LinkTap Gateway MQTT Client Integration

V2.1

Disclaimer

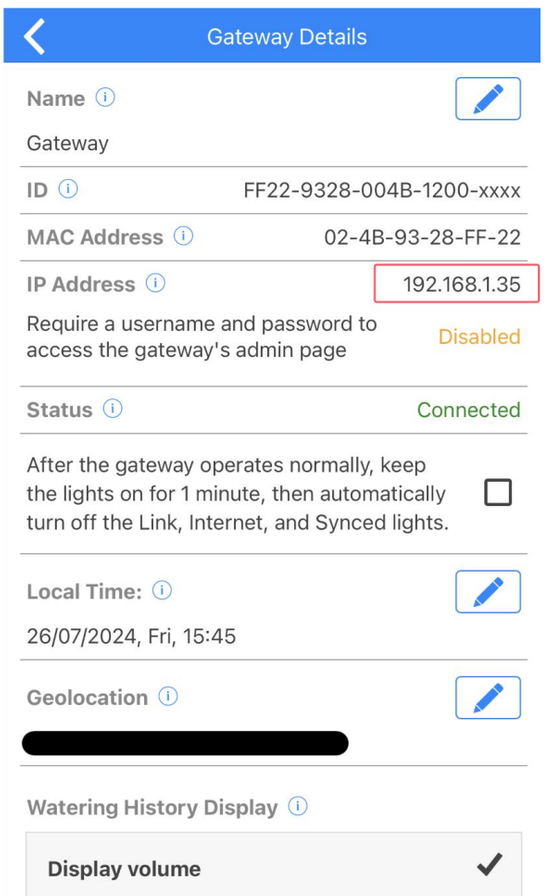
The use of the MQTT/HTTP API function of the LinkTap gateway is considered as user's secondary development or DIY actions. It requires a comprehensive understanding of the device, including its principles and associated risks. LinkTap has taken every measure to ensure the reliable operation of the gateway's MQTT/HTTP API function. However, due to various real-world factors beyond our control (such as network delays or interruptions, equipment malfunctions, battery quality issues, misuse, defects in user-written or third-party scripts or programs, etc.), abnormalities in the API function may occur, leading to unexpected device behavior. Use the LinkTap MQTT/HTTP API functions at your own risk. LinkTap and its authorized agents are not liable for any potential losses resulting from the use of the MQTT/HTTP API.

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"
V2.1	2024/08/02	Update and optimize document content

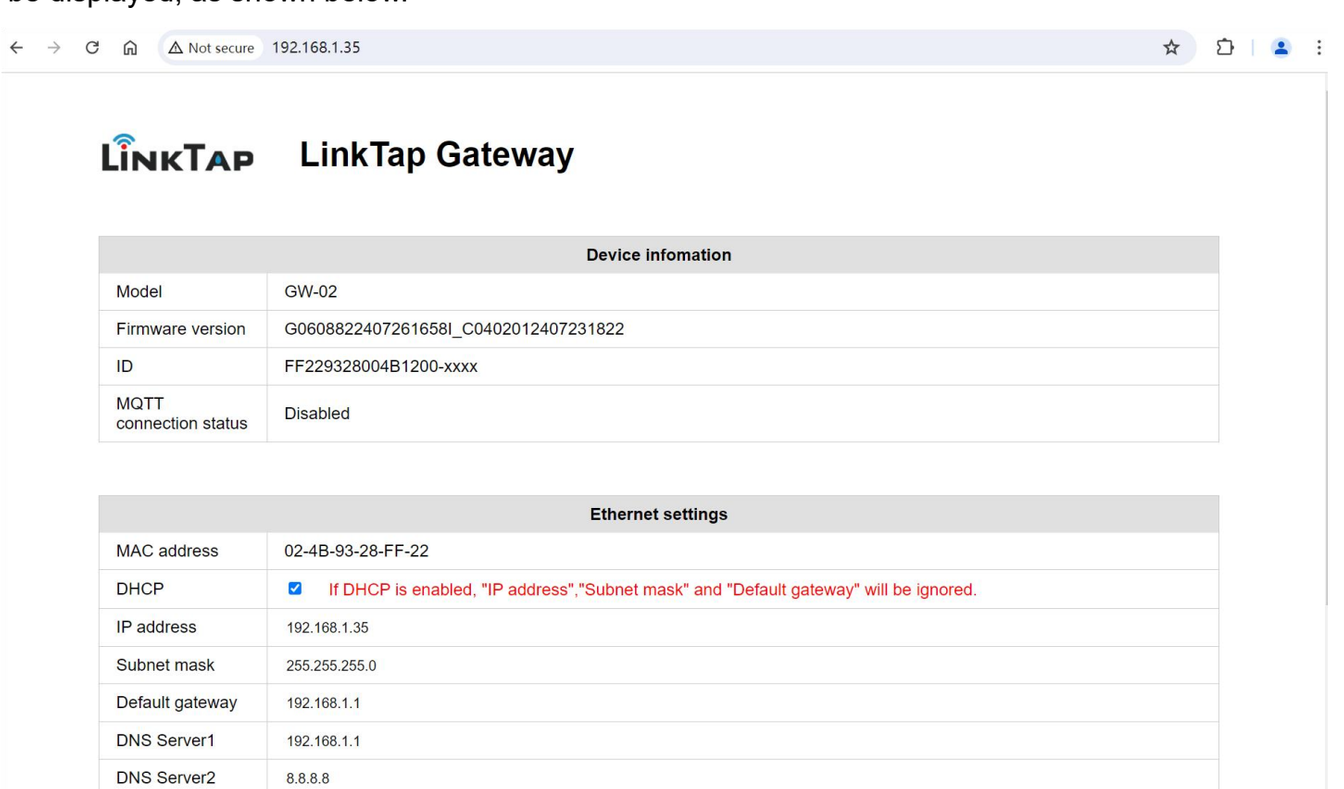
TABLE OF CONTENTS

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.



Open a browser, enter the gateway's IP address (e.g., 192.168.1.35), the gateway's admin page will be displayed, as shown below.



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

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	192.168.1.92
Broker port	1883
Client ID	FF229328004B1200
User name	test
Password	••••••
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_replay/topic

- 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. This means that users can only interact with the LinkTap devices through the gateway using MQTT/HTTP API commands locally.
  - ✧ Used as a MQTT Client (Beta).

  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 an MQTT Client" option is selected, both a third-party MQTT client and the LinkTap App can send commands to the gateway. However, as this feature is still in its beta stage, the consistency of watering plans and other settings between the MQTT client and the LinkTap App has not yet been fully resolved.

  To avoid conflicts and unforeseen issues with water timer control, please refrain from using a third-party MQTT client and the LinkTap App simultaneously to start/stop watering or change watering modes.
- Host type: The MQTT server type. The Host type determines the specific definition of the MQTT topic and message (payload). When you select "Regular MQTT Broker", the messages between the gateway and the MQTT Broker follow LinkTap's design, as explained in "3. Integrate with Regular MQTT Broker". If you choose "Home Assistant", the messages are set up according to the Home Assistant MQTT discovery format (https://www.home-assistant.io/integrations/mqtt/#configuration-via-mqtt-discovery). You simply need to enter the MQTT Broker's IP address, port, username, and password on the gateway, and LinkTap devices will be automatically detected in Home Assistant. For more info, please refer to "2. Integrate with Home Assistant".
- 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 as the gateway's ID for easy identification in MQTT Broker.
- User name: The username required to access the MQTT broker. If you're using Mosquitto as the MQTT Broker on Home Assistant, then by default, the username is your Home Assistant account username.
- Password: The password required to access the MQTT broker. If you're using Mosquitto as the MQTT Broker on Home Assistant, then by default, the password is your Home Assistant account password.
- Keep alive interval: The time (in seconds) that the MQTT client uses to maintain its connection with the broker by sending messages. The gateway will instantly send the device status to the

MQTT Broker when the device's status changes. If the status stays the same, the gateway will periodically send the device status to the MQTT Broker based on the "keep alive interval."

- Uplink: It is a required field for the "Regular MQTT Broker". There are two options available for configuring how to publish device status.
  - ✧ (Default) All devices' status are grouped and published to one topic. The LinkTap Gateway combines the status of multiple devices into one message that is less than the MTU (Max transmission unit, the value is 1500 in the gateway) in length.
  - ✧ (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 the "regular MQTT broker", as well as for providing MQTT topics to "Home Assistant" to configure "X_total_duration", "X_volume_limit", and "X_failsafe_duration" (refer to sections 2.5/2.6/2.7 for details).
- Download reply: Used for "Regular MQTT Broker".

1.3. Local HTTP API settings

Local HTTP API settings	
Function	☐ Enable HTTP Client
Server URL	http://Domain_Name_or_IP:Port/Path
Response	☑ Wrap the gateway's response in HTML

Refer to section "4. Integrate with HTTP API" for the functions corresponding to the following settings.

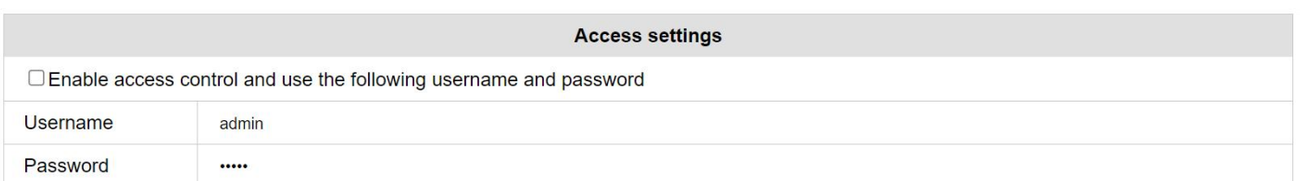
- Function: You can enable or disable the gateway's HTTP client. When enabled, the gateway will send device status messages to the specified "Server URL" when the device status changes, or the device status remains unchanged for more than 2 minutes.
- Server URL: The HTTP server that receives messages from the gateways HTTP client. Currently, only HTTP plaintext transmission is supported, while HTTPS is not available yet.
- Response: Should the gateway's HTTP API response be wrapped in HTML tags?

1.4. Misc Settings

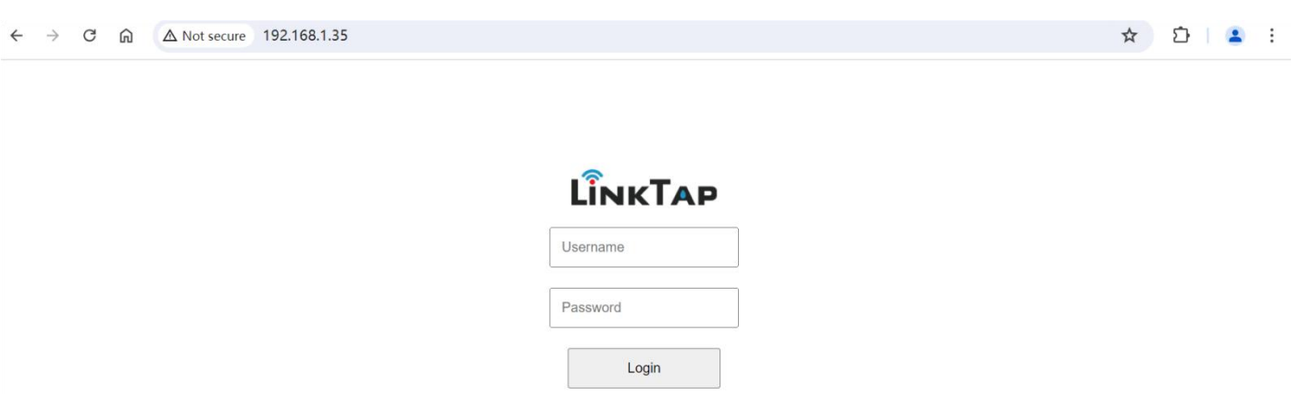
Misc settings	
Volume unit	◉ Litre
	○ Gallon (1 Gallon = 3.785 Litres)
Decimal separator (in Home Assistant UI)	◉ Point (e.g., 2.34)
	○ Comma (e.g., 2,34)
Persistent volume_limit	☑ Enable
Persistent total_duration	☑ Enable
Failsafe duration	86340 seconds
Enable mDNS responder	☐ Enable

- 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.
- Persistent volume_limit: Decide whether the "total duration" configured in Home Assistant should be retained permanently or used as a one-time setting. For details, refer to section "2.5 Set Watering Duration (value of sensor.X_total_duration entity)".
- Persistent total_duration: Decide whether the "volume_limit" configured in Home Assistant should be retained permanently or used as a one-time setting. For related functions, refer to section "2.7 Set Watering Volume (value of sensor.X_volume_limit entity)".
- Enable mDNS responder: For gateway model GW-02 only, the gateway will respond to the mDNS query if it is enabled.

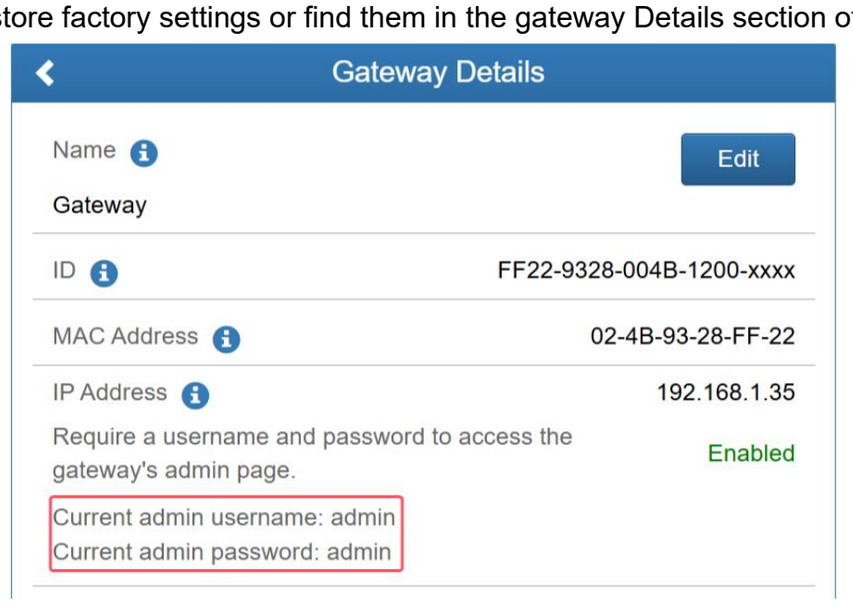
1.5. Access settings



You can enable access control on the gateway's management page, requiring a login before making any configuration changes to prevent unauthorized access.



If you forget your username or password, you can either press the gateway's reset button for 3 seconds to restore factory settings or find them in the gateway Details section of the LinkTap App.



