Master's thesis

Central Home Hub

Bc. Martin Pilík

Faculty of Electrical Engineering
Department of Radioelectronics
Supervisor: Ing. Vladimír Janíček, Ph.D.

August 14, 2020

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Pilík Martin**	Personal ID number:	**434898**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Radioelectronics**		
Study program:	**Electronics and Communications**		
Specialisation:	**Technology of the Internet of Things**		

II. Master’s thesis details

Master’s thesis title in English:

Central Home Hub

Master’s thesis title in Czech:

Centrální domácí HUB

Guidelines:

1) Study the problems of process control in the house, detection of operational variables (air quality etc.).
2) Design a home hub that will act as a central hub with security, regulatory, control and multimedia functions. Use Home Assistant or Openhab as the core system.
3) The device will consist of a central unit and subordinate wireless units of the IoT type. Internet access will be used to share measured values and get weather forecasts.
4) User data will be displayed on the central unit and on the webportal.
5) Design a schematic diagram of the central and remote unit and implement the equipment in your prototype.
6) Evaluate functionality and compare reached parameters with commercial products.

Bibliography / sources:

[1] G. O'Driscoll, Smart Home Automation Essential Guides Box Set: The box set includes the first seven books of the Smart Home Automation Essential Guides series, plus a bonus book about Wearable devices, ASIN: B00YIUMVXC
[2] P Seneviratne, Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks, ISBN-13: 978-1484243565.

Name and workplace of master’s thesis supervisor:

Ing. Vladimír Janíček, Ph.D., Department of Microelectronics, FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **12.02.2020** Deadline for master's thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

Ing. Vladimír Janíček, Ph.D. Supervisor’s signature	doc. Ing. Josef Dobeš, CSc. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt — Student’s signature

Acknowledgements

I would like to express my deepest appreciation to my thesis supervisor Mr. Ing. Vladimír Janíček, Ph.D. for his valuable advice and helpful contributions. I would also like to extend my deepest gratitude to my family for their relentless support.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as school work under the provisions of Article 60(1) of the Act.

In Prague on August 14, 2020 . . . .

Citation of this thesis

Pilík, Martin. *Central Home Hub.* Master's thesis. Czech Technical University in Prague, Faculty of Electrical Engineering, 2020.

Abstract

The thesis contains a detailed description of the design and implementation of a device called Central Home Hub. The device is capable of running a home automation system, a media server and a home security system. Custom hardware design, built with the help of Raspberry Pi, grants users a vast range of possibilities for home automation.

The aim of the thesis is to create and test a universal platform that users can expand according to their needs. To achieve this, Central Home Hub is equipped with Home Assistant software and Zigbee radio unit that allows the user to connect an extensive supply of smart devices.

Keywords Internet of Things, smart technology, smart home, Zigbee, MQTT, zigbee2mqtt, Z-Stack, Home Assistant, security system, Raspberry Pi

Abstrakt

Práce obsahuje detailní popis návrhu a realizace zařízení pojmenované Central Home Hub. Toto zařízení umožňuje implementaci systému domácí automatizace, multimediálního serveru a domácího zabezpečovacího systému. Vlastní návrh hardwaru, postaveného na Raspberry Pi, přináší uživatelům široké možnosti domácí automatizace.

Cílem práce je vytvořit univerzální platformu, kterou si uživatelé mohou rozšiřovat a obohacovat podle svým potřeb, a ověřit její funkčnost. K dosažení tohoto cíle je v Central Home Hub nainstalován software Home Assistant a je k němu připojen radiový modul Zigbee. To uživatelům umožňuje připojení velkého množství chytrých zařízení.

Klíčová slova Internet věcí, chytré technologie, chytrá domácnost, Zigbee, MQTT, zigbee2mqtt, Z-Stack, Home Assistant, zabezpečovací systém, Raspberrz Pi

Contents

Introduction **1**

1 Protocols and technologies **3**
1.1 Internet of Things 3
1.2 Zigbee 6
1.3 MQTT 9
1.4 Zigbee2MQTT 12

2 Home Automation **15**
2.1 Domoticz 16
2.2 openHAB 17
2.3 Home Assistant 18
2.4 Automation 21
2.5 Home automation platforms comparison 25

3 Hardware solution **27**
3.1 Raspberry Pi 4 Model B 27
3.2 Zigbee radio design 29
3.3 Other peripheries 32
3.4 Connected sensors and devices 33

4 Central Home Hub **35**
4.1 Set up 36
4.2 Central Home Hub overview 39
4.3 Zigbee network 42
4.4 Comparison with other commercial products 43

Conclusion **45**

Bibliography **47**

A List of acronyms **51**

B Custom radio unit board schematic **53**

C Custom radio unit board PCB layout **55**

List of Figures

1.1 Growth of IoT [1] 4
1.2 HAN medium classification diagram [2] 4
1.3 Zigbee Protocol Layers [3] 7
1.4 Types of Zigbee nodes in a Zigbee network [4] 8
1.5 Zigbee topology [4] 9
1.6 MQTT architecture diagram [5] 10
1.7 MQTT QoS diagram [6] 11
1.8 MQTT use cases [7] 11
1.9 Zigbee2MQTT architecture [8] 12

2.1 Home automation architecture overview [9] 16
2.2 Example of a Domoticz dashboard [10] 17
2.3 Example of an openHAB dashboard [11] 18
2.4 Types of Home Assistant installations [9] 19
2.5 Example of a configuration.yaml file [12] 20
2.6 Example of an automation.yaml file [12] 22
2.7 Home Assistant integration page [12] 23
2.8 Example of Home Assistant dashboard [12] 24

3.1 Raspberry Pi 4 Model B [13] 28
3.2 Comparison of Raspberry Pi 4 and Raspberry Pi 3 [14] 29
3.3 Schematic of a matching network 31
3.4 Designed PCB 32
3.5 Used sensors 34

4.1 Central Home Hub 35
4.2 Mosquitto configuration file 37
4.3 Zigbee2MQTT configuration file 38
4.4 PLEX multimedia server [15] 38
4.5 Weather forecast tab 40
4.6 Alarm control panel 41

4.7 Zigbee network 43

B.1 Custom radio unit board schematic 54

C.1 PCB layout top layer view 56
C.2 PCB layout bottom layer view 57

Introduction

Automation is a very progressive field found in both industrial use and every day life. Processes which were done manually are being replaced with automated solutions. These innovations save time, money and human resources. Nowadays, we are able to produce automation systems and devices that are cost effective and reliable. This is possible due to rapid development in such fields as communication, power management and energy saving, sensor technology, nanotechnology, data science and computation.[16]

Other important areas of technological development are miniaturization and integration. With devices becoming increasingly smaller while maintaining sufficient computation resources, we are able to integrate many functions into a single device without the need of having these functions distributed within our system. Single board computers, and Raspberry Pi in particular have become powerful enough to run various demanding processes at the same time, for example video transcoding, file sharing and many others. As a result, the process of developing a customized electronic design system has become more affordable and easier.

With the previously mentioned technologies advancing, automation has started to play an important role not only in industrial deployment but also in commercial areas. Home automation is an excellent example of the use of aforementioned technologies for commercial purposes. More and more consumers are keen to equip their residences with an automation system, making their everyday lives easier. Security systems were the most common component for a long period of time. Recently, more advanced home control systems have become increasingly popular, consequently directing the attention of many well - established consumer electronics companies towards providing the end customer with choices between many different platforms and a vast selection of smart devices for a reasonable price.

The objective of this master thesis is to design and manufacture a home hub - a device which acts as a central unit of a system with security, control and multimedia functions. The ambition is to develop a system that is able to integrate various different applications within a single unit.

The core of the system is a single board computer Raspberry Pi, which itself consists of many components that are essential for the system. To expand its usage, it is connected to a custom designed printed circuit board (PCB) and a display. The custom PCB is used in order to provide communication using a high-level IEEE 802.15.4-based Zigbee protocol. Zigbee creates a personal area network for low-powered devices to which subordinate wireless units are connected. The obtained data and other types of information, such as weather forecast, are displayed on a connected touch screen display.

As mentioned before, many companies provide proprietary home automation solutions. Unfortunately, these solutions are isolated in most cases, therefore, if one wants to integrate products from different vendors, several proprietary gateways are needed, which is not only an inconvenience but also a more expensive solution. The system developed in this thesis substitutes the need for several gateways and allows the user to integrate sensors and devices from different vendors using just one device which acts like a universal gateway. For further home control and automation, Home Assistant software platform is used.

CHAPTER 1

Protocols and technologies

In this chapter general overview of how to connect devices to a central node is described. Later on, protocols used in this thesis are listed and use of these protocols within the solution is explained.

Communication between devices is a major aspect of any home automation. We need to be able to receive data from distant nodes in the central node. Since wired communication brings a lot of inconveniences, wireless communication is a necessity in modern systems. The ever growing field of Internet of Things provides many protocols that make the node to node communication simple and reliable.

1.1 Internet of Things

The term *Internet of Things (IoT)* refers to a network of devices, sensors and everyday objects that have the capability of computation and network connectivity and normally are not considered as computers. These devices allow to generate, exchange and use data with minimal human interaction. This concept of monitoring and controlling devices within a network has existed for many years. However, the recent trends are making the Internet of Things more available, and thus more widespread.[17]

Different communication models are implemented in IoT to ensure the flexibility of the network. Some applications may require high-speed connection, others need very reliable and low latency connection. While the core concept is always similar, one can find many different definitions of what IoT is. IoT can refer to a broadband cellular network (e.g. 5G network) as well as to a narrowband sensor network running in unlicensed frequency bands. Therefore many communication protocols are considered as a part of IoT. In this thesis, only protocols that suit home automation and sensor networks are mentioned.[17]

Although most protocols and standards are wireless using radio frequency spectrum for the means of communication, there are also some that uses wires

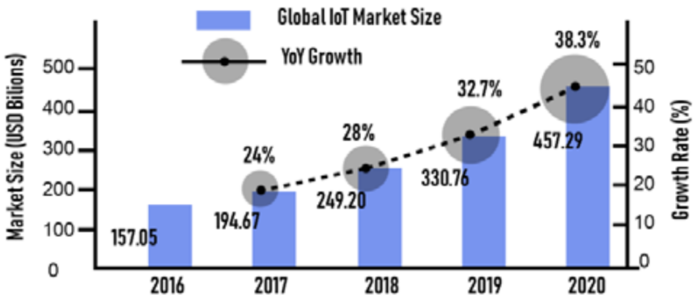

Figure 1.1: Growth of IoT [1]

and some that combine wired and wireless solutions. The aim is to enable *Machine to Machine (M2M)* network as an efficient means to provide automated communication among distributed devices and create a *Home Area Network (HAN)*. HAN can comprise several different networks which employ different technologies to work as a home automation platform. No standard or protocol has been adopted universally, on that account some of the most commonly used protocols are described in the following section. The focus is mostly on Zigbee specification as it is used in the solution of this thesis.[2]

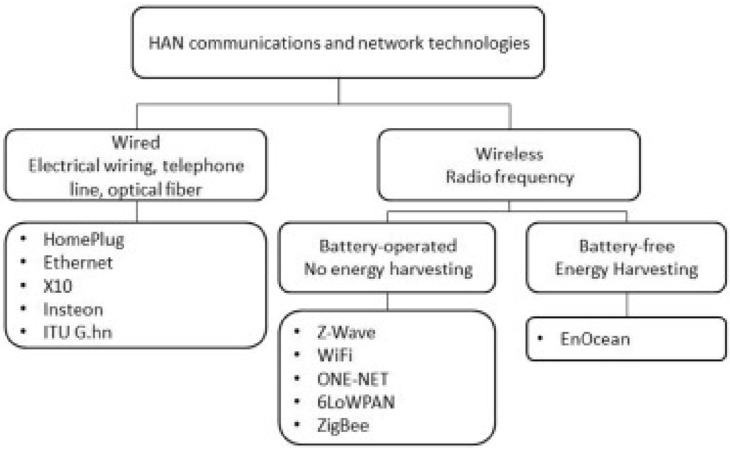

Figure 1.2: HAN medium classification diagram [2]

1.1.1 Internet protocol (IP)

The Internet protocol is the foundation on which the Internet is built. The infrastructure is already present in most homes as part of Wi-Fi or Ethernet network. The first problem of using the legacy infrastructure is that Ethernet sockets are not positioned where needed (or are not present at all) and that

Wi-Fi is energy inefficient - it does not allow the use of battery powered devices. The second problem is the integration of enormous number of IP-based devices with Non-IP based devices. The integration can be done with a multi-protocol gateway bridging. However, the bridge between different technologies is limited and the creation and maintenance of such bridge is very demanding. For these reasons, Internet protocol is not the best solution for home automation and Home Area Network. Nevertheless, Internet protocol finds its place in the Central Home Hub when it comes to data sharing and multimedia streaming.[2]

1.1.2 Z-Wave

Z-Wave is one of the most popular automation technologies. It was developed in 2001 by the Danish company Zensys specifically for the home automation market. Current standard is maintained by the Z-Wave Alliance. Z-Wave operates in the sub-1GHz frequency band therefore it does not interfere with other technologies in the 2.4 GHz band. Low powered radio frequency communication supports full mesh networks without the need for a coordinate node. The data transmission speed is relatively slow (between 40 and 100 Kbps), however, for home automation this is not a major disadvantage since Z-Wave is used to send commands instead of streaming large amounts of data. Mesh networking increases the network range as commands can hop from one device to another. Up to 232 devices can be connected to one network.[18][19]

All the previously mentioned features along with a high availability of supported devices (over 2400 interoperable products) make Z-Wave a good solution for home automation. This protocol is widely deployed in commercial activities, for instance, many hotels and cruise ships use Z-Wave for their automation.

