

Mikael Wuollet

Smart home automation to aid seniors

SMART HOME AUTOMATION TO AID SENIORS

Wuollet Mikael
Thesis
Autumn 2021
Business Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Business Information Technology

Author: Wuollet Mikael

Title of Bachelor's thesis: Smart home automation to aid seniors.

Supervisor: Minna Kamula

Term and year of completion: Autumn 2021

Number of pages: 50

Number of appendices: 35

The objective of this thesis was to build an automation system for an elderly person. The goal was to have different solutions to problems or to assist an elderly person. To help the caregiver or family member who is responsible for simple daily events. There was no specific client who will be mentioned other than it was for someone's elderly family member.

There are many different systems that are being created in the elderly care field. The problem is that these systems are usually created to tackle one problem or issue. There will be some research addressing these systems, but the goal of this thesis was to create one system that would address many issues or problems.³⁹

The main goal of the thesis was to create one system that would provide a way for a caregiver or family member to have access to the system. This way they would be able to know more about the elderly person without having to interfere too much into their daily lives. Independence is very important to every adult. The system also had to be a hands-off experience because the user as well as the caregiver or family member is likely not to have the knowledge to use the system at a high level.

The copyright associated with this thesis is displayed on the download page and that is still true, but there is a **Disclaimer** accessible on page 44. This Disclaimer says that the author is not liable for the software "code" used by another party.

Keywords: Smart home, Automation, Home Assistant, Senior, Assistance.

CONTENTS

1	INTRODUCTION	6
1.1	Background	6
1.2	Project Objective	6
1.3	Acronyms and Concept Definitions	7
2	SMART HOME AUTOMATION, IOT, AND PROJECT DEVICE LIST	9
2.1	Smart home automation and IoT	9
2.2	Smart home device list	10
3	INSTALLATION OF HOME ASSISTANT	11
3.1	Home Assistant	11
3.2	Raspberry Pi 4 model B	11
3.3	Preparing device and installation	12
3.4	Onboarding Home Assistant and Router Settings	13
4	HOME ASSISTANT ADD-ONS	16
4.1	Add-on File Editor and/or Visual Studio Code	16
4.2	Add-on Samba share for local access	17
4.3	Add-on DuckDNS for remote access	18
4.4	Add-on NGINX Home Assistant SSL proxy	20
4.5	Add-on Dnsmasq	22
5	EXTERNAL ADD-ON DEVICES AND SERVICES	25
5.1	Home Assistant application for client	25
5.2	Telegram application for caregiver or family member	26
5.2.1	Initial Telegram application and bot setup	26
5.2.2	Integration into Home Assistant	27
5.3	Zigbee protocol and device	27
5.3.1	Connecting and Integration of Conbee 2 Device into Home Assistant	28
5.3.2	Connecting a Zigbee device to Home Assistant	29
5.4	Kodi Multimedia system	30
5.4.1	Raspberry Pi 2	30
5.4.2	Preparing device and installation	31
5.4.3	Integration into Home Assistant	32

5.5	Integration of Google Calendar	32
5.5.1	Configuration in Google Developers Console	33
5.5.2	Integration into Home Assistant	34
6	AUTOMATIONS AND INTEGRATIONS	36
6.1	Medication Reminder automation.....	36
6.2	Audio played through Media center.....	39
6.3	Event reminder automations.....	39
6.4	User has failed to take medication (Telegram).....	39
6.5	Has the user opened the refrigerator? (Telegram)	40
6.6	User device monitoring for safety (Telegram).....	40
6.7	Low battery level charge reminder	40
7	DISCUSSION	42
8	CONCLUSION.....	43
9	DISCLAIMER.....	44
	REFERENCES	45
	APPENDICES.....	49

1 INTRODUCTION

1.1 Background

What types of services are provided or offered for elderly care at home? What are the requirements to be qualify for home care? Here the author will research into these questions and then provide an alternative for elderly people that do not need home care all the time.

There are mainly two types of home care available in Finland. There are private for-profit companies and the public sector. The public sector option is a lot cheaper than the for-profit options. Kela is the public health care provider in Finland. According to Wikipedia, Kela is a Finnish government agency that provides social security, health care and many other programs. (32). To qualify for care allowance for pensioners you need to have many difficulties. One difficulty will not be enough reason for you to receive an allowance. According to Kela's eligibility requirements page, you need to have at least three of the difficulties, so that you can be eligible for help. (20). There is basic social assistance that applies to pensioners. According to Kela's basic social assistance for pensioners page there are two different ways to receive benefits, housing allowance and care allowance. (21). There are many obstacles to overcome for a senior to qualify for social assistance and this project would help with addressing some of these issues.

1.2 Project Objective

The project objective is to create an automation solution for seniors that are living at home. Many seniors want to continue living at their own homes for as long as possible. This can create a burden for family members because now they need to assist this senior. There are many types of care that are offered. These types of services that are offered are expensive, and many times leave the senior needing more than what is offered. Seniors need to have social interactions as much if not more than other people. Retired seniors might not have any activities, or do not have the ability to take part in activities.

This project would assist in some of the activities that family members need to do while assisting a senior. This takes a bit of the burden of assisting the senior and focus on using their energy to

socially interact with the senior. This project would help with reminding the senior to take their medication and notify a caregiver or family member if medication is not being taken. A system for family members to be able to add new and modify when medication needs to be taken, and if there are important events to remember. It would also be able to notify a family member if the senior is away from home for more than a certain time frame. The fact that this system relies a lot on the senior's mobile phone, this project will also remind the senior to charge their mobile phone when it is under twenty percent battery by voice using a speaker.

1.3 Acronyms and Concept Definitions

API	Application Programming Interface, software that allows two applications to communicate with each other.
Bot	A software application that can carry out tasks that are programmed. For Example, repetitive tasks.
DHCP	Dynamic Host Configuration Protocol is a network management protocol used by Internet protocol networks. Internet service providers use this to assign IP addresses.
DNS	Domain Name System is a naming system that names devices in a way a human can remember.
Docker	Operating System level virtualization software.
HA OS	Home Assistant Operating System.
IoT	Internet of Things Physical devices that connect to the internet and transmit from sensors.
IP address	Internet Protocol address. A numerical label assigned to each device connected to that network.
ISP	Internet Service Provider, A company that provides an internet service.
JeOS	Just enough Operating System
LAN	Local area network. A group of devices connected within a small area.
Nat-loopback	A Nat-loopback is when you use software to redirect a WAN address to an LAN address, so you do not need to access the internal device with the internal address.
OS	Operating System.
SSL	Secure Shell protocol for secure connections over an unsecured network.
TTS	Text-to-speech is used to artificially produce human speech from text.

UI	User Interface is where a user can interact with systems.
USB	Universal Serial Bus cable, it is used to connect peripheral devices with other devices for example a keyboard to a computer.
WAN	Wide area network. A large group of devices connected within a large area that are all connected for example the internet.
YAML	YAML Ain't Markup Language is a human-readable data-serialization standard.
ZHA	Zigbee Home Automation integration for Home Assistant that connects an external Zigbee controller with Home Assistant.
Zigbee	Wireless protocol based on the 2.4Ghz radio spectrum.

2 SMART HOME AUTOMATION, IOT, AND PROJECT DEVICE LIST

2.1 Smart home automation and IoT

Home automation is when the home can gather information and use the devices within the home to make people living there easier. If there is more interaction with a smart home system than there is without it, that is not home automation. According to Techopedia an automation system in a home is also called a smart home. The automation system allows most of the electronic and technology-based functions to be automated within a home. (29.)

Smart home to assist seniors is a great way to help without direct involvement. According to Miller M. who is the author of My Smart Home for Seniors, most people want to stay in their own home for as long as possible. If they stay longer at home, they keep their independence longer. This in turn, keeps the amount of assistance that they need lower which benefits everyone. (24).

The Internet of Things (IoT) is the name of a group of devices that can connect to a network and transmit data. According to Kenton W. these devices are usually smart when talking about home IoT devices. These IoT devices can gather information and deliver the data to a central system. (22.)

2.2 Smart home device list

The devices that were selected for this project are in the list below. The way that the devices and services will be used are shown in the Figure 1. The reason why these devices were chosen was that the prices were low, and the availability of these products were easy to find.

- Raspberry Pi 4 model B, Home Assistant OS
- Raspberry Pi 2 model B, LibreElec media center (Kodi Media Player)
- Zigbee Aqara Zigbee door sensors
- ConBee II, Zigbee USB gateway
- Android phone, user of system
- Android phone, caregiver/family member
- Generic speaker
- Router

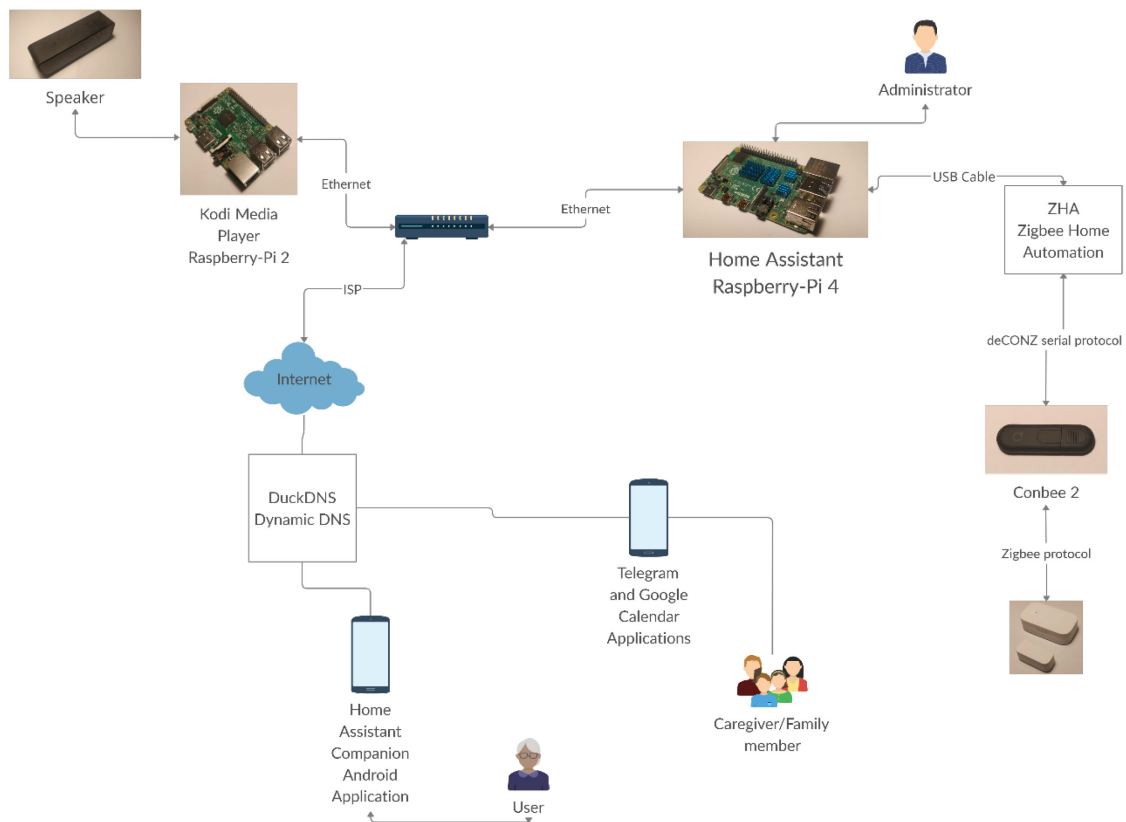


Figure 1 Diagram of the whole system.

3 INSTALLATION OF HOME ASSISTANT

3.1 Home Assistant

Home Assistant is an open-source home automation platform that is sponsored by Nabu Casa and is maintained by the open-source community. The focus of this platform is security and keeping your data local and in control of your own data. There are many platforms offered by various companies, but many of them require you to use the cloud. This means they have access to your data. Home Assistant can run on many devices. (9.)

There are four community supported install methods. First is the Home Assistant Operating System (OS) that is a community supported platform and is the installation method that is used for this project. Home Assistant Container that is based on a docker container. Home Assistant Core that can be installed on a Linux platform. Home Assistant Supervised that also can be installed on a Linux platform. The device that is used in this project will be the Raspberry Pi 4 and is one of the supported devices with the Home Assistant OS. (9.)

3.2 Raspberry Pi 4 model B

The Raspberry Pi 4 model B pictured in Figure 2 is made by the Raspberry Pi Foundation located in the United Kingdom. (28). The Raspberry Pi 4 was the device that was selected for its ease of use, and cheap price. It is the most popular single-board computer. It is known for its usage with education institutions in teaching students simple coding concepts. (28). It is also used by people to create hobby projects as well as deployments in the commercial space.

The single-board computer has impressive specifications. According to the Raspberry Pi Foundation it has a Broadcom BCM2711, quad core Cortex-A72 (ARM v8) 64-bit Soc at 1.5GHz. 2GB, 4GB, 8GB SDRAM depending on the variant. It also comes with built in 2.4 and 5.0GHz wireless radio and Bluetooth 5.0. Gigabit Ethernet. Two USB 3.0 and two USB 2.0 ports. Two Display ports. (27.)

