INDUSTRIAL CONTROL DIRECT
A B&D Industrial Company

TRAINING

i^3 ...Display, Control, Communicate

Tutorial

Remote I/O

IMO

Table of Contents

Introduction

The purpose of this tutorial is to demonstrate the use of remote Modbus ***i***OS modules, and *i*CAN/Modbus Smart I/O modules with the ***i***^{3}. For the ***i***OS modules there is a pre-written configuration utility that can be used as a stand-alone application, or it can be grafted into most existing programs. For the Smart I/O, a simple ladder program will be created to demonstrate the operation.

IMO remote I/O can add both analogue and digital I/O to a system. Available in Modbus, DeviceNet, Profibus or *i*CAN the Smart I/O units can be connected into various networks reducing I/O cabling and wiring time within the control panel, whereas, the ***i***OS modules are available only in Modbus.

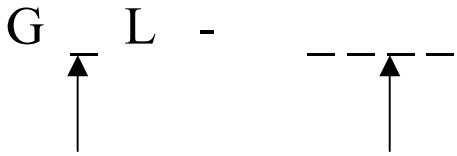
Six ***i***^{3} models have separate *i*CAN ports for direct connection to IMO Smart I/O products. Modbus Smart I/O can also be used with all ***i***^{3} models through one of the two configurable serial ports (MJ1 or MJ2). The structure for setting up a Modbus smart I/O remote module is the same as setting up a standard Modbus slave.

With the i3A12X/10D03-SCH, i3A12X/10B04-SCH and all of the i3B12Y range having three ports they could be used for data exchange. It is also possible to program the ***i***^{3} through the *i*CAN port.

In this tutorial:

- We will set up two examples for Smart I/O. The first example will be an example of an ***i***^{3} using *i*CAN remote I/O and the second will be of an ***i***^{3} linked to a Modbus Smart I/O module.
- We will run through how to use the ***i***OS configuration utility.
- We will also set up a Modbus comms version of the *i*Smart as a simple and cheap way to add extra I/O to an ***i***^{3}.

Smart I/O Part Numbers



Network

S – Modbus
D – DeviceNet
P – Profibus
C – iCAN

Digital

RY2A – 16-point relay output
TR2A – 16-point transistor output
TR4A – 32-point transistor output
D22A – 16-point PNP input
D24A – 32-point PNP input
DT4A – 16/16 input/ transistor output

Analogue

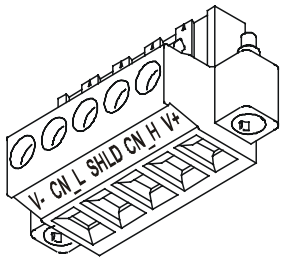
A21A – 12 analogue input
AT1A – 8 analogue input 4 analogue output

*i*OS Part Numbers

iOS / M 04 I P X - D1	Input - 4 Channel RTD
iOS / M 04 I V X - D1	Input - 4 Channel DC voltage
iOS / M 04 I T X - D1	Input - 4 Channel TC
iOS / M 04 I C X - D1	Input - 4 Channel DC current
iOS / M 08 I C X - D1	Input - 8 Channel DC current
iOS / M 08 I V X - D1	Input - 8 Channel DC voltage
iOS / M 08 I T X - D1	Input - 8 Channel TC Input
iOS / M 02 O X A - D1	Output - 2 Channel DC Voltage / Current
iOS / M 04 O X A - D1	Output - 4 Channel DC Voltage / Current
iOS / M 08 O X V - D2	Output - 8 Channel DC Voltage
iOS / M 08 B D R - D1	Input / Output - 4 digital inputs + 4 Relay outputs
iOS / M 12 I D X - D1	Input - 12 Channel Digital Inputs
iOS / M 12 B D D - D1	Input / Output - 4 digital inputs + 8 Transistor outputs

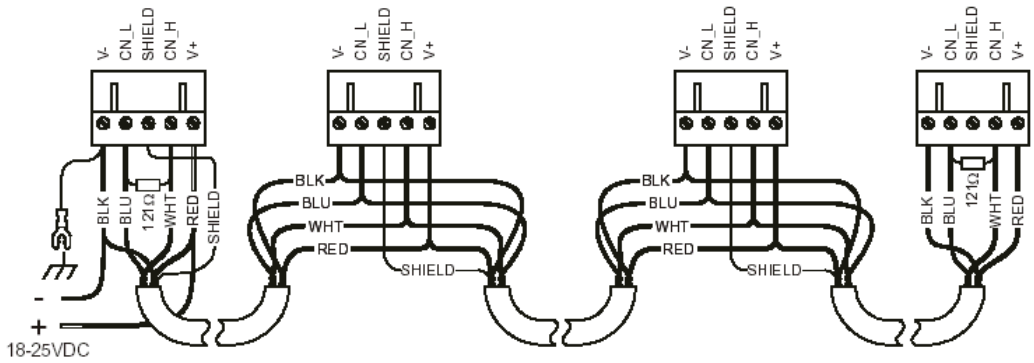
Please note:
Not all combinations are available from stock.
Please contact IMO for the latest information on availability.

Wiring - *i*CAN



*i*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	–

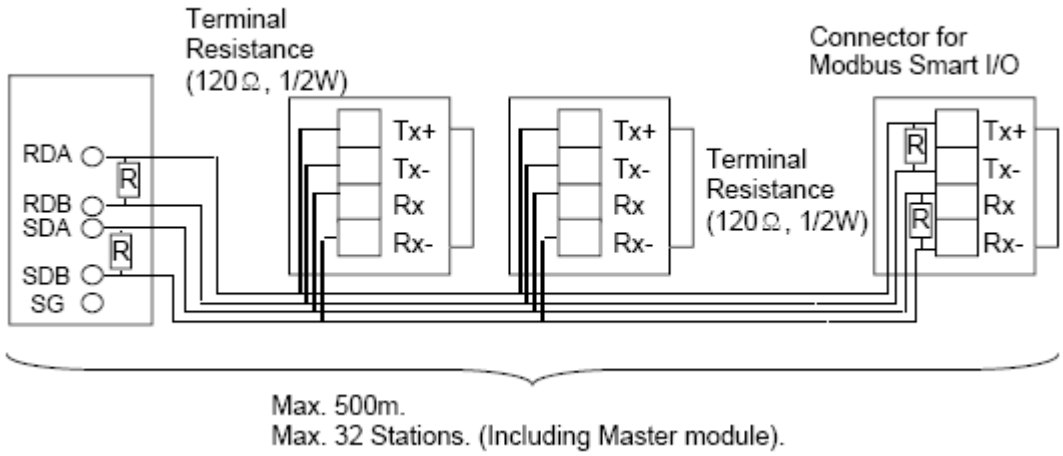
Recommended Cable	
Thick: (Max Distance = 500m)	Belden 3082A
Thin: (Max Distance = 100m)	Belden 3084A



The *i*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.

Wiring – Modbus

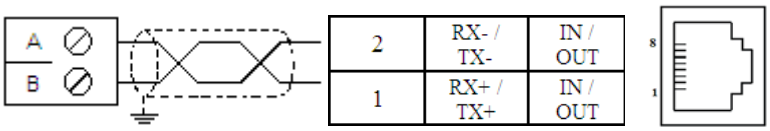
RS422 connection is shown below however a RS485 connection is also possible by linking Tx+ to Rx+ and Tx- to Rx-.



