



BOSCH
Technik fürs Leben



Aufbau eines Fernüberwachungssystems eines Räucherofens

Bachelor of Engineering

des Studiengangs Elektrotechnik
an der Dualen Hochschule Baden-Württemberg Stuttgart

von

Stefan Gerhartz und Paul Schmidt

23. Januar 2022

Bearbeitungszeitraum	27.09.2021-23.01.2021
Matrikelnummer, Kurs	8463838 (SG), 3308538 (PS); TEL19GR3
Hochschule	Duale Hochschule Baden-Württemberg
Betreuer	Prof. Dr. -Ing. Johannes Moosheimer

Duale Hochschule Baden Württemberg, Stuttgart

Fakultät Technik - Studiengang

Bericht über Studienarbeit an der dualen Hochschule Baden-Württemberg

5. Studienhalbjahr

Name des Studierenden: Stefan Gerhartz, Paul Schmidt

Studienjahrgang: 2019

vom: 27.09.2021

bis: 23.01.2022

Thema: Aufbau eines Fernüberwachungssystems eines Räucherofens

Betreuer: Johannes Moosheimer

Stellungnahme des Betreuers:

Dieser Bericht wurde geprüft und ist sachlich und fachlich richtig.

Ort

Datum

Unterschrift

Selbstständigkeitserklärung des Studenten:

Ich versichere hiermit, dass ich meine mit dem Thema: selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt, falls beide Fassungen gefordert sind.

Ort

Datum

Unterschrift

Ort

Datum

Unterschrift

KURZFASSUNG

Die Automatisierung von Gegenständen des täglichen Lebens schreitet immer weiter voran (IOT). Ziel ist die Vereinfachung der Bedienung dieser Gegenstände und eine Effizienzsteigerung. Diese Studienarbeit beschreibt die Verbindung des Konzepts IOT mit einem Räucherofen zur Veredelung von Lebensmitteln.

Die Idee ist, den Räucherprozess fern zu überwachen und damit in der Handhabung komfortabler und zeitsparender zu machen. Der Fokus der Studienarbeit liegt auf der Sammlung relevanter Daten des Prozesses und dem Senden an ein mobiles Endgerät. Auf diesem Endgerät sollen die Daten jederzeit abrufbar sein.

Zunächst werden die Erfassung der Daten und das Senden an ein Endgerät getrennt voneinander entwickelt. Die Sensoren werden in diesem Bericht ausgewählt und die Funktionen überprüft. Es wird gezeigt, dass die Sensoren über den Mikrocontroller ausgelesen werden können. Ebenfalls wird das Vorgehen beim Senden von Daten erläutert. Der Datentransfer vom Mikrocontroller an die Internetseite, sowie das Herunterladen in die App wird beschrieben.

Die Ergebnisse können dazu genutzt werden, eine finale Version für das System zu entwickeln.

INHALTSVERZEICHNIS:

ABKÜRZUNGSVERZEICHNIS	3
ABBILDUNGSVERZEICHNIS	4
1. EINLEITUNG	6
2. AUFGABENSTELLUNG	7
2.1 Problemstellung	7
2.2 Aufgabe	7
3. KONZEPT.....	8
3.1 Räucherofen	8
3.2 Technik am Ofen	9
3.3 Hand-Held-Gerät	9
4. VORGEHENSWEISE	10
5. STAND DER TECHNIK	11
5.1 Aufbau eines Räucherofens	11
5.2 Konzepte des Räucherns	12
5.2.1 Heißräuchern	12
5.2.2 Warmräuchern	12
5.2.3 Kalträuchern	12
6. HARDWARE.....	13
6.1 Anforderungen	13
6.1.1 Allgemeine Platzierung	13
6.1.2 Temperatur	15
6.1.3 Feuchtigkeit	15
6.1.4 Partikelmessung	16
6.1.5 Feuererkennung.....	16
6.1.6 Tür Zustandserkennung	16
6.1.7 Display und Taster	16
6.1.8 μC	17
6.2 Auswahl	18
6.2.1 Allgemein.....	18
6.2.2 Temperatur & Feuchtigkeit.....	18
6.2.3 Partikelmessung	18

6.2.4	Feuererkennung.....	19
6.2.5	Tür Zustandserkennung.....	19
6.2.6	Display und Taster	20
6.2.7	µC Auswahl.....	20
6.3	Testaufbau Sensorfunktion	21
6.3.1	Ziel und Vorgehensweise	21
6.3.2	Testschaltplan.....	22
6.3.3	Hardwareaufbau	23
6.3.4	Testläufe.....	24
7.	SOFTWARE.....	25
7.1	µC Testsoftware Sensordaten	26
7.2	µC Testsoftware Wifi.....	27
7.2.1	Setup.....	27
7.2.2	Loop.....	29
7.2.3	HTTP-Kommunikation.....	31
7.2.4	Port-Forwarding und dynamisches DNS	33
7.3	Smartphone App	35
7.3.1	Allgemeines	35
7.3.2	Main Activity.....	36
7.3.3	Downloader	40
8.	ZUSAMMENFASSUNG.....	42
9.	AUSBLICK	43
10.	ANHANG	44

ABKÜRZUNGSVERZEICHNIS

<i>Abk</i>	<i>Abkürzung</i>
<i>API</i>	<i>Application Programming Interface (Programmierschnittstelle)</i>
<i>App</i>	<i>Applikation (meist als Software Applikation für Smartphones)</i>
<i>DHBW</i>	<i>Duale Hochschule Baden-Württemberg</i>
<i>DNS</i>	<i>Domain Name System</i>
<i>FIFO</i>	<i>First in First out (Speicher)</i>
<i>GUI</i>	<i>Graphical User Interface</i>
<i>HTML</i>	<i>Hypertext Markup Language</i>
<i>HTTP</i>	<i>Hypertext Transfer Protocol</i>
<i>IoT</i>	<i>Internet of things</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>LCD</i>	<i>Light Crystal Display</i>
<i>MAC</i>	<i>Media-Access-Control</i>
<i>μC</i>	<i>Micro Controller</i>
<i>SPI</i>	<i>Serial Peripheral Interface</i>
<i>SSID</i>	<i>Service Set Identifier</i>
<i>UI</i>	<i>User Interface</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
<i>USB</i>	<i>Universal Serial Bus</i>
<i>WiFi</i>	<i>Wireless Fidelity</i>
<i>WIFI</i>	<i>Wireless Local Area Network</i>

ABBILDUNGSVERZEICHNIS

ABBILDUNG 1: KONZEPT DES VOLLSTÄNDIG ERWEITERTEN SYSTEMS RÄUCHEROFEN.....	8
ABBILDUNG 2: SCHEMATISCHER AUFBAU EINES RÄUCHEROFENS.	11
ABBILDUNG 3: AUFBAU EINES TYPISCHEN RÄUCHEROFENS. NUMMERIERTE PUNKTE MARKIEREN STELLEN, AN DENEN HARDWARE PLATZIERT WERDEN SOLL.	14
ABBILDUNG 4 SCHALTPLAN DES STECKBRETTAUFBAUES FÜR DIE TESTLÄUFE DER SENSOREN	22
ABBILDUNG 5 STECKBRETTAUFBAU FÜR DIE TESTLÄUFE DER SENSOREN	23
ABBILDUNG 6: AUFBAU DER STUDIENARBEIT AUFGETEILT NACH DEM GRUNDKONZEPT SIEHE KAPITEL 3; DURCH DIE PFEILE IST DER DATENSTROM ZWISCHEN DEN KOMPONENTEN DARGESTELLT.....	25
ABBILDUNG 7: MENÜFÜHRUNG DER SOFTWARE FÜR DEN SENSORTEST.....	26
ABBILDUNG 8: ABLAUF DER SETUP FUNKTION DER WIFI TESTSOFTWARE; NACH DEM START DER SERIELLEN KOMMUNIKATION, WIRD DIE WIFI-VERBINDUNG HERGESTELLT UND ANSCHLIEßEND DER WEBSERVER GESTARTET.	28
ABBILDUNG 9: ABLAUF DER LOOP FUNKTION DER WIFI TESTSOFTWARE; NACH DEM ABFRAGEN NEUER DATEN, WERDEN DIE WIFI-VERBINDUNG, DER STATUS DES WEBSERVERS UND DAS VORHANDENSEIN VON CLIENTS ABGEFRAGT.	29
ABBILDUNG 10: PRINZIP EINES FIFO-SPEICHERS; NEUE DATEN (LINKS) WERDEN AN ERSTE FREIE STELLE GESCHRIEBEN; WERDEN ELEMENTE AUS DEM SPEICHER GENOMMEN, WERDEN SIE VON VORNE (RECHTS) GENOMMEN.	30
ABBILDUNG 11: ANTWORT-KOPFZEILE AN CLIENT NACH HTTP-REQUEST.....	31
ABBILDUNG 12: PRINZIPIELLER AUFBAU DER HTTP_COMMUNICATION() FUNKTION; NACH DEM VOLLSTÄNDIGEN AUSLESEN DER ANFRAGE DES CLIENTS, WIRD MIT EINEM STATISCHEN ANTWORT HEADER UND ANSCHLIEßEND DEM ZUSAMMENGEBAUTEN HTML QUELLCODE INKLUSIVE ALLER GEWÜNSCHTEN DATEN GEANTWORTET UND DIE VERBINDUNG GETRENNT.	32
ABBILDUNG 13: DARSTELLUNG DER HTTP-ANTWORT DES µC IN EINEM BROWSER. AUFGERUFEN ÜBER DIE LOKALE IP-ADRESSE, DIE LETZTEN SENSORDATEN OBERHALB DER TABELLE ALLER GESPEICHERTER DATEN.	33
ABBILDUNG 14: BEISPIEL DES PORT-FORWARDINGS ANHAND EINER ANFRAGE AUS DEM INTERNET AN EINEN SPEZIELLEN PORT DES ROUTERS.....	34
ABBILDUNG 15: NETZWERKEINSTELLUNGEN FÜR EINEN VIRTUELLEN SERVER; AUFRUFE DES NETZWERKES WERDEN STANDARDMÄßIG AN DIE SPEZIFIZIERTE IP-ADRESSE AN PORT 80 WEITERGELEITET.....	35
ABBILDUNG 16: PRINZIPIELLER ABLAUF DER MAIN ACTIVITY DER SMARTPHONE APP; NACH DEM START DER ANWENDUNG WERDEN DOWNLOAD, EXTRAHIERUNG UND DARSTELLUNG DER DATEN ZYKLISCH ÜBERPRÜFT.	37
ABBILDUNG 17: DEFINITION DER SCHNITTSTELLEN ZWISCHEN REFRESH BUTTON (OBEN) UND DES TIMERS (UNTEN) MIT DER STARTDOWNLOAD() METHODE	37
ABBILDUNG 18: SCREENSHOT DES ANDROID EMULATORS (API 26) NACH ERFOLGREICHEM HERUNTERLADEN UND DECODIEREN DES WEBSITE QUELLCODES. TEXTUELLE AUSGABE DER LETZTEN DATEN (OBERER BILDHÄLFTE); ZEITLICHER VERLAUF DER TESTDATEN ZU	

TEMPERATUR (ROT) UND FEUCHTIGKEIT (BLAU) IN DEN LETZTEN 80 S (UNTERE BILDHÄLFTE).	39
ABBILDUNG 19: PRINZIPIELLER ABLAUF DES HERUNTERLADENS DER WEBSITE IN EINEM HINTERGRUNDTHREAD UND RÜCKGABE DES SOURCECODES DER WEBSITE AN DIE MAIN ACTIVITY.....	41
ABBILDUNG 20: VERTEILUNG DER ANDROID VERSIONEN / API LEVEL MIT ANTEIL DER GERÄTE DIE DIESE ANWENDUNGEN NUTZEN KÖNNEN AUS DER ANDROIDSTUDIO APP.....	44
ABBILDUNG 21 DATENBLATT FLAMMSENSOR.....	45
ABBILDUNG 22 DATENBLATT HALL-SENSOR.....	46
ABBILDUNG 23 DATENBLATT TEMPERATUR- UND FEUCHTIGKEITSSENSOR.....	47

1. EINLEITUNG

Im Zeitalter der Digitalisierung spielt die zunehmende Vernetzung verschiedenster Gegenstände eine wichtige Rolle. Die Anfänge aus der Industrie wurden nun in den privaten Nutzungsbereich übernommen. Mehr und mehr Geräte werden über das Internet verbunden. Dieser IOT-Gedanke spielt in der folgenden Arbeit eine zentrale Rolle.