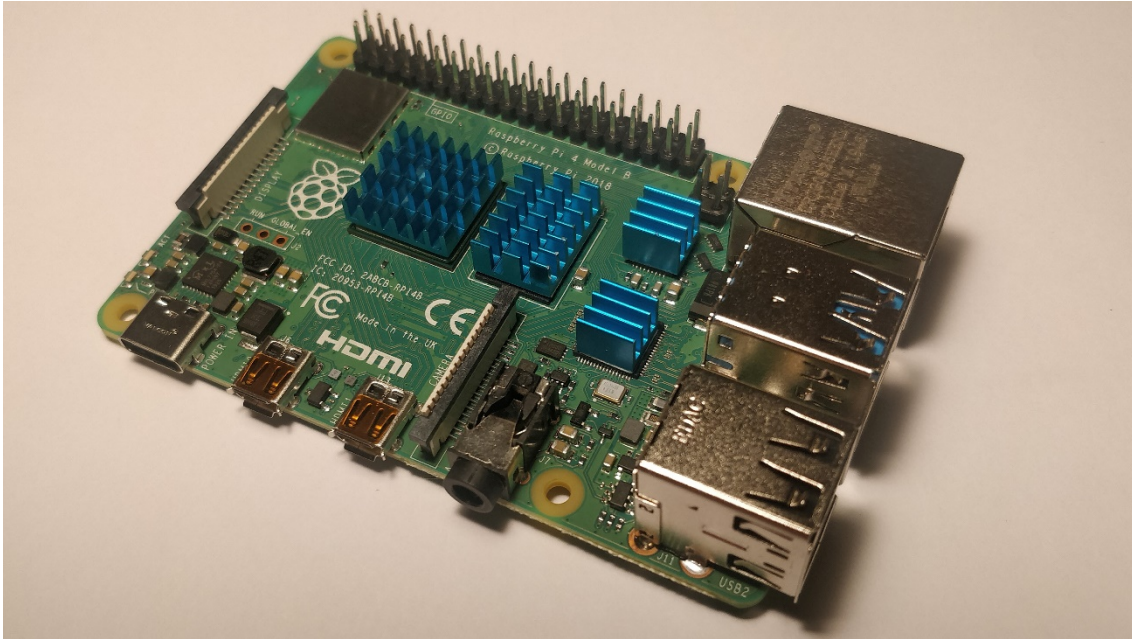


Figure 2 Raspberry Pi 4 model B created by the Raspberry Pi Foundation.

3.3 Preparing device and installation

Home assistant OS boot image was downloaded from the Home Assistant's download page. For this operating system to be used on the Raspberry Pi, it needs to be flashed on to a micro-SD card. The program Etcher shown in Figure 3 was used. This is available for download at <https://www.balena.io/etcher/>. A USB micro-SD card reader is needed to connect the micro-SD to the computer. Depending on the speed of the micro-SD card the flashing could take up to 10 minutes to finish. Select the downloaded image, the correct drive and then flash.

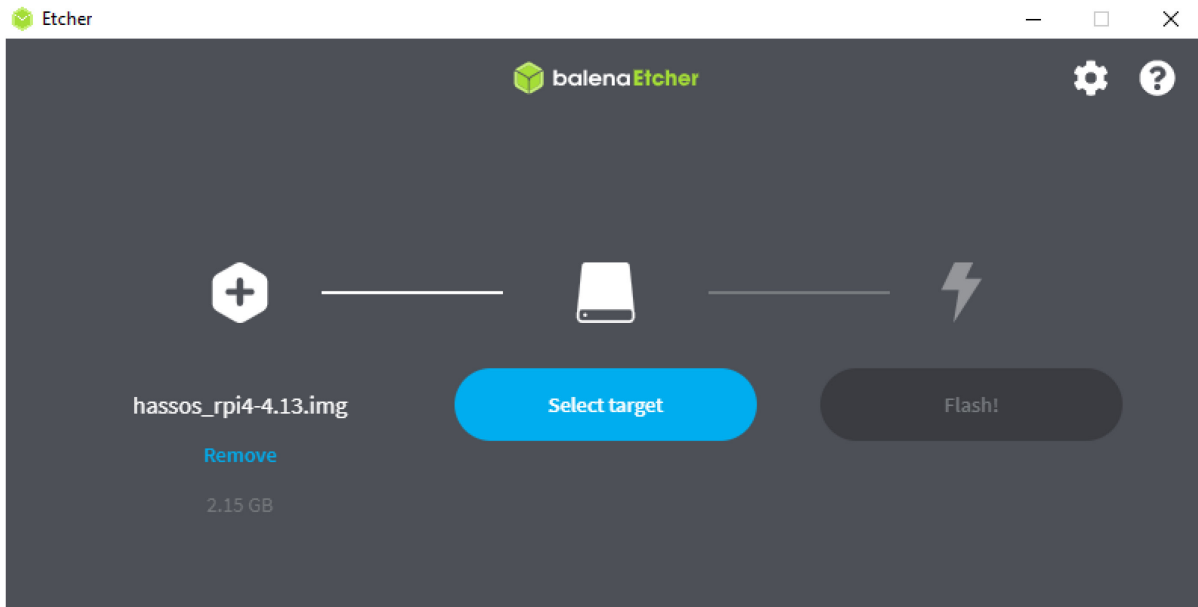


Figure 3 Etcher, Flash operating systems to an external device (example Micro SD-card.)

After flashing the micro-SD card, connect an Ethernet cable from a router to the Raspberry Pi and a good power supply to the USB Type-C port. The Raspberry Pi will boot up and start the installation. According to Home Assistant this could take up to 20 minutes to finish because it will need to pull any new updates and install them. (12).

3.4 Onboarding Home Assistant and Router Settings

According to Home Assistant when this is finished the web interface can be accessed through your browser at <http://homeassistant.local:8123>. (12). If this does not work typing in the IP address of your device for example <http://192.168.1.100:8123>. When visiting the web interface for the first time a name, username and password is required. After this the default dashboard will be displayed like in Figure 4.

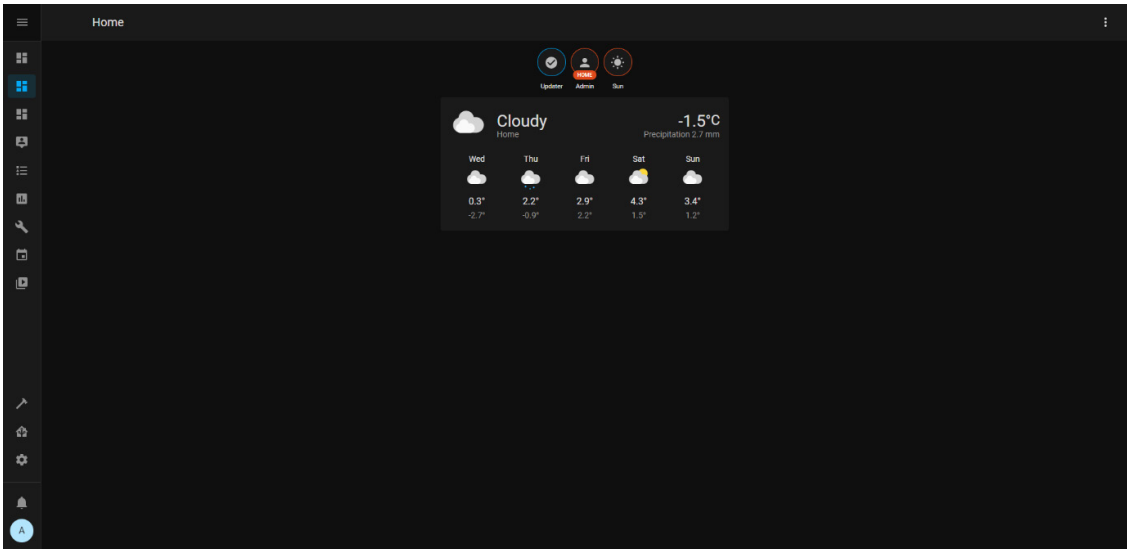


Figure 4 Home Assistant default dashboard.

To be able to check your configuration against a version of a healthy Home Assistant file system you need to enable advanced mode on your user profile. This can be done by clicking on your profile and changing the switch to “on.” Home Assistant now needs to be restarted to enable it. This can be done by going to “Configuration” tab just below the “Supervisor” tab. Selecting Server controls near the bottom. Click restart.

One of the most important issues to address are back-ups. The easiest way to ensure you have up-to-date backups are by using the snapshot feature by going to Supervisor and into Snapshots. Here a name can be given to the back-up and add a password if wanted for extra security. These back-ups can then be downloaded and put in a safe place. For example, in a cloud back-up service.

Before going any further, this would be a great time to set a static IP for the Home Assistant Raspberry Pi device. This ensures that the Home Assistant instance is always located at the same IP address. That is important because when connecting remotely the IP address needs to be the same address for it to connect. This can be done in different ways, but the easiest way is to assign the static IP using your router. When certain add-ons are added, the Home Assistant device will need to have a static IP.

Before moving over to the router settings, we will need an IP address off the Home Assistant Raspberry Pi device. We can access this by accessing the supervisor tab and clicking system. Here the basic system information is shown. The IP address is in the host section shown in Figure

5. The one that is needed is the one that starts with something like 192.168.1.1, 10.0.0.1, 172.0.0.1. The IP address shown for this device is 192.168.2.150.

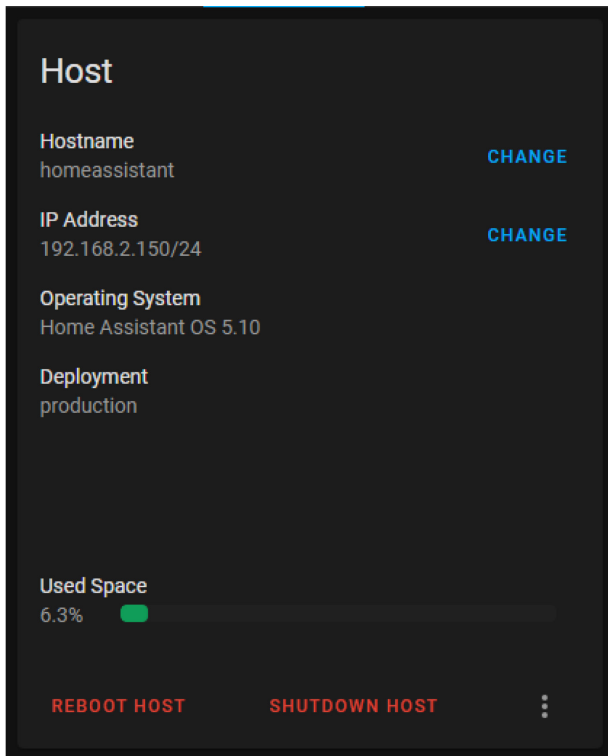


Figure 5 System host information, IP address of Home Assistant.

Now that we have the IP address, we can move on to the router settings. To access your router, you will need to check your router brand and model. You can check what the default IP, username, and password by checking either your router manual or by searching the internet with your router brand and model. The router being used in this situation is a TP-Link Wireless Router Archer C8 V1. Settings of different brands and models may be different.

Before assigning the static IP, we need to find the MAC address of the Home Assistant Raspberry Pi device. Usually, the home page of the router settings has connected clients. When accessing these clients find the device with the Home Assistant device's IP. Next to this IP should have a MAC address. Copy this MAC address to apply to the static IP settings, that will be done next.

Usually, the static IP can be assigned by accessing the advanced section of the settings and locating the DHCP settings. In the DHCP settings address reservation needs to be found. Here you will enter the MAC address of the Raspberry Pi device. Reserve the IP address that you found in the Home Assistant supervisor tab. Restart the router and those settings will be in effect.

4 HOME ASSISTANT ADD-ONS

The following add-ons are additional applications built by various people in the open-source community. Almost all the add-ons are completely open source. Some of them are official add-ons that you can add to your configuration by just installing them. The following add-ons are needed for access to the file system, access home assistant locally, and remote access and internal redirects. All these add-ons can be installed by going to the supervisor and accessing the Add-on Store from the landing page after logging in as shown in Figure 6.

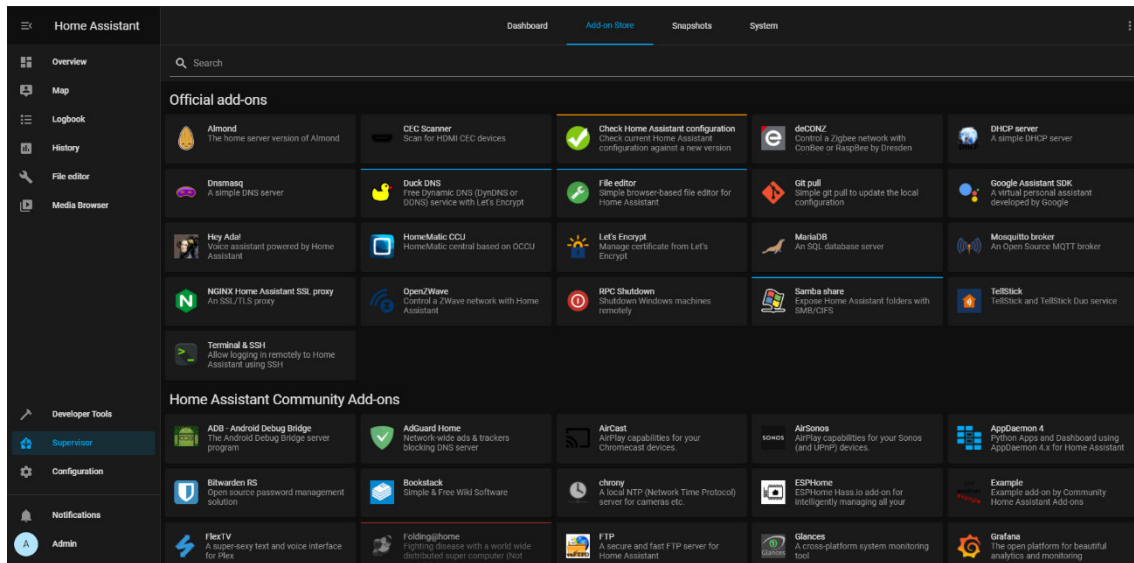


Figure 6 Add-on Store, where additional applications can be added to home assistant.

