

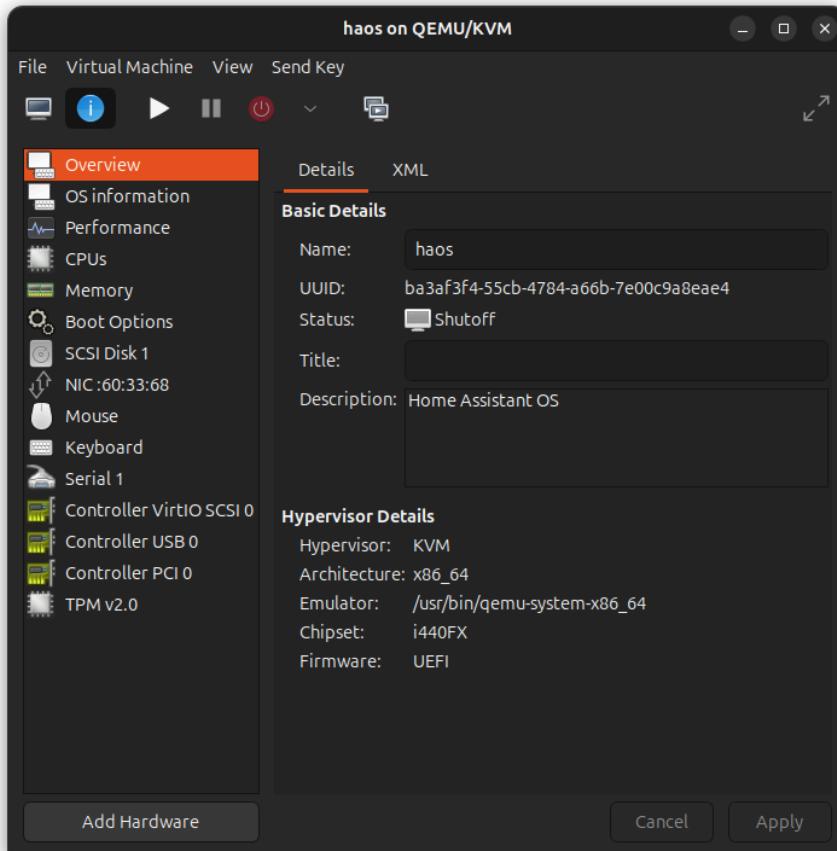
Labor 04

Christopher R. Jäger
304874

1. Installation of Home Assistant

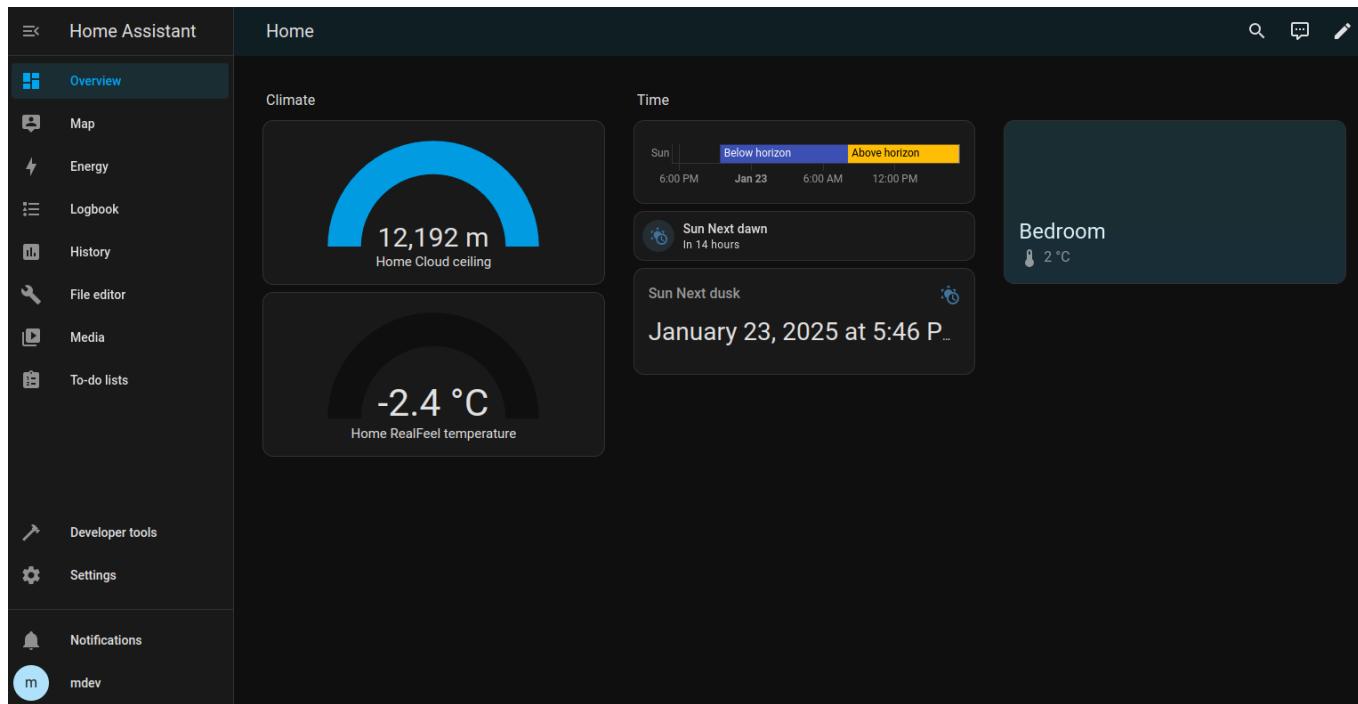
For the installation of Home Assistant, KVM was used as the virtualization software, based on QEMU. The required image was downloaded directly from the official Home Assistant website. To run the image in KVM, a predefined command provided by Home Assistant was used and adjusted to account for the image's file path:

```
virt-install --name haos --description "Home Assistant OS" --os-variant=generic --ram=4096 --vcpus=2 --disk ./Downloads/haos_ova-14.1.qcow2,bus=scsi --controller type=scsi,model=virtio-scsi --import --graphics none --boot uefi
```



