



...Display, Control, Communicate



# Tutorial

*i*CAN Network

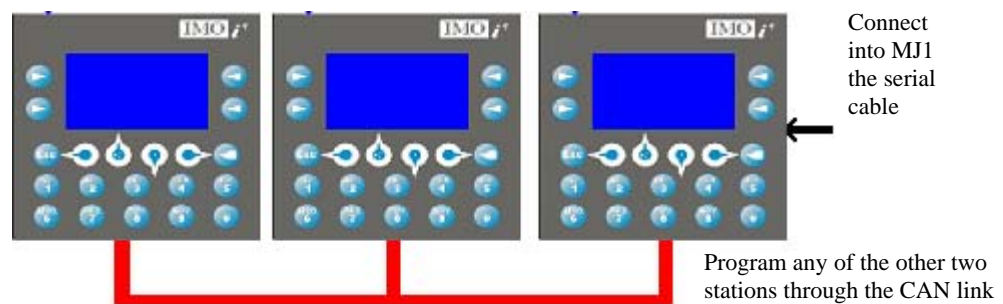
## Table of contents

<i>Table of contents</i> .....	2
<i>Introduction</i> .....	3
<i>Checking the ID and Programming a CAN i<sup>3</sup></i> .....	4
<i>CAN Network Wiring</i> .....	6
<i>Programming one i<sup>3</sup> through the CAN port of another</i> .....	7
<i>CAN Ladder Instructions</i> .....	8
<b>Net Get Words</b> .....	8
<b>Net Put Words</b> .....	9
<b>Net Get Heartbeat</b> .....	10
<b>Net Put Heartbeat</b> .....	10
<i>Programming Example</i> .....	12
<b>i<sup>3</sup> ID:1</b> .....	12
<b>i<sup>3</sup> ID:2</b> .....	14
<b>Running the Program</b> .....	15

## Introduction

With two models of the i3 there is an additional communication port, the i3A12X/10B04-SCH, the i3A12X/10D03-SCH, and all of the new i3B touch-screen range. The CAN port allows the *i*<sup>3</sup> to be an iCAN master and communicate with iCAN enabled devices such as remote Smart IO and other *i*<sup>3</sup>'s. This allows for a network of up to 252 iCAN enabled devices interconnected, sharing data. iCan can ONLY be a mix of *i*<sup>3</sup>'s and Smart IO. For programming with the CAN remote IO modules please see the Remote IO tutorial. A limited version of CAN Open will be released as an optional firmware upgrade in the future.

Another feature that is possible through the CAN port is the remote programming of *i*<sup>3</sup>'s on the network. In the *i*<sup>3</sup>-configurator software it is possible to select the target *i*<sup>3</sup> connected in the network and program it through another *i*<sup>3</sup>. This gives us the benefit of have an *i*<sup>3</sup> configured with the internal modem (using MJ1) and being able to change the program through the CAN port whilst it is running.

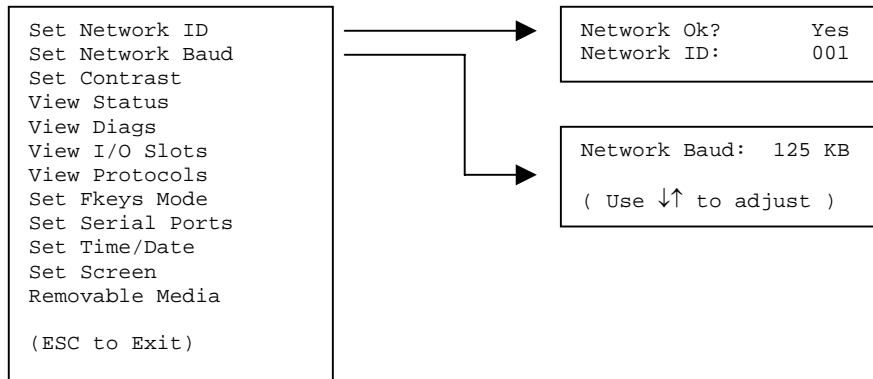


In this tutorial we will set up and demonstrate a small network of iCAN enabled *i*<sup>3</sup>'s communicating and sharing data.

