Home Assistant Application for IP POWER

(For Windows Version)

AVIOSYS's IP POWER can now be used in Home Assistant. In this system, you can set your IP POWER according to your own needs. Please follow the steps in the text:

1. Home Assistant installation:

To install Virtual Machine(VM), you can use the officially provided VM or use other VMs to operate.

This manual will use "Oracle vm virtualbox" as an operation demonstration.
If you are already using other Visual Machines, you can skip the installation steps directly.

Before installation, please make sure you see the basic requirements below:
(Please refer to the official website for the minimum requirements)

(1) A computer that can operate normally, preferably a CPU runs with multiple cores.
(2) Your PC's RAM at least 4 GB
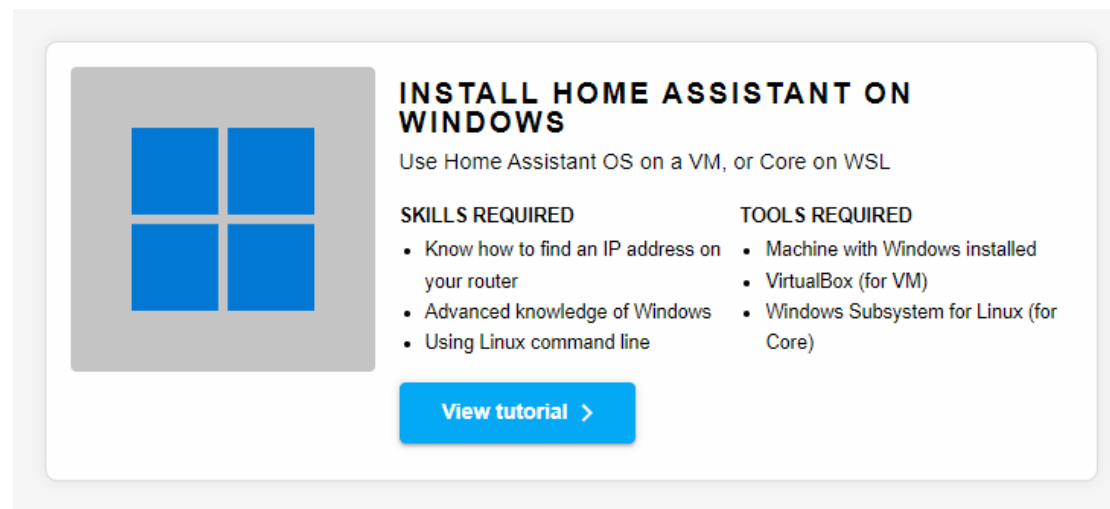(3) The disk has sufficient capacity (at least 32GB)

In addition to VM, be sure to install VirtualBox (.vdi), otherwise HomeAssistant cannot run.

Setup tutorial:

1. Download the file
(1-1) Please enter the official download page of Home Assistant, select the Windows OS version installation tutorial on this page, and click View tutorial.

Home Assistant web link: https://www.home-assistant.io/installation/

(1-2) After entering the webpage, please download and install the files according to the links provided on the official website. VirtualBox (.vdi) must be installed, and other files can be decided based on user preferences.

For users using VM for the first time, it is recommended to follow the instructions on the official website or refer to user operations on the internet.

(1-3) VM installation VirtualBox (.vdi) and configuration tutorial (VM may be different depending on the user, this method is for reference only)

Demonstration VM: Oracle vm virtualbox

(1-4) After the Oracle vm virtualbox installation is complete, download Virtual Box (.vdi), create a new folder in your disk, put Virtual Box (.vdi) into this folder and unzip it.

After Unzip the file:
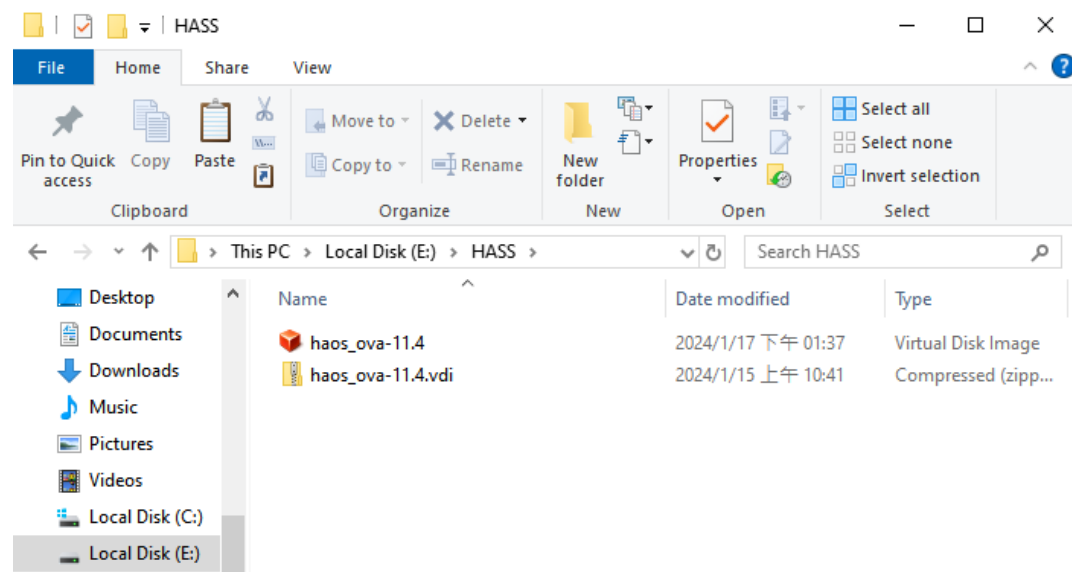
2. Install Home Assistant

(2-1) Open Oracle vm virtualbox and click "new". There are three main places to set:

   (1) Name and operating system – fill in name only
   (2) Hardware – Set your VM CPU and RAM usage.
   (3) Hard disk – click "Use an existing virtual hard disk file"
            and select the Haos_ova-11.2 file

After the settings are completed, you can see the newly added devices, then click Settings to complete the final process.

*Notice: Home Assistant cannot be started normally if it is not completed.

(2-2) Enter the VM settings, please change the following two.

(1) Audio: Please make sure your audio controller is set to Intel HD Audio by
    default.
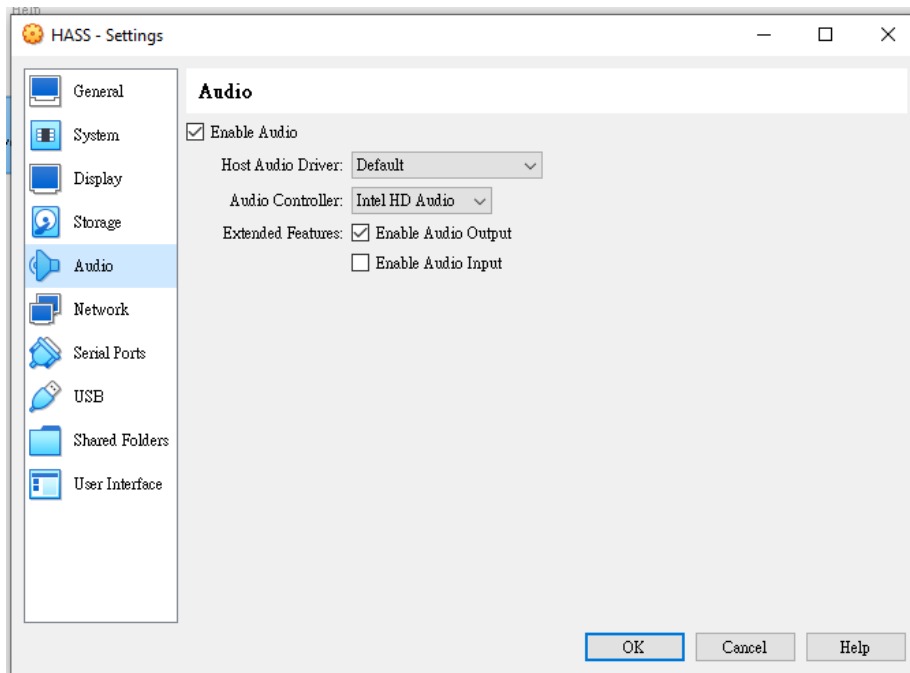
(2) Network: Select "Bridge Adapter" in "Attached to". and select the name
inside.

