Newflow

NÅNO

Visual C||CURE Comms Links

Newflow Ltd
George House
Derwent Road
Malton
North Yorkshire
YO17 6YB
UK

Tel: +44 1653 310 000
Email: sales@newflow.co.uk

Document Information

Document Name: NÅNO Visual C‖Cure Comms Links
Document Identifier   VCCL

Rev	Date	Changes	Prepared	Checked	Authorised
1	23/09/21	Initial release	MPFJ	GPL	MPFJ
2	27 Apr 2023	Updated OPC UA Added OPC UA Client Update datamap import Add "single shot" mode Minor formatting changes	MPFJ	GPL	MPFJ
3	12 Feb 2024	Updated MQTT Updated Adding a Comms Link Section	MPFJ	GPL	MPFJ

Table of Contents

1   Introduction

Visual C‖Cure provides access to a range of communications links.

These can be used for:-

- data transfer to/from field devices
- communication to/from supervisory systems
- data transfer between NÅNO devices

Each chapter in this document outlines the basic steps on how to configure each type of link currently supported.

Due to the design of the NÅNO architecture, new types of comms link can easily be added. Contact your supplier for more information.

2   Datamaps

Many comms links require a datamap, which maps an address or name to a particular database pin in the application.

Comms links that have a fixed functionality usually do not require a datamap.

Other comms links (for example the Localpanel comms link) use the displays tree to define the data transferred over the comms link.

Datamaps are managed via the Datamaps tab (but may also be created when adding a new comms link).

Here datamaps can be added, renamed, deleted or copied to the clipboard.

Copying the datamap to the clipboard allows an HTML representation to be pasted into a Word document for ease of documentation.

2.1 Datamap Layout

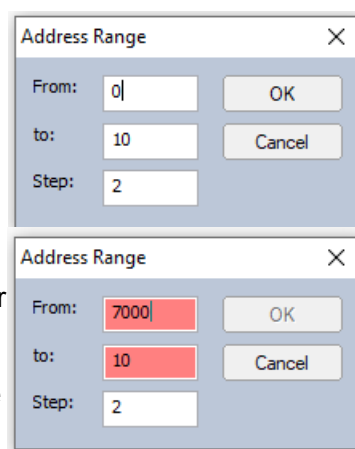Each datamap contains five columns of data as follows:-

Address	This defines the address of the data to be transferred.
	Depending on the comms link, this could just be a reference for readability or used in the comms link itself. See section 2.3 - TAGNAMES Datamap.
I/O Point	This defines which pin within the application is connected to this address.

| Index | When using an array pin, this column allows you to choose which array index is to be referenced.

A blank entry means the pin used is not an array pin. |
|---|---|
| **Format** | This column defines the data type and byte order used when transferring data to/from that address. Depending on the comms link, this column may or may not be used. |
| **Description** | There column is used for information purposes.

Note that some comms links (for example the MQTT comms link) use this column to define additional information required for each pin.

Where applicable, this additional functionality will be explained in the relevant comms link chapter. |

2.2 Managing Addresses

Right-clicking on the datamap allows you to add, remove and change address entries as follows:-

| **Add Addresses** | This brings up a dialog window where you can specify a start and end range of addresses to add.

The "Step" option also allows you to skip every n-th address when adding the new address.

The first example shown would add addresses 0, 2, 4, 6, 8 and 10.

If a clash with any existing address would occur when adding the new addresses, a red highlight will be show to indicate an error. The example shown on the right has an error as the datamap already contains address 7000. |
|---|---|
| **Clear Addresses** | This option clears the application pin(s) defined for the selected address(es). |
| **Delete Addresses** | This option deletes the selected address(es) from the datamap. |
| **Change Format** | This allows you to change the data type and byte order for the selected addresses (if applicable). |
| **Print** | Prints a copy of the full datamap. |
| **Locate** | This locates the selected pin in the application and opens a tab showing its parent container or codeblock. This can also be achieved by double clicking the I/O Point database pin. |
| **Export** | Exports the full datamap as a TSV file.

This file can then be opened as a spreadsheet for manipulation. |

Import	Imports a datamap from a TSV file.
	On selecting this option, a dialog prompts to chose "**Merge**" or "**Overwrite**".
	Merge TSV addresses that already exist in the datamap are not imported.
	Overwrite Existing addresses are overwritten by any matching entries in the TSV file.
	New addresses are added to the existing datamap.
	Existing addresses that are in the datamap, but not in the TSV file that is being imported, are left unchanged.

2.3 TAGNAMES Datamap

Certain comms links (for example OPC UA) make use of a "magic" datamap called TAGNAMES.

These comms links do not use the "Address" column. Instead they require a unique name for each entry rather than an address.

The "Description" column is used to define this name.

The "Format" column is not used.

Below is an example of a TAGNAMES datamap.

Datamaps

TAGNAMES

Add	Rename	Delete	Copy

Address	I/O Point	Index	Format	Description
168	Mass Flow Rate		<auto>	M1FRmass
169	Density of Water		<auto>	M1KVDwater
170	Density Unit		<auto>	FCKVDunit
171	Temperature Unit		<auto>	FCKVTunit
172	Pressure Unit		<auto>	FCKVPselectunit
173	Line_Pressure_Mode		<auto>	FCKVPMTRunittype
174	Vapor Pressure Mode		<auto>	FCKVPEunittype
175	Line Pressure Reference Selection		<auto>	FCSTPMTRunit
176	Override_Pe_unit		<auto>	FCSTPEunit
177	Patm_unit		<auto>	FCSTPATMunit
178	Main Line Volume K Factor Unit		<auto>	FCKVVOLKFselectunit
179	Main Line Mass K Factor Unit		<auto>	FCKVMASSKFselectunit
180	K Factor Volume Unit		<auto>	FCKVVOLKFunit
181	K Factor Mass Unit		<auto>	FCKVMASSKFunit
182	Main Line K Factor Type		<auto>	FCKVKFtype
183	Volume Unit		<auto>	FCKVVOLunit
184	Mass Unit		<auto>	FCKVmassunit
185	Flow Rate Time Unit		<auto>	FCKVFRtimeunit
186	Volume Unit		<auto>	FCKVGVOLunit
187	SVol_Unit		<auto>	FCKVSVOLunit
188	Mass Unit		<auto>	FCKVmassunit
189	Cumulative Total	DP_GOOD	<auto>	M1FTCMTpulse
190	Cumulative Total	IV	<auto>	M1FTCMTIV
191	Cumulative Total	GSV	<auto>	M1FTCMTGSV
192	Cumulative Total	NSV	<auto>	M1FTCMTNSV
193	Cumulative Total	SWV	<auto>	M1FTCMTSWV
194	Cumulative Total	MASS	<auto>	M1FTCMTmass

Add Addresses	Clear Addresses	Delete Addresses

Workspace   Displays   Datamaps   Reports / Live Screens   Comms   Alarms   Historical   Remote