Examples

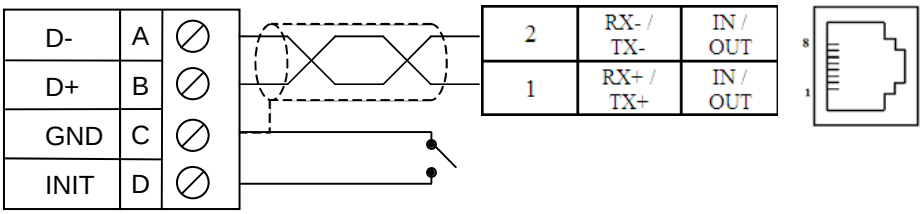
MJ1 two-wire RS485 comms to Smart I/O

Smart I/O ***i*³**



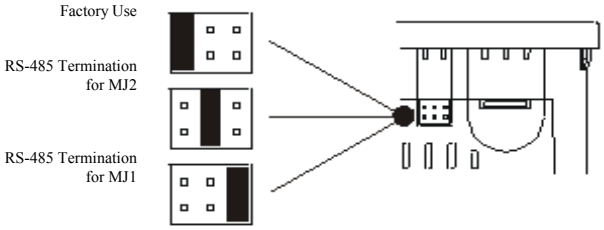
MJ2 two-wire RS485 comms to ***i***OS

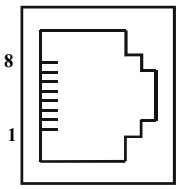
***i*OS** ***i*³**



*i*³ Port Connections

The *i*³ connection details of ports MJ1 and MJ2 are shown below. For the Modbus network you can use either RS422 or RS485 connections.





Pin	MJ1 Pins		MJ2 Pins	
8	TXD	OUT	TXD	OUT
7	RXD	IN	RXD	IN
6	0 V	Gnd	0 V	Gnd
5	N/C	N/C	N/C	N/C
4	CTS	OUT	TX-	OUT
3	RTS	IN	TX+	OUT
2	RX- / TX-	IN / OUT	RX-	IN
1	RX+ / TX+	IN / OUT	RX+	IN

Modbus Addressing

To address the Modbus Smart I/O, we need to map across an internal reference to a Modbus reference. The Digital inputs on a Smart I/O stick are read by reading the address 30001 and the outputs are written to by using the address 40001.

Slave Addressing

Modbus Code	Function code name	Modbus Address	Smart I/O Mapping	Remarks
04	Read input register (Read Input Registers)	3XXXX(word-input)	%IW0~%IW3	Word read
06	Write output register 1 word (Preset Single Register)	4XXXX(word-output)	%QW0~%QW3	Word write

i³ Master Modbus Map

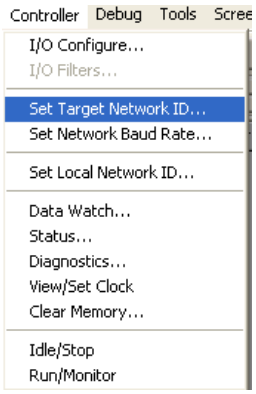
Modbus Master Mapping					
Internal Reference	**Maximum Range**	**Traditional Modbus Reference**	**Expanded Modbus Reference**	**Modbus Command(s)**	**Modbus Offset**
%Q1	2048	00001	000001	Read Coil Status (1) Force Single Coil (5) Force Multiple Coils (15)	00000
%M1	2048	03001	003001		03000
%T1	2048	06001	006001		06000
%QG1	256	09001	009001		09000
%I1	2048	10001	100001	Read Input Status (2)	00000
%IG1	256	13001	103001		03000
%S1	256	14001	104001		04000
%K1	256	15001	105001		05000
%AI1	512	30001	300001	Read Input Register (4)	00000
%AIG1	32	33001	303001		03000
%SR1	32	34001	304001		04000
%AQ1	512	40001	400001	Read Holding Registers (3) Preset Single Register (6) Preset Multiple Registers (16)	00000
%R	9999	(Previously 43001 for 2048 registers)	410001		03000
					06000
%AQG1	32	46001	406001		10000

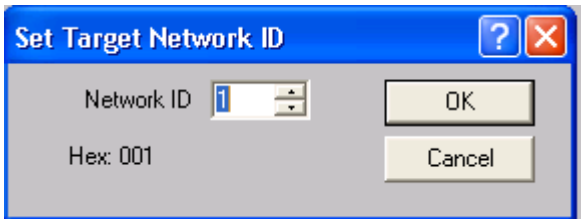
Connection to *i*CAN enabled i^3

An error may occur when trying to connect an i3A12X/10D03-SCH or an i3A12X/10B04-SCH to i^3 configurator. This is due to the i^3's station address not matching the target ID in the configurator software.

To check the station ID of the i^3 press both the UP and DOWN arrow together to enter the menu, then select the station number menu. Here we can check and change the station ID.

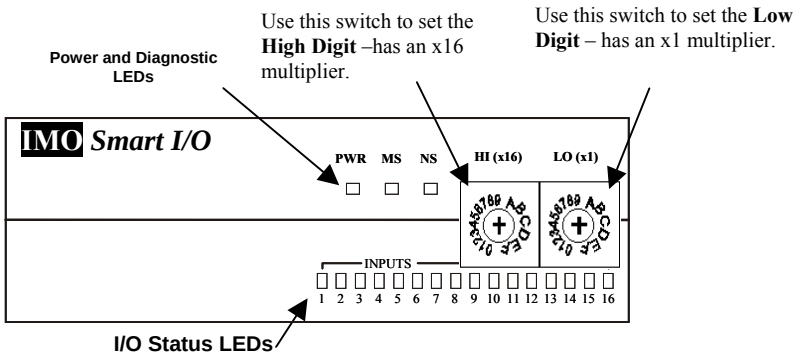
This target station number can also be checked and changed in the configurator software by selecting the "Set Target Network ID" from the controller drop down menu.





Setting the Station Number on the Smart I/O

Each node in the network must have a unique station address (number). This is set using the rotary switches on the unit, where there is a High digit and a low digit rotary knob used to set a hexadecimal number for the node. The master will use the station number to address the node in the main program.



