CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Home Automation for Existing Devices

Automating Non-Smart Devices Using Bluetooth Mesh Technology Controlled by Android Smartphones

Bachelor's of Science Thesis in Computer Science and Engineering

MARKUS ANDERSSON
ANDERS BASTOS
SEBASTIAN BLOMBERG
EMMA NYBORG
ALBIN PÅLSSON
PHILIP THAM

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2015

Home Automation for Existing Devices

Markus Andersson
Anders Bastos
Sebastian Blomberg
Emma Nyborg
Albin Pålsson
Philip Tham

Examiner: Arne Linde

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000
Department of Computer Science and Engineering
Göteborg, Sweden June 2015

Abstract

A growing topic in modern society is the concept of smart homes, which consist of smart automated devices, such as home appliances, that interact with each other and users. These devices could, however, be quite expensive, and automating a home could thus become an economical setback. This Bachelor's thesis examines how a home-automation system could be developed to automate existing devices, providing them with smart technology, in a more cost effective way rather than replacing them entirely.

The resulting home-automation system consists of adapters that enable existing non-smart devices to be automated. Adapters have the ability to control power supply to an existing device and interact with each other through a Bluetooth mesh network, allowing for an extensible system. A user can control these adapters with an Android application, either directly connected to the network of adapters, or through the Internet with the help of a gateway installed in the network. Furthermore, the application provides several utilities for managing the home-automation system and increasing user convenience.

Several features of the resulting home-automation system were evaluated by test groups, and considered to be an improvement in their homes. Other features were considered to be of less importance and possibly causing inconvenience. The overall assessment of the system was considered to be positive. However, there is room for improvement.

Sammandrag

I dagens samhälle är smarta hem ett växande ämne. Konceptet med smarta hem går ut på att kunna styra över apparater som kan kommunicera med varandra, exempelvis smarta temperaturregulatorer eller fjärrstyrbara lampor. Dessa apparater kan dock vara kostsamma, både i inköp och installation, vilket gör att många drar sig för att automatisera sina hem. Denna rapport undersöker hur ett hemstyrningssystem, på ett kostnadseffektivt sätt, kan utvecklas för att automatisera befintliga apparater utan att behöva ersätta dem.

Det utvecklade hemstyrningssystemet består av adaptrar som ger existerande icke-smarta apparater möjlighet att autamatiseras. Adaptrarna har möjlighet att styra strömförsörjningen till en existerande apparat, samt interagera med varandra över ett Bluetooth mesh nätverk, vilket medför ett expanderbart system, där användaren kan utöka antal adaptrar i systemet efter behov. Användaren kan styra dessa adaptrar med en Android-applikation, antingen direkt ansluten till nätverket, eller genom Internet via en gateway som då installerats i hemmet. Applikationen tillhandahåller ett antal verktyg för att hantera hemstyrningssystemet, vilka gynnar bekvämligheten för användaren.

Utvärderingar, gjorda i testgrupper, visade att flera egenskaper i det resulterande hemstyrningssystemet bidrog till förbättringar i användarnas hem. Andra egenskaper visade sig vara mindre betydelsefulla och bidrog i vissa fall även till besvär för användaren. Den resulterande bedömning av systemet visade ett överlag positivt intryck, dock finns det utrymme för förbättringar.

Acknowledgment

We would like to thank Lennart Hansson for his guidance, expertise and time spent on this project. We would also like to express gratitude towards those who participated in our survey and tested our prototype.

Contents

List of Abbreviations

AC Active Current

ACK Acknowledgement

AI Artificial Intelligence

AP Access Point

API Application Programming Interface

BLE Bluetooth Low Energy

DC Direct Current

GAP Generic Access Profile

GATT Generic Attribute Protocol

GUI Graphical User Interface

HTTP HyperText Transfer Protocol

IDE Integrated Development Environment

IoT Internet Of Things, a term for everyday devices having an embedded computer with the ability to represent itself and communicate intelligently with others in a network

JSON JavaScript Object Notation

LED Light-Emitting Diode

MANET Mobile Ad Hoc Network

MVC Model, View, Controller. A design pattern for developing software with user interface

Opcode Operation Code

ORM Object-relational Mapping

OSI Open Systems Interconnection

REST Representational State Transfer

RSSI Received Signal Strength Indication

SBC Single Board Computer

SBMN Simple BLE Mesh Network

SOAP Simple Object Access Protocol

SSL Secure Socket Layer

UI User Interface

URI Uniform Resource Identifiers

URL Uniform Resource Locator

WSDL Web Service Description Language

XML Extensible Markup Language

1

Introduction

Home automation is the concept of controlling and automating the use of home appliances and other electrical equipment, such as light bulbs, temperature gauges or door locks. The topic of home automation is nothing new. In fact, computer based systems were developed already in the 1960s [1]; however, there has not been any wide scale breakthrough until recently. According to Berg Insight, a market research firm based in Sweden, there were approximately 2.9 million smart homes [2], i.e. homes with automation technology controlled by a computer, in the U.S. 2012. Furthermore, the installed base is expected to reach 31.4 million by 2017 [2], potentially increasing economical and technical interests.

The concept Internet Of Things (IoT) can be closely tied together with home automation. IoT devices, such as smart thermometers, can be controlled by, for example, a smartphone and thus possibly providing worldwide range through the Internet. These devices together with the rising popularity of the smartphone [3] account for one of the reasons to the increase in home automation. Home automation with IoT devices provides great convenience and means of optimizing energy consumption by, for example, enabling automatic energy saving or presenting power consumption data to a user in real time. However, it could become expensive if conventional devices were to be replaced by their IoT equivalents. Therefore, when converting to a smart house, a major issue is making all non-IoT devices compatible with a home-automation system [4]. Gill et al. [5] identify the intrusiveness of installations as one of the areas that have hindered consumer adoption of home automation technologies. Consequently, converting non-IoT devices should be done in a manner such that a user would easily be able to install a system, without the need to make hardware changes in the house.

IoT devices also have their drawbacks, mainly in the way they communicate with each other. According to Gill et al. [5] the interoperability between home automation technologies, such as the vast use of different communication protocols, is one of the major factors impeding consumer adoption of home-automation system. Therefore, in an ideal home-automation system, the communication protocols used should preferably be those supported by common consumer devices, in order to provide maximum compatibility.

This section further explains the purpose of creating a concept for automating existing home appliances and electrical devices.

1.1 Purpose

The purpose of this project is to develop a concept and a prototype of a smart power control system that allows for scalable home automation and can be used to operate existing home appliances and electrical equipment. The concept aims to increase the convenience of controlling home appliances for a broad audience, by being simple to install, maintain, and use.

1.2 Scope

This project aims to create a concept for a home-automation system with the use of existing technologies. Control of a vast area of existing devices should be taken into consideration; however, as a limitation of the prototype, only functionality to control power supply to existing devices will be implemented. In order for users to control the system, this project will develop an Android application; however, aesthetics of this application will not be prioritized, as functionality is considered of higher importance. Furthermore, a home-automation system could involve features such as artificial intelligence (AI) or power measuring capabilities, but this is beyond the scope of this project. However, this project will attempt to implement simple automation functionality that could be beneficial to the user. Security aspects are also taken into consideration, but will not be implemented due to time restrictions.

1.3 Method

In order to make the purpose applicable and practicable to implement, specifics regarding what should be done needed to be gathered. Therefore, a preliminary survey was conducted with the intent of gaining perspective of the current home automation needs. The participants were asked questions regarding previous knowledge of home automation-systems and to give feedback on desired system features.

Based on the result of the preliminary survey, research was conducted on available technologies and utilization of these. Development was logically divided into several sub-areas, which were relatively independent of each other. Each project group member was assigned to mainly work on and be responsible for one such sub-area. This allowed different group members to work in parallel in a structured way.

When the project group believed the product to be in a sufficient state, user tests were conducted to evaluate the result of the project. Each participant was handed a number of prototype units, as well as the Android application. When the test period was over, the participants evaluated the system by answering a set of questions in the form of a survey.

1.4 Outline

The development of this project was divided into five fields: Hardware, Network, Android, Cloud Server and Gateway. This division is reflected in the thesis accordingly and illustrated in figure 1. Each section in this thesis describes the methods and theory, as well as the resulting research, conclusions and discussions of each field. Moreover, these sections are ordered in a bottom-up perspective, presenting the most fundamental fields first.

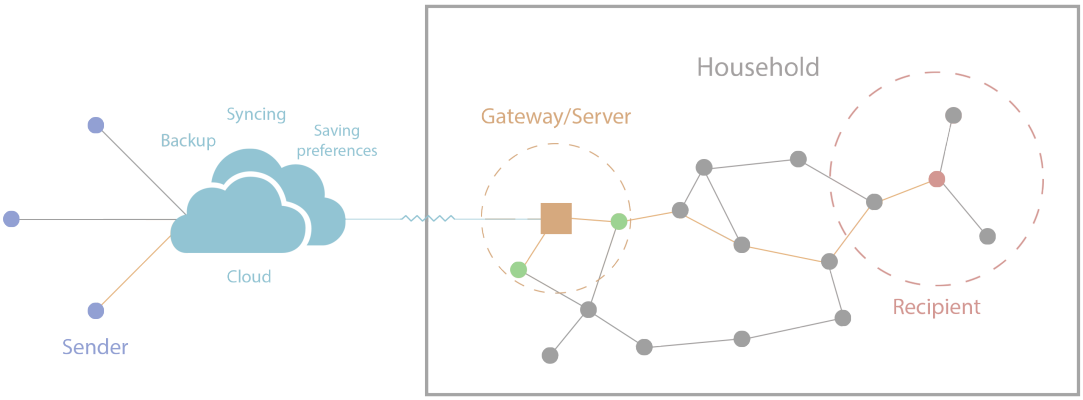


Figure 1: Illustration of the different sections of the project. The hardware (circles in the household), the network (connecting the circles), the Android application (the sender), the cloud server, and the gateway.

The Hardware chapter describes the methods and results of creating hardware able to control existing devices. Furthermore, in the Network section, the communication protocol used to connect these adapters is presented and discussed. The Android Application chapter presents the interface to which the user can control the system. Furthermore, the Cloud Server chapter introduces methods for backing up and synchronizing information used in the application. The Gateway chapter describes opportunities for connection between an end-user at a distant location and the home-automation system. Finally, the User Tests chapter presents the results and discussion of the evaluation of the system.

2

Preliminary Survey

To gain knowledge of a consumer's perspective of current home-automation systems and the potential needs of future systems, a preliminary survey was conducted. Statistics Sweden, a government-managed administrative agency, reports that 50% of the Swedish population live in proprietary houses, 15% in condominiums, and 25% in rental apartments [6]. The survey was distributed to twenty persons of different gender and age of the product's target audience. The participants were chosen by type of housing to reflect the housing type distribution of Swedish citizens.

Four topics received focus in the survey: user friendliness, scalability, accessibility, and convenience. Participants were asked to answer whether they had any previous home-automation system, and if so their experience with these systems. Moreover, participants were asked to give feedback on potential desires for system features, with regards to the four topics mentioned above.

2.1 Results

With regards to accessibility and scalability, the survey showed that all the participants estimate the average distance between their home appliances to be less than 10 meters. Furthermore, participants living in houses estimated that the longest distance between them and any device in their homes was less than 20 meters. The corresponding distance, answered by participants living in apartments, is estimated to be between 1 and 10 meters. Additionally, controlling devices from a distant location (outside of the property area), proved to be a desired feature amongst 50% of the participants of the survey. 30% expressed desire to have the ability of scheduling events, 10% expressed desire to have environmental functionalities such as energy saving based on the brightness of daylight, or monitoring power usage, as illustrated in figure 2.

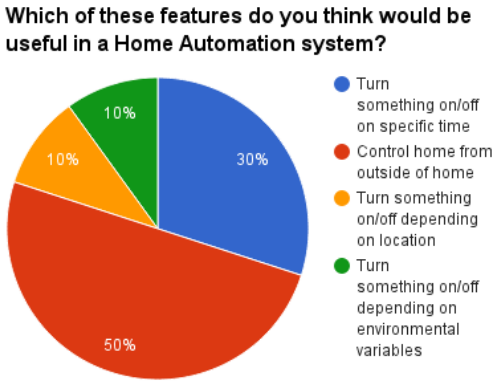


Figure 2: Graph showing the features desired by the survey participants.

In reference to convenience and user friendliness, the survey revealed that the main reasons for not acquiring a home-automation system are: expenses (43%), complexity of installing and using it (29%), and lack of usefulness (14%). Moreover, 23% of the participants expressed that the most inconvenient factor of having a home-automation system is the often required excessive devices, such as remote controls. Almost 40% of the participants also found the expansion of current systems problematic.

2.2 Discussion and Conclusion

The survey revealed that one of the most prominent flaws in existing home-automation systems is the inconvenience of having to use excessive devices. Furthermore, these devices, such as remote controls, may be inaccessible at times. A solution for this could be to integrate controlling software into a smartphone, as the majority of the inhabitants in Sweden own such a device [7] and is likely to carry it with them at most times. Therefore, our system will be controlled via smartphones. An application will be developed for Android as it is currently the most used smartphone operating system [8]. Developing an application allows for greater flexibility in dynamically presenting system components, as compared to a fixed layout of a hardware remote control.

To satisfy the interest of controlling a home-automation system from distant locations, remote access to the system through the Internet will be implemented. This allows for greater accessibility and convenience. The Android application must allow seamless integration of both on-site and off-site controlling, through the use of a gateway that bridges the system functionality to the Internet. However, as scalability is of importance, this gateway should be optional, i.e. the system should also work without a gateway, but in a more limited capacity.

Furthermore, the survey showed that the range between a user and any device could exceed 20 meters, meaning that the network connecting automated devices should be available to the user and exceed 20 meter in range in any direction from the user.

Although participants expressed interest in environmental functionalities, implementing such features is not prioritized in this project, as the main purpose is to develop a functioning prototype. However, all components will be developed with consideration of such features for potential future implementation.

With these results in mind, this project will examine how to develop a home-automation system that effectively fulfills the four criteria: user friendliness, scalability, accessibility and convenience. The developed prototype should be convenient and easy to use, as well as add useful functionality to the everyday life of consumers. It is also of high importance that the system is designed in such matter that new devices can easily be added and remain connectable. Furthermore, all devices should be able to be controlled by the user's smartphone in a radius of 20 meters. Lastly, the user needs to be able to control the system remotely, over the Internet.

3

Hardware

The most fundamental part of a home-automation system is the hardware of the automated devices. As the purpose of this project is to enable IoT functionality for non-smart existing devices, hardware that connects existing devices with the system was developed. There are many ways to enable IoT functionality for existing devices, for example building hardware that regulates temperature, opens curtains or measures power consumption. To limit the scope of the project a limitation was enforced to only build hardware, with networking capabilities, that enables control of power supply to existing devices, such as lamps or coffeemakers. During this project, two types of power controllers, a power switch and a dimmer, were designed and constructed. From here on out, these power controllers are referred to as adapters. This section describes different methods for controlling high voltage current with hardware solutions, and the results and discussion of the outcome.

3.1 Controlling High Voltage with Low Voltage

Most electronic consumer devices, if connected to a power grid, run on high voltage alternating current (AC). According to Worldstandards.eu [9], the majority of national power grids in the world supply 220-240 V with a frequency of either 50 or 60 Hz. To control high voltage AC through the use of low level direct current (DC), a power switch component utilizing galvanic isolation between the two currents is required. With such a component, a microcontroller, which uses low level DC, can control high voltage AC. Traditionally, an electromechanical relay is used for this purpose. An electromechanical relay is a power switch toggled by an electromagnet, driven by low voltage current. Furthermore, electromagnetic relays can be divided into two categories: latching and non-latching. Non-latching relays requires a low current, driving the electromagnet, to keep its high voltage circuit closed [10]. As soon as the low current disappears, the non-latching relay breaks its circuit. In contrast, latching relays require only a quick power pulse to close its circuit. Once closed the latching relay will break its circuit only if another power pulse is received, hence it does not require a constant current to keep its circuit closed. As a consequence, latching relays are widely used in applications where low power consumption is prioritized [10]. In comparison to non-latching relays, latching relays tend to be quite expensive.

In addition to relays, there are other components that can be used to control high voltage current. A triac is a bidirectional thyristor that allows a high voltage AC to be controlled by lower AC [11, p. 498]. Compared to a mechanical relay, a triac does not have any mechanical parts and is therefore able to switch faster. To achieve galvanic isolation, an optically isolated (Optocoupler) triac driver can be used. By running a low level DC through the optocoupler, a light-emitting diode (LED) produces light which toggles an isolated light sensitive transistor or thyristor to let a high voltage current through to the triac [12].

3.2 Controlling the Brightness of a Lamp

Sometimes it is desirable to control the amount of power going to a device, for example to control the brightness of a lamp, rather than just turning the power on or off. A common method to do so is Phase-Control, in which the sinusoidal wave form of an alternating current is partially removed or cut off [13, p. 3], as illustrated in figure 3. Removing parts of the sinusoidal wave reduces the voltage root-mean-square V_{rms} of the waveform, which makes the power output lower. The smaller the area under the sinusoidal wave, the lower V_{rms}. Phase-Control can be achieved by using a triac, which is able to toggle 100 times per second required by an AC at 50 Hz.

For the power output to remain stable, cutting needs to take place at the same time for each wave. Therefore, a point of reference is needed. The natural point of reference is the zero crossing point of the sinusoidal wave and cutting from the it is called leading edge cutting [13], illustrated in figure 3. Leading edge cutting allows for the controlling of resistive loads [14], such as light bulbs, and inductive loads, such as electrical engines. Another way of cutting the sinusoidal wave is trailing edge cutting, illustrated in figure 3. Compared to leading edge cutting, trailing edge cutting cuts the latter part of a half period. Trailing edge allows for resistive loads [14] and capacitive loads, such as electrical transformers. When dimming incandescent light bulbs or halogen lamps, both approaches work, as they can both control resistive loads. However, in the case of dimmable LED lamps the method which should be used depends on the specification of the LED lamps [13]. Some LED lamps only allows for leading edge cutting, and others only for falling edge cutting. There does not seem to be a clear standard.

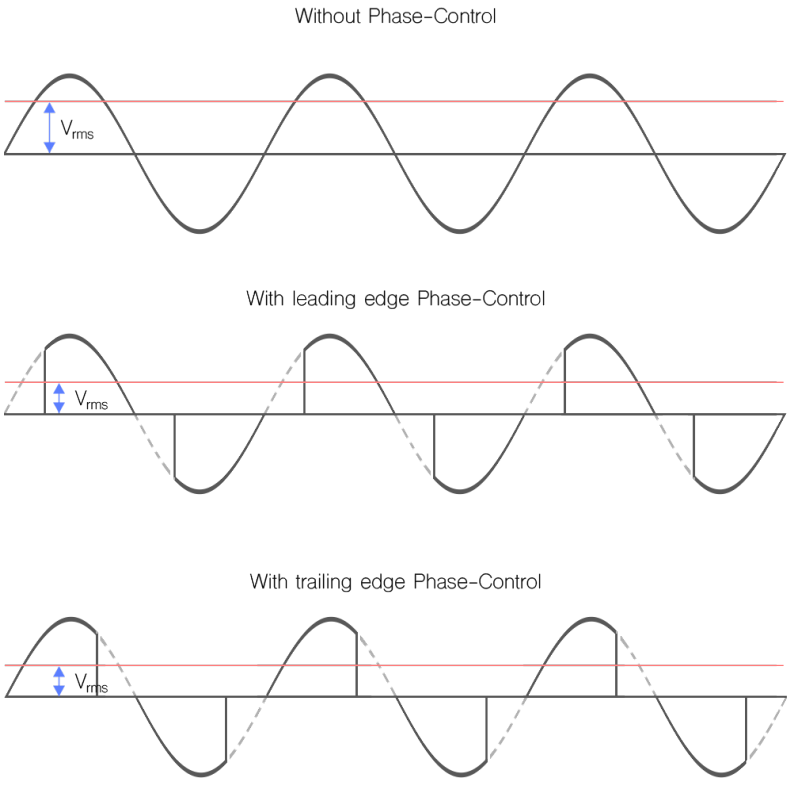


Figure 3: Illustration of an unmodified AC sinusoidal wave (top) and an AC that has been modified using leading edge Phase-Control (middle), and an AC that has been modified using trailing edge Phase-Control (bottom). As illustrated, Phase-Controll reduces the V_{rms}.

3.3 Results

The simplest form of controlling power to an existing device is by a power switch. Our design of the power switch adapter developed in this project uses a timer casing, as seen in figure 4a. This allows a user to plug the adapter into a power socket, and a device directly into the socket of the timer casing. Furthermore, our design includes an electronic circuit, which consists of a non-latching relay to toggle the high voltage AC, a transistor to toggle the relay, an AC/DC transformer to power the circuit, as well as other components specified in Appendix D. The power switch circuit is built to be operated by a microcontroller, with one of its pins connected to the transistor, thus enabling control of the relay.

Similar to the power switch adapter developed in this project, our dimmer adapter design also uses a timer casing, as can be seen in figure 4b. However, instead of using a relay, the dimmer adapter uses a triac to perform leading edge Phase-Control, thus controlling the amount of power to the connected device. The dimmer circuit is based on diy_bloke's reference design [15] and built primarily to control the brightness of a lamp (for further reference see appendix H). To perform phase control, the dimmer adapter circuit includes a zero crossing mechanism which toggles the pin of a dedicated microprocessor, Atmel's ATtiny85. Our program running on the ATtiny85 turns on the triac, via a triac driver, on every zero crossing indication. Based on a brightness level $B \in [0.255]$, the program then turns off the triac after $\frac{B \cdot d}{255}$ seconds, where d is the duration of half a wavelength. The value of d is static and needs to be changed according to the frequency of the power grid the dimmer adapter is to be used with. The program running on the ATtiny85 allows communication with other microcontrollers using UART, a well established protocol for serial communication [16]. Through UART, other microcontrollers can transmit a new brightness value b which the program running on the ATtiny85 will accept and obey.