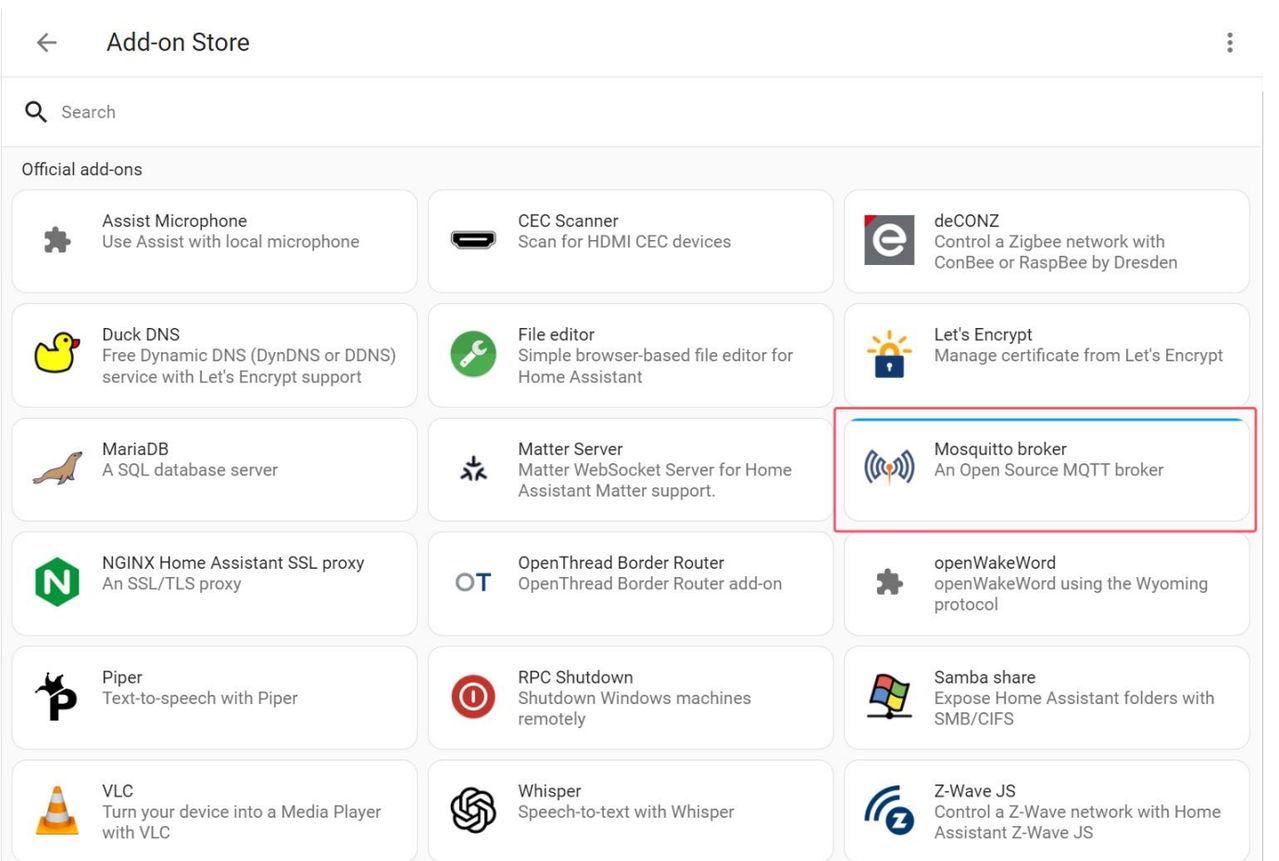
2. Integrate with Home Assistant

2.1. Install Home Assistant

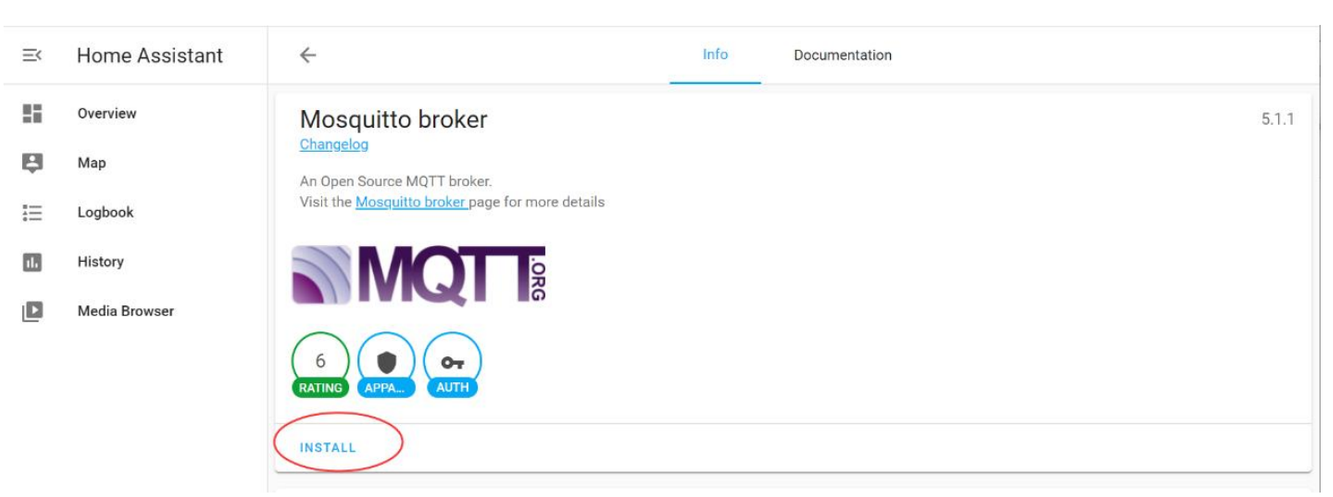
Please refer to https://www.home-assistant.io/installation for the Home Assistant installation.

2.2. Install MQTT Broker

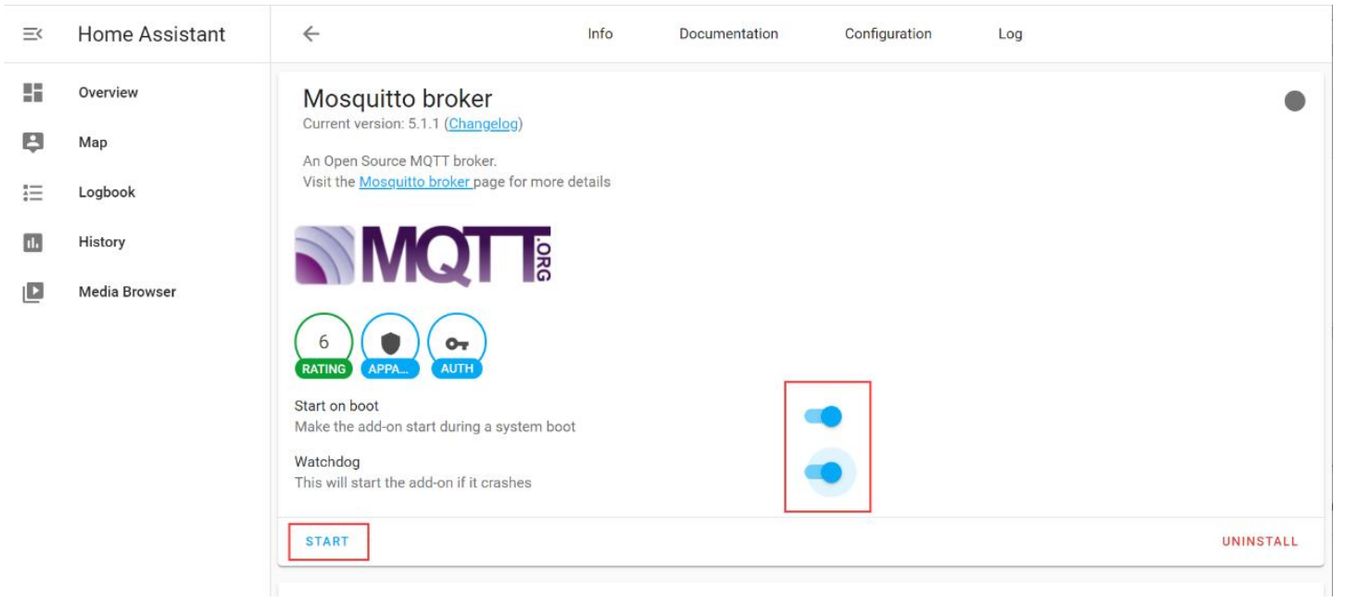
2.2.1. After installing Home Assistant, log in to Home Assistant in your browser, click "Settings" on the left sidebar, then navigate to "Add-ons"->"ADD-ON STORE", find "Mosquitto broker" and click on it.



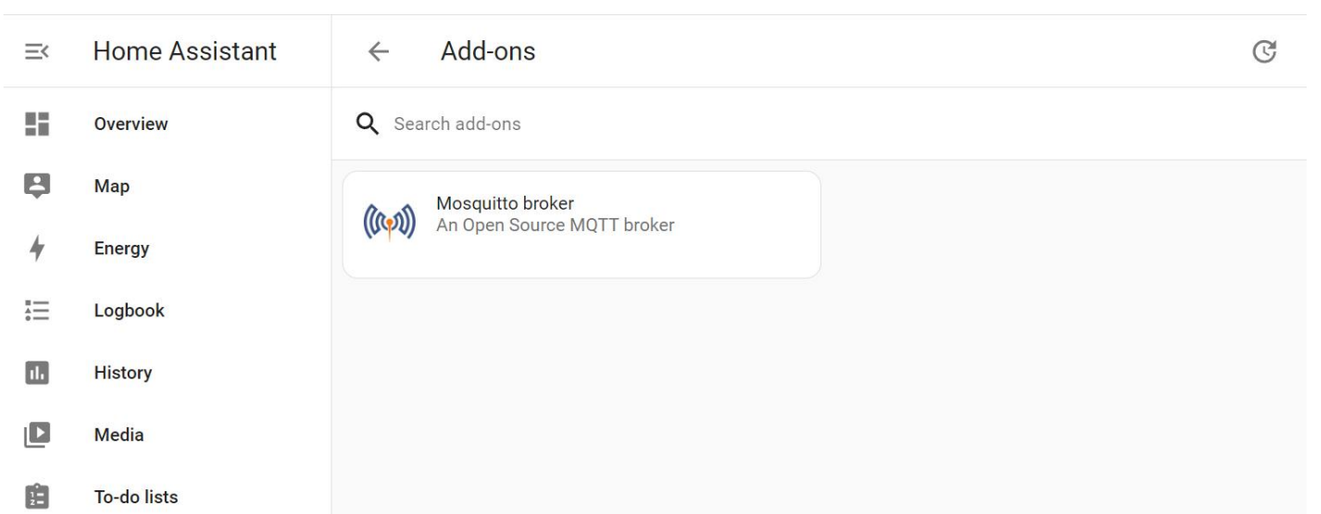
2.2.2. Click "INSTALL" to start the installation.



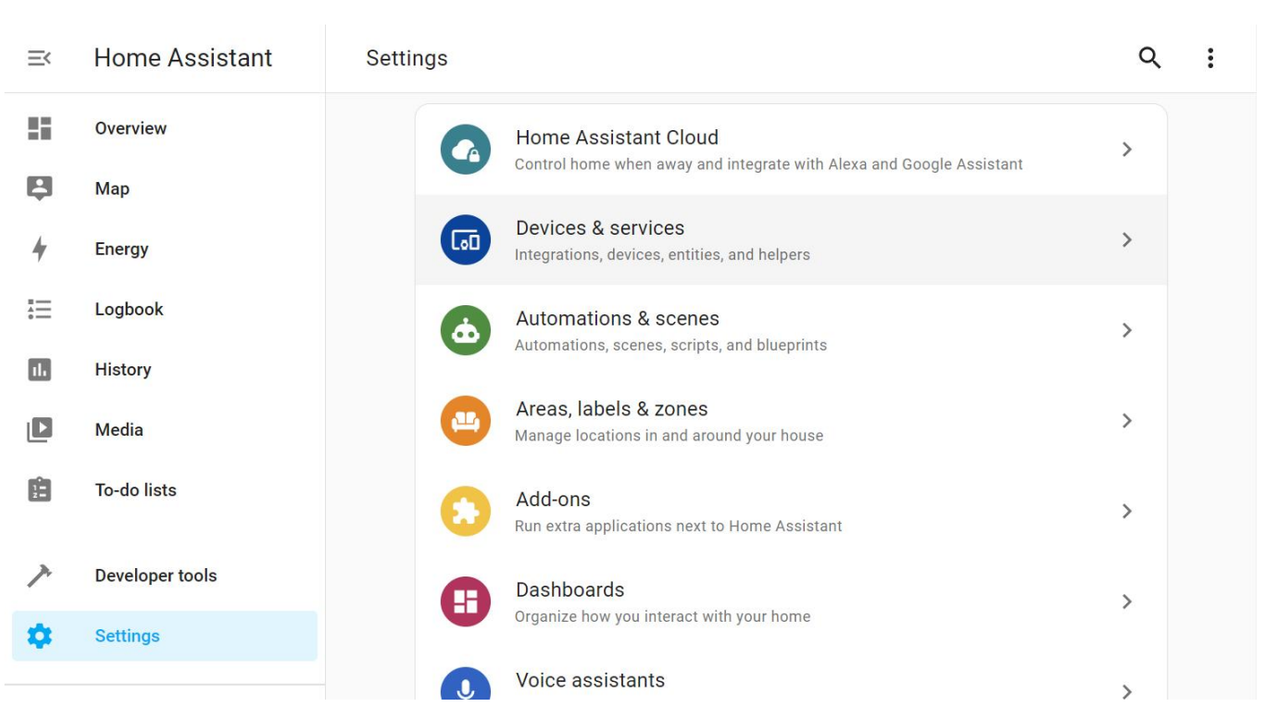
2.2.3. When the installation is completed, make sure "Start on boot" and "Watchdog" are enabled, then click "START" to start the MQTT broker.



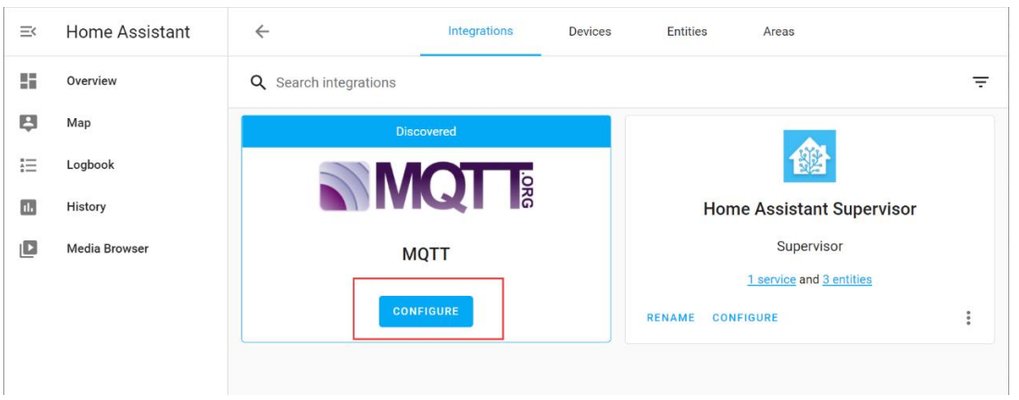
2.2.4. You should now be able to see the newly installed "Mosquitto broker" on the "Add-ons" interface.

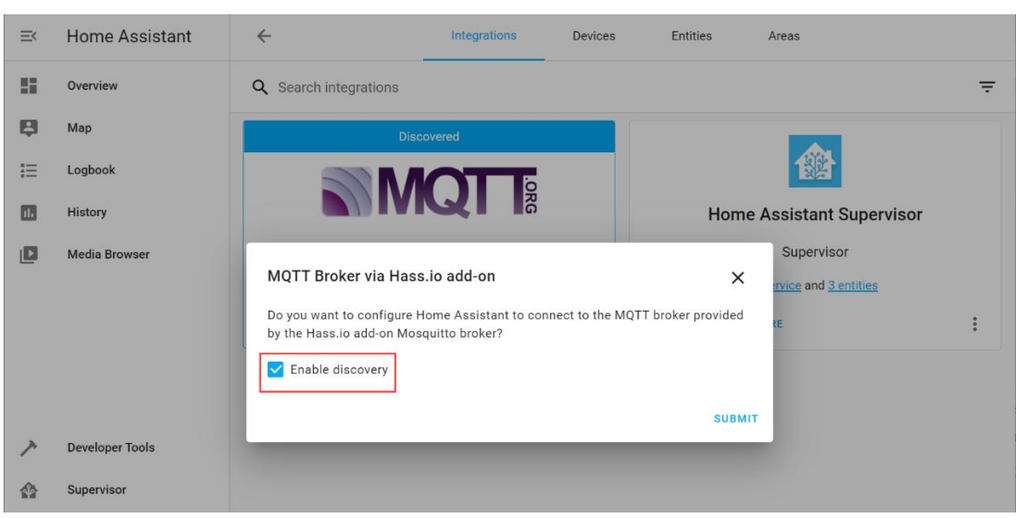


2.2.5. Click "Settings" on the left sidebar, then choose "Devices & services".



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".)



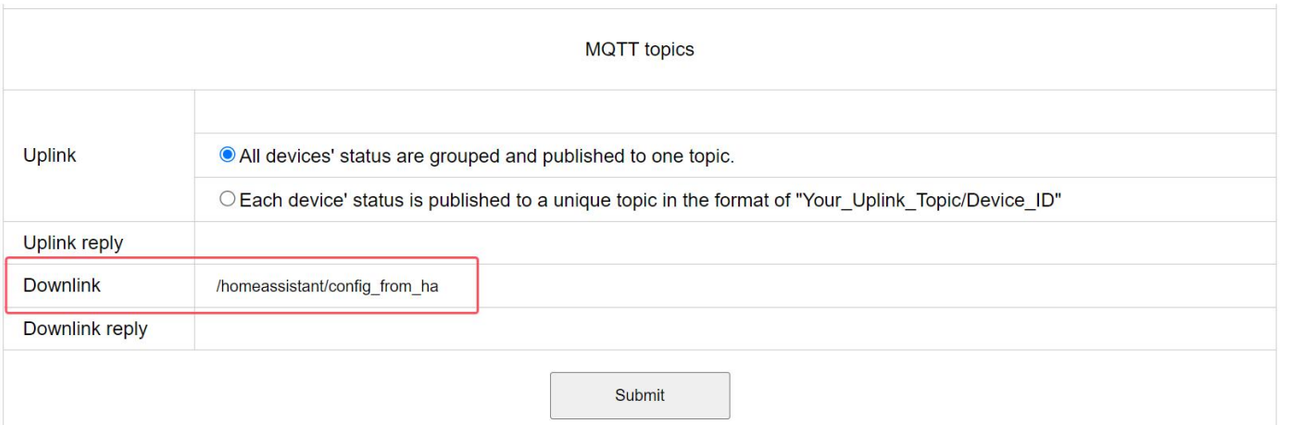


2.3. MQTT Client Settings for LinkTap Gateway

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".

