Amazon Dash Button Integration with HomeSeer HS3 using Dasher and MQTT

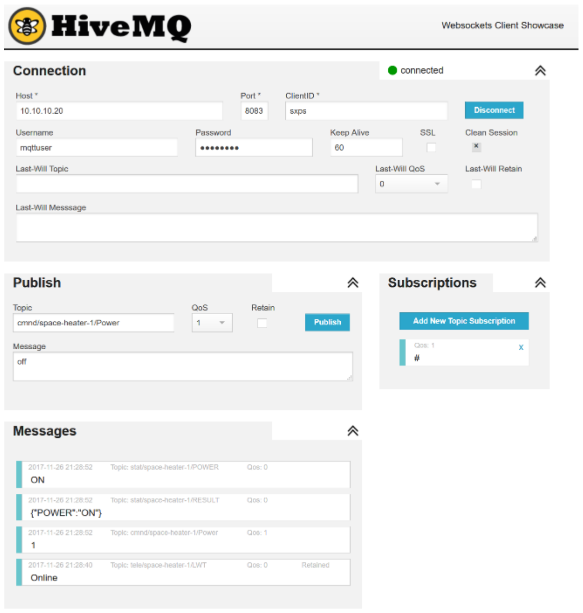
This guide shows how to integrate the Amazon Dash Button with HomeSeer HS3 using the MQTT Plugin and the Dasher app.

I recommend getting yourself familiar with the basics of MQTT. A very good explanation is available here.

Step 1:

Install, configure and test an MQTT Broker.

- The Mosquitto 1.4 MQTT broker works well with the HS3 MQTT Plugin. In this example it is installed on an Ubuntu 16.04 server. Here is a guide to follow to set it up. A Raspberry Pi can also be used as an MQTT broker. Here's a guide.
- Test and confirm that Publishing and Subscribing works on the broker – as shown in the guide.
- Make sure to configure Websockets on the broker. It will be useful later in troubleshooting by allowing the use of a web based MQTT client to see the messages to/from the broker. A good example is the open source HiveMQ websocket client.



Connection — connected

Host *: 10.10.10.20
Port *: 8083
ClientID *: sxps
Username: mqttuser
Password: ••••••••
Keep Alive: 60
SSL: ☐
Clean Session: ☒
Last-Will Topic:
Last-Will QoS: 0
Last-Will Retain: ☐
Last-Will Messsage:

Publish

Topic: cmnd/space-heater-1/Power
QoS: 1
Retain: ☐
Message: off

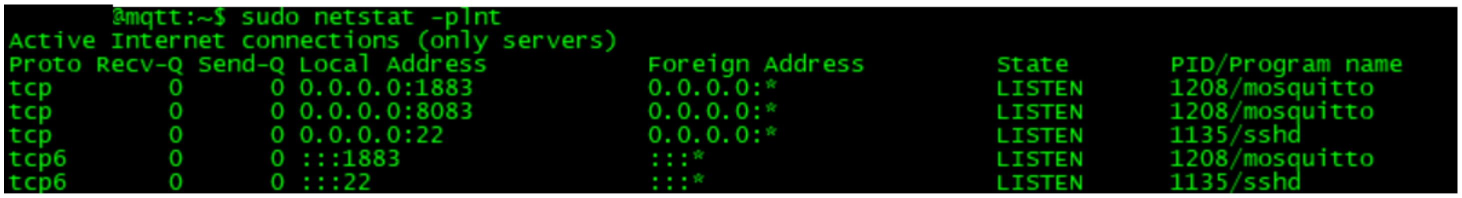
Subscriptions

Qos: 1
#

Messages

```
2017-11-26 21:28:52   Topic: stat/space-heater-1/POWER    Qos: 0
ON
2017-11-26 21:28:52   Topic: stat/space-heater-1/RESULT   Qos: 0
{"POWER":"ON"}
2017-11-26 21:28:52   Topic: cmnd/space-heater-1/Power    Qos: 1
1
2017-11-26 21:28:40   Topic: tele/space-heater-1/LWT      Qos: 0     Retained
Online
```

Confirm that the MQTT Broker is listening on the standard (1883) and websocket (8083) ports.



```
[redacted]@mqtt:~$ sudo netstat -plnt
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:1883            0.0.0.0:*               LISTEN      1208/mosquitto
tcp        0      0 0.0.0.0:8083            0.0.0.0:*               LISTEN      1208/mosquitto
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1135/sshd
tcp6       0      0 :::1883                 :::*                    LISTEN      1208/mosquitto
tcp6       0      0 :::22                   :::*                    LISTEN      1135/sshd
```

Step 2:

Configure the Dash Button on WiFi network

Use the directions on Amazon to partially set-up the Dash button – but **do not** perform the step to select the product to be ordered with the Dash. Exit the set-up process at this point.

Step 3:

Install and set-up Dasher.

Dasher is an app that runs on Raspberry Pi or Linux that detects when the Dash button is pressed and either sends an HTTP message or runs a local command. The HTTP method can be used for IFTTT integration. We will use the the function of running a local command to publish an MQTT topic which can be used to trigger events in HS3.

Install Dasher on Ubuntu 16.04

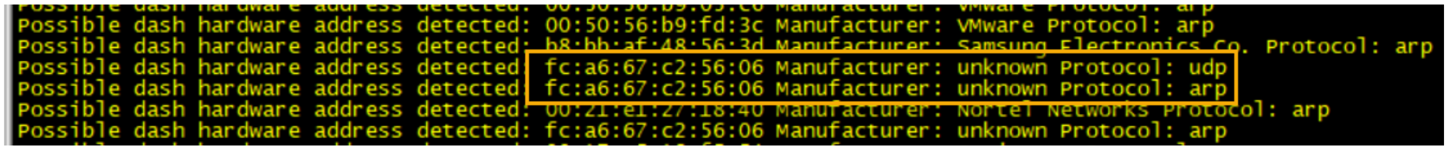
```
sudo apt-get install libpcap-dev
sudo apt-get install npm
sudo apt-get install git
git clone https://github.com/maddox/dasher.git
cd dasher
npm install
```

Find the Dash button's MAC address:

Issue the command:

```
sudo ~/dasher/script/find_button
```

..and press the Dash button once (or more if needed). A list of addresses being detected will show on the console and the Dash button's MAC may be seen as *Manufacturer: Amazon Technologies Inc.* or *unknown Protocol* as seen below:



```
Possible dash hardware address detected: 00:50:56:b9:fd:3c Manufacturer: VMware Protocol: arp
Possible dash hardware address detected: b8:bb:af:48:56:3d Manufacturer: Samsung Electronics Co. Protocol: arp
Possible dash hardware address detected: fc:a6:67:c2:56:06 Manufacturer: unknown Protocol: udp
Possible dash hardware address detected: fc:a6:67:c2:56:06 Manufacturer: unknown Protocol: arp
Possible dash hardware address detected: 00:21:e1:27:18:40 Manufacturer: Nortel Networks Protocol: arp
Possible dash hardware address detected: fc:a6:67:c2:56:06 Manufacturer: unknown Protocol: arp
```

The Dash button's MAC address can also be found in your router or DHCP server's DHCP lease table. Look for the most recent DHCP lease after the button is pressed. The Dash button is active only for the few seconds after it is pressed and will respond to a “ping” during this time - but not when the button's light is off.

<u>Create the Dasher config file and local command script</u>:

Create a file called `config.json` in the `/dasher/config` directory with the content below:

```
{"buttons":[
  {
    "name": "Name_to_identify_Dash_Button_1",
    "address": "12:34:56:78:9a:bc",
    "cmd": "/path/to/command or script #1"
  }
  {
    "name": "Name_to_identify_Dash_Button_2",
    "address": "12:34:56:78:9a:bd",
    "cmd": "/path/to/command or script #2"
  }
]}
```

Dash button MAC address found from previous step

Dash button MAC address found from previous step

The local command that would be invoked by the Dasher config file is a simple script called `FamilyRoom-Light-Off-MQTT.sh` that contains the following:

```
#!/bin/bash
mosquitto_pub -h 10.10.10.20 -t "familyroom/light" -m "0" -u "username" -P "password" &>/dev/null &
```

In the example above, the Mosquitto client is installed on the same host as Dasher and the command publishes a topic of `familyroom/light` with payload `0` to MQTT broker `10.10.10.20`

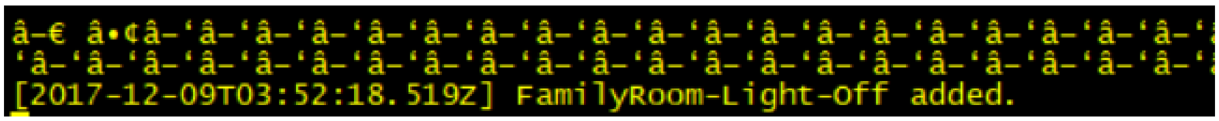
Note: A separate command/script will be requiredfor each Dash button.

<u>Start Dasher:</u>

Run this command in the `/dasher` directory to start Dasher.

```
sudo npm run start
```

You should see something like this on the console:



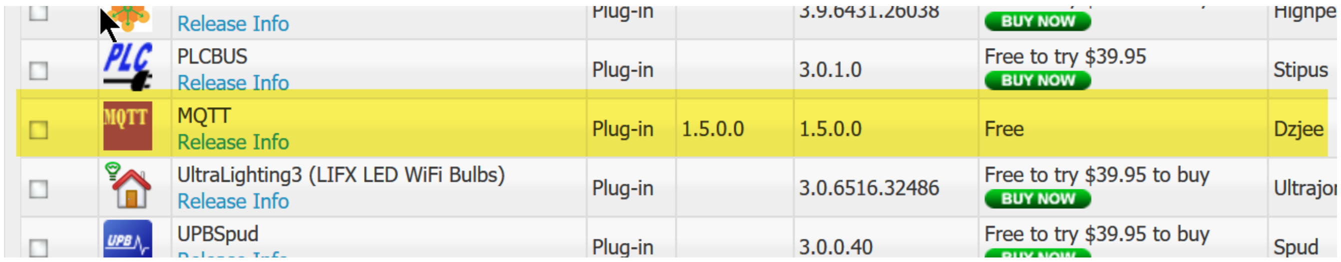
```
â–€ â•¢â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'
'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'â–'
[2017-12-09T03:52:18.519Z] FamilyRoom-Light-Off added.
```

Step 4:

HS3 Configuration:

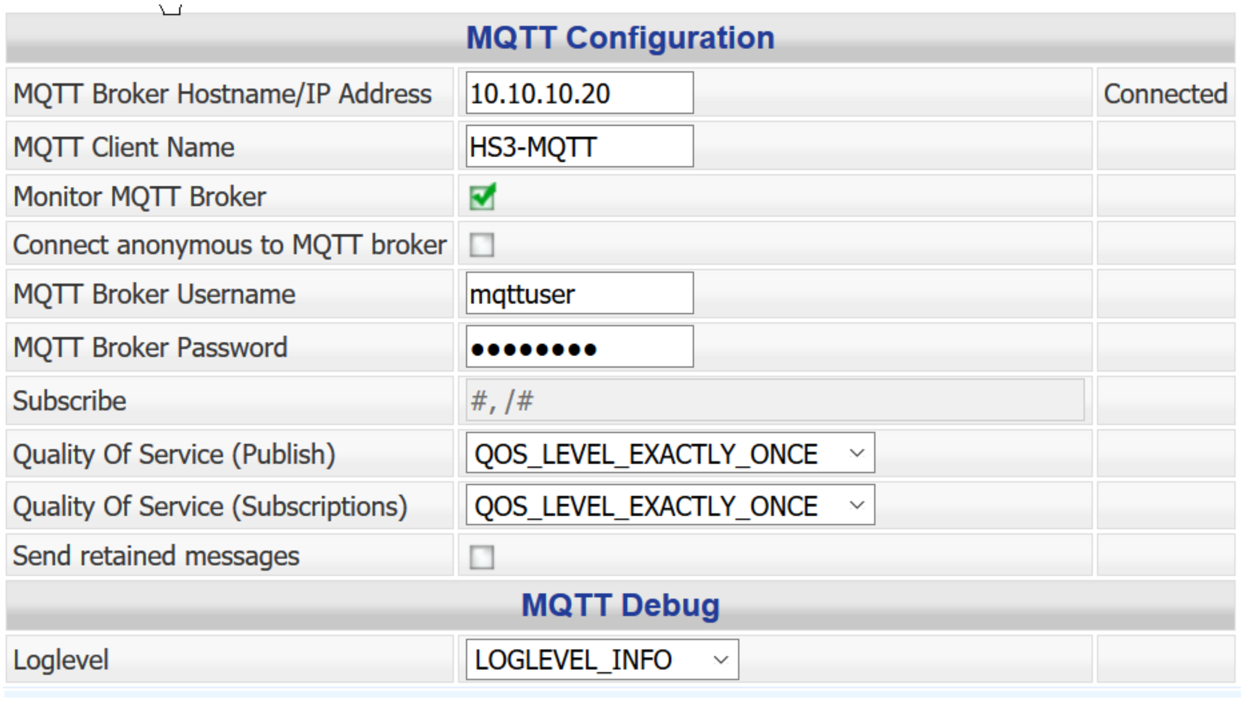
- ➢ Install the HS3 MQTT Plugin and configure initial setup.

The MQTT Plugin is free and available under the “Lighting and Primary Technology” section:



☐	Release Info	Plug-in		3.9.6431.26038	BUY NOW	Highpe
☐	PLCBUS Release Info	Plug-in		3.0.1.0	Free to try $39.95 BUY NOW	Stipus
☐	MQTT Release Info	Plug-in	1.5.0.0	1.5.0.0	Free	Dzjee
☐	UltraLighting3 (LIFX LED WiFi Bulbs) Release Info	Plug-in		3.0.6516.32486	Free to try $39.95 to buy BUY NOW	Ultrajo
☐	UPBSpud	Plug-in		3.0.0.40	Free to try $39.95 to buy	Spud

Here is a sample MQTT Configuration:



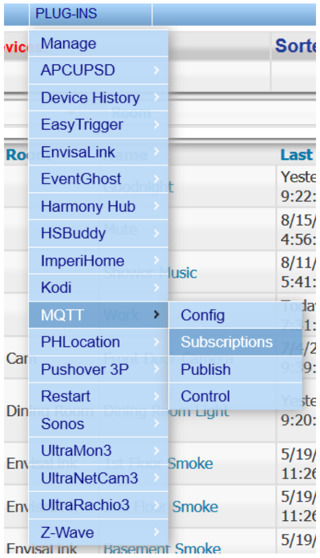
MQTT Configuration

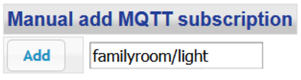
MQTT Broker Hostname/IP Address	10.10.10.20	Connected
MQTT Client Name	HS3-MQTT	
Monitor MQTT Broker	✓	
Connect anonymous to MQTT broker	☐	
MQTT Broker Username	mqttuser	
MQTT Broker Password	●●●●●●●●	
Subscribe	#, /#	
Quality Of Service (Publish)	QOS_LEVEL_EXACTLY_ONCE	
Quality Of Service (Subscriptions)	QOS_LEVEL_EXACTLY_ONCE	
Send retained messages	☐	

MQTT Debug

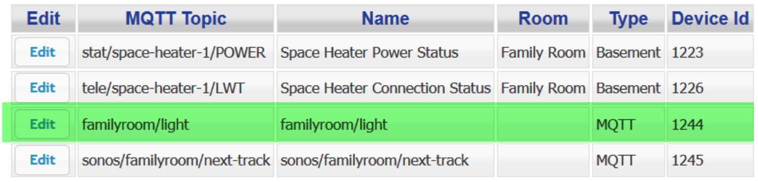
Loglevel	LOGLEVEL_INFO	

- Add a Manual Subscription for the topic that will be published by the script with the Mosquitto client command in the previous steps. This will create a virtual MQTT device with the name of the Subscription.



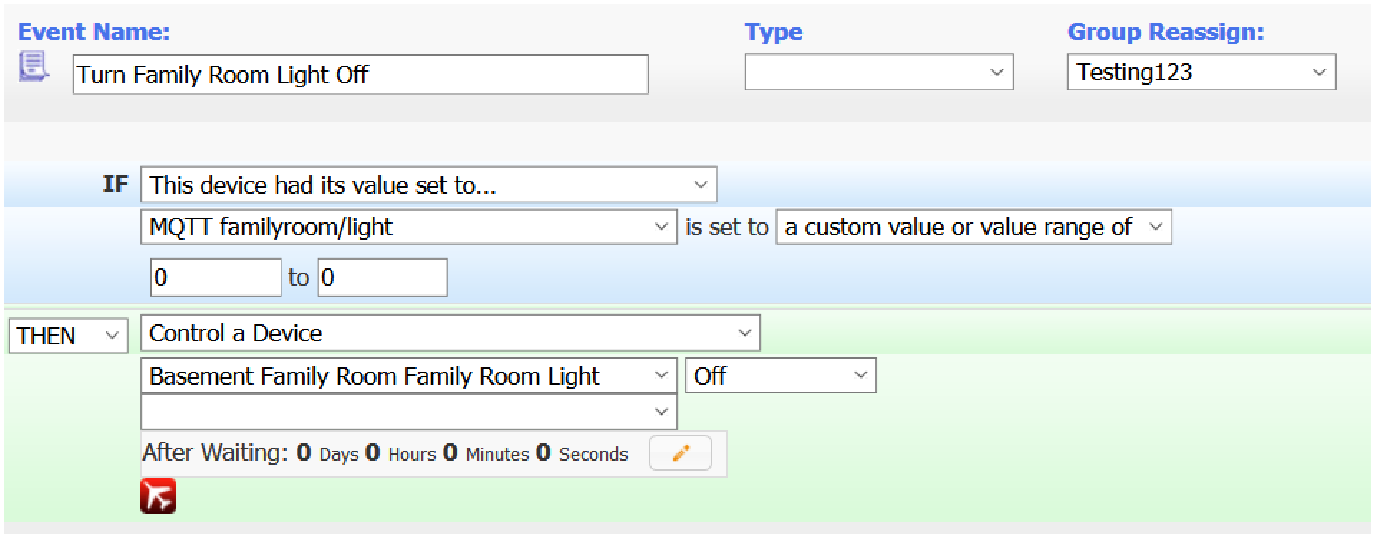

Manual add MQTT subscription

Add: familyroom/light



Edit	MQTT Topic	Name	Room	Type	Device Id
Edit	stat/space-heater-1/POWER	Space Heater Power Status	Family Room	Basement	1223
Edit	tele/space-heater-1/LWT	Space Heater Connection Status	Family Room	Basement	1226
Edit	familyroom/light	familyroom/light		MQTT	1244
Edit	sonos/familyroom/next-track	sonos/familyroom/next-track		MQTT	1245

- Create an Event In HS3 that triggers an action based on the value set to the MQTT Subscription virtual device. Here, the Family Room Light which is a Zwave HS3 Dimmer, is being turned off when a topic of “familyroom/light” and payload of “0” is published - when the Dash button is pressed.



Event Name: Turn Family Room Light Off
Type:
Group Reassign: Testing123

IF This device had its value set to...
MQTT familyroom/light is set to a custom value or value range of
0 to 0

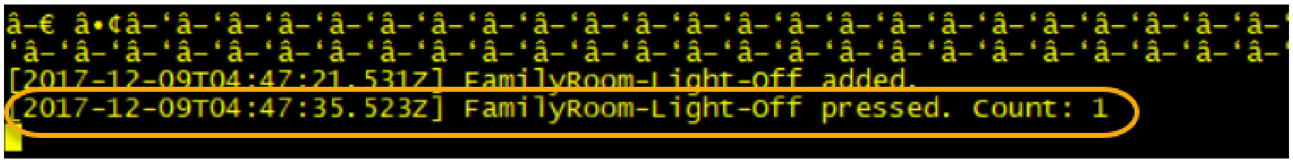
THEN Control a Device
Basement Family Room Family Room Light — Off

After Waiting: 0 Days 0 Hours 0 Minutes 0 Seconds

Step 5:

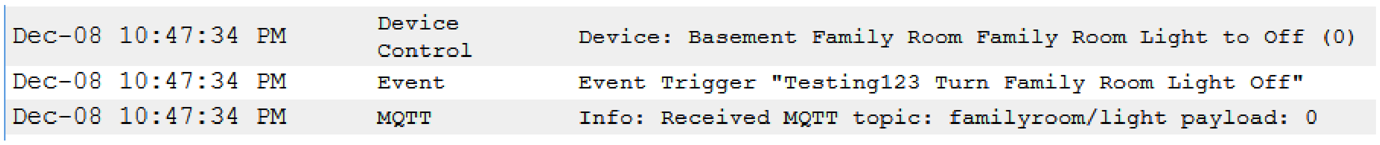
Press the Dash Button to initiate an HS3 Event

You should see something like this on the console:



```
â–€ â•¢â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘
‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘â–‘
[2017-12-09T04:47:21.531Z] FamilyRoom-Light-Off added.
[2017-12-09T04:47:35.523Z] FamilyRoom-Light-Off pressed. Count: 1
```

This will trigger Dasher to detect the Dash button press and run the shell script with the Mosquitto client MQTT publish command – which publishes the topic and payload that HS3 is subscribed to. This, in turn, triggers the Event to turn off the Family Room Light. Here's what the log looks like 4 seconds after the Dash button is pressed:



```
Dec-08 10:47:34 PM    Device Control    Device: Basement Family Room Family Room Light to Off (0)
Dec-08 10:47:34 PM    Event             Event Trigger "Testing123 Turn Family Room Light Off"
Dec-08 10:47:34 PM    MQTT              Info: Received MQTT topic: familyroom/light payload: 0
```

Enjoy!!

-taylormia