# SMART TARGET

Rev 1.x

USER MANUAL

**Smart Target System**
*Model: SmartTargetV1.0*

## PRODUCT OVERVIEW

The Smart Target is a modular WiFi enabled target for shooting galleries, interactive attractions, and advanced automation solutions. The smart target is unique in that it is an all-in-one solution that has built in computer power, eye catching visuals, and electronics to control props without additional equipment.

Each target supports both standalone functionality and seamless integration with Home Assistant (an open-source platform for automating smart devices and controlling home systems from a single interface). This enables integration with a broad array of home automation products, supporting scalable expansion, streamlined system development, and the implementation of advanced automation scenarios previously not possible in shooting galleries.

## APPLICATIONS

- Shooting galleries and amusement booths
- Servo-actuated reveals for haunted houses
- Reactive gaming components or automation props
- General home automation, such as lighting or IR device control

## FEATURES

- LED control for visual feedback
- RS232 port for serial communication
- PWM outputs for external devices or servos
- GPIO inputs for external sensors or buttons
- Analog inputs for variable signal sensing
- Power passthrough capability
- Flexible power input options

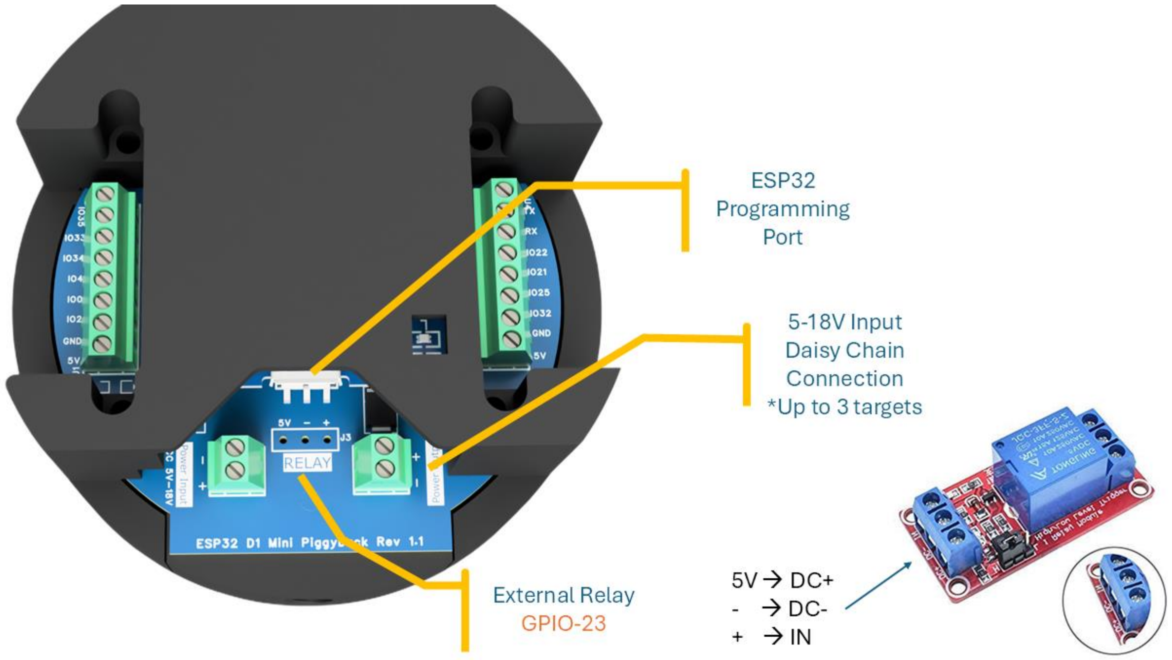
## INTEGRATION & CONNECTIVITY

- ESP32 platform running ESPHome
- Wi-Fi connectivity for network integration
- Compatible with Home Assistant
- Over-the-air (OTA) firmware updates
- Infrared (IR) control support
- RS232 communication

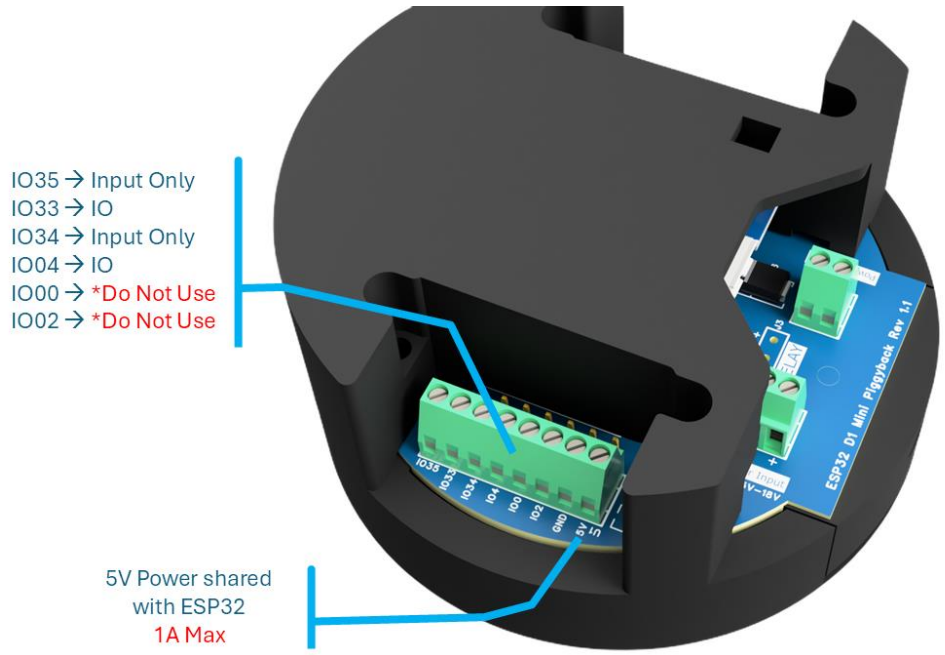
## Electrical & Mechanical Specifications

| **Item** | **Specification** |
|---|---|
| Power Input | 5-18V DC |
| WS2812 Output | 1x data-only pin |
| PWM Control | 3x PWM output (50Hz, 3.3V logic) |
| Analog Input | 2x 0-3.3V ADC |
| GPIO Inputs | 5x digital inputs |
| IR Receiver | NEC protocol |
| RS232 | TX/RX headers |

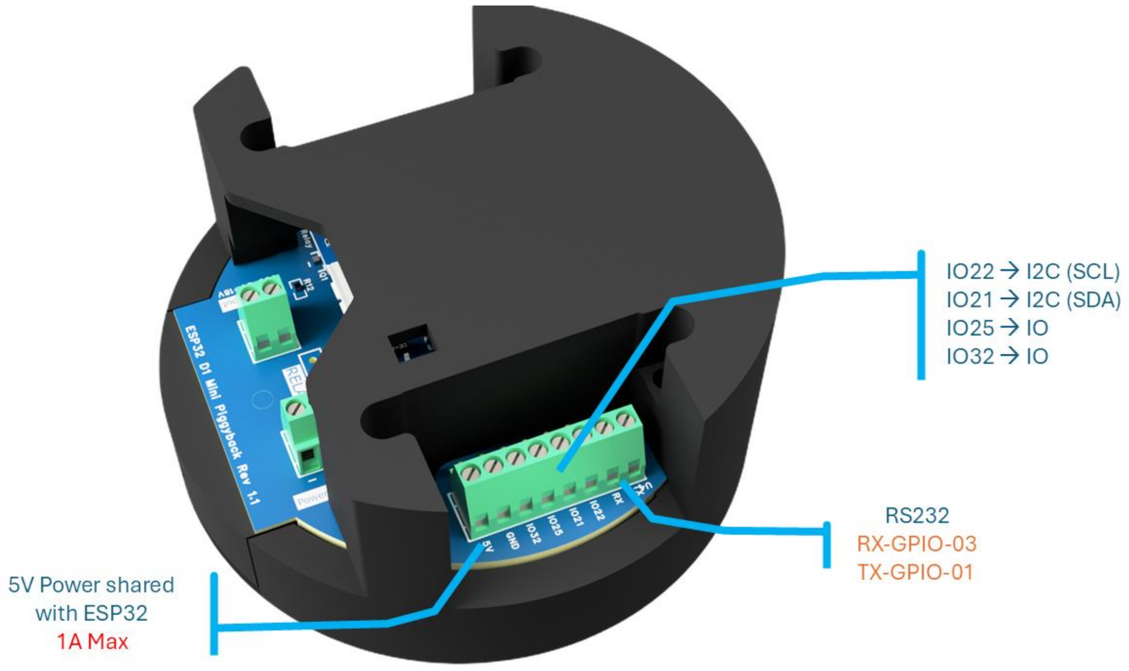
## Inputs and outputs



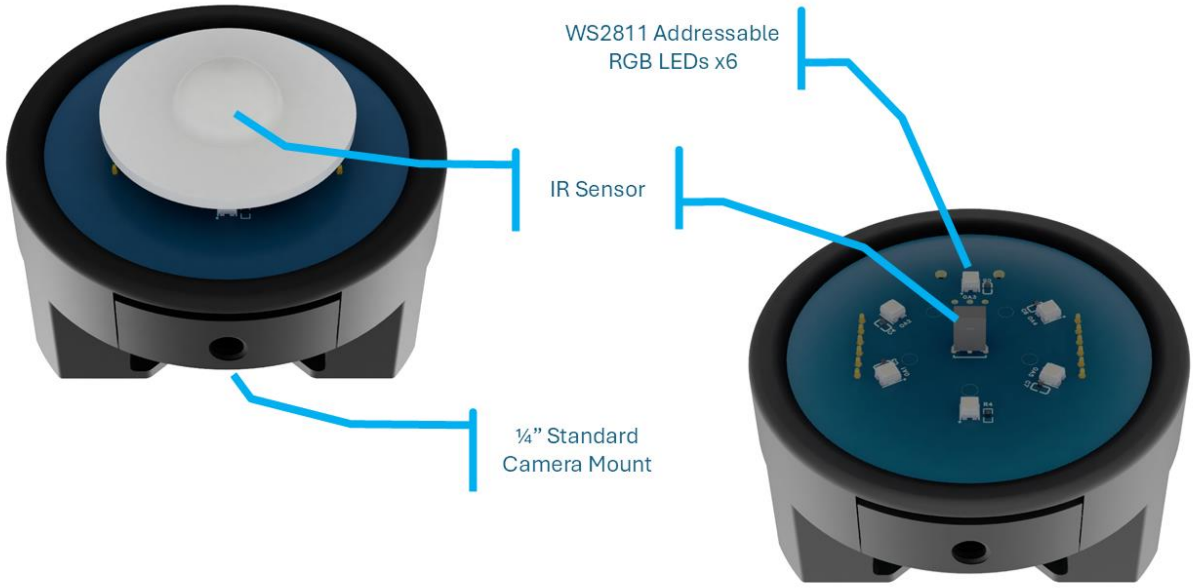
**Smart Target System**
*Model: SmartTargetV1.0*



* Pinouts used for bootloader

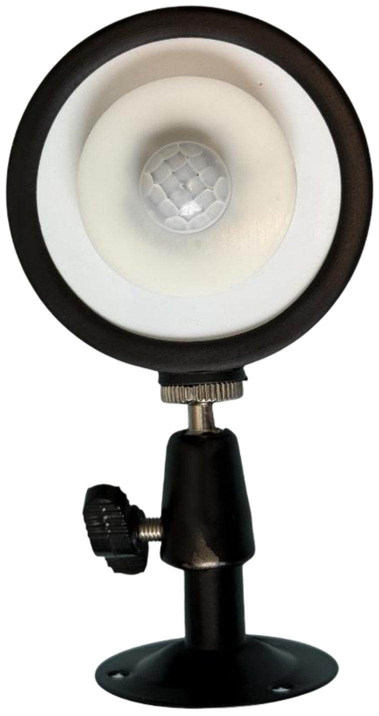


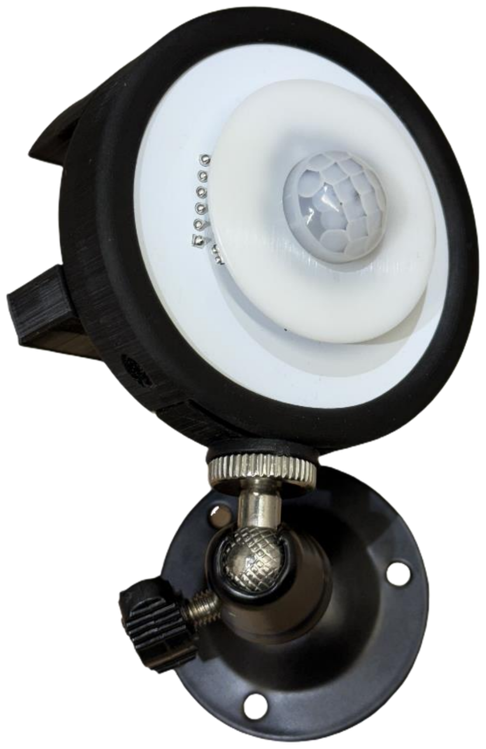
**Smart Target System**
*Model: SmartTargetV1.0*



## Target Mounting

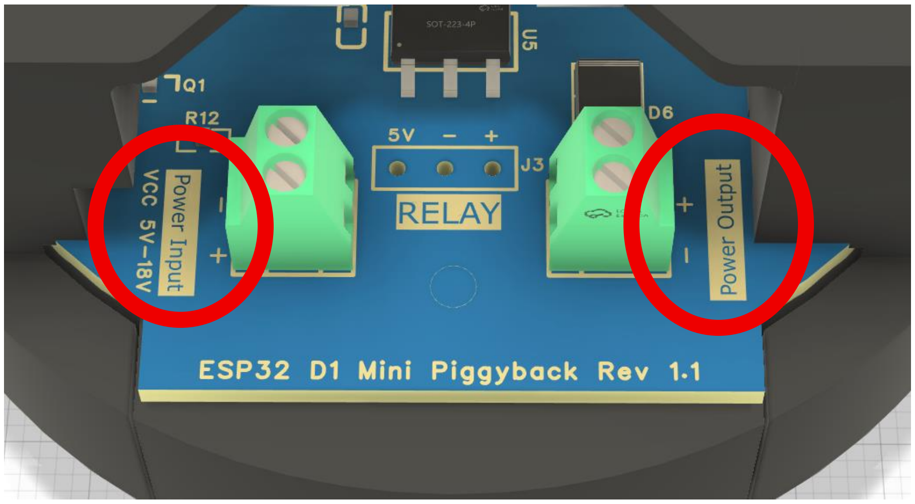
The target includes a standard camera mount hole on the bottom and a CCTV swivel mount, enabling versatile mounting in any direction, on walls, or even upside down.





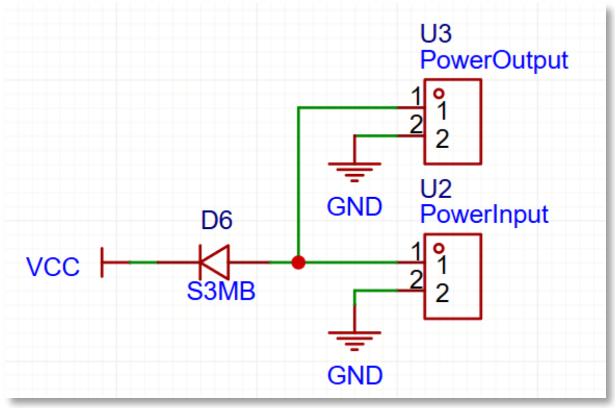
## Power Input

To power the target, connect 12V@5A power supply to either screw terminal circled below.



*The power input **does** have reverse connection protection.

A 12V input is recommended; however, any voltage within the 5-18V range is acceptable. Dual power terminals are provided to allow for the connection of multiple targets without requiring separate power supplies, enabling sequential linking of devices. Due to considerations regarding cable length and overall power consumption, it is advisable not to connect more than five targets to a single power source. For long-distance connections, shielded cables should be employed to ensure signal integrity.



*Schematic 1: Power Connector*

# Standalone Target

## Default Functionality

The targets are preconfigured for standalone operation and do not feature an ‘on’ switch. They automatically power on when connected to a power source. In standalone mode, the targets operate independently and do not require a local server or Wi-Fi access point. This configuration is ideal for single-player shooting gallery setups.

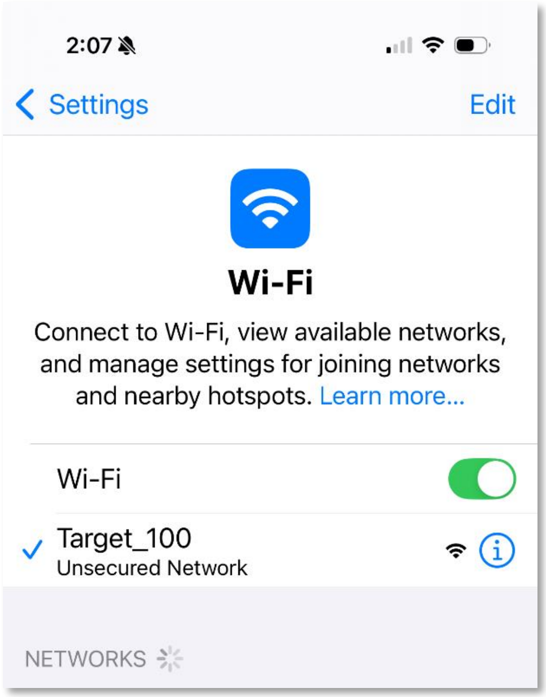
Upon startup, the target’s LED display will illuminate white and oscillate, indicating that the unit is operational. This default idle state can be customized as needed (refer to the custom settings and programming sections).

When activated by an IR transmitter utilizing the NEC protocol, the internal relay will engage for three seconds. During this period, the LEDs will display a colorful pinwheel effect to signal that the target has been hit.

## Wi-Fi Connection

In standalone mode, each target operates as an access point that can be connected to from your phone or computer as a typical wireless network.

The default SSID each target broadcasts on is “Target_####”. Where the last four numbers are random. The default password for the target is ‘target123’.



2:07

Settings — Edit

**Wi-Fi**

Connect to Wi-Fi, view available networks, and manage settings for joining networks and nearby hotspots. Learn more...