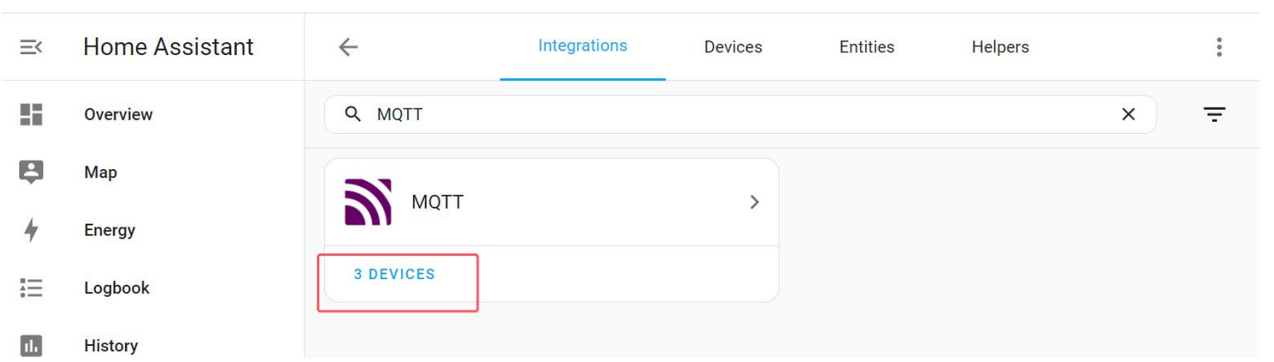
MQTT client settings	
Function	○ Disable MQTT Client
	○ Used as a MQTT Client only
	◉ Used as a MQTT Client (Beta)
Host type	○ Regular MQTT Broker
	◉ Home Assistant
Home Assistant prefix	homeassistant
Broker address	192.168.1.92
Broker port	1883
Client ID	FF229328004B1200
User name	test
Password	••••••
Keep alive interval	120
Submit	

In order to send configuration parameters from Home Assistant to the gateway, you can set the "downlink" topic in the MQTT topic here.

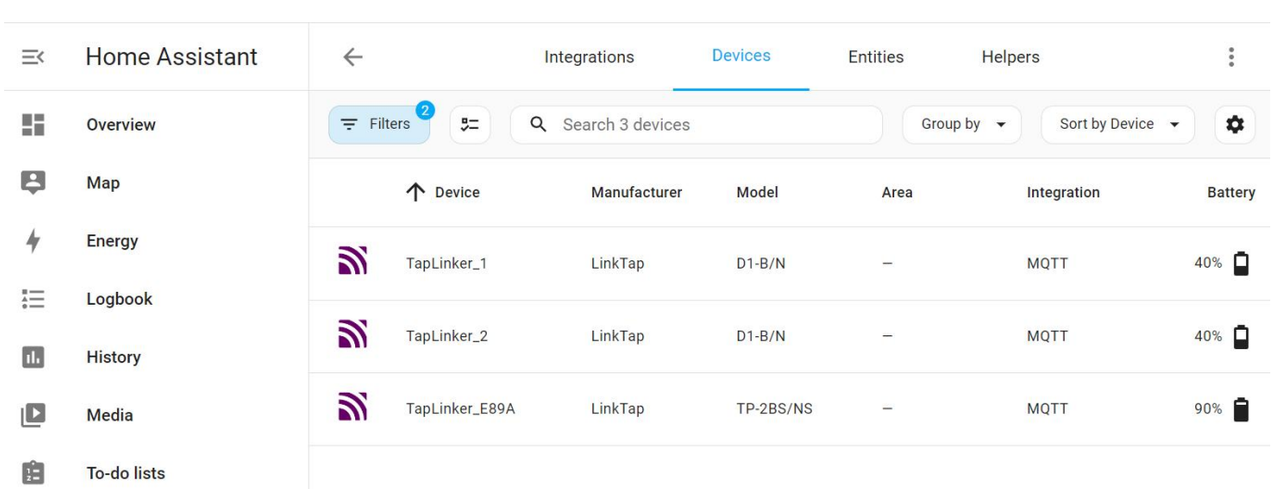


Lastly, you need to click "Restart Gateway" for the changes to take effect.

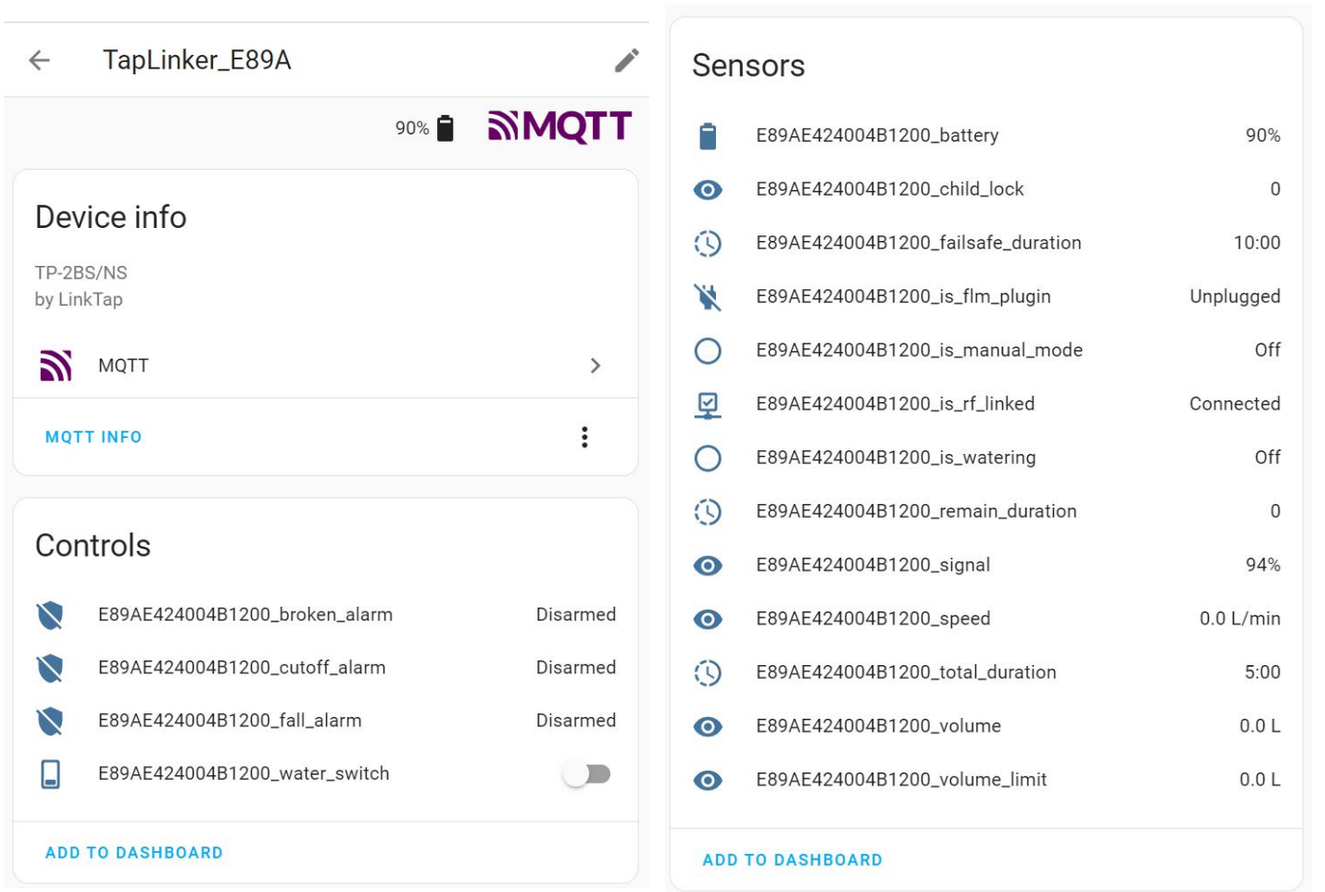
2.3.2. After the gateway is rebooted, you should be able to see "devices" are added into the "Mosquitto broker".(Note: The number of devices that are discovered depends on how many valves your gateway is currently associated with.)



2.3.3. Click "X DEVICES" to view the devices discovered by Home Assistant.



Next, click on a LinkTap device, and you'll see the entities associated with that device.



2.3.4. Entities of a LinkTap water timer

The entities of the a LinkTap water timer are named in the format of "water timer ID_ entity name". The "X" in the table represents the ID of the water timer.

Home Assistant Component	Entity name	Note
switch	X_water_switch	Watering switch. **NOTE: The switch operates in "optimistic mode," meaning it updates its status immediately upon receiving a command, rather than waiting for confirmation from the device. (Refer to https://www.home-assistant.io/integrations/switch.mqtt/ for more details.) As a result, do not rely on the state of "X_water_switch" in Automations or Scripts to determine the actual watering status of the water timer. Instead, use the state of the "X_is_watering" entity for accuracy.**
binary_sensor	X_is_watering	Is water timer currently watering? **NOTE: Use the state of this entity in Automations or Scripts to detect the actual watering status of the water timer.**
	X_is_manual_mode	Is watering by press "Manual On/Off" button (on Taplinker G1S/G2S/D1, Valvelinker V1-1Z/V1-2Z/V1-4Z)?

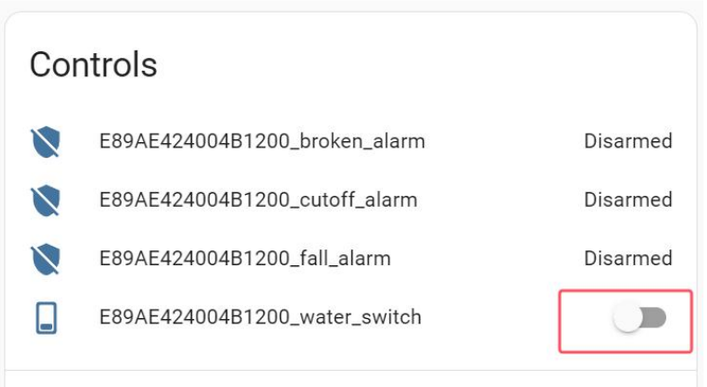
	X_is_rf_linked	Water timer's connection status. "on" means the water timer is connected to the gateway.
	X_is_flm_plugin	Flow meter's connection status. "on" means the flow meter is connected.
	X_child_lock	Is child-lock of "Manual On/Off" button (on Taplinker G1S/G2S/D1, Valvelinker V1-1Z/V1-2Z/V1-4Z) enabled?
sensor	X_total_duration	Total watering duration in seconds. **To prevent unexpected issues caused by communication interruptions between the water timer, the gateway, and Home Assistant during watering, please ensure that a reasonable value is set for this entity. For more details, refer to "2.5.Set watering duration (value of sensor.X_total_duration entity)" and "2.8.An example of the combined use of X_total_duration, X_failsafe_duration, and X_volume_limit".**
	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. **To prevent unexpected issues caused by communication interruptions between the water timer, the gateway, and Home Assistant during watering, please ensure that a reasonable value is set for this entity. For more details, refer to "2.6.Set failsafe watering duration(value of sensor.X_failsafe_duration entity)" and "2.8.An example of the combined use of X_total_duration, X_failsafe_duration, and X_volume_limit".**
	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.

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. Test the valve switch function through the Switch entity

Now, you can find the "X_water_switch" entity from the associated devices and click on it to control the water timer's valve to open or close.

It's important to note that there is a delay in the wireless interaction between the water timer and the LinkTap Gateway, which affects the switch state of the Switch entity. Since the Switch entity operates in optimistic mode, its status in Home Assistant may not accurately reflect the water timer's actual state. The "X_is_watering" entity shows the true switch state of the water timer. When using Scripts or Automations to control the water timer, refer to "X_is_watering" instead of "X_water_switch" to determine its actual state.

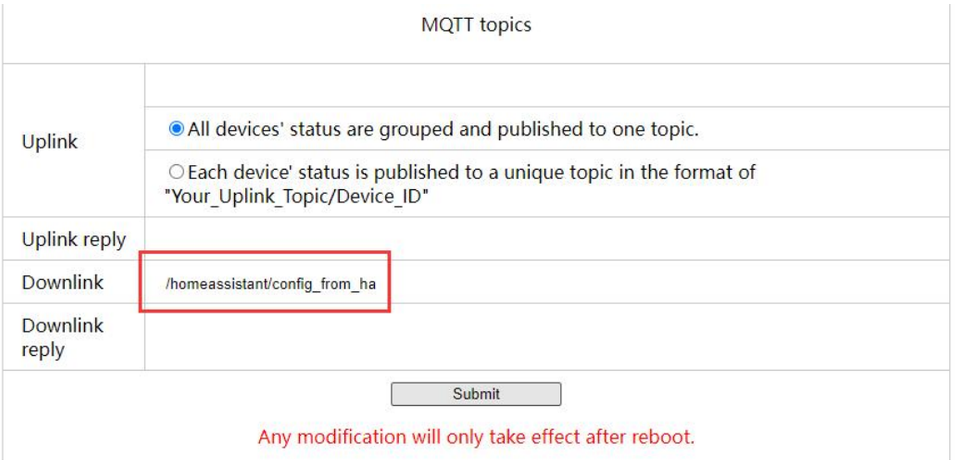


Additionally, before using Scripts or Automations to control the water timer, it's essential to read and understand the content in sections 2.5 and 2.6 below.

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.



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.

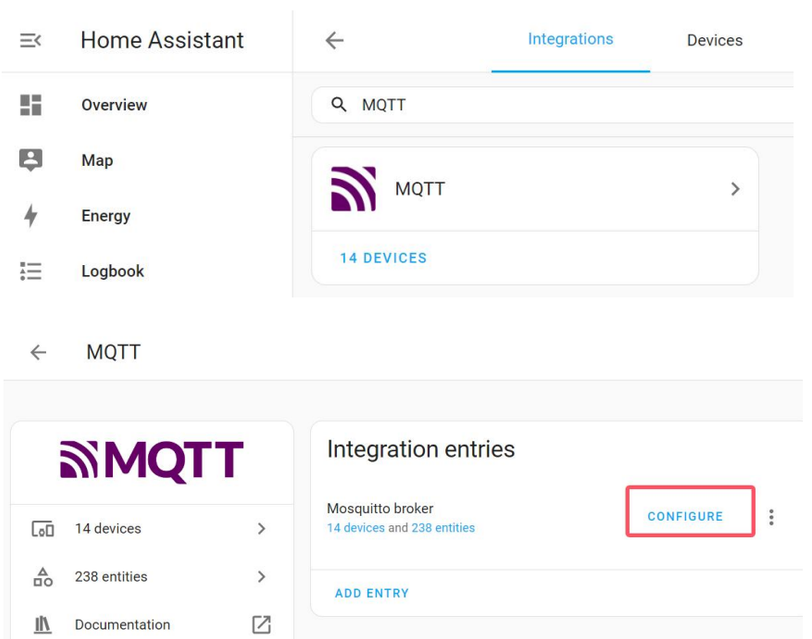
The format of the MQTT payload that sets "total_duration" is:

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

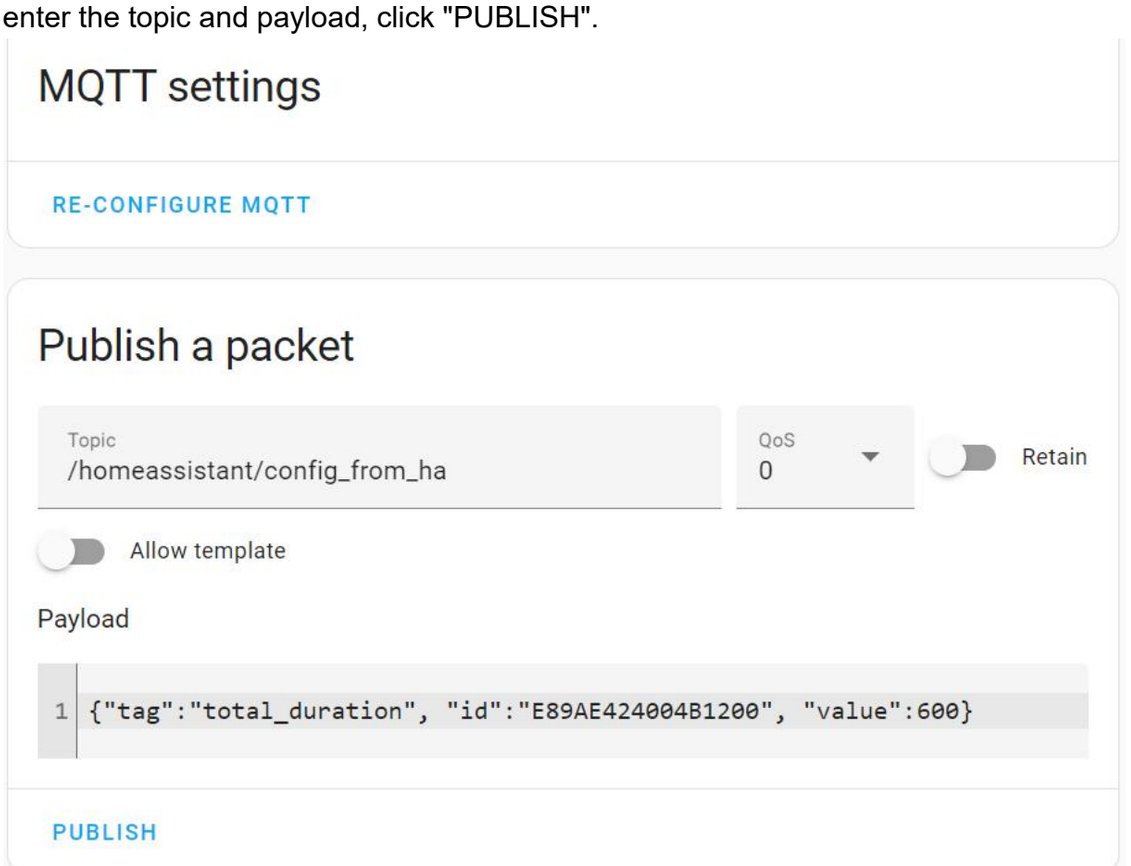
where "tag" is the name of the configuration, "id" is the ID of the Taplinker, "value" is the value to be set to the configuration, in seconds here, with a minimum of 3 and a maximum of 86340.

