UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Enrih Sinilaid

Monitoring and controlling smart home appliances using IoT devices

Bachelo's Thesis (9 ECTS)

Supervisor: Chinmaya Kumar Dehury

Tartu 2021

Monitoring and controlling smart home appliances using IoT devices

Abstract:

Usage of different smart home appliances and systems is becoming increasingly more popular in many households. There are many key points for rising number of adopters. For some it is the price as these systems and appliances are not that new to the market anymore and thus are more reasonably priced. Also, with the development in both hardware and software areas processing and memory units have become both faster and smaller which makes designing and developing different smart home devices more viable for building and selling commercially. This ensures that this smart home systems market is not dominated by few companies and gives a chance for anyone to try them out price wise. The second key point is the versatility of different smart home devices that are out in the market, ranging from lights to home automation and security. This variability lets people start out with few cheaper products like smart lighting or media devices and see if this is something for them.

Smart home appliances are devices that could be a common sight at many households like lights, speakers, TVs, air conditioners and so on but what makes them different is the built-in functionality for connecting to internet and then be monitored and controlled remotely. This ability to be remotely controlled and monitored makes it possible to develop automations that could further enhance the way these devices are used.

The aim of this thesis was to try and connect different smart home devices to one central system that could let the user of this system to control and monitor different smart home appliances and devices using their smartphones or computers. Additionally, this thesis aims to develop and showcase how to try and automate these devices from the central system. This automation will try to control the energy consumption of these smart devices by the user's location so that if user is not present then some devices could turn off or start using power saving profile. The central system will be hosted by IoT device running OpenHab OS.

Keywords:

openHAB, openHABian, IoT, Raspberry Pi, SSH, MQTT, Python, Home Assistant, Apple HomeKit

CERCS:

Targa kodu seadmete seire ja juhtimine kasutades asjade interneti seadmeid

Lühikokkuvõte:

Erinevate nutikate kodumasinate ja -süsteemide kasutamine on muutumas paljudes majapidamistes üha populaarsemaks. On mitmeid faktoreid, miks nende kasutusele võtjate arv kasvab. Mõne jaoks on see hind, sest need seadmed pole enam turul uued, seega ka enamikel seadmetel on hinnad palju käepärasemaks muutunud. Samuti on nii riist- kui ka tarkvara arengu tõttu protsessorid ja mäluüksused muutunud kiiremaks kui ka väiksemaks, mis muudab erinevate nutikodu seadmete kujundamise ja arendamise elujõulisemaks äride jaoks See tagab, et vähesed suuremad ettevõtted ei domineeri nutikodu süsteemide turgu, andes võimaluse ka väiksematel ettevõtetel oma lahendusi luua ja müüa, mis annab kõigile võimaluse neid seadmeid hinnatarkalt proovida.

Teine faktor on mitmesuguste turul olevate nutikate koduseadmete mitmekülgsus, alates tuledest kuni koduautomaatika ja turvalisuseni välja. See varieeruvus võimaldab inimestel alustada mõne odavama tootega, näiteks nutivalgustuse või meediumiseadmetega, ja vaadata, kas see on midagi nende jaoks.

Nutikad kodumasinad on seadmed, mis võivad olla paljudes majapidamistes tavalised nähtused, näiteks valgustid, kõlarid, telerid, konditsioneerid ja nii edasi, kuid mis muudab need erinevaks on sisseehitatud funktsionaalsus interneti-ühenduse loomiseks ning seejärel lasta kasutajal neid kaugelt juhtida ja seirata. See kaugjuhtimise ja seiramise võimalus võimaldab välja töötada automaatika, mis võiks veelgi täiendada nende seadmete kasutamist.

Selle lõputöö eesmärk oli proovida ühendada erinevad nutikodu seadmed ühte kesksüsteemi, mis võimaldaks selle süsteemi kasutajal juhtida ja jälgida nutitelefonide või arvutite abil erinevaid nutikaid kodutehnikat ja -seadmeid. Lisaks on selle lõputöö eesmärk välja töötada ja tutvustada, kuidas neid seadmeid kesksüsteemi abil automatiseerida. See automaatika püüab nende nutiseadmete energiatarbimist kasutaja asukoha järgi kohandada, nii et kui kasutajat pole kohal, saaksid mõned seadmed välja lülituda või hakata energiasäästuprofiili kasutama. Keskne süsteem hoiustatakse IoT-seadmel, milles töötab OpenHab operatsioonisüsteem.