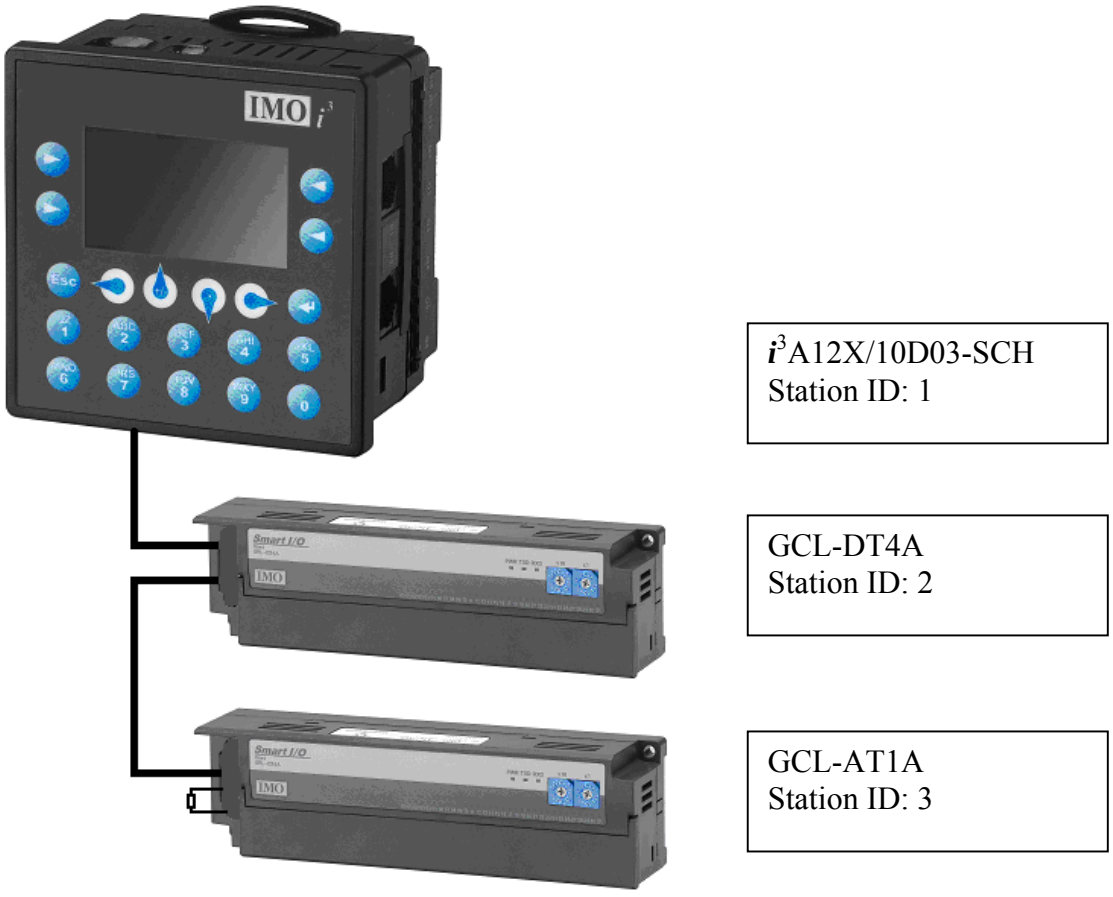
*i*CAN Remote I/O Programming Example

Program a small tutorial to demonstrate the CAN digital and analogue remote I/O configuration.

Configure one screen to demonstrate CAN network analogue remote I/O, another to demonstrate the digital remote I/O.

The example program is CAN_remote_IO.csp

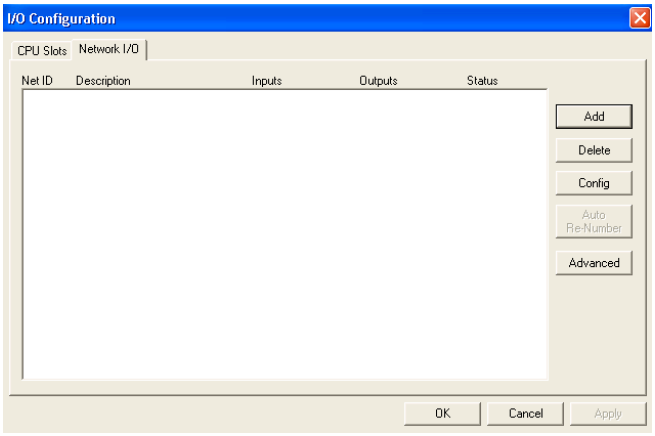
CAN Network Diagram



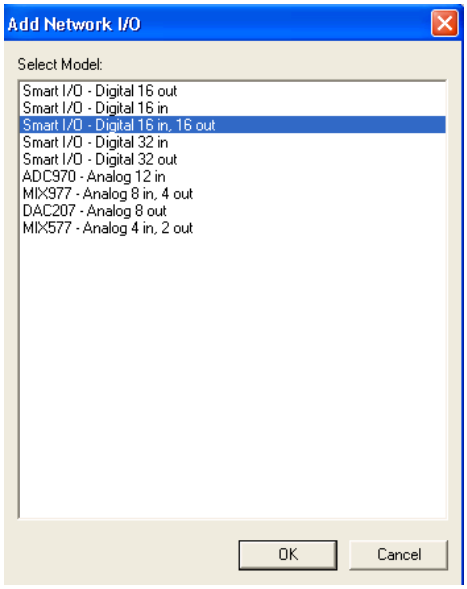
Connect the i^3 in the network as shown above, following the wiring from the section Wiring – CAN.

Configuring the iCAN Network

With the ***i***^{3} connected to the PC enter the I/O configuration menu and Auto-Config the unit. Then select the Network I/O tab to configure the network.



To add a node to the network, click the Add button.



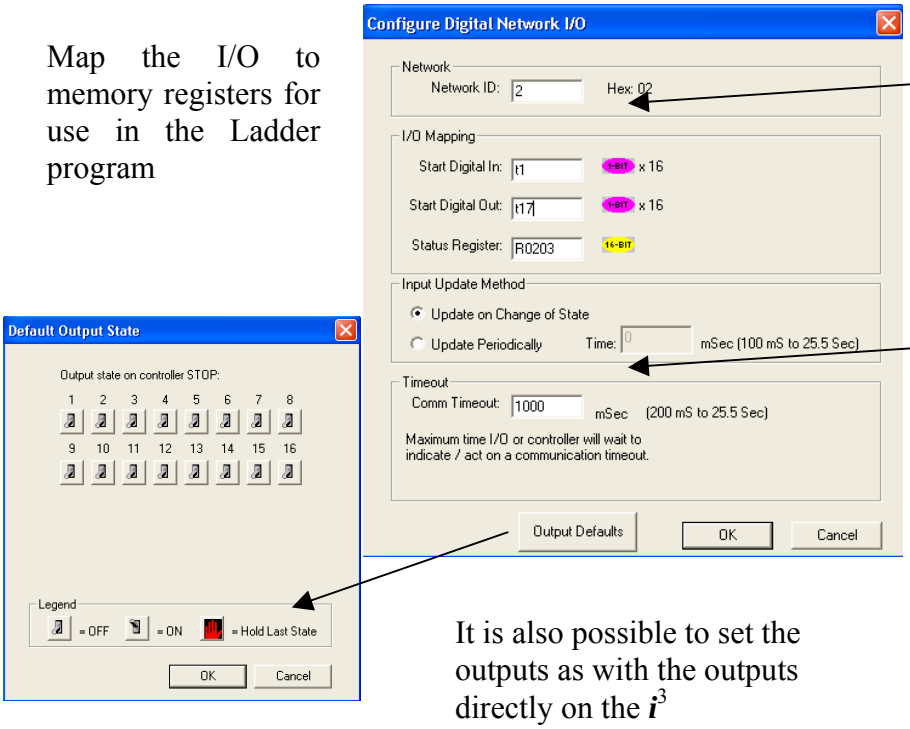
Currently available Smart I/O units are listed.

Select the smart I/O unit that you wish to insert in the network and click ok.

Configuring iCAN Remote Digital I/O

After selecting the node to insert, the I/O has to be mapped to registers in the ***i***^{3}.

