

Advanced Setup

Covering more advanced setup topics as I learn them

- [433 MHz Devices](#)
- [Frigate NVR](#)

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](#)[†]
- [OPUS XT300](#)[†]

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.

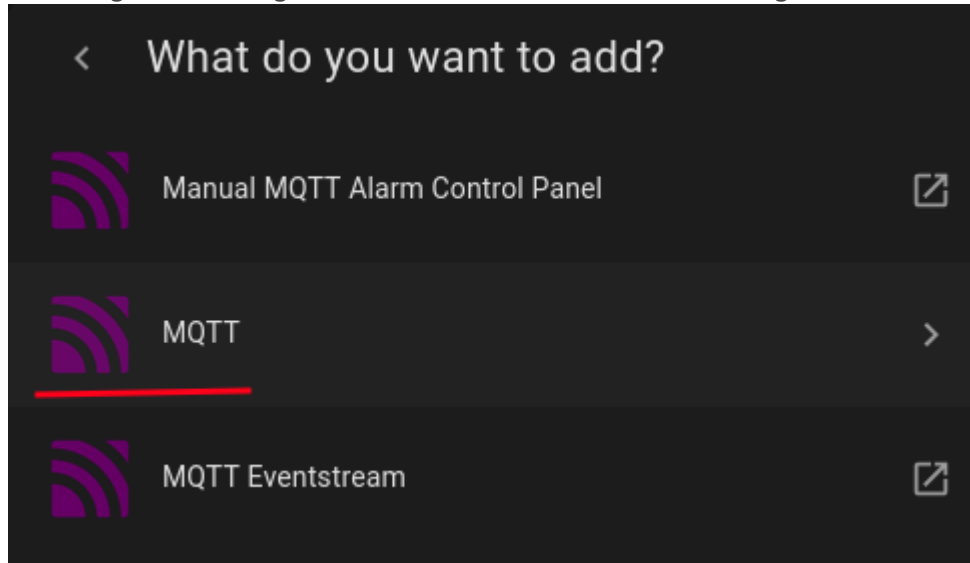
Setting Up Your SDR Receiver

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.

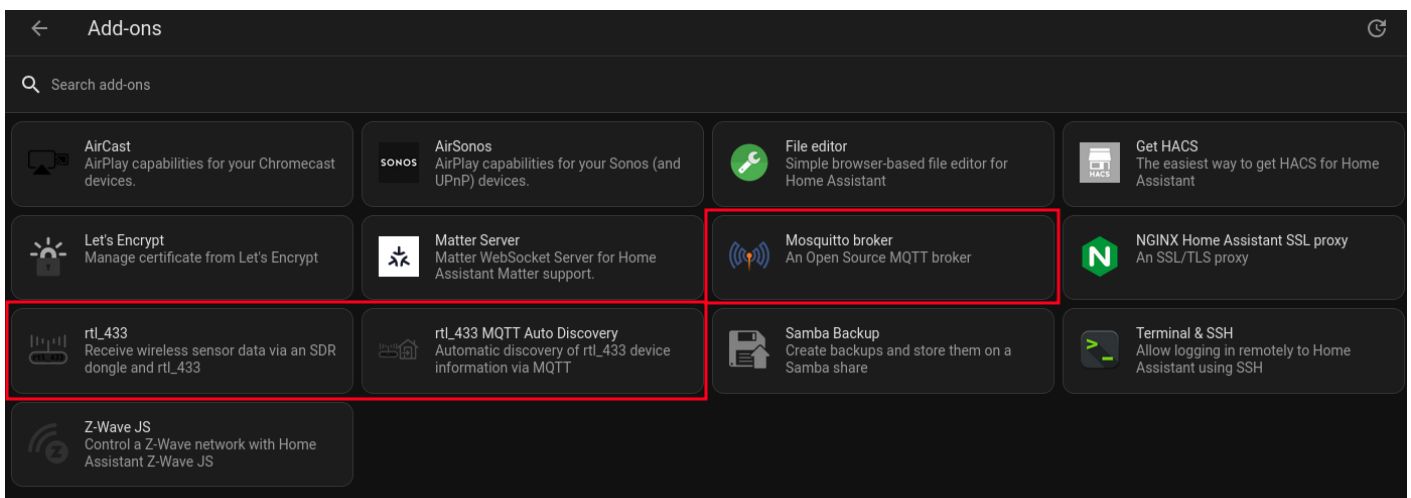
Note: If you want to play with the SDR receiver on your linux desktop/laptop first, see my guide here: <https://www.wswapps.com/books/debian-ubuntu-pop-os/page/install-configure-sdr-receiver>

