

# **Umweltdatenmessung mit dem Raspberry Pi**

Vorwissenschaftliche Arbeit verfasst von

**Lukas Winkler**

Klasse 8A



Betreuer: MMag. Matthias Kittel

BRG Rechte Kremszeile  
Rechte Kremszeile 54  
3500 Krems an der Donau

Krems an der Donau, Januar 2015

Diese Arbeit wurde mit Texmaker geschrieben, in Palatino mit Hilfe von [pdfLATEX](#) und [Biber](#) gesetzt.

Die LATEX Vorlage von Karl Voit basiert auf [KOMA script](#) und steht im Internet zum Download bereit: <https://github.com/novoid/LaTeX-KOMA-template>

# Abstract

This is a placeholder for the abstract. It summarizes the whole thesis to give a very short overview. Usually, this the abstract is written when the whole thesis text is finished.

# Inhaltsverzeichnis

<b>Abstract</b>	<b>iii</b>
<b>1. Einleitung</b>	<b>1</b>
<b>2. Hardware</b>	<b>2</b>
2.1. Der Raspberry Pi . . . . .	2
2.1.1. Geschichte . . . . .	3
2.1.2. Technische Daten . . . . .	3
2.2. Sensoren . . . . .	4
2.2.1. Temperatur . . . . .	4
2.2.2. Luftfeuchtigkeit . . . . .	5
2.2.3. Luftdruck . . . . .	6
2.2.4. Luftqualität . . . . .	7
2.3. Display . . . . .	8
2.4. Anschluss . . . . .	8
<b>3. Software</b>	<b>9</b>
3.1. main.sh . . . . .	9
3.1.1. Allgemeines . . . . .	9
3.1.2. Messung . . . . .	10
3.1.3. Speichern, Aufbereiten und Verarbeiten . . . . .	12
3.2. Display . . . . .	14
3.3. Webinterface . . . . .	14
3.3.1. Livedaten . . . . .	15
3.3.2. Diagramme . . . . .	16
3.4. Endauswertung . . . . .	18
3.5. diverses . . . . .	18
3.5.1. autostart . . . . .	18
3.5.2. mitternacht.sh . . . . .	18

## Inhaltsverzeichnis

3.5.3. Weather Underground . . . . .	18
<b>A. Weitere Informationen</b>	<b>20</b>
<b>Literatur</b>	<b>21</b>
<b>Abbildungsverzeichnis</b>	<b>24</b>
<b>Dateiverzeichnis</b>	<b>25</b>
<b>Glossar</b>	<b>26</b>

# Todo list

genauere Beschreibung des Projekts . . . . .	1
Einleitung . . . . .	2
Anzahl hinzufügen . . . . .	9
eventuell entfernen . . . . .	18

# 1. Einleitung

Im letzten Jahr habe ich mich damit beschäftigt, wie man mithilfe eines Raspberry Pi Umweltdaten messen, aufzeichnen und auswerten kann. Hierzu verwende ich mehrere Sensoren, die die Lufttemperatur (sowohl im Klassenraum, als auch außen), Luftfeuchtigkeit, Luftdruck und die relative Luftqualität. Diese Daten werden als [CSV-Datei](#) gespeichert und können grafisch und rechnerisch ausgewertet werden.

genauere  
Beschrei-  
bung des  
Projekts



Abbildung 1.1.: Messstation (eigenes Werk)

## 2. Hardware

### 2.1. Der Raspberry Pi

Der *Raspberry Pi* ist ein Einplatinencomputer, der 2012 von der *Raspberry Pi Foundation* auf den Markt gebracht wurde.

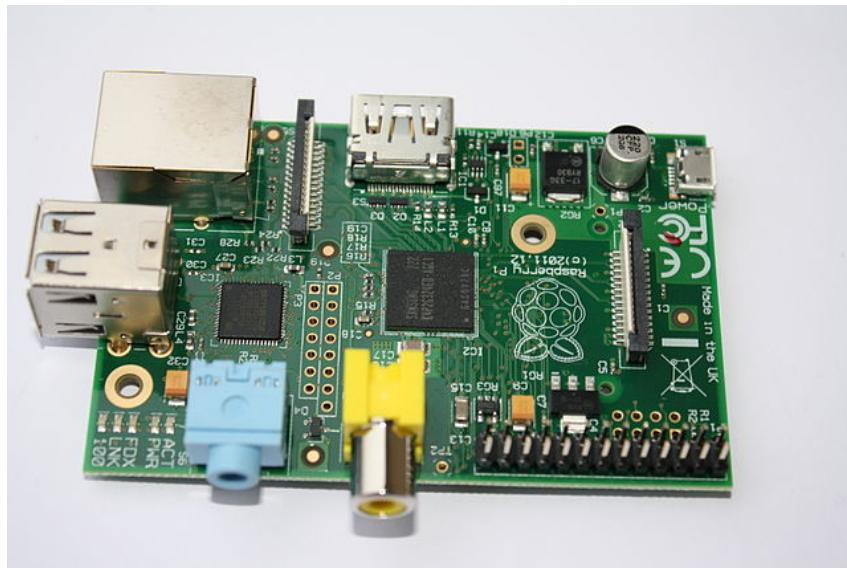


Abbildung 2.1.: Raspberry Pi - Modell B<sup>1</sup>

---

<sup>1</sup>Bohk, *Rev. 2 des Raspberry Pi Model B - made in UK*

## 2. Hardware

### 2.1.1. Geschichte

Ursprünglich war er als günstiger Computer gedacht, um britischen Jugendlichen das Programmieren näher zu bringen. An der *University of Cambridge* stellte man fest, dass die Vorkenntnisse von Studienanfängern immer geringer wurden, weil sie – sowohl privat als auch in der Schule – sich immer weniger mit der Funktionsweise von Computern und Programmen beschäftigen. Daher wollte man einen Computer entwickeln, mit dem die Jugendlichen experimentieren können.<sup>2,3</sup>

Inzwischen wurden 3,8 Millionen Stück verkauft (Stand Oktober 2014<sup>4</sup>) und 5 verschiedene Modelle entwickelt.

### 2.1.2. Technische Daten

Die Technik in einem Raspberry Pi ist vergleichbar mit der eines Smartphones. Der Raspberry Pi hat eine **CPU** mit **700 MHz**, welche auf bis zu **1 GHz** Übertaktbar ist, und je nach Modell 256 oder 512 MB Arbeitsspeicher. Als Speicher für das Betriebssystem (verschiedene Linux-Distributionen stehen zur Auswahl) wird eine SD-Karte bzw. eine microSD-Karte verwendet.