Märksõnad:

openHAB, openHABian, IoT, Raspberry Pi, SSH, MQTT, Python, Home Assistant, Apple HomeKit

CERCS:

Table of Contents

1. Introduction

With the progress in various information technology fields, many aspects of human society have changed. Access to information has never been quicker and easier as devices capable of accessing information anywhere with internet coverage have become very common. Today many devices around us can receive, store, process and transfer data to connected devices . This system is known as the internet of things (IoT). Over the years, the popularity in this field has been increasing rapidly. According to the GSM Association, it is expected that the number of IoT devices will grow to 25.1 billion by 2025 [1].

One of many popular real-world applications for IoT is in smart home ecosystems. Through IoT, it becomes possible to monitor, control and automate smart home devices at some level, even for private consumers with no prior knowledge in this field. A large part of the popularity also stems from the fact that it is possible to create smart home systems with relatively more nominal cost than in the past. This is due to the development and the broader availability of older technologies. Adroid Market Research, a global market research firm, points out that smart home system's take-up is growing and that smart home systems' global market is expected to exceed USD 95 billion by 2025 [2].

A smart home is essentially a home where through wireless technologies, various appliances and devices can be monitored and controlled using mobiles or other networked devices. Data generated by sensors and smart home appliances make it possible to monitor conditions at home and its devices. Devices themselves can communicate with each other or with smart home central platform through WIFI and Bluetooth. This enables for smart home solutions where through IoT capabilities, more complex automations are possible.

Smart homes are not only more popular due to the capability and affordability of smart devices but also due to the fact that there are open-source smart home platforms. Some of these platforms are easy to implement and do not require huge investment. They can enable the user to tie in existing smart systems in the house to one central system. These platforms work by using wireless technologies and IoT capabilities to communicate with smart solutions from different ecosystems.

The flexibility of some open-source smart home platforms is not only limited to connecting together different smart systems in homes. With some research and time, users can try to build their own solutions on top of these platforms or implement advanced automations. This level of flexibility gives the users a certain degree of control of how their smart home systems handles information flow around the house. This enables users to have better control over their privacy and be sure that their private information will not reach any third party.

The degree of control over the private information gives certain open-source smart home platforms advantage over commercial smart home platforms. This is not the only advantage of choosing open-source platform for smart home. Open-source solutions often have better transparency than commercial solutions. This means that users could have better idea how open-source smart home platform works and how it works behind the scenes, thus possibly gaining better trust than their commercial counterparts.

Although smart home platforms importance in smart home is not low, it would be almost useless without any devices to control, monitor or automate. These devices can almost be anything in the house that could gather information, use information to accomplish some task or perform simple actions when triggered. These devices could be smart entertainment, climate, kitchen, security devices and the list does not end there.

With microcontrollers and single-board computers, even normal devices could act as smart devices. As an example, it is possible to monitor ordinary devices and appliances' energy usage and implement a switch for turning them off and on. This allows for creating a home system with efficient electricity usage and thus lowers the overall cost in electricity. Due to that, home owners could invest more into field of smart homes without having to worry about accumalating huge electricity bills.

1.1 Aim of the thesis

The aim of this thesis is to create smart home system that could monitor and control smart home appliances through the use of IoT. To better demonstrate the IoT capabilities in the smart home system, a user tracking system will be developed. This system will collect users location data in real-time, which will be used to control devices around the user. This thesis uses an open-source home automation platform and various single-board computers to handle smart home devices and track users. As the tracking should differentiate between different users, users are located via their smartphones using either WIFI or Bluetooth. Communication between the home automation platform and user tracking devices is done by using MQTT messaging protocol.

