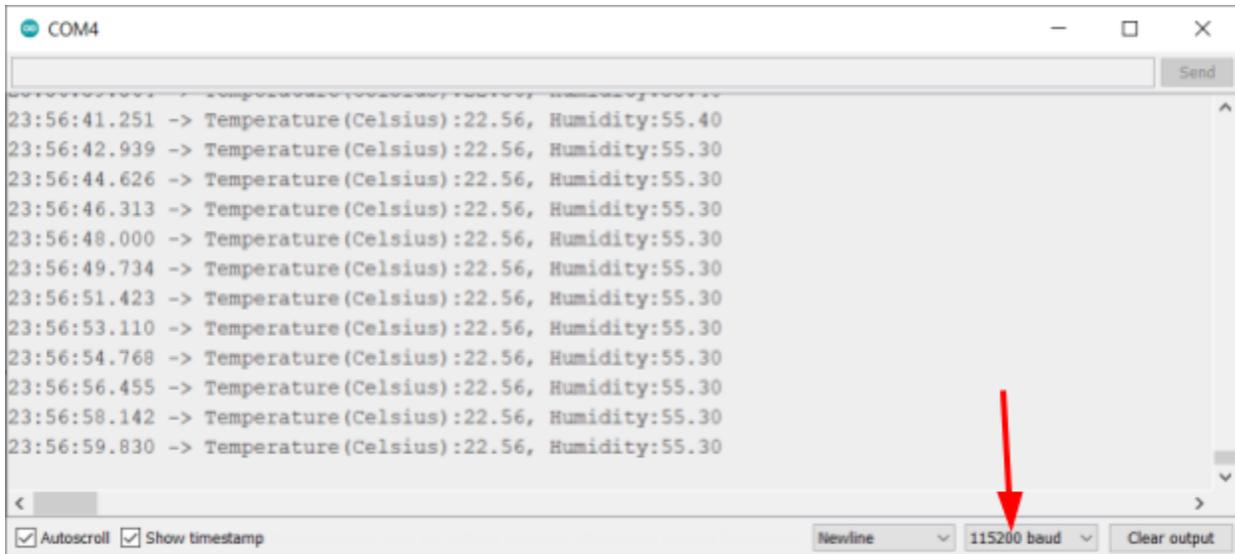




Nachdem Sie den Code hinzugefügt haben, schließen Sie den ESP32 an, laden Sie den Code hoch und öffnen Sie dann den seriellen Monitor.



Stellen Sie die Baudrate des seriellen Monitors auf 115200 ein und dann sollte der serielle Monitor die Sensorwerte ausdrucken.



ESP mit Thingsboard verbinden

Arduino Code

Erstellen Sie eine neue Arduino-Skizze und fügen Sie den folgenden Code ein:

```
#include <OneWire.h> #include <Wire.h> #include <config.h>

//-----Temperature Sensor-----
#include <DallasTemperature.h>
const int SENSOR_PIN = 15; // Arduino pin connected to DS18B20 sensor's DQ pin
OneWire oneWire(SENSOR_PIN); // setup a oneWire instance
DallasTemperature tempSensor(&oneWire); //pass oneWire to DallasTemperature library
float tempCelsius; // temperature in Celsius
float temperature=0;

//-----Humidity Sensor-----
#include <DHT.h>
#define DHTPIN 19
#define DHTTYPE DHT22 // DHT 22 (AM2302)
DHT dht(DHTPIN, DHTTYPE); // Initialize DHT sensor for normal 16mhz Arduino
float humidity=0;

//-----ThingsBoard-----
#include "ThingsBoard.h"
#include <Arduino.h>
#if defined(ESP32)
```

```

#include <WiFi.h>
#ifdef ESP8266 #include <ESP8266WiFi.h>
#endif
#define WIFI_AP "YOUR_WIFI_AP"
#define WIFI_PASSWORD "YOUR_WIFI_PASSWORD"
#define TOKEN "YOUR_ACCESS_TOKEN"
#define THINGSBOARD_SERVER "demo.thingsboard.io"
// Baud rate for debug serial
#define SERIAL_DEBUG_BAUD 115200
// Initialize ThingsBoard client
WiFiClient espClient;
// Initialize ThingsBoard instance
ThingsBoard tb(espClient);
// the Wifi radio's status
int status = WL_IDLE_STATUS;

void setup() {
  Serial.begin(115200);

  dht.begin();

  //ThingsBoard
  WiFi.begin(WIFI_AP, WIFI_PASSWORD); InitWiFi();
}

void loop() {
  thingsboard_setup(); send_data(); delay(2000);
}

void thingsboard_setup() {
  if (WiFi.status() != WL_CONNECTED) {
    reconnect();
  }
  if (!tb.connected()) {
    // Connect to the ThingsBoard
    Serial.print("Connecting to: ");
    Serial.print(THINGSBOARD_SERVER);
    Serial.print(" with token ");
    Serial.println(TOKEN);
    if (!tb.connect(THINGSBOARD_SERVER, TOKEN)) {

