

Mathias Hazibar



INTELLIGENTE PRÄVENTIVLÖSUNGEN ZUR UNFALLVERHÜTUNG IN FUNKBASIERTEN SMART HOMES



KfV – DIPLOMARBEITSREIHE

Mathias Hazibar

INTELLIGENTE PRÄVENTIVLÖSUNGEN ZUR UNFALLVERHÜTUNG IN FUNKBASIERTEN SMART HOMES



KfV-Diplomarbeitenreihe

Vom KfV (Kuratorium für Verkehrssicherheit) geförderte Abschlussarbeit, ausgeführt zum Zwecke der Erlangung des akademischen Grades eines „Bachelor of Science in Engineering“ im Studiengang BSA

Betreuung

FH-Prof. Mag. Dipl.-Ing. Dr. Friedrich Praus
Fachhochschule Technikum Wien
Studiengangsleiter Informations- und Kommunikationssysteme
Studiengangsleiter Smart Homes und Assistive Technologien
Department Electronic Engineering

Ansprechpartner KfV

Dr. Robert Bauer

Mai 2019

Kurzfassung

Diese Bachelorarbeit befasst sich mit dem Thema „Intelligente Präventivlösungen zur Unfallverhütung in funkbasierten Smart Homes“. Die Arbeit kann in zwei Schwerpunkte gegliedert werden. Der erste Schwerpunkt beschäftigt sich mit einer umfassenden State of the Art Recherche. Diese soll dem/der Leser/in einen umfassenden Einblick, über Produkte und Lösungen im Bereich der smarten Unfallprävention geben. Des Weiteren befasst sich die Recherche damit, dem/der Leser/in einen Überblick über die gängigsten Smart Home Funkprotokolle zu geben.

Der zweite Schwerpunkt dieser Arbeit liegt in der Beschreibung von zwei verschiedenen Use Cases, welche in Zusammenarbeit mit dem Kuratorium für Verkehrssicherheit entworfen wurden. Beide Use Cases wurden praktisch im neu errichteten Smart Living Lab der FH Technikum Wien umgesetzt. Die Use Cases befassen sich mit funkbasierten Lösungen zur automatischen Herdabschaltung und einer Kindersicherung für Smart Home. Zur Umsetzung der Use Cases werden Geräte verwendet, welche über die gängigsten Smart Home Funkprotokolle, wie zum Beispiel Z-Wave oder EnOcean, kommunizieren.

Schlagwörter: Smart Home, Funkbasiert, Z-Wave, Unfallprävention, Fibaro, Homee, Herdabschaltung, Kuratorium für Verkehrssicherheit (KfV)

Abstract

This bachelor thesis deals with the topic "Intelligent Solutions to provide Accident in Radio-based Smart Homes". The work can be divided into two main areas. The first area focuses on a comprehensive state-of-the-art research. This should give the reader a comprehensive insight into products and solutions in the field of smart accident prevention. Furthermore, the state-of-the-art research is concerned with giving the reader a comprehensive overview of the most common smart home radio protocols. The second focus is on the description of two different use cases designed in collaboration with the Kuratorium für Verkehrssicherheit (KfV). Both use cases were practically implemented in the newly built Smart Living Lab at FH Technikum Wien. The use cases deal with radio-based automatic cooker shut-off solutions and a solution for child safety in smart homes. To implement the use cases, devices are used that communicate via the most common smart home radio protocols such as Z-Wave or EnOcean.

Keywords: Smart Home, Wireless, Z-Wave, Accident Prevention, Fibaro, Homee, Cooker Shutdown, Kuratorium für Verkehrssicherheit

Inhaltsverzeichnis

1	Unfallvermeidung im Smart Home	7
1.1	Unfälle im Eigenheim	7
1.2	Forschungsfrage	7
1.3	Projektumfeld	7
1.4	Beschreibung der Use Cases	8
1.4.1	Automatische Herdabschaltung	8
1.4.2	Fensterüberwachung	8
1.5	Struktur der Arbeit	9
2	Stand der Technik	9
2.1	Funktechnologien	9
2.1.1	Z-Wave	9
2.1.2	ZigBee	12
2.2	Hausnotrufdienste	15
2.3	Notlagenerkennung durch Bilderkennung	15
2.4	Airbagsysteme zur Sturzvermeidung	16
2.5	Smarte Böden	16
3	Anforderungen	17
3.1	Funktionelle Anforderungen	17
3.1.1	Push Up Nachrichten empfangen	17
3.1.2	Zugriff auf das Smart Home Gateway	17
3.1.3	Hinzufügen von Smart Home fähigen Geräten	17
3.1.4	Zeitliche Beschränkungen „Automatische Herdabschaltung“	17
3.1.5	Löschen von Geräten im System	18
3.2	Nicht funktionelle Anforderungen	18
3.2.1	Auswahl der Geräte	18
3.2.2	Layout des Smart Home Gateways	18
3.2.3	Skalierbarkeit und Flexibilität	18
3.2.4	Videodreh zur Vorstellung des Projekts	18
4	Planung	19
4.1	Hardware	19
4.1.1	Smart Home Gateway	19

4.1.2	Bedieneinheit	20
4.1.3	PIR Bewegungsmelder	20
4.2	Software	21
4.2.1	WebHooks	21
5	Umsetzung eines Testaufbaus im Smart Living Lab.....	22
5.1	Verwendete Hardware	22
5.1.1	Homee Gateway	22
5.1.2	Fibaro MotionSensor.....	23
5.1.3	Aeotec MultiSensor 6.....	24
5.2	Verwendete Software.....	25
5.2.1	Homee App & Webapp	25
5.2.2	Node-RED	25
5.3	Szenario Herdabschaltung	31
5.3.1	Allgemein	31
5.3.2	Umsetzung.....	32
5.4	Szenario Kindersicherung	44
5.4.1	Allgemein	44
5.4.2	Umsetzung.....	44
6	Evaluierung.....	51
7	Conclusio	51
8	Literaturverzeichnis.....	52
	Abbildungsverzeichnis.....	54
	Anhang A: Bedienungsanleitung	55

1 Unfallvermeidung im Smart Home

1.1 Unfälle im Eigenheim

Jedes Jahr ereignen sich allein in Österreich rund 300.000 Unfälle im Eigenheim (Stand 2017). Von diesen Unfällen sind alle Altersgruppen betroffen, wobei Kinder und Senioren besonders gefährdet sind. Es stellt sich somit die Frage wie die Zahl an Unfällen vermindert werden kann.

1.2 Forschungsfrage

Grundlegend beschäftigt sich die hier beschriebene Arbeit mit der Frage, wie und mit welchen Geräten eine präventive Unfallverhütung in privaten Haushalten sichergestellt werden kann. Des Weiteren stellt sich die Frage welche Szenarien im realen Leben Vorteile mit sich bringen und welche diese sind.

1.3 Projektumfeld

Das in dieser Bachelorarbeit beschriebene Projekt wurde in Zusammenarbeit mit dem KfV – Kuratorium für Verkehrssicherheit und der FH Technikum Wien umgesetzt. Das Kuratorium war, als Auftraggeber des Projekts, in der Phase der Use Case Definition und der allgemeinen Zieldefinition involviert. Die aus den gemeinsamen Gesprächen resultierenden Use Cases wurden zwischen den im Bachelorstudiengang Smart Home und Assistive Technologien Studierenden, Manuel Schweiger und Mathias Hazibar aufgeteilt. Manuel Schweiger erarbeitete in Folge den Use Case „Automatische Beleuchtung und Sprachsteuerung“ (Schweiger, 2019). Mathias Hazibar entschied sich den Use Case „Fensterüberwachung“ zu erarbeiten. Gemeinsam setzten beide Studenten den dritten und aufwändigsten Use Case „Automatische Herdabschaltung“ um.

Herr FH-Prof. Mag. DI Dr. Friedrich Praus, welcher die Position als Studiengangsleiter des Bachelor Studiengangs Smart Home und Assistive Technologien innehat, betreute die Studierenden in der Planungs- und Umsetzungsphase sowie beim Schreiben der Bachelorarbeit.

Im Zuge des Bachelorprojekts wird eine praktische Umsetzung der Use Cases im neuen Smart Living Lab der FH Technikum Wien durchgeführt. Diese Installation dient der Demonstration der Use Cases für das KfV und für Führungen, welche externen Personen die Umsetzung von Smart Homes näherbringen soll.

1.4 Beschreibung der Use Cases

1.4.1 Automatische Herdabschaltung

Szenario

Ein Elternteil kocht das Mittagessen für seine Kinder. Es ist gestresst, da die Kinder in Kürze von der Schule abgeholt werden müssen. In Eile vergisst der Elternteil den Herd abzuschalten und verlässt das Haus. Das Essen wird inzwischen auf der weiterhin eingeschalteten Herdplatte stehen gelassen. Da das Haus schon etwas älter ist, gibt es in der Küche keinen Brandmelder oder sonstige Schutzgeräte, die einen Brand feststellen und einen Alarm auslösen würden. Nach einiger Zeit fängt das sich im Topf befindliche Gericht Feuer, welches sich schnell ausbreitet und einen großen materiellen bzw. gesundheitlichen Schaden verursacht.

Ziel:

Das definierte Ziel dieses Use Cases ist es die Bewohner des Hauses vor einem durch den Herd ausgelösten Brand zu schützen. Es soll verhindert werden, dass der Herd ohne die Anwesenheit einer Person in der Küche eingeschaltet bleibt. Aufgabe ist es eine Lösung zu entwickeln, welche erkennt, ob sich eine Person in der Nähe des Herdes aufhält. Ist dies nicht der Fall, soll der Herd nach einer zuvor festgelegten Zeit automatisch ausgeschaltet werden.

1.4.2 Fensterüberwachung

Szenario:

Ein Kleinkind sitzt mit seinen Eltern auf der Couch und spielt ein Spiel. Nach einiger Zeit verlässt es den Raum und verschwindet in seinem Zimmer. Im Zimmer steht das Fenster offen, welches von seinen Eltern zum Durchlüften geöffnet wurde. Getrieben von seinem Instinktiven Entdeckungsdrang, beschließt das Kind auf die Fensterbank zu klettern, um nach draußen sehen zu können. Am Fensterbrett angekommen, stellt sich das Kind gerade auf, als es plötzlich das Gleichgewicht verliert und aus dem Fenster stürzt.

Ziel:

Dieser Use Case beschäftigt sich mit dem Schutz von Kleinkindern, in einem Raum mit geöffneten Fenstern oder Balkontüren. Es soll eine Lösung entwickelt werden, um mittels smarterer Geräte festzustellen ob ein Fenster oder eine Balkontüre im Raum geöffnet ist. Des Weiteren soll festgestellt werden, ob sich ein Kleinkind, ohne Beisein eines Erwachsenen, in diesem Raum aufhält. Treten diese beiden Fälle auf, soll das System in der Lage sein, die Eltern oder die Aufsichtsperson zu alarmieren. Durch die Alarmierung sollen die Erwachsenen Personen, auf das sich in vermeintlicher Gefahr befindliche Kind aufmerksam gemacht werden. So soll verhindert werden, dass sich ein unbeaufsichtigtes Kleinkind einem geöffneten Fenster nähert oder durch die geöffnete Balkontüre ins Freie gelangt.

1.5 Struktur der Arbeit

In *Kapitel 2*, wird eine umfangreiche Stand der Technik Recherche durchgeführt. Diese zeigt dem/der LeserIn auf, welche Produkte bereits auf dem Markt erhältlich sind, beziehungsweise an welchen Techniken derzeit geforscht wird. Des Weiteren befasst sich die Recherche mit den bekanntesten Funkprotokollen, welche derzeit in funkbasierten Smart Home Systemen eingesetzt werden. In *Kapitel 3* werden die funktionellen und nicht funktionellen Anforderungen definiert, welche beim Aufbau des Systems beachtet werden mussten. *Kapitel 4* beschäftigt sich mit der Planung des Projektaufbaus. Es wird aufgezeigt, welche generelle Hard- und Software verwendet wurde um die, definierten Use Cases umzusetzen. Die einzelnen Hard und Softwarekomponenten werden allgemein beschrieben, um dem/der LeserIn eine genauere Vorstellung über die Funktionsweise der Geräte bzw. der Software zu vermitteln. Kapitel 5 beschäftigt sich mit der Umsetzung eines Test Aufbaus im Smart Living Lab der Fachhochschule Technikum Wien. Es wird aufgezeigt, welche Hard- und Softwarekomponenten für jeden einzelnen Use Case verwendet wurden. Danach werden die einzelnen Schritte beschrieben, die für die Umsetzung des Projekts nötig sind. *Kapitel 6* beschäftigt sich mit der Evaluierung des Systems. Im letzten Kapitel, *Kapitel 7*, wird eine Zusammenfassung der Resultate sowie ein Ausblick in die Zukunft gegeben.

2 Stand der Technik

Dieser Teil der Bachelorarbeit beschäftigt sich mit dem Aufzeigen von Technologien, die einerseits schon frei auf dem Markt verfügbar sind oder kurz vor der Serienreife stehen.

2.1 Funktechnologien

2.1.1 Z-Wave

Allgemein

Z-Wave ist ein von Zensys, erstmals als proprietär entwickelter Funkstandard, welcher sich über mehrere Jahre zunehmend zu einer herstellerübergreifenden Unternehmenslösung entwickelte. In Folge dessen beschlossen einige Firmen, die diese Funktechnologie für ihre Geräte nutzen die sogenannte Z-Wave Allianz zu gründen. Diese besteht seit 2005. (Yassein, 2016)

Eigenschaften

Z-Wave ist eines der meist verbreiteten Funkprotokolle im Bereich der Gebäudeautomatisierung. Die Geräte, die diesen Funkstandard verwenden bilden dabei ein sogenanntes Mesh Network. Damit ein solches Netzwerk funktionieren kann, muss jeder Teilnehmer eines Netzwerkes einen eindeutigen Identifikator, wie zum Beispiel eine ID,

besitzen. Das Z-Wave Protokoll legt dabei zwei verschiedene Identifikatoren fest. Dies ist zum einen die sogenannte Home ID und zum anderen die Node ID. (Vesternet Ltd, 2012)

Die Home ID ist eine gemeinsame Identifikationsnummer aller sich in Netz befindenden Nodes. Sie grenzt das Netzwerk von anderen Netzwerken ab. Nur Geräte mit der gleichen Home ID können miteinander kommunizieren. (Yassein, 2016)

Die Node ID ist eine im Netzwerk einzigartige Identifikationsnummer, welche jedes Gerät (Node) in einem Z-Wave Netzwerk besitzt und die dieses eindeutig kennzeichnet. (Yassein, 2016)

Controller und Slaves

Jedes Z-Wave Netzwerk besteht aus mindestens einem Controller und einem Slave. Der Controller wird auch als Primärkontroller bezeichnet. Seine Home ID ist im Regelfall 1. Seine Aufgabe ist es das Netzwerk aufzubauen und zu verwalten. Er speichert alle Daten, wie Home ID und Node IDs, über das Netzwerk. Weiters dient er den Slaves als einzige Zugangsstelle zum Netzwerk. Dies bedeutet, dass der Primärkontroller der einzige Punkt ist, den ein Slave nutzen kann, um sich mit dem Netzwerk zu verbinden. Im Bereich von Smart Home ist der Controller meist ein Z-Wave fähiges Gateway, welches zusätzlich mit dem Internet verbunden ist. (König, 2018)

Das Verhältnis von Controllern zu Slaves beträgt mindestens 1:1. Slaves sind allerlei Geräte wie Bewegungsmelder, Schalter und Dimmer. Im Gegensatz zum Controller verfügen diese über keine Routing Tabellen. (OpenZWave Library, 2019)

Meshing und Routing

Mit Z-Wave ist ein Controller in einem Netzwerk in der Lage mit einem Slave zu kommunizieren, selbst wenn sich dieser im eigentlichen Sinne außerhalb seiner Reichweite befindet. Dies ist der Eigenschaft des Mesh Networkings zu verdanken. Durch das aufgebaute Mesh Netzwerk und die vergebenen Node IDs, ist es dem Controller möglich, seine Nachricht von mehreren Nodes in Netzwerk weiterleiten zu lassen. Hierzu benötigt er die Node IDs der im Netz befindlichen Nodes, welche beim Einbinden eines Slaves sofort an diesen vergeben wurde.

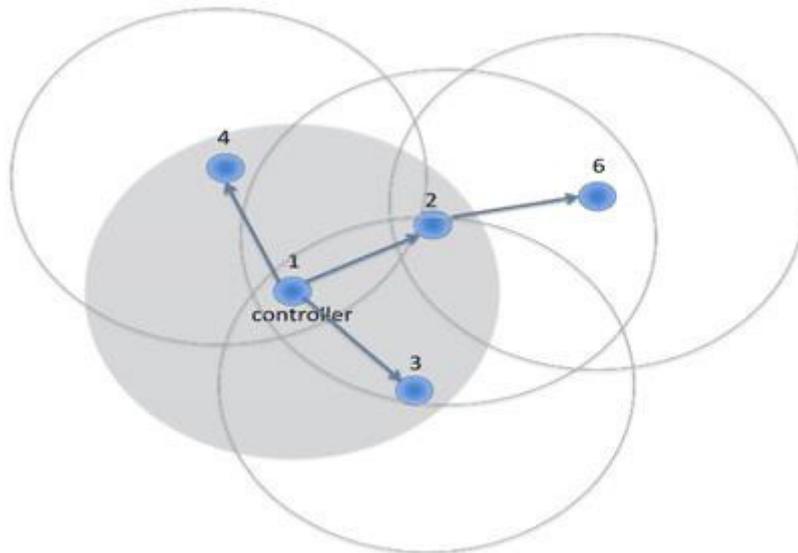


Abbildung 1: Z-Wave Netzwerk mit Controller (Vesternet Ltd, 2012)

In Abbildung 1 kann man erkennen das der Controller mit den Nodes 2,3 und 4 direkt kommunizieren kann. Node 6 ist für ihn jedoch nicht direkt zu erreichen. Um diesen Node zu erreichen, kommt die beschriebene Netzwerktopologie des Mesh Networking zum Einsatz. Da ein Z-Wave Netzwerk ein Mesh Network ist, kann der Controller nun eine Node in seiner Reichweite verwenden, um die Nachricht über sie an den Node welcher außer Reichweite ist zu senden. Im Fall, der in Abbildung 1 gezeigt wird, würde der Controller im einfachsten Fall seine Nachricht an Node 2 weiterleiten. Node 2 würde die Nachricht anschließend an Node 6 weitersenden, da diese in seiner Reichweite liegt. Das Prinzip des Weiterreichens einer Nachricht ist durch das Funkprotokoll auf maximal vier Hops beschränkt.

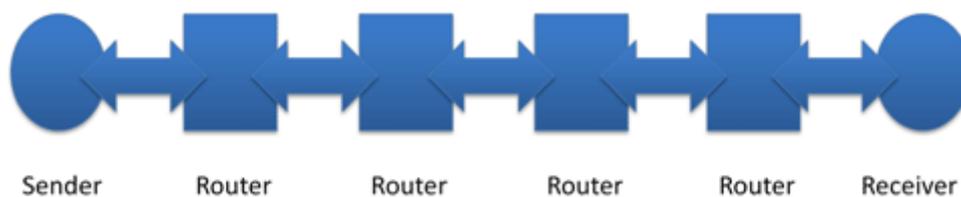


Abbildung 2: Maximaler Weg eines Pakets in einem Z-Wave Netzwerk (Vesternet Ltd, 2012)

Technische Spezifikationen

- Maximal 232 Geräte pro Netzwerk
- Verschlüsselte Kommunikation
- Netz- und batteriebetriebene Geräte verfügbar
- 868 MHz-Frequenzband
- Geräte durch Allianz zertifiziert
- Reichweite: Freiland bis zu 200m, Gebäude bis zu 30m

(Janssen, 2016)

2.1.2 ZigBee

Allgemein

ZigBee ist ein Standard für funkbasierte Sensor Netzwerke. Die Sensoren haben folgende Eigenschaften:

- Wenig Energieverbrauch
- Geringe Datenrate
- Kostengünstig
- Zeitgemäße Security

Der Standard wurde 2002 von der ZigBee Alliance veröffentlicht. Seitdem hat er sich im Bereich des Internet of Things (IoT) weit verbreitet. Den größten Anteil an ZigBee Geräten machen im Smart Home Bereich Lampen und generell smarte Leuchtmittel aus.

Netzwerktopologie

Ein ZigBee Netzwerk kann verschiedene Netzwerktopologien nutzen. So ist es möglich ein Netzwerk in Form einer Stern-, Baum-, oder Mesh Topologie aufzubauen.

