REMO OBERHOLZER UND STEFAN SCHMUTZ
3-12-2019
4.00

ANLEITUNG ZUR APP-PROGRAMMIERUNG

IN EINEM ERSTEN SCHRITT WIRD DIE TEMPERATUR UND LUFTFEUCHTIGKEIT MIT EINEM ESP32 AUFGEZEICHNET UND AN EINE APP GESENDET

Image Credit *ZHAW, IUNR*

ANLEITUNG ZUR APP-PROGRAMMIERUNG

IN EINEM ERSTEN SCHRITT WIRD DIE TEMPERATUR UND LUFTFEUCHTIGKEIT MIT EINEM ESP32 AUFGEZEICHNET UND AN EINE APP GESENDET

AUFGABE 1

Ziel:

Mit einem ESP32 soll mit Hilfe des externen Temperatur- und Luftfeuchtigkeitssensors DHT11 Daten aufgezeichnet werden und dann über eine WEB Datenbank an ein mobile Phone übertragen werden. Das ESP32 ist vergleichbar mit einem Arduino Mega, ausser, dass es bereits eine Bluetooth- und Wifi-Schnittstelle eingebaut hat. Das ESP32 kann mit derselben Software wie das Arduino programmiert werden.

Sowohl die Software, wie auch die APP für das mobile Phone werden wir selber schreiben.

Vorgehen:

SCHRITT 1

In den nächsten Abschnitten geht es darum, eine App zu schreiben, welche auf eine Onlinedatenbank zugreifen kann und Daten liest sowie einfügt (read/write).

Zusätzlich soll ein ESP32 Microcontroller Daten von einer Datenbank auslesen und eigenständig verarbeiten.

Verwendet wird:
x.thunkable
google.firebase
ESP 32 auf der Arduino IDE

LERNZIELE

Die erworbenen Programmierkenntnisse mit Arduino für die Programmierung von ESP32 zu nutzen. In dieser Anleitung sollen Temperatur- und Feuchtigkeitswerte von einem Sensor zur Verfügung gestellt werden und vom ESP32 in eine Datenbank geschrieben werden. Zudem wird gezeigt wie eine APP auf diese Daten zugreift.

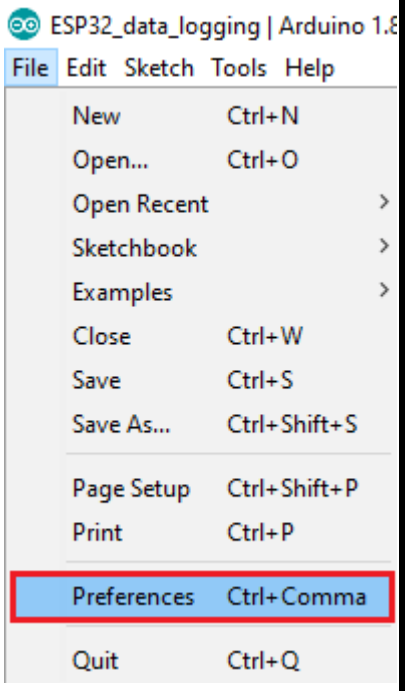
Bearbeitungszeit etwa:
4 Stunden

SCHRITT 2 ESP 32 EINRICHTUNG

Ziel:

Mit der Arduino IDE kann kann auch das ESP32 programmiert werden. Dazu muss diese aber vom Modus «Arduino mega» zu «ESP32» umgestellt werden.

Vorgehen:



1. Arduino IDE und öffnen

2. Unter Werkzeuge auf Bibliothek verwalten klicken

3. In Deiner Arduino IDE, gehe zu **“File” -> “Preferences”** (vgl. rechts)

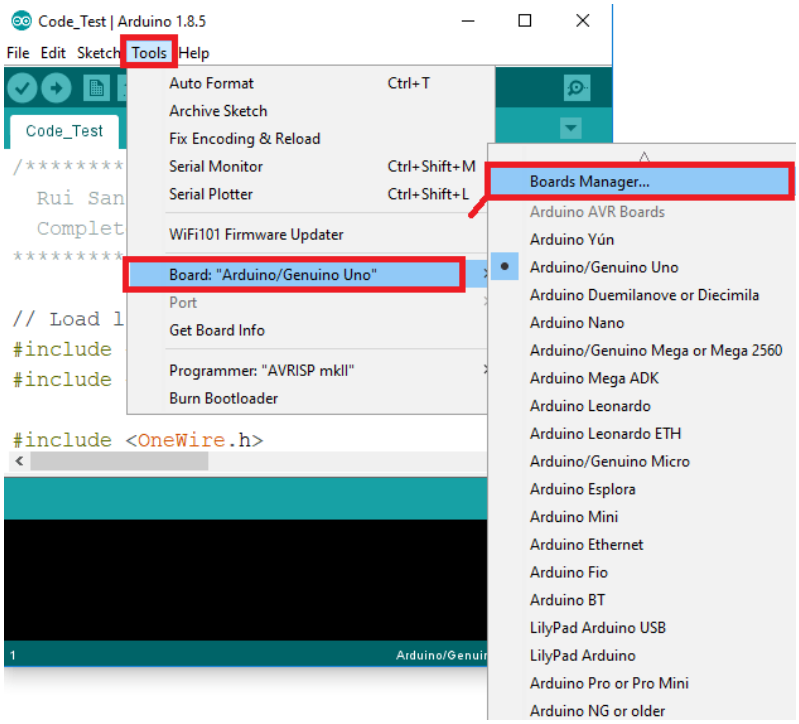
4. Gib

https://dl.espressif.com/dl/package_esp32_index.json

in den “Additional Board Manager URLs” ein und drücke “OK”

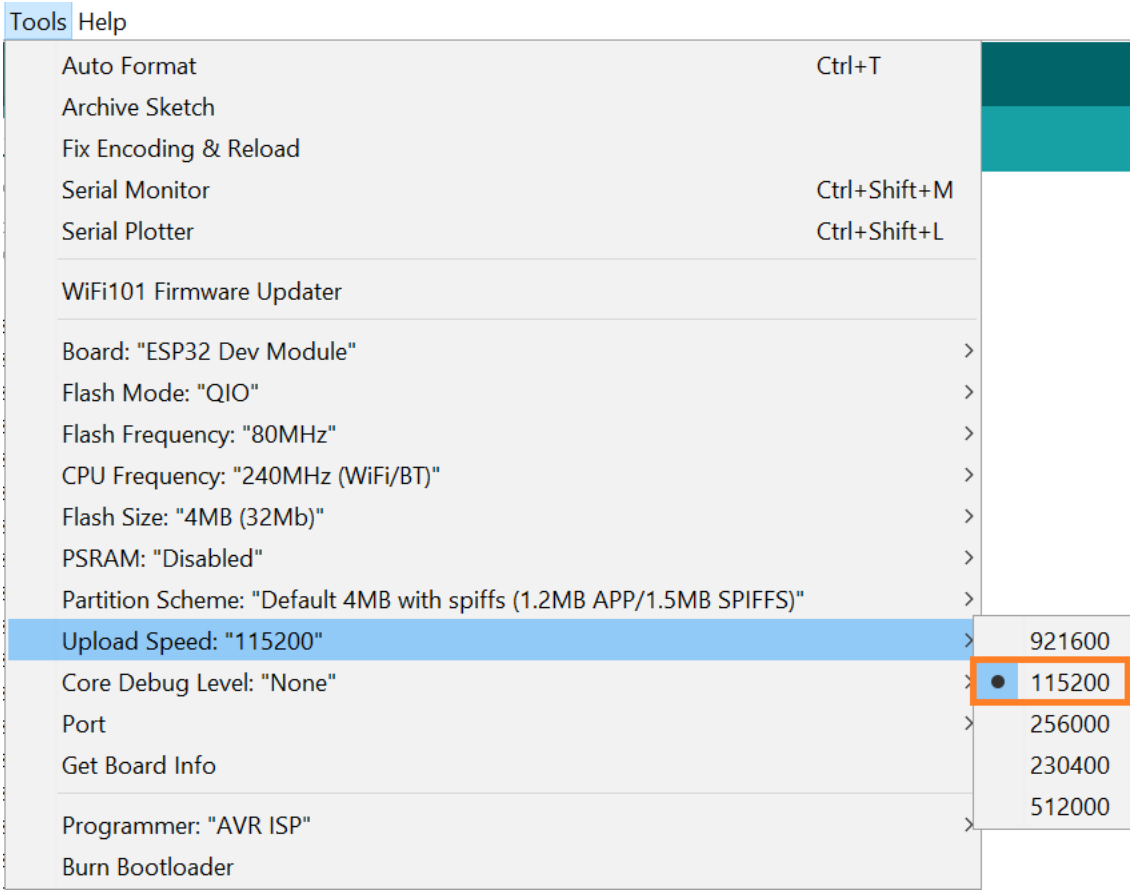
```
Falls Du bereits ESP8266-Boards verwendest, müsstest
Du diesen Link verwenden:

https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_
index.json
```