Die Vernetzung und damit auch die einhergehende Automatisierung ist ein wesentlicher Bestandteil der technischen Entwicklungen. Der Fokus liegt hierbei auf verschiedenen Bereichen. Zum einen werden mehr Funktionen im Bereich der Sicherheitstechnik entwickelt, zum anderen werden verschiedene Gegenstände automatisiert, um die Nutzung komfortabler zu gestalten.

Dies ist der Fokus der folgenden Arbeit. Die Anbindung eines herkömmlichen Räucherofens an das Internet, um eine Überwachung aus der Ferne zu realisieren. Somit ist eine ständige Kontrolle, sowie eine Anwesenheit bei einem Räucherprozess nicht mehr nötig. Diese Studienarbeit beinhaltet eine Prüfung eines Konzeptes für die Fernüberwachung. Der Fokus wird auf die Auswahl der Hardware und einer geeigneten Kommunikation in das Internet gelegt. Die Funktion wird in Tests separat voneinander bestätigt.

2. AUFGABENSTELLUNG

In dem folgenden Kapitel werden die existierenden Probleme des Veredelns von Lebensmitteln und die korrespondierende Aufgabenstellung zu diesen Problemen erläutert.

2.1 Problemstellung

Um Fleischwaren zu veredeln, muss die Rauchentwicklung innerhalb eines Räucherofens möglichst konstant gehalten werden. Dieser Prozess muss kontinuierlich überwacht werden. Durch die glühenden Späne sind sowohl die Temperatur als auch die Rauchentwicklung ein entscheidender Faktor. Ein ausbrechendes Feuer ist unbedingt zu vermeiden. Dies muss durch zeitlich getaktetes Öffnen des Ofens und Inaugenscheinnahme geschehen. Hierzu muss ein Mensch ständig vor Ort sein. Eine Überwachung aus der Ferne ist nicht möglich. Ebenfalls funktioniert eine Verstellung der Luftzufuhr nur vor Ort. Diese beeinflusst die Rauchentwicklung hauptsächlich. Es gibt keine Aktoren, die dies aus der Ferne erlauben. Zusätzlich ist eine Dokumentation der Daten sehr zeitaufwändig.

2.2 Aufgabe

Das bestehende System des Räucherofens soll durch eine Rauch- und Feuer-Fernüberwachung ergänzt werden. Die Ausgabe der erfassten Daten soll über das Internet möglich sein (IoT-Ansatz) und damit von jedem beliebigen Ort erfolgen können. Eine geeignete Datenvisualisierung an Hand-Held-Geräten (wie Smartphones) ist bereitzustellen.

Im ersten Schritt müssen die passenden Sensoren ausgewählt werden. Es muss ein Micro-Controller (kurz μC) gefunden werden, welcher diese steuern und gleichzeitig die Daten über das Internet übertragen kann.

3. KONZEPT

Dieses Kapitel beschreibt das grundlegende Konzept, um das vorliegende System Räucherofen entsprechend der Aufgaben- und Problemstellung zu erweitern. In Abbildung 1 ist ein solches Konzept gezeigt. Zu erkennen ist die Aufteilung des Systems in drei Teile, die in den folgenden Abschnitten näher erläutert werden. Diese Teilabschnitte sind über Pfeile miteinander verbunden, die die Daten darstellen, die zwischen den Teilsystemen versandt werden müssen.

In der vorliegenden Studienarbeit wird nur auf die Umsetzung der Kommunikation der Teilsysteme in einer Richtung eingegangen (von links nach rechts). Eine Implementierung der Steuerung von Aktoren sowohl vor Ort als auch aus der Ferne wird nicht betrachtet.

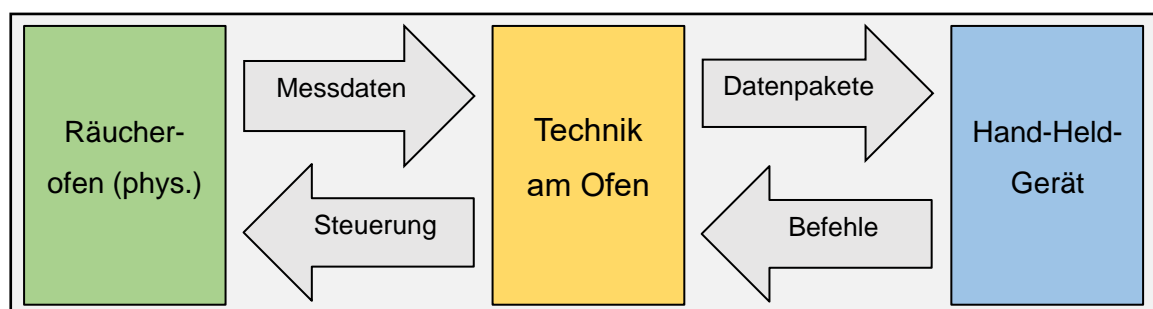


Abbildung 1: Konzept des vollständig erweiterten Systems Räucherofen.

3.1 Räucherofen

Wie in Abbildung 1 zu sehen ist, befindet sich das Teilsystem Räucherofen (physisch) am linken Rand des Gesamtsystems. Es befindet sich mit dem Teilsystem Technik am Ofen in direktem Datenaustausch, da die verbaute Sensorik ihre Messdaten an den μC weitergibt. Als messbare Größen sind die Temperatur, die Feuchtigkeit und die Partikeldichte (Rauchentwicklung) vorhanden. Außerdem ist eine Flammenerkennung erforderlich und ein Sensor, der den Zustand der Tür (offen/geschlossen) beobachtet.

3.2 **Technik am Ofen**

Die Technik am Ofen befindet sich in der Mitte zwischen dem physischen Räucherofen und dem Hand-Held-Gerät. Dadurch, dass die Sensoren kontinuierlich Messdaten weitergeben, kann der μC bei auffälligen Werten Meldungen ausgeben. Dies soll direkt über ein Nutzerinterface stattfinden. Außerdem soll der μC als Datenspeicher fungieren und die Messdaten aufbereitet über das Internet an ein Hand-Held-Gerät weitergeben

3.3 **Hand-Held-Gerät**

Das Hand-Held-Gerät befindet sich an der rechten Seite des Gesamtsystems. Es soll auf die Datenvisualisierung ein Hauptaugenmerk gelegt werden. Um dies zu erreichen, wird neben der Erreichbarkeit einer Website über einen Browser die Software für eine Smartphone Applikation bereitgestellt. Dadurch erhält der Endnutzer die Daten komfortabler und besser aufbereitet. Um dies zu erreichen wird die gesamte Website, die von dem μC bereitgestellt wird heruntergeladen, die Sensordaten werden extrahiert und anschließend sowohl in Textform als auch in einem Graphen dargestellt.

4. VORGEHENSWEISE

Dieser Abschnitt beschreibt das geplante Vorgehen, um die unter Kapitel 2.2 beschriebene Aufgabenstellung zu behandeln. Es wird auf das aufgestellte Konzept (Seite 8) zurückgegriffen, um Teile der Arbeitsschritte abgrenzen zu können.

In einem ersten Schritt findet die Bestimmung der Anforderungen an die Hardware statt. Dabei werden die Anforderungen der Aufgabe, sowie die Eigenschaften des bestehenden Systems Räucherofen mit einbezogen. Hierbei wird der Fokus auf die Auswahl der Bauteile in Abhängigkeit des Einsatzgebietes gelegt. Diese ergeben sich in erster Linie aus den baulichen Bedingungen, sowie den Vorgängen des Räucherprozesses und dessen Parametern. Als besonders wichtiger Parameter ist die teils hohe Temperatur in der Räucherkammer zu nennen.

In einem nächsten Schritt wird die ausgewählte Hardware in einem Testaufbau auf ihre Funktionalität geprüft.

Parallel zum Vorgehen der Hardwareauswahl werden die nötigen Anforderungen der Software ermittelt. Diese werden maßgeblich durch den Einsatz des IoT-Gedankens bestimmt. Es wird ein Gesamtkonzept erarbeitet, welches im Anschluss in Testversionen umgesetzt wird. Diese Softwareapplikationen stellen die Schnittstellen zwischen den Teilsystemen des Konzeptes dar (Pfeile in Abbildung 1).

Dazu werden zunächst zwei separate Applikationen für den Micro-Controller entwickelt. Das eine Programm wird für die Ansteuerung einer Testversion der Menüführung dienen. Das zweite beschäftigt sich mit der Konnektivität zwischen dem Micro-Controller und Clients über das Internet.

In einem zukünftigen Schritt werden diese beiden Stränge zusammengeführt. Dafür wird ein ganzheitliches System entwickelt und auf dem Micro-Controller implementiert.

Eine dritte, unabhängige Software beschäftigt sich mit der Datenvisualisierung auf einem Smartphone per App.

5. STAND DER TECHNIK

Die anschließenden Abschnitte sollen einen groben Eindruck über den aktuellen Stand der Technik vermitteln. Dabei wird einerseits der physische Aufbau eines Räucherofens und andererseits die verschiedenen Methoden des Räucherns behandelt.

5.1 Aufbau eines Räucherofens

In Abbildung 2 ist die skizzierte Abbildung eines Räucherofens zu sehen. Hier sind alle Bestandteile eines solchen Systems aufgeführt. Besonders hervorzuheben ist die Position, des zu räuchernden Objektes, die Tür und die Position der Räucherschale. Dies sind die verschiedenen Punkte, an denen eine Messung der Werte stattfinden soll.

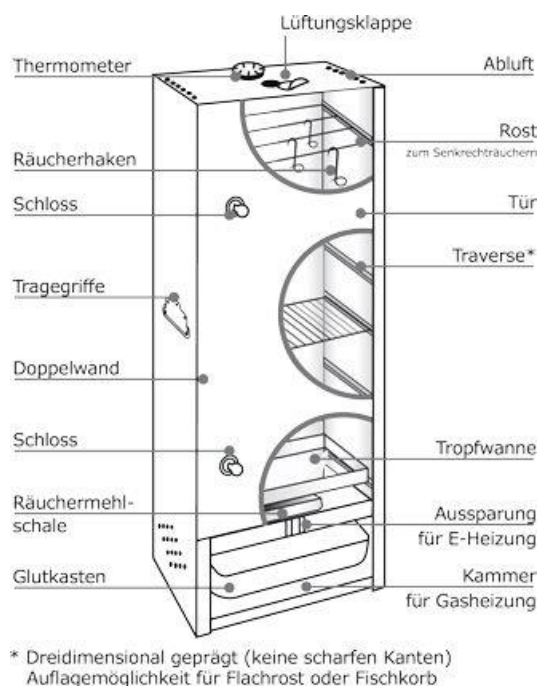


Abbildung 2: Schematischer Aufbau eines Räucherofens.

In der Räucherschale werden Holzspäne zum Glimmen gebracht. Der hieraus entstehende Rauch steigt nach oben und zieht dabei an den Lebensmitteln vorbei. Durch diesen Prozess findet eine Veredelung statt. Die Lebensmittel erhalten einen speziellen rauchigen Geschmack. Um dies zu variieren, kann man die Holzspäne, welche auch Holzschnitzel genannt werden, austauschen.

Maßgeblich für den Prozess ist eine konstante Temperatur und Glutentwicklung. Um dies sicherzustellen wird eine Tropfschale zwischen die Späne und die Lebensmittel geschoben, diese verhindert ein Löschen der Glut durch herabtropfende Flüssigkeit.

5.2 Konzepte des Räucherns

Der folgende Abschnitt soll einen Überblick über die [verschiedenen Methoden](#) des Räucherns geben. Dafür wird auf die Methoden des Heiß-, Warm- und Kalträucherns eingegangen.