1.1.3 Bluetooth

Bluetooth is a universally used, widespread technology. It is a wireless radio protocol using the 2.4 GHz band with a Master/Slave setup. Up to seven slave devices can be connected to one master. Bluetooth allows faster data transmission than Z-Wave or Zigbee (Bluetooth 3.0 up to 24 Mbps) and is a good choice for local audio streaming to headphones and speakers. However, its short range can be a limitation. Bluetooth is mainly popular due to the simple pairing method as it is easy to discover other devices and connect to them.[18][20]

Bluetooth Low Energy (BLE) became available with the release of Bluetooth 4.0 protocol. The key feature is the BLE's low power consumption which enables the connected devices to run on small battery devices for years. BLE devices remain in sleeping mode until connection is initiated. Connection times are only several ms long. Thus, BLE can be found in M2M and IoT applications. Even though the protocol is suitable for home automation, not many BLE devices are available, hence BLE has not been so widely used in home automation yet.[20]

1.1.4 Thread

Thread is a low-rate and low-power wireless personal area network standard. It connects devices into a mesh network which allows them to communicate with each other and with the cloud. Thread radio uses 2.4 GHz frequency band. Thread brings the internet to the Internet of Things as it is based on Internet Protocol version 6 (IPv6) with 6LoWPAN. This means that all devices run on an IP network and can be connected to the Internet and also to existing devices such as phones or tablets. There is a similarity with Wi-Fi, but Thread is designed with low power consumption which makes the standard suitable for home automation.[18][21]

The standard is managed by Thread Group. Many major electronics companies are a part of the group, including Google/Nest, Apple and Samsung. To implement and practise Thread technology, an annual membership fee must be paid. In order to accelerate the development of Thread supporting devices Google released an open source implementation of Thread called OpenThread. [18][21][22]

1.2 Zigbee

In this section, Zigbee is described in detail as it is the protocol that is used in the home automation solution designed in this thesis.

The Zigbee standard is a high level communication protocol based on the IEEE 802.15.4 standard. It was developed to suit low-power and low-speed short-distance communication. It runs in a 2.4 GHz frequency band with sixteen 2 MHz wide channels. Even though Zigbee can operate in the 915 MHz (USA) and the 868 MHz (Europe) frequency bands, most applications use the 2.4 GHz band only. The latest Zigbee release supports frequency hopping that allows Zigbee network to switch from one channel to another if overloading or high interference occurs. Data transmission offers speed up to 250 kbps which is enough for sending commands through the network. Zigbee uses Direct Sequence Spread Spectrum (DSSS) access mode and Offset - Quadrature Phase Shift Keying modulation (O-QPSK) scheme in the 2.4 GHz band.[23][2]

Zigbee consists of four layers (shown in figure 1.3):

1. Physical layer (PHY)
2. Medium Access Control Layer (MAC)
3. Network Layer (NWK)

4. Application layer (APL)

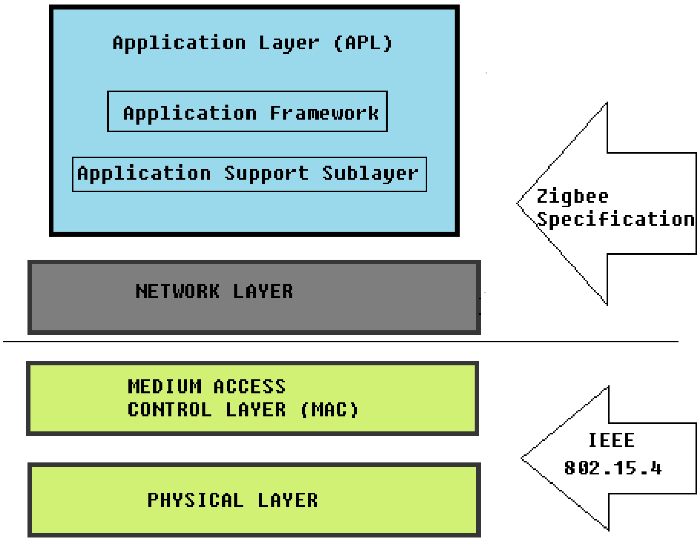

Figure 1.3: Zigbee Protocol Layers [3]

The PHY layer and MAC layer are defined by IEEE 802.15.4 standard. Zigbee protocol defines only the NWK and APL layers. In addition, the Application layer is divided into two sub layers. The top sub layer is designed for the public profile which allows vendors to create interoperable products like Home automation and Smart energy. The remaining sub layer is defined for the development of manufacturer's proprietary profiles.[23]

The IEEE 802.15.4 standard defines two types of devices. The first type is a *Full Functioning Device (FFD)* which can communicate with any other device in the network and can be a Personal Area Network Coordinator. The FFD can also act as an end node (sensor/actuator) or it can relay messages to other devices. The second type is a *Reduced Functioning Device (RFD)* which can be connected only to a FFD node. RFD devices are usually used as sensors or actuators and they have reduced communication and processing capabilities, consequently their production cost is lower.[24]

Zigbee protocol defines three types of devices[23][24]:

1. **Zigbee coordinator:** ZC initializes, controls and maintains the Zigbee network. ZC also acts as MAC coordinator. Additionally, it manages the network security and stores information about the network. The device has to be a FFD and it cannot be set into sleep mode, hence, it is usually not battery powered.

2. **Zigbee router:** the function of ZR is to relay messages from other nodes. It has to be connected to a coordinator or to another router. ZR is a FFD and also cannot be in a sleep mode. However, ZR can have a sensor/actuator.

3. **Zigbee end device:** ZED is a RFD and is capable of sensing or actuating. However, ZED does not take part in network routing and thus can be in sleep mode most of the time and use battery power. ZED is not connected to any other ZED.

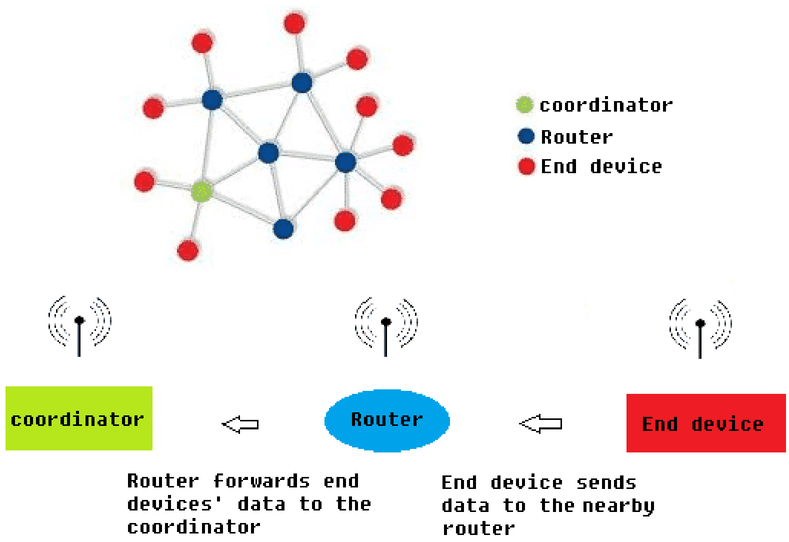

Figure 1.4: Types of Zigbee nodes in a Zigbee network [4]

Zigbee supports three topologies: star topology, mesh topology and cluster tree topology. In star topology, each node is connected directly to the coordinator. Mesh topology is either peer to peer or a multihop network. Data passes through other nodes within their radio range until they reach their destination. Multiple different routing paths can be established between the original and destination nodes. In cluster tree topology, routers expand the network. End devices are connected either to the coordinator or the routers. All three topologies are shown in the figure 1.5.[23]

The Zigbee Alliance is responsible for development, certification and promotion of the protocol. The Alliance is composed of many major electronics companies such as Honeywell, Motorola and Philips (Signify). Zigbee applications include smart lighting, home automation, security systems or general wireless sensor networks.

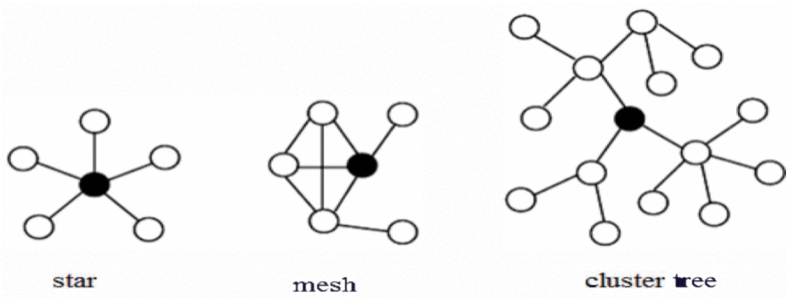

Figure 1.5: Zigbee topology [4]

The choice of Zigbee as a communication protocol in this thesis was made based on the availability and affordability of Zigbee devices, simple integration of the protocol into home automation systems and its flexibility.

1.3 MQTT

MQTT is an abbreviation for Message Queuing Telemetry Transport. It is a lightweight publish/subscribe messaging protocol which provides a simple way of transporting data between devices. The protocol is used for M2M communication and usually runs over TCP/IP or other network protocol that supports ordered, bi-directional connection. MQTT is an important protocol used in the Internet of Things. It is a good choice for networks that experience bandwidth constrains and latency.[7]

MQTT was originally created in 1999 to remotely communicate between a server and distant monitoring devices in the oil and gas industry. At that time, the only option for communicating with remote locations was satellite communication. Satellite communication is very expensive and billed for the amount of transferred data, therefore the industry needed a solution that provides a simple-to-use solution with minimal data rates. The organization for the Advancement of Structured Information Standards (OASIS) standardised MQTT in 2013 as an open source protocol.[5]

As mentioned before, MQTT uses publish/subscribe topology with two subjects: client and broker. MQTT broker works in a server to which devices (clients) are connected. The broker receives data from clients and distributes them further, hence clients do not communicate with each other directly. The operation when a client (device) wants to send data to broker (server) is called *publish*. The reverse operation when a client wants to obtain data from a server is called *subscribe*. Each client can be either a subscriber, a publisher or both. A client publishes only when there is data to be sent and a broker sends data to subscribers only after it receives the data. This reduces data transmission to minimum. Protocol architecture diagram is shown in the figure 1.6.[5]

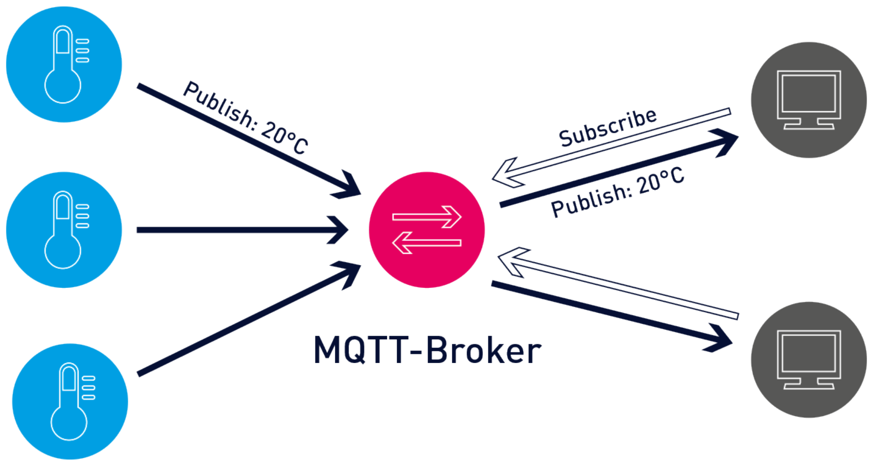

Figure 1.6: MQTT architecture diagram [5]

Messages are published as topics that are structured hierarchically with a similar structure to the directory tree of a computer file system. This allows subscriber to specify what topics it wants to receive. Topics are created by publishers. Each time broker receives a new topic, broker establishes the topic within the network and subscribers can connect to it. If the connection between subscribing client and broker is disconnected, broker buffers messages and sends them once the subscriber is back online. If a publisher is disconnected improperly, broker sends cached last will message, that was previously sent by publisher. This special message contains appropriate sequence of actions that should be done by all subscribers.

To minimize transmission, MQTT uses small message construction. Each message contains fixed header of 2 bytes and the payload is limited to 256 MB. Additional header can be used to increase the size of the message. MQTT also defines Quality of Service that can either minimize the data transmission or maximize the network reliability. The principle of QoS in MQTT is explained in this figure 1.7.

