

# How to use HABridge with Z-way Server (Razberry) - Google Home Assistant or Amazon Alexa



## Google Home

Please note that in the beginning of 2018 Google change the way of communication. That caused issues - HABridge is now not able to communicate with Google Assistant directly. You need to link account - which you can't - as this is not a real bridge. 😞 Why? No clue.

You can look eventually into other solutions - like OpenHAB - which is not so easy to set up- but could be used with Google. Or wait a month for new Z-way server which probably would have finally direct communication. It is already supporting Alexa.

Normally, if you are doing home automation - you would expect that it would be possible to steer your home using voice systems. Of course, we can spend some time on OpenHAB to make this voice activated... but it would cost a lot of time.

The best solution would be to use existent on market - proven solutions. We have actually two of those... Amazon Alexa voice assistance system and Google Home Assistant. Amazon is actually indicating that there is a skill for Z-way server. There is only one small issue... it is not working - in most cases. And for Google assistant - there is nothing as well. But... both systems work with Philips Hue solution. Of course, we do not have Hue - but a z-wave system. What we need to do we have to create a bridge. To create a bridge we would use a great tool - HaBridge - which you can download from [GIT - JUST HERE](#).

On Git website you would have as well short description how it works... Just install this on the same system where your Z-way server is located.

For Z-way server - we can use two or even three methods - sending HTTP requests directly through API, using MQTT and what I would present here - using scripts.

Sending HTTP requests to API would work if you switch off authentication. This is not something that we want to do. MQTT is nice standard - but would cost you some more complicated configuration. The third method is the one which we would use. We would do through script authentication., then we would send a direct command to Z-way server. This is quite similar to what we did in this document - [Z-way server API and PYTHON - using authentication with API](#) - of course, in that case, we would add an extra value which would refer to command - like 'on' or 'off' or for dimmers - '86' as example percentage value.

## Step-by-step guide

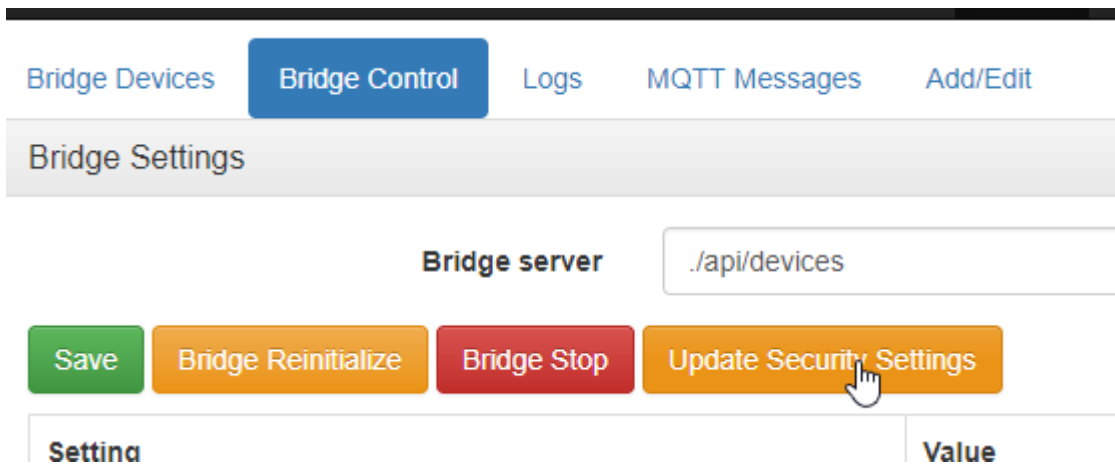
Let's start. First of all, install HA-Bridge on your server. Just follow standard instruction for Raspberry Pi - [look here](#)

Be sure you use port 80. Only this port would be visible for voice systems.

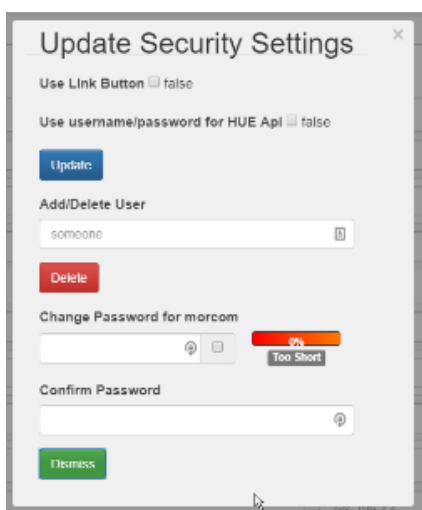
The screenshot shows the HA Bridge web interface. At the top, there is a navigation bar with 'Home' selected. Below it, there are tabs for 'Bridge Devices', 'Bridge Control', 'Logs', 'MQTT Messages', and 'Add/Edit'. The main content area shows 'Current devices (1)' and a 'Renumbr Devices' button. Below this is a table with the following data:

Row	ID	Name	Descriptio	Device State	Type	Target	Inactive	No Stati	Actions
1	4	My room	Strych	on=false,br=false,hue=0,sat=0,effect=none,ct=0,alert=none,colormode=,reachable=true,XYList=	mqtt	zway	false	false	Test ON Test Dim Test OFF Edit/Copy Delete

After login, you should see something like this. HA Bridge is working. At the next step, I would suggest protecting HA Bridge with some password. Go to Bridge Control and update security.



