Neohub JSON commands

Rev 2.6

Neohub JSON commands description

This document is a complete API guide to using the Heatmiser Neohub system and associated devices. It is intended for Application developers who wish to communicate with the Neohub system directly. Connection to the NeoHub is made on port 4242 using a TCP connection.

Revision history. Rev 0	*Original version taken from V1 system*
Rev 1.000	*V2 commands added*
Rev 1.001	*Web sockets added*
Rev 2.000	*Cooling commands, new data caches*
REV 2.501	*Reformatted to include more Examples, descriptions and references*
REV2.502	*Added Home kit flag and HUB_TYPE variable*
REV2.503	*Added Addendum page 47*
REV2.504	*Minor correction to grouped commands added hub type variable to addendum on page 47*
REV2.505	*Updated hold command to include cooling*

Contents

JSON commands

1.1. Syntax description

<device(s)> is either a device, an array of devices and/or group names or a group name so "device1" or ["device1","device2","ourgroup"] or "ourgroup"

<group> is always a group name like "ourgroup"

anything inside <> ending with name is the name as a string "name"

<temp> can be an integer or a floating-point value

So the basic command structure is like this

when there are no arguments

{"cmd":0}

these will not check the values at all, so {"cmd":[1,2,3,4,{"fsdfdsf":3}]} would work as well

when there is one argument
{"cmd":<arg>}

when there are more

{"cmd":[<args>]}

where devices is always one argument, for one device or group it can be just the device zone name or device number or group name, if it's more they'll need to be an array (to stay the one devices argument)

so

{"cmd":[1,"kitchen"]} for one device in a multiple argument command

{"cmd":[1,["kitchen", "bathroom"]]} for multiple argument multiple devices

such commands will only check if there are enough arguments (and the correct contents)

if above the one argument is the devices it would be {"cmd":"kitchen"} or
{"cmd":["kitchen":"bathroom"]}, single device arrays are also allowed {"cmd":["kitchen”]}

V2 commands overview

Changes to the system

The operation of the V2 system is based on the idea that downloading or uploading data that has not changed since the last up/download is inefficient.

To this end the data has been broken down into sections. Each section includes a time stamp to indicate if the data has changed.

Constantly changing data is stored in the live data array which also contains the timestamps for all the other sections.

To display current information, the app needs only to download this single file. To check the validity of previously stored files it then needs to scan the timestamps and if the timestamps match those previously downloaded, nothing further needs to be done. If the timestamps have changed, then the associated file needs to be downloaded again.

The system must retain local (to the device) copies of the various settings and profiles held in the Neohub itself.

Synchronisation relies on the correct use and interpretation of the timestamps.

Switching levels

Thermostats can now have 4 or 6 comfort levels this is a global setting chosen by the user choice.

Profiles

The New system makes greater use of profiles

Profiles are numbered and named so that only the profile number needs to be sent between the system and the Neohub, the user uses the named local copy to select the profile and the app sends the profile number. The Neohub then uses its own copy of the profile to send the comfort level or timeclock settings to the device or devices.

The Neostat reports the profile it is currently using as part of the live data field

Profile 0

This profile is not really a profile, its purpose is to allow the system to report back changes on the thermostat itself and allow the user to use monitor or to change a single room.

If we assume that the kitchen is currently running a profile called “night shift” and the user manually changes the settings, then the kitchen Neostat is no longer running the night shift profile its using its own comfort levels. This change resets the profile byte in the live data to 0.

Profile 0 is therefore a copy of the comfort levels for a device that’s not running a profile.

It needs to be uploaded when indicated. Named Profiles 1 through 255 are available to the user.

Device Types

Device types tell you what the physical form of each device is, which indicates how to handle the information in the live data array and other caches. For example, the neostat is device type 1 but could be a thermostat or a timeclock the device type is located in the Engineers cache. NeoAirs are device type 7 and can be thermostats, timeclocks or a combination of both.

For a complete list of device types see Appendix E

Neohub commands

System Reboot

To force a soft reboot of the Neohub use **{"RESET":0}** Note this command only works with firmware version 2027 and above.

Set Channel

This command can be used to change the ZigBee channel used by the Neohub.

{"SET_CHANNEL":11} allowed channels are 11,14,15,19,20,24 and 25

Please note that if Neoairs or Neostats are connected to the Neohub, care must be taken to ensure all devices are online when the command is sent, otherwise the Neostat will not know the channel has been changed and may need to be re-paired to the network.

If you attempt to set a channel and the Neohub detects a lot of interference, it will refuse to change, therefore, you need to verify the channel change has taken place. You will find the channel number in the system cache.

Read DCB:100 Returns system wide variables.

This command has been deprecated use {"GET_SYSTEM":0}

System wide (Global) settings

Some feature settings within the Neostats are global, for example the choice of degrees Celsius or Fahrenheit, changing any of these settings will instruct the Neohub to change every device to the same settings.

{"SET_TEMP_FORMAT":"C"} or {"SET_TEMP_FORMAT":"F"} sets the system to run in Celsius (C) or Fahrenheit (F)

Note changing from C to F or vice versa will invalidate historical data so this will be automatically deleted.

{"SET_FORMAT":"7DAY"} note that changing the format on 1 device will change it on all connected devices.

Available formats are

1. "NONPROGRAMMABLE" this is a fixed temperature
2. "24HOURSFIXED" this is every day the same
3. "5DAY/2DAY" this is weekdays different from weekends
4. "7DAY" this is every day different.

Time clocks cannot be nonprogrammable.

If the system is moved to nonprogrammable any timeclock will remain on the previous settings

This setting can be read from "ALT_TIMER_FORMAT": variable in the system cache.

Note

Changing the program format will invalidate any stored profiles so these will be automatically deleted.

Global Functions

These functions affect every device on the system

Away Function Used for when the property is unoccupied for an unknown length of time.

The away function turns everything off putting timeclocks and thermostats into standby. Thermostats put in standby will maintain the frost protection temperature set by the user, ignoring any schedule. Timeclocks will remain off ignoring any schedule.

Commands are.

{'AWAY_ON':0} and **{'AWAY_OFF':0}** which affects all devices.

These commands can also be targeted to specific devices.

Example

{"AWAY_ON":["lounge","bed1"]} or {"AWAY_OFF":["lounge","bed1"]}

However, using the **{'AWAY_OFF':0}** command will cancel these, so it's better to use the FROST_ON /OFF commands for individual rooms.

Holiday Function

Used for when the property is unoccupied until a specified date and time.

The holiday function turns everything off putting timeclocks and thermostats into standby. Thermostats put in standby will maintain the frost protection temperature set by the user ignoring any schedule. Timeclocks will remain off ignoring any schedule.

{"HOLIDAY":["START TIME AND DATE ","END TIME AND DATE"]}

{"HOLIDAY":["HHMMSSDDMMYYYY","HHMMSSDDMMYYYY"]};

Example

{"HOLIDAY":["14450001062016","14450002062016"]}

In this example the start time is 2:45pm 0 seconds 01/06/2016 and the end time is 2:45pm 0 seconds 02/06/2016 14:45:00 01 06 2016, 14:45:00 02 06 2016"]}

{"GET_HOLIDAY":0}
Returns the holiday date/times from the Neohub.

Example

{"end": "Sun Feb 11 19:02:00 2018\n","ids": ["Office"],"start": "Fri Feb 9 14:33:29 2018\n"}

{"CANCEL_HOLIDAY":0}

Cancels the holiday all devices revert to following their schedules.

Caches and data stores

There are 7 blocks of data that rarely change these are stored in the Neohub and where they are needed in the controlling device usually an App

These blocks are

System.

Device lists

Engineers data

Profile_0

Profile_comfort levels

Profile_timers

Profile_timers_0

The process is that the controller sends a command, if that command affects any of the stored data the cache is updated along with its timestamp.

The timestamp is then copied to the live data field.

The controller scans the timestamps and uploads the cache if the timestamp is newer than the one it already holds. In this way data is only uploaded when it changes.

Live data

The live data array contains up to the minute status information about the thermostats and the timestamps for all the caches.