```

```

        Serial.println("Failed to connect"); return;
    } }
}

void InitWiFi() {
    Serial.println("Connecting to AP ..."); // attempt to connect to WiFi network

    WiFi.begin(WIFI_AP, WIFI_PASSWORD);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500); Serial.print(".");
    }
    Serial.println("Connected to AP"); }

void reconnect() {
    // Loop until we're reconnected status = WiFi.status();
    if ( status != WL_CONNECTED) {
        WiFi.begin(WIFI_AP, WIFI_PASSWORD);
        while (WiFi.status() != WL_CONNECTED) {
            delay(500); Serial.print(".");
        }
        Serial.println("Connected to AP"); }
}

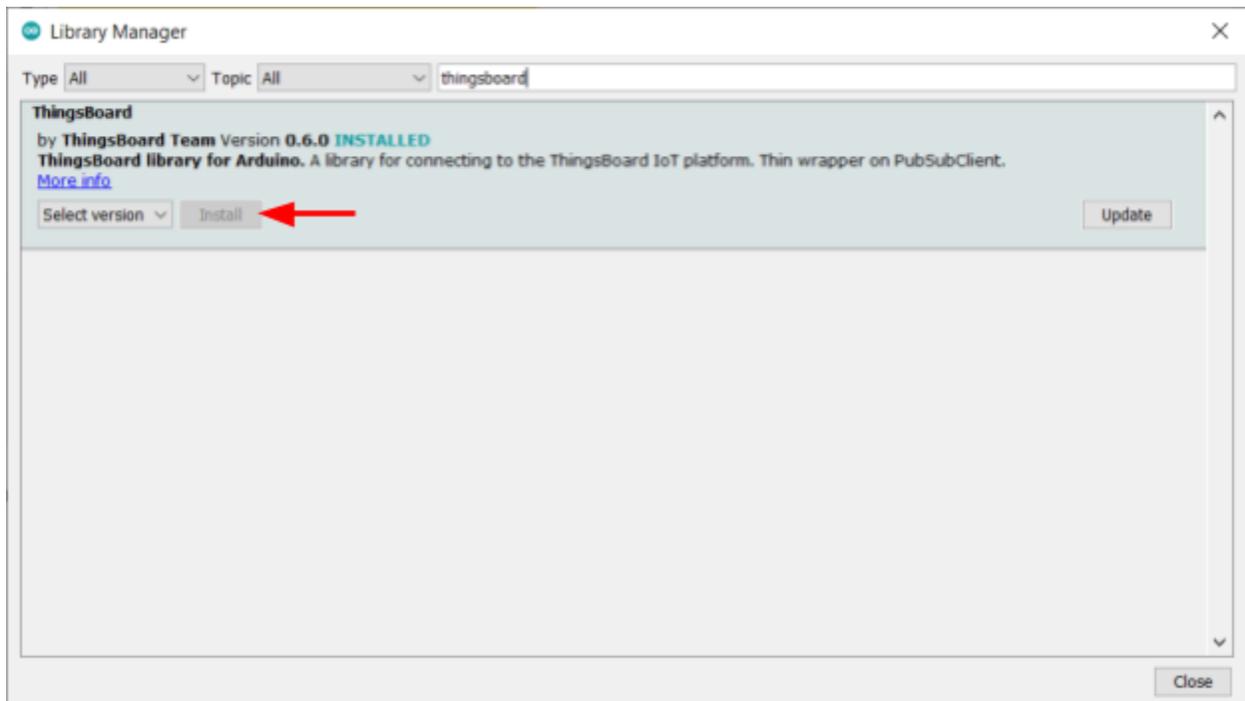
void send_data() {
    //temperature
    temperature = readTemperature(); // read your temperature sensor to execute
    temperature compensation
    humidity=dht.readHumidity();
    Serial.print("Temperature:");
    Serial.print(temperature);
    Serial.print(", Humidity:"); Serial.println(humidity);

    tb.sendTelemetryFloat("Temperature", temperature);
    tb.sendTelemetryFloat("Humidity", humidity); tb.loop();
}

```

```
float readTemperature()  
{  
  //add your code here to get the temperature from your temperature sensor  
  tempSensor.requestTemperatures();           // send the command to get  
  temperatures  
  tempCelsius = tempSensor.getTempCByIndex(0); // read temperature in Celsius  
  
  return tempCelsius;  
}
```

Fügen Sie die Thingsboard-Bibliothek hinzu, indem Sie auf Extras > Bibliotheken verwalten klicken, dann „thingsboard“ in die Suchleiste eingeben und die Bibliothek installieren. Fügen Sie auf dieselbe Weise alle anderen fehlenden Bibliotheken hinzu.



