

# Home Assistant

My journey with [Home Assistant](#) and the things I've learned which just might help you.

- [Setup](#)
  - [Setup Home Assistant on a Mini PC](#)
  - [Enable HACS - Home Assistant Community Store](#)
  - [Home Assistant Dynamic DNS & SSL](#)
- [Integration Notes](#)
  - [Tesla](#)
  - [Midea AC](#)
  - [TP-Link Smart Home: Tapo & Kasa devices](#)
- [Advanced Setup](#)
  - [433 MHz Devices](#)
  - [Frigate NVR](#)

# Setup

# Setup Home Assistant on a Mini PC

**NOTE:** You will need a dedicated computer to run Home Assistant. There are other options available, but most of them require a new device. It's possible to use VirtualBox or VMware, but there are big limitations with that so I would only recommend it for trying things out. I found it easier to just dive in fully.

After investigating several options, I decided the easiest route to start out with Home Assistant was to purchase a cheap Mini PC and install HAOS on it directly, without a Hypervisor.

## Finding a cheap Mini PC

I browsed <https://lowcostminipcs.com/> which is a site that links to good Mini PC deals on eBay. Almost any will do, but I splurged and paid \$74 for an HP Elitedesk. If you just want the basics and don't want to set anything up, you can get a [Home Assistant Green](#) device, I just knew I could get a lot more computer for a lot less price.

## Installing HAOS on a Mini PC

The [installation instructions](#) Home Assistant provides for a Generic x86-64 are pretty good. If you've ever installed linux from a Live CD before then it's nothing too complicated. A few things to watch out for though:

- HAOS does not provide a live ISO for installing. You must create an Ubuntu Live USB device and then once in the Ubuntu trial environment you download the HAOS image and use Ubuntu to write that image to your disk. A little odd but it works
- You may need to modify your BIOS boot order and disable Secure Boot.
- Your computer probably won't boot after following the directions. There's a step they hide at the end to create the EFI boot entry. If you follow their instructions step-by-step you'll have to boot back into Ubuntu live mode just to run one more command.
- Once installed, you need ethernet on your first boot, but you don't need a monitor on your first boot. HAOS does a great job of launching a web interface automatically and

immediately with everything you need, so don't worry about lugging a monitor around after the initial install.

- I installed HAOS over wifi where my monitors were and then moved it to my router where the ethernet was for the first boot and didn't need to bring a monitor.

# Backups

One of the first things you should do is [enable backups](#) and have them copied off of your device. You'll want to think about where you can send them too. One popular option is an add-on to send them to [Google Drive](#). In my case I have a network share available over Samba/SMB that I can send them to. Connecting to a Samba share no longer requires any add-ons, you can add it directly from **Settings > System > Storage > Add Network Storage**.

In **Settings > System > Backups** select the 3-dot menu in the upper right and select *Change default backup location* to change the backups to use your network share instead of being saved locally. Create a backup and check it. Unfortunately you cannot specify a sub-folder to place the backup in, so if you want it in a folder you may have to do some trickery to do that. For me, I can create a Samba share that goes straight to the desired folder.

Scheduling backups is easy in **Settings > Automation & Scenes > Create Automation**. Set a *Time Pattern* trigger. For action search *Backup* and select *Home Assistant Supervisor: Create a full backup*.

Over time I will need to watch the space needed by the backups as I've heard they can grow quite large. I've also heard that the Visual Studio Code Add-on can cause backups to grow quite large, so avoiding that may be one solution.

# Enable HACS - Home Assistant Community Store

Many devices you may want to control are not supported by the default included Home Assistant install. When searching online for how to connect a device with search terms like "Home Assistant [Your Device]" you will find Github repos with installation instructions. Most of these use HACS to install them, so you'll need to install HACS first.

Don't be [dumb like me](#) and try to install HACS add-ons through the native Home Assistant Add-on page. Install HACS first!

## Installing HACS

Follow the most recent instructions provided by HACS: <https://www.hacs.xyz/docs/use/>

In summary, this will include:

1. Adding the HACS Add-On Store in **Settings > Add-ons** within Home Assistant.
2. Checking the **Settings > System > Logs > Get HACS** for any messages about additional manual steps (I had none)
3. Restarting Home Assistant in **Settings > System** and press the power icon in the upper right.
4. Adding the HACS Service under **Settings > Devices & Services**

Once installed, HACS will appear in the main Home Assistant Side menu.

## Installing HACS Integrations

**WARNING:** HACS addons are generally not as stable as official ones. Many are abandoned & broken already, or may break over time. Go to the integration's Github Repo to report any issues or ask any questions. Also check if they are still actively maintained.

From the HACS option in the side-menu, you can browse and find additional devices/services that are supported or add the repository URL for any you find online not listed.

Some GitHub repos even provide a link that will route you straight to your Home Assistant install and add the integration when clicked.

Once an integration is Downloaded/Installed, you set it up through the usual **Settings > Devices & Services** page just like a native integration.

I found that after first installing HACS and downloading some custom integrations, adding them from the **Devices & Services** page was not working. The integration seemed to just load forever with a "Please wait, starting configuration wizard" message. Home Assistant does not provide any way to back out of this menu, so you may need to just close the app/webpage and reload it to try again. I left it for the night and in the morning had no issues. If you have this issue after any HACS integrations, just give it some time and come back to try again. A restart of Home Assistant did *not* resolve the issue as the restart was blocked by whatever process is being slow behind the scenes.

# Home Assistant Dynamic DNS & SSL

There are numerous reasons to get an SSL certificate for your Home Assistant environment, particularly if you plan on allowing any external access.

Dynamic DNS is also extremely useful for external access so you can have a consistent domain name to use to access your HA environment even when your local ISP changes your IP address.

## Dynamic DNS

For most people, I would recommend using the existing Duck DNS Add-On found in **Settings > Add-Ons**. In my case, I already have a separate Dynamic DNS setup for my home network. Rather than getting a SSL certificate for that DDNS provider I instead used an existing domain I own like this:

1. Setup a *CNAME* record from *newsubdomain.mywebsite.com* to *mysubdomain.ddnsprovider.com*

## SSL - Lets Encrypt

Add the Lets Encrypt Add-on found in **Settings > Add-Ons**.