Map the I/O to memory registers for use in the Ladder program



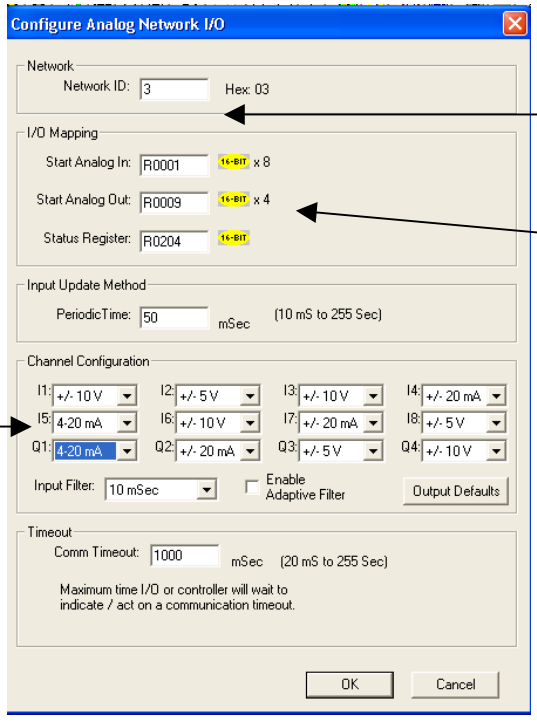
Set the network ID to match the node address set on the Smart I/O.

The input update can also be configured to suit

It is also possible to set the outputs as with the outputs directly on the ***i***^{3}

Configuring iCAN Analogue Remote I/O

The analogue Smart I/O units are set up in a very similar way but more registers are required (one register per Input or Output). With the very high baud rate used in a CAN network the update response can be very quick, almost instantaneous.



Set the ID of the node to match the rotary switch

Map the I/O to corresponding registers to be used in the program

Configure the analogue I/O as required: 0-10V, 0-5V, 0-20mA or 4-20mA.

Ladder Logic Programming

The ladder logic programming utilises the registers set up in the configuration of the network when the I/O was mapped to internal registers. The analogue values will be scaled to an internal memory register i.e. %T, %R or %M as with the analogue I/O direct to the ***i***[3].

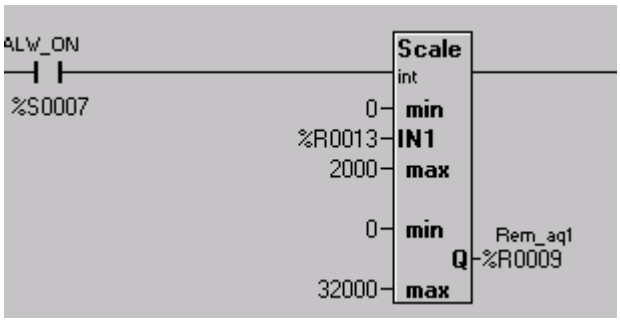
We are going to set up the logic to stimulate the first three outputs on the digital remote I/O block and scale the first analogue output so that the user can enter a value through the screen.

Insert three NO contacts on three separate Rungs, assign the contacts to %M1, %M2 and %M3. On each Rung add a N/O coils and assign to %T17, %T18 and %T19.

In effect %M1 will operate the first output on the digital remote I/O station. With %M2 and %M3 operating the second and third outputs.

Now we have to set up a scaling function for the analogue output, which will allow the user to enter in a more meaningful value.

Start a new Rung with a NO contact assigned to %S7 "Always On" (ALW_ON). Select the scaling function from the Advanced Maths operations menu and insert it in the Rung.



The user will enter a value, scaled between 0 and 2000 that will represent a 0-20mA output on the first analogue remote output, mapped to %R9.

Screen Editor Programming

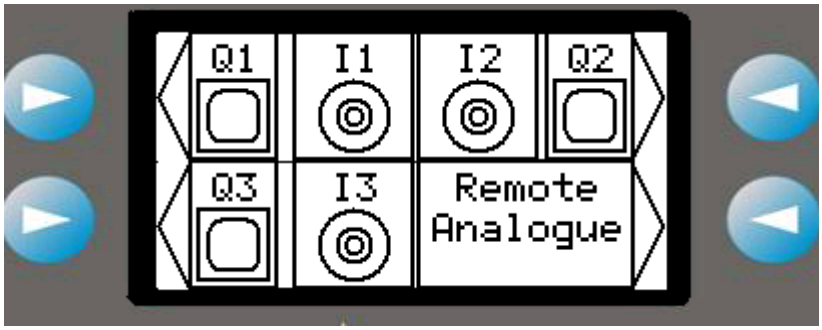
The HMI of the ***i***^{3} will have a user menu where the user can view and operate the digital remote I/O and a second screen where they can output to the first analogue output and read the first analogue input.

Screen 1

Assign three of the soft keys to the touch switch function  , so that on a momentary push they will operate %M1, %M2 and %M3. Give them the legends %Q1, %Q2 and %Q3 accordingly. Assign the forth soft key to a screen jump function  to jump to screen 2, give it a legend to inform the user what screen it is going to.

Finally place 3 lamps  on the screen assigned to the first three remote digital inputs, %T1, %T2 and %T3. Give them legends accordingly to inform the user.

Screen 1 should now resemble the screen below.



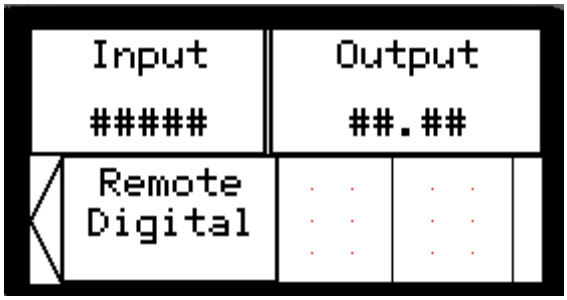
Screen 2

Screen 2 will display the first analogue remote input and allow the user to enter data that will be scale the first analogue remote output.

Insert two data numeric function tags  , the first to only display the first analogue input and the second editable so the user can operate the first analogue output.

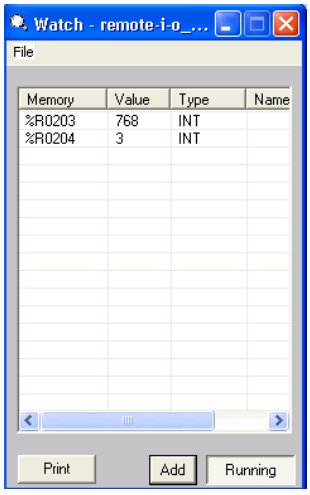
Assign the first numeric function to %R1, the mapped first remote analogue input. Make the function non-editable and enter the legend “input”. Assign the second numeric function to %R9 the register that is going to be scaled to the output. Set the digits to 4, decimal points to 2 and the limits to 0 to 2000. Enter the text “output” in the legend. This will allow the user in effect to enter a value that appears to be 0 to 20mA with a two decimal place precision.

Finally assign a soft key to a screen jump function  to go back to the first screen. Enter a legend of “Remote Digital” to inform the user where it takes them.



Running the Program

Once the program has been downloaded and the ***i***^{3} put into run mode the user should be able to press one of the soft keys and see an output lamp light on the remote digital I/O. Similarly if an input is triggered on the remote I/O the associated lamp should come on in screen 1. The user will also be able to monitor the analogue remote I/O through screen 2.



If there is a problem with the remote I/O network then the user can view the status registers using the data watch function.

As well as the ***i***^{3} having a status register for the network the remote I/O terminals have status LED's to provide a visual indication of the network's status.