Ersetzen Sie die Werte für WIFI_AP und WIFI_PASSWORD durch den Namen und das Passwort des WLAN-Netzwerks. Dadurch kann der ESP32 eine Verbindung zum Internet herstellen.

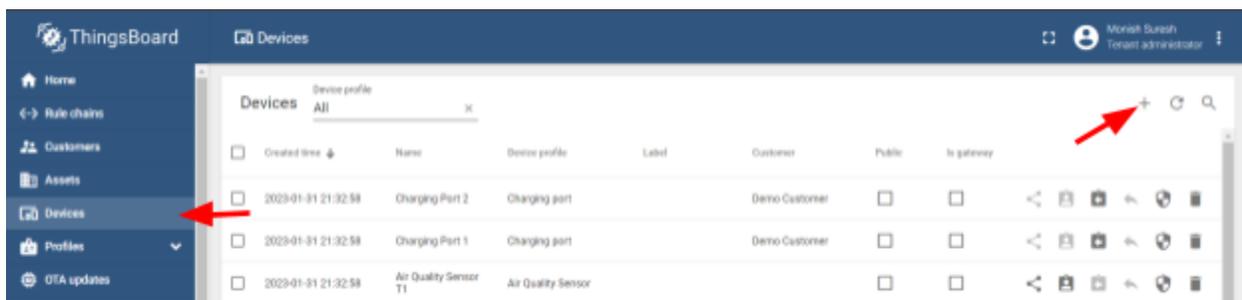
```

//-----ThingsBoard-----
#include "ThingsBoard.h"
#include <Arduino.h>
#include defined(ESP32)
#include <WiFi.h>
#include defined(ESP8266)
#include <ESP8266WiFi.h>
#include defined
#define WIFI_AP "DESKTOP-SB2AKHS 4577"
#define WIFI_PASSWORD "password"

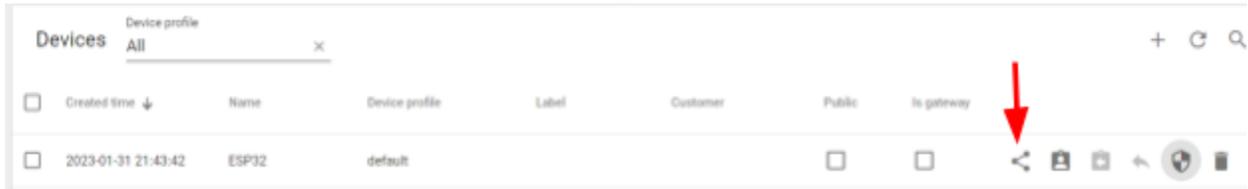
```

Thingsboard einrichten

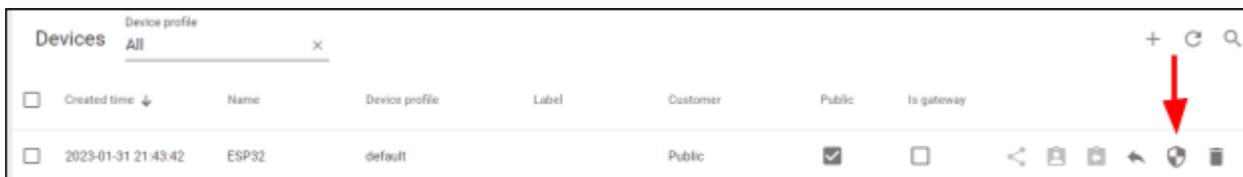
Besuchen Sie <https://demo.thingsboard.io/login> und erstellen Sie ein Konto. Klicken Sie nach der Anmeldung auf Geräte und wählen Sie dann Gerät hinzufügen. Name the device ESP32 and click add.



Machen Sie das Gerät öffentlich, indem Sie auf „Gerät öffentlich machen“ klicken.



Der ESP32 benötigt ein „Zugriffstoken“, um mit Thingsboard zu arbeiten. Um das Token zu erhalten, klicken Sie auf „Anmeldeinformationen verwalten“ und kopieren Sie das Zugriffstoken. Fügen Sie dieses Zugriffstoken als Wert für TOKEN in den Arduino-Code ein.