Each MQTT session can be divided into four procedures: connection, authentication, communication and termination. Clients start the session by creating connection to the broker using the broker's port. There are two standard ports: 1883 for non-encrypted connections and 8883 for encrypted connections using SSL/TLS. However, secure connection using SSL/TLS might not always be preferable or possible, especially while working with IoT devices. For that reason, another way of authentication is enabled. Clients may send cleartext username and password to the broker. For anonymous connection username and password are left blank. If the network in which MQTT works is secured, then the security of MQTT data transmission is irrelevant. In this case, MQTT communication is as safe as the network in which MQTT

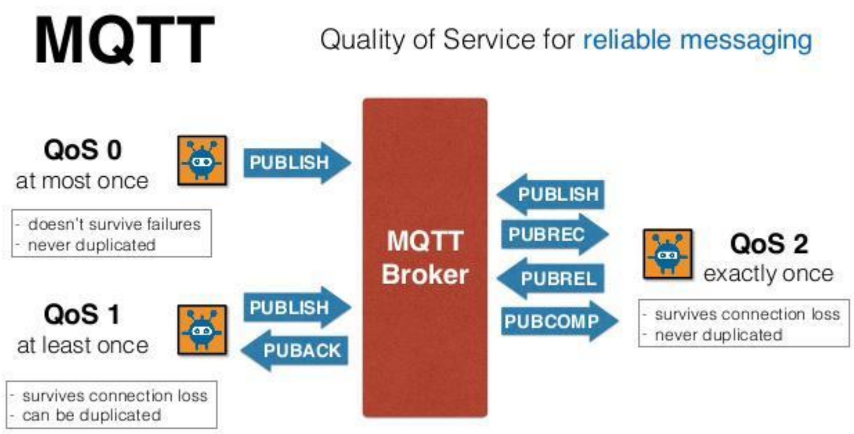

Figure 1.7: MQTT QoS diagram [6]

works.[5][7]

MQTT finds its applications in various fields, both industrial and commercial. It has a big impact on transportation and logistics, energy industries or healthcare. It is also a great solution for monitoring in environments, including home automation. MQTT is also supported by major cloud services providers including Google Cloud or Microsoft Azure. In addition, Facebook currently uses MQTT for its Messenger app.[7]

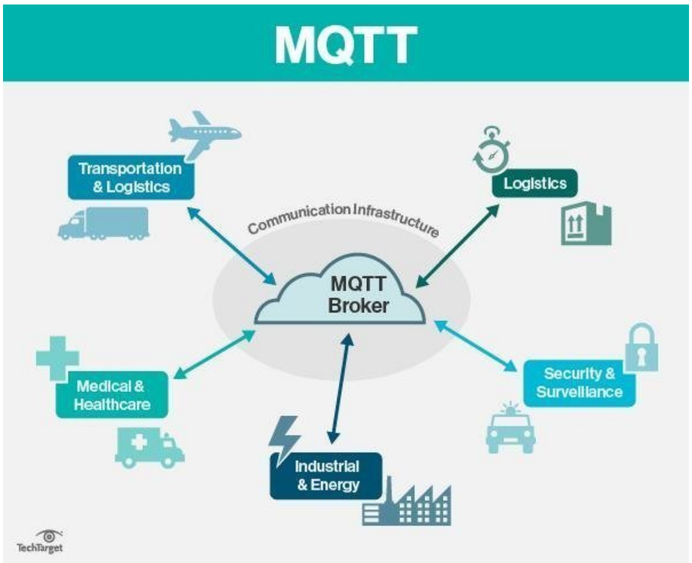

Figure 1.8: MQTT use cases [7]

1.4 Zigbee2MQTT

Zigbee2MQTT is not a standardised protocol that is maintained by acknowledged organisation, it is an open source project developed by Koen Kanters and is publicly available on GitHub. This project connects the two previously described protocols to work together. It bridges events and allows to control Zigbee devices via MQTT. This way it allows to integrate Zigbee devices with any smart home infrastructure platform. By using Zigbee2MQTT we are able to easily integrate Zigbee devices from many different vendors into home automation systems like Home Assistant, Domoticz or openHAB. This is all possible by using one universal Zigbee2MQTT gateway instead of using several proprietary bridges or gateways.

Being an open source project which is not supported by a technical organization, not much information can be found about Zigbee2MQTT except for well maintained documentation can be found on the official project website[25] and Koen Kanter's GitHub repository.[8] These two sources contain information about how to flash supported hardware with appropriate firmware, how to set up a Zigbee2MQTT gateway running on different platforms, how to pair devices to the created network or how to integrate Zigbee2MQTT into different home automation platforms. In the project website[25], a list of supported devices can be found. The list currently contains 919 devices from 155 different vendors. Even Zigbee devices that are not listed can be added to the network following *How to support new devices* the instructions that can be found on the same website.

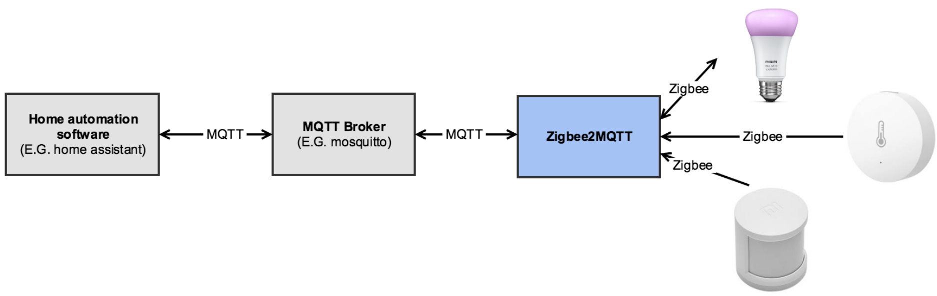

Figure 1.9: Zigbee2MQTT architecture [8]

Zigbee2MQTT is made up of three modules. Each of them is developed in its own GitHub project. First module, zigbee-herdsman, is an open source Zigbee gateway solution. It connects a Zigbee hardware adapter with higher levels of the protocol stack through an API that is enabled. Thus, zigbee-herdsman handles the core Zigbee communication. For instance, for Texas Instruments hardware (which is used in this thesis solution), the module uses the *TI zStack monitoring and test API*[26] to communicate with the adapter. The second module, zigbee-herdsman-converters, is responsible for mapping particular Zigbee devices into appropriate Zigbee clusters. Zigbee clusters are the top layers of Zigbee protocol which define how the different sets of devices, for example sensors and lights, communicate with each other over the network. The last module, Zigbee2MQTT, uses the previous two modules and maps the Zigbee messages to MQTT messages. It also maintains the whole system and keeps track of its state in *database.db* file. This text file is a JSON database and stores data about connected devices and their capabilities. The Zigbee2MQTT architecture is shown as a diagram in figure 1.9.

Chapter 2

Home Automation

Basic overview and description of widely used home automation platforms is provided in this chapter. Their advantages and disadvantages are discussed as well as supported protocols and integration possibilities.

The home automation system integrates many home domains such as lighting, entertainment systems and home security into one platform. The system usually connects distributed devices into one hub. One of its functions is to receive data from sensors monitoring the environment and automate actions based on this data. The other function is to display the received data and allow people to control the whole system with only a few clicks on their smartphones, PCs, laptops or wall-mounted terminals. An advanced system should also provide the function of remote access to allow people to control their households wherever they are. Nowadays, many platforms provide some or all of these functions. The three biggest open source home automation platforms are described in this chapter:

- Domoticz,
- openHAB,
- Home Assistant.

To get a better idea of how home automation works, an example of the architecture is shown in figure 2.1. The figure shows different components of the system in appropriate places and communication streams. The home control component is responsible for collecting data and controlling devices. Devices can be controlled directly from user input, as a home automation or smart home commands. While smart home component triggers commands based on previous behaviour, the home automation component triggers commands based on user configuration.

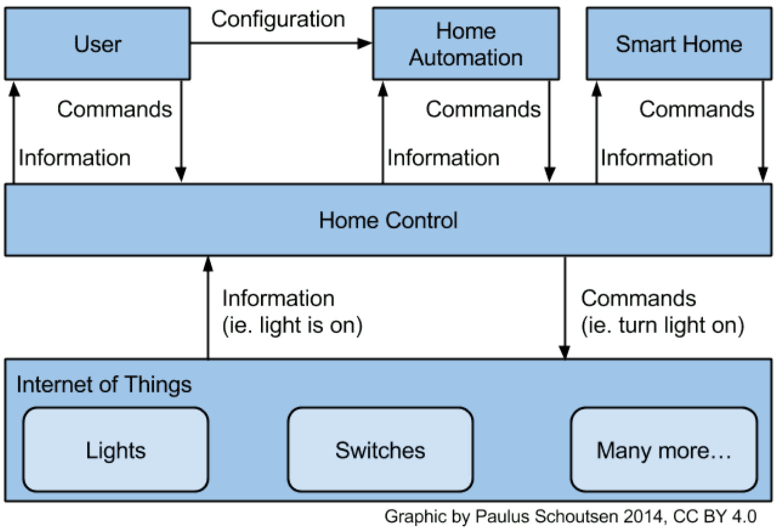

Figure 2.1: Home automation architecture overview [9]

2.1 Domoticz

Domoticz is a light weight home automation system which enables monitoring and configuration of many different kinds of devices. Connection of lights, sensors, switches, energy consumption monitoring systems and many more. The platform is an open source project and is accessible for free. The only investment needed to be made is the purchase of hardware to run Domoticz on along with the devices that are connected to the system.[27]

The platform can be installed on several systems. The most frequently used ones are Linux distributions (including Raspberry Pi installation), Windows, macOS or Synology NAS. Domoticz is highly customizable and allows to write custom scripts and configurations. It also supports integration of a large number of devices including Z-Wave, Zigbee and Bluetooth products. A list of supported devices can be found on the official Domoticz Wiki website [27]. Domoticz also supports the integration of custom Python plugins which are shared within the community of users. One of the plugins provides the integration of Zigbee2MQTT project. Users can also create their own dashboards and customize user interface of the home automation system. An example of Domoticz dashboard is shown in the figure 2.2. The platform can be controlled from web interface or mobile apps for iOS and Android. An application for Tizen (smart watch) is also available.

Figure 2.2: Example of a Domoticz dashboard [10]

2.2 openHAB

The openHAB platform is another powerful home automation solution that allows users to integrate almost any device or system from different vendors into a single solution. The name is an abbreviation for open Home Automation Bus. It is an open source project running on a local device and acting like a central node of the network of connected lights, sensors, cameras or actuators. openHAB is more complex than Domoticz and offers advanced possibilities of automation and customization. Nevertheless, if a user wants to create a more complex home automation system, some programming skills and deeper IT knowledge is required as advanced configuration is done through text files and scripts. Therefore, as stated on the openHAB official website [28], openHAB is not a commercial off-the-shelf product, but rather a platform for *tech-savvy people*.[11]

openHAB can be executed on many platforms such as macOS, different Linux distributions and Windows. Supported Linux distributions can be hosted on servers, computers like Raspberry Pi or NAS stations, for example, Sysnology DiskStation.A popular solution is to run openHAB on Raspberry Pi, for which pre-configured Linux operation system openHABian image has been released. openHABian provides a self-configuring Linux system setup with all the necessities that openHAB requires.[29]

Users of openHAB can choose from numerous Add-ons that are easily integrated to communicate with their devices. The list of supported protocols

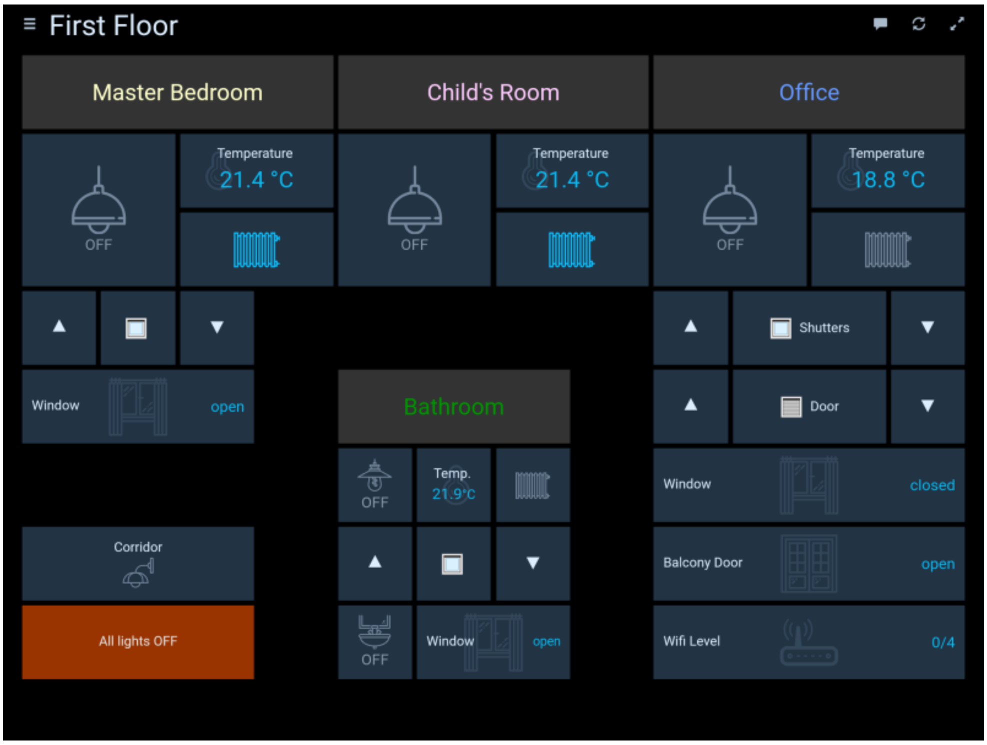

Figure 2.3: Example of an openHAB dashboard [11]

and technologies is extensive and contains all the widely used ones - Bluetooth, Z-Wave, Zigbee, MQTT, Philips Hue devices and Ikea TRÅDFRI devices. On top of that, such multimedia services as Sonos, Kodi and Plex, can be part of the system too. openHAB does not require any cloud service to work and runs on a local device, however, it is cloud-friendly, and integration with popular cloud-based smart home platforms like Amazon Alexa, Google Asssistant, IFTTT is available. Remote access is available too, as well as mobile apps for Andriod and iOS. Well maintained documentation is kept on openHAB official website. In order to see more detailed information and guides or to find help when struggling with a problem, one can find many answers in the openHAB community section on the same website.[11]

2.3 Home Assistant

Home Assistant is another robust platform for home automation that prioritizes local control and privacy. In many aspects, Home Assistant is similar to openHAB. The platform also supports the integration of all of the major Internet of Things ecosystems and offers advanced customization and automation. Because the designed *Central Home Hub* uses Home Assistant for home automation, the description is more detailed. All of the information that can be found in the following sections is taken from Home Assistant documentation [12]. The documentation provides extensive information and user guides in order to help users with installation, configuration, automation and with setting up more advanced functions of Home Assistant. Even more information with examples can be found within the Home Assistant community [30]. The community and the number of people that contribute to this open source project is one of the reasons why Home Assistant is so popular. In 2019 *State of the Octoverse* Home Assistant was listed on GitHub as the tenth biggest open source project based on the number of active contributors [31].

