



## Technik Brennpunkt



**Smart Home**  
Up to 50% OFF Coupon Deals



*It's time for  
smart living.*

**S**eit einiger Zeit hat Amazons Alexa einen Platz in unserem Wohnzimmer eingenommen. Lichter und Steckdosen steuern ist hier nichts mehr neues und schnell kam die Überlegung das Ambilight, welches über Hyperion auf dem Raspberry Pi läuft, ebenfalls mit Alexa ansteuern zu können.



direkte Steuerung von Hyperion über Alexa. So könnt ihr dann Befehle geben wie “Alexa, schalte Ambilight an”.

Wie ihr das alles einrichtet und was sich sonst noch alles damit machen lässt, erfahrt ihr in diesem Beitrag.

## ANLEITUNG:

### ha-bridge installieren

loggt euch auf eurem Raspberry Pi auf dem Hyperion läuft über SSH ein und prüft, ob ihr Java installiert habt.

```
java -version
```

Wenn keine Versionsnummer angezeigt wird und ein Fehler erscheint, dann müsst ihr erst Java mit diesem Befehl installieren:

```
sudo apt-get update && sudo apt-get install oracle-java8-jdk
```

Im Home Ordner führt ihr jetzt nacheinander diese Befehle aus, um ha-bridge zu installieren:

```
mkdir habridge
```

```
cd habridge
```

```
wget https://github.com/bwssystems/ha-bridge/releases/download/v5.1.0/ha-bridge-5.1.0.jar
```

### habridge Service



Folgenden Text einfügen. Wählt als Port z.B. 81 oder einen über 1024, damit ihr keine Root Rechte benötigt.

```
[Unit]
```

```
Description=HA Bridge
```

```
Wants=network.target
```

```
After=network.target
```

```
[Service]
```

```
Type=simple
```

```
ExecStart=/usr/bin/java -jar -Dserver.port=81-
```

```
Dconfig.file=/home/pi/habridge/data/habridge.config  
/home/pi/habridge/ha-bridge.jar
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Mit Strg+X schließen und “Y” mit anschließendem Enter die Datei speichern.

Jetzt die systemctl neustarten und habridge starten.

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable habridge.service
```

Sollten Fehler erscheinen, könnt ihr diese im Log nachsehen.

```
tail -f /var/log/syslog
```



```
cd ~/habridge
```

```
nano starthabridge.sh
```

..fügt den folgenden Code ein. Anschließend wieder mit Strg+X und Y schließen und speichern:

```
CD /HOME/PI/HABRIDGE
```

```
RM /HOME/PI/HABRIDGE/HABRIDGE-LOG.TXT
```

```
NOHUP JAVA -JAR -
```

```
DCONFIG.FILE=/HOME/PI/HABRIDGE/DATA/HABRIDGE
```

```
.CONFIG /HOME/PI/HABRIDGE/HA-BRIDGE-5.1.0.JAR
```

```
> /HOME/PI/HABRIDGE/HABRIDGE-LOG.TXT 2>&1 &
```

```
CHMOD 777 /HOME/PI/HABRIDGE/HABRIDGE-
```

```
LOG.TXT
```

Dem Skript die benötigten Rechte verteilen:

```
chmod u+x starthabridge.sh
```

## Hyperion Skripte

Ich habe mir nun im Home Ordner den Ordner "hyperion" angelegt, (könnt dieses natürlich auch anders nennen) und habe dort meine ganzen Skripte reingelegt.

In meinem Fall habe ich eine neue Hyperion Config angelegt und hier keinen Grabber ausgewählt, Starteffekt rausgenommen und Startfarbe



Starten des Rasperrys das Ambilight erst mal ausbleibt.

Das “AN” Skript sieht dann folgendermaßen aus und startet die richtige Config.

```
#!/BIN/SH
SUDO KILLALL HYPERIOND
SLEEP 1
SUDO /USR/BIN/HYPERIOND
/ETC/HYPERION/HYPERION.CONFIG.JSON
</DEV/NULL >/DEV/NULL 2>&1 &
```

Speichert es dann zB. unter dem Namen hyperion\_an.sh in eurem eben erstellen Ordner ab.

Für das “AUS” Skript habe ich folgendes gewählt und hyperion\_aus.sh genannt.

```
#!/BIN/BASH
CURL -V -H "ACCEPT: APPLICATION/JSON" -H
"CONTENT-TYPE: APPLICATION/JSON" -X POST -D
'{"ID":1,"JSONRPC":"2.0","METHOD":"GUI.SHOWNOTI
FICATION","PARAMS":
{"TITLE":"HYPERION","MESSAGE":"ENABLED","IMAGE
":"/HOME/PI/HYPERION.PNG}}'
HTTP://127.0.0.1:8080/JSONRPC
HYPERION-REMOTE --COLOR BLACK
```

Dieses setzt dann einfach die Farbe des Ambilight auf Schwarz. Das hat den Vorteil, dass es durch Störungen oder Stromschwankungen



Wichtig an dieser Stelle: Gebt den Skripten 777 Berechtigungen, sodass sie ausführbar sind.

```
chmod 777 ~/hyperion/skript.sh
```

## habridge Webinterface / Konfiguration

Loggt euch nun mit dem Browser auf das ha-bridge Interface mit dem von euch gewählten Port ein:

Bsp. <http://192.168.1.100:81>

Das sollte dann so aussehen:

Row	ID	Name	Descri	Device State	Type	Target	Inactive	No Stat	Actions
1	2	Lounge		on=.bri=.hue=.sat=.effect=.ct=.alert=.colormode=.reachable=.XYList=	exec	Encapsulated	false	false	Test ON Test Dim Test OFF Edit/Copy Delete
2	3	Ambilight		on=.bri=.hue=.sat=.effect=.ct=.alert=.colormode=.reachable=.XYList=	exec		false	false	Test ON Test Dim Test OFF Edit/Copy Delete

Klickt in der Menüleiste auf Add/Edit und fügt ein neues Gerät hinzu. Als Name irgendetwas eindeutiges wählen, was später dann das Signalwort für Alexa ist.

Device Type und Map Type auf “Execute Command/Script...” setzen.

Bei On bzw. Off Items unten wählt ihr ebenfalls “Execute Command/Script..” und gebt bei Target Item den Pfad zu eurem Skript ein.

```
/home/pi/hyperion/skript.sh
```



Inactive	<input type="checkbox"/> false																														
No State (Do not update state for device)	<input type="checkbox"/> false																														
Off State Resets Bri	<input type="checkbox"/> false																														
Filter Address (comma separated list)	Only use if you want to restrict this device to a specific caller(s)																														
Target	default																														
Device Type (Informational)	Execute Script/Program																														
Map Type (Legacy)	Execute Command/Script/Program																														
Unique Id (used for Hue responses)	00:17:86:5E:03:03-03																														
Map ID	1111																														
On Items	<table border="1"> <thead> <tr> <th>Type</th> <th>Target Item</th> <th>Delay</th> <th>Count</th> <th>Filter IPs</th> <th>Http Verb</th> <th>Http Body</th> <th>Http Headers</th> <th>Content Type</th> <th>Manage</th> </tr> </thead> <tbody> <tr> <td>Execute Command</td> <td>/home/hyperion/hyperion_amb</td> <td>ms</td> <td>number</td> <td>restrict IPs</td> <td>---</td> <td>body args</td> <td>format like: [{"name": "A"}]</td> <td>---Please select---</td> <td>Del</td> </tr> <tr> <td></td> <td>The Call</td> <td>ms</td> <td>number</td> <td>restrict IPs</td> <td>---</td> <td>body args</td> <td>format like: [{"name": "A"}]</td> <td>---Please select---</td> <td>Add</td> </tr> </tbody> </table>	Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Headers	Content Type	Manage	Execute Command	/home/hyperion/hyperion_amb	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Del		The Call	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Add
Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Headers	Content Type	Manage																						
Execute Command	/home/hyperion/hyperion_amb	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Del																						
	The Call	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Add																						
Dim Items	<table border="1"> <thead> <tr> <th>Type</th> <th>Target Item</th> <th>Delay</th> <th>Count</th> <th>Filter IPs</th> <th>Http Verb</th> <th>Http Body</th> <th>Http Headers</th> <th>Content Type</th> <th>Manage</th> </tr> </thead> <tbody> <tr> <td></td> <td>The Call</td> <td>ms</td> <td>number</td> <td>restrict IPs</td> <td>---</td> <td>body args</td> <td>format like: [{"name": "A"}]</td> <td>---Please select---</td> <td>Add</td> </tr> </tbody> </table>	Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Headers	Content Type	Manage		The Call	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Add										
Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Headers	Content Type	Manage																						
	The Call	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Add																						
Off Items	<table border="1"> <thead> <tr> <th>Type</th> <th>Target Item</th> <th>Delay</th> <th>Count</th> <th>Filter IPs</th> <th>Http Verb</th> <th>Http Body</th> <th>Http Headers</th> <th>Content Type</th> <th>Manage</th> </tr> </thead> <tbody> <tr> <td>Execute Command</td> <td>/home/hyperion/hyperion_amb</td> <td>ms</td> <td>number</td> <td>restrict IPs</td> <td>---</td> <td>body args</td> <td>format like: [{"name": "A"}]</td> <td>---Please select---</td> <td>Del</td> </tr> </tbody> </table>	Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Headers	Content Type	Manage	Execute Command	/home/hyperion/hyperion_amb	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Del										
Type	Target Item	Delay	Count	Filter IPs	Http Verb	Http Body	Http Headers	Content Type	Manage																						
Execute Command	/home/hyperion/hyperion_amb	ms	number	restrict IPs	---	body args	format like: [{"name": "A"}]	---Please select---	Del																						

Rechts dann auf *Add* klicken und danach oben auf “*Add Bridge Device*” zum Abspeichern.

Jetzt sagt ihr Alexa “Suche nach neuen Geräten” oder macht es manuell über die App und euer neu hinzugefügtes Gerät sollte jetzt erscheinen und ansteuerbar sein.

## FERTIG.

Möchtet ihr nun auch noch Effekte und Farben steuern, müsst ihr für jedes ein extra Skript anlegen.

Skript Beispiel zum Starten des Effektes “Red mood blobs” wäre:

```
#!/bin/bash
sudo hyperion-remote -e "Red mood blobs"
```

Skript Beispiel zum Anschalten der Farbe Blau

```
#!/bin/bash
sudo hyperion-remote -c blue
```



Farbe Schwarz zum Ausschalten. Gibt natürlich auch paar andere Möglichkeiten.

Wenn ihr auch noch gute Ideen für die Steuerung und coole Skripte habt, freue ich mich über Kommentare :).



## 5 COMMENTS

Reply

**SEBASTIAN**

*Apr 05, 2019*

Hi,

coole Sache und vielen Dank für die Anleitung!

Laut dem Webinterface kann ich mein Hyperion schalten, aber mit dem "on" Skript passiert nichts am LED Strip .. vermutlich weil nur der hyperiond restartet wird (der läuft bei mir aber sowieso automatisch mit Boot)?

Kannst Du etwas ausführlicher Beschreiben, wie Du die Konfiguration zum Farben umschalten gebaut hast? Ich habe die Skripte jetzt unter "Color Items" – ist das falsch?

Meine Alexa findet das Gerät leider nicht. Braucht es noch die Hue App, um die emulierte Bridge zu finden?





ambilight.jar -> https://www.technikbrennpunkt.de/ambilight-ueber-amazon-alexa-steuern-1376/

Ich würde empfehlen einen symlink zu erzeugen, um "ha-bridge.jar" auf das heruntergeladene File zu linken.

Reply

**MAIK**

*Okt 26, 2018*

Hallo

Hab Es so wie beschrieben ausgeführt ha bridge kann ich starten.das ambilight als Gerät wird auch erkannt. Wenn ich jetzt testen möchte kommen grüne Popups gehe ich auf aus geht ambilight aus gehe ich auf on bleibt es aus obwohl die Wege zum Script stimmen

Hast du mal einen tip

Reply

**PATRICK**

*Okt 26, 2018*

Was steht denn in den PopUps drin? Fehlermeldung oder Request Executed: success...

Wie sieht dein EIN-Script aus? Hast du die richtige Hyperion Config dort angegeben (also stimmt Name der Config etc) ?

Reply

**RAJIV**

*Jul 10, 2018*

Hi Patrik,

ich versuche auf meine RPi 3B das mit OSMC (frisch installiert wegen Java) das ganze durchzuführen damit ich meine Amibilight mit alexa steuern kann, klappt leider nicht.

Bekomme kein ha-bridge installiert :/

Welches betriebssystem hast du?

Reply

**PATRICK**

*Jul 10, 2018*

Ich habe es mit Raspbian Stretch und Libreelec ohne Probleme einrichten können.

Bekommst du eine Fehlermeldung bei der Installation oder was genau funktioniert



...minut, was eine recht hohe Zahl ergibt. Ich bin mir nicht sicher, ob diese Zahl durch andere Dienste belegt ist.

Wenn du mir genauere Informationen noch geben kannst, kann ich dir eventuell helfen.

## LEAVE A REPLY

Deine E-Mail-Adresse wird nicht veröffentlicht. Erforderliche Felder sind mit \* markiert.



Your name \*



Your e-mail \*



Your website



Your comments \*

SUBMIT COMMENT



---

## Impressum und Datenschutzerklärung