LinkTap Gateway MQTT Client Integration

V2.0

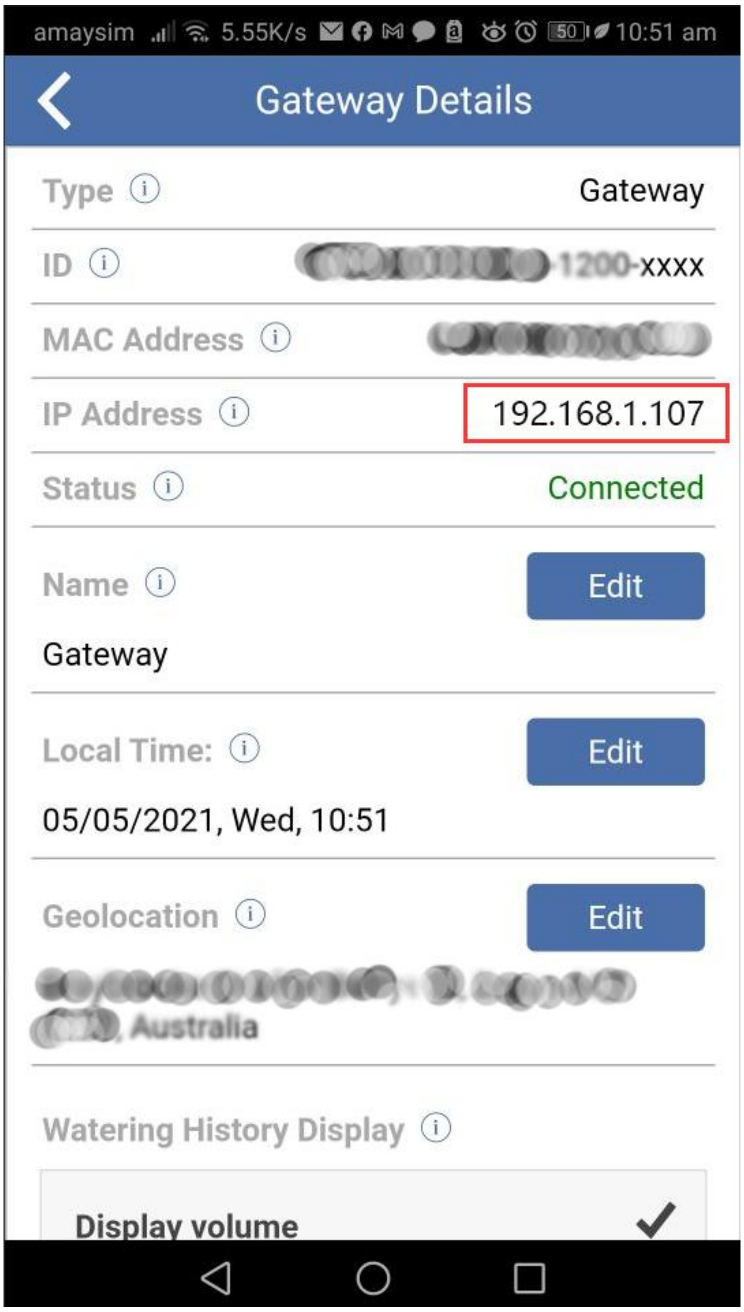
Revision history		
Version	Date	Changes
V1.0	2021/06/01	Initial version
V1.1	2021/06/06	Add a description of “Function” options in MQTT client settings.
V1.2	2021/06/15	Add chapter “3. Integrate with General MQTT Broker”.
V1.3	2021/06/20	Add a description about Taplinker’s fault alerts on Home Assistant. Add chapter “4. Zero Configuration Device Discovery”.
V1.4	2021/10/20	Add chapter “1.3. Misc Settings”. Add a description about options of the water timer' status message publishing format.
V1.5	2021/10/31	Correct the entity type error in the water timer associated entity list.
V1.6	2022/01/15	Change chapter 4 to chapter 5, Add new chapter “4. Integrate with HTTP API”.
V1.7	2022/03/27	Add a description of how to set a failsafe watering duration for LinkTap devices in Home Assistant.
V1.8	2022/08/05	Add chapter “2.7. Create a script via Blueprint to control Taplinker’s watering”.
V1.9	2022/08/10	Add instructions for "set watering duration and volume" and changed the order of some chapters.
V2.0	2023/09/15	Add chapter “2.10. Track Taplinker water usage in the Energy dashboard”

TABLE OF CONTENTS

1. LinkTap Gateway's Admin Page 3
1.1. Ethernet Settings 5
1.2. MQTT Client Settings 5
1.3. Misc Settings 6
2. Integrate with Home Assistant 6
2.1. Install Home Assistant 6
2.2. Install MQTT Broker 6
2.3. MQTT Client Settings for LinkTap Gateway 10
2.4. Watering Control Through Entity 12
2.5. Set watering duration (value of sensor.X_total_duration entity) 27
2.6. Set watering volume (value of sensor.X_volume_limit entity) 28
2.7. Set failsafe watering duration 30
2.8. Taplinker's Fault Alerts on Home Assistant 33
2.9. Create a script via Blueprint to control Taplinker's watering 35
2.10. Track Taplinker water usage in the Energy dashboard 38
3. Integrate with General MQTT Broker 45
3.1. System Diagram 45
3.2. LinkTap Gateway Settings 46
3.3. LinkTap Gateway MQTT Messages Interaction 48
3.4. Example of the Interaction 49
3.5. Implementation of Third-Party Application 52
4. Integrate with HTTP API 53
4.1. Push data to server from the gateway 53
4.2. Send Commands via HTTP to the gateway for end-device control 54
5. Zero Configuration Device Discovery 55

1. LinkTap Gateway’s Admin Page

By default, the LinkTap Gateway uses DHCP to obtain an IP address from a router or other DHCP servers. After the Gateway is connected to the LinkTap server, its current IP address can be seen from the LinkTap App.



Type Gateway

ID [redacted]1200-xxxx

MAC Address [redacted]

IP Address 192.168.1.107

Status Connected

Name

Gateway

Local Time:

05/05/2021, Wed, 10:51

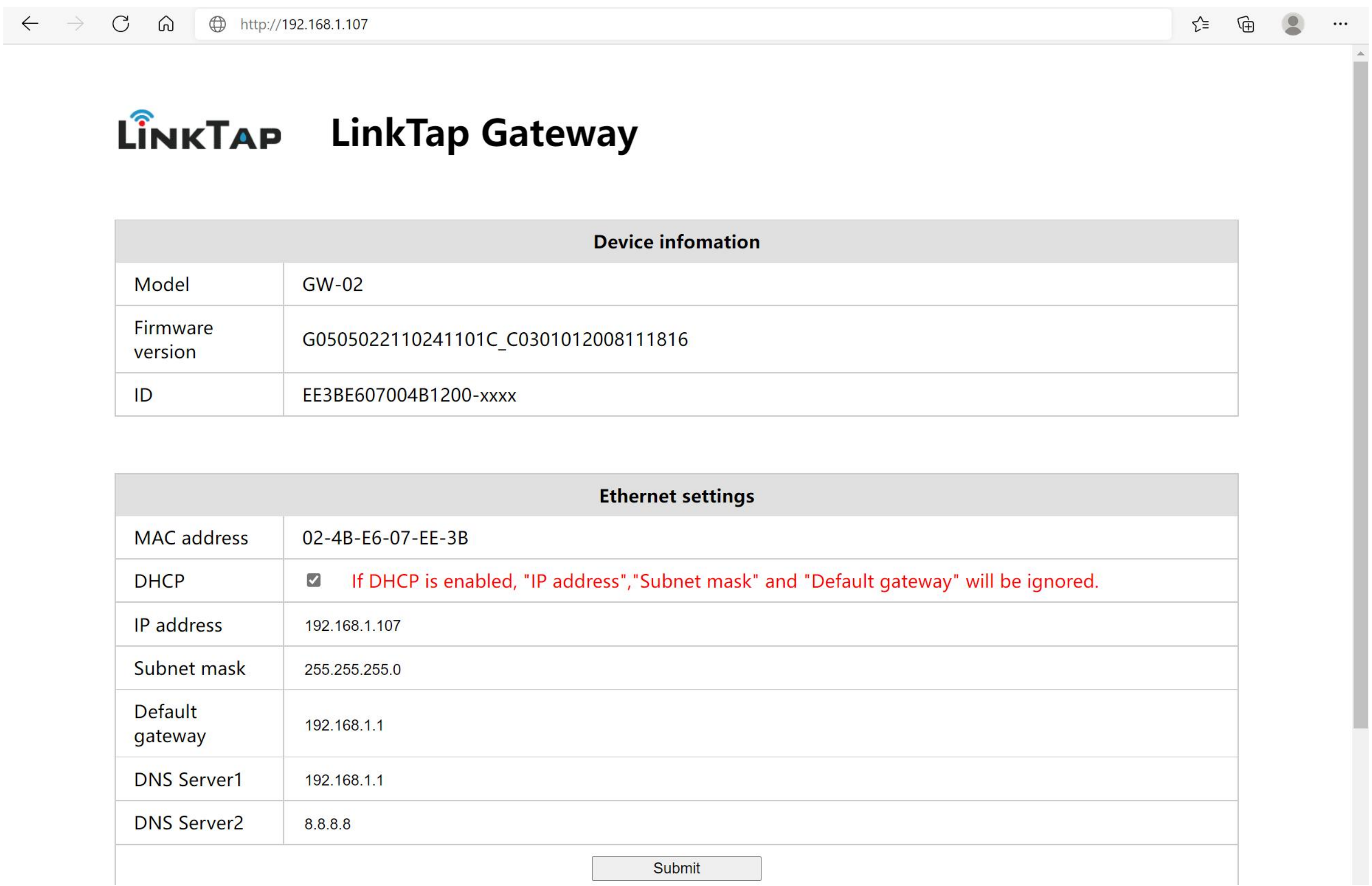
Geolocation

[redacted] Australia

Watering History Display

Display volume ✓

Open a browser, enter the Gateway’s IP address (e.g., 192.168.1.107), the Gateway’s admin page will be displayed, as shown below.



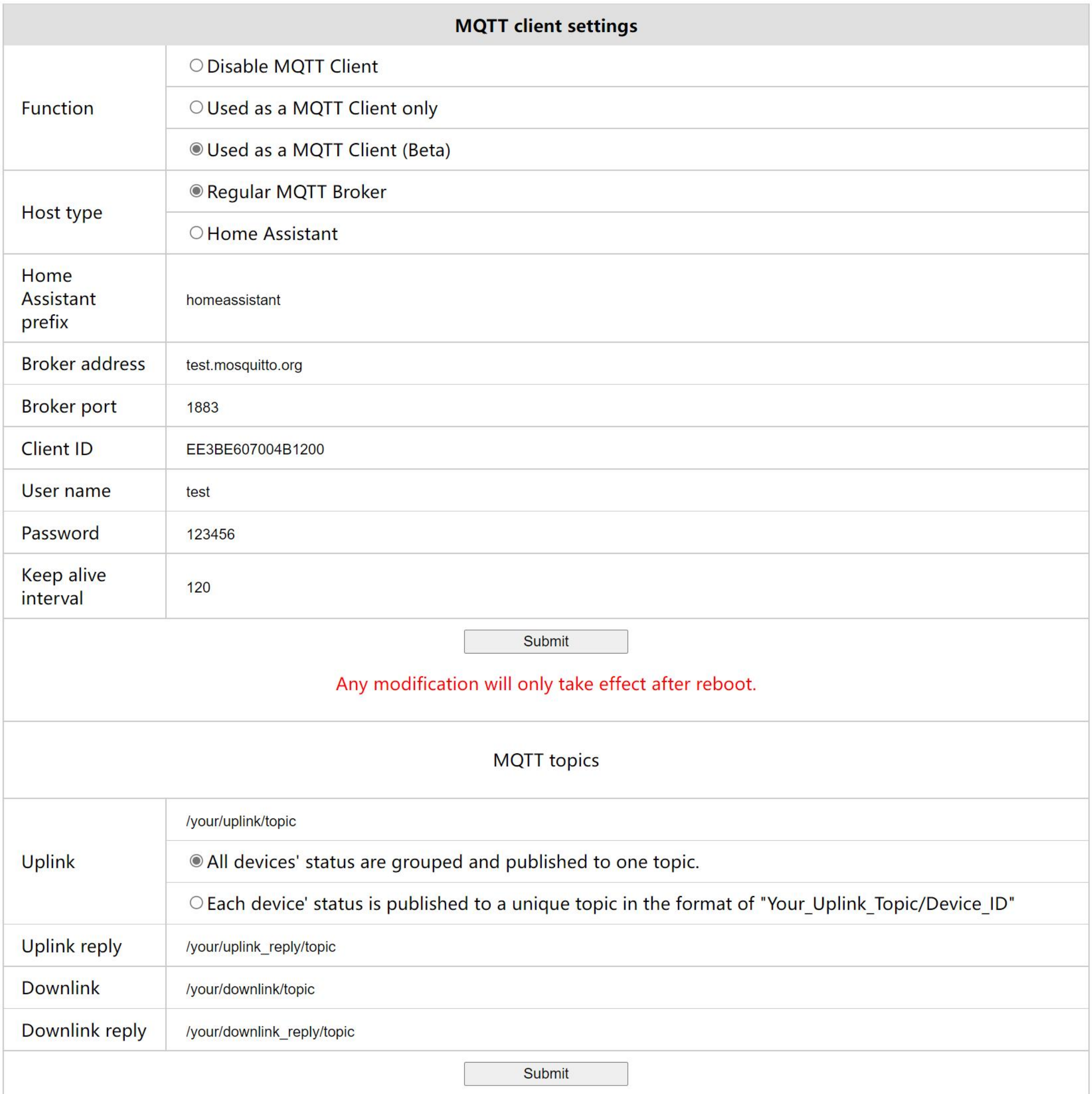
http://192.168.1.107

LinkTap Gateway

Device infomation	
Model	GW-02
Firmware version	G0505022110241101C_C0301012008111816
ID	EE3BE607004B1200-xxxx

Ethernet settings	
MAC address	02-4B-E6-07-EE-3B
DHCP	☑ If DHCP is enabled, "IP address","Subnet mask" and "Default gateway" will be ignored.
IP address	192.168.1.107
Subnet mask	255.255.255.0
Default gateway	192.168.1.1
DNS Server1	192.168.1.1
DNS Server2	8.8.8.8

Submit



MQTT client settings	
Function	○ Disable MQTT Client
	○ Used as a MQTT Client only
	● Used as a MQTT Client (Beta)
Host type	● Regular MQTT Broker
	○ Home Assistant
Home Assistant prefix	homeassistant
Broker address	test.mosquitto.org
Broker port	1883
Client ID	EE3BE607004B1200
User name	test
Password	123456
Keep alive interval	120

Submit

Any modification will only take effect after reboot.

MQTT topics	
Uplink	/your/uplink/topic
	● All devices' status are grouped and published to one topic.
	○ Each device' status is published to a unique topic in the format of "Your_Uplink_Topic/Device_ID"
Uplink reply	/your/uplink_reply/topic
Downlink	/your/downlink/topic
Downlink reply	/your/downlink_reply/topic

Submit

1.1. Ethernet Settings

- MAC address: The Gateway's MAC address, not editable. The MAC address can be used to assign a static IP address to the Gateway from your router's admin page.
- DHCP is enabled by default. When the DHCP is enabled, the "IP address", "Subnet mask" and "Default gateway" will be ignored.
- If the user wants to assign a static IP address to the Gateway without modifying the router or the DHCP server, uncheck the DHCP, then set the "IP address", "Subnet mask" and "Default gateway" fields. Make sure that the "IP address" is not in the DHCP address pool of the router or DHCP server to avoid IP conflicts in the LAN.
- Reset Gateway to factory settings: If the Gateway's admin page cannot be accessed due to configuration errors or other reasons, also the Gateway cannot connect to the LinkTap server, user can then press and hold the reset button of the Gateway after it has started (indicated by Link LED turning to solid green) until *Link*, *Internet* and *Sync* LEDs flash green three times at the same time. After that the Gateway will automatically restart to restore the factory settings.

1.2. MQTT Client Settings

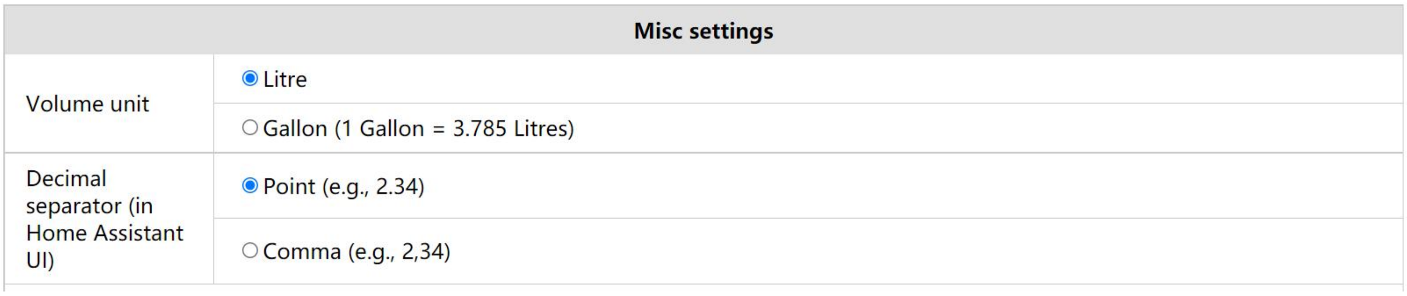
- Function:
  - ✧ Disable MQTT Client function. The LinkTap Gateway will become non-discoverable for MQTT server.
  - ✧ Used as a MQTT client only. The LinkTap Gateway will be disconnected from the LinkTap server. You will NOT be able to control your LinkTap devices through the LinkTap app.
  - ✧ Used as a MQTT client. The LinkTap Gateway also maintains a connection with the LinkTap server. You can still control your LinkTap devices through the LinkTap app.

> NOTE: When the "Used as a MQTT client" option is selected, both third-party MQTT client and LinkTap App can send commands to the gateway. But since this function is still in Beta stage, the consistency of the watering plan and other settings between the MQTT client and LinkTap App has yet to be resolved. Therefore, before the consistency issue has been solved, it is not recommended to use third-party MQTT client and LinkTap App in parallel to start/stop watering or modify the watering mode, etc.

- Host type: The MQTT server type. The Host type determines the specific definition of the MQTT topic and message (payload).
- Home Assistant prefix: The prefix of the Home Assistant. The default value is "homeassistant". "Home Assistant prefix" is valid only when the "Host type" is "Home Assistant".
- Broker address: The address of the MQTT broker. It can be an IP address or a domain name.
- Broker port: The port number of the MQTT broker. The default value is 1883.
- Client ID: The MQTT client's ID. You can set it to the Gateway's ID when the "Host type" is "Home Assistant".
- User name: The username required to access the MQTT broker. It is the username of Home Assistant when the "Host type" is "Home Assistant".
- Password: The password required to access the MQTT broker. It is the password for logging in to the Home Assistant when the "Host type" is "Home Assistant".
- Keep alive interval: Keep-alive interval (unit: second) between the MQTT client and the server. If it times out and also there is no communication, it is considered as disconnected communication.

- Uplink: Used for "Regular MQTT Broker". There are two options for configuring how the device status is published.
    - ✧ (Default) All devices' status are grouped and published to one topic. The LinkTap Gateway combines the status of multiple devices into one message; its length is less than the MTU (Max transmission unit, the value is 1500 in the Gateway).
    - ✧ (Optional) Each device' status is published to a unique topic in the format of "Your_Uplink_Topic/Device_ID".
- Uplink reply: Used for "Regular MQTT Broker".
- Downlink: Used for "Regular MQTT Broker".
- Download reply: Used for "Regular MQTT Broker".

1.3. Misc Settings



Misc settings	
Volume unit	● Litre
	○ Gallon (1 Gallon = 3.785 Litres)
Decimal separator (in Home Assistant UI)	● Point (e.g., 2.34)
	○ Comma (e.g., 2,34)

- Volume unit: the water flow rate and volume values reported by the water timer in litres or gallons.
- Decimal separator: The configuration of decimal display in the Home Assistant UI.

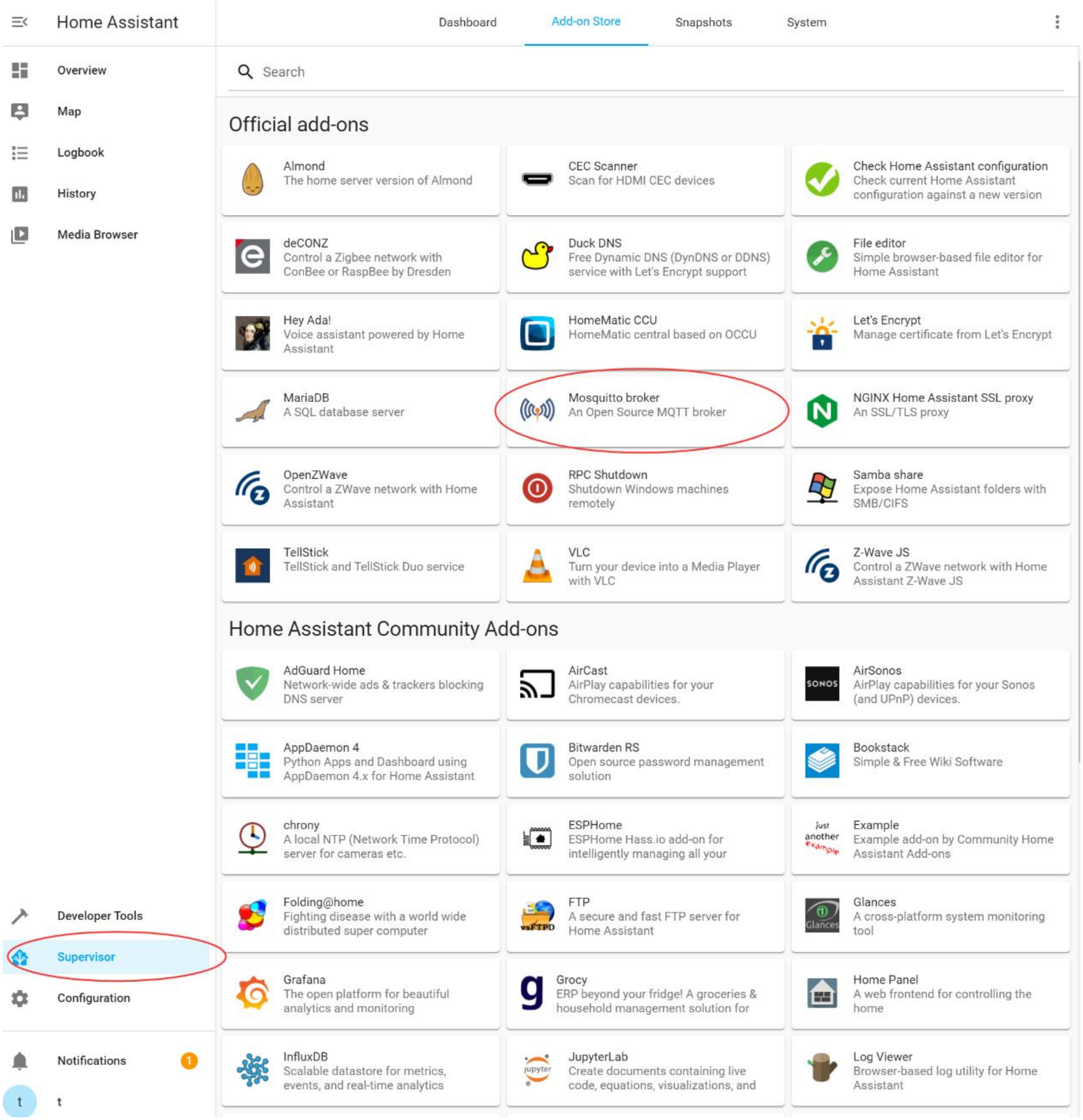
2. Integrate with Home Assistant

2.1. Install Home Assistant

Please refer to https://www.home-assistant.io/installation for the Home Assistant installation.

2.2. Install MQTT Broker

2.2.1. After installing Home Assistant, log in to Home Assistant in your browser, click "Supervisor" on the left sidebar, then select "Add-on Store", find "Mosquitto broker" and click.



Official add-ons

Almond – The home server version of Almond

CEC Scanner – Scan for HDMI CEC devices

Check Home Assistant configuration – Check current Home Assistant configuration against a new version

deCONZ – Control a Zigbee network with ConBee or RaspBee by Dresden

Duck DNS – Free Dynamic DNS (DynDNS or DDNS) service with Let's Encrypt support

File editor – Simple browser-based file editor for Home Assistant

Hey Ada! – Voice assistant powered by Home Assistant

HomeMatic CCU – HomeMatic central based on OCCU

Let's Encrypt – Manage certificate from Let's Encrypt

MariaDB – A SQL database server

Mosquitto broker – An Open Source MQTT broker

NGINX Home Assistant SSL proxy – An SSL/TLS proxy

OpenZWave – Control a ZWave network with Home Assistant

RPC Shutdown – Shutdown Windows machines remotely

Samba share – Expose Home Assistant folders with SMB/CIFS

TellStick – TellStick and TellStick Duo service

VLC – Turn your device into a Media Player with VLC

Z-Wave JS – Control a ZWave network with Home Assistant Z-Wave JS

Home Assistant Community Add-ons

AdGuard Home – Network-wide ads & trackers blocking DNS server

AirCast – AirPlay capabilities for your Chromecast devices.

AirSonos – AirPlay capabilities for your Sonos (and UPnP) devices.

AppDaemon 4 – Python Apps and Dashboard using AppDaemon 4.x for Home Assistant

Bitwarden RS – Open source password management solution

Bookstack – Simple & Free Wiki Software

chrony – A local NTP (Network Time Protocol) server for cameras etc.

ESPHome – ESPHome Hass.io add-on for intelligently managing all your

Example – Example add-on by Community Home Assistant Add-ons

Folding@home – Fighting disease with a world wide distributed super computer

FTP – A secure and fast FTP server for Home Assistant

Glances – A cross-platform system monitoring tool

Grafana – The open platform for beautiful analytics and monitoring

Grocy – ERP beyond your fridge! A groceries & household management solution for

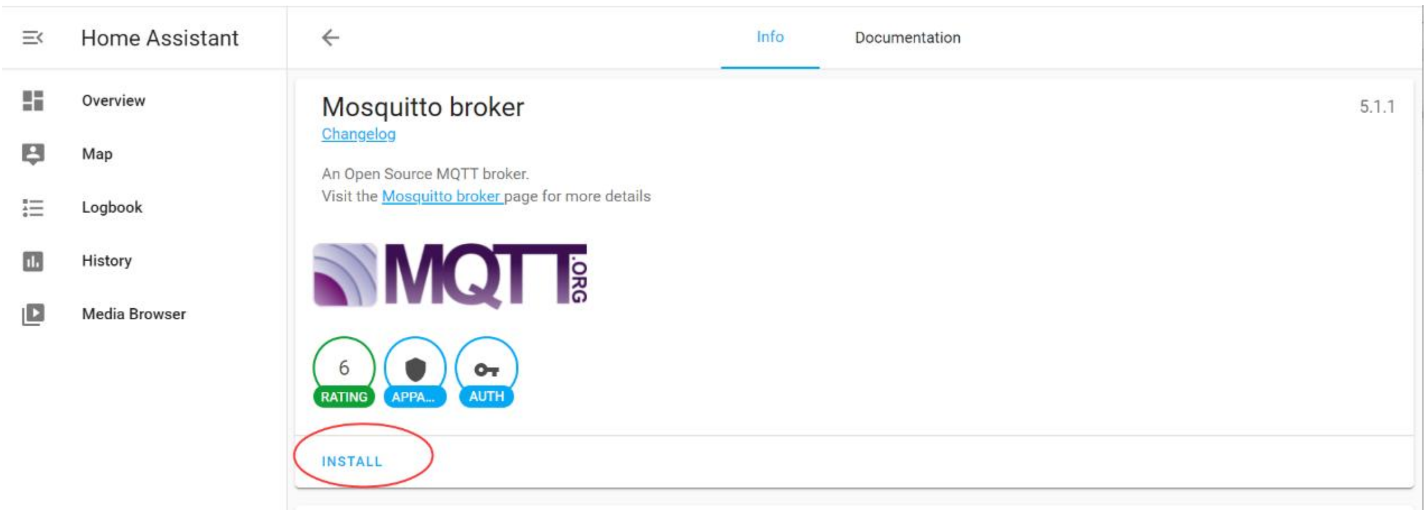
Home Panel – A web frontend for controlling the home

InfluxDB – Scalable datastore for metrics, events, and real-time analytics

JupyterLab – Create documents containing live code, equations, visualizations, and

Log Viewer – Browser-based log utility for Home Assistant

2.2.2. Click "INSTALL" to start the installation.

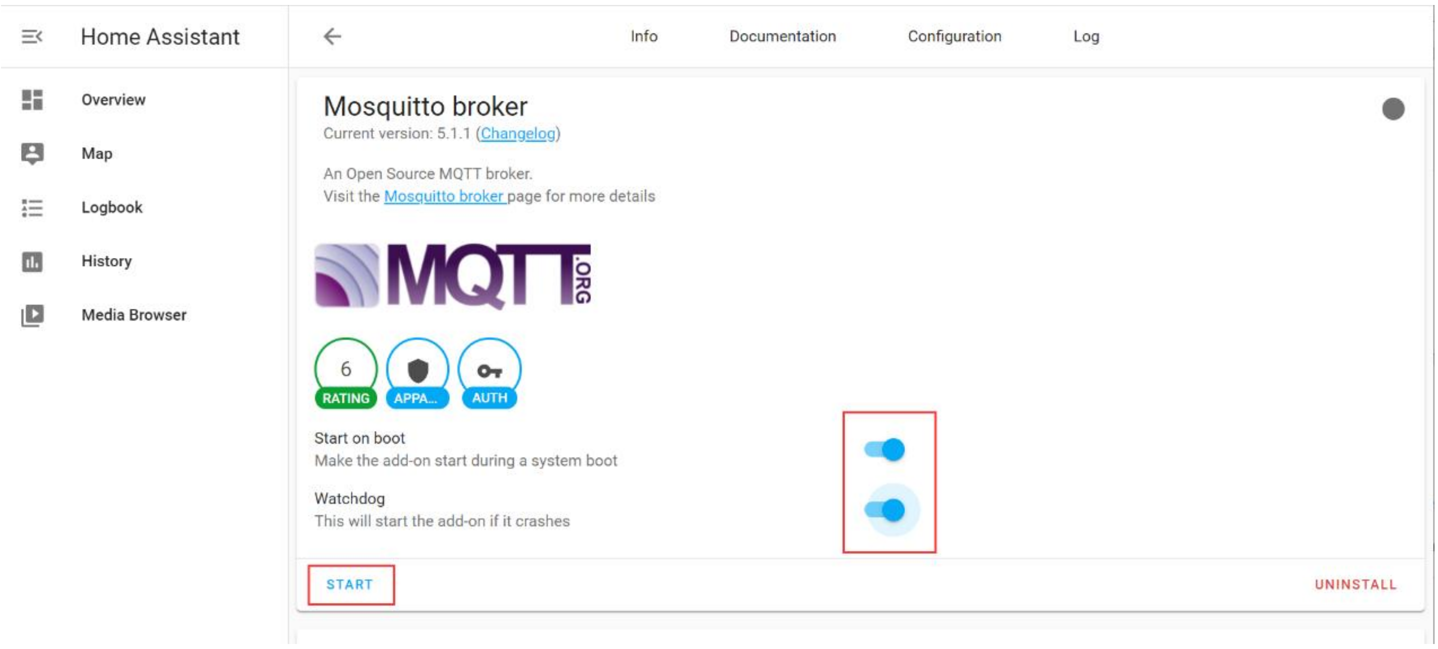


Mosquitto broker 5.1.1

Changelog

An Open Source MQTT broker.
Visit the Mosquitto broker page for more details

2.2.3. When the installation is completed, make sure “Start on boot” and “Watchdog” are enabled, then click “START” to start the MQTT broker.



Mosquitto broker

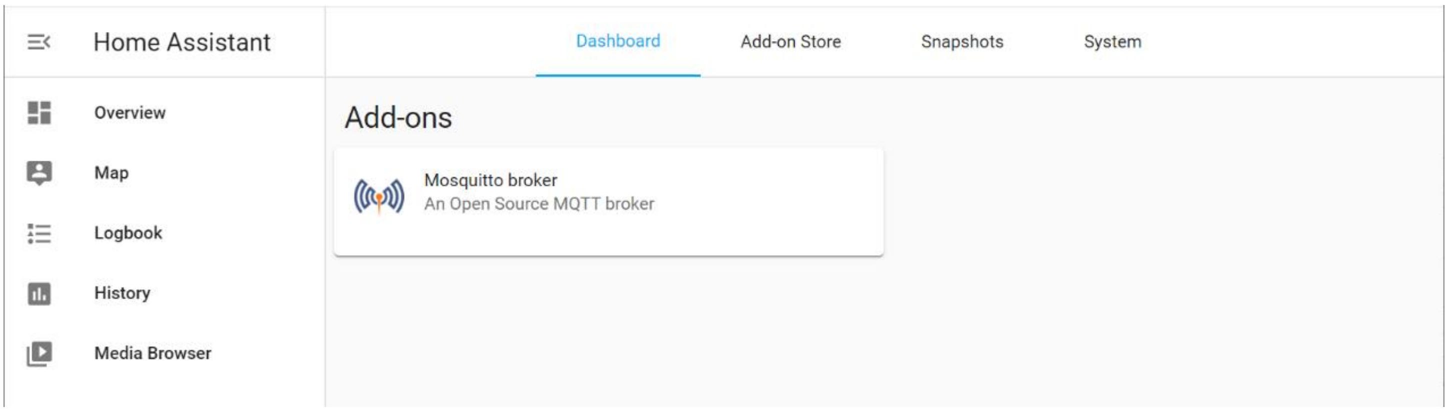
Current version: 5.1.1 (Changelog)

An Open Source MQTT broker.
Visit the Mosquitto broker page for more details

Start on boot
Make the add-on start during a system boot

Watchdog
This will start the add-on if it crashes

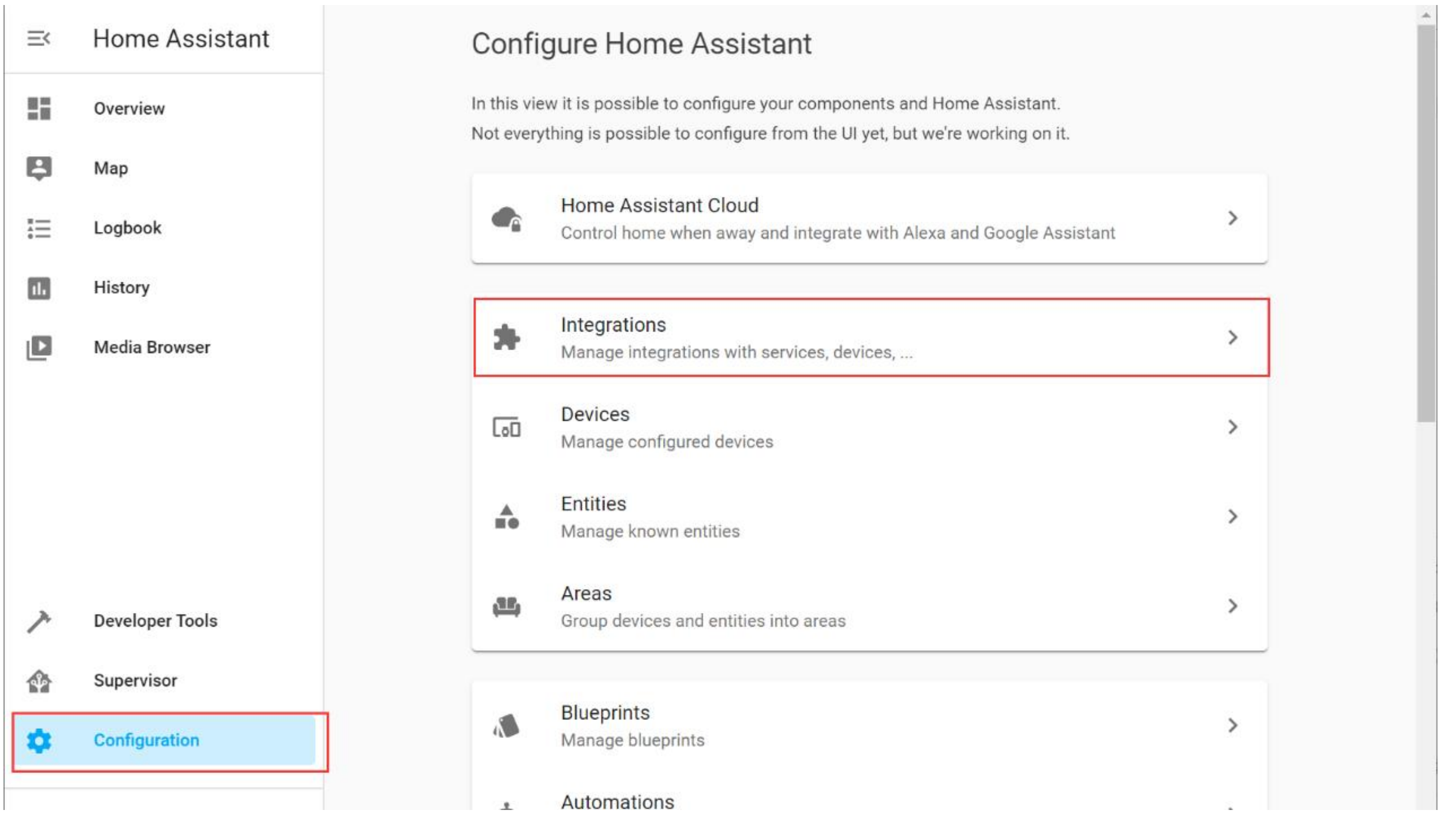
2.2.4. You should be able to see “Mosquitto broker” now on the Dashboard.



Add-ons

Mosquitto broker
An Open Source MQTT broker

2.2.5. Click “Configuration” on the left sidebar, then choose “Integrations”.



Configure Home Assistant

In this view it is possible to configure your components and Home Assistant.
Not everything is possible to configure from the UI yet, but we're working on it.

Home Assistant Cloud
Control home when away and integrate with Alexa and Google Assistant

Integrations
Manage integrations with services, devices, ...

