Shelly i4 → 2PM Cover Controller (Gen3)

⚠ Hardware Requirement: Integrated End Switches

This script must only be used with **roller shutters equipped with built-in mechanical end stops** (hardware limit switches).
The Shelly 2PM relies on these end positions to perform calibration and position tracking.
Using motors without integrated limit switches (or with external relay control only) will lead to incorrect position reporting and unreliable operation.

This project implements a **smart two-part control system** for Shelly devices to manage blinds or shutters with precise position and slat (lamella) adjustment.
It is designed for **Shelly Gen3 devices running firmware 1.7.x or newer**.

Overview

Device	Script	Function
Shelly i4 Gen3	`ScriptI4.js`	Button interface — translates button presses (`single`, `long`, `double`) into high-level commands and sends them to the 2PM.
Shelly 2PM Gen3	`Script2PM.js`	Core controller — handles calibrated blind movement, nudging, preset heights, and slat positioning.

Communication between the two devices uses the **KVS RPC interface** (`/rpc/KVS.Set`).

Hardware Setup

Function	Device	Input / Output
Up Button	i4	`Input0`
Down Button	i4	`Input1`
Cover Motor	2PM	`Relay0 / Relay1`
Communication	LAN or WLAN	Direct HTTP RPC between i4 and 2PM

Button Mapping (on i4)

Action	Input	Result
Short press	Up / Down	Move up / down if idle, stop if moving
Long press	Up	Nudge up (`nudge_up_ms`)
Long press	Down	Nudge down (`nudge_down_ms`)
Double press	Up	Move to preset height `preset_1`, then slat adjust to `slat_pos_1`

Action	Input	Result
Double press	Down	Move to preset height `preset_2`, then slat adjust to `slat_pos_2`

🧠 Features

- Fully event-driven via i4 inputs
- No internal polling delay on button recognition
- Calibrated mode only (script inactive on uncalibrated covers)
- Fine-grained nudge times for slat angle control
- Asymmetric up/down full-travel times
- Persistent configuration via KVS on the 2PM
- Zero flash wear during runtime — KVS written only on startup

🔗 Communication Setup (Target IP)

The **i4 script** sends all button actions to the **2PM** using an HTTP RPC call:

```
http://<2PM-IP>/rpc/KVS.Set?key=coverex_cmd&value=<command>
```

To make this work, the i4 needs to know the IP address of the target 2PM.

Automatic setup

- On first run, the i4 checks for the KVS key `target_2pm_ip`.
- If it's missing, the script creates it with the default IP (e.g. `"192.168.1.64"`).
- You can read or change it via:

  ```
  http://<I4-IP>/rpc/KVS.Get?key=target_2pm_ip
  http://<I4-IP>/rpc/KVS.Set?key=target_2pm_ip&value="<NEW_IP>"
  ```

- The i4 transmits every button action (e.g. `single_up`, `double_down`, `long_up`, …) to the 2PM's KVS.

Example

If your 2PM has the IP `192.168.1.63`, update the i4 like this:

```
http://192.168.1.65/rpc/KVS.Set?key=target_2pm_ip&value="192.168.1.63"
```

Console log output:

```
i4 target_2pm_ip = 192.168.1.63
```

Default Parameters (on 2PM)

```
nudge_up_ms: 800
nudge_down_ms: 600
preset_1: 40
preset_2: 60
slat_pos_1: 50
slat_pos_2: 25
slat_full_up_ms: 1000
slat_full_down_ms: 1000
poll_interval_ms: 50
run_timeout_ms: 90000
```

All values can be modified via:

```
http://<2PM-IP>/rpc/KVS.Set?key=<param>&value=<new_value>
```

License