In Netzwerken mit einer **Sterntopologie**, gibt es einen Koordinator und viele weitere Endgeräte. Der Koordinator steht in dieser Topologie, bildlich gesehen, in der Mitte des Sterns. Er kennt alle weiteren Geräte in seinem Netzwerk und steht mit ihnen in direkter Verbindung. (Ramaya, 2011)

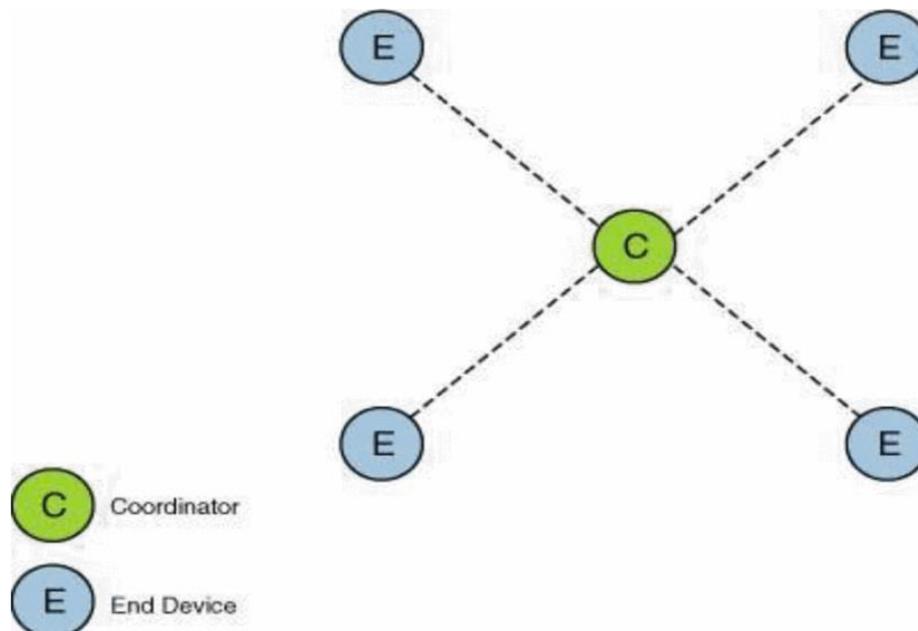


Abbildung 3: Vektor Grafik Sterntopologie, (Kumar, 2016)

Möchte ein Endgerät (Abbildung 3), eine Nachricht an ein anderes Endgerät schicken, so teilt es dies dem Koordinator mit. In Folge dessen, schickt der Koordinator, der alle Teilnehmer des Netzwerks kennt, das Paket an das gewünschte Endgerät weiter. Der größte Nachteil

dieser Topologie ist jedoch, dass der Koordinator einen sogenannten Bottleneck darstellt. Fällt er aus, ist keine Kommunikation mehr möglich. (Ramaya, 2011)

Eine weitere Netzwerktopologie, die verwendet wird, um ein ZigBee Netzwerk aufzubauen, ist die sogenannte **Baumtopologie**. Im Gegensatz zur Sterntopologie, ist der Koordinator nicht mehr direkt mit allen Endgeräten verbunden. Vielmehr ist die Topologie nun stufenweise aufgebaut. (Nimi, 2017)

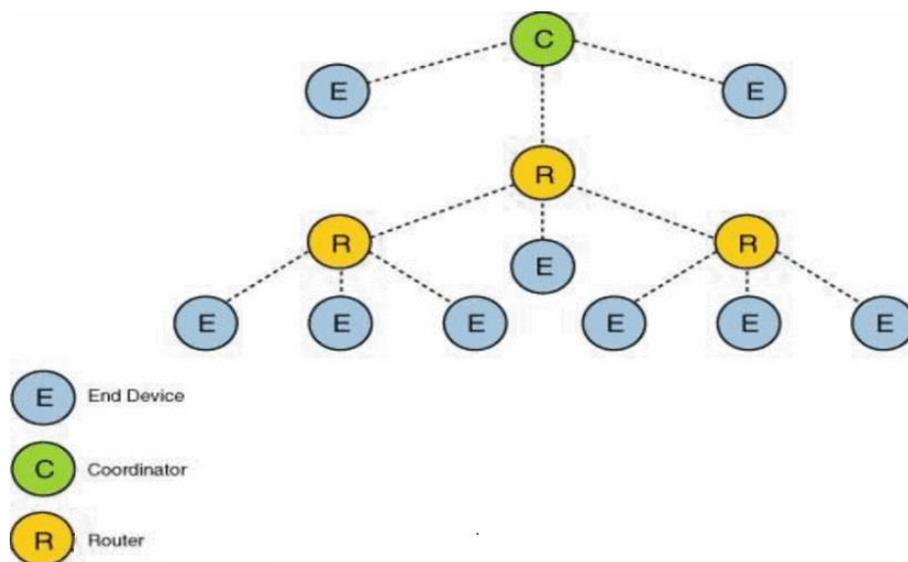


Abbildung 4: Vektor Grafik Baumtopologie, (Kumar, 2016)

In Abbildung 4 sieht man einen symbolischen Aufbau eines Netzwerkes mit Baumtopologie. Diese Topologie wird verwendet, um die Reichweite des Netzwerkes zu erhöhen. Man sieht, dass der Koordinator mit den Endgeräten in seiner unmittelbaren Funkreichweite kommunizieren kann. Damit auch eine Kommunikation mit Endgeräten außerhalb seiner Funkreichweite ermöglicht wird, werden Router in das Netzwerk eingebaut. Diese können wie auch der Koordinator, mit Endgeräten in ihrer Reichweite kommunizieren.

Möchte der Koordinator nun eine Nachricht an ein Endgerät senden, welches sich außerhalb seiner direkten Reichweite befindet, so schickt er die Nachricht an einen Router in seinem Netzwerk. Dieser empfängt die Nachricht und leitet diese an das gewünschte Endgerät weiter. Die größte Schwäche dieser Topologie zeigt sich, wenn ein Router oder der Koordinator ausfällt. Passiert dies, sind zumindest Abschnitte des Netzwerkes nicht mehr erreichbar. (Nimi, 2017)

Die dritte Topologie nennt sich **Mesh Topologie**. Ein Mesh ZigBee Netzwerk besteht, wie auch ein Netzwerk in Baumtopologie, aus einem Koordinator, Routern und Endgeräten.

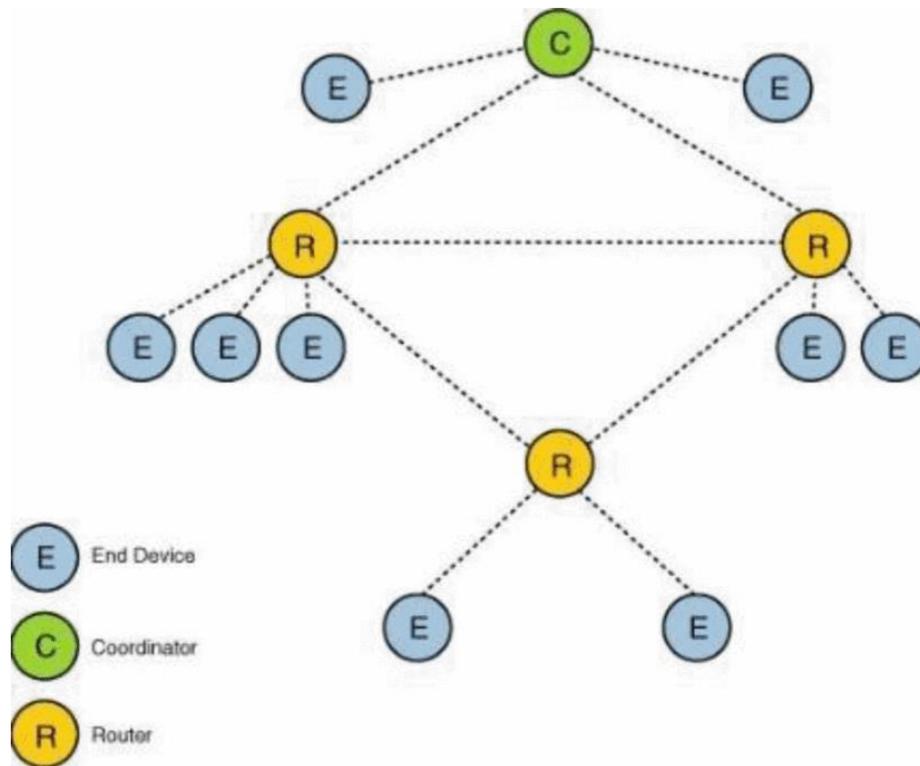


Abbildung 5: Vektor Grafik Mesh Topologie, (Kumar, 2016)

Im Gegensatz zur Baumtopologie, besteht in dieser Topologie jedoch der Vorteil, dass alle Router miteinander kommunizieren können (Abbildung 5). Daraus resultierend, gibt es mehrere Wege, um eine Nachricht von einem Netzwerkteilnehmer „A“ zu einem Netzwerkteilnehmer „B“ zu schicken. Diesen Eigenschaften ist es zu verdanken, dass bei einem Ausfall eines Routers, nur ein kleiner Teil des Netzwerks nicht mehr erreichbar ist. Des Weiteren gibt es Mechanismen in Mesh Netzwerken, welche versuchen den ausgefallenen Router zu ersetzen. (Nimi, 2017)

Technische Spezifikationen

- Zuverlässig und selbstheilend
- Kostengünstig
- Wartungsarm (vernetzen, selbstorganisierend)
- Einfach zu implementieren
- Auf Standards basierende Sicherheit [AES 128]
- Geringer Stromverbrauch (Fähigkeit, mit Batterien betrieben zu werden, gemessen in Jahren)
- Firmwareupdates aus der Ferne
- Sehr lange Akkulaufzeit
- Offenes Standard-Protokoll ohne Lizenzgebühren

2.2 Hausnotrufdienste

Eine Branche, die sich schon seit langem mit dem Erkennen von Stürzen oder Notsituationen befasst, ist die Branche der Hausnotruf Dienstleister. Die Lösungen dieser Branche sind breit gefächert und sollen Personen in einer Notlage zu einer schnellstmöglichen Hilfe durch Hilfsorganisationen wie dem Roten Kreuz verhelfen. Die eingesetzten Technologien reichen hierbei von sogenannten Notfalluhren bis hin zu einfachen Kordeln, die von der Decke hängen und beim Dranziehen einen Alarm auslösen. Meist wird durch das Auslösen eines Alarms eine Direktverbindung zu einem Call Center hergestellt, welchem die betroffene Person ihre Notlage erläutern kann. Je nach Situation wird dann eine Aktion gesetzt, um der Person in Notlage zu helfen. Diese Aktionen können je nach Anbieter unterschiedlich ausfallen. Einerseits gibt es Unternehmen die bei Eintreffen eines Notrufes sofort die nächstgelegene Hilfsorganisation (Rotes Kreuz, Samariterbund, MA70) alarmieren, andererseits werden auch Lösungen angeboten, welche in besagten Situationen einen selbständigen Anruf zu einer im Vorhinein eingespeicherten Person aufnehmen. Das größte Problem dieser Systeme ist der Fall, in dem eine Person durch einen Sturz oder einem anderen meist krankheitsbedingtem Grund nicht mehr in der Lage ist, einen Notruf abzusetzen. Man stelle sich vor, eine Person stürzt im Treppenhaus und wird ohnmächtig wird. Dieser Person ist es durch die eingetretenen Umstände nicht mehr möglich die Notrufuhr auszulösen oder den Notrufknopf zu betätigen. Ein weiterer Punkt, der in diesem Kontext angesprochen werden muss, ist die meist geringen Reichweite der sogenannten Basisstation, die mit der Uhr verbunden sein muss, um einen Notruf absetzen zu können. Diese Basisstation verfügt meistens über eine eigene SIM-Card, über welche ein Anruf zum Call Center getätigt wird. Genau hier tritt ein neues Problem auf. Steht die Basisstation an einer Position, an der sie keinen Telefonempfang hat, ist es nicht mehr möglich einen Notruf abzusetzen.

Um die Probleme, die bei den oben beschriebenen Lösungen bestehen, zu beheben, werden derzeit neue Systeme erforscht und entwickelt um Personen, welche sich in einer Notlage befinden automatisch zu erkennen. Grundsätzlich gibt es hierbei zwei Lösungsansätze, die sich durchgesetzt haben. Der erste Ansatz stützt sich auf die Technologie der Bilderkennung (siehe Kapitel 2.3). Der zweite Ansatz basiert auf einem Airbagsystem, welches Stürze erkennt (siehe Kapitel 2.4).

2.3 Notlagenerkennung durch Bilderkennung

Derzeit gibt es noch keine wirklich zuverlässigen Lösungen am Markt. Eine Vielzahl von Forschungsprojekten beschäftigen sich jedoch mit der automatischen Erkennung von Stürzen. Die Sturzerkennung basiert meist auf einer Domkamera die zentral im Zimmer installiert ist. Eine separate Software, die auf einem Gateway läuft, beschäftigt sich mit der live Auswertung der aufgezeichneten Bilder. Die Software wendet dabei in den meisten Forschungsprojekten die sogenannte Standard-Vordergrund-Hintergrund-Segmentierung an. Dabei wird der

Hintergrund des Zimmers bestimmt und den Pixeln des Hintergrunds eine 0 zugewiesen. Daraus folgt, dass das betroffene Zimmer komplett schwarz dargestellt wird, während Objekte und Personen, die das Zimmer betreten weiß dargestellt werden. Anschließend wird anhand der ausgewerteten Daten bestimmt, in welcher Lage sich eine im Raum befindliche Person befindet. Es kann somit bestimmt werden ob die Person steht, sitzt oder auf dem Boden liegt. Dies geschieht durch im Vorhinein eingelernte Muster, welche die Software verwendet, um so Situationen zu vergleichen und diese zu bewerten. (Gövercin, 2008)

2.4 Airbagsysteme zur Sturzvermeidung

Mittlerweile gibt es eine Vielzahl an Unternehmen, die sich mit der Entwicklung von zuverlässigen Airbagsystemen zur Sturzvermeidung von Fußgängern beschäftigen. Es hat sich dabei bewährt auf zwei verschiedene Messungen, ausgewertet durch einen μC , zu setzen. Zum einen ist das eine Messung durch einen Beschleunigungssensor, zum anderen setzt man auf ein Gyroskop, das die Ausrichtung des Trägers misst. (Shi, 2009)

Ob der Airbag ausgelöst wird oder nicht, entscheidet ein im Gürtel verbauter μC , welcher anhand gespeicherter Fallmuster entscheidet, ob es sich um eine fallende Person handelt oder nicht. Die Fallmuster werden von den einzelnen Firmen simuliert. Wie zuverlässig der Airbaggürtel funktioniert, ist des Weiteren abhängig davon, wie genau die Fallsimulationen der einzelnen Firmen durchgeführt wurden und wie viele dieser Simulationen vollzogen und ausgewertet wurden. Erkennt der μC einen Sturz anhand der Daten des Beschleunigungssensors und des Gyroskops, wird das Ventil einer Gaskartusche geöffnet und der Airbag aufgeblasen. (Shi, 2009)

2.5 Smarte Böden

Mit smarten Böden ist es möglich, vor allem in der Nacht, Unfälle zu vermeiden. So kann ein Orientierungslicht gezielt in den Bereichen, in denen sich eine Person befindet, eingeschaltet werden. Dieses System kann auch gut in Schlafzimmern verbaut werden, da das Orientierungslicht im Zimmer erst dann eingeschaltet wird, wenn die Person tatsächlich aus dem Bett aufsteht. Im Gegensatz dazu führen Sensoren wie Bewegungsmelder oder Präsenzmelder oft dazu, dass ein etwaiges Orientierungslicht auch ausgelöst wird, wenn die Personen sich im Schlaf bewegen. Auch in Pflegeheimen und Krankenhäusern werden solche Systeme verwendet, um dem Pflegepersonal mitzuteilen ob und wann ein Patient das Bett verlassen hat. (Sommer, kein Datum) Der smarte Boden besteht aus einem dünnen piezoelektrischen Polymer-Film, welcher unter Belastung, ein anderes Signal als im entspannten Zustand an einen Filter zurückgibt. (Minvielle, 2017)

3 Anforderungen

In diesem Kapitel werden sowohl funktionelle als auch nicht funktionelle Anforderungen definiert und beschrieben.

3.1 Funktionelle Anforderungen

3.1.1 Push Up Nachrichten empfangen

Im aufgebauten System sollen dem/der UserIn Nachrichten auf seine/ihre Apple Smart Watch gesendet werden. Dadurch soll es ermöglicht werden, dem/der UserIn systemrelevante Nachrichten bzw. Alarmierungen mitzuteilen. Um diese Funktionalität mit der, vom KfV, gewünschten, Apple Watch umzusetzen, ist ein Apple iPhone notwendig. Ohne dieses zusätzliche Smartphone ist es nicht möglich, Nachrichten auf der Apple Watch zu empfangen, da diese immer eine Verbindung zu einem Apple Smartphone benötigt.

3.1.2 Zugriff auf das Smart Home Gateway

Der Zugriff auf das Smart Home Gateway (siehe Kapitel 5.1.1) soll sowohl über eine App als auch über eine Weboberfläche möglich sein. Dies soll dem/der UserIn die maximale Flexibilität in der Bedienung einräumen. Unterstützte Betriebssysteme sollen sowohl IOS, Android, sowie Windows sein.

Eine weitere funktionelle Anforderung an das Smart Home Gateway ist, dass es nicht nur im lokalen Netzwerk erreichbar ist, sondern auch von außerhalb. Dies gibt dem/der UserIn die Möglichkeit, das Gateway von überall aus bedienen zu können.

3.1.3 Hinzufügen von Smart Home fähigen Geräten

Es soll möglich sein, dem System jederzeit weitere Geräte hinzuzufügen. Das Hinzufügen soll einerseits über die App sowie andererseits über die Weboberfläche möglich sein. Das System soll Funktechnologien wie Z-Wave, EnOcean, ZigBee und WLAN unterstützen.

3.1.4 Zeitliche Beschränkungen „Automatische Herdabschaltung“

Im Use Case „Automatische Herdabschaltung“ wird ein Herd zeitlich verzögert ausgeschaltet, sobald sich keine Personen mehr im Bereich der Küche aufhalten. Die Verzögerung soll dabei von dem/der UserIn frei bestimmbar sein. Zusätzlich soll ein verbaute Bewegungsmelder alle 30 Sekunden erneut abfragen, ob sich eine Person in der Küche aufhält oder nicht.

3.1.5 Löschen von Geräten im System

Es soll dem/der UserIn möglich sein, Geräte im System zu löschen beziehungsweise diese zurückzusetzen. Dies soll wie auch das Hinzufügen von Geräten via App und Weboberfläche möglich sein.

3.2 Nicht funktionelle Anforderungen

3.2.1 Auswahl der Geräte

Die im System verbauten Materialien und Geräte sollen im freien Handel erhältlich sein. Diese Anforderung wurde aufgestellt, da es interessierten Personen möglich sein soll die in dieser Arbeit beschriebenen Use Cases für den Eigengebrauch nachzubauen.

3.2.2 Layout des Smart Home Gateways

Das Layout des Gateways soll eine intuitive und leicht verständliche Bedienung ermöglichen. Diese Anforderung ist eine der wichtigsten, da die Hauptzielgruppe des Systems ältere Personen sind, welchen es an technischem Knowhow fehlen könnte.

3.2.3 Skalierbarkeit und Flexibilität

Das System soll von dem/der UserIn jederzeit erweitert werden können. Diese Anforderung beinhaltet nicht nur das Erweitern des Systems um Smart Home Geräte, sondern auch um das Erweitern um Funktionen, die benötigt werden, um neue Geräte miteinander zu verknüpfen.

3.2.4 Videodreh zur Vorstellung des Projekts

Im Rahmen der Projektarbeit soll in Zusammenarbeit mit der FH Technikum Wien ein Video gedreht werden. In diesem Video sollen die einzelnen Use Cases vorgestellt, erklärt und dem Viewer anhand praktischer Beispiele nähergebracht werden.

4 Planung

Dieses Kapitel beschäftigt sich mit der Hard- und Software, die in diesem Projekt benutzt wurden, um das System umzusetzen.

4.1 Hardware

4.1.1 Smart Home Gateway

Heutzutage gibt es viele verschiedene Funkstandards, die im Bereich Smart Home eingesetzt werden. Die meistverwendeten in dieser Branche sind ZigBee, Z-Wave und EnOcean. All diese Funkprotokolle wurden geschaffen, um Geräte in einem intelligenten Haushalt miteinander zu vernetzen. Es stellt sich jedoch das Problem, dass oftmals Geräte welche unterschiedliche Funkstandards unterstützen, miteinander kommunizieren sollen. Außerdem soll es dem/der UserIn ermöglicht werden, dass Smart Home nicht nur zentral, sondern auch dezentral zu steuern. (Schiller, 2016)

Aufgrund dessen werden in Smart Homes sogenannte Gateways benötigt, welche die Daten der verschiedenen Funkprotokolle interpretieren, verarbeiten und übersetzen. Grundsätzlich gibt es in diesem Kontext zwei verschiedene Arten von Gateways. Zum einen sind das universale Gateways, die eine Vielzahl an Funkprotokollen und Herstellern unterstützen, zum andern gibt es auch herstellerspezifische Gateways, welche eine Verbindung, nur mit Geräten eines speziellen Herstellers unterstützen.