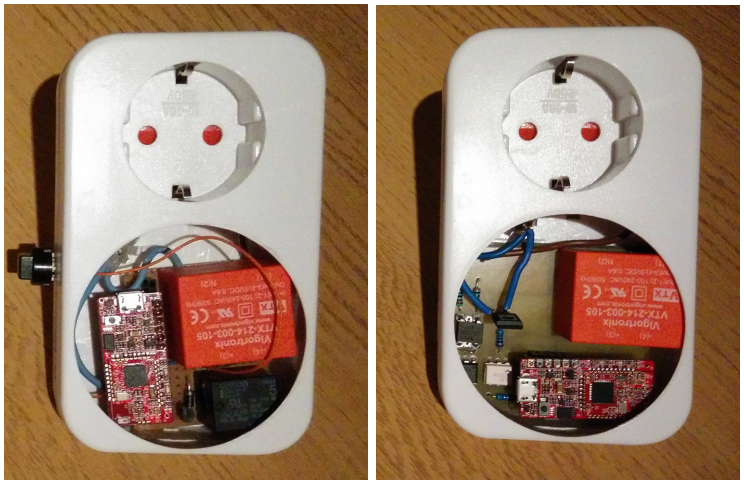


Figure 4: The two types of adapters designed for the project. The left (a) illustrates a power switch adapter. The right (b) illustrates a dimmer adapter.

3.4 Discussion

The limitation of only building adapters for controlling high-voltage power supply is due to two reasons. Firstly, the adapters are applicable to most types of existing devices and thus allow control of multiple devices by simply regulating power supply. Secondly, in consideration of the time limitations of this project, the adapters were simple to construct. Furthermore, the dimmer was developed only to be used with lamps, as a study performed

by the Lightning Research Center [17, p. 2] shows an energy saving of 6% from manual dimming. Thus by enabling dimming remotely, the system could potentially reduce the power consumption of a home.

Our power switch adapter design uses a non-latching relay, as opposed to a latching relay. The main reason for this is the increased cost a latching relay would have contributed to. However, a latching relay would have been preferable to a non-latching relay in an environmental aspect, as it only consumes a fraction of the power of a non-latching relay. Furthermore, we did not consider using a triac instead of a relay in the power switch adapter. A triac could potentially minimize the design, although we have not investigated any potential side effects this may cause.

While using a timer casing provides easy means of encasing a circuit, it also leaves much unused space. Our adapter designs could therefore be minimized, allowing for better use in, for example, power strips. Furthermore, in an ethical aspect, our design does not fully isolate high voltage AC from a user, since the circuit is fully exposed. This, of course, needs to be addressed if the prototypes were to be put into production. Moreover, the form factor of a timer casing is not ideal in all cases. One might imagine an adapter that fits right into a light bulb socket instead of a standard wall socket, thus enabling easier installations in ceiling lamps.

Our dimmer adapter design uses leading edge Phase-Control to dim a light bulb. As the dimmer adapter is intended to be used to control the brightness of a lamp, and both leading edge and falling edge Phase-Control can be used to so, it would have worked with falling edge Phase-Control as well. Leading edge Phase-Control is however able to control inductive loads, thus allowing users to control the speed of electric motors in devices such as a fan. Controlling inductive loads may be useful in a home-automation system, however our dimmer design would require minor changes to allow for this, as it was designed only to be used with resistive loads, such as lamps.

As opposed to changes in hardware, the dimmer functionality could be improved by a simple software change. As described in the result section, the duration value d must be statically programmed into the microprocessor. To avoid this, an initialization routine could be written, logging the time between zero crossings for 1 second. The resulting logs can then be used to automatically calculate the frequency of the national power grid.

4

Network

A fundamental part of a home-automation system is the network that connects the automated devices with each other, as well as the user. This can be done in many different ways by using different structures and communication protocols. This chapter will examine how a network can be designed and implemented in order to achieve desired features, and the result.

4.1 Network topologies

The first matter to consider is how the devices connect to each other and how to control them. Every network can be said to have a topology. Some of the most common network topologies are the Star and Mesh-topologies [18, p. 10]. In a Star topology every node is connected to, and communicates through, a central hub, which relays messages to their destinations, as illustrated by figure 5a. In a Mesh topology every node is directly connected to every other node, enabling direct communication. Furthermore, there is a variation of the Mesh topology, known as the Hybrid Mesh topology, in which there exists a path between every node. However, not every node is connected with every other else, as illustrated by figure 5b.

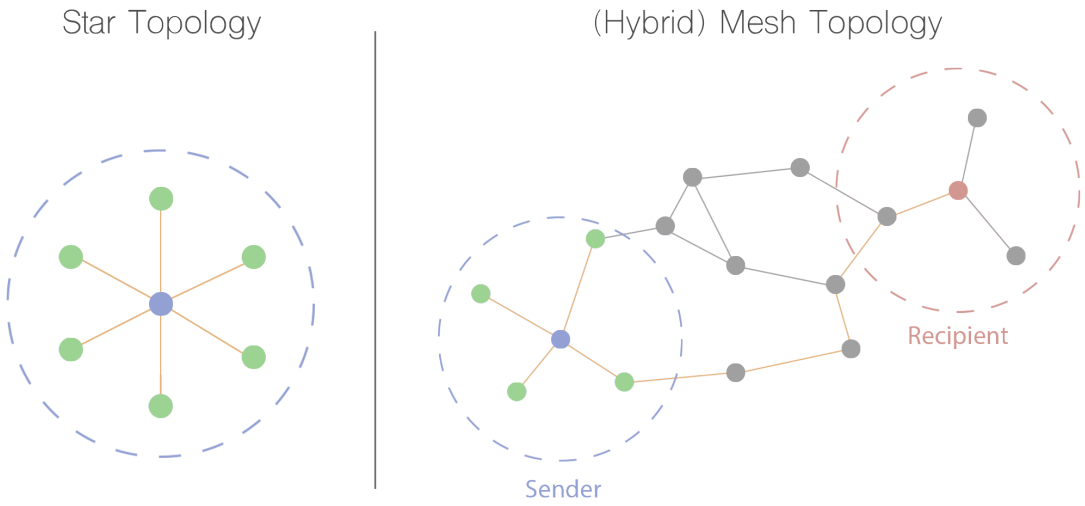


Figure 5: The left illustration (a) depicts a star topology, where devices (green nodes) are dependent on a central hub (blue node) for communicating. The right illustration (b) depicts a mesh topology, where every node is connected to every other node, eliminating the need for a central hub. This allows the sending node (blue) to communicate with the receiving node (red).

In addition to being defined by a topology, wireless networks can also be categorized by the number of hops (single or multiple) required to transmit a message [19]. In single-hop wireless networks the communicating devices are within range of one another and transmit directly (forming a star or point-to-point topology). In contrast, communication in multi-

hop wireless networks may be relayed by intermediary nodes, enabling communication between out of range devices. Such networks are referred to as wireless mesh networks [19].

Each wireless network can be categorized as either infrastructure-based or infrastructure-less. In an infrastructure-based wireless network a base station is in place which connects and handles communication throughout the network, as well as to neighbouring networks. In contrast, an infrastructure-less wireless network must coordinate communication in a distributed fashion.

4.2 Network Protocols

There are several different networking technologies that utilize different topologies, which could be used to implement communication in a home-automation system. Some are purpose-built to be used in distributed environments and consume little energy, but are not widely supported; others may be widely supported but are not suitable for home-automation systems.

Gill et al. identifies "the complexity and expense of the architectures adopted by existing systems" and "the intrusiveness of the system installations" [5] as two of the main reasons that have impeded consumer adoption of home-automation systems technologies. When choosing networking technologies, these reasons must be taken into account. To make the installation as non-obstructive as possible, a wireless network technology would be preferable compared to a wired one. The ideal technology would also allow usage without the need for excessive devices, such as adapters for smartphones or range extenders, which increases the complexity and expense of a system. This section describes four different technologies and previous applications in related fields.

4.2.1 Wi-Fi

IEEE 802.11, more commonly know as Wi-Fi, is a widely used wireless network protocol designed for short range high speed communication between devices ranging up to a few hundred meters [20, p. 12], depending on the protocol version and external factors. The latest standard, 802.11ac, offers a data transfer rate of up to 1300 Mb/s [21, p. 3]. Due to the high speed communication, the current power consumption of Wi-Fi devices can be regarded as relatively high. Recently, however, low power Wi-Fi modules have become available that offer significantly lower power consumption. According to the datasheet of one such device, the RN-131G, the active power consumptioncurrent usage is between 40 mA (receiving) and 210 mA(transmitting) [22, p. 2].

Wi-Fi is a single-hop infrastructure-based technology utilizing a star topology with an access point (AP), as base station. However there is an amendment to the 802.11 standard called 802.11s which enables an infrastructure-based multi-hop mesh networking functionality [23]. An 802.11s network consists of several mesh access points (MAP), that communicate and relay messages amongst themselves. Every MAP also grants clients access to the mesh network, consequently only allowing the clients to communicate with a MAP, and not with other clients.

4.2.2 ZigBee

A fairly new wireless network protocol, well suited to home-automation systems [5] , is ZigBee. Utilizing radio frequency at three bands [24], 2.4 GHz, 912 MHz, 868 MHz, the ZigBee protocol is designed to provide a responsive network with fast transmissions. However, the protocol also restricts the data rate to 250 kb/s, 40 kb/s and 20 kb/s for respective

frequency. This type of design gives the advantage of miniscule power consumption. Moreover, ZigBee natively supports both the mesh and star topologies, allowing the formation of either a single or multi-hop wireless network.

Despite the appealing factors of ZigBee, the technology has drawbacks, mainly its lack of native support in most consumer devices such as smartphones. To directly control a ZigBee network from a smartphone, additional hardware, an adapter, must be installed. Gill et al. [5, p. 426] proposes building a gateway, bridging a Wi-Fi network with a ZigBee network, thus removing the need for an adapter in every smartphone, while also enabling access to the Internet. However, such a gateway also presents challenges. Firstly, there is the issue of how to easily configure it to work on an existing Wi-Fi network. Secondly, Gill et al. [5, p. 428] discovered that the average access delay doubled when using Wi-Fi, compared to using ZigBee directly. Thirdly, a gateway limits the access range of the network to that of the Wi-Fi network, thus removing some of the advantages of ZigBee's mesh topology.

4.2.3 Ant

Another short range, low power, wireless network protocol is Ant. Mostly found in sports gear today, Ant is designed to run on small power sources and mainly to be used as a wireless sensor network, thus employing low data rate transmissions. Ant has been designed to work within ranges of 30 meters, given clear line of sight [25].

Similar to ZigBee, Ant also natively supports the mesh and the star topologies. However, unlike ZigBee, Ant is natively supported by some high-end Android smartphones, but not any iPhone model [26]. Thus, a major part of all smartphones still requires additional hardware to connect to an Ant network.

4.2.4 Bluetooth Low Energy

Bluetooth is a wireless technology developed to replace cables between electrical devices within short range [27, p. 17]. Bluetooth low energy (BLE), or Bluetooth v4.0, is currently the latest major Bluetooth standard introduced by the Bluetooth Special Interest Group. This standard is intended to support low hardware development costs and, especially, low power consumption. The technology is optimized to be powered by small power sources, such as coin cell batteries [28]. According to the datasheet of one such commonly used device, Texas Instruments' CC2540, the active power consumption is between 19.6 mA (receiving) and 24 mA (transmitting) [29, p. 1]. However, studies show that the average power consumption of the CC2540 is less than 0.1mA in a normal scenario, with energy saving enabled [30]. Moreover, the BLE standard imposes no range limitations [28], but most Bluetooth devices have a range of 10 meters.

Bluetooth is a single-hop infrastructure-less technology utilizing a star topology. Every Bluetooth device can either be a master (central) that initiates connections, or a slave (peripheral) that accepts connections. In BLE these are called Generic Access Profile (GAP) roles and communication is only permitted in a connection between a master and a slave. Within an established connection, communication is regulated by the Generic Attribute Protocol [27, p. 519](GATT). This protocol utilizes a service, intended to support a specific feature, such as a heart rate monitor or temperature sensor. Communication between devices is performed via characteristics [27, p. 529], which are data fields accessible from both parts of a connection, declared in the GATT service. By writing to and reading from these fields two devices are able to communicate.

Before transferring data, a designated master initiates connections with up to seven slaves, forming a star topology known as a piconet. Similarly to Wi-Fi, Bluetooth does not natively support mesh topologies. However, research has shown a possibility of combining several piconets to form a scatternet (mesh topology), thus forming a multi-hop infrastructure-less mesh network, as illustrated by figure 6. In a scatternet at least one slave in every piconet also participates as a slave in another piconet, and thereby enables relaying of messages between piconets. A major topic of research is formation of scatternets, i.e. how to distributedly designate masters and relay slaves. There are several methods for forming scatternets including MTFS [31], BlueTrees [32] and BlueStar [33]. MTFS, developed by Sunkavalli and Ramamurthy, compared to BlueTrees and BlueStar, is a distributed formation algorithm, thus the scatternet formation delay is independent of the number of nodes, and is on average 7 seconds for a network with 0.2 nodes/m^2 [31, p. 3598]. Using a smartphone, which is often a master, in such a scatternet could therefore entail a few seconds delay during formation or changing the topological layout through geographical relocation.

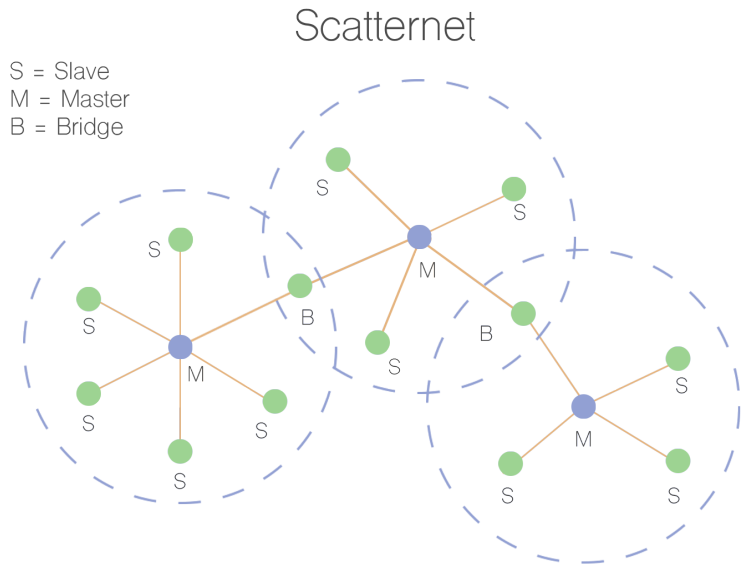


Figure 6: Combining several piconets forms a scatternet, allowing for a multi-hop infrastructure-less mesh network.

Bluetooth Low Energy enables communication without having an established connection, by using advertisements. BLE advertisements are broadcasts sent from a device and are available to any other device, within range, scanning for advertisements. In support of advertisements, BLE introduced the two roles of Broadcaster and Observer, in addition to the already existing slave/peripheral and master/central roles. Both Broadcaster and Observer are limited to one-way communication through advertisements: the Broadcaster only sending advertisements on a regular interval, and the Observer only scanning for advertisements. However, these roles can be combined with the central or peripheral role, to enable connection functionality. Advertising is however limited in the amount of data, carried in one advertisement packet [27, p. 389]. The maximum length of the data carried in an advertising packet is 31 bytes.

Through the use of advertisements, T. Snekvik of the Norwegian University of Science and Technology, has developed the nRF51-ble-broadcast-mesh [34], which is a connectionless based Bluetooth mesh protocol for Nordic Semiconductor's nRF51 series. Compared to a scatternet, which requires the establishment of connections, the nRF51-ble-broadcast-

mesh has no notable formation delay and is not affected by any changes in topology structure. Instead the broadcast mesh works by flooding every message to every node through advertising channels on which everyone can receive. The mesh network has a global state, containing some values, and all nodes in the network have a local state copy of the global state. Whenever a value change has been observed, each node relays the change to the rest of the network, thus synchronizing its local state with the global mesh network state. As a consequence of communicating via advertisements, the maximum data size per value, in the mesh network, is 28 bytes [35], which may not suffice in some cases.

Normally, communication is done only in a connection. Within an established connection BLE utilizes Generic Attribute Protocol [27, p. 519], GATT, a communication protocol and service framework controlling the communication flow between devices. The communication is regulated with the use of a GATT Service, which is a service intended to support a specific feature, such as a heart rate monitor or temperature sensor. Communication between devices is performed via a concept known as characteristics, declared in the GATT service. A characteristic [27, p. 529] is a data field, provided with a definition describing properties, descriptors and permissions, accessible from both parts of a connection. By writing to and reading from these fields two devices are able to communicate with each other.

4.2.5 Broadcast Storm

In a mesh network that uses a flooding technique, a Broadcast storm could occur. Broadcast storm is a phenomenon where several nodes in a network attempt to uncontrollably forward messages by flooding, often resulting in severe redundancy and collisions [36, p. 151]. Ni et al. [36] introduced several mechanisms: Probabilistic scheme, Counter-based scheme, Distance-based scheme, Location-based scheme and Cluster-based scheme, to prevent this phenomenon in a Mobile Ad Hoc Network (MANET). All of these schemes attempt to limit forwarding of redundant broadcasts. Ni et al. found their location-based scheme, which limits forwarding based on location, to be the most effective. It reduces up to 50% [36, p. 159] of redundant forwardings, while maintaining about 95% accessibility in a 5r x 5r area containing 100 nodes (where r is the transmission radius). However, this scheme requires location information about each individual node, which may not always be available. The Counter-based scheme [36, p. 155], does not require any location information, but instead makes use of a random backoff mechanism, in which every node waits for a random amount of time before forwarding a message m. During this wait, a counter c is incremented every time the same message m is received. If c reaches above a counter threshold C, the forwarding of message m is canceled. A lower threshold value C decreases the number of redundant advertisements in dense areas, but also decreases the accessibility in non-dense areas, whereas a high C-value has the opposite effect. Ni et al. found their counter-based scheme to decrease the number of forwardings by about 30% [36, p. 158], while maintaining a reachability of close to 100%, in a 5r x 5r area containing 100 nodes, using C=3.

4.3 Results

This project implemented mesh networking utilizing Bluetooth low energy technology. Similar to the nRF51-ble-broadcast-mesh protocol described in section 4.2.4, the network protocol developed, from here on referred to as Simple BLE Mesh Network (SBMN), employs advertisements to communicate, without connection establishment, amongst nodes. However the SBMN protocol does not have a global state, but instead function purely as

transport protocol. Furthermore, an application framework, utilizing the SBMN protocol, was developed to achieve the desired functionalities of the home-automation system. This section describes the resulting SBMN protocol, the concept of implementation of the SBMN protocol, and the application framework.

4.3.1 Simple BLE Mesh Network

The SBMN protocol provides BLE nodes with the ability to communicate through a multi-hop infrastructure-less mesh network. The protocol handles only the transport and delivering of data, and relies on the BLE stack to handle the physical transmission of data. In comparison with the OSI model [19, p. 78] the SBMN protocol implements the functionality of both the Transport and Network layers, in order to decrease header size, while the BLE stack implements the Link and Physical layers (See figure 7). Furthermore, an application, in the application layer, supplies the SBMN with the message to send.

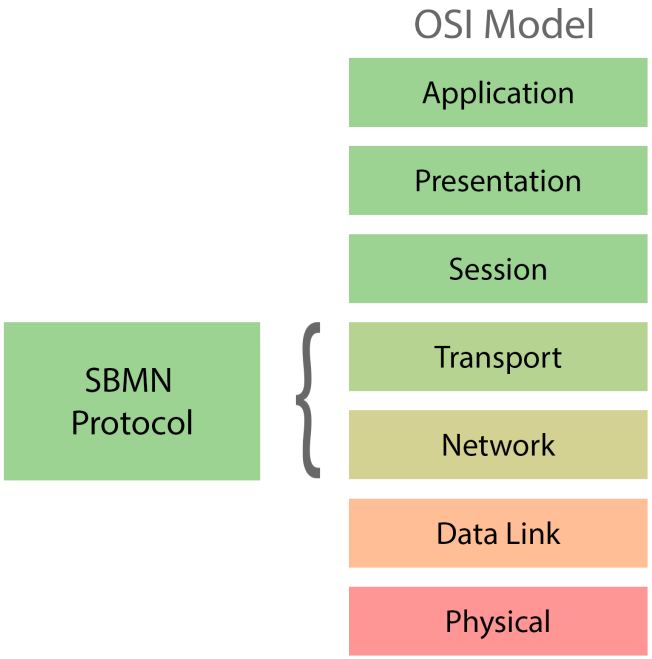


Figure 7: The SBMN protocol implements the functionality of both the Transport and Network layers of the OSI model.

Each user of SBMN is required to run a combination of the BLE peripheral-and observer GAP role. This allows both for detecting advertisements and sending advertisements, but also provides the ability for connection establishment. Furthermore, each user of the SBMN protocol must hold a network identifier, representing the network to which they are connected, and a node identifier, uniquely representing itself within the network to which it is connected.

The SBMN protocol transports data through BLE advertisements. Every advertisement sent by the SBMN protocol, from here on called SBMN advertisement, consists of a header and a message. The header, as illustrated by figure 8, includes information such as the network identifier, the source and SBMN advertisement type. There are four different SBMN advertisement types available: broadcast, group broadcast, stateless message and stateful message. Broadcasts are received by every node in a network, and thus requires no destination. Group Broadcasts, however, are only received by nodes which have joined the

group to which the message is addressed (See appendix A). Moreover, both Stateless and Stateful messages are only received by the node to which they are addressed. A stateful message differs from a stateless message in that it requires an acknowledgement (ACK) to be sent by the receiving node upon successful transmission. If no ACK is received the stateful message will be resent after a period of time, or possibly raise an error (See appendix A).