Initially, VirtualBox was used as the virtualization software. However, an issue arose: the image could not be started on the same machine. An error message indicated that the image intended

for VirtualBox did not contain a bootable system.



2. Adding Devices/Services and Automating Home Assistant

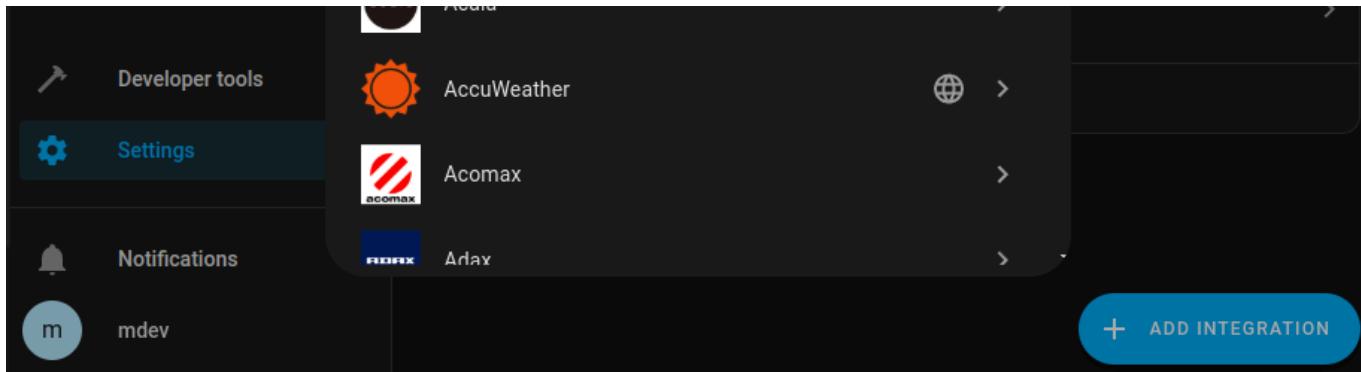
2.1. AccuWeather Integration

To obtain weather data for a specific location, various public APIs can be used. In this case, AccuWeather was chosen. First, an account was created on the AccuWeather platform, followed by the creation of a new app. For this example, the app was named `ubi`. The API key generated for the app was then copied and added to Home Assistant.

The screenshot shows the Home Assistant configuration interface for the 'ub' integration. It includes sections for Service info (AccuWeather), Sensors (Home, Air quality day 0-4, Cloud ceiling), Automations (none), and Scenes (none). The Logbook on the right shows recent weather events like pressure tendency changing to falling and conditions changing from overcast to partly cloudy.

In Home Assistant, the integration process began by navigating to `Settings > Devices & Services` and selecting `+ Add Integration`. From the list of integrations, AccuWeather was

chosen. The previously obtained API key was entered in the form field, and the latitude and longitude values for Konstanz were used.