Device Credentials

Credentials type
Access token

Access token *

uuXQPCDXSbt4AUfQXA8F

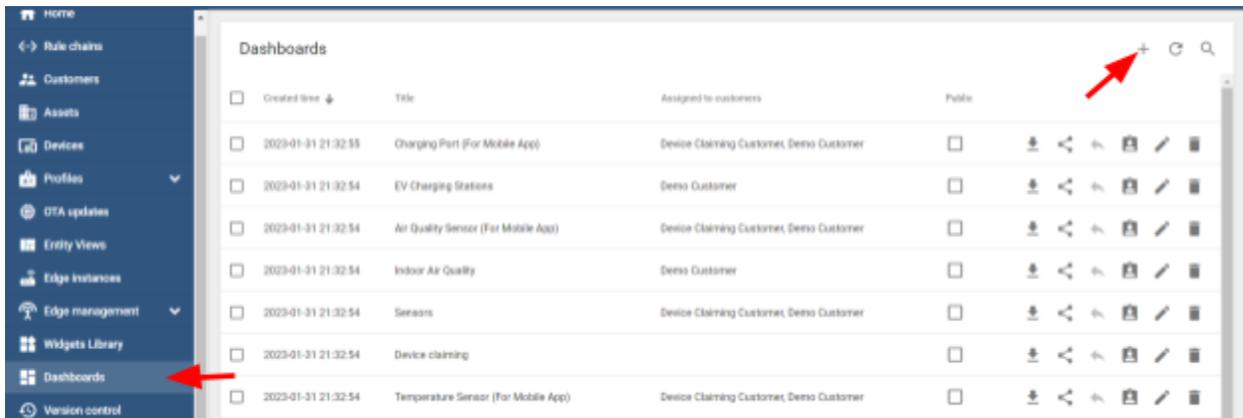
Cancel Save

```
//-----ThingsBoard-----  
#include "ThingsBoard.h"  
#include <Arduino.h>  
#if defined(ESP32)  
#include <WiFi.h>  
#elif defined(ESP8266)  
#include <ESP8266WiFi.h>  
#endif  
#define WIFI_AP "DESKTOP-SB2AKHS 4577"  
#define WIFI_PASSWORD "password"  
#define TOKEN "aDdhM0iEeJrN9o2X2JQc" ←  
#define THINGSBOARD_SERVER "demo.thingsboard.io"  
// Baud rate for debug serial
```

Schließen Sie das ESP an und laden Sie den Code hoch. Das ESP überträgt nun jede Sekunde die Sensorwerte an Thingsboard.

Erstellen des Dashboards

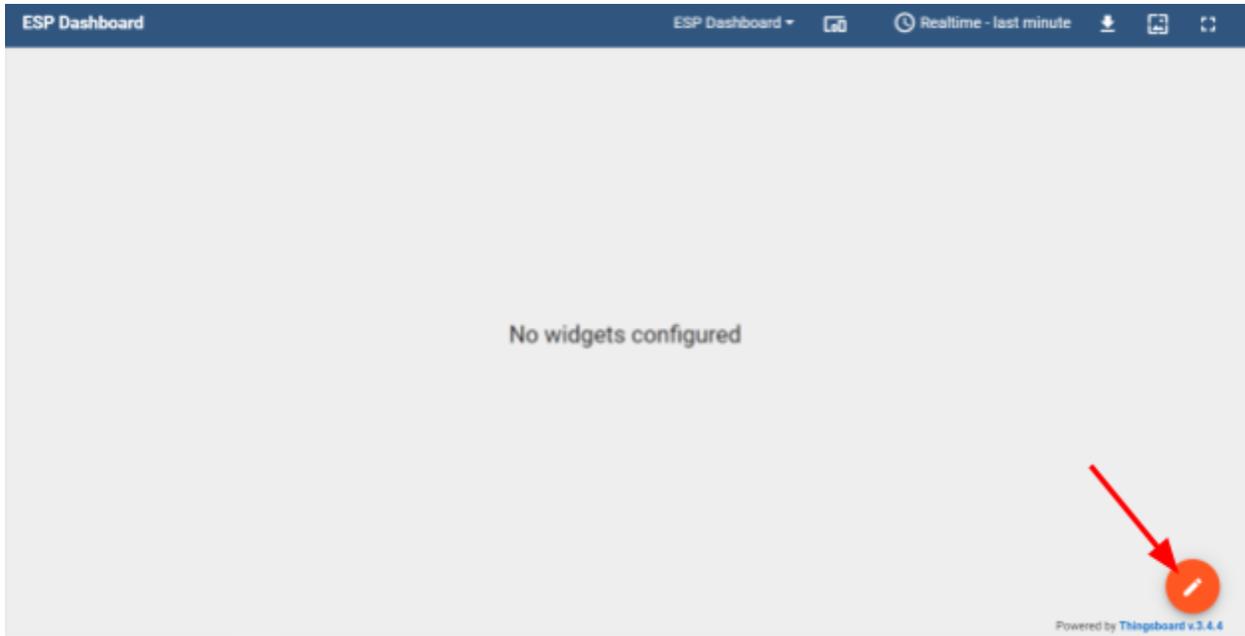
Im Dashboard kann Thingsboard die vom ESP empfangenen Messwerte anzeigen. Um ein neues Dashboard zu erstellen, klicken Sie auf Dashboards, dann auf das Plus-Symbol und wählen Sie „Neues Dashboard erstellen“.