Durch das Nutzen eines Gateways wird es beispielweise ermöglicht, dass ein EnOcean Funktaster mit einer ZigBee Lampe kommuniziert, während man von einem Handy aus, den Zustand des Tasters oder der Lampe überprüfen kann.



Abbildung 6: Schematische Darstellung Smart Home Gateway

4.1.2 Bedieneinheit

Ein wichtiger Bestandteil eines jeden Smart Homes, ist die Bedieneinheit. Diese stellt das Bindeglied zwischen dem/der UserIn und der digitalen Welt dar. Als Bedieneinheit werden in den meisten Fällen Smartphones, Tablets oder Terminals verwendet. In seltenen Fällen auch Stand-PCs oder Laptops.

Die Auswahl des Mediums, welches als Bedieneinheit im System verwendet werden soll, gestaltet sich jedoch schwierig. Es gibt einige Kriterien, die vom Medium erfüllt werden müssen, um eine intuitive und angenehme Bedienung des Systems zu gewährleisten.

Kriterien die zu bedenken sind:

- Bildschirmgröße
- Größe der Bedieneinheit
- Handlichkeit
- Auflösung
- Bei Smartphones Kompatibilität mit App
- Rechenleistung
- Fest-montiert oder mobil

All diese Kriterien müssen unbedingt beachtet und bewertet werden, um dem/der UserIn eine leichte und intuitive Bedienbarkeit zu bieten. So ist die Frage nach der Bildschirmgröße eine der wichtigsten, da dieses Kriterium ausschlaggebend dafür ist, ob der/die BenutzerIn das User Interface gut bedienen kann oder nicht. Auch die Kompatibilität mit der App und die Rechenleistung eines etwaigen Smartphones oder Tablets ist ein wichtiger Punkt, der beachtet werden muss.

4.1.3 PIR Bewegungsmelder

PIR (Passive Infrared) Sensoren werden im Bereich der Bewegungserkennung eingesetzt. Sie arbeiten mit temperaturabhängigen Pyrosensoren und gehören zu der Klasse der thermischen Detektoren. Thermo-Detektoren können einfallende Strahlung durch Temperaturänderungen messen. PIR Bewegungsmelder haben die Eigenschaft, nur auf Infrarot Strahlung einer bestimmten Wellenlänge zu reagieren. Dies dient dazu, dass nur Menschen erkannt werden und keine anderen sich im Spektrum befindlichen Materialien, welche Infrarotstrahlung ausstrahlen. Diese Eigenschaft erlangen die Sensoren, indem ein geeignetes absorbierendes Material über die Detektorlinse gelegt wird. Die Materialien führen dazu, dass der Bewegungsmelder nur mehr in einem Wellenbereich von 8 bis 14 μ m auslöst. Dies ist der Bereich, in dem die menschliche Infrarotstrahlung liegt. (Zappi, 2010)

PIR Bewegungsmelder sind eine kostengünstige und energieeffiziente Variante, um menschliche Bewegungen zu erkennen. Aufgrund dessen sind sie weltweit im hohen Ausmaß verbreitet und werden in vielen verschiedenen Szenarien eingesetzt. Der häufigste Einsatzort ist dabei wohl die Branche der Lichttechnik, wo Bewegungsmelder eingesetzt werden, um Lampen zu steuern.

4.2 Software

4.2.1 WebHooks

Ein WebHook ist eine Methode, um eventbasierte Daten mittels HTTP POST Methode auszusenden. Ein WebHook nutzt dabei Mechanismen, welche es ermöglichen Daten von einem Server aus an einen oder mehreren Clients zu „pushen“. Die Daten werden im Kontext eines WebHooks ausgesendet, sobald ein bestimmtes Event eintritt. Dies macht es möglich, eine effiziente Kommunikation zwischen Server und Client zu erstellen. Verwendet man keinen WebHook, ist das herkömmliche Prinzip eines, bei welchem der Client beim Server eine Anfrage stellt, um herauszufinden ob neue Daten vorhanden sind. Dies führt in vielen Situationen zu einem enormen Overhead an unnötigen Daten, da der Server, sofern keine Änderung vorliegt, dies dem Client jedes Mal aufs Neue mitteilt. Dieses Prinzip nennt man Polling. Um diesem unnötigen Datenaustausch vorzubeugen, wurden WebHooks entwickelt. Hierbei wird eine Serversoftware auf einem physikalischen Server implementiert. Diese Software reagiert auf bestimmte Events, welche im Vorhinein festgelegt wurden. (Adewole, 2018) So gibt es beispielweise WebHooks, welche auf neue Facebook Benachrichtigungen reagieren. Tritt ein Event ein, auf das der WebHook reagiert, wird dies den Clients mittels HTTP Post Methode mitgeteilt. Diese können dann auf das eingetretene Event reagieren. WebHooks können also, wie schon erwähnt, dazu beitragen, die Kommunikation beziehungsweise den Datenaustausch zwischen Client und Server, um einiges effizienter zu gestalten, da der Server seinen Clients nun immer mitteilt (push), wenn es etwas Neues gibt und nicht jedes Mal aufs Neue ein GET Request injizieren werden muss, um dies herauszufinden.

Ein großes Problem, das beim Verwenden von WebHooks beachtet werden muss, ist die Security. Ein WebHook kann über seine URL von jedem empfangen werden, sobald er diese URL kennt. Dies muss bei der Architektur eines Systems beachtet werden. (Adewole, 2018)

5 Umsetzung eines Testaufbaus im Smart Living Lab

5.1 Verwendete Hardware

5.1.1 Homee Gateway

Das Homee System ist ein sogenanntes modulares Gateway, welches für den Bereich Smart Home entwickelt wurde und auch hauptsächlich in diesem verwendet wird. Homee kann durch sogenannte Cubes in seinem Funktionsspektrum erweitert werden. Eine KNX-Schnittstelle für die Kommunikation mit ein KNX IP Gateway oder ähnlichem ist leider nicht vorhanden. Folgende Cubes werden von Homee angeboten:

- Brain Cube
- Z-Wave Cube
- ZigBee Cube
- EnOcean Cube

Jeder Cube unterstützt dabei andere Funkprotokolle. So ist es mit einem Brain Cube (weiß) möglich, Geräte wie zum Beispiel Nuki Smart Lock, Netatmo oder Belinko WeMo über WLAN zu verbinden. Der Brain Cube, ist weiters auch der einzige Cube, welcher selbständig betrieben werden kann. Er sammelt Informationen des Smartphones und ist für die Kommunikation mit der App bzw. mit dem Netzwerk verantwortlich. Alle anderen Cubes müssen in Verbindung mit dem Brain Cube betrieben werden, da diese über keine eigenständige Spannungsversorgungen verfügen. Die Spannungsversorgung wird durch das Verbinden der benötigten Cubes mit dem Brain Cube bewerkstelligt. Somit kann der Brain Cube auch als eine Base Station angesehen werden, ohne der ein Betrieb des Systems nicht möglich wäre.

Der Z-Wave Cube (Grün oder Violet) bietet dem/der UserIn die Möglichkeit, das System, um den Funkstandard Z-Wave zu erweitern.

Der ZigBee Cube (Orange) bietet dem/der UserIn die Möglichkeit, das System, um den Funkstandard ZigBee zu erweitern.

Der EnOcean Cube (Blau) bietet dem/der UserIn die Möglichkeit, das System, um den Funkstandard EnOcean zu erweitern.

Homee bietet des Weiteren eine kostenlose App für IOS und Android an und gibt dem/der UserIn die Möglichkeit, das System auch über die hauseigene Webapp zu konfigurieren (*Kapitel 5.2.1*).

5.1.2 Fibaro MotionSensor

Der Fibaro Motion Sensor ist ein Bewegungsmelder, welcher mit der PIR Technologie (siehe *Kapitel 4.1.3*) arbeitet. Er ist ein Multifunktionssensor. Durch verschiedenste verbaute Sensoren im Gerät, ist der Motion Sensor in der Lage, mehrere Messwerte zu erheben. Um diese an das Gateway zu senden, nutzt er das Funkprotokoll Z-Wave.



Abbildung 7: Fibaro Motion Sensor (Fibaro, 2018)

Der Betrieb des Geräts ist über eine CR123A Batterie möglich. Dies ermöglicht den Einsatz des Bewegungsmelders auf einfachste Art und Weise, da kein neues Kabel eingezogen werden muss, um das Gerät mit Strom zu versorgen. Die Lebensdauer der Batterien, beträgt laut Hersteller, abhängig vom sogenannten Wake-up Intervall, bis zu zwei Jahre. Das Wake-up Intervall kann über gerätspezifische Parameter eingestellt werden. Die Umstellung erfolgt am Smart Home Gateway. Die einzelnen Parameter können in der herstellereigenen Betriebsanleitung nachgelesen werden.

Der Fibaro MotionSensor verfügt neben einem PIR, welcher zur Erkennung einer Person benötigt wird, auch über weitere Sensoren. Ein integrierter Licht/UV Sensor gibt dem Benutzer die Möglichkeit, eine LUX-Messung durchzuführen. Diese kann anschließend dazu verwendet werden, eine intelligente Jalousiesteuerung umzusetzen. Ein weiteres Feature, das der Sensor bietet, ist die Möglichkeit die Temperatur im Raum zu messen. Diese Messung geschieht über einen, im Gerät verbauten Sensor. Dies macht es möglich den MotionSensor auch für eine intelligente Heizungssteuerung zu verwenden. Der letzte im Gerät verbaute Sensor ist ein Vibrationssensor. Dieser stellt Vibrationen fest und kann als sogenannter Manipulationsalarm verwendet werden. Je nach Gateway ist es möglich, einen Alarm an den/die UserIn zu senden, wenn der Manipulationsalarm ausgelöst wurde, um einen Diebstahl des Geräts zu verhindern. (Fibaro, 2018)

5.1.3 Aeotec MultiSensor 6

Der PIR Bewegungsmelder (siehe *Kapitel 4.1.3*) MultiSensor 6 der Marke Aeotec, ist ein Multifunktionales Z-Wave Gerät. Der Aeotec MultiSensor 6 ist ein speziell für den Einsatz in Smart Homes entwickeltes Gerät. Er verwendet das Funkprotokoll Z-Wave, um seine Signale per Funk an ein Gateway zu senden.



Abbildung 8: Aeotec MultiSensor 6 (Aeotec, 2018)

Der Betrieb des Geräts ist sowohl über zwei CR123A Batterien als auch über ein USB-C Kabel möglich. Dies ermöglicht den Einsatz des Bewegungsmelders auf einfachste Art und Weise, da kein neues Kabel eingezogen werden muss, um das Gerät mit Strom zu versorgen. Die Lebensdauer der Batterien, beträgt laut Hersteller, abhängig vom sogenannten Wake-up Intervall, bis zu zwei Jahre.

Der Aeotec MultiSensor verfügt neben einem PIR, welcher zur Erkennung einer Person benötigt wird, auch über weitere Sensoren. Ein integrierter Licht/UV Sensor gibt dem Benutzer die Möglichkeit, eine Lichtmessung zwischen 0 und 3000 Lux durchzuführen. Dies kann dazu verwendet werden, eine intelligente Jalousiesteuerung umzusetzen. Ein weiteres Feature, das der Sensor bietet, ist die Möglichkeit die Temperatur im Raum zu messen. Diese Messung geschieht über einen, im Gerät verbauten, Sensor, welcher auch in der Lage ist, die Luftfeuchtigkeit zu messen. Dies macht es möglich den MultiSensor 6 auch für eine intelligente Heizungssteuerung zu verwenden. Der letzte im Gerät verbaute Sensor ist ein Vibrationssensor. Dieser stellt Vibrationen fest und kann als sogenannter Manipulationsalarm verwendet werden. Je nach Gateway ist es möglich, einen Alarm an den/die UserIn zu senden, wenn der Manipulationsalarm ausgelöst wurde, um einen Diebstahl des Geräts zu verhindern. Laut Hersteller ist der Vibrationsalarm empfindlich genug, um sogar seismische Aktivitäten zu erkennen. So könnten Leute, die in einer durch Erdbeben gefährdeten Region leben, durch den MultiSensor 6 gewarnt werden, wenn die Erde anfängt zu beben. (Aeotec, 2018)

5.2 Verwendete Software

5.2.1 Homee App & Webapp

Das Homee Gateway kann per App oder Webanbindung vom User / von der Userin erreicht werden. In dieser Bachelorarbeit wurde, die von Homee kostenlos zur Verfügung gestellte IOS App und der Webzugang benutzt, um Geräte hinzuzufügen, beziehungsweise um diese zu verwalten. Die Homee Software ermöglicht dem/der UserIn, Geräte hinzuzufügen, Homeegramme und Gruppen zu erstellen sowie den Gerätestatus abzufragen. Im Kontext der Arbeit stellte sich das Erstellen von Homeegrammen als besonders hilfreich dar, da mit dieser Funktion sogenannte if-this-then-that Abläufe programmiert werden können.

5.2.2 Node-RED

Node-RED ist ein im Jahre 2013, von IBM entwickelte Open Source IDE (Integrated Development Environment). (Node-RED, 2017) Es ist ein graphisches Entwicklungstool welches speziell für den Bereich IoT entwickelt wurde und nach dem Programmierstil des Flow-Based Programming (Morrison, 2010) arbeitet. Dem Nutzer wird eine graphische Oberfläche geboten, in welcher per Drag and Drop Prinzip, graphisch dargestellte Blöcke, welche als Knoten (engl. Nodes) bezeichnet werden, in einen sogenannten Sketch gezogen werden können. Die Nodes lassen sich dann je nach spezifischer Anforderung verbinden. Die Nodes kann der/die UserIn durch graphisch dargestellte Linien miteinander verbinden. (Node-RED, 2017)

Node-RED bringt standardmäßig einige Funktionsbausteine mit, mit denen der/die UserIn einfache IoT Applikationen umsetzen kann. Sind diese Bausteine jedoch nicht ausreichend, werden dem Benutzer unterschiedliche Wege angeboten, um Node-RED zu erweitern. Zwei der Varianten, um Node-RED zu erweitern, sind das Einbinden von Plugins, sowie das Schreiben eigener Funktionsbausteine mit JavaScript.

Installation von Node-RED

In diesem Unterabschnitt wird erklärt, wie man Node-RED für einen Lokalen Windows PC installiert. Eine Installationsanleitung für Linux, Raspberry Pi, Android und weitere kann auf der offiziellen Website von Node-RED (<https://nodered.org/docs/getting-started/>) gefunden werden.

Schritt 1:

Node-RED basiert auf Node.js. Daher muss vor der eigentlichen Installation, noch Node.js heruntergeladen werden. Auf der offiziellen Website von Node.js kann ein Download für alle gängigen Systeme durchgeführt werden. Nach der Installation von Node.js, kann über den CMD-Befehl `node --version && npm -version` überprüft werden, ob die Installation richtig durchgeführt wurde. Wurde die Installation sachgemäß durchgeführt, sieht das Ergebnis, wie in Abbildung 9 gezeigt, aus.

```
C:\Users\matze>node --version && npm --version
v11.2.0
6.4.1
```

Abbildung 9: Ergebnis der Node.js Installationsüberprüfung in CMD

Schritt 2:

Node-RED wird, auf Windows Rechnern, mittels des Befehls `npm install -g --unsafe-perm node-red` installiert. Dieser Befehl wird in CMD eingegeben, um die Installation durchzuführen.

Schritt 3:

Node-RED wurde lokal auf dem Rechner installiert. Um Node-RED zu starten wird der CMD Befehl `node-red` eingegeben. Wurde die Installation richtig durchgeführt, startet Node-RED. Die Ausgabe in CMD muss der in Abbildung 10 dargestellten Ausgabe gleichen.

```
C:\Users\matze>node-red
12 Jan 17:10:58 - [info]

Welcome to Node-RED
=====

12 Jan 17:10:58 - [info] Node-RED version: v0.19.5
12 Jan 17:10:58 - [info] Node.js version: v11.2.0
12 Jan 17:10:58 - [info] Windows_NT 10.0.17134 x64 LE
12 Jan 17:10:59 - [info] Loading palette nodes
[info] "2019-01-12T16:11:00.416Z" 'Loading knx: KNXnet/IP protocol implementation for Node(>=4.x), version: 2.3.1'
12 Jan 17:11:01 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
12 Jan 17:11:01 - [warn] -----
12 Jan 17:11:01 - [warn] [node-red/tail] Not currently supported on Windows.
12 Jan 17:11:01 - [warn] -----
12 Jan 17:11:01 - [info] Settings file : \Users\matze\.node-red\settings.js
12 Jan 17:11:01 - [info] Context store : 'default' [module=memory]
12 Jan 17:11:01 - [info] User directory : \Users\matze\.node-red
12 Jan 17:11:01 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Jan 17:11:01 - [info] Flows file : \Users\matze\.node-red\flows_LAPTOP-OLSP605U.json
12 Jan 17:11:01 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
```

Abbildung 10: Node-RED gestartet

Schritt 4:

Die graphische Oberfläche von Node-RED kann nun durch die Eingabe `http://localhost:1880` in einen Browser erreicht werden.

Nodes

Die Funktionsbausteine, die in Node-RED verwendet werden, werden Nodes genannt. Die, im Kontext der Bachelorarbeit, wichtigsten Nodes werden in diesem Kapitel beschrieben.

Die „inject“ Node:

Die Inject Node (Abbildung 11) gibt dem Benutzer die Möglichkeit eine Nachricht in einen Flow zu senden. Dies kann entweder manuell, durch Anklicken des Buttons neben der Node, oder automatisch, in Form einer zyklischen Aussendung, passieren. Sie ist immer der Anfang eines jeden Flow und kann als Startbutton angesehen werden, welcher den Ablauf eines Flows startet.

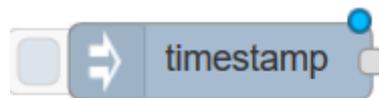


Abbildung 11: Inject Node in Node-RED

Um eine Inject Node vielfältig nutzen zu können gibt es verschiedene Eigenschaften, die man im Inject Node Menü (Abbildung 12) ändern kann.

Abbildung 12 zeigt das Einstellungsmenü der Inject Node. Das Menü ist in zwei Hauptbereiche unterteilt: 'node properties' und 'node settings'.
Im Bereich 'node properties' sind folgende Optionen zu sehen:
- 'Payload': Ein Dropdown-Menü mit der Auswahl 'timestamp'.
- 'Topic': Ein Textfeld mit dem Inhalt 'read'.
- Ein Kontrollkästchen für 'Inject once after 0.1 seconds, then'.
- 'Repeat': Ein Dropdown-Menü mit der Auswahl 'interval'.
- 'every 30 seconds': Ein Textfeld mit '30' und ein Dropdown-Menü mit 'seconds'.
- 'Name': Ein Textfeld mit dem Inhalt 'Name'.
Unter dem 'node properties' Bereich befindet sich ein gelber Hinweisfeld mit dem Text:
"Note: 'interval between times' and 'at a specific time' will use cron.
'interval' should be less than 596 hours.
See info box for details."
Der Bereich 'node settings' ist unterhalb des Hinweisfelds zu sehen, aber seine Inhalte sind nicht vollständig dargestellt.

Abbildung 12: Einstellungsmenü der Inject Node

Der in den Flow geschickte Nachrichtentyp kann in den Einstellungen der Node, den spezifischen Aufgabenstellungen angepasst werden. Zur Auswahl stehen folgende sogenannte Payloads, welche den Datentyp beschreiben:

- String
- Current Time
- Flow
- Global
- Number
- Boolean
- JSON

Neben dem Payload kann optional ein beschreibendes Topic angegeben werden. Dieses kann beispielsweise „read“ oder „write“ sein. Die Node kann manuell oder automatisch ausgelöst werden. Soll die Inject Node automatisch ausgeführt werden, so ist die sich zyklisch wiederholende Zeit, im Menüpunkt „Repeat“ anzugeben. Das maximale Intervall liegt dabei bei 24 Stunden.

Die „debug“ Node:

Die Debug Node (Abbildung 13) dient, wie der Name sagt dem Debuggen des Flows. Sie gibt die empfangene Nachricht in der Browseroberfläche von Node-RED aus und liefert damit wichtige Informationen während des Entwicklungsprozesses. Während der Ausführung des Programms kann Sie genutzt werden, um Nachrichten zu loggen und aufzuzeichnen. Jede von der Debug Node angezeigte Nachricht wird mit einem Timestamp versehen, um im Falle einer Fehlfunktion, den Zeitpunkt des Eintretens bestimmen zu können.



