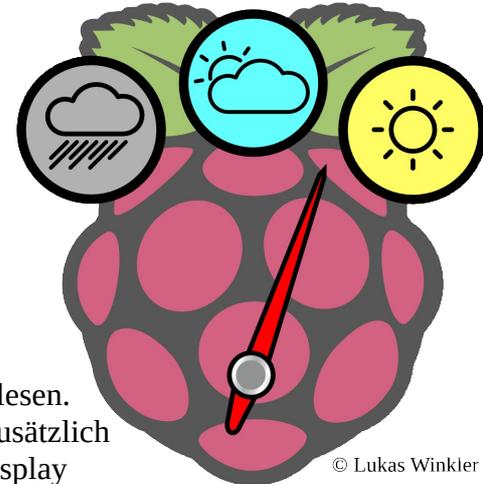


# Umweltdatenmessung mit dem Raspberry Pi

Lukas Winkler

In meinem Projekt möchte ich mithilfe eines Raspberry Pi Umweltdaten aufzeichnen, speichern und auswerten.

Zur Aufzeichnung verwende ich einen Temperatursensor, einen Luftdrucksensor, einen Luftfeuchtesensor und einen Luftqualitätssensor. Die Daten werden abgespeichert und einerseits grafisch mithilfe von Javascript dargestellt und andererseits rechnerisch mit einem von mir geschriebenen Python-Programm ausgewertet.



## Aufzeichnung:

Die Sensoren werden regelmäßig von einem Shell-Script ([main.sh](#)) ausgelesen. Dieses Skript bereitet die Daten auf und speichert sie in eine csv-Datei. Zusätzlich werden Textdateien generiert, welche von der Weboberfläche und dem Display regelmäßig ausgelesen werden. Bei jedem 1000. Aufruf wird die CSV-Datei gesichert, per E-Mail an mich verschickt und ich darüber mit einer Pushbenachrichtigung am iPad benachrichtigt.

## Display:

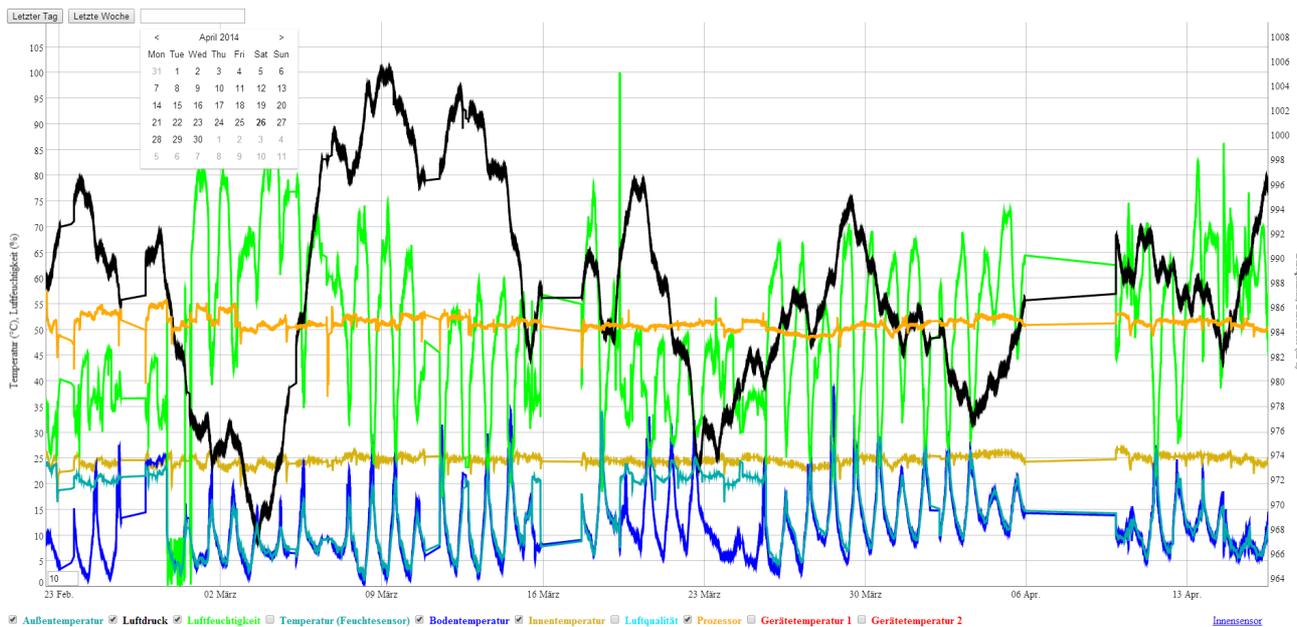
Damit man nicht immer zum Computer laufen muss, um die aktuellen Werte zu erfahren habe ich ein Display an den Raspberry Pi angeschlossen. Ich habe ein [Beispielprogramm](#) erweitert, damit es statt einen statischen Text anzuzeigen, eine von main.sh generierte Textdatei ausliest und der Inhalt auf mehreren Seiten dargestellt wird. (→ [display.py](#))