Fill out the Configuration information using the DNS challenge. You can use the http challenge if you expose your install's port to the web, but I didn't want to do that yet so DNS was a better option. You will need to read the add-ons documentation in details to get your configuration correct. Depending on your DNS provider you need to enter different settings, and I found the Documentation to be incorrect in places.

The short version is:

- Fill out your domain and email address
  - In my case, the domain would be: *newsubdomain.mywebsite.com*
- Find your DNS provider in the [Example Configurations section of the Add-on Docs](#).

- Do not use the "DNS providers" section dropdown for this, some of the settings are incorrect/missing.
- Copy the settings under the "dns" section and paste those into your HA add-on configuration page. Example:

▼ easyDNS

easyDNS REST API access must be requested and granted [signup.php](#) after logging into your account.

```
email: your.email@example.com
domains:
  - your.domain.tld
certfile: fullchain.pem
keyfile: privkey.pem
challenge: dns
dns:
  provider: dns-easydns
  easydns_token: 0123456789abcdef
  easydns_key: ****
  easydns_endpoint: https://rest.easydns.net
```

- 
- Search for how to get any required credentials for your specific DNS provider. Fill in those fields.
- Save the **Configuration** page.
- Go to the **Info** page of the Add-on. and click **Start**.
  - The Add-on will begin trying to get your SSL certificate. Switch to the **Log** tab to see if it was successful or if there were any errors. It may take several minutes, there's a **refresh** button at the bottom to reload get any new logs.

```
s6-rc: info: service legacy-services successfully started
[10:34:13] INFO: Selected DNS Provider: 
[10:34:13] INFO: Use propagation seconds: 60
[10:34:14] INFO: Detecting existing certificate type for .com
Saving debug log to /var/log/letsencrypt/letsencrypt.log
[10:34:15] INFO: No certificate found - using 'ecdsa' key type.
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Account registered.
Requesting a certificate for 
Waiting 120 seconds for DNS changes to propagate

Successfully received certificate.
```



# Integration Notes

Helpful notes about configuring and using various integrations.

# Tesla

You'll find several Tesla integrations available in the **Settings > Devices & Services > Add Integration** page. Here's a brief comparison to help you pick them:

- **Tesla Fleet** - Integrates with your Tesla Vehicles. Also supports Tesla Solar/Powerwall. All communication done through Tesla's cloud API. Provides only high-level solar sensors
- **Tesla Powerwall** - Integrates with Tesla Solar/Powerwall. All communication is done locally with your Tesla Backup Gateway. Provides much more fine-grain solar and battery sensors.
- **Tesla Wall Connector** - Integrates with Tesla Wall Connector / Charger. All communication is done locally with your devices.

If you have Tesla Solar, I recommend you use the Powerwall Integration so that your communication is done locally and you get all the extra detail. You can install all the Tesla integrations at the same time if you want to connect your Solar, Cars, and chargers.

## Configure Energy Dashboard

Dashboard Configuration is under **Settings > Dashboards > Energy**

The following lists which sensor you should use for the various Dashboard energy settings.

### For Tesla Powerwall/Solar

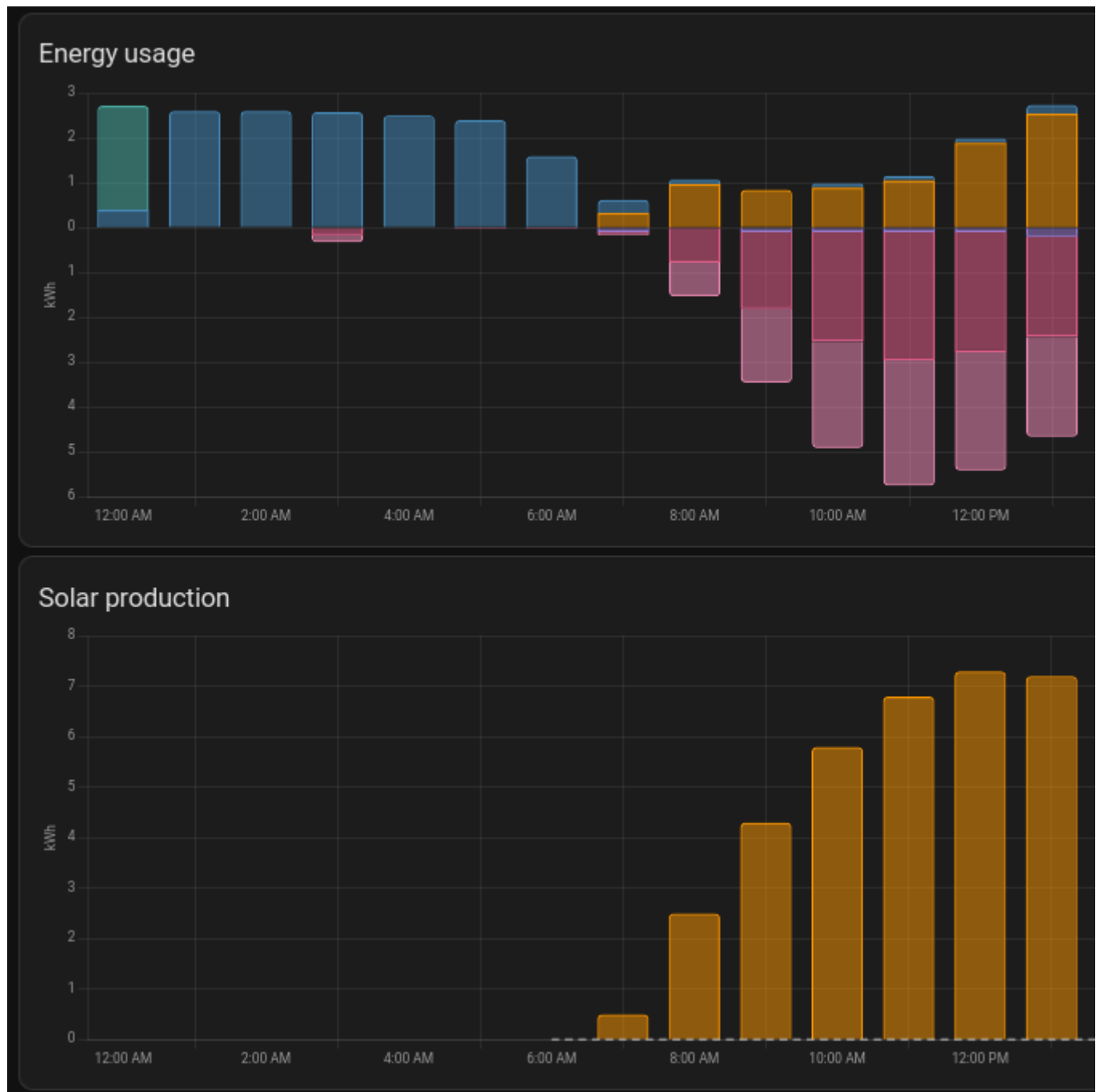
- **Electric Grid**
  - Grid consumption: My Home Site import (sensor.my\_home\_site\_import)
  - Return to grid: My Home Site export (sensor.my\_home\_site\_export)
- **Solar Panels**
  - Solar production: My Home Solar export (sensor.my\_home\_solar\_export)
- **Home battery storage**
  - Battery systems
    - Energy going in to the battery: My Home Battery import(sensor.my\_home\_battery\_import)
    - Energy coming out of the battery: My Home Battery export (sensor.my\_home\_battery\_export)
  - **Note:** If you have multiple powerwalls you can either configure them as a single battery (shown above) or add them as separate batteries. Don't add both the summary and the individual battery or you'll be double counting. Try out both ways and see which you prefer on the dashboard.

