

## Ein paar Gedanken zum Projekt Wasserqualität

### Vorbemerkungen:

Ein Messgerät mit einem Webserver aufzusetzen, das im Feld zum Einsatz kommen soll, macht keinen Sinn, denn in der Regel gibt es dort keinen Internetzugang.

Allenfalls kann man ein Smartphone als Hotspot verwenden. Aber auch dann macht ein Webserver keinen Sinn, weil die Online-Zeit sehr begrenzt ist. Und wer sollte ihn in der kurzen Zeit finden und dann darauf zugreifen? Zudem sind die Kosten wegen des anfallenden Datenvolumens möglicherweise recht hoch.

Daher bietet sich in diesem Fall an, die Daten auf einen IOT-Broker (bspw. Ubidots) hochzuladen. Ubidots ist einfach zu bedienen und für Ausbildungszwecke kostenfrei. Auf diese Webseite kann jederzeit von jedem Ort zugegriffen werden, auch wenn das Messgerät nicht Online ist.

### Vorüberlegungen

Die Sensoren sind zwar gesetzt, dennoch müsst Ihr noch ein paar Dinge festlegen, bevor Ihr anfangt.

### Welche Anzeige wollt Ihr verwenden? Mit oder ohne Touchscreen? Welche Größe soll das Display haben?

Es gibt diverse Möglichkeiten mit unterschiedlichen Konsequenzen.

#### a) Touchscreen und oder Tastatur:

In der Praxis, d.h. im Labor und besonders im Feld sind Tastaturen eher hinderlich. Daher besitzen die meisten modernen Messgeräte heute Touchscreens.

#### b) Displays

Es gibt sehr unterschiedliche Anschlussmöglichkeiten für Displays, z.B. I2C, SPI, CSI, HDMI ...

##### 1. Displays, die direkt auf den GPIO aufgesteckt werden.

Beispiel (willkürliche Auswahl!):

<https://www.berrybase.de/4-0-display-fuer-raspberry-pi-mit-resistivem-touchscreen>

**Vorteil:** Keine Verkabelung nötig

**Nachteil:** Der GPIO-Port ist blockiert

##### 2. Displays mit HDMI

Beispiel (willkürliche Auswahl!):

<https://www.berrybase.de/universal-5-0-display-mit-hdmi-eingang-und-resistivem-touchscreen>

**Vorteile:** einfache Verkabelung, Kabellänge „beliebig lang“, der GPIO-Port bleibt zugänglich, auch PC-Monitoren können verwendet werden

**Nachteile:** Für das Touchdisplay muss ein weiteres Kabel verwendet werden, das einen USB-Port blockiert.

##### 3. Displays mit CSI

Beispiele (willkürliche Auswahl!):

<https://www.berrybase.de/7-zoll-kapazitives-touch-display-fuer-raspberry-pi-800x480>

<https://www.berrybase.de/raspberry-pi-touch-display-2>

<https://de.aliexpress.com/item/1005007476546878.html>

**Vorteile:** einfache Verkabelung, der GPIO-Port bleibt zugänglich, kein Extra-Kabel für den Touchscreen

**Nachteile:** Kabellänge kurz: Der Raspberry Pi muss am Display befestigt werden

Anmerkung: Nach meinem Dafürhalten ist die dritte Option die beste.

#### c) Displaygröße

Die Wahl der Displaygröße ist für die gewünschte Kompaktheit des Messgerätes mit entscheidend. Wegen der Ablesbarkeit sollte das Display jedoch nicht kleiner als 2,8“ (bei 5 anzuzeigenden Messwerten) gewählt werden.

## Anzeige der Daten / GUI:

Das Programmieren mit Tkinter ist nicht so intuitiv, wie es allgemein dargestellt wird. Bei komplexeren Layouts verliert man leicht die Übersicht.

Daher sollte man sich unbedingt eine Layout-Skizze anfertigen, in der alle Fenster mit Namen, Überschriften, Position (x-start, y-start) und Größe (x-länge, y-länge) eingetragen sind, **bevor** man mit der Programmierung des Displays beginnt.

Hier gibt es ein gutes Tutorial zu den Koordinaten eines Displays, das allerdings für Arduino C++ ausgelegt ist. Die Methoden könnt ihr zwar nicht verwenden, aber das Prinzip der Koordinaten ist das gleiche.

<https://learn.adafruit.com/adafruit-gfx-graphics-library/coordinate-system-and-units>

## Material:

- Raspberry Pi 3
- Temperatur: DS18B20
- Gravity: Analog Turbidity Sensor for Arduino
- Gravity: Analog Dissolved Oxygen / DO Sensor Meter Kit for Arduino
- Gravity: Analog pH Sensor/Meter Kit V2 (Arduino & Raspberry Pi & micro:bit Compatible)
- Gravity: Analog TDS Sensor/ Meter for Arduino
- ADS1115
- SD-Karte
- **Display**
- ???

