Heatmiser Neo Driver

Revision: 5.01
Date: 15 January 2026

This documentation contains information on the HeatmiserNeo SIMPL module. This module is compatible with the NeoStat, NeoPlug and the NeoStatHC. This module is only compatible with the V2 series of the NeoHub devices. This module has been written for the 3=series and 4 series Crestron control systems ONLY.

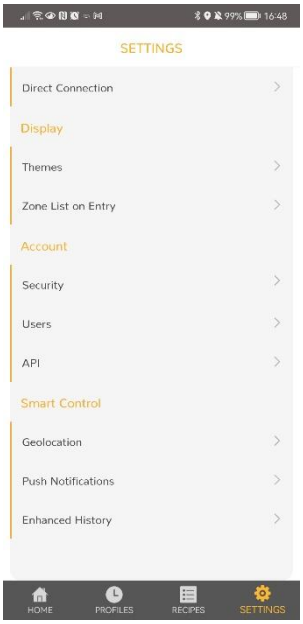
Installation Notes

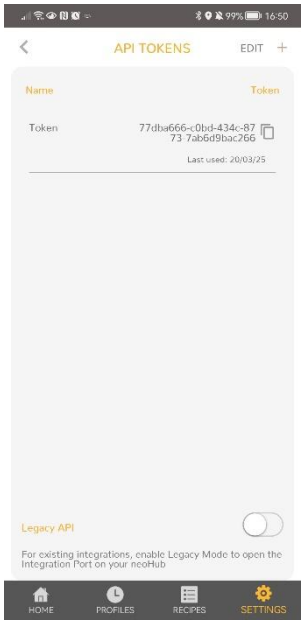
The Crestron system communicates to a Heatmiser installation directly via an IP connection to the Heatmiser Hub.

Equipment Setup

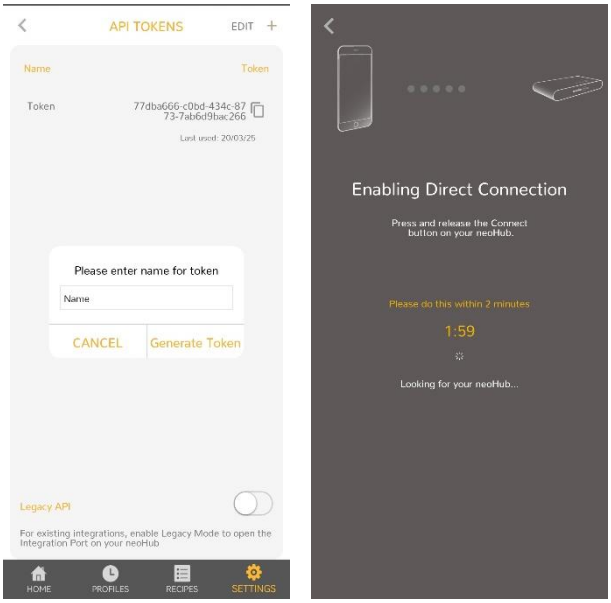
Connect the Heatmiser Neo equipment and setup the system using the Heatmiser Neo app as per Heatmiser's installation instructions: https://www.heatmiser.com/heatmiser-neo-overview/

In order to use this module, you must generate an API key on the Heatmiser App and the Legacy API must be disabled.





To generate the API key in the app, go to the "Settings" tab, then scroll down and click on the "API" tab. At the bottom of the screen the "Legacy API" must be toggled off as shown above.



To create the API key click on the + button at the top right of this screen and follow the instructions that show, you must be physically near the Neo Hub to pair as you need to press a button on the hub during the pairing process.

DHCP Reservation Requirements

There is currently no way to set a static IP address for a Heatmiser Hub.

To ensure communications with the Crestron module the Heatmiser Neo Hub must have a DHCP reservation set in your router or DHCP server.

This solution is compatible with multiple Heatmiser Hubs in a single installation.

