

SMART TARGET

USER MANUAL



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.

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

- Built in mechanical relay for external device control
- Servo output for mechanical actuation
- LED control for visual feedback
- Power outputs for auxiliary devices
- RS232 port for serial communication
- GPIO inputs for external sensors or buttons
- Analog input for variable signal sensing
- Power passthrough capability
- Flexible power input options
- Low idle current consumption

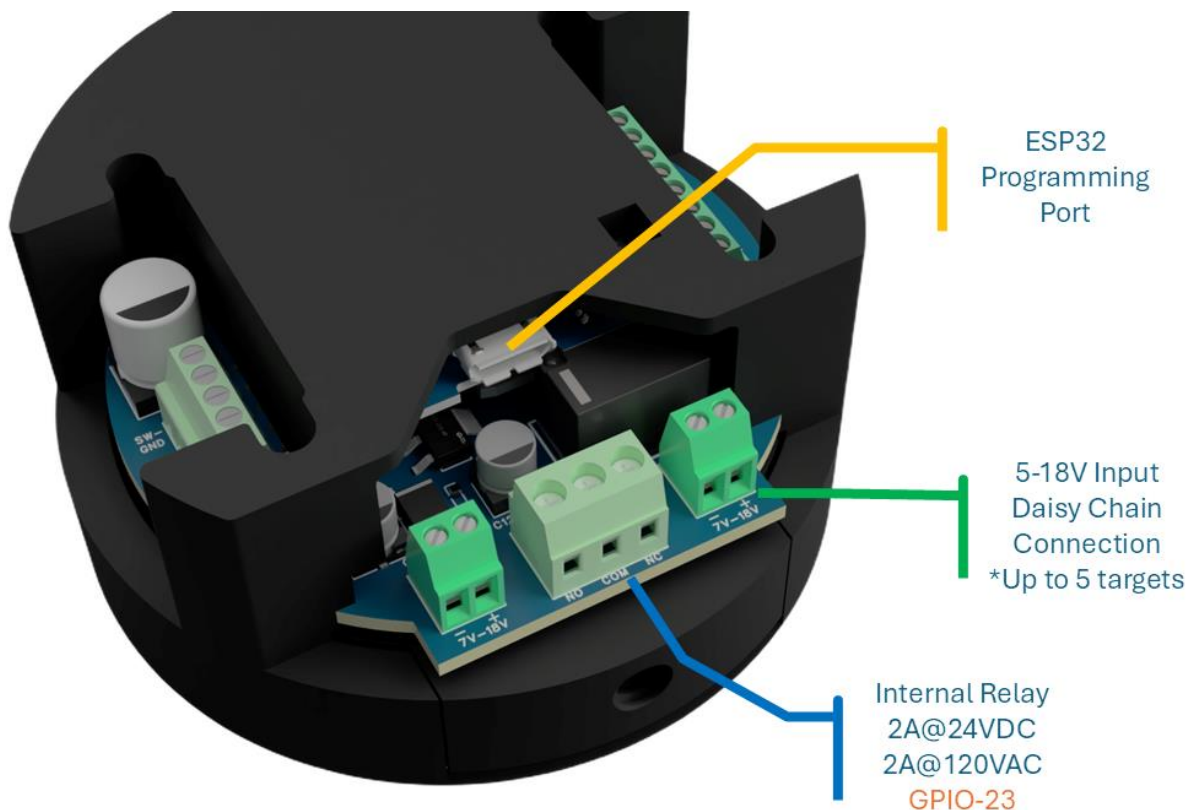
INTEGRATION & CONNECTIVITY

- 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

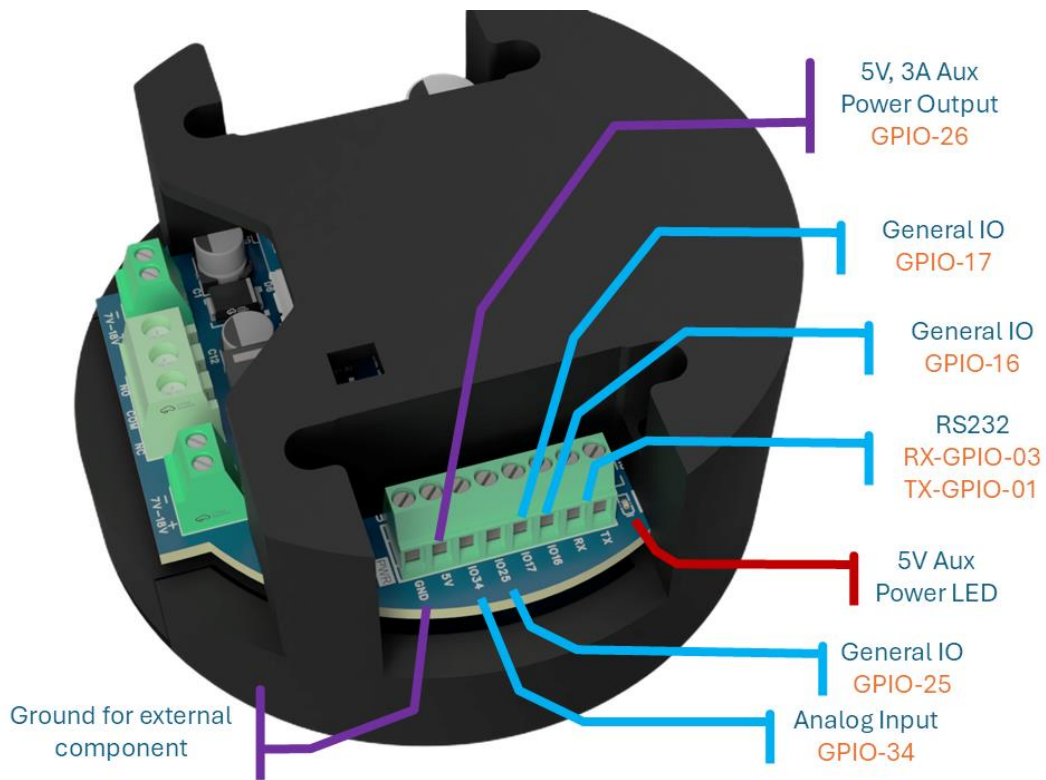
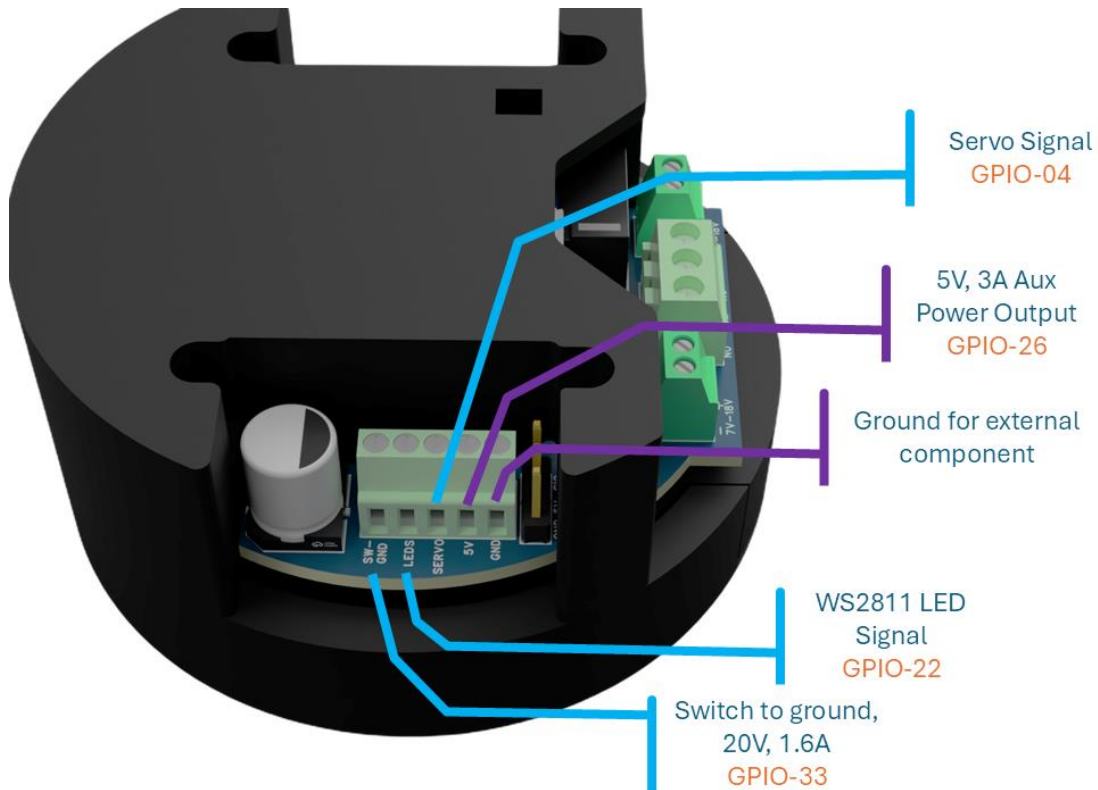
Electrical & Mechanical Specifications

Item	Specification
Power Input	5-18V DC
Relay Output	2A @ 24VDC / 2A @ 120VAC
Ground Switch Output	1.6A max
External 5V Outputs	2x 3A max
WS2811 Output	1x data-only pin
Servo Control	1x PWM output (50Hz, 3.3V logic)
Analog Input	0-3.3V ADC
GPIO Inputs	3x digital inputs
IR Receiver	NEC protocol
RS232	TX/RX headers

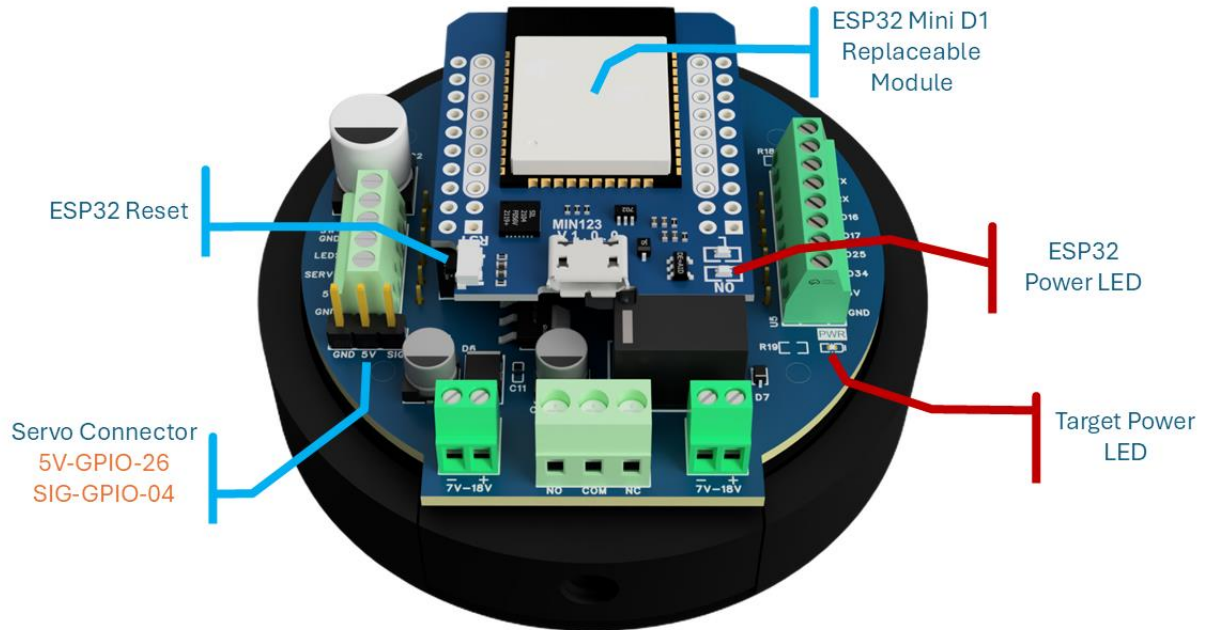
Inputs and outputs



Smart Target System
Model: SmartTargetV3.2



Smart Target System
Model: SmartTargetV3.2



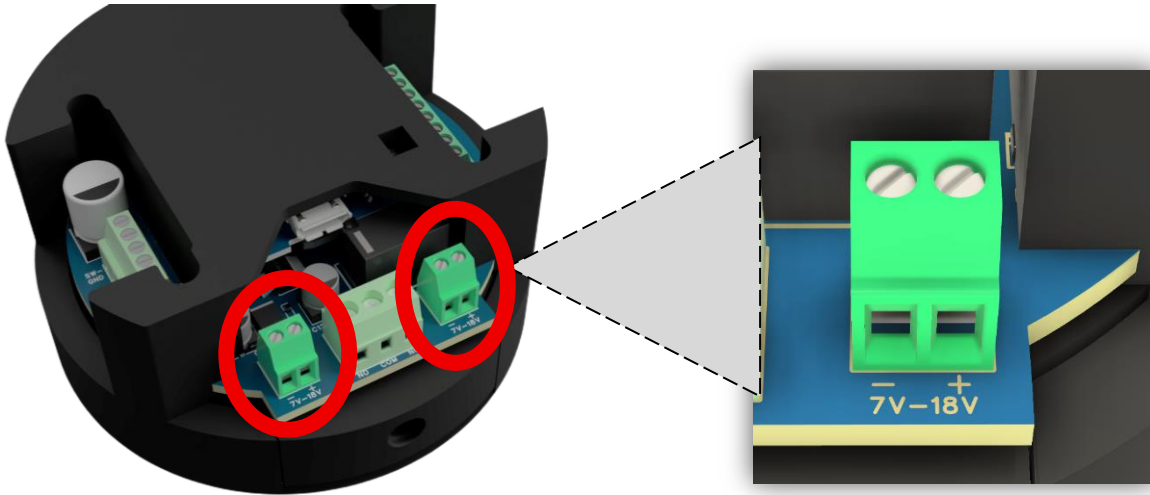
Target Mounting

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

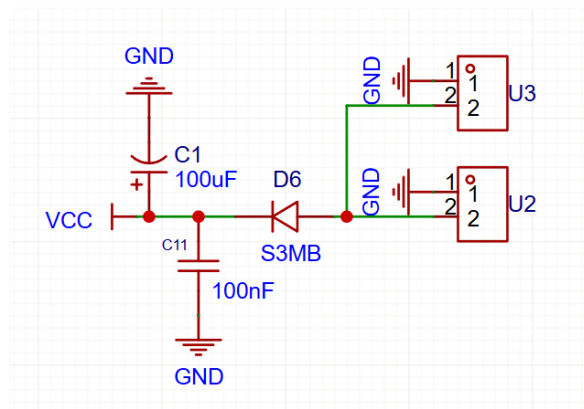
To power the target, connect 12V@5A power supply to either screw terminal circled below.