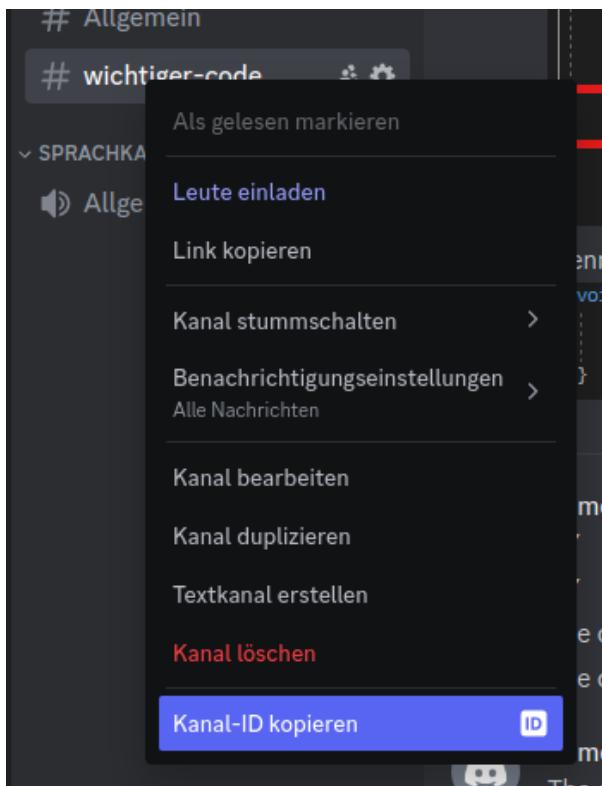


2.2. Notification to Discord

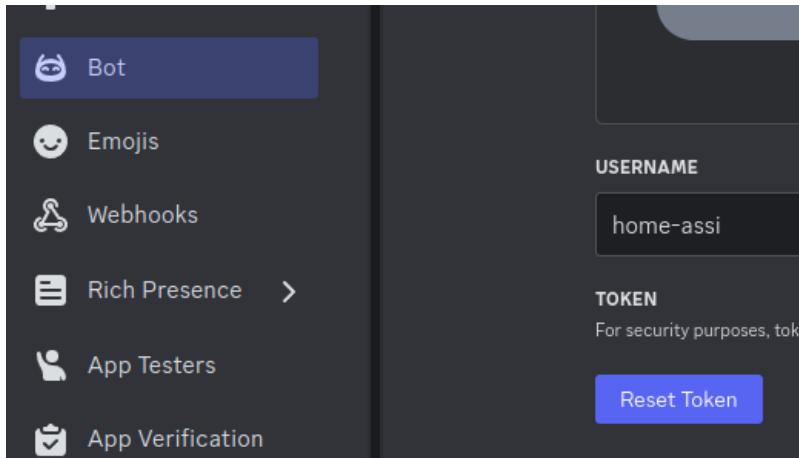
Initially, Telegram was chosen for notifications. However, due to a bug in the Telegram API that prevented message delivery, Discord was used as a workaround to save time.

To enable notifications via Discord, the first step is to create a Discord bot or application in the [Discord Developer Portal](#). Once the application is created, the bot can be added to a specific server where it will send notifications. To do this, navigate to the `OAuth2` tab in the developer portal and use the `OAuth2 URL Generator`. Select the `bot` scope and grant the `Send Messages` permission. This will generate a URL, which must be opened in a browser to install the bot on the desired server. Discord will prompt the user to select the server where the bot should be added.

After the bot is installed on the server, the channel ID where the notifications will be sent needs to be copied. To do this, open Discord, navigate to the desired server, right-click the target channel, and select `Copy Channel ID`.



Next, the bot needs to be integrated with Home Assistant. In the Discord Developer Portal, open the `Bot` tab of the application and click the `Reset Token` button in the `Token` section to generate a new bot token. This token is visible only once, so it should be securely saved. If lost, a new token can be generated.



In Home Assistant, navigate to `Devices & Services` and add a new Discord integration. Enter the bot token in the `API Token` field of the form to complete the integration.

To send messages automatically, a new automation must be created in Home Assistant. Go to the `Automations & Scenes` section in the settings and click `Create Automation`. Select `Create new automation` to configure a custom automation. For this task, the goal is to send a message when the sun rises, when it sets, and at regular intervals.

When

- When the sun rises offset by 00:00:00
- When the sun sets offset by 00:00:00
- Trigger every 5 minutes of every hour

+ ADD TRIGGER

And if (optional)