We will also demonstrate programming one *i*<sup>3</sup> through another's CAN port. This is very useful when using an advanced communication option such as the GSM modem or Ethernet.


## Checking the ID and Programming a CAN *i*<sup>3</sup>

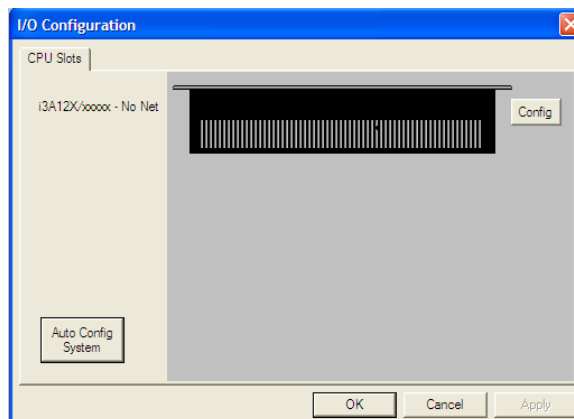
When programming the CAN enabled devices it is important to check the station ID of the *i*<sup>3</sup> through the main system menu. Press the UP and DOWN arrow to gain access to this menu.



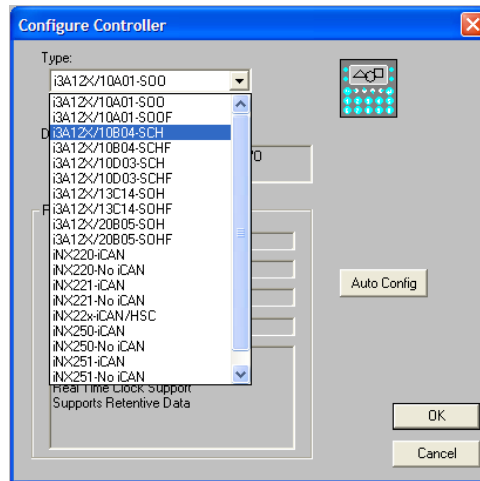
To check the Network ID press enter on the option Set Network ID. If you want to change the ID then press enter on the Network ID and when highlighted you can change the number between 1 and 253. It is also possible to change the network baud rate of the CAN network through the Set Network Baud option.

With having to set and check the Network ID, it is not always possible to use the Auto-config option in the *i*<sup>3</sup>-configurator software. The best procedure for connection is as follows:

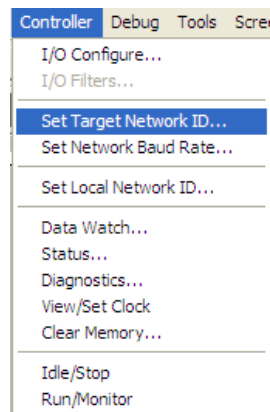
1. Click on the IO configure icon 
2. Click on the configure button at the top right next to the *i*<sup>3</sup> base.



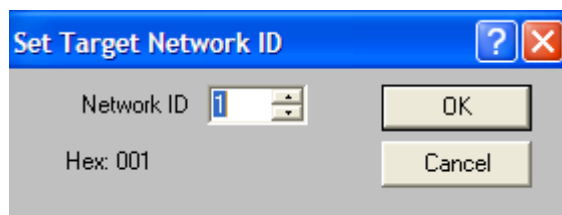
3. Manually select any *i*<sup>3</sup> that is CAN enabled and click OK to confirm and exit. Click OK back on the IO configure window to exit back to the main programming window.



- Now select from the Controller menu “Select Target ID”

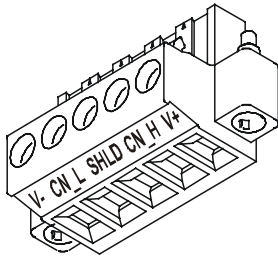


- Enter the number to match what is set in the  $i^3$  connected and click OK.