Programming Notes

Each of the module files should be placed either in the host program's project folder or, to make the Heatmiser interface available to all Crestron programs, in the SIMPL Windows installation's User Macro (for .umc files) and User SIMPL+ (for .usp, .ush and .clz files) directories. This PDF should be placed in both directories for SIMPL's F1 help function to work properly.

The module is broken into two parts:

- A hub module that handles all communications between the Crestron processor and a single Heatmiser Hub. The program may contain multiple hubs, but each hub and its associated devices can be considered as a single logical unit.
- One or more device modules that present the control and feedback signals to the host program. It is normal to have one instance for each physical thermostat or plug.

The Hub Module

The hub module handles all the IP communications with the Heatmiser Hub. As previously noted, there can be more than 1 hub in a program.

1. Set the IP Address of the Hub (that is reserved by your router following the DHCP reservation step) into the IpAddress parameter.
2. If you require an offline key (as mentioned in the Licencing section), enter this into the OptionalOfflineKey parameter. If not leave default.

3. Enter the API key that you generated previously in the setup stage into the ApiKey parameter.
4. Enter the identifier that you want to use to name this hub into the HubId parameter. This identifier should then be placed into all the NeoStat and NeoPlug modules that are connected to this hub.
5. The PollFrequency parameter set the rate that the data is updated and displayed. Set this to the time, in seconds, that you require.

Inputs and Outputs

Inputs

Signal Name	Description
InitialiseHub	Upon rising this will initiate the hub, connecting to the API and making it available for all the connected NeoStats and NeoPlugs to see. This should only be triggered successfully once. Should you fail to connect you can make this signal rise again to reattempt the connection.
QueryData	Upon rising this will update the data displayed should you want to update the data before the poll occurs.
AwayOn	Upon rising this will set the Heatmiser system connected to the hub to away. Away mode will pause the scheduler and put all the devices into frost-protect mode.
AwayOff	Upon rising this will set the Heatmiser system connected to the hub to no longer be away.
SetCelcius	Upon rising this will set the system to use Degrees Celsius. Changing this will wipe the recent temperatures until enough time passes for new data to be recorded.
SetFahrenheit	Upon rising this will set the system to use Degrees Fahrenheit. Likewise changing will wipe the recent temperatures.
CancelHoliday	Upon rising this will cancel the system wide holiday.

StartHoliday	Upon rising this will take the values set further down and start the holiday and end the holiday on the specified dates.
StartHolidayForDays	Upon rising this will take the value in HolidayForDays# and start a holiday immediately. It will end at the same time you started on the day specified.
HolidayForDays#	The number of days you wish to put the system on holiday for.
HolidayStartDay#	The day you wish to start the holiday.
HolidayStartMonth#	The month you wish to start the holiday.
HolidayStartYear#	The year you wish to start the holiday.
HolidayStartHour#	The hour you wish to start the holiday.
HolidayStartMinute#	The minute you wish to start the holiday.
HolidayEndDay#	The day you wish to end the holiday.
HolidayEndMonth#	The month you wish to end the holiday.
HolidayEndYear#	The year you wish to end the holiday.
HolidayEndHour#	The hour you wish to end the holiday.
HolidayEndMinute#	The minute you wish to end the holiday.

