



**HOCHSCHULE
MITTWEIDA**
University of Applied Sciences

Bachelorarbeit

Herr
Hang Yu

**Entwicklungsansatz zur
Erkennung von gefährlichen
Stoffen/Gasen für Persönliche
Schutzausrüstungen**

Mittweida, 2024

Fakultät: Ingenieurwissenschaften

Bachelorarbeit

Entwicklungsansatz zur Erkennung von gefährlichen Stoffen/Gasen für Persönliche Schutzausrüstungen

Autor:

Herr Hang Yu

Studiengang:

Elektro- und Informationstechnik

Seminargruppe:

EI20wA-BC

Erstprüfer:

Prof. Dr.-Ing. Michael Kuhl

Zweitprüfer:

M. Sc. Kevin Blümel

Einreichung:

Mittweida, 15.08.2024

Verteidigung/Bewertung:

Mittweida, 2024

Bachelor Thesis

Development approach for the detection of hazardous substances/gases for per- sonal protective equipment

author:

Mr.

Hang Yu

course of studies:

Electrical and information Technology

seminar group:

EI20wA - BC

first examiner:

Prof. Dr.-Ing. Michael Kuhl

second examiner:

M. Sc. Kevin Blümel

submission:

Mittweida, 15.08.2024

defense/evaluation:

Mittweida, 2024

Bibliografische Beschreibung:

Ipsain, cumque ut hic, neque at vitae odio neque sit dolores sint quo! Utquae, nesciunt quas animi sint velit ad modi in natus! Ineveniet ad iusto ad quos ratione sint nesciunt ullam hic modi id libero tempore neque amet. Dolorequo et dolore ipsum sit totam quae et, dolor vel, natus sed lorem. Quiut, sed unde dolores culpa laborum nobis unde, nobis vel. Iustodolore at neque elit dicta nesciunt dolor libero quam at modi eveniet dicta vero hic. Quiaunde neque iure et cumque, velit laborum ipsum facilis amet eveniet in natus aut dicta. Sitquasi rem at neque quia nihil at, tempore eos neque sed quam dolores eos! Nequeat facilis qui dolore ut dolores vel, dolore sint in laborum.

Referat:

Diese Studie entwickelte ein miniaturisiertes Gefahrstoffdetektionssystem basierend auf MEMS-MOS-Gassensoren, mit dem Ziel, die Sicherheit von Hochrisikoberufen wie Feuerwehrleuten zu verbessern. Das System kombiniert innovativ fortschrittliche Sensortechnologie mit tragbarem Gerätedesign und ermöglicht Echtzeit-Multigas-Detektion sowie Cloud-Plattform-Interaktion. Die Kerninnovation liegt in der Verwendung von TinyML-Algorithmen für intelligente Detektion, was die Genauigkeit und Stabilität des Systems in komplexen Umgebungen erheblich verbessert. Experimentelle Ergebnisse zeigen, dass das System bei der Erkennung hochkonzentrierter Gefahrstoffe hervorragende Leistungen erbringt und sich gut an Umweltveränderungen anpasst. Darüber hinaus erhöhen das modulare Design und die Szenario-Voreinstellungsfunktion die Flexibilität und Anwendbarkeit des Systems. Diese Studie bietet neue Ansätze für die Intelligenz von persönlicher Schutzausrüstung und hat breite Anwendungsperspektiven zur Verbesserung der Arbeitssicherheit.

Inhalt

Inhalt	I
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VI
Abkürzungsverzeichnis	VII
1 Einleitung	1
2 Aufgabenstellung und These	3
2.1 Aufgabenspektrum	3
2.2 Antizipierte Lösungsansätze	3
3 Stand der Technik	5
3.1 Übersicht über PSA.....	5
3.2 Interviews mit Feuerwehreinsetzungskräften	5
3.3 Analyse bestehender Gefahrstoffdetektionssysteme.....	6
3.3.1 Häufige Gefahrstoffe bei Bränden	6
3.3.2 Gängige Detektionstechnologien.....	8
3.3.3 Tragbare Gasetektionsgeräte	9
3.3.4 Integrationsmethoden für Detektionsgeräte	10
4 Aufgabenspezifikation	12
4.1 Systemfunktionen.....	12
4.2 Leistungsindikatoren	12
4.3 Benutzeroberfläche	13
4.4 Innovative Designkonzepte	15
5 Umsetzung	16
5.1 Gesamtkonzept.....	16
5.1.1 Gesamtarchitektur des Systems und Modulzuweisung.....	16
5.1.2 Hardware-Architektur	19
5.1.3 Software-Architektur	19
5.2 Hardware-Design	22
5.2.1 Hardwareauswahl.....	22
5.2.2 Schaltungsentwurf.....	33
5.2.3 Schaltungstest.....	35

5.2.4	Leiterplattendesign und -herstellung.....	36
5.3	Softwaredesign	36
5.3.1	Codestruktur.....	37
5.3.2	Verwendete Bibliotheken.....	37
5.3.3	Globale Definitionen	38
5.3.4	Initialisierungsfunktion und Hauptschleifenfunktion.....	39
5.3.5	Datenerfassungs- und Verarbeitungsalgorithmen.....	41
5.3.6	Erkennung und Alarmierung von Gefahrstoffen.....	44
5.3.7	Benutzeroberfläche	47
5.3.8	Datenverwaltung und -speicherung	51
5.3.9	Drahtlose Kommunikation und Datenvisualisierung.....	52
5.3.10	Erfassung von Zeit und Temperatur	53
5.3.11	Energiemanagement	54
5.3.12	Task-Scheduler	54
6	Systemaufbau und Inbetriebnahme.....	56
6.1	Systemimplementierung.....	56
6.1.1	Hardwaremontage.....	56
6.1.2	Softwareimplementierung.....	58
6.1.3	Stromverbrauchsanalyse.....	61
6.2	Probleme und deren Lösungsansätze.....	62
6.2.1	PCB-bezogene Problematik	62
6.2.2	Herausforderungen bei der Sensorleistung	62
7	Versuchsdurchführung.....	72
7.1	Versuchsaufbau	72
7.1.1	Zielsetzung der Versuche.....	72
7.1.2	Versuchshypothesen.....	72
7.1.3	Versuchsaufbau	73
7.1.4	Versuchsmaterialien und -ausrüstung	74
7.2	Versuchsdurchführung.....	74
7.3	Analyse und Bewertung der Versuchsergebnisse	76
7.3.1	Analyse der Detektionsfähigkeit	76
7.3.2	Analyse der Konsistenz und Reproduzierbarkeit	76
7.3.3	Vergleichende Methodenanalyse	77
7.3.4	Analyse des Einflusses von Umweltfaktoren	77

8 Zusammenfassung	79
8.1 Überblick über die Forschungsergebnisse	79
8.2 Zusammenfassung der Innovationspunkte	79
8.3 Analyse der Anwendungsperspektiven.....	80
9 Ausblick	81
9.1 Vorschläge zur Systemverbesserung	81
9.2 Zukünftige Forschungsrichtungen	81
Literatur	I
Komponentenliste	IV
Leiterplatten	V
Codeausschnitte	VII
Bedienungsanleitung	VIII
Erklärung zur Codeverfügbarkeit	XIII
Selbstständigkeitserklärung	XIV

Abbildungsverzeichnis

Abbildung 1-1 Verletzungen am Brandort nach Verletzungsursache: Durchschnitt 2017-2021	1
Abbildung 3-1 Arbeiter mit Dräger X-am 5600	10
Abbildung 3-2 Sensor-Steuerplatine und Menschenmodell mit zusätzlichen Sensoren	11
Abbildung 5-1 Systemarchitektur	17
Abbildung 5-2 Modulfunktionen	18
Abbildung 5-3 Hardware-Architektur	19
Abbildung 5-4 Multitasking-Implementierung in FreeRTOS	20
Abbildung 5-5 Software-Ablauf	22
Abbildung 5-6 ESP32-C6-DevKitC-1-N8 Entwicklungskit	23
Abbildung 5-7 Detaillierte Spezifikationen des ESP32-C6-DevKitC-1	25
Abbildung 5-8 MiCS-4514 Modul	26
Abbildung 5-9 Ansprechcharakteristik der RED-Sensoreinheit für reduzierende Gase (Messbedingungen: 25°C, 50% relative Luftfeuchtigkeit)	27
Abbildung 5-10 Ansprechcharakteristik der OX-Sensoreinheit für oxidierende Gase (Messbedingungen: 25°C, 50% relative Luftfeuchtigkeit)	28
Abbildung 5-11 Waveshare 1,5-Zoll RGB OLED-Displaymodul	29
Abbildung 5-12 Aktiver Summer	30
Abbildung 5-13 Mikroschalter	30
Abbildung 5-14 MicroSD-Kartenslot-Modul	31
Abbildung 5-15 DS3231 RTC-Modul	31
Abbildung 5-16 LM2596 LM2596 Spannungsabsenkungsmodul	32
Abbildung 5-17 Spannungsteilermodul	33
Abbildung 5-18 Schaltplan des Detektionssystems	34
Abbildung 5-19 Schaltungstest auf dem Steckbrett	35
Abbildung 5-20 Mit CNC-Verfahren hergestellte Leiterplatte (links) und manuelle Installation von Durchkontaktierungen (rechts)	36
Abbildung 5-21 Ablaufdiagramm der Initialisierung	40
Abbildung 5-22 Hauptschleifenfunktion	41

Abbildung 5-23 Ablaufdiagramm der Datenerfassungs- und Verarbeitungstasks	41
Abbildung 5-24 Beispiel eines Tasks.....	42
Abbildung 5-25 Auslesen von ppm-Werten vom Sensor.....	42
Abbildung 5-26 WMA-Filteralgorithmus	43
Abbildung 5-27 WMA-Gewichtungsverteilung.....	44
Abbildung 5-28 Ablaufdiagramm des Gefahrstofferkennungsmoduls	44
Abbildung 5-29 Code zur Bestimmung von Gefahrstoffen	46
Abbildung 5-30 Alarmverarbeitungsfunktion	47
Abbildung 5-31 Schematische Darstellung der Tasten-Zustandsmaschine	48
Abbildung 6-1 Montiertes System.....	57
Abbildung 6-2 Manuelle Justierung des Step-Down-Modulausgangs	57
Abbildung 6-3 Stabilitätstest der Ausgangsspannung des Step-Down-Moduls	58
Abbildung 6-4 Partielle Funktionsdarstellung des ThingsBoard-Dashboards.....	60
Abbildung 6-5 Messung des Gesamtstromverbrauchs des Systems	61
Abbildung 6-6 Datenakquisition	64
Abbildung 6-7 Aus Rohdaten extrahierte Merkmale.....	65
Abbildung 6-8 Erwartete Leistungsdaten des Modells	66
Abbildung 6-9 Detailliertes Flussdiagramm der TinyML-Funktion	68

Tabellenverzeichnis

Tabelle 5-1 Vergleich der Eigenschaften verschiedener Mikrocontroller-Plattformen...	24
Tabelle 5-2 Vergleichende Analyse der Eigenschaften verschiedener Gassensoren...	26
Tabelle 5-3 Detektionsbereich des MiCS-4514 Sensors.....	28
Tabelle 5-4 Technische Spezifikationen des Waveshare 1,5-Zoll RGB OLED-Displaymoduls.....	29
Tabelle 5-5 Signalpin-Zuweisungen für verschiedene Module.....	35
Tabelle 5-6 Messbereiche einiger Gase für MiCS-4514 (Teil 1)	45
Tabelle 5-7 Messbereiche einiger Gase für MiCS-4514 (Teil 2)	45
Tabelle 7-1 Auflistung der für das Experiment benötigten Materialien und Ausrüstungen	74

Abkürzungsverzeichnis

ADC	Analog-to-Digital Converter
CNC	Computer Numerical Control
EON	Edge Optimized Neural
GPIO	General-Purpose IO ports
GPS	Global Positioning System
I2C	Inter-Integrated Circuit
IDLH	Immediately Dangerous to Life or Health
IoT	Internet of Things
LED	Light-emitting diode
LEL	Lower Explosion Limit
LXC	Linux Containers
MEMS	Micro-Electro-Mechanical Systems
ML	Machine Learning
MOS	Metal-Oxide Semiconductor
MQTT	Message Queuing Telemetry Transport
NDIR	Nondispersive Infrared
NIOSH	National Institute for Occupational Safety and Health
NTP	Network Time Protocol
OLED	Organic Light-Emitting Diode
OPFRs	Organophosphate flame retardants
OSHA	Occupational Safety and Health Administration
PAS	Photoacoustic spectroscopy
PBDEs	Polybrominated diphenyl ethers
PC	Personal computer

PCB	Printed Circuit Board
PEL	Permissible Exposure Limit
PID	Photoionization detector
PSA	Persönliche Schutzausrüstung
PVC	Polyvinyl chloride
QCL	Quantum Cascade Laser
RAM	Random Access Memory
REL	Recommended Exposure Limit
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
ROM	Read-Only Memory
RTOS	Real-Time Operating System
RTC	Real-Time Clock
SEPIC	Single ended primary inductance converter
SMA	Simple Moving Average
SPI	Serial Peripheral Interface
TDLAS	Tunable diode laser absorption spectroscopy
THT	Through Hole Technology
TWA	Time-weighted average
UEL	Upper Explosion
USB	Universal Serial Bus
WMA	Weighted Moving Average
VOCs	Volatile Organic Compounds

1 Einleitung

Die fortschreitende globale Industrialisierung und Urbanisierung stellen Arbeitnehmer vor beispiellose Herausforderungen hinsichtlich ihrer beruflichen Sicherheit. Insbesondere in Hochrisiko-Branchen wie der chemischen, biologischen und pharmazeutischen Industrie sind Arbeitskräfte häufig potenziell gesundheitsschädlichen chemischen Substanzen ausgesetzt, was eine erhebliche Bedrohung für ihre Gesundheit und Sicherheit darstellt.[1,2] Unter den zahlreichen Hochrisiko-Berufen stechen die Gefahren, denen Feuerwehrleute ausgesetzt sind, besonders hervor.[3] Bei der Ausübung ihrer Pflichten befinden sich Feuerwehrleute oft in komplexen und dynamischen Gefahrensituationen. Statistische Daten von 2017 bis 2021 (siehe Abbildung 1-1) zeigen, dass der Kontakt mit brandortbezogenen Faktoren wie hohen Temperaturen, Rauch oder giftigen Substanzen eine der Hauptursachen für Verletzungen bei Feuerwehrleuten darstellt und etwa 39% aller Verletzungen am Brandort ausmacht.[4] Diese Zahlen unterstreichen die Dringlichkeit, die Sicherheitsschutzmaßnahmen für Feuerwehrleute zu verbessern.

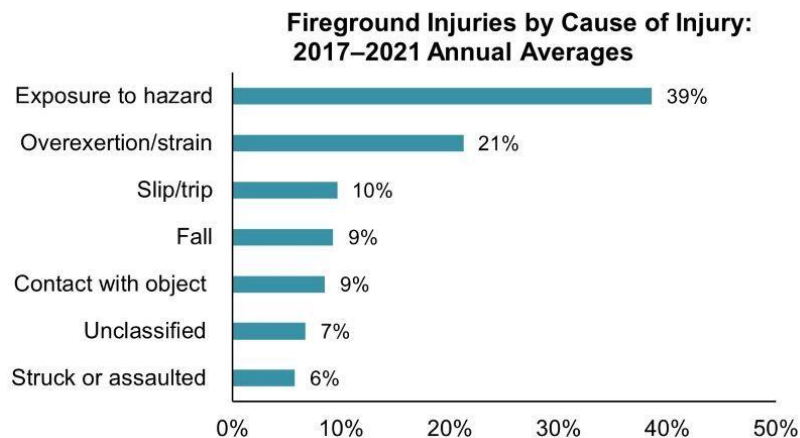


Abbildung 1-1 Verletzungen am Brandort nach Verletzungsursache: Durchschnitt 2017-2021

Hinweis: [Online-Grafik] Entnommen aus "Firefighter Injuries on the Fireground", von R. Campbell, 2023, NFPA. (<https://www.nfpa.org/education-and-research/research/nfpa-research/fire-statistical-reports/patterns-of-firefighter-fire-ground-injuries>) [abgerufen am 26.04.2024].

Feuerwehrleute, die an Löschaktionen bei Industriebränden in der chemischen, biologischen und pharmazeutischen Industrie beteiligt sind, sind noch gravierenderen Risiken ausgesetzt. Diese Umgebungen bergen nicht nur die üblichen Brandgefahren, sondern können auch komplexe chemische Reaktionen, die Freisetzung schädlicher Gase und potenzielle Explosionsrisiken umfassen. Studien belegen, dass bei chemischen Unfällen Ersthelfer wie Feuerwehrleute und Polizisten am häufigsten durch den Kontakt mit toxischen Substanzen verletzt werden.[5] Bemerkenswert ist, dass selbst bei gewöhnlichen Wohnungsbränden die bei der Verbrennung moderner Baumaterialien und Möbel freigesetzten schädlichen Gase eine Bedrohung für die langfristige Gesundheit von Feuerwehrleuten darstellen.[6] Langfristige Exposition gegenüber diesen gefährlichen Substanzen kann das Risiko für Feuerwehrleute, an Herz-Kreislauf-, Atemwegs- oder Tumorerkrankungen zu erkranken, signifikant erhöhen.[7]

Angesichts dieser Herausforderungen wird die PSA zu einem Schlüsselement zum Schutz der Sicherheit von Arbeitnehmern, insbesondere von Feuerwehrleuten.[8] Allerdings bietet herkömmliche PSA hauptsächlich passive physische Barrieren und ermöglicht keine Echtzeit-Überwachung und Warnung vor schädlichen Substanzen und Gasen in der Umgebung. Diese passive Schutzweise erweist sich als besonders unzureichend bei der Bewältigung plötzlicher

Freisetzungen schädlicher Gase oder unerwarteter Unfälle. Um die Sicherheit der Arbeitnehmer zu gewährleisten, wird die Entwicklung intelligenter PSA, die gefährliche Substanzen in der Umgebung in Echtzeit erkennen kann, unerlässlich. Trotz Fortschritten in der Genauigkeit bestehender Überwachungsgeräte gibt es generell noch Herausforderungen wie langsame Reaktionszeiten, große Abmessungen und Anfälligkeit für Umgebungsstörungen.

In den letzten Jahren haben die rasanten Entwicklungen in den Bereichen Sensortechnologie, IoT und Datenverarbeitung neue Möglichkeiten zur Lösung dieser Problematik eröffnet. Insbesondere im Bereich der tragbaren Geräte haben Fortschritte bei intelligenten Sensoren und integrierten Schaltkreisen die Integration fortschrittlicher Überwachungstechnologien in PSA realisierbar gemacht. Die Weiterentwicklung der drahtlosen Kommunikationstechnologie hat zudem neue Wege für Echtzeitdatenübertragung und Fernüberwachung erschlossen.

Die vorliegende Studie zielt darauf ab, ein innovatives Detektionssystem für gefährliche Substanzen zu entwickeln, das miniaturisierte Sensoren und drahtlose Kommunikationsfunktionen integriert. Dieses System soll ein wesentlicher Bestandteil der PSA für Feuerwehrleute und andere Arbeitskräfte in Hochrisiko-Bereichen sein, die potenziell schädlichen Gasen ausgesetzt sind. Das Forschungsziel ist es, die Limitationen bestehender Überwachungsgeräte hinsichtlich Reaktionsgeschwindigkeit, Größe und Umgebungsanpassungsfähigkeit zu überwinden und eine miniaturisierte, hochempfindliche und energieeffiziente Echtzeit-Überwachungslösung bereitzustellen.

Die Innovation dieser Forschungsarbeit liegt in der Synergie fortschrittlicher Sensortechnologie mit dem Design tragbarer Geräte. Diese Kombination kann nicht nur Arbeitnehmern unmittelbares Sicherheitsfeedback liefern, um rechtzeitig Präventivmaßnahmen zu ergreifen, sondern auch dem Managementpersonal effektive Überwachungsinstrumente zur Verfügung stellen, um Notfälle zu antizipieren und adäquat darauf zu reagieren. Durch diesen Ansatz wird eine substantielle Verbesserung des Sicherheitsniveaus von Arbeitnehmern in Gefahrenzonen erwartet, was einen bedeutenden Fortschritt im Bereich der Arbeitssicherheit darstellt.

2 Aufgabenstellung und These

Dieses Kapitel expliziert die Kernaufgaben und Forschungsfragen der vorliegenden Studie. Aufbauend auf dem im ersten Kapitel dargelegten Hintergrund und den identifizierten Herausforderungen werden die Forschungsziele präzise definiert und die antizipierten Lösungsansätze skizziert. Dies bildet das Fundament für die nachfolgende Forschungs- und Entwicklungsarbeit und bietet zugleich einen strukturellen Rahmen für die gesamte Abhandlung.

2.1 Aufgabenspektrum

Die zentrale Aufgabe dieser Studie besteht in der Entwicklung einer innovativen Methode zur Detektion gefährlicher Substanzen, die in PSA integrierbar ist. Diese soll eine Echtzeit-Erkennung gefährlicher Stoffe und Gase in der Umgebung ermöglichen, wobei ein besonderer Fokus auf den Anforderungen von Hochrisiko-Berufen, insbesondere dem Feuerwehrdienst, liegt. Das System zielt darauf ab, die Limitationen bestehender PSA bei der Erkennung gefährlicher Substanzen zu überwinden und das Sicherheitsniveau von Einsatzkräften in Gefahrenzonen signifikant zu erhöhen.

Das Aufgabenspektrum umfasst folgende Schwerpunkte:

- Systematische Analyse der Charakteristika und Leistungsparameter existierender portabler Detektionsgeräte oder -technologien zur Erkennung gefährlicher Substanzen, mit besonderem Augenmerk auf ihre Performanz in Brandszenarien.
- Erforschung und Selektion avancierter Sensortechnologien, die sich optimal zur Integration in PSA eignen.
- Konzeption eines Detektionssystems, das nahtlos in bestehende PSA integriert werden kann.
- Entwicklung Algorithmen zur Echtzeit-Datenverarbeitung und -analyse zur präzisen Identifizierung gefährlicher Substanzen im Umfeld.
- Implementierung eines robusten drahtlosen Kommunikationssystems zur Echtzeit-Übertragung von Detektionsdaten und Ermöglichung einer effektiven Fernüberwachung.
- Gestaltung einer intuitiven Benutzerschnittstelle zur mühelosen Interpretation der Detektionsergebnisse.
- Durchführung umfassender Systemtests zur rigorosen Evaluation der Leistungsfähigkeit und Zuverlässigkeit.

2.2 Antizipierte Lösungsansätze

Zum Abschluss der Forschungsarbeit wird die Entwicklung eines in PSA integrierten Detektionssystems für gefährliche Substanzen anvisiert, das das Sicherheitsniveau von Einsatzkräften in Gefahrenzonen substanziell erhöht. Das System soll folgende Schlüsselmerkmale aufweisen:

- Echtzeit-Detektionsfähigkeit: Präzise Erkennung und Identifizierung eines breiten Spektrums gefährlicher Gase und chemischer Substanzen.

-
- Miniaturisiertes Design: Einsatz hochintegrierter Mikrosensoren, die nahtlos in Standard-PSA implementiert werden können.
 - Energieeffizientes Design: Optimierung des Gesamtenergieverbrauchs des Systems bei gleichzeitiger Gewährleistung robuster Grundfunktionalitäten.
 - Drahtlose Kommunikation: Realisierung einer zuverlässigen Echtzeit-Datenübertragung.
 - Intuitive Benutzerschnittstelle und effektives Alarmsystem: Übersichtliche Visualisierung der Detektionsergebnisse und prompte Alarmierung in Gefahrensituationen.
 - Umgebungsresistenz: Gewährleistung einer stabilen Funktionalität auch unter widrigen Umgebungsbedingungen.
 - Modulares Design: Ermöglicht eine unkomplizierte Wartung und Aufrüstung, mit der Möglichkeit zur branchenspezifischen Anpassung.

3 Stand der Technik

Das vorliegende Kapitel bietet einen umfassenden Überblick über den aktuellen Stand der Technik im Bereich intelligenter PSA und Gefahrstoffdetektion. Es legt damit die Grundlage für die nachfolgende Systemkonzeption und -entwicklung. Der Inhalt umfasst eine detaillierte Darstellung der PSA, eine fundierte Analyse der Bedürfnisse von Feuerwehreinsatzkräften, eine kritische Bewertung existierender Gefahrstoffdetektionssysteme sowie eine Betrachtung technologischer Entwicklungstrends. Ziel ist es, einen präzisen technischen Hintergrund zu vermitteln und als Leitfaden für innovative Lösungsansätze in den folgenden Kapiteln zu dienen.

Um die Kernaussagen und zentralen Konzepte hervorzuheben, wurde der Inhalt dieses Kapitels sorgfältig aufbereitet. Für Leser, die an detaillierteren Informationen interessiert sind, steht eine erweiterte Version als Anlage zur Verfügung. Diese befindet sich in der Datei "Ergänzende Lektüre zum dritten Kapitel der Abschlussarbeit.docx" im Projektordner und enthält tiefere technische Analysen sowie umfangreiche Hintergrundinformationen.

3.1 Übersicht über PSA

Die PSA ist von essenzieller Bedeutung für den Schutz von Arbeitskräften vor Gefahren in ihrer Arbeitsumgebung. Im Feuerwehrwesen kommt der PSA eine besonders herausragende Rolle zu. Sie umfasst primär Feuerschutzanzüge, Atemschutzgeräte, Schutzhelme, Handschuhe und Sicherheitstiefel. Diese Ausrüstung gewährleistet Feuerwehreinsatzkräften einen umfassenden körperlichen Schutz gegen die vielfältigen Gefahren wie extreme Hitze, dichten Rauch und toxische Gase. Allerdings bietet herkömmliche PSA überwiegend nur passiven Schutz und es mangelt an Funktionen zur Echtzeitüberwachung und -warnung. Daher zeichnet sich als zukunftsweisender Trend die Integration moderner Sensortechnologien und intelligenter Systeme in die PSA ab.

Atemschutzgeräte spielen eine zentrale Rolle beim Schutz von Feuerwehreinsatzkräften in Umgebungen mit Rauch und giftigen Gasen. Gängige Atemschutzsysteme umfassen Filtergeräte und SCBA. SCBA-Systeme können Einsatzkräften reine Atemluft zur Verfügung stellen und so ihre Sicherheit in toxischen Umgebungen gewährleisten.[9] Die Forschung und Entwicklung zur Integration von Gefahrstoffdetektionssystemen in SCBA-Geräte birgt ein erhebliches Potenzial und eröffnet vielversprechende Perspektiven für die Zukunft.

3.2 Interviews mit Feuerwehreinsatzkräften

Die untersuchte Feuerweereinheit setzt gegenwärtig tragbare Multigasdetektoren ein, die nach dem Pumpansaugprinzip funktionieren. Diese Methodik ermöglicht jedoch keine Echtzeitüberwachung und -alarmierung, was die Reaktionszeit und folglich die Sicherheit der Einsatzkräfte erheblich beeinträchtigt.

Durch eingehende Interviews mit einer spezialisierten Feuerweereinheit wurden folgende Schlüsselbedürfnisse identifiziert:

- Echtzeitüberwachung und -alarmierung: Bestehende Handgeräte können den Anforderungen an Echtzeitfähigkeit nicht gerecht werden.
- Intuitive Bedienung und Rückmeldung: Die Geräte sollten eine einfache Handhabung ermöglichen und unmittelbar erkennbare Rückmeldungen liefern.
- Schnelle Einsatzbereitschaft: Die Ausrüstung muss binnen kürzester Zeit einsatzfähig sein.
- Hoher Integrationsgrad: Beeinträchtigungen der Bewegungsfreiheit der Einsatzkräfte sind zu minimieren.
- Robustheit und Langlebigkeit: Die Geräte müssen den extremen Einsatzbedingungen standhalten können.
- Wartungsfreundlichkeit: Eine einfache Instandhaltung zur Sicherstellung der langfristigen Einsatzfähigkeit ist erforderlich.
- Kosteneffizienz: Sowohl Anschaffungs- als auch Wartungskosten sind zu berücksichtigen.

Diese Designanforderungen zielen darauf ab, die Praktikabilität und Effizienz der Ausrüstung zu optimieren und dabei gleichzeitig die Sicherheit und Bewegungsfreiheit der Einsatzkräfte zu gewährleisten. Sie bilden somit eine essenzielle Grundlage für die nachfolgende Systemkonzeption.

3.3 Analyse bestehender Gefahrstoffdetektionssysteme

Gefahrstoffdetektionssysteme nehmen eine Schlüsselrolle bei Brandeinsätzen und in anderen Gefahrensituationen ein und tragen maßgeblich zum Schutz der Einsatzkräfte bei. Der aktuelle Markt bietet eine Vielzahl von Detektionstechnologien und -geräten, jeweils mit spezifischen Charakteristika und Einsatzbereichen. Dieser Abschnitt widmet sich der Untersuchung häufig auftretender Gefahrstoffe bei Bränden, analysiert gängige Detektionstechnologien und evaluiert die Leistungsfähigkeit sowie Anwendungsmöglichkeiten diverser tragbarer und integrierter Detektionsgeräte. Ziel ist es, ein umfassendes Verständnis der Merkmale, Vorteile und Limitationen bestehender Systeme zu erlangen. Diese detaillierte Analyse soll wertvolle Erkenntnisse und Orientierung für die Systemkonzeption in der vorliegenden Studie liefern.

3.3.1 Häufige Gefahrstoffe bei Bränden

Die bei Bränden auftretenden Gefahrstoffe lassen sich primär in drei Kategorien einteilen: toxische Gase, brennbare Stoffe und explosive Substanzen. Jede dieser Kategorien stellt spezifische Risiken für die Gesundheit und Sicherheit der Einsatzkräfte dar.

3.3.1.1 Toxische Gase

Toxische Gase zählen zu den häufigsten und gefährlichsten Stoffen bei Bränden.[10] Sie entstehen vorwiegend durch unvollständige Verbrennung von Materialien oder werden direkt aus bestimmten Chemikalien freigesetzt.[11,12] Zu den häufig auftretenden toxischen Gasen gehören:

- Kohlenmonoxid (CO): Ein farb- und geruchloses Gas, das zu den Haupttodesursachen bei Bränden zählt.[13,14]
- Chlorwasserstoff (HCl): Entsteht bei der Verbrennung chlorhaltiger Materialien wie PVC und wirkt stark reizend auf die Atemwege.
- Cyanwasserstoff (HCN): Ein hochgiftiges Gas, das bei der Verbrennung stickstoffhaltiger Materialien freigesetzt wird und die zelluläre Atmung hemmen kann.
- Stickstoffdioxid (NO₂): Entsteht bei der Verbrennung fossiler Brennstoffe und hat eine stark reizende Wirkung auf die Atemwege.
- Kohlendioxid (CO₂): Wird hauptsächlich durch die Verbrennung organischer Stoffe und Atmungsprozesse freigesetzt. Hohe Konzentrationen können zu Erstickung führen.
- Benzol: Ein bekanntes Karzinogen, das bei der Verbrennung kohlenstoffhaltiger Materialien freigesetzt wird.
- Flüchtige organische Verbindungen (VOCs): Entstehen bei der Verbrennung verschiedener Materialien, insbesondere Kunststoffe, und weisen toxische sowie karzinogene Eigenschaften auf.

3.3.1.2 Brennbare Stoffe

Die Kategorie der brennbaren Stoffe umfasst brennbare Feststoffe, Flüssigkeiten und Gase [15, 16]:

- Brennbare Feststoffe: Hierzu zählen Materialien wie Holz, Baumwolle, Papier und diverse Kunststoffe.
- Brennbare Flüssigkeiten: Dazu gehören hochentzündliche Substanzen wie Benzin, Alkohol und Diethylether.
- Brennbare Gase: Beispiele sind Wasserstoff, Methan und Propan, die in bestimmten Konzentrationsbereichen explosionsfähige Gemische bilden können.

3.3.1.3 Explosive Stoffe

Explosive Stoffe zeichnen sich dadurch aus, dass sie unter Hitzeeinwirkung oder mechanischem Schock heftige chemische Reaktionen auslösen können:

- Explosive Chemikalien: Hierunter fallen verschiedene Arten von Sprengstoffen.[17]
- Selbstentzündliche Materialien: Dazu gehören Substanzen wie Phosphor oder bestimmte Metallpulver, die sich bei Kontakt mit Luft oder Wasser spontan entzünden können.[18]

3.3.1.4 Weitere Gefahrstoffe

Neben den genannten Hauptkategorien müssen auch spezifische chemische Gefahren berücksichtigt werden. Hierzu zählen radioaktive Substanzen sowie toxische Materialien wie PBDEs und

OPFRs, die bei Bränden besondere Risiken darstellen und spezielle Schutzmaßnahmen erfordern.

3.3.2 Gängige Detektionstechnologien

Die Technologien zur Detektion gefährlicher Substanzen an Brandorten lassen sich hauptsächlich in zwei Kategorien einteilen: kontaktbasierte und kontaktlose Reaktionsmethoden.

3.3.2.1 Kontaktbasierte Gassensoren

1. Metalloxid-Halbleiter (MOS) Sensoren:
 - Prinzip: Basiert auf Änderungen der elektrischen Leitfähigkeit, die durch die Wechselwirkung zwischen Gas und der Oberfläche des Halbleitermaterials hervorgerufen werden.[19]
 - Eigenschaften: Kostengünstig, kompakt, empfindlich für verschiedene Gase, jedoch geringe Selektivität und anfällig für Umgebungseinflüsse.
2. Katalytische Verbrennungssensoren:
 - Prinzip: Nutzt die bei der Verbrennung brennbarer Gase an einer katalytischen Oberfläche entstehende Wärme zur Bestimmung der Gaskonzentration.[20]
 - Eigenschaften: Einfacher Aufbau, schnelle Reaktion, geeignet für brennbare Gase, benötigt jedoch die Anwesenheit von Sauerstoff.
3. Elektrochemische Sensoren:
 - Prinzip: Basiert auf elektrischen Strömen, die durch Redoxreaktionen zwischen Gas und Elektroden erzeugt werden.[21]
 - Eigenschaften: Hohe Selektivität, niedriger Energieverbrauch, geeignet für die Detektion toxischer Gase, kann jedoch regelmäßige Kalibrierung erfordern.

3.3.2.2 Kontaktlose Gassensoren

1. PID:
 - Prinzip: Diese Detektoren nutzen hochenergetische UV-Strahlung zur Ionisierung von Gasmolekülen, wodurch ein messbarer Stromfluss entsteht.[22, 23]
 - Eigenschaften: PIDs zeichnen sich durch hohe Empfindlichkeit und schnelle Ansprechzeit aus. Sie eignen sich besonders gut für die Detektion VOCs, können jedoch nicht zwischen spezifischen chemischen Substanzen unterscheiden.
2. NDIR-Technologie:
 - Prinzip: Diese Technologie basiert auf den spezifischen Absorptionseigenschaften von Gasmolekülen für bestimmte Infrarot-Wellenlängen.[24]
 - Eigenschaften: NDIR-Sensoren bieten eine hohe Selektivität und gute Langzeitstabilität. Sie sind besonders geeignet für die Detektion von Gasen wie CO und CO₂.

3. TDLAS:

- Prinzip: Bei dieser Methode wird ein durchstimmbarer Laser verwendet, um einen spezifischen Wellenlängenbereich abzutasten und Gasabsorptionslinien zu detektieren.[25, 26]
- Eigenschaften: TDLAS zeichnet sich durch hohe Selektivität und Empfindlichkeit aus. Diese Technologie ist besonders nützlich für die Identifizierung und Quantifizierung spezifischer Gase in komplexen Gasgemischen.

4. QCL-Technologie:

- Prinzip: QCLs nutzen Laser im mittleren Infrarot- bis Terahertz-Bereich zur Detektion verschiedener Gasmoleküle.[27]
- Eigenschaften: Diese Technologie bietet eine hohe Ausgangsleistung und einen breiten Wellenlängenbereich, was sie für die Detektion verschiedener Gase geeignet macht. Allerdings ist sie vergleichsweise kostenintensiv.

5. PAS:

- Prinzip: PAS basiert auf dem photoakustischen Effekt, bei dem absorbierte Lichtenergie in Schallwellen umgewandelt wird.[28]
- Eigenschaften: Diese Methode zeichnet sich durch hohe Empfindlichkeit und einen großen dynamischen Bereich aus. Sie ist unempfindlich gegenüber Rauch und Staub, kann jedoch durch Umgebungsgeräusche beeinträchtigt werden.

3.3.3 Tragbare Gasdetektionsgeräte

Tragbare Gasdetektionsgeräte spielen eine entscheidende Rolle bei der Sicherheit in gefährlichen Umgebungen. Dieser Abschnitt gibt einen Überblick über gängige Typen solcher Geräte, stellt typische Produkte und ihre Merkmale vor und beleuchtet die Unzulänglichkeiten bestehender Technologien.