Abbildung 13: Debug Node in Node-RED

Die „knx device“ Node:

Diese Node ist nicht standardmäßig in Node-RED verfügbar. Sie ist ein Funktionsbaustein welcher durch das node-red-contrib-knxjs Plugin hinzugefügt wird. Mit dieser Node ist es möglich den Zustand eines KNX Geräts über einen KNX IP Router abzufragen, oder dem Gerät einen neuen Status zuzusenden.



Abbildung 14: KNX Device Node in Node-RED

Um die KNX Device Node zu nutzen, müssen einige Einstellungen im Einstellungsmenü (Abbildung 15) getroffen werden.

The screenshot shows a configuration window titled "Edit knx-device node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below this is a section labeled "node properties" with a downward arrow. It contains several fields with expandable icons on the left: "Name" (text input with "Strommessung"), "Definition Mode" (dropdown menu with "Manual"), "Connection" (text input with "224.0.23.12:3671 phys:15.15.15" and a small edit icon), "Group Address" (text input with "11/5/172"), "Status GA" (empty text input), and "DPT" (dropdown menu with "DPT9 (16-bit floating point)" and a "See this" link). Below the "DPT" field is a blue link: "document on how to format your payload". At the bottom of the window is a section labeled "node settings" with a rightward arrow, which is currently collapsed.

Abbildung 15: Einstellungsmenü der KNX Device Node

Im Einstellungsmenü gibt es verschiedene Punkte, die eingestellt werden können. Der erste wichtige Punkt nennt sich „Connection“. Hier muss ein KNX Controller erstellt werden, um eine Verbindung zu einer KNX Installation herstellen zu können. Hierfür wird die IP-Adresse, die physische Adresse und der IP Port des KNX IP Routers benötigt.

Der nächste einzustellende Menüpunkt nennt sich „Group Adress“. Hier muss die Gruppenadresse des gewünschten KNX Aktors oder Sensors eingetragen werden.

Im Punkt „DPT“ muss der Datapoint Type des KNX Geräts eingestellt werden. Dieser kann beispielweise „DPT9“ lauten. Dies würde bedeuten, dass das KNX Gerät seine Daten in Form eines 16-Bit floating Point Formats sendet oder empfängt.

Um festzulegen, ob der Zustand des gewünschten KNX Geräts ausgelesen werden soll oder ob ein Status an das Gerät geschickt werden soll, muss der KNX Device Node entweder „read“ oder „write“ über das Topic schicken.

Die „function“ Node:

Die Function Node (Abbildung 16) gibt dem Benutzer die Möglichkeit, eigene Funktionsbausteine zu programmieren. Die verwendete Programmiersprache ist JavaScript.



Abbildung 16: Function Node in Node-RED

Mit diesem Funktionsblock ist es möglich, direkt in einem Node-RED Flow zu programmieren. Je nach erstelltem Programm ist der neu entstandene Funktionsblock in der Lage auf die im Flow gesendeten Nachrichten, spezifisch zu reagieren. Anwendungsgebiete können beispielsweise Berechnungen auf Grundlage von empfangenen KNX Sensordaten (Lichtmessung, Strommessung, etc.) oder benutzerspezifische Abfragen sein.

5.3 Szenario Herdabschaltung

5.3.1 Allgemein

5.3.1.1 Homee

Im Szenario Herdabschaltung wird neben dem Homee Brain Cube auch der Z-Wave- und der ZigBee Cube verwendet (siehe *Kapitel 5.1.1*), um die benötigten Sensoren in das System einzubinden. Der Funkstandard Z-Wave, wird im *Kapitel 2.1.1* beschrieben. Der Funkstandard ZigBee, wird im *Kapitel 2.1.2* beschrieben.

5.3.1.2 PIR Bewegungsmelder “Küche”

In diesem Use Case soll unter anderem erkannt werden ob sich eine Person in der Küche aufhält. Um dies herauszufinden wird ein smarterer PIR Bewegungsmelder (siehe *Kapitel 4.1.3*) verwendet. Der Bewegungsmelder, der in diesem Projekt verwendet wurde, ist der Multi Senso 6 der Marke AeonLab (siehe *Kapitel 5.1.3*).

5.3.1.3 Siemens Leistungsaktor

Der Leistungsaktor der Marke Siemens, wird verwendet, um den von der Dunstabzugshaube verbrauchten Strom zu messen. Mit Hilfe des Aktors kann eine Leistungsmessung durchgeführt werden. So wird erkannt, ob die Dunstabzugshaube und der Herd in Betrieb sind oder nicht. Der Siemens Leistungsaktor ist ein KNX-verifiziertes Gerät, welches über eine im Smart Living Lab installierten KNX-Bus kommuniziert.

5.3.1.4 WebHook

Die Umsetzung des Use Cases macht einen Einsatz von WebHooks (siehe *Kapitel 4.2.1*) notwendig. Diese WebHooks werden verwendet, um auf eintretende Events zu reagieren.

5.3.1.5 Node-RED

Node-RED wird verwendet, um die Siemens Leistungsaktoren mit dem Funknetzwerk kommunizieren zu lassen.

5.3.2 Umsetzung

Aufbau

Wie schon in *Kapitel 1.4.1* beschrieben, beschäftigt sich dieser Use Case mit dem automatischen Abschalten eines Herds, wenn sich keine Person in der Küche befindet. Hierfür wurde in erster Linie ein Bewegungsmelder der Firma Aeotec (siehe *Kapitel 5.1.3*) an der Decke angebracht. Dieser hat die Aufgabe den Küchenbereich zu überwachen. Stellt der Bewegungsmelder keine Bewegung in der Küche fest, so wird davon ausgegangen, dass sich keine Person mehr in der Küche aufhält. Der Bewegungsmelder wird im Batteriebetrieb betrieben und nutzt das Z-Wave Funkprotokoll zur Kommunikation mit dem Homee Gateway. Das Homee Gateway hat die Aufgabe die empfangenen Daten des Bewegungsmelders auszuwerten und zu interpretieren. Die vom Homee ausgewerteten Daten werden in weiterer Folge an die Serversoftware Node-RED gesendet. Diese läuft auf einem Windows Stand-PC unter der URL: <http://172.22.0.192:1880/> und ist im *aat-technikum2,4GHz* Netzwerk erreichbar. Node-RED dient als Bindeglied zwischen einem Homee Funkgateway (siehe *Kapitel 5.1.1*) und der im Smart Living Lab aufgebauten KNX Installation. Durch ein Node-RED Plugin Namens *node-red-contrib-knxjs*, das auf der physischen Servermaschine installiert wurde, ist es möglich von Node-RED aus, Signale an einen KNX IP Router zu senden.



Abbildung 17: Verwendete Geräte im Use Case Herdabschaltung



Abbildung 18: Bewegungsmelder (li., ob.) und zu überwachender Bereich (rot)

Funktioneller Ablauf

Der insgesamt funktionelle Ablauf dieses Use Cases gliedert sich in vier Hauptaufgaben:

- 1.) Zustandserkennung Herd
- 2.) Erkennen, ob sich eine Person in der Küche aufhält
- 3.) Verbindung zwischen Homee Gateway und KNX IP Router schaffen und Herd im definierten Fall ausschalten

Schritt 1:

Der erste der vier Hauptabläufe wird mit einem Leistungsaktor der Firma Siemens umgesetzt. Dieser wurden durch qualifiziertes Fachpersonal verbaut. Es handelt sich hierbei um einen KNX Leistungsaktor, der über einen KNX-Feldbus mit dem KNX IP Router kommuniziert. Das Anbringen des Leistungsaktors ermöglicht es den aktuellen Stromverbrauch der Abzugshaube des Herds zu messen. So lässt sich leicht feststellen, ob die Kochplatte in Betrieb ist oder nicht. Die vom Leistungsaktor ermittelten Werte werden darauffolgend von der Serversoftware Node-RED abgefragt. Dies geschieht in einem zyklischen Intervall von 30 Sekunden. Die Interpretation der Werte durch Node-RED wird in *Schritt 3* ausführlicher erklärt.

Schritt 2:

Nachdem der Zustand der Herdplatte durch das Lösen der ersten Hauptaufgabe bestimmt werden kann, muss nun festgestellt werden, ob sich eine Person in der Küche aufhält oder nicht. Hierfür wird auf den in der Küche angebrachten Funk-Bewegungsmelder zurückgegriffen. Dieser Bewegungsmelder wird genutzt, um Bewegungen in der Küche zu erkennen. Die durch den Bewegungsmelder gewonnenen Erkenntnisse, also ob sich eine Person in der Küche aufhält oder nicht, werden dem Homee Gateway über Funk mitgeteilt. Das Homee Gateway dient in diesem speziellen Fall nur als Zwischenstation für die durch den

Bewegungsmelder gesammelten Daten. Auf die Daten, die von Homee interpretiert wurden, wird im nächsten Schritt von Node-RED zugegriffen, da dieser nicht direkt mit dem verbauten Bewegungsmelder (MultiSensor 6) kommunizieren kann. Das Einbinden des MultiSensor 6 in das System wird in Anhang A/ S.60,63 beschrieben,

Schritt 3:

Die Verbindung des Z-Wave Funknetzwerkes und der KNX Installation wird mit dem Serverprogramm Node-RED realisiert. Dafür muss das kostenlose Programm Node-RED installiert werden (siehe *Kapitel 5.2.2*). Nach der erfolgreichen Installation muss Node-RED mittels cmd Befehl „*node-red*“ gestartet werden. Nach Eingabe dieses Befehls wird Node-RED lokal auf dem verwendeten Windows PC gestartet. Dieser wird in Folge als physischer Server verwendet. Der PC ist mit dem *aat-technikum2,4GHz* Netzwerks des Smart Living Labs verbunden. Sobald Node-RED gestartet wurde, kann von jedem Teilnehmer im Netzwerk darauf zugegriffen werden. Der Zugriff erfolgt durch das eingeben einer URL im folgenden Format: *http://< IP-Adresse des PCs >:1880*. Im Fall des Bachelorprojects lautet die URL: *http://172.22.0.192:1880*. Wurde die URL richtig eingegeben erscheint die browserbasierte-Benutzeroberfläche von Node-RED (Abbildung 19)

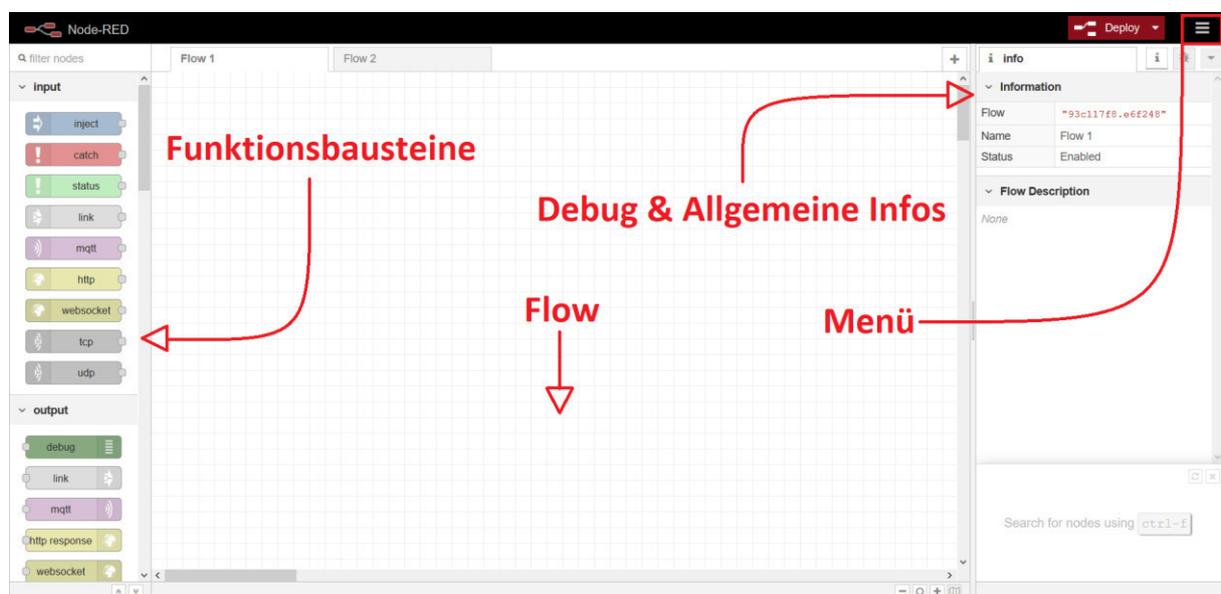


Abbildung 19: Node-RED Weboberfläche

Nun kann der Flow welcher benötigt wird, um den Use Case umzusetzen, erstellt werden. Dafür werden die einzelnen Funktionsbausteine in den Flow gezogen und anschließend verbunden.

Der Flow muss in der Lage sein, die Daten des Bewegungsmelders in der Küche über den Homee Gateway zu empfangen. Auch die Messung des in Schritt 1 beschriebenen Leistungsaktors muss in Node-RED empfangen werden. Die durch die Messungen ermittelten

Daten werden gebraucht, um festzustellen, ob sich eine Person in der Küche befindet und ob der Herd eingeschaltet ist. Um diese Funktionalitäten einzubinden ist die Installation des Plugins *node-red-contrib-knxjs* bzw. *node-red-contrib-homee* nötig. Die Plugins können direkt in der Weboberfläche von Node-RED eingebunden werden. Um sie einzubinden muss folgendermaßen vorgegangen werden:

- „Menü“ -> „Manage Palette“
- User Settings öffnen sich (Abbildung 20)
- Nach *node-red-contrib-knxjs* bzw. *node-red-contrib-homee* suchen.
- Plugin installieren.

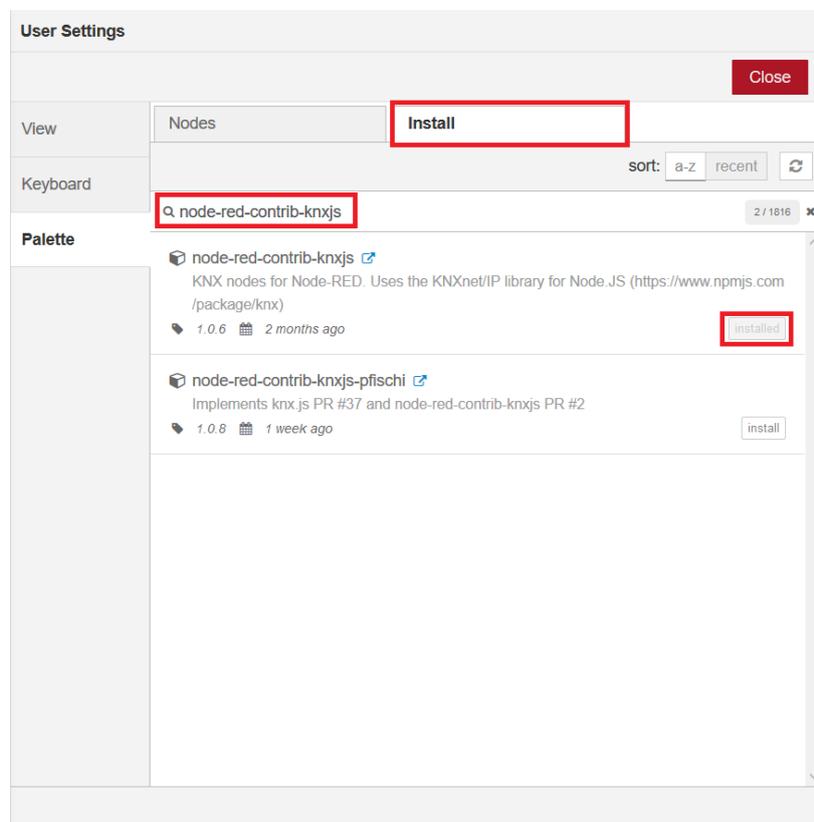


Abbildung 20: User Settings in Node-RED

Nach erfolgreicher Installation, sind sowohl drei KNX sowie ein Homee Funktionsbaustein hinzugefügt worden. Diese findet man im Funktionsbausteinmenü im Unterpunkt „Home Automation“ bzw. „Function“. Durch die eingebundenen Bausteine ist es nun möglich, folgenden Flow zu entwerfen:

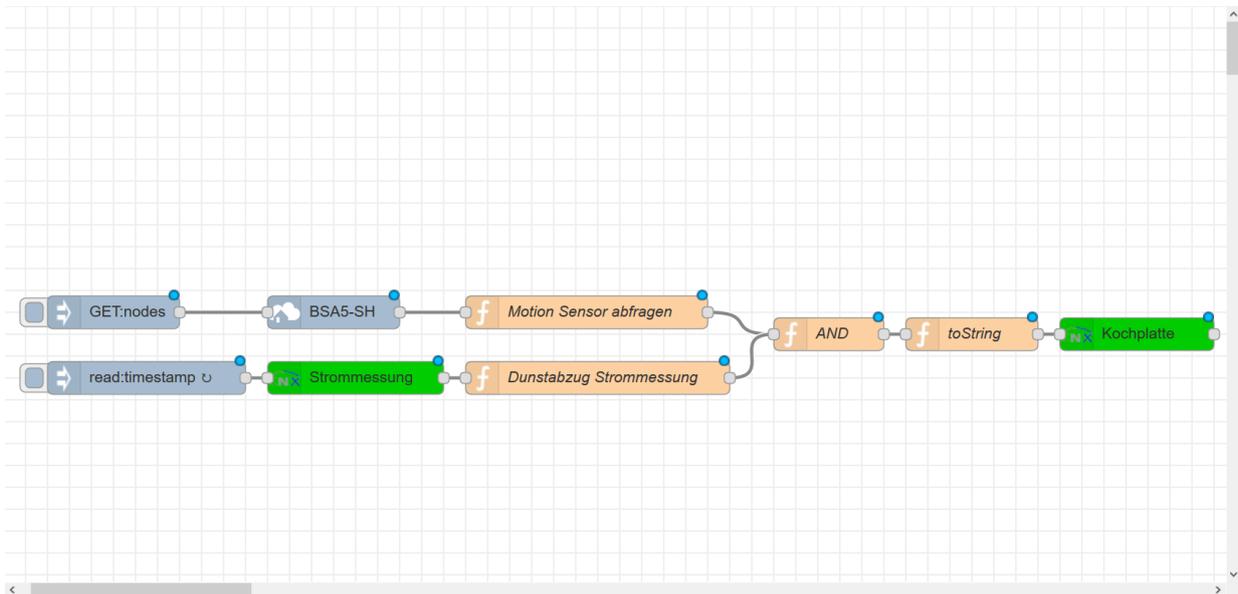


Abbildung 21: Node-RED Flow für den Use Case „Automatische Herdabschaltung“

Der in Abbildung 21 zu sehende Flow, setzt den Schritt 3 zur Gänze um. Es werden alle vom Homee gespeicherten Daten empfangen und die gespeicherten Daten des Bewegungsmelders verarbeitet. Auch die Werte der Leistungsaktoren werden abgefragt und verarbeitet.

Die einzelnen Schritte können in sechs Punkte unterteilt werden, welche die genaue Funktionalität des Flows erklären.

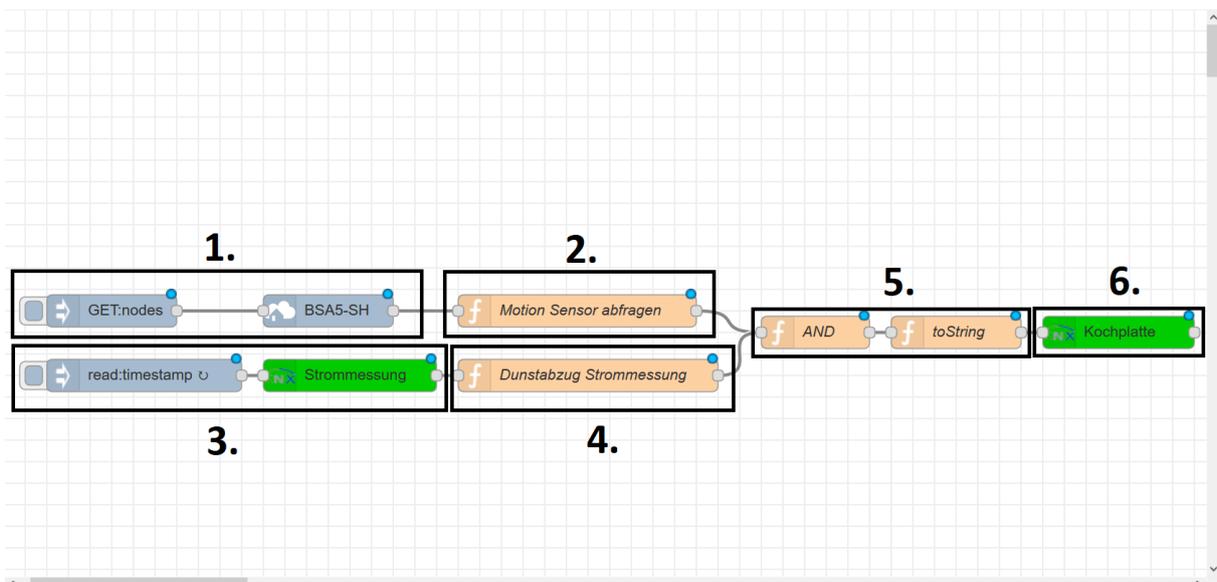


Abbildung 22: Unterteilung des Flows in sechs Punkte

Punkt 1:

In diesem Punkt werden die Daten des Homee Gateways empfangen. Dies geschieht durch eine Kombination aus einer Inject Node und einer Homee Node. Sind die beiden Nodes richtig konfiguriert, empfängt Node-RED jedes Mal alle vom Homee gespeicherten Daten, wenn sich der Zustand eines Sensors oder Aktors ändert. Um dies umzusetzen, müssen die Nodes wie in Abbildung 23 zu sehen, konfiguriert werden.