Zur Stromversorgung genügt ein normales Handyladegerät mit Micro-USB-Anschluss und mindestens 1 **Ampere** Stromstärke, denn der Raspberry Pi verbraucht nur 3.5 Watt<sup>5</sup> (Modell B).

Zum Anschließen anderer Hardware gibt es zwei USB-Anschlüsse und 26 **GPIO-Pins**.

---

<sup>2</sup>Raspberry Pi Foundation, *The Making of Pi*.

<sup>3</sup>Wikipedia, *Raspberry Pi— Wikipedia, Die freie Enzyklopädie*, Geschichte.

<sup>4</sup>@Raspberry\_Pi, @ruskin147 As of today, it looks like 3.8 million - that's an \*awful lot of computers\*.

<sup>5</sup>elinux, *RPi Hardware - Power*.

## 2. Hardware

### 2.2. Sensoren

Zur Messung der Umweltdaten werden folgende Sensoren verwendet:

- 4 Temperatursensoren *DS18B20* ([2.2.1](#))
- Luftfeuchtesensor *DHT22* ([2.2.2](#))
- Luftdrucksensor *BMP085* ([2.2.3](#))
- Luftqualitätssensor *VOLTCRAFT CO-20* ([2.2.4](#))
- CPU-Temperatur des Raspberry Pi

#### 2.2.1. Temperatur

Mithilfe von 4 Sensoren des Typ *DS18B20* werden die Innentemperatur, die Gehäusetemperatur und die Bodentemperatur (Außen) gemessen. Diese haben eine Messgenauigkeit von  $\pm 0.5 \text{ }^{\circ}\text{C}$  und einen Messbereich von  $-10 \text{ }^{\circ}\text{C}$  bis  $85 \text{ }^{\circ}\text{C}$ .<sup>6</sup>

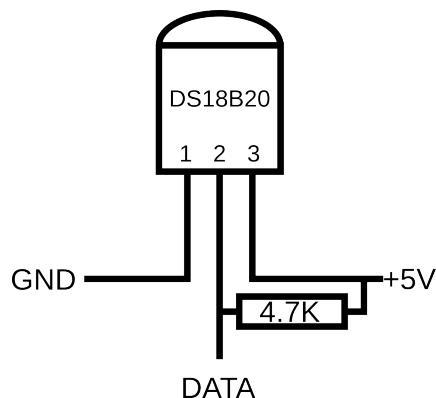


Abbildung 2.2.: Pinbelegung des DS18B20 (eigenes Werk)

Der Sensor wird mithilfe von einem 1-Wire-Bus ausgelesen. Hierbei benötigt man (außer für die Stromversorgung mit 5 Volt nur ein Kabel, auf dem die Daten übertragen werden.<sup>7</sup> Zusätzlich wird ein  $4.7 \text{ k}\Omega$  Widerstand zwischen dem Pin für Daten und dem Pin für +5 V benötigt. (siehe Abb. 2.2) Ein

<sup>6</sup>Maxim Integrated Products, *DS18B20 - Data Sheet*, S. 20.

<sup>7</sup>FHEMWiki, *Kategorie:1-Wire - FHEMWiki*.

## 2. Hardware

weiterer Vorteil von 1-Wire ist, dass nahezu beliebig viele Sensoren auf einem Datenkabel parallel geschaltet werden können.

Die Messdaten des *DS18B20* können auf dem Raspberry Pi sehr einfach ausgelesen werden, weil dies von einem Linux-[Kernelmodul](#) erledigt wird. Um die Temperatur zu erhalten, muss nur eine virtuelle Datei ausgelesen werden, welche das Messergebnis in tausendstel Grad Celsius enthält. (Siehe Abbildung 2.3)

```
pi@raspberry /sys/bus/w1/devices $ cat 10-00080277a5db/w1_slave 10-00080277abe1/w1_slave
2f 00 4b 46 ff ff 08 10 78 : crc=78 YES
2f 00 4b 46 ff ff 08 10 78 t=23250
2f 00 4b 46 ff ff 08 10 78 : crc=78 YES
2f 00 4b 46 ff ff 08 10 78 t=23250
pi@raspberry /sys/bus/w1/devices $ cat 10-00080277a5db/w1_slave 10-00080277abe1/w1_slave
2f 00 4b 46 ff ff 01 10 ca : crc=ca YES
2f 00 4b 46 ff ff 01 10 ca t=23687
2f 00 4b 46 ff ff 07 10 60 : crc=60 YES
2f 00 4b 46 ff ff 07 10 60 t=23312
```

Abbildung 2.3.: Die erste erfolgreiche Messung (eigenes Werk)

### 2.2.2. Luftfeuchtigkeit

Zum Messen der Luftfeuchtigkeit der Außenluft wird der *DHT22* verwendet. Dieser kann auch die Temperatur messen. Die Messgenauigkeit beträgt  $\pm 0.5\text{ }^{\circ}\text{C}$  und  $\pm 2\%$  relative Luftfeuchte.<sup>8</sup> Wie der *DS18B20* (2.2.1) benötigt der Luftfeuchtigkeitssensor zusätzlich zur Stromversorgung nur ein Kabel zur Datenübertragung. Es können jedoch nicht mehrere Sensoren parallel geschaltet werden.<sup>9</sup>

Die Daten des Sensors werden von einem C-Programm von Adafruit ausgelesen.<sup>10</sup>

---

<sup>8</sup>Aosong Electronics Co.,Ltd, [Digital-output relative humidity & temperature sensor/module DHT22](#).

<sup>9</sup>Adafruit User LADY ADA, [DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging](#), Wiring.

<sup>10</sup>Ebd., Software Install.

## 2. Hardware

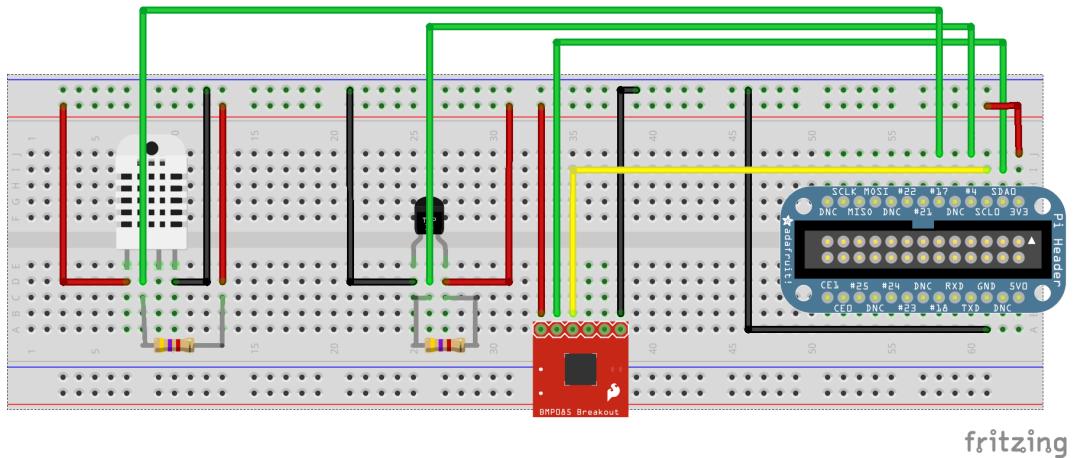


Abbildung 2.4.: Anschlusskitze von *DS18B20* (Mitte; 2.2.1), *DHT22* (Links; 2.2.2) und *BMP085* (Rechts; 2.2.3) (eigenes Werk)