Example

{“GET_LIVE_DATA”:0}

Returns

{
"CLOSE_DELAY": 0,
"COOL_INPUT": false,
"GLOBAL_SYSTEM_TYPE": "HeatOnly",
"HOLIDAY_END": 0,
"HUB_AWAY": false,
"HUB_HOLIDAY": false,
"HUB_TIME": 1518613752,
"OPEN_DELAY": 0,
"TIMESTAMP_DEVICE_LISTS": 1518607836,
"TIMESTAMP_ENGINEERS": 1518607837,
"TIMESTAMP_PROFILE_0": 1518607836,
"TIMESTAMP_PROFILE_COMFORT_LEVELS": 1518604883,
"TIMESTAMP_PROFILE_TIMERS": 1518600089,
"TIMESTAMP_PROFILE_TIMERS_0": 1518607918,
"TIMESTAMP_SYSTEM": 1518607836,

"devices": [{"ACTIVE_LEVEL": 2,
"ACTIVE_PROFILE": 25,
"ACTUAL_TEMP": "25.7",
"AVAILABLE_MODES": ["heat"],
"AWAY": false,

"COOL_MODE": false,
"COOL_ON": false,
"COOL_TEMP": 0,
"CURRENT_FLOOR_TEMPERATURE": 127,
"DATE": "Wednesday",
"DEVICE_ID": 1,
"FAN_CONTROL": "Automatic",
"FAN_SPEED": "Custom",
"FLOOR_LIMIT": false,
"HC_MODE": "VENT",
"HEAT_MODE": true,
"HEAT_ON": false,
"HOLD_OFF": true,
"HOLD_ON": false,
"HOLD_TEMP": 5,
"HOLD_TIME": "0:00",
"HOLIDAY": false,
"LOCK": false,
"LOW_BATTERY": false,
"MANUAL_OFF": true,
"MODELOCK": false,
"MODULATION_LEVEL": 0,
"OFFLINE": false,
"PIN_NUMBER": "1111",
"PREHEAT_ACTIVE": false,
"RECENT_TEMPS": [
"25.6","25.7","25.8","25.9","26.0","26.2","26.3","26.2","26.5","26.7","26.8","26.7","26.7","26.8,"27.0","26
.9","26.4","26.0","25.5","24.9","24.6","24.1","24.0","24.0","24.0","23.8","23.6","23.""23.3","23.1","22.9","
22.7","22.6","22.4","22.3","22.1","22.0","22.0","21.6","21.1","20.9","20.5,"20.4","20.3","20.2","20.1","20.
0","19.9",19.8","19.7","19.8","19.7","19.6","19.6","19.5","19.3,"19.4","19.3","19.3","19.2","19.1","19.1","
19.1","19.0","19.0","19.0","19.0","20.0","20.5","21.1,"21.5","22.0","22.3","22.6","22.9","23.2","23.5","23.
7","23.9","24.2","24.4","24.5","24.6","24.""25.0","24.9","25.0","25.1","25.2","25.2","25.2","25.3","25.4","
25.5","25.6","25.6"],
"SET_TEMP": "17.0",
"STANDBY": false,
"SWITCH_DELAY_LEFT": "0:00",
"TEMPORARY_SET_FLAG": false,
"THERMOSTAT": true,
"TIME": "13:08",
"TIMER_ON": false,
"WINDOW_OPEN": false,
"WRITE_COUNT": 110,
"ZONE_NAME": "Bathroom"
},

The devices section is then duplicated for everything attached to the system.

Not all variables in this list are used by every device, for example the Lock variable is not used by a neoplug so will always return False. However, the write count is incremented when a command is received and actioned. It is therefore used by all devices.

Below are the cached files returned by a Neohub version 2081

"TIMESTAMP_SYSTEM": 1518607836,

System Files

{“GET_SYSTEM”:0}

Returns

{
"ALT_TIMER_FORMAT": 2,
"CORF": "C",
"DEVICE_ID": "NeoHub",
"DST_AUTO": true,
"DST_ON": false,
"FORMAT": 2,
"HEATING_LEVELS": 4,
"HEATORCOOL": "HeatOnly",
"HUB_VERSION": 2081,
"NTP_ON": "Running",
"PARTITION": "4",
"TIMESTAMP": 1518607836,
"TIME_ZONE": 0,
"UTC": 1518611554

}

"TIMESTAMP_DEVICE_LISTS": 1518607836,

Device Lists Timestamps keeps track of anything added or removed from the system.

It therefore reflects changes made to any of the following

“GET_DEVICES” “GET_DEVICE_LIST” “GET_ZONES” “DEVICES_SN”.

Examples

{"GET_ZONES":0} returns the zone name and ID number of all Neostats

{"Bathroom": 1,"Room name ": 2,"Office": 3,"plug": 4}

Additional equipment that can be added to a Neohub are listed as devices a full list of these devices is stored in the Devices list. To access this list use

{"GET_DEVICES":0} returns a list of all devices except Neostats

{"result": ["plug"]}

In this case we have a device called plug

Once on the system a device can be linked to a Neostat Zone name.

{“GET_DEVICE_LIST”:"room name"} returns

{"room name": []}

In this case nothing is linked to "room name"

{"DEVICES_SN":0} returns the serial numbers for all attached devices.

Engineers data

"TIMESTAMP_ENGINEERS": 1518607837,

Contains setup information about all the Neostats

Example

{“GET_ENGINEERS”:0}

{
"Bathroom": {
"DEADBAND": 0,
"DEVICE_ID": 1,
"DEVICE_TYPE": 1,
"FLOOR_LIMIT": 28,
"FROST_TEMP": 12,
"MAX_PREHEAT": 3,
"OUTPUT_DELAY": 0,
"PUMP_DELAY": 0,
"RF_SENSOR_MODE": "self",
"STAT_FAILSAFE": 0,
"STAT_VERSION": 101,
"SWITCHING DIFFERENTIAL": 1,
"SWITCH_DELAY": 0,
"SYSTEM_TYPE": 0,
"TIMESTAMP": 1518535921,
"USER_LIMIT": 0,
"WINDOW_SWITCH_OPEN": false
},

this data is then repeated for every device on the system.

"TIMESTAMP_PROFILE_COMFORT_LEVELS": 1518604883,

Profile comfort levels contains the saved thermostat profiles.

"TIMESTAMP_PROFILE_TIMERS": 1518600089,

Profile timers contains the saved time clock profiles.

"TIMESTAMP_PROFILE_0": 1518607836,

Refers to the last time there was a change on the profile 0 (comfort levels) of any device. Accessed by **{“GET_PROFILE_0”:<device>}.**

"TIMESTAMP_PROFILE_TIMERS_0": 1518607918,

Refers to the last time there was a change on the profile 0 (timer levels) of any device. Accessed by **{“GET_TIMER_0”:<device>}.**

If a device doesn’t have an active profile and the TIMESTAMP_PROFILE_COMFORT_LEVELS or TIMESTAMP_PROFILE_TIMERS is newer than the last time, then you should update the cache for that device

If a device has an active profile and the TIMESTAMP_PROFILE_COMFORT_LEVELS or TIMESTAMP_PROFILE_TIMERS isn't newer than the last time, then an update to the cache isn't needed

Comfort levels

A comfort level is 2 variables these are time and temperature. For heating and cooling systems there are 4 variables

The thermostat will attempt to achieve the required temperature at the stated time, then maintain that temperature until the next comfort level is reached.

By default, there are 4 comfort levels per day this can be increased to 6 comfort levels per day.

The commands to do this are

{"SET_LEVEL_4":0} sets 4 comfort levels per day

{"SET_LEVEL_6":0} sets 6 comfort levels per day.

A group of comfort levels is called a **Profile**

Read comfort levels

Returns the complete set of comfort levels stored in a thermostat.

{"READ_COMFORT_LEVELS":<device(s)>}

This command has been deprecated use GET_PROFILE_0

SET_COMFORT_LEVELS

This command has been deprecated use STORE_PROFILE _0

{"SET_COMFORT_LEVELS":<device(s)>}

Note *a switching time of 24:00 hours or 00:00 hours will disable that level, the target temperature will then remain where it was set by the last valid comfort level, until the next comfort level with a valid time is reached or the user changes it manually. This will show as 2 dashes – on the thermostat.*

Time and Date settings.

The Neohub will automatically connect to the network time server to get the current time and date(GMT)this function can be disabled. If disabled, the time and date will need to be set manually. The Neohub will then use its built-in clock to synchronise all devices. The back-up battery will keep the clock running if power fails for up to 4 hours.

The time and date in the Neohub can be seen in Unix format as UTC in system cache. HUB TIME shown in live data includes daylight savings and time zone adjustments and is the time shown on the Neostats.

{"NTP_OFF":0} disconnects the neohub from the time server
{"NTP_ON":0} will attempt to reconnect to the time server,

note in some circumstances the reconnection attempt will fail, the neohub will then keep trying to reconnect until it succeeds.

Setting the Date

Under normal circumstances you will not need to set the date, but it can be set if the network time server (ntp) cannot be contacted, please note setting the time/date manually will turn off the ntp.

{"SET_DATE":[2018, 12, 09]} Sets the current date

{"SET_TIME":[14,25]} Sets the current time

The current time is available in Unix format in the live data array.

Time zone adjustment.

{"TIME_ZONE":10.5} will adjust the clock 10 hours 30 minutes ahead of GMT, the .5 is for those areas that use half hour offsets

{"TIME_ZONE":-5.0} will adjust the clock backwards by 5 hours from GMT.

Because most countries use UTC + Time Zone + DST it is possible to set the time by only adjusting the time zone variable, but only in 15-minute steps.

Daylight Savings

THE DAYLIGHT SAVINGS COMMAND SHOULD ONLY BE USED IN EUROPEAN COUNTRIES.

This is because daylight savings times and dates vary from country to country and sometimes year to year around the world. The function automatically advances the clock by 1 hour on the last Sunday of March and reduces it by 1 hour on the last Sunday of October.