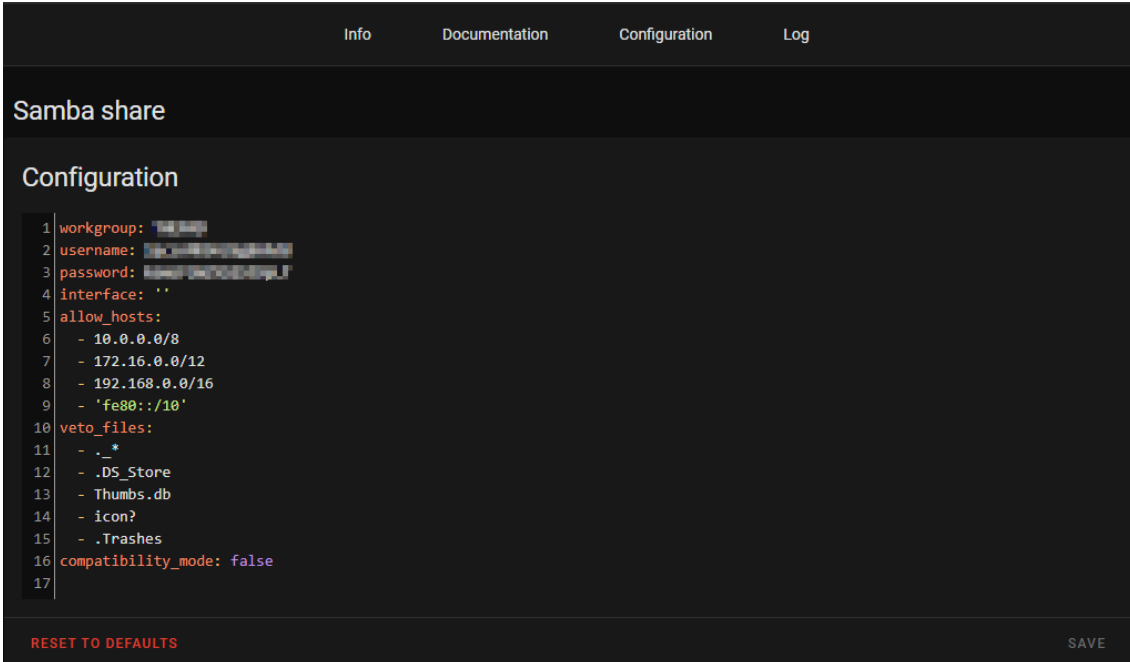
4.1 Add-on File Editor and/or Visual Studio Code

The file editor is a simple in browser file editor that can access all the files and folders of the system. The files can also be edited. This add-on does not need any configuration. Just an install and start of the add-on.

Another way to access the files is using the Visual Studio Code add-on. The add-on can be installed, and then started. There are no changes to the configuration that are needed. The Visual Studio Code add-on requires 64-bit version of Home Assistant.

4.2 Add-on Samba share for local access

The Samba share will be used to access the Home Assistant file system from a Windows computer. The reason for this is to simplify the process of accessing, adding, and editing files on the file system. According to the Samba share documentation, the Samba share add-on is installed, and the configuration needs to be edited to what is shown in Figure 7. In the configuration tab the username and password must be changed, and it needs to be recorded because it will be needed later. The workgroup can be changed if there is an assigned work group on your network already, otherwise it can remain as it is. Save the configuration and then start the add-on. Now Home Assistant needs to be restarted. (4.)



```
1 workgroup: 'WORKGROUP'
2 username: 'admin'
3 password: 'password'
4 interface: '*'
5 allow_hosts:
6 - 10.0.0.0/8
7 - 172.16.0.0/12
8 - 192.168.0.0/16
9 - 'fe80::/10'
10 veto_files:
11 - *
12 - .DS_Store
13 - Thumbs.db
14 - icon?
15 - .Trashes
16 compatibility_mode: false
17
```

Figure 7 Samba share add-on configuration, username, password, and workgroup are blurred out.

Going to a windows computer on your network you will be able to access the file system by mapping a network drive. Open file explorer and right click “This PC” and select “Add a Network location” or “Map Network Drive.” Select a drive letter (it does not matter what it is.) Write in the Home Assistant IP Address, in this project it was like this, “\\192.168.2.150\config” and select finish. You will need to now enter your username and password you created in the configuration file. (4.)

Now that access is possible from a browser, you will be able to access and edit the files locally with the preferred file editor. In this project Visual Studio Code on PC by Microsoft was used.

4.3 Add-on DuckDNS for remote access

Having remote access is important for monitoring as well as making changes to the system without having to be on-site. Not only is remote access needed for the administrator but also any other devices accessing home assistant outside of the local network.

According to Nabu Casa they provide the Home Assistant Cloud service. This service is a subscription-based service that costs five euros a month. (25.)

To avoid this cost the DuckDNS add-on will be used. DuckDNS is a free dynamic DNS service and SSL service. This is possible even if Home Assistant is hosted on a dynamic IP address provided by the ISP. First Install DuckDNS. Do not start the add-on. According to the DuckDNS documentation the configuration needs to be edited like in Figure 8 before doing so. (2).

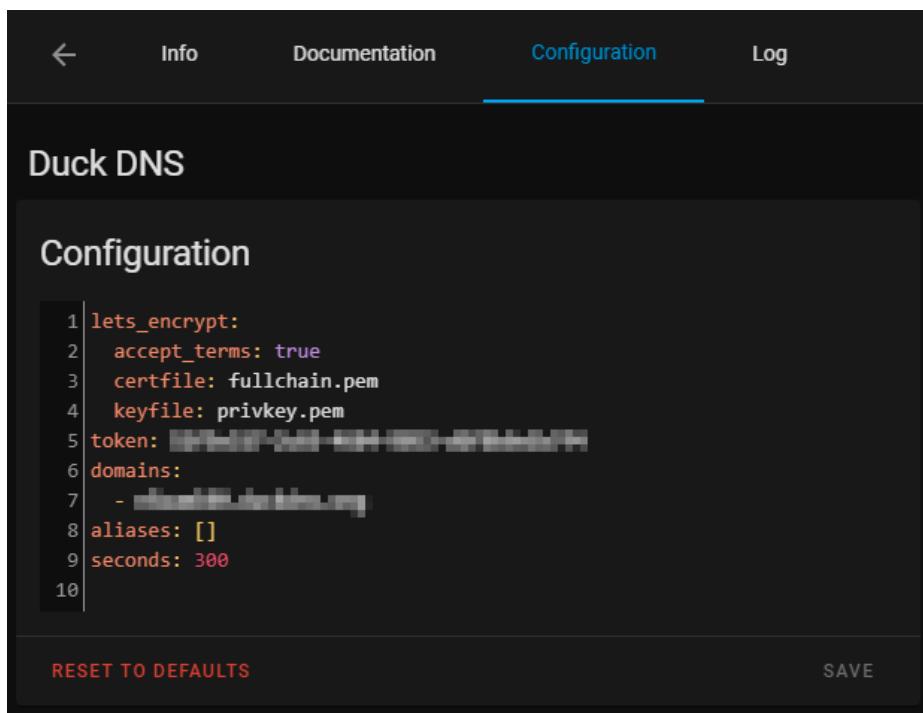


Figure 8 Duck DNS add-on configuration. Token and Duck DNS domain blurred.

The token and Duck DNS domain are the two pieces we need to complete the configuration. To get these, we need to visit <https://www.duckdns.org>. Create a new free account. Once created you can create a unique new sub domain. One example would be <http://home.duckdns.org>. Once this is created you will now have access to the token that is needed. An example token would be 2w483jy4-4k94-3758-0695-g3k8s9n3k6k7. According to the DuckDNS documentation copy this

token and paste it into the configuration file. The domain you created will need to be copied into the domains section. The one more step is to change “accept_terms”. From false to true. Click save. Then back to the info page you can proceed to start the add-on. (2.)

Now we need to open the router ports. Every router is different, so the settings will vary from router to router. In this project the router settings looked like what is in Figure 9. Under the advanced settings the setting we are looking for is port forwarding or virtual servers. If you can name the service name, you can name it home assistant for example. The service port will be 8123. The internal port will be 8123. The IP address will be the IP address of your Home Assistant device. For this setup, it is 192.168.2.150. The protocol is TCP.

Add or Modify a Virtual Server Entry

Service Port: (XX-XX or XX)

Internal Port: (XX, Only valid for single Service Port or leave it blank)

IP Address:

Protocol:

Status:

Common Service Port:

Figure 9 Virtual server/Port forwarding settings to port 443.

The DuckDNS service also provides a secure sockets layer. For this add-on to work we need to configure Home Assistant to use SSL and DuckDNS. Open the editor of your choice either the file editor, VS-Code add-on or through the samba-share add-on. Using the editor, we want to open the configuration.yaml file. According to the DuckDNS documentation we need to add the following code shown in Figure 10. In Figure 11 you can see the whole picture of what is going on. (2).

```
#####  
# DuckDNS + Letsencrypt settings  
#####  
http:  
  - ssl_certificate: /ssl/fullchain.pem  
  - ssl_key: /ssl/privkey.pem
```

Figure 10 Add this DuckDNS configuration code to your configuration.yaml file.

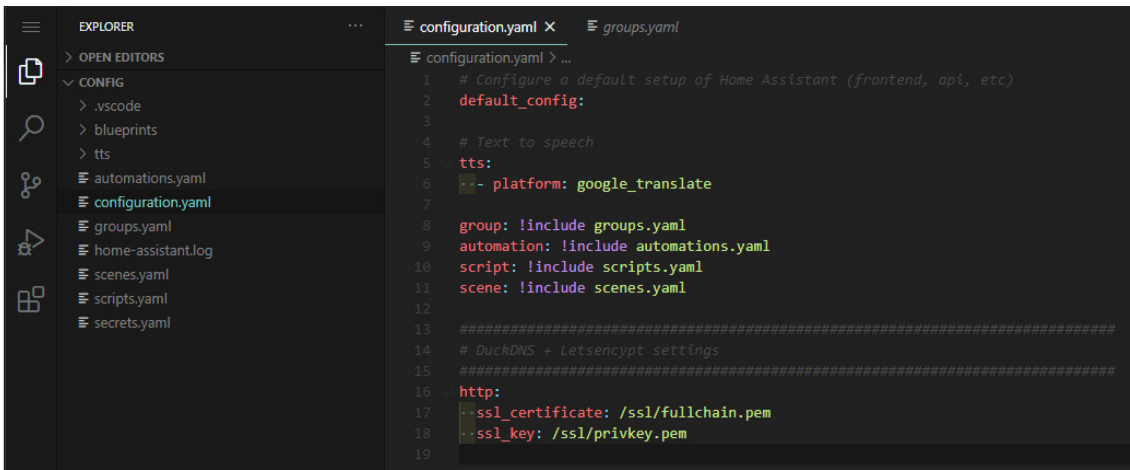


Figure 11 Here is the overview of where the configuration.yaml file is located.

After this file is saved. Home Assistant needs to be restarted. This can be done by going to “Configuration” tab just below the “Supervisor” tab. Selecting Server controls near the bottom. Click “Check Config.” If this is valid that you may “Restart”. If it is not valid you will need to go through your steps to make sure you followed them correctly. Now the Home Assistant instance will be accessible through secure connection through your DuckDNS domain name. In a browser type “https://home.duckdns.org:8123.”

4.4 Add-on NGINX Home Assistant SSL proxy

There are still a few problems with the configuration of Home Assistant. The multimedia system that will be implemented later needs to have a non-SSL connection to Home Assistant. Unless your router supports Network-loopback it will not be possible without this add-on. The NGINX Home Assistant SSL proxy add-on provides a way around this issue.

According to the NGINX documentation you need to start by installing the add-on. Then changing the configuration to the code that is shown in Figure 12. (3).

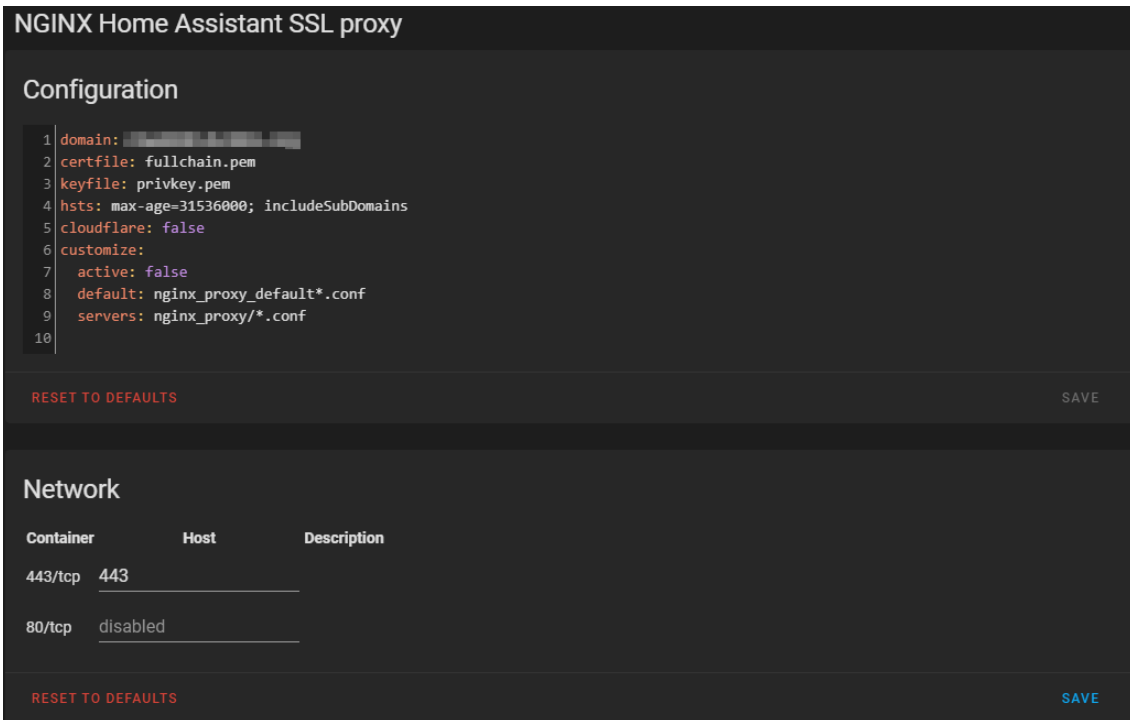


Figure 12 NGINX Home Assistant SSL proxy add-on configuration.