3   Adding a Comms Link

To add a new comms link, select the Comms tab and click on the "Add Comms Link" button at the top.
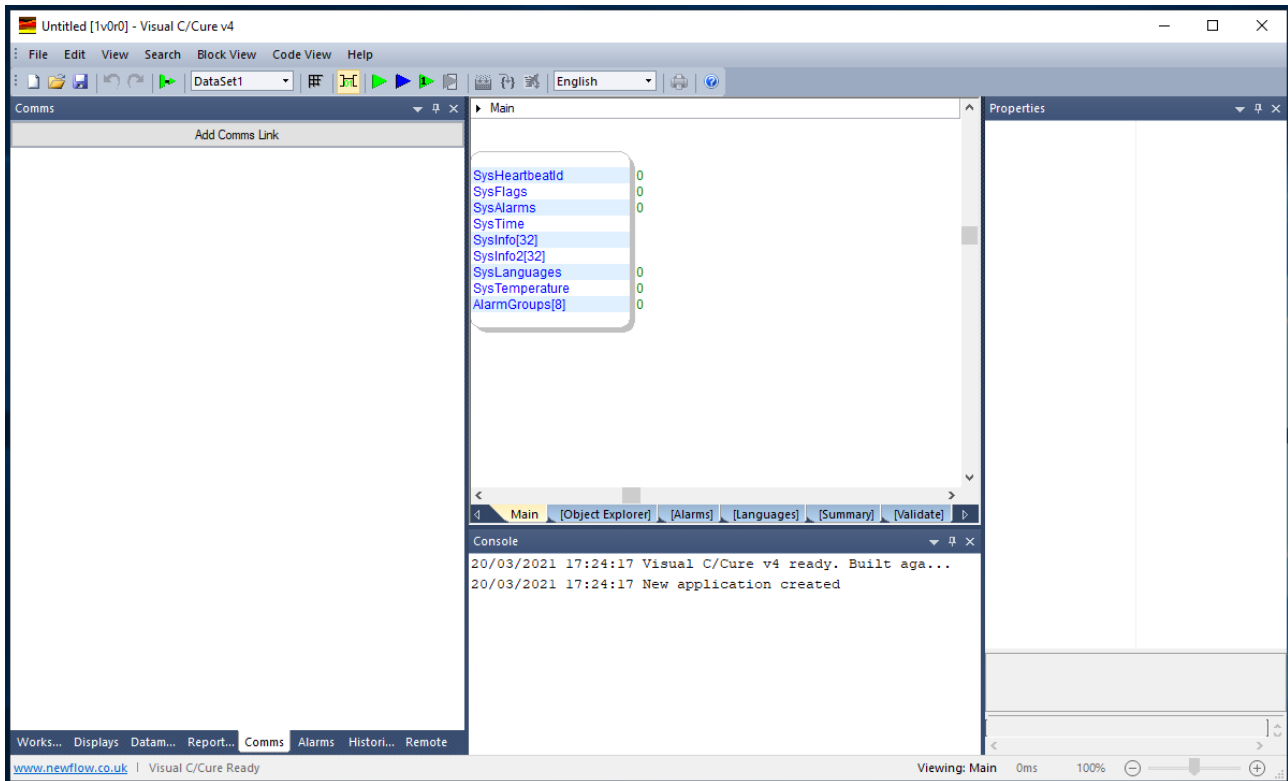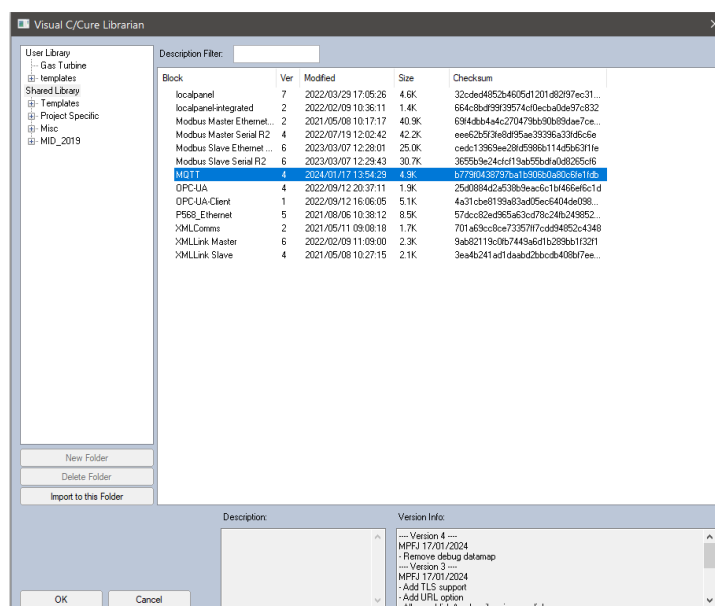

Depending on whether there are any existing comms links in the app, the header may look more like this;
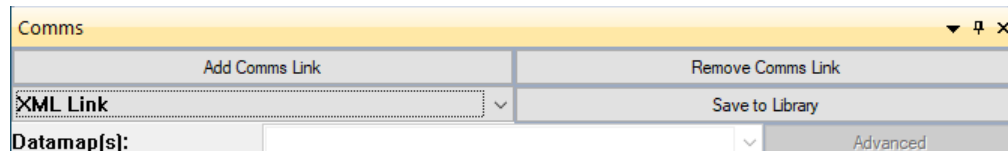
3.1 Choose the Comms Link type

A dialog window will appear, showing the list of available links.

Select location of the library, such as Shared, then select the link type from the list.

Then click the OK button.

Note that the name of the link can be modified using the same method as modifying the name of a Codeblock or Container (by clicking on the block in the main pane and modifying the name under the Properties pane). If no name or invalid characters are entered or the name already exists in the application, the box will turn red.

3.2 Select a Datamap

If the link requires a datamap, a second "Select Datamap" dialog will be shown as follows:-

3.2.1   Select the Default Datamap

This will create a new datamap (called the same as the comms link name) and load the predefined settings from the comms link file.
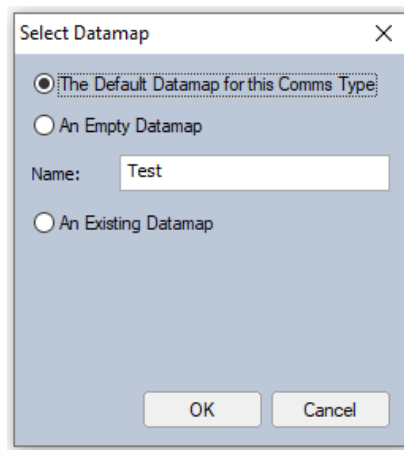
### 3.2.2   Create a New Empty Datamap

This will create a new datamap using the name in the text entry box.