{"DST_ON":0} turns the daylight-saving function on and {"DST_OFF":0} turns the daylight-saving function off

When turning daylight savings off you must also send the **{"MANUAL_DST":0}** command to prevent errors.

Manual DST

Adds 1 hour to the neohub time which moves the system to daylight saving

{"MANUAL_DST":0} Neohub Time = (GMT + Time zone) = no daylight savings

{"MANUAL_DST":1} Neohub Time = (GMT + Time zone) +1 = daylight savings of 1 hour.

Adding thermostats, devices and accessories to the network.

Most devices are added to the system using the standard command

{"PERMIT_JOIN":[120,"kitchen"]}

In this case you have 120 seconds to pair the device you want to call kitchen.

The device will show up as a zone in the get Zones list or a device in the Get Devices list.

However, repeaters or boosters use a different paring command

{"PERMIT_JOIN":["repeater", 120]}

In this case you have 120 seconds to pair your repeater to the neohub.

Each repeater will then be visible as repeaternodexxxxx, where xxxx is a random but unique identifier.

Repeater status information is available in the Live data function.

Care must be taken when removing repeaters from the system as the same repeater can be added multiple times, each time with a new unique ID but the old status information will remain until it is properly removed.

Zone Title

Used to change the room name after installation

{"ZONE_TITLE":["HCtest","HCtest2"]}

Removing a device

{"REMOVE_ZONE":"test"}

This command will tell the device called "test" to disconnect from the network and all reference to it in the neohub will be deleted.

The wifi connected symbol on the thermostats will go off and accessing feature setting 01 will show no connection.

Note if the command fails to reach the neostat before the deletion process within the hub completes the neostat will need to be factory reset before it can be paired to the system again.

Similarly {"REMOVE_REPEATER":<repeater>} will remove a repeater.

Neoplugs do not indicate when they are connected instead they flash a red led every 20 seconds to indicate no connection.

Devices

Neostats are listed as zones within the system, anything that is not a Neostat or NeoAir is a Device.

Neoplugs are the most common devices.

It is possible to link devices to zones by linking them to a neostat that's already in that zone, this allows all neo devices to be displayed under the same zone name. The commands to do this are

{"GET_DEVICES":0}

Returns a list of all devices attached to the Neohub.

LINK_DEVICE

{"LINK_DEVICE":[<zone>,<device>]}

Links a device to a thermostat zone.

Example

{"LINK_DEVICE":["Kitchen ","neoplug"]}

{"GET_DEVICE_LIST":"zone name"}

Returns a list of devices linked to a Neostat zone, e.g. kitchen, living room etc.

DETACH_DEVICE

Detaches a device from the a Neostat zone.

{"DETACH_DEVICE":["zone name ","device name "]}

Example

{"DETACH_DEVICE":["Kitchen ","neoplug "]}

Note the link is broken but both devices remain connected to the system.

DEVICES_SN returns the serial number of the device

{"DEVICES_SN":0}

CLEAR DEVICE LIST

Deletes all items in a zones device list and disconnects them

Example

{"CLEAR_DEVICE_LIST":"Kitchen"}

Profiles

Profiles are a collection of comfort levels or a collection of switching times for timeclocks

The exact layout of a profile is determined by the program mode and number of comfort levels in a day.

7day mode with 4 levels has 28 comfort levels in a profile

7day mode with 6 levels has 42 comfort levels in a profile

5/2 day mode with 4 levels has 8 comfort levels in a profile

5/2 day mode with 6 levels has 12 comfort levels in a profile

24hour mode with 4 levels has 4 comfort levels in a profile

24hour mode with 6 levels has 6 comfort levels in a profile

Profiles cannot be used in nonprogrammable mode because there are no comfort levels.

Timeclocks always have 4 on and 4 off times regardless of the number of comfort levels per day that have been set. Timeclock profiles therefore have a different structure and different commands than comfort level profiles.

7day mode has 28 on times and 28 off times in a profile

5/2day mode has 8 on times and 8 off times in a profile

24hour mode has 4 on times and 4 off times in a profile

Time clocks cannot be set to nonprogrammable mode.

It follows that changing between program modes and the number of levels will affect the operation of saved profiles, for this reason changing program mode will automatically delete any saved profiles.

Json structures within profiles.

The introduction of 6 levels forced a change in the layout of the structure from the original..

wake, leave, return and sleep which is still used for 4 comfort levels per day to -

wake, 1, 2, 3, 4, sleep which is used for 6 comfort levels.

This was done to maintain backward compatibility with older Neostat hardware.

For Neohub firmware versions up to Version 2077 the JSon structure for each comfort level is "wake":["07:00",21], from version 2079 onwards this has changed to "wake":["10:00",25.0,0.0,false].

The variables are [time, temperature 1, temperature 2, Enable temperature 2.] The difference is due to upcoming products that at the time of writing have not been released.

An example of a complete profile for a Neohub in 5/2day mode with 4 levels prior to version 2079 is given below.

4 levels per day

{"sunday": {"wake":["07:00",21], "leave":["09:30",17], "return":["17:30",21], "sleep":["22:30",15]},

"monday": {"wake":["07:00",14], "leave":["09:30",17], "return":["17:30",25], "sleep":["22:30",15]

6 levels per day

{"monday":{"level1":["17:00",14],"level2":["23:00",16],"level3":["22:00"],"level4":["22:05",15],"sleep":["23:45",35],"wake":["10:00",25]},

"sunday":{"level1":["24:00",23],"level2":["24:00",18],"level3":["24:00",21],"level4":["24:00",16],"sleep":["24:00",16],"wake":["24:00",21]

From Neohub version 2079 onwards this has changed to

4 levels

{"sunday": {"wake":["07:00",21.0,0.0,false], "leave":["09:30",17.0,0.0,false], "return":["17:30",21.0,0.0,false], "sleep":["22:30",15.0,0.0,false]},

"monday": {"wake":["07:00",14.0,0.0,false], "leave":["09:30",17.0,0.0,false], "return":["17:30",25.0,0.0,false], "sleep":["22:30",15.0,0.0,false]

6 levels

{"monday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"sunday":{"level1":["24:00",23.0,0.0,false],"level2":["24:00",18.0,0.0,false],"level3":["24:00",21.0,0.0,false],"level4":["24:00",16.0,0.0,false],"sleep":["24:00",16.0,0.0,false],"wake":["24:00",21.0,0.0,false]

Note the order of each level is not significant but all levels must be included and match the Neohub settings.

Complete tested examples for all modes of these profiles are given in Appendix C

Saving profiles

Profiles are stored in the Neohub using the STORE PROFILE COMMAND

{"STORE_PROFILE":{"info":{"sunday": {"wake":["07:00",21], "leave":["09:30",17], "return":["17:30",21], "sleep":["22:30",15]}, "monday": {"wake":["07:00",14], "leave":["09:30",17], "return":["17:30",25], "sleep":["22:30",15]}}, "name":"my profile"}}

{"STORE_PROFILE"{info": {"monday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","24:00"],"time4": ["24:00","24:00"]},"sunday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","24:00"],"time4": ["24:00","24:00"]}},"name": " my timer profile"}}

When saved they are automatically given a profile id number, this profile id can then be used to load the profile into a Neostat without sending all the data again.

Retrieving profiles

To load a complete list of all profiles stored on the Neohub use the GET PROFILES command

{"GET_PROFILES":0} which returns all the thermostat profiles complete with names and profile ids

{"my profile": {"PROFILE_ID": 25,"info": {"monday": {"leave": ["09:30",17],"return": ["17:30",25],"sleep": ["22:30",15],"wake": ["07:00",14]},"sunday": {"leave": ["09:30",17],"return": ["17:30",21],"sleep": ["22:30",15],"wake": ["07:00",21]}},"name": "my profile"}}

{"GET_PROFILE_NAMES":0} returns a list of thermostat profile names.

{"GET_PROFILE":"kitchen"} returns the named profile.

{"GET_PROFILE_TIMERS":0} returns timeclock profiles.

Example

{"GET_PROFILE_TIMERS":0}

{"Timer": {"PROFILE_ID": 26,"info": {"monday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","24:00"],"time4": ["24:00","24:00"]},"sunday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","24:00"],"time4": ["24:00","24:00"]}},"name": "Timer"}}

Editing Profiles

To edit profiles, you must first load the profile then over write the existing data, the profile ID number and the profiles timestamp will be updated.

Example

GET_ PROFILES returns the original

'ball': {'PROFILE_ID': 25,'info': {'monday': {'leave': ['09:30',17],'return': ['17:30',25],'sleep': ['22:30',15],'wake': ['07:00',14]},'sunday': {'leave': ['09:30',17],'return': ['17:30',21],'sleep': ['22:30',15],'wake': ['07:00',21]}},'name': 'ball'}}

Sending

{'STORE_PROFILE': {'PROFILE_ID': 25,'info': {'monday': {'leave': ['09:30',17],'return': ['17:30',25],'sleep': ['22:30',15],'wake': ['01:00',14]},'sunday': {'leave': ['09:30',17],'return': ['17:30',21],'sleep': ['22:30',15],'wake': ['07:00',21]}},'name': 'ball'}}

GET_ PROFILES then returns the modified profile with a new profile id but the same name.

