Rev: Q 02 May 2002

CCU2/C-MXMB

Installation and Configuration Manual

TABLE OF CONTENTS

1. General

1.1. Introduction

The CCU2/C-MXMB provides a MODBUS interface to a number of MX panels on an MXNet. CCU/IO boards may also be connected to provide general I/O devices accessed through the MODBUS interface.

The CCU2/C-MXMB connects to one of the MX panels on the MXNet via a TLI-800 (TPI) interface card using RS232 (PL2 socket). It connects to MODBUS via either RS232, RS485 (default) or RS422 connection. Another port allows up to 8 CCU/IO boards to be connected.

Each CCU/IO has 8 relay outputs that can be used as inputs to the MX. These contacts are controlled via WRITE commands to the MODBUS map. Each CCU/IO also has 8 supervised inputs whose status can be read from the MODBUS map.

The MODBUS map of the CCU2/C-MXMB is configured using the **CCU-MXMB Setup** program. The MODBUS map can be accessed directly via the Global Map or through a submap:

- Global Map – Each MX panel and CCU/IO board is added to the map as a static item, the user gets to choose the offsets of the bits for each panel (viz Modbus addresses). Each MODBUS address points to the corresponding address in the Global Map
- Submap – Up to 1000 MODBUS bits can be configured individually by the user to point to any address in the Global Map.

1.2. Ordering Configurations:

There is only one configuration. Part Number: CCU2/C-MXMB-B. This configuration is within an aluminium box. The box is screwed to a mounting bracket to facilitate wall/panel mounting of the box.

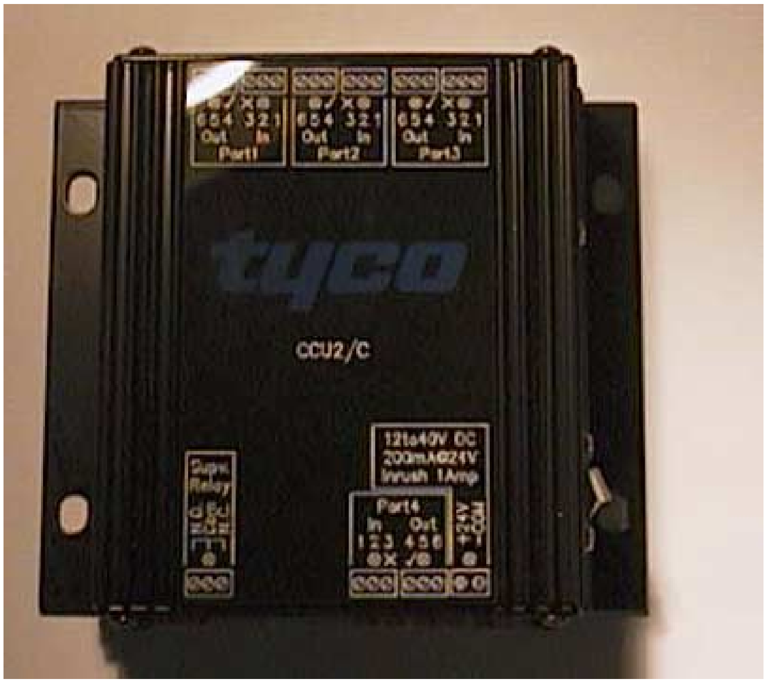


Figure 1:CCU2/C-MXMB-B

1.3. System Requirements for CCU2/C-MXMB Setup program

The minimum requirements for the computer system are:

- 15 Mb of disk space for installation.
- 16MB Ram
- SVGA Monitor
- Windows® 95, 98 or NT operating system
- Microsoft® Windows compatible pointing device (such as a mouse)
- A communications port for reprogramming the CCU2/C-MXMB.

Running the Setup.exe application will install the MXMB Setup program to Program Files\Tyco. A shortcut is added into the Start menu under Programs | Tyco | MXMB Programmer.

2. Installation

This section describes the how to physically install the CCU2/C-MXMB and provides no information on configuration.

2.1. Supplied:

1	x	CCU2001/MX board with firmware for MXMB (MODBUS)
1	x	Mounting Bracket
4	x	M4 Machine Screws
4	x	Flat Washers
1	x	Power connector for CCU2001/MX
1	x	Supervision relay connector for CCU2001/MX
1	x	interface cable from TLI-800 to CCU2/C-MXMB
1	x	Floppy disk with MXMB software

2.2. You will need:

1	x	24V DC plug pack or Power Source
1	x	Interface cable for MODBUS communication
1	x	Programming cable (same as MODBUS RS232 Interface cable)