Tragbare Gasdetektionsgeräte lassen sich hauptsächlich in drei Kategorien einteilen: Diffusions-, Pumpen- und Ferndetektionstypen. Diffusionsgeräte nutzen das Prinzip der natürlichen Gasdiffusion und zeichnen sich durch eine einfache Struktur aus. Pumpengeräte saugen Gasproben aktiv an und bieten eine schnelle Reaktionszeit. Ferndetektionsgeräte verwenden kontaktlose Technologien und ermöglichen Messungen aus sicherer Entfernung. Je nach Anzahl der gleichzeitig überwachten Gase unterscheidet man zudem zwischen Eingas- und Mehrgasdetektoren.

Auf dem Markt finden sich verschiedene typische Produkte, darunter der Teledyne GT-Fire, Teledyne PS-200, die Dräger Pac 6000-Serie und der Sewerin SR-LD 800. Jedes dieser Geräte hat seine spezifischen Merkmale: Der GT-Fire wurde speziell für die Anforderungen der Feuerwehr konzipiert, der PS-200 besticht durch seine flexible Struktur, die Pac 6000-Serie zeichnet sich durch ihre einfache Tragbarkeit aus, und der SR-LD 800 ermöglicht die Ferndetektion von Methan.

Die Mehrheit der tragbaren Gasdetektionsgeräte verfügt über fortschrittliche Funktionen wie akustische und optische Alarmsysteme, modulare Bauweise sowie Wasser- und Staubschutz. Hochwertige Modelle bieten darüber hinaus oft Datenaufzeichnungs- und Analysefunktionen.

Trotz dieser Fortschritte gibt es verhältnismäßig wenige Produkte, die spezifisch auf die besonderen Bedürfnisse der Feuerwehr zugeschnitten sind. Dies deutet auf ein erhebliches Innovationspotenzial in diesem Bereich hin. Die Hauptmängel der gegenwärtigen Technologien umfassen:

- Unzureichende Integration mit anderen Komponenten der Feuerwehrausrüstung
- Begrenzte Anpassungsfähigkeit der Hardware an verschiedene Einsatzszenarien
- Mangel an intelligenten Analyse- und Entscheidungsunterstützungsfunktionen für komplexe Brandsituationen
- Verbesserungsbedürftige Zuverlässigkeit unter extremen Bedingungen (z.B. sehr hohe Temperaturen, hohe Luftfeuchtigkeit, dichter Rauch)
- Benutzeroberflächen, die den Anforderungen an eine intuitive Bedienung in Notfallsituationen nicht vollständig gerecht werden.

Zukünftige Designs sollten sich auf die Lösung dieser Probleme konzentrieren. Es sind nicht nur Durchbrüche in der Hardwareleistung erforderlich, sondern auch umfassende Innovationen in Softwarealgorithmen, Datenanalyse und Benutzeroberflächen-Design, um intelligenteren, zuverlässigeren und benutzerfreundlicheren Gasdetektionslösungen bereitzustellen.

3.3.4 Integrationsmethoden für Detektionsgeräte

In Zeiten des technologischen Fortschritts und steigender Sicherheitsanforderungen hat sich die Integration von Gefahrstoff- oder Gasdetektionsfunktionen in PSA als entscheidender Ansatz zur Verbesserung der Sicherheit von Einsatzkräften etabliert. Die Art und Weise, wie Detektionsgeräte integriert werden, basiert auf spezifischen Einsatzszenarien und beeinflusst maßgeblich deren Effektivität und Zuverlässigkeit. Aus diesem Grund ist die Integrationsmethodik zu einem Schlüsselfaktor in Forschung und Entwicklung avanciert.

Gegenwärtig lassen sich drei Hauptmethoden der Integration von Detektionsgeräten unterscheiden: externe Befestigung, interne Integration und vernetzte Integrationssysteme.

Die externe Befestigung stellt die traditionellste Methode dar. Hierbei wird das Detektionsgerät direkt an der Außenseite der PSA angebracht. Ein anschauliches Beispiel hierfür ist der Dräger X-am 5600 Multigasdetektor, der problemlos an der Arbeitskleidung getragen werden kann (Abbildung 3-1) und eine Echtzeitüberwachung gefährlicher Gase in der Umgebung ermöglicht.

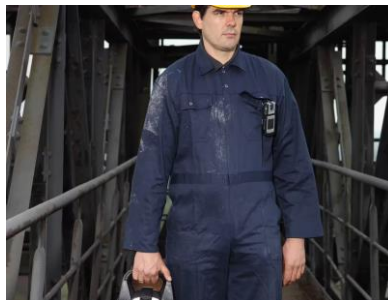


Abbildung 3-1 Arbeiter mit Dräger X-am 5600

Hinweis: [Online-Bild] Quelle: Dräger Website (https://www.draeger.com/de_de/Products/X-am-5600) [abgerufen am 19.07.2024].

Bei der internen Integration werden die Detektionselemente direkt in die Schutzausrüstung eingebettet. Ein innovatives Beispiel hierfür lieferten Kah et al. mit der Entwicklung eines drahtlosen Echtzeit-Gasdetektors, der aus 8 Gassensoren und einer Steuerplatine besteht und direkt in die PSA integriert wird (Abbildung 3-2). Dieser Ansatz steigert nicht nur die Detektionseffizienz, sondern adressiert auch potenzielle Kontaminationsprobleme, die bei traditionellen passiven Adsorptionsmessmethoden auftreten können.[29]

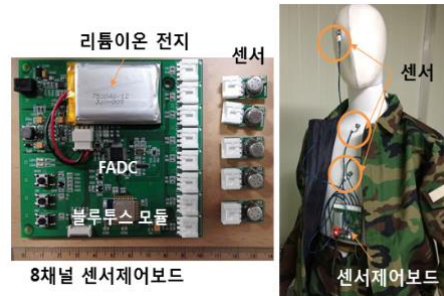


Abbildung 3-2 Sensor-Steuerplatine und Menschenmodell mit zusätzlichen Sensoren

Hinweis: [Bild] Entnommen aus "Development of wireless real-time gas detector system for chemical protection performance test of personal protective equipment", von D.-H. Kah, 2020, Journal of the Korea Institute of Military Science and Technology, 23(3), S.297.

Vernetzte Integrationssysteme repräsentieren den neuesten Stand der Technik, indem sie persönliche Detektionsgeräte mit umfassenderen Überwachungsnetzwerken verbinden. Zhang et al. präsentierten einen wegweisenden Ansatz zur Implementierung eines spezialisierten Überwachungssystems für toxische und schädliche Gase im Bereich der Brandbekämpfung und Rettung. Dieses System nutzt vor Ort installierte temporäre Hard- und Softwarekomponenten, integriert diverse Datenquellen und implementiert ein auf einem RNN-Modell basierendes dynamisches Frühwarnsystem.[30]

Diese Forschungs- und Anwendungsbeispiele verdeutlichen die rasante Entwicklung im Bereich der Integrationsmethoden für Gefahrstoffdetektionssysteme. Zukünftig könnte die synergetische Kombination verschiedener Integrationsmethoden den Einsatzkräften einen noch umfassenderen und zuverlässigeren Schutz bieten. Trotz dieser vielversprechenden Fortschritte bleiben einige Herausforderungen bestehen. Dazu zählen insbesondere die Verbesserung der Haltbarkeit der Geräte, die Optimierung der Echtzeit-Datenerfassung und -genauigkeit sowie die nahtlose Koordination mit anderen PSA-Funktionen. Diese Aspekte stellen wichtige Forschungsfelder für die Zukunft dar und werden maßgeblich zur weiteren Verbesserung der Sicherheit von Einsatzkräften beitragen.

4 Aufgabenspezifikation

In diesem Kapitel wird auf Basis einer eingehenden Analyse bestehender Technologien und Nutzerbedürfnisse eine detaillierte Aufgabenspezifikation für das Schadstoffdetektionssystem erarbeitet. Es werden die funktionalen Anforderungen, Leistungsindikatoren und das Benutzeroberflächen-Design ausführlich dargelegt, um eine klare Richtung für die anschließende Systementwicklung vorzugeben. Der Fokus liegt dabei auf der Behebung der im dritten Kapitel identifizierten technischen Mängel und der Präsentation innovativer Lösungsansätze.

4.1 Systemfunktionen

Basierend auf der vorangegangenen Analyse der Bedürfnisse von Feuerwehrleuten und der Evaluation existierender Detektionstechnologien muss das in dieser Studie entwickelte Schadstoffdetektionssystem (im Folgenden als "System" bezeichnet) folgende Kernfunktionen erfüllen:

- Echtzeit-Multigasdetektion: Simultane Erfassung mehrerer häufig auftretender Gefahrstoffe, einschließlich toxischer Gase (wie CO, HCN, NO₂ etc.) und brennbarer Gase (wie CH₄, H₂, C₃H₈ etc.).
- Intelligentes Alarmsystem: Integration von Multi-Modus-Alarmfunktionen, die sowohl akustische als auch optische Warnungen umfassen. Die Alarmschwellen sollen individuell an die Benutzerbedürfnisse anpassbar sein.
- Datenaufzeichnung und -analyse: Automatisierte Erfassung von Detektionsdaten, einschließlich Gaskonzentrationen und Umgebungstemperaturveränderungen. Die Daten sollen für spätere Analysen exportierbar sein, um Unfalluntersuchungen und Trainingsoptimierungen zu unterstützen.
- Drahtlose Kommunikation: Der Client soll über die Fähigkeit zur bidirektionalen drahtlosen Kommunikation mit einem Server (z.B. Kommandozentrale) verfügen, um Daten in Echtzeit auszutauschen, Gefahreninformationen umgehend zu übermitteln und Anweisungen zu empfangen.
- Selbsttest beim Einschalten und regelmäßige Selbstdiagnose: Integration einer Selbsttestfunktion beim Systemstart sowie regelmäßige Überprüfung des Betriebszustands aller Sensoren.
- Energiemanagement: Implementierung eines stromsparenden Designs zur Maximierung der Batterielaufzeit, inklusive einer Batteriestandsanzeige und einer Warnung bei kritischem Batteriestand.
- Erfassung zusätzlicher Umgebungsparameter: Optionale Echtzeit-Aufzeichnung von Umgebungstemperatur und -feuchtigkeit zur Unterstützung weiterführender Analysen.

4.2 Leistungsindikatoren

Zur Gewährleistung, dass das System den realen Anforderungen von Feuerwehrleuten und anderen Einsatzkräften gerecht wird, wurden folgende Schlüsselleistungskennzahlen definiert:

Gasdetektion

- **Detektionsreichweite:** Das System soll prioritär gefährliche Substanzen in unmittelbarer Nähe zum menschlichen Körper erkennen. Berührungslose Sensoren müssen einen Erfassungsbereich von 1 m direkt vor dem Sensor aufweisen. Kontaktgassensoren sollen eine Detektionsreichweite von mindestens 5 cm erreichen. Bei stationären Detektionsknoten sind vor den Kontaktsensoren kleine Ventilatoren zu installieren, um schädliche Substanzen aus größerer Entfernung zu detektieren.
- **Detektionsgenauigkeit:** Für diverse Gasarten ist eine Detektionsgenauigkeit von $\pm 10\%$ des Messbereichsendwerts zu erreichen.
- **Ansprechzeit:** Die T90-Ansprechzeit (Zeit bis zum Erreichen von 90% des Endwertes) darf 10 Sekunden nicht überschreiten.

Anpassungsfähigkeit an die Arbeitsumgebung

- **Temperaturbereich:** -10°C bis $+60^{\circ}\text{C}$
- **Feuchtigkeitsbereich:** 15% bis 95% relative Luftfeuchtigkeit (nicht kondensierend)
- **Schutzart:** Das Endprodukt muss über eine Gummiarmierung verfügen und die Schutzart IP67 erreichen, um Staub- und Wasserdichtigkeit zu gewährleisten.

Akkulaufzeit

- Unter normalen Einsatzbedingungen muss das System eine kontinuierliche Betriebszeit von mindestens 8 Stunden gewährleisten, um eine Standardschicht abzudecken.

Gewicht und Größe

- Das Gesamtgewicht des Systems darf 250 g (einschließlich Akku) nicht überschreiten, bei einem Volumen von weniger als 300 cm^3 . Dies stellt sicher, dass keine signifikante zusätzliche Belastung für die Einsatzkräfte entsteht.

Kommunikationsreichweite

- Bei einem Gebäudebrand mit Kommunikationsknoten auf jeder Etage muss die Kommunikationsdistanz zwischen System und Knoten mindestens 100 Meter (im Freien) betragen.

Datenspeicherkapazität

- Das System muss in der Lage sein, kontinuierliche Überwachungsdaten von mindestens 7 Tagen zu speichern.

4.3 Benutzeroberfläche

In Anbetracht der Nutzungsanforderungen von Feuerwehrleuten in Hochstresssituationen folgt das Design der Benutzeroberfläche den Prinzipien der Einfachheit, Intuitivität und leichten Bedienbarkeit. Die spezifischen Designaspekte umfassen:

Display

- Einsatz eines kontrastreichen OLED-Displays zur Gewährleistung der klaren Ablesbarkeit von Informationen unter verschiedenen Lichtbedingungen.
- Die Bildschirmgröße ist zwischen 1,5 und 2,5 Zoll zu wählen, um eine optimale Balance zwischen Lesbarkeit und Tragbarkeit zu erreichen.

Informationslayout

- Der Hauptbildschirm zeigt simultan die Echtzeitkonzentrationen aller überwachten Gase sowie die Umgebungstemperatur an.
- Verwendung farbcodierter Symbole und Texte zur intuitiven Darstellung verschiedener Gefahrenstufen (Grün für Sicherheit, Gelb für Warnung, Rot für Gefahr).
- Permanente Anzeige des Akkustands und anderer relevanter System- und Softwarezustände am oberen Bildschirmrand.

Akustisches und visuelles Feedback

- Integration einer hochleuchtenden LED-Anzeige zur farblichen Indikation des aktuellen Systemstatus.
- Einbau eines Hochleistungssummers (>85dB) zur Sicherstellung der deutlichen Wahrnehmbarkeit von Alarmen auch in geräuschintensiven Umgebungen.

Bedienelemente

- Minimierung der Anzahl physischer Tasten, idealerweise auf drei beschränkt: eine Bestätigungstaste sowie zwei Tasten zur Anpassung von Einstellungswerten.
- Implementierung weiterer Funktionen durch Tastenkombinationen.

Menüstruktur

- Implementierung einer flachen, einfach hierarchischen Menüstruktur, die den Zugriff auf jede Funktion mit maximal drei Bedienschritten gewährleistet.

Anpassbarkeit

- Benutzerdefinierte Einstellungen: Bereitstellung einer intuitiven Einstellungsoberfläche für Benutzer oder Administratoren zur Anpassung von Alarmgrenzwerten, Anzeigeeinheiten und anderen Parametern, um verschiedene Einsatzszenarien und individuelle Präferenzen zu berücksichtigen.
- Szenario Auswahl: Implementierung einer benutzerfreundlichen Oberfläche zur schnellen Auswahl voreingestellter Szenarien (z.B. Düngemittelbetriebe, Möbelunternehmen). Das System passt automatisch die entsprechende Sensorkonfiguration und Alarmgrenzwerte an.

Lokale Datenvisualisierung

- Integration einer Trendfunktion (z.B. Balken- oder Liniendiagramme) zur Darstellung der Gaskonzentrationsänderungen über definierte Zeiträume, um Benutzern eine schnelle Beurteilung von Situationsänderungen zu ermöglichen.

4.4 Innovative Designkonzepte

Zur Adressierung der in Kapitel 3 identifizierten technischen Mängel werden folgende innovative Designkonzepte vorgeschlagen:

- **Modulares Sensorsystem:** Durch den Einsatz austauschbarer Sensormodule ermöglicht das System eine rasche Anpassung an diverse Brandszenarien. Dies überwindet die mangelnde Flexibilität herkömmlicher Detektionssysteme.
- **Szenario-Voreinstellungsfunktion:** Die Software inkorporiert multiple Szenario-Voreinstellungen, darunter Wohngebäude, Chemiebetriebe und Möbelfabriken. Jedes Szenario verfügt über spezifische Sensorkonfigurationen und Alarmgrenzwertbereiche. Dies erhöht die Präzision der Systemreaktion auf verschiedene Brandsituationen und optimiert die Zielgerichtetheit sowie Genauigkeit der Detektion.
- **Integriertes und eigenständiges Doppelmodus-Design:** Das System bietet die Flexibilität zur Integration in PSA, beispielsweise durch Platzierung von Sensoren in der SCBA zur Überwachung der PSA-internen Umgebung. Gleichzeitig kann es als eigenständiges Gerät fungieren, entweder extern an der PSA befestigt oder als stationärer Detektionsknoten, was die Einsatzflexibilität signifikant erhöht.
- **Cloud-basierte Datenanalyse und Echtzeit-Feedback:** Durch Echtzeitkommunikation mit Cloud-Servern ermöglicht das System nicht nur unmittelbare Gefahrenwarnungen, sondern nutzt auch Big-Data-Analysen zur Optimierung langfristiger Detektionsstrategien.

Diese detaillierte Funktionsplanung, die Festlegung präziser Leistungskennzahlen und die Integration innovativer Designkonzepte definieren eine klare Entwicklungsrichtung für das Schadstoffdetektionssystem. Die spezifizierten Anforderungen berücksichtigen sowohl die technische Machbarkeit als auch die realen Bedürfnisse und Nutzungserfahrungen von Einsatzkräften im Feld. Gleichzeitig adressieren sie effektiv die Hauptmängel existierender Technologien. Diese Anforderungen bilden das Fundament für den nachfolgenden Systementwicklungsprozess, der auf kontinuierliche Optimierung und Verbesserung abzielt. Das ultimative Ziel ist die Realisierung eines hocheffizienten, zuverlässigen, flexiblen und benutzerfreundlichen integrierten Detektionssystems, das den anspruchsvollen Anforderungen moderner Feuerwehreinsätze gerecht wird.

5 Umsetzung

Dieses Kapitel erläutert detailliert den Prozess der Systemimplementierung für die Integration der Schadstofferkennung in persönliche Schutzausrüstung (PSA). Basierend auf den in Kapitel 4 festgelegten funktionalen Anforderungen und Leistungsindikatoren wird die Implementierung des Systems aus drei Perspektiven beleuchtet: Gesamtdesign, Hardware-Design und Software-Design.

Das Gesamtdesign skizziert die Systemarchitektur, die Modulzuweisung sowie die Hardware- und Software-Struktur und legt damit das Fundament für die nachfolgende spezifische Implementierung. Im Hardware-Design wird der gesamte Prozess von der Anforderungsanalyse bis zum PCB-Design und zur Fertigung durchlaufen, um sicherzustellen, dass die physische Umsetzung des Systems den vorgegebenen Anforderungen entspricht. Das Software-Design erläutert ausführlich die Software-Architektur, die Algorithmenimplementierung und das Design der Benutzeroberfläche, um die Intelligenz und Mensch-Maschine-Interaktion des Systems zu realisieren.

Durch diese detaillierten Design- und Implementierungsprozesse wird angestrebt, ein effizientes, zuverlässiges und benutzerfreundliches integriertes Detektionssystem zu entwickeln, das den Sicherheitsanforderungen von Feuerwehrleuten in Gefahrensituationen gerecht wird.

5.1 Gesamtkonzept

Das Gesamtdesign des Systems bildet das Herzstück des gesamten Implementierungsprozesses und bestimmt maßgeblich die Architektur und Funktionalität des Systems. Dieser Abschnitt gibt einen Überblick über das Gesamtdesign des Systems, wobei der Fokus auf der Gesamtarchitektur, der Funktionsmodulaufteilung sowie der Hardware- und Software-Architektur liegt.

5.1.1 Gesamtarchitektur des Systems und Modulzuweisung

5.1.1.1 Gesamtarchitektur

Das System basiert auf einer für eingebettete Systeme typischen Schichtenarchitektur, die von unten nach oben aus der Hardware-Schicht und der Software-Schicht besteht. Die Software-Schicht lässt sich weiter in die Hardware-Treiber-Schicht, die Funktionsmodul-Schicht, die Geschäftslogik-Schicht und die Anwendungsschicht untergliedern (siehe Abbildung 5-1).

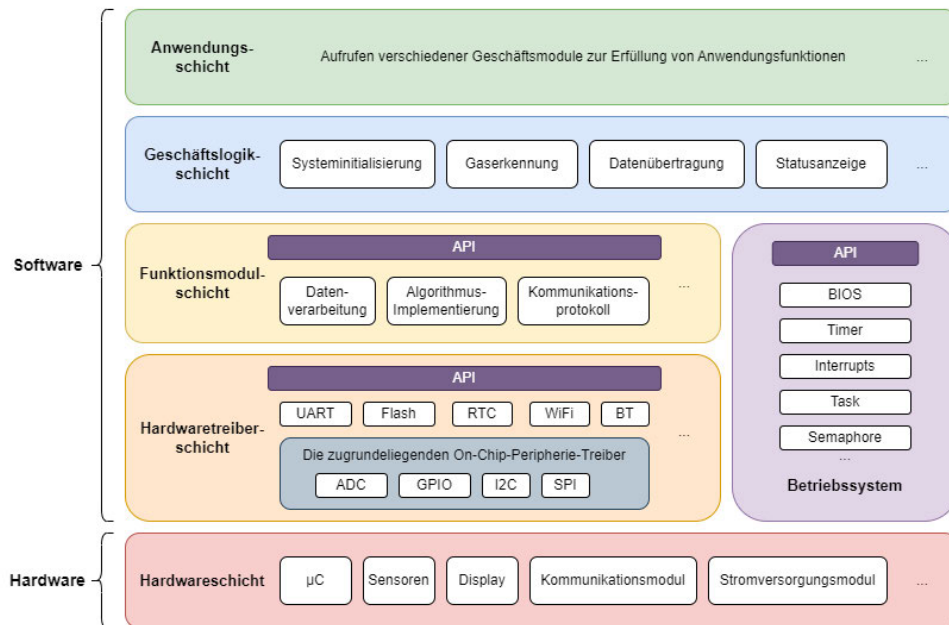


Abbildung 5-1 Systemarchitektur

- **Hardware-Schicht:** Die Basis des Systems, die alle physischen Hardwarekomponenten umfasst, wie Sensoren, Mikrocontroller, Kommunikationsmodule und Displays.
- **Hardware-Treiber-Schicht:** Baut auf der Hardware-Schicht auf und interagiert direkt mit der Hardware. Sie steuert und verwaltet die verschiedenen Hardwaremodule, einschließlich On-Chip- und Off-Chip-Peripherietreiber (z.B. Timer, ADC, GPIO, I2C, SPI) sowie andere Hardwaretreiber (z.B. UART, FLASH, RTC). Diese Schicht stellt standardisierte Schnittstellen für die darüberliegenden Anwendungen bereit.
- **Funktionsmodul-Schicht:** Diese Zwischenschicht beherbergt verschiedene Funktionsbibliotheken für Aufgaben wie Datenverarbeitung, Algorithmenimplementierung und Kommunikationsprotokolle.
- **Geschäftslogik-Schicht:** Hier wird die Kernfunktionalität des Systems implementiert, einschließlich Systeminitialisierung, Gaserkennung, Datenübertragung und Statusanzeige.
- **Anwendungsschicht:** Die oberste Ebene des Systems koordiniert und ruft verschiedene Geschäftsmodul auf, um vollständige Anwendungsfunktionen und Benutzerinteraktionen zu ermöglichen.

Die Kommunikation zwischen den Schichten erfolgt über standardisierte Schnittstellen, was die Flexibilität und Wartbarkeit des Systems erhöht. Der modulare Aufbau ermöglicht zudem die unabhängige Entwicklung und Prüfung der einzelnen Funktionsmodule, was sowohl die Entwicklungseffizienz als auch die Systemzuverlässigkeit steigert.

5.1.1.2 Moduluweisung

Basierend auf den in Kapitel 4 definierten Systemfunktionen lässt sich das System in neun Hauptfunktionsmodule unterteilen:

- **Datenerfassungsmodul:** Zuständig für die Erfassung der Sensordaten.

- Datenverarbeitungs- und Analysemodul: Berechnet Gaskonzentrationen und identifiziert gefährliche Substanzen.
- Alarmsteuerungsmodul: Beurteilt Alarmschwellenwerte, steuert akustische und optische Alarme und klassifiziert Alarmmeldungen.
- Benutzeroberflächen-Modul: Ermöglicht die direkte Interaktion mit dem Benutzer, zeigt Echtzeitdaten und Alarminformationen an, visualisiert den Systemstatus und ermöglicht die Befehlseingabe.
- Konfigurationsmanagement-Modul: Verantwortlich für die Einstellung von Alarmschwellenwerten, die Anpassung von Systemparametern und die Aktivierung bzw. Deaktivierung spezifischer Funktionen.
- Datenspeichermodul: Zeichnet Echtzeitdaten auf, verwaltet historische Daten und organisiert den Speicherplatz.
- Drahtloses Kommunikationsmodul: Implementiert das drahtlose Kommunikationsprotokoll und optional die verschlüsselte Datenübertragung.
- Systemselbsttest-Modul: Führt Selbsttests beim Systemstart und in regelmäßigen Abständen während des Betriebs durch.
- Energiemanagement-Modul: Überwacht den Batterieladezustand und gibt Warnungen bei niedrigem Batteriestand aus.

Abbildung 5-2 veranschaulicht die Funktionen dieser Module:

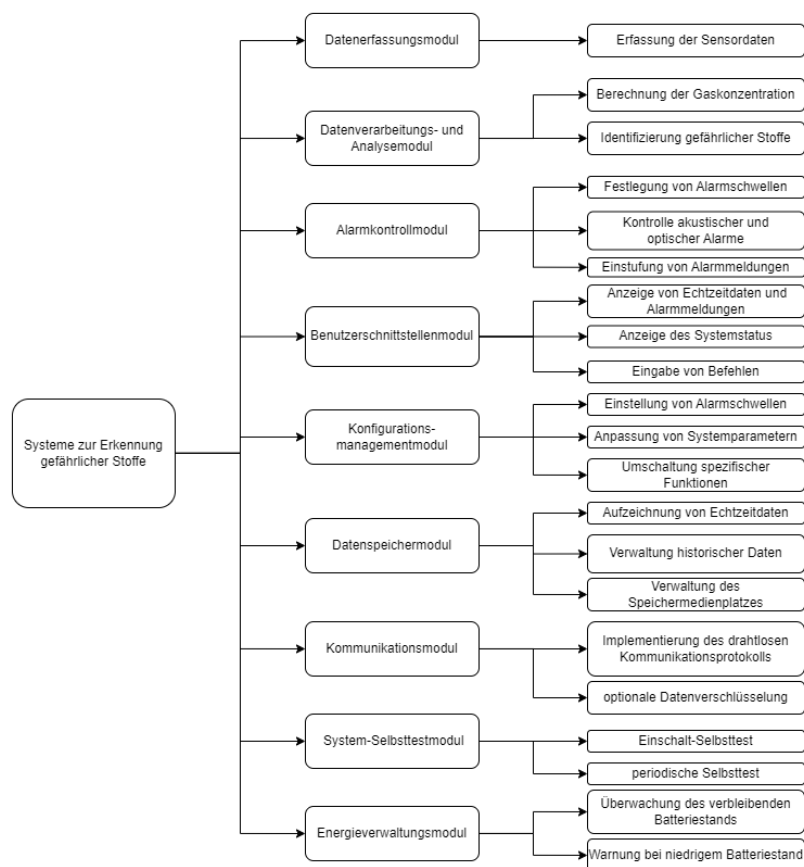


Abbildung 5-2 Modulfunktionen

5.1.2 Hardware-Architektur

Um alle in Kapitel 4 definierten Systemfunktionen zu realisieren, wurde eine Minimalsystem-Hardware-Architektur konzipiert. Diese Architektur umfasst folgende Hauptkomponenten:

- Mikrocontroller: Bildet das Herzstück des Systems und ist verantwortlich für Datenverarbeitung, Steuerung und Koordination der verschiedenen Module.
- Multigasensor oder Sensorarray: Dient der Erkennung verschiedener toxischer und brennbarer Gase.
- LED-Anzeigen und Summer: Geben visuelle und akustische Warnungen aus.
- OLED-Display: Zeigt Echtzeitdaten und den Systemstatus an.
- Tasten: Ermöglichen die Benutzerinteraktion mit dem System.
- Stromversorgung und Energiemanagement-Modul: Umfasst Batterie und Energiemanagement-Chip.
- Speicher: Dient der Datenaufzeichnung und Speicherung von Konfigurationsdateien.
- Drahtloses Kommunikationsmodul: Realisiert die Datenübertragung zur Einsatzzentrale.

Abbildung 5-3 illustriert die Hardware-Architektur des Systems und die entsprechenden Datenflüsse:

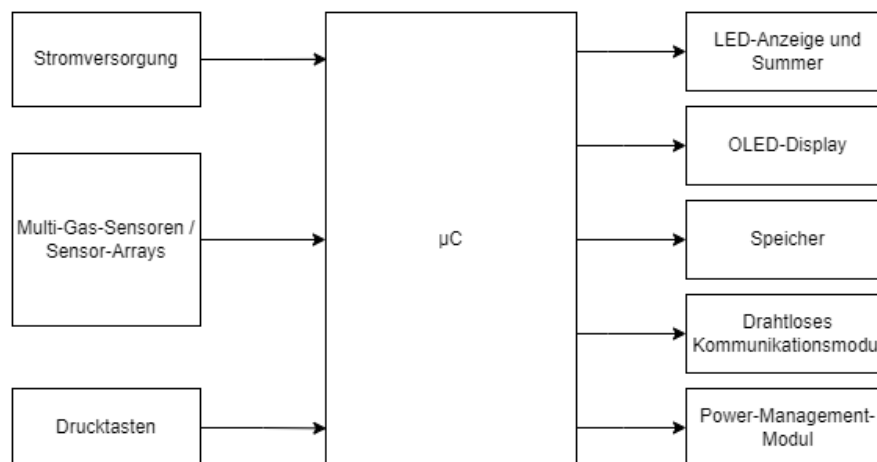


Abbildung 5-3 Hardware-Architektur

5.1.3 Software-Architektur

5.1.3.1 Echtzeit-Betriebssystem

Die Software-Architektur des vorliegenden Systems basiert auf einem Multitasking-Design mit einem Echtzeit-Betriebssystem (RTOS). Diese Wahl zielt darauf ab, die Systemreaktionszeit zu optimieren, die Ressourcennutzung zu effektivieren und die Flexibilität der Geschäftsmodule zu erhöhen. Als RTOS wurde FreeRTOS ausgewählt, da es durch seinen geringen

Ressourcenverbrauch, hohe Zuverlässigkeit und schlanke Struktur optimal den Anforderungen dieses Forschungsprojekts entspricht.

Überblick über Multitasking-Betriebssysteme

Das Herzstück eines jeden Betriebssystems ist der Kernel. Fortschrittliche Betriebssysteme wie Linux nutzen Kernel, die es ermöglichen, die simultane Nutzung des Computers durch mehrere Benutzer zu simulieren. In der Praxis bedeutet dies, dass mehrere Benutzer gleichzeitig verschiedene Programme ausführen können, wobei jedes dieser Programme als eine vom Betriebssystem gesteuerte Aufgabe (oder Thread) fungiert. Betriebssysteme mit dieser Fähigkeit werden als Multitasking-Betriebssysteme bezeichnet.

Vorteile von Multitasking-Betriebssystemen

Der Einsatz eines Multitasking-Betriebssystems bietet erhebliche Vorteile bei der Entwicklung komplexer Softwareanwendungen:

- Die Multitasking- und Interprozesskommunikationsfunktionen des Betriebssystems ermöglichen es, umfangreiche Anwendungen in kleinere, überschaubare Aufgaben zu zerlegen.
- Diese Modularisierung vereinfacht das Software-Testing, optimiert die Arbeitsverteilung im Entwicklerteam und fördert die Wiederverwendbarkeit des Codes.
- Komplexe Timing- und Scheduling-Details können aus dem Anwendungscode ausgelagert und vom Betriebssystem verwaltet werden, was die Codestruktur vereinfacht.

Umsetzungsprinzip der Multitasking-Nebenläufigkeit

Selbst auf Einzelkernprozessoren können Multitasking-Betriebssysteme durch rasches Umschalten zwischen Aufgaben den Eindruck einer parallelen Ausführung erwecken. Abbildung 5-4 veranschaulicht das Ausführungsmuster von drei zeitabhängigen Aufgaben. Die farbigen Linien repräsentieren dabei die jeweils aktive Aufgabe. Der obere Bildteil zeigt die wahrgenommene simultane Ausführung, während der untere Teil das tatsächliche sequenzielle Multitasking-Muster darstellt.

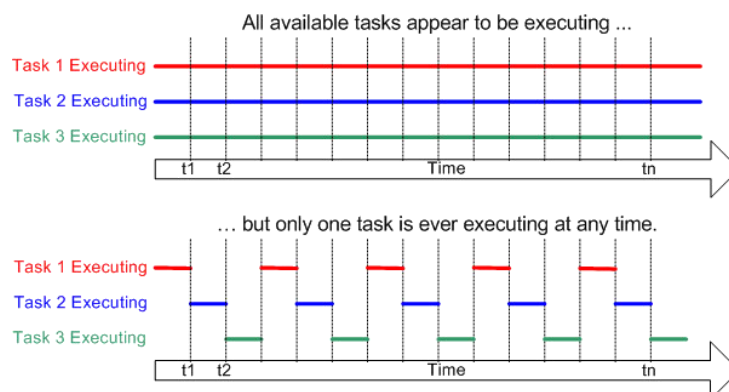


Abbildung 5-4 Multitasking-Implementierung in FreeRTOS

Hinweis: [Online-Bild] Quelle: FreeRTOS Website (<https://www.freertos.org/zh-cn-cmn-s/implementation/a00004.html>) [abgerufen am 29.07.2024].

Task-Scheduling

Der Scheduler, eine Kernkomponente des Betriebssystems, entscheidet darüber, welche Aufgaben zu welchem Zeitpunkt ausgeführt werden. Er kann Aufgaben während ihres Lebenszyklus mehrfach unterbrechen und später wieder aufnehmen. Die Scheduling-Strategie, ein Algorithmus des Schedulers, bestimmt die Ausführungsreihenfolge der Aufgaben. In Nicht-Echtzeit-Systemen könnte diese Strategie darauf abzielen, jeder Aufgabe einen "fairen" Anteil an der Prozessorzeit zuzuweisen.

Neben der vom Kernel erzwungenen Unterbrechung können sich Aufgaben auch selbst aussetzen. Dies geschieht, wenn eine Aufgabe für eine bestimmte Zeit pausieren (schlafen) oder auf die Verfügbarkeit einer Ressource (z.B. einen seriellen Port) oder den Eintritt eines Ereignisses (z.B. einen Tastendruck) warten (blockieren) muss. Blockierte oder schlafende Aufgaben können nicht ausgeführt werden und erhalten keine Prozessorzeit.

Ein anschauliches Beispiel bietet Abbildung 5-4, in der die Aufgaben 1 bis 3 nach absteigender Priorität geordnet sind. Muss eine höher priorisierte Aufgabe ausgeführt werden, unterbricht das System-Scheduling die Aufgaben mit niedrigerer Priorität und führt stattdessen die wichtigere Aufgabe aus.

Vorteile von FreeRTOS

Für komplexe IoT-Geräte wie den ESP32 erweist sich FreeRTOS als ideale Wahl. Forschungsergebnisse zeigen, dass der Super-Loop-Ansatz von Arduino (Ausführung aller Aufgaben in einer loop()-Schleife) zwar bei einfachen Aufgaben geringfügig schneller sein kann, FreeRTOS jedoch ein überlegenes Multitasking-Management, bessere Echtzeitfähigkeit und höhere Skalierbarkeit bietet. Diese Eigenschaften sind für den ESP32, der mehrere gleichzeitige Aufgaben bewältigen muss, von entscheidender Bedeutung.[31]

5.1.3.2 Überblick über den Software-Ablauf

Die grundlegende Softwarearchitektur unseres Systems, das auf FreeRTOS als Betriebssystem basiert, wird in Abbildung 5-5 schematisch dargestellt. Es ist wichtig zu beachten, dass aufgrund der Vielzahl und der unterschiedlichen Komplexität der implementierten Aufgaben eine vollständige Darstellung aller Prozesse in einem einzigen Flussdiagramm nicht realisierbar ist. Um diesem Umstand Rechnung zu tragen, werden detaillierte Software-Flussdiagramme für jedes einzelne Softwaremodul im Verlauf ihrer jeweiligen Vorstellung separat präsentiert. Diese Herangehensweise ermöglicht eine präzisere und übersichtlichere Darstellung der spezifischen Abläufe innerhalb der verschiedenen Systemkomponenten.

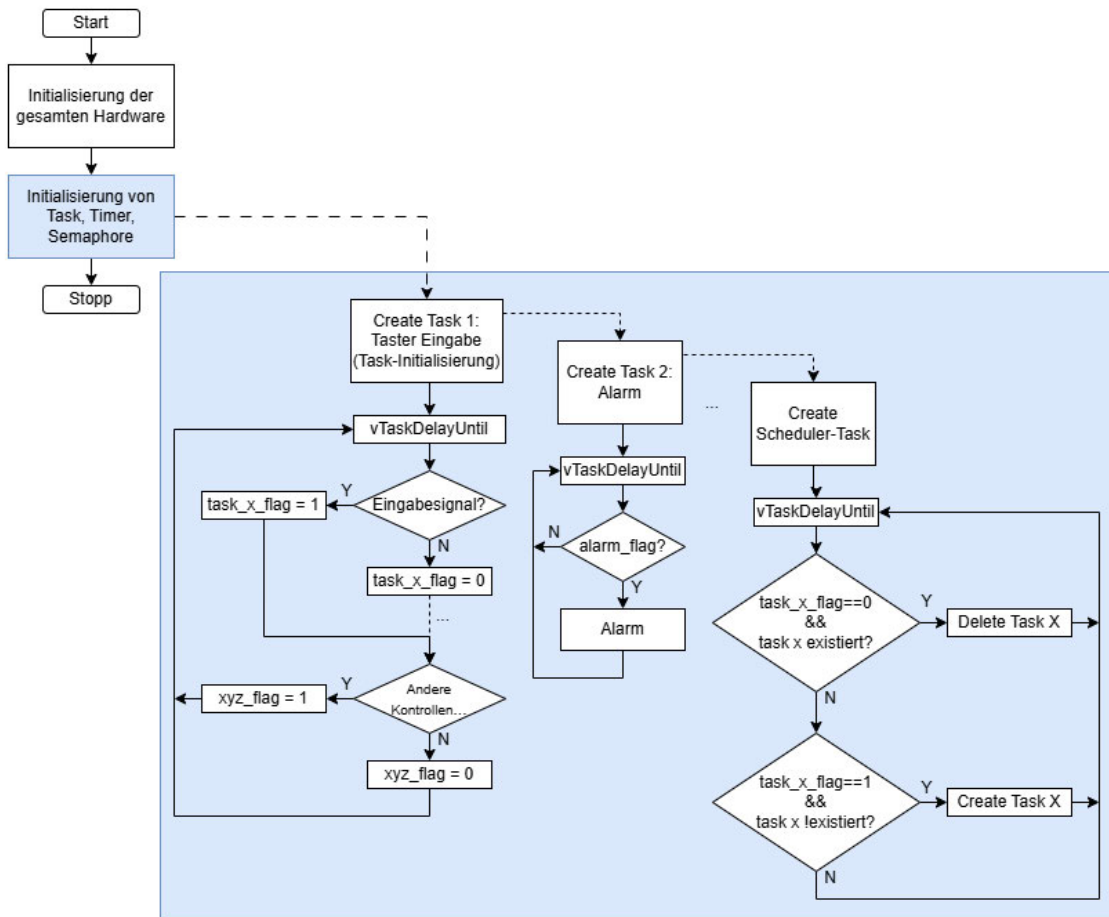


Abbildung 5-5 Software-Ablauf

5.2 Hardware-Design

In diesem Abschnitt wird der Hardware-Designprozess des Schadstoffdetektionssystems eingehend erläutert. Jede Phase, von der initialen Hardwareauswahl bis zur finalen Montage, wurde sorgfältig konzipiert und umgesetzt. Ziel war es, ein System zu entwickeln, das die vorgegebenen funktionalen Anforderungen und Leistungskriterien optimal erfüllt.

5.2.1 Hardwareauswahl

Die Auswahl der Hardware stellt einen kritischen Schritt im Systemdesign dar und hat direkten Einfluss auf Leistung, Energieeffizienz, Kosten und Zuverlässigkeit des Gesamtsystems. Dieser Abschnitt beleuchtet detailliert den Auswahlprozess und die Begründungen für die wichtigsten Hardwarekomponenten. Besonderes Augenmerk liegt dabei auf der Wahl des Mikrocontrollers, der Sensoren, des Speichermoduls und der Stromversorgungseinheit. Durch umfassende Vergleiche und Analysen wurde sichergestellt, dass die ausgewählten Hardwarekomponenten optimal auf die Anforderungen dieser Forschungsarbeit abgestimmt sind und das System seine vorgesehenen Funktionen effizient und zuverlässig erfüllen kann.

5.2.1.1 Auswahl des Mikrocontrollers

Der Mikrocontroller fungiert als zentrale Verarbeitungseinheit des Systems. Seine Leistungsfähigkeit und Eigenschaften sind daher entscheidend für die Funktionalität und Effizienz des Gesamtsystems. Nach eingehender Evaluation verschiedener Optionen fiel die Wahl auf das ESP32-C6-DevKitC-1-N8 Entwicklungskit (Abbildung 5-6) von Espressif als Mikrocontroller-Plattform. Im Folgenden werden der Auswahlprozess und die ausschlaggebenden Faktoren für diese Entscheidung detailliert erläutert.



Abbildung 5-6 ESP32-C6-DevKitC-1-N8 Entwicklungskit

Hinweis: [Online-Bild] Quelle: Espressif Website (https://docs.espressif.com/projects/espressif-esp-dev-kits/zh_CN/latest/esp32c6/esp32-c6-devkitc-1/user_guide.html) [abgerufen am 21.05.2024].

Analyse der Leistungsanforderungen

Basierend auf der vorangegangenen Bedarfsanalyse wurden folgende Hauptanforderungen an den Mikrocontroller für dieses System identifiziert:

- Ausreichende Rechenleistung für die Echtzeit-Detektion und Verarbeitung von Daten mehrerer Gase.
- Niedriger Energieverbrauch zur Gewährleistung langer Betriebszeiten.
- Vielfältige Peripherieschnittstellen für den Anschluss verschiedener Sensoren und Anzeigevorrichtungen.
- Unterstützung eines Echtzeit-Betriebssystems für effizientes Multitasking.
- Vorzugsweise integrierte Funktionen für drahtlose Kommunikation.

Vergleich der Alternativen

Im Rahmen dieser Studie wurden mehrere marktführende Mikrocontroller-Plattformen evaluiert, darunter die ESP32-Serie, die STM32-Serie und die Arduino-Serie. Als repräsentative Beispiele wurden der ESP32-C6, die STM32F4-Serie und der Arduino Nano 33 IoT hinsichtlich ihrer Hauptmerkmale verglichen. Eine detaillierte Gegenüberstellung finden Sie in Tabelle 5-1.

Eigenschaft	ESP32-C6	STM32F4-Serie	Arduino Nano 33 IoT
Prozessor	32-Bit Single-Core RISC-V, 160 MHz	32-Bit Single-Core ARM Cortex-M4, 180 MHz	32-Bit Single-Core ARM Cortex-M0+, 48 MHz
ROM	320 KB	Typischerweise 1-2 MB	448 KB

RAM	512 KB HP SRAM + 16 KB LP SRAM	192-512 KB SRAM	32 KB SRAM
Flash	4 MB im Modul integriert, erweiterbar, insgesamt 8 MB auf dem Entwicklungsboard	512 KB - 2 MB (on-chip)	256 KB
GPIO-Anzahl	30 (QFN40) oder 22 (QFN32)	Bis zu 140 (modellabhängig)	14 digitale Pins, 8 analoge Eingangspins
Peripherie-Unterstützung	SPI, UART, I2C, I2S, RMT, TWAI, PWM, SDIO, Motor Control PWM, 12-bit ADC, USB Serial/JTAG	SPI, UART/USART, I2C, I2S, CAN, PWM, SDIO, 12-bit ADC, DAC, USB OTG, Ethernet	SPI, UART, I2C, PWM, 12-bit ADC, USB
WiFi	Integriertes 2,4 GHz Wi-Fi 6 (802.11ax)	Keine integrierte Drahtlosfunktion, externes Modul erforderlich	Integriertes Wi-Fi
Bluetooth	Integriertes Bluetooth 5 (LE)	Keine integrierte Drahtlosfunktion, externes Modul erforderlich	Integriertes Bluetooth (LE)
Stromverbrauch	Relativ niedrig	Mittel	Relativ niedrig
Entwicklungs-Ökosystem	ESP-IDF (offizielles Espressif SDK), Arduino IDE-Unterstützung, kompatibel mit den meisten Arduino-Bibliotheken und Drittanbieter-Bibliotheken	STM32CubeIDE (offizielles ST IDE), Unterstützung für verschiedene Entwicklungsumgebungen (Keil, IAR usw.), ausgereiftes Ökosystem und umfangreiche Bibliotheken	Arduino IDE, umfangreiche Community-Ressourcen, einfach zu bedienen
Preis	Relativ niedrig	Relativ hoch	Mittel

Tabelle 5-1 Vergleich der Eigenschaften verschiedener Mikrocontroller-Plattformen

Begründung für die Wahl des ESP32-C6

Nach sorgfältiger Abwägung der Systemanforderungen und der Eigenschaften verschiedener Plattformen fiel die Wahl dieser Studie auf das ESP32-C6-DevKitC-1-N8 Entwicklungskit als Mikrocontroller-Plattform. Die ausschlaggebenden Gründe für diese Entscheidung sind:

- **Optimale Balance zwischen Leistung und Energieeffizienz:** Der ESP32-C6 basiert auf einer Single-Core RISC-V-Architektur mit 160 MHz Taktfrequenz, die den Rechenanforderungen des Systems vollauf genügt. Die Single-Core-Konfiguration trägt zudem zur Minimierung des Stromverbrauchs bei, was den Anforderungen an lange Betriebszeiten entgegenkommt.
- **Integrierte Drahtloskommunikation:** Mit integriertem 2,4 GHz WLAN und Bluetooth 5 macht der ESP32-C6 zusätzliche Kommunikationsmodule überflüssig. Dies vereinfacht das Hardware-Design und reduziert gleichzeitig Systemgröße und Energieverbrauch.
- **Umfangreiche Peripherieschnittstellen:** Der ESP32-C6 verfügt über zahlreiche GPIO-Schnittstellen und unterstützt diverse Protokolle, was den Anschluss verschiedener Sensoren und Anzeigevorrichtungen ermöglicht. Das ESP32-C6-DevKitC-1-N8 Entwicklungskit ist mit zwei USB-C-Anschlüssen ausgestattet, die eine simultane Stromversorgung erlauben. Ein integrierter 3,3V LDO-Chip wandelt die Eingangsspannung von USB oder 5V-Pin in die für den SoC benötigten 3,3V um.

- RTOS-Unterstützung: Die vollständige Unterstützung von FreeRTOS durch den ESP32-C6 ermöglicht effizientes Multitasking und erfüllt die Echtzeitanforderungen des Systems.
- Robustes Entwicklungsökosystem: Trotz seiner relativen Neuheit ist der ESP32-C6 kompatibel mit dem ESP-IDF Entwicklungsframework und der Arduino IDE. Dies erlaubt die Nutzung der umfangreichen Ressourcen des ESP32- und Arduino-Ökosystems, was den Entwicklungsprozess erheblich beschleunigt.
- Hervorragendes Preis-Leistungs-Verhältnis: Im Vergleich zur STM32-Serie bietet der ESP32-C6 bei vergleichbarer Leistung ein besseres Preis-Leistungs-Verhältnis. Gegenüber dem Arduino Nano 33 IoT zeichnet er sich durch niedrigere Kosten, höhere Leistung und umfangreichere Funktionalität aus.
- Zukunftssicherheit: Mit 8 MB Flash-Speicher und 512 KB RAM bietet das ESP32-C6-DevKitC-1-N8 Entwicklungskit ausreichend Kapazität für zukünftige Funktionserweiterungen.

Eine umfassende Übersicht über die ESP32-C6-DevKitC-1 Entwicklungskit-Serie ist in Abbildung 5-7 dargestellt:

ESP32-C6-DevKitC-1

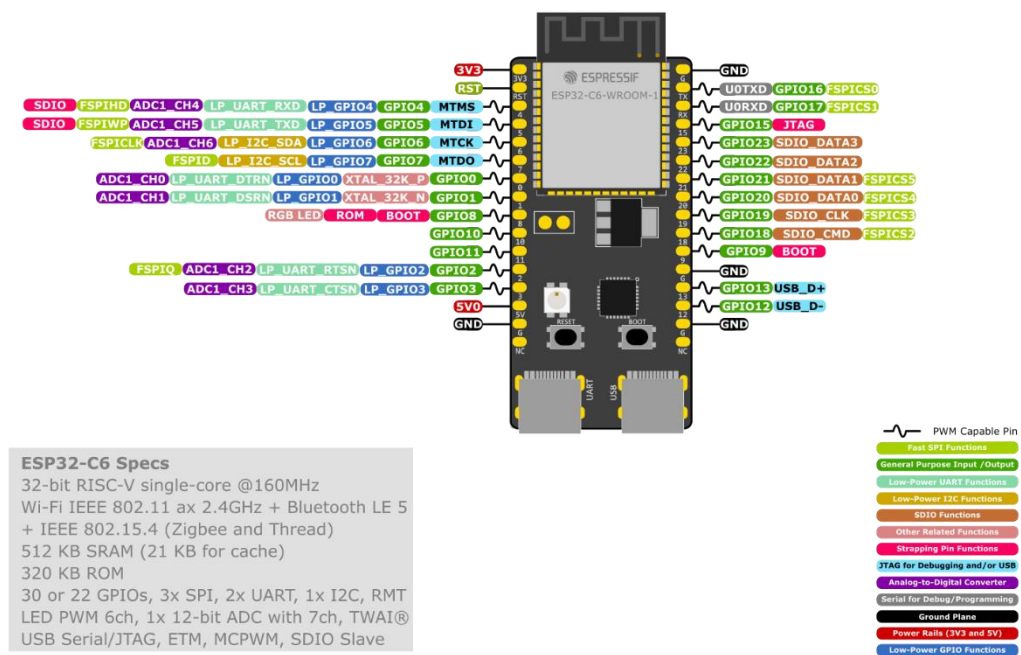


Abbildung 5-7 Detaillierte Spezifikationen des ESP32-C6-DevKitC-1

Hinweis: [Online-Bild] Quelle: Espressif Website (https://docs.espressif.com/projects/espressif-esp-dev-kits/zh_CN/latest/esp32c6/esp32-c6-devkitc-1/user_guide.html) [abgerufen am 21.05.2024].

5.2.1.2 Sensorauswahl

Sensoren bilden das Herzstück eines Gasetektionssystems. Ihre Leistungsfähigkeit ist ausschlaggebend für die Präzision und Zuverlässigkeit des gesamten Systems. In der vorliegenden Studie fiel die Wahl auf das MEMS-MOS-Sensormodul MiCS-4514 (siehe Abbildung 5-8) als primäres Gasetektionselement.

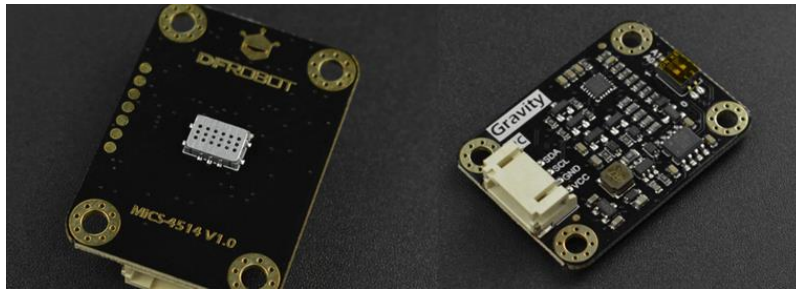


Abbildung 5-8 MiCS-4514 Modul

Hinweis: [Online-Bild] Quelle: DFRobot Wiki (https://wiki.dfrobot.com/_SKU_SEN0377_Gravity_MEMS_Gas_Sensor_CO_Alcohol_NO2_NH3_I2C_MiCS_4514#target_1) [abgerufen am 23.05.2024].

Vergleichende Analyse der Alternativen

Im Rahmen dieser Untersuchung wurden mehrere marktgängige Sensoren in Betracht gezogen, darunter Modelle der MiCS-, MQ- und ZE03-Serien. Eine detaillierte Gegenüberstellung der Hauptmerkmale dieser Sensoren findet sich in Tabelle 5-2.

Eigen-schaft	MiCS-4514 (MEMS)	MQ-Serie (MOS)	ZE03 (elektrochemisch)
Detektier-bare Gase	CO, NO ₂ , C ₂ H ₅ OH, H ₂ , NH ₃ , CH ₄	CH ₄ , C ₂ H ₅ OH, CO, H ₂ , brennbare Gase, Rauch, Luftqualität etc.	CO / O ₂ / NH ₃ / H ₂ S / NO ₂ / O ₃ / SO ₂ / CL ₂ / HF
Messgenau-igkeit	Kalibrierungsabhängig	Kalibrierungsabhängig	Keine Herstellerangaben; ver-gleichbare Produkte: ±1% des Messbereichs, max. ±10% bei Temperaturänderungen
Stromver-brauch	0,45 W bei 5 V DC (Modul)	Keine Herstellerangaben; gemessen: ca. 0,75 W bei 5 V DC	< 5 mA bei 5 V DC
Ansprech-zeit	Keine Herstellerangaben; gemessen: < 3 s (ver-gleichbare Produkte < 10 s)	Keine Herstellerangaben; gemessen: < 3 s (ver-gleichbare Produkte < 10 s)	15-150 s (gasabhängig)
Aufwärm-zeit	180 s (empfohlen)	60 bis 90 s	Nicht erforderlich
Betriebs-temperatur	-30 °C bis +85 °C	-20°C bis +50°C	-20°C bis +50°C
Betriebs-feuchtigkeit	15-90 % rF (relative Feuchte), nicht kondensierend	< 95 % rF	15-90 % rF, nicht konden-sierend
Integrati-onsgrad	Hoch (MEMS-Technologie)	Mittel	Mittel
Relative Kosten	Mittel	Niedrig	Hoch

Tabelle 5-2 Vergleichende Analyse der Eigenschaften verschiedener Gassensoren

Begründung der Sensorauswahl: MiCS-4514

Die Entscheidung für den MiCS-4514 Sensor basiert auf folgenden Schlüsselaspekten, die sich aus dem obigen Vergleich und den spezifischen Systemanforderungen ergeben:

- Multifunktionale Gasetektion: Der MiCS-4514 vereint zwei Sensorelemente - eines für reduzierende Gase (RED, z.B. CO) und eines für oxidierende Gase (OX, z.B. NO_x). Diese Kombination ermöglicht die simultane Erfassung zweier kritischer Kategorien toxischer und brennbarer Gase, was die Gesamtzahl der benötigten Sensoren im System reduziert.

- **Kompaktes Design:** Die Anwendung der MEMS-Technologie resultiert in deutlich geringeren Abmessungen des MiCS-4514 im Vergleich zu konventionellen Sensoren der MQ- und ZE03-Serien. Dies trägt wesentlich zur Miniaturisierung des Gesamtsystems bei.
- **Energieeffizienz:** Der MiCS-4514 zeichnet sich durch einen signifikant niedrigeren Stromverbrauch aus als Sensoren der MQ-Serie. Dies prädestiniert ihn für den Einsatz in tragbaren Geräten mit langen Betriebszeiten. Im Gegensatz dazu erfordert die MQ-Serie aufgrund ihres begrenzten Detektionsspektrums oft den Einsatz mehrerer Sensoren in einem Array, was den Gesamtenergiebedarf des Systems weiter erhöht.
- **Rasche Reaktionszeit:** MOS-Sensoren mit aktivem Heizmodul, wie der MiCS-4514, reagieren deutlich schneller als elektrochemische Sensoren. Diese Eigenschaft ist besonders vorteilhaft für die Detektion gefährlicher Substanzen in Brandszenarios, wo eine prompte Reaktion entscheidend ist.
- **Erweiterter Betriebstemperaturbereich:** Der MiCS-4514 funktioniert zuverlässig im Temperaturbereich von -30°C bis $+85^{\circ}\text{C}$. Damit übertrifft er die Systemanforderungen von -10°C bis $+60^{\circ}\text{C}$ und gewährleistet eine stabile Leistung unter diversen Umgebungsbedingungen.
- **Wirtschaftlichkeit:** Hochselektive, präzise elektrochemische Sensoren wie der ZE03 sind auf die Detektion eines einzelnen Gases beschränkt und übersteigen den Preis des MiCS-4514 um ein Vielfaches. Für die Erfassung von sechs verschiedenen Gasen wären sechs ZE03-Sensoren in einem Array erforderlich. Der Einsatz von MEMS-MOS-Sensoren ermöglicht hier eine erhebliche Kostenreduktion.

Einführung in den MiCS-4514 Sensor

Der MiCS-4514 Sensor ist ein dualer Gassensor, der aus zwei spezialisierten Sensoreinheiten besteht: einer für reduzierende Gase (RED) und einer für oxidierende Gase (OX). Die charakteristischen Ansprechkurven beider Sensoreinheiten sind in den Abbildung 5-9 und Abbildung 5-10 dargestellt. Diese Kurven veranschaulichen die Korrelation zwischen der Gaskonzentration (in ppm) und dem Widerstandsverhältnis des Sensors (R_s/R_0).

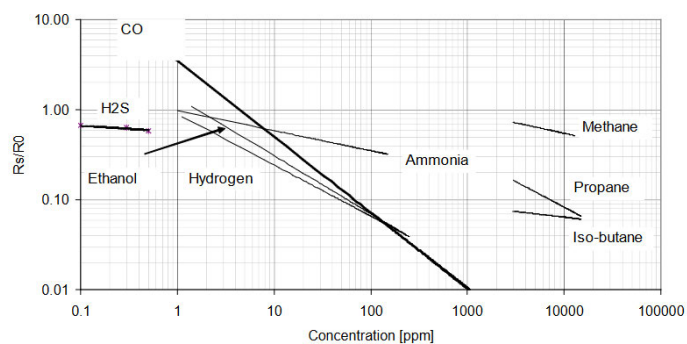


Abbildung 5-9 Ansprechcharakteristik der RED-Sensoreinheit für reduzierende Gase (Messbedingungen: 25°C , 50% relative Luftfeuchtigkeit)

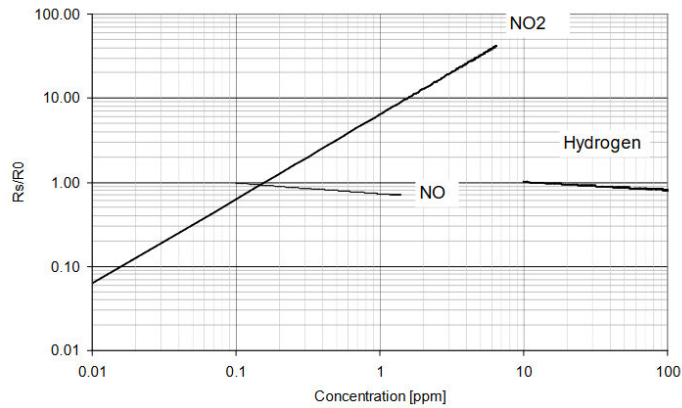


Abbildung 5-10 Ansprechcharakteristik der OX-Sensoreinheit für oxidierende Gase (Messbedingungen: 25°C, 50% relative Luftfeuchtigkeit)

Detektionsbereich des MiCS-4514

Tabelle 5-3 gibt einen Überblick über die vom MiCS-4514 detektierbaren Gasarten und deren jeweilige Messbereiche.

Gasart	Detektionsbereich
Methan (CH ₄)	1000-25000 ppm
Ethanol (C ₂ H ₅ OH)	10-500 PPM
Wasserstoff (H ₂)	1-1000 PPM
Ammoniak (NH ₃)	1-500 PPM
Kohlenmonoxid (CO)	1-1000 PPM
Stickstoffdioxid (NO ₂)	0,1-10 ppm

Tabelle 5-3 Detektionsbereich des MiCS-4514 Sensors

Limitationen des MiCS-4514

Als Vertreter der Metalloxid-Halbleiter (MOS) Sensoren weist der MiCS-4514 einige charakteristische Einschränkungen auf, insbesondere hinsichtlich der Selektivität bei der Gasunterscheidung und der erforderlichen Aufwärmzeit. Die Ansprechkurven in den Abbildung 5-9 und Abbildung 5-10 verdeutlichen, dass einem spezifischen Rs/R0-Wert potenziell mehrere Konzentrationen verschiedener Einzelgase oder Gasmischungen zugeordnet werden können. Folglich ist eine präzise Differenzierung der detektierten Gase allein anhand des Sensorausgangs nicht möglich.

Unter Berücksichtigung dieser inhärenten Eigenschaft basiert das Kernprinzip der Systeminterpretation auf folgender Annahme: Ein signifikanter Sensorausgang indiziert mit hoher Wahrscheinlichkeit die Präsenz eines oder mehrerer der oben genannten Gase. Diese Herangehensweise ermöglicht eine zuverlässige Detektion der Anwesenheit relevanter Gasarten, ohne jedoch eine exakte Quantifizierung oder Differenzierung einzelner Komponenten zu gewährleisten.

5.2.1.3 Hardwareauswahl für die Benutzerschnittstelle

Die Benutzerschnittstelle fungiert als zentrales Interaktionsmedium zwischen System und Anwender und ist somit von entscheidender Bedeutung für die Benutzerfreundlichkeit und das Nutzererlebnis des Geräts. Dieser Abschnitt beleuchtet detailliert die Hardwareauswahl für die

Benutzerschnittstelle, wobei drei Hauptkomponenten im Fokus stehen: Display, Alarmsystem und Bedienelemente.

Displayauswahl

Das Display stellt die primäre Informationsquelle für den Benutzer dar und beeinflusst maßgeblich die Lesbarkeit der Daten sowie die Praktikabilität des Gesamtsystems. Für diese Studie wurde ein 1,5-Zoll RGB OLED-Displaymodul mit SSD1351-Treiber (Abbildung 5-11) ausgewählt. Im Vergleich zu konventionellen 0,96-Zoll-Monochrom-OLEDs mit SSD1306-Treiber bietet dieses Modul eine signifikant größere Anzeigefläche und ein deutlich erweitertes Farbspektrum. Diese Wahl ermöglicht nicht nur eine präzisere und detailliertere Informationsdarstellung, sondern auch eine intuitive Visualisierung verschiedener Systemzustände und Alarmstufen durch Farbkodierung, was die Situationswahrnehmung des Benutzers erheblich verbessert. Tabelle 5-4 fasst die wichtigsten technischen Spezifikationen des OLED-Moduls zusammen.



Abbildung 5-11 Waveshare 1,5-Zoll RGB OLED-Displaymodul

Hinweis: [Online-Bild] Quelle: Waveshare Website (<https://www.waveshare.com/1.5inch-rgb-oled-module.htm>) [abgerufen am 27.05.2024].

OPERATING VOLTAGE	3.3V/5V	RESOLUTION	128×128 pixels
COMMUNICATION INTERFACE	3/4-wire SPI	DISPLAY SIZE	26.86 × 26.86 mm
DISPLAY PANEL	OLED	PIXEL SIZE	0.045 × 0.194 mm
DRIVER	1351	DIMENSIONS	44.5 × 37.0 mm

Tabelle 5-4 Technische Spezifikationen des Waveshare 1,5-Zoll RGB OLED-Displaymoduls

Hinweis: [Online-Tabelle] Quelle: Waveshare Website (<https://www.waveshare.com/1.5inch-rgb-oled-module.htm>) [abgerufen am 27.05.2024].

Hardwareauswahl für das Alarmsystem

Das akustische und optische Alarmsystem ist ein kritischer Bestandteil zur zeitnahen Erkennung potenzieller Gefahren durch den Benutzer. Der ESP32-C6-Entwicklungsboard verfügt über GPIO-Ausgänge mit einer Strombelastbarkeit von bis zu 40 mA, was für die direkte Ansteuerung der meisten Summer und LEDs ausreichend ist.

Im Wesentlichen unterscheidet man zwischen zwei Arten von Summern: aktive und passive. Der Hauptunterschied liegt in der integrierten Oszillatorschaltung bei aktiven Summern. Diese erzeugen bei Anlegen einer Gleichspannung einen Ton mit fester Frequenz. Passive Summer hingegen benötigen ein externes Frequenzsignal (z.B. eine Rechteckwelle) zur Tonerzeugung, wobei die Tonfrequenz durch Modulation des Eingangssignals variiert werden kann.

Für dieses System wurde ein aktiver Summer (Abbildung 5-12) als akustisches Warngerät gewählt. Ausschlaggebend für diese Entscheidung waren die einfache Ansteuerung mittels Gleichspannung sowie die daraus resultierende höhere Zuverlässigkeit und Stabilität, insbesondere für Anwendungen, die eine kontinuierliche Alarmierung erfordern. Der Summer zeichnet sich durch ein schnelles Ansprechverhalten und einen klaren Ton aus, wodurch eine effektive Aufmerksamkeitserregung auch in geräuschintensiven Umgebungen gewährleistet ist.



Abbildung 5-12 Aktiver Summer

Hinweis: [Online-Bild] Quelle: Roboter Bausatz Website (<https://www.roboter-bausatz.de/p/10-stueck-aktiver-buzzer-3-3v-5v>) [abgerufen am 27.05.2024].

Ergänzend zum akustischen Alarm ist das System mit roten und grünen LEDs als visuelle Warn-elemente ausgestattet. Dieser duale Alarmmechanismus gewährleistet eine zuverlässige Warnung unter verschiedensten Umgebungsbedingungen.

Hardwareauswahl für Bedienelemente

Die Bedienungstasten bilden die primäre Schnittstelle für die Befehlseingabe und Systeminteraktion durch den Benutzer. Unter Berücksichtigung der komplexen Einsatzumgebungen und der hohen Anforderungen an die Betriebszuverlässigkeit wurden für dieses System drei robuste Mikroschalter als Bedienungstasten (Abbildung 5-13) implementiert. Diese Wahl stellt einen optimalen Kompromiss zwischen Bedienkomfort und Gerätehaltbarkeit dar. Durch ein durchdachtes Funktionsdesign der Tasten können vielfältige Operationen wie Menünavigation, Parametereinstellung und Moduswechsel realisiert werden, wodurch das Gerät flexibel an verschiedene Einsatzszenarien angepasst werden kann.



Abbildung 5-13 Mikroschalter

Hinweis: [Online-Bild] Quelle: Mouser Electronics Website (<https://eu.mouser.com/ProductDetail/CK/PTS645SL432LFS?qs=UxeAxwACbqkgBd6NUV%2F7qQ%3D%3D>) [abgerufen am 27.05.2024].

5.2.1.4 Auswahl der Speichermodule

Die zuverlässige Datenspeicherung und präzise Zeiterfassung sind fundamentale Komponenten für die Funktionalität des Geräts und die Durchführung nachfolgender Analysen. Dieser Abschnitt beleuchtet detailliert die Auswahl der speicherbezogenen Module, wobei der Fokus auf zwei Hauptaspekten liegt: Datenspeichergeräte und Echtzeituhr-Module.

MicroSD-Modul

Um den anspruchsvollen Anforderungen an Speicherkapazität und Zuverlässigkeit gerecht zu werden, fiel die Wahl auf eine Kombination aus einem 5V-betriebenen MicroSD-Kartenslot-Modul (Abbildung 5-14) und einer 8GB MicroSD-Karte.

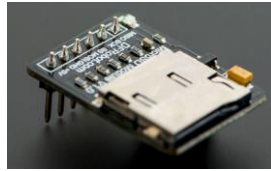


Abbildung 5-14 MicroSD-Kartenslot-Modul

Hinweis: [Online-Bild] Quelle: DFRobot Website (<https://www.dfrobot.com/product-875.html>) [abgerufen am 28.05.2024].

Diese Konfiguration bietet mehrere entscheidende Vorteile:

1. **Hohe Speicherkapazität:** MicroSD-Karten ermöglichen die Speicherung großer Datenmengen, was für langfristige und hochfrequente Datenaufzeichnungen unerlässlich ist.
2. **Flexibilität:** Die Abnehmbarkeit der SD-Karte erleichtert den Datentransfer und die Datensicherung, was die Systemflexibilität erheblich steigert.
3. **Zuverlässigkeit:** Die ausgereifte SD-Karten-Technologie gewährleistet hohe Kompatibilität und Stabilität, was eine sichere Datenspeicherung auch unter anspruchsvollen Umgebungsbedingungen sicherstellt.

Diese sorgfältig gewählte Speicherlösung bildet das Fundament für zuverlässige Datenspeicherung, was sowohl für nachfolgende Datenanalysen als auch für Langzeitüberwachungen von entscheidender Bedeutung ist.

RTC-Modul

Um die Einschränkung des ESP32-Mikrocontrollers hinsichtlich der kontinuierlichen Zeiterfassung bei Stromausfall zu überwinden, wurde ein DS3231 Echtzeituhr (RTC)-Modul (Abbildung 5-15) in das System integriert.



Abbildung 5-15 DS3231 RTC-Modul

Hinweis: [Online-Bild] Quelle: Az-delivery Website (<https://www.az-delivery.de/en/products/ds3231-real-time-clock>) [abgerufen am 06.06.2024].

Der DS3231 zeichnet sich als hochpräziser und energieeffizienter RTC-Chip aus. Neben seiner Hauptfunktion der genauen Zeiterfassung bietet er zusätzlich die Möglichkeit, Temperaturinformationen zu liefern. Diese Doppelfunktionalität eliminiert die Notwendigkeit eines separaten Temperatursensors für Anwendungen, die keine hochpräzisen Temperaturmessungen erfordern. Dadurch werden sowohl die Systemkomplexität als auch der Energieverbrauch effektiv reduziert, was die Gesamteffizienz des Geräts steigert.

5.2.1.5 Auswahl des Stromversorgungsmoduls

Die Wahl des Stromversorgungsmoduls ist von entscheidender Bedeutung für die Leistungsfähigkeit und Zuverlässigkeit des Gesamtsystems. Dieser Abschnitt beleuchtet die sorgfältige

Auswahl der Stromversorgungskomponenten, die sich auf drei Hauptaspekte konzentriert: die Stromquelle, das Spannungsabsenkungskonzept und das Batterieüberwachungssystem.

Auswahl der Stromquelle

Die Analyse kommerzieller tragbarer Geräte in Kapitel 3 zeigt, dass die Mehrheit hochkapazitive wiederaufladbare Batterien für lange Betriebszeiten einsetzt. Für die vorliegende Forschungsphase, die sich auf den grundlegenden Systemaufbau und -test konzentriert, war es jedoch erforderlich, einen Kompromiss zwischen Leistung und Kosten zu finden. Aus diesem Grund fiel die Wahl auf eine 9V-Alkalibatterie als primäre Stromquelle. Diese Entscheidung bietet mehrere Vorteile:

- Hohe Energiedichte: 9V-Alkalibatterien liefern in einem kompakten Format eine beachtliche Energiemenge.
- Praktikabilität: Die einfache Verfügbarkeit und der unkomplizierte Austausch erleichtern schnelle Tests und Iterationen im Entwicklungsprozess.
- Kosteneffizienz: Im Vergleich zu wiederaufladbaren Lithium-Batterien sind sie kostengünstiger, was sie ideal für die Prototypenentwicklung macht.

Spannungsabsenkungskonzept

Da die 9V-Spannung nicht direkt für die meisten Systemkomponenten geeignet ist, wurde ein Spannungsabsenkungsmodul als zentrales Element des Energiemanagements integriert. Nach eingehender Analyse wurde das LM2596-Modul (Abbildung 5-16) als optimale Lösung identifiziert.



Abbildung 5-16 LM2596 LM2596 Spannungsabsenkungsmodul

Hinweis: [Online-Bild] Quelle: Roboter Bausatz Website (<https://www.roboter-bausatz.de/p/spannungswandler-lm2596-dc-dc-step-down-modul-2a>) [abgerufen am 28.05.2024].

Der LM2596, ein Buck-Typ DC-DC-Abwärtswandler, zeichnet sich durch folgende Eigenschaften aus:

- Hohe Effizienz: Mit einer Energieumwandlungseffizienz von 85-95% bei großen Spannungsdifferenzen übertrifft er lineare Spannungsregler deutlich.
- Leistungsfähigkeit: Die Fähigkeit, höhere Ströme zu liefern, macht ihn ideal für anspruchsvolle Anwendungen.
- Flexibilität: Ein breiter Eingangsbereich ermöglicht die Anpassung an Batterieschwankungen bei gleichzeitiger Gewährleistung einer stabilen Ausgangsspannung.

Im direkten Vergleich zu LDO-Lösungen (wie dem AMS1117) bietet der LM2596 signifikante Vorteile in diesem Anwendungsszenario. Während LDOs durch ihre einfache Struktur und schnelle Reaktionszeit punkten, sind sie bei großen Spannungsdifferenzen ineffizient und daher für langlebige tragbare Geräte ungeeignet. Andere DC-DC-Topologien wie SEPIC oder Flyback wurden

ebenfalls in Betracht gezogen, aber unter Berücksichtigung von Systemkomplexität und Kosteneffizienz erwies sich der Buck-Wandler als die optimale Wahl.

Batterieüberwachung

Um die verbleibende Batteriekapazität in Echtzeit zu überwachen, integriert das System auch einen externen Spannungssensor (Abbildung 5-17) als Spannungsdetektionsschaltung. Die Schaltungsstruktur des Spannungssensors ist tatsächlich ein einfacher Spannungsteiler. Mit diesem Spannungsteilermodul kann der ADC-Spannungsbereich des ESP32 auf DC 0 bis 16,5V erweitert werden. Dieses Design ermöglicht es dem System, die Batteriespannung kontinuierlich zu überwachen und die verbleibende Kapazität durch einen voreingestellten Algorithmus zu schätzen. Dies ermöglicht nicht nur rechtzeitige Benachrichtigungen an den Benutzer zum Batteriewechsel, sondern bietet dem System auch die Möglichkeit, den Betriebsmodus basierend auf dem Batteriestand anzupassen, was die Energienutzung weiter optimiert.



Abbildung 5-17 Spannungsteilermodul

Hinweis: [Online-Bild] Quelle: Roboter Bausatz Website (<https://www.roboter-bausatz.de/p/spannung-sensor-modul-dc0-25v?number=RBS11893>) [abgerufen am 28.05.2024].

5.2.2 Schaltungsentwurf

Der Schaltungsentwurf dieses Systems basiert auf der zuvor beschriebenen Hardwareauswahl und zielt darauf ab, ein funktional vollständiges und leistungsstabiles System zur Detektion gefährlicher Substanzen zu realisieren. Der Schaltungsentwurf wurde mit der Software Altium Designer 24.5.2 durchgeführt, wie in Abbildung 5-18 dargestellt. Für Module oder Komponenten, für die keine offiziellen Bauteilsymbole verfügbar waren, wurden die Symbole manuell erstellt und alle für das Projekt benötigten Bauteilsymbole in einer einzigen Symbolbibliothek zusammengeführt.

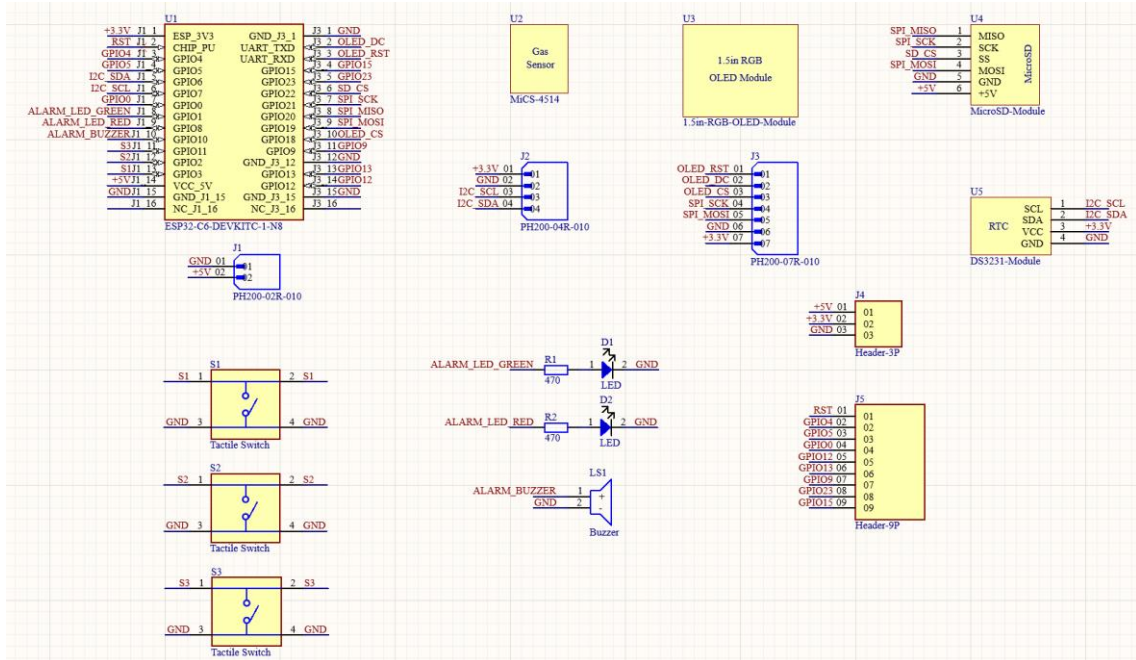


Abbildung 5-18 Schaltplan des Detektionssystems

Die Hauptkomponenten der Schaltung sind wie folgt:

- **Zentrale Steuereinheit:** Verwendet das ESP32-C6-DEVKITC-1-N8-Modul als Hauptsteuer-einheit, verantwortlich für die Datenverarbeitung und Steuerung des gesamten Systems.
- **Sensorschnittstelle:** Verbindet den MiCS-4514-Gassensor über eine I2C-Schnittstelle zur Detektion der Zielgase.
- **Anzeigemodul:** Verwendet ein 1,5-Zoll-RGB-OLED-Modul, das über eine SPI-Schnittstelle mit der Hauptsteuereinheit kommuniziert, um Detektionsdaten und Systemstatus anzuzei-gen.
- **Speichermodul:** Integriert ein MicroSD-Kartenmodul, das über eine SPI-Schnittstelle mit dem ESP32 verbunden ist, für die Datenspeicherung.
- **Echtzeituhr:** Verwendet ein DS3231-Modul, das über eine I2C-Schnittstelle mit dem ESP32 verbunden ist, um präzise Zeitinformationen bereitzustellen.
- **Benutzereingabe:** Drei Taster (S1, S2, S3) sind entworfen und mit den GPIO-Schnittstellen des ESP32 verbunden, für die Benutzerinteraktion.
- **Alarmsystem:** Umfasst zwei LEDs (grün und rot) und einen Signalgeber, jeweils mit den GPIO-Schnittstellen des ESP32 verbunden, für visuelle und akustische Warnungen.
- **Energieversorgung:** Entwurf von 3,3V- und 5V-Stromversorgungen, um den Strombedarf verschiedener Module zu decken.

Es ist zu beachten, dass die batteriebezogene Hardware nicht in diesem Schaltplan enthalten ist, da sie unabhängig von der Leiterplatte entworfen wurde.

Die Standard-Pin-Konfiguration des ESP32-C6-Entwicklungsboards ist wie folgt:

- I2C-Standardpins: SDA: GPIO 23, SCL: GPIO 22
- SPI-Standardpins: SS: GPIO 18, MOSI: GPIO 19, MISO: GPIO 20, SCK: GPIO 21

Die Signalpin-Zuweisungen für die verschiedenen Module sind in Tabelle 5-5 dargestellt.

Gassensor- und RTC-Modulverkabelung (I2C)	OLED-Verkabelung (SPI)	SD-Kartenmodulverkabelung (SPI)
SDA (grünes Kabel) — GPIO 6 SCL (blaues Kabel) — GPIO 7	DIN(MOSI) (blaues Kabel) — GPIO 19 CLK (gelbes Kabel) — GPIO 21 CS (oranges Kabel) — GPIO 18 DC (grünes Kabel) — GPIO 16 (TX) RST (weißes Kabel) — GPIO 17 (RX)	MISO — GPIO 20 SCK— GPIO 21 SS/CS — GPIO 22 MOSI — GPIO 19
Akustische und visuelle Alarmverkabelung (GPIO-Ausgabemodus)	Tastenverkabelung (GPIO-Lesemodus)	Spannungsteiler
绿色 LED — GPIO 1 Rote LED — GPIO 8 Summer — GPIO 10	S1 — GPIO 3 S2 — GPIO 2 S3 — GPIO 11	Spannungssensor — GPIO0

Tabelle 5-5 Signalpin-Zuweisungen für verschiedene Module

Nicht zugewiesene GPIO-Pins umfassen: RST, 4, 5, 9, 12, 13, 15 und 23.

5.2.3 Schaltungstest

Vor der Herstellung der Leiterplatte (PCB) wurde zunächst ein Funktionstest der Schaltung auf einem Steckbrett durchgeführt, wie in Abbildung 5-19 dargestellt.

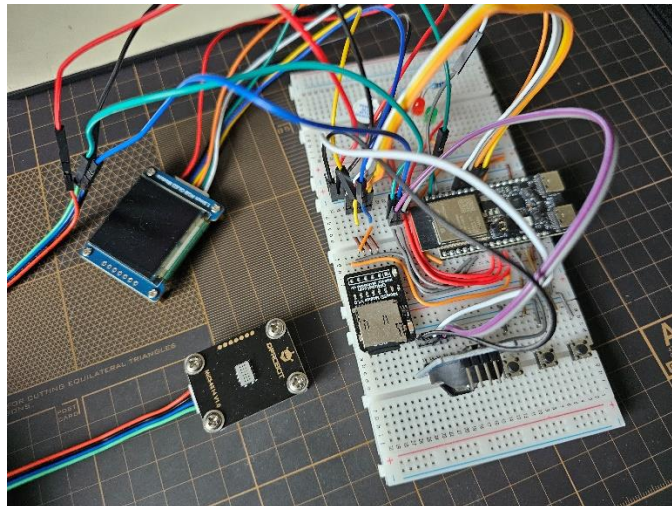


Abbildung 5-19 Schaltungstest auf dem Steckbrett

Die spezifischen Testschritte waren wie folgt:

1. Aufbau der Prototypschaltung auf dem Steckbrett gemäß dem Schaltplan, um die korrekte Verbindung zwischen den Modulen sicherzustellen.
2. Überprüfung des Stromversorgungskreises mit einem Multimeter, um zu gewährleisten, dass die Versorgungsspannung für jedes Modul den Anforderungen entspricht.
3. Schrittweise Überprüfung jedes Funktionsmoduls:
 - Verifizierung der I2C-Kommunikation zwischen ESP32 und Gassensor.
 - Test der OLED-Anzeigefunktion.
 - Überprüfung der Lese- und Schreibfunktion der MicroSD-Karte.

- Verifizierung der Zeiterhaltungsfunktion des RTC-Moduls.
 - Test der Reaktion der Taster.
 - Überprüfung der Alarmfunktion von LED und Signalgeber.
 - Test der Funktionen der Batterieversorgung.
4. Durchführung eines Systemintegrationstests zur Überprüfung der Stabilität des Zusammenspiels aller Module.

Durch den Test auf dem Steckbrett wurde die Funktionalität jedes einzelnen Moduls sowie des Gesamtsystems verifiziert. Dies schuf eine solide Grundlage für das nachfolgende Design der Leiterplatte.

5.2.4 Leiterplattendesign und -herstellung

Basierend auf dem Schaltplan und den Ergebnissen des Steckbretttests wurde das Leiterplattendesign mit Altium Designer 24.5.2 durchgeführt. Detaillierte Designzeichnungen finden sich im Abschnitt "Leiterplatten" im Anhang. Für Module oder Komponenten ohne offizielle Footprints wurden diese durch manuelle Messung und Zeichnung erstellt und mit den zuvor erstellten Bauteilsymbolen verknüpft. Alle im Projekt benötigten Footprints wurden in einer einzigen Bibliothek zusammengeführt.

Nach Abschluss des Leiterplattendesigns erfolgte die Herstellung mit CNC-Geräten im Labor der Hochschule Mittweida (wie in Abbildung 5-20 links dargestellt). Diese Methode bietet eine kürzere Fertigungszeit im Vergleich zu traditionellen chemischen Ätzverfahren. Vor dem Lötén müssen Durchkontaktierungen manuell installiert werden (wie in Abbildung 5-20 rechts gezeigt), um die elektrische Verbindung der doppelseitigen Leiterplatte sicherzustellen.

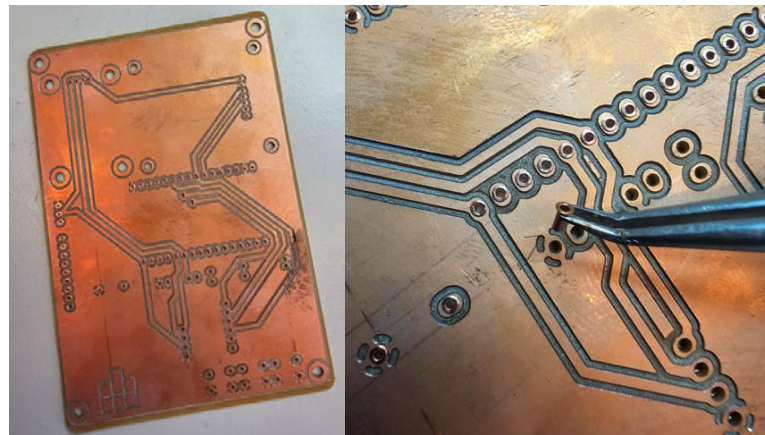


Abbildung 5-20 Mit CNC-Verfahren hergestellte Leiterplatte (links) und manuelle Installation von Durchkontaktierungen (rechts)

5.3 Softwaredesign

Dieser Abschnitt beschreibt detailliert das Softwaredesign des Systems zur Detektion gefährlicher Substanzen. Da der vollständige Code des Systems etwa 4000 Zeilen umfasst, wird die Analyse nach Modulen gegliedert. Es werden die wichtigsten Codeabschnitte der Kernmodule präsentiert

und mit entsprechenden Ablaufdiagrammen ergänzt, um den Programmablauf zu veranschaulichen. Für die übrigen Teile werden hauptsächlich Ablaufdiagramme verwendet und die Position des Codes im Gesamtprogramm angegeben, um einen einfachen Abgleich zu ermöglichen.

5.3.1 Codestruktur

Um die ESP32-Plattform mit der Arduino-Entwicklungsumgebung kompatibel zu machen, hat Espressif das offizielle `espressif/arduino-esp32`-Projekt entwickelt und als Open Source veröffentlicht (<https://github.com/espressif/arduino-esp32>). Diese Studie verwendet die Arduino IDE 2.3.2 in Kombination mit Version 3.0.3 von `espressif/arduino-esp32`, um die offiziellen ESP-Anleitungen und Arduino-Community-Ressourcen optimal zu nutzen.

Die Gesamtstruktur des Codes gliedert sich wie folgt:

1. Globale Definitionen
2. Funktionsabschnitt (alle Funktionen außer der Hauptinitialisierung und der Hauptschleife)
3. Hauptinitialisierungsfunktion (Arduino's `setup()`-Funktion)
4. Hauptschleifenfunktion (Arduino's `loop()`-Funktion)

5.3.2 Verwendete Bibliotheken

1. System- und hardwarebezogene Bibliotheken
 - `Wire.h`: Für I2C-Kommunikation. Von Espressif bereitgestellte native Arduino-Bibliothek.
 - `SPI.h`: Für SPI-Kommunikation. Von Espressif bereitgestellte native Arduino-Bibliothek.
2. Sensorbezogene Bibliotheken
 - `DFRobot_MICS.h`: Zur Steuerung und Datenauslesung des Gassensors. Open-Source-Bibliothek vom Sensormodulhersteller DFRobot (`DFRobot/DFRobot_MICS`-Projekt, https://github.com/DFRobot/DFRobot_MICS). MIT-Lizenz. Verwendete Version: 1.1.
3. Displaybezogene Bibliotheken
 - `SSD_13XX.h`: Zur Steuerung des OLED-Bildschirms. Von sumotoy bereitgestellte Open-Source-Bibliothek (`sumotoy/SSD_13XX`-Projekt, https://github.com/sumotoy/SSD_13XX). GPL-3.0-Lizenz. Verwendete Version: 1.0 stable.
 - `_icons/logo.c`: Enthält Bilddaten für das Startlogo. Generiert mit der Open-Source-Software `lcd-image-converter` (<https://sourceforge.net/projects/lcd-image-converter>) für das Logo der Hochschule Mittweida. Verwendete Softwareversion: rev.030b30d.
4. Speicherbezogene Bibliotheken
 - `SD.h`: Für SD-Karten-Lese- und Schreiboperationen. Von Espressif bereitgestellte native Arduino-Bibliothek.
 - `ArduinoJson.h`: Zur Analyse und Generierung von JSON-Daten. Von Benoît Blanchon bereitgestellte Open-Source-Bibliothek (`bblanchon/ArduinoJson`-Projekt, <https://github.com/bblanchon/ArduinoJson>). MIT-Lizenz. Verwendete Version: 7.1.0.

5. Zeitbezogene Bibliotheken

- `RTCLib.h`: Zur Steuerung des externen RTC (Real-Time Clock) Moduls. Von Adafruit bereitgestellte Open-Source-Bibliothek (adafruit/RTCLib-Projekt, <https://github.com/adafruit/RTCLib>). MIT-Lizenz. Verwendete Version: 2.1.4. Kompilierungsabhängigkeit: Adafruit BusIO-Bibliothek (Version 1.16.1).
- `ESP32Time.h`: Zur Steuerung der internen RTC des ESP32. Von Felix Biego bereitgestellte Open-Source-Bibliothek (fbiego/ESP32Time-Projekt, <https://github.com/fbiego/ESP32Time>). MIT-Lizenz. Verwendete Version: 2.0.6.

6. Netzwerkbezogene Bibliotheken

- `WiFi.h`: Für WiFi-Verbindungen. Von Espressif bereitgestellte native Arduino-Bibliothek.
- `ThingsBoard.h`: Zur Interaktion mit der ThingsBoard IoT-Plattform. Von Thingsboard offiziell bereitgestellte Open-Source-Bibliothek, `thingsboard-client-sdk`-Komponente (thingsboard/thingsboard-client-sdk-Projekt, <https://github.com/thingsboard/thingsboard-client-sdk>). MIT-Lizenz. Verwendete Version: 0.13.0. Kompilierungsabhängigkeiten: `TBPubSubClient` (Version 2.9.4) und `ArduinoHttpClient` (Version 0.6.1).
- `WiFiClientSecure.h`: Für sichere WiFi-Client-Verbindungen. `thingsboard-client-sdk`-Komponente, Abhängigkeitsbibliothek für ThingsBoard.
- `Arduino_MQTT_Client.h`: Für MQTT-Protokollkommunikation. `thingsboard-client-sdk`-Komponente, Abhängigkeitsbibliothek für ThingsBoard.

7. Weitere Funktionsbibliotheken

- `cfloat`: Für Kurvenzeichnung, bietet Maximal- und Minimalwerte für Fließkommazahlen (`FLT_MAX` und `FLT_MIN`). C-Standard-Bibliothekskopfdatei.
- `vector`: Für Kurvenzeichnung, bietet dynamische Array-Funktionalität. C++-Standard-Bibliothekskopfdatei.

Diese Bibliotheken vereinfachen den Entwicklungsprozess des Systems erheblich und ermöglichen es den Entwicklern, sich auf die Implementierung von Kernalgorithmen und -funktionen zu konzentrieren. In den folgenden Abschnitten wird die spezifische Anwendung dieser Bibliotheken in verschiedenen Modulen des Systems detailliert erläutert.

5.3.3 Globale Definitionen

Der Abschnitt der globalen Definitionen bildet das Fundament für die Systemkonfiguration und -operation. Er umfasst essenzielle Einstellungen und Deklarationen, die sich in folgende Kategorien gliedern:

1. Header-Dateien: Integration sämtlicher erforderlicher Bibliotheken für die Systemfunktionalität.
2. Konfigurationsparameter: Flexibel anpassbare globale Konstanten zur Systemoptimierung.
3. Hardware-Spezifikationen: Präzise Definitionen für Pins, I2C-Adressen und weitere hardwarenahe Komponenten.

4. Unveränderliche globale Konstanten: Systemkritische Werte, die in der Regel keiner Anpassung bedürfen.
5. Datenstrukturen: Detaillierte Definitionen von Enumerationen und Strukturen zur effizienten Datenverwaltung.
6. Globale Variablen: Systemweit zugängliche Variablen für den dynamischen Betrieb.
7. FreeRTOS-Handle-Initialisierung: Vorbereitung von Task-, Timer- und Mutex-Handles für das Echtzeitbetriebssystem.
8. Steuerungsflags: Boolesche Indikatoren zur präzisen Kontrolle diverser Systemfunktionen.

Die globalen Definitionen erstrecken sich über die Codezeilen 17 bis 526, wobei die konfigurierbaren Konstanten und die Anfangsvariablen in den Zeilen 52 bis 132 lokalisiert sind. Dieser spezifische Codeabschnitt ist im Anhang unter "Codeausschnitte" vollständig einsehbar. Die konfigurierbaren Konstanten adressieren kritische Systemaspekte wie Vorwärmzeiten, Zeitzoneneinstellungen und WiFi-Zugangsdaten. Jeder Parameter ist mit ausführlichen Kommentaren versehen, die dessen Funktion und potenzielle Auswirkungen erläutern.

Diese strukturierte Herangehensweise ermöglicht es sowohl Entwicklern als auch Endanwendern, das System effizient an spezifische Einsatzszenarien oder Hardware-Konfigurationen anzupassen, ohne tiefgreifende Codemodifikationen vornehmen zu müssen. Dies fördert die Flexibilität und Wartbarkeit des Systems erheblich.

5.3.4 Initialisierungsfunktion und Hauptschleifenfunktion

Dieser Abschnitt stellt die Initialisierungsfunktion und die Hauptschleifenfunktion im Arduino-Framework vor. Die Initialisierungsfunktion ist für die Einrichtung des Anfangszustands des Systems verantwortlich, während die Hauptschleifenfunktion in diesem System anders verwendet wird als in Standard-Arduino-Anwendungen.

5.3.4.1 Initialisierungsfunktion

Der Initialisierungsprozess des Systems wird in der `setup()`-Funktion konzentriert. Diese Funktion ist der Standardeinstiegspunkt im Arduino-Framework und führt einmalige Initialisierungsoperationen beim Systemstart durch. Die Initialisierungsfunktion entspricht den Zeilen 3639 bis 3748 im Code.

Die `setup()`-Funktion führt die Initialisierungsoperationen in der in Abbildung 5-21 dargestellten Reihenfolge aus. Sie stellt sicher, dass sich alle Hardware- und Softwarekomponenten des Systems vor Beginn der Hauptschleife im korrekten Anfangszustand befinden. Wenn einige kritische Kernhardware-Komponenten (wie der Gassensor) nach einer festgelegten Anzahl von Versuchen immer noch nicht normal verbunden werden können, startet das System automatisch neu.

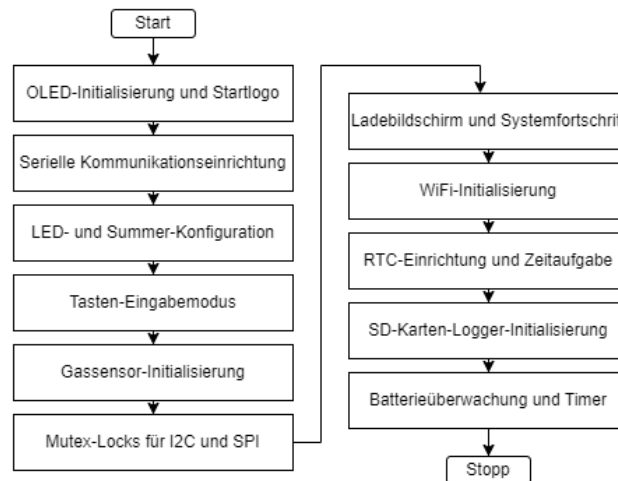


Abbildung 5-21 Ablaufdiagramm der Initialisierung

Die Aktualisierung des Aufwärmbildschirms wird durch die Erstellung einer einmaligen Aufgabe realisiert. Dies ermöglicht die gleichzeitige Initialisierung anderer Hardware, ohne deren Funktionalität zu beeinträchtigen. Innerhalb der Standard-Aufwärmzeit von 3 Minuten werden alle Hardware- und Software-Initialisierungen abgeschlossen. Nach Beendigung der Aufwärmphase erstellt diese Aufgabe relevante Aufgaben für die Bildschirmanzeige, Tastensteuerung, Gaserkennung usw. und löscht sich dann selbst.

Die Initialisierung anderer Hardware umfasst die Verbindung von WLAN und SD-Karte, die Initialisierung der externen Echtzeituhr (RTC) sowie die Einrichtung von Daueraufgaben (wie das sekundliche Abrufen der Echtzeit) und zeitgesteuerten Aufgaben (wie SD-Karten-Speicherverwaltung, Überprüfung des Batteriestatus).

Bei fehlgeschlagener WLAN-Verbindung deaktiviert das System die WLAN-Funktion, um den Gesamtstromverbrauch zu reduzieren. Nach erfolgreicher WLAN-Verbindung versucht das externe RTC-Modul, die Zeit mit dem angegebenen NTP-Server zu synchronisieren. Wenn alle NTP-Server nicht verfügbar sind oder WLAN nicht verbunden ist, verwendet das System die im externen RTC gespeicherte Zeit, um die interne RTC des ESP32 zu aktualisieren. Nach erfolgreicher Zeitabfrage vom NTP-Server aktualisiert das System zuerst die interne RTC und dann die externe RTC. Das System liest hauptsächlich die Echtzeit-Weltzeit vom externen RTC, während die interne RTC für temporäre oder kurzzeitige Zeitmessungen verwendet wird.

Der SD-Karten-Initialisierungsprozess umfasst die Erkennung des SD-Kartentyps, das Lesen oder Erstellen der Konfigurationsdatei und das Erstellen einer neuen Protokolldatei. Das Namensformat der Protokolldatei lautet "Jahr-Monat-Tag_Stunde-Minute-Sekunde.txt". Wenn keine Zeit abgerufen werden kann, wird "UnknownStartTime_" mit einer automatisch berechneten Sequenznummer verwendet.

5.3.4.2 Hauptschleifenfunktion

Da das System FreeRTOS-Taskmanagement einsetzt und mehrere Tasks erstellt, um verschiedene Funktionsmodule parallel zu verarbeiten, ist die Verwendung der Standard-Hauptschleifenfunktion `loop()` des Arduino-Frameworks nicht erforderlich. In FreeRTOS läuft die `loop()`-Funktion jedoch weiterhin als Task und kann mittels der `taskYIELD()`-Funktion CPU-Ressourcen für andere

Tasks freigeben oder zur Systemstatusüberwachung genutzt werden. Die Hauptschleifenfunktion entspricht den Zeilen 3751 bis 3755 im Code (Abbildung 5-22).

```
/* Hauptschleifenfunktion -----*/
void loop() {
  taskYIELD(); // CPU-Zeit aktiv anderen Aufgaben überlassen
  vTaskDelay(pdMS_TO_TICKS(10)); // Kurze Verzögerung möglich, hilft, den Stromverbrauch zu senken
}
```

Abbildung 5-22 Hauptschleifenfunktion

Die Systemaufgaben umfassen in der Reihenfolge ihrer Aktivierung: Sekundengenauiges Abrufen der Echtzeit, Überprüfung des Batteriestatus, Auslesen der Gaskonzentration, Überprüfung auf Überschreitung der Konzentration, akustische und visuelle Alarmierung, Aktualisierung des Bildschirminhalts, Verarbeitung von Tasteneingaben, SD-Karten-Protokollierung (nur bei angeschlossener SD-Karte), serielle Protokollausgabe (nur bei angeschlossener serieller Schnittstelle), Datensynchronisation mit der Thingsboard-Plattform sowie den Aufgabenplaner.

5.3.5 Datenerfassungs- und Verarbeitungsalgorithmen

Die Datenerfassung und -verarbeitung sind zentrale Elemente der Gefahrstoffdetektion. Dieser Abschnitt erläutert detailliert die Methoden der Datenerfassung sowie die Filtertechniken zur Verbesserung der Datenzuverlässigkeit. Durch präzise Datenerfassung und effektive Signalverarbeitung bietet das System eine zuverlässige Grundlage für die anschließende Gefahrstofferkennung.

5.3.5.1 Modulablauf

Abbildung 5-23 zeigt den Gesamtprozess der Datenerfassung und -verarbeitung.

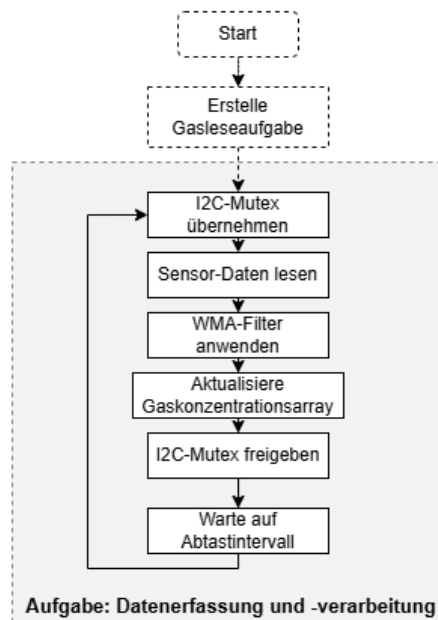


Abbildung 5-23 Ablaufdiagramm der Datenerfassungs- und Verarbeitungstasks

Gemäß dem Softwaredesign dieses Systems werden diese Funktionen als einzelne Tasks gekapselt und in Schleifen innerhalb der Tasks aufgerufen. Die Codestruktur eines einzelnen Tasks ist in Abbildung 5-24 dargestellt.

```

/* Aufgabe: Gaskonzentration vom Sensor lesen (Ziel-Funktion enthält I2C-Mutex) -----
void ReadGasTask_func(void *parameter) {
    TickType_t xLastWakeTime;
    const ticktype_t xFrequency = pdMS_TO_TICKS(SAMPLING_INTERVAL); // Konvertierung mit dem Makro pdMS_TO_TICKS

    Serial.println("[INFO] Task: Read Gas - Started");
    Serial.println();

    initGasBuffers(); // Initialisierung der Gaspuffer (für die Filterung)

    xLastWakeTime = xTaskGetTickCount(); // Aktuelle Systemzeit als Startzeit holen

    // Eine Aufgabe sollte niemals zurückkehren oder beenden
    for (;;) {

        // Ziel-Funktion aufrufen
        getGasData();

        vTaskDelayUntil(&xLastWakeTime, xFrequency); // Warten, damit die Aufgabe mit der angegebenen Frequenz ausgeführt wird
    }
}

```

Abbildung 5-24 Beispiel eines Tasks

Alle Funktionen, die zyklisch aufgerufen werden müssen, werden modulweise in verschiedene Tasks gekapselt. Das Starten, Stoppen oder Aussetzen dieser Tasks wird einheitlich durch den Task-Scheduler (siehe Abschnitt 5.3.12) gesteuert und über die von FreeRTOS bereitgestellten API-Funktionen verwaltet. Dieses Designprinzip wird in den folgenden Abschnitten bei der Vorstellung verschiedener Softwaremodule nicht wiederholt.

5.3.5.2 Datenerfassungsmethode

Dieses System verwendet den MiCS-4514-Sensor zur Gaskonzentrationserkennung. Es nutzt die `getGasData()`-Funktion aus der vom Sensormodulhersteller bereitgestellten speziellen Bibliothek zum Auslesen und Umwandeln der Gaskonzentrationsdaten (Abbildung 5-25). Diese Funktion integriert mehrere Gaskonzentrationsumrechnungsformeln und kann direkt umgerechnete ppm-Werte ausgeben.

```

/* Gasdaten vom Sensor erfassen, filtern und in das Array schreiben. Mit I2C-Mutex -----
void getGasData() {
    #if USE_MUTEX
        if (xSemaphoreTake(i2cMutex_handle, portMAX_DELAY) == pdTRUE) { // I2C-Mutex übernehmen
            #endif

            for (int i = 0; i < 6; ++i) {
                float rawValue = mics.getGasData(GAS_TYPE[i]);
                gasConcentrations[i] = applyWMA(rawValue, i);
            }

            #if USE_MUTEX
                xSemaphoreGive(i2cMutex_handle); // I2C-Mutex freigeben
            }
        #endif
    }
}

```

Abbildung 5-25 Auslesen von ppm-Werten vom Sensor

5.3.5.3 Datenfilterungsmethode

Um die Stabilität und Zuverlässigkeit der Daten zu erhöhen, verwendet diese Studie den gewichteten gleitenden Mittelwert (Weighted Moving Average, WMA) als Filteralgorithmus. Die Wahl von WMA basiert hauptsächlich auf folgenden Überlegungen:

- Glättungseffekt: WMA kann Datenkurven effektiv glätten und gleichzeitig Trendinformationen bewahren, was entscheidend ist, um kurzfristige Schwankungen zu reduzieren und Trends in der Gaskonzentration genau widerzuspiegeln.

- Anpassungsfähigkeit: Angesichts der 4-Hz-Abtastrate des Systems (alle 250 ms eine Abtastung) kann WMA kurzfristige Schwankungen effektiv glätten und bleibt gleichzeitig empfindlich für neue Daten, was den allgemeinen Eigenschaften von Gaskonzentrationsänderungen entspricht.
- Parameteranpassung: Im Vergleich zu anderen komplexen Filtern ist die Parameteranpassung bei WMA intuitiver und erfordert keine komplexe Frequenzbereichsanalyse, was den Prozess der Systemfehlersuche und -optimierung vereinfacht.
- Recheneffizienz: Obwohl die Effizienz von WMA etwas geringer ist als die des einfachen gleitenden Mittelwerts (SMA), ist sie immer noch wesentlich höher als komplexere Methoden wie der Kalman-Filter, was besonders für ressourcenbeschränkte eingebettete Systeme geeignet ist.
-
- Ansprechverhalten: WMA reagiert empfindlicher auf neue Daten als SMA und kann Trends in der Gaskonzentration besser verfolgen, was eine ausgewogenere Lösung zwischen Glättungseffekt und Ansprechgeschwindigkeit bietet.
- Verarbeitung mehrerer Datenströme: Angesichts der Tatsache, dass das System gleichzeitig Daten von mehreren Gasen verarbeiten muss, ermöglicht die moderate Berechnungskomplexität und der Speicherbedarf von WMA eine effiziente Verarbeitung mehrerer Datenströme, was ein wichtiger Vorteil für umfassende Gasdetektionssysteme ist.

Der in dieser Studie implementierte WMA-Filteralgorithmus ist in Abbildung 5-26 dargestellt. In dieser Implementierung ist die Standardfenstergröße (WMA_WINDOW_SIZE) auf 5 festgelegt, was bedeutet, dass der Filter die Daten der letzten 5 Messpunkte berücksichtigt. Bei einer Abtastrate von 4 Hz entspricht diese Fenstergröße einer Zeitspanne von 1,25 Sekunden, was kurzfristige Schwankungen effektiv glätten und gleichzeitig die Empfindlichkeit gegenüber Änderungen der Gaskonzentration bewahren kann.

```
/* Anwendung des WMA-Filters -----  
float applyWMA(float newValue, int gasType) {  
    // Aktualisierung des Puffers  
    gasBuffer[gasType][bufferIndex[gasType]] = newValue;  
    bufferIndex[gasType] = (bufferIndex[gasType] + 1) % WMA_WINDOW_SIZE;  
  
    // Berechnung des gewichteten Durchschnitts  
    float sum = 0;  
    for (int i = 0; i < WMA_WINDOW_SIZE; ++i) {  
        int index = (bufferIndex[gasType] - i + WMA_WINDOW_SIZE) % WMA_WINDOW_SIZE;  
        sum += gasBuffer[gasType][index] * WMA_WEIGHTS[i];  
    }  
  
    return sum;  
}
```

Abbildung 5-26 WMA-Filteralgorithmus

WMA verwendet eine nichtlineare Gewichtungsverteilung, wie in Abbildung 5-27 gezeigt. Das Gewichtungsarray WMA_WEIGHTS definiert die spezifische Gewichtungsverteilung. Der neueste Datenpunkt erhält das höchste Gewicht (0,3), während der älteste Datenpunkt das niedrigste Gewicht (0,1) erhält. Diese nichtlineare Gewichtungsverteilungsstrategie kann im Vergleich zur linearen Gewichtungsverteilung die Wichtigkeit der neuesten Daten und den Einfluss historischer Daten besser ausbalancieren und somit Trendinformationen bewahren und gleichzeitig Rauschen effektiver unterdrücken.

```
// Datenverarbeitungseinstellungen
constexpr uint8_t WMA_WINDOW_SIZE = 5; // WMA-Filter Gleitfenstergröße
constexpr float WMA_WEIGHTS[WMA_WINDOW_SIZE] = { 0.1, 0.15, 0.2, 0.25, 0.3 }; // WMA-Filter Gewichtsarray, wobei die neuesten Daten das größte Gewicht haben
```

Abbildung 5-27 WMA-Gewichtungsverteilung

5.3.6 Erkennung und Alarmierung von Gefahrstoffen

Die präzise Erkennung von Gefahrstoffen und eine zeitnahe Alarmierung sind für die Sicherheit der Anwender von entscheidender Bedeutung. Dieser Abschnitt stellt zunächst die Daten zu gefährlichen Konzentrationen der sechs vom System detektierten Hauptgase vor und erläutert, wie auf Basis dieser Daten angemessene Alarmschwellen festgelegt werden. Anschließend wird detailliert beschrieben, wie das System anhand der erfassten Daten Gefahrstoffe identifiziert und wie der entsprechende akustische und optische Alarmmechanismus konzipiert ist, um sicherzustellen, dass die Anwender bei Erkennung einer Gefahr umgehend benachrichtigt werden, damit sie die erforderlichen Schutzmaßnahmen ergreifen können.

5.3.6.1 Modulablauf

Der Gesamtablauf der Gefahrstofferkennung ist in Abbildung 5-28 dargestellt.

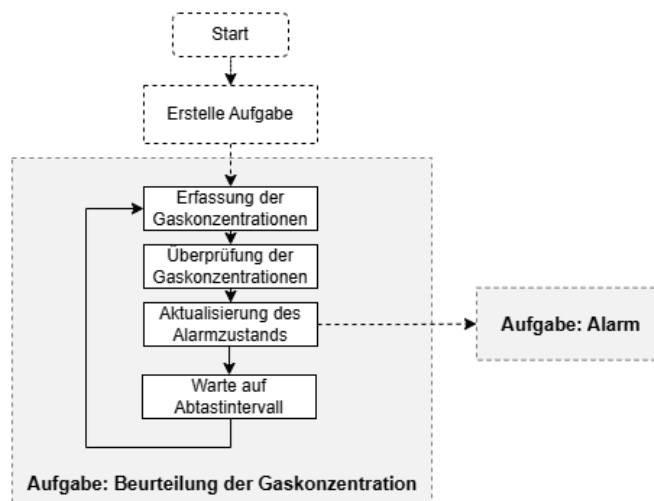


Abbildung 5-28 Ablaufdiagramm des Gefahrstofferkennungsmoduls

5.3.6.2 Daten zu gefährlichen Gaskonzentrationen und Alarmschwellen

Basierend auf dem "Pocket Guide to Chemical Hazards" des NIOSH [32] sind die Informationen zu gefährlichen Konzentrationen und die festzulegenden Schwellenwerte für Kohlenmonoxid, Stickstoffdioxid, Ammoniak und Ethanol in Tabelle 5-6 aufgeführt.

Kohlenmonoxid (CO)	Stickstoffdioxid (NO ₂)
IDLH: 1200 ppm	IDLH: 20 ppm
NIOSH REL: TWA 35 ppm (40 mg/m ³), C(Ceiling) 200 ppm (229 mg/m ³)	NIOSH REL: ST(Short-Term) 1 ppm (1.8 mg/m ³)
OSHA PEL: TWA 50 ppm (55 mg/m ³)	OSHA PEL: C 5 ppm (9 mg/m ³)

Ethanol (C₂H₅OH)	Ammoniak (NH₃)
IDLH: 3300 ppm (10%LEL) NIOSH REL: TWA 1000 ppm (1900 mg/m ³) OSHA PEL: TWA 1000 ppm (1900 mg/m ³)	IDLH: 300 ppm NIOSH REL: TWA 25 ppm (18 mg/m ³), ST 35 ppm (27 mg/m ³) OSHA PEL: TWA 50 ppm (35 mg/m ³)

Tabelle 5-6 Messbereiche einiger Gase für MiCS-4514 (Teil 1)

Darüber hinaus beschreibt ein Bericht von Alberta Agriculture die Expositionsgefahren und Auswirkungen von Methan [33]; die Verbrennungseigenschaften von Wasserstoff werden in Kapitel 1.8 von "Solar Hydrogen Production" detailliert beschrieben [34]. Die spezifischen Informationen sind in Tabelle 5-7 aufgeführt.

Methan (CH₄)	Wasserstoff (H₂)
NIOSH 8h TWA: 1000 ppm Potentieller Explosionsbereich: 50,000 - 150,000 ppm (5% - 15%) Erstickungskonzentration: 500,000 ppm (50%)	Brennbarer Bereich: 4% - 75% (40,000 - 750,000 ppm) Explosionsbereich (LEL-UEL): 15% - 59% (150,000 - 590,000 ppm)

Tabelle 5-7 Messbereiche einiger Gase für MiCS-4514 (Teil 2)

Für die anschließenden Leistungstests hat diese Studie basierend auf relevanten Sicherheitsstandards und Sensormessbereichen Standardalarmschwellen für verschiedene gefährliche Gase festgelegt. Die spezifischen Einstellungen sind wie folgt:

- Kohlenmonoxid (CO): 200 ppm, basierend auf der NIOSH REL-Höchstkonzentration (Ceiling).
- Stickstoffdioxid (NO₂): 5 ppm, in Anlehnung an die OSHA PEL-Höchstkonzentration.
- Ethanol (C₂H₅OH): 250 ppm, entsprechend 50% des maximalen Sensorbereichs.
- Ammoniak (NH₃): 150 ppm, basierend auf 50% des IDLH-Werts.
- Methan (CH₄): 10.000 ppm, entsprechend dem 10-fachen des von NIOSH angegebenen TWA.
- Wasserstoff (H₂): 500 ppm, entsprechend 50% des maximalen Sensorbereichs.

5.3.6.3 Methode zur Bestimmung von Gefahrstoffen

Dieses System verwendet eine dauerbasierte Alarmentscheidungsmethode, die darauf abzielt, die Genauigkeit der Erkennung zu erhöhen und Fehlalarme zu reduzieren. Der Kerngedanke dieser Methode ist, dass ein Alarm nur dann ausgelöst wird, wenn die Gaskonzentration über einen bestimmten Zeitraum hinweg den Schwellenwert überschreitet. Ebenso wird der Alarmzustand nur dann aufgehoben, wenn die Konzentration für eine bestimmte Zeit unter dem Schwellenwert bleibt. Diese Methode vermeidet effektiv Fehlalarme oder nicht erkannte Gefahrensituationen aufgrund kurzzeitiger Konzentrationsschwankungen. Die spezifische Entscheidungslogik ist wie folgt:

- Bestimmung der Schwellenwertüberschreitung: Wenn die Konzentration eines bestimmten Gases den voreingestellten Schwellenwert überschreitet, beginnt das System zu zählen.

Wenn die Konzentration den Schwellenwert für die voreingestellte Prüfzeit (GAS_CHECK_TIME) kontinuierlich überschreitet, wird der Alarmzustand für dieses Gas ausgelöst. Fällt die Konzentration während dieser Zeit unter den Schwellenwert, wird der Zähler zurückgesetzt.

- Bestimmung der Schwellenwertunterschreitung: Wenn die Gaskonzentration unter den Schwellenwert fällt, beginnt das System zu zählen. Bleibt die Konzentration für die voreingestellte Prüfzeit kontinuierlich unter dem Schwellenwert, wird der Alarmzustand für dieses Gas aufgehoben. Steigt die Konzentration während dieser Zeit wieder über den Schwellenwert, wird der Zähler zurückgesetzt.
- Bestimmung des Gesamtalarmzustands: Befindet sich ein Gas im Alarmzustand, wird der Gesamtalarmzustand des Systems auf Alarm gesetzt. Nur wenn sich kein Gas im Alarmzustand befindet, wird der Gesamtalarmzustand des Systems aufgehoben.

Die Implementierung dieser Bestimmungsmethode ist in Abbildung 5-29 dargestellt.

```

/* Überprüfen, ob die Gaskonzentration die Schwellenwerte erreicht -----
void checkGasConcentration() {
    // Initialisiere den allgemeinen Alarmstatus auf false
    isTotalAlarmState = false;

    // Prüfe, ob die Konzentration jedes Gases den festgelegten Schwellenwert erreicht oder überschreitet
    for (int i = 0; i < 6; ++i) {
        if (gasConcentrations[i] >= settings[i].threshold) {
            gasAboveThresholdCount[i]++;
            gasBelowThresholdCount[i] = 0; // Zähler für Werte unterhalb des Schwellenwerts zurücksetzen

            if (gasAboveThresholdCount[i] >= GAS_CHECK_TIME) {
                isAlarmState[i] = true;
            }
        } else {
            gasBelowThresholdCount[i]++;
            gasAboveThresholdCount[i] = 0; // Zähler für Werte oberhalb des Schwellenwerts zurücksetzen

            if (gasBelowThresholdCount[i] >= GAS_CHECK_TIME) {
                isAlarmState[i] = false;
            }
        }

        // Wenn der Alarmstatus irgendeines Gases true ist, dann ist der allgemeine Alarmstatus ebenfalls true
        if (isAlarmState[i]) {
            isTotalAlarmState = true;
        }
    }

    // Überprüfe, ob der Alarmstatus aller Gase false ist
    if (!isTotalAlarmState) {
        bool allFalse = true;
        for (int i = 0; i < 6; ++i) {
            if (isAlarmState[i]) {
                allFalse = false;
                break;
            }
        }

        isTotalAlarmState = !allFalse;
    }
}

```

Abbildung 5-29 Code zur Bestimmung von Gefahrstoffen

5.3.6.4 Konzeption des akustischen und optischen Alarmsystems

Nach der Detektion eines Gefahrstoffs muss das System den Anwender unverzüglich und effektiv alarmieren. In dieser Studie wurde ein integriertes akustisches und optisches Alarmsystem entwickelt, das den Anwender sowohl visuell als auch akustisch auf potenzielle Gefahren aufmerksam macht. Die zentrale Implementierungsfunktion des Alarmsystems ist in Abbildung 5-30 dargestellt.

```

/* Akustisch-optischer Alarm -----
void checkTotalAlarmState() {
  alarmCurrentMillis = millis();

  if (isTotalAlarmState) {
    // Wenn der Gesamtalarmstatus wahr ist, beginne sofort mit dem Alarm
    alarmEndMillis = 0; // Alarmende-Zeit zurücksetzen
    // Alle 500 Millisekunden den Status umschalten
    if (alarmCurrentMillis - alarmPreviousMillis >= ALARM_INTERVAL) {
      alarmPreviousMillis = alarmCurrentMillis;
      digitalWrite(REDDLED_PIN, digitalRead(REDDLED_PIN)); // Rote LED umschalten
      digitalWrite(BUZZER_PIN, digitalRead(BUZZER_PIN)); // Summer umschalten
    }
    digitalWrite(GREENLED_PIN, LOW); // Grüne LED ausschalten
  } else {
    // Wenn der Gesamtalarmstatus falsch wird, zeichne die Zeit auf, zu der der Alarm endete
    if (isLastTotalAlarmState && alarmEndMillis == 0) {
      alarmEndMillis = alarmCurrentMillis;
    }

    // Alarm deaktivieren, wenn die Zeit seit dem Alarmende mehr als 2 Sekunden beträgt
    if (alarmEndMillis > 0 && alarmCurrentMillis - alarmEndMillis >= ALARM_DELAY) {
      digitalWrite(GREENLED_PIN, HIGH); // Grüne LED dauerhaft einschalten
      digitalWrite(REDDLED_PIN, LOW); // Rote LED ausschalten
      digitalWrite(BUZZER_PIN, LOW); // Summer ausschalten
      alarmEndMillis = 0; // Alarmende-Zeit zurücksetzen
    } else if (alarmEndMillis > 0) {
      // Alarmzustand aufrechterhalten
      if (alarmCurrentMillis - alarmPreviousMillis >= ALARM_INTERVAL) {
        alarmPreviousMillis = alarmCurrentMillis;
        digitalWrite(REDDLED_PIN, digitalRead(REDDLED_PIN)); // Rote LED umschalten
        digitalWrite(BUZZER_PIN, digitalRead(BUZZER_PIN)); // Summer umschalten
      }
    }
  }

  isLastTotalAlarmState = isTotalAlarmState; // Den letzten Alarmstatus aktualisieren
}

```

Abbildung 5-30 Alarmverarbeitungsfunktion

Diese Funktion implementiert Schlüsselfunktionalitäten wie Alarmzustandssteuerung, akustische und optische Alarmierung, Sicherheitszustandsanzeige und verzögerte Alarmdeaktivierung. Im Detail bestimmt die Funktion anhand der Variable `isTotalAlarmState`, ob der Alarmzustand aktiviert werden muss. Diese Variable wird durch die zuvor erläuterte Logik zur Gefahrstofferkennung gesetzt. Befindet sich das System im Alarmzustand, wechseln sich die rote LED und der Summer in festen Zeitintervallen (definiert durch `ALARM_INTERVAL`) ab, wodurch ein blinkendes Lichtsignal und ein intermittierendes akustisches Signal erzeugt werden, um die Aufmerksamkeit des Anwenders effektiv zu erregen. Im Normalzustand leuchtet die grüne LED kontinuierlich, um dem Anwender die Sicherheit der Umgebung zu signalisieren.

Um zu verhindern, dass der Alarm aufgrund kurzzeitiger Konzentrationsschwankungen häufig aktiviert und deaktiviert wird, wurde ein Mechanismus zur verzögerten Alarmabschaltung implementiert. Dieser hält den Alarmzustand für eine definierte Zeitspanne (festgelegt durch `ALARM_DELAY`) aufrecht, nachdem die Alarmbedingungen nicht mehr erfüllt sind. Dieses Konzept erhöht die Systemstabilität und verbessert die Benutzererfahrung. Durch die Verwendung der Variable `isLastTotalAlarmState` zur Nachverfolgung des vorherigen Alarmzustands gewährleistet die Funktion einen reibungslosen Zustandsübergang.

5.3.7 Benutzeroberfläche

Die Benutzeroberfläche fungiert als Schnittstelle zwischen Anwender und System und beeinflusst maßgeblich die Bedienbarkeit und Zweckmäßigkeit des Geräts. Dieser Abschnitt erläutert detailliert die für dieses System entwickelte Benutzeroberfläche, wobei der Schwerpunkt auf zwei Kernaspekten liegt: dem Eingabemechanismus (Tastenbedienung) und der Ausgabeanzeige (Display). Durch eine sorgfältig konzipierte Interaktionslogik wird angestrebt, dem Anwender eine intuitive, reaktionsschnelle und funktionsreiche Bedienungserfahrung zu bieten, um so die Leistungsfähigkeit des Geräts in realen Einsatzszenarien effektiv zu steigern.

5.3.7.1 Konzeption der Tastenbedienung

Das System verwendet ein multifunktionales Tastensystem, das mittels drei physischer Tasten (SELECT, LEFT und RIGHT) eine Vielzahl von Interaktionsfunktionen ermöglicht. Dieses Konzept erfüllt nicht nur die Bedienungsanforderungen in verschiedenen Umgebungen, sondern bietet den Anwendern auch eine intuitive und flexible Steuerungsmethode. Kernstück des Tastensystems ist eine Zustandsmaschine (Abbildung 5-31), die verschiedenen Arten von Tasteneingaben, einschließlich kurzem Drücken, langem Drücken und Gedrückthalten, präzise erkennen und entsprechend reagieren kann.

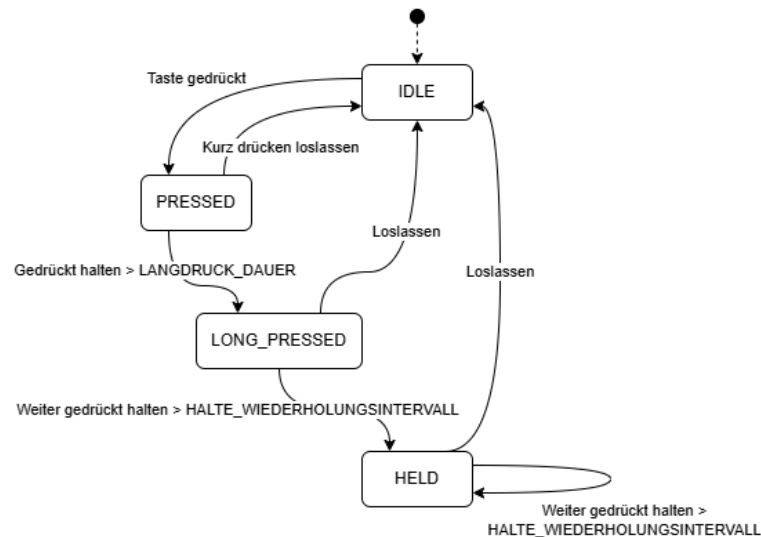


Abbildung 5-31 Schematische Darstellung der Tasten-Zustandsmaschine

Die Implementierung der Tasten-Zustandsmaschine verwendet vier durch einen Aufzählungstyp definierte Zustände: IDLE (Ruhezustand), PRESSED (Gedrückt), LONG_PRESSED (Lang gedrückt) und HELD (Gehalten). Mittels der `handleButtons()`-Funktion überwacht das System kontinuierlich die Zustandsänderungen jeder Taste und löst entsprechende Aktionen basierend auf verschiedenen Zuständen und Zeitschwellenwerten aus. Diese Konzeption ermöglicht es dem System, zwischen kurzem Drücken, langem Drücken und Gedrückthalten zu differenzieren, wodurch vielfältige Funktionen mit einer begrenzten Anzahl physischer Tasten realisiert werden können.

Kurzes Drücken wird primär verwendet, um zwischen verschiedenen Benutzeroberflächen zu navigieren oder Einstellungen anzupassen. Beispielsweise kann ein kurzer Druck auf die SELECT-Taste zwischen der Haupt- und Unteroberfläche wechseln oder im Einstellungsmenü zwischen verschiedenen Optionen umschalten. Die LEFT- und RIGHT-Tasten dienen der Anpassung von Parameterwerten im Einstellungsmenü. Langes Drücken wird genutzt, um das Einstellungsmenü aufzurufen oder zu verlassen. Durch langes Drücken der SELECT-Taste kann der Anwender bequem in den Einstellungsmodus wechseln oder Einstellungen speichern und zur Hauptoberfläche zurückkehren.

Zur Verbesserung der Benutzerfreundlichkeit wurde eine Gedrückthalten-Funktion implementiert, die es Anwendern ermöglicht, Parameter wie Schwellenwerte rasch anzupassen. Diese Funktionalität wird durch die `handleHold()`-Funktion realisiert. Beim Gedrückthalten der LEFT- oder

RIGHT-Taste ändert sich der entsprechende Parameterwert in größeren Schritten, was die Effizienz der Einstellung erhöht.

Darüber hinaus berücksichtigt das System die Präzision und Sicherheit der Parameteranpassung. Die `changeThreshold()`-Funktion verwendet nicht nur unterschiedliche Anpassungsschritte für verschiedene Parametertypen, sondern stellt auch durch die `constrain()`-Funktion sicher, dass der angepasste Wert stets innerhalb des vordefinierten Sicherheitsbereichs bleibt. Diese Konzeption verhindert effektiv Parameterüberschreitungen durch Fehlbedienungen und erhöht somit die Stabilität und Zuverlässigkeit des Systems.

Zur Gewährleistung der Persistenz der Einstellungen wendet das System automatisch temporäre Einstellungen auf globale Variablen an, wenn der Anwender das Einstellungsmenü verlässt, und speichert die Konfiguration auf der SD-Karte, falls verfügbar. Diese Funktionalität wird durch die Funktionen `applyTempSettingsToGlobal()` und `saveConfigToSD()` implementiert und bietet dem Anwender eine komfortable Methode zur Konfigurationsverwaltung.

5.3.7.2 Konzeption der Displayanzeige

Die Displayanzeige als zentrales Ausgabeelement der Benutzeroberfläche ist entscheidend für die Bereitstellung intuitiver und zeitnaher Informationsrückmeldungen. Dieser Abschnitt erörtert detailliert die Konzeption und Implementierung der Displayanzeige, einschließlich des Gesamtlayouts, der Hauptfunktionsoberflächen und des dynamischen Aktualisierungsmechanismus.

Gesamtlayout-Konzeption

Die Displayanzeige folgt einem modularen Konzept, das sich in drei Hauptoberflächen gliedert: Hauptansicht, Nebenansicht und Einstellungsmenü. Diese Struktur ermöglicht Anwendern ein flexibles Navigieren zwischen verschiedenen Funktionsmodulen, gewährleistet die Vollständigkeit der Informationen und steigert gleichzeitig die Bedienungsfreundlichkeit. Jede Oberfläche verfügt über ein spezifisches Layout und dedizierte Funktionen:

- Hauptansicht: Visualisierung von Echtzeit-Gaskonzentrationsdaten und Systemstatus.
- Nebenansicht: Darstellung historischer Trendgrafiken der Gaskonzentrationen.
- Einstellungsmenü: Ermöglicht Anwendern die Anpassung von Systemparametern und Funktionsschaltern.

Am oberen Rand der Hauptansicht befindet sich eine Statusleiste, die einen schnellen Überblick über kritische Informationen wie Zeit, Temperatur und Systemstatus bietet.

Dynamischer Aktualisierungsmechanismus

Zur Optimierung der Anzeigeperformance und Verbesserung der Benutzererfahrung wurde ein effizienter dynamischer Aktualisierungsmechanismus entwickelt. Dieser basiert auf folgenden Kernstrategien:

- Bedingte Aktualisierung: Nur bei Datenänderungen wird der entsprechende Displaybereich aktualisiert, um redundante Aktualisierungsvorgänge zu minimieren.

- Partielle Aktualisierung: Durch präzise Lokalisierung der zu aktualisierenden Displaybereiche wird der Aktualisierungsumfang minimiert und die Effizienz gesteigert.
- Forciertes Aktualisierungszeitfenster: Innerhalb eines definierten Zeitraums nach einer Datenänderung wird eine Aktualisierung erzwungen, um die zeitnahe Aktualisierung kritischer Informationen sicherzustellen.

Dieser Mechanismus reduziert effektiv den Systemressourcenverbrauch bei gleichzeitiger Gewährleistung der Echtzeitanzeige.

Konzeption der Hauptansicht

Die Hauptansicht bildet das primäre Interface für Benutzerinteraktionen. Ihr Design fokussiert sich auf die klare Darstellung und schnelle Erfassung von Informationen. Zentrale Merkmale umfassen:

- Gaskonzentrationsanzeige: Echtzeitkonzentrationsdaten von sechs Gasen werden übersichtlich von oben nach unten angeordnet.
- Farbcodierung: Verwendung unterschiedlicher Farben (Grün, Gelb, Rot) basierend auf den Gaskonzentrationsniveaus zur intuitiven Gefahrengradvisualisierung.
- Alarmindikation: Anzeige eines prägnanten roten Warndreiecks bei Überschreitung der Gaskonzentrationsniveaus.

Konzeption der Nebenansicht

Die Nebenansicht dient primär zur Darstellung historischer Trends der Gaskonzentrationen und nutzt eine prägnante, effektive grafische Repräsentationsmethode:

- Multi-Koordinatensystem-Layout: Simultane Visualisierung von Konzentrationskurven für sechs Gase zur erleichterten Vergleichs- und Analysearbeit.
- Dynamische Aktualisierung: Echtzeit-Update der Kurvendaten für kontinuierliche Trendvisualisierung.
- Adaptive Koordinaten: Automatische Skalierung der Koordinatenachsen basierend auf dem Datenbereich zur Sicherstellung optimaler Kurvensichtbarkeit.

Konzeption des Einstellungsmenüs

Das Einstellungsmenü bietet Funktionen zur benutzerdefinierten Anpassung von Systemparametern. Seine Konzeption fokussiert sich auf intuitive Bedienung und unmittelbares Feedback:

- Listenbasiertes Layout: Übersichtliche Darstellung aller anpassbaren Parameter auf einer Oberfläche ohne Scrollnotwendigkeit.
- Echtzeit-Feedback: Unmittelbare Anzeige neuer Werte nach Parameteranpassungen für intuitives Bedienungsfeedback.
- Hervorhebung selektierter Elemente: Einsatz von Animationseffekten zur Akzentuierung des aktuell gewählten Einstellungselements.

Konzeption der Statusleiste

Die Statusleiste als integraler Bestandteil der Hauptansicht bietet einen Schnellüberblick über kritische Systeminformationen:

- Zeit- und Temperaturanzeige: Echtzeit-Updates für Umgebungsinformationen.
- Systemstatussymbole: Inklusive Alarmstatus, WLAN-Verbindung, SD-Kartenstatus und Energiestatus.
- Farbcodierung: Nutzung verschiedener Hintergrundfarben der Statusleiste zur Indikation des Gesamtsystemstatus, z.B. Grün für Normalbetrieb, Rot für Warnzustand.

Konzeption grafischer Elemente

Zur Steigerung der Lesbarkeit und Ästhetik der Benutzeroberfläche wurde eine Reihe grafischer Elemente sorgfältig konzipiert:

- Maßgeschneiderte Icons: Sämtliche Icons wurden vom Autor pixelgenau manuell entworfen, darunter Statusanzeigen für das akustische und optische Alarmsystem, WLAN, SD-Karte und Batterie, um sowohl die Informationserkennbarkeit als auch die visuelle Anmutung zu optimieren.
- Dynamischer Iconstatus: Füllstand und Farbe des Batteriesymbols variieren dynamisch entsprechend der verbleibenden Ladung; Alarm- und WLAN-Symbole passen sich dynamisch dem aktuellen Status an.
- Präzise Pixelsteuerung: Durch exaktes pixelgenaues Rendering werden reichhaltige visuelle Effekte auf begrenzter Displayfläche realisiert.

5.3.8 Datenverwaltung und -speicherung

Die Konzeption des Datenverwaltungs- und Speichermoduls ist von entscheidender Bedeutung für die Sicherstellung der Zuverlässigkeit, Persistenz und Zugänglichkeit der Daten. Dieser Abschnitt erläutert detailliert die Konzeption und Implementierung der Datenverwaltung und -speicherung des Systems, einschließlich des Datenprotokollierungsmechanismus, der Speicherplatzverwaltungsstrategie und der Datenexportmethoden. Diese Komponenten bilden gemeinsam ein umfassendes Datenverarbeitungssystem, das eine effiziente Aufzeichnung, intelligente Verwaltung sowie benutzerfreundlichen Zugriff und Analyse der Gaskonzentrationsdaten ermöglicht.

Datenspeicherung

Das System nutzt eine SD-Karte als primären Datenspeicher. Mittels der Funktion `logSensorData()` werden die vom Sensor erfassten Gaskonzentrationsdaten in regelmäßigen Intervallen in Logdateien auf der SD-Karte geschrieben. Jeder Datensatz umfasst die Systemlaufzeit, den aktuellen Zeitstempel (sofern ein externes RTC-Modul angeschlossen ist), Temperaturinformationen sowie die Konzentrationswerte von sechs Gasen. Diese strukturierte Datenspeichermethode erleichtert die nachfolgende Datenanalyse und -verarbeitung erheblich.

Datenverwaltung

Zur Gewährleistung einer effizienten Nutzung des Speicherplatzes implementiert das System einen automatischen Speicherverwaltungsmechanismus. In regelmäßigen Abständen wird die Funktion `checkFreeSpaceAndManageLogs()` aufgerufen, um den verfügbaren Speicherplatz auf der SD-Karte zu überprüfen. Unterschreitet der prozentuale Anteil des freien Speicherplatzes einen vordefinierten Schwellenwert, initiiert das System automatisch die Funktion `deleteOldestLogFile()`, um die älteste Logdatei zu löschen und somit Speicherplatz freizugeben. Dieser Mechanismus gewährleistet einen langfristigen, unterbrechungsfreien Betrieb des Systems, ohne dass die Datenaufzeichnung aufgrund von Speicherplatzmangel zum Erliegen kommt.

Zusätzlich bietet das System eine Funktion zur Speicherung von Konfigurationsdateien. Die Funktion `saveConfigToSD()` speichert die aktuellen Systemeinstellungen im JSON-Format auf der SD-Karte. Dies umfasst Parameter wie Gasbezeichnungen, Minimal- und Maximalwerte sowie Alarmschwellenwerte für die verschiedenen Gase. Diese Funktionalität ermöglicht eine persistente Speicherung der Systemkonfiguration und erleichtert die Wiederherstellung vorheriger Einstellungen nach einem Systemneustart.

Datenexport

Neben der Speicherung auf der SD-Karte exportiert das System Gaskonzentrationsdaten in Echtzeit über die serielle Schnittstelle. Die Funktion `serialPrintf` formatiert die Ausgabe der Konzentrationswerte der sechs Gase, was eine Echtzeitüberwachung und effiziente Fehlerdiagnose ermöglicht. Diese Echtzeit-Datenexportfunktion spielt eine zentrale Rolle bei der Vor-Ort-Analyse und der Evaluierung der Systemleistung.

5.3.9 Drahtlose Kommunikation und Datenvisualisierung

Dieser Abschnitt erläutert die Konzeption und Implementierung der drahtlosen Kommunikation und Datenvisualisierung des Systems, einschließlich des WLAN-Verbindungsmechanismus, der Anwendung des MQTT-Protokolls und der Integration der Thingsboard-Plattform. Diese Komponenten bilden gemeinsam ein umfassendes System für Datenübertragung und -darstellung, das eine Echtzeitübertragung und Fernzugriff auf Gasetektionsdaten ermöglicht und diese dem Anwender auf intuitive Weise präsentiert.

WLAN-Verbindung

Das System nutzt WLAN als Basis für die drahtlose Kommunikation, hauptsächlich aufgrund seiner weiten Verbreitung und einfachen Integration. Ein automatischer Reconnect-Mechanismus wurde implementiert, um die Zuverlässigkeit des Systems bei instabilen Netzwerkbedingungen zu gewährleisten.

MQTT-Protokoll

Das System verwendet das MQTT-Protokoll für die Datenübertragung, basierend auf dessen Eigenschaften wie geringer Overhead, niedriger Bandbreitennutzung und Unterstützung des Publish/Subscribe-Modells, was es ideal für IoT-Anwendungsszenarien macht. Die MQTT-Implementierung erfolgt primär über die Thingsboard-Client-Bibliothek, wodurch die spezifischen Implementierungsdetails des Protokolls abstrahiert werden.

Thingsboard-Plattform

Thingsboard, als Open-Source-IoT-Plattform für Datenvisualisierung und Geräteverwaltung, bietet leistungsstarke Frontend- und Backend-Unterstützung für das Modul zur drahtlosen Kommunikation und Datenvisualisierung. Die Interaktion des Systems mit Thingsboard umfasst hauptsächlich folgende Aspekte:

1. Daten-Upload:

```
"tb.sendTelemetryData(GAS_NAME[i], gasConcentrations[i]);"
```

Dieser Code-Abschnitt überträgt in regelmäßigen Intervallen Gaskonzentrationsdaten als Telemetriedaten an die Thingsboard-Plattform.

2. Attributsynchronisation:

```
"tb.sendAttributeData(SUBSCRIBED_SHARED_ATTRIBUTES[i], settings[i].threshold);"
```

Das System synchronisiert periodisch lokale Einstellungen mit Thingsboard, um eine Fernüberwachung des Gerätestatus zu ermöglichen.

3. Fernsteuerung:

```
"const Shared_Attribute_Callback<MAX_ATTRIBUTES> callback(&processSharedAttributeUpdate, SUBSCRIBED_SHARED_ATTRIBUTES.cbegin(), SUBSCRIBED_SHARED_ATTRIBUTES.cend());"
```

Durch das Abonnieren von Aktualisierungen gemeinsam genutzter Attribute implementiert das System eine Ferneinstellung von Parametern und Funktionsschaltern.

Diese Architektur befähigt das System zur Echtzeitübertragung von Sensordaten und ermöglicht gleichzeitig Fernüberwachung und -steuerung. Dies steigert signifikant die Verwaltbarkeit des Systems und den Grad der Datenvisualisierung.

5.3.10 Erfassung von Zeit und Temperatur

Die präzise Erfassung von Zeit- und Temperaturinformationen liefert essentielle Zusatzdaten für die Datenprotokollierung und Umgebungsüberwachung des Gasdetektionssystems. Dieser Abschnitt erläutert die Methodik zur Akquisition von Zeit- und Temperaturdaten mittels eines externen Echtzeituhr-Moduls (RTC) sowie die entsprechenden Verarbeitungs- und Speicherverfahren.

Das System nutzt ein externes RTC-Modul zur Bereitstellung präziser Zeit- und Temperaturinformationen. Diese Herangehensweise bietet signifikante Vorteile hinsichtlich Genauigkeit, Energieeffizienz und Datenpersistenz. Durch die Implementation der spezialisierten Funktion `getRealTimeAndTempFromOutRTC()` realisiert das System eine effiziente Erfassung und Verarbeitung der Zeit- und Temperaturdaten.

Die Kernkonzepte dieser Funktion umfassen Datennormalisierung, Fehlerprävention und globale Zugriffsmöglichkeit. Zeitinformationen werden in ein standardisiertes Format konvertiert und separat gespeichert, was die Flexibilität der Datennutzung erhöht. Temperaturdaten werden direkt vom RTC-Modul ausgelesen und bieten Echtzeit-Datenunterstützung für das

Umgebungsmonitoring. Durch die Speicherung der Daten in globalen Variablen gewährleistet das System die unmittelbare Verfügbarkeit der Informationen.

Dieser modulare Ansatz steigert nicht nur die Wiederverwendbarkeit und Wartbarkeit des Codes, sondern eröffnet auch Potenziale zur Leistungsoptimierung des Systems. Durch die gezielte Steuerung der Funktionsaufruffrequenz kann das System eine optimale Balance zwischen der Aktualität der Datenaktualisierung und dem Ressourcenverbrauch erreichen.

5.3.11 Energiemanagement

Bei batteriebetriebenen Systemen ist die Bedeutung des Energiemanagements von zentraler Wichtigkeit. Dieser Abschnitt fokussiert sich auf die Implementierungsmethoden zur Schätzung der Batteriekapazität und Zustandsüberwachung.

Schätzung der Batteriekapazität

Das System implementiert eine spannungsbasierte, stückweise lineare Interpolationsmethode zur Schätzung der Batteriekapazität. Diese Methode berücksichtigt die nichtlinearen Charakteristika der Entladekurve von 9V-Alkalibatterien, indem der Spannungsbereich in mehrere Intervalle unterteilt wird, innerhalb derer jeweils eine lineare Interpolation angewendet wird. Dies ermöglicht eine relativ präzise Kapazitätsschätzung. Die Methode stellt einen Kompromiss zwischen Recheneffizienz und Schätzgenauigkeit dar und ist optimal auf die Ressourcenlimitierungen eingebetteter Systeme abgestimmt.

Batteriezustandsüberwachung

Fundament der Batteriezustandsüberwachung ist eine hochpräzise Spannungsmessung. Das System nutzt die kontinuierliche Abtastfunktion des ADC des ESP32 für eine effiziente Analog-Digital-Wandlung. Durch multiple konsekutive Abtastungen erzielt das System stabilere Spannungsmesswerte. Ein Mittelwertfilter-Algorithmus wird zur Verarbeitung der Abtastdaten eingesetzt, um Rauscheinflüsse effektiv zu minimieren und die Messgenauigkeit zu optimieren. Nach Erfassung der ADC-Messwerte wird die tatsächliche Batteriespannung unter Berücksichtigung des Spannungsteiler-Prinzips kalkuliert. Dieses Konzept ermöglicht nicht nur die Überwachung des Batterieladezustands, sondern auch die Detektion des Batterieanschlusstatus, was die Systemzuverlässigkeit signifikant erhöht. Zusätzlich wurde post Spannungsberechnung eine optionale Temperaturkompensationsfunktion implementiert, um eine noch präzisere Batteriespannungsbestimmung zu gewährleisten.

Das System implementiert zudem eine mehrstufige Ladezustandsanzeige und Warnfunktion, die Alarme generiert, sobald der Batterieladezustand definierte Schwellenwerte unterschreitet. Dies bietet dem Anwender zeitkritische Informationen über den Batteriezustand und präveniert Datenverluste oder Systeminterruptionen aufgrund von Batterieerschöpfung.

5.3.12 Task-Scheduler

Das System verwendet einen auf FreeRTOS basierenden Task-Scheduler, dessen Architektur vom Konzept der Daemon-Prozesse in Linux-Betriebssystemen inspiriert ist. Dieser Scheduler

operiert als permanenter Task und ist für das Management und die Koordination der Ausführungszustände anderer System-Tasks verantwortlich.

Die Kernfunktionalität des Schedulers liegt in der dynamischen Verwaltung kritischer Tasks, aktuell fokussiert auf das Management des Alarm-Tasks (AlarmTask). Durch zyklische Evaluation der Aktivierungsflags und Ausführungszustände dieser Tasks ermöglicht er die dynamische Generierung, Terminierung und Umschaltung von Tasks. Diese Konzeption steigert die Flexibilität und Ressourceneffizienz des Systems signifikant. Der Scheduler determiniert basierend auf dem Status des isAlarmTaskEnabled-Flags, ob der AlarmTask zu initiieren oder zu terminieren ist. Bei Beendigung des Alarm-Tasks ist er ebenfalls für die Deaktivierung relevanter Hardware wie Buzzer und LED-Indikatoren verantwortlich, um die Konsistenz des Systemzustands zu garantieren.

6 Systemaufbau und Inbetriebnahme

In diesem Kapitel wird der Inbetriebnahmeprozess des Schadstoffdetektionssystems detailliert erläutert. Dies umfasst die Hardwaremontage, Softwareimplementierung sowie die Herausforderungen und deren Lösungen während dieses Prozesses.

6.1 Systemimplementierung

Die Hardwaremontage konzentriert sich hauptsächlich auf den Funktionstest der Leiterplatten und die Integration der verschiedenen Module. Die Softwareimplementierung beinhaltet das Flashen und Debuggen der Firmware des eingebetteten Systems sowie den Kommunikationstest mit der Host-Software (Arduino IDE Serial).

6.1.1 Hardwaremontage

Nach der Fertigstellung der PCB wurden folgende Schritte zur Hardwaremontage durchgeführt:

1. Komponentenvorbereitung: Alle für die Montage erforderlichen elektronischen Bauteile und Module wurden gemäß der Komponentenliste (siehe Anhang "Komponentenliste") bereitgestellt.
2. Lötsequenzplanung: Die Lötung erfolgte systematisch von kleinen zu großen Komponenten und von Niedrigtemperatur- zu Hochtemperaturkomponenten.
3. Löten der THT-Komponenten: Da ausschließlich THT-Komponenten verwendet wurden, erfolgte die Lötung aller Bauteile manuell mit einem Lötkolben.
4. Reinigung: Aufgrund der minimalen Rückstände von Flussmittel und aus Umweltschutzgründen konnte auf eine Reinigung der PCB mit Isopropanol verzichtet werden.
5. Qualitätskontrolle: Sämtliche Lötstellen wurden mittels Lupe und Multimeter auf Kurzschlüsse und kalte Lötstellen überprüft.
6. Funktionstest: Ein umfassender Funktionstest der montierten PCB wurde nach einem vorab erstellten Testprotokoll durchgeführt.

Abbildung 6-1 zeigt das vollständig montierte System, einschließlich des Hauptsystems und des externen Stromversorgungsmoduls.

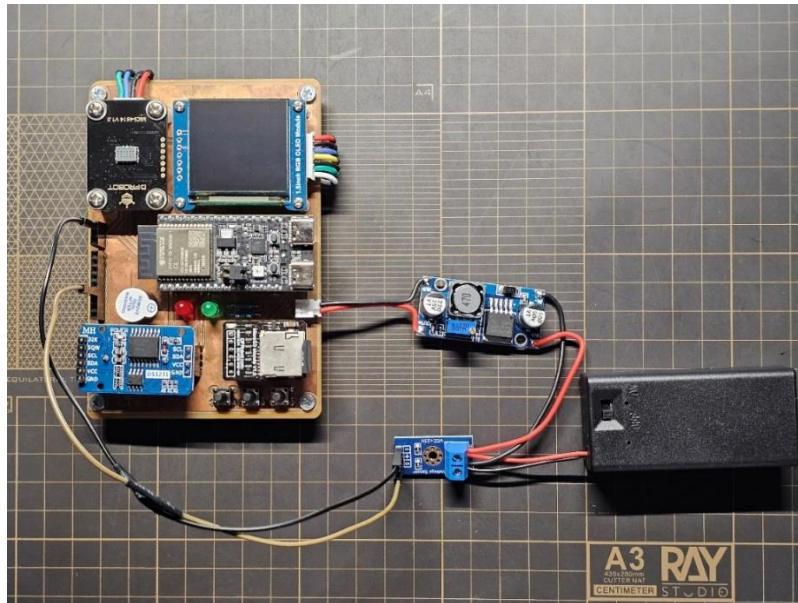


Abbildung 6-1 Montiertes System

Das modulare Design des Systems nutzt verschiedene Verbindungstechniken. Sensormodule, OLED-Display und Stromversorgungsschnittstelle sind über selbst gefertigte kurze PH2.0-Kabel mit PH2.0-Steckverbindern angeschlossen. Das ESP32-Entwicklungsboard, das SD-Modul und das RTC-Modul sind mittels 2,54-mm-Buchsenleisten mit der PCB verbunden. Diese modulare Bauweise erleichtert das separate Testen oder Austauschen von Hardwarekomponenten während der Entwicklungsphase. Nicht genutzte GPIO-, 5V-, 3,3V- und GND-Anschlüsse sind über 2,54-mm-Buchsenleisten auf der linken Seite der PCB zugänglich. Die rechte Seite der PCB beherbergt einheitlich die USB-Stromversorgung, den Batterieanschluss und den SD-Kartenslot. Weitere Komponenten sind direkt auf die PCB gelötet.

Die Kalibrierung und Leistungsprüfung des Step-Down-Moduls erforderte den Einsatz eines Präzisionsmultimeters. Initial wurde der Ausgang des Step-Down-Moduls justiert. Hierfür diente eine neue 9V-Alkalibatterie (Anfangsspannung 9,44V) als Stromquelle am Eingang des Step-Down-Moduls. Die Ausgangsspannung wurde manuell auf einen Wert möglichst nahe 5V eingestellt, wobei eine finale Ausgangsspannung von 5,0145V erreicht wurde (siehe Abbildung 6-2).

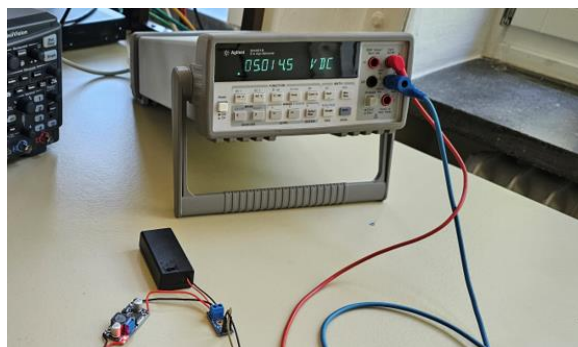


Abbildung 6-2 Manuelle Justierung des Step-Down-Modulausgangs

Anschließend wurde der Spannungsabfall am Eingang unter Last evaluiert. Das System wurde mit der Batterie in Kombination mit dem Step-Down-Modul versorgt, und die Batteriespannung wurde mit einem Multimeter gemessen. Dabei wurde ein Wert von 8,7371V ermittelt. Die Ausgangsspannung betrug 5,0132V, was nahezu identisch mit dem initialen Wert von 5,0145V war.

Zur weiteren Untersuchung der Stabilität des Step-Down-Moduls bei variierenden Eingangsspannungen kam ein Labornetzgerät zum Einsatz. Verschiedene Eingangsspannungen wurden eingestellt und die korrespondierenden Ausgangsspannungen erfasst, um die Zuverlässigkeit des Step-Down-Moduls zu beurteilen. Die Eingangsspannungen wurden auf 10V, 9V, 8V, 7V, 6V, 5,6V, 5,5V, 5,4V, 5,3V, 5,2V und 5V eingestellt. Dies resultierte in Ausgangsspannungen von 5,015V, 5,014V, 5,011V, 5,007V, 4,996V, 4,976V, 4,888V, 4,789V, 4,690V, 4,591V und 4,394V (siehe Abbildung 6-3).

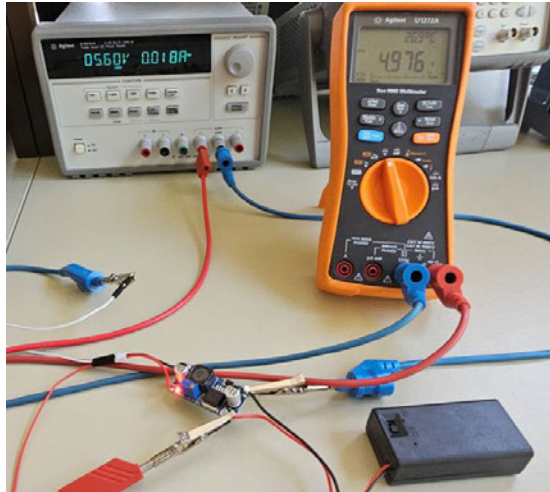


Abbildung 6-3 Stabilitätstest der Ausgangsspannung des Step-Down-Moduls

Zusammenfassend lässt sich konstatieren, dass das LM2596-Step-Down-Modul für die Stromversorgung dieses Systems eine adäquate Stabilität aufweist. Um eine stabile Stromversorgung im Batteriebetrieb zu gewährleisten, wird empfohlen, sicherzustellen, dass die Eingangsspannung des Step-Down-Moduls nicht unter 5,6V bis 6V abfällt. Diese Erkenntnisse liefern zudem wertvolle empirische Daten für die Festlegung des minimalen Batteriespannungsschwellwerts in der Softwarekonfiguration.

6.1.2 Softwareimplementierung

Dieser Abschnitt erläutert detailliert den Implementierungsprozess der Detektionssystem-Software und der IoT-Cloud-Plattform, einschließlich der Softwareinstallation, Teststrategien und Plattformkonfiguration.

6.1.2.1 Implementierung der Detektionssystem-Software

Die Implementierung der Detektionssystem-Software folgte einem rigorosen Prozess, um höchste Zuverlässigkeit und Funktionsintegrität zu gewährleisten. Die spezifischen Schritte umfassten:

1. Firmware-Flashen

- Die entwickelte Firmware wurde auf den ESP32-C6-Mikrocontroller aufgespielt, um die Basis für weitere Operationen zu schaffen.

2. Unit-Tests

Umfassende Unit-Tests wurden für diverse Softwaremodule durchgeführt, mit Schwerpunkt auf folgenden Aspekten:

- Sensordatenerfassung: Verifizierung der Präzision und Stabilität der Datenauslesung
- Datenverarbeitungsalgorithmen: Sicherstellung, dass die Algorithmusausgaben den Erwartungen entsprechen und Gefahrensituationen zuverlässig erkennen
- Datenspeicherfunktion: Evaluation der Datenspeicherungs- und -abruffunktionen
- Benutzeroberfläche und Interaktion: Prüfung der Robustheit und Zuverlässigkeit der Systeminteraktion

3. Integrationstests

Im Anschluss an die Unit-Tests wurden umfangreiche Systemintegrationstests durchgeführt, mit Fokus auf:

- Synergie zwischen Hardware und Software
- Langzeitstabilität unter realen Betriebsbedingungen

4. Leistungsoptimierung

Basierend auf den Testergebnissen wurden gezielte Feinabstimmungen und Optimierungen vorgenommen, um die Gesamtleistung und Zuverlässigkeit zu maximieren.

Durch diese Sequenz rigoroser Implementierungs- und Testprozesse wurde die stabile und effiziente Funktionsweise des Detektionssystems sichergestellt.

6.1.2.2 Implementierung der IoT-Cloud-Plattform

Die Implementierung der IoT-Cloud-Plattform basierte auf einer flexiblen lokalen Teststrategie, um höchste Zuverlässigkeit und Leistungsfähigkeit zu gewährleisten. Der Implementierungsprozess gestaltete sich wie folgt:

1. Auswahl der Implementierungsumgebung

Für diese Studie wurden Implementierung und Tests auf einem persönlichen Home-Server und Laptop durchgeführt:

- Home-Server: Nutzung der LXC-Container-Technologie in der Proxmox Virtual Environment zur Generierung einer Debian 11-Systemumgebung, in der eine Docker-Umgebung installiert wurde
- Laptop: Basierend auf einem Windows 11-Host-System in Kombination mit der Docker Desktop-Umgebung

2. Plattformimplementierung

- Einsatz des offiziellen Docker-Images thingsboard/tb-postgres von ThingsBoard für die Implementierung, unter Verwendung von Docker Compose für eine automatisierte Bereitstellung, was den Konfigurationsprozess erheblich vereinfachte.

3. Plattformkonfiguration

Nach erfolgreicher Implementierung der ThingsBoard-Plattform wurden essentielle Konfigurationen vorgenommen, mit besonderem Augenmerk auf den "Save Client Attributes"-Knoten in der Regelkette:

- Konfiguration von "Entity attributes scope" auf "Shared attributes", wodurch die Attributwerte aus eingehenden Nachrichten als gemeinsame Attribute (Shared attributes) der Entität gespeichert werden, anstatt als Client-Attribute (Client attributes). Dies ermöglicht den späteren Zugriff und die Modifikation durch serverseitige Anwendungen.
- Optional kann "Save attributes only if the value changes" aktiviert werden, um die Datenspeicherung zu optimieren.

Die Regelketten-Konfigurationsdatei (im .json-Format) wurde als Anlage im Projektordner zur Verfügung gestellt, um zukünftige Reproduktion und Modifikation zu erleichtern.

4. Dashboard-Erstellung

Zur Realisierung aller in der Programmkonzeption vorgesehenen Interaktionsfunktionen wurden multiple Widgets manuell integriert und konfiguriert. Abbildung 6-4 illustriert einen Ausschnitt der Funktionalitäten des finalen Dashboards. Die vollständige Dashboard-Konfigurationsdatei (im .json-Format) wurde als Anlage im Projektordner bereitgestellt, um zukünftige Reproduktion und Erweiterung zu ermöglichen.

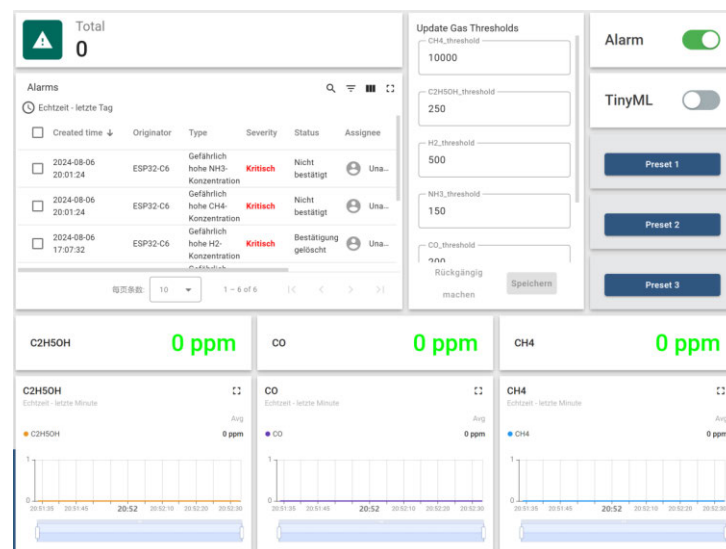


Abbildung 6-4 Partielle Funktionsdarstellung des ThingsBoard-Dashboards

5. Funktionstest

- In der Testphase wurden sämtliche Widgets des Dashboards einer umfassenden Prüfung unterzogen, um zu verifizieren, dass die Funktionen jeder Komponente den Spezifikationen entsprechen und die Benutzeroberfläche ein hohes Maß an Usability und Reliabilität aufweist.

Durch diesen Ansatz der lokalen Implementierung und Testdurchführung wurde nicht nur die funktionale Integrität der IoT-Cloud-Plattform verifiziert, sondern auch ein solides Fundament für zukünftige Systemerweiterungen und Optimierungen geschaffen.

6.1.3 Stromverbrauchsanalyse

Nach erfolgreichem Abschluss der integrierten Hard- und Softwaretests wurde eine umfassende Stromverbrauchsanalyse des Systems durchgeführt. Für diese Untersuchung wurde die USB-Schnittstelle zur Stromversorgung genutzt, wobei die auf dem ESP32-C6-Entwicklungsboard integrierte Strommessschnittstelle zum Einsatz kam. Durch mehrfache Messungen während des Systembetriebs und anschließende Mittelwertbildung wurde ermittelt, dass bei einer Eingangsspannung von 5V der Gesamtstromverbrauch 240mA beträgt (siehe Abbildung 6-5).

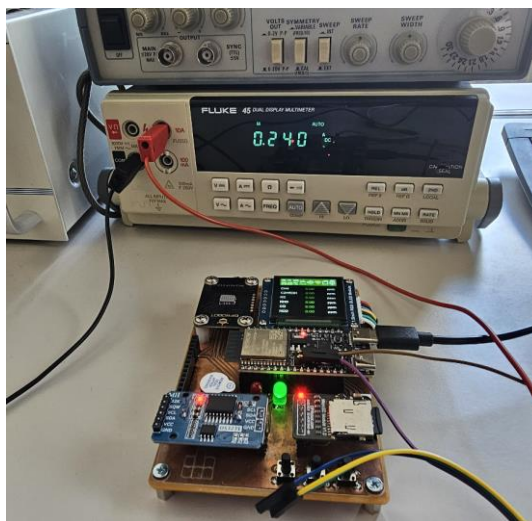


Abbildung 6-5 Messung des Gesamtstromverbrauchs des Systems

In einer schrittweisen Analyse wurde zunächst die Stromversorgung des Displaymoduls unterbrochen, wodurch der Gesamtstromverbrauch auf 200mA sank. Bei zusätzlicher Deaktivierung der RTC- und SD-Module reduzierte sich der Stromverbrauch weiter auf 196mA. Im finalen Schritt, nach Abtrennung des Sensormoduls und bei ausschließlichem Betrieb des ESP32, belief sich der Stromverbrauch auf lediglich 32mA. Unter Berücksichtigung der Verluste durch den On-Board 3,3V LDO lässt sich der Stromverbrauch des ESP32-C6-Entwicklungsboards bei Ausführung der implementierten Software auf circa 150mW schätzen. Der Gassensor weist einen Verbrauch von 800mW auf, während das Displaymodul bei aktiver Benutzeroberfläche 200mW benötigt. Die SD- und RTC-Module zusammen verbrauchen mit 20mW vergleichsweise wenig Energie.

Der Gesamtstromverbrauch des vollständigen Systems beläuft sich auf 1,2W, wobei der Gassensor als Hauptverbraucher identifiziert wurde. Es wird vermutet, dass insbesondere das integrierte Heizmodul für den erhöhten Energiebedarf verantwortlich ist. Eine 9V-Alkalibatterie mit einer typischen Kapazität von etwa 300mAh erweist sich als unzureichend, um einen 8-stündigen Systembetrieb zu gewährleisten. Für zukünftige Optimierungen sollte die Integration einer hochkapazitiven, wiederaufladbaren Lithium-Ionen-Batterie als alternative Energiequelle in Betracht gezogen werden, um eine längere Betriebsdauer zu ermöglichen und die Energieeffizienz des Gesamtsystems zu steigern.

6.2 Probleme und deren Lösungsansätze

Im Verlauf der Systemabstimmung und -prüfung traten einige Herausforderungen zutage, die sich vorwiegend auf zwei Bereiche konzentrierten: die Leiterplatte und die Sensorleistung.

6.2.1 PCB-bezogene Problematik

Bei der Funktionsprüfung der Leiterplatte wurde eine Anomalie festgestellt, die während der vorherigen Testphase auf dem Steckbrett nicht in Erscheinung getreten war: Bei der Verbindung des ESP32 über USB mit dem PC konnte der Geräte-Manager die serielle Schnittstelle nicht detektieren. Eine akribische Untersuchung führte zur Identifikation der Problemursache: Ein äußerst feiner Kupferdraht, mit hoher Wahrscheinlichkeit ein Residuum des CNC-Fertigungsprozesses, hatte einen Kurzschluss zwischen dem D-Minus-Pin des USB und der GND-Kupferfläche verursacht, was die Kommunikation beeinträchtigte.

Lösungsansatz:

Durch gezielte Beleuchtung der Leiterplattenrückseite mittels einer Taschenlampe gelang es, den mit bloßem Auge kaum wahrnehmbaren Kupferdraht zu lokalisieren. Nach behutsamer Entfernung des Drahtes konnte die volle Funktionalität des Systems wiederhergestellt werden. Die erfolgreiche Bewältigung dieser Herausforderung unterstreicht nicht nur die Relevanz minutiöser Beobachtung im Rahmen des Hardware-Debugging-Prozesses, sondern akzentuiert auch die Notwendigkeit einer umfassenden Qualitätskontrolle im Anschluss an die PCB-Fertigung.

6.2.2 Herausforderungen bei der Sensorleistung

Die Entscheidung für den MiCS-4514 MOS MEMS-Sensor basierte auf einer ganzheitlichen Betrachtung von Kosten, Reaktionszeit und Erfassungsbereich. Praktische Tests offenbarten jedoch signifikante Defizite bei der Gasdifferenzierung. Diese manifestierten sich in der Tatsache, dass bei der Detektion eines spezifischen Gases auch die Messwerte anderer Gase Fluktuationen aufwiesen, was das Risiko von Fehlalarmen in komplexen Umgebungen substantziell erhöhte.

In einem exemplarischen Versuch wurde 99,9%iger Ethanol in 5 cm Distanz zum Sensor positioniert und die vom System generierten Messwerte analysiert. Die Resultate zeigten nicht nur Ausschläge für C_2H_5OH (Ethanol), sondern auch für CH_4 (Methan). Aufgrund des höheren Messbereichs des MiCS-4514 für CH_4 im Vergleich zu C_2H_5OH besteht die Gefahr, dass das System irrtümlich einen CH_4 -Alarm triggert, anstelle des antizipierten C_2H_5OH -Alarms. Diese Beobachtung korrespondiert mit der Analyse von Abbildung 5-9 in Abschnitt 5.2.1.2.

Lösungsstrategien:

Zur Bewältigung dieser Problematik wurden zwei potenzielle Lösungsansätze evaluiert:

1. Migration zu elektrochemischen Sensoren: Diese offerieren eine präzisere Gasdifferenzierung und höhere Genauigkeit. Allerdings überstiegen die damit verbundenen Kosten (mindestens ca. 70 Euro pro Sensor, kumulativ 420 Euro) das vorgesehene Projektbudget.

Zudem weisen elektrochemische Sensoren signifikant längere Reaktionszeiten als MOS-Sensoren auf.

2. Optimierung der Software-Algorithmen: Beispielsweise durch die Integration von Machine-Learning-Techniken zur Unterscheidung diverser Gase.

Letztendlich fiel die Entscheidung zugunsten einer softwarebasierten Lösung. Inspiriert durch Roni Bandinis bhopal84-Projekt (<https://github.com/ronibandini/bhopal84>) wurde die TinyML-Technologie implementiert, um die Fähigkeit des Systems zur Gasdifferenzierung zu potenzieren. TinyML repräsentiert ein emergentes Edge-Computing-Konzept, das auf extrem energieeffiziente, kostengünstige, leistungsstarke und datenschutzfreundliche Maschinenlern-Inferenz abzielt.[35] Dieser Ansatz konserviert nicht nur die Kosteneffizienz und Reaktionsschnelligkeit der originären Hardware, sondern steigert auch signifikant die Erkennungsgenauigkeit des Systems. TinyML eignet sich besonders für ressourcenlimitierte Mikrocontroller wie den in diesem Projekt eingesetzten ESP32. Durch Techniken wie Quantisierung, Pruning und Clustering ermöglicht TinyML die Exekution komplexer Maschinenlernmodelle unter Berücksichtigung limitierter Rechen- und Speicherressourcen.[36] Dies harmonisiert mit den aktuellen Systemanforderungen und ermöglicht eine substantielle Verbesserung der Gaserkennungspräzision und der Gesamtleistung des Systems ohne zusätzliche Hardwareinvestitionen.

6.2.2.1 Datenakquisition

Die Datenerfassungssoftware basiert auf einer optimierten Version der Schleifenfunktion aus der Datei "gasAcquisition2.ino" des bhopal84-Projekts. Die Kernfunktionalitäten umfassen:

- Periodische Auslesung von Konzentrationsdaten für sechs Gase (CH_4 , $\text{C}_2\text{H}_5\text{OH}$, H_2 , NH_3 , CO , NO_2) vom MiCS-4514-Sensor.
- Applikation eines gewichteten gleitenden Mittelwerts (WMA) zur Filterung der Rohdaten zwecks Rauschminimierung.
- Aggregation multipler Proben innerhalb eines prädefinierten Zeitfensters (in dieser Studie standardmäßig 1500 ms).
- Kalkulation von Durchschnitt, Minimum und Maximum der Gaskonzentrationen für jedes Zeitfenster.
- Persistierung der prozessierten Daten in einer CSV-Datei auf der SD-Karte bei simultaner Visualisierung relevanter Informationen auf dem OLED-Display.

Diese elaborierte Datenerfassungsmethodik garantiert die Akquisition hochwertiger, kontinuierlicher Gaskonzentrationsdaten und etabliert ein solides Fundament für das subsequente Modelltraining.

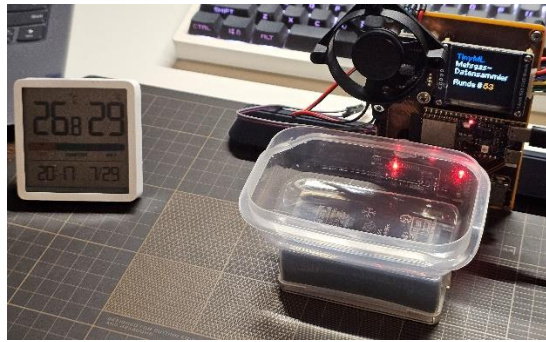


Abbildung 6-6 Datenakquisition

6.2.2.2 Modelltraining

Das Modelltraining stellt einen kritischen Schritt zur Optimierung der Erkennungsfähigkeit gefährlicher Substanzen dar. Nach eingehender Analyse wurde der Classification (Keras) Algorithmus der Edge Impulse Plattform als optimal identifiziert. Bei der Konzeption des Trainingsprozesses lag der Fokus auf der zeitlichen Sequenz der Daten, der Modellkomplexität sowie den Ressourcenlimitationen des eingebetteten Systems.

In der Phase der Datenvorverarbeitung wurde eine Abtastrate von 0,67 Hz implementiert. Diese Frequenz wurde aus dualer Motivation gewählt: Sie ermöglicht einerseits die Erfassung dynamischer Charakteristika der Gaskonzentrationsfluktuationen, limitiert andererseits die Datenpunktgenerierung, um eine Überlastung des eingebetteten Systems zu vermeiden. Die Entscheidung fiel auf die Verwendung der Rohdaten aller Achsen als Input, um ein Maximum an Informationen zu konservieren und dem Modell die autonome Extraktion der relevantesten Features zu ermöglichen.

Für die Modellarchitektur wurde ein Keras-basierter Klassifizierungsalgorithmus selektiert. Keras offeriert flexible Tools zur Konstruktion neuronaler Netze bei simultaner effizienter Verarbeitung von Zeitreihendaten. Der Modell-Output wurde in zwei Kategorien stratifiziert: regular (normale Gase) und harmful (schädliche Gase). Diese binäre Klassifikationsstruktur simplifiziert die Problemstellung und fokussiert das Modell auf die Differenzierung zwischen potenziell gefährlichen und unbedenklichen Gasen.

Die Akquisition der Rohdaten präsentierte diverse Herausforderungen, insbesondere im Kontext schädlicher Gase. Unter Berücksichtigung von Sicherheitsaspekten und Realisierbarkeit wurden folgende Datenquellen herangezogen:

- Schädliche Gase (harmful-Label): 99,9%iger Ethanol fungierte als primäre Datenquelle. Trotz der Limitation in der Repräsentation aller schädlichen Gase bietet dies eine sichere und kontrollierbare Simulationsumgebung.
- Normale Gase (regular-Label):
 1. Utilisation von Gasen, die aus alltäglichen Flüssigkeiten (wie Gewürzen, Säften etc.) evaporieren. Hier dienten 24,9%iger Essigextrakt (aqua-diluierte reine Essigsäure) und Sojasauce als Datenquellen.

2. Integration von 5%, 2,5% und 1% Ethanolösungen als regular-Einträge zur Verbesserung der Modellkapazität in der Differenzierung identischer Substanzen mit signifikanten Konzentrationsdivergenzen.
3. Inkorporation von Daten normaler Gase aus Roni Bandinis bhopal84-Projekt zur Amplifikation der Datendiversität und -repräsentativität.

Post-Merkmalsextraktion aus den Rohdaten manifestiert sich ein hoher Diskriminierungsgrad zwischen den beiden Merkmalstypen (Abbildung 6-7), was eine präzise Differenzierung diverser Gastypen durch das Modell begünstigt.

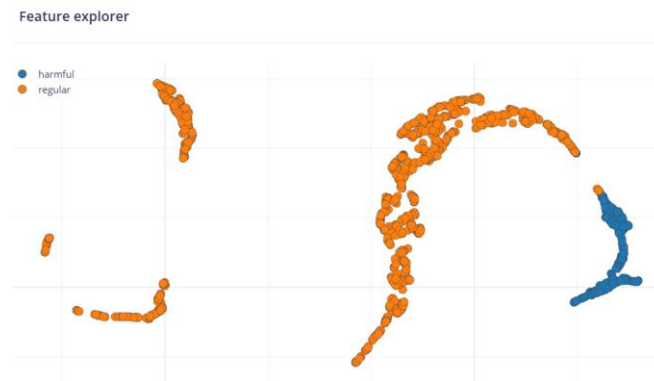


Abbildung 6-7 Aus Rohdaten extrahierte Merkmale

Der Optimierungsprozess des Modelltrainings involvierte multiple Experimente, die in der Determination folgender Hyperparameter kulminierten:

- Trainingsepochen: 60. Dieser Wert demonstrierte in den Experimenten eine optimale Balance zwischen suffizientem Lernen der Datencharakteristika und Prävention von Überanpassung.
- Lernrate: 0,0005. Dieser konservative Wert fördert die stabile Konvergenz des Modells und minimiert exzessive Fluktuationen während des Trainingsprozesses.
- Validierungsset-Größe: 20%. Diese Proportion offeriert adäquate Daten zur Evaluierung der Modellperformanz bei simultaner Reservation der Majorität der Daten für das Training.
- Automatische Klassengewichtung (Auto-weight classes): Aktiviert. Dies ist essenziell für die Handhabung potenzieller Klassenimbancen und gewährleistet eine äquivalente Erkennungskapazität für beide Kategorien.
- Netzwerkstruktur: Implementierung einer zusätzlichen Dropout-Schicht mit einer DropoutRate von 0,1. Dieses Design zielt auf die Augmentation der Generalisierungsfähigkeit des Modells und die Prävention von Überanpassung ab, insbesondere angesichts potenzieller Limitationen in den Trainingsdaten.

Die Selektion dieser Parameter reflektiert eine sorgfältig kalibrierte Balance zwischen Modellperformanz und computationalen Ressourcen. Durch iterative Feinjustierung wird das ultimative Ziel verfolgt, ein Modell zu generieren, das sowohl auf ressourcenlimitierten eingebetteten Systemen operabel ist als auch eine präzise Gasklassifizierung liefert. In subsequenten Testphasen wird die Effizienz dieser Konfigurationen weiter evaluiert und bei Bedarf basierend auf der empirischen Performanz readjustiert.

6.2.2.3 Modellevaluierung und theoretische Leistungsanalyse

Nach Abschluss des Trainings führte die Edge Impulse Plattform eine umfassende Evaluierung des Modells durch. Die Evaluierungsergebnisse demonstrieren eine bemerkenswerte Leistung des Modells und offenbaren ein signifikantes Potenzial für praktische Anwendungen.

Die in Abbildung 6-8 dargestellten erwarteten Leistungsdaten des trainierten Modells sind äußerst vielversprechend. Auf dem Validierungsset erreichte das Modell eine beeindruckende Genauigkeit von 98,7% bei einem minimalen Verlust von lediglich 0,01. Diese Resultate indizieren eine exzellente Generalisierungsfähigkeit des Modells, das effizient zwischen schädlichen und normalen Gasen zu differenzieren vermag. Die Konfusionsmatrix untermauert diese Erkenntnis und zeigt eine perfekte Erkennungsrate von 100% für schädliche Gase, während die Präzision bei der Identifikation normaler Gase bei beachtlichen 98,5% liegt. Diese ausgewogene Distribution hoher Leistung reflektiert die außerordentliche Effizienz des Modells im Kontext der Gaserkennung.

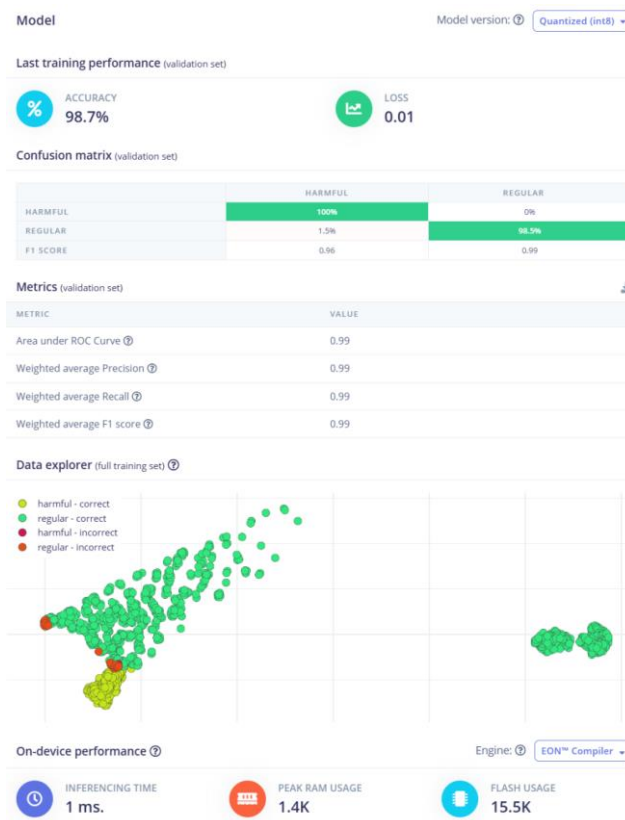


Abbildung 6-8 Erwartete Leistungsdaten des Modells

Die herausragende Performance des Modells manifestiert sich auch in weiteren Schlüsselindikatoren: Die Area Under the ROC Curve (AUC) erreicht einen nahezu perfekten Wert von 0,99, was auf eine exzeptionelle Diskriminierungsfähigkeit des Modells hindeutet. Die gewichteten Durchschnittswerte für Präzision, Recall und F1-Score konvergieren ebenfalls bei 0,99, was die balancierte und exzellente Leistung des Modells über verschiedene Kategorien hinweg weiter bekräftigt. Diese Indikatoren demonstrieren kumulativ die robuste Leistungsfähigkeit des Modells bei der Distinktion zwischen schädlichen und harmlosen Gasen.

Die Datenexplorationsanalyse offenbart die Fähigkeit des Modells, Proben verschiedener Kategorien im Merkmalsraum effektiv zu separieren, wobei Fehlklassifikationen eine Rarität darstellen.

Die Proben exhibieren einen deutlichen Clustering-Trend im Merkmalsraum, was den Lern- und Generalisierungsprozess des Modells signifikant begünstigt. Diese Observationen validieren nachdrücklich die Effektivität der implementierten Feature-Engineering- und Modelldesignstrategien.

Hinsichtlich der On-Device-Performance demonstriert das Modell eine bemerkenswerte Effizienz. Bei der Kompilierung mit der EON Compiler Engine wird eine einzelne Inferenz in nur 1 Millisekunde prognostiziert, bei einer minimalen Spitzen-RAM-Nutzung von 1,4 KB und einem Flash-Verbrauch von lediglich 15,5 KB. Diese Metriken attestieren dem Modell eine exzellente Eignung für den Einsatz in ressourcenlimitierten eingebetteten Systemen und erfüllen vollumfänglich die Anforderungen an Echtzeiterkennungssysteme.

Trotz der insgesamt exzellenten Performance des Modells könnte die relative Homogenität der Trainingsdaten bei der Konfrontation mit komplexeren Mehrkomponentengasen potenzielle Herausforderungen bergen. Zukünftige Optimierungsstrategien könnten folgende Aspekte adressieren: Augmentation der Diversität harmloser und schädlicher Gasproben im Trainingsdatensatz, Exploration diskriminativerer Features, Durchführung von Tests und Feinabstimmungen unter heterogeneren Umgebungsbedingungen zur Sicherstellung konsistenter Hochleistung in diversen praktischen Anwendungsszenarien, sowie die Evaluation adaptierter Klassifizierungsschwellen oder alternativer Modellarchitekturen. Diese Maßnahmen versprechen eine weitere Steigerung der Gesamtleistung des Modells, insbesondere im Hinblick auf die Minimierung der Falsch-Positiv-Rate.

Resümierend präsentiert dieses TinyML-Modell eine äußerst zufriedenstellende Leistung und Praktikabilität im Bereich der Gaserkennung und etabliert damit ein solides Fundament für vielfältige praktische Applikationen.

6.2.2.4 TinyML-Implementierung

Diese Studie präsentiert die Integration von TinyML-Algorithmen als innovative Strategie zur Optimierung der Sensorleistung. Die Implementierung erfolgt durch eine einzelne, als autonome FreeRTOS-Aufgabe konzipierte Funktion. Dieser Ansatz gewährleistet nicht nur eine nahtlose Integration in die bestehende Systemarchitektur, sondern erhöht auch signifikant die Modularität und Wartbarkeit des Gesamtsystems.

Die Funktion `InferenceUsingTinyML()` bildet das Herzstück der TinyML-Implementierung. Ihr Design basiert auf einer Restrukturierung der Schleifenfunktion aus der Datei "bhopal4.ino" des bhopal84-Projekts. Diese Methodik ermöglicht eine synergetische Fusion bewährter Lösungsansätze mit spezifischen Optimierungen für die vorliegenden Anforderungen.

Das in Abbildung 6-9 illustrierte Flussdiagramm visualisiert den Arbeitsprozess der TinyML-Funktion als zyklisches, selbstneustartendes Verfahren:

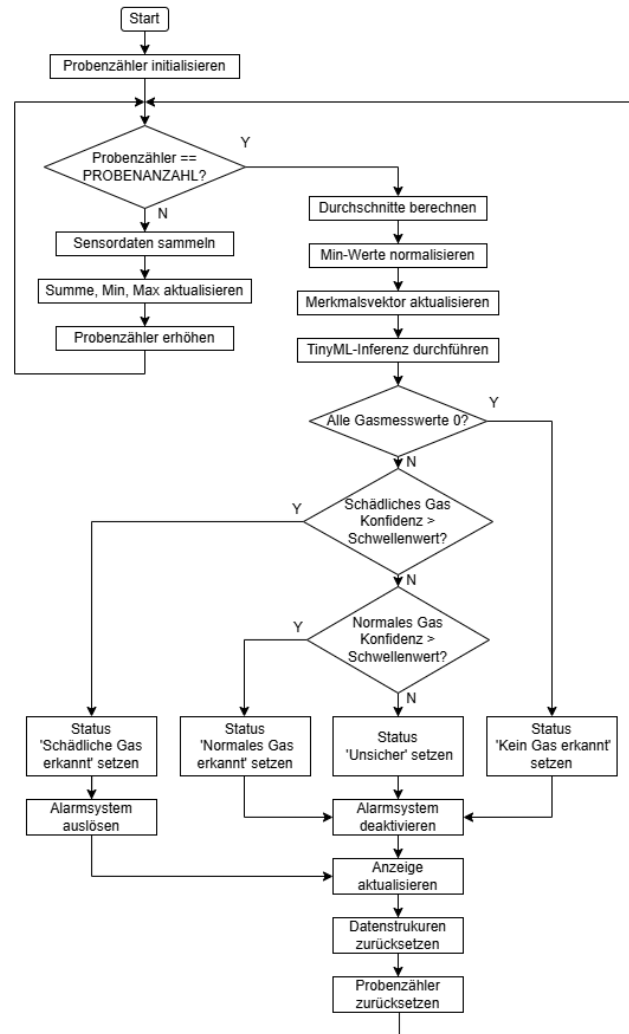


Abbildung 6-9 Detailliertes Flussdiagramm der TinyML-Funktion

Der Prozess initiiert mit der Initialisierung eines Probenzählers, der den Beginn jedes Datenerfassungszyklus markiert. Das System evaluiert kontinuierlich, ob der Probenzähler die vorgegebene Probenanzahl erreicht hat. Bis zur Erreichung dieses Schwellenwerts akquiriert das System fortlaufend Sensordaten, aktualisiert kumulierte Summen sowie Extremwerte und inkrementiert den Probenzähler. Diese iterative Schleife garantiert eine ausreichende Datenbasis für eine statistisch robuste Analyse.

Bei Erreichen der erforderlichen Probenzahl tritt das System in die Datenverarbeitungsphase ein. Initial erfolgt die Berechnung des arithmetischen Mittels, gefolgt von einer Normalisierung der Minimalwerte. Diese statistischen Kenngrößen fließen in die Aktualisierung des Merkmalsvektors ein, der als Grundlage für die nachfolgende TinyML-Inferenz dient. Die Konstruktion des Merkmalsvektors integriert multiple statistische Metriken und liefert dadurch eine umfassende Charakterisierung der Gaskonzentrationsfluktuationen.

Im Anschluss erfolgt die Durchführung der TinyML-Inferenz - das Kernstück des Gesamtprozesses. Hierbei wird das vortrainierte Modell zur Klassifizierung der aktuellen Umgebungsbedingungen herangezogen. Das resultierende Inferenzergebnis durchläuft eine kaskadierte Reihe von Bedingungsprüfungen: Initial wird evaluiert, ob sämtliche Gaskonzentrationen den Nullwert aufweisen; in diesem Fall wird der Status "Kein Gas detektiert" gesetzt. Andernfalls erfolgt eine

Prüfung, ob die Konfidenz für die Präsenz schädlicher Gase den definierten Schwellenwert überschreitet. Bei positiver Evaluation initiiert das System einen Alarmzustand; bei negativer Evaluation wird die Konfidenz für reguläre Gase analysiert. Diese hierarchische Entscheidungsbaumstruktur befähigt das System zu einer nuancierten Reaktion auf diverse Umgebungsszenarien. Unabhängig vom finalen Klassifikationsergebnis aktualisiert das System die Benutzeroberfläche, um dem Anwender ein präzises visuelles Feedback zu präsentieren. Dies gewährleistet eine kontinuierliche Information des Benutzers über den aktuellen Umgebungszustand und potenzielle Risikofaktoren.

Nach Abschluss eines vollständigen Detektions- und Analysezyklus initiiert das System eine Reinitialisierung der Datenstrukturen und des Probenzählers, um den nächsten Datenerfassungszyklus vorzubereiten. Diese zyklische Architektur ermöglicht eine kontinuierliche Umgebungsüberwachung mit der Fähigkeit zur zeitnahen Detektion von Gaskonzentrationsfluktuationen.

6.2.2.5 Ergänzende Systemmodifikationen

Zur Realisierung einer flexiblen Umschaltung zwischen TinyML-basierter und konventioneller Detektionsmethodik wurde in den Systemeinstellungen eine innovative Schalteroption implementiert. Diese Designentscheidung eröffnet dem Benutzer die Möglichkeit, situativ die optimale Detektionsmethode zu selektieren, wodurch eine Optimierung der Systemperformanz und eine Adaption an variierende Umgebungsbedingungen ermöglicht wird.

Zur Gewährleistung eines rapiden und stabilen Wechsels zwischen den Detektionsmodi wurden signifikante Modifikationen am Task-Scheduler vorgenommen. Der erweiterte Scheduler übernimmt nun nicht nur die Kontrolle über Start und Terminierung von Alarmaufgaben, sondern führt auch ein kontinuierliches Monitoring des TinyML-Schalterstatus durch und adjustiert die aktiven Tasks dynamisch nach Bedarf. Bei einer Transition in den TinyML-Modus initiiert der Scheduler primär die Terminierung konventioneller Detektionsaufgaben, gibt erforderlichenfalls assoziierte Mutexes frei und startet subsequently die TinyML-Task. Analog wird bei einer Rückkehr zum konventionellen Detektionsmodus verfahren. Dieser sophisticatede Umschaltmechanismus garantiert eine effiziente Ressourcenallokation bei simultaner Minimierung von Systeminterruptionen während der Modustransitionen.

Im Bereich der Benutzeroberfläche wurden ebenfalls substantielle Modifikationen zur Integration der TinyML-Funktionalität implementiert. Der Hauptbildschirm wurde um ein dediziertes Set von UI-Elementen erweitert, die spezifisch für die Visualisierung von TinyML-Detektionsinformationen konzipiert wurden. Diese UI-Komponenten präsentieren die TinyML-Analyseergebnisse in intuitiver Form, wodurch Benutzer eine rasche Erfassung des aktuellen Umgebungszustands ermöglicht wird. Komplementär wurde in der oberen Statusleiste ein Indikator-Icon zur Anzeige des TinyML-Schalterstatus integriert.

Diese umfassenden Anpassungen konstituieren einen holistischen Systemupgrade-Plan, der Optimierungen auf multiplen Ebenen wie Taskmanagement, User Interface und Systemkonfiguration umfasst. Sie ermöglichen eine erfolgreiche Integration der TinyML-Funktionalität in die existierende Systemarchitektur bei simultaner Preservation der Gesamtsystemflexibilität.

6.2.2.6 Vorläufige Tests

Zur Validierung der Effektivität und Leistungsfähigkeit des neu integrierten TinyML-Modus wurde eine Reihe präliminärer Tests durchgeführt. Diese evaluierten die Systemperformanz in realistischen Anwendungsszenarien, mit besonderem Fokus auf die Fähigkeit zur Detektion potenzieller Gefahrgase und zur Reduktion von Fehlalarmen.

Initial wurde die Kapazität des Systems zur Erkennung hochkonzentrierter Gefahrgase evaluiert. Als Testsubstanz diente Ethanol in Konzentrationen von 99,9% und 20%. Die Selektion dieser spezifischen Konzentrationsniveaus erfolgte strategisch: 99,9% repräsentiert ein Extremszenario, während 20% ein frequenteres, jedoch immer noch potenziell gefährliches Expositionsniveau simuliert. Die Testresultate demonstrierten, dass der TinyML-Modus beide Ethanolkonzentrationen mit außerordentlich hoher Präzision korrekt als Gefahrgase klassifizierte. Diese Performanz korreliert stark mit den post-Training antizipierten Leistungsparametern und verifiziert die exzellente Effizienz der TinyML-Implementierung bei der Identifikation eindeutig gefährlicher Szenarien.

Die eigentliche Herausforderung manifestierte sich jedoch in der Systemreaktion auf Situationen mit erhöhtem Fehlalarmpotenzial. Konventionelle Detektionsmethoden tendieren dazu, bestimmte hochkonzentrierte, jedoch harmlose Gase (wie sie von diversen Saucen oder Gewürzen emittiert werden) fälschlicherweise als gefährliche Substanzen zu kategorisieren. Zur Evaluation der Verbesserungen durch den TinyML-Ansatz in diesem Kontext wurde spezifisch die Systemreaktion auf Gase getestet, die von Essigessenz generiert werden. Die Testresultate indizierten, dass der TinyML-Modus in der Majorität der Fälle erfolgreich diese Gase korrekt als normale Gase klassifizierte, anstatt sie irrtümlich als gefährliche Substanzen einzustufen. Diese Optimierung ist von signifikanter Relevanz für die praktische Systemapplikation. Sie steigert nicht nur die Detektionsgenauigkeit, sondern minimiert auch substantziell die Inzidenz falsch positiver Resultate, die potenziell zu Benutzerermüdung oder Warnsignalignoranz führen könnten.

Diese präliminären Testergebnisse offenbaren mehrere Schlüsselvorteile des TinyML-Ansatzes:

- Hochpräzise Detektion von Hochrisikogasen: Bei der Identifikation eindeutiger Gefahrgase (exemplifiziert durch hochkonzentriertes Ethanol) demonstrierte das System eine nahezu perfekte Akkuratessse. Dies garantiert eine zuverlässige Alarmierung in tatsächlich gefährlichen Situationen.
- Signifikante Reduktion der Fehlalarmrate: Durch die akkurate Identifikation frequenter harmloser Gase minimiert der TinyML-Ansatz die Inzidenz von Fehlalarmen substantiell.
- Optimierte Umgebungsadaptabilität: Im Vergleich zu konventionellen Methoden exhibiert der TinyML-Modus eine superiore Performanz bei der Prozessierung komplexer, ambiguer Gaskombinationen. Diese Fähigkeit ist essentiell für den Einsatz in diversen realen Umgebungen.

Obgleich diese initialen Resultate höchst satisfaktionierend sind, existiert noch signifikantes Optimierungspotenzial für die TinyML-Detektionsmethodik. Exemplarisch könnte das System, bedingt durch die Limitationen des Trainingsdatensatzes, in spezifischen Grenzsituationen noch Fehlklassifikationen generieren. Konsequenterweise sind extensivere und diversifiziertere Trainings- und Testphasen imperativ.

In der Gesamtbetrachtung untermauern diese präliminären Testergebnisse nachdrücklich die strategische Entscheidung zur Implementation der TinyML-Methode. Sie optimiert nicht nur die Systemgenauigkeit bei der Detektion von Gefahrstoffen, sondern eleviert auch signifikant die Systemperformanz bei der Prozessierung komplexer, potenziell missverständlicher Umgebungen. Dies konstituiert das Fundament für einen zuverlässigeren und intelligenteren Schutz in praktischen Anwendungsszenarien dieses Systems.

7 Versuchsdurchführung

In diesem Kapitel wird die Leistungsfähigkeit und Zuverlässigkeit des auf MEMS-MOS-Gassensoren basierenden Gefahrstoffdetektionssystems durch ein systematisch konzipiertes und durchgeführtes Versuchsdesign validiert. Trotz der Limitationen hinsichtlich der Versuchsbedingungen und der verfügbaren Ausrüstung wurde ein reproduzierbares und verifizierbares Versuchsprotokoll entwickelt, um die Systemperformanz unter realistischen Anwendungsbedingungen zu evaluieren. Das Kapitel präsentiert eine detaillierte Beschreibung des Versuchsplans, des Versuchsablaufs sowie der Ergebnisanalyse, um eine umfassende Bewertung der Funktionalität und Leistungsfähigkeit des Systems zu ermöglichen.

7.1 Versuchsaufbau

7.1.1 Zielsetzung der Versuche

Die durchgeführten Versuche zielen darauf ab, die Leistungsfähigkeit und Zuverlässigkeit des auf MEMS-MOS-Gassensoren basierenden Gefahrstoffdetektionssystems zu validieren. Dabei liegt der Fokus auf folgenden Aspekten:

- Evaluierung der Systemkapazität zur Detektion verschiedener Konzentrationen von Gefahrstoffen.
- Analyse der Konsistenz und Reproduzierbarkeit der Systemresponse.
- Komparative Leistungsbewertung zwischen konventionellen Detektionsmethoden (Schwellenwertvergleich) und TinyML-basierten Detektionsansätzen.
- Quantifizierung des Einflusses von Umweltparametern auf die Detektionsresultate.

7.1.2 Versuchshypothesen

Basierend auf den Erkenntnissen aus vorangegangenen Forschungsarbeiten und dem implementierten Systemdesign werden folgende Versuchshypothesen formuliert:

1. Das System ist in der Lage, innerhalb eines definierten Spektrums die Präsenz von Gefahrstoffen zu detektieren und weist differenzierte Reaktionen auf variierende Konzentrationen dieser Substanzen auf.
2. Unter konstanten Versuchsbedingungen demonstrieren die Detektionsresultate des Systems eine hohe Konsistenz und Reproduzierbarkeit.
3. Im Vergleich zu traditionellen Schwellenwertvergleichsmethoden könnte der TinyML-basierte Detektionsansatz in komplexen Umgebungsszenarien eine superior Detektionsgenauigkeit aufweisen.
4. Umweltfaktoren wie Temperatur und Luftfeuchtigkeit könnten einen signifikanten Einfluss auf die Präzision der Detektionsergebnisse ausüben.

7.1.3 Versuchsaufbau

Unter Berücksichtigung der vorhandenen Versuchsbedingungen wurde ein auf relativen Vergleichen basierender Versuchsplan konzipiert, um die Systemleistung umfassend zu evaluieren:

1. Evaluierung der Detektionsempfindlichkeit und Diskriminierungsfähigkeit:

- Einsatz von Ethanolösungen variierender Konzentrationen (99,9%, 20%, 1%) sowie 24,9%iger Essigsäure als Testsubstanzen. Destilliertes Wasser fungiert als Kontrollprobe.
- Positionierung eines Behälters mit 10g Testlösung in definiertem Abstand (Behälterrand 3cm vor dem Sensor, Flüssigkeitsoberfläche 5cm unterhalb des Sensors).
- Installation eines 5010 Axialventilators 0,8cm vor dem Sensor, betrieben mit 5V, zur Simulation realistischer Luftströmungsverhältnisse.
- Protokollierung der Systemreaktion (Alarmauslösung ja/nein).

2. Reproduzierbarkeitsanalyse:

- Mehrfache Wiederholung der Tests für jede Konzentrationsstufe (mindestens 5 Iterationen).
- Detaillierte Aufzeichnung der Testergebnisse zur Bewertung der Konsistenz der Systemresponse.

3. Komparative Methodenanalyse:

- Parallele Durchführung der Detektion mittels konventioneller Schwellenwertvergleichsmethode und TinyML-Ansatz.
- Vergleichende Leistungsbewertung beider Methoden unter diversen Testszenarien.

4. Umweltfaktoreinflussanalyse:

- Wiederholung der vorgenannten Testserien unter variierenden Umgebungstemperaturen und Luftfeuchtigkeitsbedingungen.
- Erfassung der Umweltparameter und korrespondierenden Systemreaktionen zur Quantifizierung des Umwelteinflusses.

Antizipierte Resultate:

- Das System sollte innerhalb des definierten Versuchsspektrums die Präsenz von Gefahrstoffen zuverlässig detektieren und differenziert auf verschiedene Probenkonzentrationen reagieren. Für den TinyML-Ansatz wird erwartet, dass 99,9%iges Ethanol als Gefahrstoff klassifiziert wird und eine Alarmauslösung erfolgt, während 1%iges Ethanol und 24,9%ige Essigsäure als unbedenkliche Substanzen eingestuft werden. 20%iges Ethanol dient als Interferenztest.
- Unter konstanten Versuchsbedingungen sollten die Detektionsresultate eine hohe Konsistenz und Reproduzierbarkeit aufweisen.
- Es wird antizipiert, dass die TinyML-Detektionsmethode im Vergleich zum konventionellen Schwellenwertvergleich eine überlegene Detektionsgenauigkeit und Diskriminierungsfähigkeit demonstriert.

- Umweltfaktoren wie Temperatur und Luftfeuchtigkeit sollten zwar einen messbaren Einfluss auf die Detektionsergebnisse ausüben, ohne jedoch die Gesamtpräzision der Klassifikation signifikant zu kompromittieren.

7.1.4 Versuchsmaterialien und -ausrüstung

Die für die Durchführung des Experiments erforderlichen Materialien und Ausrüstungen sind in Tabelle 7-1 detailliert aufgeführt.

Bezeichnung	Anzahl
Gefahrstoffdetektionssystem auf Basis von MEMS-MOS-Gassensoren	1
5010 Axiallüfter (Dimensionen: 50*50*10mm, Nennspannung: 12V)	1
Ethanol Lösungen in Konzentrationen von 99,9%, 20% und 1%	jeweils 50g pro Konzentration
Essigessenz (24,9%ige Essigsäure)	50g
Destilliertes Wasser	50g
Präzisionswaage (Genauigkeit: 0,1g)	1
Saubere Petrischale	1
Temperatur- und Feuchtigkeitsmessgerät	1
Stoppuhr	1
Protokollierungstabelle	1

Tabelle 7-1 Auflistung der für das Experiment benötigten Materialien und Ausrüstungen

7.2 Versuchsdurchführung

1. Präparative Maßnahmen:

- Sorgfältige Kalibrierung sämtlicher Messapparaturen.
- Herstellung und hermetische Versiegelung von Ethanol Lösungen diverser Konzentrationen, konzentrierter Essigsäure sowie destilliertem Wasser zur Prävention von Verdunstungseffekten.
- Akribische Dokumentation der initialen Temperatur- und Feuchtigkeitswerte im Versuchsareal.

2. Einzelversuchsprotokoll:

- Präzise Einwägung von 10g Testlösung mittels geeichter elektronischer Analysenwaage.
- Initiierung des Systemstarts und Abwarten der definierten Aufwärmphase.
- Positionierung der Testlösung in einem sterilisierten schalenförmigen Rezeptakel vor dem Sensor, unter strikter Einhaltung der vorgegebenen Geometrie (Behälterrand 3cm vor dem Sensor, Flüssigkeitsniveau 5cm unterhalb des Sensors).

- Erfassung der Alarmauslösung und der korrespondierenden Reaktionslatenz. Anschließend 10-sekündige Observationsphase zur Verifizierung der Alarmstabilität; bei Konstanz des Alarmstatus wird dieser als stabil klassifiziert.
 - Post-experimentelle Reinigung des Rezeptakels und suffiziente Ventilation des Versuchsareals.
3. Reproduzierbarkeitsanalyse:
- Quintupel Iteration der Einzelversuchsprozedur (Schritt 2) für jede Testlösung.
 - Gewährleistung der Systemrückkehr in den definierten Initialzustand zwischen den Versuchsdurchläufen.
 - Komparative Methodenevaluation:
 - Parallele Durchführung der Detektion mittels konventioneller Schwellenwertkomparation und TinyML-Methodik.
 - Detaillierte Protokollierung der Resultate jedes Versuchsdurchlaufs für beide Methoden.
4. Umweltfaktorenanalyse:
- Kontrollierte Modifikation der Umgebungstemperatur und -luftfeuchtigkeit mit anschließender Replikation der vorgenannten Versuchsreihen.
 - Präkonditionierung der Apparatur vor jedem Versuchsdurchlauf unter standardisierten Bedingungen, die den Innenraumklimata einer Feuerwehrrstation oder eines Einsatzfahrzeugs entsprechen, zur Sicherstellung konsistenter Ausgangsbedingungen.
 - Minutiöse Erfassung der Umweltparameter und korrespondierenden Versuchsergebnisse für jeden Durchgang.
5. Datenkonsolidierung und -analyse:
- Systematische Aggregation und komparative Evaluation der Leistungsprofile beider Detektionsmethoden.
 - Komprehensiv Zusammenstellung und Analyse der Versuchsergebnisse unter variierenden Temperatur- und Feuchtigkeitsregimen zur Beurteilung der Systemstabilität und -reliabilität in diversen Umgebungsconfigurationen.

Sämtliche im Rahmen dieser Studie erhobenen Versuchsdaten sind in der Tabelle " Versuchsdatenprotokoll.xlsx" im Anhang zusammengefasst. Diese umfassende Tabelle beinhaltet die Rohdaten aller durchgeführten Tests und umfasst unter anderem folgende Informationen:

- Präzise Zeitstempel (Datum und Uhrzeit) für jeden Testdurchlauf
- Umgebungsparameter: Temperatur und relative Luftfeuchtigkeit während der Tests
- Spezifikationen der Testlösungen: Typ und exakte Konzentration
- Vergleichende Detektionsergebnisse: Schwellenwertmethode vs. TinyML-Ansatz
- Systemperformanz: Reaktionszeiten und Ansprechverhalten
- Detaillierte Aufzeichnungen zur Stabilität und Zuverlässigkeit des Alarmsystems
- Reproduzierbarkeitsanalyse: Ergebnisse wiederholter Testdurchläufe

- Robustheitsprüfung: Testergebnisse unter variierenden Umgebungsbedingungen

Die Daten wurden einer sorgfältigen systematischen Aufbereitung unterzogen, um eine effiziente und aussagekräftige nachfolgende Datenanalyse sowie eine fundierte Diskussion der Ergebnisse zu ermöglichen. Diese strukturierte Datensammlung bildet die Grundlage für eine umfassende Evaluation der Leistungsfähigkeit und Zuverlässigkeit des entwickelten Detektionssystems unter verschiedenen Betriebsbedingungen.

7.3 Analyse und Bewertung der Versuchsergebnisse

In diesem Abschnitt werden die Versuchsergebnisse eingehend analysiert und die Systemleistung im Hinblick auf die Erwartungen bewertet. Die Analyse umfasst vier Aspekte: Detektionsfähigkeit, Konsistenz und Reproduzierbarkeit, Methodenvergleich sowie Umwelteinflüsse.

7.3.1 Analyse der Detektionsfähigkeit

Die Versuchsergebnisse zeigen, dass das auf MEMS-MOS-Gassensoren basierende Gefahrstoffdetektionssystem unterschiedliche Detektionsfähigkeiten bei verschiedenen Konzentrationen von Ethanol- und Essigsäurelösungen aufweist.

Für die traditionelle Schwellenwertmethode:

- 99,9% und 20% Ethanol Lösungen wurden zu 100% korrekt detektiert, mit durchschnittlichen Reaktionszeiten von 1,73 bzw. 2,77 Sekunden.
- 1% Ethanol Lösung konnte nicht korrekt detektiert werden, mit einer Erfolgsrate von 0%.
- 24,9% Essigsäurelösung hatte eine niedrige Detektionsrate von nur 20%, mit einer durchschnittlichen Reaktionszeit von 6,67 Sekunden.

Für die TinyML-Methode:

- 99,9% und 20% Ethanol Lösungen wurden ebenfalls zu 100% korrekt detektiert und klassifiziert, mit durchschnittlichen Reaktionszeiten von 2,07 bzw. 2,37 Sekunden.
- Die Detektions- und Klassifizierungsrate für 1% Ethanol Lösung stieg auf 32%.
- Die Detektions- und Klassifizierungsrate für 24,9% Essigsäurelösung verbesserte sich deutlich auf 80%, mit einer verkürzten durchschnittlichen Reaktionszeit von 2,69 Sekunden. .

Diese Ergebnisse entsprechen im Wesentlichen unseren Versuchshypothesen, dass das System in der Lage ist, Gefahrstoffe innerhalb eines bestimmten Bereichs zu detektieren und unterschiedlich auf verschiedene Konzentrationen zu reagieren. Allerdings besteht bei der Detektion von niedrig konzentriertem (1%) Ethanol noch Verbesserungsbedarf.

7.3.2 Analyse der Konsistenz und Reproduzierbarkeit

Durch mehrfache Wiederholung der Tests für jede Probenkonzentration konnten wir die Konsistenz und Reproduzierbarkeit der Systemreaktion bewerten. Die Versuchsdaten zeigen:

- Für hochkonzentrierte (99,9% und 20%) Ethanol Lösungen zeigten sowohl die traditionelle Schwellenwertmethode als auch die TinyML-Methode exzellente Konsistenz und Reproduzierbarkeit mit einer Detektionsrate von 100%.
- Für niedrig konzentrierte (1%) Ethanol Lösung waren die Detektionsergebnisse beider Methoden instabil, wobei die TinyML-Methode etwas besser abschnitt.
- Für 24,9% Essigsäurelösung zeigte die TinyML-Methode deutlich bessere Konsistenz und Reproduzierbarkeit als die traditionelle Schwellenwertmethode.

Diese Ergebnisse unterstützen teilweise Versuchshypothese, dass die Detektionsergebnisse des Systems unter gleichen Bedingungen eine gute Konsistenz und Reproduzierbarkeit aufweisen. Allerdings besteht bei Proben mit niedriger Konzentration und Nicht-Ziel-Gefahrstoffen (wie Essigsäure) noch Verbesserungsbedarf hinsichtlich Konsistenz und Reproduzierbarkeit.

7.3.3 Vergleichende Methodenanalyse

Beim Vergleich der Leistung der traditionellen Schwellenwertmethode mit der TinyML-Methode lassen sich folgende Schlussfolgerungen ziehen:

- Für hochkonzentriertes Ethanol (99,9% und 20%) zeigten beide Methoden exzellente Detektionsfähigkeiten mit einer Erfolgsrate von 100%.
- Für niedrig konzentriertes Ethanol (1%) zeigte die TinyML-Methode einen deutlichen Vorteil mit einer Detektionsrate von 32% gegenüber 0% bei der traditionellen Schwellenwertmethode.
- Für 24,9% Essigsäurelösung war die Detektionsrate der TinyML-Methode (80%) signifikant höher als die der traditionellen Schwellenwertmethode (20%).
- Die TinyML-Methode zeigte eine bessere Unterscheidungsfähigkeit bei der Behandlung von Nicht-Ziel-Gefahrstoffen (wie Essigsäure).
- Die durchschnittlichen Reaktionszeiten der TinyML-Methode waren relativ stabil, während die traditionelle Schwellenwertmethode bei der Verarbeitung von Proben mit niedriger Konzentration und Nicht-Zielsubstanzen deutlich längere Reaktionszeiten aufwies.

Diese Ergebnisse unterstützen die vorherige Annahme, dass die TinyML-Detektionsmethode im Vergleich zur traditionellen Schwellenwertvergleichsmethode in komplexen Umgebungen eine bessere Detektionsgenauigkeit aufweist.

7.3.4 Analyse des Einflusses von Umweltfaktoren

Temperatureinfluss:

- Bei der traditionellen Schwellenwertmethode waren die Reaktionszeiten unter Hochtemperaturbedingungen generell kürzer als unter Niedrigtemperaturbedingungen, aber der Einfluss auf die Detektionsrate war gering.
- Die Leistung der TinyML-Methode blieb unter verschiedenen Temperaturbedingungen relativ stabil, mit geringen Veränderungen in der Detektionsrate und Reaktionszeit.

Feuchtigkeitseinfluss:

- Die traditionelle Schwellenwertmethode zeigte eine verringerte Detektionsrate für 24,9% Essigsäure unter hoher Luftfeuchtigkeit (von 40% bei niedriger Luftfeuchtigkeit auf 10% bei hoher Luftfeuchtigkeit).
- Die TinyML-Methode zeigte eine relativ stabile Leistung unter verschiedenen Feuchtigkeitsbedingungen, jedoch mit einer leichten Abnahme der Detektionsrate für 24,9% Essigsäure unter hoher Luftfeuchtigkeit (von 100% bei niedriger Luftfeuchtigkeit auf 70% bei hoher Luftfeuchtigkeit).

Diese Ergebnisse unterstützen teilweise die vorherige Annahme, dass Umweltfaktoren (wie Temperatur und Luftfeuchtigkeit) einen gewissen Einfluss auf die Detektionsergebnisse haben, ohne die Gesamtgenauigkeit der Beurteilung ernsthaft zu beeinträchtigen. Bemerkenswert ist, dass die TinyML-Methode unter verschiedenen Umweltbedingungen eine bessere Stabilität zeigte.

Insgesamt entsprechen die Versuchsergebnisse weitgehend den Erwartungen und Hypothesen. Das auf MEMS-MOS-Gassensoren basierende Gefahrstoffdetektionssystem zeigt gute Detektionsfähigkeiten, insbesondere bei Verwendung der TinyML-Methode. Allerdings besteht noch Verbesserungspotenzial bei der Detektion von Proben mit niedriger Konzentration und der Leistung unter extremen Umweltbedingungen. Zukünftige Forschungen könnten sich auf die weitere Optimierung der Algorithmen konzentrieren, um die Detektionsgenauigkeit und Stabilität des Systems unter verschiedenen komplexen Bedingungen zu verbessern.

8 Zusammenfassung

8.1 Überblick über die Forschungsergebnisse

In dieser Studie wurde erfolgreich ein Gefahrstoffdetektionssystem auf Basis von MEMS-MOS-Gassensoren entwickelt, das speziell auf die Bedürfnisse von Hochrisikoberufen wie Feuerwehrleuten zugeschnitten ist. Das System zeichnet sich durch folgende Hauptmerkmale aus:

- Echtzeit-Mehrgas-Detektion zur simultanen Überwachung mehrerer häufig vorkommender Gefahrstoffe.
- Miniaturisiertes Design für einfache Integration in bestehende PSA.
- Modulares Design mit austauschbaren Komponenten für erhöhte Flexibilität.
- Drahtlose Kommunikationsfunktion für Echtzeit-Datenübertragung und Fernüberwachung.
- Benutzerfreundliche Schnittstelle mit einem Multimode-Alarmsystem für intuitive Bedienung.

Die experimentellen Ergebnisse belegen die hervorragende Leistung des Systems bei der Erkennung hochkonzentrierter Gefahrstoffe. Bei der Detektion von Proben mit niedriger Konzentration besteht noch Optimierungspotenzial. Bemerkenswert ist die gute Stabilität des Systems unter verschiedenen Temperatur- und Feuchtigkeitsbedingungen, insbesondere bei Anwendung der TinyML-Methode.

8.2 Zusammenfassung der Innovationspunkte

Die Studie zeichnet sich durch folgende Hauptinnovationen aus:

- Synergetische Kombination von MEMS-MOS-Sensortechnologie mit tragbarem Gerätedesign, die Miniaturisierung und hohe Empfindlichkeit vereint.
- Entwicklung eines auf TinyML basierenden intelligenten Detektionsalgorithmus, der gegenüber der traditionellen Schwellenwertmethode eine signifikant höhere Genauigkeit und Stabilität aufweist.
- Konzeption eines modularen Sensorsystems zur Steigerung der Flexibilität und Anpassungsfähigkeit.
- Implementation einer Funktion zum Fernumschalten von Szenario-Voreinstellungen, die die Adaptivität des Systems an verschiedene Brandsituationen erhöht.
- Einsatz eines integrierten und eigenständigen Doppelmodusdesigns zur Maximierung der Nutzungsflexibilität.
- Integration von Cloud-Datenanalyse und Echtzeit-Feedback-Funktionen, die eine kontinuierliche Optimierung der Detektionsstrategien ermöglichen.

8.3 Analyse der Anwendungsperspektiven

- Das entwickelte Gefahrstoffdetektionssystem eröffnet vielfältige Anwendungsmöglichkeiten:
- Feuerwehrwesen: Signifikante Verbesserung der Sicherheit von Einsatzkräften an Brandorten.
- Chemieindustrie: Einsatz zur Fabrikumgebungsüberwachung und zum persönlichen Schutz der Arbeiter.
- Bergbau: Anwendung zur Gasetektion in Bergwerken zur Erhöhung der Bergarbeitersicherheit.
- Notfallrettung: Umweltüberwachung an diversen Katastrophenorten.
- Umweltmonitoring: Aufbau von urbanen Luftqualitätsüberwachungsnetzwerken.

Das modulare Design und die hohe Anpassungsfähigkeit des Systems ermöglichen es, branchenspezifische Anforderungen zu erfüllen und leisten damit einen substanziellen Beitrag zur Verbesserung der Arbeitssicherheit in verschiedenen Sektoren.

9 Ausblick

9.1 Vorschläge zur Systemverbesserung

Trotz der vielversprechenden Ergebnisse dieser Studie gibt es noch erhebliches Potenzial für Weiterentwicklungen:

- Erweiterung des Sensorspektrums: Integration einer breiteren Palette von Sensoren für umfassendere Systemtests und verbesserte Detektionsfähigkeiten.
- Steigerung der Sensitivität: Fokussierung auf die Erhöhung der Empfindlichkeit und Genauigkeit bei der Erkennung von Gefahrstoffen in niedrigen Konzentrationen.
- Algorithmus-Optimierung: Weiterentwicklung des TinyML-Algorithmus zur Verbesserung der Systemleistung unter komplexen Umgebungsbedingungen.
- Robustheit gegen Störeinflüsse: Verstärkung der Widerstandsfähigkeit gegenüber Umweltfaktoren zur Minimierung von Fehldetektionen.
- Energieeffizienz: Verbesserung des Energiemanagements zur signifikanten Verlängerung der Betriebszeit des Geräts.
- Erweiterung der Anwendungsszenarien: Entwicklung zusätzlicher Szenario-Voreinstellungen zur Abdeckung eines breiteren Einsatzspektrums.

9.2 Zukünftige Forschungsrichtungen

Aufbauend auf den Erkenntnissen dieser Studie ergeben sich vielversprechende Forschungsperspektiven:

- Multisensor-Fusionstechnologie: Synergetische Integration verschiedener Sensortypen (z.B. optische Sensoren) zur Steigerung der Detektionsumfassendheit und -präzision.
- KI und Big-Data-Analyse: Vertiefung der Anwendung fortschrittlicher KI-Algorithmen in der Gefahrstofferkennung und Nutzung von Big-Data-Ansätzen zur kontinuierlichen Optimierung von Detektionsstrategien.
- Biosensortechnologie: Erforschung der Integration von Biosensoren zur Erweiterung des Detektionsspektrums auf biologische Gefahrenstoffe.
- Innovationen in der Wearable-Technologie: Entwicklung ergonomischerer und leichter integrierbarer Lösungen zur nahtlosen Einbindung des Systems in tragbare Schutzausrüstung.
- Interdisziplinäre Kooperationen: Intensivierung der Zusammenarbeit mit Experten aus Materialwissenschaften, Umweltwissenschaften und verwandten Disziplinen zur ganzheitlichen Weiterentwicklung der Detektionstechnologie.

Durch die konsequente Verfolgung technologischer Innovationen und die Förderung interdisziplinärer Zusammenarbeit besitzt das Gefahrstoffdetektionssystem das Potenzial, einen signifikanten Beitrag zur Verbesserung der Arbeitssicherheit und des Umweltschutzes zu leisten. Es kann

somit maßgeblich zur Schaffung sichererer und gesünderer Arbeitsumgebungen beitragen und neue Standards in der präventiven Gefahrenerkennung setzen.

Literatur

- [1] Smith, C., Kelsey, K., & Christiani, D. (1993). Risk Assessment and Occupational Health: Overview and Recommendations. *NEW SOLUTIONS: A Journal of Environmental and Occupational Health Policy*, 3, 26 - 38. <https://doi.org/10.2190/NS3.2.e>.
- [2] Sadhra, Steven S, et al. (2022). Chemical hazards. *Oxford Handbook of Occupational Health 3e*, 3 edn, 45-116. doi: 10.1093/med/9780198849803.003.0002.
- [3] Anderson, N., Marcum, J., Bonauto, D., Siegel, M., & LaSee, C. (2023). The Relative Burden of Occupational Injuries and Illnesses in Firefighters: An Analysis of Washington Workers' Compensation Claims, 2006-2020. *International journal of environmental research and public health*, 20(22), 7077. <https://doi.org/10.3390/ijerph20227077>.
- [4] Campbell, R. (2023). Firefighter injuries on the fireground. *National Fire Protection Association*. <https://www.nfpa.org/education-and-research/research/nfpa-research/fire-statistical-reports/patterns-of-firefighter-fireground-injuries>.
- [5] Melius, J. (2001). Occupational health for firefighters. *Occupational medicine*, 16 1, 101-8
- [6] Melnikova, N., Wu, J., Yang, A., & Orr, M. (2017). Acute Chemical Incidents With Injured First Responders, 2002-2012. *Disaster Medicine and Public Health Preparedness*, 12, 211 - 221. <https://doi.org/10.1017/dmp.2017.50>.
- [7] Brandt-rauf, P., Fallon, L., Tarantini, T., Idema, C., & Andrews, L. (1988). Health hazards of fire fighters: exposure assessment.. *British Journal of Industrial Medicine*, 45, 606 - 612. <https://doi.org/10.1136/oem.45.9.606>.
- [8] Sawada, S., Kuklane, K., Wakatsuki, K., & Morikawa, H. (2017). New development of research on personal protective equipment (PPE) for occupational safety and health. *Industrial Health*, 55, 471 - 472. <https://doi.org/10.2486/indhealth.55-471>.
- [9] Han, D., Choi, S., & Lee, S. (2020). *Hazardous Gas Analysis during Fire Investigation*. , 34, 94-103. <https://doi.org/10.7731/KIFSE.26C6D4AB>.
- [10] Terrill, J. B., Montgomery, R. R., & Reinhardt, C. F. (1978). Toxic gases from fires. *Science* (New York, N.Y.), 200(4348), 1343–1347. <https://doi.org/10.1126/science.208143>.
- [11] Treitman, R. D., Burgess, W. A., & Gold, A. (1980). Air contaminants encountered by fire-fighters. *American Industrial Hygiene Association journal*, 41(11), 796–802. <https://doi.org/10.1080/15298668091425662>.
- [12] Fent, K. W., Mayer, A., Bertke, S., Kerber, S., Smith, D., & Horn, G. P. (2019). Understanding airborne contaminants produced by different fuel packages during training fires. *Journal of occupational and environmental hygiene*, 16(8), 532–543. <https://doi.org/10.1080/15459624.2019.1617870>.
- [13] Alarie, Y. (2002). Toxicity of Fire Smoke. *Critical Reviews in Toxicology*, 32, 259 - 289. <https://doi.org/10.1080/20024091064246>.

