

Datenlogger mit ESP-01

Wetterdaten
Aktueller Wert

ESP-01

Das ESP-01 ist das kleinste Modul aus der Reihe der Baugruppen mit dem ESP8266. Neben dem IC ist auch ein Speicher mit 512kByte oder 1MByte enthalten.

Inbetriebnahme



Das ESP-Modul benötigt 3,3V als Betriebsspannung. Zur Programmierung nutze ich einen USB-Seriell-Wandler. Zwischen das ESP-Modul und den USB-Seriell-Wandler habe ich eine Adapterplatine gesteckt, die die 5V auf 3,3V wandelt und den TX-Pegel mit einem Spannungsteiler (1k/2,2k) reduziert. Außerdem sind darauf zwei Taster an RST und GPIO untergebracht, die für die Programmierung notwendig sind.

Zusätzlich wurde ein 5-poliger Steckplatz für den Anschluss von I²C-Komponenten vorgesehen. Der ESP8266 verwendet hierfür GPIO0 (SDA) und GPIO2 (SCL). Am Steckverbinder liegen außerdem 5V und 3,3V an. Der zweite 3-polige Stecker führt 3,3V, GND, und GPIO2 für den Anschluss eines DHT11/22. Im Betrieb wird das System mit einem 5V-Steckernetzteil betrieben.

Quellen:

<https://blog.thesen.eu/esp8266-wlan-microcontroller-mit-der-arduino-ide-programmieren/>

<http://www.instructables.com/id/Using-the-4-Pins-of-the-ESP8266-01/>

Anschlüsse

Das ESP_01-Modul hat die zwei seriellen Anschlüsse Rx und Tx, die u.U. auch als GPIOs genutzt werden können und die GPIOs 0 und 2.

Quellen:

<http://stefanfrings.de/esp8266/index.html>

Sensoren

Die ersten Versuche wurden mit einem Sensor für Temperatur und Luftfeuchtigkeit (DHT11) gemacht, der am ESP_01-Modul am Anschluss GPIO2 angeschlossen wurde.

Die aktuelle Variante arbeitet mit einem Sensor BME280, der über I²C angeschlossen ist und Temperatur, Luftfeuchte und Luftdruck misst. Die beiden I²C-Ports am ESP8266 sind GPIO0 (SDA) und GPIO2 (SCL).

Luftfeuchte-Sensor

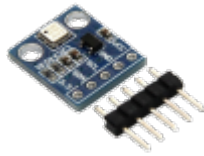


Der Sensor für die Luftfeuchtigkeit arbeitet mit einem Chip von Silicon Labs, Si7021, oder anderer Hersteller, die einen ähnlichen Chip anbieten (SHT21, HUT21). Dieser Chip liefert gleichzeitig noch einen Temperatur-Wert. Versorgt wird die Baugruppe aus der 5V-Betriebsspannung. Auf der kleinen

Leiterplatte ist ein 3,3V-Regler enthalten.

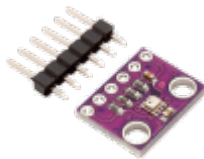
Nach einer Weile stellte ich fest, dass der IC sehr abweichende Feuchte-Werte lieferte. Ich habe den Chip gegen einen HDC1080, der ein gleiches Pinout hat, gewechselt. (12.01.2019)

Luftdruck-Sensor



Der Sensor für den Luftdruck arbeitet mit einem Chip der kompatibel zum Bosch BMP085 ist. Dieser Chip liefert ebenfalls einen Temperaturwert. Auch diese Leiterplatte hat einen 3,3V-Regler, so dass die Baugruppe mit 5V versorgt wird.

Temperatur-, Luftdruck-, Luftfeuchte-Sensor



Der Sensor arbeitet mit einem Chip der kompatibel zum Bosch BME280 ist. Dieser Chip liefert einen Temperaturwert, den Luftdruck und die relative Luftfeuchtigkeit. Diesen Sensor nutze ich seit 26.06.2019, da der HDC1080 fehlerhafte Werte bei der Luftfeuchtigkeit lieferte.