2.3. Power Requirements:

Typical: 150mA at 24VDC   Voltage Input: 12-40 VDC

2.4. Board Layout:

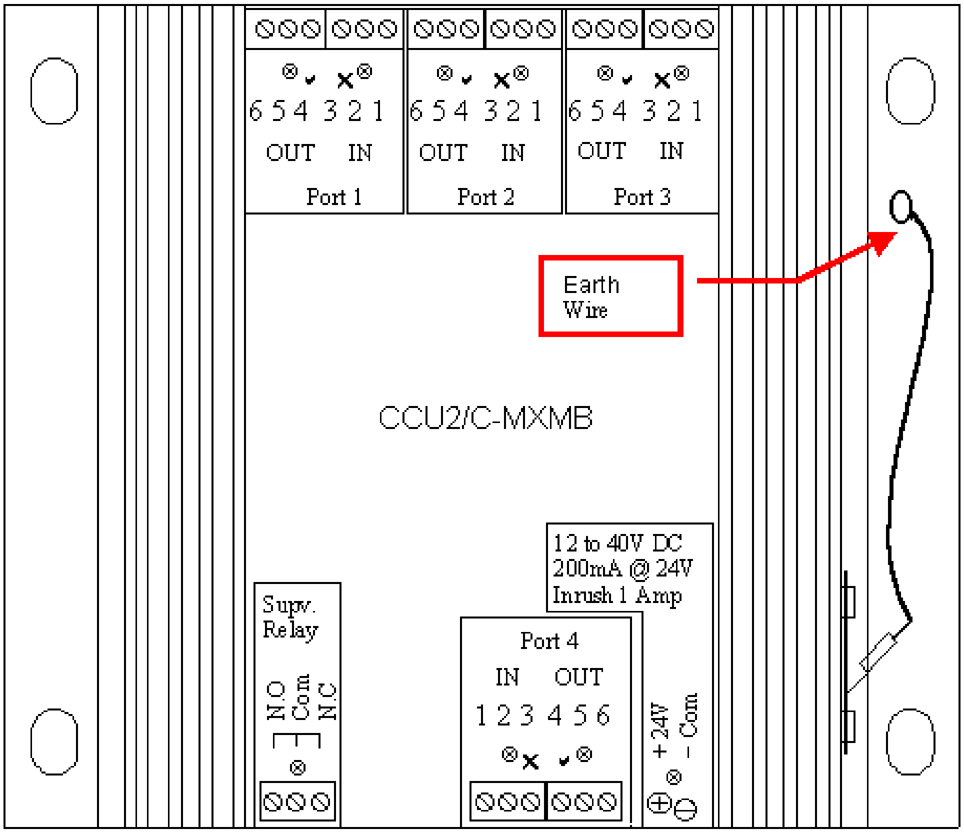


Figure 2:CCU2/C-MXMB Face Plate

2.5. Installing the mounting bracket:

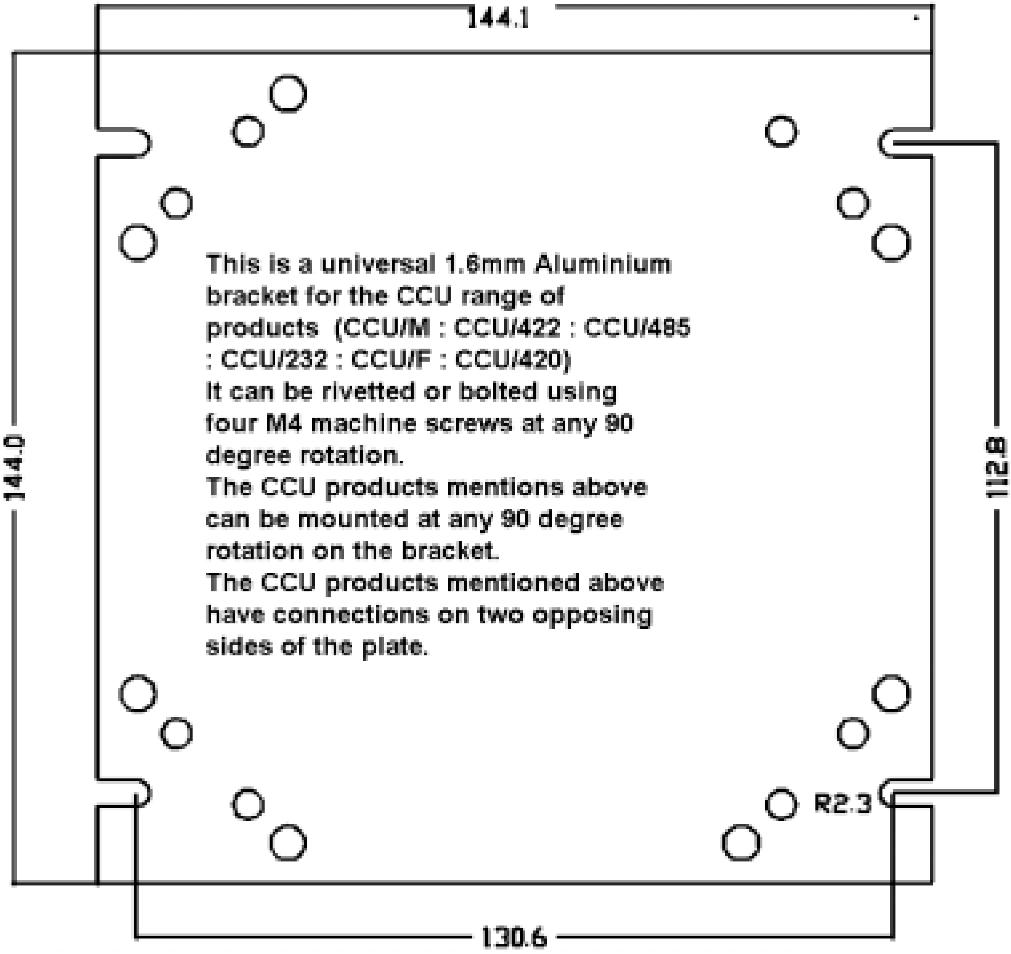

144.1
144.0
112.8
This is a universal 1.6mm Aluminium bracket for the CCU range of products (CCU/M : CCU/422 : CCU/485 : CCU/232 : CCU/F : CCU/420)
It can be rivetted or bolted using four M4 machine screws at any 90 degree rotation.
The CCU products mentions above can be mounted at any 90 degree rotation on the bracket.
The CCU products mentioned above have connections on two opposing sides of the plate.
R2.3
130.6

Figure 3: Mounting Bracket

2.6. Connecting the CCU2/C-MXMB:

Port 4 on the CCU2/C-MXMB connects to a TLI-800 (TPI) network interface on the MX. Port 3 connects to the MODBUS. Port 2 may be connected to a CCUioNET of CCU/IO or CCU/I boards, for added inputs and outputs. Port 1 is the reprogramming port as well as the debugging port.