Outputs

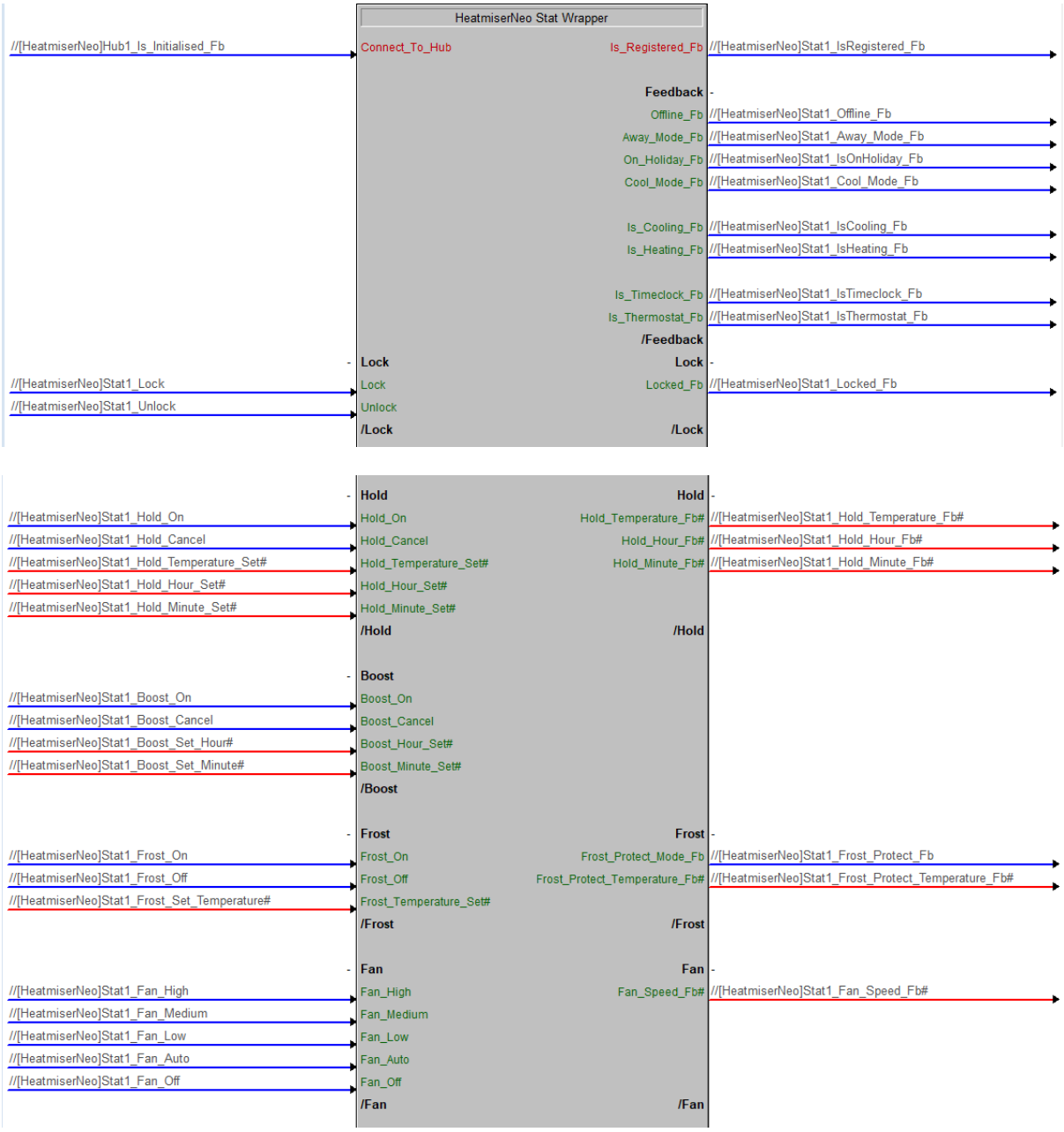
Signal Name	Description
Is_Initialised_Fb	The signal will be high upon completing the initialisation of the hub. The hub is considered initialised once a connection is made to the hub API successfully. This signal can then be used to trigger the Connect_To_Hub signal in related devices.
IsLicenced_Fb	The signal will be high if you have a licence active, even a trial licence.
IsInTrial_Fb	The signal will be high if the trial is active, this can be used with IsLicenced_Fb to check your licence status.