The image shows two side-by-side configuration panels in Node-RED. The left panel is titled 'Edit inject node' and the right panel is titled 'Edit homee node'. Both panels have a 'Delete' button on the left and 'Cancel' and 'Done' buttons on the right. The 'Edit inject node' panel has a 'node properties' section with a 'Payload' dropdown set to 'GET:nodes', an empty 'Topic' field, an 'Inject once after' checkbox with a value of '0.1' seconds, a 'Repeat' dropdown set to 'none', and a 'Name' field with the value 'Name'. A yellow note box contains the text: 'Note: "interval between times" and "at a specific time" will use cron. "interval" should be less than 596 hours. See info box for details.' The 'Edit homee node' panel has a 'node properties' section with a 'Name' field set to 'BSA5-SH', an 'IP/Host' field set to '172.22.0.61', a 'Username' field set to 'BSA5-SH', and a 'Password' field with masked characters. Both panels have a 'node settings' section at the bottom.

Abbildung 23: Konfiguration GET:nodes (li.) und Homee Node (re.)

In der Homee Node müssen neben der IP Adresse des Homees, auch der Username und das Passwort des Homee Gateways eingegeben werden. In unserem Fall lautet die statisch vergebene IP Adresse des Homee Gateways *172.22.2.61*. Der Username, sowie das Passwort, lauten *BSA5-SH*.

Punkt 2:

Hier werden die vom Homee empfangenen Daten in einer Funktion verarbeitet. Diese ist in JavaScript geschrieben und hat die Aufgabe, zu überprüfen, ob der Status des Bewegungsmelders 0 (keine Bewegung erkannt) oder 1 (Bewegung erkannt) ist. (Abbildung 24)

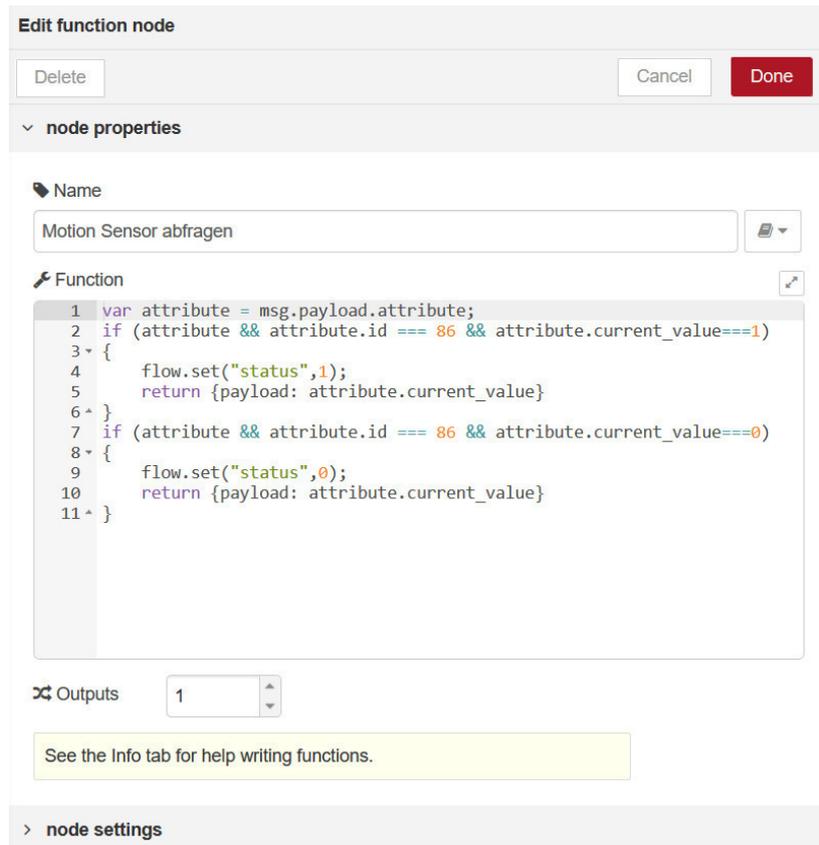


Abbildung 24: Abfragen des Bewegungsmelders

In der vorangehenden Abbildung 24 ist der JavaScript Code zu sehen, welcher von Node RED verwendet wird, um die Daten des Bewegungsmelders der Küche abzufragen. Dies geschieht durch das Auslesen des `msg.payload`. In diesem befinden sich alle Daten, welche in Node RED durch den Flow geschickt werden. Anschließend wird das Attribut mit der ID 86 genommen und abgefragt, ob der Wert dieses Attributs 0 oder 1 ist. Das Attribut mit der ID 86 ist im Fall dieses Projekts eine Variable, welche vom Homee empfangen wurde und den Zustand des Bewegungsmelders gespeichert hat. Nachdem die Abfragen, zu sehen in Zeile zwei und sieben, durchgeführt wurden, wird je nach Zustand, eine Variable gesetzt, welche den Wert 0 oder 1 zugewiesen bekommt. Dies geschieht mit dem Befehl `flow.set(<"Variablennamen", "Wert">)`. Auf die gesetzte Variable wird in Punkt 5 zugegriffen.

Punkt 3:

Das Ergebnis der Leistungsmessung des KNX Leistungsaktors soll ausgelesen werden. Diese Daten werden benötigt, um in Punkt 4 zu bestimmen, ob die Kochplatte eingeschaltet ist oder nicht. Um diese Funktionalität in Node RED zu implementieren, wird eine *Inject Node* in Verbindung mit einer *KNX device Node* verwendet. (Abbildung 25)

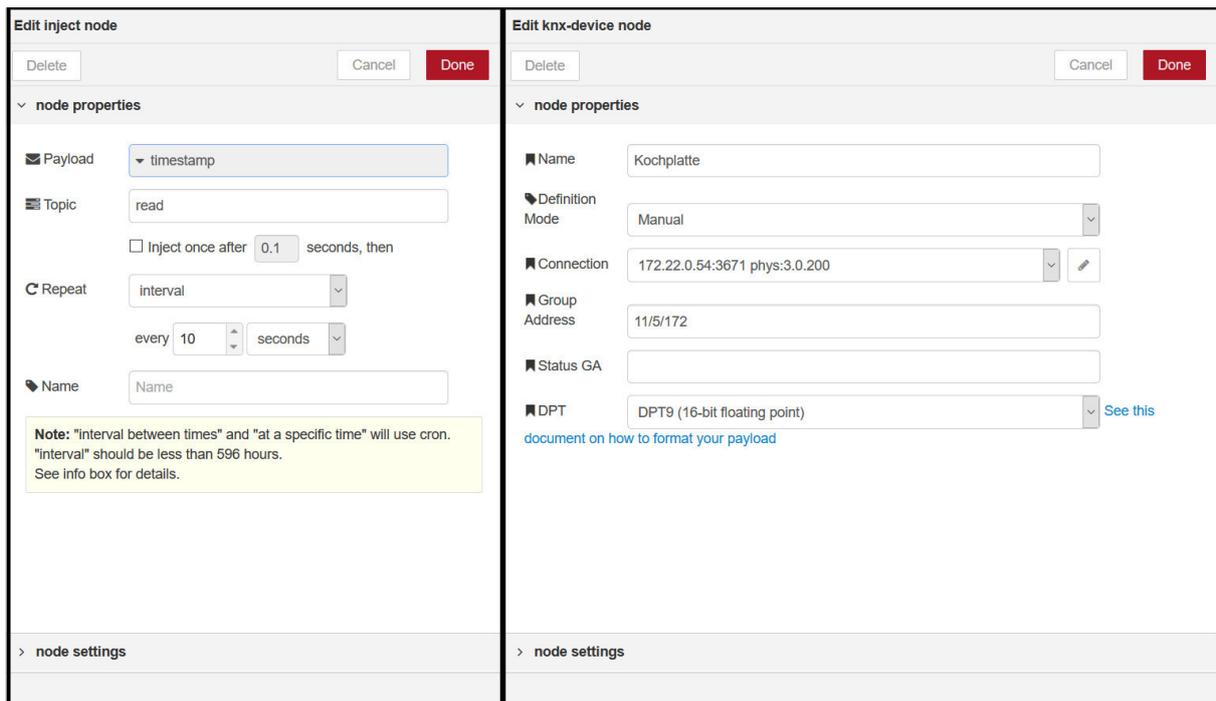


Abbildung 25: Konfigurationseinstellung Inject Node (li.) und KNX Device Node (re.)

In Abbildung 25 sind die Konfigurationseinstellungen der zwei benötigten Nodes zu sehen. In der Inject Node muss als Payload, der Typ „*timestamp*“ ausgewählt werden. Als Topic ist „*read*“ einzutragen. Diese beiden Einstellungen sind vorzunehmen, da nur diese Einstellungen von der KNX Device Node, welche mit der Inject Node verbunden ist, verarbeitet werden können. Das Topic „*read*“ gibt dabei an, dass Daten vom KNX Gerät ausgelesen werden sollen. Die Inject Node wird intervallmäßig im Abstand von 10 Sekunden ausgelöst. Die Intervallauslösung bedingt, dass der *read* Befehl alle 10 Sekunden ausgelöst wird. Die Zeit wurde mit 10 Sekunden gewählt, um eine zu hohe Belastung des Systems, durch übermäßiges Abfragen zu verhindern.

Hat die Inject Node ausgelöst, bekommt die KNX Device Node den Befehl, Daten von einem KNX Gerät auszulesen. Um auf das Gerät zugreifen zu können, müssen in der KNX Device Node (Abbildung 25 (re.)) einige Voreinstellungen getroffen werden. In erster Linie muss der KNX Device Node mitgeteilt werden, über welchen KNX IP Router er mit dem gewünschten Gerät kommunizieren kann. Dazu muss in den Node Einstellungen ein neuer KNX-Controller

erstellt werden (Abbildung 26). Die Erstellung dieses Controllers benötigt neben der IP Adresse des KNX IP Routers, auch den IP Port und die physikalische Adresse. Im Fall dieses Projekts lautet die IP Adresse *172.22.0.54*. Der Port ist standardmäßig auf *3671* gesetzt und die physikalische Adresse lautet *3.0.200*. Wurde der Controller erstellt, muss der Node nun die Gruppenadresse des gewünschten Geräts, sowie der Datapoint Type mitgeteilt werden. Dies geschieht durch eintragen der Daten in die entsprechenden Felder der KNX Device Node. Im Fall des Projekts lautet die Gruppenadresse des Leistungsaktor *11/5/172* und der Datentyp lautet *DPT9 (16-Bit floating point)*. Sind all diese Einstellungen getroffen, so ist Node RED in der Lage mit dem gewünschten KNX IP Router zu kommunizieren, und die Daten eines Geräts auszulesen.

Punkt 4:

Die Daten, welche in Punkt 3 vom KNX Gerät ausgelesen wurden, werden nun verarbeitet und interpretiert. Verwendet wird hierfür eine function Node. Diese Node hat die Aufgabe, die Daten mit einem im Vorhinein festgelegtem Schwellwert von 2mA zu vergleichen. Ist der gemessene Stromverbrauch des Leistungsaktors höher als dieser Schwellwert, so wertet die Funktion dies als „Kochplatte eingeschaltet“ und vergibt einer Variable den Wert 1. Anderenfalls wird die Variable auf 0 gesetzt. (Abbildung 26).

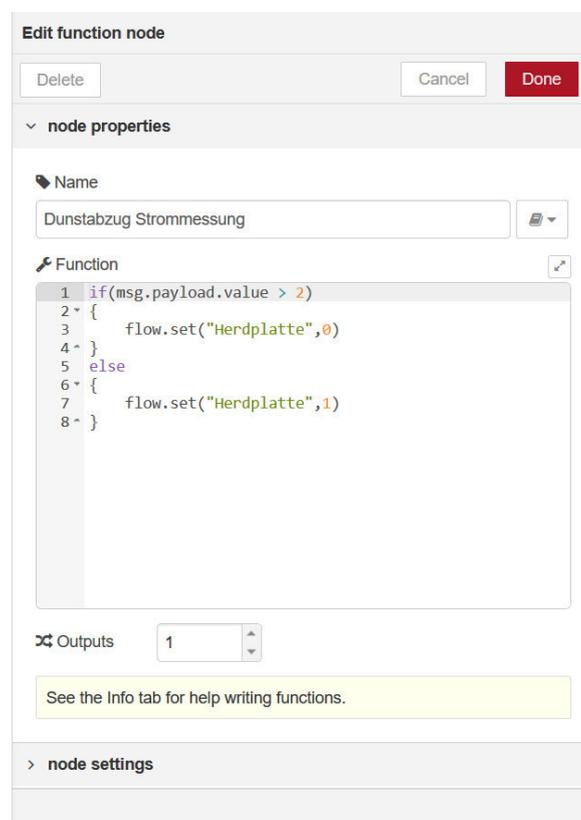


Abbildung 26: Auswertung der Strommessung

Der Vergleich der Messung mit dem Schwellwert, wird durch eine *if-else* Anweisung umgesetzt. Ist der Wert größer als 2, wird die Variable *Herdplatte*, durch den Befehl *flow.set(„Herdplatte“,1)* auf 1 gesetzt. Anderenfalls wird der Variable der Wert 0 zugewiesen.

Punkt 5:

In Punkt 5 geht es darum, die erhaltenen Daten vom Leistungsaktor und dem Bewegungsmelder mittels einer AND Funktion zu verarbeiten. Die AND Funktion arbeitet wie ein AND Logikbaustein und hat die Aufgabe, einer Variable den Wert 1 zuzuweisen, falls der Bewegungsmelder keine Bewegung mehr erkennt und der Leistungsaktor den definierten Schwellwert der Strommessung überschritten hat.

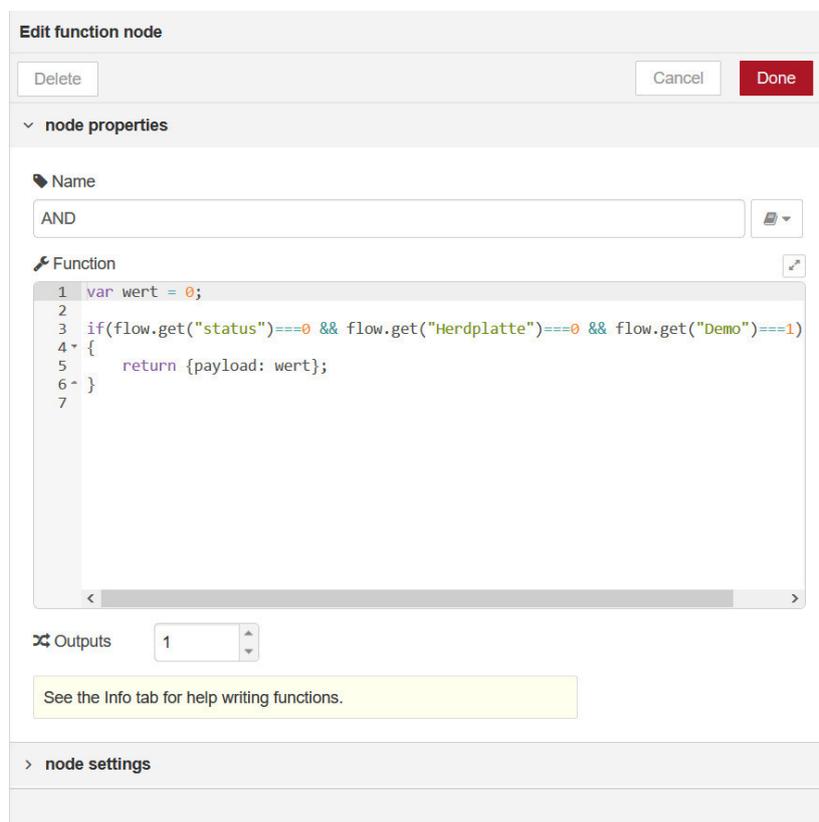


Abbildung 27: AND-Funktion in NodeRed

In Abbildung 27 ist der Code der AND Funktion in NodeRed zu sehen. Diese wurde mit einem Funktionsbaustein umgesetzt. In Zeile 1 wird die Variable *wert* erstellt, welche den Wert 0 zugewiesen bekommt. Die Variable wird in Zeile 5 benutzt, um den Wert 0 an die Funktion *toString*, welche mit der AND Funktion verbunden ist, zu senden. In Zeile 3 wird eine *if*-Abfrage durchgeführt. Diese Abfrage vergleicht, ob der Status der Variable *status*, welche den Status des Bewegungsmelders beinhaltet, und der Status der Variable *Herdplatte* den Wert 0 haben. Weiters wird in der Abfrage abgefragt, ob die Variable *Demo* den Wert 1 hat. Die Variable *Demo* dient dem Speichern des Status, ob der Demo Mode eingeschaltet ist, also den Wert 1

hat, oder ausgeschaltet ist, also den Wert 0 hat. Genauere Informationen zum Demo Mode finden sich in der Bachelorarbeit von Schweiger Manuel. (Schweiger, 2019)

Ist die Abfrage erfolgreich, so wird der Wert 0 mit dem Befehl `return{payload: wert};` an die Funktion `toString` übergeben. Die Funktion `toString` wird verwendet, um der KNX-Device Node einen Ein- bzw. Ausschaltbefehl mitzuteilen. Die Nachricht muss der Node dabei in einem bestimmten Format übergeben werden, da sie ansonsten nicht verarbeitet werden kann. Die Formatierung der Nachricht passiert in der Funktionsnode `toString`, zu sehen in Abbildung 28.

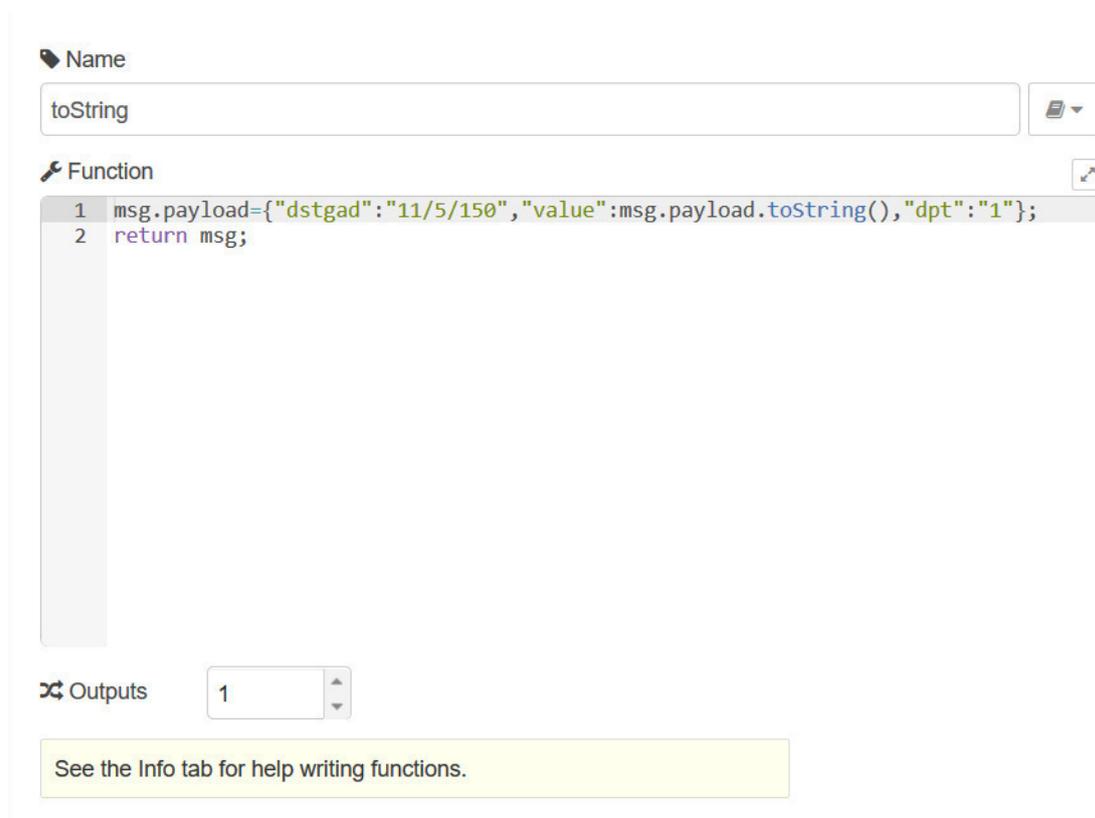


Abbildung 28: `toString` Funktion in NodeRED

In Zeile 1 zu sehen in Abbildung 28, wird eine Nachricht erstellt, welche die KNX-Device Node verarbeiten kann. Die Struktur der Nachricht muss dabei wie folgt lauten:

`{\"dstgad\": \"<Gruppenadresse des KNX Geräts>\", \"value\": \"< 0 oder 1>\", \"dpt\": \"<Datapoint type>\"}`

Im Fall der `toString` Funktion lautet die Gruppenadresse des KNX Leistungsaktors 11/5/150. Der Wert der übergeben werden soll, steht im `msg.payload`, welcher von der **AND** Funktion übergeben wurde. Der datapoint type des KNX Geräts ist 1. Der datapoint type kann der Betriebsanleitung des Geräts entnommen werden.