### 2.2.3. Luftdruck

Der *BMP085* ist der präziseste Sensor. Er wird zum Messen des Luftdruckes und der Außentemperatur verwendet und hat dabei eine Genauigkeit von  $\pm 1.0 \text{ hPa}$  und  $0.5^\circ\text{C}$  bei  $25^\circ\text{C}$ <sup>11</sup>

Die Messdaten überträgt der Sensor über einen  $\text{I}^2\text{C}$ -Bus. Dabei werden (zusätzlich zur Stromversorgung) **zwei** Kabel zur Datenübertragung benötigt. (siehe Abbildung 2.4) Zum einen ist das das gelbe Kabel, über welches der Raspberry Pi dem Sensor die Taktfrequenz schickt, in der er die Daten übertragen soll, und zum anderen das grüne Kabel, über das die eigentlichen Daten übertragen werden.<sup>12</sup>

Auch hier werden die Daten von einem Programm von Adafruit ausgelesen.<sup>13</sup>

<sup>11</sup> Bosch Sensortec, *BMP085 Digital pressure Sensor - Data Sheet*, S. 6.

<sup>12</sup> Adafruit User KEVIN TOWNSEND, *Using the BMP085/180 with Raspberry Pi or Beaglebone Black*, Hooking Everything Up.

<sup>13</sup> Ebd., Using the Adafruit BMP Python Library (Updated).

## 2. Hardware

### 2.2.4. Luftqualität

Der letzte Sensor, der hinzugekommen ist, ist der VOLTCRAFT CO-20. Da CO<sub>2</sub>-Sensoren und andere genaue Luftqualitätssensoren teuer sind, habe ich mich für einen einfachen VOC-Sensor entschieden. Dieser misst die Menge an *Flüchtigen organischen Verbindungen* in der Luft. Dies sind Stoffe, die schon bei niedrigen Temperaturen verdampfen. Sie können von verschiedenen Quellen stammen (z. B.: Benzindämpfe, Tabakrauch, Lacke)<sup>14</sup> und von leichten Kopfschmerzen und Konzentrationsstörungen bis zu bleibenden Gesundheitsschäden führen.<sup>15</sup>

Der Sensor gibt einen Wert an, der die relative Verschlechterung seit dem Einschalten angibt. Hierbei steht 450 für die anfängliche Qualität und ein höherer Wert für eine schlechtere Luftqualität. Da der VOLTCRAFT CO-20 jedoch nicht mehr erhältlich ist, verwende ich den baugleichen Raumluftfühler von Velux.<sup>16</sup>



Abbildung 2.5.: Velux Raumluftfühler (eigenes Werk)

Der Sensor wird über USB an den Raspberry Pi angeschlossen. Um die Daten unter Linux auszulesen, wird das Programm *usb-sensors-linux*

<sup>14</sup>Innenraumlufthygiene-Kommission des Umweltbundesamtes, [LEITFADEN FÜR DIE INNENRAUMHYGIENE IN SCHULGEBÄUDEN](#), S. 41 ff.

<sup>15</sup>WISSEN Wiki, [Flüchtige organische Verbindung](#), Gesundheitliche Wirkung.

<sup>16</sup>Velux, [VELUX Raumluftfühler](#).

## 2. Hardware

verwendet.<sup>17</sup>

### 2.3. Display

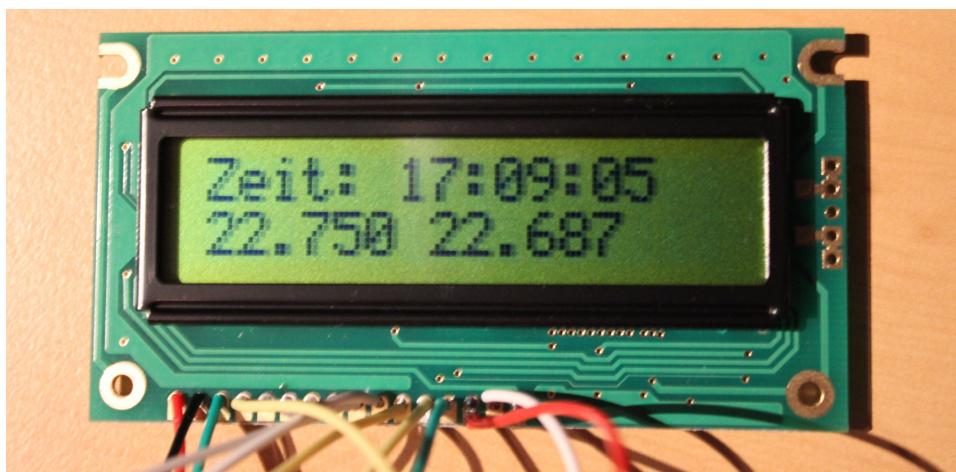


Abbildung 2.6.: Erstes Display (eigenes Werk)

Damit nicht immer ein Computer benötigt wird, um die aktuellen Messwerte zu erfahren, verwende ich ein Display, welches diese anzeigt. Ursprünglich habe ich ein 16x2 Zeichen Display von *Conrad Electronic* verwendet.<sup>18</sup> Dieses wird nach der Anleitung von [www.schnatterente.net](http://www.schnatterente.net)<sup>19</sup> angeschlossen.

Da jedoch fix angelötete Kabel unflexibel sind, bin ich auf ein Display von *Pollin.de*<sup>20</sup> umgestiegen, welches mit einer Steckverbindung angeschlossen wird.

### 2.4. Anschluss

---

<sup>17</sup>usb-sensors-linux, [Install AirSensor on Linux](#).

<sup>18</sup>ANAG VISION, [AV1624 Datasheet](#).

<sup>19</sup>Schnatterente.net, [Raspberry Pi: 32 Zeichen Hitachi HD44780 Display](#).