5. Öffne den Board Manager: Gehe zu **Tool > Board > Board Manager** …(vgl. Abbildung unten)



6. Suche nach ESP32 and drücke den Knopf «Install» für “ESP32 by Espressif Systems ESP 32»

7. Suche nochmals nach ESP32 und Du solltest folgendes Bild erhalten:



8. Wähle nun unter Werkzeuge das Board “ESP32 Dev Module” aus und setze den Uploadspeed auf 115200 (ja, das ESP ist zackig unterwegs 😉).

9. Fertig! Nun sollte alles geklappt haben und das ESP32 ist bereit für Arduino IDE.

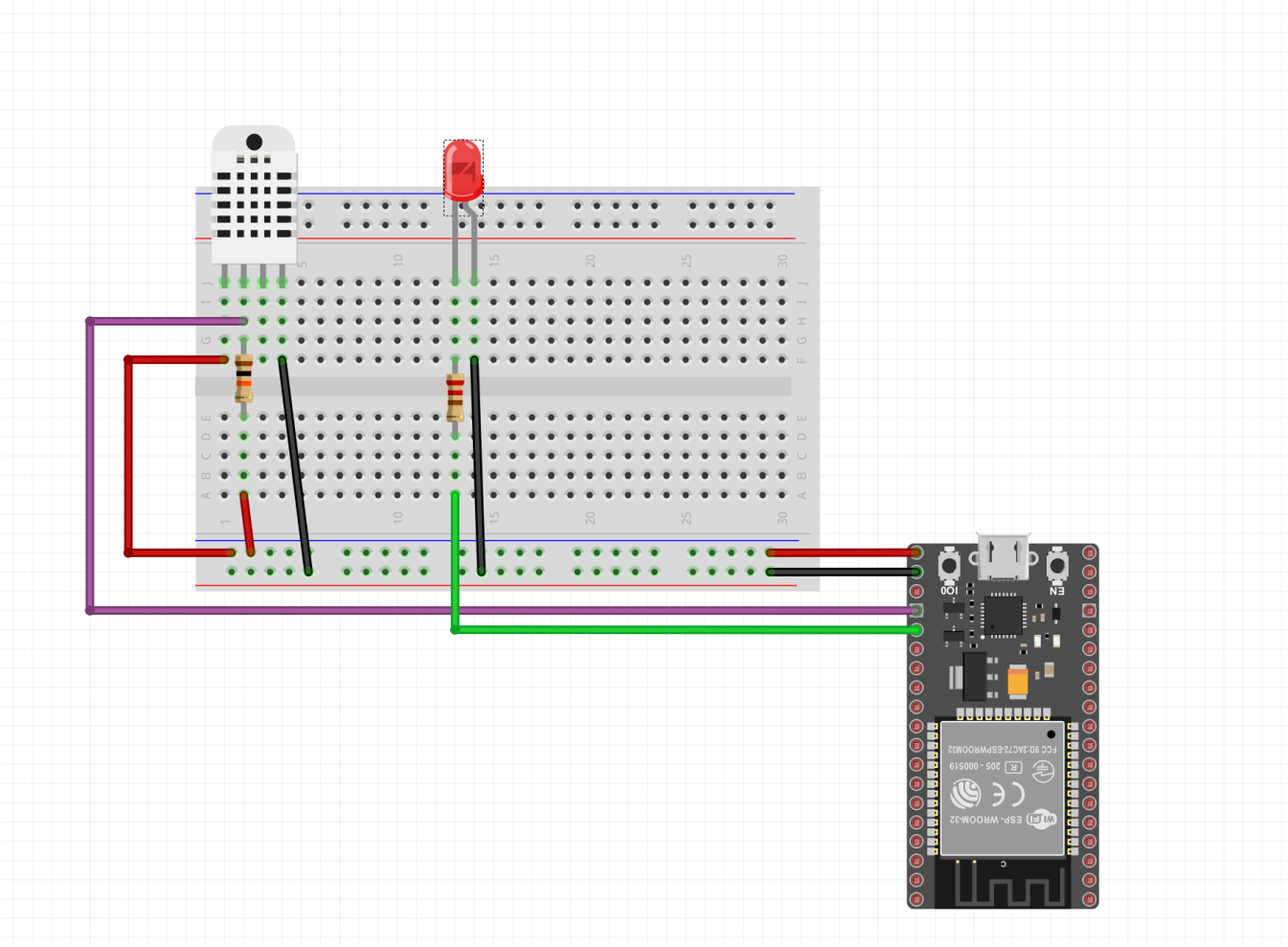
SCHRITT 3 VERBINDEN DES SONSORS MIT DEM ESP

Ziel:

Um den nachfolgenden Sketch laufen lassen zu können, müssen wir den Temperaturfühler an das ESP32 anschliessen.