Legen Sie den Titel auf „ESP Dashboard“ fest und klicken Sie auf „Hinzufügen“. Machen Sie das Dashboard öffentlich, indem Sie auf das Symbol klicken und dann auf ESP-Dashboard klicken, um es zu öffnen.

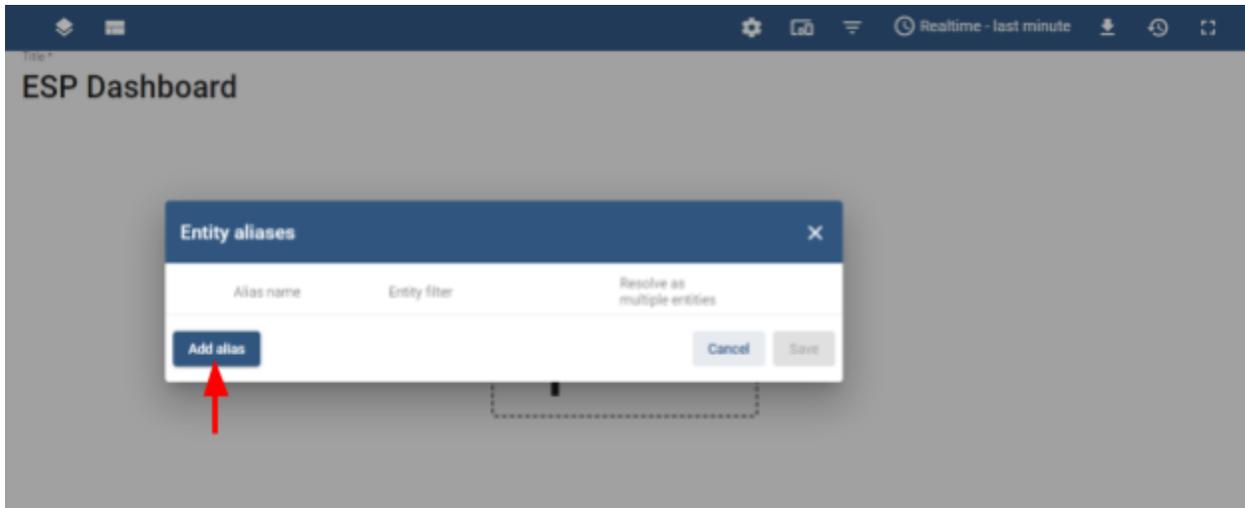


Das Dashboard muss mit dem Gerät verknüpft sein, das wir im vorherigen Abschnitt hinzugefügt haben. Wechseln Sie dazu in den Bearbeitungsmodus, indem Sie auf das Symbol unten rechts klicken.