{"ball": {"PROFILE_ID": 28,"info": {"monday": {"leave": ["09:30",17],"return": ["17:30",25],"sleep": ["22:30",15],"wake": ["01:00",14]},"sunday": {"leave": ["09:30",17],"return": ["17:30",21],"sleep": ["22:30",15],"wake": ["01:00",21]}},"name": "ball"}}

Changing profile names

{"PROFILE_TITLE":["old name","new name"]}

Example

{"PROFILE_TITLE":["summer","winter"]}

Activating Profiles

Once the profile ID is known it can be used with the command

RUN_PROFILE_ID

{“RUN_PROFILE_ID”:[25,"Kitchen","Lounge"]} which will make the Neohub send the complete profile to each Neostat named in the command. The Neostats will report the profile id they are using in the live data field. Eliminating the need to send the full profile more than once to the App.

Deleting Profiles.

Profiles can be deleted by name or ID number.

Examples

{"CLEAR_PROFILE":"winter"}

{"CLEAR_PROFILE_ID":2}

PROFILE 0

Comfort levels or switching times that have been programmed into a Neostat but have not been saved to the Neohub are always called profile_0. These profiles use the profile id of 0 which is reported as 0 in the live data field and indicates the need to read the device directly. This is done using the GET_PROFILE_0 command

GET_PROFILE_0 returns the data in the named thermostat.

Example

{"GET_PROFILE_0":"Bathroom"} which returns

{"TIMESTAMP": 1518180429,"profiles": [{"device": "Bathroom","monday": {"leave": ["09:30",17],"return": ["17:30",25,],"sleep": ["22:30",15],"wake": ["07:00",14]},"sunday": {"leave": ["09:30",17],"return": ["17:30",21,],"sleep": ["22:30",15],"wake": ["07:00",21]}}]}

GET_TIMER_0

{"GET_TIMER_0":"Office"} returns the data in the named timeclock

{"TIMESTAMP": 1518535946,"profiles": [{"device": "Office","monday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","00:00"],"time4": ["24:00","00:00"]},"sunday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","00:00"],"time4": ["24:00","00:00"]}}]}

Saving profiles directly to the Device

STORE PROFILE 0 this command will store the profile directly to the named thermostat.

Example

{"STORE_PROFILE_0":[{"sunday": {"wake":["07:00",22], "leave":["24:00",17],"return":["24:00",21], "sleep":["22:30",17]},"monday": {"wake":["07:00",34], "leave":["24:00",17],"return":["24:00",21], "sleep":["22:30",15]}},"bedroom1"]}

STORE_PROFILE_TIMER_0 this command will store the profile directly to the named timeclock.

Example

{"STORE_PROFILE_TIMER_0":[{"monday":{"time1":["01:11","09:00"],"time2":["16:00","20:00"],"time3" :["24:00","00:00"],"time4":["24:00","00:00"]},"sunday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]}},"Office"]}

The following profile commands have been deprecated. The original profile command structure included the target devices within it. The newer profiles structure does not. Note for backward compatibility these commands are still valid.

{"STORE_PROFILE":<profile ob>} – deprecated

{"CLEAR_PROFILE":<profile name>} – deprecated

{"GET_PROFILE":<profile name>} – deprecated

{"RUN_PROFILE":<profile name>} – deprecated

{"GET_PROFILE_NAMES":0} – deprecated

Boost Function

Used to turn a timeclock on or off which overrides the schedule for a fixed length of time.

To turn it on when its off use BOOST ON to turn it off when its, on use BOOST OFF

{"BOOST_ON":[{"hours":1,"minutes":10},["floor1","clock"]]}

to cancel send

{"BOOST_ON":[{"hours":0,"minutes":0},["floor1","clock"]]}

{"BOOST_OFF":[{"hours":1,"minutes":10},["floor1","clock"]]}

to cancel send

{"BOOST_OFF":[{"hours":0,"minutes":0},["floor1","clock"]]}

Engineers settings

Engineers settings are those found in the neostat feature menu that is accessed on the device itself through the setup menu. Note changing these setting affects the target device only.

Settings that must be done during installation only are not available.

Some of these settings can be edited remotely.

{"SET_DIFF":[2,["test"]]} adjust the switching differential in thermostat to 2 degrees.

{"SET_FLOOR":[28, "floor1"]} sets the upper limit to the floor temperature.

{"SET_PREHEAT":[3,["test"]]} sets the maximum preheat time in the optimization function.

{"SET_FROST":[9,"test"]} sets the frost protection temperature.

{"SET_DELAY":[5, "test"]} sets the output delay in minutes.

Failsafe

Used on NeoAir thermostats to enable the failsafe function in the radio signal sent to the Rf Switch and UH8-RF radio receivers. If no signal is received for 50 minutes fail safe brings on the heating for 12 minutes every hour, until signal is received.

Example

{'SET_FAILSAFE':[true, 'neoair']} to turn the failsafe on

{'SET_FAILSAFE':[false, 'neoair']} to turn the failsafe off

{"FIRMWARE":0} Returns the Neohub firmware version

Create Group

Groups different zones under a command group name

Allows a single command to be sent to all devices in the group.

{"CREATE_GROUP":[["lounge","bed1"], "bob"]}

This puts the "lounge" and "bed1" in the "bob" group.

Any command sent to bob will be sent to the lounge and bed1.

{"GET_GROUPS":0} Returns a list of groups

{"DELETE_GROUP":"bob"} Deletes the group bob but has no effect on the members of the group.

{"CANCEL_HGROUP":<group>} cancels the holds in any group.

Hold Function

Tells a thermostat to maintain the current temperature for a fixed time.

To hold the heating temperature

{“HOLD”:[{“temp”:16, “hours”:2, “minutes”:30",id":"Box"}, },<devices>]}

To hold the cooling temperature

{“HOLD”:[{ “cool”:24, “hours”:2, “minutes”:30, "id":"Box"}, },<devices>]}

To hold both heating and cooling temperatures

{“HOLD”:[{“temp”:16, “cool”:24, “hours”:2, “minutes”:30, "id":"Box"}, },<devices>]}

To cancel

{“HOLD”:[{“temp”:16, “cool”:24, “hours”:0, “minutes”:00, "id":"Box"}, },<devices>]}

The Neohub will read the hold time and if 00 cancel the hold so the other variables will be ignored.

Cancel hold all

Cancels all hold commands

Example {"CANCEL_HOLD_ALL":0}

{"GET_HOLD":0} returns the active Holds set by the App but not on the device.

Identify

Used to identify a device on the system. Introduced for Homekit compliance but can be used directly.

{"IDENTIFY":0}

This command flashes the link led on the neohub. To be used correctly it needs to be sent to the correct ip-address for the neohub. It is therefore best used as confirmation that you have the correct ip address.

{"IDENTIFY_DEV":"test"}

This command flashes the lcd backlight on neostats connected to a neohub. It serves to indicate which physical device matches the name.

INFO command this command returns real time data from all devices connected to a Neohub

{"INFO":0}. deprecated use the live data command.

Lock command

This command pin locks the thermostats.

This command can be used in conjunction with the user limit to allow limited access to the device.

LOCK

{"LOCK":[[<pin1>,<pin2>,<pin3>,<pin4>], <device(s)>]}

Possible results

{"result":"locked"}

{"error":"lock failed1"} (setting pin failed)

{"error":"lock failed2"} (pin was set but locking failed)

{"error":"LOCK arguments not in an array"}

{"error":"LOCK first (pin) argument not in an array"}

{"error":"Invalid argument to LOCK, should be array of 4 integers (0-9)"}

{"error":"Invalid second argument to LOCK, should be a valid device or array of valid devices"}

Example

{"LOCK":[[1,2,3,4], ["Bathroom","kitchen"]]} Locks both the Bathroom and kitchen Neostats with pin 1234

UNLOCK

{"UNLOCK":<device(s)>}

Example

{"UNLOCK":["Bathroom","kitchen"]} Will unlock both the bathroom and the kitchen

Neoplug commands

The Neoplugs use the same structure as any of the timeclocks and therefore use the same commands with the addition of the on/off command

See Time Clock Commands page 26

Standby function

The standby function was initially called frost protection on, which automatically disabled the schedule and enabled frost protection, the command reflects this original usage.

{"FROST_ON":<device(s)>}

Disables the schedule and sets the target temperature to the preset frost protection temperature.

Note that when this is active the highest temperature that can be set is 17 degrees Celsius.

Possible results

{"result":"frost on"}

{"error":"Could not complete frost on"}

{"error":"Invalid argument to FROST_ON, should be a valid device or array of valid devices"}

Example

{"FROST_ON":["bed1","lounge"]}

For timeclocks it disables the schedule and turns the output off

{"FROST_OFF":<device(s)>}

Cancels frost protection and reinstates the schedule.