1.2 Outline of the thesis

This thesis consist of three stategs that are:

- Setting up home automation platform. In this stage an open-source platform OpenHAB is used for monitoring, controlling and automating various smart home devices across the house. As such this stage will consist of exploring OpenHAB functionalities and possibilities, setting up the OpenHAB, connecting devices into it and implementing automation and UI for controlling and monitoring smart home devices. This stage will also describe any problems or findings found.
- Developing user tracking system. This stage will cover how house user tracking system was developed. This will cover the research into different approaches and what approach was used in the end. As such there will also be steps taken in development phase and all risen problems will also be covered.
- Connecting home automation and user tracking system. In this stage the method of connecting the home automation and user tracking system will be described. Also this stage will also contain steps for creating automations using the data that user tracking system provides.

As for thesis structure it is as following:

- Background, section for information about the technologies and tools used in this thesis.
- System specification, section describing use cases and requirements for this system developd in this thesis.
- House automation system, section about setting up home automation platform, connecting smart home devices into it and automating said devices.
- User tracking system, section about developing system for tracking users in the house.
- Final assembely and automation, section about how location data is sent between systems and used in automating smart home devices.
- Summary, section describing work and suggestions for future research.

2. Background

This section will give information about the system built in the thesis and introduce different components and technologies used in the implementation. This section will cover different open-source house automation platforms and explain why OpenHAB was used in the end. Next, there will be information about what tools and hardware will be used for the user tracking system. Finally, the smart home devices that are going to be used in this thesis will be introduced.

2.1 Idea

This thesis aims to create an IoT-based smart home central system that could control, monitor, and automate different smart home devices in the house. This system will make use of a small single-board computer and open-source platform capabilities of good modularity, easy development and low cost. The user tracking system will be built with the purpose to gather data about users whereabouts in the house. This data will be used to automate smart devices around the house to respond when users are near them or have left their range. The main purpose of interaction between user location tracking and smart appliance control is to show what IoT-based smart home system is capable of.

Not only is the location of users monitored but also the smart home devices. The smart home system will use the data gathered from each smart home device and enable a manual control panel for house users to access. Said control panel would show each device's state, have an option to control these devices manually, and possibly show any data that is useful for the user, for example, time, date, the climate in and out of the house, energy consumption and more. This control panel will be able in both web and mobile application form.

An example of a possible scenario that should be possible with this system: Joe arrives at home. The system detects that and will turn on lights in the vestibule and cloakroom. Next, as it is a warm summer day, the system will turn the air conditioner on. As Joe reaches his room, the system will turn off lights in previous locations and turn on the Joes room lights. As Joe likes classical music, then the system will start playing a piece from Mozarts collection.

After some time, Joe leaves, and the system will turn off all the active devices and as indoor temperature is still relatively high, the air conditioner is set to run in an eco mode for more efficient power consumption.

Later, Joes wife reaches home and starts doing laundry. As she is working in the laundry room and going around the house collecting old bedsheets, the system automatically switching on and off lights in the rooms she is visiting. In Joe's room system only turns on lights and not the music as the room visitor is not Joe.

According to this example, the system should work with multiple users and should be able to differentiate between them to make customised automations possible. The following subsections will introduce the components and tools used for making this system possible.

2.2 Home Automation Platform

A home automation platform is a system that is responsible for monitoring and controlling home devices like lightning, climate, entertainment systems, appliances and even home security. These devices are typically connected to a central hub that is managed by a house automation platform. There are many competing vendors for these platforms, and there is even several open-source systems. Home automation has a high potential to lead towards energy-efficient solutions that could positively impact the future [3].

2.2.1 Platforms

2.2.2 Chosen Platform

2.3 User tracking system

This thesis aims to show what IoT is capable of accomplishing in a smart home environment. This thesis will develop a user tracking system that could gather data about users whereabouts around the house and transmit that data through MQTT to the home automation platform. The user tracking system will essentially allow for more complex automations and smart home system control thanks to IoT capabilities.

User tracking systems can be implemented in a variety of different forms. One way to divide them would be a systems that either could or could not differentiate between users. Both systems have their use-cases, methodology and implementations. For a system that could not acknowledge one user from another, it is usually implemented to sense movement or existence of entity around the system. For this, these systems would use sensors that could detect movement or heat emission from a foreign entity. An example of where these systems are most often found is in security to detect intrusion.