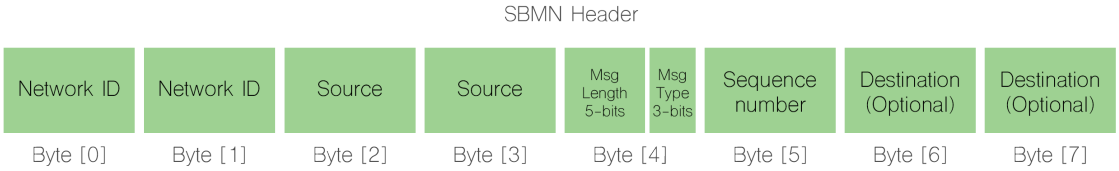


Figure 8: The structure of a SBMN header.

When a message is to be sent using the SBMN protocol, a SBMN advertisement of desired type is constructed and passed to an outgoing advertisement queue. This advertisement queue shall be implemented by the user of the SBMN protocol, and the user is responsible for advertising the enqueued items, thus linking the SBMN protocol with the Link and Physical layers of the BLE stack. BLE uses advertising based on intervals [27, p. 968], meaning that a BLE device will keep sending the same advertisement, on interval, until it is stopped. Consequently, some BLE devices, such as the CC2540, does not allow a single advertisement to be sent only once, and thus SBMN requires the user to implement an advertising timeout, as illustrated by figure 9. This timeout must be at most 250 ms, allowing a theoretical throughput of at least 4 SBMN advertisements or 92 bytes per second. Is up to the user of SBMN to decide the advertising interval, and advertising timeout to best fit the BLE device that SBMN is used on.

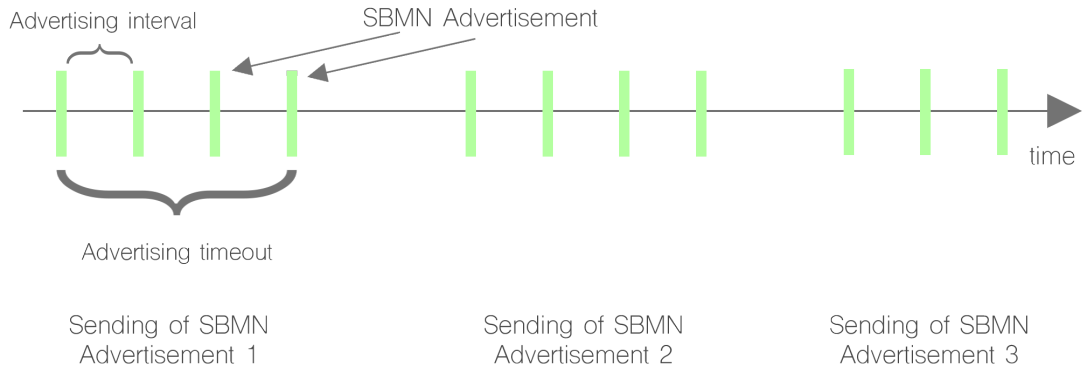


Figure 9: Illustration of SBMN Advertising process. The image shows periodical advertising with an advertising timeout, required by the SBMN protocol.

The SBMN protocol uses a flooding technique to deliver advertisements to their destination. Using a flooding technique, as opposed to using a routing technique in which a message has a predetermined way from node to node, means that SBMN advertisements are forwarded by every node until it reaches its destination. The default state for any node is to continuously scan for advertisements. Upon receiving an advertisement, it is passed to the SBMN protocol for processing and, if addressed to the processing node, delivered to the application layer. If not addressed to the processing node, the SBMN advertisement will be forwarded to the rest of the network. However, forwarding is only done under certain conditions. Firstly, the same SBMN advertisement is never forwarded

twice, based on sequence number. Secondly, SBMN employs the forwarding conditions of Counter scheme, as described in section 4.2.5, to prevent a broadcast storm. The default counter threshold value is C = 3, but can be changed by the user. During this project, an implementation of the SBMN protocol was developed to be used on a BLE Mini, which is a breakout board for Texas Instruments' CC2540 processor [37]. The CC2540 runs a combined peripheral and observer GAP role profile, which is required by SBMN. In order to find the optimized advertising settings for the GAP role, an experiment called the SBMN advertising experiment, further described in Appendix C, was performed. In this experiment, four CC2540 devices received and forwarded advertisements from a fifth, which sent advertisements, illustrated by figure 10. The experiment was performed multiple times with different advertising settings. The results showed that the best setting for detecting advertisements is a long advertisement timeout (85+ ms) and a short scanning interval (16 ms). Furthermore, the advertising interval had little effect on the reception rate.

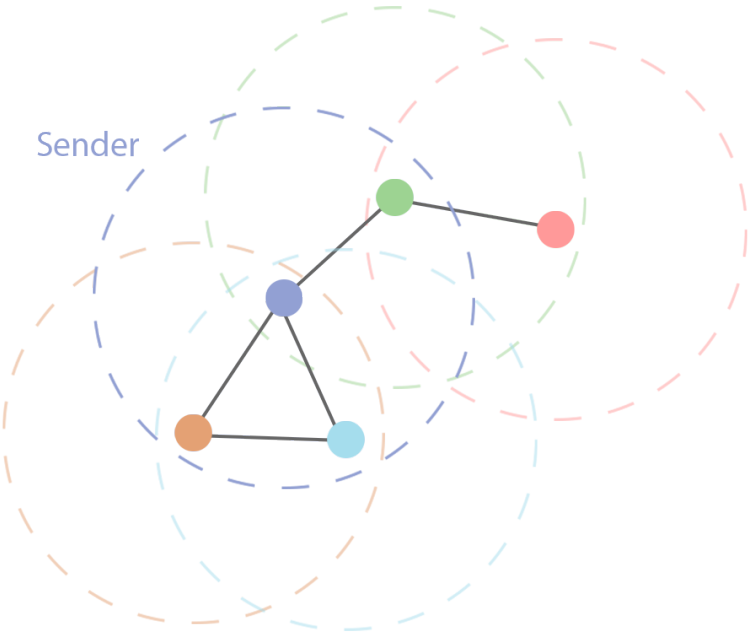


Figure 10: The set-up of the transmission/receiving experiment.

The SBMN advertising experiment was performed again, with one of the receiving nodes also sending a response to the sender. The sender logged the received responses and these were analysed. The results of this experiment, as opposed to the first, showed that the best setting for the sender was a short advertisement timeout (around 60-80 ms), a long advertising interval (30 - 35 ms), and a long scanning interval (30 ms). If the advertising timeout was set to be long (100+ ms), the response was sent while still advertising the original message. This caused problems in receiving the response, mainly because there is no way to control exactly when to advertise and when to scan on the CC2540. Instead, this is handled by an operating system that schedules these tasks at an appropriate time, not always ideal for this set-up. Furthermore, if both the advertising and scanning intervals were set to be low (20, 16 respectively) there seemed to be overhead in scheduling these tasks. Using longer advertisement and scanning intervals, we believe is to decrease the overhead, which results in better reception.

For optimal results, the experiment showed that the scanning window, scanning interval, advertising interval, and the advertising timeout of the GAP profile should be set

to 30 ms, 35 ms, 37,5 ms and 85 ms respectively. This would yield an average reception rate of at least 86%.

4.3.2 Node Applications

In order to enable applications to utilize the SBMN protocol, the CC2540 implementation has an application framework. Each application in the framework has a unique identifier, and each message sent from, or to, an application contains this identifier. Whenever an incoming message is received from the SBMN protocol, a unit called Node Application Manager routes that message to the application it is addressed to, as illustrated by figure 11. Furthermore, each application has a set of opcodes (operation codes) dictating the action to be performed, and every message contains one opcode. For example, consider an application with an identifier 33. The application controls a relay, with opcode 1 to turn on the relay. Sending the message [33, 1] to a node via the SBMN protocol would cause the message to be processed by the relay application, which would consequently turn on the relay.

There are mainly four applications used in the prototype created in this project. Firstly, the Relay Switch application, which is used to control the relay in the power switch adapter, as described in section 3.3. Secondly, the Dimmer application, which is used to control the dimmer in the dimmer adapter. Thirdly, the Node Information Service, which is used to handle node properties such as name and services provided by a node. Finally, the Network Information Services, which is used to handle network properties such as network name and the pairing of new nodes.

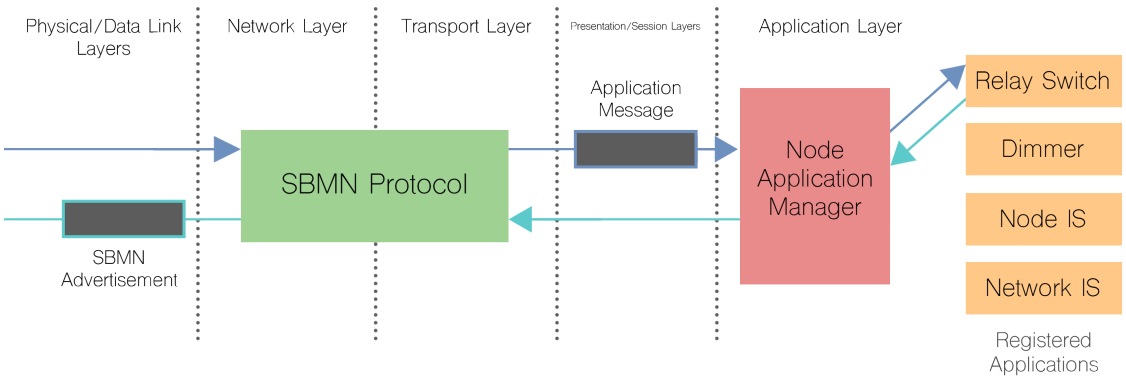


Figure 11: The lifecycle of communication using SBMN and Node Applications illustrated in context of the OSI model. The figure shows how data is carried through the different layers of the OSI model, first into the SBMN protocol and then on to the application layer.

4.3.3 Connecting a Central

The SBMN protocol enables mesh networking abilities to BLE devices which can operate in a combined peripheral and observer role, as required by SBMN. However, some devices, for example smartphones using an earlier version than Android 5.0 [38], only support the central role, and not the peripheral role. To enable access for these devices, referred to as centrals, to an SBMN driven network, all nodes allow a central to establish a connection. This entails that every node must send connectable advertisements, not to be confused with SBMN advertisements, for a central to discover them. As it does not matter to which

node in the network a central connects, these advertisements are identical in content. Their content include the network identifier and name of SBMN network.

To connect to any node in the network, the central scans for connectable advertisements, and, grouping by network, selects the node which has the highest RSSI (Received signal strength indication) to establish a connection. This procedure can also be run at a constant interval to maintain a connection to the network, if the central is mobile. For example, if the central moves to a node, closer than the currently connected one, it would discover a node with greater RSSI, and thus connect to it. This enables, for instance, smartphone users to move about and remain connected to a SBMN network.

For a central to communicate with a SBMN network, each nodes supplies a mesh GATT service containing two characteristics, RX (receive) and TX (transmit). To communicate with the SBMN network the central can write data to the TX characteristic according to appendix A.8. The node will then forward the data to its SBMN protocol. Furthermore, each node application will also write any incoming response to the RX characteristic, which the central can subscribe to. Operating in a connectable state, a node may have to handle a connection, scan for SBMN advertisements, and advertise, simultaneously. Results show that when conducting the SBMN advertising experiment while in connection, the average reception rate of a CC2540 node drops from 86% to 79%. For further details, refer to Appendix C.3. Seeing as how an average reception rate of 79% may be deemed insufficient, a central in connection with a CC2540 may conduct a scanning of its own, to detect SBMN advertisements. Whenever a central sends a message that will generate a response, the central may conduct a BLE scan for SBMN advertisements. This requires that the central has a passive version of the SBMN protocol, only able to parse incoming advertisements. Conducting the SBMN advertising experiment with a connected central (LG Nexus 5) scanning for advertisement resulted in a 95% reception rate, which is an increase of 16%.

4.4 Discussion

Previously mentioned network topologies and protocols all come with different advantages and disadvantages. To achieve the purpose of building a scalable home-automation system, a multi-hop mesh topology is prefered to a single-hop star topology, for two reasons. Firstly, an infrastructure-less multi-hop mesh topology is well suited for distributed environments, allowing the system to be used in houses of different sizes and shapes. Conversely, using a single-hop star topology would require that a base station be placed in a central location, and even then its range may not cover an entire house. Secondly, a mesh topology allows for redundancy in the network, making it more fault tolerant. In contrast, a single-hop star topology would stop function all together if the base station failed.

Regarding the choice of technology, arguably ZigBee or Ant would probably be the easiest to use, since they both natively support mesh topologies. However, as the project aims to develop a convenient system, controlled by a smartphone or similar device, the technology used needs to be supported by a majority of such devices, which both ZigBee and Ant do not. Therefore, the choice is narrowed down to Wi-Fi or BLE. There are two main reasons for choosing BLE. Firstly, BLE is intended to be used on short distances between devices only requiring low data rate transfer, thus in general having a lower power consumption than Wi-Fi. Since the preliminary survey showed that the average distance between home appliances and other devices is less than 10 meters, BLE's range is sufficient. Secondly, to conveniently use a Wi-Fi mesh network with a smartphone, the Wi-Fi mesh would have to operate on the same network as an existing home Wi-Fi network, in order to not have to switch between networks. We believe that this could potentially be a

difficult problem to solve, both in a user friendly manner and technically. In contrast, a BLE central device, such as a smartphone, is theoretically able to connect to 7 peripheral devices, avoiding the problem described above.

The resulting SBMN protocol uses advertisements with a flooding technique to deliver messages in a mesh topology. There are two reasons for choosing advertisements over forming a scatternet in order to enable BLE mesh functionality. Firstly, there is no formation delay when using advertisements since they are connectionless. Secondly, the average amount of data needed to be sent between nodes in this system can be contained within a few advertisements. If that had not be the case, a connection based scatternet may have been a better choice, since the SBMN protocol only has a low data transfer rate. As a consequence of using advertisements, we believe that using a flooding technique is the preferable choice of delivery. Using a routing scheme would require the header to include two additional bytes for the next-hop destination, leaving only 19 bytes to the message. Furthermore, using a routing technique would require more memory on the BLE device, which could be a strained resource.

Although the SBMN protocol, in theory, may be sufficient to transfer data, implementing the support required of the physical and link layers on the CC2540 proved to be difficult. As shown by the results of the SBMN advertising experiment, the implementation can be considered insufficient during two way communication, only having an average reception rate of 86%. This number should be regarded with caution, since no account was taken to other devices using the same frequency-bands. However, we believe that by implementing the SBMN protocol on another BLE chip, which allows for more exact control over when to advertise and scan, the average reception rate would increase. This would also reduce the need for a central to scan for SBMN advertisements, as described in section 4.3.3, as scanning does have some drawbacks. Firstly, scanning for BLE advertisements contributes to a higher power consumption, since it involves performing more tasks. Secondly, there is a potential loss in range. When relying on a connected node to relay a response, the central can be at a distance r from the SBMN network, if r is the range between the central and the connected node, as illustrated by figure 12. However, to detect SBMN advertisements, the central must be within range of the other nodes in the SBMN network, as the connected node will not forward messages addressed to itself.

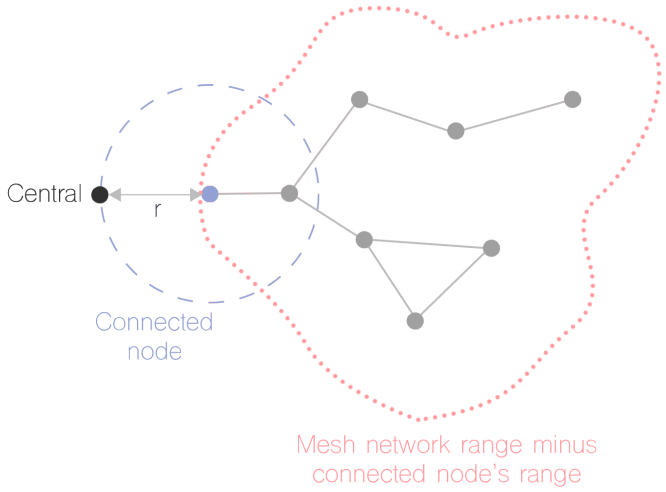


Figure 12: The potential range r is lost if scanning for SBMN advertisements with a BLE central device, as it must be within the red line.

Even though implementing the SBMN protocol on another BLE device might improve performance, we cannot conclude that the protocol would work better, or even equivalently, in larger networks, since the experiment only involved five nodes. A simulation would be required to evaluate its performance in large scale networks. We believe that in large networks, even though measures were taken to prevent a broadcast storm, the average reception rate will somewhat decrease. For such networks (e.g. >10000 nodes) it might be better to implement a routing technique, instead of a flooding technique. However, we believe that the average user will not possess a large number of nodes, in which case the SBMN protocol may be sufficient enough.

5

Android Application

This home-automation system needs a graphical user interface (GUI) in order for a user to connect to the network and control their adapters. As the goal of this project is to create a convenient and user friendly system, this interface should naturally be available and easily operated. A common possession is a smart device, such as smartphones or tablets, which is a suitable platform to develop an interface for. Currently, Android holds a 76.6% [8] worldwide smartphone operating system market share, thus the interactive interface for this system has been developed for Android systems. This section introduces background theory of Android programming, as well as the design and implementation of the resulting application. It will also present and discuss synchronization and handling of important data structures.

5.1 Android API

The main element of an Android application is the Activity [39] [40] class, which handles content and the user interaction. An interactive Android application is always centered around at least one Activity. The task of an Activity is to both generate a graphical interface and execute Java code.

An Activity may contain Fragment [41] [42] objects, a component similar to an Activity. The Fragment class is a way to create user interfaces where sections of the screen change, without having to switch between Activities. The main differences between an Activity and a Fragment is that a Fragment is not a Context [43] and that the Fragment's life cycle is dependent on the life cycle of its parent Activity. Fragments can communicate with the parent Activity and other Fragment objects through the Activity that contains them.

Objects such as text fields and buttons mostly have their own classes. These classes are often, if not always, extended from the View [44] class, which is the basic building block for any graphical component in an Android application.

5.2 Three-tier Architecture

Three-tier architecture is a widely used coding standard, mainly for client-server systems. Xu describes the structure of the three-tier architecture with the three main components: the presentation layer, business logic layer, and the data access layer [45]. The presentation layer is often used as the user interface layer of a system, the business layer is the core layer that carries out a specific function, and the data access layer contains all data, such as a database.

Xu explains that the main idea behind keeping the three layers separate is to guarantee the security of the data access layer, as clients that use the presentation layer do not have direct access to it. Instead the presentation layer communicates with the business logic layer, which in turn communicates with the data access layer.

5.3 Results

The goal of designing the GUI of the Android application is to create an interface that is easy to use. Thus the application is meant to give users necessary functionalities that allow them to use the system without feeling constrained. As the application starts up, users can open the Navigation Drawer [46] by pulling it out from the left side of the GUI. The Navigation Drawer contains options to display and access all the features of the application. Pressing any of these options opens their respective fragments with interfaces for each function: Adapters, Categories, Presets, and Schedule. As for the underlying code structure, the whole application has been developed with the three-tier architecture in mind. Therefore, different functionalities of the system have their own Manager, which is the interface to the business logic layer, and Service, which is the interface to the data access layer.

In this home-automation system the user must be able to control the connected adapters. The interface needs to be easy to understand and provide information about the system's state. Moreover, the interface also needs to have options to make changes to the system state, such as turning on a device connected to a power switch adapter. This is the most basic and necessary function that the application provides. The basic adapter information of a system can be found in the Adapters fragment and is contained in a back-end storage (NetworkService), retrieved by the GUI from a NetworkManager. The information is displayed in a ListView [47], illustrated by figure 13, with a custom ArrayAdapter-class [48] called NodeAdapter. By using the ToggleButton [49] displayed on the far right of the list item, the user can toggle the status of the adapter to on or off. Adapters with the function of acting as dimmers are represented with an extra Seekbar [50], controlling the brightness value of the dimmer.

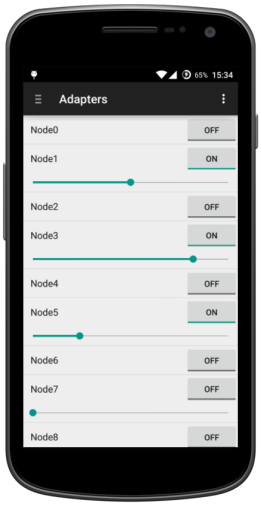


Figure 13: The Adapter fragment displays the connected network's adapters and allows users to control the system.

Should a user have a high amount of adapters, it could become tedious to scroll through the whole list of available adapters in order to find a certain set. In such cases, users should have the option of creating their own collections of adapters and save them for future use. Users may also wish to control these collections, by for example toggling a specific group of adapters. Such user-created adapter lists can be created in the Categories fragment. The graphical structure of the Categories fragment is similar to the Adapters fragment, as illustrated in figure 14. The fragment displays the categories in a ListView

with a custom ArrayAdapter-class called CategoriesAdapter. The list of categories that the CategoriesAdapter uses to generate the ListView interface is stored in a CategoriesService, and retrieved by the GUI from a CategoriesManager. The Categories fragment also uses the CategoriesManager to create, edit, and delete categories. To create a category, the user clicks the create-button on the action bar [51]. Once clicked, the user is prompted to input a name for the category and choose which adapters are to be included. After this process, a new list item that represents a category can be seen in the ListView. This clickable list item contains the name of the created category and a clickable ToggleButton. By clicking the list item, an editable list of the adapters linked to the clicked category is displayed. By long-clicking, i.e. press and hold, a category, a menu is displayed with options to delete and rename the chosen category. A category's associated ToggleButton indicates the amount of adapters that are activated and the total amount of adapters. The ToggleButton toggles all the adapters in the category when clicked; if there are any adapters on, all adapters are turned off, and if there are no active adapters all adapters are turned on.

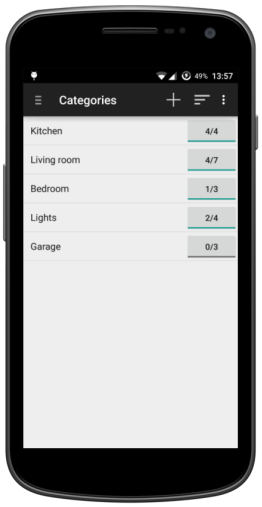


Figure 14: The Category fragment allows users to organize adapters in custom lists. Adapters in categories can be toggled together by clicking the category's respective ToggleButton.