You can change the name of the datamap but the name must be unique. Any clash with any existing datamap will be highlighted in red.
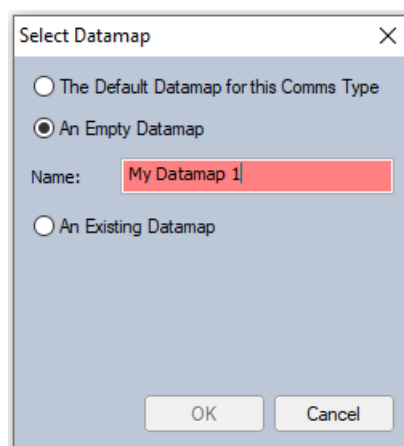
3.2.3　Choose an Existing Datamap

This option allows you to chose an existing datamap for use with this comms link. This is useful when the same datamap is required in multiple comms links as it only needs to be defined once.
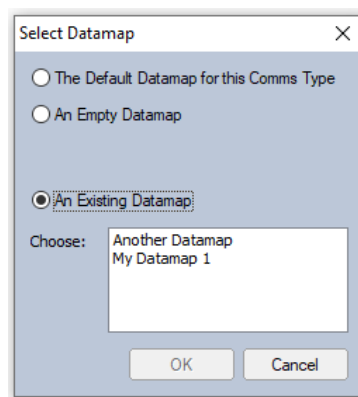

3.3 Managing Comms Polls

Once the comms link has been added, the comms tab will change to show the newly added comms link, plus any pre-selected datamap.

If the added comms link is capable of polling, you can right-click and select "New Poll", otherwise when right-clicking the entries will be greyed out.
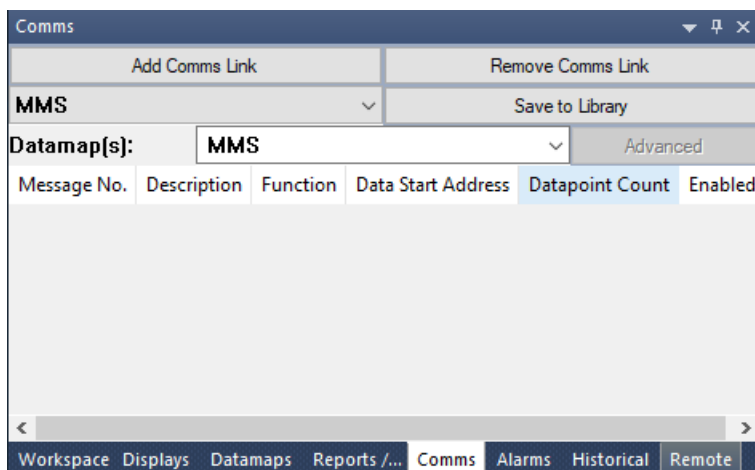
This will then open the "Poll Properties" dialog showing the options and parameters specific to the comms link. Refer to the relevant section for more details.
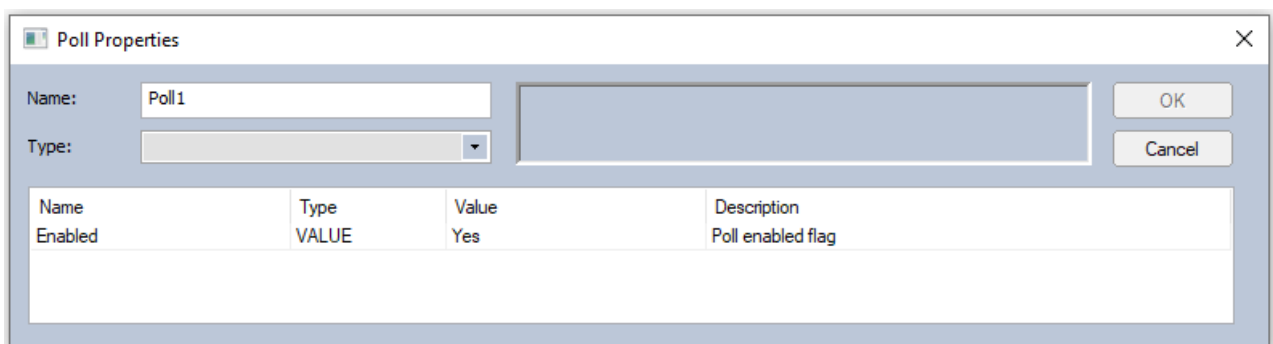
After a poll has been added, it will be shown in the list for the comms link. For example:-

Double-clicking any poll will load up the "Poll Properties" dialog allowing you to edit the poll parameters.

Right-clicking shows further options that can now be used:-

Delete	Deletes the poll from the list.
Move Up in Poll Order	Polls can be moved up / down in the poll order, which determines the order in which they are processed when the comms link is active.
Move Down in Poll Order	

The displayed top to bottom order defines the poll sequence, not the poll name as this name can be changed.

4   Modbus Master Serial R2

The Modbus Master Serial comms link allows connections to Modbus serial slave devices.
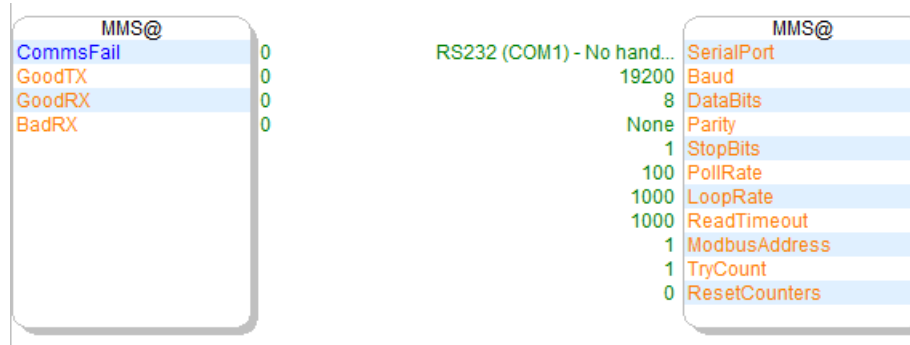
The link uses Modbus polls, each of which is sent in turn to the slave device. Once all the polls have been processed, the link loops back to the first poll and starts all over again.

Both an inter-poll and an inter-loop delay can be defined.

4.1 Control Pins

SerialPort	Selects the comms port to be used.
Baud	Selects the baud rate to be used.
DataBits	Selects the number of databits to be used.
Parity	Selects the parity type to be used.
StopBits	Selects the number of stopbits to be used.
PollRate	Sets the time (in milliseconds) between sending each configured poll in the list of polls. Known as the Inter-Poll delay.
LoopRate	Sets the time (in milliseconds) between sending each list of polls. At the end of the poll list, this is the time the comms waits before starting at the top of the list again. Known as the Inter-Loop delay.
ReadTimeout	Sets the timeout (in milliseconds) to wait for a reply from the slave device. This setting, along with the TryCount value, determines when to increment the "BadRX" count.
ModbusAddress	Sets the address of the slave device.
TryCount	Sets the number of read timeouts before incrementing the "BadRX" count.
ResetCounters	Setting this pin to a '1' will reset the CommsFail and Good/Bad status pins at the start of the next poll loop. The pin is automatically reset to a '0' when the pins have been cleared.