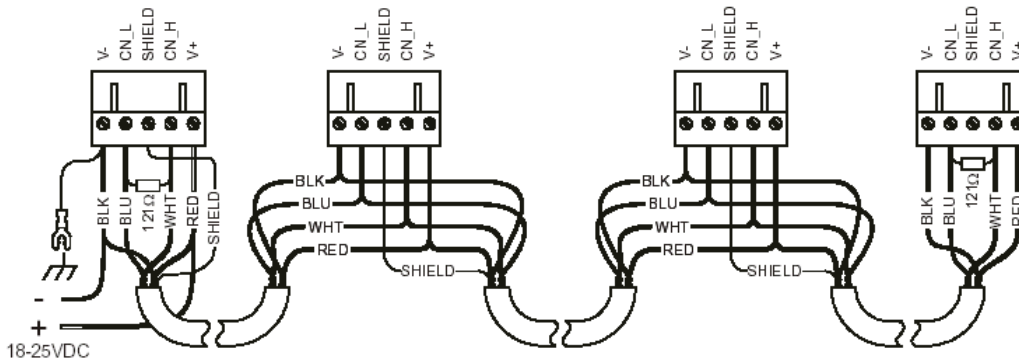
- Now the traffic lights should indicate what Mode the  $i^3$  is in.
- With the ID set correctly and communicating with the software it is now possible to go back into the IO configure menu and Auto-Config the  $i^3$  to match the  $i^3$  connected correctly.

## CAN Network Wiring



CAN Port Pin Assignments				
Pin	Colour	Signal	Signal Description	Direction
1	Black	V-	CAN Ground	–
2	Blue	CN_L	CAN Data Low	In/Out
3		SHLD	Shield Ground	–
4	White	CN_H	CAN Data High	In/Out
5	Red	V+	CAN Positive	–

Recommended Cable	
Thick: (Max Distance = 500m)	Belden 3082A
Thin: (Max Distance = 100m)	Belden 3084A

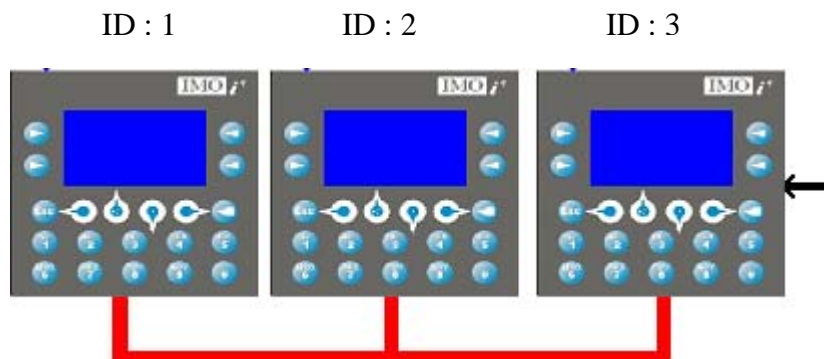


The CAN network is fixed to a baud rate of 125k. Using thick cable a maximum distance of 500m can be achieved with 64 nodes, by adding 1 repeater a distance of 1000m can be achieved, 2 repeaters 1500m, 3 repeaters 2000m and 256 nodes. The distance between nodes cannot exceed 6m (drop line length). A termination resistor of value 121Ω must be inserted at the start and end of the communication line. The maximum cable length is measured between terminating resistors or the node before the exceeding drop line length.

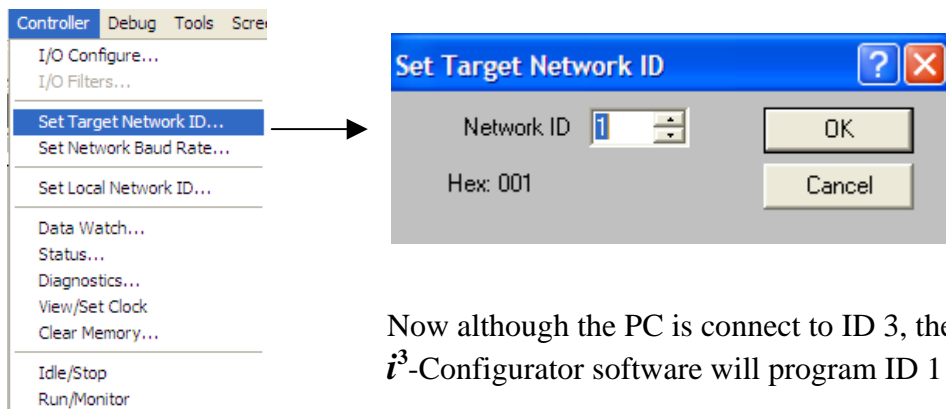
## Programming one *i*<sup>3</sup> through the CAN port of another