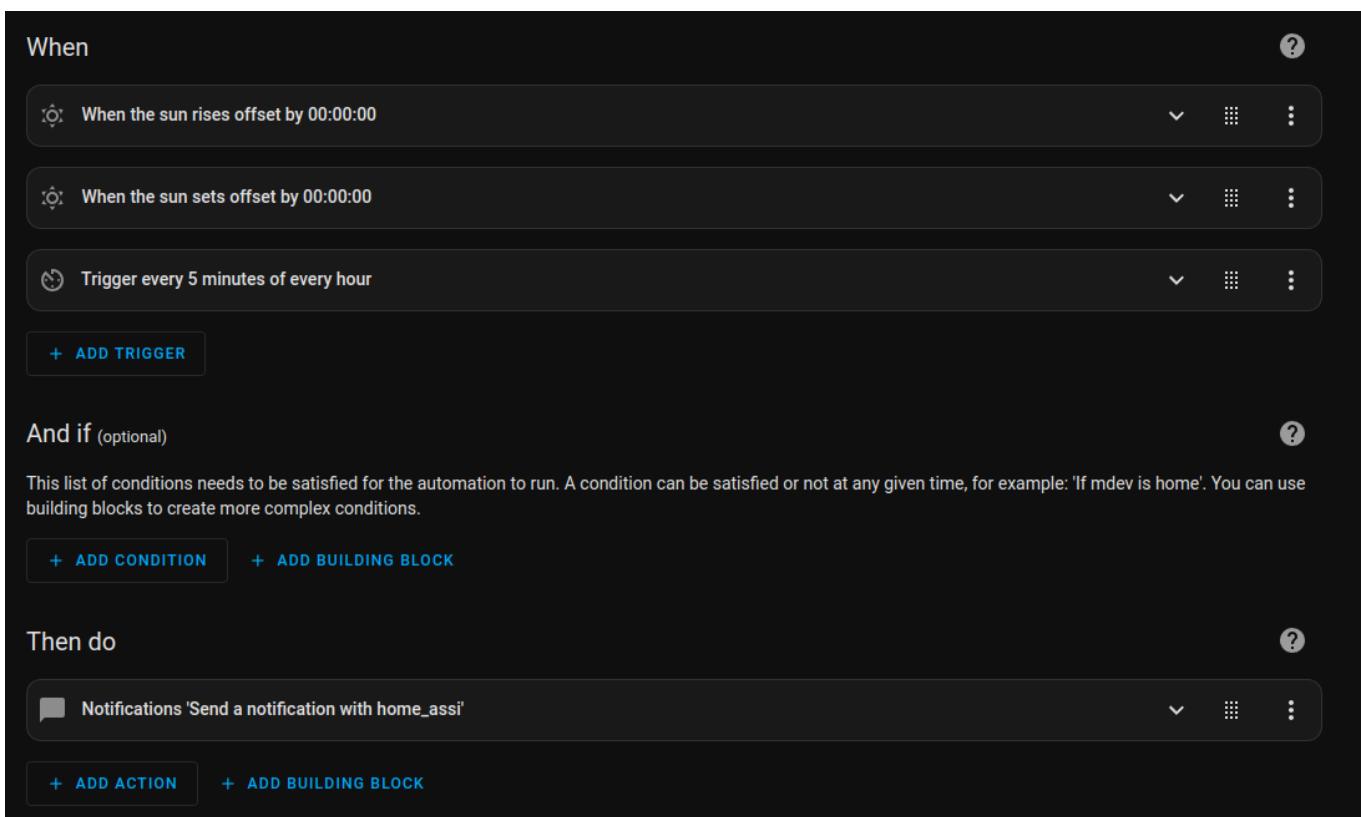
This list of conditions needs to be satisfied for the automation to run. A condition can be satisfied or not at any given time, for example: 'If mdev is home'. You can use building blocks to create more complex conditions.

+ ADD CONDITION + ADD BUILDING BLOCK

Then do

- Notifications 'Send a notification with home_assi'

+ ADD ACTION + ADD BUILDING BLOCK



To define the triggers, add a condition for "When the sun rises" with an offset of 00:00:00 (the offset specifies a delay after sunrise). Similarly, add a trigger for "When the sun sets" and another for a periodic trigger that runs every 5 minutes. When these triggers are activated, the automation will send a message.

In the YAML configuration editor, set the `action` to `notify.home_assi` (the name of the Discord integration). Under the `data` object, specify the `target` property as the previously copied channel ID. The message should include dynamic content, such as the current temperature. For example, the message could be: "The current temperature is ... °C," where the value is retrieved from the `sensor.home_realfeel_temperature` (provided by AccuWeather).