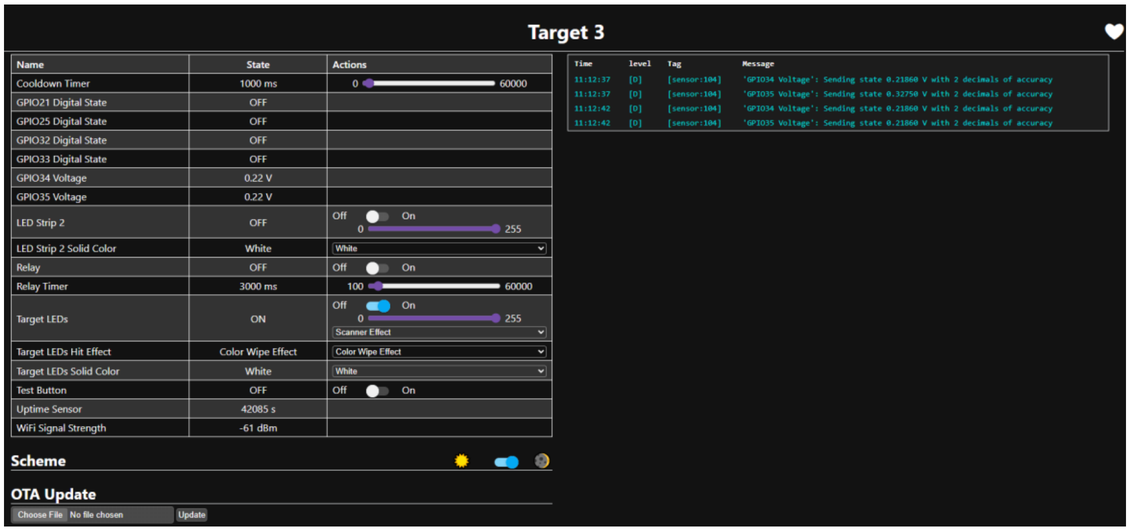
| | |
|---|---|
| Wi-Fi | On |
| ✓ Target_100<br>Unsecured Network | |

NETWORKS

*Figure 1: SSID Access Point Example*

Once connected to the target's Wi-Fi, open a web browser and go one of the following addresses:

- **target-###.local**
- **192.168.4.1**



**Target 3**

| Name | State | Actions |
|---|---|---|
| Cooldown Timer | 1000 ms | 0 — 60000 |
| GPIO21 Digital State | OFF | |
| GPIO25 Digital State | OFF | |
| GPIO32 Digital State | OFF | |
| GPIO33 Digital State | OFF | |
| GPIO34 Voltage | 0.22 V | |
| GPIO35 Voltage | 0.22 V | |
| LED Strip 2 | OFF | Off [✓ Off] On; 0 — 255 |
| LED Strip 2 Solid Color | White | White |
| Relay | OFF | Off [✓ Off] On |
| Relay Timer | 3000 ms | 100 — 60000 |
| Target LEDs | ON | Off [✓ On]; 0 — 255; Scanner Effect |
| Target LEDs Hit Effect | Color Wipe Effect | Color Wipe Effect |
| Target LEDs Solid Color | White | White |
| Test Button | OFF | Off [✓ Off] On |
| Uptime Sensor | 42085 s | |
| WiFi Signal Strength | -61 dBm | |

```
Time      level  Tag            Message
11:12:37  [D]    [sensor:104]   'GPIO34 Voltage': Sending state 0.21860 V with 2 decimals of accuracy
11:12:37  [D]    [sensor:104]   'GPIO35 Voltage': Sending state 0.32750 V with 2 decimals of accuracy
11:12:42  [D]    [sensor:104]   'GPIO34 Voltage': Sending state 0.21860 V with 2 decimals of accuracy
11:12:42  [D]    [sensor:104]   'GPIO35 Voltage': Sending state 0.21860 V with 2 decimals of accuracy
```

*Figure 2: Target web page example*

The left side of the screen will have the user controls. On the right side of the screen are log messages being sent by the target.

The target web page displays the following information about the target:

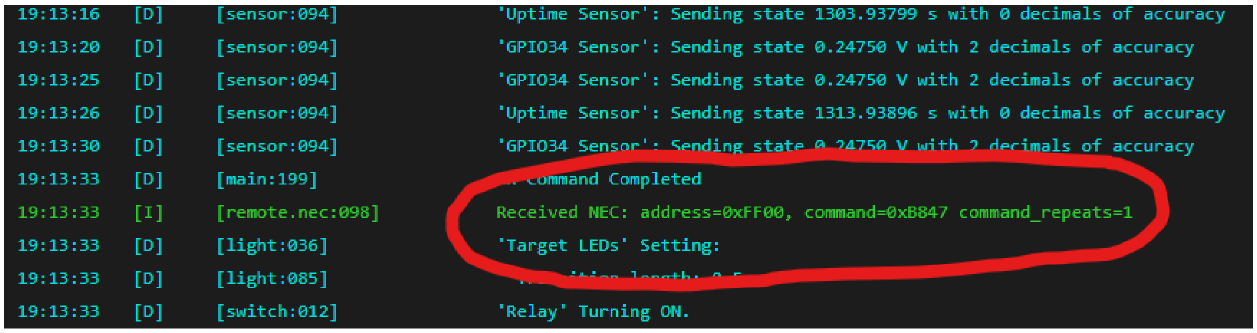
| Item | Description | States | GUI Control |
|---|---|---|---|
| **Cooldown timer** | The amount of time the target will be inactive after the being hit. This allows for props to reset to home position if needed. Value is saved through power cycles. | 0ms – 60,000ms<br>(1 minute max) | Yes |
| **GPIO 21, 25, 32, 33** | Indicates the digital input state at this GPIO terminal | ON or OFF | No |
| **GPIO 34, 35** | Indicates the analog input at this GPIO terminal | 0.0-3.3V | No |
| **LED Strip 2**<br>**(GPIO 22)** | WS2811 and WS2812 control of external addressable LED strip. | ON or OFF<br>Brightness<br>Effect | Yes |
| **LED Strip 2 Solid Color**<br>**(GPIO 22)** | Selectable solid colors for when the effect is 'solid', 'scanner', or 'twinkle' | Dropdown selectable | Yes |
| **Relay** | Internal mechanical relay | On or Off | Yes |
| **Relay Timer** | When target is hit, this is how long the relay stays closed and hit effect is displayed. Value is saved through power cycles. | 100ms – 60,000ms<br>(.1 second min)<br>(1 minute max) | Yes |
| **Target LEDs** | Target face addressable LED control. Effect runs at idle and should return to selected effect after target hit pattern. | ON or OFF<br>Brightness<br>Effect | Yes |

| | | | |
|---|---|---|---|
| **Target LEDs Hit Effect** | Selectable effect for when the target is hit. | Dropdown selectable | Yes |
| **Target LEDs Solid Color** | Selectable solid colors for when the target is idle and the effect is ‘solid’, ‘scanner’, or ‘twinkle’ | Dropdown selectable | Yes |
| **Test Button** | Target hit stimulation. Switch will turn back to off when target hit sequence is complete. | ON or OFF | Yes |
| **Uptime Sensor** | Time in seconds the target has been on. | Seconds | No |
| **WiFi Signal Strength** | Only seen when in networkable mode. Displays current dBm. | dBm range<br><br>Low -60’s is good | No |

## IR Receiver

The target uses a common HS0038B IR receiver. This infrared (IR) receiver module is designed for remote control applications, operating at a standard frequency of 38kHz. It features high sensitivity to IR signals and excellent noise suppression, making it reliable in environments with ambient light interference.

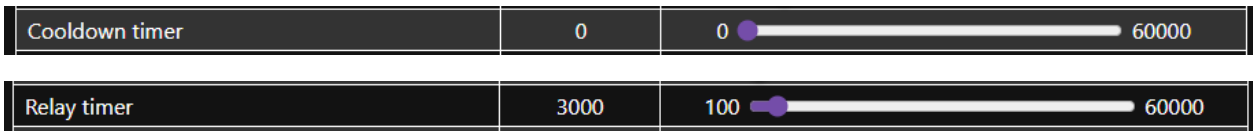
The default application will trigger on any NEC remote control protocol, however this signal can be filtered to only trigger on specific signals. When triggered, the selected hit effect will display for the time set on the relay timer slider. In addition, the remote-control code will be displayed in the log window.



```
19:13:16   [D]   [sensor:094]         'Uptime Sensor': Sending state 1303.93799 s with 0 decimals of accuracy
19:13:20   [D]   [sensor:094]         'GPIO34 Sensor': Sending state 0.24750 V with 2 decimals of accuracy
19:13:25   [D]   [sensor:094]         'GPIO34 Sensor': Sending state 0.24750 V with 2 decimals of accuracy
19:13:26   [D]   [sensor:094]         'Uptime Sensor': Sending state 1313.93896 s with 0 decimals of accuracy
19:13:30   [D]   [sensor:094]         'GPIO34 Sensor': Sending state 0.24750 V with 2 decimals of accuracy
19:13:33   [D]   [main:199]           [illegible] Command Completed
19:13:33   [I]   [remote.nec:098]     Received NEC: address=0xFF00, command=0xB847 command_repeats=1
19:13:33   [D]   [light:036]          'Target LEDs' Setting:
19:13:33   [D]   [light:085]            [illegible]ition length: [illegible]
19:13:33   [D]   [switch:012]         'Relay' Turning ON.
```

This address and command will be important if multiple players are hitting targets for score keeping or filtering to trigger on specific commands.

After the target is hit, the IR receiver is disabled until the relay timeout and cooldown timers have expired. This means a player cannot trigger (hit) the same target again until these timers are done.


Cooldown timer 0 0 60000
Relay timer 3000 100 60000

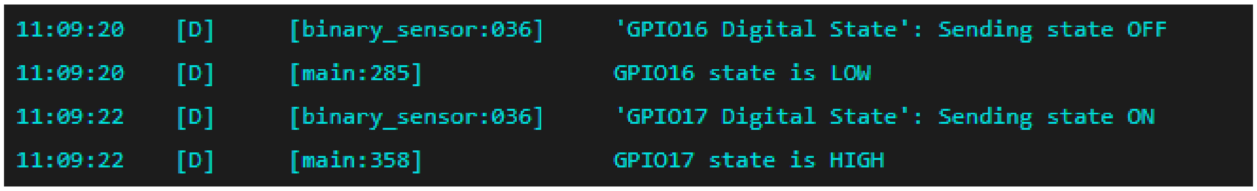
By default, the IR receiver is disabled for 3 seconds after each hit.

## GPIO Input

There are three general purpose input output (GPIO) pins that connect directly to the ESP32. The default standalone application does not use these inputs; the target will only indicate their state. The default state for GPIO 21,25, 32 and 33 is OFF. However, each has an internal pullup and the input is inverted. This means when nothing is connected, the ESP32 will have a high signal (due to the internal pullup resistor) and report OFF. By grounding the inputs, the ESP32 will read a low signal and report ON.

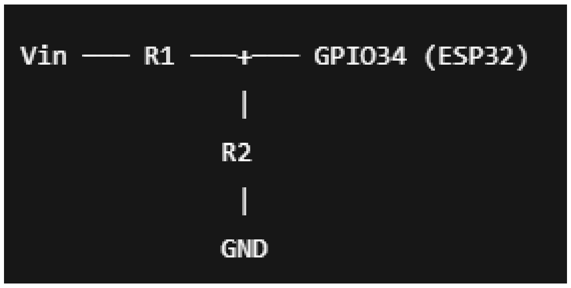
The purpose of these GPIO is to connect a switch to ground. An example of a switch would be a pressure switch that is grounded when stepped on.

When looking at the log window, the report of the digital input will be reported when it changes.


```
11:09:20    [D]     [binary_sensor:036]     'GPIO16 Digital State': Sending state OFF
11:09:20    [D]     [main:285]              GPIO16 state is LOW
11:09:22    [D]     [binary_sensor:036]     'GPIO17 Digital State': Sending state ON
11:09:22    [D]     [main:358]              GPIO17 state is HIGH
```

*WARNING: because there is a direct connection between the screw terminal and the ESP32, applying more than 3.3V as input could damage the microcontroller. This is a known issue with Rev 1.x target.

There is two analog inputs GPIO 34 and 35 that can be used for various voltage readings such as pressure. To use this input without damage to the ESP32 you should use a voltage divider based on the voltage your sensor requires.



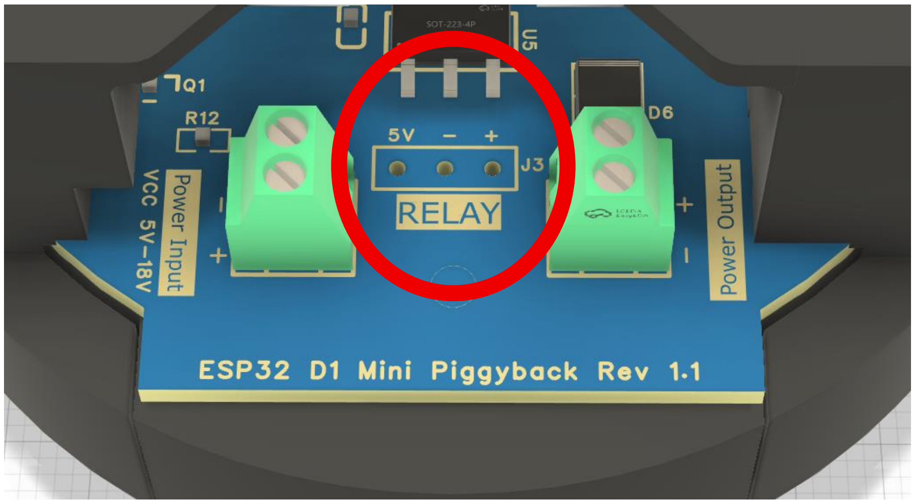
```
Vin ── R1 ──┼── GPIO34 (ESP32)
            |
            R2
            |
            GND
```

| Sensor Voltage | R1 | R2 | Max Reading |
|---|---|---|---|
| **12V** | 15kΩ | 5.6kΩ | 3.27V |
| **10V** | 6.8kΩ | 3.3kΩ | 3.3V |
| **5V** | 3.3KΩ | 2.2kΩ | 3.0V |

*WARNING: because there is a directly connection between the screw terminal and the ESP32, applying more than 3.3V as input could damage the microcontroller. This is a known issue with Rev 3.2 target.

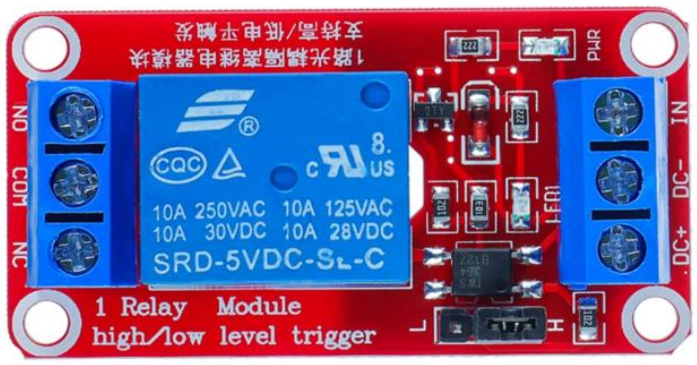
## Relay

The Rev 1.x target is design for external relays. This is not an internal relay but a connection to trigger a relay module.



Specifically, this connection is designed to connect to the following relay modules.

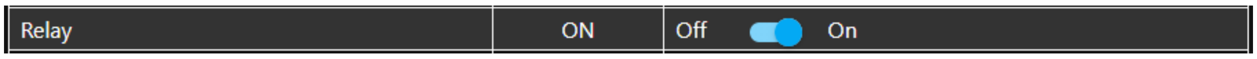
*Note: The relay must be set to trigger on LOW. Set the jumper to ‘L’.



Connection table:

| Relay Module | Target 1.x Relay Connection |
|---|---|
| DC+ | 5V |
| DC- | - |
| IN | + |

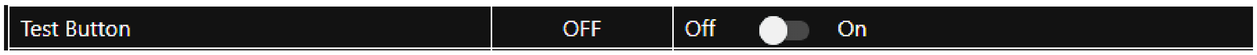
To test, turn the ‘Relay’ on.



| | | |
|---|---|---|
| Relay | ON | Off ● On |

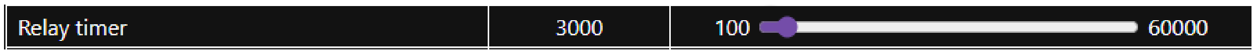
When the relay is ON, you should see the relay module LED illuminated indicating the relay is closed (COM connected to NO).

To test the relay ON time, select the ‘Test Button’. This will stimulate the target as if it was hit by an IR transmitter.



| | | |
|---|---|---|
| Test Button | OFF | Off ● On |

When using the ‘Test Button’ or when an IR signal is received, the relay will stay on for the amount of time indicated by the ‘Relay Timer’ slider.



| | | |
|---|---|---|
| Relay timer | 3000 | 100 ● 60000 |

The default for the relay on time is 3 seconds with the minimum being 0.1 seconds and the maximum being 1 minute or 60,000 milliseconds. The slider value will be saved through power cycles and is used to set up prop motion or external triggers.

If more current is required than the internal relay can support, an external more capable relay can be used instead. There are several ways to do this, one method is described in the Example Programs.

## Target Face LEDs

The target face LEDs are six WS2812 addressable RGB LEDs. Because they are addressable, each one can be controlled individually.

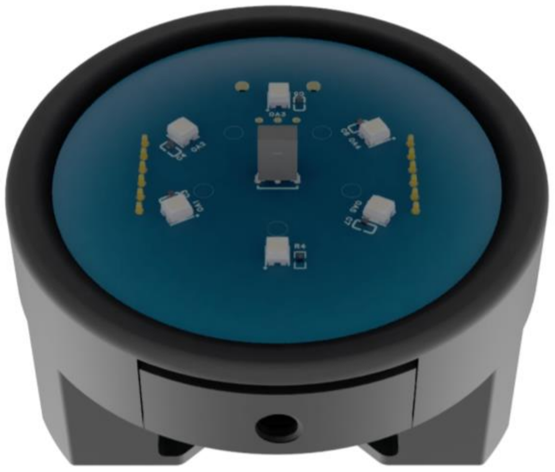
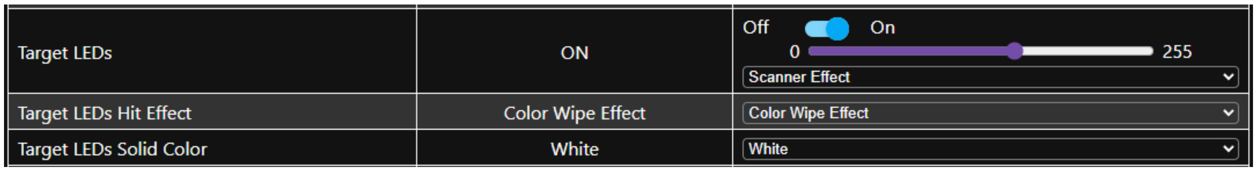


*Figure 3: Target RGB LEDs*

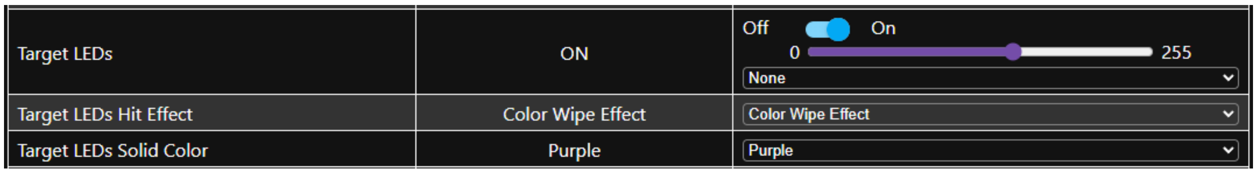
Because each LED can be controlled individually, we can achieve highly customized effects for the targets.



| | | |
|---|---|---|
| Target LEDs | ON | Off On<br>0 255<br>Scanner Effect |
| Target LEDs Hit Effect | Color Wipe Effect | Color Wipe Effect |
| Target LEDs Solid Color | White | White |

There are two configurable effects, one when the target is in idle (waiting for target hit) and the other when the target has been hit. The default idle effect is ‘scanner effect’ and the default hit effect is 'color wipe’.