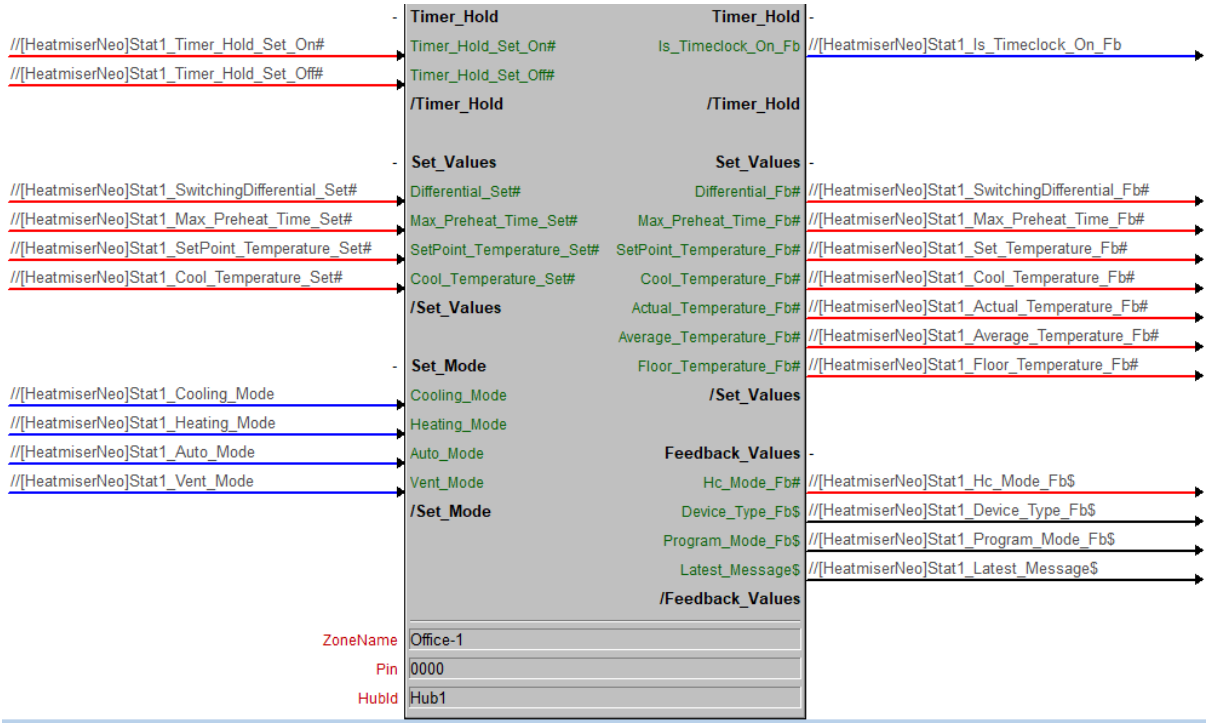
ApiConnection_Fb	The signal will be high whilst the program can communicate with the NeoHub. If this signal goes low, check your NeoHub.
Away_Fb	The signal is high if the system is away, low otherwise
Holiday_Fb	The signal is high if the system is on holiday, low otherwise
HolidayEndDay_Fb#	The day the holiday is set to end, if on holiday.
HolidayEndMonth_Fb#	The month the holiday is set to end, if on holiday
HolidayEndYear_Fb#	The year the holiday is set to end, if on holiday
HolidayEndHour_Fb#	The hour the holiday is set to end, if on holiday
HolidayEndMinute_Fb#	The minute the holiday is set to end, if on holiday
HolidayEnd$	A string containing the end date and time in the format: “Sun Mar 23 09:19:24 2025”
TemperatureFormat$	Displays the temperature units the system uses: “F” for Fahrenheit, “C” for Celsius.
LatestMessage$	Displays the latest message about the commands being sent to the hub.

The NeoStat Module

This module talks internally to the Hub module and controls and monitors a single Heatmiser NeoStat device. It is compatible with NeoStats in both Thermostat and Timeclock mode as well as NeoStatHC devices.

NOTE: Temperatures should be entered and will be displayed as 10 times what they should be. For example, the temperature 21.5 C should be entered and will be displayed as 215.





