HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

MASTERARBEIT

Herr

Xin Chu

Umsetzung eines autarken sensorbasierten Alarmsystems fur Einsatzkrafte mitels eines ZigBee-Mesh-Netzwerkes

2023

Fakultät: Ingenieurwissenschaften

MASTERARBEIT

Umsetzung eines autarken sensorbasierten Alarmsystems fur Einsatzkrafte mitels eines ZigBee-Mesh-Netzwerkes

Autor:

Herr Xin Chu

Studiengang:

Elektrotechnik-Automation

Seminargruppe:

EA21wA-M

Erstprüfer:

Prof. Dr. -Ing. Michael Kuhl

Zweitprüfer:

M. Sc. Kevin Blümel

Einreichung:

Mittweida,08.08.2023

Faculty of engineering sciences

MASTER THESIS

Implementation of an autarkic sensor-based alarm system for emergency forces by means of a ZiaBee mesh network

author:

Mr. Xin Chu

course of studies:

Electrotechnical-Automation

seminar group:

EA21wA-M

first examiner:

Prof. Dr.-Ing. Michael Kuhl

second examiner:

M.Sc. Kevin Blümel

submission:

Mittweida,08.08.2023

Bibliografische Angaben

Nachname, Vorname: Chu Xin

Umsetzung eines autarken sensorbasierten Alarmsystems fur Einsatzkrafte mitels eines ZigBee-Mesh-Netzwerkes

Implementation of an autarkic sensor-based alarm system for emergency forces by means of a ZiaBee mesh network

119 Seiten, Hochschule Mittweida, University of Applied Sciences,
Fakultät Ingenieurwissenschaften, Masterarbeit, 2023

Inhaltsverzeichnis

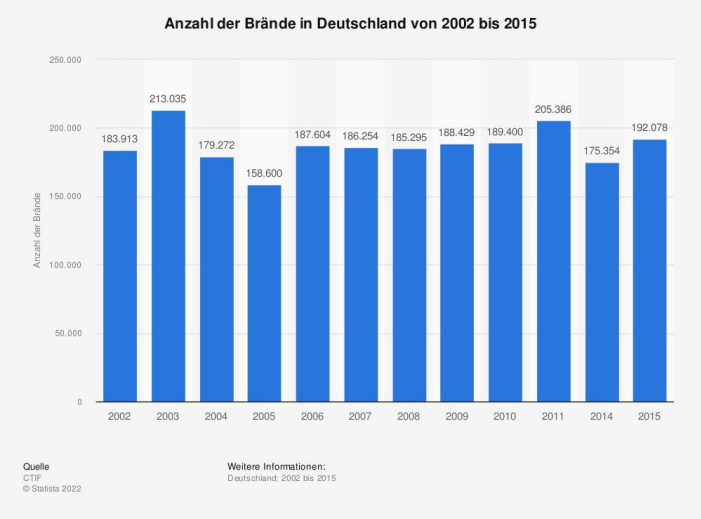
Abbildungsverzeichnis

Ein (C) im Namen weist darauf hin, dass es sich bei dem Bild nicht um ein Original handelt

Tabellenverzeichnis

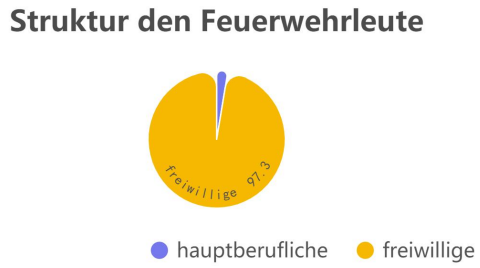
Diagramm

0 Einleitung



1Anzahl der Brände in Deutschland von 2002 bis 2015

Jedes Jahr kommt es in Deutschland zu einer großen Anzahl von Brandereignissen, die enorme wirtschaftliche Verluste und Opfer verursacht haben. Nach Angaben der Deutschen Feuerwehr brechen jedes Jahr etwa 180.000 Brände aus, etwa die Hälfte davon sind Innenbrände. Diese Brände haben Tausende Opfer und wirtschaftliche Schäden in Milliardenhöhe verursacht. Jedes Jahr nimmt eine große Anzahl von Feuermannn an der Brandrettung und Brandbekämpfung teil.Am 31.12.2020 waren in Deutschland 1.006.638 Personen in den Freiwilligen Feuermann aktiv. In den Berufsfeuerwehren waren 35.041 Personen tätig.Jedes Jahr verlieren einige Feuermann ihr Leben aufgrund von Unfällen im Dienst. In den letzten Jahren war die Sterblichkeitsrate von Feuermannn in Deutschland relativ niedrig, wobei durchschnittlich 1-3 Feuermann pro Jahr aufgrund ihrer Pflichten starben. Diese Todesfälle sind oft auf den Rauch, die Flammen und die Hitze des Feuers zurückzuführen.



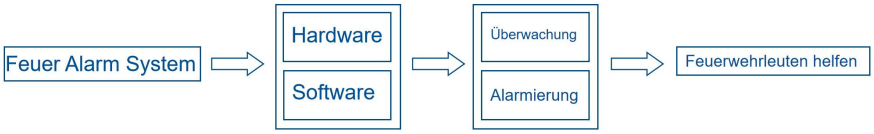
Um den Feuermann eine sicherere Teilnahme an der Brandbekämpfung am Brandort zu ermöglichen, beteiligte ich mich am Projekt Brandmeldeanlage mit dem Ziel, ein vollständiges Informationsübertragungssystem aufzubauen. Der spezifische Inhalt besteht darin, den drahtlose Übertragungssystem zu verwenden, um ein vollständiges Branderkennungs-, Informationsübertragungs- und Analysesystem basierend auf dem Mesh-Netzwerk aufzubauen.

Struktur den Feuermann：https://www.feuerwehrverband.de/presse/statistik/

1 Feuer Alarm System

1.1 Was ist ein Feuer Alarm System

Das Feuer Alarm System besteht aus einer Reihe von Geräten und Software zur Überwachung und Alarmierung und wird verwendet, um rechtzeitig einen Alarm auszulösen, wenn ein Feuer in Innenräumen auftritt. Das System umfasst ein Sensorsystem, ein akustisches und visuelles Alarmsystem und ein Informationsübertragungssystem. Das Feuer Alarm System kann Feuermannn helfen, die Situation im Brandfall rechtzeitig zu ermitteln und die Sicherheit der Feuermann zu gewährleisten.



2Feuer Alarm System

Feuermann tragen zwei Arten von Device mit sich, bevor sie den Brandort betreten, eine ist die am Brandort platzierte Endgeräte und die andere ist die am Körper befestigte Alarmgeräte. Nachdem Feuermann den Brandort betreten haben, platzieren sie regelmäßig Endgeräte an den Orten, an denen sie vorbeikommen, wie zum Beispiel Fluren oder Räumen. Nach dem Start des Endgerätes sendet es in regelmäßigen Abständen Daten an die Leitstelle und zeichnet in Echtzeit die Arbeitsbedingungen der Geräte selbst und die Temperaturveränderungen am Brandort auf. Wenn der Bereich, an dem die Feuermann vorbeikommen, erneut brennt, sendet das Endgerät Alarmdaten, und die Leitstelle sieht die vom Endgerät zurückgesendeten Temperaturänderungsdaten und erhält eine Alarmerinnerung. Die Alarmgeräte der Feuermann schlägt ebenfalls Alarm und macht die Feuermann auf Veränderungen am Brandort aufmerksam. Nach Abschluss der Brandbekämpfungsarbeiten werden die Endgeräte recycelt und bewertet.

1.2 Komponenten einen Feuer Alarm System

Die Brandüberwachung muss viele Faktoren berücksichtigen, darunter:

1. Brandmelder: Verwenden Sie verschiedene Arten von Brandmeldern, um Brände zu überwachen, wie z. B. Rauchmelder, Infrarot-Flammenmelder, pyroelektrische Induktionsmelder usw.

2. Automatisches Brandmeldesystem: Wenn der Brandmelder einen Brand erkennt, muss er das automatische Brandmeldesystem auslösen, um das zuständige Personal und die Feuerwehr zu benachrichtigen.

3. Brandschutz- und Kontrollsystem: Das Brandüberwachungssystem sollte mit dem Brandschutz- und Kontrollsystem kombiniert werden, einschließlich Brandmeldesystem, Brandbekämpfungssystem und Brandnotevakuierungssystem.

4. Netzwerkverbindung: Das Brandmeldesystem sollte in der Lage sein, sich mit anderen Systemen zu verbinden, wie z. B. Sicherheitsüberwachungssystem und Brandleitstelle usw.

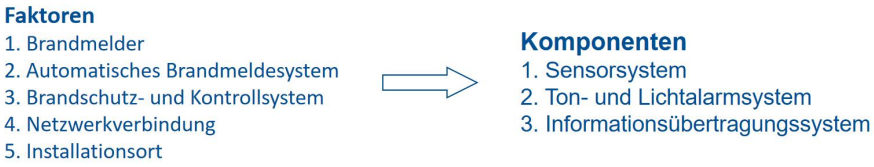
5. Installationsort: Bei der Auswahl eines Brandmelders müssen Sie auf den Installationsort achten, um sicherzustellen, dass der Melder Feuer effektiv erkennen kann.

Für den Aufbau einen Feuer Alarm System werden daher folgende Teile benötigt:

1. Sensorsystem: dient zur Erkennung von Feuer.

2. Ton- und Lichtalarmsystem: Wird verwendet, um im Brandfall einen Alarm auszulösen.

3. Informationsübertragungssystem: Wird zur Überwachung von Bränden verwendet, das Brandinformationen an Feuermann,Leitstelle senden kann.



3Komponenten

1.3 Eigenschaften

Das Feuer Alarm System hat die folgenden Eigenschaften:

Das Endgerät kann den Übertragungsweg frei wählen und hat die Funktion, nach Verbindungsabbruch selbstständig andere Teilnehmer zu finden.

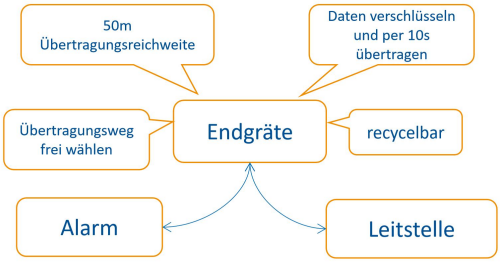
Endgeräte sind recycelbar.

Die maximale Übertragungsreichweite zwischen EndKnoten beträgt 50 Meter.

Daten werden alle 10 Sekunden übertragen. Die Daten werden verschlüsselt, um die Genauigkeit und Sicherheit der Daten zu gewährleisten.

Das Alarmgerät(Feuermann tragen immer) kann gleichzeitig mit der Alarmierung prüfen, welches Endknoten den Alarm ausgibt.

Alle Daten können drahtlos an die Leitstelle gesendet werden.Die Leitstelle kann den Gerätestatus überprüfen, die Temperaturänderung der Brandstelle und die Brandalarmerinnerung überwachen.



4 Eigenschaften

2 Übertragungssystem

Der wichtigste Teil der Feuer Alarm System ist das Informationsübertragungssystem.

Es gibt viele Lösungen für die drahtlose Übertragung über kurze Entfernungen auf dem Markt. Im Folgenden sind einige gängige Lösungen für die drahtlose Übertragung über kurze Entfernungen, die ich gefunden habe, und ihre Vor- und Nachteile aufgeführt:

1.Bluetooth: Bluetooth ist eine drahtlose Technologie mit kurzer Reichweite. Bluetooth hat eine hohe Datenübertragungsrate und Zuverlässigkeit und unterstützt auch die Verbindung mehrerer Geräte. Der Nachteil ist, dass die Kommunikationsreichweite von Bluetooth relativ kurz ist, normalerweise nicht mehr als 10 Meter, und die Übertragungsrate beeinträchtigt wird, wenn mehrere Geräte verbunden sind.

2.Wi-Fi Direct: Wi-Fi Direct ist eine Punkt-zu-Punkt-Wi-Fi-Verbindungsmethode, die zum direkten Herstellen einer Verbindung zwischen Geräten ohne Netzwerkgerät geeignet ist. Wi-Fi Direct kann eine höhere Datenübertragungsrate und eine längere Kommunikationsentfernung bieten, erfordert jedoch, dass das Gerät die Wi-Fi Direct-Funktion unterstützt.

3.NFC (Near Field Communication): NFC ist eine auf Funkwellen basierende Kommunikationstechnologie mit kurzer Reichweite, die häufig in Anwendungen wie Zahlungs- und Zugangskontrolle verwendet wird. Die NFC-Kommunikationsdistanz ist sehr kurz, normalerweise nicht mehr als 10 cm, aber die Sicherheit ist hoch.

4.Zigbee: Zigbee ist eine drahtlose Kommunikationstechnologie mit geringem Stromverbrauch und niedriger Geschwindigkeit und kurzer Reichweite, die hauptsächlich in IoT-Anwendungsszenarien verwendet wird. Zigbee eignet sich für Low-Power-Anwendungen und kann eine äußerst zuverlässige Kommunikation und Konnektivität mit mehreren Geräten bieten, aber seine Kommunikationsreichweite ist relativ gering.

5.LoRa: LoRa (Long Range) ist eine drahtlose Kommunikationstechnologie mit geringem Stromverbrauch und großer Bandbreite.LoRa eignet sich für Anwendungsszenarien, die Fernkommunikation und groß angelegte Knotenverbindungen erfordern, aber seine Übertragungsrate ist langsam. Daher ist es nicht für Anwendungsszenarien geeignet, die eine Hochgeschwindigkeits-Datenübertragung erfordern.

	Bluetooth	NFC	Wi-Fi Direct	LoRa	Zigbee
Menge	Max. 7	eins zu eins	Dutzende	Hunderte	Hunderte
Übertragungsdistanz (Max)	10m	10cm	200m	2km	50m

Übertragungsgeschwindigkeit(Max)	2 Mbps	424 kbps	250 Mbps	50 kbps	250 kbps
Sicherheit	AES,ECDH	DES,3DES,AES	WPA2,AES	AES	AES
Energieverbrauch	Niedrig	Sehr Niedrig	Hoch	Sehr Niedrig	Niedrig
Kosten	Günstig	Günstig	Teuer	Teuer	Günstig

1Parameter des Übertragungsprotokolls

AES: Advanced Encryption Standard

ECDH: Elliptic Curve Diffie-Hellman

DES: Data Encryption Standard

WPA2: Wi-Fi Protected Access II

2.1 Warum wählen Zigbee

Das Feuer Alarm System zeichnet sich durch Gerätekommunikation über kurze Entfernungen, geringen Stromverbrauch, hohe Zuverlässigkeit und Konnektivität mit mehreren Geräten aus.Durch den Vergleich verschiedener drahtloser Kurzstrecken-Übertragungsschemata und die Kombination der Eigenschaften des Feuer Alarm Systems entschied ich mich schließlich für die drahtlose Zigbee-Kommunikationstechnologie.Und die Erweiterungsleistung von Zigbee ist sehr gut, was dazu beiträgt, später weitere Funktionen hinzuzufügen。

2.2 Was ist Zigbee

5 Zigbee Mark(C)

Zigbee ist eine drahtlose Kommunikationstechnologie mit geringem Stromverbrauch und niedriger Datenrate, die eine geschichtete Topologie für die Netzwerkkonnektivität verwendet.Es wurde speziell für Fernsteuerungs- und Sensornetzwerke wie Heimautomation, Smart Grid, Industrieautomation, Gesundheitswesen und andere Bereiche entwickelt.

Das Topologiemodell des Zigbee-Netzwerks ist hauptsächlich in drei Typen unterteilt: Netzwerkschicht, Datenverbindungsschicht und physikalische Schicht.

1. Netzwerkschicht: definiert die Struktur des Netzwerks, z. B. wie Geräte organisiert und Daten geroutet werden. Auf der Netzwerkebene gibt es zwei Haupttopologien: Sternnetzwerk und Ringnetzwerk. Ein Sternnetzwerk bedeutet, dass alle Geräte mit

einem zentralen Gerät verbunden sind, während ein Ringnetzwerk bedeutet, dass alle Geräte eine Ringstruktur bilden.

2. Datenverbindungsschicht: definiert die Art und Weise der Datenübertragung, einschließlich des Verbindungsaufbaus, des Sendens von Daten und der Bestätigung des Datenempfangs.

3. Physikalische Schicht:definiert die physikalischen Eigenschaften der Kommunikation wie Frequenz, Datenrate, Leistung usw. Die physikalische Schicht von Zigbee übernimmt hauptsächlich den IEEE 802.15.4-Standard.

2.3 Merkmale von Zigbee

1. Geringer Stromverbrauch: Zigbee-Geräte verwenden normalerweise Funkwellen mit geringer Leistung, sodass sie mit langer Batterieleistung betrieben werden können. Im Standby-Modus mit geringem Stromverbrauch können zwei gewöhnliche AA-Batterien 6 bis 24 Monate lang verwendet werden.

2. Niedrige Kosten: Das Zigbee-Protokoll ermöglicht die Verwendung einfacher und kostengünstiger Hardware.

3. Miniaturisierung: Zigbee-Geräte sind normalerweise klein, sodass sie leicht in andere Geräte integriert werden können.

4. Niedrige Geschwindigkeit: Die Zigbee-Datenübertragungsrate ist relativ langsam und eignet sich für die Übertragung kleiner Datenmengen.

5. Mehrpunktkommunikation: Zigbee ermöglicht die Kommunikation zwischen mehreren Geräten.

6. Zuverlässigkeit: Das Zigbee-Protokoll ist zuverlässig und kann eine korrekte Datenübertragung sicherstellen.

7. Große Netzwerkkapazität: Das Netzwerk kann 65.000 Geräte aufnehmen.

8. Kleine effektive Reichweite: Die effektive Reichweite beträgt 10 bis 75 Meter (mit einem Signalverstärker kann er 1 km übertragen)

6Merkmale

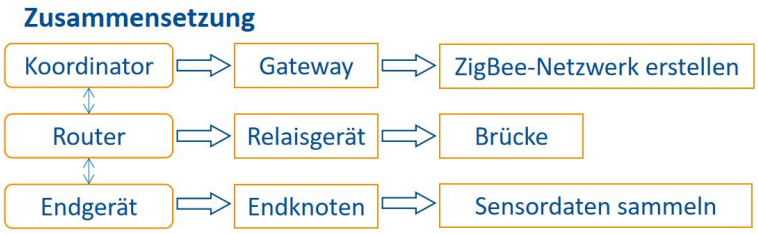
2.4 Zusammensetzung

In einem Zigbee-Netzwerk hat jedes Gerät eine eindeutige 16-Bit-Adresse und kann drei Rollen zugewiesen werden: Coordinator (Koordinator), Router (Router) und End-Device (Endgerät).

1. ZigBee-Koordinator: Als Gateway im ZigBee-Netzwerk ist er für die Erstellung des gesamten ZigBee-Netzwerks und die Realisierung der Datenkommunikation mit dem Computer über die serielle Schnittstelle verantwortlich.

2. ZigBee-Router: Als Relaisgerät im ZigBee-Netzwerk können ZigBee-Endgeräte dem ZigBee-Netzwerk über den ZigBee-Router beitreten, um das gesamte ZigBee-Netzwerk zu erweitern.

3. ZigBee-Endgerät: Als Endknoten in einem ZigBee-Netzwerk ist es sozusagen das „Ende“ des Netzwerks. Es kann dem ZigBee-Netzwerk beitreten und wird hauptsächlich zum Sammeln von Daten und zum Senden von Steuerinformationen verwendet.

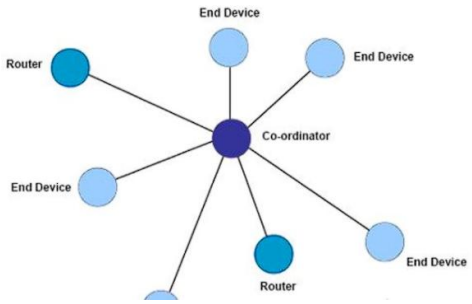


7Zusammensetzung

2.5 Netzwerktopologie

Es gibt drei Haupttypen von Zigbee-Netzwerktopologien:

1. Sterntopologie



8 Sterntopologie

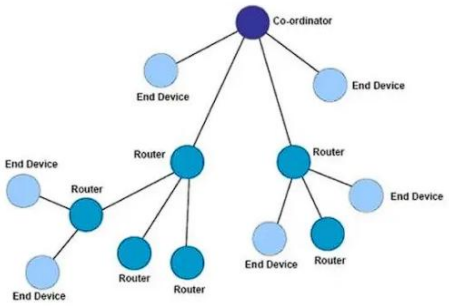
Die Sterntopologie ist die einfachste Topologie im Zigbee-Netzwerk und zeichnet sich dadurch aus, dass alle Geräte direkt oder indirekt mit dem Koordinator verbunden sind.

Der Vorteil dieser Topologie besteht darin, dass sie leicht zu verstehen, einfach bereitzustellen und zu verwalten ist und dass es keine redundanten Router-Knoten gibt, was die Komplexität des Netzwerks reduziert.

Allerdings hat die Sterntopologie auch ihre Nachteile: Da alle Daten den Koordinator passieren müssen, wird der Koordinator zum Engpass im gesamten Netzwerk, was zu

einer eingeschränkten Netzwerkleistung führt. Wenn der Koordinator ausfällt, fällt das gesamte Netzwerk aus.

2.Baumtopologie



9 Baumtopologie

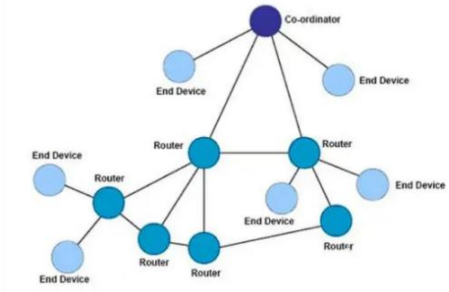
Die Baumtopologie ist die am häufigsten verwendete Topologie in Zigbee-Netzwerken und zeichnet sich dadurch aus, dass alle Geräte mit dem Wurzelknoten (Koordinator) verbunden sind.

Der Vorteil dieser Topologie besteht darin, dass sie hochgradig skalierbar und zuverlässig ist, da Datenpakete über mehrere Pfade übertragen werden können, wodurch die Auswirkungen eines Single Point of Failure auf das Netzwerk reduziert werden.

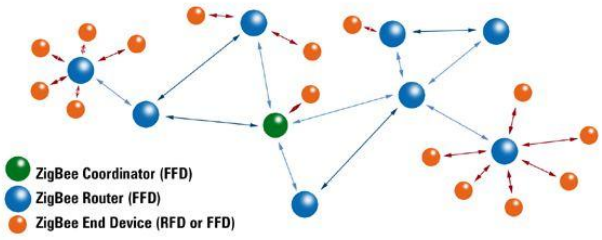
Der Nachteil besteht darin, dass mehr Router-Knoten benötigt werden, um das gesamte Netzwerk zu warten, was zu einer erhöhten Netzwerkkomplexität und höheren Verwaltungs- und Bereitstellungskosten führt.

Die Baumtopologie kann die Netzwerkleistung optimieren und die Netzwerkeffizienz und -zuverlässigkeit verbessern, indem sie Routingalgorithmen verwendet.

3.Mesh-Topologie



10 Mesh-Topologie 1



11 Mesh-Topologie 2(C)

Mesh-Topologie ist eine Multipath-Topologie, die auch als Mesh-Topologie bezeichnet wird. Seine Besonderheit besteht darin, dass alle Geräte miteinander verbunden werden können und jedes Gerät zu einem Durchgangspunkt werden kann, um Datenpakete untereinander zu übertragen.

Der Vorteil dieser Topologie besteht darin, dass sie mehrere Pfadoptionen bereitstellen kann, um die Zuverlässigkeit und Belastbarkeit des Netzwerks zu erhöhen.Wenn einige Geräte ausfallen, kann die Datenübertragung über andere Pfade durchgeführt werden, und das Netzwerk kann weiter betrieben werden.

Der Nachteil besteht darin, dass die Netzwerkverwaltungs- und Bereitstellungskosten hoch sind, da eine große Anzahl von Routerknoten konfiguriert und gewartet werden muss, was zu einer erhöhten Netzwerkkomplexität führt.

Im Mesh-Netzwerk von Zigbee fungiert jedes Gerät als Router, um Daten vom Sender zum Empfänger zu übertragen.Das bedeutet, dass selbst bei Ausfall einiger Geräte Daten über andere Geräte weitergeleitet werden können, um die Kontinuität des Netzwerks zu gewährleisten.

Zigbee kann sein eigenes Mesh-Netzwerkprotokoll verwenden, um die Mesh-Netzwerkstruktur zu realisieren.

In einem Zigbee Mesh-Netzwerk kann jeder Knoten direkt oder indirekt mit anderen Knoten kommunizieren. Zwei Knoten heißen direkt benachbart, wenn sie zwischen ihnen direkt erreichbar sind. Andernfalls muss die Kommunikation zwischen ihnen über einige Zwischenknoten übertragen werden.

Beim Aufbau eines Mesh-Netzwerks sucht jeder Knoten aktiv nach direkten Nachbarknoten und stellt eine direkte Nachbarbeziehung her, nachdem er sie gefunden hat. Im gesamten Netzwerk führt jeder Knoten eine Routing-Tabelle, in der der kürzeste Weg von diesem Knoten zu anderen Knoten festgehalten wird.

Wenn Daten vom Quellknoten gesendet werden, sucht er zuerst den Standort des Zielknotens in seiner Routing-Tabelle und liefert die Daten dann auf dem kürzesten Weg an den Zielknoten. Wenn festgestellt wird, dass ein Knoten während des Übertragungsprozesses nicht erreichbar ist, versucht der Knoten, die direkte Nachbarbeziehung wiederherzustellen und die Routing-Tabelle zu aktualisieren.

Das Zigbee-Mesh-Netzwerk hat die Eigenschaften der automatischen Erkennung, automatischen Konfiguration, des automatischen Routings und der automatischen Rekonstruktion usw. Diese Eigenschaften machen es für den Aufbau großer und hochgradig verteilter drahtloser Netzwerke geeignet.

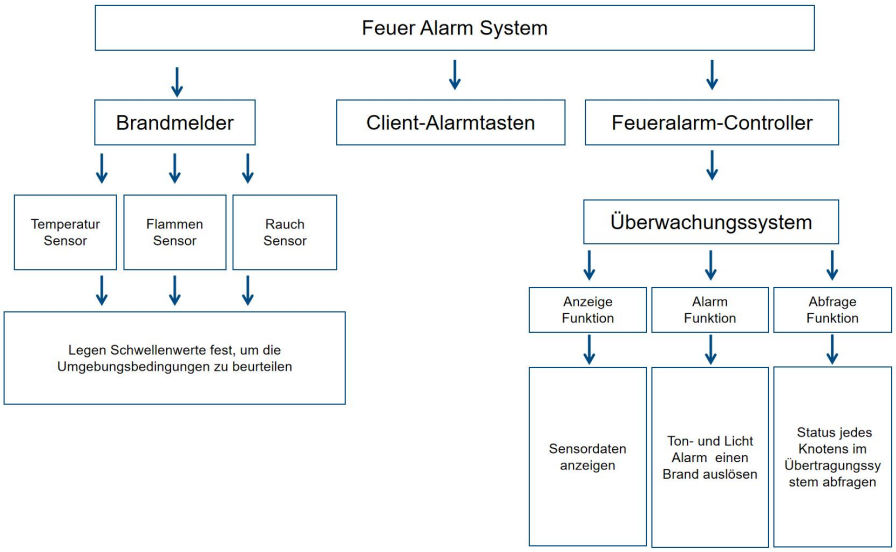
Beim Vergleich der drei Netzwerktopologien hat die Mesh-Topologie die deutlichsten Vorteile. Das Ziel dieses Projekts ist es, ein Mesh-Netzwerk aufzubauen. Die relevanten Inhalte des Netzwerkbaus werden im Beitrag ausführlich vorgestellt.

3 Gesamtkonzept

Bei diesem System handelt es sich um ein drahtloses Brandmeldesystem, das auf der ZigBee-Technologie basiert. Das System ändert die Busstruktur des herkömmlichen automatischen Brandmeldesystems und nutzt das ZigBee-Clusterstrukturnetzwerk für die Datenübertragung. Es löst eine Reihe von Problemen, wie z.B. die schwierige Verkabelung des herkömmlichen automatischen Brandmeldesystems, die komplizierte Wartung in der späteren Phase, die hohe Fehlalarmrate und die Unanwendbarkeit in einigen besonderen Fällen. Gleichzeitig ist das Systemnetzwerk mit entsprechender Host-Computersoftware ausgestattet und sein humanisiertes Design erleichtert den Bedienern die Überwachung des gesamten Netzwerks.

3.1 Funktionen des Feuer Alarm System

Die Hauptaufgabe des Systems besteht darin, die Temperatur und den Rauch in der Umgebung zu überwachen. Solange der Knoten einen bestimmten Indexwert misst, der den Schwellenwert überschreitet, wird ein Alarm ausgelöst.Darüber hinaus ist jedes Terminal des Netzwerks mit einem unabhängigen Alarmsystem ausgestattet. Wenn ein Brand auftritt und der Umgebungswert der Geräteerkennung den vom System festgelegten Alarmwert erreicht, wird der Überwachungsraum alarmiert. Nach Erhalt der Alarminformationen wird der Der Host-Computer sendet einen Alarm und zeigt den Alarmort an, um den Bediener daran zu erinnern. Im Notfall können über die am Host-Computer eingestellte Notevakuierungstaste Anweisungen an alle Knoten im Netzwerk gesendet werden. Wenn das Terminal einen Notfall-Evakuierungsbefehl erhält, löst es einen akustischen und visuellen Alarm aus, um die Menschen im Gebäude daran zu erinnern, dringend zu evakuieren.

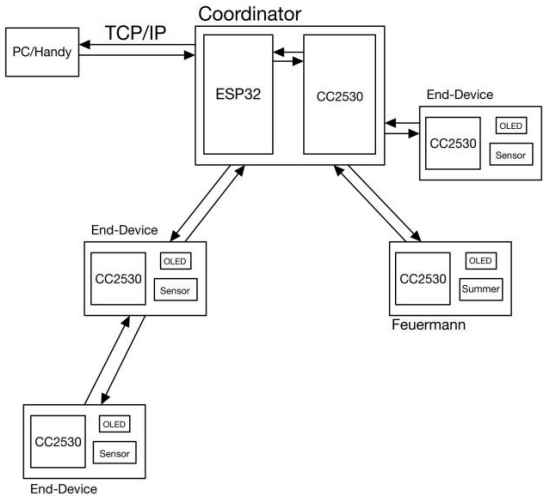


12Funktionen des Feuer Alarm System

3.2 Konzept

Ein vollständiges ZigBee-System benötigt einen Koordinator, einen oder mehrere Router und viele Endknoten, um den Netzwerkaufbau, die Pfadverteilung sowie die

Datenerfassungs- und -übertragungsaufgaben abzuschließen.Je nach den Anforderungen der tatsächlichen Vernetzung kann das ZigBee-Netzwerk verschiedene topologische Strukturen wie Sternnetzwerke und Mesh-Netzwerke bilden.



Das auf Zigbee basierende Brandüberwachungssystem besteht aus einem Mesh-Netzwerk. Da es sich um ein experimentelles Produkt handelt, habe ich die Struktur des Netzwerks vereinfacht und das gesamte System besteht aus vier Teilen. Dies sind: Endknotenteil (Feuerwehrpunkt), Routingknotenteil (Sensorpunkt), Koordinatorteil und Hostcomputerteil. Das Netzwerk besteht aus einem voll funktionsfähigen Koordinator (FFD) mit einem Gateway, mehreren Routing-Knoten mit Temperatur-, Rauch- und Flammensensoren sowie einem Endknoten mit Sensoren und vollständiger Informationsanzeige, um Eins-zu-Viele und Viele-zu zu erreichen -eins, eins, eins, eins, eine Übertragung. Der Koordinator ist über die serielle Schnittstelle mit dem Gateway verbunden, die Sensorknoten werden an verschiedenen Positionen des Umgebungsüberwachungsbereichs platziert, die Parameter der Umgebung werden über die darauf befindlichen Sensoren überwacht und die überwachten Umgebungsdaten werden über diesen an den Koordinator gesendet Im ZigBee-Netzwerk kann der Host-Computer aufgrund des Koordinators und der Gateway-Verbindung zu diesem Zeitpunkt die Überwachungsergebnisse der Umgebung über das drahtlose Netzwerk empfangen und anzeigen und die Überwachung der Knotentemperatur und des Rauchs realisieren

3.3 Geräte

Ein Zigbee-Gerät besteht aus folgenden Teilen:

1. Drahtloser Transceiver: Transceiver werden verwendet, um Daten von Zigbee-Netzwerken zu empfangen und an andere Netzwerke zu senden.

2. Prozessor: Der Prozessor ist dafür verantwortlich, die empfangenen Daten zu verarbeiten und in das entsprechende Format zu konvertieren.

3. Sensoren und Aktoren: Sensoren und Aktoren werden verwendet, um Daten zu sammeln und Geräte zu steuern.

4. Netzwerkschnittstelle: Eine Netzwerkschnittstelle wird verwendet, um Daten an andere Netzwerke zu senden.

5. Energieverwaltung: Die Energieverwaltung wird verwendet, um die Stromversorgung des Geräts aufrechtzuerhalten.

6. Benutzerschnittstelle: Die Benutzerschnittstelle wird verwendet, um Informationen und Steuerung für den Benutzer bereitzustellen.

7. Zigbee-Protokollstack: Der Zigbee-Protokollstack stellt die notwendige Software für das Gateway-Gerät dar. Es enthält den Implementierungscode des Zigbee-Protokolls und ist für die Datenübertragung zwischen dem Gateway-Gerät und dem Zigbee-Netzwerk verantwortlich.

3.4 Gateway-Gerät

Ein Zigbee-Gateway-Gerät ist ein drahtloses Netzwerkgerät, das zum Aufbau eines Zigbee-Netzwerks und zum Verbinden des Zigbee-Netzwerks mit anderen Arten von Netzwerken wie Wi-Fi oder Ethernet verwendet wird. Es kann Daten von Geräten in Zigbee-Netzwerk sammeln und an Cloud-Server oder andere Netzwerkgeräte weiterleiten.

Koordinatorknoten verfügen über Rechenleistung und Kommunikationsfähigkeiten. Die Hauptfunktion besteht darin, jeden Unterknoten im drahtlosen ZigBee-Netzwerk zu verwalten, die vom Überwachungszentrum des Host-Computers ausgegebenen Überwachungsinformationen an die Unterknoten zu senden, die über das ZigBee-Netzwerk überwacht und gesteuert werden müssen, und gleichzeitig zu empfangen den Status, die Datenerfassung und andere von jedem Unterknoten gesendete Informationen. Und über die serielle Schnittstelle zum Gateway hochladen, um die Daten zu verarbeiten und zu speichern. Wenn der Benutzer eine Kommunikationsanforderung über das Überwachungssystem des Host-Computers sendet, sendet der Einzelchip-Mikrocomputer gültige Daten über die serielle Schnittstelle an den Host-Computer, und der Bediener kann die Datenerfassung, Diagrammzeichnung, Datenstatistik und Datenanalyse über den Host abschließen Computer.

Das Gerät muss beim Design die folgenden wichtigen Faktoren berücksichtigen:

1.Kommunikationsprotokoll: Bestimmen das vom Gateway-Gerät unterstützte Kommunikationsprotokoll, z. B. Zigbee usw.

2.Konnektivität: Bestimmen die Konnektivität des Gateway-Geräts, z. B. unterstützte Schnittstellentypen wie Ethernet, USB, seriell usw.

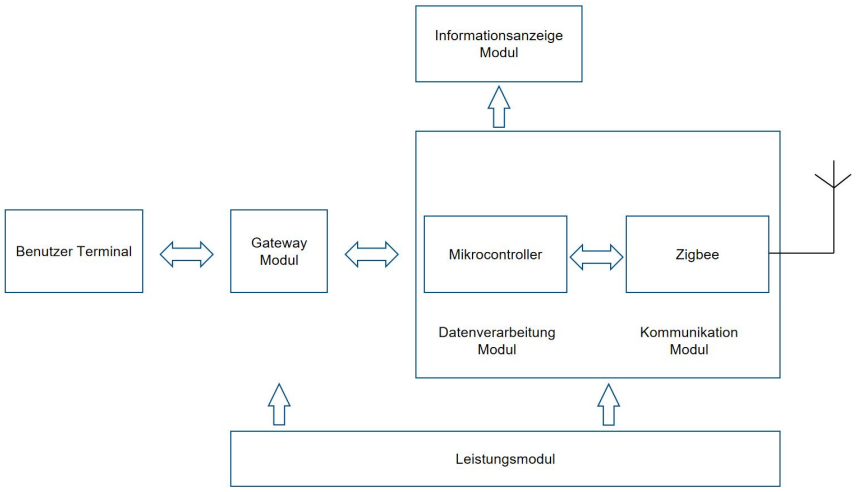
3.Datenverarbeitungsfähigkeit: Bestimmen die Datenverarbeitungsfähigkeit des Gateway-Geräts, wie z. B. CPU, Arbeitsspeicher, Speicherplatz usw.

4.Sicherheit: Bestimmen die Sicherheit des Gateway-Geräts, z. B. unterstützte Sicherheitsprotokolle, Firewalls, Verschlüsselung usw.

5. Stromversorgung: Bestimmen den Stromversorgungstyp des Gateway-Geräts, z. B. AC/DC, USB usw.

6.Volumengröße: Bestimmen die Volumengröße des Gateway-Geräts für die Installation in praktischen Anwendungen.

7.Zuverlässigkeit: Bestimmen die Zuverlässigkeit der Gateway-Ausrüstung, um sicherzustellen, dass sie bei langfristiger Verwendung normal funktioniert.



13Die Struktur des Koordinatorknotens

3.4.1 Datenübertragung

Zigbee-Gateway-Geräte verwenden typischerweise ein serielles Kommunikationsprotokoll wie UART, um Daten an einen anderen Netzwerktyp wie Wi-Fi oder Ethernet zu übertragen. Wenn ein Zigbee-Gateway Daten von einem Gerät in einem Zigbee-Netzwerk empfängt, analysiert und formatiert es sie in das entsprechende Format. Anschließend sendet es die Daten über ein geeignetes Protokoll wie TCP/IP an einen Cloud-Server oder ein anderes Netzwerkgerät. Wenn das Gateway umgekehrt Daten von Cloud-Servern oder anderen Netzwerkgeräten empfängt, konvertiert es diese in das Zigbee-Protokoll und sendet sie an das Zigbee-Netzwerk.

Kurz gesagt, das Zigbee-Gateway-Gerät ist eine Brücke, die zwei verschiedene Netzwerke verbindet. Es kann Daten analysieren und formatieren, sodass Daten aus verschiedenen Netzwerken miteinander übertragen werden können.

3.4.2 Eigenschaft

Gemäß den Designanforderungen des Gateway-Geräts enthält das von mir entworfene Gateway-Gerät die folgende Hardware:

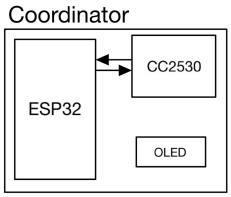
1.ESP32: Wird verwendet, um Daten aus dem Zigbee-Netzwerk zu empfangen und in ein geeignetes Format zu konvertieren und an andere Netzwerke zu senden.

2.TI cc2530: Als Koordinator zum Erstellen des gesamten ZigBee-Netzwerks Daten von ZigBee-Endgeräten empfangen.

3.OLED:Zur Anzeige der Netzwerkparameter und der Anzahl der Endgeräteverbindungen

4.LED:Daten Empfang Anzeigen

5.Batterie: Stromversorgung bereitstellen.



14 Gateway

3.5 Endgeräte

Bei der Gestaltung von Endgeräten sind folgende wichtige Faktoren zu berücksichtigen:

1. Funktionalität: Bestimmen die Funktionalität des Endgeräts, wie Steuerung, Überwachung, Datenerfassung usw.

2. Interkonnektivität: Bestimmen die Interkonnektivität von Endgeräten, z. B. unterstützte Kommunikationsprotokolle, Verbindungsmethoden usw.

3. Benutzeroberfläche: Bestimmen die Benutzeroberfläche des Endgeräts, wie z. B. Bildschirm, Schaltflächen usw.

4. Stromtyp: Bestimmen den Stromtyp des Endgeräts, z. B. AC/DC, USB usw.

5. Größe und Gewicht: Bestimmen die Größe und das Gewicht der Endgeräte, um eine einfache Installation und Wartung in der Anwendung zu gewährleisten.

6. Elektrische Schnittstelle: Bestimmen die Art und Spezifikation der elektrischen Schnittstelle des Endgeräts, wie Stecker, Klemmen usw.

7. Haltbarkeit: Bestimmen die Haltbarkeit des Endgeräts, um sicherzustellen, dass es bei langfristiger Verwendung normal funktioniert.

Basierend auf diesen Faktoren habe ich einige Ideen entwickelt. Die Endgeräte werden hier in zwei Kategorien eingeteilt. Ein Typ ist die Ausrüstung, die in dem Brandort platziert wird, um Brandortinformationen zu sammeln. Die andere ist Ausrüstung, die von Feuermannn getragen wird, um den Alarm auszulösen.

3.5.1 Datenübertragung

Zigbee-Endgeräte können bei der Datenübertragung zwei Rollen spielen, eine ist der Datensender und die andere der Datensender.

Wenn das Zigbee-Endgerät als Datensender fungiert, nachdem das Endgerät Sensordaten gesammelt hat, verwendet es das Zigbee-Protokoll, um die Daten an das Netzwerk zu senden. Endgeräte senden Daten, indem sie Informationsrahmen senden. Informationsrahmen enthält Daten und Zieladresse. Endgeräte können Daten an Gateways oder andere Endgeräte senden.

Endgeräte können auch andere Endgeräte steuern, indem sie Steuertelegramme senden. Der Steuerrahmen enthält die Zieladresse und Steueranweisungen, und das Endgerät kann den Status anderer Endgeräte durch den Steuerrahmen steuern.

Wenn ein Zigbee-Endgerät als Brück fungiert, kann das Endgerät auch Daten von anderen Endgeräten oder Gateways empfangen. Wenn das Endgerät die Daten empfängt, analysiert es die Daten mithilfe des Zigbee-Protokolls und speichert sie in seinem Speicher und sendet die Daten dann an das Gateway. Zu diesem Zeitpunkt ist das Endgerät eine Brücke, die das Gateway und andere Endgeräte verbindet, wodurch die Entfernung zwischen dem Endgerät und dem Gateway erweitert und sichergestellt werden kann, dass die vom Endgerät gesendeten Daten so weit wie möglich an das Gateway gesendet werden .

3.5.2 Eigenschaft

Gemäß den Designanforderungen des Endgeräts umfasst das von mir entworfene Endgerät die folgende Hardware:

1.TI cc2530:Als Endknoten senden Geräte Sensordaten an das Gateway.

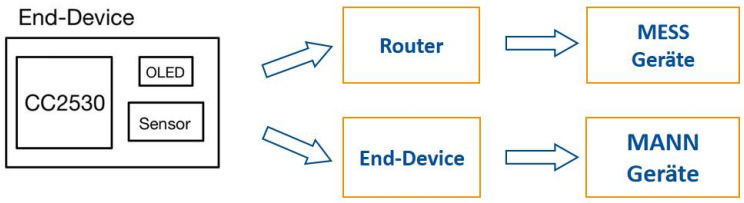
2.PT100-Temperatursensor: Erfassen Sie die Umgebungstemperatur.

3.Rauchsensor MQ-2: erkennt die Konzentration von Umgebungsrauch

4.OLED: Wird verwendet, um Netzwerkparameter anzuzeigen

5.Summer: Sendet es einen Alarm, wenn die Arbeitsumgebung des Geräts einen gefährlichen Zustand erreicht

6. Batterie: Stromversorgung bereitstellen.



15Endgeräte

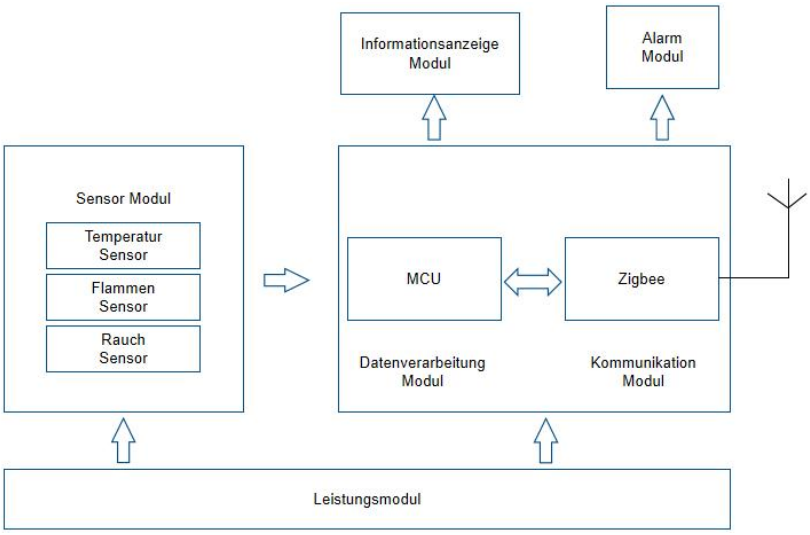
Es befindet sich derzeit in der Labordesignphase, daher ist geplant, ein System mit nicht weniger als 3 Endgeräten zu entwerfen.

Das Gerät überträgt alle 10 Sekunden Daten und verfügt über eine Trennungs-Neustartfunktion. Die Endgeräte garantieren eine effektive Übertragungsdistanz von mindestens 50 Metern.

3.5.3 MESS-Knoten

Wenn ein Stoff verbrennt, gibt er zwangsläufig eine gewisse Wärmemenge ab. Darüber hinaus steigt auch die Temperatur der Umgebung bis zu einem gewissen Grad an. Bei der ersten Verbrennung eines Stoffes werden nach und nach brennbare Gase wie Kohlenmonoxid, Wasserstoff und Methan freigesetzt. Darüber hinaus wird auch eine geringe Menge an Schwebeteilchen erzeugt.Ich wähle aus Sicht der Einsatzumgebung Temperatursensoren, Flammensensoren und Rauchsensoren aus, um Daten zu sammeln.

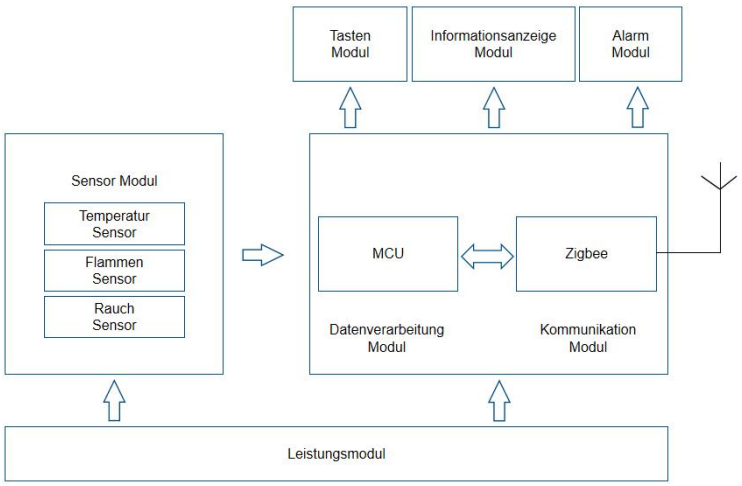
Der Endknoten besteht aus einem Leistungsmodul, einem Temperatursensor, einem Flammensensor, einem Rauchsensor, einem Zigbee-Chip, einem Bildschirm und einer Kontrollleuchte.Sie realisieren die Datenerfassung im Überwachungsbereich. Die gesammelten Signale werden von der MCU verarbeitet, die Daten werden integriert und verpackt und drahtlos übertragen. Schließlich werden die Daten an den Koordinator aggregiert. Gleichzeitig löst das Gerät im Brandfall einen Alarm aus.



16Struktur von MESS-Knoten

3.5.4 MANN Knoten

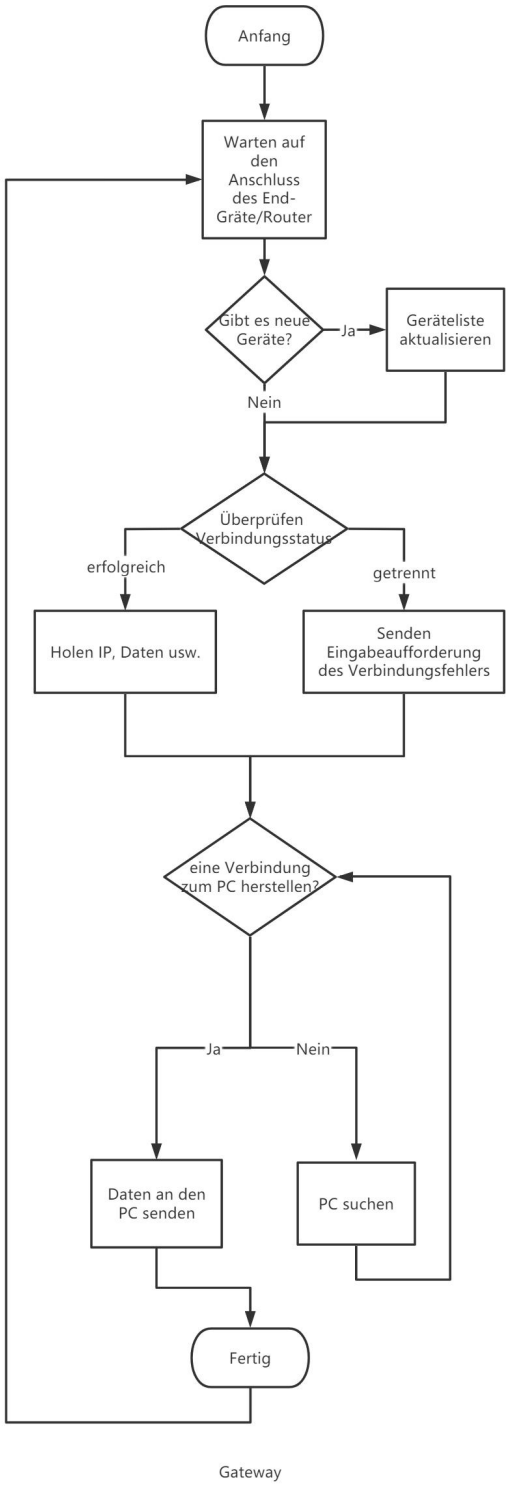
Der Feuerwehrknoten verfügt über alle Funktionen des Sensorknotens. Auf der Grundlage des Sensorknotens kann der Bediener im Überwachungsraum im Notfall, wenn ein Notfall vorliegt und Personen evakuiert werden müssen, über den Hostcomputer auch Befehle an den Endknoten senden. Z.B. ein Summeralarm, der Menschen an die Evakuierung erinnert. Darüber hinaus verfügt der Feuerwehrknoten über eine Reihe von Tastensteuerungsmodulen zur Steuerung der Ausrüstung.



17Struktur von MANN Knoten

3.6 Flussdiagramm

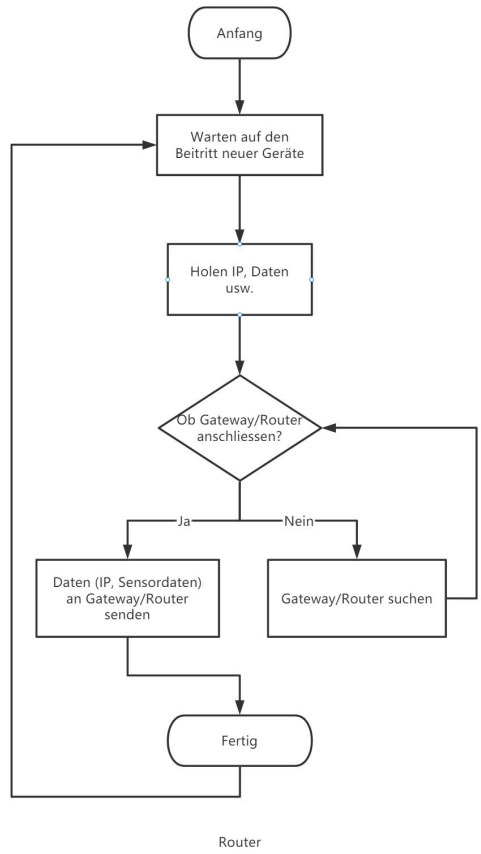
Gateway(Koordinator)



18 Flussdiagramm des Gateway*(Koordinator)*

Die Aufgabe des Gateways besteht darin, das Netzwerk aufzubauen. Erstes Einschalten zur Initialisierung. Warten Sie dann, bis der End-Gerät oder Router eine Verbindung zum Gateway hergestellt hat. Durch den Vergleich von Datenbankdaten, um festzustellen, ob neue Geräte hinzugefügt wurden. Wenn ein neues Gerät hinzugefügt wird, aktualisieren Sie die Datenbankdaten. Wenn im Gegensatz dazu kein neues Gerät hinzukommt, wird der Verbindungsstatus der vorhandenen Geräte im Netzwerk beurteilt. Wenn das Gerät erfolgreich verbunden ist, erhalten Sie die IP-Adresse, Sensordaten und andere Informationen. Wenn die Geräteverbindung fehlschlägt, wird eine Aufforderungsmeldung ausgegeben, um die Feuermann daran zu erinnern, dass die Geräteverbindung anormal ist. Der nächste Schritt besteht darin, zu beurteilen, ob eine Verbindung zum Computer hergestellt werden soll. Wenn die Verbindung erfolgreich ist, werden die gesammelten Daten an den Computer übertragen. Andernfalls versucht die Verbindung weiterhin, eine Verbindung zum Computer herzustellen.

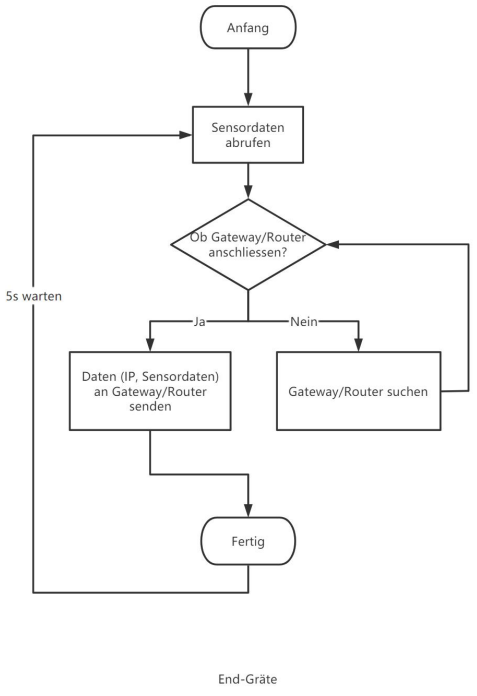
Router(MESS-Knoten)



Die Aufgabe des Routers besteht darin, Informationen weiterzugeben. Erstes Einschalten zur Initialisierung. Warten Sie dann, bis der End-Gerät eine Verbindung zum Router hergestellt hat. Durch den Vergleich von Datenbankdaten, um festzustellen, ob neue Geräte hinzugefügt wurden. Wenn das Gerät erfolgreich verbunden ist, erhalten Sie die IP-Adresse, Sensordaten und andere Informationen. Der nächste Schritt besteht darin, zu beurteilen, ob der Router mit dem Gateway oder anderen Routern verbunden ist. Wenn die Verbindung erfolgreich ist, werden die gesammelten Daten an das Gateway oder andere Router übertragen. Andernfalls wird die Verbindung weiterhin versuchen, eine Verbindung zum Gateway herzustellen oder andere Router.

19 Flussdiagramm des Router(MESS-Knoten)

End-Gräte(MANN-Knoten)



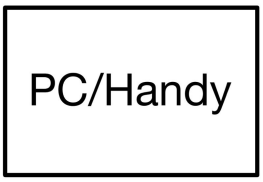
Die Aufgabe des End-Geräts ist es, Sensordaten zu sammeln. Erstes Einschalten zur Initialisierung. Anschließend werden Sensordaten gesammelt. Im nächsten Schritt muss festgestellt werden, ob das End-Gerät mit dem Router verbunden ist. Wenn das Gerät erfolgreich verbunden ist, sendet es Informationen wie IP-Adresse und Sensordaten. Andernfalls, wenn die Verbindung fehlschlägt, wird weiterhin versucht, eine Verbindung zum Router herzustellen. Daten werden alle fünf Sekunden gesendet.

20 Flussdiagramm des End-Gräte(MANN-Knoten)

3.7 Client-System

Das Client-System verarbeitet die vom Gateway gesendete Zeichenfolge und extrahiert die Sensordaten. Wenn der Wert den eingestellten Schwellenwert überschreitet, blinkt die Alarmanzeigeleuchte auf der Clientseite, um anzuzeigen, dass ein Feuer vorliegt, und gleichzeitig Der Bediener drückt die Notevakuierungstaste des Client-System, um den Ton- und Lichtalarm des Endknotens zu steuern und den Temperaturwert und die Rauchinformationen zu jedem Zeitpunkt zur späteren Anzeige zu speichern

Da es meine Aufgabe ist, für den Aufbau des ZigBee-Übertragungssystems verantwortlich zu sein, besteht der Entwurf des Client-Systems nur darin, zu testen, ob das Übertragungssystem normal läuft. Deshalb habe ich eine einfache Version des Client-Systems (mobile APP) erstellt kann alle meine Designideen umsetzen, auch wenn diese bei Target realisiert werden.



3.8 Kapitelzusammenfassung

Dieses Kapitel fasst die Gesamtentwurfsidee des Systems zusammen und vereinfacht den Entwurf. Das System besteht hauptsächlich aus vier Teilen: MANN Knoten (Feuermannknoten),MESS-Knoten (Sensorknoten), Koordinatorteil und Client-System. Der Sensor schließt die Datenerfassung ab und sendet sie drahtlos an den Koordinator. Der Koordinator stellt über das Gateway eine Verbindung mit dem Client-System her, um die Überwachung der Sensorknoten abzuschließen.

4 Hardware

4.1 Zigbee

TI CC2530

21 Zigbee CC2530(C)

Der CC2530 von TI ist ein drahtloser Mikrocontroller (MCU), der für drahtlose Anwendungen mit geringem Stromverbrauch ausgelegt ist. Es basiert auf den Industriestandard-Protokollen Zigbee und IEEE 802.15.4, wodurch es mit einer Vielzahl von drahtlosen Geräten und Netzwerken kompatibel ist. Der CC2530 basiert auf der Low-Power-Mikrocontroller-Architektur MSP430 von Texas Instruments und ist mit einem 2,4-GHz-Funktransceiver ausgestattet, der Datenraten von bis zu 250 kbps unterstützt. Es verfügt außerdem über eine Vielzahl von Peripherieschnittstellen, darunter UART, SPI, I2C und ADC, wodurch es für eine Vielzahl von Anwendungen geeignet ist, z. B. Heimautomatisierung, industrielle Steuerung und medizinische Geräte.

Name	*TI CC2530F256*
RAM	*8KB*
Flash-Speicher	*256KB*
Betriebstemperaturbereich	*-40 - 125*
Betriebsspannung	*2V bis 3.6V*
Strom	• *Active-Mode : 24 mA* • *Mode 1 (4 µs Wake-Up): 0.2 mA* • *Mode 2 (Sleep Timer Running): 1 µA* • *Mode 3 (External Interrupts): 0.4 µA*
Übertragungsabstand	*50m(ohne Hindernis)*
Vorteil	*2,4-GHz-IEEE 802.15.4-konformer HF-Transceiver*
	Hervorragende Empfängerempfindlichkeit und Robustheit gegenüber Störungen
	Programmierbare Ausgangsleistung bis zu 4,5 dBm

2Zigbee CC2530

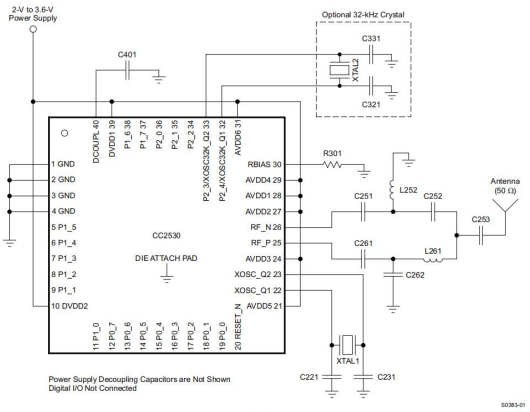


Figure 19. CC2530 Application Circuit

Table 3. Overview of External Components (Excluding Supply Decoupling Capacitors)

Component	Description	Value
C251	Part of the RF matching network	18 pF
C261	Part of the RF matching network	18 pF
L252	Part of the RF matching network	2 nH
L261	Part of the RF matching network	2 nH
C262	Part of the RF matching network	1 pF
C252	Part of the RF matching network	1 pF
C253	Part of the RF matching network	2.2 pF
C331	32kHz xtal loading capacitor	15 pF
C321	32kHz xtal loading capacitor	15 pF
C231	32MHz xtal loading capacitor	27 pF
C221	32MHz xtal loading capacitor	27 pF
C401	Decoupling capacitor for the internal digital regulator	1 µF

22 Zigbee CC2530 Application(C)

4.2 Gateway

ESP32

23 ESP32(C)

Der ESP32 ist ein kostengünstiger System-on-a-Chip (SoC)-Mikrocontroller mit geringem Stromverbrauch und integrierten Wi-Fi- und Bluetooth-Funktionen. Es wird von Espressif Systems entwickelt und basiert auf dem ESP8266 SoC des Unternehmens, das für Internet of Things (IoT)-Anwendungen beliebt ist.

Der ESP32 verfügt über zwei CPU-Kerne, die unabhängig voneinander gesteuert und geplant werden können. Es hat auch ein eingebautes Bluetooth 4.2 und Wi-Fi 802.11 b/g/n/e/i (2,4 GHz und 5 GHz) Funk. Das Wi-Fi kann eine maximale Datenrate von 150 Mbit / s unterstützen, und Bluetooth unterstützt klassisches Bluetooth und Bluetooth Low Energy (BLE).

Weitere Merkmale des ESP32 sind eine breite Palette von Peripherieschnittstellen, darunter UART, SPI, I2C, I2S, PWM, ADC, DAC und mehr. Es enthält auch einen eingebauten Hallsensor, Temperatursensor und Berührungssensor. Mit seinem geringen Stromverbrauch und der breiten Palette an Funktionen eignet sich der ESP32

gut für eine Vielzahl von IoT-Anwendungen, wie z. B. Heimautomatisierung, tragbare Geräte und industrielle Steuerungssysteme.

Der ESP32 unterstützt auch viele Programmiersprachen, einschließlich C, C++ und Python, mit einer großen Anzahl von Bibliotheken, was eine einfache Entwicklung und Portabilität auf verschiedenen Systemen ermöglicht.

Name	*ESPRESSIF ESP32S*
Größe	*56 x 28 x 13 mm*
RAM	*512 kB*
Flash-Speicher	*4 MB*
Taktfrequenzbereich	*80 MHz / 240 MHz*
Stromversorgunsspannung (USB)	*5 V*
Benötigter Betriebsstrom	*min. 500 mA*
Schnittstellen	*SPI, I2C, I2S, CAN, UART*
Wi-Fi Protokolle	*802.11 b/g/n (802.11n bis zu 150 Mbps)*
Wi-Fi Frequenz	*2.4 GHz*
Bluetooth	*V4.2 - BLE und Classic Bluetooth*

3ESP32

4.3 Sensor

Bei der Auswahl eines Sensors sind mehrere wichtige Faktoren zu berücksichtigen:

1 Sensorleistung: Bestimmen die Leistungsparameter des Sensors wie Empfindlichkeit, Linearität, Wiederholbarkeit und Stabilität.

2 Arbeitsumgebung: Bestimmen die Arbeitstemperatur, Feuchtigkeit, Druck, Licht und andere Umgebungsbedingungen des Sensors.

3 Signalausgang: Bestimmen den Signalausgangstyp und die Spezifikation des Sensors, wie z. B. Spannung, Strom, Frequenz usw.

4 Elektrische Schnittstelle: Bestimmen Art und Spezifikation der elektrischen Schnittstelle des Sensors, wie Stecker, Klemmen usw.

5 Größe und Gewicht: Bestimmen die Größe und das Gewicht des Sensors, um eine einfache Installation und Wartung in der Anwendung zu gewährleisten.

6 Haltbarkeit: Bestimmen die Haltbarkeit des Sensors, um sicherzustellen, dass er im Langzeitgebrauch normal funktioniert.

7 Kosten: Bestimmen die Produktionskosten des Sensors, um beim Verkauf einen angemessenen Gewinn zu erzielen.

4.3.1 PT100

24 PT100(C)

Ein PT100-Temperatursensor ist eine Art Widerstandstemperaturdetektor (RTD), der zur Temperaturmessung verwendet wird. Es besteht aus einem Platindraht, der bei 0 Grad Celsius (32 Grad Fahrenheit) einen Widerstand von 100 Ohm hat. Der Widerstand des PT100 ändert sich mit der Temperatur, sodass die Temperatur durch Messen des Widerstands des Drahts gemessen werden kann.

Um einen PT100-Temperatursensor zu verwenden, benötige ich eine Schaltung, die den Widerstand des Sensors messen und in einen Temperaturmesswert umwandeln kann. Hier eine Übersicht über die Schritte zur Verwendung eines PT100-Temperatursensors:

1.Schließen den PT100-Sensor an Stromkreis an: Der PT100-Sensor hat normalerweise drei Drähte, einschließlich einer positiven Leitung (rot oder gelb), einer negativen Leitung (schwarz oder blau) und einer Signalleitung (weiß). Diese Drähte sind normalerweise mit einem Schaltkreis verbunden, der den Widerstand des PT100 messen kann.

2.Messen den Widerstand des PT100: Der Widerstand des PT100 wird normalerweise mit einer Wheatstone-Brückenschaltung oder einem Messgerät wie einem Multimeter gemessen. Dies kann mithilfe der Standard-RTD-Gleichung in Temperatur umgewandelt werden.

3.Konvertieren die Widerstandsmessung in einen Temperaturmesswert: Der Widerstand des PT100 ändert sich linear mit der Temperatur, sodass ich eine Linearisierungsgleichung verwenden kann, um den gemessenen Widerstand in einen Temperaturmesswert umzuwandeln. Die gebräuchlichste Gleichung verwendet die Callendar-Van Dusen-Gleichung.

Die Temperaturmessung erfolgt, ich kann einen Mikrocontroller oder Mikroprozessor verwenden, um die Sensorausgabe zu lesen und die Daten zu verarbeiten, um die Temperaturmesswerte zu erhalten.Ich kann die Daten auch zur späteren Analyse protokollieren oder zu Überwachungs- oder Kontrollzwecken an ein entferntes Gerät oder einen Server senden.

Es ist wichtig zu beachten, dass PT100 auch verschiedene Genauigkeitsklassen haben, Klasse A, Klasse B und Klasse B, normalerweise hat Klasse A die beste Genauigkeit, aber die teuerste.

Name	*PT100*
Typ	*wasserdichter Wärmewiderstand aus Platin*
Größe	*4mm x 30mm*
Gewicht	*3.7g*
Äußere Abschirmung	*isolierte Abschirmung*
Temperaturbereich	*-20~250 ℃*
Vorhandene Pins	*GND,VCC,A0(input)*

4 PT100

4.3.2 MQ-2

25 MQ-2(C)

Der Rauchsensor MQ-2 ist ein häufig verwendeter Rauchmelder. Es verwendet einen Sensor, der als Gassensor bezeichnet wird, um Rauch in der Luft zu erkennen. Wenn die Rauchkonzentration ein bestimmtes Niveau erreicht, sendet der Sensor ein Signal aus, das den Benutzer auf die Brandgefahr aufmerksam macht. Der Rauchsensor MQ-2 ist sehr empfindlich und kann Rauch von den meisten Verbrennungsprodukten erkennen. Solche Sensoren werden üblicherweise in der persönlichen Heimsicherheit, Feuerüberwachungssystemen und der industriellen Prozesssteuerung verwendet.

Name	*QT-MQ-2*
Größe	*32mm×20mm×22mm*
Gewicht	*7.4g*
Betriebsspannung(V)	*DC 5V*
Strom	*150mA*
Gasart	*Rauch, Flüssiggas, Erdgas und Propangas usw.*
Konzentrationsbereich	*300~10000ppm*
Vorhandene Pins	*VCC,GND,A0(output)*

5 MQ-2

4.4 Andere Hardware

4.4.1 OLED

26 SSD1306(C)

Name	*SSD1306*
Auflösung	*128 x 64 Pixel*
Größe	*27mm x 27mm x 4,1mm*
Sichtbarer Winkel	*> 160°*
Betriebsspannung	*3.3V bis 5V*
Gewicht	*20 g*
Betriebstemperatur	*-30°C bis +80°C*
Niedriger Stromverbrauch	*0,04W im normalen Betrieb*
Vorhandene Pins	*VCC, GND, SCL, SDA*

6 SSD1306

4.4.2 Summer

27 Summer(C)

Name	*Elektronischer Summer*
Durchmesser	*12 mm*
Schalldruckpegel	*85 dB*
Betriebsspannung	*3V bis12V*
Gewicht	*5 g*
Vorhandene Pins	*GND,A0*

7 Summer

5 Hardware Test

5.1 Vorbereitung

Bevor ich mit dem Hardwaretest fortfuhr, habe ich über die folgenden Vorbereitungen nachgedacht:

1. Definieren die Testziele:Ich mache mich klar über die Testziele und die Ergebnisse, die ich während des Tests erreichen möchte.

2. Bestimmen die Testumgebung: Ich bestimme die geeignete Umgebung zum Testen der Hardware. Dazu gehören bestimmte Anforderungen an Temperatur, Luftfeuchtigkeit usw.

3. Sicherung der Testausrüstung: Ich stelle vor dem Testen sicher, dass ich über die erforderlichen Testausrüstungen und Werkzeuge verfüge. Dazu können Prüfgeräte, Sensoren, Oszilloskope, Netzteile usw. gehören. Entsprechend den Eigenschaften der Hardware wähle ich die passende Ausstattung aus.

4. Testplan vorbereiten: Ich erstelle einen detaillierten Testplan mit Testschritten, Testmethoden und erwarteten Ergebnissen und ich stelle sicher, dass alle erforderlichen Testaspekte abgedeckt sind.

5. Erstellen eine Testumgebung:Ich richte eine Testumgebung ein, einschließlich der Verbindung und Konfiguration von Testgeräten, um sicherzustellen, dass sie ordnungsgemäß funktionieren.Ich richte nach Bedarf eine Standardtestumgebung ein.

6.Testdaten vorbereiten:Ich bereite einen geeigneten Testdatensatz vor, um die Funktionalität und Leistung der Hardware zu überprüfen. Dabei werden geeignete Eingabedaten erstellt, verschiedene Anwendungsfälle simuliert und Ausgabedaten automatisch erfasst.

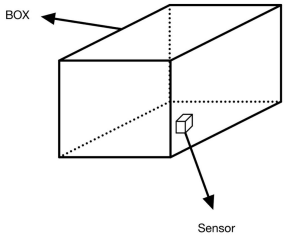
7.Erstellen einen Fehlerbehandlungsplan: Ich berücksichtige mögliche Probleme oder Fehler und formuliere Gegenmaßnahmen. Ich stelle sicher, dass ich weiß, wie ich mit möglichen Testfehlern oder Hardwaredefekten umgehen soll.

8.Ergebnisse aufzeichnen und analysieren: ich bin darauf vorbereitet, die Ergebnisse und beobachteten Probleme während des Tests aufzuzeichnen. Dies erleichtert die spätere Analyse und Verbesserung.

9.Testergebnisse auswerten und überprüfen:Ich bewerte entsprechend den Testzielen die Gültigkeit der Testergebnisse und überprüfe, ob die Hardware die erwarteten Spezifikationen und Leistungsanforderungen erfüllt.

5.1.1 Testumgebung

Um korrekte Sensordaten zu erhalten und durch Umgebungsveränderungen verursachte Datenanomalien zu reduzieren, plane ich die Schaffung einer Standardtestumgebung. Also habe ich einen Standardprüfstand aus einer Acrylplatte gebaut.



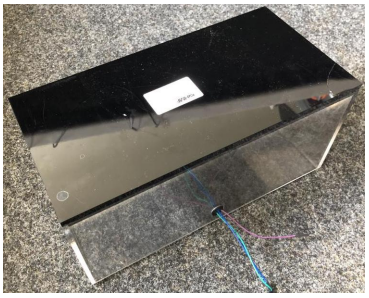
28stabile Testumgebung

Zunächst bereite ich eine Box geeigneter Größe vor, die entsprechend der Größe des Sensors und der Größe des Testobjekts ausgewählt wird. Ich habe dafür gesorgt, dass in der Box genügend Platz für den Sensor und das Testobjekt vorhanden ist, und habe etwas Platz für die Kabel und den Computer gelassen. Nach einiger Auswahl habe ich mich schließlich für eine transparente Box aus Acrylplatte entschieden, die trotz ihrer stabilen Hülle den gesamten Testprozess problemlos beobachten kann. Und das Acrylmaterial gab mir die Möglichkeit, Löcher in die Schachtel zu stanzen.

Ich wähle eine geeignete Stelle auf einer Seite des Kastens und stanze mit einem Bohrer geeigneter Größe Löcher in den Kasten. Anzahl und Lage dieser Löcher stimmen mit den Anforderungen des Sensors und der Anschlussdrähte überein und stellen sicher, dass die Löcher so platziert sind, dass der Sensor während des Tests problemlos mit dem Testobjekt verbunden werden kann.

29 Gehäuse mit Löcher

Nachdem die Löcher gemacht waren, habe ich die Sensoren durch diese Löcher in das Innere der Box eingeführt. Ich stelle sicher, dass der Sensor fest in der Box sitzt, um unerwünschte Bewegungen oder Lockerungen während des Tests zu vermeiden. Dann setze ich den schwarzen Deckel auf und stelle die Box in einen stabilen Stand, damit sie nicht wackelt.



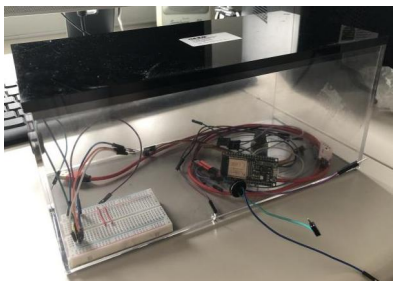
30 Gehäuse mit schwarzen Deckel

Als nächstes habe ich die Fugen der Box mit Klebeband abgedichtet. Ich verwende hochwertiges, breites Klebeband, um sicherzustellen, dass in den versiegelten Verbindungen keine Löcher entstehen. Dies trägt dazu bei, die Beeinträchtigung der Testumgebung durch externe Faktoren zu reduzieren.

31 abgedichtet

Ich habe den Rest des Kartons überprüft, um sicherzustellen, dass er keine Risse oder Löcher aufweist. Sollten Probleme festgestellt werden, können diese mit Klebeband oder einem anderen geeigneten Dichtungsmaterial repariert werden.

Zum Schluss stecke ich die Drähte vom Sensorende in das Loch in der Box und verbinde das andere Ende mit dem Computer oder dem Datenerfassungsgerät. Dadurch kann ich sicherstellen, dass das Kabel fest und sicher angeschlossen ist und eine zuverlässige Signalübertragung gewährleistet ist.

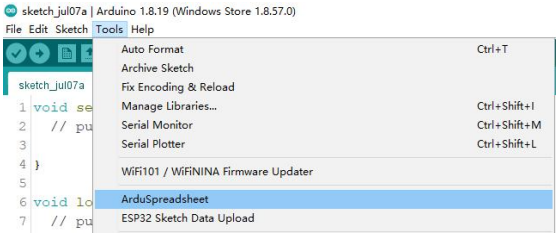


32 komplette Box

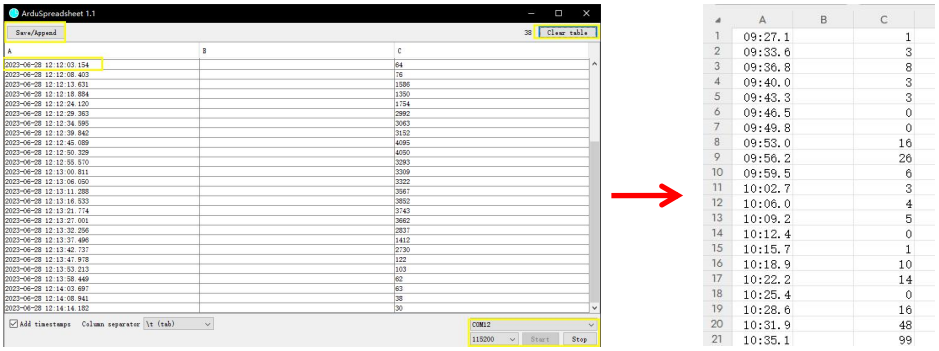
Nachdem ich die oben genannten Schritte ausgeführt hatte, gelang es mir, eine Box zu erstellen, die eine stabile Testumgebung bietet. Die Verwendung dieser Box kann die Beeinflussung des Sensors durch externe Faktoren reduzieren und eine genauere und zuverlässigere Testumgebung für Sensortests und Datenerfassung bieten.

5.1.2 Bequeme Datenaufzeichnungsmethode

Die herkömmliche Datenaufzeichnungsmethode besteht darin, sie einzeln mit einem Stift aufzuzeichnen, was sehr ineffizient und fehleranfällig bei der Ausgabe von Aufzeichnungen ist. Um die schnelle Aufzeichnung von Messdaten zu realisieren. Ich suchte nach einer Möglichkeit, Daten zu protokollieren. Durch den Vergleich der Schwierigkeit und Bequemlichkeit der Verwendung verschiedener Software habe ich mich schließlich für ein Plug-in namens **ArduSpreadsheet** entschieden, das die Arduino IDE unterstützt. Es kann direkt von Arduino aus gestartet werden, kann die Baudrate und die Nummer der seriellen Schnittstelle ändern, die Daten der seriellen Schnittstelle in Echtzeit anzeigen, verfügt über eine Funktion zum Speichern von Daten, kann die Daten direkt in einer Excel-Tabelle speichern und einen Zeitstempel hinzufügen um die Datenanalyse zu erleichtern.

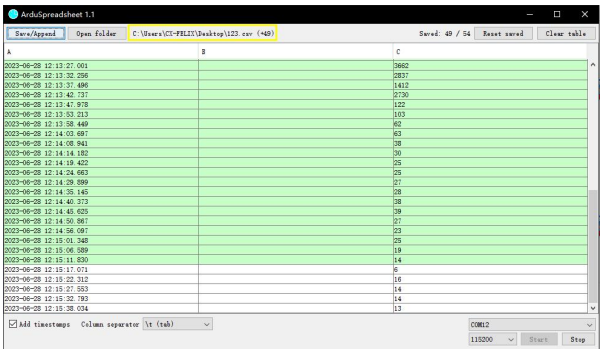


33 ArduSpreadsheet 1



34 ArduSpreadsheet 2

Nach dem Speichern der Daten wird die Speicheradresse der Datei angezeigt. Ich verwende unterschiedliche Hintergrundfarben, um gespeicherte und nicht gespeicherte Daten der seriellen Schnittstelle zu unterscheiden.

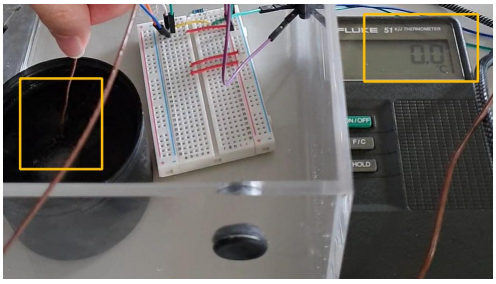


35 ArduSpreadsheet 3

5.1.3 Kalibrierung von Messgeräten

Digitales Thermometer

Gefrierpunktkalibrierung: Zuerst bereite ich einen Behälter voller Eiswürfel und Wasser vor und stelle sicher, dass sie sich gut vermischen und eine stabile Eis-Wasser-Mischung bilden. Anschließend tauchte ich die Sonde des Digitalthermometers in die Eis-Wasser-Mischung und wartete, bis sich die Temperatur stabilisierte. Ich notiere den Messwert des Digitalthermometers und vergleiche ihn mit der Standardtemperatur des Gefrierpunkts (0 Grad Celsius). Dazu stelle ich den Knopf unter dem Digitalthermometer ein, bis die richtige Gefriertemperatur angezeigt wird.



36 Kalibrierung Thermometer

Laser-Entfernungsmesser

Während des Experiments muss ich oft den Abstand zwischen den Versuchsobjekten messen. Manchmal ist der Abstand zu groß, um ein Maßband zu verwenden, deshalb habe ich einen Laser-Entfernungsmesser in das Experiment einbezogen. Durch den Vergleich der Messwerte des Laser-Entfernungsmessers mit den Messwerten des Maßbandes kann ich schlussfolgern, dass die Messwerte des Laser-Entfernungsmessers genau sind und zur Aufzeichnung experimenteller Daten verwendet werden können.

37 Kalibrierung Maßband

5.2 Aufgabenanalyse

Als ich den Sensor testete, hatte ich folgende Gedanken:

1.Testumgebung: Ich verstehe den Betriebsbereich und die Genauigkeitsanforderungen des Sensors und halte die Testumgebung während des Tests stabil.

2.Sicherheitsmaßnahmen: Ich stelle sicher, dass bei der Durchführung von Sensortests geeignete Sicherheitsmaßnahmen ergriffen werden, um Brände oder andere Gefahren zu verhindern. Dies kann die Belüftung der Testumgebung und die Vorbereitung von Feuerlöschgeräten umfassen.

3.Genauigkeitstest: Der Sensor wird mit einem bekannten Standardtestinstrument wie einem Standardthermometer auf Genauigkeit getestet. Ich vergleiche die Sensorwerte mit den tatsächlichen Temperaturwerten, um Genauigkeit und Abweichungen zu beurteilen.

4.Stabilitätstest: Ich führe einen Stabilitätstest des Sensors durch, um seine Leistung im Dauereinsatz oder bei wechselnden Umgebungsbedingungen zu beobachten.

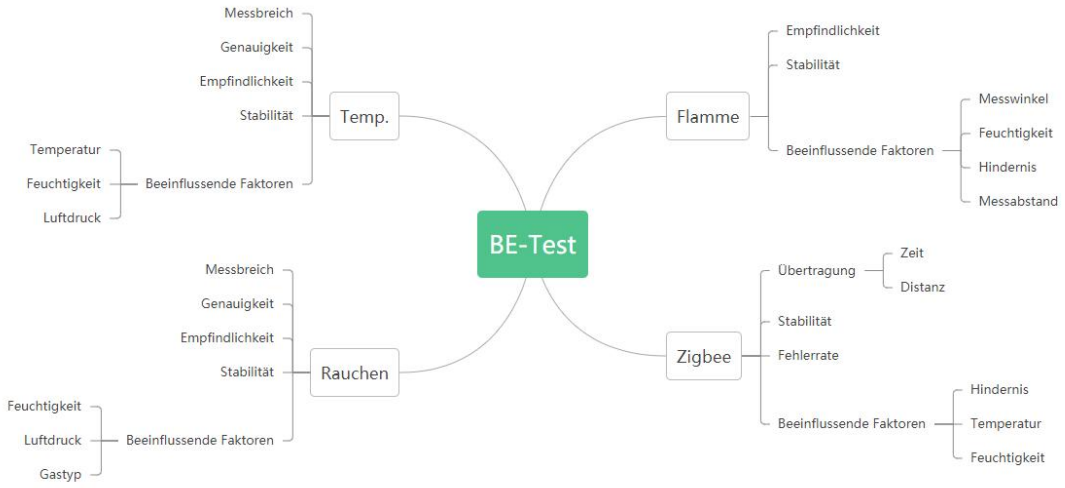
5.Empfindlichkeitstest:Ich teste die Empfindlichkeit des Sensors gegenüber Änderungen am Testobjekt und bewerte die Empfindlichkeit und den Reaktionsbereich des Sensors, indem ich das Versuchsobjekt variiere und die Reaktion des Sensors aufzeichne.

6.Umgebungsinterferenztest: Ich achte beim Sensortest auf die Auswirkungen von Umgebungsinterferenzfaktoren auf die Sensorleistung. Zum Beispiel andere Wärmequellen, Licht, Feuchtigkeitsschwankungen, Staub, Hindernisse.

7.Datenprotokollierung und -analyse: Ich stelle sicher, dass ich während des Tests Daten protokollieren, einschließlich Sensormesswerten, Umgebungsbedingungen und beobachteten Problemen. Die Daten werden mithilfe geeigneter Datenanalysetools analysiert, um wichtige Leistungsindikatoren und Schlussfolgerungen zu erhalten.

Zunächst habe ich ein Brainstorming für den Hardwaretest durchgeführt, die Sensoren analysiert, die mit dem Testprozess verbundenen Probleme und die Faktoren analysiert, die die Hardwaretestergebnisse während des Testprozesses beeinflussen können.

Dazu habe ich ein Diagramm erstellt, in dem detailliert aufgeführt ist, worauf jeder Sensor getestet werden muss.



38 Brainstorming

5.3 Sensortest

5.3.1 Temperatursensor

Testeinflussfaktoren:

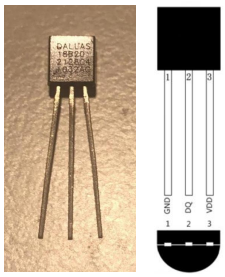
1 Strahlungswärmequelle: In der Testumgebung können andere Strahlungswärmequellen vorhanden sein, z. B. Sonnenlicht, starkes Licht usw. Diese Strahlungswärmequellen können direkt auf den Sensor strahlen und zu Fehlern in den Messergebnissen führen.

2 Wärmekapazität des Materials: Verschiedene Materialien haben eine unterschiedliche Wärmekapazität (Wärmeträgheit), d. h. die Reaktionsgeschwindigkeit des Materials auf Wärme und die Fähigkeit, Wärme zu speichern, sind unterschiedlich. Den Sensor umgebende Materialien können die Reaktionszeit und Messgenauigkeit des Sensors beeinträchtigen.

Ich habe mich für zwei verschiedene Modelle von Temperatursensoren entschieden, nämlich DS18B20 und PT100. Sie haben ihre eigenen Vor- und Nachteile, und schließlich habe ich mich durch Vergleich der Testergebnisse für eines davon als endgültigen Temperatursensor entschieden.

DS18B20

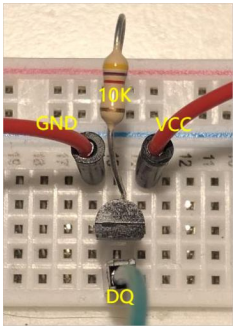
DS18B20 ist ein häufig verwendeter digitaler Temperatursensor mit integriertem Chip, der die Eigenschaften Verschleißfestigkeit und Schlagfestigkeit, geringe Größe, starke Entstörungsfähigkeit und hohe Präzision aufweist.DS18B20 ist einfach zu verdrahten und kann nach dem Verpacken bei vielen Gelegenheiten verwendet werden. Sein Aussehen ändert sich hauptsächlich je nach Anwendungsanlass.



39 DS18B20

Testablauf

Die Leitungsverbindung ist einfach: VCC, DQ und GND des DS18B20 sind jeweils mit 5 V, dem IO-Port und GND verbunden. Ein Ende des 10K Ohm-Widerstands ist mit VCC und das andere Ende mit DQ verbunden. Der Pull-up-Widerstand wird verwendet, um die Ansteuerfähigkeit des E/A-Ports zu verbessern.

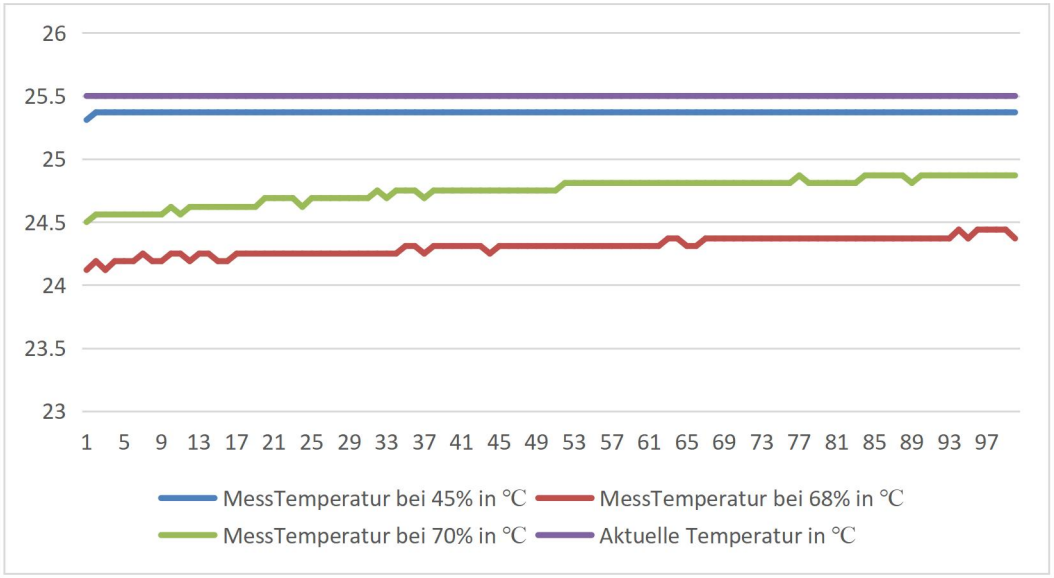


40 Schaltplan DS18B20

Testsdaten

Ich verwende ein Messgerät, um die Luftfeuchtigkeit rund um den Sensor zu messen. Der Temperatursensor wurde unter normalen und feuchten Innenbedingungen getestet. Der Temperatursensor misst die Temperatur mit einer Frequenz von 1 Sekunde unter verschiedenen Luftfeuchtigkeitsbedingungen. Datenerfassung, Datenanalyse und grafische Darstellung.

Gemäß den Messergebnissen von Standardmessgeräten beträgt der Umgebungstemperaturbereich in diesem Experiment 25,5–25,6 °C und die Lichtintensität 9 Lux.



1 Temperatur bei unterschiedlicher Feuchtigkeit (°C) DS18B20

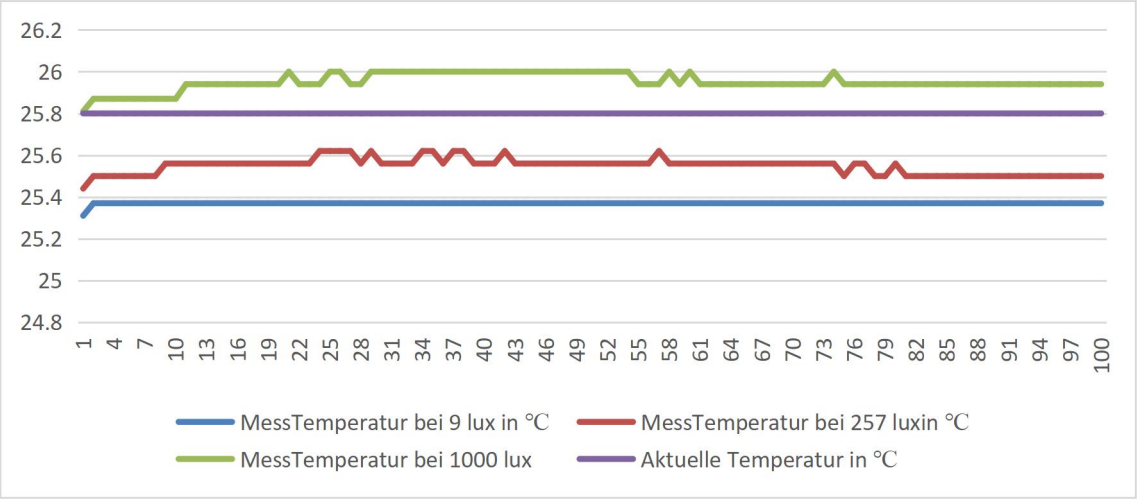
Analyse: Durch den Vergleich der Daten verschiedener Luftfeuchtigkeitsumgebungen kann festgestellt werden, dass mit zunehmender Luftfeuchtigkeit die Abweichung zwischen den Sensordaten und der tatsächlichen Temperatur größer wird und auch die Datenstabilität abnimmt. Die Abweichung beträgt etwa ein Grad Celsius.

Ich denke, es gibt zwei Gründe für dieses Ergebnis:

1 Wärmeleitung: Feuchtigkeit kann die Wärmeleitungseigenschaften der Umgebungsluft verändern und dadurch die Anzeige des Temperatursensors beeinflussen. Luft in einer Umgebung mit hoher Luftfeuchtigkeit leitet Wärme im Allgemeinen besser als Luft in einer Umgebung mit niedriger Luftfeuchtigkeit, was zu niedrigen Temperatursensorwerten führen kann.

2 Verdunstung: In einer Umgebung mit hoher Luftfeuchtigkeit kann es zu Wasserverdunstung auf der Oberfläche des Temperatursensors kommen. Während dieses Vorgangs wird Wärme absorbiert und der Sensor zeigt möglicherweise einen niedrigeren Temperaturwert an.

Ich verwende ein Messgerät, um die Lichtintensität um den Sensor herum zu messen. Ich teste den Temperaturregler, ohne das Licht einzuschalten, die Innenbeleuchtung einzuschalten und bei starkem Licht. Der Temperatursensor misst die Temperatur mit einer Frequenz von 1 Sekunde bei unterschiedlichen Lichtintensitäten. Datenerfassung, Datenanalyse und grafische Darstellung.



2Temperatur bei unterschiedlicher lux (°C) DS18B20

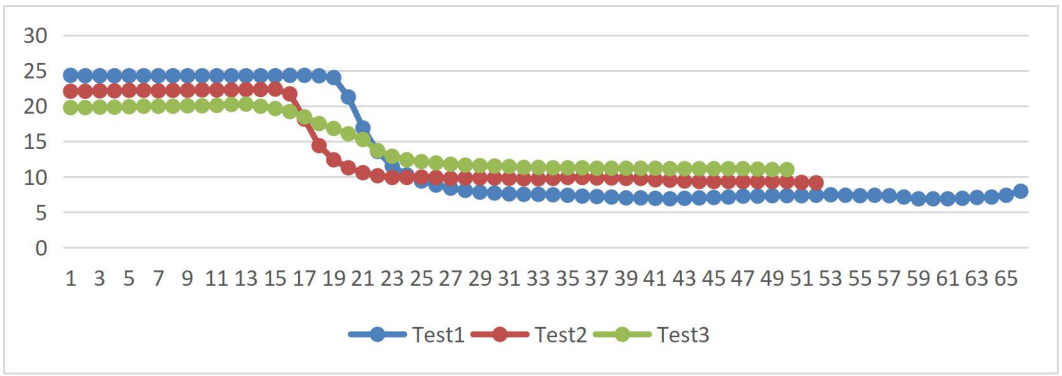
Analyse:Durch den Vergleich der Daten verschiedener Beleuchtungsumgebungen kann festgestellt werden, dass die Sensordaten in einer dunklen Umgebung stabil um 0,1 Grad Celsius niedriger sind als die tatsächliche Temperatur. Mit zunehmender Lichtintensität werden die Sensordaten immer größer und überschreiten allmählich die tatsächliche Temperatur, und auch die Datenstabilität nimmt ab.

Gründe für dieses Ergebnis:

1 Strahlungswärme: Die Strahlungsenergie im Licht kann direkt in Wärmeenergie umgewandelt werden, was zu hohen Messwerten des Temperatursensors führen kann. Insbesondere wenn der Temperatursensor starker Sonneneinstrahlung ausgesetzt ist, kann die Strahlungswärme im Licht die Messergebnisse des Sensors erheblich beeinträchtigen.

Empfindlichkeits- und Stabilitätstests

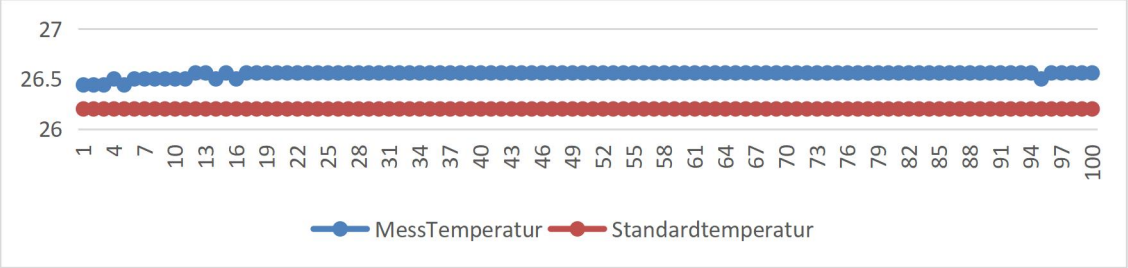
Die Sensoren wurden drei Testreihen unterzogen. Zuerst platziere ich den Sensor in einer Luftumgebung und trockne den Sensor mit einem Papiertuch. Nachdem ich die Stromversorgung angeschlossen hatte, öffnete ich die serielle Schnittstelle und begann mit der Datenaufzeichnung. Ich habe die Temperatur in der Luft gemessen und aufgezeichnet, dann habe ich den Sensor in Wasser getaucht und aufgezeichnet, wie sich die Daten während des Prozesses verändert haben.



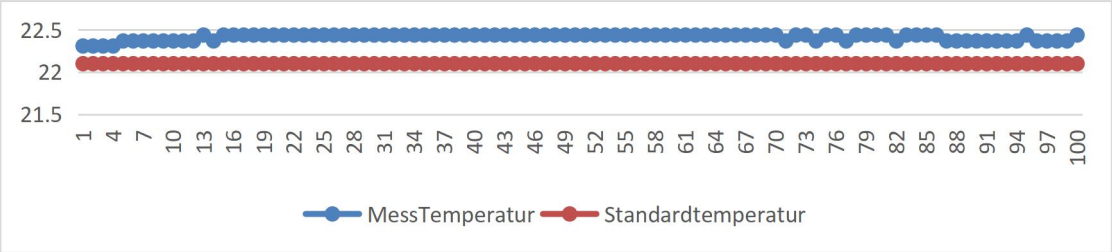
3 Empfindlichkeitstests DS18B20

Analyse: Nachdem der Sensor in Wasser gelegt wurde, ändern sich die Messdaten innerhalb kurzer Zeit schnell und die Messtemperatur sinkt schnell, was darauf hinweist, dass der Sensor normal arbeiten kann und eine hohe Empfindlichkeit aufweist, wenn sich die Temperatur schnell ändert. Nach längerem Einlegen in Luft und Wasser ist der Datenänderungsbereich sehr gering und tendenziell stabil, was darauf hinweist, dass der Sensor Daten in der entsprechenden Umgebung stabil und kontinuierlich aufzeichnen kann und die Stabilität sehr gut ist.

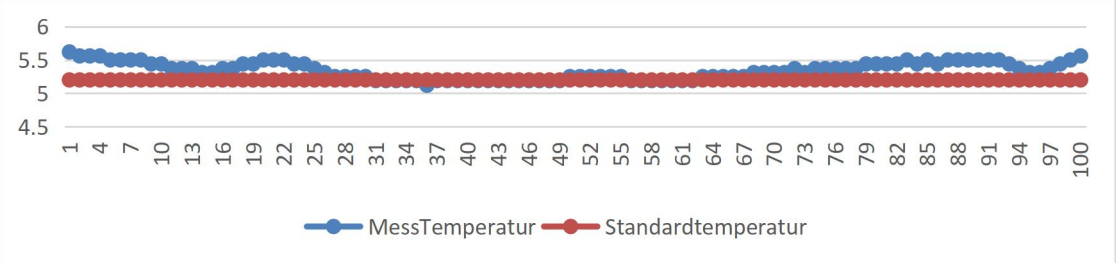
Genauigkeit



4 GenauigkeitTest 1 DS18B20



5 GenauigkeitTest 2 DS18B20



6 GenauigkeitTest 3 DS18B20

Analyse: Durch den Vergleich des Fehlers zwischen den Sensordaten und der tatsächlichen Temperatur bei verschiedenen Temperaturen beträgt der tatsächliche Fehler weniger als 0,5 Grad Celsius, was innerhalb der Designsicherheitsspezifikation liegt und die Produktanforderungen erfüllt.

Vorteile und Nachteile:

Vorteil:

1. Geringe Größe, geringer Hardware-Overhead, starke Anti-Interferenz-Fähigkeit

2. Es ist relativ einfach zu bedienen und erfordert keine komplizierte Programmierung.

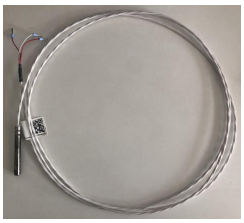
3. Die Temperaturmessung kann ohne den Einsatz externer Komponenten realisiert werden.

4. Unterdruckeigenschaften. Wenn die Polarität der Stromversorgung umgekehrt wird, verbrennt das Thermometer nicht durch Hitze, es funktioniert jedoch nicht normal.

Nachteile:

1. Die Genauigkeit ist durchschnittlich und kann nur in Fällen erfüllt werden, in denen die Genauigkeitsanforderungen nicht hoch sind

PT100

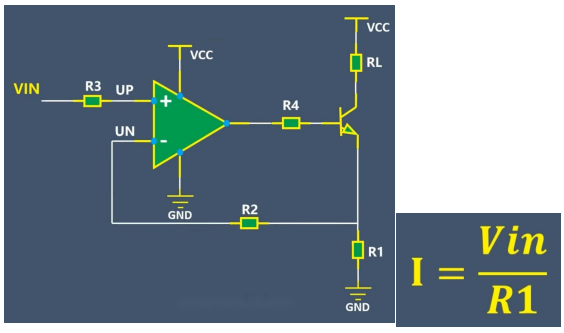


41 PT100

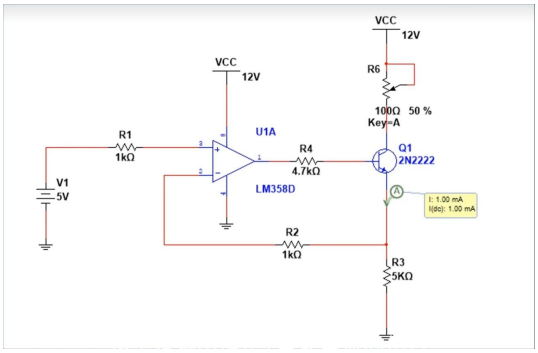
Der PT100-Sensor hat einen Widerstandswert von 100 Ohm bei 0°C, daher der Name PT100. Der PT100-Sensor verwendet Platin als Widerstandsmaterial, das eine lineare Temperatur-Widerstandsbeziehung aufweist. Es hat einen Temperaturkoeffizienten von 3850 ppm/°C, was bedeutet, dass sich der Widerstandswert bei jeder Grad Celsius-Änderung um etwa 0,385 % des Temperaturänderungswerts ändert.

Testablauf

Für den normalen Gebrauch muss PT100 mit einer Konstantstromquelle verwendet werden. Also baute ich eine einfache Konstantstromquelle und nutzte den internen ADC des Mikrocontrollers, um den Wert des PT100-Sensors abzulesen und die Temperatur zu berechnen.

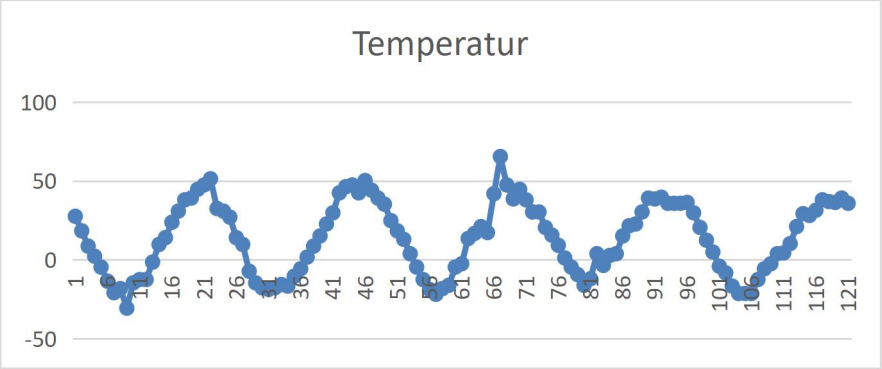


42 Schaltplan 1 PT100



43 Schaltplan 2 PT100

Testsdaten



7 Messdaten PT100

Analyse：Periodisch wechselnde Temperaturwerte können durch Rauschen und Störungen im Stromkreis verursacht werden. Diese Störungen können von Netzteilen, Kabeln, Platinenführung usw. herrühren. Im analogen Eingangssignal können diese Störungen durch die Änderung des Widerstandswertes das Temperaturmessergebnis beeinflussen.

Durch den Einsatz komplexerer Konstantstromquellen und Komponenten können Störungen reduziert werden.

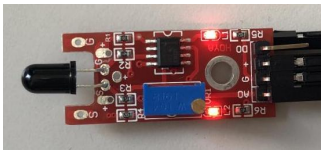
Da der PT100-Sensor in Kombination mit einer einfachen Konstantstromquelle keine korrekte Temperaturmessung erreichen kann, werde ich nach Abschluss der grundlegendsten Tests keine Stabilitäts- und Genauigkeitstests mehr durchführen.

	Genauigkeit	*Empfindlichkeit*	*Stabilität*
DS18B20	*good*	*good*	*good*
PT100	*schlecht*	*schlecht*	*schlecht*

8 Verglichung

Gemäß den experimentellen Zielen und tatsächlichen Testergebnissen sind die Testergebnisse von DS18B20 besser und können die Projektanforderungen erfüllen, sodass schließlich der DS18B20-Sensor ausgewählt wird

5.3.2 Flammensensor HY-A1



44 HY-A1

Testeinflussfaktoren:

1 Lichtverhältnisse: Flammensensoren verwenden typischerweise lichtempfindliche Elemente, um das Vorhandensein von Flammen zu erkennen. Daher können die Lichtverhältnisse in der Testumgebung einen Einfluss auf die Messergebnisse des Sensors haben. Starke Lichtquellen, gestreutes Sonnenlicht oder andere Lichtquellen können die Flammenerkennung des Sensors beeinträchtigen.

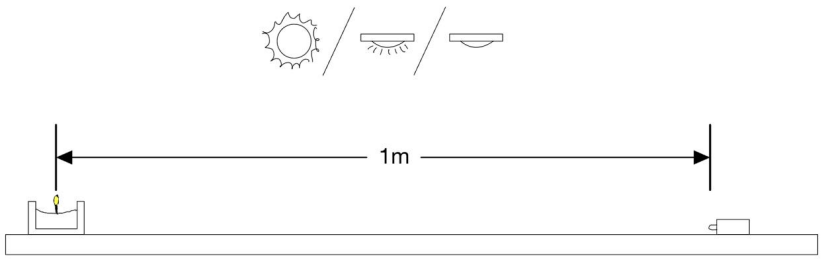
2 Rauch und Staub: Rauch, Staub oder andere Partikel in der Testumgebung können die Messung des Flammensensors beeinträchtigen. Diese Partikel können das vom Feuer erzeugte Lichtsignal streuen oder absorbieren und dadurch die Erkennung des Sensors beeinträchtigen.

3 Störquellen: Das Vorhandensein anderer Wärmequellen oder Lichtquellen kann die Messergebnisse des Flammensensors beeinträchtigen. Beispielsweise können in der Nähe befindliche Heizgeräte, Lichter oder andere Flammenquellen zu falsch-positiven oder falsch-negativen Ergebnissen führen.

Merkmale: Die analoge Ausgangsspannung nimmt mit zunehmender Lichtintensität in der Erkennungsumgebung ab.Die Messdatengröße des Flammensensors ist umgekehrt proportional zur gemessenen Lichtintensität. Je größer die Lichtintensität, desto kleiner ist der Messwert.

Testablauf

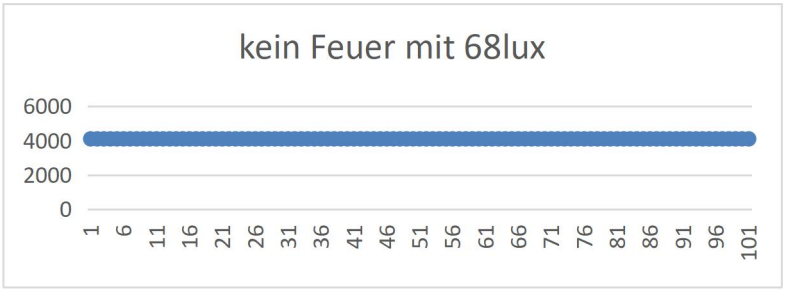
Ich befestige die zu testende Lichtquelle in einem Abstand von einem Meter vom Sensor, schalte den Strom ein, öffne die serielle Schnittstelle, um Daten aufzuzeichnen, und beobachte die Datenänderungen. Ich zeichne die vom Sensor erfassten Daten in einer dunklen Innenumgebung (Licht ausschalten), in einer hellen Innenumgebung (Licht einschalten) und in einer hellen Außenlichtumgebung (Sonneneinstrahlung) auf. Dann vergleiche ich die Unterschiede zwischen den Daten.



45 Testablauf HY

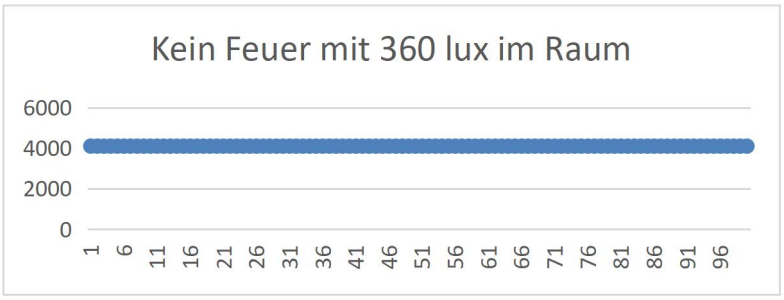
Testsdaten

Ich platziere den Sensor in einem schwach beleuchteten Raum. Die relative Luftfeuchtigkeit beträgt 49 % und die relative Temperatur 25,7 °C



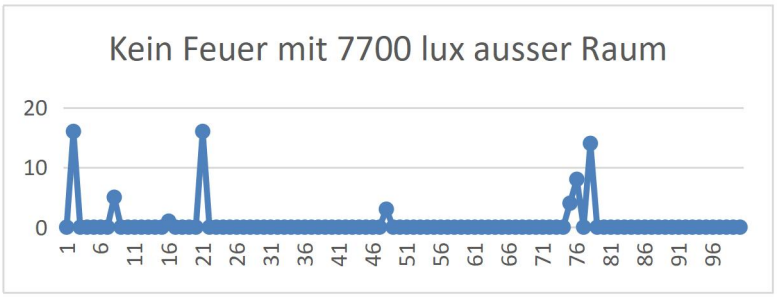
8 kein Feuer mit 68lux HY

Dann stelle ich den Sensor in helles Innenlicht und füge daneben eine gleichmäßige Lichtquelle hinzu. Die relative Luftfeuchtigkeit beträgt 49 % und die relative Temperatur 25,7 °C



9 Kein Feuer mit 360 lux im Raum HY

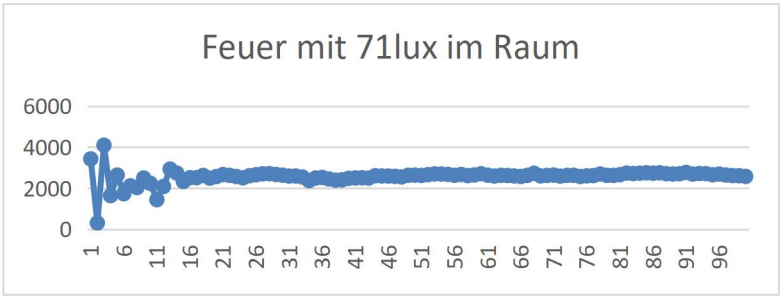
Als nächstes stelle ich den Sensor in helles Außenlicht. Die relative Luftfeuchtigkeit beträgt 48 % und die relative Temperatur 26,3 °C



10 Kein Feuer mit 7700 lux ausser Raum HY

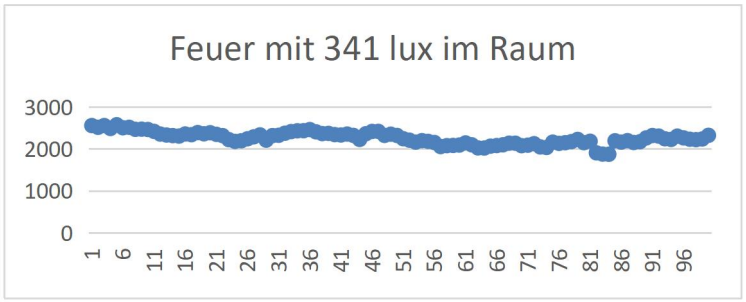
Analyse: Wenn keine Flamme vorhanden ist und der Sensor starken Lichtumgebungen wie Sonnenlicht oder starker Beleuchtung ausgesetzt ist, können diese zusätzlichen Lichtquellen fälschlicherweise als Flammen identifiziert werden, was die vom Sensor gemessenen Daten beeinträchtigt und zu Fehlalarmen führt oder Ungenauigkeiten Die Messergebnisse der Messergebnisse.

Als nächstes stelle ich den Sensor in einen schwach beleuchteten Raum und zünde eine Kerze an. Die relative Luftfeuchtigkeit beträgt 49 % und die relative Temperatur 25,7 °C



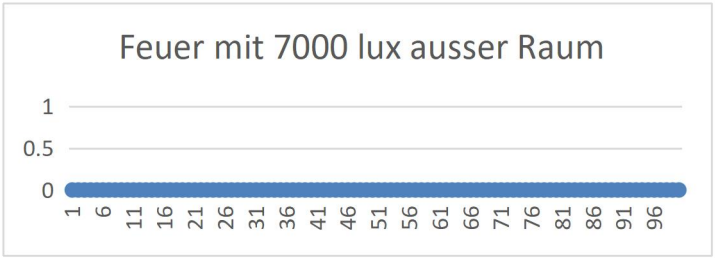
11 Feuer mit 71lux im Raum HY

Dann stellte ich den Sensor in eine helle Raumlichtumgebung mit einer gleichmäßigen Lichtquelle daneben und zündete eine Kerze an. Die relative Luftfeuchtigkeit beträgt 49 % und die relative Temperatur 25,7 °C



12 Feuer mit 341 lux im Raum HY

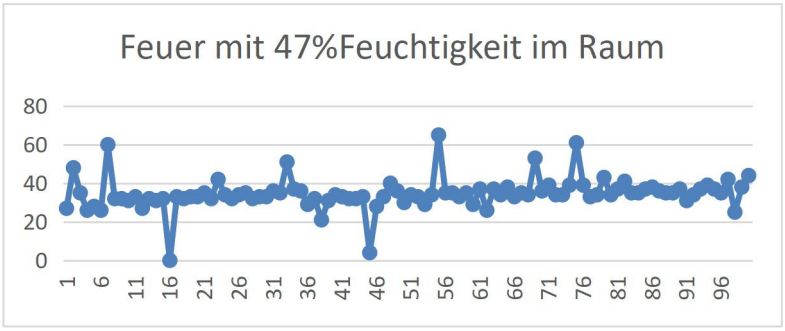
Anschließend habe ich den Sensor in helles Außenlicht gestellt und eine Kerze angezündet. Die relative Luftfeuchtigkeit beträgt 48 % und die relative Temperatur 26,3 °C



13 Feuer mit 7000 lux ausser Raum HY

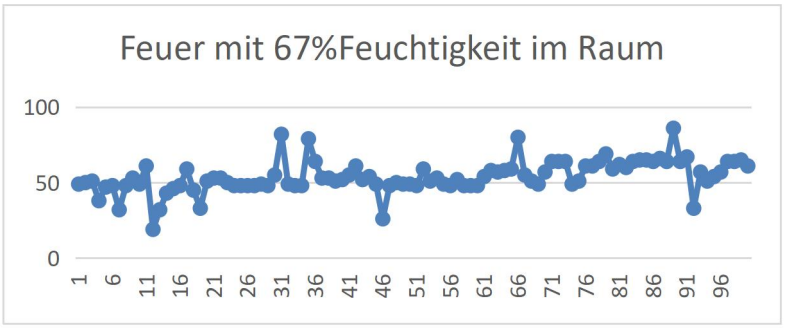
Analyse: Wenn eine Flamme vorhanden ist und der Sensor starkem Licht ausgesetzt ist, beeinträchtigt dies auch die vom Sensor gemessenen Daten. Mit zunehmender Intensität des Umgebungslichts nimmt der Durchschnittswert der Messung allmählich ab Einfluss auf die Messergebnisse haben.

Ich stelle den Sensor in einen schwach beleuchteten Raum und zünde eine Kerze an. Die relative Luftfeuchtigkeit beträgt 47 % und die Temperatur 26,7 °C



14 Feuer mit 47%Feuchtigkeit im Raum HY

Ich stelle den Sensor in einen schwach beleuchteten Raum und zünde eine Kerze an. Die relative Luftfeuchtigkeit beträgt 67 % und die Temperatur beträgt 26,7 °C

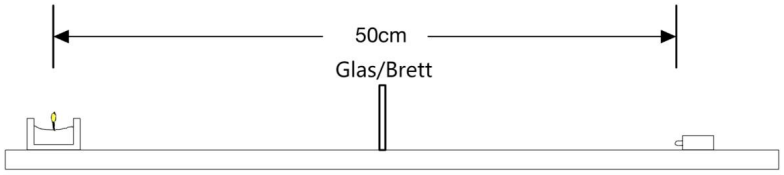


15 Feuer mit 67 %Feuchtigkeit im Raum HY

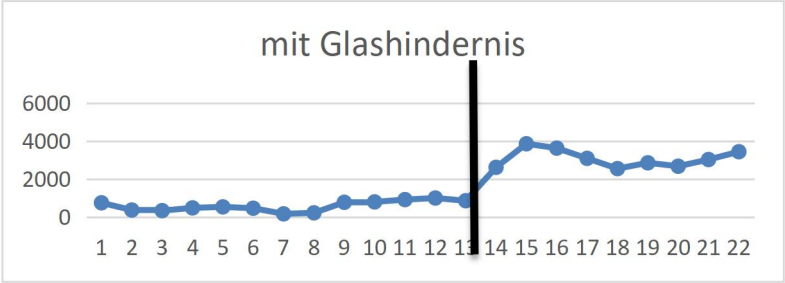
Analyse: Wenn eine Flamme vorhanden ist und der Sensor einer feuchten Umgebung ausgesetzt ist, beeinträchtigt dies auch die vom Sensor gemessenen Daten. In einer Umgebung mit hoher Luftfeuchtigkeit befindet sich mehr Feuchtigkeit in der Luft, was zu einer stärkeren Lichtstreuung führen kann in der Luft. Diese Streuphänomene können die vom Flammensensor empfangene Lichtintensität schwächen und die Empfindlichkeit und Genauigkeit der Messergebnisse beeinträchtigen. Mit zunehmender Luftfeuchtigkeit nimmt die Wahrscheinlichkeit einer Lichtstreuung zu, was zu einem allmählichen Anstieg des Messwertmittelwerts führt und die Messergebnisse beeinflusst.

Mit Hindernistest

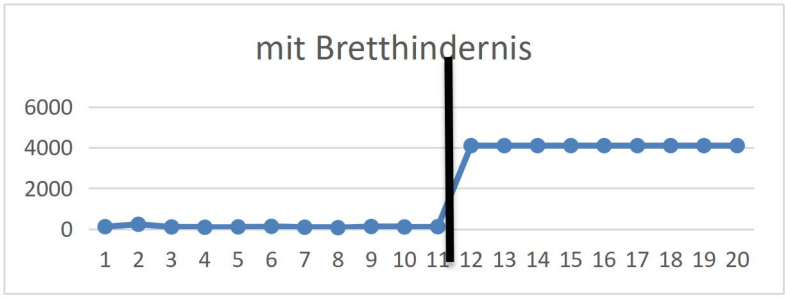
Testablauf: Die Versuchsumgebung ist dunkel und es gibt keine anderen Störquellen. Ich platziere den Sensor 50 cm von der Kerze entfernt.Dann platzieren ich eine Glasscheibe und ein Holzbrett zwischen Kerze und Sensor, schalte den Strom ein und zeichne die Sensordaten auf. Ich beobachte die Änderungen der Sensordaten vor und nach dem Platzieren von Hindernissen.



46 Hindernistest HY



16 mit Glashindernis HY



17 mit Bretthindernis HY

Analyse: Durch den Vergleich der Sensordaten vor und nach der Platzierung des Hindernisses kann festgestellt werden, dass sich die Daten vor und nach der Platzierung stark verändert haben. Selbst wenn es sich bei dem Hindernis um eine Glasscheibe handelt, die Licht durchlassen kann, wird dadurch die Intensität des Infrarotlichts verursacht Die vom Sensor empfangenen Daten nehmen ab, was zu größeren Daten führt.

5.3.3 Rauchsensor MQ2

Testeinflussfaktoren:

1 Rauchkonzentration: Die Messergebnisse des Rauchsensors werden durch die Rauchkonzentration in der Umgebung beeinflusst. Höhere Rauchkonzentrationen führen dazu, dass der Sensor stärker reagiert, während niedrigere Rauchkonzentrationen vom Sensor möglicherweise nicht richtig erkannt werden.

2 Feinstaubarten: Die Empfindlichkeit des Rauchsensors kann je nach Feinstaubart in der Umgebung variieren. Verschiedene Arten von Rauchpartikeln (z. B. durch Verbrennung entstehende Partikel, Dampf oder Dunst usw.) haben unterschiedliche Auswirkungen auf die Erkennungsfähigkeit des Sensors.

3 Temperatur und Luftfeuchtigkeit: Änderungen der Temperatur und Luftfeuchtigkeit der Umgebung können die Messergebnisse des Rauchsensors beeinflussen. Extreme Temperatur- oder Feuchtigkeitsbedingungen können zu einer Verschlechterung der Sensorleistung oder Fehlalarmen führen.

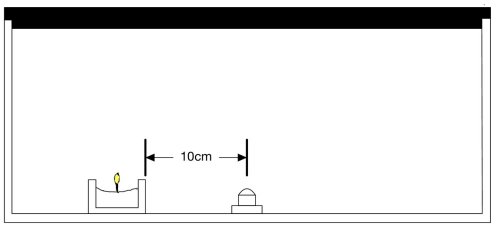
47 MQ2

Merkmale: Großer Erfassungsbereich, hohe Empfindlichkeit, schnelle Reaktionswiederherstellung, ausgezeichnete Stabilität, lange Lebensdauer, einfache Antriebsschaltung. Die Empfindlichkeit kann durch Einstellen des Potentiometers eingestellt werden. Generell gilt: Je kleiner der Widerstandswert des Potentiometers, desto höher die Empfindlichkeit.Die analoge Ausgangsspannung steigt mit zunehmender Gaskonzentration in der Detektionsumgebung

Es ist zu beachten, dass der Rauchsensor MQ-2 nur brennbare Gase und Rauch erkennen kann und nicht zwischen verschiedenen Arten brennbarer Gase unterscheiden kann. Es muss während des Gebrauchs eine Zeit lang vorgewärmt werden, damit das interne Heizelement die Arbeitstemperatur erreicht und dann normal erkannt werden kann.

Testablauf

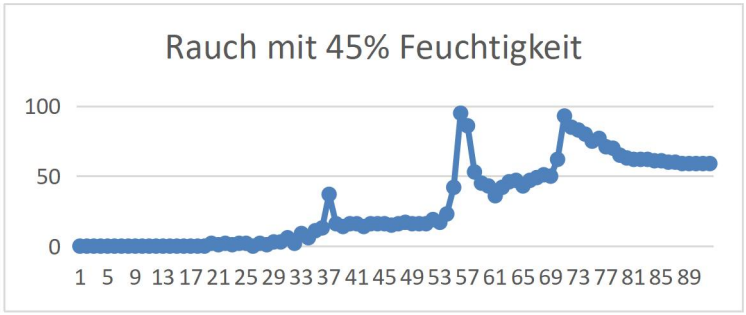
Ich habe den Rauchsensor in die Testbox gelegt und den Sensor unten in der Mitte der Box festgeklebt. Dann schaltete ich den Strom ein, wartete ein paar Minuten, öffnete die serielle Schnittstelle zum Beobachten, drückte den Feuerzeugschalter, zündete aber nicht, ließ das brennbare Gas im Feuerzeug herausspritzen und testete damit den Sensor und beobachtete, ob das Der Computer hat die Daten korrekt empfangen. Nachdem ich bestätigt hatte, dass es richtig war, zündete ich die Kerze an und platzierte die Kerze in einem Abstand von 10 cm vom Sensor. Dann habe ich den Deckel aufgesetzt und die Verbindung mit Klebeband abgedichtet. Ich stelle die Box auf einen stabilen Ort. Ich habe eine Weile gewartet, bis die Kerze brennt, bevor ich Daten protokolliert habe.



48 Testablauf MQ2(1)

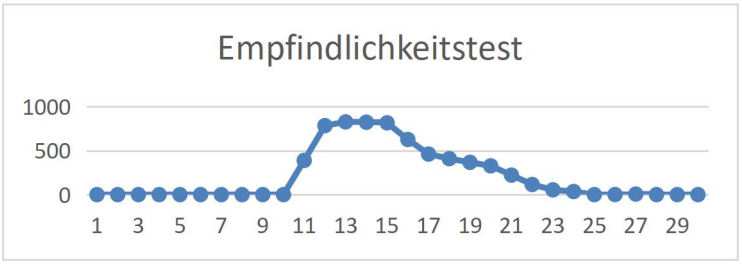
49 Testablauf MQ2(2)

Testsdaten



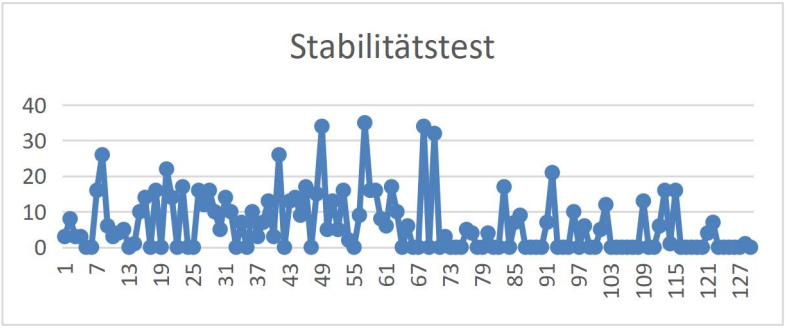
18 Rauch mit 45% Feuchtigkeit

Empfindlichkeitstest: Ich drücke den Feuerzeugschalter, zünde aber nicht, lasse das brennbare Gas im Feuerzeug herausspritzen, ändere in kurzer Zeit die Konzentration des brennbaren Gases in der Testumgebung und beobachte, ob sich die Daten entsprechend ändern.



19 Empfindlichkeitstest MQ2

Stabilitätstest: Ich beobachte, ob die Daten für einen bestimmten Zeitraum nach Abschluss des Vorgangs abnormale Daten enthalten.



20 Stabilitätstest MQ2

Datenanalyse:

Wenn die Sauerstoffkonzentration in der Box immer geringer wird, brennt die Kerze allmählich unvollständig ab, die brennbaren Partikel in der Luft nehmen allmählich zu und die Sensordaten nehmen allmählich zu. Nachdem die sauerstoffarme Kerze gelöscht wurde, verfestigt sich das Wachs aufgrund des Temperaturabfalls allmählich, sodass die Konzentration der brennbaren Stoffe nicht ständig zunimmt, sondern nach einiger Zeit die maximale Konzentration erreicht. Wenn dann die Flamme erlischt, sinkt die Temperatur des Gases in der Box allmählich, da die Boxwand mit der Außenwelt in Kontakt steht und die Abkühlungsrate schneller ist als die der Innenluft. Die brennbaren Stoffe im Tank werden allmählich abgekühlt reduziert, und die Sensordaten werden daher schrittweise reduziert.

Vorteile und Nachteile:

Vorteil:

1. Hohe Empfindlichkeit

2. Schnelle Reaktion und Wiederherstellung

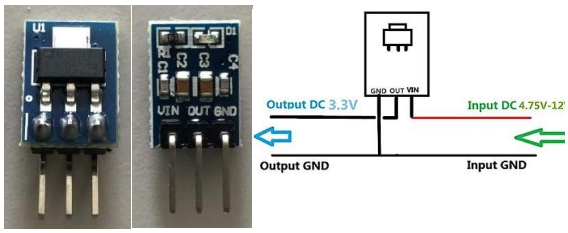
3. Die Antriebsschaltung ist einfach

Nachteile:

1. Verschiedene Arten brennbarer Gase können nicht unterschieden werden.

2. Es muss eine Zeit lang aufgewärmt werden, bevor es normal erkannt werden kann.

5.4 Spannungsumwandlungschip



50 Spannungsumwandlungschip

Dieses Modul ist ein positives Spannungsregler-Abwärtsstromversorgungsmodul, das einen DC-Eingang von 4,75–12 V und eine feste Spannung von 3,3 V und einen Stromausgang von 1A unterstützt.

VIN	*4,75 V bis 12 V*
VOUT	*3,267 bis 3.333 V*
Ausgangsstrom	*1A*

9 Parameter Spannungsumwandlungschip

Testablauf

Ich verwende ein einstellbares Netzteil, das an den Eingang angeschlossen ist, um die Eingangsspannung bereitzustellen, und ein an den Ausgang angeschlossenes Voltmeter, um die Ausgangsspannung zu messen. Ich passe die Eingangsspannung an und beobachte die Änderung der Ausgangsspannung, um zu beurteilen, ob die Leistung des Abwärtsmoduls mit den Parametern übereinstimmt.

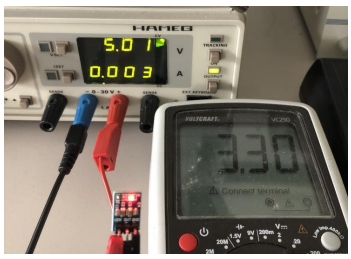
Testsdaten

51 Umwandlungsspannung(4.7-12)

Gemäß den theoretischen Parametern habe ich die Umwandlungsspannung gemessen, wenn die Eingangsspannung 4, 7 V und 12 V beträgt und die Ausgangsspannung bei 3,3 V stabilisiert werden kann.

Stabilitätstest

Die im Test verwendete Eingangsspannung beträgt 5 V, daher habe ich die Stabilität bei 5 V gemessen. Die Ausgangsspannung kann auf 3,3 V stabilisiert werden. Nach einer Testphase kam es zu keiner Überhitzung des Chips.



52 Umwandlungsspannung(5V)

5.5 Zigbee Übertragungssystem

5.5.1 Punkt-zu-Punkt-Übertragung

5.5.1.1 Übertragungsgeschwindigkeit Test

Die Testmethode besteht darin, denselben Computer zu verwenden, um den Endknoten und den Koordinator jeweils über die serielle Schnittstelle zu verbinden. Ich schreibe ein Sendeprogramm. Wenn der Endknoten Informationen sendet, sendet er gleichzeitig Eingabeaufforderungsinformationen an die serielle Schnittstelle. Der Koordinator druckt die Daten aus und zeigt sie über die serielle Schnittstelle an, nachdem er die Daten empfangen hat. Ich verwende den Zeitstempel, um die Zeitdifferenz zwischen Senden und Empfangen zu berechnen. Mehrere Messungen wurden gemittelt.

Die Messergebnisse sind in der Abbildung dargestellt. Vom Terminal, das Daten sendet, bis zum Koordinator, der Daten empfängt, bis hin zur seriellen Schnittstelle des Computers zum Lesen und Empfangen von Informationen dauert der gesamte Vorgang nur 10 ms, was sehr schnell ist. Daher kann die Aktualität der vom Koordinator empfangenen Daten garantiert und die Arbeitsumgebung des Endknotens schnell wiedergegeben werden, was für das Benutzerterminal von Vorteil ist, um eine schnelle Beurteilung zu treffen.

```
[20:41:03.781] 5E9D,0000          [20:41:03.791] &&END5E9D$
[20:41:33.780] 5E9D,0000          [20:41:33.790] &&END5E9D$
```

53Ergebnisse des Übertragungsgeschwindigkeitstests

5.5.1.2 Übertragungsstabilität

1.Die Testumgebung einrichten:Ich stelle sicher, dass sich in der Testumgebung keine anderen drahtlosen Störquellen befinden, und ich lege den Abstand zwischen dem

Testgerät und dem Empfangsgerät fest. Es kann eine feste Sendeleistung und Datenrate verwendet werden.

2.Die ÜbertragungsstabilitätTesten: MESS-Knoten senden kontinuierlich eine Reihe von Datenpaketen im eingestellten Abstand und ich zeichne die Anzahl der erfolgreich vom empfangenden Gerät empfangenen Datenpakete auf. Es kann ein Zeitfenster eingestellt werden, beispielsweise 10 Sekunden, und die Anzahl der innerhalb dieses Zeitfensters erfolgreich empfangenen Datenpakete gezählt werden.

3.Die Übertragungsentfernung Erhöhen: Ich erhöhe schrittweise die Entfernung zwischen dem Testgerät und dem Empfangsgerät und wiederhole Schritt 2, um die Anzahl der erfolgreich empfangenen Datenpakete in unterschiedlichen Entfernungen aufzuzeichnen.

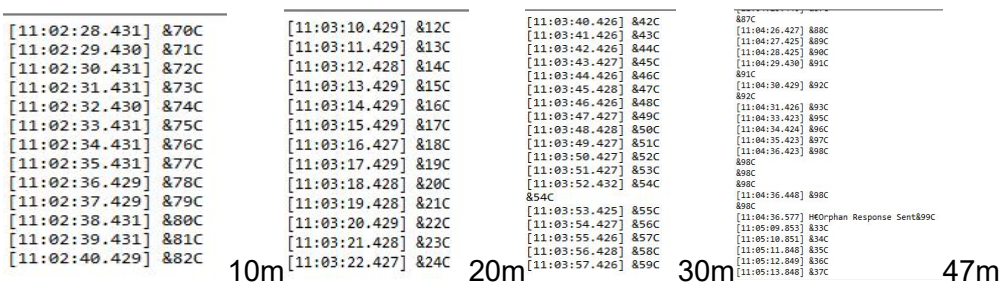
4. Daten analysieren: Anhand der Anzahl der aufgezeichneten Datenpakete kann die Übertragungsstabilität bei unterschiedlichen Entfernungen berechnet werden, beispielsweise der Prozentsatz erfolgreich empfangener Datenpakete oder die Verlustrate von Datenpaketen. Es ist auch möglich, die durchschnittliche Anzahl erfolgreich empfangener Pakete pro Sekunde zu berechnen.

5.Ein Diagramm der Beziehung zwischen Übertragungsentfernung und Übertragungsstabilität zeichnen: Ich nehme die Übertragungsentfernung als Abszisse und den Übertragungsstabilitätsindex als Ordinate und zeichne ein Diagramm des Einflusses der Übertragungsentfernung auf die Übertragungsstabilität. Auf diese Weise kann der Trend der Übertragungsstabilität mit zunehmender Übertragungsentfernung beobachtet werden.

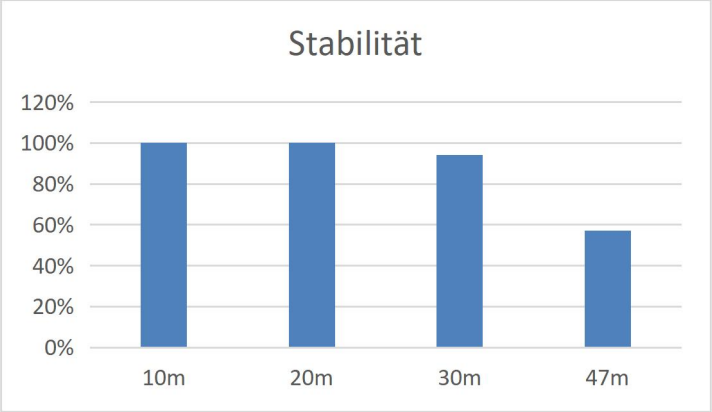
Durch die oben genannten Schritte kann die Einflussbeziehung der Übertragungsentfernung auf die Stabilität des Zigbee-Übertragungssystems ermittelt werden. Normalerweise kann die Übertragungsstabilität mit zunehmender Übertragungsentfernung aufgrund der Signaldämpfung und der Erhöhung der Übertragungsfehlerrate abnehmen, was zu einem erhöhten Paketverlust führt.

Spezifischer Testinhalt: Der Koordinator bleibt fest und stellt über die serielle Schnittstelle eine Verbindung zum Computer her, um die vom Endknoten gesendeten Informationen zu empfangen. Ich ermittle die Positionen fester Punkte, z. B. 10 m, 20 m und 30 m vom Koordinator entfernt, platziere den Endknoten auf dem Punkt und beginne dann mit der Datenübertragung. Ich überprüfe die vom Koordinator erhaltenen Daten. Ich analysiere abschließend die empfangenen Daten und beurteile die Übertragungsstabilität.

Nachfolgend die Messergebnisse ohne Hindernisse:



54Ergebnisse des Übertragungsstabilitätstests



21Ergebnisse des Übertragungsstabilitätstests

Ich wähle bei jedem Test ein Zeitfenster aus. Durch Berechnen des Verhältnisses der Anzahl korrekter Datenübertragungen zu allen Sendezeiten von Daten kann die Stabilität der Übertragung unter der aktuellen Übertragungsentfernung ermittelt werden. Die Umwandlung der Daten in ein Diagramm zeigt intuitiv, dass mit zunehmender Entfernung der Anteil der korrekten Übertragung immer geringer wird und die Übertragungsstabilität schlechter wird.

Um zu überprüfen, ob Zigbee über einen längeren Zeitraum normal funktionieren kann, habe ich einen Dauertest durchgeführt. Wie in der Abbildung gezeigt, kann der Koordinator die Sensordaten weiterhin normal empfangen, wenn ich 12 Stunden lang weiterhin Daten übertrage. Dies bedeutet, dass das ZigBee-Übertragungssystem eine hervorragende Haltbarkeit aufweist und Daten über einen langen Zeitraum übertragen kann.

```
[21:27:05.985] &&ROU45B1$ Zahl 000008545 Co 000008545 Fire 000008545 Temp2 000008545 $$
[21:27:10.986] &&ROU45B1$ Zahl 000008546 Co 000008546 Fire 000008546 Temp2 000008546 $$
[21:27:15.994] &&ROU45B1$ Zahl 000008547 Co 000008547 Fire 000008547 Temp2 000008547 $$
[21:27:20.985] &&ROU45B1$ Zahl 000008548 Co 000008548 Fire 000008548 Temp2 000008548 $$
[21:27:25.985] &&ROU45B1$ Zahl 000008549 Co 000008549 Fire 000008549 Temp2 000008549 $$
```

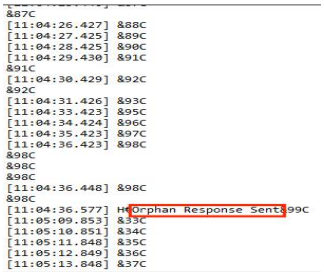
55Haltbarkeit Test

8549*5sec=42745sec≈12Stunden(9Uhr-21Uhr)

5.5.1.3 Übertragungsreichweite

Punkt-zu-Punkt-Übertragung

Keine Hindernisse: Zusammen mit dem Stabilitätstest wurde auch die Übertragungsreichweite gemessen. Gemäß den oben genannten Testergebnissen sendet das Gerät unter Außenbedingungen ohne Hindernisse nach mehreren Messungen, wenn die Übertragungsentfernung **47 m** erreicht, eine Orphan Response Sent-Nachricht, was bedeutet, dass der Knoten zu einem Waisenknoten geworden ist und die Verbindung getrennt wurde. Daher kann davon ausgegangen werden, dass die maximale Übertragungsentfernung zwischen einem einzelnen Knoten 47 m beträgt.

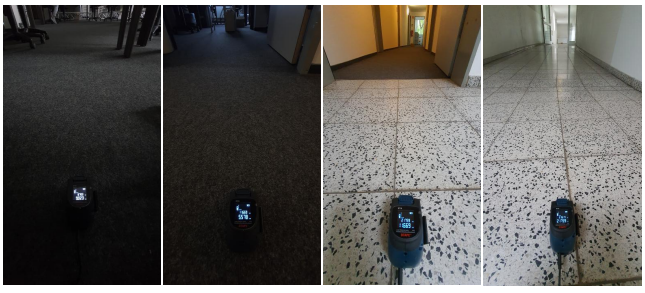


56Maximale Übertragungsentfernung ohne hindernisse

Mit Hindernisse: Ich verwende die Laborumgebung, um die Innenumgebung des Brandorts zu simulieren. Zu diesem Zweck habe ich ein schematisches Diagramm erstellt, das die Hausstruktur der Versuchsumgebung zeigt. In der Versuchsumgebung gibt es Hindernisse aus verschiedenen Materialien wie Holz, Stahlstangen, Beton, Glas usw., die den Brandort in einem Innenbereich gut simulieren können. Die roten Punkte in der Abbildung stellen den Koordinator dar und die schwarzen Punkte stellen die Sensorknoten dar. Das blaue Liniensegment stellt den gemessenen Abstand zwischen Messpunkten dar, und das orangefarbene Liniensegment stellt die maximale geradlinige Übertragungsentfernung zwischen Knoten dar.

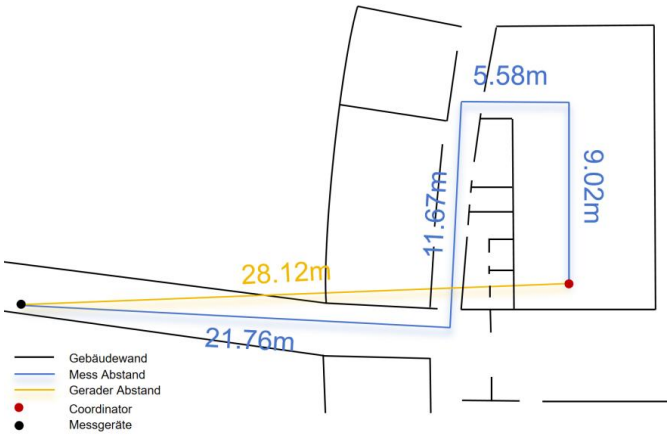
Wenn sich der Endknoten bewegt, nimmt die Übertragungsstabilität mit zunehmender Bewegungsdistanz allmählich ab und es kommt zu Datenverlust. Wenn sich das Gerät zum schwarzen Punkt bewegt, kann der Koordinator keine Informationen empfangen. Nach einer gewissen Wartezeit kann der Koordinator die Daten immer noch nicht empfangen. Nach der Rückkehr zum ursprünglichen Pfad verbindet sich das Endgerät erneut und überträgt Daten. Nach wiederholten Tests werden die Knoten in der Nähe des schwarzen Punktes getrennt. Daraus lässt sich schließen, dass, wenn sich das Endgerät in die Position des schwarzen Punkts bewegt, der Abstand zwischen den Geräten den maximalen Übertragungsabstand überschreitet und der Abstand zu diesem Zeitpunkt ungefähr dem maximalen Abstand für die Innenübertragung entspricht. Durch den Vergleich der Messdaten ohne Hindernisse kann geschlossen werden, dass bei Hindernissen in der Versuchsumgebung die Übertragungsreichweite des Geräts abnimmt.

Um die maximale Entfernung für die Innenübertragung zu messen, habe ich mehrere Messpunkte eingerichtet. Durch Festlegen von Messpunkten, anschließendes Messen des Abstands zwischen jedem Messpunkt und Umwandeln der unregelmäßigen Innenraumumgebung in eine geradlinige Abstandsberechnung kann der geradlinige Abstand zwischen Knoten berechnet werden, d. h. der maximale Übertragungsabstand in Innenräumen.



57Messpunkte Punkt-zu-Punkt-Übertragung

Das orangefarbene Liniensegment in der Abbildung stellt die maximale geradlinige Übertragungsentfernung zwischen Knoten dar und die Entfernung beträgt **28,12 m.**



58Schematisches Diagramm Punkt-zu-Punkt

5.5.2 Übertragung der MeshNetzstruktur

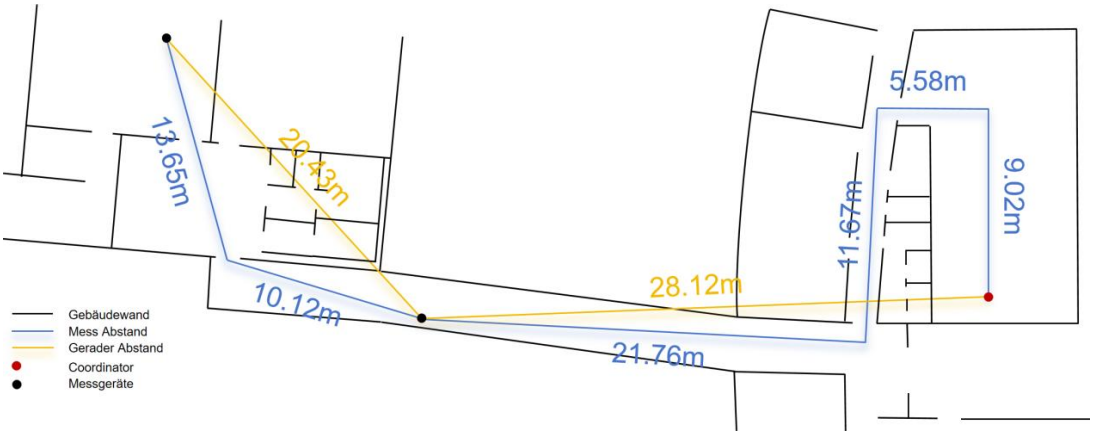
Wenn die Entfernung zwischen dem Knoten und dem Koordinator die Übertragungsentfernung des Systems überschreitet. Wenn sie zu diesem Zeitpunkt eine Datenübertragung realisieren möchten, müssen sie zwischen beiden einen Router als Datenübertragungs-Relaisstation platzieren.

59Messpunkte MeshNetz

Daher habe ich auf der Grundlage des Punkt-zu-Punkt-Übertragungstests des Geräts auch einen Datenübertragungsentfernungstest durchgeführt. Die Testergebnisse sind in der Abbildung dargestellt.

Die Testmethode und der Testprozess stimmen mit der oben genannten Punkt-zu-Punkt-Übertragung überein, und der Durchschnittswert wird aus mehreren Messungen ermittelt. Das endgültige Testergebnis für die Datenübertragungsentfernung beträgt **20,43 m.**

Einerseits haben die endgültigen experimentellen Ergebnisse die Annahme bestätigt, dass das ZigBee-Übertragungssystem eine Datenübertragung realisieren kann, und andererseits hat es auch bestätigt, dass die Zunahme von Hindernissen die Übertragungsentfernung erheblich schwächen wird. Die experimentelle Umgebung für Punkt Die -Punkt-Übertragung auf der linken Seite ist komplizierter, wie aus dem Plan ersichtlich ist. Dazwischen gibt es mehr Hindernisse.



60Schematisches Diagramm MeshNetz

5.6 Kapitelzusammenfassung

Beim Testen der Hardware stand ich vor einigen Herausforderungen, aber durch ständige Anstrengung und Problemlösung habe ich positive Ergebnisse erzielt.

Erstens hatte ich Probleme damit, dass der Sensor nicht schnell genug reagierte. Um dieses Problem zu beheben, habe ich die Funktionsweise und Spezifikationen des Sensors durchgesehen und einige Bereiche gefunden, die optimiert werden könnten. Durch Anpassen der Abtastrate, Erhöhen der Empfindlichkeit des Sensors und Optimieren des Datenverarbeitungsalgorithmus ist es mir gelungen, die Reaktionsgeschwindigkeit des Sensors zu verbessern und sicherzustellen, dass er Änderungen rechtzeitig erkennen kann.

Zweitens habe ich festgestellt, dass die Leistung von ZigBee-Übertragungsgeräten unter bestimmten Umgebungsbedingungen nicht stabil ist. Um dieses Problem zu lösen, haben wir die Testumgebung sorgfältig analysiert und einige mögliche Störquellen gefunden, wie z. B. die LORA-Übertragung und die WLAN-Datenübertragung in der Testumgebung. Beim Experimentieren habe ich unnötige Geräte ausgeschaltet, wodurch Störungen erfolgreich reduziert und die Stabilität und Genauigkeit des Sensors verbessert wurden.

Während des Testprozesses habe ich auch auf die Zusammenarbeit mit meinen Kollegen geachtet, damit ich die Probleme im Testprozess besser verstehen und lösen kann. Dieser Geist der Zusammenarbeit spielt eine wichtige Rolle bei der Lösung komplexer Probleme und der Optimierung der Geräteleistung.

Zusammenfassend lässt sich sagen, dass es mir trotz einiger Herausforderungen während des Hardwaretestprozesses durch kontinuierliche Bemühungen und Problemlösungen gelungen ist, diese zu meistern und zufriedenstellende Ergebnisse zu erzielen. Dieser Test hat nicht nur mein Verständnis der Hardwareleistung und -eigenschaften erweitert, sondern auch wertvolle Erfahrungen und Referenzen für zukünftige Anwendungen und Entwicklungen geliefert.

6 Design und Test von Systemsoftware

6.1 Software und Treiber

Das Softwaredesign des Zigbee-Übertragungssystems basiert auf dem Z-Stack-Protokollstapel für die Programmierung in C-Sprache, und die verwendete Software ist IAR Embedded Workbench für 8051. Dieses System umfasst Koordinatorknoten und Sensorknoten. Nach dem Einschalten des Koordinators wird das Netzwerk eingerichtet und die Sensorknoten erkennen das Netzwerk automatisch und treten dem Netzwerk bei. Der Koordinatorknoten ist für das Senden und Empfangen von Daten sowie die Kommunikation mit dem Gateway verantwortlich, und der Sensorknoten ist für den Empfang und die Weiterleitung von Daten verantwortlich.

6.1.1 Einführung in den Protokollstapel

Z-Stack ist die spezifische Implementierungsform des ZigBee-Protokolls. Im Allgemeinen stellt es eine Schnittstelle zwischen dem ZigBee-Protokoll und dem Benutzer dar. Wir müssen dieses Protokoll über den Protokollstapel verwenden, um die drahtlose Datenübertragung und den drahtlosen Datenempfang zu realisieren. Das ZigBee-Protokoll ist in zwei Teile gegliedert. IEEE802.15.4 definiert die technischen Spezifikationen der physikalischen Schicht und der Medienzugriffsschicht; die ZigBee Alliance definiert die technischen Spezifikationen der Netzwerkschicht und der Anwendungsschicht. Der Z-Stack-Protokollstapel ist eine Sammlung von Protokollen, die von jeder Schicht definiert, in Form von Funktionen implementiert und Benutzern APIs (Anwendungsschichten) bereitgestellt werden. Wir können verwandte APIs aufrufen, die von OSAL für die Multitask-Programmierung bereitgestellt werden, und unsere eigene Anwendung Programme werden als eigenständige Aufgabe implementiert.

Der gesamte Z-Stack verwendet eine mehrschichtige Softwarestruktur, und die Hardware-Abstraktionsschicht (HAL) stellt Treiber für verschiedene Hardwaremodule bereit, darunter Timer-Timer, allgemeinen E/A-Port-GPIO, universellen asynchronen Transceiver-Sender UART und die Anwendung von Analog-zu -Digitale Konvertierung ADC-Programmschnittstelle API, die einen erweiterten Satz verschiedener Dienste bereitstellt. Z-Stack basiert auf der Idee eines Betriebssystems, realisiert die Aufgabenplanung durch die Zeitscheibenrotationsfunktion, bietet einen Multitasking-Verarbeitungsmechanismus und realisiert eine benutzerfreundliche Betriebssystemplattform. Es unterscheidet sich jedoch immer noch stark vom Standardbetriebssystem: Es realisiert nur einige betriebssystemähnliche Funktionen, kann aber nicht im eigentlichen Sinne als Betriebssystem bezeichnet werden.

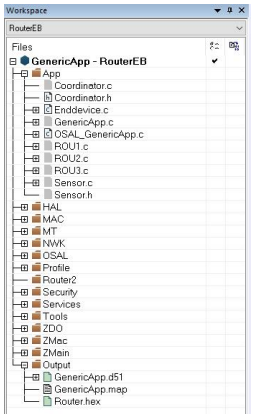
6.1.2 IAR

Das Design der Systemsoftware erfolgt auf der Grundlage des Hardwaredesigns. Ein gutes Softwaredesign ist ein wichtiger Bestandteil der Realisierung von Systemfunktionen und der Schlüssel zur Verbesserung der Systemleistung. Der ZStack-Protokollstack von TI wird in ein Projekt geladen, das auf der IAR-Entwicklungsumgebung basiert. Die leistungsstarke IAR Embedded Workbench bietet

nicht nur Kompilierungs- und Download-Funktionen, sondern kann auch mit einem Programmierer kombiniert werden, um einstufiges Trace-Debugging durchzuführen und On-Chip-Register, Flash-Daten usw. zu überwachen. Z-Stack ist gemäß den Standards IEEE 802.15.4 und ZigBee in die folgenden Schichten unterteilt:

API(Application Programming Interface),HAL(Hardware Abstract Layer),MAC(Media Access Control),NWK(Zigbee Network Layer),OSAL(OperatiSystem Abstract System),Security, Service,ZDO(Zigbee Device Objects).

Nachdem ich die Projektdatei GeneriApp.eww mit IAR geöffnet habe, kann ich die Ordnerverteilung des gesamten Protokollstapels von der HA-Schicht bis zur APP-Schicht anzeigen. Der Protokollstapel kann komplexe Netzwerkverbindungen implementieren und eine nichtflüchtige Speicherung der Routing-Tabelle und der Bindungstabelle im Koordinierungsknoten realisieren, sodass das Netzwerk über eine bestimmte Speicherkapazität verfügt.



61Projektdateien

APP: Verzeichnis der Benutzeranwendungsschicht, der Bereich, in dem Benutzer verschiedene Anwendungsprojekte erstellen, und diesem Verzeichnis können benutzerdefinierte Aufgaben hinzugefügt werden.

HAL: Hardware-Layer-Verzeichnis. Die Common Directory-Datei ist eine allgemeine Datei, die für die Hardware grundsätzlich irrelevant ist. Das Zielverzeichnis ist eine Datei, die sich auf die Hardwareplattform bezieht.

MAC:MAC-Layer-Verzeichnis,High-Level- und Low-Level-Verzeichnisse repräsentieren die oberen bzw. unteren Schichten der MAC-Schicht.

MT: Überwachungs- und Debugging-Verzeichnis, einschließlich der Konfiguration der Netzwerkschicht-Parameterdatei und der Schnittstellendatei der Netzwerkschicht-Bibliotheksfunktion. Diese Verzeichnisdatei wird hauptsächlich für Debugging-Zwecke verwendet.

NWK:Netzwerkschichtverzeichnis, einschließlich Netzwerkschicht-Konfigurationsparameterdateien, Funktionsschnittstellendateien der Netzwerkschichtbibliothek und Funktionsschnittstelle der APS-Schichtbibliothek.

OSAL: Das Betriebssystemverzeichnis, einschließlich einer Datei für einen rotierenden Abfrage-Betriebssystemprotokollstapel.

Profil:AF-Schichtverzeichnis,einschließlich Netzwerkschicht-Parameterdateikonfiguration und Bibliotheksfunktionsschnittstellendateien usw.

Security: Verzeichnis der Sicherheitsschicht, einschließlich Schnittstellendateien für die Verarbeitungsfunktion der Sicherheitsschicht.

Services: Adressverarbeitungsfunktionsverzeichnis, einschließlich der Definition des Adressmodus und der Adressverarbeitungsfunktionsdateien usw.

Tools: Projektkonfigurationsverzeichnis, einschließlich Konfigurationsinformationen zum Protokollstapel.

ZDO: Geräteobjektverzeichnis. Über die Funktion in diesem Verzeichnis kann ich das benutzerdefinierte Objekt verwenden, um den Dienst der APS-Unterschicht und der NWK-Schicht aufzurufen.

ZMac: MAC-Layer-Verzeichnis, einschließlich MAC-Layer-Parameterkonfiguration und Rückrufverarbeitungsfunktion der MAC-Layer-Bibliotheksfunktion.

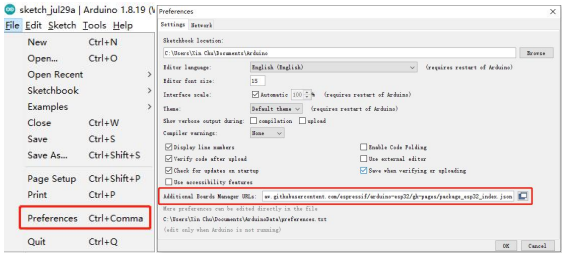
Zmain: Hauptfunktionsverzeichnis, einschließlich der Eingabefunktion main() und der Hardwarekonfigurationsdateien des gesamten Projekts

Output: Das Ausgabedateiverzeichnis, das von der Entwicklungsumgebung IAR automatisch kompiliert und generiert wird.

6.1.3 Arduino IDE

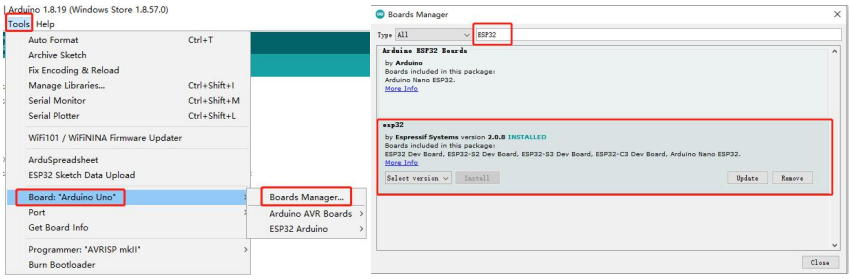
Öffnen die Arduino IDE, klicken auf Datei -> Einstellungen -> Adresse des anderen Entwicklungsboard-Managers. Fügen die Adresse von ESP32 hinzu：

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



62Arduino-Setup

Klicken links auf den Entwicklungsboard-Manager, um nach ESP32 zu suchen, suchen es und klicken auf Installieren.



63Option für ESP32 hinzufügen

Klicken auf Tools->Board->ESP32->Eigenes Modell auswählen

6.2 Einführung in die Systemsoftwarefunktionen

Der Schwerpunkt und die Schwierigkeit des gesamten Systems besteht darin, Feuermannn Hilfe zu leisten. Wie erhält die Kommandozentrale Informationen am Brandort, welche Informationen sollten gesammelt werden, wie kann der Status von Knoten schnell beurteilt werden und wie Feuermann können die Dynamik schnell erfassen der Status jedes Knotens? Status, wie die Kommandozentrale schnell eine Informationsübertragung mit Feuermannn herstellen und Feuermann am Brandort warnen kann. Wenn der Knoten die Verbindung verliert, wie kann der Knoten neu gestartet werden, um sich wieder dem Netzwerk anzuschließen und weiterhin Daten zu senden? Und wie man den Knoten nach Abschluss der Aufgabe möglichst bequem als Erkennungsgerät wiederverwendet.

6.2.1 Zigbee-Übertragungssystem

6.2.1.1 Aufbau eines Mesh-Netzwerks

Gateway(Koordinator)

Gemäß den funktionalen Anforderungen des Koordinatorknotens wird in der Anwendungsschicht des Knotens eine Aufgabe GenericApp definiert, um die Datenerfassungs- und Kommunikationsfunktionen der Benutzerschicht abzuschließen. Kommunikationsereignisse für serielle Ports und drahtlose Kommunikationsereignisse werden jeweils in Benutzeraufgaben definiert. Das Kommunikationsereignis über die serielle Schnittstelle vervollständigt hauptsächlich die Datenkommunikation mit dem Gateway.Bei drahtlosen Kommunikationsereignissen handelt es sich hauptsächlich um die gegenseitige Kommunikation zwischen Knoten, einschließlich gesammelter Daten wie Temperatur und Rauch, Steuerbefehlen, Verbindungsstatuserkennung sowie Knotenrouting- und Topologieinformationen. Die Kommunikation zwischen dem Koordinatorknoten und jedem Knoten erfolgt über eine Punkt-zu-Punkt-Übertragung, und das Endgerät kann mit dem Koordinator oder mit anderen Endknoten kommunizieren.Um diese Funktion zu realisieren, muss der Koordinator die Netzwerkadresse jedes Knotens kennen, was erfordert, dass jeder Knoten nach dem Beitritt zum Netzwerk die Netzwerkadresse an den Koordinator sendet. Nach Erhalt der Netzwerkadresse erstellt der Koordinator eine Adresstabelle und speichert sie Daher ist es für Benutzer erforderlich, Daten gemäß der Adresstabelle zu sammeln, um die Daten jedes Sensors zu sammeln, was die Kommunikation zwischen dem Koordinatorknoten und dem Sensorknoten erleichtert.Die maximale Länge des IEEE802.15.4MAC-Datenpakets beträgt 127 Bytes, und jede Zahl besteht aus Header-Byte und 16 CRC-Werten. Bei der Datenübertragung wird der Antwortdatenübertragungsmechanismus verwendet. Wenn das ACK-Flag-Bit auf 1 gesetzt ist, ist dies der Fall vom Empfänger beantwortet. Wenn innerhalb einer bestimmten Zeitspanne keine Antwort eingeht, ist ein Beweis dafür, dass am Sammelknoten ein Fehler aufgetreten ist.

Der erste Schritt erfordert, dass der Koordinatorknoten die Initialisierung funktionaler Programme wie der seriellen Schnittstelle und des Netzwerkbetriebssystems abschließt, einschließlich der Initialisierung des Betriebssystems, der Initialisierung der

seriellen Schnittstelle und der Hardware-Initialisierung. Die Initialisierung wird durch die Funktionen osal_init_system(), MT_UartInit() bzw. HAL_BOARD_INIT() durchgeführt.

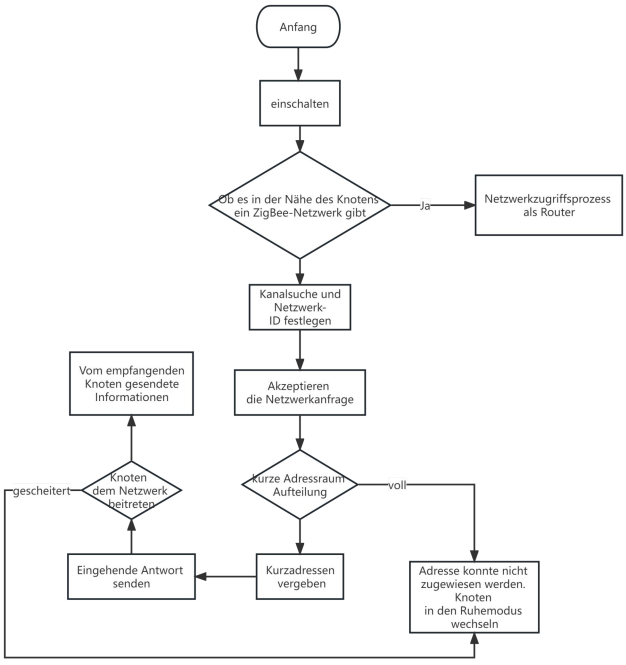
Der zweite Schritt besteht darin, das 2,4-GHz-IEEE802.15-Frequenzband zu scannen, um einen freien Kanal unter den 16 Kanälen zu finden, um durch Kanalscannen ein neues Netzwerk aufzubauen. Der Kanalscanvorgang wird durch die Funktion MAC_MlmeseanReq(maeMlmeSCanReq_t *pData) abgeschlossen.

Der Koordinator sendet durch aktives Scannen einen Beacon-Anforderungsbefehl (Beacon-Anforderungsbefehl) und legt einen Scan-Zeitraum (T_scan_duration) fest. Wenn innerhalb des Zeitlimits kein Antwort-Beacon erkannt wird, wird davon ausgegangen, dass sich kein anderer Koordinator in seiner Reichweite befindet Zu diesem Zeitpunkt kann das Gerät ein eigenes ZigBee-Netzwerk aufbauen und als Koordinator des Netzwerks fungieren.

Der dritte Schritt besteht darin, ein Netzwerk einzurichten. Wenn der Kanal erfolgreich gescannt wurde, wird das Netzwerk durch Aufrufen der Funktion NLME_NetworkDisoveryRequest() eingerichtet.

Im vierten Schritt muss auf dem ausgewählten Kanal die Netzwerk-ID (PAN-ID) eindeutig sein und darf nicht mit anderen ZigBee-Netzwerken in Konflikt geraten. Sie können die festgelegte PAN-ID verwenden oder zufällig eine ID-Nummer auswählen, die nicht in Konflikt gerät, indem Sie die IDs anderer Netzwerke überwachen. Wenn Routing-Knoten oder -Geräte in das Netzwerk gelangen, weist der Koordinator den Knoten kurze Adressen für die Kommunikation zu. Für den Koordinator ist die Netzwerkadresse immer 0x0000.

Der fünfte Schritt ist der Datenempfang. Nachdem der Knoten erfolgreich dem Netzwerk beigetreten ist, wechselt das gesamte drahtlose Sensornetzwerk in den Brandüberwachungszustand.



64Aufbau eines Mesh-Netzwerks(Coo.)

Router-Konten(MESS-Knoten)

Sensorknoten werden hauptsächlich dazu verwendet, Felddaten zu sammeln und an den Koordinator zu senden. Gleichzeitig kann es auch Befehle vom Koordinator empfangen: zum Beispiel den Summer ein- oder ausschalten, die Abholzeit anpassen usw. Die Daten werden alle 10 Sekunden erfasst.

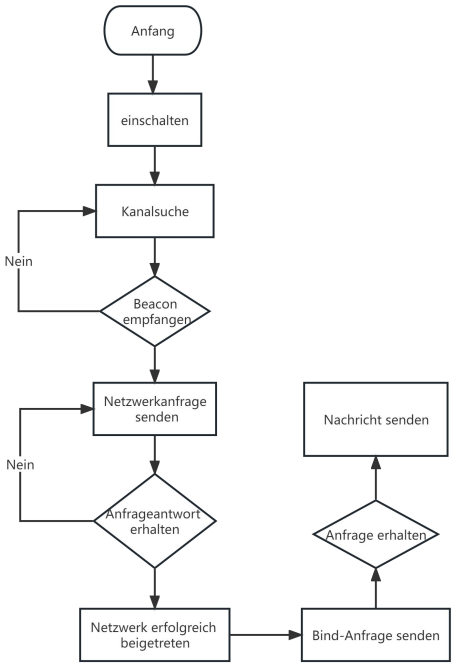
Nachdem der Koordinatorknoten das Netzwerk eingerichtet hat, kann der Sensorknoten mit dem Beitritt zum eingerichteten Netzwerk beginnen. Schließen Sie zunächst die Initialisierung des Betriebssystems und der seriellen Schnittstelle des Knotens ab, dann können Sie dem Netzwerk beitreten, indem Sie über die entsprechende Funktion eine Anfrage zum Beitritt zum Netzwerk initiieren, und schließlich müssen Sie den Sensorknoten und den Koordinatorknoten binden. Der spezifische Vorgang ist wie folgt:

Der erste Schritt besteht darin, das Betriebssystem und den seriellen UART-Port jedes Knotens des Sensors zu initialisieren, der durch Aufrufen von osal_init_system(), HalUARTInit() und anderen Funktionen initialisiert werden muss.

Wenn im zweiten Schritt ein Netzwerk gefunden wird, gibt die Netzwerkschicht dem Sensorknoten ZDO-Schicht-Erkennungsnetzwerk-Feedback-Informationen. Um eine Anwendung zum Beitritt zum Netzwerk über die Netzwerkschicht zu initiieren, muss die Funktion NLMENetworkFormationRequest() aufgerufen werden, um dem Netzwerk beizutreten.

Im dritten Schritt muss die Bindung zwischen dem Sensorknoten und dem Koordinatorknoten hergestellt werden. Nachdem der Sensorknoten die Antwort auf den Beitritt zum Netzwerk erhalten hat, kann er die Funktion ZDP-EndDeviceBindReq() aufrufen, um eine Bindung anzufordern. Nach der Bindung abgeschlossen ist, können die Daten gesendet und empfangen werden. übernehmen.

Der vierte Schritt besteht darin, Daten zu senden. Sie müssen Ereignisse registrieren, die Anzahl und die Sendezeit festlegen usw., wobei die Funktion GenericApp_TaskID die Funktion für das Registrierungsereignis, die Funktion GenericApp_ClusterList die Funktion für die Einstellungsnummer und die Funktion osal_start_timerEx(uint8 taskID) ist , uint16 event_id, uint16 timeout_value) sind die regelmäßigen Sendedaten. Stellen Sie die Zeit ein. Wenn das registrierte Ereignis eintritt, muss zum Senden von Informationen die Funktion AF DataRequest aufgerufen werden. Bei dieser Funktion handelt es sich um eine Strukturfunktion, die die hardwarebezogenen Funktionen im Protokollstapel aufruft und schließlich die Daten über die Antenne sendet.



65Aufbau eines Mesh-Netzwerks(MESS)

End-Knoten(MANN Knoten)

Der Softwareprozess ist der gleiche wie der des Routerknotens, daher wird die Erklärung hier nicht wiederholt

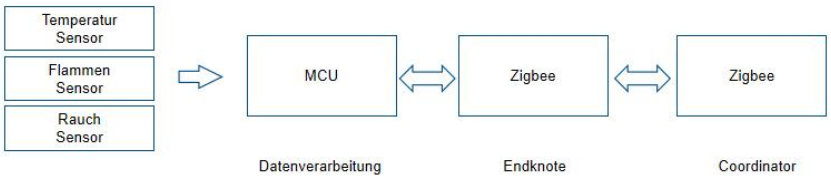
6.2.1.2 Benutzerdefinierte Zigbee-Funktion

Nachdem der Koordinator und jeder Endknoten die Anfangseinstellungen abgeschlossen haben, können sie mit der Ausführung der benutzerdefinierten Programmfunktion beginnen. Um das Verständnis dieses Teils zu erleichtern, werde ich den spezifischen Code kombinieren, um die spezifische Verwendung jeder Funktion zu erläutern.

Senden und Empfangen von Sensordaten

Die Sensordaten werden zunächst von ESP32 gesammelt und zusammengefasst und dann über die serielle Schnittstelle an das Zigbee-Terminal gesendet. Zigbee liest die von der seriellen Schnittstelle gesendeten Daten und sendet sie dann an den Koordinator.

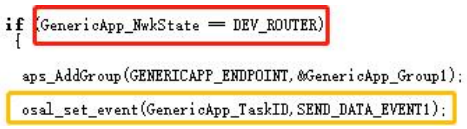
Der Datenübertragungsprozess ist in der Abbildung dargestellt：



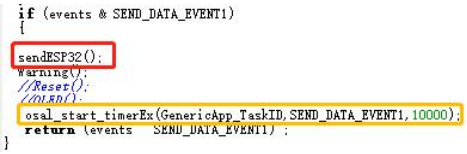
66Informationsübertragungsprozess

Datenversand

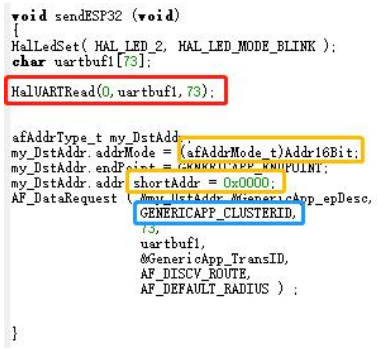
Zunächst beurteilt der Endknoten seinen eigenen Knotentyp (rotes Kästchen). Wenn es sich um einen Sensorknoten (Routingknoten) handelt, generiert er Ereignis 1 (oranges Kästchen).



Ereignis eins enthält die Funktion zum Senden von Sensordaten. Wenn die Ereignispriorität auf Ereignis 1 umgestellt wird, beginnt die Verarbeitung der Funktion zum Senden von Sensordaten (rotes Feld). Der Inhalt im gelben Feld ist die Triggerereignisfunktion. Ereignis 1 wird alle zehn Sekunden ausgelöst, dh gesendet Knotensensordaten alle zehn Sekunden an den Koordinator.

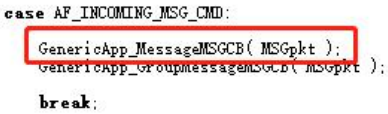


Lesen MESS-Knoten bei der Funktion zum Senden von Sensordaten zuerst die serielle Schnittstelle (rotes Feld), speichern sie die Sensordaten im Speicher und senden sie dann den gespeicherten Inhalt über Unicast (oranges Feld) über die oben erwähnte AF DataRequestl-Funktion an den Koordinator (The Die Netzwerkadresse des Koordinators ist 0x0000). Die Funktion enthält auch das Triggerereignis in der Ereignisfunktion zum Empfangen von Informationen, das Ereignis GENERICAPP_CLUSTERID (blaues Feld).



Datenempfang

Wenn der Koordinator die vom Endknoten gesendeten Daten empfängt, verwendet er zunächst die Funktion osal_msg_receiv (), um die Nachricht aus der Nachrichtenwarteschlange zu empfangen, und ruft dann die Funktion GenericApp_MessageMSGCB () (rotes Feld) auf, sodass sie vom empfangen werden muss GenericApp_MessageMSGCB()-Funktion Die empfangenen Daten werden über die serielle Schnittstelle an das Gateway gesendet.



In der oben erwähnten AF DataRequest()-Funktion ist der Triggertyp das GENERICAPP_CLUSTERID-Ereignis (rotes Kästchen) in der Empfangsinformationsereignisfunktion, und seine Verwendung wird hier gezeigt. Nachdem der Koordinator das Ereignis GENERICAPP_CLUSTERID ausgelöst hat, verwendet Koordinator die Funktion osal_memcpy() (Oranges Feld), um die empfangenen Daten in das Pufferarray zu kopieren und die Daten dann über die serielle Schnittstelle an das Gateway zu senden.

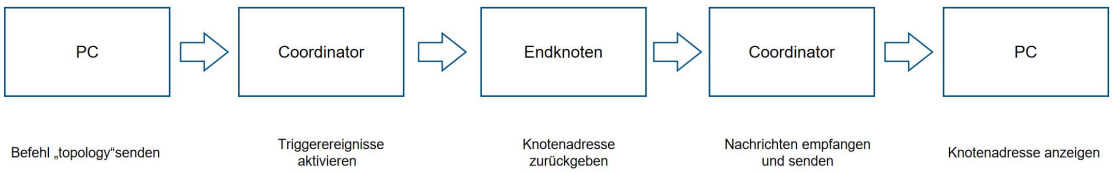
```
void GenericApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  RFTX1 rftx1;
  char buf1[73];
  char buf6[8];

  switch ( pkt->clusterId )
  {

    case GENERICAPP_CLUSTERID:
      HalLedSet( HAL_LED_2, HAL_LED_MODE_BLINK );
      osal_memcpy (buf1, pkt->cmd.Data, 73);
      HalUARTWrite(0, buf1, 73);
      HalUARTWrite(0, "\n", 1);
    break;
```

Die endgültige Zeichenfolge, die von der seriellen PC-Schnittstelle erhalten wird, lautet wie folgt: eine Zeichenfolge mit der Länge 73, einschließlich des Knotennamens und der Sensordaten.

```
[2023-07-27 11:40:31.747]
&&MES0001$ Zahl 000100 Temp -00127 Fire 003885 Rauch 003885 Co 000000 $$
[2023-07-27 11:40:32.284]
&&MES0002$ Zahl 000101 Temp -00127 Fire 003889 Rauch 003889 Co 000000 $$
[2023-07-27 11:40:32.830]
&&MAN0001$ Zahl 000106 Temp -00127 Fire 003851 Rauch 003851 Co 000000 $$
[2023-07-27 11:40:33.392]
&&MES0003$ Zahl 000102 Temp -00127 Fire 003883 Rauch 003883 Co 000000 $$
```

Anzeige der Netzwerktopologie

Diese Funktion verwendet die Rückruffunktion der seriellen ZigBee-Schnittstelle, um das entsprechende Ereignis auszulösen, indem sie den von der seriellen Schnittstelle empfangenen Zeicheninhalt beurteilt.



67Prozess zur Anzeige der Netzwerktopologie

Im Koordinator:

Zunächst aktiviere ich die Rückruffunktion (rotes Kästchen) in der Aufgabeninitialisierungsfunktion auf Benutzerebene.

```
void GenericApp_Init( byte task_id )
{
  halUARTCfg_t uartConfig;
  GenericApp_TaskID           = task_id;
  GenericApp_TransID          = 0;
GenericApp_epDesc.endPoint    = GENERICAPP_ENDPOINT;
  GenericApp_epDesc.task_id   = &GenericApp_TaskID;
  GenericApp_epDesc.simpleDesc = (SimpleDescriptionFormat_t *)&GenericApp_SimpleDesc;
  GenericApp_epDesc.latencyReq = noLatencyReqs;
 afRegister( &GenericApp_epDesc );

 uartConfig.configured         = TRUE;
 uartConfig.baudRate           = HAL_UART_BR_115200;
 uartConfig.flowControl        = FALSE;
 uartConfig.callBackFunc       = rxCB;
 HalUARTOpen (0, &uartConfig);
```

Lesen Koordinator in der Rückruffunktion zuerst die serielle Schnittstelle (rotes Kästchen), passen die Triggerereigniszeichenfolge als „Topologie“ an und beurteilen dann mithilfe der Funktion osal_memcmp (), ob der Inhalt der empfangenen Zeichenfolge mit der Triggerereigniszeichenfolge übereinstimmt (blaues Kästchen). Abschließend wird die Trigger-Ereigniszeichenfolge per Broadcast an jeden Knoten im Netzwerk gesendet (oranges Feld). Die Funktion AF DataRequest() enthält das Triggerereignis in der Ereignisfunktion „Informationen empfangen“ als Ereignis GENERICAPP_FASONG (Blackbox).

```
static void rxCB (uint8 port,uint8 event)
{

uint8 buf[8];
uint8 uartbuf[16];
uint8 i = 0;
HalUARTRead (0, buf, 8);

afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)AddrBroadcast;
my_DstAddr.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr = 0xffff;

afAddrType_t my_DstAddr_Group1;
my_DstAddr_Group1.addrMode = (afAddrMode_t)AddrGroup;
my_DstAddr_Group1.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr_Group1.addr.shortAddr = GenericApp_Group1.ID;

afAddrType_t my_DstAddr_Group2;
my_DstAddr_Group2.addrMode = (afAddrMode_t)AddrGroup;
my_DstAddr_Group2.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr_Group2.addr.shortAddr = GenericApp_Group2.ID;



if(osal_memcmp (buf,"topology",8))
{
unsigned char *theMessageData = "topology";

AF_DataRequest (&my_DstAddr,&GenericApp_epDesc,
                GENERICAPP_FASONG,
                osal_strlen(theMessageData)+1,
                theMessageData,
                &GenericApp_TransID,
                AF_DISCV_ROUTE,
                AF_DEFAULT_RADIUS );
}
```

Im Endknoten:

Der Informationsempfangsprozess des Endknotens stimmt mit dem oben genannten „Datenempfang“ überein.

Der Knoten löst die Informationen aus, um das Ereignis GENERICAPP_FASONG zu empfangen, vergleicht dann die empfangenen Informationen mit der String-Topologie und führt dann die Funktion SendInfo() aus.

```
void GenericApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  char buf1[9];
  char buf2[5];
  char buf3[6];
  char buf6[11];
  switch ( pkt->clusterId )
{
  case FEEDBACK:
      osal_memcpy (buf6, pkt->cmd.Data, osal_strlen("Statecheck")+1);
    if(osal_memcmp(buf6, "Statecheck", osal_strlen("Statecheck")+1))
    {
    Feedback1();
    Feedback2();
    n=0;
    }
    break;

  case GENERICAPP_FASONG:
   osal_memcpy(buf1, pkt->cmd.Data, osal_strlen("topology")+1);

    if(osal_memcmp(buf1, "topology", osal_strlen("topology")+1))
    {
   SendInfo();
    }
    break;
}
```

Ich rufe in der SendInfo()-Funktion die NLME-Funktion GetShortAddr() auf, um die Netzwerkadresse des Knotens zu erhalten, da der Rückgabewert der NLME-Funktion GetShortAddr() (oranges Feld) die Netzwerkadresse des Knotens ist, also kann es sein direkt einer Variablen zugewiesen. Dann verwende ich die Funktion To_string(), um die Netzwerkadresse in hexadezimaler Form an die serielle Schnittstelle auszugeben. Ich

rufe die Funktion NLMEGetCoordShortAddr () auf, um die Netzwerkadresse des übergeordneten Knotens abzurufen.Dann verwende ich die Funktion AF DataRequest() zum Senden an den Koordinator im Unicast-Modus (blaues Feld), das das Auslöseereignis in der Ereignisfunktion zum Empfangen von Informationen als GENERICAPP_FASONG-Ereignis (schwarzes Feld) enthält.

```
void SendInfo(void)
{
RFTX2 rftx2;
uint16 nwk ;
//uint8 buf1[8];

osal_memcpy (rftx2.type,"MESS1",5) ;

nwk =NLME_GetShortAddr() ;
to_string (rftx2.myNWK,(uint8 *)&nwk, 2);

//To_string (rftx2.myMAC,NLME_GetExtAddr(), 8);

nwk = NLME_GetCoordShortAddr() ;
to_string (rftx2.pNWK, (uint8 *)&nwk,2);
//NLME_GetCoordExtAddr(buf1);
//To_string (rftx2.pMAC, buf1, 8);
afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)Addr16Bit;
my_DstAddr.endPoint = GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr = 0x0000;
AF_DataRequest ( &my_DstAddr, &genericApp_epDesc,
                 GENERICAPP_FASONG,
                 13,
                 (uint8 *)&rftx2,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS ) ;
}
```

Wenn im Projektentwicklungsprozess Datenpakete verwendet werden, werden im Allgemeinen Strukturen verwendet, um die für das gesamte Datenpaket erforderlichen Daten zu enthalten, sodass die Programmiereffizienz höher ist. Hier ist RFTX2 (rotes Feld) die benutzerdefinierte Struktur.

```
typedef  struct RFTXBUF2
  {
uint8 type[5] ;
uint8  myNWK[4];
//uint8  myMAC[16];
uint8  pNWK[4] ;
//uint8  pMAC[16] ;
}RFTX2;
```

Wieder zurück im Koordinator:

In Übereinstimmung mit dem obigen Datenempfangsprozess löst dieses Mal die Funktion GenericApp_GroupMessageMSGCB() (rotes Feld) aus, löst dann das Ereignis GENERICAPP_FASONG aus und verwendet dann die Funktion osal_memcpy() (oranges Feld), um die empfangenen Daten in die benutzerdefinierte RFTX2-Struktur zu kopieren. Anschließend werden die Kurzadresse jedes Knotens und die Kurzadresse des übergeordneten Knotens über die Ausgabe des seriellen Ports abgerufen.

```
case AF_INCOMING_MSG_CMD:
    GenericApp_MessageMSGCB( MSGpkt );
    GenericApp_GroupMessageMSGCB( MSGpkt );
    break;
```

```
void GenericApp_GroupMessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  RFTX2 nodeinfo;

  char buf2[14];
  char buf3[14];
  char buf4[14];
  char buf5[14];

  switch ( pkt->clusterId )
  {
    case GENERICAPP_FASONG:
    osal_memcpy(&nodeinfo,pkt->cmd.Data,13);
    HalUARTWrite(0," Type: ",6);
    HalUARTWrite(0,nodeinfo.type,5);
    HalUARTWrite(0," NWK: ",6);
    HalUARTWrite(0,nodeinfo.myNWK,4);
    //HalUARTWrite(0," MAC: ",6);
    //HalUARTWrite(0,nodeinfo[i].myMAC,16);
    HalUARTWrite(0," pNWK: ",7);
    HalUARTWrite(0,nodeinfo.pNWK,4);
    //HalUARTWrite(0," pMAC: ",7);
    //HalUARTWrite(0,nodeinfo[i].pMAC,16);
    HalUARTWrite(0,"\n",1);

    break;
  }
}
```

Die endgültige Zeichenfolge, die von der seriellen PC-Schnittstelle erhalten wird, lautet wie folgt und enthält den Knotentyp, die Kurzadresse des Knotens und die Kurzadresse des übergeordneten Knotens.

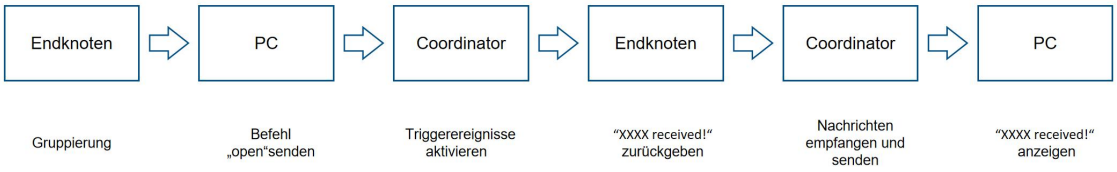
```
Type: MANN1 NWK: 0F68 pNWK: 019C
Type: MESS2 NWK: 019C pNWK: 01A0
Type: MESS1 NWK: 01A0 pNWK: 0000
Type: MESS3 NWK: 5E9D pNWK: 0F68
```

Unicast, Broadcast, Multicast

Unicast und Broadcast werden jeweils in den beiden oben vorgestellten Funktionen verwendet und die Funktionsprinzipien von Unicast und Broadcast wurden ausführlich erläutert, sodass die Einführung hier nicht wiederholt wird. Daher konzentriert sich dieser Teil auf die Multicast-Methode und die durch diese Methode realisierte Knotengruppenalarmfunktion.

Um die Multicast-Funktion zu realisieren, muss dem System mitgeteilt werden, dass jeder Knoten im Netzwerk eine Gruppierung durchgeführt hat und die Gruppierung kennt.

Wenn ich Multicast zum Senden von Daten verwende, muss ich einer bestimmten Gruppe beitreten. Das Problem, das jetzt gelöst werden muss, ist: Wie stellt man eine Gruppe dar und wie bringt man Knoten dazu, der Gruppe beizutreten?



68Prozess zur Multicast-Funktion

Zunächst werden zwei Variablen GenericApp_Group1 und GenericApp_Group2 vom Typ aps Group_t definiert.

```
aps_Group_t GenericApp_Group1;
aps_Group_t GenericApp_Group2;
```

In der Datei apsgruops.h gibt es eine Definition der Struktur aps_Group_t wie folgt:

```
typedef struct
{
  uint16 ID;                      // Unique to this table
  uint8  name[APS_GROUP_NAME_LEN]; // Human readable name of group
} aps_Group_t;
```

Jede Gruppe hat eine bestimmte ID und dann den Gruppennamen, der im Namensarray gespeichert ist.

Ich initialisiere die Gruppen-IDs in der Userland-Task-Initialisierungsfunktion auf 0x0001 und 0x0002.Dann verwende ich die Funktion osal memcpy(), um die Gruppennamen „Gruppe1“ und „Gruppe2“ in das Namensarray zu kopieren.

```
void GenericApp_Init( byte task_id )
{
  halUARTCfg_t uartConfig;
  GenericApp_TaskID            = task_id;
  GenericApp_TransID           = 0;
GenericApp_epDesc.endPoint     = GENERICAPP_ENDPOINT;
  GenericApp_epDesc.task_id    = &GenericApp_TaskID;
  GenericApp_epDesc.simpleDesc = (SimpleDescriptionFormat_t *)&GenericApp_Sim
  GenericApp_epDesc.latencyReq = noLatencyReqs;
afRegister( &GenericApp_epDesc );

uartConfig.configured          = TRUE;
uartConfig.baudRate            = HAL_UART_BR_115200;
uartConfig.flowControl         = FALSE;
uartConfig.callBackFunc        = rxCB;
HalUARTOpen (0, &uartConfig);

GenericApp_Group1.ID=0x0001;
GenericApp_Group2.ID=0x0002;

osal_memcpy(&(GenericApp_Group1.name[1]), "Group1", 6);
osal_memcpy(&(GenericApp_Group2.name[1]), "Group2", 6);
}
```

Dadurch können Knoten mithilfe der aps-Funktion AddGroup() zu Gruppen hinzugefügt werden:

aps_AddGroup(GENERICAPP_ENDPOINT,&GenericApp_Group1);

aps_AddGroup(GENERICAPP_ENDPOINT,&GenericApp_Group2);

Im Koordinator:

Zuerst lese ich die serielle Schnittstelle mit der oben vorgestellten Rückruffunktion aus. Hier ist ein Beispiel für das Einschalten des Summeralarms des Sensorknotens. Die benutzerdefinierte Trigger-Ereigniszeichenfolge ist „offen“. Ich verwende die Funktion osal_memcmp (), um zu beurteilen, ob der Inhalt der empfangenen Zeichenfolge mit der Trigger-Ereigniszeichenfolge (oranges Feld) übereinstimmt. Schließlich wird die Trigger-Ereigniszeichenfolge per Multicast (rotes Kästchen) an Knoten der Gruppe 1 im Netzwerk gesendet. Die Funktion AF DataRequest() enthält das Triggerereignis in der Ereignisfunktion zum Empfangen von Informationen, das Ereignis GENERICAPP_HUISHOU1 (blaues Feld).

```
static void rxCB (uint8 port,uint8 event)
{

uint8 buf[8];
uint8 uartbuf[16];
uint8 i = 0 ;
HalUARTRead (0,buf,8);

afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)AddrBroadcast;
my_DstAddr.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr = 0xffff;

afAddrType_t my_DstAddr_Group1;
my_DstAddr_Group1.addrMode = (afAddrMode_t)AddrGroup;
my_DstAddr_Group1.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr_Group1.addr.shortAddr = GenericApp_Group1.ID;

afAddrType_t my_DstAddr_Group2;
my_DstAddr_Group2.addrMode = (afAddrMode_t)AddrGroup;
my_DstAddr_Group2.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr_Group2.addr.shortAddr = GenericApp_Group2.ID;

if(osal_memcmp (buf,"open",4))
{
  unsigned char *theMessageData = "open";

AF_DataRequest ( &my_DstAddr_Group1,&GenericApp_epDesc,
                 GENERICAPP_HUISHOU1,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS );
}
```

Im Endknoten:

Erstens sind die Betriebsschritte dieselben wie im Koordinator.Ich definiere den Gruppennamen und die Gruppen-ID in der Aufgabeninitialisierungsfunktion der Benutzerschicht und füge dann die Funktion aps AddGroup() in der Knotentypbeurteilung hinzu, um den Knoten zur Gruppe hinzuzufügen 1.

```
if (GenericApp_NwkState == DEV_ROUTER)
 {
 aps_AddGroup(GENERICAPP_ENDPOINT, &GenericApp_Group1);
  osal_set_event(GenericApp_TaskID, SEND_DATA_EVENT1);
 }
```

Der Knoten beurteilt zunächst die Gruppen-ID, löst dann den Empfang der Nachricht durch das Ereignis GENERICAPP_HUISHOU1 aus, vergleicht dann die empfangene Nachricht mit der Zeichenfolge open und führt dann die Funktion GenericApp_SendTheMessage () aus.

```
void GenericApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  char buf1[9];
  char buf2[5];
  char buf3[6];
  char buf6[11];
  switch ( pkt->clusterId )
{
 if(0x0001 == pkt->groupId)
 {
    if(GENERICAPP_ENDPOINT == pkt->endPoint)
    {
  switch ( pkt->clusterId )
  {
  case GENERICAPP_HUISHOU1:
    osal_memcpy(buf2, pkt->cmd.Data, osal_strlen("open")+1);
    if(osal_memcmp(buf2, "open", osal_strlen("open")+1))
    {
   GenericApp_SendTheMessage();
    }
    break;
```

Ich rufe in der Funktion GenericApp_SendTheMessage() die Funktion HAL_TURN_ON_LED1() (rotes Feld) auf, damit der mit dem Summer verbundene Pin einen hohen Pegel erzeugt und den Summer alarmiert. Ich verwende die Funktion AF DataRequest(), um per Unicast eine Nachricht an den Koordinator zu senden (oranges Feld), um dem Koordinator mitzuteilen, dass der Knoten die Anweisung erhalten hat. Unter diesen ist das auslösende Ereignis in der Funktion des Informationsempfangsereignisses das Ereignis GENERICAPP_HUISHOU1 (blaues Kästchen).

```
void GenericApp_SendTheMessage ( void )
{
  HAL_TURN_ON_LED1();
unsigned char *theMessageData = "MESS1 received";
afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)Addr16Bit;
my_DstAddr.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr = 0x0000;
AF_DataRequest ( &my_DstAddr, &GenericApp_epDesc,
                 GENERICAPP_HUISHOU1,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS );

}
```

Zurück im Koordinator:

Ich löse im Einklang mit dem oben beschriebenen Datenempfangsprozess das Ereignis GENERICAPP_HUISHOU1 (rotes Kästchen) aus und gebe den PC über die serielle Schnittstelle aus, um die Knoten-Feedback-Informationen zu erhalten.

```
void GenericApp_GroupMessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  RFTX2 nodeinfo;

  char buf2[14];
  char buf3[14];
  char buf4[14];
  char buf5[14];


  switch ( pkt->clusterId )
  {

    case GENERICAPP_HUISHOU1:
    osal_memcpy (buf2, pkt->cmd.Data, 14);
    HalUARTWrite(0, buf2, 14);
    HalUARTWrite(0, "\n", 1);
    break;
```

Das Folgende ist ein Beispiel dafür, wie die serielle Schnittstelle des PCs eine Rückmeldung vom MESS-Knoten erhält, nachdem die serielle Schnittstelle den Befehl „open“ gesendet hat.

```
[2023-07-27 07:27:30.419]
MESS1 received!
MESS2 received!
MESS3 received!
```

Feedback zur Knotenübertragung

LED blinkt, um die Signalübertragung anzuzeigen

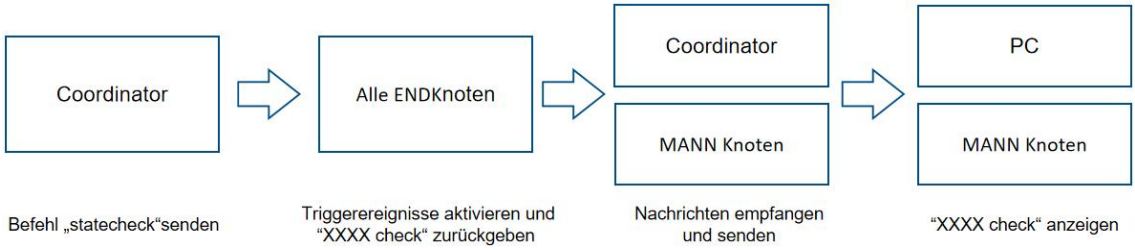
Um die vom Endknoten gesendeten und vom Koordinator empfangenen Daten visuell anzuzeigen, habe ich die LED-Lichtblinkfunktion hinzugefügt. Immer wenn der Endknoten ein Ereignis zum Senden von Sensordaten auslöst, blinkt die mit dem P1_1-Pin verbundene LED einmal. Immer wenn der Koordinator den Empfang von Sensordaten auslöst, blinkt die mit dem P1_1-Pin verbundene LED einmal. Anhand der LED-Leuchten kann ich auch beurteilen, ob das ZigBee-Netzwerk Daten normal überträgt.

```
void sendESP32 (void)
{
HalLedSet( HAL_LED_2, HAL_LED_MODE_BLINK );
char uartbuf1[73];

HalUARTRead(0, uartbuf1, 73);


afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)Addr16B
my_DstAddr.endPoint = GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr = 0x0000;
AF_DataRequest ( &my_DstAddr, &GenericApp_ep
                 GENERICAPP_CLUSTERID,
                 73,
                 uartbuf1,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS ) ;

}
```

```
void GenericApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  RFTX1 rftx1;
  char buf1[73];
  char buf6[8];

  switch ( pkt->clusterId )
  {

    case GENERICAPP_CLUSTERID:
      HalLedSet( HAL_LED_2, HAL_LED_MODE_BLINK );
      osal_memcpy (buf1, pkt->cmd.Data, 73);
      HalUARTWrite(0, buf1, 73);
      HalUARTWrite(0, "\n", 1);
    break;
```

Beurteilung des Knotenverbindungsstatus:

Damit der Host-Computer und die Feuerwehrknoten den Verbindungsstatus des gesamten Netzwerks anzeigen können, habe ich eine Funktion zur Beurteilung des Knotenverbindungsstatus entwickelt. Das Arbeitsprinzip besteht darin, dass der Koordinator alle 15 Sekunden Statusabfrageinformationen an alle Endknoten sendet. Wenn der Knoten die entsprechende Nachricht normal beantworten kann, bedeutet dies, dass die Verbindung zum Knoten nicht unterbrochen ist. Wenn der Koordinator innerhalb einer Minute keine Antwortnachricht von einem Knoten erhält, bedeutet dies, dass die Verbindung zum Knoten aus irgendeinem Grund getrennt wurde. Mit dieser Funktion können sowohl der Host-Computer als auch der Firefighter-Knoten überprüfen, welcher Knoten verloren geht.

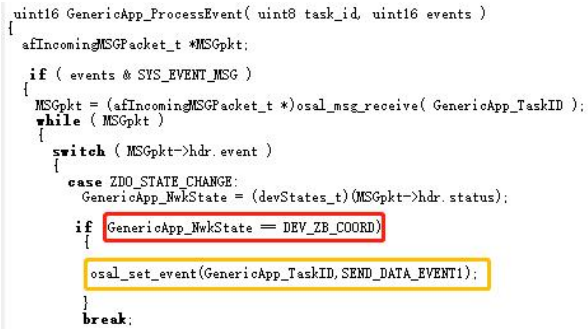


69Prozess zur Beurteilung des Knotenverbindungsstatus:

Im Koordinator：

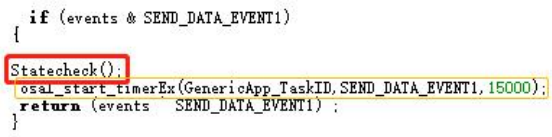
Die Broadcast-Methode wird zum Senden von Statusabfrageinformationen verwendet, daher ähnelt der Code der Funktion zum Anzeigen der Netzwerktopologie oben. Es ist nur so, dass die Funktion der Netzwerktopologie darin besteht, Anfragen unregelmäßig über die serielle Schnittstelle zu senden, und die Funktion zur Beurteilung der Statusverbindung darin besteht, das Sendeereignis für Nachrichten festzulegen und Anfragen regelmäßig zu senden.

Zunächst generiert der Koordinator durch Beurteilung des Gerätetyps Ereignis 1 (oranges Feld).



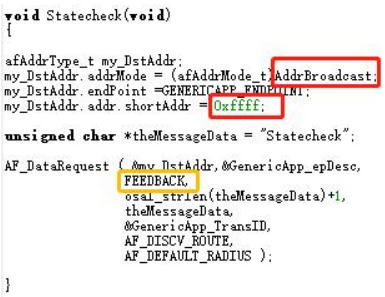
```
uint16 GenericApp_ProcessEvent( uint8 task_id, uint16 events )
{
  afIncomingMSGPacket_t *MSGpkt;

  if ( events & SYS_EVENT_MSG )
  {
    MSGpkt = (afIncomingMSGPacket_t *)osal_msg_receive( GenericApp_TaskID );
    while ( MSGpkt )
    {
      switch ( MSGpkt->hdr.event )
      {
        case ZDO_STATE_CHANGE:
          GenericApp_NwkState = (devStates_t)(MSGpkt->hdr.status);

          if (GenericApp_NwkState == DEV_ZB_COORD)
          {

          osal_set_event(GenericApp_TaskID,SEND_DATA_EVENT1);

          }
          break;
```

Ereignis eins enthält die Funktion Statecheck(). Wenn die Ereignispriorität auf Ereignis 1 umgestellt wird, beginnt die Verarbeitung der Funktion zum Senden von Sensordaten (rotes Feld). Der Inhalt im gelben Feld ist die Triggerereignisfunktion, die alle fünfzehn Sekunden, also alle fünfzehn Sekunden, Ereignis 1 auslöst an alle Der Knoten sendet eine Verbindungsstatus-Erkennungsnachricht.



```
  if (events & SEND_DATA_EVENT1)
  {

  Statecheck();
   osal_start_timerEx(GenericApp_TaskID,SEND_DATA_EVENT1,15000);
   return (events ^ SEND_DATA_EVENT1) ;
  }
```

In der Funktion Statecheck() wird die Funktion AF DataRequestl verwendet, um den gespeicherten Inhalt per Broadcast an jeden Knoten zu senden (rotes Feld). Die Funktion AF DataRequest() enthält das FEEDBACK-Ereignis (oranges Feld) in der Ereignisfunktion zum Empfangen von Informationen.



```
void Statecheck(void)
{

afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)AddrBroadcast;
my_DstAddr.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr =0xffff;

unsigned char *theMessageData = "Statecheck";

AF_DataRequest ( &my_DstAddr,&GenericApp_epDesc,
                 FEEDBACK,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS );

}
```

Im Endknoten:

Nachdem der Knoten die Anweisung empfangen hat, löst er die Informationen zum Empfang des FEEDBACK-Ereignisses aus, vergleicht dann die empfangenen Informationen mit der Zeichenfolge Statecheck und führt dann die Funktionen Feedback1() und Feedback2() aus.

```
void GenericApp_MessageMSGCB( afIncomingMSGPacket_t *pkt )
{
  char buf1[9];
  char buf2[5];
  char buf3[6];
  char buf6[11];
  switch ( pkt->clusterId )
{
  case FEEDBACK:
    osal_memcpy(buf6,pkt->cmd.Data,osal_strlen("Statecheck")+1);
    if(osal_memcmp(buf6,"Statecheck",osal_strlen("Statecheck")+1))
    {
    Feedback1();
    Feedback2();
    }
    break;
```

Die Funktion Feedback1() und die Funktion Feedback2() verwenden beide die Funktion AF DataRequest(), um Antwortinformationen zurückzugeben. Der Unterschied besteht darin, dass die Funktion Feedback1() Unicast verwendet und das Objekt der Koordinator ist. Die Funktion Feedback2() verwendet Multicast und das Objekt ist der Firefighter-Knoten. Die Funktion AF DataRequest() enthält verschiedene auslösende Ereignisse in der Ereignisfunktion zum Empfangen von Informationen.

```
void Feedback1(void)
{

unsigned char *theMessageData = "MESS0001";

afAddrType_t my_DstAddr;
my_DstAddr.addrMode = (afAddrMode_t)Addr16Bit;
my_DstAddr.endPoint = GENERICAPP_ENDPOINT;
my_DstAddr.addr.shortAddr = 0x0000;
AF_DataRequest ( &my_DstAddr,&GenericApp_epDesc,
                 FEEDBACK,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS ) ;

}
```

```
void Feedback2(void)
{

unsigned char *theMessageData = "MESS0001";

afAddrType_t my_DstAddr_Group2;
my_DstAddr_Group2.addrMode = (afAddrMode_t)AddrGroup;
my_DstAddr_Group2.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr_Group2.addr.shortAddr = GenericApp_Group2.ID;
AF_DataRequest ( &my_DstAddr_Group2,&GenericApp_epDesc,
                 LINKZUSTAND1,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS );
}
```

Nachdem der Koordinator die zurückgegebenen Informationen erhalten hat, löst er das entsprechende Ereignis aus und beurteilt den Inhalt der Informationen. Wenn die Rückgabeinformationen vom Knoten empfangen werden, wird die entsprechende Nachricht über die serielle Schnittstelle gesendet, um den nächsten Schritt der Informationsverarbeitung durch ESP32 zu erleichtern. **Der Inhalt der ESP32-Informationsverarbeitung wird im nächsten Teil des Dokuments ausführlich erläutert.**

```
case FEEDBACK:
  osal_memcpy (buf6,pkt->cmd.Data,8);

  if(osal_memcmp(buf6,"MANN0001",8))
{
HalUARTWrite(0,"MANN0001 check",14);
HalUARTWrite(0,"\n",1);
}
  if(osal_memcmp(buf6,"MESS0001",8))
{
HalUARTWrite(0,"MESS0001 check",14);
HalUARTWrite(0,"\n",1);
}
  if(osal_memcmp(buf6,"MESS0002",8))
{
HalUARTWrite(0,"MESS0002 check",14);
HalUARTWrite(0,"\n",1);
}
  if(osal_memcmp(buf6,"MESS0003",8))
{
HalUARTWrite(0,"MESS0003 check",14);
HalUARTWrite(0,"\n",1);
}
break;
```

Der Prozess des Empfangs der Antwortnachricht durch den Feuerwehrknoten stimmt mit dem des Koordinators überein, daher wird die Erklärung hier nicht wiederholt.

```
case LINKZUSTAND1:
  osal_memcpy (buf13,pkt->cmd.Data,osal_strlen("MESS0001")+1);

  if(osal_memcmp(buf13,"MESS0001",osal_strlen("MESS0001")+1)
{
HalUARTWrite(0,"MESS0001 check",14);
}

break;

   case LINKZUSTAND2:
  osal_memcpy (buf14,pkt->cmd.Data,osal_strlen("MESS0002")+1);

  if(osal_memcmp(buf14,"MESS0002",osal_strlen("MESS0002")+1))
{
HalUARTWrite(0,"MESS0002 check",14);
}

break;

   case LINKZUSTAND3:
  osal_memcpy (buf15,pkt->cmd.Data,osal_strlen("MESS0003")+1);

  if(osal_memcmp(buf15,"MESS0003",osal_strlen("MESS0003")+1))
{
HalUARTWrite(0,"MESS0003 check",14);
}

break;
```

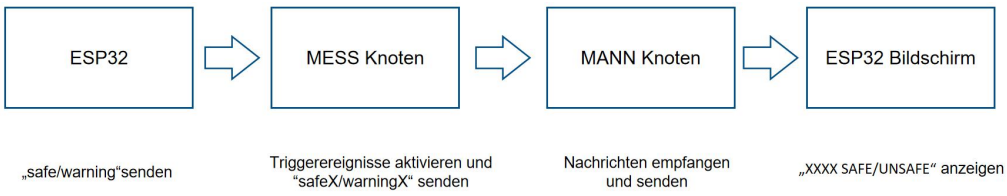
Das Folgende sind die Rückmeldungsinformationen des Endknotens, die der Koordinator erhält, wenn der Endknoten normal verbunden ist.

[2023-07-29 11:25:11.312]MANN0001 checkMESS0003 checkMESS0001 check

Beurteilung des Knotensicherheitsstatus

Ob die Umgebung, in der sich die Sensorknoten befinden, sicher ist, ist auch die Information, auf die sich Feuermann konzentrieren. Damit Feuermann schnell Informationen am Brandort erhalten können, habe ich die Funktion zur Beurteilung des Knotensicherheitsstatus entwickelt.

Das Prinzip besteht darin, dass ESP32 zunächst anhand der erfassten Sensordaten eine vorläufige Beurteilung der Umgebung vornimmt und das Beurteilungsergebnis über die serielle Schnittstelle an das ZigBee-Modul sendet. Nach Erhalt der Sicherheitsstatusinformationen sendet das ZigBee-Modul die entsprechenden Informationen an den Feuerwehrknoten. Auf diese Weise können Feuermann das ZigBee-Übertragungssystem nutzen, um schnell die Dynamik des Brandorts zu erfassen.



70Prozess zur Bewertung des Knotensicherheitsstatus

Die Sicherheitsstatusinformationen werden zusammen mit den Sensordaten an den Feuerwehrknoten gesendet, sodass der Sendezyklus alle zehn Sekunden erfolgt.

Sensorknoten verwenden Multicast, um Informationen an Feuerwehrknoten zu senden. Die Vorgehensweise ist die gleiche wie bei Multicast in anderen Funktionen.

Zuerst liest die serielle Schnittstelle die Sicherheitsstatusinformationen, beurteilt dann den Inhalt der Informationen und verwendet dann die Sendefunktion, um die entsprechenden Informationen per Multicast an den Feuerwehrknoten zu senden. Unterschiedliche Informationen entsprechen unterschiedlichen Auslöseereignissen.

```
void Warning( void )
{
char buf [8];
HalUARTRead (0, buf, 8);

afAddrType_t my_DstAddr_Group2;
my_DstAddr_Group2.addrMode = (afAddrMode_t)AddrGroup;
my_DstAddr_Group2.endPoint =GENERICAPP_ENDPOINT;
my_DstAddr_Group2.addr.shortAddr = GenericApp_Group2.ID;

  if(osal_memcmp (buf,"warning",7))
{

  unsigned char *theMessageData = "warning1";

AF_DataRequest ( &my_DstAddr_Group2, &GenericApp_epDesc,
                 GENERICAPP_MESS11,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS );


}

if(osal_memcmp (buf,"safe",4))
{
  unsigned char *theMessageData = "safe1";

AF_DataRequest ( &my_DstAddr_Group2, &GenericApp_epDesc,
                 GENERICAPP_MESS12,
                 osal_strlen(theMessageData)+1,
                 theMessageData,
                 &GenericApp_TransID,
                 AF_DISCV_ROUTE,
                 AF_DEFAULT_RADIUS );
}

}
```

Nachdem die Feuermann die Multicast-Informationen erhalten haben, führen sie die entsprechende Verarbeitungsfunktion entsprechend dem Auslöseereignis aus. Schließlich wird der Sicherheitsstatus jedes Knotens über die serielle Schnittstelle an ESP32 gesendet und die nachfolgenden Informationen werden von ESP32 verarbeitet. Der Inhalt der ESP32-Informationsverarbeitung wird im nächsten Teil des Dokuments ausführlich erläutert.

```
case GENERICAPP_MESS11:
  osal_memcpy(buf7, pkt->cmd.Data, osal_strlen("warning1")+1);
  if(osal_memcmp(buf7, "warning1", osal_strlen("warning1")+1))
  {
  GenericApp_SendTheMessage7();
  }
  break;

case GENERICAPP_MESS12:
  osal_memcpy(buf8, pkt->cmd.Data, osal_strlen("safe1")+1);
  if(osal_memcmp(buf8, "safe1", osal_strlen("safe1")+1))
  {
  GenericApp_SendTheMessage8();
  }
  break;
case GENERICAPP_MESS21:
  osal_memcpy(buf9, pkt->cmd.Data, osal_strlen("warning2")+1);
  if(osal_memcmp(buf9, "warning2", osal_strlen("warning2")+1))
  {
  GenericApp_SendTheMessage9();
  }
  break;

case GENERICAPP_MESS22:
  osal_memcpy(buf10, pkt->cmd.Data, osal_strlen("safe2")+1);
  if(osal_memcmp(buf10, "safe2", osal_strlen("safe2")+1))
  {
  GenericApp_SendTheMessage10();
  }
  break;
case GENERICAPP_MESS31:
  osal_memcpy(buf11, pkt->cmd.Data, osal_strlen("warning3")+1);
  if(osal_memcmp(buf11, "warning3", osal_strlen("warning3")+1))
  {
  GenericApp_SendTheMessage11();
  }
  break;

case GENERICAPP_MESS32:
  osal_memcpy(buf12, pkt->cmd.Data, osal_strlen("safe3")+1);
  if(osal_memcmp(buf12, "afe3", osal_strlen("safe3")+1))
  {
  GenericApp_SendTheMessage12();
  }
  break;
```

```
void GenericApp_SendTheMessage7 ( void )
{
HalUARTWrite(0, "warning1", 8);
}

void GenericApp_SendTheMessage8 ( void )
{
HalUARTWrite(0, "safe1", 5);
}
void GenericApp_SendTheMessage9 ( void )
{
HalUARTWrite(0, "warning2", 8);
}

void GenericApp_SendTheMessage10 ( void )
{
HalUARTWrite(0, "safe2", 5);
}
void GenericApp_SendTheMessage11 ( void )
{
HalUARTWrite(0, "warning3", 8);
}

void GenericApp_SendTheMessage12 ( void )
{
HalUARTWrite(0, "safe3", 5);
}
```

Die folgende Abbildung zeigt, dass der Feuermannknoten die Rückmeldung zum Sicherheitsstatus von den Sensorknoten 1 und 3 erhält. Zu diesem Zeitpunkt ist die Testumgebung des Sensorknotens ungefährlich

```
[2023-08-19 11:24:52.201]safe1safe3
```

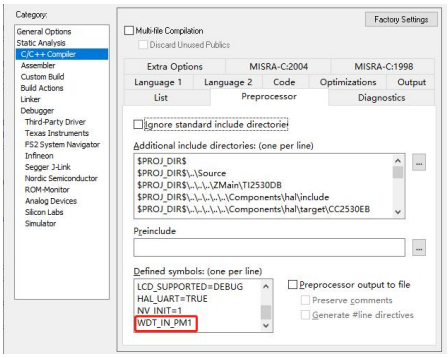
Watchdog-Funktion

Ich verwende die Watchdog-Funktion im Protokollstapel, um ein Zurücksetzen des Systems zu erzwingen und die Funktion des Software-Resets und Neustarts zu realisieren. Da der Kommunikationsfehler auftritt, nachdem das Gerät eine Zeit lang normal funktioniert, liegt ein kleiner Teil der Ursache in der Unterbrechung der Verbindung, die durch die Entfernungsänderung verursacht wird, und der größere Grund besteht darin, dass der Programmfehler dazu führt, dass das Programm nicht mehr durchläuft trennen, also nur Es muss ein Reset-Zyklus eingestellt werden und die Reset-Funktion kann innerhalb des angegebenen Zyklus ausgelöst werden. Es stimmt zufällig mit der Zustandserkennungsfunktion im Programm überein, um die Reset-Funktion besser zu realisieren.

Der Protokollstapel hat die Watchdog-Funktion bereits geschrieben, ich muss nur die entsprechende Funktion aktivieren.

Der spezifische Prozess ist:

Ich füge WDT_IN_PM1 in den Vorkompilierungsoptionen hinzu.



Nach dem Hinzufügen dieser Kompilierungsoption Zmain.c ---->main()--->WatchDogEnable(WDTIMX); wird der Watchdog aktiviert und initialisiert, wenn das Programm ausgeführt wird. Der WDTIMX-Parameter ist das Antwortzeitlimit des Programms. Der Standardwert ist 0, was bedeutet, dass das Timeout 1 Sekunde beträgt. (Die maximale Reaktionszeit beträgt 1 Sekunde).

```
#ifdef WDT_IN_PM1
  /* If WDT is used, this is a good place to enable it. */
  WatchDogEnable( WDTIMX );
#endif
```

Ich füge den entsprechenden Codeausschnitt des Programms hinzu.

```
#define WD_KICK()           st( WDCTL = (0xA0 | WDCTL & 0x0F); WDCTL = (0x50 | WDCTL & 0x0F); )
```

Ich füge den Code entsprechend hinzu und stelle sicher, dass der Programmzeitraum kürzer ist als das Antwort-Timeout. Beispielsweise kann WD_KICK-Funktion zu osal_start_system der Hauptfunktion hinzugefügt werden:

```
void osal_start_system( void )
{
#if !defined ( ZBIT ) && !defined ( UBIT )
  for(;;)  // Forever Loop
#endif
  {
    osal_run_system();
    WD_KICK();
  }
}
```

Funktion zum Neustart der Gerätetrennung:

Diese Funktion zielt darauf ab, das Programm normal auszuführen, aber das Gerät verliert aus irgendeinem Grund die Verbindung, sodass das Gerät kontinuierlich das Netzwerk durchsucht und sich wieder dem Koordinatornetzwerk anschließt.

```
if (events & SEND_DATA_EVENT1)
{

sendESP32();
OLED();
Reset();

 osal_start_timerEx(GenericApp_TaskID,SEND_DATA_EVENT1,10000);
return (events ^ SEND_DATA_EVENT1) ;
```

```
case FEEDBACK:
   osal_memcpy (buf6,pkt->cmd.Data,osal_strlen("Statecheck")+1);
 if(osal_memcmp(buf6,"Statecheck",osal_strlen("Statecheck")+1))
  {
 Feedback();
 n=0;
  }
 break;
```

```
void Reset( void )
{
 n=n+1;
  if(n>6)
  {
    SystemReset();
  }
}
```

Die Funktion SystemReset() enthält die Funktion HAL_SYSTEM_RESET(), und die Funktion HAL_SYSTEM_RESET() ist eigentlich eine manuelle Ausführung der Watchdog-Funktion. Für die oben erwähnte Watchdog-Funktion wird das Watchdog-Programm in der Funktion Main () festgelegt und nur gestartet, wenn das Programm abstürzt und keine Schleife ausgeführt werden kann. Es gibt viele Gründe für die Trennung, und ich muss nicht alle Gründe kennen. Mein oberstes Ziel ist es, das Gerät wieder mit dem Netzwerk zu verbinden. Ich muss also nur eine Möglichkeit finden, das Gerät zu initialisieren, wenn Probleme auftreten. Deshalb habe ich zusätzlich dieses Systeminitialisierungsprogramm eingerichtet, um das Gerät möglichst lange am Laufen zu halten und eine Trennung vom Netzwerk zu vermeiden.

```
// Restart system from absolute beginning
// Disables interrupts, forces WatchDog reset
#define SystemReset()       \
{                           \
  HAL_DISABLE_INTERRUPTS(); \
  HAL_SYSTEM_RESET();       \
}
```

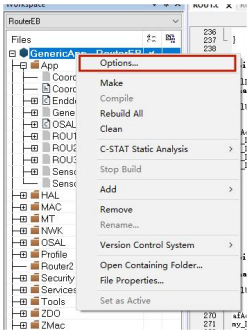
```
/* disable interrupts, set watchdog timer, wait for reset */
#define HAL_SYSTEM_RESET()  st( HAL_DISABLE_INTERRUPTS(); WDCTL = WD_RESET1; WDCTL = WD_RESET2; for(;;); )
```

6.2.1.3 Einige Probleme und Lösungen bei der Verwendung von Zigbee

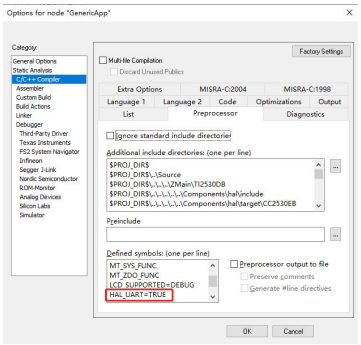
UART-Nutzung

Da das ZigBee-Protokoll eine bedingte Kompilierung verwendet, ist es bei Verwendung von UART erforderlich, das HAL_UART-Makro zu definieren und seinen Wert TRUE zuzuweisen. In der IAR-Entwicklungsumgebung können Sie die folgende Methode verwenden, um die Makrodefinition für UART zu öffnen.

Klicken Sie mit der rechten Maustaste auf das Projekt GenericApp-RouterEB und wählen Sie „Optionen“ aus dem Popup-Dropdown-Menü。



Wählen Sie die Registerkarte „C/C++-Compiler“ aus, wählen Sie die Registerkarte „Präprozessor“ auf der rechten Seite des Fensters aus, geben Sie dann „HAL UART=TRUE“ in das Dropdown-Listenfeld „Definierte Symbole“ ein und klicken Sie schließlich auf „OK“. (HAL_UART=TRUE: Die serielle Schnittstelle ist immer geöffnet, sodass beim Senden der seriellen Schnittstelle kein „#“ nach dem Befehl für die serielle Schnittstelle hinzugefügt werden muss.)



Beschreibung der CC-Debugger-Schnittstelle

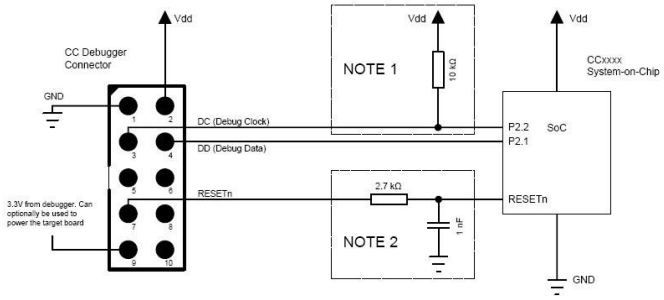
Schnittstelleninformationen:

Beachten Sie die Richtung der Kerbe am CC-Debugger

71CC-Debugger-Schnittstelle(C)

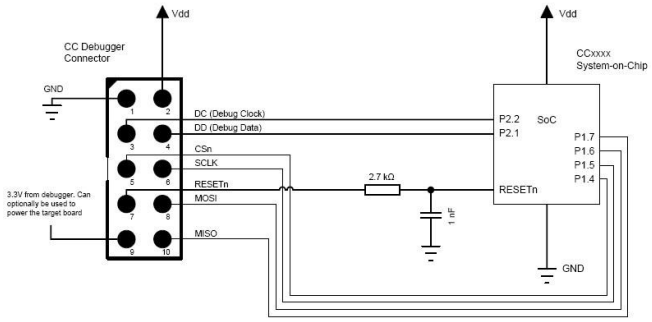
Verbindungen unter IAR Embedded Workbench für C8051, SmartRF Flash Programmer, SmartRF Studio-Umgebung:

Wenn das ZigBee-Gerät nicht separat mit Strom versorgt wird, müssen mindestens fünf Pins angeschlossen werden, nämlich **VCC, GND, RST, P21, P22.**



72CC-Debugger-Schnittstelle für Flash Programmer(C)

Verbindung in der SmartRF Packet Sniffer-Umgebung:

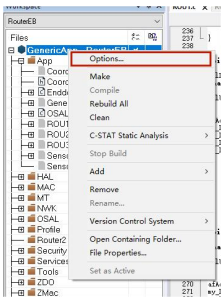


73CC-Debugger-Schnittstelle für Packet Sniffer(C)

SmartRF

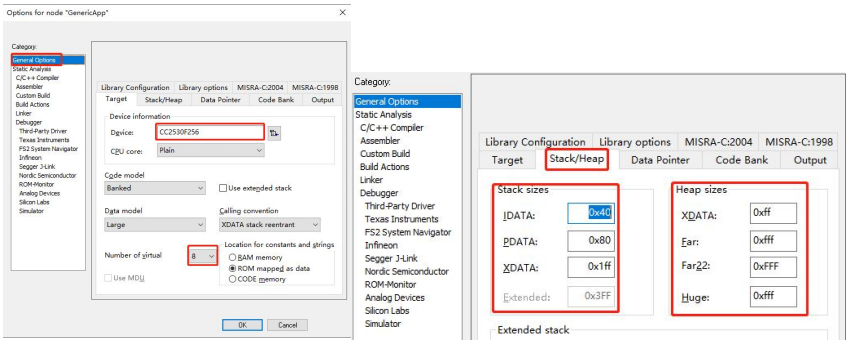
Bevor Sie das Programmierprogramm verwenden, müssen Sie die Programmierdatei korrekt ausgeben. Der spezifische Vorgang ist wie folgt:

Klicken Sie zunächst mit der rechten Maustaste und wählen Sie im Popup-Dropdown-Menü „Optionen“.

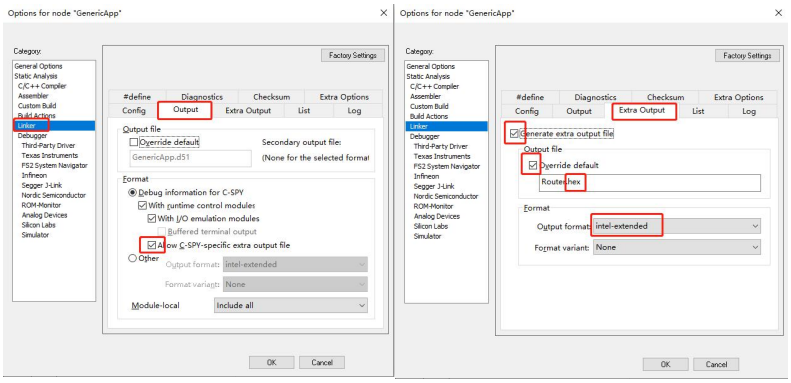


Befolgen Sie dann die nachstehende Abbildung, um die Konfiguration des Hex-Ausgangs abzuschließen。

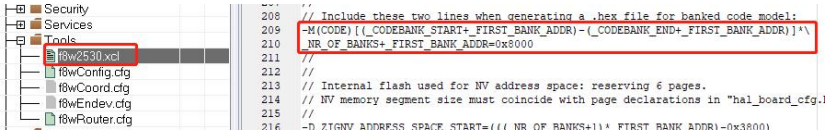
Wählen Sie zunächst das Chipmodell und die Konfigurationsparameter richtig aus.



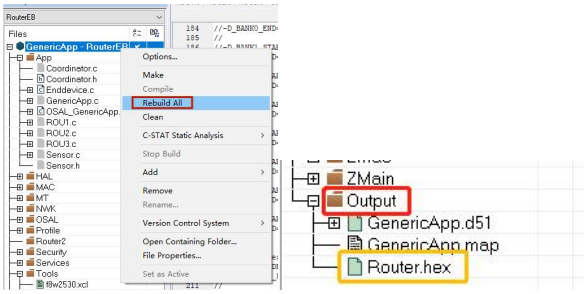
Aktivieren Sie dann die Option im roten Kästchen im Bild.



Beachten Sie, dass Sie nach Abschluss Ihres eigenen Programms die Datei f8w2530.xcl im Ordner „Tools“ finden müssen. In den Zeilen 209 und 210 müssen Sie das // vor diesen beiden Zeilen löschen, andernfalls wird „Speicherplatz überschritten“ angezeigt Brennen des Programms im nächsten Schritt „Fehlermeldung;

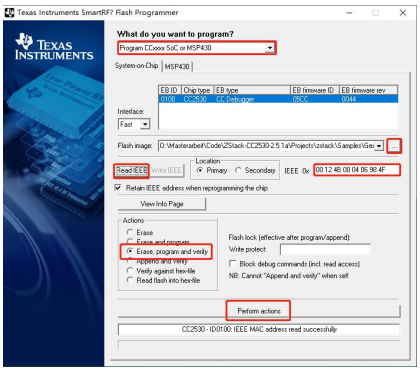


Klicken Sie dann mit der rechten Maustaste auf die Projektdatei und wählen Sie im Popup-Dropdown-Menü „ALLE neu erstellen“. Wenn im Programm kein Fehler vorliegt, kompiliert das System das Programm, um eine .hex-Datei zu generieren, bei der es sich um die Brenndatei handelt wir brauchen.

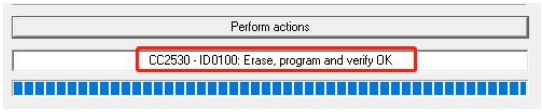


Öffnen Sie dann die installierte SmartRF Flash Programmer-Software, schließen Sie den CC Debuger-Downloader an, die Software wählt automatisch den 430-Treiber aus, klicken Sie auf die Dateiauswahlschaltfläche, um die Hex-Datei zu finden, die in den Chip gebrannt werden soll, und klicken Sie auf die Schaltfläche „IEEE lesen“, um zu überprüfen, ob die Wenn die Software den IEEE-Code des Chips richtig erkennt,

wählen Sie die Option „Überprüfen“, um das vorherige Programm im Chip zu löschen und Programmfehler zu vermeiden, und klicken Sie dann auf „Aktionen ausführen“, um mit der Programmierung des Programms zu beginnen.



Wenn der Fortschrittsbalken unten abgeschlossen ist und das Eingabeaufforderungsfeld wie in der Abbildung unten angezeigt wird, bedeutet dies, dass die Programmierung abgeschlossen ist. Wenn andere Eingabeaufforderungen angezeigt werden, handelt es sich um einen Programmierfehler.

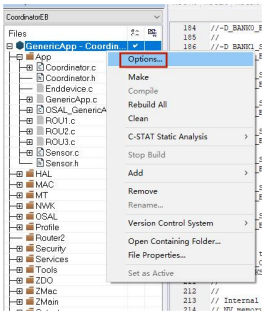


Koordinator ausschalten und neu starten

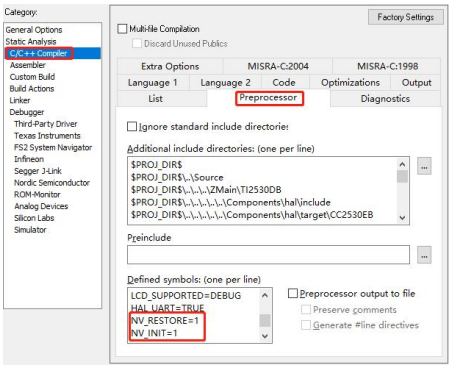
Nachdem der Koordinator erfolgreich vernetzt wurde und der Koordinator ausgeschaltet und dann erneut verbunden wird, kann das Endgerät (Netzwerkstatus des Routers) keine erneute Verbindung zum Koordinator herstellen. Der Grund dafür ist, dass das Endgerät den Koordinator trennt, nachdem der Koordinator vernetzt ist und alle Endgeräte (der Netzwerkstatus des Routers) ein neues Netzwerk bilden. Nach dem erneuten Verbinden des Koordinators stimmt der Netzwerkkanal des Koordinators nicht mit dem Netzwerkkanal überein des Endgeräts. Dies führt dazu, dass der Koordinator die Verbindung nicht wiederherstellen kann. Die Lösung besteht also darin, die PanID des Koordinators und den Netzwerkkanal festzulegen, sodass sich Zigbee immer im selben Netzwerk befindet.

Lösung:

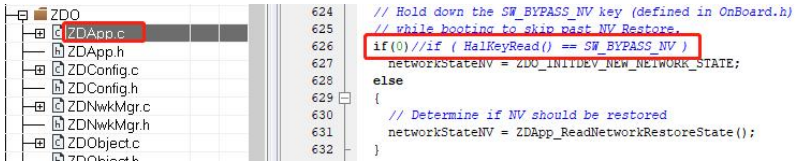
Klicken Sie mit der rechten Maustaste unter die Koordinator-Projektdatei und wählen Sie Optionen aus.



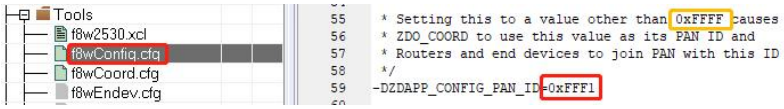
Wählen Sie die Registerkarte „C/C++-Compiler“ aus, wählen Sie die Registerkarte „Präprozessor“ auf der rechten Seite des Fensters aus und geben Sie „NV_RESTORE=1 NV_INIT=1“ in das Dropdown-Listenfeld „Definierte Symbole“ ein



Gehen Sie zu Zeile 626 von ZDApp.c, kommentieren Sie if (HalKeyRead() == SW_BYPASS_NV) aus und ändern Sie es in if(0).



Geben Sie Zeile 59 von f8wConfig.cfg ein. Der Standardwert von PADID ist 0xFFFF, was bedeutet, dass der Koordinator bei jedem Einschalten zufällig eine PanID generiert. Ändern Sie die PanID in 0xFFF1, dann bleibt die PADID des Koordinators für immer unverändert und ist immer 0xFFF1.



Nach diesen Einstellungen können andere Knoten nach dem Neustart des Koordinators direkt wieder eine Verbindung herstellen, ohne den Koordinator erneut finden zu müssen.

6.2.2 ESP32-System

6.2.2.1 Datenerfassung und Verarbeitungsanalyse

Bei der Datenerfassung werden die der Erkennungsumgebung entsprechenden Daten durch Auslesen des vom Mikrocontroller-Pin erfassten analogen Sensorausgangssignals ermittelt.

Bei der Datenverarbeitung wird der Durchschnittswert durch Mehrfachabtastung ermittelt, um Fehler bei der Dateninterferenz zu reduzieren.Ich verwende die ersten fünf Messdaten, um die Durchschnittszahl zu berechnen, und das Ergebnis wird als Sensorausgangswert verwendet, wodurch zufällige Fehler reduziert, Präzision und Genauigkeit verbessert, Systemfehler beseitigt, der Einfluss abnormaler Werte gemildert und die Stabilität erhöht werden können.

Die spezifische Methode zur Ermittlung des Durchschnittswerts ist wie folgt:

```
// Definieren die Größe des Arrays, initialisiert auf 5
const int numReadings1 = 5;
// Definieren ein Array zum Speichern der gelesenen Werte
int readings1[numReadings1];
// Definieren den Index, der den Wert des Arrays angibt
int readIndex1 = 0;
// Definieren die Gesamtzahl der gespeicherten Array-Werte
int total1 = 0;
// Definieren den Mittelwert der Array-Werte
int average1 = 0;
```

```
sensors.requestTemperatures();
fire = analogRead(Pin2);
rauch = analogRead(Pin1);
co = analogRead(Pin3);
```

```
total1 = total1 - readings1[readIndex1];
// Lesen den Wert des aktuellen Sensors und speichern Sie ihn im letzten Bit des Arrays
v1 =sensors.getTempCByIndex(0);
// Addieren den zuletzt gelesenen Wert zur Gesamtsumme
readings1[readIndex1]=v1;//*7/12-145;
total1= total1 + readings1[readIndex1];
// Addieren 1 zum vom Array angegebenen Indexwert
readIndex1 = readIndex1 + 1;
// Stellen Sie fest, ob der Array-Indikatorindex den Bereich des Arrays überschreitet,
// und setzen Sie in diesem Fall den Array-Indikatorindex auf 0 zurück
if (readIndex1 >= numReadings1) {
  readIndex1 = 0;
}
//Berechnen den Durchschnitt eines Arrays
 average1 = total1 / numReadings1;
```

74 Sensor Code

Eine Methode zur Analyse der Umgebung, bei der mehrere Sensorwerte miteinander kooperieren.Die Vorteile dieses Ansatzes sind:

1.Bereitstellung umfassender Umgebungsinformationen: Verschiedene Arten von Sensoren können verschiedene Aspekte der Umgebung überwachen, wie z. B. Temperatur, Luftfeuchtigkeit, Druck, Licht, Gaskonzentration usw. Die Zusammenarbeit mit den Daten verschiedener Sensoren kann umfassendere und globalere Umweltinformationen liefern und uns helfen, ein umfassendes Verständnis der Umwelt zu erlangen.

2.Erhöhen die Datengenauigkeit und -zuverlässigkeit: Ein einzelner Sensor kann durch Rauschen, Fehler oder Ausfälle beeinträchtigt sein, aber durch die Zusammenarbeit mehrerer Sensoren können sie sich gegenseitig ergänzen und die Datengenauigkeit und -zuverlässigkeit verbessern.

3.Verbesserte Echtzeitleistung: In manchen Anwendungsszenarien ist ein einzelner Sensor möglicherweise nicht in der Lage, die erforderlichen Informationen zeitnah bereitzustellen. Der Einsatz mehrerer Sensoren kann die Echtzeitleistung von Daten verbessern und schneller Informationen über Umweltveränderungen erhalten.

4.Kompensieren Sensoreinschränkungen: Jeder Sensor hat seine eigenen Einschränkungen und ist möglicherweise nur innerhalb bestimmter Bereiche oder Bedingungen wirksam. Durch den gemeinsamen Einsatz mehrerer Sensoren können deren jeweilige Grenzen ausgeglichen und der Erfassungsbereich und die Fähigkeiten erweitert werden.

Ich sortiere die Sensordaten und füge den Datennamen hinzu, während ich die Daten über die serielle Schnittstelle ausgebe, um die Datenübertragung und -klassifizierung zu erleichtern.

Die endgültigen Ausgabedateninformationen lauten wie folgt:

```
[2023-07-27 11:40:59.642]
&&MES0003$ Zahl 000104 Temp -00127 Fire 003881 Rauch 003881 Co 000000 $$
[2023-07-27 11:41:01.793]
&&MES0001$ Zahl 000103 Temp -00127 Fire 003886 Rauch 003886 Co 000000 $$
[2023-07-27 11:41:02.338]
&&MES0002$ Zahl 000104 Temp -00127 Fire 003891 Rauch 003891 Co 000000 $$
[2023-07-27 11:41:02.883]
&&MAN0001$ Zahl 000109 Temp -00127 Fire 003851 Rauch 003851 Co 000000 $$
```

Ich habe die Länge der gesamten Zeichenfolge auf 73 eingestellt. Es enthält folgende Informationen:

&&: Das Anfangszeichen, das als Prüfziffer der übertragenen Informationen verwendet wird. Das System akzeptiert eine Zeichenfolge nur, wenn sie die Zeichenfolge am Anfang enthält. Dadurch kann das System Fehlermeldungen wie verstümmelte Zeichen filtern.

MES0001/MAN0001: Adresse der Seriennummer des Endgeräts. Es wird verwendet, um die Quelle der Sensordaten zu markieren, was für den Host-Computer-Client die Klassifizierung der Daten erleichtert.

$: Informationsintervallsymbol. Separate Gerätemodell- und Sensordaten. Vermeiden Sie Verwechslungen zwischen Geräteseriennummern und Sensordaten.

Zahl: Anzahl der gesendeten Daten. Jedes Mal, wenn Daten gesendet werden, wird der Zähler um eins erhöht. Sie können anhand des Zählerwerts beurteilen, ob das Gerät Daten normal sendet, ob Daten fehlen oder ob ein Systemfehler oder andere Probleme vorliegen.

Temp,Feuer,Rauch,Co:entsprechend den Sensoren für Temperatur, Flamme, brennbares Gas und Kohlenmonoxid. Alle Daten haben eine feste sechsstellige Länge, was nicht nur die vollständige Anzeige der Daten gewährleistet, sondern auch die Verarbeitung der Zeichenfolgen durch den Host-Computer erleichtert und die Datenerfassung und -analyse erleichtert.

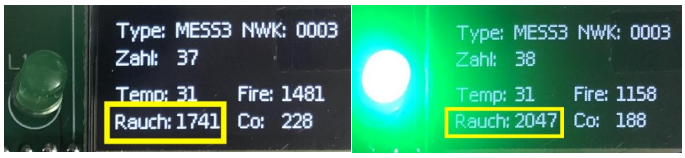
$$: Das Endzeichen, das dieselbe Funktion wie das Anfangszeichen hat.

Ich stelle die Alarmschwelle ein und verwende das LED-Licht als Alarmerinnerung

Durch die oben beschriebene Verarbeitung und Analyse von Sensordaten kann anhand der Daten eine Beurteilung der Umweltgefährdung vorgenommen werden. Wenn die Temperatur, die Konzentration an brennbaren Stoffen oder die Kohlenmonoxidkonzentration den eingestellten Alarmwert (rotes Kästchen) erreicht, leuchtet die LED auf.

Der Flammensensor wird durch Licht gestört, sodass die LED nicht direkt leuchtet, wenn der Wert den Alarmwert erreicht, und gleichzeitig wird der Wert des Temperatursensors auch als Referenz einbezogen, nur wenn die Temperatur erreicht ist Wenn die Werte des Flammensensors gleichzeitig den Alarmwert erreichen (oranges Feld), leuchtet das LED-Licht auf.

```
  if(average1>50||average3>2000||average4>2000)
{
  digitalWrite(LED1, HIGH);
  Serial2.print("warning");
  //Serial.print("warning");
}
else if(average2>2000&&average1>50)
{
  digitalWrite(LED1, HIGH);
  Serial2.print("warning");
  //Serial.print("warning");
}
 else
 {
  digitalWrite(LED1, LOW);
  //Serial.print("safe");
  Serial2.print("safe");
 }
```



75LED Alarm

6.2.2.2 OLED-Bildschirmanzeige

Als wichtiger Teil des Geräts spielt der Bildschirm eine wichtige Rolle. Es dient zur Anzeige der wichtigsten Informationen zu jedem Gerät.

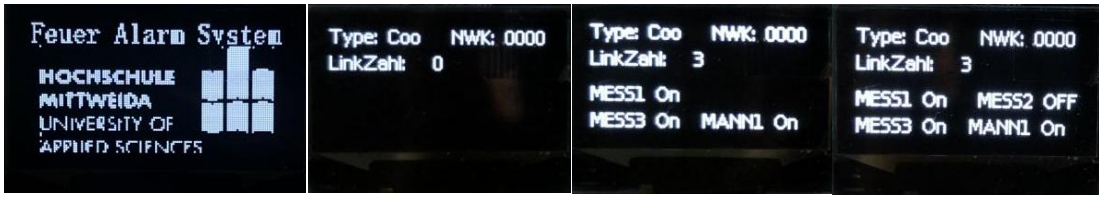
Als nächstes werde ich die Informationen vorstellen, die auf den Bildschirmen verschiedener Geräte im System angezeigt werden.

Koordinator

Der Bildschirm verfügt über eine Startanimation. Nach Abschluss der Initialisierung werden die aktuellen Knoteninformationen, die Anzahl der Knotenverbindungen und der angeschlossene Knoten angezeigt.

Die Anzeigelogik besteht hier darin, dass der Bildschirm nur dann die Knoteninformationen anzeigt, wenn der Knoten tatsächlich zum ersten Mal mit dem Koordinator verbunden ist, anstatt direkt den Verbindungsstatus jedes Knotens anzuzeigen, um eine Fehleinschätzung des Knotenstatus zu vermeiden. Der spezifische Prozess ist: Nach der Initialisierung des Koordinators besteht innerhalb kurzer Zeit keine Knotenverbindung, die Anzahl der auf dem Bildschirm angezeigten Verbindungen beträgt 0 und es werden unten keine Informationen angezeigt. Wenn ein Knoten zum ersten Mal dem Netzwerk beitritt, werden auf dem Bildschirm die Anzahl der Verbindungen sowie der entsprechende Knotenname und Verbindungsstatus angezeigt. Wenn das Gerät nicht innerhalb einer Minute auf die Nachricht zur Überprüfung des Verbindungsstatus antwortet, wird auf dem Bildschirm angezeigt, dass die Verbindung zum Knoten getrennt ist, und der Anzeigestatus der verlorenen Verbindung bleibt bestehen, bis sich der Verbindungsstatus des Knotens ändert und sich der Bildschirm nicht ändert.

Wenn der serielle Port die Zeichenfolge „XXXX check“ zum ersten Mal erkennt, wird „XXXX On“ auf dem Bildschirm angezeigt. Wenn „XXXX check“ nicht innerhalb einer Minute erneut empfangen wird, wird „XXXX OFF“ angezeigt



76OLED-Bildschirmanzeige Koordinator

MESS-Knoten

Der Bildschirm verfügt über eine Startanimation. Nach Abschluss der Initialisierung werden die Gerätenummer des aktuellen Knotens, die Anzahl der gesendeten Daten und der Wert jedes Sensors angezeigt. Da die Sensorknoten nicht ständig von Menschen gesteuert werden und Informationen anzeigen müssen und keine komplexen Schnittstellen entwerfen und Inhalte anzeigen müssen, nimmt der angezeigte Inhalt ein festes Format an.

77OLED-Bildschirmanzeige MESS

Mehrstufige Menüanzeige für Feuermannknoten

Das Informationsanzeigedesign des Feuerwehrknotens ist ein wichtiger Teil dieses Projekts und kann als einer der schwierigsten Punkte des gesamten Projekts bezeichnet werden. Ich verbringe viel Zeit damit, die Anzeigelogik und die Art und Weise, wie die Informationen angezeigt werden, zu entwerfen. Als nächstes werde ich die Anzeigefunktion des Feuerwehrknotens im Detail vorstellen.

Erstens wird der Feuerwehrknoten von Feuermannn getragen und ist für Feuermann ein wichtiges Mittel, um die Informationen zum Brandort zu verstehen. Damit Feuermann möglichst viele Informationen über den Brandort verstehen und die Informationen zum Brandort klar und ohne Missverständnisse verstehen können, habe ich eine mehrstufige Menüanzeigefunktion entwickelt. Jede Schnittstelle entspricht unterschiedlichen Inhalten.

Tastensteuerung

Da der angezeigte Inhalt eine mehrstufige Menüfunktion hat und hin- und hergeschaltet werden muss, habe ich drei Tasten zugeordnet, um die Auswahl bzw. Bestätigung nach oben und unten zu steuern.

Der zugrunde liegende Code des mehrstufigen Menüs verwendet ein Open-Source-Projekt auf Github. Die Besonderheit dieses Projekts besteht darin, dass die Animation sehr flüssig ist und Übergangseffekte aufweist. Auf dieser Basis habe ich eine Weiterentwicklung durchgeführt, die Menüsteuerung und mein Projekt integriert und optimiert. Lassen Sie mich als Nächstes die spezifische Funktion jeder Option im Menü vorstellen.

Die Sicherheitsstatusanzeige jedes Knotens

Für Feuermann ist die Sicherheit jedes Sensorknotens die wichtigste Information. Daher werde ich als erstes Element den Sicherheitsstatus jedes Knotens anzeigen. Wenn der Feuerwehrmann auf diese Funktion klickt, sieht er deutlich den Sicherheitsstatus jedes Knotens. Die Bewertung des Sicherheitsstatus wird von mir selbst entschieden und die Beurteilung erfolgt auf der Grundlage der Sensordaten, die nicht im Widerspruch zum Alarm des Hosts stehen Computer. Wenn die vom Sensorknoten erfassten Daten im sicheren Bereich liegen, zeigt der Bildschirm an, dass sich der entsprechende Knoten in einem sicheren Zustand befindet. Andernfalls wird es als unsicher angezeigt. Um zu verhindern, dass Feuermann den Knotenstatus falsch einschätzen, zeigt das Display keine Informationen über seinen Sicherheitsstatus mehr an, wenn ein Sensorknoten die Verbindung verliert und keine Informationen mehr normal senden kann, sodass Feuermann nicht nur eine Fehleinschätzung des Sicherheitsstatus vermeiden können Durch Vergleich mit den zuvor erhaltenen Informationen kann festgestellt werden, dass ein bestimmter Knoten die Verbindung verloren hat. Beispielsweise zeigt der Bildschirm zu Beginn immer an, dass Knoten 1 sicher ist, aber nach einer gewissen Zeit liegen keine Informationen über Knoten 1 vor, was bedeutet, dass Knoten 1 getrennt wurde.

Beispiel: Nur der Fireman-Knoten tritt dem Netzwerk im aktuellen Gerät bei, sodass nur der Sicherheitsstatus des Fireman-Knotens angezeigt wird und der Sicherheitsstatus anderer Knoten nicht angezeigt wird.

Anzeige des Verbindungsstatus jedes Knotens

Ob der Knoten sicher ist, stellt für Feuermann eine sehr wichtige Information dar. Darüber hinaus ist es auch eine Information, auf die sich Feuermann konzentrieren, ob der Knoten angeschlossen ist und normal funktioniert.

Dieser Teil der Anzeige ähnelt dem, der auf dem Koordinatordisplay angezeigt wird. Die Schnittstelle kann die Anzahl der Knotenverbindungen und den Knoten angeben, mit dem eine Verbindung hergestellt werden soll. Und genau wie die Anzeigelogik des Koordinators wird der Verbindungsstatus nur angezeigt, wenn der Knoten tatsächlich zum ersten Mal dem Netzwerk beitritt. Wenn die Verbindung zum Knoten nach dieser Zeit getrennt wird, wird „OFF“ angezeigt.

Beispiel: Nur Firefighter-Knoten im aktuellen Gerät treten dem Netzwerk bei. Daher zeigt die Anzeige an, dass die Anzahl der Verbindungen 1 beträgt und der Firefighter-

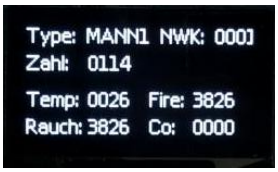
Knoten sich im Online-Zustand befindet. Wenn die Verbindung zum Firefighter-ZigBee-Knoten zu diesem Zeitpunkt getrennt wird, zeigt das Display nach einer Weile an, dass sich der Firefighter-Knoten im Offline-Zustand befindet und die Anzahl der Verbindungen 0 wird.

Anzeige der Knotensensordaten

Der Feuerwehrknoten verfügt über denselben Sensor wie der Sensorknoten, um mit dem Sensorknoten konsistente Daten zu übertragen.

Damit Feuermann Daten aus ihrer Umgebung einsehen können, wurde eine Funktion zur Anzeige von Sensordaten hinzugefügt.

Der angezeigte Inhalt ist die Gerätenummer des aktuellen Knotens und der Wert jedes Sensors. Der angezeigte Inhalt ist derselbe wie der des Sensorknotens.



Anzeige der Netzwerktopologie des Geräts

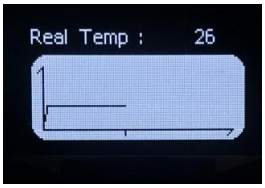
Diese Funktion basiert auf der Verwendung von Zigbee zum Anzeigen der Netzwerktopologie. Feuermann können sehen, mit welchem Sensorknoten ihr Knoten gerade verbunden ist.Ich zeige die Informationen auf dem Bildschirm an, indem ich die vom seriellen Port gesendeten Knoteninformationen lese.Wenn auf dem Bildschirm normalerweise die Kurzadresse des Knotens und die Kurzadresse des übergeordneten Knotens angezeigt werden können, bedeutet dies, dass der Knoten zu diesem Zeitpunkt normal verbunden ist und keine Unterbrechung vorliegt. Wenn der Knoten die Verbindung verliert, wird die entsprechende Adresse nicht angezeigt. Dies ist auch eine Möglichkeit zu beurteilen, ob ein Knoten erfolgreich eine Verbindung zum Netzwerk hergestellt hat.

Im Beispiel lautet die Kurzadresse des Feuerwehrknotens zu diesem Zeitpunkt 4F25 und die Kurzadresse des übergeordneten Knotens lautet 0000. Im ZigBee-Netzwerk ist 0000 immer die Kurzadresse des Koordinators. Daher besteht die Netzwerkstruktur zu diesem Zeitpunkt darin, dass der Feuerwehrknoten direkt mit dem Koordinator verbunden ist.

Zeichne ein Diagramm

Die Temperatur ist ein wichtiger Sicherheitsindikator für Feuermann an einem Brandort. Daher verarbeite ich die Temperaturdaten speziell separat, verwende die Temperaturdaten als Datenquelle für das Symbol und zeichne ein Temperaturänderungstrenddiagramm. Durch die Beobachtung der Veränderungen in der Temperaturkurve kann man einfach analysieren, ob die Umgebung sicher ist. Das Diagramm kann Daten 50 Mal aufzeichnen. Wenn die 51. Daten empfangen werden, wird die Kurve auf dem Bildschirm zurückgesetzt und das Diagramm erneut gezeichnet. Es ist nicht erforderlich, das Diagramm manuell zu löschen.

Das Beispiel im Bild besteht darin, die Raumtemperatur zu erfassen und ein Temperaturänderungstrenddiagramm zu zeichnen. Es ist ersichtlich, dass die Raumtemperatur über einen bestimmten Zeitraum auf 26 Grad Celsius gehalten wurde und die Kurve den Zustand der Raumtemperatur ohne jegliche Änderung korrekt wiedergeben kann.



Andere Funktionen

Konvertierung des Bildformats

Dies ist eine Erkenntnis, die ich zufällig bei meiner Abschlussarbeit gelernt habe. Mithilfe von Software kann theoretisch jedes .Jpg.png-Format in das .c-Format konvertiert werden. Also nutzte ich diese Gelegenheit, um eine Kofferraumabdeckung anzufertigen. Um zu verdeutlichen, dass der Inhalt des Bildes mit dem Abschlussentwurf in Zusammenhang steht, umfasst der von mir erstellte Titelinhalt das Thema des Abschlussentwurfs und das Symbol der Schule.

Video

Das Grundprinzip besteht darin, das Video in ein Einzelbild umzuwandeln und dann jedes Bild in ein .c-Format zu konvertieren. Ich speichere die .c-Datei im Einzelchip-Mikrocomputer und lasse den Einzelchip-Mikrocomputer die Dateien dann der Reihe nach entsprechend dem Dateinamen lesen, sodass ein Animationsanzeigeeffekt entsteht.

Die drei Tasten entsprechen jeweils den Funktionen „Wiedergabe“, „Pause“ und „Beenden“.

Drücken Sie die Wiedergabe-Taste, um die Wiedergabe des Videos zu starten, und drücken Sie die Pause-Taste, um das Pausensymbol anzuzeigen. Wenn an dieser

Stelle die Wiedergabe-Taste gedrückt wird, wird das Video weiter abgespielt. Kehren Sie zur Menüoberfläche zurück, wenn die Exit-Taste gedrückt wird.

Hier speichere ich eine Videodatei einer Figur, die Basketball spielt.

Masterarbeit-Teilnehmer zeigen

Ich verwende die Übergangsanimation, um den Avatar des Charakters und den entsprechenden Namen anzuzeigen. Um das Abschlussprojekt interessanter zu gestalten, zeige ich hier die Avatar-Informationen von mir und Professor Kuhl.

Geräteinformation

Zeigt einige grundlegende Informationen des Mikrocontrollers an.

6.2.3 WLAN und Server einrichten

Die Informationsübertragung erfordert Übertragungskanäle und Serverunterstützung. Ich bin der Meinung, dass die Umgebung, in der das Gerät verwendet wird, nicht unbedingt über die beiden oben genannten Elemente verfügt und das Fehlen eines dieser Elemente dazu führen wird, dass das System die Informationsübertragung nicht realisiert. Meine Idee ist es also, selbst einen Satz WLAN und Server zu erstellen, sodass das gesamte System beim Betrieb nicht auf die Nutzungsumgebung Rücksicht nehmen muss und beim Einschalten sofort verwendet werden kann.

Hinsichtlich der Datenübertragung habe ich zwei Übertragungsprotokolle in Betracht gezogen, UDP und TCP.

Ich habe die Vor- und Nachteile der beiden analysiert: Das UDP-Übertragungsprotokoll muss vor der Datenübertragung keine Verbindung herstellen, daher ist die Übertragungsgeschwindigkeit schneller, mit geringer Latenz und schnellen Übertragungseigenschaften, aber wenn das Netzwerk überlastet ist, besteht die Möglichkeit Der Datenpaketverlust wird zunehmen. . TCP ist ein zuverlässiges Übertragungsprotokoll, das die ordnungsgemäße Übertragung und den zuverlässigen Empfang von Datenpaketen durch Verbindungsaufbau und -wartung gewährleistet. TCP verfügt über einen Fehlererkennungs- und Neuübertragungsmechanismus, um die Integrität und Richtigkeit von Datenpaketen sicherzustellen. Aufgrund der Notwendigkeit komplexer Mechanismen wie Verbindungsaufbau, Fehlererkennung und

erneuter Übertragung weist TCP eine langsamere Übertragungsgeschwindigkeit als UDP auf und es kommt zu einer gewissen Übertragungsverzögerung.

Systemnutzungsumgebung: Das von mir entworfene System verfügt während des Betriebs nur über ein Client-Empfangsgerät, und ich benötige die Integrität und Korrektheit der Sensordatenübertragung. Das System sendet alle zehn Sekunden Daten, daher ist die Anforderung an die Latenz nicht hoch.

Durch die Kombination der Eigenschaften der beiden Übertragungsprotokolle und der Systemnutzungsumgebung zur Analyse entschied ich mich schließlich für die Verwendung des TCP-Übertragungsprotokolls, um die Kommunikation mit dem Host-Computer herzustellen.

Ich verwende den AT-Befehl, um die entsprechende Funktion zu aktivieren:

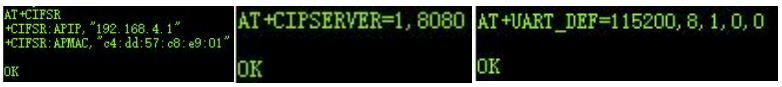
AT-Befehl	*Funktion*
AT+CWMODE=2	*AP-Modus (WLAN erstellen)*
AT+CWSAP="ESP32-Xin Chu","12345678",11,3	*Einstellparameter: Name, Passwort, Kanalnummer, Verschlüsselungsmethode*
AT+CIPMUX=1	*Mehrere Verbindungen zulassen*
AT+CIPSERVERMAXCONN=5	*Die maximal zulässige Anzahl von Geräteverbindungen beträgt 5*
AT+CIPSERVER=1,8080	*Build-Server*
AT+UART_DEF=115200,8,1,0,0	*Die serielle Kommunikation einrichten*

10 AT-Befehl

```
void ESP32_Init()
{
  mySerial1.println("AT+CWMODE=2");
  delay(500);
  mySerial1.println("AT+CWSAP=\"ESP32-Xin Chu\",\"12345678\",11,3");
  delay(500);
  mySerial1.println("AT+CIPMUX=1");
  delay(500);
  mySerial1.println("AT+CIPSERVERMAXCONN=5");
  delay(500);
  mySerial1.println("AT+CIPSERVER=1,8080");
  delay(500);
  mySerial1.println("AT+UART_DEF=115200,8,1,0,0");
  delay(500);
}
```

Nach Erhalt des AT-Befehls aktiviert ESP32 seine eigene Wi-Fi-Funktion. Die IP-Adresse lautet 192.168.4.1. Dann richtet ESP32 den Server ein und öffnet die Kommunikation über die serielle Schnittstelle, sodass die Daten normal empfangen und an den Client gesendet werden können.

78 ESP32-WiFi



79AT-Befehl Server Feedback

Hinweis: Bevor Sie AT-Befehle verwenden, müssen Sie die AT-Firmware für das Entwicklungsboard programmieren. Der Vorgang des Brennens der Firmware wird hier nicht beschrieben. Die Verwendung von AT-Befehlen kann nur über die serielle Schnittstelle 2 der Entwicklungsplatine gesendet werden, und die entsprechenden Pins sind 16 (RX), 17 (TX).

6.3 Kapitelzusammenfassung

Dieses Kapitel gibt eine detaillierte Einführung in den Softwareentwurf des Systems, der meiner Meinung nach auch der schwierigste Teil ist. Ich habe während dieses Prozesses viel neues Wissen gelernt und die meiste Zeit verbringe ich mit dem Programmentwurf und dem Debuggen. und drahtlose Übertragung von Daten, hauptsächlich Datenempfang und -übertragung. Der Z-Stack-Protokollstapel bietet eine Plattform für die ZigBee-Technologie und hilft mir, die drahtlose Informationsübertragungstechnologie besser zu verstehen.

7 Client-System Design

7.1 Designzweck und Ideen

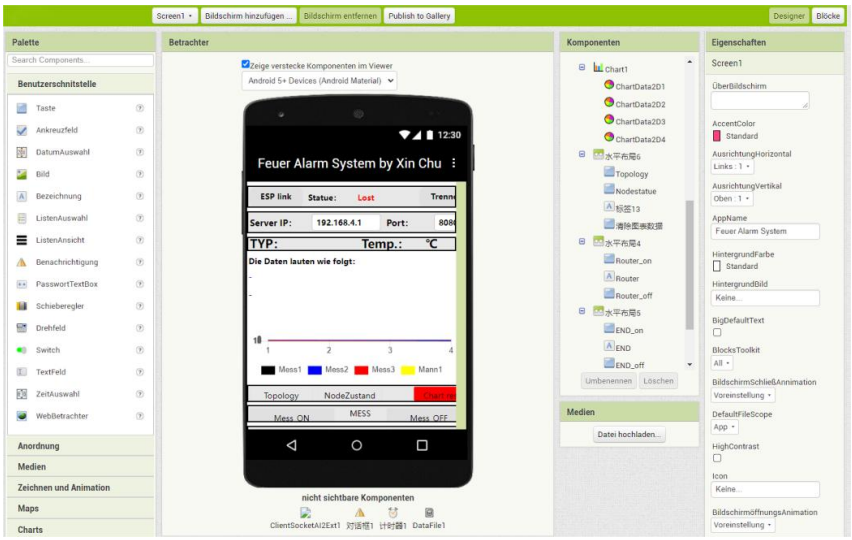
Um zu überprüfen, ob das gesamte System normal läuft, plane ich, einen einfachen Host-Computer für Tests und experimentelle Simulationen zu entwerfen.

Aus diesem Grund habe ich mich dafür entschieden, App Inventor als Entwicklungsplattform zu verwenden, eine mobile APP zu entwerfen und zu erstellen und sie auf dem Host-Computer zu testen.

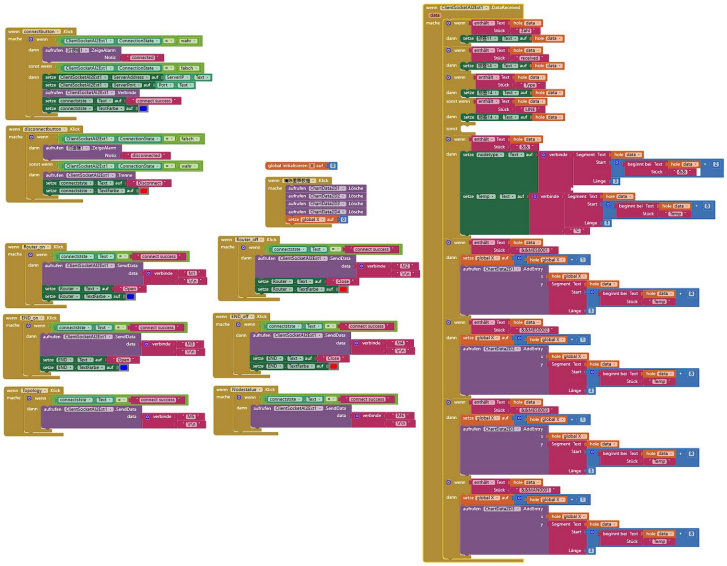
App Inventor ist eine kostenlose Online-Plattform, die vom Massachusetts Institute of Technology (MIT) zum Erstellen von Android-Anwendungen entwickelt wurde. Es ermöglicht Anfängern und Nicht-Entwicklern, mithilfe visueller Programmierung funktionsreiche mobile Anwendungen zu erstellen, ohne komplizierten Code schreiben zu müssen.

App Inventor bietet eine visuelle Benutzeroberfläche, und Benutzer können die Benutzeroberfläche und Funktionen der Anwendung durch Ziehen und Ablegen von Komponenten und Logikblöcken entwerfen. Es verwendet eine grafische Programmiermethode, die Bausteinen ähnelt, um die Funktionen der Anwendung in verschiedene Module aufzuteilen und dann die Funktionen der Anwendung durch Verbinden dieser Module zu realisieren.

Besuchen Sie die offizielle Website (appinventor.mit.edu) in einem Webbrowser, melden Sie sich an und beginnen Sie mit der Erstellung Ihrer eigenen Android-Anwendungen.



80Visuelle Content-Design-Schnittstelle



81Logikblöcken-Design-Schnittstelle

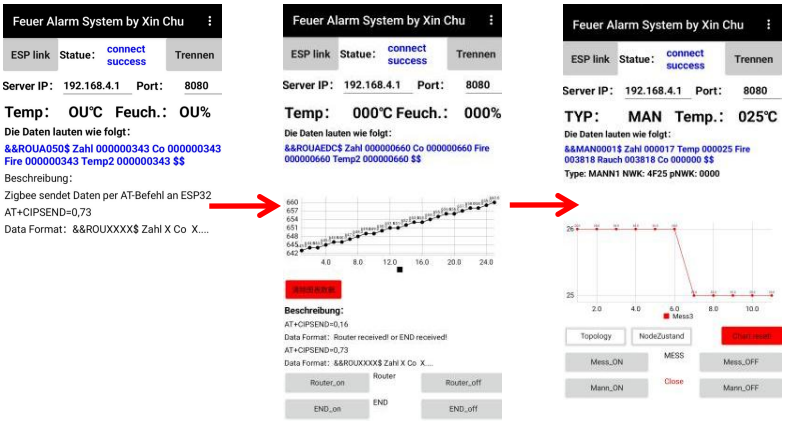
7.2 APP-Upgrades und Funktionen

7.2.1 APP-Symbol

82APP-Symbol

7.2.2 APP-Upgrades

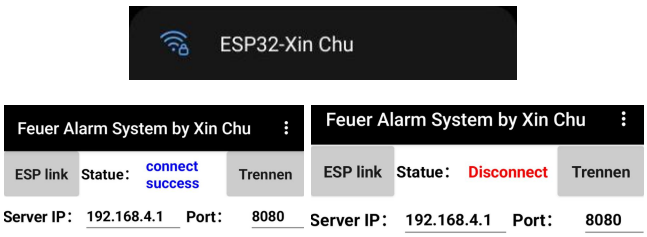
Die APP-Produktion hat lange gedauert, angefangen beim Erlernen der APP-Erstellung, von der einfachsten Verbindung zum Server bis hin zum schrittweisen Hinzufügen weiterer Funktionen. Endlich wurden alle Funktionen realisiert, die ich mir vorgestellt hatte.



83APP-Upgrades

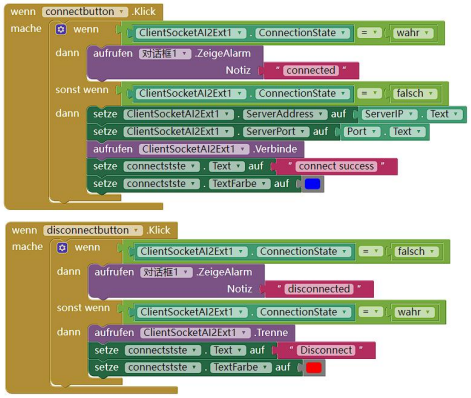
7.2.2.1 Grundlegende Verbindungsfunktion

Als Client muss sich das Mobiltelefon mit dem Server verbinden, um Daten zu empfangen. Die erste Funktion besteht also darin, den Server zu verbinden und zu trennen. Dafür habe ich Schaltflächen zum Verbinden und Trennen des Servers hinzugefügt. Der erste Schritt besteht darin, eine Verbindung zum von ESP32 erstellten WLAN herzustellen. Nachdem die Verbindung erfolgreich hergestellt wurde, öffnen Sie die APP, geben Sie die Server-IP-Adresse 192.168.4.1 an der entsprechenden Position der Server-IP ein und geben Sie die Portnummer 8080 in den Port-Port ein Position. Klicken Sie auf die graue Schaltfläche „ESP link“ auf der linken Seite, wenn das WLAN korrekt verbunden ist und das Gateway initialisiert wurde. Nach dem Klicken auf die Schaltfläche „ESP link“ ändert sich der Status in der mittleren Statusleiste in „connect success“. Dies bedeutet, dass das Mobiltelefon hat sich erfolgreich mit dem Server verbunden. Zu diesem Zeitpunkt sollte die Mobiltelefonschnittstelle die Sensordaten angezeigt haben. Klicken Sie rechts auf die Schaltfläche „Trennen“ und der Status in der mittleren Statusleiste ändert sich in „Disconnect“. Dies bedeutet, dass das Telefon vom Server getrennt wurde.



84Server verbinden

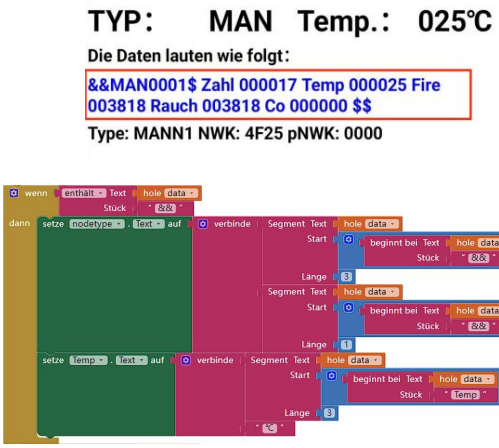
Der entsprechende logische Block besteht darin, das Client-Socket-Plug-In zu verwenden, damit die App die Funktion hat, eine Verbindung zum Server herzustellen.



7.2.2.2 Anzeige der Knoten-Sensordaten

Nach Abschluss der Serververbindung besteht der nächste Schritt darin, Sensordaten zu empfangen. Um die vom System empfangenen Daten vollständig anzeigen zu können. Ich habe zunächst ein Datenanzeigefenster hinzugefügt, das die Knotendaten vollständig anzeigt, ohne dass die Daten verarbeitet werden. Einerseits können die Daten visuell dargestellt werden, und wenn ein Fehler in den übertragenen Daten auftritt, kann der Fehlerort schnell gefunden werden. Andererseits ebnet es auch den Weg für nachfolgende Datenverarbeitungsfunktionen.

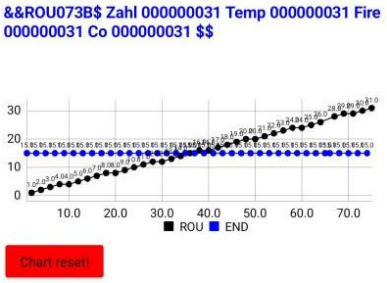
Die schnelle Überprüfung der aktuellen Umgebungstemperatur jedes Geräts ist eine sehr wichtige Aufgabe im gesamten System. Daher extrahiere ich den Knotennamen und die Temperaturdaten aus den empfangenen Daten und zeige sie prominent über den vollständigen Daten an.



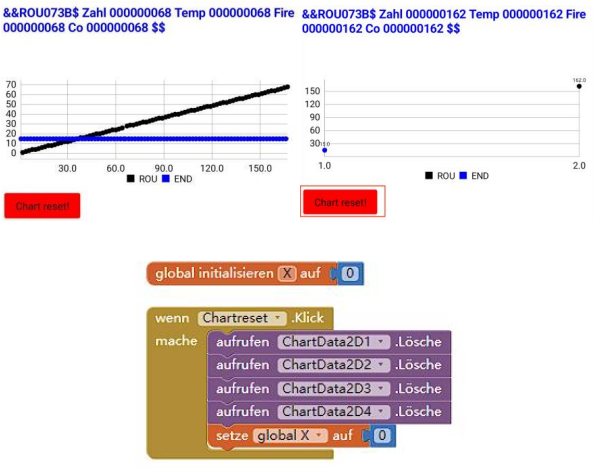
7.2.2.3 Diagrammerstellung

Die Temperatur ist ein wichtiger Sicherheitsindikator für die Beurteilung der Umgebung. Deshalb habe ich den Temperaturwert speziell in Form eines Diagramms erstellt, um die Temperaturänderung rückmelden zu können.

Die Temperatur aller Knoten kann gleichzeitig im Diagramm angezeigt werden und die Temperaturänderungen aller Knoten können gleichzeitig beobachtet werden.



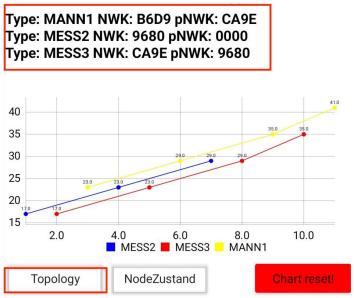
Wenn im Diagramm zu viele Daten aufgezeichnet sind und die Punkte im Diagramm zu dicht und schwer sichtbar sind, können Sie auf die Schaltfläche „Löschen“ klicken, um das Diagramm neu zu zeichnen.



7.2.2.4 Anzeige der Netzwerktopologie

Um die Mesh-Struktur des ZigBee-Netzwerks intuitiver anzuzeigen, habe ich die Topologie-Anzeigefunktion entwickelt. Klicken Sie auf die Schaltfläche „Topologie“, um den aktuellen Verbindungsstatus des ZigBee-Netzwerks anzuzeigen. Zu den angezeigten Informationen gehören die Knotennummer, die Kurzadresse des Knotens selbst und die Kurzadresse des übergeordneten Knotens. Auf diese Weise kann aus der übergeordneten Knotenadresse jedes Geräts umgekehrt auf die Struktur des gesamten Netzwerks geschlossen werden.

Das Bild zeigt die Netzwerkstruktur von zwei Knoten im Netzwerk, und es ist leicht zu erkennen, dass es sich um eine Baumstruktur handelt.



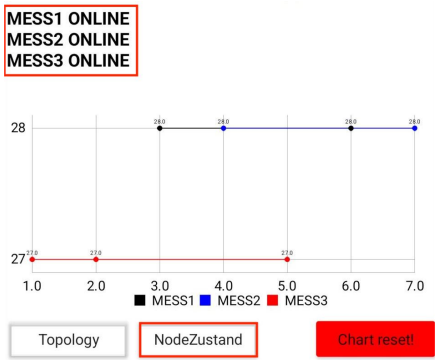
Wenn auf die Schaltfläche „Topology“ geklickt wird, werden Anweisungen über die serielle Schnittstelle an den Koordinator gesendet. Nach Erhalt der Anweisung aktiviert der Koordinator die „Netzwerktopologie“ in der oben genannten Zigbee-Funktion und sendet schließlich die zurückgegebenen Topologieinformationen über die serielle Schnittstelle an den Client, um die Funktion zur Anzeige der Netzwerktopologie zu realisieren

7.2.2.5 Anzeige des Knotenverbindungsstatus

Der Verbindungsstatus des Knotens kann beurteilt werden, indem beobachtet wird, ob der Knoten Daten normal sendet. Um jedoch den Verbindungsstatus des Knotens schnell zu überprüfen, habe ich diese Funktion entworfen. Klicken Sie auf die Schaltfläche NodeZustand, um den aktuellen Verbindungsstatus aller Geräte anzuzeigen.

Die Anzeigelogik dieser Funktions-APP stimmt mit dem obigen OLED-Bildschirm überein. Die Anzeigelogik besteht darin, dass der Bildschirm nur dann die Knoteninformationen anzeigt, wenn der Knoten zum ersten Mal tatsächlich mit dem Koordinator verbunden ist, anstatt direkt den Verbindungsstatus anzuzeigen jedes Knotens, um Missverständnisse über den Knotenstatus zu vermeiden.

Das Beispiel in der Abbildung zeigt, dass derzeit nur MESS-Knoten dem Netzwerk beitreten, also nur MESS-Knoten online sind. MANN-Knoten sind nicht aktiv, daher wird kein entsprechender Status angezeigt.

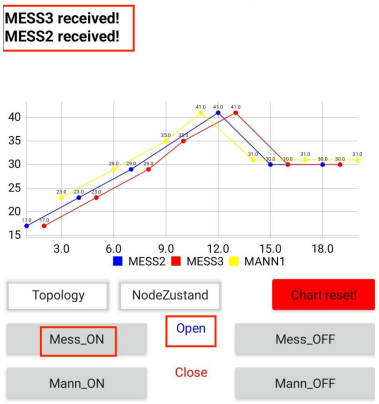


7.2.2.6 Knotensteuerung und Statusanzeigefunktion

Die Funktion der Knotensteuerungstaste besteht darin, den Knotensummer ein- und auszuschalten, der dazu dient, eine gefährliche Situation am Brandort zu simulieren und alle Feuermann mit einer Taste zur Evakuierung aufzufordern. Nachdem das Feuer gelöscht ist, kann der Summer des Sensorknotens eingeschaltet werden, um die Suche nach Geräten am Brandort zu erleichtern.

Mit der Knotensteuerungstaste können verschiedene Knotentypen unabhängig voneinander gesteuert und der Öffnungsstatus der entsprechenden Funktion überprüft werden. Um Fehleinschätzungen zu vermeiden, habe ich die Funktion zur Verhinderung von Fehleinschätzungen hinzugefügt. Diese Schaltflächen sind nur aktiviert, wenn das Telefon ordnungsgemäß mit dem WLAN und dem Server verbunden ist. Wenn das Mobiltelefon nicht ordnungsgemäß mit esp32WiFi verbunden ist, ändert sich der Status der Statusanzeigeleiste nicht, selbst wenn auf die Steuertaste geklickt wird.

Wenn Sie auf die Steuerschaltfläche klicken, wird in der Nachrichtenleiste die entsprechende Meldung angezeigt, die den Benutzer darüber informiert, dass der Knoten den Befehl empfangen hat und die Funktion normal aktiviert wurde.

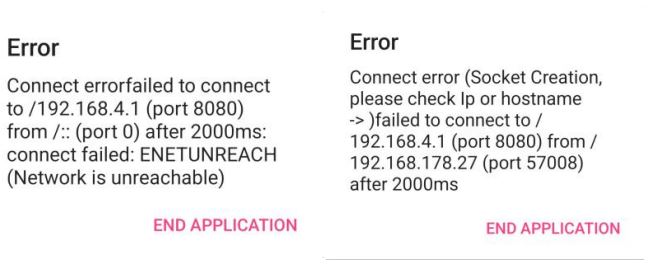


7.2.2.7 Fehlermeldung

Alle oben genannten Funktionen können nur unter der Voraussetzung einer normalen Verbindung verwendet werden. Allerdings kann das Gerät nicht immer eine normale

Verbindung herstellen. Wenn ein Problem auftritt und korrekte Eingabeaufforderungsinformationen vorliegen, kann der Benutzer das Problem schneller finden und lösen. Deshalb habe ich eine Erinnerung bei Verbindungsfehlern hinzugefügt, damit Benutzer Probleme schneller und besser finden und lösen können.

Wenn das Mobiltelefon nicht mit WLAN verbunden ist oder das Mobiltelefon mit anderen drahtlosen Netzwerken verbunden ist oder der Server nicht geöffnet werden kann, wird eine Fehlermeldung auf dem Bildschirm angezeigt und die aktuelle IP-Adresse des Mobiltelefons wird angezeigt die Information.



7.3 Kapitelzusammenfassung

Dieses Kapitel konzentriert sich auf die Einführung der App Inventor-Software und stellt die Verwendung der App Inventor-Software zum Programmieren der Anzeige von Übertragungsdaten über serielle Schnittstellen vor. Schließlich werden die Diagrammanzeige von Daten, die Anzeige der Netzwerktopologie und die Anzeige des Knotenverbindungsstatus realisiert.Gleichzeitig steuert der Client-System mit der Notevakuierungstaste den Ton- und Lichtalarm des Endknotens, um die Menschen daran zu erinnern, das Gebäude zu verlassen.

8 PCB-Design

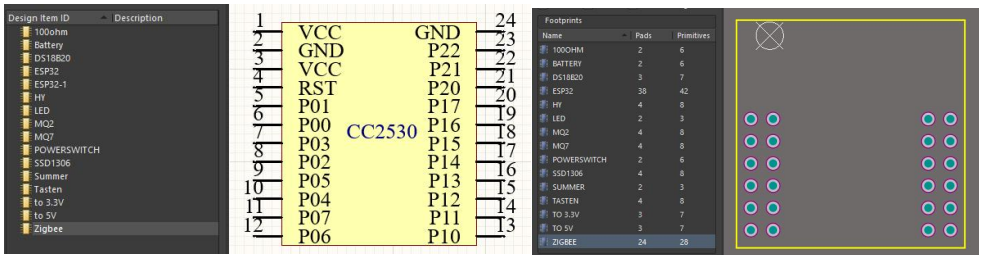
8.1 Komponentenschema und Verpackungsproduktion

Ich habe schematische Diagramme und Pakete aller verwendeten Komponenten und der Punkte erstellt, die beachtet werden müssen:

1 Komponentenübersicht.

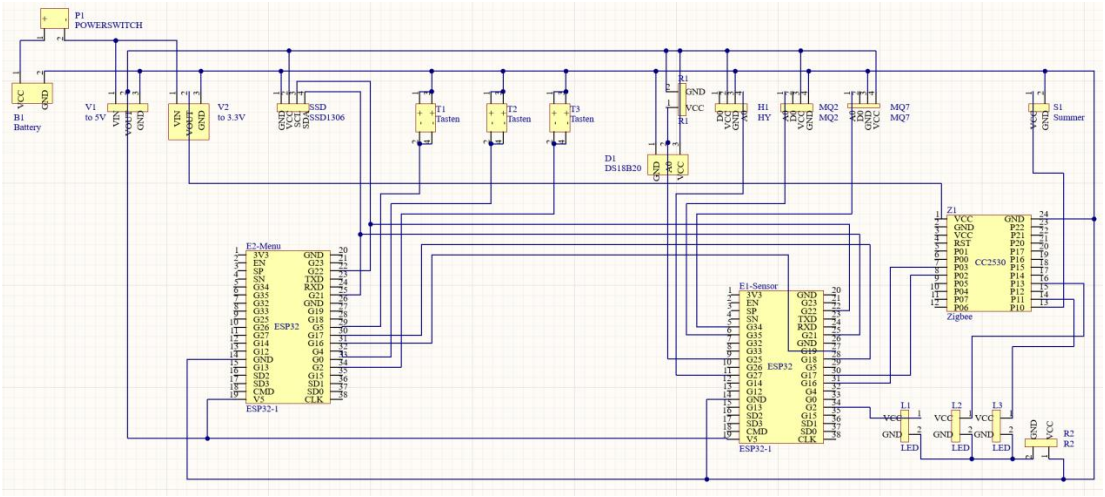
2 Die Pin-Nummer der Komponente.

Der Rastermaß der 3 Komponenten beträgt 2,54 mm.

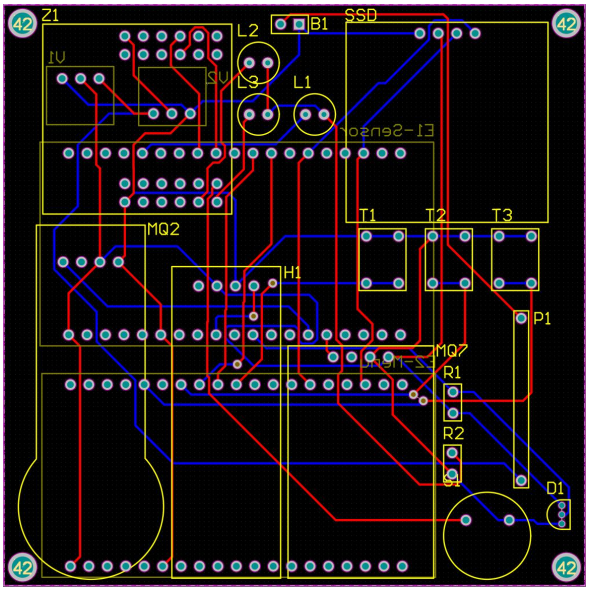


85Komponentenschema und Verpackungsproduktion

8.2 Schaltplan- und Leiterplattendesign



86Schaltplan der Leiterplatte



87 Leiterplatteverkabelung

Beim Routing auf der Leiterplatte habe ich folgende wichtige Faktoren berücksichtigt:

1.Signalintegrität: Ich stelle sicher, dass das Signal bei der Übertragung auf der Leiterplatte nicht beschädigt oder verzerrt wird. Durch die Anpassung der Anordnung der Komponenten versuche ich, die Signalleitungen gleich lang zu machen, zu lange Signalleitungen zu vermeiden und Interferenzen zu vermeiden zwischen Signalleitungen.

2.Stromversorgung und Erdungskabel: Um eine stabile Stromversorgung und einen stabilen Rückweg zu gewährleisten, achte ich darauf, die Verkabelung von Stromversorgung und Erdungskabel zu optimieren, die Schleifenimpedanz zu reduzieren und Rauschstörungen zu reduzieren.

3.Breite und Abstand der Signalspuren: Entsprechend den Eigenschaften und der Übertragungsrate des Signals wähle ich die geeignete Spurbreite und den entsprechenden Abstand aus, um die Anforderungen an Impedanzanpassung und Signalintegrität zu erfüllen.

4.Gerätelayout:Ich ordne elektronische Komponenten und Geräte rational an und minimiere die Länge und Kreuzung von Signalleitungen, um Störungen zu reduzieren.

5.PCB-Hierarchieplanung: Ich kann ein zweischichtiges PCB-Design übernehmen, um die Verdrahtungsdichte und den Signalisolationseffekt zu verbessern.

6.Sicherheitsabstand:Ich stelle eine angemessene Sicherheitstrennung zwischen verschiedenen Stromkreisen sowie zwischen Stromkreisen und Grenzen sicher, um Probleme mit der elektrischen Isolierung zu vermeiden.

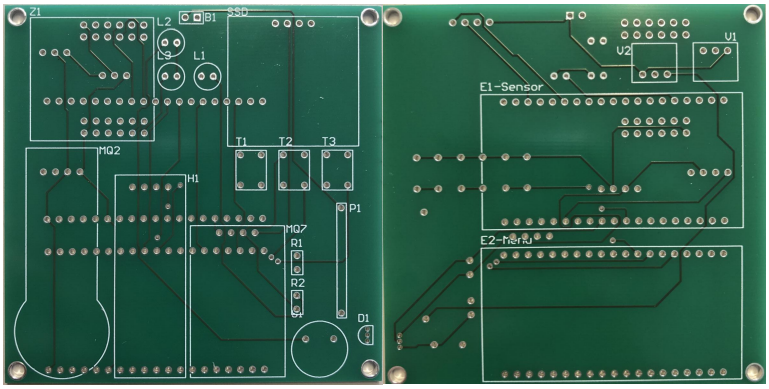
7.EMI/EMC-Design:Ich berücksichtige elektromagnetische Störungen und elektromagnetische Verträglichkeit und ergreife entsprechende Abschirmungs- und Filtermaßnahmen. Ich platziere den Einzelchip-Mikrocomputerchip auf der Rückseite der Leiterplatte, um sicherzustellen, dass das Schaltungssystem in verschiedenen Umgebungen normal funktionieren kann.

Schließlich habe ich eine Platine entworfen, die die Anforderungen aller Knoten erfüllt. Eine Platine kann die Funktionen aller Geräte realisieren und muss nur je nach Knotentyp unterschiedliche Hardware löten. Dies kann die Schweißeffizienz verbessern und die Fehleranfälligkeit beim Schweißen verringern.

9 Abschlussprüfung

9.1 Präsentation des fertigen Produkts

9.1.1 Leiterplattenanzeige

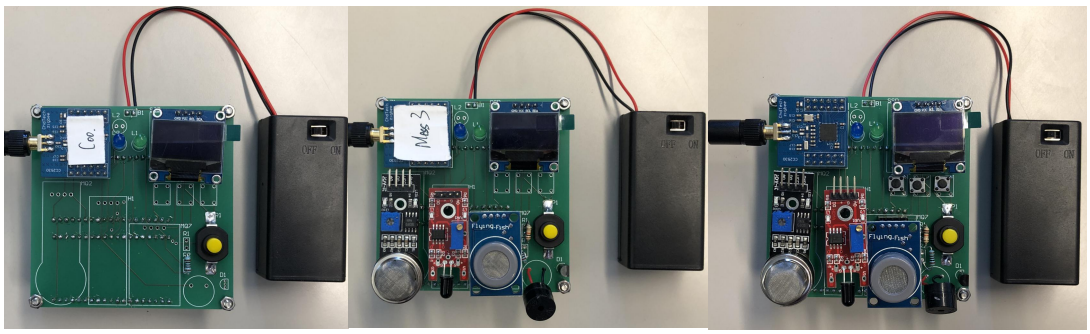


88Leiterplatte

Lagen	*2 Lagen*
Größe (x/y)	*80,0mm x 80,0mm = 0,64dm²*
Oberfläche	*HAL Bleifrei*
Leiterbahnabstand, -breite, Restring	*100µm*
Lötstopp	*doppelseitig, grün*
Positionsdruck	*doppelseitig, weiß*

11Leiterplatte-Parameter

9.1.2 Fertigen Produkt

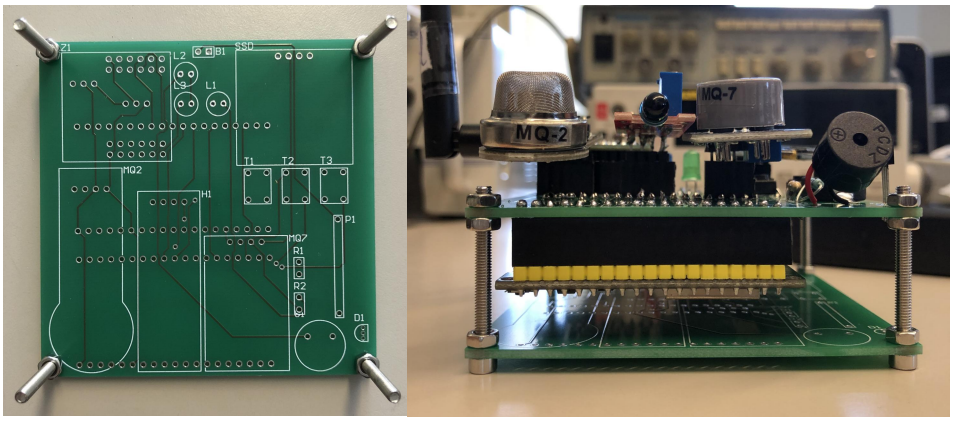


89Fertigen Produkt

Die Abbildung zeigt die Endprodukte des Koordinators, des Sensorknotens und des Feuermannknotens.

Design der Gerätestabilität

Um eine stabile Platzierung der Geräte zu ermöglichen, verwende ich geschickt gestapelte Leiterplatten mit Bolzen, um eine feste Struktur aufzubauen. Das Gerät kann stabil platziert werden und sorgt gleichzeitig für eine Wärmeableitung. Wie im Bild rechts gezeigt, sorgt die Position der Einstellmutter dafür, dass das Gerät eine stabile Struktur bildet. Zu diesem Zeitpunkt kann das Gerät problemlos auf dem Tisch platziert werden und jedes komponenten verfügt über ausreichend Platz für die Wärmeableitung.

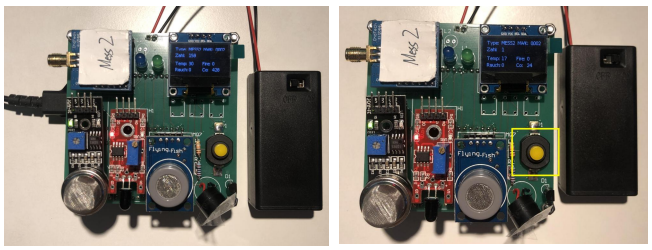


90gestapelte Leiterplatten mit Bolzen

Das Bild rechts zeigt auch eine weitere versteckte Idee von mir: Um zu realisieren, dass die Bauteile nach Abschluss des Projekts von anderen wiederverwendet werden können, habe ich die Bauteile nicht direkt auf die Platine gelötet, sondern Header verwendet. Auf diese Weise können die komponenten einfach von der Leiterplatte entfernt werden.

Zwei Stromversorgung Methoden

Ich habe zwei Designs für die Stromversorgung des Systems bereitgestellt.Unter Berücksichtigung der Einsatzszenarien und der Tragbarkeit des Geräts habe ich zusätzlich zur Verwendung der grundlegenden USB-Stromversorgung zusätzlich eine Batterie zur Stromversorgung jedes Knotens entwickelt und einen Hauptnetzschalter (gelbes Kästchen in rechts Bild)hinzugefügt.Daher stehen dem Gerät zwei Stromversorgungsmethoden zur Auswahl.

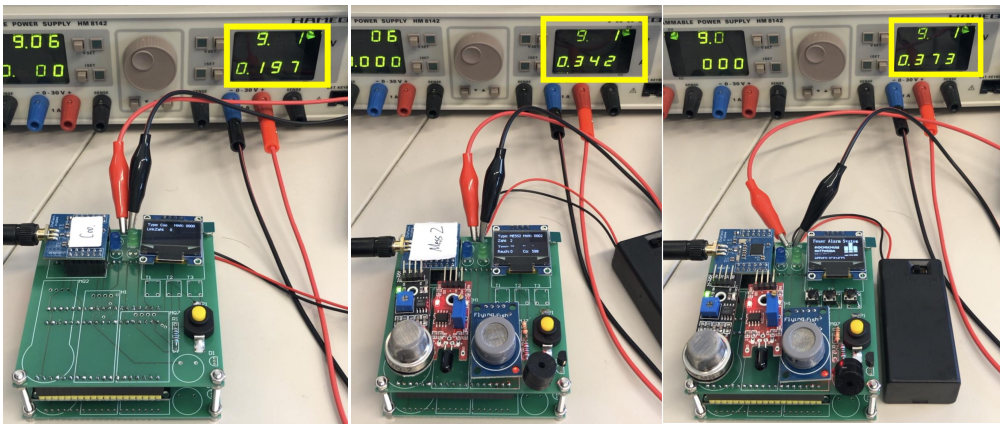


91Stromversorgung mit USB(links) und Batterie(rechts)

9.1.3 Stromverbrauch Test

Um den Gesamtstromverbrauch des Geräts zu testen, verwende ich ein einstellbares Netzteil und stelle die Eingangsspannung auf 9 V ein, um den Batteriestrom zu simulieren. Nach dem Testen habe ich den Arbeitsstrom des Koordinators, des Sensorknotens und des Feuerwehrmannknotens ermittelt.

Das Endergebnis ist, dass der Betriebsstrom des Koordinators nur etwa 200 mA(1.8w) beträgt. Der Gesamtstromverbrauch ist sehr niedrig. Der Betriebsstrom des Sensorknotens beträgt ca. 350 mA(3.15w). Da der Knoten den Sensor mit Strom versorgen muss, ist der Strom gestiegen. Der Arbeitsstrom des Feuermannknotens beträgt etwa 380 mA(3.42w). Der Feuermannknoten verfügt über einen zusätzlichen ESP32, um mehrstufige Menüfunktionen bereitzustellen, sodass der Strom im Vergleich zugenommen hat zum Sensorknoten.

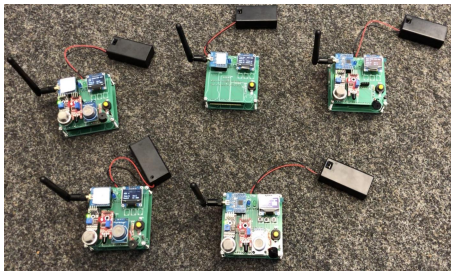


92Stromverbrauch Test

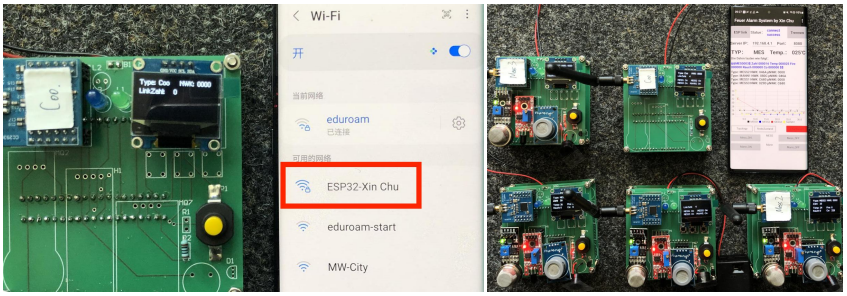
9.2 Demonstration des Systembetrieb Prozesses

9.2.1 Laufender Prozess:

Drücken Sie zunächst den Netzschalter des Koordinatormoduls. Drücken Sie die Netzschalter des Sensorknotens bzw. des Feuermannknotens, nachdem die Boot-Animation beendet ist, und das Endgerät muss die Boot-Sequenz nicht berücksichtigen. Nachdem jeder Knoten eingeschaltet wurde, beginnt der Bildschirm mit der Anzeige von Daten. Mit die Tasten des Feuermannknotens können Sie den Bildschirm umschalten, um verschiedene Funktionsinhalte anzuzeigen. Beobachten Sie dann den Bildschirm des Koordinatormoduls. Sie können sehen, dass der Verbindungsstatus jedes Knotens auf dem Bildschirm angezeigt wird. Stellen Sie dann mit Ihrem Mobiltelefon eine Verbindung zum WLAN mit dem Namen „ESP32-Xin Chu“ her. Öffnen Sie die APP und klicken Sie links auf die Schaltfläche „ESP Link“. Danach werden die von jedem Knoten gesendeten Daten auf dem Bildschirm angezeigt. Simulieren Sie zu diesem Zeitpunkt eine gefährliche Situation am Brandort, klicken Sie auf die Schaltfläche auf dem Mobiltelefonbildschirm und schalten Sie den Feuerwehrknotenalarm ein. Simulieren Sie dann die Wiederherstellung des Geräts, klicken Sie auf die Schaltfläche auf dem Mobiltelefonbildschirm und aktivieren Sie den Sensorknotenalarm, um die Suche nach dem Gerät zu erleichtern.



93 Alle fünf Geräten

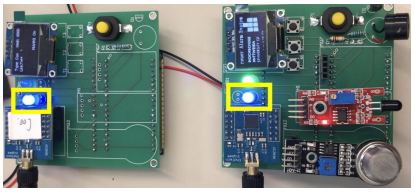


94 Hardware mit Software

9.2.2 LED Funktion

LED Blink

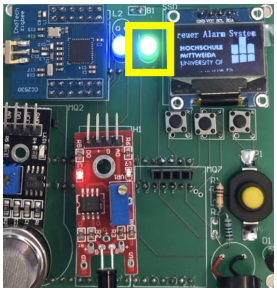
Wenn der Endknoten Informationen sendet und der Koordinator Informationen empfängt, wird dies von einem LED-Blinken begleitet. Da die Übertragungszeit sehr kurz ist, blinken beide fast gleichzeitig. Der Schusseffekt ist in der folgenden Abbildung dargestellt. Die blauen LED-Leuchten auf der linken Seite der Koordinator- und Feuermannknoten leuchten gleichzeitig.



95LED Blink

LED Alarm

Wenn der Sensorwert den voreingestellten Alarmwert überschreitet, leuchtet die LED rechts auf.

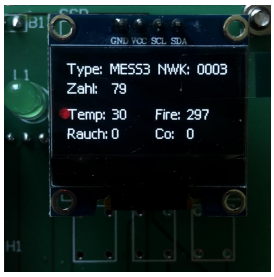


96 LED Alarm

9.2.3 Sensordaten Anzeige für Messknoten.

Sensorknoten zeigen Sensordaten korrekt an.

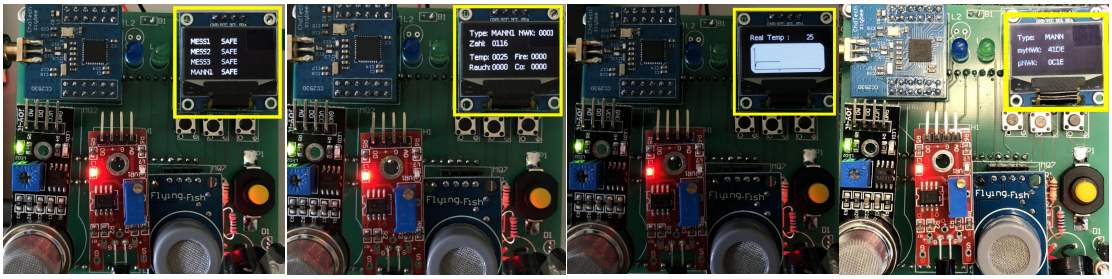
Da der Rauchsensor aufgewärmt werden muss, bevor er normal verwendet werden kann, sind die Sensordaten zu Beginn nach dem Einschalten des Geräts relativ groß und der Sensorwert wird mit zunehmender Erwärmung des Sensors immer kleiner. In einer Umgebung ohne Rauch geht der Wert schließlich auf etwa 0 zurück.



97Sensordaten Anzeige

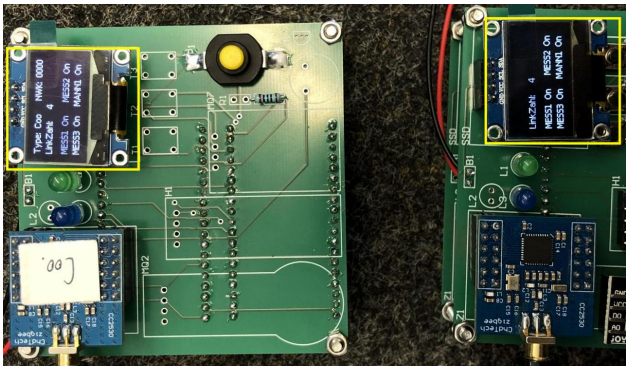
9.2.4 Mehrstufige Menüfunktionsanzeige für Feuermannknoten

Der in der folgenden Abbildung dargestellte Inhalt ist die Anzeige der **Sicherheitsstatus, Sensordaten,Diagramm** und **Netzwerktopologie** der Knoten.



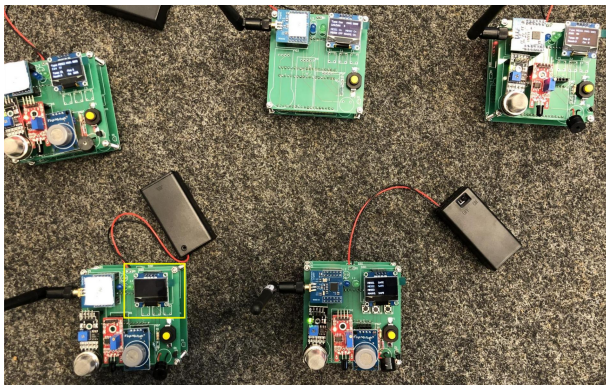
98Menüfunktion

Wenn der Feuermannknoten die Funktion zum Anzeigen des Geräteverbindungsstatus aktiviert, können sowohl der Koordinator als auch der Feuermannknoten schnell den aktuellen Verbindungsstatus aller Geräte anzeigen. Wie in der Abbildung gezeigt, sind zu diesem Zeitpunkt beispielsweise alle Knoten normal verbunden.



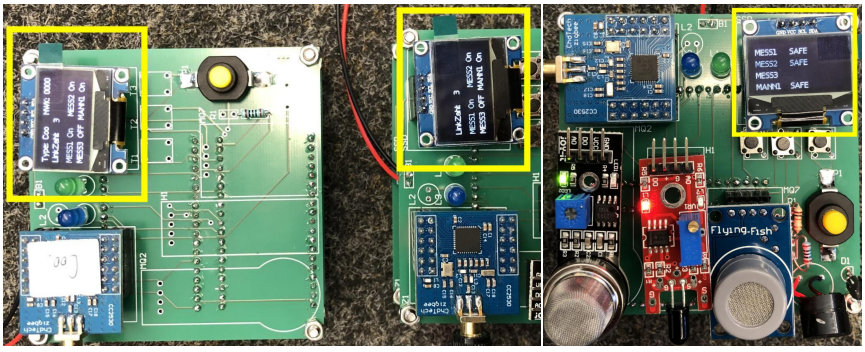
Als nächstes simuliere ich die Trennung eines Endgeräts.

Wie in der Abbildung gezeigt, habe ich den Netzschalter von Sensorknoten 3 in der unteren linken Ecke ausgeschaltet, woraufhin das Gerät ausgeschaltet wurde und die Verbindung verloren ging.



99Trennung eines Endgeräts simulieiren

Nach einiger Zeit zeigen die Koordinator- und Feuermannknoten an, dass Sensorknoten 3 das Netzwerk verlassen hat. Wenn der Feuermannknoten auf die Funktion zum Anzeigen des Knotensicherheitsstatus umschaltet, werden die Statusinformationen von Sensorknoten 3 nicht mehr angezeigt und die Statusinformationen anderer Knoten werden normal angezeigt.

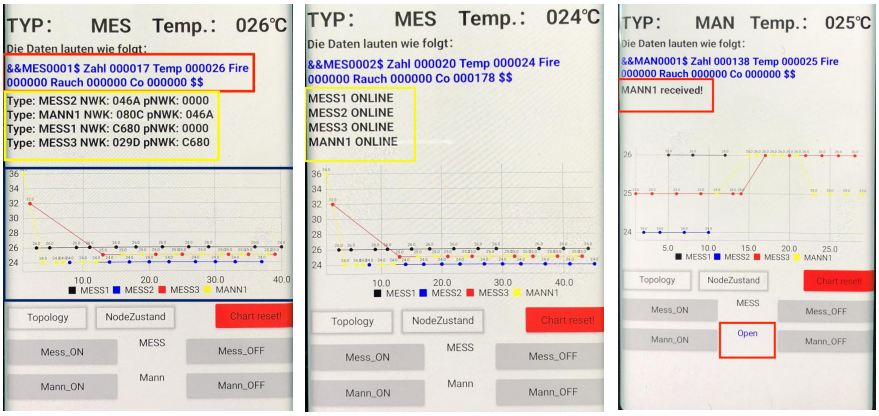


9.2.5 Mobile APP-Funktion

Klicken Sie auf verschiedene Schaltflächen auf dem Bildschirm, um verschiedene Funktionen zu aktivieren.Der in der folgenden Abbildung dargestellte Inhalt ist die Anzeige der **Netzwerktopologie,Diagramm,Verbindungsstatus und Feedback** Funktionen.

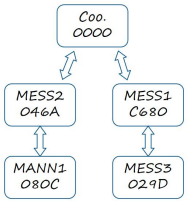
Unterschiedliche Farben repräsentieren unterschiedliche Knoten im Diagramm.

Das Gerät antwortet „XXXX received!“nach Erhalt des Befehls.



100Mobile APP-Funktion

Basierend auf den Informationen des übergeordneten Knotens, die von den Knoten im Bild zurückgegeben werden, kann die aktuelle Netzwerktopologie umgekehrt berechnet werden.



101aktuelle Netzwerktopologie

Bisher wurden alle Funktionen dieses Projekts demonstriert, alle Geräte können gut funktionieren und alle Funktionen wurden erfolgreich implementiert.

Hinweis： Wenn Sie die von mir erstellte APP auf Ihrem Android Telefon installieren möchten, finden Sie die entsprechende .aia-Datei auf der CD. Importieren Sie die Datei auf der offiziellen APP INVENTOR-Website, um das Projekt zu generieren. Generieren Sie dann die APK-Datei, um den Downloader zu installieren.

9.3 Kapitelzusammenfassung

Der Inhalt dieses Kapitels besteht darin, die Leiterplattenproduktion und das Schweißen sowie den Systemtest abzuschließen. Durch eine Demonstration konnte ich nachweisen, dass das System erfolgreich lief und alle Funktionen in Ordnung waren und die Projektziele erreicht wurden. Es ist ein perfekter Abschluss für das Abschlussprojekt.

Zusammenfassung

In Masterarbeit werden die Merkmale der Zigbee-Technologie und ihre Anwendung im Bereich der Feueralarmierung zusammengefasst und ein Entwurfsschema für ein drahtloses Feueralarm-Informationsübertragungssystem basierend auf der ZigBee-Technologie vorgeschlagen. In diesem Artikel werden hauptsächlich die folgenden Inhalte untersucht:

1. Ich habe die technischen Eigenschaften, die Protokollspezifikation, die Struktur, die Topologie und den Gerätetyp von ZigBee untersucht und die Anwendung der ZigBee-Technologie in Systemen zur Übertragung von Brandmeldeinformationen analysiert

2.Leiterplatten-Hardwaredesign mit dem Prozessor als Kern. Ich habe ein drahtloses Feuer Alarm System mithilfe von CC2530 aufgebaut, einem drahtlosen Hochfrequenzchip, der auf dem ZigBee-Standard basiert. Die Hauptfunktion besteht darin, den Koordinatorknoten zu steuern, um das Empfangen und Senden von Daten abzuschließen, den Sensorknoten zu steuern, um Daten zu sammeln, das APP-Steuersignal zu empfangen und das Empfangen und Senden von Daten abzuschließen.

3.Der Koordinatorknoten kommuniziert über die serielle Schnittstelle mit dem Host-Computer. Ich verwende die APP Inventor-Software, um die Host-Computer-Schnittstelle zu entwerfen, um die Anzeige von Sensordaten, die Anzeige numerischer Temperaturdiagramme und die Steuerung von Endknoten zu realisieren.

4.Ich habe das Funktionsprinzip des Temperatursensors, des Rauchsensors usw. analysiert und tatsächliche Tests durchgeführt.

5.Das Softwaredesign des Systems umfasst hauptsächlich das Design des Sendeprogramms und des Empfangsprogramms des Koordinatorknotens und des Sensorknotens sowie das Design des Kommunikationsprogramms für die serielle Schnittstelle zwischen dem Koordinator und dem oberen Computer. Das gesamte Programm wird implementiert in C-Sprache basierend auf dem Z-Stack-Protokollstapel. Ich programmiere diese Funktion, indem Sie Aufgaben und Ereignisfunktionen festlegen.

Name	Drahtloses Feuer Alarm System
Größe(Länge*Breite*Höhe)	8cm*8cm*5cm
Gewicht	max 150g
Übertragungsmethode	Zigbee+WLAN
System Anzahl	1 Gateway+3 MESS Geräte+1 MANN Geräte
Übertragungsabstand	47m(ohne Hindernis)
Messfrequenz	10s
Netzwerksturktur	Mesh-Neztwerk
Messdata	Temperatur,Rauch,Feuer
Visualisierung	Summer und LED

12 Parameter des Drahtloses Feuer Alarm System

Die Ergebnisse zeigen, dass die Brandmeldeanlage kostengünstig ist und stabil funktioniert. Im Vergleich zum herkömmlichen kabelgebundenen Brandmeldesystem beseitigt dieses kabellose Brandmeldesystem die Nachteile des herkömmlichen Systems, wie z. B. feste Drahtverbindung, leichte Alterung oder Korrosion, Rattenbiss, Verschleiß, hohe Ausfallrate und Fehlalarmrate. Aufgrund der Verwendung drahtloser Übertragungstechnologie weist das System jedoch Mängel wie die Sicherheit der Informationsübertragung und eine relativ geringe Kommunikationsentfernung zwischen Knoten auf. In Zukunft kann man das System weiter optimieren und verbessern.

Literaturverzeichnis

Brände:https://de.statista.com/statistik/daten/studie/155263/umfrage/entwicklung-der-gesamtanzahl-der-braende-in-deutschland-seit-2002/

Zigbee:https://www.ti.com.cn/cn/lit/ds/symlink/cc2530.pdf?ts=1674877299072&ref_url=https%253A%252F%252Fwww.ti.com.cn%252Fproduct%252Fcn%252FCC2530

ESP32:https://www.espressif.com/en/products/devkits

TI CC2530:https://www.ti.com.cn/product/cn/CC2530

SSD1306:https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf

PT100:https://www.omega.de/prodinfo/widerstandsfuehler-baugruppen.html

MQ-2:https://www.pololu.com/file/0J309/MQ2.pdf

ArduSpreadsheet:https://circuitjournal.com/arduino-serial-to-spreadsheet

DS18B20:https://www.analog.com/media/en/technical-documentation/data-sheets/DS18B20.pdf

CC Debugger:https://www.waveshare.net/txt/CC-Debugger-Interface.htm

APP Inventor 2：https://ai2.appinventor.mit.edu

Anhang

Stückliste

Name	Menge
Zigbee CC2530	5
ESP32	7
DS18B20	4
PT100	3
HY-A1	4
MQ2	4
MQ7	4
Tasten	3
SSD1306	5
Summer und LED	14
100Ohm Widerstand	5
10K Ohm Widerstand	5
PCB-Board	10
Steckbrett	3
Bolzen	20
Laptop	1
Kabel	viele

13Stückliste

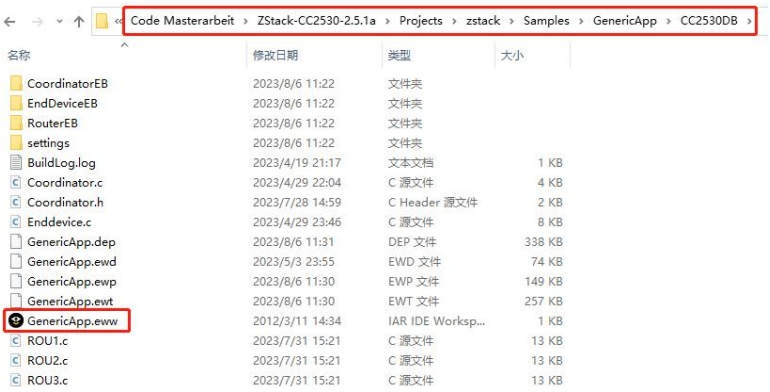
Programmiercode

Da der Codeinhalt zu groß ist, habe ich den gesamten Code auf die CD gelegt.

Zur Vereinfachung für die Leser habe ich die Codes nach Gerätetyp kategorisiert.

102Programmiercode

Hinweis: Im Ordner „Zigbee Firmwares“ werden die Brenndateien gespeichert. Der Zigbee-bezogene Code befindet sich im Ordner ZStack-CC2530-2.5.1a. Der spezifische Pfad ist in der Abbildung dargestellt.



103spezifische Pfad des Zigbee Code

Eigenständigkeitserklärung

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.

Mittweida,08.08.2023 Xin Chu