When a solid color is preferred for either hit or idle, the color can be chosen by the solid color dropdown. An example is shown below.



| | | |
|---|---|---|
| Target LEDs | ON | Off ● On<br>0 — 255<br>None |
| Target LEDs Hit Effect | Color Wipe Effect | Color Wipe Effect |
| Target LEDs Solid Color | Purple | Purple |

*Note: Only the solid\none, twinkle, and scanner effects have selectable colors. Rainbow and color wipe have their own colors.

## LED Strip 2

The target can control 2 separate LED strips. The first however is the target face LEDs and the second is the terminal connection labeld 'LEDS'.

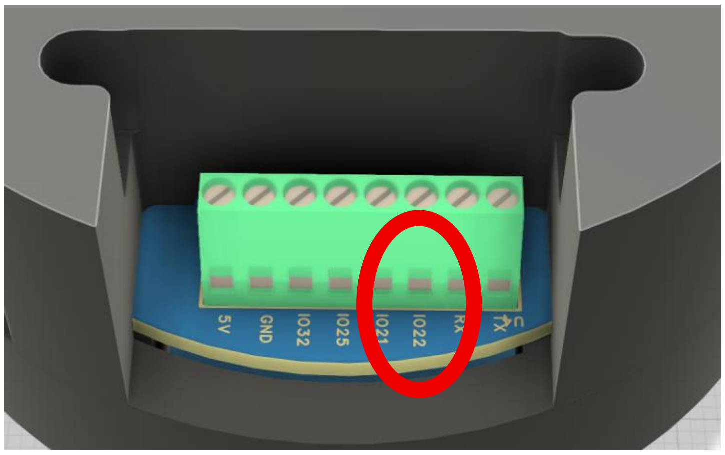
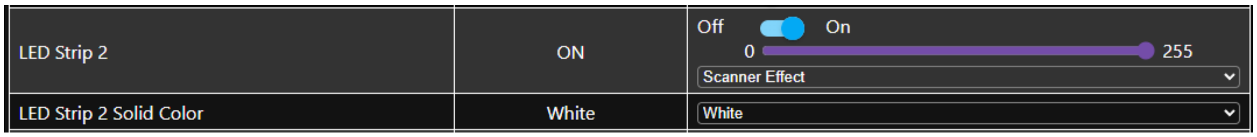


*Figure 4: LED Strip 2 (IO22)*



| | | |
|---|---|---|
| LED Strip 2 | ON | Off ● On<br>0 — 255<br>Scanner Effect |
| LED Strip 2 Solid Color | White | White |

When a solid color is preferred, the color can be chosen by the solid color dropdown for LED Strip 2. The default number of LEDs on a strip is 50 and can be changed in the configuration file (See custom programming section).

*Note: Only the solid\none, twinkle, and scanner effects have selectable colors. Rainbow and color wipe have their own colors.

## Customize Standalone

This section is if you want to customize how the standalone target works.

Updating the default standalone software to do exactly what you want is “easy”. I say this in quotes because you don’t need to know how to program in any language and you don’t need a special development environment application although I suggest using Visual Studio Code. It is a free download and works on Mac and PC.

Most of the application changes can be accomplished using ChatGPT for the solution. It is even easier if you use Copilot plugin for Visual Studio Code and it will edit the file contents for you.

ESPHome is the software running on the ESP32, the instructions to install the command line ESPHome can be found here: https://esphome.io/guides/installing_esphome

You will need the ESPHome command line to compile and flash the target from your computer.

After ESPHome is installed, download the smart target repository located here: https://github.com/CodeMakesItGo/SmartTarget

If you need to install Visual Studio Code, you can download it directly from the official website: https://code.visualstudio.com/download

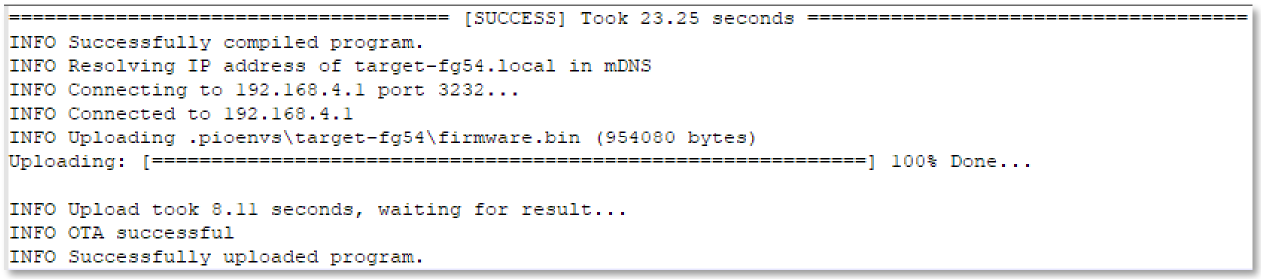
Next, connect your computer to the target by USB or connect to the target’s access point SSID if it is broadcasting.

The command below will compile and try to flash the target with the updates.

```
esphome -s id #### run .\target_rev1_x.yaml
```

Is the command above, the #### is any value you’d like for the target ID. The SSID will change to this value. However, it is suggested to re-use the target’s ID if it is known.

After running this command, you should see something similar to the output below indicating the programming was successful:



```
==================================== [SUCCESS] Took 23.25 seconds ====================================
INFO Successfully compiled program.
INFO Resolving IP address of target-fg54.local in mDNS
INFO Connecting to 192.168.4.1 port 3232...
INFO Connected to 192.168.4.1
INFO Uploading .pioenvs\target-fg54\firmware.bin (954080 bytes)
Uploading: [============================================================] 100% Done...

INFO Upload took 8.11 seconds, waiting for result...
INFO OTA successful
INFO Successfully uploaded program.
```

*Figure 5: Successful Programming*

## Networked Target

Each target may be configured to connect to a server, enabling enhanced control, feedback, automation, score tracking, and additional features. Networking the targets requires a server and a Wi-Fi router. Additionally, each target must be updated with the network-enabled version of the application. An overview of the network configuration is provided below.

Converting standalone targets to networked targets requires several steps. It is important to follow the setup instructions provided by Home Assistant closely. After a few hours, the first target can be added to the network, which may simplify subsequent configurations.

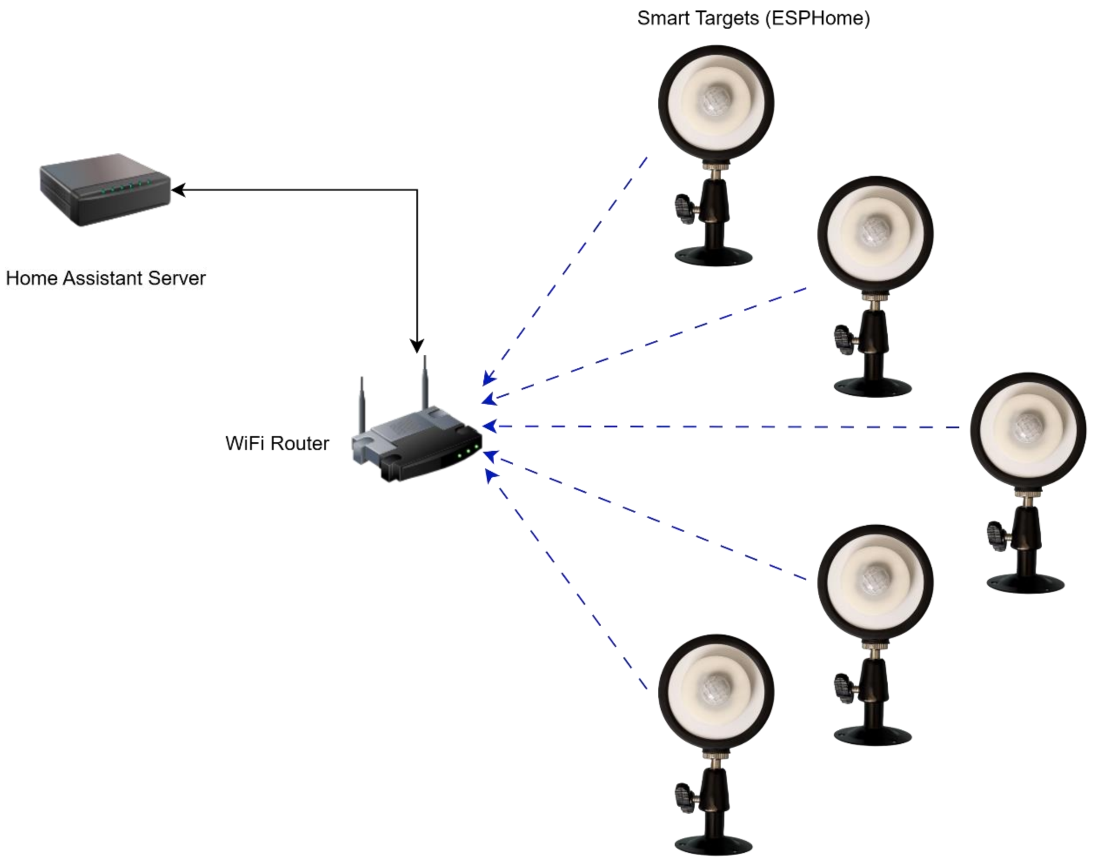


*Figure 6: Smart Target Basic Network*

The network architecture can be highly sophisticated, supporting a wide array of devices and enabling automated interactions among them. For instance, activating a single target may initiate a series of actions such as dimming lights, operating smoke machines, and playing audio cues. This illustrates how the system can be expanded to create a more interactive shooting gallery experience. Furthermore, the network enables remote operators to monitor the shooting gallery and reprogram targets as needed from an off-site location.

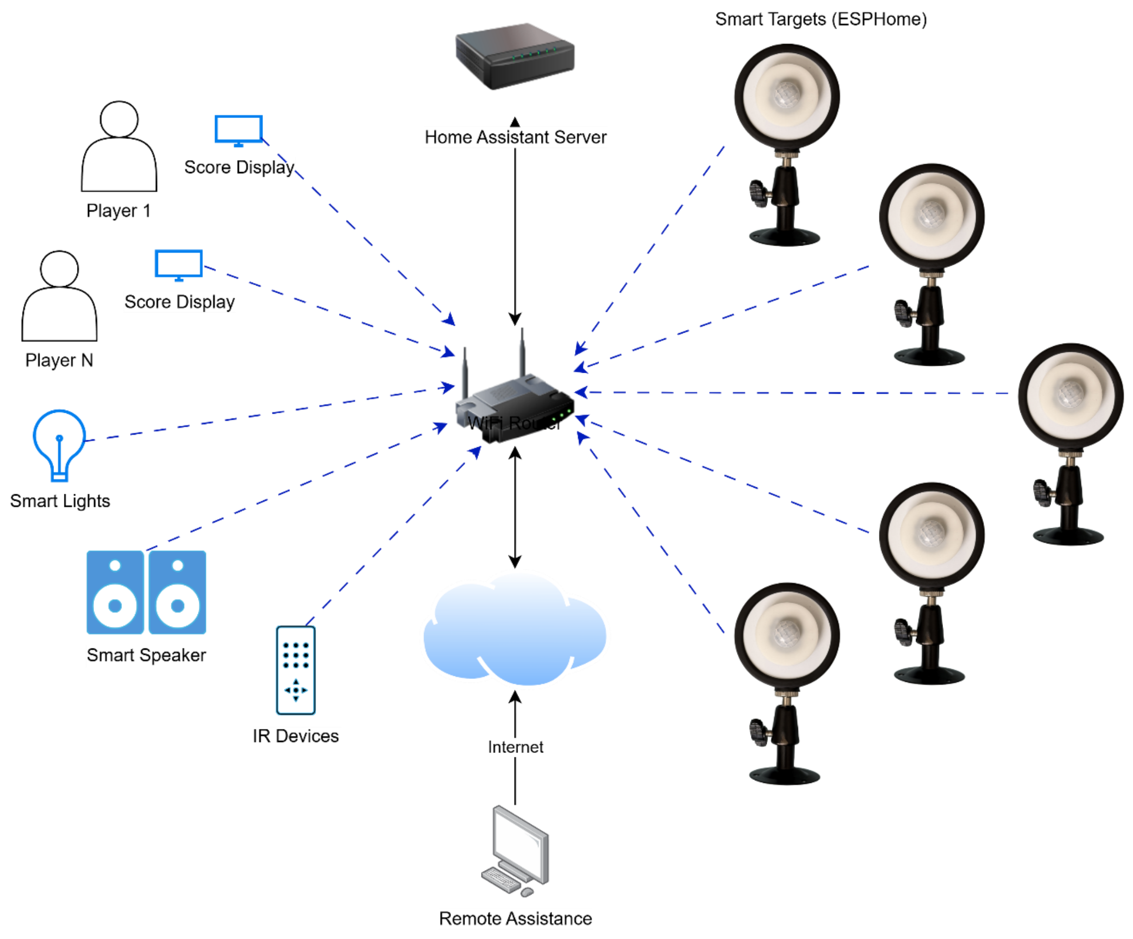


*Figure 7: More complex network*

The server can be a raspberry pi version 4 or 5 with at least 4MB RAM. This is what I use for shooting galleries. I also suggest getting a USB drive instead of using the SD Card as they tend to not be as reliable.

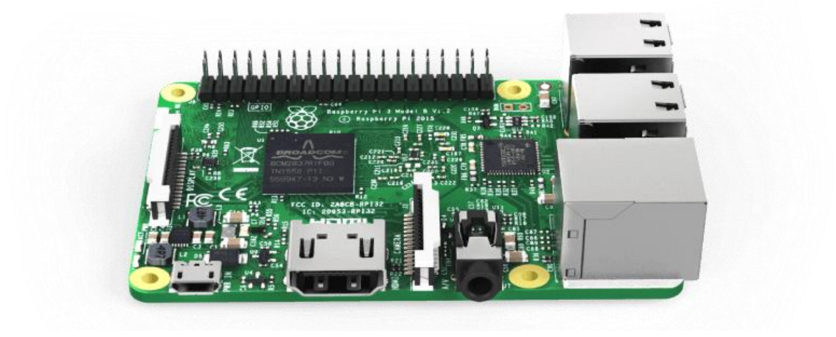


*Figure 8: Raspberry PI*

The Wi-Fi router should be DHCP capable, although each target should be using a static address, DHCP makes it easier for other equipment and computers to connect. If the area is large, you can instead use a mesh network or more capable access point connected to the router. It is important that all targets have a good signal to the server to reduce latency. So be sure to design your wireless network carefully and test each target's signal strength.

## Resources

The current applications and examples can be downloaded here:
https://github.com/CodeMakesItGo/SmartTarget

Home Assistant is the server application. It is free to use and the installation instructions can be found here to install it on a Raspberry Pi:
https://www.home-assistant.io/installation/raspberrypi

Raspberry Pi Imager for creating the Home Assistant image can be found here:
https://www.raspberrypi.com/software/

## Home Assistant Server

The easiest method to install Home Assistant is using the Raspberry Pi Imager and is the method I use.

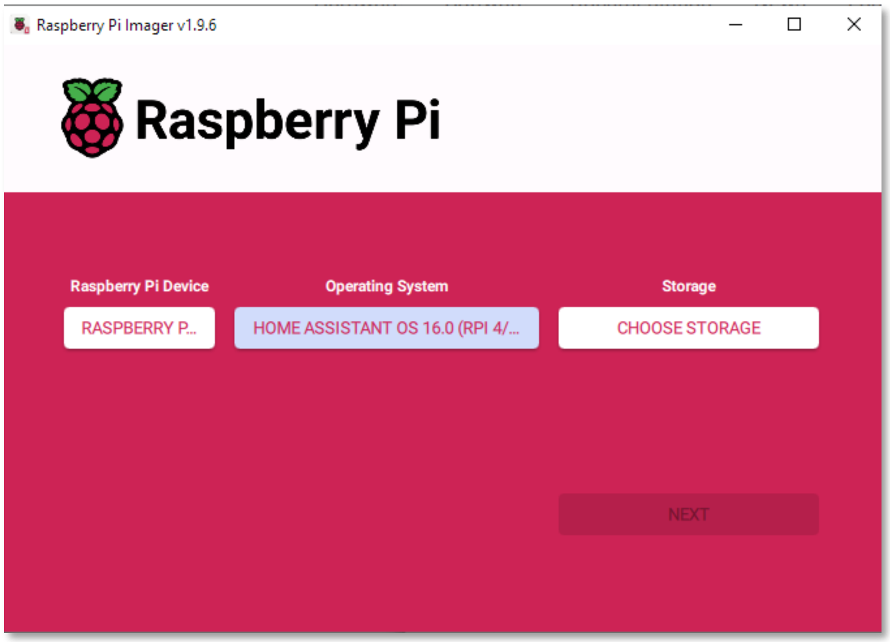


*Figure 9: Raspberry Pi Imager*

The first boot of the Raspberry Pi will take a while. If you have a monitor attached to the server you can watch when it is complete or just keep trying to log into the Home Assistant server using the address: http://homeassistant.local:8123/

Once the previous address is working, you will then have to complete the Home Assistant onboarding. The instructions for this are found here: https://www.home-assistant.io/getting-started/onboarding/

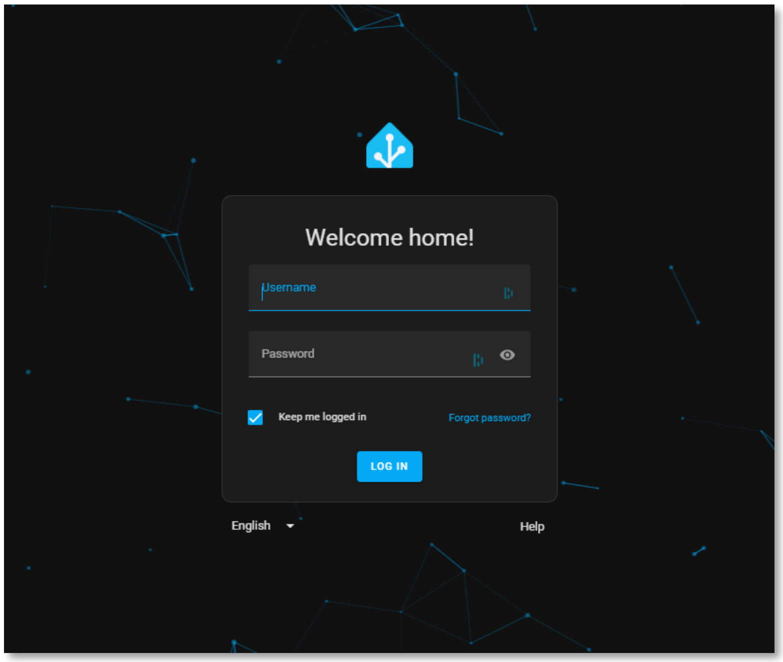


*Figure 10: Home Assistant Login Page*

Finally, log into Home Assistant for the next step.