There may be times where users wish to not only turn on an entire set of adapters, but also have the option to have some of them turned off. It would then be preferable to have a feature that allows the user to group a number of adapters together and save them for future use with the ability to turn some of them off. Users can gather adapters in custom lists, called presets (depicted in figure 15), and set desired statuses for certain situations. These presets can then be used to toggle the set of adapters to their specified statuses. The similarities and differences of categories and presets are apparent in the Presets fragment, as both fragments' layouts utilize the same simple ListView structure, but have different custom ArrayAdapters; the Presets fragment has its own PresetsAdapter class. Moreover, instead of a ToggleButton, the Presets fragment has two buttons for activating and deactivating the linked preset. Presets are managed in the same manner as adapters and categories: a PresetService contains the information, while a PresetManager handles the retrieving, creating, deleting, and editing of presets. Users can create new presets by clicking the create-button on the action bar. Similarly to categories, users will be prompted to name the preset and select adapters to include. Presets can also be created from a category by selecting the appropriate option when long-clicking a category in the Category fragment. A preset can be clicked to replace the Presets fragment with a fragment

Figure 15: A preset functions similarly as categories, with the possibility to set a specific status for adapters for certain situations.

that displays an editable list of associated adapters, where Switches [52] can be clicked to toggle adapter statuses. Figure 16 illustrates an example of this. Deletion and renaming of presets can be performed by choosing the respective option in the menu that appears when long-clicking a preset. A preset's associated ON-button toggles the adapters to their set status, while the OFF-button turns off the adapters that are set to be turned on.

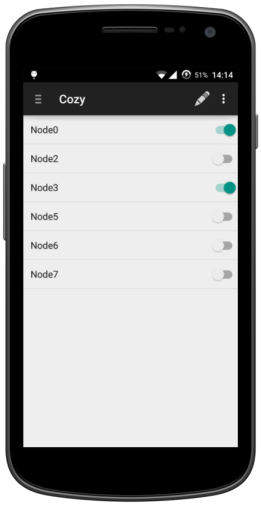


Figure 16: An example of a preset. Green Switch-buttons indicate that the respective adapter is to be turned on when the preset is activated. Gray Switch-buttons indicate the opposite.

Users may find themselves repeating the same action over and over again. In times like these, the user may wish to automate their behavior by creating events, being executed on certain conditions. These events should be viewable and editable. For users that own a gateway, an extra device that extends the functionalities of the system (described further in section 7), the application includes features for scheduling adapter status changes. When the Schedule option in the navigation drawer is clicked, the user is presented with a calendar view of all their events in a flowing scheme. A library called Android Week View by Alam

Kanak [53] was used to display the calendar view. Events appearing on the calendar are managed by an EventManager. An event is basically a combination of a preset and triggers. A trigger is a condition for an event to activate, such as time and date. Creation, editing, and deletion of events are all performed by the EventManager and the actual events are stored in the EventService. By using the Android Week View library, retrieved events are automatically rendered with a one-month margin, meaning that the current month, the previous month, and the next month are rendered at once. By doing so, the possibility of interface delay (lag) when scrolling through the calendar is reduced [54]. The create-button on the action bar allows users to input name and time for a new event. Users can also choose color for the event title that appears on the calendar view when they click the save-button.

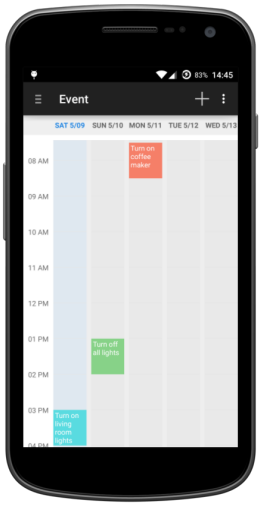


Figure 17: The schedule displays examples of events that users can create.

In order to increase user convenience, information about categories, presets, and schedules are bound to accounts rather than smartphones, and are stored in a database on a server. Sending requests to a server everytime a user wants to access objects would cause slowdown in the application. This slowdown is reduced by keeping a local state in the phone while also synchronizing with the information on the database. The expressed state is basically a list of all objects associated with the user. When a service for categories, presets, or schedules is asked for their respective objects, it checks how long ago they were updated from the server. If they were recently updated, the local state is returned and if not, a request for the given resources is sent to the server. When the client receives a response, the local state is updated according to the server's state. When a user adds or updates objects, the server is notified instantly. If the server is unreachable for any reason, the object to be updated will be added to a queue. This queue is checked at given intervals to push the application's local content to the server when changed.

For users who wish to use multiple home-automation systems, an option for choosing personal networks can be found in the Navigation Drawer. This allows them to separate their collections of adapters, categories, presets, and schedules, by associating the data structures with their respective personal networks. Upon startup, the application will automatically search for, and connect to, any adjacent SBMN network that have been previously added to a user's list of personal networks. However, if the user has no personal networks, the application will search for new networks. The user can then choose to

add found networks to their collection of personal networks, saving them for future use. Moreover, the user can view which personal networks that are available, as well as connect to one by tapping its name in the Navigation Drawer.

5.4 Discussion

While Android 5.0 enables BLE functionality [38], such as advertising, the application developed for this system does not make use of any such functionality. Instead, users must connect to and control the system by using an adapter as a peripheral device, as all Android devices do not have BLE advertising functionality. As a consequence to the application requiring a connection with an adapter, users must disconnect from any other Bluetooth device in use, in order to use the system. This may seem inconvenient, but we prioritized supporting as many Android devices as possible, rather than only devices with Android 5.0 or above.

The current interface design of the Category fragment allows users to toggle all adapters of a chosen category by clicking the associated ToggleButton. However, as specified in the Results section above, if there are any adapters that are turned on in a category, toggling the category will turn all of its adapters off. This means that if a user wishes to turn on all adapters for a category that contains at least one adapter that is already turned on, the category must be toggled twice. Using a design with two separate buttons for toggling categories on and off would avoid this situation. However, this would take up more space of a list item and also make the design of the Category and Preset fragments similar, possibly making them difficult to distinguish.

The current implementation of triggers only allows for events to happen once. Common scheduling applications, such as Google Calendar, include the possibility of having events repeat themselves [55]. This feature was not prioritized due to time constraints. There were also plans on having more types of triggers, for example triggers based on weather conditions or electricity price. This would require collecting data from third party APIs. However, the project group found that there was not enough time to research and implement this, leaving the scheduler with only one type of trigger.

The synchronization of objects has been implemented in a relatively simple manner due to lack of time. There are multiple anomalies that can occur if several clients manipulate resources in parallel. Even though they may be managed correctly by the server, misleading information may be shown to the user. However, situations where this may occur are considered non-critical. To make sure the local state in the application is updated, the user may simply restart the application. This is not ideal, but is a solution found to be sufficient enough for a prototype.

The idea of giving the server functionality for pushing content to an Android device was considered. This would erase extra network and computation costs from polling resources to examine whether they have changed or not [56]. The smartphone would receive a notification when a resource of interest is changed and the application would then request the resource from the server. This would also allow the user to receive updates made from other users instantly. While this improves user convenience, it would require a lot of work, and thus a decision was made to not utilize any advanced form of synchronization.

6

Cloud Server

If a client application would lose vital information that is only saved locally, recovery can be a difficult task. However, by allowing data to be stored in a remote location, information can be restored without greater effort. It also allows for the ability to synchronize data between several clients. There are several ways to access remotely stored information, one of which is to utilize a cloud server, allowing for data to be managed remotely while still providing a high rate of accessibility. In this system, a cloud server is employed to back up and synchronize settings in the Android application. This section will present means of achieving the necessary features allowing for a cloud server to unfold.

6.1 Web Service Architecture Styles

Web service architectures provide means of interoperating between software, independently of programming languages and operating systems [57]. This is done by identifying global elements needed for interoperating between web services. Furthermore, a web service can be defined as a software system, designed to support compatibility between computers over a network.

6.1.1 Representational State Transfer

Representational State Transfer (REST) is an architecture style for web services. A RESTful web service follows some basic design principles, such as being stateless, using directory structure-like URIs (uniform resource identifiers), and using Hypertext Transfer Protocol (HTTP) methods explicitly [58]. The statelessness of the service increases performance and simplicity as there is no need for the server to synchronize session data with the client. The directory structure of the URIs is similar to a tree data structure [59]. This structure allows for a straightforward use of URIs for developing services. Another principle for REST is that it can transfer Extensible Markup Language (XML) and JavaScript Object Notation (JSON) between end devices. These are formats for interchanging data, using different syntax but both being text only [60] [61]. REST services use these data formats in order to return data representing a resource (object) requested by clients. These features make RESTful APIs a powerful tool to provide different kinds of applications with resources in standard formats. It is also useful when interconnecting systems.

6.1.2 Simple Object Access Protocol

Simple Object Access Protocol (SOAP) is a messaging protocol for exchanging data in a web service [62]. SOAP uses XML as message format and most commonly relies on HTTP for transmission, but can be used with any data transfer protocol [63]. Another important characteristic of SOAP is that it supports the use of any programming style. Web Service Description Language (WSDL) is often used in combination with SOAP to define what functionality the web service offers.

6.2 Server Back-End Solutions

A web application is a program usually stored on a remote server, and run in a web browser. The creation of web applications is often done with the help of web application frameworks. These frameworks simplify the development process by providing solutions to common problems, allowing developers to focus on specific goals and key features of their projects.

Node.js is a framework used for developing web applications. The framework is designed for building lightweight scalable network applications [64]. Node.js applications are written with JavaScript, which is a programming language most commonly used to implement scripts running in web browsers. Executing code in a web browser, in contrast to executing code on a web server, can save bandwidth as well as speed up the results of computations and processes.

PHP is a server-side scripting language designed for web development, but can also be used for general-purpose programming [65]. As PHP scripts are generally run on a server, a user does not need to have any specific software installed. Additionally, users cannot access the source code, improving security aspects. PHP code can be mixed with HTML code, as well as be used in combination with various templating engines and web frameworks.

There are many helpful tools for building RESTful APIs based on PHP. Laravel is the most popular PHP web application framework [66], focusing on simplicity and readability [67]. Laravel supports many features such as smooth routing of HTTP-requests, RESTful controllers, and object-relational mapping (ORM) for databases. ORM is a technique that allows developers to manipulate and access databases through virtual objects in programming languages. As a consequence, programmers do not need to know how objects relate to their data source [68]. Laravel also supports built-in authentication based on sessions. A session is a dialogue between a client and a server, allowing the server to associate information to the client. Using session ID cookies to maintain a login session for users is the preferred choice due to several security reasons [69].

An alternative to both PHP and Node.js is Microsoft's ASP.NET, which is a server-side framework used to build web application using HTML, CSS, and JavaScript [70]. It allows developers to write code using any .NET language (e.g. C#, C++); however, using ASP.NET restricts developers to the Windows operating systems.

6.3 Results

The cloud server was set up utilizing a SQLite database for storing information regarding accounts, categories, presets, schedules, and other related objects (see ER-diagram in Appendix G for a more detailed explanation). In order for a user to remotely access the data from the database, a web service is run on the cloud server. For this purpose, a REST API was created using Laravel and PHP. By routing incoming HTTP-requests to their respective service on the server, clients are provided access to their resources (illustrated by figure 19). The cloud server allows for end-user applications to register and to store system settings to the cloud by using various URLs.

The Laravel ORM is used for database manipulation. Laravel uses something called Models, which is a class corresponding to a database table [71]. By implementing subclasses and connecting them to the system's database, virtual objects of database rows can be employed and manipulated with PHP. In addition, logic is implemented both in the web service and the end-user application to ensure correct manipulation of the database. Furthermore, for security reasons, a user must be verified in order to access information

from the database.

To ensure that the end-user application can interpret the HTTP-responses from the server, the objects from the database are JSON-encoded. Additionally, JSON is used by the server for interpreting received HTTP-requests. JSON data is parsed to some extent using predefined functions provided by the Laravel ORM, but also by custom implementation. The JSON structure and URL-mapping for requests are logically grouped, allowing for easy implementation on client side.

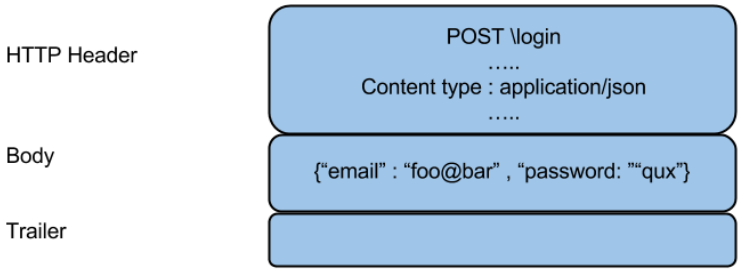


Figure 18: The HTTP request during login attempt.

Depending on the path name in the URL, an HTTP request is routed to its associated controller, which handles the request. Furthermore, the following routes were created: /account, /login, /preset, /category, /schedule, /network, /usernetwork, each of which has a corresponding controller. In addition, each controller has a corresponding service class, which handles the database queries with the help of a model, as illustrated in figure 19.

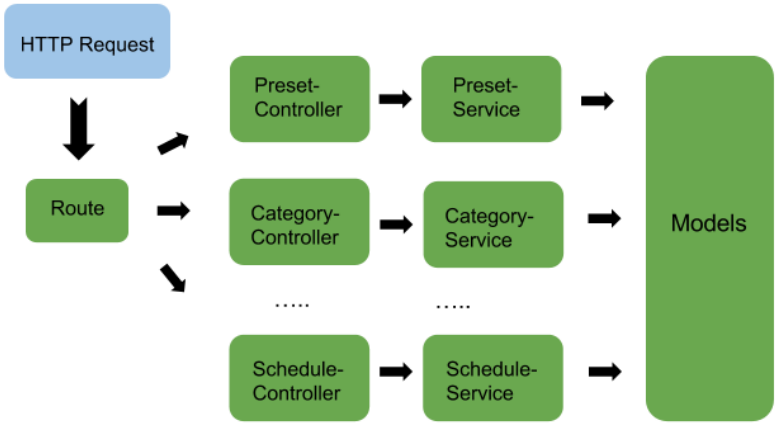


Figure 19: The routing path used for the web project.

Simulation tests were performed in order to benchmark the cloud server's accessibility. Each test was designed to mimic the actions conducted by the application, where a login was performed, followed by a retrieval of a category, in order to estimate the average response time. The simulations repeatedly increased the amount of requests, in steps of 5, up to 30 requests. Ultimately, a simulation test consisting of 50 concurrent simulations was conducted. Figure 20 shows the response time rising steadily as the number of concurrent requests increases.

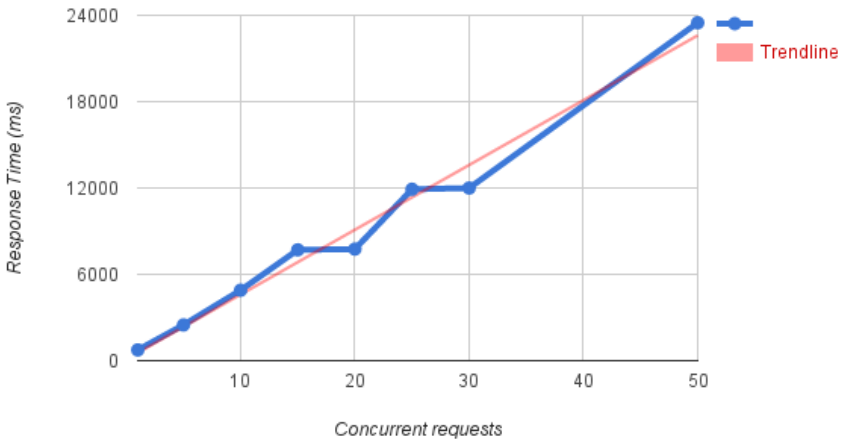


Figure 20: Graph showing the stress test results.

6.4 Discussion

There are several reasons to why REST was used for this project. One reason is the simplicity that the statelessness of REST provides. After comparing SOAP to REST, we decided to use REST as it seemed more suitable for this home-automation system. SOAP could be advantageous if more advanced functions were desired. However, it was deemed to be unnecessary for this system.

A RESTful API implementation on the cloud server provided the desired results for this system. REST covered all the needs of functionality and no planned feature were excluded because of development complexity. Furthermore, accessing resources on the server through the API worked as desired as a result of the use of JSON as well as the comprehensible URI structure.

If the ASP.NET framework would have been used instead, limitations would arise, as it enforces the use of Windows as operating system. PHP and Laravel were chosen above Node.js because of group members' previous experiences. Additionally, PHP allows Unix commands to be run natively, which assisted in the process of development.

Tests showed relatively poor results in regards of response times. Nevertheless, it was deemed to be sufficient to support development and testing of the prototype. The hardware used is not built for handling synchronization of end-user applications on a larger scale, and thus possible improvements of the system would be to utilize a server architecture with greater capacity. Although response time could be reduced by optimizing code and database usage, the server hardware is believed to be the bottleneck for resource manipulations.

The usage of the Laravel framework adds extra overhead to instructions, but also allows developers to put more focus on the database usage rather than the structure for handling HTTP requests. However, during the development process of the cloud server solution, unexpected problems regarding the settings on Laravel were encountered which were time consuming.

As minimum focus was put on security aspects of the server, only one security aspect was implemented, namely the need for authentication in order to access information from the cloud. However, the authentication credentials are sent in plain text, making it insecure

for eavesdropping. This could have been prevented by utilizing SSL (Secure Socket Layer), which is used in order to establish an encrypted link between a server and a client. [72].

By saving personal information on a server, problems of integrity may arise. Sensitive information, such as age and location of users, could be useful for statistics, but is not directly beneficial for users, thus creating an ethical dilemma. Therefore, the current solution of the cloud server does not save sensitive data regarding users' personal information. However, data that is relevant to the home-automation system, such as categories and schedules, is saved. The server is publicly visible and exposed to attacks, opening up the possibility of information being leaked. Examples of measures to protect data integrity could be to store the physical server in a location unreachable by the public and utilizing strict protocols for online authorization.

7

Gateway

The preliminary survey showed that a desirable feature for a home-automation system is remote availability. A solution to this is to run a server, from here on referred to as gateway, in the user's home. This would enable communication between the end-user and the adapter network from distant locations.

Due to the adapters utilizing the SBMN protocol when communicating, the gateway needs to have similar capabilities. For this, the gateway must have a BLE device capable of running the SBMN protocol connected. In addition, to make the user experience convenient, the gateway must be easy to install, have low power consumption, and be silent. With this in mind, single-board computers (SBC) make for an economical solution.

A survey made by the LinuxGizmos.com shows that the Raspberry Pi Model B [73] and the BeagleBone [74] Black are the two most popular single-board computers [75]. Both are low-cost and low-power [76] [77] development platforms that run the Linux [78] operating system, implying an open-source environment. Neither of these SBCs have built-in BLE devices; however, they both have extension pins on top of the board allowing for BLE devices to be installed.

7.1 Results

A gateway used as a portal between the Internet and the SBMN network is used to satisfy the need of remote control of the home-automation system. The gateway is not a requirement for users but merely a tool to add additional functionality, such as control through the Internet and scheduling. Similarly to the cloud server, the gateway also utilizes Laravel in order to handle end-user application requests as well as requests from the cloud server.

The gateway synchronizes with the cloud server by utilizing a separate thread, which repeatedly requests all scheduled events every minute. This is done by using the built-in queue workers within Laravel [79]. The synchronization is realized in order for the gateway to be able to handle scheduled events. Upon synchronization, a copy of each scheduled event is added to a table, called Pending, in the gateway's database, which is then utilized by a schedule handler worker. Furthermore, the handler uses two more tables, called Active and History, in order to keep track of triggered events, as illustrated in figure 21.

The handler's purpose is to act as a scheduler, dividing the scheduling process into two. In the first process, the handler checks the Pending table for events to be triggered; if so, the events are triggered and moved to the Active table. In the second process, it checks the entries within the Active table and deactivates any event that meets the requirements of being finished and moves it to the History table, where the entry will be stored for future reference. Once the worker reaches this state, it adds an identical worker to the queue before terminating.

To make the gateway as easily installable as possible, without taking up much space, a Raspberry Pi, which is a credit card-sized SBC with low energy consumption, is used. The Raspberry Pi runs the operating system Raspbian, which is a customized version of the Linux distribution Debian.

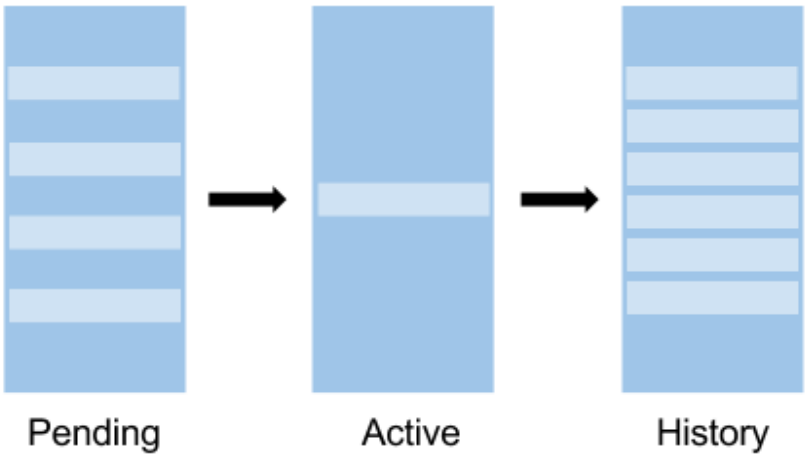


Figure 21: The tables used by the schedule handler.

The gateway is configured so that an end-user only needs to connect it to an Internet-connected router with an RJ-45 connector, i.e. an Ethernet cable. Once it is plugged in, the gateway will automatically connect to the cloud server in order to advertise itself, allowing the cloud server to keep track of the IP-address of the gateway. Through the cloud server, the Android application is able to receive this IP-address and communicate directly with the gateway from a distant location, thus allowing control of the SBMN network.