2.3.1 Installation

The installation of Home Assistant can be quite unclear, since multiple ways of installation are supported. The system can be installed in four different ways. To understand them better, components of Home Assistant are explained first.

The full installation consists of the Home Assistant Core, Supervisor and Home Assistant OS. The Home Assistant Core is a Python program that provides the ability to track, control and automate connected devices. Additional functions and improved user experience are provided by the Supervisor. The Supervisor runs the Home Assistant core, manages all parts of the system and keeps them up to date. It also allows users to use Add-ons to integrate devices easily and to configure them directly from user interface. The last part of the installation is Home Assistant OS which is an operating system optimized for hosting Home Assistant and its Add-ons. Home Assistant OS is not based on regular Linux distribution and is designed to be run specifically on single board computers such as Raspberry Pi, ODROID, NUC and Thinker Board.

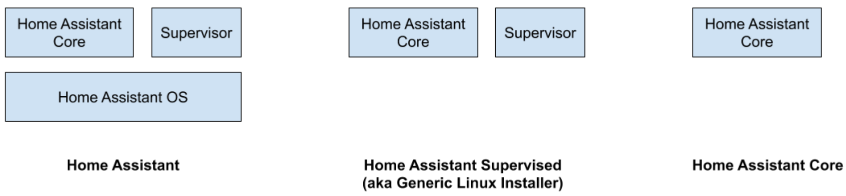

Figure 2.4: Types of Home Assistant installations [9]

The first two installation options run just the Home Assistant Core. The difference between them is that one runs the core directly in a Python virtual environment while the other runs the core in a Docker container. The third is called Home Assistant Supervised and provides the full Home Assistant experience on a regular Linux operating system, ubuntu or raspbian. This option contains the Home Assistant Core and the Supervisor. The last option is called simply Home Assistant and contains all three previously described components including the operating system which is installed on a single board computer or in a virtual machine. This option is highly recommended, however, the Home Assistant Supervised installation is perhaps the most common one.

2.3.2 Configuration

When Home Assistant is launched for the first time, it creates a configuration folder in which a configuration file can be found. The file enables web interface and device discovery. Users can configure the system either directly from the web interface or by editing the configuration file. Even if a user configure Home Assistant from the web interface, all the changes are automatically reflected in the configuration file. To easily access and edit the configuration file, an *File Editor* add-on can be installed which enables to edit files in the configuration folder directly from the web interface.

Configuration file is written in YAML language. YAML is a human friendly data serialization standard. It is commonly used for configuration files and is easily readable due to its minimal syntax. To indicate nesting, YAML uses Python-style indentation.

```yaml
homeassistant:
  name: Home
  latitude: 32.87336
  longitude: 117.22743
  elevation: 430
  unit_system: metric
  time_zone: America/Los_Angeles
  external_url: "https://www.example.com"
  internal_url: "http://homeassistant.local:8123"
  allowlist_external_dirs:
    - /usr/var/dumping-ground
    - /tmp
  allowlist_external_urls:
    - "http://images.com/image1.png"
```

Figure 2.5: Example of a configuration.yaml file [12]

In addition to the basic setting, integrations that user may wish to add to Home Assistant are listed in the configuration file. Even though many components are already supported for direct integration from the web interface, there are still components that have to be added manually into the configuration file. If the ambition is to create a complex home automation system using Home Assistant, the knowledge of managing the configuration file is necessary.

Aside from adding integrations into the system, configuration file is also responsible for most of the customization of Home Assistant. New devices are set up in the file. They can be further customized and arranged into different groups. The configuration file is the core of the whole Home Assistant. All functions of the system can be used only if needed components are correctly arranged first. Other functionalities such as remote control or cloud service integration is also configured in the file. As the system grows and becomes more and more complex, the configuration file expands too. When the file becomes too long, it can be useful to split it into more manageable pieces. Separated files then have to be included in the main configuration file using the *!include filename.yaml* command.

2.4 Automation

Automations in Home Assistant are managed in a similar manner as configuration. They can be created and controlled directly using the web interface or can be edited manually in another YAML file (automation.yaml by default). Home Assistant is capable of a wide range of custom automations.

The basic structure of each automation is simple and consists of three different parts:

- Trigger
- Condition (optional)
- Action

Triggers describe events that trigger the automation rule. For example, the trigger can be an event, certain entity state at a time, a specific MQTT received message or a numeric state, for instance when temperature raises above a certain value. The list of supported triggers is extensive, and users can benefit from using multiple triggers.

Conditions are an optional part of automation. Conditions can prevent an action from happening when the automation is triggered and the set condition does not pass. Conditions do not react to any change of state, they only observe the state of any system component and return a true or false value.

Actions interact with actuators and other devices that are capable of changing their behaviour. Only when a rule is triggered and all conditions pass, an action is executed. An action can, for example, turn the light on/off, change its brightness, start air conditioning or activate a scene. Multiple actions can be executed within one automation. Additional conditions can be a part of an action as well in order to determine the appropriate operation which should be executed.

Creating automations from the web interface is an excellent feature that makes the whole process short and uncomplicated. Simpler automations can be easily done this way, however, for longer and more complex automations

```yaml
automation:
  trigger:
    platform: sun
    event: sunset
    offset: "-00:45:00"
  condition:
    condition: state
    entity_id: all
    state: home
  action:
    service: light.turn_on
    entity_id: group.living_room_lights
```

Figure 2.6: Example of an automation.yaml file [12]

editing the *automation.yaml* might be a preferable option. Example of a simple text automation is shown in the figure 2.6.

For even more complex home automation, Home Assistant allows users to write their own scripts. Scripts are either accessible as an entity to the system or integrated into automations. Their function is to execute sequence of actions defined by user. To provide users more scripting options, routines, known from commonly used programming languages, such as loops or cycles, are available. Since scripts are written in YAML too, they may seem to be chaotic and confusing. Possibility of scripting in one of the many available programming languages would be a valuable feature that would make scripting clearer and more user friendly.

Home Assistant allows users to write their own scenes. The user selects all of the devices that are desired to be incorporated in the particular scene and sets their states in order to create an environment appropriate for a certain moment or activity. For instance, a scene *movie-watching* would dim all the lights in the living room, turn on the TV and the audio system and set the temperature one degree higher. In addition, scenes can be used as scripts and automations. Automations, scripts and scenes provide a powerful tool that enables users to create almost any kind of automation that they can think of.

2.4.1 Integrations

Home Assistant supports numerous different platforms that can be integrated into the system. Therefore, wide range of devices can be connected to create an extensive home automation environment. Currently,a total number of 1632 platforms and vendor's product lines can be incorporated into Home Assistant. Apart from typical platforms such as Zigbee, Z-Wave, Philips Hue or MQTT, users can integrate platforms like BMW Connected Drive, iRobot Roomba and Apple TV. The possibilities of integration are almost limitless and even more integrations are available using community supported (not yet officially supported) platforms. To mention a few more, cameras and security devices, doorbells, calendars, mailboxes, media players, routes and network components, weather forecast platforms - all these and many more can be integrated. Home Assistant also supports integration with cloud services, for instance Google Assistant or Amazon Alexa, which enable voice control functions.

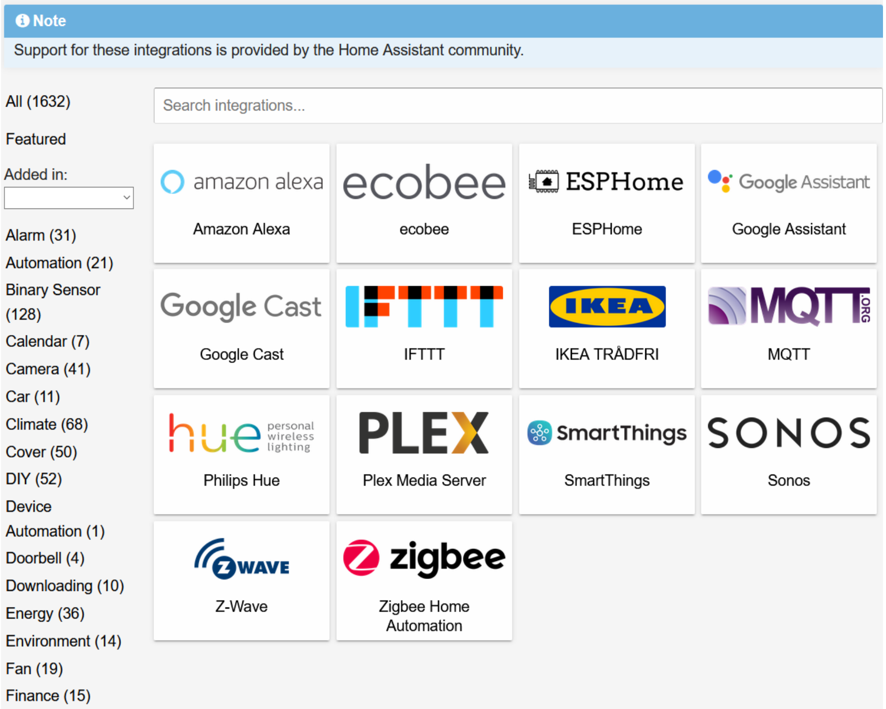

Figure 2.7: Home Assistant integration page [12]

2.4.2 Lovelace UI

An important part of home automation systems is the way they display data and enable users to control their homes. For this reasons, Home Assistant provides highly customizable user interface called Lovelace. Through this user interface, users can manage their homes from their smart phones, PCs or laptops.

Customized dashboards are the perfect way to control what entities should be displayed and in what position they should be displayed. Various different tabs can be added to serve different functions of home automations. Users can also create custom themes which define the colour set and background of the user interface. This allows to use brighter colours during the day and darker ones during the night.

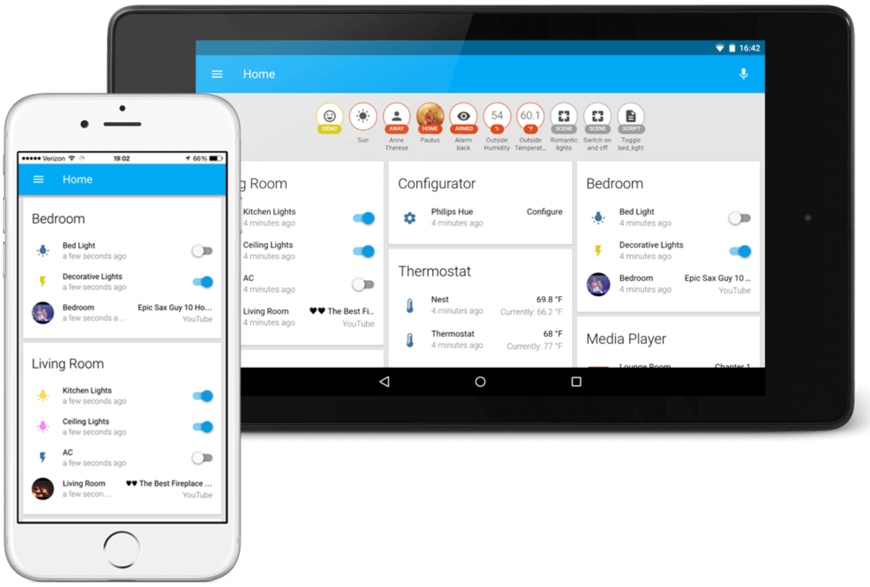

Figure 2.8: Example of Home Assistant dashboard [12]

Similarly, Lovelace can also be configured either directly from the user interface or by using YAML files. The dashboard is filled by adding cards with different entities. Home Assistant provides 26 cards by default and custom cards from the community are fully supported as well. Each of these cards have numerous options to configure how data is presented. For instance, one card can show the weather forecast, another - the room temperature with its graphic representation -, and the third one can be a live view from the garage, filmed with an installed camera.

2.4.3 Remote access

There are two ways to access Home Assistant remotely. The first is the way of using the official Home Assistant Cloud. Up to this point, the use of Home Assistant was free of charge. However, Home Assistant Cloud is a paid service which currently costs 5 USD per month (31 day free trial is available). In return, users get conveniently configurable, fully encrypted remote connection to their local home automation system. Another advantage is that using Home Assistant Cloud makes the integration of Google Assistant and Amazon Alexa very simple, thus voice control is easy to configure. Lastly, by paying the monthly fee users can support the founders of Home Assistant.

The second option is to make Home Assistant remotely accessible manually. This set up is more difficult, and users have to pay attention to security hazards as they have to secure the connection themselves with TLS/SSL.

Users also need a dedicated IP address. Problems can arise if user's Internet Services Provider offers dynamic IP addresses. Nonetheless, this can be solved by using a free Dynamic DNS service, which is available as one of the add-ons. The last thing that is necessary is the set up of port forwarding on a router to the port 8183 on the computer that hosts Home Assistant.

2.5 Home automation platforms comparison

All three previously mentioned open source platforms provide good solutions for home automation. Compared to the other options, Domowitz is lightweight and it uses less hardware resources. However, openHAB and Home Assistant offer wider selection of integrable platforms and devices as well as bigger and more active community of users. Home Assistant and openHAB are very similar in most aspects. Home Assistant is more user-friendly, yet, more experienced users might find limitations in scripting with a YAML syntax. openHAB offers more in this aspect, as it is possible to write scripts in JavaScript. Another difference is that Home Assistant issues new releases more often, hence new devices are supported sooner.[32]

Chapter 3

Hardware solution

In this chapter, hardware components of the Central home hub and the designed Home Area Network are described. The whole system consists of several components. The core of it is a single board computer Raspberry Pi 4 to which all other components are connected. The system consists of a central node (the Central home hub), wireless sensors, a light bulb and a camera. The wireless communication uses Zigbee and IP protocols.