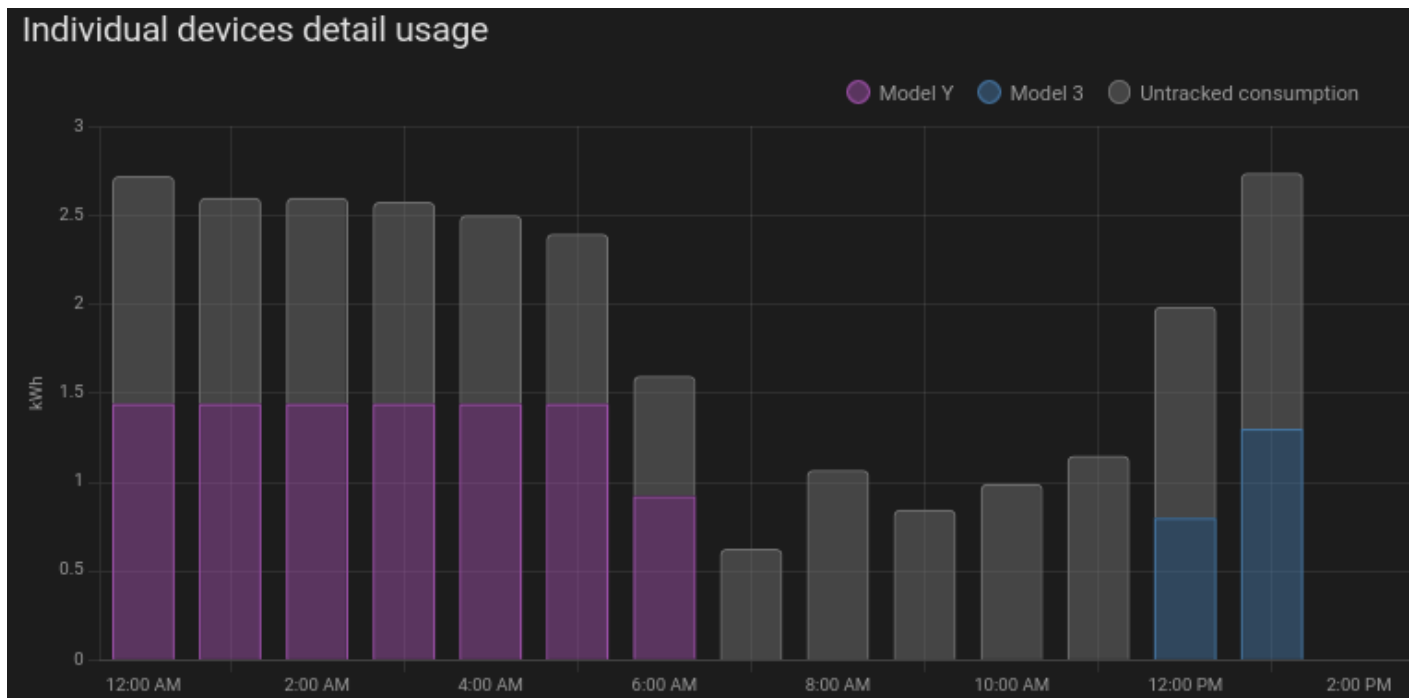
# For Tesla Vehicles

- **Individual Devices**

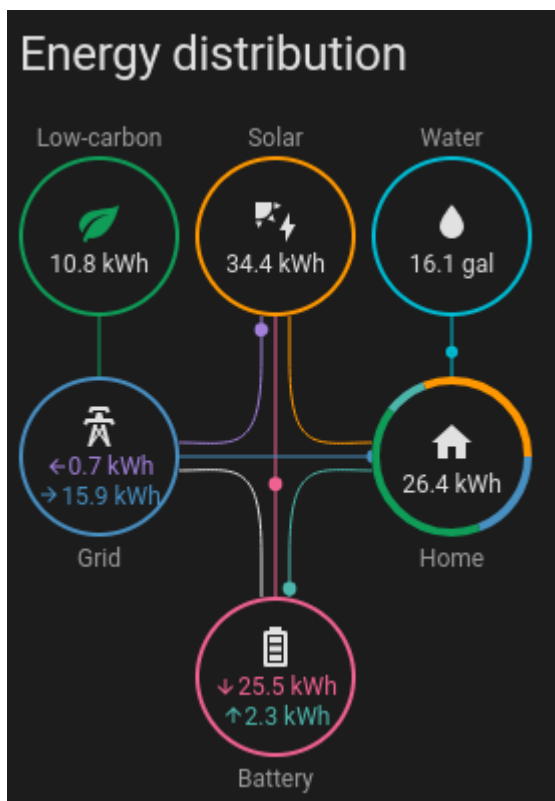
- Devices: [Car Name] Charge energy added (sensor.charge\_energy\_added)

## Energy Dashboard Screenshot Examples





**Note:** The Energy distribution diagram shows **total daily** power (hence the kWh unit) and not instantaneous usage like the Tesla App does.