The tracking systems that could differentiate between users in its detection range are usually implemented to keep track of users' whereabouts. This is done for a couple of reasons. One of them is for security. If it is possible to know where somebody is, then it is possible to seal off access to areas they are not privileged for. Many offices use this approach to keep the personnel to their specified areas.

The second use case is for controlling and automating the environment around the user. This is mainly implemented in smart home solutions for more complex, personalised and convenient automations. An example of a home automation system with this trackin system would be when one user could enter the bathroom and bath would be preheated to their preferred temperature.

For this system to differentiate between users and track their whereabouts, the system would need to read some kind of identification data from the user when they enter a new area. The data used for identification could be something that is always on the user, for example, biometrics data or an external data carrier. In the case of using biometrics, then the system could either implement face recognition through cameras or fingerprint scanners at each entrance.

Using external identity data carriers, the system could be implemented in various ways depending on the carrier. The carriers could share the data two ways, either in contact or near range. For contact range, the carriers are not active and only passively share data once they come in contact with some sort of reader. This system could be implemented through the usage of personalised keycards and card readers at each entrance. As for the near range carriers, they could be active and connect to nearby scanners to transmit identification data from range. This means that users would not need to take action at each entrance but simply pass by, and the system would record their whereabouts by itself.

For this thesis, the tracking system will be able to differentiate between users using the external identity data carriers that the system from a close range could read. This will make it possible to implement automations near the users seamlessly and be more convenient as the user does not need to perform an identification action at each entrance. The system will consist of multiple trackers, one per room, to automatically detect the user when they are near or in the room.

2.3.1 Trackers hardware

2.3.2 Trackers software

2.4 Smart Home Devices

In this thesis, due to limited amount of resources, Philips Hue smart light bulbs will be used as smart home devices. These smart lights were used as they are the only smart devices available in thesis authors home. In total 3 Philips Hue ambient and color smart lights will be used.

The Philips Hue line of smart LED lights is manufactured by Signify N.V. This line of LED lamps can have colour changing, light level, and temperature control capabilities and come in various forms. Apart from more traditional bulbs, they can also be in light strip, outdoor lamp, floodlamp, and many other forms. [14]

What makes Philips Hue lights smart is the capability for them to be controlled wirelessly. For that, there are few ways. With newer lights, it is possible to connect to them over Bluetooth and manage them through smartphone applications or computers. The second way is to have Hue Bridge, a central controller for Philips Hue lights. The Hue Bridge can be connected to home internet through an ethernet cable, and then Hue Bridge will handle each light by itself. This means that users do not need to connect to each ligh separately and easily control them through the internet.

This thesis uses only these lights as they are capable of different automations and thus sufficient for demonstrating IoT capabilities in monitoring and controlling smart home devices.

3. Requirements

This section will give overview of functional and non-functional requirements. These requirements will describe what was focused on in setup and development process of automation and user tracking systems. Described requirements are based on presented idea in section 2 and goals set with thesis supervisor.

3.1 Functional requirements

This subsection will describe the functionalities that both smart home and tracking system will provide to users. Description will follow the schema of functionality and then description of functionality.