Bibliotheken und Beispiele sind in der Arduino-Umgebung vorhanden. Bei der Adressierung ist zu beachten, dass die I²C-Adresse von der Belegung des SDO-Pins abhängig ist. Ggf. muss eine Datei der Bibliothek angepasst werden:

- SDO=L → 0x76
- SDO=H → 0x77

Installation

Die Sensoren befinden sich in einer gelöcherten Box auf dem Fensterbrett auf der Nordseite des Hauses im 8. Stock. Der ESP8266 befindet sich auf der Innenseite und wird mit einem Steckernetzteil versorgt.

Aufgrund des Standortes gibt es im Sommer durch Sonneneinstrahlung trotz einiger Abschirmversuche eine abendliche Temperaturspitze.

Programm-Ablauf

Die Sensoren werden ein Mal pro Minute abgefragt und es erfolgt eine gleitende Mittelwertbildung. Alle 10 Minuten werden die gemittelten Daten an einen PHP-Script auf der Webseite übergeben. Dort werden die Daten in eine MySQL-Datenbank gespeichert, um spätere Auswertungen zu ermöglichen.

Arduino-IDE

Ich habe mich für die Arduino-IDE als Programmierumgebung entschieden. Hierfür gibt es eine [IDE-Erweiterung](#) die auch die ESP8266WiFi-Libraries mitbringt.

Wie diese installiert wird und der Funktionsumfang ist hier beschrieben:

<http://arduino-esp8266.readthedocs.io/en/latest/installing.html>

Weiterhin wird die Bibliothek für die I²C-Verbindung genutzt

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

Diese muss mit `wire.begin(0,2)` für die beiden Ports eingestellt werden.

Für den BMP085 und Si7021 nutze ich die Adafruit-Bibliotheken:

```
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
#include "Adafruit_Si7021.h"
```

Für den HDC1080 gibt es folgende Bibliothek

```
#include <ClosedCube_HDC1080.h>
```

Für den BME280 gibt es folgende Bibliothek

```
#include <Adafruit_BME280.h>
```

Um einen Zeitstempel zu erhalten wird auf einen NTP-Server zugegriffen und die Zeit abgefragt. Dafür nutze ich die beiden Dateien `time_ntp.h/.cpp` von Stefan Thesen/Michael Margolis.

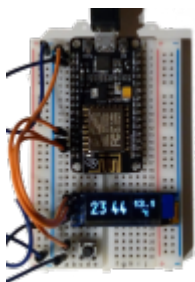
Web-Server

Getestet habe ich die Funktionen auf einem RaspberryPi, der einen Apache-Webserver mit MySQL-Datenbank installiert hat.

Wie oben beschrieben, verbindet sich der ESP8266 mit dem Web-Server, indem er über GET die Daten an einen PHP-Script übergibt. Dieser Script fügt die Daten einer MySQL-Tabelle zu.

Die Daten können dann über Web-Seiten mit PHP-Scripts abgefragt und dargestellt werden. Die Abfrage erzeugt eine [Ausgabe im JSON-Format](#). Eine Möglichkeit ist die Nutzung von [Google-Charts](#) wie im obigen Beispiel der Wetterdaten. Dabei muss beachtet werden dass bei diesem Datumsformat der Monat ab 0 indiziert sein muss.

MQTT-Client



Neben der Speicherung der Daten habe ich Datenverbindungen mit einem MQTT-Server aufgebaut, wie z.B. [CloudMQTT](#), wo es für einfache Tests ein kostenloses Angebot gibt. Für den Arduino gibt es die Bibliothek „PubSubClient“ mit welcher Daten zu einem MQTT-Broker veröffentlicht und auch entsprechende Daten aboniert werden können. Ein einfacher Client wurde mit einem ESP-12-Modul und mit einer OLED-Anzeige 128×32-Pixel realisiert.

Eine weitere Möglichkeit zur Darstellung der aktuellen Wetterdaten ist das Smartphone mit der App [MQTT-Dash](#).

From:

<http://simandit.de/simwiki/> - Wiki

Permanent link:

<http://simandit.de/simwiki/doku.php?id=weiteres:esp8266:datalogger>

Last update: **2019/07/02 22:24**