5.2.1 Heißräuchern

Das Heißräuchern zeichnet sich durch Temperaturen zwischen 60 °C und 120 °C aus. Entscheidend ist hierbei, dass das Lebensmittel während des Prozesses zusätzlich gegart wird. Somit eignet sich dies Verfahren vor allem für Fisch und Fleisch.

Um diese hohen Temperaturen zu erreichen braucht es oft eine zusätzliche Hitzequelle. Dies können Kohlen oder ein separater Gasbrenner sein.

5.2.2 Warmräuchern

Warmräuchern findet bei Temperaturen zwischen 30 °C und 50°C statt. Hierbei werden die Lebensmittel schonend gegart. Der charakteristische Rauchgeschmack wird hierbei direkt an die Lebensmittel abgegeben. Rohe Lebensmittel werden jedoch nicht völlig durchgegart.

Zeitlich dauert so ein Prozess ein paar Stunden. Aufgrund der Spaltung des Eiweißes sollte man die Speisen direkt verzehren.

5.2.3 Kalträuchern

Bei Temperaturen um die 30 °C kann der Räucherprozess Tage dauern. Hierbei wird jedoch bei vielen Lebensmitteln die Haltbarkeit gesteigert. Diese sind im Anschluss teilweise Wochen bis Monate haltbar. Auch bei dieser Methode des Räucherns nehmen die Lebensmittel den Rauchgeschmack an.

6. **HARDWARE**

Das Kapitel Hardware beschreibt neben den definierten Anforderungen an die auszuwählende elektronische Hardware auch den Auswahlprozess (Seite 18ff) und den Testaufbau für die Sensorfunktionalität (Seite 21ff).

6.1 **Anforderungen**

Im Folgenden werden die Anforderungen an die Hardware definiert. Dazu zählen die Sensorik, der auswertende Micro-Controller und die Hardware für das Nutzerinterface direkt am μC .

6.1.1 **Allgemeine Platzierung**

Maßgeblich für die Auswahl der verschiedenen Hardwarekomponenten ist die Position jedes Elements in und an dem Ofen. Es werden verschiedene Sensoren für die Qualitäts- und Prozessüberwachung eingesetzt.

Es werden die Überwachung der Temperatur, der Feuchtigkeit, der Rauchqualität und die Detektion von Feuer ausgewählt. Die Wahl dieser Komponenten ergibt sich aus den Anforderungen, sowie eigener Überlegung zur Projektentwicklung.

In Abbildung 3 ist der vereinfachte Aufbau eines Räucherofens zu sehen. Die verschiedenen Einbaupunkte sind hier von 1 bis 5 durchnummeriert. Jeder einzelne Punkt hat verschiedene Anforderungen an die verbaute Hardware. Die unterschiedlichen Stellen sind nötig, da verschiedenen Parameter aufgezeichnet werden müssen.

Ebenfalls muss darauf geachtet werden, dass der Räucherofen draußen steht und Temperaturen von $-5\text{ }^{\circ}\text{C}$ bis $30\text{ }^{\circ}\text{C}$ Umgebungstemperatur ausgesetzt ist. Weicht die Außentemperatur von diesen Werten ab, muss die Hardware abgebaut werden.

Neben den technischen Anforderungen sind die Kosten der Hardware zu beachten. Diese fließen ebenfalls in die Hardwarekonzepte ein.

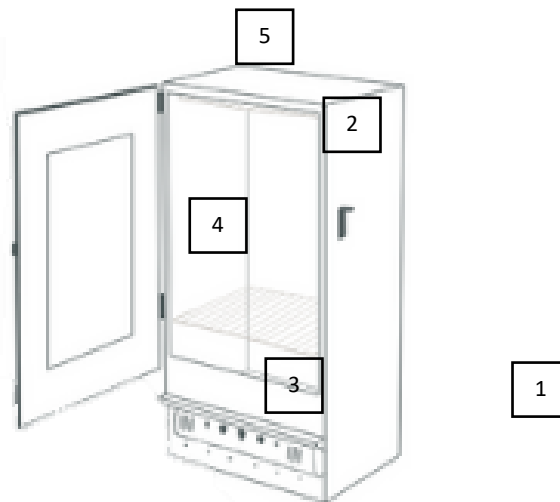


Abbildung 3: Aufbau eines typischen Räucherofens. Nummerierte Punkte markieren Stellen, an denen Hardware platziert werden soll.

Position 1 befindet sich außerhalb des Ofens. Hier herrschen keine besonderen Temperaturanforderungen. Aufgrund der nicht geforderten Standardisierung, sind auch keine weiteren Schutzmaßnahmen zu treffen. Trotzdem ist die platzierte Hardware vor Spritzwasser geschützt auszulegen. Hier ist die Hauptrecheneinheit, sowie das Nutzerinterface platziert.

Position 2 befindet sich an der Außenseite des Ofens. Diese Position eignet sich, um den Zustand der Tür zu überwachen. Dieser ist wichtig, damit mögliche Alarmsysteme bei Öffnung der Tür deaktiviert werden können. Hier ist auf den Temperatureinfluss und den Spritzwasserschutz zu achten.

Position 3 befindet sich unterhalb der Abtropfschale. Diese Position ist für die Feuererkennung notwendig. Die Temperaturanforderungen sind hier am höchsten. Während des Heißräucherprozesses treten im Ofen Temperaturen zwischen 60 °C und 120 °C auf (vgl. Konzept des Räucherns). Ein mögliches Eindringen von Rußpartikeln in die Sensortechnik ist zu beachten.

Position 4 befindet sich in der Nähe des Räuchergutes. Hier werden die Luftfeuchtigkeit und die Temperatur gemessen. Diese beeinflussen den Räucherprozess maßgeblich. Ebenfalls gelten hier die gleichen Anforderungen an Temperatur und Rußpartikelbeständigkeit wie an Position 3.

Position 5 befindet sich am Rauchausslass. Hier ist die Temperatur geringer als in der Räucherammer. Deshalb ist dies eine mögliche Position für Sensoren, welche nur eine geringere Temperatur aushalten können.

6.1.2 Temperatur

Der Temperatursensor dient zur primären Überwachung des Räuchervorgangs. Die Temperatur ist durch die verschiedenen Räucherverfahren vorgegeben. Somit liegen die erwarteten Temperaturen zwischen 30 °C und 120 °C. Diese Temperaturen müssen am Räuchergut herrschen, somit muss dieser Sensor an Position 4 sein. Nach unten bilden die Anforderungen der Außentemperatur eine Mindesttemperatur von – 5°C (vgl. Allgemeine Platzierung).

Der Sensor soll eine Genauigkeit von ± 2 °C haben. Die Schnittstelle muss mit dem Mikrocomputer kompatibel sein.

6.1.3 Feuchtigkeit

Der Feuchtigkeitssensor dient zur Erfassung weiterer Daten des Prozesses. Eine genaue Datenbasis über die Anforderungen liegt nicht vor. Der Sensor soll in der Nähe des Räuchergutes angebracht werden. Somit befindet er sich zusammen mit dem Temperatursensor auf Position 4. Er muss ebenfalls Temperaturen zwischen -5 °C und 120 °C aushalten, um allen Prozessen standzuhalten.

Die Genauigkeit soll ± 2 % betragen. Die Schnittstelle muss mit dem Mikrocomputer kompatibel sein.

6.1.4 Partikelmessung

Die Partikelmessung über einen Feinstaubsensor ist ebenfalls nicht durch eine Datenbasis gestützt. Die gemessenen Daten müssen rückwirkend anhand der Ergebnisse beurteilt werden. Die Messung soll direkt am Grillgut, somit an Position 4, stattfinden. Hier herrscht ein Temperaturbereich von -5 °C bis 120 °C.

Die Genauigkeit ist nicht festgelegt. Die Schnittstelle muss mit dem Mikrocomputer kompatibel sein.

6.1.5 Feuererkennung

Der Flammsensor soll ein Feuer des Brennmaterials detektieren. Dies kann nur direkt an der Glut stattfinden. Somit eignet sich hier Position 3 für die Überwachung. Hier sind Temperaturen von -5 °C bis 150 °C zu erwarten. Der eingesetzte Sensor sollte diese aushalten können.

Er soll vorrangig melden, wenn es ein Feuer geben könnte. Ein tatsächliches Feuer soll er mit 100 % erkennen. Die Komptabilität mit dem µC soll gegeben sein.

6.1.6 Tür Zustandserkennung

Der angebrachte Sensor soll erkennen, ob die Tür geöffnet oder geschlossen ist. Hierbei soll er von jeglichem Material unbeeinflusst bleiben. Er muss im Temperaturbereich von -5 °C bis 150 °C aushalten.

6.1.7 Display und Taster

Das Display und der Taster sollen zur Visierung von Daten direkt am Räucherofen genutzt werden. Hier genügt eine simple Ausgabe von Zeichenketten. Die Nutzung wird auf die Ausgabe von aktuellen Werten, sowie die Eingabe von Befehlen ausgelegt.

Es eignet sich Position 1, sodass das Nutzerinterface außerhalb des Ofens platziert ist. Somit muss nur den generellen Temperaturanforderungen von -5 °C bis 30 °C standgehalten werden.

6.1.8 μC

Der Mikrokontroller soll als zentrale Recheneinheit dienen. Er soll die Sensorauswertung, sowie die Weiterleitung der Daten durchführen. Ebenfalls soll er in der Lage sein, bei einer Weiterentwicklung des Projektes, Aktoren oder weitere Sensoren zu steuern.

Um die einzelnen Sensoren zu steuern, muss er über die nötigen Kommunikationsprotokolle verfügen. Diese gestalten sich in Abhängigkeit der Sensoren. Der Mikrokontroller muss das Display ansteuern, sowie die Eingabe externer Tasten auswerten können. Im Gesamten muss die Anzahl der Aus- und Eingänge für das aktuelle Projekt, sowie für mögliche begrenzte Erweiterungen ausreichend sein. Dies ist nach jetzigem Stand gegeben, wenn noch mindestens fünf weitere Anschlüsse am μC frei sind.

Der Mikrocontroller wird laut Konzept außerhalb des Ofens positioniert. Somit ist er an Position 1. Deshalb muss er nur den generellen Temperaturanforderungen von -5 °C bis 30 °C entsprechen.

Um die Daten über das Internet zu verschicken, benötigt der μC eine Schnittstelle. Diese ist so zu wählen, dass er sich kabellos in ein Netzwerk einloggen kann (WIFI-Schnittstelle).

6.2 Auswahl

6.2.1 Allgemein

Das Vorgehen bei der Sensorauswahl wird bei jedem Sensortypen einzeln angewandt. Hierbei fließen die allgemeinen Anforderungen, sowie die individuellen Anforderungen mit ein. Erfüllen zwei Sensoren gleichermaßen die Anforderung, so wird der Preis als Hauptkriterium hinzugezogen.

Um die Ansteuerung zu vereinfachen werden Sensoren zusammengefasst. Diese nutzen somit nur eine Schnittstelle, womit die Implementierung der Kommunikation einfacher wird.

6.2.2 Temperatur & Feuchtigkeit

Bei der Auswahl des Temperatursensors gibt es verschiedenste Möglichkeiten. Die günstigste Variante wäre ein temperaturabhängiger Widerstand. Dies benötigt jedoch eine Beschaltung und ist von der Genauigkeit nicht geeignet. Somit wird ein fertiges Modul gewählt. Dies sendet die Daten an den Mikrokontroller. Geeignet wäre z.B. LM135.

Bei den Feuchtigkeitssensoren gibt es nur wenige die den Temperaturanforderungen von -5 °C bis 120 °C entsprechen. Einer dieser wäre der elektrolytischer Feuchtesensor EFS-10 (vgl. Datenblatt Abbildung 22).

Um die Anforderung der Reduktion der Schnittstellen zu erfüllen, fällt die Wahl auf den Hygrochip HYT-221. Dieser kombiniert Feuchtigkeit- und Temperaturmessung in einem Bauteil und sendet die Daten über SPI an den Mikrokontroller.