<sup>20</sup>Pollin.de, [LCD-Modul TC1602E-01](#).

Anzahl  
hinzufü-  
gen

## 3. Software

Die Software, die verwendet wird, teilt sich in (?) Teile auf:

- Auslesen der Sensoren, Aufbereiten der Daten und allgemeine Steuerung (main.sh)
- Steuern des Displays
- Endauswertung
- Webinterface
- sonstiges

### 3.1. main.sh

Das wichtigste Programm ist das Bash-Script *main.sh*. Mithilfe eines Bash-Scriptes können Programme automatisiert gestartet und ihre Ausgaben ausgewertet werden. Dieses Bash- Script kümmert sich um die Aufzeichnung und Speicherung der Daten und die Steuerung der anderen Programme.

#### 3.1.1. Allgemeines

Zunächst werden die Pins angegeben, an denen die LEDs angeschlossen sind. In den Zeilen 11-13 wird nun die grüne LED eingeschaltet, um zu zeigen, dass das Programm läuft.

```
8 gpio mode 13 out # gelb
9 gpio mode 12 out # rot
10 gpio mode 3 out #grün
11 gpio write 13 0 # nur grün einschalten
12 gpio write 12 0
```

### 3. Software

```
13 gpi o write 3 1
```

Datei 1: main.sh (Zeile 8 bis 13)

Nun startet das eigentliche Programm. Alles, was nun folgt wird wiederholt, bis die Aufzeichnung beendet wird.

```
27 while true  
28 do
```

Datei 2: main.sh (Zeile 27 bis 28)

In den folgenden drei Zeilen wird der aktuelle Zeitpunkt in drei verschiedenen Formaten für drei verschiedene Zwecke gespeichert.

Code	Beispiel	Verwendung
%Y/%m/%d %H:%M:%S	2014/11/23 16:47:50	Format zum Abspeichern in <a href="#">CSV-Datei</a>
%d.%m %H:%M:%S	23.11 16:47:50	einfach lesbares Format für Display
%d.%m.%y %H:%M:%S	23.11.2014 16:47:50	einfaches, exaktes Format für Webinterface

Tabelle 3.1.: Datumsformate

```
29 uhrzeit=$( date +%Y/%m/%d\ %H:%M:%S )  
30 uhrzeit_display=$( date +%d.%m\ %H:%M:%S )  
31 uhrzeit_lang=$( date +%d.%m.%y\ %H:%M:%S )
```

Datei 3: main.sh (Zeile 29 bis 31)

#### 3.1.2. Messung

Als erstes werden die Sensoren ausgelesen. Am einfachsten kann mit dem im Raspberry Pi integrierten Thermometer die **CPU**-Temperatur ausgelesen werden:

### 3. Software

```
32 rasp=$(/opt/vc/bin/vcgencmd measure_temp | cut -c  
       6,7,8,9)
```

Datei 4: main.sh (Zeile 32)

Nur wenig aufwändiger sind die Temperatursensoren (*DS18B20*, siehe [2.2.1](#)). Da die Sensoren manchmal ungültige Werte zurückgeben, wird nach der ersten Messung überprüft, ob dies der Fall ist (Zeile 34) und die Messung solange wiederholt, bis eine gültige Messung erfolgt.

```
33 temp1=$(echo "scale=3; $(grep 't=' /sys/bus/w1/devices/  
      w1_bus_master1/10-000802b53835/w1_slave | awk 'F  
      ='{print $2}')/1000" | bc -l)  
34 while [ "$temp1" == "-1.250" ] || [ "$temp1" == "85.000"  
      ]  
35 do  
36   gpiowrite 13 1  
37   echo "Temp1: $temp1"  
38   temp1=$(echo "scale=3; $(grep 't=' /sys/bus/w1/  
      devices/w1_bus_master1/10-00080277abe1/w1_slave |  
      awk 'F={print $2}')/1000" | bc -l)  
39   gpiowrite 13 0  
40 done
```

Datei 5: main.sh (Zeile 33 bis 40)

Die Adafruit-Programme<sup>[2122](#)</sup>, die den Luftfeuchtesensor (siehe [2.2.2](#)) und den Luftdrucksensor (siehe [2.2.3](#)) auslesen, geben die Feuchtigkeit und die Temperatur durch einen Strichpunkt getrennt an. Daher wird dies zu Beginn als Trennzeichen angegeben.

```
4 IFS=";"
```

Datei 6: main.sh (Zeile 4)

<sup>21</sup>Adafruit User LADY ADA, *DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging*.

<sup>22</sup>Adafruit User KEVIN TOWNSEND, *Using the BMP085/180 with Raspberry Pi or Beaglebone Black*.

### 3. Software

Dadurch kann die Ausgabe einfach aufgetrennt werden:

```
66 luft_roh=$(sudo python /home/pi/Temperaturmessung/
    Fremddateien/AdafruitDHT.py 2302 17)
67 set -- $luft_roh
68 luft_temp=$1
69 luft_feucht=$2
```

Datei 7: main.sh (Zeile 66 bis 69)

Auch hier wird bei ungültigen Messwerten mehrmals gemessen.

*usb-sensors-linux*<sup>23</sup> gibt direkt den relativen Wert für die Luftqualität zurück, der nicht weiterbearbeitet werden muss.

```
84 qualitat=$(sudo /home/pi/Temperaturmessung/Fremddateien/
    /airsensor -v -o)
```

Datei 8: main.sh (Zeile 84)

#### 3.1.3. Speichern, Aufbereiten und Verarbeiten

Nachdem alle Sensoren ausgelesen und die Messwerte in Variablen gespeichert wurden, müssen sie dauerhaft gespeichert werden. Hierzu werden alle Werte durch ein Komma getrennt und als neue Zeile an die bisherigen Messungen angehängt.

```
89 ausgabe=${uhrzeit}\ ${temp1}\ ${temp2}\ ${temp3}\ ${temp4}\ ${luft_temp}\ ${luft_feucht}\ ${druck}\ ${temp_druck}\ ${rasp}\ ${qualitat}
90 echo $ausgabe >>/home/pi/Temperaturmessung/dygraphs.csv
```

Datei 9: main.sh (Zeile 89 bis 90)

Hierdurch entsteht eine [CSV-Datei](#)-Datei die wie folgt aussehen kann.