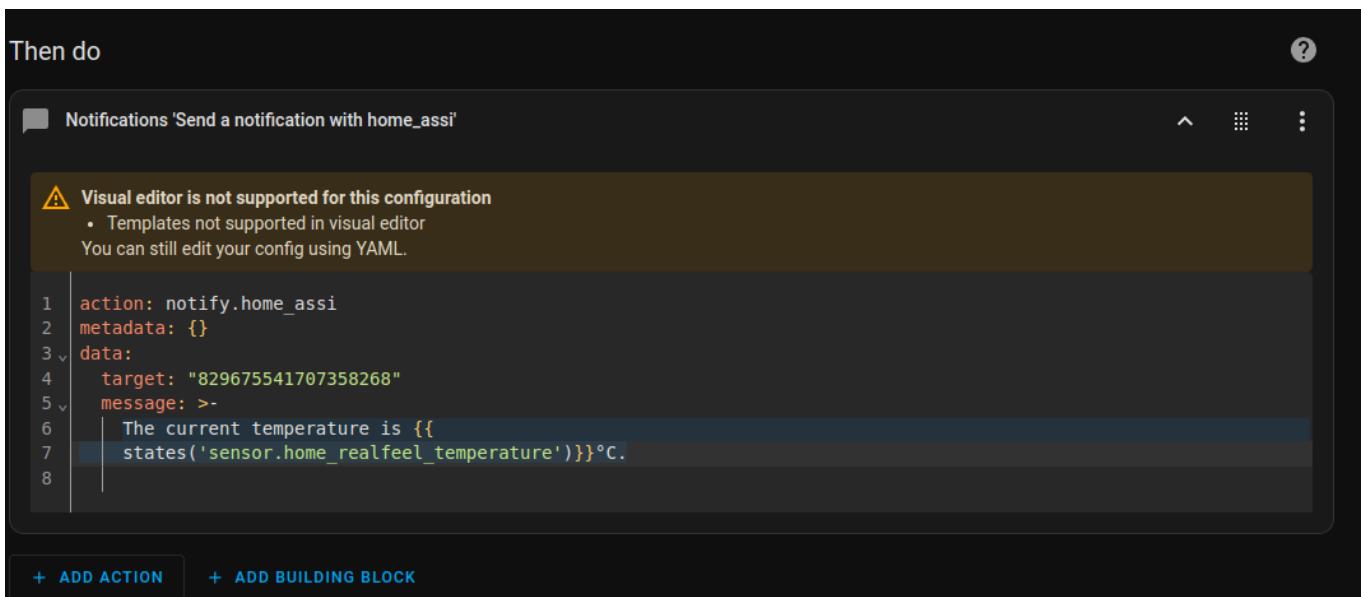
Then do

- Notifications 'Send a notification with home_assi'

⚠ Visual editor is not supported for this configuration
• Templates not supported in visual editor
You can still edit your config using YAML.

```
1 action: notify.home_assi
2 metadata: {}
3 v data:
4   target: "829675541707358268"
5 v   message: >-
6     The current temperature is {{ 
7       states('sensor.home_realfeel_temperature') }}°C.
8 
```

+ ADD ACTION + ADD BUILDING BLOCK



This setup ensures that notifications are sent to Discord automatically based on the defined triggers.