Vorgehen:

Verdrahte den Sensor mit dem ESP 32 wie abgebildet:



SCHRITT 4 ZU INSTALLIERENDE PAKETE AUF ARDUINO IDE

Ziel:

Um den nachfolgenden Sketch laufen lassen zu können, müssen einige zusätzliche Libraries installiert werden.

Vorgehen:

Die folgenden Libraries sind notwendig, welche über den Library Manager installiert werden können.

1. Firebase esp32 client
2. DHT sensor library
3. Adafruit Unified Sensor

Hier stellen wir Euch detaillierte Informationen in Unterricht zur Verfügung.

SCHRITT 5 ARDUINO DER SKETCH

Hier ist der Code, welcher auf dem ESP32 programmiert werden muss:

```
#include <WiFi.h>
#include <FirebaseESP32.h>

#define FIREBASE_HOST "ledtest-47835.firebaseio.com" // Deine Datenbankadresse ohne https:// hier einfügen
#define FIREBASE_AUTH "kLVTwn0wclcUxmVYQ27dXHN9QA7SwZoRwPKX75Z4" //Firebase Secret Key hier einfügen.
#define WIFI_SSID "pink fluffy unicorn"                          // Netzwerkname hier einfügen
#define WIFI_PASSWORD "12345678"                    // Netwerkpasswort hier einfügen

FirebaseData firebaseData;                          // Kreiert ein Objekt



#include <DHT.h>                                    // Nachfolgend Temperatur/Feuchtigkeits Sensor Einstellungen

#define DHTPIN 2
#define DHTTYPE    DHT11

DHT dht(DHTPIN, DHTTYPE);


int ledPin = 4;                                     // LED PINOUT am ESP32 Bord
int warten = 5000;                                  // Warteschleife für Serial.print, damit Serieller Monitor nicht überfüllt
```

```
void setup(){
  Serial.begin(115200);                                 // Prozessorgeschwindigkeit ESP32
  dht.begin();

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);                 // Verbindungsaufbau mit Wlan
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED){
    Serial.print(".");                                  // Output bis verbunden
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");                  // Output wenn verbunden
  Serial.println(WiFi.localIP());
  Serial.println();



  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);


  Firebase.setReadTimeout(firebaseData, 1000 * 60);  //Database read timeout to 1 minute (max 15 minutes)
  Firebase.setwriteSizeLimit(firebaseData, "tiny");  //schreibtimeout:tiny1s,small10s,medium30s,large60s, unlimited


  pinMode(ledPin, OUTPUT);                              // Zahlangabe als Pinout wiedergeben
  }
```

```
void loop(){
  float t = dht.readTemperature();                  // Temperatur an Sensor auslesen
  float h = dht.readHumidity();                     // Feuchtigkeit am Sensor auslesen

  Firebase.setFloat(firebaseData,"/TEM",t);         // Eingabe Wert von /TEM. in Firebase
  Firebase.setFloat(firebaseData,"/HUM",h);         // Eingabe Wert von /HUM. in Firebase
  Firebase.getInt(firebaseData, "/LED");            // Abfrage Wert /LED in Firebase
                                                    // Wenn Datenbankast Tiefer als ein Wert ist, sieht Abfrage so aus;
                                                    // Firebase.setFloat(firebaseData,"/Haus/Erdgeschoss/Küchenlampe",anauswert);

  if( millis() > warten ){
    Serial.println(t);
    warten = warten + warten;
  }

  int LED = firebaseData.intData();                 // Abspeichern von Objektwert

  if(LED == 1){                                     // Einfaches An/Aus. Hier könnte man zb. Relais steuern,PWM, etc...
    Serial.println("LED AN");
    digitalWrite(ledPin, HIGH);
  }
  if(LED == 0){
    digitalWrite(ledPin, LOW);
  }}
```

SCHRITT 6 FIREBASE INSTALLIEREN

Ziel:

Installiere bitte die Firebase-Datenbank mit folgenden Schritten:

Vorgehen:

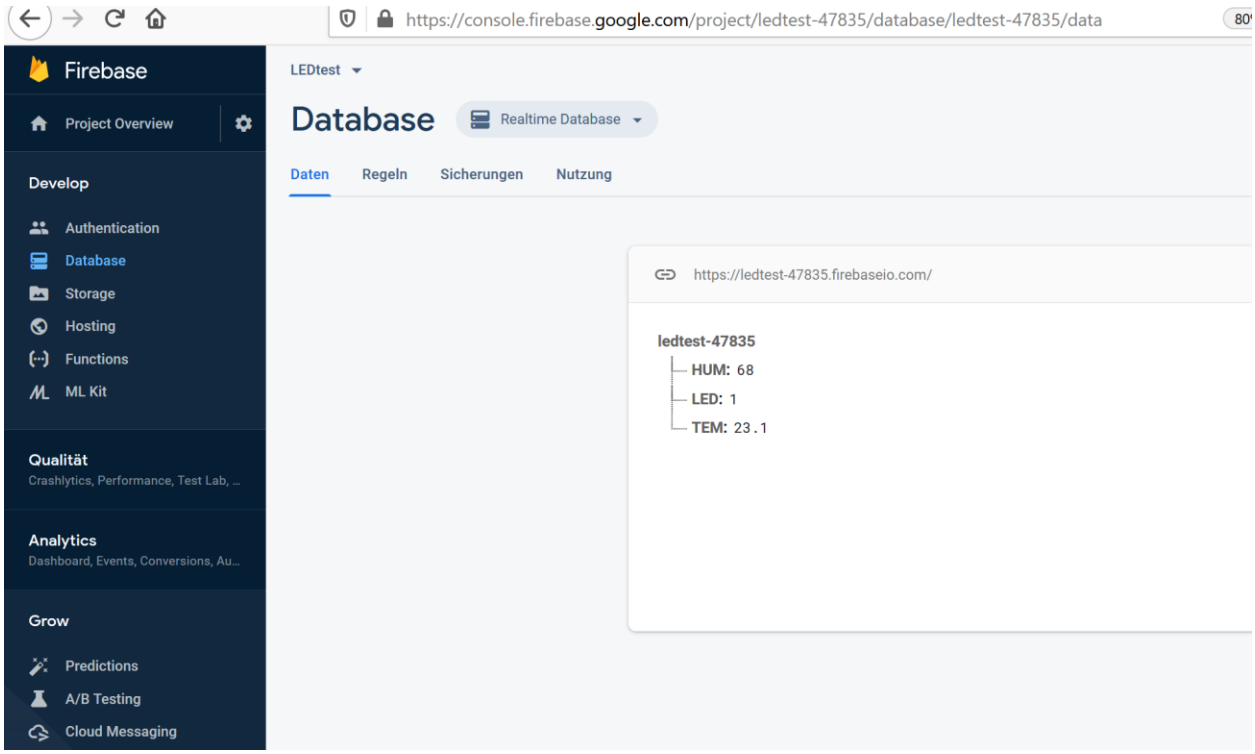
1. Kostenlos Registrieren auf https://firebase.google.com
2. Klicke auf «zur Konsole» oben rechts.
3. Projekt hinzufügen klicken
4. Projektname eingeben (egal was)
5. Google analytics deaktivieren, Projekt erstellen klicken
6. Klicke auf «Datenbank erstellen» weiter unten in Rubrik Realtime Database
7. im gesperrten Modus starten
8. auf «Regeln» klicken, abändern.

```
{
  "rules": {
    ".read": true,
    ".write": true
  }
}
```