Again, this needs to be saved. This add-on is now ready to be started from the Info tab. There are still a few steps that need to be completed. The ports on the router need to be changed. During DuckDNS add-on configuration the port 8123 was forwarded. This needs to be changed from 8123 to 443 shown in Figure 13.

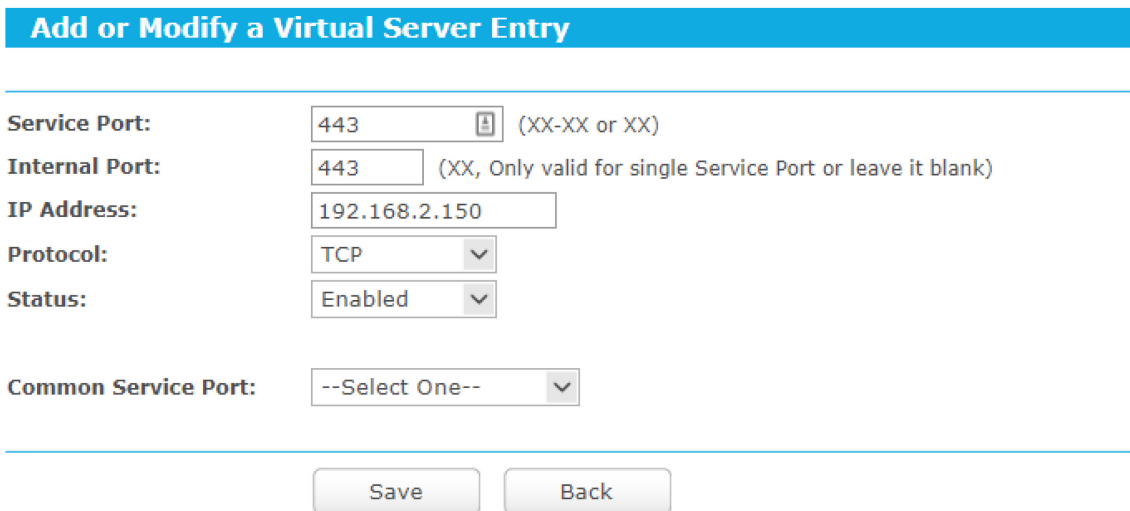


Figure 13 Virtual server/Port forwarding settings to port 443.

Now this is changed and saved. Back in the Home Assistant configuration.yaml file in the file editor. The http: section that was added, requires Home Assistant to communicate only through the https protocol. By commenting it out this will get rid of that requirement within the network. (Figure 14.)

According to the NGINX documentation the NGINX add-on will handle the https protocol communication with DuckDNS. (3.)

```
#####  
# DuckDNS + Letsencrypt settings  
#####  
# http:  
#...ssl_certificate: /ssl/fullchain.pem  
#...ssl_key: /ssl/privkey.pem
```

Figure 14 Configuration.yaml file with http: commented out.

Now the Home Assistant needs to be restarted. Navigate to Configuration. Check config. If valid then proceed to restart. Home Assistant now should be accessed using its own IP address. This system's Home Assistant is accessible at 192.168.2.150:8123 within local network.

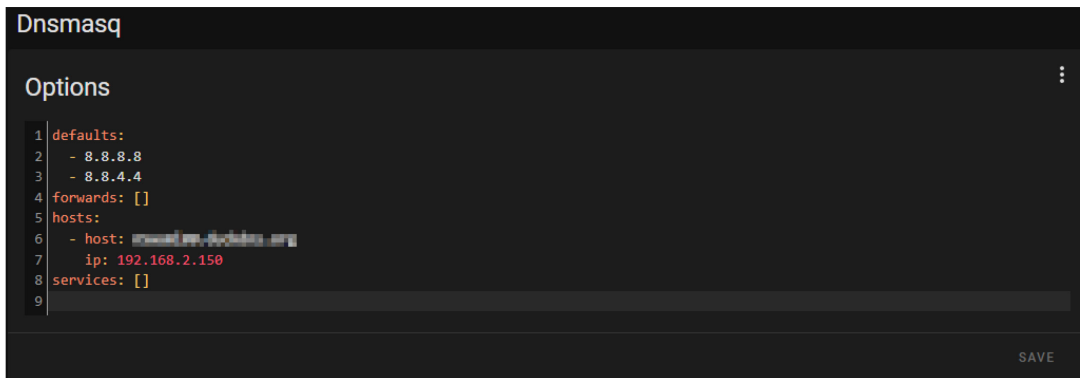
For Home Assistant to be accessible from outside of the local network the code shown in Appendix 7. This code will ban IP address that attempt to access the Home Assistant system with an incorrect password more than five times. According to the Home Assistant NGINX documentation, this will also enable local login without requiring authenticating. The "trusted_networks" in "auth_providers" was 192.168.2.0/24 in this project. It can be different depending on the local network subnet. These changes can be seen in Appendix 7. After these changes Home Assistant needs to be restarted once more. (3.)

4.5 Add-on Dnsmasq

This add-on will make accessing Home Assistant a more intuitive. There will be only one URL that will be used. To complete the solution of the hairpin-nat or nat-loopback router issue. The Dnsmasq add-on solves this issue by using a split DNS. According to the Dnsmasq documentation if a request comes within the network for the DuckDNS URL it redirects that request straight to the Home Assistant device. (8).

To add Dnsmasq to Home Assistant it needs to be installed. When it is finished installing the configuration needs to be modified. According to the Home Assistant Dnsmasq documentation, to fix the issue with the router the DuckDNS URL needs to be added as a host assigned to the Home Assistant Device's IP. It will then redirect that traffic directly to that device within the network. The

Dnsmasq add-on documentation was used to configure the settings. In the configuration tab the following code in Figure 15 needs to be added. (8.)



```
1 defaults:
2   - 8.8.8.8
3   - 8.8.4.4
4 forwards: []
5 hosts:
6   - host: [redacted]
7     ip: 192.168.2.150
8 services: []
9
```

Figure 15 Dnsmasq configuration.

The host would be your DuckDNS subdomain for example, home.duckdns.org. The IP would be your Home Assistant local IP. In this project it is 192.168.2.150. Clicking save will keep this configuration settings. The defaults: area is for your DNS settings. The ones provided are Google's public DNS servers and these do not need to be changed.

To complete the set-up of the add-on the DNS servers of your router need to be changed to be pointed to your Dnsmasq add-on. Dnsmasq will be able to check all incoming requests against the hosts that are listed in the configuration. This will keep your DuckDNS subdomain URL request local and will not send that out to the internet when accessing Home Assistant from within your local network (LAN). All other requests will then be sent to the DNS server you specified in the defaults section from before.

The router settings will vary from brand and model. The keyword is DNS settings. In the case of this specific router DHCP settings where DNS settings are located. The Primary DNS needs to be your Home Assistant IP address where your Dnsmasq add-on will wait for any requests. For this project it is 192.168.2.150. Now with this changed it can be saved and the router can be restarted. Now when the created DuckDNS subdomain is accessed, it will remain local and will not leave the local network.

To complete the entire setup of the three add-ons we need to change the external URL. This can be done by logging into Home Assistant. Navigating to "Configuration," scrolling all the way down

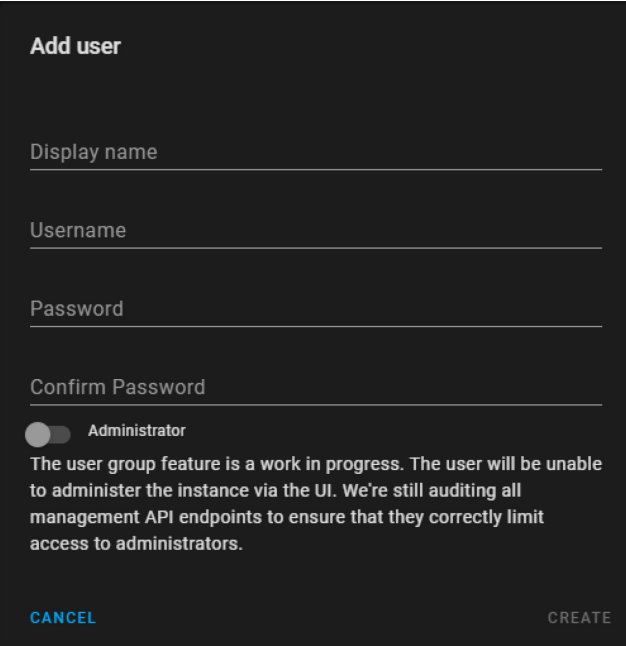
to “General,” and changing the “External URL” to the DuckDNS subdomain that was created. For example, <https://home.duckdns.org>.

These three add-ons NGINX, DuckDNS, and Dnsmasq work together and create a secure connection when Home Assistant is accessed from in and outside of the LAN. The changes that were made to the configuration.yaml file can be seen in Appendix 7.

5 EXTERNAL ADD-ON DEVICES AND SERVICES

5.1 Home Assistant application for client

Using the official mobile phone application named Home Assistant. This will enable the user or client to receive notifications from Home Assistant. The user will use their own credentials to sign into the Home Assistant platform. Signing into your *home.duckDNS.com* and providing the username and password. Create a new user by accessing Configuration on the Home Assistant Web interface. Select users and then click add user. A window will be displayed like what is shown in Figure 16. Create a display name, username, and a password. this user should not be an administrator because the user will not need those permissions. The new user created for the project was “user_client”



Add user

Display name

Username

Password

Confirm Password

Administrator

The user group feature is a work in progress. The user will be unable to administer the instance via the UI. We're still auditing all management API endpoints to ensure that they correctly limit access to administrators.

CANCEL CREATE

Figure 16 Adding a new user to Home Assistant.

The Home Assistant platform has both Android and iOS applications available. In this project the Android application will be used by the user client and, caregiver or family member.

The Android application “Home Assistant” can be downloaded from the Google Play store. Linked here: <https://play.google.com/store/apps/details?id=io.homeassistant.companion.android>. The application was opened and the DuckDNS URL that was created from earlier can be filled in. For example, <https://home.duckdns.org>. Log in using the newly created credentials. Enable location

tracking because this will come in use later when settings up automations. Clicking continue will finish the configuring of the connection.

Two small pieces of code needs to be added to the “configuration.yaml” file. The code “mobile_app:” and “discovery:” needs to be added in separate lines. This will enable the Home Assistant system to recognize the Android device.

5.2 Telegram application for caregiver or family member

There needs to be a way for a caregiver or family member to be able to get information about the user. There are many options to choose from. According to Telegram the Telegram application is free and is also a multi-platform application. (31). This was chosen also because it is very flexible, and the limits are endless.

5.2.1 Initial Telegram application and bot setup

The Telegram application can be installed on many platforms. It can be downloaded for Android from the iOS app store, Google play store, or from the link <https://play.google.com/store/apps/details?id=org.telegram.messenger&hl=en>. According to Telegram there are applications made for the most popular platforms including Windows, Linux, MacOS, IOS, and Android. (31). Android phones are very popular so in this project an Android phone will be used. The application is installed from the Google play store and the onboarding process is completed.

Now the bot that will interact with Home Assistant can be created. In the next section 5.2.2 the configuration will be added to Home Assistant. According to the Home Assistant Telegram Bot documentation the following steps need to be completed. The bot is created by searching for BotFather. The BotFather is a verified bot that has been created by Telegram. It is important to use the verified bot that has a blue check mark. After clicking on the correct BotFather, create a new bot with the command “/newbot” and give the bot a name. Record the HTTP API token. (30.)

Now the chat ID and needs to be found. According to the Home Assistant Telegram Bot documentation, start with searching for the newly created bot with the search bar. Start a

conservation and write in “test”. This will start communication with the bot. The chat ID can be found by visiting the website provided and filling the “YOUR_API_TOKEN” with the HTTP API token “https://api.telegram.org/botYOUR_API_TOKEN/getUpdates” and this was found from the BotFather. The response should look like what is in Figure 17. If there is only the first part {"ok":true,"result":[]}. Then another message should be sent. The chat ID will be the number after “id” like this, “id”:2840382059. Record this ID for use later. (30.)

```
{
  "ok": true,
  "result": [
    {
      "update_id": 123456789,
      "message": {
        "message_id": 1,
        "from": {
          "id": 2840382059,
          "is_bot": false,
          "first_name": "Test",
          "last_name": "User",
          "language_code": "en",
          "chat": {
            "id": 2840382059,
            "first_name": "Test",
            "last_name": "User",
            "type": "private",
            "date": 1511464217,
            "text": "test"
          }
        }
      }
    }
  ]
}
```

Figure 17 Chat ID that is needed for integration with Home Assistant.

