TDS SENSOR – WASSERQUALITÄT MESSEN

Eigenes Projekt 2023/24

Borycki Oliver Dominik
oborycki@student.tgm.ac.at

04.06.2024

Inhalt

Kurze Beschreibung 1
Komponenten: 1
TDS Sensor Pins: 1
I2C Display 1
Ausgeben auf dem Display 1
Probleme dabei: 2
Vorschritt: Mit dem WiFi zu verbinden 2
Auf ThingSpeak ausgeben 2
Code: 2
Foto: 5

Kurze Beschreibung

Im Folgenden Programm messen wir die Wasserqualität mithilfe eines TDS-Sensors und geben sie auf einem Display aus. Zusätzlich zudem sollte ich auf der Plattform Wasser Qualitätsmesser - ThingSpeak IoT eine Graphik von dem Wert sehen, dieser verändert sich minimal.

Komponenten:

7 Jumperkabel, TDS Sensor, Micro USB kabel, i2c Display

TDS Sensor Pins:

TDS Sensor	**ESP32**
GND	GND
VCC	3.3V
Data	GPIO 27

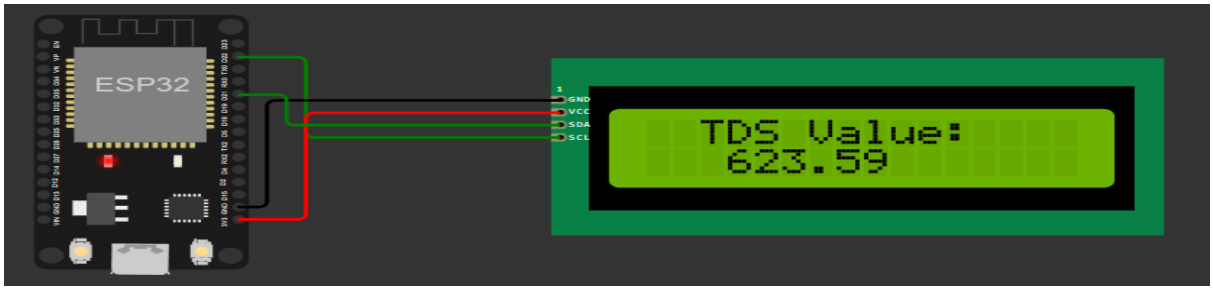
I2C Display

I2C	**ESP32**
GND	GND
VCC	3.3V

SDA	PIN 21
SCL	PIN 22

Ausgeben auf dem Display

Um zu überprüfen wie alles nach Plan laufen würde, ohne das mein i2c Display dabei spinnt, habe ich es ipubiumah - Wokwi ESP32, STM32, Arduino Simulator laufen lassen.



Probleme dabei:

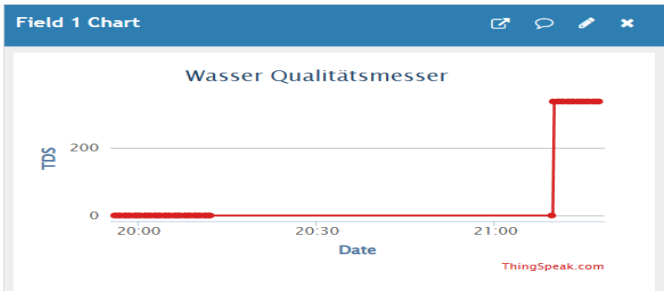
Es gab einige Probleme, bei dieser Funktion. Lösungen gab es keine, mein Kompromiss war es zumindest auf Wokwi auszugeben. Gesagt getan, oben sieht man das Ergebnis. Problem war vermutlich mein Bildschirm, da muss man sehr vorsichtig sein beim Einkaufen.

Vorschritt: Mit dem WiFi zu verbinden

Bevor wir uns mit ThingSpeak verbinden sollten wir uns mit WiFi verbinden. Dazu gabe es einige Vorlagen aus dem Internet. Probleme gab es trotzdem. Die Probleme konnte ich diesmal, aber lösen. Ich habe bei meinem Hotspot ein paar Sachen umgestellt und schon hat es funktioniert. Notfalls habe ich mit einem fremden Hotspot verbunden.