Add-on Installation

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



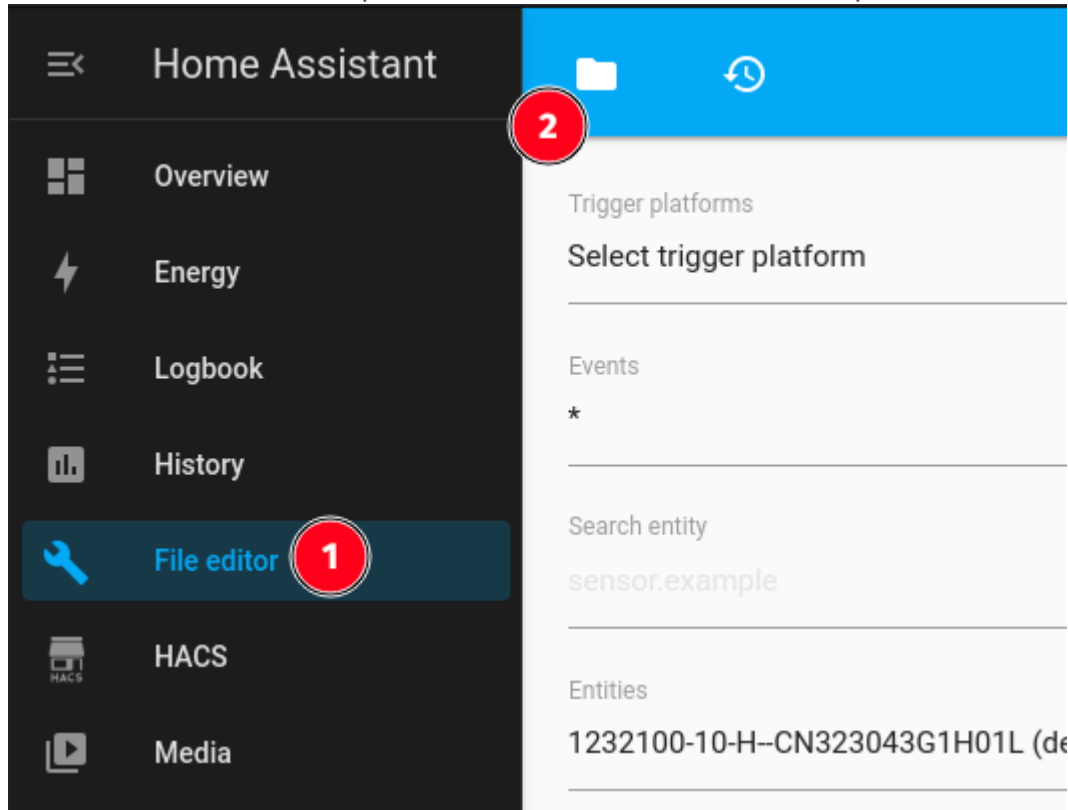
- All your future devices will show up under this MQTT service
- Next, go to Settings > Add-ons > Add-on Store. Select and install: **Mosquito broker**
- Start Mosquito Broker
 - No configuration is needed
 - This add-on is necessary to take in the detected sensors/messages and publish them to Home Assistant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



