

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 pdfL^AT_EX und Biber gesetzt.

Die L^AT_EX 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

Abstract	iv
1 Einleitung	1
2 Hardware	2
2.1 Der Raspberry Pi	2
2.1.1 Geschichte	3
2.1.2 Technische Daten	3
2.2 Sensoren	3
2.2.1 Temperatur	4
2.2.2 Luftfeuchtigkeit	5
2.2.3 Luftdruck	5
2.2.4 Luftqualität	6
2.3 Display	7
2.4 Anschluss	8
3 Software	10
3.1 main.sh	10
Literatur	15
Abbildungsverzeichnis	17
Programmcode	18
Glossar	19

Todo list

genauere Beschreibung des Projekts	1
Einleitung	2
evtl Bild mit Steckverbindung/löschen/zu Software verschieben . . .	8

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 Beschreibung des Projekts

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.

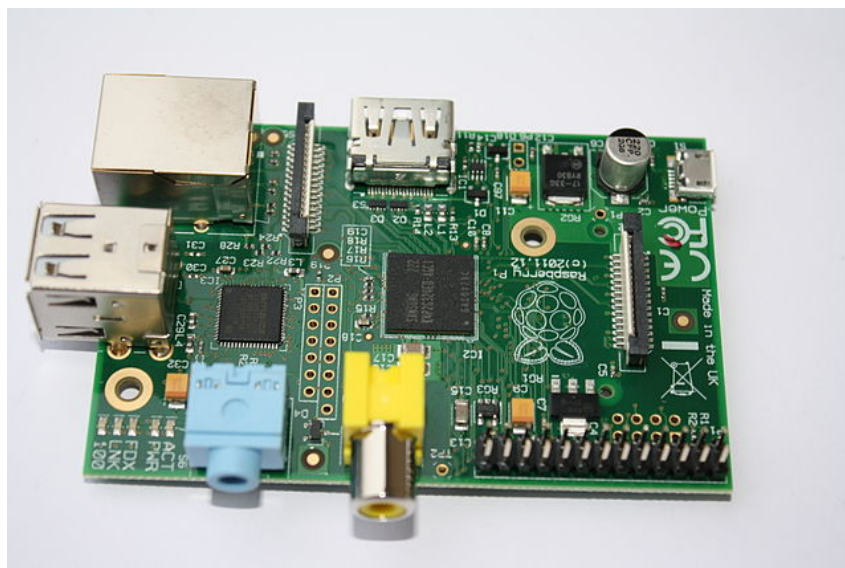


Abbildung 2.1: Raspberry Pi - Modell B¹

¹Bohk, 2012

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.^{2,3}

Inzwischen wurden 3,8 Millionen Stück verkauft (Stand Oktober 2014⁴) 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 Watt⁵ (Modell B).

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

2.2 Sensoren

Zur Messung der Umweltdaten werden folgende Sensoren verwendet:

²Raspberry Pi Foundation, 2012.

³Wikipedia, 2014b, Geschichte.

⁴@Raspberry_Pi, 2014.

⁵elinux, 2014.

2 Hardware

- 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 der Temperatursensoren werden die Innentemperatur, die Gehäuse-temperatur und die Bodentemperatur (Außen) gemessen. Diese haben eine Messgenauigkeit von $\pm 0.5^\circ\text{C}$ und einen Messbereich von -10°C bis 85°C .⁶

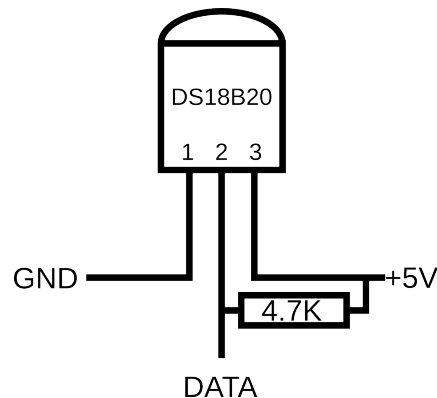


Abbildung 2.2: Pinbelegung des DS18B20 (eigenes Werk)

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.⁷ 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 weiterer Vorteil von 1-Wire ist, dass nahezu beliebig viele Sensoren auf einem Datenkabel parallel geschaltet werden können.

⁶Maxim Integrated Products, 2008, S. 20.

⁷FHEMWiki, 2014.

2 Hardware

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@raspberrypi /sys/bus/w1/devices $ cat 10-00080277a5db/w1_slave 10-00080277a5db/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@raspberrypi /sys/bus/w1/devices $ cat 10-00080277a5db/w1_slave 10-00080277a5db/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 $\pm 0.5^\circ\text{C}$ und $\pm 2\%$ relative Luftfeuchte.⁸ 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.⁹

Die Daten des Sensors werden von einem C-Programm von Adafruit ausgelesen.¹⁰

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°C bei 25°C ¹¹

⁸Aosong Electronics Co.,Ltd, [2011](#).

⁹Adafruit User LADY ADA, [2013](#), Wiring.

¹⁰[Ebd.](#), Software Install.

¹¹Bosch Sensortec, [2009](#), S. 6.

2 Hardware

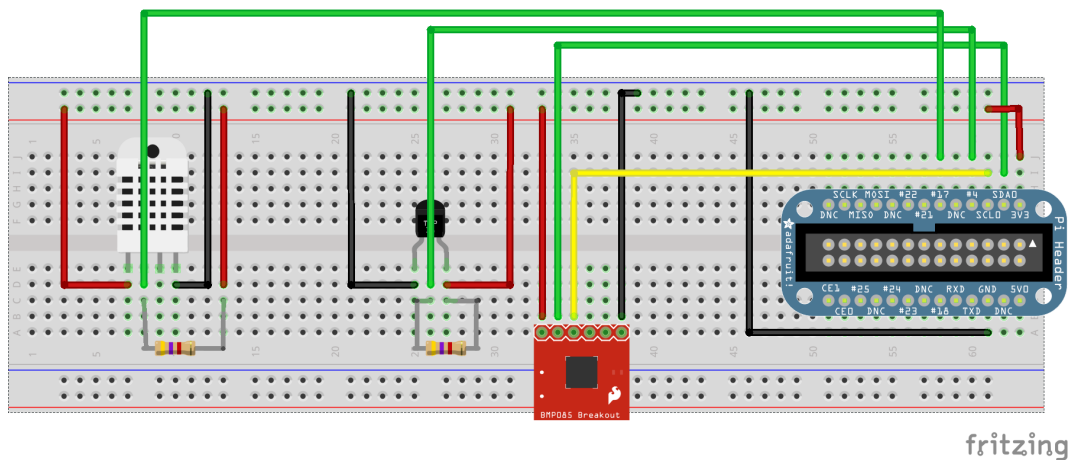


Abbildung 2.4: Anschlusskitze von *DS18B20* (Mitte; 2.2.1), *DHT22* (Links; 2.2.2) und *BMP085* (Rechts; 2.2.3) (eigenes Werk)

Die Messdaten überträgt der Sensor über einen I^2C -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.¹²

Auch hier werden die Daten von einem Programm von Adafruit ausgelesen.¹³

2.2.4 Luftqualität

Der letzte Sensor, der hinzugekommen ist, ist der *VOLTCRAFT CO-20*. Da CO_2 -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 verschiedensten Quellen stammen (z.B.: Benzindämpfe, Tabakrauch,

¹²Adafruit User KEVIN TOWNSEND, 2013, Hooking Everything Up.

¹³Ebd., Using the Adafruit BMP Python Library (Updated).

2 Hardware

Lacke)¹⁴ und von leichten Kopfschmerzen und Konzentrationsstörungen bis zu bleibenden Gesundheitsschäden führen.¹⁵

Der Sensor gibt einen Wert an, der die relative Verschlechterung seit dem Einschalten angibt. Hierbei steht 450 für die anfängliche Qualität ist 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.¹⁶



Abbildung 2.5: Velux Raumluftfühler

Der Sensor wird über USB an den Raspberry Pi angeschlossen. Um die Daten unter Linux auszulesen, wird *usb-sensors-linux* verwendet.¹⁷

2.3 Display

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.¹⁸

¹⁴Innenraumlufthygiene-Kommission des Umweltbundesamtes, 2008, S. 41 ff.

¹⁵WISSEN Wiki, 2014, Gesundheitliche Wirkung.

¹⁶Velux, 2014.

¹⁷usb-sensors-linux, 2013.

¹⁸VISION, 2006.

2 Hardware

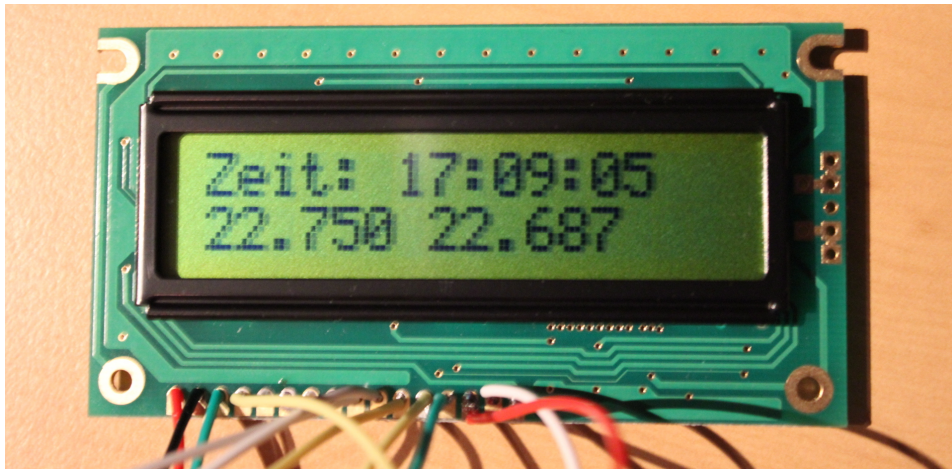


Abbildung 2.6: Erstes Display (eigenes Werk)

evtl Bild mit Steckverbindung/löschen/zu Software verschieben

Da jedoch fix angelötete Kabel unflexibel sind, bin ich auf ein Display von *Pollin.de*¹⁹ umgestiegen, welches mit einer Steckverbindung angeschlossen ist.

2.4 Anschluss

¹⁹Pollin.de, 2014.

2 Hardware



Abbildung 2.7: eingebautes Display (eigenes Werk)

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. Die Datei beginnt mit einem *Shebang* (auch *Magic Line* genannt). Diese Zeile sagt dem Betriebssystem, womit die Datei ausgeführt werden soll.

```
1 #!/bin/bash
```

Programmcode 1: *main.sh* (Zeile 1)

Die folgenden Zeilen geben allgemeine Einstellungen an und definieren später gebrauchte Variablen. Man kann den Pfad zum Webserver, auf dem das Webinterface liegt, angeben. In Zeile 6 und 7 werden die Zugangsdaten für Pushbullet aus einer anderen Datei ausgelesen. Die Zeilen 8-10 geben die Pins an, an denen die LEDs angeschlossen sind. In Zeile 11-13 wird die grüne LED eingeschaltet, um zu zeigen, dass das Programm läuft.

3 Software

```
2 PFAD="/var/www/" #Pfad zum Web-Verzeichnis
3 r=0 # Backup-Zahl auf Null setzen
4 IFS=";_" #Spezial-Variable – enthält Trennzeichen zum
   Trennen von Luftdruck und -temperatur
5 re='^[0-9]+$' # Regulärer Ausdruck, ob Variable eine
   Zahl ist
6 pushbullet_api_key=$(cat /home/pi/Temperaturmessung/
   Fremddateien/pushbullet_settings.txt | head -n 1)
7 pushbullet_device=$(cat /home/pi/Temperaturmessung/
   Fremddateien/pushbullet_settings.txt | tail -n 1)
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
13 gpio write 3 1
```

Programmcode 2: main.sh (Zeile 2 bis 13)

Zeile 14 bis 26 kümmern sich um Parameter, die an das Programm übergeben werden. Mit „-d“ kann die letzte Aufzeichnung überschrieben werden und mit „-h“ wird ein kurzer Info-Text angezeigt.

```
lukas@kinderzimmer:~$ main.sh -h
-d csv-Datei leeren
für weitere Informationen siehe http://winkler.kremszeile.at/
oder https://github.com/Findus23/Umweltdatenmessung
```

```
14 if [ $1 ] # if- und case- Abfrage für Startparameter
15 then
16 __case "$1" in
17 __"-d")rm /home/pi/Temperaturmessung/dygraph.csv
18 ____;;
19 __"-h") echo -e "-d__csv-Datei_leeren_\nfür_weitere_
   Informationen_siehe_http://winkler.kremszeile.at/_
   oder_https://github.com/Findus23/Umweltdatenmessung"
```

3 Software

```
20 _____exit 1
21 _____;;
22 _____*) echo "unbekannter_Parameter_-_Für_Hilfe_-h"
23 _____exit
24 _____;;
25 ____esac
26 ____fi
```

Programmcode 3: main.sh (Zeile 14 bis 26)

Alles, was nun folgt, wird unendlich wiederholt, bis das Programm beendet wird.

```
27 while true
28 do
```

Programmcode 4: main.sh (Zeile 27 bis 28)

In den folgenden drei Zeilen wird die aktuelle Uhrzeit und Datum in drei verschiedenen Formaten für drei verschiedene Zwecke.

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

Tabelle 3.1: Datumsformate

```
29 ____uhrzeit=$(date +%Y/%m/%d\ %H:%M:%S) # z.B.:
    2014/10/05 11:00:00 (für csv-Datei)
30 ____uhrzeit_display=$(date +%d.%m\ %H:%M:%S) # z.B.:
    05.10 11:00:00 (für Display)
31 ____uhrzeit_lang=$(date +%d.%m.%y\ %H:%M:%S) # z.B.:
    05.10.2014 11:00:00 (für Webinterface)
```

Programmcode 5: main.sh (Zeile 29 bis 31)

3 Software

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

Programmcode 6: main.sh (Zeile 32)

Anhang

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).
- 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).
- 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. 5).
- elinux. *RPi Hardware - Power*. 2014. URL: http://elinux.org/index.php?title=RPi_Hardware&oldid=341192#Power (besucht am 04. 07. 2014) (siehe S. 3).
- 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).

Literatur

- Innenraumlufthygiene-Kommission des Umweltbundesamtes. *LEITFADEN FÜR DIE INNENRAUMHYGIENE IN SCHULGEBÄUDEN*. 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 (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 (besucht am 22.11.2014) (siehe S. 3).
- usb-sensors-linux. *Install AirSensor on Linux*. 29. Apr. 2013. URL: https://code.google.com/p/usb-sensors-linux/wiki/Install_AirSensor_Linux (besucht am 08.11.2014) (siehe S. 7).
- Velux. *VELUX Raumluftfühler*. 2014. URL: http://www.velux.de/privatkunden/produkte/integra_system/produkte/produkteempfehlung/raumluftfuehler (besucht am 08.11.2014) (siehe S. 7).
- VISION, ANAG. *AV1624 Datasheet*. 22. Mai 2006. URL: http://conrad.ru/doci/tekstovyy_displey_stn_anag_vision_av1624gfbw_sj__seryy_181664_en.pdf (besucht am 23.11.2014) (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. 19).
- *Raspberry Pi*— *Wikipedia, Die freie Enzyklopädie*. 2014. URL: 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=Fl%C3%83%C2%BCchtige_organische_Verbindung&oldid=41478 (besucht am 11.11.2014) (siehe S. 7).

Abbildungsverzeichnis

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	Anschlussskizze 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	7
2.6	Erstes Display (eigenes Werk)	8
2.7	eingebautes Display (eigenes Werk)	9

Programmcode

1	main.sh (Zeile 1)	10
2	main.sh (Zeile 2 bis 13)	11
3	main.sh (Zeile 14 bis 26)	11
4	main.sh (Zeile 27 bis 28)	12
5	main.sh (Zeile 29 bis 31)	12
6	main.sh (Zeile 32)	13

Glossar

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

A

Ampere

die SI-Basiseinheit der elektrischen Stromstärke 4, 21

C

C

C ist eine sehr weit verbreitete Programmiersprache

Hier wird sie oft zum Auslesen der Sensoren verwendet, da sie sehr schnell ausgeführt wird 6, 21

CPU

Central Processing Unit 4, 21

CSV-Datei

Comma-separated values

Hierbei werden Messungen in einer Textdatei durch Zeilenumbrüche und einzelne Werte durch Beistriche getrennt 1, 13, 21

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.*¹ 21, 24

G

GPIO

General Purpose Input/Output

Kontakte, die Softwareseitig für verschiedene Zwecke angesteuert

¹Wikipedia, 2014a.

Glossar

werden können

z.B.: Auslesen von Sensoren, Ansteuern von Displays 4, 21

H

Hertz

die SI-Basiseinheit für die Frequenz

Sie gibt die Wiederholungen pro Sekunden an (hier: Schwingungen pro Sekunde 4, 21

I

I²C

Inter-Integrated Circuit (auf Deutsch gesprochen: *I-Quadrat-C*)

ein sehr weit verbreiteter [Datenbus](#) 7, 21

K

Kernelmodul

ein Programm, welches in das Betriebssystem geladen werden kann und oft zur Unterstützung von Hardware verwendet wird 6, 21

V

VOC

volatile organic compound (dt. Flüchtige organische Verbindungen) 7, 21

Volt

die SI-Basiseinheit der elektrischen Spannung 5, 21

Eidesstattliche Erklärung

Ich, Lukas Winkler, erkläre hiermit eidesstattlich, dass ich diese vorwissenschaftliche Arbeit selbständig 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