A BLE mini is connected to the pins located on the Raspberry Pi, allowing the gateway to serve as a link between the Internet and the local mesh network. The BLE chip works just like any other node in the SBMN network. By relaying the user's requests, the gateway gives the client access to the same functions through the Internet as when connected directly to SBMN network.

7.2 Discussion

The gateway acts as both an interpreter and a standalone device, allowing for synchronization to be possible through the built-in queue workers within Laravel. The workers repeatedly request scheduled events from the cloud server every minute, which results in unnecessary requests made by the gateway. This could be prevented by letting the cloud server instantly push recently made changes to the gateway. However, this function was not implemented due to lack of time and priority.

The prototype developed in this project places the scheduling feature of the system on the gateway. Alternatives to this approach would be to implement scheduling either in the Android application or in each adapter. The advantage of using these two alternate approaches is that a gateway would not be needed in order to enable scheduling; however, they both have their drawbacks. By placing the scheduling functionality in the Android application, a smartphone must be connected to the SBMN network when an event is triggered. The Android device would also have to be constantly running which increases battery consumption. If the scheduling functionality was to be implemented on each adapter in the SBMN network, neither a smartphone nor a gateway would be required. However, this would introduce other technical difficulties. The BLE chip placed in each adapter has a limited amount of available memory [37] and thus limits the amount of schedule events that could be used in the system. Furthermore, problems regarding time

synchronization would arise. The clocks of all adapters in the network would need to be synchronized in order to execute events correctly. To avoid these issues, the scheduling functionality was instead implemented on the gateway.

A gateway introduces the possibility to reliably store statistics regarding the system and the users. With this in mind, AI may be implemented, allowing the gateway to learn about users' behavior and make smart suggestions; for example, controlling the system's adapters according to users' behavior patterns. This could benefit users by potentially decreasing power consumption and increasing convenience. However, due to time restrictions and no group member having experience within the field of AI, no such feature was planned to be implemented in the prototype. The gateway could also potentially detect any adapters still active when a user leaves home and notify the user to turn them off, resulting in decreased power consumption. This was a feature intended to be implemented in the prototype. However, due to time restrictions, this feature was not prioritized.

The gateway, being active on a daily basis, contributes to additional power consumption. Because of this, the gateway is run on an SBC, which has less power consumption than a multi-board computer. Furthermore, there is also a difference in computational power when comparing the two; however, the computational power provided by the Raspberry Pi is sufficient enough for the required tasks performed by a gateway.

The same ethical aspects, regarding the storage of user credentials, apply for the gateway, as for the cloud server. Moreover, placing a gateway in a home environment connected to the Internet introduces a risk for cyber attacks. Depending on the level of sophistication of an attack, the attacker may potentially gain full access to the household network. This risk could have been greatly reduced if we would have focused on the security aspects, which was not prioritized.

8

User Tests

When the project group had decided that the product was in a sufficiently useable state, the product was to be subjected to user tests. This process involved uniformly distributing a number of prototype units to five people that would test the product in their home environment for a short period of time (1-2 days). After the testing period, the units were returned to the project group and the users answered a survey containing questions relevant to the results and purpose of this project.

8.1 Results

All the participants in the test group live in apartments with a surface area less than 80 square meters and no one had used a home-automated system before. The participants gave a positive response to the system overall, especially for easy installation. No comments were made regarding the reception. Smartphone application usage was perceived as satisfactory, but some complaints were made about bugs causing crashes in the application. These bugs were also the most commonly expressed reason to why the system was not as convenient as anticipated.

Furthermore, the survey showed that category and preset functionality was useful to the testers. 60% of the testers did not try the scheduling feature but the ones who did found the idea to not be useful. The few participants who tested using multiple phones to control the system expressed satisfaction with the synchronization of categories, presets and schedules.

60% of the participants were given the opportunity to try the system with a gateway connected. The possibility of being able to see the status of adapters from a distant location proved to be a convenient feature to all of these users. However, 66% of the users testing the gateway stated that they only used this feature once.

With exception of various bugs, few problems were observed in the survey. Problems arose for participants who wished to use their Bluetooth connection for other purposes, as they can only have one Bluetooth connection at a time. Another concern was the fact that requests by users for toggling adapters were not delivered successfully to the intended adapter on occasions.

The survey revealed that 40% of the participants were, in some situations, not pleased that their old behaviour of using a wall power switch for turning on their lamps was no longer viable. To toggle an adapter in stressful situations, bringing forth the smartphone, opening the application, and waiting for it to connect to the network, was inconvenient. Furthermore, when returning the prototypes four participants commented, orally, that they had not been able to use the dimmer adapter. They stated that they only used non-dimmable LED lamps in places where they could consider to use the dimmer adapter.

8.2 Discussion

Results showed that there was little use for scheduling, although this might be a bit misleading. For a user to be able to appreciate scheduling it would have to be integrated into the user's everyday life. The testers of our system did, however, only get to use the system for a short period of time, which decreased the opportunities of using the scheduling functionality.

Reception proved to work well, which makes the choice of using a mesh network a success. We believe that reported problems toggling nodes were not due to failures in reception but rather other factors such as transport failures by the SBMN protocol.

The fact that only few participants used the functionality of toggling adapters from a distant location was anticipated. This was never supposed to be frequently used, however it is still as important and would certainly be missed if unsupported.

Problems with users using their Bluetooth connection for something else was anticipated, as described in section 5.4. A person using for example Bluetooth headphones in their home will struggle using this system as they would have to drop the connection to their headphones in order to use the home-automation system. However, this problem can be avoided if a user utilizes the gateway.

The fact that the test group only consisted of five participants may, in our opinion, give a misleading result. The test group is not sufficient to statistically represent the targeted audience. For further development of this product, we believe that the test group should be increased for a more objective result.

9

Results

The preliminary survey revealed several desired features and ideas providing guidance to the development of this project. These ideas include controlling devices from remote locations and expanding home-automation systems with additional devices. The survey also showed that the most prominent opinion was that excessive devices is of great inconvenience to users.

The resulting home-automation system includes two types of adapters that enable non-IoT devices to be automated. Firstly, the power switch adapter, which enables toggling of power to an existing non-IoT device. Secondly, the dimmer adapter, which allows a lamp to be dimmed. These adapters are fitted into a timer casing, to allow an easy installation in standard European power sockets.

To connect the adapters with each other and users, the home-automation system employs Simple BLE Mesh Network (SBMN), a mesh topology Bluetooth Low Energy (BLE) network protocol, developed during this project. SBMN utilizes BLE's advertising functionality to distribute messages using a flooding technique, i.e. by all nodes collectively forwarding messages to their destination. This network protocol is implemented on Texas Instruments' CC2540 BLE microprocessor, which in turn is connected to an adapter. Moreover, each node implements a custom application framework to enable desired system functionality, for example controlling the adapters and network settings. The application framework allows for further modular expansion of applications to meet future needs.

Users of home-automation systems need an interface. This project has developed a graphical user interface in the form of a smartphone application for Android devices. The application presents users to the multitude of features that the system has to offer by dividing them up into multiple fragments. The Adapter fragment has the basic functionality of toggling adapters on and off. The Categories fragment allows users to create custom lists of adapters for easy organization and to toggle multiple adapters at once. In addition to the functionalities of the Categories, the Presets fragment also lets users set adapters in customs lists to be toggled either on or off for various situations. The Schedule fragment is where users can schedule future events, such as turning on lights or a coffee maker.

The possibility of accessing the home-automation system from distant places was one of the desired features suggested by the preliminary survey. To enable this, the home-automation system employs a gateway that connects the SBMN network to the Internet, thus allowing the Android application to control the system whenever connected to the Internet. Moreover, the gateway enables scheduling of events to take place when not at home.

To synchronize user data, such as categories and presets, between several users, as well as provide backup, this system employs a cloud solution. Every user has an account in the cloud to which all their data is saved, enabling a user to log in on multiple devices and still maintain all settings. Furthermore, settings for different networks are synchronized between users, allowing multiple users to co-exist on the same network while maintaining the same state.

In order to evaluate the results of this project, user testing was performed. The results of these tests indicated that the developed system is easy to install and has good range. Moreover, the testers found utilities, such as categories and presets, as well as the synchronization of these utilities to be useful. However, as the system is only a prototype, it does suffer from some flaws, causing crashes or freezes, which were noted by the testers. The testers also noted some areas in which the system failed to meet its purpose. For example, when attaching a ceiling lamp to a power switch adapter, the usual routine of toggling a wall power switch is no longer applicable, and the user must instead use a smartphone. In certain situations, this may be more complex than using a physical power switch.

10

Discussion

The system developed in this project is simply a prototype. This means that it has been developed in consideration of being a concept, rather than a product made for profit. Thus, it does not function flawlessly and has issues.

While the application eliminated the need of excessive control devices, the results of the user tests indicated that it was also a point of inconvenience. Without the possibility of manually turning off adapters with physical light switches, users are forced to use the application, which requires more effort at times. One solution would be to design a separate hardware power switch, included in the system, that could be programmed to toggle an adapter. As a result, the old habits of turning on lights manually would still be applicable.

The results of the user tests included no comments on the range of the system. This leads us to believe that the range is sufficient enough for most apartments less than 80 square meters and that using a mesh topology is a suitable option. However, the results also show that the reception is sometimes not satisfactory, caused by the implementation of the SBMN protocol on the CC2540. As discussed in section 4.4, we believe that this could be solved by using other BLE hardware, and that the SBMN protocol can still be used in a system like this, although it would require more research. However, a fundamental flaw, according to one user, is that while already connected to some other Bluetooth device, the Android application of the home-automation system terminates the existing connection, in favor of connecting to the SBMN network. If this aspect is important to the targeted user, another type of communication technology may be preferred. However, we believe that benefits of using BLE outweighs the benefit of other technologies presented in this thesis.

User responses from the prototype testing showed that dimmer adapters were not particularly useful, as they are not compatible with non-dimmable LED lights. Therefore, other kinds of adapters, for example a curtain opener, may be more usable than a dimmer adapter. However, due to the small size of the test group, further investigation would be required to confirm if non-dimmable LED lamps is a common occurrence.

One aspect overlooked in this project was the support of any established home automation standard protocols. The project instead aimed at investigating technologies and features applicable for a home-automation system. Future work developing this project could involve adapting established standards, in order to make this home-automation system compatible with other systems of the same kind.

In early stages of planning this project, aspects of machine learning and artificial intelligence were excluded. By including such services, the convenience and user friendliness of the home-automation system could potentially increase. For example, if the system could learn the habits of a user, it could predict when the user leaves home and automatically turn off any active device, or reduce electric heating to later on turn it back up. Thus the system would be able to intelligently reduce power consumption and contribute to a more eco-friendly home. Future work on this project could therefore be to introduce AI and introduce an environmental factor to a greater extent.

11

Conclusion

Development of a concept and prototype of a home-automation system for existing devices has been described in this thesis. For the duration of the 14 weeks, the project group has planned, implemented, and tested the produced system according to four goals: user friendliness, scalability, accessibility, and convenience.

The goal of taking a concept of a home-automation system and developing a prototype of it is considered to have been successful to a certain extent. User friendliness has been achieved by offering an Android application, which is used to control the entire system, thus no additional remote controls are required to control the system. Furthermore, the Android application offers several features, such as categories and presets, to facilitate usage that according to a test group of five people were considered to be useful.

Accessibility has partially been achieved, as the system's mesh topology allows a user to control the entire system from anywhere in a house, as long as there is at least one path between each node in the network. However, the concrete implementation of our SBMN protocol suffers from deficiencies in two-way communication, which may cause errors in transmission. Further research is required to confirm if the SBMN protocol could perform better on other BLE devices than the CC2540 used in this project. Moreover, if a gateway is installed, a user can control the entire system from anywhere with an Internet connection.

In theory, scalability has been achieved, as adapters can be added, on demand, to expand the system. Furthermore, both our SBMN protocol and associated application framework allows for extensibility, in terms of adding new features to the system. The concrete implementation of the SBMN network has, however, only been tested with up to five adapters simultaneously. Therefore, further research is required to evaluate its performance for large scale networks. Another factor contributing to scalability is users being able to simultaneously control the same system.

Convenience has partially been achieved. The system enables users to conveniently control power supply to electrical devices through their smartphone, and provides means of scheduling system behavior. Furthermore, synchronization of system settings between users provides convenience in cooperatively using the system. This feature was appreciated by users in the test group. Nevertheless, the system failed to meet the convenience factor in some areas. No consideration was taken to the normal habits of users, which resulted in dissatisfaction when user's habit changed due to using the system. For example, the system forces the user to use the Android application to toggle a ceiling lamp, thus breaking the normal habit of using a wall power switch.

To conclude, many of the features introduced in this home-automation system proved useful in theory, while further research is needed to evaluate performance and usage. The system is also missing some key features in terms of convenience, but overall the project group considers the resulting system a successful model on which further research could be conducted.

References

[1] D. Hendricks, "The History of Smart Homes," http://www.iotevolutionworld.com/m2m/articles/376816-history-smart-homes.htm, (Accessed on 2015-05-14).

[2] L. Kurkinen, "Smart Homes and Home Automation | Berg Insight," http://www.berginsight.com/ReportPDF/ProductSheet/bi-sh2-ps.pdf, (Accessed on 2015-05-14).

[3] D. Bosomworth, "Mobile marketing statistics 2015," http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/, (Accessed on 2015-05-14).

[4] A. B. Brush, B. Lee, R. Mahajan, S. Agarwal, S. Saroiu, and C. Dixon, "Home Automation in the Wild: Challenges and Opportunities," http://research.microsoft.com/pubs/145863/HomeOSCHI_cameraready_Final.pdf, (Accessed on 2015-05-14).

[5] K. Gill, S.-H. Yang, F. Yao, and X. Lu, "A zigbee-based home-automation system," *IEEE Trans. on Consum. Electron.*, vol. 55, no. 2, pp. 422–430, May 2009. [Online]. Available: http://dx.doi.org/10.1109/TCE.2009.5174403

[6] L. Sköld, "Villa vanligaste boendeformen - Statistiska centralbyrån," http://www.scb.se/sv_/Hitta-statistik/Artiklar/Villa-vanligaste-boendeformen/, 04 2014, (Accessed on 2015-05-14).

[7] S. centralbyrån, "Privatpersoners användning av datorer och internet 2014," http://www.scb.se/Statistik/_Publikationer/LE0108_2014A01_BR_IT01BR1402.pdf, 2014, (Accessed on 2015-05-14).

[8] "IDC: Smartphone OS Market Share 2014, 2013, 2012, and 2011," http://www.idc.com/prodserv/smartphone-os-market-share.jsp, (Accessed on 2015-05-14).

[9] "Complete list: Plug, socket & voltage by country - World Standards," http://www.worldstandards.eu/electricity/plug-voltage-by-country/, (Accessed on 2015-05-14).

[10] M. Ruan, J. Shen, and C. Wheeler, "Latching microelectromagnetic relays," *Sensors and Actuators A: Physical*, vol. 91, no. 3, pp. 346 – 350, 2001, proceedings of the Technical Digest of the 2000 Solid-State Sensors and Actuators Workshop. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0924424701006069

[11] "Power Control with Thyristors and Triacs," http://www.nxp.com/documents/application_note/APPCHP6.pdf, (Accessed on 2015-05-14).

[12] "How Photocouplers / Optocouplers Are Used | Renesas Electronics," http://www.renesas.com/products/opto/technology/usage/index.jsp, (Accessed on 2015-05-14).

[13] "Dimming LEDs with Phase-Cut Dimmers: The Specifier's Process for Maximizing Success," http://apps1.eere.energy.gov/buildings/publications/pdfs/ssl/2013_gateway_dimming.pdf, 08 2014, (Accessed on 2015-05-14).

[14] "Guide to dimming low-voltage lighting," http://www.lutron.com/zh-TW/ResourceLibrary/362219.pdf, 2002, (Visited on 2015-05-30).

[15] diy_bloke, "Arduino controlled light dimmer," http://www.instructables.com/id/Arduino-controlled-light-dimmer-The-circuit/, (Accessed on 2015-05-14).

[16] "Electric imp - uart explained," https://electricimp.com/docs/resources/uart/, (Accessed on 2015-05-14).

[17] L. R. Center, "Demonstration and Evaluation of Lighting Technologies and Applications ,."

[18] D. Groth and T. Skandier, "Network+, Study Guide," ftp://ftp.sybex.com/4406/4406ch01pt01.pdf, 2005, (Accessed on 2015-05-17).

[19] F. Kurose and W. Ross, "Computer Networking: a top-down approch," 2013.

[20] E. Ferro and F. Potorti, "Bluetooth and wi-fi wireless protocols: A survey and a comparison," *Wireless Commun.*, vol. 12, no. 1, pp. 12–26, Feb. 2005. [Online]. Available: http://dx.doi.org/10.1109/MWC.2005.1404569

[21] CISCO, "802.11ac: The fifth generation of wi-fi," https://www.cisco.com/c/en/us/products/collateral/wireless/aironet-3600-series/white_paper_c11-713103.pdf, (Accessed on 2015-05-17).

[22] "Rn-131g & rn-131c 802.11 b/g wireless lan module," http://ww1.microchip.com/downloads/en/DeviceDoc/rn-131-ds-v3.2r.pdf, (Accessed on 2015-05-17).

[23] J. Henry, "802.11s mesh networking," https://www.cwnp.com/uploads/802-11s_mesh_networking_v1-0.pdf, (Accessed on 2015-05-17).

[24] P. Singh, M. Kumar, A.K.Jaiswal, and R. Saxena, "Analysis of ZigBee (IEEE 802.15.4 standard) for Star Topology with AODV Protocol," http://www.ijitee.org/attachments/File/v3i1/A0951063113.pdf, (Accessed on 2015-05-17).

[25] ANT+, "What is the maximum range? - this is ant," http://www.thisisant.com/developer/resources/tech-faq/what-is-the-maximum-range/, (Accessed on 2015-05-17).

[26] ——, "Ant+ in mobile - this is ant," http://www.thisisant.com/business/opportunities/mobile/, (Accessed on 2015-05-17).

[27] Bluetooth SIG Proprietary, "Specification of the Bluetooth System," 2014.

[28] "Bluetooth low energy | ble | bluetooth | bluetooth technology website," http://www.bluetooth.com/Pages/low-energy-tech-info.aspx, (Accessed on 2015-05-17).

[29] T. INSTRUMENTS, "2.4-ghz bluetooth® low energy system-on-chip," http://www.ti.com/lit/ds/symlink/cc2540.pdf, (Accessed on 2015-05-17).

[30] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," http://www.mdpi.com/1424-8220/12/9/11734/htm, (Accessed on 2015-05-17).

[31] S. Sunkavai and B. Rarnalmurthy, "MTSF: a fast mesh scatternet formation algorithm for Bluetooth networks," in *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, vol. 6, Nov 2004, pp. 3594–3598 Vol.6.

[32] G. Zaruba, S. Basagni, and I. Chlamtac, "Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 1, Jun 2001, pp. 273–277 vol.1.

[33] C. Petrioli, S. Basagni, and I. Chlamtac, "Configuring BlueStars: multihop scatternet formation for Bluetooth networks," *Computers, IEEE Transactions on*, vol. 52, no. 6, pp. 779–790, June 2003.

[34] "NordicSemiconductor/nRF51-ble-bcast-mesh · GitHub," https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh, (Accessed on 2015-05-17).

[35] T. Snekvik, "nRF51-ble-bcast-mesh/how it works · GitHub," https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh/blob/master/docs/how_it_works.adoc, (Accessed on 2015-05-17).

[36] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network, booktitle = Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking," ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 151–162. [Online]. Available: http://doi.acm.org/10.1145/313451.313525

[37] R. B. Lab, "Ble mini — redbearlab," http://redbearlab.com/blemini/, 2015, (Accessed on 2015-05-17).

[38] Android, "Android 5.0 APIs | Android Developers," https://developer.android.com/about/versions/android-5.0.html, (Accessed on 2015-05-17).

[39] ——, "Activity | Android Developers," http://developer.android.com/reference/android/app/Activity.html, (Accessed on 2015-05-09).

[40] ——, "Activities | Android Developers," http://developer.android.com/guide/components/activities.html, (Accessed on 2015-05-09).

[41] ——, "Fragment | Android Developers," http://developer.android.com/reference/android/app/Fragment.html, (Accessed on 2015-05-09).

[42] ——, "Fragments | Android Developers," http://developer.android.com/guide/components/fragments.html, (Accessed on 2015-05-09).

[43] ——, "Context | Android Developers," http://developer.android.com/reference/android/content/Context.html, (Accessed on 2015-05-09).

[44] ——, "View | Android Developers," http://developer.android.com/reference/android/view/View.html, (Accessed on 2015-05-09).

[45] B. Xu, "General design of control class based on three-tier architecture," in *Software Engineering and Service Science (ICSESS), 2013 4th IEEE International Conference on*, May 2013, pp. 166–169.

[46] Android, "Navigation Drawer | Android Developers," http://developer.android.com/design/patterns/navigation-drawer.html, (Accessed on 2015-05-16).

[47] ——, "List View | Android Developers," http://developer.android.com/guide/topics/ui/layout/listview.html, (Accessed on 2015-05-17).

[48] ——, "ArrayAdapter | Android Developers," http://developer.android.com/reference/android/widget/ArrayAdapter.html, (Accessed on 2015-05-17).

[49] ——, "ToggleButton | Android Developers," http://developer.android.com/reference/android/widget/ToggleButton.html, (Accessed on 2015-05-16).

[50] ——, "SeekBar | Android Developers," http://developer.android.com/reference/android/widget/SeekBar.html, (Accessed on 2015-05-16).

[51] ——, "Action Bar | Android Developers," http://developer.android.com/guide/topics/ui/actionbar.html#Adding, (Accessed on 2015-05-16).

[52] ——, "Switch | Android Developers," http://developer.android.com/reference/android/widget/Switch.html, (Accessed on 2015-05-16).