Possible results

{"result":"frost off"}

{"error":"Could not complete frost off"}

{"error":"Invalid argument to FROST_OFF, should be a valid device or array of valid devices"}

Example

{"FROST_OFF":["bed1","lounge"]}

Summer

A variation of the Frost command it was originally used to put all thermostat into frost protection but has no effect on time clocks

{"SUMMER_OFF":["Bathroom"]}

{"SUMMER_ON":["lounge","bed1"]}

STATISTICS

{"GET_HOURSRUN":<devices>}

Returns the number of hours each week the output was on in the named device.

Example

{"GET_HOURSRUN":"Kitchen"}

{"GET_TEMPLOG":<device(s)>}

Returns the temperature reading for the previous 7 days and up to the current time in 15minute intervals.

Example

{"GET_TEMPLOG":["Kitchen, "bedroom"]}

{"STATISTICS":0} deprecated do not use.

HOME KIT

"Homekit": true, This flag indicates the Neohub is compatible with apple home kit for details on how to use the home kit features please contact Apple.

The absence of this flag indicates the device is not compatible.

NB this flag will be deprecated and replaced by a HUB_TYPE variables

Currently there are 2 types G1 and G2 where G2 is home kit compatible. Others are expected to be added.

Time clock commands.

Commands specific to time clocks and neoplugs

TIMER_HOLD_ON

If a device output is currently off or unknown, this command will override it on for the stated time the schedule will be ignored.

{"TIMER_HOLD_ON":[<minutes>, <device(s)>]}

Minutes is a number, 0 cancels

Examples

{"TIMER_HOLD_ON":[15, "clock"]} to cancel {"TIMER_HOLD_ON":[0, "clock"]}

TIMER_HOLD_OFF

If a device output is currently on or unknown, this command will override it off for the stated time the schedule will be ignored.

{"TIMER_HOLD_OFF":[<minutes>, <device(s)>]}

Minutes is a number, 0 cancels

Examples

{"TIMER_HOLD_OFF":[15, "clock"]} to cancel {"TIMER_HOLD_OFF":[0, "clock"]}

Neoplug commands

The neoplugs use the same structure as any of the timeclocks and therefore use the same commands with the addition of the on/off command

TIMER_ON turns the output on {"TIMER_ON":"plug"}

TIMER_OFF turns the output off {"TIMER_OFF":"plug"}

Manual on / off enables or disables the timeclock built into the neoplug

{“MANUAL_OFF”:<devices>} {“MANUAL_ON”:<devices>}

This allows the timeclock to be switched off and puts the neoplug fully under user control.

Time clock switching times

Returns the full set of programmed on/off times from a neostat, neoplug or Neoair.

Note when a neoair is operating in combined mode a full set of comfort levels and switching times is available in the device. The Neohub will treat it as 2 different devices so this command will only return the time clock settings.

READ_TIMECLOCK

{"READ_TIMECLOCK":<device(s)>}

Possible results

an object with keys named after the requested devices, the values will be objects with weekday names as keys, of which the value will be yet another object with keys "time1", "time2", "time3" and "time4", the values there will be an array with the start and end time for that period:

{"kitchen":{"friday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]},"monday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"saturday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]},"sunday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"thursday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]},"tuesday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"wednesday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]}}}

Example

{"READ_TIMECLOCK":"Hot water"}

SET_TIMECLOCK

{"SET_TIMECLOCK":[<levels>, <device(s)>]}

For <levels> see READ_TIMECLOCK result

Example

{"SET_TIMECLOCK":[{"friday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]},"monday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"saturday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]},"sunday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"thursday":{"time1":["07:00","09:00"],"time2":["16:00","20:00"],"time3":["24:00","00:00"],"time4":["24:00","00:00"]},"tuesday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"wednesday":{"time1":["01:00","02:00"],"time2":["03:00","04:00"],"time3":["05:00","06:01"],"time4":["12:34","15:59"]},"kitchen"]}

User Limit

The user limit sets a range limit for the target temperature settings. Any non-zero number sets the limit to that value.

So if the programmed temperature is 20 degrees and the user limit is set to 2. Then the user can only adjust the target temperature up or down by 2 degrees using the buttons on the thermostat.

When the thermostat is locked the user limit allows minor adjustments to the target temperature without the need to unlock the device.

This function is useful in public areas or children's rooms.

Setting the user limit to 0 removes the restrictions on unlocked thermostats and reinstates the lock on locked devices so that a pin number is always required to change the temperature.

Example

{"USER_LIMIT":[5, ["bed1","lounge"]]}

to cancel use

{"USER_LIMIT":[0, ["bed1","lounge"]]}

Optimisation

Calculates how long it will take for a room to reach the next comfort level target temperature.

If enabled the thermostat will start preheating the room before the programmed switching time to ensure it reaches the desired temperature at the stated time.

The user can set a maximum pre heat period to ensure they are not disturbed by the heating system starting up too early in the morning. This can be set on the device or with the set preheat command

{"SET_PREHEAT":[hours, <device(s)>]}

Example

{"SET_PREHEAT":[3, "Bathroom"]} sets maximum preheat time to 3 hours.

ROC

ROC is the rate of change, this is the time it takes the heating system to raise the room temperature by 1 degree. It is used in the optimum start function and is recalculated at every comfort level.

The calculated value van be viewed on the device itself or by using the view_roc command.

{"VIEW_ROC":["Bathroom1","HCtest"]}

Will return the calculated rate of change for these rooms in minutes per degree.

Set Temperature

Used to adjust the thermostat's set temperature.

This setting is temporary and will be reset when the next comfort level is reached, if you need to adjust the temperature beyond this then use the hold command.

{"SET_TEMP":[34,["Bathroom","test"]]}

Will set the bathroom and test room to 34 degrees

Cooling

Changes have been made in the structure of profiles to allow for the introduction of the Neostat HC this device is a heating and cooling fan coil thermostat. To use these devices the Neohub firmware must be up to date.

The most significant change is in the JSon structure of the comfort levels

The new structure is composed of set temperatures which can be set in 0.5 degrees increments but this has not yet been introduced to the hardware and must not be used. The structure format is

[time, heating temperature, cooling temperature, cooling enabled.].

Example

"wake":["10:00",25.0,28.0,true]

The profile examples in Appendix A show the complete structures including the cooling variables.

Mixing existing Neostat and Neostat HC devices on the same system is allowed, the Neostats will just ignore the cooling information.

The Neostat HC is designed to work with different types of heating and cooling systems. But every device on the system must be in the same mode or they all must be completely independent.

Regardless of the different systems the heating systems the Neostat can be in 1 of 3 modes

Heating or cooling, cooling only and independent. Which mean the neo system needs to be set up to match.

Setting up the system

Step 1 Set up the Global System type using the Global System type command

{"GLOBAL_SYSTEM_TYPE":"HeatOrCool"}

This will allow the Neostat HC to be in heating or cooling mode, this is normally used for 2 pipe district heating systems which provide heat in the winter and cooling in the summer.

{"GLOBAL_SYSTEM_TYPE":"CoolOnly"}

This will allow the Neostat HC to be in cooling mode only, this is normally used for 2 pipe cooling systems which only provide cooling.

{"GLOBAL_SYSTEM_TYPE":"Independent"}

This will allow the Neostat HC's to be in any mode required for the system on a room by room basis.

It should also be used with mixed systems.

These are global setting and will affect the entire system and limit the choices in step 2

Step 2 Adjust the Neohub settings as required.

For {"GLOBAL_SYSTEM_TYPE":"HeatOrCool"} you can set

1. {"SET_GLOBAL_HC_MODE":"heating"}
2. {"SET_GLOBAL_HC_MODE":"cooling"}

For {"GLOBAL_SYSTEM_TYPE":"CoolOnly"} you can only set

1. {"SET_GLOBAL_HC_MODE":"cooling"}

For {"GLOBAL_SYSTEM_TYPE":"Independent"} you can set

1. {"SET_GLOBAL_HC_MODE":"heating"}
2. {"SET_GLOBAL_HC_MODE":"cooling"}
3. {"SET_GLOBAL_HC_MODE":"auto"}

These are global settings and will affect the entire system and limit the choices in step 3

Step 3 is automatically carried out by step 2, for heating and cooling systems and cooling only systems. But **must** be carried out for independent systems.

Step 3

This sets the mode on the individual thermostats

{"SET_HC_MODE":["COOLING",["Device1", " device2", "device3","deviceN"]]}

Sets the HC to cooling mode

{"SET_HC_MODE":["HEATING", ["Device1", " device2", "device3","deviceN"]]}

Sets the HC to heating mode

{"SET_HC_MODE":["AUTO", ["Device1", " device2", "device3","deviceN"]]}

Sets the HC to auto mode which means it can heat or cool the room as required.

{"SET_HC_MODE":["VENT", ["Device1", " device2", "device3","deviceN"]]}

Runs the fan on the fan coil without heating or cooling the room, as the name suggests it is used to circulate air in the room or draw in air from the outside.

VENT mode is always available regardless of the system settings in step 1 and step 2. The stat will alternate between vent mode and heating or cooling depending on the settings in step 2.