Ok, that was a lot but if you made it here, the hard part is over. You should now have a login screen to access your Home Assistant server from your computer.

## ESPHome Setup

To operate ESPHome devices, we need to install the ESPHome add-on. On the bottom left panel of Home Assistant there will be 'Settings'. Click on this.

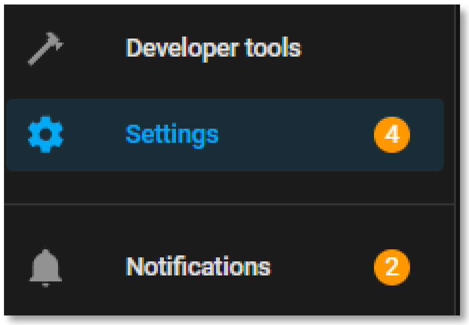


*Figure 11: HA Settings*

Next click on 'Add-ons'

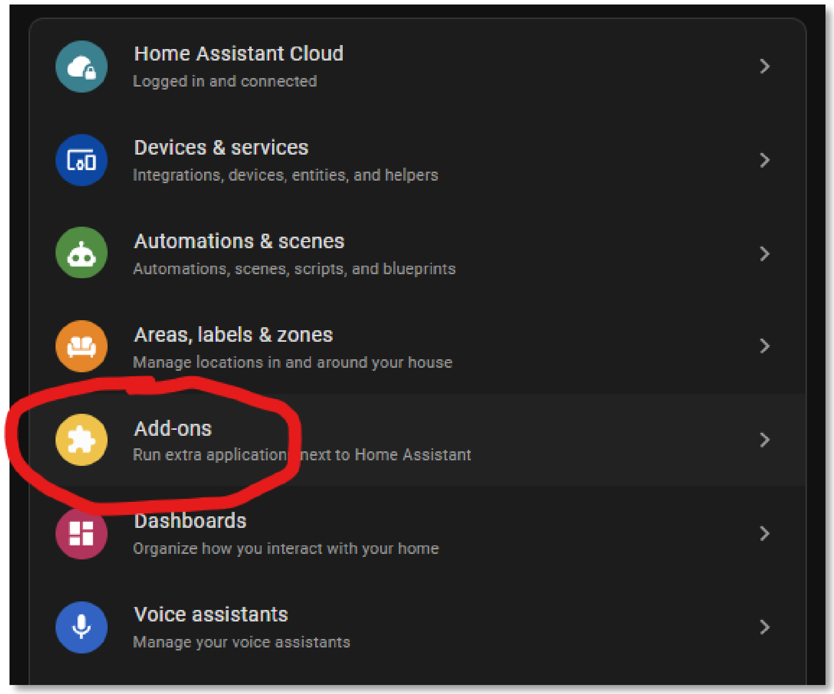

Home Assistant Cloud
Logged in and connected
Devices & services
Integrations, devices, entities, and helpers
Automations & scenes
Automations, scenes, scripts, and blueprints
Areas, labels & zones
Manage locations in and around your house
Add-ons
Run extra applications next to Home Assistant
Dashboards
Organize how you interact with your home
Voice assistants
Manage your voice assistants

*Figure 12: HA Add-ons*

Then search for 'ESPHome' and install it.

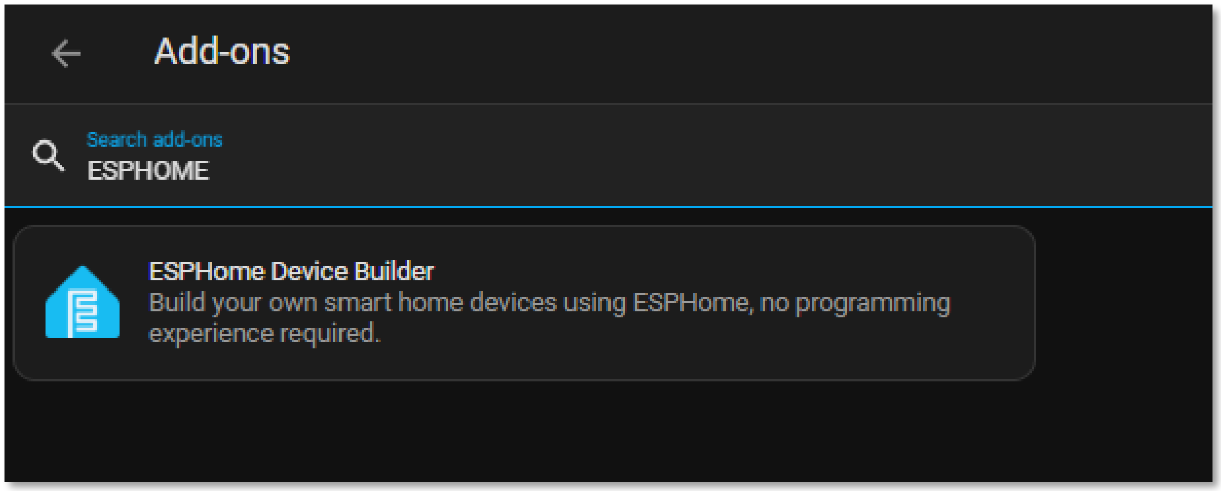

Add-ons
Search add-ons
ESPHOME
ESPHome Device Builder
Build your own smart home devices using ESPHome, no programming experience required.

*Figure 13: HA Install ESPHome*

Now in the left panel you should have an item labeled 'ESPHome Builder'

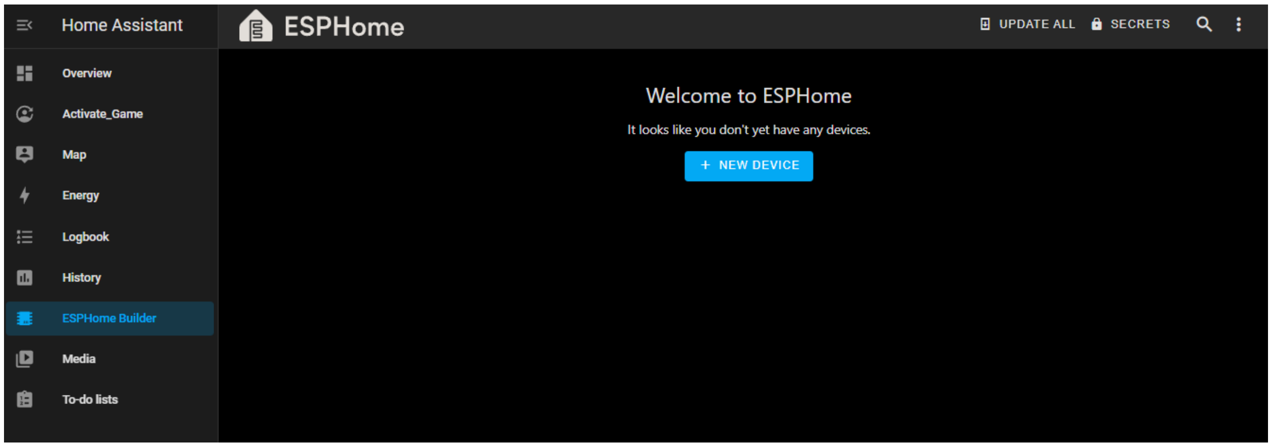


*Figure 14: ESPHome*

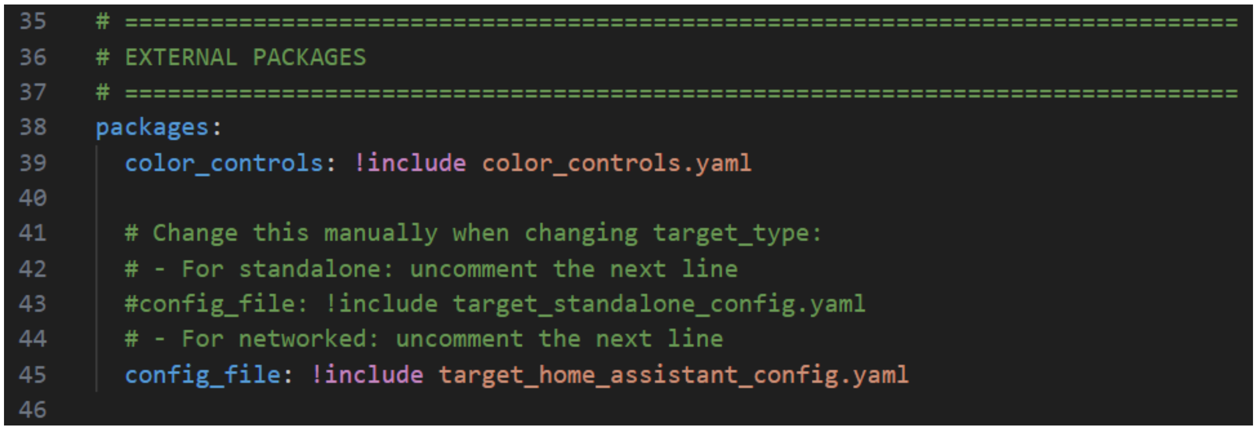
This part is done. The next thing we need to do is make a target appear as an ESPHome device. To do this we must reprogram the target as a network device.

## Programming Networkable Targets

There are two way to program the target through the Home Assistant server. I am going to show you easy method. This is assuming you have ESPHome command line installed. If you do not, please go to the 'Custom Programming' section to read how to set this up.

Download the smart target repository from the GitHub page.
https://github.com/CodeMakesItGo/SmartTarget

Open the 'target_revx.x.yaml' file in a text editor and find the 'External Packages' section in the file.



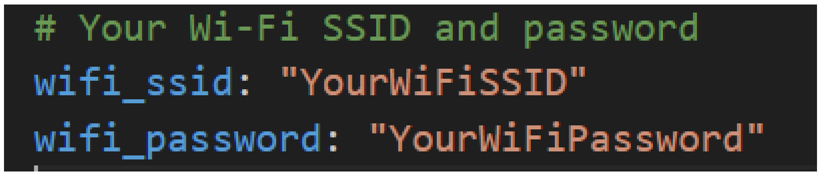
```yaml
# ============================================================================
# EXTERNAL PACKAGES
# ============================================================================
packages:
  color_controls: !include color_controls.yaml

  # Change this manually when changing target_type:
  # - For standalone: uncomment the next line
  #config_file: !include target_standalone_config.yaml
  # - For networked: uncomment the next line
  config_file: !include target_home_assistant_config.yaml

```

Ensure that the standalone line is commented out with a '#' and remove the '#' from the networked line. Basically, ensure this section in your file looks like the above snapshot where line 45 is uncommented and line 43 is commented.

Next you will need to update your WiFi network's information. Update the information in the secrets.yaml file so that the target knows how to log into your network.


```
# Your Wi-Fi SSID and password
wifi_ssid: "YourWiFiSSID"
wifi_password: "YourWiFiPassword"
```

Now program the target, after connecting it to your machine with a USB cable, using the following command.

```
esphome -s ip 2 run .\target_rev1_x.yaml
```

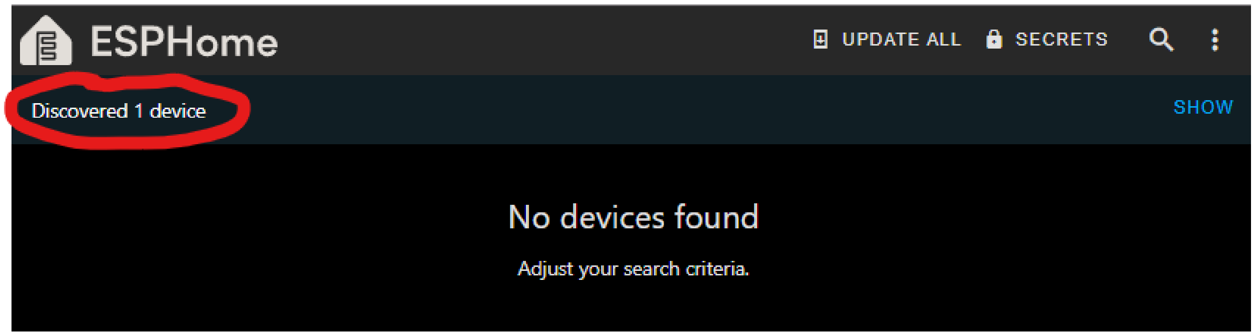
This will set the target's IP address to 192.168.8.2. It is best to use static addresses for better reliability.

*Warning: Be sure to set each target's IP address differently.

*Tip: Organize your addresses so you can quickly reference what a device might be based on its address. Here is an example of an address scheme I have used in a previous shooting gallery.

| Group | IP | Item |
|---|---|---|
| Network | 1 | DHCP Router |
| | 2 | Access Point |
| Compute | 11 | HA server |
| Props | 101-135 | Targets |
| | 111-112 | Speakers |
| Players | 201-204 | Displays |
| | 211-214 | Speakers |
| | 221-224 | Guns |
| | 231-234 | Lanterns\Lights |

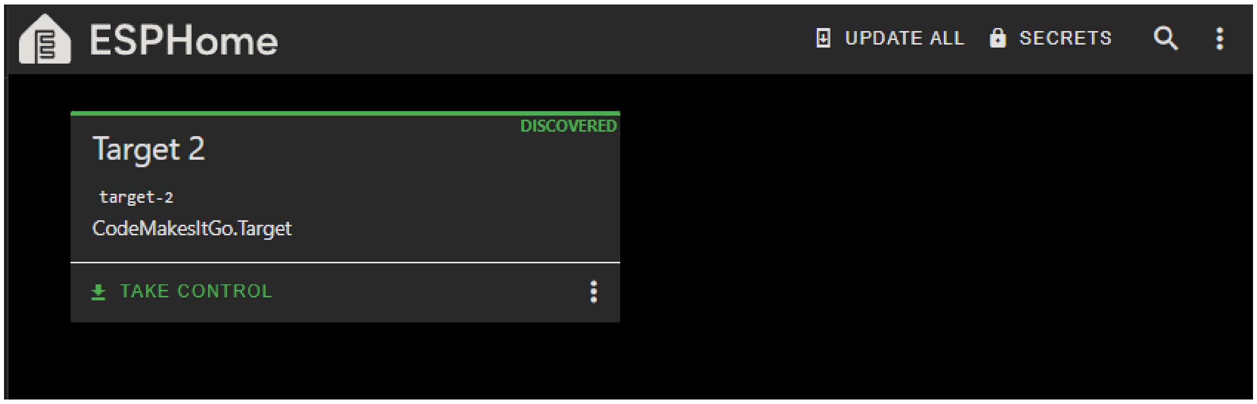
Go back to the Home Assistant server under the 'ESPHome Builder' tab. You should now see a 'Discovered 1 device'.



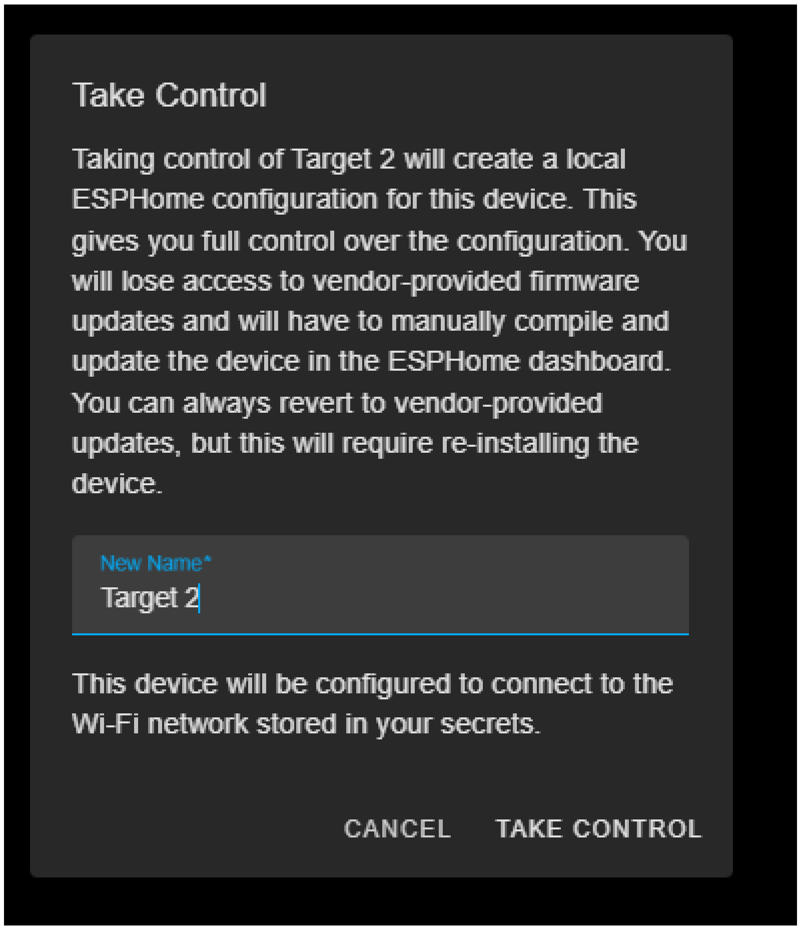
No devices found

Adjust your search criteria.