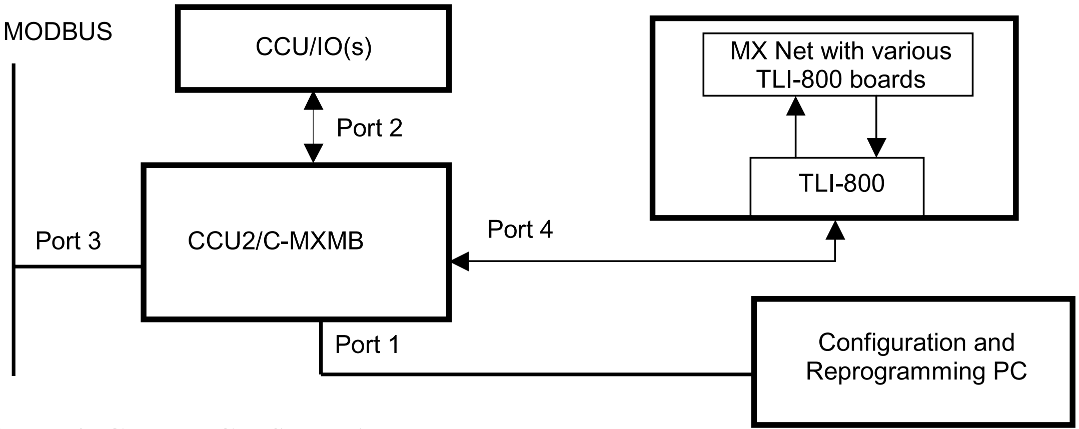
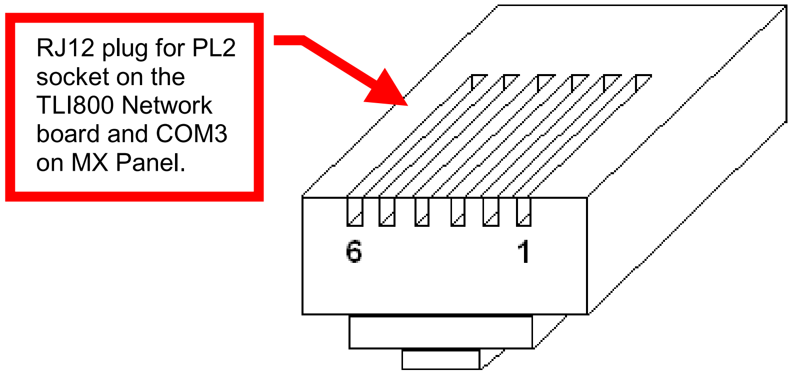

MODBUS
CCU/IO(s)
Port 2
MX Net with various TLI-800 boards
TLI-800
Port 3
CCU2/C-MXMB
Port 4
Port 1
Configuration and Reprogramming PC

Figure 4: General Configuration

2.6.1. Connecting the CCU2/C-MXMB to the MXNet or MX Panel:



RJ12 plug for PL2 socket on the TLI800 Network board and COM3 on MX Panel.

Figure 5: RJ12 Plug for TLI800 and COM3 Panel Connection.

Pinout for Cable Connection from the CCU2/C-MXMB to the TLI800 Network Interface Board on MX Panel	
RJ12 Pin Number	CCU2 Port 4 Pin Number 2 Orange 3pin (3.5 pitch) Weid Muller plug
6	NC
5	NC
4	1
3	3
2	4
1	2 & 5

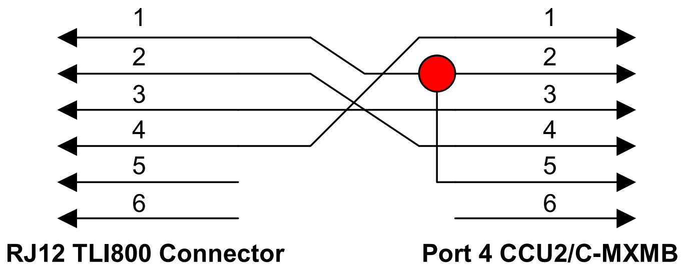


Figure 6: Diagram Pinout for cable connection between CCU2/C-MXMB and the MX TLI800 Network board for MX.

Pinout for Cable Connection from the CCU2/C-MXMB to COM3 on MX Panel	
RJ12 Pin Number	CCU2 Port 4 Pin Number
6	NC
5	1
4	NC
3	4
2	3
1	2 & 5

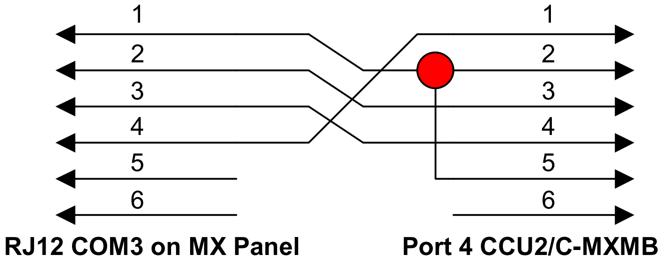


Figure 7: Diagram Pinout for cable connection between CCU2/C-MXMB and COM3 on MX.

