

CRESTRON INTEGRATION MODULE



Version: 4.00
Date: 07/08/2020

INTRODUCTION

This datasheet relates to Ultamation's latest version of our Heatmiser Neo integration modules for Crestron control systems. This module is compatible with both the NeoStat and the NeoPlug. It provides the essential information for integration between the Heatmiser system and the Crestron control processor, and for programming of the module with a host Crestron program.

This new version has been re-written for compatibility with Crestron's 4-Series processors and we have added an improved upon the previous modules. Please note that the licencing for the module has also be updated to a PER PROCESSOR model.

Please Note

This module has been written for 3-series and 4-series Crestron control systems ONLY and is NOT compatible with the 2-series range. If you wish to use Heatmiser products with a Crestron 2-series processor we recommend the WiFi or wired (RS485) solution, combined with our appropriate control module for that technology. There is currently no control for the Heatmiser Neo Stat/Plug on a 2-series processor.

INSTALLATION NOTES

The Crestron system communicates to 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. <http://neo.heatmiser.com/>

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.

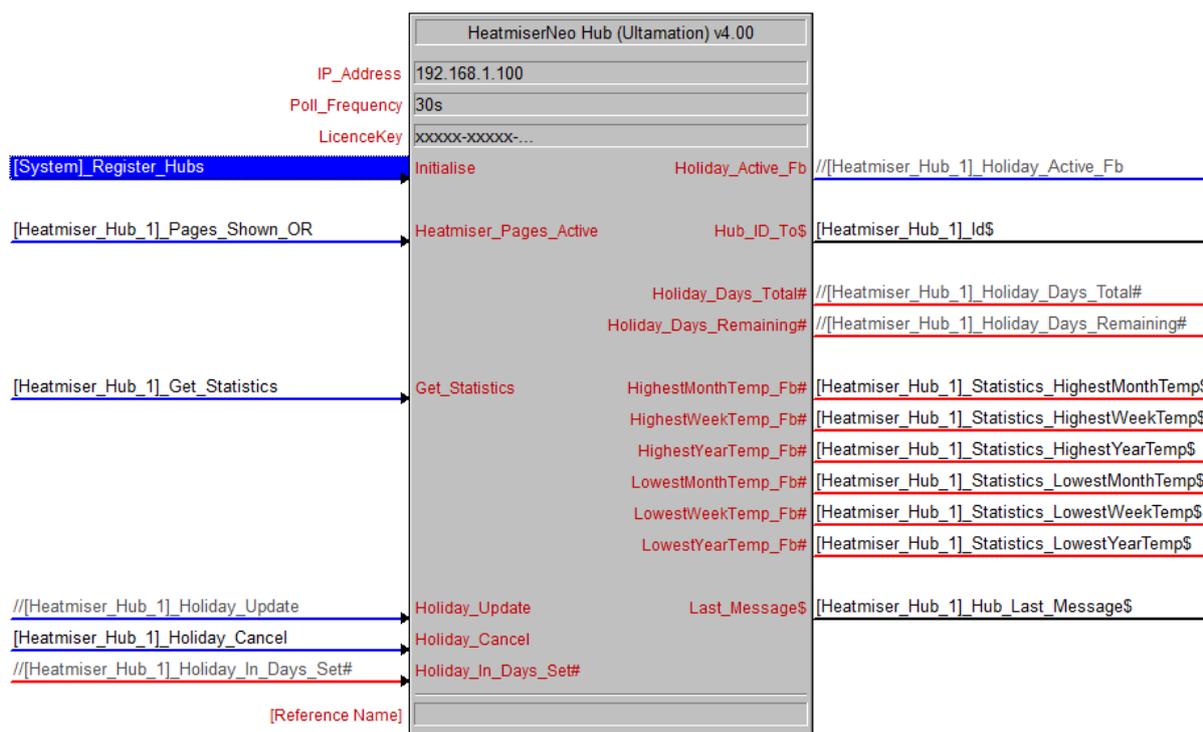
The module is broken into two parts:

- ✿ 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

This module handles all of the IP communications to 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 IP_Address parameter.
2. Set the Poll_Interval parameter to a value between 10 seconds and 180 seconds. This is flexible to allow for site specific requirements. This value is used to control the polling frequency when the "Heatmiser_Pages_Active" signal goes high.
3. Enter the Licence Key for the specific Crestron Processor into the "LicenceKey" parameter. This will have been emailed to you when you purchase the module.

4. You should not generally put a '1' on the Heatmiser_Pages_Active. This will normally be high only when Heatmiser Neo device pages are shown on a UI. This reduces the load on the processor when more rapid heating information updates are required.