Zeile 2 sendet die kreierte Nachricht, mittels des Befehls *return msg*; an die KNX Device Node welche im nächsten Punkt beschrieben wird.

Punkt 6:

Der Punkt 6 wird mit einer KNX-Device Node umgesetzt. Diese Node dient als Schnittstelle zwischen NodeRed und der KNX Installation. Mit dieser Node ist es nun möglich, die in Punkt 5 erstellte Nachricht an den KNX Bus zu senden. Wurde die Nachricht in Punkt 5 richtig erstellt, so wird der Wert, welcher in der Nachricht angegeben wurde, an die in der Nachricht definierte Gruppenadresse geschickt. Ist der Wert in der Nachricht *0*, so wird die Stromzufuhr der Kochplatte mittels dem Leistungsaktor gekappt. Der Leistungsaktor kann dann per Hand wieder eingeschaltet werden.

5.4 Szenario Kindersicherung

5.4.1 Allgemein

5.4.1.1 Homee

In diesem Use Case wird wie auch im Use Case „Herdabschaltung“, ein Homee Gateway genutzt, um mit den Sensoren im System zu kommunizieren. Benötigt werden für diesen Use Case neben dem Brain Cube auch der EnOcean Cube und der Z-Wave Cube.

5.4.1.2 Fibaro Motion Sensor

Der Fibaro Motion Sensor ist ein PIR-Bewegungsmelder (*siehe 4.1.3*), welcher in diesem Use Case verbaut wird, um Erwachsene Personen im festgelegten Bereich zu erkennen. Hierfür wird seine Empfindlichkeit über die Homee App auf „Niedrig“ gestellt. Auch der Winkel, in dem der Sensor angebracht wird, wird berücksichtigt, um sicherzustellen, dass nur auf Bewegungen, welche in einer Mindesthöhe von 50cm passieren, reagiert wird.

5.4.1.3 Aeon Labs Aeotec MultiSensor 6

Der Aeotec MultiSensor 6 wird verwendet, um alle Bewegungen, in einem vordefinierten Bereich vor dem zu überwachenden Fenster zu erkennen. Dafür wird seine Empfindlichkeit im Expertenmodus der Homee App, auf „Hoch“ gesetzt. Durch diese Einstellung nimmt der Bewegungsmelder schon kleinste Bewegungen wahr. Dies ist nötig, um auch Kleinkinder und Babys zu erkennen.

5.4.2 Umsetzung

Wie schon in Kapitel 1.4.2 beschrieben, beschäftigt sich dieser Use Case damit, die Aufsichtspersonen eines Kindes oder Babys zu alarmieren, falls sich dieses einem geöffneten Fenster gefährlich nähert. So sollen Stürze von Kindern oder Babys aus dem Fenster verhindert werden, indem die alarmierte Person rechtzeitig eingreifen kann. Die konkrete Umsetzung besteht dabei aus einem Homee Brain Cube, einem Homee Z-Wave Cube, einem Homee EnOcean Cube, einem Fibaro Motion Sensor, einem Aeotec MultiSensor 6 und einem EnOcean Fenstergriff der Marke Hoppe. Zur Alarmierung der Aufsichtsperson wird auf eine Apple Watch und ein Apple iPad zurückgegriffen, um der zu benachrichtigenden Person eine Nachricht zu senden. Zusätzlich besteht die Möglichkeit einer rein visuellen Alarmierung, mittels einer Philips HUE LED Strip. Diese kann im definierten Fall beginnen zu blinken, um einen Alarm anzuzeigen. Im Szenario muss in erster Linie ein Bereich festgelegt werden, den die Bewegungsmelder auf eintretende Bewegungen überwachen sollen. Die Sensoren müssen dann in ihrer Entfernung und ihrem Winkel so installiert werden, dass sie den zu überwachenden Bereich auch tatsächlich überwachen. Eine allgemeine Angabe, welche Entfernung und welcher Winkel hierbei zu wählen sind, ist schwer zu treffen, da vor allem der Winkel des Fibaro Motion Sensors, auf das zu überwachende Kind einzustellen ist.



Abbildung 29: Verwendete Geräte im Use Case Kindersicherung

Um ein Kind mittels der zwei verbauten Bewegungsmelder zu erkennen, ist es nötig, diese logisch miteinander zu verknüpfen. Dafür wird der erste Bewegungsmelder (Fibaro Motion Sensor), wie schon beschrieben, so kalibriert, dass er nur Personen im zu überwachenden Bereich erkennt, welche größer als 50cm sind. Der zweite Bewegungsmelder (Aeon MultiSensor 6) hingegen, wird so kalibriert, dass er jegliche Bewegung im Überwachungsbereich wahrnimmt. Auf Basis der getroffenen Einstellungen, kann nun ein Homeegramm am Homee erstellt werden, welches nur dann einen Alarm ausgibt, wenn der Aeon MultiSensor 6 eine Bewegung erkennt, während der Fibaro Motion Sensor keine Bewegung erkennt. Dies bedeutet nämlich, dass ein Kind erkannt wurde. Würde auch der Fibaro Motion Sensor eine Bewegung erkennen, so würde dies aufgrund der getroffenen Einstellungen bedeuten das sich eine erwachsene Person im Raum befindet. Veranschaulicht wird das Prinzip in Abbildung 30.

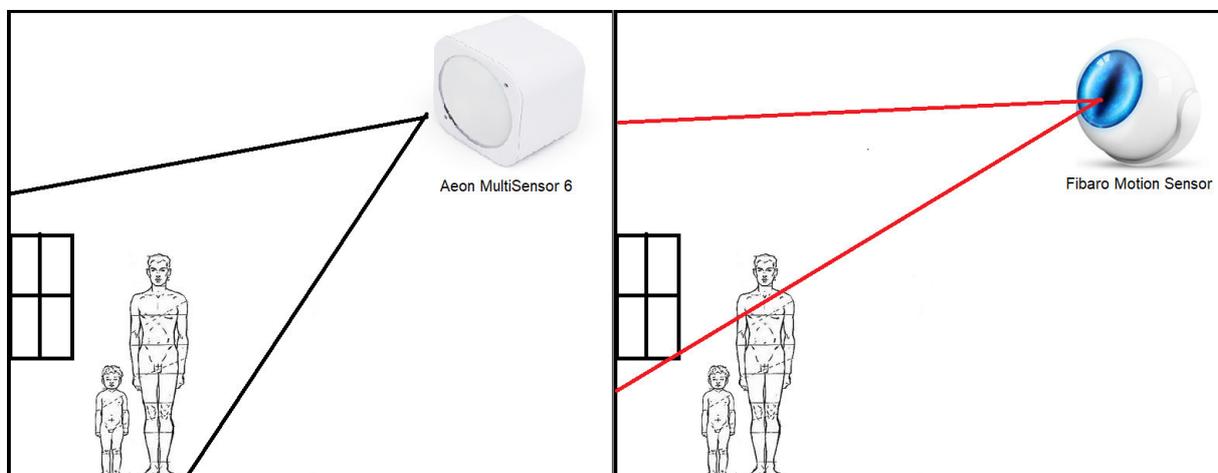


Abbildung 30: Überwachungsbereich Aeon MultiSensor 6 (li.) und Fibaro MultiSensor (re.)

In Abbildung 30 sind die jeweiligen Bereiche ersichtlich, in welchen die Bewegungsmelder eine Bewegung erkennen. Der Aeon MultiSensor 6 ist dabei so eingestellt, dass er nicht nur Kinder, sondern auch Erwachsene im vordefinierten Überwachungsbereich erkennt, während der Fibaro Motion Sensor nur erwachsene Personen erkennt.

Das für den Use Case benötigte Homeegramm kann wahlweise in der Homee App oder in der Homee Weboberfläche erstellt werden. Das Homeegramm hat die Aufgabe, die zwei Bewegungsmelder miteinander zu verknüpfen und zu überprüfen, ob das Fenster geöffnet ist und ob sich ein Kind im zu überwachenden Bereich befindet. Das Homeegramm basiert dabei auf dem if-this-then-that Prinzip. Die Erstellung des benötigten Homeegramms muss wie folgt geschehen, um diese Funktionalitäten zu gewährleisten:

1.) Neues Homeegramm erstellen:

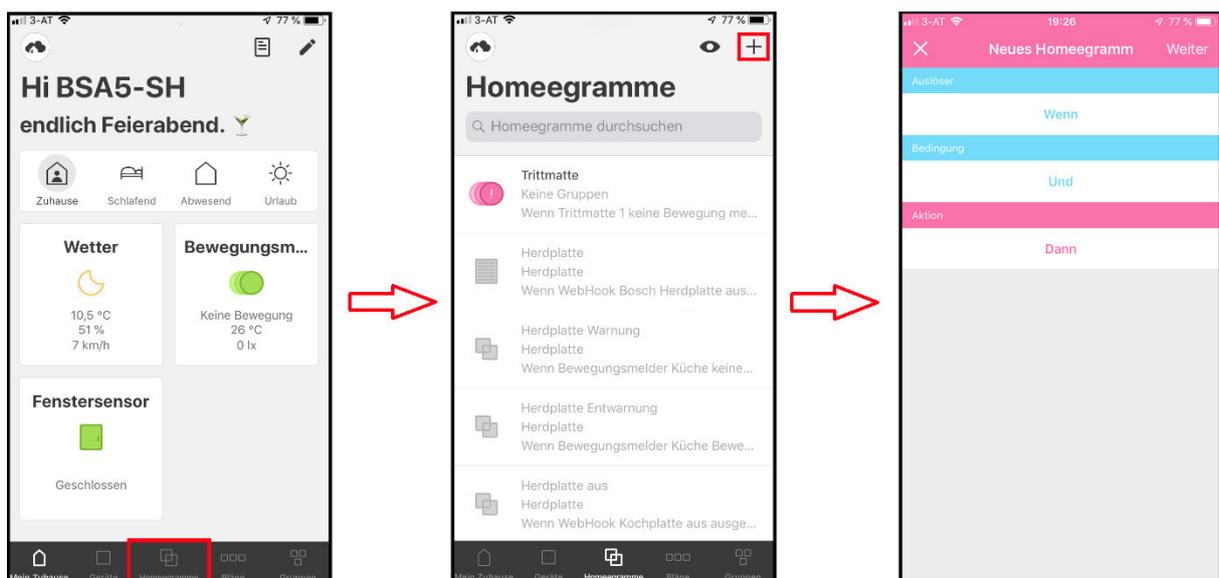


Abbildung 31: Bildanleitung zur Erstellung eines neuen Homeegramms

In Abbildung 31 wird Schritt für Schritt gezeigt, wie ein neues Homeegramm in der IOS App von Homee erstellt werden kann. Zuerst muss dafür der Menüpunkt Homeegramm ausgewählt werden. Auf der nun erscheinenden Seite sind alle schon vorhandenen Homeegramme zu sehen. Um ein neues Homeegramm zu erstellen muss nun das „+“ Icon rechts oben ausgewählt werden. Wurde das „+“ Icon ausgewählt, erscheint eine neue Seite, auf der ein neues Homeegramm erstellt werden kann.

2.) „Wenn“-Bedingung festlegen:

Die „Wenn“-Bedingung ist sozusagen der Auslöser des Homeegramms. In einem Homeegramm kann es mehrere „Wenn“-Bedingungen geben, wobei mindestens eine Bedingung eintreten muss, damit das Homeegramm auslöst. Im aktuellen Kontext wird eine „Wenn“-Bedingung benötigt, welche auf das Auslösen des Aeotec MultiSensor 6 reagiert. Diese „Wenn“- Bedingung wird, wie in Abbildung 32 zu sehen erstellt.

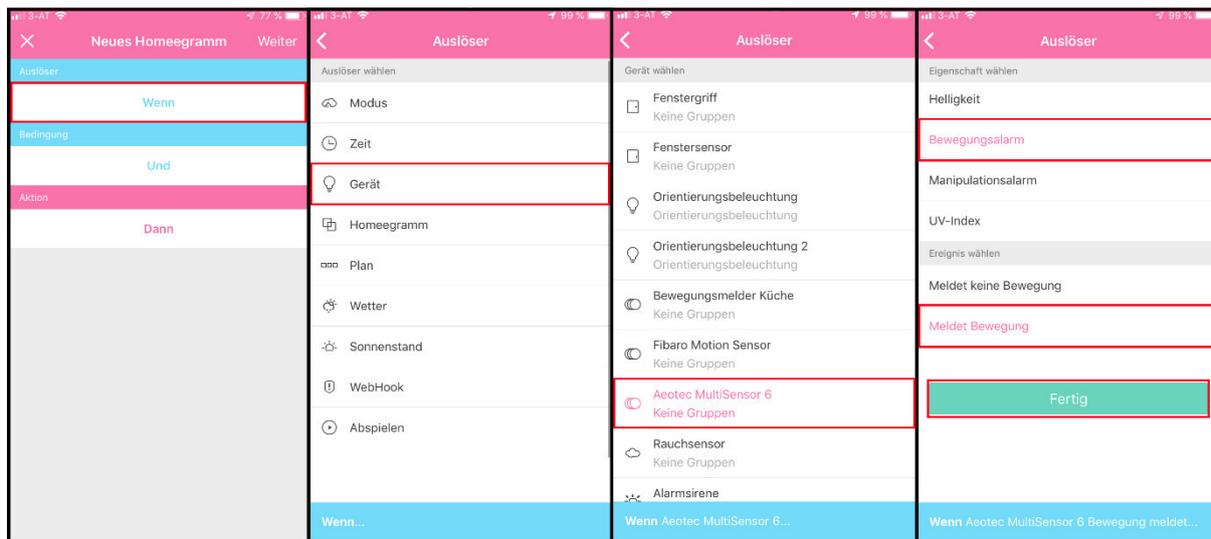


Abbildung 32: Erstellen der benötigten "Wenn"- Bedingung

3.) „Und“-Bedingungen festlegen:

Die „Und“-Bedingungen können verwendet werden, um ein Auslösen des Homeegramms weiter zu spezifizieren. So kann mit den sogenannten „Und“-Bedingungen ein Homeegramm erstellt werden, welches nur dann auslöst, wenn sowohl die „Wenn“-Bedingung als auch die zugehörige „Und“-Bedingung erfüllt ist. Auch im Falle der „Und“-Bedingungen können mehrere Bedingungen erstellt werden. Im Gegensatz zu den „Wenn“-Bedingungen, müssen jedoch alle erstellten „Und“-Bedingungen zusammen mit mindestens einer „Wenn“-Bedingung zutreffen. Im aktuellen Kontext werden zwei „Und“-Bedingungen benötigt. Die erste „Und“-Bedingung wird benötigt, um zu überprüfen, ob sich der Hoppe Fenstergriff, im geöffneten Zustand befindet, also ob das Fenster geöffnet ist. Denn nur wenn ein Fenster auch tatsächlich geöffnet ist, und sich ein Kind dem Fenster nähert, soll ein Alarm ausgesendet werden. Die Erstellung dieser Bedingung ist der Abbildung 33 zu entnehmen.

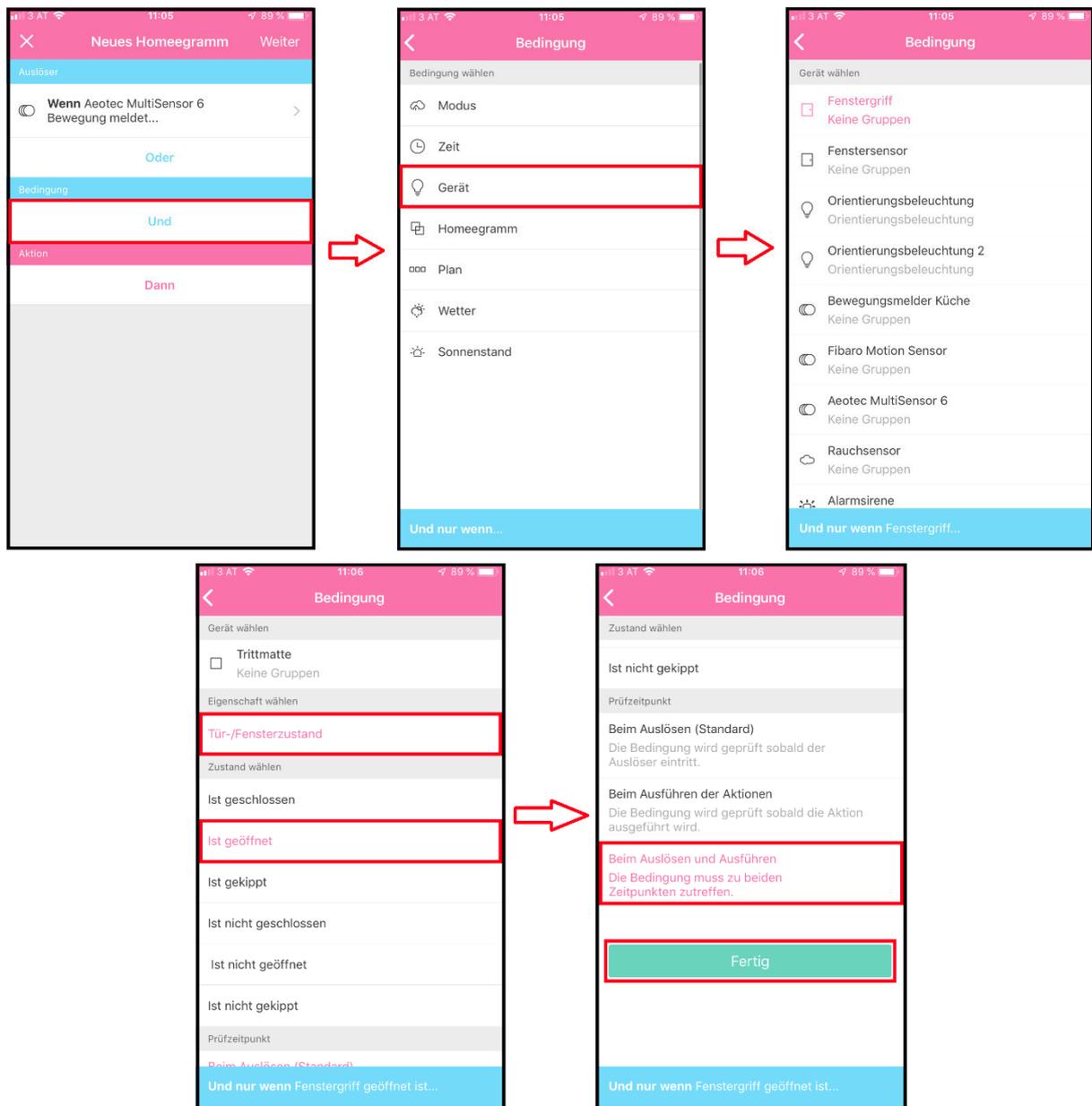


Abbildung 33: Erstellen der Fenstergriffbezogenen "Und"- Bedingung

Die zweite benötigte „Und“-Bedingung bezieht sich auf den Fibaro Motion Sensor. Es soll eine Bedingung erstellt werden, welche abfragt, ob der Fibaro Motion Sensor keine Bewegung erkannt hat. Dies ist, wie auch auf den vorherigen Seiten schon beschrieben, nötig, um zu erkennen, ob es sich bei der erkannten Person um einen Erwachsenen oder ein Kind handelt. Die Erstellung dieser „Und“-Bedingung kann Abbildung 34 entnommen werden.