Devices
Manage configured devices

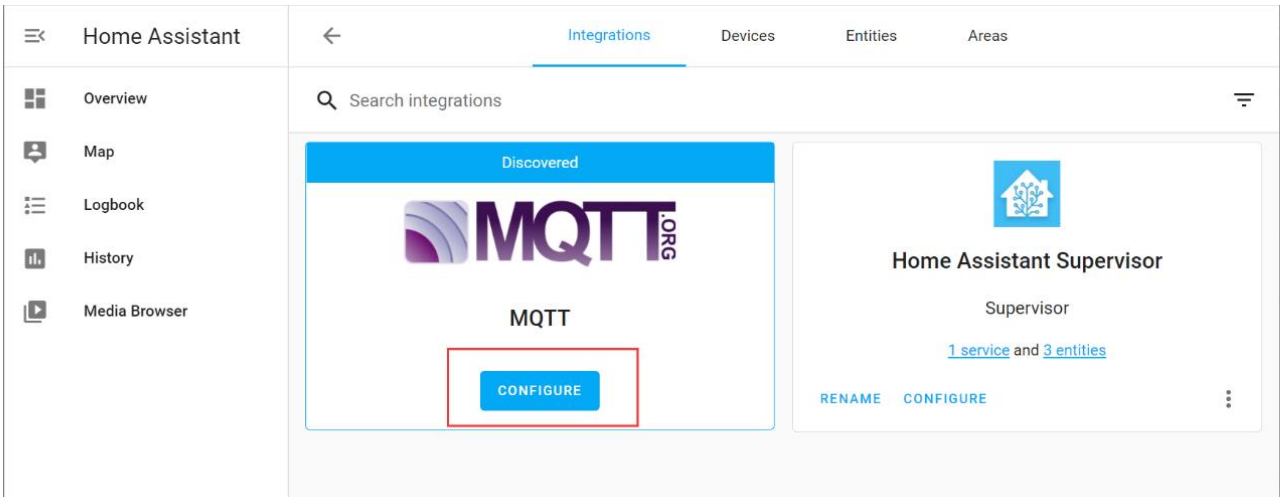
Entities
Manage known entities

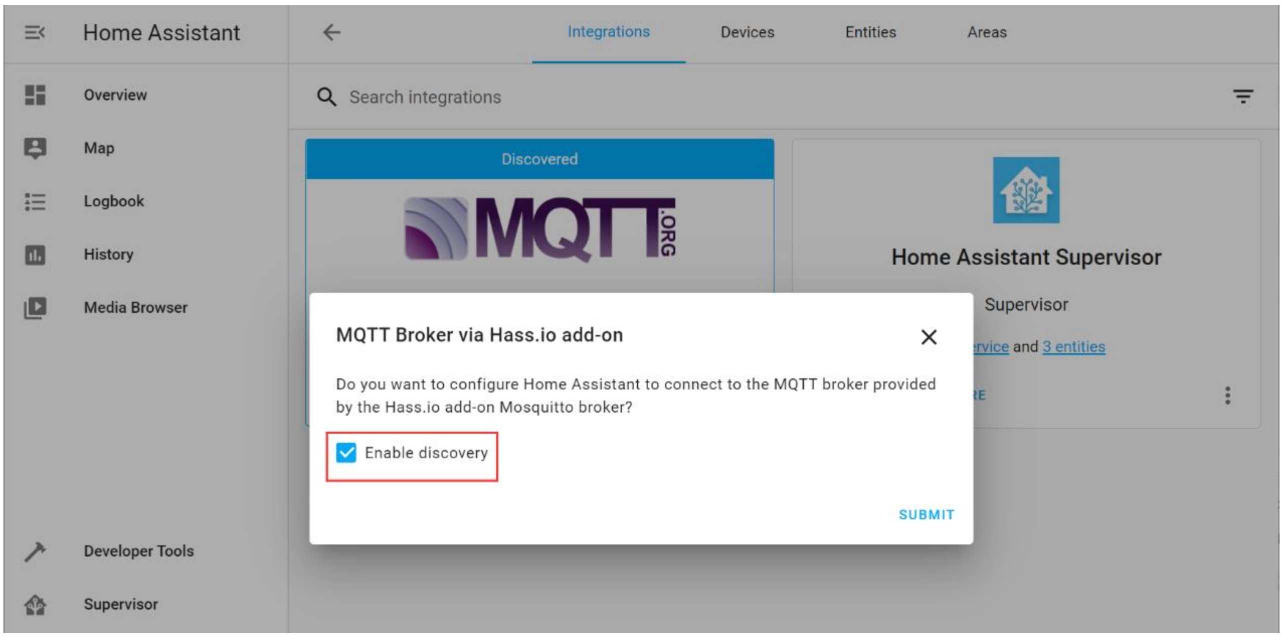
Areas
Group devices and entities into areas

Blueprints
Manage blueprints

Automations

2.2.6. Find “MQTT”, click "CONFIGURE”, then enable "Enable discovery”, click “SUBMIT”. The installation of MQTT broker is completed now. (If the "Enable discovery” option is not shown, then just click “SUBMIT”.)





MQTT Broker via Hass.io add-on

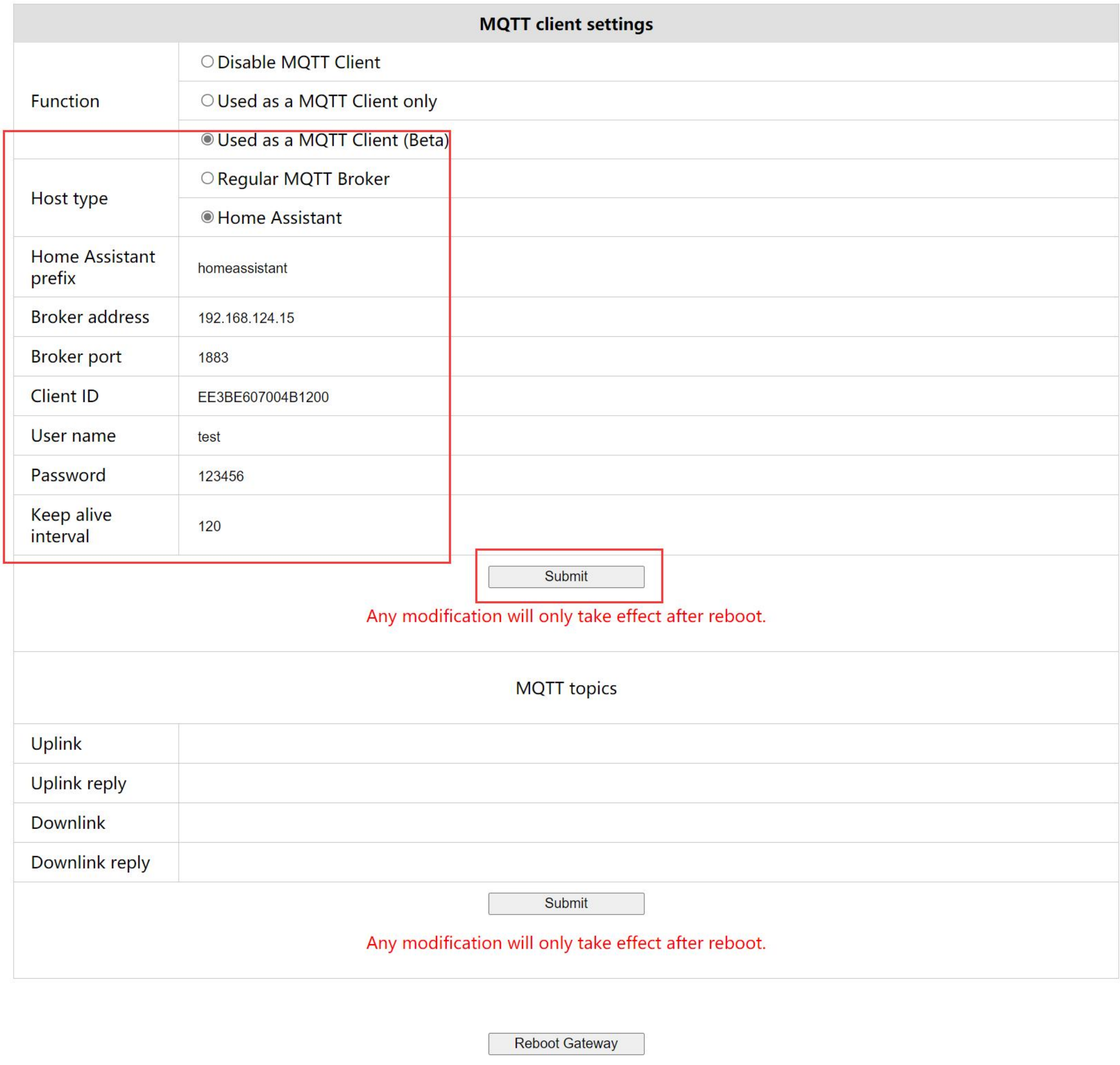
Do you want to configure Home Assistant to connect to the MQTT broker provided by the Hass.io add-on Mosquitto broker?

☑ Enable discovery

SUBMIT

2.3. MQTT Client Settings for LinkTap Gateway

2.3.1. Enter your LinkTap Gateway's IP address in a browser from a computer to open your Gateway's admin page. Make sure the computer and your Gateway are in the same LAN. As an example shown below, enter your Home Assistant's IP address, user name, password and the Gateway's ID etc., then click "Submit". Lastly, you need to click "Restart Gateway" for the changes to take effect.



MQTT client settings	
Function	○ Disable MQTT Client ○ Used as a MQTT Client only ● Used as a MQTT Client (Beta)
Host type	○ Regular MQTT Broker ● Home Assistant
Home Assistant prefix	homeassistant
Broker address	192.168.124.15
Broker port	1883
Client ID	EE3BE607004B1200
User name	test
Password	123456
Keep alive interval	120

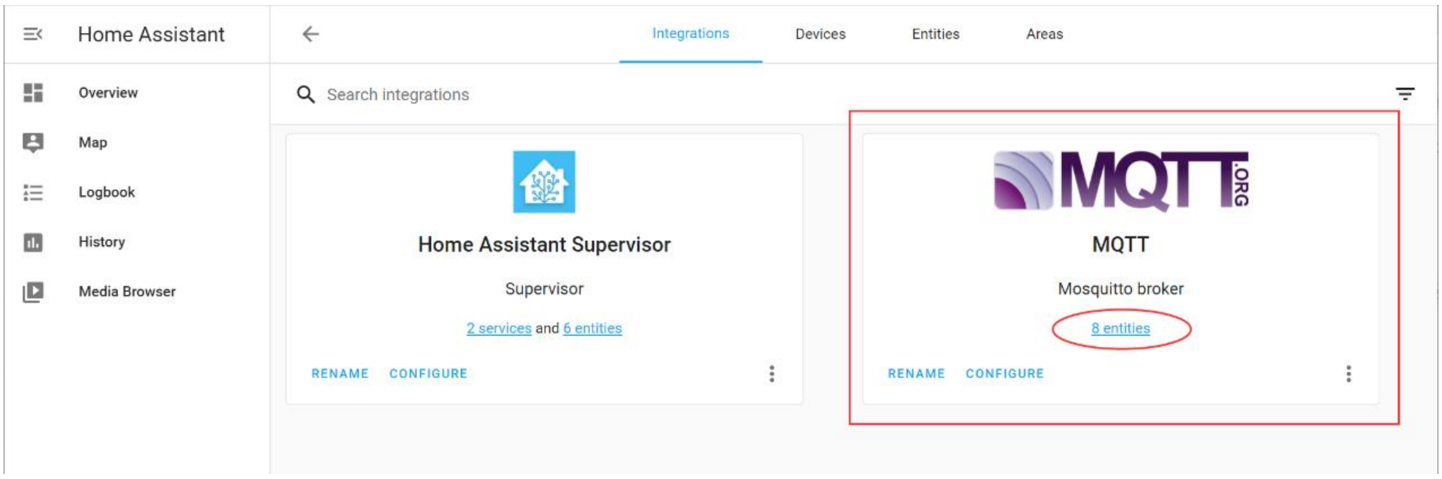
Any modification will only take effect after reboot.

MQTT topics	
Uplink	
Uplink reply	
Downlink	
Downlink reply	

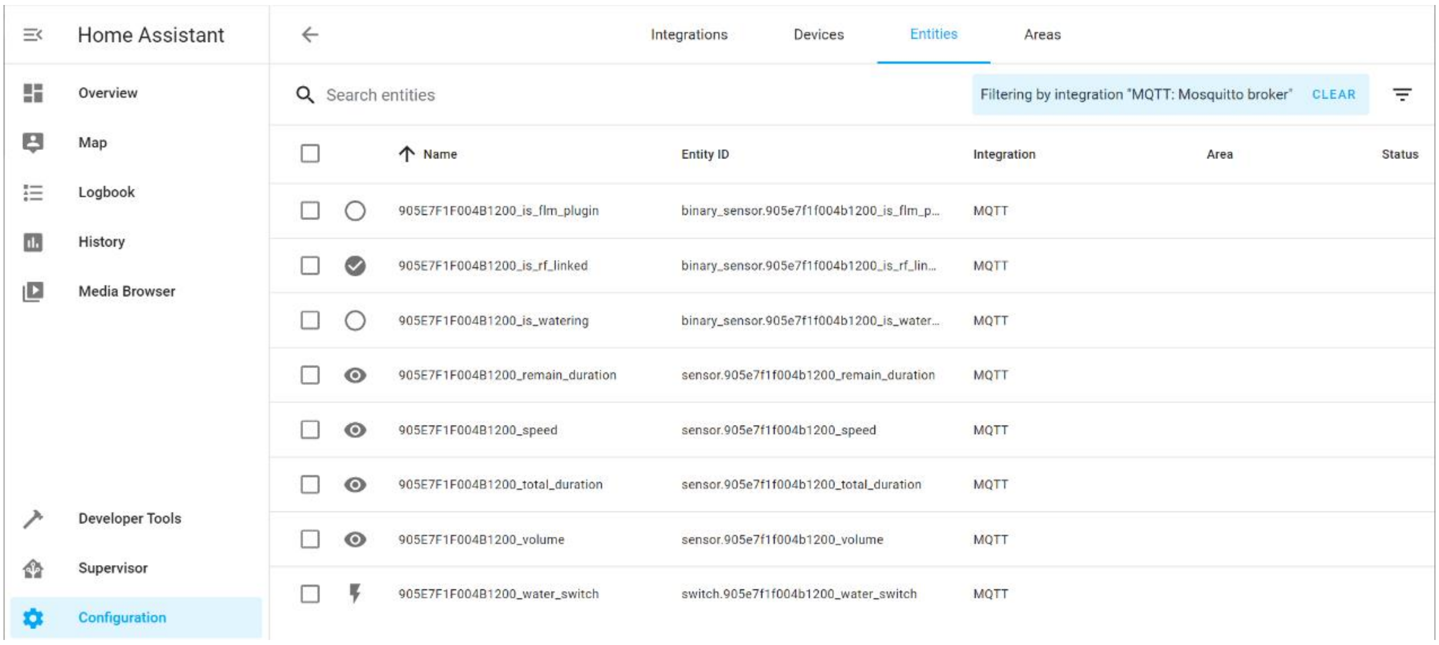
Any modification will only take effect after reboot.

2.3.2. After the Gateway is rebooted, you should be able to see "entities" are added into the

2.3.3. "Mosquitto broker".



2.3.4. Click “entities” to view the entities detected by MQTT broker.



□		Name	Entity ID	Integration	Area	Status
□	○	905E7F1F004B1200_is_flm_plugin	binary_sensor.905e7f1f004b1200_is_flm_p...	MQTT		
□	✓	905E7F1F004B1200_is_rf_linked	binary_sensor.905e7f1f004b1200_is_rf_lin...	MQTT		
□	○	905E7F1F004B1200_is_watering	binary_sensor.905e7f1f004b1200_is_water...	MQTT		
□	◉	905E7F1F004B1200_remain_duration	sensor.905e7f1f004b1200_remain_duration	MQTT		
□	◉	905E7F1F004B1200_speed	sensor.905e7f1f004b1200_speed	MQTT		
□	◉	905E7F1F004B1200_total_duration	sensor.905e7f1f004b1200_total_duration	MQTT		
□	◉	905E7F1F004B1200_volume	sensor.905e7f1f004b1200_volume	MQTT		
□	⚡	905E7F1F004B1200_water_switch	switch.905e7f1f004b1200_water_switch	MQTT		

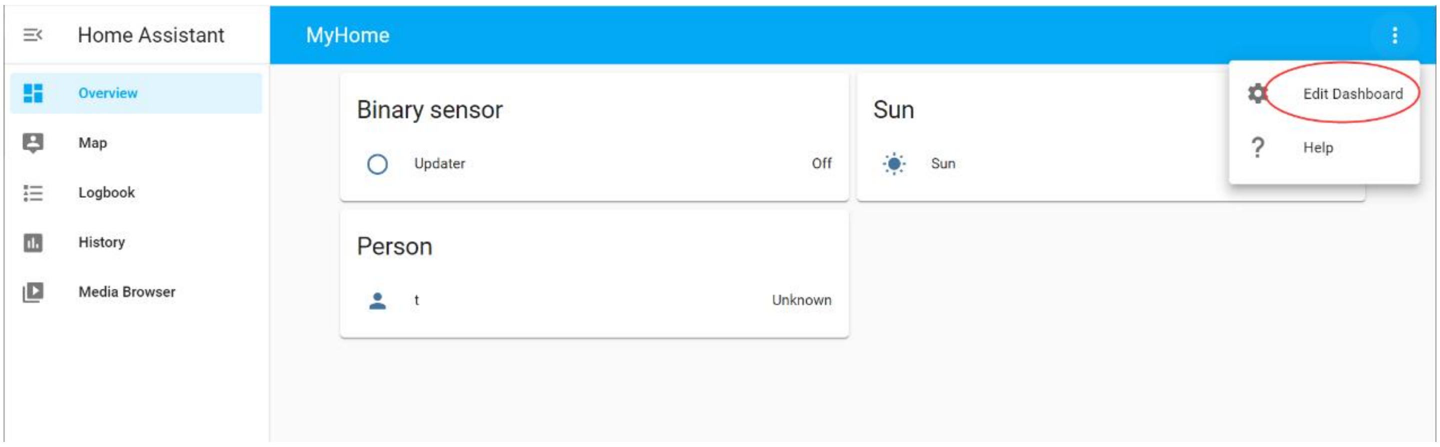
The associated entity of a LinkTap water timer is named in the format of "water timer ID_entity name". "X" in the table below represents the ID of a water timer.

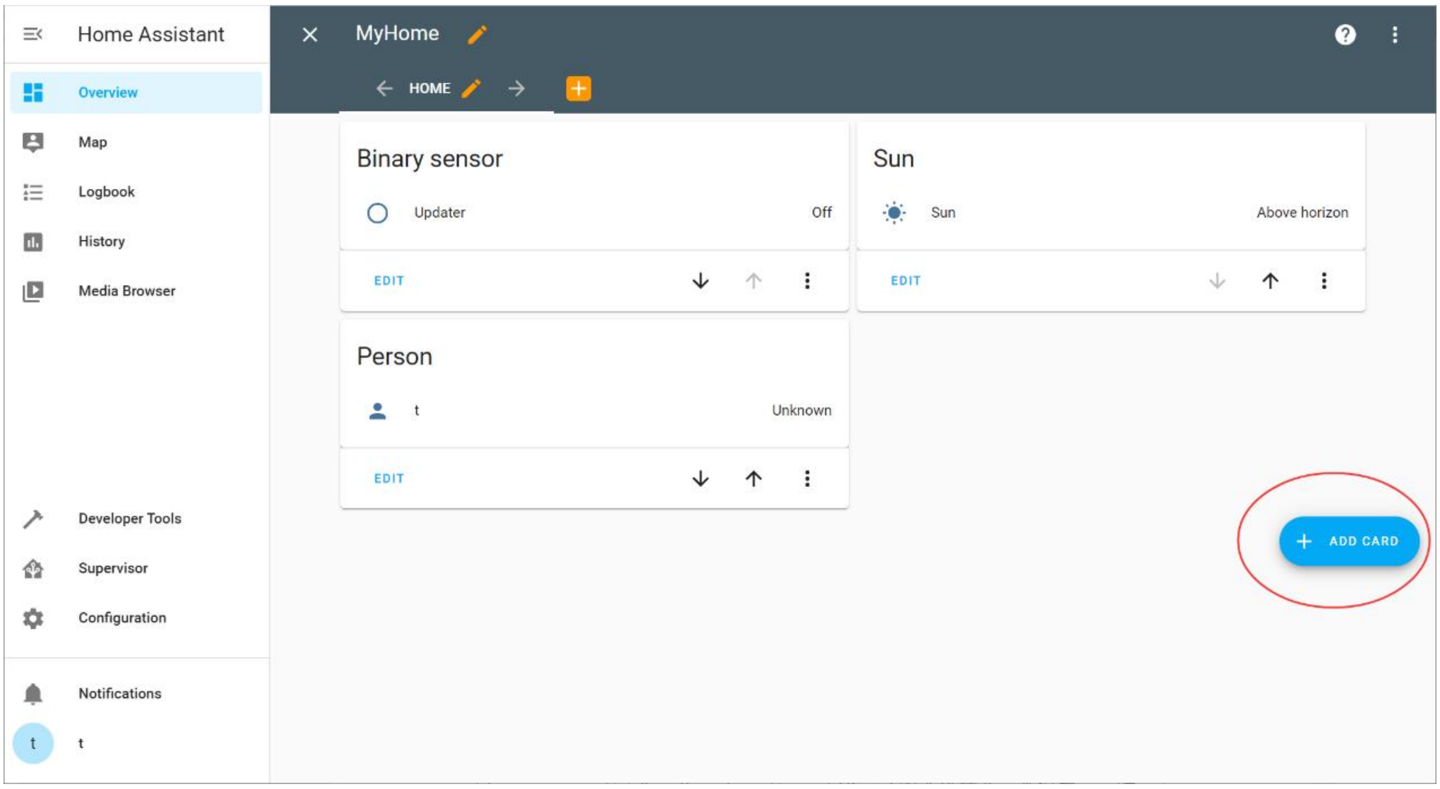
Home Assistant Component	Entity name	Note
binary_sensor	X_is_flm_plugin	Flow meter’s connection status. “on” means the flow meter is connected.
	X_is_rf_linked	Water timer’s connection status. “on” means the water timer is connected to the Gateway.
	X_is_watering	Is water timer currently watering?
	X_is_manual_mode	Is watering by press “Manual On/Off” button (on Taplinker G1S/G2S)?
	X_child_lock	Is child-lock of “Manual On/Off” button (on Taplinker G1S/G2S) enabled?
sensor	X_total_duration	Total watering duration in seconds.
	X_remain_duration	The remaining watering duration in seconds.
	X_speed	Current flow rate (LPM).
	X_volume	Accumulated volume (Litre).
	X_volume_limit	The target volume for current watering cycle. Once the “volume” reaches the “volume_limit”, the Taplinker will stop watering. When it is 0, the watering process retains until “remain_duration” runs out.
	X_signal	Percentage of signal strength of water timer
	X_battery	Percentage of remaining battery power of water timer.
	X_ failsafe_duration	Configurable failsafe duration, in seconds. When the “water_switch” is turned on, the failsafe duration will be used as "total_duration". When it is

		0, "total_duration" uses the default value on the gateway.
switch	X_water_switch	Watering switch.
alarm_control_panel	X_fall_alarm	Alarm status: --- Disarmed. The “TapLinker fall alert” (In LinkTap App) is disabled. ---Arming. The “TapLinker fall alert” is enabled, but not be triggered yet. --- Triggered. The “TapLinker fall alert” is enabled, and the water timer has fell from faucet.
	X_broken_alarm	Alarm status: --- Disarmed. The “Water shut-off failure alert” (In LinkTap App) is disabled. ---Arming. The “Water shut-off failure alert” is enabled, but not be triggered yet. --- Triggered. The “Water shut-off failure alert” is enabled, and the water timer was failed to shut-off.
	X_cutoff_alarm	Alarm status: --- Disarmed. The “Water cut-off or stop flowing alert” (In LinkTap App) is disabled. ---Arming. The “Water cut-off or stop flowing alert” is enabled, but not be triggered yet. --- Triggered. The “Water cut-off or stop flowing alert” is enabled, and “water cut-off” or “stop flowing” situation has happened.

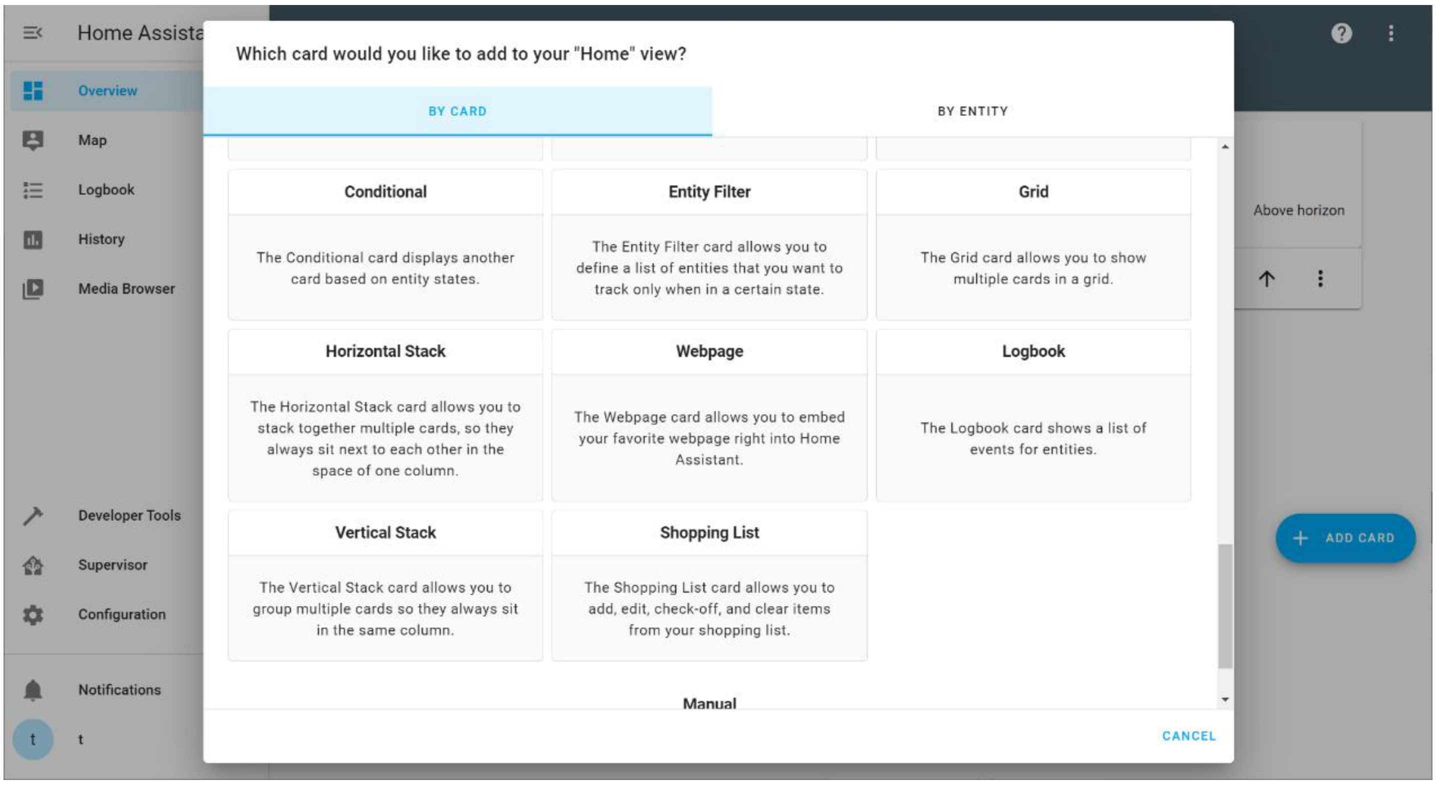
2.4. Watering Control Through Entity

2.4.1. Click “Overview” on the left sidebar, then “3-dot icon” on top-right corner, then “Edit Dashboard”, then click “ADD CARD”.





2.4.2. Choose your preferable layout, e.g., “Vertical Stack”.



Which card would you like to add to your "Home" view?

Conditional

The Conditional card displays another card based on entity states.

Entity Filter

The Entity Filter card allows you to define a list of entities that you want to track only when in a certain state.

Grid

The Grid card allows you to show multiple cards in a grid.

Horizontal Stack

The Horizontal Stack card allows you to stack together multiple cards, so they always sit next to each other in the space of one column.

Webpage

The Webpage card allows you to embed your favorite webpage right into Home Assistant.

Logbook

The Logbook card shows a list of events for entities.

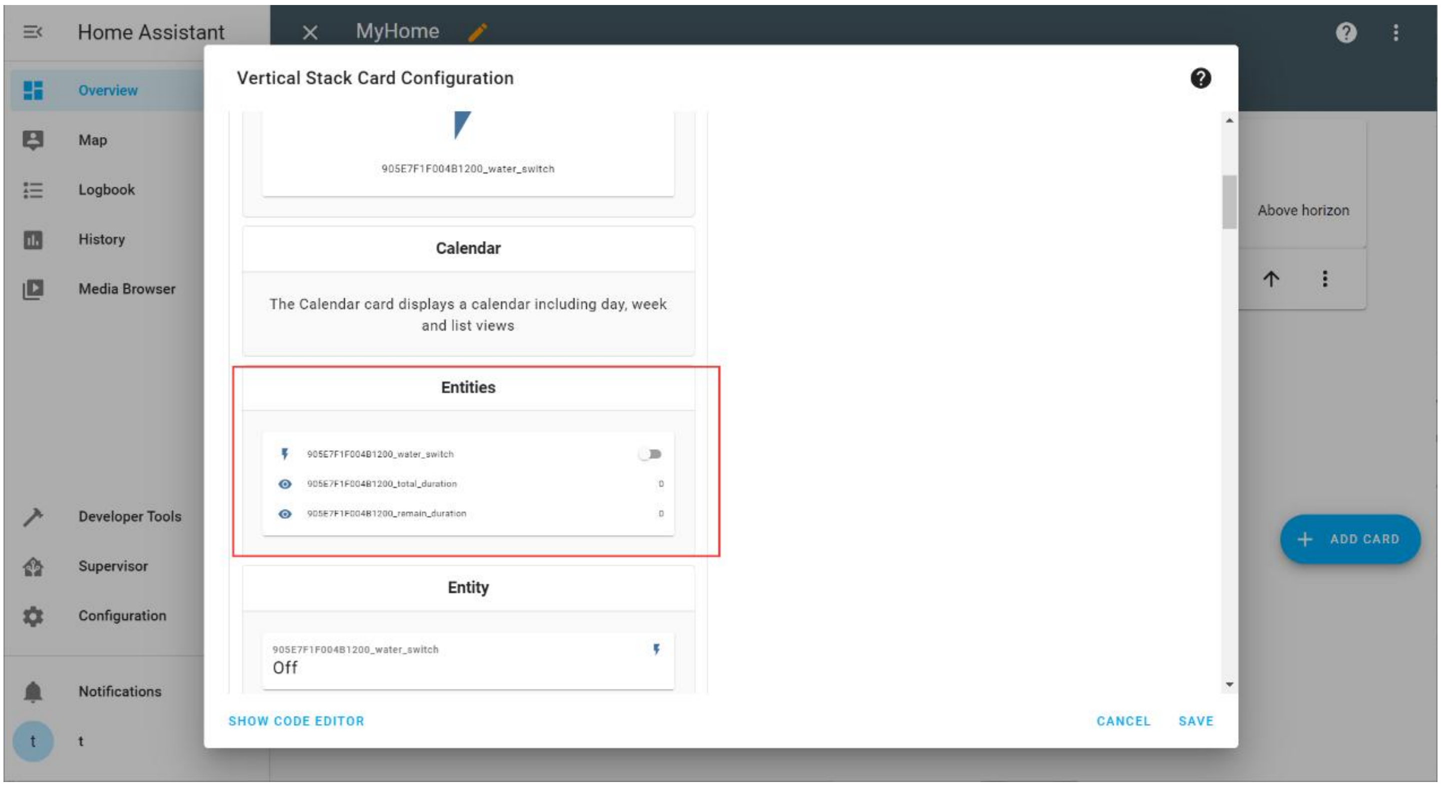
Vertical Stack

The Vertical Stack card allows you to group multiple cards so they always sit in the same column.

Shopping List

The Shopping List card allows you to add, edit, check-off, and clear items from your shopping list.

2.4.3. Configure the card, add LinkTap's entities into the card, then save. If the entity's value is "undefined", log into your Link app, go to Gateway details page, then click "Reset Gateway".



Vertical Stack Card Configuration

905E7F1F004B1200_water_switch

Calendar

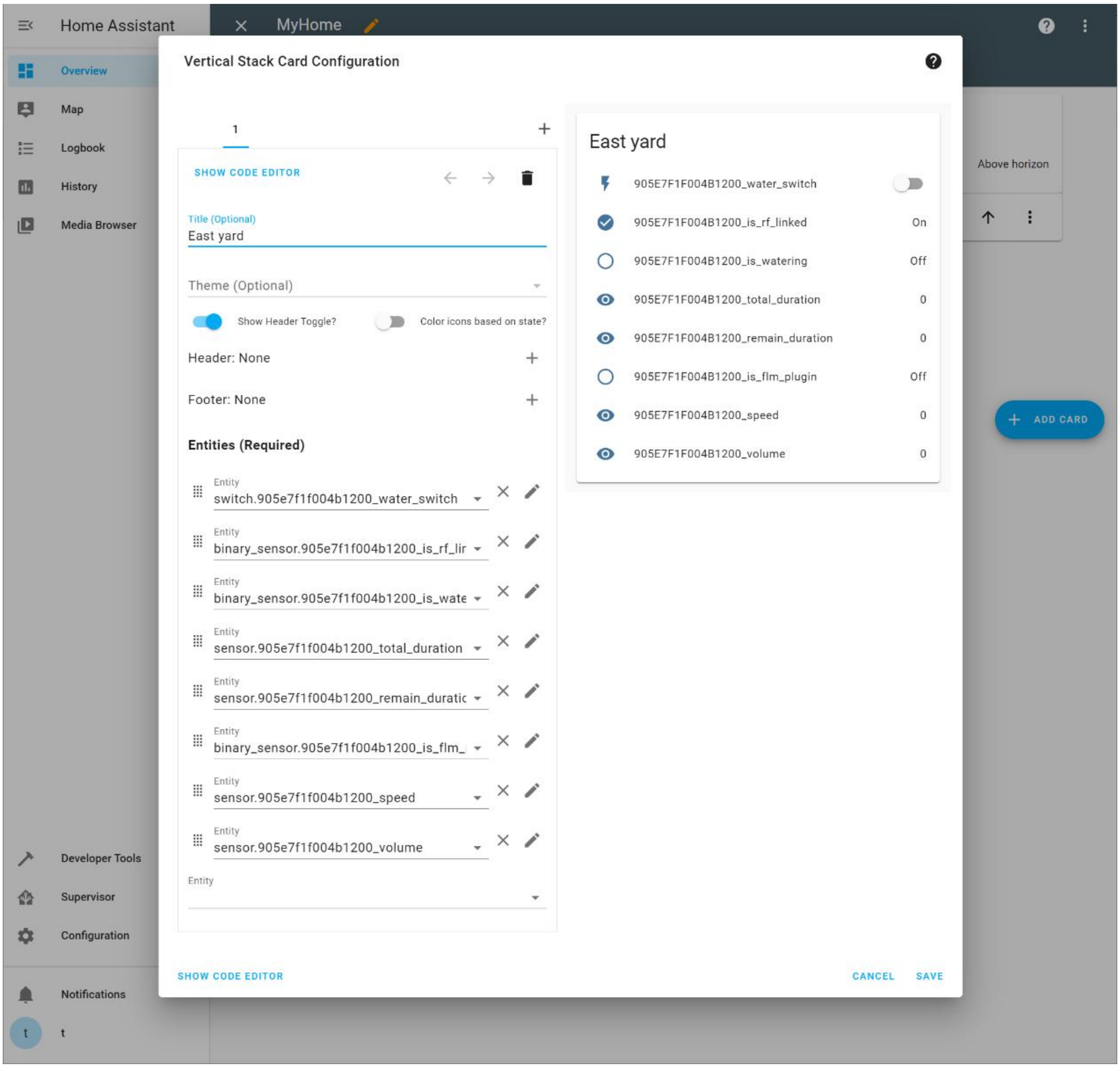
The Calendar card displays a calendar including day, week and list views

Entities

905E7F1F004B1200_water_switch	
905E7F1F004B1200_total_duration	0
905E7F1F004B1200_remain_duration	0

Entity

905E7F1F004B1200_water_switch
Off



Vertical Stack Card Configuration

1

Title (Optional)
East yard

Theme (Optional)

Show Header Toggle? Color icons based on state?

Header: None

Footer: None

Entities (Required)

Entity
switch.905e7f1f004b1200_water_switch

Entity
binary_sensor.905e7f1f004b1200_is_rf_lir

Entity
binary_sensor.905e7f1f004b1200_is_wate

Entity
sensor.905e7f1f004b1200_total_duration

Entity
sensor.905e7f1f004b1200_remain_duratic

Entity
binary_sensor.905e7f1f004b1200_is_flm_

Entity
sensor.905e7f1f004b1200_speed

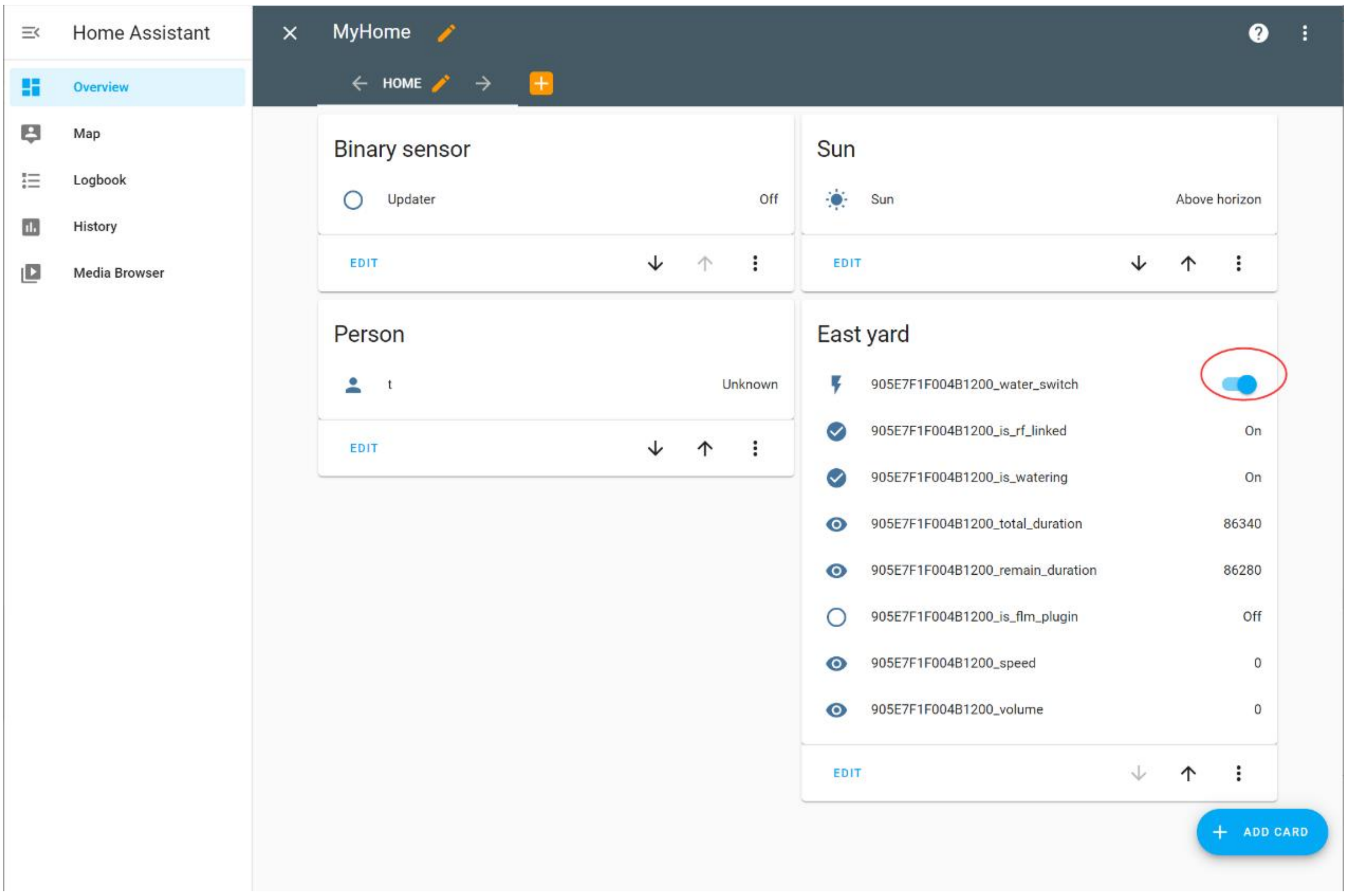
Entity
sensor.905e7f1f004b1200_volume

Entity

East yard

905E7F1F004B1200_water_switch	
905E7F1F004B1200_is_rf_linked	On
905E7F1F004B1200_is_watering	Off
905E7F1F004B1200_total_duration	0
905E7F1F004B1200_remain_duration	0
905E7F1F004B1200_is_flm_plugin	Off
905E7F1F004B1200_speed	0
905E7F1F004B1200_volume	0

2.4.4. Test watering control from Home Assistant. On the card, click the watering switch. Within 15 seconds, the water timer should start watering for 86340 seconds (1439 minutes), and the state of the watering switch will automatically switch to the "On" state. The status of other related entities will also change accordingly.