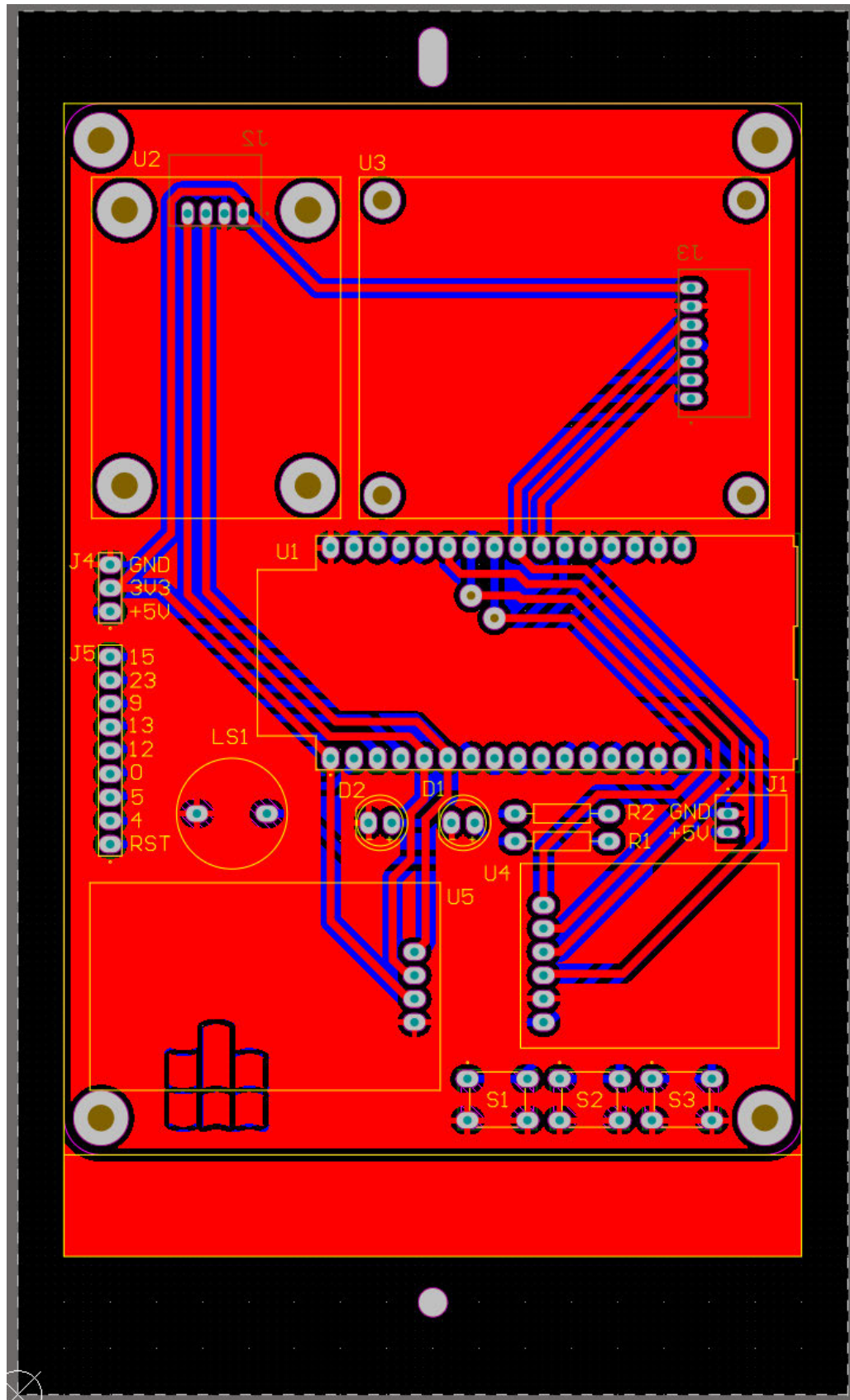
- [14] Inkster, S. E. (2004). Health hazard assessment of CO poisoning associated with emissions from a portable, 5.5 Kilowatt, gasoline-powered generator. *Washington, D.C.: U.S. Consumer Product Safety Commission*. <https://www.cpsc.gov/s3fs-public/pdfs/portgenco.pdf>. [abgerufen am 06.05.2024].
- [15] Wang, Z. (2013). Overview on fire hazard grading standard of flammable and explosive dangerous goods. *Chemical Industry and Engineering Progress*.
- [16] Litton, C., Perera, I., Harteis, S., Teacoach, K., Derosa, M., Thomas, R., & Smith, A. (2018). Some relevant parameters for assessing fire hazards of combustible mine materials using laboratory scale experiments.. *Fuel*, 218, 306-315 . <https://doi.org/10.1016/J.FUEL.2017.12.106>.
- [17] Gray, B. (2016). Spontaneous Combustion and Self-Heating. *SFPE Handbook of Fire Protection Engineering*, 604-632. https://doi.org/10.1007/978-1-4939-2565-0_20.
- [18] Wenograd, J. (1961). The behaviour of explosives at very high temperatures. *Transactions of The Faraday Society*, 57, 1612-1620. <https://doi.org/10.1039/TF9615701612>.
- [19] Yamazoe, N., & Shimanoe, K. (2011). Basic approach to the transducer function of oxide semiconductor gas sensors. *Sensors and Actuators B: Chemical*, 160(1), 1352-1362. <https://doi.org/10.1016/j.snb.2011.09.075>.
- [20] Xu, D., Zhang, Z., & Yu, Z. (2009). Analysis on thermal balance and design principles of catalytic combustion gas sensor. *2009 4th IEEE International Conference on Nano/Micro Engineered and Molecular Systems*, 898-903. <https://doi.org/10.1109/NEMS.2009.5068719>.
- [21] Miura, N., Kato, H., Yamazoe, N., & Seiyama, T. (1986). Amperometric proton-conductor sensor for detecting hydrogen and carbon monoxide at room temperature. *Fundamentals and applications of chemical sensors (ACS Symposium Series, Vol. 309, pp. 203-214)*. American Chemical Society. <https://doi.org/10.1021/bk-1986-0309.ch012>.
- [22] Xu, W., Cai, Y., Gao, S., Hou, S., Yong, Y., Yusen, D., Qingyan, F., Chen, F., & Wu, J. (2020). New understanding of miniaturized VOCs monitoring device: PID-type sensors performance evaluations in ambient air. *Sensors and Actuators B-chemical*, 129285. <https://doi.org/10.1016/j.snb.2020.129285>.
- [23] Kapila, S., Bornhop, D., Manahan, S., & Nickell, G. (1983). Design of a combined photoionization detector and photoionization-based electron-capture detector. *Journal of Chromatography A*, 259, 205-210. [https://doi.org/10.1016/S0021-9673\(01\)88002-X](https://doi.org/10.1016/S0021-9673(01)88002-X).
- [24] Jha, R. (2022). Non-Dispersive Infrared Gas Sensing Technology: A Review. *IEEE Sensors Journal*, 22, 6-15. <https://doi.org/10.1109/jsen.2021.3130034>.
- [25] Li, J., Yu, B., Zhao, W., & Chen, W. (2014). A Review of Signal Enhancement and Noise Reduction Techniques for Tunable Diode Laser Absorption Spectroscopy. *Applied Spectroscopy Reviews*, 49, 666 - 691.
- [26] Dong, L., Tittel, F., Li, C., Sanchez, N., Wu, H., Zheng, C., Yu, Y., Sampaolo, A., & Griffin, R. (2016). Compact TDLAS based sensor design using interband cascade lasers for mid-IR trace gas sensing.. *Optics express*, 24 6, A528-35 . <https://doi.org/10.1364/OE.24.00A528>.

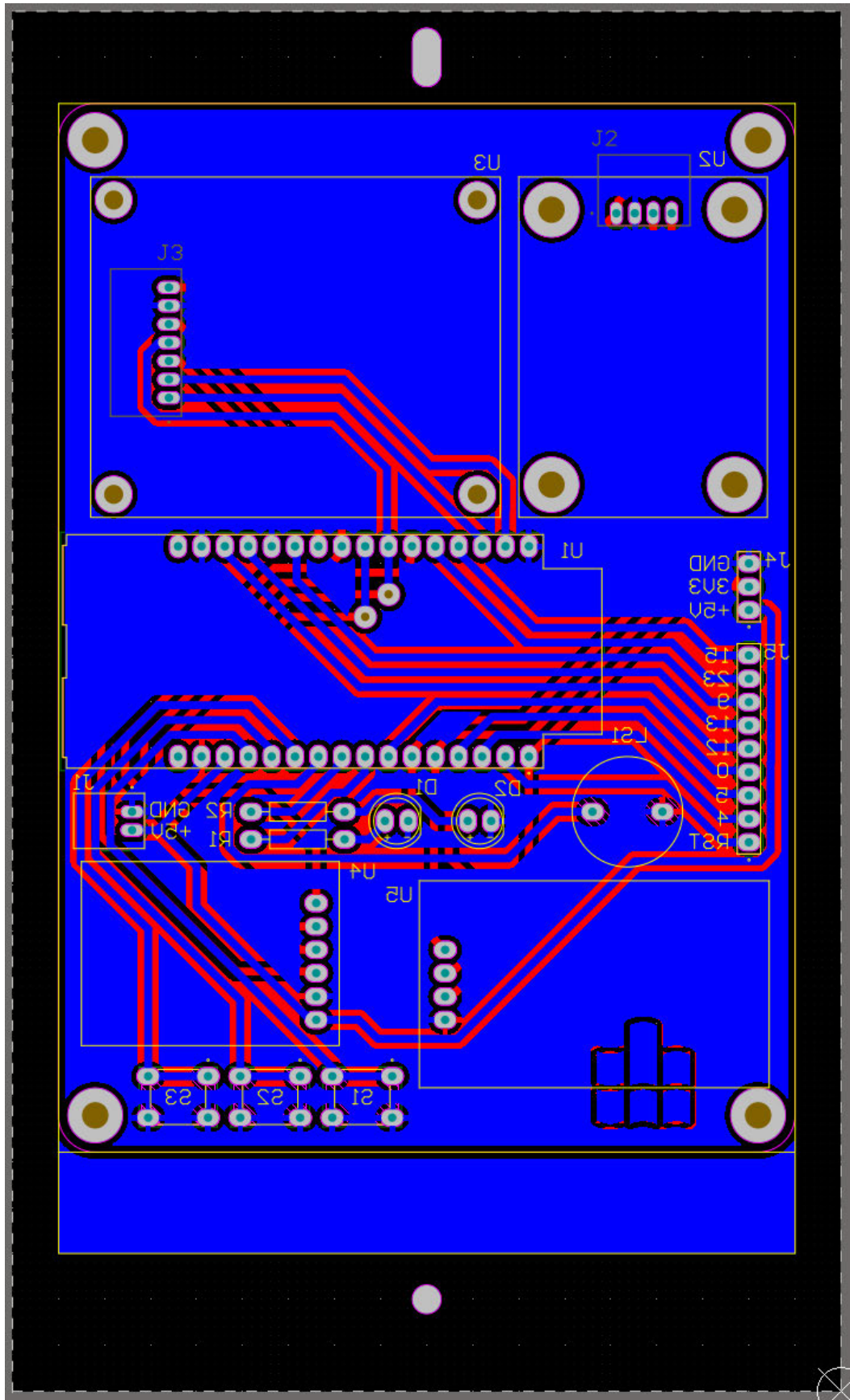
- [27] Cao, Y., Sanchez, N., Jiang, W., Griffin, R., Xie, F., Hughes, L., Zah, C., & Tittel, F. (2015). Simultaneous atmospheric nitrous oxide, methane and water vapor detection with a single continuous wave quantum cascade laser.. *Optics express*, 23 3, 2121-32 . <https://doi.org/10.1364/OE.23.002121>.
- [28] Li, J., Chen, W., & Yu, B. (2011). Recent Progress on Infrared Photoacoustic Spectroscopy Techniques. *Applied Spectroscopy Reviews*, 46, 440 - 471. <https://doi.org/10.1080/05704928.2011.570835>.
- [29] Kah, D.-H. (2020). Development of wireless real-time gas detector system for chemical protection performance test of personal protective equipment. *Journal of the Korea Institute of Military Science and Technology*, 23(3), 294-301. <https://doi.org/10.9766/KIMST.2020.23.3.294>.
- [30] Fan, Y., Zhu, X., Sui, H., Sun, H., & Wang, Z. (2019). Design and Application of Toxic and Harmful Gas Monitoring System in Fire Fighting. *Sensors* (Basel, Switzerland), 19. <https://doi.org/10.3390/s19020369>.
- [31] Gong, D.-H., & Shin, S.-J. (2018). Comparative Analysis between Super Loop and FreeRTOS Methods for Arduino Multitasking. *The Journal of the Institute of Webcasting, Internet and Telecommunication*, 18, 133-137. <https://doi.org/10.7236/JIIBC.2018.18.6.133>.
- [32] Barsan, Michael E. (2007). *NIOSH pocket guide to chemical hazards*. https://stacks.cdc.gov/view/cdc/21265/cdc_21265_DS1.pdf. [abgerufen am 21.07.2024].
- [33] Alberta Agriculture, Food and Rural Development. (2004). Methane (CH₄) Safety. Agri-Facts. <https://www1.agric.gov.ab.ca/%24department/deptdocs.nsf/all/agdex9038/%24file/729-2.pdf>. [abgerufen am 21.07.2024].
- [34] Keçebaş, A., & Kayfeci, M. (2019). Hydrogen properties. In F. Calise, M. D. D'Accadia, M. Santarelli, A. Lanzini, & D. Ferrero (Eds.), *Solar Hydrogen Production* (pp. 3-29). Academic Press. <https://doi.org/10.1016/B978-0-12-814853-2.00001-1>.
- [35] Immonen, R., & Hämäläinen, T. (2022). Tiny Machine Learning for Resource-Constrained Microcontrollers. *Journal of Sensors*. <https://doi.org/10.1155/2022/7437023>.
- [36] Han, H., & Siebert, J. (2022). TinyML: A Systematic Review and Synthesis of Existing Research. *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, 269-274. <https://doi.org/10.1109/ICAIIIC54071.2022.9722636>.

Komponentenliste

Komponententyp	Bezeichnung / Spezifikation	Anzahl
Mikrocontroller	ESP32-C6-DevKitC-1-N8 Entwicklungsboard	1
Gassensor	DFRobot Gravity: MEMS Gas Sensor (CO, Alkohol, NO2 & NH3) - I2C - MiCS-4514; SKU: SEN0377	1
OLED-Display	Waveshare 1.5 Zoll RGB OLED Display Module, 65K RGB Farben, 128x128, SPI; SKU: 14747	1
Taktiler Schalter	Sensorschalter SWITCH TACTILE - PTS645SL432LFS	1
Signalgeber	Aktive Summer - TMB12A05	1
LED	F5, rot und grün	je 1
Widerstände	470 Ω oder 1 k Ω , je nach LED-Modell	2
SD-Kartenslot	DFRobot Fermion: MicroSD Card Module for Arduino (Breakout); SKU: DFR0229	1
Speicherkarte	Intenso microSDHC 8GB Class 10	1
Echtzeituhr	Az-delivery RTC-Modul DS3231 mit I2C-Schnittstelle	1
Spannungsregler	LM2596 DC-DC Step-Down Wandler Modul, 2A	1
Spannungsmesser	Analoges Spannungssensor-Modul	1
Batteriehalter	Batteriegehäuse für 9V-Akkus mit integriertem Ein/Aus-Schalter	1
Steckverbinder	Buchsenleiste, 2,54 mm Rastermaß	nach Bedarf
Steckverbinder	PH2.0 Buchsengehäuse, 2-polig, 4-polig und 7-polig	je 1
Verbindungskabel	PH2.0 kompatible Anschlusskabel, passend für 2-, 4- und 7-polige Buchsen	je 1
Montagematerial	Abstandsbolzen, M3 und M2	nach Bedarf
Befestigungsmaterial	Schrauben, M3 und M2	nach Bedarf
Lüfter	12V Axiallüfter, 50x50x10mm, 2-Pin Anschluss	1

Leiterplatten





Codeausschnitte

Konfiguration

```

/* Konfiguration .....*/

// Vorheizzeit-Einstellung
constexpr uint8_t CALIBRATION_TIME = 3; // Standard-Vorheizkalibrierungszeit beträgt 3 Minuten

// Mutex-Einstellungen
#define USE_MUTEX true // Mutex (Mutual Exclusion), standardmäßig aktiviert für SPI und I2C

// WLAN-Einstellungen
constexpr char *WIFI_SSID = " "; // WLAN-Name
constexpr char *WIFI_PASSWORD = " "; // WLAN-Passwort
constexpr uint8_t WIFI_INIT_RETRY = 5; // WLAN-Neustartversuche beim Start

// Zeiteinstellungen
constexpr int32_t UTC_OFFSET_SEC = 3600; // UTC-Differenz, Winterzeit in De
constexpr int32_t DAYLIGHT_OFFSET_SEC = 3600; // Sommerzeit-Einstellung, Sommerz
constexpr char DAYS_OF_THE_WEEK[7][12] = { "Sonntag", "Montag", "Dienstag", "Mittwoch", "Donnerstag", "Freitag", "Samstag" }; // Namen der Wochentage
constexpr char *NTP_SERVER[] = { "pool.ntp.org", "time.nist.gov", "time.windows.com" }; // NTP-Serveradressen
constexpr uint8_t NTP_MAX_RETRY = 2; // Maximale Anzahl der Wiederholun

// Abtastung und Bildschirm-Einstellungen
constexpr uint16_t SAMPLING_INTERVAL = 250; // Abtastintervall des Gassensors (Millisekunden), Änderungen sind normalerweise nicht erforderlich
constexpr uint16_t REFRESH_INTERVAL = 250; // Bildschirmaktualisierungsintervall (Millisekunden), Änderungen sind normalerweise nicht erforderlich

// Datenverarbeitungseinstellungen
constexpr uint8_t MMA_WINDOW_SIZE = 5; // MMA-Filter Gleitfenstergröße
constexpr float MMA_WEIGHTS[MMA_WINDOW_SIZE] = { 0.1, 0.15, 0.2, 0.25, 0.3 }; // MMA-Filter Gewichtsarray, wobei die neuesten Daten das größte Gewicht haben

// Alarmp-Einstellungen
uint8_t GAS_CHECK_time = 3; // Globale Variable, Schwellenwert für die Anzahl der aufeinanderfolgenden Konzentrationsmessungen (Anzahl der aufeinanderfolgende
uint16_t ALARM_interval = 250; // Globale Variable, Intervallzeit für das intermittierende Umschalten des akustisch-visuellen Alarms (in Millisekunden)
uint16_t ALARM_delay = 1500; // Globale Variable, Verzögerungszeit nach dem Ende des Alarms (in Millisekunden)

// Hauptanzeigeneinstellungen
uint8_t YELLOW_percentage = 80; // Globale Variable, Startschwelle für die Anzeige der Gaskonzentration in Gelb (prozentualer Anteil des festg
constexpr uint16_t KEEP_REFRESH_TIME = 3000; // Dauer der kontinuierlichen Aktualisierung des Displays nach einer Gaswertänderung (Millisekunden)

// SD-Karteneinstellungen
constexpr uint32_t SD_CHECK_PERIOD_MS = 60 * 1000; // Überprüfungsintervall für SD-Karte (Millisekunden), standardmäßig 1 Minute
uint8_t MIN_FREE_SPACE_percentage = 10; // Globale Variable, minimaler verbleibender Speicherplatz auf der SD-Karte in Prozent, standardmäßig 10%
constexpr char *LOG_FILE_EXTENSION = ".txt";
constexpr char *CONFIG_FILE_NAME = "/config.json";

// Tastatureinstellungen
constexpr uint16_t LONG_PRESS_DURATION = 600; // Dauer bis zur Aktivierung durch Langdruck (Millisekunden)
constexpr uint16_t HOLD_REPEAT_INTERVAL = 200; // Intervall für wiederholte Aktivierungen bei gehaltener Taste (Millisekunden)

// Serielle Schnittstelleneinstellungen
constexpr uint8_t SERIAL_RETRY_TIME = 3; // Versuche zur Herstellung einer seriellen Verbindung

// ThingsBoard-Einstellungen
#define ENCRYPTED false // Kommunikation verschlüsselt?
constexpr char *TOKEN = " "; // TOKEN
constexpr char *THINGSBOARD_SERVER = " "; // Server-IP-Adresse
constexpr uint16_t THINGSBOARD_PORT = ENCRYPTED ? 8883U : 1883U; // Server-Port, abhängig von der Verschlüsselung
constexpr uint16_t MAX_MESSAGE_SIZE = 256U; // Maximale Nachrichtengröße
// ThingsBoard Gemeinsameattribute-Einstellungen
constexpr size_t ATTRIBUTE_COUNT = 14U; // Anzahl der gemeinsamen Attribute [Achtung: Wenn die Anzahl der zu abonnierenden gemeinsa
/* !!! Achtung: Weitere Einstellungen für die ThingsBoard-Gemeinsame-Attribute finden Sie im Abschnitt „ThingsBoard Gemeinsame Attri

// TinyML-Einstellungen
#define SERIAL_PRINT_ROUND true // Enddaten jeder Runde drucken?
#define SERIAL_PRINT_SAMPLE true // Jedes Sample drucken?
constexpr uint8_t SAMPLE_NUMBER = 4; // Anzahl der Samples pro Runde
constexpr uint16_t SAMPLE_TIME_FRAME = 375; // Sampling-Intervall in ms
float CONFIDENCE_threshold = 0.80; // Globale Variable, Schwellenwert für die Konfidenz

// Einstellungen des Spannungssensors (Teiler)
constexpr float R1 = 30000.0f; // Widerstand R1 beträgt 30 kOhm
constexpr float R2 = 7500.0f; // Widerstand R2 beträgt 7,5 kOhm
constexpr float ADC_OFFSET = 0.02; // Manuelle Kalibrierung der ADC-Spannung
constexpr float VOL_OFFSET = 0.00; // Manuelle Kompensation der Batteriespannung
constexpr uint32_t BAT_CHECK_PERIOD_MS = 5 * 1000; // Überprüfungsintervall der Batterie (Millisekunden), standardmäßig 30 Sekunden
constexpr uint32_t ADC_SAMPLING_FREQ = 1000; // ADC-Samplingfrequenz
constexpr uint16_t CONVERSIONS_PER_PIN = 50; // Anzahl der Umwandlungen pro Pin zur Mittelwertberechnung 英
// Einstellung der Batterie-Kapazitätsschätzung
constexpr float BATTERY_MAX_VOLTAGE = 9.0; // Maximale Spannung einer 9V-Batterie
constexpr float BATTERY_MIN_VOLTAGE = 5.4; // Minimale Betriebsspannung einer 9V-Batterie
constexpr float BATTERY_CONNECTION_MIN_THRESHOLD = 3.0; // Annahme: Bei weniger als 3V gilt die Batterie als nicht verbunden
constexpr float BATTERY_CONNECTION_MAX_THRESHOLD = 11.0; // Annahme: Bei mehr als 11V gilt die Batterie als nicht verbunden
#define TEMP_COMPENSATION true // Bei der Spannungsberechnung Temperaturkompensation verwenden

```

Bedienungsanleitung

A. Gefahrstofferkennungssystem

1. Einschalten und Aufwärmen

Dieser Prototyp des Gefahrstofferkennungssystems bietet zwei Stromversorgungsoptionen:

a) USB Typ-C-Stromversorgung für ESP32-C6: Beide integrierten Typ-C-Anschlüsse sind nutzbar; Unterstützung für simultane Stromversorgung über zwei Typ-C-Anschlüsse zur Realisierung eines redundanten Stromversorgungskonzepts.

b) 5V-Stromversorgung über die vorgesehene PH2.0-Schnittstelle

Hinweis: Bitte wählen Sie nur eine der beiden Stromversorgungsmethoden. Eine gleichzeitige Nutzung beider Methoden wird nicht unterstützt.

Systemstartprozess:

Nach dem Einschalten initialisiert sich das System automatisch und überprüft den Verbindungsstatus der Sensoren. Bei erfolgreicher Verbindung wird der Aufwärmmodus aktiviert; bei fehlgeschlagener Verbindung werden Wiederverbindungsversuche unternommen. Überschreitet die Anzahl der Fehlversuche einen definierten Schwellenwert, führt das System einen automatischen Neustart durch.

Aufwärmmodus-Benutzeroberfläche:



Anzeige der verbleibenden Sensoraufwärmzeit (Standardwert: 180 Sekunden); ein Fortschrittsbalken visualisiert den Aufwärmvorgang. Statusanzeigen im unteren Bildschirmbereich:

- Verbindungsstatus des Gassensors (Rot: Fehlgelungen, Grün: Erfolgreich)
- WLAN-Status (Rot: Nicht verbunden, Grün: Verbunden)
- Externer RTC-Status (Rot: Hardware nicht erkannt, Gelb: Verbunden, aber NTP nicht synchronisiert, Grün: Verbunden und NTP synchronisiert)
- MicroSD-Karten-Status (Rot: Hardware nicht erkannt, Gelb: Kartenschacht erkannt, aber keine SD-Karte eingelegt, Grün: SD-Karte betriebsbereit)

Nach Abschluss des Aufwärmvorgangs wechselt das System automatisch zur Hauptbenutzeroberfläche.

2. Hauptbenutzeroberfläche

Die Hauptbenutzeroberfläche ist übersichtlich und intuitiv gestaltet. Sie bietet dem Anwender Echtzeitüberwachungsdaten und Systeminformationen. Nach Aufruf der Hauptbenutzeroberfläche führt das System automatisch eine Reihe von Erkennungsaufgaben im Hintergrund aus, ohne dass der Benutzer manuell eingreifen muss.



a) Oberer Statusbereich: Die Farbgebung passt sich dem Alarmstatus an (Grün: Normal, Gelb: Aufmerksamkeit erforderlich, Rot: Alarm); links im Statusbereich werden das aktuelle Datum und die Uhrzeit in Echtzeit angezeigt; rechts befinden sich fünf Statussymbole:

- TinyML-Modus-Status (Grau: Deaktiviert, Farbig: Aktiviert)
- Alarmaufgaben-Status (Weiße Kurve: Aktiviert, Rotes "X": Deaktiviert)
- WLAN-Status (Normal: Keine Markierung, Gelber Schrägstrich: Verbindung unterbrochen, Roter Schrägstrich: Komplette deaktiviert)
- SD-Karten-Status (Normal: Keine Markierung, Roter Schrägstrich: Nicht verbunden)
- Stromversorgungsstatus (USB-Symbol: USB-Stromversorgung, Batteriesymbol: Batteriebetrieb, Ladezustandsanzeige variiert entsprechend der Kapazität)

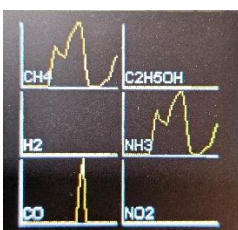
b) Hauptanzeigebereich:

Listenförmige Darstellung der Echtzeit-ppm-Werte für sechs Gase; die Farbgebung der Werte passt sich dem Alarmstatus an; bei Alarmauslösung erscheint rechts neben dem entsprechenden Gasnamen ein dreieckiges Warnsymbol.

Durch kurzes Drücken der Taste S1 am unteren Rand der Leiterplatte können Sie in der Hauptbenutzeroberfläche, zwischen der Haupt- und der sekundären Benutzeroberfläche wechseln.

3. Sekundäre Benutzeroberfläche

Die sekundäre Benutzeroberfläche zeigt gleichzeitig sechs Liniendiagramme der Gaskonzentrationen an. Diese skalieren sich automatisch, um dem Benutzer die Trends der Gaskonzentrationsveränderungen anschaulich zu präsentieren.



4. Einstellungsmenü



Durch langes Drücken der S1-Taste in der Haupt- oder sekundären Benutzeroberfläche gelangen Sie in das Einstellungsmenü. Erneutes langes Drücken der S1-Taste speichert die aktuellen Einstellungen und führt zurück zur Hauptbenutzeroberfläche.

Das Einstellungsmenü präsentiert die Funktionsschalter in Listenform. Von oben nach unten finden Sie den Alarmschalter, den TinyML-Modusschalter sowie die Schwellenwerteneinstellungen für die sechs Gase. Rechts neben dem Namen jeder Einstellungsoption stehen in Klammern die möglichen Zustände oder Minimal- und Maximalwerte, ganz rechts der aktuelle Status der jeweiligen Option.

Durch kurzes Drücken der S1-Taste navigieren Sie zwischen den Einstellungsoptionen. Ein blinkendes weißes Rechteck auf der rechten Seite kennzeichnet die aktuell ausgewählte Option.

Die Tasten S2 und S3 sind nur im Einstellungsmenü aktiv und dienen hauptsächlich zur Anpassung der Einstellungswerte. S2 verringert den Wert, S3 erhöht ihn. Kurzes Drücken von S2 oder S3 ändert den Wert um einen vordefinierten Schritt. Langes Drücken und Halten von S2 oder S3 ermöglicht eine kontinuierliche Wertanpassung in vordefinierten Schritten.

5. Hauptbenutzeroberfläche im TinyML-Modus

Wenn die TinyML-Option im Einstellungsmenü aktiviert wurde, wechselt der Hauptanzeigebereich der Hauptbenutzeroberfläche von der Listenansicht zu einer textbasierten Anzeige.



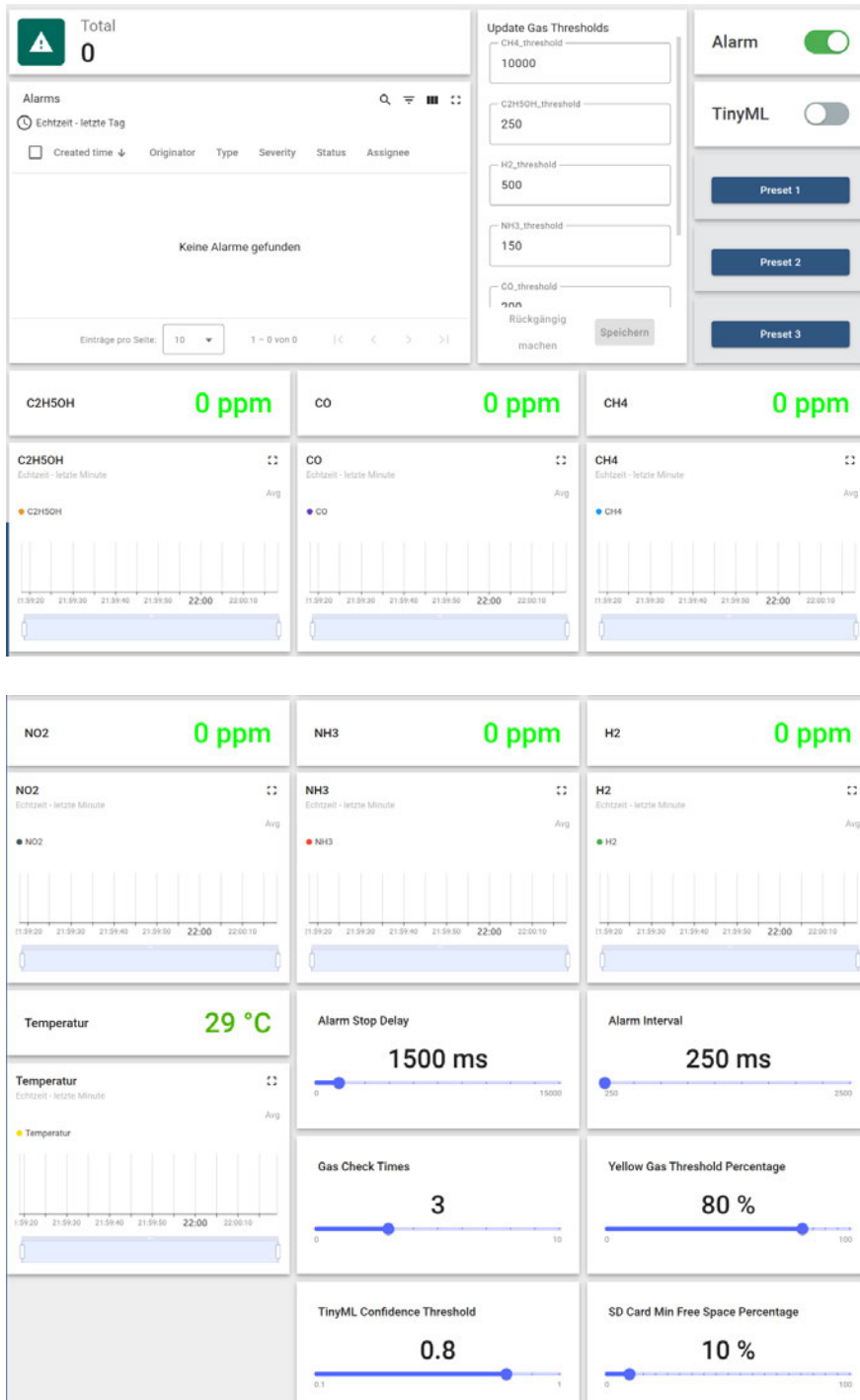
Sobald die Sensorwerte von Null abweichen, werden das vom Modell ermittelte Ergebnis sowie der Konfidenzwert angezeigt.

6. Lokale Datenspeicherung

Bei angeschlossener SD-Karte werden die von den Sensoren erfassten Daten auf der SD-Karte gespeichert. Die Protokolldateien enthalten vollständige Kopfzeilen.

7. IoT-Plattform (Daten-Upload und Fernsteuerung)

Bei bestehender WLAN-Verbindung überträgt das System die von den Sensoren erfassten Informationen in Echtzeit an die im Programm spezifizierte Thingsboard IoT-Plattform auf dem Server. Zudem besteht die Möglichkeit, das System über die Thingsboard-Plattform fernzusteuern.



B. TinyML-Datenerfassungssystem

1. Einschalten und Aufwärmen

Ähnlich wie beim Gefahrstofferkennungssystem beginnt nach dem Einschalten der Aufwärmprozess, dessen Status auf dem Bildschirm angezeigt wird.

2. Datenerfassungsoberfläche

Nach Abschluss des Aufwärmvorgangs wird die Datenerfassungsoberfläche angezeigt. Hier wird lediglich die aktuelle Erfassungsrunde angezeigt.

3. Datenerfassungsmethoden

Bei Verbindung mit einem Host-Computer über die serielle Schnittstelle werden die von den Sensoren erfassten Daten an das Terminal des Host-Computers gesendet.

Bei angeschlossener SD-Karte werden die von den Sensoren erfassten Daten direkt als .csv-Datei auf der SD-Karte gespeichert. Diese Dateien verfügen über vollständige Kopfzeilen und können unmittelbar für Trainingszwecke verwendet werden.

Erklärung zur Codeverfügbarkeit

Der gesamte für diese Forschungsarbeit relevante Quellcode wurde als Open-Source-Software veröffentlicht und ist im folgenden GitHub-Repository zugänglich: <https://github.com/Atlantis-Gura/hazardous-substances-detection>

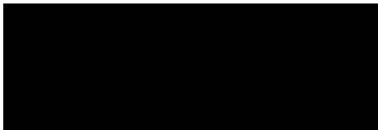
Dieses Repository beinhaltet sämtlichen in dieser Studie verwendeten Quellcode, einschließlich aller Drittanbieterbibliotheken und Softwarewerkzeuge. Detaillierte Anleitungen zur Nutzung des Codes, Lizenzinformationen sowie weiterführende Erläuterungen finden Sie in der README-Datei und den begleitenden Dokumentationen innerhalb des Repositories.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.



13.08.2024 Chemnitz