Initialise	Rising edge. Queries the hub for attached stats, and then communicates the hub IP to the device modules. This should only be triggered once, upon start-up.
Hub_ID_To\$	ID used to connect associated devices to the hub.
Heatmiser_Pages_Active	ONLY KEEP THIS HIGH WHEN HEATMISER UPDTAES ARE REQUIRED. This instructs the Hub to query all physical devices on the period defined in Poll_Frequency and to update feedback information on device modules.
Get_Statistics	Rising Edge. Gets the Energy Monitoring statistics from the Heatmiser Neo system.
Highest*#,Lowest*#	Updated when Get_Statistics is triggered. Shows the highest and lowest system wide energy monitoring values.
Holiday_Update	Rising Edge. Polls the hub for Holiday state information. This would normally be polled once a day when in holiday mode.
Holiday_Cancel	Rising Edge. Cancels any system wide holidays.
Holiday_In_Days#	Analog. Sets a system wide Holiday on change.
Holiday_Active_Fb	When high, indicates that the hub is in Holiday mode.
Holiday_Days_Total#	Updated on poll: The total number of days in the current Holiday period.
Holiday_Days_Remaining#	Updated on poll: The number of whole days remaining in the current Holiday period.
Last_Message\$	For Debug Purposes. Updated with various messages during normal operation.

The NeoStat Module

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

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



1. Enter the Zone_Name exactly as it is set in the Heatmiser Neo app.
Note: This IS case sensitive
2. Set the Lock_Code to be a value between 0000 and 9999 to be used if the Crestron system is required to lock the devices.

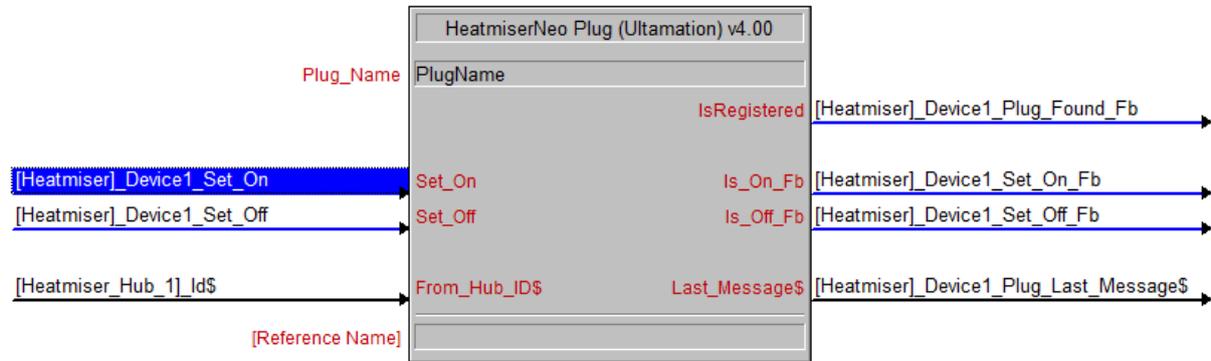
IsRegistered	When NeoStat device is registered this signal will go high.
Zone_Name	Parameter. Set this to the same value as set in the Heatmiser Neo App for the device you want to control
Lock_Code	Parameter. Set this to a value between 0 and 9999 to reflect the lock code you wish the Crestron system to lock the device with.
Away_*	Rising Edge. Sets the Away state of the system. This pauses the Heatmiser scheduler, and will put all of the stats into frost-protect mode.
Away_Fb	Real time feedback of the Away state of the device.
FrostProtect_*	Rising Edge. Sets the Frost Protection state of the device. This is equivalent to "off" for a single stat.
FrostProtect_Fb	Real time feedback of the Frost Protection state of the device.
Lock	Rising Edge. Locks the device with the lock code specified in the Lock_Code parameter.
Unlock	Rising Edge. Unlocks the device.
Locked_Fb	Real time feedback of the lock state of the device.
Hold	Rising Edge. Starts a hold period with the values set in Hold_Temp#, Hold_Hours# and Hold_Mins#
Hold_Cancel	Rising Edge. Cancels any in progress Hold. Has no effect if there is no Hold in progress.
Hold_Fb	Real time feedback of the hold state of the device.
Offline_Fb	This will show if the Heatmiser Neo system is aware of the device but it hasn't responded in a while. This can be due to comms or power failure.
IsHeating_Fb	Real time feedback of the Heating state of the device
IsCooling_Fb	For future use. Real time feedback of the Cooling state of the device.
InHoliday_Fb	Real time feedback of the Holiday state of the device
Is_Thermostat_Fb, Is_Timeclock_Fb	Real time feedback of the device type.
Timer_Hold_On#	For stats configured as timers, or stats with Hot Water functionality, set this to a value to turn on the timer/hot water for a given number of minutes. Set to 0 to switch off.
Timer_Hold_Off#	This is the inverse of the above signal and allows a