- - Ground input is on the left side of the terminal
- + Voltage is on the right side.

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

A 12V input is recommended; however, any voltage within the 7-18V 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 colourful 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'.

Smart Target System

Model: SmartTargetV3.2

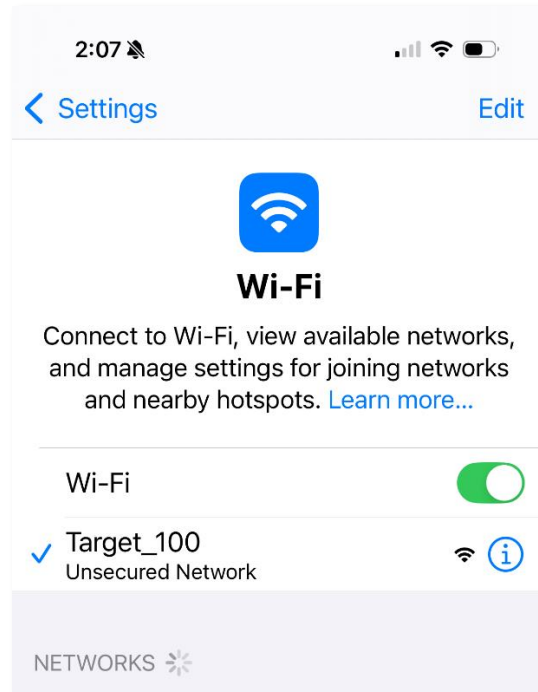


Figure 1: SSID Access Point Example

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**

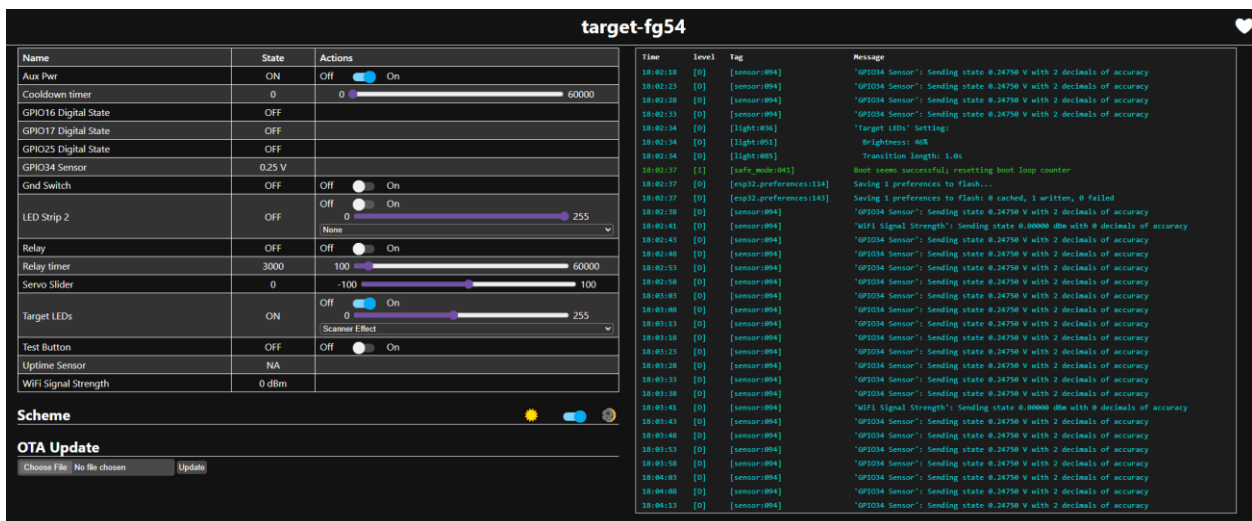


Figure 2: Target web page example

Smart Target System
Model: SmartTargetV3.2

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
Aux Pwr	The auxiliary power is separate from the ESP32 power. It powers the servo 5V, the 5V external power, and the internal relay.	ON or OFF	Yes
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 (1 minute max)	Yes
GPIO 16	Indicates the digital input state at this GPIO terminal	ON or OFF	No
GPIO 17	Indicates the digital input state at this GPIO terminal	ON or OFF	No
GPIO 25	Indicates the digital input state at this GPIO terminal	ON or OFF	No
GPIO 34	Indicates the analog input at this GPIO terminal	0.0-3.3V	No
Gnd Switch	FET switch to ground. Acts like a solid-state switch for external component power control.	ON or OFF	Yes
LED Strip 2	WS2811 and WS2812 control of external addressable LED strip.	ON or OFF Brightness Effect	Yes
LED Strip 2	Selectable solid colors for when	Dropdown selectable	Yes