Click on the 'show' text on the right side of the screen. You will see a box with the target name as a discovered device. Select 'Take Control'



target-2
CodeMakesItGo.Target


Take Control

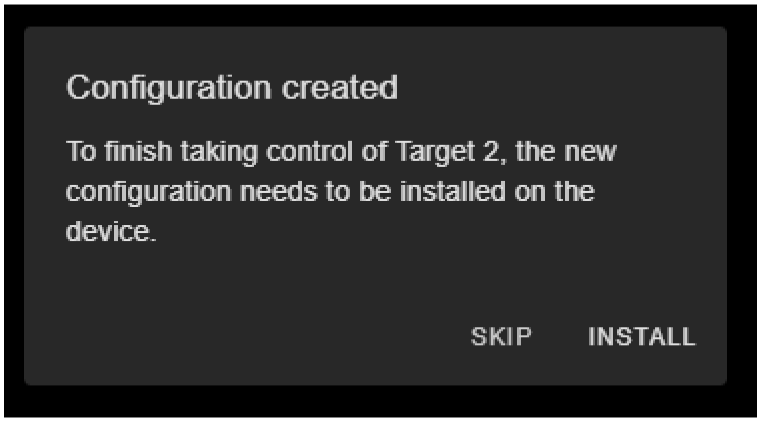
Taking control of Target 2 will create a local ESPHome configuration for this device. This gives you full control over the configuration. You will lose access to vendor-provided firmware updates and will have to manually compile and update the device in the ESPHome dashboard. You can always revert to vendor-provided updates, but this will require re-installing the device.

New Name*: Target 2

This device will be configured to connect to the Wi-Fi network stored in your secrets.

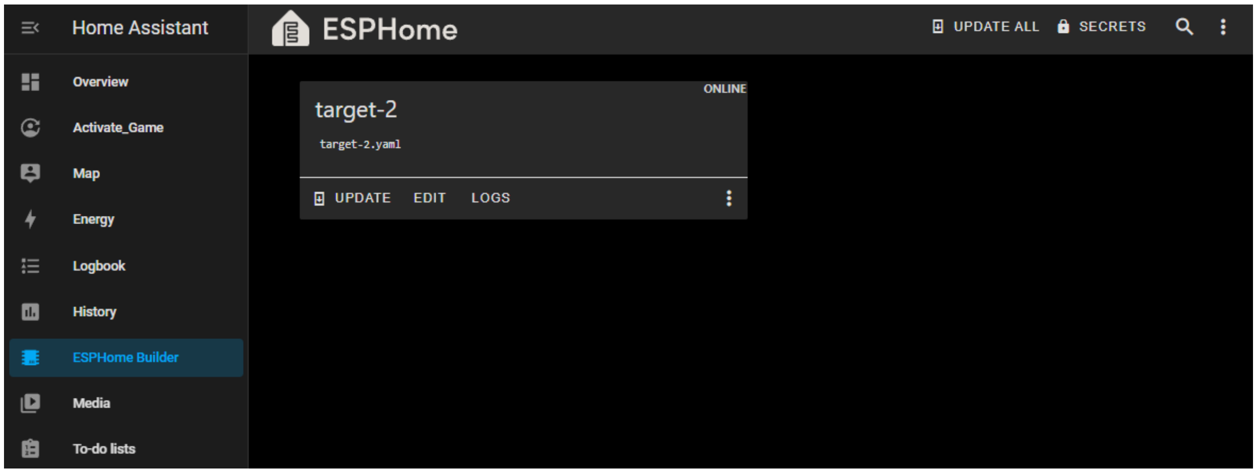
A dialog box will appear asking you to name the target. You can always rename them later but if you know what this target will do then you can name it now. For example, if this is the target that operates the coffin prop, then maybe name it 'target-coffin'. Select 'Take Control' when you are ready.

Another dialog box will appear, select 'skip' because we have already programmed the target.


Configuration created

To finish taking control of Target 2, the new configuration needs to be installed on the device.

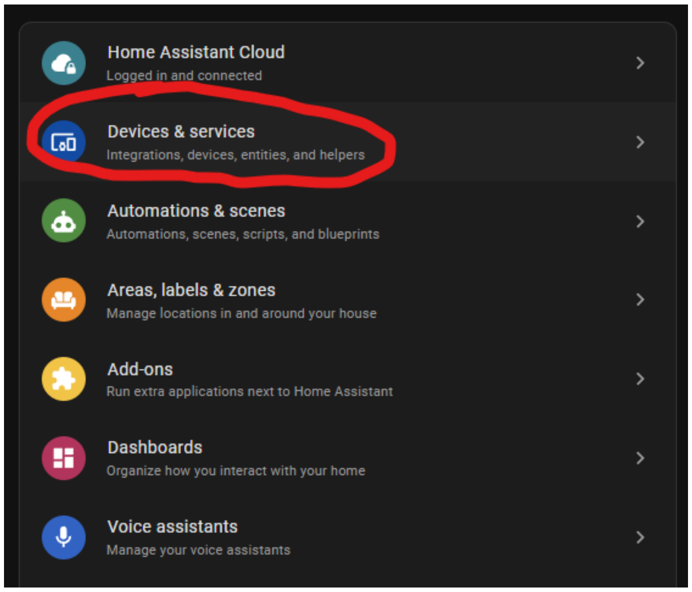
After this step, the 'ESPHome Builder' tab should display the target as being online.



## ESPHome Devices

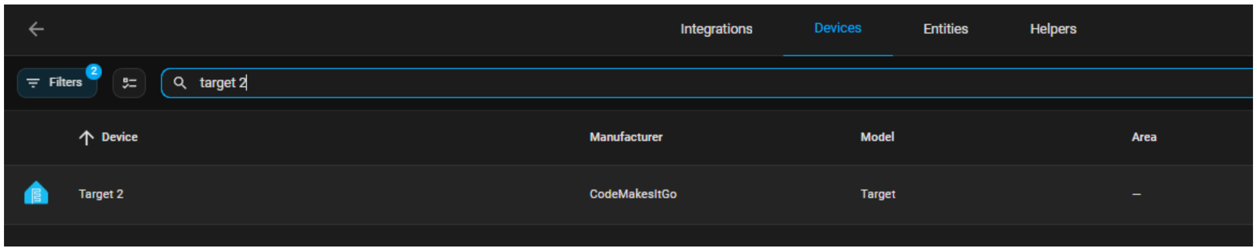
Once the target is programmed and added to the Home Assistant you can view and modify its settings through Home Assistant.

Go to Settings→'Devices & Services'


Home Assistant Cloud
Logged in and connected
Devices & services
Integrations, devices, entities, and helpers
Automations & scenes
Automations, scenes, scripts, and blueprints
Areas, labels & zones
Manage locations in and around your house
Add-ons
Run extra applications next to Home Assistant
Dashboards
Organize how you interact with your home
Voice assistants
Manage your voice assistants

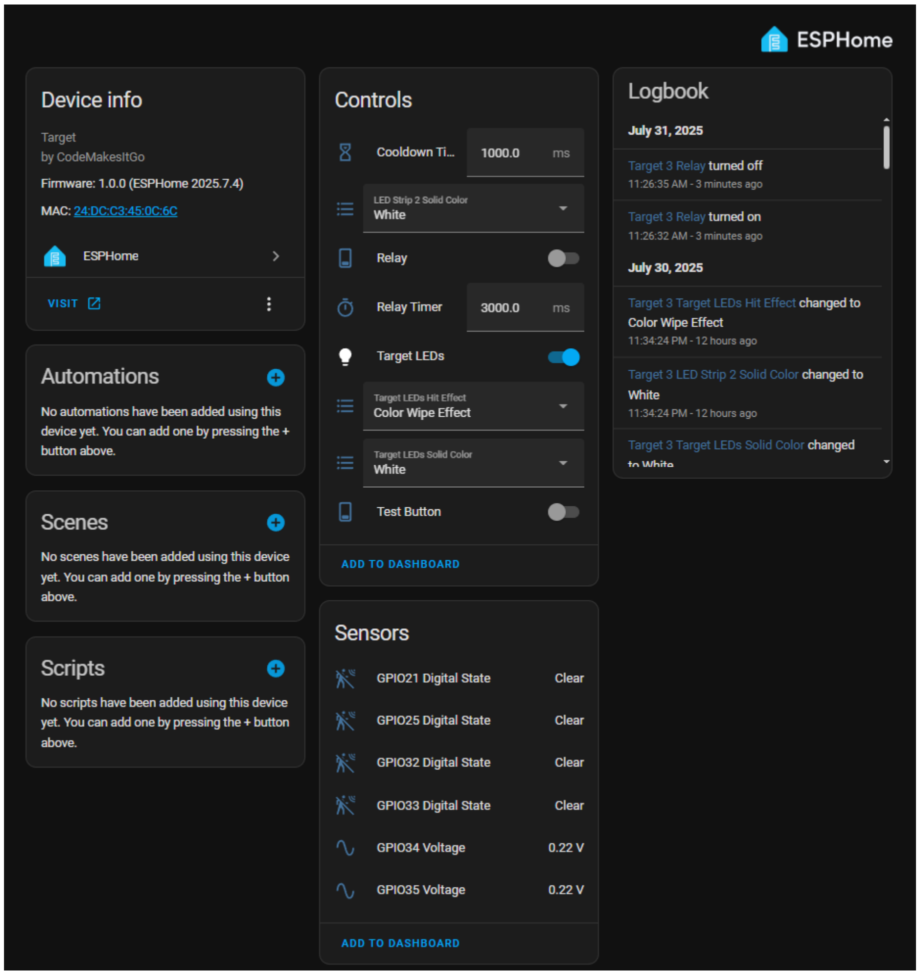
Then select 'Devices' on the top menu bar. This will display all your devices connected to the Home Assistant.

Search and select the name of the target you just programmed.



| Device | Manufacturer | Model | Area |
|---|---|---|---|
| Target 2 | CodeMakesItGo | Target | – |

Similar to the target's web page, you can use controls on the target from this page.



Device info
Target
by CodeMakesItGo
Firmware: 1.0.0 (ESPHome 2025.7.4)
MAC: 24:DC:C3:45:0C:6C
ESPHome

Automations
No automations have been added using this device yet. You can add one by pressing the + button above.

Scenes
No scenes have been added using this device yet. You can add one by pressing the + button above.

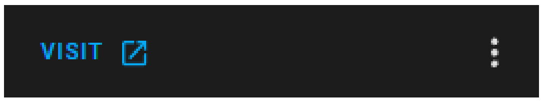
Scripts
No scripts have been added using this device yet. You can add one by pressing the + button above.

Controls
Cooldown Ti... 1000.0 ms
LED Strip 2 Solid Color: White
Relay
Relay Timer 3000.0 ms
Target LEDs
Target LEDs Hit Effect: Color Wipe Effect
Target LEDs Solid Color: White
Test Button

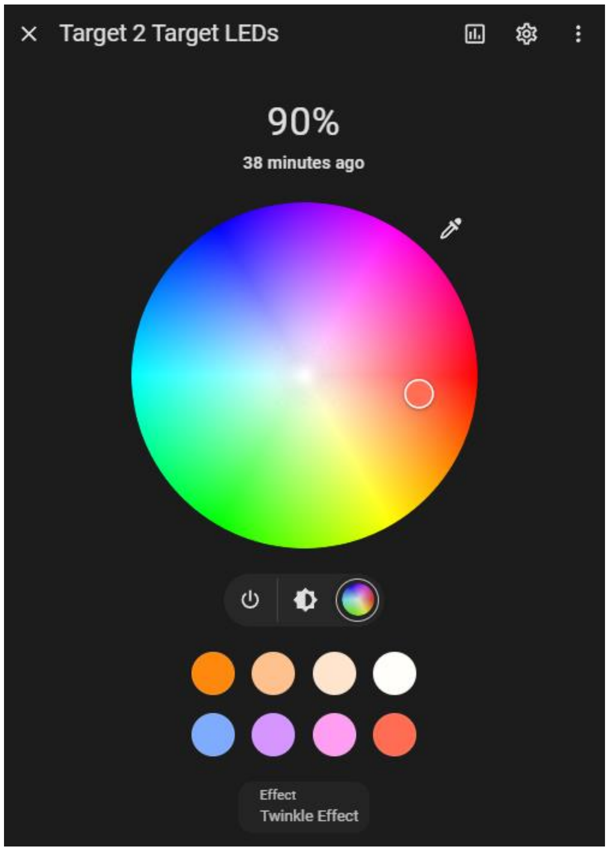
Sensors
GPIO21 Digital State Clear
GPIO25 Digital State Clear
GPIO32 Digital State Clear
GPIO33 Digital State Clear
GPIO34 Voltage 0.22 V
GPIO35 Voltage 0.22 V

Logbook
July 31, 2025
Target 3 Relay turned off
11:26:35 AM - 3 minutes ago
Target 3 Relay turned on
11:26:32 AM - 3 minutes ago
July 30, 2025
Target 3 Target LEDs Hit Effect changed to Color Wipe Effect
11:34:24 PM - 12 hours ago
Target 3 LED Strip 2 Solid Color changed to White
11:34:24 PM - 12 hours ago
Target 3 Target LEDs Solid Color changed to White

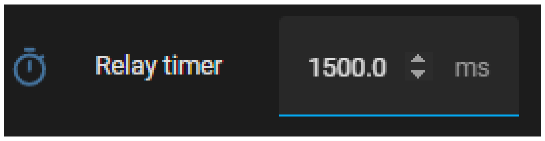
If you want to visit the target's web page, you can click on the 'visit' button, and you will be taken to the target's web page. This only works if the machine you are using is also connected to the target network.



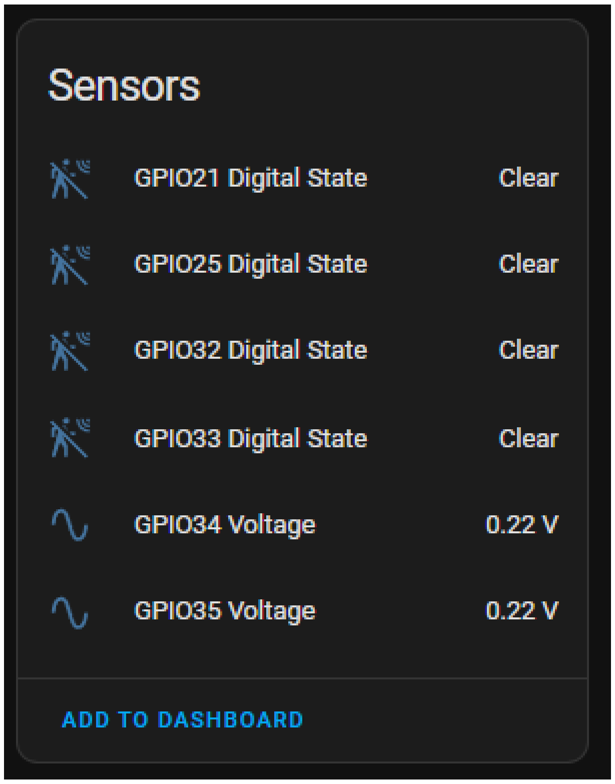
The device page in Home Assistant differs slightly from the one on the target's webpage; for instance, LED color controls are more detailed.



Additionally, users are provided with more precise control over the time values by allowing direct input, as opposed to relying solely on a slider.

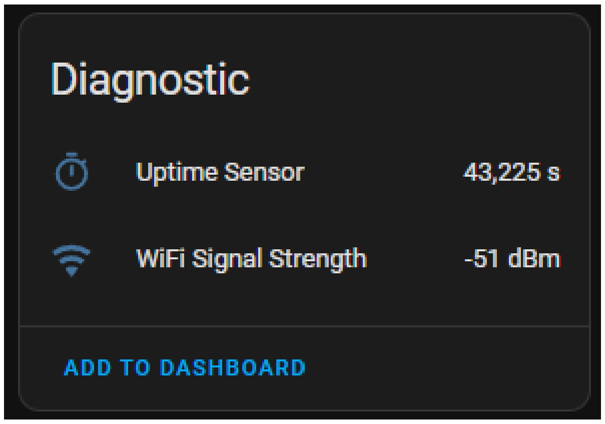

Relay timer 1500.0 ms

You will also find there is better grouping of controls. The sensors and Diagnostics are separated into their own groups.



Sensors

| | |
|---|---|
| GPIO21 Digital State | Clear |
| GPIO25 Digital State | Clear |
| GPIO32 Digital State | Clear |
| GPIO33 Digital State | Clear |
| GPIO34 Voltage | 0.22 V |
| GPIO35 Voltage | 0.22 V |



Diagnostic

| | |
|---|---|
| Uptime Sensor | 43,225 s |
| WiFi Signal Strength | -51 dBm |

This screen is also where you should ensure all targets have a good Wi-Fi signal by evaluating the reported 'WiFi Signal Strength' and referencing the table below. Ideally you will want the signal to be in the low -60's range.

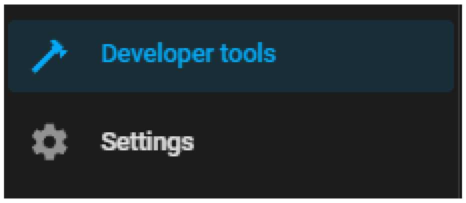
| Signal Strength (dBm) | Quality | Meaning |
|---|---|---|
| **-30 dBm** | Amazing | Maximum strength, ideal if next to the router |
| **-50 dBm** | Excellent | Strong signal, fast and reliable connection |

| | | |
|---|---|---|
| **-60 dBm** | Very Good | Reliable for most uses including streaming/video calls |
| **-67 dBm** | Good | Minimum for stable VoIP and streaming |
| **-70 dBm** | Fair | OK for web browsing, not ideal for streaming |
| **-80 dBm** | Poor | Unreliable, may cause disconnects |
| **-90 dBm or lower** | Unusable | Likely disconnected or non-functional |

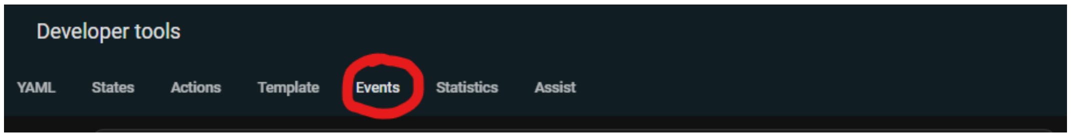
## Listening to events

A big part of automation for a networked component is the ability for the target to communicate when it was hit and by which player. Now that the target is connected to the server we can test if the server is receiving a message when the target is hit.