---

<sup>23</sup>[usb-sensors-linux, Install AirSensor on Linux.](#)

### 3. Software

```
1 2014/10/03 12:47:36,27.562,29.437,17.375,29.437,19.1,71.4,1000.95,19.30,53.0,1181
2 2014/10/03 12:48:07,27.625,29.437,17.375,29.437,19.1,71.4,1000.86,19.30,53.0,1140
3 2014/10/03 12:48:34,27.625,29.437,17.437,29.500,19.2,71.5,1001.00,19.40,53.0,1151
4 2014/10/03 12:49:02,27.625,29.500,17.437,29.500,19.2,71.5,1000.85,19.40,53.0,1147
```

Datei 10: dygraphs.csv

Diese Datei wird auch in den Ordner des Webservers kopiert, damit es grafisch dargestellt werden kann (siehe [3.3.2](#)). Weiters verwendet die *Endauswertung* (siehe [3.4](#)) auch diese Datei zur rechnerischen Auswertung.

Als nächstes wird der Text für das Display (siehe [2.3](#)) erzeugt. Da dort der Platz beschränkt ist (16x2 Zeichen), werden alle Messwerte um 3 Stellen (bzw. 2 bei Luftdruck) gekürzt. Anschließend werden diese Daten in text.txt (für Display) und text\_ws.txt (für Webinterface) exportiert.

```
92 temp1_r=$( echo $temp1 | rev | cut -c 3- | rev )
```

Datei 11: main.sh (Zeile 92)

```
1 03.10 14:49:02
2 27.3
3 30.6
4 25.7
5 30.6
6 25
7 50
8 25.7
9 989.5
10 59.5
11 450
```

Datei 12: text.txt

```
1 03.10.14 14:49:02,27.3,30.6,25.7,30.6,25,50,25.7,989.5,59.5,450
```

Datei 13: text\_ws.txt

Abschließend wird noch 8 Sekunden gewartet und jedes tausende Mal ein Backup gemacht und mir per E-Mail geschickt, bevor die nächste Messung von vorne beginnt.

### 3. Software

## 3.2. Display

Um die aktuellen Messungen auch ohne Computer zu sehen, werden sie auch direkt am Raspberry Pi auf einem Display angezeigt (siehe auch [2.3](#)). Um das Display anzusteuern wird ein Programm<sup>24</sup> von [www.schnatterente.net](http://www.schnatterente.net) verwendet. Dieses wurde von mir um einige Funktionen ergänzt.

Das Programm liest aus *text.txt* (siehe Datei [12](#)) die aktuellen Messwerte aus. Da der Platz jedoch stark beschränkt ist, werden diese auf 11 Seiten aufgeteilt, zwischen denen das Display alle 3 Sekunden wechselt.



Abbildung 3.1.: eingebautes Display (eigenes Werk)

## 3.3. Webinterface

Einer der wichtigsten Teile des Projektes ist die grafische Auswertung. Diese kann live auf der Webseite des Raspberry Pi und zeitverzögert unter [winkler.kremszeile.at](http://winkler.kremszeile.at) angesehen werden. Die Auswertung besteht aus zwei von einander unabhängigen Teilen. Zum einen gibt es die Anzeige der Live-Daten, zum anderen die Darstellung der kompletten Aufzeichnung als interaktives Diagramm.

---

<sup>24</sup>Schnatterente.net, [displaytest.py](http://displaytest.py).

### 3. Software

#### 3.3.1. Livedaten

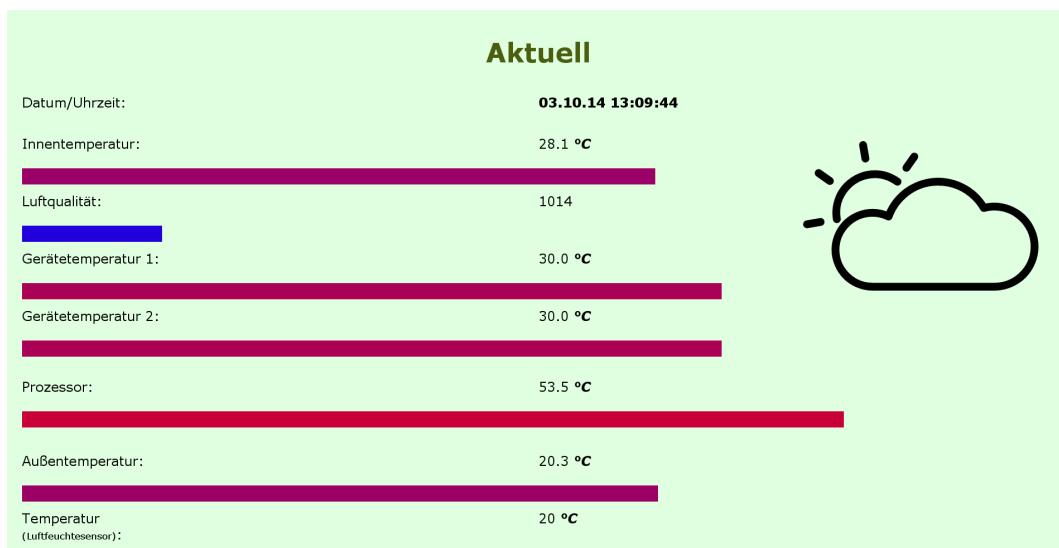


Abbildung 3.2.: Livedaten (Screenshot von [winkler.kremszeile.at/aktuell.html](http://winkler.kremszeile.at/aktuell.html))

Auf der Webseite des Raspberry Pi können die aktuellen Messwerte angezeigt und grafisch veranschaulicht werden. Hierzu wird alle 5 Sekunden mithilfe von [JavaScript](#) die Datei *text\_ws.txt* nachgeladen und ausgewertet. Zusätzlich zur Anzeige der Zahlenwerte werden die Messungen mithilfe von Balken und Farbverläufen angezeigt. Für ältere Browser gibt es auch eine einfache tabellarische Ansicht<sup>25</sup>

#### Wetterrater

Um die Daten auch anders zu nutzen habe ich einen *Wetterrater* programmiert. Dieser versucht auf Basis von einfachen Berechnungen und Schätzungen das aktuelle Wetter zu „erraten“. So wird zum Beispiel die aktuelle Außentemperatur mit der nach Jahreszeit und Tageszeit zu erwartenden Temperatur verglichen, um Rückschlüsse auf den Bewölkungsgrad zu ziehen oder aufgrund der Luftfeuchtigkeit ermittelt, ob es Niederschlag gibt.