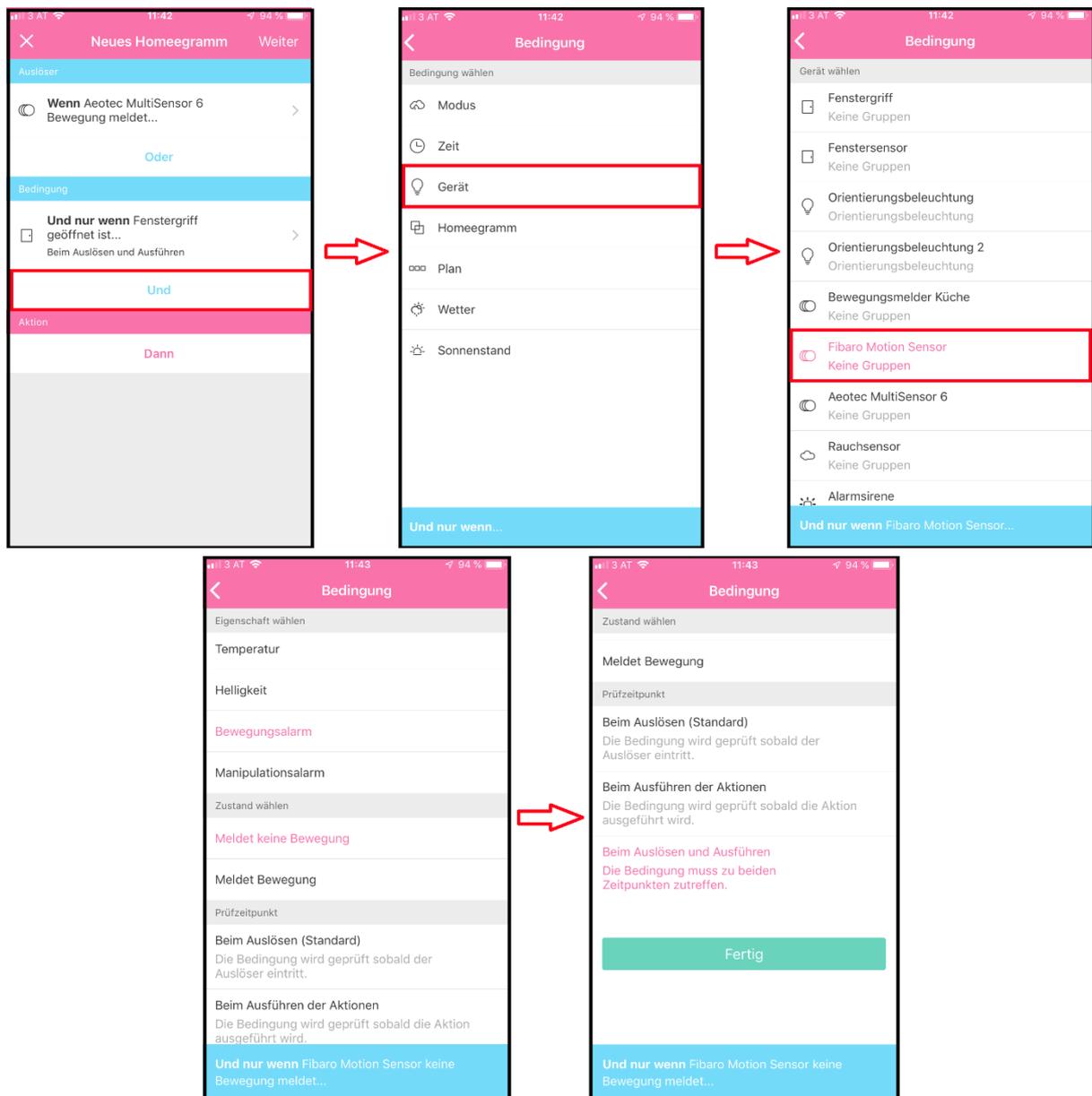


Abbildung 34: Erstellen der Fibaro Motion Sensor -bezogenen "Und"- Bedingung

4.) „Dann“-Ereignisse festlegen:

Die „Dann“- Ereignisse können als Ergebnis eines Homeogramms beschrieben werden. Mit ihnen kann festgelegt werden, was passieren soll, wenn die Abfrage aller Bedingungen, zu einem positiven Ergebnis geführt hat. So kann beispielsweise festgelegt werden, dass eine Nachricht auf alle Geräte, die mit dem Homee Konto verknüpft sind, gesendet wird. Diese Funktion wird auch im aktuellen Use Case genutzt, um die Aufsichtspersonen des zu überwachenden Kinds zu alarmieren. Die Erstellung dieses „Dann“- Ereignisses, kann der Abbildung 35 entnommen werden.

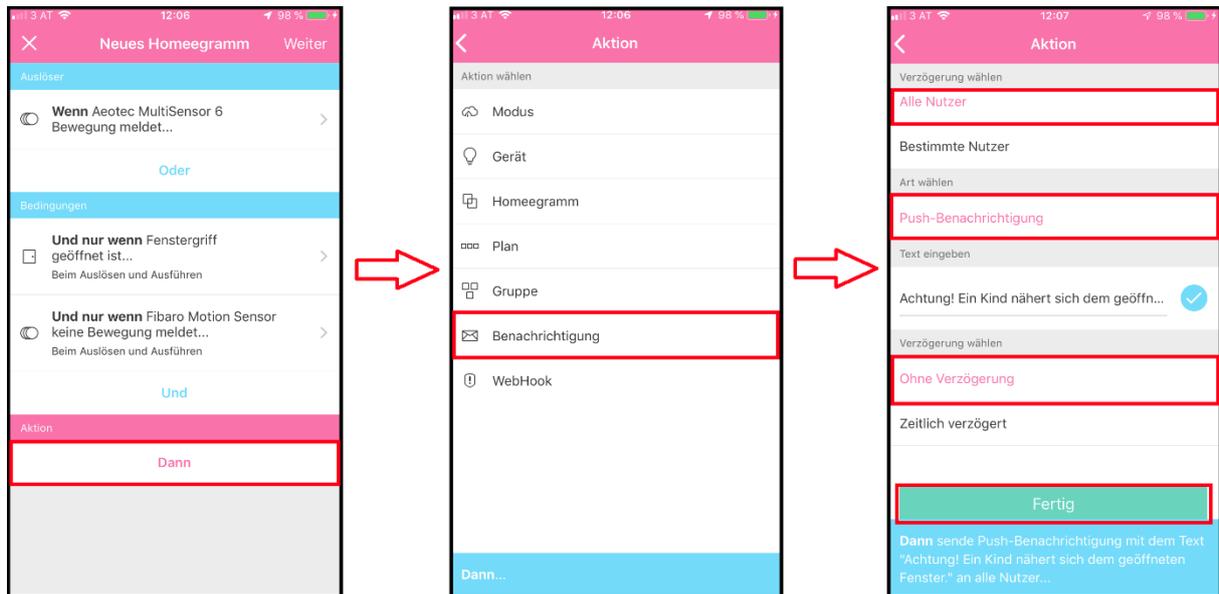


Abbildung 35: Erstellen des Use Case spezifischen "Dann"- Ereignisses

Mit dem Festlegen eines, oder mehrerer „Dann“-Ereignisse, ist das Homeeogram fertig erstellt. Das für diesen Use Case benötigte Homeeogram soll im endgültigen Zustand, wie in Abbildung 36 zu sehen, aussehen.

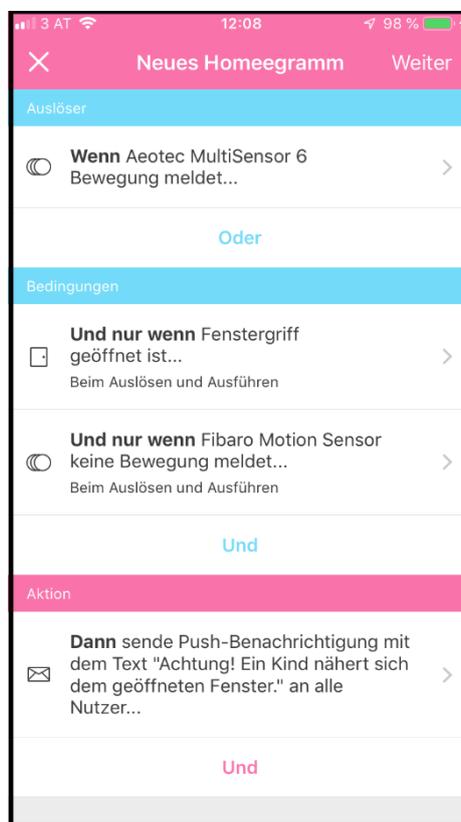


Abbildung 36: Fertiges Homeeogram für den Use Case

6 Evaluierung

Im Zuge der Evaluierung wird ermittelt ob das System stabil läuft und den Anforderungen entsprechend umgesetzt wurde. Die Anforderungen sind dabei im *Kapitel 3.1* und *Kapitel 3.2* näher beschrieben worden.

Evaluierung der Systemstabilität

Die Evaluierung der Stabilität des Systems, also die Untersuchung, ob das System fehlerfrei läuft, wurde in zwei voneinander unabhängigen Testläufen durchgeführt. Der erste Test erfolgte im Zuge der internen Präsentation der Bachelorprojekte am 21.01.2019. An dieser Präsentation nahmen neben den Lektoren aus dem Bereich Smart Home auch Lektoren aus anderen Fachbereichen wie AAL und Webdesign teil. Im Zuge dessen wurden alle Funktionen des Systems präsentiert. Auch das Layout der Homee App wurde präsentiert. Bei diesem Testlauf wurden keinerlei Fehlfunktionen entdeckt.

Der zweite Testlauf fand am 12.02.2019 statt. Hier wurde das System im Zuge der Projektabgabe dem Kuratorium für Verkehrssicherheit präsentiert. In diesem Test wurde gemeinsam mit Herrn Dr. Bauer - Kuratorium für Verkehrssicherheit, getestet ob alle Use Cases sinngerecht umgesetzt wurden und das System fehlerfrei funktioniert. Während diesem Testlauf wurden keinerlei Probleme im System und dessen Ablaufweise entdeckt.

7 Conclusio

Ziel dieser Arbeit war es, einen funktionstüchtigen Aufbau im Kontext von Unfallverhütung in den eigenen vier Wänden mit Bezug auf Smart Home zu schaffen. Dieser Aufbau erfolgte in der neuen Smart Home Testwohnung der FH Technikum Wien. Die Use Cases wurden in Zusammenarbeit mit dem Kuratorium für Verkehrssicherheit definiert. Die Auswahl der Komponenten und der Aufbau wurden von den Bachelor Smart Home und Assistive Technologien Studenten, Manuel Schweiger und Mathias Hazibar, durchgeführt. Im Zuge des Aufbaus wurden verschiedene funkbasierte Geräte sowie ein Smart Home Gateway angeschafft, um die Use Cases praktisch umzusetzen. Die größte Herausforderung stellte der Use Case *Herdabschaltung* dar, da die in der Küche verbaute Home Connect Kochplatte nicht, wie erwartet, über die Home Connect App ein- bzw. ausgeschaltet werden konnte. Das Problem wurde nach einigen Fehlversuchen, mit dem Programm NodeRED, in Verbindung mit bereits vorhandenen KNX Leistungsaktoren der Marke Siemens gelöst. Für diesen Use Case bedeutet dies, dass er mit jedem herkömmlichen Herd funktioniert, welcher über einen 3-phasigen E-Herd Anschluss versorgt wird.

Abschließend kann gesagt werden, dass funkbasierte Smart Home Geräte eine kosteneffiziente und gut skalierbare Möglichkeit bieten, um sich selbst in den eigenen vier Wänden vor Unfällen zu schützen. Abgesehen von den in dieser Arbeit vorgestellten Use Cases, gibt es noch eine Vielzahl anderer vorstellbarer Use Cases, welche im Kontext der Unfallprävention mit Smart Home Geräten umgesetzt werden können.

8 Literaturverzeichnis

- [1] 1&1 IONOS. (10. Januar 2018). *Was ist ein Server - Ein Begriff zwei Definitionen*. (1&1) Abgerufen am 11. Januar 2019 von 1&1 IONOS SE: <https://www.ionos.de/digitalguide/server/knowhow/was-ist-ein-server-ein-begriff-zwei-definitionen/>
- [2] Adewole, A. (Mai 2018). *C# and .NET Core Test-Driven Development*. Birmingham: Packt Publishing Ltd. Abgerufen am 13. Januar 2019 von <https://books.google.at/books?id=kF9dDwAAQBAJ&pg=PA194&dq=webhook&hl=de&sa=X&ved=0ahUKEwiTnKqfxurfAhWyposKHZO-B7AQ6AEIPjAD#v=onepage&q=webhook&f=false>
- [3] Aeotec. (2018). *Multisensor 6*. Abgerufen am 13. Januar 2019 von aeotc.com: <https://aeotec.com/z-wave-sensor>
- [4] Chantelau, K. (2010). *Multimediale Client-Server-Systeme*. Berlin Heidelberg: Springer Verlag.
- [5] Fibaro. (2018). *Fibaro Manuals*. Abgerufen am 15. Januar 2019 von fibaro.com: <https://manuals.fibaro.com/>
- [6] Gövercin, M. (2008, April). *researchgate.net*. Retrieved Jänner 07, 2018, from https://www.researchgate.net/publication/235976803_Visual_fall_detection_system_in_home_environments
- [7] Janssen, D. (15. Juni 2016). *Z-Wave*. Abgerufen am 10. Januar 2019 von reicheltpedia, das Wiki für Elektronik: <https://www.reichelt.de/reicheltpedia/index.php/Z-Wave>
- [8] König, R. (6. November 2018). *Z-Wave*. Abgerufen am 10. Januar 2019 von https://wiki.fhem.de/wiki/Z-Wave#Home-Id_und_Node-ID
- [9] Kumar, T. (17. Dezember 2016). *ZigBee topology: A survey*. Abgerufen am 11. Januar 2019 von IEEE: <https://ieeexplore.ieee.org/document/7987937>
- [10] Minvielle, L. (14. September 2017). *Fall Detection Using Smart Floor Sensor and Supervised Learning*. Abgerufen am 08. Jänner 2019 von <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8037597>
- [11] Morrison, J. P. (2010). *Flow-Based Programming, 2nd Edition: A New Approach to Application Development*. Paramount, CA: CreateSpace.
- [12] Nimi, T. (20. Okt. 2017). *Comparative analysis of ZigBee network with tree and mesh topology for different range of frequencies*. Abgerufen am 11. Januar 2019 von IEEE.org: <https://ieeexplore.ieee.org/document/8321140>

- [13] Node-RED. (2017). *About Node-RED*. Von nodered.org: <https://nodered.org/about/> abgerufen
- [14] OpenZWave Library. (6. Januar 2019). *Introduction to OpenZWave*, 1.8.11. Abgerufen am 11. Januar 2019 von Z-Wave Concepts: <http://www.openzwave.com/dev/>
- [15] Ramaya, C. M. (10. April 2011). *Study on ZigBee technology*. Abgerufen am 11. Januar 2019 von IEEE.org: <https://ieeexplore.ieee.org/document/5942102/authors#authors>
- [16] Schiller, K. (31. 01 2016). *Das Gateway - Definition und praktischer Einsatz im SmartHome*. Abgerufen am 09. Jänner 2019 von <https://www.homeandsmart.de/gateway-definition-praktischer-einsatz-im-smart-home>
- [17] Schweiger, M. (2019). *Unfallprävention in den eigenen vier wänden mit smart home lösungen*. Bachelorarbeit, FH Technikum Wien. Abgerufen am 10. Januar 2019
- [18] Shi, G. (05. Mai 2009). *Mobile Human Airbag System for Fall Protection*. Abgerufen am 07. Jänner 2019 von IEEE SENSORS JOURNAL: <https://www.ieee-nems.org/ftp/papers/ieee-sensors-2009-lhchau.pdf>
- [19] Sommer, K. (kein Datum). *SensFloor: Fußbodensensoren können Stürze verhindern* . Abgerufen am 08. Jänner 2019 von <https://www.smart-wohnen.de/smart-home-senioren/artikel/sensfloor-fussbodensensoren-koennen-stuerze-verhindern/>
- [20] Technopedia. (Unbekannt). *Server Software*. Abgerufen am 12. Januar 2019 von Technopedia.com: <https://www.techopedia.com/definition/23735/server-software>
- [21] Vesternet Ltd. (2012). *Understanding Z-Wave Networks, Nodes & Devices*. Abgerufen am 10. Januar 2019 von <https://www.vesternet.com/resources/technology-indepth/understanding-z-wave-networks/>
- [22] Yassein, M. B. (22. 09 2016). *Smart homes automation using Z-wave protocol*. Abgerufen am 09. Jänner 2019 von IEEE Xplore: <https://ieeexplore.ieee.org/document/7745306>
- [23] Zappi, P. (08. July 2010). *Tracking Motion Direction and Distance With Pyroelectric IR Sensors*. Abgerufen am 11. Januar 2019 von IEEE.org: <https://ieeexplore.ieee.org/document/5503973>

Abbildungsverzeichnis

Abbildung 1: Z-Wave Netzwerk mit Controller (Vesternet Ltd, 2012)	11
Abbildung 2: Maximaler Weg eines Pakets in einem Z-Wave Netzwerk (Vesternet Ltd, 2012)	11
Abbildung 3: Vektor Grafik Sterntopologie, (Kumar, 2016)	12
Abbildung 4: Vektor Grafik Baumtopologie, (Kumar, 2016).....	13
Abbildung 5: Vektor Grafik Mesh Topologie, (Kumar, 2016)	14
Abbildung 6: Schematische Darstellung Smart Home Gateway.....	19
Abbildung 7: Fibaro Motion Sensor (Fibaro, 2018)	23
Abbildung 8: Aeotec MultiSensor 6 (Aeotec, 2018).....	24
Abbildung 9: Ergebnis der Node.js Installationsüberprüfung in CMD	26
Abbildung 10: Node-RED gestartet.....	26
Abbildung 11: Inject Node in Node-RED.....	27
Abbildung 12: Einstellungsmenü der Inject Node.....	27
Abbildung 13: Debug Node in Node-RED.....	28
Abbildung 14: KNX Device Node in Node-RED	28
Abbildung 15: Einstellungsmenü der KNX Device Node	29
Abbildung 16: Function Node in Node-RED.....	30
Abbildung 17: Verwendete Geräte im Use Case Herdabschaltung	32
Abbildung 18: Bewegungsmelder (li., ob.) und zu überwachender Bereich (rot)	33
Abbildung 19: Node-RED Weboberfläche.....	34
Abbildung 20: User Settings in Node-RED	35
Abbildung 21: Node-RED Flow für den Use Case „Automatische Herdabschaltung“	36
Abbildung 22: Unterteilung des Flows in sechs Punkte.....	36
Abbildung 23: Konfiguration GET:nodes (li.) und Homee Node (re.).....	37
Abbildung 24: Abfragen des Bewegungsmelders	38
Abbildung 25: Konfigurationseinstellung Inject Node (li.) und KNX Device Node (re.)	39
Abbildung 26: Auswertung der Strommessung	40
Abbildung 27: AND-Funktion in NodeRed.....	41
Abbildung 28: toString Funktion in NodeRED	42
Abbildung 29: Verwendete Geräte im Use Case Kindersicherung	45
Abbildung 30: Überwachungsbereich Aeon MultiSensor 6 (li.) und Fibaro MultiSensor (re.).....	45
Abbildung 31: Bildanleitung zur Erstellung eines neuen Homeegramms	46
Abbildung 32: Erstellen der benötigten "Wenn"- Bedingung	47
Abbildung 33: Erstellen der Fenstergriffbezogenen "Und"- Bedingung	48
Abbildung 34: Erstellen der Fibaro Motion Sensor -bezogenen "Und"- Bedingung	49
Abbildung 35: Erstellen des Use Case spezifischen "Dann"- Ereignisses.....	50
Abbildung 36: Fertiges Homeegramm für den Use Case.....	50

Anhang A: Bedienungsanleitung

Kurzfassung:

Dieser Anhang, welcher von Schweiger Manuel und Hazibar Mathias erstellt worden ist, dient der Konfiguration bzw. der Bedienung der einzelnen Sensoren und Aktoren des Systems.

Homee:

Allgemein:

Homee ist ein modulares Smart Home Gateway. Das homee System bietet seinen Benutzerinnen und Benutzern die Möglichkeit, das System modular aufzubauen bzw. zu erweitern. Jeder sogenannte „Cube“ erweitert das System um eine spezifische Funktechnologie wie EnOcean, ZigBee oder Z-Wave. Der sogenannte Brain Cube (Basiswürfel - weiß) stellt nicht nur die Stromversorgung für die übrigen Cubes dar, sondern bietet auch die Möglichkeit, WLAN fähige Geräte zu integrieren.