To enable Zigbee communication, custom printed circuit board with a 2.4 GHz radio unit is designed and manufactured. In order to view the sensor obtained data and to control the whole system, a touch screen display is connected to the Raspberry Pi.

3.1 Raspberry Pi 4 Model B

The Raspberry Pi 4 Model B is the latest release of the Raspberry Pi device family. It is designed by Raspberry Pi Foundation, a nonprofit, education-focused organization. It is probably the best known single board computer that is broadly used for development of embedded devices, programming experiments and more and more as a budget desktop PC duet to its increase in computing abilities.[14]

The Raspberry Pi is a compact single board computer, which means that all of its components are placed on a single PCB. Only one major component is missing on the board - Raspberry Pi does not have any kind of a hard drive. Users have to install their own operating system onto a microSD card that is later inserted into the Raspberry Pi. Users can choose from various Linux distributions (e.g. the Raspbian is the most common one) or operating system images with single purpose, such as the Home Assistant OS. Some of the Linux distributions offer a lightweight version, which is known as the *headless* version. This version does not support graphical user interface and can only be controlled via terminal.[14]

Most applications running on Raspberry Pi make use of the 40-pin General Input/Output (GPIO) connector. Through this connector other devices can communicate with the board, using one of the many interfaces, e.g. UART, I2C, SPI, 1-wire or JTAG. This 40-pin GPIO header is fully backwards compatible with previous boards.

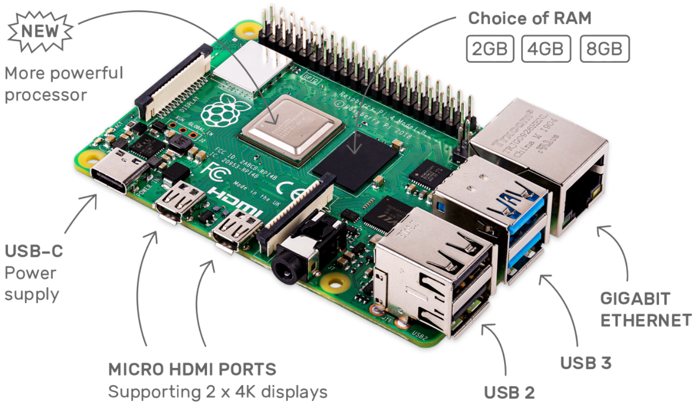

Figure 3.1: Raspberry Pi 4 Model B [13]

Computation capabilities of the latest released Raspberry Pi have been greatly improved compared to its predecessors. It benefits from a new Broadcom quad-core SoC processor unit, and for the first time, users can choose from different sizes of RAM (1 GB, 2 GB, 4 GB and even 8 GB versions are available). Raspberry Pi 4 also presents improved set of peripheries. Two USB 3.0 and two USB 2.0 Type A connectors, as well as a Gigabit Ethernet port are available. In addition, two 4K monitors can be connected through two micro HDMI connectors, and wireless communication is enabled by two radios for 802.11ac Wi-Fi and Bluetooth 5.0. Power supply is secured by 3.0 A USB-C connector. In general, the new Raspberry Pi is more powerful and allows users to set up more and faster connections with additional peripheries. In the figure 3.2, comparison of Raspberry Pi 4 and Raspberry Pi 3 can be viewed.[13]

The significantly better computing performance comes with a weakness - the board produces more heat compared to the previous models. Users need to be aware of this potential issue and, if possible, provide additional cooling. If the temperature reaches 80 degrees Celsius, a warning icon is displayed, furthermore, if the temperature reaches over 85 degrees Celsius, the CPU starts to throttle which can lead to a crash and shutdown.[14]

During the development of Central home hub device problems with Raspberry Pi overheating occurred, with temperature rising over 80 °C. It was caused by a simple mistake in Linux administration, when one service was initiated and killed in a short period of time over and over again. This used a major part of the computation resources. After this mistake was fixed, the

Raspberry Pi 4 vs.
Raspberry Pi 3 Model B+

	Raspberry Pi 4	Raspberry Pi 3 Model B+
Price	$55 (4GB model)	$35
Processor	1.5GHz Broadcom BCM2711 (quad-core Cortex-A72) SoC	1.4GHz Broadcom BCM2837B0 (quad-core Cortex-A53) SoC
Memory	4GB LPDDR4 (1GB and 2GB versions also available)	1GB LPDDR2
Graphics	ARM VideoCore VI (500MHz)	ARM VideoCore IV (400MHz)
Supported Codecs	OpenGL ES 3.0 graphics. H.265 (4kp60 decode); H.264 (1080p60 decode, 1080p30 encode)	OpenGL ES 2.0 graphics. H.264, MPEG-4 decode (1080p30); H.264 encode (1080p30)
Video Output(s)	Two micro HDMI	Full-size HDMI
General Purpose Input/Output (GPIO)	40-pin header	40-pin header
USB Ports	Two USB 3.0 ports; two USB 2.0 ports	Four USB 2.0 ports
Wired Networking	Gigabit Ethernet	Ethernet (maximum throughput 300Mbps)
Wireless Networking	802.11ac Wi-Fi; Bluetooth 5.0	802.11ac Wi-Fi; Bluetooth 4.2

Figure 3.2: Comparison of Raspberry Pi 4 and Raspberry Pi 3 [14]

system running Home Assistant remains at a temperature around 65 °C. When 4K video streaming is running the temperature can reach 71 °C, which is still a safe value.

In the solution provided in this thesis, Raspberry Pi runs the headless Raspbian operating system on which Home Assistant Supervised is installed, as well as a PLEX media server and Samba NAS. Raspberry also controls the attached display and manages communication with a Zigbee radio unit custom board.

3.2 Zigbee radio design

To create a home automation system, communication with wireless devices had to be established. For this purpose, Zigbee standard was chosen. To enable communication with Zigbee devices, custom board with a IEEE 802.15.4 radio unit was designed.

The custom board consists of several different parts. The first necessary part is a 40-pin connector, which enables connection to the Raspberry Pi. Through this connector, communication between the radio unit and Raspberry CPU is running. The connector also connects a 5V power supply line from the Raspberry Pi to the custom board. Additionally, several GPIO pins are connected to the radio unit chip to enable its programming. The second part is a voltage regulator that is connected to the Raspberry Pi 5V power supply line and outputs 3.3V for the radio unit. The most critical part of the board is the radio unit itself, which consists of a SoC Multi Control Unit CC2530, a 2.4 GHz IEEE 802.15.4 compliant RF transceiver and a 2.4 GHz range extender power amplifier CC2592. In the following sections the custom board design is described in detail.

3.2.1 Power supply

In order to understand the power supply solution better, general overview of how Raspberry Pi is powered is provided. Raspberry Pi 4 requires a 5V power supply with a 3A current. The power is drawn from a USB-C connector. From the 5V power supply Raspberry makes a 3.3V power supply line for its own operation and enables the supply further to the GPIO connector. However, the 3.3V power supply line has a current limit (approximately 800mA) and, if the drawn current exceeds this limit, Raspberry Pi's operation is endangered. On the other hand, 5V supply line is wired directly from the USB-C connector to the GPIO connector.

It is recommended to use a 5V power supply from the GPIO connector to power devices connected to Raspberry Pi. Even though the available current varies depending on the number of connected peripheries, approximately 1.5A should be available. It is a favourable approach to power custom devices from the 5V supply, especially when they draw more than 100mA.[33][34]

With taking the Raspberry Pi power limit into account, the custom board is powered from the 5V power supply. However, the RF chips needs to be powered with 3.3V supply. For this reason, a voltage regulator *TS1117BCW33* is used. The regulator converts the input 5V voltage to output 3.3V voltage with maximum output current of 1A. The regulator provides simple application with only two capacitors connected to the part.[35]

3.2.2 Radio for 2.4-GHz IEEE 802.15.4

For realisation of the radio hardware part, Texas Instruments products were chosen, specifically, Soc IEEE 802.15.4 compliant *CC2530* and its front end *CC2530* range extender. The choice of these products was made based on the provided documentation and reference design, as well as its compatibility with Texas Instrument's protocol stack - *Z-Stack*. In addition, the CC2530 product family and the Z-Stack protocol is fully supported by Zigbee2MQTT.

CC2530 can be used as a standalone solution, however, to increase the range of the coordinator, power amplifier chip CC2592 was added. Both circuits are powered by a 3.3V supply and they are linked with a bi-polar RF line. The CC2530 communicates with the Raspberry Pi via UART.

The two chips cannot operate by themselves and require several external components. External capacitors for power decoupling (power line filtering) must be used for optimum performance, as well as bias resistors to set an accurate bias current. More information about CC2530 and its implementation can be found in an official documentation on the product page [36], Similarly, additional information about CC2592 can be found on its product page [37]. Furthermore, Texas Instruments provide their customers with a reference design of this Zigbee network solution [38].

Use of these two components provides a complete radio solution with a great performance. The combination improves the receiver sensitivity by 2-3 dB compared to CC2530 standalone solution and increases a total link budget to 120 dB. Configurable output power can be set up to +22dBm. All these features increase the Zigbee network range significantly.

An important part of the high radio frequency design is input/output matching and filtering. The CC2592 includes an internal balun and a matching network which makes the interface for the CC2530 seamless. However, additional matching network needs to be implemented to match the CC2592 with the 50 Ω feeding line to the antenna. The RF output can be routed to either a PCB antenna or a SMA connector. Designers are advised to follow the reference design and PCB stack-up as closely as possible, because any change can have massive impact on the radio performance. The schematics of the whole custom board can be view in appendix B.1. The detailed schematics of the matching network is shown in the figure 3.3.

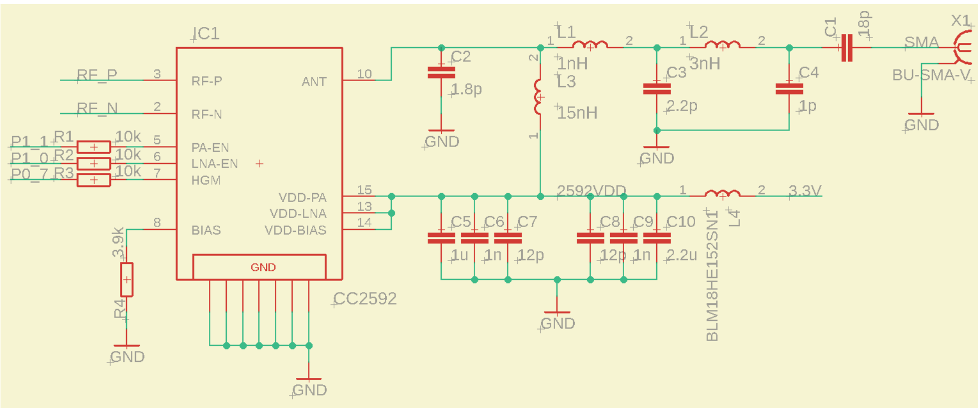

Figure 3.3: Schematic of a matching network

Even though the use of 4-layer PCB is recommended, in order to lower the cost of the device, 2-layer PCB was designed. That required additional changes to the reference design. To be able to achieve 50 Ω feeding line to the SMA connector, the thickness of the PCB was lowered to 0.8 mm and the width of the feeding line was increased to 1 mm. In general, radio frequency PCB design is a complex and challenging task, especially for an inexperienced hardware designer. Nonetheless, the resulting design performs very well. The range of the created Zigbee coordinator is good enough for most apartments and smaller houses without the need of using routers in the network. A photography of the designed custom board is shown in the figure 3.4

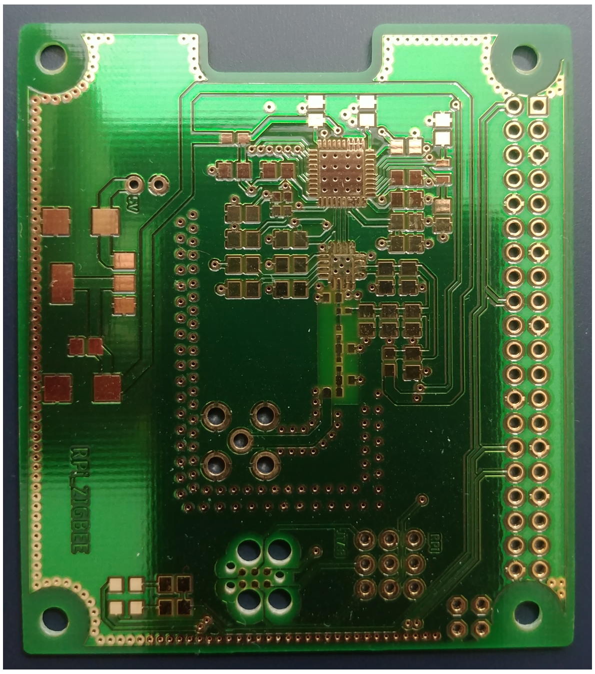

Figure 3.4: Designed PCB

3.3 Other peripheries

Other connected devices that are used as a part of the Central home hub and are essential for the home automation system are a touch screen display, a USB GSM modem, a back-up power supply and a USB flash disk.

The display enables users to control and view the environment. The official Raspberry Pi Touch Display was bought for this purpose. Even though the display works well and can be conveniently connected to the Raspberry Pi, a display with a higher resolution and a bigger screen than just 7 inches would have been a better choice. The resolution of 800 x 400 pixels limits the amount of information that can be viewed.

Since Central home hub also acts like a home security system, a back-up power supply and an option of communication even if disconnected from the Internet is necessary. USB GSM module is an easy and elegant solution which can notify the user even if he/she or the home automation system are disconnected from the Internet. To ensure that the Central Home Hub is powered even in case of a power outage, the device is powered through a power bank. The used power bank needs to be able to charge 15W (5V/3A).

The USB flash disk serves only as data storage for the multimedia server and a network file sharing system.