<sup>25</sup>[winkler.kremszeile.at/aktuell\\_einfach.html](http://winkler.kremszeile.at/aktuell_einfach.html)

### 3. Software

Hierzu wird zum Beispiel die Temperaturschwankung über einen Tag als Cosinusfunktion mit einer Schwankung von 5 °C angenommen.

$$\text{Schätzung} = -5 \cdot \cos\left(\frac{\text{Stunde} \cdot 2 \cdot \pi}{24}\right) + \text{Mittlere Temperatur des Tages};$$

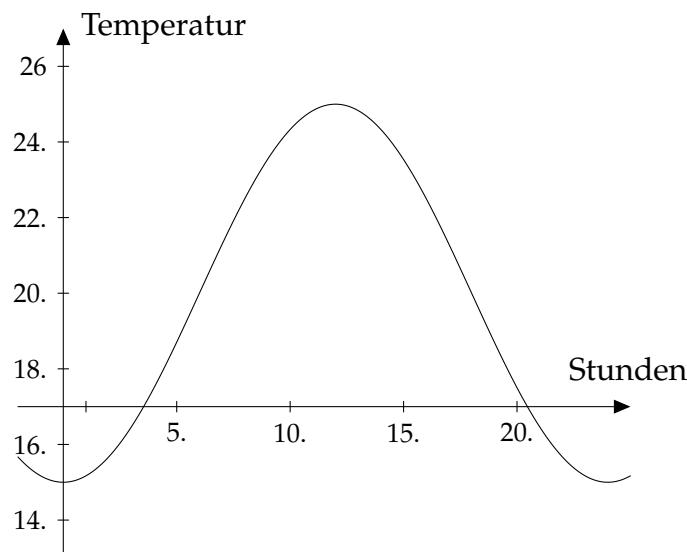


Abbildung 3.3.: Graph der Funktion (eigenes Werk)

Der Wetterräte gibt drei Werte aus:

- Tag oder Nacht
- Niederschlag (keiner/Gewitter/Schnee/Regen/Schneeregen)
- Bewölkungsgrad (sonnig/leicht bewölkt/stark bewölkt)

Aufgrund von diesen Werten wird dann eine Grafik (siehe Abbildung 3.4) ausgewählt, welche dann angezeigt wird.

#### 3.3.2. Diagramme

Damit auch die vergangenen Messergebnisse angesehen werden können, werden diese im Webinterface auf einer eigenen Seite als Diagramm

### 3. Software



Abbildung 3.4.: Wettericons (eigenes Werk)

dargestellt. Diese werden mithilfe von *dygraphs*<sup>26</sup>, einer **JavaScript**-Bibliothek für interaktive Diagramme, erstellt.<sup>27</sup>

Hierzu lädt der Webbrower die komplette Aufzeichnung in Form der Datei *dygraphs.csv* (siehe Datei 10) nach. Daraus wird ein Diagramm über den gesamten Zeitraum der Messung erstellt. Im Gegensatz zu anderen Darstellungen kann hier jedoch einfach hineingezoomt werden. So können einfach einzelne Wochen oder Tage betrachtet werden. Da das Diagramm mit allen 10 Messkurven auf einmal überladen wäre, können einzelne Kurven aktiviert und deaktiviert werden. Zusätzlich kann ein Faktor eingegeben werden, um den die Kurven automatisch geglättet werden sollen. Mit einem Tastendruck kann man auch auf die letzten 24 Stunden oder 7 Tage zoomen oder in einem Kalender den Zeitraum auswählen. Es gibt auch ein Auswahlmenü, um den gewünschten Datensatz auszuwählen. Das Diagramm ist auch darauf angepasst, auch mit Tablets bzw. Touchscreens bedient zu werden. Auf Youtube gibt es ein Video, wo man die Verwendung in Aktion sehen kann: [www.youtube.com/watch?v=1bv6CEXuN5c](http://www.youtube.com/watch?v=1bv6CEXuN5c)

---

<sup>26</sup>dygraphs, *Homepage*.

<sup>27</sup>[winkler.kremszeile.at/dygraphs\\_aussen.html](http://winkler.kremszeile.at/dygraphs_aussen.html)

### 3. Software

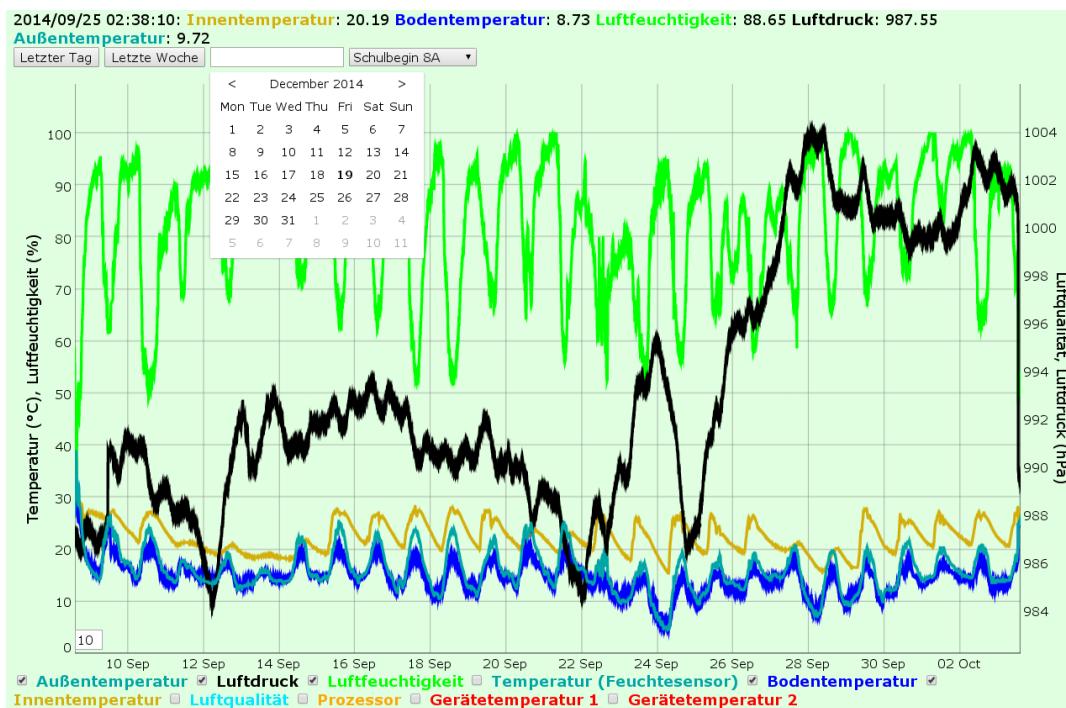


Abbildung 3.5.: Diagramm (eigenes Werk)

## 3.4. Endauswertung

## 3.5. diverses

### 3.5.1. autostart

### 3.5.2. mitternacht.sh

eventuell  
entfernen

### 3.5.3. Weather Underground

### 3. Software



Abbildung 3.6.: Diagramm am Tablet (eigenes Werk)

