* Step 1

Set up your Raspberry Pi. Open a command line, or open an SSH terminal.

* 2

Step 2

Move into a directory you want to install the *milights-bridge* inside.

cd ./scripts/

* 3

Step 3

**Install *milights-bridge***

Clone the *milights-bridge* project from GitHub.

sudo git clone https://github.com/KevinVR/milights-bridge.git

Get into the *milights-bridge*folder

cd ./milights-bridge/

Ensure NodeJS and NPM are installed. If unsure, run the following commands.

sudo apt-get update

sudo apt-get install nodejs

sudo apt-get install npm

Install the required dependencies (make sure you have NPM installed)

sudo npm install

Copy the example configuration file.

sudo cp config.example.js config.js

Modify the configuration to suit your needs, follow the instructions within the file.

sudo vi config.js

or

sudo nano config.js

Turn on the Milights-Bridge software

sudo node server.js

You should see some output from the command line, such as:

Setting up EJS...

Initialising Milight bridge connection (version v6)

Checking for updates...

Setting up port 3000

milights-bridge is ready, please open your browser at http://192.168.1.60:3000

Open a browser page to the URL shown in the output.

http://192.168.1.60:3000

Test out the *milights-bridge* features, ensure it works

* 4

Step 4

**Install the *ha-bridge* software**

Follow ***[this guide](https://community.home-assistant.io/t/general-ha-bridge-guide-hue-bridge-for-alexa-or-google-home/6758" \t "_blank)***for setting up *ha-bridge*. Make sure that you get the latest version of ha-bridge, however (since there could be more features).

Once it is running, come back to this guide.

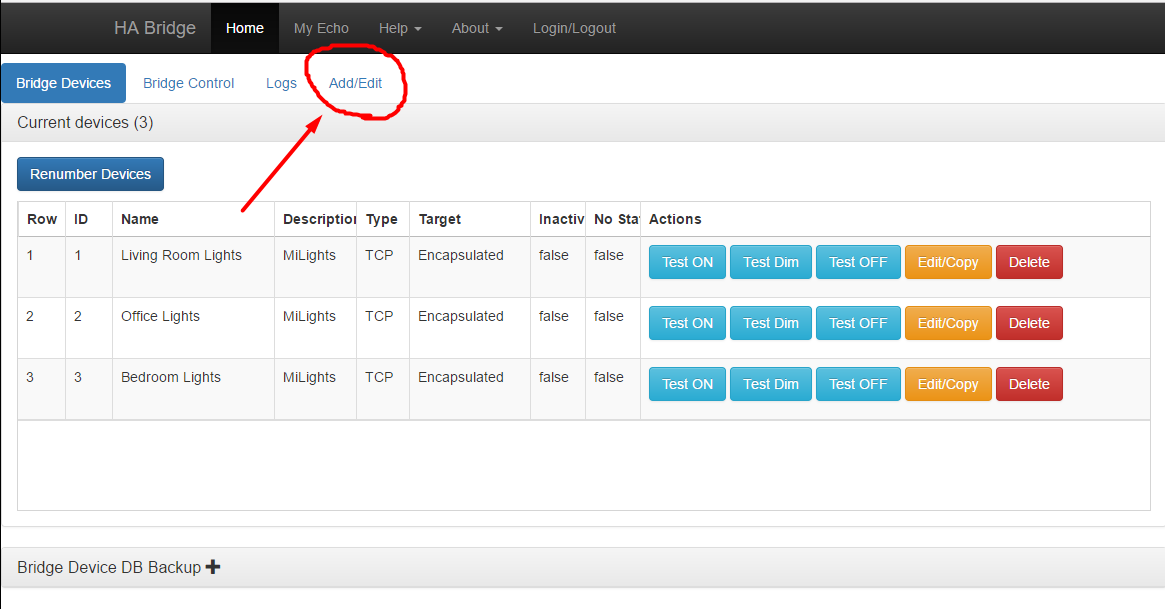
* 5

Step 5

**Add your *Milight*lights to the *ha-bridge* software**

Open up the *ha-bridge* software in your browser ([http://raspberry-pi-IP-address:80/](http://raspberry-pi-ip-address/)), it runs on port 80 by default.

You'll see a list of bridge devices, click add.

[](https://cdn.hackaday.io/images/7377841497098713352.png)

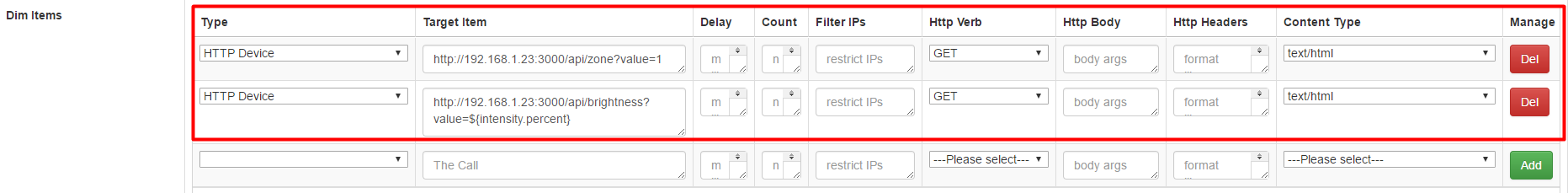
Set a name and description to your likings.

* + Device type: TCP
  + Map Type: Hue device

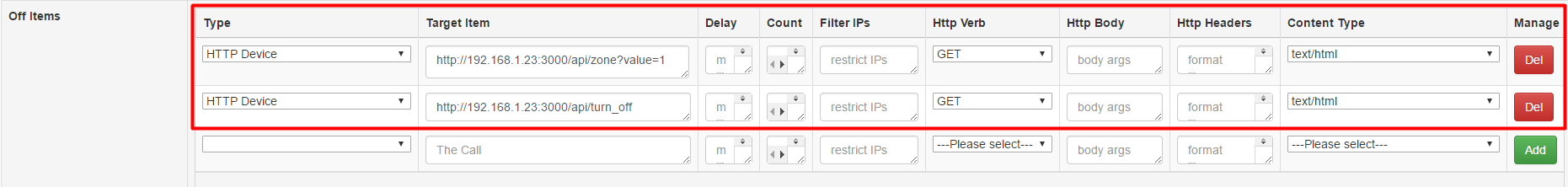
Then there is a place to set "On items". Copy the settings from the below image. Change the IP address to your Raspberry Pi address. The port should be the port you have installed *milights-bridge* on. Change X in the *zone?value={X}* to the zone ID of your choice for this device.

[](https://cdn.hackaday.io/images/2098631497098956084.png)

Repeat the same process for dim items, however, the settings are slightly different.

[](https://cdn.hackaday.io/images/7732811497098983597.png)

Repeat the same process for off items.

[](https://cdn.hackaday.io/images/9113451497099019972.png)

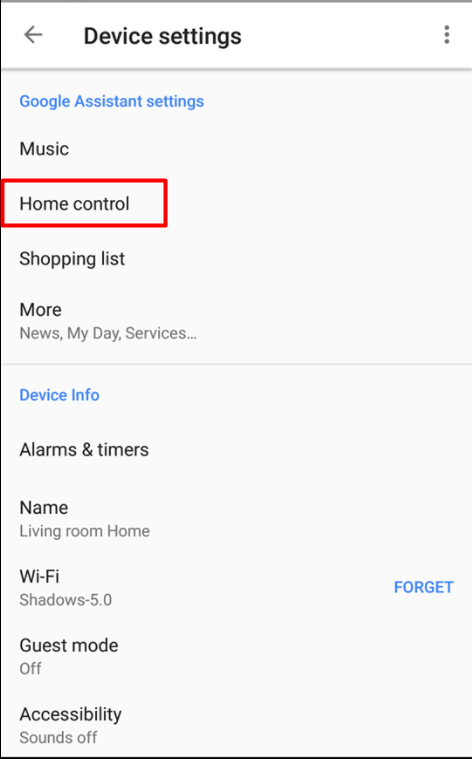
Save and repeat for all your Milight zones.

* 6

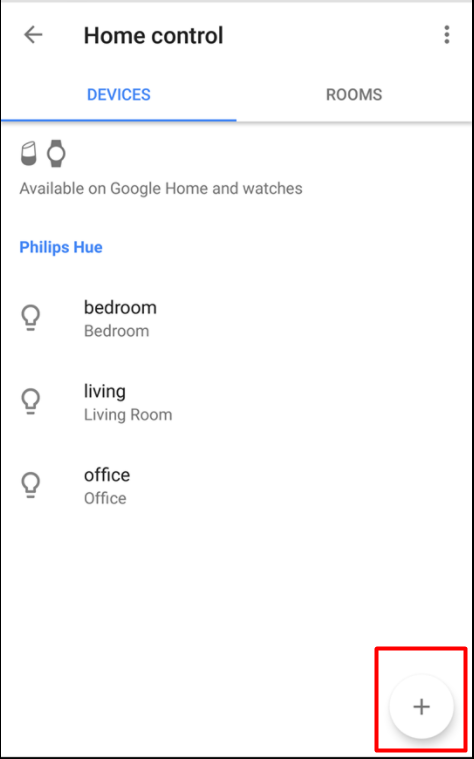
Step 6

**Setup the lights in Google Home**

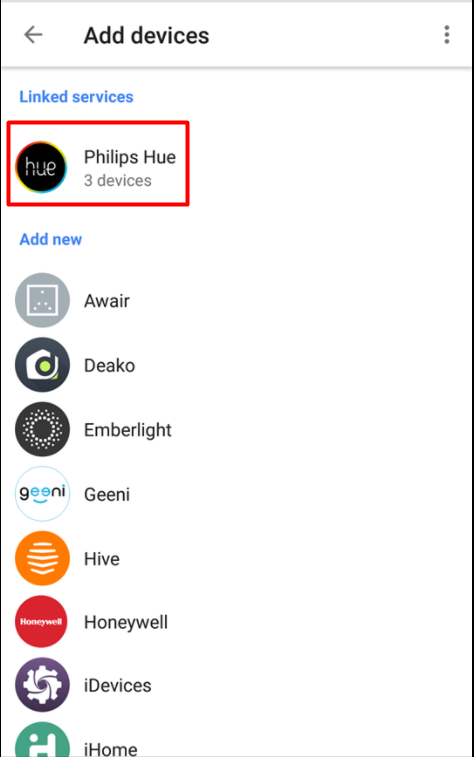
Open the Google Home app on your mobile phone. Click the devices button on the right upper corner of the screen. Click the 3 dots on the upper right corner of your Home device, click Settings. You will arrive in the following screen:

[](https://cdn.hackaday.io/images/262741497099621001.png)

Then click Home control. Click the "+" button on the bottom.

[](https://cdn.hackaday.io/images/9925671497099634161.png)

Choose Philips Hue and continue the Google Home instructions.

[](https://cdn.hackaday.io/images/1595671497099645570.png)

* 7

Step 7

**Talk to your Google Home!**

Hey Google! Turn on living.

Hey Google! Turn off living.

Hey Google! Turn on lights.

Hey Google! Set living brightness to 50%.

Hey Google! Turn off lights.

**Congratulations, all is set up! And now you can also enjoy the *milights-bridge* software as an extra remote control for the milight lights.**

An advantage of the *milights-bridge* app is that, it is fast to load and click. However, the official Milight app is slow (start the app, wait for it to find the bridge, then initialise the bridge). The reason the *milights-bridge* app is faster, is because it keeps the bridge loaded at all times. So when opening the page, it does not need to load the connection, since it's already loaded.

* 8

Step 8

**Ensure that *milights-bridge* is always running.**

Install forever.

sudo npm install -g forever

Forever can now be used to run *milights-bridge.*

sudo forever start server.js

sudo forever stop server.js

Add this command to your crontab, to make it run on startup.

sudo crontab -e

Add this line to the end of your crontab.

@reboot forever /path/to/milights-bridge/server.js