(*Notice: The image content is for reference only. The content may vary
depending on the user's device content.)
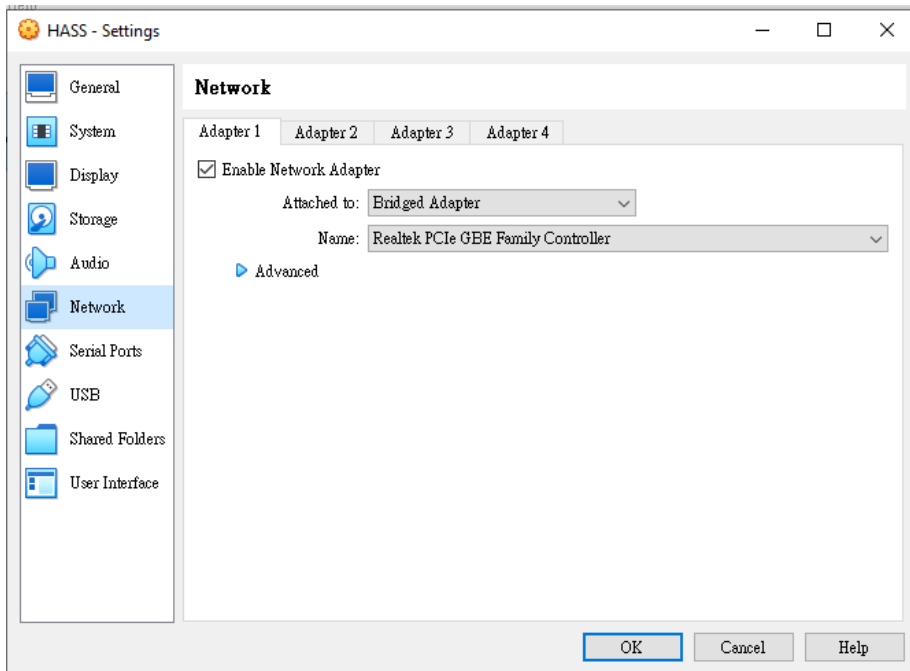
After setting the above changes, click "OK" below to complete the
modifications.

After the setting is completed, start Home Assistant. When the VM finishes running the program code, the IP of Home Assistant will appear. Please copy the IP, type it on the domain and add ":8123" after it. 8123 is the official designated communication port.

EX: "192.xx.xxx.xx:8123" or "10.xx.xxx.xx:8123"

*If you are using an older version of Windows, please use homeassistant:8123 instead of IP input.

Pic:
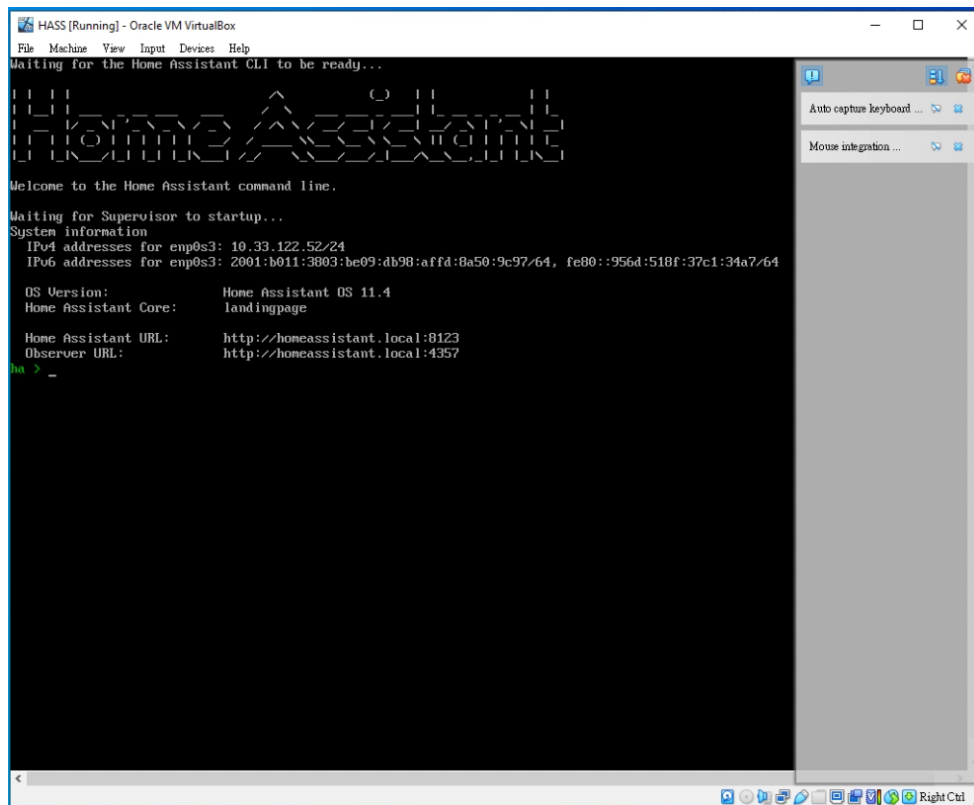
(2-3) After entering the official website, you can see that Home Assistant is connected. As this is your first time opening it, please be aware that you'll need to wait for 10-30 minutes for system running. After this initial wait, there is no need to wait, you can simply log in with your username and password afterwards.
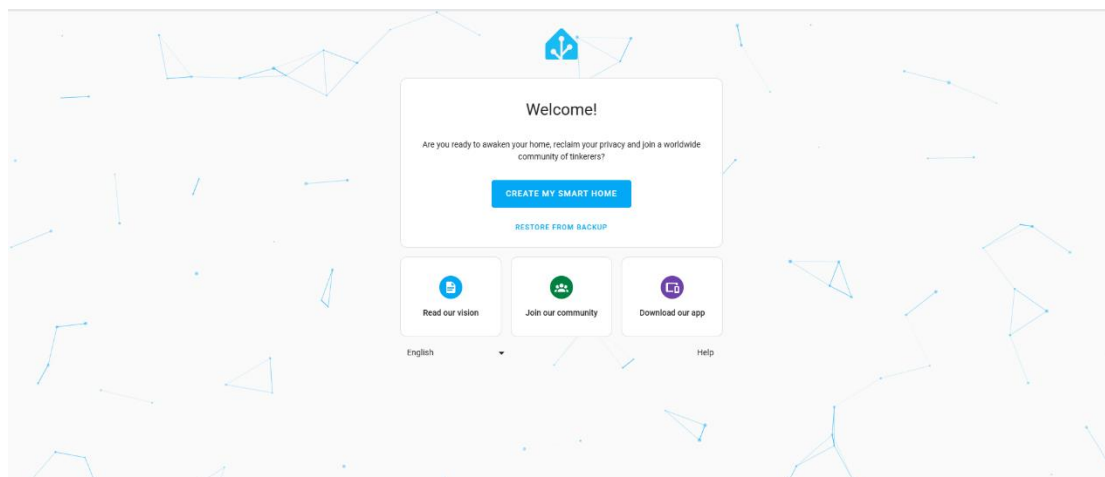
Once the waiting period is over, proceed to create your account and password. Please remember that this account and password belong to the virtual machine(VM) you establish, not to Home Assistant's online account. If you ever lose your account and password, simply delete the current VM and restart a new one to set it up again.