To program one *i*<sup>3</sup> through the CAN port of an *i*<sup>3</sup> connected through MJ1 to a PC you need to select the target ID of *i*<sup>3</sup> you wish to program.

Firstly set the ID of the *i*<sup>3</sup> though the main system menu. Each *i*<sup>3</sup> on the network has to have a unique ID.



Now with the *i*<sup>3</sup>'s connected select the Target ID from the controller menu and enter the ID number you wish to program.

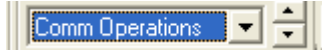


Now although the PC is connect to ID 3, the *i*<sup>3</sup>-Configurator software will program ID 1

This has the advantage of being able to program and monitor an *i*<sup>3</sup> through the CAN port with the GSM modem or Ethernet installed, as these communication options utilise MJ1.

## CAN Ladder Instructions

The CAN communication functions are location in the Comm Operations menu of the Ladder functions.



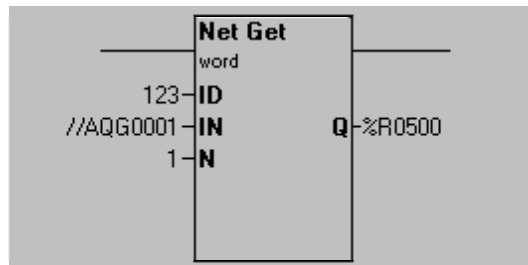
For the sharing of data there are four functions.



1. Network Get
2. Network Put
3. Get network Heartbeat
4. Send Network Heartbeat

### Net Get Words

This function allows global data from any device on the network to be copied into any set of registers. If the device defined by the source ID has not transmitted data, this function block will not pass power flow and will send a request for the data to be sent. Once the requested data has been received, power flow from this function block will turn on.



**ID** - This register or constant defines the source node for the global data. If the ID is not valid, the function will do nothing and will not pass power.

**IN** - This defines the starting point for the requested global data. This can be a %AQG or %QG register. Note that %QG registers must be on a word boundary (1, 17, 33...). This is a network register, a register assigned and produced by the transmitting ID.

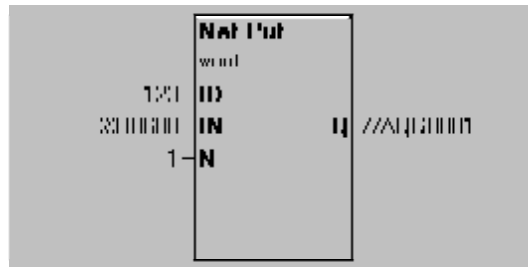
**N** - This defines the number of words to get from the source ID. The valid range is 1 to 32.

**Q** - This defines the starting register for the destination of the data. This is a register in the local controller.



## Net Put Words

This function allows sending global data using multiple networks IDs. When the "Send on Change of State" option is not checked, this function will copy the data from the source registers and attempt to transmit the data every scan that this function receives power. When the "Send on Change of State" option is checked and the SEND trigger is set LOW while the function receives power, this function will copy data from the source registers and attempt to transmit the data only if it changes. When the "Send on Change of State" option is checked and the SEND trigger is set HIGH while the function receives power, data will be sent every scan that the function receives power.



The function passes power if the ID is legal and in the range defined by the network ID and the total number of ID assigned to that node.

**ID** - This is a register or constant for the ID to use when transmitting data on the network. It must be in the range defined by the primary network ID and the total nodes allocated for this target.

**IN** - This is the starting register for the source data to send on the network. This is a register local to the controller.