To get the "Low-carbon" indication, you also need to configure your "Grid carbon footprint" setting under "Electricity grid" in the Energy dashboard settings. This requires setting up an API key with the free "Electricity Maps" service available within Home Assistant.

# Midea AC

The default Midea integration does not support most common Midea devices. To support common Midea M-Smart devices like Air conditioners, Fans, Water heaters, Washers, etc you will need to install a HACS addon (See [Enable HACS](#) if you don't have/know about that first).

Unfortunately the main "Midea AC Lan" add-on is broken and appears abandoned by the initial developer (a common problems with HACS addons). However there are a few forks which have continued development and fixed issues. The most well supported fork I have found is:

[https://github.com/wuwentao/midea\\_ac\\_lan](https://github.com/wuwentao/midea_ac_lan)

## To install

1. Go to the HACS menu in Home Assistant
2. Select "Custom Repositories" from the upper-right 3-dot menu
3. For the Repository field enter: [https://github.com/wuwentao/midea\\_ac\\_lan](https://github.com/wuwentao/midea_ac_lan)
4. For the Type field select: **Integration**
5. Click Add and then Download.
6. After restarting Home Assistant you should see it in the usual **Settings > Devices & services > Add Integration** menu

**Note:** If install fails, make sure you did not select the other "Midea AC Lan" in HACS. It has more downloads listed, but is also broken.

# TP-Link Smart Home: Tapo & Kasa devices

This section is a work in progress, it will cover guides for setting up and connecting to Tapo and Kasa devices.

One particular area it will cover (once I figure it out) is how to get devices reliably connected if they are on a separate network segment, such as a network segment specifically to keep IoT devices apart from the rest of your network.

Currently I am not able to get most of my Kasa or Tapo devices to connect when they are on a separate IoT specific network segment.

# Advanced Setup

Covering more advanced setup topics as I learn them

# 433 MHz Devices

## Why 433 MHz?

While Wifi, Zigbee, Z-wave, and Matter/Thread are great, sometimes you just need a simple sensor that does one thing really cheaply. 433MHz devices (and others like 915MHz and 868MHz) are one-way sensors that can talk to Home Assistant to report things like temperature, water, moisture, open/close and much more.

### The advantages of these devices are:

- They are cheap
- They are everywhere
- They have extremely long battery life
- They have decent range
- You might already have some
- The communication protocols are simple and reliable, no device pairing needed

### The downsides are:

- You need a special receiver, called an SDR (Software Defined Radio)
- Sensors can *sometimes* require more configuration
- It can be difficult to determine which devices are supported
- Not everything listed as 433MHz can be used
- Setting up your SDR receiver takes more steps, but this guide will show you how

And as either a downside or an upside depending on your view is that you might learn a lot about your neighbors.

## Equipment

### SDR receiver

The most important thing you will need is a SDR receiver. This is a USB device with an antenna which can receive radio signals and pass them to your HA device.

I purchased the [Nooelec NESDR Mini 2+](#) as an entry level device for under \$40, but there are a range of devices of all prices out there. Just do a quick search to ensure it works with Home Assistant. If the Mini 2+ isn't on sale, the regular



[NESDR Mini 2](#) is great too. Any device using a Realtek chip should work. Other companies make chips, but these are very affordable ones. You'll often see the term `rtl` referenced, which stands for *Realtek Limited* (Examples: rtl-srd, rtl-433).

## Sensors

You're going to want something to sense. Here are some good suggestions to start with, but as you get comfortable search around for more device types, as you can do so much with SDR!

- **Temperature/Humidity:**

- [AcuRite](#)
  - Get Sensor #2 without ABC since we don't need channel selection and it's cheaper
- [ecowitt](#)

- **Water/Leak:**

- [Govee](#)
  - I have not yet verified these actually use 433, they may have changed to LoRa

- **Motion Detection:**

- Still investigating

- **Soil Moisture:**

- [Springfield 91746](#)<sup>†</sup>
- [OPUS XT300](#)<sup>†</sup>

Remember how I mentioned learning about your neighbors? Chances are that you and your neighbors already have some devices transmitting on 433MHz. Before I ever turned on my first sensor, my alarm system door sensors and my neighbors soil sensors both popped up in Home Assistant! Car TPMS (Tire Pressure Monitoring Systems) also show up, but because you likely don't want to have a dashboard filled with every car that drives by these are disabled by the default config below.

† These devices I have not tested myself, but my neighbors must be using them because I've picked up their signal and have been able to read them.

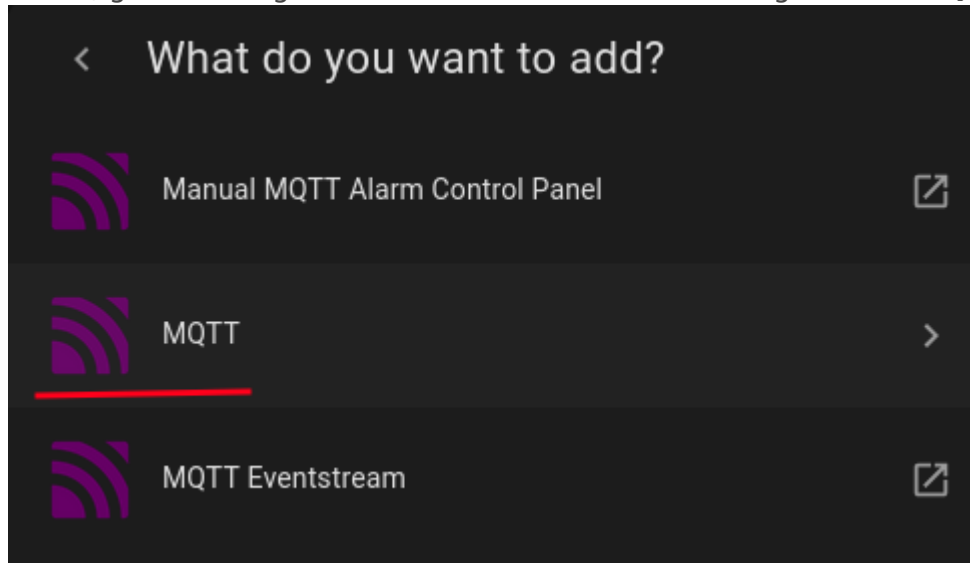
## Setting Up Your SDR Receiver

Your receiver has arrived, let's set it up! If your sensor has arrived too, great! If it hasn't let's set the receiver up anyway and see what we find.