	timer/hot water mode that is currently active to be interrupted for the specified number of minutes.
Setpoint_Temp#	Sets the current set point of the device.
Setpoint_Temp_Fb#	Real time feedback of the current set point of the device.
Frost_Temp#	Set the frost protection set point of the device
Frost_Temp_Fb#	Real time feedback of the current frost protection set point of the device.
Current_Temp_Fb#	Real time feedback of the current temperature of the device.
Current_Floor_Temp_Fb#	Real time feedback of the current floor temperature of the device.
Hold_Temp#	Sets the Hold temperate set point to be used when the Hold signal is triggered.
Hold_Hours#, Hold_Mins#	Sets the Hold time to be used when the Hold signal is triggered.
Hold_Hours_Fb#, Hold_Mins_Fb#	Real time feedback of the current Hold time. This may be used as a countdown time feedback.
Program_Mode_Fb\$	Real time feedback of the program schedule mode of the device.
SensorType_Fb\$	Real time feedback of the attached sensor type of the device.
DeviceType_Fb\$	An identifier of the device type of stat being reported.
From_Hub_ID\$	Connect this signal to the Hub_ID_To\$ signal of the associated hub.
Last_Message\$	For Debug Purposes. Updated with various messages during normal operation.

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.

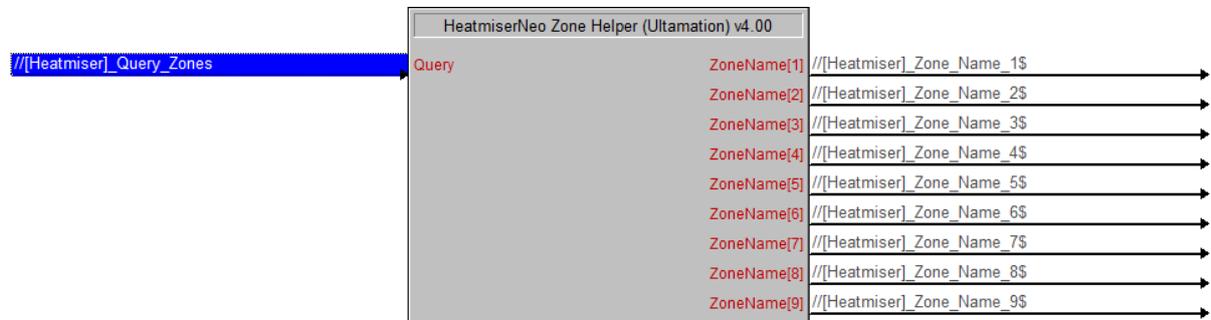


1. Enter the Plug_Name exactly as it is set in the Heatmiser Neo app.
Note: This IS case sensitive

Plug_Name	Parameter. Set this to the same value as set in the Heatmiser Neo App for the device you want to control
IsRegistered	When NeoPlug device is registered this signal will go high.
Set_On	Rising Edge. Sets the On state of the device.
Is_On_Fb	Real time feedback of the On state of the device.
Set_Off	Rising Edge. Sets the Off state of the device.
Is_Off_Fb	Real time feedback of the Off state of the device.
From_Hub_ID\$	Connect this signal to the Hub_ID_To\$ signal of the associated hub.
Last_Message\$	For Debug Purposes. Updated with various messages during normal operation.

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 in this program slot.



<p>Query</p>	<p>After Hub initialisation, trigger this signal to populate the outputs with all of the stat zones that have been discovered in the Heatmiser Neo system.</p> <p>The information will detail both the hub IP address and the case sensitive zone name.</p> <p>A hub must be reachable for the zones to appear in the list.</p>
<p>ZoneName[n]</p>	<p>Each stat zone will be listed by name, along with its parent's hub IP Address. This can then be checked against the Zone_Name parameters in the Stat modules.</p>

Licence

This module (including software, images and any and all other associated assets distributed as part of the purchased download package) is licenced PER Processor.

A licence key is generated at the point of purchase and is linked at that time to specific information that MUST be provided at the time of purchase. A purchase should not be completed without correct information as refunds cannot be issued for errors or changes made to details following purchase.

The licence key for each processor will be delivered via email along with links to download the module. There is no physical delivery.

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