6.2.3 Partikelmessung

Bei der Auswahl eines Sensors für die Partikelmessung ist der Preis besonders wichtig. Aufgrund des Preises, sowie der möglichen Ansteuerung wurde der Sensor Sensirions SPS30 ausgewählt. Alle anderen geeigneten Sensoren sind zu teuer für das Projekt. Um

diesen Sensor zu nutzen muss ein Kompromiss eingegangen werden. Der Sensor ist nur auf Temperaturen bis zu 60 °C ausgelegt (vgl. Datenblatt). Somit ist er für eine Messung an Position 4 (Abbildung 3) nicht geeignet. Die Prozesse des Warm- und Heißräucherns könnten somit nicht durchgeführt werden. Alternativ kann der Sensor an Position 5 gesetzt werden. Hier ist die Temperatur niedriger. Eine genaue Untersuchung ist noch durchzuführen.

6.2.4 Feuererkennung

Für die Flammerkennung gibt es ein Modul von Arduino (KY-026 vgl. Datenblatt in Abbildung 20). Dieses passt in den preislichen Rahmen und stellt eine direkte Justierung, sowie eine aufbereitete Schnittstelle bereit. Neben diesem gibt es noch ein weiteres Modul in der Preisklasse. Dieses erfüllt jedoch nicht die Temperaturanforderungen. Die Auswertelogik des KY-026, welche direkt am eigentlichen Sensor verbaut ist, kann nicht den erforderlichen Temperaturbereich erfüllen. Um dieses Problem zu umgehen, wird in der finalen Version die Verbindung zwischen Sensor und Logik ausgebaut bzw. verlängert. Die Funktion sollte dabei erhalten bleiben.

6.2.5 Tür Zustandserkennung

Bei der Auswahl des Türsensors gibt es zwei Optionen. Ein fertiger Sensor, welcher aus zwei Teilen besteht oder eine selbst entwickelt Variante. Die fertigen Sensoren erfüllen nicht das erforderliche Hitzeprofil. Somit wird eine Kombination als Hall-Sensor (SS 495 vgl. Datenblatt in Abbildung 21) und Magneten genutzt. Diese ist bis 125 °C Hitzebeständig.

6.2.6 Display und Taster

Den Anforderungen entsprechend wird ein einfaches Display ausgewählt. Die Wahl fiel auf ein klassisches LC Display (Liquid Crystal Display, kurz LCD). Dieses hat zwei Zeilen, womit mehrere Daten gleichzeitig dargestellt werden können.

Als Taster wurden simple Modelle gewählt, da hierauf kein Fokus liegt.

6.2.7 µC Auswahl

Neben den geforderten Kriterien, bezüglich Schnittstellen und Anzahl an möglichen Verbindungen spielt die Programmierung eine wichtige Rolle. Neben dem [Arduino Nano 33 IoT](#) stand auch der DEBO JT ESP32 zur Wahl. Beide sind über die Arduino IDE in C programmierbar.

Die Wahl fällt auf den Arduino, da im Zusammenspiel mit dem Flammensor die Integrationen einfacher zu gestalten ist. Ebenfalls sind die Möglichkeiten bezüglich weiterer Schnittstellen größer.

6.3 Testaufbau Sensorfunktion

6.3.1 Ziel und Vorgehensweise

Ziel dieses Aufbaues ist es, die generelle Funktion der Bauteile zu überprüfen. Der Fokus dieses Tests liegt auf der Interaktion der Sensoren miteinander, sowie der Möglichkeiten der Auswertungen. Hierbei werden die einzelnen Sensoren in Betrieb genommen und überprüft, ob sie den Anforderungen entsprechen. Die Auswertung erfolgt stichprobenartig. Aufgrund der Testmöglichkeiten können die Tests nur unter Zimmertemperaturbedingungen stattfinden. Es wird davon ausgegangen, dass eine Überprüfung der Funktionen in einem Temperaturbereich von 10 °C bis 20°C ausreichend ist.

Hierfür werden die Sensoren, sowie der Micro Controller auf einem Steckbrett miteinander verbunden.

Es wird eine Menüführung implementiert, welche die Funktionen der einzelnen Sensoren testet. Diese ist ein Vorläufer der finalen Version, welche für die endgültige Hardwarekonfiguration entwickelt wird. Mit dieser werden die Tasten zur Eingabe und das Arbeiten mit dem Display getestet. Eine genaue Beschreibung der Menüstruktur, sowie die Implementierung befindet sich unter Abschnitt 7.1 µC Testsoftware Sensordaten.

6.3.3 Hardwareaufbau

Der bereits gezeigte Schaltplan wurde auf einem Steckbrett umgesetzt. Hier werden die einzelnen Komponenten über Steckverbindungen verbunden. Diese können somit temporär verbunden und das System überprüft werden.

In Abbildung 5 ist der Testaufbau auf dem Steckbrett zu sehen.

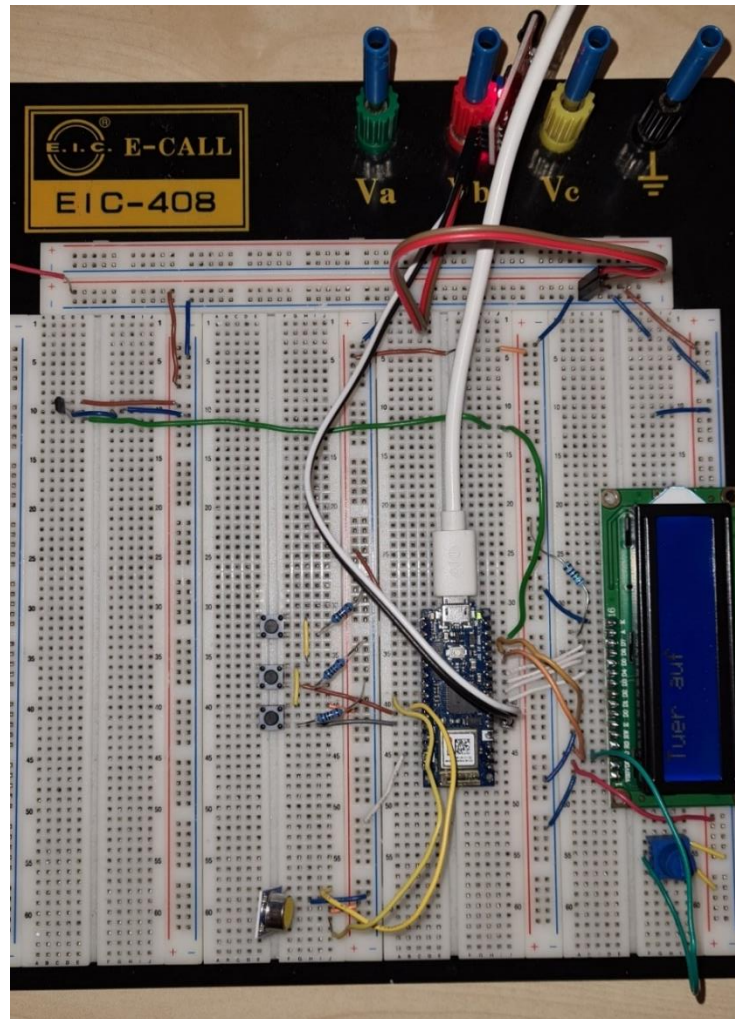


Abbildung 5 Steckbrettaufbau für die Testläufe der Sensoren

Die Steuerung durch die Menüführung ist durch die Taster an der linken Seite gegeben. Auf der Abbildung 5 ist der Steckbrettaufbau zu sehen, sowie die verschiedenen Sensoren für Temperatur, Feuchtigkeit, Feuererkennung und der Zustand der Tür. Wie eingangs erwähnt ist der Feinstaubsensor nicht verbaut.

6.3.4 Testläufe

Wie eingangs erwähnt, werden die Testläufe bei Temperaturen zwischen 10 °C und 25 °C durchgeführt.

Beim Flammsensor wird vor den Sensor die Flamme eines Feuerzeuges gehalten. Hierbei wird über das Display der aktuelle Ausgabewert des Sensors dargestellt. Dieser schwankt bei erkannter Flamme zwischen Feuer erkannt und nicht erkannt hin und her. Die Ausgabe muss im Folgenden noch am Ofen justiert werden und eine Filterung des Signales, zur eindeutigen Darstellung durchgeführt werden. Die grundsätzliche Funktion wurde mit diesen Testläufen festgestellt. Somit kann der Sensor im weiteren Verlauf des Projektes eingesetzt werden.

Der Hall Sensor wird angeschlossen und im Anschluss an einen Magneten herangeführt. Hierbei gibt er als Ausgabewert die Versorgungsspannung aus. Diese wird über den Mikrocontroller eingelesen und auf dem Display dargestellt. Eine genaue Erprobung im Einsatzumfeld ist zu empfehlen, da die Materialauswirkungen des Räucherofens mit einfließen könnten.

Der Temperatur- und Feuchtigkeitssensor wird während der Testtemperaturen ausgelesen. Hierbei sendet er konstant Datenpakete über SPI an den Mikrocontroller. Diese werden aktuell nur als hexadezimal Werte auf dem Display ausgegeben. Die Umrechnung der Daten in lesbare Dezimalwerte muss noch erfolgen. Bei Veränderung der Temperatur werden auch die hexadezimalen Messwerte modifiziert. Somit ist die grundlegende Funktion gezeigt. Weitere Erprobungen, sowie die finale Auswertung erfolgen später.

Das Display, sowie die Taster zur Nutzereingabe wurden während der Tests zur Ausgabe und zur Navigation durch das Menü benutzt. Somit wurde eine Funktion dieser Konfiguration bestätigt.

7. SOFTWARE

Das Kapitel Software gibt einen Einblick in die Funktion der einzelnen Komponenten der Studienarbeit, wie sie bereits in Kapitel 3 - Konzept vorgestellt wurden. Abbildung 6 zeigt den prinzipiellen Aufbau der Studienarbeit, nach dem beschriebenen Konzept aufgeteilt. Die Pfeile stellen dabei den Datenfluss von der Sensorik über den auswertenden Microcontroller an die Visualisierung vor Ort (via Display) und an die Hand-Held-Geräte (Zugriff via Browser und App) dar. In den folgenden Abschnitten wird die Software beschrieben, die die Kommunikation zwischen den einzelnen Komponenten ermöglicht.

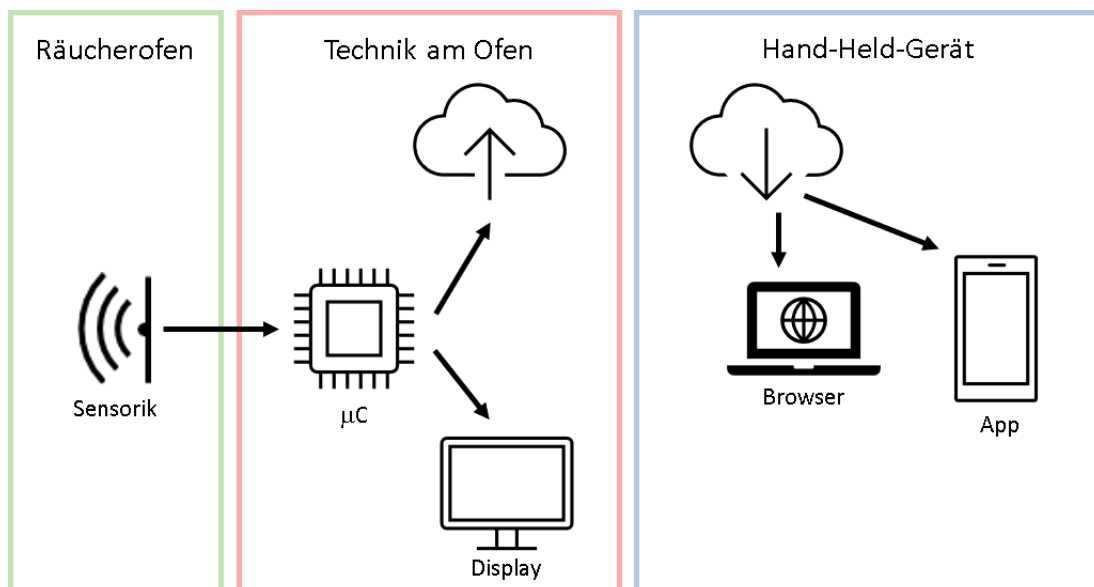


Abbildung 6: Aufbau der Studienarbeit aufgeteilt nach dem Grundkonzept siehe Kapitel 3; durch die Pfeile ist der Datenstrom zwischen den Komponenten dargestellt.