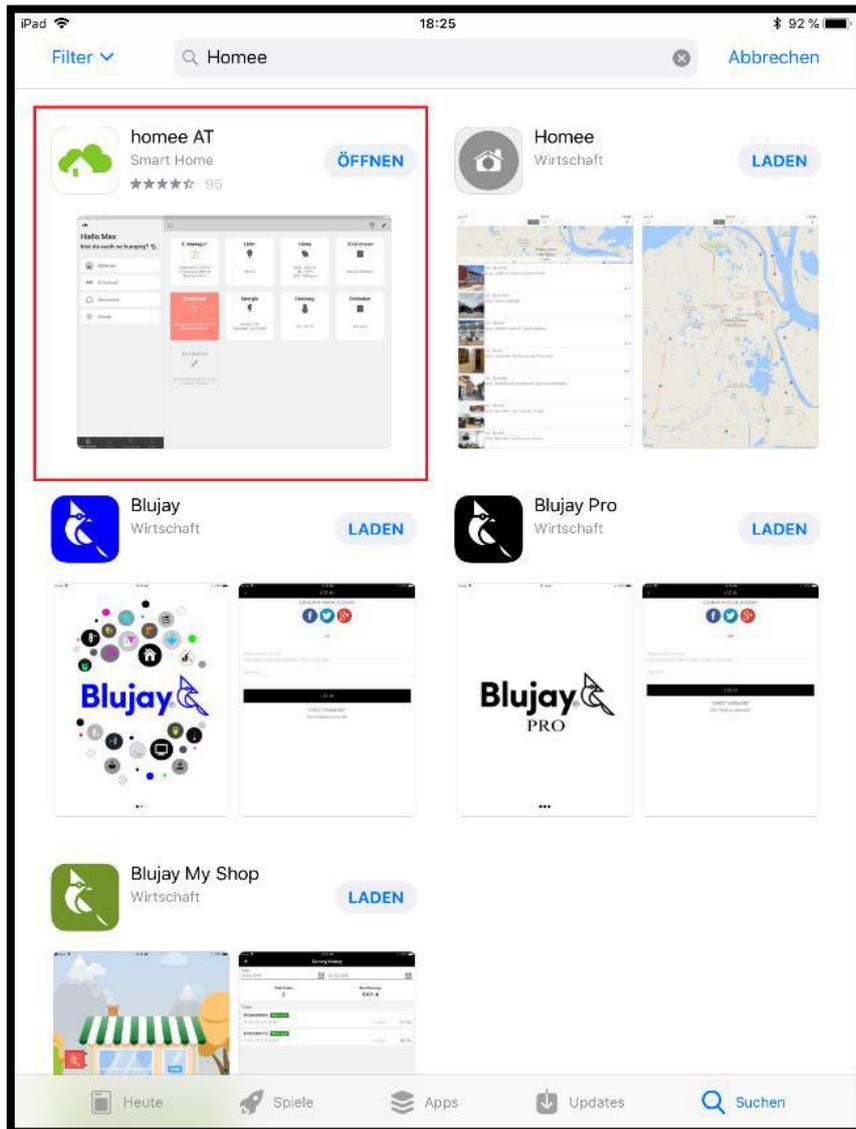
Das eingesetzte Homee System besteht aus den folgenden Elementen:

- Homee Brain Cube, Weiß
- Homee Z-Wave Cube, Grün
- Homee ZigBee Cube, Orange
- Homee EnOcean Cube, Blau

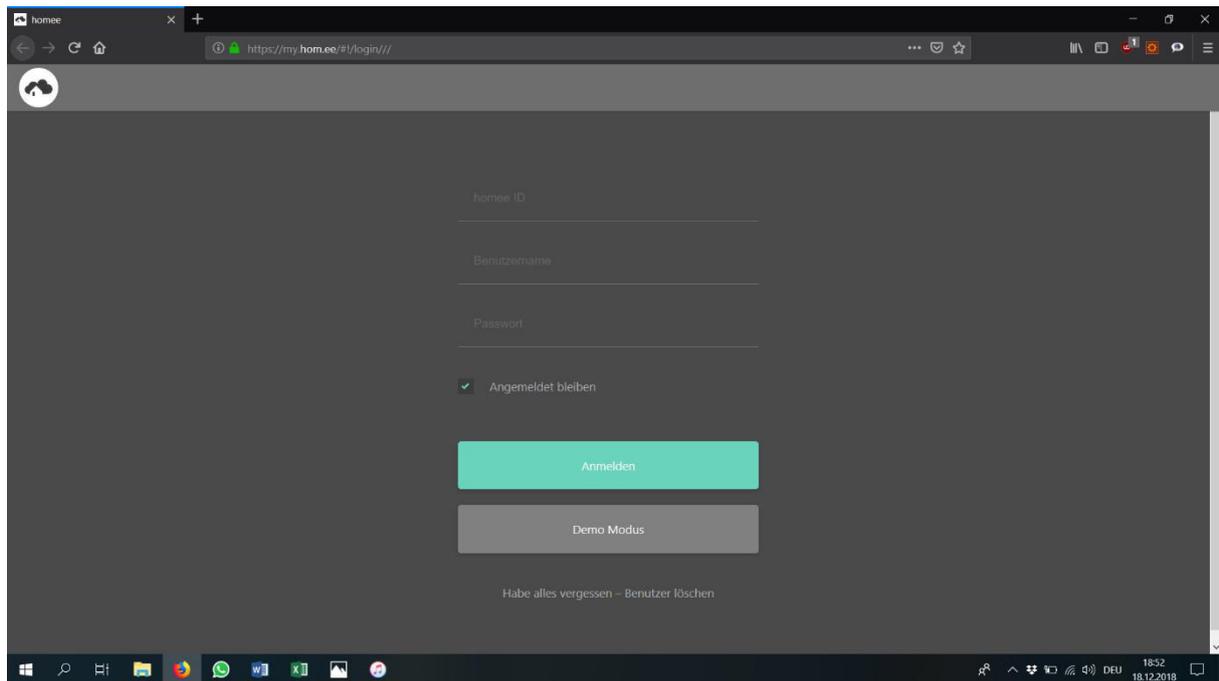
Einrichten des Homee Gateways:

Schritt 1: Erste Schritte

Um eine Verbindung mit Ihrem homee Brain Cube herstellen zu können, müssen Sie im ersten Schritt die „homee AT“ App aus dem App Store laden. Diese App ermöglicht Ihnen eine WLAN Verbindung zu homee aufzubauen und diesen per Smartphone einzurichten.



Falls Ihnen kein Smartphone zur Verfügung steht, gibt es die Möglichkeit, homee über einen PC oder Laptop zu konfigurieren. Besuchen Sie dazu die angegebene Webseite: <https://my.hom.ee/#!/login///>



Schritt 2: Homee anschließen

In diesem Schritt wird das Anschließen des homee Brain Cubes an die Netzspannung beschrieben.

1. Schließen Sie den homee Brain Cube an das mitgelieferte Netzteil an.
2. Schließen Sie das Netzteil an eine herkömmliche 230V/50Hz Steckdose an.
3. Warten Sie, bis die LED an der Rückseite des Brain Cubes orange leuchtet.

Schritt 3: Verbindung zu homee herstellen:

In diesem Schritt wird das Herstellen der Internetverbindung zu homee erläutert. Dies erfordert folgende Schritte:

1. Gehen Sie in die WLAN Einstellungen Ihres iPhones, iPads oder Android Gerät und verbinden Sie sich mit dem Netzwerk „homee“. Warten Sie anschließend, bis die Verbindung mit dem Netzwerk hergestellt ist. Dies kann in manchen Fällen mehrere Minuten in Anspruch nehmen.
2. Öffnen Sie die in *Schritt 1* heruntergeladene „homee“ App. Drücken Sie im erschienen Pop-up Fenster auf den Banner „homee einrichten“.
3. Nun muss dem System die homee-ID mitgeteilt werden. Dies kann entweder manuell durch Eingabe oder automatisch durch das Scannen des QR-Codes auf der Unterseite des Brain Cubes erfolgen.
4. Nachdem die homee-ID erfolgreich eingetragen worden ist, muss nun ein Benutzer erstellt werden. Folgen Sie hierfür den Anweisungen der App. **Merken Sie sich Ihr Passwort!**

Schritt 4: Netzwerkeinstellungen

In diesem Schritt muss entschieden werden, ob man den homee im sogenannten Stand-Alone-Modus oder in einem lokalen WLAN-Netzwerk betreiben will. Beide Betriebsvarianten bieten dabei Vor- und Nachteile.

Im **Stand-Alone-Modus** errichtet der Brain Cube sein eigenes WLAN Netzwerk. Dieses dient jedoch nur dem Aufbau einer Verbindung zwischen Smartphone, Tablet oder PC mit homee und stellt keine Internetverbindung her. Der Vorteil dieses Modus ist, dass keine Verbindung zwischen dem Smart Home und dem Internet besteht. Des Weiteren wird kein WLAN Router benötigt. Wenn homee in diesem Modus betrieben wird, gibt es keine Möglichkeit der Fernsteuerung und die Reichweite ist stark eingeschränkt.

Das Betreiben des homee in einem **lokalen WLAN-Netzwerk** bietet Ihnen den Vorteil, dass Sie von überall auf Ihren homee zugreifen können, da sich dieser dann im Internet befindet. Sie haben die Möglichkeit, von überall aus Ihre Smart Home Geräte zu steuern.

Möchten Sie Ihren homee im herkömmlichen Modus betreiben, sind folgende Schritte zu befolgen:

1. Wählen Sie in der App das lokale Netzwerk, welches Sie für die Anbindung mit homee bevorzugen. Verbinden Sie sich mit dem gewählten Netzwerk, indem Sie das WLAN Passwort eingeben.
2. Nachdem das Passwort richtig eingegeben worden ist, ändert sich die Farbe der LED an der Rückseite des Brain Cubes auf Grün.
3. Folgen Sie nun den Anweisungen der App, um die Netzwerkkonfiguration Ihres homee abzuschließen.

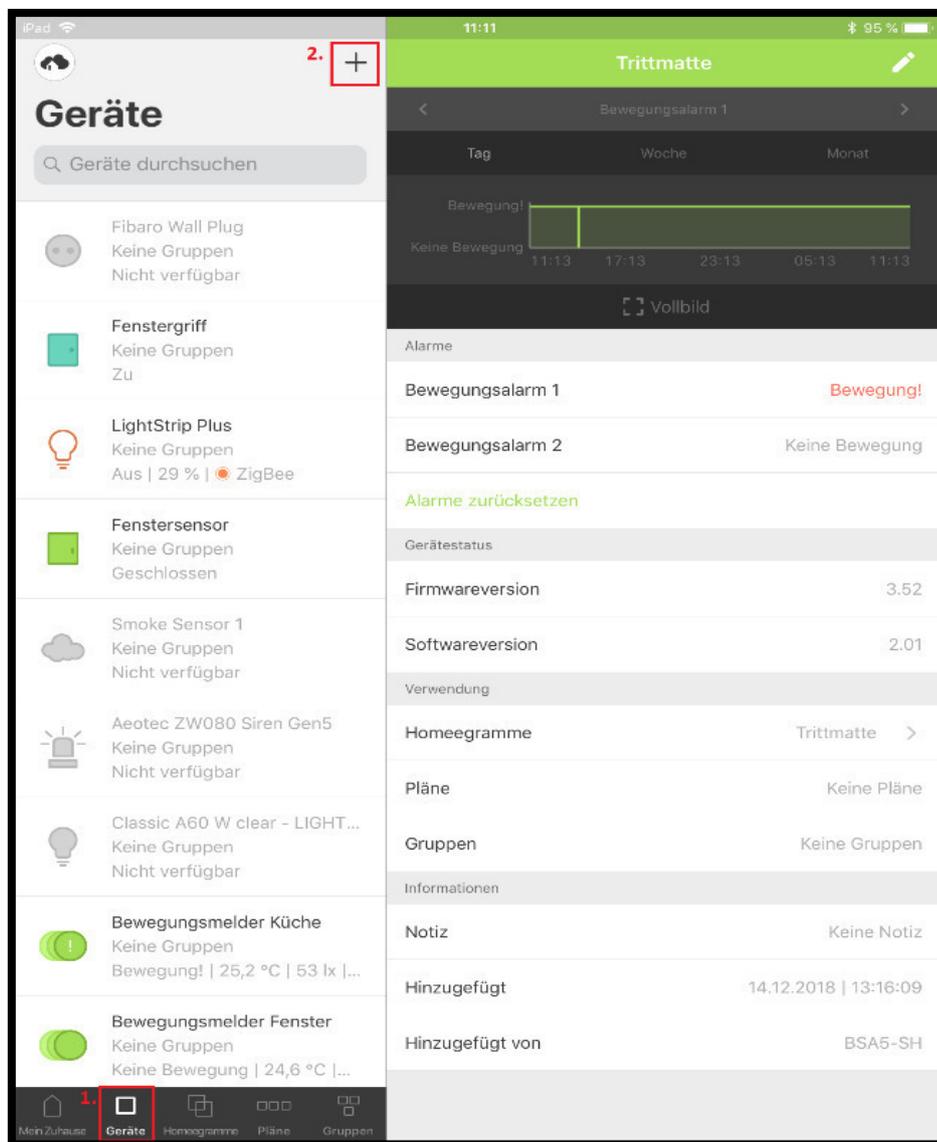
Ihr homee ist nun fertig konfiguriert und kann benutzt werden.

Geräte in das Homee System integrieren

In diesem Kapitel werden die Abläufe erklärt, die notwendig sind um ZigBee, EnOcean, WLAN oder Z-Wave Geräte in das System einzubinden.

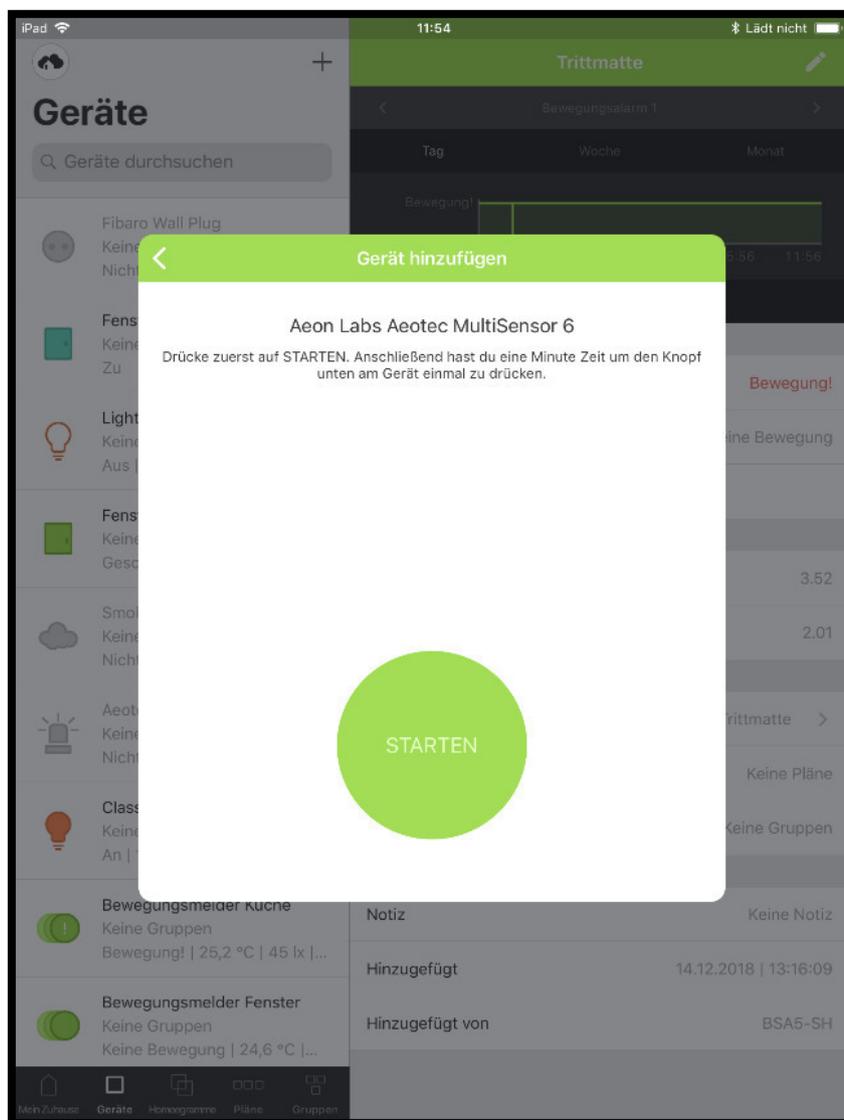
Schritt 1: Voreinstellung

1. Um ein Gerät hinzuzufügen, müssen Sie die homee App oder Weboberfläche öffnen und den Unterpunkt *Geräte* auswählen.
2. Drücken Sie anschließend auf das „+“ Symbol. Sie gelangen nun in das „Gerät hinzufügen“ Menü.



Schritt 2: Hinzufügen eines Gerätes

1. Wählen Sie im Menü die Technologie, die Ihr Gerät unterstützt.
2. Sie gelangen nun in ein neues Untermenü, wo Sie die Geräte einbinden können.
3. Nachdem Sie das Gerät ausgewählt haben, erscheint eine Kurzanleitung, welche zeigt, wie das Gerät hinzugefügt werden kann. Lesen Sie die Anleitung in der Applikation genau und folgen Sie den Anweisungen



Ihr neues Smart Home Gerät wurde nun erfolgreich hinzugefügt.

Sollte das Verbinden mit einem neuen Gerät fehlschlagen, setzen Sie dieses bitte laut gerätespezifischer Herstelleranleitung zurück und probieren Sie es erneut.

Konfiguration des Fibaro Bewegungsmelder

In diesem Kapitel wird die Kalibrierung des **Fibaro Bewegungsmelders** beschrieben, um ihn für das Szenario „Fensterüberwachung“ zu konfigurieren.

1. Öffnen sie die homee App oder die homee Webansicht.
2. Suchen Sie das gewünschte Gerät in der Liste aller Geräte.
3. Öffnen Sie die Einstellungen des Geräts (Stift-Symbol rechts oben).
4. Wählen Sie nun den Menüpunkt „*Expertenmodus*“. Sie haben nun die Möglichkeit, die Parameter des Geräts zu ändern, um gerätespezifische Einstellungen zu verändern. Die allgemeinen Parameter, Bytegrößen und Werte können der herstellereigenen Geräteanleitung entnommen werden.
5. Tragen Sie nun folgende Werte ein:
 - Parameter: 1
 - Bytegröße: 2 Byte
 - Wert: 255

Expertenmodus Speichern

Parameter setzen

Hier kannst du beliebige Parameter an das Gerät senden. Beachte hierzu bitte genau die jeweilige Geräteanleitung. homee sendet die von dir eingegebenen Parameter an das Gerät. Leider bekommen wir kein Feedback ob es funktioniert hat. Falls dein Gerät ein batteriebetriebenes Gerät ist, musst du darauf achten, dass dein Gerät dabei "wach" ist

Parameter 1

Bytegröße 1 Byte 2 Byte 4 Byte

Wert 255

6. Öffnen Sie anschließend den Fibaro Bewegungsmelder. Drücken Sie den B-Button neben der Batterie. Das Gerät synchronisiert anschließend die neuen Parameter.
7. Das Gerät ist nun für den spezifischen Use Case konfiguriert und kann verwendet werden.

Konfiguration Aeotec MultiSensor 6

In diesem Abschnitt wird die Konfiguration des **Aeon MultiSensor 6** erläutert.

1. Öffnen Sie die homee App oder die homee Webansicht.
2. Suchen Sie das gewünschte Gerät in der Liste aller Geräte.
3. Öffnen Sie die Einstellungen des Geräts (Stift-Symbol rechts oben).
4. Wählen Sie nun den Menüpunkt „*Expertenmodus*“. Sie haben nun die Möglichkeit, die Parameter, des Geräts zu ändern, um gerätespezifische Einstellungen zu verändern. Die allgemeinen Parameter, Bytegrößen und Werte können der herstellereigenen Geräteanleitung entnommen werden.
5. Tragen Sie nun folgende Werte ein:
 - Parameter: 4
 - Bytegröße: 1 Byte
 - Wert: 5

Expertenmodus Speichern

Parameter setzen

Hier kannst du beliebige Parameter an das Gerät senden. Beachte hierzu bitte genau die jeweilige Geräteanleitung. homee sendet die von dir eingegebenen Parameter an das Gerät. Leider bekommen wir kein Feedback ob es funktioniert hat. Falls dein Gerät ein batteriebetriebenes Gerät ist, musst du darauf achten, dass dein Gerät dabei "wach" ist

Parameter 4

Bytegröße 1 Byte 2 Byte 4 Byte

Wert 5

6. Öffnen Sie anschließend die Abdeckung des Aeon MultiSensor 6. Drücken Sie den B-Button neben der Batterie. Das Gerät synchronisiert anschließend die neuen Parameter.
7. Das Gerät ist nun für den spezifischen Use Case konfiguriert und kann verwendet werden.