Create a new user with password. Do not use security for API - unless you would like to have issues.



After change - save the configuration. It would re-load HA-Bridge and would ask you to log in again. We are now ready to go.

## Creating Python scripts

As mentioned before we would use Python scripts. Let's look again at the basic script - which would include only two commands - 'on' and 'off'. We would build that one based on mentioned at the beginning solution. We would add just some modifications. We want to make this script universal. So - as an entry, we would use device-id, a value which we want to set - actually we would be very strict - so 'on' to switch on, 'off' to switch off and number to set dimmer. That should solve most of the cases. Username and password we would keep secret.

So our universal script would look like the one below:

## Universal on-off script for z-way

```
import json
import requests
import optparse

#define proper values below
#---Start configuration---

topLevelUrl = 'http://localhost:8083' #define proper address of the Z-way server - it could be actually
correct address in that case - so you can create local user
username = 'admin' #modify that - I would suggest to create a dedicated user
password = 'password' #Try not to use special chars in password

#-----END-----

#Here we have some magic

parser = optparse.OptionParser()

parser.add_option('-d', '--device', action="store", dest="device", help="Please define your device ID",
default="")
parser.add_option('-c', '--command', action="store", dest="command", help="Please define your command",
default="")
parser.add_option('-v', '--value', action="store", dest="value", help="Please define value to set -
could be on or off or numerical value", default="-1")

options, args = parser.parse_args()

if options.value is '-1':
    RequestUrl= topLevelUrl + '/ZAutomation/api/v1/devices/'+options.device+'/command/'+options.command
else:
    RequestUrl= topLevelUrl + '/ZAutomation/api/v1/devices/'+options.device+'/command/'+options.command+'?
level='+options.value

LoginUrl = topLevelUrl + '/ZAutomation/api/v1/login'

LoginHeader = {'User-Agent': 'Mozilla/5.0', 'Content-Type': 'application/json'}
Formlogin = '{"form": true, "login": "'+username+'", "password": "'+password+'", "keepme": false,
"default_ui": 1}'

session = requests.Session()
session.post(LoginUrl,headers=LoginHeader, data=Formlogin)

response = session.get(RequestUrl)
#html = response.text
#print html
parsed_json = response.json()
print parsed_json
```

Of course, we can polish this script more - to react differently on numerical values and in a different way on 'on' or 'off'. Now if you run the script in a standard way - like:

```
# python script.py --help
Usage: zway.py [options]

Options:
  -h, --help            show this help message and exit
  -d DEVICE, --device=DEVICE
                        Please define your device ID
  -c COMMAND, --command=COMMAND
                        Please define your command - on or off or exact
  -v VALUE, --value=VALUE
                        Please define value to set for dimmers

# python script.py -d DEVICE_ID_FROM_ZWAY -c COMMAND -v VALUE
```

it should interact with your system. Please make a test to confirm, that it is working fine.

## Compile Python script

If all works fine - I would suggest compiling script - so it could be executed directly.

I would suggest using NUITKA compiler.

Please visit [their site for detailed installation](#). On my system I was able to install *nuitka* directly from repository.

```
# apt install nuitka
```

After installation just run command

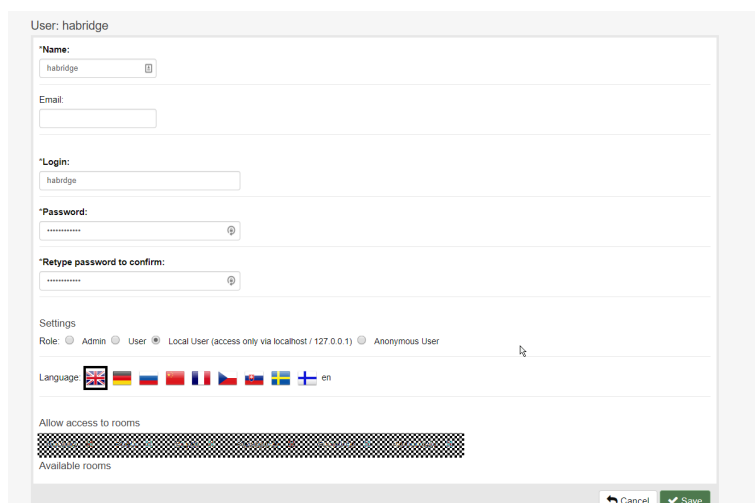
```
# nuitka script.py
```

After compilation process you should have ready to use ELF program - named script.exe... we love Microsoft.... Now you can set up HA-Bridge to work with this script. Be sure it is stored outside of the root folder and has 766 user rights - so anyone could run the script.

## Set up HA Bridge to work with script

Login to your Z-way server and first define new user - allowing access only from localhost.

Assign to the user all available rooms - so it would have access to your devices. I have created *habridge* user.



User: habridge

\*Name: habridge

Email:

\*Login: habridge

\*Password: .....

\*Retype password to confirm: .....

Settings

Role:  Admin  User  Local User (access only via localhost / 127.0.0.1)  Anonymous User

Language:  en

Allow access to rooms

Available rooms

Cancel Save

Now go and try your compiled command. Be sure to use a proper username and password defined above.

```
/opt/scripts# ./zway.exe -d ZWayVDev_zway_6-0-37 -c on
{'u'error': None, u'data': None, u'message': u'200 OK', u'code': 200}
```

As you can see after execution my command was delivering code 200 and light was on.

Now time to set up HA Bridge.

First of all, we need to modify standard config file - as we want to indicate the place where our scripts are located.

Go to

/etc/systemd/system and here we have service file

habridge.service

Please edit and add folder location for option -Dexec.garden

```
[Unit]
Description=HA Bridge
Wants=network.target
After=network.target

[Service]
Type=simple
WorkingDirectory=/opt/habridge
ExecStart=/usr/bin/java -jar -Dexec.garden=/opt/scripts/ -Dconfig.file=/opt/habridge/data/habridge.
config /opt/habridge/ha-bridge-4.5.6.jar

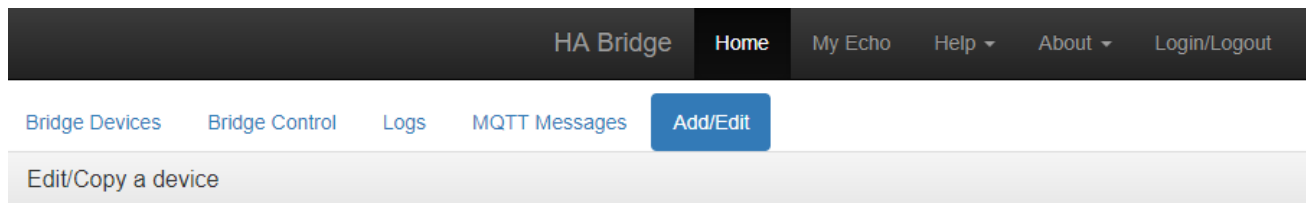
[Install]
WantedBy=multi-user.target
```

Then reload daemon and service.

```
systemctl daemon-reload
```

```
systemctl restart habridge.service
```

Open HA Bridge page on the Z-way server and go to Add/Edit tab.



This screen allows the modification of many fields the bridge uses. Please use care when updating these fields as you may break the settings u

This area allows you to create any http, tcp or udp call to an endpoint. You can use the default GET or select the http verb type below and config

Define parameters - Keep in mind that NAME is used by your voice system. So if I would put *My room* as the name - then in Voice activated system I would need to use the command like *Switch off my room*.

Be sure to choose that you do not want updates... Actually, it doesn't matter. But just in case. And choose that we are going to execute script /command.

<b>No State (Do not update state for device)</b>	<input checked="" type="checkbox"/> true
<b>Off State Resets Bri</b>	<input type="checkbox"/>
<b>Filter Address (comma separated list)</b>	Only use if you want to restrict this device to a specific c
<b>Target</b>	default
<b>Device Type (Informational)</b>	Execute Script/Program ▾

Now the most important - we define two actions - for ON and OFF

This would include a place where the script is stored. Just add command - as path is define as exec.garden value. After adding command click ADD button... then save all.

Device Type (informational)	Execute Script/Program ▾								
Map Type (Legacy)	Execute Command/Script/Program ▾								
Unique Id (used for Hue responses)	00:17:88:5E:D3:05-05								
Map ID	1111								
On Items	Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Heade	Content Type
	Execute Command/Script/Program ▾	zway.exe -d ZWayVDev_zway_6-0-37 -c on	millis	number	restric	---Please select---	body	for	---Please select---
		The Call	millis	number	restric	---Please select---	body	for	---Please select---

Remember to define off command - with -c off.

If you have dimmer - just set this in a similar way:

```
zway.exe -d DEVICE_ID -c exact -v ${intensity.percent.hex}
```

It would send value from 0 to 100.

You can play with 3 values through HA Bridge - you can use ON, OFF and DIM value. You can as well mix all available commands which are accessible on [API Documentation site for Z-way server](#).

## Activate your new lights in Google Assistant or in Alexa

Activation is quite standard. Just visit your Alexa skill list and add Philips Hue skill. That's it. After a few moments, it should be able to see all your equipment. You can switch on lights just saying *Alexa, switch on my Kitchen light* etc. You can as well activate your security systems by voice or cancel the smoke alert.

In Google Assistant - go to Home automation and add Hue to your list. It should detect your HA Bridge and it should report this as ready. You should be able to see all defined devices. Now you can as well create groups or assign devices to rooms. Keep in mind that it would not report a real state - as we do not have direct method to verify lights state. It would just switch lights on or off. Or Dim lights as requested.

## Related articles

- [How to use HABridge with Z-way Server \(Razberry\) - Google Home Assistant or Amazon Alexa](#)
- [Z-way server API and PYTHON - using authentication with API](#)