Binary sensor

Updater	Off

Sun

Sun	Above horizon

Person

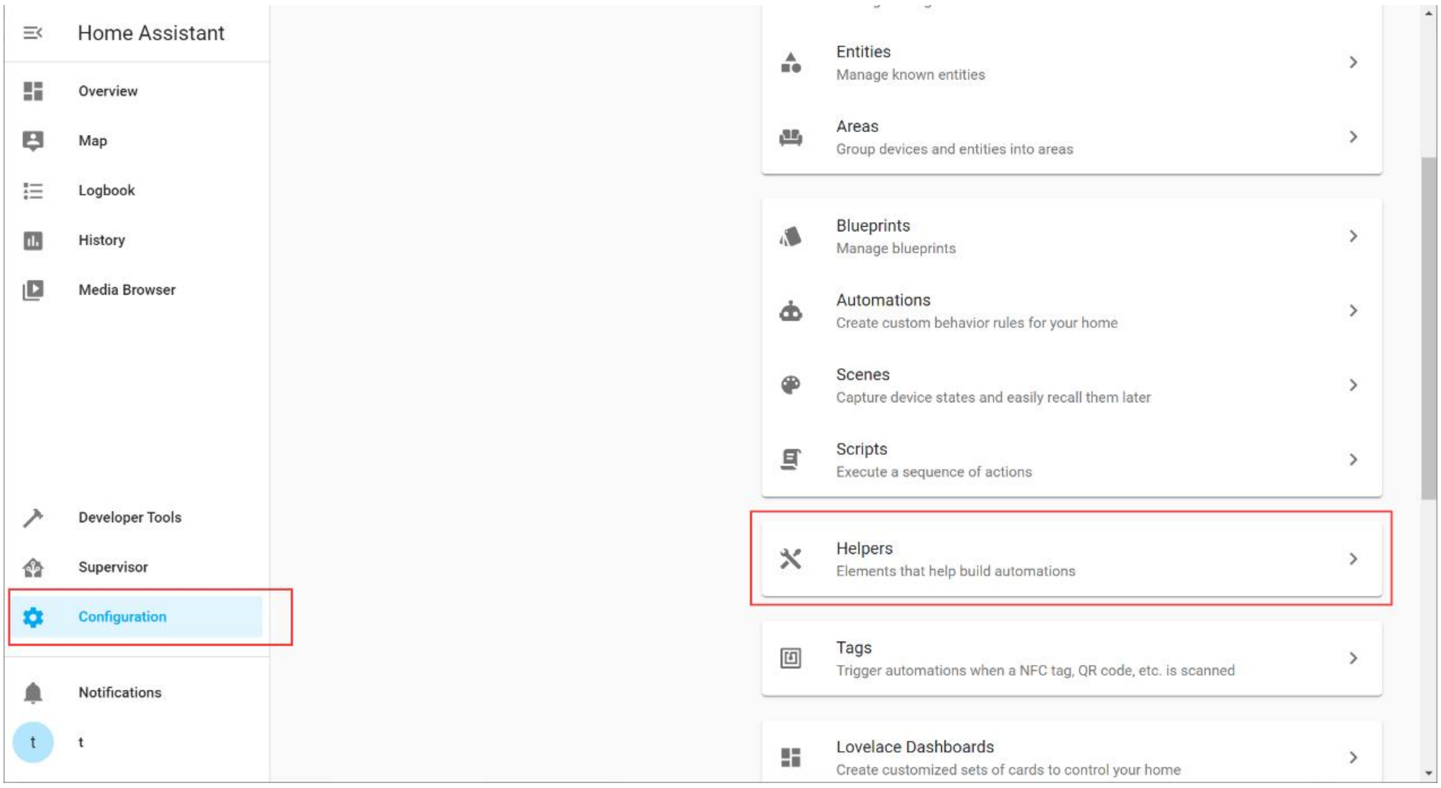
t	Unknown

East yard

905E7F1F004B1200_water_switch	
905E7F1F004B1200_is_rf_linked	On
905E7F1F004B1200_is_watering	On
905E7F1F004B1200_total_duration	86340
905E7F1F004B1200_remain_duration	86280
905E7F1F004B1200_is_flm_plugin	Off
905E7F1F004B1200_speed	0
905E7F1F004B1200_volume	0

2.4.5. Example: Control the watering time

1) As shown below, create two Helpers: One is the Text used to enter a watering duration. The other is a Timer used to shut off the water when the watering duration runs out.



Entities
Manage known entities

Areas
Group devices and entities into areas

Blueprints
Manage blueprints

Automations
Create custom behavior rules for your home

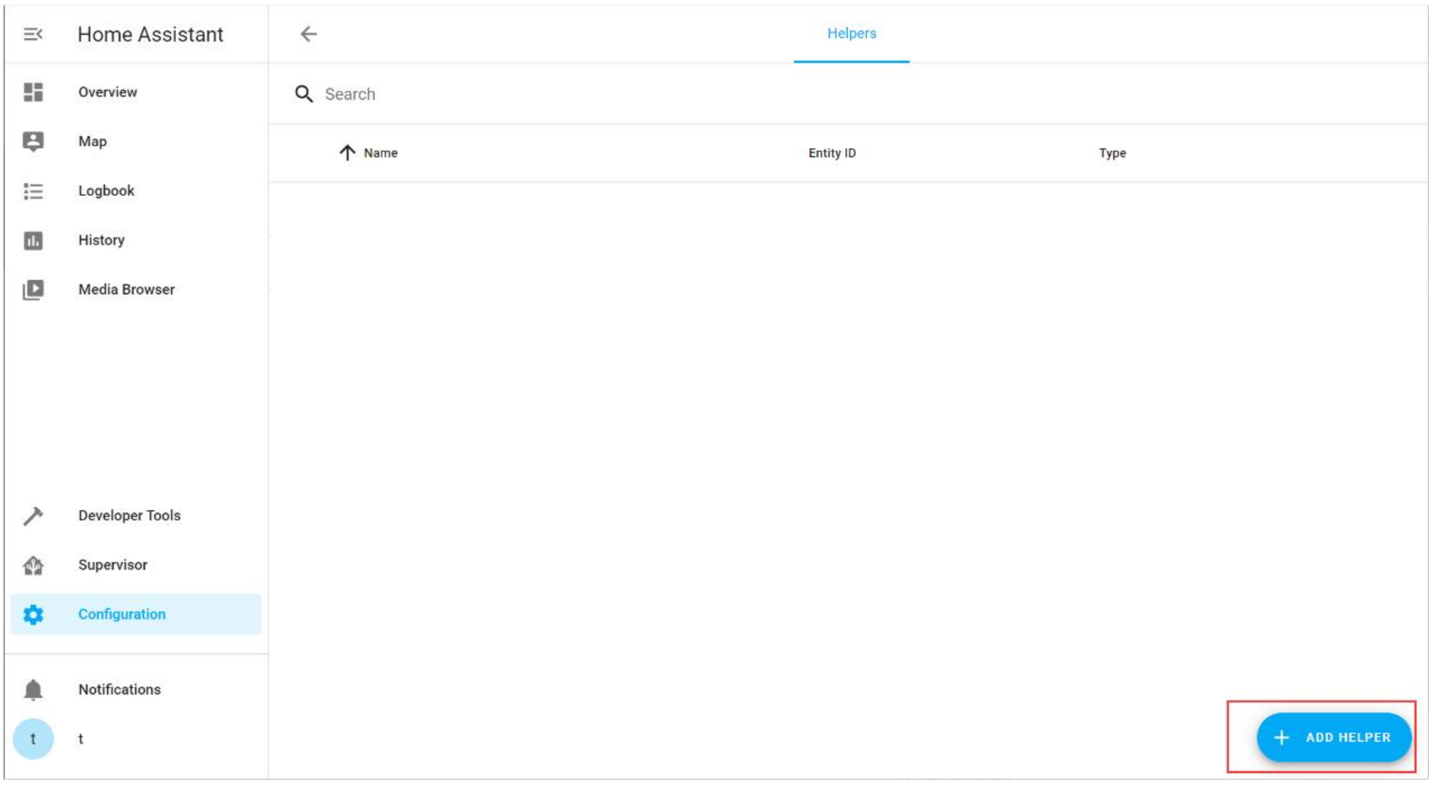
Scenes
Capture device states and easily recall them later

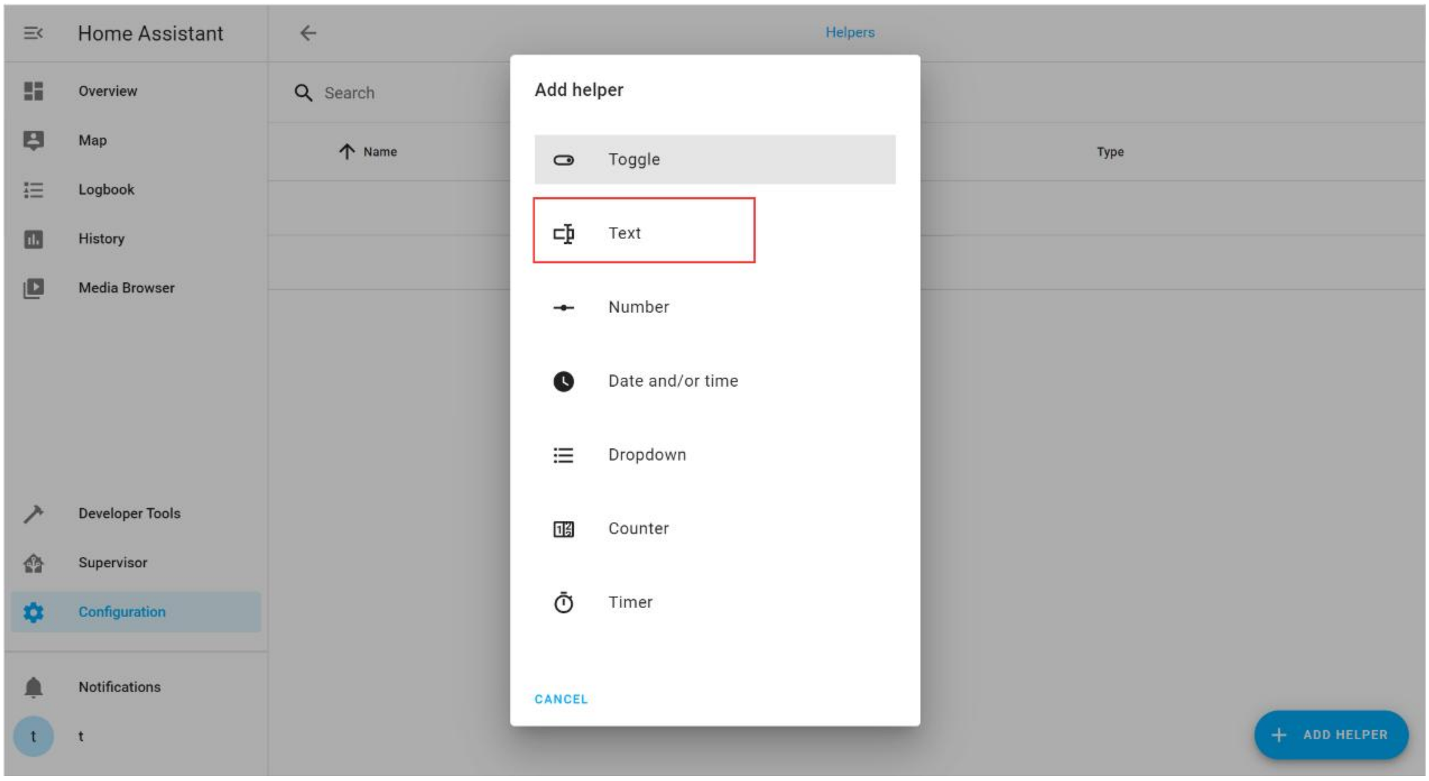
Scripts
Execute a sequence of actions

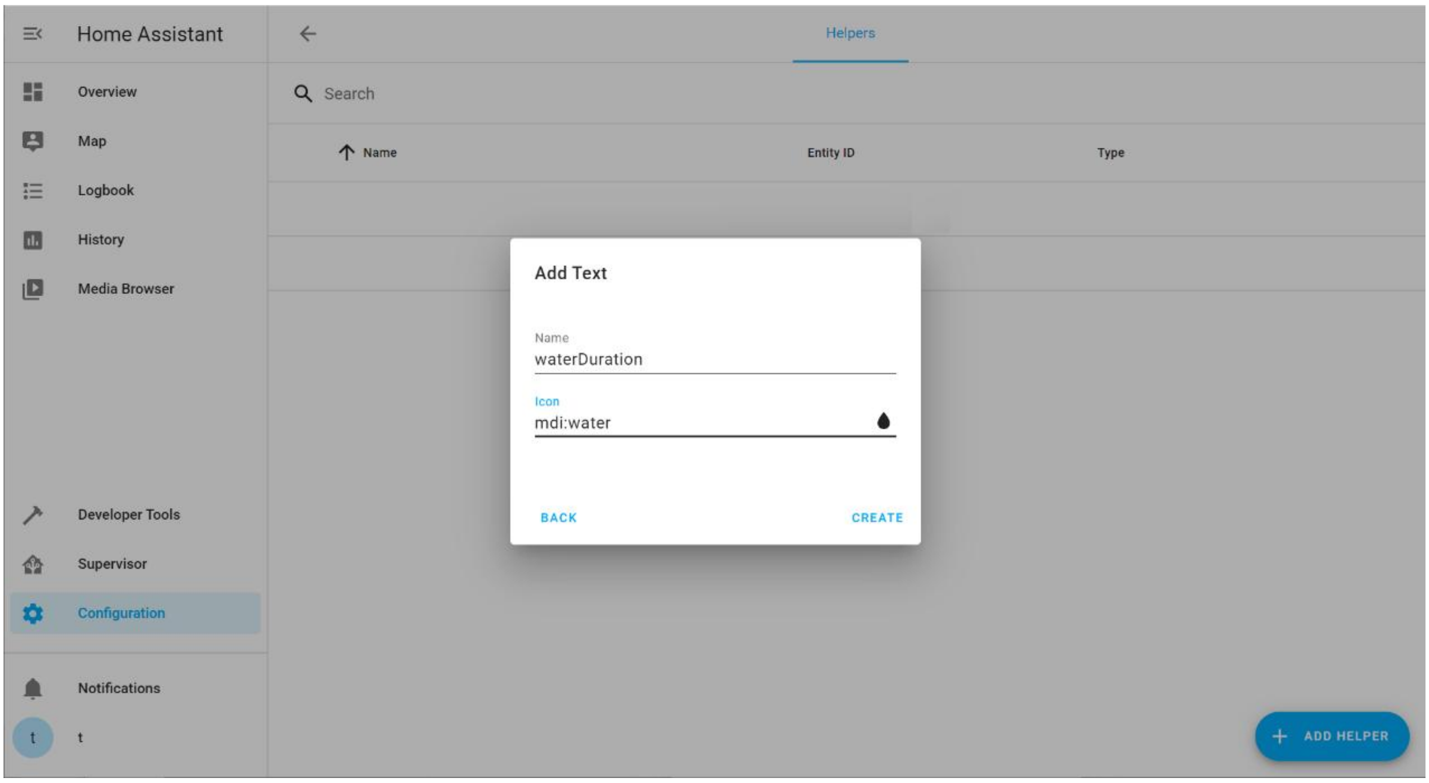
Helpers
Elements that help build automations

Tags
Trigger automations when a NFC tag, QR code, etc. is scanned

Lovelace Dashboards
Create customized sets of cards to control your home



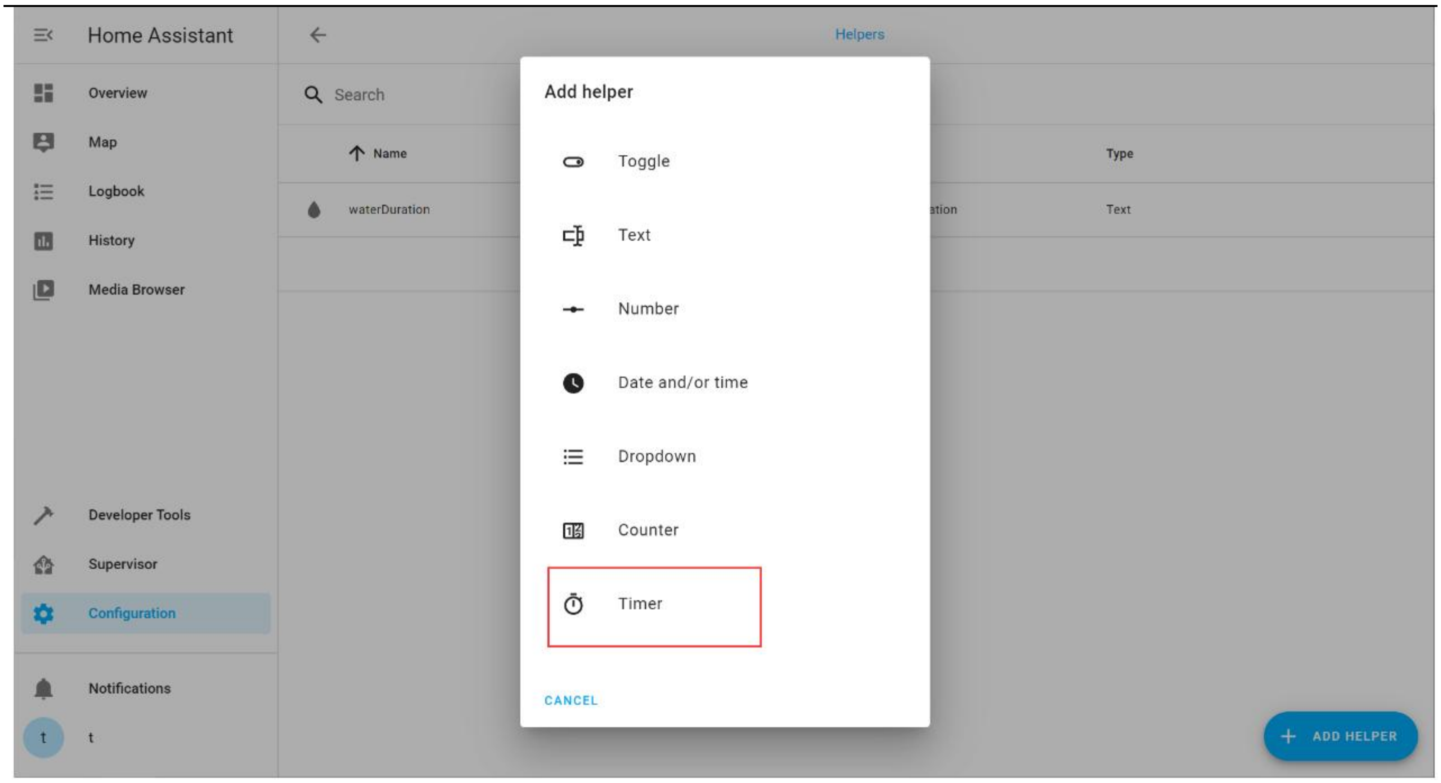


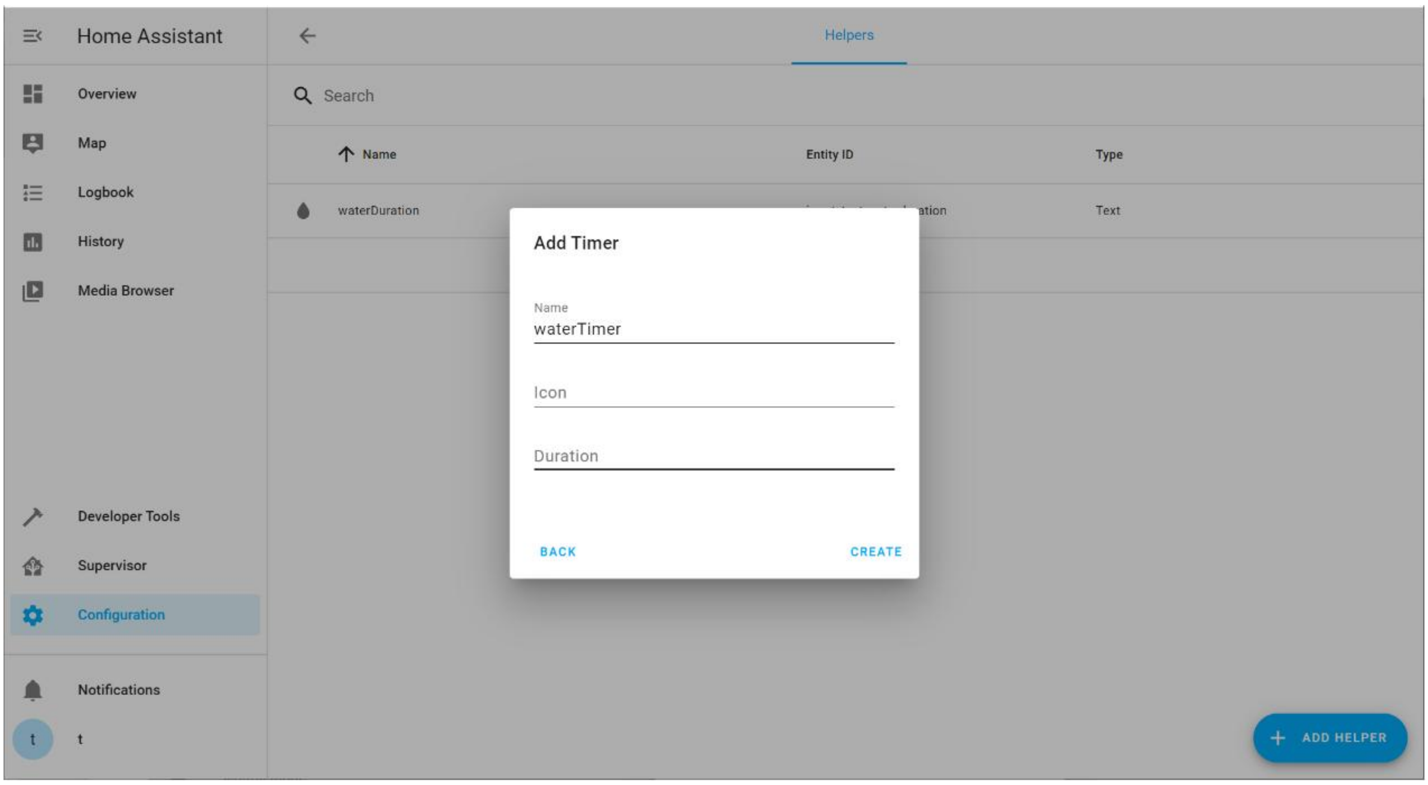


Add Text

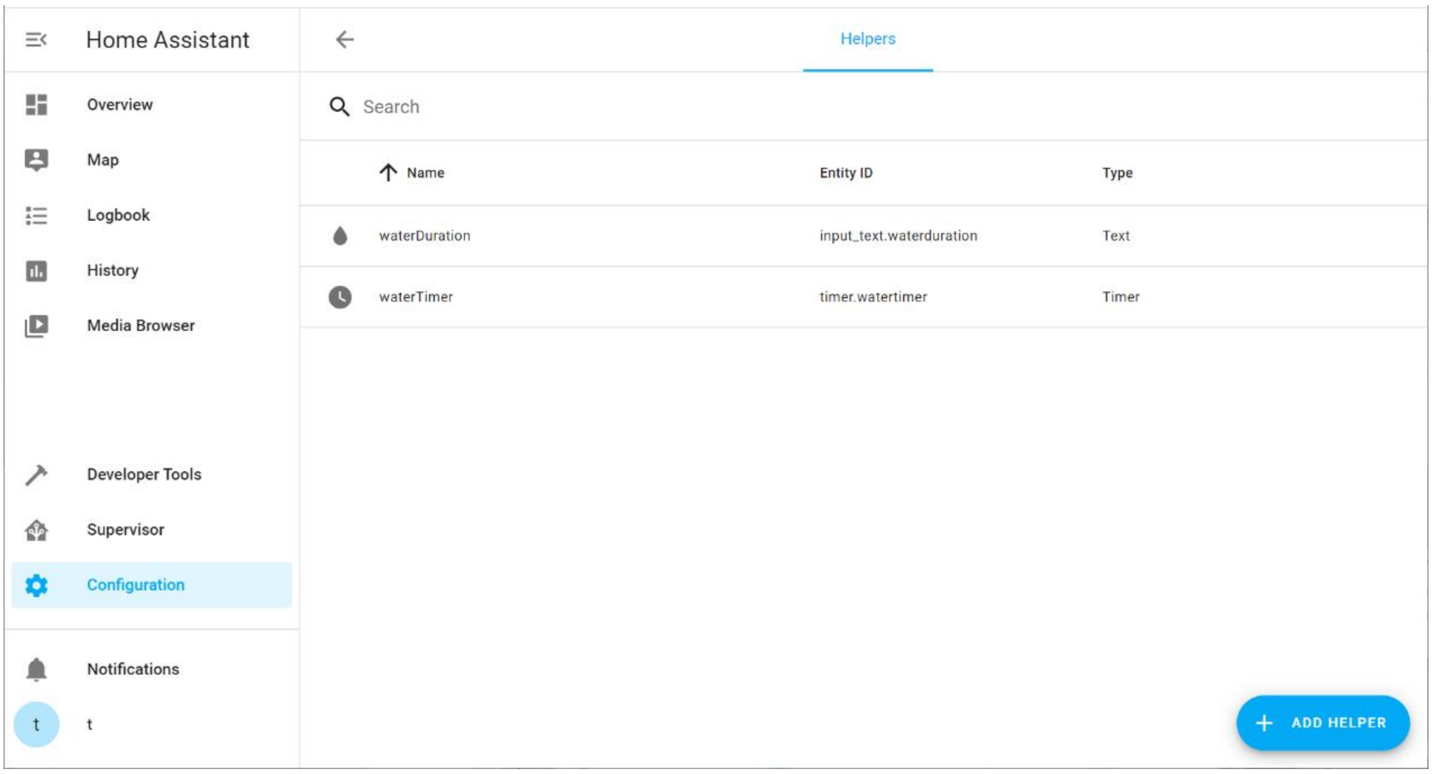
Name: waterDuration

Icon: mdi:water



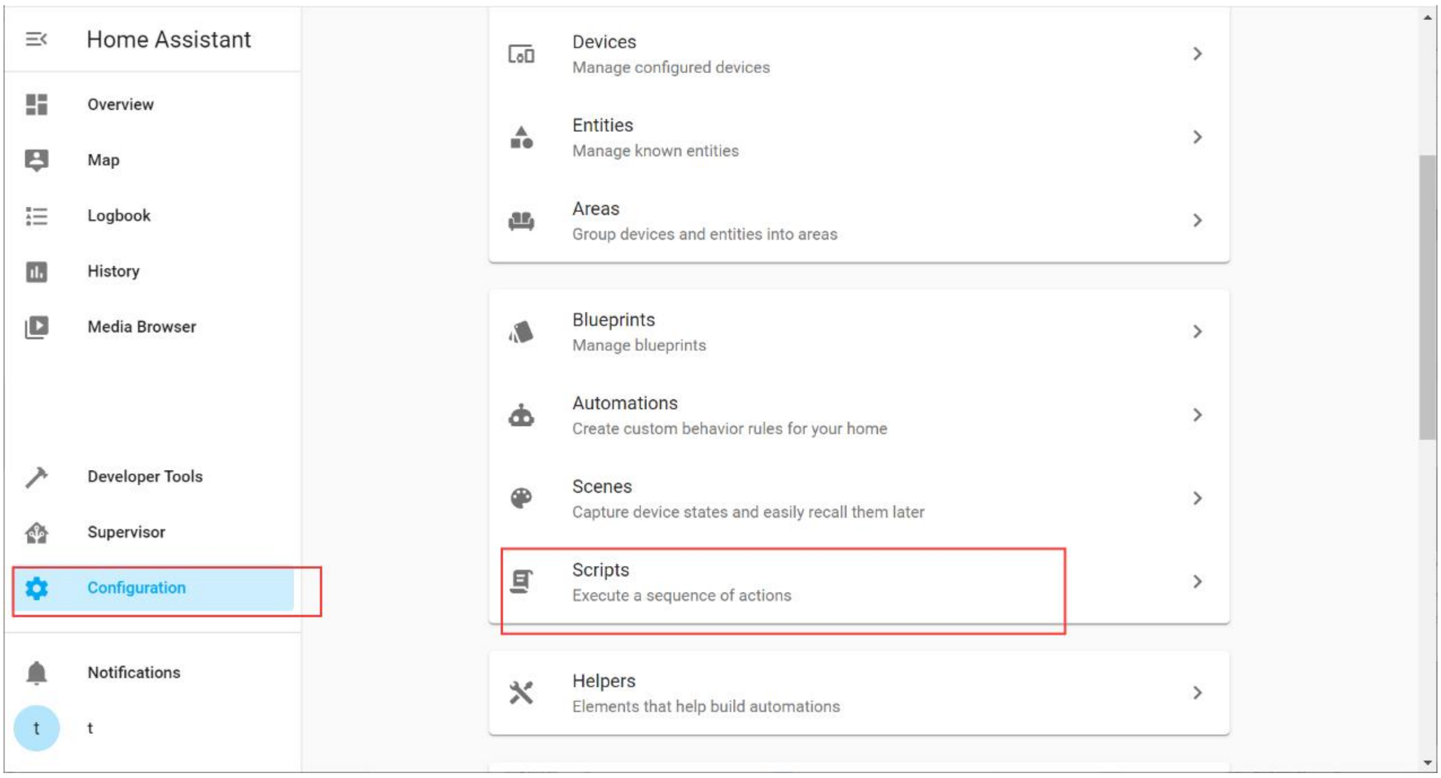


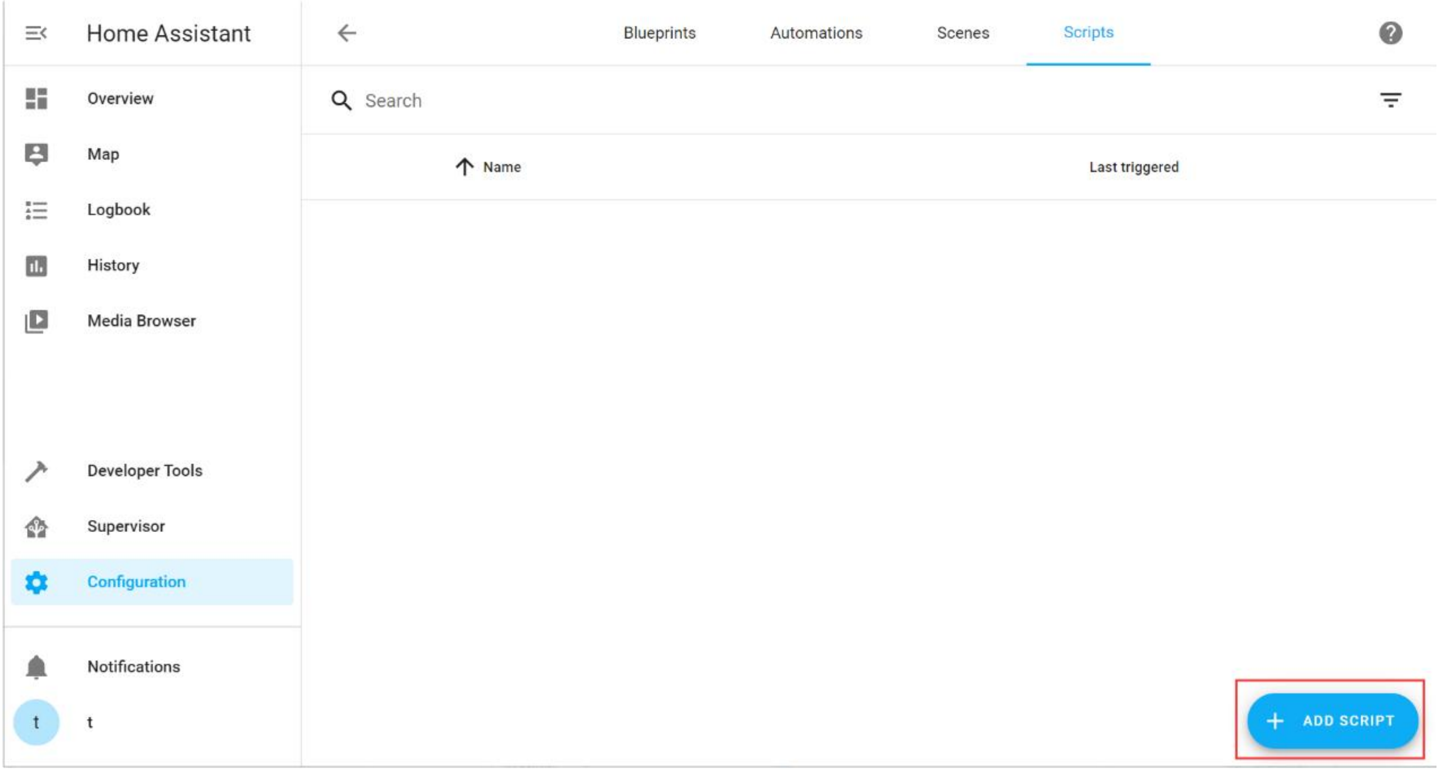
Name
waterTimer

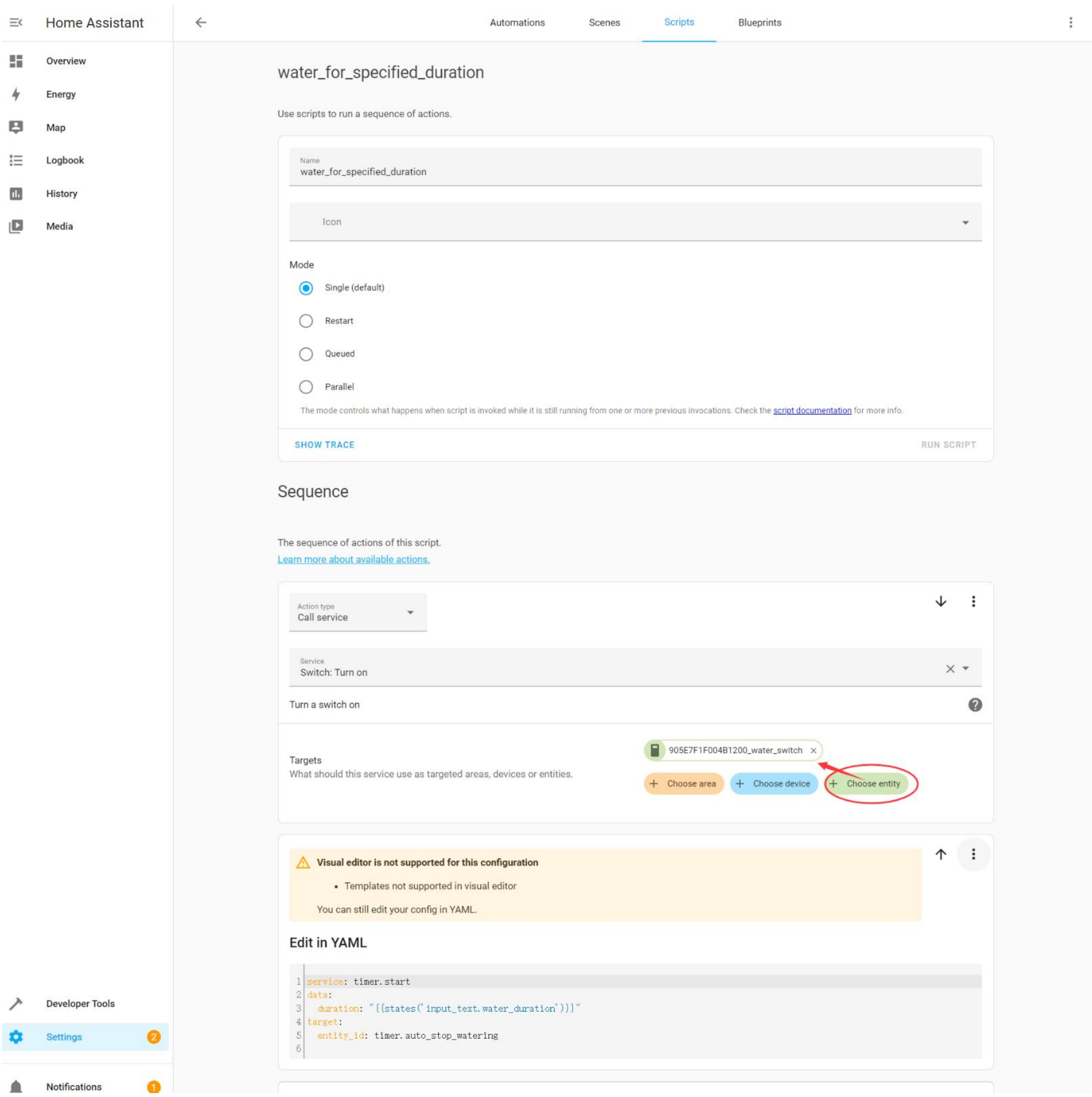


Name	Entity ID	Type
waterDuration	input_text.waterduration	Text
waterTimer	timer.watertimer	Timer

2) Then create a Script which is used to assign the watering duration to the Timer, and enable watering switch.


Home Assistant
Overview
Map
Logbook
History
Media Browser
Developer Tools
Supervisor
Configuration
Notifications
t
Devices
Manage configured devices
Entities
Manage known entities
Areas
Group devices and entities into areas
Blueprints
Manage blueprints
Automations
Create custom behavior rules for your home
Scenes
Capture device states and easily recall them later
Scripts
Execute a sequence of actions
Helpers
Elements that help build automations





water_for_specified_duration

Use scripts to run a sequence of actions.

Name: water_for_specified_duration

Icon

Mode

● Single (default)

○ Restart

○ Queued

○ Parallel

The mode controls what happens when script is invoked while it is still running from one or more previous invocations. Check the script documentation for more info.

Sequence

The sequence of actions of this script.
Learn more about available actions.

Action type: Call service

Service: Switch: Turn on

Turn a switch on

Targets
What should this service use as targeted areas, devices or entities.

905E7F1F004B1200_water_switch ×

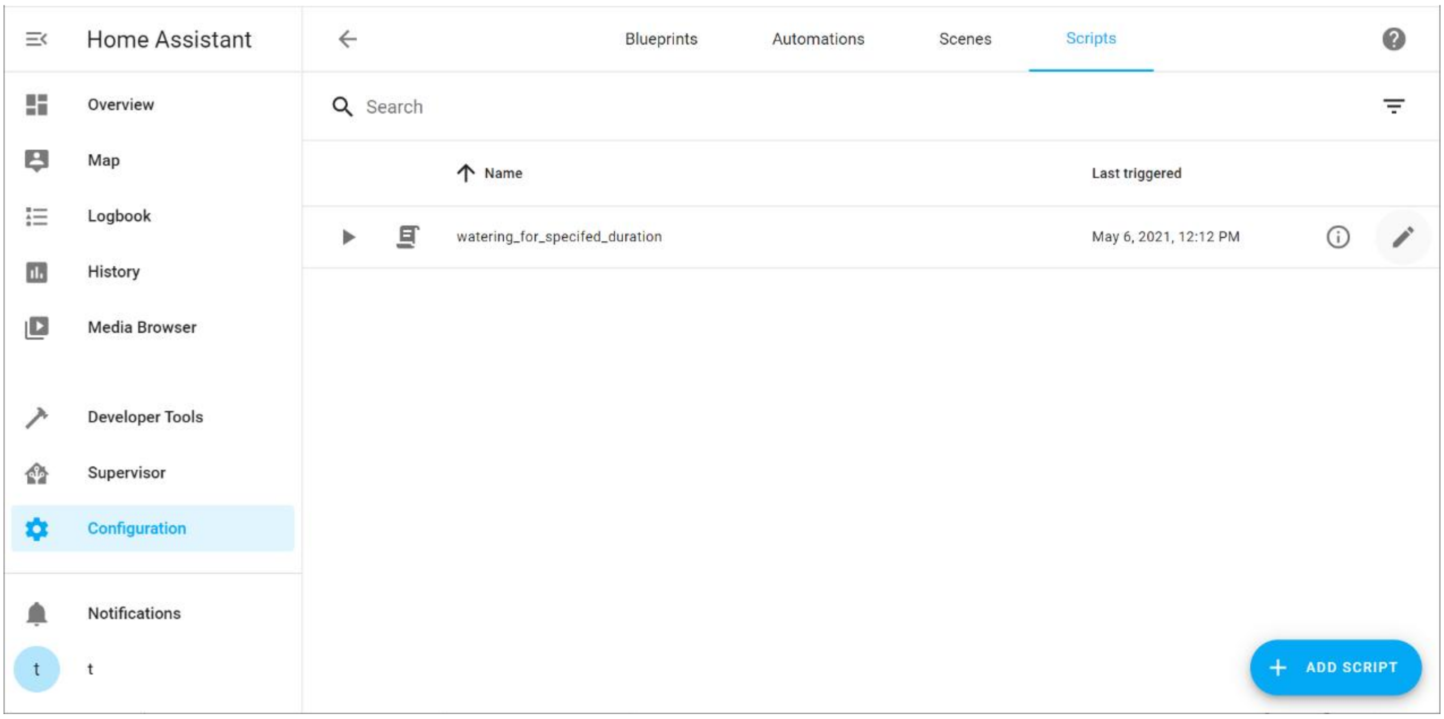
Visual editor is not supported for this configuration

- Templates not supported in visual editor

You can still edit your config in YAML.