*For your security, it is recommended to strengthen the password when creating a user, and also remember the account and password you set.
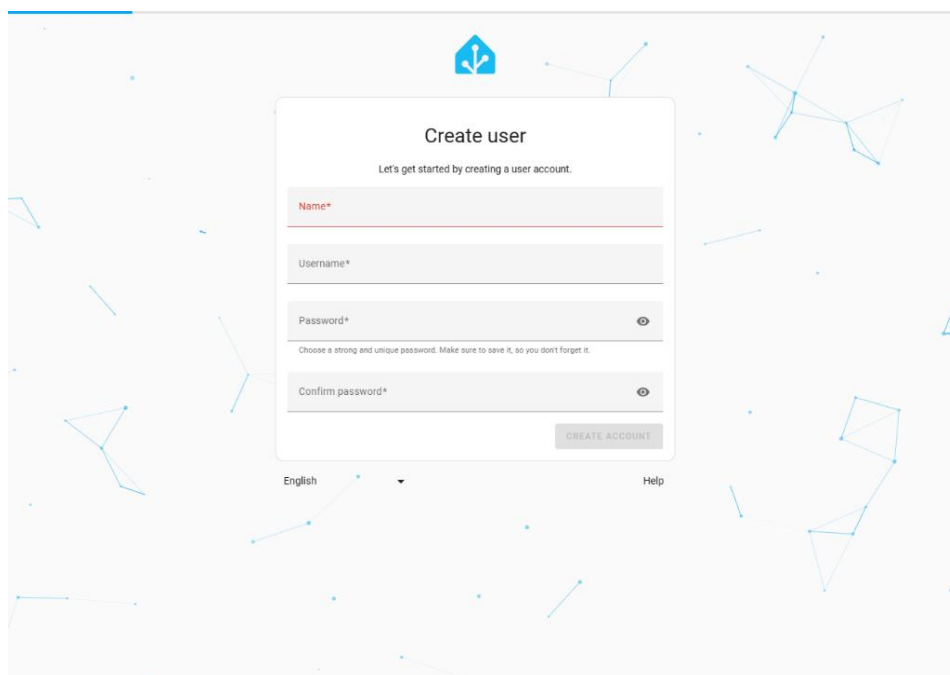
(2-4) To set your location, you don't need to fill in details,
     just fill the city where you at.
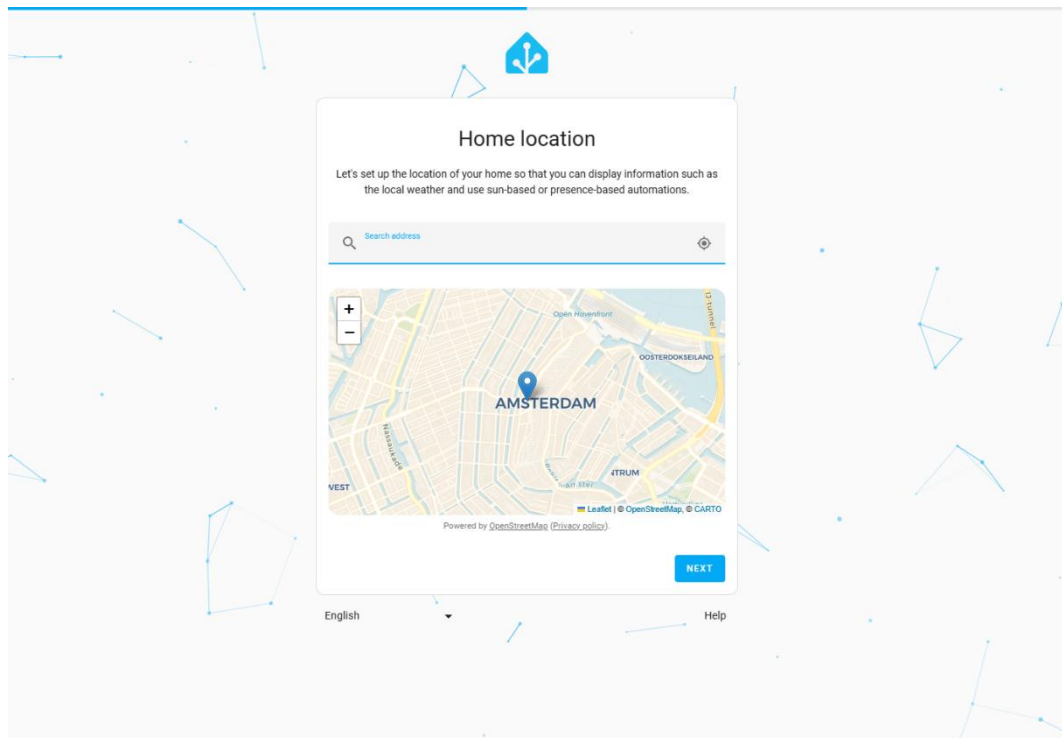
(2-5) When you enter the main page, it means you have successfully entered
Home Assistant. Follow the red arrow and click Settings below the picture.
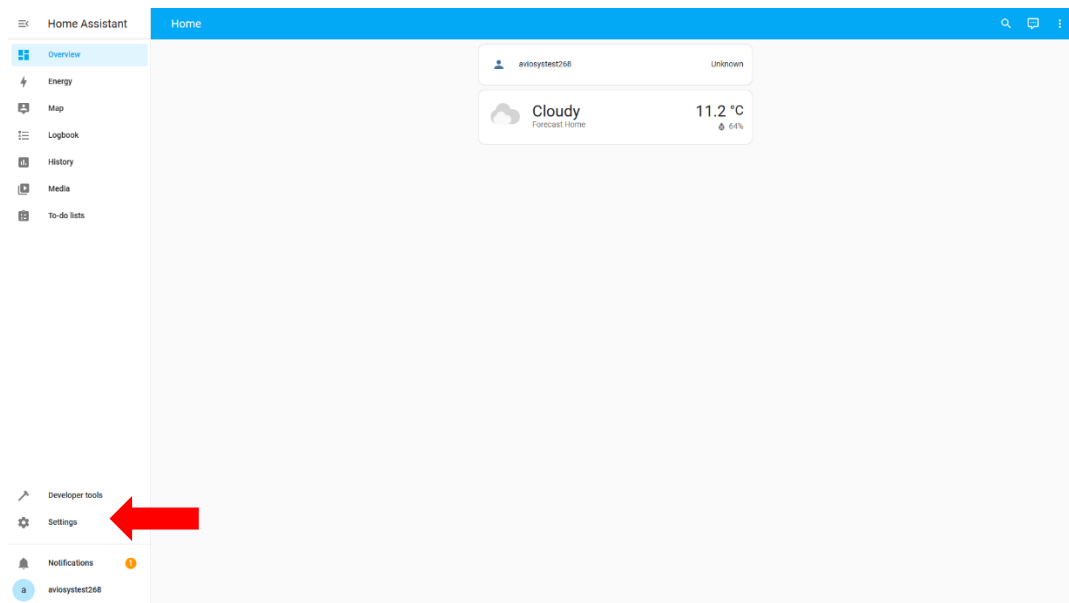
When you entering the setting webpage, click the "Add-ons"(1), and click
"ADD-ON STORE" in the lower right corner.

(1)
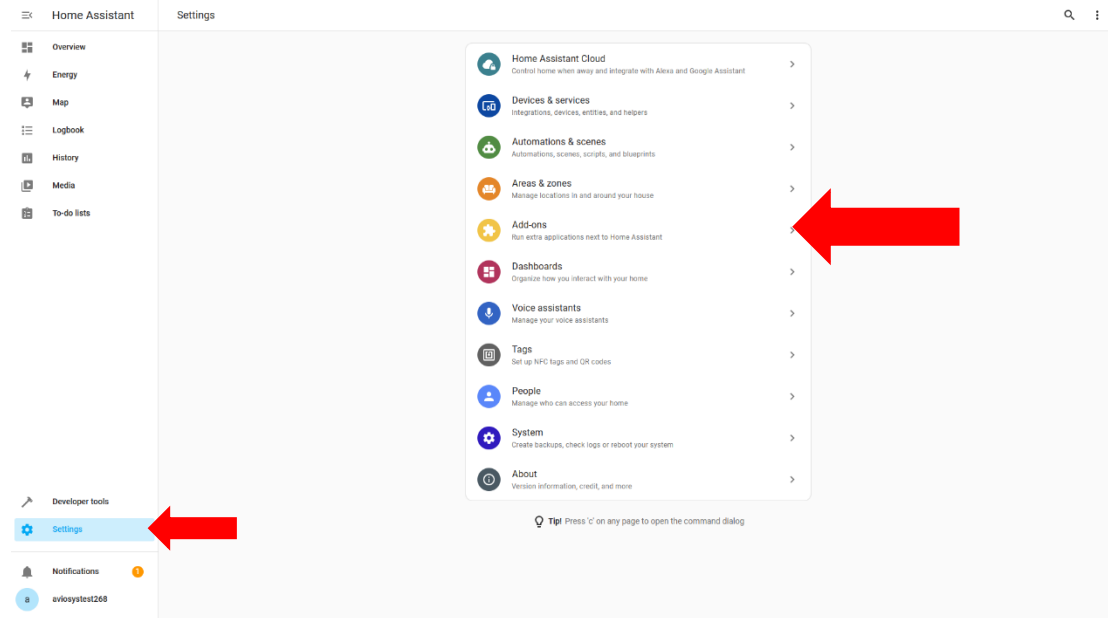
(2)
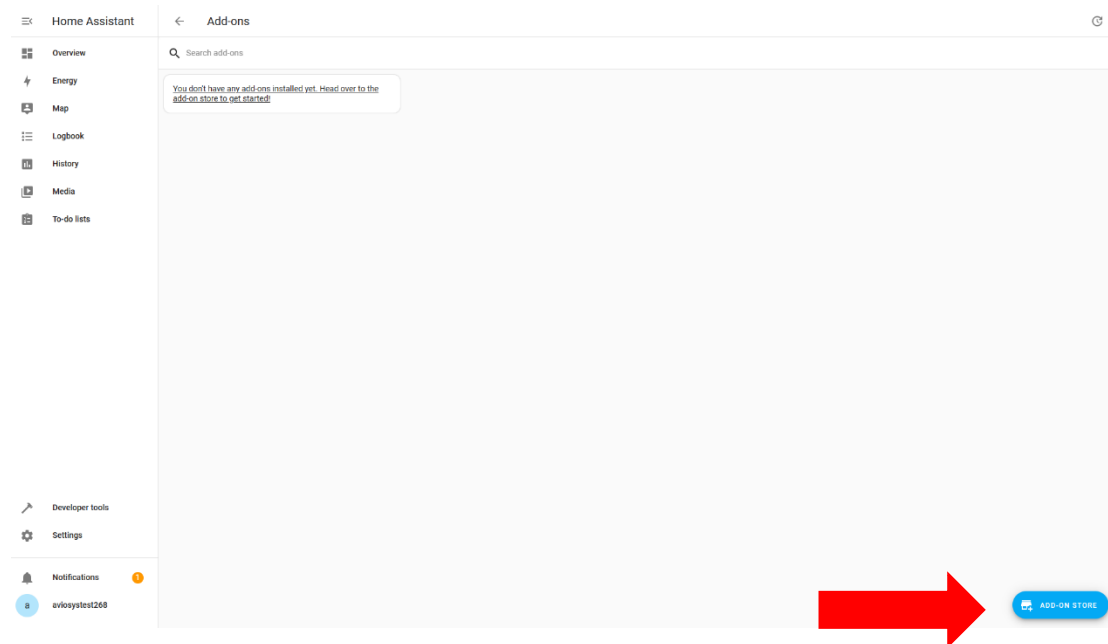
Find the "File Editor" and install it to your Home Assistant.
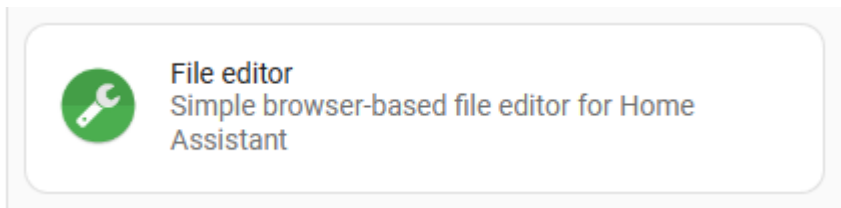
After the installation, remember to activate (1) to open at startup and (2) to display in the sidebar. The watchdog function can be determined based on personal preferences.
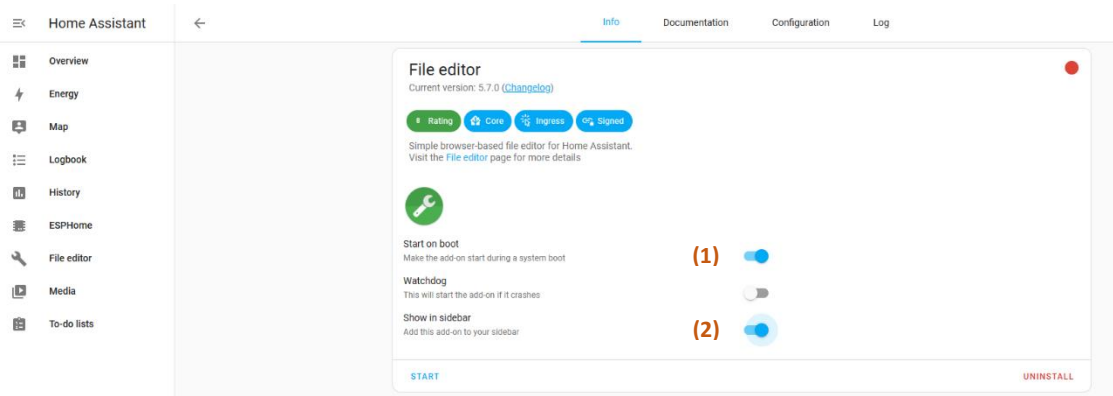
3. Setting your IP POWER