Below is an example that the TapLinker (ID: E89AE424004B1200) is configured via the Mosquitto broker from the HA.

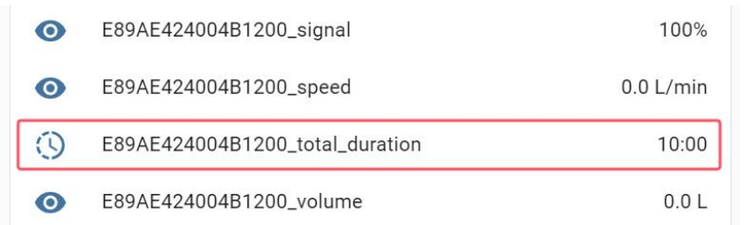
First, Navigate to Settings > Integrations > MQTT > CONFIGURE.



Then, enter the topic and payload, click "PUBLISH".

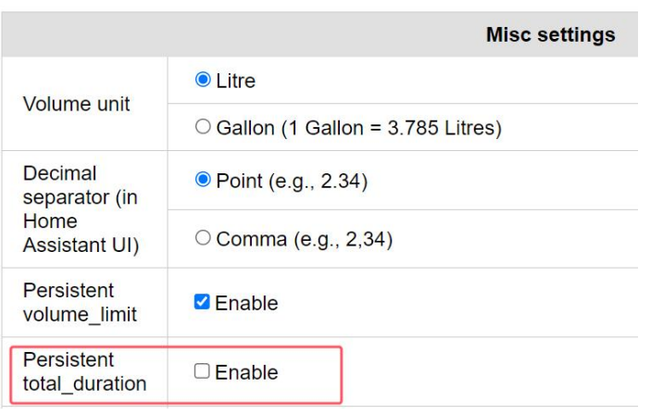


Once the configuration message is published successfully, the value of "YOUR_TAPLINKER_ID_total_duration" will be updated accordingly.



Please note that by default, once the "total duration" is set, it will be retained. This means that any watering cycles you start using the "X_water_switch" will use this value until you change it again.

If you want the "total_duration" to reset to the default value after each watering, you can disable "Persistent total_duration" on the gateway management page.



If the duration of your watering sessions might vary, you can use a script to first set the "total_duration" and then turn on the "water_switch", for example:

```
alias: set_total_duration_then_start_watering
sequence:
  - service: mqtt.publish
    data:
      topic: /homeassistant/config_from_ha
      payload: "{\"tag\":\"total_duration\",\"id\":\"E89AE424004B1200\",\"value\":123}"
  - service: switch.turn_on
    data: {}
    target:
      entity_id: switch. e89ae424004b1200_water_switch
mode: single
```

2.6. Set failsafe watering duration(value of sensor.X_failsafe_duration entity)

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

The "failsafe_duration" and "total_duration" both control how long a watering cycle lasts. If both are set (meaning neither is 0), the gateway will use the shorter of the two for the current cycle. However, there are some differences between them:

	total_duration	*failsafe_duration*
Lifespan	*By default, the "total duration" is saved permanently once set. However, it can be configured to be non-persistent, resetting to the default value (86340) after each watering.*	*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.*

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

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. Watering will not stop until the "remain_duration" on the TapLinker counts down to zero or the TapLinker receives a "water_switch (OFF)" command from HA. If the TapLinker is actively watering and the connection between Home Assistant, the gateway, and the TapLinker is lost, the "water_switch (OFF)" command will not reach the TapLinker. This can result in overwatering and potential losses.

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.

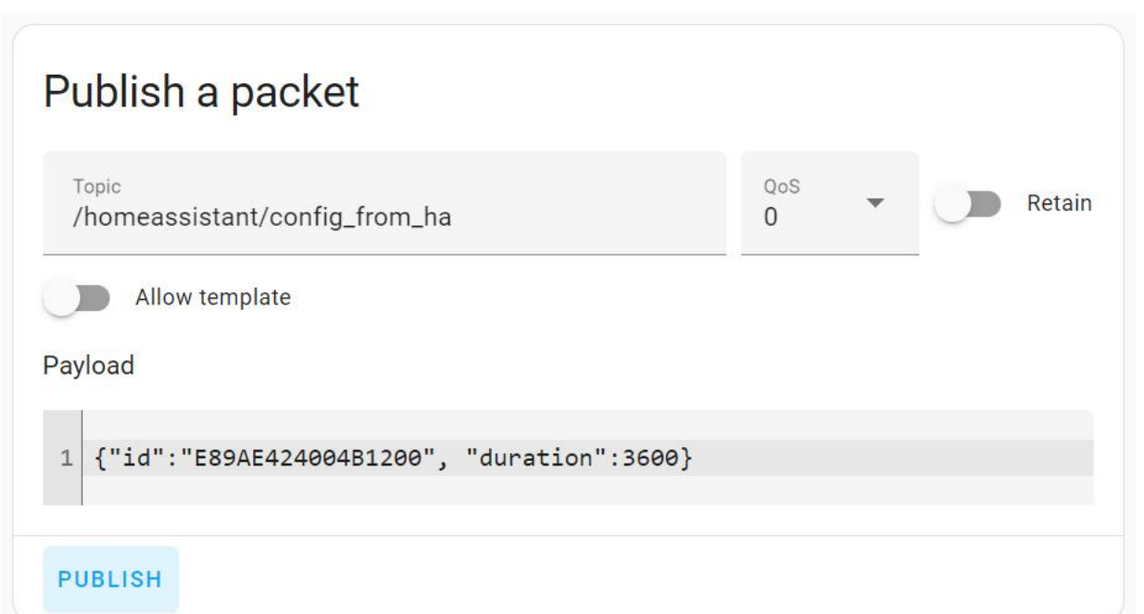
You can set the "failsafe_duration" by publishing content to the topic from the HA (or other MQTT Clients subscribed to the topic). The content format for this topic is:

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

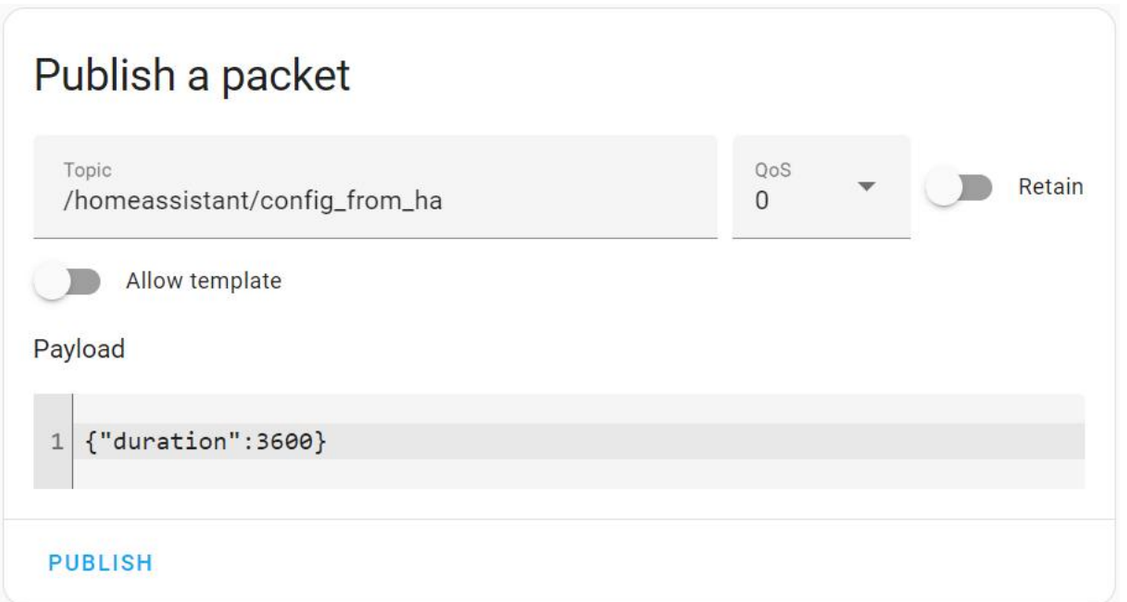
where "id" is the ID of your TapLinker, and "duration" is the value of "failsafe_duration" in seconds. If "id" is omitted, then the "duration" will be applied for all TapLinkers, and all TapLinkers' "failsafe_duration" previously set will be overridden.

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: E89AE424004B1200) 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., 3600) to the entity named "E89AE424004B1200_failsafe_duration" on the HA. Now if watering is started through "E89AE424004B1200_water_switch", the TapLinker will water for 3600 seconds.



If you want to set a "failsafe_duration" for all water timers under the same gateway, just remove the "id" field in the message, as shown in the image below:



In addition, on the gateway management page under "Misc settings", there's a configuration option that makes it easier to set the "failsafe_duration" for all water timers under the gateway, as shown below:

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)
Persistent volume_limit	☑ Enable
Persistent total_duration	☑ Enable
Failsafe duration	3600 seconds

2.7. 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.

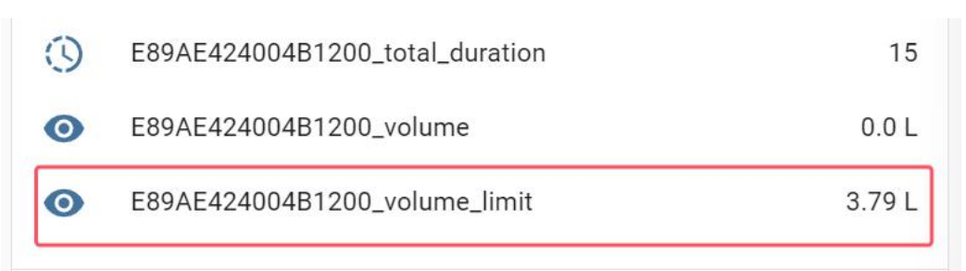
The format of the MQTT payload that sets "volume_limit" is:

`{"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":"E89AE424004B1200", "value":3.79}

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



Please note that by default, once the "volume_limit" is set, it will be retained. This means that any watering cycles you start using the "X_water_switch" will use this value until you change it again.

If you want the "volume_limit" to reset to the default value (0, which means no watering capacity limit) after each watering, you can disable "Persistent volume_limit" on the gateway management page.

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)
Persistent volume_limit	☐ Enable
Persistent total_duration	☑ Enable