Smart Target System
Model: SmartTargetV3.2

Solid Color	the effect is 'solid', 'scanner', or 'twinkle'		
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 (.1 second min) (1 minute max)	Yes
Servo Slider	Servo position control	-100 – 100 -100 = max left 0 = center 100 = max right	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 Brightness 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 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] 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] 'Relay' Turning ON.
19:13:33 [D] [switch:012]
```

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.

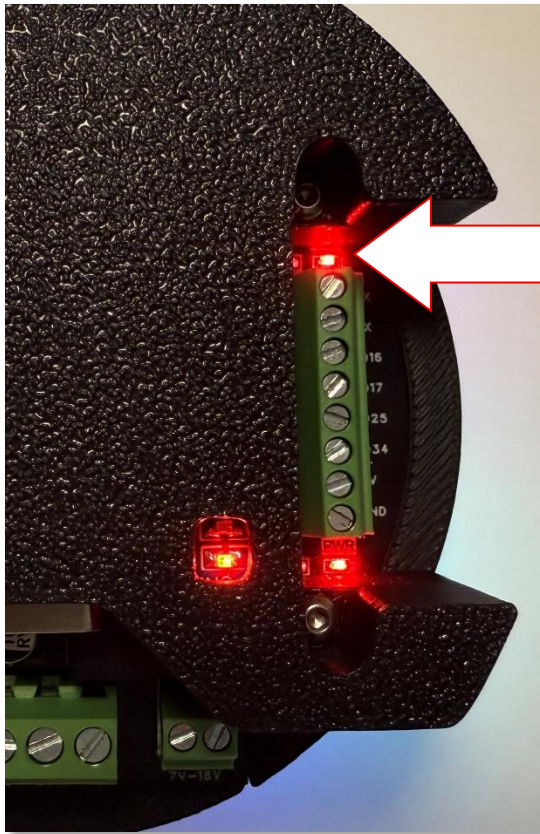
Cooldown timer	0	<input type="range" value="0"/>	60000
Relay timer	3000	<input type="range" value="100"/>	60000

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

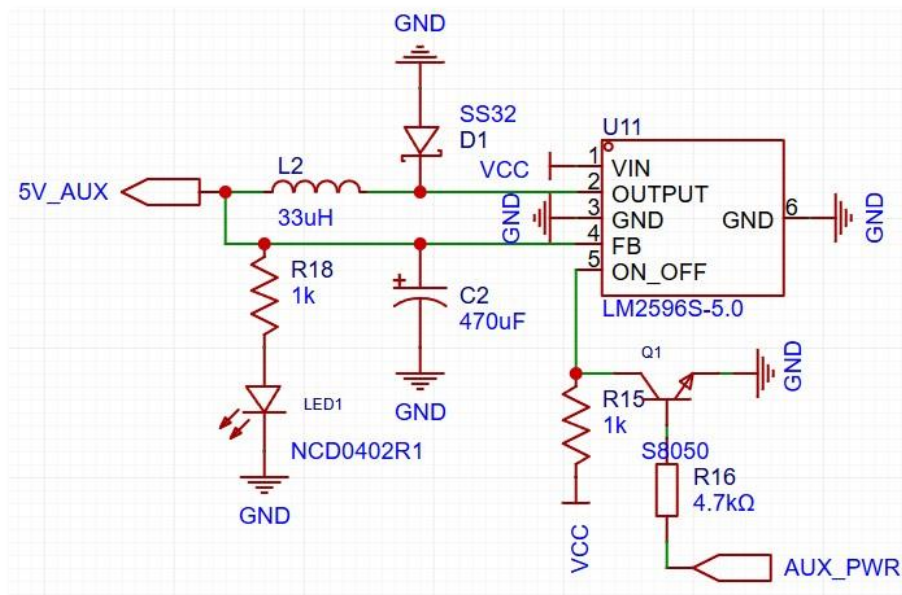
Aux Power

The servo power, 5V external power, and internal relay are all powered through an auxiliary power regulator. The ESP32, target display LEDs, IR receiver, and GND Switch are powered by a separate voltage regulator. The purpose of the auxiliary power is to allow power control of items that may need to be turned off to conserve energy. For example, the target can turn off the power going to the servo when not in use.

The Aux regulator has its own LED to indicate when it is on.



AUX Power LED (ON)



Schematic 2: AUX Power Circuit

The 5V regulator employed is the LM2596, which is capable of supplying an output current of up to 3A with high efficiency. For optimal performance, it is advisable that the input voltage supplied to the target device supports at least 5A. The LM2596 operates at a fixed

switching frequency of 150kHz, contributing to reduced output noise and enabling the use of relatively compact external components. The LM2596 features built-in thermal shutdown, current limit, and short-circuit protection, enhancing system reliability.

In this application, the LM2596 ensures stable and efficient power delivery to auxiliary loads such as servos and relays, while preventing overloading of the system's main 5V regulator. When the Aux regulator (LED indicator illuminated) is switched on, the LM2596 provides the necessary voltage and current to its designated components, supporting both steady operation and energy-saving strategies during idle periods.

To test, turn the 'Aux Pwr' ON. It is on by default and turns on automatically on startup.



You should be able to measure 5V at the screw terminals, otherwise the LED and power output will be OFF and the measurement will be 0V.

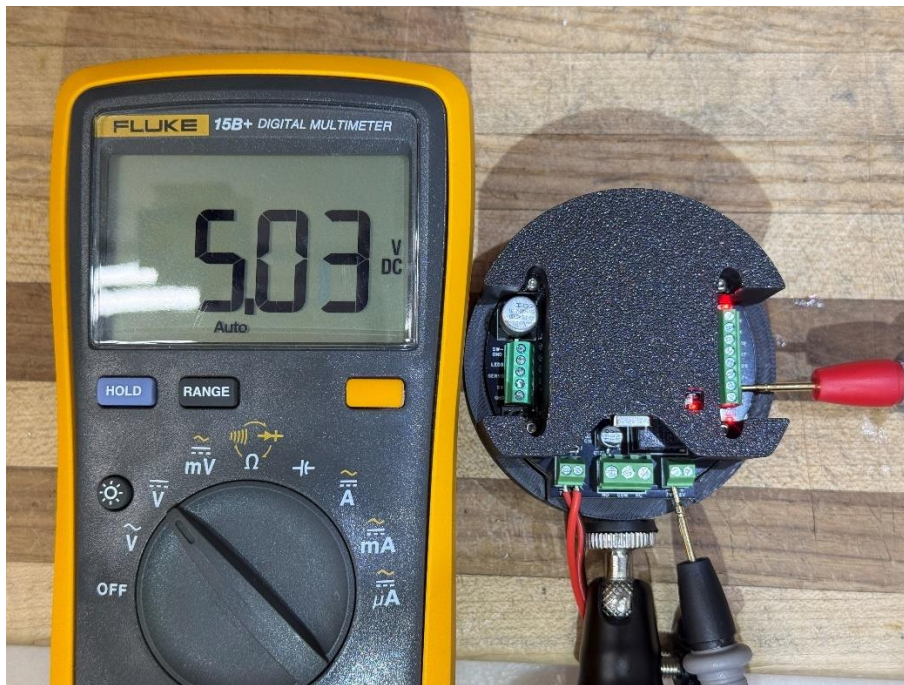


Figure 3: 5v Aux Pwr Measurement

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 GPIO16,17, and 25 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.

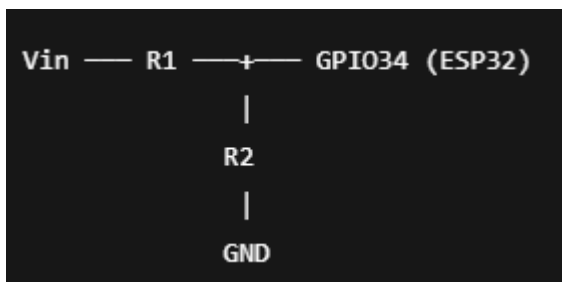
The purpose of these GPIO is to connect a switch to ground. An example of a switch would be a pressure switch that is grounded when stepped on.

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 3.2 target.

There is one analog input 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.

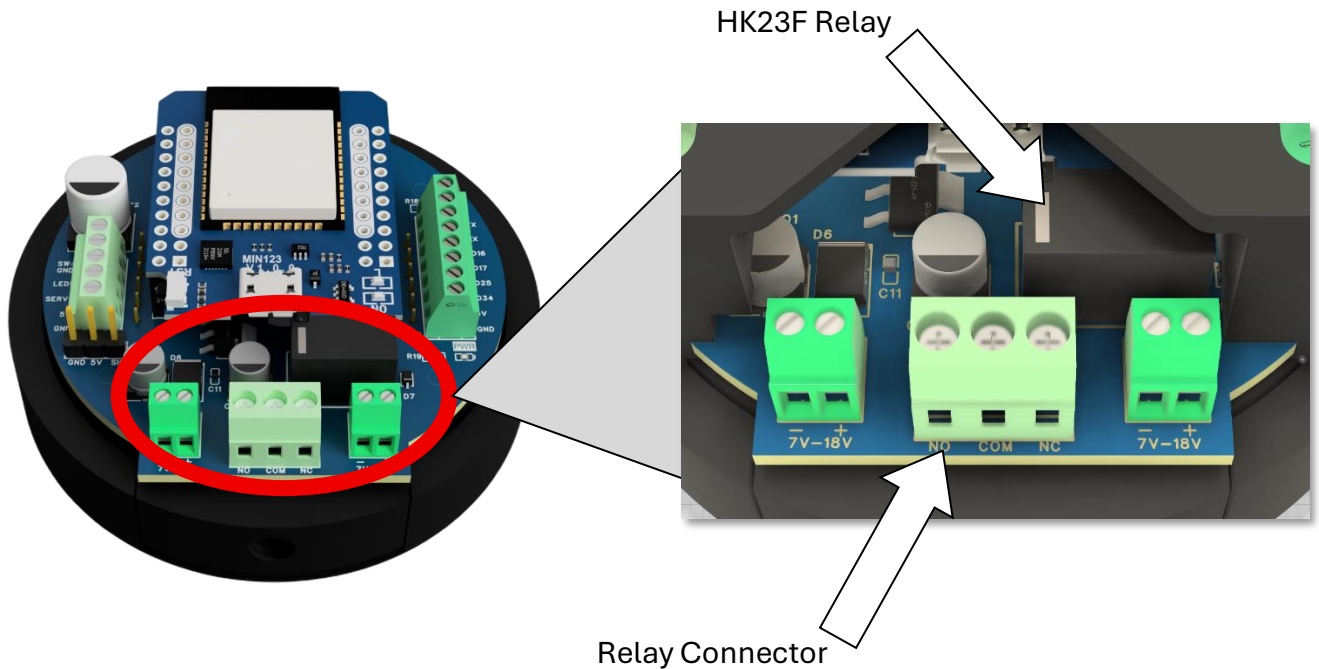


Sensor Voltage	R1	R2	Max Reading
12V	15k Ω	5.6k Ω	3.27V
10V	6.8k Ω	3.3k Ω	3.3V
5V	3.3K Ω	2.2k Ω	3.0V

***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

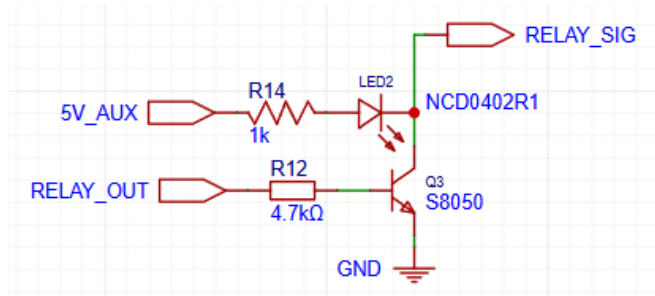
The internal relay will switch from the normally closed position to the normally open when the target is hit. The relay is a HK23F and is the little black box pictured below. It can handle up to 2A @ 24VDC or 2A @ 120VAC.



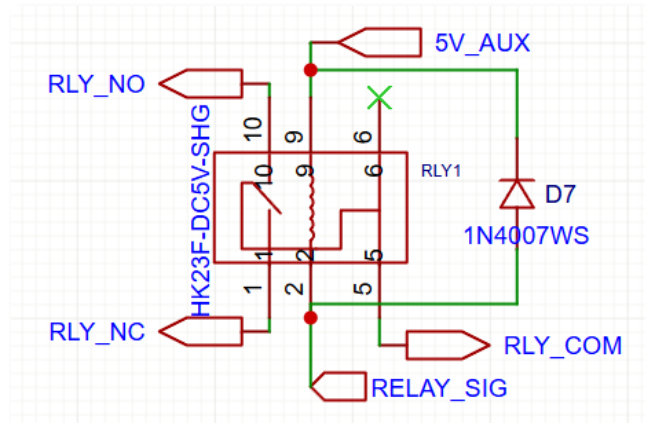
To test, turn the 'Relay' on.

Relay	ON	Off <input checked="" type="checkbox"/> On
-------	----	--

When the relay is ON, COM and NO will connect and the connection from COM to NC will open. The opposite is true when the relay is OFF.



Schematic 3: Relay Trigger Circuit



Schematic 4: Relay Circuit

You can see in the schematic that the 5V Auxiliary power must be on for the relay to work.

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



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.

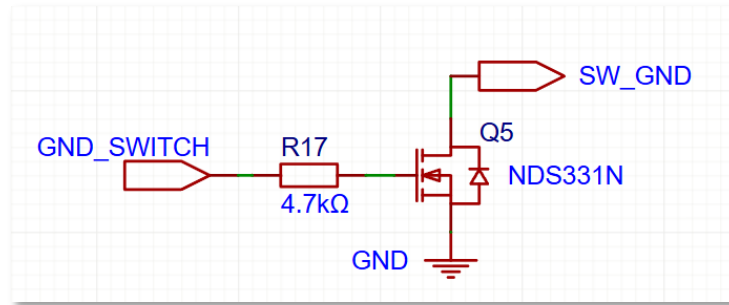


The default for the relay on time is 3 seconds with the minimum being 0.1 seconds and the maximum being 1 minute or 60,000 milliseconds. The slider value will be saved through power cycles and is used to set up prop motion or external triggers.

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.

Ground Switch

By default, the 'SW-GND' will turn on when the relay turns on for the standalone application. The 'SW-GND' output is a NDS331N N-Channel FET that can handle a continuous 20V @1.3A when ON. This output allows you to turn on devices and external components that are powered by the target's auxiliary 5V power or voltage input power supply. It can also be used to trigger an external relay if more amperage is required than the internal relay can safely handle.



Schematic 5: SW-GND Circuit

This ground switch **will** operate when the 5V Auxiliary power is ON or OFF.

To test, turn the 'Gnd Switch' ON and measure the voltage between the 'SW-GND' terminal and the input voltage terminal. You should read the input voltage value.

Gnd Switch	ON	Off <input checked="" type="checkbox"/> On
------------	----	--

Next, turn the 'Gnd Switch' OFF.

Gnd Switch	ON	Off <input type="checkbox"/> On
------------	----	---------------------------------

The measurement should now read 0.0V or close to zero.

Servo

There is a standard servo connector that can power most servos. This connector's 5V source is from the 5V Auxiliary Power so the 'Aux Pwr' must be on to use the servo.

Aux Pwr	ON	Off <input checked="" type="checkbox"/> On
---------	----	--

The 'Aux Pwr' can be turned off to conserve power, if the servo gets hot, or if the servo is noisy during idle.

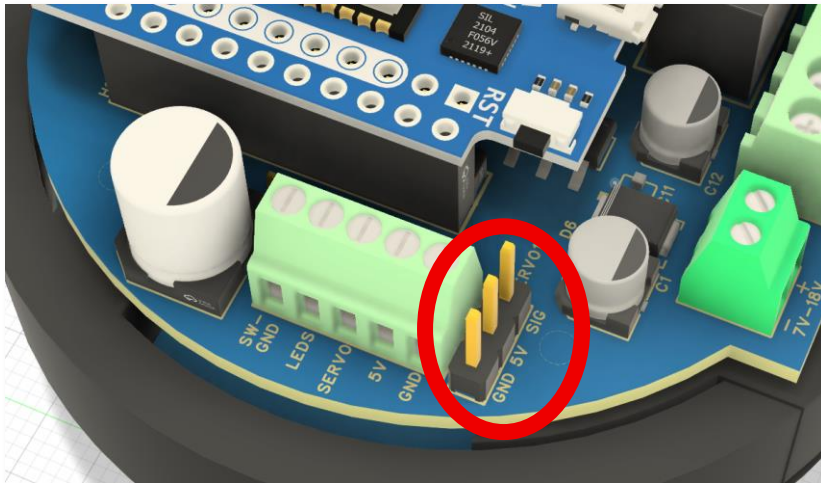


Figure 4: Servo Connector



Figure 5: Servo Connected Example

To test the servo, plug the servo into the target and adjust the slider control. The servo should move according to the slider position.



If your servo is very high torque and/or requires more than the 3A, then you can instead use the Servo terminal connection for the signal and power the servo separately.

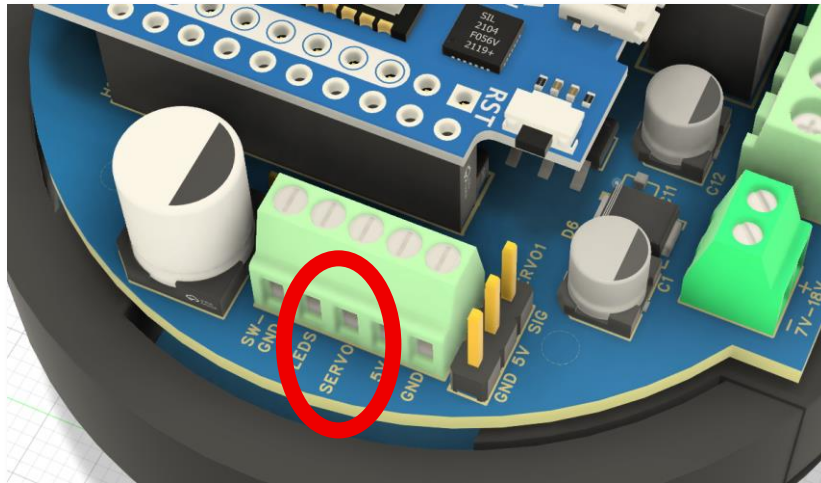


Figure 6: External Servo Signal

*Note: The 'Servo' terminal and the servo connector are the same signal. There are **not** two separate servo outputs, they are the same signal.

Target Face LEDs

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



Figure 7: Target RGB LEDs

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

Target LEDs	ON	Off <input checked="" type="checkbox"/> On 0 <input type="range"/> 255 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 <input checked="" type="checkbox"/> On 0 <input type="range"/> 255 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 labeled 'LEDS'.

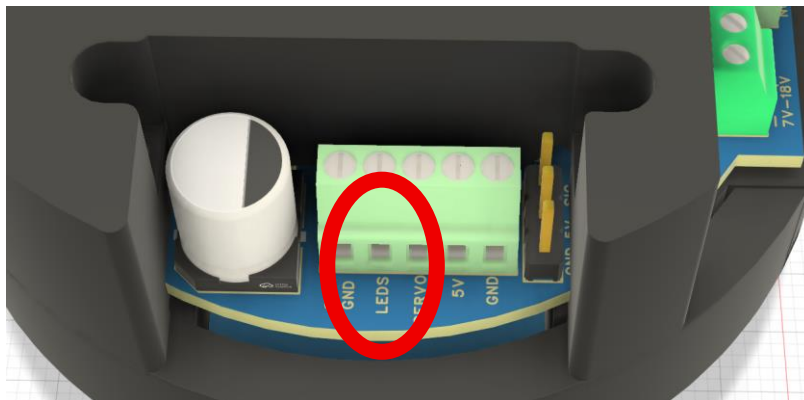


Figure 8: LED Strip 2

LED Strip 2	ON	Off <input checked="" type="checkbox"/> On 0 <input type="range"/> 255 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.

Custom Programming

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_rev3_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:

Smart Target System
Model: SmartTargetV3.2

```
===== [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 9: 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.