The Neostat HC will respond to all normal commands however, if the target temperature is set to 36 or above it will disable cooling for that level. The lowest cooling temperature that can be set is 18 degrees C.

Neostat HC commands

Those commands that are unique to the Neostat HC are below.

To set the cooling temperature use

{"SET_COOL_TEMP": temp, < device >}

Example

{"SET_COOL_TEMP":[27,["HCtest"]]}

{"SET_COOL_TEMP":[27,[" device1", " device2", "device3","deviceN"]]}

The speed of the Fans can be set manually or put in automatic control.

Please note in Vent mode automatic control is disabled.

To set the Fan Speed use

{"SET_FAN_SPEED":["HIGH",["device1", " device2", "device3","deviceN"]]}
{"SET_FAN_SPEED":["AUTO",["device1", " device2", "device3","deviceN"]]}
{"SET_FAN_SPEED":["MED", ["device1", " device2", "device3","deviceN"]]}
{"SET_FAN_SPEED":["LOW", ["Device1", " device2", "device3","deviceN"]]}
{"SET_FAN_SPEED":["OFF", ["Device1", " device2", "device3","deviceN"]]}

{'AUTO_MODE_OFF':["Device1", " device2", "device3","device"]} will disable the automatic fan speed control but leave the fan running at it's current speed

Appendix A

Profiles.

Worked examples of profile 0 and named profiles for each of the programmable modes and 4 and 6 levels.

PROFILE 0

4 levels	
5/2 day	{"STORE_PROFILE_0":[{"monday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,28.0,true]},"sunday":{ "leave":["24:00",21.0,0.0,false],"return":["24:00",16.0,0.0,false],"sleep":["24:00",16.0
24 hour	{"STORE_PROFILE_0":[{"sunday":{ "leave":["24:00",21.0,0.0,false],"return":["24:00",16.0,0.0,false],"sleep":["24:00",16.0,0.0,false],"wake":["11:00",21.0,0.0,false]}},"HCtest"]}
7 day	{"STORE_PROFILE_0":[{"monday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,28.0,true]},"tuesday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"wednesday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"thursday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"friday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"saturday":{"leave":["22:00",14.0,0.0,false],"return":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,28.0,true]},"sunday":{ "leave":["24:00",21.0,0.0,false],"return":["24:00",16.0,0.0,false],"sleep":["24:00",16.0,0.0,false],"wake":["11:00",21.0,0.0,false]}},"HCtest"]}

6 levels	
5/2 Day	{"STORE_PROFILE_0":[{"monday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"sunday":{"level1":["24:00",23.0,0.0,false],"level2":["24:00",18.0,0.0,false],"level3":["24:00",21.0,0.0,false],"level4":["24:00",16.0,0.0,false],"sleep":["24:00",16.0,0.0,false],"wake":["24:00",21.0,0.0,false]}},"HCtest"]}
24 hour	{"STORE_PROFILE_0":[{"sunday":{"level1":["10:00",23.0,29.0,true],"level2":["24:00",18.0,0.0,false],"level3":["24:00",21.0,0.0,false],"level4":["24:00",16.0,0.0,false],"sleep":["24:00",16.0,0.0,false],"wake":["24:00",21.0,0.0,false]}},"HCtest"]}
7 day	{"STORE_PROFILE_0":[{"monday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,28.0,true]},"tuesday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"wednesday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"thursday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"friday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,0.0,false]},"saturday":{"level1":["17:00",14.0,0.0,false],"level2":["23:00",16.0,0.0,false],"level3":["22:00",14.0,0.0,false],"level4":["22:05",15.0,0.0,false],"sleep":["23:45",35.0,0.0,false],"wake":["10:00",25.0,28.0,true]},"sunday":{"level1":["24:00",23.0,0.0,false],"level2":["24:00",18.0,0.0,false],"level3":["24:00",21.0,0.0,false],"level4":["24:00",16.0,0.0,false],"sleep":["24:00",16.0,0.0,false],"wake":["11:00",21.0,0.0,false]}},"HCtest"]}

Named Profiles

4 levels	
5/2 day	{"STORE_PROFILE": {"info": {"sunday": {"leave": ["11:00",11.0, 25.0,true],"return": ["12:00",12.0, 25.0,true],"sleep": ["13:00",13.0, 25.0,true],"wake": ["10:00",10.0, 25.0,true]},"monday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]}},"name": "Test21"}}
24 hour	{"STORE_PROFILE": {"info": {"sunday": {"leave": ["11:00",11.0, 25.0,true],"return": ["12:00",12.0, 25.0,true],"sleep": ["13:00",13.0, 25.0,true],"wake": ["10:00",10.0, 25.0,true]},"monday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]}},"name": "Test23"}}
7 day	{"STORE_PROFILE": {"info": {"sunday": {"leave": ["11:00",11.0, 25.0,true],"return": ["12:00",12.0, 25.0,true],"sleep": ["13:00",13.0, 25.0,true],"wake": ["10:00",10.0, 25.0,true]},"monday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]}, "tuesday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]},"wednesday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]},"thursday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]}, "friday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]}, "saturday": {"leave": ["16:00",16.0, 25.0,true],"return": ["17:00",17.0, 25.0,true],"sleep": ["18:00",18.0, 25.0,true],"wake": ["15:00",15.0, 25.0,true]}},"name": "Test24"}}

6 levels	
5/2 DAY	{"STORE_PROFILE": {"info": {"sunday": {"level1": ["10:00",11.0, 25.0,true],"level2": ["11:00",12.0, 25.0,true],"level3": ["12:00",13.0, 25.0,true],"level4": ["13:00",14.0, 25.0,true],"sleep": ["14:00",15.0, 25.0,true],"wake": ["09:00",16.0, 25.0,true]}, "monday": {"level1": ["16:00",15.0, 25.0,true],"level2": ["17:00",17.0, 25.0,true],"level3": ["18:00",18.0, 25.0,true],"level4": ["19:00",19.0, 25.0,true],"sleep": ["20:00",20.0, 25.0,true],"wake": ["15:00",21.0, 25.0,true]}},"name": "Test8"}}
24 hour	{"STORE_PROFILE": {"info": {"sunday": {"level1":["10:00",15.0, 25.0,true],"level2": ["11:00",15.0, 25.0,true],"level3": ["12:00",15.0, 25.0,true],"level4": ["13:00",15.0, 25.0,true],"sleep": ["14:00",15.0, 25.0,true],"wake": ["09:00",15.0, 25.0,true]}},"name": "Test1"}}
7 day	{"STORE_PROFILE": {"info": {"friday": {"level1": ["11:00",16.0, 25.0,true],"level2": ["12:00",11.0, 25.0,true],"level3": ["13:00",12.0, 25.0,true],"level4": ["14:00",13.0, 25.0,true],"sleep": ["15:00",14.0, 25.0,true],"wake": ["10:00",10.0, 25.0,true]}, "saturday": {"level1": ["11:00",11.0, 25.0,true],"level2": ["12:00",12.0, 25.0,true],"level3": ["13:00",13.0, 25.0,true],"level4": ["14:00",13.0, 25.0,true],"sleep": ["15:00",15.0, 25.0,true],"wake": ["09:00",09.0, 25.0,true]}, "sunday": {"level1": ["10:00",15.0, 25.0,true],"level2": ["10:00",15.0, 25.0,true],"level3": ["10:00",15.0, 25.0,true],"level4": ["10:00",15.0, 25.0,true],"sleep": ["10:00",15.0, 25.0,true],"wake": ["10:00",15.0, 25.0,true]}, "monday": {"level1": ["10:00",15.0, 25.0,true],"level2": ["10:00",15.0, 25.0,true],"level3": ["10:00",15.0, 25.0,true],"level4": ["10:00",15.0, 25.0,true],"sleep": ["10:00",15.0, 25.0,true],"wake": ["10:00",15.0, 25.0,true]}, "tuesday": {"level1": ["02:00",12.0, 25.0,true],"level2": ["03:00",13.0, 25.0,true],"level3": ["04:00",14.0, 25.0,true],"level4": ["05:00",15.0, 25.0,true],"sleep": ["06:00",16.0, 25.0,false],"wake": ["01:00",10.0, 25.0,true]}, "wednesday": {"level1": ["10:00",15.0, 25.0,true],"level2": ["10:00",15.0, 25.0,true],"level3": ["10:00",15.0, 25.0,true],"level4": ["10:00",15.0, 25.0,true],"sleep": ["10:00",15.0, 25.0,true],"wake": ["10:00",15.0, 25.0,true]}, "thursday": {"level1": ["10:00",15.0, 25.0,true],"level2": ["10:00",15.0, 25.0,true],"level3": ["10:00",15.0, 25.0,true],"level4": ["10:00",15.0, 25.0,true],"sleep": ["10:00",15.0, 25.0,true],"wake": ["10:00",15.0, 25.0,true]}},"name": "Test2"}}

Editing Profiles to edit profiles load the original, edit it and save, please note the profile ID number changes.

editing profiles	
get the original	'ball': {'PROFILE_ID': 25,'info': {'monday': {'leave': ['09:30',17],'return': ['17:30',25],'sleep': ['22:30',15],'wake': ['07:00',14]},'sunday': {'leave': ['09:30',17],'return': ['17:30',21],'sleep': ['22:30',15],'wake': ['07:00',21]}},'name': 'ball'}}
edit and store it	{'STORE_PROFILE': {'PROFILE_ID': 25,'info': {'monday': {'leave': ['09:30',17],'return': ['17:30',25],'sleep': ['22:30',15],'wake': ['01:00',14]},'sunday': {'leave': ['09:30',17],'return': ['17:30',21],'sleep': ['22:30',15],'wake': ['07:00',21]}},'name': 'ball'}}
get the updated copy	{"ball": {"PROFILE_ID": 28,"info": {"monday": {"leave": ["09:30",17],"return": ["17:30",25],"sleep": ["22:30",15],"wake": ["01:00",14]},"sunday": {"leave": ["09:30",17],"return": ["17:30",21],"sleep": ["22:30",15],"wake":

Note to use any of the above profiles, make sure you change the profile's name or create a zone with the same name.

Appendix B

The following commands have been deprecated.

Note for backward compatibility these commands are still valid but should not be used in new designs.

{"INFO":0}

{"ENGINEERS_DATA":0}

{"STORE_PROFILE":<profile ob>}

{"CLEAR_PROFILE":<profile name>}

{"GET_PROFILE":<profile name>}

{"RUN_PROFILE":<profile name>}

{"GET_PROFILE_NAMES":0}

Appendix C

Alphabetical Command list.

		Page
AUTO_MODE_OFF	{'AUTO_MODE_OFF':'HCtest'}	33
AWAY_OFF	{"AWAY_OFF":<device(s)>}	7
AWAY_ON	{"AWAY_ON":<device(s)>}	7
BOOST_OFF	{"BOOST_OFF":[{"hours":0,"minutes":10},<devices>]}	21
BOOST_ON	{"BOOST_ON":[{"hours":0,"minutes":10},<devices>]]}	21
CANCEL_HGROUP	{"CANCEL_HGROUP":<group>}	22
CANCEL_HOLD_ALL	{"CANCEL_HOLD_ALL":0}	22
CANCEL_HOLIDAY	{"CANCEL_HOLIDAY":0}	7
CLEAR_DEVICE_LIST	{“CLEAR_DEVICE_LIST”:<device>}	15
CLEAR_PROFILE	{"CLEAR_PROFILE":<profile name>}	19
CLEAR_PROFILE_ID	{“CLEAR_PROFILE_ID”:<number>}	19
CREATE_GROUP	{"CREATE_GROUP":[[<devices>], <name>]}	22
DELETE_GROUP	{"DELETE_GROUP":<group>}	22
DETACH_DEVICE	{“DETACH_DEVICE”:[<zone>,<device>]}	14
DEVICES_SN	{“DEVICES_SN”:0}	10
DST_OFF	{"DST_OFF":0}	13
DST_ON	{"DST_ON":0}	13
ENGINEERS_DATA	{"ENGINEERS_DATA":0}	39
FAN Speed	{"SET_FAN_SPEED":["HIGH",<device>]} {"SET_FAN_SPEED":["AUTO", <device>]} {"SET_FAN_SPEED":["MED",<device>]} {"SET_FAN_SPEED":["LOW", <device>]} {"SET_FAN_SPEED":["OFF", "Device1", " device2", "device3","deviceN"]}	33
FIRMWARE	{"FIRMWARE":0}	21
FROST_OFF	{"FROST_OFF":<device(s)>}	24
FROST_ON	{"FROST_ON":<device(s)>}	24
GET_DEVICE_LIST	{“GET_DEVICE_LIST”:<device>}	10
GET_DEVICES	{“GET_DEVICES”:0}	15
GET_ENGINEERS	{“GET_ENGINEERS”:0}	12
GET_GROUPS	{"GET_GROUPS":0}	22
GET_HOLD	{"GET_HOLD":0}	22
GET_HOLIDAY	{"GET_HOLIDAY":0}	7
GET_HOURSRUN	{"GET_HOURSRUN":<device(s)>}	23
GET_LIVE_DATA	{“GET_LIVE_DATA”:0}	8
GET_PROFILE	{"GET_PROFILE":<profile name>}	18
GET_PROFILE_0	{“GET_PROFILE_0”:<devices>}	19
GET_PROFILE_NAMES	{"GET_PROFILE_NAMES":0}	18
GET_PROFILE_TIMERS	{“GET_PROFILE_TIMERS”:0}	18
GET_PROFILES	{"GET_PROFILES":0}	18
GET_SYSTEM	{“GET_SYSTEM”:0}	10
GET_TEMPLOG	{"GET_TEMPLOG":<device(s)>}	25
GET_TIMER_0	{"GET_TIMER_0":<device>}	20

GET_ZONES	{"GET_ZONES":0}	10
GLOBAL_DEV_LIST	{“GLOBAL_DEV_LIST”:<devices>}	12
HOLD	{"HOLD":[{"temp":<number>, "id":<string>,"hours":<number>, "minutes":<number>}, <device(s)>]}	22
HOLIDAY	{"HOLIDAY":["HHMMSSDDMMYYYY","HHMMSSDDMMYYYY"]};	7
IDENTIFY	{"IDENTIFY":0}	22
IDENTIFY_DEVICE	{"IDENTIFY_DEV":<device>}	23
INFO	{"INFO":0}	23
LINK_DEVICE	{“LINK_DEVICE”:[<zone>,<device>]}	15
LOCK	{"LOCK":[[<pin1>,<pin2>,<pin3>,<pin4>], <device(s)>]}	23
MANUAL_DST	{"MANUAL_DST":<number>}	13
MANUAL_OFF	{“MANUAL_OFF”:<devices>}	26
MANUAL_ON	{“MANUAL_ON”:<devices>}	26
NTP_OFF	{"NTP_OFF":0}	13
NTP_ON	{"NTP_ON":0}	13
PERMIT_JOIN	{"PERMIT_JOIN":[<seconds>,<device>]}	14
PERMIT_JOIN	{"PERMIT_JOIN":["repeater", <seconds>]}	14
PROFILE_TITLE	{“PROFILE_TITLE”:[<oldname>,<newname>]}	19
READ_COMFORT_LEVELS	{"READ_COMFORT_LEVELS":<device(s)>}	12
READ_DCB	{"READ_DCB":100}	6
READ_TIMECLOCK	{"READ_TIMECLOCK":<device(s)>}	27
REMOVE_REPEATER	{"REMOVE_REPEATER":<repeater>}	14
REMOVE_ZONE	{"REMOVE_ZONE":<zone>}	14
RUN_PROFILE	{"RUN_PROFILE":<profile name>}	20
RUN_PROFILE_ID	{“RUN_PROFILE_ID”:[<profid>,<devices>]}	19
SET_CHANNEL	{"SET_CHANNEL":<channel>}	6
SET_CLOSE_DELAY	{"SET_CLOSE_DELAY":X}	15
SET_COMFORT_LEVELS	{"SET_COMFORT_LEVELS":[<levels>, <device(s)>]}	12
SET_COOL_TEMP	{"SET_COOL_TEMP":[<temp>, <device(s)>]};	32
SET_DATE	{"SET_DATE":[<year>, <month>, <day>]}	13
SET_DELAY	{"SET_DELAY":[<delay>, <device(s)>]}	21
SET_DIFF	{"SET_DIFF":[<switching differential>, <device(s)>]}	21
SET_FAILSAFE	{'SET_FAILSAFE':[true, 'neoair']} or {'SET_FAILSAFE':[false, 'neoair']}	21
SET_FLOOR	{"SET_FLOOR":[<temp>, <device(s)>]}	21
SET_FORMAT	{"SET_FORMAT":<format>}	6
SET_FROST	{"SET_FROST":[<temp>, <device(s)>]}	21
SET_LEVEL_4	{“SET_LEVEL_4”:0}	12
SET_LEVEL_6	{“SET_LEVEL_6”:0}	12
SET_OPEN_DELAY	{"SET_OPEN_DELAY":X}	15
SET_PREHEAT	{"SET_PREHEAT":[<temp>, <device(s)>]}	21
SET_RF_MODE	{"SET_RF_MODE":[<mode>, <devices>]}	16
SET_TEMP	{"SET_TEMP":[<temp>, <device(s)>]}	29
SET_TEMP_FORMAT	{"SET_TEMP_FORMAT":<format>}	6

SET_TIME	{"SET_TIME":[<hours>,<minutes>]}	13
SET_TIMECLOCK	{"SET_TIMECLOCK":[<levels>, <device(s)>]}	27
STATISTICS	{"STATISTICS":0}	25
STORE_PROFILE	{"STORE_PROFILE":<profile ob>}	18
STORE_PROFILE_0	{"STORE_PROFILE_0":[profile obj, devices]}	20
STORE_PROFILE_TIMER_0	{"STORE_PROFILE_TIMER_0":[profile obj, devices]}	20
SUMMER_OFF	{"SUMMER_OFF":<device(s)>}	24
SUMMER_ON	{"SUMMER_ON":<device(s)>}	24
TIME_ZONE	{"TIME_ZONE":<timezone offset>}	13
TIMER_HOLD_OFF	{"TIMER_HOLD_OFF":[<minutes>, <device(s)>]}	26
TIMER_HOLD_ON	{"TIMER_HOLD_ON":[<minutes>, <device(s)>]}	26
TIMER_OFF	{"TIMER_OFF":<device(s)>}	26
TIMER_ON	{"TIMER_ON":<device(s)>}	26
UNLOCK	{"UNLOCK":<device(s)>}	23
USER_LIMIT	{"USER_LIMIT":[<int>, <device(s)>]}	28
VIEW_ROC	{"VIEW_ROC":<device(s)>}	28
ZONE_TITLE	{"ZONE_TITLE":[<oldname>, <newname>]}	14

Appendix D

Examples of cached files.

Get System


```
{
"ALT_TIMER_FORMAT": 2,
"CORF": "C",
"DEVICE_ID": "NeoHub",
"DST_AUTO": false,
"DST_ON": false,
"FORMAT": 2,
"HEATING_LEVELS": 4,
"HEATORCOOL": "HeatOnly",
"HUB_VERSION": 2081,
"NTP_ON": " Running",
"PARTITION": "4",
"TIMESTAMP": 1518607836,
"TIME_ZONE": 0,
"UTC": 1519038792
}
```


Live Data


```
{
"CLOSE_DELAY": 0,
"COOL_INPUT": false,
"GLOBAL_SYSTEM_TYPE": "HeatOnly",
"HOLIDAY_END": 0,
"HUB_AWAY": false,
"HUB_HOLIDAY": false,
"HUB_TIME": 1519038584,
"OPEN_DELAY": 0,
"TIMESTAMP_DEVICE_LISTS": 1519038219,
"TIMESTAMP_ENGINEERS": 1519038219,
"TIMESTAMP_PROFILE_0": 1519038270,
"TIMESTAMP_PROFILE_COMFORT_LEVELS": 1518604883,
"TIMESTAMP_PROFILE_TIMERS": 1518600089,
"TIMESTAMP_PROFILE_TIMERS_0": 1519038219,
"TIMESTAMP_SYSTEM": 1518607836,
"devices": [
{
"ACTIVE_LEVEL": 2,
"ACTIVE_PROFILE": 28,
"ACTUAL_TEMP": "24.5",
"AVAILABLE_MODES": [
"heat"
],
"AWAY": false,
"COOL_MODE": false,
"COOL_ON": false,
"COOL_TEMP": 0,
"CURRENT_FLOOR_TEMPERATURE": 127,
"DATE": "monday",
"DEVICE_ID": 1,
"FAN_CONTROL": "Automatic",
"FAN_SPEED": "Custom",
"FLOOR_LIMIT": false,
"HC_MODE": "VENT",
```

"HEAT_MODE": true,
"HEAT_ON": false,
"HOLD_OFF": true,
"HOLD_ON": false,
"HOLD_TEMP": 30,
"HOLD_TIME": "0:00",
"HOLIDAY": false,
"LOCK": false,
"LOW_BATTERY": false,
"MANUAL_OFF": true,
"MODELOCK": false,
"MODULATION_LEVEL": 0,
"OFFLINE": false,
"PIN_NUMBER": "1111",
"PREHEAT_ACTIVE": false,
"RECENT_TEMPS": [
"18.4", up to 96 temperature readings.

"24.2"
],
"SET_TEMP": "17.0",
"STANDBY": false,
"SWITCH_DELAY_LEFT": "0:00",
"TEMPORARY_SET_FLAG": false,
"THERMOSTAT": true,
"TIME": "11:08",
"TIMER_ON": false,
"WINDOW_OPEN": false,
"WRITE_COUNT": 115,
"ZONE_NAME": "Bathroom"
},
{
"ACTIVE_LEVEL": 0,
"ACTIVE_PROFILE": 0,
"ACTUAL_TEMP": "23.6",
"AVAILABLE_MODES": [
"heat"
],
"AWAY": false,
"COOL_MODE": false,
"COOL_ON": false,
"COOL_TEMP": 23,
"CURRENT_FLOOR_TEMPERATURE": 127,
"DATE": "monday",
"DEVICE_ID": 3,
"FAN_CONTROL": "Manual",
"FAN_SPEED": "Off",
"FLOOR_LIMIT": false,
"HC_MODE": "HEATING",
"HEAT_MODE": true,
"HEAT_ON": false,
"HOLD_OFF": true,
"HOLD_ON": false,
"HOLD_TEMP": 0,
"HOLD_TIME": "0:00",

"HOLIDAY": false,
"LOCK": false,
"LOW_BATTERY": false,
"MANUAL_OFF": true,
"MODELOCK": false,
"MODULATION_LEVEL": 0,
"OFFLINE": false,
"PIN_NUMBER": "0000",
"PREHEAT_ACTIVE": false,
"RECENT_TEMPS": [
"18.0",   up to 96 temperature readings.
"23.5"
],
"SET_TEMP": "16.0",
"STANDBY": false,
"SWITCH_DELAY_LEFT": "0:00",
"TEMPORARY_SET_FLAG": false,
"TIME": "11:09",
"TIMECLOCK": true,
"TIMER_ON": false,
"WINDOW_OPEN": false,
"WRITE_COUNT": 70,
"ZONE_NAME": "Office"
}]}

Get Engineers

{
"Bathroom": {
"DEADBAND": 0,
"DEVICE_ID": 1,
"DEVICE_TYPE": 1,
"FLOOR_LIMIT": 28,
"FROST_TEMP": 12,
"MAX_PREHEAT": 3,
"OUTPUT_DELAY": 0,
"PUMP_DELAY": 0,
"RF_SENSOR_MODE": "self",
"STAT_FAILSAFE": 0,
"STAT_VERSION": 101,
"SWITCHING DIFFERENTIAL": 1,
"SWITCH_DELAY": 0,
"SYSTEM_TYPE": 0,
"TIMESTAMP": 1519038219,
"USER_LIMIT": 0,
"WINDOW_SWITCH_OPEN": false
},
"Office": {
"DEADBAND": 2,
"DEVICE_ID": 3,
"DEVICE_TYPE": 12,

"FLOOR_LIMIT": 28,
"FROST_TEMP": 12,
"MAX_PREHEAT": 0,
"OUTPUT_DELAY": 0,
"PUMP_DELAY": 0,
"RF_SENSOR_MODE": "self",
"STAT_FAILSAFE": 0,
"STAT_VERSION": 12,
"SWITCHING DIFFERENTIAL": 1,
"SWITCH_DELAY": 0,
"SYSTEM_TYPE": 0,
"TIMESTAMP": 1519038219,
"USER_LIMIT": 0,
"WINDOW_SWITCH_OPEN": false
}}

Get Profiles

{"ball": {"PROFILE_ID": 28,"group": null,"info": {"monday": {"leave": ["09:30",17],"return": ["17:30",25],"sleep": ["22:30",15],"wake": ["01:00",14]},"sunday": {"leave": ["09:30",17],"return": ["17:30",21],"sleep": ["22:30",15],"wake": ["01:00",21]}},"name": "ball"}}.

Get Profile Timers

{"Timer": {"PROFILE_ID": 26,"group": null,"info": {"monday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","24:00"],"time4": ["24:00","24:00"]},"sunday": {"time1": ["07:00","09:00"],"time2": ["16:00","20:00"],"time3": ["24:00","24:00"],"time4": ["24:00","24:00"]}},"name": "Timer"}}

Appendix E

Neo System device type list.

DEVICE TYPE ID NUMBERS
TCM = 01
Wi-Fi STAT = SMARTSTAT = 02
COOLSWITCH = 03
TCM-RH = 04
WDS = 05
NEOPLUG = 06
NEOAIR = 07
Smart Stat HC = 08
NeoAir HW = 09 (combined model)
REPEATER = 10
NEOSTAT - HC =11
Neostat-V2 = 12
Neoair V2 =13
Remote Air sensor =14
NeoAir-V2 combined mode = 15
RF Switch Wifi = 16
Edge wifi thermostat = 17

Note. The neostat and neostat V2 have different device id's this is because they use different hardware and firmware and the neohub needs to know the difference when doing firmware updates. The same applies to NeoAir V1 and NeoAir v2.
The JSon commands are unaffected by the differences.

Addendum

Additional settings

Constant Fan,
Used to keep fans running when there is no call for heat or cooling, often used it large buildings to maintain a constant air flow.
Setting it true or false 1 or zero and corresponds to feature setting number 12 in neostat HC.
Location = engineers cache

Cool Proof

Used to delay the start of the fans to give the water in the pipes time to warm up.
Settings are 00 to 95 seconds in 5 seconds steps.
Location = engineers cache

"HUB_TYPE": 2,
A new variable used to denote the model of neohub being used

Type 1 original neohub generation 1
Type 2 neohub generation 2 with home kit
Type 3 Pending