5.2.2 Integration into Home Assistant

The Telegram integration will need to be setup by accessing the configuration.yaml file. Using your preferred editor access the file and add what is in Figure 18 to the configuration file. The notify configuration should be added at this point as well. Notify will enable you to receive notifications from Home Assistant through the Telegram bot to the family member or caregiver’s Telegram application. According to the Telegram bot and Telegram polling documentation the following configuration is based upon the example configuration provided by these documentations. (13). (14). The following configuration is this project unique configuration and is shown in the second part of Figure 18. More of these “my_chat_id” can be added for example all caregivers or family member’s chat ids.

```
#####
#•Telegram•configuration
#####
telegram_bot:
  - platform: polling
  - api_key: !secret telegram_api_token
  - allowed_chat_ids:
    - !secret my_chat_id

#####
#•Notify•configuration
#####
notify:
  - #•Telegram•configuration
  - name: smarthome_bot
  - platform: telegram
  - chat_id: !secret my_chat_id
```

Figure 18 Telegram configuration in configuration.yaml file.

5.3 Zigbee protocol and device

According to the Zigbee Alliance, they have developed a cheap, low power and bidirectional wireless communication standard. (33). It is a standard that is used for low powered IoT networks.

The device that will be used in this project will be the Conbee 2 which will be the Zigbee coordinator. The coordinator can communicate with other Zigbee routers and Zigbee end devices.

5.3.1 Connecting and Integration of Conbee 2 Device into Home Assistant

To connect and use the ConBee 2 Zigbee coordinator with Home Assistant it needs to use an integration or an add-on. There is an add-on that Dresden Elektronik maintains for using the ConBee 2 coordinator with Home Assistant. This add-on works very well, but it adds another layer to Home Assistant. That is why the method that this project will use the ConBee 2 coordinator will be the Zigbee Home Automation integration.

According to the Zigbee Home Automation (ZHA) documentation, ZHA is an integration that enables you to connect Zigbee devices to Home Assistant by using a Zigbee coordinator. (17). The coordinators that work quite seamlessly are listed on the Home Assistant ZHA integration website.

The ConBee 2 installation is very straight forward. The ConBee 2 USB stick should be plugged into an USB extension cable to keep the device away from the host device. The reason for this is to cut down on interference that can affect the performance of the ConBee 2 Zigbee frequencies. (19). After connecting the coordinator Home Assistant can be restarted. According to the Home Assistant ZHA documentation when Home Assistant has restarted navigate to “Configuration” and “Integrations.” A new integration can be added by clicking the blue “Add Integration” button. The Zigbee Home Automation integration can be searched with the term ZHA. Clicking will start the configuration. Select and submit the serial port. The minimum serial port is `/dev/ttyACM0` with the ConBee 2 coordinator. The best serial port to select is one that starts with `/dev/serial/by-id/`. (Figure 19.) This makes a direct connection to the coordinator. If this selection is unavailable, it is possible to configure this by selecting manual and filling out the information. Finally, select an area that the coordinator will be physically in the home. The documentation specifies everything that is required for the configuration. (17.)

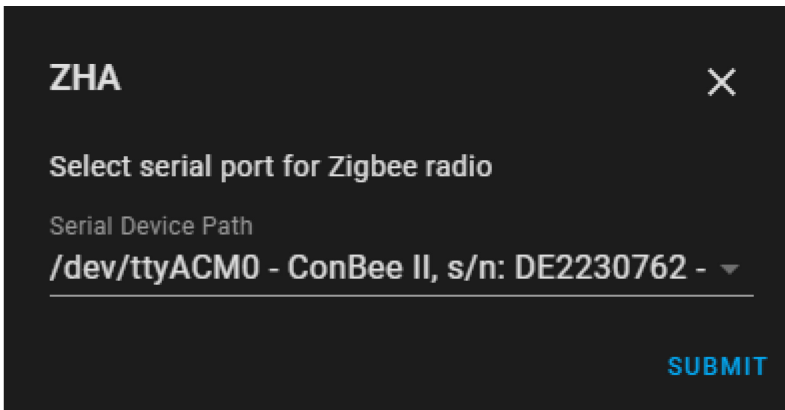


Figure 19 Zigbee radio serial port.

5.3.2 Connecting a Zigbee device to Home Assistant

There are hundreds of Zigbee devices that can be connected to Home Assistant using the ZHA integration with the ConBee 2 coordinator. A new Zigbee device can be added by again going to integrations and finding the ZHA integration and click configure. Here the “Add Device” button. This will put the coordinator into a searching state and will attempt to connect to any Zigbee device that is in pairing mode.

The first device that was connected to Home Assistant through the Conbee 2 coordinator through ZHA was the Aqara door and window sensor shown in Figure 20.



Figure 20 Aqara door and window sensor.

The Aqara door and window sensor has a small button that needs to be held down for three seconds to put it into pairing mode. If those the sensor and ZHA are in pairing mode, ZHA should automatically find the Aqara sensor and the area where the device will be placed can be selected. If the sensor is not found repeat steps until successful. These steps were taken with all devices that were connected to Home Assistant.

5.4 Kodi Multimedia system

There are a handful of different open-source multimedia systems that are available on the web. The one that stood out was the Kodi. Kodi is well integrated into Home Assistant with an integration already created. The integration also can be done in the UI keeping things simple without having to dive into YAML files. The Kodi integration has many services that can be called to control the Kodi multimedia system. (10.)

5.4.1 Raspberry Pi 2

The Raspberry Pi 2 model B pictured in Figure 21 is made by the Raspberry Pi Foundation located in the United Kingdom. The Raspberry Pi 2 was the device that was selected for its ease of use, and cheap price. This series of single-board computers are the most popular. The single-board computer has moderate specifications. According to the Raspberry Pi Foundation a Broadcom BCM2836, quad core Cortex-A7, 32-bit Soc at 900 MHz, 1GB SDRAM, 100 base Ethernet, four USB 2.0 ports and one HDMI display port according to the Raspberry Pi Foundation. (26.)



Figure 21 Raspberry Pi 2 model B created by the Raspberry Pi Foundation.

5.4.2 Preparing device and installation

Raspberry Pi 4 was used for the Home Assistant device because it requires more power. The Raspberry Pi 2 will be used for the multimedia system. This multimedia system does not need the power that the Raspberry Pi 4 has. The operating system that we will use is LibreELEC. According to LibreElec this operating system is a (JeOS) Just enough Operating System. This means the operating system contains the bare minimum that is required for usage. This device will be dedicated to being a multimedia system and nothing else, so the minimum is how to get the most out of the devices resources. The operating system is only running the Kodi software. It is quick to boot and does not have any unnecessary parts included. (23).

Like the Home Assistant OS, the LibreELEC OS image needs to be installed onto a micro-SD card that will be mounted in the Raspberry Pi 2 micro-SD card slot. The LibreELEC OS image can be downloaded from the LibreELEC website.

The image can be flashed on to the micro-SD card with the same program as the Home Assistant image, Etcher. Looks like what was shown in Figure 3, but instead with the LibreELEC OS image. The micro-SD card needs to be inserted into the computer. Flash the image wait for it to complete and then insert the micro-SD card can be inserted into the Raspberry Pi 2. A mouse, a keyboard, a monitor, a speaker and an ethernet cable should be plugged in. Connect power and let it boot up.

The initial boot will take a few minutes to configure the device. It will reboot once and then the default desktop will appear. There are a few steps in the setup that need to be completed. Selecting language, selecting a hostname, configuring networking. Then there is sharing and remote access. SSH can be enabled but needs the default password to be changed for security. This can be changed and recorded. The samba share should be enabled by default and is needed if files will be added to the device remotely. For this project files were not moved to the Kodi device.

Before moving forward, the IP and MAC addresses of the Raspberry Pi 2 needs to be recorded. This can be done by accessing the settings (gear icon.) Selecting "System information" and then "Network." The IP and MAC addresses located here. Record these two addresses. Now access the routers settings like before when setting up the Home Assistant device. Access the DHCP

settings and reserve an IP address with the IP and MAC addresses that were recorded. Once this changed is completed it is a good practice to restart the router to make sure these settings are enabled. Now this device will always be assigned the same IP address whenever it is connected to this network.

5.4.3 Integration into Home Assistant

The integration into Home Assistant is very simple because Home Assistant usually will find your new Kodi device automatically. If it is automatically found it will appear in the notifications of Home Assistant. If not, the integration can also be added by accessing “Configuration” then “Integrations” and “Add Integration.” Configuration of the integration starts with enabling web sockets. This will not be needed and as stated the port number needs to be deleted and submitted. To finish up the setup the device needs a location in the house. In this project the device will be in the living room. The configuration documentation is all recorded in the Home Assistant integrations page for Kodi. (10.)

The final piece to get text-to-speech enabled and working is to add to the “configuration.yaml” file. According to the Text-to-Speech (TTS) documentation for Home Assistant to send the media file there needs to be some code added to the “configuration.yaml” file. The internal address does not have HTTPS enabled. (16). In this project the following code shown in Figure 22 was needed.

```
#####  
# Text-to-speech  
#####  
tts:  
  - platform: google_translate  
  - cache: true  
  - cache_dir: ./config/tts  
  - time_memory: 300  
  - base_url: http://192.168.2.150:8123  
  - service_name: google_translate_say  
  - language: "en"
```

Figure 22 Text-to-Speech YAML code to enable TTS message sent to Kodi media player.

5.5 Integration of Google Calendar

The Google calendar integration will be used to pull in events from a calendar that used by the caregiver or family member. They will be able to add important events to the calendar. These events will trigger a notification to the user’s mobile phone and/or announce it on the Raspberry Pi 2

device's speaker. There can be multiple uses with this integration. Announcing important events like a doctor's appointment or a medication reminder.

5.5.1 Configuration in Google Developers Console

To use the Google calendar API, a Google account is needed. Any Gmail address is considered a Google account. This will enable you to access the google developer's console. The integration has great documentation, and this is the info that was needed to create the API connection to Home Assistant.

According to the Google Calendar Event documentation after getting access to the console, you will need to create a new project. This can be done by accessing this website. <https://console.developers.google.com/start/api?id=calendar>. Create a new project and click continue. The calendar API should be automatically enabled, but if not select "Enable" if needed. Go to credentials and this can be canceled, it is not needed. Go to credentials that is in the left sidebar. There the "Configure consent screen" needs to be clicked. Select "External" and create. The "app name" was "Home-Assistant" for this project. "User support email" can be your Gmail account that is being used. The "App domain" can be left blank" The "Developer contact information" will need an email address. The same Gmail address will be used from earlier. Then select "Save and continue". The "OAuth consent screen" will be loaded automatically. In the "Scopes" tab there is nothing to be added. Select "Save and continue" In the "Test users" tab, select "Add Users" and add the Gmail address that has been used and select add. Select "Save and Continue" The "Summary" tab will be loaded, here it will show what you provided earlier. Nothing needs to be completed here. (7.)

According to the Google Calendar Event documentation, select "Credentials" in the left sidebar. Here credentials need to be made for Home Assistant to be granted access. Select "Create Credentials" and then "OAuth client ID". Set the application type to "TV and Limited Input devices" and give it a name. This is shown in Figure 23. After clicking "Create" a popup will show with the "Client ID" and your "Client Secret". These two need to be recorded for the Home Assistant configuration.yaml file. (7.)

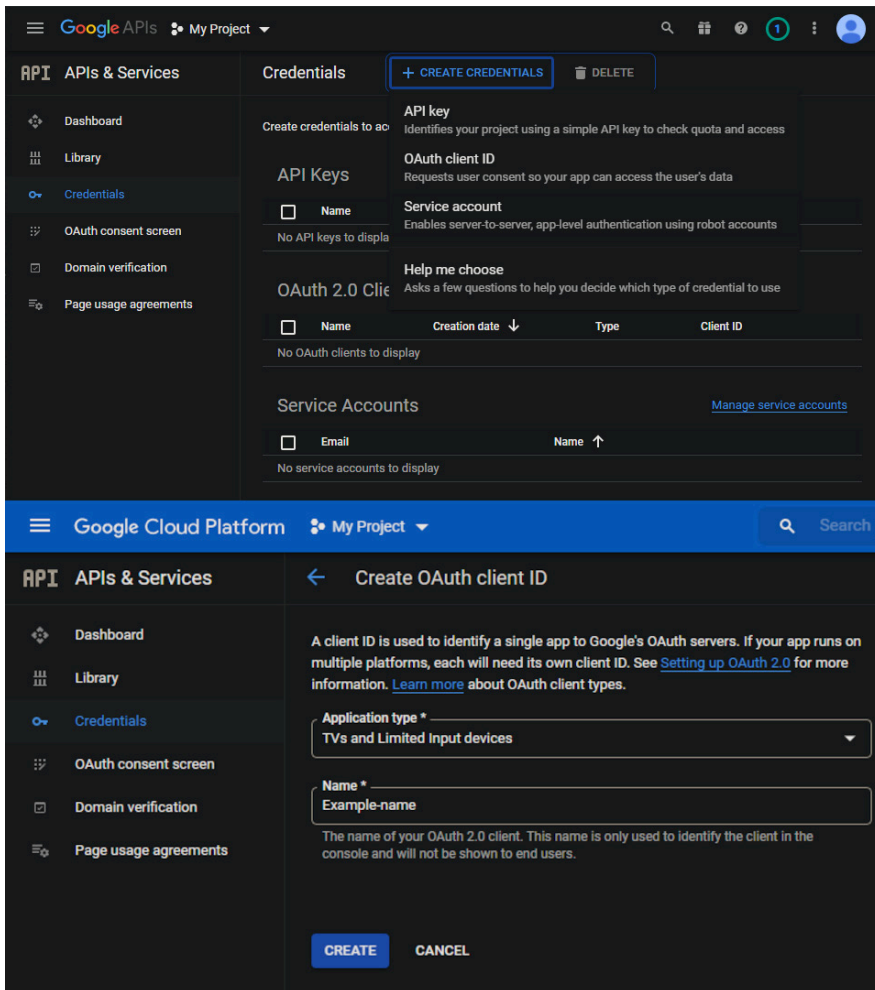


Figure 23 Create credentials and giving name to TVs and limited input devices pages.