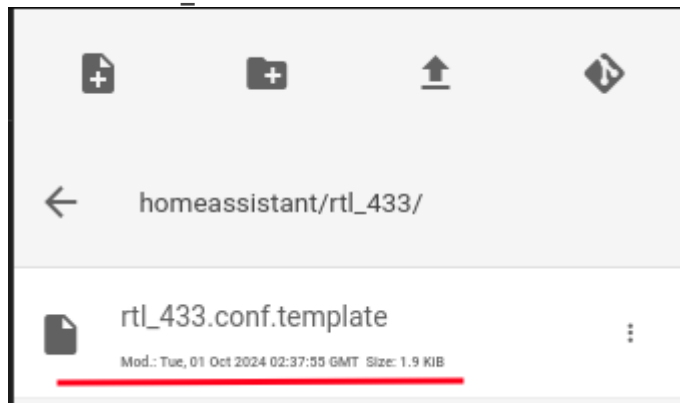
Add-On Configuration

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

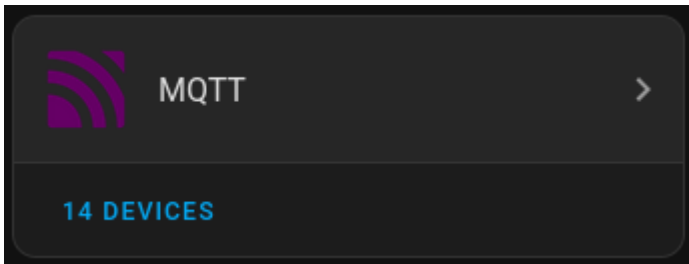
# Disable TPMS sensors by default. These can cause an overwhelming number of
# devices and entities to show up in Home Assistant.
# This list is generated by running:
# rtl_433 -R help 2>&1 | grep -i tpms | sd '.*[(\d+)\.]*' 'protocol -/run.sh'
# [59] Steelmate TPMS
# [60] Schrader TPMS
# [82] Citroen 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://www.reddit.com/r/homeassistant/comments/10pkerb/i_have_no_idea_how_to_use_rtl_433/
- 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.

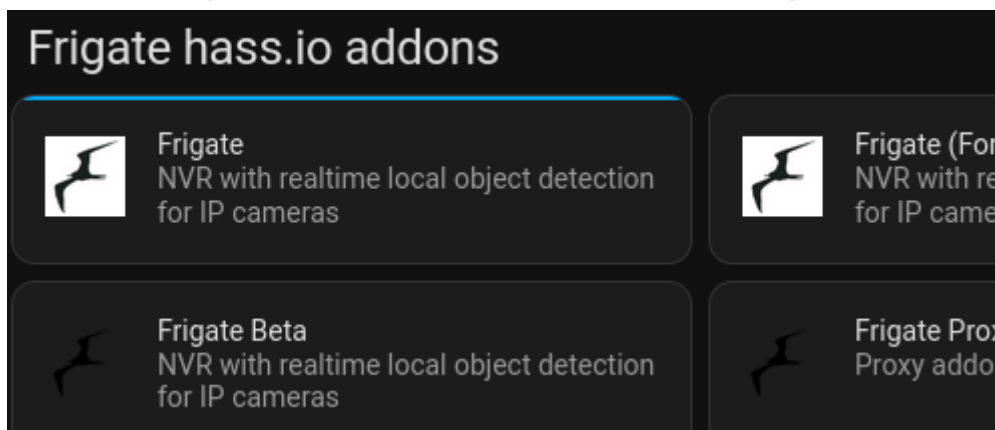
Setup Cameras

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 haas.io addons** section, select **Frigate** and click **Install**