Open the File Editor, click on the folder that the red circle shown, and open the "configuration.yaml" file.
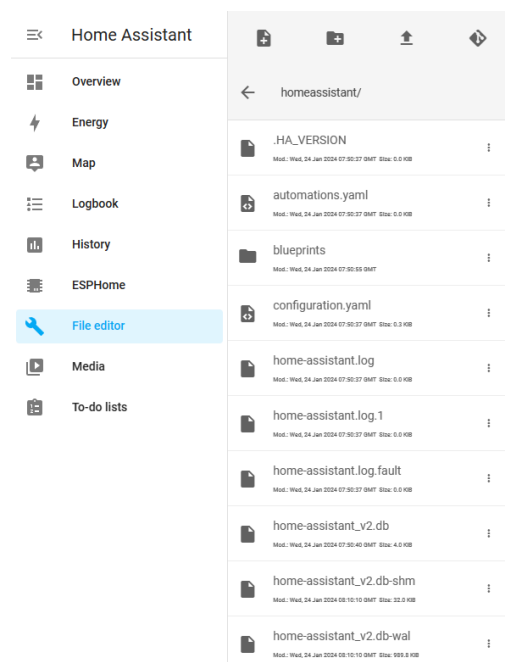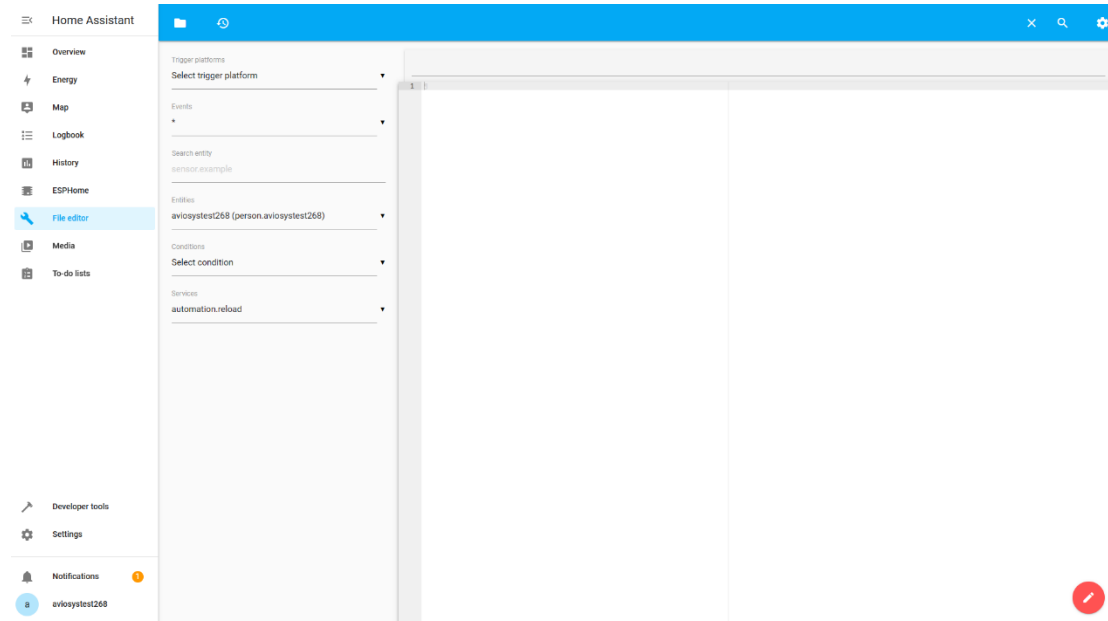
After the file is added, please select the IP Power you want to configure, and then follow the instructions on page P.19 to enter the script.

* Notice: Please be sure to enter the correct format, IP, text, etc. in the command code. If there is any error, the editor won't let user save the script, as long as the system gives a green check icon, you can save the settings.

After entering the script code, as long as the system gives a green check icon, you can save the setting.
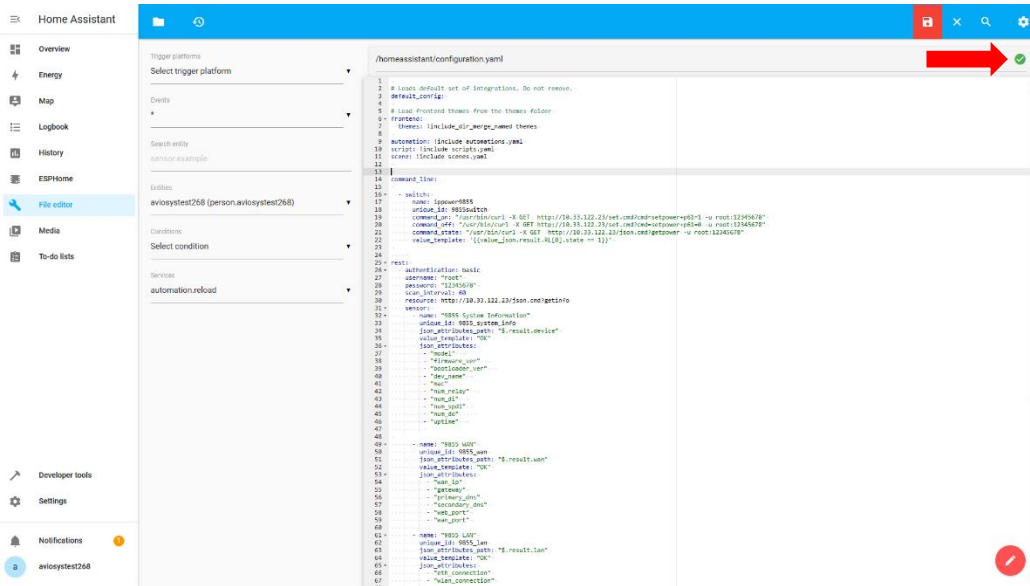
After saving, Home Assistant will not immediately show the IP POWER you have configured. A restart is required for the changes to take effect.