7.1 µC Testsoftware Sensordaten

In Abbildung 7 ist der funktionale Zustandsgraf der Software für den Sensortest zu sehen. Jeder Zustand, welcher mit z.B. mit Display_1 bezeichnet wird beschreibt eine andere Ausgabe auf dem LC-Display.

In den verschiedenen Zuständen werden die Sensordaten abgebildet. So wird zum Beispiel in Display_3 die Temperatur und die Feuchtigkeit auf dem Display ausgegeben. Die Änderung der Zustände erfolgt über die „up“ und „down“ Taster. Diese dienen zur Navigation durch die Menüführung. Es wird auch ein „OK“ Taster implementiert, welcher Programmtechnisch noch keine Anwendung findet.

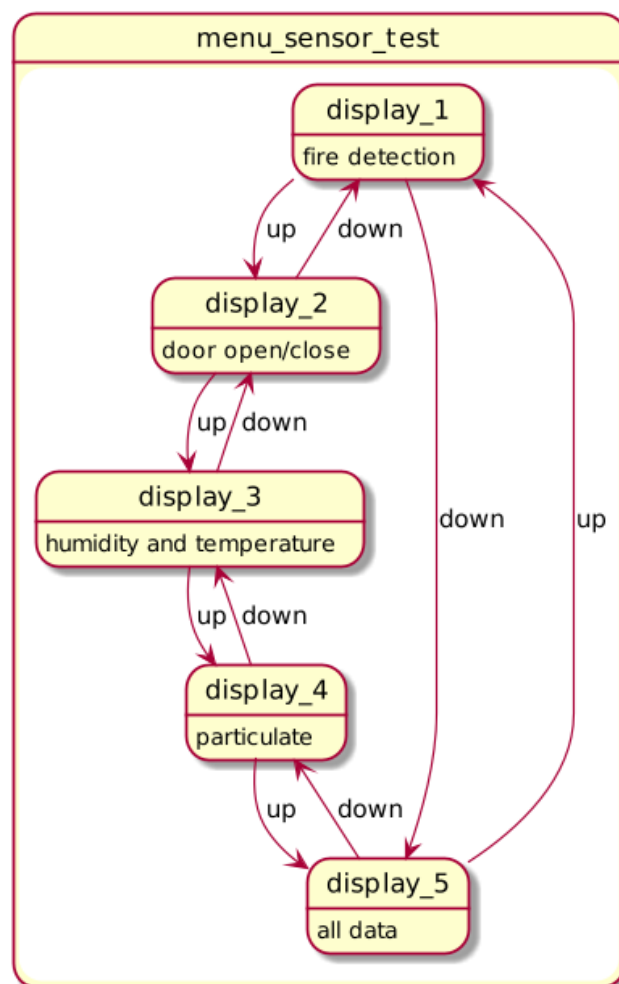


Abbildung 7: Menüführung der Software für den Sensortest

7.2 µC Testsoftware Wifi

Der Abschnitt µC Testsoftware Wifi beschreibt Aufbau, Funktion und Ablauf der Software, die für die Kommunikation des Microcontrollers mit der Umwelt über das WIFI-Modul verantwortlich ist. Dazu gehört die Verbindung mit einem spezifizierten WIFI-Netzwerk, der Aufbau eines Webserver und die Abwicklung der Kommunikation mithilfe des Hypertext Transfer Protocols (HTTP). Außerdem soll die Testsoftware bereits das Grundgerüst für das zyklische Anfordern der Sensordaten aufweisen, sodass die Software unter möglichst realitätsnahen Umständen getestet werden kann.

Für die WIFI-Verbindung und die Kommunikation des µC mit dem WIFI-Modul werden die Funktionen der [WiFinINA](#) Bibliothek und der [SPI](#) Bibliothek genutzt.

7.2.1 Setup

Die Setup Funktion des Microcontrollers wird einmal zu Beginn des Programms aufgerufen. In Abbildung 8 ist der prinzipielle Ablauf dargestellt. Am Anfang wird zu Debug-Zwecken die serielle Kommunikation gestartet, sodass die restlichen Softwarekomponenten besser überwacht werden können. Im Anschluss wird überprüft, ob das WIFI-Modul des µC aktiv ist oder ein Software Update verfügbar ist. Reagiert das WIFI-Modul nicht, wird das Programm nicht weiter ausgeführt.

Damit sich der µC mit einem Netzwerk verbinden kann, werden in einer eigenen Header Datei („secrets.h“) der Name des Netzwerks (die SSID) und das Passwort in Klartext hinterlegt.

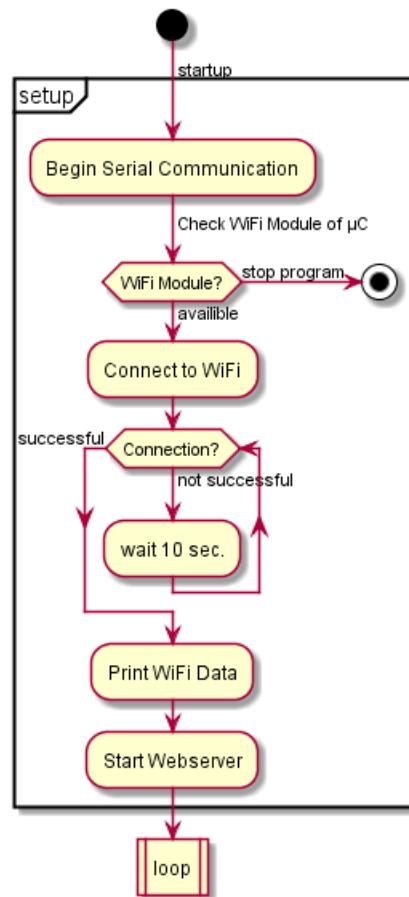


Abbildung 8: Ablauf der Setup Funktion der WiFi Testsoftware; nach dem Start der seriellen Kommunikation, wird die WiFi-Verbindung hergestellt und anschließend der Webserver gestartet.

Für jeden Versuch des Aufbaus einer Verbindung werden Router und WIFI-Modul 10 Sekunden Zeit gegeben, bevor der Status abgefragt wird. Bei erfolgreicher Verbindung werden die Netzwerkdetails über die Serielle Schnittstelle ausgegeben. Mithilfe der so gewonnenen lokalen IP-Adresse und der MAC-Adresse können in einem späteren Schritt die Netzwerkeinstellungen des Routers angepasst werden, sodass der Webserver auch von außerhalb des lokalen Netzwerks erreichbar ist.

Der Webserver wird zum Schluss mit Port 80 gestartet. Anschließend fährt das Programm mit der loop Funktion fort.

7.2.2 Loop

Die loop Funktion des μC wird nach der Setup Funktion kontinuierlich durchlaufen und enthält Aufrufe von Funktionen, die zyklisch abgearbeitet werden sollen. In Abbildung 9 ist der prinzipielle Ablauf der Loop Funktion dargestellt.

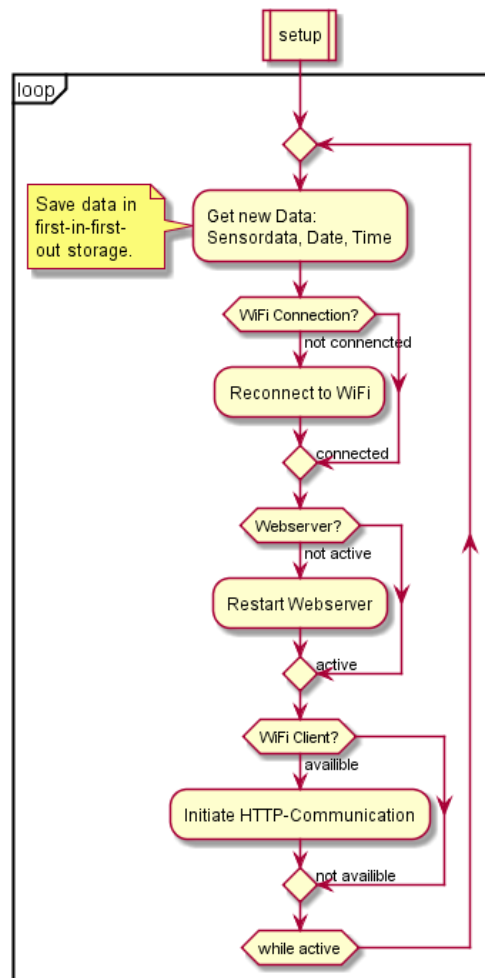


Abbildung 9: Ablauf der Loop Funktion der WiFi Testsoftware; nach dem Abfragen neuer Daten, werden die WIFI-Verbindung, der Status des Webservers und das Vorhandensein von Clients abgefragt.

Zu beachten ist dabei, dass die Aufrufe der Funktionen nicht in jedem Durchlauf der loop, sondern nur in festgelegten Intervallen stattfinden. Diese werden über den Vergleich zwischen aktueller Systemzeit und Systemzeit des letzten Aufrufs dieser Funktion überprüft.

Der erste Aufruf der loop besteht in dem Abfragen von neuen Daten. Dabei werden in einer Datenstruktur die Sensordaten für den Türsensor, die Flammerkennung, die

Temperaturmessung und die Feuchtigkeitsmessung abgespeichert. Zusätzlich dazu werden die aktuelle Uhrzeit und das Datum über das WIFI Modul angefordert und formatiert gespeichert. Für die Formatierung wird die Bibliothek [TimeLib](#) genutzt.

Im vorliegenden Testfall werden künstlich generierte Sensordaten genutzt. Dabei werden die Temperatur und die Feuchtigkeit mit jedem Aufruf um Eins erhöht, während die Flammdetektion zwischen Null und Eins alterniert und die Türerfassung durchgehend auf null steht.

Die Datenstruktur wird anschließend in einem sogenannten FIFO (first in first out) Speicher gespeichert. Der Speicher ist, wie in Abbildung 10 zu sehen, als festes Feld von Datensätzen aufgebaut. Dabei wird das aktuelle Datenpaket an die erste freie Stelle gesetzt (D_{in}). Ist keine Stelle im Feld frei, wird das erste Element aus dem Speicher entfernt (D_{out}), alle Elemente rutschen um einen Platz auf und der neue Datensatz wird an die letzte Stelle geschrieben. In der aktuellen Implementierung werden maximal die letzten 20 Datensätze gespeichert. Hierbei sind jedoch auch größere Speicher vorstellbar. Diese sind hauptsächlich durch die Größe des dynamischen Arbeitsspeichers begrenzt, den der Microcontroller zur Verfügung hat und der Verzögerung in der HTTP-Kommunikation bei großen Datenmengen.

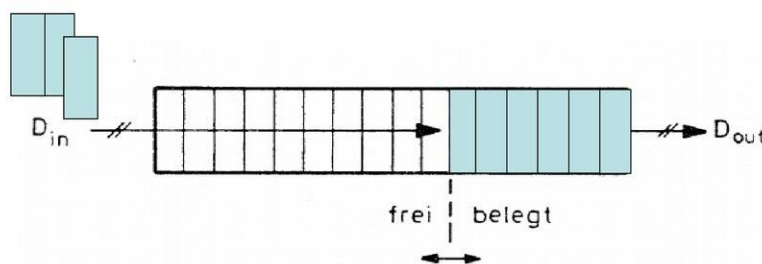


Abbildung 10: Prinzip eines FIFO-Speichers; neue Daten (links) werden an erste freie Stelle geschrieben; werden Elemente aus dem Speicher genommen, werden sie von vorne (rechts) genommen.

Wie in Abbildung 9 weiter zu sehen ist, wird im Anschluss an die Datenabfrage der Status des WIFI-Moduls und des Webserver überprüft. Wirft einer dieser Checks ein unerwünschtes Ergebnis (WIFI nicht verbunden bzw. Webserver nicht aktiv), so wird dies versucht zu korrigieren. Dazu können Funktionen, die bereits in der Setup Funktion

genutzt werden, wiederaufgerufen werden, um erneut mit dem WIFI-Netzwerk verbunden zu werden bzw. den Webserver neu zu starten.