5.5.2 Integration into Home Assistant

According to the Google Calendar Event documentation, with the Google calendar API set up and enabled it can now be integrated into Home Assistant. To start the configuration.yaml needs to be opened in either Visual Studio Code or the File Editor add-ons. The code in Figure 24 needs to be added to the configuration.yaml file and then the file can be saved. (7.)

```

32 # Google Calendar API
33 google:
34   - client_id: YOUR_CLIENT_ID
35   - client_secret: YOUR_CLIENT_SECRET

```

Figure 24 Google Calendar API code in configuration.yaml.

After the code is added for it to go into effect Home Assistant needs to be restarted. After the restart there will be a new notification with a link and code shown like in Figure 25. Copy the code and click the link. Paste the code and click next. Sign into the Gmail account used. This app is not verified by Google, but you have created it. Click continue and allow. Now the connection between the Google calendar API and your Home Assistant instance is completed. Now back at your Home Assistant dashboard you will have received another notification showing there are calendars that have been found.

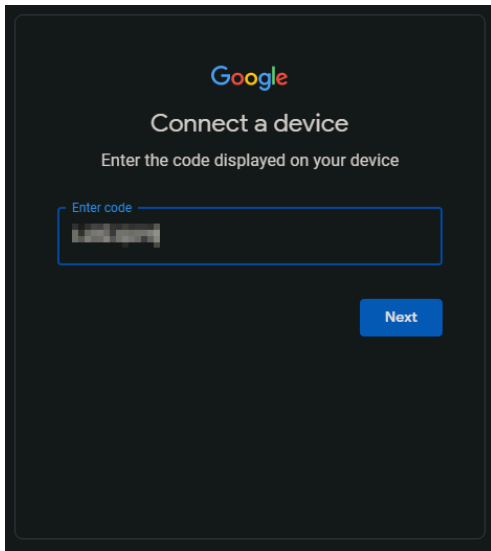


Figure 25 The code needs to be pasted here to enable access.

There will be times when the Google calendar id needs to be found for example in the Calendar Reminder automation Appendix 1. This calendar id can be found in the `google_calendars.yaml` file. This YAML file was created when the Google calendar integration was completed and successful.

6 AUTOMATIONS AND INTEGRATIONS

Part of the goal of setting up system for seniors was for the system to help with different tasks and strictly not to rely on this system. This system is not meant to replace a caregiver or family member, but just to assist with the tasks.

In this chapter, the focus will be on using all the add-ons and devices that have been integrated into Home Assistant. There will be medication reminder automations. A reminder automation for events. Reminder automation for charging the user's mobile phone battery. There will be notifications to the caregiver or family member and how they can interact with that information. User device monitoring if away for an extended period of time.

There was helpful documentation that was used to create the automations and scripts. These documentation links are provided in the references. The link provided goes to the Home Assistant documentation. Where the following topics were read to help with creating these unique YAML code. The topics are under Automation: Triggers, Conditions, Actions, Services, Templates, and YAML. The topic: Script was also used. (18).

The full configuration.yaml files are shown in the following appendices:

Configuration.yaml file part 1 appendix 3

Configuration.yaml file part 2 appendix 4

Configuration.yaml file part 3 appendix 5

6.1 Medication Reminder automation

The medication reminder automation is collection of multiple automations and scripts to aid a user with remembering to take their medication. The goal is not to rely on the system, but only to support the user. The medication reminder system works by pulling medication calendar events through the google calendar integration. A family member or caregiver can add and remove medication events in google calendar. A working overview is shown in Figure 26.

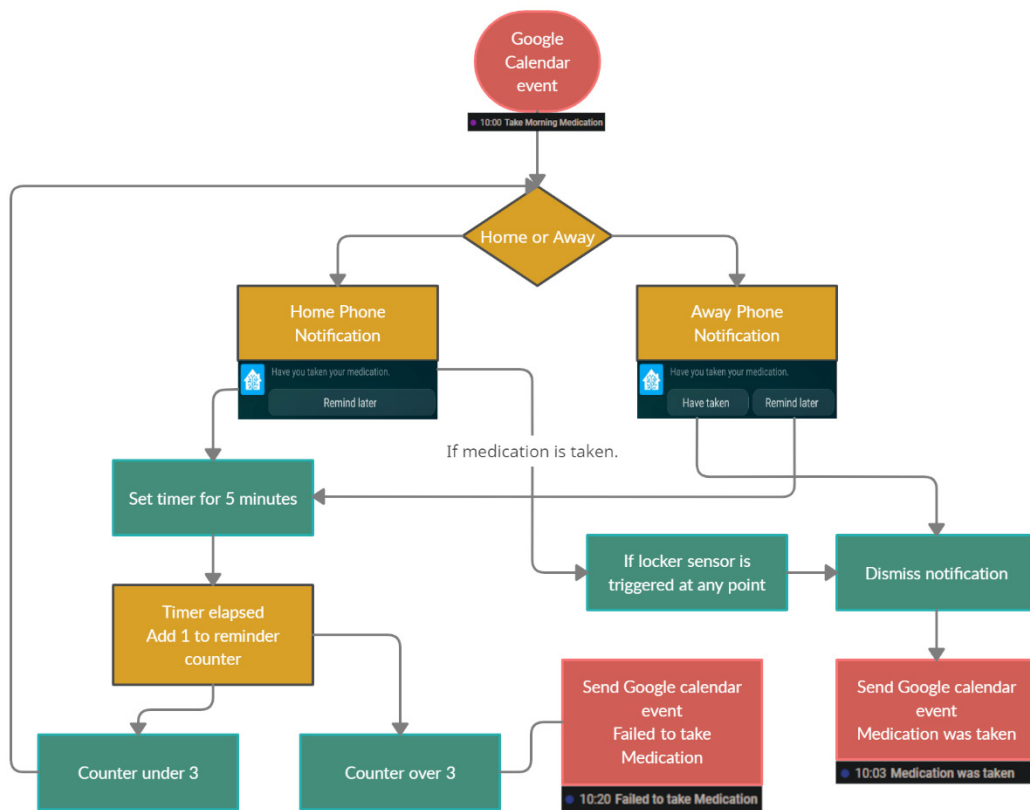


Figure 26 Flowchart of the medication reminder system.

When a new medication event is received from the Google calendar API, the automation is fired off. The automation checks if the user is home or away by using the location data that is sent to Home Assistant through the Home Assistant companion android application on the user's mobile phone.

Depending on they are home or away they will receive a notification to take their medication. If they are at home, they will need to go to the medication locker to get that notification cleared. The user does have the option to select "Remind later." Even though they can select that it will assume this is true if there is no option selected. A timer will be activated for specific period for example five minutes. They will receive another notification as soon as that time is up. Each time this timer runs out a counter will count how many times this has happened. In this project the max amount is three. When the counter has reached three, the last notification reminder will be sent to the user's mobile phone as well as the media center speaker as shown in section 6.2.

If the medication is taken it will be recorded into the google calendar. This way the family member or caregiver can see the history. If the counter has reached the limit, it will record that the medication was not taken. The objective of the medication reminder automation is to aid the user with remembering to take their medications. The system is not meant to replace person's responsibility to take their medication but to be helpful.

All the automation and script's code that was written for this to work are in the following appendices:

Calendar Reminder automation appendix 1

Clear mobile notification script appendix 2

Failed to take Medication automation appendix 8

Google calendar event medication failed script appendix 11

Google calendar event medication taken script appendix 12

Input Boolean reminder if away from home appendix 13

Input Boolean voice player notification appendix 14

Input Booleans related to Medication automations appendix 15

Medication failed week counter appendix 16

Medication locker opened automation appendix 17

Medication reminder script appendix 18

Medication taken script appendix 19

Medication timer appendix 20

Medication timer counter appendix 21

Medication timer finished automation appendix 22

Notify away medication reminder automation appendix 24

Notify have taken away action automation appendix 25

Notify home medication reminder automation appendix 26

Notify Kodi TTS script appendix 27

Notify remind later away action automation appendix 28

Notify remind later home action automation appendix 29

Reminder away from home appendix 31

Set away mode for reminders automation appendix 32

Turn on Input Booleans for away or for home script appendix 33

Whether home or away notification script appendix 35

6.2 Audio played through Media center.

The Kodi LibreELEC multimedia center will play audio through a speaker connected to the Raspberry Pi 2 device. A script was created to take in data from another script or automation and then execute the audio to the multimedia center. The script in Figure 27 uses template data so that the data can be dynamic.

```
#####
#Notify through kodi tts message to take medication.
#Change volume up to 50% and then change down to 20% volume.
#####
notify_kodi_tts:
- sequence:
- - service: media_player.volume_set
- - data:
- - - entity_id: media_player.livingroom
- - - volume_level: 0.5
- - service: tts.google_translate_say
- - entity_id: media_player.livingroom
- - data_template:
- - - message: "{{ message }}"
- - delay: 20
- - service: media_player.volume_set
- - data:
- - - entity_id: media_player.livingroom
- - - volume_level: 0.2
```

Figure 27 Audio YAML script using Kodi LibreELEC.

6.3 Event reminder automations.

The event reminder automations are meant to aid the user in remembering when important events are going to take place. These automations can be used in multiple ways. In this project the events will be played on the media center speaker once one hour before the scheduled event. The automation is in Appendix 6 and using the script as shown in Figure 27 of section 6.2 .

6.4 User has failed to take medication (Telegram)

The Telegram integration is being used to notify the caregiver or family member of events that happen. Telegram is installed on the caregiver or family member's mobile phone.

There are three-part automations that are used to notify the caregiver or family member about the user failing the medication reminders. First, there the med_add_calendar_event_failed.yaml script shown in Appendix 11 that also sends the failed medication reminder event to the Google Calendar. The second part of the script sends a Telegram message to the caregiver or family member that

the user failed the medication reminder. The third part is the counter for the next section and Input Boolean seen in Appendix 16.

The second automation is the weekly message of how many times the medication reminders were failed. The message uses the Input Boolean that is Appendix 16. This message is sent every Sunday at 18:00 in the evening. It is also then reset for the next week. This would be a method of following how the reminder system is working with the user. Then the issue can be known of and resolved with more interacting with the user. This automation is seen in Appendix 30.

6.5 Has the user opened the refrigerator? (Telegram)

Like the previous automation in 6.4 the Telegram integration is being used to notify the caregiver or family member of events that happen.

This automation is using a simple door sensor to record how many times the refrigerator door is opened. If the user has not opened the refrigerator more than two times, one could assume that they have not eaten enough for that day. Of course, the number of times that the refrigerator is opened is completely customizable. At 19:30 in the evening the system uses the notification system to notify the caregiver or family member as shown in section 6.4. The automation that can be seen in Appendix 9 and the counter in Appendix 10.

6.6 User device monitoring for safety (Telegram)

When the user leaves the general area of the home the system can notify through the Telegram application to a caregiver or family member. To give a bit more privacy and independence the system is only notifying when the user is away from the area for more than two hours. The automation will reset after the user has returned home. Automation YAML code can be seen in Appendix 34.

6.7 Low battery level charge reminder

A notification reminding the user to charge their phone is sent to the user's mobile phone and to the media center speaker when the mobile phone battery level is below twenty percent. This reminder will be triggered every five minutes until it has been plugged in to the charger. The

reminder is only activated if the time is between eight in the morning to eight in the evening.
Automation YAML code is in Appendix 23.

7 DISCUSSION

The testing of the system was completed by having the system in use by the author. The author simulated the use of the system through a couple of weeks. Testing the system was done to make sure bugs are found and then fixed.

The system was successful in the testing with minor bugs, and these were easily fixed. One problem was making sure IP addresses were correctly configured and that they were static in the router settings. This was solved by just making sure that the router was saving the IP addresses when adding them to the router settings.

The Telegram application was also a challenge when testing and confirming that the information was correct and was sent to the family member or caregiver.

One problem that was not addressed completely in the project was the delay there was when the internet connection with the mobile devices were spotty. When the devices did connect to the network, they received the notifications and information without problem. This problem is not something that could be addressed in this project and would need to address the redundancy of the mobile connections.

The base system is setup in a way to be a good first deployment. The system worked as planned and was in scope of what the objective was from the beginning of the project. The system was able to send the correct notifications and fired off the corrected scripts and automations as designed.

8 CONCLUSION

The smart home automation system for seniors was successfully completed and tested. The project discussed and demonstrated the development process of the system. The system was able to display that it is possible to create a system that can help a senior. It was also able to aid the caregiver or family member with data and a way to interact with the system.

There were many ideas that would improve the system. For example, the medication locker would have a smart lock instead of the door sensor. A specific timeframe for taking the medication could be added and would prevent the medication locker from being opened outside of this timeframe. For this project the Google calendar was used to show data to the caregiver or family member. Better metrics could be gathered and stored by using a database. The data could then be displayed to the caregiver or family member in a user-friendly dashboard.

Adding a device that tracks activity and sleep could be added to the system and displayed on a dashboard.

There also could be a system added that would add seamless video calling to check up or socialize with the senior. A great addition would be a system that would track where within the home the senior is located. There could be multiple use cases for example, turning on and off lights depending on where the user is. As well as being able to notify a family member if a user is in a certain room for extended periods of time.

Another improvement would be to the multimedia player. This could also have video or music that could be played to the senior when they wanted it.

The last addition would be voice controlled devices. The senior could interact with the smart home this way. Turning on and off lights, multimedia players, and asking for help directly to the caregiver or family member.

After the system was completed and tested. It is obvious that a smart home automation system for seniors would be extremely beneficial to all parties involved. Seniors at home need help and the caregivers and family members need to balance their daily lives. These types of systems would aid in that flexibility without hurting the quality of care.

All the code that was created for this thesis and project are included in the appendices.