If you want to change the water capacity of each your watering sessions, you can use a script to first set the "volume_limit" and then turn on the "water_switch", for example:

```
alias: set_volume_limit_then_start_watering
sequence:
  - service: mqtt.publish
    data:
      topic: /homeassistant/config_from_ha
      payload: "{\"tag\":\"volume_limit\",\"id\":\"E89AE424004B1200\",\"value\":3.79}"
  - service: switch.turn_on
    data: {}
    target:
      entity_id: switch. e89ae424004b1200_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.

2.8. An example of the combined use of X_total_duration, X_failsafe_duration, and X_volume_limit

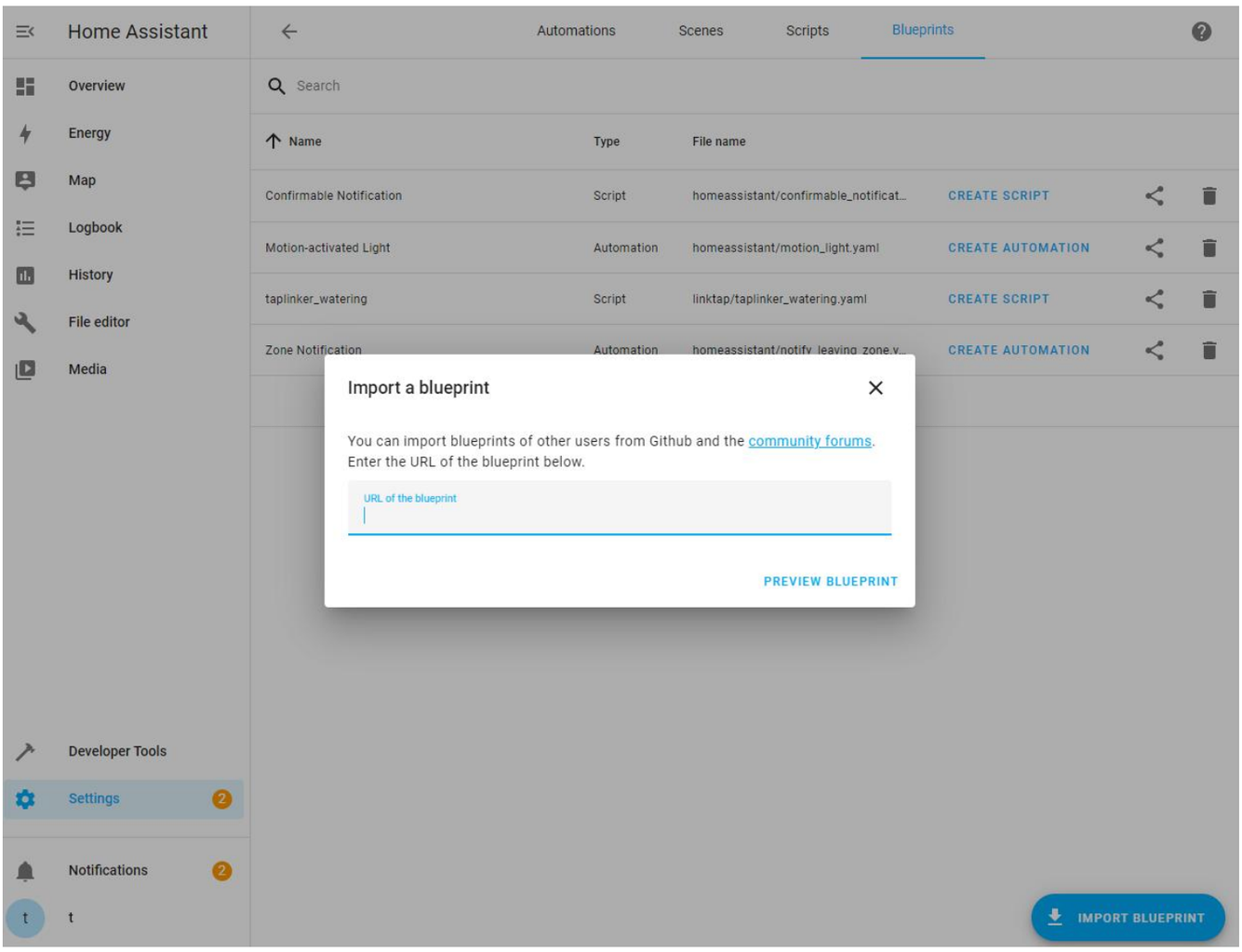
Let's say you have three water timers A/B/C under the same gateway. Timer A is for watering the underground lawn sprinkler, Timer B is for drip irrigation on potted plants by the porch, and Timer C is for refilling the swimming pool. Based on your experience, you usually water the lawn for no more than 30 minutes, water the potted plants for 15 minutes and don't want to exceed 20 liters, and when refilling the pool, you don't want to exceed 500 liters (assuming it's not possible to take more than 30 minutes to fill 500 liters).

So, you can:
1) Send {"duration":1800} to the "Downlink" topic, or set the Failsafe duration to 1800 in the gateway management page to limit the maximum watering duration of all water timers to 1800 seconds (30 minutes);
2) Send {"tag":"total_duration","id":"ID_of_water_timer_B","value":900} and {"tag":"volume_limit","id":"ID_of_water_timer_B","value":20} to the "Downlink" topic separately, to restrict Timer B's watering time and water capacity;
3) Send {"tag":"volume_limit","id":"ID_of_water_timer_C","value":500} to the "Downlink" topic, limiting Timer C's maximum watering capacity to 500 liters.

That way, when you separately turn on "X_water_switch" from Home Assistant:
1) Water timer A can water for up to 30 minutes and will automatically stop when the time is up, so you don't have to worry about whether it'll get the stop message from Home Assistant;
2) Water timer B can water for up to 15 minutes, and will also automatically stop when the time is up without needing the stop message from Home Assistant, but it might stop early if it reaches 20 liters before the time is up;
3) Since you have set a global "failsafe_duration" of 1800 seconds, water timer C can also water for up to 30 minutes, but it might stop early if it reaches 500 liters before the time is up.

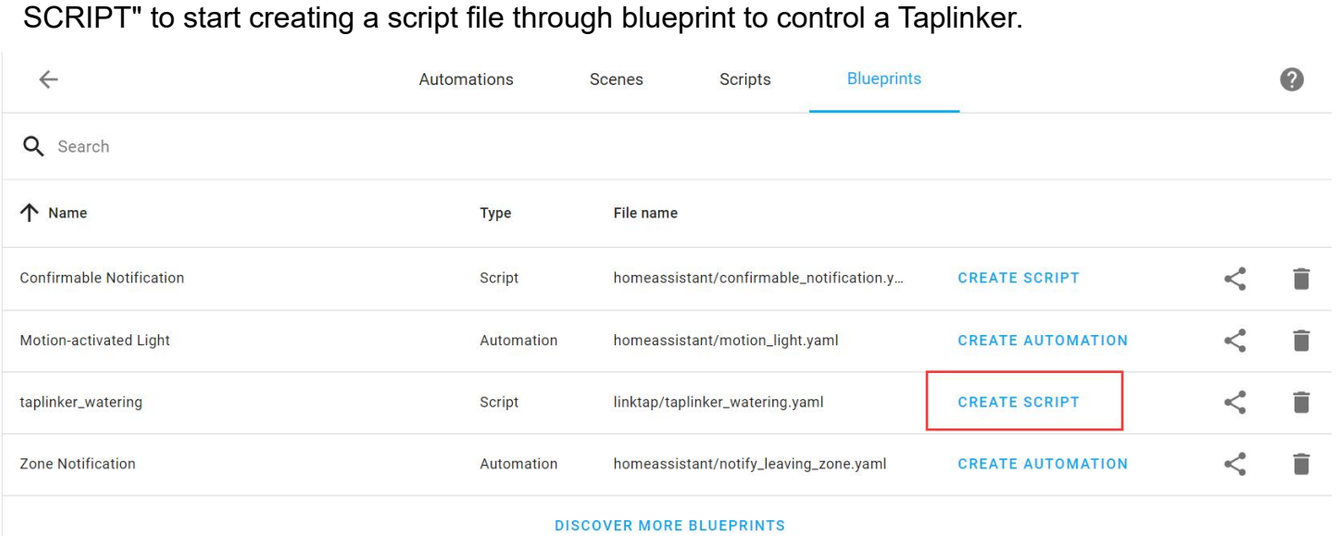
2.9. Create a script via Blueprint to control Taplinker's watering

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

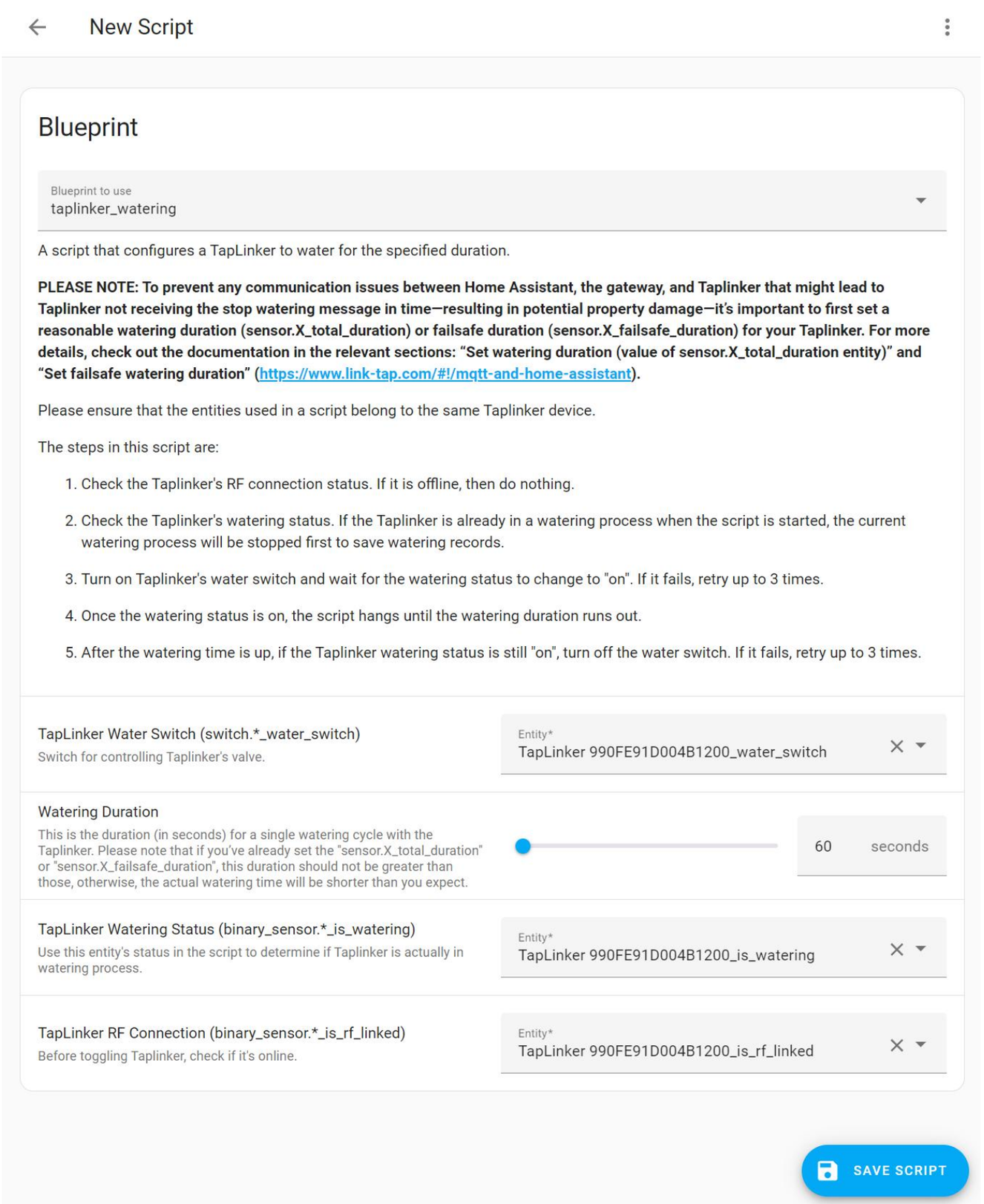


Click "PREVIEW BLUEPRINT" to view the description of the blueprint, then click "Import blueprint" to import the blueprint.

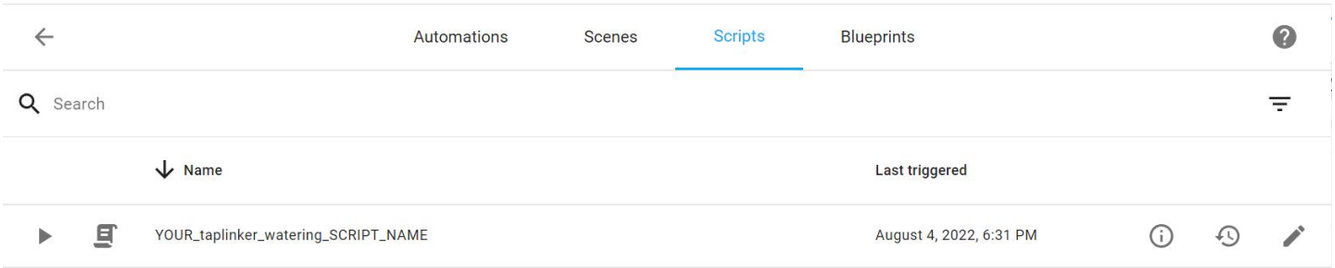