- The smart home system should show data about connected devices. This means that users should be able to see any meaningful information about the devices connected to the smart home system through use of user interface. For example, the data could be about the power state of devices or information collected by sensors.
- User should be able to manually control the devices. The devices that are connected to smart home system and can be controlled should be controllable by user. This means that user should be able to use user interface to control devices in any meaningful way.
- User should be able to add new devices to the smart home system easily. This means that smart home system should have functionality to easily add any supported device. For example this system could find new devices by looking through devices connected to network.
- The smart home system should allow writing automation scripts and rules. This means that smart home system used in this thesis should allow for writing custom automation scripts for devices connected to it.
- The smart home system should be able to store data to it. This means that the smart home system should enable for automatically or manually storing information to it or to the cloud. This is for saving states of devices for automation and monitor purposes and creating graphs using selected data collected over time.
- Guests locations should be included in location based automations in the smart home system. This means that smart home system could make use of information regarding guests location and enable generic automations based on their location.
- The smart home system should have software switches for user tracking automation. This means that it should be possible to manually turn off automations that make use of user location data. The switches could be toggeled from user interface by user.
- The user tracking system should be able to frequently share information with the smart home system. This means that the smart home system should have access to user location that is updated frequently. This enables for user location based automations that could trigger whenever user is near to a certain room or location.
- The user tracking system trackers should be plugged in and out without any problem. This means that users could unplug the trackers from the electricity and replug them without facing any problem. The trackers should automatically start scanning for users when turned on.
- The user tracking system should be able to track guests locations. This means that user tracking systems trackers would not only track the users but also any unknown user and their devices. This would enable for the smart home system automations to work with guest locations aswell.

3.2 Non-functional requirements

This subsection will describe the non-functional requirements that act as a critia for both the smart home system and the user tracking system. Descriptions will follow the schema of non-functionality and then description of non-functionality.

- The smart home system user location automation should trigger almost immediately when users location changes. This means that there should not be large time difference between user changing their location and location based automations triggering.
- The smart home system should allow for multiple automations to trigger simutainosly. This means that system could handle multiple automations at the same time which would prevent tasks from pailing up.
- The smart home system user interface should work on mobile devices. This means that using mobile devices the smart home user interface could be accessed. This could be implemented trough an mobile or web application.

4. System Architecture & Configuration

This section will cover the architecture and configuration of the home automation platform and the user tracking system. The following subsections will describe how both systems were designed and cover the steps taken to build them.

4.1 System Overview

The home automation system built in this thesis consists of two sub-systems, the home automation platform and the user tracking system. The home automation platform will be the central system that handles the smart devices and services, enables automations and user interaction. The user tracking system consists of stand-alone tracker devices, each responsibe for specific rooms in the home. This tracking system could be viewed as a service for the home automation platform. Essentially the tracking system is responsible for collecting user location data and sending that to the home automation platform for user location-based automations.

These two sub-systems communicate through the MQTT network. The network will consist of an MQTT broker, hosted by the home automation platform, and MQTT clients on each tracker. Each MQTT client on trackers will be publishing collected data to the MQTT broker. The home automation platform will also have an MQTT client that is subscribed to the broker. This enables the platform to manage sent data and use that information in automation. The described MQTT network is represented in Figure 1.

(Figure 1)

4.2 Home Automation Platform Configuration

This section will focus on the home automation platform. The following subsections will cover the overview of the system, the setup and configuration processes, adding the devices and services to the platform and implementing the automation.

4.2.1 System

4.2.2 Initial Setup

With hardware and software introduced in the previous section, the home automation platform can be set up.

The home automation platform setup on Raspberry Pi single-board computer requires the following:

- Raspberry Pi 4
- openHABian image file
- Flash tool
- 16 GB Micro-SD card
- Micro-SD card reader
- Ethernet access

To get the openHAB platform running on the Raspberry Pi 4, the openHABian image file is needed. It can be acquired on the openHAB project GitHub page[2], accessed easily through the documentation page, covering the openHABian installation process. There are multiple versions of openHABian images, but the image used in this thesis is version 1.6.1, which has version 2 of openHAB called openHAB 2. This image was at the time of installation the

[2] https://github.com/openhab/openhabian

newest available stable release. Since then, there have been new releases with updates to openHAB tools and packages. Even a new version of openHAB has been introduced, which is openHAB 3.

After the image file has been acquired, it can be flashed onto a Micro-SD card. There are plenty of tools available for flashing, but one used in this thesis is balenaEtcher[3] by balena. This tool is relatively easy to use, as it only needs few inputs from the user. Using this tool first image file was selected and then the destination. It should be noted that when using balenaEtcher, the openHABian image file should be unpackaged before flashing to a storage device.