Before doing so, to ensure that your settings work properly, please click on the developer tools and select:

1. Check and restart → Check configuration
2. YAML configuration reloading → All yaml configuration
As long as the system does not display any errors, you can proceed with the restart.
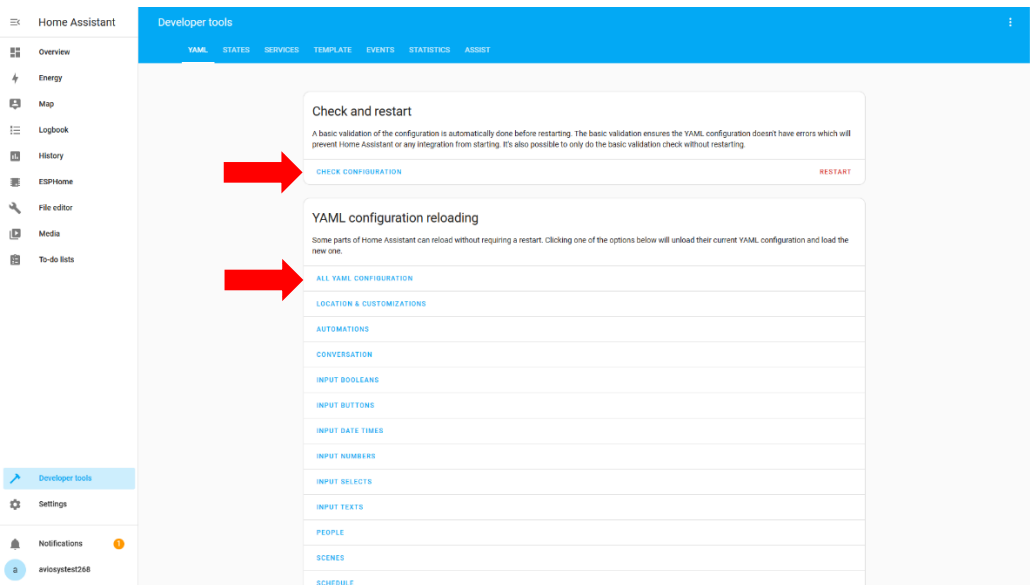
Restart you Home Assistant:

(Optional, just in case the display fails after saving settings.)

Click the icon in the red circle on the upper right and click "Restart Home Assistant".
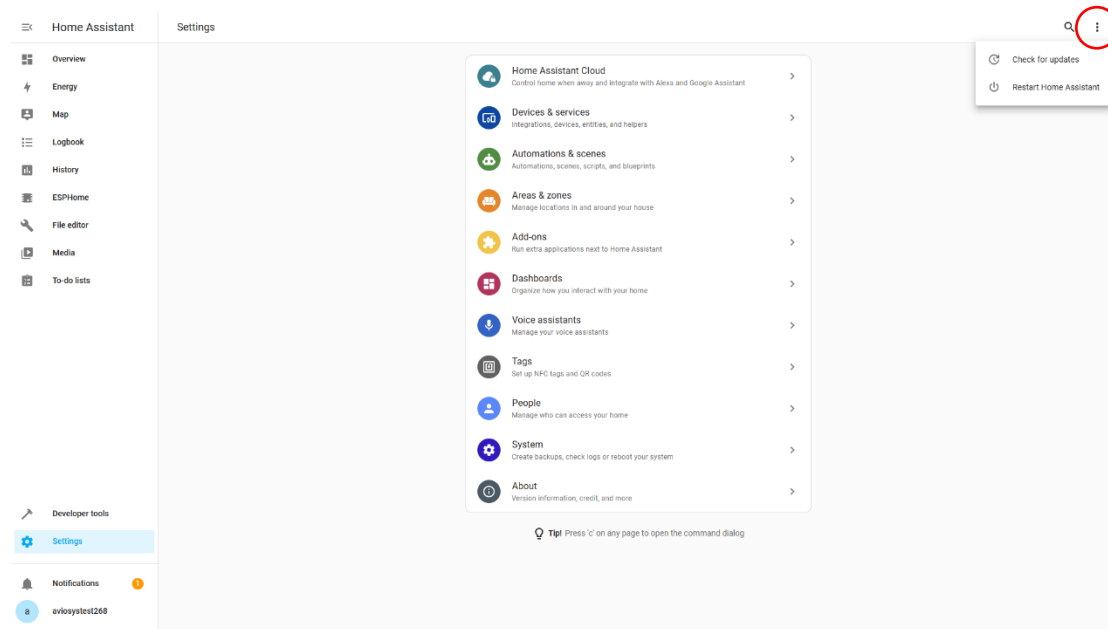
Please click on the yellow icon "Restart Home Assistant" to initiate the restart process for the VM.
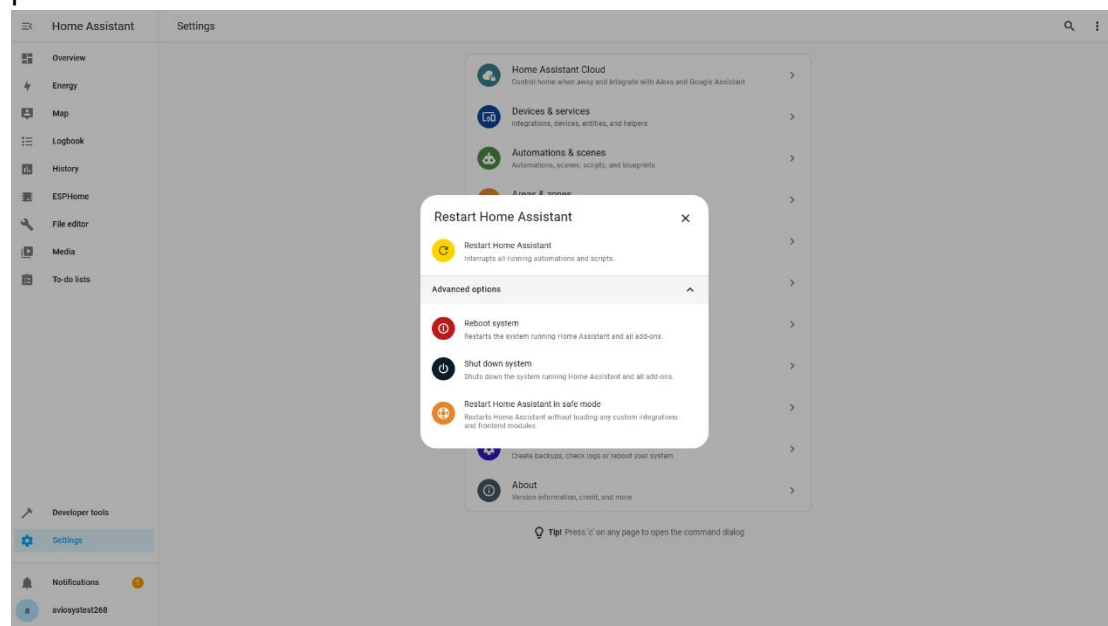
After the system restart, you can see that the IP POWER device has been integrated into Home Assistant and can perform operations.
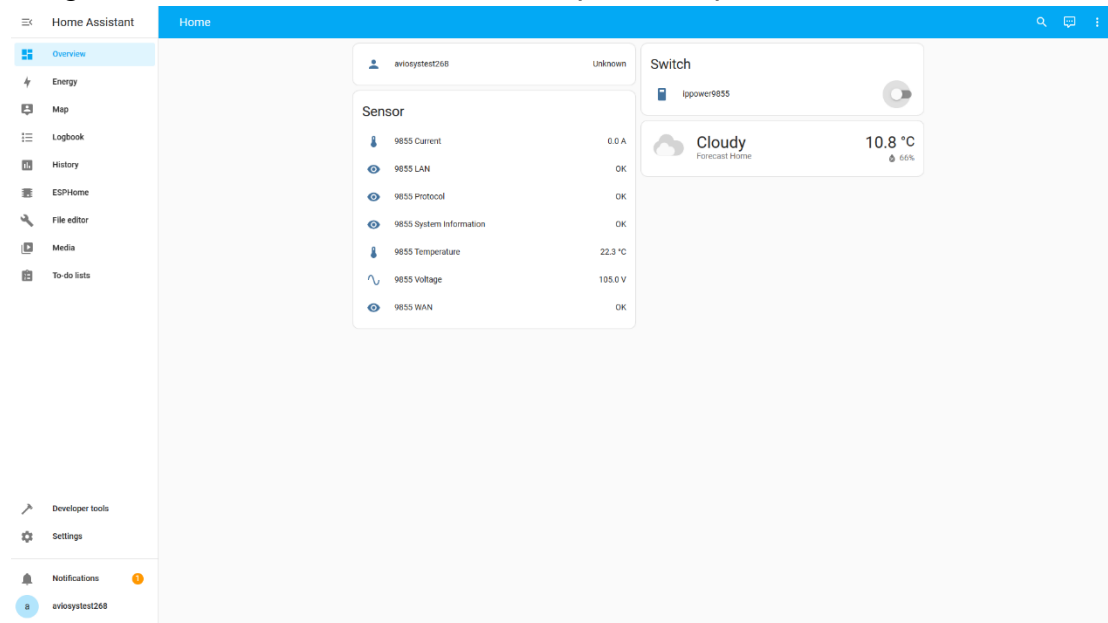
You can also click on other functions to observe records, such as temperature, voltage, and current.