Now you'll see the "taplinker_watering" in the Blueprints list. Next, as shown below, click "CREATE SCRIPT" to start creating a script file through blueprint to control a Taplinker.



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.



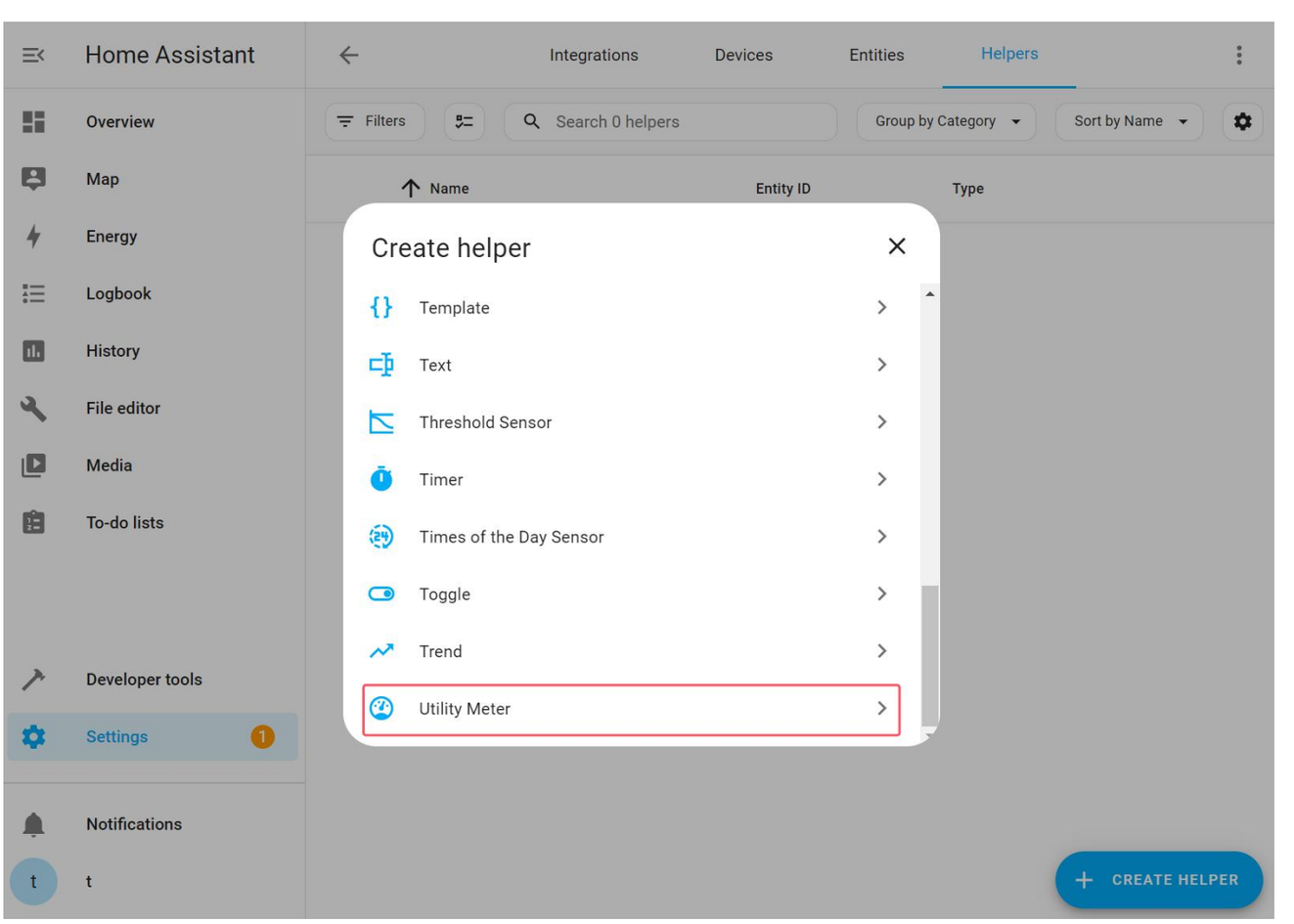
As shown below, the newly created script will be listed in the "Scripts" page. You can then click the triangle icon (on the left of the list) to execute the script and check if it behaves as expected.



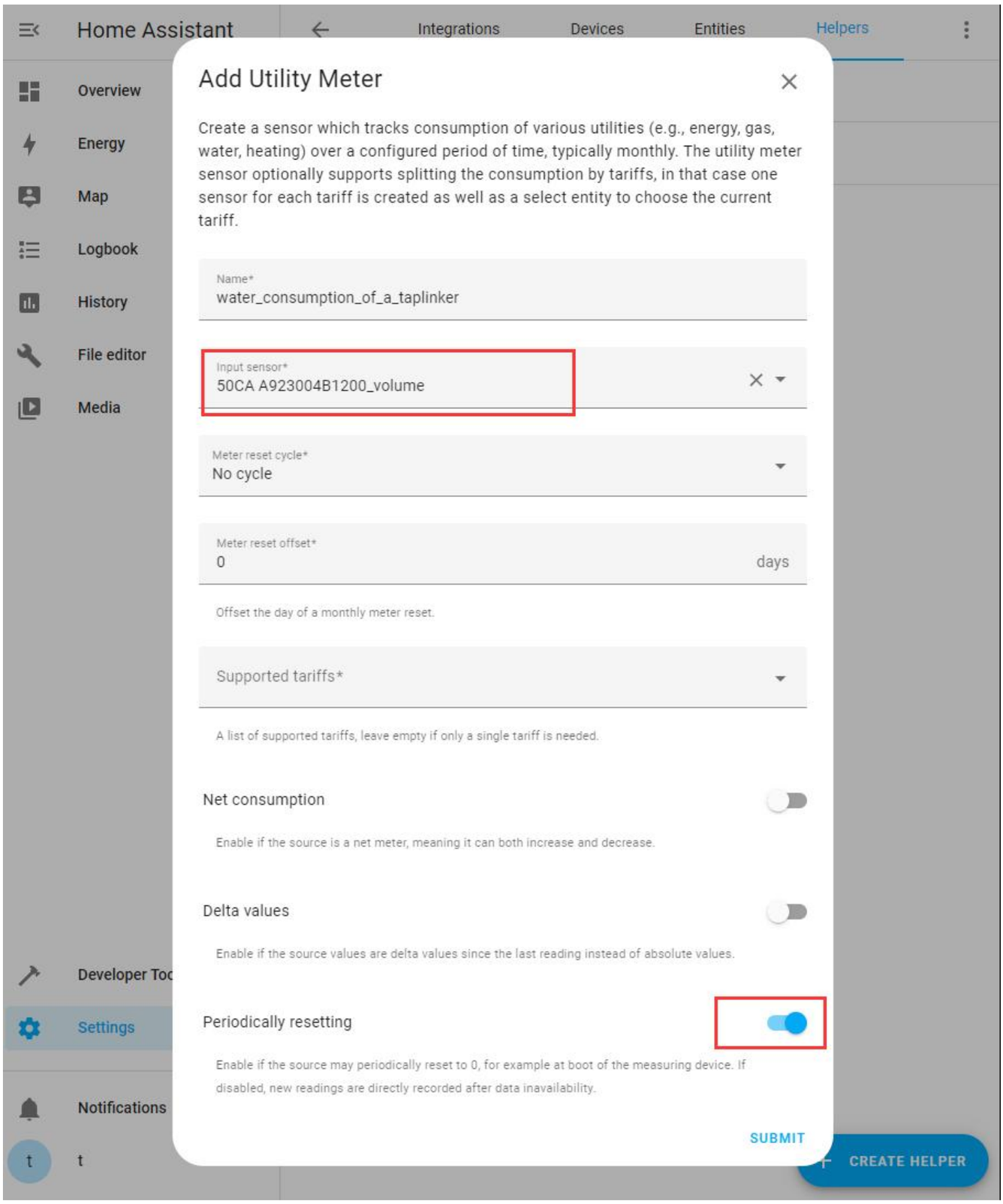
2.10. Track Taplinker water usage in the Energy dashboard

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.

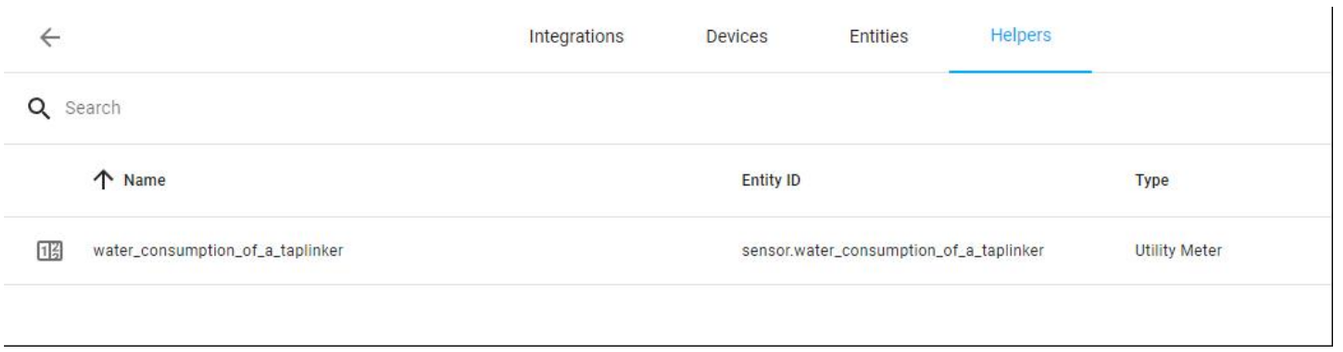
First, navigate to Settings > Device & services > Helpers, then click on "CREATE HELPER" in the bottom-right corner to create a Utility Meter helper entity.



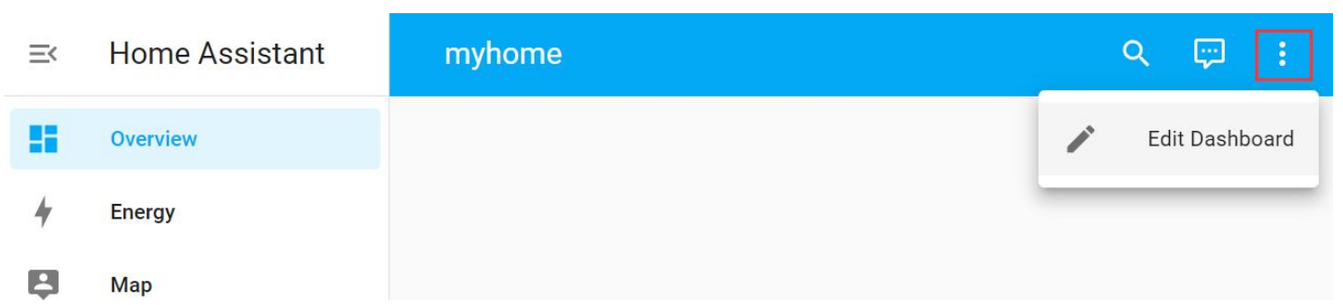
During the "Add Utility Meter" process, you can give the meter an appropriate name based on your needs. This name will be used as the entity name later on. Next, select the "X_volume" entity under a specific Taplinker as your data source. For "Meter reset cycle", choose "No cycle". An important note: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



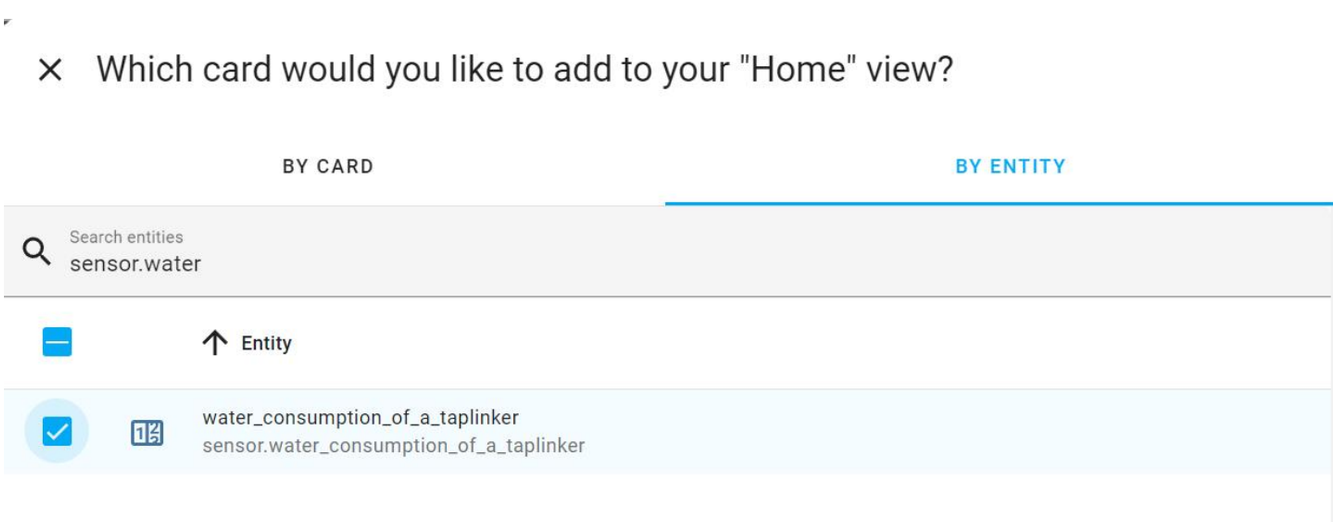
After clicking "SUBMIT" to submit, the creation of the Utility Meter is complete.

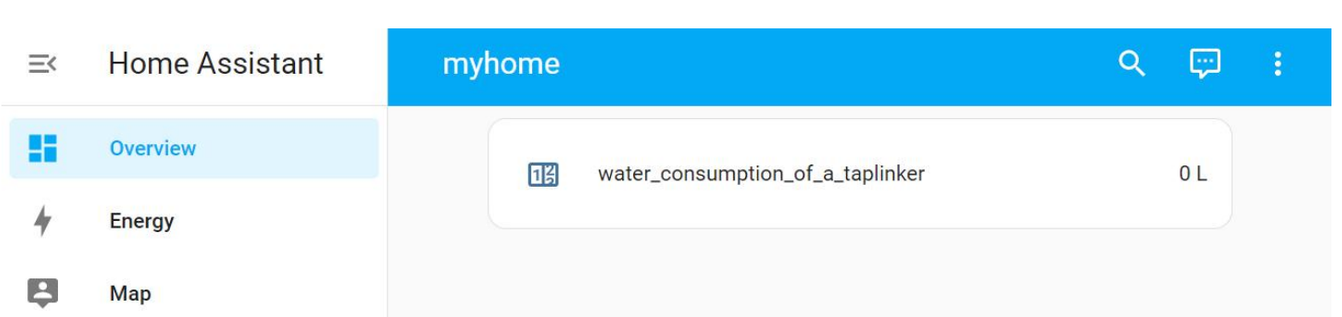


At this point, you can go to the "Overview" interface in Home Assistant, click on the three-dot icon in the upper right corner, then select "Edit Dashboard", and finally, click "ADD CARD" to add a display card.



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

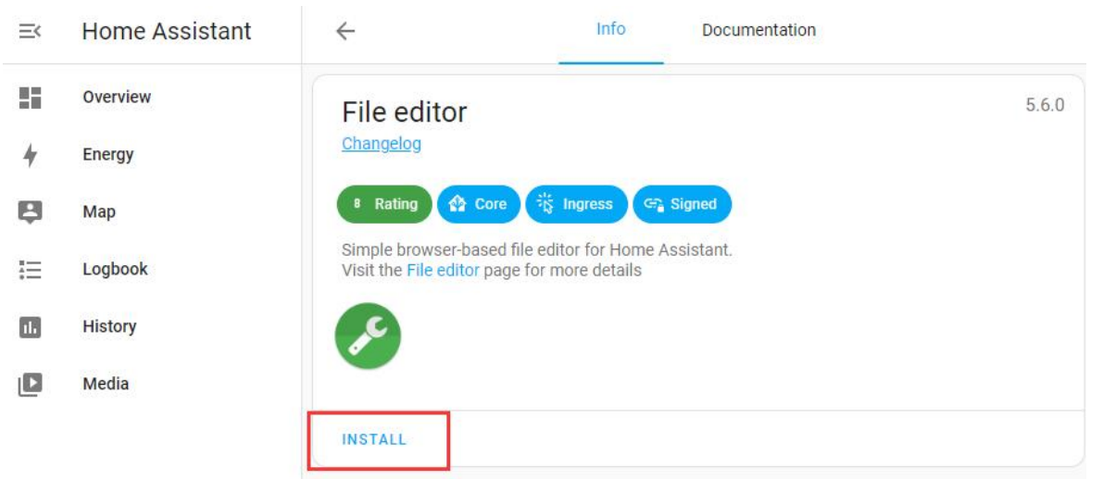


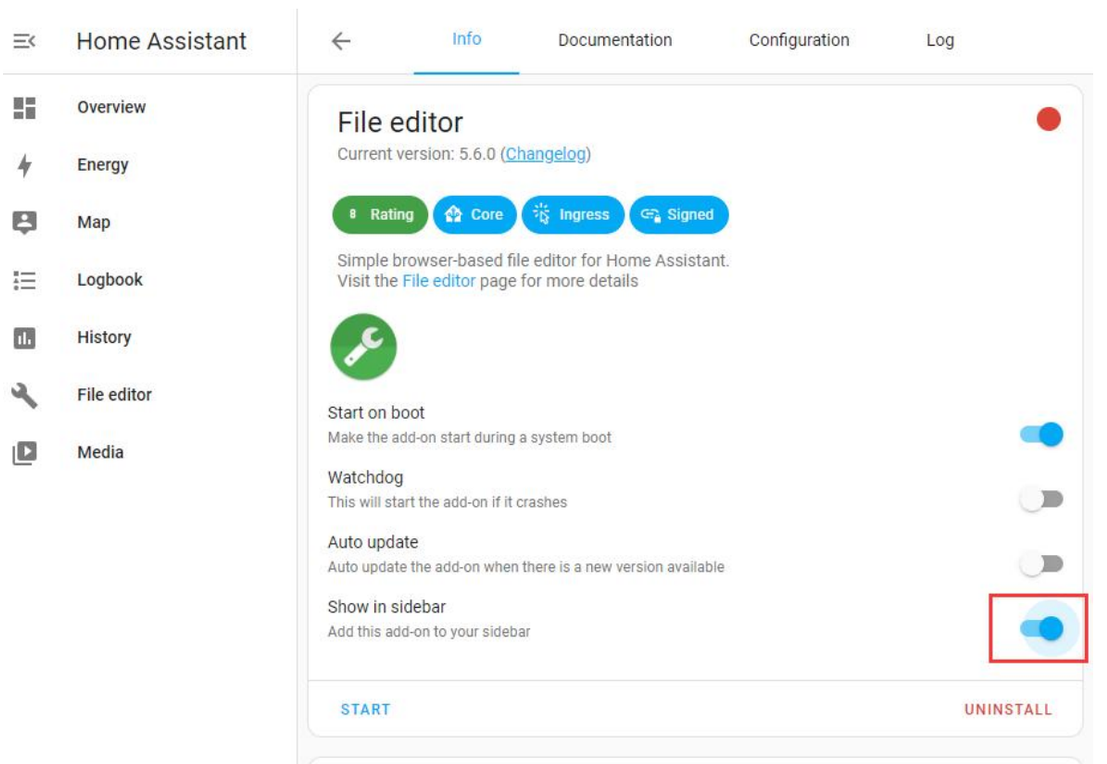


After following the above steps, you have successfully achieved continuous accumulation of watering volume for a specific Taplinker. If you wish to further track watering statistics on a daily, weekly, monthly, and yearly basis, you can use Home Assistant's Energy dashboard.

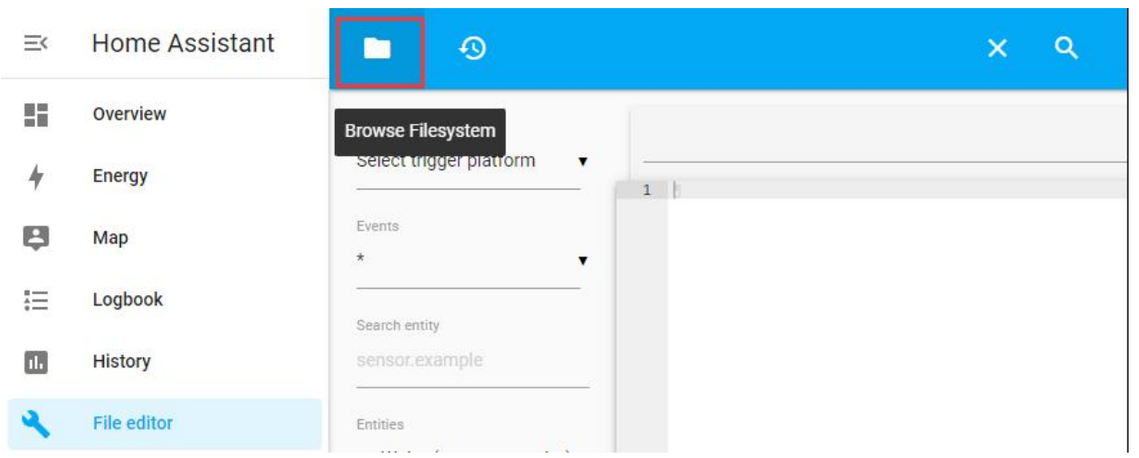
Since the Energy dashboard tracks water consumption through a sensor with a "device_class" type of "water", you will need to manually create such a sensor entity in Home Assistant's "configuration.yaml".

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





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.

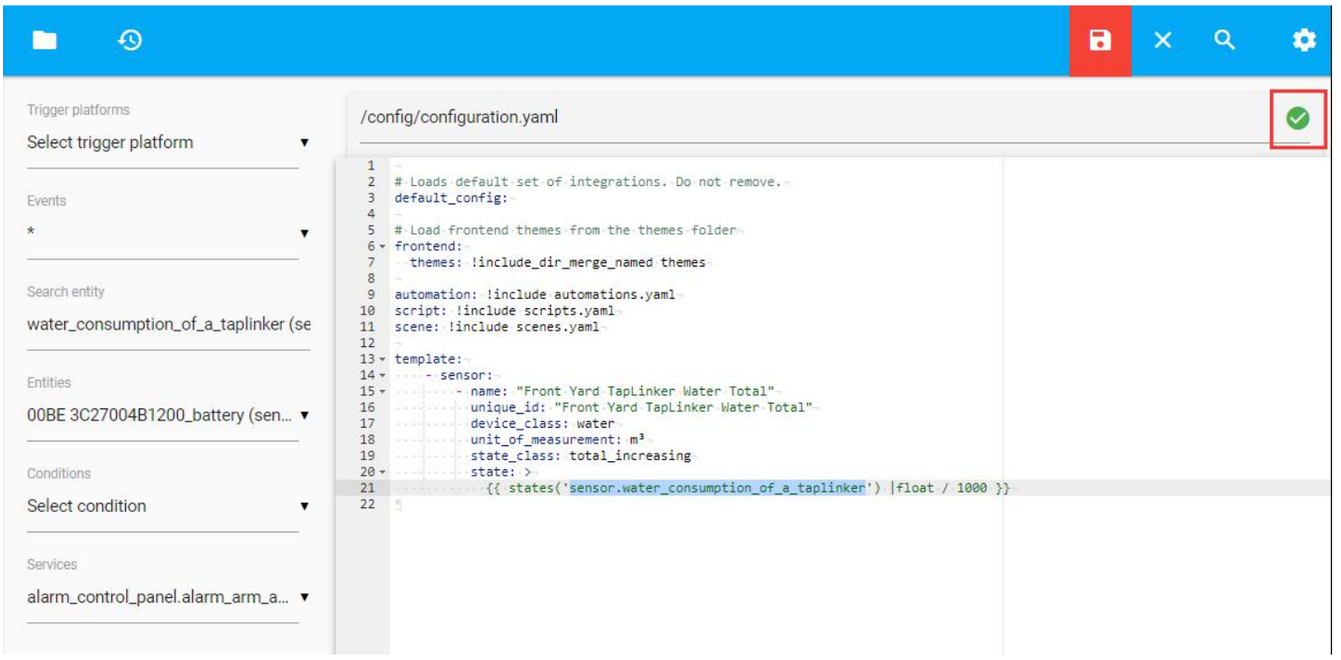


Afterward, at the end of the "configuration.yaml" file, enter content similar to the following.

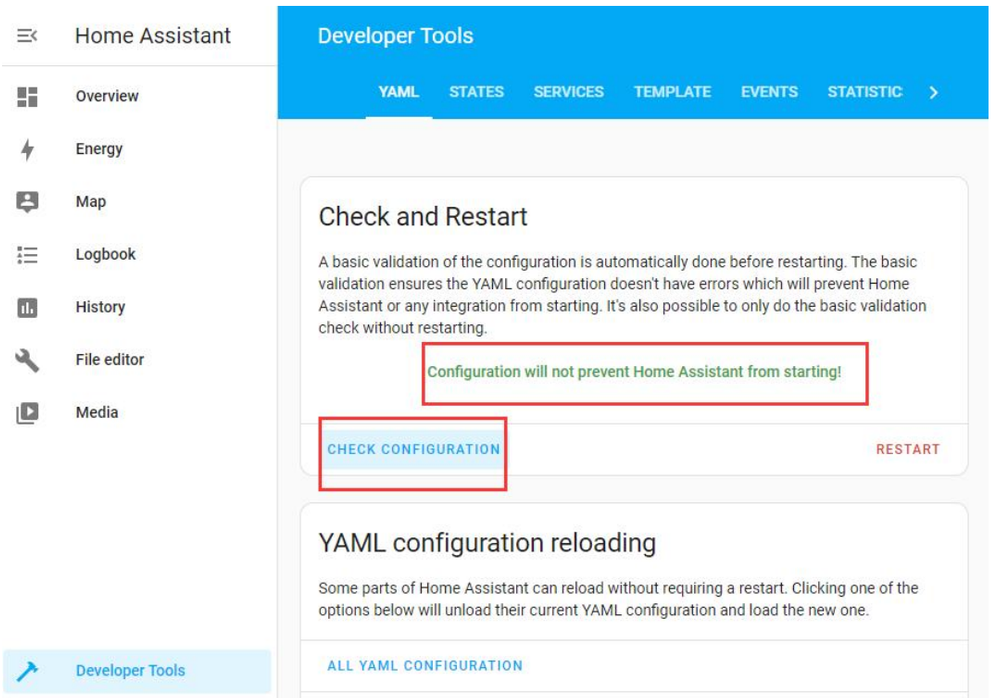
```
template:
  - sensor:
      - name: "Front Yard TapLinker Water Total"
        unique_id: "Front Yard TapLinker Water Total"
        device_class: water
        unit_of_measurement: m³
        state_class: total_increasing
        state: >
          {{ states('sensor.water_consumption_of_a_taplinker')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

