

Diplomarbeit

Sensorphalanx für Heimautomation

Projektbezeichnung Thema

Konzipierung und Entwicklung einer Sensorphalanx für Smart Home

Lehrgang, Jahrgang, Klasse

6. Automation 18H-21H

Diplomand

Beckmann, Raphael
Gehrenstrasse 21
9230 Flawil
Telefon: 079 277 96 13
E-Mail: Raphael.Beckmann1983@gmail.com

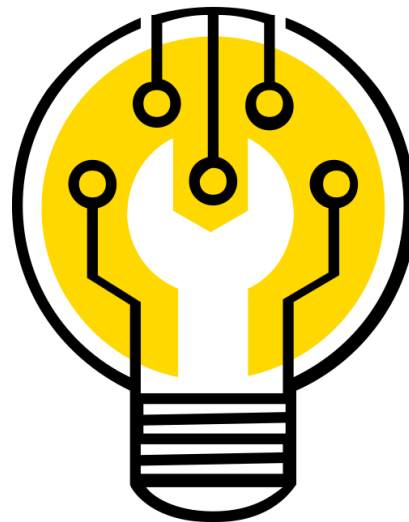
Betreuer

Experte, / Expertin

Auftraggeber

Der Diplomand

Grafik und Projektmotto



Let's build things!

Vorwort

Management Summary

Am Schluss schreiben!!!

Ausgangslage und Zielsetzung

Vorgehen

Ergebnisse

Inhaltsverzeichnis

1	Ausgangslage	1
1.1	Problemstellung.....	1
1.2	Zielsetzung	1
1.2.1	Wunschziel	2
1.3	Persönliche Motivation	2
1.4	Rahmenbedingungen und Abgrenzungen	3
1.5	Termine und Lieferobjekte	4
1.5.1	Meilensteinplan.....	4
1.5.2	Terminplan.....	5
1.5.3	Lieferobjekte	7
1.6	Präferenzmatrix für Projektkriterien	7
2	Anforderungsanalyse und Konzeption	10
2.1	Ziel und Nutzen Sensorphalanx	10
2.2	Funktionale Anforderungen.....	10
2.2.1	Allgemeine Systemfunktionen	10
2.3	Nicht-Funktionale Anforderungen.....	12
2.3.1	Allgemeine Anforderungen (Rahmenbedingung)	12
2.4	Komponentenauswahl.....	12
2.4.1	Smart Home Systeme.....	12
2.4.1.1	Open Source	13
2.4.1.2	Close Source.....	13
2.4.1.3	Entscheidungsbegründung	13
2.4.1.4	Open Source Smart Home Systeme.....	14
2.4.1.4.1	openHAB.....	15
2.4.1.5	Home Assistant	17
2.4.1.5.1	ioBroker.....	19
2.4.1.5.2	Entscheidungsbegründung	20
2.4.1.6	Analyse Siegervariante.....	20

2.4.2	Kommunikation / Protokoll	21
2.4.2.1	MQTT	21
2.4.2.2	HTML.....	23
2.4.2.3	???????	23
2.4.2.4	Entscheidungsbegründung	23
2.4.3	Kommunikation innerhalb der Sensorephalanx	23
2.4.3.1	Inter-Integrated Circuit (I2C).....	23
2.4.3.2	Serial.....	23
2.4.4	Hardware Smart Home System.....	23
2.4.4.1	Raspberry Pi.....	23
2.4.4.1.1	Version 3 B Spezifikation	24
2.4.4.1.2	Version 3 B+ Spezifikation.....	24
2.4.4.1.3	Version 4 B Spezifikation	25
2.4.4.1.4	Entscheidungsbegründung	25
2.4.5	Hardware Sensorphalanx.....	26
2.4.5.1	Mikroprozessor.....	26
2.4.5.1.1	Arduino.....	26
2.4.5.1.2	ESP8266.....	26
2.4.5.1.3	ESP32.....	27
2.4.5.1.4	Entscheidungsbegründung	29
2.4.5.2	Energieversorgung	30
2.4.5.2.1	Lithium-Ionen-Akkumulator.....	30
2.4.5.2.2	Lithium-Polymer-Akkumulator.....	31
2.4.5.2.3	Entscheidungsbegründung	31
2.4.5.3	Laden.....	32
2.4.5.4	Sensoren.....	32
2.4.5.4.1	Bewegungserkennung	32
2.4.5.4.2	Helligkeit.....	33
2.4.5.4.3	Ladezustandserkennung und Anzeige	33
2.4.5.4.4	Luftqualität	34
2.4.5.4.5	Temperatur	35
2.4.5.4.6	Kombinationssensor: Temperatur & Feuchtigkeit.....	35
2.4.5.4.7	Kombinationssensor: Temperatur & Luftdruck.....	36
2.4.5.4.8	Kombinationssensor: Temperatur, Feuchtigkeit & Luftdruck	36
2.4.5.4.9	Kombinationssensor: Temperatur, Feuchtigkeit, Luftdruck & Luftqualität	37
2.4.5.4.10	Varianten Übersicht.....	37
2.4.6	Zusammenfassung Hardwarebestellung.....	37
3	Umsetzung.....	38
3.1	Raspberry	38
3.1.1	Installieren.....	38
3.1.2	Einrichten	38
3.1.3	Testen	38
3.2	Schaltplan entwerfen	38
3.3	Lochraster Prototyp.....	38

3.3.1	Aufbau	39
3.3.2	Testen	39
3.4	PCB Design.....	39
3.5	Gehäuse entwerfen.....	39
3.6	Software entwickeln.....	39
3.6.1	Kommunikation aufbauen	39
3.6.2	Daten Sammeln	39
3.6.3	Daten Schicken.....	39
3.7	Datenauswertung -aufbereiten auf dem Raspberry	39
3.7.1	Datenspeicherung.....	39
3.7.2	Visualisierung	39
4	Fazit.....	40
	Literaturverzeichnis.....	I
	Abbildungsverzeichnis.....	II
	Tabellenverzeichnis	III
	Anhänge.....	III
	Eidesstattliche Erklärung.....	III

1 Ausgangslage

“Smart Home” und “Home Automation” erfreuen sich immer grösser Beliebtheit. Fast jeder Neubau wird mit einer zentralen Steuerungseinheit gebaut, die mindestens das Raumklima steuert. Darüber hinaus gibt es viele Varianten und Zusatzfunktionen von einfachen Steckdosen- und Einbaugeräteschaltungen über Energiemanagement bis hin zur vollautomatischen Zugangs- und Zutrittsregelungen.

Allerdings gibt es bei Bestandsgebäuden einige Hindernisse, die mit konventionellen Produkten nicht oder nur unzureichend abgedeckt werden können.

Beispielsweise kann in Mietwohnungen nur sehr begrenzt Eingriff in die Elektroinstallation vorgenommen werden. Da es viele Maker*innen gibt, die ebenfalls eine Möglichkeit haben möchten, ihre Wohnung auch im Automationsbereich zu gestalten, besteht hier ein Markt für Bausätze. Daher ist es sinnvoll, einen Bausatz zu entwickeln, der ihnen die Möglichkeit einräumt, ihre eigenen Ideen umzusetzen und auch ermöglicht, die Sensordaten dort zu sammeln, wo sie benötigt werden und nicht, wie bei vielen konventionellen Produkten, am Aktor selber.

Ebenso ist das Thema IT-Sicherheit ein wichtiger Punkt. Immer wieder werden gravierende Sicherheitslücken öffentlich diskutiert. Ein Beispiel hierfür sind WLAN-kompatible Leuchtmittel, die unter Umständen den unverschlüsselten Standort und die WLAN-Daten speichern oder ohne direkte Einwilligung des Nutzers an den Herstellern übermitteln. Ein weiterer Faktor ist auch die anzustrebende Unabhängigkeit von Herstellerressourcen wie Cloud-Service und App-Entwicklung, da diese ohne konkreten Einfluss des Nutzers offline genommen oder nicht mehr weiterentwickelt werden können.

1.1 Problemstellung

Die Herstellung eines Bausatzes einer Sensorphanx, die es Maker*innen ermöglicht, Raumdaten zu sammeln und für ihr Smart Home zu nutzen.

1.2 Zielsetzung

Teil dieses Projektes wird es sein, einen lokalen Server für Home Automation aufzusetzen. Vorzugsweise soll dies auf eine günstige und gut verfügbare Hard- und Softwarebasis realisiert werden.

Der Fokus liegt auf der Entwicklung einer Hardware Sensorphanx diese soll modular aufgebaut werden, sodass es dem User möglich ist, anwendungsspezifisch einzelne Sensormodule zu verbauen. Zur Kommunikation soll WLAN verwendet werden. Die

Sensorphalanx wird keinen direkten Kontakt mit dem Internet haben, sondern nur mit dem Home Automation-Server kommunizieren.

Die Sensoren Phalanx soll keine CE-Zertifizierung erhalten müssen, da sie nur als rudimentärer Bausatz und in Form von Plänen und Programmcode für erfahrene User zugänglich gemacht werden soll.

Weiter soll die Phalanx über keine Aktoren, die über ein Feedback für den User hinausgehen, verfügen. Da die Phalanx an zentralen Punkten in einer Mietwohnung montiert werden soll, liegt ein Schwerpunkt auch auf der Stromversorgung. Diese soll mittels eines Elektrospeichers erfolgen, der auf einfachem Wege wieder aufladbar sein muss.

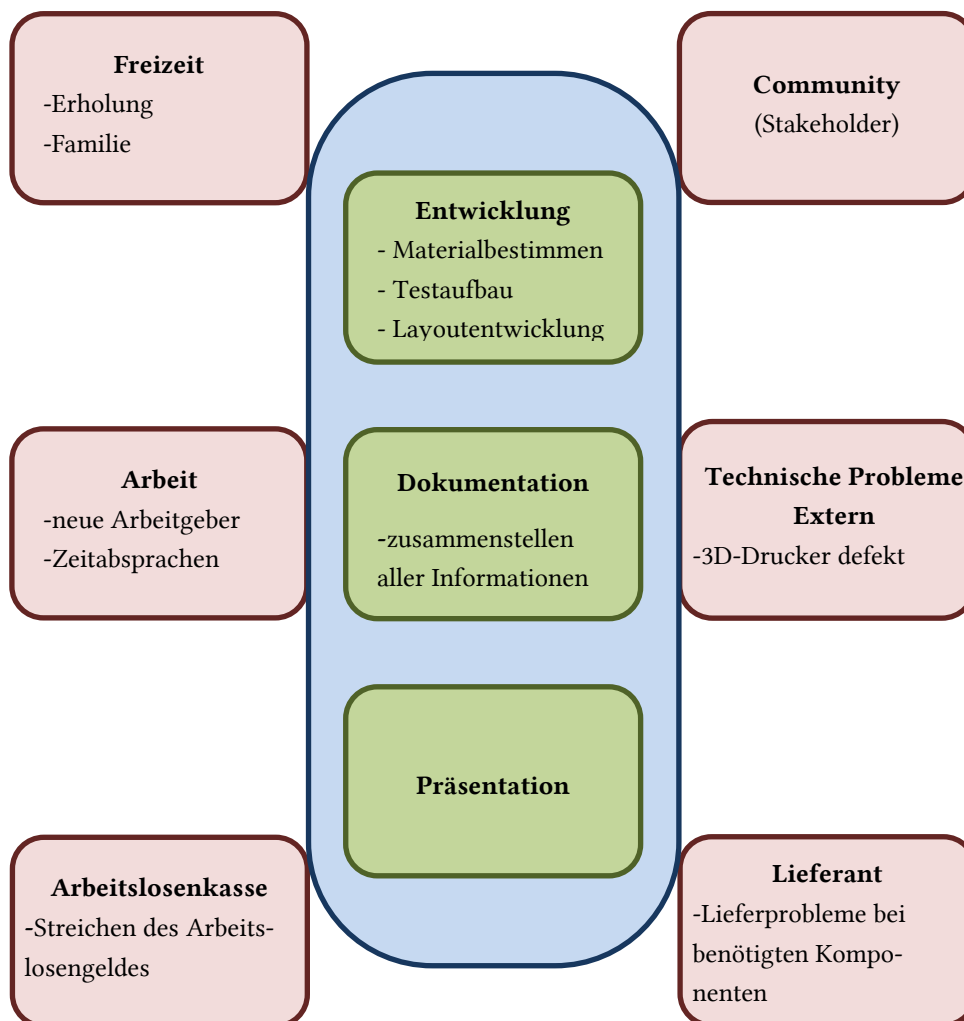
1.2.1 **Wunschziel**

Um anderen Maker*innen zu ermöglichen, sich die Sensorphalanx selber bauen zu können oder auf der Grundlage des Ergebnisses weiter zu entwickeln, sollen zum Abschluss der Diplomarbeit alle erarbeiteten Inhalte online verfügbar gemacht werden.

1.3 **Persönliche Motivation**

Der Autor möchte schon länger die Möglichkeit haben, in seiner Mietwohnung ein zusammenhängendes Smart Home System einzusetzen und um sinnvolle und auf die persönlichen Bedürfnisse ausrichten zu können, hat er sich entschlossen, dieses selbst zu realisieren.

1.4 Rahmenbedingungen und Abgrenzungen



Die Diplomarbeit soll ein funktionsfähiger Prototyp einer Sensoreinheit für die Home Automation ergeben. Diese Einheit soll so dokumentiert werden, dass ein Nachbau für kundige Maker*innen gut möglich ist.

Die Schnittstellen nach aussen sind in diesem Projekt nicht zahlreich, da der Diplomand ebenfalls Auftraggeber wie auch Endnutzer ist. Da das Ergebnis aber im Anschluss der interessierten Öffentlichkeit als Bausatz angeboten werden soll, ist vorgesehen, während der Diplomarbeit den Fortschritt auf einen Blog festzuhalten und wenn möglich mit Videos zu begleiten. Dieses ist nicht Teil der Diplomarbeit und erfolgt parallel, sofern dem Diplomanden genug Ressourcen zur Verfügung stehen.

1.5 Termine und Lieferobjekte

1.5.1 Meilensteinplan

Nr.	Meilenstein	Plan Datum	Ist Datum
1	Grobkonzept	30.08.2021	30.08.2021
2	Feinkonzept	28.10.2021	
3	Hausleitsystem betriebsbereit	04.11.2021	
4	Lochrasterprototyp	09.11.2021	
5	Sensorsoftware betriebsbereit	23.11.2021	
6	PCB / Gehäuse prototyping	06.12.2021	
7	Diplomarbeit Abschluss	05.01.2022	

Tabelle 1: Meilensteinplan

1.5.2 Terminplan

Monat		August														Oktober														November													
KW	Datum	31	31	32	33	34	35	40	41	42	43	44	45	46	47	48																											
ZfW Termine	Termin Zeit																																										
Urlaub																																											
MS 1	30.08.21						X																																				
Projektantrag Reviewversion erstellen	12 h	4	4	4																																							
MS 2	28.10.21										X																																
MS 2.1 Feinkonzept	Soll 31 h							2	2	2		2	3	4	4	4																											
MS 2.1 Feinkonzept	Ist 46 h							2	2	2		2	3	4	4	2																											
MS 2.2 Variantenwahl	Soll 4 h										4																																
MS 2.2 Variantenwahl	Ist 4 h										4																																
MS 2.2.1 Software	Soll 6 h										4																																
MS 2.2.1 Software	Ist 10 h										4																																
MS 2.2.2 Hardware	Soll 10 h										4	4	2																														
MS 2.2.2 Hardware	Ist 18 h										4	4	2																														
Summe	Soll 51 h																																										
Summe	Ist 78 h																																										
MS 3	04.11.21																																										
MS 3.1 Raspberry vorbereiten	Soll 6 h										4																																
MS 3.1 Raspberry vorbereiten	Ist 2 h																																										
MS 3.2 Software installieren	Soll 8 h											2	4	2																													
MS 3.2 Software installieren	Ist 4 h														4	4																											
Summe	Soll 14 h																																										
Summe	Ist 6 h																																										
MS 4	09.11.21																																										
MS 4.1 Schaltplan entwerfen	Soll 4 h																																										
MS 4.1 Schaltplan entwerfen	Ist 2 h																																										
MS 4.2 Lochraster bestücken	Soll 4 h																																										
MS 4.2 Lochraster bestücken	Ist 6 h																																										
MS 4.3 Lochraster testen	Soll 2 h																																										
MS 4.3 Lochraster testen	Ist 2 h																																										
Summe	Soll 10 h																																										
Summe	Ist 10 h																																										
MS 7																																											
MS 3.1 Dokumentation erstellen	Soll 8 h																																										
MS 3.1 Dokumentation erstellen	Ist 20 h																																										
MS 3.2 Presentation vorbereiten	Soll 0 h																																										
MS 3.2 Presentation vorbereiten	Ist 0 h																																										

Tabelle 2: Terminplan Teil 1

Monat			November															Dezember															Januar																																																											
KW	Datum	Termin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
ZbW Termine																																																																																												
Urlaub																																																																																												
MS 5																																																																																												
MS 5			23.11.21																																																																																									
MS 5.1	Sensordaten Auslesen	Soll	8 h																																																																																									
MS 5.1	Sensordaten Auslesen	Ist	2 h																																																																																									
MS 5.2	Sensordaten Sammeln	Soll	12 h																																																																																									
MS 5.2	Sensordaten Sammeln	Ist	0 h																																																																																									
MS 5.3	Verbindungsaufbau mit OpenFab	Soll	8 h																																																																																									
MS 5.3	Verbindungsaufbau mit OpenFab	Ist	0 h																																																																																									
MS 5.4	Daten in OpenFab ablegen	Soll	14 h																																																																																									
MS 5.4	Daten in OpenFab ablegen	Ist	0 h																																																																																									
Summe			Soll 42 h Ist 2 h																																																																																									
MS 6																																																																																												
MS 6			06.12.21																																																																																									
MS 5.1	PCB Entwerfen	Soll	14 h																																																																																									
MS 5.1	PCB Entwerfen	Ist	0 h																																																																																									
MS 5.2	PCB Bestellen	Soll	2 h																																																																																									
MS 5.2	PCB Bestellen	Ist	0 h																																																																																									
MS 5.3	Gehäuse Entwerfen	Soll	14 h																																																																																									
MS 5.3	Gehäuse Entwerfen	Ist	0 h																																																																																									
Summe			Soll 32 h Ist 0 h																																																																																									
MS 7																																																																																												
MS 7			05.01.22																																																																																									
MS 3.1	Dokumentation erstellen	Soll	24 h																																																																																									
MS 3.1	Dokumentation erstellen	Ist	28 h																																																																																									
MS 3.2	Presentation vorbereiten	Soll	16 h																																																																																									
MS 3.2	Presentation vorbereiten	Ist	0 h																																																																																									
Summe			Soll 48 h Ist 28 h																																																																																									
Summe			Soll 209 h Ist 136 h																																																																																									

Tabelle 3: Terminplan Teil 2

1.5.3 Lieferobjekte

Nr.	Ziel	Muss	Kann
1.	Elektrolayout	X	
2.	PCB-Layout	X	
3.	Sensordefinition	X	
4.	OpenHAB auf Raspberry Pi	X	
5.	Aufzeichnung von Messdaten	X	
6.	Veröffentlichung des Fortschrittes		X
7.	Veröffentlichung des Source Code und der Schemata		X
8.	PCB mit SMD Bauteilen bestellt		X
9.	Serienfähiger Prototyp		X
10.	3D gedrucktes Gehäuse		X
11.	Lochraster Prototyp	X	

Tabelle 4: Lieferobjekte

1.6 Präferenzmatrix für Projektkriterien

Um für die im weiteren Verlauf nötigen Entscheidungen eine Grundlage zu haben, wurde die Präferenzmatrix gewählt. In dieser wurden alle oben genannten Attribute eingetragen und gegeneinander gewichtet, als Ergebnis gibt es eine Priorisierung der einzelnen Kriterien, dieser werden sortiert aufgelistet, wobei bei Gleichstand im Rang das direkte Vergleichsergebnis aus der Matrix herangezogen wird.



Präferenzmatrix für das Projekt Sensorphalanx			
Gewicht	Rangfolge	Anzahl Nennungen	Kriterien
8.791	5	8	a Modulare Sensoren
8.791	5	8	b Günstig zu beschaffende Sensoren (Breakout Boards)
4.396	10	4	c W-Lan Anbindung
10.99	2	10	d Unabhängigkeit von "konventionellen" Anbietern
10.99	2	10	e Offene Software
5.495	9	5	f Autonome Energieversorgung
7.692	8	7	g kabelloses Design
12.09	1	11	h einfach nachzubauen
2.198	12	2	i Batteriezustand anzeigen (LED)
2.198	12	2	j Batteriezustand übermitteln
4.396	10	4	k unscheinbare Optik
10.99	2	10	l gut rückbaubar
8.791	5	8	m hohe Kompatibilität mit Smart Home Systemen
2.198	12	2	n PCB günstig in der Herstellung
104.4		91	

Tabelle 5: Präferenzmatrix Projektkriterien

Rang	Kriterium
1	einfach nachzubauen
2	Unabhängigkeit von "konventionellen" Anbietern
3	Offene Software
4	gut rückbaubar
5	Modulare Sensoren
6	hohe Kompatibilität mit Smart Home Systemen
7	Günstig zu beschaffende Sensoren (Breakout Boards)
8	kabelloses Design
9	Autonome Energieversorgung
10	gut rückbaubar
11	W-Lan Anbindung
12	Batteriezustand übermitteln
13	Batteriezustand anzeigen (LED)
14	PCB günstig in der Herstellung

Tabelle 6: Präferenzmatrix Rangliste

2 Anforderungsanalyse und Konzeption

2.1 Ziel und Nutzen Sensorphalanx

Wesentliches Ziel der Diplomarbeit ist die Erstellung eines funktionalen Prototyps einer Sensorphalanx. Diese Phalanx soll in der Lage sein verschiedene Raumdaten zu sammeln und an ein Smart Home Systeme zu übermitteln.

Dem Benutzer hat die Möglichkeit, über ein Webinterface die Aktuellere der Sensoreinheit einzusehen und wenn mehrere Phalanxen vorhanden sind, diese in einer übersichtlichen Darstellung direkt miteinander zu vergleichen. Weiter wird es möglich sein, eine Einsicht in die Langzeitaufzeichnung der Daten nehmen zu können und diese mittels Kurven zu Visualisieren.

2.2 Funktionale Anforderungen

Im Folgenden werden die funktionalen Anforderungen an das Projekt erläutert. Das heisst, alle Funktionen, die im Zusammenspiel von Sensorphalanx, Smart Home Systeme und Benutzer werden vorher festgelegt.

- Die Phalanx sammelt alle Raumdaten und sendet sie an das Smart Home Systeme
- Der Nutzen kann über ein Webinterface die Istwerte einsehen
- Der Nutzer hat über das Webinterface Zugriff auf die historischen Daten und kann sich diese grafisch anzeigen lassen.

2.2.1 Allgemeine Systemfunktionen

Die allgemeinen Systemfunktionen gliedern sich in:

- Datensammlung: Sammeln und Zusammenstellung für eine Übermittlung an das Smart Home Systeme
- Verarbeitungsfunktionen: Verarbeitung der Informationen im Smart Home Systeme
- Dialogfunktion: Der Nutzer erhält Informationen über ein Webinterface und gegebenenfalls Meldungen bei über- und unterschreiten von Grenzwerten.

In den nachfolgenden Tabellen werden nur die wichtigsten Funktionen aufgeführt und beschrieben. Dialogfunktionen sind nicht genauer erläutert.

Datensammlung	
Auslesen	Daten der Sensormodule werden zyklisch (10 sec) ausgelesen
Zusammenführen	Die Sensordaten werden am Ende eines Zyklus mit einem Zeitstempel und der ID (MAC-Adresse) der Sensorphanx versehen und in eine Telegramm verpackt
Senden	Daten werden an das Smart Home Systeme übermittelt

Tabelle 7: Funktionale Anforderungen: Datensammlung

Verarbeitungsfunktionen	
Ablegen	Die von der Sensorphanx erhaltenen Daten werden im Smart Home Systeme abgelegt
Auswerten	Die Daten werden auf etwaige Grenzüberschreitungen überprüft
Alarm-Visualisieren	Grenzwertüberschreitungen werden im Webinterface visualisiert
Visualisieren	Je nach ausgewählter Ansicht werden die entsprechenden Informationen im Webinterface angezeigt.

Tabelle 8: Funktionale Anforderungen: Verarbeitungsfunktionen

2.3 Nicht-Funktionale Anforderungen

In den nicht-funktionalen Anforderungen werden alle Eigenschaften festgelegt, die das Projekt später erfüllen soll.

2.3.1 Allgemeine Anforderungen (Rahmenbedingung)

Smart Home Systeme

- Die Basis für das Webinterface wird vom Smart Home Systeme zur Verfügung gestellt und unterliegt den Grenzen dieses Systems, soll aber dennoch benutzerfreundlich und intuitiv zu bedienen sein.

Sensorphalanx

- Nach dem einbinden in das System soll die Sensorphalanx autonom funktionieren und in abständen von 1 min die Raumdaten sammeln und an das System übermitteln.
- Es ist sicherzustellen, dass bei nicht erreichbar sein des Smart Home Systemes keine Daten verloren gehen, sondern diese lokal mit Zeitstempel versehen, gesichert werden, um dann bei wieder erreichen des Systems die gesammelten Daten übermittelt zu können.
- Die Sensorphalanx soll in der Lage sein mit seiner Energiereserve mindestens 4 Wochen ohne Nutzereingriff funktionieren zu können.
- Die Kommunikation zwischen Sensorphalanx und Smart Home Systeme soll verschlüsselt erfolgen.

2.4 Komponentenauswahl

Da es eine Vielzahl von möglichen Varianten im Bereich Software und Hardware gibt, werden im folgenden Kapitel die verschiedenen Lösungsmöglichkeiten aufgezeigt und der Entscheidungsweg näher beleuchtet.

2.4.1 Smart Home Systeme

Die Basis eines Smart Home Systeme stellt die Zentrale dar, die per Funk mit weiteren Komponenten interagiert. Die Zentrale erkennt anhand von angebunden Sensoren Umweltinformationen und reagiert durch das Steuern von Aktoren darauf. Wird die Smart Home Zentrale beispielsweise von einem Bewegungsmelder im Raum über eine Bewegung informiert, kann sie abhängig von anderen Daten wie Helligkeit und Uhrzeit bestimmte Beleuchtungssettings aktivieren. Die Logik des Systems befindet sich somit allein auf der Zentrale, deren Auswahl dementsprechend gut überlegt sein sollte.

Abhängig vom gewähltem Funkstandard des Smart Home Systems (z.B. WLAN oder Bluetooth) stehen unterschiedlich viele Geräte zur Auswahl. Während manche konventionellen Lösungen nur mit Geräten des eigenen Herstellers vernetzbar sind, bieten andere Smart Home Systeme auch die Möglichkeit kompatible Komponenten anderer Hersteller einzubinden. Einen marktübergreifenden Standard gibt es derzeit nicht.

Es gibt grundsätzlich zwei unterschiedliche Herangehensweisen an die Zentrale Smart Home Software: Open Source oder Close Source.

2.4.1.1 Open Source

Als Open Source bezeichnet man Programmcode der der Öffentlichkeit zugänglich gemacht wird, das heißt, jeder kann ihn einsehen sowie nach Belieben verändern, verteilen und weiterentwickeln.

Open Source-Software stützt sich auf Peer-Review und Community-Produktion. Die aus diesem Prozess entstandene Software ist in der Regel flexibler und langlebiger als proprietäre Produkte, da sie von einer Community entwickelt wird und nicht von einem einzelnen Unternehmen, das seine Geschäftsgrundlage sichern möchte. Weit verbreitete Open Source Projekte haben auch den Ruf sichere Software zu sein, da hier viele Augen auf den Code schauen und so Fehler und Sicherheitslücken schneller gefunden werden und gegebenenfalls direkt vom Entdecker gelöst werden können.

2.4.1.2 Close Source

Close Source oder auch proprietäre Software bezeichnet Software, die die Möglichkeit der Einsicht und Weiterentwicklung stark einschränkt oder verunmöglicht. Der Vorteil der Close Source Software ist das, sie auf klar definierter Hardware eingesetzt werden kann und so passgenau zugeschnitten wird. Weiter bietet diese Art der Software den Herstellenden Unternehmen bessere Möglichkeiten der Vermarktung.

Als Nachteil ist hier aber zu nennen das Sicherheitsupdates nicht immer am Endgerät ankommt da Nutzer nicht immer auf dem aktuellen Stand sind. Weiter ist darauf hinzuweisen das Sicherheitslücken nicht zwingend dem Anbieter gemeldet werden und so negativ genutzt werden können.

2.4.1.3 Entscheidungsbegründung

Aufgrund der Vorgaben aus der Präferenzmatrix die auf Rang Zwei „Unabhängigkeit von „konventionellen“ Anbietern“ und auf Rang Drei „Offene Software“ festlegt, fällt hier die Entscheidung auf eine Open Source Lösung.

2.4.1.4 Open Source Smart Home Systeme

Im Bereich der Open Source Smart Home Systeme gibt es eine schwer zu überblickender Menge an möglichen Softwarevarianten und da sie Community getrieben entwickelt werden hält jede Gruppe ihre Variante für die Beste, das macht eine Auswahl sehr schwierig. Da das Smart Home System in diesem Projekt nur eine untergeordnete Rolle einnimmt wurde das Auswahlverfahren auf lediglich 3 Kandidaten eingegrenzt. diese wurden anhand durch eine Googelsuche eingegrenzt, als Suchbegriff dienten die Begriffe: Open Source Smart Home Systeme, ausgewählt wurden die ersten drei Ergebnisse die direkt auf eine Projektseite verweisen.

Name	Projektseite
openHAB	https://www.openhab.org/
Home Assistant	https://www.home-assistant.io/
ioBroker	https://www.iobroker.net/

Tabelle 9: Google Ergebnisse: Open Source Smart Home Systeme

2.4.1.4.1 openHAB

Geschichte

Gemäss Wikipedia wurde das auf Java geschriebene Open Source Projekt openHAB im Jahr 2010 erstmalig veröffentlicht. (1) Die Aktuelle Version 3.1.0 ist derzeit zum freien download verfügbar. OpenHAB ist Betriebssystemen unabhängig und um zusätzliche Technologien/Protokolle erweiterbar. Die von der Eclipse Public License geschützte und als Open Source veröffentlichte Software bietet als Userinterfaces sowohl die Möglichkeit einer Weboberfläche als auch, Android- oder Apple-iOS Apps. (2)

Verbreitung

Zum Zeitpunkt des Verfassens dieses Textes waren im Forum von OpenHab 40.400 User registriert und hatten 718.000 Posts verfasst was auf eine aktive Community schliessen lässt.

Site Statistics

	Last 24 hours	Last 7 days	Last 30 days	All Time
Topics	16	144	504	56.3k
Posts	158	1.3k	5.5k	718k
Users	9	54	246	40.4k
Active Users	754	2.0k	3.3k	—
Likes	35	232	1.3k	125k

Abbildung 1: Screenshot Forum Statistik openHAB (31.10.2021 16:45)

Userinterface

Es gibt ein «Standardisierten» Userinterface auf Web Basis dieses bietet die Möglichkeit auch Anpassungen vom Nutzer vorzunehmen bis hin eigene Designs zu entwickeln oder Templates zu beziehen.

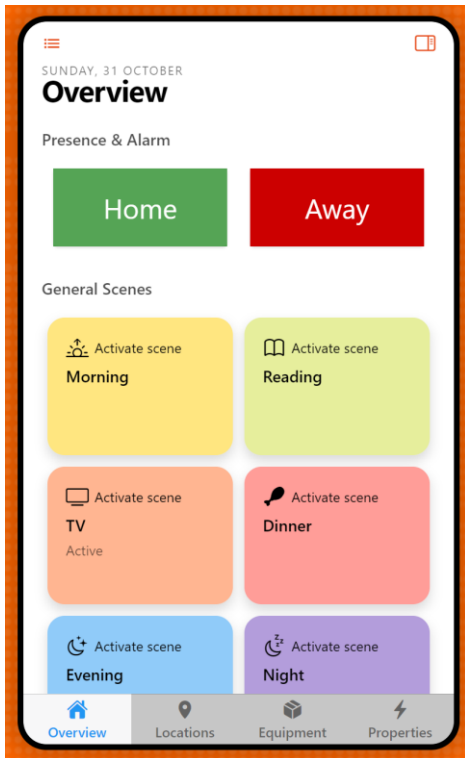


Abbildung 2: Screenshot Overview openHAB 3 Demo



Abbildung 4: Screenshot 3D übersichtsplan openHAB 3 Demo

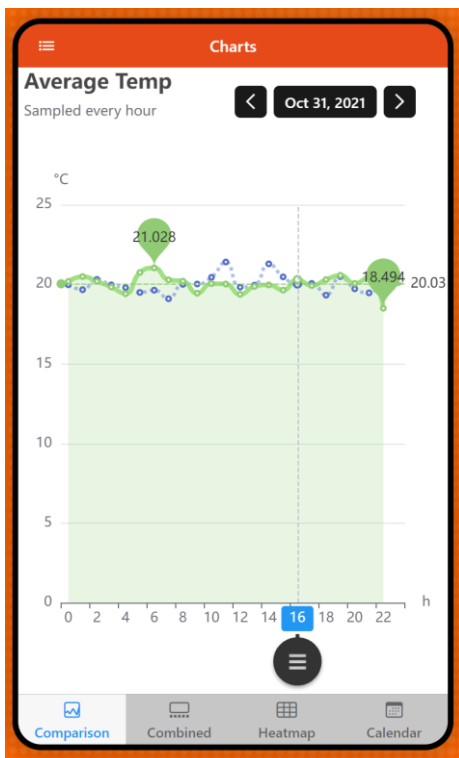


Abbildung 3: Screenshot Temperaturkurve openHAB 3 Demo

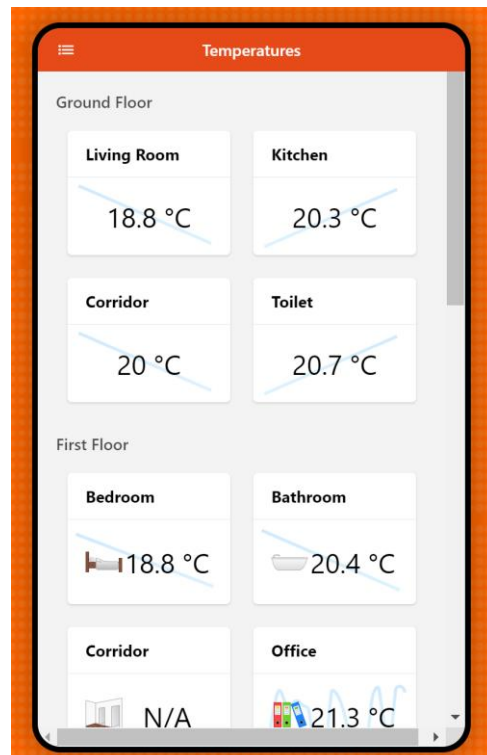


Abbildung 5: Screenshot Temperaturübersicht openHAB 3 Demo

2.4.1.5 Home Assistant

Geschichte

Home Assistant wird seit September 2013 entwickelt und im November 2013 erfolgte die Veröffentlichung der ersten Kernfunktion. Das Smart Home System basiert auf der Programmiersprache Phyten und legt Wert auf die lokale Steuerung und Privatsphäre der Nutzer. Das Projekt verfügt über Open-Source-Anwendungen für Android und iOS. (3)

Verbreitung

Mitte 2020 umfasste das Team der Entwickler mehr als 1930 Mitglieder die am Kern des Projektes arbeiten. Beim GitHub «State of the Octoverse» im Jahr 2019 wurde Home Assistant, Beiträgen von 63.000 Mitwirkenden als zehntgrösstes Open-Sorce-Projekt auf GitHub geführt.

Zum Zeitpunkt des Verfassens dieses Textes waren im Forum von Home Assistant 110.000 User registriert und hatten 1.600.000 Posts verfasst was auf eine aktive Community schliessen lässt.

Site Statistics				
	Last 24 hours	Last 7 days	Last 30 days	All Time
Topics	103	681	2.7k	130k
Posts	1.1k	7.8k	34.0k	1.6M
Users	124	721	3.0k	110k
Active Users	5.6k	15.3k	25.9k	—
Likes	236	1.6k	7.0k	321k

Abbildung 6: Screenshot Forum Statistik Home Assistant (31.10.2021 16:46)

Useriterface

Auch Home Assistant bietet die Möglichkeit einen Webbasierten Userinterface das modular anpassbar ist und ebenfalls die Möglichkeit bietet Templeise für verschiedene Designs einzubinden.



Abbildung 7: Screenshot Temperaturkurve ARS Home Design / Home Assistant Demo

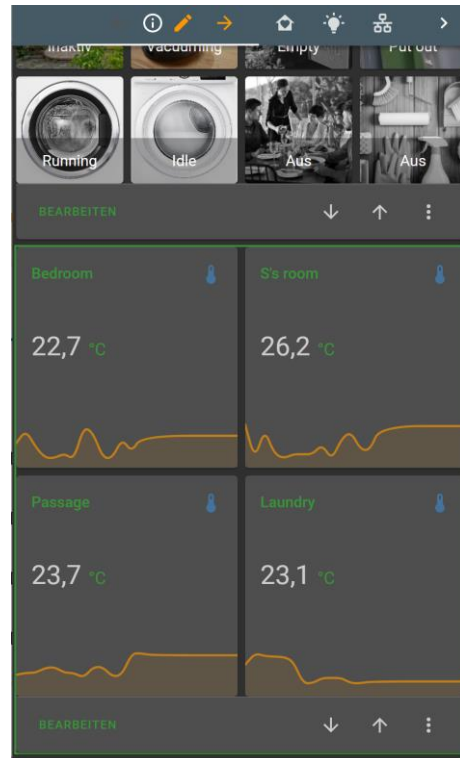


Abbildung 9: Screenshot Temperaturkurven Isa`s mobile friendly LL Design / Home Assistant Demo

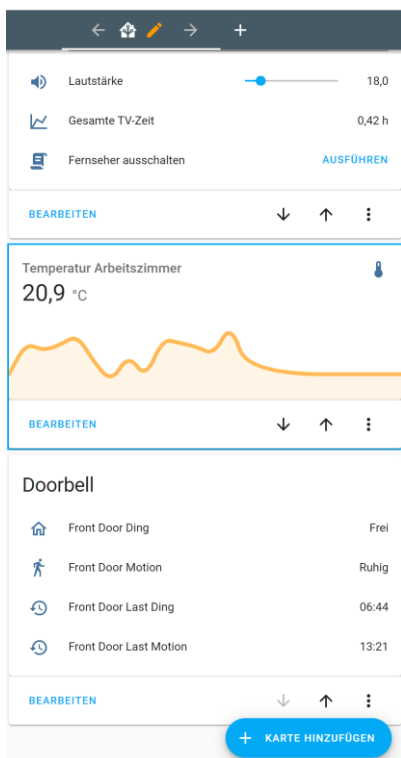


Abbildung 8: Screenshot Temperaturkurve ARS Home Design / Home Assistant Demo

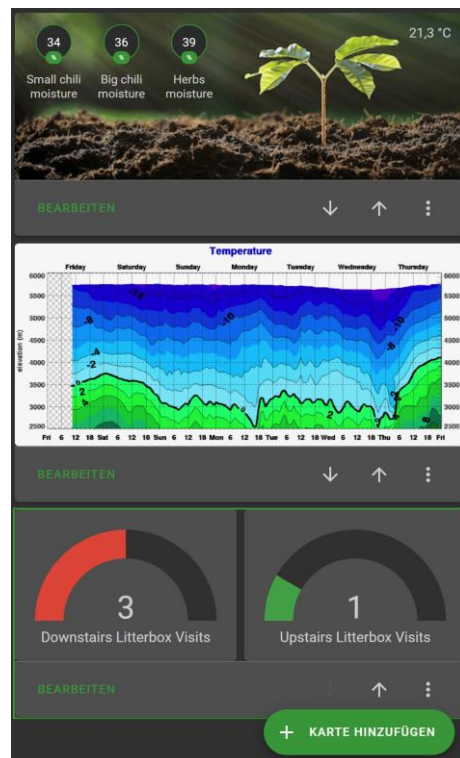


Abbildung 10: Screenshot Temperaturkurven Isa`s mobile friendly LL Design / Home Assistant Demo

2.4.1.5.1 ioBroker

Geschichte

IoBroker hat sich aus dem CCU.IO Projekt entwickelt, existiert aber seit 10. August 2014 als eigenständiges Projekt auf GitHub. Als Basisprogrammiersprache für dient JavaScript. (4) Das Projekt verfügt über keine Anwendungen für Android und iOS die direkt von ioBroker unterstützt werden lediglich «dritt Anbieter» Abs sind verfügbar.

Verbreitung

Zum Zeitpunkt des Verfassens dieses Textes waren im Forum von Home Assistant 42.700 User registriert und hatten 694.000 Posts verfasst was auf eine Engagierte Community schliessen lässt.

1.7k Online	42.7k Benutzer	48.6k Themen	694.0k Beiträge
----------------	-------------------	-----------------	--------------------

Abbildung 11: Screenshot Forum Statistik ioBroker (31.10.2021 16:47)

Userinterface

Das ioBroker Webinterface ist Modular aufgebaut und ermöglicht den User die freie Gestaltung der Darstellung, es besteht auch die Möglichkeit die Symbole mit anderen Kits auszutauschen und zu ergänzen.

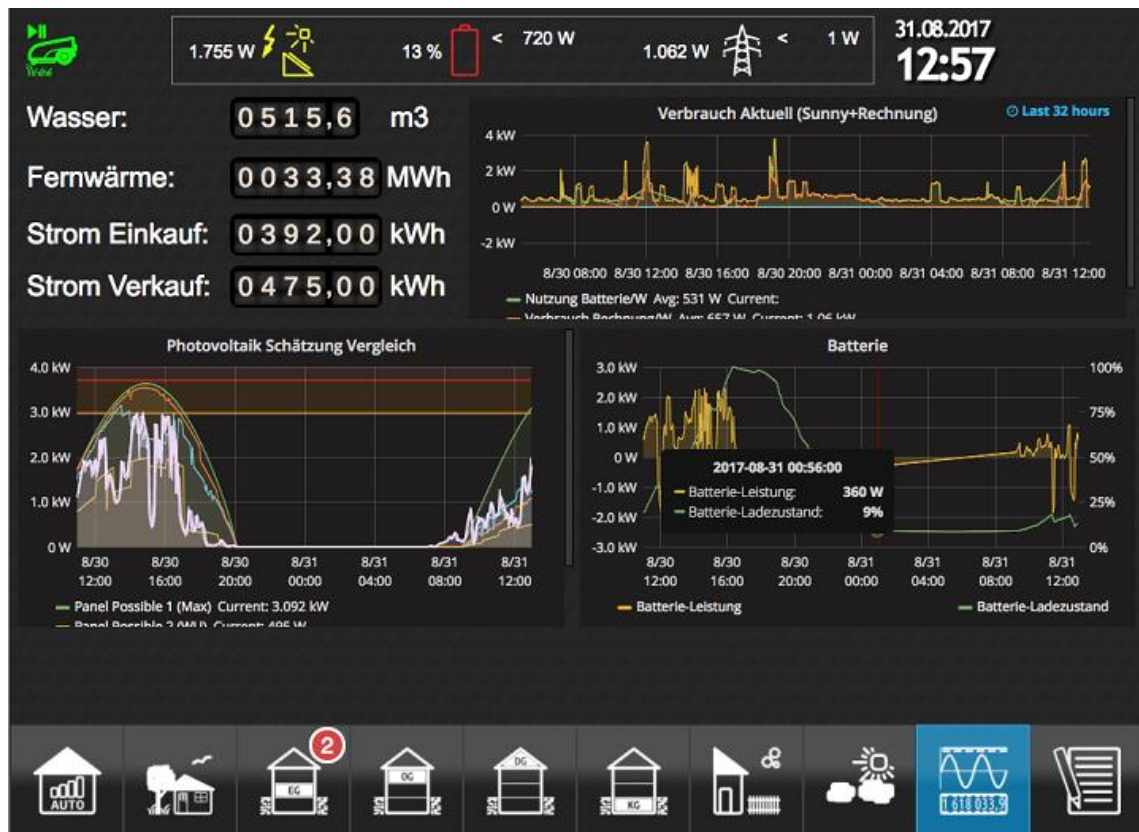


Abbildung 12: Screenshot Beispiel für eine selbst erstellte VIS-Benutzeroberfläche ioBroker (5)

2.4.1.5.2 Entscheidungsbegründung

Da im Grunde kein Smart Home System sich besonders hervorhebt und für das Projekt nicht relevant ist für welches System sich entschieden wird, da eine Adaption auf an anders Smart Home System im Anschluss an die Diplomarbeit jederzeit möglich ist, wurde für die Auswahl des Smart Home System ein besonderes Augenmerk auf die Aktivität der dahinter versammelten Community gelegt. Da innerhalb dieser Gruppe gegeben falls das grösste Interesse am Ergebnis dieser Diplomarbeit vorliegen dürfte und im Bedarfsfall auch der Pull an potenziellen Unterstützern bei Problemen mit dem System an grössten ist.

Eigenschaften	openHAB	Home Assistant	ioBroker
	Positiv	Positiv	Negativ
	Positiv	Positiv	Negativ
	Negativ	Negativ	Positiv
	Positiv	Positiv	Negativ

Tabelle 10: Entscheidungsübersicht Smart Home System

2.4.1.6 Analyse Siegervariante

Hier wird die Siegervariante nochmals tiefergehend analysiert. Der Schwerpunkt der Analyse liegt hierbei auf die notwendigen Eigenheiten die zur Kommunikation mit der Sensorphalanx nötig ist.

Kernarchitektur

Der Home Assistant Core besteht aus vier Hauptteilen.

- Event Bus:** ermöglicht es auf Ereignissen zu reagieren, also die Basisfunktionalität von Home Assistant.
- State Machine:** überwacht verschiedene Zustände und meldet eine Statusänderung, wenn sich ein Zustand ändert.
- Service Registry:** lauscht auf dem Ereignisbus und reagiert auf das Ereignis «call_service» und ermöglicht so anderen Code die Registrierung von Diensten.
- Timer:** sendet jede 1 Sekunde ein Ereignis auf den Ereignisbus.

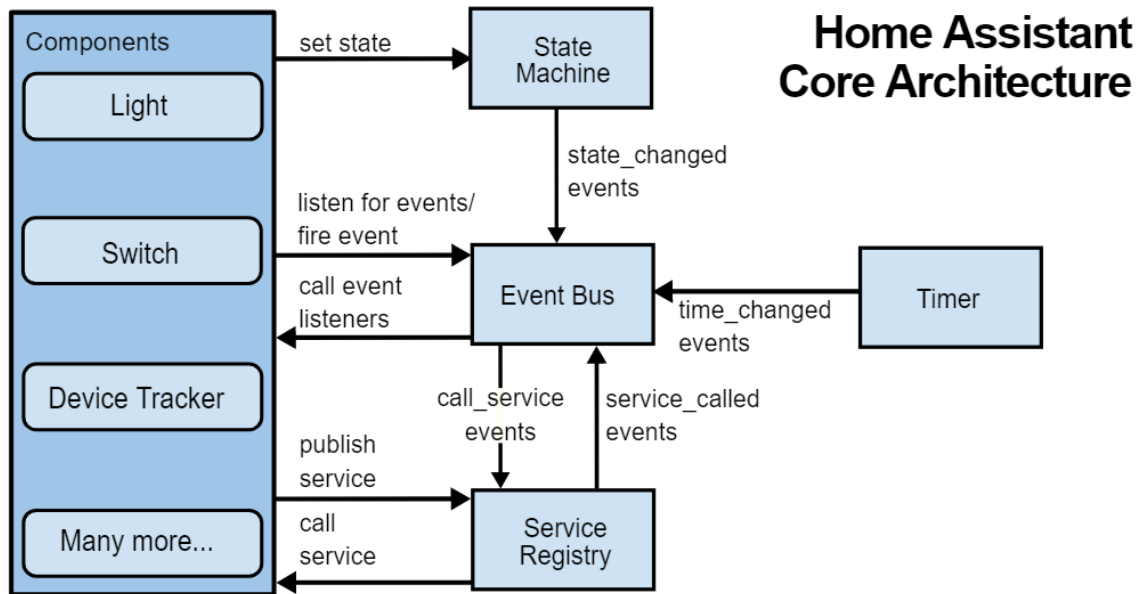


Abbildung 13: Überblick über die Kernarchitektur von Home Assistant (6)

2.4.2 Kommunikation / Protokoll

Der gesamte Datenaustausch mit der Sensoreinheit erfolgt über W-Lan, da dieses in der Regel in Wohnungen flächendeckend vorhanden ist. ~~Als Protokoll wird, auf Grund von Vorkenntnissen aus dem Studienfach Webtechnologie II, MQTT (Message Queuing Telemetry Transport) verwendet.~~

2.4.2.1 MQTT

Beim MQTT Protokoll handelt es sich um ein Offenes Netzwerkprotokoll das zur Maschine zu Maschinen Kommunikation eingesetzt wird und basiert auf einer Client Server («Broker») Struktur. Die Client Senden dem Server Nachrichten mit einem Topic, diese werden vom Server hierarchisch eingestuft und an Client die diese Topics Abonniert haben weitergeleitet.

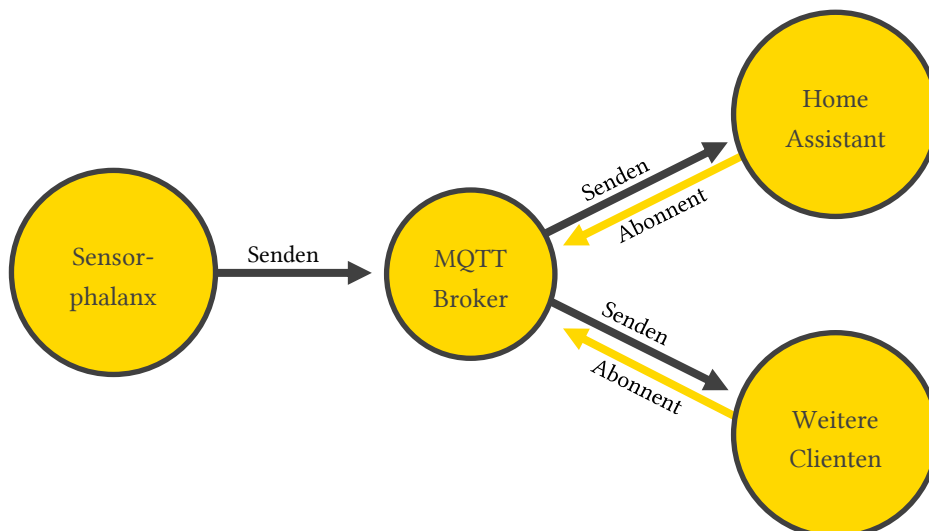


Abbildung 14: Funktionsweise MQTT Senden/Abonnieren-Architektur

Das Topic einer Nachricht kann sich wie folgt zusammensetzen: **Wohnung/Kinderzimmer/Temperatur**

Weiter bekommen die Nachrichten noch einen Quality of Service mit der wie folgt definiert ist

- at most once** die Nachricht wird einmal gesendet und kommt bei Verbindungsunterbrechung möglicherweise nicht an.
- at least once** die Nachricht wird so lange gesendet, bis der Empfang bestätigt wird, und kann beim Empfänger mehrfach ankommen.
- exactly once** hierbei wird sichergestellt, dass die Nachricht auch bei Verbindungsunterbrechung genau einmal ankommt.

MQTT bietet auch die Möglichkeit einen «Letzten Willen» zu definieren, diese Nachricht wird dann an alle Abonnenten verschickt, wenn die Verbindung zum Client unterbrochen wird. (7)

Um die Kommunikation zu steuern verwendet MQTT Kontrollpakete, in der folgenden Tabelle sind alle Kontrollpakete aufgelistet.

Name	Wert	Flussrichtung	Beschreibung
Reserviert	0	Client zu Server	Reserviert
CONNECT	1	Server zu Client	Verbindungsanfrage
CONNACK	2	Client zu Server oder Server zu Client	Verbindungsbestätigung
PUBLISH	3	Client zu Server oder Server zu Client	Publish Nachricht
PUBACK	4	Client zu Server oder Server zu Client	Publish Bestätigung (QoS 1)
PUBREC	5	Client zu Server oder Server zu Client	Publish empfangen (QoS 2 Zulieferung part 1)
PUBREL	6	Client zu Server oder Server zu Client	Publish veröffentlicht (QoS 2 Zulieferung part 2)
PUBCOMP	7	Client zu Server oder Server zu Client	Publish vollständig (QoS 2 Zulieferung part 3)
SUBSCRIBE	8	Client zu Server	Subscribe Anfrage
SUBACK	9	Server zu Client	Subscribe Bestätigung
UNSUBSCRIBE	10	Client zu Server	Unsubscribe Anfrage
UNSUBACK	11	Server zu Client	Unsubscribe Bestätigung

PINGREQ	12	Client zu Server	Ping Anfrage
PINGRESP	13	Server zu Client	Ping Antwort
DISCONNECT	14	Client zu Server oder Server zu Client	Disconnect Nachricht
AUTH	15	Client zu Server oder Server zu Client	Authentifizierungsaustausch

Tabelle 11: Kontrollpakettypen (7)

2.4.2.2 HTML

2.4.2.3 ????????

2.4.2.4 Entscheidungsbegründung

Eigenschaften	MQTT	HTML	???
	Positiv	Positiv	Negativ
	Positiv	Positiv	
	Negativ	Negativ	Positiv
	Positiv	Positiv	Negativ

Tabelle 12: Entscheidungsübersicht Protokolle

2.4.3 Kommunikation innerhalb der Sensorephalanx

2.4.3.1 Inter-Integrated Circuit (I2C)

2.4.3.2 Serial

2.4.4 Hardware Smart Home System

Home Assistant bietet viele Varianten der Installation an, neben den klassischen x86 und x64er varianten auf Linux und Windows Basis verfügt es auch über eine eigens entwickelte Hardware namens «Home Assistant Blue» die bereits eine vollständige Installation enthält. (8)

Für dieses Projekt wird eine Raspberry Pi verwendet, da diese eine gut verfügbare Hardware darstellt die sowohl in der Variante PI3 und PI4 ausreichend Leistung für die nötige Software zur Verfügung stellt.

2.4.4.1 Raspberry Pi

Ursprünglich wurde der Raspberry Pi vom Ingenieure Eben Upton entwickelt um Kindern und Jugendliche eine kostengünstige Möglichkeit zu bieten das Programmieren zu lernen. Mittlerweile erfreut sich der Raspberry Pi aber grosser Beliebtheit bei Bastlern

jeglichen alters und wurde so zum meist verkauften PCs Grossbritanniens (Mai2021 40Millionen Exemplare). Ein erster Prototyp wurde bereits 2006 veröffentlicht. In dem Darauf folgenden Boom in der Entwicklung von Smartphonekomponenten kamen aber bald leistungsstarke ARM Prozessoren auf dem Markt so das Anfang 2012 die erste Serienmodelle des Raspberry Pi für die breite Öffentlichkeit verfügbar waren. (9) Der scheckkartengrosse Minicomputer ist mittlerweile in der 4 Generation angekommen Zusätzlich zu den Modellen 1-4 in ihren Varianten A, A+, B und B+(nicht jede Variante ist für jedes Model verfügbar) gibt es noch zwei Zero varianten die sich hauptsächlich in ihrem Formfaktor und Anschlüssen von den normalen Modellen unterscheiden des Weiteren gibt es noch eine mehrere Computer Module die in ihrer Bauform anDDR2 Ram erinnern und keine üblichen I/O Anschlüsse aufweist diese lassen sich aber mit einem I/O Board bei Bedarf nachrüsten.

Für das Projekt wird ein Raspberry Pi 3 oder Raspberry Pi 4 benötigt.

2.4.4.1.1 Version 3 B Spezifikation

Komponente	Spezifikation
CPU	ARM Cortex-A53
Kern(e)	4
CPU Takt	1200 Mhz
RAM	1024 MB
LAN	10/100
W-LAN	b/g/n
Bluetooth	4.2
MicroSD-Slot	ja
Spannungsversorgung	5V DC / 2.5A

Tabelle 13: Auszug Hardwarespezifikation Raspberry Pi 3 B (10)

2.4.4.1.2 Version 3 B+ Spezifikation

Die Version 3B+ ist die letzte Version des Raspberry Pi 3B

Komponente	Spezifikation
CPU	ARM Cortex-A53
Kern(e)	4
CPU Takt	1400 Mhz
RAM	1024 MB

LAN	10/100/1000
W-LAN	b/g/n/ac
Bluetooth	4.2
MicroSD-Slot	ja
Spannungsversorgung	5V DC / 2.5A

Tabelle 14: Auszug Hardwarespezifikation Raspberry Pi 3 B+ (10)

2.4.4.1.3 Version 4 B Spezifikation

Die Raspberry PI 4 B Variante verfügt über vier ausführenden die sich lediglich in der Grösse ihres arbeitsspeichert unterscheiden.

Komponente	Spezifikation
CPU	ARM Cortex-A72
Kern(e)	4
CPU Takt	1500 Mhz
RAM	1024 / 2048 / 4096 / 8192 MB
LAN	10/100/1000
W-LAN	b/g/n/ac
Bluetooth	5.0
MicroSD-Slot	ja
Spannungsversorgung	USB Typ C 5V DC / 3A

Tabelle 15: Auszug Hardwarespezifikation Raspberry Pi 4 B (11)

2.4.4.1.4 Entscheidungsbegründung

Aus Gründen der grösseren Leistungsreserve für weitere Projekte und Erweiterungen am Smart Home System ist ein Raspberry Pi Model 4 einem Model 3 vorzuziehen. Allerdings sind auf Grund der Momentanen Chipkreise (12) diverse Modelle der Raspberry Pi schwer zu bekommen so dass bei der Bestellung darauf zu achten ist welches Model innert nützlicher Frist zu einem angemessenen Preis zu bekommen ist.

Eigenschaften	Raspberry Pi 3 B	Raspberry Pi 3 B+	Raspberry Pi 4 B
Prozessortackt	Negativ	Negativ	Positiv

Arbeitsspeicher	Negativ	Negativ	Positiv
LAN 1000	Negativ	Positiv	Positiv
Lieferbarkeit	Negativ	Positiv	Negativ

Tabelle 16: Entscheidungsübersicht Smart Home System Hardware

Bei der Bestellung wurde sich aus oben genannten Gründen für ein «Das Reichelt Raspberry PI 3 B+ All-In-Bundle» entschieden.

2.4.5 Hardware Sensorphanx

In den folgenden Kapiteln werden die verschiedenen Varianten der möglichen Hardware für die Sensorphanx vorgestellt und beschrieben. Die Auswahl erfolgt im Anschluss jeder unter Kategorie durch eine einfache Abwägung unter Berücksichtigung der im Kapitel 1.6 aufgelisteten Kriterien für diese Projekt.

2.4.5.1 Mikroprozessor

Der Mikroprozessor bildet die Zentrale Einheit der Sensorphanx seine Hauptaufgabe besteht darin die Daten der Sensoren zu sammeln und für die übermitteln an das Smart Home System zu übertragen. Weiter muss er die Möglichkeit einer Interrupt Funktion aufweisen da gegebenenfalls auch Informationen gesammelt werden müssen die unmittelbar übertragen werden müssen z.B. Bewegung.

2.4.5.1.1 Arduino

Am häufigsten verbreitet in der Maker*innen- und Bastler*innenszene sind wohl verschiedene Varianten von Arduino Boards als Basis für Projekte, diese aus Soft- und Hardware bestehende Physical-Computing-Plattform ist quelloffen und besteht aus einer I/O-Plattform mit einem AVR Mikroprozessor und einer Entwicklerumgebung. Diese I/O-Plattformen haben im Laufe der Zeit viele Entwicklungsschritte durchlaufen und, dass sie quelloffen sind, auch viele Nachahmer die aber kompatibel zur Entwicklungsumgebung sind, gefunden. In der Regel haben Arduino-Plattform keine W-LAN Anbindung diese können aber mittels Shields nachgerüstet werden. Bei diesen Shields handelt es sich oftmals um Platinen die mittels eines ESP8266 oder ESP32 Mikroprozessors eine W-LAN Schnittstelle erzeugen. Da es mittlerweile aber kompatibel I/O-Plattformen gibt die direkt auf einen ESP8266 oder ESP32 aufbauen erscheint es logisch diese auch direkt zu verwenden.

2.4.5.1.2 ESP8266

Der ESP2688 ist ein von der Chinesischen Firma espressif entwickelter 32-Bit-Mikrocontroller mit geringem Leistungsbedarf. Er ermöglicht auf Grund seiner integrierten

Wireless Local Area Network (WLAN nach IEEE 802.11 b/g/n) und seiner offenen Bauweise die Entwicklung von kompakten Aktor- und Sensormodulen.

In der lesenswerten Block-Artikel Serie Analysiert Thorsten von Eicken den Energieverbrauch in verschiedenen Betriebszuständen des ESPs 8266 (13).

ESP8266 Deep-Sleep with Periodic Wake-up Run-Time

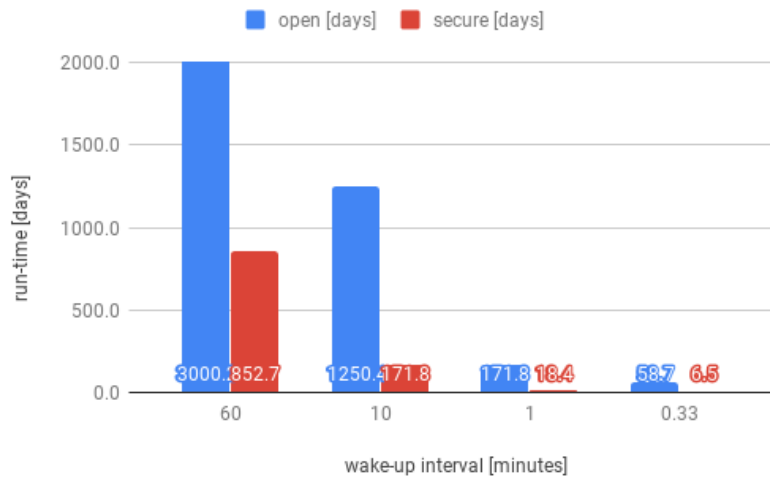


Abbildung 15: ESP8266 Deep-Sleep Laufzeit bei periodischem wecken mit einem 1000mAh Akku (Blau = offenes W-Lan Rot= Sicheres W-Lan) (13)

Technische Daten:

Prozessor:	Tensilica Xtensa mit 80 MHz
Memory:	Flash 4MB SRAM 160KB
WiFi / Bluetooth	IEEE 802.11 b/g/n bis max. 72,2 Mb/s
Ein/Ausgänge	16 GPIOs
PWM	8
Schnittstellen	SPI, I2C, I2S, UART
Stromverbrauch	80mA

Die Firma Ai-Thinker hat als Drittanbieter verschiedene Modelvarianten des ESP2866 auf dem Markt gebracht diese haben sich auf dem Markt auch entsprechend durchgesetzt.

2.4.5.1.3 ESP32

Der ESP32 ist der Nachfolger des ESP2688 und bietet neben einer leistungsfähigeren CPU zusätzlich die Möglichkeit einer Bluetooth Schnittstelle und deutlich mehr GPIO Pins die alle als Analogeingang oder -ausgang genutzt werden können.

In seiner Artikelserie hat Thorsten von Eicken ebenfalls den Energiebedarf eines ESP32 analysiert und eine Einschätzung der Zykluslebensdauer eines 1000mAh Akku beurteilt. (13)

ESP32 Deep-Sleep with Periodic Wake-up

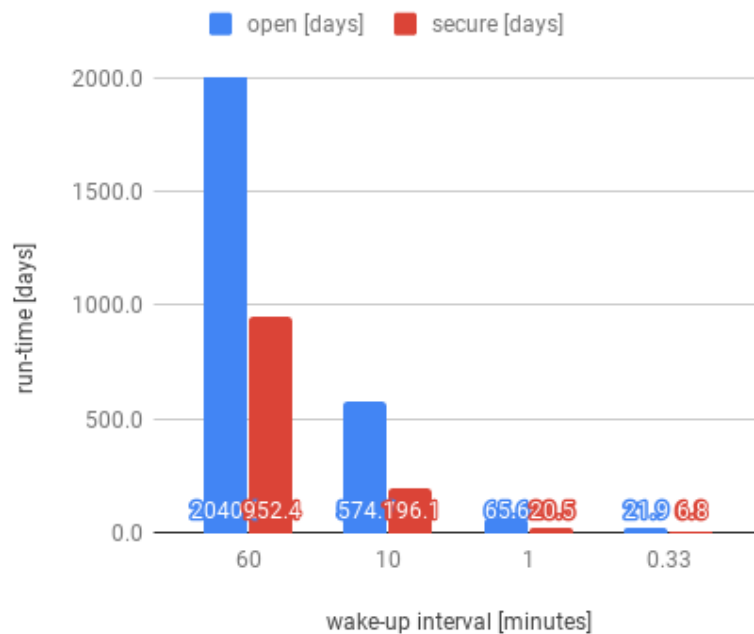


Abbildung 16: ESP32 Deep-Sleep Laufzeit bei periodischem wecken mit einem 1000mAh Akku (Blau = offenes W-Lan Rot= Sicheres W-Lan) (14)

Technische Daten:

Prozessor:	Xtensa dual-core 32bit LX6 mit 160 bis 240 MHz
Memory:	Flash 4MB SRAM 520KB
WiFi / Bluetooth	802.11 b/g/n – 2.4 GHz bis max. 150 Mb/s Bluetooth Low Energy
Ein/Ausgänge	48 GPIOs
PWM	bis 48 möglich
Schnittstellen	SPI, I2C, I2S, UART, CAN bus 2.0
Stromverbrauch	260mA

Besonders anzumerken ist das Thorsten von Eicken in seinem Artikel auf ein spezielles ESP32 Board () aufmerksam macht das eine aussergewöhnlich kleine Stromaufnahme (13uA-15uA) im Deep Sleep Modus hat. (14)

2.4.5.1.4 Entscheidungsbegründung

Für dieses Projekt ist es wichtig eine möglichst effiziente Ausnutzung der zur Verfügung stehenden Energie zu erreichen. Daher ist es sinnvoll den energieverbrauch der beiden ESP varianten miteinander zu vergleichen hier liegt der Fokus aber nicht auf den im Datenblatt angegebenen Strom bedarf, sondern auf dem Verbrauch im Deep-Sleep Modus, da zu erwarten ist das dieser am meisten aktiv ist bei einer zyklischen Abfrage der Sensordaten.

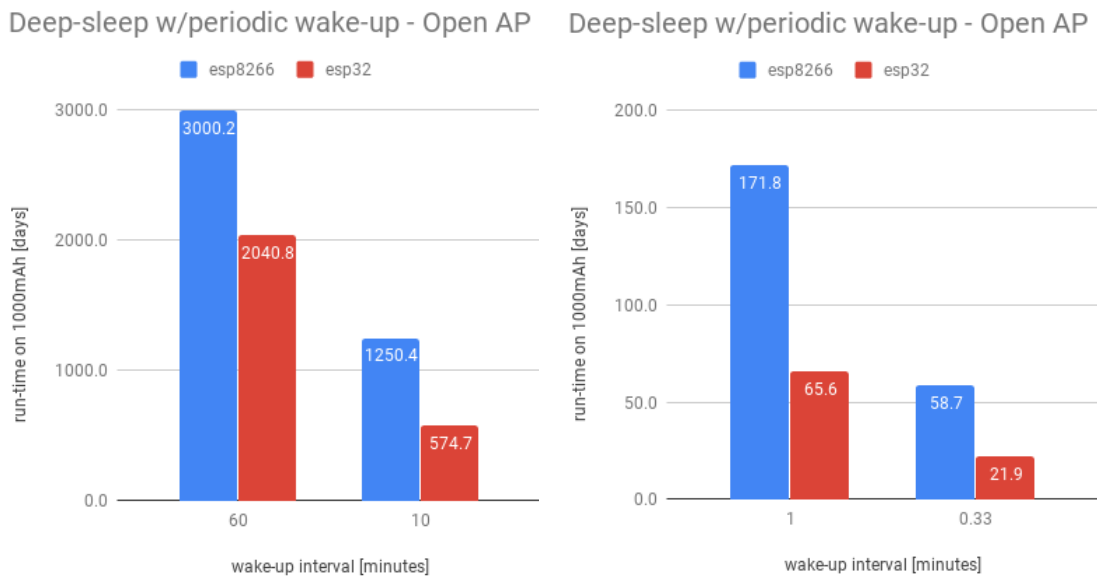


Abbildung 17: ESP8266 6 ESP32 im Vergleich Deep-Sleep Laufzeit bei periodischem wecken mit einem 1000mAh Akku (15)

Als Ergebnis dieses Vergleichs wird sich für eine ESP32 Board entschieden. Es wird versucht das «EzSBC ESP32-01 Breakout and Development Board» zu erhalten, da der Artikel aber ausschliesslich in den USA zu bekommen ist wird als alternative das «No-deMCU ESP32» Bord der Firma joy-it bestellt.

Eigenschaften	ESP8266	ESP32
Energieverbrauch im Deep Sleep	Negativ	Positiv
Anzahl GPIO-Pins	Negativ	Positiv
Bluetooth	Negativ	Positiv
Anzahl Schnittstellen	Negativ	Positiv

Tabelle 17: Entscheidungsübersicht Sensor CPU

2.4.5.2 Energieversorgung

Aus der Tabelle Präferenzmatrix Rangliste ist zu entnehmen das ein kabelloses Design sowie eine autonome Energieversorgung, wichtige Kriterien der Sensorphalanx sind, daher muss eine Möglichkeit gefunden werden die benötigte Energie bereitzustellen. Dazu gibt es verschiedene Möglichkeiten.

Zum einen wären da Handelsübliche diese sind aber auszuschliessen da sie nur einmalig genutzt werden können und somit Wegwerfprodukt sind.

Weiter gibt es noch die Möglichkeit Akkumodule zu verwenden, diese bieten die Möglichkeit, bei entsprechender Vorbereitung, direkt an der Phalanx geladen zu werden als auch Ausserhalb vom Gerät. Als Möglichkeiten kommen hier Lithium-Polymer-Akkumulator (LiPo) oder Lithium-Ionen-Akkumulator (Li-Ion) in Frage, diese werden wir im Nächten Kapitel genauer betrachten.

2.4.5.2.1 Lithium-Ionen-Akkumulator

Lithium-Ionen-Akkumulatoren ist ein Oberbegriff für Akkumulatoren auf Lithium Basis. Sie haben ein Hohes Energie pro Masse Verhältnis reagieren aber auf Tiefenentladung und Überladung nachteilig, dies kann bis zur Zerstörung führen, weshalb eine Schutzschaltung nötig ist. Handelsübliche Bauformen sind zylindrische Zellen die eine feste Zellenbezeichnung eine Typische Kapazität und eine definierte Abmessung haben.

Zellbezeichnung	Kapazität (in Ah)	Abmessungen (ø × l in mm)
14430	0,6–0,7	14 × 43
14500	0,7–0,8	14 × 53
14650	0,9–1,6	14 × 65
16340	0,6–1,0	16 × 34
16500	0,8–1,2	16 × 50
16650	2–3	16 × 65
17500	0,7–1,2	17,3 × 50
17650	1,2–2,5	17 × 65
18350	0,7–1,2	18 × 35
18500	1,1–2,2	18,3 × 49,8
18650	0,8–3,5	18,6 × 65,2

Tabelle 18: Zellenbezeichnung und Demissionierung Lithium-Ionen-Akku (16)

Ein Grosser Vorteil von Lithium-Ionen-Akkumulatoren ist das sie keinen Memory-Effekt unterlegen wie er bei Akkumulatoren auf NiCd oder NIMH Basis vorkommt. Zu erwähnen wäre noch das Lithium-Ionen Akkumulatoren einer Alterung unterliegen die stark von der Umgebungstemperatur abhängig ist, als Grund hierfür gelten parasitäre unumkehrbare chemische Reaktionen. (16)

Gefahren beim Umgang mit Lithium-Ionen-Akkus

Mechanische Belastungen wie in die Zelle eindringende Objekte könne innerhalb der Zelle zu Kurzschlüssen führen. Die Daraus resultierenden Hohen Ströme führen zur Erhitzung des Akkumulators und können zum entzünden des Gehäuses führen.

Chemische Reaktion kann bei einem geladenen Akku zu Überhitzung führen und so durch ein thermisches Durchgehen welches die im Akku gespeicherte Energie innert kürzester Zeit in Form von Wärme freisetzt. (16)

2.4.5.2.2 Lithium-Polymer-Akkumulator

Lithium-Polymer-Akkumulatoren sind spezielle Bauformen des Lithium-Ionen-Akkus, Die Besonderheit bildet hier die Konsistenz des Elektrolyts, dies ist beim Lithium-Polymer-Akku in einer eine Feste bis gallertartige Folie auf Polymerbasis. Dies ermöglicht eine freie Gestaltung die Zellen wie zum Beispiel als flache Zelle, dies hat aber den Nachteiligen Effekt das Bauformen sich ändern könne oder ganz vom Markt verschwinden was eine Nachbeschaffung erschweren könnte.

Gefahren beim Umgang mit Lithium-Polymer-Akkus

Ebenso wie Lithium-Ionen-Akkus sind Lithium-Polymer-Akkus empfindlich gegen Beschädigungen, Überladen, Tiefentladen, zu hohe Ströme sowie betreiben ausserhalb des Temperaturbereichs (0-60°C).

2.4.5.2.3 Entscheidungsbegründung

Für dieses Projekt werden zwei Lithium-Ionen-Akkus vom Typ 18650 verwendet, da diese eine einheitliche Grösse haben und Somit mittels Sockel am PCB befestigt werden können. Aufgrund der standardisierten Bauform ist auch eine einfache Beschaffung und möglich.

Eigenschaften	Lithium-Ionen-Akkus	Lithium-Polymer-Akkus
Feste definierte Bauform	Positiv	Negativ
Einfache Beschaffung	Positiv	Negativ

An Umgebung anpassbare Bauform	Negativ	Positiv
Sockelmontage ohne Stecker	Positiv	Negativ

Tabelle 19: Entscheidungsübersicht Lithium-Akku

2.4.5.3 Laden

Das Laden erfolgt, in der Prototypenvariante auf Lochraster, mittels einer fertigen PCB auf Basis des IP5306 Battery Management ICs. Dieser ermöglicht es den Akku sowohl zu laden als auch den Betrieb der CPU während des Ladevorgangs nicht zu unterbrechen und verfügt über alle nötigen Sicherheitsfunktionen um die Lithiumakkus zu schützen. Der IC bietet gemäss Datenblatt die Möglichkeit einer I2C Kommunikation was in der PCB variante ermöglicht diesen direkt auszulesen.

2.4.5.4 Sensoren

Bei der Auswahl ist zu berücksichtigen das, dass Projekt modular aufgebaut werden soll. Dadurch soll dem Endnutzer die Möglichkeit gegeben werden aus verschiedenen Modulen selber die für seinen Bedarf passenden Sensoren auswählen zu können.

Das bedeutet das für den Prototypen auch Sensortypen berücksichtigt werden dessen werte schon von anderen Sensoren abgedeckt wird. So kann sich der Nutzer entscheiden ob er beispielsweise einen Temperatur-/Feuchtefühler nutzen möchte oder auf einen Sensor der sowohl Temperatur und Luftfeuchtigkeit als auch Luftqualität erfassen kann.

Die hier vorgestellten Sensormodule umfassen ein breites Spektrum an möglichen Varianten, ist aber sicherlich nicht erschöpfend. Daher ist davon auszugehen das auch nach Abschluss der Diplomarbeit weitere Sensormodelle hinzukommen.

2.4.5.4.1 Bewegungserkennung

Zum Erkennen von Bewegungen gibt es aktive und passive Sensoren. Zu den Aktiven Möglichkeiten gehören elektromagnetische Wellen (HF, Microwellen oder Dopplerradar) oder mittels Ultraschalles. Als Passive und die am häufigsten genutzte Möglichkeit ist die Infrarotstrahlungserkennung zu nennen, diese detektiert kleine Änderungen der Temperatur in der Umgebung wenn sich Personen oder Tiere bewegen. (17)

Grundsätzlich gibt es meist in zwei unterschiedliche Baugrößen von Infrarotbewegungsmeldern die kleine Variante deckt einen Bereich einstellbar bis 3 Meter ab die grössere bis 7m. Aufgrund der Tatsache das die Sensorphalanx unter die Die Wohnungsdecke montiert werden soll ist es sinnvoll die Variante mit einem Erfassungsbereich von 7m zu nehmen da in der Regen die Deckenhöhe einer Wohnung bei min 2.3m liegt und so eine grosse Fläche innerhalb eines Raumes abgedeckt werden kann.

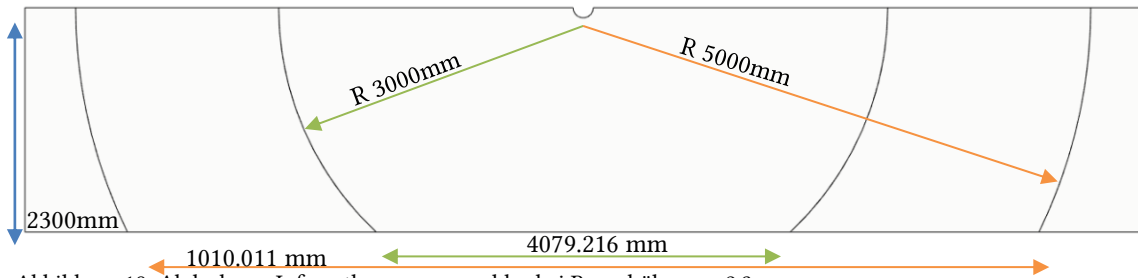


Abbildung 18: Abdeckung Infrarotbewegungsmelder bei Raumhöhe von 2.3m

Sensor	Technische Daten		Wertung
Sensor Typ: LHI778	Stromaufnahme	50uA	Positiv
	Erfassungsbereich	7m	Positiv
	Sensorlinse Ø	23mm	Negativ
Sensor Typ: AS312	Stromaufnahme	< 1mA	Negativ
	Erfassungsbereich	3m	Negativ
	Sensorlinse Ø	10.5mm	Positiv

Tabelle 20: Eigenschaften DHT11 und DHT22

2.4.5.4.2 Helligkeit

Als Sensor zur Erfassung der Helligkeit bietet sich ein LDR Sensor an diesem Sensor liegt preislich im Rappen Bereich, da es sich um einen einfachen lichtsensitiven Widerstand der günstig in grossen Mengen produziert werden kann. Der LDR Sensor benötigt lediglich ein Vorweiderstand und wird direkt an einem Analog-In der CPU angeschlossen.

Als zukünftige Varianten sind noch Module auf Basis der Bauteile BH1750 und TSL2591 beide Sensoren verwenden zur Kommunikation I2C. Für das Projekt wurde sich zunächst gegen die Varianten entschieden da der LDR alles bietet was benötigt wird und deutlich günstiger zu bekommen ist als die Alternativen.

2.4.5.4.3 Ladezustandserkennung und Anzeige

Für die Erfassung des Akkuladezustands wird im Lochrasterprototyp auf einen 10kOhm Widerstand zurückgegriffen dieser wird parallel zum Akku platziert, damit ist ein Auslesen von Spannungen bis 5 Volt möglich (Spannungen >5Volt zerstören aber den Prozessor).

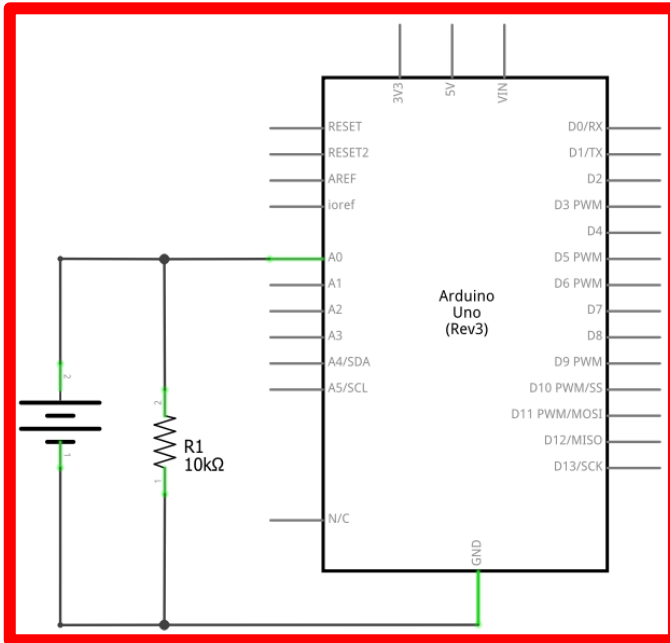


Abbildung 19: Schaltplan für die Spannungsmessung mit dem ESP32

In der Weiterentwickelten PCB Variante der Sensoreinheit kann der Ladezustand direkt vom IP5306 Battery Management IC mittels I2C ausgelesen werden.

Zur visuellen Darstellung des Ladezustands am Sensor wird eine RGB-LED, die in regelmäßigen Abständen für kurze Zeit aufblinkt, eingesetzt.

2.4.5.4.4 Luftqualität

Zur Luftqualitätsmessung steht unter anderem der Sensor vom Typ CCS811 zur Verfügung, dieser Sensor ist in der Lage die Anzahl an flüchtigen, organischen Verbindungen (VOC) messen und das Kohlendioxidäquivalent (eCO₂) zu berechnen. Zu beachten ist das der Sensor eine Burn-In Phase von 48 Stunden benötigt, sowie eine Aufwärmzeit von 20 Minuten nach längerem Aussetzen der Messungen empfohlen, wird bevor gültige Daten zu erwarten sind. Die Schnittstelle zur Kommunikation mit der CPU ist I2C über dieser werden je nach eingestelltem Modus eine Messung pro Sekunde, eine Messung alle 10 Sekunden eine Messung alle 60 Sekunden oder eine Messung alle 250ms gesendet.

Als Alternative wäre hier der Sensor vom Typ SGP30 zu nennen, dieser unterscheidet sich im Wesentlichen darin dass er keine Burn-In Phase oder Aufwärmzeit benötigt. Weiter hat er nur zwei umschaltbare Modi und zwar einen Arbeitsmodus und einen Sleepmodus. Ein zusätzlicher Vorteil ist dass der Sensor im Gegensatz zum CCS811 kein I2C-Takt- Stretching (**erklären**).

Sensor	Eigenschaften	Wertung
--------	---------------	---------

CCS811	Fester einlese Zyklus	Negativ
	I2C-Takt- Stretching	Negativ
	Burn-In Phase & Aufwärmzeit nötig	Negativ
SGP30	Einlesen bei Bedarf sonst Sleep	Positiv
	Kein I2C-Takt- Stretching	Positiv
	Direkte Messungen möglich	Positiv

Tabelle 21: Eigenschaften CCS811 und SGP30

2.4.5.4.5 Temperatur

Zur Messung der Temperatur wird eine ST1147 der Firma IDUINO verwendet, dieser basiert auf einen NTC-MF52 9350. Der NTC Widerstand weist einen Negativen Temperaturkoeffizienten auf der bei hohen Temperaturen den elektrischen Strom besser leitet als bei tiefen Temperaturen. Alternativ wären hier noch PTC Widerstände, genormte PT100 Temperaturfühler oder ein Modul auf Basis eines DS18B20, diese sind aber momentan nur schwer zu bekommen oder deutlich teurer als die oben genannte NTC Variante.

2.4.5.4.6 Kombinationssensor: Temperatur & Feuchtigkeit

Für den Abdeckung der Feuchtigkeit und Temperaturmessung sind die günstigen DHT11 und DHT22 Sensoren eine gut geeignet. Die DHT11 und DHT22 Sensoren haben allerdings einen signifikanten Nachteil, die Abtastrate beim DHT11 beträgt 1 Hz und beim DHT 22 0.5Hz das bedeutet das alle 1 bzw. 2 Sekunden neue Daten an die CPU geschickt werden, dies könnte sich negativ auf die Datenerfassung nach einem aufwachen der CPU aus dem Deep Sleep auswirken. **Dies verhalten wird in der Praxis zu überprüfen sein und gegebenenfalls mittels Software zu Lösen sein.** Die Kommunikation der DHT Sensoren erfolgt mittels eines «eigenen» Datenbus.

Sensor	Messwert	Arbeitsbereich	Genauigkeit
DHT11	Luftfeuchtigkeit	20 – 100 %	±5.0 %
	Temperatur	0 – +50 °C	±2.0 °C
DHT22	Luftfeuchtigkeit	0 – 100 %	2 – 5 %
	Temperatur	-40 – +80 °C	±0.5 °C

Tabelle 22: Eigenschaften DHT11 und DHT22

Im Zuge des Projektes fällt die Entscheidung auf dein DHT22 Sensor da dieser eine deutlich höhere Genauigkeit hat.

Eigenschaften	DHT11	DHT22
Genauigkeit	Negativ	Positiv
Preis	Negativ	Positiv
Abtastrate	Positiv	Negativ

Tabelle 23: Entscheidungsübersicht Feuchtigkeit/Temperatur Sensor

2.4.5.4.7 Kombinationssensor: Temperatur & Luftdruck

Der Bosch BME180 vereint die Messung von Luftdruck und Temperatur als Alternative gibt es noch den MPL115A2 Chip die beiden Sensoren unterscheiden sich in ihren Eigenschaften nur in dem Punkt ihrer Messtoleranz der MPL115A2 hat eine Ungenauigkeit von ± 1 kPa der BME180 ± 0.012 kPa zudem ist der MPL deutlich teurer als im Vergleich zum BME180. Beide Varianten verfügen über die Möglichkeit der Kommunikation mittels des I2C Protokolls das den Verdrahtungsaufwand und den Programmieraufwand deutlich minimiert.

Eigenschaften	Bosch BME180	MPL115A2
Genauigkeit	Positiv	Negativ
Preis	Positiv	Negativ
I2C Protokoll	Positiv	Positiv

Tabelle 24: Entscheidungsübersicht Druck/Temperatur Sensor

2.4.5.4.8 Kombinationssensor: Temperatur, Feuchtigkeit & Luftdruck

Für die Messung der Einheiten Temperatur, Feuchtigkeit und Luftdruck gibt es auch kombiniert Sensoren gibt es auch. Es konnten zwei Modelle gefunden werden BME280 und MS8607 Beide Module unterscheiden sich nur minimal. Zur Kommunikation wird in beiden Fällen I2C verwendet.

Die Auswahl fiel auf dem BME280 da dieser zum Zeitpunkt der Bestellung beim Hautlieferant auf Vorrat Vorhand war wohingegen der MS8607 nur in Speziellen Shops verfügbar ist.

2.4.5.4.9 Kombinationssensor: Temperatur, Feuchtigkeit, Luftdruck & Luftqualität

Es gibt auch Module die vier wichtigsten Raumdaten in einem Modul vereint. Die Module basieren auf einen Bosch BME680 und vereint die Möglichkeiten Temperatur, Feuchtigkeit, Luftqualität und Luftdruck zu messen und die Werte mittel I2C an die CPU zu übermitteln.

2.4.5.4.10 Varianten Übersicht

In der folgenden Tabelle ist eine Übersicht aller ausgewählten Sensortypen und ihren ermittelbaren Werten dargestellt.

Sensor \ Zweck	LHI778	LDR	10K Ω	SGP30	NTC	DHT22	BMP180	BMP280	BME680
Bewegungs-erkennung	X								
Helligkeit		X							
Ladestand			X						
Luftqualität				X					X
Temperatur					X	X	X	X	X
Luft-feuchtigkeit						X		X	X
Luftdruck							X	X	X

Tabelle 25: Auflistung der ausgewählten Sensoren / Erkennungswert

Die obere Tabelle zeigt auch auf das die Sensoren LHI778, LDR, und die Ladestandsanzeige in jeder möglichen Konfiguration verwendet werden kann wohingegen die Restlichen Sensoren untereinander ersetzen oder gegebenenfalls ergänzen dafür aber Sensorwerte mehrfach vorkommen können, diese doppelten Sensor daten können gegebenenfalls zur Plausibilitätskontrolle genutzt werden.

2.4.6 Zusammenfassung Hardwarebestellung

Die ausgewählte Hardware ist für die Diplomarbeit zweckmässig aber kann durch andere Module im Nachgang weiter ausgebaut werden, dadurch ist es möglich eine hohe Modularität herzustellen. Die ersten Teile sind bereits nach eine Woche eingetroffen so das mit der Umsetzung begonnen werden konnte.

3 Umsetzung

3.1 Raspberry

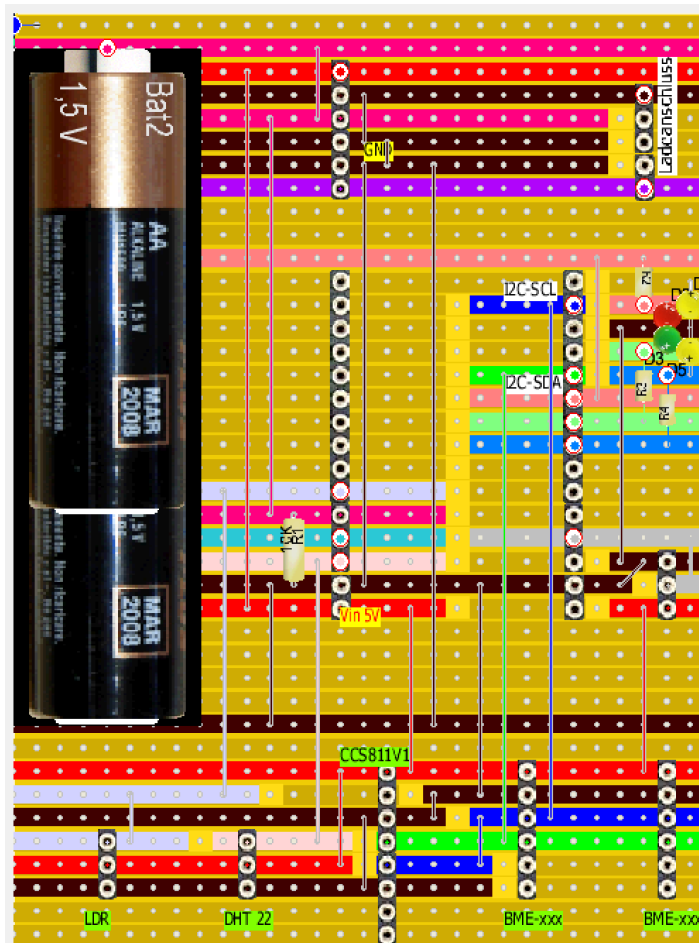
3.1.1 Installieren

3.1.2 Einrichten

3.1.3 Testen

3.2 Schaltplan entwerfen

3.3 Lochraster Prototyp



3.3.1 Aufbau

3.3.2 Testen

3.4 PCB Design

3.5 Gehäuse entwerfen

3.6 Software entwickeln

3.6.1 Kommunikation aufbauen

3.6.2 Daten Sammeln

3.6.3 Daten Schicken

3.7 Datenauswertung -aufbereiten auf dem Raspberry

3.7.1 Datenspeicherung

3.7.2 Visualisierung

4 Fazit

Literaturverzeichnis

1. **contributors, Wikipedia.** "OpenHAB.". *Wikipedia*. [Online] Wikipedia, 11. 09 2021.
[Zitat vom: 31. 10 2021.]
<https://en.wikipedia.org/w/index.php?title=OpenHAB&oldid=1043655906>.
2. **openHAB.** [Online] **openHAB Foundation e.V.** [Zitat vom: 31. 10 2021.]
<https://www.openhab.org>.
3. **Wikipedia-Autoren. Home Assistant.** [Online] **Wikipedia – Die freie Enzyklopädie**, 12. 10 2021. [Zitat vom: 01. 11 2021.]
https://de.wikipedia.org/w/index.php?title=Home_Assistant&oldid=216315742.
4. —. **IoBroker.** [Online] **Wikipedia – Die freie Enzyklopädie**, 18. 10 2021. [Zitat vom: 01. 11 2021.]
<https://de.wikipedia.org/w/index.php?title=IoBroker&oldid=216468214>.
5. **iobroker. iobroker.** [Online] [Zitat vom: 01. 11 2021.]
<https://www.iobroker.net/#de/documentation/README.md>.
6. **home-assistant, developers @. Kernarchitektur. developers.home-assistant.io.** [Online] 02. 11 2021. [Zitat vom: 23. 05 2021.] <https://developers.home-assistant.io/docs/architecture/core>.
7. **Zimmer, Jonas. MQTT 5 als Anwendungsprotokoll iim Internet der Dinge.** Hochschule Offenburg. Offenburg : Hochschule Offenburg, 2019.
8. **Assistant, Home. Install Home Assistant. home-assistant.io.** [Online] [Zitat vom: 03. 11 2021.] <https://www.home-assistant.io/installation/>.
9. **Cording, Stuart. elektormagazine. Was ist Raspberry Pi?** [Online] 26. 05 2021. [Zitat vom: 04. 11 2021.] <https://www.elektormagazine.de/news/was-ist-raspberry-pi>.
10. **Reichelt.de. RASPBERRY PI 3B+. Reichelt.de.** [Online] [Zitat vom: 04. 11 2021.] <https://www.reichelt.de/ch/de/raspberry-pi-3-b-4x-1-4-ghz-1-gb-ram-wlan-bt-raspberry-pi-3b--p217696.html?search=raspberry+pi+3+b%2B&&r=1>.
11. —. **RASP PI 4 B 4GB. reichelt.de.** [Online] [Zitat vom: 04. 11 2021.] https://www.reichelt.de/ch/de/raspberry-pi-4-b-4x-1-5-ghz-4-gb-ram-wlan-bt-rasp-pi-4-b-4gb-p259920.html?&trstct=pos_2&NBC=1.
12. **Silvio Werner. Chipkrise: Der Raspberry Pi bekommt eine saftige Preiserhöhung, Lieferbarkeit eingeschränkt. Notebookcheck.com.** [Online] 20. 10 2021. [Zitat vom: 04. 11 2021.]
<https://www.notebookcheck.com/Chipkrise-Der-Raspberry-Pi-bekommt-eine-saftige-Preiserhoehung-Lieferbarkeit-eingeschraenkt.574162.0.html>.
13. **Eicken, Thorsten von. Running Wifi Microcontrollers on Battery. TvE`s Blog.** [Online] 18. 11 2018. [Zitat vom: 07. 11 2021.]
<https://blog.voneicken.com/projects/low-power-wifi-intro/>.

14. —. **Low-Power ESP32 Boards.** *TvE`s Blog.* [Online] 25. 08 2019. [Zitat vom: 07. 11 2021.] <https://blog.voneicken.com/2018/lp-wifi-esp32-boards/>.
15. —. **ESP8266 vs. ESP32 on Battery Power.** *TvE`s Blog.* [Online] 10. 12 2018. [Zitat vom: 07. 11 2021.] <https://blog.voneicken.com/2018/lp-wifi-esp-comparison/>.
16. **Wikipedia-Autoren. Lithium-Ionen-Akkumulator.** [Online] Wikipedia – Die freie Enzyklopädie., 07. 11 2021. [Zitat vom: 09. 11 2021.] <https://de.wikipedia.org/w/index.php?title=Lithium-Ionen-Akkumulator&oldid=217053483>.
17. —. **Bewegungsmelder.** *Wikipedia – Die freie Enzyklopädie.* [Online] 05. 05 2021. [Zitat vom: 11. 11 2021.] <https://de.wikipedia.org/w/index.php?title=Bewegungsmelder&oldid=210585129>.
18. **GitHub ioBroker Insights.** [Online] Microsoft. [Zitat vom: 01. 11 2021.] <https://github.com/ioBroker/ioBroker/graphs/contributors>.
19. **Wikipedia-Autoren. ESP8266.** [Online] Wikipedia – Die freie Enzyklopädie, 09. 02 2021. [Zitat vom: 07. 11 2021.] <https://de.wikipedia.org/w/index.php?title=ESP8266&oldid=208631060>.
20. **Reichelt. DEBO BMP280 .** *Reichelt.de.* [Online] [Zitat vom: 10. 11 2021.] https://www.reichelt.com/ch/de/entwicklerboards-temperatur-und-drucksensor-bmp280-debo-bmp280-p266034.html?&trstct=pos_0&nbc=1.

Abbildungsverzeichnis

Abbildung 1: Screenshot Forum Statistik openHAB (31.10.2021 16:45)	15
Abbildung 2: Screenshot Overview openHAB 3 Demo	16
Abbildung 3: Screenshot Temperaturkurve openHAB 3 Demo	16
Abbildung 4: Screenshot 3D übersichtsplan openHAB 3 Demo	16
Abbildung 5: Screenshot Temperaturübersicht openHAB 3 Demo.....	16
Abbildung 6: Screenshot Forum Statistik Home Assistant (31.10.2021 16:46)	17
Abbildung 7: Screenshot Temperaturkurve ARS Home Design / Home Assistant Demo..	18
Abbildung 8: Screenshot Temperaturkurve ARS Home Design / Home Assistant Demo..	18
Abbildung 9: Screenshot Temperaturkurven Isa`s mobile friendly LL Design / Home Assistant Demo	18
Abbildung 10: Screenshot Temperaturkurven Isa`s mobile friendly LL Design / Home Assistant Demo	18
Abbildung 11: Screenshot Forum Statistik ioBroker (31.10.2021 16:47)	19
Abbildung 12: Screenshot Beispiel für eine selbst erstellte VIS-Benutzeroberfläche ioBroker (5).....	19
Abbildung 13: Überblick über die Kernarchitektur von Home Assistant (6)	21
Abbildung 14: Funktionsweise MQTT Senden/Abonnieren-Architektur	22
Abbildung 15: ESP8266 Deep-Sleep Laufzeit bei periodischem wecken mit einem 1000mAh Akku (Blau = offenes W-Lan Rot= Sicheres W-Lan) (13).....	27
Abbildung 16: ESP32 Deep-Sleep Laufzeit bei periodischem wecken mit einem 1000mAh Akku (Blau = offenes W-Lan Rot= Sicheres W-Lan) (14).....	28

Abbildung 17: ESP8266 6 ESP32 im Vergleich Deep-Sleep Laufzeit bei periodischem wecken mit einem 1000mAh Akku (15).....	29
Abbildung 18: Abdeckung Infrarotbewegungsmelder bei Raumhöhe von 2.3m.....	33
Abbildung 19: Schaltplan für die Spannungsmessung mit dem ESP32	34

Tabellenverzeichnis

Tabelle 1: Meilensteinplan.....	4
Tabelle 2: Terminplan Teil 1	5
Tabelle 3: Terminplan Teil 2	6
Tabelle 4: Lieferobjekte	7
Tabelle 5: Präferenzmatrix Projektkriterien.....	8
Tabelle 6: Präferenzmatrix Rangliste	9
Tabelle 7: Funktionale Anforderungen: Datensammlung.....	11
Tabelle 8: Funktionale Anforderungen: Verarbeitungsfunktionen	11
Tabelle 9: Google Ergebnisse: Open Source Smart Home Systeme.....	14
Tabelle 10: Entscheidungsübersicht Smart Home System.....	20
Tabelle 11: Kontrollpakettypen (7)	23
Tabelle 12: Entscheidungsübersicht Protokolle.....	23
Tabelle 13: Auszug Hardwarespezifikation Raspberry Pi 3 B (10).....	24
Tabelle 14: Auszug Hardwarespezifikation Raspberry Pi 3 B+ (10).....	25
Tabelle 15: Auszug Hardwarespezifikation Raspberry Pi 4 B (11).....	25
Tabelle 16: Entscheidungsübersicht Smart Home System Hardware	26
Tabelle 17: Entscheidungsübersicht Sensor CPU	30
Tabelle 18: Zellenbezeichnung und Demissionierung Lithium-Ionen-Akku (16).....	31
Tabelle 19: Entscheidungsübersicht Lithium-Akku	32
Tabelle 20: Eigenschaften DHT11 und DHT22.....	33
Tabelle 21: Eigenschaften CCS811 und SGP30	35
Tabelle 22: Eigenschaften DHT11 und DHT22.....	35
Tabelle 23: Entscheidungsübersicht Feuchtigkeit/Temperatur Sensor.....	36
Tabelle 24: Entscheidungsübersicht Druck/Temperatur Sensor	36

Anhänge

Eidesstattliche Erklärung