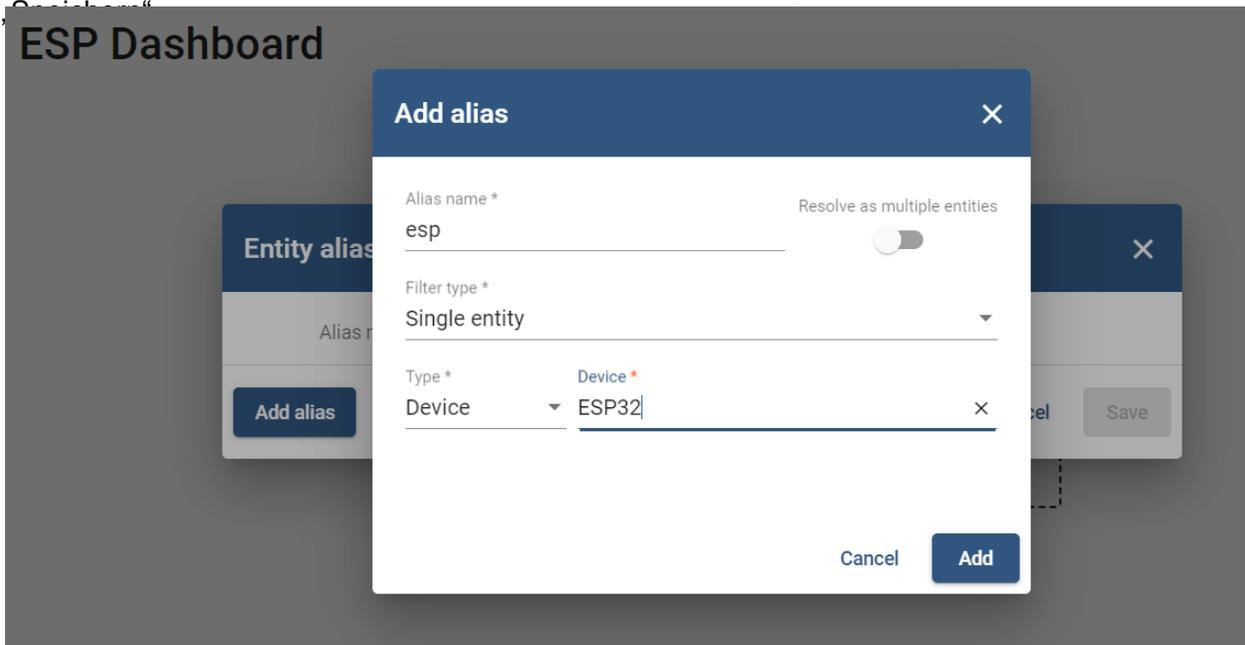


Klicken Sie dann auf „Entitätsalias“ und befolgen Sie die Schritte in den folgenden Bildern.