[53] "alamkanak/Android-Week-View," https://github.com/alamkanak/Android-Week-View, (Accessed on 2015-05-16).

[54] "Android-Week-View/README.md," https://github.com/alamkanak/Android-Week-View/blob/master/README.md, (Accessed on 2015-05-16).

[55] "Create a repeating event - Calendar Help," https://support.google.com/calendar/answer/37115, (Accessed on 2015-05-16).

[56] "Push notifications - drive rest api — google developers," https://developers.google.com/drive/web/push, (Accessed on 2015-05-18).

[57] "Web Services Architecture," http://www.w3.org/TR/2003/WD-ws-arch-20030514/#id2607990, (Accessed on 2015-05-16).

[58] "RESTful Web services: The basics," http://www.ibm.com/developerworks/library/ws-restful/, (Accessed on 2015-05-16).

[59] "IBM Knowledge Center," http://www-01.ibm.com/support/knowledgecenter/SSYJ99_8.0.0/admin/adurlmap_trees.dita, (Accessed on 2015-05-16).

[60] "Extensible Markup Language (XML)," http://www.w3.org/XML/, (Accessed on 2015-05-16).

[61] "JSON," http://json.org/, (Accessed on 2015-05-16).

[62] "What is SOAP (Simple Object Access Protocol)? ," http://searchsoa.techtarget.com/definition/SOAP, (Accessed on 2015-05-16).

[63] L. Noabeb, "Anatomy of a web service: Xml, soap and wsdl for platform-independent data exchange | webreference," http://www.webreference.com/authoring/web_service/index.html, (Accessed on 2015-05-16).

[64] "Node.js," https://nodejs.org/, (Accessed on 2015-05-16).

[65] "PHP: Hypertext Preprocessor," http://php.net/, (Accessed on 2015-05-16).

[66] "Best PHP Framework for 2015 - SitePoint Survey Results," http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/, (Accessed on 2015-05-16).

[67] "Laravel - The PHP Framework For Web Artisans," http://laravel.com/, (Accessed on 2015-05-16).

[68] "What is object-relational mapping (ORM)?" http://searchwindevelopment.techtarget.com/definition/object-relational-mapping, (Accessed on 2015-05-16).

[69] P. Johnston, "Authentication and Session Management on the Web," http://repo.hackerzvoice.net/depot_cehv6/CEHv6%20Module%2018%20Web%20based%20Password%20Cracking%20Techniques/Paul_Johnston_GSEC.pdf, (Accessed on 2015-05-16).

[70] "ASP.NET | The ASP.NET Site," http://www.asp.net/, (Accessed on 2015-05-16).

[71] "Eloquent ORM - Laravel - The PHP Framework For Web Artisans," http://laravel.com/docs/4.2/eloquent, (Accessed on 2015-05-16).

[72] "RFC 6101 - The Secure Sockets Layer (SSL) Protocol Version 3.0," https://tools.ietf.org/html/rfc6101, (Accessed on 2015-05-16).

[73] R. Pi, "Raspberry Pi 1 Model B," https://www.raspberrypi.org/products/model-b/, (Accessed on 2015-05-17).

[74] BeagleBoard.org, "BeagleBoard.org - black," http://beagleboard.org/BLACK, (Accessed on 2015-05-17).

[75] E. Brown, "Survey results: Top 10 hacker SBCs. LinuxGizmos.com," http://linuxgizmos.com/top-10-hacker-sbcs-survey-results/, (Accessed on 2015-05-17).

[76] "Frequently Asked Questions | BeagleBone."

[77] "Frequently Asked Questions | Raspberry Pi," https://www.raspberrypi.org/help/faqs/#powerReqs, (Accessed on 2015-06-01).

[78] L. Foundation, "FAQ | The Linux Foundation," http://www.linuxfoundation.org/about/faq, (Accessed on 2015-05-17).

[79] Laravel, "Queues - Laravel - The PHP Framework For Web Artisans," http://laravel.com/docs/4.2/queues, (Accessed on 2015-05-17).

[80] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," http://www.cs.berkeley.edu/~culler/cs294-f03/papers/bcast-storm.pdf, (Accessed on 2015-05-16).

[81] "Simple-BLE-Mesh-Network | GitHub," https://github.com/sebbe33/Simple-BLE-Mesh-Network/, (Accessed on 2015-05-16).

A - Appendix - Simple BLE Mesh Network

The Simple BLE Mesh network (SBMN) is a network/transport layer protocol that enables mesh networking abilities for Bluetooth Low Energy (BLE). SBMN utilizes BLE advertisements to transfer data amongst nodes, by flooding the network. Each node forwards so called SBMN advertisements, containing a header and a message (under certain conditions, described in A.6) to the rest of the network, allowing a longer range than that of a traditional Bluetooth connection. This appendix describes the specifics of the protocol and how it should be used.

A.1 - GAP Profile

A BLE device running SBMN is required to run a combination of the BLE peripheral-and observer GAP role, in order to enable both advertising and scanning for advertisements.

A.2 - SBMN Advertisement Types

There are four different types advertisements, dictating how an advertisement should be transported. This section further describes these four types in detail.

A.2.1 - Broadcast

If an SBMN advertisement is labeled as a broadcast it will be forwarded and received by every node within the same network. An advertisement with the broadcast type is stateless, meaning than no assurances are made to confirm delivery of the advertisement to any node in a network.

A.2.2 - Group Broadcast

SBMN Advertisements labeled as group broadcast will be forwarded by every node within the same network, but only received by members of the group to which it is addresses (see Header/Destination). Any node can assign itself a member of any group and thus receive group broadcasts addressed to a specific group. Similar to broadcast, advertisements labeled with the group broadcast type is stateless, meaning than no assurances are made to confirm delivery of the advertisement to any node in a within a broadcasting group.

A.2.3 - Stateless

Stateless SBMN advertisements are, in contrast to broadcasts, addressed to a single node, however they are still forwarded by every other node within the same network to reach their destinations. As the name implies, stateless SBMN advertisements are stateless and no assurances are made to confirm delivery of the advertisement the receiver (destination).

A.2.4 - Stateful

A stateful SBMN advertisement is forwarded by every node in a network, however it is only received by one node, its destination. As opposed to every other advertisement type, a stateful SBMN advertisement is stateful, meaning the measures are taken to ensure its

delivery. Upon receiving a stateful SBMN advertisement a node must send an acknowledgement (ACK) back to the sender of the advertisement. If the sender does not receive an ACK within STATEFUL_TIMEOUT, which is a constant defined by the implementor of the SBMN protocol, the same message is sent again. This procedure is repeated for a STATEFUL_RESEND_MAX number of times, defined by the implementor of the SBMN protocol. If still no ACK has been received, the SBMN protocol should give an error, FAILED_DELIVERY, to the application utilizing the protocol.

A.3 - Header

The first bytes of a SBMN advertisement contains a header with information about how to transport the advertisement. This section further describes the data structure of the SBMN header.

A.3.1 - Network Identifier - 2 bytes

The first two (2) bytes consist of a network identifier, which is required in every header. Every node belongs to one network, represented by this network identifier. The network identifier may take any value [1, 65534], however the identifiers 0 (hex: 0x0000) and 655355 (0xFFFF) are reserved as a universal broadcast networks. The universal broadcast network identifiers should only be used temporarily to establish contact between nodes belonging to different networks.

A.3.2 - Source - 2 bytes

The third and fourth bytes contains the node identifier, which is required in every header. This node identifier represents the node in a network, that sent the message, and should be unique within said network. Further, this identifier may take any value [0, 65535].

A.3.3 - Length - 5 bits

The first five (5) bits of the fifth byte contains the length of the message (i.e. the advertisements excluding the header), which is required in every header.

A.3.4 - Advertisement Type - 3 bits

The last three (3) bits of the fifth byte contains the message type, which is required in every header. The advertisement type may take any value [0, 4] representing one of the advertisement types described in section A.2, or the stateful ACK.

0	Broadcast
1	Group Broadcast
2	Stateless
3	Stateful
4	Stateful ACK

Table 1: Advertisement bits.

A.3.5 - Sequence Number - 1 byte

The sixth (6) byte contains the sequence number of an SBMN advertisement and is required in every header. The sequence number represents the sequential order of which a SBMN

advertisement was sent from a source; e.g. if one SBMN advertisement has a sequence number 100, then the next one send from the same source will have a sequence number 101. The sequence number may take any value [0, 255] and is unique within a certain timespan. When a node has sent 256 advertisements (the first seq. number is 0), the sequence number counter will flip and the next sequence number will be 0. Being unique within a certain timespan, the sequence number is used when sending ACKs of a stateful message, and to control forwarding of SBMN advertisements (see section A.2.4).

A.3.6 - Destination - 2 bytes

The seventh (7) and eighth (8) bytes contains the destination, however a destination is optional depending on the advertisement type. If the advertisement is a broadcast, destination will not be included. For every other advertisement type the destination is included. The destination may take any value [0, 65535] and represents the node or broadcast group to which the SBMN advertisement should be delivered.

A.4 - SBMN Module

The SBMN protocol should be implemented in such a way that the physical transmission of an advertisement is handled by the BLE stack. Consequently, the SBMN implementation, from here on out referred to as the SBMN module, should not have dependencies to any BLE specific functionality or any other platform specific code. By having no such dependencies, the SBMN module is applicable for unit testing. To achieve non-dependencies the SBMN module should upon initialization, be given callbacks (for example function pointers in C, or a callback interfaces in Java). Thereby, these callbacks can be switched for a mock implementation during testing. Furthermore, upon initialization the SBMN module should be given the network identifier and the node identifier which will be used when construction headers of SBMN advertisements.

```
void initializeMeshConnectionProtocol(uint16
    networkIdentifier,uint16 deviceIdentifier,
     advertiseDataFunction dataFunction,
     onMessageRecieved messageCallback,
     getSystemTimestampFunction timestampFunction,
     randomFunction randFun,
     cancelAdvertisementDataFunction
         cancelDataFunction);
```

Listing 1: Initialization function.

A.5 - Sending SBMN Advertisements

When sending a SBMN advertisement the SBMN module requires the following information: advertisement type, the message to be sent, and optionally a destination. The SBMN module constructs the header according to these parameters, and adds the additional header data according to its current state, i.e. it adds the network identifier, node identifier, and the current sequence number + 1. Further the SBMN module concatenates the message with the header to form the SBMN advertisement.

To send the advertisement, the SBMN modules relies on the physical layers of the BLE stack to perform the advertising. Furthermore, the SBMN module requires the user of the module to implement an advertisement queue, and provide access to that queue. The advertisement queue is used by the SBMN to enqueue advertisements which are about

to be sent. A queue is needed, since while the BLE stack already is advertising, the SBMN module may receive a request to send another message, and thus the advertising of that message needs to take place only when the advertising of the current i finished. Furthermore, the advertisement queue needs to implement an option to specify a delay. If a delay is specified, the advertisement may not be advertised before the time of delay has run out. Moreover, the SBMN requires every advertisement it enqueues be sent within a reasonable amount of time, with the exception of the queue being full, in which case the advertisement can simply be dropped.

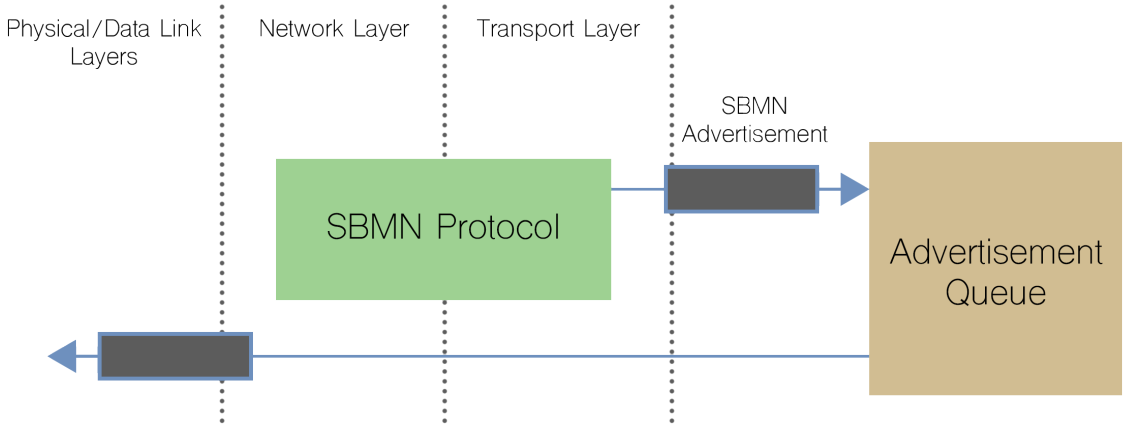


Figure 22: Illustration of enqueuing a SBMN advertisement.

When advertising SBMN advertisements, the SBMN protocols requires the user of the protocol to implement an advertising timeout, as illustrated by figure 23. The advertising timeout must be at less than 250 ms, resulting theoretical throughput of at least 92b/s over a single link. The advertising interval, scanning interval and scanning window may be set to best fit the device on which SBMN is implemented.

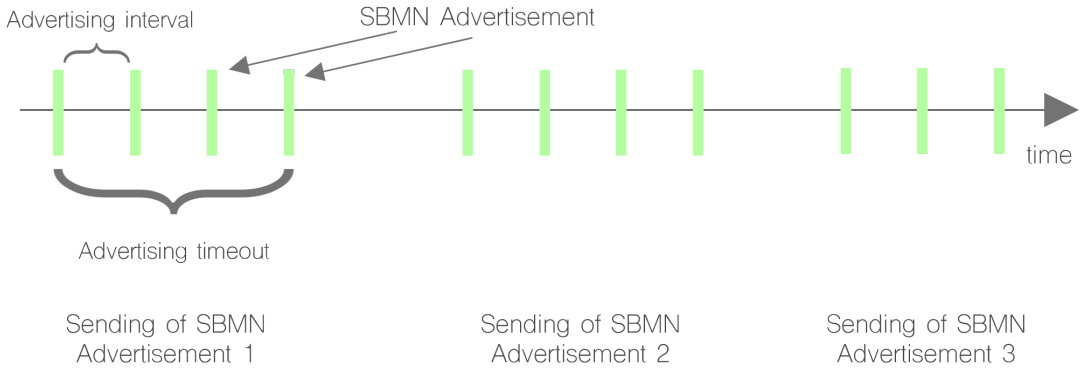


Figure 23: Illustration of the advertising.

In the case of sending a stateful message, the message itself, in addition to the time of enqueuing, sequence number and destination is saved by the SBMN module. If an ack has not been received within STATEFUL_TIMEOUT, the saved message is resent, by the SBMN module, with associated details. However, the resent message does use a new sequence number, and the same as the one associated with the saved message. This procedure is repeated for STATEFUL_RESEND_MAX number of times, before the delivery of the stateful message is considered a failure.

A.6 - Processing SBMN Advertisements

Whenever any incoming BLE advertisement is discovered, it shall be forwarded to the SBMN module, which in turns processes the advertisement. If the advertisement is not a valid SBMN advertisement, or addressed to another network, it will be disregarded. Otherwise, the SBMN advertisement will be processed according to the following rules:

- If a SBMN advertisement from the same source, with the same sequence number, has already been processes within PROCESS_ ADV_REMOVAL_TIME milliseconds, it will be disregarded.
- If the SBMN advertisement is addressed to the node, or any broadcast group to which the node is a member of, or if it's type is broadcast, the message of the SBMN, along with the source, will be sent to the user of the protocol.
- If the advertisement type of the SBMN advertisement is broadcast, or group broadcast, it will be applicable for forwarding.
- If the advertisement is not addressed to the node, it will be applicable for forwarding.

A SBMN advertisement being applicable for forwarding, means that it will be forwarded under the dictation of Counter-scheme, developed by Ni et al. [80], to prevent a Broadcast storm. The Counter-scheme employs a random back-off mechanism before a message is forwarded, and during that wait counts how many instances of the same message is detected. If that count c is greater than a threshold value C, the forwarding of said message is canceled. For further details regarding Counter-scheme, please refer to The Broadcast Storm Problem in a Mobile Ad Hoc Network by Ni et al. Since SBMN module employs Counter-scheme, a random number generating function is required to be accessible. Through this a random back-off time is generated as rand() * BACKOFF_INTERVAL, where BACKOFF_INTERVAL is the average time between an SBMN advertisement is detected, to when it is being forwarded. The back-off time is always sent as the delay parameter to the advertisement queue when forwarding messages. Furthermore the SBMN modules saves the amount of times each identical advertisement have been processed. If that amount is greater than FORW_COUNTER_THRESHOLD, it will be removed from the advertisement queue, and thereby not forwarded.

A.7 - Implementation in C

An implementation of the SBMN module is available in the programming language C [81]. This section gives a brief description of the interface of the SBMN module, and the constants used.

A.7.1 - SBMN Module Interface

The following functions are available in the C implementation of the SBMN protocol.

void initializeSBMNProtocol(uint16 networkIdentifier, uint16 nodeIdentifier, enqueueAdvertisementFunction enqueueFunPointer, dequeueAdvertisementFunction dequeueFunPointer, onMessageRecieved messageCallback, getSystemTimestampFunction timestampFunction, randomFunction randFun);	
This function initializes the SBMN modules and must be called before any other function in the SBMN module.	
Parameter	Description
networkIdentifier	The identifier of the SBMN network
nodeIdentifier	The identifier of the node
enqueueAdvertisementFunction	A pointer to a function to be called when enqueuing an advertisement, in the queue implemented by the user of the SBMN protocol.
onMessageRecieved	A pointer to a callback function which is called when an incoming SBMN advertisement is addressed to the node. Through this callback, the message of the SBMN advertisement is delivered.
getSystemTimestampFunction	A pointer to a function which return the current time system time (in unix format)
randomFunction	A pointer to a function which returns a randomly generated number

Table 2

void processIncomingMessage(uint8* data, uint8 length);	
To be called when an advertisement has been discovered during a scan. Processes the advertisement according to the rules of the SBMN protocol.	
Parameter	Description
data	The data of the advertisement
length	The length of the data

Table 3

void broadcastMessage(uint8* message, uint8 length); Sends a SBMN advertisement (broadcast type), containing a message, addressed to every node within the network.	
Parameter	Description
message	The message to be sent
length	The length of the message

Table 4

void broadcastGroupMessage(uint16 groupDestination, uint8* message, uint8 length)	
Sends a SBMN advertisement (broadcast group type), containing a message, addressed to every node within the specified group, within the network.	
Parameter	Description
groupDestination	The identifier of the group to which the message shall be sent.
message	The message to be sent
length	The length of the message

Table 5

void sendStatelessMessage(uint16 destination, uint8* message, uint8 length)	
Sends a SBMN advertisement (stateless type), containing a message, addressed the specified node.	
Parameter	Description
destination	The identifier of the node that shall receive the message.
message	The message to be sent
length	The length of the message

Table 6

void sendStatefullMessage(uint16 destination, uint8* message, uint8 length)	
Sends a SBMN advertisement (stateless type), containing a message, addressed the specified node.	
Parameter	Description
destination	The identifier of the node that shall receive the message.
message	The message to be sent
length	The length of the message

Table 7

uint8 joinGroup(uint16 groupId);	
Adds the node as a member of the specified group. Returns 1 if successful (enough memory and not already a member of the specified group) or 0 otherwise	
Parameter	Description
groupId	The identifier of the group the node shall become a member of.

Table 8

uint8 leaveGroup(uint16 groupId);	
Removes the node, as a member, from the specified group. Returns 1 if successful (if the node was a member and successfully removed) or 0 otherwise	
Parameter	Description
groupId	T he identifier of the group the node shall be removed from.
void destructMeshConnectionProtocol();	
To be called when use of the SBMN protocol is finished. Releases any allocated resources.	

Table 9

void periodicTask();
To be called periodically by the user of the protocol. In this function the SBMN clears processed messages and executes resending of unACK'ed stateful SBMN advertisements. This function must be called at least once every PROCESS_ADV_REMOVAL_TIME or STATEFUL_TIMEOUT milliseconds, whichever is set to be the lowest.

Table 10

A.7.2 - Constants

STATEFUL_TIMEOUT	
Value	Description
5000 ms	The maximum time to wait before resending a stateful SBMN advertisement, if no ACK for it has been received.

Table 11

STATEFUL_RESEND_MAX	
Value	Description
3	The number of times to resend a stateful message before before notifying the user of the SBMN protocol that the transmission was unsuccessful.

Table 12

FAILED_DELIVERY	
Value	Description
240 (0xF0)	The error code for failed delivery

Table 13

PROCESS_ADV_REMOVAL_TIME	
Value	Description
3000 ms	The amount of time before a processed message is removed from the set of processed messages.

Table 14

BACKOFF_INTERVAL	
Value	Description
30 ms	The average time between an SBMN advertisement is detected, to when it is being forwarded

Table 15

A.8 - Mesh GATT Service

The SBMN mesh GATT service handles communication between a central and a network with the use of two characteristics; one for writing data to the network(TX), and one for receiving data from the network(RX). Whenever a central has written a message to the TX characteristic, the node in the SBMN network forwards the message to its destination. The central can also subscribe to the RX characteristic, in which any incoming response from the network will be written.

The message written to the TX characteristic must have a header, composed of 4 bytes. The first byte represents the type of the SBMN advertisement , ie stateful, stateless, broadcast or group broadcast. If the type is not broadcast, the following two bytes provides the destination to which the message is to be sent. The last byte of the header contains the application id that should process the message when delivered.

Data received by the RX characteristic consist of a 3 byte header followed by the message. The first two bytes in the header identifies the source of the incoming message and the last byte provides information about which application that should process the message.

Appendix B - Node Applications

The SBMN protocol described in Appendix A has the corresponding functionality of the transport and network-layers in the OSI-model. To achieve modularity, in accordance with the OSI-model, different tasks required to be performed by a node in a SBMN network are split into applications. In addition, every node in a SBMN network implements an application handler, that forwards incoming messages to their respective applications. The following appendix describes the node applications, as well as the node application manager developed during this project.