Zum Abschluss der loop Funktion, wird bei jedem der Durchläufe überprüft, ob dem Webserver ein Client zur Verfügung steht. Ist dies der Fall, so wird die Funktion HTTP-Kommunikation aufgerufen, die im folgenden Abschnitt behandelt wird.

7.2.3 HTTP-Kommunikation

Das Unterkapitel HTTP-Kommunikation beschreibt Aufbau, Funktion und Ergebnisse der `http_communication()` Funktion. In Abbildung 12 ist der prinzipielle Ablauf dieser Funktionalität gezeigt. Wird die Funktion nach dem Erkennen eines verfügbaren Clients des Webserver aufgerufen, so wird zu Beginn der HTTP-Request (Anfrage) dieses Clients komplett ausgelesen. Erkennt das Programm beim Lesen zwei leere Zeilen hintereinander, so ist die Anfrage des Clients fertig und der μ C beginnt mit dem Senden der Antwort. Dafür wird zuerst ein standard HTTP-Antwort-Header (Kopfzeile) gesendet, der die wichtigsten Informationen für die Verbindung mit dem Client zusammenfasst (siehe Abbildung 11).

```
HTTP/1.1 200 OK
Content-Type: text/html
Connection: close
```

Abbildung 11: Antwort-Kopfzeile an Client nach HTTP-Request.

Dabei wird dem Client mitgeteilt, dass der Aufruf der Website erfolgreich war (HTTP-Status Code „200 OK“), dass die Verbindung zum Client einmalig ist, also nicht automatisch erneuert wird und, dass der Inhalt der Übertragung aus Text besteht.

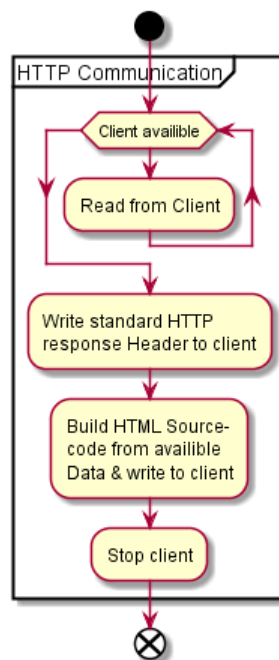


Abbildung 12: Prinzipieller Aufbau der HTTP_Communication() Funktion; nach dem vollständigen Auslesen der Anfrage des Clients, wird mit einem statischen Antwort Header und anschließend dem zusammengebauten HTML Quellcode inklusive aller gewünschten Daten geantwortet und die Verbindung getrennt.

Nachdem die Antwort-Kopfzeile feststeht, werden die zu sendenden Daten aus dem First-in-First-out Speicher in ein Konstrukt eingefügt, sodass der Quellcode abschließend als String an den Client übertragen wird. Dieses ist in der Hypertext Markup Language (HTML) geschrieben und sieht vor, dass die entstehende Website aus zwei Teilen besteht (siehe Abbildung 13). Dabei repräsentiert der obere Teil die jeweils letzten Sensordaten (und zugehörigen Zeiten). Der untere Teil der Website wird von einer Tabelle der letzten gespeicherten Sensordaten eingenommen. Auf diese Weise ist gesichert, dass ein Endbenutzer einfach zwischen einem Blick auf die letzten Daten und einer kompletten Betrachtung aller aufgezeichneter Daten wechseln kann. Im vorliegenden Fall wird die Website über die lokale IP-Adresse des μ C aufgerufen.

<div> ← → ↺ 🏠 🔒 🔗 192.168.1.101 </div> <div> ⚙️ Meistbesucht 📁 basic 📁 Projekte </div>					
Latest Sensordata					
humidity: 36					
temperature: 37					
flame detection: 0					
door open: 0					
time + date: 20:47:59 15.1.2022					
humidity	temperature	flame detection	door open	time	date
17	18	1	0	20:46:24	15.1.2022
18	19	0	0	20:46:29	15.1.2022
19	20	1	0	20:46:34	15.1.2022
20	21	0	0	20:46:39	15.1.2022
21	22	1	0	20:46:44	15.1.2022
22	23	0	0	20:46:49	15.1.2022
23	24	1	0	20:46:54	15.1.2022
24	25	0	0	20:46:59	15.1.2022
25	26	1	0	20:47:4	15.1.2022
26	27	0	0	20:47:9	15.1.2022
27	28	1	0	20:47:14	15.1.2022
28	29	0	0	20:47:19	15.1.2022
29	30	1	0	20:47:24	15.1.2022
30	31	0	0	20:47:29	15.1.2022
31	32	1	0	20:47:34	15.1.2022
32	33	0	0	20:47:39	15.1.2022
33	34	1	0	20:47:44	15.1.2022
34	35	0	0	20:47:49	15.1.2022
35	36	1	0	20:47:54	15.1.2022
36	37	0	0	20:47:59	15.1.2022

Abbildung 13: Darstellung der HTTP-Antwort des μ C in einem Browser. Aufgerufen über die lokale IP-Adresse, die letzten Sensordaten oberhalb der Tabelle aller gespeicherter Daten.

7.2.4 Port-Forwarding und dynamisches DNS

Damit der Zugriff auf den μ C auch außerhalb des lokalen Netzwerkes möglich ist, müssen die Netzwerkeinstellungen des Heimnetzwerkes angepasst werden. Dazu gehören das Aktivieren eines Hosts für dynamisches DNS und das Aktivieren eines virtuellen Servers für Port-Forwarding.

Ein dynamisches DNS (Dynamisches Domännennamensystem) ist notwendig, um die dynamische IP-Adresse, die das Heimnetzwerk im World Wide Web zugewiesen bekommt mit einem unveränderlichen Host-Namen zu verknüpfen. Auf diese Weise ist das Netzwerk trotz veränderter IP-Adresse über den Host-Namen erreichbar. Im

Anwendungsfall wurde der Host-Name „studienarbeit21.ddns.net“ bei dem Anbieter no-ip.com gewählt und über die Netzwerkeinstellungen des Routers mit dem Heimnetzwerk verbunden.

Ein virtueller Server ist nötig, damit der lokale Server über das Internet bereitgestellt werden kann. Dabei wird der virtuelle Server über einen externen Port und eine interne IP-Adresse definiert. Alle Verbindungsanfragen, die diesen Port erreichen, werden an die lokale IP-Adresse an dem internen Port weitergegeben. Dies wird auch als Port-Forwarding bezeichnet und ist in Abbildung 14 an einem Beispiel gezeigt. Dabei erhält der lokale Router mit der IP-Adresse 72.243.14.15 eine Anfrage an Port 443. Die Anfrage wird somit an Port 443 weitergeleitet (lokale IP-Adresse 192.168.1.70).

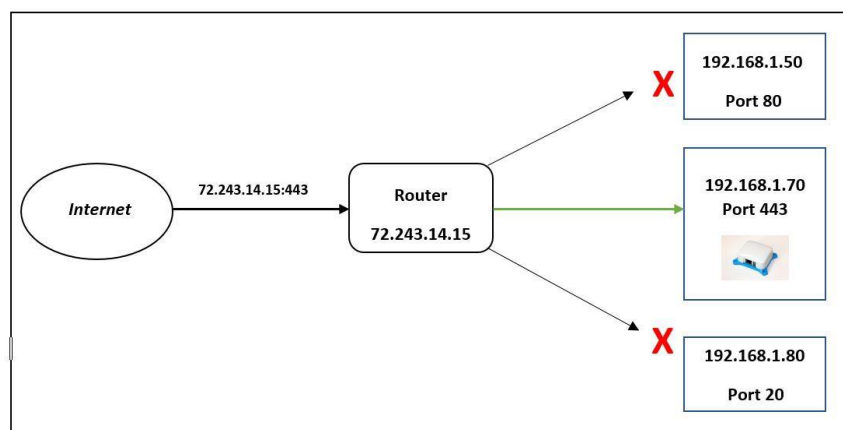


Abbildung 14: Beispiel des Port-Forwardings anhand einer Anfrage aus dem Internet an einen speziellen Port des Routers.

In Abbildung 15 ist die Konfiguration des virtuellen Servers im Anwendungsfall gezeigt. Dabei wird der Externe Port auf den Standardwert 80 gesetzt. Das heißt, dass jeder, der die IP-Adresse oder den Host-Namen des Netzwerkes ohne weitere Spezifizierung aufruft, automatisch an den virtuellen Server weitergeleitet wird.

Die interne IP-Adresse ist die lokale IP-Adresse des μ C, die bereits in Abbildung 13 genutzt wird, um die Website im lokalen Netzwerk aufzurufen. Damit das Port-Forwarding gelingen kann muss die IP-Adresse des μ C statisch sein und darf sich nicht ändern. Dies kann auch in den Netzwerkeinstellungen des Routers festgelegt werden. Die dafür benötigte MAC-Adresse und IP-Adresse des μ C werden bei jedem Programmstart in der Setup Funktion über die Serielle Schnittstelle ausgegeben.

Der Interne Port muss in den Netzwerkeinstellungen die Nummer 80 tragen, da der Webserver, der bereits in der Setup Funktion gestartet wird, auch mit Port 80 definiert ist.

Externer Port	Interne IP-Adresse	Interner Port	Protokoll
80	192.168.1.101	80	TCP

Abbildung 15: Netzwerkeinstellungen für einen virtuellen Server; Aufrufe des Netzwerkes werden standardmäßig an die spezifizierte IP-Adresse an Port 80 weitergeleitet.

7.3 **Smartphone App**

Der folgende Abschnitt geht auf die Implementierung einer Android Applikation zur Darstellung von Daten von einer HTML Website ein.

7.3.1 **Allgemeines**

Ziel der Applikation (kurz App) ist in diesem Anwendungsfall, dass der Endnutzer nach dem Öffnen der Anwendung die aktuellen Sensordaten einsehen kann. Um dies zu erreichen wird festgelegt, dass die App in Android Studio für Android Version 8 designt wird.

Die Entscheidung kommt daher, dass der Marktanteil des Android Betriebssystems am Smartphone Absatz 2020 bei [84,1 %](#) lag. Von den genutzten Android-Geräten sind wiederum ca. 82,7 % mit Android 8 und niedriger kompatibel (siehe Anhang Abbildung 20). Das bedeutet, dass das Level der Programmierschnittstelle (engl. API) 26 oder niedriger sein muss, während zum aktuellen Zeitpunkt bereits API Level 30 (Android 11) verfügbar ist. Somit existiert ein Kompromiss aus hoher Verfügbarkeit für eine möglichst breite Anwenderbasis und den verbesserten Funktionalitäten höherer Versionen.

Auf funktionaler Ebene arbeitet die App, wie in Kapitel 3.3 beschrieben nach einem simplen Prinzip:

1. Download der Website
2. Extrahieren der Daten
3. Darstellen der Daten

Dabei werden zwei separate Prozesse (engl. Threads) genutzt. Der erste Thread wird im Vordergrund ausgeführt und ist für die Darstellung der App für den Endnutzer verantwortlich (UI-Thread, User-Interface-Thread). Ein zweiter Thread wird im Hintergrund aufgerufen, um den Download der Website auszuführen.

7.3.2 Main Activity

In diesem Abschnitt werden die Funktion und der Aufbau der Android App, genauer der Main Activity im sogenannten UI-Thread beschrieben. Dies beinhaltet alle Aufgaben, die die App im Vordergrund ausführen muss. Dazu gehören die Verwaltung der graphischen Benutzeroberfläche (GUI, graphical user interface), das Extrahieren der Sensordaten aus dem Quellcode der Website und letztendlich die Darstellung der Werte.

In Abbildung 16 ist der prinzipielle Ablauf der Main Activity gezeigt, die für die korrekte Darstellung der Sensordaten auf dem Smartphone zuständig ist. Zu erkennen ist, dass die App nach dem Starten durch den Anwender ihre Startup-Routine durchläuft und anschließend bis zum Schließen der Anwendung prüft, ob die Bedingung für einen Download der Website erreicht ist.

