

RaspBerryPi-Homekit-Switch

I was looking for a solution to control non HomeKit devices (lights, coffee machines, etc.) via HomeKit.

It turned out that a Raspberry Pi is a perfect platform for doing such things.

Homebridge is a very extendable base with a variety of already developed plugins.

Unfortunately all the available plugins did not work for some reason. My solution was to control the GPIO pins via shell script and use [homebridge-script2](#) to control the script.

[Smartapfel](#) has a very useful guide to setup homebridge. However I wrote all the steps down to have it in a single guide.

The repo includes this guide, my shell script to control the GPIO pins and an example config.json for homebridge.

Step by Step Guide

Install Avahi

```
sudo apt-get install libavahi-compat-libdnssd-dev
```

Install Node

1. First you need to determine your platform: `uname -m`
2. Then go to the NodeJS download page: <https://nodejs.org/dist/latest/>
3. And copy the link for your platform. (make sure you copy the one ending with *.tar.gz)
4. Download the file: `wget https://nodejs.org/dist/latest/node-v11.3.0-linux-armv7l.tar.gz`
5. Extract the file: `tar xf node-v11.3.0-linux-armv7l.tar.gz`
6. Now you can copy the files: `sudo cp -R node-v10.4.1-linux-armv7l/* /usr/local/`

Install homebridge

To install homebridge do a: `sudo npm install -g --unsafe-perm homebridge`

Configure a Service

We need to configure a service to start homebridge on boot. To do so, follow the steps below:

1. Create a service account: `sudo useradd -m -c "Homebridge Service" -s /bin/bash homebridge`
2. We need to configure permissions for that user. Therefore we need to create a file: `sudo nano /etc/sudoers.d/homebridge`
And add the following into it: `homebridge ALL=(root) SETENV:NOPASSWD:
/usr/local/bin/npm, /bin/systemctl restart homebridge, /bin/journalctl,
/usr/local/bin/node`
3. Now we need to set permissions for that file: `sudo chmod 640 /etc/sudoers.d/homebridge`
4. Now we create the service file: `sudo nano /etc/systemd/system/homebridge.service` and add the following content:

```
[Unit]
Description=Node.js HomeKit Server
After=syslog.target network-online.target

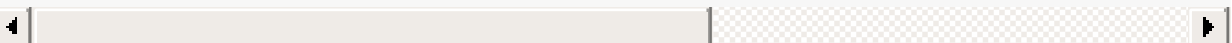
[Service]
Type=simple
User=homebridge
EnvironmentFile=/etc/default/homebridge
ExecStart=/usr/local/bin/homebridge $HOMEBRIDGE_OPTS
Restart=on-failure
RestartSec=10
KillMode=process

[Install]
WantedBy=multi-user.target
```

5. And a 2nd file to configure the environment: `sudo nano /etc/default/homebridge` with content:

```
# Defaults / Configuration options for homebridge
# The following settings tells homebridge where to find the config.json file and wh
HOMEBRIDGE_OPTS=-I -U /var/homebridge

# If you uncomment the following line, homebridge will log more
# You can display this via systemd's journalctl: journalctl -f -u homebridge
# DEBUG=*
```



6. Next we need to reload systemd: `sudo systemctl daemon-reload`
And enable our service: `sudo systemctl enable homebridge`
7. To manage the service we can use the following commands:

- Start: `sudo systemctl enable homebridge`
- Stop: `sudo systemctl stop homebridge`
- Restart: `sudo systemctl restart homebridge`
- Display Log: `sudo journalctl -fau homebridge`

Configure Homebridge

To Configure homebridge, we create a config directory and place our config.json into it.

- Create directory: `sudo mkdir -p /var/homebridge`
- Create our config file: `sudo nano /var/homebridge/config.json`
with the following content:

```
{
  "bridge": {
    "name": "SWITCHBOX-4P-001",
    "username": "02:68:B3:29:DA:98",
    "port": 51826,
    "pin": "094-31-749"
  },
  "description": "This is my configuration",
  "accessories": [
    {
      "accessory": "Script2",
      "name": "Relay 01",
      "on": "/var/homebridge/relaycontrol/relaycontroller.sh on 17",
      "off": "/var/homebridge/relaycontrol/relaycontroller.sh off 17",
      "state": "/var/homebridge/relaycontrol/relaycontroller.sh status 17",
      "on_value" : "ON"
    },
    {
      "accessory": "Script2",
      "name": "Relay 02",
      "on": "/var/homebridge/relaycontrol/relaycontroller.sh on 18",
      "off": "/var/homebridge/relaycontrol/relaycontroller.sh off 18",
      "state": "/var/homebridge/relaycontrol/relaycontroller.sh status 18",
      "on_value" : "ON"
    },
    {
      "accessory": "Script2",
      "name": "Relay 03",
      "on": "/var/homebridge/relaycontrol/relaycontroller.sh on 23",
      "off": "/var/homebridge/relaycontrol/relaycontroller.sh off 23",
      "state": "/var/homebridge/relaycontrol/relaycontroller.sh status 23",
      "on_value" : "ON"
    },
    {
      "accessory": "Script2",
      "name": "Relay 04",
      "on": "/var/homebridge/relaycontrol/relaycontroller.sh on 24",
      "off": "/var/homebridge/relaycontrol/relaycontroller.sh off 24",
```

```
        "state": "/var/homebridge/relaycontrol/relaycontroller.sh status 24",  
        "on_value" : "ON"  
    },  
    ],  
    "platforms": [  
    ]  
}
```

3. Change permissions for our config directory: `sudo chown -R homebridge:homebridge /var/homebridge`