Notice!!! The IP Power device you have set up must have these functions, otherwise, it won't display on Home Assistant.

Temperature:
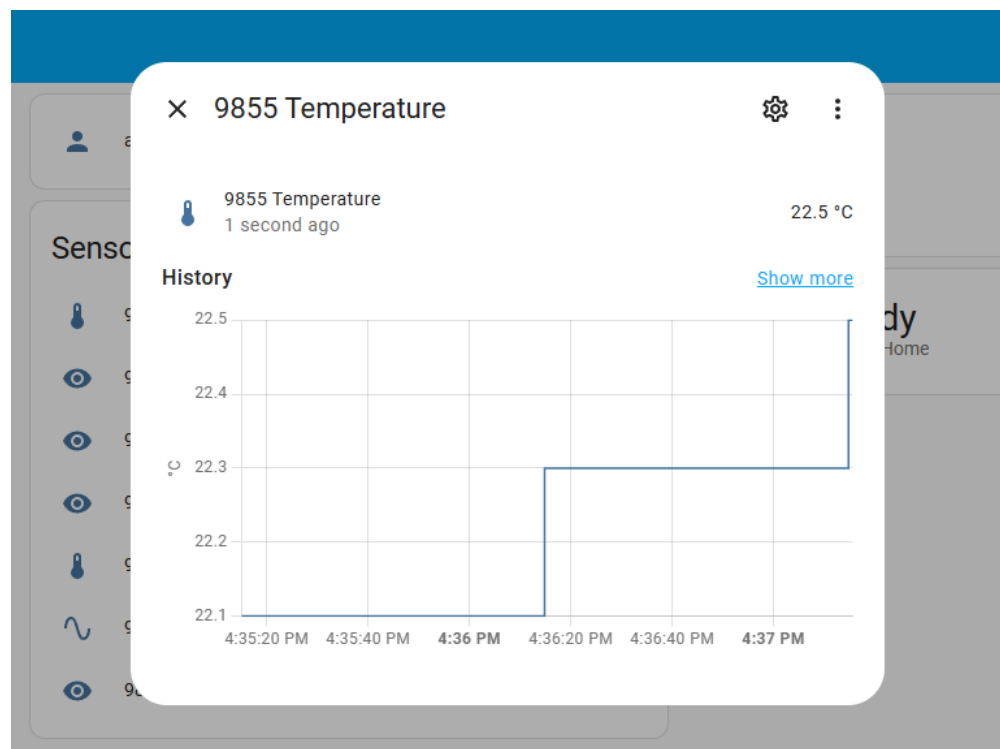
Voltage:
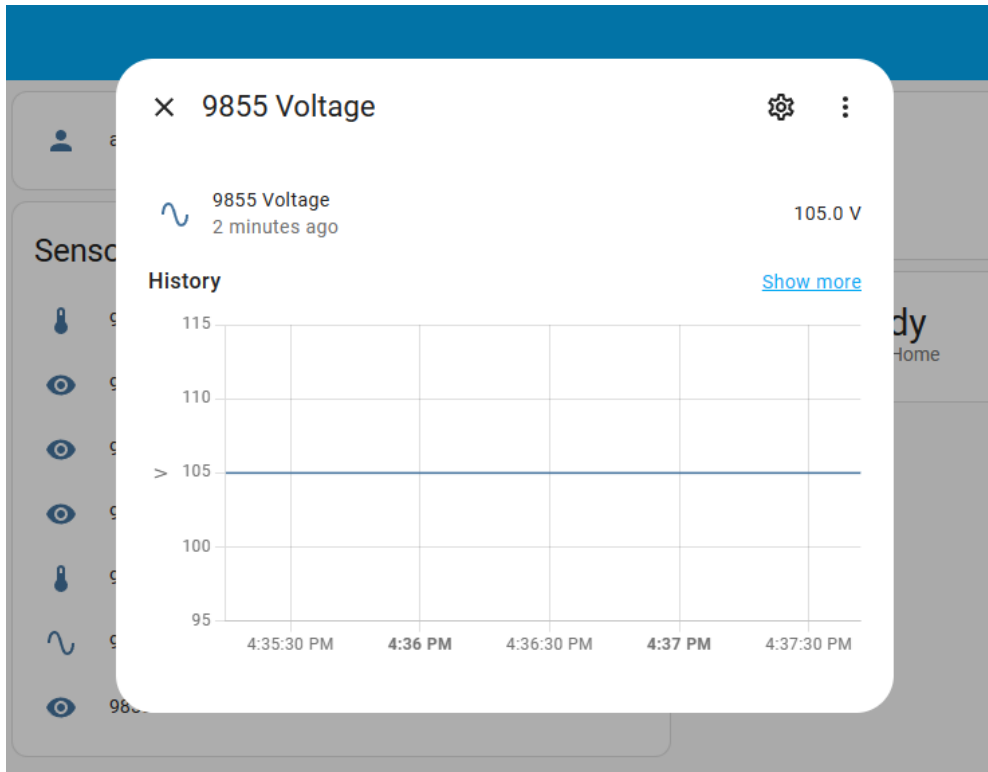
Current:
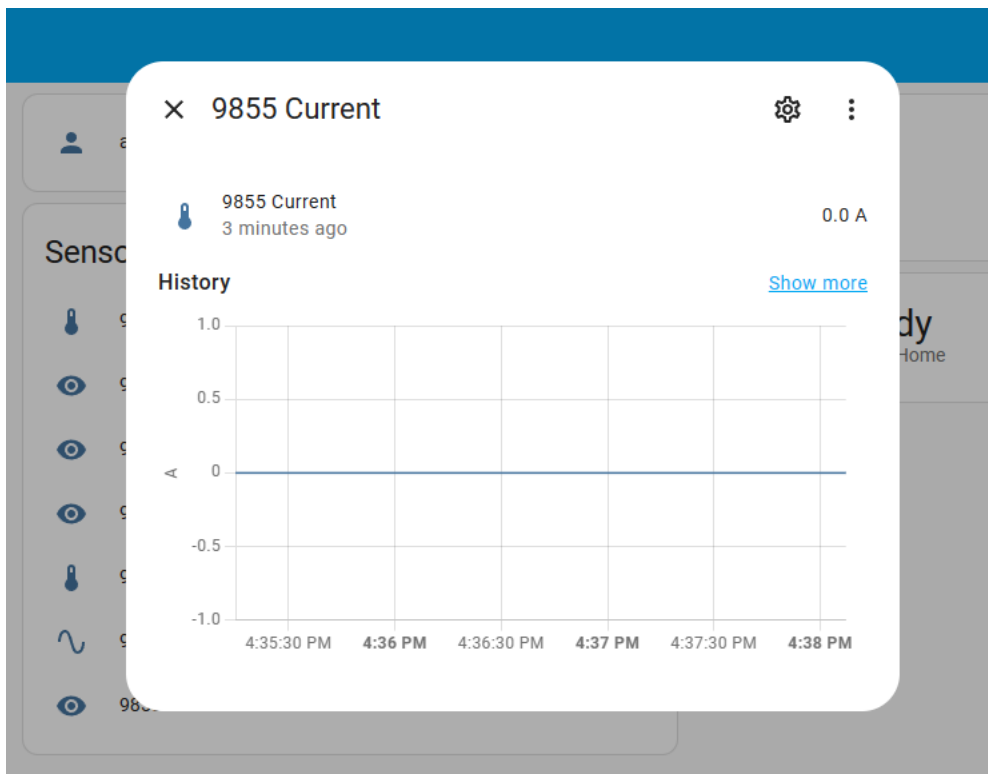
4. Using MQTT script:

This allows users to connect and control IP Power devices outside without connecting the intranet. (Currently only supports IP Power 98 series)

Please follow the procedure below:

(4-1) First, please turn on your IP POWER device and enter Application→IP Service.

Select CNT(MQTT) Setting, and click MQTT Service, select Custom.

CNT(MQTT) Setting (Remote control)	
MQTT Serivce	IPPower ▾
Enable	☐
Status	
Apply	

CNT(MQTT) Setting (Remote control)	
MQTT Serivce	Custom ▾
Enable	☐
Server :	
Port :	
User :	
Password :	
SSL/TLS	☐
Enable Server Certificate	☐
Status	
Apply	

(4-2) If you already have a dedicated MQTT broker, please fill in the information, SSL/TLS and Certificate should be configured according to user requirements, but remember to click "Enable" After setting, please click "Apply". If "Connect" appears in Status after clicking, it means it is successfully activated.