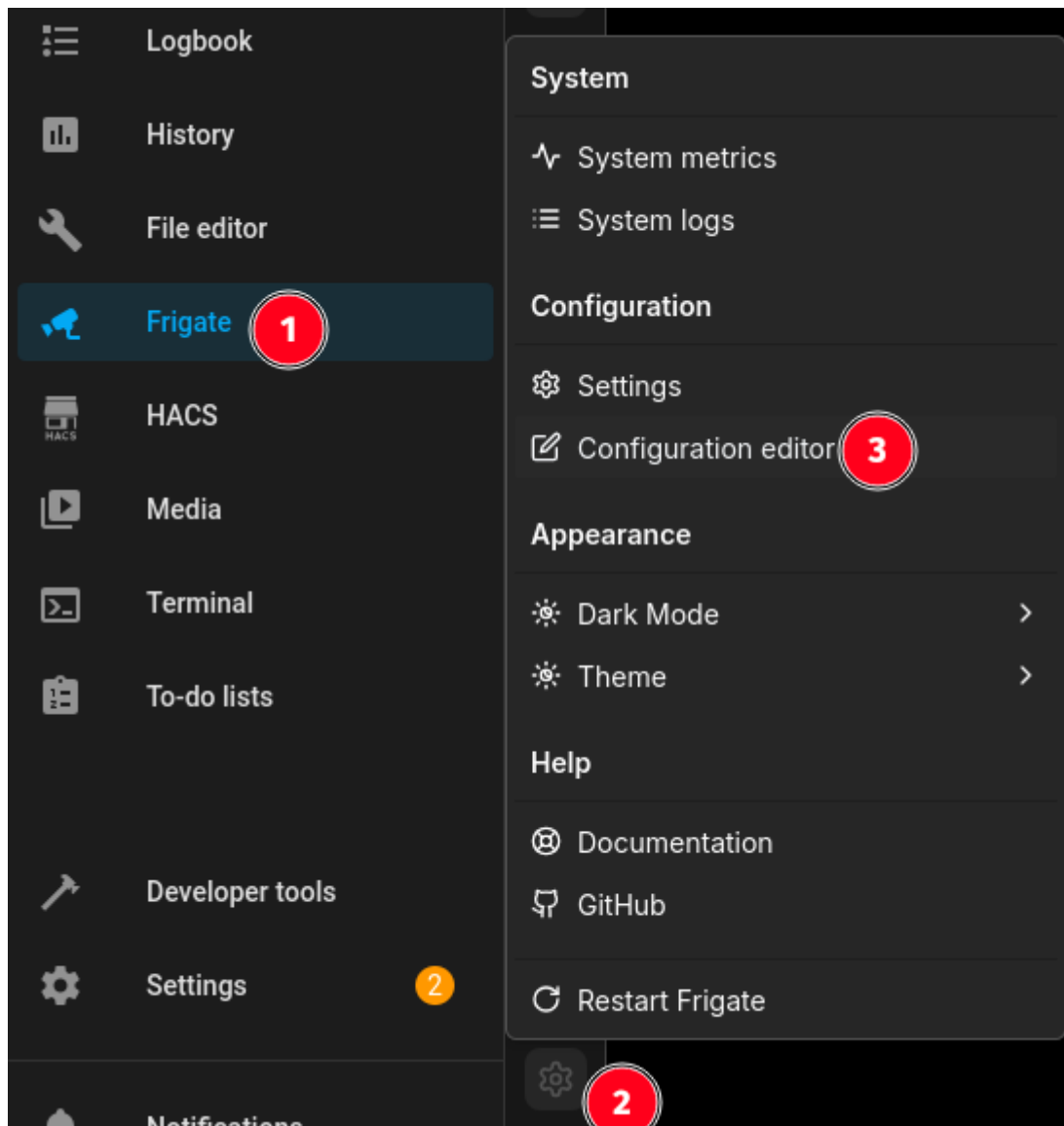


- Frigate (Full Access) is another option that can be used if the regular Frigate is not able to properly connect to your devices. It is not recommended unless absolutely needed since it has unrestricted access to your computer. Install the regular Frigate for now and you can always install Full Access 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
Tapo C210	2k, Pan & Tilt	✓	Internet Required for Setup
Tapo C110	2k, Cheap	✗	Internet Required for Setup RTSP not stable
Wyze	Cheap	✗	No RTSP Support. Hacky workaround available