## Analog-Digital-Wandler ADC1115

### Anmerkung:

Die vier analogen Sensoren liefern an ihrem Ausgang eine Spannung, die proportional bzw. umgekehrt proportional zum Messwert ist.

**Ein Problem, das zu lösen ist: Alle analogen Sensoren müssen vor dem Messbeginn kalibriert werden!**

**Der Raspberry Pi besitzt keinen AD-Wandler.** Daher muss zusätzlich ein ADC eingesetzt werden. Eine gute Wahl ist der ADC1115, der 4 Eingangskanäle besitzt, d.h. es können 4 Sensoren angeschlossen werden.

**Das Auslesen aller Sensorwerte ist damit gleich. Allein die Interpretation dieser Messwerte unterscheidet sich! Siehe dazu die Wiki zu den Sensoren.**

Der ADC muss zu Programmbeginn initialisiert werden. Im Programm muss dann dem ADC mitgeteilt werden, welchen Kanal und damit Sensor man auslesen möchte.

Ein sehr gutes Tutorial zum ADC1115 gibt es hier:

<https://learn.adafruit.com/adafruit-4-channel-adc-breakouts>

Hier könnt ihr nachlesen, wie die Messwerte zu interpretieren sind und wie man die Sensoren kalibriert.

**Gravity: Analog Turbidity Sensor for Arduino**

[https://wiki.dfrobot.com/Gravity\\_Analog\\_TDS\\_Sensor\\_Meter\\_For\\_Arduino\\_SKU\\_SEN0244](https://wiki.dfrobot.com/Gravity_Analog_TDS_Sensor_Meter_For_Arduino_SKU_SEN0244)

**Gravity: Analog Dissolved Oxygen / DO Sensor Meter Kit for Arduino**

[https://wiki.dfrobot.com/Gravity\\_Analog\\_Dissolved\\_Oxygen\\_Sensor\\_SKU\\_SEN0237](https://wiki.dfrobot.com/Gravity_Analog_Dissolved_Oxygen_Sensor_SKU_SEN0237)

**Gravity: Analog pH Sensor/Meter Kit V2 (Arduino & Raspberry Pi & micro:bit Compatible)**

[https://wiki.dfrobot.com/Gravity\\_Analog\\_pH\\_Sensor\\_Meter\\_Kit\\_V2\\_SKU\\_SEN0161-V2](https://wiki.dfrobot.com/Gravity_Analog_pH_Sensor_Meter_Kit_V2_SKU_SEN0161-V2)

**Gravity: Analog TDS Sensor/ Meter for Arduino**

[https://wiki.dfrobot.com/Gravity\\_Analog\\_TDS\\_Sensor\\_Meter\\_For\\_Arduino\\_SKU\\_SEN0244](https://wiki.dfrobot.com/Gravity_Analog_TDS_Sensor_Meter_For_Arduino_SKU_SEN0244)

**Temperatursensor DS18B20**

Der Temperatursensor DS18B20 ist bereits kalibriert und liefert aufbereitete, digitale Messwerte.

Jeder DS18B20- Sensor besitzt eine eigene ID, die man ermitteln muss (siehe nachfolgende Beispiele).

Hier sind ein paar Links mit Programmbeispielen.

<https://tutorials-raspberrypi.de/raspberry-pi-temperatur-mittels-sensor-messen/>

<https://st-page.de/2018/01/20/tutorial-raspberry-pi-temperaturmessung-mit-ds18b20/>

<https://cbrell.de/blog/raspilab-wetterstation-dritte-mission-temperatur-messen-mit-dem-ds18b20/>

Nachfolgend noch ein Link mit einem Beispiel zum Versenden der Daten an einen MQTT-Broker (müsste noch für Ubidots angepasst werden):

[https://www.laub-home.de/wiki/Raspberry\\_Pi\\_DS18B20\\_Temperatur\\_Sensor](https://www.laub-home.de/wiki/Raspberry_Pi_DS18B20_Temperatur_Sensor)

**IOT mit Ubidots**

Es empfiehlt sich, sich in das Arbeiten mit Ubidots mit einem Programm wie dem nachfolgenden einzuarbeiten.

**Beachtet:** Es werden nur Werte und die Sensorzuordnung versendet. Das erfolgt für alle Sensoren in gleicher Weise.

**Beachtet:** Bevor man ein Programm testen kann, muss man einen Account bei Ubidots angelegt haben. Ubidots bietet neben teuren, professionellen Accounts auch kostenlose „educational accounts“, die völlig ausreichend sind.

<https://help.ubidots.com/en/articles/1077054-diy-raspberry-pi-temperature-system-with-ubidots>