**Note:** If you want to play with the SDR receiver on your linux desktop/laptop first, see my guide here: <https://www.wswapps.com/books/debian-ubuntu-pop-os/page/install-configure->

## Add-on Installation

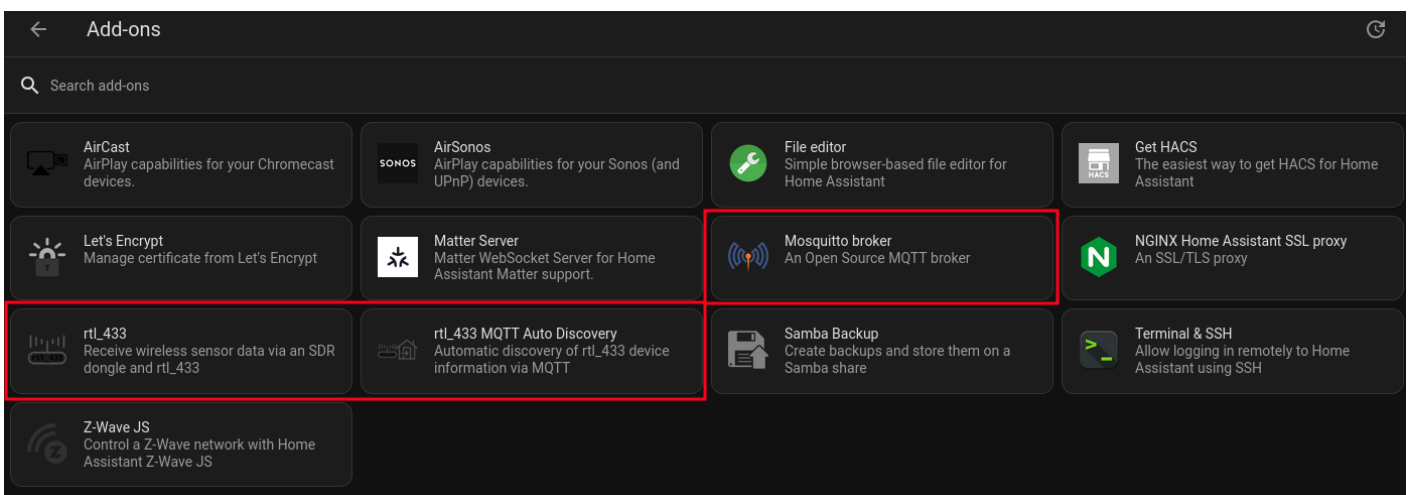
- From HA, go to Settings > Devices & Services > Add Integration > **MQTT**



- All your future devices will show up under this MQTT service
- Next, go to Settings > Add-ons > Add-on Store. Select and install: **Mosquito broker**
- Start Mosquito Broker
  - No configuration is needed
  - This add-on is necessary to take in the detected sensors/messages and publish them to Home Assistant as devices and sensors
- Also from the Add-on store, select the 3-Dot Menu > Repositories and Add the following:
 

[https://github.com/pbkhrv/rtl\\_433-hass-addons](https://github.com/pbkhrv/rtl_433-hass-addons)

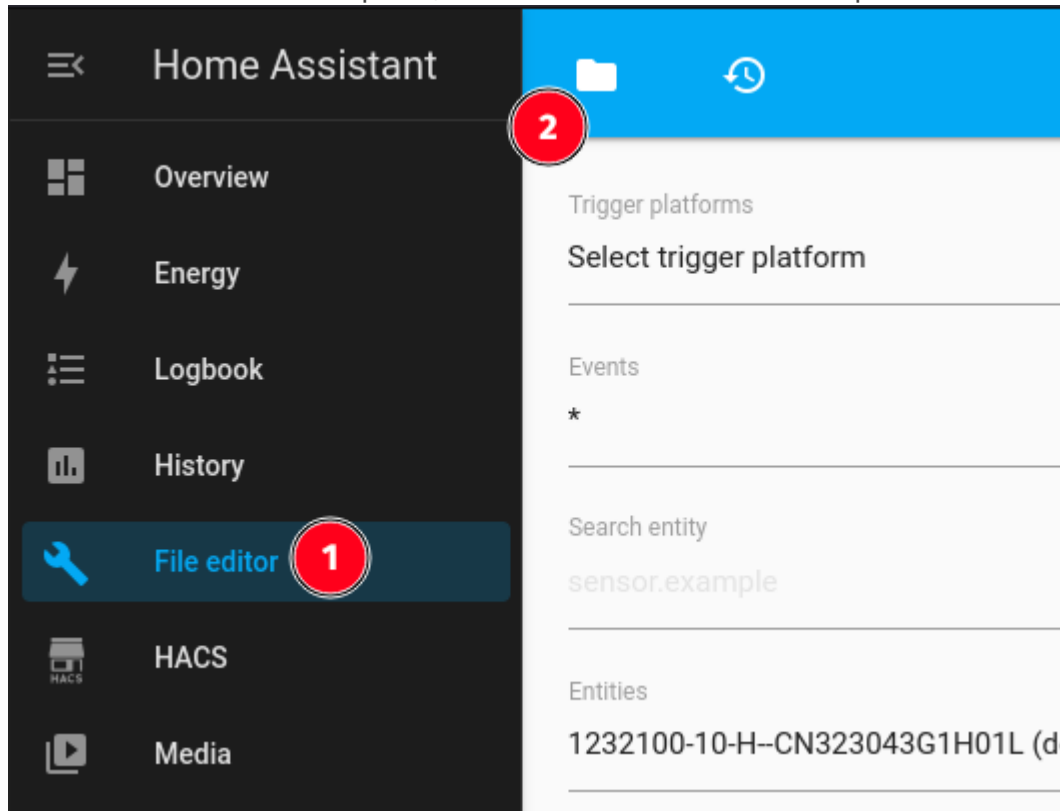
  - This new repository contains the repos we need to use the SDR receiver
- Install both of the following Add-ons now and **start** them:
  - rtl\_433**
  - rtl\_433 MQTT Auto Discovery**
    - You can remove/disable this later if you prefer, but it will make getting started a lot easier. This add-on will auto-detect nearby devices and add them to HA MQTT automatically