4.2 Status Pins

CommsFail	Returns a '1' when a comms error occurs.
GoodTX	Shows the number of good transmitted messages.
GoodRX	Shows the number of good received messages.

BadRX	Shows the number of failed received messages.
	This may be due to a timeout, a checksum error or formatting issue.

4.3 Datamap

This link requires a datamap to be configured and selected in the Datamaps drop-down list.

4.4 Message Polls

To add a new poll, right-click on the poll list and select "New Poll". This will bring up the "Poll Properties" dialog.

Enter a name for the poll, then select the Modbus function type from the drop-down list.

This will then populate the dialog with a list of parameters to be filled in.

Each parameter has the four following columns:-

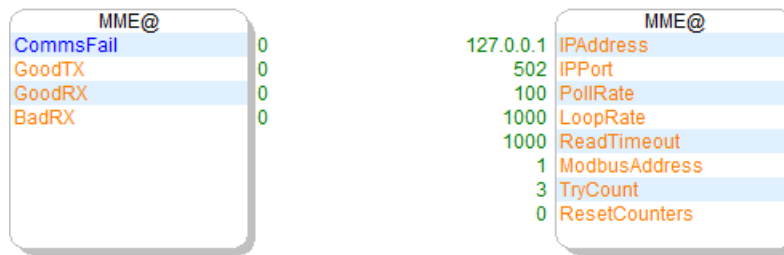| Name	Parameter name
Type	This determines how the list of options in the "Value" column is populated.
	VALUE A simple Yes or No option
	INPUT Populates the "Value" column with a list of all the control pins into the comms link as shown in 4.1 - Control Pins.
	Selecting one of the control pins will use the value of that pin for the parameter setting.
	OUTPUT Populates the "Value" column with a list of all the status pins from the comms link as shown in 4.2 - Status Pins.
	Selecting one of the status pins will use the value of that pin for the parameter setting.
	CCURE <reserved for future use>
Value	Sets the value to use for this parameter.
Description	Information on what the parameter is used for.

The following parameters are shown:-

Enabled	When this parameter "INPUT" or "OUTPUT", the pin value is used to enable or disable the poll, as follows:-
	value < 0 **Single shot**: the poll is processed once and then disabled. • If the pin is persistent, the pin value is set to -999 to indicate the poll has been completed. When the pin value next changes, the poll is processed as normal. • For non-persistent pins, to re-enable the poll, the value must first be set to 0 (to clear the single shot mode) and then set to non-zero as required.
	value = 0 **Disable**: the poll is skipped and not processed in the poll loop.
	value > 0 **Enable**: the poll is processed in the poll loop.
	By default, this is set to "VALUE" type "Yes" so the poll is always enabled.
Slave Address	Sets the Modbus Slave Address to be used. By default, this is set to "INPUT" type "ModbusAddress".
Start Address	Sets the start address / register for the poll. This address must correspond with an address in the datamap. If no such address exists, the poll will be ignored.
Count	Sets the number of addresses / registers to be polled for.
Offset	This parameter allows an offset to be applied to the datamap address. This can be useful when requesting data from multiple slave devices with the same Modbus address map but where the datamap address is different per device.
Format	Currently the option here is "Custom".
Addressing Mode	Two Modbus addressing modes are currently supported:- **Modicon** Each address represents a 16bit register. Data types with more than 16bits (e.g. FLOAT3210) will span multiple addresses. **Logical** Each address represents a single datatype, regardless of the bit size.
Data Type	Lists the currently supported data types along with the byte ordering variations. For example, the decimal integer value 305419896 is 0x12345678 in hexadecimal. For INT3210, the data would be sent as 12 34 56 78. For INT2301, the data would be sent as 34 12 78 56. For INT0123, the data would be sent as 78 56 34 12. For INT1032, the data would be sent as 56 78 12 34.

The information box (to the left of the OK and Cancel buttons) provides a summary of how the Modbus message will be formatted based on the selection of Addressing Mode and Data Type.

5   Modbus Master Ethernet R2

The Modbus Master Ethernet comms link allows connections to ModbusTCP slave devices.

```
        MME@                              MME@
CommsFail      0          127.0.0.1  IPAddress
GoodTX         0                502  IPPort
GoodRX         0                100  PollRate
BadRX          0               1000  LoopRate
                               1000  ReadTimeout
                                  1  ModbusAddress
                                  3  TryCount
                                  0  ResetCounters
```

5.1 Control Pins

IPAddress	Selects the IP address of the slave device.
IPPort	Selects the network port of the slave device.
PollRate	Sets the time (in milliseconds) between sending each configured poll in the list of polls. Known as the Inter-Poll delay.
LoopRate	Sets the time (in milliseconds) between sending each list of polls. At the end of the poll list, this is the time the comms waits before starting at the top of the list again. Known as the Inter-Loop delay.
ReadTimeout	Sets the timeout (in milliseconds) to wait for a reply from the slave device. This setting, along with the TryCount value, determines when to increment the "BadRX" count.
ModbusAddress	Sets the address of the slave device.
TryCount	Sets the number of read timeouts before incrementing the "BadRX" count.
ResetCounters	Setting this pin to a '1' will reset the CommsFail and Good/Bad control pins at the start of the next poll loop. The pin is automatically reset to a '0' when the pins have been cleared.

5.2 Status Pins

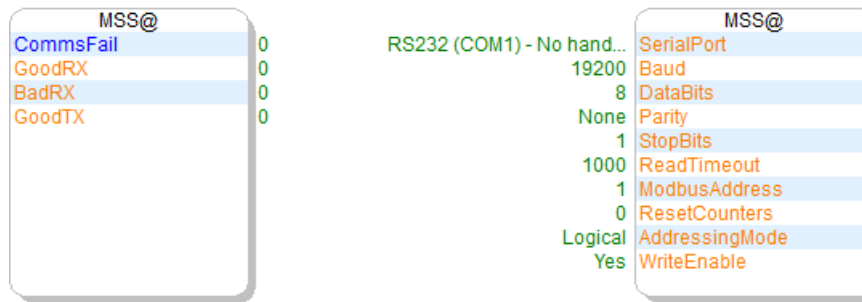| CommsFail	Returns a '1' when a comms error occurs.
GoodTX	Shows the number of good transmitted messages.
GoodRX	Shows the number of good received messages.
BadRX	Shows the number of failed received messages. This may be due to a timeout, checksum error or formatting issue.

5.3 Datamap

This link requires a datamap to be configured and selected in the Datamaps drop-down list.

5.4 Message Polls

Identical to the Modbus Master Serial link (see section 4.4 - Message Polls).

6 Modbus Slave Serial R2

The Modbus Slave Serial comms link allows the unit to respond to Modbus serial master requests.

```
         MSS@                                          MSS@
CommsFail          0         RS232 (COM1) - No hand...  SerialPort
GoodRX             0                          19200     Baud
BadRX              0                              8      DataBits
GoodTX             0                           None      Parity
                                                  1      StopBits
                                               1000      ReadTimeout
                                                  1      ModbusAddress
                                                  0      ResetCounters
                                            Logical      AddressingMode
                                                Yes      WriteEnable
```

6.1 Control Pins

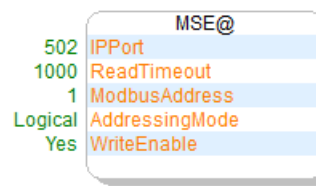|	
SerialPort	Selects the comms port to be used.
Baud	Selects the baud rate to be used.
DataBits	Selects the number of databits to be used.
Parity	Selects the parity type to be used.
StopBits	Selects the number of stopbits to be used.
ReadTimeout	Sets the timeout (in milliseconds) to wait for a request from the master device. Setting to a value of 0 (zero) disables this check.
ModbusAddress	Sets the address of the slave device.
ResetCounters	Setting this pin to a '1' will reset the CommsFail and Good/Bad status pins at the start of the next poll loop. The pin is automatically reset to a '0' when the pins have been cleared.
AddressingMode	Selects the Modbus addressing mode for this link, either Modicon or Logical.
WriteEnable	Selects whether writes are enabled for the link. Any non-zero value will allow writes to take place on this link. If this option is set to "No" (defined as value 0), write requests result in a Modbus exception.

6.2 Status Pins

CommsFail	Returns a '1' when a comms error occurs.
GoodRX	Shows the number of good received messages.
BadRX	Shows the number of failed received messages. This may be due to a timeout, checksum error or formatting issue.
GoodTX	Shows the number of good transmitted messages.

6.3 Datamap

This link requires a datamap to be configured and selected in the Datamaps drop-down list.

7   Modbus Slave Ethernet R2

The Modbus Slave Ethernet comms link allows the unit to respond to ModbusTCP master requests.

```
               MSE@
     502 | IPPort
    1000 | ReadTimeout
       1 | ModbusAddress
 Logical | AddressingMode
     Yes | WriteEnable
```

7.1 Control Pins

IPPort	Selects the network IP port to be used.
ReadTimeout	Sets the timeout (in milliseconds) to wait for a request from the master device.
ModbusAddress	Sets the address of the slave device.
AddressingMode	Selects the Modbus addressing mode for this link, either Modicon or Logical.
WriteEnable	Selects whether writes are enabled for the link. Any non-zero value will allow writes to take place on this link. If this option is set to "No" (defined as value 0), write requests result in a Modbus exception.

7.2 Status Pins

This link has no status pins.

7.3 Datamap

This link requires a datamap to be configured and selected in the Datamaps drop-down list.

8   XMLComms

The XMLComms comms link provides access to the pins defined in the "Remote" displays tree, using the NÅNO XML schema. It also provides limited support for OPC XML-DA.

Refer to the NÅNO XML Communications document (Ref: NF_XMLM).

8.1 Control Pins

This link has no control pins.

8.2 Status Pins

This link has no status pins.

8.3 Datamap

This link does not use a datamap, however to have limited support for OPC XML-DA the TAGNAMES datamap can also be used (see section 2.3 - TAGNAMES Datamap).

9   XMLLink Master / Slave

The XMLLink Master and XMLLink Slave comms links allow quick and easy transfer of data between one master NÅNO device and up-to 31 slave NÅNO devices on a local network.

No complex configuration (IP address, IP ports, etc) is required as all devices broadcast their "Ident" value and then only listen for packets from the master that match that value.

When any XMLLink master or slave device starts, it announces itself across the network allowing an automatic connection to be established between the master and slaves.

The master has the ability to enable / disable comms traffic to individual slave devices using the "SlaveEnable" control pin. When a slave is disabled, the master will not send data to or receive data from the slave device. However, the master will still "ping" each slave to ensure the communication link has not been lost.

The status of each slave is also monitored and reported back as online or offline via the "LinkStatus" status pin.

Once a connection is established and the slave is enabled, data will be exchanged on every heartbeat.

9.1 Master Link

9.1.1   Control Pins

Ident	Sets the link identification value.
	This value must be unique for each XMLLink Master comms link used on the local network.
	Setting the Ident value to zero will disable the link.
ReadTimeout	Sets the timeout (in milliseconds) to wait for a reply from the slave devices.
	Any slave device that has been "seen" must reply within this time, otherwise the relevant bit in the "LinkStatus" pin will be set.
SlaveEnable	Allows the master to temporarily enable or disable any slave device.
	This pin uses a bit mask with each bit representing a single slave.
	Slave 1 Bit 0 Slave 2 Bit 1 Slave 3 Bit 2 … Slave 29 Bit 28 Slave 30 Bit 29 Slave 31 Bit 30
	Setting the bit to '0' will disabled comms to that slave.
	Setting the bit to '1' will enable comms to that slave.
	Bit 31 is reserved for future use.

9.1.2   Status Pins

LinkStatus	Indicates the status of each slave device.
	This pin uses a bit mask with each bit representing a single slave.
	Slave 1 Bit 0 Slave 2 Bit 1 Slave 3 Bit 2 … Slave 29 Bit 28 Slave 30 Bit 29 Slave 31 Bit 30
	A bit value of '0' indicates the salve device has either:-
	a) never been seen or
	b) has now gone offline
	A bit value of '1' indicates the slave device is currently online.
	Bit 31 is reserved for future use.

9.2 Slave Link

9.2.1   Control Pins

Ident	Sets the link identification value.
	This value must be match the relevant XMLLink Master comms link.
ReadTimeout	Sets the timeout (in milliseconds) to wait for a message from the master device.
	The master device must poll within this time, otherwise the relevant bit in the "LinkStatus" pin will be set.
Address	This is a unique address for the comms link and is used to decide which values in the Master's Datamap to transfer to/from this slave.
	It is also used to indicate at the Master end, whether the comms is Up or Down to this address.

9.2.2   Status Pins

LinkStatus	Indicates the status of the link to the master device, as follows:-
	0 Offline **1** Online but comms has been "disabled" by the master **2** [not used] **3** Online with comms

9.3 Datamap

For any particular link, the applications installed on the master device and all slaves devices must have aligned datamaps.

For the datamap on the master device, each slave has its own address range, split into two blocks of 5000 addresses. The first block defines the data to be sent from the master to the slave and the second block defines the data to be sent from the slave back to the master.

The table below shows how the addresses in the master's datamap map to each slave.

Slave	Master Out, Slave In	Master In, Slave Out
1	10000 … 14999	15000 … 19999
2	20000 … 24999	25000 … 29999
3	30000 … 34999	35000 … 35999
…	…	…
29	280000 … 284999	285000 … 289999
30	290000 … 294999	295000 … 299999
31	300000 … 304999	305000 … 309999

For the datamap on each slave device, only the 10000 … 19999 range is used.

The master device automatically adjusts the relevant slave address range in the master datamap onto the slave datamap addresses so that each slave only needs to support the 1xxxx range.

For example, consider a simple setup with a master device and three slave devices, each of which has a datamap with only five addresses:-

10000	Command	Data sent from the master to the slave.
10001	Data1	
10002	Data2	
15000	System Time	Data sent from the slave back to the master.
15001	Status	

The diagram below shows the relationship between the master datamap addresses and the slave datamap addresses.
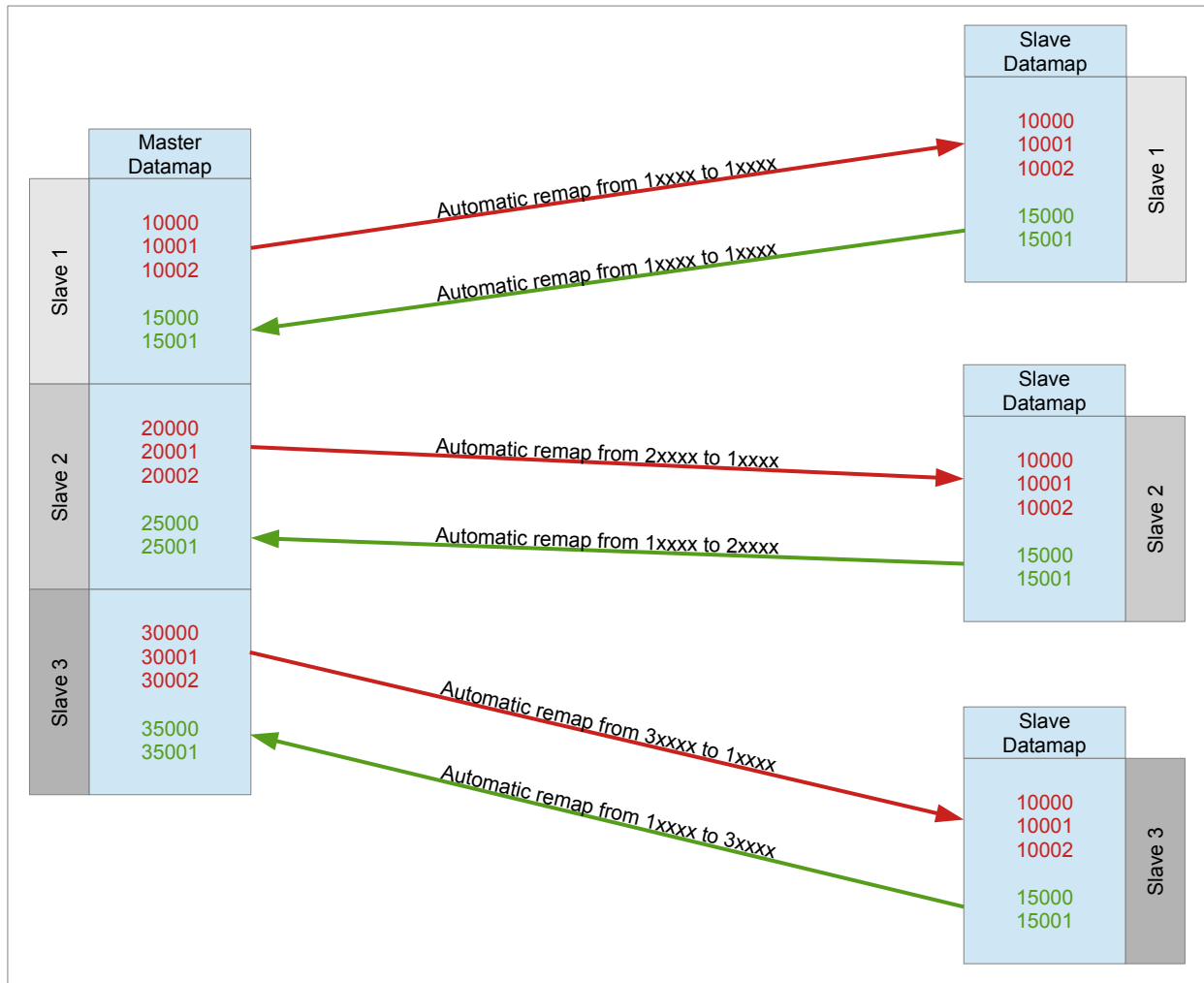
For data being sent from the master to the slave:-

- The master maps the datamap "slave 1" address range (10000…19999) directly onto the same address range as the slave's datamap.

- The master takes the datamap "slave 2" address range (20000…29999) and offsets it down by 10000 so it remaps directly onto the address range of the slave's datamap.

- The master takes the datamap "slave 3" address range (30000…39999) and offsets it down by 20000 so it remaps directly onto the address range of the slave's datamap.

For data being sent back to the master from to the slave:-

- For slave 1, the master maps the slave address range (10000…19999) directly onto the same address range in the master's datamap.

- For slave 2, the master takes the slave address range (10000…19999) and offsets it up by 10000 to remaps onto the "slave 2" address range (20000…29999) of the master's datamap.

- For slave 3, the master takes the slave address range (10000…19999) and offsets it up by 20000 to remaps onto the "slave 3" address range (30000…39999) of the master's datamap.

10   OPC UA
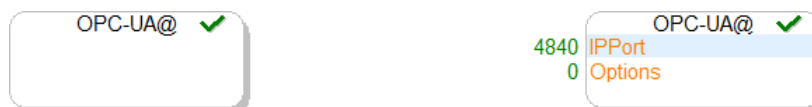
The OPC UA comms link allows the NÅNO to act as an OPC UA server.

Below is a screenshot of a NÅNO connected to the UaExpert OPC client from Unified Automation.

10.1   Control Pins

IPPort	Selects the server port.
	To disable the link, this pin can be set to 0.
Options	Enables / disables the following options:-
	Bit 0 Enables ControlWeb support mode. Please contact your supplier for more details.
	Bit 31..1 Unused

10.2   Status Pins

This link has no status pins.

10.3   Datamap

This link uses the TAGNAMES datamap (see section 2.3 - TAGNAMES Datamap).

10.4   Alarms

All alarm items defined in the application are listed under "Root/Objects/Alarms".

Due to its complexity, the standard Alarms and Conditions section of OPC UA has not been implemented in this comms link.

Instead, a much simpler system has been designed using only two nodes per alarm:- Accept and State.

State	The State node shows the current state of the alarm as follows:-	
	0 (Clear)	The item is not in its alarm condition and has not been triggered since the last time it was accepted.
	1 (Set)	The alarm is in its alarm condition but has been accepted / acknowledged.
	2 (UnAccepted / Clear)	The alarm is not in its alarm condition but its previous alarm condition has not been accepted / acknowledged.
	3 (UnAccepted / Set)	The alarm is in its alarm condition and has not been accepted / acknowledged.
Accept	The Accept node is used to accept an unaccepted alarm.	
	Writing a '1' to this node will accept the alarm. Once accepted, the State node will change as follows:-	
	Old State	**New State**
	2 (UnAccepted / Clear) → 0 (Clear)	
	3 (UnAccepted / Set) → 1 (Set)	

10.5   Reports

All reports defined in the application are listed under "Root/Objects/Reports".

Each report has four nodes:- DateStamp, NewestId, RequestId and Text.

| DateStamp	This node shows the date and time of the current requested report.
	If no report is being viewed, the value is reset to "zero".
NewestId	This node shows the ident number of the most recent report available for this report type.
	The node can be monitored to determine when new reports have been created.

RequestId	Writing a valid ident number to this node will extract the report data from the NÅNO device and populate the "Text" node array.
	The node then shows the ident number of the report currently being view.
	If an invalid ident number is used, the node value will reset to zero and the "Text" node will be cleared.
Text	This is a string array node that displays the contents of the currently selected report.
	A maximum of 128 report lines is supported. Reports longer or shorter than this will be truncated / padded as required.

Taking the above screenshot as an example, we can see that ident number 42 is the most recent report available and there is no report currently selected (all the other nodes are reset).

If we now write 42 to the "RequestId" node, the NÅNO will fill in the other node accordingly, as follows:-

| #	Server	Node Id	Display Name	Value	Datatype
1	Liquid Flow Computer	NS3|String|Daily Report:DateStamp	DateStamp	2021-03-20T02:24:44.763Z	DateTime
2	Liquid Flow Computer	NS3|String|Daily Report:NewestId	NewestId	42	Int32
3	Liquid Flow Computer	NS3|String|Daily Report:RequestId	RequestId	42	Int32
4	Liquid Flow Computer	NS3|String|Daily Report:Text	Text	{'<b><u>DAILY REPORT ...	String

In the UaExport tool, we can also double-click on the "Text" node to show the text array in full.
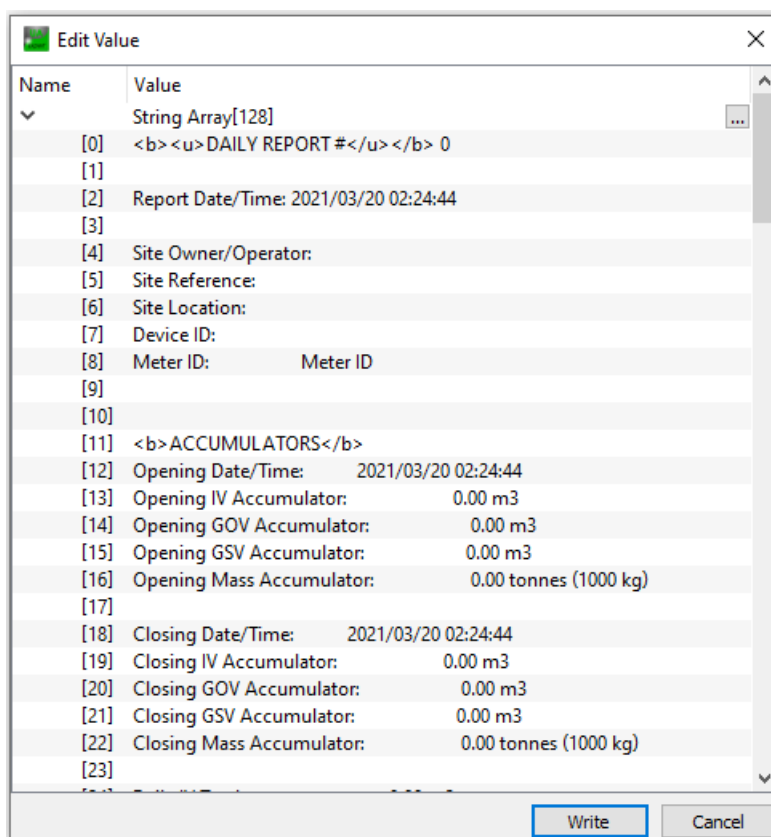
10.6   TagNames

All valid data items defined in the TAGNAMES datamap are listed under "Root/Objects/TagNames".

11 OPC UA Client
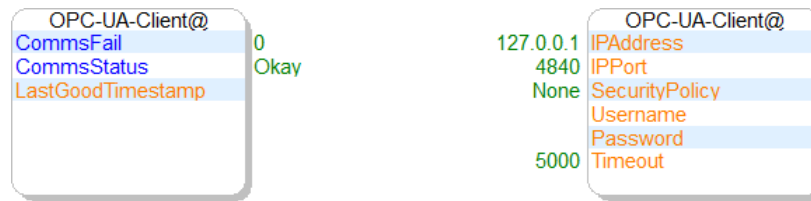
The OPC UA Client comms link allows the NÅNO to communicate with an external OPC UA server.

11.1 Control Pins

IPAddress	Selects the IP address of the server.
	To disable the link, this pin can be set to address 0.0.0.0.
IPPort	Selects the network port of the server.
	To disable the link, this pin can be set to 0.
SecurityPolicy	Selects the Security Policy to be used.
	None No additional security is used.
	Sign All messages are signed but not encrypted.
	Sign & Encrypt All messages are signed and encrypted.
Username	If needed, sets the username for the login process.
	Note that "anonymous" logins are possible by simply leaving the username blank.
Password	If needed, sets the password for the login process.
Timeout	Sets the time (in milliseconds) before a "comms fail" is raised if no communication occurs between the client and server.

11.2 Status Pins

CommsFail	Returns a '1' when a comms error occurs.
CommsStatus	Shows the current state of the link as follows:-
	0 (Okay) The link between NÅNO and server is good.
	-1 (Disabled) The link is currently disabled.
	-2 (Connection Timeout) The NÅNO is unable to connect to the server at the specified IP address & port.
	-3 (Invalid login) The supplied username & password are not valid.
	Other status values may be displayed – these related to the standard OPC UA status codes and are included for support purposes only.
LastGoodTimestamp	Shows the timestamp of the last successful poll.

11.3   Datamap

This link requires a datamap to be configured and selected in the Datamaps drop-down list.

Below is an example of a simple datamap configured for an OPC UA Client link.

MyClientMap				
Add	Rename	Delete	Copy	
Filter:				

Address	I/O Point	Index	Format	Description
0	ClientDouble		<auto>	ns=2;s=TAG_MYDOUBLE
1	ClientInt		<auto>	ns=2;s=TAG_MYINT
2	ClientTime		<auto>	ns=2;s=TAG_TIME
3	ClientString		<auto>	ns=2;s=TAG_UPTIME
10000	ClientWrite		<auto>	ns=2;s=TAG_WRITE_TEST

Each entry requires three pieces of information, as follows:-

Address	For any entry, two operating modes are supported:-
	Read-only — Where the NÅNO only wants to read data from the server, use an address in the range 0-9999.
	For any such pin, when the value on the server changes, the corresponding pin on the client is updated.
	Note that if the NÅNO application updates the pin value, this change is not sent down to the server – the NÅNO will only update pin value when the server value is next changed.
	Read/write — Where the NÅNO wants to both read data from and write data to the server, use an address above 10000.
	For any such pin, when the value on the server changes, the corresponding pin in the NÅNO application is updated.
	If the NÅNO application updates the pin value, this change is sent down to the server.
	Note that the address itself is irrelevant – it is simply used to specify which mode the entry is to use.
I/O Point	The pin in the NÅNO application to read/write data from/to.
Description	This is the server node ID of the data point to be used.
	This can be in any of the following standard OPC UA formats:-
	• `i=12345` • `ns=10;i=1` • `ns=4;s=Reader.Temperature` • `g=09087e75-8e5e-499b-954f-f2a9603db28a` • `ns=2;b=b3BlbjYyNTQxIQ==`

So in the example datamap shown previously, there are four "read-only" items and a single "read/write" item.

For address 0, the NÅNO will connect to the server and request node "TAG_MYDOUBLE" in namespace 2. Whenever the value of this node changes, the new value is pushed up from the server and the NÅNO stores the new value in the application pin "ClientDouble".
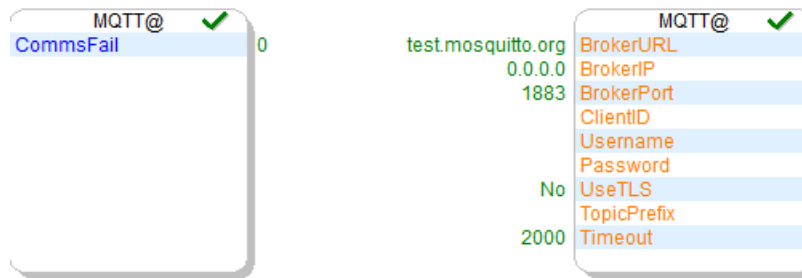
The same occurs for the items shown in addresses 1, 2 and 3.

For the item at address 10000, this is a "read/write" item. Again, the NÅNO will connect to the server and, this time, request node "TAG_WRITE_TEST" in namespace 2. Whenever the value of this node changes, the new value is pushed up from the server and the NÅNO stores the new value in the application pin "ClientWrite".

However, in addition, if the value of pin "ClientWrite" is changed by the application, this value will be sent to the server.

12 MQTT

The MQTT comms link provides access to the industry standard MQTT protocol (ISO/IEC 20922) with the NÅNO acting as a client.

Note that the MQTT comms link does not support topic wildcards.

12.1 Control Pins

BrokerURL	Sets the URL of the broker.
	If this is left blank, the BrokerIP will be used.
BrokerIP	Sets the IP address of the broker.
	This is only used when the BrokerURL is blank.
BrokerPort	Sets the IP port of the broker.
ClientID	Sets the (optional) Client ID to send to the broker.
Username	Sets the (optional) username to send to the broker.
Password	Sets the (optional) password to send to the broker.
UseTLS	Selects the Security Policy to be used, as follows:-
	No No encryption is used.
	Yes Connect using TLS encryption.
TopicPrefix	An optional prefix can be added to each topic. This will only be added when the first character of the topic (as set in the "Description" cell of the datamap entry) is a '/'.
	If the prefix is left blank, the default of "nanonano/<DEVICEID>" is used where <DEVICEID> is the unique 12 character serial number of the NÅNO unit, for example "nanonano/C8A0308399A3".
	A maximum of 64 characters is allowed.
	As an example, if the prefix is set to "helloworld", consider the two datamap entries:-
	Address **I/O Point** **Description** 0 SysTime /unit1/system_time 1 SysTemperature unit1/system_temp
	The first topic starts with a '/' character so the prefix will be added to create a topic "helloworld/unit1/system_time".
	However, the second topic does **not** start with a '/' character so the prefix will not be added and the topic will remain at "unit1/system_temp".

Timeout	Sets the timeout (in milliseconds) to wait for the broker to connect or respond.

12.2   Status Pins

CommsFail	Returns a '1' when a comms error occurs.

12.3   Datamap

This link requires a datamap to be configured and selected in the Datamaps drop-down list.

Addresses in the range 0 to 4999 can be used to publish topics.

Addresses in the range 5000 to 9999 can be used to subscribe to topics.

The screenshot below shows a simple datamap with two points.

Address 0	This will publish the current system temperature using topic "mynanotest/unit1/system_temp".
Address 5000	This will subscribe to the topic "factory/conveyor_belt/speed" and store any updates into the "NewValue" I/O point.

13 Localpanel

The Localpanel comms link is used to communicate to a suitable local touch panel.

Currently only the iCOP PDX2 (a 9" resistive touch panel) is supported.

The Localpanel comms link uses the menu stricture of the "Local" displays tree to enable a simple screen-based HMI interface to the NÅNO.

The following primitives are supported:-

- Information panels supporting Title, Value and Units (or any combination)
- Data entry via on-screen keyboard (Integer / Double / String / DateTime)
- View / Accept alarms
- Buttons
- Configurable row and column layout
- Report viewing
- QR Codes
- Toggle panels
- PIN code entry
- Selection lists

Most primitives support:-

- Background / foreground colours
- Text sizes
- Text styles (bold, italic, strike-through, underline, centre)

A mimic of the local panel screens is also available with the internal website, although an exact replica is not possible due to font limitiations.

Refer to the NÅNO Website / Local Panel Attributes and XML Schema for further details.

13.1   Control Pins

This link has no control pins.

13.2   Status Pins

This link has no status pins.

13.3   Datamap

This link does not use a datamap.

14 Localpanel-Lid

The Localpanel-Lid comms link is used to communicate to the optional lid display available for the NÅNO.

Currently only a 2.7" panel is supported.

The Localpanel-Lid comms link uses the menu structure of the "Lid" displays tree to enable a simple "display-only" screen-based HMI interface to the NÅNO.

By default, the lid takes the current active menu of the "Lid" displays tree and displays the Title, Value and Units of each screen item defined therein.

The lid loops through each item, displaying them for 5 seconds.

Additionally, up to 30 active alarm conditions can be displayed on the bottom line of the display.

Future enhancements planned are:-

- Sound beeper
- Text message override
- Line drawing
- Box drawing
- Single line chart

14.1   Control Pins

This link has no control pins.

14.2   Status Pins

This link has no status pins.

14.3   Datamap

This link does not use a datamap.