## Webinterface:

Zur grafischen Darstellung der Daten habe ich zwei Webseiten erstellt, die auf einem Apache-Webserver am Raspberry Pi liegen. Mithilfe von Javascript und Dygraphs wird ein Diagramm der Daten dargestellt. ([Beispiel](#), Vorsicht, lange Ladezeit). Zur Darstellung der aktuellen Werte gibt es [eine zweite Seite](#), die mithilfe von Javascript alle 5 Sekunden die aktuellen Daten nachläd und grafisch darstellt. Beide Seiten können auf mobilen Geräten (iOS, eventuell auch Android) nachdem sie zum Homebildschirm hinzugefügt wurden als Web-Application genutzt werden (im Vollbildmodus, mit eigenem Icon; funktioniert bei statischem Beispiel nicht). Wenn die Aufzeichnung gerade aus irgend einem Grund nicht laufen soll, wird der Webseitenbesucher benachrichtigt.

|                                 |                   |                                                                                      |
|---------------------------------|-------------------|--------------------------------------------------------------------------------------|
| Datum/Uhrzeit:                  | 16.04.14 13:11:46 |                                                                                      |
| Innentemperatur:°C              | 24.5              |  |
| Luftqualität:                   | 1202              |   |
| Gerätetemperatur 1:°C           | 25.9              |  |
| Gerätetemperatur 2:°C           | 25.9              |  |
| Prozessor:                      | 50.8              |  |
| Außentemperatur:°C              | 12.1              |   |
| Temperatur (Luftfeuchtesensor): | °C 12.0           |   |
| Bodentemperatur:°C              | 14.1              |   |
| Luftfeuchte:% rel LF            | 44.3              |   |
| Luftdruck:hPa                   | 995.4             |   |



## Endauswertung:

Nachdem Daten aufgezeichnet wurden, kann die fertige CSV-Datei rechnerisch ausgewertet werden. Zunächst muss sie [mit einem Shell-Script](#) vorbereitet werden. Danach kann es ein [von mir geschriebenes Python\(3\)-Programm](#) auswerten. Zunächst wird die Datei nach Ausreißern durchsucht und etwaige werden angezeigt. Danach kann man den Zeitraum auswählen, in dem die Messung ausgewertet wird. Danach wird das arithmetische Mittel, Minimum, Maximum und Standardabweichung für jeden Sensor berechnet. ([„Video“ des Programms](#))

## Gnuplot:

Ursprünglich habe ich versucht die Diagramme mit [Gnuplot](#) zu generieren. Da die fertigen Bilder jedoch statisch sind und der Raspberry Pi mit zunehmender Datenmenge sehr lange benötigt, habe ich dies aufgegeben (zu Gunsten von Dyngraphs)

## Autostart:

Damit der Start und Stop der Aufzeichnung (und des Displays) einfacher funktioniert habe ich ein [init.d-Skript](#) erstellt. Nun kann mit „sudo service aufzeichnung {start/stop/restart} die Aufzeichnung einfach gesteuert werden. Zusätzlich wird die Aufzeichnung beim Hochfahren gestartet und vorm Herunterfahren gestoppt.

## diverses:

Zusätzlich zu den wichtigen Teilen habe ich noch ein paar kleinere Funktionen, wie ein [Windows Vista/7-Desktop-Gadget](#) oder ein (etwas zu buntes) [Logo](#) für das Projekt.

## Fremddateien:

Für mein Projekt nutze ich sehr viele Programmteile, die nicht von mir sind (zum Auslesen der Sensoren, zur Darstellung der Webseite). Diese findet man unter [Fremddateien](#) bzw. [Webinterface/Fremddateien](#).

## Links mit mehr Informationen:

Alle Dateien zum Projekt sammle ich zentral bei Github:

<https://github.com/Findus23/Umweltdatenmessung>

Viele (zum Teil alte) Beispiele findet man unter [Github-Pages \(Quelltext dazu\)](#)

Ursprünglich habe ich auch [eine Webseite](#) für das Projekt eingerichtet, jedoch ist es zu viel Aufwand diese regelmäßig zu aktualisieren und daher ist der Großteil des Inhalts schon längst veraltet.

Die tagesaktuellen Neuerungen findet man [auf Github](#), größere Änderungen [hier](#)

Knapp 100 Bilder, die die Entstehung des Projekts dokumentieren, findet man [auf Flickr](#).

Manchmal habe ich auch [Videos auf Youtube](#) hochgeladen.

## Hardwareliste:

- DS18B20
  - Temperatursensor
  - sehr zuverlässig
  - sehr einfach auszulesen
  - Genauigkeit: +- 0,5°C
  - mehrere Sensoren lassen sich parallel anschließen
- DHT11/DHT22
  - Luftfeuchte und -temperatur
  - DHT11 sehr ungenau, daher DHT22
  - leider sehr viele Ausfälle
    - gibt jedes 3. Mal kein Ergebnis zurück
  - gute Anleitung
- BMP085
  - Luftdruck, Luftqualität
  - enorm zuverlässig (einziger Sensor, der nie Messfehler macht)
  - ursprünglich für Arduino gedacht
- „Luftqualitätssensor“
  - misst nicht direkt die Luftqualität
  - VOC-Sensor
  - misst nur relativen Qualitätsunterschied
  - VOLTCRAFT CO-20
    - leider nicht mehr erhältlich (Eigenmarke von Conrad)
    - identisches Modell wird bei Velux Deutschland verkauft
- 32 Zeichen LCD-Display