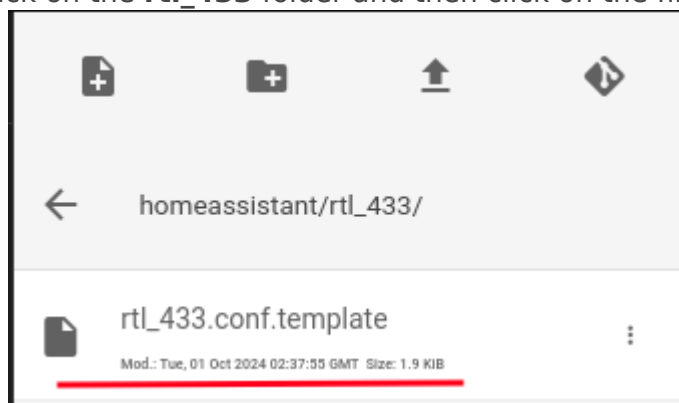
# Add-On Configuration

We're almost done. We now need to create a configuration file for rtl\_433. To do that you need a way to upload or edit files in Home Assistant. If you have a favorite way, use that. For this demo I will use the **File Editor** which you can install from the Add-on section. make sure to enable "Show in sidebar" for the File Editor after installing and starting it.

- From the File Editor menu option, click the folder icon at the top



- Click on the **rtl\_433** folder and then click on the file named: **rtl\_433.conf.template**



- The default config file will be displayed. We will just make a few changes and I'll post the full file below in case that's easier:
  - Remove the `#` before `output kv`
    - This will enable you to see pretty logs on the rtl\_433 Add-on of anything it detects.
  - Add the following new lines (anywhere, but I recommend just after the output kv line):
    - `frequency 433.92M`
    - `convert si`

- Those lines set the frequency to 433Mhz devices and ensure that measurement units are standardized. Optional but recommended.
- The rest of the file can remain as-is. It configures passing the data to MQTT and disables any Tire Pressure sensors that are detected.
- Click the Save icon at the top to save your changes.
- The full file at the end should look like this:

### rtl\_433.conf.template

```
# This is an empty template for configuring rtl_433. mqtt information will be
# automatically added. Create multiple files ending in '.conf.template' to
# manage multiple rtl_433 radios, being sure to set the 'device' setting. The
# device must be set before mqtt output lines.
# https://github.com/merbanan/rtl_433/blob/master/conf/rtl_433.example.conf

output mqtt://${host}:${port},user=${username},pass=${password},retain=${retain}
report_meta time:iso:usec:tz

# To keep the same topics when switching between the normal and edge versions,
# use this output line instead.
# output
mqtt://${host}:${port},user=${username},pass=${password},retain=${retain},devices=rtl_433/9b13b3f4-
-rtl433/devices[/type][/model][/subtype][/channel][/id],events=rtl_433/9b13b3f4-
rtl433/events,states=rtl_433/9b13b3f4-rtl433/states

# Uncomment the following line to also enable the default "table" output to the
# addon logs.
output kv

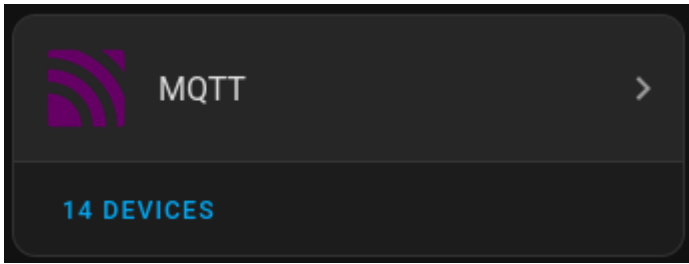
frequency 433.92M
convert si

# Disable TPMS sensors by default. These can cause an overwhelming number of
# devices and entities to show up in Home Assistant.
# This list is generated by running:
# rtl_433 -R help 2>&1 | grep -i tpms | sed '.*\[(\d+)\].*' 'protocol -/run.sh'
# [59] Steelmate TPMS
# [60] Schrader TPMS
# [82] Citroen TPMS
```

```
# [88] Toyota TPMS
# [89] Ford TPMS
# [90] Renault TPMS
# [95] Schrader TPMS EG53MA4, PA66GF35
# [110] PMV-107J (Toyota) TPMS
# [123]* Jansite TPMS Model TY02S
# [140] Elantra2012 TPMS
# [156] Abarth 124 Spider TPMS
# [168] Schrader TPMS SMD3MA4 (Subaru)
# [180] Jansite TPMS Model Solar
# [186] Hyundai TPMS (VDO)
# [201] Unbranded SolarTPMS for trucks
# [203] Porsche Boxster/Cayman TPMS
protocol -59
protocol -60
protocol -82
protocol -88
protocol -89
protocol -90
protocol -95
protocol -110
protocol -123
protocol -140
protocol -156
protocol -168
protocol -180
protocol -186
protocol -201
protocol -203
```

- As you learn more, you can update this config file. [This example config file](#) shows lots of good options that can be configured
- Later on we'll edit this config file to listen to more than one frequency in case you have devices on other frequencies.
- Go back to Settings > Add-ons, select **rtl\_433** and **restart** it.
- Check the Log tab and you should see your SDR receiver detected. Let it run for a few minutes and refresh the logs to see if it detects anything. It may not if you aren't nearby any sensors.

This setup process will automatically detect new sensors and add them to the MQTT integration within HA. There will not be a notification of a new device, they will just show up and you can Disable them if you don't want them. If you find too many devices are showing up, you can disable the **rtl\_433 MQTT Auto Discovery** Add-on. Down below we'll discuss adding a sensor manually if needed.



## Configure a Sensor

If you have the **rtl\_433 MQTT Auto Discovery** Add-on enabled, you don't need to do anything. Any commonly recognized devices will show up automatically in the MQTT integration for you to begin using. If you aren't using Auto Discovery or your device is not one that can be auto discovered, it can be manually configured.

**TODO:** documentation on manually configuring a sensor

## Monitor Multiple Frequencies

To monitor multiple frequencies, you need to update your **rtl\_433.conf.template** file.