2.6.2. Connecting the CCU2/C-MXMB to the MODBUS Network:

Pinout for Cable Connection from the CCU2/C-MXMB to a MODBUS 485 Network	
RS485 Wiring	CCU2 Port 3 Pin Number
NC	1
+ve	2
-ve	3
+ve	4
NC	5
-ve	6

Pinout for Cable Connection from the CCU2/C-MXMB to a MODBUS RS232 Port	
RS232 Wiring (DTE)	CCU2 Port 3 Pin Number
7	1
5	2
3	3
2	4
5	5
8	6

2.6.3. Connecting the CCU2/C-MXMB to a PC:

Pinout for Cable Connection from the CCU2/C-MXMB to the PC COM Port	
PC COM Port DB9 Pin Number	CCU2001 Port 1 Pin Number
7	1
5	2
3	3
2	4
5	5
8	6

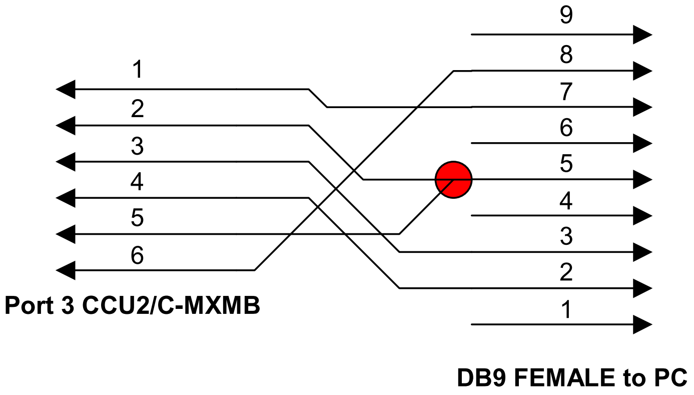
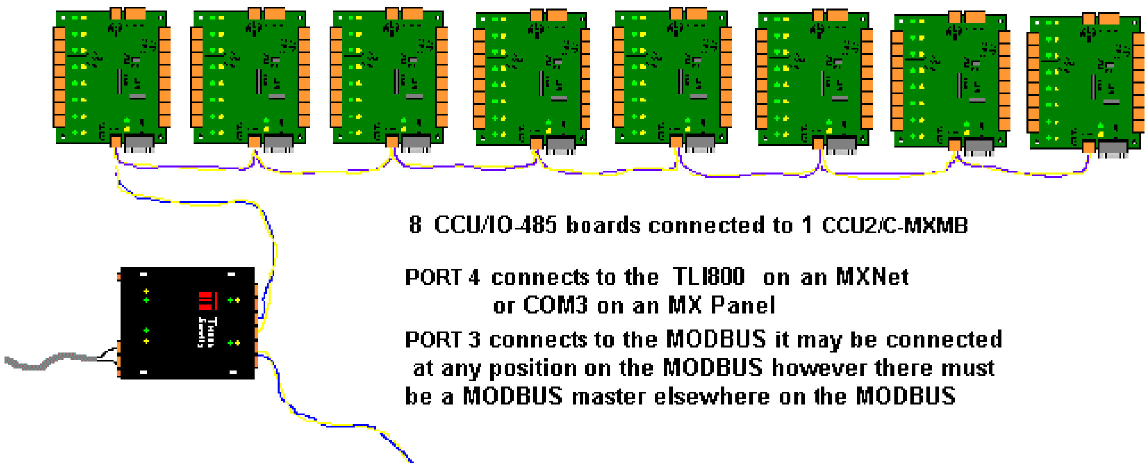


Figure 8: Diagram Pinout for cable connection between CCU2/C-MXMB and PC COM Port.

2.6.4. Connecting the CCU2/C-MXMB to the CCUioNET

To connect the CCU2/C-MXMB to a chain of CCU/IO-485s or CCU/I-485s connect the RS485 (SKT33) port of the CCU/IO-485 (or CCU/I-485) to PORT 2 of the CCU2/C-MXMB using the following connection diagram (Figure 5). Up to eight CCU/IO-485 boards can be chained off the CCU2/C-MXMB. Appendix B contains information on wiring for this configuration. Check the CCU/IO-485 documentation for jumper settings for this configuration. Alternatively if the CCU2/C-MXMB has Port 2 configured in RS232 mode you may connect a single CCU/IO-232 or CCU/I-232 to Port 2 of the CCU2/C-MXMB as seen in Figure 6.



8 CCU/IO-485 boards connected to 1 CCU2/C-MXMB

PORT 4 connects to the TLI800 on an MXNet
or COM3 on an MX Panel

PORT 3 connects to the MODBUS it may be connected
at any position on the MODBUS however there must
be a MODBUS master elsewhere on the MODBUS

Figure 9: Connection of CCU/IO-485(s) to a CCU2/C-MXMB

CCU/IO-485 RS485 (SKT33)	CCU2/C-MXMB Port 2
+	2,4
-	3,6
middle pin	Not Connected

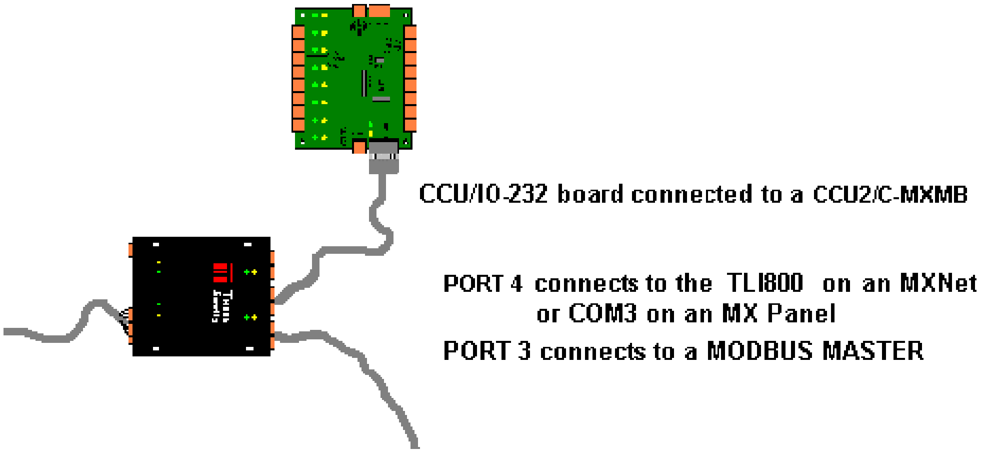

CCU/IO-232 board connected to a CCU2/C-MXMB
PORT 4 connects to the TLI800 on an MXNet
or COM3 on an MX Panel
PORT 3 connects to a MODBUS MASTER

Figure 10: Connection of a CCU/IO-232 or CCU/I-232 to a CCU2/C-MXMB

CCU/IO-232 RS232 (DCE)	CCU2/C-MXMB Port 2
2	3
3	4
5	2, 5

2.6.5. Connecting the CCU2/C-MXMB to Power

Connect the 24V DC power source to the supplied connector. The power connector is next to Port 4. Pin connections are:
PWR (+24VDC)   RETURN/COM(0V)

These connections are written on the top of the unit.

Pin number	Function
–	COM
+	24V

2.6.6. Setting the MODBUS Address of the CCU2/C-MXMB

The MODBUS address of the CCU2/C-MXMB is set through the **CCU2-MXMB Setup** program.

2.7. CCU2 Port Configuration

Each Port on the CCU2 can be jumper selected for RS232, RS485 or RS422 operation. The following sections will denote the jumper settings for each of these modes of communication and the appropriate pinout (with reference to the figure below).

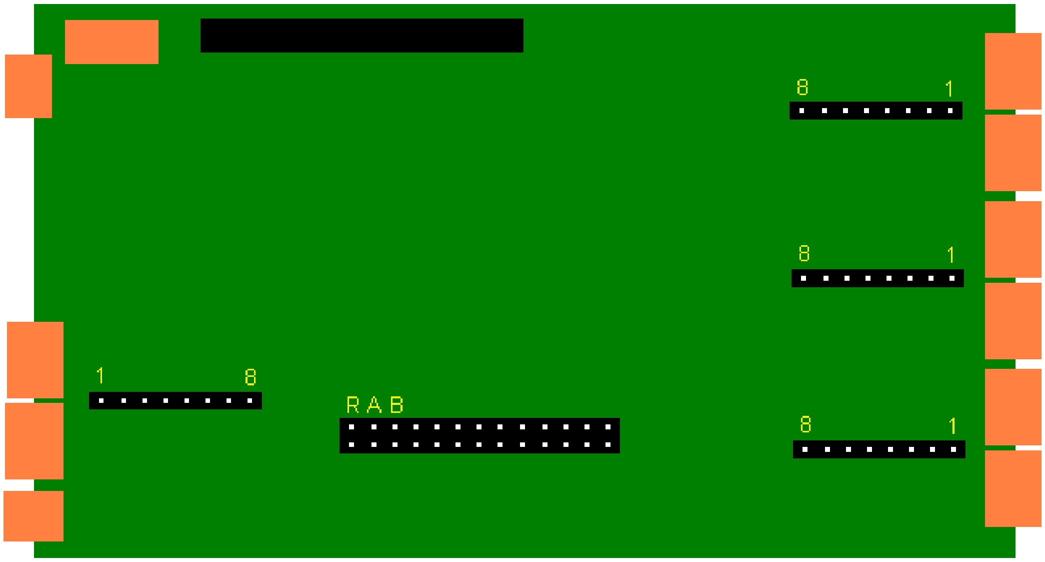


Figure 11: CCU2 PCB

Each Port has 8 jumper pins 1 to 8. These are the only jumpers that need to be changed.

2.7.1. RS232 Operation

For RS232 Port Operation there is only one jumper and this is placed on **PINS 2 and 3**. The pinout for the port is the following

Pin Number	Function
1	Clear to Send
2	GND
3	Rx
4	Tx
5	GND
6	Ready to Send

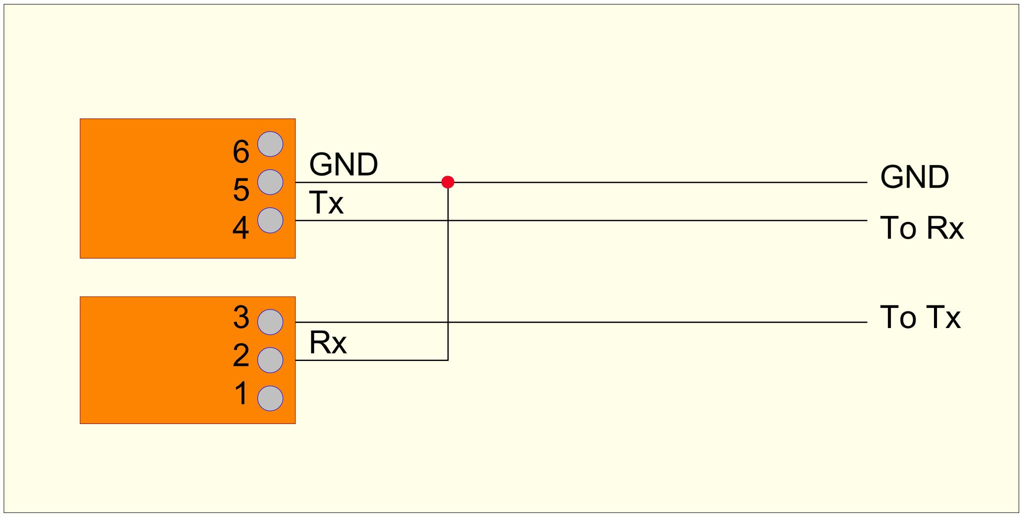


Figure 12: RS232 Connection

2.7.2. RS485 Operation

For RS485 Operation there are 2 jumpers, one placed on **PINS 7 and 8** and the other on **PINS 1 and 2**.
The pinout for the port is the following

Pin Number	Function
1	Not Connected
2	Tx+
3	Tx -
4	Tx +
5	Not Connected
6	Tx -

Note: Pins 2 and 4 are connected together. Pins 3 and 6 are also connected together.

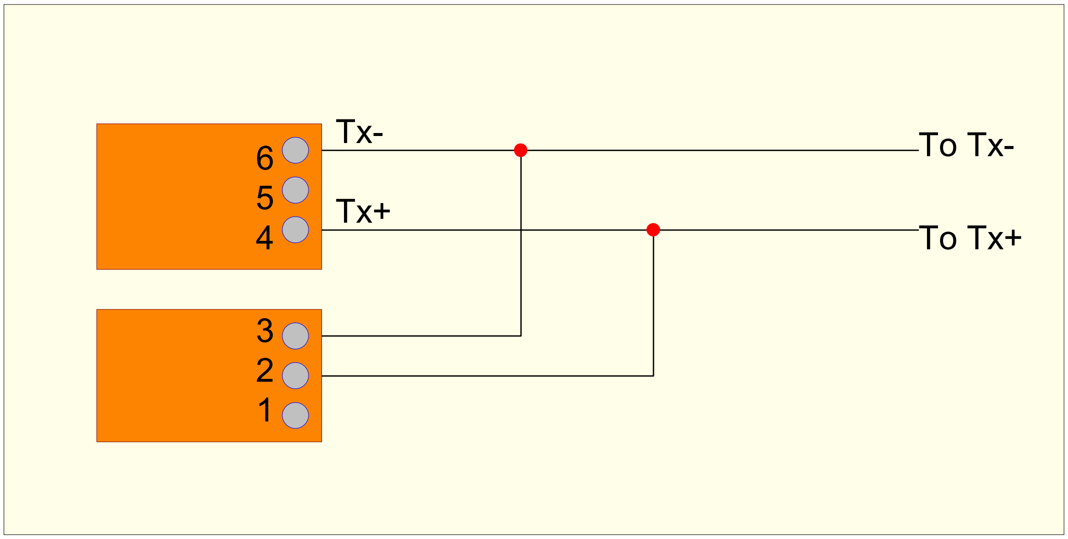


Figure 13: RS485 Connection

2.7.3. RS422 Operation

For RS422 Operation there are 2 jumpers, one placed on **PINS 5 and 6** and the other on **PINS 1 and 2**.
The pinout for the port is as follows:

Pin Number	Function
1	Not Connected
2	Rx+
3	Rx-
4	Tx+
5	Not Connected
6	Tx-

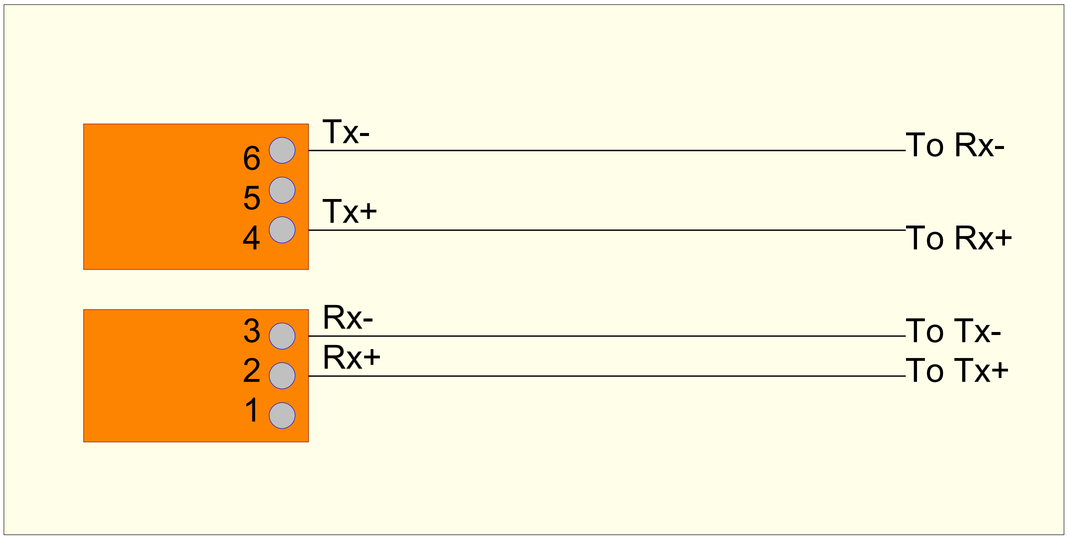


Figure 14: RS422 Connection

2.8. Normal and Fault Indication

2.8.1. Operation of LEDs

Each port has a Green and a Yellow LED. The green led flashes when data is transmitted on the corresponding port. The yellow LED glows when there has been supervision failure on the port or it is unable to make a connection. If a port has been disabled in the software then neither LED will glow.

Port 1 (Reprogramming Port):

- Green LED Flashes to indicate Data Transmission (for both downloading of firmware and configuration data).

Port 2 (connected to CCUioNET):

- If CCUioNET is disabled then both LEDs will remain extinguished.
- Green LED Flashes to indicate Data Transmission.
- Yellow LED indicates Supervision failure.

Port 3 (MODBUS RS232 port):

- Green LED Flashes to indicate Data Transmission.
- Yellow LED indicates Supervision failure.

Port 4 (TLI800 port):

- Green LED Flashes to indicate Data Transmission.
- Yellow LED indicates Supervision failure.

2.8.2. Supervision Relay

The CCU2/C-MXMB will supervise the MODBUS connection. In the event of supervision failure of the MODBUS, the relay on the CCU2/C-MXMB will operate. This can optionally be connected to an input on the MX for annunciation of MODBUS fault.

3. Configuration

3.1. Interfacing to the MX MODBUS.

3.1.1. General

The CCU2/C-MXMB connects to a MX MODBUS as a slave. It recognises 4 MODBUS commands:

Code	Description
2	Read Input Status
4	Read Input Registers
15	Force Multiple Coils
16	Preset Multiple Registers

The CCU2/C-MXMB is programmed with a Global map of all of the MX panels connected to it. The MODBUS offset address of each of the panels added into the map is configurable. However it is recommended that the original offsets address be kept to allow consistency. The bits within the panels and CCUs are in a fixed order and are of a fixed size.

A CCUioNET map is provided after the SUBMAP to utilise CCUioNET devices (CCU/I or CCU/IOs) that can be connected to the CCU2/C-MXMB

The CCU2/C-MXMB MODBUS map can be accessed by the MODBUS master in two different ways-
1. Direct access to the first 65535 bits of the Global Map.
2. Through a submap which mirrors 1000 bits of the global map. This submap is completely configurable and each bit in the submap can mirror any bit in the Global map.

Note: The submap always occupies the first 1000 bits of the MODBUS map.

3.1.2 MX CCU Global Map

The entire MOBBUS map is made up of two parts: a submap and a global map. The first block of 8192 bits is reserved to include the submap, CCUioNET MXNet and Sector map. Each MX panel is given a following block of 8192 bits that fully describe it. This global map may be accessed directly or addressed via a submap of 1000 bits that point to specific bits out of the global MODBUS map. The **CCU2-MXMB Setup** program includes a facility to individually offset each MX control panel map. This allows the PLC a more flexible approach to its memory management.

Register Map	Bit Map	Registers Usage
Register 0 Register 511	Bit 0 Bit 8191	Submap, CCUioNet, MXNet and Reserved
*Register 512 Register 1023	Bit 8192 Bit 16385	MX panel 1
*Register 1024 Register 1535	Bit 16384 Bit 24575	MX panel 2
*Register 1536 Register 2047	Bit 24576 Bit 24767	MX Panel 3
*Register 2048	Bit 32768	MX Panel 4

Register 2559	Bit 40959	
*Register 2560 Register 3071	Bit 40960 Bit 49151	MX Panel 5
*Register 3072 Register 3583	Bit 49152 Bit 57343	MX Panel 6
*Register 3584 Register 4095	Bit 57344 Bit 65535	MX Panel 7

Modbus Map Overview

The Modbus memory map is arranged in groups.

The Global Map exists in all CCU2/C-MXMBs, however it is not necessary to know the structure of it unless the user is addressing it directly. If the submap is used then this section will be of no use to the user.

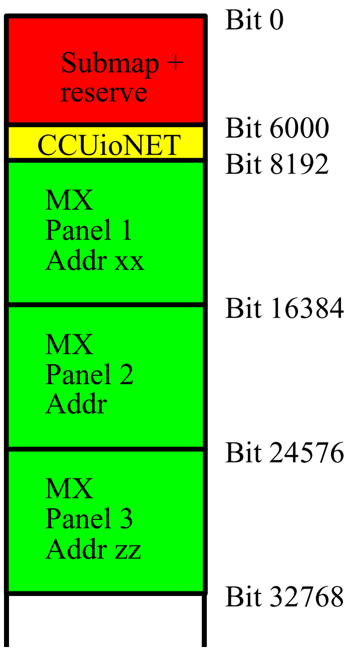


Figure 15: Global MODBUS map overview

CCUioNET Block

Bit Number	Bit Usage
Register 375, Bit 0-7	CCUioNet Node offline
Register 375 through to Register 406, Bit 8 – 519	128 inputs with 4 bits each
	Bit 0 – active, Bit 1- supervision fault, Bit 2 – reserved, Bit 3 – reserved
Register 407 through to Register 417, Bit 520 – 584	64 outputs (relay energise bits)

CCUioNET block Register/bit allocation

MXNet Block

This block provides some MXNet information to the PLC. The MX CCU configuration software is programmed with information relating to valid MXNet addresses. This information shall be available to the PLC equipment in accordance with the tables below.

Register	Register Usage
Register 420, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 1 to 15)
Register 421, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 16 to 31)
Register 422, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 32 to 47)
Register 423, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 48 to 63)
Register 424, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 64 to 79)
Register 425, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 80 to 95)
Register 426, Bit 0 – Bit 15	Valid MXNet Control address (MXNet address 96 to 99)
Register 427, Bit 0 – Bit 15	Spare

MXNet address allocation

Reg No	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
420	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	-
421	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
422	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
423	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
424	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64
425	95	94	93	92	91	90	89	88	87	86	85	84	83	82	81	80
426	-	-	-	-	-	-	-	-	-	-	-	-	99	98	97	96

MXNet Node address allocations

If bit 0 of register 420 is set at 0, then the CCU will determine that there is an MXNet system connected. It shall determine the number of nodes, and node address by reading the contents of registers 420 to 426 inclusive. If bit 0 of register 420 is set to 1, then the CCU will determine that the there is no MXNet system connected, but instead, it shall determine that a standalone MX fire control panel is connected.

Immediately after the valid node address setup table, we have the network sector status map. This map provides the PLC with indications of the total network sector alarm status. If any device in any of the 240 available sectors is in the alarm, fault or isolated status, a bit shall be set in this area of the map.

Register Numbers	MX Network Sector Status Map
Register 428 through to Register 488	Sectors 1 to 240 with 4 bits each Bit 0 – Alarm, Bit 1 – Fault, Bit 2 – Isolated, Bit 3 - Reserved

MXNet Sector Status Map

Time and Date Register

Typically one node of the MX network is set as the time and date keeper. This node periodically broadcasts time and date information via a special packet. If the TLI800 node is set as the time and date keeper then this register shall hold the date, to be transmitted to the MXNetwork. Register 489, is used for network date information, the value for each bit is as below:

Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
64	32	16	8	4	2	1	8	4	2	1	16	8	4	2	1
Year							Month				Day				

Date register

Time Register

If the TLI800 node is set as the time and date keeper then this register shall hold the time, to be transmitted to the MXNetwork.
Register 490 and 491 are used for network time information, the value for each bit is as below:

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
-	-	-	-	32	16	8	4	2	1	32	16	8	4	2	1
Not Used				Minutes						Seconds					