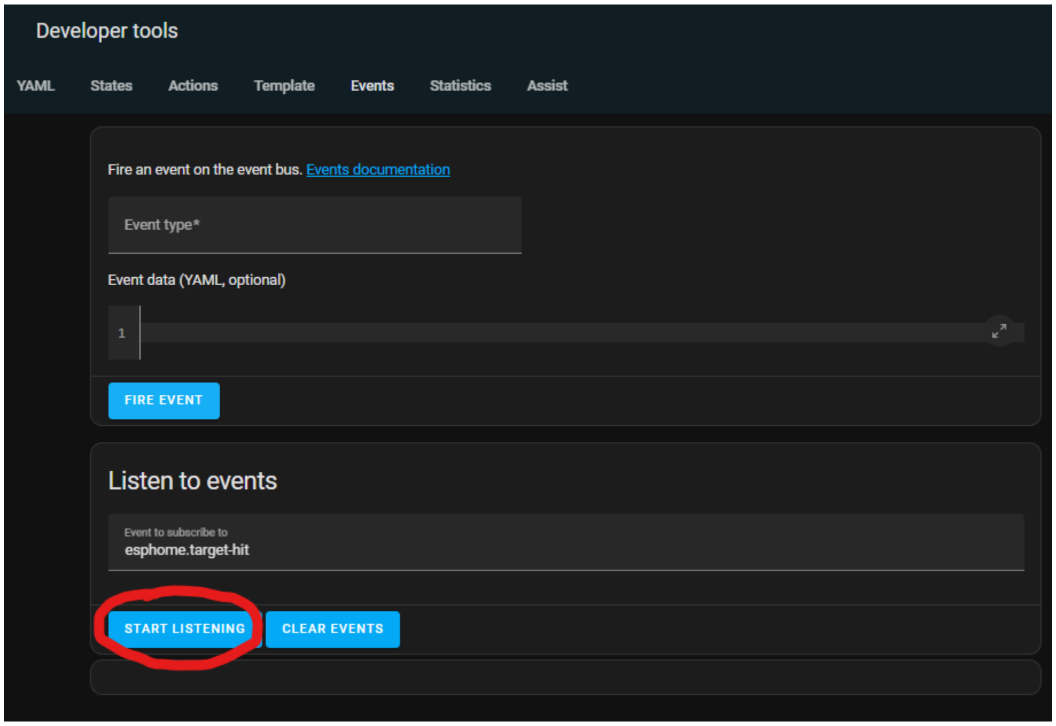
To do this go to the 'Developer tools' panel.



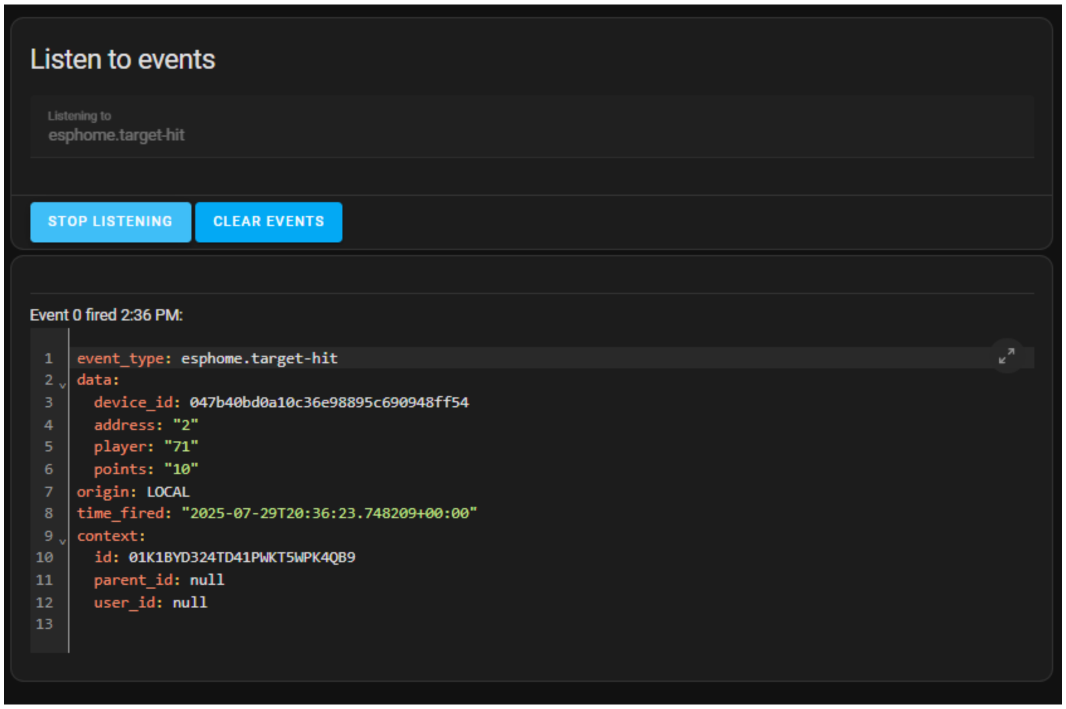
Then click on the 'Events' menu at the top.



Got to the 'Listen to events' group and type in the following: 'esphome.target-hit' and click on the 'Start Listening' button.



Now, using an IR transmitter, trigger the target. You should see an output like the following:



Event 0 fired 2:36 PM:

```
event_type: esphome.target-hit
data:
  device_id: 047b40bd0a10c36e98895c690948ff54
  address: "2"
  player: "71"
  points: "10"
origin: LOCAL
time_fired: "2025-07-29T20:36:23.748209+00:00"
context:
  id: 01K1BYD324TD41PWKT5WPK4QB9
  parent_id: null
  user_id: null
```

Under the ‘data’ section you can see the following information:

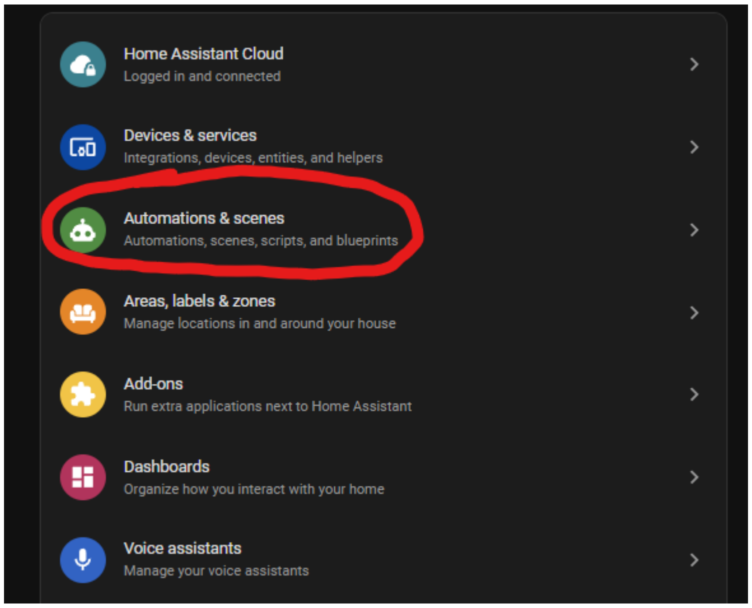
| Data Item | Description |
|---|---|
| **Address** | The ID of the target reporting the hit. |
| **Player** | The command from the IR signal. This should be coded so that you know which player hit the target |
| **Points** | The configured points are awarded to the player for hitting this target. This can be changed on each target with the 'default_points' value. |

If you are receiving this data from a hit event then we can finally make an automation to trigger some other event.

## Automation Scripts

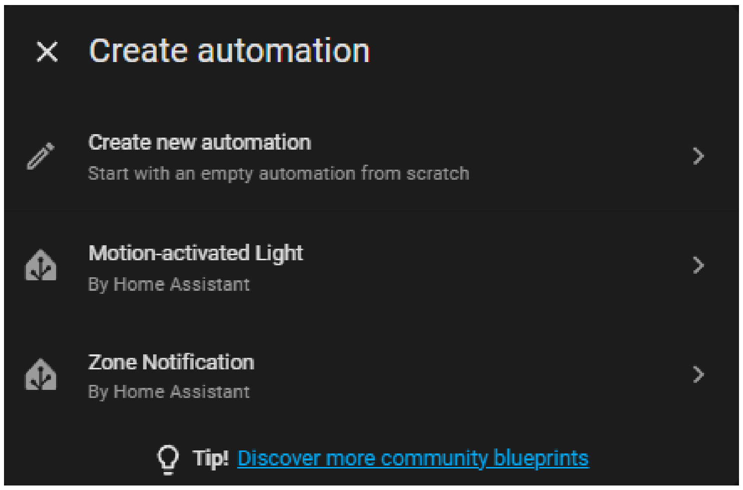
For this example, we will trigger a smart bulb to turn on for 5 seconds after the target is hit.

Got to settings and select the 'Automations & scenes'


Home Assistant Cloud
Logged in and connected
Devices & services
Integrations, devices, entities, and helpers
Automations & scenes
Automations, scenes, scripts, and blueprints
Areas, labels & zones
Manage locations in and around your house
Add-ons
Run extra applications next to Home Assistant
Dashboards
Organize how you interact with your home
Voice assistants
Manage your voice assistants

On the bottom right side of the screen select '+ Create Automation'.

A dialog will appear for creating an automation. Select 'Create new automation'



Create automation

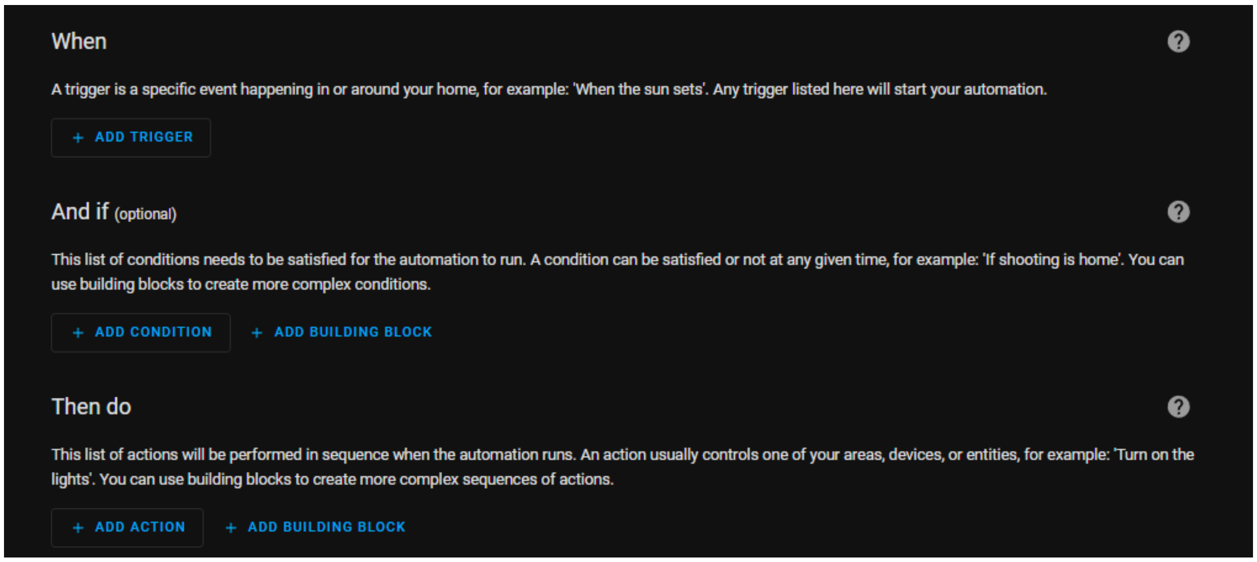
Create new automation
Start with an empty automation from scratch

Motion-activated Light
By Home Assistant

Zone Notification
By Home Assistant

Tip! Discover more community blueprints

Automations are based on events. The next screen helps you describe the event graphically and what to do when the event happens. Under the 'When' section select '+ Add Trigger'.



When

A trigger is a specific event happening in or around your home, for example: 'When the sun sets'. Any trigger listed here will start your automation.

+ ADD TRIGGER

And if (optional)

This list of conditions needs to be satisfied for the automation to run. A condition can be satisfied or not at any given time, for example: 'If shooting is home'. You can use building blocks to create more complex conditions.

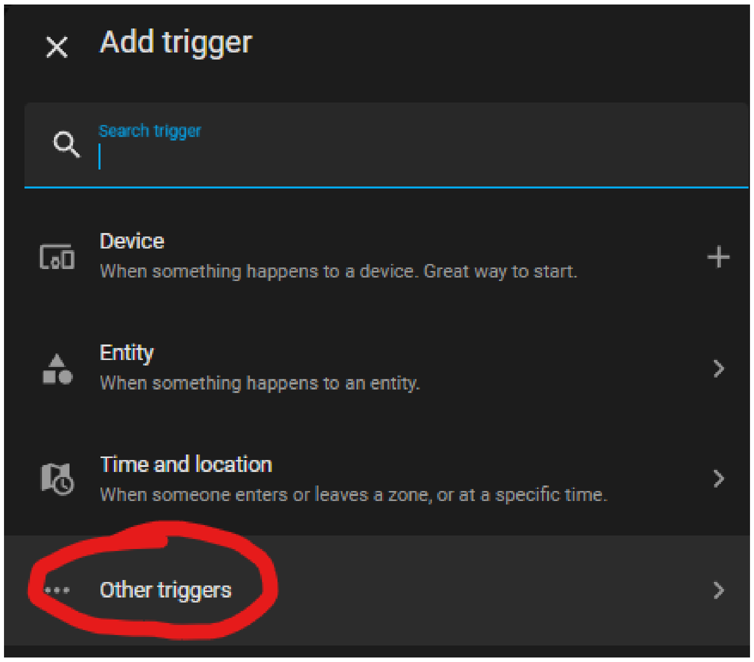
+ ADD CONDITION  + ADD BUILDING BLOCK

Then do

This list of actions will be performed in sequence when the automation runs. An action usually controls one of your areas, devices, or entities, for example: 'Turn on the lights'. You can use building blocks to create more complex sequences of actions.

+ ADD ACTION  + ADD BUILDING BLOCK

Then in the next dialog, select '… Other triggers'



Add trigger

Search trigger

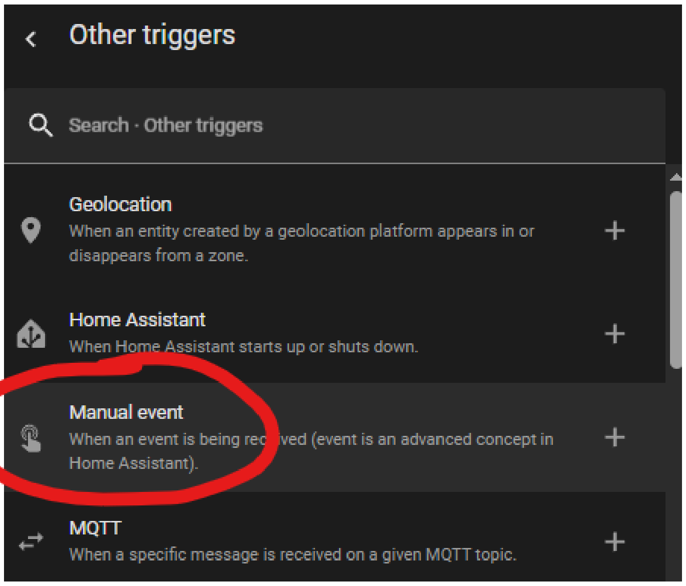
Device
When something happens to a device. Great way to start.

Entity
When something happens to an entity.

Time and location
When someone enters or leaves a zone, or at a specific time.

Other triggers

Next select 'Manual event'



Other triggers

Search · Other triggers

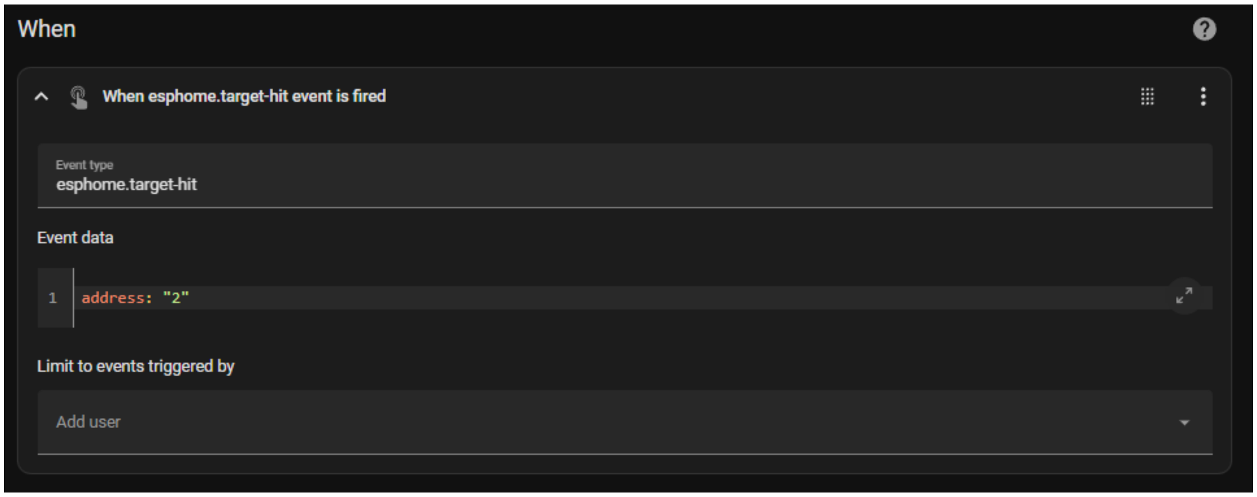
Geolocation
When an entity created by a geolocation platform appears in or disappears from a zone.

Home Assistant
When Home Assistant starts up or shuts down.

Manual event
When an event is being received (event is an advanced concept in Home Assistant).

MQTT
When a specific message is received on a given MQTT topic.

Set the event type to 'esphome.target-hit' and the event data to 'address: "2"'



When

When esphome.target-hit event is fired

Event type
esphome.target-hit

Event data

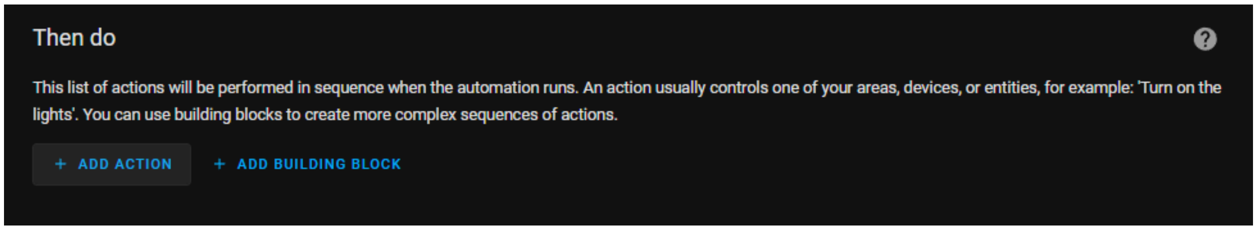
```
address: "2"
```

Limit to events triggered by

Add user

Now that the event trigger is set, we must set an action.