**N** - This is the number of words to send on the network.

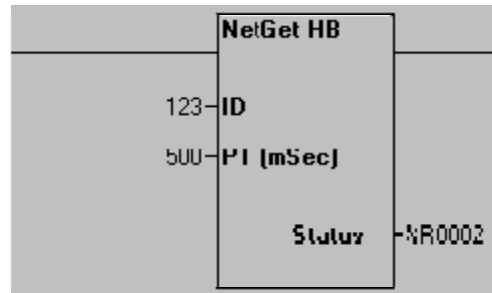
**Q** - This is the starting register for the destination of the data. Note that %QG registers must be on a word boundary (1, 17, 33...). This is a network register assigned to the network ID.

**SEND** - If the option to "Send on Change of State" is checked within the Net Put configuration, a Send trigger must be configured. This register, when high, forces a data transmission and ignores the Change of State.

### Net Get Heartbeat

This function allows the detection of a network heartbeat from another *i*<sup>3</sup>. This function does not generate any network traffic.

This function will not pass power flow if the ID is not in the legal range or if the device being monitored does not send a heartbeat message in the timeout defined by PT.



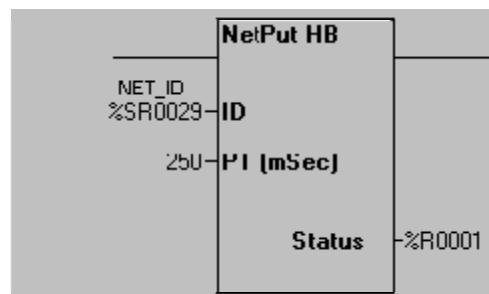
**ID** - This is a register or constant defining the ID of the device to monitor for a heartbeat.

**PT** - This is the maximum amount of time to wait for the heartbeat from the monitored *i*<sup>3</sup>. This timeout should be greater than the rate the device is sending heartbeat messages. Depending on network traffic and scan rates the GET timeout should be 10 to 1000 milliseconds greater than the PUT. This has a range of 1 to 6553 milliseconds.

**Status** - This register is currently used for internal record keeping. Do not allow other functions to write to this register.

### Net Put Heartbeat

This function allows the *i*<sup>3</sup> to transmit a heartbeat iCAN message at a given rate to indicate to other devices it is on-line and operating normally. This function does generate network traffic. The message generated normally does not affect bandwidth, but if many devices send heartbeat messages frequently it may cause reduction in bandwidth.



This function will not pass power flow if the ID is not in the legal range.

ID - This register or constant is usually the primary network ID of the device (%SR29), but can be in the range defined by the primary network ID and the total number of IDs assigned to this device.

PT - This is how often in milliseconds to send the heartbeat message. This has a range of 1 to 6553.

Status - This register is currently used for internal record keeping. Do not allow other function to write to this register.

## Programming Example

In this programming example we will have two *i<sup>3</sup>*'s connected on a CAN network sharing information. This will demonstrate the all of the functions listed in the previous chapter.

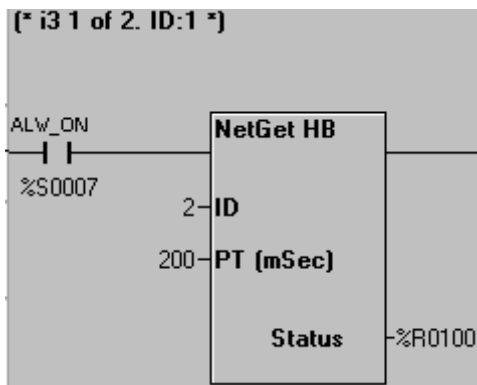


### *i<sup>3</sup>* ID:1

Before programming please ensure the ID of the first *i<sup>3</sup>* is set to 1.

In the first have the user being able to press the function keys and operate the outputs on the second. Unit 2 will be set to send a heart beat which unit 1 must receive in order to ensure that the second unit is operational.

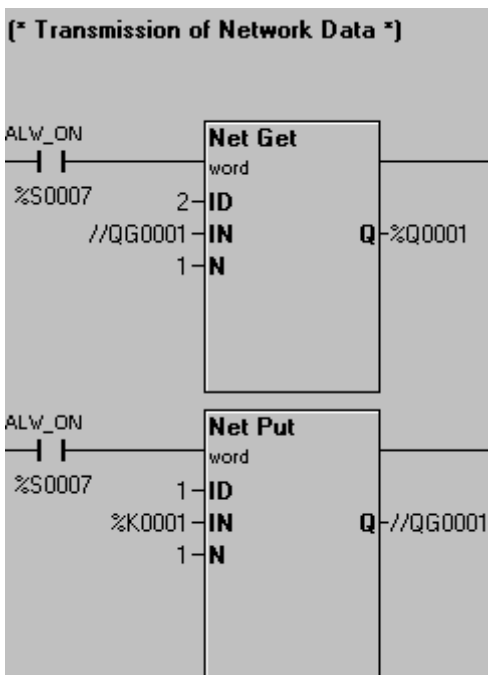
### Ladder Programming



We will begin by inputting a NetGet HB function to get the heartbeat created by station ID:2.

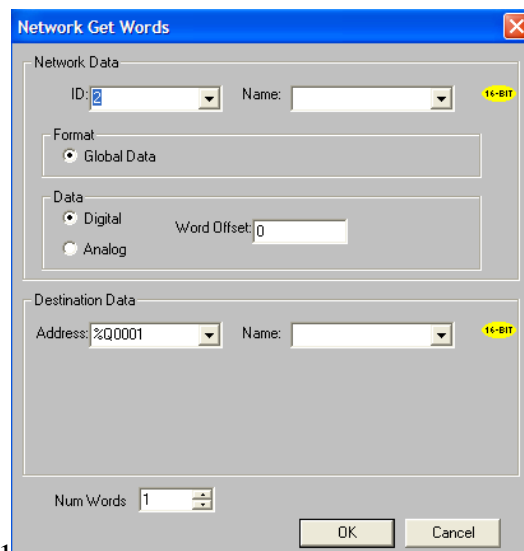
The ID will be 2, the timeout will be 100ms (needs to be greater than the heartbeat pulse of 100ms) and the status can be stored in %R100.

Next we need to enter the ladder logic for Getting and Putting data onto the network.



Enter a Net Get function on a rung with a NO always on contact. Set the ID to 2

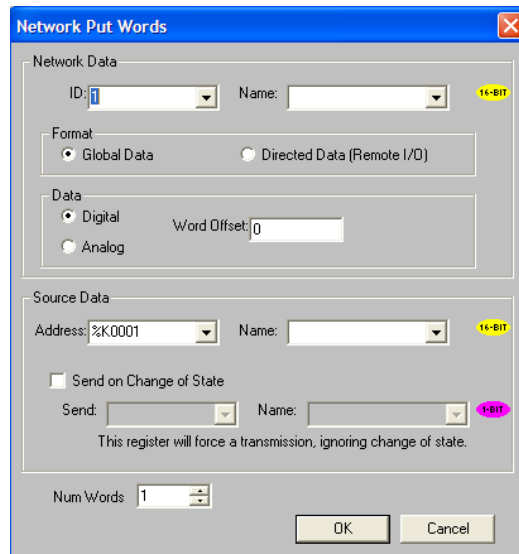
Set the data to digital and the word offset to 0, as it is the first data to be sent by ID:2. The number of words will be 1.



For the network put, set the ID to 1 as that is where we are sending the data from. Set the format to Global data and digital, with the offset 0.

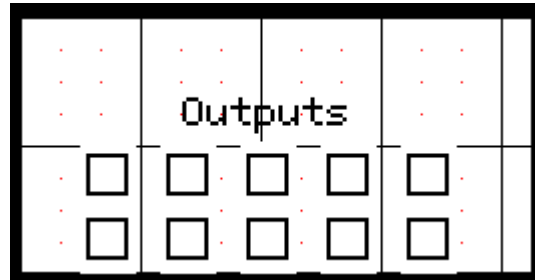
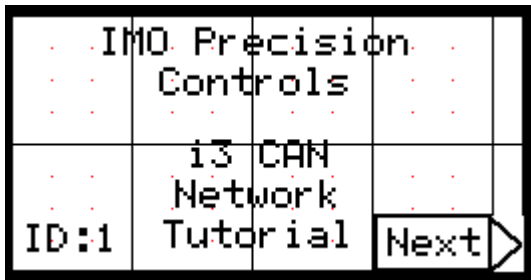
The source of the data will be %K001 (this will point to the start of a 16-bit register to send)

The number of words will be 1.



### Screen Editor Programming

The screens for both the programs will be very similar. There will be two screens, a main entry displaying the tutorial name and the  $i^3$  ID number with a screen jump function to take the user to a menu that will allow the user to see the outputs. Configure the screen jump button to go to screen 2 and allow the ESC button to return to screen 1.



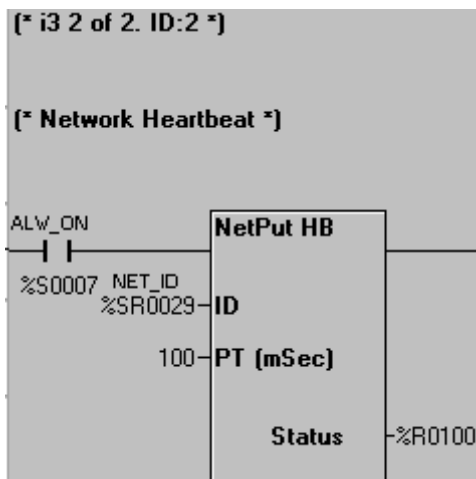
Configure the lamps on screen 2 to display the status of the outputs %Q1- %Q10.

## *i*<sup>3</sup> ID:2

Before programming please ensure the ID of the second *i*<sup>3</sup> is set to 2.

In the second *i*<sup>3</sup> we will have similar with the user being operating the first *i*<sup>3</sup> outputs by pressing the function keys on the first *i*<sup>3</sup>, however the Network Put function will be operated on a change of bit. Unit 2 will also send a heartbeat that will be received by unit 1.

### Ladder Programming



The input to the NetPut Heartbeat is a NO always on contact.

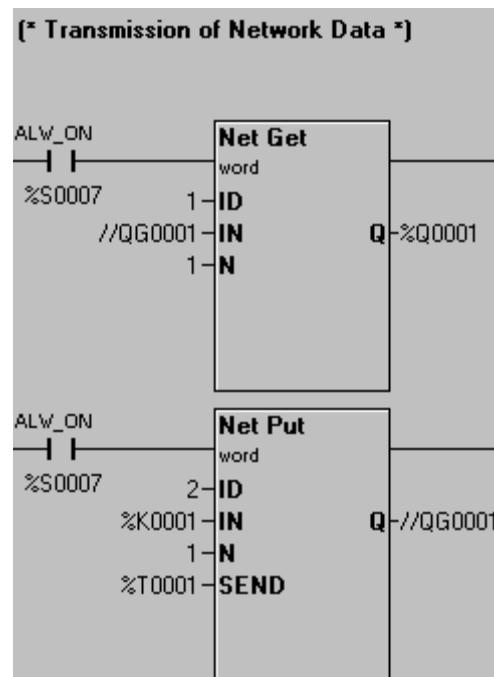
Set the ID to  $\%SR29$  the network ID, and timeout to 100ms (half of the Heartbeat get function)

Store the status in  $\%R100$ .

Enter a Net Get and a Net Put similar to that of the ID:1

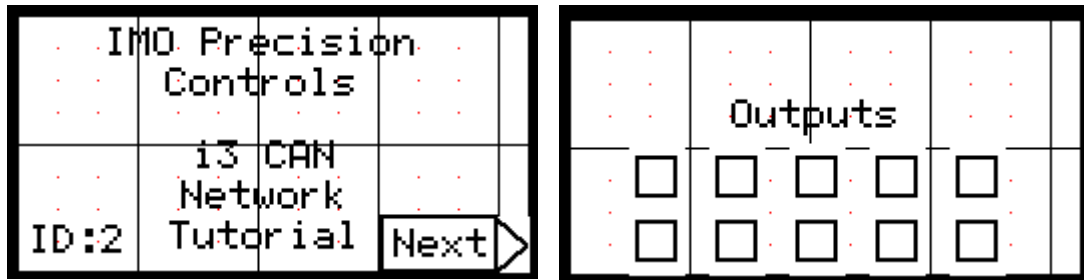
The data of the function blocks will be the same but with the ID's just reversed, i.e. Getting from ID1 and Putting from ID2.

The only other difference is that the NetPut has the option of Send on Change checked. This option reduces the amount of traffic on the network.



## Screen Editor Programming

There will be two screens, a main entry displaying the tutorial name and the  $i^3$  ID number with a screen jump function to take the user to a menu that will allow the user to see the outputs. Configure the screen jump button to go to screen 2 and allow the ESC button to return to screen 1.



Configure the lamps on screen 2 to display the status of the outputs %Q1- %Q10.

## Running the Program

With the  $i^3$ 's programmed and the CAN port connected the user will now be able to see that when the function keys are pressed on ID:1, the output lamps on ID:2 illuminate, and vice versa.



**IMO Precision Controls Limited**  
 1000 North Circular Road  
 Staples Corner, London  
 NW2 7JP United Kingdom  
 Tel: +44 (0)20 8452 6444  
 Fax: +44 (0)20 8450 2274  
 Email: [imo@imopc.com](mailto:imo@imopc.com)  
 Web: [www.imopc.com](http://www.imopc.com)



**IMO Jeambrun Automation SAS**  
 165 Rue Jean Jaures,  
 94700 Maisons Alfort  
 Paris, France  
 Tel: +33 (0)1 45 13 47 05  
 Fax: +33 (0)1 45 13 47 37  
 Email: [info@imopc.fr](mailto:info@imopc.fr)  
 Web: [www.imopc.fr](http://www.imopc.fr)



**IMO Deutschland**  
 Für weitere Einzelheiten  
 zu IMO Agenten und  
 Distributoren in Ihrer Nähe  
 schreiben Sie bitte ein E-mail  
 an folgende Adresse:  
[imo@imopc.com](mailto:imo@imopc.com)



**IMO Italia**  
 Viale A. Volta 127/a  
 50131 Firenze, Italia  
 Tel: +39 800 783281  
 Fax: +39 800 783282  
 Email: [info@imopc.it](mailto:info@imopc.it)  
 Web: [www.imopc.it](http://www.imopc.it)



**IMO Canada**  
 Unit 10, Whitmore Road  
 Woodbridge, Ontario.  
 L4L 8G4 Canada  
 Tel: +1 905 265 9844  
 Fax: +1 905 265 1749  
 Email: [imocanada@imopc.com](mailto:imocanada@imopc.com)



- Cam Switches
- Din Terminals
- Drives
- Enclosures
- Fieldbus remote I/O
- Isolators & Switch Fuses
- MCB & RCD
- Motor Circuit Breakers
- Motor Control Gear
- Panel Meters
- Relays
- Signal Conditioning
- Sockets
- Timers
- Transformers & Power Supplies



- Drives
- Intelligent Terminals/HMI
- Limit Switches
- Photoelectric Switches
- PLCs
- Proximity Switches
- Temperature Controls



- Data Acquisition & Control
- Drives
- Intelligent Terminals/HMI
- Limit Switches
- Photoelectric Switches
- Proximity Switches
- PLCs
- Signal Conditioning
- Temperature Controls



- Lightguards
- Safety Limit Switches
- Safety Relays

All IMO products are tried, tested and approved to relevant international quality standards



- Jaguar VXM 0.37-500KW
- Jaguar VXSM 0.37-7.5KW
- Jaguar CUB 0.37-2.2KW



- Audible devices
- Chip-on-Board
- Device programmers
- LEDs & 7 seg. displays
- PCB Terminal blocks
- Relays - automotive
- Relays - power
- Relays - signal
- Switches