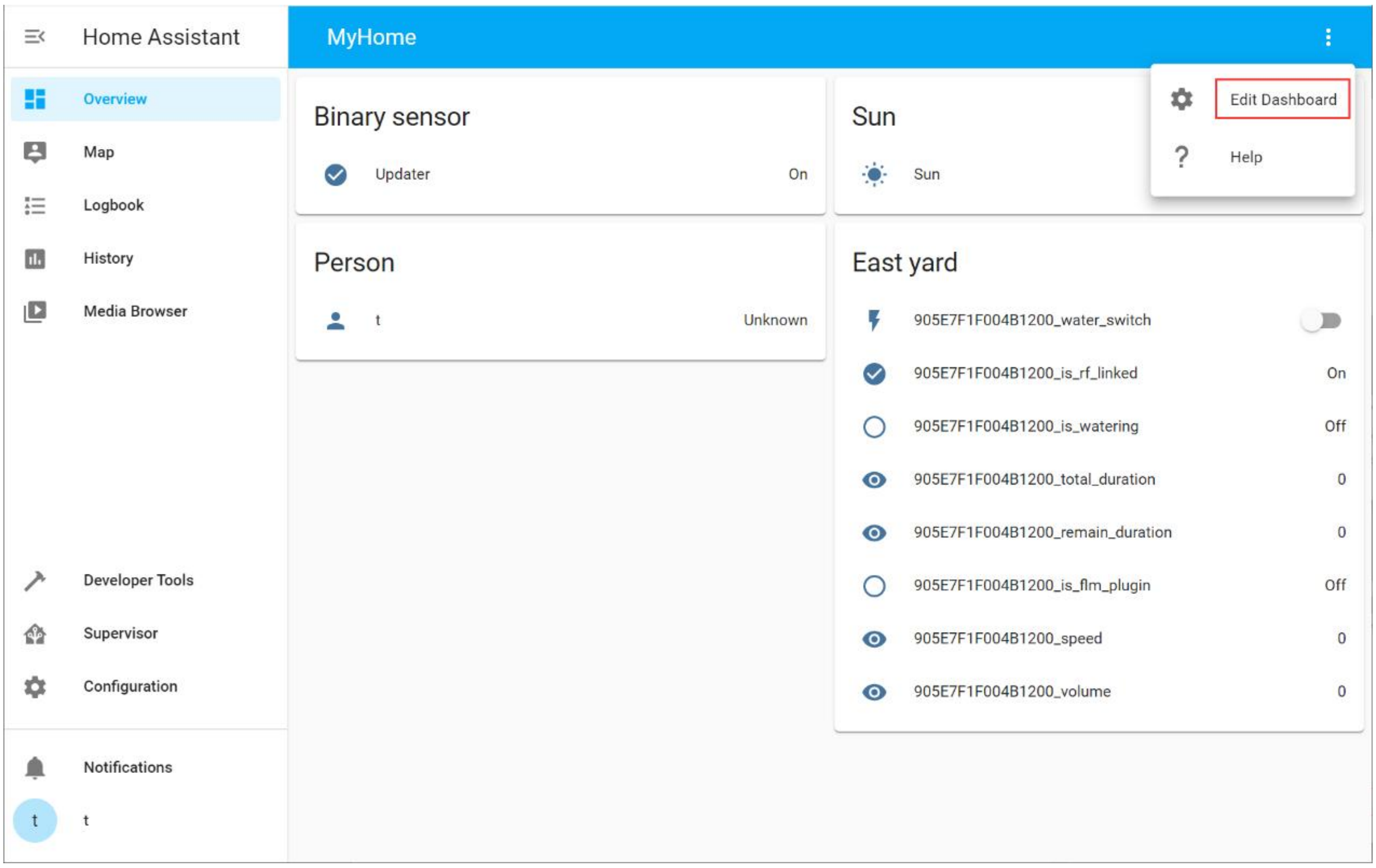
Edit in YAML

```
service: timer.start
data:
  duration: ”{{states(’input_text.water_duration’)}}”
target:
  entity_id: timer.auto_stop_watering
```



Name	Last triggered
watering_for_specifed_duration	May 6, 2021, 12:12 PM

3) On the “Overview” page, click “Edit Dashboard” as shown below.”



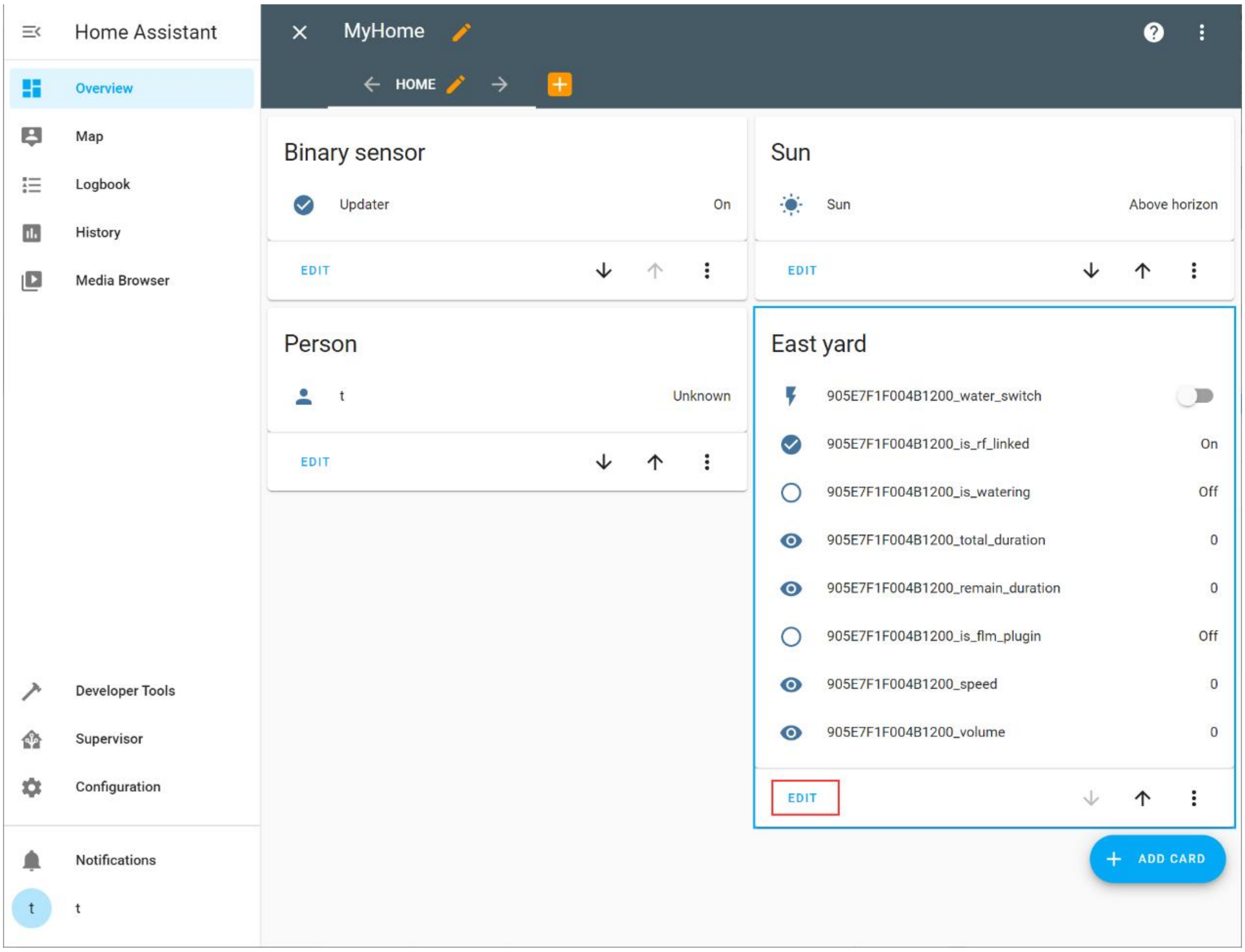
Binary sensor
Updater On

Sun
Sun

Person
t Unknown

East yard
905E7F1F004B1200_water_switch
905E7F1F004B1200_is_rf_linked On
905E7F1F004B1200_is_watering Off
905E7F1F004B1200_total_duration 0
905E7F1F004B1200_remain_duration 0
905E7F1F004B1200_is_flm_plugin Off
905E7F1F004B1200_speed 0
905E7F1F004B1200_volume 0

Click ”EDIT” as shown below.



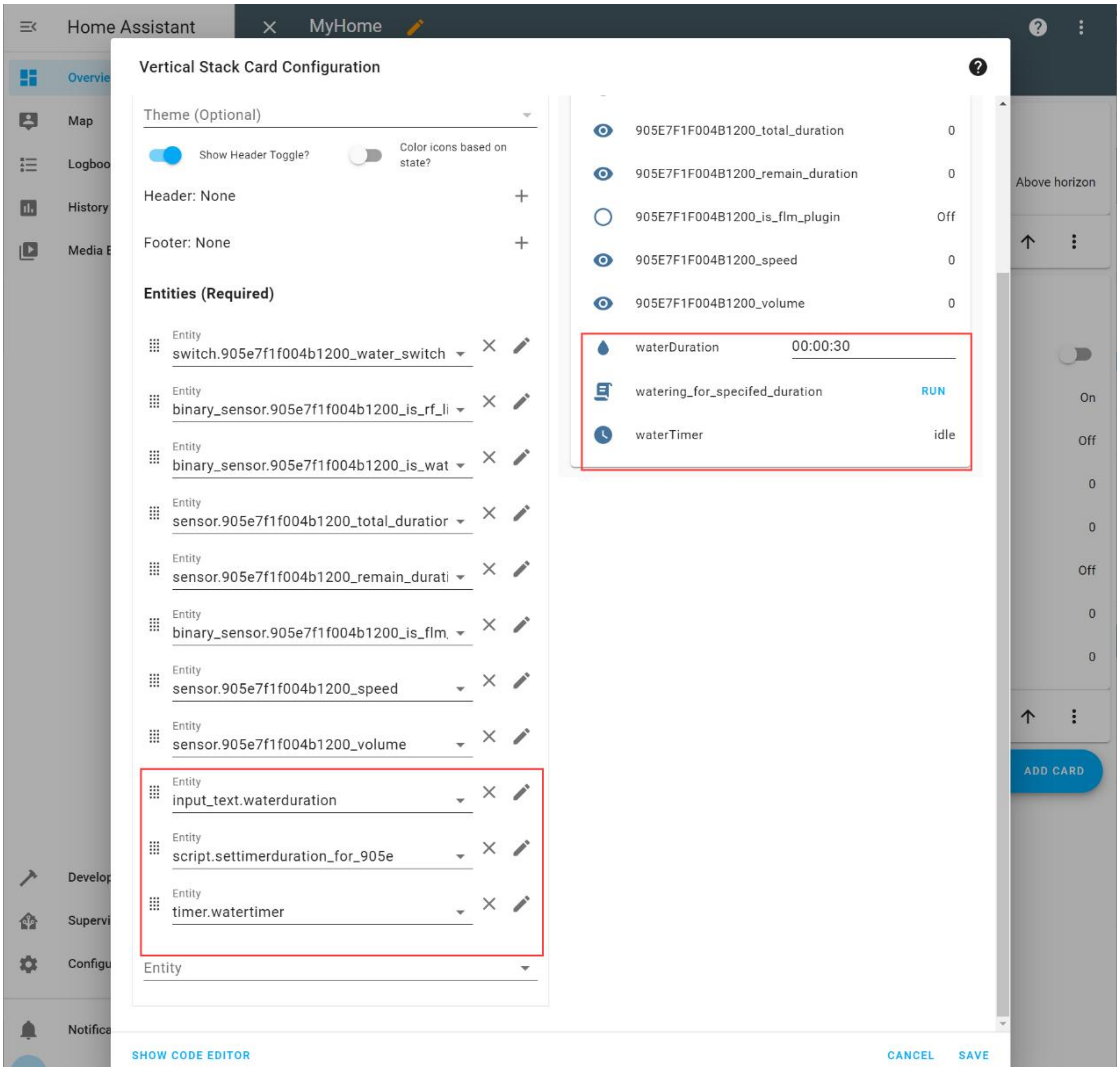
Binary sensor
Updater On

Sun
Sun Above horizon

Person
t Unknown

East yard
905E7F1F004B1200_water_switch
905E7F1F004B1200_is_rf_linked On
905E7F1F004B1200_is_watering Off
905E7F1F004B1200_total_duration 0
905E7F1F004B1200_remain_duration 0
905E7F1F004B1200_is_flm_plugin Off
905E7F1F004B1200_speed 0
905E7F1F004B1200_volume 0

Add the Helpers and script created in the previous steps. Click “SAVE”.



Vertical Stack Card Configuration

Theme (Optional)

Show Header Toggle? Color icons based on state?

Header: None

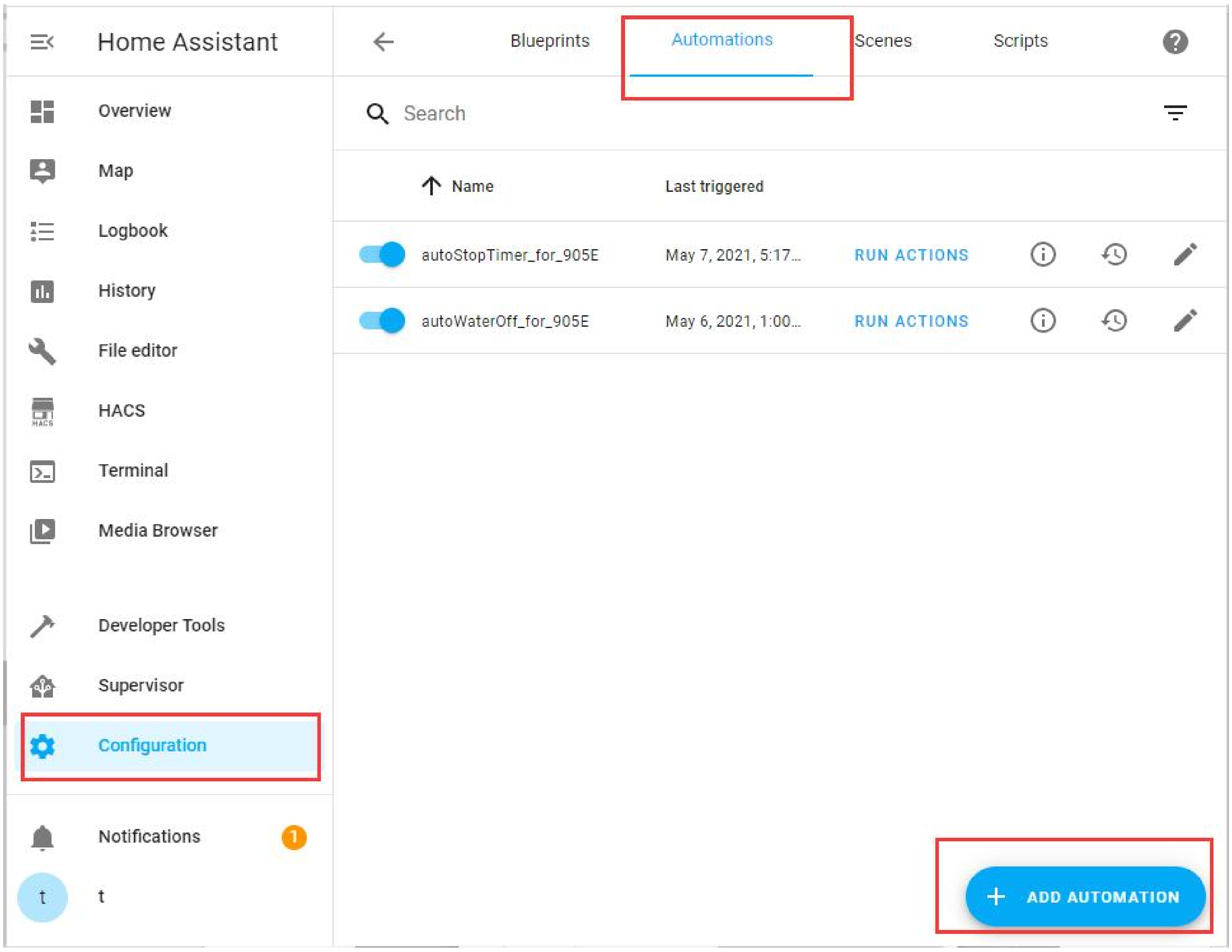
Footer: None

Entities (Required)

Entity: switch.905e7f1f004b1200_water_switch
Entity: binary_sensor.905e7f1f004b1200_is_rf_li
Entity: binary_sensor.905e7f1f004b1200_is_wat
Entity: sensor.905e7f1f004b1200_total_duratior
Entity: sensor.905e7f1f004b1200_remain_durati
Entity: binary_sensor.905e7f1f004b1200_is_flm,
Entity: sensor.905e7f1f004b1200_speed
Entity: sensor.905e7f1f004b1200_volume
Entity: input_text.waterduration
Entity: script.settimerduration_for_905e
Entity: timer.watertimer
Entity

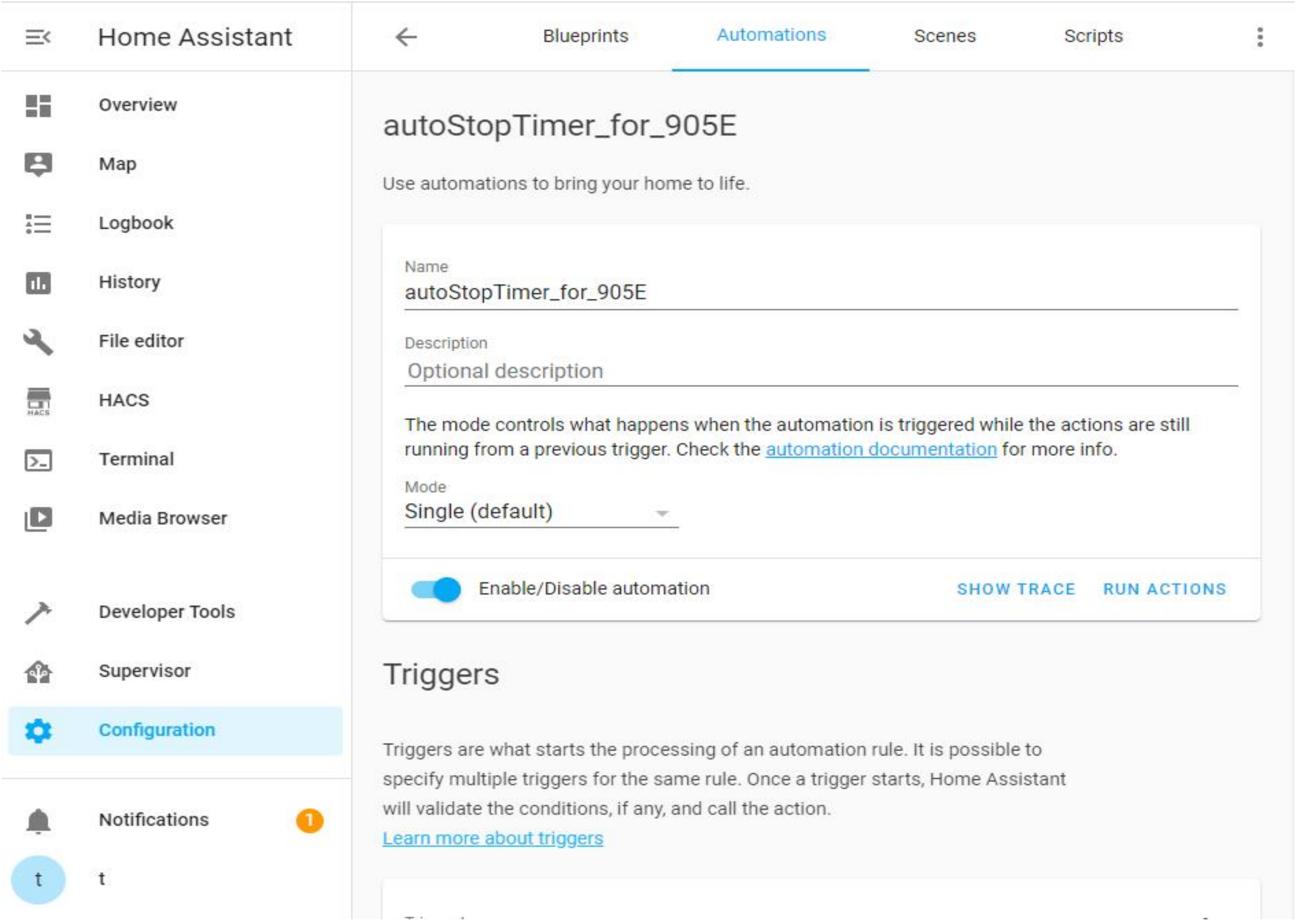
905E7F1F004B1200_total_duration	0
905E7F1F004B1200_remain_duration	0
905E7F1F004B1200_is_flm_plugin	Off
905E7F1F004B1200_speed	0
905E7F1F004B1200_volume	0
waterDuration	00:00:30
watering_for_specifed_duration	RUN
waterTimer	idle

4) On the “Configuration” page, add two ”Automations” for A) Auto stop timer if the watering is terminated before the timer runs out, and B) Auto stop watering if watering still continues after the timer runs out.



Name	Last triggered
autoStopTimer_for_905E	May 7, 2021, 5:17...
autoWaterOff_for_905E	May 6, 2021, 1:00...

The first Automation: autoStopTimer_for_905E



autoStopTimer_for_905E

Use automations to bring your home to life.

Name
autoStopTimer_for_905E

Description
Optional description

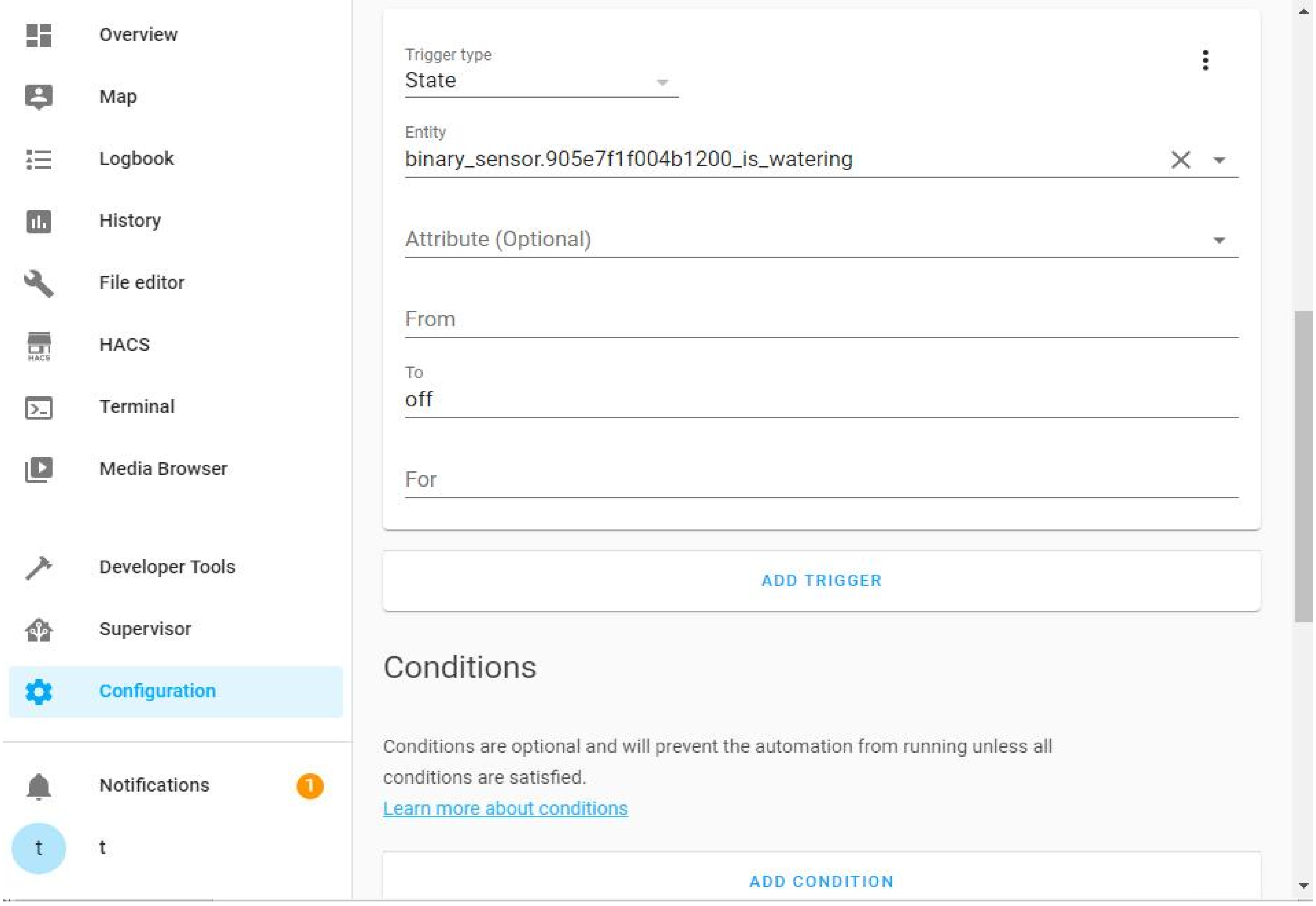
The mode controls what happens when the automation is triggered while the actions are still running from a previous trigger. Check the automation documentation for more info.

Mode
Single (default)

Enable/Disable automation

Triggers

Triggers are what starts the processing of an automation rule. It is possible to specify multiple triggers for the same rule. Once a trigger starts, Home Assistant will validate the conditions, if any, and call the action.
Learn more about triggers



Trigger type
State

Entity
binary_sensor.905e7f1f004b1200_is_watering

Attribute (Optional)

From

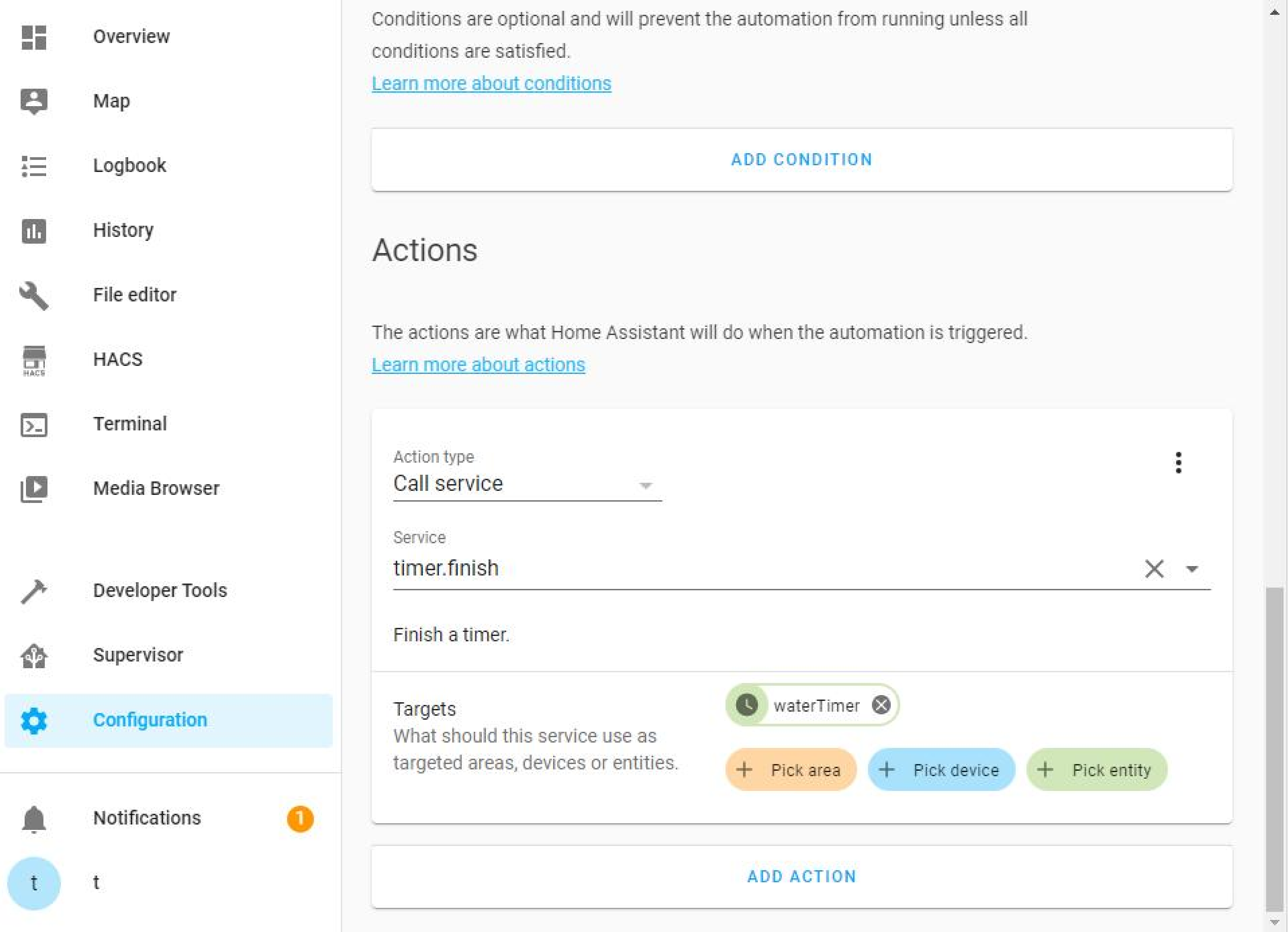
To
off

For

Conditions

Conditions are optional and will prevent the automation from running unless all conditions are satisfied.

Learn more about conditions



Conditions are optional and will prevent the automation from running unless all conditions are satisfied.

Learn more about conditions

Actions

The actions are what Home Assistant will do when the automation is triggered.

Learn more about actions

Action type
Call service

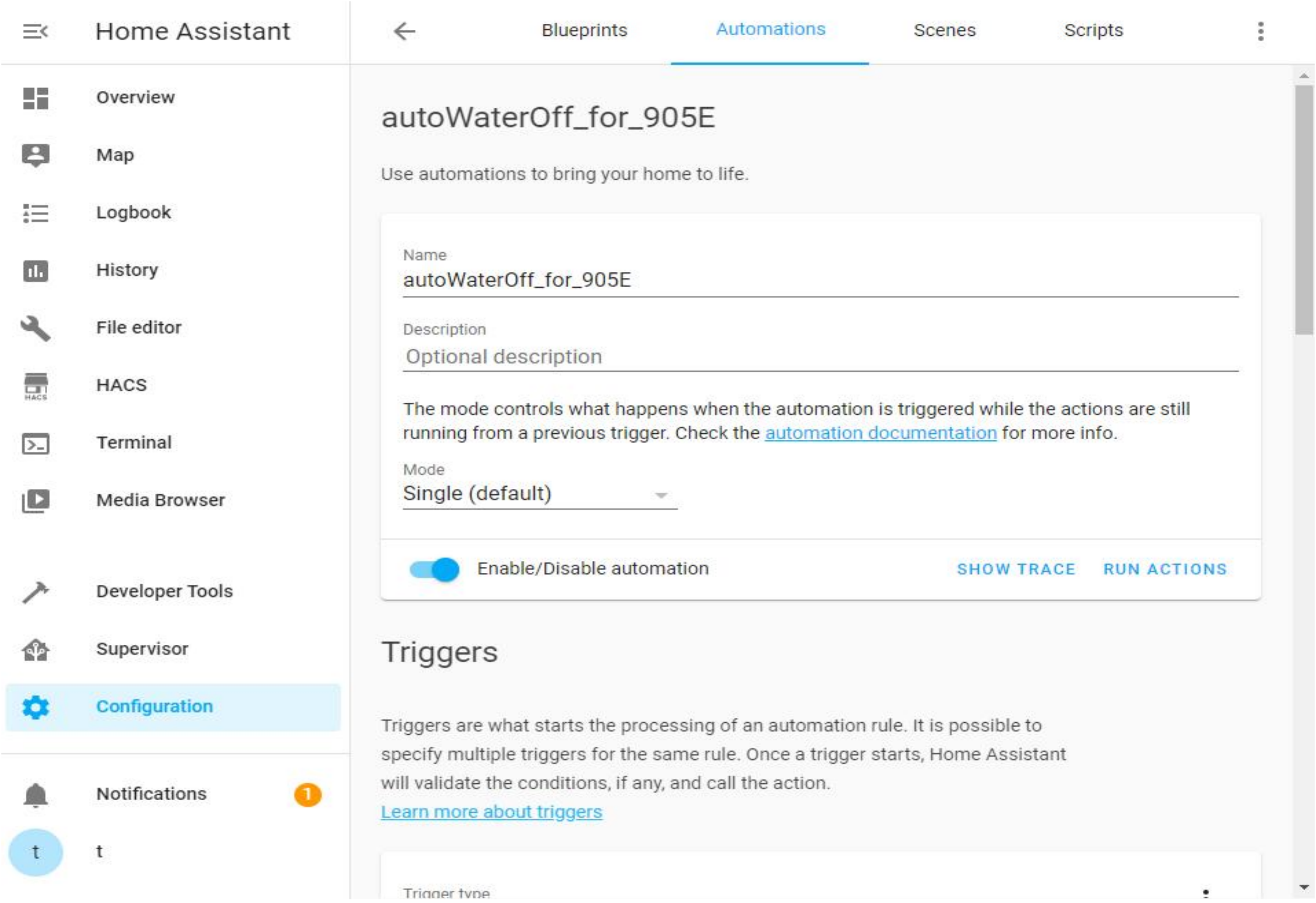
Service
timer.finish

Finish a timer.

Targets
What should this service use as targeted areas, devices or entities.

waterTimer

The second Automation: autoWaterOff_for_905E



autoWaterOff_for_905E

Use automations to bring your home to life.

Name: autoWaterOff_for_905E

Description: Optional description

The mode controls what happens when the automation is triggered while the actions are still running from a previous trigger. Check the automation documentation for more info.