In der Startup Phase werden sowohl die graphischen Elemente der App instanziiert als auch Schnittstellen, die bestimmen, wann bzw. wie der Download der Website gestartet wird.

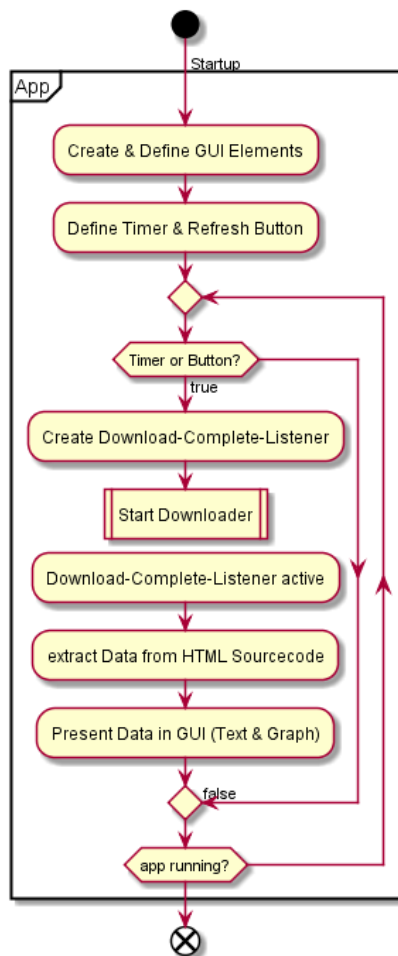


Abbildung 16: Prinzipieller Ablauf der Main Activity der Smartphone App; nach dem Start der Anwendung werden Download, Extrahierung und Darstellung der Daten zyklisch überprüft.

Dabei wird für den manuellen Refresh Button (Knopf für manuelles aktualisieren der Daten) die bereitgestellte `setOnClickListener()` Methode verwendet, sodass bei jedem Aktivieren des Knopfes die `startDownload` Methode aufgerufen wird. Die Timer eigene Methode `schedule()` wird genutzt, sodass von dem Timer-Objekt periodisch alle 30 s die `startDownload()` Methode aufgerufen wird (siehe Abbildung 17).

```

btRefresh.setOnClickListener(b -> startDownload());
timer.schedule(() -> { startDownload(); }, delay: 100, period: 30000);

```

Abbildung 17: Definition der Schnittstellen zwischen Refresh Button (oben) und des Timers (unten) mit der `startDownload()` Methode

Wird nach der Startup Phase der Refresh Button gedrückt oder läuft der Timer aus, so ist die Bedingung für ein Herunterladen der Website erfüllt und es wird ein Objekt der Klasse DownloadCompleteListener angelegt (siehe Abbildung 16).

Dies ist nötig, da das Herunterladen der Website im Hintergrund erfolgt, während der Rest der App im Vordergrund läuft. Damit der UI-Thread erfährt, dass der Hintergrundthread fertig ist wird dieses sogenannte Interface genutzt.

Anschließend wird ein Objekt der eigenen Downloader Klasse erstellt und der Download gestartet. Nötig ist für diesen Start des Downloaders nur eine korrekt übergebene URL (Uniform Resource Locator) zur Website, die heruntergeladen werden soll. Die Anzahl der Threads, die gleichzeitig im Hintergrund versuchen die Website herunterzuladen ist auf einen beschränkt, um Überlastungen sowohl der App als auch des Webserver durch eine zu hohe Anzahl von Anfragen zu unterbinden.

Wird nach einem erfolgreichen Download der DownloadCompleteListener ausgelöst, erfolgt die Verarbeitung der heruntergeladenen Daten. Da der gesamte Quellcode der Website, in HTML codiert heruntergeladen wird, müssen die erwünschten Sensordaten des μ C herausgefiltert werden. Dies geschieht mithilfe der [Jsoup](#) Bibliothek, die den gesamten Quellcode der Website einliest und es ermöglicht nach Schlagwörtern zu sortieren. Somit kann die ursprüngliche Tabelle der Sensordaten des μ C wieder zusammengesetzt werden.

Der nächste Schritt im Aufbau der App sieht die korrekte Darstellung der erhaltenen Daten vor. Dies geschieht sowohl in einem Graphen (mithilfe der [GraphView](#) Bibliothek) als auch per Textausgabe. Dabei werden die textuellen Darstellungen für die aktuell letzten Sensordaten genutzt, während der Graph auch den zeitlichen Verlauf der Sensordaten berücksichtigt. In Abbildung 18 ist ein Screenshot des genutzten Android Emulators abgebildet, nachdem die App erfolgreich die Website des Microcontrollers heruntergeladen, relevante Daten extrahiert und anschließend als Text und Graph dargestellt hat. In der oberen Bildhälfte sind die letzten Sensordaten in Textform dargestellt (neben dem Refresh Button in der rechten oberen Ecke). Die untere Bildhälfte beinhaltet die zeitlichen Verläufe der Temperatur in Rot und der Luftfeuchtigkeit in Blau (in diesem Fall für die letzten 80 Sekunden).

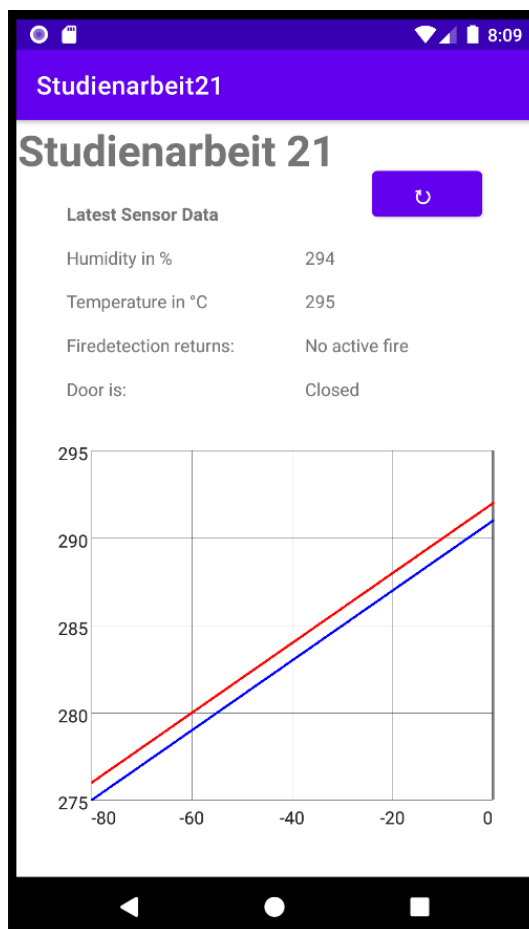


Abbildung 18: Screenshot des Android Emulators (API 26) nach erfolgreichem Herunterladen und Decodieren des Website Quellcodes. Textuelle Ausgabe der letzten Daten (oberer Bildhälfte); zeitlicher Verlauf der Testdaten zu Temperatur (rot) und Feuchtigkeit (blau) in den letzten 80 s (untere Bildhälfte).

7.3.3 Downloader

Der folgende Abschnitt beschäftigt sich mit dem Aufbau und der Funktion der implementierten Klasse Downloader. Wie bereits unter 7.3.1 Allgemeines erwähnt, soll das Herunterladen der Website im Hintergrund erfolgen. Dies liegt daran, dass dieser Vorgang unter Umständen mehrere Sekunden in Anspruch nehmen kann. Diese Verzögerung setzt sich dabei hauptsächlich aus den Latenzzeiten der Kommunikation über das HTTP-Protokoll mit dem Webserver zusammen. Dazu kommen außerdem Verzögerungen durch die Server von noip.com und den lokalen Routern.

Würde das Herunterladen der Website im Vordergrund stattfinden, so würden während diesem Vorgang keine anderen Aufgaben abgearbeitet werden (wie beispielsweise Anklicken von Knöpfen oder Aktualisieren von Anzeigen). Das Android Betriebssystem würde die Anwendung nach 5 Sekunden mit der Meldung „Die App reagiert nicht mehr“ beenden. Außerdem kann der UI-Thread seit Android Version 3 generell nicht mehr für Netzwerkverbindungen genutzt werden.

Aus diesem Grund wird beim Aufruf des Downloads aus der Main Activity ein neuer Thread erstellt, der im Hintergrund parallel zum UI-Thread läuft. Nach dem Start des Hintergrund-Threads als asynchroner Task, startet der eigentliche Download (siehe Abbildung 19).

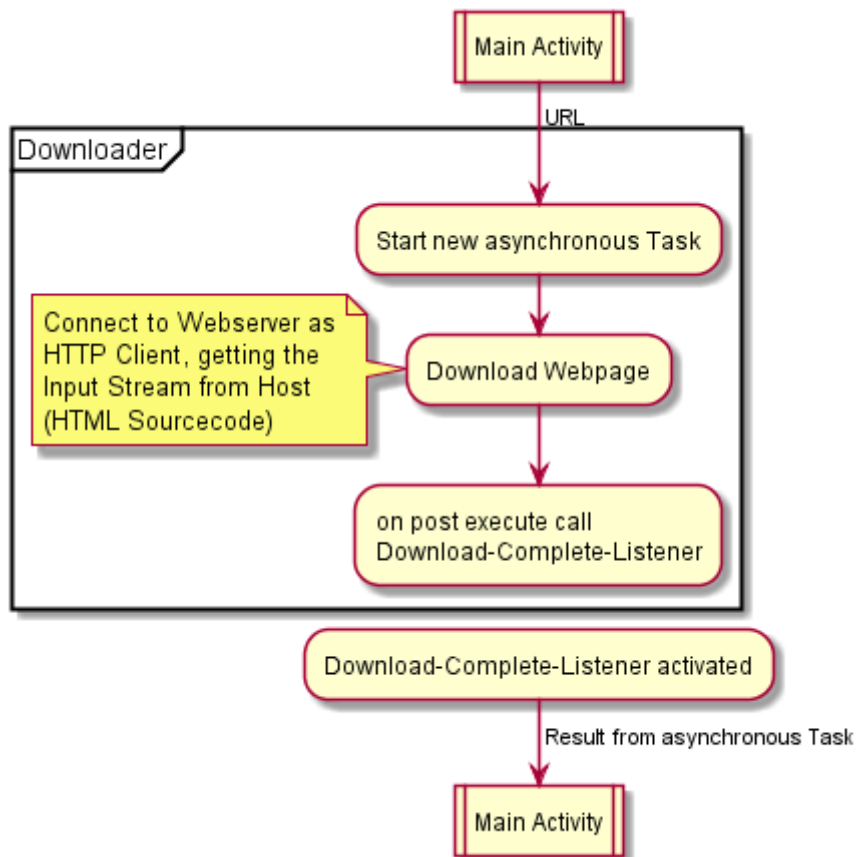


Abbildung 19: Prinzipieller Ablauf des Herunterladens der Website in einem Hintergrundthread und Rückgabe des Sourcecodes der Website an die Main Activity.

Während dem Download verbindet sich das Endgerät mit der übergebenen URL als HTTP-Client. Anschließend wird der einkommende Datenstrom gespeichert und als String zurückgegeben. Über das Interface `DownloadCompleteListener`, welches bereits in der Main Activity, also dem UI-Thread erstellt wurde (7.3.2), wird der heruntergeladene Sourcecode der Website an die Main Activity zurückgegeben.

8. ZUSAMMENFASSUNG

Nach dem systematischen Aufstellen von Hardware- und Software Konzepten, haben wir die nötigen Hardware Spezifikationen definiert. Die ausgewählten Komponenten haben wir in einem Testaufbau auf ihre Funktionalität geprüft und die Kompatibilität untereinander getestet. Die Umsetzbarkeit des Projektes ist von Seiten der Hardware somit sichergestellt.

Das Software Konzept wurde in verschiedene Entwicklungsstränge aufgeteilt und implementiert. Dafür haben wir eine Anwendung entwickelt, die das Auslesen der Sensoren und die Menüführung übernimmt. Des Weiteren ist durch uns eine Anwendung entwickelt worden, die es erlaubt Daten per WIFI an HTTP-Clients zu schicken. Zusätzlich dazu haben wir eine Smartphone App entwickelt, die die Darstellung der Daten an einem Hand-Held-Gerät übernimmt. Die Machbarkeit des Projektes ist auch auf Software Ebene sichergestellt und getestet.