9 DISCLAIMER

The disclaimer notice below applies to the software (code) included in the project.

THE SOFTWARE (CODE) IN THIS THESIS IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

REFERENCES

1. Home Assistant. 2020. Advanced Configuration. Cited 15.7.2020, <https://www.home-assistant.io/getting-started/configuration/>.
2. Home Assistant. 2020. Home Assistant Add-on: DuckDNS. Cited 29.7.2020, <https://github.com/home-assistant/addons/blob/master/duckdns/DOCS.md/>.
3. Home Assistant. 2020. Home Assistant Add-on: NGINX Home Assistant SSL proxy. Cited 1.8.2020, https://github.com/home-assistant/addons/blob/master/nginx_proxy/DOCS.md/.
4. Home Assistant. 2020. Home Assistant Add-on: Samba share. Cited 20.7.2020, <https://github.com/home-assistant/addons/blob/master/samba/DOCS.md/>.
5. Home Assistant. 2020. Getting Started. Cited 5.7.2020, https://companion.home-assistant.io/docs/getting_started/getting-started/.
6. Home Assistant. 2020. Actionable Notifications. Cited 20.8.2020, <https://companion.home-assistant.io/docs/notifications/actionable-notifications/>.
7. Home Assistant. 2020. Google Calendar Event. Cited 20.8.2020, <https://www.home-assistant.io/integrations/calendar.google/>.
8. Home Assistant. 2020. Home Assistant Add-on: Dnsmasq. Cited 1.8.2020, <https://github.com/home-assistant/addons/blob/master/dnsmasq/DOCS.md/>.
9. Home Assistant. 2020. Installation. Cited 15.7.2020, <https://www.home-assistant.io/installation/>.
10. Home Assistant. 2020. Kodi. Cited 10.9.2020, <https://www.home-assistant.io/integrations/kodi/>.

11. Home Assistant. 2020. Notification Cleared. Cited 23.8.2020, <https://companion.home-assistant.io/docs/notifications/notification-cleared/>.
12. Home Assistant. 2020. Onboarding. Cited 15.7.2020, <https://www.home-assistant.io/getting-started/onboarding/>.
13. Home Assistant. 2020. Telegram bot. Cited 3.10.2020, https://www.home-assistant.io/integrations/telegram_bot/.
14. Home Assistant. 2020. Telegram polling. Cited 3.10.2020, https://www.home-assistant.io/integrations/telegram_polling/.
15. Home Assistant. 2020. Telegram. Cited 3.10.2020, <https://www.home-assistant.io/integrations/telegram/>.
16. Home Assistant. 2020. Text-to-Speech (TTS). Cited 10.9.2020, <https://www.home-assistant.io/integrations/tts/>.
17. Home Assistant. 2020. Zigbee Home Automation. Cited 10.8.2020, <https://www.home-assistant.io/integrations/zha/#configuration---gui>.
18. Home Assistant. Documentation. Cited 5.7.2020, <https://www.home-assistant.io/docs/>.
19. Intel Corporation. 2012. USB 3.0* Radio Frequency Interference Impact on 2.4GHz Wireless Devices. Cited 10.8.2020, <https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/usb3-frequency-interference-papers.pdf/>.
20. Kela. 7.11.2017. Eligibility. Kela. Cited 1.9.2020, <https://www.kela.fi/web/en/care-allowance-for-pensioners-eligibility/>.
21. Kela. 7.3.2019. Basic social assistance for pensioners. Cited 1.9.2020, <https://www.kela.fi/web/en/social-assistance-pensioners/>.

22. Kenton, W. 28.05.2021. The Internet of Things (IoT). Cited 10.10.2021, <https://www.investopedia.com/terms/i/internet-things.asp/>.
23. LibreElec. 2018. Raspberry Pi 2. Cited 10.9.2020, https://libreelec.tv/downloads_new/raspberry-pi-2/.
24. Miller, M. 2017. My Smart Home for Seniors. Cited 15.6.2020, <https://learning.oreilly.com/library/view/smart-home-automation/9781430258872/?ar>.
25. Nabu Casa. 2020. Remote UI. Cited 29.7.2020, <https://www.nabucasa.com/config/remote/>.
26. Raspberry Pi Foundation. 2015 Raspberry Pi 2 Model B. Cited 20.9.2020, <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>.
27. Raspberry Pi Foundation. 2019. Raspberry Pi 4 Tech Specs. Cited 7.7.2020, <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>.
28. Raspberry Pi Foundation. 2020. About us. Cited. 10.7.2020, <https://www.raspberrypi.org/about/>.
29. Techopedia. 12.06.2018. Home Automation System. Cited 10.8.2021, <https://www.techopedia.com/definition/29999/home-automation-system/>.
30. Telegram Messenger Inc. 2020. Bots: An introduction for developers. Cited 3.10.2020, <https://core.telegram.org/bots/>.
31. Telegram. 2020. Telegram a new era of messaging. Cited 01.10.2020, <https://www.telegram.org/>.
32. Wikipedia. 2021. Kela (institution). Cited 30.4.2021, [https://en.wikipedia.org/wiki/Kela_\(institution\)](https://en.wikipedia.org/wiki/Kela_(institution)).

33. Zigbee Alliance. 2015. Zigbee Control your world. Cited 02.02.2021, <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf/>.

Appendices

- Calendar Reminder automation appendix 1
- Clear mobile notification script appendix 2
- Configuration.yaml file part 1 appendix 3
- Configuration.yaml file part 2 appendix 4
- Configuration.yaml file part 3 appendix 5
- Event reminder automation appendix 6
- External access changes to configuration.yaml appendix 7
- Failed to take Medication automation appendix 8
- Fridge has been opened count automation appendix 9
- Fridge opened counter appendix 10
- Google calendar event medication failed script appendix 11
- Google calendar event medication taken script appendix 12
- Input Boolean reminder if away from home appendix 13
- Input Boolean voice player notification appendix 14
- Input Booleans related to Medication automations appendix 15
- Medication failed week counter appendix 16
- Medication locker opened automation appendix 17
- Medication reminder script appendix 18
- Medication taken script appendix 19
- Medication timer appendix 20
- Medication timer counter appendix 21
- Medication timer finished automation appendix 22
- Mobile phone low battery reminder automation appendix 23
- Notify away medication reminder automation appendix 24
- Notify have taken away action automation appendix 25
- Notify home medication reminder automation appendix 26
- Notify Kodi TTS script appendix 27
- Notify remind later away action automation appendix 28
- Notify remind later home action automation appendix 29
- Notify Telegram failed med reminders for the week appendix 30
- Reminder away from home appendix 31