Another requirement for flashing image on a Micro-SD card is to have a Micro-SD card with 16 GB of storage and a way for the computer to manage the flashing process. The 16 GB of storage is not mandatory for the Micro-SD card as the image does not take up a large amount of space, but is recommended by the documentation. For computer to access Micro-SD card, it needs to have a SD card reader compatible with Micro-SD cards. As the computer used for flashing purpose already had an inbuilt reader, no additional tool was necessary.

After the flashing process has finished, the next step is to set up openHAB on Raspberry Pi 4. The Micro-SD card with flashed image needs to be installed on the Raspberry Pi 4, connected to the internet either through WIFI or Ethernet cable. Using the Ethernet cable is easier as connecting with WIFI involves modifying the configuration file with WIFI SSID and password before the first boot. After the Micro-SD card is installed and a way for connecting to the internet is provided, the single-board computer can be booted.

After the Raspberry Pi 4 is booted for the first time, it will automatically set everything up. This process length is entirely based on the internet connection, Micro-SD card writes and reads speed and the Raspberry Pi computers processing capabilities. In the installation documentation, it is mentioned that this process could take around 15 to 45 minutes. It should be noted that Raspberry Pi 4 in this thesis took about 20 minutes to finish this process.

When the process finishes, the openHAB web server is set up. Web servers user interface can be accessed on any computer on network through a web browser with the network router designated IP address for Raspberry Pi 4 on port 8080. For example, when the router sets the IP address for the Raspberry device to be 192.168.1.3, then the user interface can be accessed on a web browser at 192.168.1.3:8080.

4.2.3 Configuration

4.3 User Tracking System Configuration

This section will focus on the user tracking system. The following subsections will cover the overview of the system, the setup process and finally, explain the script used for user tracking.

4.3.1 System

4.3.2 Initial Setup

4.3.3 User Tracking Script

[3] https://www.balena.io/etcher/

4.4 Faced problems

In implementing the home automation system and creating the user tracking system, there were few problems that needed to be resolved. The more significant issues on openHAB were file system corrupting and MQTT broker frequent crashes. As for the tracking system, the main problem was with IP address changing.

The most severe problem was the file system corruption on the openHAB system. This occurred after the power was cut from the openHAB system without prior safe shut down. As openHABian is Linux based system, then ungraceful shutdowns like this can easily break the file system. To overcome this, there are a couple of ways. One of them is to have a UPS, which could power the system for a short time after power is cut. During this time, the UPS could send a shutdown signal to the openHAB system and prevent file systems from corrupting.

The other way is to occasionally back up the system on an external storage device. In the event of corruption occurring, the system could be easily restored from the external backup. This solution was used in order to prevent setting up and reconfiguring a new system, should it happen again.

The problem with MQTT crashing on openHAB could not be traced to a cause. This problem seemed to be mainly because multiple clients were publishing data to broker and overwhelming it in the process. To solve this, the user locating scripts main loop was slowed down. After that, MQTT on openHAB seemed to work fine.

The only problem faced with trackers was that their IP address could change when they were rebooted. This problem made it difficult to access the trackers through an SSH session, as the new IP address was necessary. There were multiple ways of solving this problem, but the easiest way was to modify these devices on the network router and assign a permanent IP address to them.

5. Conclusion

This thesis aimed to create a smart home system that could monitor and control smart home appliances through IoT. As a result of this thesis, an IoT system was built to manage these devices and achive automation using data sent through an IoT-friendly communication protocol. This system consisted of two sub-systems, the home automation platform and the user tracking system.

The sub-system responsible for managing the home through monitoring, controlling and automating smart home devices and services was the home automation platform. This platform was implemented through the use of an open-source smart home platform called openHAB. It allowed for connecting and managing multiple different devices and services from diverse ecosystems and could be run on single-board computers like Raspberry Pi. As such, the Raspberry Pi 4 was used to run the home automation platform.

To demonstrate the IoT capabilities in a smart home environment, the IoT friendly messaging protocol MQTT was used to enable the two sub-systems to communicate. The home automation platform housed a Mosquitto MQTT broker and subscriber client, where the subscriber client logged the user location data that was sent over the IoT network. This data was used by the automation rule to switch Philips Hue lights on and off, based on the users' proximity to them.