If you don't have your own MQTT broker, you can choose from some online free MQTT brokers. For example: broker.emqx.io. Please do not choose test.mosquitto.org because it has compatibility issues with our devices. When selecting a public broker, please try to use TLS secure connection on port 8883 as it will be exposed to all platforms integrated with this MQTT broker, so please be mindful of security risks.

*If you want to use our company's IP POWER as MQTT Broker,
 please contact with us.

(4-3) After setting up, please enter the UI of Home Assistant, click on the settings in the bottom left corner, and select "Devices and Services."
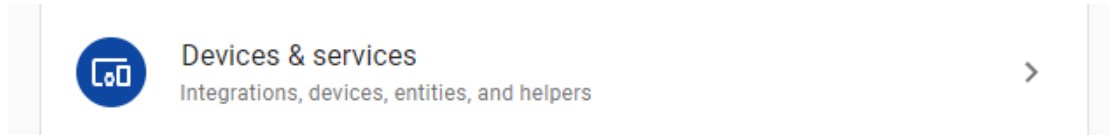
(4-4) Once inside, click on "Add Integration" in the bottom right corner of the screen.

Add MQTT and integrate it into your Home Assistant. After confirming the addition, click on MQTT and start configuring your MQTT.
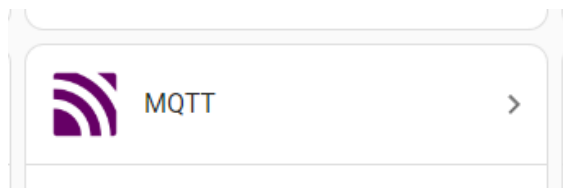
After entering, click ADDD ENTRY.
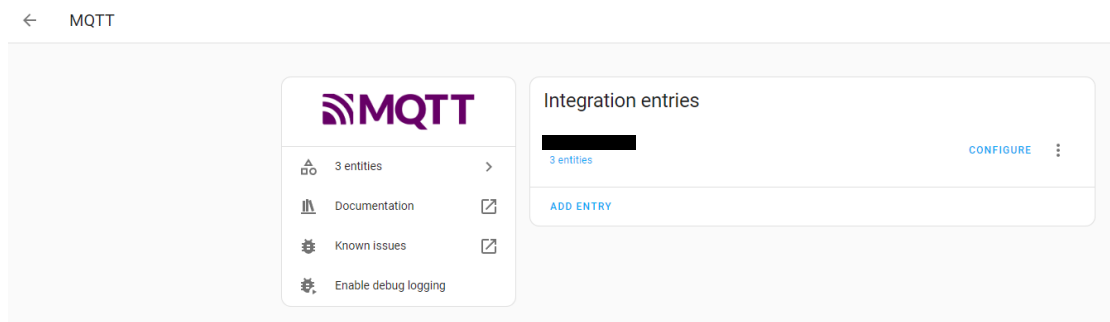
Configure your Broker settings. Please note that they must be consistent with the settings on your IP Power device.

(4-5) After confirming the MQTT configuration, please enter the file editor.

In the configuration.yaml file, which has completed the IP Power script, enter the MQTT script at the bottom.

* MQTT script: P.25

After saving, return to the main screen and restart to confirm whether the input is successful.

5.Home Assistant App WAN settings:

Home Assistant has its own dedicated app for device control, but the app normally only supports local network control.

If you want to control IP Power at home or your workplace using the Home Assistant app on your smartphone or tablet from outside, please follow the procedure below:

(5-1) Please copy your Home Assistant IP and login to your router web UI.
    Find Port forwarding -> Virtual Server Settings.

*Note: Each Routers have different company-own UI interfaces and operations, please search the brand company of your router for how to configure port forwarding.

After setting up port forwarding, locate your internet IP in your router and copy it. This will be used for configuring remote connection on your smartphone(roaming).

* Due to the fact that the network IP of the router is provided by the supplier, the IP will change approximately every 3 to 4 days.

If the IP changes midway, please remember to update it within Home Assistant. Before logging in with the mobile app, please check the network IP of your router to avoid login issues.

(5-2) Download the Home Assistant app on your smartphone or tablet.

(5-3) When opening the app, enter the copied router IP for login, and confirm if your app functions properly. After confirmation, turn on "Roaming" control. If the connection is successful, it means you can use the mobile app to control IP Power remotely.

EX: http://your router IP: 8123/

* If you are still within the local network, you can first copy the Home Assistant IP from your PC for login. Then, go to "Settings -> Mobile App" to set up a new server, and input the router's IP in the app settings.

* For your convenient, we recommended to create two users, one for use on the internal network and the other for the router IP (external network).

6. IP Power 98 series Script, for Home Assistant only:

This script is adapted based on the CGI commands in the IP POWER product manual. User can also modify this script according to their own needs, creating a control script that is more suitable for their preferences.

*Note:

(1) The script in red is for reference only, please modify the content to be written.

(2) This script is only for 98 series. For control 9255 and 9258, please move to P.23

*Modifiable parts include:

(1) Domain name or IP address, MAC address of the device

(2) User name and password for accessing the device

(3) Name and unique_id for each switch and sensor

(4) Scan_interval, the interval in seconds for retrieving the switch status, as indicated in red in the code. *Note: unique_id must not be duplicated.

(1) IP POWER 98 series script:

```
command_line:
    - switch:
        name: ippower98XX_RL1
        unique_id: 98XXswitch_RL1
        command_on: "/usr/bin/curl -X GET
http://192.168.31.146/set.cmd?cmd=setpower+p61=1 -u admin:12345678"
        command_off: "/usr/bin/curl -X GET
http://192.168.31.146/set.cmd?cmd=setpower+p61=0 -u admin:12345678"
        command_state: "/usr/bin/curl -X GET
http://192.168.31.146/json.cmd?getpower -u admin:12345678"
        value_template: '{{value_json.result.RL[0].state == 1}}'
```

```yaml
switch:
  - platform: rest
    name: ippower98XX_RL2
    unique_id: 98XXswitch_RL2
    resource: http://192.168.31.146/json.cmd?
    state_resource: http://192.168.31.146/json.cmd?getpower
    body_on: '{"cmd":"setpower", "RL":[{"id":2, "action":1}]}'
    body_off: '{"cmd":"setpower", "RL":[{"id":2, "action":0}]}'
    is_on_template: '{{value_json.result.RL[1].state == 1}}'
    username: 'admin'
    password: '12345678'
    device_class : 'outlet'


  - platform: rest
    name: ippower98XX_RL3
    unique_id: 98XXswitch_RL3
    resource: http://192.168.31.146/json.cmd?
    state_resource: http://192.168.31.146/json.cmd?getpower
    body_on: '{"cmd":"setpower", "RL":[{"id":3, "action":1}]}'
    body_off: '{"cmd":"setpower", "RL":[{"id":3, "action":0}]}'
    is_on_template: '{{value_json.result.RL[2].state == 1}}'
    username: 'admin'
    password: '12345678'
    device_class : 'outlet'

  - platform: rest
    name: ippower98XX_RL4
    unique_id: 98XXswitch_RL4
    resource: http://192.168.31.146/json.cmd?
    state_resource: http://192.168.31.146/json.cmd?getpower
    body_on: '{"cmd":"setpower", "RL":[{"id":4, "action":1}]}'
    body_off: '{"cmd":"setpower", "RL":[{"id":4, "action":0}]}'
    is_on_template: '{{value_json.result.RL[3].state == 1}}'
    username: 'admin'
    password: '12345678'
    device_class : 'outlet'
```

```yaml
rest:
  - authentication: basic
    username: "admin"
    password: "12345678"
    scan_interval: 60
    resource: http://192.168.31.146/json.cmd?getinfo
    sensor:
      - name: "98XX System Information"
        unique_id: 98XX_system_info
        json_attributes_path: "$.result.device"
        value_template: "OK"
        json_attributes:
          - "model"
          - "firmware_ver"
          - "bootloader_ver"
          - "dev_name"
          - "mac"
          - "num_relay"
          - "num_di"
          - "num_spdi"
          - "num_do"
          - "uptime"

      - name: "98XX WAN"
        unique_id: 98XX_wan
        json_attributes_path: "$.result.wan"
        value_template: "OK"
        json_attributes:
          - "wan_ip"
          - "gateway"
          - "primary_dns"
          - "secondary_dns"
          - "web_port"
          - "wan_port"

      - name: "98XX LAN"
        unique_id: 98XX_lan
        json_attributes_path: "$.result.lan"
```

```yaml
            value_template: "OK"
            json_attributes:
                - "eth_connection"
                - "wlan_connection"


        - name: "98XX Protocol"
            unique_id: 98XX_protocol
            json_attributes_path: "$.result.protocol"
            value_template: "OK"
            json_attributes:
                - "snmp"
                - "telnet"
                - "modbus_tcp"
                - "modbus_rtu"
                - "bacnet_ip"
                - "upnp"
                - "ipservice"
                - "mqtt"
  - authentication: basic
      username: "admin"
      password: "12345678"
      scan_interval: 60
      resource: http://192.168.31.146/json.cmd?getpower
      sensor:
          - name: "98XX Voltage"
              unique_id: 98XX_voltage
              unit_of_measurement: "V"
              value_template: "{{ value_json.result.global_measure.voltage }}"
              device_class: voltage


          - name: "98XX Temperature"
              unique_id: 98XX_temperature
              unit_of_measurement: "°C"
              value_template: "{{ value_json.result.global_measure.temperature |
round(1) }}"
              device_class: temperature
```