Diagnostic LED	State	Meaning
MS Indicates fault status of module	Solid Red	RAM / ROM test failed
	Blinking Red	I/O Test failed
	Blinking Green	Module is in Power-up state
	Solid Green	Module is running normally
NS Indicates fault status of network	Solid Red	Network Ack or Duplicate ID test failed
	Blinking Red	Network ID test failed
	Blinking Green	Module is in life expectancy default state
	Solid Green	Network is running normally

Modbus Smart I/O Programming Example

Program a small example to demonstrate the ***i***^{3} controlling a network of 3 remote I/O blocks, GSL-DT4A, GSL-D24A and a GSL-RY2A.

The tutorial will concentrate on the network configuration and therefore the ladder logic program will just consist of an incrementing counter on a second pulse.

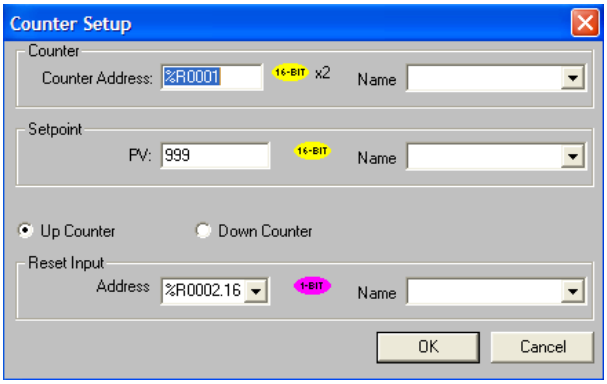
The example program is Modbus_remote_IO.csp

Modbus Network Diagram

Ladder Logic Program

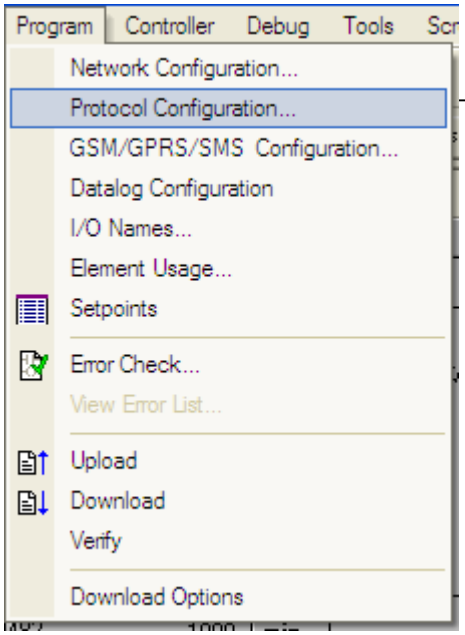
The ladder program will only consist of an up counter that will reset itself when it reaches the preset value. The counter will be operated on the internal one-second flag.

Insert a NO contact ⊣⊢ on the first rung, followed by a Counter function block CNT. Set up the counter function block as follows:

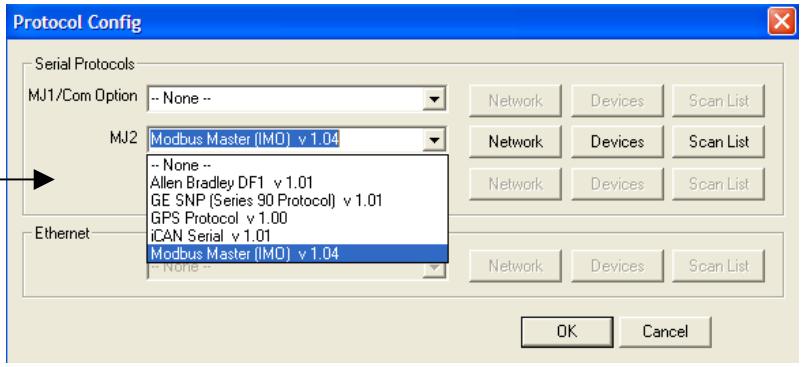


Configuring the Modbus Smart I/O Network

To configure the port MJ2 select the menu Protocol Config from the program drop down menu.

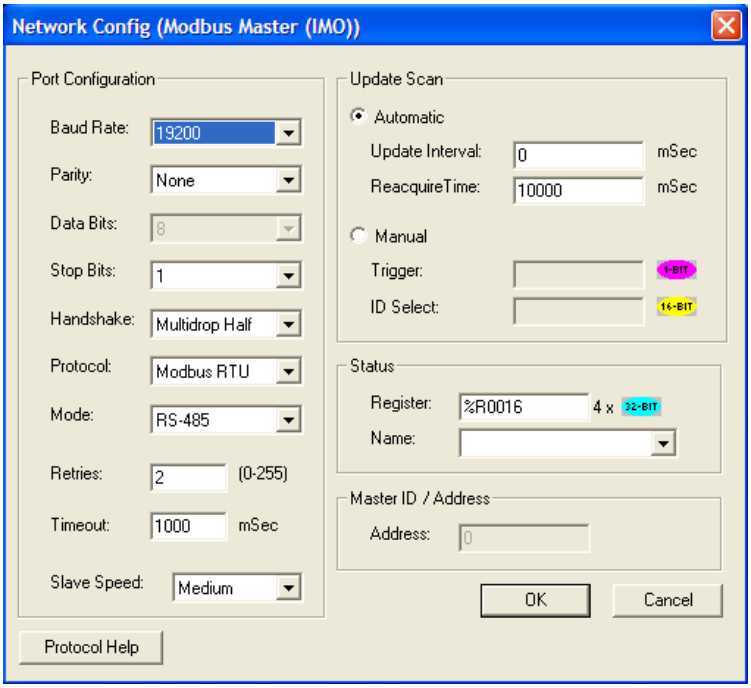


Select the protocol Modbus Master (IMO) V1.04. Then click the network button to set the network parameters.



Network Configuration

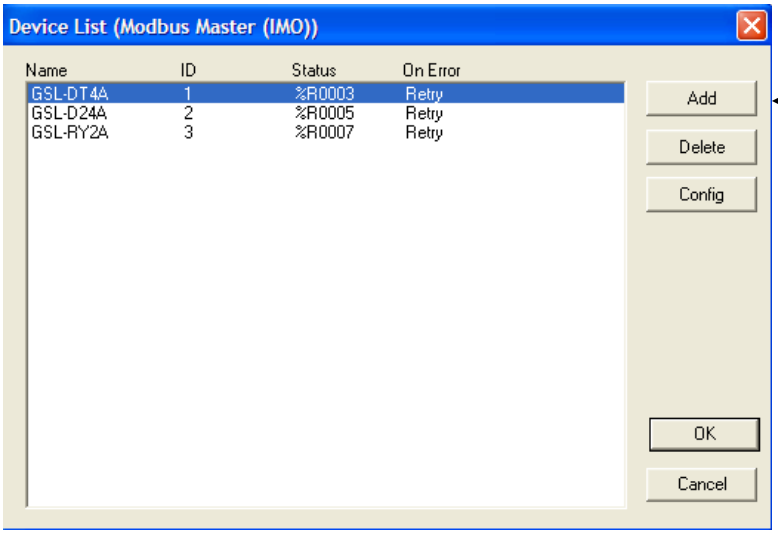
It the network configuration the communication parameters are set. Set the parameters as shown below. We are going to use the protocol Modbus RTU and communicate on RS485, which is a multi-drop half duplex system.



It is important that the Slave speed is set to a maximum of Medium for good communications. A status register is not necessary but very useful for debugging.