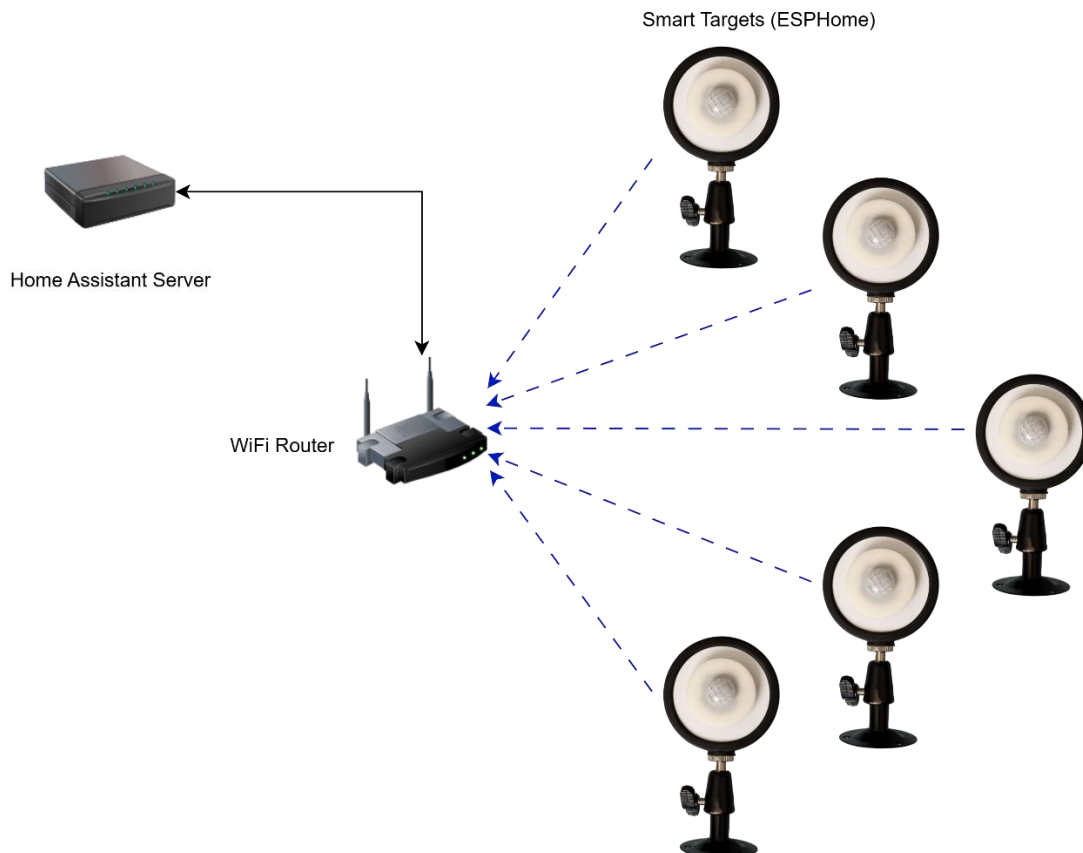


Figure 10: Smart Target Basic Network

Smart Target System

Model: SmartTargetV3.2

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.

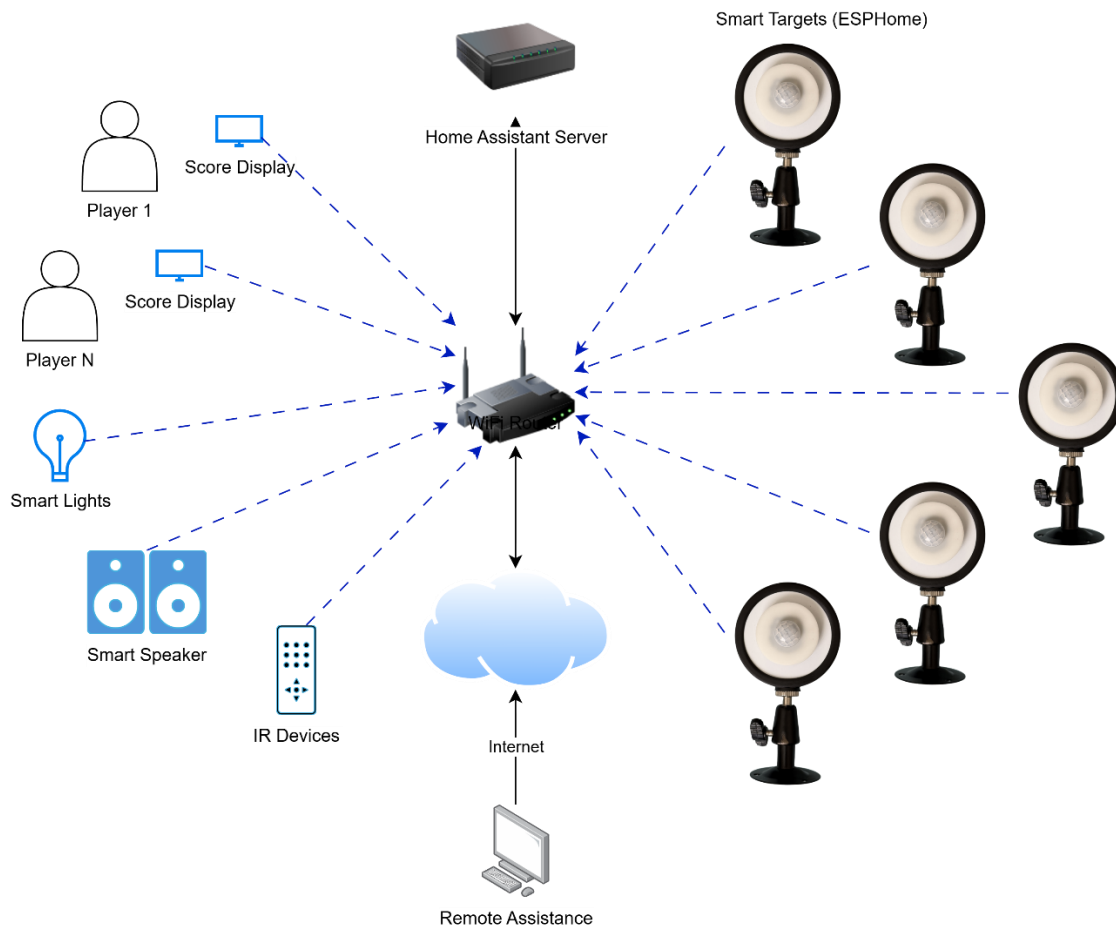


Figure 11: More complex network

The server can be a raspberry pi version 4 or 5 with at least 4GB 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 12: 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 13: 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/>

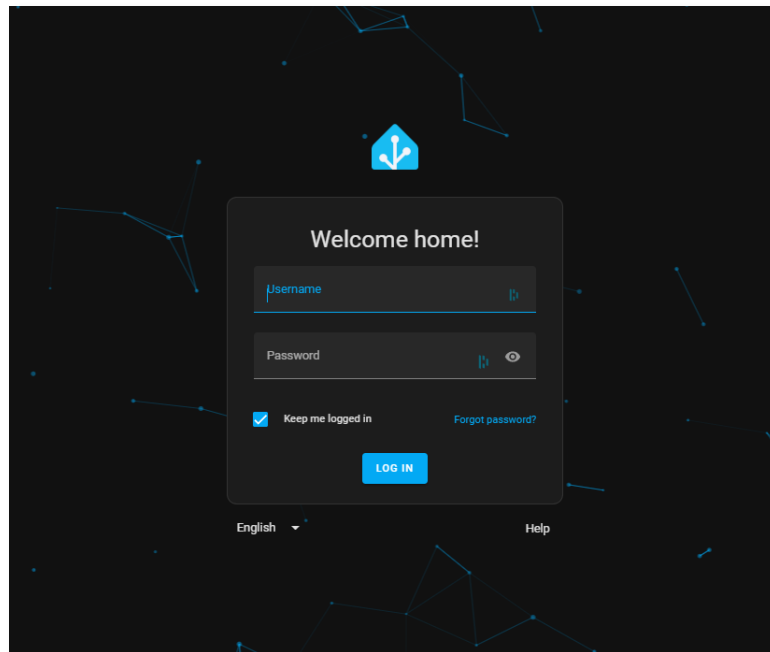


Figure 14: Home Assistant Login Page

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.

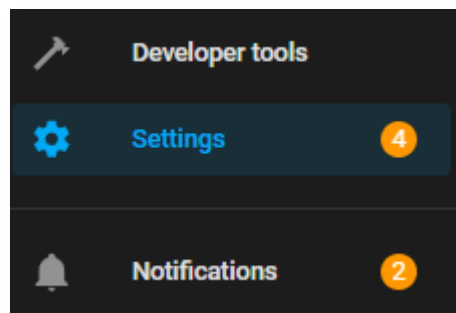


Figure 15: HA Settings

Next click on 'Add-ons'

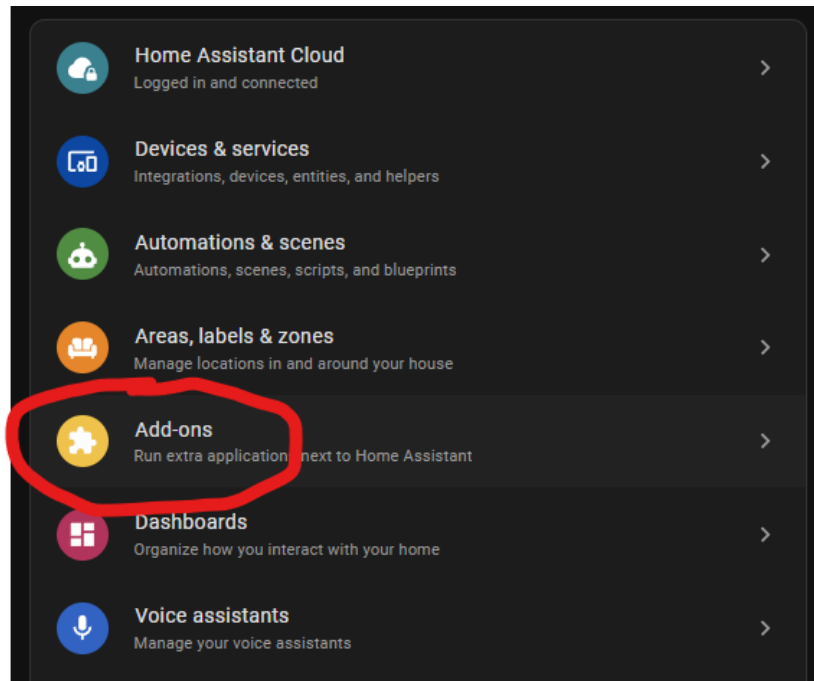


Figure 16: HA Add-ons

Then search for 'ESPHome' and install it.

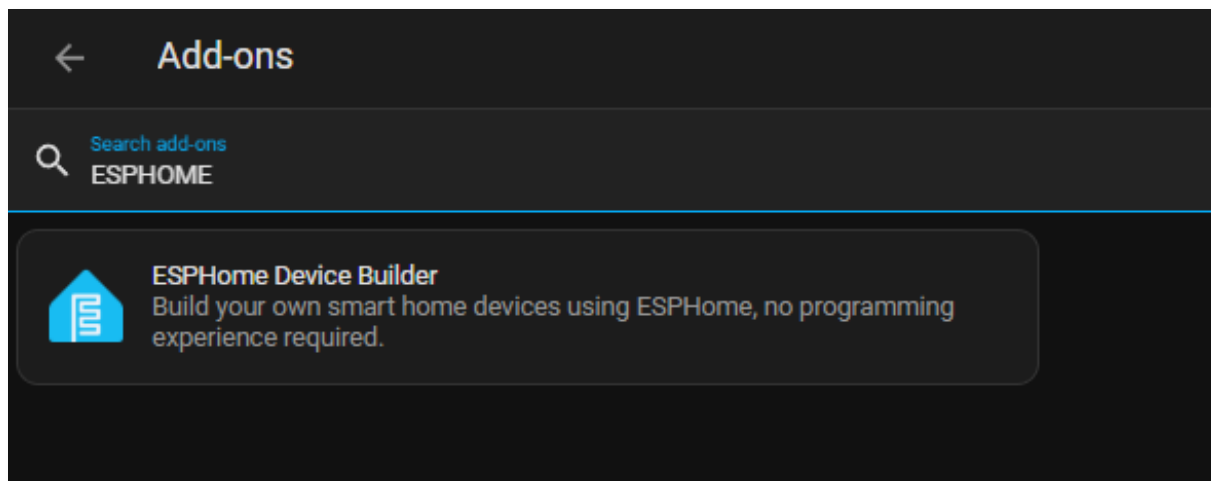


Figure 17: HA Install ESPHome

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

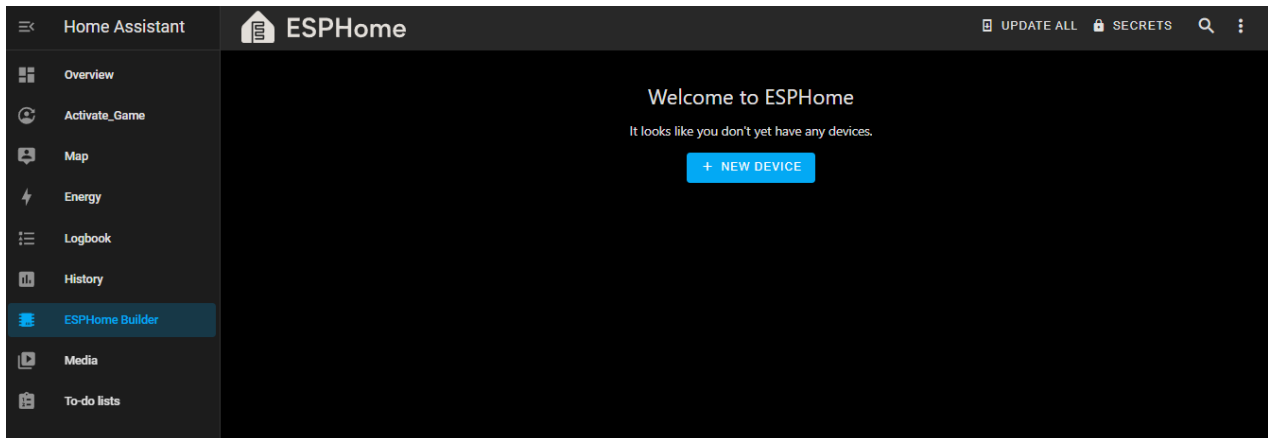


Figure 18: ESPHome

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.

```
35  # =====
36  # EXTERNAL PACKAGES
37  # =====
38  packages:
39      color_controls: !include color_controls.yaml
40
41      # Change this manually when changing target_type:
42      # - For standalone: uncomment the next line
43      #config_file: !include target_standalone_config.yaml
44      # - For networked: uncomment the next line
45      config_file: !include target_home_assistant_config.yaml
46
```

Smart Target System

Model: SmartTargetV3.2

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.

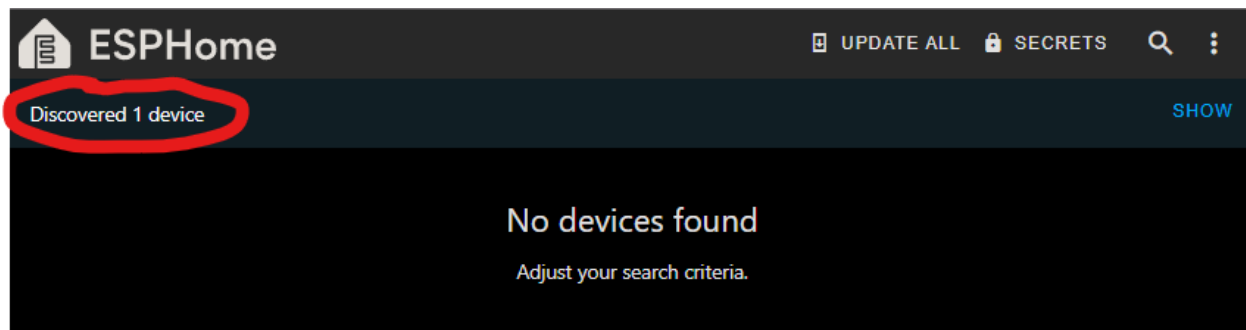
Now program the target, after connecting it to your machine with a USB cable, using the following command.

```
esphome -s ip 2 run .\target_rev3_x.yaml
```

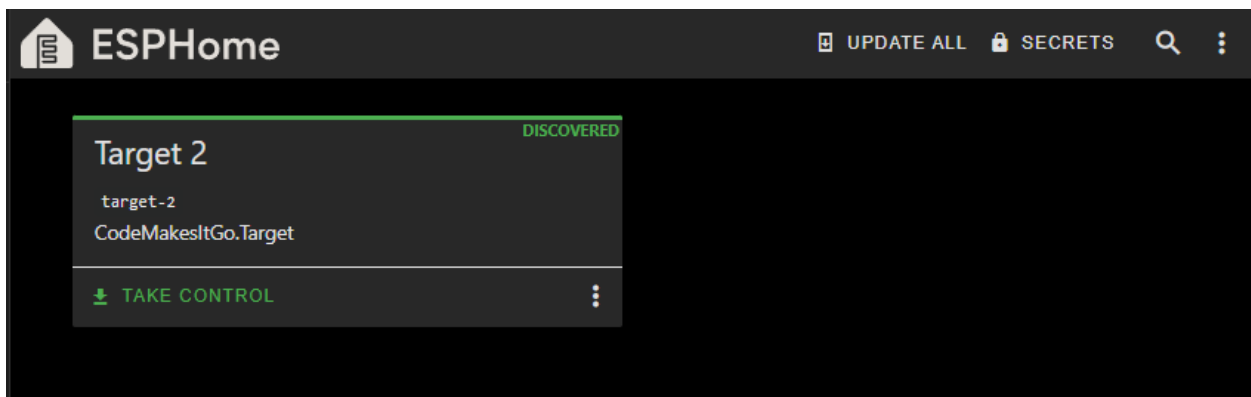
This will set the target's IP address to 192.168.8.2. It is best to use static addresses for better reliability.

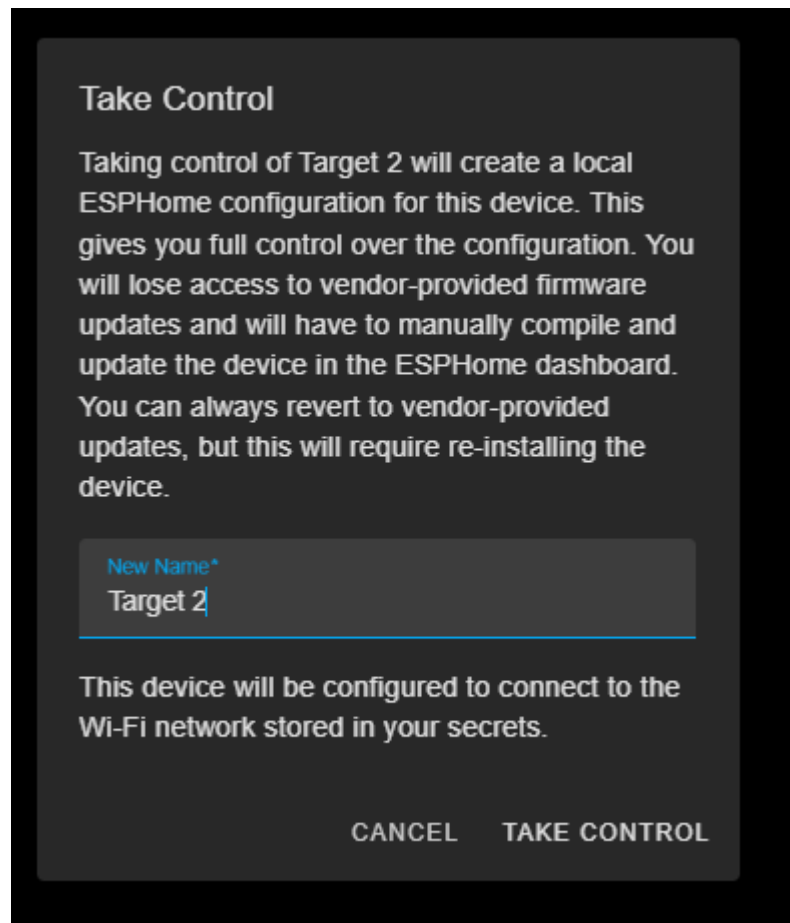
*Warning: Be sure to set each target's IP address differently.

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



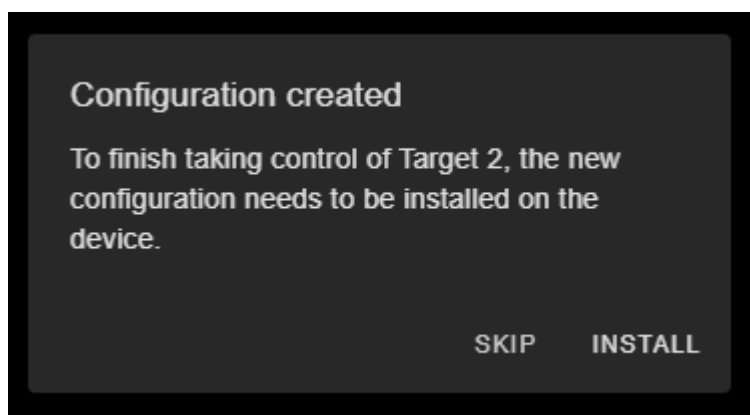
Click on the 'show' text on the right side of the screen. You will see a box with the target name as a discovered device. Select 'Take Control'





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.

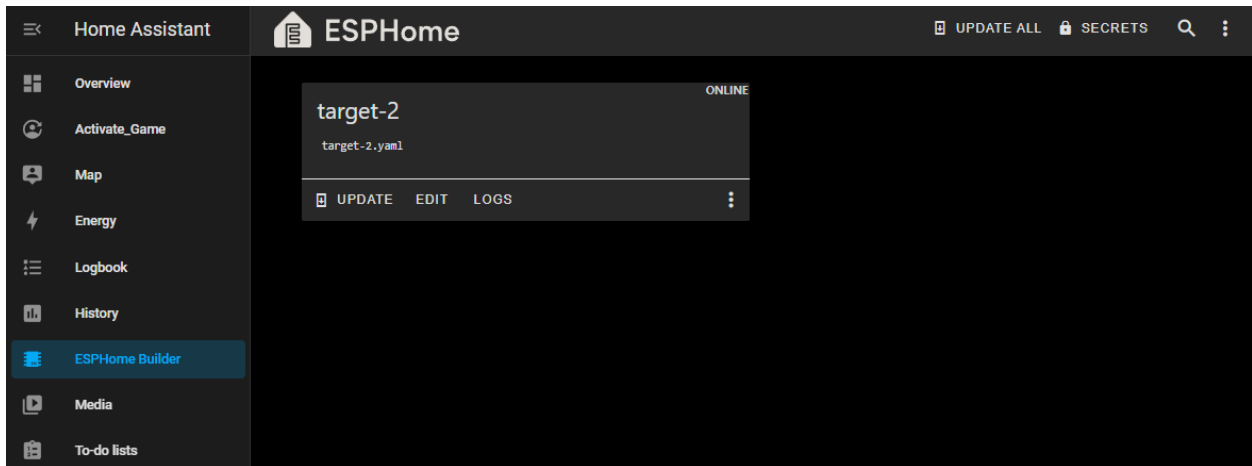
Another dialog box will appear, select 'skip' because we have already programmed the target.



Smart Target System

Model: SmartTargetV3.2

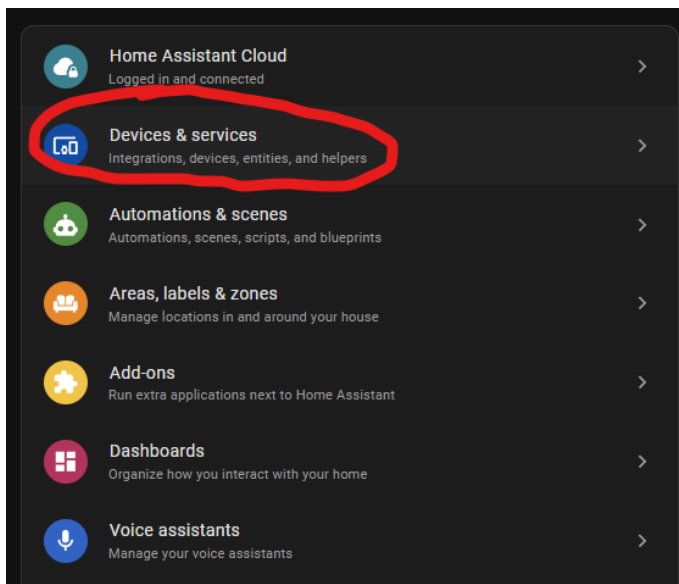
After this step, the 'ESPHome Builder' tab should display the target as being online.



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'

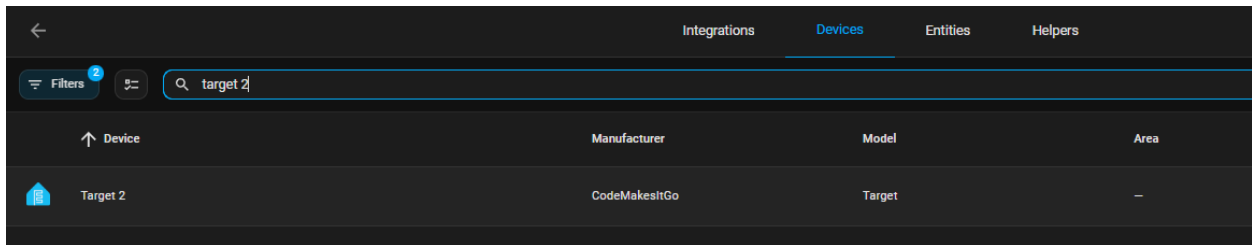


Then select 'Devices' on the top menu bar. This will display all your devices connected to the Home Assistant.

Smart Target System

Model: SmartTargetV3.2

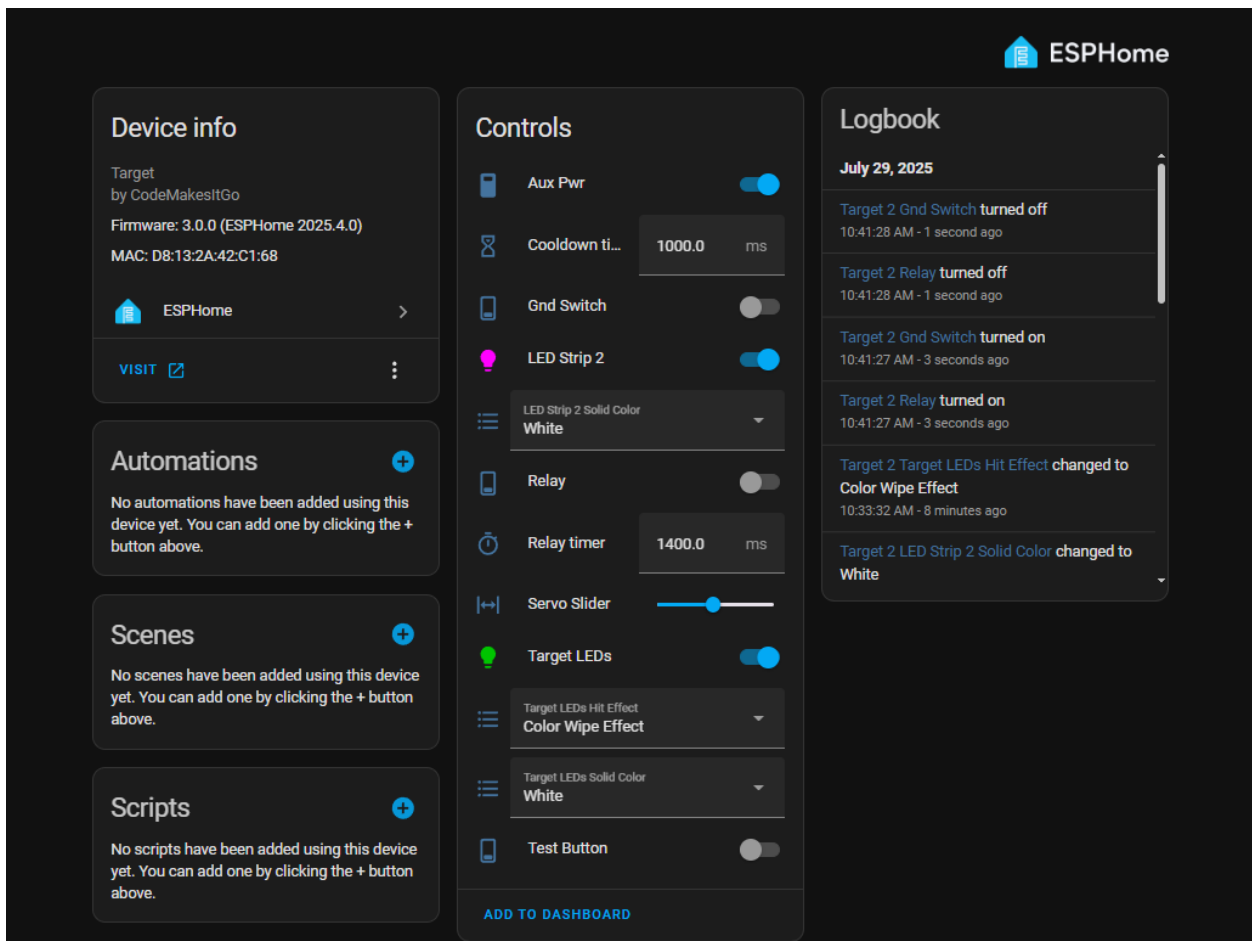
Search and select the name of the target you just programmed.



The screenshot shows the ESPHome web interface. At the top, there are tabs for 'Integrations', 'Devices', 'Entities', and 'Helpers'. The 'Devices' tab is selected. Below the tabs, there is a search bar with the text 'target 2'. To the left of the search bar, there are icons for 'Filters' and 'Settings'. Below the search bar, there is a table with the following columns: 'Device', 'Manufacturer', 'Model', and 'Area'. The table contains one row with the following data: 'Target 2', 'CodeMakesItGo', 'Target', and '-'.

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.



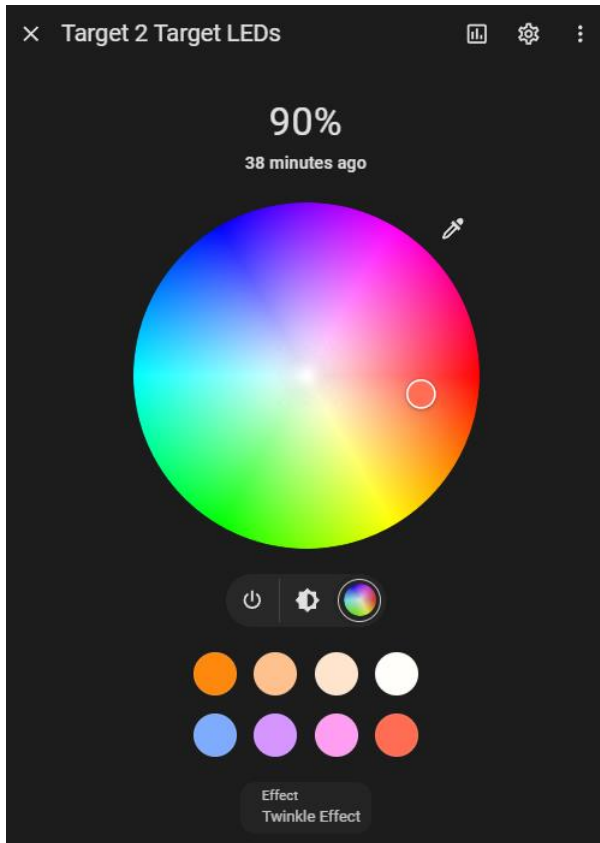
The screenshot shows the ESPHome web interface for a device named 'Target 2'. The interface is divided into several sections: 'Device info', 'Automations', 'Scenes', 'Scripts', 'Controls', and 'Logbook'. The 'Device info' section shows the device name, manufacturer, firmware version, and MAC address. The 'Automations', 'Scenes', and 'Scripts' sections show that no automations, scenes, or scripts have been added yet. The 'Controls' section shows various controls for the device, including 'Aux Pwr', 'Cooldown ti...', 'Gnd Switch', 'LED Strip 2', 'Relay', 'Relay timer', 'Servo Slider', 'Target LEDs', and 'Test Button'. The 'Logbook' section shows a list of events, including 'Target 2 Gnd Switch turned off', 'Target 2 Relay turned off', 'Target 2 Gnd Switch turned on', 'Target 2 Relay turned on', 'Target 2 Target LEDs Hit Effect changed to Color Wipe Effect', and 'Target 2 LED Strip 2 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.

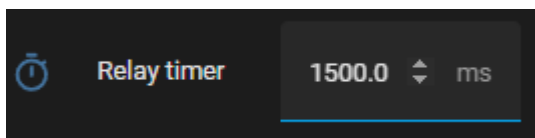


Smart Target System
Model: SmartTargetV3.2

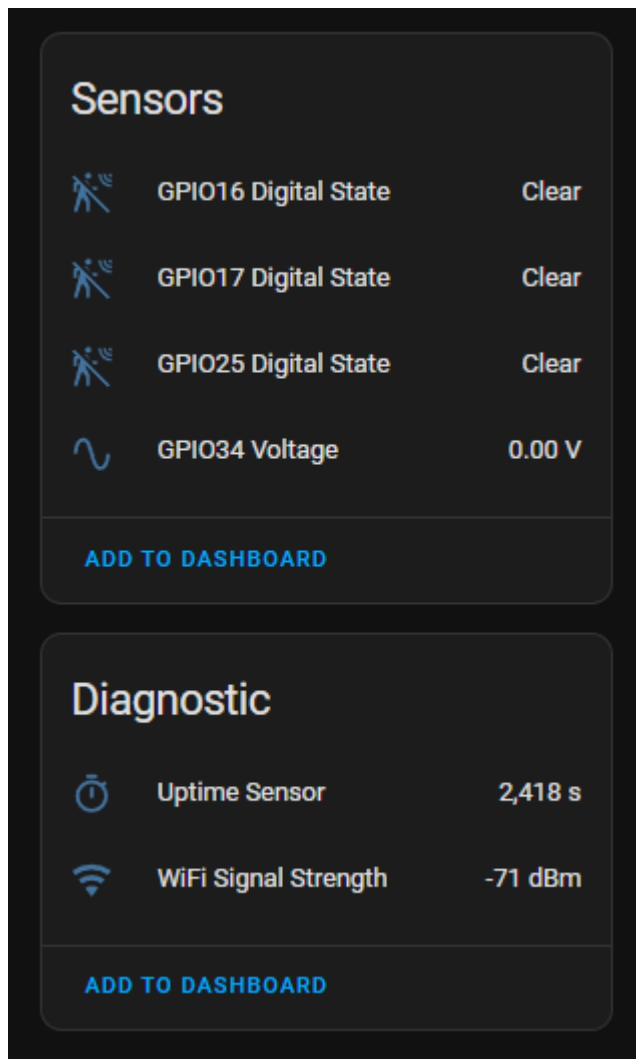
The device page in Home Assistant differs slightly from the one on the target's webpage; for instance, LED color controls are more detailed.



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



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



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.

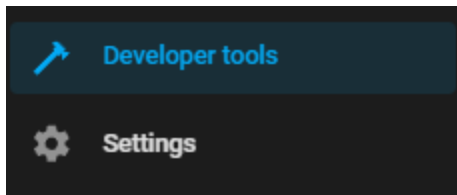
Signal Strength (dBm)	Quality	Meaning
-30 dBm	Amazing	Maximum strength, ideal if next to the router
-50 dBm	Excellent	Strong signal, fast and reliable connection
-60 dBm	Very Good	Reliable for most uses including streaming/video calls

-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

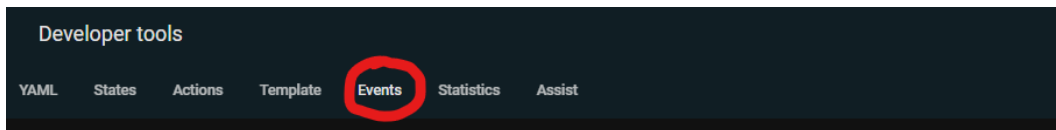
Listening to events

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.

Smart Target System
Model: SmartTargetV3.2

Developer tools

YAML States Actions Template **Events** Statistics Assist

Fire an event on the event bus. [Events documentation](#)

Event type*

Event data (YAML, optional)

1

FIRE EVENT

Listen to events

Event to subscribe to
esphome.target-hit

START LISTENING CLEAR EVENTS

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

Listen to events

Listening to
esphome.target-hit

STOP LISTENING CLEAR EVENTS

Event 0 fired 2:36 PM:

```
1 event_type: esphome.target-hit
2 data:
3   device_id: 047b40bd0a10c36e98895c690948ff54
4   address: "2"
5   player: "71"
6   points: "10"
7   origin: LOCAL
8   time_fired: "2025-07-29T20:36:23.748209+00:00"
9   context:
10    id: 01K1BYD324TD41PWKT5WPK4QB9
11    parent_id: null
12    user_id: null
13
```

Under the 'data' section you can see the following information:

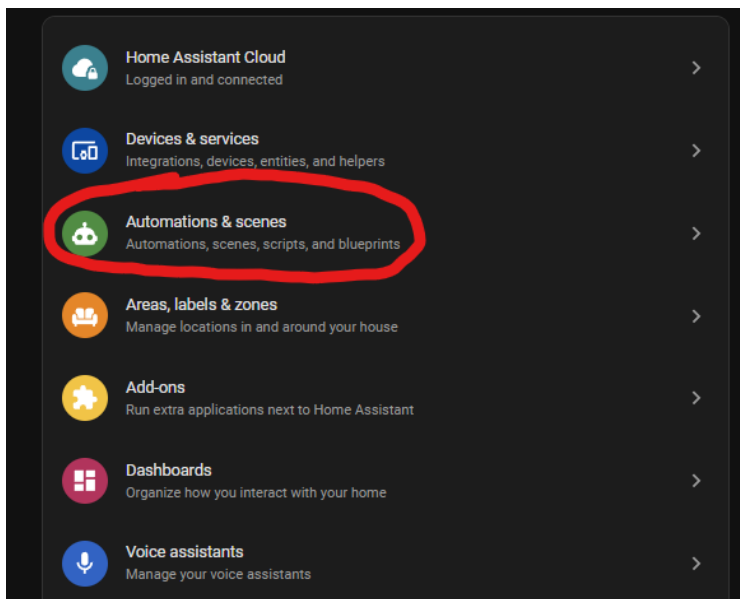
Data Item	Description
Address	The ID of the target reporting the hit.
Player	The command from the IR signal. This should be coded so that you know which player hit the target
Points	The configured points are awarded to the player for hitting this target. This can be changed on each target with the 'default_points'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

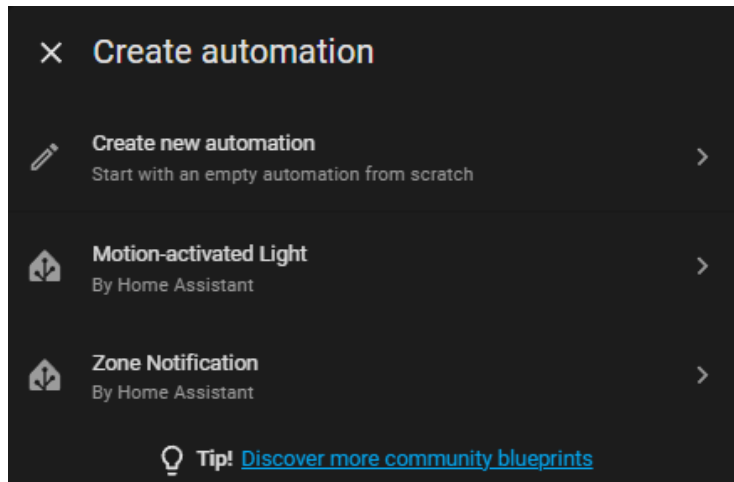


On the bottom right side of the screen select '+ Create Automation'.

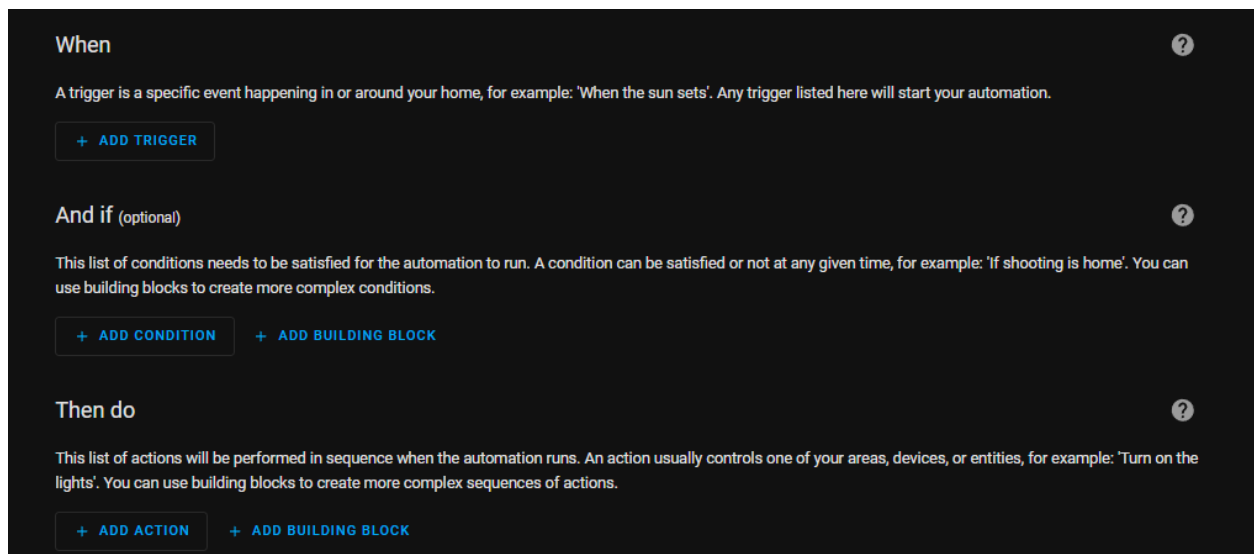
A dialog will appear for creating an automation. Select 'Create new automation'

Smart Target System

Model: SmartTargetV3.2

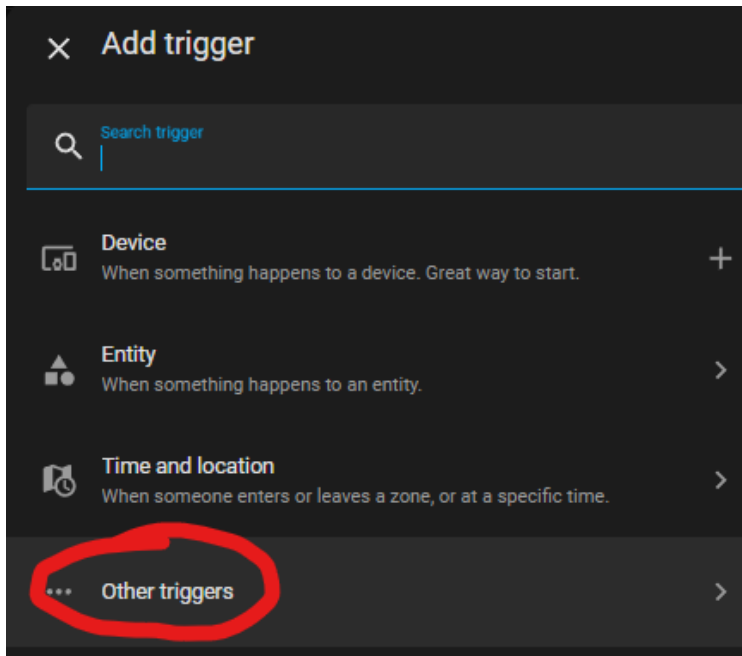


Automations are based on events. The next screen helps you describe the event graphically and what to do when the event happens. Under the 'When' section select '+ Add Trigger'.

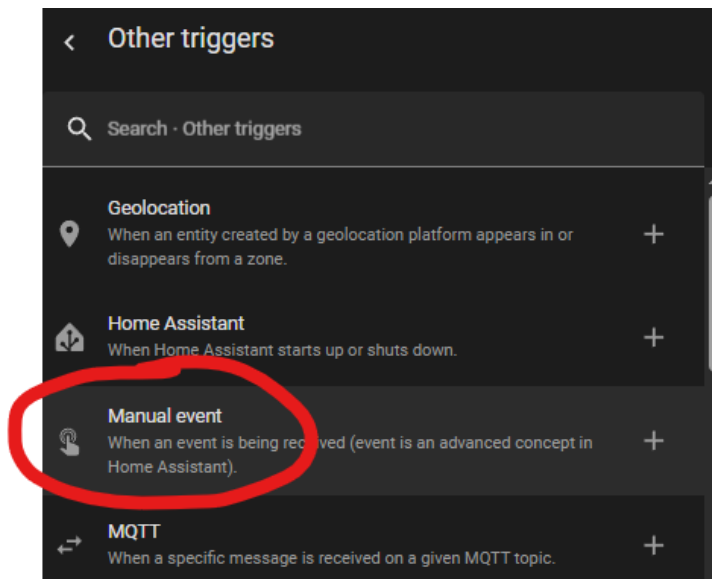


Then in the next dialog, select '... Other triggers'

Smart Target System
Model: SmartTargetV3.2



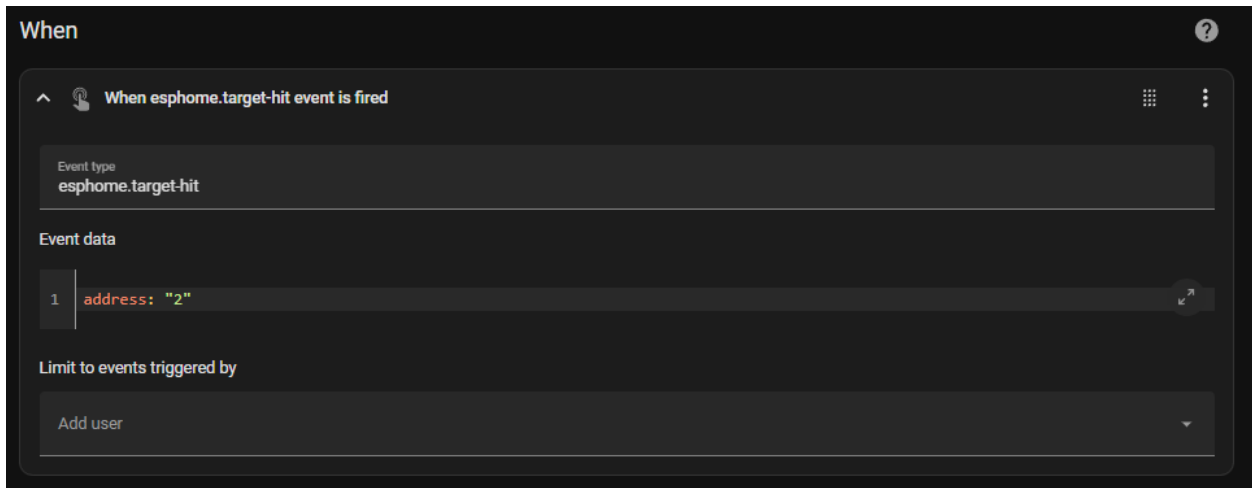
Next select 'Manual event'



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

Smart Target System

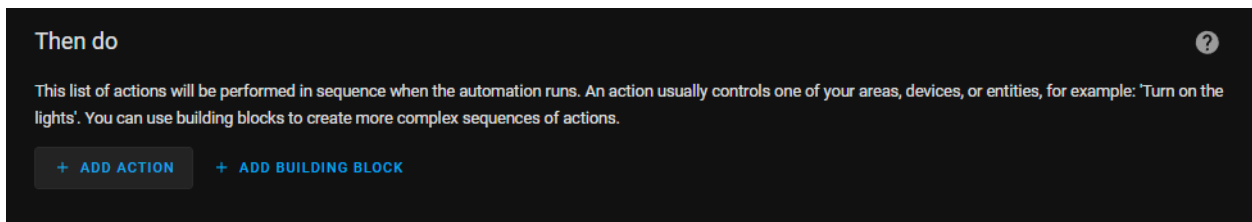
Model: SmartTargetV3.2



The 'When' section is titled 'When' with a help icon. It contains a trigger event: 'When esphome.target-hit event is fired'. Below this, the 'Event type' is set to 'esphome.target-hit'. The 'Event data' section shows a list with one item: '1 address: "2"'. At the bottom, there is a 'Limit to events triggered by' section with a dropdown menu currently showing 'Add user'.

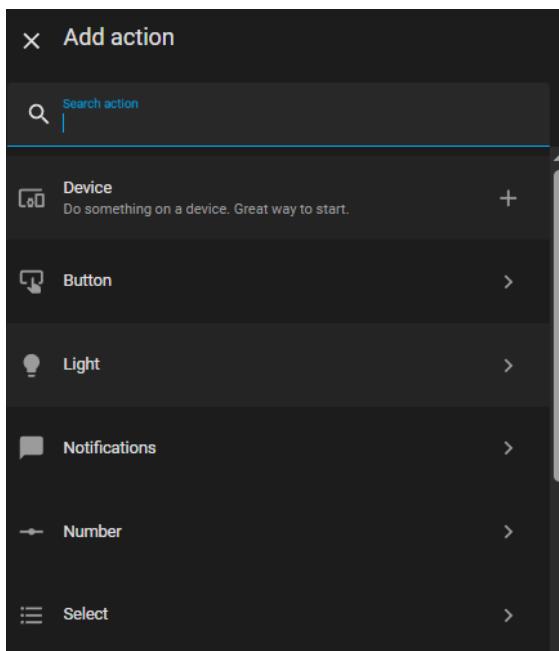
Now that the event trigger is set, we must set an action.

Under the 'Then do' section select the '+ Add Action'.



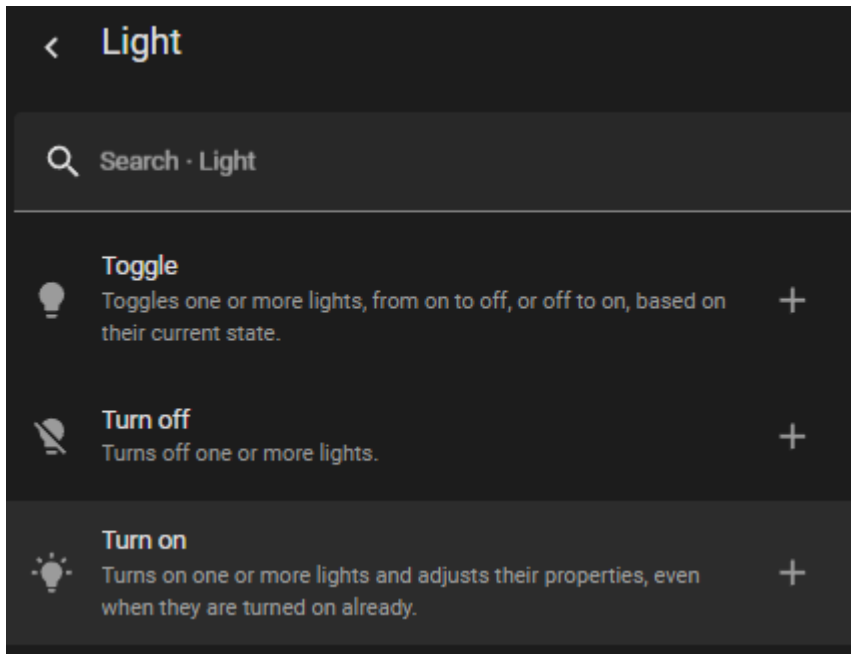
The 'Then do' section is titled 'Then do' with a help icon. It contains a descriptive text: 'This list of actions will be performed in sequence when the automation runs. An action usually controls one of your areas, devices, or entities, for example: "Turn on the lights". You can use building blocks to create more complex sequences of actions.' Below the text are two buttons: '+ ADD ACTION' and '+ ADD BUILDING BLOCK'.

Then select 'Light'

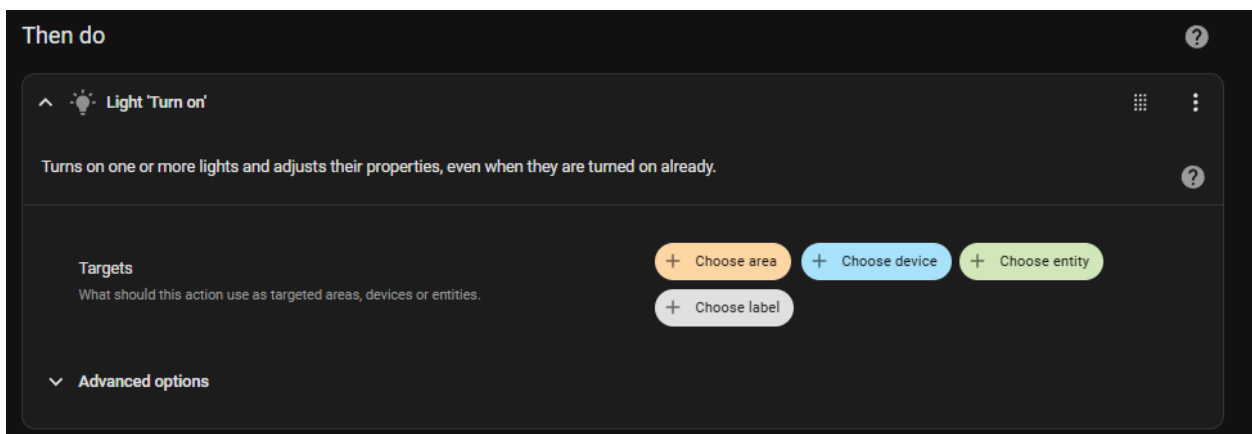


The 'Add action' dialog is titled 'Add action' with a close icon. It features a search bar labeled 'Search action'. Below the search bar is a list of action categories: 'Device' (with a description 'Do something on a device. Great way to start.' and a plus icon), 'Button' (with a right arrow), 'Light' (with a right arrow), 'Notifications' (with a right arrow), 'Number' (with a right arrow), and 'Select' (with a right arrow).

Then select 'Turn on'.



The 'Then do' section will now look like the following:



Select 'Choose entity' and then 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 knows this is a light bulb.

Smart Target System

Model: SmartTargetV3.2

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.

☐ **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

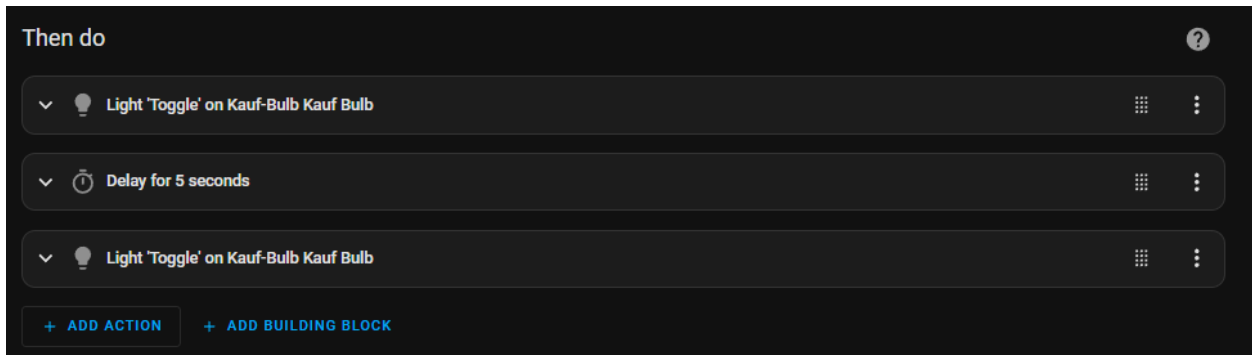
Next, search and add a new action for a delay.

✕ 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:

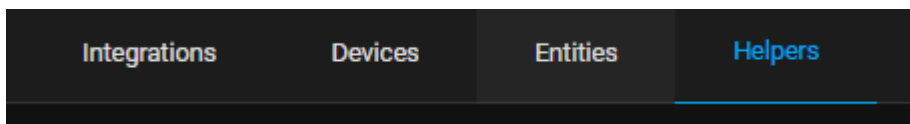


When the target is hit, the light bulb will\should turn on for 5 seconds and then back off.

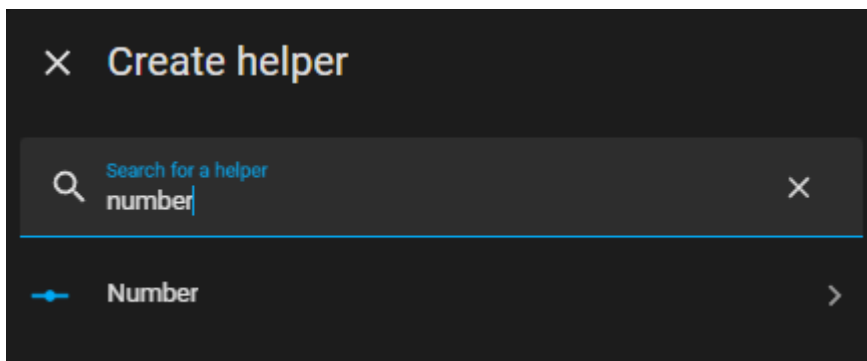
Score Keeping

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

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



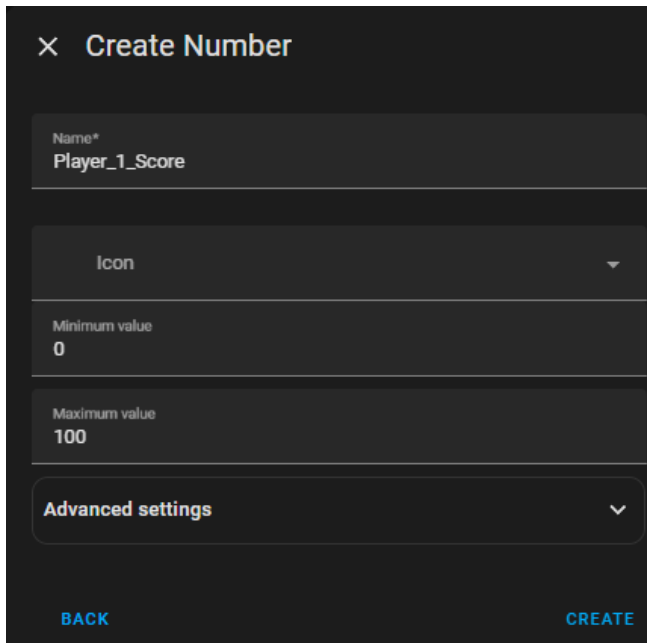
On the bottom right side of the screen click 'Create Helper'.



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

Smart Target System

Model: SmartTargetV3.2



×

Create Number

Name*
Player_1_Score

Icon

Minimum value
0

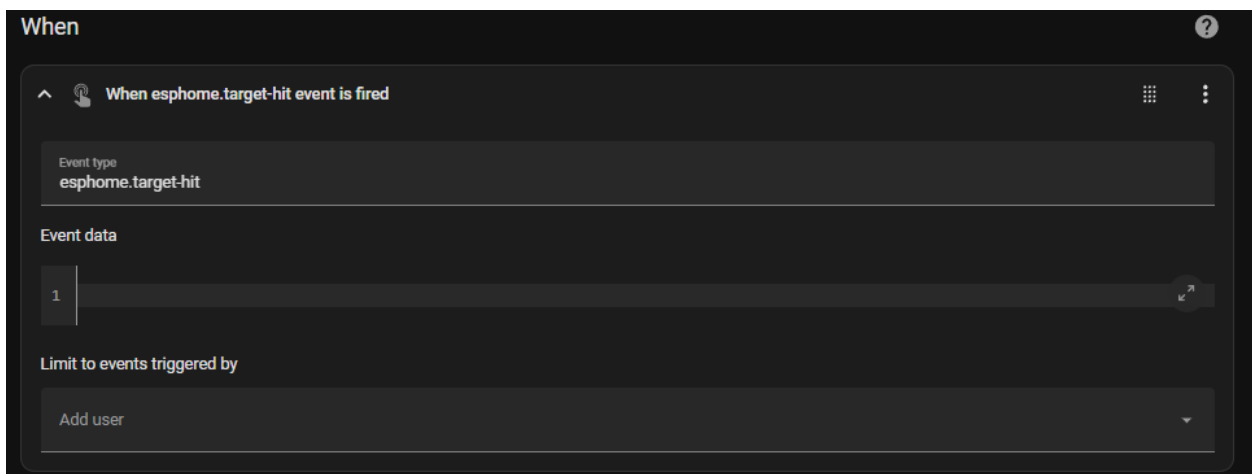
Maximum value
100

Advanced settings

BACK CREATE

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

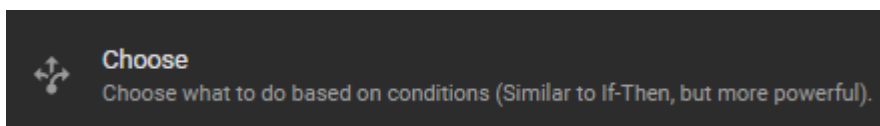
1

Limit to events triggered by

Add user

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

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



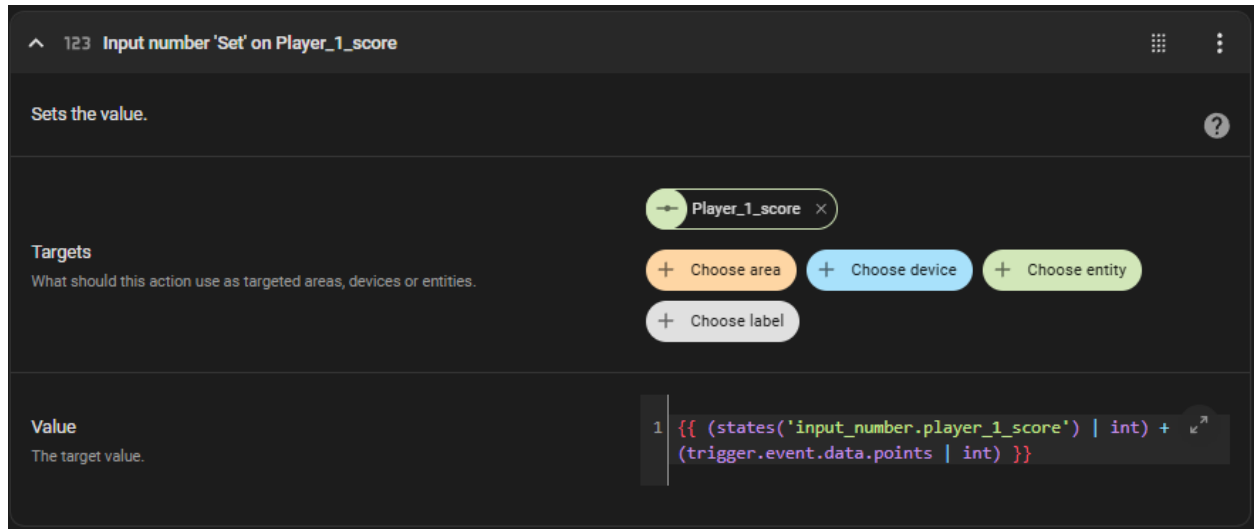
Choose

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

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

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

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



The target is our player_1_score helper we just created.

The value is:

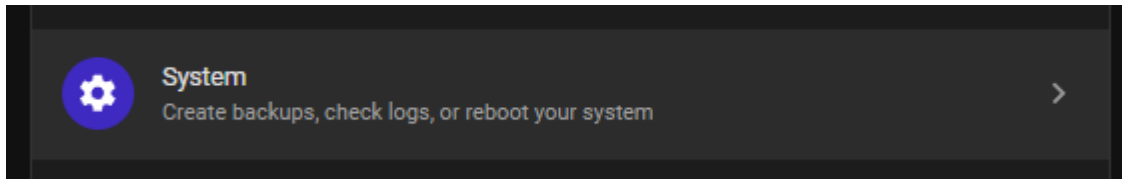
```
{{ (states('input_number.player_1_score') | int) + (trigger.event.data.points | int) }}
```

This weird looking code adds the target points to the player that hit the target.

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.

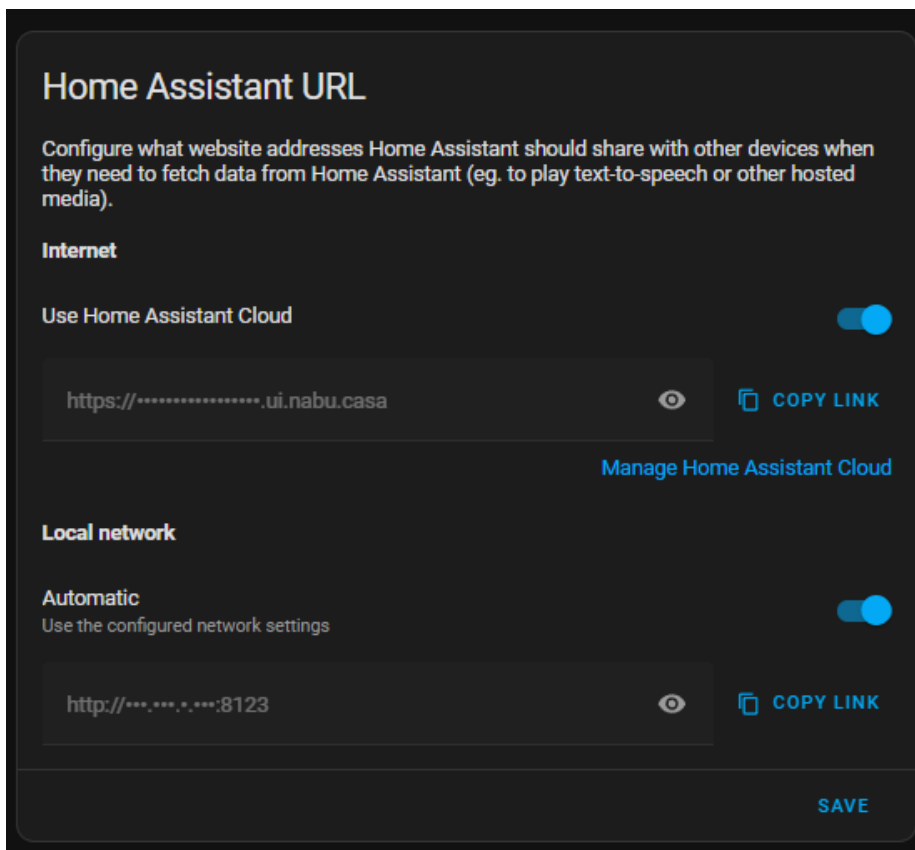
Got to settings then 'System'.



Then go to 'Network'



Then find the 'Home Assistant URL' section.



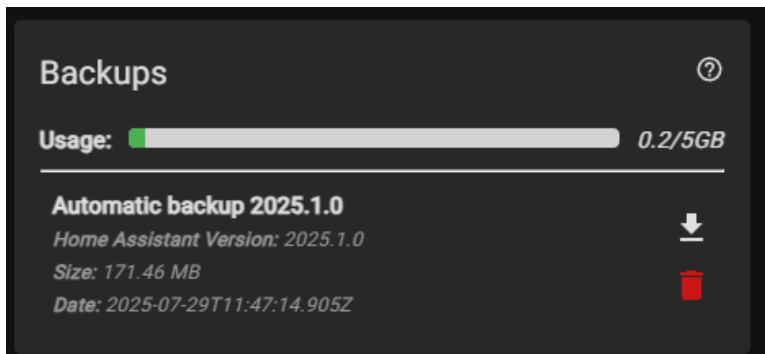
Smart Target System
Model: SmartTargetV3.2

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.



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>