- Add the following two lines, then Save the file:

- `hop_interval 30`
  - `frequency 915M`

- Example:

- ```
16 output kv
17
18 hop_interval 30
19 frequency 433.92M
20 frequency 915M
21 convert si
22
```

- Restart the rtl\_433 Add-on.

The `hop_interval` defines how many seconds each frequency will be scanned, before hopping to the next. If you want one frequency scanned more than another, you can list the same frequency multiple times.

The downside of this is that there is a period of time where you are not scanning each frequency and can miss messages. To scan each frequency without missing anything you would need to get

an additional SDR receiver per frequency and configure multiple config files. This guide doesn't cover that setup (yet).

## More Info

Here is a collection of resources and guides I used to help me get started with SDR and 433Mhz in Home Assistant:

- <https://community.home-assistant.io/t/home-assistant-add-on-rtl-433-with-mqtt-auto-discovery/260665>
- [https://static.xtremeownage.com/blog/2021/433mhz-automation/#installing-rtl\\_433](https://static.xtremeownage.com/blog/2021/433mhz-automation/#installing-rtl_433)
- [https://www.reddit.com/r/homeassistant/comments/10pkerb/i\\_have\\_no\\_idea\\_how\\_to\\_use\\_rtl\\_433/](https://www.reddit.com/r/homeassistant/comments/10pkerb/i_have_no_idea_how_to_use_rtl_433/)
- [https://www.youtube.com/watch?v=\\_COwsvkxyFA](https://www.youtube.com/watch?v=_COwsvkxyFA)

# Frigate NVR

The goal with a Network Video Recorder (NVR) is to free yourself from cloud subscriptions for security cameras. You can (and probably should) run a standalone NVR, but in this guide we'll setup Frigate NVR as a Home Assistant Add-On.

## Equipment

To run an NVR alongside Home Assistant you will likely need more CPU power than a Raspberry Pi provides, but most mini PCs should have enough. You can always try it out and find out.

If you plan to use any detection features at all (object tracking, notifications for specific objects) then you will need some sort of hardware acceleration. The [Google Coral USB Accelerator](#) device is recommended and is what we'll cover in this guide.

The [Frigate recommended hardware guide](#) has more details on both hardware acceleration and CPU recommendations.

## Cameras

See list of cameras I've tested at the bottom

You will need cameras that support RTSP and optionally ONVIF. ONVIF is needed for PTC cameras if you want Frigate to be able to control their movement. You also ideally want cameras that can output 2 streams, 1 high and 1 low resolution. High resolution will be used for viewing and the lower resolution for detection tasks.

In this guide I will be using a [Tapo C210](#) camera which can sometimes be found on sale under \$20 and supports Pan & Tilt as well as 2K resolution. Tapo cameras support RTSP and ONVIF as well as 2 different quality streams. Unfortunately they also require the Tapo app to perform the initial setup. I will be searching for other cameras that can be used entirely offline in the future and update this guide.

## Setup Cameras

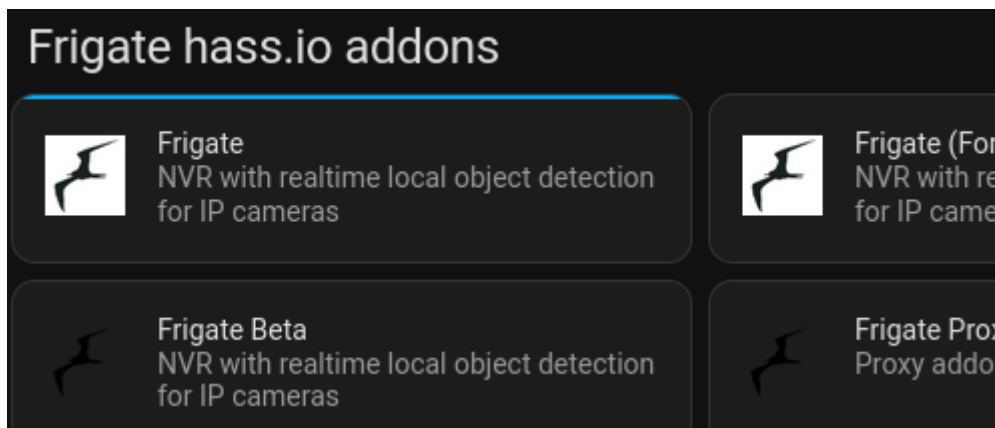


The first step is to setup your cameras and enable RTSP. This will vary based on the brand.

- Be sure to set a secure random username and password for your cameras when enabling RTSP
- In your router, reserve the camera's IP address so that it won't change on you. **Do not** expose the camera to the internet.
- Test your camera's RTSP stream in something like VLC. The exact string will differ based on the brand, but common examples are:
  - Tapo: `rtsp://username:password192.168.0.101:554/stream1`
    - Using `stream2` will load the low quality 720p stream.
- Make sure your RTSP stream is working before moving forward

## Install Frigate

- In HA go to Settings > Add-Ons > Add-On Store
- In the upper right click the 3 dots, select Repositories.
  - Enter `https://github.com/blakeblackshear/frigate-hass-addons` and click **Add**
- Now under the **Frigate hass.io addons** section, select **Frigate** and click **Install**