3.4 Connected sensors and devices

For home automation purposes several sensors, a light bulb and a camera were purchased:

- Philips Hue light bulb E27
- MiJia Honeywell smoke detector
- MiJia human body movement sensor
- Aqara human body movement and illuminance sensor
- Aqara door & window contact sensor
- Aqara water leak sensor
- Aqara temperature, humidity and pressure sensor
- TP-LINK Tapo C200 IP camera

The list of devices provides the basic set of equipment for the needs of home automation. With only a few devices a security system can be set up and automation is possible. Users can add many more devices that would customize their automation to their needs and desires, however, as an example of the capability of the system, this set is sufficient.

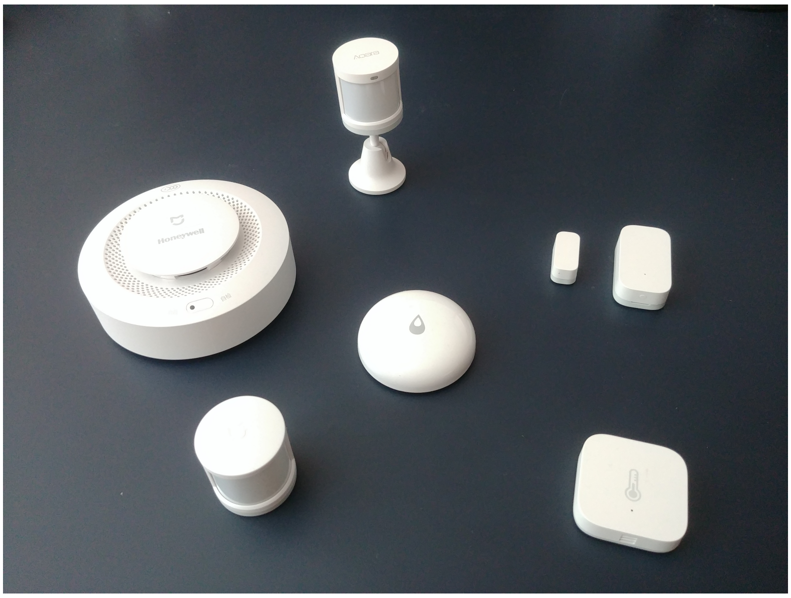

Figure 3.5: Used sensors

CHAPTER 4

Central Home Hub

Having the knowledge of IoT protocols, home automation platforms and available hardware components, it is time to present what can be achieved by mutually integrating and combining them. It is time to present Central Home Hub and its capabilities and features.

In this chapter product overview is presented, actions taken to complete the goal of the thesis are outlined, achieved parameters and functions of the final product are discussed. Additionally, information about the created Zigbee network is given and Central Home Hub is compared with commercial products.

Figure 4.1: Central Home Hub

4.1 Set up

In order to make Central Home Hub operate, several steps had to be taken. From the installation of the operating system on Raspberry Pi, through flashing the radio unit with firmware, to adjusting Home Assistant to suit our purposes. A summary of these steps is shortly presented in the following subsections. To follow these procedures, basic Linux administration skills are required.

4.1.1 Installation of Raspberry Pi

Numerous operating systems can be installed on Raspberry Pi. Headless Raspbian operating system was chosen for this application. Many tutorials of the installation process can be found on the Internet. Following the official tutorials is recommended, as they walk the user through the whole process. For the installation of headless version of the operating system, SSH and wireless networking needs to be enabled when creating the operating system image. All the other actions are equivalent to those of the standard version installation. When the installation is finished, users can login and control their Raspberry Pi using SSH.[39]

4.1.2 Flashing firmware on the CC2530

The recommended way of flashing the CC2530 with the Zigbee2MQTT firmware is using the *CC Debugger* from Texas Instruments. However, CC Debugger is only an additional cost, and is needed only for the flashing process. Another option is to use GPIO pins of Raspberry Pi to flash the required firmware. This option is enabled by default on the custom radio unit board and does not require any additional wiring.

A tutorial providing all of the necessary information for the flashing process is available at [40]. However, the *CC2530_CC2592* version of firmware needs to be downloaded and flashed instead of the *CC2531* version.

4.1.3 Installation of Home Assistant Supervised

To get Home Assistant running on Raspberry Pi, Home Assistant Supervised method of installation was chosen. Tutorial for this installation can be found at [41]. It guides through all the steps, and, afterwards, the Home Assistant is ready to use. It can be reached by typing *http://ip_address:8123/* into a web browser from any device that is connected to the same network as the Raspberry Pi. The actual IP address of the Raspberry Pi needs to be inserted in that command, for instance *http://192.168.1.5:8123*.

4.1.4 MQTT Broker and Zigbee2MQTT add-on setup

With Home Assistant running, connection of devices through the Zigbee network has to be enabled. In order to do so, Mosquitto Broker add-on and Zigbee2MQTT add-on needs to be installed. This can be done easily in the Supervisor section of Home Assistant. When both add-ons are installed, they require to be configured to communicate with each other.

Choosing Mosquitto Broker in the Supervisor menu allows to change its configuration. As showed in the figure 4.2, custom username and password needs to be set up. These credentials are then used in the Zigbee2MQTT add-on configuration.

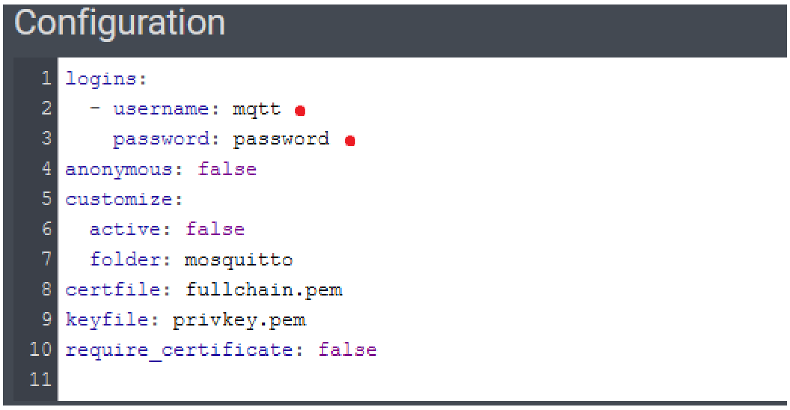

Figure 4.2: Mosquitto configuration file

Choosing Zigbee2MQTT in the supervisor menu allows to change its configuration. The add-on configuration file is shown in the figure 4.3. In the Zigbee2MQTT configuration file username and password has to be changed to match the credentials used in Mosquitto configuration. In addition, an address of the MQTT server needs to be specified in a form of *'mqtt://ip_address:1883'* and the serial port needs to be set to */dev/ttyAMA0*. To enable devices to join the network, *permit_join* is required to be set to *true*. When the connection of the new devices is created, the *permit_join* should be switched back to *false* in order to control which devices can join the network.

Once these two add-ons are configured, it is possible to connect new devices to the Zigbee network and integrate them into Home Assistant. In most cases, devices connect to a new network after they have gone through a reset. Detailed instructions for the connection of each particular device (and a list of all connectable devices) can be found at [42].

4.1.5 Multimedia server installation

Moving away from Home Assistant, the installation of PLEX multimedia server is described in this subsection. PLEX server runs on Windows, ma-

Configuration

```yaml
data_path: /share/zigbee2mqtt
devices: devices.yaml
groups: groups.yaml
homeassistant: true
permit_join: false
mqtt:
  base_topic: zigbee2mqtt
  server: 'mqtt://192.168.1.102:1883'
  user: mqtt
  password: hesloveslo
serial:
  port: /dev/ttyAMA0
```

Figure 4.3: Zigbee2MQTT configuration file

cOS and Linux and it provides access to personal media (for example movies, home videos, TV shows, photos and music) which are stored on the server. The media are streamed over the local network and can be viewed on any device that is connected to the same network. It can also be set up to be accessed from outside the local network. PLEX media library can be viewed from any web browser or dedicated apps that can be installed on smartphones, tablets or smart TVs. The media stored on a PLEX server are also accessible from third-party applications via DLNA stream.

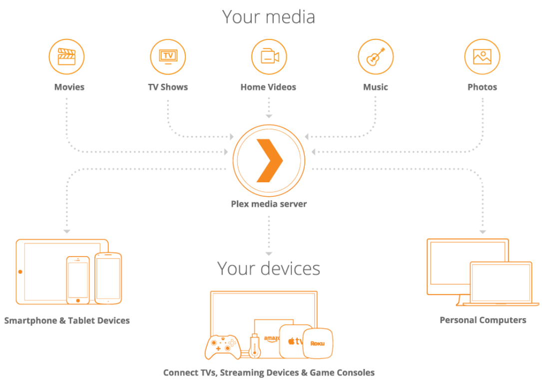

Figure 4.4: PLEX multimedia server [15]

A detailed installation guide is available at [43]. Media can be added by inserting a USB stick to the Raspberry Pi or the Raspberry Pi can act as a NAS, so media files can be transferred over the network. The NAS set up is described in the following subsection 4.1.6.

4.1.6 Network attached storage setup

The last component that needs to be set up is Network attached storage (NAS). Many different platforms can be used to provide network file transfers. Samba was chosen because it provides simple set up and can be integrated into Home Assistant.

Samba uses SMB/CIFS protocols to share files and directories over the local network. After the initial setup, which can be done by following the setup guide at [44], files can quickly be transferred from the Raspberry Pi to any device in the same network and vice versa. As a storage for Raspberry, USB stick was plugged in, however, different types of storage, such as external USB hard drives, can be used.

4.2 Central Home Hub overview

The Central Home Hub is a product which integrates many functions in a single system. It is embedded in a fashionable tablet-like device that users can place almost anywhere in their homes. All it needs is to be positioned in the vicinity of a power outlet and, ideally, near an Ethernet chord. Most of its functions can be either viewed or controlled from a touch screen display. Nevertheless, Central Home Hub can also be controlled from any laptop, PC or tablet that is connected to the same network as the Central Home Hub, or from a smart phone which additionally allows remote access, therefore the possibility of control from anywhere in the world.

4.2.1 Home automation function

The Central Home Hub can communicate with a vast number of connected devices. For home automation, different sensors, actuators and lights are connected. This gives the opportunity to monitor the environment in the household and to control it at the same time. Furthermore, since the Central Home Hub runs Home Assistant, advanced automation can be established and users can replace many manual tasks with automated ones.

Even though the capabilities of the Central Home hub are immense, due to a restricted budget for this thesis project, only a few devices were used. There is a possibility to connect many more devices, for instance, air conditioning, heating systems and electrical window blinders can be integrated in the home automation system. The possibilities are limited only by the amount of devices available on the market and their abilities.

Even with a small set of sensors and lights an interesting and useful home automation solution was created. After sunset lights are automatically turned on when a person enters the apartment. The lights can also be controlled from a smartphone with the use of speech recognition and voice-control. A notification to the user's smartphone and SMS is sent in case of an alert (e.g.water leak in the bathroom). Users can view whether the windows are closed or kept open. With more devices connected, the automation can grow, and users can create new scenarios that would make their lives easier.

To view what is going on in the household and to control it, a dashboard was created. The dashboard can be viewed on any of the previously mentioned device. The dashboard is divided into three tabs. The first tab shows temperature, humidity and air pressure in the apartment, lights' status is displayed and can be controlled with a single click. Moreover, time and date information is presented, as well as the active location of the members of the household. Weather forecast is available in the second tab. Users can view the weather forecast for the next few hours or for longer periods of time. Detailed weather data, including the wind speed and cloudiness, are also provided. This tab is shown in the figure 4.5. The last tab displays data related to security. Alarm control panel and camera live view are complemented with occupancy sensors and door contact statuses. A button that for 10 minutes allows devices to join the network is also present in this tab.

Figure 4.5: Weather forecast tab

4.2.2 Home security function

Another important functionality of the Central Home Hub is the home security system. It was tested with two PIR occupancy sensors and a door contact sensor, however, many more sensors and detectors can be incorporated. Furthermore, an IP camera and a smoke sensor are included in the system, which provides additional security.

The alarm system is created in Home Assistant using a *Bwalarm* custom component, which is available in the Home Assistant community store. The component creates an alarm control panel through which users arm and disarm their alarm system. The alarm control panel is shown in figure 4.6.

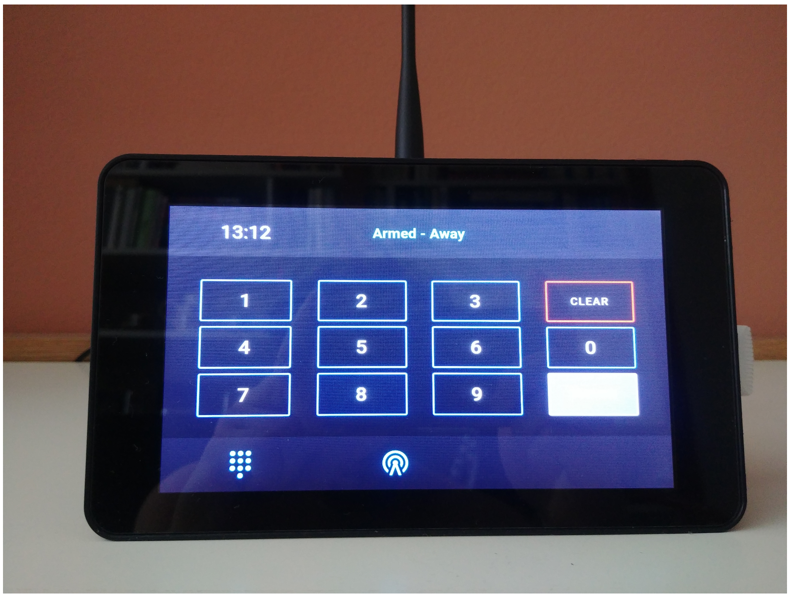

Figure 4.6: Alarm control panel

The Bwalarm component allows users to adjust their alarm settings. Arming code can be changed here, as well as pending, warning and trigger times. This makes the alarm system fully adjustable. The alarm is automatically disarmed when user arrives at home. With automations, the security system allows to send SMS and notification if a complication appears. The connected camera (using RTSP streaming) displays live view of a certain room, thus remote visual control achieved. The connected GSM module and power backup ensure that the system is resistant to power outages.