(2) IP POWER 9255, 9258 series script:

*Note: After integrating 9258 into Home Assistant, it may intermittently emit a "beep" sound. It is recommended to access the web interface of 9258 and navigate to System --> Setup to disable the Beeper. This will resolve the issue of 9258 emitting sounds at irregular intervals.

```
  - switch:
      name: ippower9258_RL1
      unique_id: 9258switch_RL1
      command_on: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p61=1 -u admin:12345678"
      command_off: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p61=0 -u admin:12345678"
      command_state: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=getpower -u admin:12345678"
      value_template: '{{ value | regex_findall_index("p61=(\\d+)", 0) | int == 1 }}'
      scan_interval: 10


  - switch:
      name: ippower9258_RL2
      unique_id: 9258switch_RL2
      command_on: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p62=1 -u admin:12345678"
      command_off: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p62=0 -u admin:12345678"
      command_state: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=getpower -u admin:12345678"
      value_template: '{{ value | regex_findall_index("p62=(\\d+)", 0) | int == 1 }}'
      scan_interval: 10
```

```yaml
  - switch:
      name: ippower9258_RL3
      unique_id: 9258switch_RL3
      command_on: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p63=1 -u admin:12345678"
      command_off: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p63=0 -u admin:12345678"
      command_state: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=getpower -u admin:12345678"
      value_template: '{{ value | regex_findall_index("p63=(\\d+)", 0) | int == 1 }}'
      scan_interval: 10


  - switch:
      name: ippower9258_RL4
      unique_id: 9258switch_RL4
      command_on: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p64=1 -u admin:12345678"
      command_off: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=setpower+p64=0 -u admin:12345678"
      command_state: "/usr/bin/curl -X GET
http://10.33.122.49/set.cmd?cmd=getpower -u admin:12345678"
      value_template: '{{ value | regex_findall_index("p64=(\\d+)", 0) | int == 1 }}'
      scan_interval: 10
```

(3) MQTT script:
Demonstration model: IP Power 9828-P

(3-1) Simulated Switches: mqtt getpower and mqtt getsysinfo
We have configured a simulated switch for 9828-P named mqtt getpower, which is always in the OFF state. Its function is to send commands to obtain device switch status information.

When entering the Home assistant control interface for the first time, you should click this switch to get the correct switch status, or you can click this switch anytime you want to confirm the latest status of the switch.
Another simulated switch is mqtt getsysinfo for 9828-P, which is used to retrieve other system information of the device.

To view this information, you need to use the sensor 9828-P mqtt information.

(3-2) MQTT Switch Configuration:
We have set up two real switch controls, namely 9828-P mqtt switch1 and 9828-P mqtt switch2. If you need to define more switches, you only need to modify the RL[?] array index in the configuration file. Please refer to the previous definition of JSON format commands; RL[0]-RL[7] represent the first to eighth switches respectively, where "RL":[{"id":1, "action":1}] represents the first switch with the action set to ON.

After modifying the MQTT YAML configuration information and reloading all YAML configurations, you may find that all switches are unavailable. This is because the MQTT inside the home assistant is not connected for the first time and cannot obtain the available status information of the device. If the switches are triggered through other control methods (such as web or HTTP CGI), then the status information will be updated. Otherwise, you will need to restart the entire Home Assistant.

Notice! When entering your IP Power MAC, please follow the example format. MAC1 itself does not have ";", and must capitalize the entire text.
EX: 00762800156C

```yaml
#-------------------9828-P mqtt switch--------------------------
mqtt:
  - switch:
      name: "9828-P mqtt Switch1"
      unique_id: 9828-P_mqtt_switch1
      state_topic: "mac/your device's mac/cmd_ack/id_ha"
      command_topic: "mac/your device's mac/cmd"
      availability_topic: "mac/your device's mac/status"
      availability_template: "{% if 'status=online' in value %} online {% elif 'status=offline' in value %} offline {% else %} null{% endif %}"
      payload_on: '{"tag":"on", "user":"admin", "pass":"12345678","appid":"id_ha", "cmd":"setpower", "RL":[{"id":1, "action":1}]}'
      payload_off: '{"tag":"off", "user":"admin", "pass":"12345678","appid":"id_ha", "cmd":"setpower", "RL":[{"id":1, "action":0}]}'
      state_on: "1"
      state_off: "0"
      value_template: "{% if value_json.tag == 'on' and value_json.result.message == 'success' %} 1 {% elif value_json.tag == 'off' and value_json.result.message == 'success' %} 0{% else %} 0{% endif %}"
      optimistic: false
      qos: 0
      retain: false
```

```yaml
- switch:
        name: "9828-P mqtt Switch2"
        unique_id: 9828-P_mqtt_switch2
        state_topic: "mac/your device's mac/cmd_ack/id_ha"
        command_topic: "mac/your device's mac/cmd"
        availability_topic: "mac/your device's mac/status"
        availability_template: "{% if 'status=online' in value %} online {% elif
'status=offline' in value %} offline {% else %} null{% endif %}"
        payload_on: '{"tag":"on", "user":"admin",
"pass":"12345678","appid":"id_ha", "cmd":"setpower", "RL":[{"id":2,
"action":1}]}'
        payload_off: '{"tag":"off", "user":"admin",
"pass":"12345678","appid":"id_ha", "cmd":"setpower", "RL":[{"id":2,
"action":0}]}'
        state_on: "1"
        state_off: "0"
        value_template: "{% if value_json.tag == 'on' and
value_json.result.message == 'success' %} 1 {% elif value_json.tag == 'off' and
value_json.result.message == 'success' %} 0{% else %} 0{% endif %}"
        optimistic: false
        qos: 0
        retain: false
```

```yaml
#------------------9828-P mqtt get power---------------------------
  - switch:
      name: "9828-P mqtt getpower"
      unique_id: 9828-P _mqtt_getpower
      state_topic: "mac/ your device's mac /cmd_ack/id_ha"
      command_topic: "mac/ your device's mac/cmd"
      availability_topic: "mac/ your device's mac /status"
      availability_template: "{% if 'status=online' in value %} online {% elif
'status=offline' in value %} offline {% else %} null{% endif %}"
      payload_on: '{"tag":"on", "user":"admin",
"pass":"12345678","appid":"id_ha", "cmd":"getpower"}'
      payload_off: '{"tag":"off", "user":"admin",
"pass":"12345678","appid":"id_ha", "cmd":"getpower"}'
      state_on: "1"
      state_off: "0"
      value_template: "{% if value_json.tag == 'on' and
value_json.result.message == 'success' %} 1 {% elif value_json.tag == 'off' and
value_json.result.message == 'success' %} 0{% else %} 0{% endif %}"
      optimistic: false
      qos: 0
      retain: false

#------------------9828-P mqtt sensor----------------------------
  - sensor:
      name: "9828-P mqtt information"
      unique_id: 9828-P_mqtt_information
      state_topic: "mac/your device's mac/cmd_ack/id_ha"
      value_template: "{{ value_json.result.global_measure.temperature |
round(1) }}"
      unit_of_measurement: "°C"
      availability_topic: "mac/ your device's mac /status"
      availability_template: "{% if 'status=online' in value %} online {% elif
'status=offline' in value %} offline {% else %} null{% endif %}"
      json_attributes_topic: "mac/ your device's mac /cmd_ack/id_ha"
```

Disclaimer:

The content of this manual is provided for reference and general informational purposes only for IP Power users, and may be subject to change without prior notice. We make no express or implied warranties or representations regarding the accuracy, completeness, and reliability of the information on this manual related to IP Power applications with Home Assistant.

Any information obtained from this manual is at the user's own risk. We disclaim any and all liability for any direct or indirect losses or damages, including but not limited to loss of profits, business interruption, data loss, or other financial losses, even if we have been advised of the possibility of such damages.

This manual may contain links or references to third-party websites. We assume no responsibility for the content or availability of these third-party websites and do not endorse, guarantee, or make any representations about them. You should assess and assume the risks of using these third-party websites on your own.

We reserve the right to change or modify this disclaimer at any time. It is recommended that you regularly check this page for any updates. By continuing to use this manual, you agree to comply with these terms and conditions.

If you have any questions regarding this disclaimer, please contact us.