B.1 - Node Application Manager

Every node application has a one (1) byte identifier which may take any value [0,255]. This identifier is used by the application manager to route incoming messages to their corresponding applications. The first byte in each message sent to the SBMN protocol must therefore contain the application identifier of the application it was sent from. Furthermore, each application must be registered in the application manager's routing table, in order to receive any incoming messages. Every application must also implement a function which can be called by the application manager, through which an incoming message is delivered. For example, in C this function is defined as:

```
typedef void (*applicationProcessMessageFunction)(uint16 destination,
    uint8* data. uint8 length);
```

Listing 2: C function, incoming message is delivered.

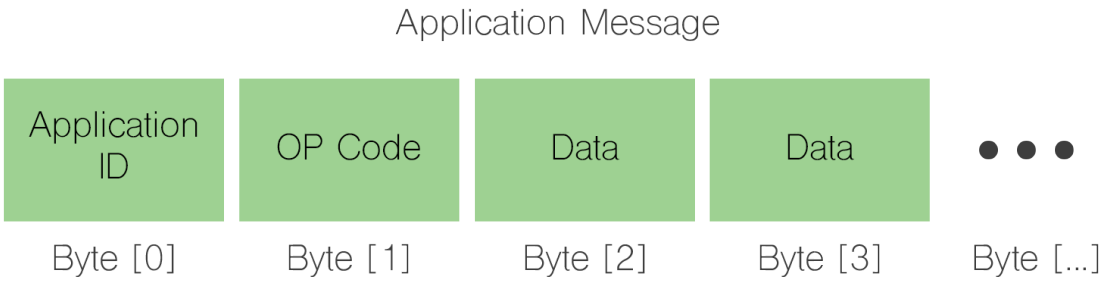


Figure 24: The common structure of an application message.

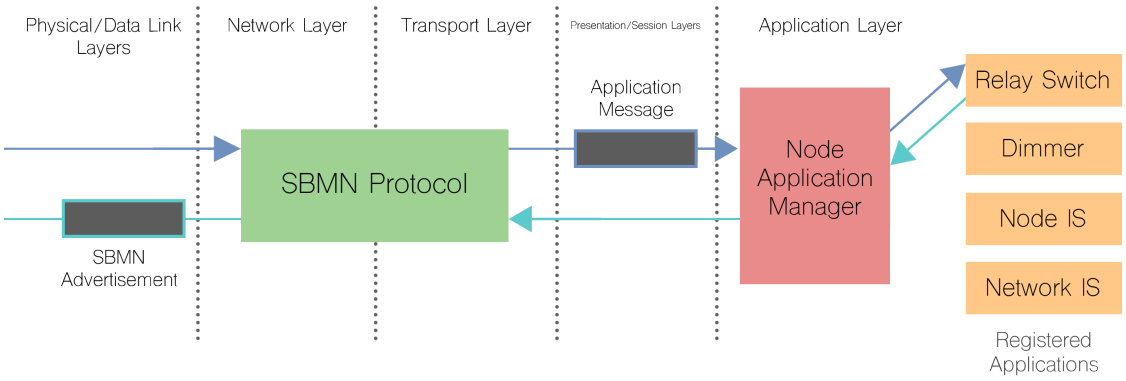


Figure 25: The lifecycle of a communication with SBMN and Node Applications.

B.2 - Relay Switch Application

The relay switch application is intended to control the status of a relay connected to the microcontroller on which the application is run. The identifier of the of the relay switch application is 1 (0x01). The following tables describes the operation codes (opcodes) of the relay switch application.

OP code	Value	Description
RELAY_SWITCH_STATUS_CHANGE	0x01	Sets the status of the relay. The byte following the OP code determines the status. If 0, the relay will be shut off. If $>$ 0, the relay will be turn on.
RELAY_SWITCH_STATUS_GET_REQUEST	0x02	A request for the status of a relay. Contains no data following the OP code.
RELAY_SWITCH_STATUS_GET_RESPONSE	0x03	A response to a RELAY_SWITCH_STATUS_GET_REQUEST. The byte following the OP code contains the status. If 0, the relay is shut off. If $>$ 0, the relay is turned on.

Table 16: Describes the operation codes of the relay switch application.

B.3 - Dimmer Application

The dimmer application controls a dimmer, connected to the microcontroller, via UART. The identifier of the of the dimmer application is 2 (0x02). The following table describes the opcodes available for the dimmer application.

OP code	Value	Description
DIMMER_SET_DIM_VALUE	0x01	Sets the dim value. The byte following the OP code is the dim value to be set [0,255]. 0 being shut off, 255 full brightness.
DIMMER_GET_DIM_VALUE_REQUEST	0x02	A request for the dim value. Contains no data following the OP code.
DIMMER_GET_DIM_VALUE_RESPONSE	0x03	A response to a DIMMER_GET_DIM_VALUE_REQUEST. The byte following the OP code contains the current dim value.

Table 17: Describes the operation codes available for the dimmer application.

B.4 - Node Information Application

The node information application handles information about the node, mainly its name and the of service it provides (currently either relay switch or dimmer).

OP code	Value	Description
NODE_INFORMATION_GENERAL_ INFO_REQUEST	0x01	A request for general information about the node. Contains no data following the OP code.
NODE_INFORMATION_GENERAL_ INFO_RESPONSE	0x02	A response to a NODE_INFORMATION_GENERAL. _INFO_RESPONSE The byte following the OP code contains the application id of the service the node provides (0x01 for relay switch, 0x02 for dimmer). Following the service byte, the current status of the service that the node provides is included (status of relay or dim value). Furthermore, the name of the node is includes encoded in ASCII. If the name is less than 17 characters long, the first byte following the status byte is the length of the name. If the name is 17 characters long, the first character of the name follows the status byte.
NODE_INFORMATION_GET_NAME_ REQUEST	0x03	A request for the node name. Contains no data following the OP code.
NODE_INFORMATION_GET_NAME_ RESPONSE	0x04	A response to a NODE_INFORMATION_GET_NAME_REQUEST. The byte following the OP code contains the length of the name If the name is less than 17 characters long. However, if the name is 17 characters long, the first character of the name follows directly. Furthermore, the name of the node is includes encoded in ASCII.
NODE_INFORMATION_SET_NAME	0x05	Sets the name of a node. The first byte following the OP code is regarded as the length of the name. The following bytes as the characters of the name, encoded in ASCII.

Table 18: Describes the operation codes of the node information application.

B.5 - Network Information Application

The network information application handles information about the network a node is connected to, mainly its name.

OP code	Value	Description
NETWORK_INFO_SET_NAME	0x01	Sets the name of a network. The first byte following the OP code is regarded as the length of the name. The following bytes as the characters of the name, encoded in ASCII. Note. A message changing the network name should be as a broadcast, allowing every node in a network to change the name simultaneously.
NETWORK_INFO_GET_NAME_REQUEST	0x02	A request for the node name. Contains no data following the OP code.
NETWORK_INFO_GET_NAME_RESPONSE	0x03	A response to a NETWORK_INFO_GET_NAME_REQUEST. The byte following the OP code contains the length of the name, followed by the characters of the name encoded in ASCII.
NETWORK_INFO_PAIRING_INFORMATION	0x04	A message with this OP code is an invitation to join the network of the sender. Following the OP code the next to bytes is the network identifier of the network to join. The two bytes after that is an available node identifier in said network.
NETWORK_INFO_PAIRING_RESPONSE	0x05	A response that a new node wants to join the network that this node offered the details of. The two bytes following the OP code is the node identifier of the new node.
NETWORK_INFO_PAIRING_ADDITIONAL_INFO	0x06	Sent by a node which received a NETWORK_INFO_PAIRING_RESPONSE, this message contains additional information of the network. The byte following the OP code is the length of the network name, and the following bytes is the characters of the network name encoded in ASCII.

Table 19: Describes the operation codes of the network information application.

Appendix C - The SBMN Advertising Experiment

In order to optimize the advertising and scanning settings on a Texas Instrument CC2540 BLE microprocessor, an experiment was set up. The experiment consisted of five CC2540s, three receivers in range of one sender, and one out of range, as illustrated by figure 26.All original advertisements sent in the experiment were sent only from one sender.Upon receiving an advertisement, every node logged the time and the sequence number of that advertisement. The experiment was performed three times, testing different factors, the specifics and results of which is described later in this appendix. Furthermore, the experiment was not conducted in an environment free from other BLE advertisements, which may have affected the outcome of the results.

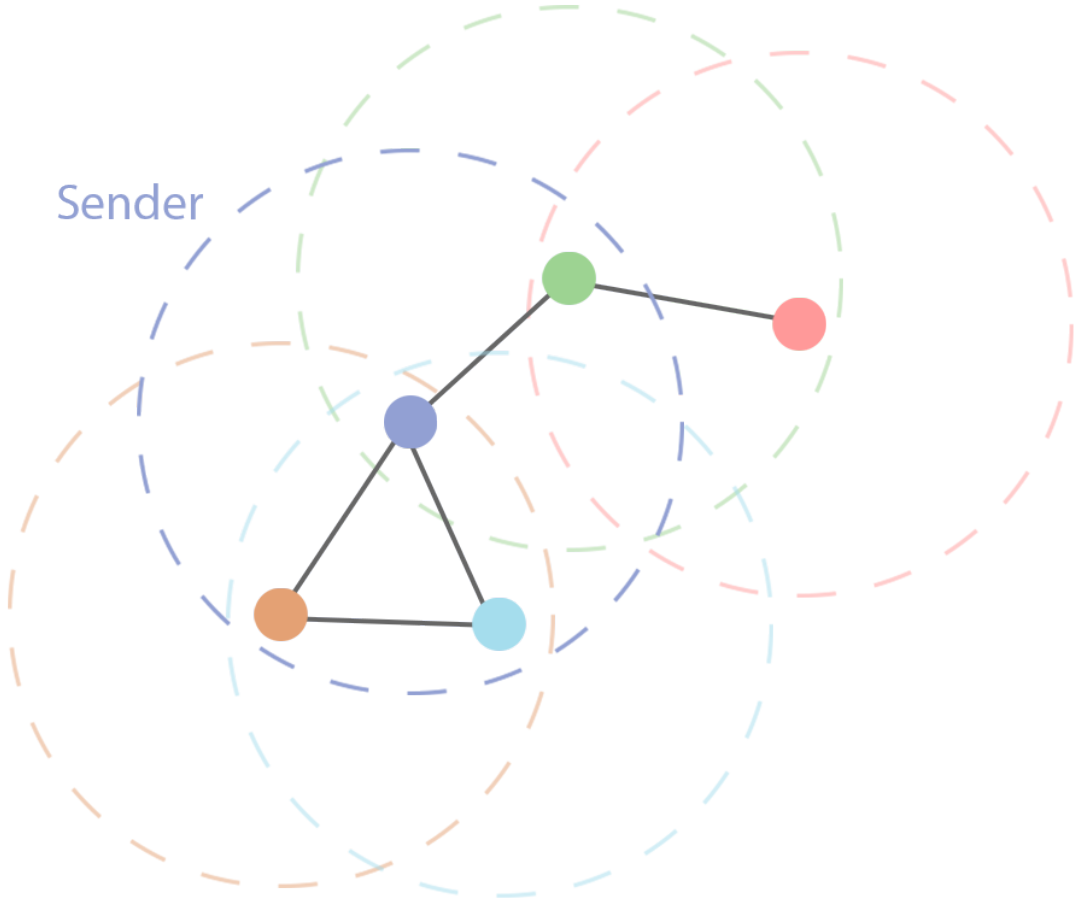


Figure 26: The setup of the SBMN advertising experiment.

C.1 - One Way Communication

In the first experiment, the sender sent a series of 6 SBMN advertisements to every node for every setting. No responses were sent. 10 different settings were tested, varying in advertising interval and timeout, and scanning interval and timeout.

C.1.1 - Results

Setting	Scanning window	Scanning interval	Advertising interval	(Reception timeout	Average received (%) rate)
A	30	35	35	85	98
B	30	35	35	170	100
C	30	35	30	100	97
D	30	35	30	65	99
E	30	35	20	65	98
F	16	16	35	85	100
G	16	16	35	170	100
H	16	16	30	100	100
I	16	16	30	65	98
J	16	16	20	65	97

Table 20: The result of the first advertising experiment.

C.1.2 - Discussion and Conclusion

As shown by the results, the average amount of received advertisements (reception rate) was high for each setting. In our opinion, the missed advertisements is due to other circumstances, such as a crowded bandwidth at the time of execution. To conclude, the result of this experiment seems to show that of these settings are suitable, although a longer advertising timeout seems to increase the reception rate.

C.2 - Two Way Communication

In this experiment, the sender sent 4 SBMN advertisements to every node for each setting, and for every advertisement received a receiving node sent a response to the sender. The amount of responses received were logged by the sender. The same 10 settings as in the one way communication experiment were tested.

C.2.1 - Result

Scanning window	Scanning interval	Advertising interval	Advertising timeout	(Reception rate)	Average received
A	30	35	35	85	86
B	30	35	35	170	60
C	30	35	30	100	69
D	30	35	30	65	83
E	30	35	20	65	73
F	16	16	35	85	65
G	16	16	35	170	51
H	16	16	30	100	49
I	16	16	30	65	58
J	16	16	20	65	59

Table 21: The result of the second advertising experiment.

C.2.2 - Discussion and Conclusion

The results show a heavy decrease in average amount of received advertisements, compared to the one way communication experiment. Furthermore, it can be seen than the settings having a long advertising timeout have a very low reception rate, as opposed to other settings having the same scanning window and interval. The reasons for this seems to be that a response is sent while the sender is still advertising. Thus the sender is busy performing other tasks and the scanning is not prioritized by the operation system on the CC2540. However, settings D and E deviates from this pattern. They have a lower advertising timeout than setting A, but a lower reception rate as well. Therefore, we believe that using low advertising interval (20-30 ms) begins to have a negative impact on the reception rate, probably because it causes more work for CC2540's operating system, thus reducing the scanning time.

Moreover, the settings F-J have lower reception rate than the settings A-E. The settings F-J uses a very low scanning window and scanning interval, which seems to be causing the low reception rate. We believe that using a low scanning window and interval, causes a lot of overhead in the operating system of the CC2540, i.e. that starting a new scanning interval requires a certain amount of work, and if started more frequently this amount of work takes up time which otherwise could be used to scan.

To conclude, based on the results of the experiment setting A yielded the best reception rate, since it has a low advertising interval, relatively high scanning interval and advertising interval.

C.3 - Two Way Communication While In Connection

Based on the result of the two way communication experiment, another experiment, using the same setup was conducted to evaluate the reception rate when the sender is also in a connection. A central device (LG Nexus 5) initiated a connection with the sender, upon which the sender advertised to and received response from the receivers. This was performed 7 times, with the sender setting a different desired connection each time. While in a connection, BLE allows for a minimum advertising interval of 100 ms [27, p. 968]. Thus the advertising interval and the advertising timeout, while in a connection, are set to 100 ms and 205 ms respectively to get the shortest possible advertising timeout, but still allowing for at least two advertisements to be sent.

C.3.1 - Result

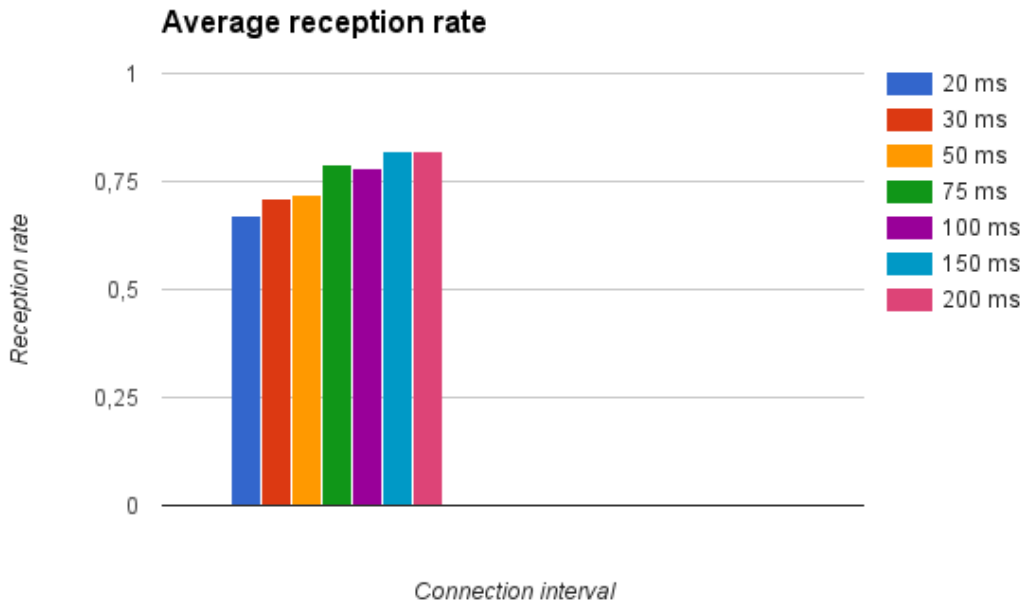


Figure 27: The setup of the SBMN advertising experiment.

C.3.2 - Discussion and Conclusion

Compared to the results of the two way communication experiment, the reception rate decreased for every setting. The highest reception rate was 82% when using a desired connection interval of 100 or 200 ms. We believe that using high connection intervals contributes to less overhead in the CC2540's operating system, and therefore results in a better average reception rate. However, using such high connection intervals contributes to a noticeable delay, from when a central sends data to the sender, to when that data is perceived by the sender. Therefore, we concluded that a desired connection interval of 75 ms, which results in an average reception rate of 79%, is deemed sufficient, with regards to such a delay.

Appendix D - Power Switch Adapter Schematic

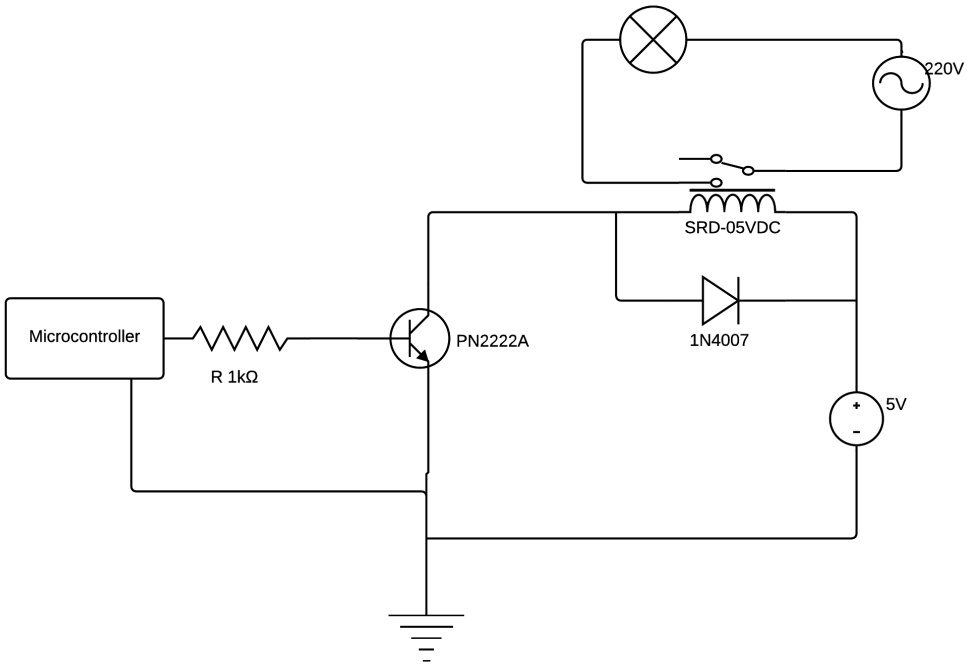


Figure 28: The schematic of the power switch adapter.

Appendix E - Preliminary survey

Preliminary Survey

*Obligatorisk

What type of household do you live in? *

- House
- Apartment

Do you have a Home Automation system at home? *

- Yes
- No

What is the average distance between your electrical appliances and your normal geographical position in your home? *

- 1-10m
- 11-20m
- > 20m

What is the longest distance between your electrical appliances and your normal geographical position in your home? *

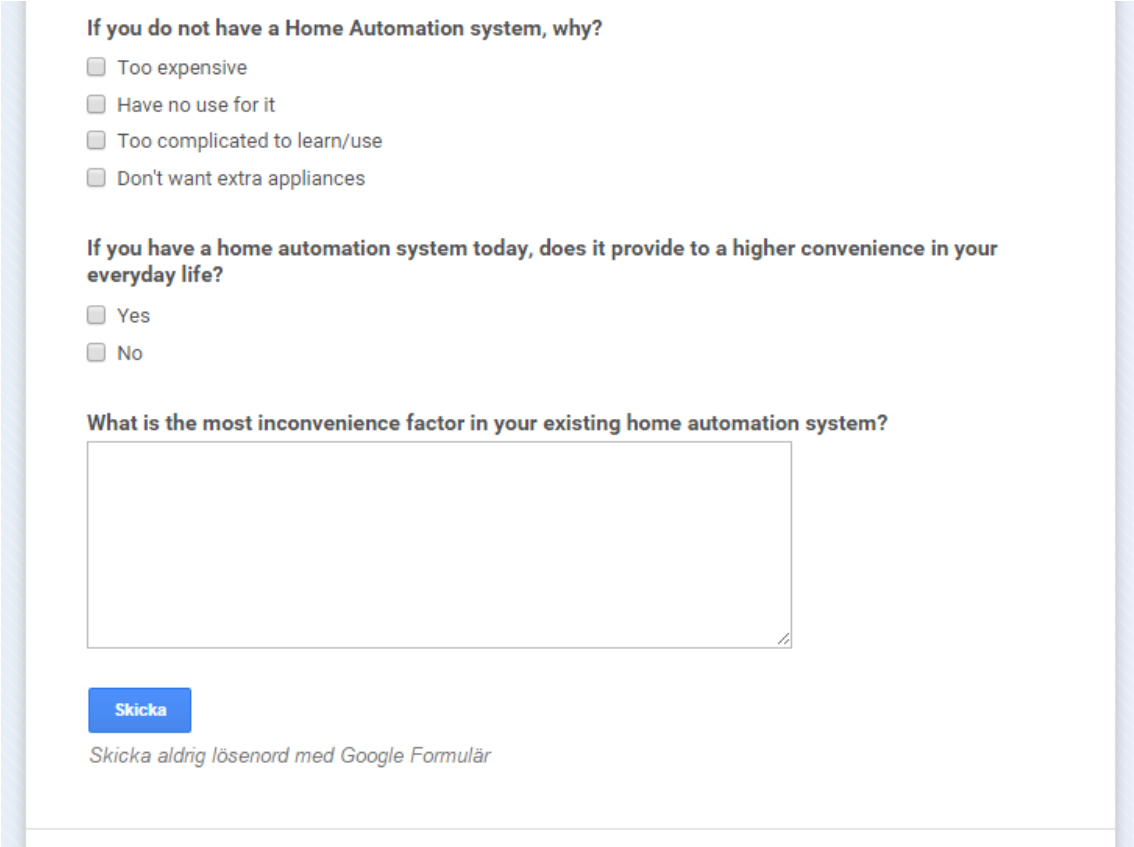
- 1-10m
- 11-20m
- > 20m

What is the longest distance between your nearest electrical appliance and your normal geographical position in your home?