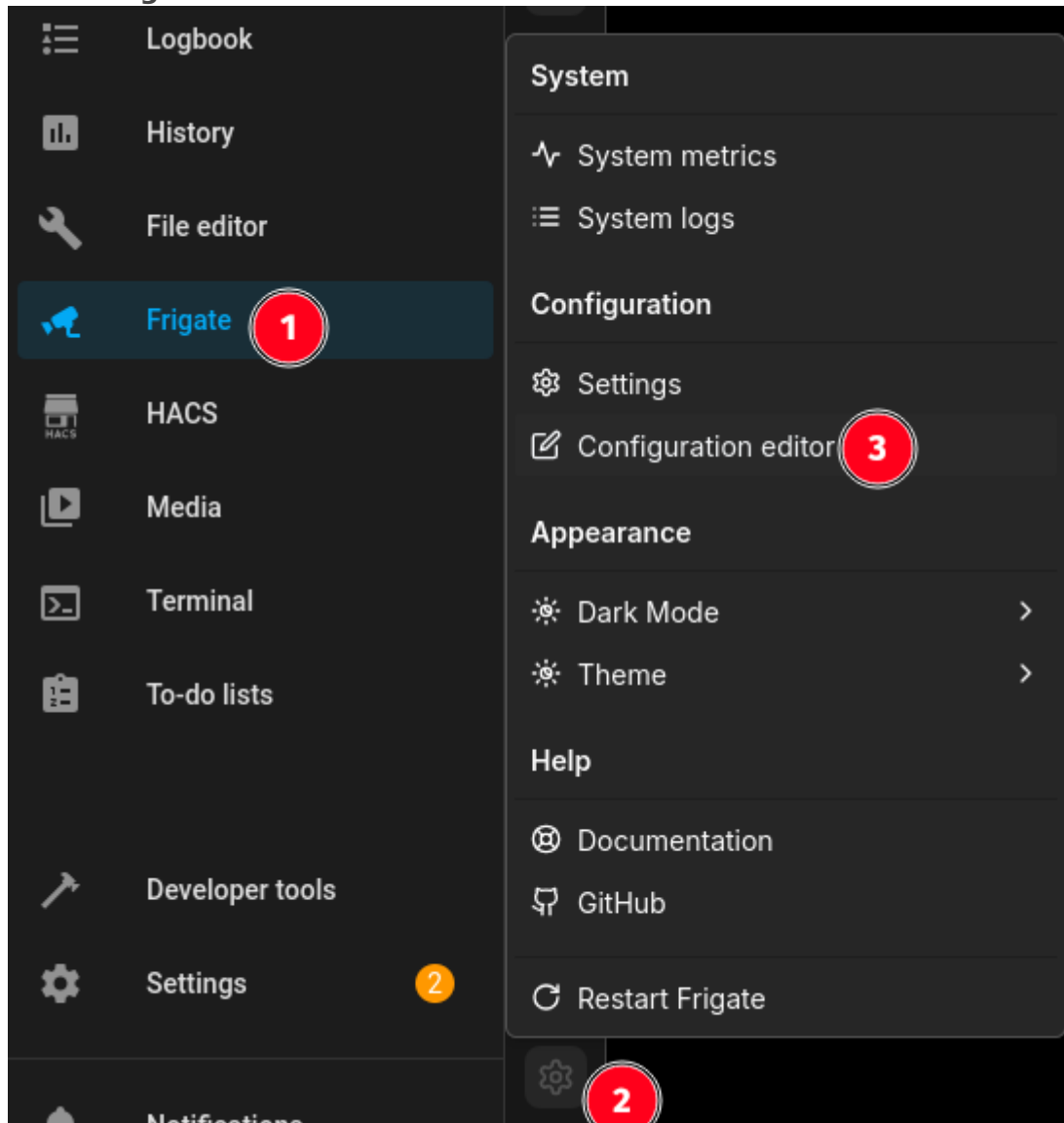


- Frigate (Full Access) is another option that can be used if the regular Frigate is not able to properly connect to your devices. It is not recommended unless absolutely needed since it has unrestricted access to your computer. Install the regular Frigate for now and you can always install Full Access later without requiring any additional configuration. Unfortunately they are not clear about exactly what reasons the Full Access version may be needed for.
- Enable the **Start on boot**, **Watchdog**, and **Show in sidebar** options.
- Click Start to start Frigate

## Add-On Configuration

We now need to update the configuration file for Frigate to tell it about any cameras. To do that you need a way to upload or edit files in Home Assistant. If you have a favorite way, use that. For this demo I will use the File Editor built into Frigate itself.

- From the Frigate menu option, click the gear icon at the bottom of the page and then select **Configuration editor**



- The default Frigate config file will be displayed in YAML format. We need to add our first camera to it.
- Under the **cameras:** section lets add our camera:

### Expand to see frigate.yaml

Keep any code above and below the cameras section that is already in the config file

```
cameras:
  C210: # <----- Name your camera
    enabled: true
    ffmpeg:
      inputs:
        - path: rtsp://username:password@192.168.0.101:554/stream1 # The High Quality stream you want to
          use for recording
```

roles:

- record

- path: rtsp://username:password@192.168.0.101:554/stream2 # The Lower Quality stream you want to use for detection

roles:

- detect

detect:

enabled: false # Disable until you have a working camera feed and hardware acceleration

width: 1280 # The resolution of the detection camera feed.

height: 720

fps: 5 # This can be raised later if we have the processing power

record:

enabled: false

retain:

days: 0 # The number of days a recording will be kept for after a motion is detected

mode: motion

events:

retain:

default: 30 # The event data will be kept for 1 day. After this period, the event data will be automatically deleted.

mode: motion

# Leave out this section if camera is not PTZ or does not support ONVIF

onvif:

host: 192.168.0.101

port: 2020 # Port may differ for your camera brand!

user: username

password: password

autotracking:

enabled: false # Disable until you have hardware acceleration

calibrate\_on\_startup: true

zooming: disabled # Can enable if camera supports Zoom

track:

- person

objects:

track:

- person

- car

- Click the **Save & Restart** button



- If Frigate fails to restart due to an error in your configuration file, in HA go to: **Settings > Add-Ons > Frigate > Log** to see the error message. Be very careful about your file formatting and spacing. Indentation and whitespaces matter in YML files.
- Going back to **Frigate** from the HA side menu, you should now see your camera feed. Clicking the camera feed opens it in full view and If you enabled ONVIF will also show your PTZ controls



Additional Setup and usage instructions are coming soon. In the meantime go ahead and setup more cameras and play around with Frigate. You can try enabling detection in your Frigate config file, but without Hardware acceleration it may bring everything to a grinding halt.

## Notifications & MQTT Integration

**Coming Soon**, detailed instructions on setting up Notifications for object detection and integrating with the rest of Home Assistant

## Motion Tracking

**Coming Soon**, detailed instructions on setting up motion tracking and other features

## Cameras Tested

| Camera Model              | Features       | Works? | Notes                                                       |
|---------------------------|----------------|--------|-------------------------------------------------------------|
| <a href="#">Tapo C210</a> | 2k, Pan & Tilt | ✓      | Internet Required for Setup                                 |
| Tapo C110                 | 2k, Cheap      | ✗      | Internet Required for Setup<br>RTSP not stable              |
| Wyze                      | Cheap          | ✗      | No RTSP Support. <a href="#">Hacky workaround available</a> |