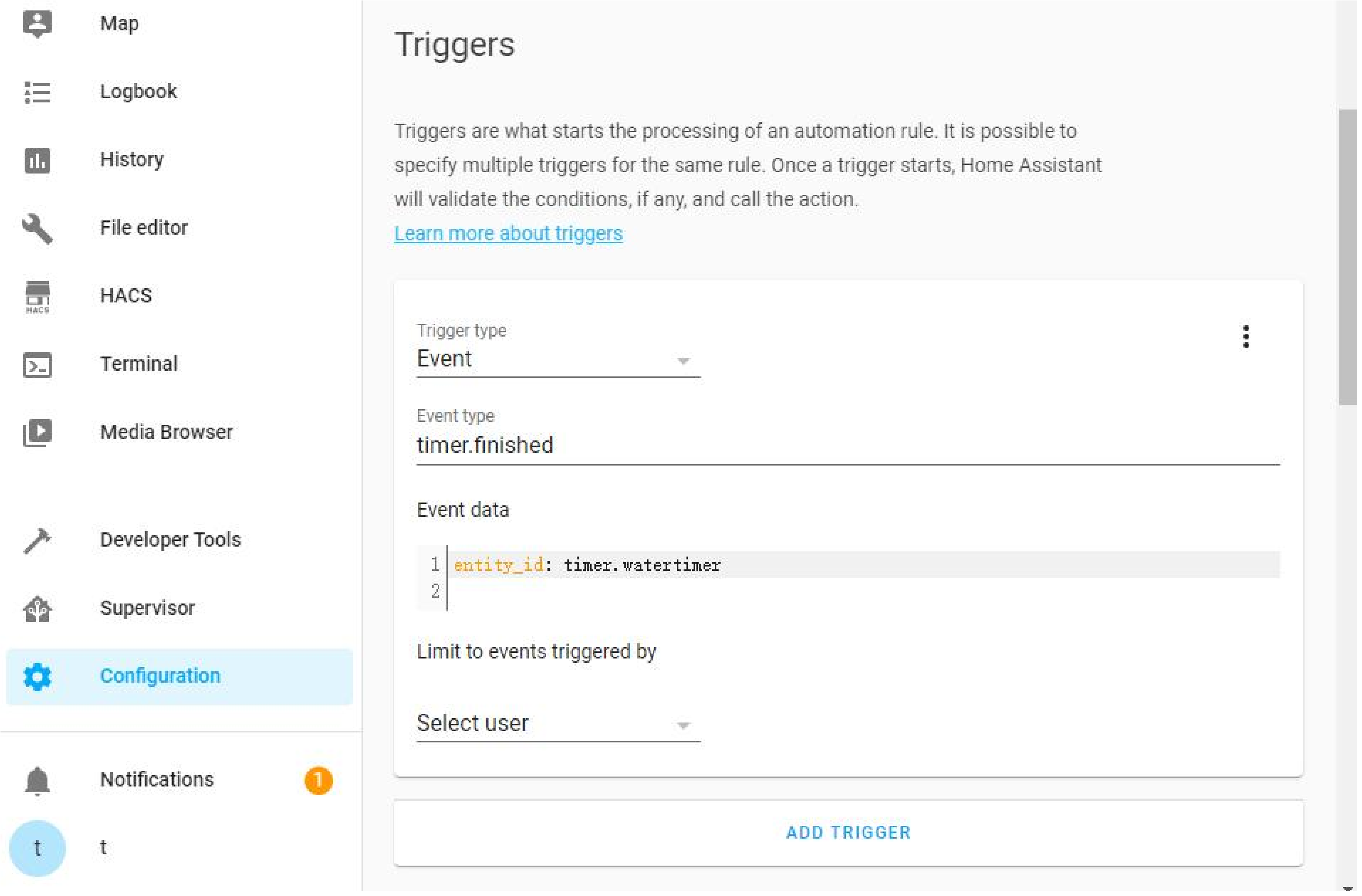
Mode: Single (default)

Enable/Disable automation

Triggers

Triggers are what starts the processing of an automation rule. It is possible to specify multiple triggers for the same rule. Once a trigger starts, Home Assistant will validate the conditions, if any, and call the action.
Learn more about triggers

Trigger type



Triggers

Triggers are what starts the processing of an automation rule. It is possible to specify multiple triggers for the same rule. Once a trigger starts, Home Assistant will validate the conditions, if any, and call the action.
Learn more about triggers

Trigger type: Event

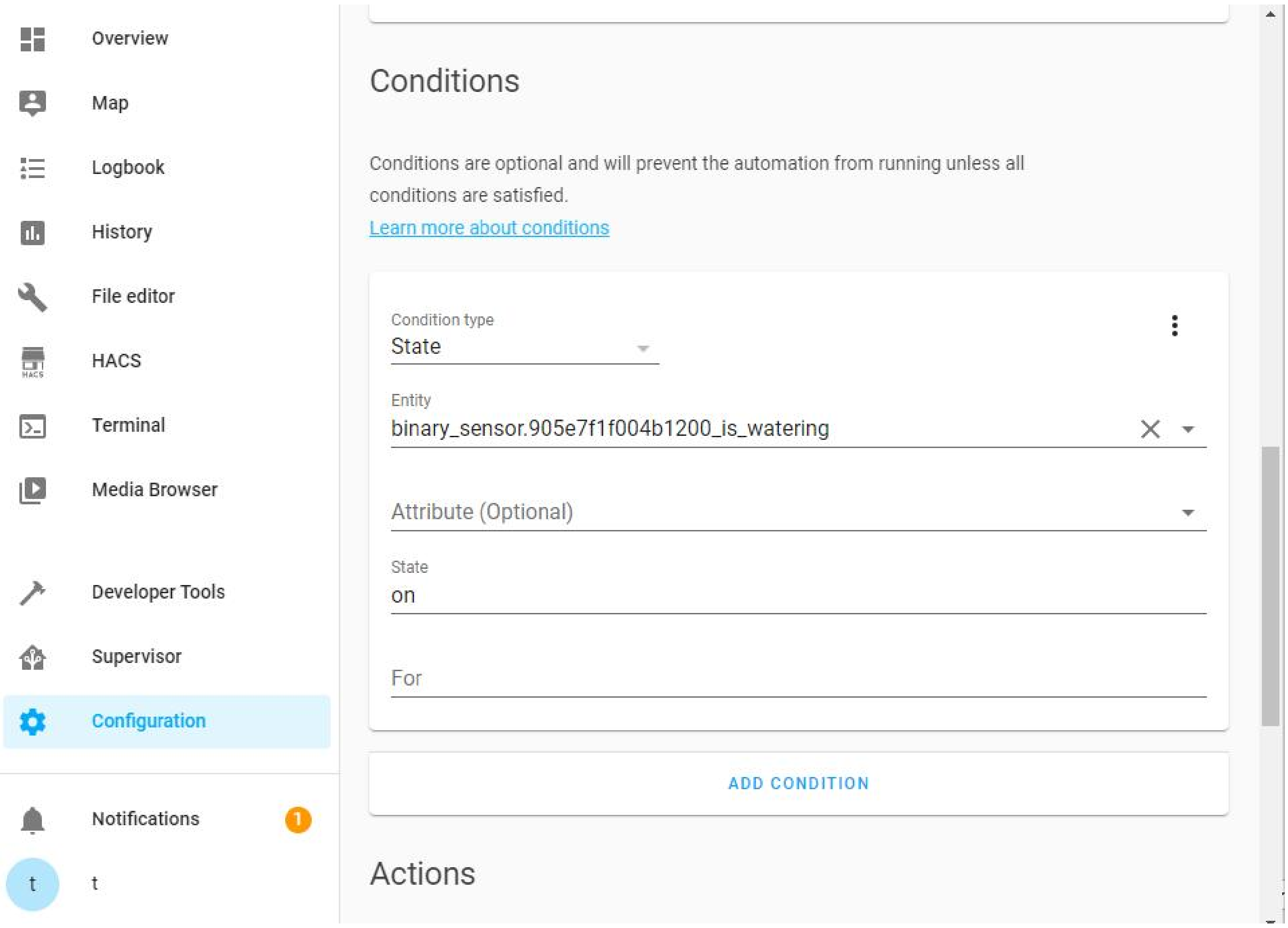
Event type: timer.finished

Event data

```
entity_id: timer.watertimer
```

Limit to events triggered by

Select user



Conditions

Conditions are optional and will prevent the automation from running unless all conditions are satisfied.
Learn more about conditions

Condition type
State

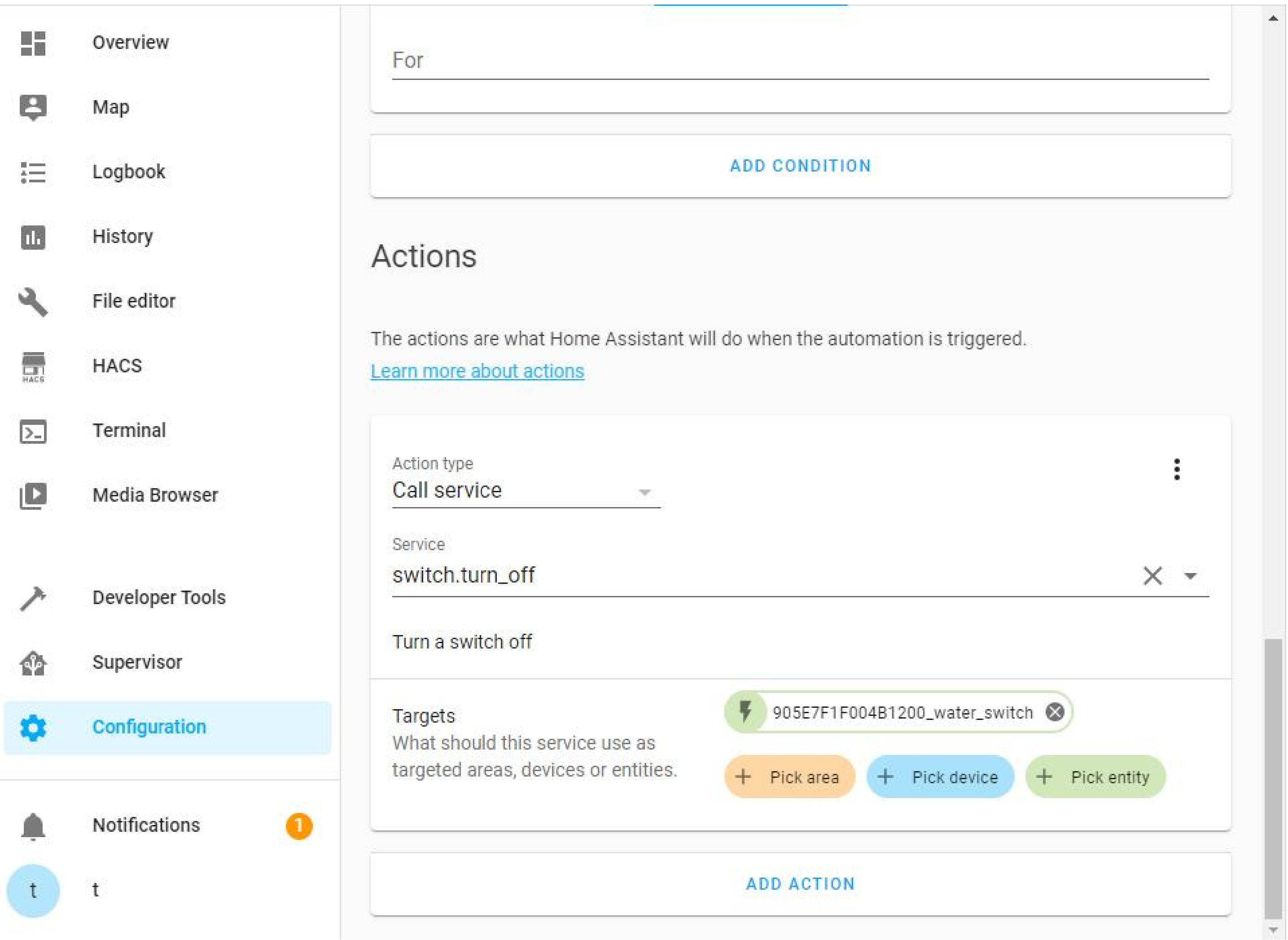
Entity
binary_sensor.905e7f1f004b1200_is_watering

Attribute (Optional)

State
on

For

Actions



For

Actions

The actions are what Home Assistant will do when the automation is triggered.
Learn more about actions

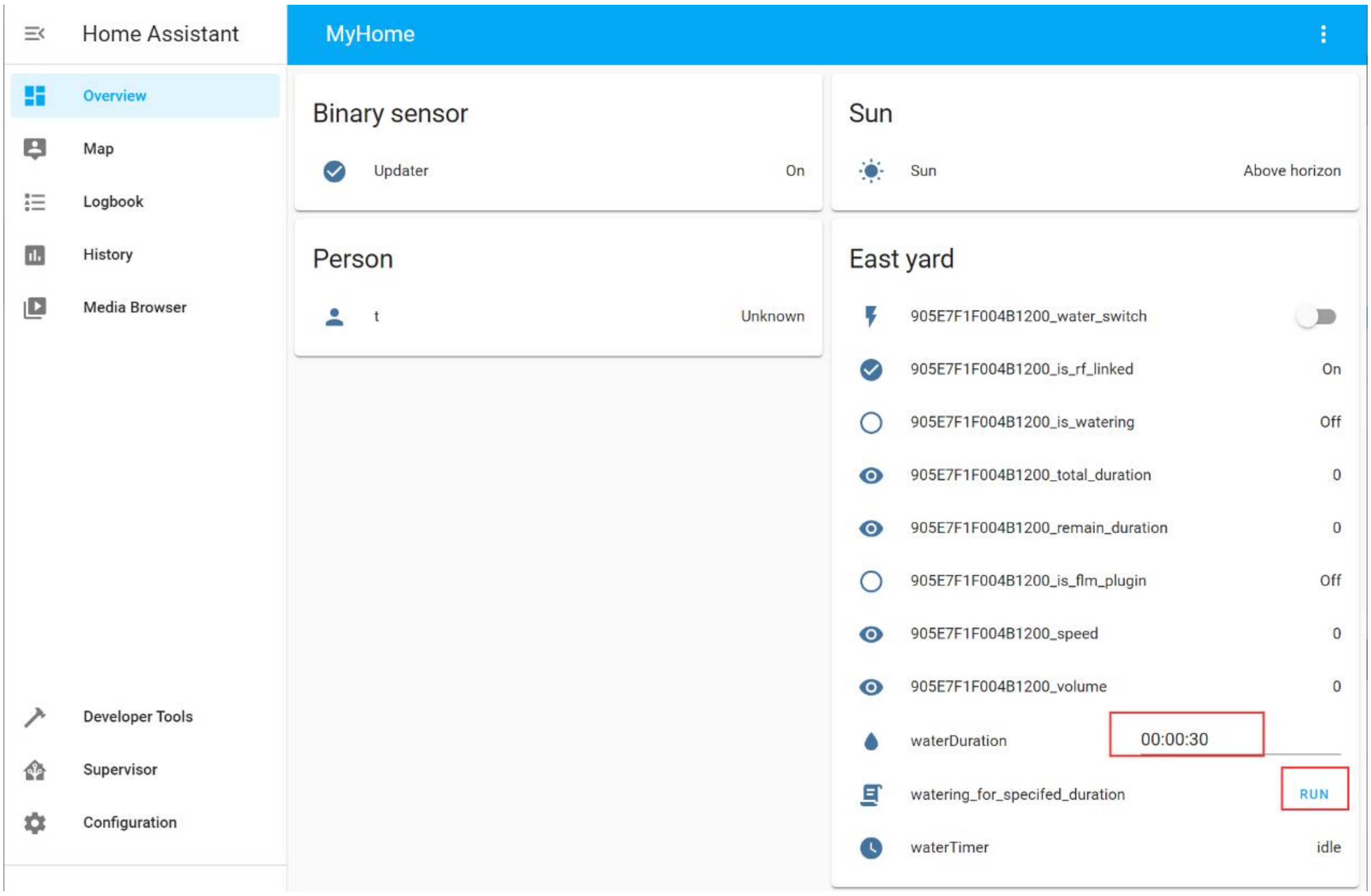
Action type
Call service

Service
switch.turn_off

Turn a switch off

Targets
What should this service use as targeted areas, devices or entities.
905E7F1F004B1200_water_switch

5) Now you can enter "waterDuration", then click "RUN" to make the TapLinker water for the specified duration.



Binary sensor

Updater	On

Sun

Sun	Above horizon

Person

t	Unknown

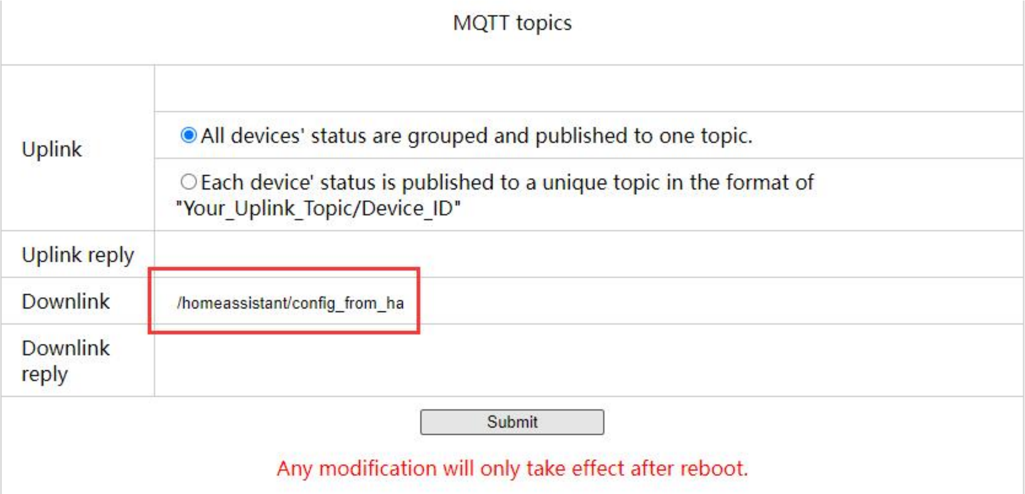
East yard

905E7F1F004B1200_water_switch	
905E7F1F004B1200_is_rf_linked	On
905E7F1F004B1200_is_watering	Off
905E7F1F004B1200_total_duration	0
905E7F1F004B1200_remain_duration	0
905E7F1F004B1200_is_flm_plugin	Off
905E7F1F004B1200_speed	0
905E7F1F004B1200_volume	0
waterDuration	00:00:30
watering_for_specifed_duration	RUN
waterTimer	idle

2.5. Set watering duration (value of sensor.X_total_duration entity)

By default, when a "water_switch (ON)" command is sent to the gateway from the HA (or from an automation script) to start watering, the default "total_duration" used by the gateway is 86340 seconds. After Taplinker receives the command to start watering for "total_duration" seconds from the HA, it will turn on the valve, then start counting down from "total_duration", and then periodically report the countdown value to the HA as "remain_duration". When "remain_duration" turns to 0, the Taplinker will automatically turn off the valve to stop watering. If needed, you can also send the "water_switch (OFF)" command from the HA at any time to stop watering. On the other hand, you can set a watering duration through “total_duration” entity before sending the "water_switch (ON)" command. The watering device will then stop watering when the “total_duration” runs up.

To set the " total _duration", you'll need to configure the "Downlink" section of an MQTT topic as shown below.



MQTT topics

Uplink	● All devices' status are grouped and published to one topic. ○ Each device' status is published to a unique topic in the format of "Your_Uplink_Topic/Device_ID"
Uplink reply	
Downlink	/homeassistant/config_from_ha
Downlink reply	

Any modification will only take effect after reboot.

Note: To avoid interaction issues between the gateway and HA, the prefix of your HA needs to be used when filling the "Downlink" section of the MQTT topic. As shown below, the "Home Assistant prefix" in our system is "homeassistant", therefore the "Downlink" section starts with "homeassistant/".

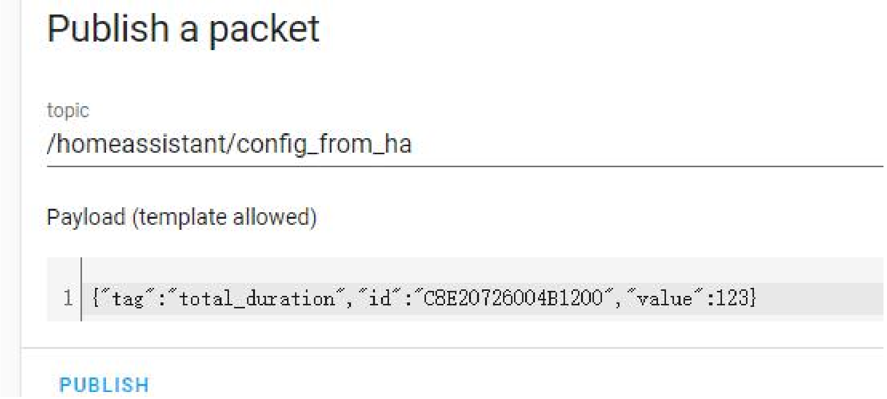
After the MQTT topic is set up, reboot the gateway for the settings to take effect.

The format of the MQTT payload that sets "total_duration" is

{"tag":"total_duration", "id":"YOUR_TAPLINKER_ID", "value": DURATION_IN_SECOND},

where “tag” is the name of the configuration, “id” is the ID of the Taplinker, “value” is the value to be set to the configuration, in seconds here, with a minimum of 3 and a maximum of 86340. For example:

{"tag":"total_duration", "id":"C8E20726004B1200", "value":123}



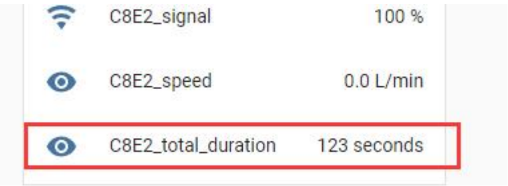
Publish a packet

topic
/homeassistant/config_from_ha

Payload (template allowed)

```
{"tag":"total_duration","id":"C8E20726004B1200","value":123}
```

Once the configuration message is published successfully, the value of "YOUR_TAPLINKER_ID_total_duration" will be updated accordingly on the Dashboard card.



C8E2_signal 100 %

C8E2_speed 0.0 L/min

C8E2_total_duration 123 seconds

Please note that "total_duration" is not retained across multiple watering cycles. For each "watering b y duration", the "total_duration" needs to be configured once. To simplify the operation, you can configure the "total_duration" and start watering together through a script. For example:

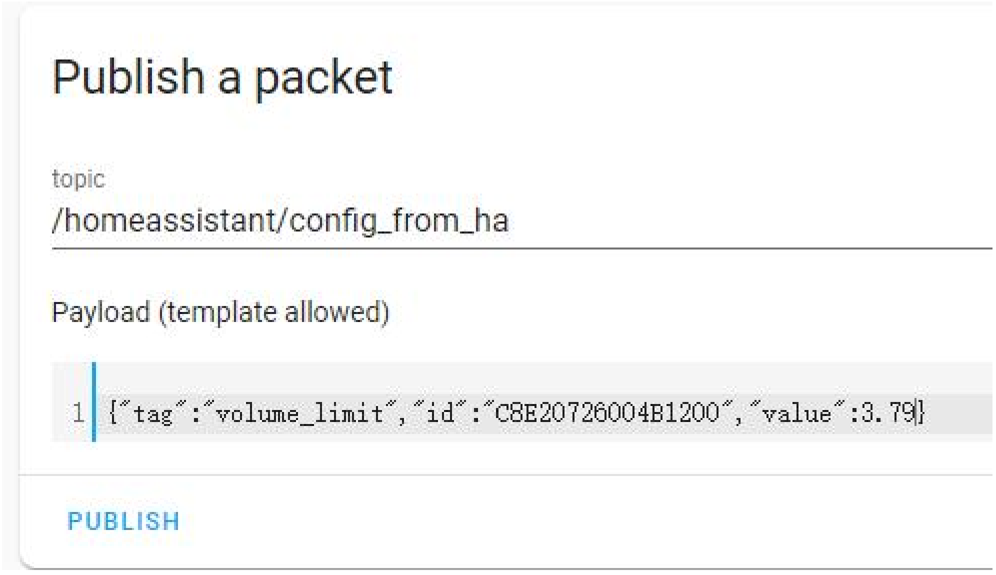
```
alias: set_total_duration_then_start_watering
sequence:
  - service: mqtt.publish
    data:
      topic: /homeassistant/config_from_ha
      payload: "{\"tag\":\"total_duration\",\"id\":\"C8E20726004B1200\",\"value\":123}"
  - service: switch.turn_on
    data: {}
    target:
      entity_id: switch. c8e20726004b1200_water_switch
mode: single
```

2.6. Set watering volume (value of sensor.X_volume_limit entity)

Apart from watering by duration, you can also water by volume. Before setting a volume limit for a watering session, as in the Chapter 2.5, in the “MQTT topics” -> “Downlink” field of the Gateway admin page, you’ll need to set a MQTT topic to receive the HA’s configuration message. If a MQTT topic has already been set when configuring the “total_duration”, then there is no need to set this topic again. The “watering volume limit”, “watering duration”, and possible future configuration will share the same MQTT topic.

The format of the MQTT payload that sets "volume_limit" is {"tag":"volume_limit", "id":"YOUR_TAPLINKER_ID", "value":VOLUME_VALUE_IN_LITRE}, where “tag” is the name of the configuration, “id” is the ID of the Taplinker, “value” is the value to be set to the configuration. For the “volume_limit”, the unit of this value is liters. For example:

```
{"tag":"volume_limit", "id":"C8E20726004B1200", "value":3.79}
```



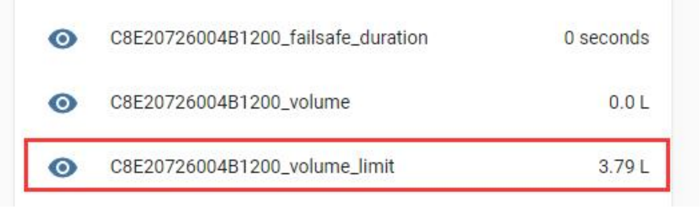
Publish a packet

topic
/homeassistant/config_from_ha

Payload (template allowed)

```
{"tag":"volume_limit","id":"C8E20726004B1200","value":3.79}
```

Once the configuration message is published successfully, the value of "YOUR_TAPLINKER_ID_volume_limit" will be updated accordingly on the Dashboard card.



C8E20726004B1200_failsafe_duration	0 seconds
C8E20726004B1200_volume	0.0 L
C8E20726004B1200_volume_limit	3.79 L

Please note that "volume_limit" is not retained across multiple watering cycles. For each "watering by volume", the "volume_limit" needs to be configured once. To simplify the operation, you can configure the "volume_limit" and start watering together through a script. For example:

```
alias: set_volume_limit_then_start_watering
sequence:
  - service: mqtt.publish
    data:
      topic: /homeassistant/config_from_ha
      payload: "{\"tag\":\"volume_limit\",\"id\":\"C8E20726004B1200\",\"value\":3.79}"
  - service: switch.turn_on
    data: {}
    target:
      entity_id: switch. c8e20726004b1200_water_switch
mode: single
```

Note: If the flowmeter of your G2S/Valvelinker/D1 is not connected, or the "volume_limit" is configured for the G1/G2/G1S that does not support flowmeter, the device will water for the duration regardless of the volume.

2.7. Set failsafe watering duration

*/******************************

The “failsafe_duration” was implemented before “total_duration”. If you are a new user, the “total_duration” is recommended to set a watering duration. If both the "total_duration" and "failsafe_duration" are set (that is, neither value is 0), the gateway will use the minimum of the two as the duration of the current watering cycle.*

The "failsafe_duration" and "total_duration" serve the similar purposes, that is, they both affect the duration of a watering cycle, but there are some differences between them:

	total_duration	*failsafe_duration*
Lifespan	*The total_duration needs to be set every time before a watering cycle is initiated from HA. If the total_duration is not set before a new watering cycle starts, the default value (86340) will be used by the gateway.*	*After set, the failsafe_duration will be permanently stored in the gateway until the gateway is restored to factory settings.*
Coverage	*One message serves one Taplinker only.*	*One message can serve either one Taplinker or all Taplinkers under a gateway.*
When to use	*1) When the watering duration needs to be changed frequently.* *2) When you want to set a one-shot watering duration that will not affect any other settings.*	*1) When you want to set a maximum watering duration for all Taplinkers under a gateway.* *2) The watering duration is relatively fixed and will not change frequently. Also you want it to be always effective after being set.*

******************************/

If the TapLinker is controlled via the Home Assistant (HA), the watering duration of the TapLinker is determined by the time interval between the HA sending the "water_switch (ON)" and “water_switch (OFF)" commands. By default, when a "water_switch (ON)" command is sent to the gateway from the HA (or from an automation script) to start watering, the default "total_duration" used by the gateway is 86340 seconds. The watering will not stop until the "remain_duration" counting down from "total_duration" becomes zero or the gateway receives a "water_switch (OFF)" command from the HA. If for some reason the connection between the Gateway and HA is lost while the TapLinker is watering, the "water_switch (OFF)" command will not be able to reach the TapLinker, resulting in overwatering.

To overcome the above issue, "failsafe_duration" is introduced. When the gateway receives the "water_switch (ON)" command, it will use the "failsaft_duration" as the "total_duration". The TapLinker will stop watering once the "failsaft_duration" is up.

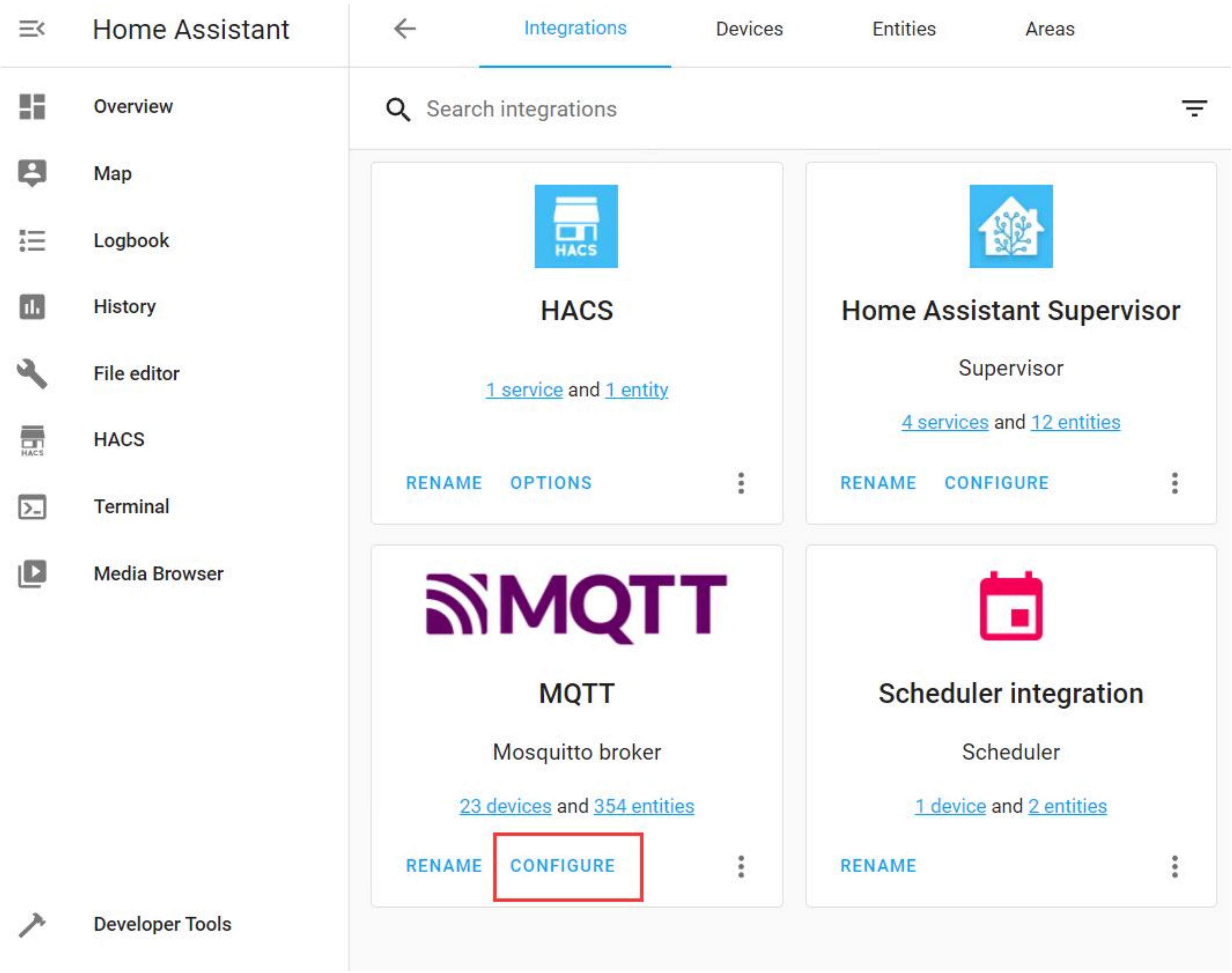
You can set the "failsafe_duration" by publishing content to the topic from the HA (or other MQTT Clients subscribed to the topic). The content format for this topic is:

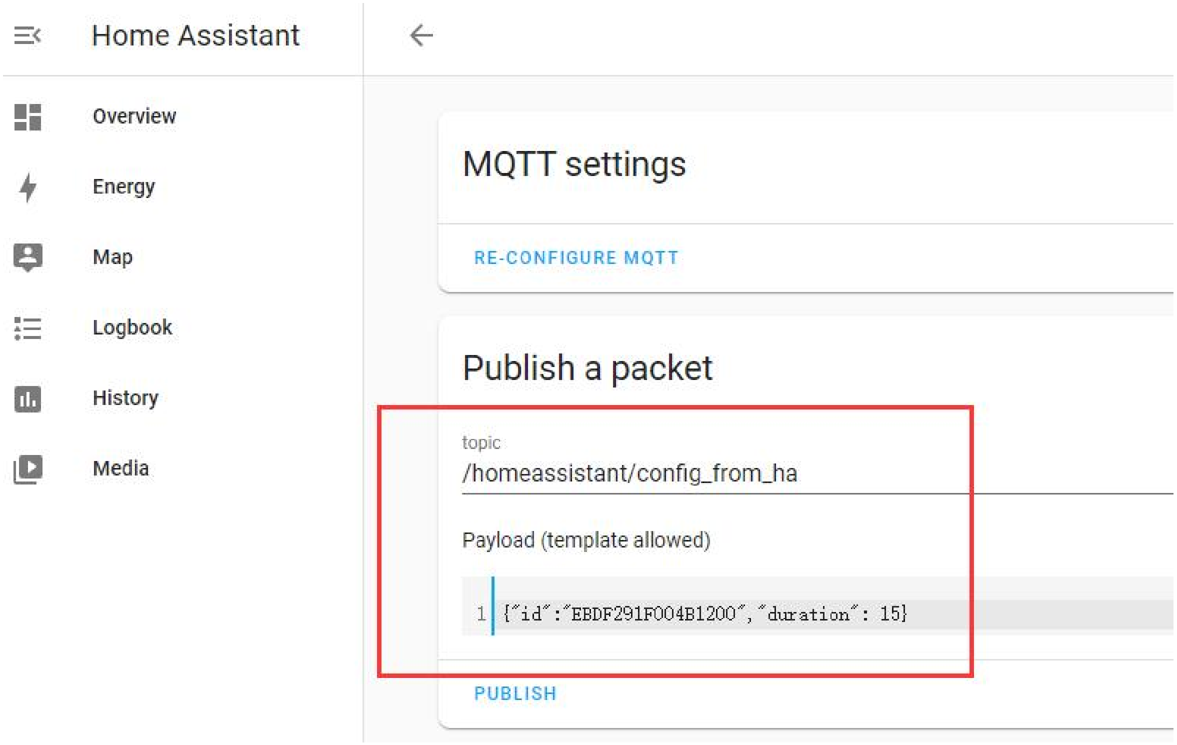
{"id":"Your_TapLinker_ID","duration": Integer_Value}, or {"duration": Integer_Value}.

where "id" is the ID of your TapLinker, and "duration" is the value of "failsafe_duration" in seconds. If "id" is omitted, then the "duration" will be applied for all TapLinkers, and all TapLinkers' "failsafe_duration" previously set will be overridden.

When the gateway receives the content published by the topic, it will save the content. If there is no change in "failsafe_duration" in the future, it will be used all the time, so users do not need to set it before starting watering each time.

Below is an example that the TapLinker (ID: EBDF291F004B1200) is configured via the Mosquitto broker from the HA. Enter the topic and payload, then click "PUBLISH". Once the gateway successfully receives the message, it will publish a new value (e.g., 15) to the entity named "EBDF291F004B1200_failsafe_duration" on the HA. Now if watering is started through "EBDF291F004B1200_water_switch", the TapLinker will water for 15 seconds.





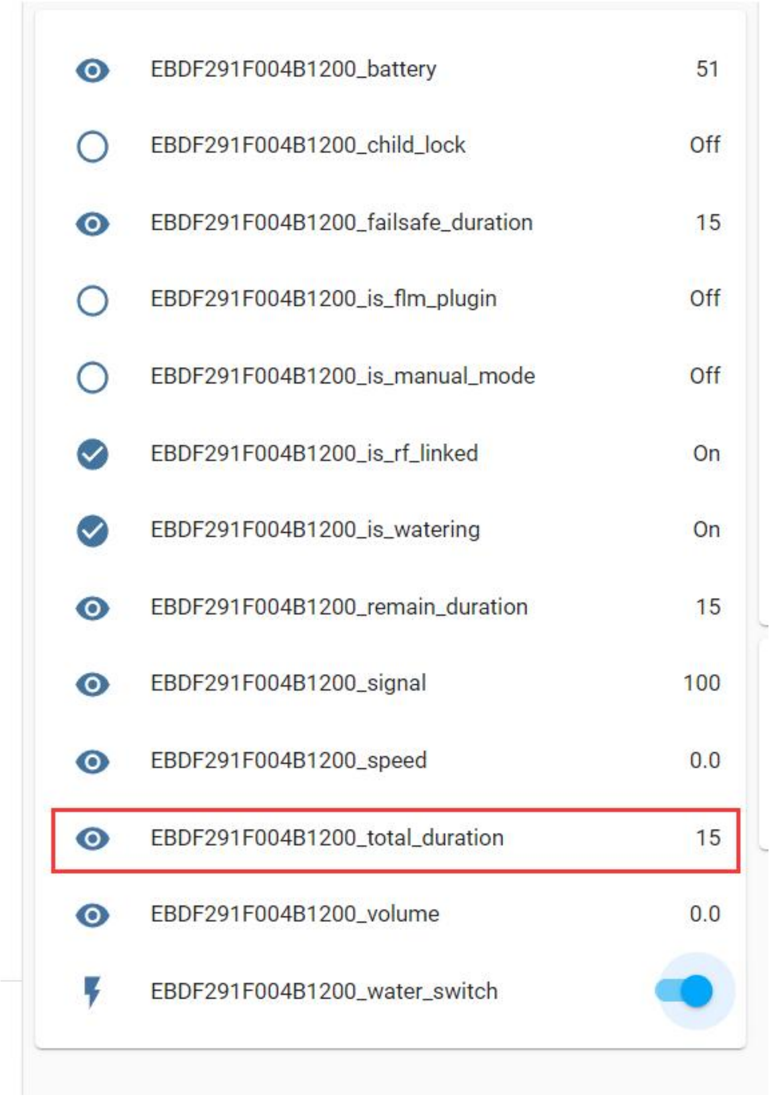
MQTT settings

Publish a packet

topic
/homeassistant/config_from_ha

Payload (template allowed)

```
{"id":"EBDF291F004B1200","duration": 15}
```



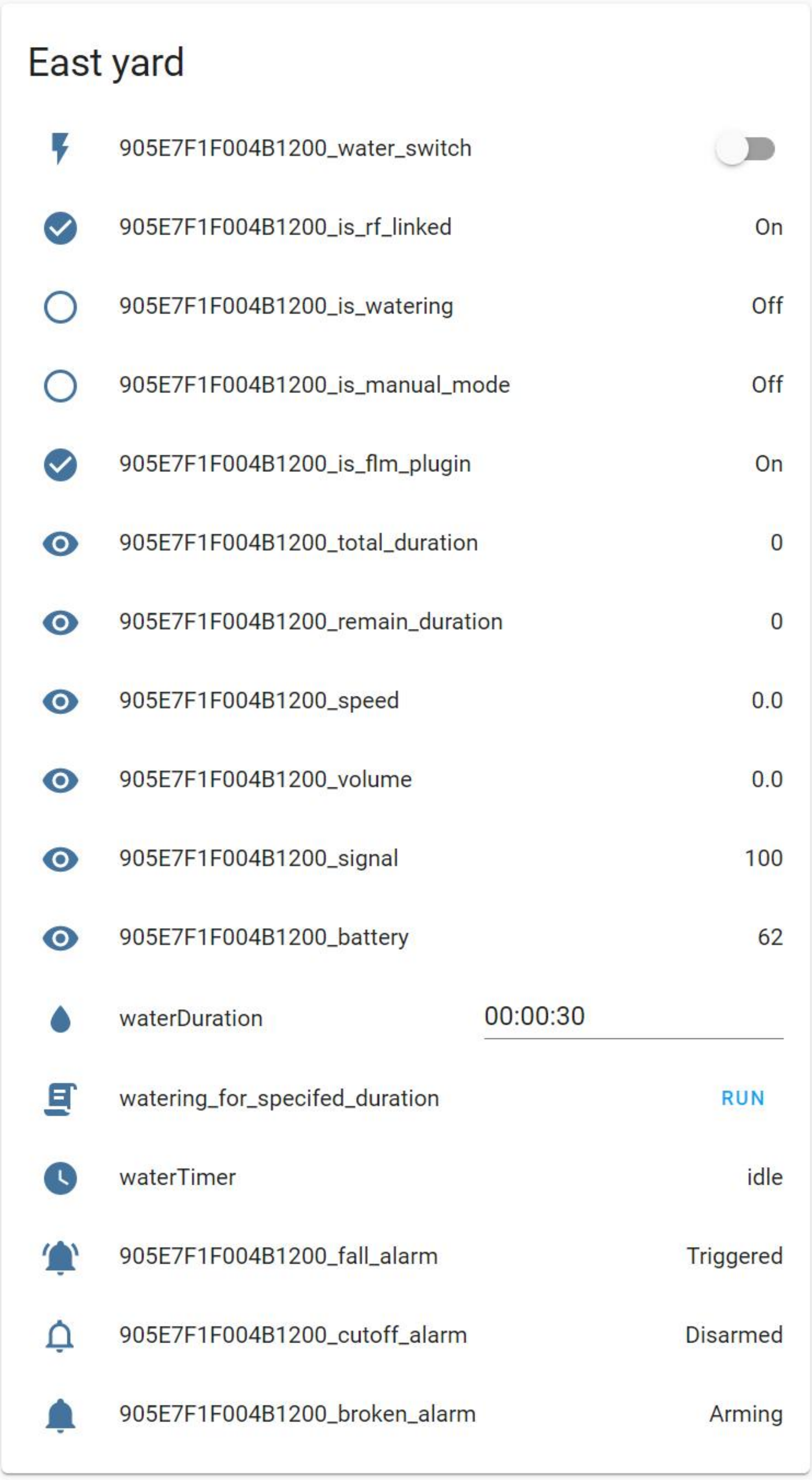
EBDF291F004B1200_battery	51
EBDF291F004B1200_child_lock	Off
EBDF291F004B1200_failsafe_duration	15
EBDF291F004B1200_is_flm_plugin	Off
EBDF291F004B1200_is_manual_mode	Off
EBDF291F004B1200_is_rf_linked	On
EBDF291F004B1200_is_watering	On
EBDF291F004B1200_remain_duration	15
EBDF291F004B1200_signal	100
EBDF291F004B1200_speed	0.0
EBDF291F004B1200_total_duration	15
EBDF291F004B1200_volume	0.0
EBDF291F004B1200_water_switch	

2.8. Taplinker’s Fault Alerts on Home Assistant

The LinkTap water timer's fault alarm function on Home Assistant is implemented through the alarm_control_panel platform. TapLinker fall alarm, water shut-off failure alarm, and water cut-off alarm can be received on Home Assistant now. Those alarms can also be dismissed from Home Assistant. Please note, enabling or disabling the alarms can only be done through the LinkTap mobile app or web app at present.

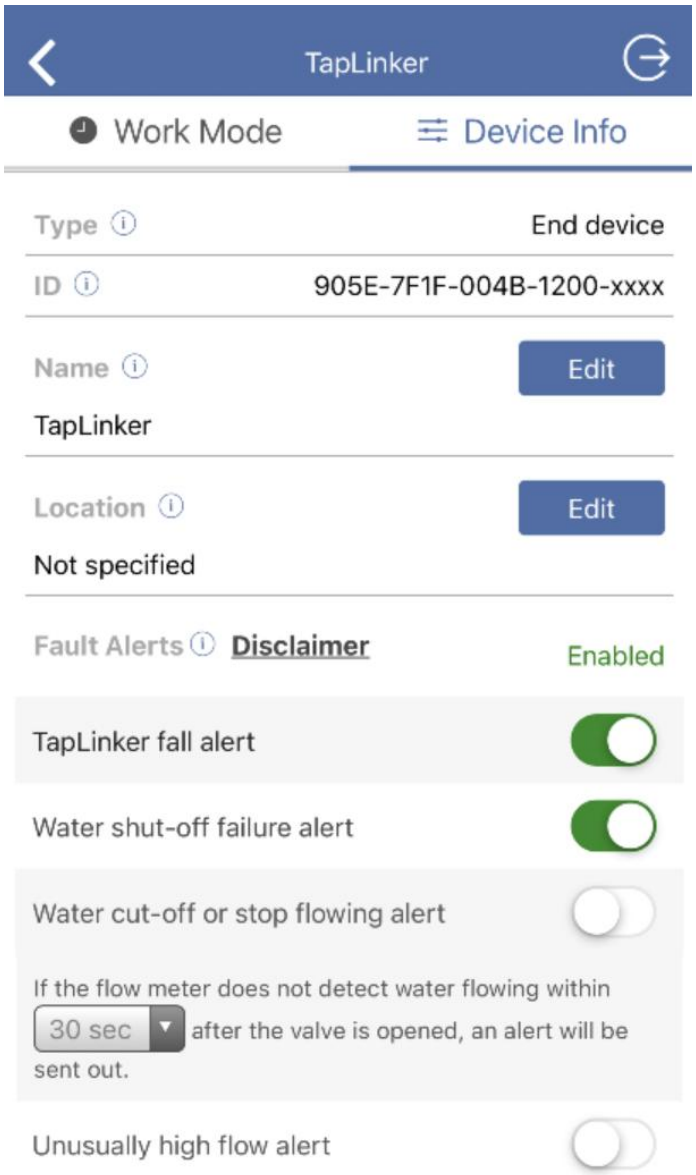
Please note that when the LinkTap Gateway is rebooted, the state switching on alarm_control_panel may be delayed due to the message interaction between TapLinker and the Gateway, or between the Gateway and the LinkTap server.

