Basic entities that are auto created in home assistant.

'window' relates to the actual AM43 blind controller

'am4d' – is the ESP32 – which can control 1 to 3 x AM43's

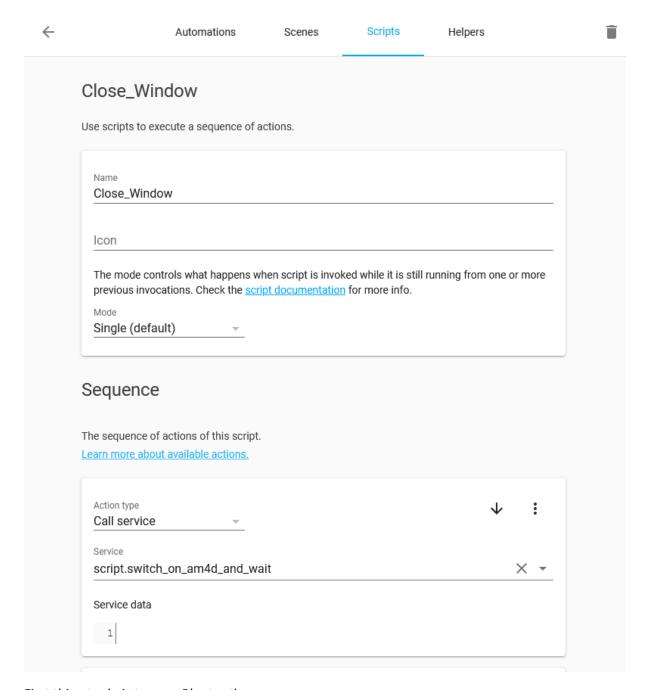
| $\leftarrow$ |          | Integrations                  | Devices        | Entities            | Areas             |                  |
|--------------|----------|-------------------------------|----------------|---------------------|-------------------|------------------|
| Q v          | window   |                               |                |                     |                   | ×Ψ               |
|              |          | ↑ Name                        | Entity         | ' ID                | Integration       | Status           |
|              | *        | am4d switch window            | switch.        | am4d_switch         | MQTT              | ĺ                |
|              |          | window                        | cover          | :window             | MQTT              | •                |
|              | ?        | window Battery                | senso          | or.window_battery   | MQTT              | 0                |
|              | ¢        | window Light                  | senso          | or.window_light     | MQTT              | 0                |
|              | *        | am4d switch window            | switch         | a.am4d_switch       | MQTT              |                  |
|              |          | I created to remember the o   | desired state  | of the switch –     | so that it can be | turned off again |
|              | am4d_on  |                               |                | input_boolean.an    | n4d_on In         | out Boolean      |
| Script       | s creat  | ed – the first two of which I | kick off the o | pening and closi    | ng                |                  |
|              | 亘        | Close_Window                  | script.        | close_window        | Scripts           | ×                |
|              | Ē        | Open_window                   | script.        | open_window         | Scripts           | ×                |
| With         | this ext | tra one being a sub script ca | illed by both  | of the two abov     | e                 |                  |
|              | 亘        | Switch on AM4d and wait       | script         | .switch_on_am4d_and | Scripts           | ×                |

|        |    |   |        |   |    | •  |      |
|--------|----|---|--------|---|----|----|------|
| Λ      | 11 | t | $\sim$ | m | 21 | ın | ns   |
| $^{-}$ | u  | L | v      |   | aι | ıv | כווי |

| Auton<br>again | nation t | hat fires whenever ESP32 switches on – | to check that it doesn't need | to be turned off |
|----------------|----------|--|-------------------------------|------------------|
|                | ф        | Am4d switched on                       | automation.am4a_switched      | Automation       |