9. AUSBLICK

Nach der Implementierung der einzelnen Software Entwicklungsstränge, ist es nötig diese in ein Gesamtsystem zu integrieren. Dadurch wird eine gemeinsame Software für die Datenermittlung und die Datensendung geschaffen.

Außerdem gilt es in einem nächsten Schritt, dass der bisherige Testaufbau direkt an einem Räucherofen installiert und ein geeignetes Gehäuse für die Hardware konstruiert werden muss.

Ein zusätzlicher Punkt ist die Anpassung der Smartphone App, um ein benutzerfreundlicheres Interface darzustellen.

10. ANHANG

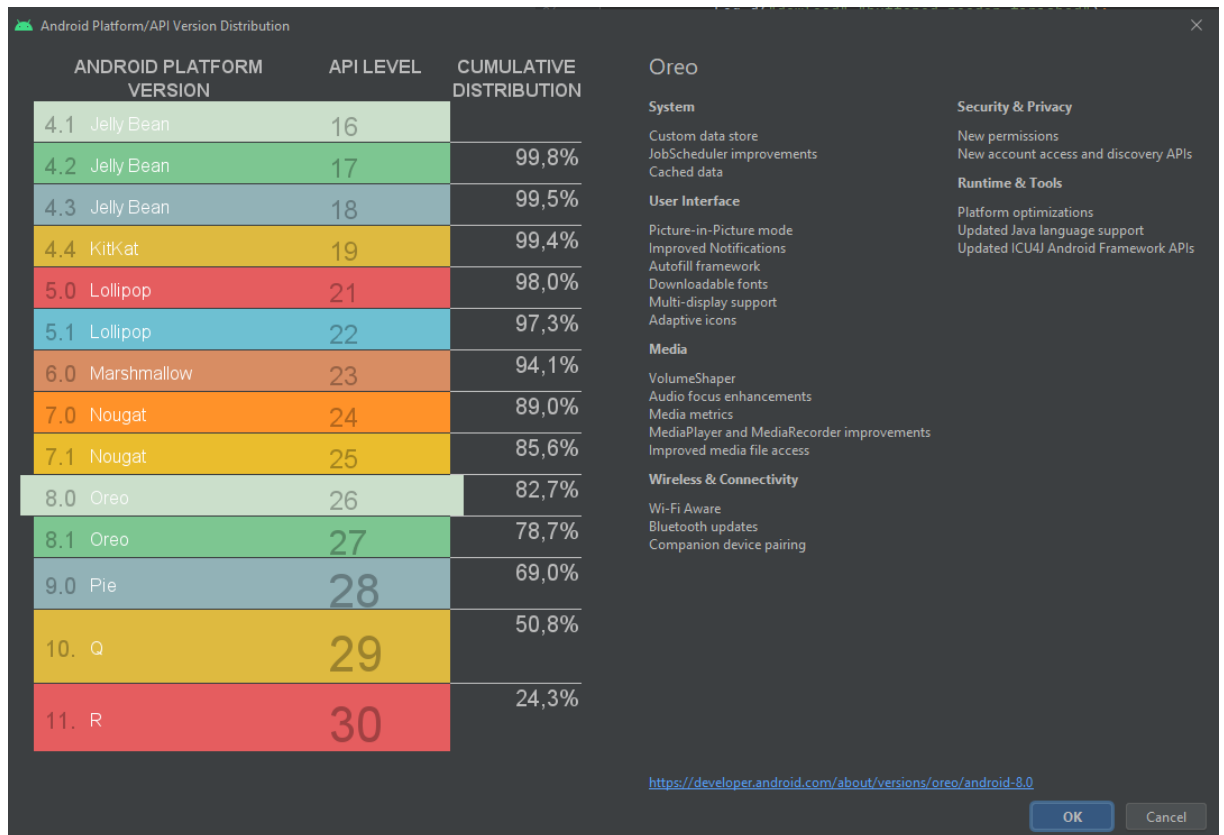


Abbildung 20: Verteilung der Android Versionen / API Level mit Anteil der Geräte die diese Anwendungen nutzen können aus der AndroidStudio App.

IDUINO for Maker's life

Flame Sensor Module(SE060)



1.Introduction

This is a flame sensor module that can be used to detect whether a flame source exists or not. It's sensitive to IR wavelength at 760nm~1100nm. Usually, the detection angle is about 60 degrees.

Specification

- Operation voltage: 5V for analog, 3.3V for digital
- Both digital and analog output pin
- Adjustable sensitive
- Detect IR wavelength: 760nm~1100nm
- Size: 30*15mm
- Weight: 3g

2.Pinout

Pin	Description
A0	Analog output pin, real-time output voltage signal on thermal resistance
D0	Digital output pin, output Low or High signal when the temperature reaches a certain threshold
+	Power(5V for analog, 3.3V for digital)
G	Ground

Abbildung 21 Datenblatt Flammensor

CHARACTERISTICS ARE AT $V_s=5.00$ WITH 4.7K OUTPUT TO MINUS WITH $T_A: -40^{\circ}\text{C}$ TO $+125^{\circ}\text{C}$ UNLESS OTHERWISE SPECIFIED

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
SENSITIVITY	$T_A = 25^{\circ}\text{C}$	3.00	3.125	3.25	mV/GAUSS
NULL	$T_A = 25^{\circ}\text{C}$	2.425	2.50	2.575	VOLTS
SUPPLY CURRENT	$T_A = 25^{\circ}\text{C}$		7	8.7	mA
OUTPUT CURRENT SOURCE SINK SINK	$V_s > 4.5$	1mA	1.5mA		
	$V_s > 4.5$.6mA	1.5mA		
	$V_s > 5.0$	1mA	1.5mA		
RESPONSE TIME			3 μ S		
OUTPUT VOLTAGE SWING VOM - VOM +	-B APPLIED	.4	.2		VOLTS
	+B APPLIED	$V_s - .4$	$V_s - .2$		VOLTS
B LIMITS FOR LINEAR OPERATION	-B MAX	-600	-670		GAUSS
	+B MAX	+600	+670		GAUSS
V_{null} DRIFT	$B = 0, T_A = 25^{\circ}\text{C}$ TO 125°C	-.06		+.06	% / $^{\circ}\text{C}$
V_{null} DRIFT	$B = 0, T_A = -125^{\circ}\text{C}$ TO $+150^{\circ}\text{C}$	-.08		+.08	% / $^{\circ}\text{C}$
SENSITIVITY DRIFT	$T_A = +25^{\circ}\text{C}$ TO $+150^{\circ}\text{C}$	-.01		+.05	% / $^{\circ}\text{C}$
SENSITIVITY DRIFT	$T_A = -40^{\circ}\text{C}$ TO $+25^{\circ}\text{C}$	0		+.06	% / $^{\circ}\text{C}$
LINEARITY	$B = -600$ TO $+600$	0	-1.0	-1.5	% OF SPAN
SUPPLY VOLTAGE	-40°C TO $+125^{\circ}\text{C}$	4.5	5.0	10.5	VOLTS
OPERATING TEMP	SEE MAX TEMPERATURE CHART	-40		+150	$^{\circ}\text{C}$

NULL SHIFT VERSUS TEMPERATURE

RATIO OF V_{null} TO V_s

Abbildung 22 Datenblatt Hall-Sensor



HYGROCHIP DIGITALER FEUCHTESENSOR HYT-221

1/3

Leistungsmerkmale

- ▶ Messbereich 0 ... 100 % rF, -40 ... 125 °C
- ▶ Genauigkeit $\pm 1,8\%$ rF, Temperatur $\pm 0,2$ °C
- ▶ Spritzwassergeschützt mit Schutzfilter
- ▶ Wasserdichter Membranfilter
- ▶ Präzise kalibriert und temperaturkompensiert
- ▶ Chemisch beständig, betauungsresistent
- ▶ Mechanisch robust
- ▶ Geringe Hysterese, kompensierter Linearitätsfehler und Temperaturdrift
- ▶ SIL-Anschlüsse, steckbar, RM 1,27mm
- ▶ I²C, Adresse 0x28 oder Alternativadresse
- ▶ Abmessungen 15,3 x 10,2 x 5,3 mm
- ▶ RoHS konform

Sowohl der Linearitätsfehler, als auch die Temperaturdrift werden on-Chip rechnerisch korrigiert, wodurch sich eine hervorragende Genauigkeit über einen weiten Einsatzbereich ergibt. Dank mechanischer Robustheit, hoher chemischer Beständigkeit und Betauungsresistenz sowie einer exzellenten Langzeitstabilität sind Feuchtemessung in der Sauna, Anwendungen im Außenbereich oder industrielle Anwendungen im Bereich Trocknungstechnik typische Einsatzbereiche.

Weitere Varianten und das ganze Spektrum der HYGROCHIP Produktserie finden Sie hier:

<http://www.ist-ag.com>

Typische Anwendungsgebiete

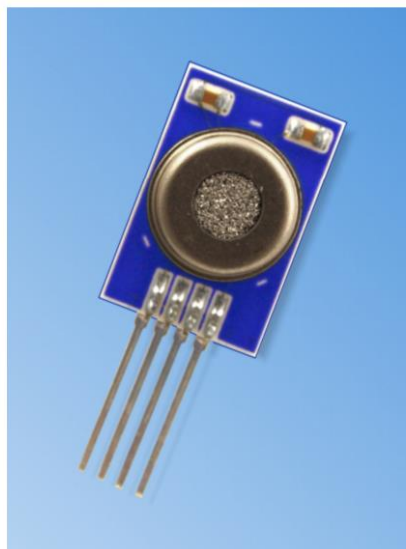
- ▶ Meteorologie
- ▶ Industrielle Trocknungstechnik
- ▶ Medizinische Geräte
- ▶ Luftfahrt
- ▶ Extremsport

Eigenschaften

Der HYT 221 - für kritische Einsatzgebiete.

Die runde Edelstahlhülse ist leicht in Gehäuseöffnungen einzubauen und kann mittels O-Ring gegen die Wandung abgedichtet werden. In Verbindung mit dem wasserdichten Membranfilter ergibt sich so ein Aufbau der selbst Spritzwasser widersteht und dichte Gehäusekonstruktionen ermöglicht, bei hoher Dynamik des Ansprechverhaltens.

Präzise kalibriert liefert der HYT 221 eine Genauigkeit von $\pm 1,8\%$ rF und $\pm 0,2$ °C. Wie alle Vertreter der HYGROCHIP Familie vereint der Sensor die Vorzüge eines präzisen, kapazitiven Polymer-Feuchtesensors mit der hohen Integrationsdichte und Funktionalität eines ASICs. Die im Sensor integrierte Signalverarbeitung bereitet die Messwerte komplett auf und liefert direkt die physikalischen Größen relative Feuchte und Temperatur über die I²C kompatible Schnittstelle als digitalen Wert. Das Modul ist im Werk präzise kalibriert und daher ohne Abgleich voll austauschbar.



Alle mechanischen Abmessungen gelten bei 25°C Umgebungstemperatur, falls nicht anders angegeben. ■ Alle Daten ausser der mechanischen Abmessungen dienen nur Informationszwecken und sind nicht als zugesicherte Eigenschaften aufzufassen. ■ Technische Änderungen ohne vorherige Ankündigung sowie Irrtümer vorbehalten. ■ Die Informationen auf diesem Datenblatt wurden sorgfältig überprüft und werden als richtig angenommen. ■ Keine Haftung bei Irrtümern. ■ Belastung mit Extremwerten über einen längeren Zeitraum kann die Zuverlässigkeit beeinflussen. Ausgabe 09/2011 Technische Änderungen vorbehalten!



INNOVATIVE SENSOR TECHNOLOGY

IST AG, Industriestrasse 2, CH-9630 Wattwil, Switzerland, Phone (+)41 71 987 73 73, Fax (+)41 71 987 73 77
e-mail info@ist-ag.com, www.ist-ag.com

Abbildung 23 Datenblatt Temperatur- und Feuchtigkeitssensor