

Systemtechnik Labor

5xHIT 2019/20, 2erTeam

# **GK9.1 Embedded Systems "IoT in der Cloud"**

Theorieprotkoll

Fabian Traxler, Karim Omar, Sam HediyeHloo

22. Januar 2020

Bewertung:

Betreuer: Michael Borko

Version: 1.1

Begonnen: 05. Dezember  
2019

Beendet: 17. April 2018

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Ziele	3
1.2	Voraussetzungen	3
1.3	Detaillierte Ausgabenbeschreibung	3
1.3.1	Recherche	3
1.3.2	Implementierung	3
1.4	Bewertung	4
1.4.1	Anforderungen <b>überwiegend erfüllt</b>	4
1.4.2	Anforderungen <b>zur Gänze erfüllt</b>	4
1.5	Auswahl und Gegenüberstellung von Kommunikationsschnittstellen [3]	5
1.5.1	I <sup>2</sup> C (Inter-Integrated Circuit)	5
1.5.2	SPI Serial Peripheral Interface bus	5
1.5.3	UART Universal Asynchronous Receiver-Transmitter	6
1.5.4	Welche ist geeignet bzw. wo sind die Unterschiede?	7
1.6	Programmierung und Konfiguration eines Single-Board-Computers	7
1.6.1	Auswahl	7
<b>2</b>	<b>Definition</b>	<b>7</b>
<b>3</b>	<b>Beschreibung der verwendeten Technologien</b>	<b>8</b>
3.1	Sensoren	8
3.1.1	Temperature DHT11	8
3.1.2	Pressure BMP280	8
3.1.3	Digital Light Sensor BH1750	9
3.2	Anzeige von Daten mit Grafana Cloud	10
3.2.1	Einführung	10
3.2.2	Funktionen	11
3.3	Persistieren der Daten mit InfluxDB Cloud	11
3.3.1	Einführung	11
3.3.2	Funktionen	11
3.4	Sammeln der Daten	11
	<b>Literaturverzeichnis</b>	<b>16</b>

# 1 Einführung

Diese Übung gibt eine Einführung in die Verwendung von Sensoren und deren Datenanzeige in der Cloud.

## 1.1 Ziele

Das Ziel ist es mit verschiedenen Sensoren Daten auszulesen und diese entsprechend aggregiert über eine Weboberfläche anzuzeigen. Um die Sensordaten verarbeiten und weiterleiten zu können ist entweder ein uC oder ein SBC (z.B. RaspberryPi) notwendig, der über geeignete Schnittstellen Daten an die Cloud weiterleitet.

## 1.2 Voraussetzungen

- Verwendung von Single-Board-Computer bzw. Mikrokontroller
- Verständnis von Bussystemen im Hardware-Bereich
- Aufsetzen von Webservices
- Implementierung von zustandslosen Schnittstellen

## 1.3 Detaillierte Ausgabenbeschreibung

### 1.3.1 Recherche

Überlegen Sie sich die notwendigen Schritte um unterschiedliche Low-Cost Sensoren an einen Gateway anschließen zu können, um die akquirierten Daten über ein Webservice anzeigen zu lassen.

**Bedenken Sie dabei die folgenden Punkte:**

- Abstrakte der Informationen
- Aggregation der Daten
- Schnittstellendefinition (Hardware, Gateway, Services, etc.)
- Energieversorgung
- Speicherverbrauch
- Verbindung

### 1.3.2 Implementierung

Implementieren Sie einen Prototypen, der die oben beschriebenen Punkte realisiert. Verwenden Sie dabei die bereitgestellte Hardware und deployen Sie Ihr Webservice auf eine öffentlich zugängliche Einheit (z.B. Heroku).

## 1.4 Bewertung

Gruppengröße: 3-4 Personen

Protokoll

Deployment auf Heroku

### 1.4.1 Anforderungen überwiegend erfüllt

- Recherche und Zusammenfassung der verwendeten Technologien
- Erstellung eines Prototyps zur Aquirierung von Sensordaten
- Deployment und Dokumentation der grafischen Anzeige der Sensordaten

### 1.4.2 Anforderungen zur Gänze erfüllt

- Aufbau des ganzen Systems mit entsprechender Langzeit-Dokumentation
- Weitere Verwertung der Sensordaten als Export über ReST-Schnittstelle

## 1.5 Auswahl und Gegenüberstellung von Kommunikationsschnittstellen [3]

Für den Raspberry Pi bzw. für SBC's gibt es ein paar geeignete Standard-Kommunikationsschnittstellen um mit verschiedenen Geräten zu kommunizieren, wie mit: Displays, Sensoren, Roboter, Eingabegeräte, Industriegeräte .... usw.

Für den Raspberry Pi gibt es 3 geeignete Protokolle, I<sup>2</sup>C, SPI und UART. Alle 3 sind digitale und serielle Kommunikationsmethoden.

### 1.5.1 I<sup>2</sup>C (Inter-Integrated Circuit)

I<sup>2</sup>C ist eine bidirektionale (Datenübertragung in beide Richtungen, Punkt zu Punkt) , synchrone, serielle Kommunikationsschnittstelle. Es wird auf zwei Leitungen im Half-Duplex betrieben. Entwickelt wurde dies von Philips Semiconductor.

Ein Master kann mit einem oder mehreren Slave-Geräten kommunizieren, dabei wird jedes Gerät mittels einer 7 bit Adresse ausgewählt. Ursprünglich war die Übertragungsrate auf 100 kbit pro Sekunde begrenzt, nun unterstützt es den "Fast-Mode", dieser erlaubt Übertragungsraten bis zu 400 kbit, sowie den "Fast-Mode Plus" mit bis zu 1 Mbit. Dabei spielen aber Kabellänge, Architektur sowie die Software eine große Rolle.

Mit der vorher erwähnten 7-Bit Adressierung, unterstützt ein I<sup>2</sup>C bis zu 127 Geräte. Die beiden Leitungen heißen **SCL** und **SDA**.

**SCL (Signal Clock)**, dies ist die Taktleitung des Busses. Deshalb wird es auch ein synchroner Bus genannt. **SDA** steht für Signal Data und ist die Datenleitung

Aus Sicht des I<sup>2</sup>C Masters unterscheidet man zwischen Read und Write Sequenzen. Bei einer Read Sequenz liest der I<sup>2</sup>C Master Daten vom I<sup>2</sup>C Slave. Bei einer Write Sequenz sendet der I<sup>2</sup>C Master Daten zum Slave.

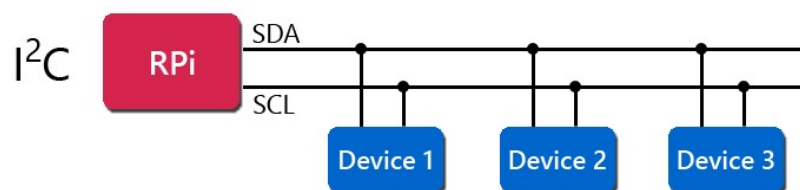


Abbildung 1: I<sup>2</sup>C-Verbindung [3]

### 1.5.2 SPI Serial Peripheral Interface bus

Das Serial Peripheral Interface (SPI) ist ein Bus-System zur synchronen, seriellen Datenübertragung zwischen einem Master und einem oder mehreren Slaves. Wie I<sup>2</sup>C ist es für kurze Distanzen geeignet. Anders aber, funktioniert die Übertragung im Vollduplex, d.h. die Daten fließen in beide Richtungen.

tungen gleichzeitig. Der Master erzeugt Takt- und Steuersignale und startet eine Datenübertragung indem er einen Slave auswählt. Jeder Slave wird vom Master über eine eigene Leitung (Slave Select) angesprochen und ist nur aktiv wenn er ausgewählt wurde. Der SPI-Bus besteht aus zwei Signal- und zwei Datenleitungen:

- Slave Select (SS)
- Signal Clock (SCK)
- Master Out - Slave In (MOSI)
- Master In - Slave Out (MISO)

Alle Leitungen sind unidirektional, also nur in eine Richtung.

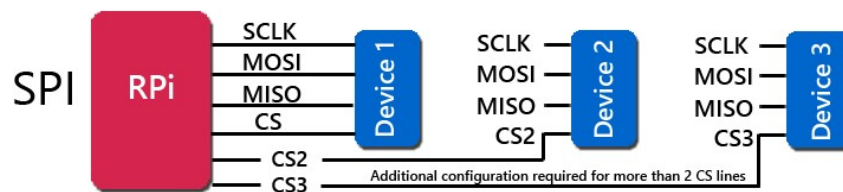


Abbildung 2: SPI-Verbindung [3]

### 1.5.3 UART Universal Asynchronous Receiver-Transmitter

UART ist eine Hardware-Implementierung, die bidirektionale, asynchrone und serielle Kommunikation unterstützt. Es werden zwei Datenleitungen benötigt - eine für das Senden und eine für das Empfangen. Die Sendeleitung einer Vorrichtung ist mit der Empfangsleitung der zweiten Vorrichtung verbunden und umgekehrt für die Übertragung in beide Richtungen.

Ein UART kann nur zwischen zwei Geräten verbunden werden. Es kann zwischen den Geräten in:

1. Simplex - Datenübertragung nur in eine Richtung
2. Halbduplex - Datenübertragung in beide Richtungen, aber nicht gleichzeitig
3. Vollduplex - Datenübertragung in beide Richtungen gleichzeitig

Bei RPi befinden sich die UARTs auf dem Bluetooth-Chip, der etwas weniger mit Funktionen versehene "mini-UART" befindet sich auf dem GPIO-PIN 8 und 10.

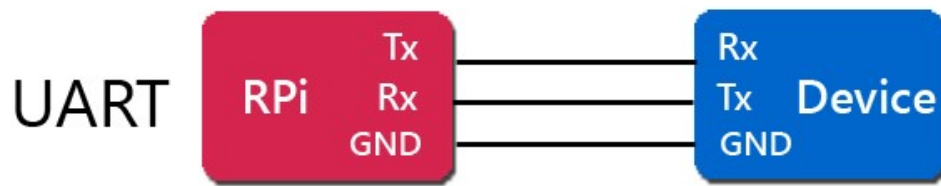


Abbildung 3: UART-Verbindung [3]

#### 1.5.4 Welche ist geeignet bzw. wo sind die Unterschiede?

Es sind einige Punkte bei dem die Auswahl eingeschränkt werden kann:

- **Geschwindigkeit:** UART ist die langsamste, I2C schneller und SPI am schnellsten. Wenn Geschwindigkeit nicht wichtig ist, dann sind alle gut.
- **Anzahl der benötigten Geräte:** UART - nur zwei (RPI + 1 Gerät), SPI - viele, aber darüber hinaus zwei Geräte (+ Pi), wird mit dem Pi durch die erforderliche CS-Leitung für jedes Gerät komplizierter, I2C - bis zu 127 ohne übermäßige Komplexität.
- **Duplex:** UART und SPI können Full-Duplex sein, I2C ist nur Half-Duplex.
- **Anzahl der Drähte:** UART verwendet 1 (nur einseitig) oder 2 (bidirektional); I2C verwendet 2 Leitungen; SPI verwendet 4. (Plus ein Erdungskabel für alle).
- **Entfernung:** Alle haben eine ziemlich gleiche Reichweite
- **Sendebestätigung:** I2C ist das einzige Protokoll unter diesen drei, das sicherstellt, dass die gesendeten Daten vom Gerät empfangen wurden.

### 1.6 Programmierung und Konfiguration eines Single-Board-Computers

Die genaue Durchführung der folgenden Schritte ist im [Readme](#) beschrieben. Es wird ein Anhand eines Raspberry Pi beschrieben, wie man einen SBC konfiguriert.

#### 1.6.1 Auswahl

Es wurde zuerst für die Auswahl des jeweiligen Sensors recherchiert. Wir beschränkten uns auf den Temperatur & Feuchtigkeitssensor von ASAIR (Model: DHT11, Link zum [Datenblatt](#)), den Lichtsensor (Model: BH1750, Link zum [Datenblatt](#)) und den Drucksensor (Model: BMP280, Link zum [Datenblatt](#)).

## 2 Definition

Das Ziel ist mit den verwendeten Sensoren Daten zu bekommen und in eine InfluxDB zu speichern. Anschließend sollen diese Daten in einem Grafana-Overlay angezeigt und ausgewertet werden.

### 3 Beschreibung der verwendeten Technologien

#### 3.1 Sensoren

##### 3.1.1 Temperature DHT11

Der DHT11 Temperatur- und Feuchtigkeitssensor vereint einen Temperatur- und Feuchtigkeitssensor mit einem kalibrierten digitalen Signalausgang. Durch die Verwendung von Digitalerfassungstechniken, sowie Temperatur und Feuchtigkeitserfassung wird eine hohe Langzeitstabilität gewährleistet.

Dieser Sensor ist ein 4-Pin Sensor und muss bei einer Kabellänge von bis zu 20 Metern mit einem 5k Widerstand verwendet werden. [6]

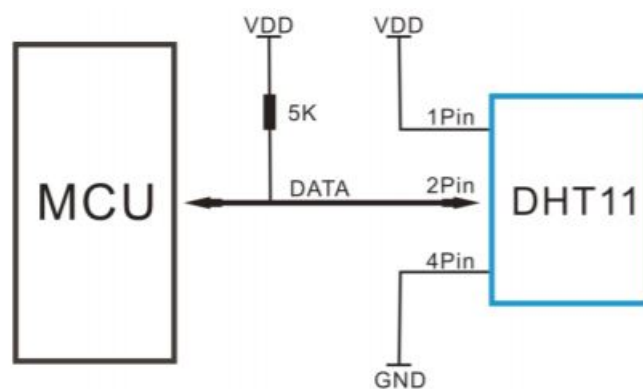


Abbildung 4: 3Pin - Null; MCU=Micro-computer Unite or single chip Computer [6]

##### 3.1.2 Pressure BMP280

BMP280 ist ein absoluter Luftdrucksensor speziell für den mobilen Einsatz konzipiert. Der Sensor ist in einem äußerst kompakten Gehäuse untergebracht. Seine geringen Abmessungen und der geringe Stromverbrauch ermöglichen den Einsatz in batteriebetriebenen Geräten wie Handys, GPS-Module oder Uhren.[1]



## Sensor

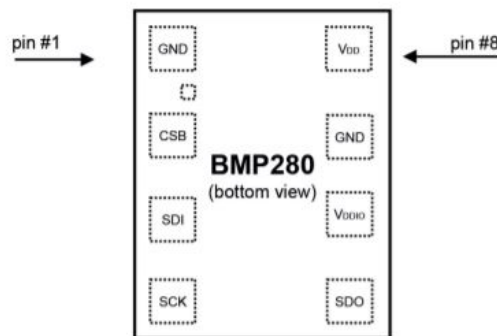


Abbildung 5: Der Sensor mit Ausgängen[1]

## Pinbezeichnung

Pin No.	Name	Function
1	GND	Ground
2	CSB	Chip select
3	SDI	Serial data
4	SCK	Serial clock input
5	SDO	Serial data output
6	VDDIO	Digital Interface supply
7	GND	Ground
8	VDD	Analogue supply

Abbildung 6: Die Pinbezeichnung[1]

### 3.1.3 Digital Light Sensor BH1750

#### Beschreibung

Der BH1750 ist ein Lichtintensitätssensor mit einem eingebauten 16-Bit-AD-Wandler, der direkt ein digitales Signal ausgeben kann ohne Berechnungen. Mit dem BH1750 Lichtsensor kann die Intensität direkt mit dem Luxmeter gemessen werden, ohne dass Berechnungen notwendig sind. Die Daten, die von diesem Sensor ausgegeben werden, werden direkt in Lux (Lx) ausgegeben. Wenn beleuchtete Objekte den Lichtstrom 1 lx in einem Quadratmeter erhalten, beträgt ihre Lichtstärke 1lx. [8]

#### Spezifikationen

- Power Supply Voltage: +3 to 5Vdc
- Messbereich: 1-65535

## Anschluss

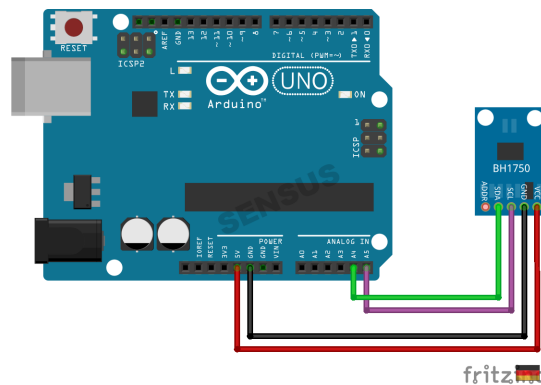


Abbildung 7: Anschluss [2]

### Pins

- PD  
Photo diode with approximately human eye response.
- AMP  
Integration-OPAMP for converting from PD current to Voltage.
- ADC  
AD converter for obtainment Digital 16bit data.
- Logic + I2C Interface
- OSC  
Internal Internal Oscillator ( typ. 320kHz ). It is CLK for internal logic.

## 3.2 Anzeige von Daten mit Grafana Cloud

### 3.2.1 Einführung

Die Grafana Cloud ist eine Instanz von Grafana mit einer hohen Performance und vollständigem Backend. Diese Möglichkeit ist wie der Name schon verrät in der Cloud verfügbar und bietet die selben Funktionen wie Grafana Lokal bietet.[4]

**Grafana** ist eine Open-Source-Software, um Daten aus verschiedenen Quellen zusammenziehen, zu analysieren, überwachen und visuell aufzubereiten. Mit Heatmaps, Histogrammen, Geomaps und sonstige sind vielfältige Visualisierungen von Daten möglich.[4]

### 3.2.2 Funktionen

## 3.3 Persistieren der Daten mit InfluxDB Cloud

### 3.3.1 Einführung

**Was ist InfluxDB?** InfluxDB ist die Open Source Timeseries-Datenbank. Sie ist für hohe Schreib- und Abfragelasten ausgelegt und bietet eine SQL-ähnliche Abfragesprache namens InfluxQL zur Interaktion mit den Daten.[5]

#### **Was ist InfluxDB Cloud?**

Es ist eine schnelle, flexible, serverlose realtime-Überwachungsplattform, eine Dashboarding-Engine, ein Analyseservice und ein Ereignis- und Metrikprozessor.[5]

### 3.3.2 Funktionen

#### **Built for time**

InfluxDB ist ein Datenspeicher, der speziell für zeitabhängige Daten geschrieben wurde. Er ermöglicht die Aufnahme, Komprimierung und Echtzeitabfrage von Daten mit hohem Durchsatz. InfluxDB kann Millionen von Daten pro Sekunde verarbeiten. Die Arbeit mit so vielen Daten über einen langen Zeitraum kann zu erhöhten Speicherkosten führen. InfluxDB komprimiert, komprimiert und downsamplet die Daten automatisch.

#### **Get data from everywhere**

Mit InfluxDB Cloud kann man ganz einfach Einblick in das System oder den Sensoren nehmen. InfluxDB Cloud verbindet direkt zu Telegraf, ein server agent zum Sammeln and Senden von Daten von überall.

#### **Integrated visualization and dashboarding**

InfluxDB Cloud hat eigene vorgefertigte Dashboards und Skripts für gewöhnliche Monitoringprojekte wie: Docker, Kubernetes, Nginx, Redis InfluxDB usw.. Desweiteren unterstützt InfluxDB die Verbindung zu Grafana

#### **Alerting and notification**

InfluxDB Cloud-Benutzer verfügen über Assistenten, die sie beim Definieren von Schwellenwerten, beim Einstellen von Warnungen und beim Senden von Benachrichtigungen über PagerDuty, SMS, E-Mail oder HTTPS unterstützen. [5]

## 3.4 Sammeln der Daten

Ich habe für das Auslesen der Werte für den Licht Sensor I2C verwendet und für den Temperatur und Luftfeuchtigkeits ein fertiges Packages das dies für mich übernimmt. Das ganze ist in einem Python-Script welches in einer While-Loop bis zum Abbruch alle 5 Sekunden die Werte des Sensors abfragt und sie in die Influx Datenbank speichert, die in unsere AWS Instanz läuft.

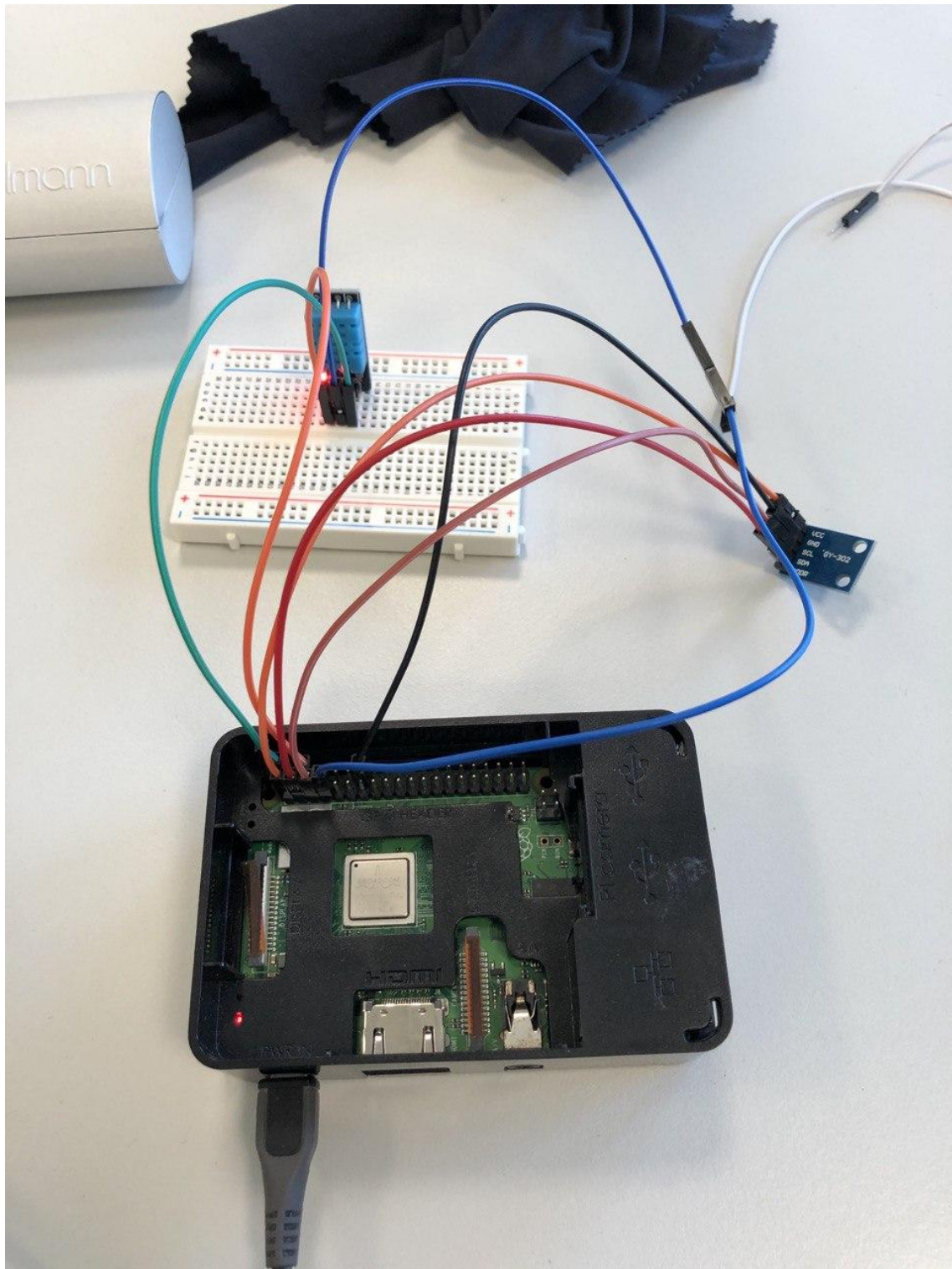


Abbildung 8: Die Schaltung unseres Raspberry Pis mit den zwei Sensoren

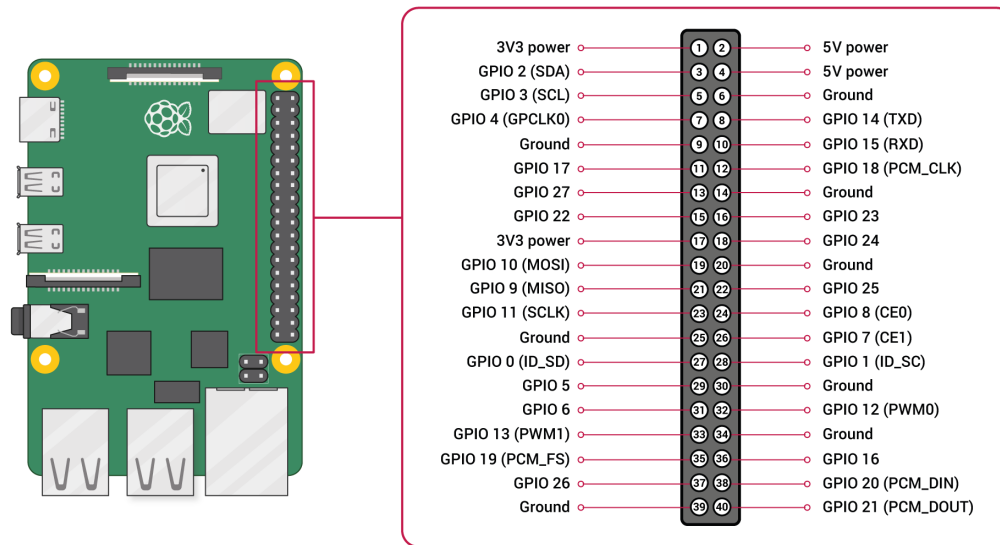


Abbildung 9: Pinout des Raspberry PI 3 B+ Models[7]

**Digital Light Sensor BH1750**

GND	Ground	Pin-16
VCC	3V3 power	Pin-1
SDA	I2C SDA	Pin-3
SLC	I2C SCL	Pin-5

**Temperature DHT11**

GND	Ground	Pin-6
VCC	5V power	Pin-2
DATA	DATA	Pin-7

---

```

import time
from influxdb import InfluxDBClient
import random
import smbus
import Adafruit_DHT

# Konstanten definieren welche im Datasheet angeschrieben sind
DEVICE= 0x23
# Diese Adresse laesst sich auch mittels dem i2cdetect -y 1 heraus finden
POWER_DOWN = 0x00 # Kein aktiver Zustand
POWER_ON = 0x01 # Power an
RESET = 0x07
# Setzt den Wert ab wieviel Lux gemessen werden soll in welchem interwall.
# Nach der Messung wird es automatisch ausgeschalten
ONE_TIME_HIGH_RES_MODE_1 = 0x20
#bus = smbus.SMBus(0)
bus = smbus.SMBus(1) # Ich muss nmlich auch den Befehl i2cdetect -y 1 eingeben und
    nicht 0
def convertToNumber(data):
    # Wandelte die 2 Byte Daten in eine Dezimale Zahl um
    result=(data[1] + (256 * data[0])) / 1.2
    return (result)
def readLight(addr=DEVICE):
    # Liest die Daten von der Kommunikationsschnittstelle I2C an der bestimmten
    Adresse ab.
    data = bus.read_i2c_block_data(addr,ONE_TIME_HIGH_RES_MODE_1)
    return convertToNumber(data)

def main():

    client = InfluxDBClient(host='35.159.21.204', port=8086)
    client.switch_database('sensordaten')

    while True:
        humidity, temperature = Adafruit_DHT.read_retry(11,2)
        light = readLight()
        print("Humidity: "+str(humidity))
        print("Temperature: "+str(temperature))
        print("Light: "+str(light))

        data = [
            {
                "measurement": "humidity",
                "tags": {

```

```
        "user": "Omar_Hediyehloo_Traxler"
    },
    "fields": {
        "value": humidity
    }
}, {
    "measurement": "temperature",
    "tags": {
        "user": "Omar_Hediyehloo_Traxler"
    },
    "fields": {
        "value": temperature
    }
}, {
    "measurement": "light",
    "tags": {
        "user": "Omar_Hediyehloo_Traxler"
    },
    "fields": {
        "value": light
    }
}
]
client.write_points(data)
print("Data written!")
time.sleep(3)

if __name__=="__main__":
    main()
```

---

## Literaturverzeichnis

- [1] Bosch. *Datenblatt*. 16.01.2020. Bosch. URL: <http://descargas.cetronic.es/BOSCHBMP280.pdf>.
- [2] Makershop. *Anschluss des BH1750*. 16.01.2020. Makershop. URL: <https://www.makershop.de/sensoren/lichtsensor/lichtsensor-bh1750/>.
- [3] mbtechworks.com. *Raspberry Pi I2C / SPI / UART Communications*. MBTechWorks. URL: <https://www.mbtechworks.com/hardware/raspberry-pi-UART-SPI-I2C.html>.
- [4] Grafana Members. *Grafana Cloud*. 16.01.2020. Grafana. URL: <https://grafana.com/products/cloud/>.
- [5] InfluxDB Members. *InfluxDB Cloud*. 16.01.2020. InfluxDB. URL: <https://www.influxdata.com/products/influxdb-cloud/>.
- [6] Mouser. *DHT11 Humidity Temperature Sensor*. 09.01.2020. Mouser. URL: [http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001405544-da-01-en-TEMP\\_UND\\_FEUCHTESENSOR\\_DHT11.pdf](http://www.produktinfo.conrad.com/datenblaetter/1400000-1499999/001405544-da-01-en-TEMP_UND_FEUCHTESENSOR_DHT11.pdf).
- [7] raspberry. *pinout*. 22.01.2020. Raspberry. URL: <https://www.raspberrypi.org/documentation/usage/gpio/>.
- [8] Wikidfrobot. *BH1750 light intensity sensor*. 15.01.2020. wiki.dfrobot. URL: [https://wiki.dfrobot.com/Light\\_Sensor\\_\\_SKU\\_SEN0097\\_](https://wiki.dfrobot.com/Light_Sensor__SKU_SEN0097_).

## Abbildungsverzeichnis

1	I <sup>2</sup> C-Verbindung [3]	5
2	SPI-Verbindung [3]	6
3	UART-Verbindung [3]	7
4	3Pin - Null; MCU=Micro-computer Unite or single chip Computer [6]	8
5	Der Sensor mit Ausgängen[1]	9
6	Die Pinbezeichnung[1]	9
7	Anschluss [2]	10
8	Die Schaltung unseres Raspberry Pis mit den zwei Sensoren	12
9	Pinout des Raspberry PI 3 B+ Models[7]	13