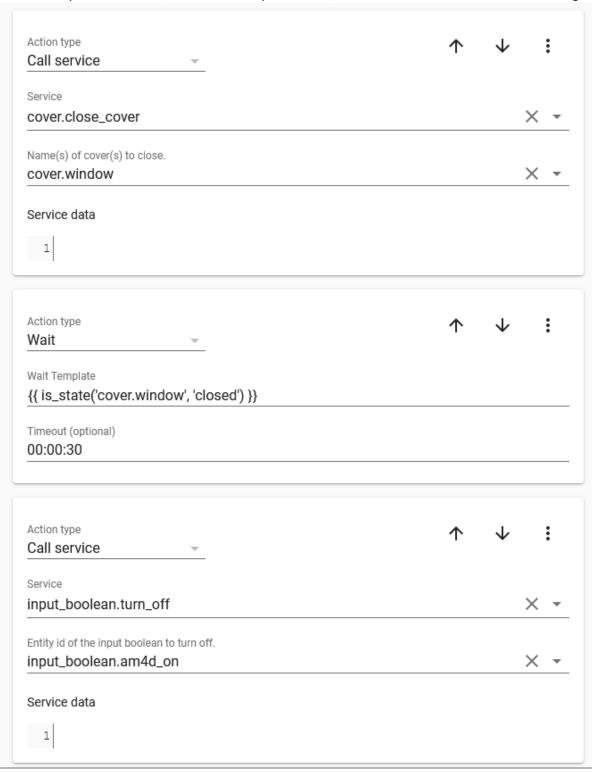
Scripts -

## Close Blind script

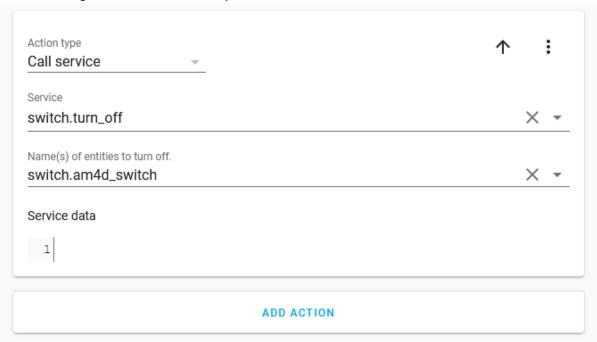


First thing to do is turn on Bluetooth...

Then actually close the blind, wait for it to report closed, then once closed first turn off switch flag

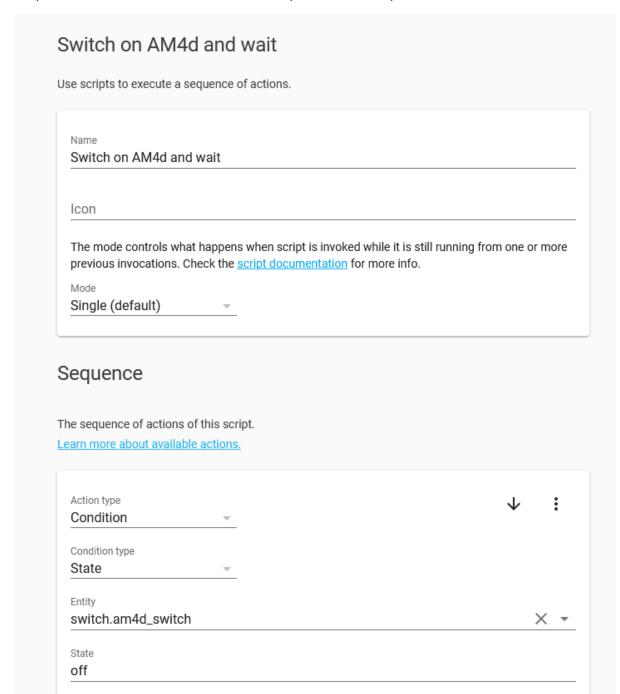


Then once flag turned off then actually turn off Bluetooth switch.

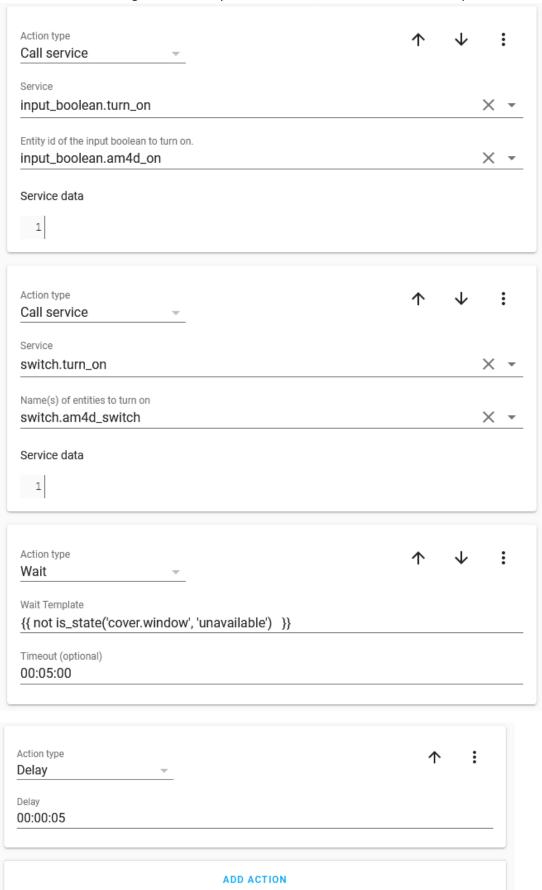


The switch on and wait sub script

Only execute if switch off – so exit immediately if switch already on



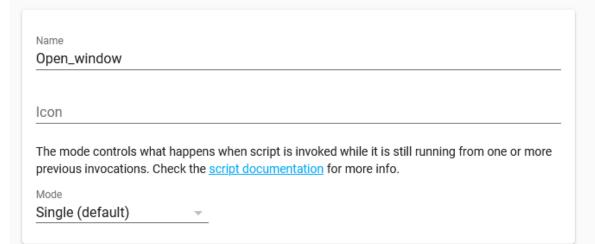
First turn on switch flag, then actually turn on switch, then wait until blind reports its status.



Open script similar to close script

## Open\_window

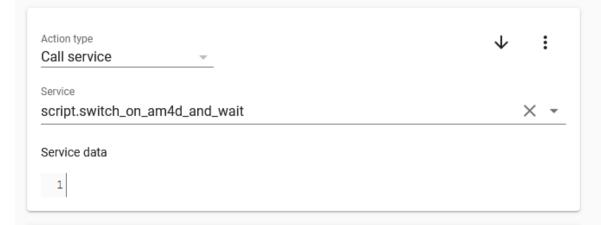
Use scripts to execute a sequence of actions.



## Sequence

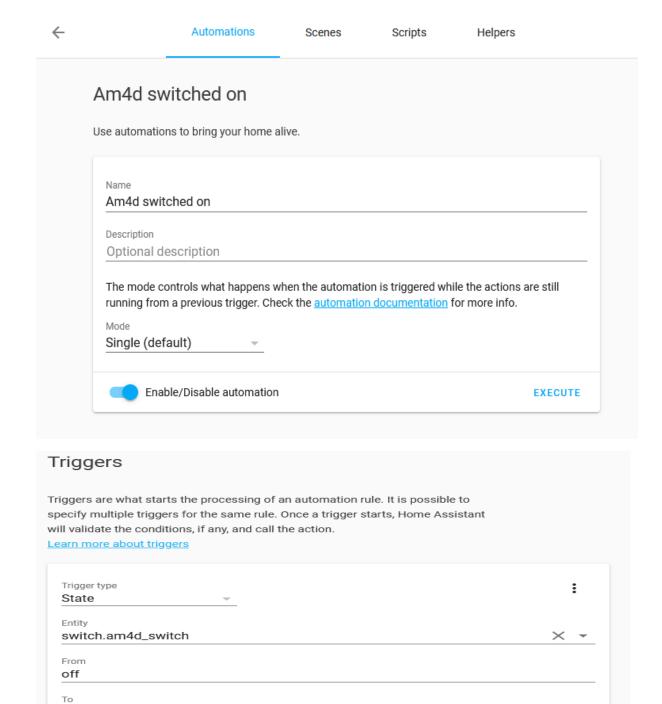
The sequence of actions of this script.

Learn more about available actions.





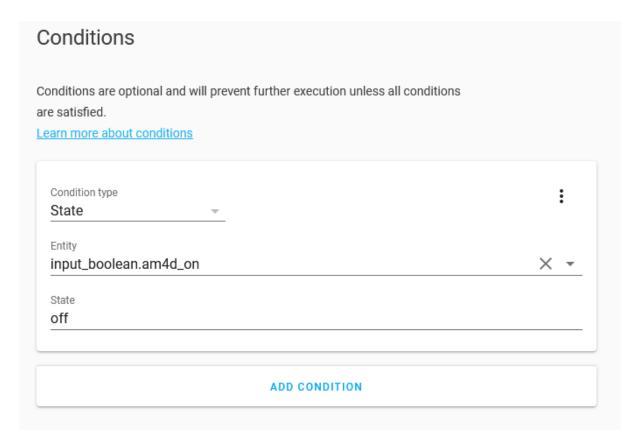
Automation that gets fired every time the ESP32 reports that it has been switched on



Automation only continues if flag indicates that switch shouldn't be on

on

For



Action then fires to turn switch back off again