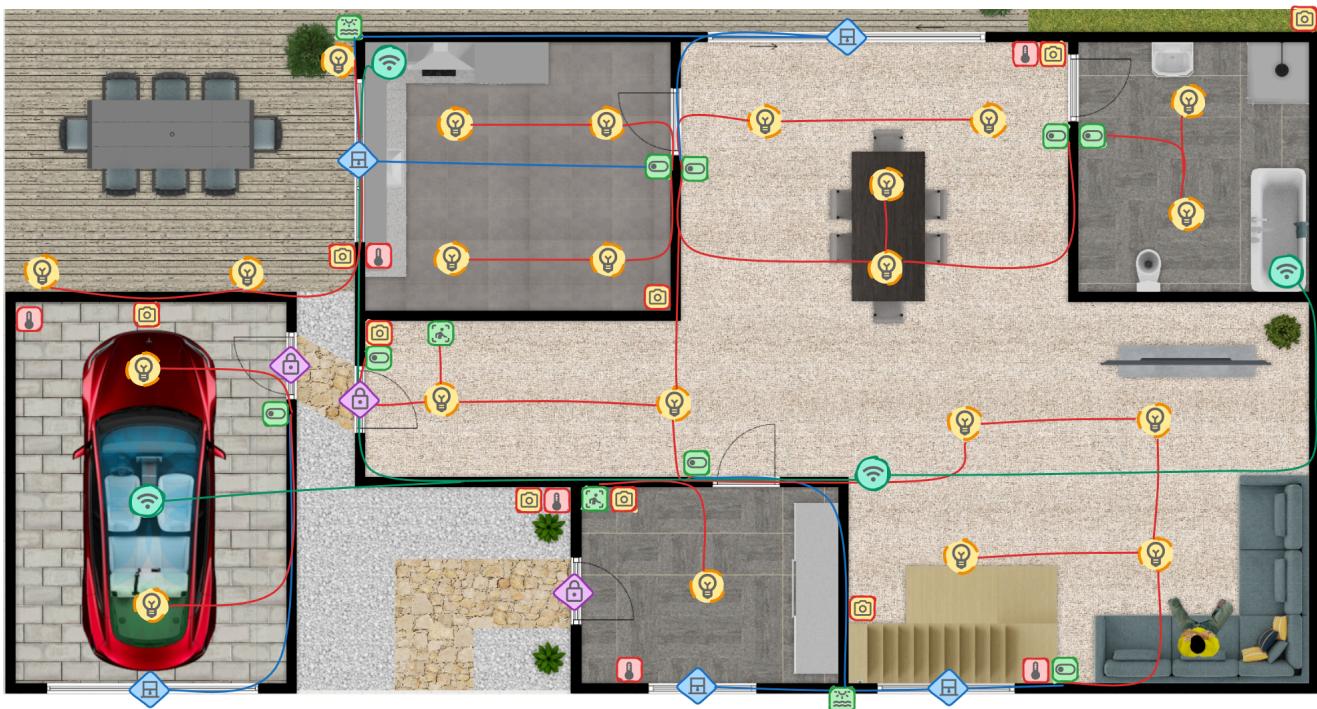
3. Smart Home Planning

3.1. Floor Plan

- 💡 Smart Light - Phillips Hue
- 📶 Access Point - UniFi
- 📷 Camera - UniFi
- 🌡 Temperature Sensor - Ultimate Sensor
- 🔌 Smart Switches - Phillips Hue
- 👤 Presence Sensor - Ultimate Sensor
- 🔒 Smart Lock - Nuki Ultra
- 💻 Roller Shutter - Somfy
- ✓ Ethernet
- ✓ Light Connection
- ✓ Shutter Connection
- ⚡ Wallbox Pulsatrix
- 💡 Light Sensor - Somfy
- 🌐 Central Control Unit - Home Assistant Dashboard
- 💻 Rackstation for Server, NAS and Gateway

