

AI-01790 Home Assistant Turn On Tapo P100

👤 Owner	Ⓜ Rigel Arcayan
☰ Tags	Problem
🕒 Created time	@July 1, 2025 7:42 AM

In home assistant, create an automation to turn on the Tapo P100 for 30 seconds and then turn it off.

Here's how to create an automation in Home Assistant to turn on a Tapo P100 smart plug for 30 seconds and then turn it off. This can be done through the UI. Assumptions:

1. You have Home Assistant installed and running.
2. You have the Tapo integration set up and your P100 plug is already discovered and accessible as an entity in Home Assistant. (If not, you'll need to add it via Settings > Devices & Services > Add Integration).
3. You know the entity ID of your Tapo P100 (e.g., switch.tapo_p100_your_device_name).

Steps to Create the Automation in Home Assistant UI:

1. Go to Settings: In Home Assistant, navigate to Settings (the gear icon in the sidebar).
2. Select Automations & Scenes: Click on Automations & Scenes.
3. Create New Automation: Click the "Create Automation" button (usually a blue plus icon or a similar button).
4. Start with an Empty Automation: Choose "Start with an empty automation".

Configuring the Automation:

5. Name (Optional but Recommended):
6. Give your automation a descriptive name, like "Tapo P100 30-Second On/Off".
7. Triggers:
8. Choose your desired trigger. This is what will start the automation. Here are a few common examples:
 - Manual Trigger (for testing):
 - Type: Trigger
 - Choose Manually run (this creates a button in the automation editor to trigger it).
 - Time-based Trigger (e.g., every day at a specific time):
 - Type: Time
 - Set the specific At time.
 - Device State Change (e.g., when another device turns on):
 - Type: State
 - Choose your Entity (the device whose state change will trigger this).
 - Set From and To states.
 - Button Press (if your Tapo plug has a physical button that can be configured in HA, or another smart button):
 - Type: Device
 - Select the Device and then the specific Action (e.g., "button pressed"). For this example, let's use a "Manually run" trigger for simplicity, allowing you to test it easily.
9. Actions:
10. Click on "Add Action".
 - Action 1: Turn on the Tapo P100
 - Type: Call service
 - Service: switch.turn_on

- Target:
 - Entity: Select your Tapo P100 entity (e.g., switch.tapo_p100_your_device_name).

11. Click on "Add Action" again.

- Action 2: Wait for 30 seconds
 - Type: Wait for time
 - Wait time: Enter 00:00:30 (Hours:Minutes:Seconds)

12. Click on "Add Action" again.

- Action 3: Turn off the Tapo P100
 - Type: Call service
 - Service: switch.turn_off
 - Target:
 - Entity: Select your Tapo P100 entity (the same one as before).

13. Save the Automation:

14. Click the "Save" button (usually in the bottom right corner).

Review of the Automation Logic:

15. Trigger: (Whatever you set, e.g., manual trigger, specific time, etc.)

16. Action 1: The switch.turn_on service is called for your Tapo P100.

17. Action 2: The automation pauses for 30 seconds.

18. Action 3: After the 30-second pause, the switch.turn_off service is called for your Tapo P100.

This sequence will ensure your Tapo P100 turns on, stays on for 30 seconds, and then turns off automatically.

Can you provide the YAML method instead

Certainly! Here's the YAML method for creating the Home Assistant automation to turn on your Tapo P100 for 30 seconds and then turn it off.

You can add this directly to your automations.yaml file, or paste it into the "YAML" tab when creating a new automation in the Home Assistant UI.

Important: Replace `switch.your_tapo_p100_entity_id` with the actual entity ID of your Tapo P100 plug. You can find this in Home Assistant under Settings → Devices & Services → Entities and search for your Tapo plug.

```
- id: '1678901234567' # A unique ID. Generate a new one if you have other automations.
  alias: Tapo P100 30-Second On/Off Cycle
  description: Turns on Tapo P100 for 30 seconds, then turns it off.
  trigger:
    - platform: manual # This allows you to manually trigger the automation from the UI.
      # You can change this to a time, state, or other trigger as needed.
  action:
    - service: switch.turn_on
      target:
        entity_id: switch.your_tapo_p100_entity_id # <<< REPLACE THIS with your actual entity ID
    - delay:
        seconds: 30
    - service: switch.turn_off
      target:
        entity_id: switch.your_tapo_p100_entity_id # <<< REPLACE THIS with your actual entity ID
  mode: single # Ensures that only one instance of the automation runs at a time.
```

How to Use This YAML:

Method 1: Directly in `automations.yaml` (Advanced Users)

- Access your Home Assistant configuration files. This is often done via SSH, Samba Share, or the File Editor add-on.
- Open your `automations.yaml` file.
- Paste the YAML code above at the end of the file.

- Important: After saving, go to Settings → System → Developer Tools → YAML tab, and click CHECK CONFIGURATION to ensure there are no errors.
- Then click RELOAD AUTOMATIONS to make Home Assistant aware of the new automation.

Method 2: Using the Home Assistant UI (Recommended)

- In Home Assistant, go to Settings (the gear icon in the sidebar).
- Click on Automations & Scenes.
- Click the "Create Automation" button.
- Instead of "Start with an empty automation," click the three dots in the top right corner and select "Edit in YAML".
- Delete any existing default YAML code and paste the code provided above.
- Crucially, replace `switch.your_tapo_p100_entity_id` with your actual entity ID.
- Click "Save" (the floppy disk icon in the bottom right). Home Assistant will automatically assign an id if you omit it, or use the one you provide.

- **Explanation of the YAML:**

- `id`: A unique identifier for your automation. Home Assistant generates this automatically if you create it via the UI without specifying it. If you add it manually, ensure it's unique across all your automations.
- `alias`: The friendly name of your automation, which will appear in the UI.
- `description`: An optional, longer description of what the automation does.
- `trigger`: Defines when the automation should run.
 - `platform: manual`: This is set up so you can easily test the automation by clicking "Run" on the automation's card in the Home Assistant UI.
 - Alternative Triggers (examples):

```
# Time-based trigger (e.g., every day at 10:00 PM)
# trigger:
#   - platform: time
```

```
# at: "22:00:00"
```

```
# State-based trigger (e.g., when a motion sensor turns on)
```

```
# trigger:
```

```
# - platform: state
```

```
# entity_id: binary_sensor.motion_sensor_front_door
```

```
# to: "on"
```

- action: The sequence of events that happen when the automation is triggered.
 - service: switch.turn_on: Calls the service to turn on a switch entity.
 - target: entity_id: switch.your_tapo_p100_entity_id: Specifies which entity the service should act upon.
 - delay: seconds: 30: Pauses the automation for 30 seconds before proceeding to the next action.
 - service: switch.turn_off: Calls the service to turn off a switch entity.
- mode: single: This is the default and recommended mode. It means if the automation is triggered while it's already running, the new trigger will be ignored until the current run finishes. Other modes include restart, queued, and parallel for more complex scenarios.