Below shows an example. The "TapLinker fall alert" and "Water shut-off failure alert" are enabled through the LinkTap app, and the "Water cut-off or stop flowing alert" is disabled. A fall alarm is generated after the TapLinker falls.



East yard

905E7F1F004B1200_water_switch	
905E7F1F004B1200_is_rf_linked	On
905E7F1F004B1200_is_watering	Off
905E7F1F004B1200_is_manual_mode	Off
905E7F1F004B1200_is_flm_plugin	On
905E7F1F004B1200_total_duration	0
905E7F1F004B1200_remain_duration	0
905E7F1F004B1200_speed	0.0
905E7F1F004B1200_volume	0.0
905E7F1F004B1200_signal	100
905E7F1F004B1200_battery	62
waterDuration	00:00:30
watering_for_specifed_duration	RUN
waterTimer	idle
905E7F1F004B1200_fall_alarm	Triggered
905E7F1F004B1200_cutoff_alarm	Disarmed
905E7F1F004B1200_broken_alarm	Arming



TapLinker

Work Mode | Device Info

Type ⓘ End device

ID ⓘ 905E-7F1F-004B-1200-xxxx

Name ⓘ Edit

TapLinker

Location ⓘ Edit

Not specified

Fault Alerts ⓘ Disclaimer Enabled

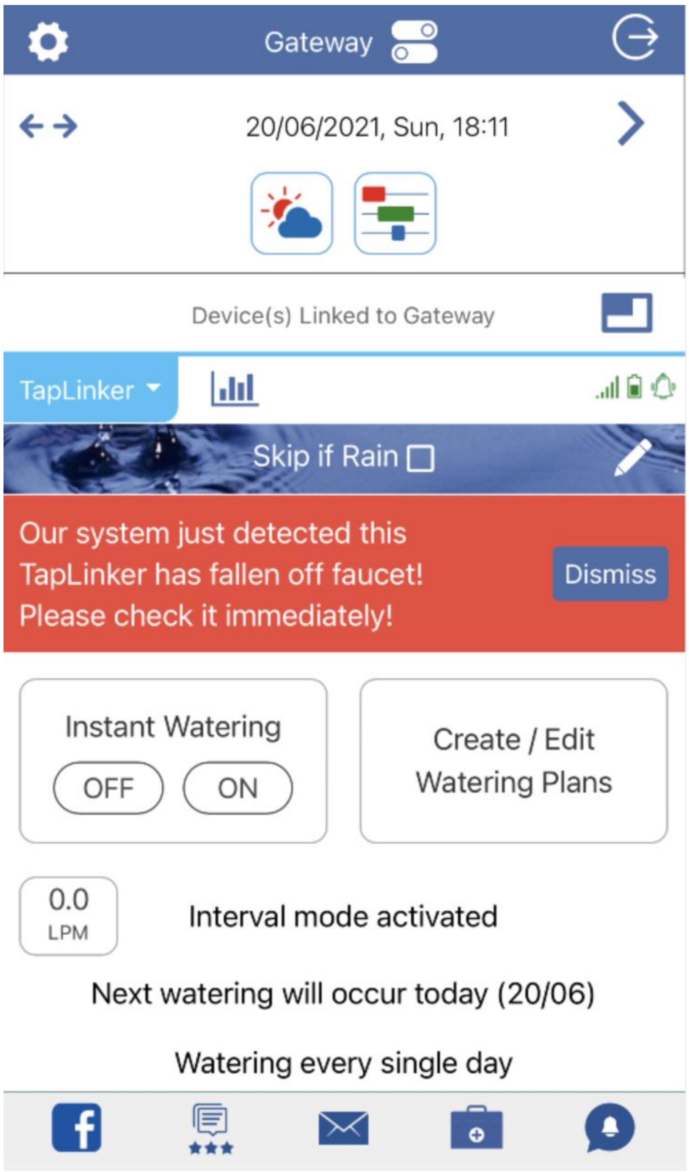
TapLinker fall alert ●

Water shut-off failure alert ●

Water cut-off or stop flowing alert ○

If the flow meter does not detect water flowing within 30 sec after the valve is opened, an alert will be sent out.

Unusually high flow alert ○



Gateway

20/06/2021, Sun, 18:11

Device(s) Linked to Gateway

TapLinker

Skip if Rain □

Our system just detected this TapLinker has fallen off faucet! Please check it immediately!

Dismiss

Instant Watering

OFF ON

Create / Edit Watering Plans

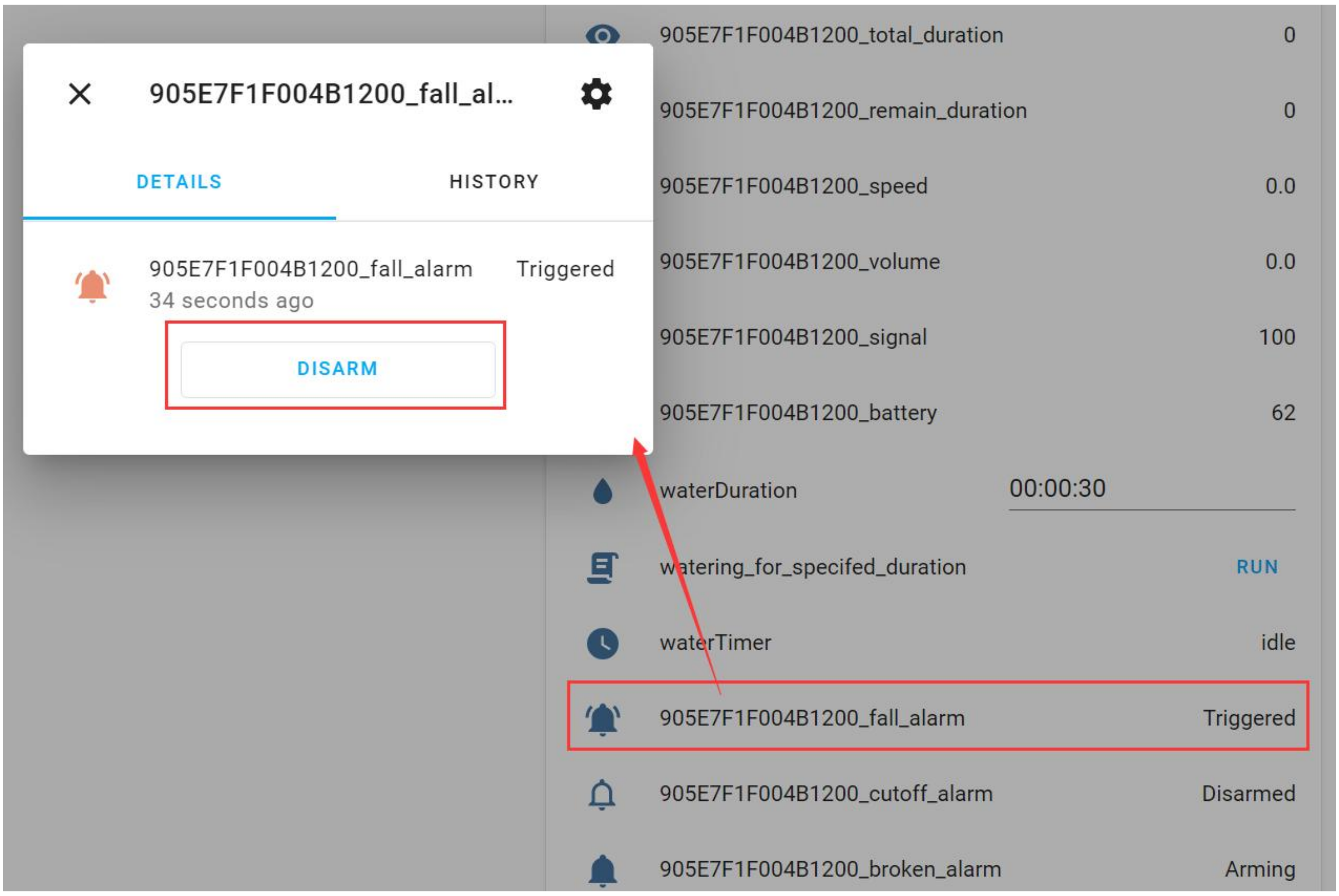
0.0 LPM

Interval mode activated

Next watering will occur today (20/06)

Watering every single day

After checking the fault, you can click "DISARM" to clear the alarm, and then the alarm_control_panel status will automatically switch to "Arming".



905E7F1F004B1200_fall_al...

DETAILS HISTORY

905E7F1F004B1200_fall_alarm Triggered
34 seconds ago

DISARM

905E7F1F004B1200_total_duration 0

905E7F1F004B1200_remain_duration 0

905E7F1F004B1200_speed 0.0

905E7F1F004B1200_volume 0.0

905E7F1F004B1200_signal 100

905E7F1F004B1200_battery 62

waterDuration 00:00:30

watering_for_specifed_duration RUN

waterTimer idle

905E7F1F004B1200_fall_alarm Triggered

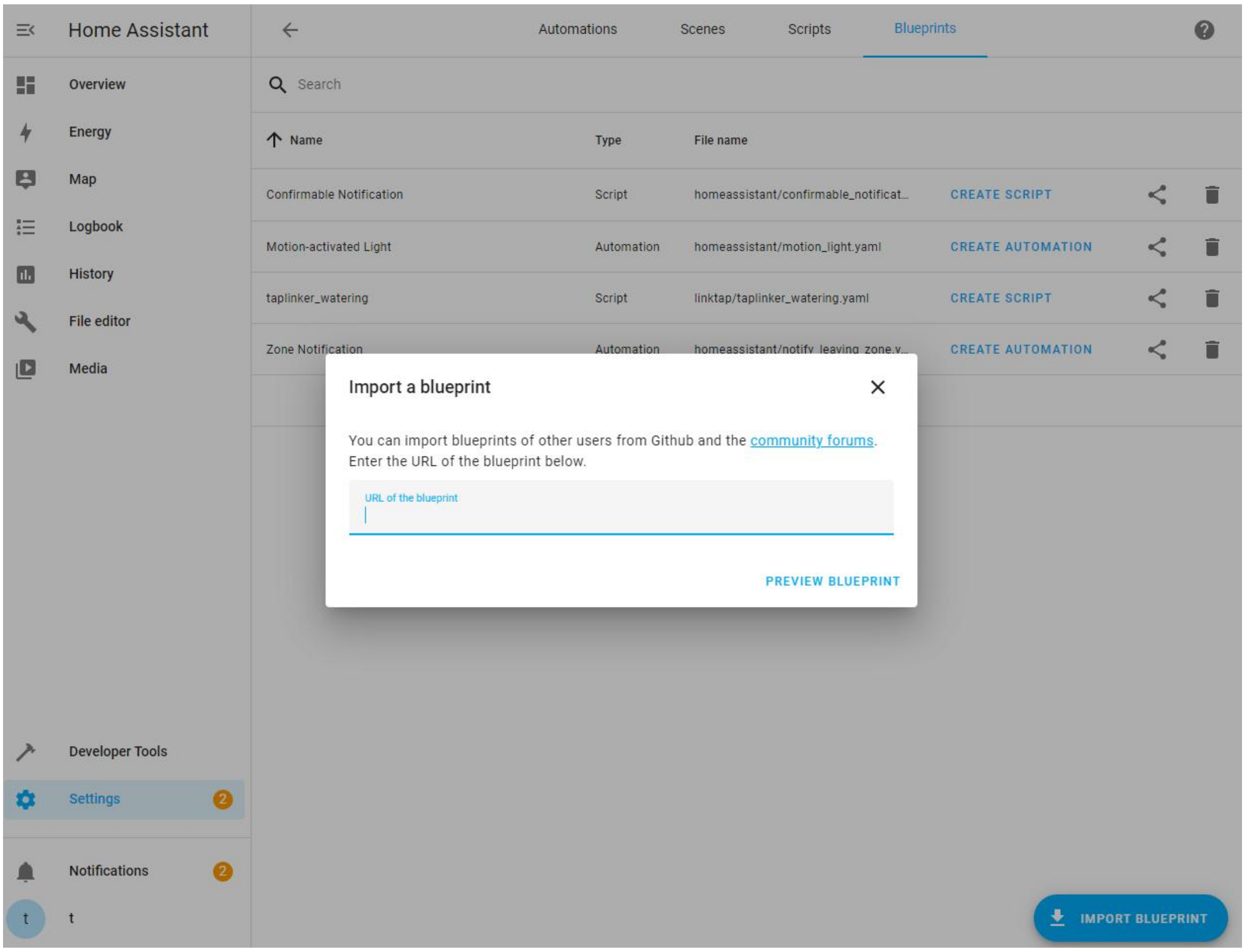
905E7F1F004B1200_cutoff_alarm Disarmed

905E7F1F004B1200_broken_alarm Arming

2.9. Create a script via Blueprint to control Taplinker’s watering

As shown below, on the "Settings > Automations & Scenes > Blueprints" page, click the "IMPORT BLUEPRINT" button, then enter the path of the blueprint named "taplinker_watering" on the pop-up window:

https://github.com/jimlktwork/ha/blob/main/taplinker_watering.yaml



Name	Type	File name			
Confirmable Notification	Script	homeassistant/confirmable_notificat...	CREATE SCRIPT		
Motion-activated Light	Automation	homeassistant/motion_light.yaml	CREATE AUTOMATION		
taplinker_watering	Script	linktap/taplinker_watering.yaml	CREATE SCRIPT		
Zone Notification	Automation	homeassistant/notify_leaving_zone.y...	CREATE AUTOMATION		

Import a blueprint

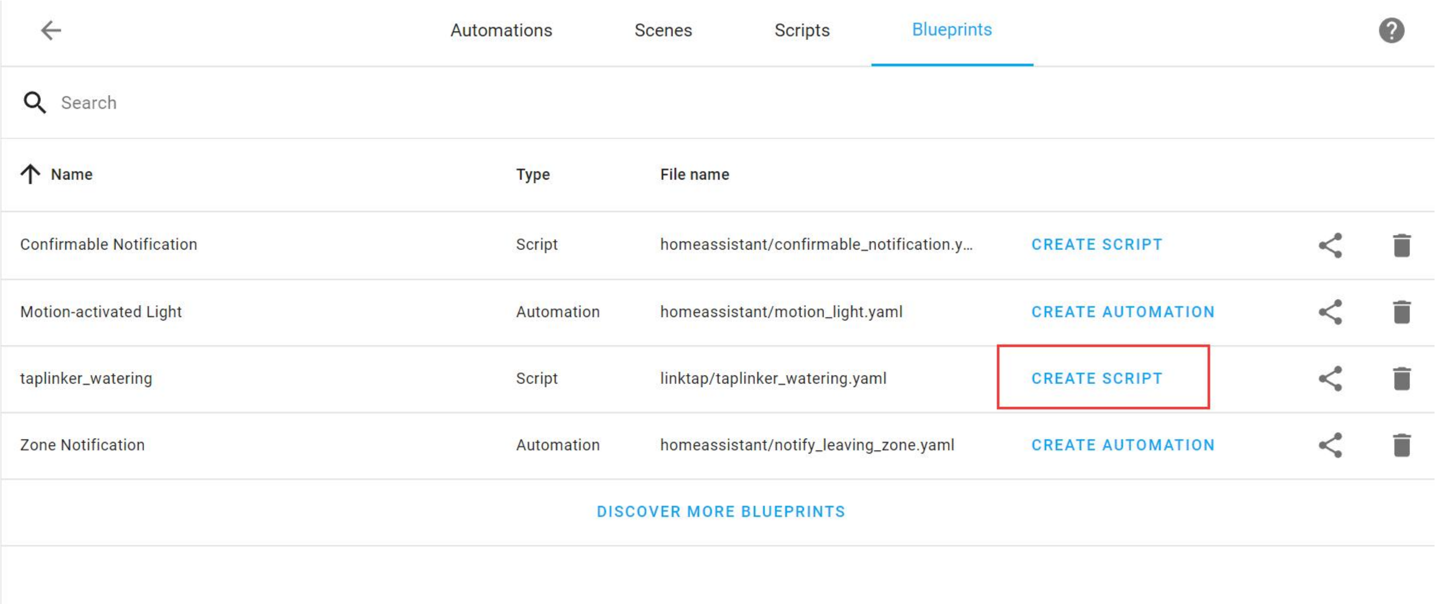
You can import blueprints of other users from Github and the community forums. Enter the URL of the blueprint below.

URL of the blueprint

PREVIEW BLUEPRINT

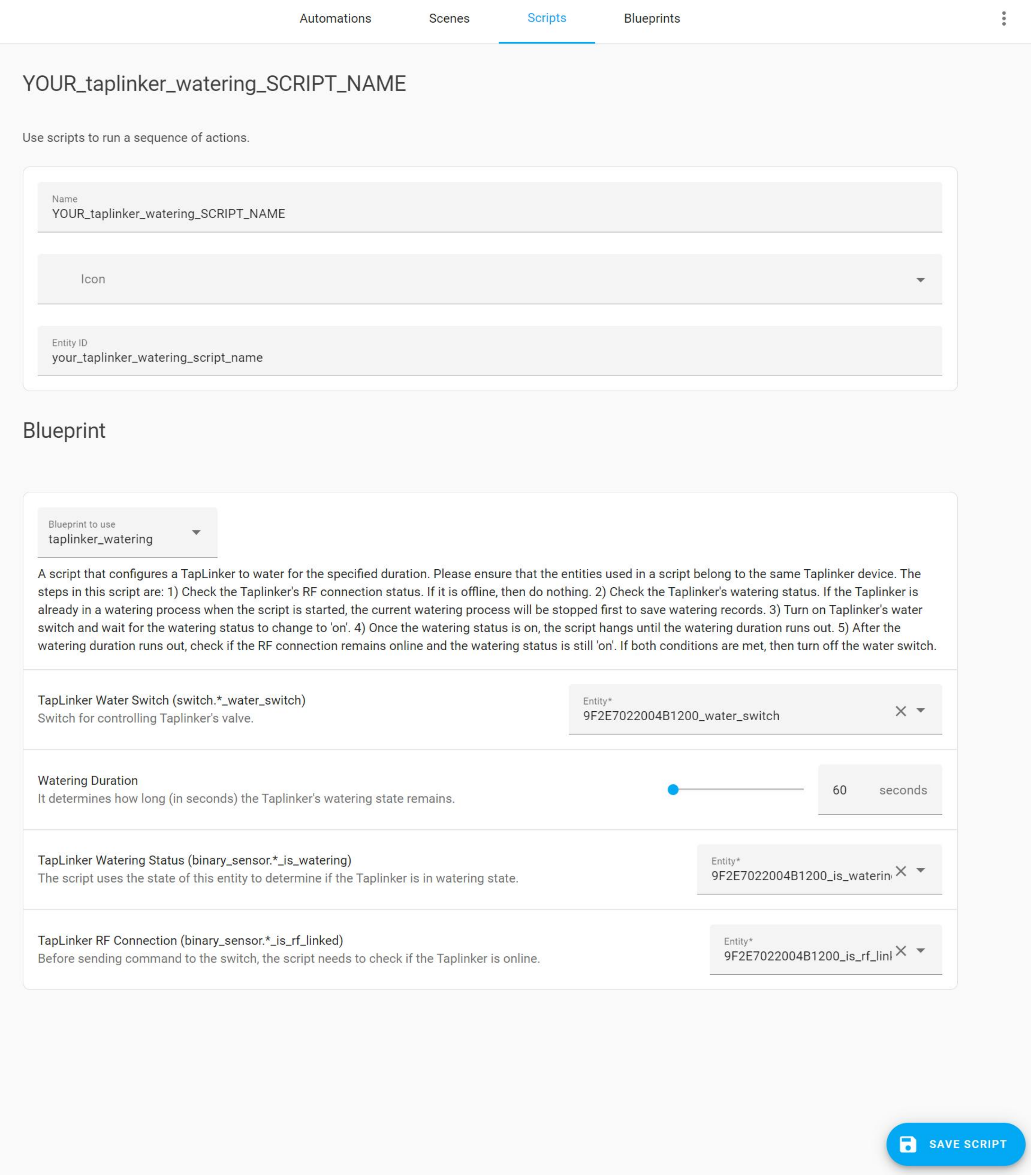
Click "PREVIEW BLUEPRINT" to view the description of the blueprint, then click "Import blueprint" to import the blueprint.

Now you'll see the "taplinker_watering" in the Blueprints list. Next, as shown below, click "CREATE SCRIPT" to start creating a script file through blueprint to control a Taplinker.



Name	Type	File name			
Confirmable Notification	Script	homeassistant/confirmable_notification.y...	CREATE SCRIPT		
Motion-activated Light	Automation	homeassistant/motion_light.yaml	CREATE AUTOMATION		
taplinker_watering	Script	linktap/taplinker_watering.yaml	CREATE SCRIPT		
Zone Notification	Automation	homeassistant/notify_leaving_zone.yaml	CREATE AUTOMATION		

On the following script build page, you just need to follow the prompts and select the three entities "switch.*_water_switch" (* indicates your LinkTap device ID), "binary_sensor.*_is_watering" and "binary_sensor.*_is_rf_linked", and set the taplinker's watering duration, then click "SAVE SCRIPT" to complete the script creation process.



YOUR_taplinker_watering_SCRIPT_NAME

Use scripts to run a sequence of actions.

Name: YOUR_taplinker_watering_SCRIPT_NAME

Icon

Entity ID: your_taplinker_watering_script_name

Blueprint

Blueprint to use: taplinker_watering

A script that configures a TapLinker to water for the specified duration. Please ensure that the entities used in a script belong to the same Taplinker device. The steps in this script are: 1) Check the Taplinker's RF connection status. If it is offline, then do nothing. 2) Check the Taplinker's watering status. If the Taplinker is already in a watering process when the script is started, the current watering process will be stopped first to save watering records. 3) Turn on Taplinker's water switch and wait for the watering status to change to 'on'. 4) Once the watering status is on, the script hangs until the watering duration runs out. 5) After the watering duration runs out, check if the RF connection remains online and the watering status is still 'on'. If both conditions are met, then turn off the water switch.

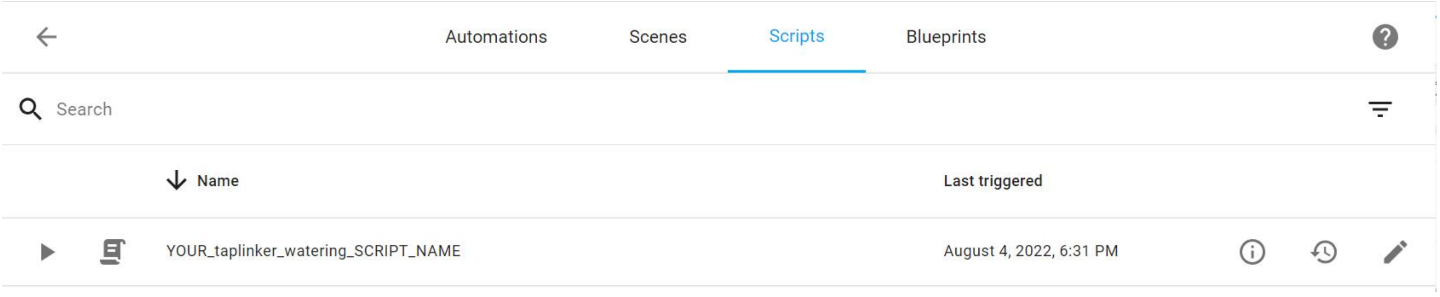
TapLinker Water Switch (switch.*_water_switch)
Switch for controlling Taplinker's valve.
Entity*: 9F2E7022004B1200_water_switch

Watering Duration
It determines how long (in seconds) the Taplinker's watering state remains.
60 seconds

TapLinker Watering Status (binary_sensor.*_is_watering)
The script uses the state of this entity to determine if the Taplinker is in watering state.
Entity*: 9F2E7022004B1200_is_waterin

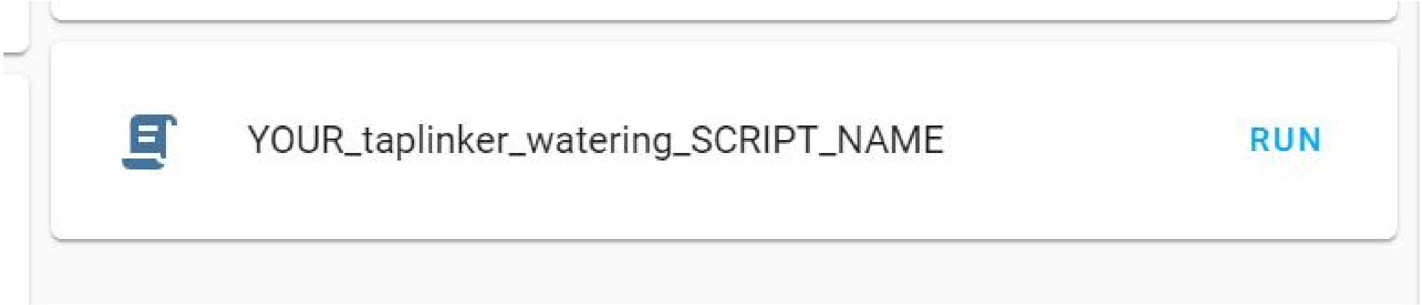
TapLinker RF Connection (binary_sensor.*_is_rf_linked)
Before sending command to the switch, the script needs to check if the Taplinker is online.
Entity*: 9F2E7022004B1200_is_rf_linl

As shown below, the newly created script will be listed in the "Scripts" page. You can then click the triangle icon (on the left of the list) to execute the script and check if it behaves as expected.



Name	Last triggered
YOUR_taplinker_watering_SCRIPT_NAME	August 4, 2022, 6:31 PM

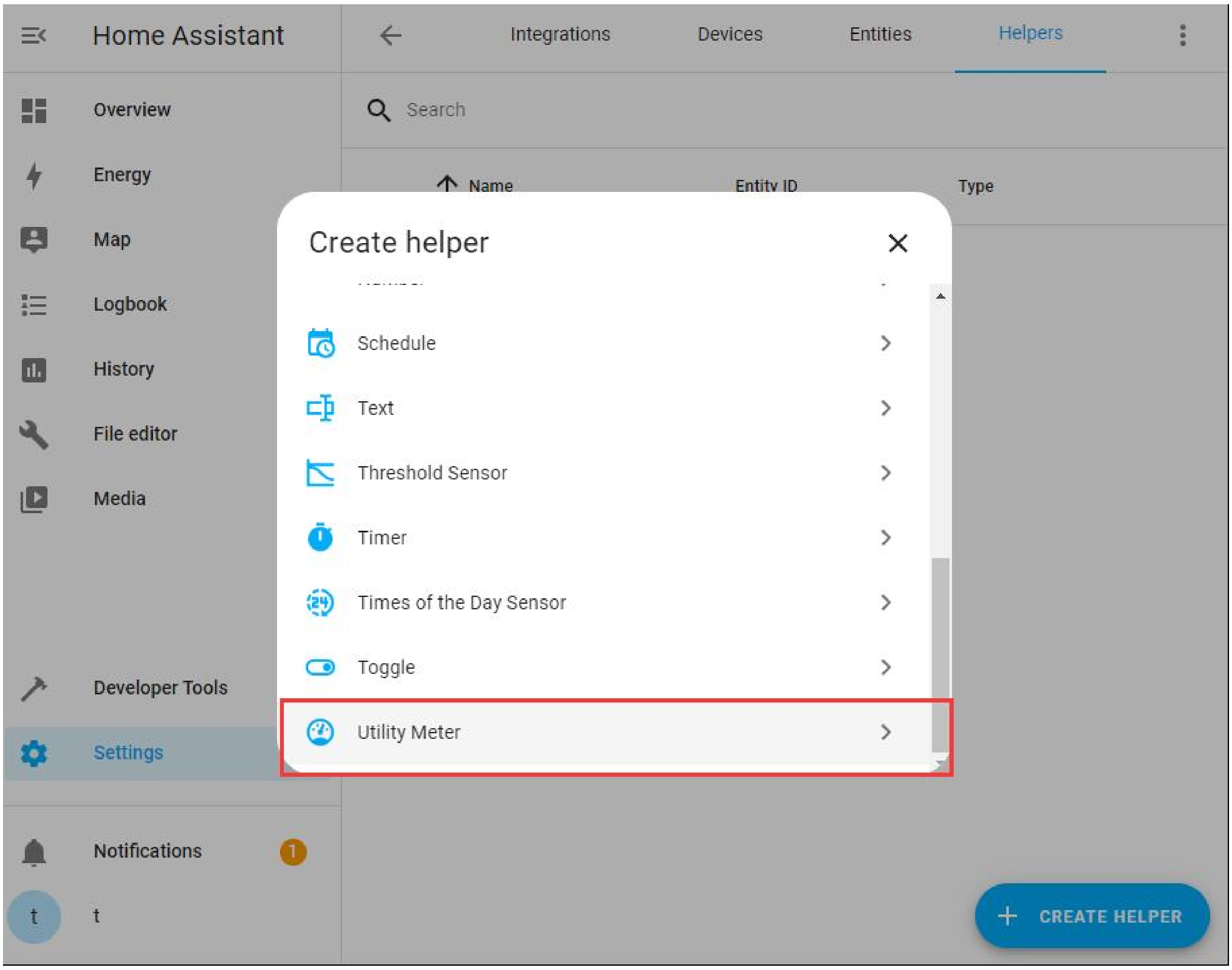
The script can also be put into a Dashboard card, or called by other scripts.



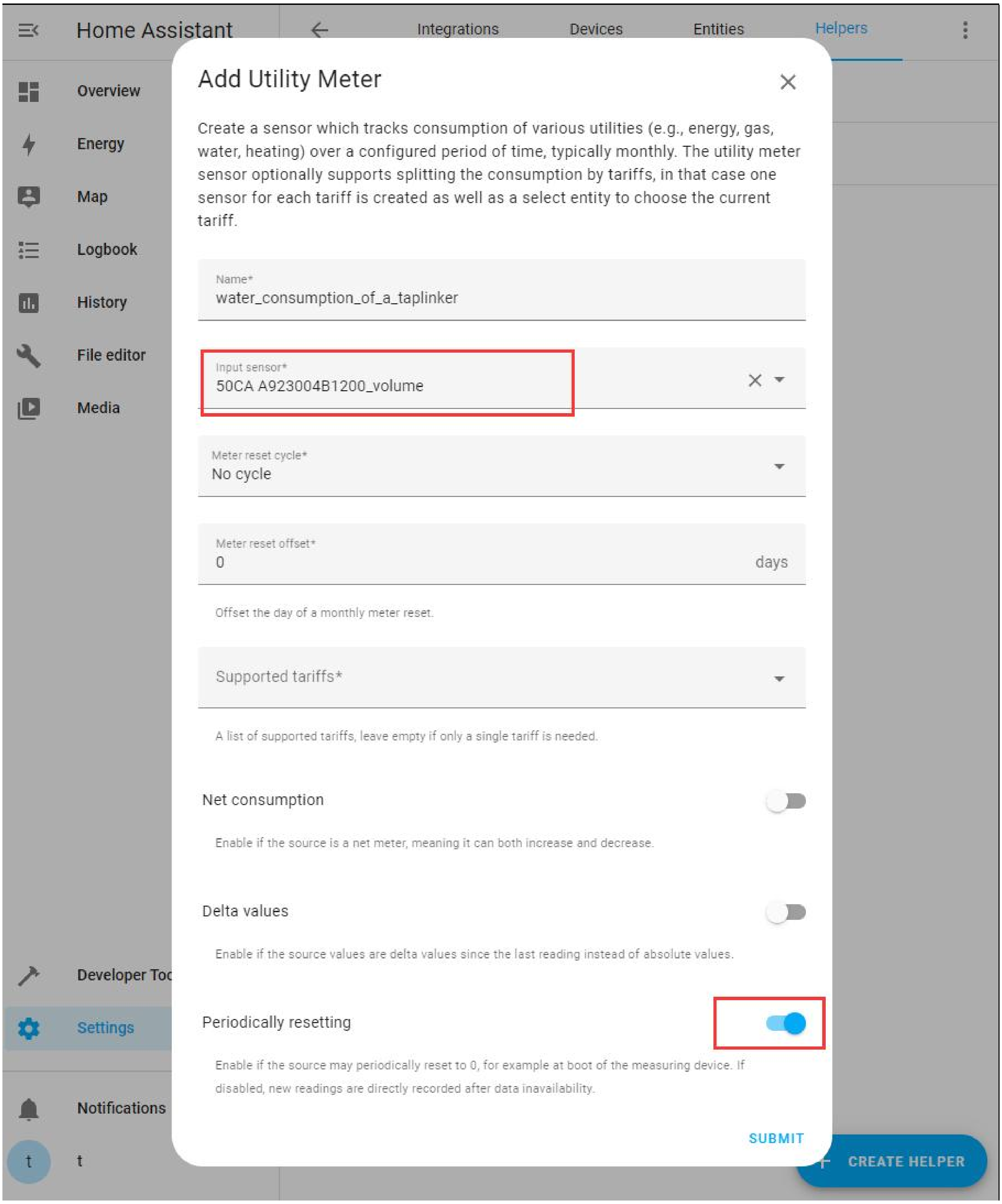
2.10. Track Taplinker water usage in the Energy dashboard

In the Taplinker entities provided by the LinkTap gateway, the 'X_volume' entity represents water consumption during an active watering process. When a new watering cycle begins, its value resets to 0, initiating a new water consumption count. The Home Assistant's Utility Meter can accumulate data from specific sources, allowing us to monitor Taplinker water consumption.

First, navigate to Settings > Helpers, then click on 'CREATE HELPER' in the bottom-right corner to create a Utility Meter helper entity.



During the 'Add Utility Meter' process, you can give the meter an appropriate name based on your needs. This name will be used as the entity name later on. Next, select the '_volume' entity under a specific Taplinker as your data source. For 'Meter reset cycle,' choose 'No cycle'. An important note: make sure to enable 'Periodically resetting'.



Add Utility Meter

Create a sensor which tracks consumption of various utilities (e.g., energy, gas, water, heating) over a configured period of time, typically monthly. The utility meter sensor optionally supports splitting the consumption by tariffs, in that case one sensor for each tariff is created as well as a select entity to choose the current tariff.

Name*: water_consumption_of_a_taplinker

Input sensor*: 50CA A923004B1200_volume

Meter reset cycle*: No cycle

Meter reset offset*: 0 days

Offset the day of a monthly meter reset.

Supported tariffs*

A list of supported tariffs, leave empty if only a single tariff is needed.

Net consumption

Enable if the source is a net meter, meaning it can both increase and decrease.

Delta values

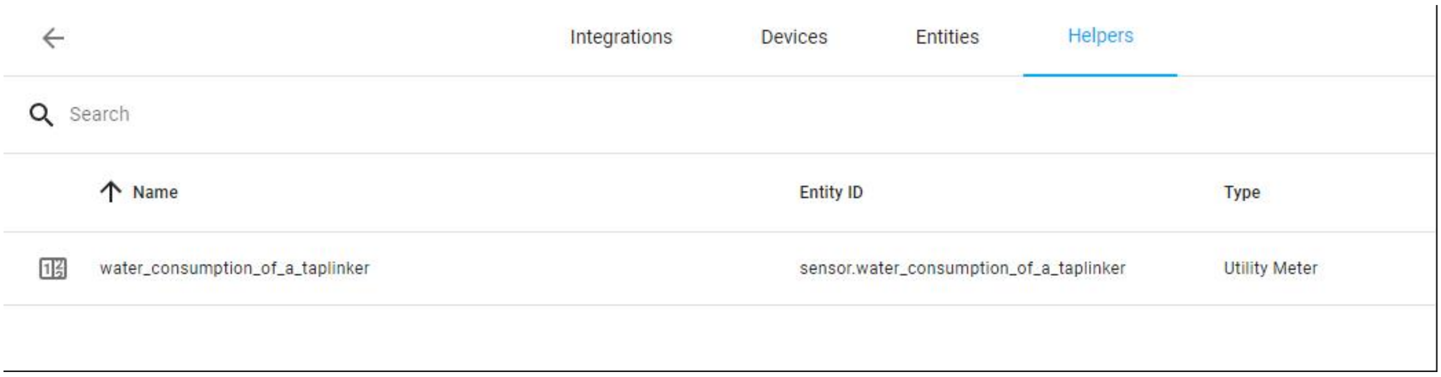
Enable if the source values are delta values since the last reading instead of absolute values.

Periodically resetting

Enable if the source may periodically reset to 0, for example at boot of the measuring device. If disabled, new readings are directly recorded after data inavailability.

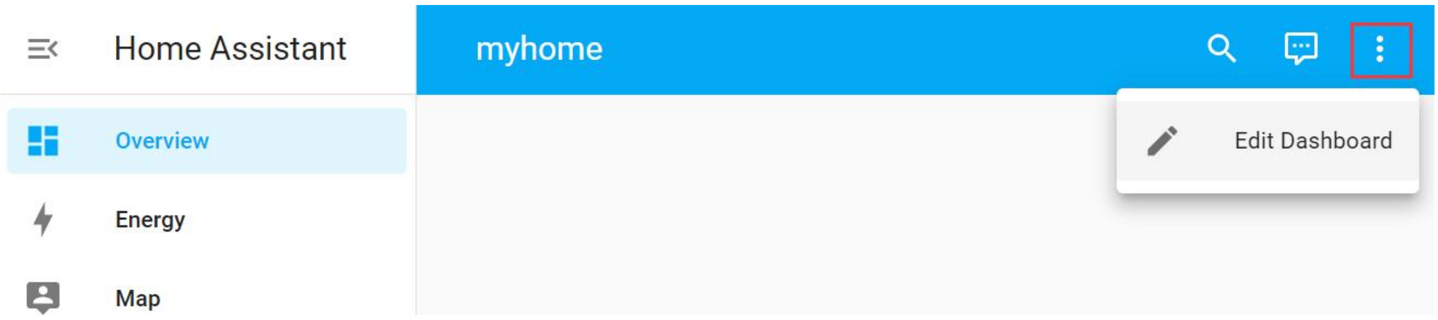
SUBMIT

After clicking 'SUBMIT' to submit, the creation of the Utility Meter is complete.

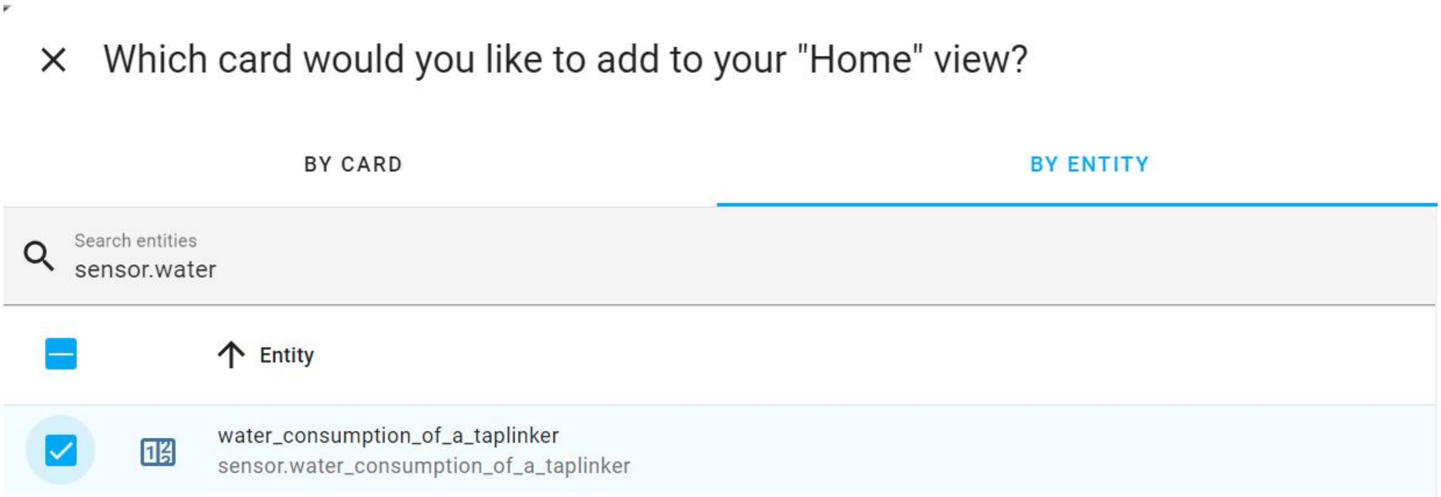


Name	Entity ID	Type
water_consumption_of_a_taplinker	sensor.water_consumption_of_a_taplinker	Utility Meter

At this point, you can go to the 'Overview' interface in Home Assistant, click on the three-dot icon in the upper right corner, then select 'Edit Dashboard,' and finally, click 'ADD CARD' to add a display card.