- 1 -10m
- 11-20m
- > 20m

Which of these features do you think would be useful in a Home Automation system? *

- Control home from outside of home
- Option to turn something on/off on specific time
- Option to turn something on/off depending on environmental variables
- Option to turn something on/off depending on location



https://docs.google.com/forms/d/16t6k6oWqEbktKVl_L0yoLnsmzvITHlqiewIoGtT6v7o/viewform?usp=send_form

Result

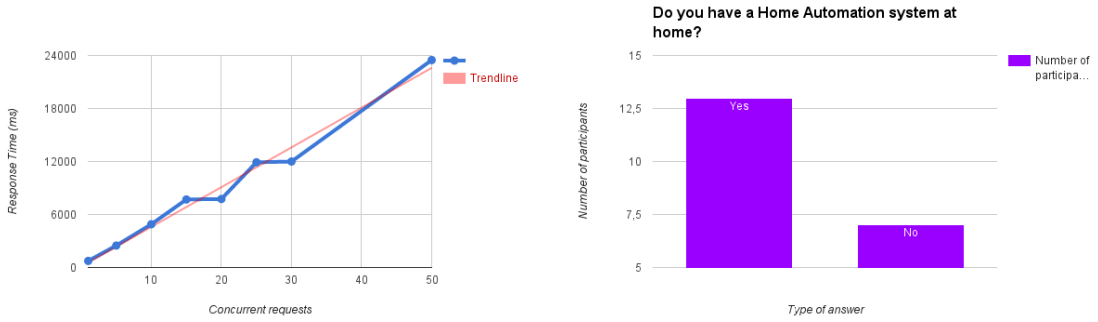


Figure 29: Questions one and two. Authors own graph.

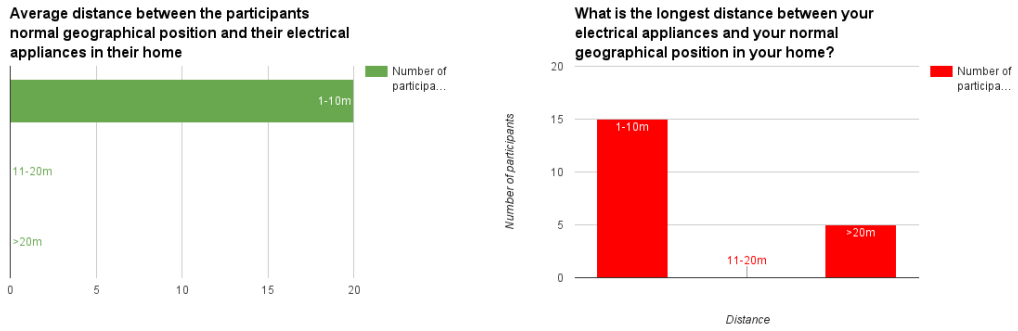


Figure 30: Questions three and four. Authors own graph.

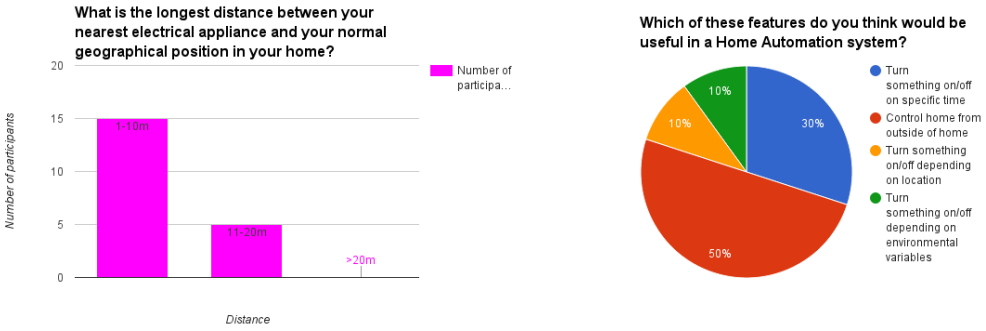


Figure 31: Questions five and six. Authors own graph.

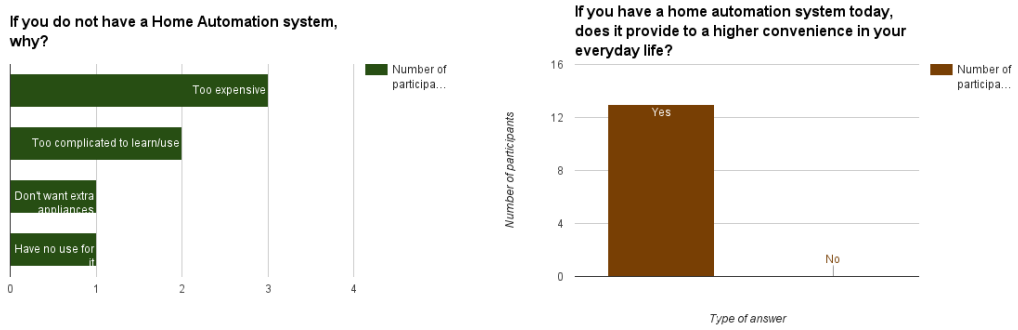


Figure 32: Questions seven and eight. Authors own graph.

What is the most inconvenience factor in your existing home automation system?
To expensive to buy more of the divices to control.
That you have to have an extra remote control
That it is very difficult to install. I would like to be able to do it myself
I would like it to be more automated. Today I can only switch my lamp on/off. I would like to be able to do more with it. Actions happening at specific time sound great!
The expenses you have to lay
I can only control my lamps with the remote control. So from outside would be cool
I do not like that I have to replace divices I already have, to be able to control them
I would like to control my coffe machine
Would like to see if anything was left on while Im at work
I always forget where the remot control are
I would like my already bought electronics to be able to be controled, not to buy new appliances
The money it costs to expand
Hard to learn how to use it

Figure 33: Questions nine. Authors own graph.

Appendix F - User tests

User Test Survey

Thank you for helping us test our product. Please give us your opinion about it by answering this survey.

*Obligatorisk

What kind of building do you live in? *

- House
- Apartment

How large is your home? *

Answer in square meters

Have you used a Home Automation system before? *

- Yes
- No

On a scale of 1-10, how easy was the application to use? *

	1	2	3	4	5	6	7	8	9	10	
Difficult	○	○	○	○	○	○	○	○	○	○	Easy

Did you find the category functionality useful? *

- Yes
- No
- Övrigt:

Did you find the preset functionality useful? *

- Yes
- No
- Övrigt:

Did you find the scheduling functionality useful although it may be in a preliminary stage??

- ○ Yes
- ○ No
- ○ I did not use it
- ○ Övrigt:

Did you use more than one phone connected to the system, if so did you find the sharing of categories/presets/schedules useful?

- ○ Yes
- ○ No
- ○ Övrigt:

What was the average distance between the adapters you tested? *

(Average distance)

Did you find any problem with the reception? *

If so, describe the situation

How easy was the system to install? *

	1	2	3	4	5	6	7	8	9	10	
Difficult	○	○	○	○	○	○	○	○	○	○	Easy

Comment on installing the system *

Did you find the system to be more convenient than your normal setup? *

If no, please describe why.

Did you feel that something was missing from the product? *

If yes, please describe what.

Did you experience any kind of problem with the product? *

If yes, please describe the problem.

Overall, how was the experience of using our product? *

	1	2	3	4	5	6	7	8	9	10	
Bad	○	○	○	○	○	○	○	○	○	○	Good

If you used the gateway, did you find it useful to know the status of your adapters from distant locations?

- ○ Yes
- ○ No
- ○ I did not use it
- ○ Övrigt:

Did you ever toggle the status of adapters from distant locations, if so how was your experience? *

Skicka

Skicka aldrig lösenord med Google Formulär

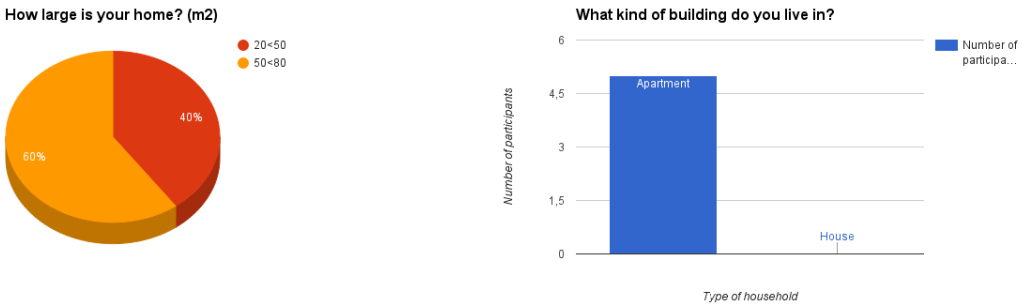


Figure 34: Questions one and two. Authors own graph.

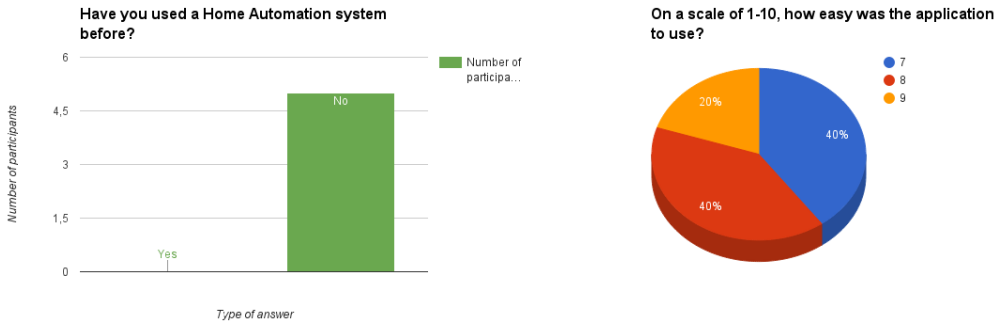


Figure 35: Questions three and four. Authors own graph.

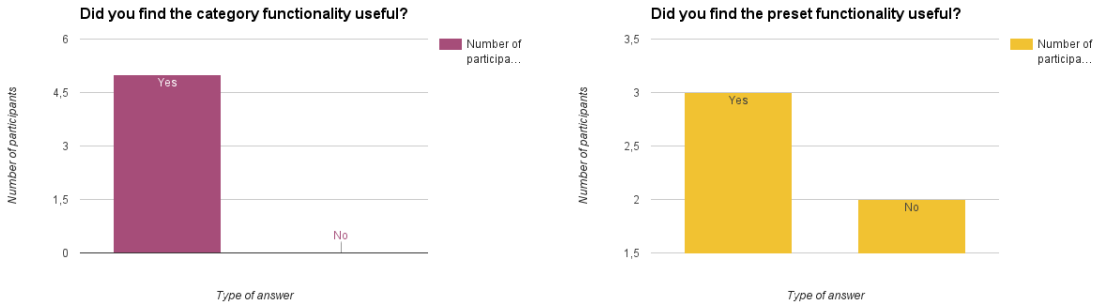


Figure 36: Questions five and six. Authors own graph.

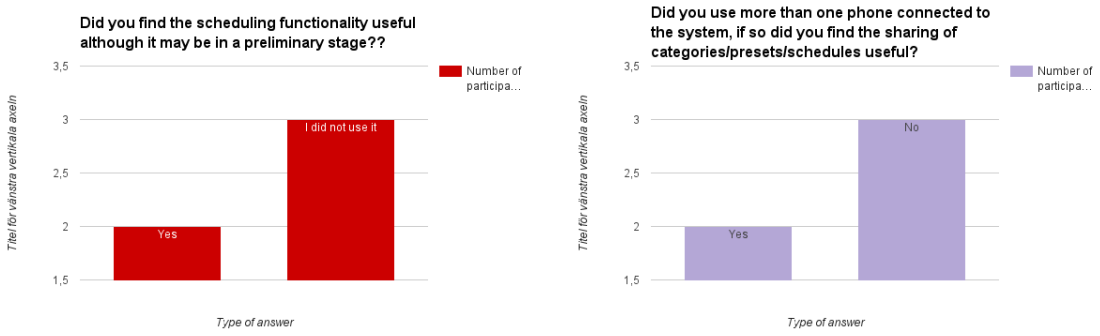


Figure 37: Questions seven and eight. Authors own graph.

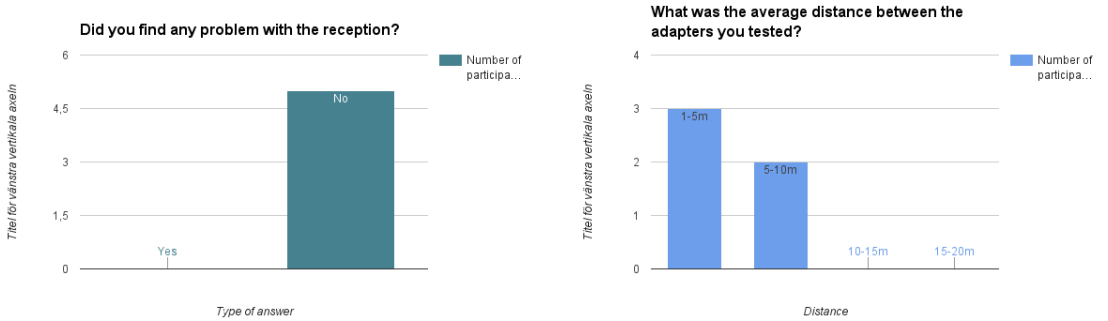
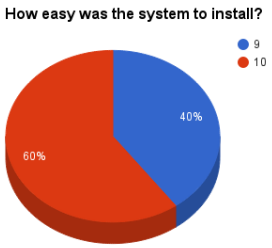


Figure 38: Questions nine and ten. Authors own graph.



Comment on installing the system
I didn't quite get the syncing process first, due to lack of description. However, once I got it, it was very easy
No problems at all
Easy
I would have liked to have more information to set up the system.
No comment

Figure 39: Questions eleven and twelve. Authors own graph.

Did you find the system to be more convenient than your normal setup?
Both yes and no. It was convenient to control my lamps when lying in bed, or in the sofa. However, I attached one adapter to my ceiling lamp. When coming home I'm used to switch it on by the wall powers witch. I found it annoying to have to pick up my phone, open the app and wait for the phone to connect.
Yes
Yes, very! And fun to show for a friend that was over one evening.
For the most part. But when attatched to the ceiling lamp, then you can't use your regular power switch on the wall. Takes time to get used to.
Yes

Did you feel that something was missing from the product?
Not really.
No
No but maybe add sensors in the future?
No
No

Figure 40: Questions thirteen and fourteen. Authors own graph.

Did you experience any kind of problem with the product?
Bugs in the application. Crashes when trying to open adapters tab without being connected for example.
No
Sometimes it felt like the application dident respond but after a new request it worked as expected
It crashed two times but I don't remember precisly when
Bugs, felt a bit slow sometimes

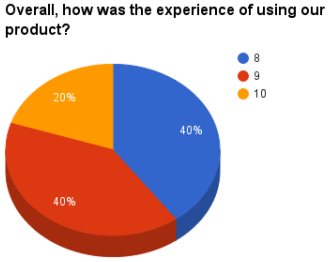
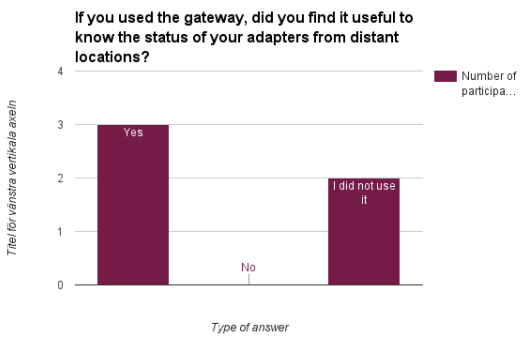


Figure 41: Questions fifteen and sixteen. Authors own graph.



Did you ever toggle the status of adapters from distant locations, if so how was your experienc
Once, to turn on the coffe machine
I did stand out on the street to test my lamp in the window and it work perfecly
Yes I did.
Did not use it
Did not use it

Figure 42: Questions seventeen and eighteen. Authors own graph.

Appendix G - ER-Diagram

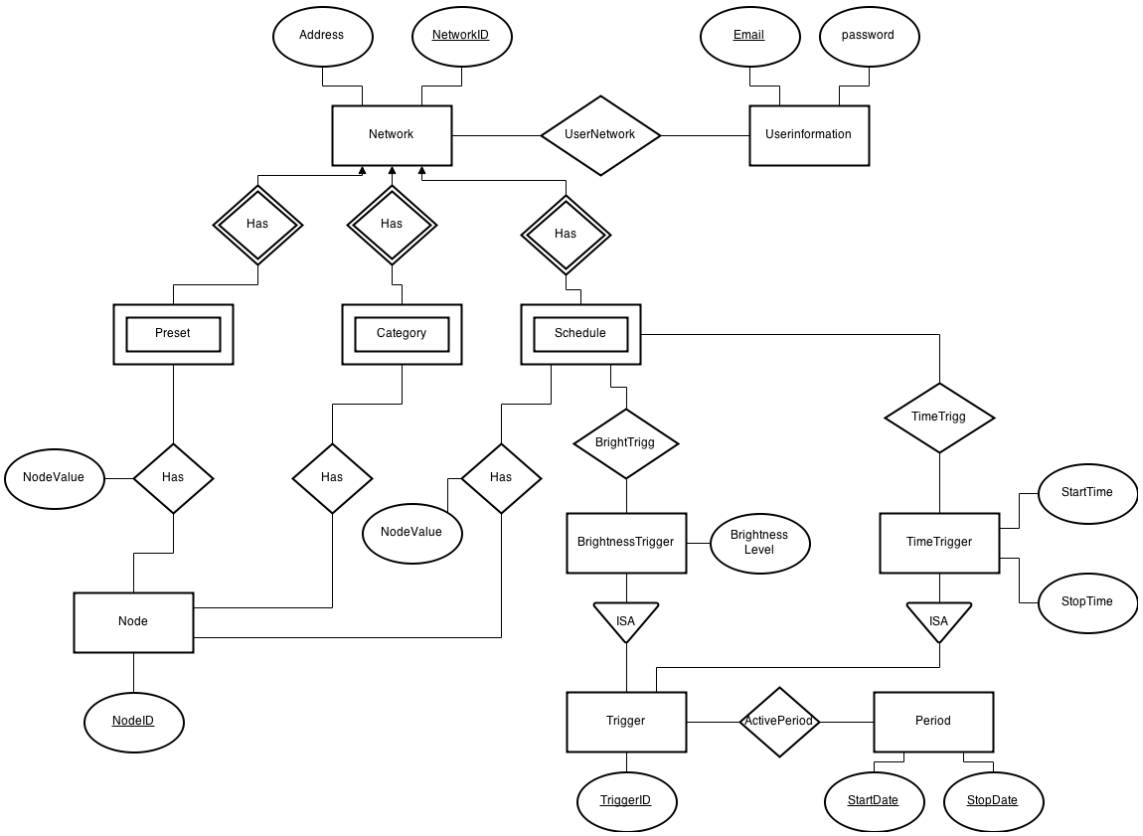


Figure 43: Entity-relationship diagram describing the database domain.

Appendix H - Dimmer Adapter Schematic

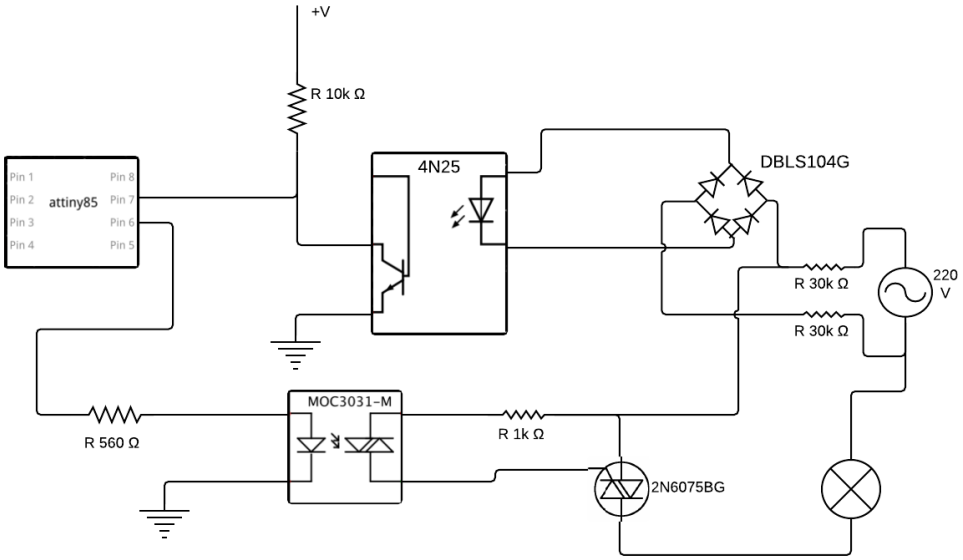


Figure 44: The schematic of the dimmer adapter.