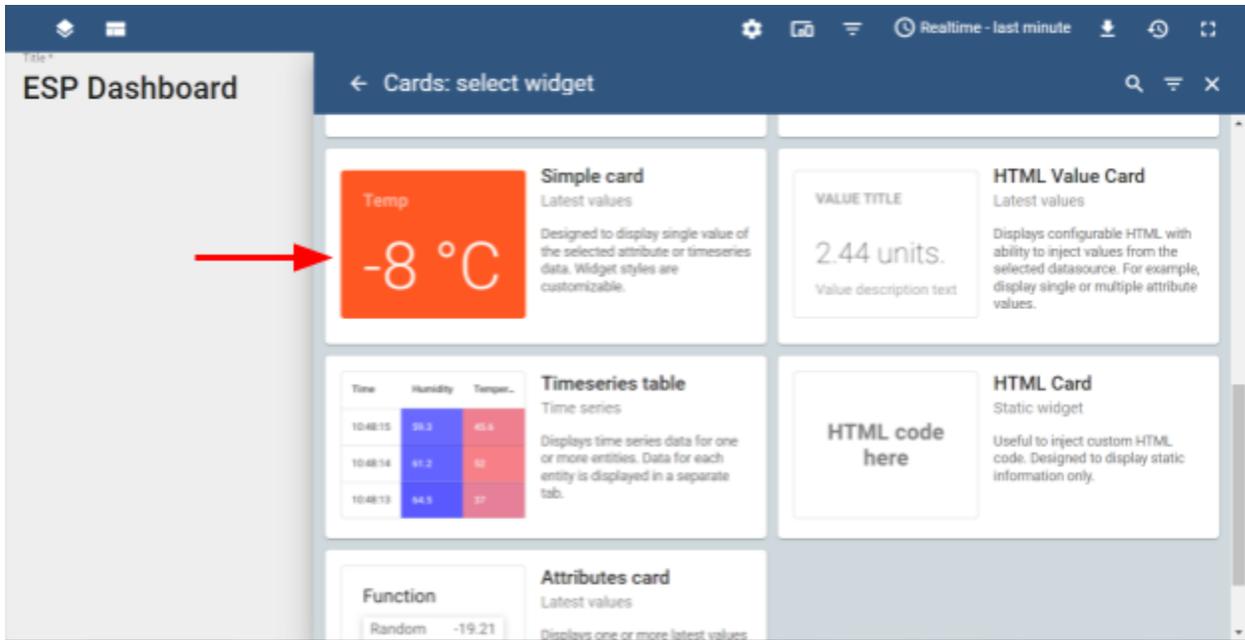




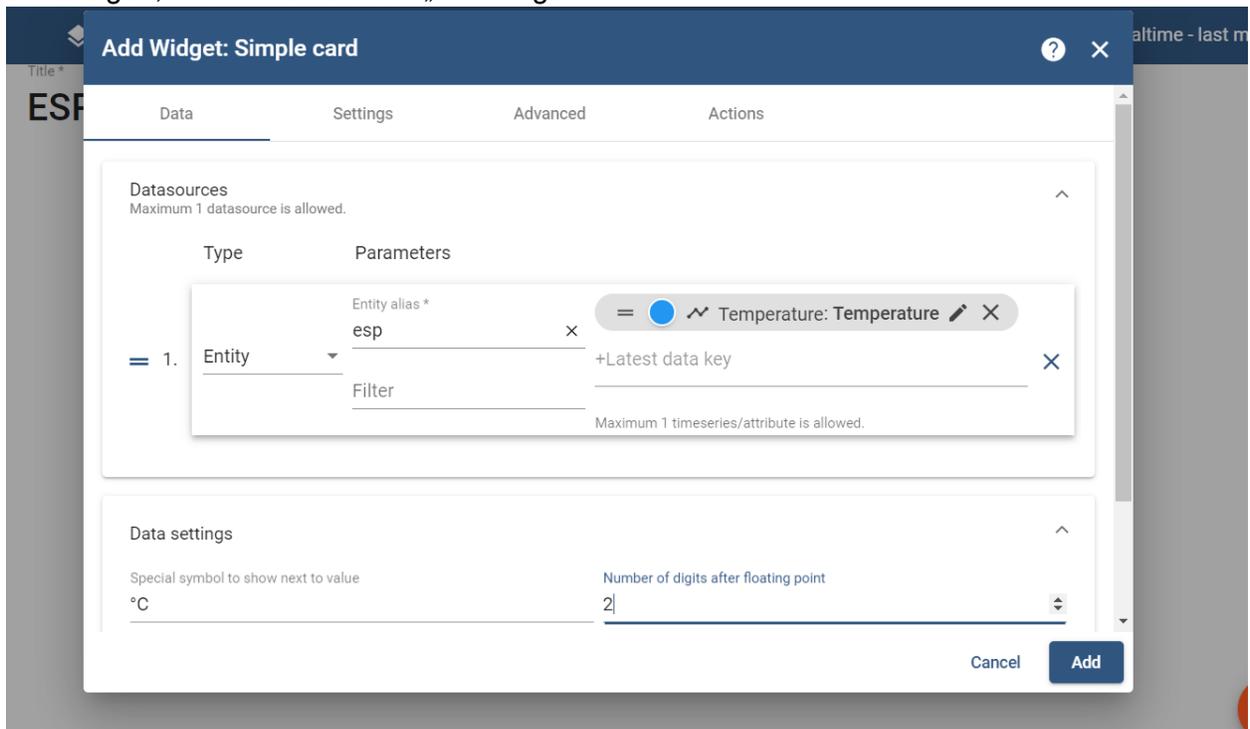
Geben Sie die folgenden Details ein, klicken Sie auf „Hinzufügen“ und dann auf „Speichern“



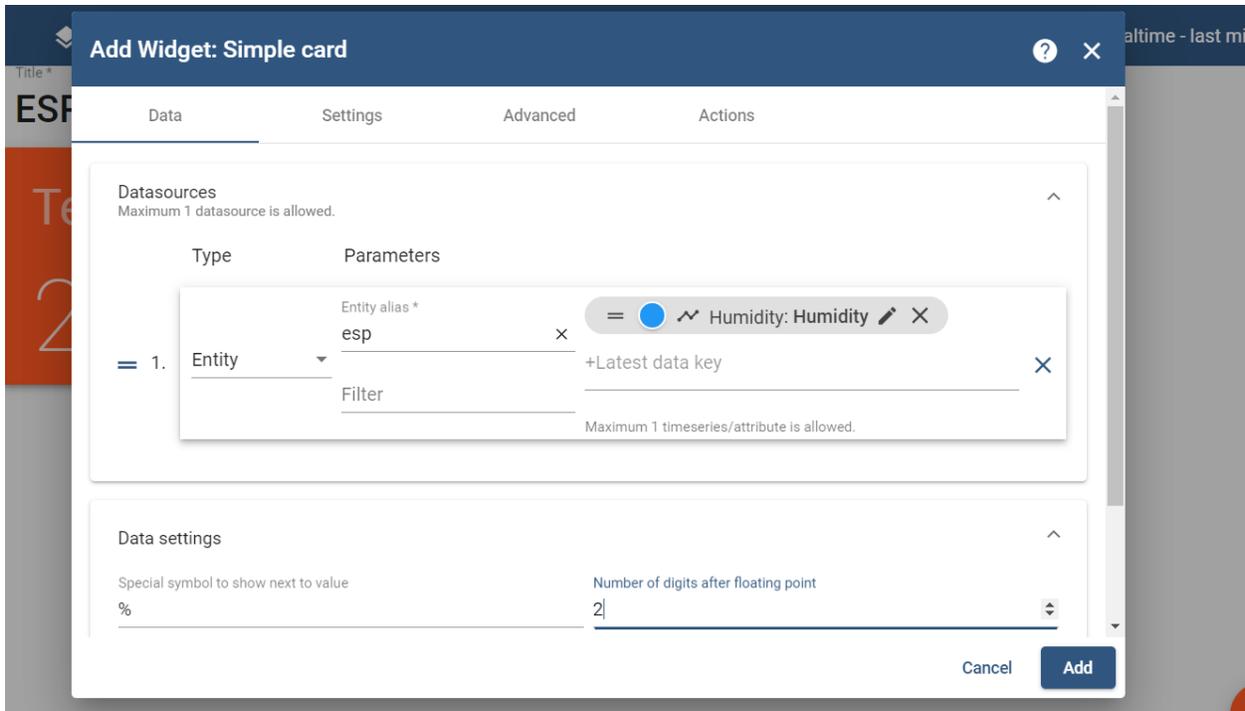
Um empfangene Daten anzuzeigen, müssen wir Widgets verwenden. Klicken Sie dazu auf Neues Widget hinzufügen > Karten > Einfache Karte.



