

Diplomarbeit

vernetzte Brandmeldeanlage vBMA

Autor: Ulrich Möhl
Baumgartiweg 1
3752 Wimmis

Ausbildung: Dipl. Techniker Elektrotechnik HF

Studiengang: B-TEL-20-T-a

Diplomdozent: Herr Jörg Schenker

Abgabetermin: 30. Oktober 2023, 1700 Uhr

Präsentation: 11. November 2023, 1345 Uhr

Inhaltsverzeichnis

0	Einleitung	4
0.1	Beruflicher Lebenslauf des Verfassers	4
0.2	Themenwahl.....	5
0.3	Vorgaben der TEKO	5
0.4	Projektphasen.....	6
0.5	Kunde / Projektorganisation.....	6
0.6	Abgrenzungen	6
0.7	Management Summary	7
1	Projektinitialisierung.....	8
1.1	Themenbeschreibung.....	8
1.2	Richtziel.....	8
1.3	Endergebnisse.....	8
1.4	Erfolgskriterien.....	9
1.5	Pflichtenheft.....	9
1.5.1	Anforderungen Sensor.....	9
1.5.2	Anforderungen Zentrale.....	10
1.6	Hilfsmittel.....	11
1.7	Zeitplan / GANT Diagramm.....	12
2	Konzeption	13
2.1	IST – Zustand	13
2.2	SOLL – Zustand	15
2.3	Varianten	16
2.3.1	Mikrokontroller	16
2.3.2	Sensoren	20
2.4	Variantenwahl.....	25
2.4.1	Wahl der Mikrokontroller	25
2.4.2	Wahl der Sensorvariante	26
3	Realisierung	27
3.1	Systemaufbau.....	27
3.1.1	Beschreibung der Hauptfunktion.....	27
3.1.2	Beschreibung der Nebenfunktionen.....	28
3.1.3	Hauptkomponenten	29
3.2	Sensoren.....	30
3.2.1	Funktionsbeschreibung Sensor	30
3.2.2	Technischer Aufbau.....	31
3.2.3	Hardware.....	32
3.2.4	Software	37
3.2.5	Umsetzung	39
3.3	Zentrale.....	41
3.3.1	Funktionsbeschreibung Zentrale.....	41
3.3.2	Technischer Aufbau.....	42
3.3.3	Hardware.....	43
3.3.4	Software / Programmierung	49
3.3.5	Umsetzung	57

4	Abschluss	59
4.1	Wirtschaftlichkeit.....	59
4.1.1	Effektive Kosten.....	59
4.2	Kontrolle der Zielerreichung.....	62
4.2.1	Kontrolle der Zielerreichung der Erfolgskriterien	62
4.2.2	Kontrolle der Zielerreichung des Pflichtenheftes	63
4.3	Fazit / Lessons Learnt	65
Anhang A	Verzeichnisse	67
Anhang B	Sourcecode	73
Anhang C	Datenblätter und Zertifikate	84
Anhang D	Eigenständigkeitserklärung	91

0 Einleitung

Im Rahmen der Ausbildung zum Techniker HF in Elektrotechnik soll als Abschlussarbeit des Studiums eine Diplomarbeit erarbeitet werden. Die hier vorliegende Diplomarbeit zeigt die Entwicklung einer vernetzten Brandmeldeanlage (im weiteren Verlauf vBMA genannt) auf, welche auf handelsüblichen zertifizierten Rauchmeldern aufbaut und einen möglichen Alarm über einen kommerziellen und zuverlässigen Dienstleister an ein Mobile Phone weiterleitet. Die Betriebsdaten der vernetzten Rauchmelder sollen ebenfalls überwacht werden. Der Fokus dieser Arbeit liegt in der Entwicklung einer Schaltung, welche den genannten Zweck erfüllt und deren Dokumentation.

0.1 Beruflicher Lebenslauf des Verfassers



Name: Möhl
Vorname: Ulrich
Alter: 44 Jahre
Wohnort: Wimmis BE
Zivilstand: verheiratet
Kinder: 2

Seit August 2019	Systemspezialist IFASS / ECM, Kommando Cyber (Armee) <ul style="list-style-type: none">• Tätigkeit als Teilprojektleiter ECM "Electronic Counter Measures"• Erprobungen und Versuche an ECM-Systemen• Technische Unterstützung im Bereich ECM der Truppenverbände vor und während der Einsätze und Übungen der Armee;
Seit August 2009	Fachausbilder EKF an der EKF-Schule 64, Jassbach <ul style="list-style-type: none">• Planen und durchführen von Ausbildungen im Bereich ECM, EPM, Broadcast, Aktionsplanung, Aktionsführung, Funk, Richtstrahl und Fahrausbildungen• Kaderbetreuung und Coaching im Bereich technischer und taktischer Führung
Januar 2006 – Juli 2009	Fachausbilder Radargerätetechnik TAFLIR FULW S 95, Dübendorf
Januar 2004 – Dezember 2005	Ausbildung zum Berufsunteroffizier Berufsunteroffiziersschule BUSA, Herisau <ul style="list-style-type: none">• inkl. Erwerb Fachausweis Ausbilder
August 2003 – Dezember 2003	Zeitmilitär LW Na / Uem Schulen 43/243, Dübendorf
Juni 2003 – Juli 2003	Radio- und Fernsehetelektriker (temporär), Radio/TV Schenk, Wattenwil
Februar 2003 – Mai 2003	Militärdienst in Dübendorf
Dezember 2002 – Januar 2003	Radio- und Fernsehetelektriker (temporär), Radio/TV Schenk, Wattenwil
Oktober 2002 – November 2002	Militärdienst in Thun
Oktober 2000 – Oktober 2002	Radio- und Fernsehetelektriker, Radio/TV Kormann, Münsingen
Juli 2000 – Oktober 2000	Militärdienst in Dübendorf
August 1996 – Juli 2000	Ausbildung zum Radio- und Fernsehetelektriker Expert Radio Moser, Thun

0.2 Themenwahl

Es liegt mir viel daran bei Projektarbeiten ein Thema zu bearbeiten, welches mich beruflich oder persönlich interessiert und einen Mehrwert generiert. Bei der Suche nach möglichen Themen haben sich bei mir folgende 3 Themen für eine mögliche Diplomarbeit herauskristallisiert welche sich wie folgt klassifizieren lassen:

Projekt	Nutzen	Schwierigkeitsgrad	Bemerkungen
Fernsteuerung von taktischen Funkgeräten für Projektaktivitäten im Bereich ECM (elektronische Gegenmassnahmen)	mittel	mittel	Da die Fernsteuerung auf Steuersignale des Funkgerätes zugreifen muss, würde die DA einer Klassifizierung unterliegen, was die Bearbeitung verkompliziert.
Hochwasserfrühwarnsystem für einen Bach	gross	schwer	Thema war für eine DA mit einem anderen Studenten angedacht (2er Team). Auflagen der TEKO für eine Teamarbeit sind zu umfangreich und kaum umsetzbar.
Vernetzte Brandmeldeanlage auf Basis von kommerziellen Rauchmeldern	gross	mittel	Aufwändig aber gut realisierbar

Tabelle 1: Themenwahl

Da aus dem beruflichen Umfeld aus vertraulichkeitsgründen eine Diplomarbeit nur schwer zu realisieren wäre, bietet sich ein Projekt aus der Haustechnik/Automation perfekt an, da ich einen persönlichen Mehrwert bei geringen Kosten mit der vBMA generieren kann. Mir ist bewusst, dass die Technologie für solche Zwecke in vielen Ausführungen besteht. Jedoch ist es mein Anspruch dies selbst und nach eigenen Bedürfnissen zu entwickeln, aufzubauen und allenfalls in meiner bestehenden Hausinstallation/Automation zu integrieren (nicht Teil dieser Diplomarbeit).

0.3 Vorgaben der TEKO¹

- Thema aus dem Bereich der Elektrotechnik;
- Einzelarbeit oder unter Auflagen eine 2er Gruppe möglich;
- Themeneingabe mittels vorgegebenen Formulars via Extranet Upload;
- Vereinbarung zweier Betreuungstermine mit der Betreuungsperson für die Zielverfolgung;
- Abgabe eines Dokumentes als pdf-Datei via Upload Extranet und an die betreuende Person, allenfalls auf Wunsch gebunden im Format A4;
- Bezeichnung des Dokumentes: DA_Jahr_Name_Vorname_Klasse.pdf
- Vor der Präsentation ist eine Onlinepublikation mit den Eckdaten des Inhaltes der Diplomarbeit zu veröffentlichen. Die Website gibt Auskunft über Endergebnisse, Nutzen und Veranschaulichung;
- Dauer der Abschlusspräsentation 15min;
- Gesamtaufwand der Diplomarbeit sollte zwischen 150-250 Arbeitsstunden liegen;

¹ (TEKO Schweizerische Fachschule, 2023) Richtlinien Diplomarbeit Herbst 2023

- Die Gliederung der Dokumentation soll folgende Eckpunkte beinhalten:
 - Deckblatt
 - Inhaltsverzeichnis
 - Management Summary
 - Beruflicher Lebenslauf
 - Aufgabenstellung mit Pflichtenheft und Zieldefinition
 - Terminplan
 - Lösungswege
 - Quellenangaben und Literaturverzeichnis
 - Reflexion Weg zum Ziel
 - Schlusswort, Verdankungen sowie eine Eigenständigkeitserklärung
- Die Bewertung wird anhand des Bewertungsformulars für Diplomarbeiten der TEKO gemacht.

0.4 Projektphasen

Die vorliegende Arbeit beinhaltet die folgenden Projektphasen:

- Projektinitialisierung
- Konzeption
- Realisierung
- Abschluss

0.5 Kunde / Projektorganisation

Diese Diplomarbeit erarbeite ich ausschliesslich für private Zwecke. Das Produkt aus dieser Diplomarbeit soll mir als Privatperson dienen. Nutzer dieses Produktes ist ausschliesslich meine Familie. Aus diesem Grund vereinen sich sämtliche Projektrollen in dieser Diplomarbeit. Kunde und Projektorganisation werden daher von einer Person abgedeckt.

0.6 Abgrenzungen

- Die Diplomarbeit wird nicht aus allen Blickwinkeln sämtlicher Stakeholder betrachtet und analysiert. Somit wird auf das Rollenmodell nach Hermes verzichtet.
- Die Hausinterne Integration und die Kombination mit anderen Systemen (Markisen- und Dachwassersteuerung) ist nicht Bestandteil dieser Diplomarbeit. Es wird in einigen Punkten nur auf die Kombinations- und Integrationsmöglichkeit hingewiesen, da dies an den anderen Systemen auch noch marginale Änderungen zur Folge hat.
- Es wird nicht für alle eingesetzten Bauteile eine Variante ausgearbeitet. Die Varianten beschränken sich daher nur auf die Art und Form des Sensors und die dafür benötigten Mikrokontroller.
- Die in den Codes verwendeten Libraries werden in der vorliegenden Diplomarbeit nicht analysiert und auch nicht dokumentiert.
- Die verwendeten Nodes in Node-RED werden in dieser Diplomarbeit nicht in ihrer Funktionsweise beschrieben. Es wird ausschliesslich auf die Grundfunktion eines Nodes hingewiesen.
- Die AT-Kommandos der Firma Hayes Microcomputer Products werden in dieser Diplomarbeit nicht ausführlich analysiert und beschrieben. Es werden ausschliesslich die im Code verwendeten AT-Kommandos genauer erklärt.
- Das akustische Sirenenmodul wird nicht selbst entwickelt und wird daher nur rudimentär in dieser Diplomarbeit beschrieben.

0.7 Management Summary

Was ist, wenn es brennt, und niemand stellt dies fest? Ein Rauchmelder allein kann diesen Umstand nicht ändern. Daher habe ich mir mit der vorliegenden Diplomarbeit folgendes Ziel gesetzt:

- Den Umstand der sehr lokalen Wahrnehmung des Rauchmelders will ich mit einer zentralen Auswertung der verschiedenen durch die Vereinigung kantonaler Feuerversicherungen (VKF) zertifizierten Rauchmelder aus den Innenräumen (2 Kinderzimmer) sowie der Autogarage in einem Brandfall oder bei starker Rauchentwicklung optimieren und auf verschiedenen zuverlässigen Kanälen alarmieren, um so die Reaktionszeit für eine Intervention oder die Alarmierung der Feuerwehr zu verkürzen. Die vernetzte Brandmeldeanlage (vBMA) soll zu einem späteren Zeitpunkt beliebig im Bereich der Hausautomation weiterentwickelt werden können.

Dafür wird in einer SOLL/IST-Analyse ein erster Ansatz einer Systemarchitektur für die vBMA aufgezeigt und Varianten für die Umsetzung erarbeitet. Dabei habe ich mich über eine Bewertungsmatrix für eine Variante entschieden, wobei ein Rauchmelder mit einem Mikrokontroller ergänzt wird, um Alarme und Betriebszustände über verschiedene Kommunikationskanäle abzusetzen. Die Varianten beinhalten auch die Wahl der zu verwendenden Mikrokontroller in den Sensoren und in der Zentrale.

Die Systemarchitektur wird in der ersten Phase der Entwicklung weiter verfeinert und mit entsprechender Hard- und Softwarelösungen ergänzt. Die Hauptfunktion der vBMA ist einerseits eine lokale Alarmierung mit visuellen und akustischen Elementen und andererseits eine dezentrale Alarmierung über ein Mobilfunkmodul. Als Nebenfunktion der vBMA werden die Betriebszustände der ergänzten Rauchmelder ausgewertet und auf einem Dashboard von Node-RED (IoT Entwicklungsumgebung), übersichtlich dargestellt. Das Dashboard ist auch ausserhalb des WiFi-Netzwerkes über eine App erreichbar. Die Programmierung der Mikrokontroller von Sensor und Zentrale wird in der Arduino IDE Entwicklungsumgebung realisiert.

In der weiteren Entwicklung wird der Rauchmelder mit einem Mikrokontroller ergänzt, welcher den Alarmimpuls am Rauchmelder zum Aufwecken des Mikrokontrollers verwendet. Im Alarmfall wird eine Nachricht mit der Rauminformation mit dem MQTT-Protokoll (Message Queuing Telemetry Transport) über das WiFi-Netzwerk zur Zentrale übertragen. Zudem werden periodisch die Betriebsdaten (Betriebsspannung und RSSI-Pegel) zur Zentrale zur Auswertung übertragen. Nach dem Absetzen der Nachrichten wird der Mikrokontroller wieder in den Energiesparmodus versetzt. Die Ergänzungsschaltung ist auf einer eigens dafür entwickelten Printplatte mit einem entsprechenden Gehäuse realisiert.

Die MQTT-Nachrichten werden vom MQTT-Broker, welcher auf einem Raspberry Pi 3B aufgebaut ist, empfangen und an seine Abonnenten versendet. Die Abonnenten sind einerseits ein Alarm-Mikrokontroller für die lokale (LEDs und Sirenenmodul) und dezentrale Alarmierung über das Mobilfunkmodul mittels AT-Befehlen. Andererseits Node-RED für die Alarm- bzw. Betriebszustandsauswertung und Darstellung auf einem Node-RED Dashboard welches als Host ebenfalls den Raspberry Pi 3B verwendet. Kritische Betriebszustände werden zudem aus Node-RED auf das Mobiltelefon des Benutzers in WhatsApp übermittelt. Die Zentrale ist ebenfalls auf einer eigens dafür entwickelten Platine mit einer gemeinsamen externen Stromversorgung realisiert. Sämtliche Module sind mit dieser Printplatte miteinander verbunden und in einem Aluminiumgehäuse untergebracht.

Die effektiven Kosten der Entwicklung (Funktionsmuster) und Erstellung der Prototypen belaufen sich auf CHF 32'754.85. Die Kosten setzen sich aus Personal-, Sach- und Infrastrukturkosten zusammen und wurden den Arbeitspaketen entsprechend erfasst.

Die Zielerreichung (bestehend aus Erfolgskriterien und Pflichtenheft) konnte mit 99,6% erfolgreich erreicht werden. Die Abzüge sind primär aus der Entwicklungsphase zu verzeichnen. Dabei entspricht die Stromversorgung der Sensoren und die lokale Alarmierung nicht genau den Anforderungen und Vorstellungen des Benutzers und sind dementsprechend noch weiterzuentwickeln.

1 Projektinitialisierung

1.1 Themenbeschreibung

Vor einigen Jahren hatten meine Frau und ich die Möglichkeit, ein Einfamilienhaus zu erwerben. Wir haben das Haus nach den heutigen Standards und Richtlinien renoviert und umgebaut. Eine Brandmeldeanlage wurde bei der Renovation leider nicht eingebaut, da wir diesem Thema zu dem Zeitpunkt keinen grossen Stellenwert beimassen. Bei der Geburt unserer beiden Kinder erhielten wir von der Gebäudeversicherung (GVB) je einen Rauchmelder für die Kinderzimmer. Dieses kleine Präsent von der GVB liess uns etwas Umdenken. Bei den ersten Funktionstests dieser Rauchmelder habe ich festgestellt, dass die Alarmierung nur sehr lokal wahrnehmbar ist. Da unser Haus auf drei Stockwerken bewohnt wird, kann es dazu führen, dass der Alarm akustisch nicht wahrgenommen wird und eine Reaktion unsererseits oder die Alarmierung der Feuerwehr nicht rechtzeitig ausgeführt werden kann. Daher sollen die Rauch- und Brandmelder, welche auf dem Markt verfügbar und von der VKF freigegeben sind, mit einem Mikrokontroller ausgestattet werden wobei die Grundfunktionalität erhalten bleibt. Die Rauchmelder werden somit in das W-LAN eingebunden und von einem zentralen Mikrokontroller ausgewertet. Im Brandfall oder bei starker Rauchentwicklung soll der zentrale Mikrokontroller einen akustischen sowie visuellen Alarm auslösen und zudem über einen zuverlässigen (kostenpflichtigen) Dienstleister eine Mitteilung auf das Smartphone übertragen. Der zentrale Mikrokontroller und/oder Mikrocomputer soll auch die Möglichkeit bieten weitere Subsysteme der Hausautomation auszuwerten und zu steuern (Dachwasser- und Markisensteuerung aus Projekt- und Semesterarbeiten an der TEKO).

1.2 Richtziel

Den Umstand der sehr lokalen Wahrnehmung des Rauchmelders will ich mit einer zentralen Auswertung der verschiedenen durch die Vereinigung kantonaler Feuerversicherungen (VKF) zertifizierten Rauchmelder aus den Innenräumen (2 Kinderzimmer) sowie der Autogarage in einem Brandfall oder bei starker Rauchentwicklung optimieren und auf verschiedenen zuverlässigen Kanälen alarmieren, um so die Reaktionszeit für eine Intervention oder die Alarmierung der Feuerwehr zu verkürzen. Die vBMA soll zu einem späteren Zeitpunkt beliebig im Bereich der Hausautomation weiterentwickelt werden können.

1.3 Endergebnisse

Anz	Beschrieb	Datum
1	Funktionsmuster Rauchmelder mit Mikrokontroller (auf Steckbrett)	30. Okt. 2023 1700
1	Funktionsmuster vBMA-Zentrale mit separatem Mikrokontroller und 4G LTE-Modul zum Versenden von SMS (auf Steckbrett)	30. Okt. 2023 1700
3	Umgebaute Rauchmelder Prototypen mit selbst entwickelter Printplatte inkl Stromversorgung	30. Okt. 2023 1700
1	Prototyp vBMA Zentrale mit eingebautem Mikrokontroller und Mikrocomputer für die Datenauswertung und Darstellung, 4G LTE-Modul, Alarmierung optisch und akustisch aufgebaut auf einer selbst entwickelten Platine	30. Okt. 2023 1700
1	Microwebsite auf dem DA-Portal der TEKO	09. Nov. 2023 2345
1	Präsentation der Diplomarbeit und der Endergebnisse von 15'	11. Nov. 2023 1345
1	Vollständige Dokumentation mit pünktlicher Abgabe an den betreuenden Dozenten sowie das Sekretariat der TEKO via Upload im Extranet	30. Okt. 2023 1700

Tabelle 2: Endergebnisse

1.4 Erfolgskriterien

Kriterium	Zeitraum	Art
Am Präsentationstag liegt minimal ein funktionsfähiges Funktionsmodell von einem Sensor und einer Zentrale oder maximal ein funktionsfähiges Prototypsystem mit 3 Sensoren und einer Zentrale vor.	kurzfristig	MUSS
Das System erfüllt alle Kriterien, welche im Pflichtenheft aufgeführt sind.	kurzfristig	MUSS
Der Zustand des Sensoren-Systems kann über ein Dashboard überwacht werden.	kurzfristig	MUSS
In der DA werden alle effektiven Kosten einer Realisierung aufgeführt inklusive der Experimentierkosten.	kurzfristig	KANN
Die DA wird fristengerecht bis am 30. Oktober 2023, 1700 Uhr abgegeben.	kurzfristig	MUSS
Eine Microwebseite wird auf dem DA-Portal der TEKO bis am 09. November 2023, 2345 Uhr erstellt und veröffentlicht.	kurzfristig	MUSS
Die DA am 11. November 2023 1345 an der TEKO in Bern präsentiert wird	kurzfristig	MUSS
Der Zeitaufwand der DA im Bereich von 150-250 Stunden liegt	kurzfristig	MUSS
Wenn durch die DA im Schadenfall ein grosses Schadenausmass verhindert werden kann.	langfristig	MUSS

Tabelle 3: Erfolgskriterien

1.5 Pflichtenheft

Die Pflichtenhefte der Sensoren und der Zentrale werden übersichtshalber separat dargestellt. In der Anforderung ist ersichtlich um welche Baugruppe es sich handelt. Die Zahl ist die Identifikationsnummer der Anforderung. Die Kennzeichnung "F" markiert eine Funktionale Anforderung. Anforderung ohne diese Markierung beschreiben eine "nicht funktionale Anforderung".

1.5.1 Anforderungen Sensor

Anforderung	Beschreibung	Art
SEN_01	Der Sensor wird auf einem von der Vereinigung kantonaler Feuerversicherungen (VKF) zertifizierten Sensor/Rauchmelder aufgebaut;	MUSS
SEN_02F	Die Grundfunktion des zertifizierten Sensors muss zwingend erhalten bleiben (lokaler akustischer / optischer Alarm)	MUSS
SEN_03F	Im Brandfall werden so lange wie möglich Brandalarme via W-LAN abgesetzt.	MUSS
SEN_04F	Bei einem Fehlalarm muss der Sensor am Objekt zu Kontrollzwecken zurückgesetzt werden.	KANN
SEN_05F	Die Stromversorgung des Mikrokontrollers muss autonom aufgebaut werden, damit die Grundfunktion des Rauchmelders nicht beeinträchtigt wird.	MUSS
SEN_06F	Die minimale Betriebsdauer eines Sensors beträgt 1 Monat, bevor die Batterie des Mikrokontrollers getauscht oder aufgeladen werden muss	KANN
SEN_07F	Betriebsparameter des Mikrokontrollers werden periodisch alle 2-4 Stunden der Zentrale zur Auswertung übertragen.	MUSS
SEN_08F	Die Betriebsparameter werden über das MQTT (Message Queue Telemetry Transport) – Protokoll als MQTT-Publisher-Message übertragen.	MUSS
SEN_09	Das Funktionsmuster des Sensormikrokontrollers wird auf einem Breadboard aufgebaut.	MUSS
SEN_10	Die Sensorprototypen werden auf einer eigens dafür entwickelten Platine aufgebaut.	KANN
SEN_11	Der Mikrokontroller und die zusätzlich dafür benötigte Elektronik werden in einem zusätzlichen Gehäuse untergebracht.	KANN
SEN_12	Der Sensor kann in allen Wohnräumlichkeiten sowie der Garage uneingeschränkt eingesetzt werden (ausgenommen Badezimmer)	KANN

Tabelle 4: Pflichtenheft / Anforderungen Sensor

1.5.2 Anforderungen Zentrale

Anforderung	Beschreibung	Art
ZEN_01	In der Zentrale ist der MQTT-Broker (Datenverteiler) einerseits für die Alarmweitergabe und andererseits für die Betriebsdatenauswertung untergebracht.	MUSS
ZEN_02	Die Betriebsdatenauswertung wird zwecks Datenverfügbarkeit zentral auf einem Mikrocomputer realisiert und ist Bestandteil der Zentrale.	MUSS
ZEN_03F	Die Auswertung der Betriebsdaten wird auf einem übersichtlichen, lokal wie dezentral verfügbaren Dashboard visualisiert.	MUSS
ZEN_04F	Der Zustand des Mikrokontrollers in den Sensoren welcher Alarmer weitergibt, muss auch ausserhalb des W-LANs überwacht werden können.	MUSS
ZEN_05F	Kritische Betriebsparameter des Mikrokontrollers der Sensoren werden über einen geeigneten Kommunikationskanal zeitnah dem Benutzer auf sein Mobiltelefon übermittelt.	KANN
ZEN_06	Die Auswertung der Alarmnachrichten (MQTT-Subscriber-Message) wird aus technischen Gründen auf einem separaten Mikrokontroller realisiert, welcher sämtliche Alarmierungen steuert.	MUSS
ZEN_07F	Alarmer werden akustisch wie auch optisch für die erweiterte lokale Wahrnehmung wiedergegeben.	MUSS
ZEN_08F	Ein ausgelöster Alarm kann nur an der Zentrale zurückgesetzt werden.	KANN
ZEN_09F	Nur die Alarmer werden über einen kostenpflichtigen Dienstleister über ein Mobilfunksystem (3G/4G/5G) zum Benutzer als SMS übertragen.	MUSS
ZEN_10F	In der Alarmnachricht ist ersichtlich in welchem Raum der Alarm ausgelöst wurde	MUSS
ZEN_11	Die Stromversorgung der Zentrale erfolgt über ein externes Netzteil auf 5VDC	KANN
ZEN_12	Das Funktionsmuster der Zentrale wird auf einem Breadboard aufgebaut, welches an das Mobilfunkmodul angeschlossen wird. Der MQTT-Broker wird beim Funktionsmuster autonom betrieben.	MUSS
ZEN_13	Der Prototyp der Zentrale wird auf einer eigens dafür entwickelten Platine aufgebaut.	KANN
ZEN_14	Der Mikrocomputer, der Mikrokontroller sowie das Sirenenmodul und die zusätzlich dafür benötigte Elektronik werden in einem Gehäuse untergebracht. Antennen werden aus dem Gehäuse herausgeführt.	KANN
ZEN_15	Es werden verschieden Kommunikationstechnologien verwendet	MUSS

Tabelle 5: Pflichtenheft / Anforderungen Zentrale

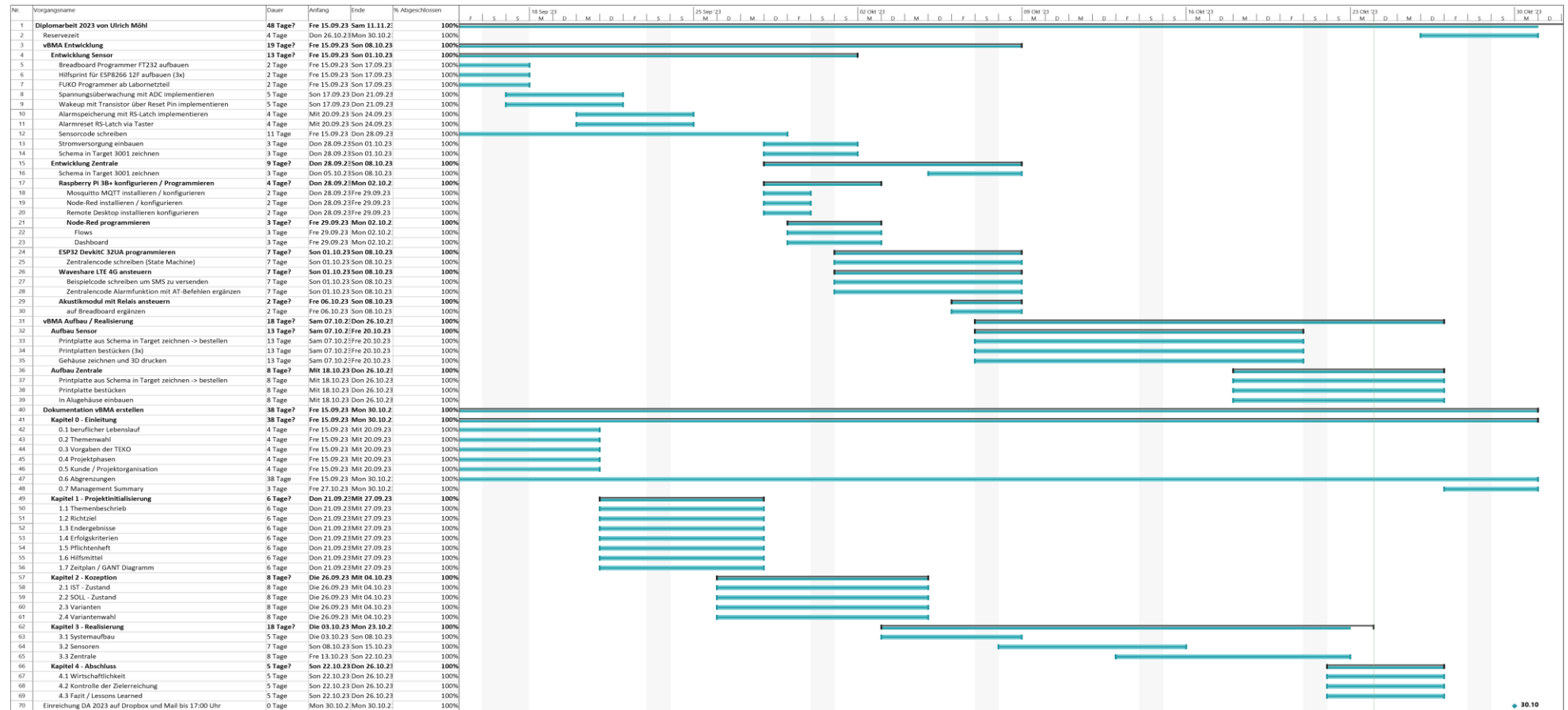
1.6 Hilfsmittel

Für die Erarbeitung der Diplomarbeit werden folgende Hilfsmittel eingesetzt:

Hilfsmittel	Zweck	Quelle
Microsoft Word (Office 365)	Textbearbeitung	Office 365
Microsoft Power Point (Office 365)	Präsentationen	Office 365
Microsoft Visio (Office 365)	Technische Illustrationen	Office 365
Microsoft Excel (Office 365)	Tabellenerfassung / Log Buch	Office 365
Microsoft Project (Office 365)	Projektplanung	Office 365
Microsoft OneNote for Windows 10	Notizen	Microsoft Windows 10
Microsoft Edge	Internetforschung	Microsoft Windows 10
Arduino IDE 2.2.1	Programmierung ESP8266 / ESP32	www.arduino.cc
Target 3001 V31	Schaltpläne / Printlayout	www.ibfriedrich.com
Autodesk Inventor 2017	Gehäuseanfertigung für 3D Print / CNC	www.autodesk.com
DeskProto 6.1	CAM-Software für CNC-Bearbeitung	www.filou.de
WinPC-NC USB	Maschinensoftware für CNC-Fräse	www.filou.de
Node-RED v3.0.2	Programmiertool für Schnittstellen	https://nodered.org
Raspberry Pi Imager	Erstellung Raspberry Pi Image	www.raspberrypi.com
Eclipse Mosquitto	MQTT-Broker auf dem Raspberry Pi	https://mosquitto.org
xRDP	Remote Desktop Verbindung zum Raspberry Pi	www.xrdp.org
2-Kanal Labornetzteil RND-320 KD3305	Testbetrieb Sensoren / Zentrale	www.distrelec.ch
2-Kanal Oszilloskop Tektronix TBS1032	Messungen / Veranschaulichungen	www.distrelec.ch
Multimeter Voltcraft VC-440	Messungen	www.conrad.ch
Multimeter Metex M-3850D	Messungen	
LötKolben / Heissluftgerät	Lötarbeiten	
Diverses Handwerkzeug	Mechanische Arbeiten	

Tabelle 6: Hilfsmittel

1.7 Zeitplan / GANT Diagramm



2 Konzeption

2.1 IST – Zustand

Das in den 50er Jahren erbaute Haus, welches wir 2015 kaufen und nach neusten Standards renovieren konnten, verfügt über keinen technischen Brandschutz. Die verbauten Materialien erfüllen alle Normen² welche von der VKF gefordert sind. Diese Normen wurden jedoch nur von den neu verbauten Materialien erfüllt, die Bausubstanz wie Gebäudehülle und tragende Konstruktionen stammen nach wie vor aus den 50er Jahren und wurden bei der Renovation nicht behandelt oder ersetzt. Somit ist in den Wohnräumen, welche nur leicht renoviert wurden, immer noch ein minimales Brandrisiko vorhanden. Diesem Brandrisiko beugen wir mit präventiven Brandschutzmittel wie Feuerlöschdecke und einem Feuerlöscher (Lightwater) vor. Zudem wird jährlich der Kaminfeger für die Reinigung des Kamins beauftragt. Sämtliche elektrischen Installationen wurden bei der Renovation ersetzt. Somit konnte auch dieses Brandrisiko stark minimiert werden. Ein Brandschutzkonzept für einfache Einfamilienhäuser ist von der VKF nicht gefordert. Dies wird erst ab einer bestimmten Geometrie (Grösse) und der Anzahl sich darin befindenden Personen (>300) gefordert.

Durch die Geburt unserer beiden Kinder erhielten wir von der Gebäudeversicherung Bern je einen Rauchmelder der Marke HEKATRON, Typ Genius H, welcher auch von der VKF zertifiziert wurde (siehe Anhang C). Diese wurden in den beiden Kinderzimmern angebracht und verrichteten bis heute ihren Zweck. Die Rauchmelder wurden regelmässig mit einem Rauchmelder-Testspray sowie der Testtaste überprüft.



Abbildung 1: Rauchmelder HEKATRON Genius H

Die Lautstärke der Alarme ist subjektiv nur sehr lokal wahrnehmbar. Die Faustformel³ für die Schalldruckberechnung besagt, dass mit der Halbierung der Strecke zwischen Schallquelle und dem Messpunkt oder Zuhörer die Lautstärke verdoppelt wird was 6dB entspricht. Ein kleiner Messaufbau mit einem nicht geeichten Schallmessgerät und nicht idealen Schallbedingungen verdeutlicht dies. Der Rauchmelder wird im Raum platziert und mit 1,5m Abstand das Messgerät positioniert. Die erste Messung ergab einen Schalldruckpegel von 86dB. Wird nun die Distanz verdoppelt und es kommt noch eine Türe als Schalldämpfung dazwischen ergab bei der zweiten Messung nur noch einen Wert von 72dB.

² (Vereinigung Kantonalen Feuerversicherungen VKF, 2015) Gemäss den geltenden Vorschriften und Normen

³ (Sengpiel, 2010) Abnahme des Schallpegels in Abhängigkeit von der Entfernung / Faustformel

6dB entsprechen der Verdoppelung der Distanz und weitere 8dB fallen an der Tür ab. Wenn nun die Distanz weiter vergrößert wird, fällt der Schalldruck rasch in eine Größenordnung von 60-70dB was der klassischen Zimmerlautstärke⁴ oder einer normalen Konversation entspricht. Somit ist klar ersichtlich, dass die Rauchmelder tagsüber rasch überhört werden können und keine Sofortmassnahmen eingeleitet werden können. Die Grafik zeigt den primitiven Schallmessaufbau.

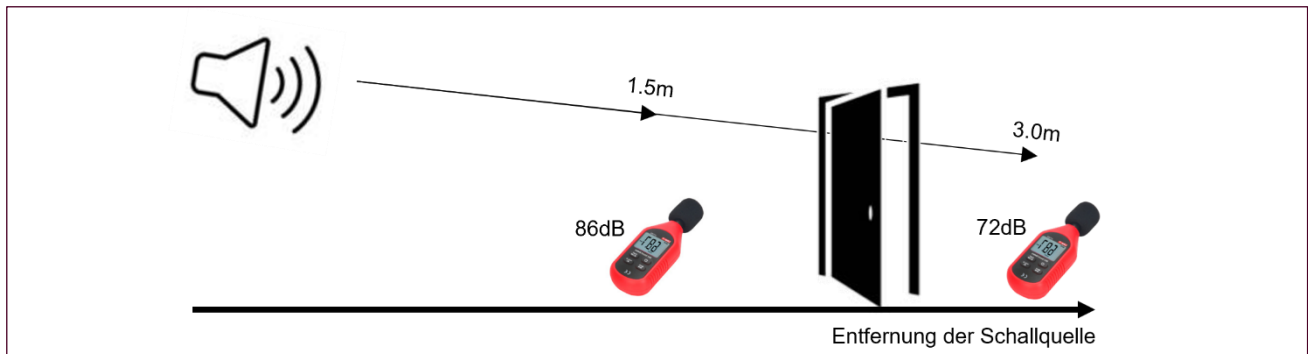


Abbildung 2: primitiver Schallmessaufbau

Die neu erbaute Garage, welche über eine kleine Werkstatt und 2 Autoabstellplätze verfügt, wird bis heute nicht mit einem technischen Brandschutz überwacht, da dieser im Brandfall oder bei starker Rauchentwicklung wie bereits erläutert nur sehr lokal wahrgenommen werden kann. In der Garage werden auch diverse leicht brennbare Reinigungsmittel und Flüssigkeiten gelagert, welche das Brandrisiko dementsprechend erhöhen.

Sämtliche Räumlichkeiten sowie die Garage liegen im Empfangsperimeter des W-LAN Routers. Die Garage liegt mit -72dBm und trockenen Wetterverhältnissen eher an der Grenze von einem guten WiFi Signalpegel. Die Situation könnte sich bei nassen oder schneereichen Wetterverhältnissen noch etwas verschärfen. Jedoch sollten Signalpegel bis -80dBm durchaus für eine Verbindung ausreichen. Die Abbildung 3 zeigt die Parzelle, in welcher das Wohnhaus und die Garage liegen. Weiter ist der Standort des W-LAN Routers eingezeichnet (Symbol). Dieser befindet sich im Innenraum.

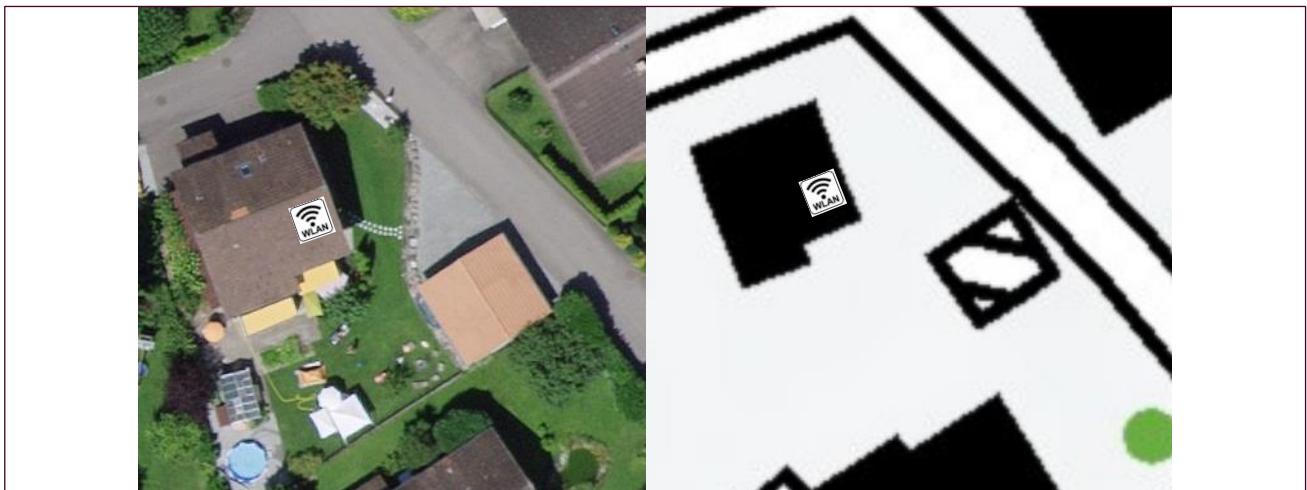


Abbildung 3: Situationsplan Baumgartweg 1 in 3752 Wimmis

⁴ (Beuth/Hanebuth/Kurz, 1996) Kapitel 7.3 Schallempfindung durch das Ohr

2.2 SOLL – Zustand

Die bestehenden Rauchmelder sollen nun miteinander zu einer vernetzten Brandmeldeanlage verbunden, und ausgewertet werden. Die abgesetzten Alarme sollen einerseits lokal akustisch und visuell und über ein Mobilfunkmodul an ein Mobiltelefon weitergeleitet werden. Um die Rauchmelder oder Sensoren miteinander zu vernetzen, werden diese mit einem Mikrokontroller ergänzt, welche sich mit den W-LAN verbinden. Alarme und auch die Betriebszustände des Mikrokontrollers werden periodisch alle 2-4 Stunden über das MQTT-Protokoll an einen lokalen MQTT-Broker gesendet (als Publisher-Nachricht).

In der Zentrale werden die Nachrichten der Sensoren zentral als Topics (Themen) den Abonnenten zur Verfügung gestellt (als Subscriber-Nachricht). Die Alarmtopics werden aus programmiertechnischen Gründen separat von einem Microcontroller ausgewertet, damit dieser die Alarme lokal weitergeben kann. Der sogenannte Alarm-Mikrokontroller steuert einerseits über eine serielle Schnittstelle das Mobilfunkmodul mittels AT-Kommandos an, damit eine Kurzmitteilung (SMS) mit einer Alarmnachricht an ein Mobiltelefon versendet werden kann und andererseits wird ein akustischer und visueller Alarm lokal ausgelöst. In der Kurzmitteilungsnachricht ist ersichtlich aus welchem Raum der Alarm ausgelöst wurde. In einem Notfall ist die technische Verfügbarkeit eines Kommunikationsdienstleisters essenziell. Daher wird die Alarmierung über einen kostenpflichtigen Dienstleister und der entsprechenden Technologie (3G/4G Mobilfunk) sichergestellt.

Die Betriebszustandstopics und die Alarmtopics der Mikrokontroller der Sensoren werden zusätzlich zur Visualisierung in einem Node-RED Dashboard übersichtlich dargestellt. Die Betriebszustände sind sowohl lokal wie auch dezentral über eine App auf dem Mobiltelefon einsehbar. Kritische Betriebszustände wie Unterspannung oder ein zu geringer WiFi-Signalpegel (auch RSSI-Pegel genannt) werden auf einem anderen Kommunikationskanal an ein Mobiltelefon gesendet.

Im Falle eines Fehlalarmes muss der lokal ausgelöste Alarm an der Brandmeldezentrale über einen Taster zurückgesetzt werden, damit der akustische und visuelle Alarm ausgeschaltet wird. Der Sensor, welcher den Alarm ausgelöst hat, muss ebenfalls am Sensor mit einem Taster zurückgesetzt werden, damit eine vernünftige Kontrolle der Räumlichkeiten durchgeführt wird.

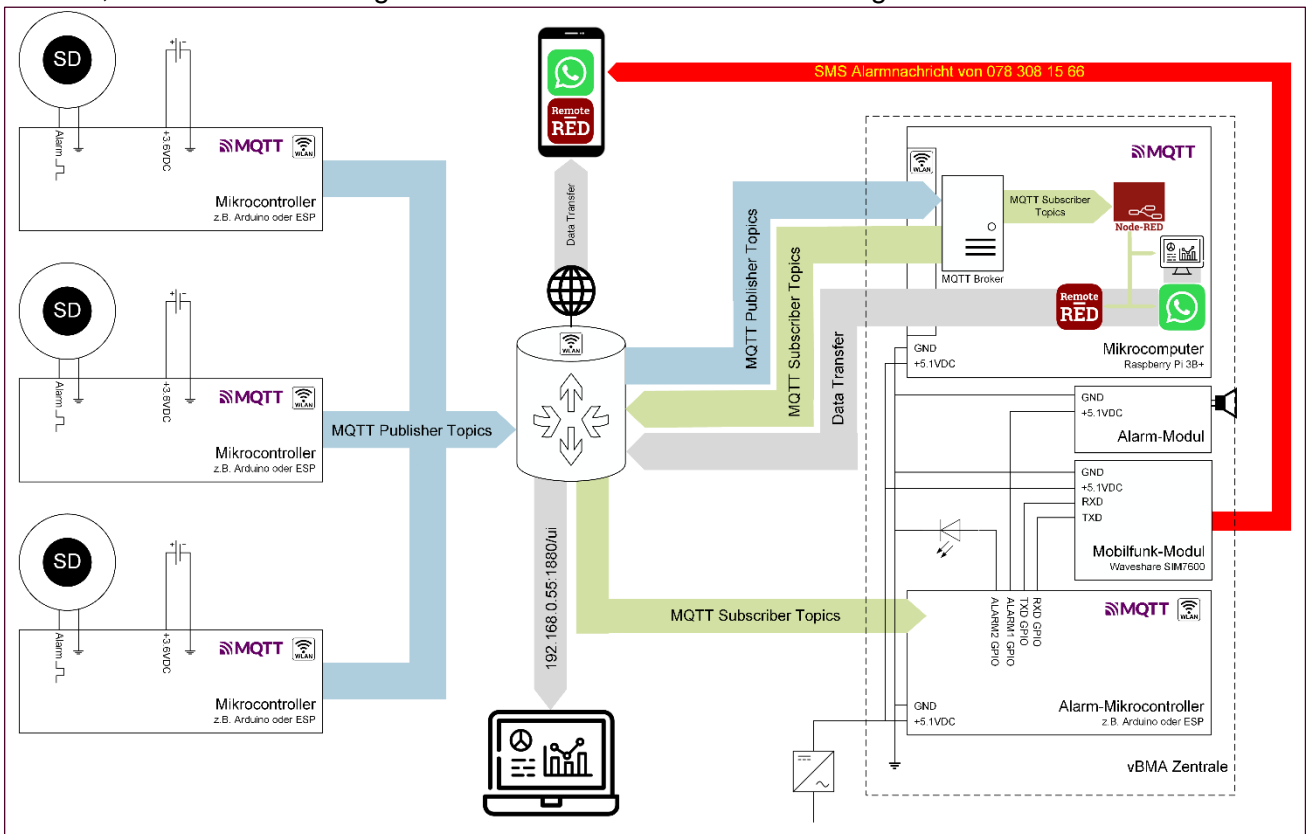


Abbildung 4: Prinzipschema vBMA / SOLL Zustand

2.3 Varianten

Um die Sensoren und die Zentrale zu realisieren, gibt es eine Vielzahl an verschiedenen Möglichkeiten dies umzusetzen. Eine zentrale Rolle für eine Realisierung spielen einerseits die Mikrokontroller und andererseits die Art von Sensoren (Brand/Rauchmelder), welche den in Kapitel 1.5 definierten Anforderungen gerecht werden müssen.

2.3.1 Mikrokontroller

Für die vBMA werden sowohl in den Sensoren wie auch in der Zentrale Mikrokontroller eingesetzt. Die Anforderungen an die Mikrokontroller, welche bei der Realisierung verwendet werden sollen, müssen mit der Arduino IDE programmiert werden können, verfügen über einen eingebaute WiFi-Chip und unterscheiden sich im Verwendungszweck (Sensor oder Zentrale). Es geht darum einen geeigneten Mikrokontroller für die Ergänzung der Sensoren sowie für die Zentrale zur Alarmauswertung und Kommunikation zu finden. Für die Codeentwicklung sollen einfach zu programmierende Entwicklungskits verwendet werden, bei welchen die Integration noch keine Rolle spielt. Die in den folgenden Unterkapitel beschriebenen Mikrokontroller standen aus meinem Fundus zur Verfügung.

2.3.1.1 Arduino Nano 33 IoT

Der Arduino Nano 33 IoT kombiniert die Leistungsmerkmale der Arduino Nano Familie mit drahtloser Konnektivität. Zudem verfügt er über diverse Onboard-Sensoren, welche jedoch für die vBMA von keiner Bedeutung sind. Für die Energieeffizienz unterstützt der Arduino Nano 33 IoT verschiedene Energiesparmodi wie Sleep-Mode, Standby-Mode, Hibernate-Mode und Idle-Mode und verschiedene Aufwachmöglichkeiten wie externe Interrupt, Reset-Signale oder Zeitgesteuerte Interrupts.

Meine Erfahrung aus verschiedenen Projekten während dem Studium haben aufgezeigt, dass der Arduino Nano 33 IoT wohl einfach in der Programmierung ist und dank der eingebauten Sensoren ein breites Spektrum an Anwendungen bietet. Darunter litt jedoch die Zuverlässigkeit des Mikrokontrollers, welche sich darin äusserte, dass sich der Mikrokontroller aufgehängt hatte. Diese Eigenschaft ist ein grosser Minuspunkt für den Einsatz in einer Brandmeldeanlage.

Spezifikationen⁵

Betriebsspannung:	5-21VDC
I/O Spannung:	3.3VDC
Analog / Digital-Converter (ADC):	an allen analogen Eingängen
Interrupts (Wake Up):	an allen digitalen Eingängen (zusätzlicher Reset Pin)
Kommunikation:	UART, I2C, SPI
Stromverbrauch (Sleep):	6mA ⁶
Batteriebetrieb möglich	Ja (mit grosser Kapazität möglich)

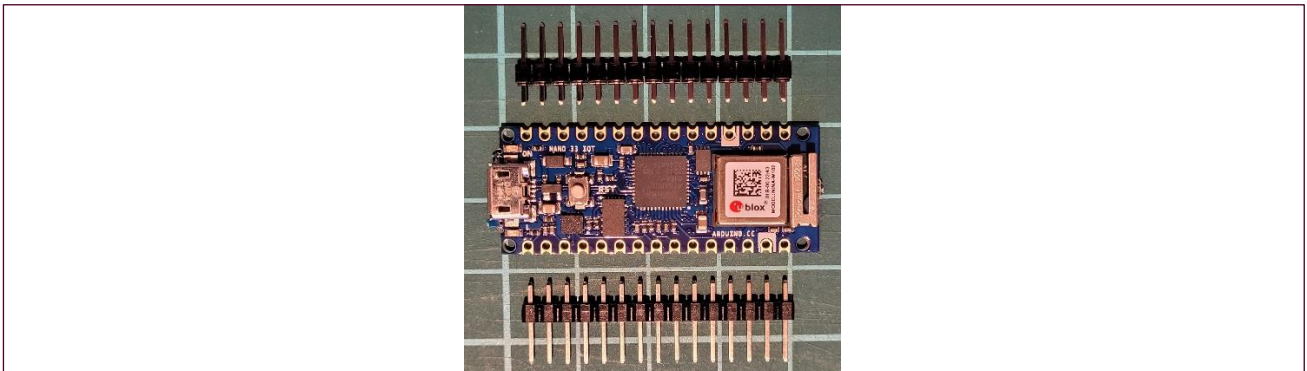


Abbildung 5: Arduino Nano 33 IoT

⁵ (Arduino.cc, 2023) technische Spezifikationen

⁶ (Staquet, 2021) How to save power with the Arduino Nano 33 IoT by sleeping

2.3.1.2 ESP8266 DevkitC und NodeMCU V2

Die ESP8266 Entwicklungskits einerseits von Espressif oder auch von Joy-It eignen sich bestens für die Codeentwicklung, da die Beschaltung des ESP8266 Chips bereits auf dem Board enthalten ist. Diese Entwicklungskits eignen sich daher, wenn bei späteren Projekten die schlanken und Stromsparenden Varianten des ESP8266 eingesetzt werden (7 oder 12E/F oder Wemos D1). Die Entwicklungskits sind auch von der Lochrasterung (2.54mm) her Breadboardkompatibel aufgebaut. Auf dem Board ist der USB/TTL Konverter-Chip, ein Spannungsregler 3,3V sowie Reset- und Flash Taste bereits enthalten. Alle Anschlüsse des ESP8266 Chips können über zwei 15er Stiftleisten angesteuert werden

Bei der Codeentwicklung muss daher zwingend das Pinout von beiden Versionen beachtet werden, wenn dies später auf den kleineren Versionen eingesetzt werden soll.

Die Zuverlässigkeit der ESP8266 Familie ist subjektiv besser, was auch aus der ESP-Community zu entnehmen ist.

Spezifikationen⁷

Betriebsspannung:	3.3VDC und 5VDC via USB
I/O Spannung:	3.3VDC
Analog / Digital-Converter (ADC):	1 an GPIO A0
Interrupts (Wake Up):	programmierbar auf alle GPIO, in deepsleep nur Reset-Pin
Kommunikation:	UART, I2C, SPI
Stromverbrauch (deepsleep):	<20 μ A ⁸
Batteriebetrieb möglich	Ja (mit grosser Kapazität möglich)

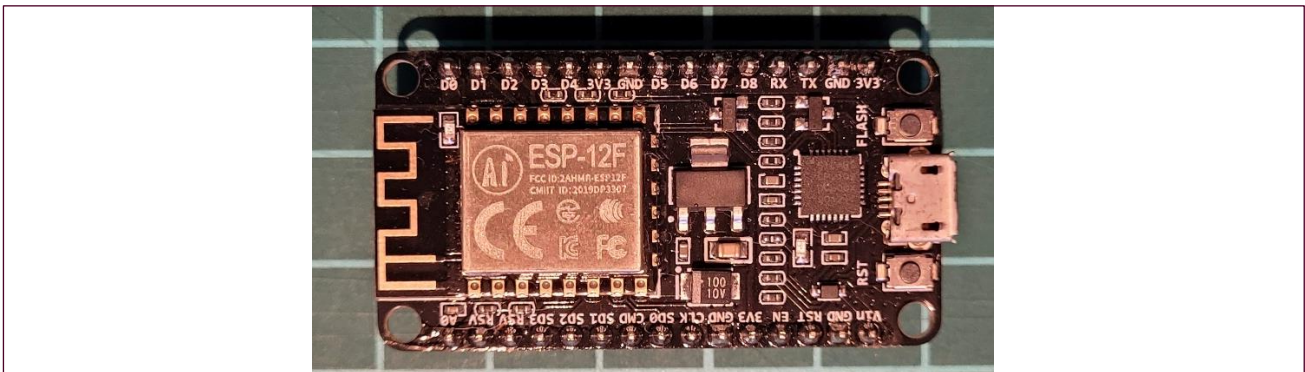


Abbildung 6: ESP8266 NodeMCU V2

2.3.1.3 ESP8266 12F

Die kompakteste Version des ESP8266 ist diejenige des 12F. Auf diesem SOC (System on a Chip) ist nur der 32Bit Mikroprozessor mit der WiFi Anbindung integriert. Dieser verfügt über 22 Anschlüsse (GPIO) welche Beschaltet werden können. Jedoch lassen sich diese schlanken Versionen schwieriger programmieren und sind nicht mit den gängigen Entwicklerboards kompatibel. Dafür sind auf dem Markt entsprechende Breakout-Boards erhältlich damit mit diesen Chips gearbeitet werden kann. Dieser kleine Chip eignet sich daher durch die geringen Abmessungen und den tiefen Stromverbrauch optimal für IoT Anwendungen, welche über das W-LAN angebunden werden sollen.

⁷ (www.joy-it.net, 2023) technische Daten

⁸ (Espressiv Systems, 2023) Eigenschaften der ESP8266 Serie

Spezifikationen⁹

Betriebsspannung:	3.3-3.6VDC
I/O Spannung:	3.3VDC
Analog / Digital-Converter (ADC):	1 an GPIO A0
Interrupts (Wake Up):	programmierbar auf alle GPIO, in deepsleep nur Reset-Pin
Kommunikation:	UART, I2C, SPI
Stromverbrauch (deepsleep):	10µA – 170mA (Peak bis 500mA)
Batteriebetrieb möglich	Ja

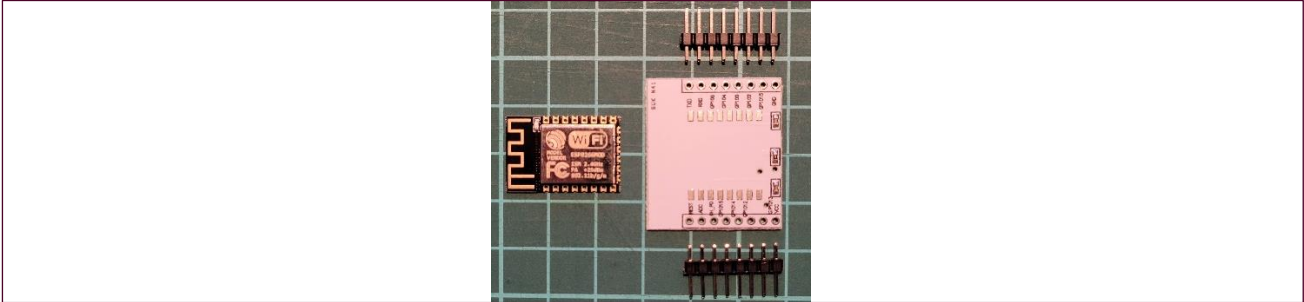


Abbildung 7: ESP8266 12F mit Breakout-Board

2.3.1.4 ESP32 NodeMCU (Joy-It)

Die ESP32 Chipfamilie ist die Nachfolgeneration der ESP8266 Chips. Diese sind leistungstärker und verfügen über 2 Kerne, welche mit bis zu 240MHz getaktet sind. Zudem wurde der interne Speicher im Gegensatz zu den ESP8266 deutlich erhöht.

Das ESP32 NodeMCU ist wie beim ESP8266 NodeMCU auch für die Codeentwicklung geeignet, da alle für die Programmierung benötigte Hardware auf dem Board vorhanden ist. Das Entwicklungsboard ist, auch wenn nur knapp, mit den Breadboard Lochraster kompatibel. Die Beschaltung der GPIOs muss unter dem Board herausgeführt werden, damit dies auf einem Standard-Breadboard noch verwendet werden kann.

Spezifikationen¹⁰

Betriebsspannung:	5VDC via USB, 7-12V via VIN, 3.3VDC via 3V3 Pin
I/O Spannung:	3.3VDC
Analog / Digital-Converter (ADC):	13
Digitale I/O:	19
Interrupts (Wake Up):	programmierbar auf alle GPIO
Kommunikation:	2 x UART, 1 x I2C, 1NodeMCU x SPI
Stromverbrauch (deepsleep):	150µA ¹¹ bis 240mA im Betrieb
Batteriebetrieb möglich	Ja

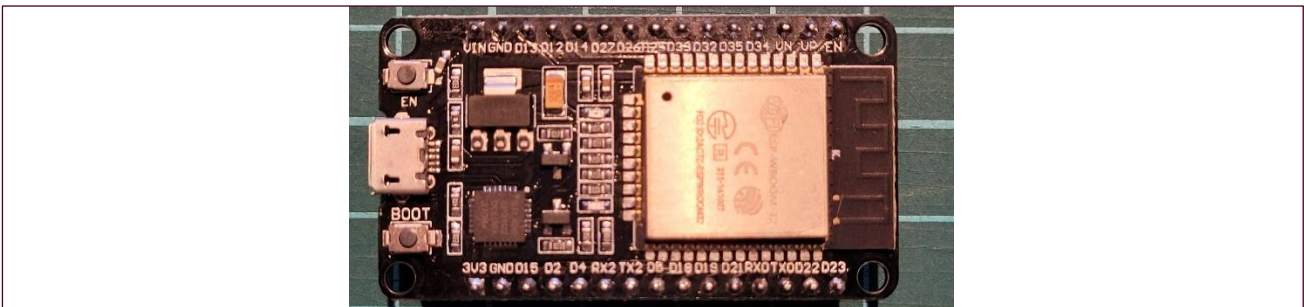


Abbildung 8: ESP32 NodeMCU mit 30 Pins

⁹ (Trojan, 2017) MQTT auf ESP8266 mit technischen Angaben auf Seite 126/127

¹⁰ (www.joy-it.net, 2023) technische Daten zum ESP32 NodeMCU

¹¹ (Brandes, 2023) Kapitel 2 Die Stromversorgung des ESP 32

2.3.1.5 ESP32S

Der ESP32S ist der leistungsstärkere Nachfolger des ESP8266 12E/F. Auch bei diesem Chip ist vorgesehen, dass dieser direkt auf die Platine gelötet wird. Auf den Entwicklungskits ist eine Form dieses Chips verbaut. Wenn dieser Chip in ein Projekt integriert werden soll, muss wie auch schon beim ESP8266 12E/F, die für den Betrieb und die Programmierung benötigte Elektronik auf dem Print beschalten werden. Diese Chipfamilie erlaubt es Projekte möglichst schlank zu realisieren.

Spezifikationen¹²

Betriebsspannung:	3.0VDC bis 3.6VDC
I/O Spannung:	3.3VDC
Analog / Digital-Converter (ADC):	20
Digitale I/O:	36
Interrupts (Wake Up):	programmierbar auf alle GPIO
Kommunikation:	3 x UART, 2 x I2C, 2 x SPI und noch weitere
Stromverbrauch (deepsleep):	20µA
Batteriebetrieb möglich	Ja

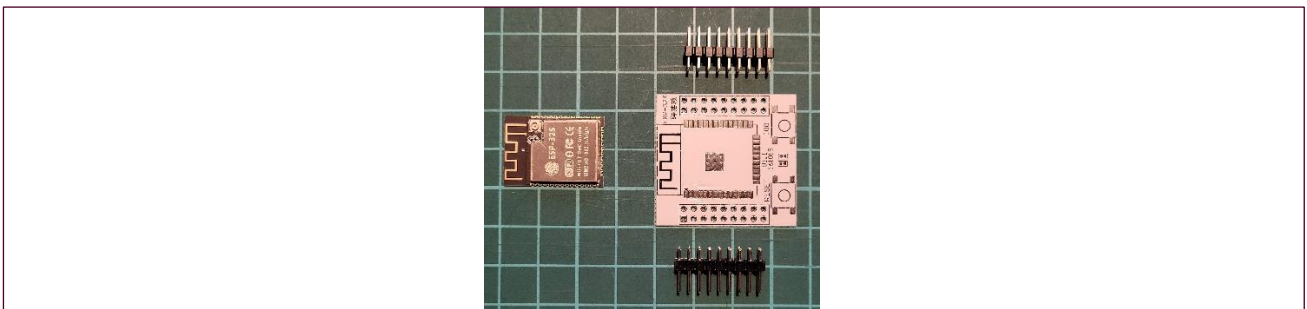


Abbildung 9: ESP32S mit Breakout-Board

¹² (Espressif Systems, 2023) ESP32S2 Datenblatt

2.3.2 Sensoren

Alle drei Sensorvarianten bauen auf dem optischen Rauchmelder, welcher von der BKW als Geschenk überreicht wurde, auf. Dieser Typ Rauchmelder von der Firma Hekatron basiert auf dem Streulichtprinzip und ist somit ein optischer Rauchmelder. Diese Art von Rauchmelder ist zurzeit die meistverbreitete Art, welche mit diesem Prinzip arbeiten. Dabei wird von einer Fotodiode ein Lichtstrahl ausgesendet (meist unsichtbar), welcher nicht direkt auf einen Empfänger trifft. Die Fotodiode und der Empfänger befinden sich in einem Gehäuse, welches mit der Umgebungsluft durch/umströmt wird, so dass bei einer bereits geringen Rauchentwicklung das Licht an den Rauchpartikeln in die Richtung des Empfängers reflektiert werden und so ein Alarm ausgelöst werden kann. Die in Abbildung 10¹³ grafisch Dargestellte Funktion zeigt den Normalfall ohne Rauch und die Abbildung 11 zeigt den Vorgang mit Rauch.

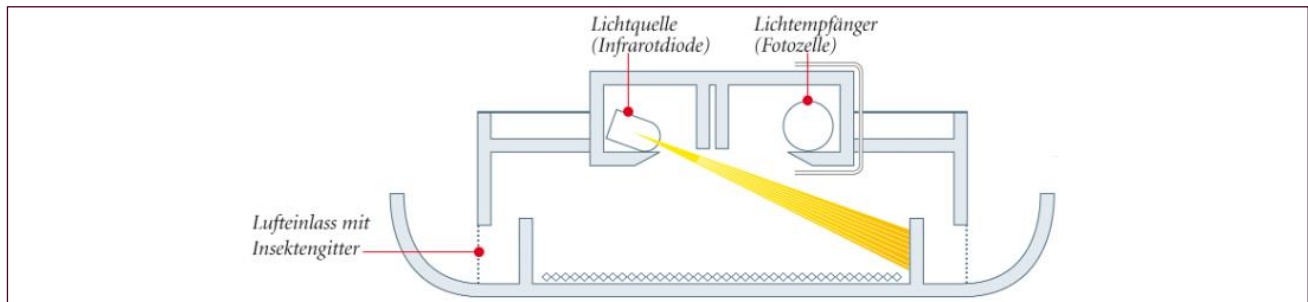


Abbildung 10: Rauchmelder im Normalzustand

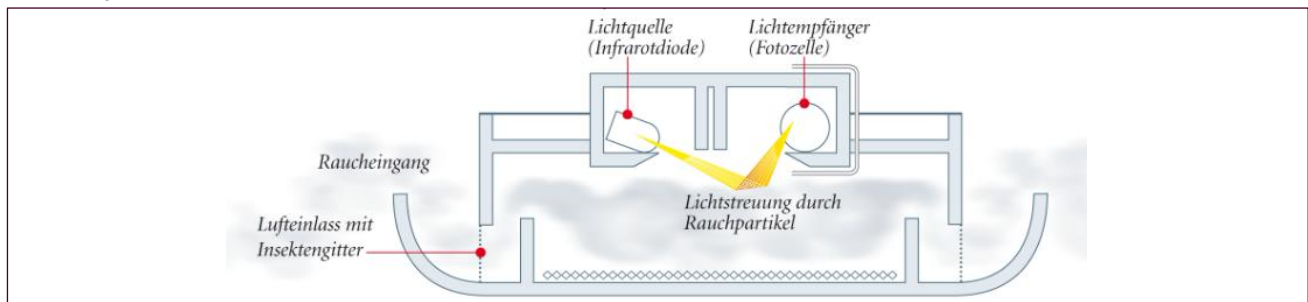


Abbildung 11: Rauchmelder bei Brandrauch in der Luft

Diese Sensoren leiden durch die Öffnungen unter Verschmutzung oder Insektenbefall, was zu einem Fehlalarm führen kann. Daher sind in den Rauchmeldern Insektengitter angebracht und sollten alle 10 Jahre ausgetauscht werden. Die heutigen Rauchmelder werden von den meisten Herstellern für die Wohnräume empfohlen. Testergebnisse¹⁴ haben gezeigt, dass sich einige Rauchmelder auch in Küchen, Kellern und in Raucherwohnungen eignen, wo es zu normaler Rauchentwicklung kommen kann. Meist sind Grenz/Schwellwerte mit einer entsprechenden Logik in den Brandmeldern implementiert, welche Fehlalarme reduzieren sollen. Da dieser Rauchmelder und seine Nachfolger einerseits von der Gebäudeversicherung des Kantons Bern empfohlen werden und andererseits die Vereinigung kantonaler Feuerversicherungen (VKF) ein Zertifikat für diese Rauchmelder ausgestellt haben wird dieser Sensor/Rauchmelder als Basis für die folgenden drei Varianten verwendet. In allen drei Varianten wird über einen Pin auf der Platine des Rauchmelders (siehe auch Kapitel 3.2.3.4) der Alarmimpuls für die Alarmauslösung abgegriffen.

¹³ (Minol Messtechnik GmbH, 2023) Fragen und Antworten zu Rauchwarnmeldern

¹⁴ (Rauchmeldertest.net, 2022) Reaktion auf Rauch (Küche/Rauchen)

2.3.2.1 Optischer Rauchmelder einfach

Der vorliegende Rauchmelder wird ausschliesslich nur mit dem Mikrokontroller ausgerüstet, welcher einen Alarm des Rauchmelders via W-LAN mittels MQTT-Publisher Nachricht an den MQTT-Broker weiterleitet. Dabei wird nur der Alarmimpuls des Sensors ausgewertet und in eine entsprechende Nachricht verpackt und weitergeleitet. Nach der Alarmabsetzung wird der Mikrokontroller wieder in den Energiesparmodus versetzt. Der Mikrokontroller wird mit einer eigenen Stromversorgung ausgerüstet, um die Lebensdauer der Batterie des Rauchmelders nicht zu beeinträchtigen sofern die Batterie überhaupt einen Mikrokontroller zu speisen vermag. Die Abbildung 12 zeigt den visualisierten Aufbau des Sensors.

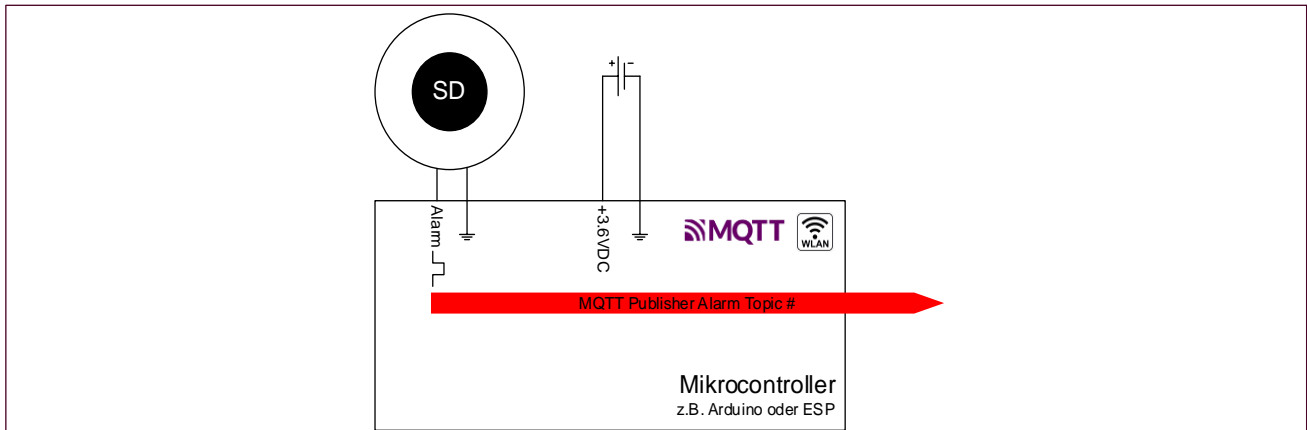


Abbildung 12: Rauchmelder Variante 1

Der Mikrokontroller wird durch den Alarmimpuls über einen Interrupt-fähigen GPIO eines Arduinos oder eines ESP-Mikrokontrollers aus seinem Energiesparmodus (Deepsleep) aktiviert und kann dann im Programmablauf eine entsprechende MQTT-Nachricht absetzen. Einige Mikrokontroller verfügen auch über entsprechende Reset-Pins für diesen Zweck. Aus Stromspargründen werden keine weiteren Informationen zur Zentrale übertragen.

Vorteile	Nachteile
<ul style="list-style-type: none"> - sehr Energieeffizient da nur im Alarmfall der Mikrokontroller aktiviert wird; - Einfacher Hardwareaufbau, da nur die eine rudimentäre Beschaltung des Mikrokontrollers notwendig ist; - sehr platzsparend, es kann auf einem kleinen zusätzlichen Print realisiert werden. 	<ul style="list-style-type: none"> - Zustand der zusätzlichen Batterie lässt sich nicht genau überprüfen; - Konnektivität des WiFi Moduls kann nicht überprüft werden.

Tabelle 7: Vor-/Nachteile der Sensorvariante 1

2.3.2.2 Optischer Rauchmelder einfach mit Betriebsdatenauswertung

Die Sensorvariante 2 baut auf der Sensorvariante 1 auf und wird um einige Funktionen erweitert. Um die Nachteile der ersten Version zu neutralisieren, wird einerseits eine periodische (alle 2-4 Stunden) Spannungsüberprüfung der Batterie und der Konnektivität durchgeführt. Die gemessene Betriebsspannung, welche an einem Analogeingang erfasst wird, wird dann auf einem eigenen MQTT-Publisher Topic dem MQTT-Broker über das W-LAN übertragen. Auch der WiFi-Empfangspegel kann in diesem periodischem Messzyklus erfasst und als ein weiteres MQTT-Publisher Topic an den MQTT-Broker gesendet werden. Hierbei muss jedoch der Sensor zwischen einem Alarm und einer Betriebsdatenübermittlung unterscheiden können. Diesem Umstand ist Hard- oder Softwaretechnisch Beachtung zu schenken. Die Abbildung 13 zeigt den visualisierten Aufbau des ergänzten Sensors.

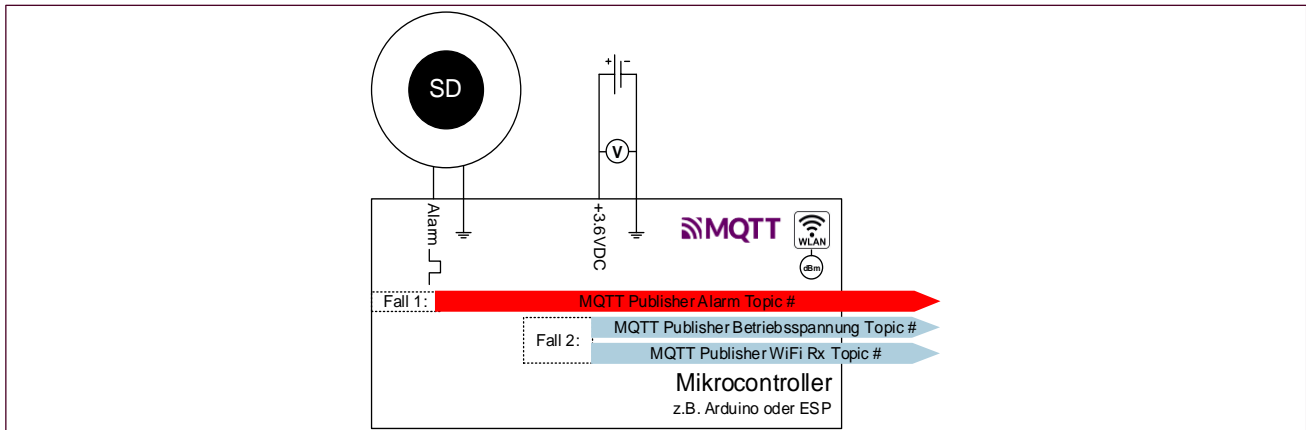


Abbildung 13: Rauchmelder Variante 2

Die zusätzlichen Schaltungen, welche für die Betriebsdatenerhebung und für die Fallunterscheidung benötigt werden müssen auf der gleichen Printplatte wie der Mikrokontroller untergebracht werden. Allenfalls kann durch die Generierung geeigneter Brücken oder Codeänderungen der Sensor zur Variante 1 einfach zurückgebaut werden falls ein Energieeffizienterer Sensor gewünscht wird.

Vorteile	Nachteile
<ul style="list-style-type: none"> - Betriebsdaten des Sensors können jederzeit überprüft werden; - In der Zentrale können kritische Betriebszustände ausgewertet werden und allenfalls den Benutzer rechtzeitig darüber informieren; - Kann durch die Periodizität der Betriebsdatenerfassung immer noch Energieeffizient betrieben werden. - Modularer Aufbau 	<ul style="list-style-type: none"> - Die Batterie wird mit zusätzlichen Aufgaben etwas stärker beansprucht und muss daher sicher öfters ausgetauscht / aufgeladen werden; - Für die Messdatenerhebung und Alarmunterscheidung muss zusätzliche Hardware eingebaut werden. Somit werden die Platinen Dimensionen etwas grösser.

Tabelle 8: Vor-/Nachteile der Sensorvariante 2

2.3.2.3 Optischer Rauchmelder ergänzt mit Betriebsdatenauswertung

Die Sensorvariante 3 ist auch wieder eine Weiterentwicklung der vorangehenden Varianten. Dabei werden zusätzliche Sensoren zum optischen Rauchmelder hinzugefügt und in einer Schaltlogik im Mikrokontroller miteinander verknüpft. Dabei können verschiedene Kombinationen von Sensoren und Schwellwerten der verschiedenen Messungen einander gegenübergestellt werden und somit Alarmbedingungen definiert werden. Die ergänzenden Sensoren könnten Gase, Feuer und Temperatur zusätzlich erfassen. Diese Art von Sensor ist vor allem für Räumlichkeiten gedacht, wo starke Rauchentwicklung oder Wasserdampf entstehen und an der Tagesordnung sind (Werkstatt, Badezimmer, Küche). Die Abbildung 14 zeigt den visualisierten Aufbau des mit weiteren Sensoren ergänzten Brandmelders.

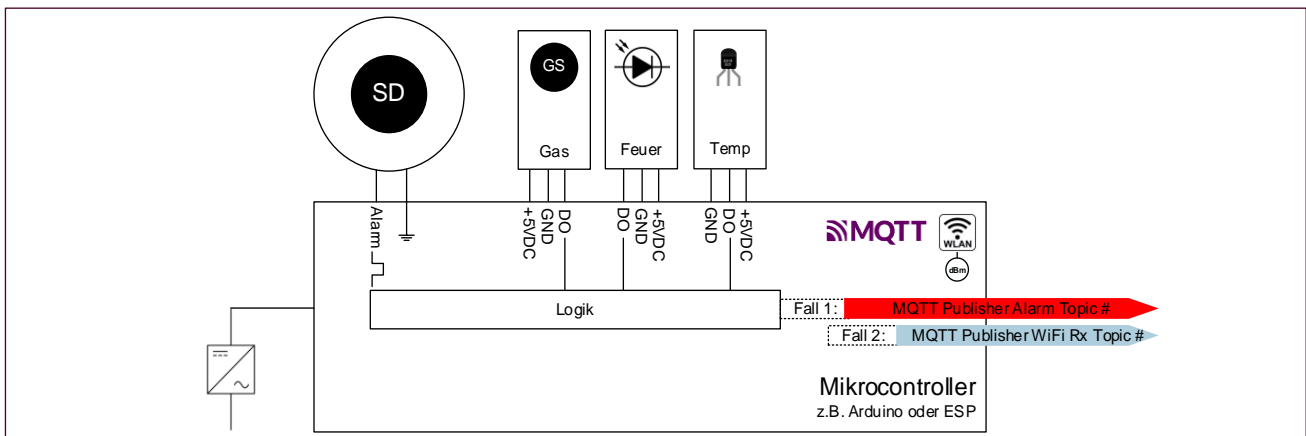


Abbildung 14: Rauchmelder Variante 3

Für die Ausarbeitung der Variante 3 habe ich mich entschieden mit den folgenden 3 Sensoren zu arbeiten:

Gassensor Joy-It SEN-MQ2¹⁵

Der Gassensor kann sowohl digital als auch analog ausgelesen werden. In der digitalen Variante kann über ein Potentiometer zusätzlich noch die Empfindlichkeit des Sensors beeinflusst werden. Zudem können unterschiedliche Gase erkannt werden, um allenfalls in einem Kellerraum oder einer Garage ein Gasleck festzustellen. Der Sensor arbeitet mit einem Heizelement und einem chemischen Sensor, welcher die Veränderung des Katalysators an der zu Messenden Substanz feststellen kann. Dabei werden am Sensor die Anzahl der Teilchen (parts per million oder ppm) gemessen und in einen elektrischen Wert umgewandelt. Somit wird die Gaskonzentration gemessen und nicht die Art des Gases. Dieser Wert kann dann von einem Mikrokontroller ausgelesen werden.

Spezifikationen

Betriebsspannung:	5VDC
Betriebsstrom:	250-300mA
Messbereich:	100-10000ppm
Analogwerte:	0-5VDC
Digitalwerte:	0 und 5VDC

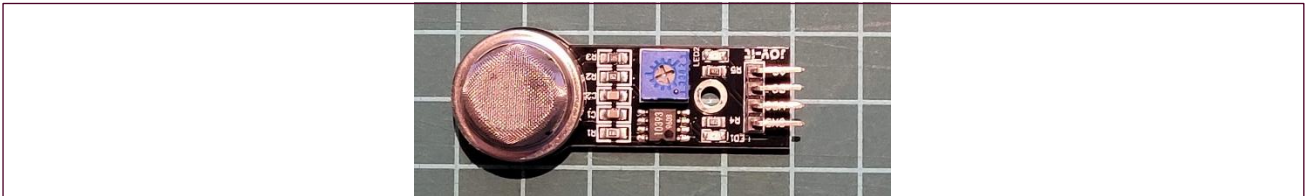


Abbildung 15: Joy-It SEN-MQ2 Gassensor

Feuersensor Purecrea IR Infrared Flame Sensor¹⁶

Der Flammensensor misst die Wärmestrahlung im Infrarotbereich. Der Sensor kann sowohl digital wie auch analog ausgewertet werden. Mit einem Potentiometer ist der Schwellwert oder Triggerwert zu beeinflussen. So kann die Art des Feuers ein wenig eingeschränkt werden und Fehlalarme reduziert werden, welche durch gebrochenes Licht oder andere im Infrarotbereich arbeitende technische Geräte verursacht werden.

Spezifikationen

Betriebsspannung:	3.3-5VDC
Messbereich:	760-1100nm (Wellenlänge)
Analogwerte:	0-5VDC
Digitalwerte:	0 und 5VDC

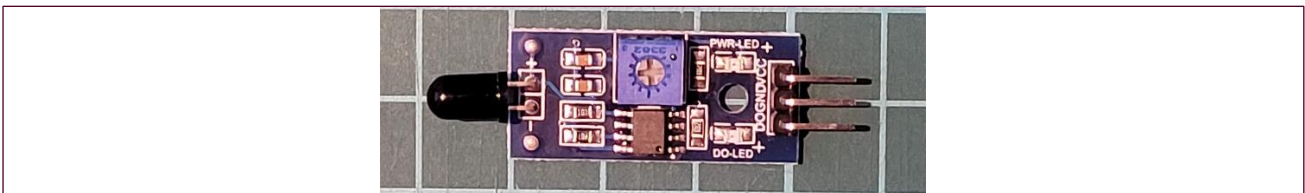


Abbildung 16: Purecrea IR Infrared Flame Sensor Module

¹⁵ (www.joy-it.net, 2023) technische Daten des Gassensors SEN-MQ2

¹⁶ (Switch Electronics, 2023) IR Infrared Flame Detector Sensor Module

Temperatursensor Dallas DS18B20¹⁷

Der Temperatursensor wird über einen sogenannten "One-Wire-Bus" angesteuert oder abgefragt und verfügt über eine eigene 64-Bit Adresse. Mit diesem Sensor kann ein rascher Temperaturanstieg in Deckenhöhe erkannt werden. Da nur die Temperaturänderung relevant ist, spielt die Genauigkeit nicht eine allzu grosse Rolle.

Spezifikationen¹⁸

Betriebsspannung:	3-5.5VDC
Messbereich genau (+/- 0.5°C):	-10 bis 85°C
Messbereich ungenau:	-55 bis 125°C

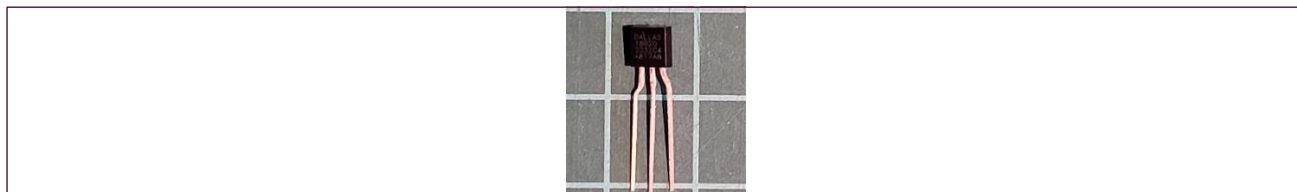


Abbildung 17: Dallas DS18B20 Temperatur Sensor

Die zusätzlichen Sensoren weisen einen deutlich höheren Stromverbrauch auf als die anderen beiden Varianten. Daher müsste diese Sensorvariante mit einem entsprechenden Netzteil oder DC/DC Wandler ausgestattet werden. Die Abfrageperiodizität könnte unter Umständen wie in den anderen Varianten beschrieben ist beibehalten werden (2-4h), wenn die zusätzlichen Sensoren erst bei einer Auslösung des optischen Sensors ausgelesen werden um dann in dieser Abhängigkeit ein Alarm herausgegeben wird.

Vorteile	Nachteile
<ul style="list-style-type: none"> - Mehr Sensoren weisen eine höhere Alarmwahrscheinlichkeit in Räumen mit einer alltäglichen Rauch- / Gasentwicklung auf; 	<ul style="list-style-type: none"> - Deutlich höhere Strombelastung der Spannungsquelle; - Komplexerer Schaltungs- und Codeaufbau; - Abmessungen des Sensors werden durch die zum Teil grossen Module grösser - Zuverlässigkeit wird nicht viel grösser sein als ohne die zusätzlichen Sensoren

Tabelle 9: Vor-/Nachteile der Sensorvariante 3

¹⁷ (Elektronik Kompendium, 2023) DS18B20 – Digitaler Temperatursensor

¹⁸ (DigiKey.ch, 2023) Datenblatt eines DS18B20 Temperatursensors

2.4 Variantenwahl

2.4.1 Wahl der Mikrokontroller

Die in Kapitel 2.3.1 dargestellten Mikrokontrollervarianten wird anhand ihrer in den Varianten genannten Eigenschaften einen Verwendungszweck zugewiesen. Es handelt sich hierbei nicht um eine Auswahl nach Punkten, da die Mikrokontroller schwer miteinander verglichen werden können. In den Begründungen wird angegeben, weshalb der entsprechende Mikrokontroller für den Verwendungszweck gewählt wurde.

Mikrokontroller	Verwendungszweck	Begründung
Arduino Nano 33 IoT	Wird nicht verwendet	- Unzuverlässig im Dauerbetrieb auf Grund von gemachten Erfahrungen
ESP8266 12F	Ergänzung Rauchsensor	- Kleine Bauform - Sehr geringer Stromverbrauch - ADC für Spannungsmessung vorhanden - Wakeup mit Reset und Timer möglich
ESP8266 DevkitC / NodeMCU V2	Codeentwicklung für die Rauchsensoren	- Basiert auf dem gleichen Chip wie der ESP8266 12F - Lässt sich einfacher programmieren - Gleicher Funktionsumfang wie ESP8266 12F - Eignet sich nicht zur Implementierung
ESP32S	Wird nicht verwendet	- Kann nur mit 3-3.6VDC gespiesen werden - Benötigt eine Programmier- und Boot-Beschaltung
ESP32 NodeMCU	Alarm-Mikrokontroller	- Verfügt über mehrere UART-Schnittstellen für die Entwicklung der Ansteuerung des SIM7600 Moduls mittels AT-Befehlen - In der Zentrale wird mit 5V Betriebsspannung gearbeitet

Tabelle 10: Wahl der Mikrokontroller

2.4.2 Wahl der Sensorvariante

Die in Kapitel 2.3.1 dargestellten Sensorvarianten werden an Messbaren Kriterien miteinander verglichen. Alle Kriterien werden mit Werten von 1-10 in einer Matrix bewertet. Am Schluss werden alle Punkte einer Sensorvariante zusammengezählt. Die Variante mit den meisten Punkten wird im Kapitel 3 Realisierung realisiert. Der Zahlenwert 1 ist somit für eine Realisierung ungeeignet und der Zahlenwert 10 bestens geeignet. Folgende Kriterien werden bewertet:

- Funktion:** Der Sensor weist Funktionen zur Überprüfung des Betriebszustandes auf und kann für weitere Entwicklungen umprogrammiert werden.
- Aufbau:** Der Mikrokontroller und die dazugehörige Elektronik soll in einem kleinen Zusatzgehäuse untergebracht werden können
- Batteriebetrieb:** Der Sensor soll mit einer Batterie / Akku möglichst platzsparend über einen längeren Zeitraum betrieben werden können ohne Netzanschluss für ein Ladegerät.
- Energieeffizienz:** Die Sensoren und ihre Vernetzung soll möglichst Energieeffizienz realisierbar sein.
- Modularität:** Der Funktionsumfang kann durch kleine mechanische Arbeiten eingeschränkt/vergrößert werden.

	Funktion	Aufbau	Batteriebetrieb	Energieeffizienz	Modularität	Total Punkte
Sensorvariante 1	2	10	10	10	0	32
Sensorvariante 2	8	6	8	8	7	37
Sensorvariante 3	10	1	0	3	10	24

Tabelle 11: Matrix für die Variantenbewertung

Aus der Bewertungsmatrix ist ersichtlich, dass die Sensorvariante 3 vor allem aus Konstruktionstechnischen und aus Energieeffizienzgründen deutlich weniger Punkte aufweist als die beiden anderen Varianten. Die Sensorvariante 1 und 2 liegen deutlich näher beieinander, wobei die Sensorvariante 2 ein guter Kompromiss von Variante 1 und 3 ist. Die Sensorvariante 2 verliert überall nur kleine Punktteile im Gegensatz zu den anderen beiden Sensorvarianten, welche beide in verschiedenen Kategorien keine Punkte erzielen. Die Vor- und Nachteile der Sensorvariante 2 bestätigen diese Bewertung ebenfalls.

Somit wird die Sensorvariante 2 im Endausbau mit einem ESP8266 12F und für die Entwicklung auf einem ESP8266 NodeMCU V2 in dem folgenden Kapitel realisiert.

Ein Bestandteil der Zentrale wird der Alarm-Mikrokontroller sein, welcher auf einem ESP32 NodeMCU realisiert wird. Die Zentrale wird zudem noch mit einem Raspberry Pi 3B als MQTT-Broker und Node-RED Plattform, das Sirenenmodul für den lokalen Akustikalarm und dem SIM7600X 3/4G LTE Mobilfunkmodul ergänzt.

3 Realisierung

3.1 Systemaufbau

Der Systemaufbau stammt aus dem Kapitel 2.2 SOLL-Zustand und wird in diesem Kapitel noch weiter ergänzt mit den Entscheiden aus Kapitel 2.4 Variantenwahl. Entgegen den Anforderungen wird der Endausbau mit 4 Sensoren realisiert, da ein Sensor auf dem Entwicklerboard (Breadboard) für Demozwecke mit einem anderen Timing aufgebaut und integriert wird.

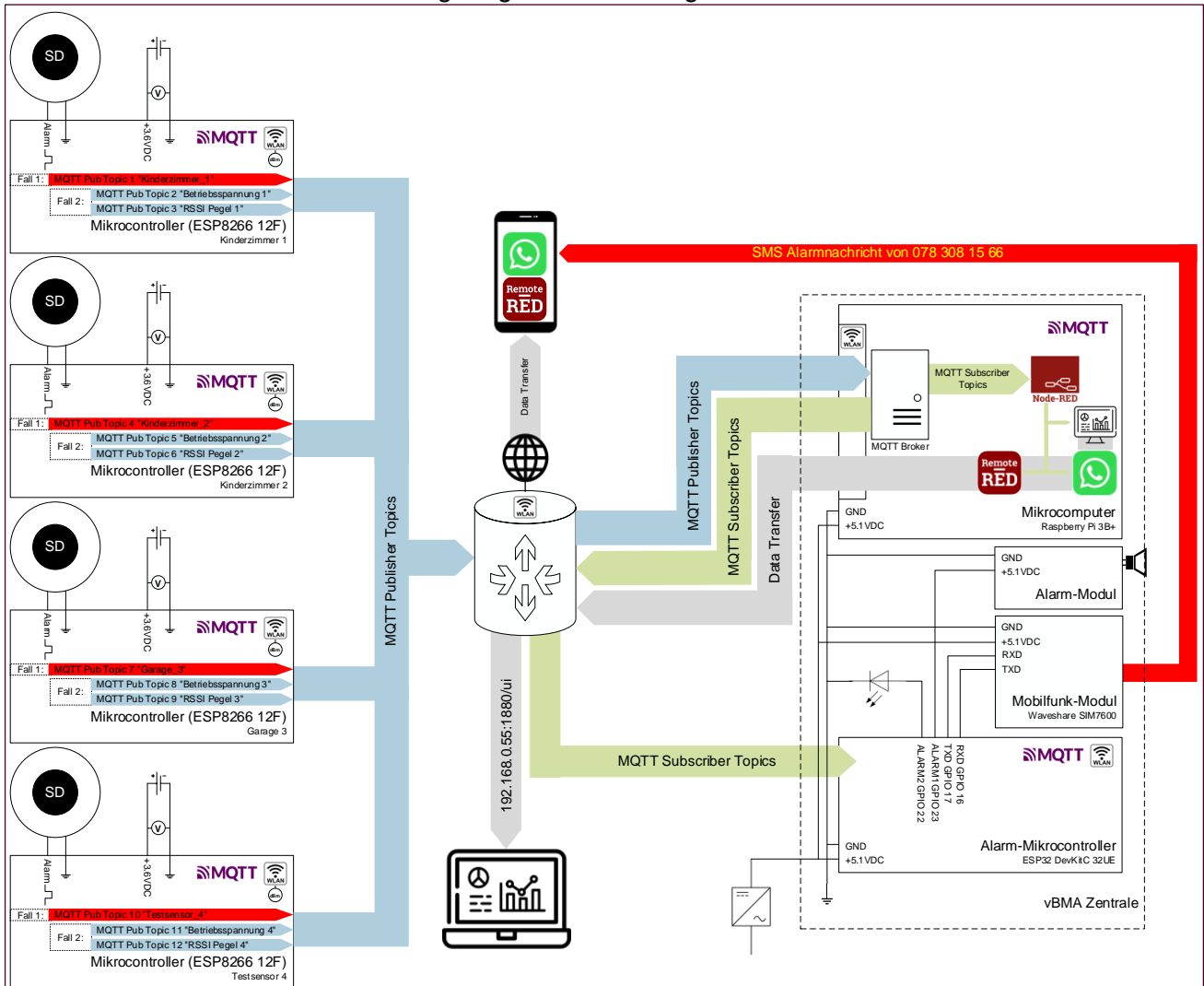


Abbildung 18: Systemaufbau vBMA

3.1.1 Beschreibung der Hauptfunktion

Wenn in einem Raum ein Feuer ausbricht oder eine starke Rauchentwicklung stattfindet soll der zertifizierte Rauchsensor einen lokalen Alarm erzeugen. Dieses Alarmsignal wird abgegriffen und ein Mikrocontroller wird aus seinem Energiesparmodus (Deepsleep) aufgeweckt. Dieser verbindet sich erst mit den W-LAN und dann mit dem MQTT-Broker und setzt dann eine spezifische Nachricht auf einem eigenen MQTT-Topic ab (z.B. Kinderzimmer_1), in welchem Raum der Rauchmelder ausgelöst wurde (z.B. "Rauchalarm aus Kinderzimmer 1!"). Danach wird der Mikrocontroller wieder in den Energiesparmodus versetzt. Wenn der Brandalarm nicht wieder ausgelöst wird, kann dies als Fehlalarm interpretiert werden.

In beiden Fällen wird die Nachricht an den MQTT-Broker weitergeleitet. Der MQTT-Broker befindet sich auf einem Raspberry Pi 3B und läuft als Hintergrundprogramm auf der Applikation von Eclipse

Mosquitto. Das übermittelte Alarm-Topic wird nun an die Abonnenten (Subscriber wie Node-RED oder Alarm-Mikrokontroller) versendet.

Ein zentraler Teil von der Alarmweiterleitung ist der Alarm-Mikrokontroller, welcher sich nicht in einem Energiesparmodus befindet, da dieser jederzeit Alarmpfeile empfangen können soll. Der Alarm-Mikrokontroller arbeitet mit einer Finite-State-Machine (FSM). Dies bedeutet, dass wenn kein Alarm anliegt sich dieser im State "kein Alarm State" befindet. Der Wechsel in einen "Alarm State" wird Anhand der Methode `nachricht.endsWith(1!, 2!, 3! oder 4!)` gemacht, wo die Endung der abonnierten Nachricht überprüft wird. Wenn eine entsprechende Endung erkannt wird, so wechselt die Finite-State-Machine in einen Alarm State.

In diesem Alarm State wird erst die lokale Alarmierung (Blitzer LED und akustischer Alarm) ausgelöst. Nach der lokalen Alarmierung werden via AT-Kommandos das SIM7600X Mobilfunkmodul angesteuert, indem erst die Zielnummer gewählt wird, dann die erhaltene Nachricht als SMS-Text eingegeben wird und zum Schluss die SMS-Nachricht versendet wird.

Solange der Reset Taster an der Alarmzentrale nicht gedrückt wird, ist der lokale Alarm aktiv. Der Rauchmelder, welcher ausgelöst hat, muss ebenfalls zurückgesetzt werden, damit kein periodischer Alarm ausgelöst wird und eine Vorortkontrolle durchgeführt wird (auch nach einem Fehlalarm).

Ein ausgelöster Alarm wird ebenfalls auf dem Node-RED Dashboard angezeigt. Die Alarm-Anzeige auf dem Node-RED Dashboard wird erst nach einer erfolgreichen Übermittlung der Betriebsdaten wieder auf "kein Alarm" gewechselt.

3.1.2 Beschreibung der Nebenfunktionen

Wenn kein Alarm auf den Rauchmeldern anliegt, so sollen periodisch (alle 3 Stunden) die jeweiligen Betriebszustände der Sensoren über jeweils ein MQTT-Topic übermittelt werden. Die relevanten Betriebszustände sind einerseits die Betriebsspannung des Akkus und andererseits der WiFi-Empfangspegel (RSSI / Received Signal Strength Indicator). Zu diesem Zweck muss der Mikrokontroller nach einer Zeit von 3 Stunden aufgeweckt werden und muss erkennen können, dass kein Alarm anliegt. Die Betriebsspannung und RSSI-Pegel werden dann von jedem Sensor über ein eigenes MQTT-Topic (z.B. "Betriebsspannung 1" und "RSSI Pegel 1") alle 3 Stunden an den MQTT-Broker übertragen.

Der MQTT-Broker reicht diese abonnierten Topics wiederrum an seine Abonnenten (Subscriber) weiter. Die Betriebsdaten der Sensoren werden ausschliesslich in Node-RED ausgewertet. Kritische Betriebszustände werden über einen Whatsapp-Node (Chatbot im Hintergrund) an den Benutzer übertragen. Dies aus dem Grund, wenn der Benutzer einen Klingelton der SMS-Rufnummer hinterlegt hat, nicht falsch reagiert wird, da der Chatbot nicht die gleiche Nummer aufweist wie das Mobilfunkmodul. Kritische Betriebszustände sind Unterspannung und zu niedrige RSSI Werte.

Sämtliche übertragenen Betriebsdaten werden übersichtlich auf einem Dashboard dargestellt, welches auch von ausserhalb des W-LAN abrufbar ist (z.B. via VPN oder RemoteRED App). Innerhalb des Netzwerkes kann auch direkt auf das Dashboard zugegriffen werden (z.B. via RDP oder direkt über den Port 1880).

Das Dashboard zeigt die aktuelle Betriebsspannung, einen Betriebsspannungsverlauf und den Verlauf der RSSI-Pegel sowie ob ein Alarm anliegt oder nicht und dies von jedem Sensor. Dabei werden die Verläufe über einen Zeitraum von 4 Wochen dargestellt.

Neben der Alarmierung und der Betriebsdatenauswertung können via MQTT-Nachricht noch weitere Topics und Dashboards auf weiteren Node-RED Dashboards einfach dargestellt werden. Aus der Semesterarbeit stehen verschiedene Raumklima- und Wetterdaten zur Verfügung, welche noch auf MQTT-Topics aufgeschlüsselt und übertragen werden müssten und so in einem neuen Node-RED Dashboard übersichtlich dargestellt werden können, ohne den sicheren Betrieb der vBMA zu beeinträchtigen.

3.1.3 Hauptkomponenten

Die Hauptkomponenten der vBMA bestehen somit aus mehreren Sensoren und einer Zentrale. Die Abbildungen 19 und 20 zeigen die Aufgebauten Funktionsmuster eines Sensors und der Zentrale. Die Zentrale besteht aus dem Raspberry Pi 3B, Alarm-Mikrokontroller, SIM7600X Mobilfunkmodul und einer Stromversorgungsplatine für Entwicklerboards. Das akustische Sirenenmodul wird im Funktionsmodul mit einer grünen LED simuliert (Übersichtlichkeit).

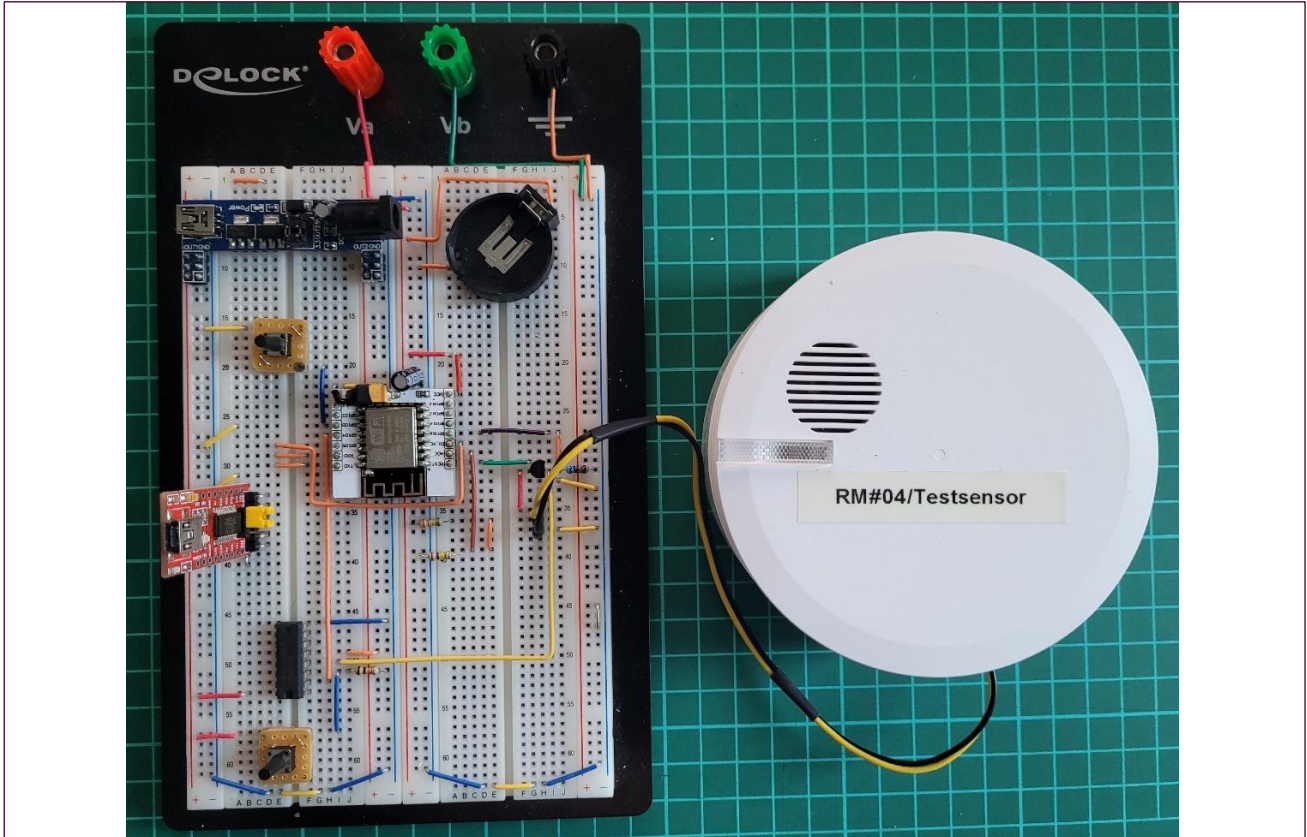


Abbildung 19: Funktionsmodell eines Sensors

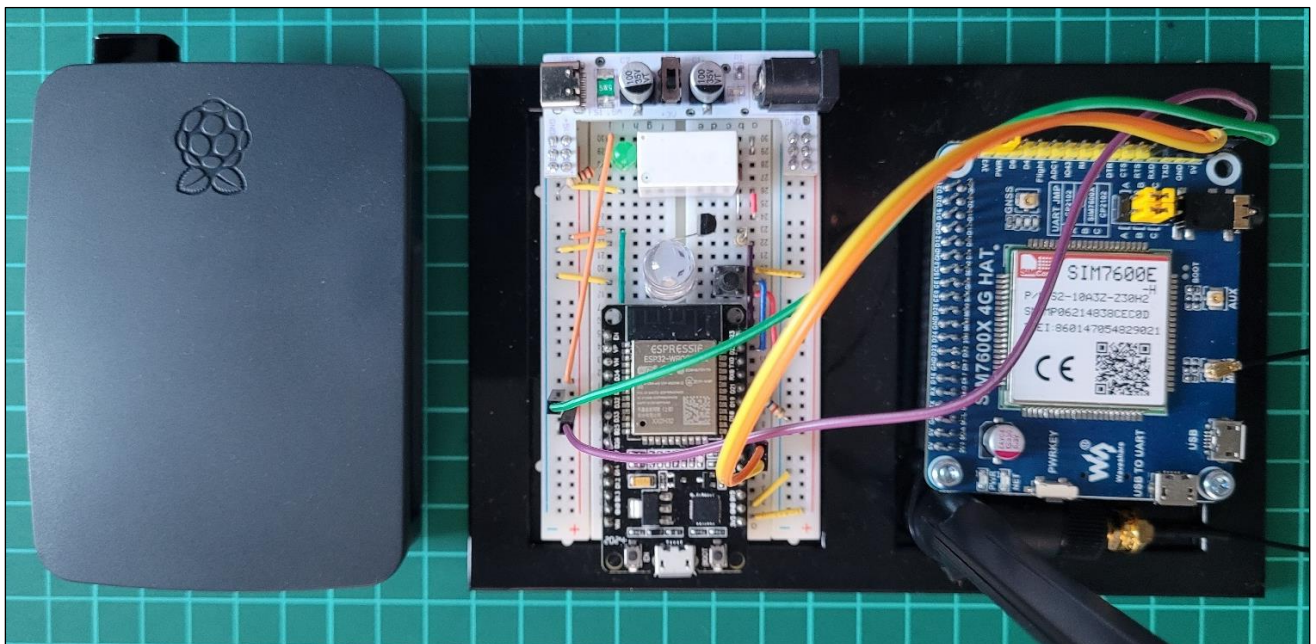


Abbildung 20: Funktionsmodell der Zentrale (reduziert)

Die Entwicklung der beiden Hauptkomponenten wird in den nächsten Kapitel 3.2 und 3.3 ausführlich beschrieben.

3.2 Sensoren

3.2.1 Funktionsbeschreibung Sensor

Der Sensor muss im Alarmfall über den Reset Pin aus dem Energiesparmodus geholt werden, damit der ESP8266 12F über das integrierte WiFi-Modul seine MQTT Alarmnachricht absetzen kann. Dies wird über den RST-Pin realisiert. Die Nachricht, welche abgesetzt wird, muss die Information enthalten um welchen Raum es sich genau handelt (z.B. "Kinderzimmer_1"). Um dies zu tun, muss sich der Sensor erst mit dem MQTT-Broker verbinden welcher sich im gleichen Netzwerk, jedoch in der vBMA-Zentrale befindet. Nachdem der Alarm abgesetzt wurde, wird der Mikrokontroller wieder in den Energiesparmodus (Deepsleep) versetzt. Wenn der Rauchmelder immer noch Alarmsignale sendet, so wird der ESP8266 12F immer wieder einen Alarm absetzen. Damit der ESP8266 12F im Alarmfall nur den Alarm absetzt und nicht noch Betriebsdaten auswertet muss dieser über ein RS-Flipflop erkennen ob ein Alarm anliegt oder nicht. Mit dem Alarmsignal wird gleichzeitig der Alarm in einem RS-Flipflop gespeichert, da der Alarm nur ein Impuls ist und dem ESP8266 12F nur während der Impulsdauer zur Auswertung zur Verfügung. Die Auswertung des Ausgangs des RS-Flipflops erfolgt über GPIO04.

Wenn kein Alarm anliegt, wird der ESP8266 12F periodisch über die interne Uhr (Real Time Clock) alle 3 Stunden über den GPIO16 und den RST-Pin aus dem Energiesparmodus geholt werden, um eine Betriebsdatenerfassung durchzuführen (Spannung, RSSI-Pegel). Die Betriebsspannung wird am ADC-Pin über einen hochohmigen Spannungsteiler digitalisiert und im Code in einen Spannungswert zurückgerechnet. Der RSSI-Pegel kann vom WiFi-Modul ausgelesen werden. Diese erfassten Daten werden über eigene MQTT-Topics abgesetzt, damit die vBMA-Zentrale diese auswerten kann. Nach dieser Aktion wird der Sensor wieder in den Energiesparmodus zurückversetzt.

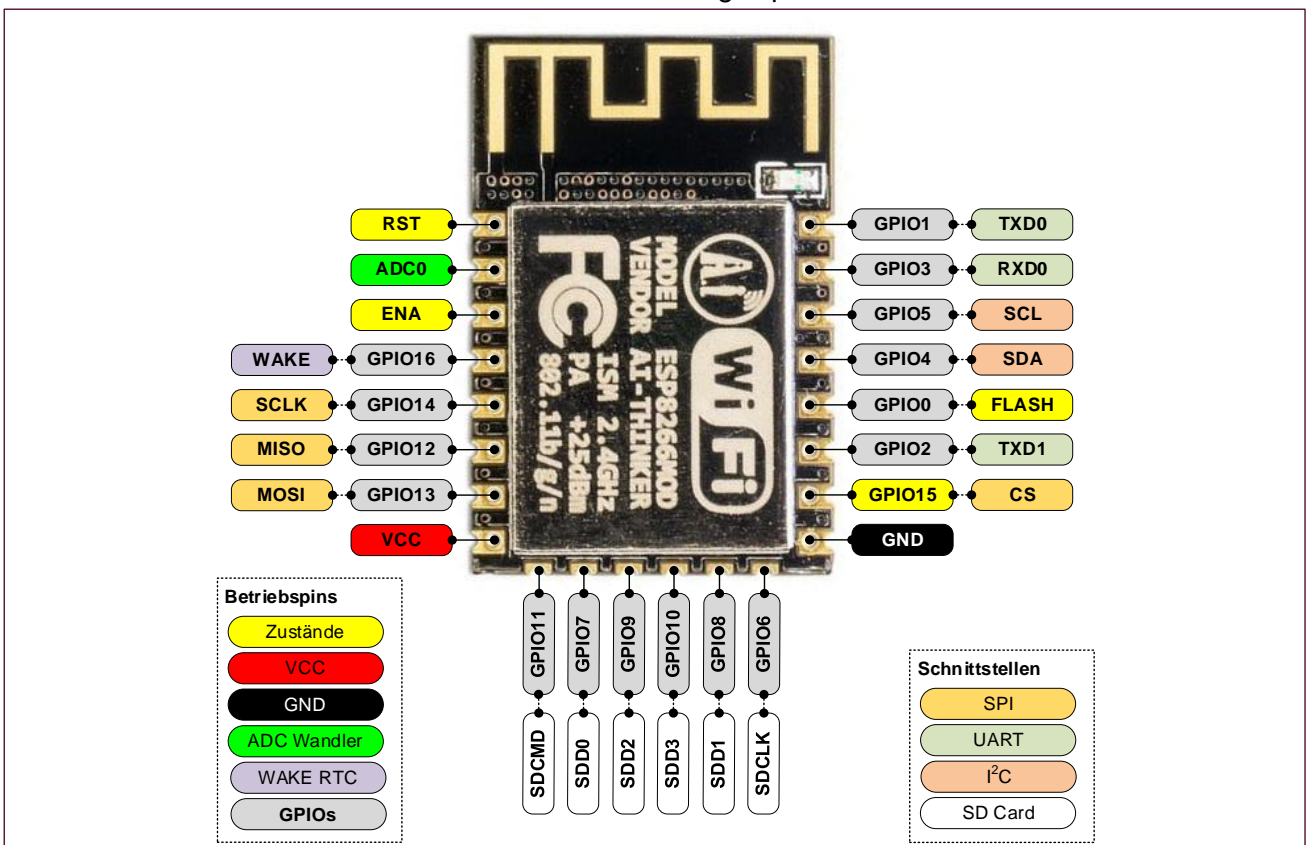


Abbildung 21: ESP8266 12F Pin-Belegungsplan¹⁹

¹⁹ (Random Nerd Tutorials, 2023) ESP8266 12-E Chip Pinout

3.2.2 Technischer Aufbau

Der technische Aufbau erfolgte etappenweise. Die Entwicklungsschritte (ohne Codeentwicklung) waren wie folgt gegliedert:

- Entwicklungsboard mit FT232TL-Programmer mit Stromversorgung vorbereitet;
- ESP8266 12F auf Hilfsplatine inklusive Stromversorgung und Programmierschaltung aufgebaut;
- Strommessungen im Energiesparmodus durchgeführt;
- Alarmauslösung mit MOSFET N-Kanal Transistor implementiert;
- Spannungsüberwachung mittels Spannungsteiler implementiert;
- Alarmspeicherung mit RS-Flipflop implementiert;
- Zeichnen eines Schaltplanes für die Dokumentation;
- Erstellen eines Gerber-Files für die Platinerstellung;
- Aufbau von 3 Sensoren

Der stufenweise Aufbau war immer mit der entsprechenden Codeentwicklung gepaart, denn nur so war es mir möglich die einzelnen Schritte nachzuvollziehen und zu testen. Den Schaltplan habe ich fortlaufend weiterergänzt, erst nur handschriftlich später dann in Target 3001. Aus diesem Programm konnten auch die Gerber-Files für die Platinerstellung exportiert werden und 1:1 für die Webseite von JLCPCB.com weiterverwendet werden. Der gesamte Schaltplan eines Sensors ist in Abbildung 22 ersichtlich.

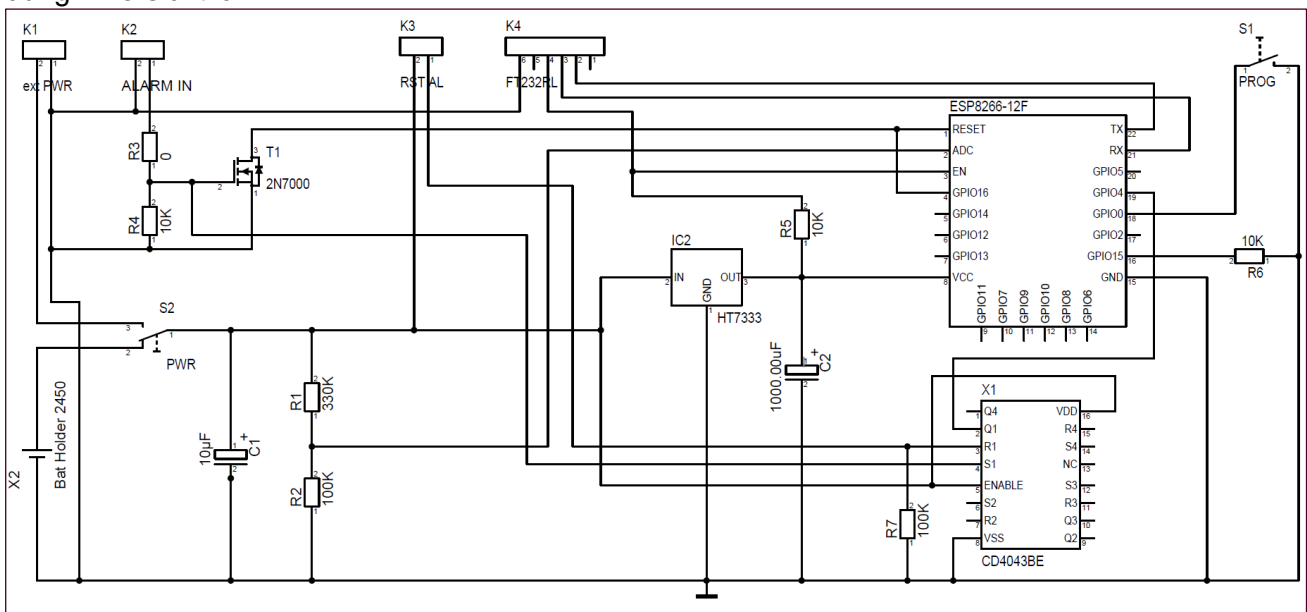


Abbildung 22: Schaltplan / technischer Aufbau eines Sensors (aus Target 3001)

Im folgenden Kapitel werden nun die einzelnen Stufen des Sensors ausführlich beschrieben und teilweise berechnet.

3.2.3 Hardware

Die Hardware eines vBMA Sensors wird in den folgenden Kapiteln schrittweise erklärt. Dafür werden die einzelnen Stufen aus dem Hauptschaltplan (siehe Abbildung 22) extrahiert und separat beschrieben.

3.2.3.1 Stromversorgung

Die Stromversorgung (Abbildung 23) kann über einen Mikroschalter (S2) ausgeschaltet oder auf eine externe Stromversorgung umgeschaltet werden. Da der ESP8266 12F Mikrokontroller nur einen Betriebsspannungsbereich von 3-3,6VDC gemäss Datenblatt zulässt und die LiPo-Akkus LIR2450 eine Klemmenspannung von bis zu 4,3VDC erreichen muss ein sogenannter LDO (Low Dropout) Spannungsregler (IC2) eingebaut werden, welcher einen möglichst kleinen Ruhestrom (Quiescent Current) aufweist. Dieser liegt gemäss Datenblatt bei 4-8µA. Um Schwankungen und Spitzen zu glätten, soll dieser am Eingang mit einem 10µF Kondensator (C1) beschaltet werden. C2 wurde weitaus grösser gewählt (1000µF), da während dem Aufstarten und vor allem während dem Senden via WiFi des ESP8266 12F hohe impulsförmige Ströme fließen, welche für eine Batterie und auch für den Spannungsregler kaum zu bewältigen sind.

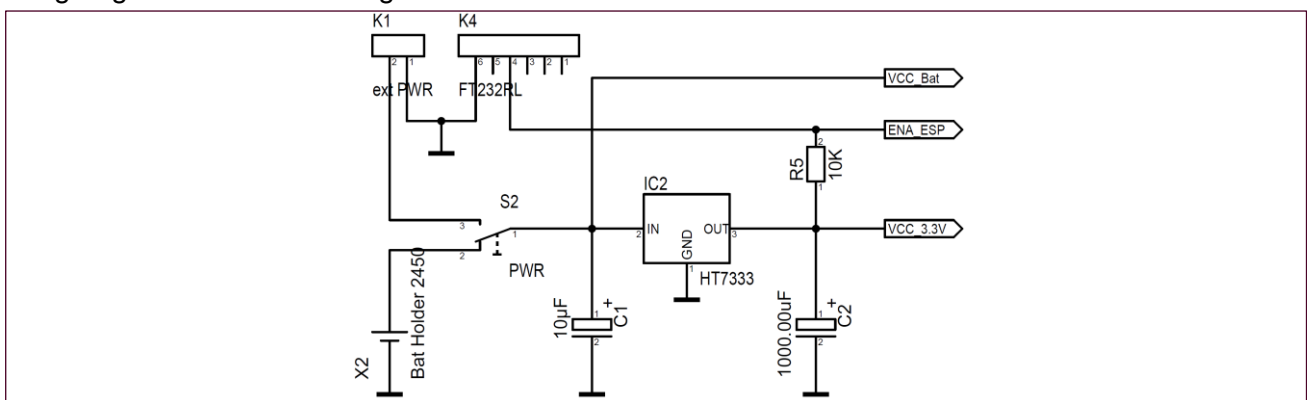


Abbildung 23: Schaltplan / technischer Aufbau Stromversorgung Sensor



Abbildung 24: impulsförmige Stromspitzen mit und ohne C2

Die Abbildung 24 zeigt die Impulse mit und ohne Kondensator. Die Spitzen wurden über einem 10Ω Widerstand im Stromkreis der Stromversorgung des Mikrokontrollers gemessen. Die Stromspitzen erreichen daher ca. 200mA und können mit einem grossen Kondensator fast gänzlich eliminiert werden, so dass dies für die Spannungsquelle keine Probleme mehr darstellt, da gemäss Datenblatt des Akkus (siehe Anhang C.D LIR2450) der maximale Entladestrom 120mA beträgt.

Der Widerstand R5 wird verwendet, um dem Enable-Pin des ESP8266 12F permanent auf HIGH zu setzen damit dieser aufstarten kann. Dies auch wenn der FT232RL-Programmer angeschlossen wird.

Die Stromversorgung mit den LIR2450 LiPo-Zellen soll mindestens eine Betriebsautonomie von 4 Wochen ermöglichen. Eine Berechnung zeigt, dass dies mit diesem Akku-Typ nicht ganz möglich ist, wenn alle Ruhestrome betrachtet werden. Folgende Berechnung zeigt dies deutlich.

Geg: $I_{R2} = 10\mu A$
 Ruhestrom HT7333 = $I_{QHT7333} = 8\mu A$
 ESP Energiesparstrom = $I_{QESP} = 16\mu A$
 RS-Flipflop CD4043BE = $I_{QCD4043BE} = 2\mu A$
 Batteriekapazität LIR2450 = $Ah = 120mA/h$
 ESP Betriebsstrom = $I_{Betrieb} = 70mA$
 Aufstartvorgänge pro Tag à 30" = $n = 8$

Ges: Ruhestromverbrauch = $I_Q / Tag = ?$
 Betriebsstromverbrauch = $I_B / Tag = ?$
 maximale Betriebsdauer in Tagen = h_{max}

Berechnungen:

$$I_{Qges} = I_{R2} + I_{QHT7333} + I_{QESP} + I_{QCD4043BE}$$

$$I_{Qges} = 10\mu A + 8\mu A + 16\mu A + 2\mu A = 36\mu A$$

$$I_{Q/Tag} = 24h \cdot 36\mu A = 864\mu A/Tag$$

$$I_{Betrieb/Tag} = 24h \cdot 36\mu A = 864\mu A/Tag$$

$$I_{B/Tag} = \frac{n \cdot t}{h} \cdot I_{Betrieb}$$

$$I_{B/Tag} = \frac{8 \cdot 30''}{60} \cdot 70mA = 4.66mA/Tag$$

$$I_{ges/Tag} = I_{B/Tag} + I_{Q/Tag}$$

$$I_{ges/Tag} = \frac{4.66mA}{Tag} + \frac{864\mu A}{Tag} = 5.524mA/Tag$$

$$h_{max} = \frac{Ah}{I_{ges/Tag}} = \frac{120mA/h}{5.524mA/Tag} = \mathbf{21,72Tage}$$

| Werte einsetzen

| 36 μA /h im Energiesparmode

| 864 μA /Tag im Energiesparmode

| 864 μA /Tag im Energiesparmode

| Werte Einsetzen

| 4,66mA pro Tag Betriebsstrom

| Werte einsetzen

| Stromverbrauch pro Tag

| Die gesamte Betriebszeit beträgt 21,72 Tage

Eine Messung des Stromverbrauchs (Abbildung 25) bei einer Betriebsspannung von 3,6VDC hat den Gesamt ruhestrom (I_{Qges}) von 36 μA und den Betriebsstrom ($I_{Betrieb}$) von 70mA bestätigt.

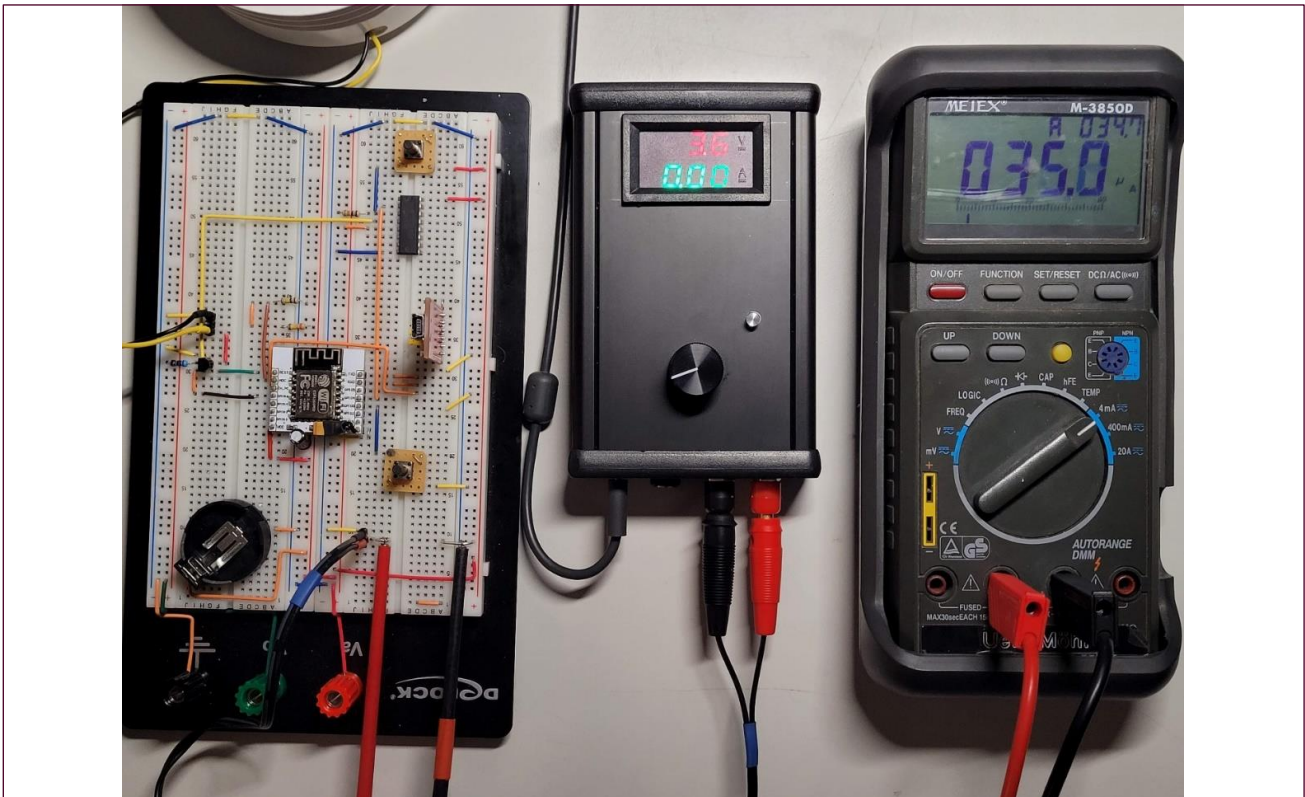


Abbildung 25: Messaufbau Ruhestrommessung (Energiesparmodus)

Aus zeitlichen Gründen konnte kein neuer Akku mehr gesucht und in das Projekt implementiert werden!

3.2.3.2 Mikrokontroller ESP8266 12F

Der Mikrokontroller wird direkt ab Spannungsregler mit 3.3VDC auf dem Pin 8 (VCC) gespeisen. Damit der ESP8266 12F aufstarten kann muss der Enable-Pin (EN) mit einem Potential von 3.3VDC anliegen. Zudem muss an GPIO15 ein Potential von 0VDC anliegen. Dieses Potential wird hochohmig über den 10kΩ Widerstand R6 auf Masse gelegt. Somit fließen dort keine grossen ausgleichsströme und es liegt jederzeit ein sauberes Potential an GPIO15. Wenn diese Bedingungen erfüllt sind, kann der ESP8266 12F aufstarten. Wenn zusätzlich zu dieser Beschaltung noch der Taster S1 gedrückt wird und somit an GPIO0 auch ein Potential von 0VDC während dem Aufstarten anliegt so wird der Mikrokontroller in den Programmiermodus versetzt und kann über die UART Schnittstelle (Rx/Tx-Pins) über den FT232RL-Programmer neu programmiert werden. Dies wird benötigt, um den Mikrokontroller an ein anderes WiFi-Netzwerk anzubinden oder wenn allfällige Codeänderungen vorgenommen werden müssen.

An GPIO4 liegt das Ausgangssignal des RS-Flipflops an. Wenn das RS_Alarm-Signal einen High-Pegel (3.3VDC) aufweist, so liegt ein Alarm an und es wird eine entsprechende Alarmmeldung versendet. Wenn an GPIO4 ein Low-Pegel (0VDC) aufweist, so werden nur die Betriebsdaten versendet.

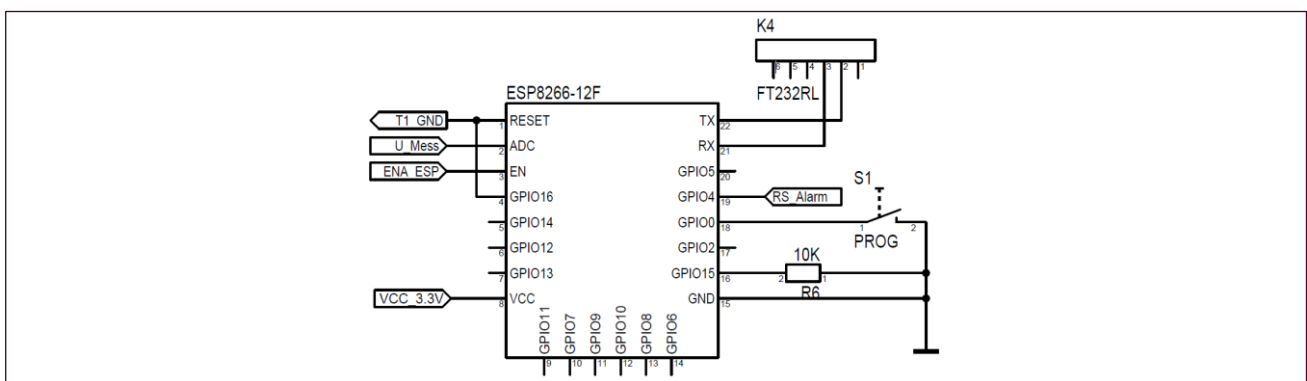


Abbildung 26: Schaltplan / technische Beschaltung des Mikrokontrollers

3.2.3.3 Betriebsdatenüberwachung

Für die Betriebsdatenerfassung wird einerseits der RSSI-Pegel (Radio Signal Strength Indikator) direkt aus der ESP8266WiFi.h Bibliothek ausgelesen. Dieser RSSI-Wert wird, wenn der Mikrokontroller und das WiFi Modul in Betrieb ist, permanent gemessen und kann im Programmcode abgerufen werden.

Andererseits wird über den ADC-Pin die Betriebsspannung gemessen, welche gemäss Spezifikation des Mikrokontrollers im Bereich von 0-1VDC sein darf. Daher muss die Betriebsspannung über einen sehr hochohmigen Spannungsteiler auf 1VDC verkleinert werden, so dass bei maximaler Betriebsspannung nicht mehr als 1VDC am Eingang des ADC Anschlusses und somit am Analog/Digital-Konverter anliegen. Zudem darf der Spannungsteiler für die Stromversorgung nur eine sehr kleine Belastung darstellen. Dieser gemessene Spannungswert wird nun mit einem 10-Bit A/D-Wandler in einen digitalen Wert von 0 bis 1023 konvertiert und kann im Code als solcher weiterverarbeitet werden.

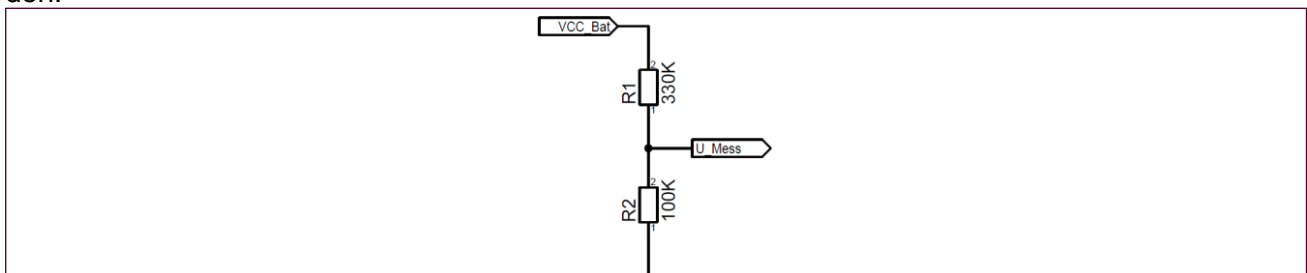


Abbildung 27: Spannungsteiler für die Betriebsdatenerfassung

Berechnung des Spannungsteilers

Geg: $R_2 = 100\text{k}\Omega$ Ges: $R_1 = ?$ Formel²⁰: $U_{20} = \frac{R_2}{R_2+R_1} \cdot U$
 $U_{\text{Messmax}} = 1\text{V} = U_{20}$
 $U_{\text{VCCBat}} = 4.5\text{V} = U$

Berechnungen:

$$U_{\text{Messmax}} = \frac{R_2}{R_1 + R_2} \cdot U_{\text{VCCBat}} \quad | \text{ nach } R_1 \text{ umstellen } \rightarrow \div U_{\text{VCCBat}}$$

$$\frac{U_{\text{Messmax}}}{U_{\text{VCCBat}}} = \frac{R_2}{R_1 + R_2} \quad | \cdot \frac{1}{x}$$

$$\frac{U_{\text{VCCBat}}}{U_{\text{Messmax}}} = \frac{R_1 + R_2}{R_2} \quad | \cdot R_2$$

$$\frac{U_{\text{VCCBat}} \cdot R_2}{U_{\text{Messmax}}} = \frac{(R_1 + R_2) \cdot R_2}{R_2} \quad | - R_2$$

$$\frac{U_{\text{VCCBat}} \cdot R_2}{U_{\text{Messmax}}} - R_2 = R_1 \quad | \text{ umstellen}$$

$$R_1 = \frac{U_{\text{VCCBat}} \cdot R_2}{U_{\text{Messmax}}} - R_2 \quad | \text{ einsetzen}$$

$$R_1 = \frac{4.5\text{V} \cdot 100\text{k}\Omega}{1\text{V}} - 100\text{k}\Omega = 4.5 \cdot 100\text{k}\Omega - 100\text{k}\Omega = \mathbf{350\text{k}\Omega}$$

Da für die Entwicklung nur die Widerstandsreihe E12 zur Verfügung stand entschied ich mich für einen etwas kleineren Widerstand von 330kΩ, da zum nächstgrösseren eine grössere Differenz besteht (390kΩ) als zum kleineren. Diese Differenz kann später im Code etwas kompensiert werden, da im Code der ADC-Wert (0-1023) mit der $U_{\text{VCCBat}} / 1023$ multipliziert wird und somit als exaktes Verhältnis zu U_{Messmax} (1VDC) darstellt. Die im Code angegebene U_{VCCBat} kann nun im Wert leicht verändert werden, um so den geringeren Widerstand zu kompensieren. Die Spannung wird etwas zu tief gemessen. Somit muss U_{VCCBat} nach oben korrigiert werden.

3.2.3.4 Alarmierung

Wie bereits in Kapitel 3.1.1 beschrieben, wird das Alarmsignal (siehe Abbildung 28) an den Signalpins des Rauchmelders abgegriffen und zum Schalten eines MOSFET N-Kanal Transistors (T1) verwendet, da das gleiche Schaltsignal auch zum Setzen des RS-Flipflops (X1) verwendet wird wobei ein Schaltpegel >3VDC am Gate des Transistors benötigt wird (siehe Datenblatt 2N7000 im Anhang C). Das Alarmsignal liegt mit ca. 7VDC während einer Sekunde an dem Gate des MOSFET Transistors an.

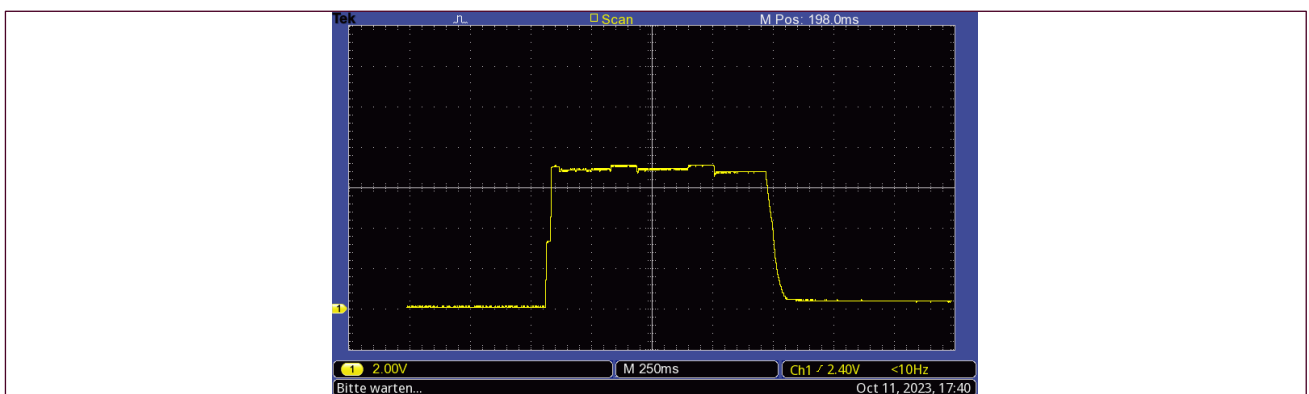


Abbildung 28: Alarmsignal des Rauchmelders

²⁰ (Europa Lehrmittel, 2020) Spannungsteiler Seite 56 unbelasteter Spannungsteiler

Der Transistor legt im Alarmfall den Reset Pin (RST) des ESP8266 12F während einer Sekunde auf Masse. Da der Mikrokontroller im Energiesparmodus (Deepsleep) am Reset-Pin einen HIGH-Pegel aufweist. Durch diesen Spannungsabfall am Reset-Pin des Mikrokontrollers, wird dieser aufgestartet.

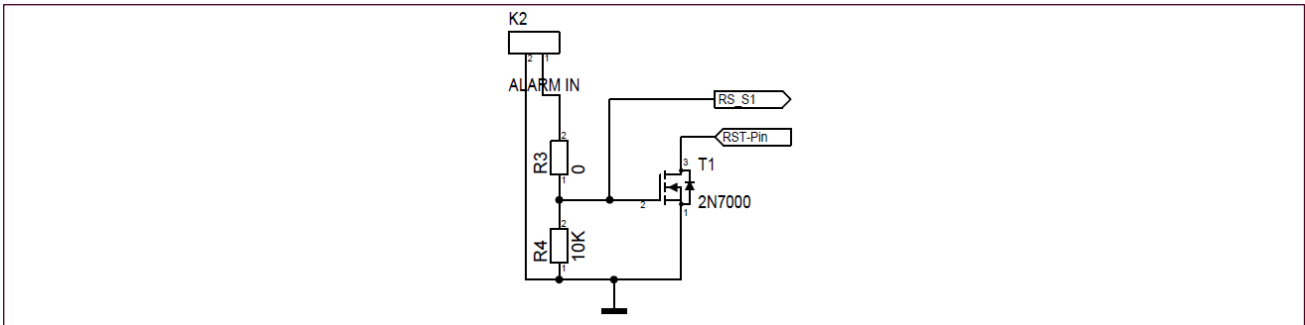


Abbildung 29: Alarmauslösung / Ansteuerung des MOSFET

Der Widerstand R4 wird verwendet damit T1 am Gate immer einen klar definierten Zustand (Low) aufweist. Ansonsten würde der Alarm bei einer Berührung oder einer anderen statischen Veränderung sofort den Transistor durchsteuern und so den Mikrokontroller immer wieder neu starten lassen. Der Widerstand R3 stammt noch aus der ersten Entwicklungsphase als noch kein RS-Flipflop verwendet wurde und an Stelle des MOSFET-Transistors ein normaler bipolarer Transistor eingesetzt wurde. So konnte die Basis Emitter Spannung etwas eingestellt werden.

Da der Alarmimpuls auch verwendet wird, um später im Code eine Unterscheidung von Alarm- und Betriebsdatenübermittlung zu machen muss der Alarmimpuls gespeichert werden, da dieser nur während einer Sekunde aktiv ist. Der Mikrokontroller braucht jedoch eine Weile, um aufzustarten und sich mit der Zentrale zu verbinden. Erst wenn dieser aufgestartet ist kann der eigentliche Alarm ausgelesen werden. Daher wird der Alarm in einem RS-Flipflop (CD4043BE, siehe Datenblatt im Anhang C) gespeichert. Dieses benötigt mindestens 5VDC, um dieses zu setzen. Da am Gate von T1 ein Signalpegel von 7VDC anliegt reicht dies aus um den Ausgang (Q1) des RS-Flipflops (X1) auf den Pegel High zu bringen. Dieser Pegel liegt nun an GPIO4 des Mikrokontrollers so lange an, bis das RS-Flipflop mit dem Reset-Taster, welcher an K3 angeschlossen wird, zurückgesetzt wird. Solange das RS-Flipflop nicht zurückgesetzt wird, gibt der Mikrokontroller bei jedem Aufstarten einen Alarm aus.

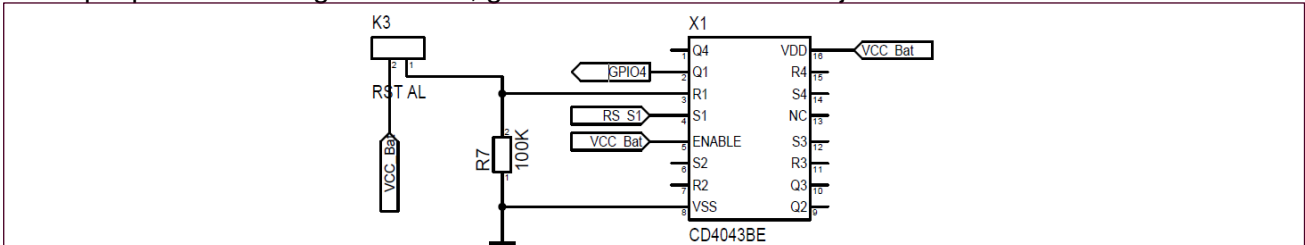


Abbildung 30: Alarmspeicherung in einem RS-Flipflop CD4043BE

Widerstand R7 hat die gleiche Funktion wie Widerstand R4, da das RS-Flipflop auch auf der CMOS Technologie aufgebaut ist und gleiche Effekte mit unklaren Potentialen aufweisen kann. Die Stromversorgung wird direkt ab der LIR2450 Zelle sichergestellt, da der Eingangsspannungsbereich zwischen 3V und 18V liegen kann.

3.2.4 Software

3.2.4.1 Kommunikation

Für die Übertragung sämtlicher Daten, welche im Sensor produziert werden, wird das schlanke MQTT-Protokoll verwendet. Dieses eignet sich für solche IoT-Projekte bestens, da es nur wenig Daten übermitteln soll und die Ressourcen des Mikrokontrollers aus energiespartechnischen Gründen nur leicht und kurz belastet werden sollen. Die zu übertragende Datenmenge besteht in jeder Nachricht (Topic) nur aus ein paar wenigen Zeichen. Auf den Sensoren werden ausschliesslich Daten publiziert, das heisst die produzierten Daten werden an den Broker über das WiFi-Netzwerk gesendet. Die Adressierung erfolgt über die sogenannten Topics oder Themenbereiche. Für jedes Thema auf jedem Sensor kann ein eigenes Topic eröffnet werden. Diese können dann beim Broker von einem anderen Benutzer (Client) abonniert werden. Die Architektur von MQTT ist im Kapitel 2.2 bereits illustriert (Abbildung 4). Damit das MQTT-Protokoll zum Publizieren von Nachrichten auf dem ESP8266 12F verwendet werden kann, muss die entsprechende Bibliothek im Sourcecode eines jeden Sensors eingebunden werden. Die Bibliothek von Arduino (ArduinoMqttClient.h) hat in Versuchen gezeigt, dass diese auch von der ESP-Mikrokontrollerfamilie genutzt werden kann und somit kompatibel ist. Die Datensicherheit ist einerseits durch den WiFi-Zugang und die implementierte Authentifizierung eines Benutzers gegeben. Zudem wird der Broker (Eclipse Mosquitto) auf einem lokalen Netzwerkinternen Raspberry-Pi betrieben. Optional könnten auch öffentliche MQTT-Broker für eine solche Anwendung verwendet werden. Die Datenqualität ist mit einem QoS-System (Quality of Service) ausgestattet. Dabei gibt es 3 verschiedene Qualitätsstufen²¹. In den Sensoren ist die mittlere Qualitätsstufe (QoS = 1) implementiert, welche dem Abonnenten einer Nachricht die Nachricht mindestens einmal gesendet und bestätigt wird. QoS 2 wird leider von der Bibliothek im Broker nicht unterstützt. Daher wurde die mittlere Qualitätsstufe gewählt. Die höheren Qualitätsstufen werden erst verwendet, wenn die Netzwerkstabilität, auf Grund von grossen Distanzen oder Abschirmung, nicht mehr gegeben ist.

Um das MQTT-Protokoll überhaupt übertragen zu können wird ein Zugang zu einem Netzwerk benötigt. Dies wird in jedem Sensor mit der ESP8266WiFi.h Bibliothek umgesetzt. Dabei werden im Sourcecode in den Definitionen die Zugangsdaten erfasst und ein WiFi Objekt erstellt, über welches dann im Programmcode verschiedene Werte oder Zustände wie Verbindungsstatus, IP-Adresse oder der RSSI-Wert ausgelesen und verarbeitet werden können (z.B. WiFi.RSSI() oder WiFi.status()). Die Zustände und Werte werden durch die Bibliothek erstellt und können im Code einfach abgefragt werden.

²¹ (Trojan, 2017) Qualitätssicherung (QoS) im Kapitel 1 Seite 16

3.2.4.2 Programmierung

Der Sourcecode eines Sensors ist im Anhang B1 ersichtlich. Grundsätzlich ist der Sensoren-Code nach dem Ablaufschema von Abbildung 31 entwickelt und realisiert worden.

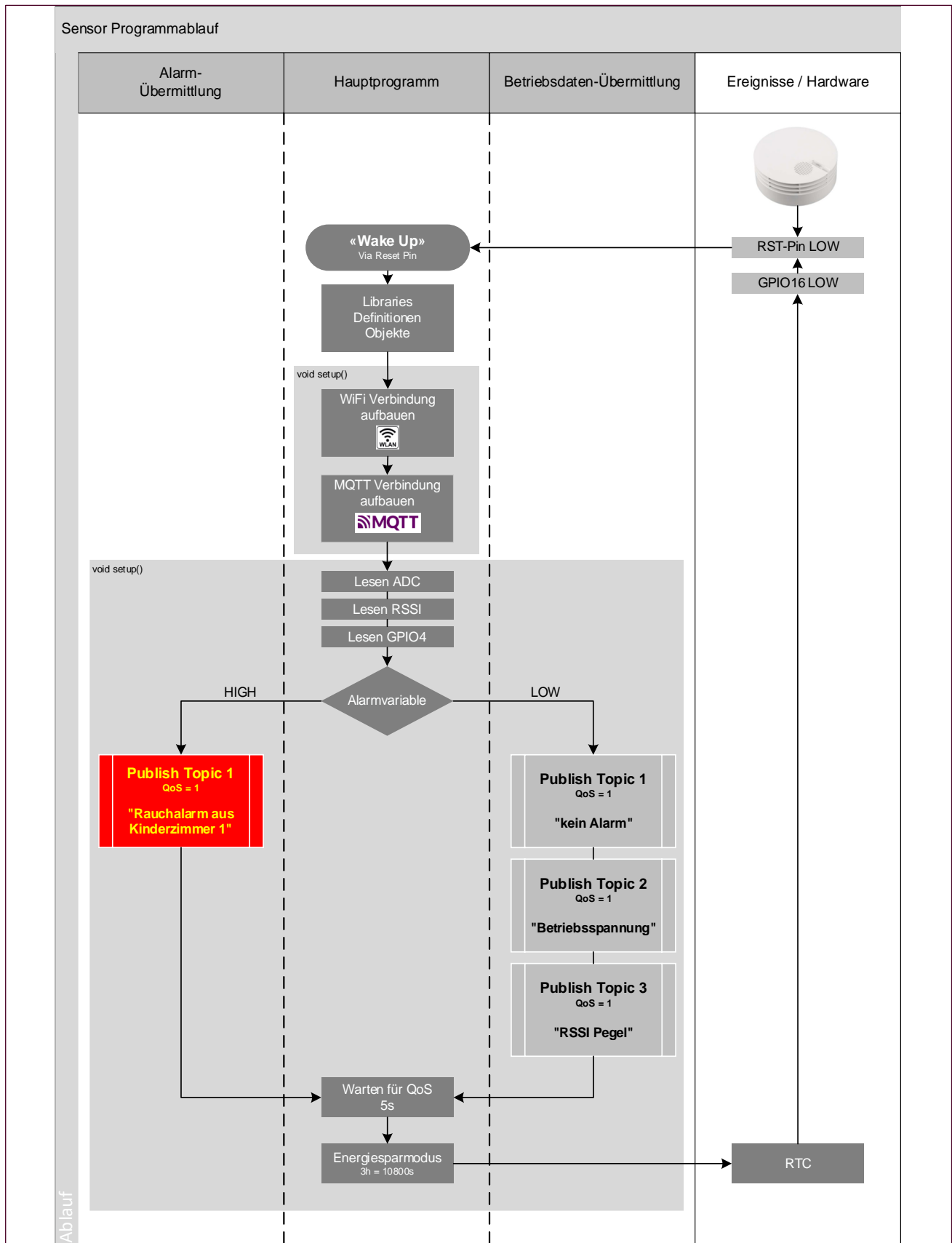


Abbildung 31: Ablaufschema des Sensorcodes

3.2.5 Umsetzung

3.2.5.1 Printplattenherstellung

Da bereits das Schema in Target-3001 gezeichnet wurde konnten nach Eingabe der Platinenabmessungen und Bohrungen die Bauteile auf dem Print nach eigenem Ermessen angeordnet werden. Es benötigte einige Versuche, bis das definitive Layout gefunden wurde. Nach dem Layout der Bauteile konnte mit der internen Autoroutingapplikation mit dem Routing der Leiterbahnen begonnen werden. Grundsätzlich ist darauf zu achten, dass die WiFi-Antenne des ESP8266 12F nicht in der Nähe von Leiterbahnen platziert wird. Dieser Umstand musste bereits bei dem Layout der Bauteile beachtet werden. Je nach Layout wurden nicht bei jedem Routing-Durchgang alle Leiterbahnen verlegt. Daher musste der Vorgang mehrere Male wiederholt werden bis das definitive Printplattenlayout (siehe Abbildung 32) mit Leiterbahnen feststand. Das Printplattenlayout wurde im Anschluss als sogenanntes Gerber-File aus Target-3001 exportiert und konnte auf der Website des Platinenherstellers (jlcpcb.com) direkt zur Fertigung der Printplatten hochgeladen werden.

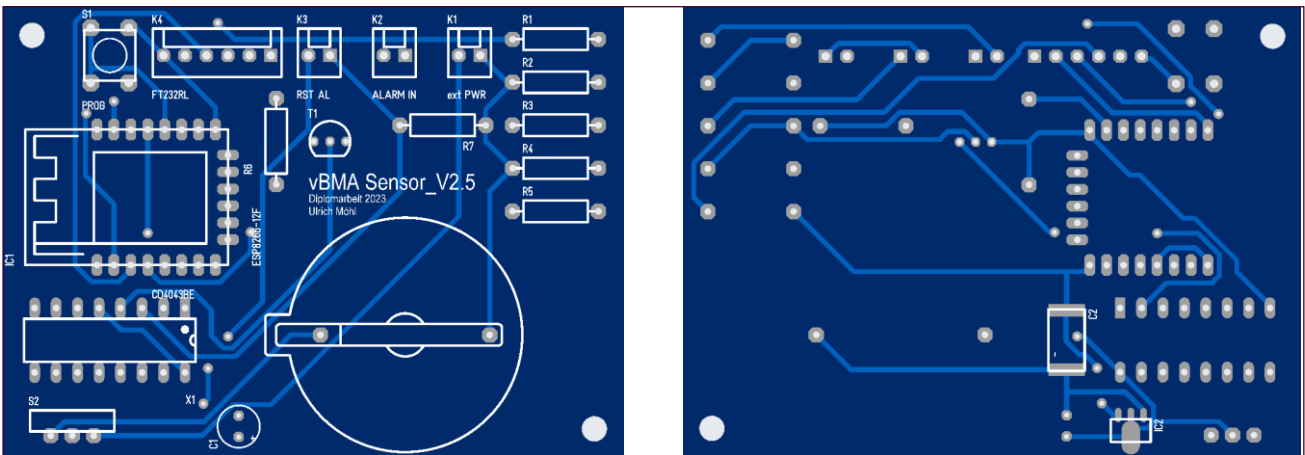


Abbildung 32: Printplattenlayout mit Vor- und Rückseite

Die Bestückung der Printplatten mit den Bauteilen und die Konfektionierung der Anschlusskabel bot keinerlei Probleme bis auf das Auflöten des ESP8266 12F Mikrokontrollers. Die Mikrokontroller wurden erst initial mit dem unter dem Kapitel 3.2.5.2 beschriebenen Programmer programmiert und anschliessend aufgelötet. Dabei stellten sich wieder die gleichen Probleme ein wie beim Auflöten auf das Breakoutboard. Einige der Mikrokontroller musste ich mit einem Heissluftgerät wieder entfernen, den Print reinigen und habe dann mit einer anderen Lötabfolge der Pins einen neuen Mikrokontroller aufbringen können.

3.2.5.2 Programmierung (Programmiersockel)

Da die Gefahr bestand, dass etwas mit der ersten Version der Printplatte nicht funktioniert, insbesondere die Programmierung nach dem Auflöten des Mikrokontrollers, war es naheliegend den Mikrokontroller vorgängig ohne Breakout-Board zu programmieren. Dies war nur mit einem Programmiersockel (siehe Abbildung 33) möglich in welchen die ESP8266 12F Mikrokontroller eingesteckt werden um dann via USB-Schnittstelle programmiert werden können ähnlich wie die Programmierung eines Entwicklungskits.

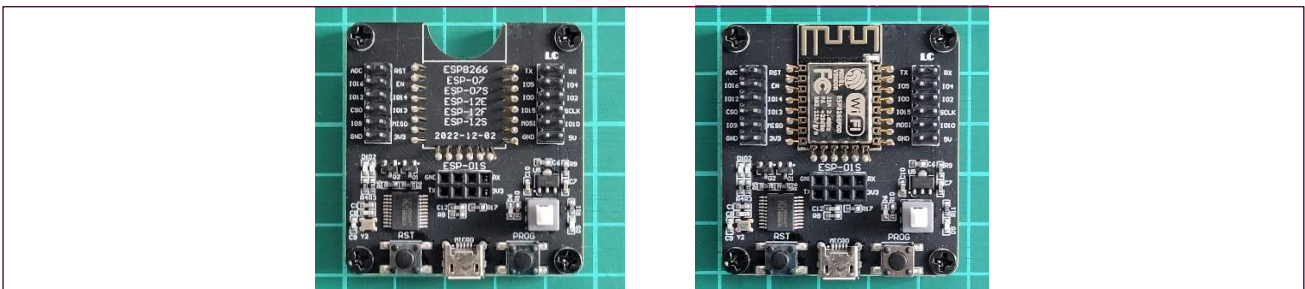


Abbildung 33: Programmiersockel mit und ohne Mikrokontroller

3.2.5.3 Gehäuse

Das Gehäuse für die Printplatte wurde den Abmessungen des Rauchmelders angepasst, damit dieser direkt auf das Gehäuse (siehe Abbildung 34) geschraubt werden kann. Das Gehäuse wurde in Autodesk Inventor gezeichnet und als dreidimensionales Objekt in einem .stl File abgelegt. Dieses File konnte dann über ein Slicer-Programm auf einem 3D-Drucker ausgedruckt werden.

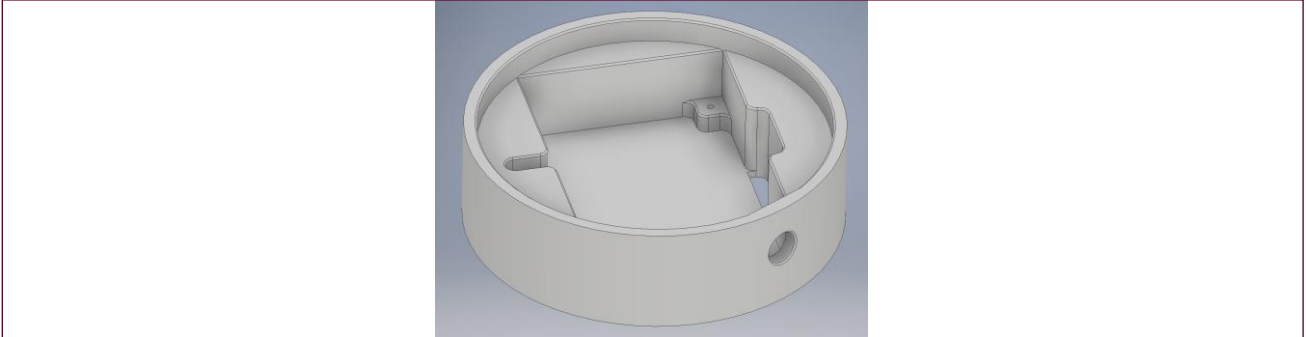


Abbildung 34: Zeichnung des Gehäuses für die Printplatte

Nach der Fertigstellung von Printplatte, Gehäuse und Kabel musste noch der Rauchmelder mit der mit der Platine über einen Stecker verbunden und der Reset Taster für den Rücksetzvorgang des RS-Flipflops eingebaut werden. Das Endergebnis des Sensors ist in Abbildung 35 ersichtlich.



Abbildung 35: Fertiger Rauchmelder mit Printplatte

3.3 Zentrale

3.3.1 Funktionsbeschreibung Zentrale

Als Primärfunktion der vBMA werden alle Alarme, welche von den Sensoren detektiert werden und mittels MQTT-Nachricht über das Netzwerk publiziert werden, als erstes an den MQTT-Broker gesendet, welcher als Hintergrundprogramm auf dem Raspberry Pi läuft. Der MQTT-Broker verteilt dann die Alarmnachricht an die Abonnenten der Topics 1, 4, 7 und 10. Die Abonnenten dieser Topics sind einerseits Node-RED auf dem Raspberry Pi und andererseits der Alarm-Mikrokontroller, welcher die gesamte Alarmierung lokal und dezentral übernimmt. Der Alarm-Mikrokontroller welcher im Kapitel 2.3.1 für diese Funktion eruiert wurde, musste für die Realisierung durch ein neueres Modell ESP32UE DevKitC V4 (siehe Abbildung 36) ausgetauscht werden, da die Alarmzentrale in einem Aluminiumgehäuse untergebracht sein wird. Durch diese Materialwahl müssen sämtliche Antennen (WiFi, 3/4G) aus dem Gehäuse geführt werden, damit ein betrieb mit Gehäusedeckel noch möglich ist. Diese Version des ESP32 verfügt über einen IPEX-Antennensockel, wo eine externe Antenne angeschlossen werden kann. Diese Änderung hat betrieblich keine Konsequenzen. Es musste bei der Konstruktion und beim Printlayout etwas mehr Platz eingerechnet werden.



Abbildung 36: Ersatzmikrokontroller ESP32UE DevKitC V4 mit IPEX-Antennensockel

Die lokale Alarmierung besteht aus einem optischen kaltweissen Blitzlicht von zwei superhellen LEDs (angeschlossen an GPIO22) und aus einem akustik-Modul, welches einen lauten Alarmton ausgeben kann. Das akustik-Modul wird über ein Relais eingeschaltet (GPIO23), da die Strombelastung für einen ESP-GPIO zu gross wäre und die Betriebsspannung des akustik-Moduls grösser ist als die maximale Ausgangsspannung an einem GPIO. Zudem wird mit einer blauen LED (GPIO2) angezeigt, dass der Alarm-Mikrokontroller mit dem MQTT-Broker verbunden ist.

Die dezentrale Alarmierung wird über das Mobilfunkmodul ausgelöst. Dieses wird über die UART-Schnittstelle mit AT-Kommandos durch den Alarm-Mikrokontroller angesteuert (GPIO16/17). Das Mobilfunkmodul verfügt über eine prepaid SIM-Karte vom Mobilfunkanbieter SALT und bildet somit den zuverlässigen kostenpflichtigen Dienstleister für die Alarmierung.

Die Alarme werden zusätzlich von den Nodes (Knoten) von Node-RED abonniert, damit diese auch auf dem Dashboard textuell dargestellt werden können. Dies dient jedoch nicht der eigentlichen Alarmierung.

Als Sekundärfunktion werden die von den Sensoren abgesetzten periodischen MQTT-Betriebsdaten-nachrichten auf dem Raspberry-Pi durch den MQTT-Broker empfangen und von Node-RED abonniert um eine übersichtliche Auswertung der Betriebs- und Alarmdaten (Betriebsspannung, RSSI-Pegel und Alarme) auf einem Dashboard darzustellen. Kritische Betriebsdaten werden in einem Node in Node-RED erkannt und über einen Whatsapp-Node an das Smartphone vom Benutzer gesandt. Diese Meldungen werden nicht über den gleichen Kommunikationskanal versandt wie die Alarmmeldungen, damit der Benutzer der Mobilfunknummer für die Alarmierung einen entsprechenden Klingelton zuweisen kann. Die Whatsapp-Nachricht wird von einem Chatbot an das Smartphone des Benutzers gesendet und weist dementsprechend eine andere Nummer auf.

3.3.3 Hardware

Die Hardware der vBMA Zentrale wird in den folgenden Kapiteln schrittweise erklärt. Dafür werden die einzelnen Stufen/Module aus dem Hauptschaltplan (siehe Abbildung 37) extrahiert und separat beschrieben.

3.3.3.1 Stromversorgung

Für die Entwicklung der einzelnen Module wurde für den Raspberry Pi ein entsprechendes Netzteil, welches genügend Strom liefern kann (2.5A bei 5VDC) verwendet. Das ESP32 DevKitC wurde erst über USB mit Strom versorgt später im Dauerbetrieb mit einem Breadboard-DC/DC Wandler. Das Mobilfunkmodul wurde ebenfalls mit dem DC/DC Wandler des Breadboards mit Strom versorgt. Da später die vBMA Zentrale mit einem externen Netzteil gespeist wird, musste das Netzteil erst dimensioniert werden. Sämtliche Module wurden einer Strommessung unterzogen, da die Herstellerangaben für diesen Zweck deutlich zu hoch angegeben sind. Die Strommessung wurde jeweils im DC-Pfad gemacht. Der jeweilige Spitzenwert wurde in Tabelle 12 erfasst, dieser entstand bei dem Mikrokontroller und dem Mikrocomputer während dem Aufstarten. Beide wurden mit entsprechender Peripherie gemessen, welche in der vBMA zum Einsatz kommt.

	Signal	Herstellerangabe	Messwert	Berechnungswert
Raspberry Pi 3B	VCC_Raspi	2.5A	476mA	0.6A
ESP32 DevKitC V4	VCC_ESP	0.5A	147mA	0.2A
SIM7600X Modul	VCC_SIM	300ma (Annahme)	166mA	0.2A
Sirenenmodul akustisch	VCC_AL	150mA	130mA	0.15A
Total		3.45A	919mA	1.15A

Tabelle 12: Stromkonsum der einzelnen Module

Somit reicht das Netzteil eines Raspberry Pi für die Stromversorgung aller Module aus, da dies einen maximalen Strom von 2,5A bei 5,1VDC liefern kann.

Das externe Netzteil wird über einen Micro-USB-Stecker mit der vBMA Zentrale verbunden, welche mit dem Stecker K1 verbunden ist. Der Kondensator C1 dient ausschliesslich der Stabilisierung der Stromversorgung. Die 5.1V werden dann an die verschiedenen Module herangeführt.

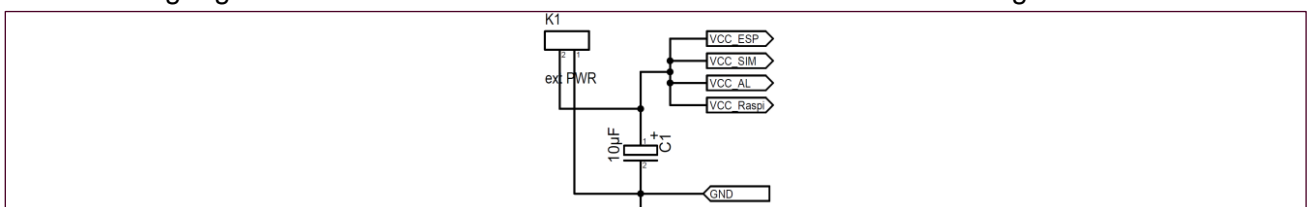


Abbildung 38: Schaltplan / technischer Aufbau der Stromversorgung

3.3.3.2 MQTT Broker / Node-Red Host (Raspberry Pi 3B)

Der Raspberry Pi 3B wird einerseits als MQTT-Broker verwendet und andererseits wird das Dashboard für die Alarm- und Betriebsdatenauswertung sowie die Kommunikation über Whatsapp verwendet. Zudem bildet dieser die Möglichkeit weitere Umgebungsdaten (Temperaturen, Lichteinfall, Regensensor etc.) von anderen Projekten auf zusätzlichen Dashboards dargestellt werden. Der Raspberry Pi muss über einen WiFi-Dongle an das W-LAN angebunden werden, da die interne WiFi-Antenne sich ebenfalls im Aluminiumgehäuse befindet.

Dafür wurde der Raspberry Pi 3B über die Konfigurationsdatei "interfaces", welche über die Kommandozeile aufgerufen wird, auf den externen W-LAN-Dongle umkonfiguriert. In diesem Konfigurationsfile muss der Anschluss von externen W-LAN Adapters freigeschaltet und das WiFi eingerichtet werden.

```
sudo nano /etc/network/interfaces

# WLAN
allow-hotplug wlan0
iface wlan0 inet manual
wpa-ssid "WLAN-NAME"
wpa-psk "WLAN-PASSWORT"
```

Abbildung 39: Konfiguration²² des externen W-LAN Adapters auf dem Raspberry Pi 3B

Der Raspberry Pi kann während dem Betrieb über eine Remote Desktop Verbindung von einem Computer aus erreicht werden, ohne dass Kabel installiert werden müssen. Dafür musste auf dem Raspberry Pi eine zusätzliche Applikation (xRDP²³) installiert werden, damit dies möglich ist.

Der Raspberry Pi 3B wird mit Distanzhülsen auf die Printplatte aufgeschraubt und über den 40-Pin-Header mit dem Stecker K6 verbunden, welcher die Stromversorgung sicherstellt. Die Verbindung zwischen Stecker K6 und 40-Pin-Header ist nicht dargestellt. Es werden die Pins 5V und GND dafür benötigt.

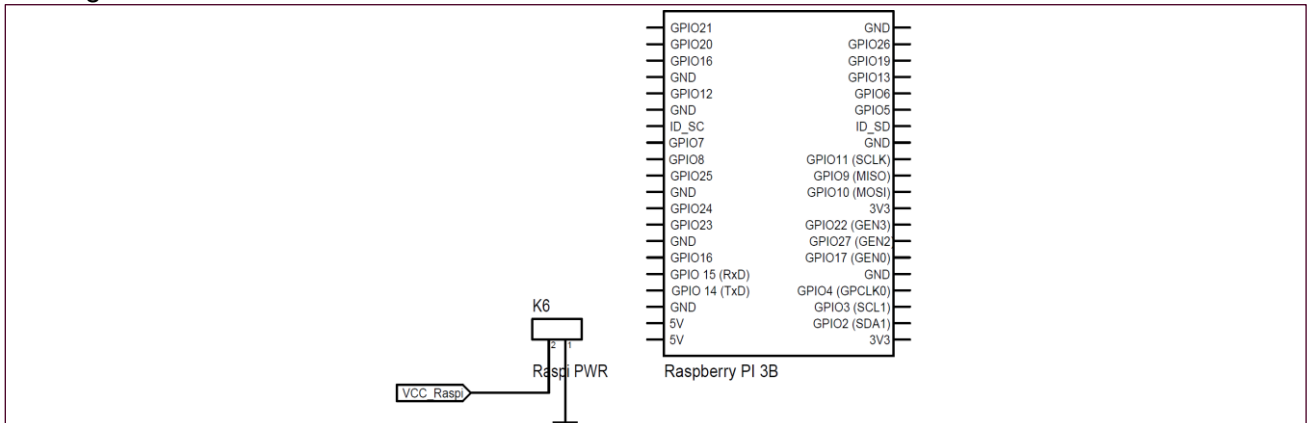


Abbildung 40: Schaltplan / technische Einbindung des Raspberry Pi 3B

²² (Dembowski, 2019) Konfigurationsdatei anpassen Kapitel 4.2.5

²³ (Tutorials for Raspberry Pi, 2023) Remote Desktop Verbindung herstellen

3.3.3.3 Alarmierungsmikrokontroller (ESP32 DevkitC 32UE V4)

Für den Alarm-Mikrokontroller wird wie bereits erwähnt wurde ein ESP32 DevkitC 32UE V4 verwendet, da dieser Mikrokontroller über eine IPEX (U.FL) Buchse für eine externe Antenne verfügt. Der Alarm-Mikrokontroller wird wie alle anderen Module auch mit 5VDC gespeisen. Die Einspeisung wird über den externen 5V Anschluss (Pin19) des Mikrokontrollers angeschlossen. Der interne DC/DC-Wandler gibt zusätzlich noch eine 3.3VDC Spannung aus, welche später für den Alarm-Reset noch verwendet wird, da die GPIOs nach wie vor nur Spannungen von 3.3VDC vertragen. Die blaue Leuchtdiode D2 wird verwendet, um die Konnektivität des Mikrokontrollers zum MQTT-Broker anzuzeigen. Grundsätzlich braucht die blaue LED keinen Vorwiderstand, da die Ausgangsspannung des ESP32 mit der Betriebsspannung der LED übereinstimmt. Auch die Strombegrenzung ist hinfällig, da der Ausgangsstrom eines GPIOs des Betriebsstromes der LED entspricht (20mA). Damit die LED nicht zu hell leuchtet, wurde Widerstand R3 vor- bzw. nachgeschaltet.

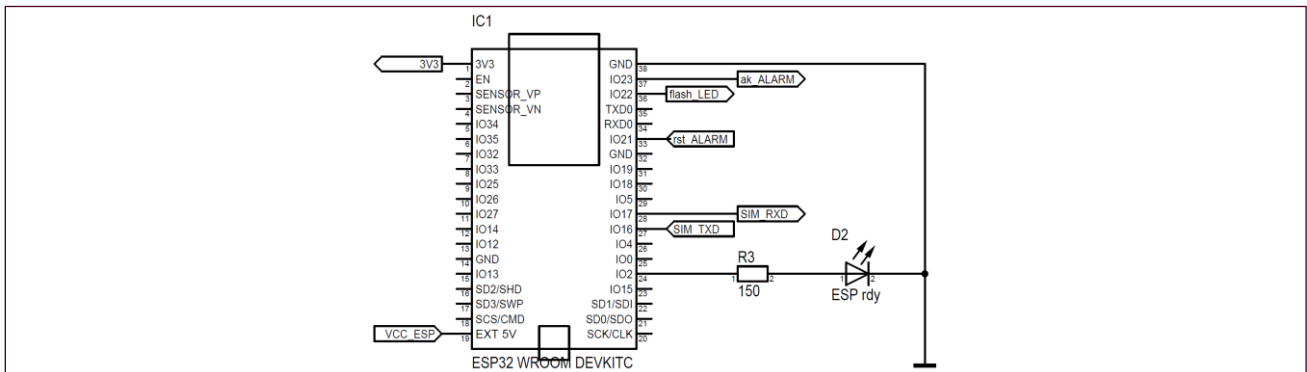


Abbildung 41: Schaltplan / technische Einbindung des ESP32 Alarm-Mikrokontrollers

Die Programmierung wird direkt über den Micro-USB-Anschluss des Mikrokontrollers gemacht, welcher die internen UART-Schnittstelle 0 (RX0/TX0) verwendet. Solange ein USB Kabel angeschlossen ist, kann die interne UART Schnittstelle 0 nicht verwendet werden. Dies ist auch ein Grund für die Verwendung dieses Typs von Mikrokontroller, da dieser über mehrere vollwertige UART Schnittstellen verfügt im Gegensatz zu der ESP8266 Familie. Dies ist auch auf dem Anschlussplan ersichtlich in Abbildung 42 ersichtlich.

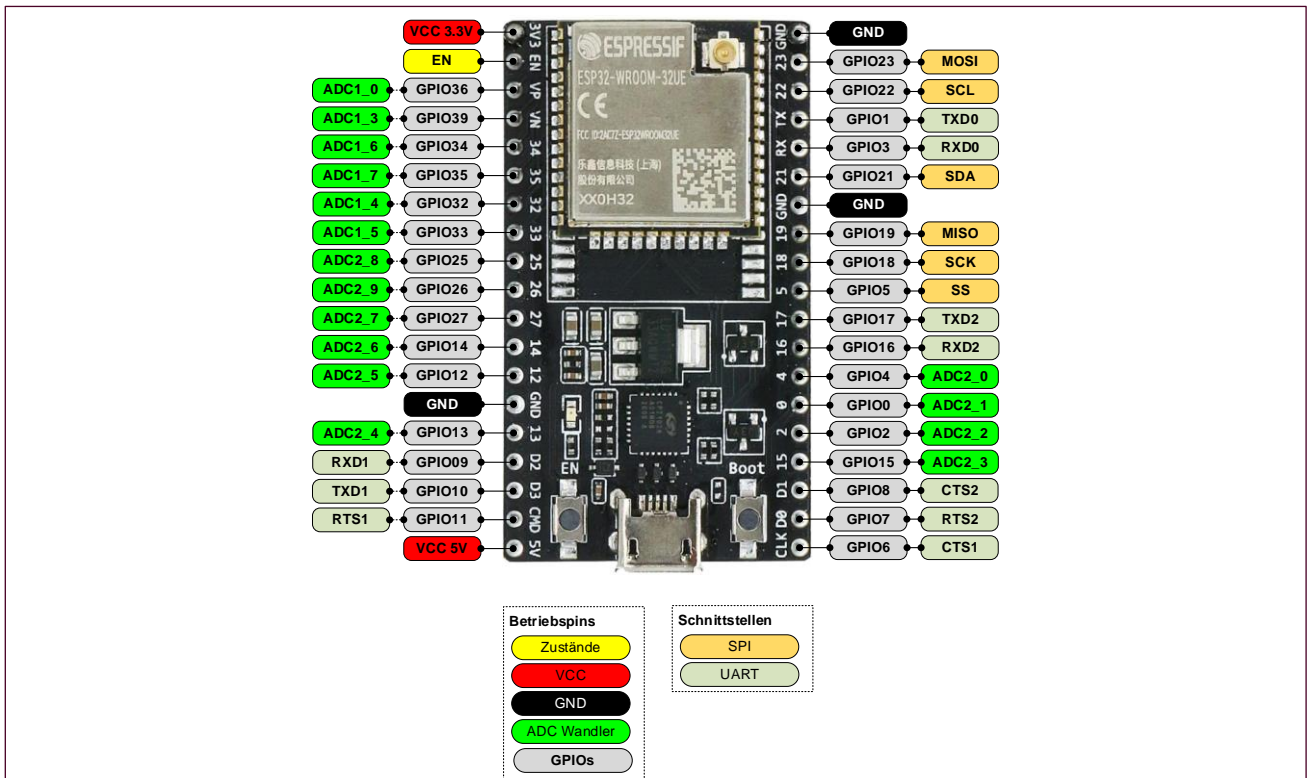


Abbildung 42: Anschlussplan eines ESP32 DevkitC 32UE V4

3.3.3.4 Externe Alarmierung (Waveshare LTE 4G HAT)

Die externe Alarmierung besteht im Wesentlichen aus einem Mobilfunkmodul. Das Mobilfunkmodul Waveshare SIM7600X 4G HAT verfügt über folgende Funktionalitäten:

- Konnektivität für Telefonie, SMS und Daten in den Netzen 2G (GSM/GPRS/EDGE), 3G (UMTS/HSPA+) und 4G (LTE);
- GNSS Satellitennavigation auf allen gängigen Systemen (GPS / GALILEO / GLONASS etc);
- Anbindung über verschiedene Netzwerkprotokolle möglich (IPv4, IPv6, TCP/IP etc);
- Verschiedene Hardwareanschlüsse für Headset, Speicherkarte, USB -Ports, UART Interfaces, SIM Slot und Antennenanschlüsse (IPEX).

Somit kann diese Mobilfunkmodul über verschieden Arten angesteuert werden. Entwickelt wurde es primär, um den Funktionsumfang in Verbindung mit einem Raspberry Pi zu nutzen. Daher verfügt das Modul auch über den 40-Pin-Header. Zusätzlich verfügt das Modul noch über ein Control Interface welches für andere Mikrokontroller wie Arduino, STM32 oder ESP entwickelt wurde.



Abbildung 43: Waveshare SIM7600X 4G HAT Mobilfunkmodul

Für die vBMA wird das Modul ausschliesslich über das UART-Interface (RXD/TXD) des Control Interfaces durch den Alarm-Mikrokontroller angesteuert, um eine Kurzmitteilung abzusetzen. Dafür werden die Pins 5V und GND für die Stromversorgung und die Pins RXD und TXD für die Kommunikation über die UART Schnittstelle mit dem Alarm-Mikrokontroller verwendet. Die sogenannten AT-Kommandos, welche im Code für das Versenden einer Kurzmitteilung verwendet werden, sind in Kapitel 3.3.4.1 noch ausführlicher erklärt.

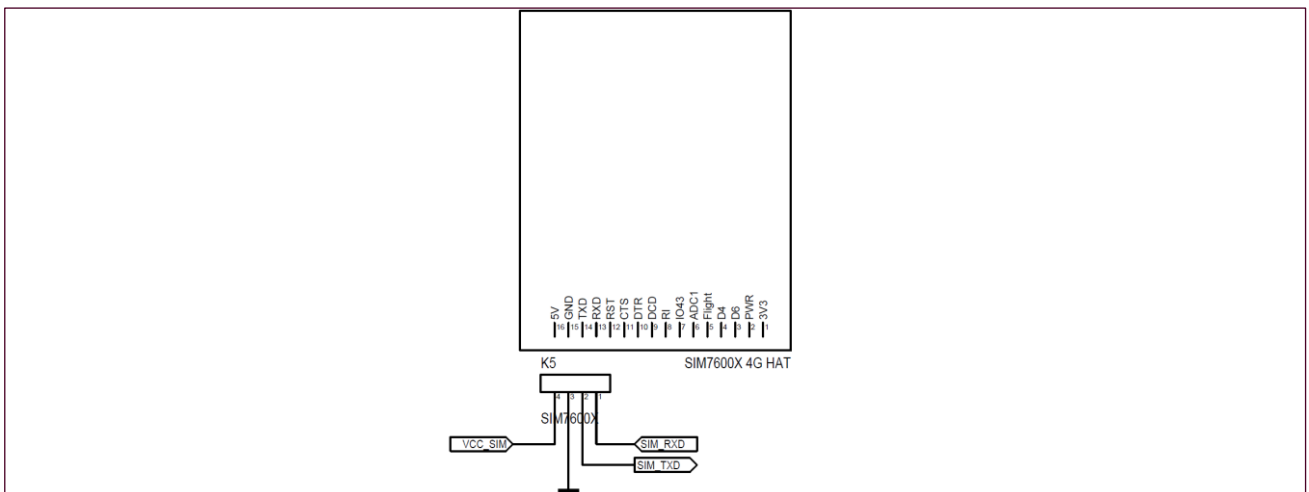


Abbildung 44: Beschaltung Mobilfunkmodul SIM7600X 4G HAT

Das Modul verfügt über eine eigene externe Antenne, welche für die Frequenzbereiche der 3 verschiedenen Technologien (2G/3G/4G) geeignet ist, wobei der Bereich 2G in der Schweiz nicht mehr kommerziell verwendet wird und umgenutzt wurde (4G/LTE).

3.3.3.5 lokale Alarmierung (akustisch / visuell)

Die lokale Alarmierung wird verwendet, um Alarme wie zum Beispiel aus der Garage, welche im Wohnhaus nicht mehr hörbar wären, dorthin weiterzuleiten. Die lokale Alarmierung wird einerseits durch ein Sirenenmodul mit Piezolautsprecher und andererseits durch zwei superhelle LEDs visuell signalisiert.

Sirenenmodul von Tru Components

Das Sirenenmodul besteht aus einer Platine und aus einem Piezo-Alarmlautsprecher. Das Sirenenmodul erzeugt laute auf- und absteigende Töne.

Spezifikationen²⁴

Betriebsspannung:	5-12VDC
Stromaufnahme:	200mA
Schalldruck (max):	110dB

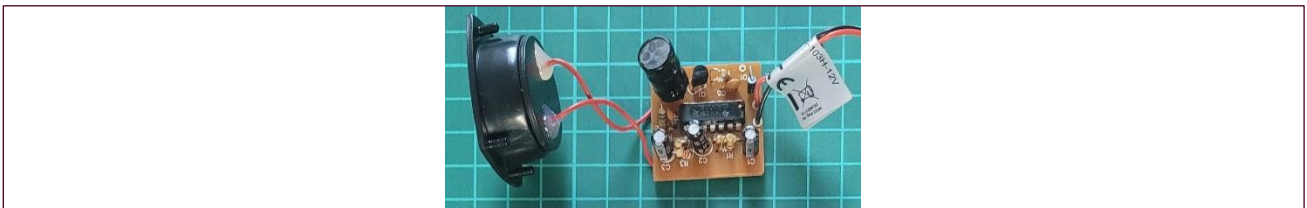


Abbildung 45: Sirenenmodul von Tru Components

Die Ansteuerung des Moduls kann nicht direkt mit einem GPIO des ESP32 realisiert werden, da dieser nicht für eine Speisung eines solchen Modules ausgelegt ist und die Versorgungsspannung eines GPIO zu niedrig ist (3.3VDC). Daher wird das Sirenenmodul am Stecker K4 über das Relais RE1 mit der 5VDC Versorgungsspannung (VCC_AL) mit einer ausreichenden Stromversorgung verbunden. Das 5V Relais RE1 wird über den Bipolartransistor T1 vom GPIO23 des ESP32 angesteuert. Wenn ein Alarm ausgegeben wird, so wird der GPIO23 mit einem HIGH-Pegel (3.3VDC) angesteuert. Der Basisstrom wird mit Widerstand R1 auf 260µA begrenzt und dieser Basisstrom schaltet den Transistor T1 durch. Somit wird das 5V Relais RE1 eingeschaltet und mit diesem auch das Sirenenmodul.

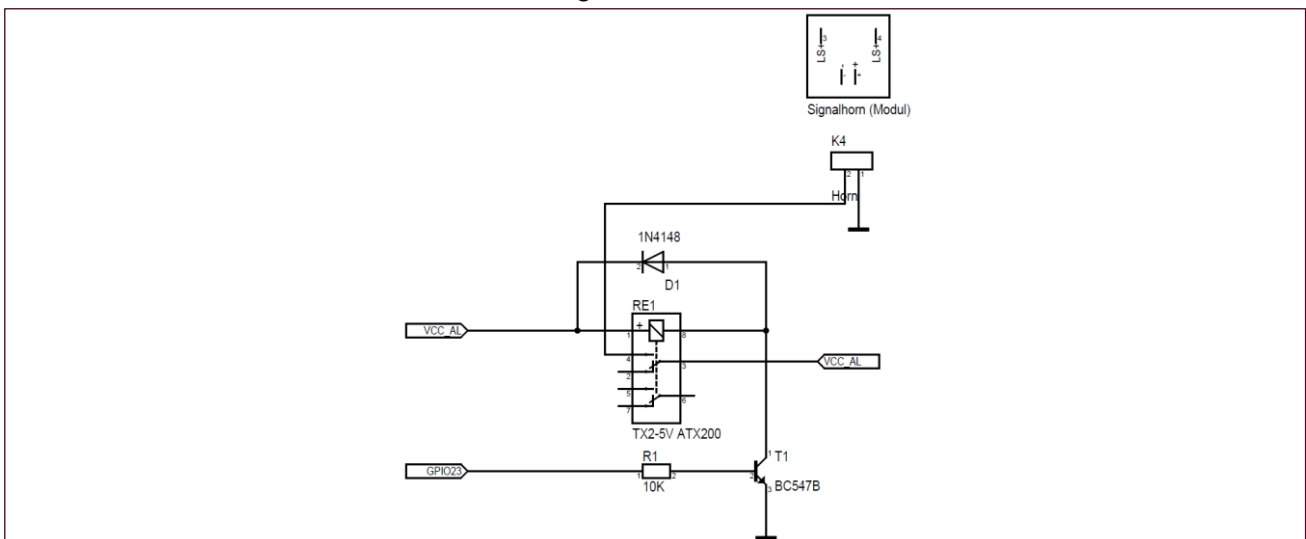


Abbildung 46: Schaltplan der Ansteuerung des Sirenenmoduls

²⁴ (Conrad Electronic SE, 2023) technische Daten des Sirenentreiber-Baustein 110dB

LED Flash

An den Stecker K4 werden zwei superhelle LEDs angeschlossen. Diese werden direkt mit der Ausgangsspannung des GPIO22 mit Strom versorgt, da die Ausgangsspannung exakt der Betriebsspannung der LEDs entsprechen (siehe Datenblatt in Anhang C.E). Der maximale Strom, welcher pro GPIO bezogen werden kann, beträgt gemäss Datenblatt²⁵ 40mA. Somit können 2 LEDs parallel direkt mit dem GPIO22 verbunden werden.

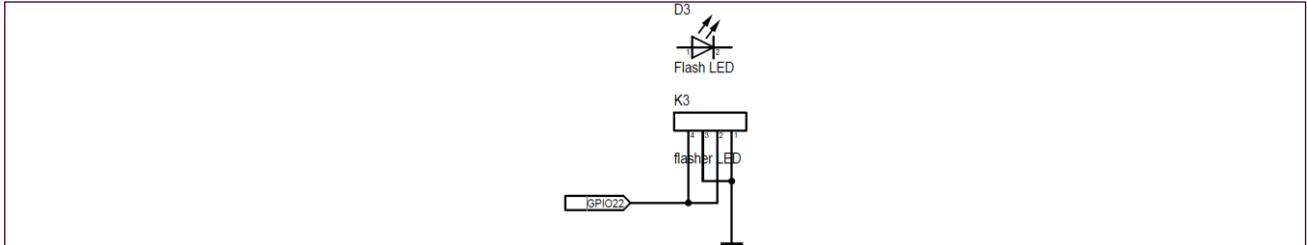


Abbildung 47: Schaltplan der Ansteuerung der superhellen LEDs

Alarmrücksetzung / Alarm-Reset

Der Code der vBMA Zentrale ist so entwickelt, das ein aktiver Alarm nur an der vBMA Zentrale durch einen Tastendruck zurückgesetzt werden kann. Erst wenn der Taster, welcher am Stecker K2 angeschlossen ist, gedrückt wird, schaltet sich der Sirenenalarm und die flash LEDs aus. Dies wird mit einem HIGH-Impuls (3.3VDC) an GPIO21 erreicht. Wenn dieser Impuls nicht anliegt, so muss ein klar definierter LOW-Pegel an GPIO21 anliegen. Daher ist GPIO21 hochohmig an Ground (GND) gezogen.

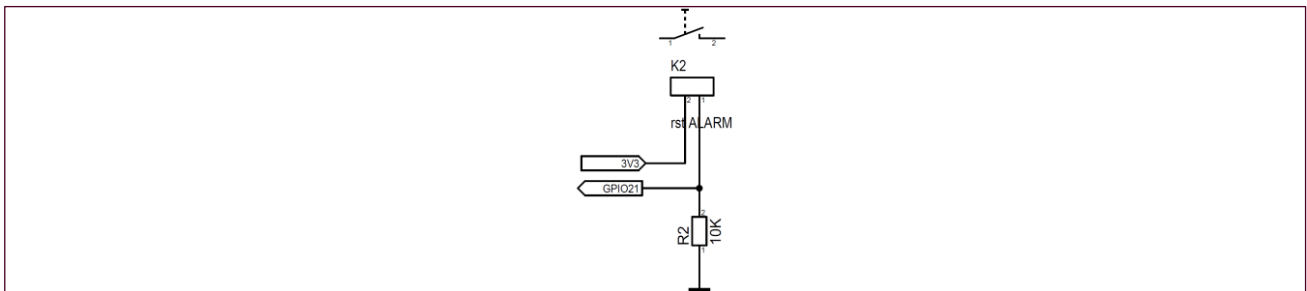


Abbildung 48: Schaltplan der für den Alarm-Reset

²⁵ (Espressif Systems Co, 2023) DC Characteristics High Level Source Current Seite 15

3.3.4 Software / Programmierung

In der Programmierung der vBMA Zentrale werden auf verschiedenen Ebenen unterschiedliche Kommunikationstechnologien, Frameworks, Protokolle und Kommandos verwendet, welche im folgenden Kapitel 3.3.4.1 kurz beschrieben werden, damit die Begrifflichkeiten in den nachfolgenden Kapiteln verständlich sind und im Arduino- sowie Node-RED-Sourcecode (siehe Anhang B) rasch wiedererkannt werden können.

3.3.4.1 Kommunikation

WiFi / W-LAN

Damit die vBMA Zentrale die Alarmer sowie die Betriebsdaten, welche als Rohdaten primär über das W-LAN übertragen werden, verwaltet und ausgewertet werden können müssen in der vBMA Zentrale der Raspberry Pi und der Alarm-Mikrokontroller über diese Konnektivität verfügen.

Der Raspberry Pi verfügt, wie bereits in Kapitel 3.3.3.2 beschrieben über einen externen WiFi-Dongle. Dieser muss mit dem gleichen Netzwerk verbunden werden wie alle anderen Module. Der Eclipse Mosquitto MQTT Broker sowie Node-RED sind somit automatisch mit diesem Netzwerk verbunden und müssen nicht wie der ESP32 Mikrokontroller über eine Bibliothek in das Netzwerk eingebunden werden.

Der ESP32 Alarm-Mikrokontroller wird über die WiFi.h Bibliothek mit dem Netzwerk verbunden. Hierfür müssen im Sourcecode (siehe Anhang B.B1) SSID (Netzwerkname) und das Passwort als char-Array deklariert werden. Weiter werden über verschiedene Methoden und Funktionen mit dem Netzwerk verbunden.

MQTT Broker²⁶

Damit das schlanke IoT-Protokoll MQTT überhaupt genutzt werden kann, wird ein sogenannter MQTT-Broker benötigt, welcher die Abonnenten (Subscriber) und Publizierer (Publisher) miteinander verbindet und die Abonnements verwaltet. Ein solcher Broker kann auf dezentralen sowie auf lokal eingebundenen Servern mit entsprechender Authentifizierung und Sicherheitsprotokollen betrieben werden. In der Heimautomation macht es aus meiner Sicht Sinn, diese Daten lokal zu verwalten.

Ein solcher MQTT-Broker wird über die Kommandozeile auf dem Raspberry Pi installiert und konfiguriert. Dabei wurde auch eine Authentifizierung eines jedes Publishers und Subscribers konfiguriert, damit die Sicherheit auch im Netzwerk als gegeben betrachtet werden kann. Die auszuführenden Schritte, welche in der Kommandozeile des Raspberry Pi ausgeführt werden müssen, sind in Abbildung 49 rudimentär beschrieben.

```
sudo apt install mosquitto mosquitto-clients

listener 1883
allow_anonymous false
password_file /etc/mosquitto/credentials

sudo mosquitto_passwd -c /etc/mosquitto/credentials mqttuser

listener 1883
allow_anonymous false
password_file /etc/mosquitto/credentials
```

Abbildung 49: Installations- und Konfigurationsschritte für den MQTT-Broker

²⁶ (Brühlmann, 2021) MQTT-Broker Seite 125

Auf dem Alarm-Mikrokontroller müssen nun die Alarm-Topics noch abonniert werden, damit ein lokaler sowie dezentraler Alarm ausgelöst werden kann. Dies wird mit der PubSubClient.h Bibliothek im Zentralencode realisiert (siehe Anhang B.B1). Um die verschiedenen Topics zu abonnieren, wird erst ein PubSubClient-Objekt definiert, dann kann direkt über die IP-Adresse des Raspberry Pi mit dem MQTT-Broker verbunden werden. Die zu abonnierenden Topics müssen in den Definitionen noch deklariert werden. Empfangen werden die verschiedenen Topics über die callback-Funktion (siehe Funktion 1 im Anhang B.B2). Somit steht dann die Payload-Nachricht des empfangenen Topics als String für die weitere Verarbeitung zur Verfügung und kann nun ausgewertet werden.

Node-RED

Node-RED ist eine Programmierumgebung, welche verschiedene Geräte, Webdienste und Schnittstellen miteinander verbindet. Node-RED wird primär für die Betriebsdatenauswertung und die Meldung von kritischen Betriebszuständen verwendet. In Node-Red werden einerseits die Betriebsdaten je Zimmer erfasst (als Subscriber/Abonnent), ausgewertet und andererseits in einem Dashboard übersichtlich dargestellt. Die Programmierung von Node-Red ist in Kapitel 3.3.4.3 genauer erklärt.

WhatsApp

WhatsApp wird in der Node-RED Programmierumgebung verwendet um kritische Betriebszustände wie Betriebsspannungen oder Signalpegel dem Benutzer auf das Mobiltelefon zu senden. Dafür wird in Node-Red ein "Send Message" -Node verwendet. Hinter diesem Node steckt ein Chatbot der Web Plattform CallMeBot.com. Um Nachrichten zu versenden, muss eine Initialnachricht (siehe Abbildung 50) an den Chatbot gesendet werden. In dem "Send Message"-Node muss dann der erhaltene API-Key angegeben werden. Nun können definierte Nachrichten aus Node-RED an das verifizierte Mobiltelefon in WhatsApp gesendet werden.

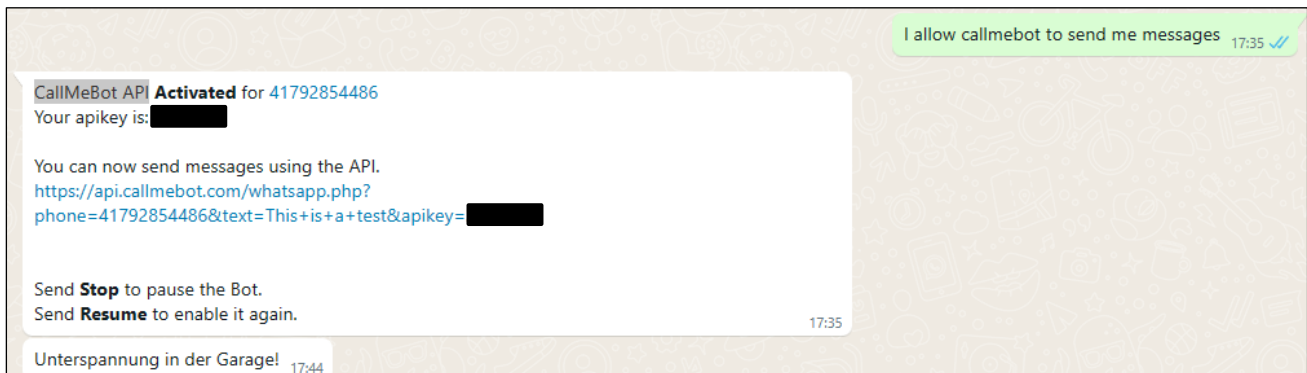


Abbildung 50: Initial- und Beispielnachricht der CallMeBot API

AT-Befehlssatz²⁷

Der AT-Befehlssatz oder die AT-Kommandos wurden in den 80er-Jahren durch die Firma Hayes Microcomputer Products entwickelt, um Markenunabhängige Modems anzusteuern. Dieser AT-Befehlssatz wurde dann von der ITU (International Telecommunication Union) als Standard festgelegt. Dabei steht AT für "ATtention" und steht immer vor einem weiteren Kommando, damit das Modem das nachfolgende Kommando erkennen kann. Dieser Befehlssatz wurde über die Jahre hinweg weiterentwickelt und so ist es möglich auch moderne Modems mit den heutigen Kommunikationstechnologien wie 3G oder 4G anzusteuern. Dabei werden die Befehle in verschiedenen Kategorien eingeteilt, um den Zweck der Kommandos zu spezifizieren wie Tests, Lesen, Setzen oder Ausführen.

²⁷ (Agnihotri, 2023) AT Commands, GSM AT Command Set

Dieser AT-Befehlssatz²⁸ wird in dieser Diplomarbeit für die Ansteuerung des Waveshare SIM7600X Mobilfunkmoduls verwendet. Aus diesem Grund wird für die Ansteuerung des Mobilfunkmoduls keine Bibliothek verwendet. Die in der Tabelle 13 erfassten AT-Befehle wurden im Sourcecode der vBMA Zentrale primär in der Funktion 3 implementiert und werden für das Versenden der Kurzmitteilung verwendet. Bei der Initialisierung des Alarm-Mikrokontrollers werden noch weitere AT-Befehle verwendet, um den Status des Mobilfunkmoduls abzufragen.

AT-Befehl	Erklärung	Beispiel
AT+IPREX	Setzt die Baudrate des Moduls permanent. Dieser Befehl wird im Code nicht ausgeführt.	C: AT+IPREX=115200 R: OK
AT	Prüft die Verbindung zwischen dem Mikrokontroller und dem Modul. \r wird im Code als Eingabe des Kommandos auf der seriellen Schnittstelle (Rx/Tx2) verwendet.	C: AT\r R: OK
AT+CMGF=1	Setzt das Modul in den SMS Mode. Dies wird mit dem Wert 1 signalisiert und mit \r eingegeben.	C: AT+CMGF=1\r R: OK
AT+CMGS	Setzt die Rufnummer, an welche die folgende Nachricht gesendet werden soll. Die Nachricht wird nach dem zweiten Backslash gesetzt durch	C: AT+CMGS=\"0792854486\" \" \" R: OK
Serial2.print(command)	Schreibt die Nachricht, welche in die Funktion durch die Finite State Machine eingegeben wurde zwischen die leeren Anführungszeichen des AT+CMGS Befehls.	
0x1A	Dieser Befehl entspricht Ctrl+Z welcher verwendet wird, um alle Eingaben abzusenden. Auf der seriellen Schnittstelle entspricht Ctrl+Z dem Hex-Wert 0x1A	

Tabelle 13: verwendete AT-Befehle und Eingaben

Da der AT-Befehlssatz sehr umfangreich ist, wurde anhand eines Codebeispiels²⁹, welches von Waveshare zur Verfügung gestellt wird, der Code für die Ansteuerung des Mobilfunkmoduls über die AT-Befehle verwendet.

3.3.4.2 Programmierung ESP32 DevkitC 32UA (Alarmierung)

Der Zentralencode wurde, wie auch der Sensoren-Code in der Arduino IDE entwickelt und besteht im Wesentlichen aus den folgenden vier Teilen:

- Definitionen (Konstanten, Variablen, Objekten, Bibliotheken)
- Konfigurationen (GPIOs, Startbefehle, Verbindungen)
- Hauptcode (Finite State Machine im folgenden FSM genannt)
- Funktionen (die Funktionen wurden übersichtshalber in Tabs ausgelagert)

Der Zentralencode mit seinen Funktionen wurde für eine übersichtlichere Darstellung in einem Ablaufschema dargestellt. Der Sourcecode der vBMA ist in Anhang B.B1-6 zur Einsicht abgelegt. Nachfolgend werden die Hauptfunktionen stichwortartig beschrieben und beziehen sich auf die Abbildung 51.

²⁸ (SIMCOM, 2021) SIM7500_SIM7600 Series_AT Command Manual

²⁹ (Waveshare Electronics, 2023) Arduino Demo

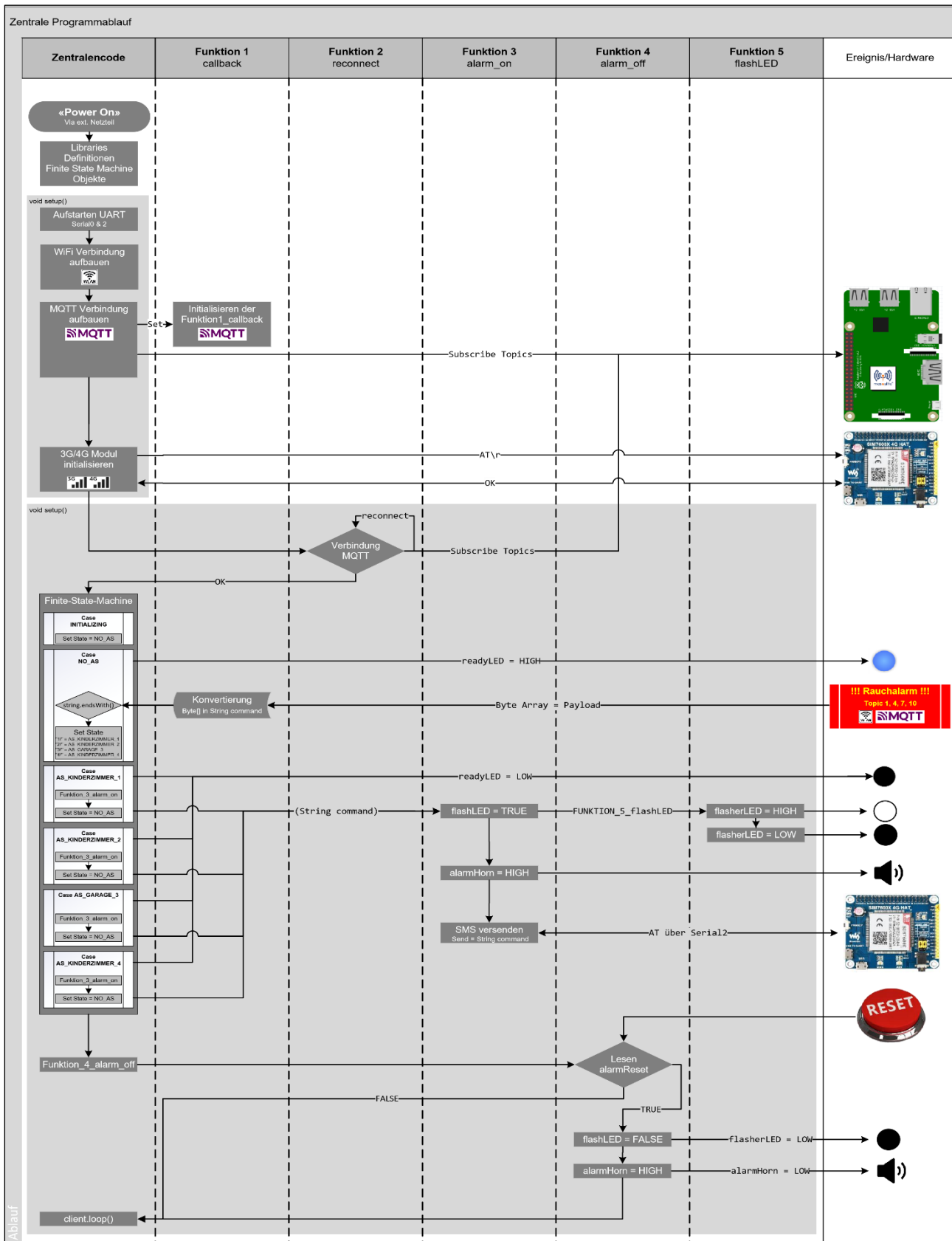


Abbildung 51: Sourcecode des Alarm-Mikrokontrollers

Codebeschreibung

- Startet automatisch nach einschalten der externen Stromversorgung ein, da die Zentrale im Dauerbetrieb verbleibt;
- Bibliotheken werden für die MQTT- und WiFi-Verbindung eingebunden;
- Sämtliche für den Betrieb, Auswertung und Alarmierung verwendeten Variablen werden global eröffnet, damit aus jeder Funktion darauf zugegriffen werden kann;
- Deklaration der FSM für die Alarmverwaltung, da jeder Alarm einen anderen State erhält, um individuell alarmieren zu können;
- Für den Betrieb müssen WiFi- und MQTT-Objekte erstellt werden, damit diese auf die Bibliotheken zugreifen können;
- Initialisieren der UART-Schnittstellen 0 und 2, damit einerseits der Benutzer über den seriellen Monitor die Aktionen im Code verfolgen kann und andererseits damit der Mikrokontroller mit dem Mobilfunkmodul kommunizieren kann;
- Aufbau von WiFi- und MQTT Verbindungen (zum MQTT-Broker auf dem Raspberry Pi);
- Initialisieren der callback-Funktion für den Empfang von MQTT Topics;
- Abonnieren der MQTT-Topics beim MQTT-Broker;
- Verbindungsüberprüfung des Mobilfunkmoduls;
- In jedem Loop-Durchgang wird die MQTT-Verbindung zum Broker abgefragt und im Falle eines Verbindungsunterbruch wird diese wiederhergestellt und die Topics neu beim MQTT-Broker abonniert;
- Initialisieren der FSM über den Default-State in den NO_AS-State, in welchem sich die FSM befindet, wenn kein Alarm vorliegt;
- Sobald eine Alarmnachricht von der Funktion 1 empfangen wird, so wird in der FSM je nach Nachrichteninhalte in einen anderen State gewechselt. Dabei werden die Alarmmeldungen betrachtet, welche immer aus einer Zahl und einem Ausrufezeichen bestehen;
- Die ESP-readyLED wird als erstes kurz ausgeschaltet um den State-wechsel zu signalisieren;
- Die States 1-4 sind dann jeweils gleich aufgebaut und unterscheiden sich nur in der Alarmnachricht, welche als "String command" an die Funktion 3 übergeben wird;
- In der Funktion 3 wird eine Variable für das Schalten der Funktion 5 verändert, damit die FlasherLEDs dann ein- bzw ausgeschaltet werden können, um so lokal visuell zu alarmieren;
- Nach der visuellen Alarmierung erfolgt die akustische Alarmierung durch Schalten des Transistors an GPIO23 welche das Relais und somit das Sirenenmodul einschaltet;
- Nach der lokalen Alarmierung wird noch der "String command" über die AT-Befehle an den SMS Empfänger gesendet;
- Zum Schluss wird noch die Funktion 4 abgearbeitet, welche den Alarm-Reset-Taster (im Schaltplan K2) der vBMA Zentrale ausliest und den lokalen Alarm ausschaltet;
- Wenn der Alarm zurückgesetzt wurde oder kein Alarm vorliegt beginnt der Zentralencode wieder von vorne;
- Damit nur eine Alarmmeldung pro Ereignis abgesetzt wird, wechselt die FSM nach einem Durchgang in den State NO_AS und die ESP-readyLED wird wieder eingeschaltet. Wenn sich jedoch noch Rauch im Zimmer befindet, so wird erneut vom Sensor ein Alarm ausgegeben und dementsprechende von der vBMA Zentrale erneut alarmiert. Auf diese Weise kann so ein Fehlalarm nahezu ausgeschlossen werden.

3.3.4.3 Programmierung Node-Red (Betriebsdatenauswertung)

Die Betriebsdatenauswertung wird für jedes Topic/Thema in einem separaten Flow realisiert. Dies bedeutet, dass für jedes Zimmer/Raum, in welchem ein modifizierter Rauchsensor angebracht wird, 3 Topics ausgewertet werden müssen. Dies sind:

- Alarmauswertung
- Betriebsspannungsauswertung
- RSSI-Pegelauswertung

Diese 3 Flows werden dann für jedes Zimmer/Raum in Node-Red untereinander erstellt. Die abzusetzenden kritischen Betriebszustände werden auf den WhatsApp-Node geführt. Sobald dort eine Payload-Nachricht erscheint, wird diese an das Mobiltelefon in WhatsApp des Benutzers gesendet. Zudem wird das gesamte Dashboard auf den remote-access-Node gegeben, damit das Dashboard auch ausserhalb des lokalen Netzwerkes betrachtet werden kann.

Die in Tabelle 14 beschriebenen Nodes sind repräsentativ für alle anderen Nodes in jedem Zimmer/Raum und werden daher nur einmal beschrieben. Der gesamte Flow der vBMA ist im Anhang B.C1 abgelegt.

Node:	Node-Typ:	Konfiguration/Programmierung	Ausgang
	"mqtt in"	<ul style="list-style-type: none"> - Verbindung mit dem MQTT-Broker - Authentifizierung - Topic Subscription - Quality of Service - Angabe der Konnektivität 	<p>20.10.2023, 16:10:00 node: Brokerdaten Testsenso</p> <p>Testsensor_4 : msg.payload : string[10]</p> <p>"kein Alarm"</p> <p>20.10.2023, 16:10:00 node: Brokerdaten Testsensor 4</p> <p>Betriebsspannung 4 : msg.payload : string[28]</p> <p>▶ "Batteriespannung 4: 3.50 V"</p> <p>20.10.2023, 16:10:00 node: Brokerdaten Testsensor 4</p> <p>RSSI Pegel 4 : msg.payload : string[31]</p> <p>▶ "Wifi Empfangspegel 4: -68 dBm"</p>
	"debug"	<ul style="list-style-type: none"> - Wird für das Debugging verwendet - Gibt die msg.payload auf dem Monitor aus 	<p>20.10.2023, 16:10:00 node: Brokerdaten Testsensor 4</p> <p>RSSI Pegel 4 : msg.payload : string[31]</p> <p>▼ string[31]</p> <p>Wifi Empfangspegel 4: -68 dBm</p>
	"function"	<ul style="list-style-type: none"> - Extrahiert den reinen Zahlenwert aus dem String msg.payload - Datentrennung durch Sonderzeichenerkennung - Datentypwandlung string in float mit der Methode parseFloat() - Rückgabe des reinen Zahlenwertes der Betriebsspannung - Funktionscode ist in Anhang B.C2 ersichtlich 	<p>20.10.2023, 16:48:19 node: debug 1</p> <p>Betriebsspannung 4 : msg.payload : number</p> <p>3.51</p>
	"function"	<ul style="list-style-type: none"> - Gleiche Funktion wie beim Spannungsdatenfilter nur mit dem RSSI-Pegel - Funktionscode ist in Anhang B.C3 ersichtlich 	<p>20.10.2023, 16:54:34 node: debug 1</p> <p>RSSI Pegel 4 : msg.payload : number</p> <p>-67</p>
	"text"	<ul style="list-style-type: none"> - Gibt den erhaltenen String auf dem Dashboard im jeweiligen Raum/Zimmer aus 	<p>Status Rauchmelder 1</p> <p>Kommando vom Broker: kein Alarm</p>
	"chart"	<ul style="list-style-type: none"> - Gibt den Spannungsverlauf der letzten 4 Wochen auf dem Dashboard aus - x- und y-Achsen können im Kontextmenü des Nodes konfiguriert werden 	<p>Spannungsverlauf 1</p> <p>The chart shows a fluctuating voltage level around 4.0V over a period of approximately 8 days. The y-axis ranges from 2.9 to 4.5, and the x-axis shows dates from 13/10 to 21/10.</p>

Node:	Node-Typ:	Konfiguration/Programmierung	Ausgang
	"gauge"	<ul style="list-style-type: none"> - Gibt den letzten gemeldeten Spannungswert auf einem Kreisinstrument auf dem Dashboard an - Spannungsbereiche für die Farbgebung sind wie folgt konfiguriert: Grün = 3.3 bis 5V Orange = 3.0 bis 3.3V Rot = 0 bis 3V 	<p>Betriebsspannung 1</p>
	"function"	<ul style="list-style-type: none"> - Erkennt auf Grund von SOLL-Werten kritische Spannungsbereiche - Sobald die Betriebsspannung unter 3.2V fällt, wird folgender text auf die msg.payload ausgegeben: <i>"Unterspannung in Kinderzimmer 1"</i> 	<p>20.10.2023, 17:11:14 node: debug 1 Betriebsspannung 4 : msg.payload : string[27] "Unterspannung Testsensor 4!"</p>
	"chart"	<ul style="list-style-type: none"> - Gibt den RSSI-Pegelverlauf der letzten 4 Wochen auf dem Dashboard aus - x- und y-Achsen können im Kontextmenü des Nodes konfiguriert werden 	<p>WiFi Empfangspegel 1</p>
	"function"	<ul style="list-style-type: none"> - Erkennt auf Grund von SOLL-Werten kritische RSSI-Pegel - Sobald die RSSI-Pegel unter -85dBm fallen, wird folgender text auf die msg.payload ausgegeben: <i>"Signalpegel in Kinderzimmer 1 zu tief"</i> 	
	"send message"	<ul style="list-style-type: none"> - Überträgt die strings von msg.payload welche von den Unterspannungs- und zu tiefen RSSI Pegel- Detektoren erhaltenen Nachricht an ein Mobiltelefon auf WhatsApp - Die Nachricht wird von einem Chatbot (CallMeBot.com) übertragen 	
	"remote-access"	<ul style="list-style-type: none"> - Überträgt das Dashboard auf ein Mobiltelefon ausserhalb des lokalen Netzwerkes - Die App RemoteRed empfängt das Dashboard und kann dies unverändert darstellen - Die App ist kostenpflichtig - Es müssen sämtliche Angaben des Hosts gemacht werden (IP, Port und die Basis URL) - Die Konnektivität zum externen Server wird ebenfalls in Node-RED angezeigt 	

Tabelle 14: Beschrieb der verwendeten Nodes in Node-RED und deren Funktion

Das Dashboard, welches in Node-RED erstellt wird, kann in seiner Zusammenstellung noch beliebig verändert werden. Das Dashboard kann wie auf Abbildung 52 entweder lokal im Netzwerk auf dem Port 1880 oder auf der RemoteRed-App abgerufen werden.



Abbildung 52: Dashboard von Node-RED

3.3.5 Umsetzung

3.3.5.1 Printplattenherstellung

Das Vorgehen für die Printplattenherstellung war exakt das gleiche wie bei der Erstellung der Printplatte für den Sensor. Die Basis war ebenfalls der Schaltplan, welcher in Target 3001 gezeichnet wurde. Da die vBMA Zentrale aus unabhängigen Modulen besteht, musste genügend Platz für die Montage eingeplant werden. Das vorliegende Gehäuse verfügt über Einschubnuten. Wo eine 1,6mm dicke Printplatte eingeschoben werden kann. Die Abmessungen betragen 12x22cm. Diese Platzverhältnisse reichen aus, um alle Module inklusive der verwendeten Elektronik für die lokale Alarmauslösung sowie die Datenleitungen für das Mobilfunkmodul zu platzieren. Da sämtliche Module über die Printplatte an die gleiche externe Stromversorgung angeschlossen werden, musste die Stromleitenden Leiterbahnen etwas grösser dimensioniert werden. Alle Antennen der Module werden ausserhalb des Gehäuses angebracht, daher müssen diese nicht auf der Printplatte berücksichtigt werden.

Das definitive Printplattenlayout wurde ebenfalls über den Datelexport als Gerber-File realisiert und so bei jlcpcb.com bestellt.

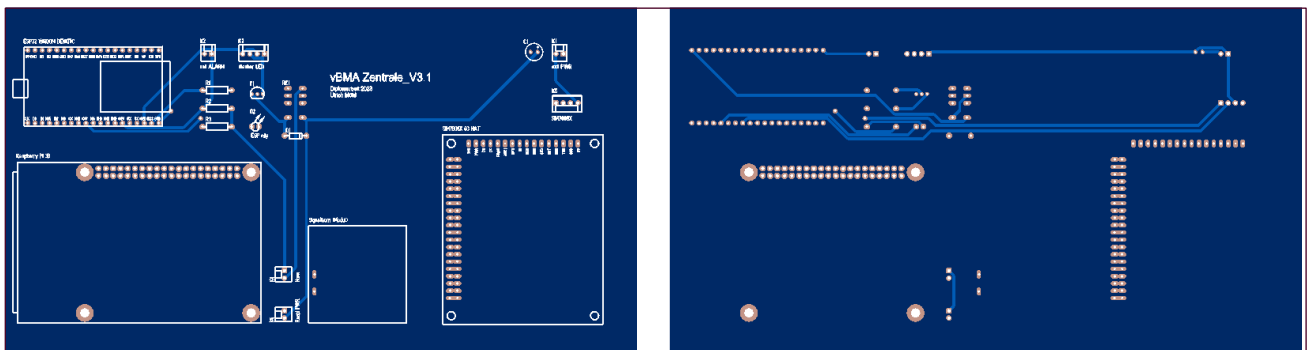


Abbildung 53: Printplattenlayout Vor- und Rückseite

Die Bestückung dieser Printplatten war einiges einfacher als die Bestückung der Printplatten des Sensors, da kein Mikrokontroller direkt aufgelötet werden muss. Sämtliche Footprints der Bauteile haben einwandfrei funktioniert und die Bauteile konnte sauber verlötet werden. Die Verkabelungen mit den Modulen konnte ebenfalls problemlos umgesetzt werden. Bei dem Mobilfunkmodul SIM7600X musste noch eine gerade Stiftleiste ausgelötet werden und durch eine abgewinkelte ersetzt werden, damit der Plexiglasdeckel eingeschoben werden konnte.

Direkt nach dem Bestücken der Printplatte konnte eine erste Funktionskontrolle durchgeführt und kleine Codekorrekturen vorgenommen werden. Die Konnektivität mit allen externen Antennen stand daher im Vordergrund. Die Verbindungen konnten jedoch über die externen Antennen hergestellt werden (blaue ESP-readyLED leuchtete). Das Sirenenmodul wurde bereits vorher mit dem Piezolautsprecher getestet.



Abbildung 54: vBMA Zentrale im Testbetrieb mit externen Antennen

3.3.5.2 Gehäuse

Das Gehäuse besteht im Wesentlichen aus einem Aluminium-Strangpressprofil. Damit sämtliche Antennen und Stromanschluss aus dem Metallgehäuse geführt werden können, mussten die entsprechenden Aussparungen mittels CNC-Fräse ausgefräst werden. Dafür wurden die seitlichen Abdeckungen in Autodesk Inventor gezeichnet (siehe Abbildung 54) und in einem CAM-Programm der Maschinencode generiert, um diesen dann auf einer CNC-Fräse zu fertigen.

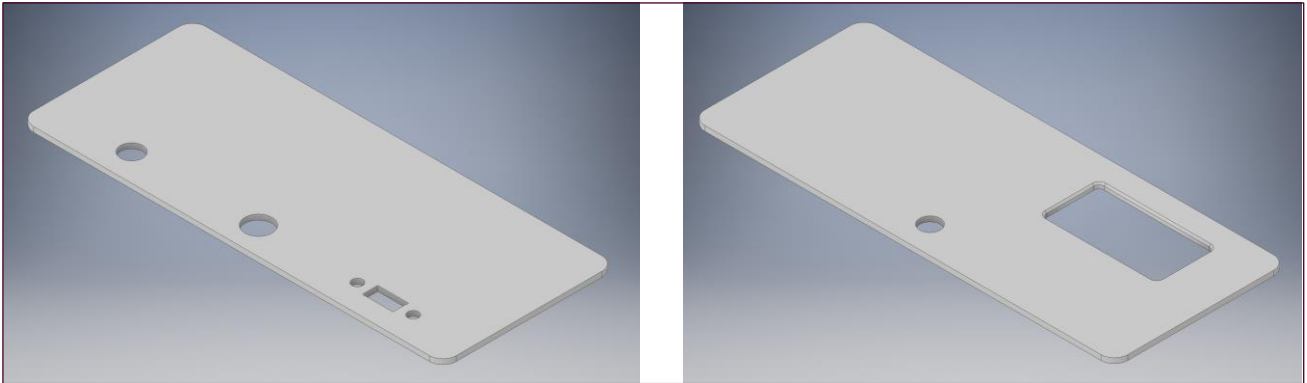


Abbildung 55: Zeichnungen der Seitenabdeckungen links und rechts

Damit später für die Präsentation die Elektronik der vBMA Zentrale ersichtlich ist, wurde ein Einschubdeckel aus Plexiglas anstelle des Aluminiumdeckels per Hand gefertigt. Die FlasherLEDs und die Alarmsirene mussten so noch in den Plexiglasdeckel eingearbeitet werden. Die fertige und voll funktionsfähige Zentrale ist in Abbildung 56 abgebildet.

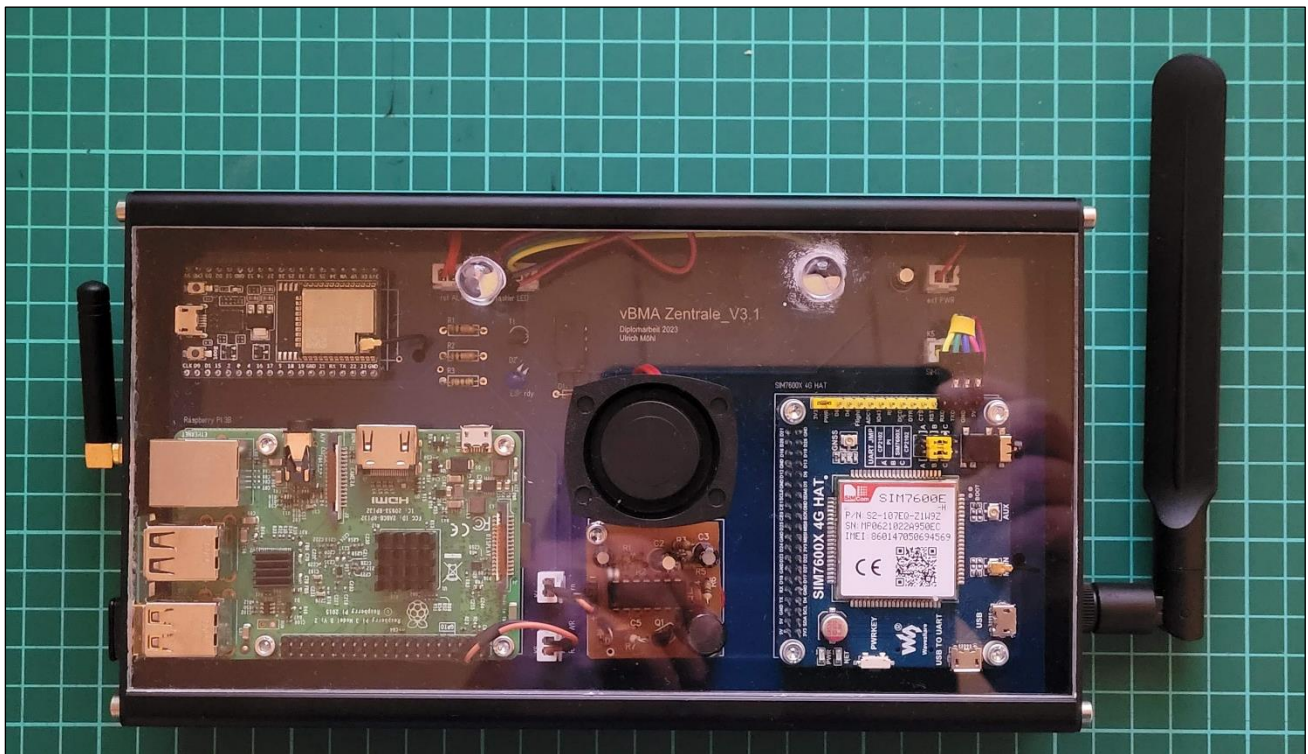


Abbildung 56: fertiggestellte vBMA Zentrale mit Plexiglasabdeckung

4 Abschluss

4.1 Wirtschaftlichkeit

Für die vorliegende Diplomarbeit wird ein Funktionsmuster und ein Prototyp erstellt, bestehend aus einer Zentrale und 3 Sensoren. Es werden die Entwicklungskosten mit dem Funktionsmuster und die Realisierungskosten des Prototyps berechnet.

4.1.1 Effektive Kosten

Die Hauptkosten dieses Projektes setzen sich aus Personal und Materialkosten zusammen. Eine untergeordnete Rolle spielen die Kosten aus benötigter Infrastruktur und Werkzeugen, da diese bereits vorhanden waren und in den Kosten nur mit pauschalen aufgeführt werden können. Die Kosten wurden anhand der Arbeitspakete bemessen. Die Erfassung und Auswertung erfolgte auf separaten Listen, welche aus übersichtlichkeitsgründen nicht dieser Diplomarbeit angefügt wurden.

4.1.1.1 Personalkosten

Arbeitspaket	Thema	Anzahl (h)	Satz	Personalkosten
AP0	Projektinitialisierung	3	CHF 130.00	CHF 390.00
AP1	Konzeption	21	CHF 130.00	CHF 2'730.00
AP2	Entwicklung	104	CHF 130.00	CHF 13'520.00
AP3	Realisierung	61	CHF 130.00	CHF 7'930.00
AP4	Dokumentation	54	CHF 130.00	CHF 7'020.00
Total Personalkosten		243		CHF 31'590.00

Tabelle 15: Aufstellung Personalkosten

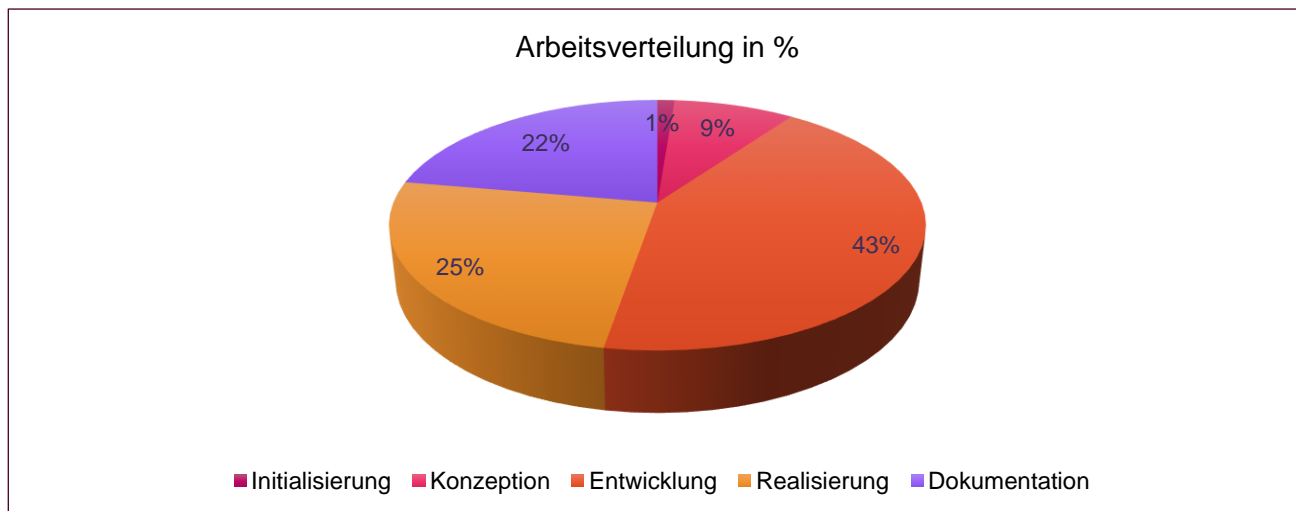


Abbildung 57: Grafik der Arbeitsverteilung in %

4.1.1.2 Sachkosten

Arbeitspaket	Thema	Artikel	Anzahl	Stückpreis	Sachkosten
AP1	Projektinitialisierung	Temperatursensoren DS18B20	3	CHF 2.00	CHF 6.00
		Flammensensor IR	3	CHF 3.50	CHF 10.50
		Gassensor	3	CHF 3.90	CHF 11.70
		ESP8266 12F	1	CHF 6.80	CHF 6.80
		Arduino Nano 33 IoT	1	CHF 29.25	CHF 29.25
		Breakout Board ESP8266 12F	4	CHF 1.20	CHF 4.80
		ESP32S	1	CHF 9.20	CHF 9.20
		ESP32S Breakout Board	1	CHF 1.75	CHF 1.75
AP2	Entwicklung	Prototype PCB Arduino UNO	2	CHF 5.50	CHF 11.00
		Arduino Uno Projekthalter	3	CHF 18.95	CHF 56.85
		ESP32 NodeMCU	2	CHF 11.95	CHF 23.90
		ESP38266 DevKitC	1	CHF 14.95	CHF 14.95
		ESP8266 12F	3	CHF 6.80	CHF 20.40
		FTDI Programmer	2	CHF 9.90	CHF 19.80
		ESP8266 12F Programmer	2	CHF 7.50	CHF 15.00
		ESP8266 NodeMCU V2	2	CHF 29.90	CHF 59.80
		HT7333 LDO Regler	10	CHF 0.40	CHF 4.00
		1000uF Tantal Kondensator	5	CHF 3.60	CHF 18.00
		CD4043 RS-Latch	10	CHF 0.75	CHF 7.50
		MC14044 Quad RS-Latch SMD	10	CHF 2.00	CHF 20.00
		Lithium Batterie 3,6V	3	CHF 14.00	CHF 42.00
		FET Transistor 2N7000	25	CHF 0.16	CHF 4.00
		Power Adapter für Breadboard	2	CHF 3.90	CHF 7.80
		Raspberry Pi 3B	2	CHF 59.00	CHF 118.00
		SIM-Karte von SALT inkl. Guthaben	1	CHF 40.00	CHF 40.00
		Waveshare SIM7600 4G Modul	2	CHF 79.90	CHF 159.80
AP3	Realisierung	Rauchwarnmelder Genius H	3	CHF 28.95	CHF 86.85
		ESP8266 12F	10	CHF 1.20	CHF 12.00
		LIR2450 Lipo Akku	4	CHF 4.85	CHF 19.40
		Batteriehalter CR2450	4	CHF 7.80	CHF 31.20
		LiPo Knopfzellenladegerät	1	CHF 13.95	CHF 13.95
		Sirenenmodul Tru Components	1	CHF 10.95	CHF 10.95
		Drucktaster	4	CHF 6.40	CHF 25.60
		Buchsenleiste	5	CHF 0.80	CHF 4.00
		Minischalter	10	CHF 1.94	CHF 19.40
		IC Sockel DIL16	10	CHF 1.55	CHF 15.50
		Taster THT	10	CHF 0.60	CHF 6.00

Arbeitspaket	Thema	Artikel	Anzahl	Stückpreis	Sachkosten
AP3	Realisierung	10mm LED Weiss Superhell	5	CHF 1.40	CHF 7.00
		10mm LED Fassung	5	CHF 0.30	CHF 1.50
		ESP32 DevKitC 32UE	2	CHF 8.75	CHF 17.50
		USB Micro B Panel	2	CHF 2.30	CHF 4.60
		5V Omron Relais	3	CHF 5.90	CHF 17.70
		WLAN Kit SMA/IPEX	2	CHF 8.75	CHF 17.50
		USB WLAN Adapter	1	CHF 14.50	CHF 14.50
		Printplatten Sensor	5	CHF 3.00	CHF 15.00
		Printplatten Zentrale	5	CHF 4.00	CHF 20.00
		Extruded Enclosure	1	CHF 39.90	CHF 39.90
		Diverse Widerstände	P	CHF 5.00	CHF 5.00
		Diverse Halbleiter	P	CHF 5.00	CHF 5.00
		Diverse Kondensatoren	P	CHF 2.00	CHF 2.00
		Total Sachkosten			

Tabelle 16: Aufstellung Sachkosten

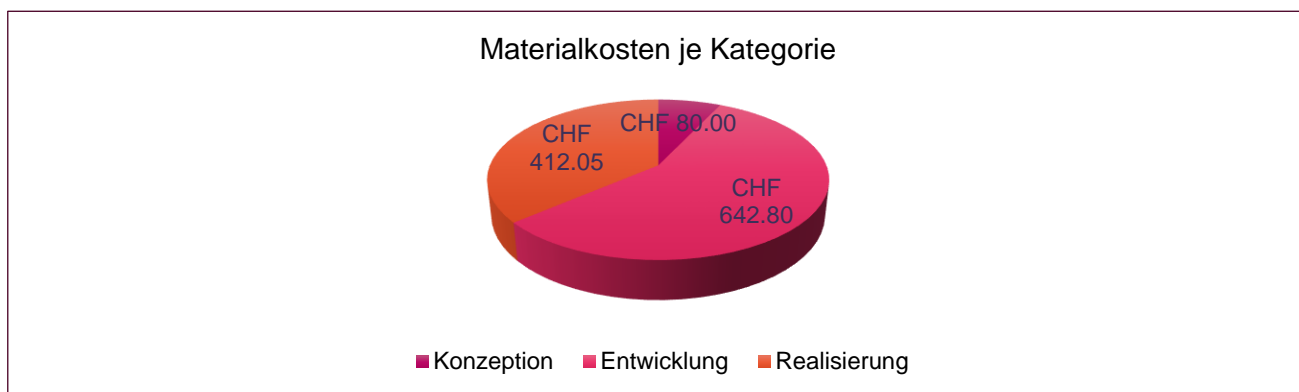


Abbildung 58: Grafik der Materialkostenverteilung je Arbeitspaket

4.1.1.3 Infrastrukturkosten

Arbeitspaket	Infrastruktur	Anzahl	Stückpreis	Sachkosten
AP0 – AP4	Mess- und Prüfgeräte (Multimeter, Oszilloskop)	P	CHF 10.00	CHF 10.00
	IT (Hard- Software, Lizenzen)	P	CHF 10.00	CHF 10.00
	Fertigungsmaschinen und Geräte	P	CHF 10.00	CHF 10.00
Total Infrastrukturkosten				CHF 30.00

Tabelle 17: Aufstellung Infrastrukturkosten

4.1.1.4 Gesamtkosten

Positionen	Kosten
Personalkosten	CHF 31'590.00
Sachkosten	CHF 1'134.85
Infrastrukturkosten	CHF 30.00
Total	CHF 32'754.85

Tabelle 18: Aufstellung Gesamtkosten

4.2 Kontrolle der Zielerreichung

Damit die Zielerreichung quantitativ gemessen werden kann müssen sämtliche in Kapitel 1 erwähnten Erfolgskriterien und Anforderungen aus dem Pflichtenheft betrachtet werden. Zudem spielte der Projektplan während der gesamten Erarbeitung dieser Diplomarbeit eine zentrale Rolle und half, den Projektfortschritt in Bezug auf den Zeitplan zu messen. Abweichungen vom Zeitplan wurden somit rasch erkannt und eine Kurskorrektur konnte rasch mit zusätzlichen Arbeitsstunden erbracht werden.

4.2.1 Kontrolle der Zielerreichung der Erfolgskriterien

Ziel	Art	Grad	Bemerkungen
Am Präsentationstag liegt minimal ein funktionsfähiges Funktionsmodell von einem Sensor und einer Zentrale oder maximal ein funktionsfähiges Prototypsystem mit 3 Sensoren und einer Zentrale vor.	MUSS	100%	Es wird zusätzlich zu den 3 Sensoren das Funktionsmuster noch eingebunden
Das System erfüllt alle Kriterien, welche im Pflichtenheft aufgeführt sind.	MUSS	97%	Siehe Kapitel 4.2.2
Der Zustand des Sensoren-Systems kann über ein Dashboard überwacht werden.	MUSS	100%	
In der DA werden alle effektiven Kosten einer Realisierung aufgeführt inklusive der Experimentierkosten.	KANN	100%	
Die DA wird fristgerecht bis am 30. Oktober 2023, 1700 Uhr abgegeben.	MUSS	100%	
Eine Microwebseite wird auf dem DA-Portal der TEKO bis am 09. November 2023, 2345 Uhr erstellt und veröffentlicht.	MUSS	100%	
Die DA am 11. November 2023 an der TEKO in Bern präsentiert wird	MUSS	100%	
Der Zeitaufwand der DA im Bereich von 150-250 Stunden liegt	MUSS	100%	
Wenn durch die DA im Schadenfall ein grosses Schadenausmass verhindert werden kann.	MUSS	NB	Da dieses Kriterium erst langfristig und in einem Schadenfall beurteilt werden kann, ist dieses Kriterium von der Bewertung ausgeschlossen.
Zielerreichungsgrad		99.6%	

Tabelle 19: Zielerreichung der Erfolgskriterien

4.2.2 Kontrolle der Zielerreichung des Pflichtenheftes

4.2.2.1 Anforderungen Sensor

Anfo	Beschreibung	Art	Grad	Bemerkungen
SEN_01	Der Sensor wird auf einem von der Vereinigung kantonaler Feuerversicherungen (VKF) zertifizierten Sensor/Rauchmelder aufgebaut;	MUSS	100%	Zertifikat ist im Anhang C.A angefügt und ist Teil der Lösung
SEN_02F	Die Grundfunktion des zertifizierten Sensors muss zwingend erhalten bleiben (lokaler akustischer / optischer Alarm)	MUSS	100%	Es wird ausschliesslich ein Alarmimpuls abgegriffen
SEN_03F	Im Brandfall werden so lange wie möglich Brandalarme via W-LAN abgesetzt.	MUSS	100%	
SEN_04F	Bei einem Fehlalarm muss der Sensor am Objekt zu Kontrollzwecken zurückgesetzt werden.	KANN	100%	
SEN_05F	Die Stromversorgung des Mikrokontrollers muss autonom aufgebaut werden, damit die Grundfunktion des Rauchmelders nicht beeinträchtigt wird.	MUSS	100%	Mit dem LIR2450 Akku sichergestellt
SEN_06F	Die minimale Betriebsdauer eines Sensors beträgt 1 Monat, bevor die Batterie des Mikrokontrollers getauscht oder aufgeladen werden muss	KANN	60%	Berechnungen haben eine Dauer von 21 Tagen ergeben. Dauertest steht noch aus!
SEN_07F	Betriebsparameter des Mikrokontrollers werden periodisch alle 2-4 Stunden der Zentrale zur Auswertung übertragen.	MUSS	100%	
SEN_08F	Die Betriebsparameter werden über das MQTT (Message Queue Telemetry Transport) – Protokoll als MQTT-Publisher-Message übertragen.	MUSS	100%	
SEN_09	Das Funktionsmuster des Sensormikrokontrollers wird auf einem Breadboard aufgebaut.	MUSS	100%	
SEN_10	Die Sensorprototypen werden auf einer eigens dafür entwickelten Platine aufgebaut.	KANN	100%	
SEN_11	Der Mikrokontroller und die zusätzlich dafür benötigte Elektronik werden in einem zusätzlichen Gehäuse untergebracht.	KANN	100%	
SEN_12	Der Sensor kann in allen Wohnräumlichkeiten sowie der Garage uneingeschränkt eingesetzt werden (ausgenommen Badezimmer)	KANN	100%	
Zielerreichungsgrad			96.66%	Teil der Erfolgskriterien

Tabelle 20: Zielerreichung der Anforderungen an den Sensor

4.2.2.2 Anforderungen Zentrale

Anfo	Beschreibung	Art	Grad	Bemerkungen
ZEN_01	In der Zentrale ist der MQTT-Broker (Datenverteiler) einerseits für die Alarmweitergabe und andererseits für die Betriebsdatenauswertung untergebracht.	MUSS	100%	
ZEN_02	Die Betriebsdatenauswertung wird zwecks Datenverfügbarkeit zentral auf einem Mikrocomputer realisiert und ist Bestandteil der Zentrale.	MUSS	100%	
ZEN_03F	Die Auswertung der Betriebsdaten wird auf einem übersichtlichen, lokal wie dezentral verfügbaren Dashboard visualisiert.	MUSS	100%	Lokal via IP / Dezentral via RemoteRED App
ZEN_04F	Der Zustand des Mikrokontrollers in den Sensoren welcher Alarme weitergibt, muss auch ausserhalb des W-LANS überwacht werden können.	MUSS	100%	RemoteRED App
ZEN_05F	Kritische Betriebsparameter des Mikrokontrollers der Sensoren werden über einen geeigneten Kommunikationskanal zeitnah dem Benutzer auf sein Mobiltelefon übermittelt.	KANN	100%	Aus Node-RED via WhatsApp
ZEN_06	Die Auswertung der Alarmnachrichten (MQTT-Subscriber-Message) wird aus technischen Gründen auf einem separaten Mikrokontroller realisiert, welcher sämtliche Alarmierungen steuert.	MUSS	100%	
ZEN_07F	Alarme werden akustisch wie auch optisch für die erweiterte lokale Wahrnehmung wiedergegeben.	MUSS	60%	Realisiert jedoch nicht ganz zufriedenstellend
ZEN_08F	Ein ausgelöster Alarm kann nur an der Zentrale zurückgesetzt werden.	KANN	100%	
ZEN_09F	Nur die Alarme werden über einen kostenpflichtigen Dienstleister über ein Mobilfunksystem (3G/4G/5G) zum Benutzer als SMS übertragen.	MUSS	100%	SALT.ch
ZEN_10F	In der Alarmnachricht ist ersichtlich in welchem Raum der Alarm ausgelöst wurde	MUSS	100%	
ZEN_11	Die Stromversorgung der Zentrale erfolgt über ein externes Netzteil auf 5VDC	KANN	100%	
ZEN_12	Das Funktionsmuster der Zentrale wird auf einem Breadboard aufgebaut, welches an das Mobilfunkmodul angeschlossen wird. Der MQTT-Broker wird beim Funktionsmuster autonom betrieben.	MUSS	100%	
ZEN_13	Der Prototyp der Zentrale wird auf einer eigens dafür entwickelten Platine aufgebaut.	KANN	100%	
ZEN_14	Der Mikrocomputer, der Mikrokontroller sowie das Sirenenmodul und die zusätzlich dafür benötigte Elektronik werden in einem Gehäuse untergebracht. Antennen werden aus dem Gehäuse herausgeführt.	KANN	100%	
ZEN_15	Es werden verschieden Kommunikationstechnologien verwendet	MUSS	100%	
Zielerreichungsgrad			97.3%	Teil der Erfolgskriterien

Tabelle 21: Zielerreichung der Anforderungen an die Zentrale

4.3 Fazit / Lessons Learnt

Die Vorbereitung und Erarbeitung der Diplomarbeit hielt einige Herausforderungen bereit, welche angegangen werden mussten. Meine Erfahrungen und Erkenntnisse habe ich während der Erarbeitung der Diplomarbeit in einem Logbuch, welches jeden einzelnen Entwicklungsschritt beinhaltet, festgehalten. Da nicht sämtliche Erkenntnisse in diesem Fazit erläutert werden können, werden diese zu Themenbereichen zusammengefasst. Aus der Ganzen Erarbeitung der Diplomarbeit nehme ich die folgenden 4 Punkte/Lessons Learnt mit:

Herangehensweise

Bei der Themeneingabe versuchte ich mir bereits ein mögliches Pflichtenheft der vBMA vorzustellen. Mit der Zeit entwickelten sich immer mehr Ideen und Funktionalitäten, welche ich noch realisiert haben möchte. Da der Start der Diplomarbeit noch nicht in greifbarer Nähe war und wir bereits Material bestellen konnten, musste ich mir vorstellen können, wie eine mögliche Realisierung aussehen könnte, ohne dabei mit der Entwicklung und somit mit der Diplomarbeit schon zu beginnen. Dabei hat mir in dieser Vorphase eine klare Struktur gefehlt, wie das Vorgehen in dieser Phase sein könnte, ohne dass zu viel oder das falsche Material bestellt wird und wie ich mit diesem Material nach der Lieferung umgehen soll. Für eine nächste solche Arbeit würde ich eine Vorstudie mit genau definierten Arbeitspaketen, welche nicht die Entwicklung und Realisierung der Diplomarbeit beinhaltet beantragen, damit Probleme mit der Hardware frühzeitig erkannt werden können und dementsprechend im Zeitplan genügend Zeit eingeplant werden kann.

Entwicklung

Die Entwicklung der vBMA beinhaltete die meisten Hürden und Stolperfallen der Diplomarbeit. Dies war einerseits der Qualität von Bauteilen (Mikrokontroller, Breadboards, Batterien etc) und andererseits den mangelnden Zeitverhältnissen geschuldet, dass Probleme nicht endlos analysiert und herausgemessen werden konnten. Um den Faktor Zeit im Griff zu behalten, half mir der zu Beginn erstellte Zeitplan enorm. Die technischen Probleme konnte ich mehrmals mit der Konsultation der Datenblätter lösen und die Schaltungen dementsprechend anpassen. Daher spielten die Datenblätter bei der Bemessung der Stromversorgung, der Beschaltung, der Pegel und Potentiale der verschiedenen Bauteile und Stufen eine sehr wichtige Rolle. Die Schaltung, zum Beispiel des Sensors, habe ich so mehrere Male umgezeichnet, was das Verständnis enorm förderte.

Die Codeentwicklung verlief daher einiges angenehmer, da diese die gebaute Hardware nur noch auslesen musste. Dabei fielen auch wieder die Fertigungsqualitäten der Bauteile auf, da die herausgemessenen Werte von Sensor zu Sensor manchmal nicht annähernd beieinander lagen trotz exakt gleichen Bedingungen. Dies konnte ich nur mit einer Codekorrektur oder einem Austausch eines Mikrokontrollers erreichen. Die Ansteuerung der externen Module verschlang beim Mobilfunkmodul etwas viel Zeit, da die AT-Befehle erst über die Eingabe im seriellen Monitor der Arduino IDE getestet wurden. Dabei können nur `.print` Befehle auf die UART-Schnittstelle geschrieben werden und keine `.write` Anweisungen. So war es nicht möglich eine SMS zu versenden, bis die `.write` Anweisung im Sourcecode eingegeben wurde (für `Ctrl+Z` resp `0x1A`).

Fertigung

Die Fertigung der Prototypen (Sensor und Zentrale) bildeten den schönen Teil der Arbeit, welcher mir leicht von der Hand lief. Die Printplattenherstellung mit Target 3001 musste erst etwas erprobt werden. Die Routing-Algorithmen lieferten nicht auf Anhieb brauchbare Resultate, so wurden einige Verbindungen erst nach ausführlichen Routingkontrollen erstellt. Die Bestellung mit den Gerber-Files verlief sehr rasch und zeitnah. Bei der Bestückung der gelieferten Printplatten stiess ich wieder auf die gleichen Probleme beim Verlöten der Mikrokontroller wie in der Entwicklungsphase. Da das Problem bereits bekannt war, konnten rasch gute Printplatten erstellt werden. Trotzdem mussten wieder Mikrokontroller mit einem Heissluftflüßgerät ersetzt werden.

Die Printplatten konnten bereits in einer ersten Version verwendet werden. Für einen ersten Versuch reichten diese erste Version völlig aus müssten nun jedoch noch etwas weiterentwickelt werden, damit die freien Anschlüsse der verschiedenen Bauteile noch klar zugewiesene Zustände erhalten.

Die Zentrale könnte in den Abmessungen noch etwas redimensioniert werden, wenn die Funktion des Alarm-Mikrokontrollers auf dem Raspberry Pi in Python implementiert würde und das Mobilfunkmodul als HAT (Hardware Attached on Top) auf dem Raspberry Pi aufgesteckt und dementsprechend programmiert würde. Dafür fehlte es leider an Zeit und Python-Kenntnissen um dies so im geforderten Zeitkorsett umzusetzen.

Zeitverhältnisse

Über die ganze Diplomarbeit gesehen waren die Zeitverhältnisse der grösste Kontrahent. Die Schwierigkeit die Entwicklung, Realisierung und Dokumentation in der geforderten Zeit umzusetzen war ein stetiger Stressor. Dabei Beruf und Familie mit 2 Kindern zu koordinieren damit niemand zu kurz kommt war in den letzten 6 Wochen eine sehr anspruchsvolle Aufgabe. Daher würde ich beim nächsten Mal den Umfang der Arbeit etwas kleiner definieren.

Zum Schluss

Mir persönlich war es wichtig, verschiedene Kommunikationstechnologien zu nutzen und am Ende etwas Fertiges in den Händen zu halten. Diese persönlichen Anforderungen wollte ich nicht in dem Pflichtenheft erfassen, da dies persönliche Anforderungen waren und für mich zeitweise über den anderen Anforderungen standen. Daher freut es mich umso mehr diese persönlichen Ziele ebenfalls erreicht zu haben und mir so eine Basis geschaffen zu haben, um die vBMA stressfrei noch zu optimieren, um dann im eigenen Haushalt einzusetzen.

So ist es auch gut möglich, dass die anderen Projektarbeiten, welche während dem Studium erarbeitet wurden, noch eine Realisierung erfahren und mit der vBMA verbunden werden.

Anhang A Verzeichnisse

Anhang A.A Literaturverzeichnis

- [27] Agnihotri, N. (2023). *www.engineersgarage.com*. Von <https://www.engineersgarage.com/at-commands-gsm-at-command-set/> abgerufen
- [5] Arduino.cc. (2023). *https://docs.arduino.cc/*. Von <https://docs.arduino.cc/hardware/nano-33-iot> abgerufen
- [4] Beuth/Hanebuth/Kurz. (1996). *Nachrichtentechnik*. Würzburg: Vogel Fachbuch.
- [11] Brandes, U. (2023). *Mikrocontroller ESP32*. Bonn: Rheiwerk Verlag.
- [26] Brühlmann, T. (2021). *Heimautomation mit Arduino, Raspberry Pi und ESP8266*. Frechen: mitp Verlags GmbH + Co.
- [24] Conrad Electronic SE. (2023). *www.conrad.ch*. Von <https://www.conrad.de/de/p/tru-components-sirenenbaustein-baustein-6-v-dc-9-v-dc-12-v-dc-2304642.html> abgerufen
- [22] Dembowski, K. (2019). *Raspberry Pi "das technische Handbuch"*. Hamburg: Springer Fachmedien Wiesbaden GmbH.
- [18] DigiKey.ch. (2023). *www.digikey.ch*. Von https://www.digikey.ch/de/products/detail/analog-devices-inc-maxim-integrated/DS18B20/420071?utm_adgroup=Analog%20Devices&utm_source=google&utm_medium=cpc&utm_campaign=Dynamic%20Search_DE_RLSA_Supplier_Site%20Visitor&utm_term=&productid=&utm_content=Analo abgerufen
- [17] Elektronik Kompendium. (2023). *www.elektronik-kompendium.de*. Von https://www.elektronik-kompendium.de/sites/praxis/bauteil_ds18b20.htm#:~:text=Die%20Kommunikation%20mit%20dem%20DS18B20,Bus%20explizit%20abfragen%20zu%20k%C3%B6nnen. abgerufen
- [12] Espressif Systems. (2023). *www.espressif.com*. Von https://www.espressif.com/sites/default/files/documentation/esp32-s2-solo_esp32-s2-solo-u_datasheet_en.pdf abgerufen
- [25] Espressif Systems Co. (2023). *www.espressif.com*. Von https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf abgerufen
- [8] Espressiv Systems. (2023). *www.espressif.com*. Von <https://www.espressif.com/en/products/modules> abgerufen
- [20] Europa Lehrmittel. (2020). *Fachkunde Elektrotechnik*. Haan-Gruiten: Europa Lehrmittel.
- [13] Minol Messtechnik GmbH. (2023). *www.minol.de*. Von <https://www.minol.de/faq-rauchmelder.html> abgerufen
- [19] Random Nerd Tutorials. (2023). *randomnerdtutorials.com*. Von <https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/> abgerufen
- [14] Rauchmeldertest.net. (2022). *www.rauchmeldertest.net*. Von <https://www.rauchmeldertest.net/hekatron-genius-plus-rauchmelder-test/> abgerufen
- [3] Sengpiel, A. (2010). *http://www.sengpielaudio.com*. Von <http://www.sengpielaudio.com/Rechner-entfernung.htm> abgerufen
- [28] SIMCOM. (2021). *www.waveshare.com*. Von https://www.waveshare.com/w/upload/a/af/SIM7500_SIM7600_Series_AT_Command_Manual_V3.00.pdf abgerufen
- [6] Staquet, O. (17. Juli 2021). *github.com*. Von <https://github.com/ostaquet/Arduino-Nano-33-IoT-Ultimate-Guide/blob/master/SavePowerSleeping.md> abgerufen
- [16] Switch Electronics. (2023). *https://www.switchelectronics.co.uk*. Von <https://www.switchelectronics.co.uk/products/ir-infrared-flame-detector-sensor-module> abgerufen
- [1] TEKO Schweizerische Fachschule. (2023). *Richtlinien Diplomarbeit Herbst 2023*. Bern.

- [9][21] Trojan, W. (2017). *Das MQTT-Praxisbuch*. Aachen: Elektor Verlag GmbH.
- [23] Tutorials for Raspberry Pi. (2023). <https://tutorials-raspberrypi.de>. Von <https://tutorials-raspberrypi.de/raspberry-pi-remote-desktop-verbinding/> abgerufen
- [2] Vereinigung Kantonaler Feuerversicherungen VKF. (01. 01 2015). <https://www.bsvonline.ch/de>. Von <https://www.bsvonline.ch/de/vorschriften/> abgerufen
- [29] Waveshare Electronics. (2023). www.waveshare.com. Von https://www.waveshare.com/wiki/SIM7600E-H_4G_HAT abgerufen
- [7] www.joy-it.net. (2023). <https://joy-it.net>. Von <https://joy-it.net/files/files/Produkte/SBC-NodeMCU/SBC-NodeMCU%20Kurzanleitung.pdf> abgerufen
- [10] www.joy-it.net. (2023). www.joy-it.net. Von <https://joy-it.net/files/files/Produkte/SBC-NodeMCU-ESP32/SBC-NodeMCU-ESP32-Anleitung-2021-06-29.pdf> abgerufen
- [15] www.joy-it.net. (2023). www.joy-it.net. Von <https://joy-it.net/de/products/SEN-MQ2> abgerufen

Anhang A.B Abbildungsverzeichnis

Abbildung 1: Rauchmelder HEKATRON Genius H	13
Abbildung 2: primitiver Schallmessaufbau.....	14
Abbildung 3: Situationsplan Baumgartiweg 1 in 3752 Wimmis	14
Abbildung 4: Prinzipschema vBMA / SOLL Zustand	15
Abbildung 5: Arduino Nano 33 IoT	16
Abbildung 6: ESP8266 NodeMCU V2	17
Abbildung 7: ESP8266 12F mit Breakout-Board	18
Abbildung 8: ESP32 NodeMCU mit 30 Pins.....	18
Abbildung 9: ESP32S mit Breakout-Board.....	19
Abbildung 10: Rauchmelder im Normalzustand	20
Abbildung 11: Rauchmelder bei Brandrauch in der Luft	20
Abbildung 12: Rauchmelder Variante 1.....	21
Abbildung 13: Rauchmelder Variante 2.....	22
Abbildung 14: Rauchmelder Variante 3.....	22
Abbildung 15: Joy-It SEN-MQ2 Gassensor	23
Abbildung 16: Purecrea IR Infrared Flame Sensor Module	23
Abbildung 17: Dallas DS18B20 Temperatur Sensor	24
Abbildung 18: Systemaufbau vBMA.....	27
Abbildung 19: Funktionsmodell eines Sensors.....	29
Abbildung 20: Funktionsmodell der Zentrale (reduziert)	29
Abbildung 21: ESP8266 12F Pin-Belegungsplan	30
Abbildung 22: Schaltplan / technischer Aufbau eines Sensors (aus Target 3001).....	31
Abbildung 23: Schaltplan / technischer Aufbau Stromversorgung Sensor.....	32
Abbildung 24: impulsförmige Stromspitzen mit und ohne C2	32
Abbildung 25: Messaufbau Ruhestrommessung (Energiesparmodus)	33
Abbildung 26: Schaltplan / technische Beschaltung des Mikrokontrollers	34
Abbildung 27: Spannungsteiler für die Betriebsdatenerfassung	34
Abbildung 28: Alarmsignal des Rauchmelders.....	35
Abbildung 29: Alarmauslösung / Ansteuerung des MOSFET	36
Abbildung 30: Alarmspeicherung in einem RS-Flipflop CD4043BE	36
Abbildung 31: Ablaufschema des Sensorcodes	38
Abbildung 32: Printplattenlayout mit Vor- und Rückseite.....	39
Abbildung 33: Programmiersockel mit und ohne Mikrokontroller.....	39
Abbildung 34: Zeichnung des Gehäuses für die Printplatte	40
Abbildung 35: Fertiger Rauchmelder mit Printplatte	40
Abbildung 36: Ersatzmikrokontroller ESP32UE DevKitC V4 mit IPEX-Antennensockel.....	41
Abbildung 37: Schaltplan / technischer Aufbau der vBMA Zentrale (aus Target 3001).....	42
Abbildung 38: Schaltplan / technischer Aufbau der Stromversorgung	43
Abbildung 39: Konfiguration des externen W-LAN Adapters auf dem Raspberry Pi 3B.....	44
Abbildung 40: Schaltplan / technische Einbindung des Raspberry Pi 3B	44
Abbildung 41: Schaltplan / technische Einbindung des ESP32 Alarm-Mikrokontrollers.....	45
Abbildung 42: Anschlussplan eines ESP32 DevkitC 32UE V4	45
Abbildung 43: Waveshare SIM7600X 4G HAT Mobilfunkmodul	46
Abbildung 44: Beschaltung Mobilfunkmodul SIM7600X 4G HAT	46

Abbildung 45: Sirenenmodul von Tru Components	47
Abbildung 46: Schaltplan der Ansteuerung des Sirenenmoduls	47
Abbildung 47: Schaltplan der Ansteuerung der superhellen LEDs	48
Abbildung 48: Schaltplan der für den Alarm-Reset.....	48
Abbildung 49: Installations- und Konfigurationsschritte für den MQTT-Broker.....	49
Abbildung 50: Initial- und Beispielnachricht der CallMeBot API.....	50
Abbildung 51: Sourcecode des Alarm-Mikrokontrollers	52
Abbildung 52: Dashboard von Node-RED.....	56
Abbildung 53: Printplattenlayout Vor- und Rückseite	57
Abbildung 54: vBMA Zentrale im Testbetrieb mit externen Antennen.....	57
Abbildung 55: Zeichnungen der Seitenabdeckungen links und rechts.....	58
Abbildung 56: fertiggestellte vBMA Zentrale mit Plexiglasabdeckung	58
Abbildung 57: Grafik der Arbeitsverteilung in %	59
Abbildung 58: Grafik der Materialkostenverteilung je Arbeitspaket.....	61

Anhang A.C Tabellenverzeichnis

Tabelle 1: Themenwahl.....	5
Tabelle 2: Endergebnisse	8
Tabelle 3: Erfolgskriterien	9
Tabelle 4: Pflichtenheft / Anforderungen Sensor	9
Tabelle 5: Pflichtenheft / Anforderungen Zentrale	10
Tabelle 6: Hilfsmittel	11
Tabelle 7: Vor-/Nachteile der Sensorvariante 1.....	21
Tabelle 8: Vor-/Nachteile der Sensorvariante 2.....	22
Tabelle 9: Vor-/Nachteile der Sensorvariante 3.....	24
Tabelle 10: Wahl der Mikrokontroller.....	25
Tabelle 11: Matrix für die Variantenbewertung.....	26
Tabelle 12: Stromkonsum der einzelnen Module	43
Tabelle 13: verwendete AT-Befehle und Eingaben	51
Tabelle 14: Beschrieb der verwendeten Nodes in Node-RED und deren Funktion.....	55
Tabelle 15: Aufstellung Personalkosten	59
Tabelle 16: Aufstellung Sachkosten	61
Tabelle 17: Aufstellung Infrastrukturkosten	61
Tabelle 18: Aufstellung Gesamtkosten.....	61
Tabelle 19: Zielerreichung der Erfolgskriterien.....	62
Tabelle 20: Zielerreichung der Anforderungen an den Sensor	63
Tabelle 21: Zielerreichung der Anforderungen an die Zentrale.....	64

Anhang A.D Glossar

Abkürzung	Erklärung
ADC	"Analog / Digital Converter" Analog Digital Wandler
AP	Arbeitspaket
AS	Alarmstate in der FSM
AT	"ATtention" -Kommandos der FA Hayes Microcomputer Products für die Modemansteuerung
DA	Diplomarbeit
dB	Dezibel ist eine logarithmische Einheit
dBm	Dezibel Milliwatt ist eine logarithmische Einheit in der Hochfrequenztechnik
DevKitC	"Development Kit C" Entwicklungsboard der ESP Mikrokontrollerfamilie
EKF	Elektronische Kriegführung (inkl Elektronischer Kampf)
ESP	Mikrokontrollerfamilie der Firma Espressif Systems
FSM	"Finite State Machine" Endliche Statusmaschine
GD	"Gas Detector" Gasmelder
GND	"Ground" entspricht dem Nullpotential
GPIO	"General Purpose In Out" Ein/Ausgänge von Mikrocontrollern
HAT	"Hardware Attached on Top" Aufsteckmodule z.B. für Raspberry Pi
MQTT	"Message Queueing Telemetry Transport" Client Server Protokoll
MQTT-Broker	Verteilzentrale der MQTT-Nachrichten
MQTT-Publisher	Anbieter von MQTT Nachrichten
MQTT-Subscriber	Abonnent von MQTT Nachrichten
MQTT-Topic	Thema einer MQTT Nachricht
NO_AS	Kein Alarm Status in der FSM
QoS	"Quality of Service" Übertragungsqualität von MQTT Nachrichten
RS	"Reset / Set"-Flipflop Rücksetzdominant
RSSI	"Radio Signal Strength Indikator" WiFi Empfangspegel in dBm
RST	"Reset" Rücksetzpin eines ESP Mikrokontrollers
RTC	"Real Time Clock" interne Uhr von Mikrocontrollern
RXD	Empfangsleitung der UART-Schnittstelle
SD	"Smoke Detector" Rauchmelder
SIM	"Subscriber Identity Module" Chipkarte for Mobiltelefone
SoC	"System on a Chip" entspricht einem Ein Chip System
TTL	"Transistor Transistor Logik" basierend auf 5V
TXD	Sendeleitung der UART-Schnittstelle
UART	"Universal Asynchronous Receiver / Transmitter " serielle Schnittstelle
vBMA	vernetzte Brandmeldeanlage
VCC	"Voltage Common Collector" positive Betriebsspannung
VDC	Gleichspannung
VKF	Vereinigung kantonaler Feuerversicherungen
VPN	"Virtual Private Network" geschützte Netzwerkverbindung
VSS	"Voltage at the source terminal" entspricht meistens GND

Anhang B Sourcecode

Anhang B.A Sourcecode Sensor (Beispiel Kinderzimmer 1)

```
//DA 2023 von Ueli Möhl
//=====

//Sensorcode_Kinderzimmer 1
//-----

#include <ESP8266WiFi.h>
#include <ArduinoMqttClient.h>

// Wi-Fi-Netzwerk-Verbindungsdaten
//-----
const char ssid[] = "B-TEKO";
const char password[] = "██████████";

// MQTT Broker-Daten
//-----
const char mqtt_broker[] = "10.100.42.91" //IP Raspi TEKO
const int mqtt_port = 1883;
const char mqtt_user[] = "mqtt_uele";
const char mqtt_password[] = "██████";
const char topic1[] = "Kinderzimmer_1";
const char topic2[] = "Betriebsspannung 1";
const char topic3[] = "RSSI Pegel 1";

// Variablendeklaration
//-----
const int alarmPin = 4;
const int adcPin = A0;
float spannung = 0;
int adcWert = 0;
int pegelWifi = 0;
bool alarmRS;

// Wi-Fi-Client
//-----
WiFiClient espClient;

// MQTT-Client
//-----
MqttClient mqttClient(espClient);

void setup()
{
  // Serielle Kommunikation initialisieren
  //-----
  Serial.begin(115200);

  // Alarmpin definieren
  //-----
  pinMode(alarmPin, INPUT);
```

```
// Wi-Fi-Verbindung herstellen
//-----
WiFi.begin(ssid, password);

Serial.println();
Serial.print("Versuche mit dem WiFi Netzwerk zu verbinden: ");
Serial.println(ssid);

while (WiFi.status() != WL_CONNECTED)
{
  delay(500);
  Serial.print(".");
}

Serial.println();
Serial.println("Du bist mit dem WiFi Netzwerk verbunden");
Serial.println();
Serial.print("Die IP Adresse lautet: ");
Serial.print(WiFi.localIP());
Serial.println();

//MQTT-Client konfigurieren
//-----
mqttClient.setUsernamePassword(mqtt_user, mqtt_password);

//MQTT-Client (Raspberry Pi) verbinden
//-----
Serial.print("Versuche mit dem MQTT-Broker zu verbinden: ");
Serial.println(mqtt_broker);

if (!mqttClient.connect(mqtt_broker, mqtt_port))
{
  Serial.print("MQTT Verbindung fehlgeschlagen! Error code = ");
  Serial.println(mqttClient.connectError());

  while (1);
}

Serial.println("Du bist mit dem MQTT-Broker verbunden!");
Serial.println();
}
//=====

void loop()
{
  // Handshake Arduino mit MQTT-Client
  //-----
  mqttClient.poll();

  // Alarmpin auslesen
  //-----
  alarmRS = digitalRead(alarmPin);

  // Betriebsdaten ausmessen
  //-----
  adcWert = analogRead(adcPin);
  spannung = adcWert * (4.52/1023.0);

  pegelWifi = WiFi.RSSI();

  // Betriebsdatenauswertung
  //-----
  if(alarmRS)
  {
    //Ausgabe der Meldung auf dem seriellen Monitor
    //-----
    Serial.println();
    Serial.println("=====");
    Serial.println("Rauchalarm aus Kinderzimmer 1!");
    Serial.println("=====");
    Serial.println();
  }
}
```

```
//MQTT Client starten mit Abonnement Topic 1 und QoS=1
//-----
    mqttClient.beginMessage(topic1, true, 1);

//Versenden der topic1 Alarmnachricht
//-----
    mqttClient.print("Rauchalarm aus Kinderzimmer 1!");
    mqttClient.endMessage();
}
else
{
//Ausgabe der Betriebsdaten
//-----
    Serial.println();
    Serial.println("-----");
    Serial.print("Kommando an Broker: ");
    Serial.println("kein Alarm");
    Serial.print("Batteriespannung: ");
    Serial.print(spannung);
    Serial.println(" V");
    Serial.print("Wifi Empfangspegel: ");
    Serial.print(pegelWifi);
    Serial.println(" dBm");
    Serial.println("-----");
    Serial.println();

//MQTT Client starten mit Abonnement Topic 1 und QoS=1
//-----
    mqttClient.beginMessage(topic1, true, 1);

//Versenden der topic1 Alarmnachricht
//-----
    mqttClient.print("kein Alarm");
    mqttClient.endMessage();

//MQTT Client starten mit Abonnement Topic 2 und QoS=1
//-----
    mqttClient.beginMessage(topic2, true, 1);

//Versenden der topic2 Betriebsspannung
//-----
    mqttClient.print("Batteriespannung: ");
    mqttClient.print(spannung);
    mqttClient.println(" V");
    mqttClient.endMessage();

//MQTT Client starten mit Abonnement Topic 3 und QoS=1
//-----
    mqttClient.beginMessage(topic3, true, 1);

//Versenden der topic3 RSSI Pegel
//-----
    mqttClient.print("Wifi Empfangspegel: ");
    mqttClient.print(pegelWifi);
    mqttClient.println(" dBm");
    mqttClient.endMessage();
}

//ESP in Tiefschlaf versetzen für 3h
//-----
    delay(5000);

    Serial.println("Ich gehe mal schlafen");
    ESP.deepSleep(10800e6);
}
```

Anhang B.B Sourcecode Alarm-Mikrokontroller (Zentrale)

1. Zentralencode

```
//DA 2023 von Ueli Möhl
//=====

//Zentralencode
//-----
#include <WiFi.h>
#include <PubSubClient.h>

// Wi-Fi-Netzwerk-Verbindungsdaten
//-----
const char ssid[] = "B-TEKO";
const char password[] = "██████████";

// MQTT Broker-Daten
//-----
const char mqtt_broker[] = "10.100.42.91" //IP Raspi TEKO
const int mqtt_port = 1883;
const char mqtt_user[] = "mqtt_uele";
const char mqtt_password[] = "██████";
const char topic1[] = "Kinderzimmer_1";
const char topic4[] = "Kinderzimmer_2";
const char topic7[] = "Garage_3";
const char topic10[] = "Testsensor_4";

// Konstantendeklaration
//-----
const int readyLED = 2;
const int flasherLED = 22;
const int alarmHorn = 23;
const int alarmReset = 21;

// Variablendeklaration
//-----
String receivedMessage = "";
bool stateMessage = true;
bool flashLED = false;

// FSM State Deklaration
//-----
enum AlarmState
{
    INITIALIZING,
    NO_AS,
    AS_KINDERZIMMER_1,
    AS_KINDERZIMMER_2,
    AS_GARAGE_3,
    AS_KINDERZIMMER_4
};

//Status FSM Initialisieren
//-----
enum AlarmState currentState = INITIALIZING;

// Wi-Fi-Client
//-----
WiFiClient espClient;

// MQTT-Client
//-----
PubSubClient client(espClient);

void setup()
{
    // Serielle Kommunikation initialisieren
    //-----
    Serial.begin(115200);
```

```
// Serielle Kommunikation mit SIM7600 initialisieren
//-----
Serial2.begin(115200);

// Ein-/Ausgänge konfigurieren
//-----
pinMode(readyLED, OUTPUT);
pinMode(flasherLED, OUTPUT);
pinMode(alarmHorn, OUTPUT);
pinMode(alarmReset, INPUT);

// Wi-Fi-Verbindung herstellen
//-----
WiFi.begin(ssid, password);

Serial.println();
Serial.print("Versuche mit dem WiFi Netzwerk zu verbinden: ");
Serial.println(ssid);

while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}

Serial.println();
Serial.println("Du bist mit dem WiFi Netzwerk verbunden");
Serial.print("Die IP Adresse lautet: ");
Serial.print(WiFi.localIP());
Serial.println();
Serial.println();

//MQTT-Client konfigurieren
//-----
client.setServer(mqtt_broker, mqtt_port);
client.setCallback(Funktion_1_callback);

//MQTT-Client (Raspberry Pi) verbinden
//-----
while (!client.connected())
{
    Serial.print("Versuche mit dem MQTT-Broker zu verbinden: ");
    Serial.println(mqtt_broker);

    if (client.connect("ESP32Client", mqtt_user, mqtt_password))
    {
        Serial.println("Du bist mit dem MQTT-Broker verbunden!");
        Serial.println();
        client.subscribe(topic1, 1);
        client.subscribe(topic4, 1);
        client.subscribe(topic7, 1);
        client.subscribe(topic10, 1);

        Serial.println("Abonierete Topics: ");
        Serial.println(topic1);
        Serial.println(topic4);
        Serial.println(topic7);
        Serial.println(topic10);
        Serial.println();
    }
    else
    {
        Serial.print("MQTT Verbindung fehlgeschlagen! Error code = ");
        Serial.println(client.state());
        delay(2000);
    }
}
}
```

```
//Initialisieren und konfigurieren des SIM7600 Modul
//-----
Serial.println("Initialisiere das SIM7600 Modul...");
delay(60000);
Serial2.println("AT\r");
delay(2000);
Serial.println("SIM7600 Modul ist aufgestartet!");

Serial.println();
}

void loop()
{
//Wiederaufbau der Verbindung
//-----
Funktion_2_reconnect();

//FSM zur Steuerung der Alarmer
//-----
switch (currentState)
{
case (INITIALIZING):
{
Serial.print("Aktueller Status ist: ");
Serial.println("INITIALIZING");
Serial.println("-----");
Serial.println();

currentState = NO_AS;
break;
}

case (NO_AS):
{
if(stateMessage)
{
Serial.print("Aktueller Status ist: ");
Serial.println("NO_AS");
Serial.println("-----");
Serial.println();
digitalWrite(readyLED, HIGH);
stateMessage = false;
}

if (receivedMessage.endsWith("1!"))
{
currentState = AS_KINDERZIMMER_1;
}
else if (receivedMessage.endsWith("2!"))
{
currentState = AS_KINDERZIMMER_2;
}
else if (receivedMessage.endsWith("3!"))
{
currentState = AS_GARAGE_3;
}
else if (receivedMessage.endsWith("4!"))
{
currentState = AS_KINDERZIMMER_4;
}
else if(flashLED)
{
Funktion_5_flashLED();
}
break;
}
}
}
```

```
case (AS_KINDERZIMMER_1):
{
    digitalWrite(readyLED, LOW);
    Serial.print("Aktueller Status ist: ");
    Serial.println("AS_KINDERZIMMER_1");
    Serial.println("-----");
    Serial.println();

    Funktion_3_alarm_on("Alarm Kinderzimmer 1!");

    receivedMessage = "";
    stateMessage = true;
    currentState = NO_AS;
    break;
}

case (AS_KINDERZIMMER_2):
{
    digitalWrite(readyLED, LOW);
    Serial.print("Aktueller Status ist: ");
    Serial.println("AS_KINDERZIMMER_2");

    Funktion_3_alarm_on("Alarm Kinderzimmer 2!");

    receivedMessage = "";
    stateMessage = true;
    currentState = NO_AS;
    break;
}

case (AS_GARAGE_3):
{
    digitalWrite(readyLED, LOW);
    Serial.print("Aktueller Status ist: ");
    Serial.println("AS_GARAGE_3");
    Serial.println("-----");
    Serial.println();

    Funktion_3_alarm_on("Alarm Garage 3!");

    receivedMessage = "";
    stateMessage = true;
    currentState = NO_AS;
    break;
}

case (AS_KINDERZIMMER_4):
{
    digitalWrite(readyLED, LOW);
    Serial.print("Aktueller Status ist: ");
    Serial.println("AS_KINDERZIMMER_4");
    Serial.println("-----");
    Serial.println();

    Funktion_3_alarm_on("Alarm Testsensor 4!");

    receivedMessage = "";
    stateMessage = true;
    currentState = NO_AS;
    break;
}

default:
{
    currentState = INITIALIZING;
}
}
//Alarm LED und Sirene ausschalten
//-----
Funktion_4_alarm_off();

//Aufrechterhaltung der Verbindung und Nachrichtenempfang
//-----
client.loop();
}
```

2. Funktion 1

```
//Funktion 1
//-----
void Funktion_1_callback(char topic[], byte payload[], unsigned int length)
{
//Byte-Array in String konvertieren
//-----
    receivedMessage = "";
    for (int i = 0; i < length; i++)
    {
        receivedMessage += (char)payload[i];
    }

//Ausgabe des Alarms
//-----
    Serial.println();
    Serial.println("=====");
    Serial.println("Empfangener Alarm:");
    Serial.println("=====");
    Serial.println(receivedMessage);
    Serial.println();
}
```

3. Funktion 2

```
//Funktion 2
//-----
void Funktion_2_reconnect()
{
    if (!client.connected())
    {
        if (client.connect("ESP8266Client", mqtt_user, mqtt_password))

//Abonnieren der Topics bei Verbindungsaufbau
//-----
        {
            Serial.println("Verbunden mit dem MQTT-Broker!");
            client.subscribe(topic1);
            client.subscribe(topic4);
            client.subscribe(topic7);
            client.subscribe(topic10);
        }
        else
        {
            Serial.print("Verbindung zum MQTT-Broker fehlgeschlagen. Fehlercode: ");
            Serial.println(client.state());
            delay(2000);
        }
    }
}
```

4. Funktion 3

```
//Funktion 3
//-----
void Funktion_3_alarm_on(String command)

//lokaler Alarm einschalten
//-----
{
  flashLED = true;
  digitalWrite(alarmHorn, HIGH);

//Antworten des LTE/4G-Modul anzeigen
//-----
while (Serial2.available())
  {
    Serial.write(Serial2.read());
  }

//Alarm-Kurznachricht SMS versenden
//-----
  Serial.println();
  Serial.println("versende die Alarmnachricht!");
  Serial.println("-----");
  Serial2.println("AT+CMGF=1\r");
  delay(100);
  Serial2.println("AT+CMGS=\"0792854486\"");
  delay(100);
  Serial2.println(command);
  delay(100);
  Serial2.write(0x1A);
  delay(100);
  Serial.println();
  Serial.println("SMS erfolgreich gesendet");
  Serial.println("-----");
  Serial.println();
}
}
```

5. Funktion 4

```
//Funktion 4
//-----
void Funktion_4_alarm_off()
{
  if(digitalRead(alarmReset))

//lokaler Alarm ausschalten
//-----
  {
    flashLED = false;
    digitalWrite(flasherLED, LOW);
    digitalWrite(alarmHorn, LOW);
  }
}
```

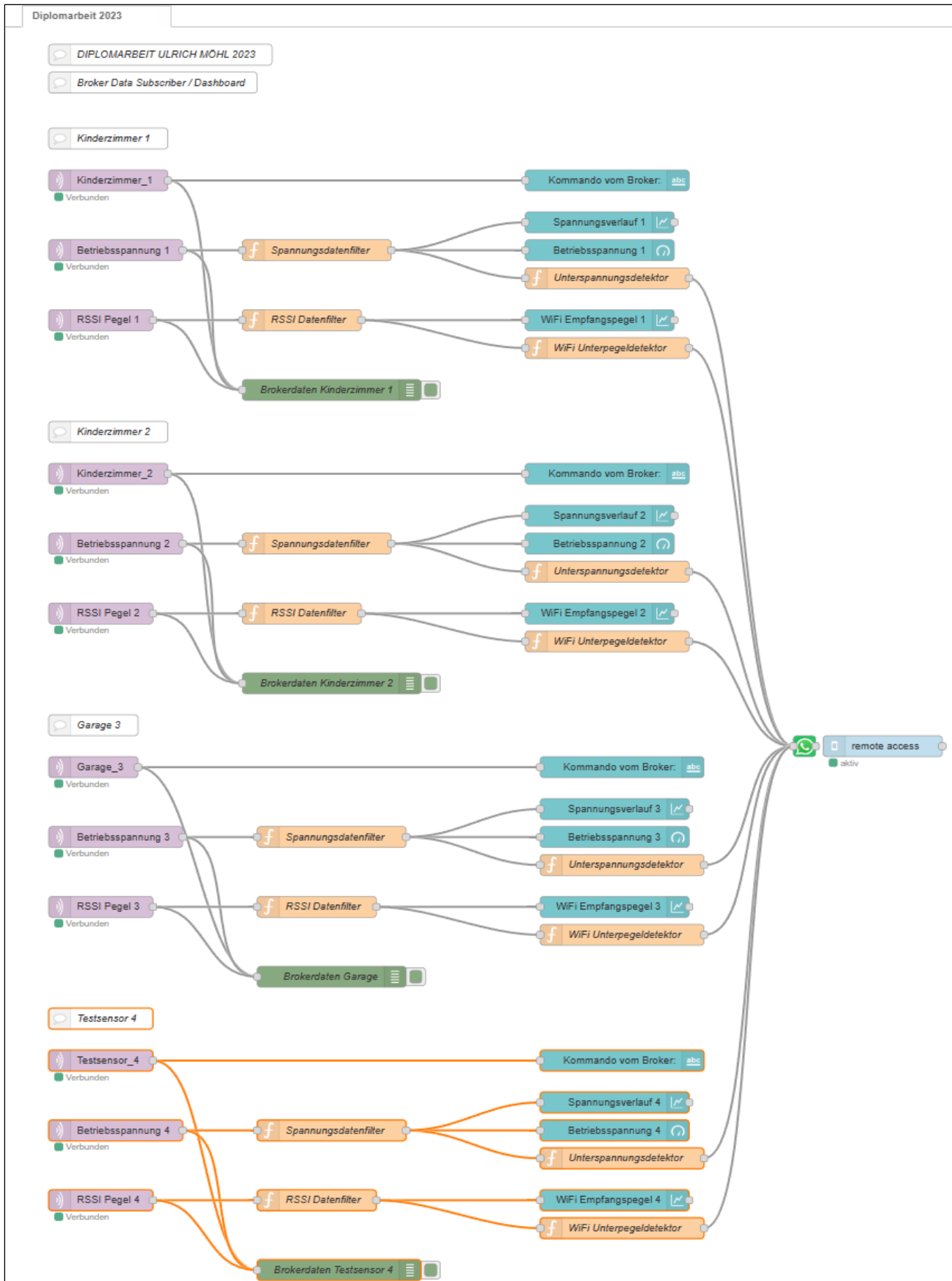
6. Funktion 5

```
//Funktion 5
//-----
void Funktion_5_flashLED()

//flash LED einschalten/ausschalten
//-----
{
  digitalWrite(flasherLED, HIGH);
  delay(50);
  digitalWrite(flasherLED, LOW);
  delay(500);
}
```

Anhang B.C Node-RED Programmierung

1. Flow Diplomarbeit 2023



2. Function-Node Programmierung Spannungsdatenfilter

```
1 // Spannungsdatenfilter DA23
2 // Ulrich Möhl
3 //=====
4 var data = msg.payload; //speichern in Variable
5 var split = data.split(":"); //Datentrennen mit Sonderzeichen
6 var spannung = split[1].substring(1,5); //Zahlzeichen extrahieren
7 var zahlWert = parseFloat(spannung); //Datentypumwandlung
8 msg.payload = zahlWert; //speichern in Variable
9 return msg; //Datenausgabe
```

3. Function-Node Programmierung RSSI-Datenfilter

```
1 //RSSI Datenfilter
2 //Ulrich Möhl
3 //=====
4 var data = msg.payload; //speichern in Variable
5 var split = data.split(":"); //Datentrennen mit Sonderzeichen
6 var rssi = split[1].substring(1,5); //Zahlzeichen extrahieren
7 var zahlWert = parseFloat(rssi); //Datentypumwandlung
8 msg.payload = zahlWert; //speichern in Variable
9 return msg; //Datenausgabe
```

4. Function-Node Programmierung Unterspannungsdetektor

```
1 // Unterspannungsdetektor DA23
2 // Ulrich Möhl
3 //=====
4 var betriebsspannung = msg.payload; //speichern in Variable
5
6 if(betriebsspannung < 3.2) //Bedingung für Alarmmeldung
7 {
8     msg.payload = "Unterspannung in Kinderzimmer 1";
9     return msg; //Ausgabe der Alarmmeldung
10 }
```

5. Function-Node Programmierung WiFi Unterpegeldetektor

```
1 // Unterpegeldetektor DA23
2 // Ulrich Möhl
3 //=====
4 var betriebspegel = msg.payload; //speichern in Variable
5
6 if(betriebspegel < -85) //Bedingung für Alarmmeldung
7 {
8     msg.payload = "Signalpegel in Kinderzimmer 1 zu tief";
9     return msg; //Ausgabe der Alarmmeldung
10 }
```

Anhang C Datenblätter und Zertifikate

Anhang C.A VKF Zertifikat



Vereinigung Kantonaler Feuerversicherungen

Auskunft über die Anwendbarkeit
gemäss den Schweizerischen
Brandschutzvorschriften

VKF Technische Auskunft Nr. 26549

Gruppe 611	Rauchwarmmelder
Gesuchsteller	Securiton AG Alpenstrasse 20 3052 Zollikofen Schweiz
Hersteller	Hekatron GmbH 79295 Sulzburg Germany
Produkt	GENIUS H
Beschrieb	Model Genius H, Sockeladapter HSD-Genius Batterielebensdauer: 10 Jahre
Anwendung	Rauchwarmmelder für Bauten und Anlagen
Untertagen	VDS, Köln: Prüfbericht 'BMA 12069' (17.07.2012), Prüfbericht '100121-AU01-SW01' (22.04.2010), Prüfbericht 'BMA 12011' (10.02.2012), Prüfbericht 'BMA 11064' (22.07.2011), Prüfbericht 'BMA 11011' (11.01.2011), Prüfbericht 'BMA 10072' (08.06.2010), Prüfbericht 'BMA 10071' (08.06.2010), Prüfbericht 'BMA 09166' (23.10.2009), Prüfbericht 'BMA 09136' (10.09.2009), Prüfbericht 'SW-2008241' (10.09.2009); Hersteller: Leistungserklärung 'CPR-31-13-001' (21.05.2014); VDS, Köln: Zertifikat 'G 209178' (19.09.2013)
Prüfbestimmungen	EN 14604
Beurteilung	Entspricht den Bestimmungen der VKF
Gültigkeitsdauer	31.12.2021
Ausstelldatum	10.02.2016
Ersetzt Anerkennung vom	-

Anerkennungsstelle der
kantonalen Brandschutzbehörden

Marcel Rumo

Roland Julmy



Anhang C.B Datenblatt MOSFET 2N7000 (Auszug)

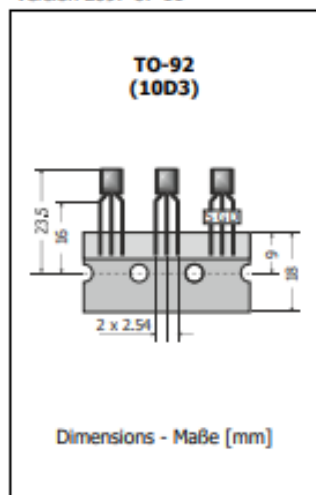
2N7000



2N7000
N-Channel Enhancement Mode FET
N-Kanal FET – Anreicherungstyp

$I_D = 200 \text{ mA}$ $V_{OSS} = 60 \text{ V}$
 $R_{DS(on)} < 5 \Omega$ $P_{tot} = 350 \text{ mW}$
 $T_{jmax} = 150^\circ\text{C}$

Version 2097-07-11



Typical Applications

Signal processing, Logic level converter, Drivers
Commercial grade
Suffix -Q: AEC-Q101 compliant ¹⁾
Suffix -AQ: in AEC-Q101 qualification ¹⁾

Features

Fast switching times
Compliant to RoHS, REACH, Conflict Minerals ¹⁾

Mechanical Data ¹⁾

Taped in ammo pack (Raster 2.54)
Weight approx.
Case material
Solder & assembly conditions



Typische Anwendungen

Signalverarbeitung, Pegelwandler, Treiberstufen
Standardausführung
Suffix -Q: AEC-Q101 konform ¹⁾
Suffix -AQ: in AEC-Q101 Qualifikation ¹⁾

Besonderheiten

Schnelle Schaltzeiten
Konform zu RoHS, REACH, Konfliktmineralien ¹⁾

Mechanische Daten ¹⁾

Gegurtet in Ammo-Pack (Raster 2.54)
Gewicht ca.
Gehäusematerial
Löt- und Einbaubedingungen

4000
0.18 g
UL 94V-0
260°C/10s
MSL N/A

Maximum ratings ²⁾

Grenzwerte ²⁾

Drain-Source-voltage – Drain-Source-Spannung	V_{OSS}	60 V
Drain-Gate-voltage – Drain-Gate-Spannung	$R_{GS} \leq 1 \text{ M}\Omega$	V_{DGS} 60 V
Gate-Source-voltage – Gate-Source-Spannung	DC $t_p < 50 \mu\text{s}$	V_{GSS} $\pm 20 \text{ V}$ $\pm 40 \text{ V}$
Power dissipation – Verlustleistung	P_{tot}	350 mW ³⁾
Drain current continuous – Drainstrom	I_D	200 mA
Peak Drain current – Drain-Spitzenstrom	I_{DM}	500 mA
Operating Junction temperature – Sperrschichttemperatur	T_j	150°C
Storage temperature – Lagerungstemperatur	T_s	-55...+150°C

Characteristics

Kennwerte

		$T_j = 25^\circ\text{C}$	Min.	Typ.	Max.
Drain-Source breakdown voltage – Drain-Source Durchbruchspannung	$I_D = 10 \mu\text{A}$	$V_{(BR)OSS}$	60 V		
Drain-Source leakage current – Drain-Source Leckstrom	$V_{GS} = 48 \text{ V}, V_{DS} = 0 \text{ V}$ $V_{GS} = 48 \text{ V}, V_{DS} = 0 \text{ V}, T_j = 125^\circ\text{C}$	I_{DSS}		1 μA 1 mA	
Gate-Body leakage current – Gate-Substrat Leckstrom	$V_{GS} = \pm 15 \text{ V}$	$\pm I_{GSS}$		10 nA	

- Please note the [detailed information on our website](#) or at the beginning of the data book
Bitte beachten Sie die [detaillierten Hinweise auf unserer Internetseite](#) bzw. am Anfang des Datenbuches
- $T_A = 25^\circ\text{C}$, unless otherwise specified – $T_A = 25^\circ\text{C}$, wenn nicht anders angegeben
- Valid, if leads are kept at ambient temperature at a distance of 2 mm from case
Gültig wenn die Anschlussdrähte in 2 mm Abstand vom Gehäuse auf Umgebungstemperatur gehalten werden

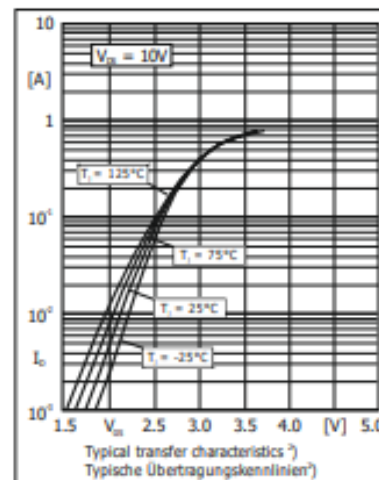
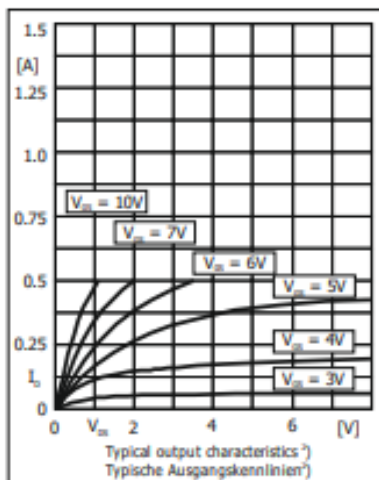


2N7000

Characteristics

Kennwerte

	$T_j = 25^\circ\text{C}$	Min.	Typ.	Max.
Gate-Threshold voltage – Gate-Source Schwellspannung $V_{GS} = V_{GS}, I_D = 1 \text{ mA}$	$V_{GS(th)}$	0.8 V	–	3 V
Drain-Source on-voltage – Drain-Source-Spannung $V_{GS} = 10 \text{ V}, I_D = 500 \text{ mA}$ $V_{GS} = 4.5 \text{ V}, I_D = 75 \text{ mA}$	$V_{DS(on)}$	–	2.5 V 0.45 V	–
Drain-Source on-state resistance – Drain-Source Einschaltwiderstand $V_{GS} = 10 \text{ V}, I_D = 500 \text{ mA}$ $V_{GS} = 4.5 \text{ V}, I_D = 75 \text{ mA}$	$R_{DS(on)}$	–	–	5 Ω 6 Ω
Forward Transconductance – Übertragungssteilheit $V_{GS} = 10 \text{ V}, I_D = 200 \text{ mA}$	g_{fs}	100 mS	–	–
Input Capacitance – Eingangskapazität $V_{GS} = 25 \text{ V}, f = 1 \text{ MHz}$	C_{iss}	–	60 pF	–
Output Capacitance – Ausgangskapazität $V_{GS} = 25 \text{ V}, f = 1 \text{ MHz}$	C_{oss}	–	25 pF	–
Reverse Transfer Capacitance – Rückwirkungskapazität $V_{GS} = 25 \text{ V}, f = 1 \text{ MHz}$	C_{rss}	–	5 pF	–
Turn-On Delay Time – Einschaltverzögerung $V_{DD} = 15 \text{ V}, R_c = 30 \Omega, I_D = 0.5 \text{ A}, V_{GS} = 10 \text{ V}, R_G = 25 \Omega$	t_{on}	–	10 ns	–
Turn-Off Delay Time – Ausschaltverzögerung $V_{DD} = 15 \text{ V}, R_c = 30 \Omega, I_D = 0.5 \text{ A}, V_{GS} = 10 \text{ V}, R_G = 25 \Omega$	t_{off}	–	10 ns	–
Thermal resistance junction to ambient Wärmewiderstand Sperrschicht – Umgebung	$R_{\theta JA}$	< 357 K/W ¹⁾		



¹⁾ Valid, if leads are kept at ambient temperature at a distance of 2 mm from case
Gültig wenn die Anschlussdrähte in 2 mm Abstand vom Gehäuse auf Umgebungstemperatur gehalten werden

Anhang C.C Datenblatt RS-Flipflop CD4043BE (Auszug)



Data sheet acquired from Hamis Semiconductor
SCH5041D - Revised October 2003

CMOS Quad 3-State R/S Latches

High-Voltage Types (20-Volt Rating)
Quad NOR R/S Latch - CD4043B
Quad NAND R/S Latch - CD4044B

■ CD4043B types are quad cross-coupled 3-state CMOS NOR latches and the CD4044B types are quad cross-coupled 3-state CMOS NAND latches. Each latch has a separate Q output and individual SET and RESET inputs. The Q outputs are controlled by a common ENABLE input. A logic "1" or high on the ENABLE input connects the latch states to the Q outputs. A logic "0" or low on the ENABLE input disconnects the latch states from the Q outputs, resulting in an open circuit condition on the Q outputs. The open circuit feature allows common busing of the outputs.

The CD4043B and CD4044B types are supplied in 16-lead hermetic dual-in-line ceramic packages (F3A suffix), 16-lead dual-in-line plastic packages (E suffix), 16-lead small-outline packages (D, DR, DT, DW, DWR, and NSR suffixes), and 16-lead thin shrink small-outline packages (PW and PWR suffixes).

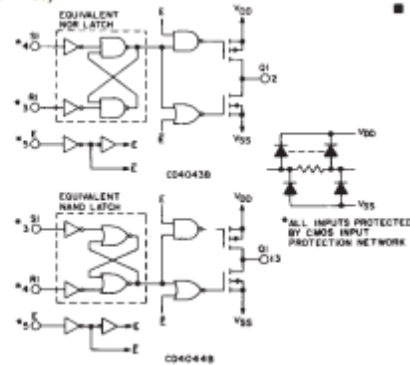


Fig. 1 - Logic diagrams.

MAXIMUM RATINGS, Absolute-Maximum Values:

DC SUPPLY-VOLTAGE RANGE, (V _{DD}) (Voltages referenced to V _{SS} Terminal)	-0.5V to +20V
INPUT VOLTAGE RANGE, ALL INPUTS	-0.5V to V _{DD} +0.5V
DC INPUT CURRENT, ANY ONE INPUT	±10mA
POWER DISSIPATION PER PACKAGE (P _D):	
For T _A = -55°C to +100°C	500mW
For T _A = +100°C to +125°C	Derate Linearly at 12mW/°C to 200mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR	
FOR T _A = FULL PACKAGE-TEMPERATURE RANGE (All Package Types)	100mW
OPERATING-TEMPERATURE RANGE (T _A)	-55°C to +125°C
STORAGE TEMPERATURE RANGE (T _{stg})	-65°C to +150°C
LEAD TEMPERATURE (DURING SOLDERING):	
At distance 1/16 ± 1/32 inch (1.59 ± 0.79mm) from case for 10s max	+265°C

CD4043B, CD4044B Types

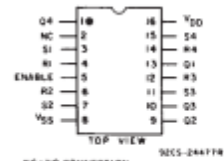
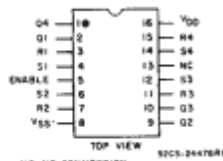
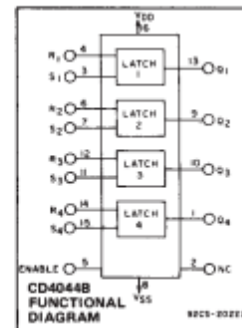
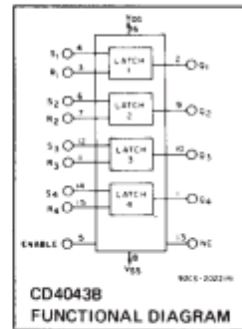
Features:

- 3-state outputs with common output ENABLE
- Separate SET and RESET inputs for each latch
- NOR and NAND configurations
- 5-V, 10-V, and 15-V parametric ratings
- Standardized symmetrical output characteristics
- 100% tested for quiescent current at 20 V
- Maximum input current of 1 μA at 18 V over full package temperature range; 100 nA at 18 V and 25°C
- Noise margin (over full package temperature range):
 - 1 V at V_{DD} = 5 V
 - 2 V at V_{DD} = 10 V
 - 2.5 V at V_{DD} = 15 V

■ Meets all requirements of JEDEC Tentative Standard No. 10B, "Standard Specifications for Description of 'B' Series CMOS Devices"

Applications:

- Holding register in multi-register system
- Four bits of independent storage with output ENABLE
- Strobed register
- General digital logic
- CD4043B for positive logic systems
- CD4044B for negative logic systems



CD4043B	TERMINAL ASSIGNMENTS	CD4044B	
S	R	E	Q
X	X	0	OC*
0	0	1	NC*
1	0	1	1
0	1	1	0
1	1	1	Δ

*OPEN CIRCUIT
+NO CHANGE
Δ DOMINATED BY S=1 INPUT

CD4044B			
S	R	E	Q
X	X	0	OC*
1	1	1	NC*
0	1	1	1
0	0	1	0
0	0	1	Δ

*OPEN CIRCUIT
+NO CHANGE
Δ Δ DOMINATED BY R=0 INPUT

TRUTH TABLES

Recommended Operating Conditions T_A = 25°C
For maximum reliability, nominal operating conditions should be selected so that operation is always within the following ranges.

Characteristic	V _{DD} (V)	Min.	Max.	Units
Supply-Voltage Range (T _A = Full Package Temperature Range)		3	18	V
SET or RESET Pulse Width, t _W	5	160	-	ns
	10	80	-	
	15	40	-	

CD4043B, CD4044B Types

STATIC ELECTRICAL CHARACTERISTICS

CHARACTERISTIC	CONDITIONS			LIMITS AT INDICATED TEMPERATURES (°C)							UNITS
	V _O (V)	V _{IN} (V)	V _{DD} (V)					+25			
				-55	-40	+85	+125	Min.	Typ.	Max.	
Quiescent Device Current, I _{DD} Max.	-	0,5	5	1	1	30	30	-	0.02	1	μA
	-	0,10	10	2	2	60	60	-	0.02	2	
	-	0,15	15	4	4	120	120	-	0.02	4	
	-	0,20	20	20	20	600	600	-	0.04	20	
Output Low (Sink) Current I _{OL} Min.	0.4	0,5	5	0.64	0.61	0.42	0.36	0.51	1	-	mA
	0.5	0,10	10	1.6	1.5	1.1	0.9	1.3	2.6	-	
	1.5	0,15	15	4.2	4	2.8	2.4	3.4	6.8	-	
Output High (Source) Current, I _{OH} Min.	4.6	0,5	5	-0.64	-0.61	-0.42	-0.36	-0.51	-1	-	mA
	2.5	0,5	5	-2	-1.8	-1.3	-1.15	-1.6	-3.2	-	
	9.5	0,10	10	-1.6	-1.5	-1.1	-0.9	-1.3	-2.6	-	
	13.5	0,15	15	-4.2	-4	-2.8	-2.4	-3.4	-6.8	-	
Output Voltage: Low-Level, VOL Max.	-	0,5	5	0.05				-	0	0.05	V
	-	0,10	10	0.05				-	0	0.05	
	-	0,15	15	0.05				-	0	0.05	
Output Voltage: High-Level, VOH Min.	-	0,5	5	4.95				4.95	5	-	V
	-	0,10	10	9.95				9.95	10	-	
	-	0,15	15	14.95				14.95	15	-	
Input Low Voltage, VIL Max.	0.5, 4.5	-	5	1.5				-	-	1.5	V
	1, 9	-	10	3				-	-	3	
	1.5, 13.5	-	15	4				-	-	4	
Input High Voltage, VIH Min.	0.5, 4.5	-	5	3.5				3.5	-	-	V
	1, 9	-	10	7				7	-	-	
	1.5, 3.5	-	15	11				11	-	-	
Input Current I _{IN} Max.	-	0,18	18	±0.1	±0.1	±1	±1	-	±10 ⁻⁵	±0.1	μA
3-State Output Leakage Current I _{OUT} Max.	0,18	0,18	18	±0.4	±0.4	±12	±12	-	±10 ⁻⁴	±0.4	μA

3
COMMERCIAL CMOS
HIGH VOLTAGE ICs

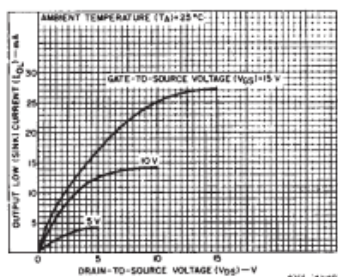


Fig. 2 - Typical output low (sink) current characteristics.

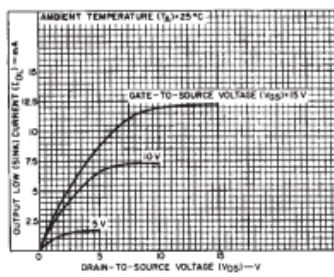


Fig. 3 - Minimum output low (sink) current characteristics.

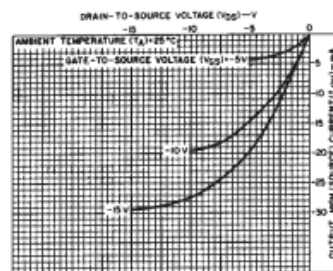


Fig. 4 - Typical output high (source) current characteristics.

Anhang C.D Datenblatt LIR2450 Akku (Auszug)

1. Preface:

The purpose of this product specification is to provide technical information for the rechargeable Lithium-ion button battery LIR2450,

2. Description and Model

2.1 Description	Rechargeable Lithium-ion button battery
2.2 Model	LIR2450

3. Specification

3.1 Capacity	120±10mAh
3.2 Charging Voltage	4.20V
3.3 Nominal Voltage	3.6V
3.4 Standard Charging Method	Constant current: 24mA constant voltage 4.20V
3.5 Cut-off Discharge Voltage	2.75V
3.6 Max. Discharge Current	120mA
3.7 Max. Charge Current	120mA
3.8 Cycle Life	>500 cycles at 0.2C mA discharge
3.9 Ambient Temperature	
for Standard Charge	0°C~45°C
for Discharge	-20°C~60°C
3.10 Storage	
for within the temperature	-20°C~60°C
for within the humidity	≤75%
3.11 Energy Density	
Wh/L	
Wh/Kg	
3.12 Weight of Bare Cell	~5.6g
3.13 Charge State Internal Impedance	≤400mΩ

4. Appearance

Appearance shall be free from any remarkable scratch, flaws, rust, discoloration or electrolyte leakage (visible or by smell).

5. Standard Test condition

5.1 Environment Conditions: Unless otherwise specified, all test stated in this Product Specification are conducted within the temperature 15~25°C and the humidity 45~85%RH.

5.2 Test Equipment:

Impedance meter: The impedance meter with AC 1kHz should be used.

Anhang C.E Datenblatt 10mm LED



Distributed by Conrad Electronic SE • Klaus-Conrad-Str. 1 • D-92240 Hirschau

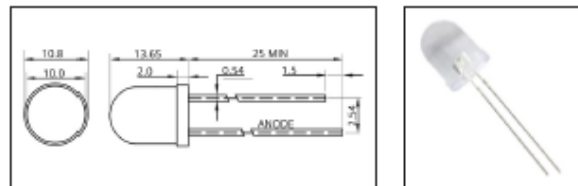
Datasheet

Item no. 1577410

Bedrahtete-Leuchtdioden

10 mm, klar

- Anschluss THR
- Ausführung rund
- Bauform 10 mm
- Linse wasserklar



Artkbez:	Farbe:	Wellenlänge*:	Helligkeit*:	Farbtemperatur*:	Winkel*:	Betriebsspannung*:
1573758	● rot	620 - 635 nm	1500 - 2000 mcd		25 °	1,9 V - 2,3 V
1577504	● rot	620 - 635 nm	800 - 1200 mcd		25 °	1,9 V - 2,3 V
1577505	● rot	620 - 635 nm	3000 - 4000 mcd		25 °	1,9 V - 2,3 V
1577408	● rot	620 - 635 nm	8000 - 9000 mcd		25 °	1,9 V - 2,3 V
1573759	● gelb	580 - 595 nm	1500 - 2000 mcd		25 °	1,9 V - 2,3 V
1577409	● gelb	580 - 595 nm	800 - 1200 mcd		25 °	1,9 V - 2,3 V
1577506	● gelb	580 - 595 nm	3000 - 4000 mcd		25 °	1,9 V - 2,3 V
1577410	● gelb	580 - 595 nm	8000 - 9000 mcd		25 °	1,9 V - 2,3 V
1573760	● grün	565 - 575 nm	800 - 1500 mcd		25 °	1,9 V - 2,3 V
1577411	● grün	520 - 530 nm	3000 - 4800 mcd		25 °	2,9 V - 3,5 V
1577507	● grün	520 - 530 nm	8000 - 9600 mcd		25 °	2,9 V - 3,5 V
1577508	● blau	460 - 470 nm	6000 - 6800 mcd		25 °	2,9 V - 3,3 V
1573761	● blau	460 - 470 nm	1500 - 2000 mcd		25 °	2,9 V - 3,3 V
1577509	○ weiß		10000 - 12000 mcd	5500 - 6500 K	25 °	2,9 V - 3,3 V
1573762	○ weiß		2000 - 3500 mcd	5500 - 6500 K	25 °	2,9 V - 3,3 V

*Betriebsstrom 30 mA

Anhang D Eigenständigkeitserklärung

Arbeiten, die nachweisbar in vollen Umfang oder in den wesentlichen Teilen unverändert oder ohne korrekte Quellenangabe übernommen werden, gelten als vorfabriziert und werden nicht bewertet.

Ich bestätige, dass ich die vorliegende Diplomarbeit selbstständig verfasst und alle benutzten Quellen gekennzeichnet habe. Diese Arbeit wurde weder in gleicher noch in ähnlicher Form bereits einer Prüfungskommission vorgelegt.

Name / Vorname:

Ulrich Möhl



Ort / Datum / Unterschrift:

Wimmis, 29. Oktober 2023