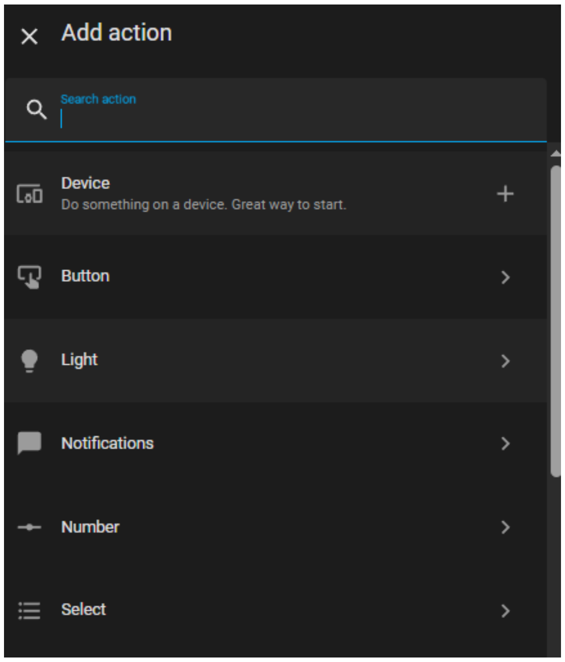
Under the ‘Then do’ section select the ‘+ Add Action’.



Then do

This list of actions will be performed in sequence when the automation runs. An action usually controls one of your areas, devices, or entities, for example: 'Turn on the lights'. You can use building blocks to create more complex sequences of actions.

Then select ‘Light’

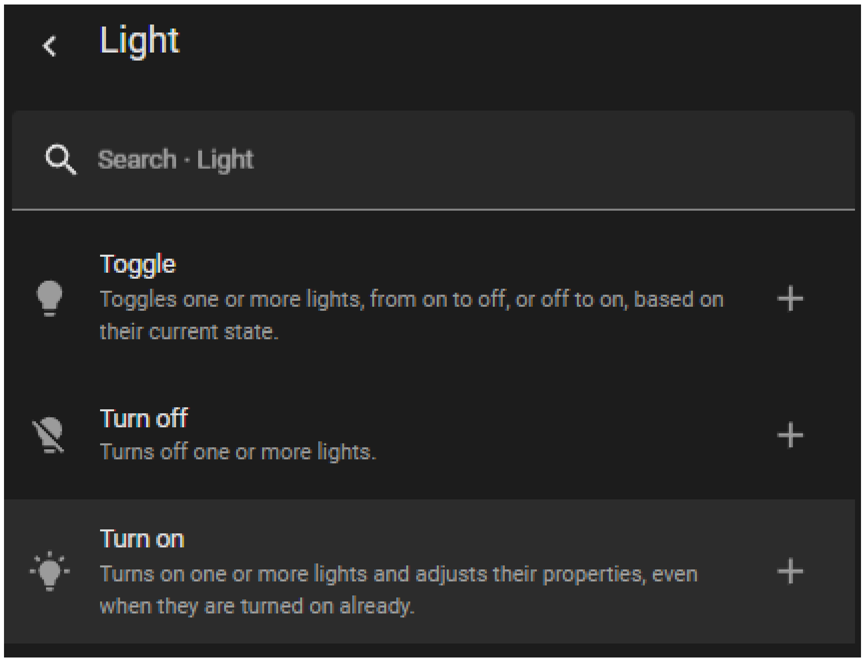


Add action

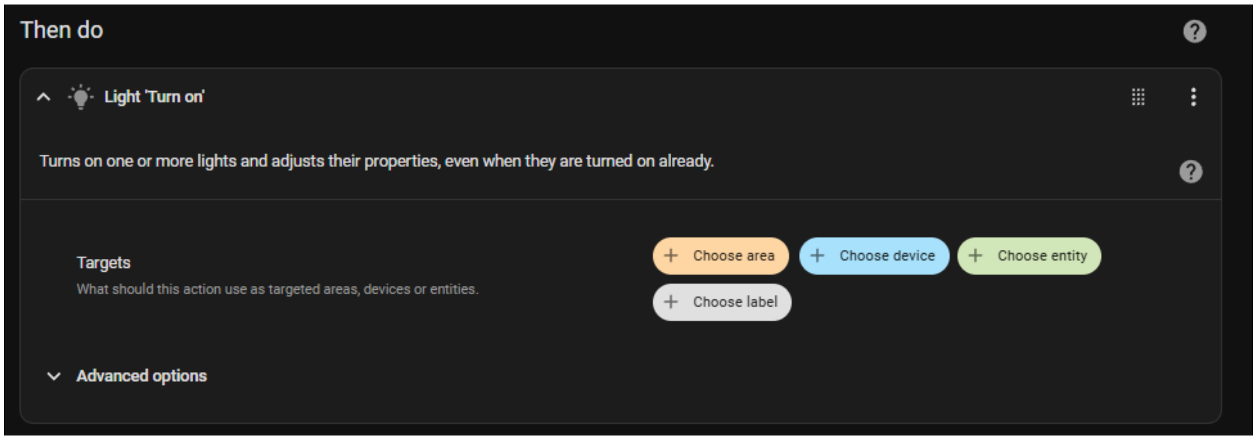
Search action

Device
Do something on a device. Great way to start.

Then select ‘Turn on’.


Light
Search · Light
Toggle
Toggles one or more lights, from on to off, or off to on, based on their current state.
Turn off
Turns off one or more lights.
Turn on
Turns on one or more lights and adjusts their properties, even when they are turned on already.

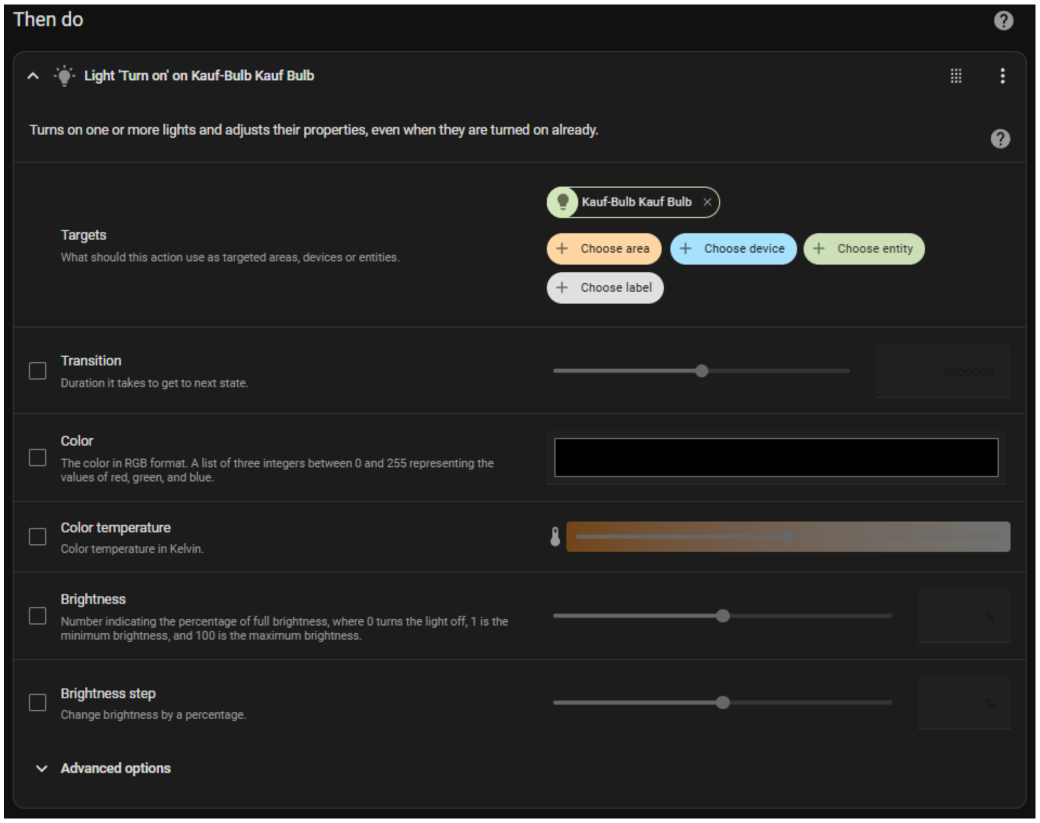
The ‘Then do’ section will now look like the following:


Then do
Light 'Turn on'
Turns on one or more lights and adjusts their properties, even when they are turned on already.
Targets
What should this action use as targeted areas, devices or entities.
Choose area
Choose device
Choose entity
Choose label
Advanced options

Select ‘Choose entity’ and the choose a smart bulb.

For this example I used a Kauf Smart Bulb that is also an ESPHome device. Setup is similar to how we installed the target as an ESPHome device. It can be purchased on Amazon. More information about them can be found here: https://kaufha.com/

Once the entity smart bulb is chosen, you can also configure the color or brightness because the Home Assistant knowns this is a light bulb.


Then do

Light 'Turn on' on Kauf-Bulb Kauf Bulb

Turns on one or more lights and adjusts their properties, even when they are turned on already.

Targets
What should this action use as targeted areas, devices or entities.
Kauf-Bulb Kauf Bulb
Choose area
Choose device
Choose entity
Choose label

Transition
Duration it takes to get to next state.
seconds

Color
The color in RGB format. A list of three integers between 0 and 255 representing the values of red, green, and blue.

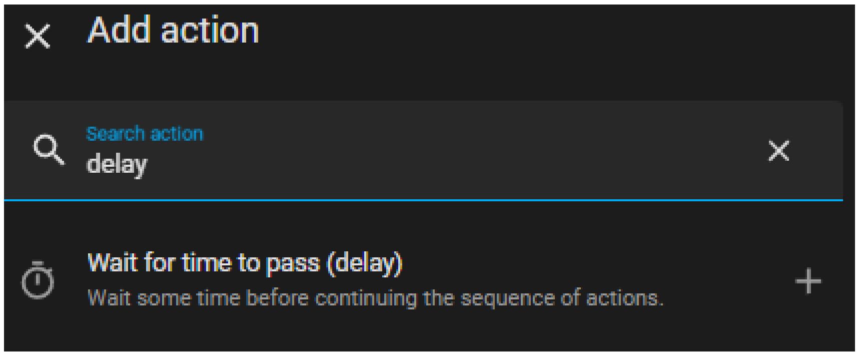
Color temperature
Color temperature in Kelvin.

Brightness
Number indicating the percentage of full brightness, where 0 turns the light off, 1 is the minimum brightness, and 100 is the maximum brightness.
%

Brightness step
Change brightness by a percentage.
%

Advanced options

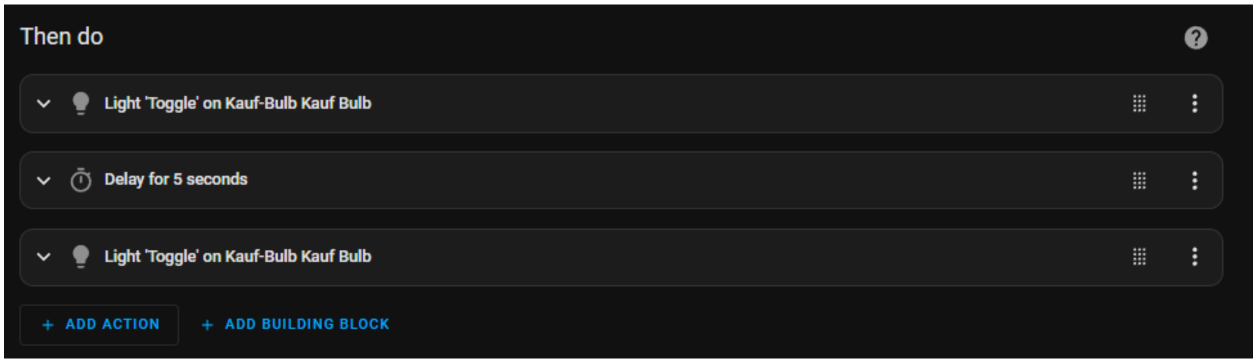
Next, search and add a new action for a delay.


Add action

Search action
delay

Wait for time to pass (delay)
Wait some time before continuing the sequence of actions.

Set the delay for 5 seconds then add another action to turn the light back off. Our final action list will look like this:

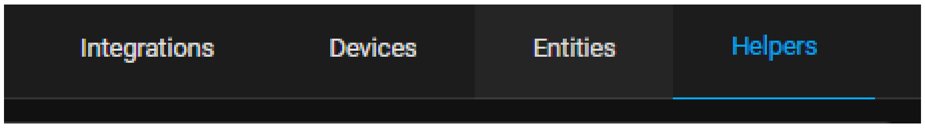

Then do
Light 'Toggle' on Kauf-Bulb Kauf Bulb
Delay for 5 seconds
Light 'Toggle' on Kauf-Bulb Kauf Bulb

When the target is hit, the light bulb will\should turn on for 5 seconds and then back off.

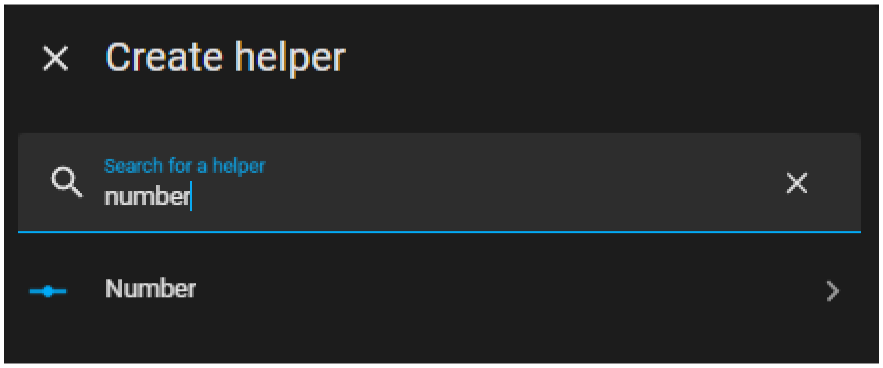
## Score Keeping

To keep score we will need to add a helper, literally.

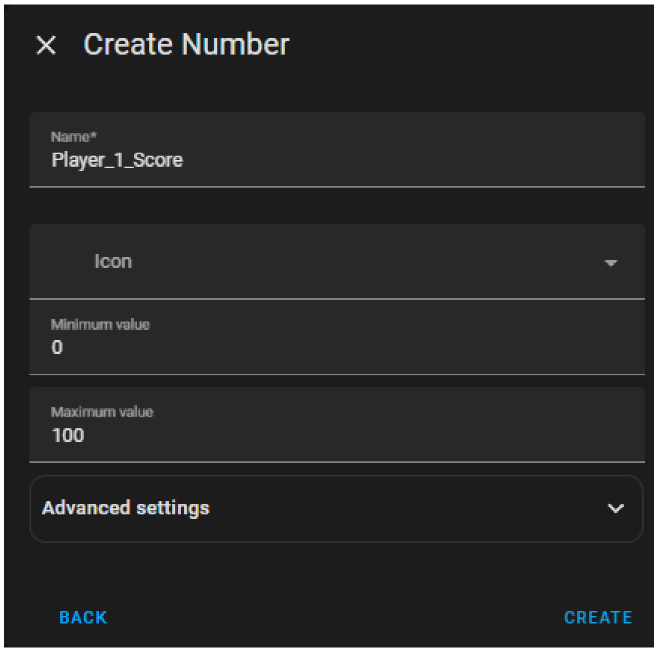
Go to 'Devices & services', then select 'Helpers'



On the bottom right side of the screen click 'Create Helper'.


Create helper
Search for a helper
number
Number

A dialog will appear, search for 'number' and select it.


Create Number

Name*
Player_1_Score

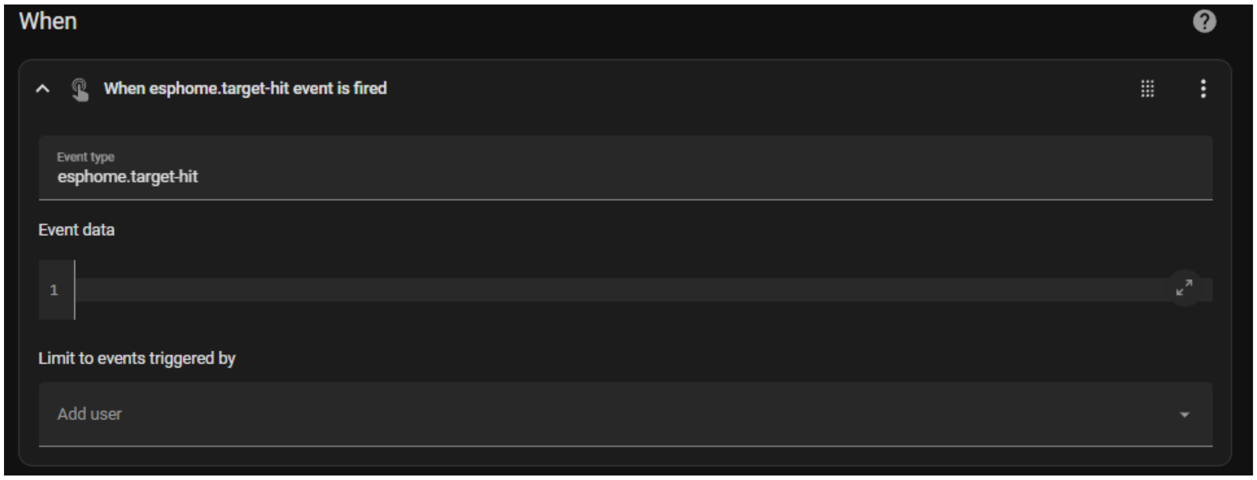
Icon

Minimum value
0

Maximum value
100

Name the number helper 'Player_1_Score' and set the maximum score. That is all we need to do with the helper. It is basically an internal variable we can use now.

Go back to the 'Automation & scenes' and create a new automation. Trigger the event on the 'esphome.target-hit' message but leave the event data section blank.


When

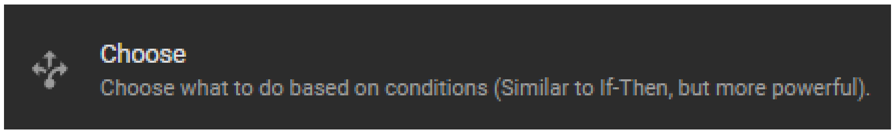
When esphome.target-hit event is fired

Event type
esphome.target-hit

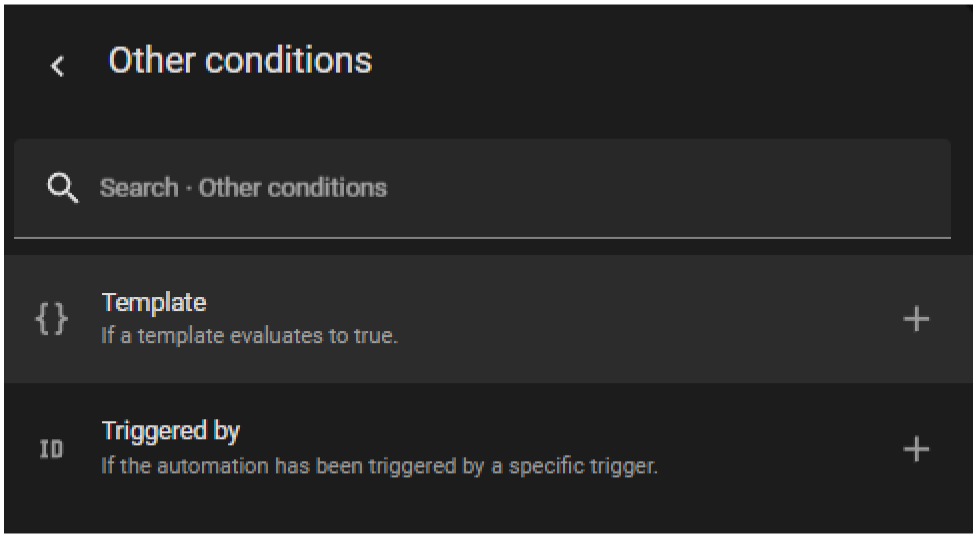
Event data

This will cause this event to be triggered when any target is hit.

Next, add a 'choose' building block in the 'Then do' section.

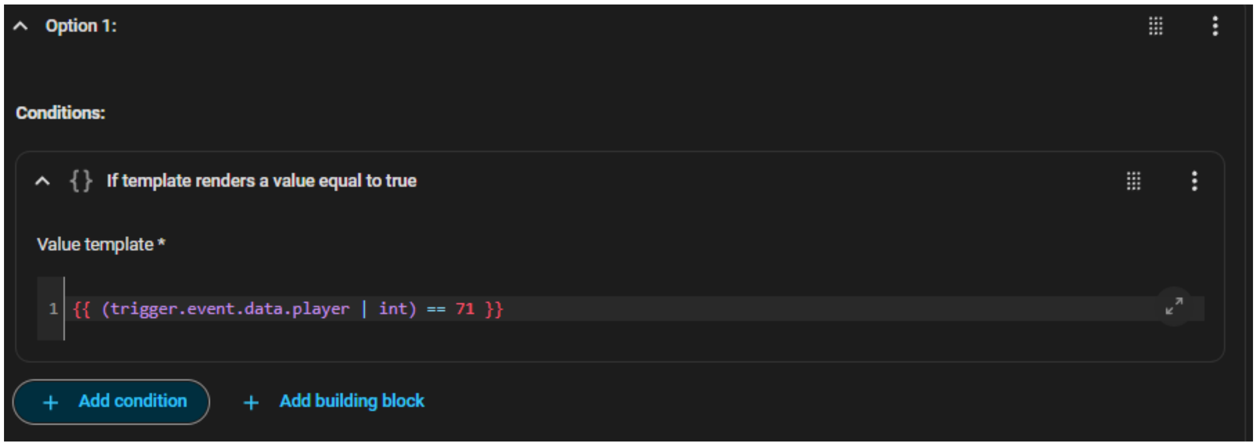

Choose
Choose what to do based on conditions (Similar to If-Then, but more powerful).

For 'option 1 condition' in the 'choose' block, add the 'template' condition.


Other conditions
Search · Other conditions
Template
If a template evaluates to true.
Triggered by
If the automation has been triggered by a specific trigger.

Then add this template to trigger option 1 when the player ID is 71.

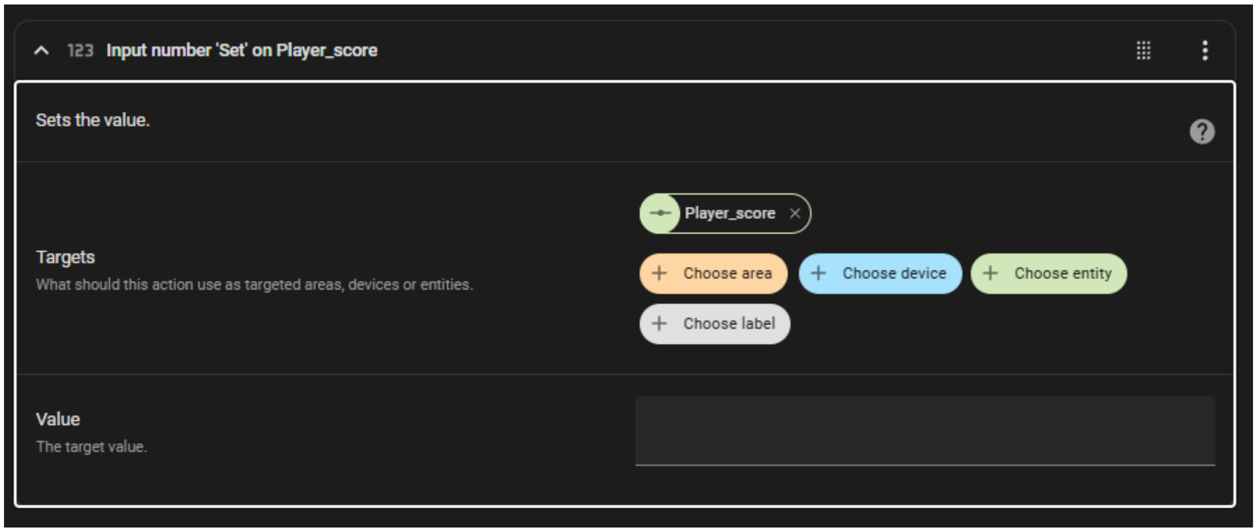
{{ (trigger.event.data.player | int) == 71 }}



```
{{ (trigger.event.data.player | int) == 71 }}
```

For the action, increase the score helper by the target's points.

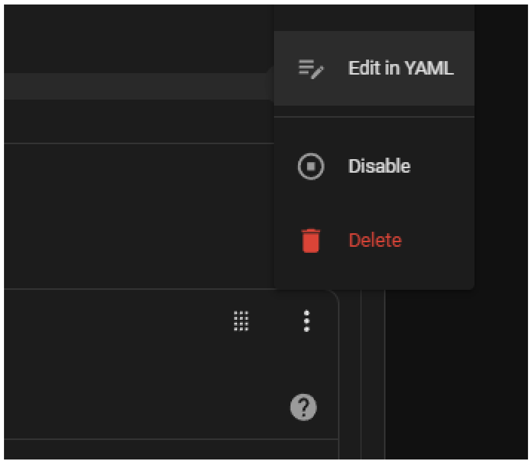
To do this add in a 'Input number Set'.

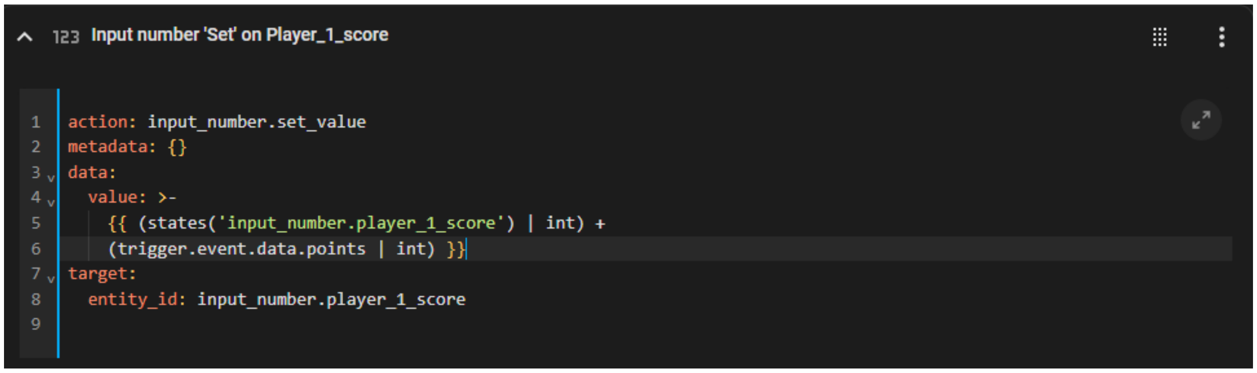

123 Input number 'Set' on Player_score
Sets the value.
Targets
What should this action use as targeted areas, devices or entities.
Player_score
Choose area
Choose device
Choose entity
Choose label
Value
The target value.

Then go to yaml view because the visual editor will only allow you to enter numbers.

The target is our player_1_score helper we just created. Click on 'Choose entity' to add it.

Next, click on the 3 dots in the right corner of the 'Input number Set' window. Then click on 'Edit in YAML'.





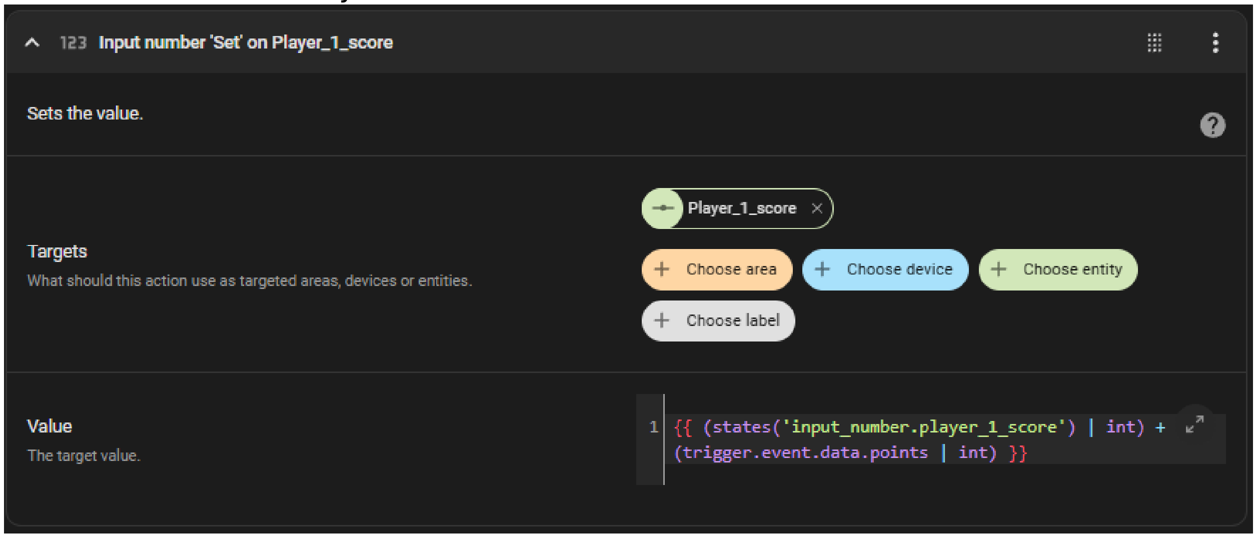
```
action: input_number.set_value
metadata: {}
data:
  value: >-
    {{ (states('input_number.player_1_score') | int) +
    (trigger.event.data.points | int) }}
target:
  entity_id: input_number.player_1_score
```

The value is:

>-

{{ (states('input_number.player_1_score') | int) + (trigger.event.data.points | int) }}

This weird looking code adds the target points to the player that hit the target. When you go back to the visual editor you should now see this:



Sets the value.

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

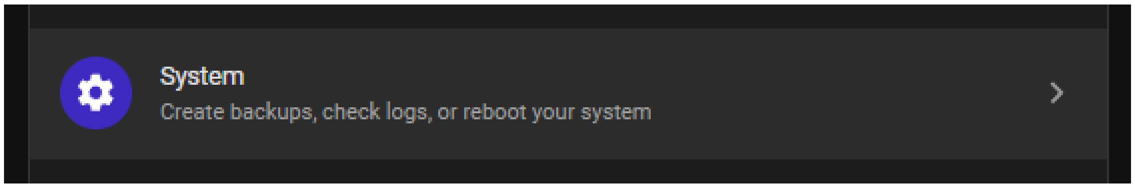
Value
The target value.

```
{{ (states('input_number.player_1_score') | int) +
(trigger.event.data.points | int) }}
```

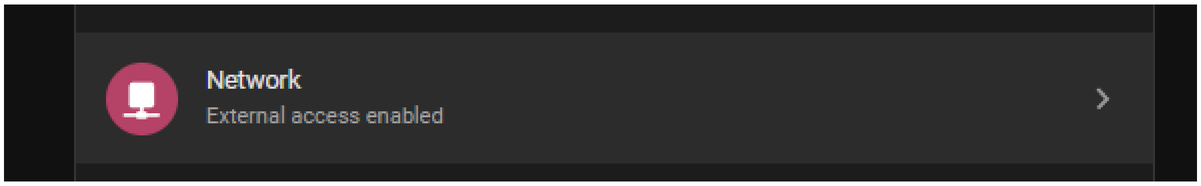
## Remote Connection

The last thing to cover is the option for remote administration. This is a service that costs about $60 a year but is great value if you need to debug a system remotely.

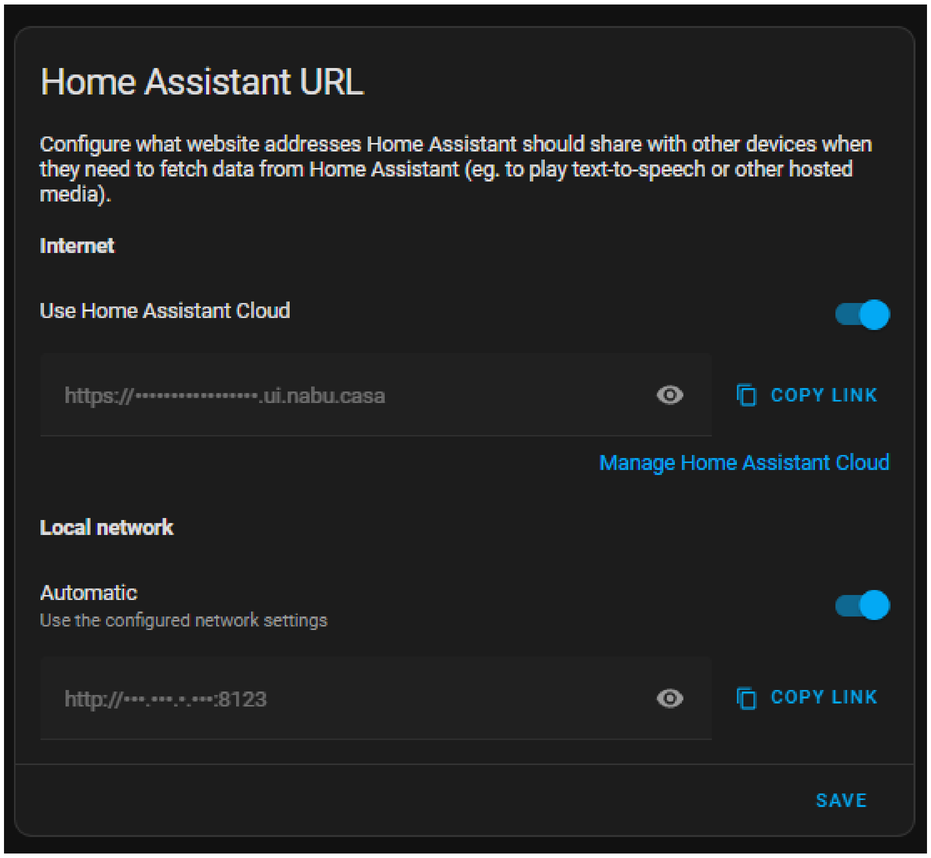
Got to settings then 'System'.



Then go to 'Network'



Then find the 'Home Assistant URL' section.



Home Assistant URL

Configure what website addresses Home Assistant should share with other devices when they need to fetch data from Home Assistant (eg. to play text-to-speech or other hosted media).

**Internet**

Use Home Assistant Cloud

https://••••••••••••••••.ui.nabu.casa

**Local network**

Automatic
Use the configured network settings

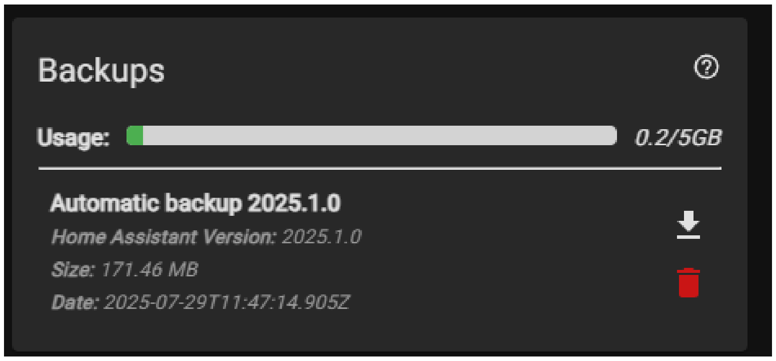
http://•••.•••.•.•••:8123

To activate your account, click on the 'Manage Home Assistant Cloud'. After your account is created you can access your Home Assistant server remotely. Super easy.

You can create your account here: https://www.nabucasa.com/

You can access your server remotely here: https://account.nabucasa.com/

After you log in there will be a link to log into your server. Backups are also stored on this remote service that you can download and restore a new server.



Backups

Usage: 0.2/5GB

Automatic backup 2025.1.0
*Home Assistant Version: 2025.1.0*
*Size: 171.46 MB*
*Date: 2025-07-29T11:47:14.905Z*

## Example Programs

The following are short examples on how to integrate different props using the targets.

### Power Conservation

This is an example on how we can use the Auxiliary Power to shut down a servo when not in use.

<Place example here>

### Wiper Motor Control

This is an example of how we can use the internal relay to control a wiper motor that is common for many different props. We can also implement the park mechanism on the wiper motor.

<Place example here>

### DC motor control.

This example will show how to connect a small DC motor with either 5V or 12V power to operate a spinner using the Ground Switch.

<Place example here>

### Servo Actuation

In this example we will trigger movement when the target is hit when the servo controller. We will also show how to use a high torque servo for prop operation.

<Place example here>

### LED Strip 2

In this example we will trigger a fuse effect on an LED strip to simulate the burning of a fuse to TNT. We can also trigger a spotlight for the explosion.

<Place example here>

## GPIO Input

In this example we will use a linear actuator with limit switches to detect when the actuator is extended. Then we will retract the actuator after a set amount of time.

<Place example here>