1. Enter the name of the device, exactly as it is in the HeatmiserNeo app into the ZoneName parameter. Note: This IS case sensitive
2. Set the Pin code to be between 0000 and 9999.
3. Set the HubId to be the HubId you entered into the Hub that this device is connected to in the Heatmiser Neo app.

Inputs and Outputs

Inputs

Signal Name	Description
Connect_To_Hub	Upon rising this will attempt to connect to the Hub you have specified. If this succeeds the Is_Registered_Fb will go high.
Lock	Upon rising this will use the pin code entered and lock the device.
Unlock	Upon rising this will unlock the device.
Hold_On	Upon rising this will start holding the temperature for the time specified below.
Hold_Cancel	Upon rising this will cancel any holds set on this device.

Hold_Temperature_Set#	This takes the temperature you wish to set for the duration of the hold. If the device is in Timeclock mode, this will only be 1 or 0.
Hold_Hour_Set#	This takes the hours you wish for the hold to last.
Hold_Minute_Set#	This takes the minutes you wish for the hold to last.
Boost_On	Upon rising this will start the boost for the time set below.
Boost_Cancel	This will cancel any boosts on this device.
Boost_Set_Hour#	The hours you wish the boost to be active for.
Boost_Set_Minute#	The minutes you wish the boost to be active for.
Frost_On	Upon rising the device will enter Frost Protect mode.
Frost_Off	Upon rising the device will leave Frost Protect mode.
Frost_Temperature_Set#	Set the frost protect temperature.
Fan_High	Upon rising the fan speed will be set to high. Only applicable to NeoStat HC devices.
Fan_Medium	Upon rising the fan speed will be set to medium. Only applicable to NeoStat HC devices.
Fan_Low	Upon rising the fan speed will be set to low. Only applicable to NeoStat HC devices.
Fan_Auto	Upon rising the fan speed will be set to auto. Only applicable to NeoStat HC devices.
Fan_Off	Upon rising the fan will be turned off. Only applicable to NeoStat HC devices.
Timer_Set_Hold_On#	Takes the time you wish for the timer to be on, in minutes. Then sets the timer to

	be on for that time. This should be used for devices with hot water capabilities.
Timer_Set_Hold_Off#	Takes the time you wish for the timer to be off, in minutes. Then sets the timer to be off for that time.
Differential_Set#	Sets the temperature difference that is allowed before the heating/cooling will activate. For a differential of 2, if the set temperature was 21 and the temperature fell to 19 the heating would activate. Can be set to 1, 2 or 3.
Max_Preheat_Time_Set#	Sets the maximum time in hours the system will preheat for. Can be set between 1 and 5 hours.
SetPoint_Temperature_Set#	Temporarily overrides the temperature set in the profile until the next comfort level is reached.
Cool_Temperature_Set#	Sets the cooling temperature.
Cooling_Mode	Upon rising the thermostat will be set to cooling mode. Only applicable to NeoStat HC devices.
Heating_Mode	Upon rising the thermostat will be set to heating mode. Only applicable to NeoStat HC devices.
Auto_Mode	Upon rising the thermostat will be set to auto mode. Only applicable to NeoStat HC devices.
Vent_Mode	Upon rising the thermostat will be set to vent mode. Only applicable to NeoStat HC devices.

Outputs

Signal Name	Description
Is_Registered_Fb	The signal will be high if the device has successfully connected to a hub.