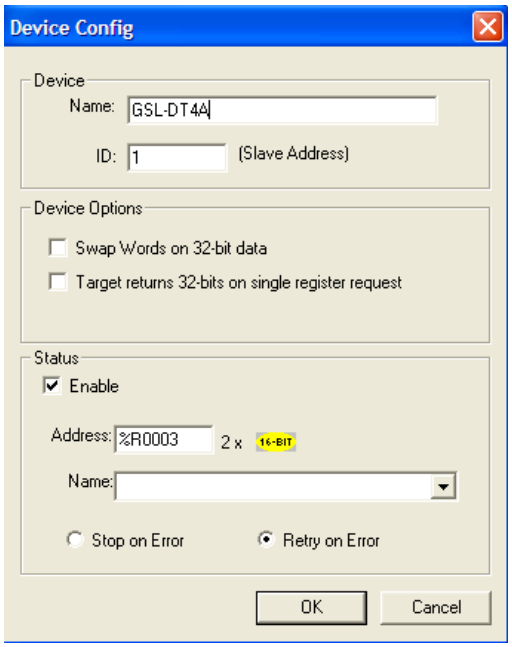
Communication Devices in the Network

Once the Network parameters have been set, the Devices need to be added to the network.



To insert a new device, click Add.

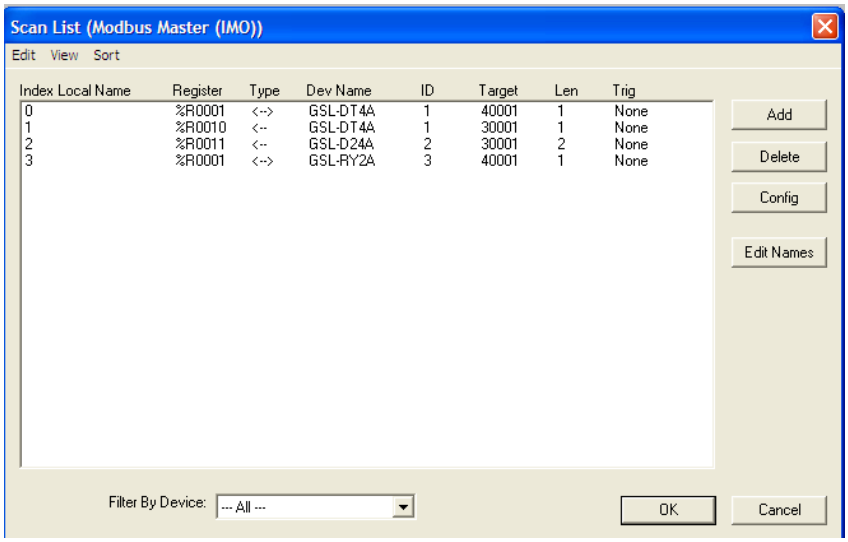
Name the device and give the device an ID. This ID should be unique and match the number on the rotary dial on the front on the Smart I/O.



Add the three individual devices as shown and click OK. Ensure that the ID has also been set on the front of the Smart I/O before powering up.

Scan List

Finally we need to add to the scan list the addresses we are going to read from and write to the Modbus slaves.



This is where we map across from the Modbus reference to the $\boldsymbol{i}^3$ memory reference. Once we have mapped across the references, we only need to consider the $\boldsymbol{i}^3$ memory reference, i.e. for the 16 relay output module we have mapped them across from ID 3 address 40001 to %R0001, if we want to switch on the individual bits on that output module then we use the bit reference of the word register, i.e. %R0001.1 is output 1.

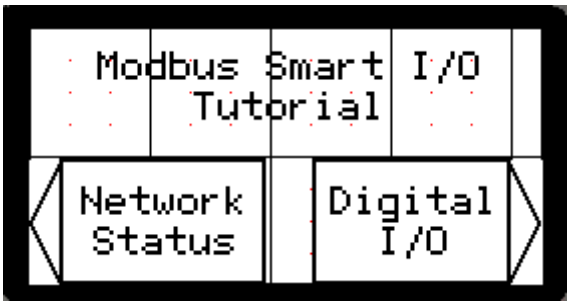
Now that the communication network has been set up we can program some screens.

Screen Editor Programming

For this program we are going to have 5 screens: Main Menu, a Communication Status and 3 screens, 1 for each Smart I/O module.

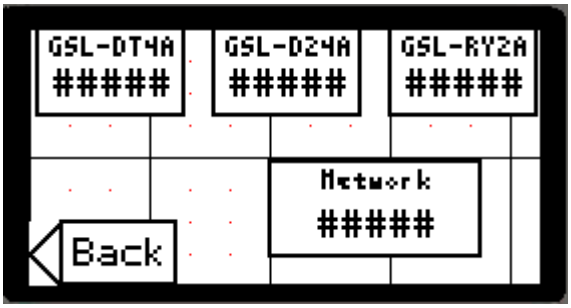
To enter the screen editor, click on the icon  . Please set up the screens as described below.

Screen 1



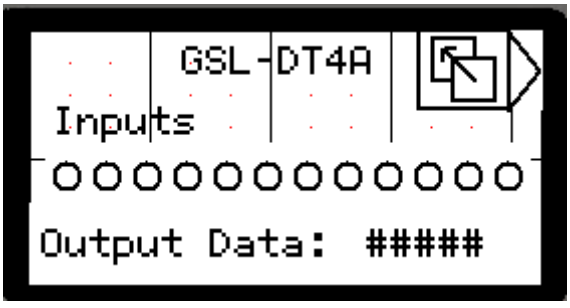
Enter some static text at the top with two jump screen buttons below it. One button to take the user to the Network status information (screen 2) and the other to move to the start of the digital I/O (screen 3).

Screen 2



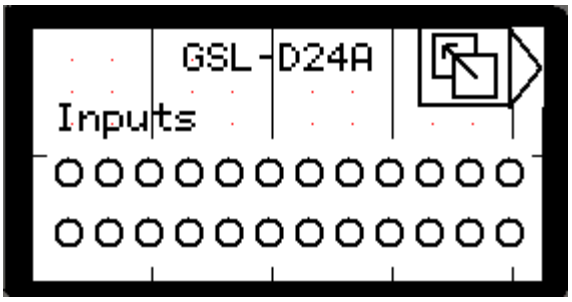
On the Network Status page, there will be 4 numeric data fields. One for each device and then a forth for the entire network. The numeric data fields will show the register value as set in the Network configuration. Add a screen jump to go back to the main screen (1)

Screen 3



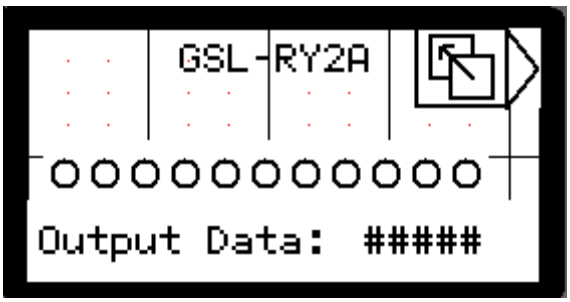
The first of the Smart I/O blocks will be the GSL-DT4A. Add some lamps to represent the inputs from the module and add a numeric field to represent the output data. The I/O will be as we mapped in the Scan List. Inputs: %R10, Outputs %R1. Add a screen jump to move to the next block (screen 4)

Screen 4



The second of the Smart I/O blocks will be the GSL-D24A. This block is inputs only so add some lamps to represent this. The Inputs are mapped in the Scan List. Inputs word: %R11 & %R12, Add a screen jump to move to the next block (screen 5)

Screen 5



The final of the Smart I/O blocks will be the GSL-RY2A. This block is outputs only but we will use lamps and a numeric data field to indicate. The Outputs are mapped in the Scan List. Output word: %R1. Add a screen jump to move back to the main screen (1).