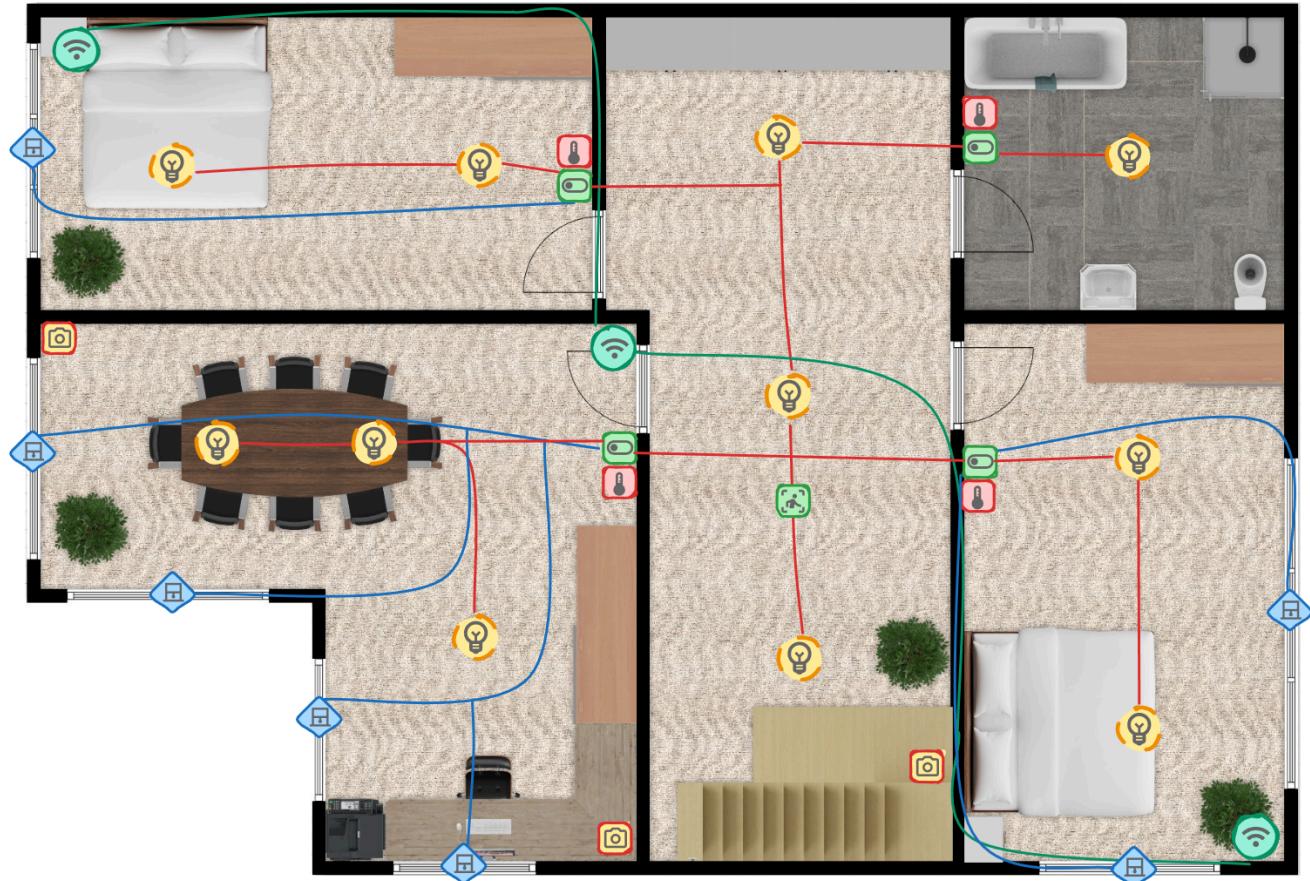
3.1.1. Ground floor

On the Ground floor exists a Garage, Terrace, Kitchen and Bathroom and Living room



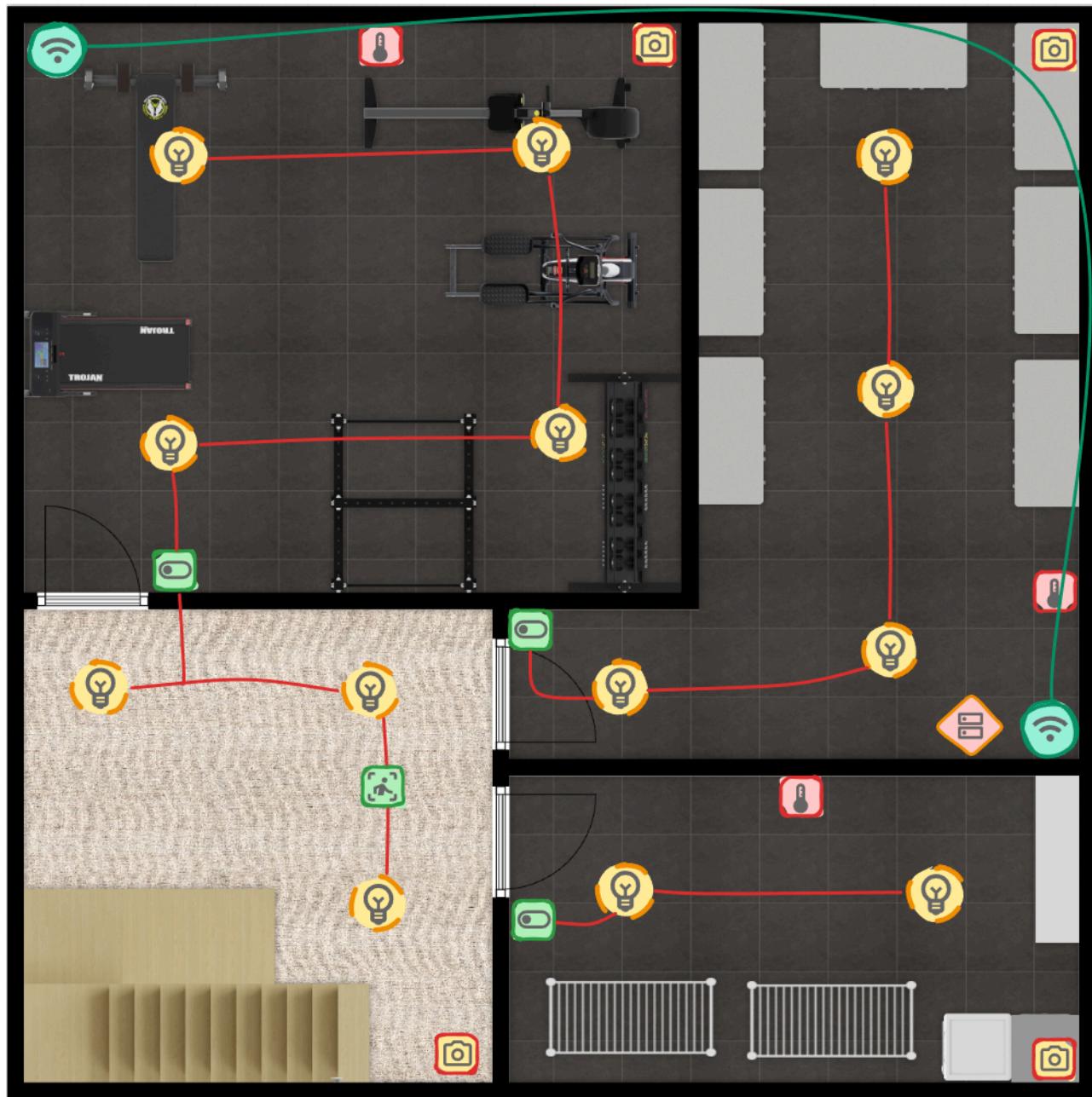
3.1.2. First floor

In the first floor there are two bedrooms, office and bathroom.