Offline_Fb	The signal will be high if the device is offline. This means the hub contains this zone but cannot connect to it so check the power for the device.
Away_Mode_Fb	The signal will be high if the system is away.
On_Holiday_Fb	The signal will be high if the system is on holiday.
Cool_Mode_Fb	The signal will be high if the device is set to be in cooling mode.
Is_Cooling_Fb	The signal will be high if the device is cooling. Only applicable to NeoStat HC devices.
Is_Heating_Fb	The signal will be high if the device is heating. Only applicable to NeoStat HC devices.
Is_Timeclock_Fb	The signal will be high if the NeoStat is in timeclock mode.
Is_Thermostat_Fb	The signal will be high if the NeoStat is in thermostat mode.
Locked_Fb	The signal will be high if the device is locked.
Hold_Temperature_Fb#	The temperature of the hold if the device is in a hold.
Hold_Hour_Fb#	The hours remaining in the hold.
Hold_Minute_Fb#	The minutes remaining in the hold.
Frost_Protect_Mode_Fb	The signal will be high if the device is in Frost Protect mode.
Frost_Protect_Temperature_Fb	The frost protect temperature of the device.
Fan_Speed_Fb#	An integer representing the current speed of the fan as shown in the Output Codes section. Only applicable to NeoStat HC devices.

Is_Timeclock_On_Fb	The signal will be high if the device is heating when in Timeclock mode.
Differential_Fb#	The differential of the device.
Max_Preheat_Time_Fb#	The current maximum time for the system to preheat to the next comfort level.
SetPoint_Temperature_Fb#	The current setpoint temperature.
Cool_Temperature_Fb#	The cooling temperature.
Actual_Temperature_Fb#	The current temperature the device is reading.
Average_Temperature_Fb#	The average temperature for the last 24 hours that the device was on.
Floor_Temperature_Fb#	The current floor temperature should that be configured.
Hc_Mode_Fb#	An integer that represents the HC(heating and cooling) mode is set too as shown in the Output Codes section.
Device_Type_Fb$	A string containing what kind of Heatmiser Device this is.
Program_Mode_Fb$	A string containing what profile type the device is set to.
Latest_Message$	A string containing the most recent message relating to the commands being sent.

Output Codes

Hc_Mode_Fb# Signal

Integer Code	Value
0	Error
1	Vent
2	Auto
3	Cooling

4	Heating

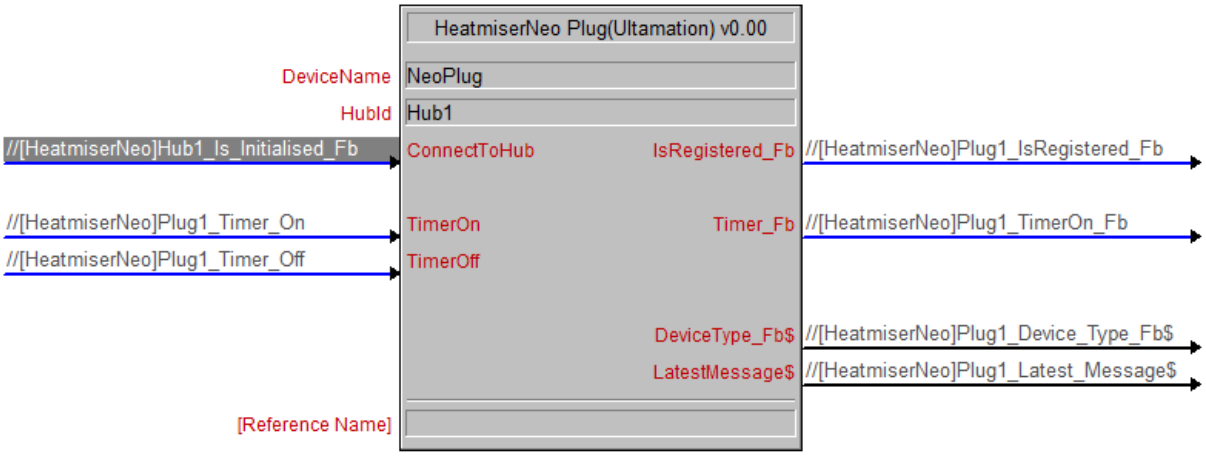
Fan_Speed_Fb# Signal

Integer Code	Value
0	Error
1	Off
2	Auto
3	Low
4	Medium
5	High

The NeoPlug Module

This module talks internally to the Hub module and controls and monitors a single Heatmiser NeoPlug device.

There should be one device module per physical Heatmiser NeoPlug device.



1. Enter the DeviceName exactly as it is set in the Heatmiser Neo app. Note: This IS case sensitive
2. Set the HubId to be the HubId you entered into the Hub that this device is connected to in the Heatmiser Neo app.

Inputs and Outputs

Inputs

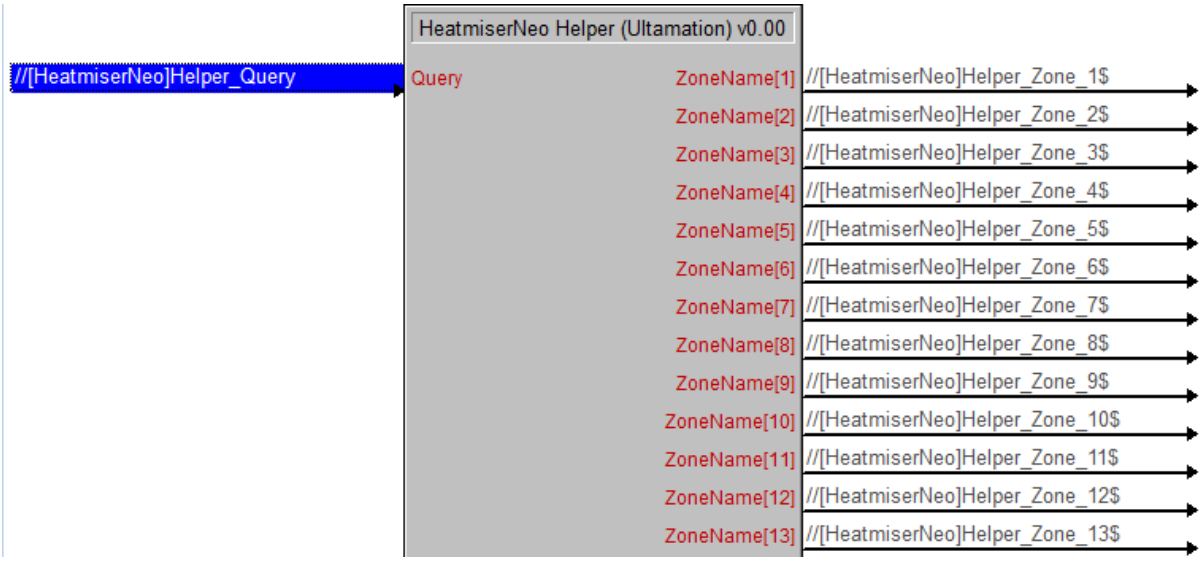
Signal Name	Description
ConnectToHub	Upon rising this will attempt to connect to the Hub you specified in the HubId parameter.
TimerOn	Upon rising this will turn the device timer on.
TimerOff	Upon rising this will turn the device timer off.

Outputs

Signal Name	Description
IsRegstered_Fb	The signal will be high if the device has successfully connected to a hub.
Timer_Fb	The signal will be high if the device's timer is on.
DeviceType_Fb$	A string containing what kind of Heatmiser Device this is.
LatestMessage$	A string containing the most recent message relating to the commands being sent.

The Helper Module

This module has no functional value whatsoever but is included to aid debugging of issues with the Neo configuration. Include the module in the same slot as the Heatmiser Hubs and when you trigger the Query signal, the module will report each device on the outputs for all hubs that have been initialised in this program slot. This helper module can display up to 50 connected devices.



Inputs and Outputs

Signal Name	Description
Query	After intialising all hubs, trigger this signal to populate the outputs with all the NeoStat and NeoPlug devices that have been discovered by the Hubs. The information will display the HubId that the device is connected to and the ZoneId of that device. Zones will only appear if a hub has been connected to the system.
ZoneName[x]	Each zone will be displayed with the parents HubId and the ZoneId for the device. The ZoneId should be used as the ZoneName's in the NeoStat and NeoPlug modules.

FAQ

1. *My Heatmiser hub isn't showing anything. What do I check?*
    a. Check the Initialised_Fb signal is high
        i. If it is check the ApiConnection_Fb signal is high
            1. If this is low that would suggest the hub cannot connect to the API.
            2. Check the NeoHub is turned on
            3. Check the NeoHub is connected to the network
            4. Check the Heatmiser app to see if your API key is there, if not try rebooting the NeoHub
        ii. If this is low, the program has not made a connection to the Hub's API successfully yet
            1. Try rising the InitialiseHub signal, the logs will then display any issues if that does not work
            2. Check that the IP address is correct
            3. Check the API key is correct
2. *How do I check if the driver is licenced?*
    a. In the program:
        i. Check the Licenced_Fb is high and the IsInTrial_Fb is low
        ii. If the IsInTrial_Fb is high, you are in trial mode and do not have a valid licence active so the module will only work for one hour
    b. For cloud licences, you can check the licence status using the following link https://portal.ultamation.com/licensing
    c. Alternatively, you can check the status of the licence by reviewing the error log after a reboot.
        i. If the driver is licenced you will see one of the following
            1. "CloudStatus: Cloud licence status is Valid"
            2. "ApplyOfflineKey: Offline key applied successfully!"
        ii. If the driver is unlicenced, you will see one of the following
            1. "CloudStatus: Cloud licence status is LicenceInvalid"
            2. "CloudStatus: Cloud licence status is LicenceRevoked"
            3. ""ApplyOfflineKey : Invalid offline key entered [<*LICENCE_KEY*>] for MAC Address : <*MAC_ADDRESS*>"
3. *Can I test the driver before purchasing a licence?*
    a. Yes, the driver will operate for 1 hour in trial mode. You can restart the trial by rebooting the Crestron processor. We recommend using this trial period to ensure you are satisfied with the driver before purchasing a licence.
4. *I have had to replace my Crestron Processor; how do I transfer my licence to the new processor?*
    a. Please send an e-mail to support@ultamation.com, with the following information. We will look into the request and depending on the circumstances, may proceed with transferring the licence to the new processor.
        i. Old MAC Address
        ii. New MAC Address

iii. Order Number (This can be found in the e-mail you received, when you purchased the licence)

Support

If you have any issues with a driver or installation please let us know by contacting Ultamation support on support@ultamation.com and please include as much detail about your issue as possible, such a recent processor error log.

Licence verification messages are posted to the error log, so please ensure you have checked this.

Licencing

This driver (including software, images and all other associated assets distributed as part of the purchased download package) is licenced on a PER PROCESSOR basis.

A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

This is an electronic product and there is no physical delivery.

The driver is provided without any warranty with respect to the reliability of the controlled device or changes to device protocol. We will endeavour, through best efforts, to maintain the driver's functionality and any bug fixes will be provided free-of-charge. Additional functionality may be released as a variation of this driver and this will be a separate, purchasable, product.

Cloud Licence

This driver contacts Ultamation's licencing server at startup. If the server finds a matching licence for the driver and processor then the driver will be licenced. Otherwise, the driver will check the offline licence key. If you purchase a licence AFTER you have loaded the driver, please reboot the system to see changes take effect.

If you purchased a licence **before** it was migrated to the cloud service, i.e. you have a licence key already, you must enter this into the **Offline Key** user attribute. If you purchased the driver **after** it was migrated, and you don't have a licence key, no further action is required.

If no licence exists for the product/processor the driver will enter a short trial period (ONE HOUR) to allow for verification of correct control or evaluation.

To request an OFFLINE key, please contact support@ultamation.com with your order details and a brief explanation why you REQUIRE offline activation. Ultamation reserve the right to refuse offline activation.

NOTE: Once an offline key has been issued no further licence changes will be granted. Moving the product to a new processor will require an additional licence purchase.