# Anhang A.

## Weitere Informationen

In dieser VWA können nur die wichtigsten Teile des Projektes erwähnt werden. Weitere Informationen gibt es unter folgenden Quellen:

- Github: <https://github.com/Findus23/Umweltdatenmessung>
  - Der komplette Programmcode und alle anderen Dateien, die entstanden sind, sind hier gesammelt.
  - Auch alle Veränderungen seit Dezember 2013 können hier angesehen werden: <https://github.com/Findus23/Umweltdatenmessung/commits/master>
  - Große Veränderungen werden zusätzlich separat gelistet: <https://github.com/Findus23/Umweltdatenmessung/releases>
- Flickr <https://www.flickr.com/photos/findus23/sets/72157637721138445/>
  - Hier sind über 100 Bilder vom Projekt zu sehen.
- Youtube: [https://www.youtube.com/playlist?list=PLjtFdocVknd4aw90\\_zVr0U4BF1RH9PatA](https://www.youtube.com/playlist?list=PLjtFdocVknd4aw90_zVr0U4BF1RH9PatA)
  - Einige Videos (z.B. vom Display) sind auf Youtube zu finden.

# Literatur

## Online-Literatur

- Adafruit User KEVIN TOWNSEND. *Using the BMP085/180 with Raspberry Pi or Beaglebone Black.* 2013. URL: <https://learn.adafruit.com/using-the-bmp085-with-raspberry-pi?view=all> (besucht am 25.10.2014) (siehe S. 6, 11).
- Adafruit User LADY ADA. *DHT Humidity Sensing on Raspberry Pi or Beaglebone Black with GDocs Logging.* 2013. URL: <https://learn.adafruit.com/dht-humidity-sensing-on-raspberry-pi-with-gdocs-logging?view=all> (besucht am 25.10.2014) (siehe S. 5, 11).
- ANAG VISION. *AV1624 Datasheet.* 22. Mai 2006. URL: [http://conrad.ru/doci/tekstovyy\\_display\\_stn\\_anag\\_vision\\_av1624gfbw\\_sj\\_\\_seryy\\_181664\\_en.pdf](http://conrad.ru/doci/tekstovyy_display_stn_anag_vision_av1624gfbw_sj__seryy_181664_en.pdf) (besucht am 23.11.2014) (siehe S. 8).
- Aosong Electronics Co.,Ltd. *Digital-output relative humidity & temperature sensor/module DHT22.* 2011. URL: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf> (besucht am 08.11.2014) (siehe S. 5).
- Bohk, Philipp. *Rev. 2 des Raspberry Pi Model B - made in UK.* 2012. URL: <https://commons.wikimedia.org/wiki/File:LambdaPlaques.jpg> (besucht am 04.07.2014) (siehe S. 2).
- Bosch Sensortec. *BMP085 Digital pressure Sensor - Data Sheet.* 15. Okt. 2009. URL: <https://www.sparkfun.com/datasheets/Components/General/BST-BMP085-DS000-05.pdf> (besucht am 25.10.2014) (siehe S. 6).
- dygraphs. *Homepage.* 2014. URL: <http://dygraphs.com/> (besucht am 19.12.2014) (siehe S. 17).
- elinux. *RPi Hardware - Power.* 2014. URL: [http://elinux.org/index.php?title=RPi\\_Hardware&oldid=341192#Power](http://elinux.org/index.php?title=RPi_Hardware&oldid=341192#Power) (besucht am 04.07.2014) (siehe S. 3).

## Literatur

- FHEMWiki. *Kategorie:1-Wire - FHEMWiki*. 2014. URL: <http://www.fhemwiki.de/w/index.php?title=Kategorie:1-Wire&oldid=5092#1-Wire> (besucht am 18. 10. 2014) (siehe S. 4).
- Innenraumlufthygiene-Kommission des Umweltbundesamtes. *LEITFADEN FÜR DIE INNENRAUMHYGIENE IN SCHULGEBÄUDE*. 2008. URL: <http://www.umweltbundesamt.de/sites/default/files/medien/publikation/long/3689.pdf> (besucht am 11. 11. 2014) (siehe S. 7).
- Maxim Integrated Products, Inc. *DS18B20 - Data Sheet*. 2008. URL: <http://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> (besucht am 17. 10. 2014) (siehe S. 4).
- Pollin.de. *LCD-Modul TC1602E-01*. 2014. URL: [http://www.pollin.de/shop/dt/0Tc10Tc40Tk-/Bauelemente\\_Bauteile/Aktive\\_Bauelemente/Displays/LCD\\_Modul\\_TC1602E\\_01.html](http://www.pollin.de/shop/dt/0Tc10Tc40Tk-/Bauelemente_Bauteile/Aktive_Bauelemente/Displays/LCD_Modul_TC1602E_01.html) (besucht am 23. 11. 2014) (siehe S. 8).
- Raspberry Pi Foundation. *The Making of Pi*. Raspberry Pi Foundation. 2012. URL: <http://www.raspberrypi.org/about/> (besucht am 04. 07. 2014) (siehe S. 3).
- @Raspberry\_Pi. @ruskin147 *As of today, it looks like 3.8 million - that's an \*awful lot of computers\**. 12. Okt. 2014. URL: [https://twitter.com/Raspberry\\_Pi/status/521065388948586497](https://twitter.com/Raspberry_Pi/status/521065388948586497) (besucht am 22. 11. 2014) (siehe S. 3).
- Schnatterente.net. *displaytest.py*. URL: <http://www.schnatterente.net/code/raspberrypi/displaytest.py> (besucht am 13. 12. 2014) (siehe S. 14).
- *Raspberry Pi: 32 Zeichen Hitachi HD44780 Display*. 10. Okt. 2014. URL: <http://www.schnatterente.net/technik/raspberry-pi-32-zeichen-hitachi-hd44780-display#> (besucht am 23. 11. 2014) (siehe S. 8).
- usb-sensors-linux. *Install AirSensor on Linux*. 29. Apr. 2013. URL: [https://code.google.com/p/usb-sensors-linux/wiki/Install\\_AirSensor\\_Linux](https://code.google.com/p/usb-sensors-linux/wiki/Install_AirSensor_Linux) (besucht am 08. 11. 2014) (siehe S. 8, 12).
- Velux. *VELUX Raumluftfühler*. 2014. URL: [http://www.velux.de/privatkunden/produkte/integra\\_system/produkte/produktempfehlung/raumluftfuehler](http://www.velux.de/privatkunden/produkte/integra_system/produkte/produktempfehlung/raumluftfuehler) (besucht am 08. 11. 2014) (siehe S. 7).
- Wikipedia. *Bus— Wikipedia, Die freie Enzyklopädie*. 2014. URL: [http://de.wikipedia.org/w/index.php?title=Bus\\_\(Datenverarbeitung\)&oldid=134938136](http://de.wikipedia.org/w/index.php?title=Bus_(Datenverarbeitung)&oldid=134938136) (besucht am 04. 07. 2014) (siehe S. 26).
- *GitHub — Wikipedia, Die freie Enzyklopädie*. 12. Dez. 2014. URL: <http://de.wikipedia.org/w/index.php?title=GitHub&oldid=136725990> (besucht am 14. 12. 2014) (siehe S. 27).

## Literatur

- Wikipedia. *Raspberry Pi*— Wikipedia, Die freie Enzyklopädie. 2014. URL: [http://de.wikipedia.org/w/index.php?title=Raspberry\\_Pi&oldid=134104012#Idee](http://de.wikipedia.org/w/index.php?title=Raspberry_Pi&oldid=134104012#Idee) (besucht am 04.07.2014) (siehe S. 3).
- WISSEN Wiki. *Flüchtige organische Verbindung*. 2014. URL: [http://www.wissenwiki.de/index.php?title=F1%C3%83%C2%BCchtige\\_organische\\_Verbindung&oldid=41478](http://www.wissenwiki.de/index.php?title=F1%C3%83%C2%BCchtige_organische_Verbindung&oldid=41478) (besucht am 11.11.2014) (siehe S. 7).

# Abbildungsverzeichnis

1.1.	Messstation (eigenes Werk) . . . . .	1
2.1.	Raspberry Pi - Modell B . . . . .	2
2.2.	Pinbelegung des DS18B20 (eigenes Werk) . . . . .	4
2.3.	Die erste erfolgreiche Messung (eigenes Werk) . . . . .	5
2.4.	Anschlusskitze von <i>DS18B20</i> (Mitte; 2.2.1), <i>DHT22</i> (Links; 2.2.2) und <i>BMP085</i> (Rechts; 2.2.3) (eigenes Werk) . . . . .	6
2.5.	Velux Raumluftfühler (eigenes Werk) . . . . .	7
2.6.	Erstes Display (eigenes Werk) . . . . .	8
3.1.	eingebautes Display (eigenes Werk) . . . . .	14
3.2.	Livedaten (Screenshot von <a href="http://winkler.kremszeile.at/aktuell.html">winkler.kremszeile.at/aktuell.html</a> )	15
3.3.	Graph der Funktion (eigenes Werk) . . . . .	16
3.4.	Wettericons (eigenes Werk) . . . . .	17
3.5.	Diagramm (eigenes Werk) . . . . .	18
3.6.	Diagramm am Tablet (eigenes Werk) . . . . .	19

# Dateiverzeichnis

1.	main.sh (Zeile 8 bis 13) . . . . .	9
2.	main.sh (Zeile 27 bis 28) . . . . .	10
3.	main.sh (Zeile 29 bis 31) . . . . .	10
4.	main.sh (Zeile 32) . . . . .	11
5.	main.sh (Zeile 33 bis 40) . . . . .	11
6.	main.sh (Zeile 4) . . . . .	11
7.	main.sh (Zeile 66 bis 69) . . . . .	12
8.	main.sh (Zeile 84) . . . . .	12
9.	main.sh (Zeile 89 bis 90) . . . . .	12
10.	dygraphs.csv . . . . .	13
11.	main.sh (Zeile 92) . . . . .	13
12.	text.txt . . . . .	13
13.	text_ws.txt . . . . .	13

# Glossar

[A](#) | [C](#) | [D](#) | [F](#) | [G](#) | [H](#) | [I](#) | [J](#) | [K](#) | [V](#)

**A**

**Ampere**

die SI-Basiseinheit der elektrischen Stromstärke [3](#), [25](#)

**C**

**C**

eine sehr weit verbreitete Programmiersprache

Hier wird sie oft zum Auslesen der Sensoren verwendet, da sie sehr schnell ausgeführt wird [5](#), [25](#)

**CPU**

Central Processing Unit [3](#), [10](#), [25](#)

**CSV-Datei**

*Comma-separated values*

Hierbei werden Messungen in einer Textdatei durch Zeilenumbrüche und einzelne Werte durch Beistriche getrennt [1](#), [10](#), [12](#), [25](#)

**D**

**Datenbus**

*ein System zur Datenübertragung zwischen mehreren Teilnehmern über einen gemeinsamen Übertragungsweg, bei dem die Teilnehmer nicht an der Datenübertragung zwischen anderen Teilnehmern beteiligt sind.*<sup>28</sup> [25](#), [27](#)

**F**

**Flickr**

ist eine Online-Plattform, auf der Fotos hochgeladen und veröffentlicht werden können [20](#), [25](#)

**G**

---

<sup>28</sup>Wikipedia, *Bus*— Wikipedia, Die freie Enzyklopädie.

## Glossar

### **Github**

ist ein webbasierter Hosting-Dienst für Software-Entwicklungsprojekte<sup>29</sup> 20,  
25

### **GPIO**

General Purpose Input/Output  
Kontakte, die Softwareseitig für verschiedene Zwecke angesteuert  
werden können  
z. B.: Auslesen von Sensoren, Ansteuern von Displays 3, 25

### **H**

#### **Hertz**

die SI-Basiseinheit für die Frequenz  
Sie gibt die Wiederholungen pro Sekunden an (hier: Schwingungen  
pro Sekunde 3, 25)

### **I**

#### **I<sup>2</sup>C**

*Inter-Integrated Circuit* (auf Deutsch gesprochen: *I-Quadrat-C*)  
ein sehr weit verbreiteter [Datenbus](#) 6, 25

### **J**

#### **JavaScript**

eine Skriptsprache für dynamische Inhalte in Webseiten 15, 17, 25

### **K**

#### **Kernelmodul**

ein Programm, welches in das Betriebssystem geladen werden kann  
und oft zur Unterstützung von Hardware verwendet wird 5, 25

### **V**

#### **VOC**

volatile organic compound (dt. Flüchtige organische Verbindungen) 7,  
25

#### **Volt**

die SI-Basiseinheit der elektrischen Spannung 4, 25

---

<sup>29</sup>Wikipedia, [GitHub — Wikipedia, Die freie Enzyklopädie](#).

# Eidesstattliche Erklärung

Ich, Lukas Winkler, erkläre hiermit eidesstattlich, dass ich diese vorwissenschaftliche Arbeit selbstständig und ohne Hilfe Dritter verfasst habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als Zitate kenntlich gemacht und alle verwendeten Quellen angegeben habe.

Krems an der Donau, am \_\_\_\_\_

Datum

Unterschrift