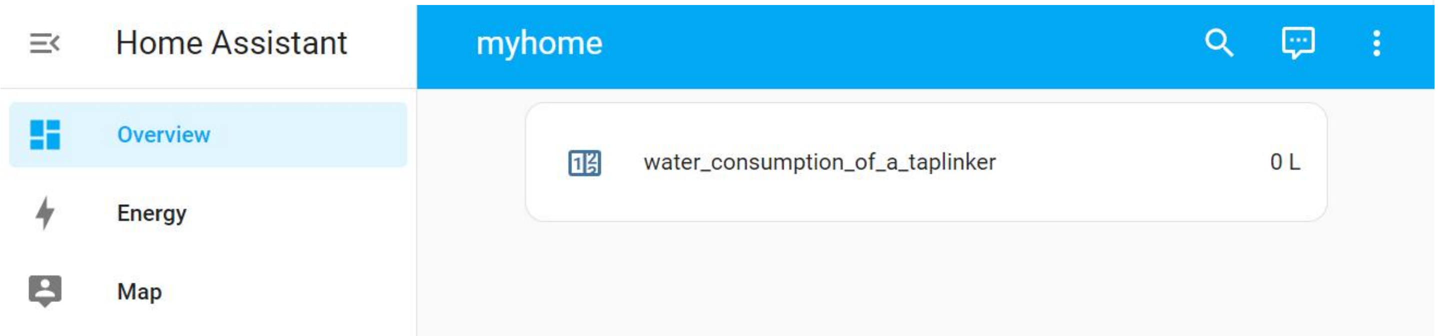


You can quickly locate the corresponding entity by using the name you set during the 'Add Utility Meter' process earlier.



Search entities
sensor.water

☑ water_consumption_of_a_taplinker
sensor.water_consumption_of_a_taplinker

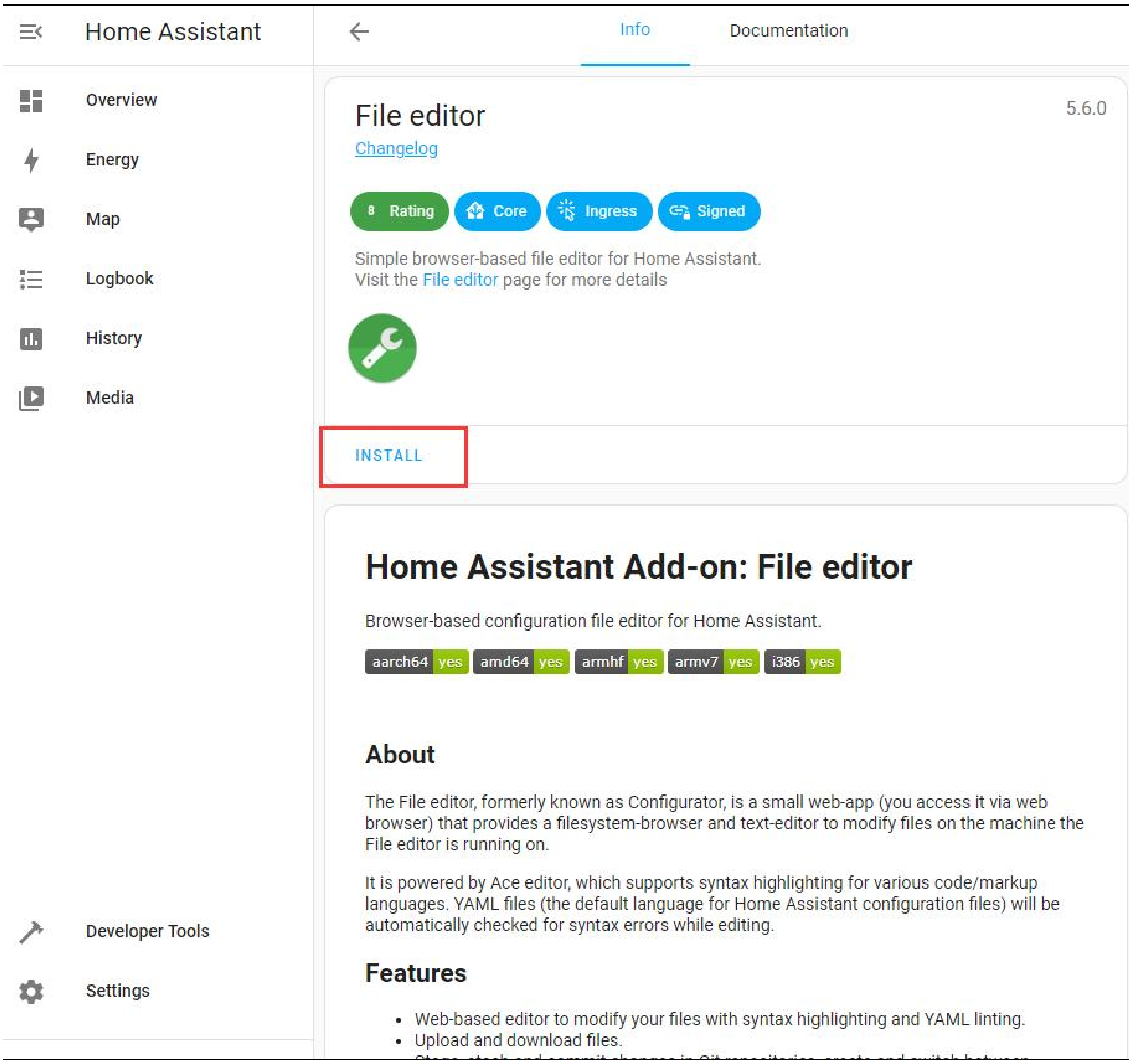


water_consumption_of_a_taplinker 0 L

After following the above steps, you have successfully achieved continuous accumulation of watering volume for a specific Taplinker. If you wish to further track watering statistics on a daily, weekly, monthly, and yearly basis, you can use Home Assistant's Energy dashboard.

Since the Energy dashboard tracks water consumption through a sensor with a 'device_class' type of 'water,' you will need to manually create such a sensor entity in Home Assistant's 'configuration.yaml.'

To edit the 'configuration.yaml' file, you can find and install 'File editor' under 'Settings > Add-ons.'



File editor

Simple browser-based file editor for Home Assistant.
Visit the File editor page for more details

Home Assistant Add-on: File editor

Browser-based configuration file editor for Home Assistant.

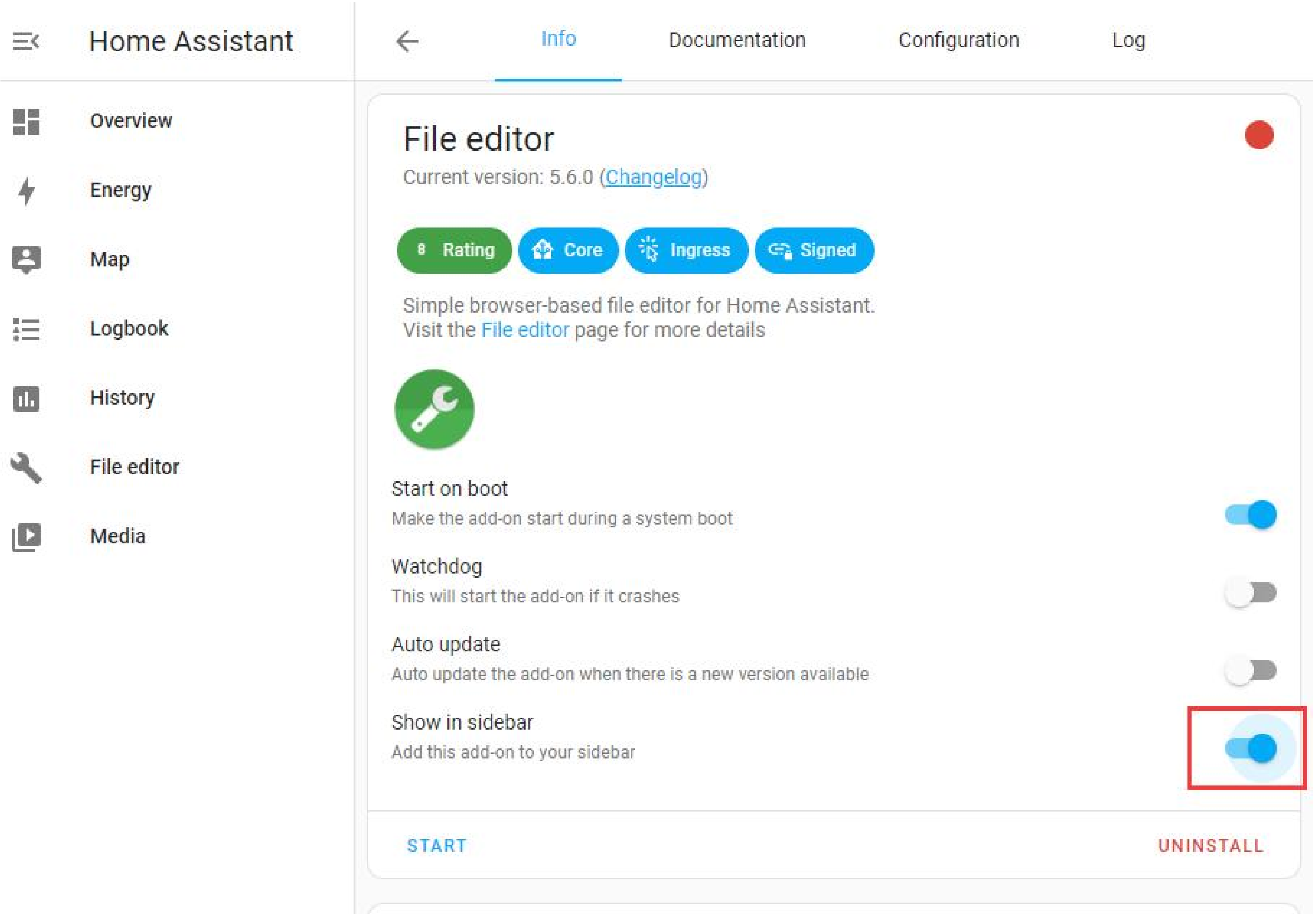
About

The File editor, formerly known as Configurator, is a small web-app (you access it via web browser) that provides a filesystem-browser and text-editor to modify files on the machine the File editor is running on.

It is powered by Ace editor, which supports syntax highlighting for various code/markup languages. YAML files (the default language for Home Assistant configuration files) will be automatically checked for syntax errors while editing.

Features

- Web-based editor to modify your files with syntax highlighting and YAML linting.
- Upload and download files.



File editor

Current version: 5.6.0 (Changelog)

Simple browser-based file editor for Home Assistant.
Visit the File editor page for more details

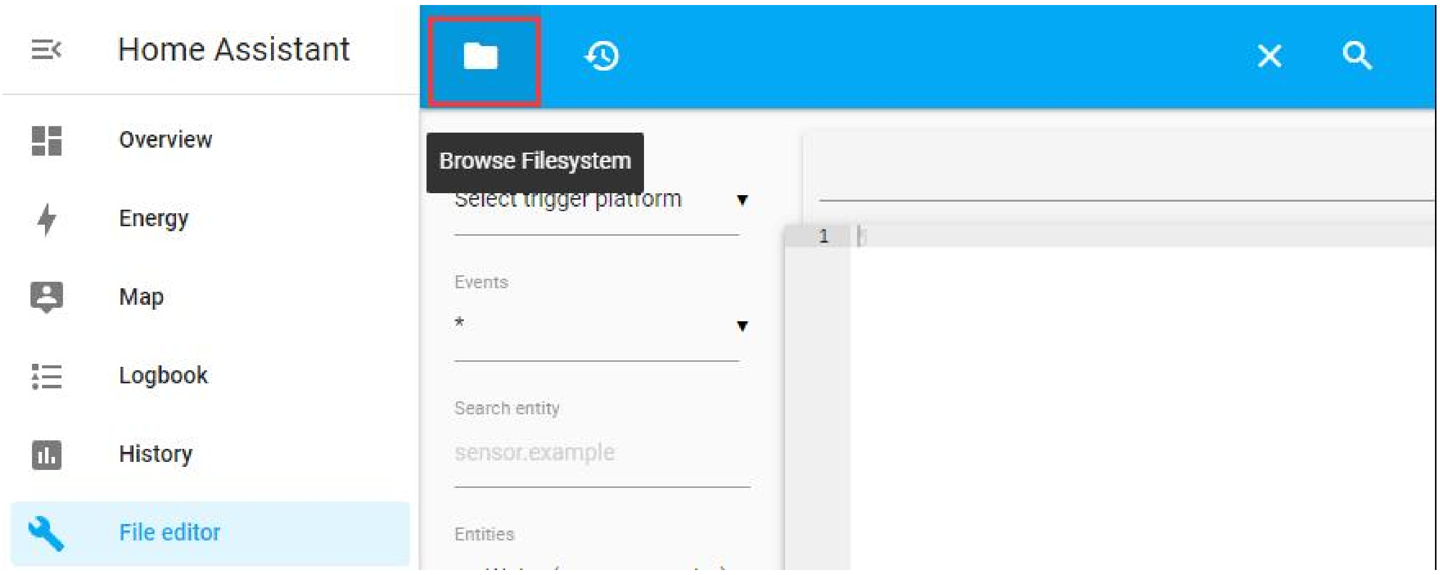
Start on boot
Make the add-on start during a system boot

Watchdog
This will start the add-on if it crashes

Auto update
Auto update the add-on when there is a new version available

Show in sidebar
Add this add-on to your sidebar

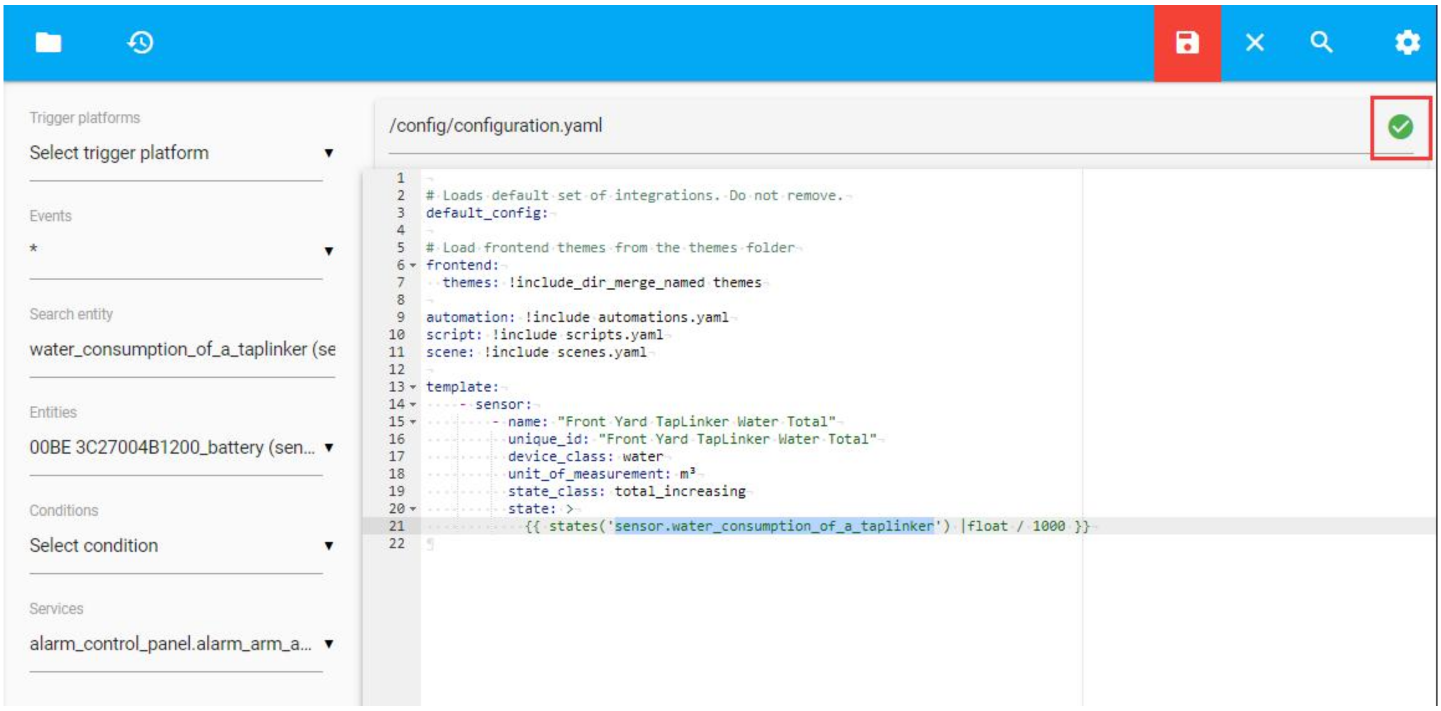
Next, access the 'File editor' from the sidebar, and then click the 'Browse Filesystem' button to locate the 'configuration.yaml' file under the '/config/' directory.



Afterward, at the end of the 'configuration.yaml' file, enter content similar to the following.

```
template:
    - sensor:
        - name: "Front Yard TapLinker Water Total"
          unique_id: "Front Yard TapLinker Water Total"
          device_class: water
          unit_of_measurement: m³
          state_class: total_increasing
          state: >
            {{ states('sensor.water_consumption_of_a_taplinker') |float / 1000 }}
```

Here, you can assign a name and unique_id that you find suitable for the sensor. It's important to note that the 'unit_of_measurement' can be CCF, ft³, gal, L, m³, and you'll need to perform the corresponding conversion in the 'state' field. For example, if the 'volume' unit configured on the gateway management page is 'Liter,' then the corresponding 'unit' for the previously added Utility Meter 'water_consumption_of_a_taplinker' is also 'Liter.' However, this sensor uses 'm³' as the unit, so you'll need to convert the value of the 'water_consumption_of_a_taplinker' entity to a float type and divide it by 1000 in the 'state' expression to convert 'Liter' to 'm³'.


Trigger platforms: Select trigger platform

Events: *

Search entity: water_consumption_of_a_taplinker (se

Entities: 00BE 3C27004B1200_battery (sen...

Conditions: Select condition

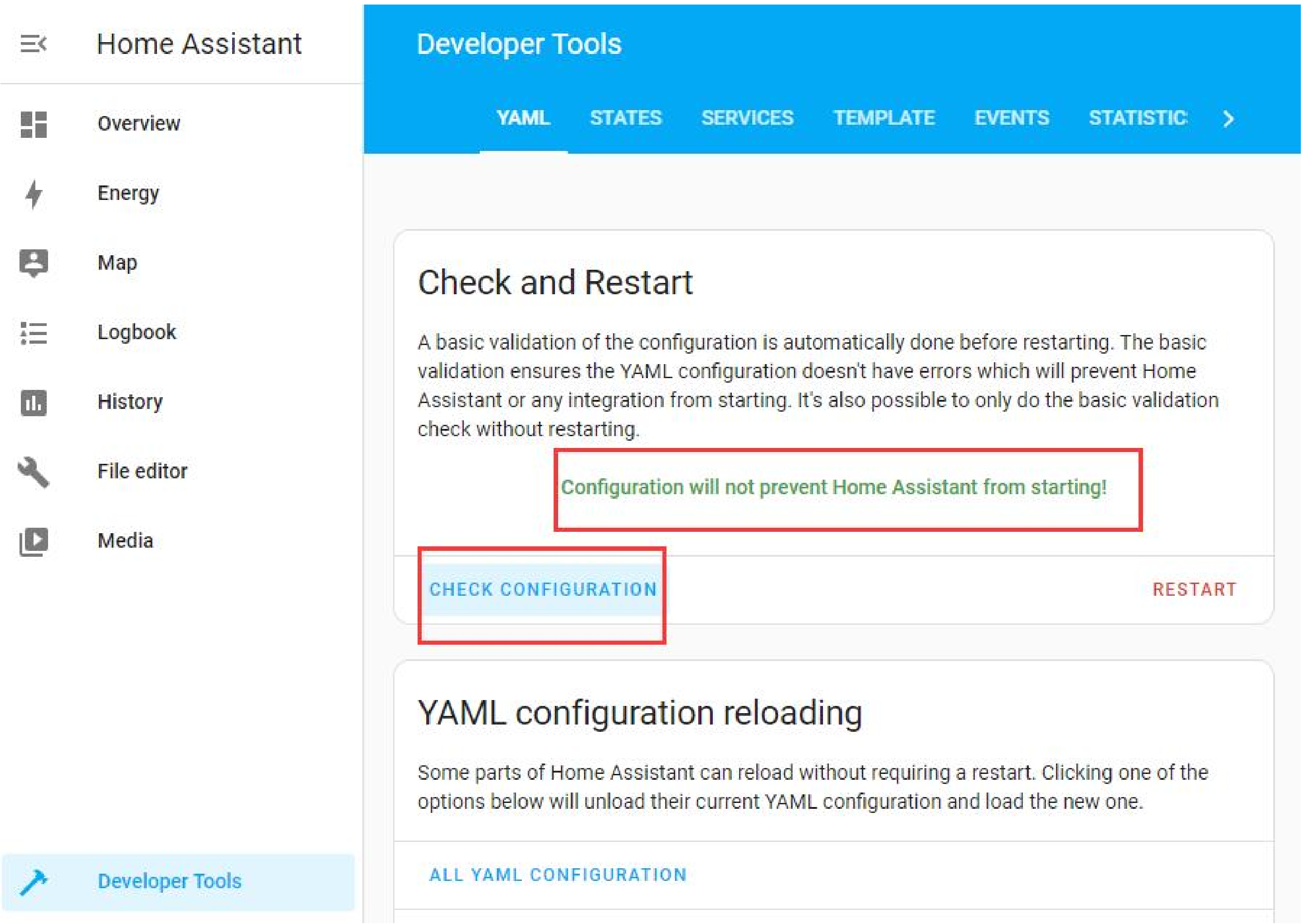
Services: alarm_control_panel.alarm_arm_a...

/config/configuration.yaml

```

# Loads default set of integrations. Do not remove.
default_config:

# Load frontend themes from the themes folder
frontend:
  themes: !include_dir_merge_named themes

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

template:
    - sensor:
        - name: "Front Yard TapLinker Water Total"
          unique_id: "Front Yard TapLinker Water Total"
          device_class: water
          unit_of_measurement: m³
          state_class: total_increasing
          state: >
            {{ states('sensor.water_consumption_of_a_taplinker') |float / 1000 }}

```

After completing the editing, you should check the icon in the upper right corner to ensure that the content you've entered is error-free. Once you've confirmed it's correct, click the red button to save the file.

Afterward, you'll need to go to 'Developer Tools' and click 'CHECK CONFIGURATION' to verify your configuration once more to ensure it won't prevent Home Assistant from starting properly.



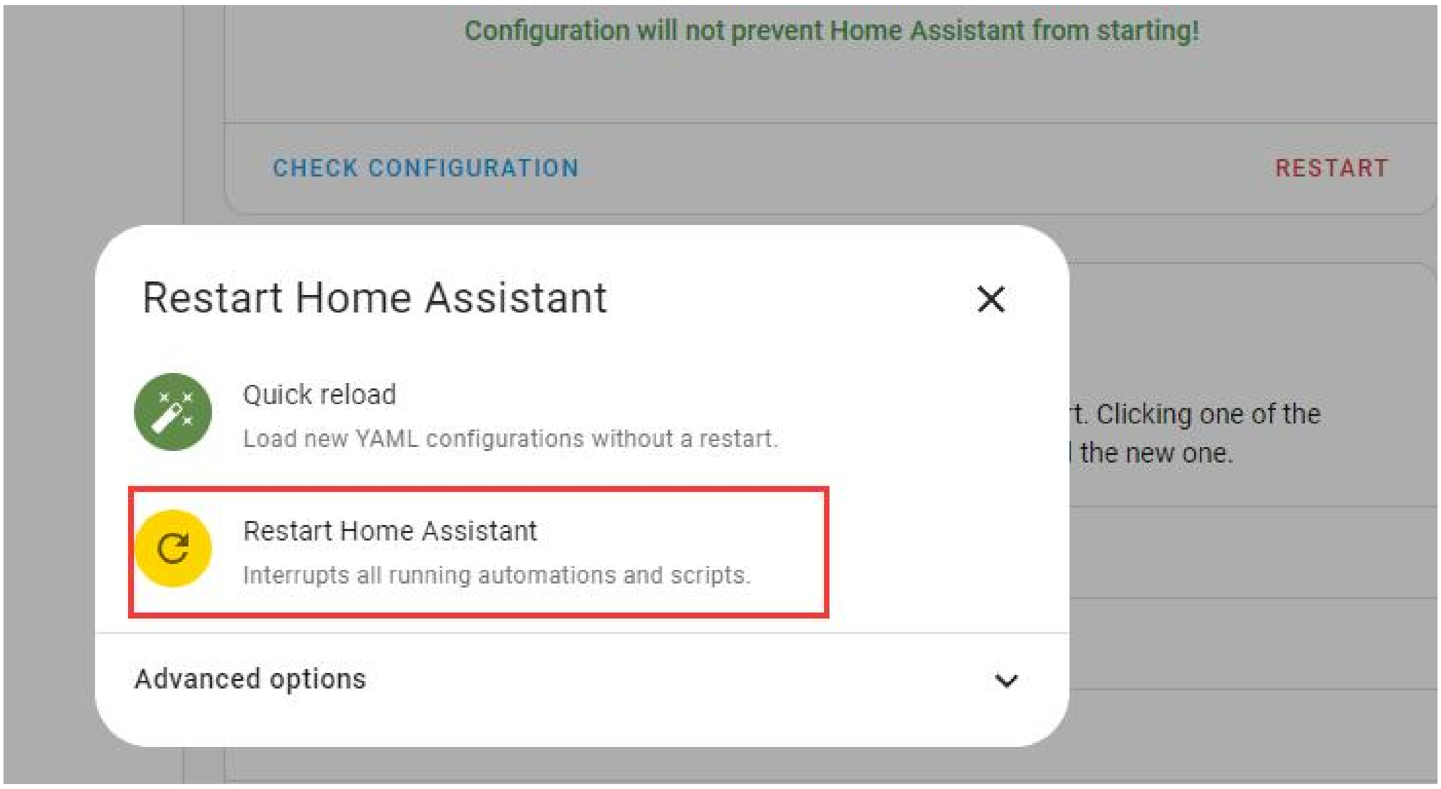
Check and Restart

A basic validation of the configuration is automatically done before restarting. The basic validation ensures the YAML configuration doesn't have errors which will prevent Home Assistant or any integration from starting. It's also possible to only do the basic validation check without restarting.

Configuration will not prevent Home Assistant from starting!

YAML configuration reloading

Some parts of Home Assistant can reload without requiring a restart. Clicking one of the options below will unload their current YAML configuration and load the new one.



Configuration will not prevent Home Assistant from starting!

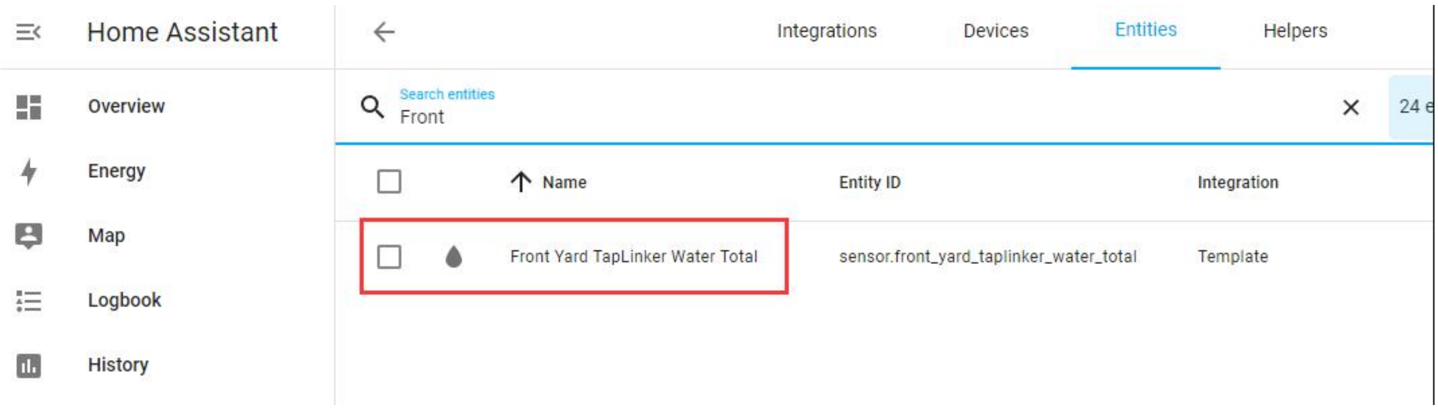
Restart Home Assistant

Load new YAML configurations without a restart.

Interrupts all running automations and scripts.

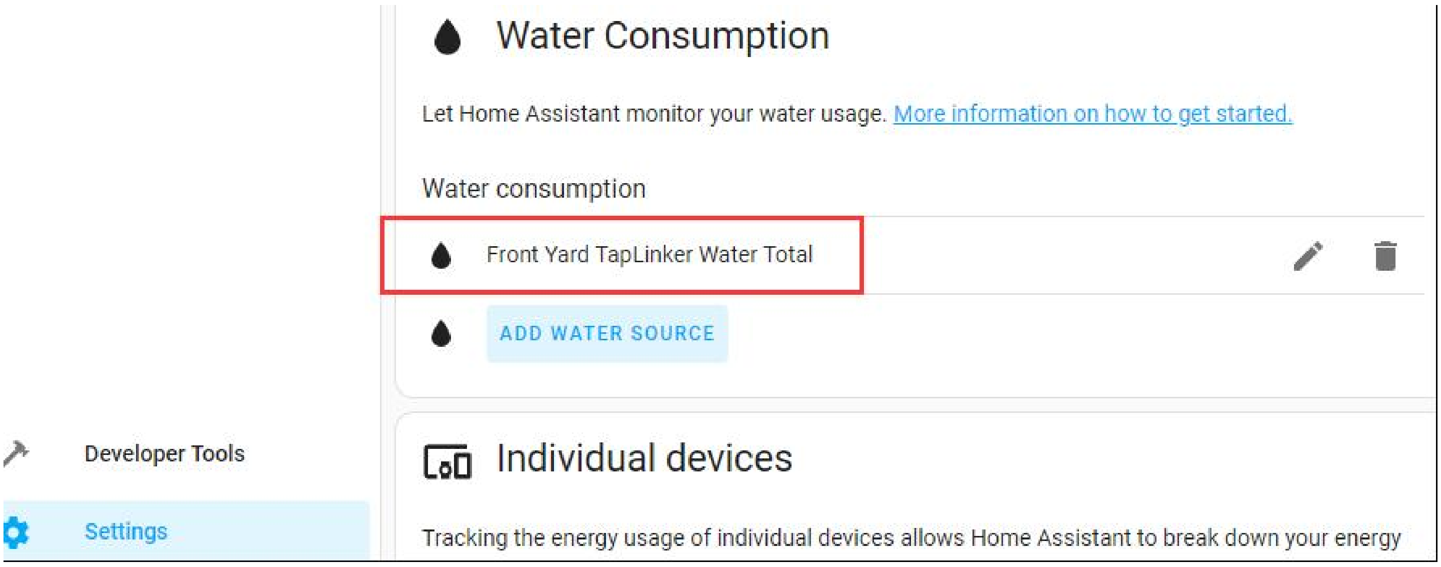
After confirming that the configuration is error-free, click on 'RESTART' to restart the entire Home Assistant and ensure that your configuration is reloaded and takes effect.

Once Home Assistant has restarted, if everything is in order, you will be able to find the previously created sensor entity with 'device_class' type set to 'water' under 'Entities.'



☐		Name	Entity ID	Integration
☐	●	Front Yard TapLinker Water Total	sensor.front_yard_taplinker_water_total	Template

By now, you can use this entity as the data source for water consumption statistics in the Energy dashboard.

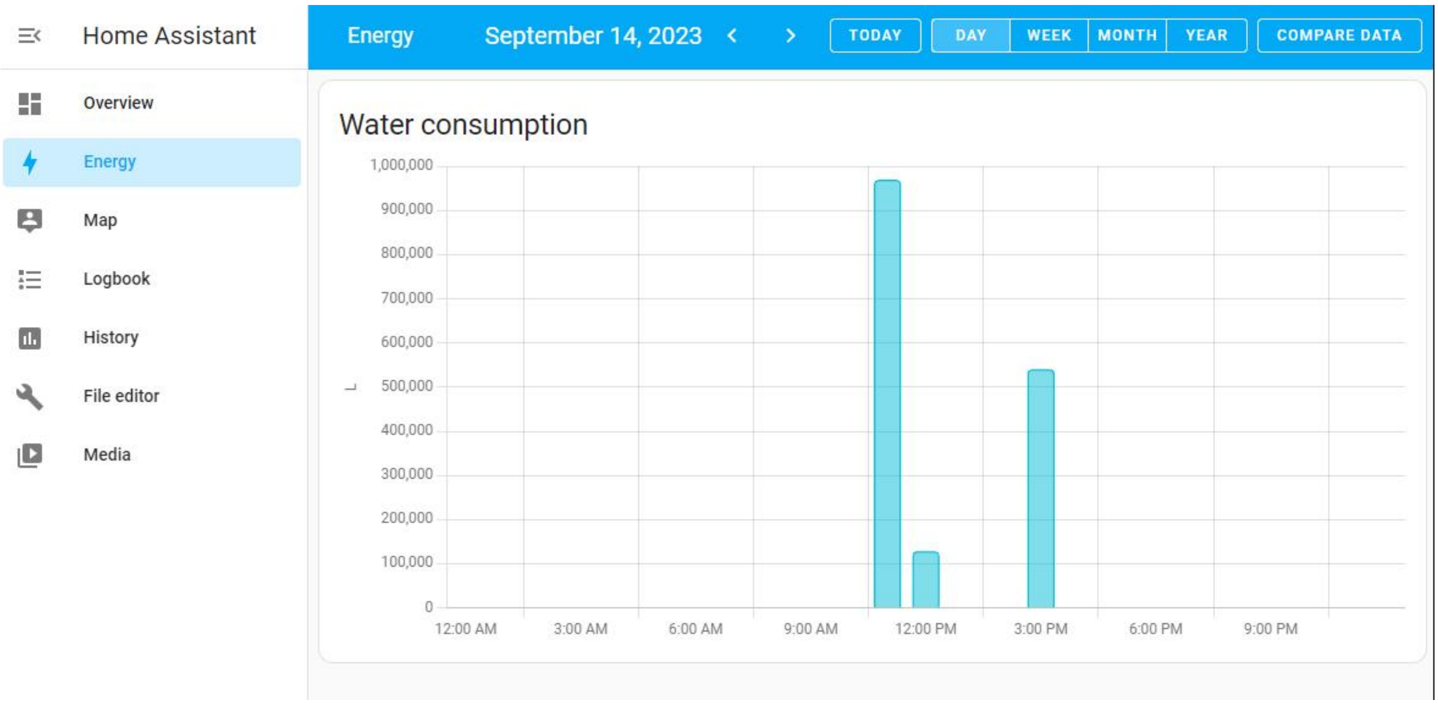


Let Home Assistant monitor your water usage. More information on how to get started.

Water consumption

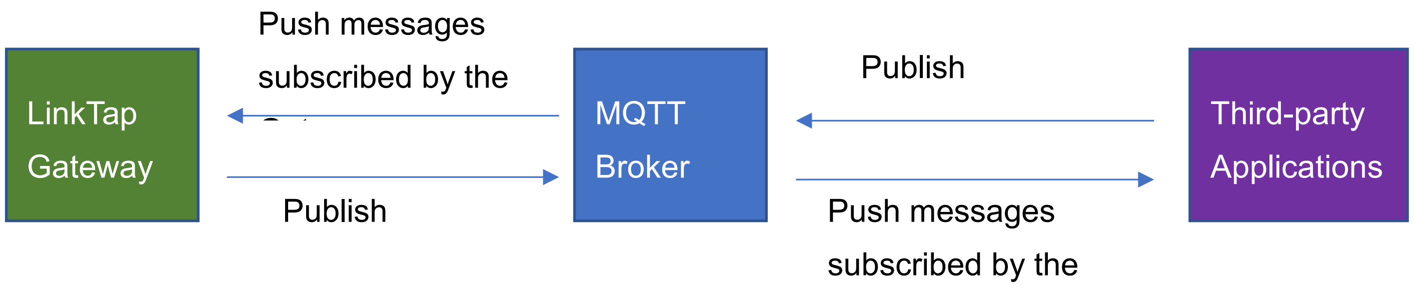
Front Yard TapLinker Water Total

Tracking the energy usage of individual devices allows Home Assistant to break down your energy



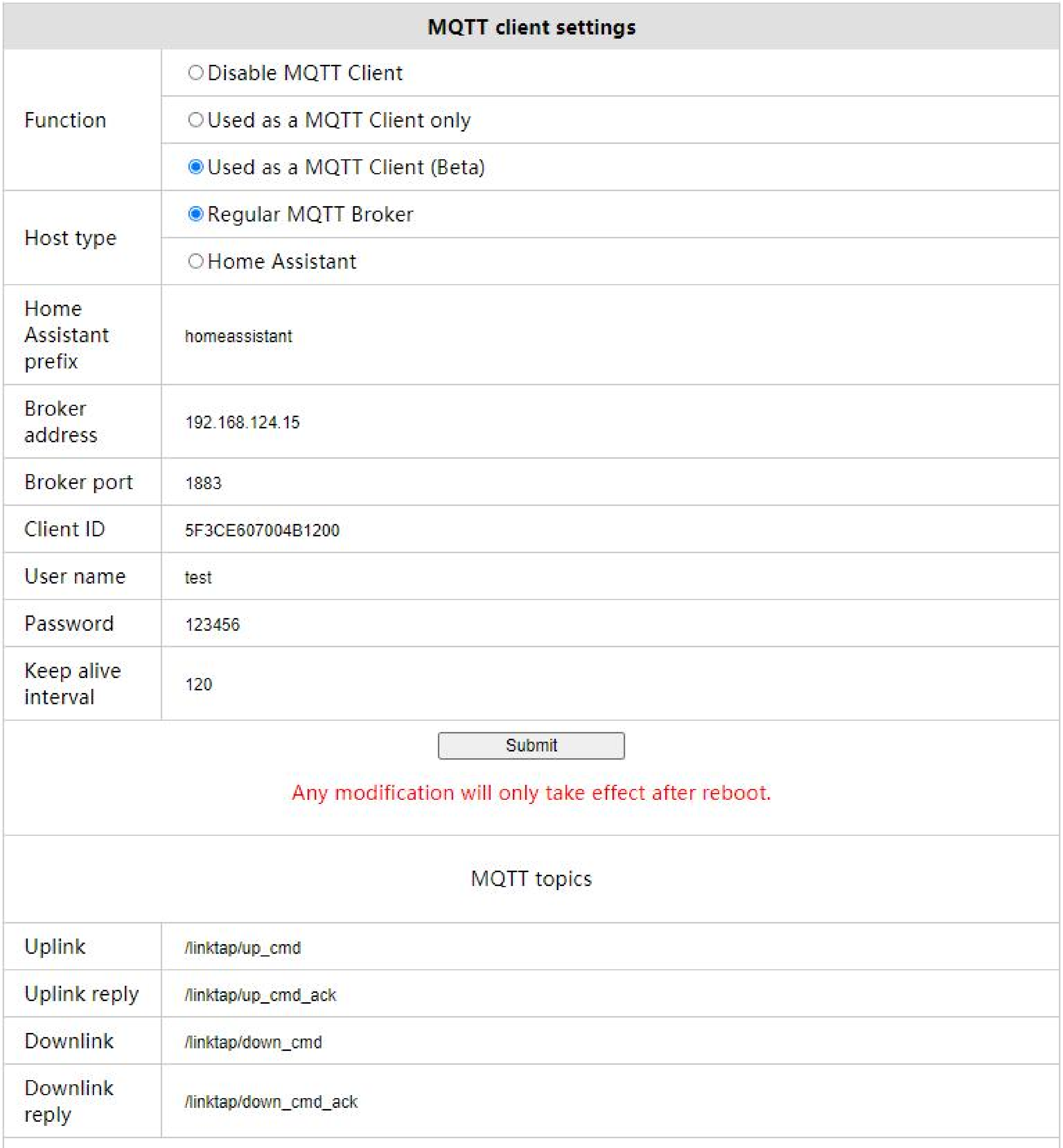
3. Integrate with General MQTT Broker

3.1. System Diagram



The LinkTap Gateway publishes messages through 2 MQTT topics “Uplink” and “Downlink reply”, and receives messages through another 2 MQTT topics "Uplink reply" and "Downlink". These four topics need to be defined first by the user in the Gateway’s web management page before the MQTT client function can be used.