To acquire the users' location in the house, the user location system, which was developed and created in this thesis, consisted of individual tracking devices placed in different rooms around the house. These tracker devices were built by using Raspberry Pi Zero W with a user locating Python script. This model of Raspberry Pi single-board computers could use WIFI and Bluetooth and run a lightweight operating system. That operating system hosted the Mosquitto MQTT service and ran the Python script responsible for collecting Bluetooth data and publishing it through the MQTT publishing client.

Both of the sub-systems create the whole smart home IoT system that monitors and controls smart home appliances.

6. References

[1] Research and Markets, “IoT Middleware Market - Growth, Trends, and Forecasts (2020 - 2025),” Research and Markets, August 2020. [Online]. [Accessed 15 March 2021].

[2] Adroit Market Research, “Smart Home Automation Market,” November 2020. [Online]. Available: https://www.adroitmarketresearch.com/industry-reports/smart-home-automation-market. [Accessed 10 December 2020].

[3] “Home automation,” 1 January 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Home_automation&oldid=1004185639. [Accessed 3 March 2021].

[4] Wikipedia, “Home Assistant,” 30 March 2021. [Online]. Available: https://en.wikipedia.org/wiki/Home_Assistant. [Accessed 31 March 2021].

[5] “Home Assistant lets you automate your smart home without giving up privacy,” 10 May 2018. [Online]. Available: https://www.the-ambient.com/features/home-assistant-automation-privacy-582. [Accessed 31 March 2021].

[6] P. Schoutsen, “Perfect Home Automation,” 19 January 2016. [Online]. Available: https://www.home-assistant.io/blog/2016/01/19/perfect-home-automation/. [Accessed 31 March 2021].

[7] Home Assistant, “Documentation,” Home Asisstant, [Online]. Available: https://www.home-assistant.io/docs/. [Accessed 31 March 2021].

[8] openHAB Foundation, “Introduction,” openHAB Foundation, [Online]. Available: https://www.openhab.org/docs/. [Accessed 31 March 2021].

[9] K. Kreuzer, “openHAB 2.0 and Eclipse SmartHome,” 16 June 2014. [Online]. Available: http://www.kaikreuzer.de/2014/06/16/openhab-20-and-eclipse-smarthome/#ug. [Accessed 31 March 2021].

[10] openHAB Foundation, “Who We Are,” openHAB Foundation, [Online]. Available: https://www.openhab.org/about/who-we-are.html. [Accessed 31 March 2021].

[11] openHAB Foundation, “Textual Rules,” openHAB Foundation, [Online]. Available: https://www.openhab.org/docs/configuration/rules-dsl.html. [Accessed 31 March 2021].

[12] Wikipedia, “HomeKit,” 10 January 2021. [Online]. Available: https://en.wikipedia.org/wiki/HomeKit. [Accessed 17 April 2021].

[13] Wikipedia, “Arduino,” 17 March 2021. [Online]. Available: https://en.wikipedia.org/wiki/Arduino. [Accessed 11 April 2021].

[14] Wikipedia, “Philips Hue,” 7 April 2021. [Online]. Available: https://en.wikipedia.org/wiki/Philips_Hue. [Accessed 11 April 2021].

[15] openHAB Foundation, “Configuration Overview,” openHAB Foundation, [Online]. Available: https://www.openhab.org/docs/configuration/. [Accessed 27 April 2021].

[16] openHAB Foundation, “Add-Ons,” openHAB Foundation, [Online]. Available: https://www.openhab.org/addons/. [Accessed 27 April 2021].

[17] openHAB Foundation, “Things,” openHAB Foundation, [Online]. Available: https://www.openhab.org/docs/configuration/things.html. [Accessed 27 April 2021].

[18] openHAB Foundation, “openHABian,” openHAB Foundation, 7 December 2020. [Online]. Available: https://www.openhab.org/docs/installation/openhabian.html#features. [Accessed 10 December 2020].

[19] RASPBERRY PI FOUNDATION, “Raspberry Pi 4 Tech Specs,” RASPBERRY PI FOUNDATION, [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/?resellerType=home. [Accessed 10 December 2020].