Auf ThingSpeak ausgeben

Jetzt der wichtigste Schritt: Die Werte auf dem Graph. Um das zumachen habe ich die ThingSpeak Bibliothek installiert und meinen Code weitergeändert. Dafür habe ich die Werte für den Kanal eingestellt und später so das sich der Graph auf den TDS Wert ändert. Scheint leicht zu gehen, hat mich trotzdem einiges an Überwindungskraft gekostet. Ich bin sicher 3 Stunden dran gesessen nur umzuverstehen, wieso ich nach dem Verbinden mit dem WiFi ganz andere Werte habe als ohne. Ich habe versucht das aus dem Setup zu bringen, da es vorher im Setup war, später habe ich noch paar Kleinigkeiten geändert. Um ehrlich zu sein, weiß ich nicht wieso es diesmal funktioniert hat und vorher nicht. So schaut es jedoch auf dem Graph aus, man ignoriere die fehlerhaften nuller Werte davor:



Code:

Und hier ist der gesamte Code:

```
#include <WiFi.h>
#include "ThingSpeak.h"

#define TdsSensorPin 27
#define VREF 3.3              // analog reference voltage(Volt) of the ADC
#define SCOUNT  30            // sum of sample point

int analogBuffer[SCOUNT];     // store the analog value in the array, read from ADC
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0;
int copyIndex = 0;

float averageVoltage = 0;
float tdsValue = 0;
float temperature = 25;

const char* ssid = "oli";   // your network SSID (name)
const char* password = "Oliver07?";   // your network password

WiFiClient  client;

unsigned long myChannelNumber = 2488467;
const char * myWriteAPIKey = "FFCXSSYISKMJ8PD5";

// Timer variables
unsigned long lastTime = 0;
unsigned long timerDelay = 30000;
unsigned long lastWiFiCheck = 0;
unsigned long wifiCheckInterval = 5000; // Check WiFi status every 5 seconds

int getMedianNum(int bArray[], int iFilterLen) {
  int bTab[iFilterLen];
  for (byte i = 0; i < iFilterLen; i++)
    bTab[i] = bArray[i];
  int i, j, bTemp;
  for (j = 0; j < iFilterLen - 1; j++) {
    for (i = 0; i < iFilterLen - j - 1; i++) {
      if (bTab[i] > bTab[i + 1]) {
```

```
        bTemp = bTab[i];
        bTab[i] = bTab[i + 1];
        bTab[i + 1] = bTemp;
      }
    }
  }
  if ((iFilterLen & 1) > 0) {
    bTemp = bTab[(iFilterLen - 1) / 2];
  }
  else {
    bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
  }
  return bTemp;
}

void setup() {
  Serial.begin(115200);  //Initialize serial
  pinMode(TdsSensorPin, INPUT);

  ThingSpeak.begin(client);  // Initialize ThingSpeak
}

void loop() {
  if ((millis() - lastTime) > timerDelay) {
    // Get a new temperature reading
    static unsigned long analogSampleTimepoint = millis();
    if (millis() - analogSampleTimepoint > 40U) { //every 40 milliseconds, read the analog
value from the ADC
      analogSampleTimepoint = millis();
      analogBuffer[analogBufferIndex] = analogRead(TdsSensorPin);    //read the analog
value and store it into the buffer
      analogBufferIndex++;
      if (analogBufferIndex == SCOUNT) {
        analogBufferIndex = 0;
      }
    }

    static unsigned long printTimepoint = millis();
    if (millis() - printTimepoint > 800U) {
      printTimepoint = millis();
      for (copyIndex = 0; copyIndex < SCOUNT; copyIndex++) {
        analogBufferTemp[copyIndex] = analogBuffer[copyIndex];

        // read the analog value more stable by the median filtering algorithm, and convert
to voltage value
        averageVoltage = getMedianNum(analogBufferTemp, SCOUNT) * (float)VREF / 4096.0;

        // temperature compensation formula: fFinalResult(25^C) = fFinalResult(cur-
rent)/(1.0+0.02*(fTP-25.0));
```

```
        float compensationCoefficient = 1.0 + 0.02 * (temperature - 25.0);
        // temperature compensation
        float compensationVoltage = averageVoltage / compensationCoefficient;

        // convert voltage value to tds value
        tdsValue = (133.42 * compensationVoltage * compensationVoltage * compensationVolt-
age - 255.86 * compensationVoltage * compensationVoltage + 857.39 * compensationVoltage) *
0.5;

        Serial.print("TDS Value:");
        Serial.print(tdsValue, 0);
        Serial.println("ppm");

        // Check WiFi status and connect if not connected
        if (millis() - lastWiFiCheck > wifiCheckInterval) {
          lastWiFiCheck = millis();
          if (WiFi.status() != WL_CONNECTED) {
            Serial.println("WiFi not connected. Attempting to connect...");
            connectToWiFi();
          }
        }

        // Write to ThingSpeak
        int x = ThingSpeak.writeField(myChannelNumber, 1, tdsValue, myWriteAPIKey);
        if (x == 200) {
          Serial.println("Channel update successful.");
        }
        else {
          Serial.println("Problem updating channel. HTTP error code " + String(x));
        }
        lastTime = millis();
      }
    }
  }
}

void connectToWiFi() {
  WiFi.begin(ssid, password);
  unsigned long startAttemptTime = millis();
  while (WiFi.status() != WL_CONNECTED && millis() - startAttemptTime < timerDelay) {
    delay(100);
  }
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("WiFi connected.");
  }
  else {
    Serial.println("Unable to connect to WiFi.");
  }
}
```

Foto: