

Alloy Fusion Hub User Guide

Last revised May 23, 2025

[1- About the Alloy Fusion Hub](#)

[2- Connecting to the Alloy Fusion Hub](#)

[3- Accessing Hubview](#)

[4- Managing the Z-Wave® network with Hubview](#)

[Adding a device](#)

[Removing a Device](#)

[Controlling devices](#)

[Descriptor](#)

[Capabilities](#)

[Configuration](#)

[Predefined configuration](#)

[Configuration properties](#)

[Device health](#)

[Rediscovering the capabilities of a device](#)

[Updating device firmware](#)

[Healing the Z-Wave® network](#)

[Factory and Z-Wave® Reset](#)

[Factory reset](#)

[Z-Wave® reset](#)

[Associations](#)

[Replication](#)

[5- Managing the WiFi connection](#)

[6- Activating indicators](#)

[7- SmartStart](#)

[8- Defining scenes](#)

[9- Scripting devices](#)

[Coding a script](#)

[Naming a script](#)

[Default variable settings](#)

[Setting variables](#)

[Remembering the history of a variable](#)

[Script start time](#)

[Defining and scheduling a job](#)

[The lifecycle of a script](#)

[Errors and failures](#)

[About syntax](#)

[Sample scripts](#)

[Managing scripts](#)

[10- Testing devices](#)

[11- Command classes](#)

[Supported command classes](#)

[Controlling command classes](#)

[Association / Multichannel Association](#)

[Basic](#)

[Battery](#)

[Binary Switch](#)

[Central Scene](#)

[Configuration](#)

[Device Reset Locally](#)

[Doorlock](#)

[Indicator](#)

[Meter](#)

[Multichannel](#)

[Multilevel Sensor](#)

[Multilevel Switch](#)

[Notification](#)

[Thermostat Fan Mode](#)

[Thermostat Fan State](#)

[Thermostat Mode](#)

[Thermostat Operating State](#)

[Thermostat Setpoint](#)

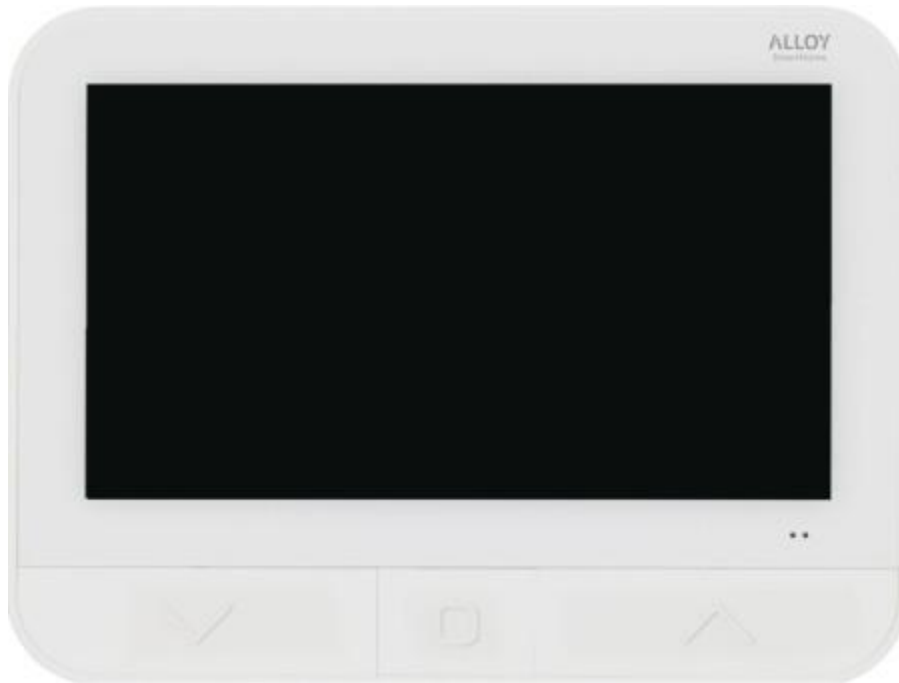
[User Code](#)

[12- Supervision](#)

1- About the Alloy Fusion Hub

The Alloy Fusion Hub is a **Security Enabled Z-Wave Plus® central static controller**.

It is responsible for managing a network of Z-Wave®-compliant devices from a fixed location.



The Alloy Fusion Hub provides extensive support for the following Z-Wave®-compliant devices:

- Thermostats
- Secure door locks
- Notification sensors
- Binary and multilevel power switches
- Sirens
- Scene controllers

All Z-Wave®-compliant devices from different manufacturers and product categories can be a part

of the Z-Wave® network controlled by the Alloy Fusion Hub. At a minimum, the Alloy Fusion Hub will control them via basic on/off commands.

Note that non-battery powered nodes can act as repeaters regardless of manufacturers.

2- Connecting to the Alloy Fusion Hub

The Alloy Fusion Hub (henceforth the Hub) is accessible via WiFi and cellular networks.

To access your Hub via WiFi, you first need to connect to your Hub's own access point. This will allow you to edit initial WiFi connection settings:

- Start the Hub by connecting it to its power supply.
- Wait for the hub to be fully started.
- From the hub's panel, navigate to Settings (click on the tab in the top-left corner).
- In Settings, navigate to Networking/Wi-Fi/Configure WiFi
- Select an SSID and enter its passphrase
- On your computer, select the same WiFi access point you configured your Hub to connect to.

3- Accessing Hubview

Hubview is the name of the Web application served by the Hub which provides access to the Hub's functions, including access to the devices it controls.

To access Hubview:

- Find the serial number of the Hub (something like *AH01235c613321011701*) on the back of the Hub, on the bar code label. You may already know it as it is the name of Hub's access point (see above). The serial number also appears on a label attached to the underside of the Hub.
- Open a Web browser and enter the address `https://<serial number>.local:4443` (for example, `https://AH01235c613321011701.local:4443`¹)
- Follow the instructions on the login page to create a password.

¹ If your computer is running Windows, you might need to use the Hub's IP address instead, for example, <https://192.168.0.33:4443>. See your router's documentation for instructions on how to see the list of IP addresses it allocates. You will recognize the Alloy Fusion Hub in the list by its serial number.

The screenshot shows the Alloy SmartHome System dashboard. In the top left corner is the 'ALLOY SmartHome' logo. In the top right corner is the text 'System dashboard'. The main content area features a light purple box with the following elements: a heading 'Enter the first 5 digits of the DSK, or the Ethernet MAC address' followed by a subtext 'The DSK is on the label at the back of your Alloy SmartHome Hub'; a text input field; a label 'Choose a password' next to a text input field with the placeholder 'At least 8 characters'; a label 'Confirm password' next to a text input field with the placeholder 'Enter the password again'; and a link '[Set password](#)' at the bottom.

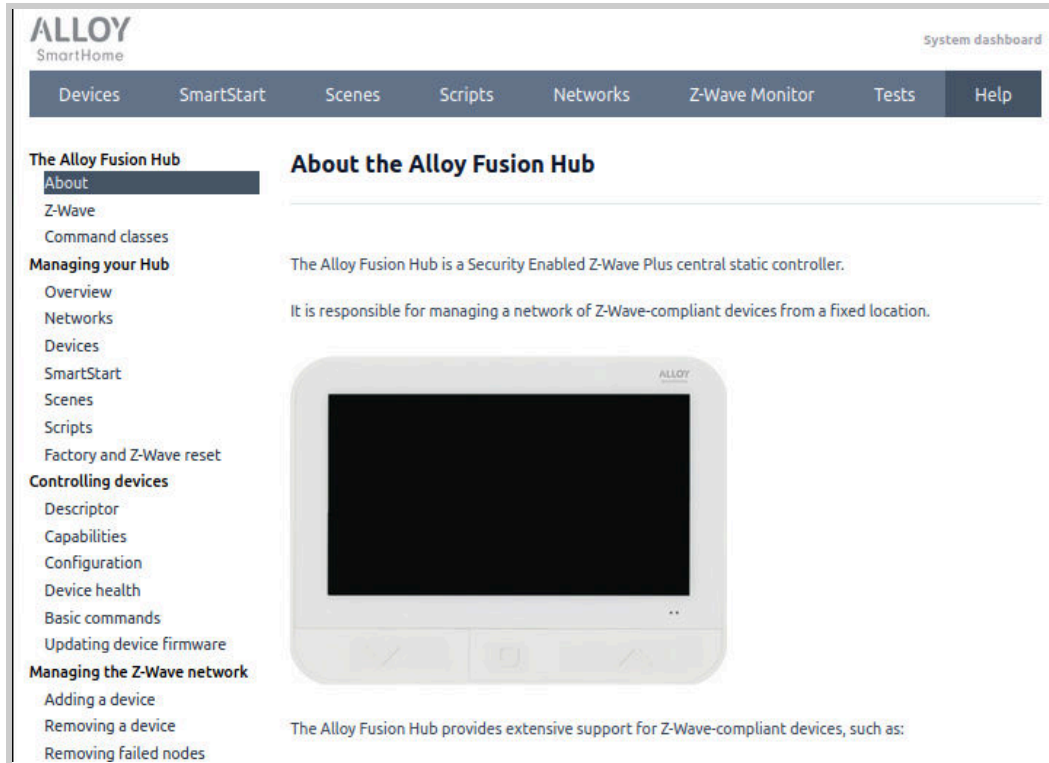
Once the password has been successfully created, you will be taken to the login screen where you will be asked to use it.

The screenshot shows the Alloy SmartHome System dashboard. In the top left corner is the 'ALLOY SmartHome' logo. In the top right corner is the text 'System dashboard'. The main content area features a light purple box with the following elements: a label 'Password' next to a text input field; a link '< a href="#">Login' to the right of the input field; and a link '< a href="#">Reset password' centered below the input field.

You can reset your password at any time when asked to login into your Hub. You will be taken to the password creation page.

Hubview gives comprehensive access to the Hub's functionality.

After logging in, select the **Help** tab for documentation on how to use Hubview to manage your Alloy Fusion Hub.



4- Managing the Z-Wave® network with Hubview

Adding a device

When the **Devices** tab is selected, an **Add a device** button is displayed under the list of already added devices.

ALLOY SmartHome System dashboard

Devices SmartStart Scenes Scripts Networks Z-Wave Monitor Tests Help

Icon	ID	Name	Status
●	1	Central controller	awake
●	10	Thermostat general v2	
●	13	Door lock	awake
○	(1)	Thermostat general v2	virtual

[Add a device](#) [Remove a device](#)

Descriptor		Capabilities	Configuration
About	Dsk	20402-13609-19969-26133-06601-31890-56710-42027	
	Firmware version	1.3	
	Manufacturer specific	manufacturer id is 57, product id is 8, product type id is 17	
	Network key	s2 unauthenticated	
	Security	secure	
	Z-wave module serial number	5817	
Attributes	Battery percent	0	
	Humidity	54.0	
	Power source	battery	
	Temperature	unit is c, temperature is 21.0	
	Thermostat cooling setpoint	unit is f, temperature is 74.0	
	Thermostat fan mode	mode is circulation	
	Thermostat fan state	state is off	
	Thermostat heating setpoint	unit is f, temperature is 70.0	
	Thermostat mode	mode is auto	
	Thermostat operating state	state is heating	
	Thermostat supported modes	modes is [off and heat and cool and auto and energy heat and energy cool]	
	Thermostat supported setpoints	setpoint types is [heating and cooling and energy save heating and energy save cooling]	
	Zwave version	firmware version is 1.3, protocol version is 4.61, hardware version is 1, library type is routing end node, other firmware versions is [1.4]	
	Tags	upstairs, bedroom set	

To add a device that is not yet part of a Z-Wave® network, press the **Add a device** button and put a device into "add mode" (follow the instructions in the documentation that came with the device).

ALLOY SmartHome System dashboard

Devices SmartStart Scenes Scripts Networks Tests Help

Icon	ID	Name	Status
●	1	Central controller	
●	45	Portable remote controller	asleep
●	47	Switch multilevel	awake
●	63	Thermostat general v2 + 2	awake
●	73	Notification sensor (temperature, home security, luminance)	asleep
●	80	Notification sensor (temperature, home security, humidity, water)	asleep
●	85	Door lock	awake

[Put a device into add mode...](#) [Cancel](#)

Descriptor		Capabilities
About	Dsk	none
	Firmware ids	0 and 1
	Firmware version	1.31.0
	Manufacturer specific	manufacturer id is 912, product id is 1, product type id is 1
	Network key	unknown
	Security	secure
Attributes	Power source	external
	Version	firmware version is 7.19, protocol version is 7.19, hardware version is 1, library type is bridge controller, other firmware versions are 7.18 and 1.0
Tags	upstairs, bedroom set	

Adding a node can be canceled by pressing the "Cancel" button.

If the device is an S2 authenticated device, in order to securely include it, you will be asked to provide its PIN, which are the first 5 digits of the device's DSK.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

45 Portable remote controller

47 Switch multilevel

63 Thermostat general v2 + 2

73 Notification sensor (temperature, home security, luminance)

80 Notification sensor (temperature, home security, humidity, water)

11111

Send

Descriptor

Capabilities

About

Dsk

Firmware ids

Firmware version

Manufacturer specific

Security

Z-wave module serial number

Attributes

Power source

Zwave version

Tags

upstairs, bedroom

set

The Alloy Fusion Hub should complete the process automatically and, after a few seconds, report success and the security level at which the device was added. The device should now appear in the list of devices.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

45 Portable remote controller

47 Switch multilevel

63 Thermostat general v2 + 2

73 Notification sensor (temperature, home security, luminance)

80 Notification sensor (temperature, home security, humidity, water)

85 Door lock

Add a device

Remove a device

Remove failed nodes

Heal network

Z-Wave reset

A door lock device was added with id 85. Security is secure.

Descriptor

Capabilities

About

Dsk

Firmware ids

Firmware version

Manufacturer specific

Security

Z-wave module serial number

Attributes

Power source

Zwave version

Tags

upstairs, bedroom

set

If the device was not added securely, the application reports the fact. The application also reports when adding a device fails or times out.

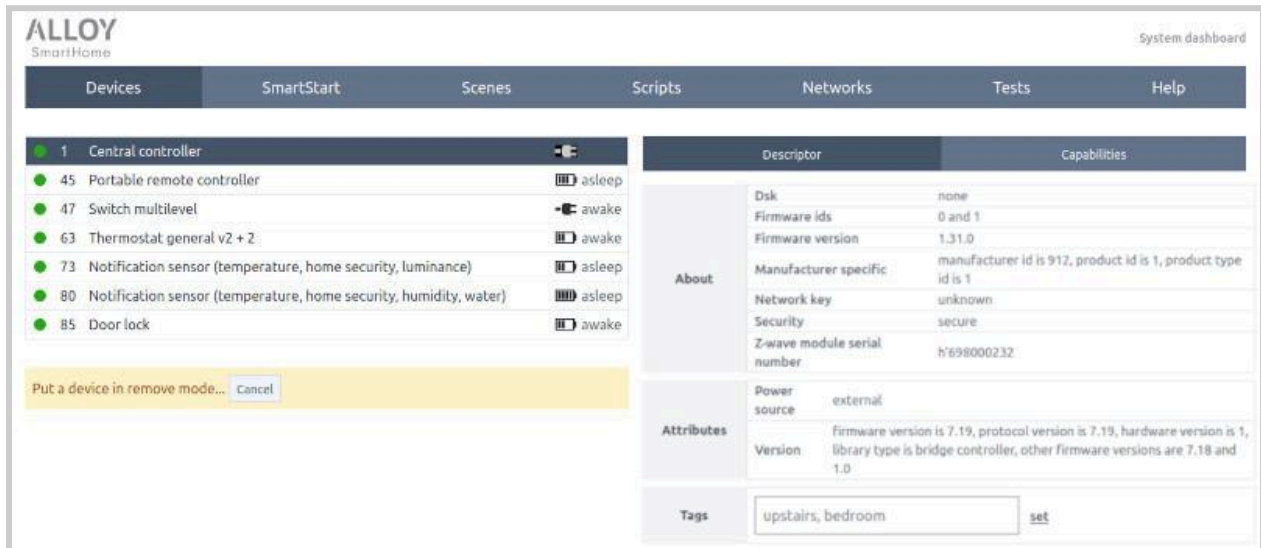
The Hub automatically discovers the capabilities of the devices added to its network when they are added and every time it starts up. Should the devices acquire new capabilities via firmware updates, these capabilities, to the extent that they are supported by the Hub, will be discovered the next time the Hub is restarted.

8

Removing a Device

When the **Devices** tab is selected, a **Remove a device** button is displayed under the list of already added devices.

To remove a device from the Z-Wave® network controlled by the Hub, press the button.



Removing a device can be canceled by pressing the “Cancel” button.

Put one of the listed devices (other than the “Gateway”) in “remove mode” (follow the instructions in the documentation that came with the device).

The Hub should complete the process automatically and, after a few seconds, report success and remove the device from the list.

The screenshot shows the ALLOY SmartHome System dashboard. At the top, there's a navigation bar with tabs: Devices, SmartStart, Scenes, Scripts, Networks, Tests, and Help. The 'Devices' tab is active. Below the navigation bar, there's a list of devices with their status (asleep or awake) and a battery icon. A green notification bar at the bottom left states: 'The door lock device with id 85 was removed'. On the right, there's a detailed view for a selected device, showing 'About' and 'Attributes' sections. The 'About' section includes fields for Dsk, Firmware ids, Firmware version, Manufacturer specific, Security, and Z-wave module serial number. The 'Attributes' section includes Power source and Zwave version. Below these, there's a 'Tags' section with a text input field containing 'upstairs, bedroom' and a 'set' button.

Descriptor	Capabilities
Dsk	none
Firmware ids	0 and 1
Firmware version	1.28.2
Manufacturer specific	manufacturer id is 912, product id is 1, product type id is 1
Security	secure
Z-wave module serial number	h'698000232

Attributes	Capabilities
Power source	external
Zwave version	firmware version is 7.19, protocol version is 7.19, hardware version is 1, library type is bridge controller, other firmware versions is [7.18 and 1.0]

Tags: upstairs, bedroom

The Hub's web interface also reports an error whenever removing a device failed or timed out.

Controlling devices

When selecting a device, you can look at the descriptor or the capabilities of the root device or of one of its end points, if it has any.

Descriptor

The *descriptor* of a device lists properties of the device as well as operational attributes that are visible to, or under the control of, the Alloy Fusion Hub.

Each device has the following properties, each with a value possibly unique to the device:

- Its DSK (device specific key)
- Firmware version
- Home id (uniquely identifying the network the device is on)
- Manufacturer-specific data (manufacturer id, product type is and product id)
- Security (the security level at which the device operates as per its network key)
- Serial number

ALLOY SmartHome System dashboard

Devices SmartStart Scenes Scripts Networks Z-Wave Monitor Tests Help

● 1	Central controller		awake
● 10	Thermostat general v2		
● 13	Door lock		awake
○ (1)	Thermostat general v2		virtual

[Add a device](#) [Remove a device](#)

Descriptor		Capabilities	Configuration
About	Dsk	20402-13609-19969-26133-06601-31890-56710-42027	
	Firmware version	1.3	
	Manufacturer specific	manufacturer id is 57, product id is 8, product type id is 17	
	Network key	s2 unauthenticated	
	Security	secure	
	Z-wave module serial number	5817	
Attributes	Battery percent	0	
	Humidity	54.0	
	Power source	battery	
	Temperature	unit is c, temperature is 21.0	
	Thermostat cooling setpoint	unit is f, temperature is 74.0	
	Thermostat fan mode	mode is circulation	
	Thermostat fan state	state is off	
	Thermostat heating setpoint	unit is f, temperature is 70.0	
	Thermostat mode	mode is auto	
	Thermostat operating state	state is heating	
	Thermostat supported modes	modes is [off and heat and cool and auto and energy heat and energy cool]	
	Thermostat supported setpoints	setpoint types is [heating and cooling and energy save heating and energy save cooling]	
	Zwave version	firmware version is 1.3, protocol version is 4.61, hardware version is 1, library type is routing end node, other firmware versions is [1.4]	
	Tags	upstairs, bedroom	set

Note that some devices may not report the values of one or more of their properties.

The attributes of a device depend on the nature of the device.

For example, a *multilevel power switch* will have a *level* attribute whereas a *thermostat* will have a *temperature*, *heating setpoint*, etc.

Some attributes can be changed on the device or via Hub's web interface (e.g. the thermostat heating setpoint), others are only reported to the Hub by the device (temperature).

The last changed attribute, as reported by the device, is highlighted.

Devices can be assigned tags. Tags are currently used by scripts to help identify the devices to which scheduled operations apply, for example, raising the temperature setpoint on all "downstairs" thermostats (see Scripts).

If a device is battery-powered and the battery level is low, a message to replace batteries soon or now is displayed.

ALLOY SmartHome System dashboard

Devices | SmartStart | Scenes | Scripts | Networks | Z-Wave Monitor | Tests | Help

Descriptor	Capabilities	Configuration
Replace batteries soon		
About	DSK	22053-40911-13082-38535-32627-59599-43597-35265
	Firmware Ids	42400 and 33033
	Firmware version	2.20.1
	Manufacturer specific	manufacturer id is 297, product id is 19925, product type id is 33033
	Network key	s2 access control
	Security	secure
	Z-Wave module serial number	h'F082C0FFFE699352
Battery percent	21	

Buttons: Add a device, Remove a device, Update firmware (target 0)

Capabilities

The capabilities of a device is the list of attributes of the device that can be obtained or changed (i.e. controlled) by the Alloy Fusion Hub.

ALLOY SmartHome System dashboard

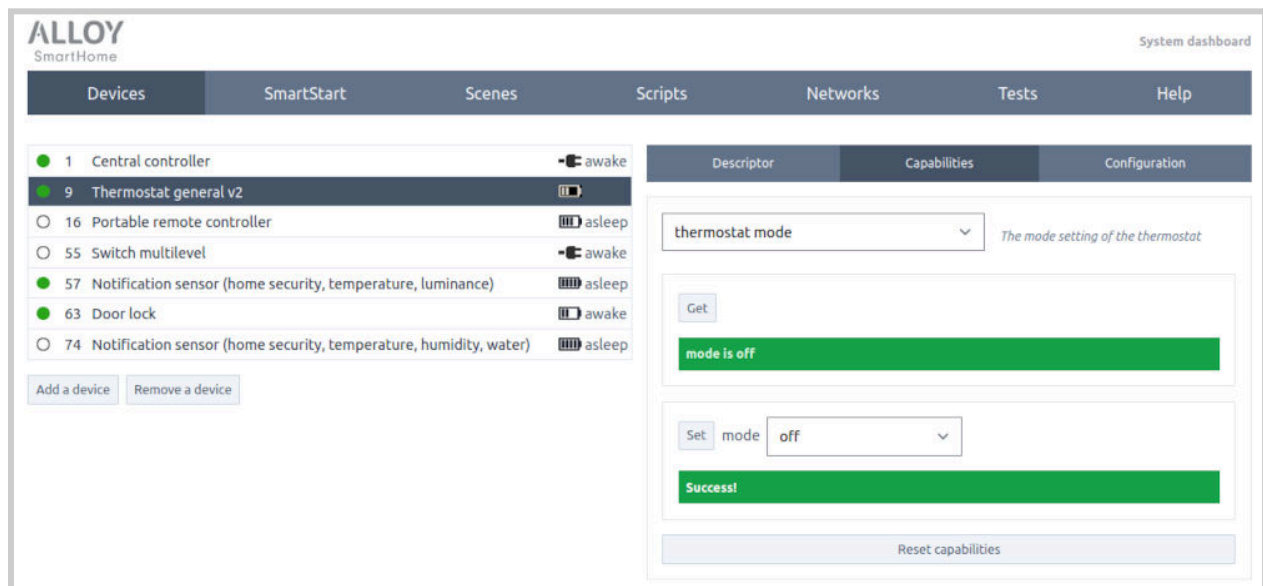
Devices | SmartStart | Scenes | Scripts | Networks | Tests | Help

Descriptor	Capabilities
level	The on/off level of the multilevel switch
Get Set duration level or Choose one	
Reset capabilities	

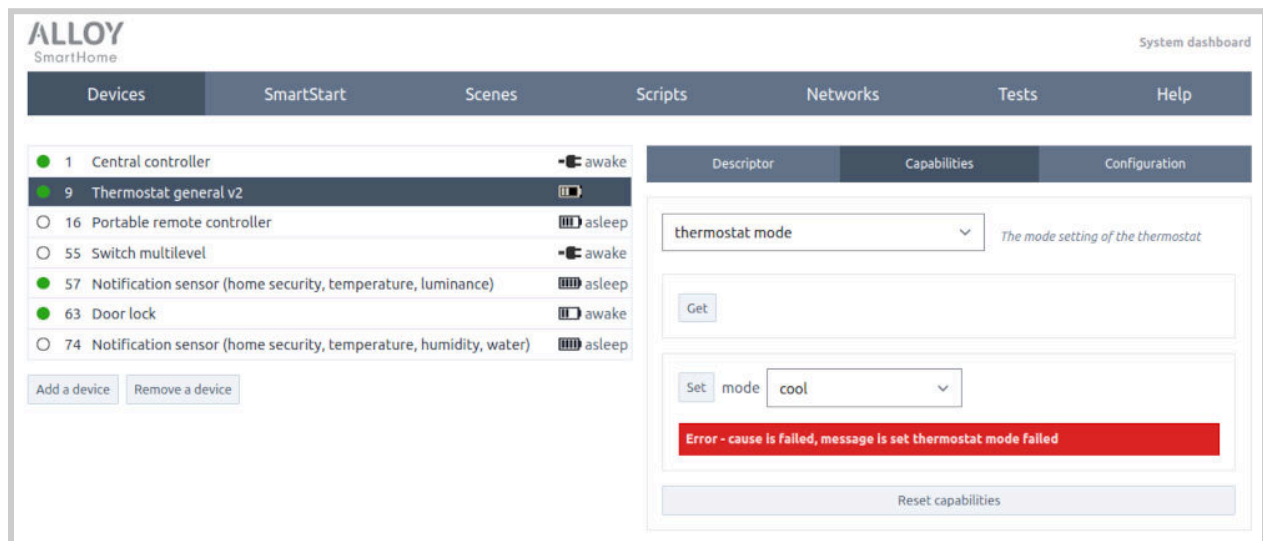
Buttons: Add a device, Remove a device

When selecting an attribute from the drop-down menu, the application will display a short text describing the attribute, plus, typically, a button to “Get” its value, and a “Set” button to change its value if it can be changed. Either button might be accompanied by input fields if getting/setting the attribute is parameterized.

When getting an attribute value, the value is displayed if getting it from the device was successful. If setting an attribute to a value is successful, a "success" message is also displayed.



If setting or getting an attribute value fails, either because an attribute parameter is not valid or the operation on the device failed, an error message is displayed.



Text input fields for attribute parameters may show placeholder text in dim grey to guide structured input. Examples of such placeholders and their meanings:

- "2,4,6" - Any list of integers - for example, "1,2,3"
- "0,...,10" - A list of bounded integers - for example, "2,4,5,10"

- “text OR 0x01,0x02,0x03” - Normal text or list of hexdecimals - for example, “energy safe” or “0x02, 0x0A”

When a device is asleep, requests to get and set its attributes are put in a queue. When the device periodically wakes up, it processes its queued commands and provides a response for each one.

The queued commands for a sleeping device are displayed. Associated responses are shown whenever they arrive.

The screenshot shows the ALLOY SmartHome System dashboard. On the left, a list of devices is displayed with their status (awake or asleep). The device '54 Notification sensor (home security, temperature, humidity, water)' is highlighted as asleep. On the right, the 'Capabilities' section for this device shows a 'battery percent' attribute. A 'Get' button has been clicked, resulting in a yellow message: 'The request was queued'. Below this, a 'Queued requests' table shows the request 'get battery percent' was made '0 sec ago' and is scheduled to be answered 'in 23 hrs'.

When asked	Request	Answer	When answered
0 sec ago	get battery percent		in 23 hrs

If you so desire, you can delete the records of all queued requests that have been answered by the device.

The screenshot shows the ALLOY SmartHome System dashboard. On the left, a list of devices is displayed with their IDs, names, and status (awake/asleep). The device with ID 54, 'Notification sensor (home security, temperature, humidity, water)', is highlighted. Below the list are buttons for 'Add a device' and 'Remove a device'. On the right, the 'Descriptor' tab is active, showing a dropdown menu for 'battery percent' and a 'Get' button. A yellow message box states 'The request was queued'. Below this is a table of 'Queued requests'.

When asked	Request	Answer	When answered
37 secs ago	get battery percent	100	0 sec ago

Note that only the requests queued during the current session (since the last login) are kept track of.

Configuration

Most devices can be configured by setting the values of their configuration parameters. A device will typically support multiple, numbered, and possibly named, parameters.

Each parameter has a factory-set (or resettable) default value. For example, a door lock may have a parameter number 3 which is the relock time in seconds, with a default value of 60.

Certain devices, but not all, can provide the list of configuration parameters they support and the properties of these parameters (default value, acceptable range, byte size, and whether their values can be changed.) They may also support the ability to reset en masse all their configuration parameters to their default values.

The Alloy Fusion Hub may provide predefined configurations for some devices. A predefined configuration is a subset of their configuration parameters, each with a value typically other than its default. A predefined configuration is either applied when the device is paired (this is not under user control), or it is applied manually by pressing the *Set configuration* button. Pressing the *Reset to defaults*, resets the parameters in the defined configuration to their default device values.

If a device supports configuration parameters, a Configuration tab will be visible for that device, giving you access to its *Predefined configuration* (if available) and/or its *Configuration properties* (but only if the device is not a wake-up device).

Predefined configuration

The predefined configuration of a device is a list of configuration parameters, possibly empty, each with a name, number, description, target setting and default value.

The screenshot shows the ALLOY SmartHome System dashboard. On the left, a list of devices is displayed with their status (awake or asleep). On the right, a table lists configuration parameters for a selected device, showing the descriptor, current setting, default value, and a status (set, reset, or unknown).

Descriptor	Capabilities	Configuration
1	Central controller	awake
45	Portable remote controller	asleep
47	Switch multilevel	awake
63	Thermostat general v2 + 2	awake
73	Notification sensor (temperature, home security, luminance)	asleep
80	Notification sensor (temperature, home security, humidity, water)	asleep
86	Door lock	asleep

Descriptor	Capabilities	Configuration
1	Silent mode (1 is high, 2 is low, 3 is off)	setting is 2 default is 1 set
3	Relock time (secs)	setting is 60 default is 30 reset
4	Wrong code entry limit	setting is 5 default is 5 reset
5	Language (1=English, 2=Spanish, 3=French)	setting is 1 default is 1 reset
7	Shut down time after wrong codes (secs)	setting is 10 default is 60 reset
8	Operating mode (0=normal, 1=vacation, 2=privacy)	setting is 0 default is 0 reset
11	One touch locking (0 is off)	setting is 0 default is 0 ?
12	Privacy button (0 is off)	setting is 0 default is 0 reset

When applying a device's configuration, an attempt is made to set the parameters to their target values, whereas, when resetting a device's configuration, an attempt is made to set the parameters to their default values.

The status of a configuration parameter is either "Set", "Reset" or "?" (unknown). Note that when the target setting and the default value are the same, the status of "Set" or "Reset" are synonymous.

When setting or resetting the configuration of a wake-up device, it may take a while until it takes effect. In the meanwhile, the status of each parameter is unknown.

Configuration properties

The configuration parameters of the device are listed, each with its number, name, acceptable range, default value, byte size, and whether its value can be changed (i.e. it is writable vs read-only). The current value of a parameter can be retrieved by pressing the *get* link and, if allowed, reset by pressing the *default is* link, or changed by inputting another value and pressing *set*.

If this is supported by the device, the configuration parameters can be reset all at once by pressing the *Reset all to defaults*. A dialog asks for confirmation.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1

Central controller

awake

9

Thermostat general v2

awake

16

Portable remote controller

asleep

19

Notification sensor (temperature, luminance, home security)

asleep

32

Door lock

asleep

54

Notification sensor (home security, temperature, humidity, water)

asleep

55

Switch multilevel

awake

Add a device

Remove a device

Descriptor

Capabilities

Configuration

Defined configuration

Configuration properties

1: Volume

from 1 to 3

default is 2

size is 1

writable

get

value is 3

set

Changed

2: Auto Relock

from 0 to 255

default is 0

size is 1

writable

get

value is 0

set

3: Relock time

from 10 to 180

default is 30

size is 1

writable

get

4: Wrong Code Entry Limit

from 3 to 10

default is 5

size is 1

writable

get

5: Language

from 1 to 3

default is 1

size is 1

writable

get

7: Shut down time

from 10 to 180

default is 60

size is 1

writable

get

8: Operating mode

from 0 to 2

default is 0

size is 1

writable

get

11: One Touch Locking

from 0 to 255

default is 255

size is 1

writable

get

12: Privacy Button

from 0 to 255

default is 0

size is 1

writable

get

13: Lock Status LED

from 0 to 255

default is 0

size is 1

writable

get

28: Expiring Pin Code Enabled Time

from 0 to 255

default is 0

size is 1

writable

get

Reset all to defaults

For devices with limited support for managing their configurations, parameters are not initially listed but must be added one by one before they can be changed or reset. Please refer to the device’s documentation for the list of configuration parameters it supports.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1

Central controller

awake

9

Thermostat general v2

asleep

16

Portable remote controller

asleep

19

Notification sensor (temperature, luminance, home security)

asleep

32

Door lock

awake

54

Notification sensor (home security, temperature, humidity, water)

asleep

55

Switch multilevel

awake

Add a device

Remove a device

Descriptor

Capabilities

Configuration

Configuration properties

1: Parameter 1

from 0 to 255

default is unknown

size is 1

writable

get

value is 2

set

4: Parameter 4

from 0 to 255

default is unknown

size is 1

writable

get

value is 2

set

5: Parameter 5

from 0 to 255

default is unknown

size is 1

writable

get

value is 2

set

6: Parameter 6

from 0 to 255

default is unknown

size is 1

writable

get

value is 0

set

Changed

Add parameter

1

lookup

For such devices, configuration parameters can only be reset one at a time.

Device health

Each device listed under the *Devices* tab is accompanied by a health indicator, a colored dot indicating how successfully the Alloy Fusion Hub controls, and communicates with, the device.

The screenshot shows the ALLOY SmartHome System dashboard. At the top, there's a navigation bar with tabs: Devices, SmartStart, Scenes, Scripts, Networks, Tests, and Help. The 'Devices' tab is active. Below the navigation bar, there's a list of devices. The device '124 Switch multilevel' is selected and highlighted. A tooltip for this device shows its health status: 'Health of device is excellent (click to see or hide details)'. To the right of the device list, there's a detailed view for the selected device, divided into sections: About, Attributes, Firmware uploads, and Tags. The 'About' section shows details like Dsk, Firmware ids, Firmware version, Manufacturer specific, Network key, Security, and Z-wave module serial number. The 'Attributes' section shows Level (off), Power source (external), and Version (firmware version is 2.11, protocol version is 7.14, hardware version is 1, library type is enhanced end node, other firmware versions are absent). The 'Firmware uploads' section has a button to 'Upload firmware file for target' and a 'Browse...' button. The 'Tags' section shows 'upstairs, bedroom' and a 'set' button.

Health is measured using data gathered over the last 24 hours, or since the Hub was turned on, whichever is more recent.

The possible colors of the health indicator are:

- Green - Excellent (no problems)
- Yellow - Good (a few problems)
- Orange - Poor (more than a few problems)
- Red - Bad (lots of problems)
- Black - Offline (the device is not responding to commands and is not notifying the Hub)
- White - TBD (health is unknown because there's no data yet for interacting with the device)

If the health indicator is neither white nor black, clicking on it will reveal a summary health report. A device's summary health report shows the number of successes and problems the Hub experienced when sending commands to the device during the last day, hour and minute. The summary report also shows the number of notifications received from the device.

Health indicators and displayed summary health reports are updated live.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1

Central controller

awake

45

Portable remote controller

asleep

107

Door lock

awake

122

Notification sensor (access control)

asleep

124

Switch multilevel

awake

	Last day	Last hour	Last minute
Successful commands	12	12	6
Failed commands	0	0	0
Timed out commands	0	0	0
All commands	12	12	6
Notifications	0	0	0
RSSI (dBm)	-69	-69	-73
Speed (kbit/sec)	100	100	100
Transmit time (msec)	1	1	1

127

Thermostat general v2

awake

128

Notification sensor (temperature, luminance, home security)

asleep

131

Notification sensor (home security, temperature, humidity, water)

asleep

Add a device

Remove a device

Descriptor

Capabilities

About

Dsk

45789-04617-00399-48681-62893-40796-51983-64547

Firmware ids

1031

Firmware version

2.11.0

Manufacturer specific

manufacturer id is 1108, product id is 7, product type id is 4

Network key

s2 authenticated

Security

secure

Z-wave module serial number

h'B0CE18A9831C

Attributes

Level

off

Power source

external

Version

firmware version is 2.11, protocol version is 7.14, hardware version is 1, library type is enhanced end node, other firmware versions are absent

Firmware uploads

Upload firmware file for target

0

Browse...

No file selected.

Tags

upstairs, bedroom

set

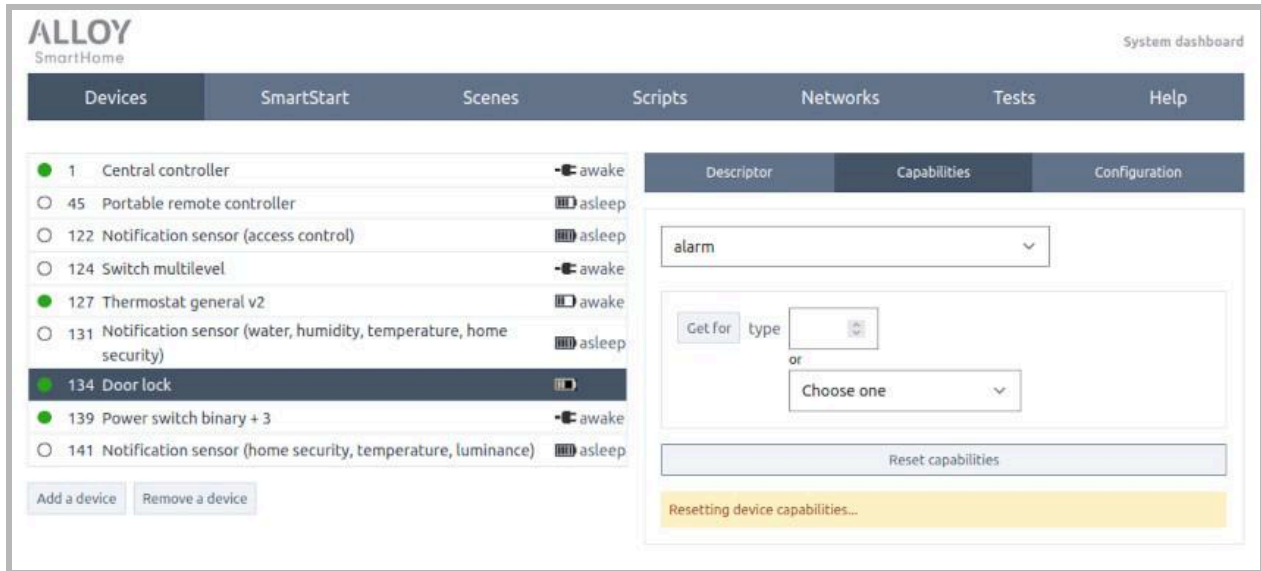
Clicking again on the health indicator will hide the summary health report. Health reports can be open on more than one device.

Rediscovering the capabilities of a device

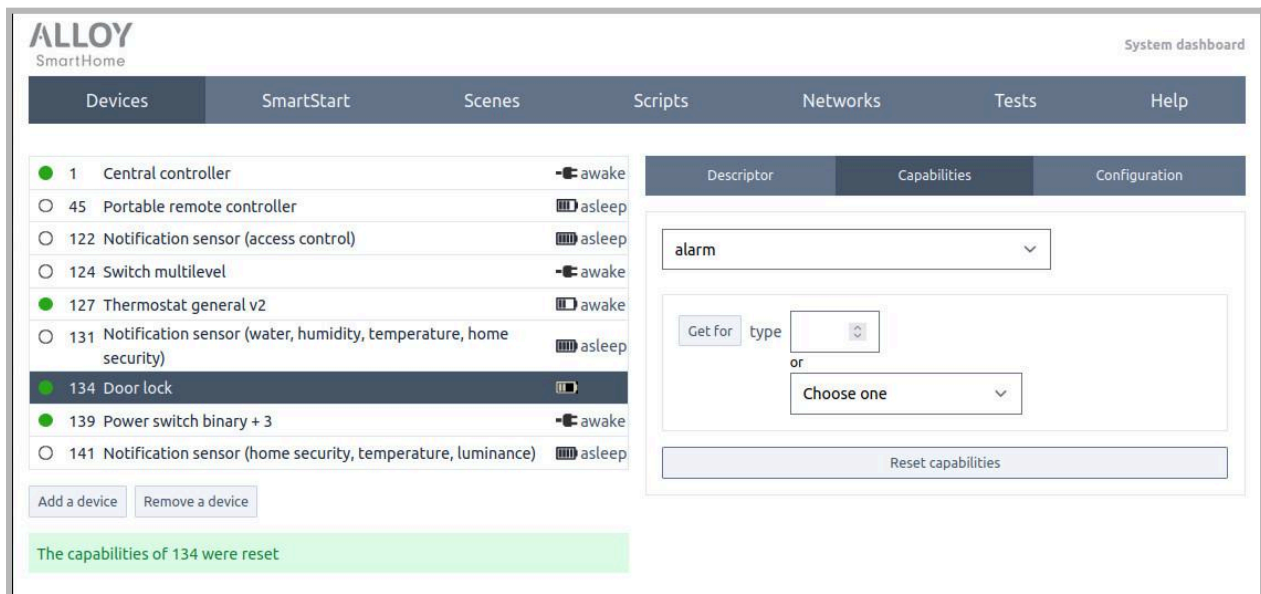
The capabilities of a device are discovered when it is first added to the network.

Should the need arise, it is possible to force a re-discovery of a device's capabilities by pressing the "Reset capabilities" button in the Capabilities tab for that device.

Please note that this function is only available for non-wake-up Z-Wave® devices.



Once the process has completed, a confirmation message will appear.



Updating device firmware

A firmware-upgradable device will expose one or more firmware “targets” for updating.

For example, a door lock might expose one target (always target 0) for updating firmware on its Z-Wave® chip, and expose another target for updating firmware on its touch panel.

If a device supports firmware updating and if it provides the information required to carry this out (manufacturer id and target firmware ids), a *Firmware Uploads* section will appear at the bottom of the device's Descriptor containing a file upload form.

The screenshot shows the ALLOY SmartHome System dashboard. On the left, a list of devices is displayed, including a Central controller, Portable remote controller, Switch multilevel, Thermostat, Notification sensors, and a Door lock (device 86). The Door lock is selected. On the right, the 'Descriptor' tab is active, showing details for the selected device. The descriptor is divided into three sections: 'About', 'Attributes', and 'Firmware uploads'.

Descriptor	Capabilities	Configuration
About	Dsk	none
	Firmware ids	42400 and 32770
	Firmware version	2.20.1
	Manufacturer specific	manufacturer id is 297, product id is 17920, product type id is 32770
	Network key	unknown
	Security	secure
Attributes	Z-wave module serial number	h'14B457FFFE0F6AFC
	Battery percent	49
	Door lock	unsecured
	Power source	battery
	Schedule type supported	number of slots year day is 1, number of slots week day is 0, number of slots daily repeating is 7
Attributes	User code capabilities	supported keypad keys is 0123456789, supported keypad modes are normal, vacation, privacy, and lockout, supported user id statuses are available, occupied, disabled, messaging, and passage, multi user code set supported? is false, multi user code report supported? is false, user code checksum supported? is false, admin code deactivation supported? is false, admin code supported? is true
	Version	firmware version is 2.20, protocol version is 7.13, hardware version is 2, library type is enhanced end node, other firmware versions are 43.19
Firmware uploads	Upload firmware File for target 0 Browse... No file selected.	
Tags	upstairs, bedroom set	

If you have access to firmware image files for your device (files with *.gbl* extensions), your first step will be to upload them one at a time to the Hub, after indicating the intended target for each one. Pressing **Browse** opens a file browser that filters for ".gbl" files. After selecting a file, the **Upload** button becomes visible. Press it to upload the file to the Alloy Fusion Hub.

ALLOY SmartHome System dashboard

Devices SmartStart Scenes Scripts Networks Tests Help

1	Central controller	awake
6	Power switch binary	awake
7	Power switch multilevel	awake
8	Door lock	awake
9	Thermostat general v2 + 2	awake
10	Portable remote controller	asleep

Add a device Remove a device

Update firmware (target 0)

Firmware image uploaded for target 0.

Descriptor Capabilities

About

Firmware ids	42400 and 32770
Manufacturer specific	manufacturer id is 297, product id is 17920, product type id is 32770
Security	secure
Serial number	h'14B457FFFE0F6AFC

Attributes

Battery percent	98
Doorlock	secured
Zwave version	firmware version is 2.13, hardware version is 2, library type is enhanced slave, other firmware versions is [43.19], protocol version is 7.13

Firmware uploads

Upload firmware file for target 0

Browse... YRD226_256_8002__V4.3.19_Dev.version.gbl Upload

Tags

upstairs, bedroom Set

Once firmware image files have been successfully uploaded, an **Update Firmware (target ...)** button will appear for each associated target. Pressing the **Update Firmware (target 0)** button will immediately start the firmware update process for target 0 of the device etc.

ALLOY SmartHome System dashboard

Devices SmartStart Scenes Scripts Networks Tests Help

1	Central controller	awake
45	Portable remote controller	asleep
47	Switch multilevel	awake
63	Thermostat general v2 + 2	awake
73	Notification sensor (temperature, home security, luminance)	asleep
80	Notification sensor (temperature, home security, humidity, water)	asleep
86	Door lock	asleep

Add a device Remove a device

Uploading firmware... (from fragment is 267, number of fragments wanted is 1)

Descriptor Capabilities Configuration

About

Dsk	46411-40341-29023-40497-07528-48410-48075-30723
Firmware ids	42400 and 32770
Firmware version	2.20.1
Manufacturer specific	manufacturer id is 297, product id is 17920, product type id is 32770
Security	secure
Z-wave module serial number	h'14B457FFFE0F6AFC

Battery percent 54

Doorlock unsecured

Power source battery

Schedule number of slots weak day is 1, number of slots weak day is 0, number

Be aware that the firmware update process might take many minutes.

A firmware update might fail for a number of reasons, including:

- The firmware image file is not valid for its target.

- The device target has already been updated to the same or a higher firmware version.
- The firmware update process was somehow unable to complete successfully.

The device will usually restart once the firmware update process has completed successfully.

Please note that when the Alloy Fusion Hub reboots, it intentionally forgets the firmware image files that were uploaded to it.

Healing the Z-Wave® network

Devices can become inaccessible because of incorrect routing data distributed over devices in the network.

Pressing the Heal network button will often resolve the issue.

The screenshot shows the Alloy SmartHome System dashboard. The top navigation bar includes 'Devices', 'SmartStart', 'Scenes', 'Scripts', 'Networks', 'Tests', and 'Help'. The 'Devices' section is active, displaying a list of devices with their status (asleep or awake). The 'Heal network' button is highlighted with a blue border. Below the device list are buttons for 'Add a device', 'Remove a device', 'Remove failed nodes', and 'Z-Wave reset'. The right sidebar shows details for the selected device, including 'About' (Dsk, Firmware ids, Firmware version, Manufacturer specific, Security, Z-wave module serial number) and 'Attributes' (Power source, Zwave version). The 'Tags' section shows 'upstairs, bedroom' with a 'set' button.

Descriptor	Capabilities
Dsk	none
Firmware ids	0 and 1
Firmware version	1.28.2
Manufacturer specific	manufacturer id is 912, product id is 1, product type id is 1
Security	secure
Z-wave module serial number	h'698000232
Power source	external
Zwave version	Firmware version is 7.19, protocol version is 7.19, hardware version is 1, library type is bridge controller, other firmware versions is [7.18 and 1.0]
Tags	upstairs, bedroom

The network healing process can take from a few seconds to minutes. The outcome is shown.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

45 Portable remote controller

47 Switch multilevel

63 Thermostat general v2 + 2

73 Notification sensor (temperature, home security, luminance)

80 Notification sensor (temperature, home security, humidity, water)

85 Door lock

asleep

awake

awake

asleep

asleep

awake

Add a device

Remove a device

Remove failed nodes

Heal network

Z-Wave reset

Network healing: done

Descriptor

Capabilities

About

Attributes

Tags

Dsk

Firmware ids

Firmware version

Manufacturer specific

Security

Z-wave module serial number

Power source

Zwave version

upstairs, bedroom

none

0 and 1

1.28.2

manufacturer id is 912, product id is 1, product type id is 1

secure

h'698000232

external

Firmware version is 7.19, protocol version is 7.19, hardware version is 1, library type is bridge controller, other Firmware versions is [7.18 and 1.0]

set

Factory and Z-Wave® Reset

Factory reset

To reset the Alloy Fusion Hub to its factory settings, you will first need to remove all devices. A **Factory reset** button will then become visible.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

Add a device

Remove a device

Remove failed nodes

Heal network

Z-Wave reset

Put in learn mode

Factory reset

Descriptor

Capabilities

About

Attributes

Tags

Dsk

Firmware ids

Firmware version

Manufacturer specific

Security

Z-wave module serial number

Power source

Zwave version

upstairs, bedroom

none

0 and 1

1.28.2

manufacturer id is 912, product id is 2, product type id is 1

secure

h'698000232

external

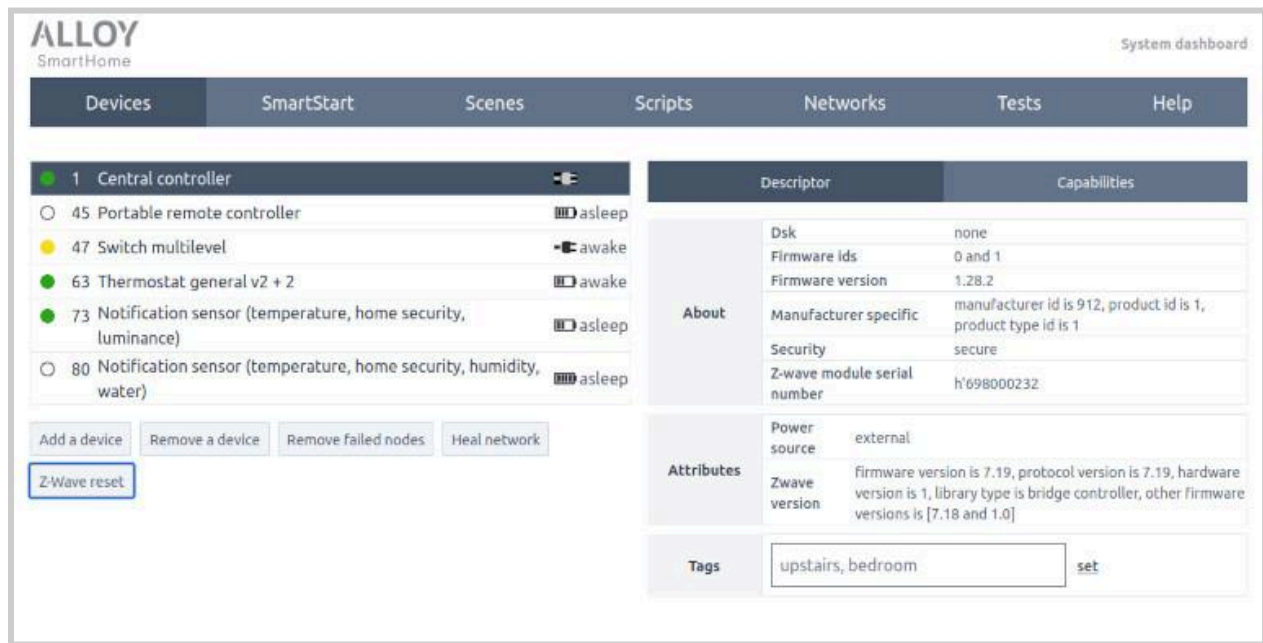
Firmware version is 7.16, protocol version is 7.16, hardware version is 1, library type is bridge controller, other Firmware versions is [7.18 and 1.0]

set

Pressing the **Factory reset** button will reset the Hub. The Hub will then reboot.

Z-Wave® reset

You can reset the Z-Wave® network without resetting the Hub to its factory settings by pressing the *Z-Wave® reset* button.



Resetting the Z-Wave® network will remove all Z-Wave® devices from the network, except for the Controller. The hub does not reboot.

Note that removed devices are not properly excluded and will need to be excluded or reset to their factory settings if they are to be re-included.

Associations

The Hub supports one Association Group. The group id is 1 and its name is Lifeline. The Hub sends the command *device_reset_locally_notification* to devices associated with the Lifeline group. The maximum number of associations for group 1 of the Hub is equal to 1.

Whenever a device is paired to the Hub, the Hub automatically sets a Lifeline association between itself, and only itself, and the device.

Notifications emitted by devices in the Z-Wave® network controlled by the Hub are propagated to Hubview, as well as to external applications connected via the Hub's secure MQTT API.

The supported types of notifications include:

- Smoke Alarm
- Water Alarm
- Access Control Event
- Home Security Event
- Siren Event
- Power Management Alarm

Attribute values shown for each device are updated upon receiving a notification for that device. Any attribute value updated from the last received notification is highlighted in yellow.

The screenshot shows the ALLOY SmartHome System dashboard. The top navigation bar includes tabs for Devices, SmartStart, Scenes, Scripts, Networks, Z-Wave Monitor, Tests, and Help. The 'Devices' tab is active, displaying a list of devices:

ID	Name	Icon	Status
1	Central controller		awake
10	Thermostat general v2		awake
13	Door lock		awake
(1)	Thermostat general v2		virtual

Below the device list are buttons for 'Add a device' and 'Remove a device'.

The right side of the dashboard displays detailed information for the selected device (Thermostat general v2). It is organized into three sections: About, Attributes, and Tags.

Section	Descriptor	Value
About	Dsk	20402-13609-19969-26133-06601-31890-56710-42027
	Firmware version	1.3
	Manufacturer specific	manufacturer id is 57, product id is 8, product type id is 17
	Network key	s2 unauthenticated
	Security	secure
	Z-wave module serial number	5817
Attributes	Battery percent	0
	Humidity	54.0
	Power source	battery
	Temperature	unit is c, temperature is 21.0
	Thermostat cooling setpoint	unit is f, temperature is 74.0
	Thermostat fan mode	mode is circulation
	Thermostat fan state	state is off
	Thermostat heating setpoint	unit is f, temperature is 70.0
	Thermostat mode	mode is auto
	Thermostat operating state	state is heating
	Thermostat supported modes	modes is [off and heat and cool and auto and energy heat and energy cool]
	Thermostat supported setpoints	setpoint types is [heating and cooling and energy save heating and energy save cooling]
Zwave version	Firmware version is 1.3, protocol version is 4.61, hardware version is 1, library type is routing end node, other firmware versions is [1.4]	
Tags	upstairs, bedroom	set

Note that associations can not be created or removed using Hubview.

Replication

The Alloy Fusion Hub is a Z-Wave® static central controller. It is responsible for managing a network of Z-Wave®-compliant devices from a fixed location.

It typically acts as the "primary" controller for the network, meaning that it is responsible for the addition (inclusion) and removal (exclusion) of devices from the network. Only primary controllers can add or remove devices. A controller maintains "routing tables" and uses them to reach the devices in its network.

The Hub adds a controller (from any manufacturer) to its network the same way it adds any new device. The routing tables on the Alloy Fusion Hub are automatically copied to the newly added, secondary, controller.

5- Managing the WiFi connection

The Hub implements a secure MQTT API accessible to authorized, remote applications. To be made accessible to such remote applications, your Hub needs to be connected to the Internet.

The Hub can connect to the Internet via a WiFi router.

ALLOY SmartHome		System dashboard
Devices SmartStart Scenes Scripts Networks Z-Wave Monitor Tests Help		
Interfaces	br0 and Ethernet and uap0 and WiFi and Cellular	
Current Interface	WiFi	
WiFi access	momo+ (strong) ▼ ↻ passphrase Select Forget	
Cellular access	service providers	wireless.twilio.com and hologram and super
	modem	QMI
	apn	wireless.twilio.com
	connection	disconnected
	esn	B0EBD500
	iccid	89012608522970083035
	imei	860516068913841
	imei svn	1F
	manufacturer	QUALCOMM INCORPORATED
	meid	A10000247C9EE1
	model	QUECTEL Mobile Broadband Module
	signal 4bars	0
	signal asu	99
	signal dbm	-128
MQTT connection	is up	
Networking status	hub ready	

To connect the Hub to a WiFi router, select the **Networks** tab and follow these instructions:

- Next to “WiFi access”, the SSIDs of WiFi access points visible to the Hub will populate a drop-down menu. Look in this menu for the access point you want to connect to. If you do not see it, press the “refresh” icon until it appears.
- Select the desired SSID, enter its passphrase (be aware that no verification is made to ensure it is the correct one!), and press Select.
- You will be reminded that this will disconnect you from your Hub, and that you may need to reload the page after the Hub has completed reconfiguring its WiFi access.

- Assuming the passphrase was correct, your Hub will connect to your WiFi LAN via the access point you just selected. The LED light will turn green after approximately 10 to 20 seconds and start blinking. This indicates that connection to your WiFi LAN was successful². To continue use of Hubview, you will need to configure your computer's WiFi access to the same SSID you set for your Hub.
- If connecting to your WiFi LAN was unsuccessful, possibly because you entered the wrong passphrase, disconnect and reconnect the power supply to restart the Hub, and then redo the procedure to connect your computer to the Hub's own access point (press the button under the steady orange LED for 5 seconds). When completed, reload Hubview, login, press the **Networks** tab and try again connecting to your WiFi LAN.
- You may need to wait up to two minutes for your computer to "see" Hubview over your WiFi LAN after switching your Hub to it.
- Windows computers need to have Apple Bonjour³ installed to see the Hub's *srh<serial number>.local* domain over your WiFi LAN⁴. However, Bonjour is not needed if you know the IP address your WiFi router assigned to your Hub. You can then access Hubview using the Hub's IP address as follows:
https://<ip address>:4443 (for example, *https://192.168.0.12:4443*).

6- Activating indicators

Some devices, the Alloy Fusion Hub being one of them, can be asked to physically identify themselves via some indicator⁵. They typically do so by flashing an LED, or by turning themselves on and off.

If a device is not a wake-up device and can identify itself, mousing over its "awake" label in the list of devices will cause a popup to show up, instructing you to click on the label to turn on or off the device's self-identification.

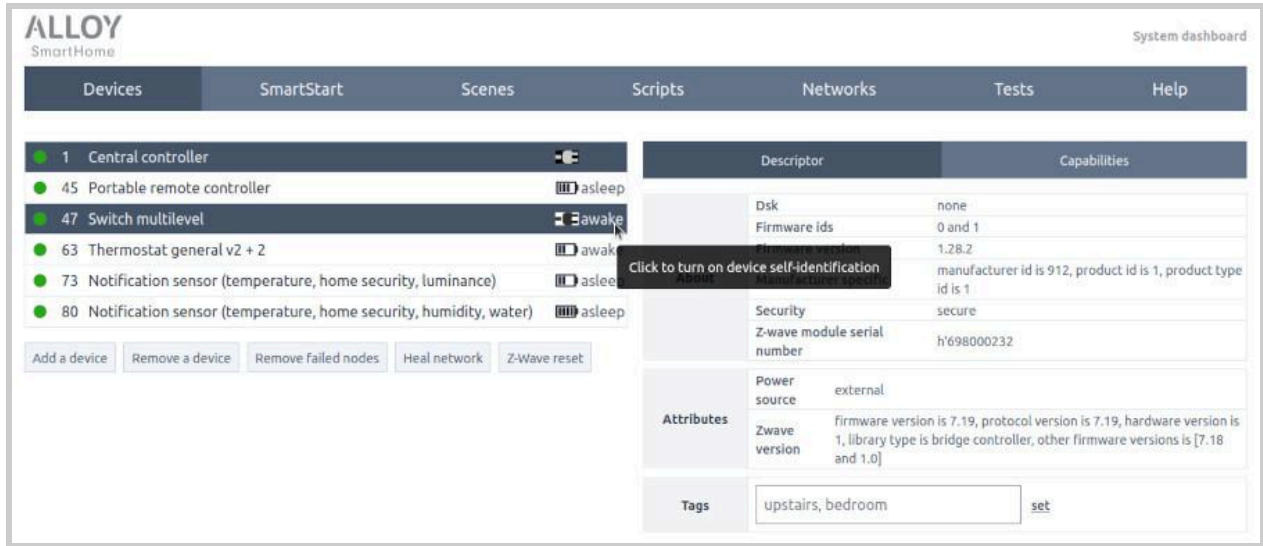
The Hub identifies itself by blinking its entire display 5 times.

² Note that if your LAN is not connected to the Internet, the Hub will reboot automatically after 15 minutes.

³ https://support.apple.com/kb/DL999?viewlocale=en_US&locale=en_US

⁴ The Alloy Fusion Hub uses the mDNS protocol to make known its *srh<serial number>.local* domain name over the local network. The Windows 10 and prior versions require Apple Bonjour installed to properly understand the mDNS protocol.

⁵ The device is asked by the Hub to activate its indicator by sending it the Indicator Set command, from the Indicator command class, with the Indicator ID set to 0x50 (identify).

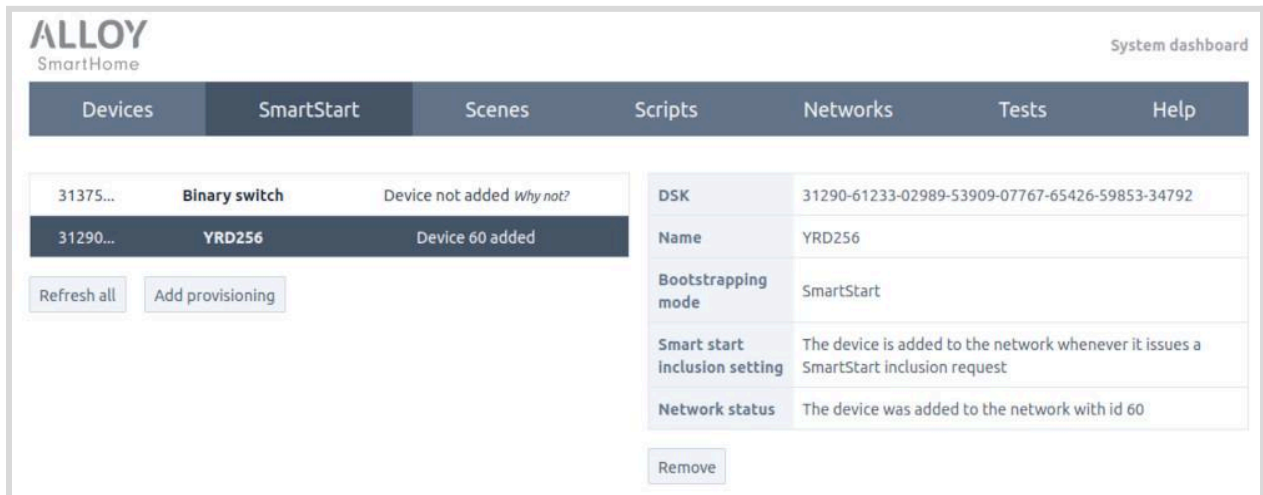


The "awake" label will pulse while the device is in self-identification mode. Multiple devices can be put into self-identification mode.

7- SmartStart

Z-Wave® SmartStart automates the secure addition of devices to the Z-Wave® network controlled by the Alloy Fusion Hub.

By giving the Hub a list of provisioned devices, you are instructing it to automatically add matching devices to the Z-Wave® network, if not already added, whenever they are detected by the Hub. Only one device can possibly match a given SmartStart provisioning.



A device is provisioned first clicking on the "Add provisioning" button and then editing the details of its provisioning. This is done by entering its 40-digit secure device key (DSK), accompanied by a meaningful name. Note that the DSK is composed of eight 5-digit numbers separated by dashes. If the device supports Z-Wave® Long Range, you can request that it be paired as a "Long range" device. Make sure the device supports it if you check the box. When done, press on the "Set" button.

The screenshot shows the ALLOY SmartHome System dashboard. The top navigation bar includes "Devices", "SmartStart", "Scenes", "Scripts", "Networks", "Tests", and "Help". The "SmartStart" tab is active. On the left, there is a table with two rows: one for a device with ID "31290..." and name "YRD256" (status: "Device 60 added") and another for a device with ID "00000..." and name "no name" (status: "Not set Why not?"). Below this table is a "Refresh all" button. On the right, there is a form for provisioning a device. It includes a "DSK" field with the value "00000-00000-00000-00000-00000-00000-00000-00000", a "Name" field with the value "no name", a "Long range" checkbox which is unchecked, and a "Set" button at the bottom.

To abandon the attempt at provisioning a device, don't press the "Set" button but press the "Refresh all" button instead. This will get rid of the "not set" device provisioning.

Once the device is provisioned, and assuming it has not yet been added to the Z-Wave® network, turn it on and after a minute or so, assuming the DSK was entered correctly, it should then automatically appear in the list of devices (accessed by clicking on the Devices button).

The status of the provisioned device, once added, will change from "Device not added" to "Device added". You may want to refresh the list of provisioned devices to see the change of status.

The screenshot shows the ALLOY SmartHome System dashboard. The top navigation bar includes "Devices", "SmartStart", "Scenes", "Scripts", "Networks", "Tests", and "Help". The "Devices" tab is active. On the left, there is a table with two rows: one for a device with ID "31375..." and name "Binary switch" (status: "Device not added Why not?") and another for a device with ID "31290..." and name "YRD256" (status: "Device 60 added"). Below this table are "Refresh all" and "Add provisioning" buttons. On the right, there is a table showing the provisioning details for the "Binary switch" device. The table has two columns: the first column lists the provisioning details, and the second column lists the values. The details are: "DSK" (31375-43976-25613-22634-59569-01900-09619-04340), "Name" (Binary switch), "Bootstrapping mode" (SmartStart), "Smart start inclusion setting" (The device is added to the network whenever it issues a SmartStart inclusion request), and "Network status" (The device is not in network). Below this table is a "Remove" button.

If you remove a provisioned device without first removing its SmartStart provisioning, the device will automatically be added back after a short delay, assuming it has not been powered off right after removal. To remove a device provisioning, select it from the list and press the "Remove" button under its details.

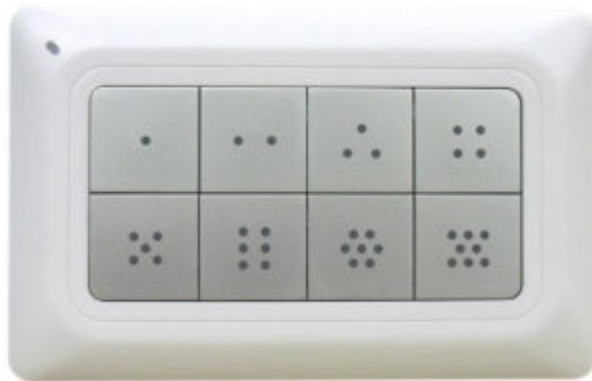
Note that a device provisioning can not be modified once it is set. To correct it, you will need to remove it and add it anew with the desired modifications.

Devices that support SmartStart ship with an DSK printed on the device or its packaging. The DSK is either printed in full or is encoded as a QR code with a printed PIN, or both.

If encoded as a QR code only, you will need to use a QR code reader/decoder app. There will usually be a DSK printed below the QR code. The PIN is the first five digits of the DSK. In the decoded QR code, find the PIN and the following 35 digits. It is the DSK that you will need to enter in groups of 5 digits, with dashes in between, to provision the device.

8- Defining scenes

Scenes define multi-device sequences of actions that can be triggered from a Scene Controller device or any other device that implements the Central Scene command class.



You can define and associate a sequence of actuations with each possible activation of a key on the activating device, or any device, for a given scene.

Each key on a typical Scene Controller is identified by a number, for example 1 to 8, corresponding to a Scene.

Each Scene can contain multiple Activation Sequences, each one associated with a specific pressing of the corresponding scene key on a specific, or any, scene activating device.

The supported key pressings can, depending on the activating device, include

- key pressed 1 time - a single click
- key pressed 2 times - a double click
- key pressed 3 times - a triple click
- key pressed 4 times - a quadruple click (!)
- key pressed 5 times - a quintuple click (!!)
- key held (for some time)
- key released (after being held)

All scenes are initially unnamed and contain no device Activation Sequences associated with scene activation devices.

You can name a Scene and define Activation Sequences associated with various key pressings on a specific scene activating device, or any such device.

An Activation Sequence is defined by adding, removing or re-ordering device Activations. There are two kinds of Activations, Attribute Activations and Configured Activations.

An Attribute Activation is the setting of an actuator attribute of a device via an attribute setting command. When adding an Attribute Activation, identify the target device, the attribute to be set, and the value(s) at which to set it.

ALLOY SmartHome System dashboard

Devices SmartStart **Scenes** Scripts Networks Tests Help

Scene 1 - Fiat lux [erase](#)

Scene name is Fiat lux

When by do

key held down any device

key pressed 1 time portable remote controller 45

key pressed 2 times

key pressed 3 times

key pressed 4 times

key pressed 5 times

key released

Set 83 - power switch binary 's switched to switched

Choose one

Done Cancel

A Configured Activation is the activation of a device that has a pre-configured level and dimming duration associated with the current scene. The device must support the Scene Activation command class to be the target of a Configured Activation. The **Add Configured Activation** button only appears if a Configured Activation can be added in the current context.

When adding a Configured Activation, you need only select a device from a list of devices supporting the Scene Activation command class.

ALLOY SmartHome System dashboard

Devices SmartStart **Scenes** Scripts Networks Tests Help

Scene 1 - Fiat lux [erase](#)

Scene name is Fiat lux

When by do

key held down any device

key pressed 1 time portable remote controller 45

key pressed 2 times

key pressed 3 times

key pressed 4 times

key pressed 5 times

key released

1 - set power switch binary 83 so that switched is on

Activate 83 - power switch binary

Done Cancel

The default configuration for such a device is to set target level "on" with a dimming duration of "instantly" upon scene activation. The default scene activation configuration of a device can be changed by setting its *scene actuation* attribute via Capabilities in the Devices tab.

The screenshot shows the ALLOY SmartHome interface. The 'Devices' tab is active. A list of devices is shown on the left, with '83 Power switch binary' selected. On the right, the 'Capabilities' section for the selected device is visible. It includes a 'scene actuation' dropdown menu. Below this, there are fields for 'Get for scene id', 'Set dimming duration' (with a dropdown set to 'instantly'), 'level' (with a dropdown set to 'on'), and 'scene id' (set to 3).

A device Activation may, at some point, become invalid. This typically happens when the targeted device is no longer included in the Alloy Fusion Hub's Z-Wave® network. A device Activation is indicated as invalid by having a line striking through its summary.

The screenshot shows the ALLOY SmartHome interface with the 'Scenes' tab active. A scene named '1 - Fiat lux' is selected. The 'When' section shows 'key pressed 2 times'. The 'by' section shows 'portable remote controller 45'. The 'do' section shows a disabled activation: '1 - set unknown 83 so that switched is on', which is crossed out with a red line. There are 'Add attribute activation' and 'Run' buttons.

An invalid device Activation is disabled and should be removed or edited to make it valid again.

9- Scripting devices

Scripts automate and coordinate changes to device attributes.

A script is a small piece of code that conditionally gets/sets attributes of paired devices. A script is organized into jobs that are scheduled or primed for execution once the script starts.

Scripts are added to the Hub for later activation. The start of a script is *scheduled* for execution the moment it is activated; a script does not necessarily start exactly when it is activated.

For example a script may be scheduled to start the first time 12PM is reached *after* it has been activated. Only when the script starts are its jobs primed for execution. This means that no job can execute until the script has started, and no script can start until it has been activated.

Scripts can be deactivated and removed from the Hub.

A script remains (de)activated across restarts of the hub. When the hub starts up, any script that was active prior to shutdown is scheduled for restart.

Coding a script

A script is written in a simple DSL (Domain-Specific Language).

A script's definition starts by stating its name. A script's name is meant to be unique among all scripts added to the Hub. If a script is added to a Hub that already has a script of that name, the old script is stopped and is replaced by the new, inactive one.

Naming a script

```
script "Energy saver" do
...
end
```

Default variable settings

The first lines of a script typically set default values for script's variables.

Note that *all* variables in a script are global to all jobs of that script, no matter how, when and where they are set.

```
default tooColdLevel is 65
default lowTemperatureLevel is 70
default highTemperatureLevel is 75
default allTstats is devices thermostat
default upstairsTstats is devices thermostat, only: ["upstairs"]
default downstairsTstats is devices thermostat, except: [5, 7]
default motionSensors is devices "notification sensor", with: "home security"
```

```
default currentCycleMode is false
default cycleTargetSetpoint is tooColdLevel
```

Setting variables

A variable can be set in the execution of a job. It could be a variable that was given a default value or a new variable.

A variable can be set to:

- a string, number, a boolean or a list thereof (`set highTemperatureLevel to 75`)
- the value of another variable (`set cycleTargetSetpoint to tooColdLevel`)
- a list of devices that match a device class,
 - with the optional exception of some devices with given ids (`set downstairsTstats to devices thermostat, except: [5, 7]`)
 - including only those that have certain tags (`upstairsTstats is devices thermostat, only: ["upstairs"]`)
 - restricted to devices that have a given attribute (`set motionSensors to devices "notification sensor", with: "home security"`)
- a filtered list of devices held in another variable (`set onLights to lights with switched == "on"`)
- the values of a named attribute for a list of devices (`set temps to temperature of upstairsTstats`)
- the value of a payload item from a message that triggered a job - the message is identified by its topic (`set theseDoors to doorsToOpen from "open"`)

A device class can be named or be the value of a variable. Similarly, an attribute can be named directly or via a variable. Attribute and device class names are strings. Variable names are atoms.

Strings for device class and attribute names are automatically converted to Piston names for device class and attributes (downcased atoms with spaces replaced by an underscore character).

For example, this script will turn all power switch binary devices upon being activated.

```
script "testing" do
  # will be converted to :power_switch_binary
  default switch is "Power switch binary"
  default deviceClass is switch
  default lights is devices deviceClass
  # will be converted to :switched
  default attributeName is "switched"
  default onOrOff is "off"

  starting now
```

```

run "test", now do
  set onLights to lights with switched = "on"
  set attributeName of onLights to onOrOff
end
end

```

Lists of devices of a given class can be filtered using `only:`, `except:` and/or `with:`. The `only:` and `except:` filters are each followed by a device id, a device tag, or a list of device ids and/or tags. A `with:` filter is followed by an attribute name (a string) or by a variable name that holds an attribute name.

If filtered by `except`, the list of devices is expunged of those devices that have *any* of the given tags or ids.

If filtered by `only`, the devices retained are those that have *all* of the tags mentioned (if any is listed), and have an id from the list (if any is listed).

If filtered by `with`, the devices retained are those that have the named attribute.

For example,

```

script "Turn lights on and off with motion" do

  default motionSensors is devices "notification sensor", with: "home security"
  default lights is devices "power switch binary", only: ["light", "motion activated"]
  default dimmers is devices "power switch multilevel", except: [10, 12]

  ...

```

Remembering the history of a variable

A script can be instructed to remember the values a variable has been set to over time.

These values are accumulated into a list held by another variable dubbed a “history variable”. History variables can be used like other variables (for e.g. they can be averaged) except they can *not* be set directly. Values in a “history variable” are retained for a set period of time. Once a remembered value is too old, it is forgotten.

Multiple variables can be remembered by a common history variable.

A variable can not be remembered by more than one history variable.

Note that only “simple” values can be retained, namely numbers, strings and booleans.

```

# Keep in special variable `lastDayTemps` the list of values `tempNow`
# was set to in the last 24 hours
remember tempNow as lastDayTemps, hours: 24

```

```
# Humidity readings are kept for a week
remember humidityReading as lastWeekHumidityReadings, days: 7
```

A time period *must* be given when declaring a variable as “remembered”. The supported time units are days, hours, and minutes.

Histories are preserved across activations of a script and across reboots of the Hub.

However, a script’s variable histories are forgotten if the script is replaced by another of the same name, or when the script is removed.

Script start time

The start time of a script is either given as now or as a cron pattern, such as `* 12 * * *` (noon-ish).

Note that cron patterns in a script resolve to minutes, not seconds. The timezone set via `NervesTimeZones.set_time_zone/1` is used to interpret the cron patterns to a local timezone. If the timezone is not set, UTC is the default local timezone.

The following ensures that the script starts at noon (more precisely ASAP between noon and 1PM). Only then will its jobs be primed for execution.

```
starting "* 12 * * *"
```

This means that if the script is activated, say, at 8PM, nothing will happen until noon the next day.

To start a script upon its activation (or upon Hub startup if the script was active when the Hub last shut down), use now:

```
starting now
```

A cron pattern can be parameterized by introducing variables. For example,

```
starting "* startTime * * *"
```

This works only if `startTime` is a variable that contains a number acceptable in a cron pattern.

Defining and scheduling a job

A script affects the Hub and its paired devices by executing its jobs.

Each job has a name that is unique within the scope of the script. Each job has a trigger determining when it is to be run once the script has started.

The trigger can be

- a schedule, which is either now or a cron pattern preceded by every,
- a variable changing value, expressed as follows whenever <variable name> changes,
- a scene activation, expressed as on scene: <scene number>, key: <key action>, times: <number of clicks> or on scene: <scene number>, key: <key action>,
- a received pubsub message, expressed as whenever receiving "<topic name>", or whenever receiving <payload item name> from "<topic name>"

```
# Run immediately once the script starts

run "Cool it to low", now do

...

end
```

```
# Every day at 2:55PM, if the script has started

run "Over-cool just before 3PM", every "55 14 * * *" do

...

end
```

```
# Every minute from 3PM to 7PM, if the script has started

run "Dampening cooling cycles", every "* 15-19 * * *" do

...

end
```

```
# Run whenever variable tempNow changes

run "Adjust heating setpoint", whenever tempNow changes do

...

end
```

```
# Run when the key for scene 4 is pressed twice on a paired scene
# controller
```

```
run "Turn lights off", on scene: 4, key: pressed, times: 2 do
...
end
```

```
# Run whenever the script receives a message with topic "door knocks"
# with a payload that contains a `count` value
run "answer door", whenever receiving count from "door knocks" do
...
end
```

```
# Run whenever the script receives a message with topic "door knocks"
# no matter what's in the message payload
run "ask who", whenever receiving "door knocks" do
...
end
```

A job execution can be triggered when the value of a variable changes. If the variable contains simple value, setting it to a different (simple) value will trigger the job. If the variable contains devices, a change in any device attribute, whatever the cause, will trigger the job.

If the job is triggered by a scene activation, the possible key actions are pressed, held or released. The times parameter, which defaults to 1 if left out, only applies for pressed key actions. It is otherwise ignored.

If the job is triggered by a message with a specified payload item, the value of that payload item is available to the job.

```
run "answer door", whenever receiving count from "door knocks" do
    set knocks to count from "door knocks"
    ...
end
```

A job defines a sequence of possibly conditional steps where each step consists of

- setting a script variable
- setting the attribute of a list of devices
- publishing information (sending a pubsub message)
- pausing for some time

```
run "Cool it to low", now do

    set thermostat_mode of upstairsTstats to "cool"

    set thermostat_cooling_setpoint of upstairsTstats to lowTemperatureLevel

end
```

The values of variables can be modified in arithmetic expressions.

```
run "Over-cool just before 3PM", every "55 14 * * *" do

    set cycleTargetPoint to lowTemperatureLevel + 2

    set currentCycleMode to true

    set thermostat_cooling_setpoint of upstairsTstats to tooColdLevel

end
```

A series of steps can be made conditional.

```
run "Dampening cooling cycles", every "* 15-19 * * *" do

    set upstairsTemp to average temperature of upstairsTstats

    if currentCycleMode == false and upstairsTemp >= cycleTargetSetpoint + 2 do

        set thermostat_cooling_setpoint of upstairsTstats to cycleTargetSetpoint

        set currentCycleMode to true

    end

    if currentCycleMode == true and upstairsTemp <= cycleTargetSetpoint do

        set cycleTargetSetpoint to cycleTargetSetpoint + 2

    end

end
```

```

        set currentCycleMode to false

        set thermostat_cooling_setpoint of upstairsTstats to highTemperatureLevel
    end

    if cycleTargetSetpoint == highTemperatureLevel do

        set cycleTargetSetpoint to highTemperatureLevel - 2

    end

end

```

Note in the job definition above the step `set upstairsTemp` to average temperature of `upstairsTstats`. It gets the list of temperature attribute values of thermostats held in variable `upstairsTstats`, averages the values obtained, and sets a variable `upstairsTemp` to the average temperature.

A list of values can be converted into a single value via aggregators, average being one. The other aggregators are:

- sum or total (they are synonyms)
- count
- highest or maximum or largest (synonyms)
- lowest or minimum or smallest (synonyms)

All aggregators require lists of numerical values, except for average which also works on non-numerical values. The average of a list of non-numerical values is the most frequent value, or any one of the values if they are all equally frequent.

Scripts can affect the external world by setting attributes of devices (turning a power switch on, locking a door, raising the heating set point of a thermostat, etc.).

They can also do so by publishing messages that are listened to by external agents. For example, when a script publishes, say, that all doors are locked, the Hub will forward the message via MQTT for the benefit of subscribed MQTT clients.

```

if (doorlock of doors) == "secured" do

    publish "locked down", doors: doors

end

```

The execution of a job can be suspended temporarily with the pause step. The pause step must indicate how long to wait before taking the next step or before completing the job, should the pause be the last step of the job.

The allowed units of time are seconds, minutes and hours. The duration must be given as an integer constant (not as a variable containing an integer).

```
pause seconds: 3
```

At most one execution of a particular job is allowed at any time. If a job is triggered while it is running, whether it is currently paused or not, the trigger is ignored.

The lifecycle of a script

1. A new script is created, named and defined.
2. The script is verified for code errors.
3. The validated script is added to the Hub in an inactive state.
4. The script is activated. The activation can include variable settings that override the script's default settings.
5. Once activated, the script is scheduled to start, either now or at a time specified by a cron pattern.
6. The script starts, and primes its jobs for execution.
7. The script's jobs, once primed, run on their respective schedules or when otherwise triggered. They run repeatedly until the script is stopped or deactivated. That's unless a job is scheduled now. Then it runs only once, when the script is started.
8. The script is deactivated and de-schedules its jobs. Any job currently running completes and is not repeated. If the script has not started yet, then it won't start unless the script is activated again.

If the Hub reboots, all scripts that were active at shutdown are re-activated and restart on their respective schedules. Scripts that were deactivated, remain so and can be activated on demand.

A script can be removed from the Hub, or replaced by another of the same name. The script removed or replaced is stopped.

Errors and failures

Scripts are compiled when they are added or when they are validated (without being added). A script is successfully added and made available for activation only if the script compiler finds it valid.

A script is invalid if

- it is syntactically incorrect,

- its execution would invoke a function not whitelisted by the compiler.

The compiler's list of allowed functions is restricted largely to the operations illustrated in the examples above.

A device attribute value is obtained via a “get” command, if one is defined for it. If not, the value is looked up in the descriptor of the device where it might have been set by a notification sent by the device (typically, a notification sensor). A device attribute value is always set by a script job by executing the “set” command associated with the attribute.

Script jobs make a best effort at getting and setting device attributes. If a job getting or setting a device's attribute fails (usually due to repeated command timeouts), the failure is reported but it does not terminate the job.

About syntax

The script syntax is relatively simple and straightforward but it does however suffer from one important limitation: Code written in it must be compliant with the Elixir syntax. This means that the following code for a script job is incorrect even though it seems it should be fine:

```
run "open all doors", whenever receiving doorsToOpen from "open" do

    set doorlock of (doorsToOpen from "open") to "unsecured"

end
```

It can be re-written thus so as not to break Elixir syntax:

```
run "open all doors", whenever receiving doorsToOpen from "open" do

    set theseDoors to doorsToOpen from "open"

    set doorlock of theseDoors to "unsecured"

end
```

When in doubt, keep each line of code short.

Internally, device class names, attribute names and some attribute values are atoms written in “snake case”, for example `:power_switch_binary`. In scripts, these can usually be expressed as more friendly strings that replace the underscores with spaces, for example, “power switch binary”.

Sometimes, doing so, will break Elixir syntax and one is then forced to use the internal, snake-cased atomic names.

For example, `set sensed to "home security" of motionSensors` is rejected as invalid syntax by the script compiler and needs to be written as `set sensed to home_security of motionSensors`.

Sample scripts

```
script "Turn the lights on and off" do

  default lights is devices power_switch_binary

  default onHour is 10

  starting now

  # Toggle the lights every minute of the hour held in the `onHour` variable

  run "very slow blink", every "* onHour * * *" do

    # Set variable onOrOff to the current switched status of most lights

    set onOrOff to average switched of lights

    # Toggle

    if onOrOff == "on" do

      set switched of lights to "off"

    else

      set switched of lights to "on"

    end

  end

end

end
```

```
script "Lock and unlock doors via switches" do

  default lights is devices "Power switch binary"

  default doors is devices "door lock"

  starting now
```

```

run "lock unlock", whenever lightSwitches changes do

  if lightSwitches == "on" do

    set doorlock of doors to "unsecured"

  else

    set doorlock of doors to "secured"

  end

end

run "blink lights", every minute do

  if (switched of lights) == "on" do

    set switched of lights to "off"

  else

    set switched of lights to "on"

  end

  set lightSwitches to switched of lights

end

run "Publish when locked down", whenever doors changes do

  # if all known door_lock attribute values for the doors is "secured"...

  if (door_lock of doors) == "secured" do

    publish "locked down", doors: doors

  end

end

end

```

```

script "Local warming" do

  default tstats is devices thermostat

  # Remember all successive values of tempNow in the last 24 hours

  remember tempNow as lastDayTemps, minutes: 3

```

```

    starting now

    run "Sample the temperature", every minute do

        set tempNow to average temperature of tstats

    End

    run "Ramp up", whenever tempNow changes do

        if tempNow <= average lastDayTemps do

            set thermostat_heating_setpoint of tstats to (average lastDayTemps) + 1

        end

    end

end

script "Lights on and off" do

    default lights is devices "Power switch binary"

    starting now

    run "Lights on", on scene: 4, key: pressed do

        set switched of lights to "on"

    end

    run "Lights off", on scene: 4, key: pressed, times: 2 do

        set switched of lights to "off"

    end

end
end

```

```

script "Come on in" do

    starting now

    default doors is devices "door lock"

    run "open all doors", whenever receiving doorsToOpen from "open" do

        publish "greet", say: "come on in"

        set theseDoors to doorsToOpen from "open"

    end

end

```

```

    set doorlock of theseDoors to "unsecured"

end

run "answer door", whenever receiving count from "door knocks" do

    set knocks to count from "door knocks"

    if knocks > 2 do

        publish "open", doorsToOpen: doors

    end

end

end
end

```

```

script "Door lock and light coordination" do

    default doors is devices door_lock

    default lights is devices power_switch_binary

    starting now

    run "Welcome home", whenever doors changes do

        if (doorlock of doors) == "unsecured" do

            set switched of lights to "on"

        end

    end

end

run "Good bye", whenever doors changes do

    if (doorlock of doors) == "secured" do

        set switched of lights to "off"

    end

end

end
end

```

```

script "Turn lights on and off with motion" do

```

```

default motionSensors is devices "notification sensor", with: "home security"

default lights is devices "power switch binary", only: ["light", "motion activated"]

default dimmers is devices "power switch multilevel", only: ["light", "motion
activated"]

# Remember the motion detections for the last 5 minutes

remember motion as motionDetections, minutes: 5

starting now

run "Register motion and turn lights on", whenever motionSensors changes do

    set sensed to home_security of motionSensors

    if "motion detection" in sensed do

        set motion to true

        set switched of lights to "on"

        set level of dimmers to 40

    end

end

run "Maybe turn off lights", every minute do

    if (count motionDetections) == 0 do

        set switched of lights to "off"

        set level of dimmers to 0

    end

end

end
end

```

```

script "Energy saver" do

    starting "* 12 * * *"

    default tooColdLevel is 65

    default lowTemperatureLevel is 70

    default highTemperatureLevel is 75

```

```

default upstairsTstats is devices thermostat, except: 34

default currentCycleMode is false

default cycleTargetSetpoint is tooColdLevel

run "Cool it to low", now do

    set thermostat_mode of upstairsTstats to "cool"

    set thermostat_cooling_setpoint of upstairsTstats to lowTemperatureLevel
end

run "Over-cool just before 3PM", every "55 14 * * *" do

    set cycleTargetPoint to lowTemperatureLevel + 2

    set currentCycleMode to true

    set thermostat_cooling_setpoint of upstairsTstats to tooColdLevel
end

run "Dampening cooling cycles", every "* 15-19 * * *" do

    set upstairsTemp to average temperature of upstairsTstats

    if currentCycleMode == false and upstairsTemp >= cycleTargetSetpoint + 2 do

        set thermostat_cooling_setpoint of upstairsTstats to cycleTargetSetpoint

        set currentCycleMode to true

    end

    if currentCycleMode == true and upstairsTemp <= cycleTargetSetpoint do

        set cycleTargetSetpoint to cycleTargetSetpoint + 2

        set currentCycleMode to false

        set thermostat_cooling_setpoint of upstairsTstats to highTemperatureLevel
    end

    if cycleTargetSetpoint == highTemperatureLevel do

        set cycleTargetSetpoint to highTemperatureLevel - 2

    end

end

end

```

end

Managing scripts

You can add, modify, activate, deactivate and remove scripts.

The screenshot displays the ALLOY SmartHome system dashboard. At the top, there is a navigation bar with tabs for Devices, SmartStart, Scenes, Scripts (which is currently selected), Networks, Tests, and Help. On the left side, there are two status cards: 'Energy saver is inactive' with an 'activate' link, and 'Turn lights on or off with door is active' with a 'deactivate' link. Below these cards are 'New' and 'Remove' buttons. A dropdown menu shows 'The time zone is' set to 'America/New_York'. The main content area is divided into three tabs: 'Code', 'Activation', and 'Log'. The 'Code' tab is active, showing a script titled 'Turn lights on or off with door'. The script code is as follows:

```
script "Turn lights on or off with door" do

  default doorSensors is devices "notification sensor", with: "access control"
  default lights is devices "switch multilevel", only: ["light"]
  # default lights is devices [7]

  starting now

  run "Turn lights on or off", whenever access_control of doorSensors changes do
    set sensed to access_control of doorSensors
    if "window_or_door_is_open" in sensed do
      set level of lights to 40
    end
    if "window_or_door_is_closed" in sensed do
```

To create a new script, press the **New** button. The newly created script is automatically selected and you can right away edit its sample code.



If you press **Remove**, the selected script will be removed from the Alloy Fusion Hub. First, if it is active, the script is deactivated before removal.

If the selected script was modified, pressing the *Revert* button abandons unsaved code and activation settings.

You can change the timezone in which the script jobs are scheduled. This will not affect already activated scripts.

You can control a script, transitioning it

- from modified to inactive (by saving it to the Hub),
- from inactive to active (by activating it),
- from active to inactive (by deactivating it)

When a script is selected, you can access its code, its activation settings and a log of events from its (eventual) activation.

When you make changes to the code of a script, you cannot save those changes unless the code has been checked successfully.



Pressing the **Check changes** button, tells you if an error was detected. You will only see the first error detected.



If an error is detected, you'll need to **Go back** to editing to correct it. You'll need to *Check changes* again. Hopefully no more errors are detected.

ALLOY SmartHome System dashboard

Devices SmartStart Scenes **Scripts** Networks Tests Help

Not named yet is new [save](#)

Energy saver is inactive [activate](#)

Turn lights on or off with door is active [deactivate](#)

[Remove](#)

The time zone is
America/New_York

Code	Activation	Log
<pre> 1 script "Turn lights on and off with motion" do 2 3 default motionSensors is devices "notification sensor", with: "home security" 4 default lights is devices "power switch binary", only: ["light", "motion activated"] 5 default dimmers is devices "power switch multilevel", only: ["light", "motion activated"] 6 7 # Remember the motion detections for the last 5 minutes 8 remember motion as motionDetections, minutes: 5 9 10 starting now 11 12 run "Register motion and turn lights on", whenever motionSensors changes do 13 set sensed to home_security of motionSensors 14 if "motion detection" in sensed.status do 15 set motion to true 16 set switched of lights to "on" 17 set level of dimmers to 40 </pre>		

[Go back](#) No error found

If so, you can **Accept** the changed code so that it is updated. This will put the script into *modified* status. Note that the changes are not sent yet to the Hub.

ALLOY SmartHome System dashboard

Devices SmartStart Scenes **Scripts** Networks Tests Help

Not named yet is new [save](#)

Energy saver is inactive [activate](#)

Turn lights on or off with door is active [deactivate](#)

[Remove](#)

The time zone is
America/New_York

Code	Activation	Log
<pre> script "Turn lights on and off with motion" do default motionSensors is devices "notification sensor", with: "home security" default lights is devices "power switch binary", only: ["light", "motion activated"] default dimmers is devices "power switch multilevel", only: ["light", "motion activated"] # Remember the motion detections for the last 5 minutes remember motion as motionDetections, minutes: 5 starting now run "Register motion and turn lights on", whenever motionSensors changes do set sensed to home_security of motionSensors if "motion detection" in sensed.status do </pre>		

[Accept](#) [Revert](#)

To push the code changes to the Hub, you need to press the **save** link next to the name of the script in the list of scripts.

Once a new or modified script is saved, it is made inactive. You could activate it now, or you may choose to add activation settings. Activation settings typically override the default settings of a script defined in its code.

When editing the activation settings of a script, you can add new settings or change the values of existing settings.

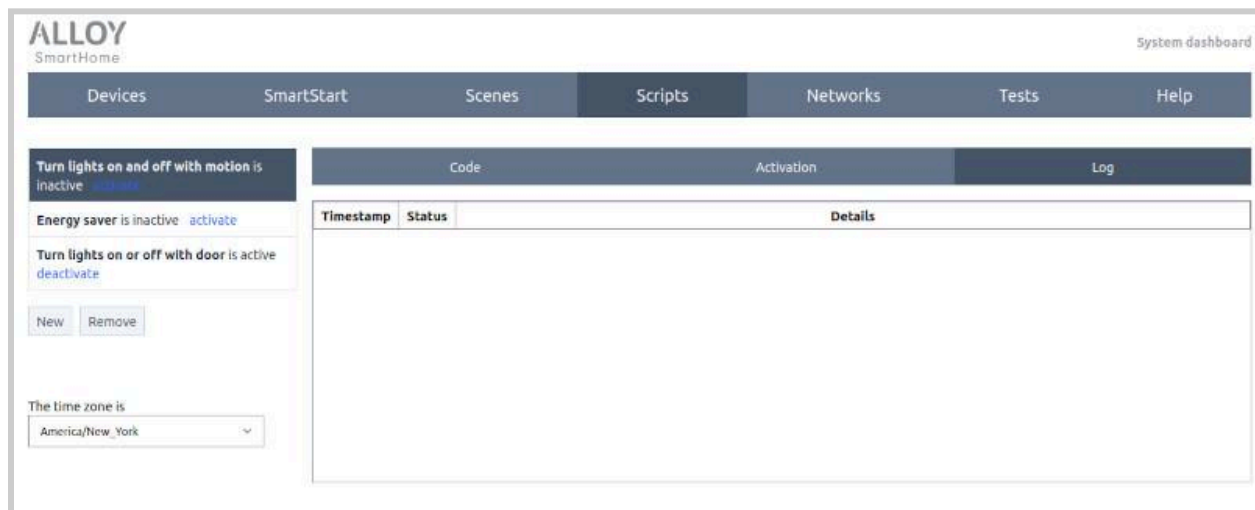
The screenshot shows the ALLOY SmartHome interface. The top navigation bar includes 'Devices', 'SmartStart', 'Scenes', 'Scripts', 'Networks', 'Tests', and 'Help'. The 'Scripts' tab is selected. On the left sidebar, a list of scripts is shown: 'Turn lights on and off with motion' (inactive), 'Energy saver' (inactive), and 'Turn lights on or off with door' (active). The main content area has three tabs: 'Code', 'Activation', and 'Log'. The 'Activation' tab is active, displaying 'Upon script activation...'. It shows a list of activation settings: 'lights' contains all 'power binary switch' devices except for '3, 5'. Below this, there is an 'Add variable' section with a text input 'var_name', a dropdown 'set to' with 'some number' selected, and a 'do it' button. At the bottom of the activation settings are 'Accept' and 'Revert' buttons. On the left sidebar, there are 'New' and 'Remove' buttons, and a time zone selector set to 'America/New_York'.

Each activation setting holds a type of value, which is set when it is created. You can **Accept** your changes to the script's activation settings, or you can **Revert** to what they were.

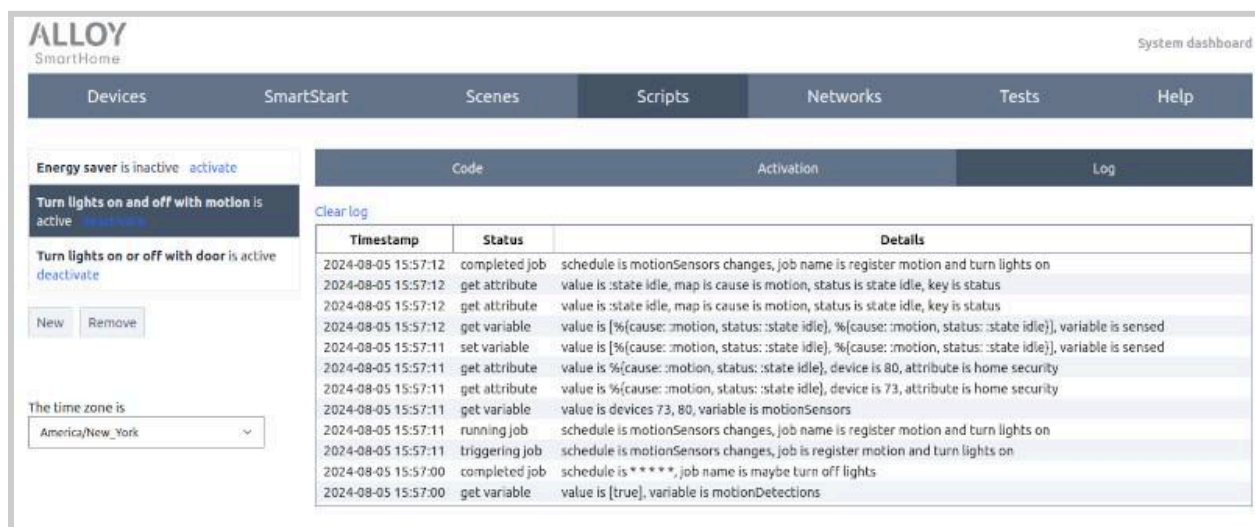
If you accept them, they are updated and the new activation setting will be used when you next activate the script. But first you'll need to **save** the changes to the Hub before you can activate the modified script.

This screenshot shows the same ALLOY SmartHome interface as the previous one, but with updates. The script 'Turn lights on and off with motion' is now marked as 'modified'. In the activation settings, a new entry 'onHours' has been added, containing the value '14'. The 'lights' entry remains. The 'Add variable' section is still present. The 'Accept' and 'Revert' buttons are at the bottom. A green notification box on the left sidebar states 'The script's activation settings were updated'. The time zone selector remains set to 'America/New_York'.

When you transition a script from inactive to active by pressing **activate** (next to the name of the script in the list of scripts), the script will start executing and producing a log which you can watch as it grows.



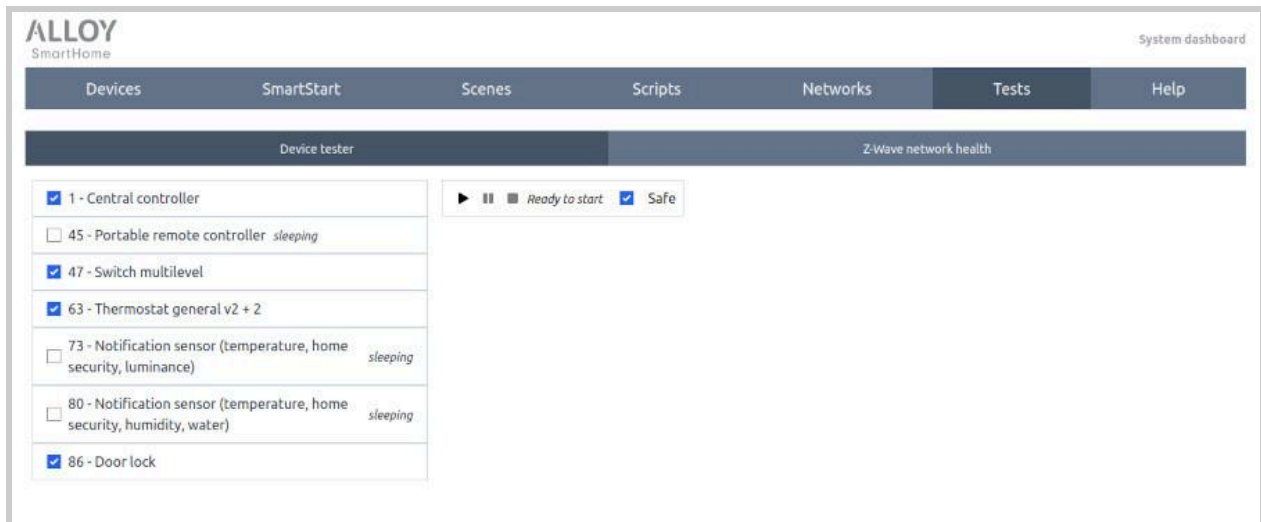
Note that the log grows from the top and that it contains only the latest 100 entries. You can clear the log at any time.



10- Testing devices

The Alloy Fusion Hub allows you to easily and systematically test the capabilities of devices paired with it, and see the results.

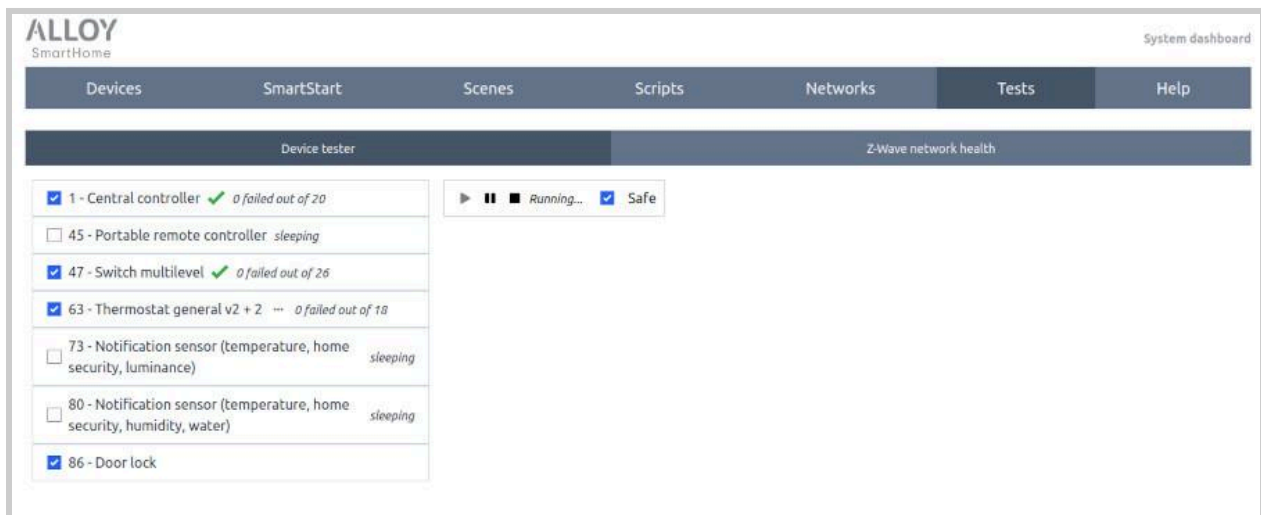
A capability is a device attribute, for example the heating setpoint of a thermostat, that can be got, set or both. Only the attributes of "awake" devices can be tested.



Before running tests, you can restrict which devices are to be tested. By default, all awake devices are selected for testing. You can deselect any number of them. You will not be able to alter your selection of devices while the tests are running, even if they are paused.

To run the tests, press the play button. To pause testing, press the pause button. Pressing play will resume testing. You can cancel testing before it completes by pressing the stop button.

Tests are run in "safe" mode by default. When in safe mode, a *best effort* is made to skip over attribute-setting tests that might change the current attribute values. Be aware that, even when testing in safe mode, some tests might change the current state of a device.



While the tests are running, you will be able to track which device is currently under test (pulsing gray dots) and which have been tested (green check mark or red cross).

Once the tests have completed, an overall score will appear, together with the number of tests run, and the time taken to run them all.

ALLOY SmartHome System dashboard

Devices | SmartStart | Scenes | Scripts | Networks | **Tests** | Help

Device tester | Z-Wave network health

☒ 1 - Central controller 0 failed out of 20
☐ 45 - Portable remote controller *sleeping*
☒ **47 - Switch multilevel** 0 failed out of 26
☒ 63 - Thermostat general v2 + 2 0 failed out of 42
☐ 73 - Notification sensor (temperature, home security, luminance) *sleeping*
☐ 80 - Notification sensor (temperature, home security, humidity, water) *sleeping*
☒ 86 - Door lock 0 failed out of 50

▶ || ■ Completed! ☒ Safe 100% success for 138 tests executed in 65 seconds

Test results for 47 - Switch multilevel

all association group info	get
association group command list	get
association group info	get
association group name	get
associations	get
battery percent	get
command class version	get
firmware ids	get
firmware update info	get
firmware upgradable	get
indicator state	get
level	get set
manufacturer info	get
multi channel associations	get
nothing	get
power level	get set
power level test	get
serial number	get
supported association groupings	get
supported multi channel association groupings	get
zwave software info	get
zwave version	get
zwaveplus info	get

Status: **success**
 Device id: **47**
 Attribute: **level**
 Arguments: **40**
 Answer: **ok**
 Run time: **141 msec**

You can, at any time, even while the tests are running, select a device to see the results of the tests that were run on it. The attributes tested will appear as a list. Click on "get" or on "set" (if shown) next to the attribute to see a test result.

11- Command classes

The Alloy Fusion Hub is itself a Z-Wave® device and as such implements a number of Z-Wave® command classes, in addition to controlling command classes of paired devices.

A command class defines a collection of Z-Wave®-accessible capabilities expressed by the Hub as attributes of the device that implements it.

For example, any device that implements the Time command class can report the current values of its "Date", "Time" and "Time offset" attributes.

Supported command classes

The Alloy Fusion Hub is a Z-Wave® Static Controller device and as such implements the following command classes:

Command class	Version	Secure	Exposed as capabilities
application status	1	no	no
association	3	yes	no
association group info	3	yes	no
crc 16 encap	1	no	no
device reset locally	1	yes	no
firmware update md	5	yes	no
inclusion controller	1	no	no
indicator	3	yes	yes
manufacturer specific	2	yes	yes
multi channel association	4	yes	no
multi command	1	yes	no
powerlevel	1	yes	yes
security	1	no	no
security 2	1	no	no
supervision	1	yes	no
time	1	no	yes
transport service	2	no	no
version	3	yes	yes
Z-Wave®plus info	2	no	yes

Controlling command classes

Association / Multichannel Association

The hub automates associating paired devices over the Lifeline group. Associating devices over other groups is not supported.

See section *Associations* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

You can only *access information* about a device's associations to the Lifeline (group 1). This can be done by accessing the Capabilities tab of a device and getting the values of attributes "Association group name" (the name of an association group), "Association Group Info" (information about one association group), "All association group info" (information about all association groups), and "Association Group Command List" (the commands sent via a given association group).

See section *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Basic

The Basic command class is exposed via the Capabilities tab of a device as the "On off state" attribute. The attribute value can be obtained or set (turned "on" or "off").

The "On off state" attribute is exposed only for devices that do **not** control other actuator command classes (command classes used to change the user-visible state of a device).

Changes to the value of this attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device's Descriptor.

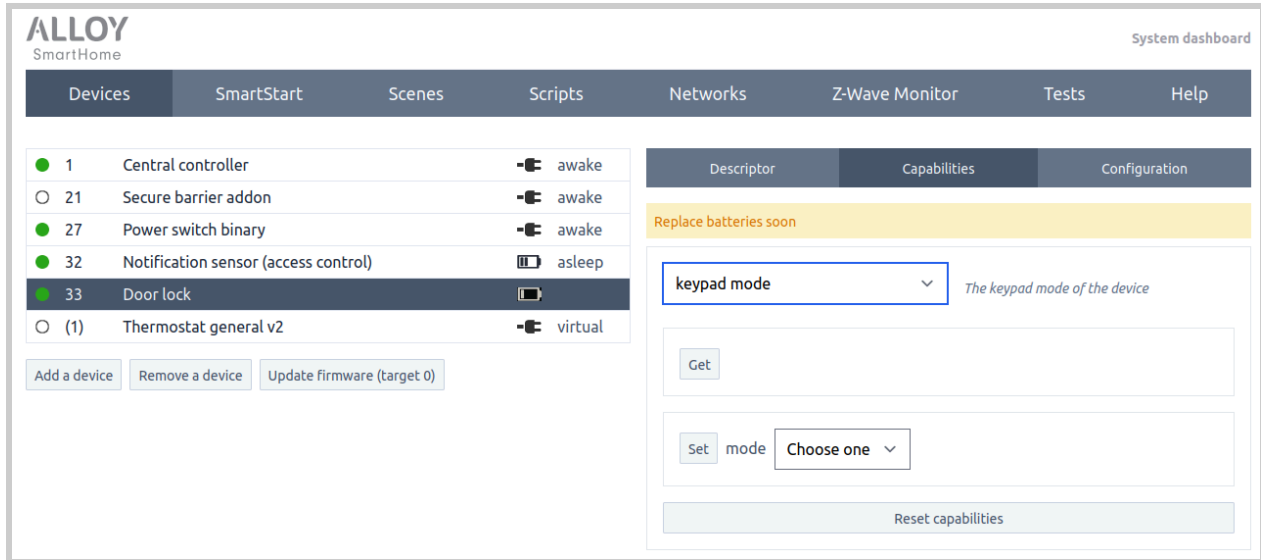
See section *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Battery

The Battery command class is controlled in a number of ways.

The percent charge of a device's battery (if it supports the Battery command class) appears in the Devices tab which lists the paired devices.

If a device's battery level is too low, a warning message will appear above its Descriptor (in the Descriptor tab for the device).



The “Battery percent” attribute value is shown under the device’s Descriptor tab, if the Battery command class is supported by the device.

Finally, the battery level of a device can be obtained as the value of its “Battery percent” attribute via the Capabilities tab of the device.

See section *Controlling devices* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Binary Switch

The Binary Switch command class is exposed as the “Switched” attribute of a device supporting it.

The “Switched” attribute value is shown under the device’s Descriptor tab, if the Switched Switch command class is supported by the device.

The value of the “Switched” attribute can be obtained via the Capabilities tab. The value and duration(if supported) can be set also via the Capabilities tab. The value is either “on” or “off”. The duration is either “default” or is set in seconds. For durations of greater than 127 seconds, the effective duration steps up by one minute.

Changes to the value of the “level” attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device’s Descriptor.

See section *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Central Scene

The Central Scene command class is primarily controlled by the Hub via the Scenes tab.

See section Chapter 8. [Defining scenes](#) for details.

Scene activations are also temporarily displayed on the Devices tab whether or not a Scene was defined to respond to the activation.

The screenshot shows the ALLOY SmartHome System dashboard. The top navigation bar includes tabs for Devices, SmartStart, Scenes, Scripts, Networks, Tests, and Help. The main content area is divided into two sections. On the left, under the 'Devices' tab, there is a list of devices with their status (awake or asleep). On the right, there is a detailed view of a device's attributes and tags.

Descriptor	Capabilities
DSK	none
Firmware ids	0 and 1
Firmware version	1.35.0
Manufacturer specific	manufacturer id is 912, product id is 1, product type id is 1
Network key	unknown
Security	secure
Z-Wave module serial number	h'698000232

Attributes	Value
Power source	external
Version	firmware version is 7.19, protocol version is 7.19, hardware version is 1, library type is bridge controller, other firmware versions are 7.18 and 1.0

Tags	Value
Tags	upstairs, bedroom

Configuration

The Configuration command class is controlled via the Configuration tab. It is shown for the selected device if the device supports the Configuration command class **and** it is not a wake-up device.

If the device supports the Configuration command class *and* is a wake-up device, configuration parameters are set automatically by the Hub at the time of pairing and can not be modified via Hubview.

See section *Configuration* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Device Reset Locally

When a device is reset to factory settings, it sends a notification informing the Hub. The Hub reacts by unpairing the device and removing it from the Z-Wave® network.

Hubview shows a temporary message alerting the user and updates itself to reflect that the device is no longer on the Z-Wave® network.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

7 Switch multilevel

9 Thermostat general v2

16 Portable remote controller

19 Notification sensor (home security, temperature, luminance)

32 Door lock

awake

awake

asleep

asleep

awake

Add a device

Remove a device

Remove failed nodes

Heal network

Z-Wave reset

Download network keys

Device 53 was reset to factory defaults -- The notification sensor device with id 53 was removed -- Exclusion done -- Device no longer in network

Descriptor

Capabilities

About

Attributes

Tags

DSK

Firmware ids

Firmware version

Manufacturer specific

Network key

Security

Z-Wave module serial number

Power source

Version

upstairs, bedroom

set

none

0 and 1

1.33.0-dev

manufacturer id is 912, product id is 1, product type id is 1

unknown

secure

h'698000232

external

firmware version is 7.19, protocol version is 7.19, hardware version is 1, library type is bridge controller, other firmware versions are 7.18 and 1.0

Doorlock

The Doorlock command class is controlled by the hub via a number of device attributes. The values of attributes may appear in the Descriptor tab for the device. Whether they do or not, they can be individually obtained and set via the Capabilities tab of the device.

The attributes are:

- “Doorlock”: The locked/unlocked state of the door lock (shown in Descriptor)
- “Door lock configuration”: The operational configuration of the door lock with lock timeout in seconds (not shown in Descriptor)
- “Door lock capabilities”: The capabilities of the door lock (not shown in Descriptor)

Changes to the value of the “doorlock” attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device’s Descriptor.

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Indicator

The Indicator command class is controlled by the hub in two ways: A device indicator can be activated from within the list of devices in the Devices tab, or it can be controlled as attributes via the Capabilities tab.

The possible attributes are (not all may be supported by the device):

- “Indicator state”: To get the state of each resource (its property id and value) of the device indicator given its id (e.g. “node identify”)

- “Indicator properties”: To get the names and values of all the properties of a device indicator
- “Indicator property”: To set the value of a property of a device indicator
- “Indicator description”: To get a detailed description of the appearance and use of the device indicator
- “Identify”: To request that the Hub identify itself in its default way

None of the Indicator attributes are shown in the device’s Descriptor tab.

See Chapter 6. [Activating indicators](#) and section *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Meter

The Meter command class is controlled by the Hub by showing the value of the device’s “Meter reading” attribute in its Descriptor tab.

The screenshot shows the ALLOY SmartHome interface. On the left, a list of devices is shown, with 'Power switch binary + 3' selected. The main panel displays the 'Descriptor' tab for this device. The 'About' section shows details like DSK, Firmware ids, and version. The 'Attributes' section lists various electric meter attributes, with 'Electric meter w' highlighted in yellow, showing a value of 0.0 W. The 'Firmware uploads' section has a dropdown set to '0' and a 'Browse...' button. The 'Tags' section shows 'upstairs, bedroom' and a 'set' button.

Descriptor	Capabilities	Configuration
Root device		
About DSK: 30058-01365-49695-39436-18902-15197-46484-33268 Firmware ids: 40963 Firmware version: 1.0.1 Manufacturer specific: manufacturer id is 634, product id is 40963, product type id is 40960 Network key: s2 unauthenticated Security: secure Z-Wave module serial number: hTFFFFFFFFFFFFFFF		
Attributes Electric meter a: 0.0 A Electric meter kwh: 1.89 kWh Electric meter v: 0.0 V Electric meter w: 0.0 W Power source: external Switched: off Version: firmware version is 2.0, protocol version is 5.3, hardware version is 2, library type is enhanced end node, other firmware versions are absent		
Firmware uploads Upload firmware file for target: 0 Browse... No file selected.		
Tags upstairs, bedroom set		

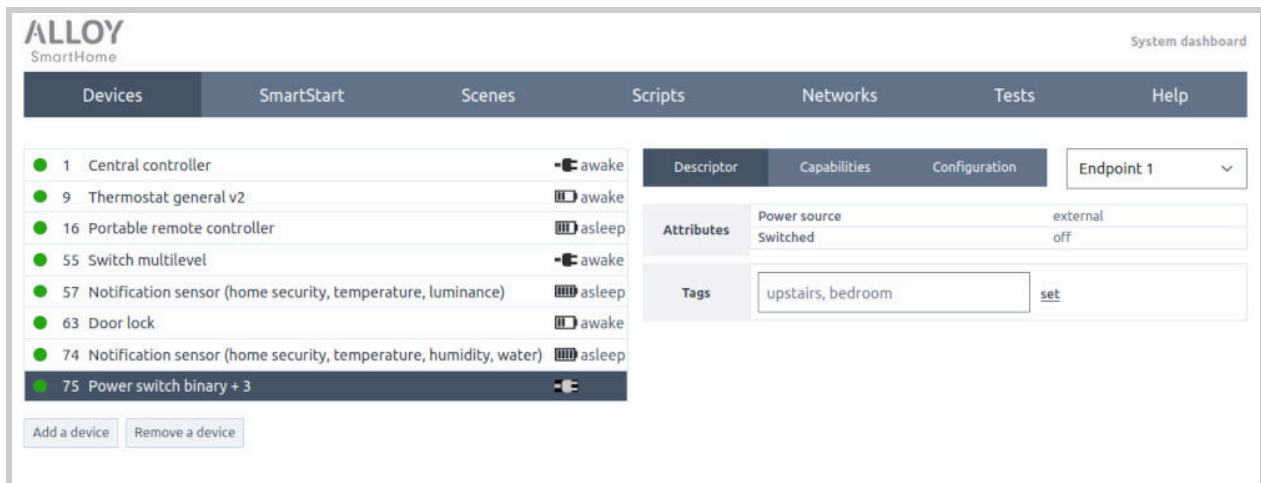
The Meter command class is also controlled via these attributes accessed via the Capabilities tab:

- “Meter reset”: To reset a meter given its type, a rate type, a scale and a value (not shown in Descriptor)
- “Meter supported”: To get the types of metering the device supports (not shown in Descriptor)
- “Electric meter w”, “Electric meter v”, etc.: To get the value of a supported meter in a supported scale (shown in Descriptor)

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Multichannel

The Hub controls the Multichannel command class by giving access to each endpoint of a device supporting this command class. An endpoint can be selected to see the endpoint's descriptor via its Descriptor tab, and to interact with the command classes the endpoint supports via its Capabilities tab.



See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Multilevel Sensor

The MultilevelSensor command class is controlled by the hub via a number of device attributes. The values of attributes may appear in the Descriptor tab for the device. Whether they do or not, they can be individually obtained via the Capabilities tab of the device.

The attributes, if supported by the device, are:

- “Supported multilevel sensors”: To get the types of multilevel sensors supported by the device (not shown in Descriptor)
- “Supported scales”: To get the kinds of units a type of multilevel sensor supports (not shown in Descriptor)
- “Sensed”: To get the value of the default type of multilevel sensing (shown in Descriptor)
- “Temperature”, “Humidity”, “Luminance”, etc.: To get the sensed value for the type of sensor (shown in Descriptor)

The device can emit reports about the sensed values of the sensor types it supports. The latest updated value is highlighted in the Descriptor.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

awake

9 Thermostat general v2

awake

16 Portable remote controller

asleep

55 Switch multilevel

awake

57 Notification sensor (home security, temperature, luminance)

asleep

63 Door lock

awake

74 Notification sensor (home security, temperature, humidity, water)

asleep

75 Power switch binary + 3

awake

Add a device

Remove a device

Descriptor

Capabilities

About

DSK

53974-60989-64758-30570-28027-62676-15229-08294

Firmware ids

absent

Firmware version

2.37.0

Manufacturer specific

manufacturer id is 305, product id is 4240, product type id is 3

Network key

s2 unauthenticated

Security

secure

Z-Wave module serial number

h'4E454F3131383132303031353634

Attributes

Battery percent

100

Home security

motion detection

Luminance

unit is lux, value is 1, sensor type is luminance

Power source

battery

Temperature

69.2 °F

Wake-up interval (milliseconds)

64800

Version

firmware version is 2.37, protocol version is 6.2, hardware version is 16, library type is routing end node, other firmware versions are absent

Tags

upstairs, bedroom

set

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Multilevel Switch

The Multilevel Switch command class is exposed as the “Level” attribute of a device supporting it.

The “Level” attribute value is shown under the device’s Descriptor tab, if the Multilevel Switch command class is supported by the device.

The value of the “Level” attribute can be obtained via the Capabilities tab. The value and duration(if supported) can be set also via the Capabilities tab. The Level value can be a number between 0 and 100, or it can be “previous” or “off”. The Level duration is either “default” or is set in seconds. For durations of greater than 127 seconds, the effective duration steps up by one minute.

See section *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Notification

The Notification command class is controlled by the hub via a number of device attributes. The values of attributes may appear in the Descriptor tab for the device. Whether they do or not,

they can be individually obtained and set via the Capabilities tab of the device. See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

The attributes are:

- “Alarm”: To get the status and current value (if available) for a given type of alarm of the device (not shown in Descriptor)
- “Alarm status”: To set the status of a type of alarm for the device (not shown in Descriptor)
- “Supported notification types”: To get the types of alarms that could be sent by the device (not shown in Descriptor)
- “Water alarm event”, “Access control event”, “Power management event”, etc.: To get the sensed value for the last notification for the given type of alarm (shown in Descriptor)

The Hub might associate an event with an attribute of the device such that the event updates the value of the attribute in question. These updated values are shown in the Descriptor with the last reported value highlighted.

For example, an “Access control” type of notification, sent on manually operating a door lock, indicates to the Hub that the state of the “Doorlock” attribute has changed.

The screenshot shows the ALLOY SmartHome System dashboard. On the left, a list of devices is displayed, with '63 Door lock' selected. The main area shows the details for this device, organized into three tabs: Descriptor, Capabilities, and Configuration. The Descriptor tab is active, showing various attributes and their current values. The 'Door lock' attribute is highlighted in yellow, indicating it is secured and the method is manual. Other attributes include Battery percent (45), Power source (battery), Schedule type supported, User code capabilities, and Version. The Firmware uploads section shows an upload button and a file selection area. The Tags section shows 'upstairs, bedroom' with a 'set' button.

Descriptor	Capabilities	Configuration
DSK	46411-40341-29023-40497-07528-48410-48075-30723	
Firmware ids	42400 and 32770	
Firmware version	2.20.1	
Manufacturer specific	manufacturer id is 297, product id is 17920, product type id is 32770	
Network key	s2 access control	
Security	secure	
Z-Wave module serial number	h'14B457FFEDF6AFC	
Battery percent	45	
Door lock	mode is secured, method is manual	
Power source	battery	
Schedule type supported	number of slots year day is 1, number of slots week day is 0, number of slots daily repeating is 7	
User code capabilities	supported keypad keys is 0123456789, supported keypad modes are normal, vacation, privacy, and lockout, supported user id statuses are available, occupied, disabled, messaging, and passage, multi user code set supported? is false, multi user code report supported? is false, user code checksum supported? is false, admin code deactivation supported? is false, admin code supported? is true	
Version	firmware version is 2.20, protocol version is 7.13, hardware version is 2, library type is enhanced end node, other firmware versions are 43.19	
Firmware uploads	Upload firmware file for target: 0 Browse... No file selected.	
Tags	upstairs, bedroom set	

If an alarm is of a type not known to the Hub, a message is displayed when it is received.

ALLOY

SmartHome

System dashboard

Devices

SmartStart

Scenes

Scripts

Networks

Tests

Help

1 Central controller

9 Thermostat general v2

16 Portable remote controller

55 Switch multilevel

77 Door lock

81 Notification sensor (luminance, temperature, home security)

82 Notification sensor (water, humidity, temperature, home security)

add

remove

remove failed nodes

heal network

Z-Wave reset

Download network keys

Unknown alarm of type 192 with event 222 was received from device 55

Descriptor

Capabilities

About

DSK

Firmware ids

Firmware version

Manufacturer specific

Network key

Security

Z-Wave module serial number

Power source

Version

Tags

upstairs, bedroom

set

Thermostat Fan Mode

The Thermostat Fan Mode command class is controlled by the hub via the “thermostat fan mode” attribute. Its value is shown in the Descriptor tab for the device, and can be set via the Capabilities tab of the device.

The sole attribute is:

- “Thermostat fan mode”: To get and set the fan mode (“auto”, “auto low” etc.) of a thermostat (shown in Descriptor)

Changes to the value of this attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device’s Descriptor.

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Thermostat Fan State

The Thermostat Fan State command class is controlled by the hub via the “thermostat fan state” attribute. Its value is shown in the Descriptor tab for the device, and can be set via the Capabilities tab of the device.

The sole attribute is:

- “Thermostat fan state”: To get the fan state (“on” or “off”) of a thermostat (shown in Descriptor)

Changes to the value of this attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device’s Descriptor.

68

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Thermostat Mode

The Thermostat Mode command class is controlled by the hub via a number of device attributes. The values of attributes may appear in the Descriptor tab for the device. Whether they do or not, they can be individually obtained and set via the Capabilities tab of the device.

The attributes are:

- “Thermostat supported modes”: To get the thermostat's supported modes (shown in Descriptor)
- “Thermostat mode”: To get the mode (“heat”, “cool”, “off”, “manufacturer_specific” etc.) of a thermostat, and set the thermostat to one of its supported modes **except** for “manufacturer specific” mode - this is to be done via the “thermostat manufacturer specific mode” attribute. (Shown in Descriptor)
- “Thermostat manufacturer specific mode”: If supported by the thermostat, to set the thermostat to “manufacturer-specific” mode with accompanying manufacturer-specific data.

Not all might be supported by the device.

Changes to the value of this attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device's Descriptor.

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Thermostat Operating State

The Thermostat Operating State command class is controlled by the hub via the “thermostat operating state” attribute. Its value is shown in the Descriptor tab for the device, and can be set via the Capabilities tab of the device.

The sole attribute is:

- “Thermostat operating state”: To get the operating state (“idle”, “heating”, “cooling” etc.) of a thermostat (shown in Descriptor)

Changes to the value of this attribute, when initiated on the devices and reported by it, are reflected and highlighted in the device's Descriptor.

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

Thermostat Setpoint

The Thermostat Setpoint command class is controlled by the hub via a number of device attributes. The values of attributes may appear in the Descriptor tab for the device. Whether they do or not, they can be individually obtained and set via the Capabilities tab of the device.

The attributes are:

- “Thermostat supported setpoints”: To get types of setpoints (“heating”, “cooling”, “energy save heating”, “energy save cooling”, etc.) supported by a thermostat (shown in Descriptor)
- “Thermostat setpoint capabilities”: To get the minimum and maximum unit values of a thermostat setpoint (not shown in Descriptor)
- “Thermostat heating setpoint”, “Thermostat cooling setpoint”, etc.: To get and set a type of setpoint on the thermostat (shown in Descriptor)

Not all might be supported by the device.

Changes to the values of attributes, where applicable and when initiated on the devices and reported by it, are reflected and highlighted in the device’s Descriptor if these attributes are displayed in it.

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

User Code

The User Code command class is controlled by the hub via a number of device attributes. The values of attributes may appear in the Descriptor tab for the device. Whether they do or not, they can be individually obtained and set via the Capabilities tab of the device.

The attributes are:

- “Users number”: To get the number of possible user codes supported by the device (not shown in Descriptor)
- “Permanent user code”: To get and set a numbered user code and its status (“occupied”, “available”, etc.) on the device (not shown in Descriptor)
- “User code capabilities”: To get the user code capabilities (whether an admin code is supported”, the supported keypad modes, etc.) of the device (shown in Descriptor)
- “Admin code”: To get and set the admin code on the device (not shown in Descriptor)
- “Keypad mode”: To get and set the keypad mode (“normal”, “privacy”, “lockout”, etc.) on the device (not shown in Descriptor)

Not all might be supported by the device.

See sections *Descriptor* and *Capabilities* in Chapter 4. [Managing the Z-Wave® network with Hubview](#) for details.

12- Supervision

The Supervision command class allows the Hub to request application-level delivery confirmation from a receiving Z-Wave® device to which a Set or Remove command was sent. The delivery annotates the device's response with a status of Working or Fail.

If a paired Z-Wave® device supports the Supervision command class, the Hub will send supervised commands as appropriate, except for the following exceptions:

Device	Command class	Command	Cause
Schlage BE469ZP Lock (CAM622) manufacturer_id: 59, product_type_id: 1, product_id: 1_129	Association	association_set	No Support error
Yale nexTouch Keypad Sectional Mortise Lock (NTM625 ZW3) manufacturer_id: 297, product_type_id: 33024, product_id: 18986 Kwikset Home Connect 620 locks manufacturer_id: 144, product_type_id: 2065, product_id: 936, 5032, 9128, 1707	Configuration	configuration_set	Always times out
Alloy Home Smart Plug3 (ah-nas-wr12z1u.us) manufacturer_id: 305, product_type_id: 4, product_id: 773	Switch Binary	switch_binary_set	Always times out