3.1.3. Basement

In the basement exists a gym, washroom and storeroom with the home server.



3.2. Used Components

Smart Lighting (e.g., Philips Hue Lights)

- **Aim:** Increasing comfort and energy efficiency
- **Connection:** Wi-Fi, Zigbee
- **Protocol/Interface:** Zigbee (for direct communication with the hub), Wi-Fi (for cloud control and app integration)

- **Power Supply:** Standard electrical power supply (AC 110/220V)
- **Functionality:** Automated lighting control, color adjustment, energy-saving modes, and remote access. Can be controlled via mobile apps or voice assistants (e.g., Alexa, Google Assistant).

Security Cameras (e.g., UniFi Camera System)

- **Aim:** Providing security and surveillance
- **Connection:** Wi-Fi, Ethernet, PoE (Power over Ethernet)
- **Protocol/Interface:** RTSP, Cloud Communication
- **Power Supply:** PoE or separate AC adapter
- **Functionality:** Surveillance, motion detection, real-time alerts, and remote monitoring through mobile apps or web portals.

Smart Locks (e.g., Nuki Smart Lock Ultra)

- **Aim:** Enhancing security and convenience
- **Connection:** Bluetooth, Wi-Fi
- **Protocol/Interface:** Bluetooth, Wi-Fi
- **Power Supply:** Batteries
- **Functionality:** Remote control of door locking, keyless entry, access logs, and integration with home automation systems for automated locking/unlocking based on user proximity.

Motion Sensors (e.g., Ultimate Sensor)

- **Aim:** Enhancing security and energy efficiency
- **Connection:** Zigbee, Z-Wave
- **Protocol/Interface:** Zigbee, Z-Wave
- **Power Supply:** Batteries
- **Functionality:** Detects movement and triggers actions like turning on lights, activating alarms, or notifying the user of unauthorized access.

Smart Roller Shutters (e.g., Somfy Roller Shutter)

- **Aim:** Providing security, comfort, and energy efficiency
- **Connection:** Z-Wave, Wi-Fi, RF (Radio Frequency)
- **Protocol/Interface:** Z-Wave, RTS (Radio Technology Somfy)
- **Power Supply:** Electrical power supply
- **Functionality:** Automated opening/closing of blinds and shutters based on time of day, occupancy, or external conditions (e.g., temperature or sunlight). Enhances privacy and security.

Electric Vehicle (EV) Charger (e.g., Wallbox Pulsatrix)

- **Aim:** Providing energy efficiency and convenience
- **Connection:** Wi-Fi, Ethernet
- **Protocol/Interface:** Wi-Fi, Cloud Communication
- **Power Supply:** AC mains power
- **Functionality:** Smart charging for electric vehicles, energy monitoring, remote control, and scheduling of charging sessions.

Home Server

- **Aim:** Centralized control, automation, and data management
- **Connection:** Ethernet, Wi-Fi
- **Protocol/Interface:** HTTP/HTTPS, MQTT, SMB, NFS
- **Power Supply:** AC mains power
- **Functionality:** The home server acts as the backbone of the home automation system, hosting services such as **Home Assistant**, **MQTT Broker**, and potentially other applications like **Nextcloud** or **Plex**. It allows for automation control, remote access, and data processing. The server ensures all smart devices work in harmony and facilitates the integration of various systems.

Network-Attached Storage (NAS)

- **Aim:** Data storage, backup, and file sharing
- **Connection:** Ethernet (preferably) or Wi-Fi
- **Protocol/Interface:** SMB, NFS, FTP, DLNA
- **Power Supply:** AC mains power
- **Functionality:** A NAS device, such as the **Synology NAS**, provides centralized storage for media files, documents, backups, and more. It can also be used for local file sharing across the home network. Additionally, it can host services like a media server (Plex, Emby), backup systems, or even as a surveillance storage system for security cameras.

UniFi Gateway (e.g., UniFi Dream Machine)

- **Aim:** Network management, security, and connectivity
- **Connection:** Ethernet, Wi-Fi
- **Protocol/Interface:** HTTP/HTTPS, VPN, DNS, DHCP
- **Power Supply:** AC mains power
- **Functionality:** The UniFi Gateway is a central networking device that manages the local network's traffic, security, and internet access. It integrates with other UniFi devices, such as access points and switches, to ensure stable and secure connectivity throughout the

building. The gateway provides features like firewall protection, intrusion detection, and remote management.