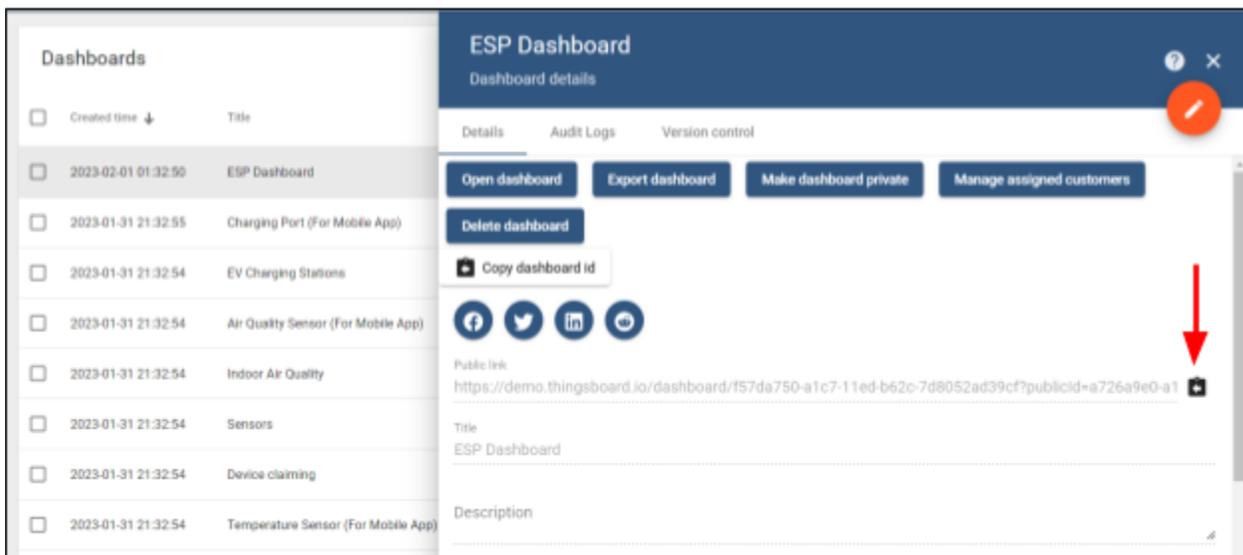
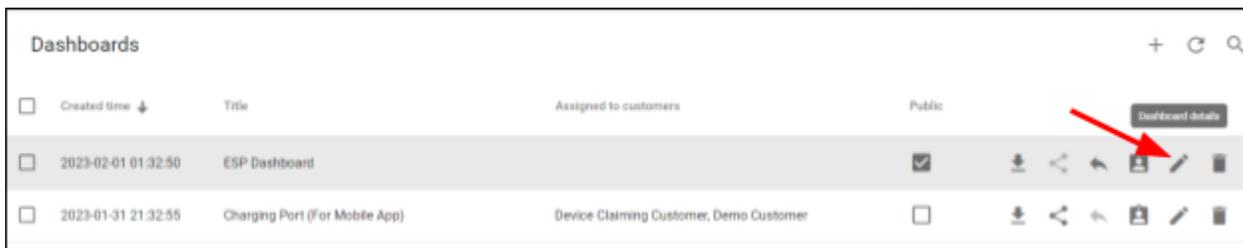
Geben Sie die folgenden Details ein, um die Messwerte des Temperatursensors anzuzeigen, und klicken Sie auf „Hinzufügen“.



Befolgen Sie in ähnlicher Weise die vorherigen Schritte, um eine Karte für den Luftfeuchtigkeitswert hinzuzufügen.



Jetzt sollten die Karten dieselben Sensorwerte anzeigen, die auf dem seriellen Monitor von Arduino gedruckt werden. Ein öffentlicher Link kann über das Dashboard-Menü abgerufen werden, indem Sie auf Dashboard-Details klicken. Über diesen Link können Sie von jedem mit dem Internet verbundenen Computer aus auf die Live-Sensorwerte zugreifen.





Temperature

22.56 °C

Humidity

54.4 %