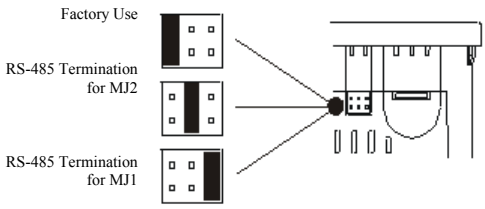
Running the Program

Please connect the network as shown and remember to set the station number on the rotary dials on the Smart I/O bases.

The communication LED's (Tx and Rx) on the Smart I/O stations should be flickering almost constantly when good communication is achieved.

Also remember to insert the jumper into the RS485 termination terminal on the ***i***^{3}.

Remember to connect the RS-485 termination jumper as shown



When all connected with communication established and in RUN mode the counter will directly output to the Smart I/O stations. The inputs can be monitored on the screens and the communication status of the network can be checked.

Please use the program: modbus_remote_io.csp

*i*OS Configuration Utility

Using the ***i***OS module has the following advantages:

1) Up to 16-bit resolution on analogue models.
2) Space-saving din rail mountable design.
3) Easy wiring with vibration resistant plug in connectors.
4) Jumper-free configuration using IMO's free configuration utility.

Step 1

Open the file below in ***i***^{3}-Configurator.

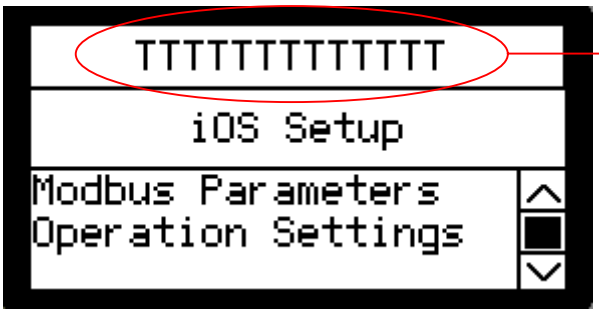
Connect the loader cable to the ***i***^{3}, run an I/O configuration to make the project compatible with the ***i***^{3} that you will be using. Then download to the ***i***^{3}.

Step 2

Wire the ***i***OS module to the ***i***^{3} as shown in the example in the wiring section of this tutorial, plugging into MJ2 of the ***i***^{3}. Put in the link between Init and Ground on the ***i***OS, and power it up.

Step 3

On the first screen displayed on the ***i***^{3}, press the soft-key corresponding to the 'INIT Default Setup'. The ***i***^{3} will then communicate briefly with the ***i***OS module; you should see the light on the ***i***OS flicker slightly. The next page appears:

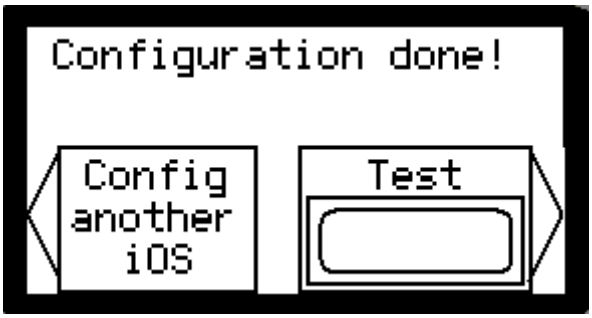


The part number of the connected unit should appear here. If it does not then comms have not been established. Try cycling power on both units, check the 'INIT' jumper and try again.

Once the correct part number is shown in the top field, use the cursor to select the following options:

1. Modbus Parameters – here you set the network ID, Baud, Parity, Stop and Data bits of the ***i***OS module ready for the ***i***^{3} program and application, along with Modbus RTU or ASCII setting.
2. Operational setting – here you configure the inputs and outputs. Enabled or disabled, voltage or current, RTD type, Thermocouple type etc.
3. Save Parameters – Save the changes to the ***i***OS module.
4. Exit without Save.

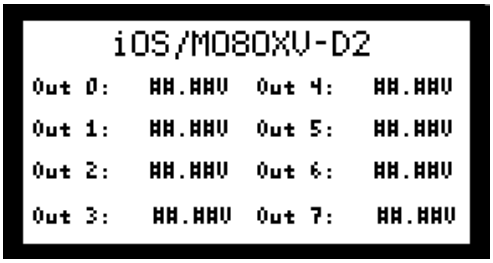
Step 4



If you selected 'Save Parameter' after altering your Operation Settings, then you will be brought to this screen. Here you can either select another ***i***OS to configure, or, test** the one you have just configured.

Note: you can only use the test function if the ***i*OS module is set to 19200, 8, 1, None, RTU. And all I/O options set for Voltage instead of Current.

Once the corresponding soft-key has been pressed to activate the Test Function, unplug the *i*OS from MJ2. Remove the 'INIT' link, and cycle the power. Then, plug the *i*OS into MJ1. By that time the screen corresponding to the previously configured unit should appear. ie.



In this case, alter the output voltages and measure with a multi-meter to see that they correspond.

Using a Modbus *i*Smart as remote I/O

To easily and affordably add extra I/O to an ***i*³** Controller; another option is to use an *i*Smart intelligent relay with Modbus Comms enabled. (Part No. SMT-CD-R20 or SMT-CD-T20 for relay or transistor outputs.)

This option for expanding the I/O in this way has several benefits:

1) This quickly adds 12 Digital inputs, 4 10-bit analogue inputs, 8 relay/transistor outputs. And can easily be expanded again, either with expansion modules as shown above, or, up to 31 iSmarts can be distributed over the Modbus network.
2) Two of the inputs are High Speed Counters (up to 1KHz).
3) One of the transistor outputs can be used for PWM. (up to 1KHz)
4) The HMI screen provides local information and I/O activity display.
5) Four programmable push buttons are added to the system.
6) Although no program is required in the *i*Smart - the ***i*³** Controller can control most variables over the comms – if the comms are lost then a program can be put into the *i*Smart to return equipment to a 'Safe' condition until comms is re-established.

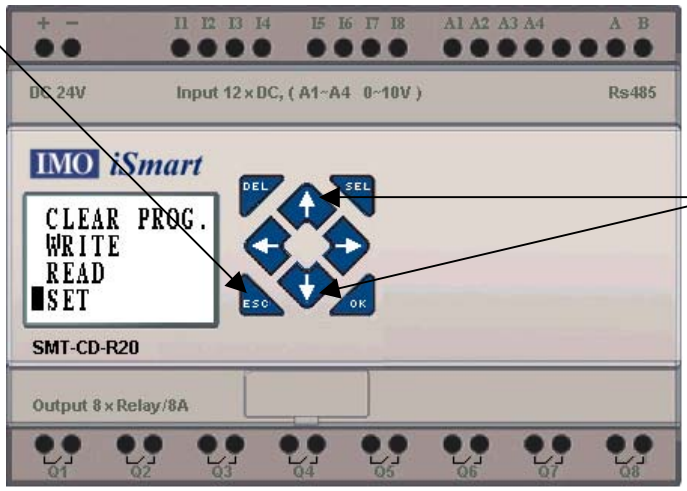
Open the attached program in ***i*³** Configurator: iSmart_remoteIO.csp

The attached program is for three iSmart Intelligent Relays communicating with an ***i*³** controller. Setting up each of the iSmarts is very simple. No program is required inside the iSmarts and they are not even required to be in 'RUN' mode if they are being purely as expansion I/O.