3.2. LinkTap Gateway Settings



MQTT client settings	
Function	○ Disable MQTT Client ○ Used as a MQTT Client only ● Used as a MQTT Client (Beta)
Host type	● Regular MQTT Broker ○ Home Assistant
Home Assistant prefix	homeassistant
Broker address	192.168.124.15
Broker port	1883
Client ID	5F3CE607004B1200
User name	test
Password	123456
Keep alive interval	120

Any modification will only take effect after reboot.

MQTT topics	
Uplink	/linktap/up_cmd
Uplink reply	/linktap/up_cmd_ack
Downlink	/linktap/down_cmd
Downlink reply	/linktap/down_cmd_ack

As shown above,

- “Host type”: need to be “Regular MQTT Broker”.
- “Broker address”: Required. The address of your MQTT broker.
- “Broker port”: Required. The port number of your MQTT broker.
- “Client ID”: Required. A unique ID to distinguish the MQTT Client. You can enter your LinkTap Gateway’s ID here.
- “User name”: The user’s name required to access the MQTT broker.
- “Password”: The password required to access the MQTT broker.
- “Keep alive interval”: Depending on your network environment and the communication requirements between the Gateway and the Broker, it is recommended to set it in the range of 30~120 seconds.

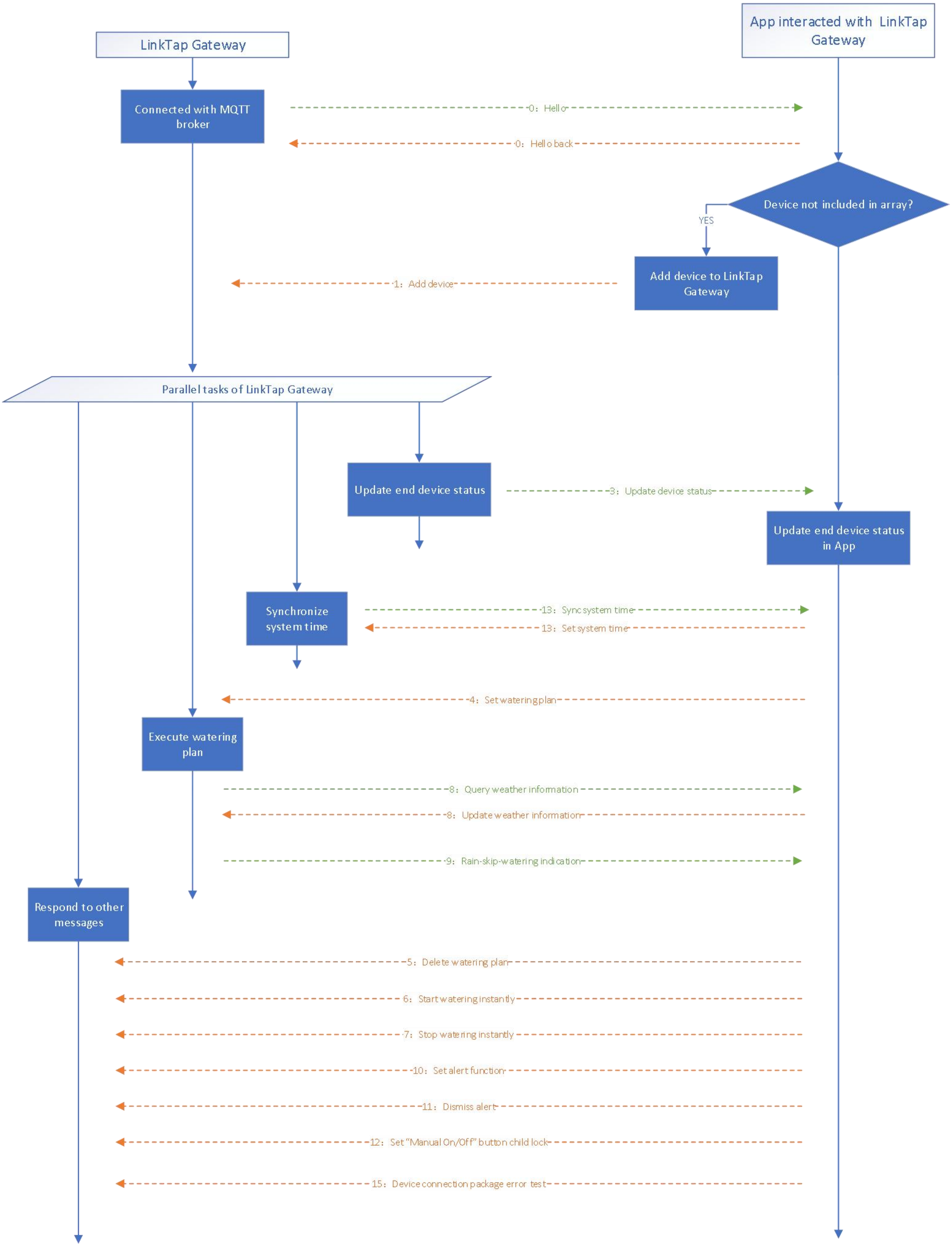
Click “Submit” button to save the settings. Next, we need to configure the MQTT topics.

- “Uplink topic”: Required. The Gateway will publish the status information of the Gateway and its associated terminal devices through this topic, and your application will subscribe to this topic to receive the information published by the Gateway. The content of this field needs to be defined according to the hierarchical structure of your MQTT topic.
- “Uplink reply topic”: Optional. The Gateway receives the processing result of your application through this topic.
- “Downlink topic”: Required. The Gateway subscribes to this topic to receive the downlink data to configure the Gateway and water timers. Your application publishes configuration information through this topic.
- “Downlink reply topic”: Optional. The Gateway publishes the processing results of the downlink data through this topic. Your application receives the downlink data processing results through this topic.

3.3. LinkTap Gateway MQTT Messages Interaction

Please refer to “LinkTap Gateway MQTT Client Interaction Messages.pdf” for the message details.

The graph below shows how the Gateway interacts with the application.


LinkTap Gateway
App interacted with LinkTap Gateway
Connected with MQTT broker
0: Hello
0: Hello back
Device not included in array?
YES
Add device to LinkTap Gateway
1: Add device
Parallel tasks of LinkTap Gateway
Update end device status
3: Update device status
Update end device status in App
Synchronize system time
13: Sync system time
13: Set system time
4: Set watering plan
Execute watering plan
8: Query weather information
8: Update weather information
9: Rain-skip-watering indication
Respond to other messages
5: Delete watering plan
6: Start watering instantly
7: Stop watering instantly
10: Set alert function
11: Dismiss alert
12: Set "Manual On/Off" button child lock
15: Device connection package error test

3.4. Example of the Interaction

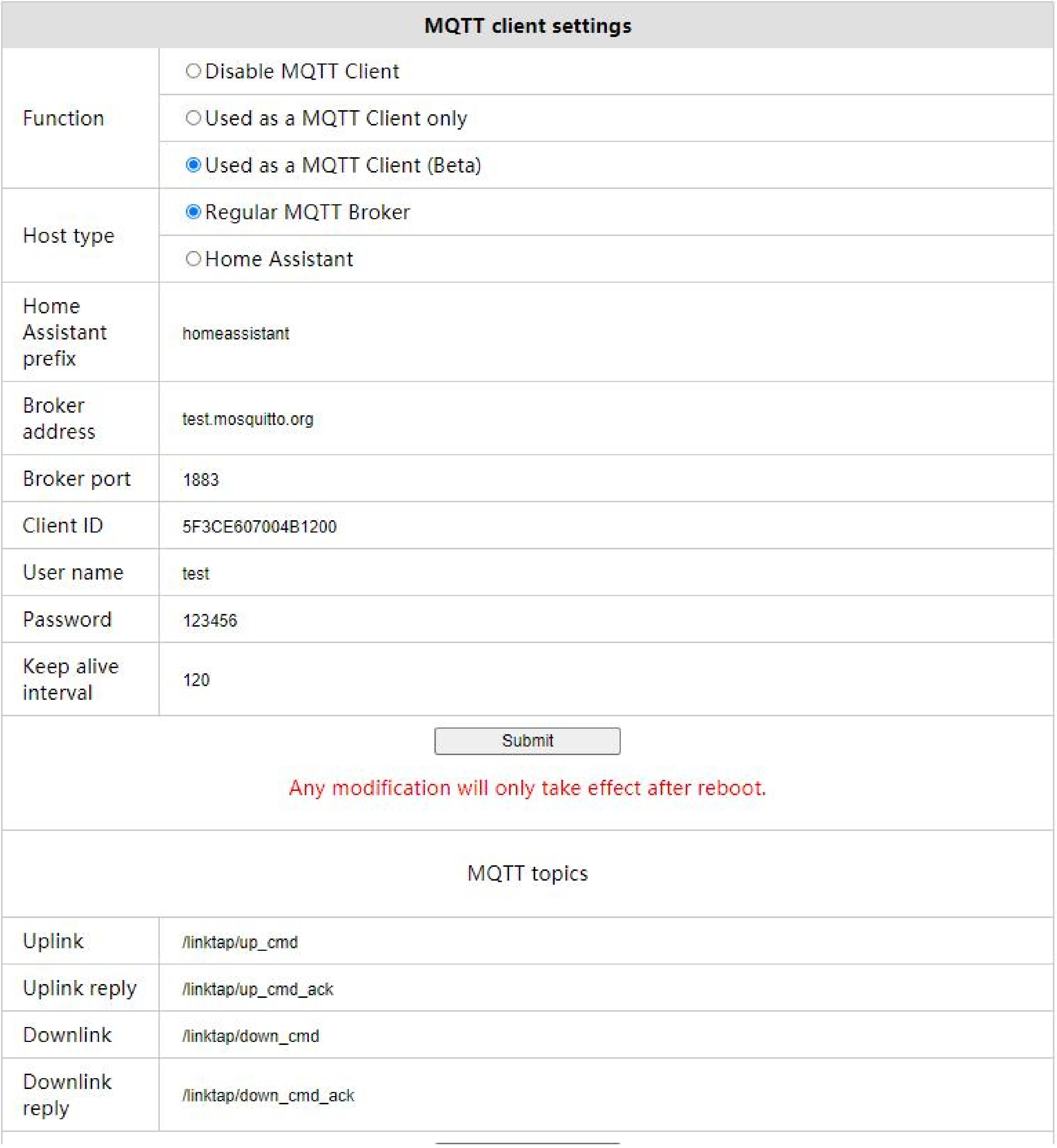
If you don't have your own MQTT Broker, you can choose a public MQTT Broker server to quickly set up a prototype to test. Here "test.mosquitto.org" is used in the example. In addition, we use the MQTT client tool "MQTT.fx" to act as a third-party application.

3.4.1. Set Up MQTT Client

Set up MQTT client on LinkTap Gateway’s admin page as shown below.

Note 1: Please set a random value for Client ID, User name, Password and Keep alive interval when using “test.mosquitto.org”.

Note 2: Don’t forget to Reboot Gateway after submitting the changes.



MQTT client settings

Function	○ Disable MQTT Client
	○ Used as a MQTT Client only
	● Used as a MQTT Client (Beta)
Host type	● Regular MQTT Broker
	○ Home Assistant
Home Assistant prefix	homeassistant
Broker address	test.mosquitto.org
Broker port	1883
Client ID	5F3CE607004B1200
User name	test
Password	123456
Keep alive interval	120

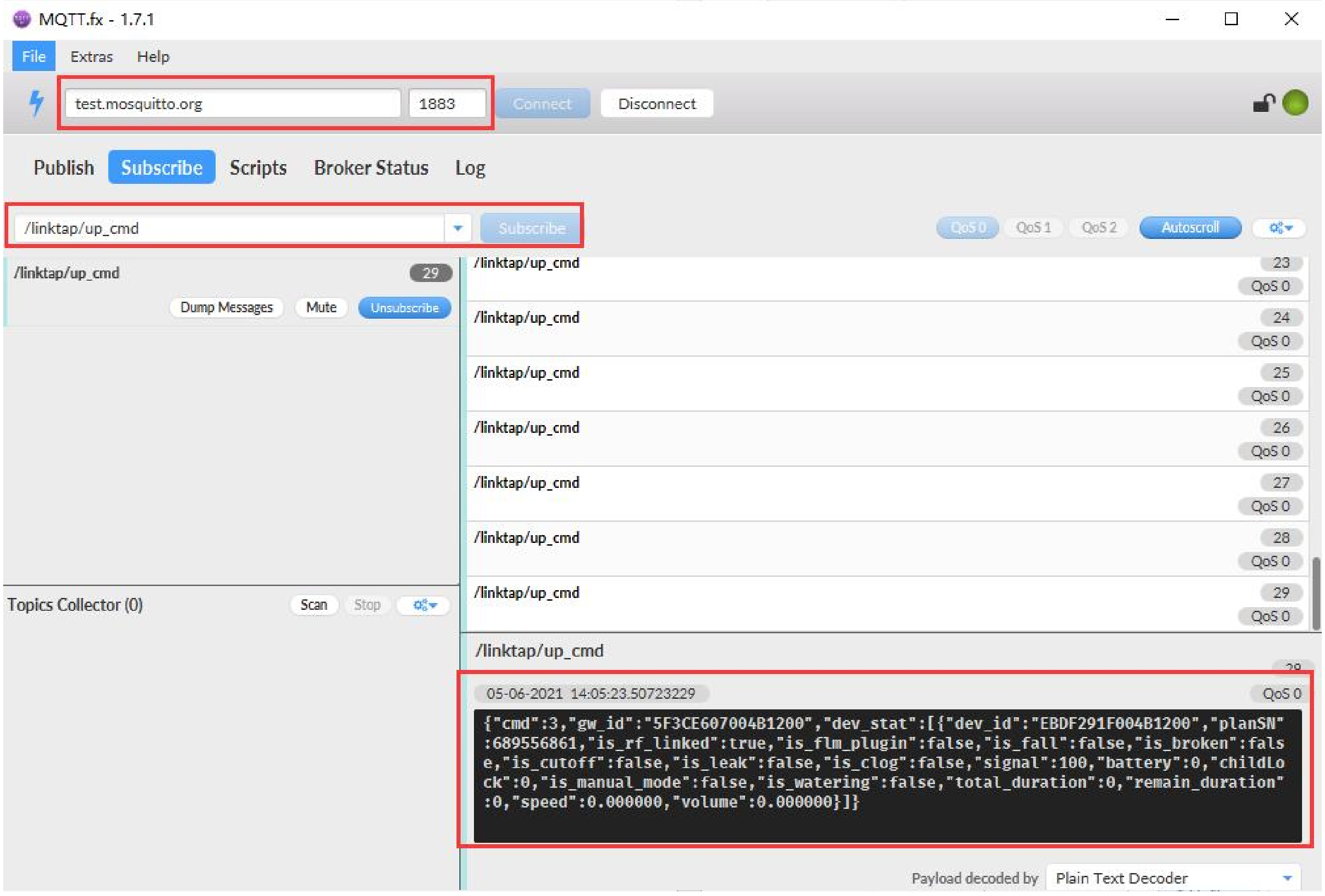
Any modification will only take effect after reboot.

MQTT topics

Uplink	/linktap/up_cmd
Uplink reply	/linktap/up_cmd_ack
Downlink	/linktap/down_cmd
Downlink reply	/linktap/down_cmd_ack

3.4.2. MQTT.fx Receives Messages from Gateway

Following the Gateway MQTT Client's settings, enter the corresponding MQTT Broker address and port on MQTT.fx. After connection is established, subscribe to the configured Uplink topic to receive the messages sent by the Gateway.



test.mosquitto.org 1883

/linktap/up_cmd

/linktap/up_cmd 29

```
/linktap/up_cmd    23  QoS 0
/linktap/up_cmd    24  QoS 0
/linktap/up_cmd    25  QoS 0
/linktap/up_cmd    26  QoS 0
/linktap/up_cmd    27  QoS 0
/linktap/up_cmd    28  QoS 0
/linktap/up_cmd    29  QoS 0
```

/linktap/up_cmd

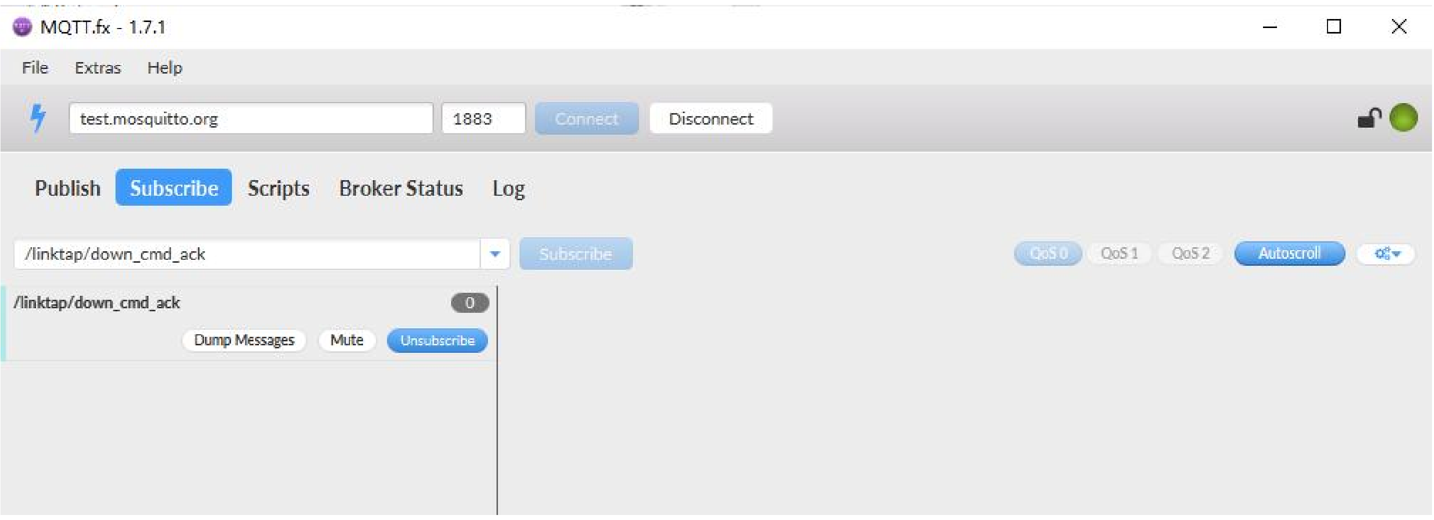
05-06-2021 14:05:23.50723229 QoS 0

```
{"cmd":3,"gw_id":"5F3CE607004B1200","dev_stat":[{"dev_id":"EBDF291F004B1200","planSN"
:689556861,"is_rf_linked":true,"is_flm_plugin":false,"is_fall":false,"is_broken":fals
e,"is_cutoff":false,"is_leak":false,"is_clog":false,"signal":100,"battery":0,"childLo
ck":0,"is_manual_mode":false,"is_watering":false,"total_duration":0,"remain_duration"
:0,"speed":0.000000,"volume":0.000000}]}
```

Payload decoded by Plain Text Decoder

3.4.3. MQTT.fx Sends Messages to Gateway, and Receives Responses from Gateway

Following the Gateway MQTT Client's settings, enter the corresponding MQTT Broker address and port on MQTT.fx. After connection is established, subscribe to the configured Downlink topic reply to receive the messages sent from the Gateway.

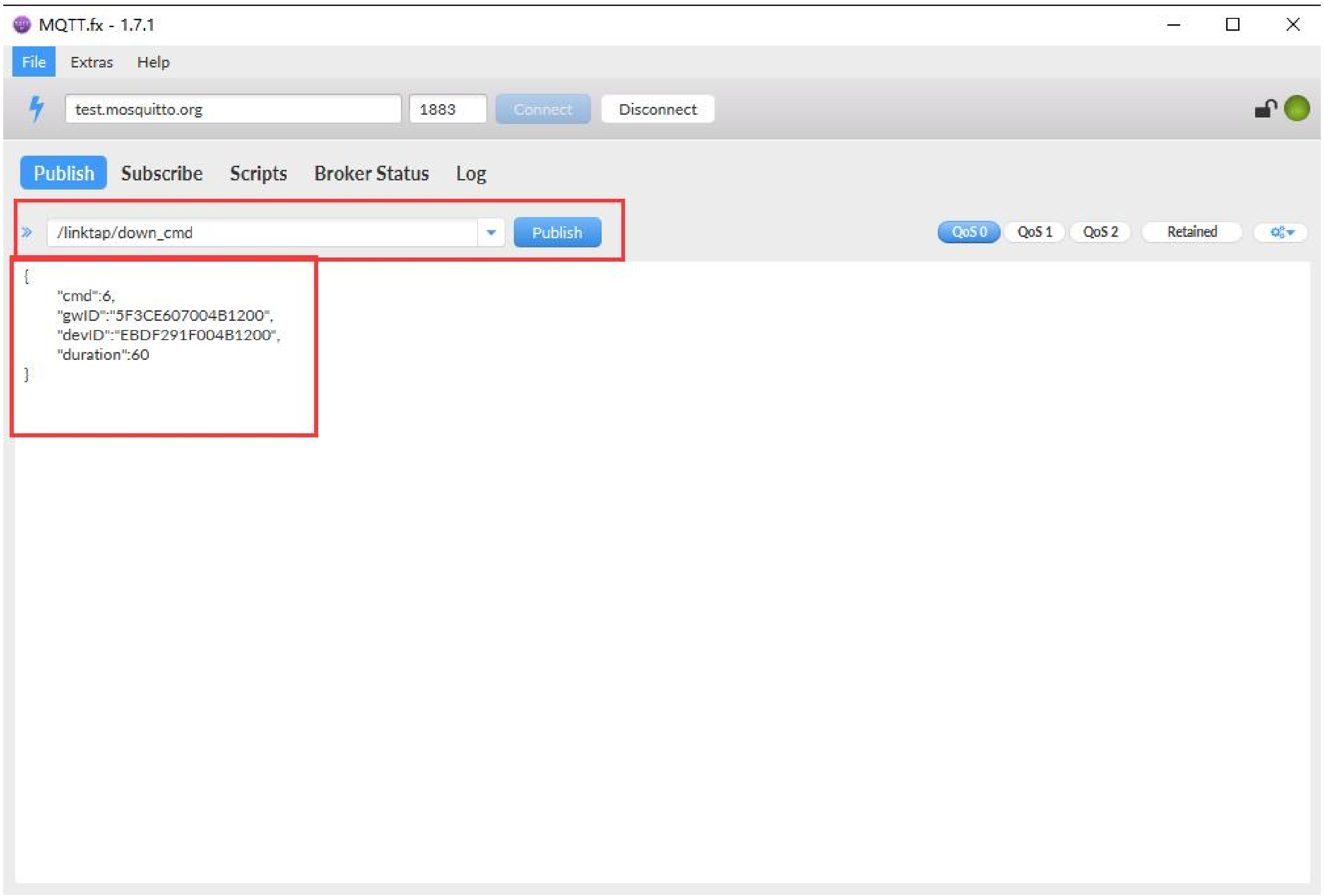


test.mosquitto.org 1883

/linktap/down_cmd_ack

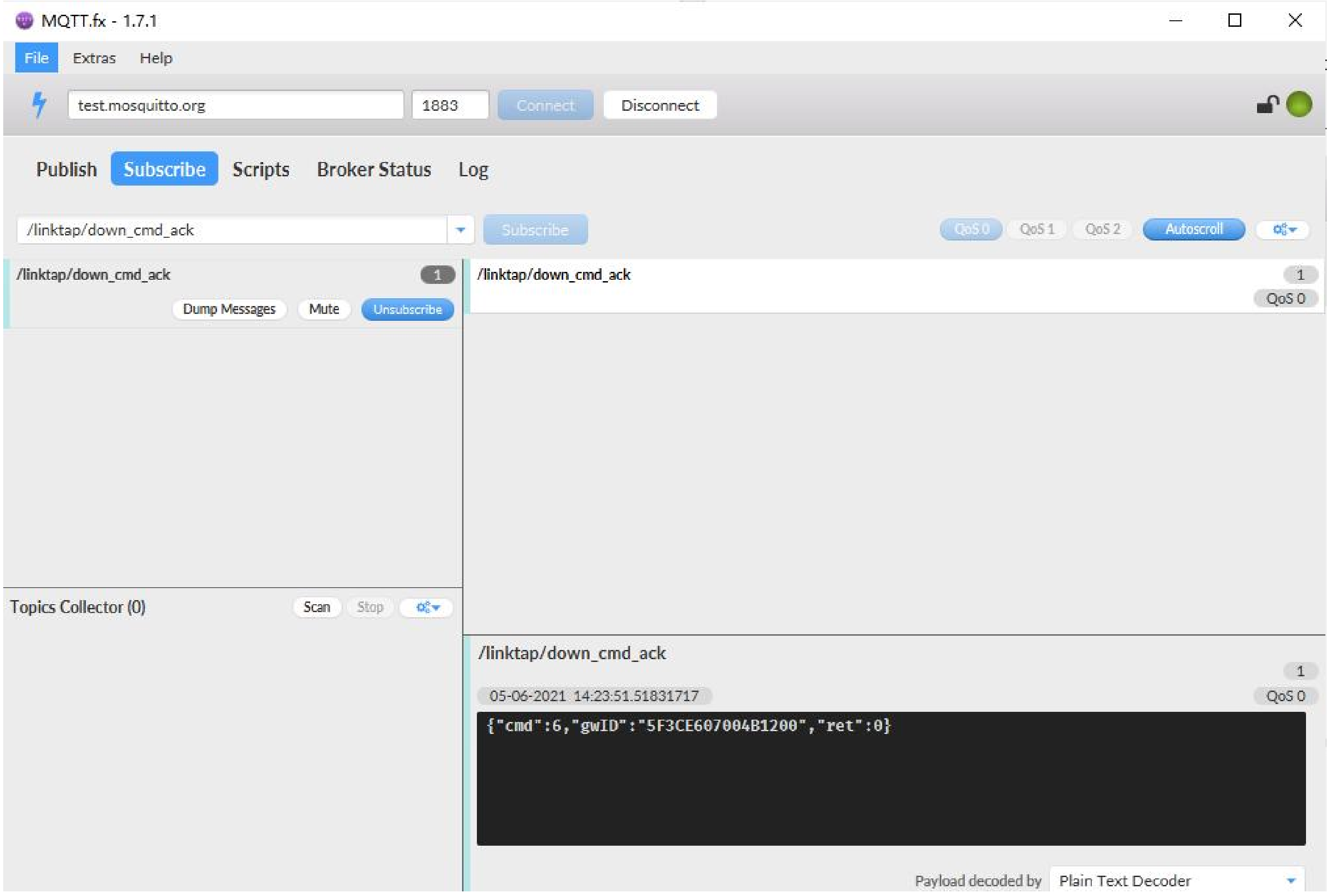
/linktap/down_cmd_ack 0

Then, following the message definition in "LinkTap Gateway MQTT Client Interaction Messages.pdf", MQTT.fx publishes the message through the Downlink topic, then the MQTT broker sends the corresponding message to the Gateway. After the message arrives at the Gateway successfully, MQTT.fx will receive the Gateway's response message.



/linktap/down_cmd

```
{
    "cmd":6,
    "gwID":"5F3CE607004B1200",
    "devID":"EBDF291F004B1200",
    "duration":60
}
```



/linktap/down_cmd_ack

05-06-2021 14:23:51.51831717

```
{"cmd":6,"gwID":"5F3CE607004B1200","ret":0}
```

Payload decoded by Plain Text Decoder

3.5. Implementation of Third-Party Application

There are two ways to program and execute watering plan in third-party applications.

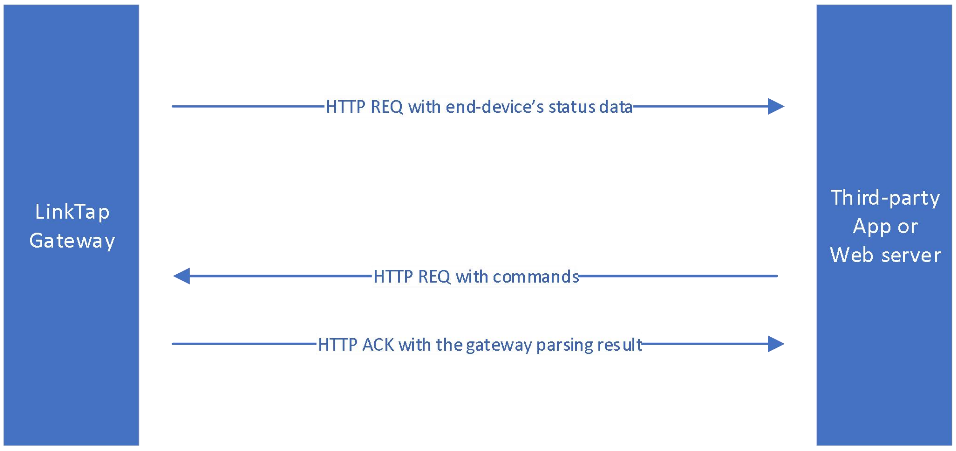
3.5.1. Program and execute watering plan within the application

In this implementation, the third-party application does not use CMD: 4 (refer to LinkTap Gateway MQTT Client Interaction Messages.pdf for command details) to send the whole watering plan to the Gateway. Instead, the third-party application programs the watering plan, and also decides when to start watering (through CMD: 6), when to stop watering (through CMD: 7), and when to skip watering for rainy days.

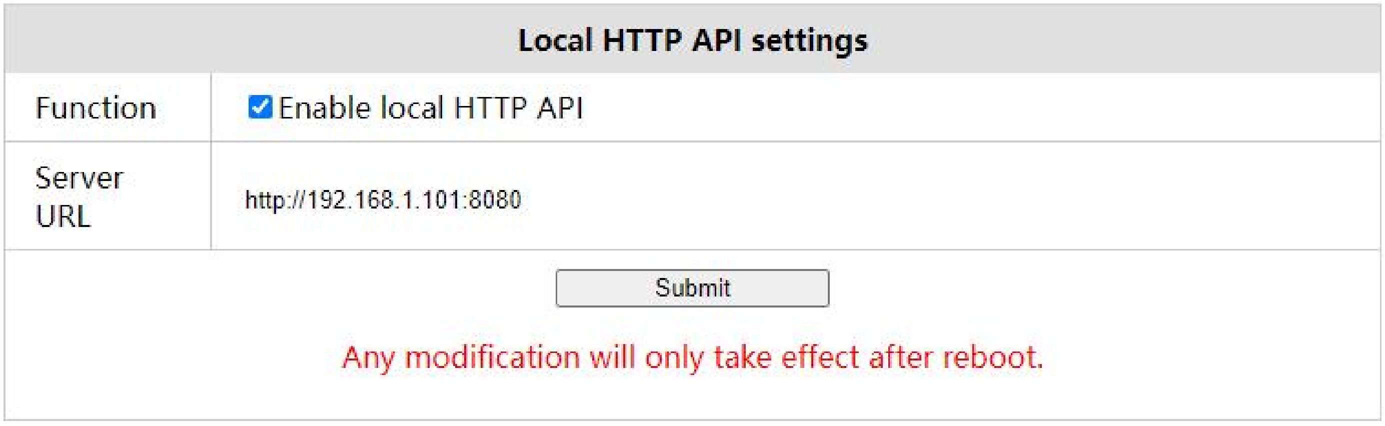
3.5.2. Program watering plan within the application, and execute the plan in the Gateway

In this implementation, the third-party application programs the watering plan, then sends the whole plan to the Gateway through CMD: 4. The Gateway will then trigger watering based on the received plan, regardless of whether the Gateway has constant network connection with the application.

4. Integrate with HTTP API


LinkTap Gateway
HTTP REQ with end-device's status data
HTTP REQ with commands
HTTP ACK with the gateway parsing result
Third-party App or Web server

When neither Internet access nor MQTT broker is available, third-party application can interact with the Gateway through HTTP commands. To do so, the user first needs to enable the "local HTTP API" function and set a proper HTTP server address on the Gateway's admin page. The format of the HTTP server address should be http://Your_Server_Domain_or_IP_address: Service_Port/Path. Some examples are http://www.xxxxxx.com:80/a/b/c , http://192.168.1.101:8080/ , http://192.168.1.101 .



Local HTTP API settings

Function	☑ Enable local HTTP API
Server URL	http://192.168.1.101:8080

Any modification will only take effect after reboot.

4.1. Push data to server from the gateway

After the local HTTP API function is enabled and a correct server URL is set, the Gateway will automatically connect to the HTTP server. Then when the status of the end devices under the Gateway changes, the Gateway will send the device status data in JSON format to the HTTP server via a HTTP POST request.

HTTP Header:

```
{
  content-type: application/json; charset=UTF-8
  host: Your_HTTP_Server_Domain_or_IP:Service_Port
  connection: keep-alive
  content-length: N
}
```

HTTP Data:

```
{"dev_id":"aaaabbbbccccdddd",……}
```

For the detailed message format, refer to CMD 3 in “LinkTap Gateway MQTT Client Interaction Messages_x.x.pdf" document.

4.2. Send Commands via HTTP to the gateway for end-device control

Third-party application can send commands to the Gateway's URL via HTTP POST requests to control the end devices. The Gateway's URL is http://Your_Gateway_IP_Address/api.shtml.

HTTP Header：

```
{
POST /api.shtml HTTP/1.1
Content-Type: application/json
Content-Length: 78
Host: 192.168.1.107
Connection: close
}
```

HTTP Data:

```
{"dev_id":"aaaabbbbccccdddd",……}
```

For the detailed message format, refer to message definitions in "LinkTap Gateway MQTT Client Interaction Messages_x.x.pdf" where "Message direction" is "App->Broker->GW".

When the Gateway receives the HTTP request, it will parse and process the JSON data of the HTTP request, and then reply to the third-party application. The format of the reply message is as follows:

HTTP Header：

```
{
HTTP/1.0 200 OK
Content-Type: text/html
Content-Length: 78
Server: LinkTap Gateway
Pragma: no-cache
}
```

HTTP Data:

```
<html>
<head><title>api</title></head>
<body>
    <!--#RET--> {"cmd":x,"gw_id":"CCCCDDDDEEEEFFFF","ret":y}
</body>
</html>
```

For the JSON data format in the reply message, refer to "LinkTap Gateway MQTT Client Interaction Messages_x.x.pdf”.

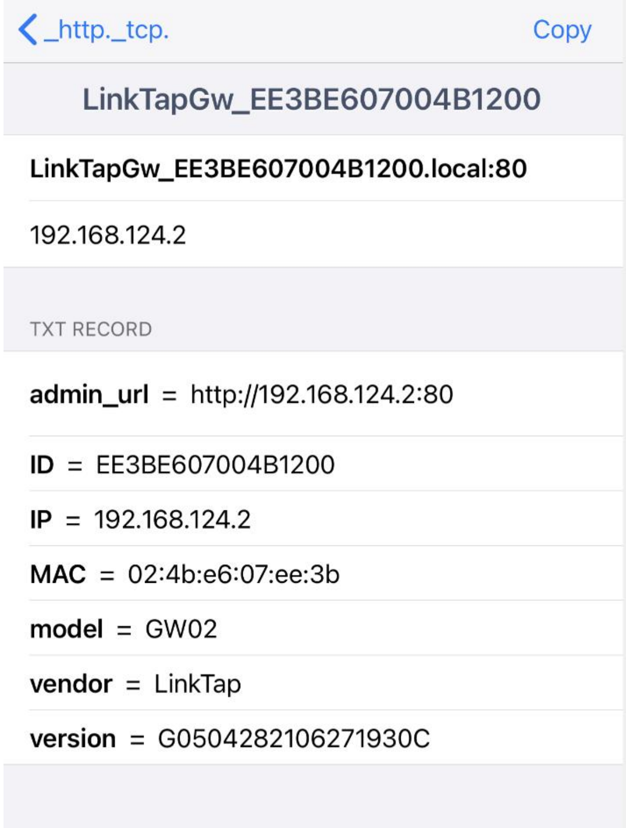
5. Zero Configuration Device Discovery

The LinkTap Gateway contains a mDNS Responder to respond to mDNS Query initiated by third-party applications. In the local area network, if the device has enabled the mDNS (multicast DNS) service, the devices will use the multicast address 224.0.0.251 to exchange information with each other through port 5353. For more information about mDNS, please refer to: Multicast DNS (http://www.multicastdns.org).

Below shows the LinkTap Gateway's mDNS answer message content.

ITEM	CONTENT	REMARK
_services._dns-sd._udp.local type PTR	_http._tcp.local	
<service>.<proto>.local type PTR	LinkTapGw_XXX._http._tcp.local	"XXX" represents the Gateway ID.
<name>.<service>.<proto>.local type SRV	LinkTapGw_XXX.local	
<name>.<service>.<proto>.local type TXT	model：The model of the Gateway. ID: The gateway ID. MAC: The ethernet MAC address of the Gateway. IP: The IP address of the Gateway. admin_url: The URL of the Gateway configuration web site. vendor: LinkTap. version: The firmware version of the Gateway.	

After the LinkTap Gateway obtains the IP address in the local area network, the user can view Gateway's information through an application with built-in mDNS service in the same local area network. Below shows an example.



LinkTapGw_EE3BE607004B1200

LinkTapGw_EE3BE607004B1200.local:80

192.168.124.2

TXT RECORD

admin_url = http://192.168.124.2:80

ID = EE3BE607004B1200

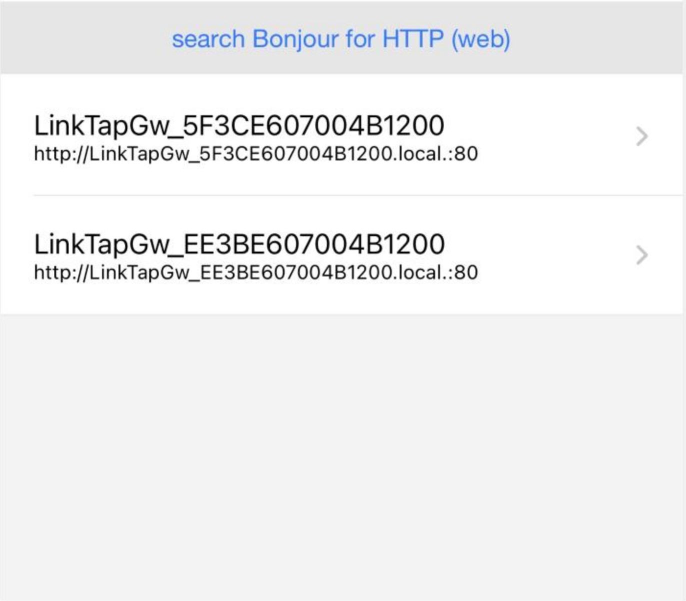
IP = 192.168.124.2

MAC = 02:4b:e6:07:ee:3b

model = GW02

vendor = LinkTap

version = G0504282106271930C



search Bonjour for HTTP (web)

LinkTapGw_5F3CE607004B1200
http://LinkTapGw_5F3CE607004B1200.local.:80

LinkTapGw_EE3BE607004B1200
http://LinkTapGw_EE3BE607004B1200.local.:80