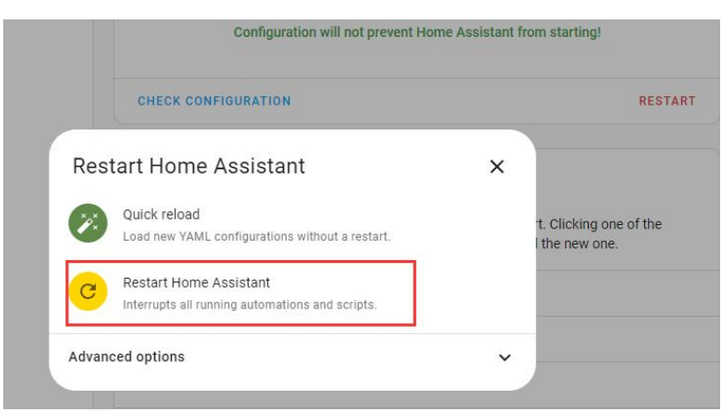
After completing the editing, you should check the icon in the upper right corner to ensure that the content you've entered is error-free. Once you've confirmed it's correct, click the red button to save the file.



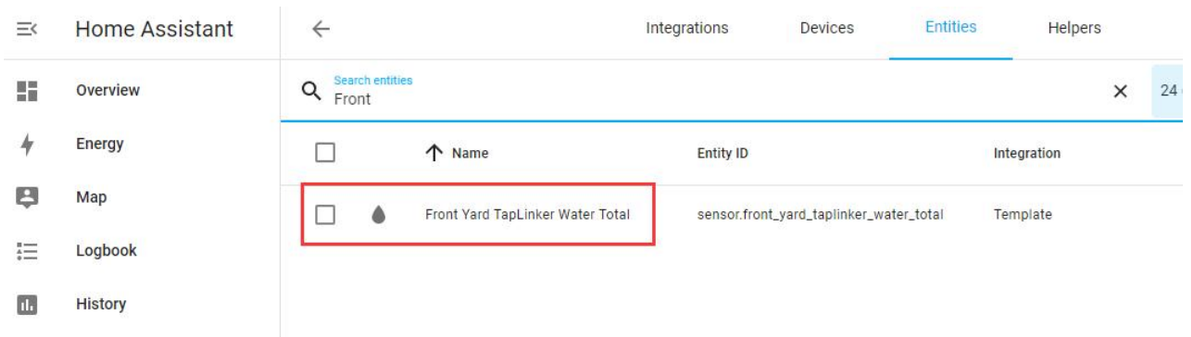
Afterward, you'll need to go to "Developer Tools" and click "CHECK CONFIGURATION" to verify your configuration once more to ensure it won't prevent Home Assistant from starting properly.



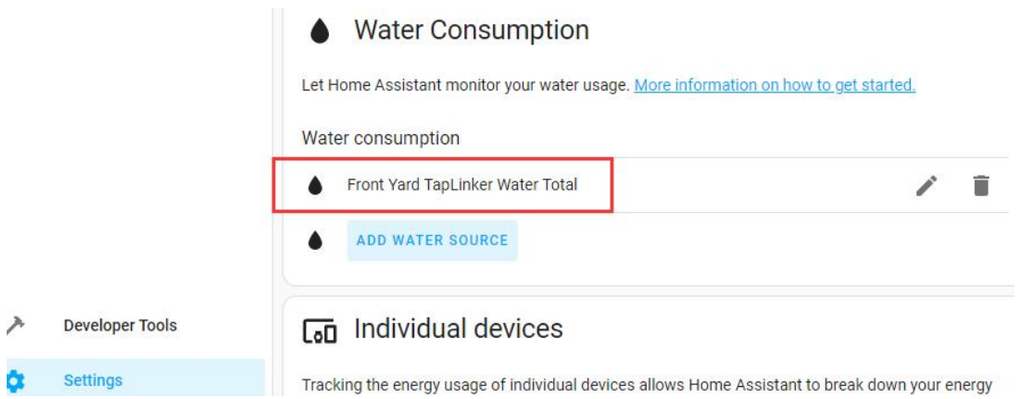
After confirming that the configuration is error-free, click on "RESTART" to restart the entire Home Assistant and ensure that your configuration is reloaded and takes effect.



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".



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

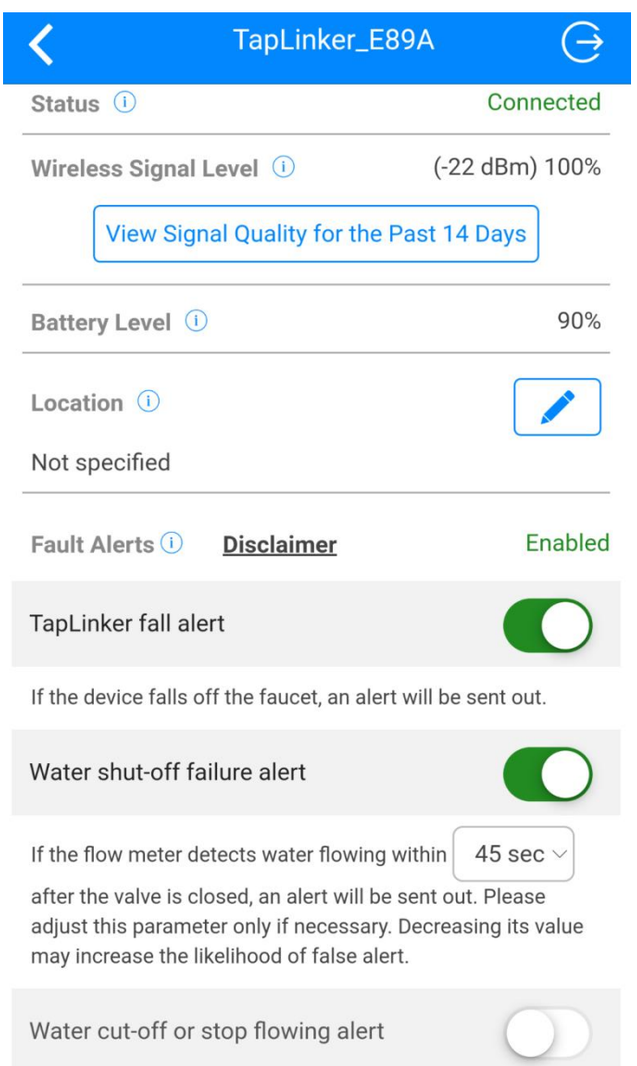


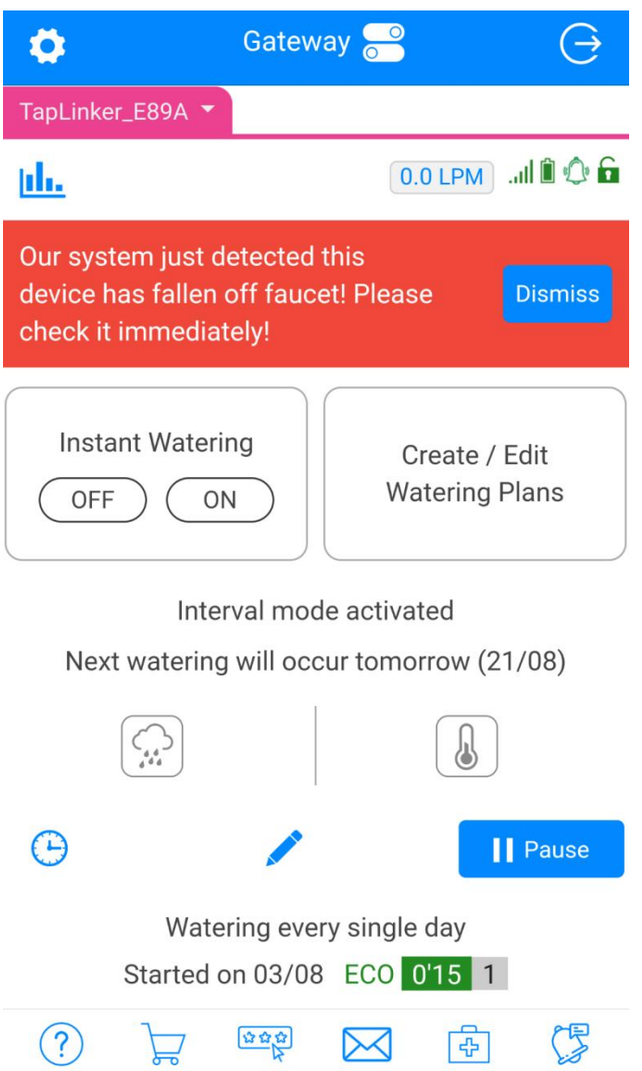
2.11. Taplinker's Fault Alerts on Home Assistant

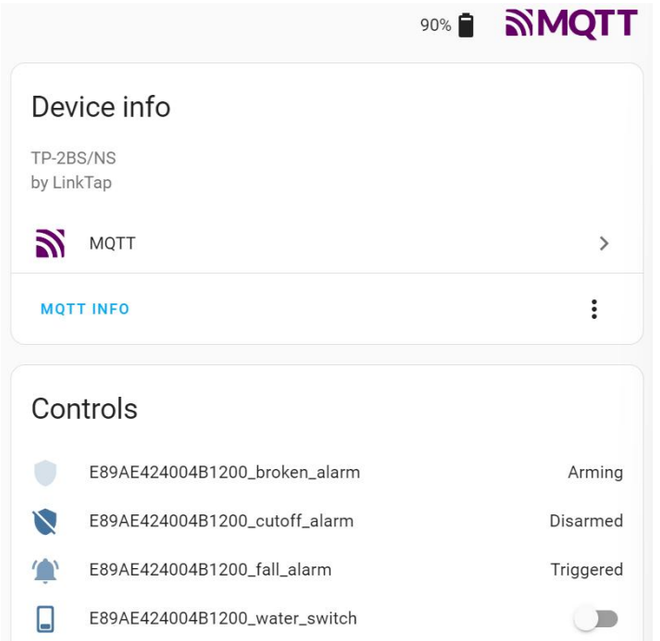
The LinkTap water timer's fault alarm function on Home Assistant is implemented through the alarm_control_panel platform. TapLinker fall alarm, water shut-off failure alarm, and water cut-off alarm can be received on Home Assistant now. Those alarms can also be dismissed from Home Assistant. Please note, 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.

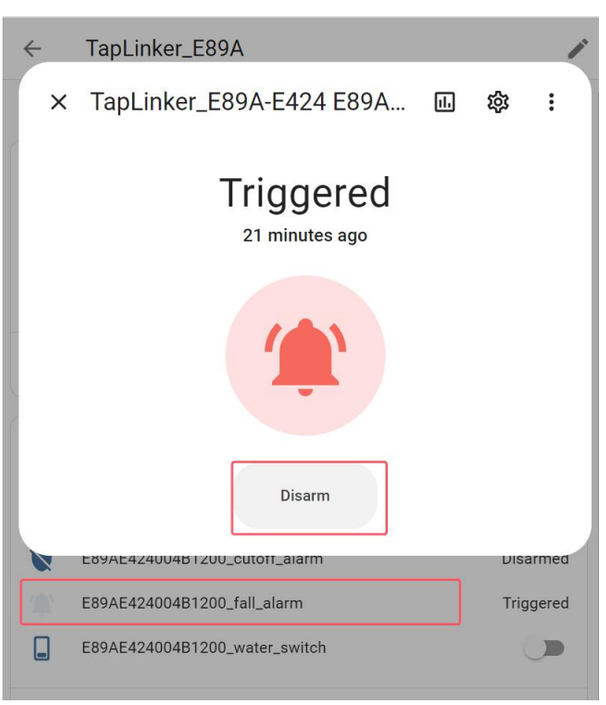
Below shows an example. The "TapLinker fall alert" and "Water shut-off failure alert" are enabled through the LinkTap app, and the "Water cut-off or stop flowing alert" is disabled. A fall alarm is generated after the TapLinker falls.





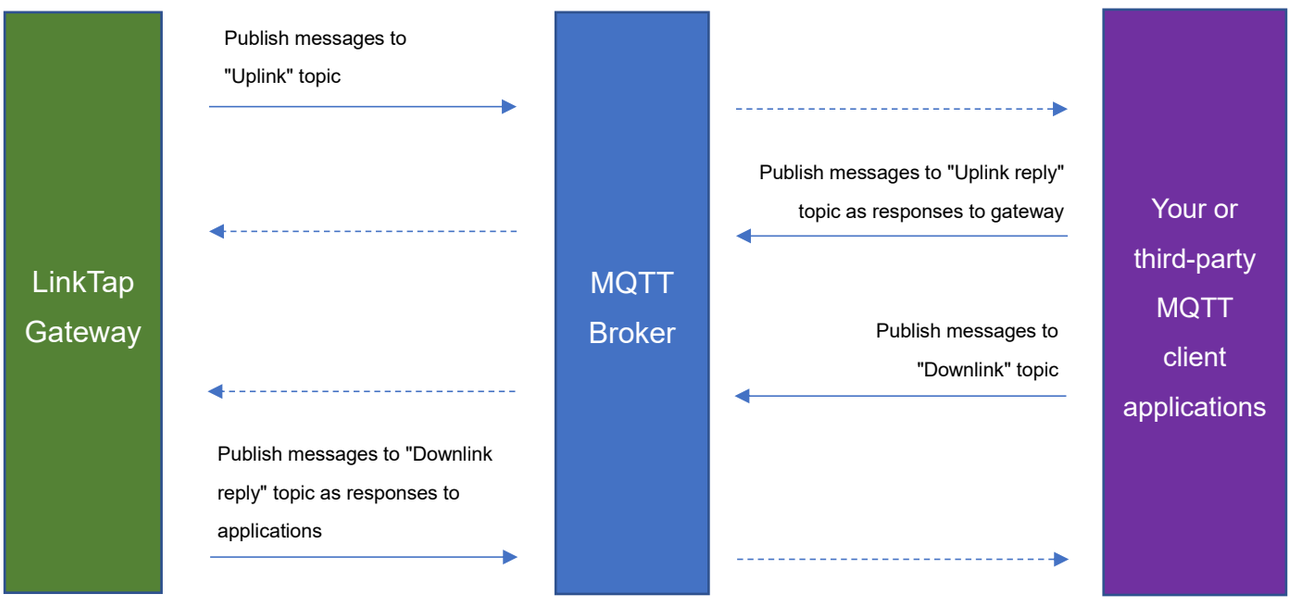


After checking the fault, you can click "Disarm" to clear the alarm, and then the alarm_control_panel status will automatically switch to "Arming".



3. Integrate with Regular MQTT Broker

3.1. System Diagram



The gateway and the third-party application communicate through four MQTT topics, which must be defined by the user on the gateway's web management page before the MQTT client function can be used. The "Uplink" and "Downlink" topics are essential. We recommend setting up all four MQTT topics.

Once the gateway connects to the MQTT broker, it regularly reports the device status via the "Uplink" topic and subscribes to the "Uplink reply" and "Downlink" topics to receive message replies and messages initiated by the third-party MQTT client. When the gateway receives a message initiated by the third-party MQTT client, it responds through the "Downlink reply" topic.

3.2. LinkTap Gateway MQTT Client 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
Submit 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.