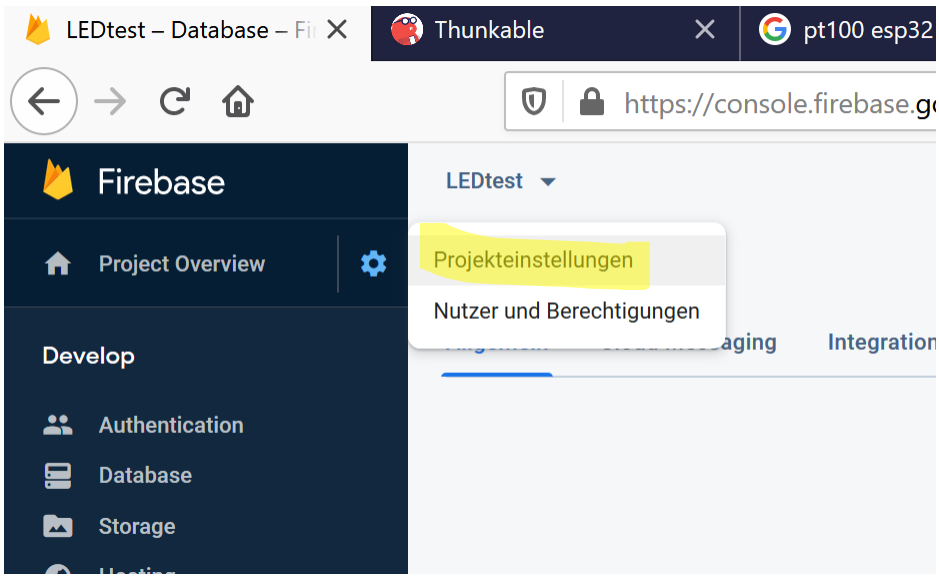
9. Auf «Veröffentlichen» klicken

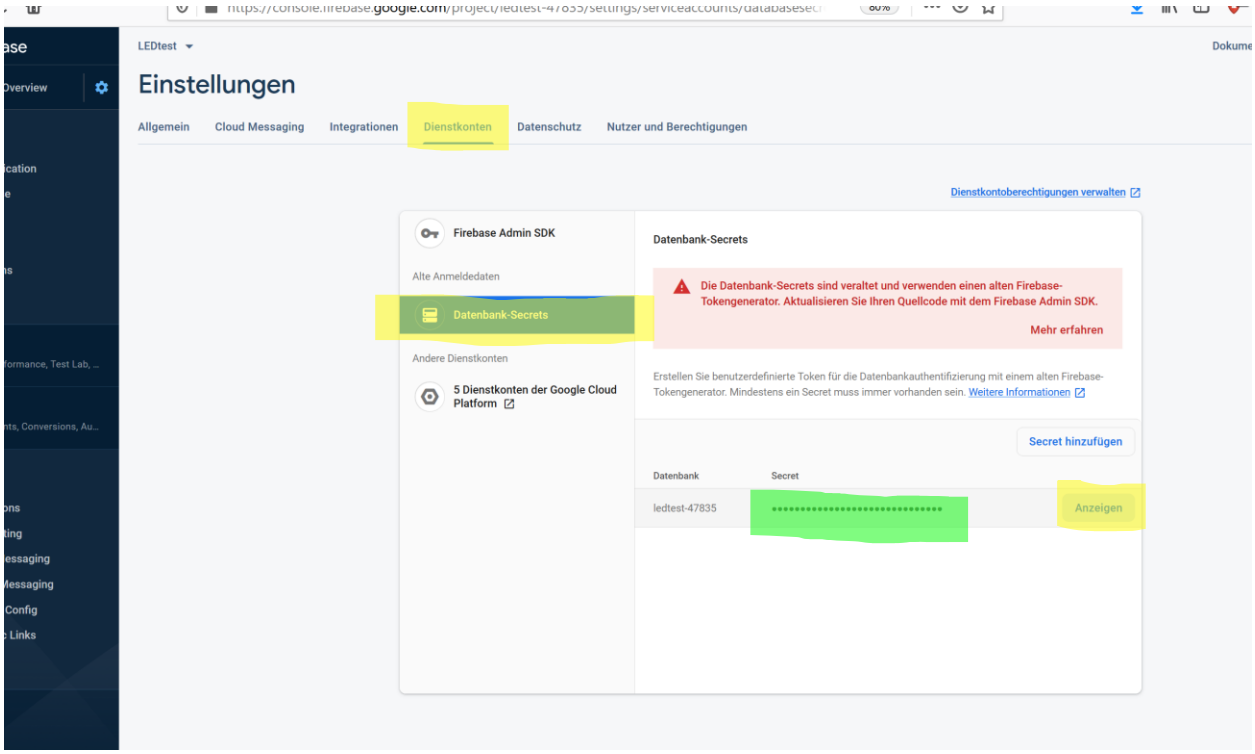
SCHRITT 7 WICHTIGE SCHLÜSSEL

HOST URL:



API KEY (SECRET KEY):





SCHRITT 8 X.THUNKABLE

Ziel:

Dies ist die Software mit welcher das APP geschrieben wird.:

Vorgehen:

1. Kostenlos registrieren auf: https://x.thunkable.com
2. APP «Thunkable live» downloaden. Auf Google Playstore oder Apple Appstore
3. klicke auf «create new APP»

Blocks Ansicht:

Frontend Ansicht: