

Hochschule für angewandte Wissenschaften Coburg  
Fakultät Elektrotechnik und Informatik

Studiengang: Informatik M.Sc.

Projektdokumentation zu Hardware cyber-physischer Systeme

AirGuardX

Benedikt Scholz, Yannik Wolf

Abgabe der Arbeit: DD. MM JJJJ

Betreut durch:  
Prof. Dr.-Ing. Matthias Mörz, Hochschule Coburg>

# Inhaltsverzeichnis

[Inhaltsverzeichnis 2](#_Toc137124663)

[Abbildungsverzeichnis 4](#_Toc137124664)

[Tabellenverzeichnis 5](#_Toc137124665)

[Codebeispielverzeichnis 6](#_Toc137124666)

[Symbolverzeichnis 7](#_Toc137124667)

[Abkürzungsverzeichnis 8](#_Toc137124668)

[1 Einleitung 9](#_Toc137124669)

[2 Sensoren 11](#_Toc137124670)

[2.1 Übersicht Sensoren 11](#_Toc137124671)

[2.2 Sensorwahl 13](#_Toc137124672)

[3 Architekturübersicht 14](#_Toc137124673)

[4 Grundlagen zu den verwendeten Technologien 16](#_Toc137124674)

[4.1 Raspberry Pi 16](#_Toc137124675)

[4.2 I2C 16](#_Toc137124676)

[4.3 WLAN 17](#_Toc137124677)

[4.4 Gateway 18](#_Toc137124678)

[4.5 HTTP-Protokoll 19](#_Toc137124679)

[4.6 IP-Adresse 19](#_Toc137124680)

[4.7 Webserver 19](#_Toc137124681)

[4.8 Cloud 20](#_Toc137124682)

[4.9 Datenbank 20](#_Toc137124683)

[4.10 Webanwendung 21](#_Toc137124684)

[5 Projektdurchführung 22](#_Toc137124685)

[5.1 Inbetriebnahme Adafruit BME680 und Raspberry Pi 22](#_Toc137124686)

[5.2 Konfiguration der I2C-Schnittstelle 24](#_Toc137124687)

[5.3 Auslesen der Sensordaten 24](#_Toc137124688)

[5.4 Konfiguration der HTTP-Schnittstelle 26](#_Toc137124689)

[5.5 Konfiguration des Webservers 26](#_Toc137124690)

[5.6 Senden der Daten an den Webserver 26](#_Toc137124691)

[5.7 Automatisierung der Datenabfrage und -sendung 26](#_Toc137124692)

[5.8 Erweiterung des Versuchsaufbau um eine LED 26](#_Toc137124693)

[5.9 Konfiguration der Telegram-Schnittstelle 26](#_Toc137124694)

[5.10 Entwicklung einer Webanwendung 26](#_Toc137124695)

[Quellenverzeichnis 27](#_Toc137124696)

[Glossar 28](#_Toc137124697)

[Anhang A 1. Formate für das gesamte Dokument 29](#_Toc137124698)

[Ehrenwörtliche Erklärung 30](#_Toc137124699)

# Abbildungsverzeichnis

[Abb. 1: Übersicht Raumluft 9](#_Toc136423532)

[Abb. 2: Versuchsaufbau 10](#_Toc136423533)

[Abb. 3: Architekturübersicht 3](#_Toc136423534)

# Tabellenverzeichnis

[Tab. 1: Übersicht Sensoren 3](#_Toc136427376)

[Tab. 2: Verkabelung Raspberry Pi und Adafruit BME680 12](#_Toc136427377)

# Codebeispielverzeichnis

[Code 1: Python Skript zur Abfrage der Sensordaten 14](#_Toc136428335)

# Symbolverzeichnis

| **Symbol** | **Bedeutung** | **[phys. Einheit]** |
| --- | --- | --- |
|  | cos-Fourierkoeffizient der -ten Harmonischen | [V] |
|  | sin-Fourierkoeffizient der -ten Harmonischen | [V] |
|  | ganzzahlige Laufvariable |  |
|  | Umfang der Messreihe oder Stichprobe |  |
|  | Signalspannungsverlauf | [V] |
|  | empirische Standardabweichung | [m] |
|  | Periodendauer | [s] |
|  | Mittelwert der Stichprobe | [m] |
|  | Einzelmesswert | [m] |
|  | ganzzahlige Laufvariable |  |
|  | Kreisfrequenz der Grundschwingung | [rad/sec] |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Abkürzungsverzeichnis

BDSG Bundesdatenschutzgesetz

EIT Master-Studiengang Elektro- und Informationstechnik

FBML Facebook Markup Language

o. V. ohne Verfasser

… …

<im Abkürzungsverzeichnis stehen alle verwendeten Abkürzungen *in alphabetischer Reihenfolge*, die *nicht* im Duden zu finden sind>

# Einleitung

Eine Idee zur Schimmelprävention und Erhaltung der Luftqualität ist die Verwendung von IoT-Sensoren und Cloudservices. Diese messen die Luftfeuchtigkeit, Temperatur, CO, CO2, Feinstaub und VOCs in einer Wohnung beziehungsweise einem Haus. Primär stehen dabei die Gesundheit der Menschen im Haushalt und die Erhaltung der Gebäudequalität im Fokus.

Schimmelbildung tritt häufig auf, wenn die Luftfeuchtigkeit zu hoch ist. CO entsteht bei unvollständiger Verbrennung, beispielsweise bei einem Gaskocher in der Küche, und wirkt in größeren Mengen als starkes Atemgift. CO2 wird von Menschen und Tieren ausgeatmet und kann sich bei einer zu hohen Menge auf die Konzentrationsfähigkeit auswirken. Ähnlich ist es bei VOCs. Diese beeinträchtigen ebenfalls die Konzentrationsfähigkeit und verursachen Kopfschmerzen, Müdigkeit und Schlafstörungen. Zuletzt steht noch der Feinstaub. Dieser wird durch das Abbrennen von Kerzen, Staubsaugen, Kochen und Drucken im Gebäude verteilt und wirkt sich negativ auf die Atemwege aus. Eine regelmäßige Überwachung und Kontrolle dieser Parameter können dazu beitragen, gesundheitlichen Problemen des Menschen vorzubeugen und Schäden in den Gebäuden zu verhindern.

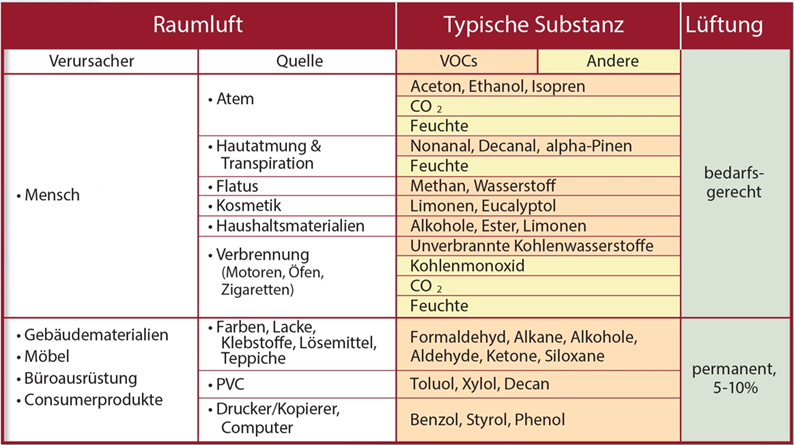


Abb. 1: Übersicht Raumluft

Die IoT-Sensoren werden in mehreren Räumen installiert und sammeln Daten zu den oben genannten Parametern. Diese Daten werden über eine Cloud-Plattform an eine zentrale Anwendung gesendet, die Warnungen und Handlungsempfehlungen per Messenger (Telegram) ausgibt, wenn ein Parameter ein bestimmtes Niveau erreicht oder überschreitet. Eine Übersicht über die aktuelle Luftsituation der einzelnen Räume, kann zudem über eine Web-Oberfläche abgefragt werden, die den zeitlichen Verlauf der Sensor-Daten berücksichtigt und dadurch Rückschlüsse ziehen kann.

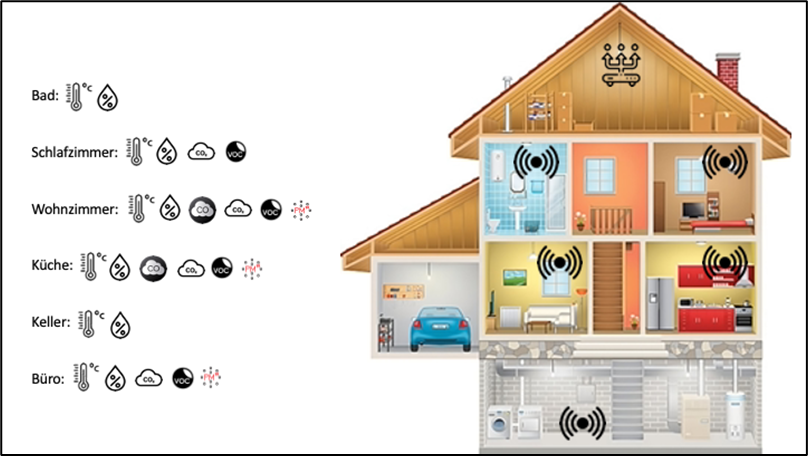


Abb. 2: Versuchsaufbau

Zusammenfassend lässt sich sagen, dass eine IoT-Lösung zur Schimmelprävention und Erhaltung der Luftqualität die Verwendung von Sensoren, Cloudservices und intelligenten Geräten zur Überwachung und Kontrolle von Luftfeuchtigkeit, Temperatur und Luftqualität umfassen würde, um sicherzustellen, dass ein Raum und die Menschen darin sicher und gesund sind.

# Sensoren

## Übersicht Sensoren

**(** favorisierte Sensoren)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Temp. | Hum. | Pres. | Alt | Gas | Daten | Preis | Link |
| Adafruit SCD-41 | X | X |  |  | CO2 (NDIR)  ±(50 ppm + 5%) |  | 50€ | <https://www.adafruit.com/product/5190> |
| Adafruit SCD-30 | X | X |  |  | CO2 (NDIR)  ±(30 ppm + 3%) | I2C | 60€ | <https://www.adafruit.com/product/4867> |
| Adafruit MiCS5524 |  |  |  |  | CO2, VOCs (MOX) |  | 15€ | <https://www.adafruit.com/product/3199> |
| Adafruit BME688 | X  ± 1° | X  ± 3% | X  ± 1 hPa | X  ± 1m | VOCs (MOX) | SPI / I2C | 20€ | <https://www.adafruit.com/product/5046> |
| Adafruit BME680 | X  ± 1° | X  ± 3% | X  ±1 hPa | X  ± 1m | VOCs (MOX) | SPI / I2C | 19€ | <https://www.adafruit.com/product/3660> |
| Adafruit SGP40 |  |  |  |  | VOCs, H2 (MOX) | I2C | 15€ | <https://www.adafruit.com/product/4829> |
| Adafruit SGP30 |  |  |  |  | VOCs, H2, CO2 (MOX) | I2C | 18€ | <https://www.adafruit.com/product/3709> |
| Adafruit ENS160 |  |  |  |  | VOCs, CO2 (MOX) | I2C | 22€ | <https://www.adafruit.com/product/5606> |
| Adafruit HTU21D-F | X  ± 1° | X  ± 2% |  |  |  | I2C | 11€ | <https://www.adafruit.com/product/1899> |
| MPL3115A2 | X  ± 1° |  | X | X |  | I2C | 10€ | <https://www.adafruit.com/product/1893> |
| Adafruit PMSA003I |  |  |  |  | Feinstaub | I2C | 45€ | <https://www.adafruit.com/product/4632> |

Tab. 1: Übersicht Sensoren

MOX = Metalloxid

* Mithilfe eines MOX-Sensors können VOCs ermittelt werden. Oft gibt der Sensor aber nur einen Wert zurück und kann die Gase nicht einzeln bestimmen.
* Sensoren benötigen meist „Warmlaufzeit“ von bis zu 48h und vor jeder weiteren Nutzung 30min im Vorlauf

VOC = Volatile organic compounds [

* Kohlenmonoxid (CO), Wasserstoff (H2)
* Stickstoffmonoxid (NO), Stickstoffdioxid (NO2), Ozon (O3), Ammoniak (NH3), Schwefeldioxid (SO2)
* Kohlenwasserstoffe (CXHY von C1-C8), z.B. Methan (CH4), Propan (C3H8)
* Alkohole (CXHYOH), z.B. Ethanol (C2H5OH) etc., weitere ausgewählte Volatile Organic Compounds (VOCs), Kältemittel (R134a etc.)

NDIR = Nondispersive infrared sensor (Gas-Sensor), oft genauer als MOX-Sensor

## Sensorwahl

In der engeren Auswahl der potentiell möglichen Raumluftsensoren stehen der Adafruit SCD-30 und der Adafruit BME680. Da in der Hochschule Coburg bereits der Adafruit BME680 vorhanden und verwendbar ist, fällt die Entscheidung auf diesen. Im Laufe des Projektes wurde dieser allerdings durch den SCD30 ersetzt.

Zusätzlich soll der Adafruit PMSA003I als Feinstaubsensor zum Einsatz kommen. Da dessen Beschaffung allerdings mit Zeit- und Geldaufwand verbunden ist, wird zunächst auf dessen Inbetriebnahme verzichtet.

# Architekturübersicht

Die nachfolgende Abbildung 3 zeigt den groben Aufbau der Gesamtarchitektur des Projektes AirGuardX. Zu Beginn sei angemerkt, dass AirGuardX mit dem Projekt SmartStall Überschneidungspunkte aufweist. Beide Projekte nutzen dieselbe Cloud, um ihre Daten abzulegen und in Form einer Webanwendung zu repräsentieren. Die Webanwendungen werden allerdings getrennt voneinander entwickelt. Die gemeinsame Nutzung der Cloud resultiert in der Ähnlichkeit der Projekte. Beide versuchen anhand von Sensordaten die Luftqualität zu messen und Handlungsanweisungen auszusprechen. AirdGuardX tut dies für den privaten Haushalt, während SmartStall auf landwirtschaftlich genutzte Tierställe abzielt. Das Ziel soll es sein, beide Projekte am Ende zusammenzuführen, um ein gemeinschaftliches, vollumfängliches All-In-One-Produkt für private als auch gewerbliche Zwecke anbieten zu können.

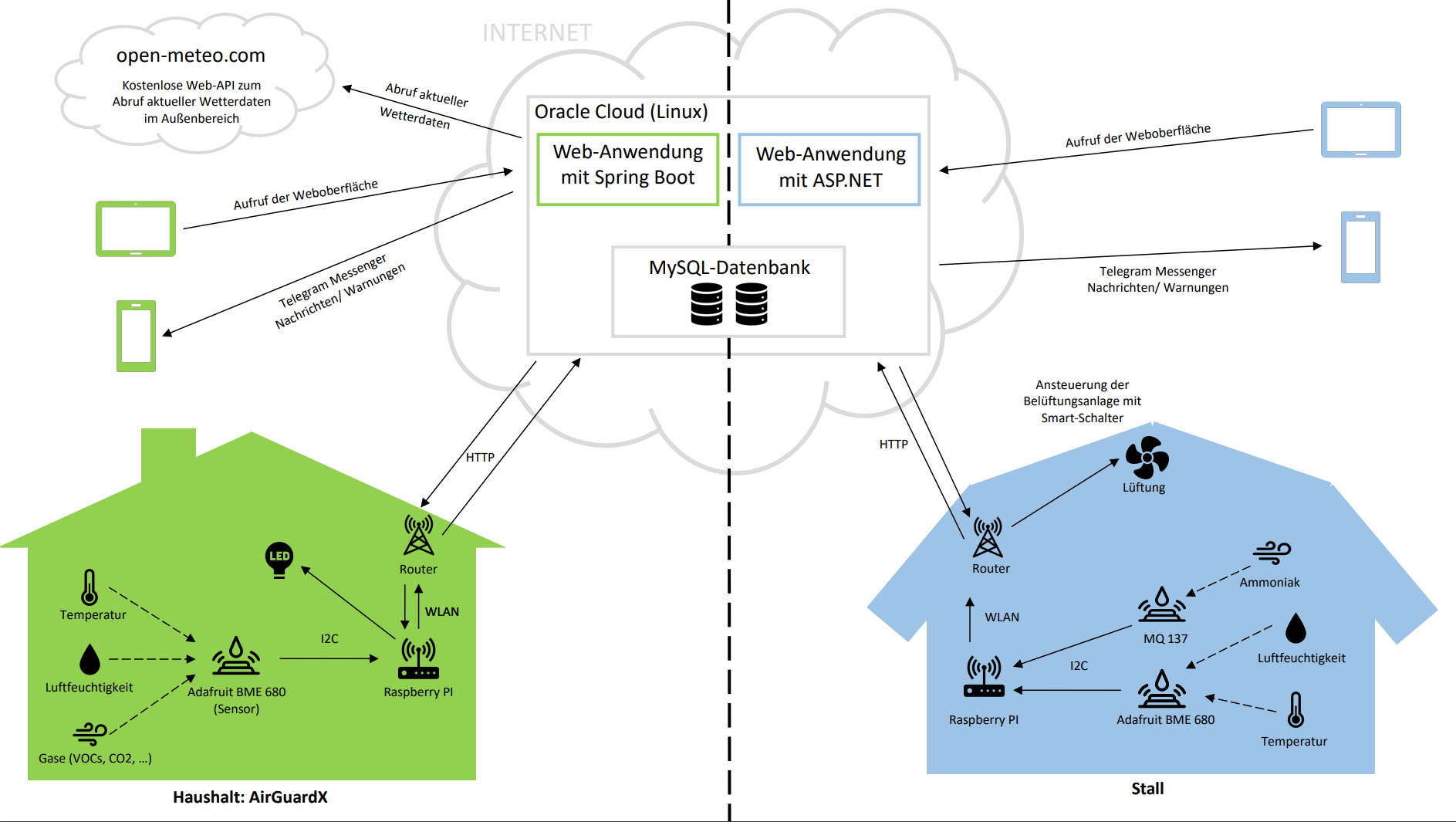


Abb. 3: Architekturübersicht

Beim Projekt AirGuardX werden die in Tabelle 1 zu sehenden Messdaten des Adafruit BME680 erhoben und über I2C an den Raspberry Pi, mit welchem der Sensor verbunden ist, übertragen. Der Raspberry Pi nutzt dann WLAN als Gateway, um die Daten alle 30 Sekunden über das HTTP-Protokoll an die IP-Adresse des Webservers zu senden. Dort werden die Daten in einer Datenbank abgelegt. Sollten bestimmte Messwerte überschritten werden, wird vom Webserver automatisch eine Push-Benachrichtigung über Telegram an den Nutzer gesendet und eine rote beziehungsweise gelbe LED leuchtet an der Messtation auf. Solange die Messwerte in einem normalen Bereich liegen, leuchtet ein grünes Licht. Des Weiteren steht auf dem Webserver noch eine Webanwendung zur Verfügung. Über diese kann der Zustand der Räume monitort und nachvollzogen werden.

Die Beschreibung der Arbeitsweise des SmartStall ist in deren gesonderter Dokumentation zu finden.

# Grundlagen zu den verwendeten Technologien

## Raspberry Pi

Ein Raspberry Pi ist ein kleiner, kostengünstiger Einplatinencomputer, der von der Raspberry Pi Foundation entwickelt wurde. Mittlerweile hat sich der Raspberry Pi jedoch zu einem beliebten Gerät für viele Anwendungen entwickelt, darunter Heimautomatisierung, Mediacenter, Robotik, IoT-Projekte und vieles mehr.

Der Raspberry Pi besteht aus einer einzigen Platine, die alle Hauptkomponenten eines Computers enthält, darunter Prozessor, Arbeitsspeicher, Speicher, Anschlüsse und Erweiterungsschnittstellen. Es gibt verschiedene Modelle des Raspberry Pi, die sich in Bezug auf Leistung, Speicher, Anschlüsse und Funktionen unterscheiden. Die meisten Modelle verwenden eine ARM-Architektur und laufen mit einem Linux-basierten Betriebssystem, wie zum Beispiel Raspbian, das auf Debian basiert.

Der Raspberry Pi verfügt über eine Vielzahl von Anschlüssen, darunter USB-Ports, HDMI- und Audioausgänge, Ethernet-Anschluss, GPIO-Pins (General Purpose Input/Output) und Kameraanschlüsse. Dadurch kann er mit verschiedenen Geräten und Sensoren verbunden werden, um komplexe Projekte umzusetzen.

## I2C

I2C steht für Inter-Integrated Circuit und ist ein serieller Kommunikationsbus, der entwickelt wurde, um die Kommunikation zwischen integrierten Schaltkreisen (ICs) zu ermöglichen. Er wird häufig zur Verbindung von Komponenten in elektronischen Geräten verwendet, insbesondere in Mikrocontrollern und eingebetteten Systemen.

I2C verwendet zwei Signalleitungen: eine Datenleitung (SDA - Serial Data Line) und eine Taktleitung (SCL - Serial Clock Line). Diese Signalleitungen ermöglichen die bidirektionale Kommunikation zwischen einem Master-Gerät und einem oder mehreren Slave-Geräten. Der Master-Gerät initiiert die Kommunikation und kontrolliert den Ablauf, während die Slave-Geräte auf Anfragen des Masters reagieren.

Das I2C-Protokoll ermöglicht es, eine große Anzahl von Geräten über eine einzige Busleitung zu verbinden. Jedes Gerät auf dem Bus hat eine eindeutige Adresse, die verwendet wird, um es zu identifizieren und anzusprechen. Dadurch können mehrere ICs miteinander kommunizieren, ohne dass jedes Gerät eine separate Verbindung zum Master benötigt.

Die Kommunikation auf dem I2C-Bus erfolgt in Form von Datenpaketen, die in Start- und Stop-Bedingungen eingebettet sind. Der Master initiiert die Kommunikation mit einer Start-Bedingung und gibt die Adresse des Slave-Geräts an, mit dem er kommunizieren möchte. Dann erfolgt der Datenaustausch, bei dem der Master Daten sendet oder vom Slave empfängt. Am Ende der Kommunikation wird eine Stop-Bedingung gesendet.

I2C bietet eine einfache und effiziente Möglichkeit für die Kommunikation zwischen ICs und ermöglicht es Geräten, Informationen auszutauschen, Befehle zu senden oder Sensordaten zu übertragen. Es ist ein weit verbreitetes Kommunikationsprotokoll in der Elektronik und findet Anwendung in verschiedenen Bereichen wie der Steuerung von Peripheriegeräten, Sensoranbindungen, Displayansteuerung und vielen anderen Anwendungen. Im Falle dieses Projektes ist es die Schnittstelle, über welche die Daten vom Sensor an den Raspberry Pi übertragen werden.

## WLAN

WLAN steht für Wireless Local Area Network. Es handelt sich um eine drahtlose Netzwerktechnologie, die es Geräten ermöglicht, miteinander zu kommunizieren und auf das Internet zuzugreifen, ohne dass physische Verbindungskabel erforderlich sind.

Ein WLAN basiert auf dem IEEE 802.11-Standard und nutzt Funkwellen, um Daten zwischen Geräten zu übertragen. Die Geräte, die WLAN nutzen möchten, müssen mit einem drahtlosen Netzwerkadapter ausgestattet sein, der in der Lage ist, die Signale zu empfangen und zu senden. Typische Geräte mit WLAN-Funktion sind Laptops, Smartphones, Tablets, Smart-TVs und viele andere internetfähige Geräte.

Ein WLAN besteht aus einem oder mehreren Access Points (APs), die als zentraler Zugangspunkt dienen und die drahtlose Kommunikation ermöglichen. Diese Access Points sind in der Regel über ein kabelgebundenes Netzwerk mit einem Router oder einem Switch verbunden, der den Internetzugang bereitstellt. Ein WLAN kann auch über mehrere Access Points hinweg erweitert werden, um eine größere Abdeckung und eine bessere Signalqualität zu gewährleisten.

Die Kommunikation in einem WLAN erfolgt über Funkwellen auf bestimmten Frequenzbändern, wie zum Beispiel 2,4 GHz oder 5 GHz. Die Übertragungsraten können je nach WLAN-Standard variieren, wobei modernere Standards wie 802.11ac oder 802.11ax höhere Datenraten ermöglichen.

Die Sicherheit in einem WLAN kann durch verschiedene Verschlüsselungsprotokolle wie WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) oder WPA2 gewährleistet werden. Diese Protokolle helfen, die drahtlose Kommunikation vor unautorisiertem Zugriff zu schützen.

WLAN bietet eine bequeme und flexible Möglichkeit, Geräte drahtlos zu vernetzen und auf das Internet zuzugreifen. Es ermöglicht die Mobilität der Geräte innerhalb der Reichweite des Netzwerks und hat sich zu einer weit verbreiteten Technologie entwickelt, die in privaten Haushalten, Büros, öffentlichen Bereichen und vielen anderen Umgebungen genutzt wird. Im Falle dieses Projektes wird es als Gateway benutzt, um die Sensordaten vom Raspberry Pi über das HTTP-Protokoll an den Webserver und somit ins Internet zu senden.

## Gateway

Ein Gateway ist eine Netzwerkgerät, das als Schnittstelle zwischen verschiedenen Netzwerken dient. Es ermöglicht die Kommunikation zwischen unterschiedlichen Netzwerkprotokollen, Netzwerktypen oder Netzwerksegmenten, die ansonsten nicht direkt miteinander kommunizieren könnten.

Ein Gateway fungiert als Übersetzer, der Datenpakete von einem Netzwerkformat in ein anderes umwandelt, um die reibungslose Kommunikation zwischen den Netzwerken zu ermöglichen. Es übersetzt die Protokolle, Adressierungen und andere Netzwerkparameter, sodass die Daten korrekt von einem Netzwerk zum anderen übertragen werden können.

Es gibt verschiedene Arten von Gateways, die je nach den spezifischen Anforderungen und Konfigurationen des Netzwerks unterschiedlich sein können.

WLAN, welches in diesem Projekt als Gateway genutzt wird, ist als Netzwerk-Gateway einzustufen. Netzwerk-Gateways verbinden Netzwerke unterschiedlicher Typen miteinander, beispielsweise ein lokales Ethernet-Netzwerk mit einem drahtlosen WLAN-Netzwerk oder einem Mobilfunknetzwerk.

## HTTP-Protokoll

HTTP steht für Hypertext Transfer Protocol. Es handelt sich um ein Protokoll, das zur Übertragung von Daten über das Internet verwendet wird. HTTP ermöglicht die Kommunikation zwischen einem Webbrowser/Client (in diesem Fall der Raspberry Pi) und einem Webserver.

HTTP funktioniert nach dem Client-Server-Modell. Der Webbrowser sendet eine Anfrage an den Webserver, der daraufhin eine Antwort zurückgibt. Diese Anfrage- und Antwortnachrichten bestehen aus einem Header und einem optionalen Nachrichteninhalt.

Der Header enthält Informationen über die Art der Anfrage oder Antwort, wie z.B. den verwendeten HTTP-Verb (GET, POST, etc.), den Hostnamen des Servers und andere Metadaten. Der Nachrichteninhalt kann optional sein und Daten wie HTML-Seiten, Bilder, Videos oder andere Ressourcen enthalten.

## IP-Adresse

Eine IP-Adresse (Internet Protocol-Adresse) ist eine eindeutige numerische Kennung, die einem Gerät zugewiesen wird, das mit einem Computernetzwerk verbunden ist. Sie dient dazu, Geräte innerhalb eines Netzwerks zu identifizieren und ihnen die Kommunikation miteinander zu ermöglichen.

## Webserver

Ein Webserver ist ein spezieller Server, der HTTP (Hypertext Transfer Protocol) verwendet, um Webseiten und andere Ressourcen über das World Wide Web bereitzustellen. Er nimmt Anfragen von Clients (meist Webbrowsern) entgegen und liefert die angeforderten Inhalte zurück.

Ein Webserver ist in der Regel eine Softwareanwendung, die auf einem physischen Server oder einer virtuellen Maschine läuft. Diese Software ermöglicht es dem Server, Webseiten, Bilder, Videos, Dateien und andere Inhalte über das Internet zu hosten und bereitzustellen.

Bekannte Webserver-Softwarelösungen sind Apache HTTP Server, Nginx, Microsoft Internet Information Services (IIS) und Lighttpd. Diese Server-Softwaren haben unterschiedliche Funktionen, Leistungsfähigkeit und Konfigurationsmöglichkeiten, aber sie dienen alle dem Zweck, Webinhalte über das Internet bereitzustellen. In diesem Projekt wird Apache genutzt.

Ein Webserver bildet die Grundlage für die Bereitstellung von Websites und Webanwendungen im Internet. Er ermöglicht es Benutzern, auf Webseiten zuzugreifen, Informationen abzurufen und mit den Inhalten zu interagieren.

## Cloud

Eine Cloud bezieht sich in der Informationstechnologie auf eine Infrastruktur, die es Unternehmen, Organisationen und Privatpersonen ermöglicht, verschiedene Dienste, Ressourcen und Anwendungen über das Internet bereitzustellen, zu nutzen und zu verwalten.

Im Wesentlichen handelt es sich bei der Cloud um eine virtuelle Umgebung, in der Daten, Software, Speicherplatz und Rechenleistung auf Servern gehostet werden, die über das Internet zugänglich sind. Diese Server können sich an verschiedenen Standorten weltweit befinden und sind über ein Netzwerk miteinander verbunden.

Die Cloud bietet verschiedene Dienste und Modelle an, darunter Infrastructure as a Service (IaaS). Hierbei werden virtuelle Maschinen, Speicher und Netzwerkinfrastruktur bereitgestellt. Benutzer können ihre eigenen Anwendungen und Betriebssysteme in der Cloud ausführen, ohne physische Hardware erwerben oder verwalten zu müssen. In diesem Projekt wurde das IaaS Angebot von Oracle genutzt um eine Linux-VM zu betreiben.

## Datenbank

Eine Datenbank ist eine organisierte Sammlung von strukturierten Daten, die elektronisch gespeichert, verwaltet und abgerufen werden können. Sie dient dazu, große Mengen von Informationen effizient zu speichern, zu organisieren und abzufragen.

Eine Datenbank besteht aus einer oder mehreren Tabellen, die Informationen in Zeilen und Spalten strukturieren. Jede Zeile repräsentiert einen Datensatz, während jede Spalte ein Attribut oder eine Eigenschaft darstellt. Durch die Verwendung von Schlüsseln und Beziehungen zwischen den Tabellen können Datenbanken komplexe Zusammenhänge und Abhängigkeiten zwischen den Datensätzen darstellen.

Datenbanken spielen eine entscheidende Rolle bei der Speicherung und Verwaltung von Daten in verschiedenen Anwendungen und Systemen. Sie ermöglichen es, große Mengen an Informationen effizient zu organisieren, abzurufen und zu analysieren.

## Webanwendung

Eine Webanwendung, auch als Web-App abgekürzt, ist eine Softwareanwendung, die über einen Webbrowser oder eine webbasierte Benutzeroberfläche aufgerufen und genutzt wird. Im Gegensatz zu herkömmlichen Desktop-Anwendungen, die auf dem lokalen Computer installiert werden, läuft eine Webanwendung auf einem Webserver und wird über das Internet bereitgestellt.

Webanwendungen werden mithilfe von Webtechnologien wie HTML (Hypertext Markup Language), CSS (Cascading Style Sheets) und JavaScript entwickelt. Sie können auf verschiedenen Plattformen und Betriebssystemen genutzt werden, da der Zugriff über den Webbrowser erfolgt, der auf den meisten Geräten verfügbar ist.

# Projektdurchführung

## Inbetriebnahme Adafruit BME680 und Raspberry Pi

Im Rahmen des Projekts "Hardware cyber-physischer Systeme" erhielt das Projektteam von Prof. Dr.-Ing. Matthias Mörz einen Raspberry Pi sowie den Adafruit BME680-Sensor. Nach Erhalt der Hardware wurde diese sorgfältig überprüft und in Betrieb genommen.

Zur Inbetriebnahme wurde das Betriebssystem Raspberry Pi OS (64-bit) with desktop mithilfe des Raspberry Pi Imagers auf die mitgelieferte microSD-Karte geschrieben und in den Raspberry Pi eingesteckt. Nach dem Anschließen an eine Stromquelle startete und bootete das Gerät innerhalb kurzer Zeit und war einsatzbereit.

Die Verbindung zum Raspberry Pi erfolgt über SSH, um eine sichere und ferngesteuerte Verbindung herzustellen. Durch Ausführen des Befehls "sudo apt update && sudo apt upgrade" wurde die installierte Software auf den neuesten Stand aktualisiert.

Als nächstes muss der Adafruit BME680-Sensor mit dem Raspberry Pi verbunden werden. Hierfür werden Jumper-Kabel benötigt, die von Herrn Mörz zur Verfügung gestellt wurden. Die genaue Pinbelegung für den Anschluss des Sensors kann mithilfe des Befehls "pin-out" abgerufen werden. Zur Ausführung dieses Befehls muss das Programm "python3-gpiozero" mit dem Befehl "apt install python3-gpiozero" installiert werden.

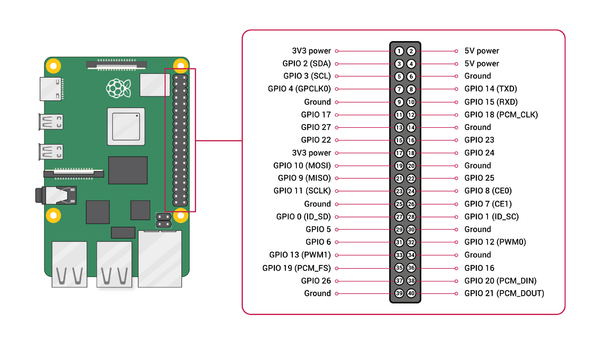


Abb. 4: Pinbelegung des Raspberry Pi

Der Adafruit BME680 bietet sieben Anschlüsse VIN, 3VO, GND, SCK, SDO, SDI und CS. Der Sensor und der Raspberry Pi werden nun mittels Jumper Kabel verkabelt.

|  |  |
| --- | --- |
| **Raspberry Pi GPIO Pin** | **Adafruit BME680 Pin** |
| 1 (3v3 power) | VIN |
| 6 (Ground) | GND |
| 5 (GPIO3 / SCL) | SCK |
| 3 (GPIO2 / SDA) | SDI |

Tab. 2: Verkabelung Raspberry Pi und Adafruit BME680

Nun sieht der Versuchsaufbau folgendermaßen aus.

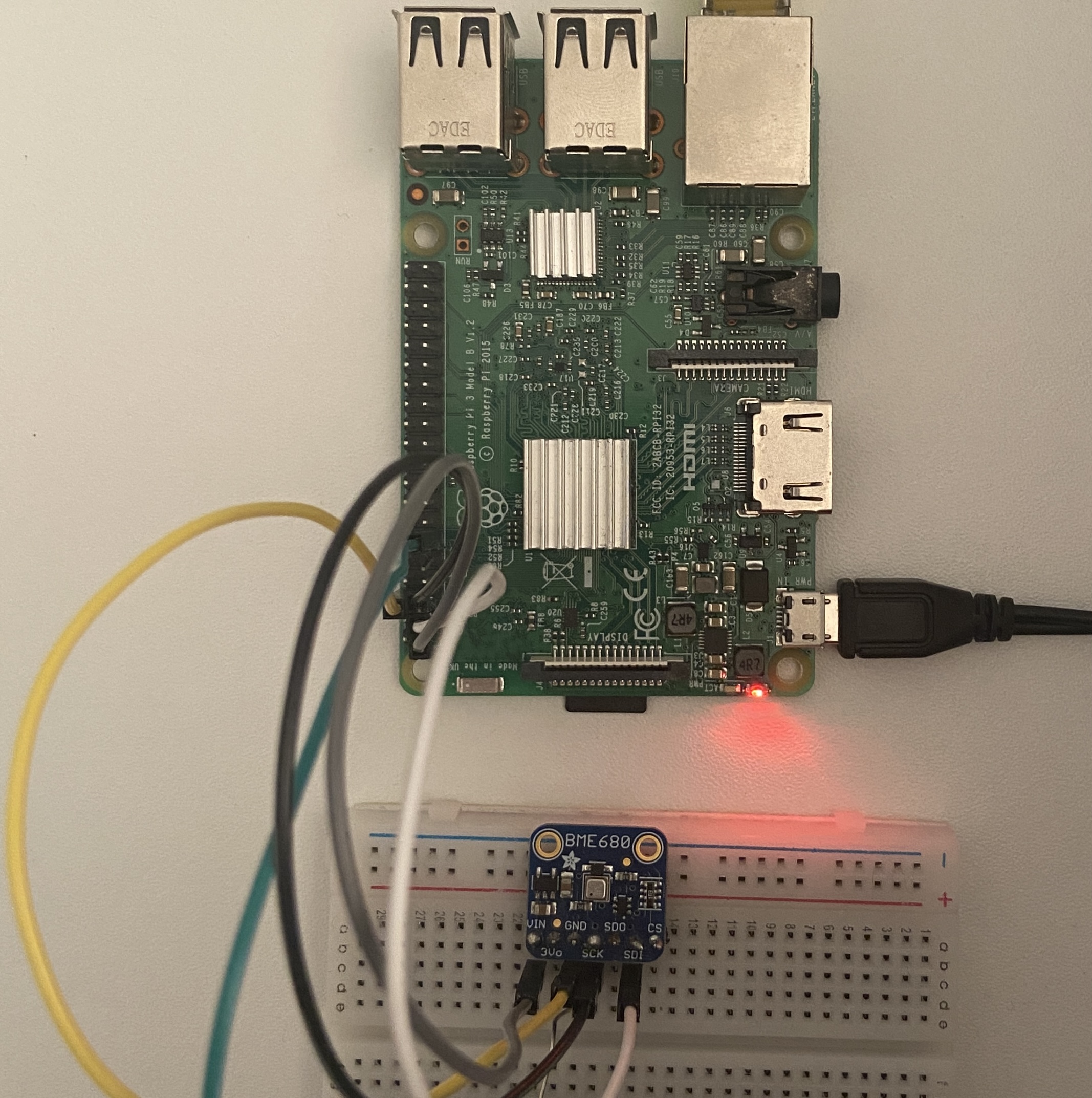


Abb. 5: Versuchsaufbau nach Verbindung von Raspberry Pi und Sensor

## Konfiguration der I2C-Schnittstelle

Um den Sensor verwenden zu können, muss zunächst das I2C Interface aktiviert werden. Dies erfolgt mit dem Befehl "raspi-config nonint do\_i2c 0"*.* Anschließend muss das Softwarepaket für das I2C Interface mittels "apt install -y python3-smbus i2c-tools" installiert werden.

Nun sollte getestet werden, ob das Vorgehen erfolgreich war und der Sensor erreichbar ist. Mittels "lsmod | grep i2c\_" erhalten wir alle geladenen Module. Das Ergebnis sollte wie folgt aussehen.

Module Size Used by

i2c\_bcm2835 16384 0

i2c\_dev 20480 0

Mit "i2cdetect -y 1" können wir uns den Sensor anzeigen (77) lassen.

0 1 2 3 4 5 6 7 8 9 a b c d e f

00: -- -- -- -- -- -- -- -- -- -- -- -- --

10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --

70: -- -- -- -- -- -- -- 77

Nachdem der Sensor korrekt angeschlossen und die Module geladen sind, kann mit dem auslesen der Werte des Sensors begonnen werden.

## Auslesen der Sensordaten

Um Daten vom BME680-Sensor zu erhalten, wird die Adafruit-Bibliothek benötigt. Zunächst müssen wir das Paketverwaltungstool pip installieren, falls es noch nicht vorhanden ist. Dies kann mit dem Befehl apt install python3-pip -y im Terminal durchgeführt werden.

Sobald pip installiert ist, können wir das Python-Paket adafruit-circuitpython-bme680 mithilfe des Befehls pip3 install --user adafruit-circuitpython-bme680 installieren. Das --user-Flag ermöglicht die Installation des Pakets für den aktuellen Benutzer. Dadurch werden keine Administratorrechte benötigt.

Nachdem die Adafruit-Bibliothek erfolgreich installiert wurde, können wir die Sensorwerte auslesen. Hierzu verwenden wir das vorliegende Python-Programm. Der Code ermöglicht die Kommunikation mit dem BME680-Sensor über den I2C-Bus.

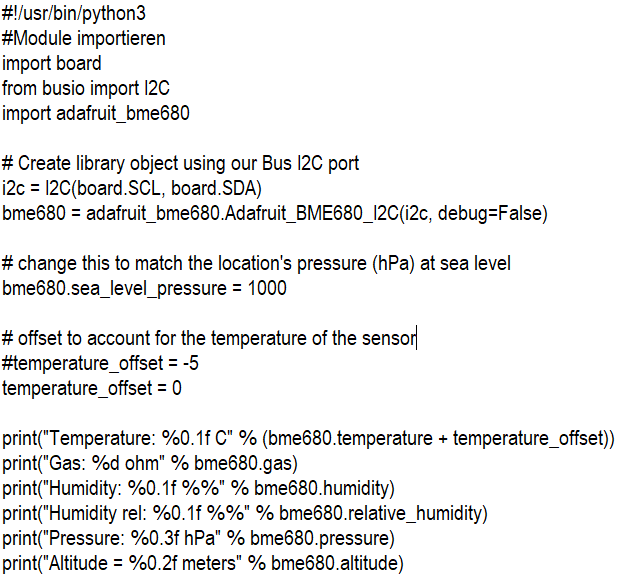
Der erste Teil des Codes importiert die erforderlichen Module und erstellt ein I2C-Objekt, das die Kommunikation über den I2C-Bus ermöglicht. Anschließend wird ein BME680-Objekt erstellt, das den Zugriff auf die Sensorfunktionen bereitstellt. Durch die Zuweisung eines Wertes zur sea\_level\_pressure-Eigenschaft kann der Druck auf Meereshöhe für den aktuellen Standort festgelegt werden.

Um genaue Temperaturwerte zu erhalten, kann ein Temperatur-Offset hinzugefügt werden. Dieser Offset wird üblicherweise verwendet, um die Abweichungen des Sensors zu korrigieren. Es wird empfohlen, einen separaten Temperatursensor zur Kalibrierung zu verwenden. In diesem Beispiel wird der temperature\_offset auf 0 gesetzt.

Schließlich werden die gemessenen Werte des Sensors ausgelesen und mit Hilfe von print() auf der Konsole ausgegeben. Die Formatierung erfolgt mithilfe von Platzhaltern (%-Operator), um die entsprechenden Werte einzufügen.

Durch Ausführen des Python-Programms erhalten wir die aktuellen Temperatur-, Gas-, Feuchtigkeits-, Druck- und Höhenwerte des BME680-Sensors.

Bitte beachten Sie, dass die genaue Funktionsweise und Verwendung des BME680-Sensors und der Adafruit-Bibliothek in der offiziellen Dokumentation zu finden sind. Diese detailliertere Anleitung dient lediglich zur Erläuterung des bereitgestellten Codes.



Code 1: Python Skript zur Abfrage der Sensordaten

Der Output sieht nun folgendermaßen aus.

Temperature: 23.0 C

Gas: 1912 ohm

Humidity: 44.8 %

Pressure: 989.394 hPa

Altitude = 89.86 meters

## Konfiguration der HTTP-Schnittstelle

## Konfiguration des Webservers

## Senden der Daten an den Webserver

Nach Erfassung der Sensordaten sollen diese an eine Webserverschnittstelle gesendet, um sie in einer Webanwendung zu visualisieren. Die Kommunikation erfolgt über das HTTP-Protokoll.

Zunächst wird die URL des Webservers definiert, an die die Daten gesendet werden sollen. In diesem Fall ist die URL 'http://141.147.6.122:8080/parameters'.

Dann werden die Sensordaten behandelt. Dabei werden die Temperatur, das Gas, die Luftfeuchtigkeit, die relative Luftfeuchtigkeit, der Druck und die Höhe direkt aus den entsprechenden Variablen des Sensors (z.B. bme680.temperature) übernommen.

Anschließend wird ein JSON-Datenobjekt erstellt, das die erfassten Daten enthält. Zusätzlich wird ein Zeitstempel generiert, der den Zeitpunkt der Erfassung angibt. Das JSON-Datenobjekt sieht wie folgt aus:

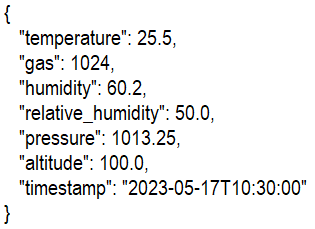
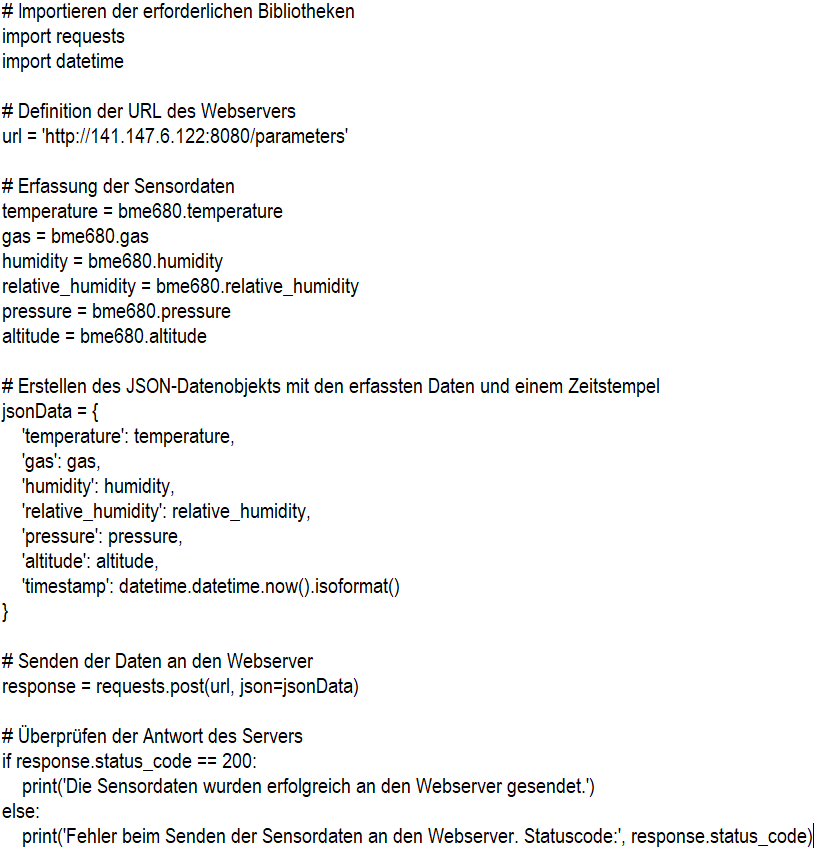


Abb. 6: JSON-Format

Schließlich werden die Daten mit Hilfe eines HTTP-POST-Requests an den Webserver gesendet. Dabei wird das JSON-Datenobjekt als Payload des Requests übermittelt.

Nach dem Senden der Daten wird die Antwort des Servers überprüft. Wenn der Statuscode der Antwort 200 ist, war der Request erfolgreich und die Sensordaten wurden erfolgreich an den Webserver gesendet. Andernfalls wird eine Fehlermeldung ausgegeben, zusammen mit dem erhaltenen Statuscode.



Code 2: Skript zum Senden der Daten an den Webserver

## Automatisierung der Datenabfrage und -sendung

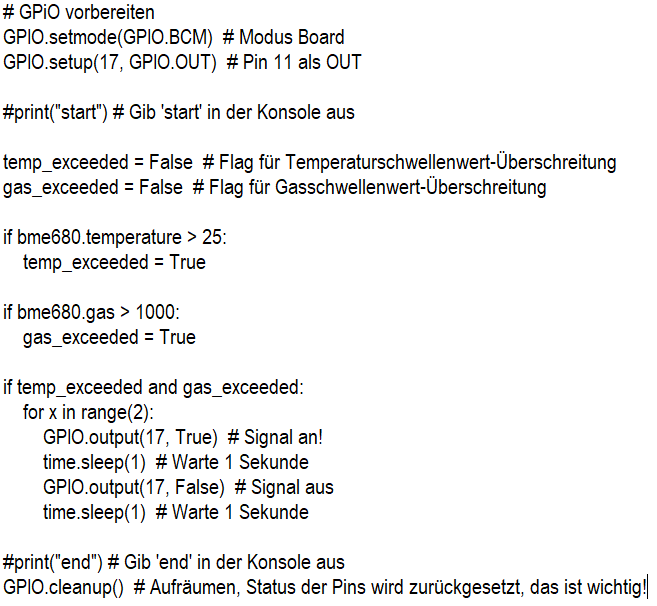
Um die automatisierte Ausführung der Skripte zur Datenabfrage und dem Senden zu steuern, wird der Cron-Dienst verwendet. Cron überprüft regelmäßig im Hintergrund, ob Aufträge vorhanden sind, die ausgeführt werden können, und führt diese dann aus. In diesem Fall soll der Nutzer die Skripte alle 5 Minuten laufen lassen, und um dies dem Cron mitzuteilen, werden sogenannte Crontabs verwendet.

Eine Crontab ist eine Tabelle, die alle erforderlichen Informationen für den Cron enthält. In diesem Fall würde die Crontab beispielsweise den Eintrag "\*/5 \* \* \* \* /usr/local/sbin/bme680.py" enthalten. Diese Angabe legt fest, dass das Skript "/usr/local/sbin/bme680.py" alle 5 Minuten ausgeführt wird.

Die Crontab besteht insgesamt aus sechs Feldern, die jeweils durch Leerzeichen getrennt sind. Die ersten fünf Felder bestimmen den Zeitpunkt der Ausführung, während das sechste Feld den auszuführenden Befehl enthält. Somit bedeutet der oben genannte Eintrag, dass das Skript "bme680.py" alle 5 Minuten an jedem Tag ausgeführt wird.

## Erweiterung des Versuchsaufbau um eine LED

Nachdem die Sensordaten erfolgreich erfasst wurden, soll nun eine automatisierte Reaktion auf bestimmte Ereignisse implementiert werden. Der vorliegende Code stellt eine Lösung dar, um auf die erfassten Sensordaten zu reagieren und die gewünschte Aktion auszuführen.



Code 3: Ansteuerung der roten LED

Zu Beginn des Codes wird die Vorbereitung des General Purpose Input/Output (GPIO) durchgeführt. Hierbei wird der Modus des GPIO auf den Board-Modus (BCM) gesetzt und der Pin 17 als Output definiert. Dieser Pin wird verwendet, um die rote LED zu steuern, die später beim Überschreiten der Schwellenwerte blinken soll.

Anschließend werden Flags initialisiert, um die Überschreitung der Schwellenwerte für Temperatur und Gas zu verfolgen. Die Flags "temp\_exceeded" und "gas\_exceeded" werden zunächst auf False gesetzt.

Im nächsten Schritt erfolgt die Überprüfung der Sensordaten auf Überschreitung der Schwellenwerte. Wenn die Temperatur den Schwellenwert von 25 Grad Celsius überschreitet, wird das Flag "temp\_exceeded" auf True gesetzt. Entsprechend wird das Flag "gas\_exceeded" auf True gesetzt, wenn der Gaswert den Schwellenwert von 1000 überschreitet.

Wenn sowohl das Flag "temp\_exceeded" als auch das Flag "gas\_exceeded" auf True stehen, wird eine Schleife gestartet, die die rote LED zum Blinken bringt. Dies wird erreicht, indem der GPIO-Pin 17 abwechselnd auf High (True) und Low (False) gesetzt wird. Die Schleife wird insgesamt zweimal durchlaufen, wodurch ein Blinkmuster erzeugt wird.

Abschließend wird der Zustand der Pins durch den Aufruf von "GPIO.cleanup()" zurückgesetzt, um eine ordnungsgemäße Bereinigung des GPIO-Status sicherzustellen.

## Austausch des BME680 gegen den SCD30 Sensor

Wie zuvor bereits erwähnt, wurde der BME680 im Laufe des Projekts durch den SCD30 ausgetauscht, da dieser in der Lage ist, genaue CO2-Werte zu messen. Die Einrichtung des Sensors erfolgte ähnlich wie zuvor beschrieben, jedoch wurden anstelle der BME680-Bibliotheken nun die entsprechenden Bibliotheken für den SCD30 verwendet.

Die Verkabelung des SCD30-Sensors mit dem Raspberry Pi erfolgte wie folgt:

|  |  |
| --- | --- |
| **Raspberry Pi GPIO Pin** | **Adafruit SCD30 Pin** |
| Pi 3V | VIN |
| Pi GND (Ground) | GND |
| Pi SCL | SCL |
| Pi SDA | SDA |

Tab. 3: Pinbelegung SCD30

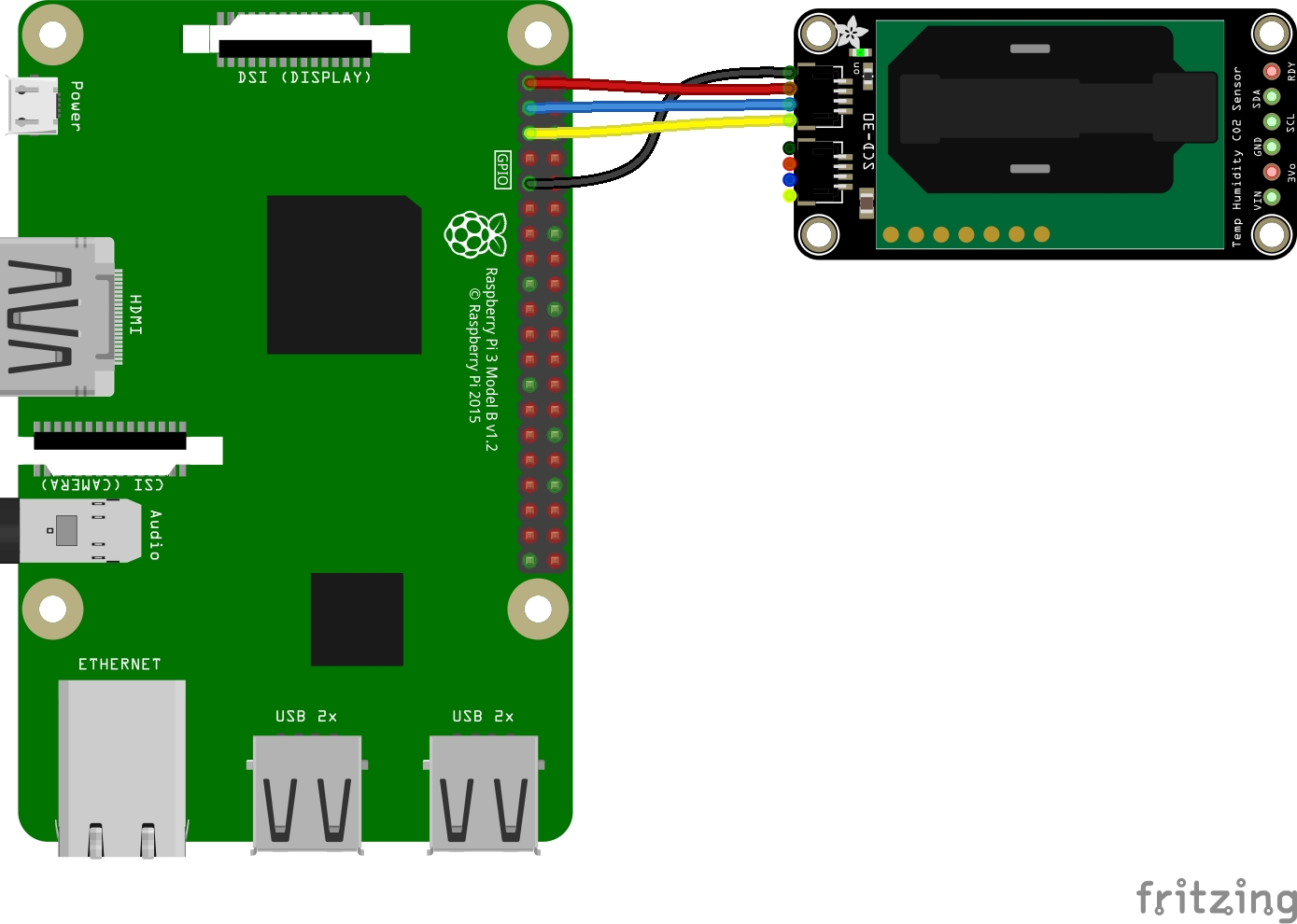
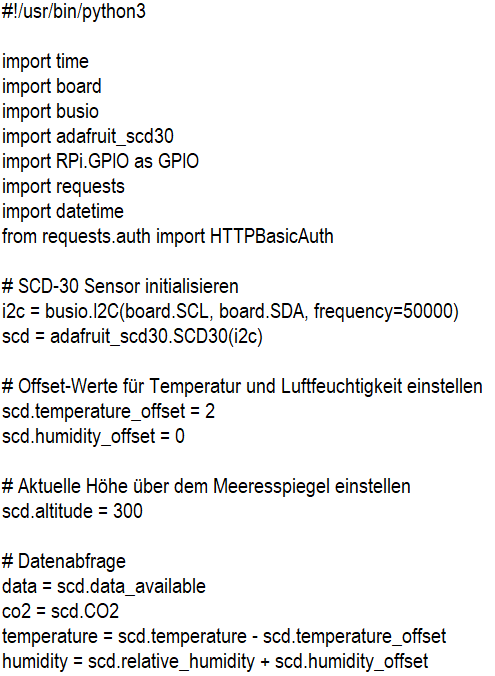


Abb. 7: Pinbelegung SCD30

Der Code zur Datenabfrage wurde angepasst.



Code 4: Datenabfrage SCD30

## Konfiguration der Telegram-Schnittstelle

## Entwicklung einer Webanwendung

# Quellenverzeichnis

[Balzert+2011] Balzert, H.; Schröder, M.; Schäfer, C.; Wissenschaftliches Arbeiten*,*   
2. Auflage, W3L GmbH Herdecke Witten, 2011.

[Catalog 2019] Catalog Library of Congress <http://catalog.loc.gov/> (Zugriff 28.05.2019).

[Duden 2019] Bibliographisches Institut Dudenverlag <http://www.duden.de>  
(Zugriff 28.05.2019).

[IDS2011] Dokumente zu den Inhalten der Rechtschreibreform, Überarbeitetes Regel Werk 2011 <http://www.ids-mannheim.de/reform/regeln2006.pdf> (Zugriff 28.05.2019).

[IEEE2018] IEEE Zitierstil und Manuskript Vorlage   
<https://www.ieee.org/conferences/publishing/templates.html>   
(Zugriff: 17.10.2019)

[HSCoburg2018] Regeln guter wissenschaftlicher Praxis an der Hochschule Coburg, Hochschule Coburg, 09.11.2018, <https://www.hs-coburg.de/fileadmin/hscoburg/Forschung/Regeln_guter_wissenschaftlicher_Praxis_Hochschule_Coburg_12_2018.pdf> (Zugriff 17.10.2019)

[Katalog 2019] Katalog der Deutschen Nationalbibliothek <https://portal.dnb.de/opac.htm> (Zugriff 28.05.2019).

[Schmidt+2008] Kories, R.; Schmidt-Walter, H.: Taschenbuch der Elektrotechnik, 8. Aufl., Verlag Harri Deutsch, Frankfurt, 2008.

# Glossar

Artefakt „(bildungssprachlich) etwas von Menschenhand Geschaffenes, (Elektronik) Störsignal" [Duden 2014]; in dieser Arbeit synonym zu Erzeugnis verwendet

Monographie “Einzelschriften, welche sich thematisch abgeschlossen mit einem einzigen Gegenstand beschäftigen. Im Gegensatz hierzu thematisieren Sammelbände mehrere Gegenstände aus einem einzelnen Themengebiet“ [Balzert+11, S. 199]

<Nachfolgenden Abschnittwechsel bitte nicht entfernen!>

1. Formate für das gesamte Dokument

# Ehrenwörtliche Erklärung

Ich versichere hiermit, dass ich meine/n *Praxisbericht/Bachelorarbeit/Masterarbeit* mit dem Titel

|  |
| --- |
|  |

selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie nicht an anderer Stelle als Prüfungsarbeit vorgelegt habe.

|  |  |  |
| --- | --- | --- |
| Ort |  |  |
| Datum |  | Unterschrift |