It is questionable whether a low-cost and minimalist solution of a home security system created in this thesis is comparable to professional solutions. For this reason, long-term behaviour and stability of the designed system will be monitored and evaluated.

4.2.3 Multimedia server and NAS

Home automation and home security systems are not the only ones that Central Home Hub offers. The Central Home Hub can also turn into a multimedia server and feed other devices at home with the media content that is stored within the server. In addition, the Central Home Hub works as a NAS station, which means that files and directories can be uploaded to the cloud from any device in the home network.

The PLEX media server enables video, music and photo streaming to smartphones, TVs etc. Video streaming of a 4K movie was tested with a very positive and satisfactory outcome. The streaming was carried out without any issues, even when another stream of a Full HD video to another device was running, at the same time.

4.3 Zigbee network

Important part for the appropriate function of the Central Home Hub is the Zigbee network. Most devices used in the solution are using Zigbee to communicate with the central unit. For this reason, considerable attention was dedicated to hardware design of the Zigbee radio unit, as was described in section 3.2.2.

Nevertheless, the first idea was to use a Zigbee USB dongle CC2531, which can be purchased for only several dollars. The decision to use a custom designed Zigbee radio was made regarding two reasons. The first, and less significant, - to make the appearance of the product likable without an overhanging PCB plugged in a USB port of the Raspberry Pi. The second reason was the inadequate performance of this Zigbee unit. The USB dongle was tested in a small apartment and even there its range did not cover the whole area.

With the custom radio board performance of the Zigbee network improved significantly. In places where sensors stopped communicating with the USB dongle, the new solution had no problem to remain connected to the sensors. Furthermore, the range of the network increased for approximately 6 - 10 meters, depending on how many walls the signal had to penetrate and interferences in the particular place with other networks in the same frequency band.

One of the biggest advantages of the provided solution is that devices from many different vendors can be connected to the created gateway. Technically, any Zigbee device can be connected to the gateway. This statement is difficult to prove. Testing a vast number of devices would exceed the budget of the thesis very quickly. For this reason only devices from Xiaomi Aquara, Xiaomi Mijia, Honeywell and Philips Hue were purchased. However, for the purpose of testing, Osram and IKEA TRÅDFRI devices were borrowed and their compatibility was proven.

Diagram of the created Zigbee network, consisting of the previously mentioned devices and the radio unit coordinator, is shown in figure 4.7.

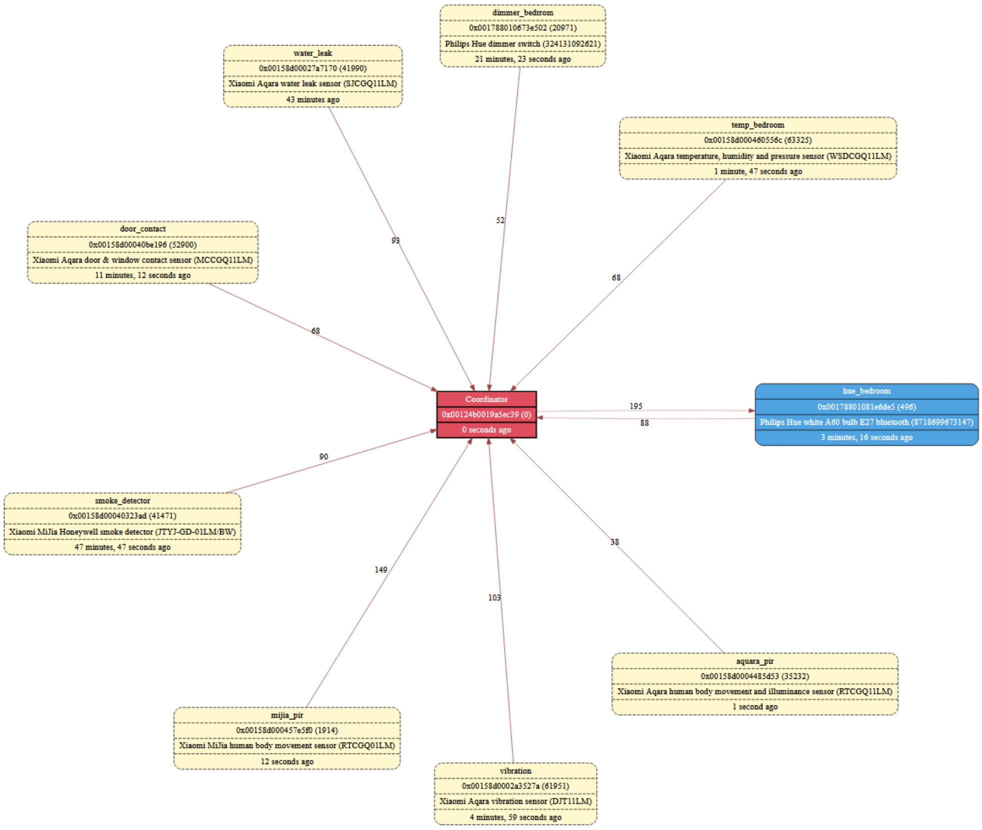

Figure 4.7: Zigbee network

4.4 Comparison with other commercial products

Nowadays, customer can choose from a large variety of home automation solutions. The biggest issue is with the flexibility of these solutions. Some products create a closed home automation environment, whereas other products aim to create an environment that could connect any device on the market. With the ever growing portfolio of smart devices, the situation becomes more and more complicated, and a platform, that would be compatible with all of them, is almost impossible to make.

The greatest advantage of the Central Home Hub is its flexibility. It enables connection of any device which uses either Zigbee, Bluetooth or Wi-Fi. Even though it does not support the Z-Wave protocol, equivalent devices can be connected using Zigbee in most cases. Home Assistant provides highly customizable platform for automation scenarios, moreover the web interface with the smart phone app and remote access is a perfect solution for the system control. Central Home Hub also supports the integration of a Google Assistant or Amazon Alexa, which enables voice control and further customization.

Most commercial products have limited support for security systems, more specifically, a dedicated security system has to be connected, which makes the solution far more expensive. Just a few commercial products are connected with a display or a tablet, thus their control is limited to their web interfaces or smart phone apps. No home automation solution in the market offers the additional functionalities of a multimedia server and NAS station, which makes the Central Home Hub a unique product.

Another aspect that makes the Central Home Hub a better solution is its price. The expenses of making this product (without the connected sensors) were approximately 200 USD, however, production of a bigger series would decrease the cost. Comparably flexible solutions of home automation (without additional functions) such as *Athom Homey Pro 2.0* cost more than 450 USD.

Customer who wants to build a home automation system can choose from different kinds of solutions. The cheapest way is to choose only one vendor and buy their gateway and devices. For instance, *Aquara Hub* will work only with Aquara smart devices, however, the gateway can be purchased for only around 70 USD. Even cheaper *Immax NEO BRIDGE PRO Smart Zigbee 3.0* is compatible with more Zigbee devices, yet not all of them are supported. Furthermore, the system does not provide such customization as Central Home Hub and can be controlled only from a smart phone or a tablet. More expensive solutions such as *Athom Homey Pro 2.0* or *FIBARO Home Center Lite* provide similar flexibility to the Central Home Hub, nevertheless, for a substantially bigger price and without the multimedia server and NAS station functionalities.

Conclusion

The objective of the thesis was to study possibilities of home control and home automation and then, based on the obtained knowledge, to design a custom device which would be capable of running a home automation system, a media server and a home security system. The final product is called Central Home Hub.

In the first chapter, I explained the idea of the Internet of Things. By describing different protocols which are used for device to device communication, I introduced the ambition that a system, which targets to provide advanced control and automation functions, has to fulfill. With definition of Zigbee, MQTT and Zigbee2MQTT, I shared the basic knowledge of these protocols, which is necessary to understand the functionality of the developed system.

Essential set of information about different home automation platforms is provided in the second chapter. In this chapter, reader gets to know the three most used open source home automation platforms that can be implemented into the developed system. With the deep explanation of the installation, configuration and customization, I presented the procedure of steps that are needed to be done in order to use the Home Assistant platform.

In the next chapter, details of the hardware implementation of the Central Home Hub are explained. I focused on the advantages and features that the single board computer Raspberry Pi provides and explained why I decided to use it in my solution. I also described the custom radio frequency hardware design of the created Zigbee radio unit and the challenges I had to overcome. Furthermore, the third chapter presents all the connected hardware components, including all the sensors and devices that enable home automation.

The last chapter provides the reader with all the steps that had to be done in order to successfully create the Central Home Hub. Later, the advantages of the designed system are described and how users can benefit from using the product. Possible capabilities of the Central Home Hub and comparison with alternative products are discussed at the end of the chapter.

The final device corresponds with the requirements stated in the thesis assignment. The Central Home Hub is a standalone solution that provides its users with a security, regulatory, control and multimedia functions. The product acts as a universal hub through which extensive supply of devices and platforms can be united in one system. The biggest advantage of the solution is that users do not have to buy expensive proprietary gateways to integrate each particular set of devices, on the contrary, they can use Central Home Hub as one universal gateway that substitutes them all.

Submission of the thesis is not the ending point of the project. I, personally, will keep working on the project in order to achieve even bigger flexibility. In addition, I decided to write documentation to the project and share it with the GitHub and Home Assistant community.

Bibliography

[1] Saxena, S.; Jain, S.; Arora, D.; aj.: *Implications of MQTT Connectivity Protocol for IoT based Device Automation using Home Assistant and OpenHAB*. 2019, 475-480 s.

[2] *Home automation networks: A survey*. [cit. 2020-08-12]. Avaiable from: https://www.sciencedirect.com/science/article/pii/S0920548916300654

[3] *ZIGBEE Architecture (ZIGBEE Stack) – All Layers and its Functions*. [cit. 2020-08-12]. Avaiable from: https://electricalfundablog.com/zigbee-architecture-zigbee-stack-layers/

[4] *What is ZIGBEE Technology in IoT – Architecture, Network Topologies, Applications*. [cit. 2020-08-12]. Avaiable from: https://electricalfundablog.com/zigbee-technology-architecture/

[5] *IT Explained: MQTT*. [cit. 2020-08-12]. Avaiable from: https://www.paessler.com/it-explained/mqtt

[6] *MQTT (MQTT: univerzální protokol nejen pro cloudové aplikace)*. [cit. 2020-08-12]. Avaiable from: https://automatizace.hw.cz/mqtt-univerzalni-protokol-nejen-pro-cloudove-aplikace.html

[7] *MQ Telemetry Transport*. [cit. 2020-08-12]. Avaiable from: https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport

[8] *Zigbee2MQTT GitHub repository*. [cit. 2020-08-12]. Avaiable from: https://github.com/Koenkk/zigbee2mqtt

[9] *Home Assistant blog*. [cit. 2020-08-13]. Avaiable from: https://www.home-assistant.io/blog/2014/12/26/home-control-home-automation-and-the-smart-home

[10] *Domoticz Forum*. [cit. 2020-08-12]. Avaiable from: https://www.domoticz.com/forum/

[11] *openHAB Documentation*. [cit. 2020-08-12]. Avaiable from: https://www.openhab.org/docs/

[12] *Home Assistant Documentation*. [cit. 2020-08-13]. Avaiable from: https://www.home-assistant.io/docs/

[13] *Raspberry Pi 4*. [cit. 2020-08-13]. Avaiable from: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/

[14] *Raspberry Pi 4 Review*. [cit. 2020-08-13]. Avaiable from: https://www.pcmag.com/reviews/raspberry-pi-4

[15] *What is Plex?* [cit. 2020-08-13]. Avaiable from: https://support.plex.tv/articles/200288286-what-is-plex/

[16] *Automation*. [cit. 2020-08-11]. Avaiable from: https://www.britannica.com/technology/automation

[17] *THE INTERNET OF THINGS: AN OVERVIEW*. [cit. 2020-08-12]. Avaiable from: https://www.internetsociety.org/wp-content/uploads/2017/08/ISOC-IoT-Overview-20151221-en.pdf

[18] *Comparison of Home Automation Protocols*. [cit. 2020-08-12]. Avaiable from: https://theiotpad.com/tips/home-automation-protocols

[19] *About Z-Wave Technology*. [cit. 2020-08-12]. Avaiable from: https://z-wavealliance.org/about_z-wave_technology/

[20] *Bluetooth Vs. Bluetooth Low Energy: What's The Difference?* [cit. 2020-08-12]. Avaiable from: https://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy

[21] *What is Thread*. [cit. 2020-08-12]. Avaiable from: https://www.threadgroup.org/What-is-Thread

[22] *OpenThread*. [cit. 2020-08-12]. Avaiable from: https://openthread.io/

[23] *A survey on Zigbee based wireless sensor networks in agriculture*. [cit. 2020-08-12]. Avaiable from: https://ieeexplore.ieee.org/document/6169090

[24] *ZigBee wireless area network for home automation and energy management: Field trials and installation approaches*. [cit. 2020-08-12]. Avaiable from: https://ieeexplore.ieee.org/document/6465691

[25] *Zigbee2mqtt documentation*. [cit. 2020-08-12]. Avaiable from: https://www.zigbee2mqtt.io/

[26] *Z-StackMonitor and Test API*. [cit. 2020-08-12]. Avaiable from: http://dev.ti.com/tirex/explore/node?node=AEEJ0Z3BgvagEoglvx9r1g__pTTHBmu__LATEST