Power up each iSmart, if only one unit is available then do not worry, as the demo will still function.

SET System Settings

Press the 'ESC' key on the front of the iSmart. This will open the system menu shown.



Use the up and down arrows to scroll down to 'SET' and press 'OK'

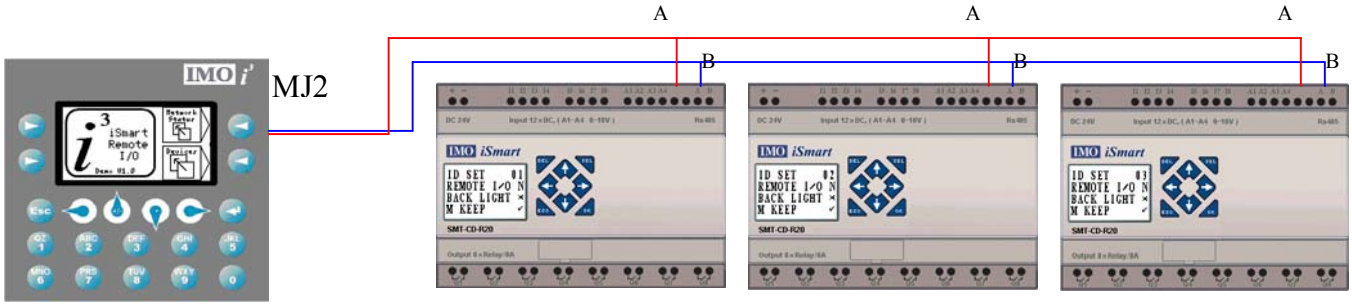
Inside the SET menu are 7 options, but we are only concerned with the first two.

ID SET 01	→	ID setting (00~99)
REMOTE I / 0 N	→	Remote I/O Mode (N: none M: Master S: Slave)
BACK LIGHT ×	→	Back light mode (√: always light ×: light for 5s after pressed.)
M KEEP √	→	M: non-Volatile (√:Volatile ×: Non- Volatile)
I/O NUMBER 0	→	Expansion I/O Points（0~3）
I/O ALARM √	→	Siren setting when is not available to Expansion I/O Points（√:Yes ×:No）
C KEEP ×	→	in stop/run switching, Counter Present Value Keeping（√:Yes ×:No）

For each iSmart 'REMOTE I/O' must be set to 'N'. The Master and Slave modes are for use only when connecting two iSmarts together, where the Slave becomes Expansion I/O to the master. We do not want this single point-to-point connection; we want the full multi-drop facility that is enabled by selecting 'N'.

Wiring to iSmarts

From the RJ45 connector of MJ2, use a twisted pair to loop through all of the iSmarts, connecting to the 'A' and 'B' Terminals (the furthest to the right along the top row). Also install a 120Ω termination resistor between the A and B terminals of the final Slave (ID3), and install the Jumper on the top of the ***i***^{3} for termination at the Master end for MJ2 (Middle position).

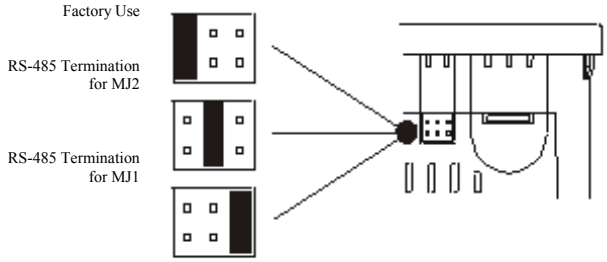


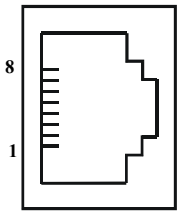
ID Set: 0 (Master)

ID Set: 1 (Slave)

ID Set: 2 (Slave)

ID Set: 3 (Slave)





Pin	MJ1 Pins		MJ2 Pins	
8	TXD	OUT	TXD	OUT
7	RXD	IN	RXD	IN
6	0 V	Gnd	0 V	Gnd
5	N/C	N/C	N/C	N/C
4	CTS	OUT	TX-	OUT
3	RTS	IN	TX+	OUT
2	RX- / TX-	IN / OUT	RX-	IN
1	RX+ / TX+	IN / OUT	RX+	IN

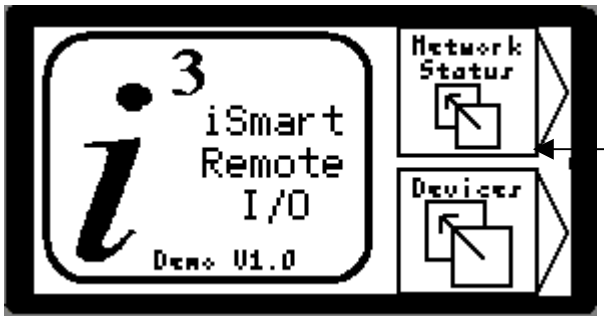
MJ2: PIN 1 ——— A

MJ2: PIN 2 ——— B

The system should then be ready to run.

Running the program

In the attached ***i***^{3} program there are three main areas in the HMI. The 'Main' screen branches into 'Network Status' and 'Devices', as shown below.

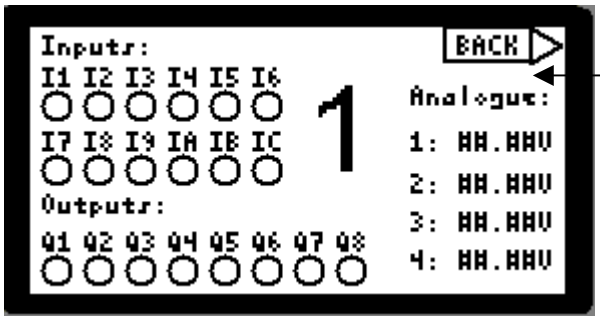


MAIN:

Allows the user to jump to either the Network Status screens, or to monitor the attached Devices.

NETWORK STATUS:

Here the user can see how the Modbus Network is performing. The **Interval** is the time in millisecond that the system takes to make a complete loop of the Scan List. If devices are missing or connected incorrectly then this interval will increase significantly.
The number of **Missing, Corrupt** and **Valid** responses are counted in 32-bit counters.
Three Text-Tables show when each **iSmart (1-3)** is communicating correctly – to help find any missing devices on the network.



DEVICES:

Each iSmart Device has its own screen to monitor the I/O. Inputs, Outputs and Analogues are displayed.

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
Web: 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
Web: 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

IMO Italia
Viale A. Volta 127/a
50131 Firenze, Italia
Tel: +39 800 783281
Fax: +39 800 783282
Email: info@imopc.it
Web: www.imopc.it

IMO Canada
Unit 10, Whitmore Road
Woodbridge, Ontario.
L4L 8G4 Cananda
Tel: +1 905 265 9844
Fax: +1 905 265 1749
Email: imocanada@imopc.com

PANEL

- 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

MACHINE

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

PROCESS

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

SAFETY

- Lightguards
- Safety Limit Switches
- Safety Relays

IMO JAGUAR DRIVES

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

PCB EMECH

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

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

nqa. ISO 9001 Registered UKAS QUALITY MANAGEMENT 016

3 YEAR GUARANTEE

5 YEAR GUARANTEE