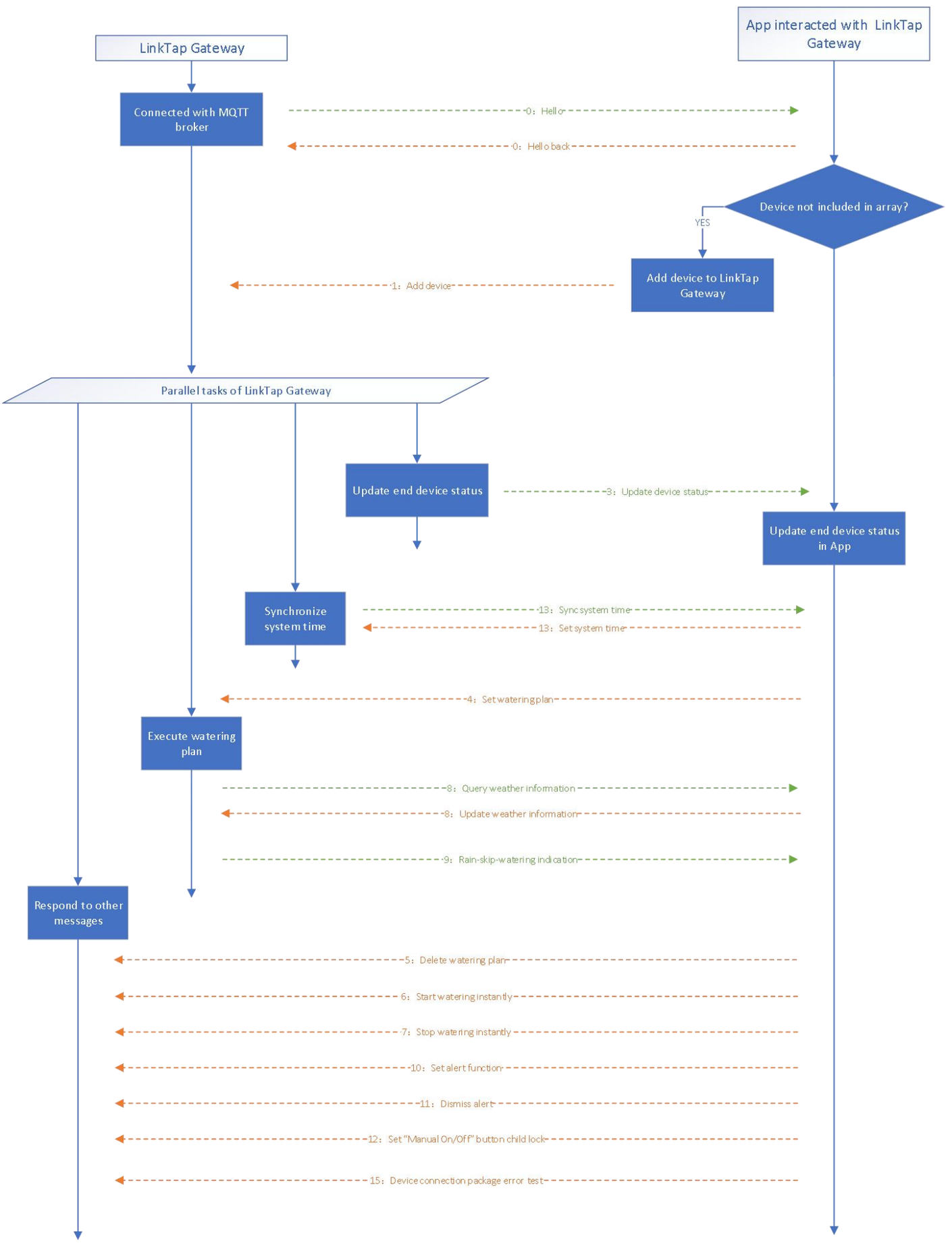
Click "Submit" button to save the settings. Next, we need to configure the MQTT topics.

- "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.



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 "MQTTX" 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.

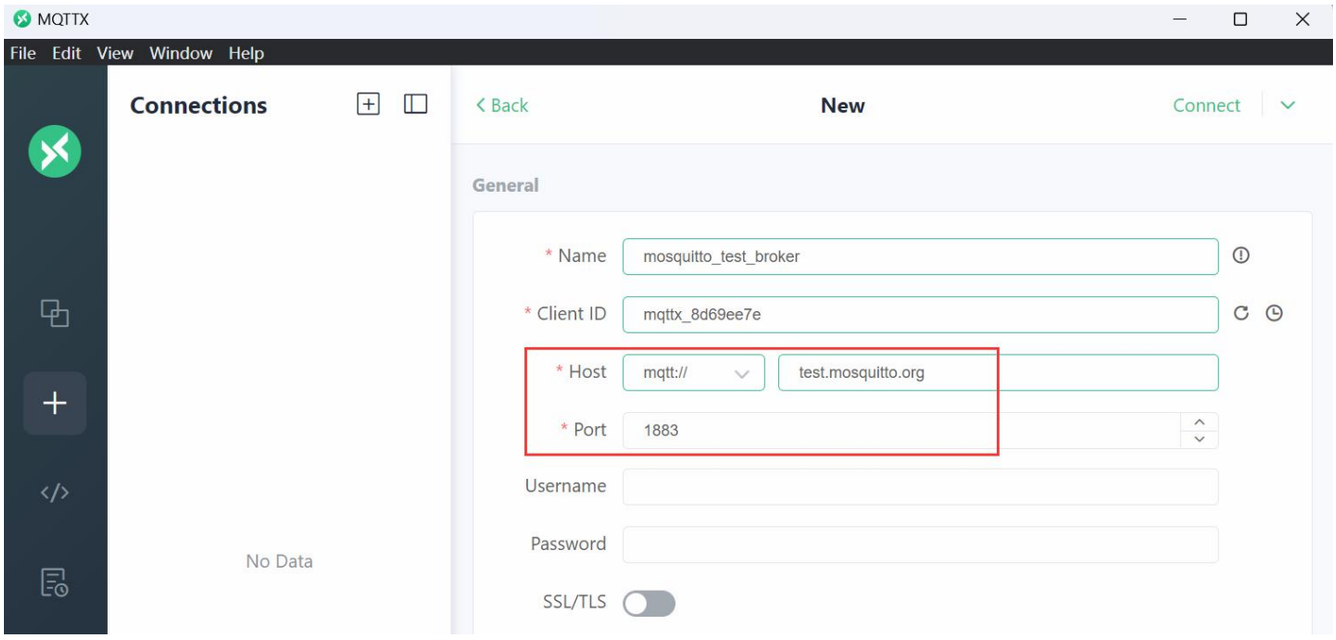
Note 1: When using "test.mosquitto.org", please set random value for Client ID, and leave the Username and Password fields blank.

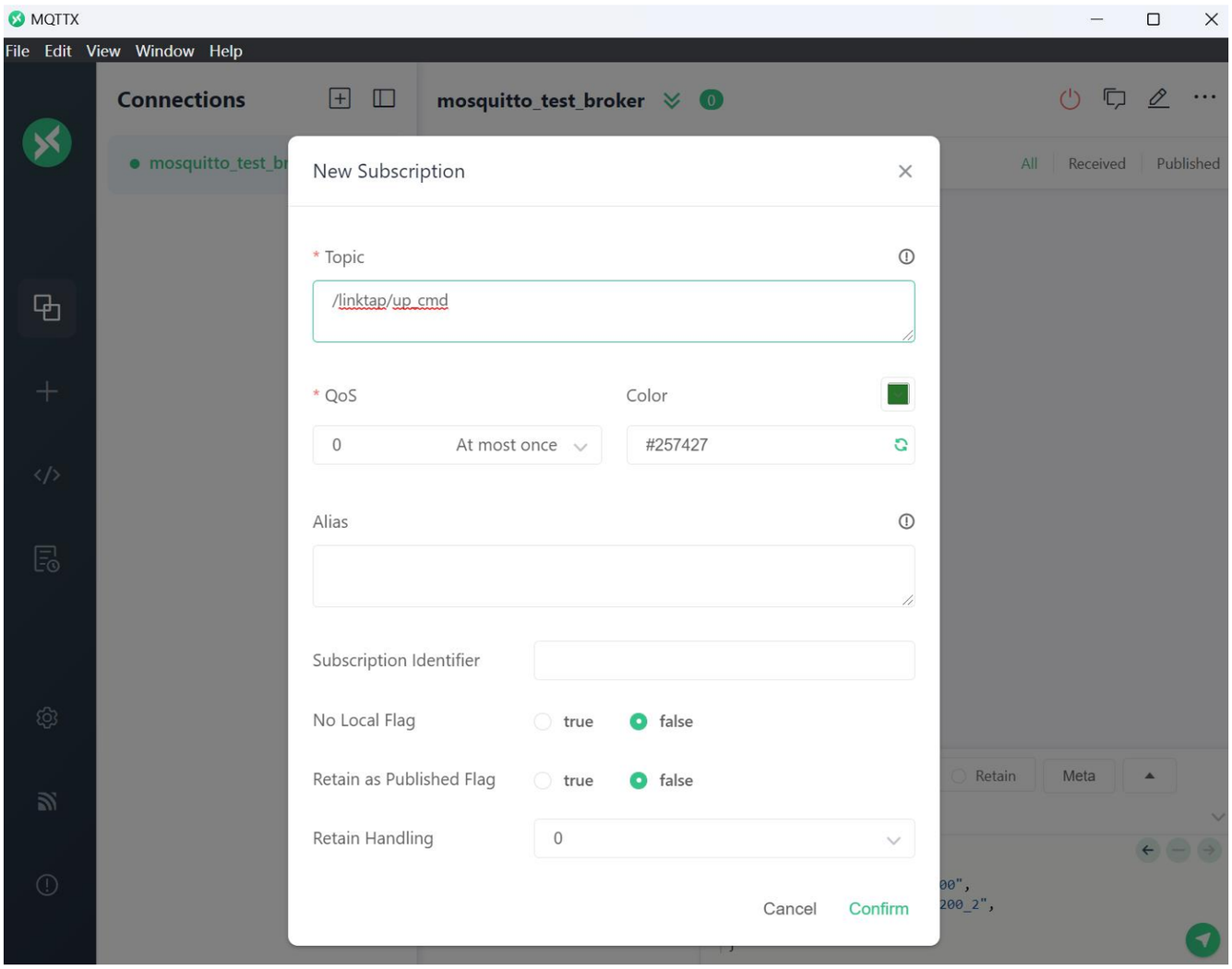
Note 2: Don't forget to reboot gateway after submitting the changes.

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	FF229328004B1200
User name	
Password	
Keep alive interval	120
Submit Any modification will only take effect after reboot.	
MQTT topics	
Uplink	/linktap/up_cmd
	◉ 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	/linktap/up_cmd_ack
Downlink	/linktap/down_cmd
Downlink reply	/linktap/down_cmd_ack

3.4.2. MQTTX Receives Messages from the LinkTap Gateway

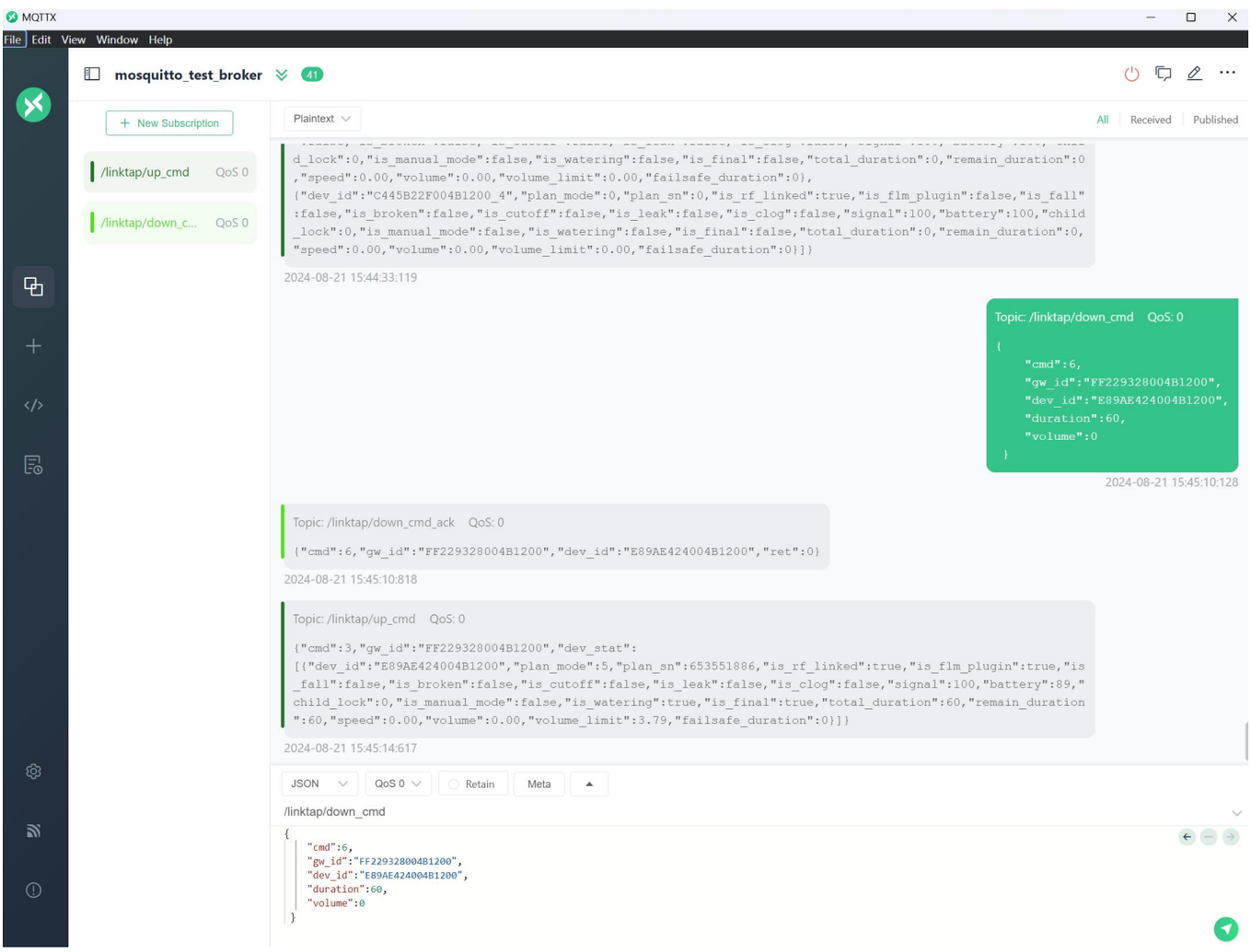
Following the gateway MQTT Client's settings, enter the corresponding MQTT Broker address and port on MQTTX, then click "Connect" to connect to the MQTT Broker. After connection is established, subscribe to the configured "Uplink" topic to receive the messages published by the gateway.





3.4.3. Sending Messages to and Receiving Responses from LinkTap Gateway via MQTTX

Subscribe to the configured "Downlink reply" topic to receive response messages from the gateway. Then, following the message definitions in the "LinkTap Gateway MQTT Client Interaction Messages.pdf", publish a message through the "Downlink" topic. After a while, if the message successfully reaches the gateway, MQTTX will receive the gateway's response message.



3.5. Implementation of Third-Party Application

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

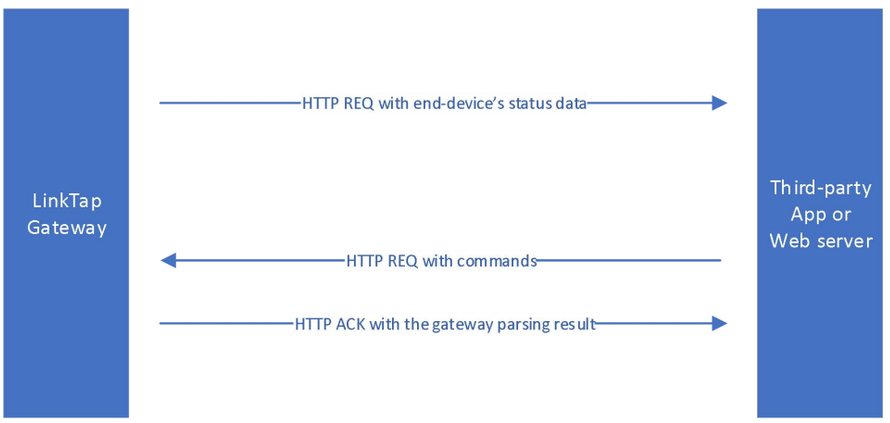
3.5.1. Program and execute watering plan within the application

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



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 "HTTP Client" 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 HTTP Client
Server URL	http://Domain_Name_or_IP:Port/Path
Response	☑ Wrap the gateway's response in HTML

4.1. Push data to server from the gateway

If the gateway's "HTTP Client" is enabled, after powering on and obtaining an IP address, the gateway's HTTP client will automatically report the status of the LinkTap water timer to the HTTP server specified by the "Server URL" via an HTTP POST request. The timing and frequency of the reports are as follows:

- When the water timer's status (watering status, signal strength, remaining battery, etc.) changes;
- When the water timer's status remains unchanged, it will report every 2 minutes.

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",……}

The HTTP interaction message format between the gateway and third-party applications or web servers is the same as the MQTT interaction messages from the gateway. For the detailed message format, refer to CMD 3 in "LinkTap Gateway MQTT Client Interaction Messages.pdf".

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.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. By default, the gateway's response will be wrapped in HTML tags, as shown below:

HTTP/1.0 200 OK
Server: LinkTap Gateway
Content-Type: text/html
Expires: Fri, 10 Apr 2008 14:00:00 GMT
Pragma: no-cache

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

You can disable "Wrap the gateway's response in HTML" in the gateway's management page to have the gateway respond directly with JSON data, as shown below:

HTTP/1.0 200 OK
Server: LinkTap Gateway
Content-Type: application/json

{"cmd":x,"gw_id":"CCCCDDDDEEEEFFFF","ret":y}

For the JSON data format in the reply message, refer to "LinkTap Gateway MQTT Client Interaction Messages.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.