[27] *Domoticz Wiki page*. [cit. 2020-08-12]. Avaiable from: https://www.domoticz.com/wiki/Main_Page

[28] *openHAB - Who We Are*. [cit. 2020-08-12]. Avaiable from: https://www.openhab.org/about/who-we-are.html

[29] *openHAB Installation Overview*. [cit. 2020-08-12]. Avaiable from: https://www.openhab.org/docs/installation/

[30] *Home Assistant Community*. [cit. 2020-08-13]. Avaiable from: https://community.home-assistant.io/

[31] *The State of the Octoverse*. [cit. 2020-08-13]. Avaiable from: https://octoverse.github.com/

[32] *OPENHAB VS HOME ASSISTANT VS DOMOTICZ*. [cit. 2020-08-13]. Avaiable from: https://www.smarthomeblog.net/openhab-home-assistant-domoticz/

[33] *Raspberry Pi hardware*. [cit. 2020-08-13]. Avaiable from: https://www.raspberrypi.org/documentation/hardware/raspberrypi/

[34] *Raspberry Pi Power Limitations*. [cit. 2020-08-13]. Avaiable from: https://raspberrypi.stackexchange.com/questions/51615/raspberry-pi-power-limitations

[35] *TS1117B Series Datasheet*. [cit. 2020-08-13]. Avaiable from: https://cz.mouser.com/datasheet/2/395/TS1117B_H1607-1114639.pdf

[36] *CC2530 product page*. [cit. 2020-08-13]. Avaiable from: https://www.ti.com/product/CC2530#product-details#features

[37] *CC2592 product page*. [cit. 2020-08-13]. Avaiable from: https://www.ti.com/product/CC2592#product-details#features

[38] *SimpleLink™ ZigBee® Network Range Extender Reference Design*. [cit. 2020-08-13]. Avaiable from: https://www.ti.com/tool/CC2530-CC2592EM-RD

[39] *Setting up a Raspberry Pi headless*. [cit. 2020-08-13]. Avaiable from: https://www.raspberrypi.org/documentation/configuration/wireless/headless.md

[40] *Alternative flashing methods*. [cit. 2020-08-13]. Avaiable from: https://www.zigbee2mqtt.io/information/alternative_flashing_methods.html

[41] *Installing Home Assistant Supervised on Raspberry Pi OS*. [cit. 2020-08-13]. Avaiable from: https://community.home-assistant.io/t/installing-home-assistant-supervised-on-raspberry-pi-os/201836

[42] *Zigbee2MQTT supported devices*. [cit. 2020-08-13]. Avaiable from: https://www.zigbee2mqtt.io/information/supported_devices.html

[43] *How to Setup a Raspberry Pi Plex Server*. [cit. 2020-08-13]. Avaiable from: https://pimylifeup.com/raspberry-pi-plex-server/

[44] *How to Setup a Raspberry Pi Samba Server*. [cit. 2020-08-13]. Avaiable from: https://pimylifeup.com/raspberry-pi-samba/

Appendix A

List of acronyms

IoT Internet of Things

MQTT Message Queuing Telemetry Transport

IEEE Institute of Electrical and Electronics Engineers

BLE Bluetooth Low Energy

6LoWPAN IPv6 over Low -Power Wireless Personal Area Networks

Wi-Fi Wireless Fidelity

RF Radio Frequency

M2M Machine to Machine

HAN Home Area Network

TCP Transmission Control Protocol

IP Internet Protocol

PHY Physical Layer

MAC Medium Access Control

QoS Quality of Service

DSSS Direct Sequence Spread Spectrum

O-QPSK Offset Quadrature Phase-Shift Keying

TLS Transport Layer Security

SSL Secure Sockets Layer

RFD Reduced Functioning Device

FFD Full Functioning Device

API Application Programming Interface

JSON JavaScript Object Notation

YAML YAML Ain't Markup Language

PCB Printed Circuit Board

NAS Network Attached Storage

IFTTT If This Then That

GPIO General Purpose Input/Output

SoC System on Chip

DNS Domain Name System

CPU Central Processing Unit

GSM Global System for Mobile Communications

UART Universal Asynchronous Receiver-Transmitter

SSH Secure Shell

DLNA Digital Living Network Alliance

Appendix B

Custom radio unit board schematic

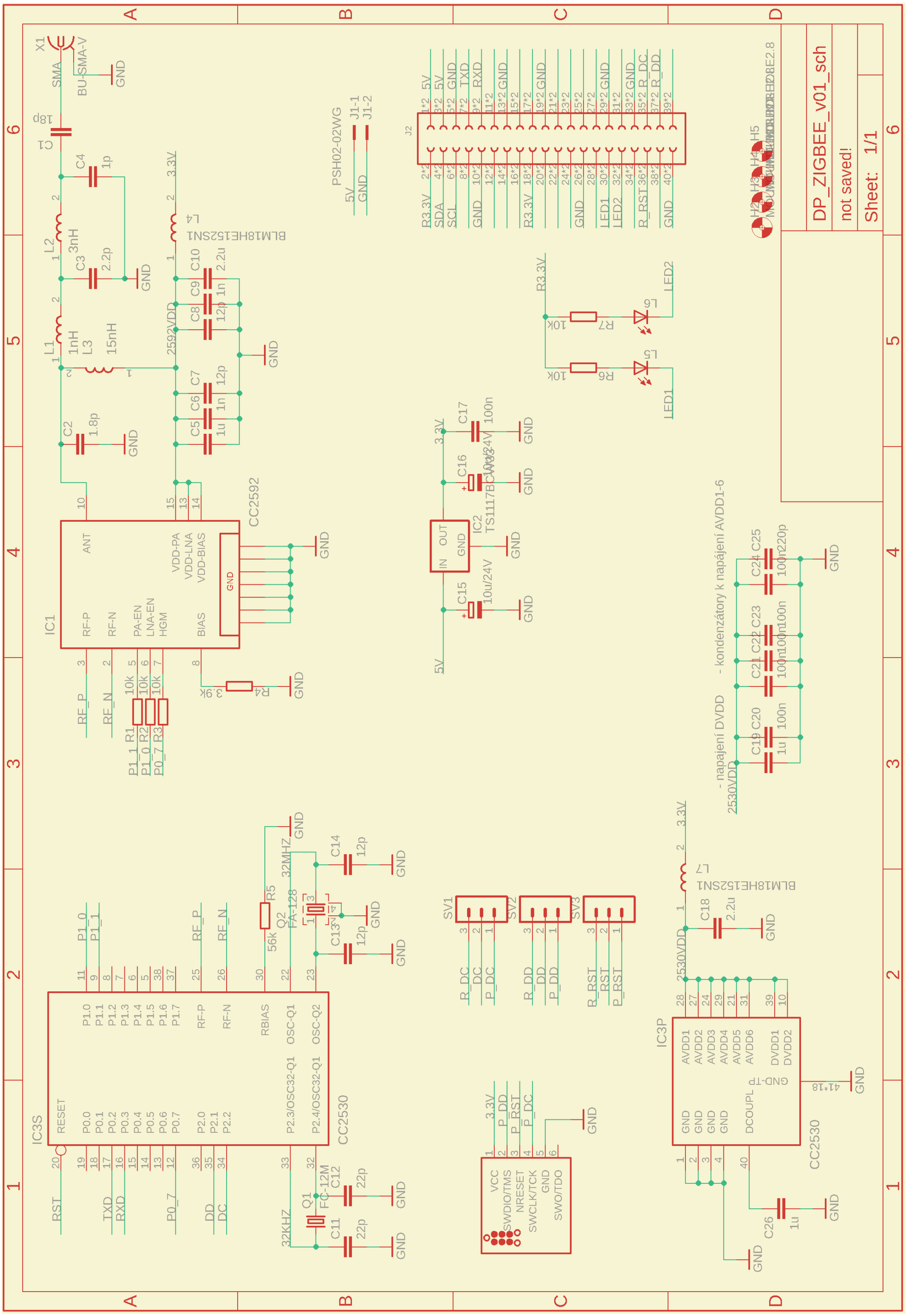

Figure B.1: Custom radio unit board schematic

Appendix C

Custom radio unit board PCB layout

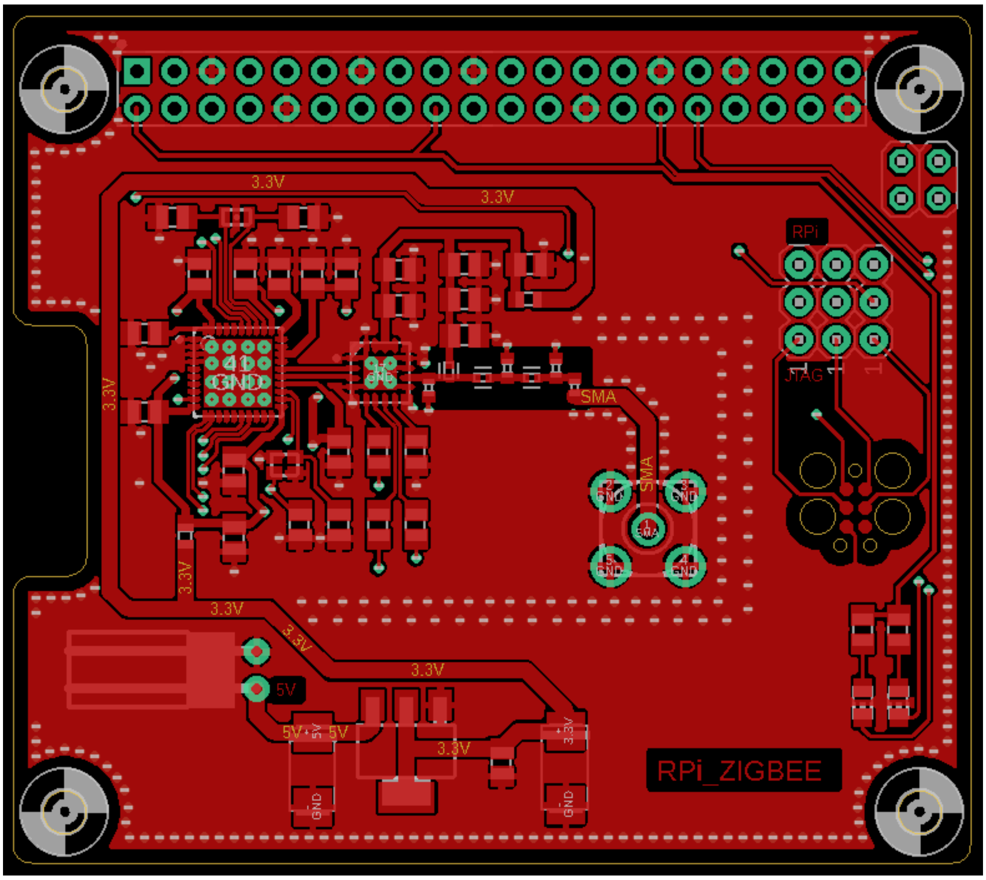

Figure C.1: PCB layout top layer view

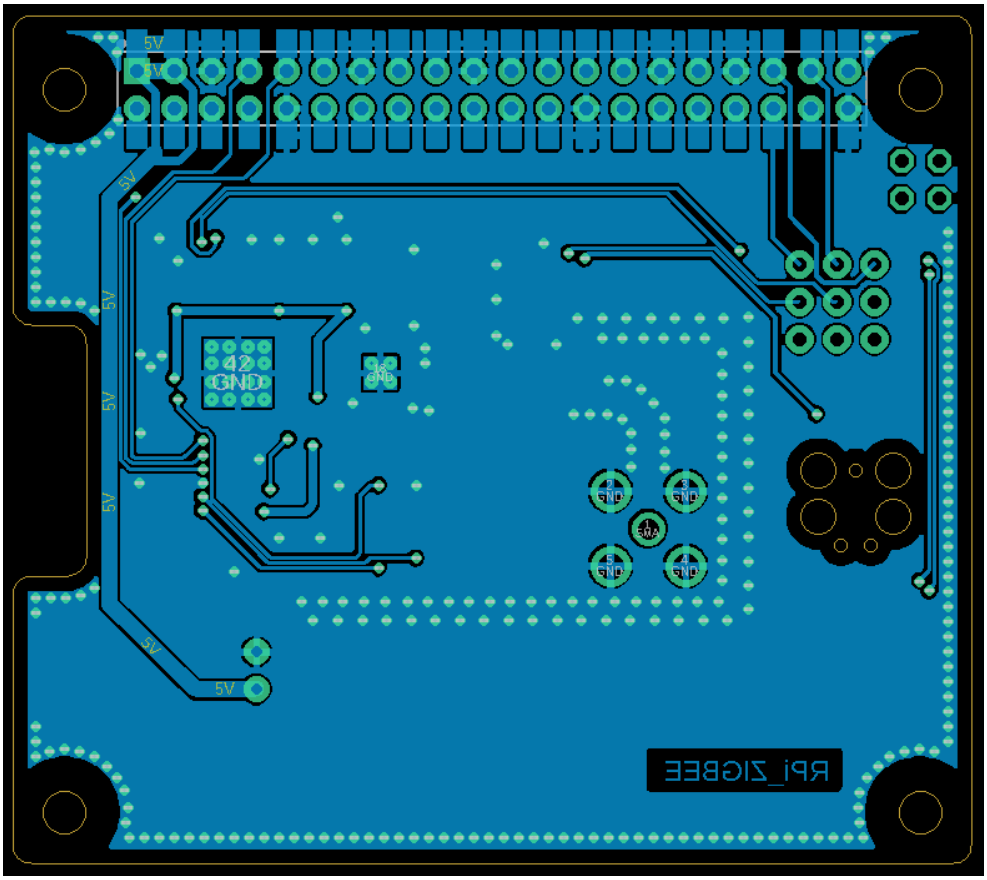

Figure C.2: PCB layout bottom layer view