Set away mode for reminders automation appendix 32

Turn on Input Booleans for away or for home script appendix 33

User away from home for extended time automation appendix 34

Whether home or away notification script appendix 35

```
#####
#MedicationReminderautomationtriggeredbyaGoogleCalendarMedicationReminder.
#TurnonthemedicationLockerautomationsothatitcannotbeactivatedwithoutthecalendarevent.
#Medicationtakenisreset.
#Counterisreset.
#Startcounter,addingone.
#Startscripttocheckifathomeoraway.
#####

alias:~"MedCalendarReminder"
id:~6835e7d3-d5a1-42a9-86a5-857235fbd3d4
trigger:
  ~platform:~state
  ~entity_id:~calendar.medication_calendar
  ~to:~"on"

action:
  ~-~service:~automation.turn_on
  ~-~entity_id:~automation.medication_locker_opened
  ~-~service:~input_boolean.turn_off
  ~-~entity_id:~input_boolean.med_is_taken
  ~-~service:~counter.reset
  ~-~entity_id:~counter.med_reminder_amount
  ~-~delay:~5
  ~-~service:~counter.increment
  ~-~entity_id:~counter.med_reminder_amount
  ~-~service:~script.med_home_or_away_notification
```

```
#####  
#·Remove·the·persistent·notification.  
#·clear_notification·to·the·tag·that·we·defined.  
#####  
med_clear_mobile_notification:  
  .. sequence:  
  .. .. service: notify.mobile_app_plk_l01·#The·moible·phone·notify.entity_id#  
  .. .. data:  
  .. .. .. message: clear_notification  
  .. .. .. data:  
  .. .. .. tag: med_notifications_group
```

```
#Home-Assistant-Configuration.yaml
default_config:
mobile_app:
discovery:

group: !include groups.yaml
automation: !include automations.yaml
script: !include scripts.yaml
scene: !include scenes.yaml

#####
#Text-to-speech
#####
tts:
  - platform: google_translate
  - cache: true
  - cache_dir: /config/tts
  - time_memory: 300
  - base_url: http://192.168.2.150:8123
  - service_name: google_translate_say
  - language: "en"

#####
#Home-Assistant-auth-providers
#####
homeassistant:
  - auth_providers:
    - type: homeassistant
    - type: trusted_networks
    - trusted_networks:
      - 192.168.2.0/24

#####
#DuckDNS+Letsencrypt-settings
#####
http:
  - use_x_forwarded_for: true
  - trusted_proxies:
    - 172.30.33.0/24
  - #...ssl_certificate: /ssl/fullchain.pem
  - #...ssl_key: /ssl/privkey.pem
  - ip_ban_enabled: true
  - login_attempts_threshold: 5
```

```
#####
#-Google-Calendar-API
#####
google:
  - client_id: !secret google_cal_client_id
  - client_secret: !secret google_cal_client_secret

#####
#-Telegram-configuration
#####
telegram_bot:
  - platform: polling
  - api_key: !secret telegram_api_token
  - allowed_chat_ids:
    - !secret my_chat_id

#####
#-Notify-configuration
#####
notify:
  - #-Telegram-configuration
  - name: smarthome_bot
  - platform: telegram
  - chat_id: !secret my_chat_id

#####
#TIMERS
#####
timer:
  - med_timer:
    - duration: "00:05:00"
    - name: Medication timer
```

```
#####
#INPUT-BOOLEANS
#####
input_boolean:
  - med_is_taken:
    - name: Medication is Taken
    - icon: mdi:pill
    - initial: on

  - med_notify_home_activate:
    - name: Medication Notify Home Activate
    - initial: off

  - med_notify_away_activate:
    - name: Medication Notify Away Activate
    - initial: off

  - voice_player_notification:
    - name: Voice player Notification
    - initial: on
    - icon: mdi:speaker

  - med_away_from_home:
    - name: Away From Home
    - initial: off
    - icon: mdi:beach

#####
#COUNTERS
#####
counter:
  - med_reminder_amount:
    - name: Reminder Amount
    - initial: 0
    - step: 1
    - maximum: 5

  - fridge_opened:
    - name: Fridge Opened
    - initial: 0
    - step: 1

  - med_failed_week_count:
    - name: Failed Med count for Week
    - initial: 0
    - step: 1
```

```
#####
#·Event·Reminder·automation·triggered·by·a·Google·Calendar·Event.
#·Trigger·automation·when·a·new·event·is·read·from·the·Google·Calendar.
#·Title·of·event·is·the·message·when·notifying.
#####

alias: "Event Reminder"
id: f5f848c5-03ba-4bba-abec-7ca1c8fb6617
trigger:
  -- platform: state
  -- entity_id: calendar.events
  -- to: "on"

action:
  -- service: notify.mobile_app_plk_l01
  -- data:
  -- -- message: "You have a {{ states.calendar.events.attributes.message }} in an hour."
  -- condition: template
  -- value_template: "{{ states('input_boolean.voice_player_notification') == 'on' }}"
  -- service: script.notify_kodi_tts
  -- data:
  -- -- message: >
  -- -- -- You have a {{ states.calendar.events.attributes.message }} in an hour.
```

```
#####  
# Home Assistant auth providers  
#####  
homeassistant:  
  - auth_providers:  
    - type: homeassistant  
    - type: trusted_networks  
    - trusted_networks:  
      - 192.168.2.0/24  
  
#####  
# DuckDNS + Letsencrypt settings  
#####  
http:  
  - use_x_forwarded_for: true  
  - trusted_proxies:  
    - 172.30.33.0/24  
  - # ssl_certificate: /ssl/fullchain.pem  
  - # ssl_key: /ssl/privkey.pem  
  - ip_ban_enabled: true  
  - login_attempts_threshold: 5
```

```
#####
#·Did·not·take·medication,·will·not·be·reminded,·send·info·to·calendar.
#####

alias:·"Failed·To·Take·Medication"
id:·6e173192-fbeb-4228-b32c-80d25924aba3
initial_state:·true
trigger:
  ····platform:·template
  ····value_template:·"{{·states('counter.med_reminder_amount')·>=·'5'·}}"

action:
  ····service:·script.med_add_calendar_event_failed
  ····service:·script.clear_mobile_notification
  ····service:·automation.turn_off
  ····entity_id:·automation.medication_locker_opened
```

```
#####
# The user has not opened the fridge more than two times during the day.
# Caretaker or family member is notified through Telegram.
#####

alias: "Fridge Times Opened"
id: b698f7d1-8e72-46f6-a876-23f837f45547
trigger:
  -- platform: state
  -- entity_id: binary_sensor.fridge_door_sensor #zigbee-door-sensor
  -- from: "on"
  -- to: "off"

action:
  -- service: counter.increment
  -- entity_id: counter.fridge_opened
  -- wait_for_trigger:
    -- platform: time
    -- at: "19:00:00"
  -- condition: numeric_state
  -- entity_id: counter.fridge_opened
  -- below: 5
  -- service: telegram_bot.send_message
  -- data:
    -- message: >
    -- Fridge was opened only {{ states('counter.fridge_opened') }} times today.
  -- delay:
    -- seconds: 5
  -- service: counter.reset
  -- entity_id: counter.fridge_opened
mode: restart
```

```
#####  
#COUNTERS.in.configuration.yaml  
#####  
counter:  
  - fridge_opened:  
    - name: Fridge-Opened  
    - initial: 0  
    - step: 1
```

```
#####
#·google·calendar·event·--failed·to·take·medication
#·Send·a·telegram·notification
#·Add·one·to·the·weekly·failed·med·counter.
#####
med_add_calendar_event_failed:
  ··sequence:
  ····-·service:·google.add_event
  ····-·data_template:
  ····-·calendar_id:·[REDACTED]@group.calendar.google.com
  ····-·summary:·"Failed to take Medication"
  ····-·start_date_time:·"{{·now().strftime('%Y-%m-%d %H:%M:%S')·}}"
  ····-·end_date_time:·"{{·now().strftime('%Y-%m-%d %H:%M:%S')·}}"
  ····-·service:·telegram_bot.send_message
  ····-·data:
  ····-·message:·The·User·just·missed·their·medication.
  ····-·title:·Medication·reminder.
  ····-·service:·counter.increment
  ····-·entity_id:·counter.med_failed_week_count
```

```
#####  
# google.calendar.event -- medication.taken  
#####  
med_add_calendar_event_taken:  
  -- sequence:  
  -- -- service: google.add_event  
  -- -- data_template:  
  -- -- -- calendar_id: [REDACTED]@group.calendar.google.com  
  -- -- -- summary: "Medication Taken"  
  -- -- -- start_date_time: "{{now().strftime('%Y-%m-%d %H:%M:%S')}}"  
  -- -- -- end_date_time: "{{now().strftime('%Y-%m-%d %H:%M:%S')}}"
```

```
#####  
#INPUT-BOOLEANS-added-to-configuration.yaml  
#####  
input_boolean:  
- reminder_away_from_home:  
- - name: Away From Home  
- - initial: off
```

```
#####  
#INPUT-BOOLEANS-added-to-configuration.yaml  
#####  
input_boolean:  
  - voice_player_notification:  
    - name: Voice player Notification  
    - initial: on  
    - icon: mdi:speaker
```

```
#####  
#INPUT-BOOLEANS-added-to-configuration.yaml  
#####  
input_boolean:  
  - med_is_taken:  
    - name: Medication is Taken  
    - initial: on  
  
  - med_notify_home_activate:  
    - name: Medication Notify Home Activate  
    - initial: off  
  
  - med_notify_away_activate:  
    - name: Medication Notify Away Activate  
    - initial: off  
  
  - voice_player_notification:  
    - name: Voice player Notification  
    - initial: on
```

```
#####  
#COUNTERS-in-configuration.yaml  
#####  
counter:  
  - med_failed_week_count:  
    - name: Failed-Med-count-for-Week  
    - initial: 0  
    - step: 1
```

```
#####  
#·Medication·Locker·has·been·opened.  
#·Execute·medication_taken·script.  
#####  
  
alias:·"Medication·Locker·Opened"  
id:·ee2b8da9-ee86-4e99-9b15-4bd23a299277  
initial_state:·false  
trigger:  
  ---platform:·state  
  ---entity_id:·binary_sensor.medication_door_sensor·#zigbee·door·sensor  
  ---from:·"on"  
  ---to:·"off"  
  
action:  
  ---delay:  
  ---·seconds:·5  
  ---service:·script.medication_taken
```

```
#####
#·Notifies·to·take·medication
#·Voice·notifications·if·at·home
#####
med_reminder:
--sequence:
--condition: template
--value_template: "{{ states('counter.med_reminder_amount') <= '4' }}"
--condition: template
--value_template: "{{ states('input_boolean.med_is_taken') == 'off' }}"
--service: script.med_home_or_away_notification
--condition: template
--value_template: "{{ states('input_boolean.voice_player_notification') == 'on' }}"
--condition: template
--value_template: "{{ states('counter.med_reminder_amount') >= '3' }}"
--wait_template: "{{ is_state('media_player.livingroom', 'idle') }}"
--continue_on_timeout: 'false'
--timeout: 00:00:30
--service: script.notify_kodi_tts
--data:
--message: >
--This is a reminder for you to take your medication. Please take your medication.
```

```
#####
# Stops timer, Records that the medication are taken, and notifies
#####
medication_taken:
--sequence:
-- --service: script.med_add_calendar_event_taken
-- --service: timer.cancel
-- --entity_id: timer.med_timer
-- --service: input_boolean.turn_on
-- --entity_id: input_boolean.med_is_taken
-- --service: automation.turn_off
-- --entity_id: automation.medication_locker_opened
-- --service: script.med_clear_mobile_notification
-- --condition: template
-- --value_template: "{{ states('input_boolean.voice_player_notification') == 'on' }}"
-- --condition: template
-- --value_template: "{{ states('counter.med_reminder_amount') >= '3' }}"
-- --service: script.notify_kodi_tts
-- --data:
-- -- --message: ->
-- -- -- Thank you for taking your medication.
```

```
#####  
#TIMERS-added-to-configuration.yaml  
#####  
timer:  
- med_timer:  
  - duration: "00:05:00"  
  - name: Medication timer
```

```
#####  
#COUNTERS in configuration.yaml  
#####  
counter:  
  - med_reminder_amount:  
    - name: Reminder Amount  
    - initial: 0  
    - step: 1  
    - maximum: 5
```

```
#####  
#-Timer-Finished--Time-for-a-reminder-if-the-medication-was-not-taken-yet  
#####  
  
alias:-"Med-Timer-Finished"  
id:-34df9cd1-4ea1-4720-963d-801f651d99f0  
initial_state:-true  
trigger:  
  --platform:-event  
  --event_type:-timer.finished  
  --event_data:  
    --entity_id:-timer.med_timer  
  
action:  
  --service:-counter.increment  
  --entity_id:-counter.med_reminder_amount  
  --delay:-5  
  --service:-script.med_reminder
```

```
#####
#Remind-user-to-charge-their-phone-if-battery-is-below-a-certain-percentage.
#Project-example-20-percent.
#####

alias: "Mobile Phone Low Battery"
id: 2f10d54b-828f-4506-8d1c-94d098bef6c3
initial_state: true
trigger:
  - platform: numeric_state
    - entity_id: sensor.plk_l01_battery_level
    - below: 21

condition:
  - condition: time
    - after: "08:00:00"
    - before: "23:00:00"
  - condition: state
    - entity_id: binary_sensor.plk_l01_is_charging
    - state: "off"

action:
  - service: notify.mobile_app_plk_l01
    - data:
      - message: "Please charge your mobile phone."
      - condition: template
      - value_template: "{{ states('input_boolean.voice_player_notification') == 'on' }}"
    - service: script.notify_kodi_tts
      - data:
        - message: "Please charge your phone. The current battery level is at {{ states('sensor.plk_l01_battery_level') }} percent."
        - delay: 300
mode: single
```

```
#####  
#·Away·#·Notify·whether·user·has·taken·or·reminder·later·actionable·notification  
#####  
  
alias: "Notify Away Medication Reminder"  
id: a7698192-b438-432a-9527-7da8b085d638  
trigger:  
  platform: state  
  entity_id: input_boolean.med_notify_away_activate  
  to: "on"  
  
action:  
  service: input_boolean.turn_off  
  entity_id: input_boolean.med_notify_away_activate  
  service: notify.mobile_app_plk_l01  
  data:  
    message: "Have you taken your medication."  
    data:  
      persistent: true  
      tag: med_notifications_group  
      actions:  
        action: notify_have_taken_away  
        title: "Have taken"  
        action: notify_remind_later_away  
        title: "Remind later"
```

```
#####  
#·Away·#·User·has·taken·medication·send·foward·that·med·is·taken.  
#####  
  
alias: "Notify·Have·Taken·Away·Action"  
id: 41cf0e8e-62bd-4edd-8110-53e5122df9e3  
trigger:  
  -- platform: event  
  -- event_type: mobile_app_notification_action  
  -- event_data:  
  -- -- action: notify_have_taken_away  
  
action:  
  -- -- service: script.med_clear_mobile_notification  
  -- -- delay: 5  
  -- -- service: script.medication_taken
```

```
#####  
#·Home·#·Notify·whether·user·wants·to·reminder·later·actionable·notification  
#·Will·be·reminded·whether·this·option·is·selected·or·not  
#####  
  
alias: "Notify Home Medication Reminder"  
id: a959b997-049e-4fcd-a7c1-bf2664e02214  
trigger:  
  platform: state  
  entity_id: input_boolean.med_notify_home_activate  
  to: "on"  
  
action:  
  service: input_boolean.turn_off  
  entity_id: input_boolean.med_notify_home_activate  
  service: notify.mobile_app_plk_l01  
  data:  
    message: "Have you taken your medication."  
    data:  
      persistent: true  
      tag: med_notifications_group  
      actions:  
        action: notify_remind_later_home  
        title: "Remind later"
```

```
#####  
# Notify through kodi tts message to take medication.  
# Change volume up to 50% and then change down to 20% volume.  
#####  
notify_kodi_tts:  
  - sequence:  
    - - service: media_player.volume_set  
      - data:  
        - entity_id: media_player.livingroom  
        - volume_level: 0.5  
    - - service: tts.google_translate_say  
      - entity_id: media_player.livingroom  
      - data_template:  
        - message: "{{ message }}"  
    - - delay: 20  
    - - service: media_player.volume_set  
      - data:  
        - entity_id: media_player.livingroom  
        - volume_level: 0.2
```

```
#####  
#·Away·#·User·wanted·to·be·reminded·later  
#·Will·be·reminded·whether·this·option·is·selected·or·not  
#####  
  
alias: "Notify Remind Later Away Action"  
id: c722590e-fed7-46b3-bf8d-935259c13fe5  
trigger:  
  -- platform: event  
  -- event_type: mobile_app_notification_action  
  -- event_data:  
    -- action: notify_remind_later_away  
  
action:  
  -- service: script.med_clear_mobile_notification
```

```
#####  
#·Home·#·User·wanted·to·be·reminded·later  
#·Will·be·reminded·whether·this·option·is·selected·or·not  
#####  
  
alias: "Notify Remind Later Home Action"  
id: .567ef195-656d-4058-b309-0078375f4d6e  
trigger:  
  .platform: event  
  .event_type: mobile_app_notification_action  
  .event_data:  
    .action: notify_remind_later_home  
  
action:  
  .service: script.med_clear_mobile_notification
```

```
#####
#If the user has missed medication reminders they will be added to a counter.
#This automation then send a notification at 18:00 every Sunday evening with that amount.
#Caretaker or family member is notified through Telegram.
#####

alias: "Notify Telegram Fail Medication for the Week"
id: 48d74675-4f28-49c0-94a9-a3a809ca8b2f
trigger:
  -- platform: time
  -- at: '18:00'

condition:
  -- condition: time
  -- after: "17:55:00"
  -- before: "18:05:00"
  -- weekday:
  -- -- sun

action:
  -- service: telegram_bot.send_message
  -- data:
  -- -- # Message that is sent to telegram family member or Caregiver.
  -- -- message: The User has missed their medication reminders {{ states('counter.med_failed_week_count') }} times this week.
  -- -- title: Medication reminder.
  -- delay: 5
  -- service: counter.reset
  -- entity_id: counter.med_failed_week_count

mode: single
```

```
#####  
#INPUT-BOOLEANS-added-to-configuration.yaml  
#####  
input_boolean:  
- - reminder_away_from_home:  
- - - name: Away From Home  
- - - initial: off  
- - - icon: mdi:beach
```

```
#####  
#·Set·mode·based·on·whether·home·or·away  
#####  
  
alias: "Set Away Mode for Reminders"  
id: a45ef888-47f3-40da-bdab-3b8113216cc5  
initial_state: true  
trigger:  
  - platform: state  
  - entity_id: person.user_client  
  
action:  
  - service_template: "input_boolean.turn_{{- 'on' if  
    states('person.user_client') == 'away' else 'off' }}"  
    entity_id: input_boolean.med_away_from_home  
  - service_template: "input_boolean.turn_{{- 'off' if  
    states('person.user_client') == 'away' else 'on' }}"  
    entity_id: input_boolean.voice_player_notification
```

```
#####  
#·Turn·on·input·booleans·for·away·or·for·home  
#####  
med_turn_on_reminder_away:  
  ··sequence:  
  ···service:·input_boolean.turn_on  
  ···entity_id:·input_boolean.med_notify_away_activate  
  
med_turn_on_reminder_home:  
  ··sequence:  
  ···service:·input_boolean.turn_on  
  ···entity_id:·input_boolean.med_notify_home_activate
```

```
#####  
#·The·user·leaves·from·the·home·for·more·than·2·hours.  
#·Caretaker·or·family·member·is·notified·through·Telegram.  
#####  
  
alias: "User Away From Home"  
id: 75990ef4-09a3-4eba-b279-12283b88c33a  
trigger:  
  platform: state  
  entity_id: person.user_client  
  to: not_home  
  for:  
    #·Amount·of·time·the·user·needs·to·be·away·from·home.  
    hours: 2  
    minutes: 0  
    seconds: 0  
  
action:  
  service: telegram_bot.send_message  
  data:  
    message: The User has left home and has not returned after being away for 2 hours.  
    title: User is away from home.  
  
mode: restart
```

```
#####  
# Send command depending on whether at home or away  
#####  
med_home_or_away_notification:  
  --sequence:  
    ---service: timer.start  
    ---entity_id: timer.med_timer  
    ---choose:  
      ---# IF -- Away from home  
      ---conditions:  
        ---condition: template  
        ---value_template: "{{ states('input_boolean.med_away_from_home') == 'on' }}"  
        ---sequence:  
          ---service: script.turn_on  
          ---entity_id: script.med_turn_on_reminder_away  
      ---# ELSE -- At home  
      ---default:  
        ---service: script.turn_on  
        ---entity_id: script.med_turn_on_reminder_home
```