Register 490

Bit15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	Bit 0
-	-	-	-	-	-	-	-	-	-	-	16	8	4	2	1
										Hours					

Register 491

Register Number on Panel Base address	Information Stored
Register 1	Command acknowledge register
Register 2 and 3	Panel Status Registers
Register 4	Panel Control Register
Register 5,6,7,8	Loop Information registers
Register 9 to 68	Zone status information
Register 69 to324	Detector information registers
Register 325 to371	Point data registers
Register 372 to 378	Isolation data registers

MX Panel register allocation

Command acknowledge register (CAR)

This register contains control bits that are required so that the PLC can initiate point data requests and also perform device isolations. **(The following communication assumes that the MX CCU is configured as a slave device).**

Bit	Meaning when set
Bit 0	Point data request done
Bit 1-7	Global bits not used
Bit 8	Isolate request done
Bit 9 - 15	Global bits not used

Command acknowledge register

Communication between PLC, MX CCU and MX panel is as follows:-
1/ The PLC checks the CAR request done bit in the MX CCU and waits until it is cleared.
2/ The PLC sends its point data or isolate request to the registers of the MX CCU.
3/ The MX CCU checks if it is a valid request and then sends the request to the MX panel.
4/ The MX panel processes the request and replies to the MX CCU.
5/ The MX CCU stores the reply in the reply register, then sets the CAR request done bit.
6/ The PLC then reads the reply to its request.
7/ The PLC then sends a “Null” command to instruct the MX CCU to clear the command done bit.
8/ The MX CCU process the “Null” command and clears the command done bit.
The isolate/de-isolate reply packet confirms that the panel has carried out the request. PLCs can check the isolate status by reading the appropriate isolate bit.
The reply to the point data request will contain all the relevant information required by the PLC for the appropriate point.

Panel Status Registers

These two registers contain information relating to MX panel status.

Register 2	Command
Bit 0	MX Panel Resetting
Bit 1	MX Panel Starting Up
Bit 2	MX Panel Not Responding
Bit 3	MX Common Alarm
Bit 4	MX Common Isolate
Bit 5	MX Common Fault
Bit 6	Main PCB failure
Bit 7	Customer EPROM checksum error
Bit 8	Sounder cct 1 fault
Bit 9	Sounder cct 2 fault
Bit 10	Mains supply fault
Bit 11	Battery fault
Bit 12	Earth fault
Bit 13	Remote Bus fault
Bit 14	Signalling device fault
Bit 15	Global bit not used

Panel Status (Register 2)

This register contains information about system isolations as follows:

Register 3	Meaning when set
bit 0	Sounder circuit 1 isolated
bit 1	Sounder circuit 2 isolated
bit 2	Fire alarm contact isolated
bit 3	Signalling device isolated
Bit 4 to 15	Global bits not used

Panel Status (Register 3)

Panel Control Register

This register is used to pass commands to the MX control panel. The commands available are as follows:

Register Contents	Command
Bit 0	Silence Buzzer
Bit 1	Silence Panel
Bit 2	Panel resound
Bit 3	Daymode
Bit 4	Nightmode
Bit 5	Function 1
Bit 6	Function 2
Bit 7	Reset
Bit 8	Fire Alarm/Evacuate
Bit 9 to 15	Global bits not used

Panel Control Register (register 4)

Loop Information

Stored in ascending order, each 16 bit register contains information about a particular loop in the system. The format is as follows:

Register Offset	Information Stored
Register 5	Loop A Information
Register 6	Loop B Information
Register 7	Loop C Information
Register 8	Loop D Information

Loop Information registers

Bit	Meaning when set
Bit 0	Loop Shutdown
Bit 1	Loop Communication Fault
Bit 2	Loop High Current Consumption
Bit 3	Loop Open circuit
Bit 4	Loop Short Circuit
Bit 5	Loop card fault
Bit 6-15	Global bits not used

Loop information register contents

Zone Status Information

The zone status information registers provide an overview of the local control panel detection zones. Each panel can have up to 240 fire detection zones configured. This batch of registers correlates directly to the Detector information registers listed in 4.8.

Register Offsets	MX Panel Zone Status Map
Register 9 through to Register 68	Zones 1 to 240 with 4 bits each Bit 0 – Alarm, Bit 1 – Fault, Bit 2 – Isolated, Bit 3 – Prealarm

Zone status information register

Detector information

Bit 2 refers to the isolation of the addressable device. If the device address has either an input or an output isolated, or both, then this bit is set.

Register Offset	Register Usage
Register 69 through to Register 132	Loop A, Points 1 to 250, with 4 bits each Bit 0 – Alarm, Bit 1 – Fault, Bit 2 – Isolated, Bit 3 – Prealarm
Register 133 through to Register 196	Loop B, Points 1 to 250, with 4 bits each
Register 197 through to Register 260	Loop C, Points 1 to 250, with 4 bits each
Register 261 through to Register 324	Loop D, Points 1 to 250, with 4 bits each

Detector information registers

Point Data Area

A point data area is located within each of the pre-designated panel global map areas. The tables below list the data as it appears in each of these areas. Refer to section 4.2 of the MX Speak Protocol Specification. The tables contain MX speak packets with the headers stripped off.

Register	Byte	Usage	Value	Comments
325	10	Node	1-99	Point Address
	11	Channel	0-6	
326	12	Channel Address	0-15	
	13	Point Category	0-7	
327	14	Point Number	0-255	
	15	Device Category	0-19	0 = All 1 = Inputs 2 = Outputs 3 = Callpoints 4 = Detectors 5 = Smoke Detectors 6 = Heat Detectors 7 = Flame Detectors 8 = CO Detectors 9 = Digital Visual Outputs
328	16	Group	1-65535	Interested in Points from this Group
	17			
329	18	Output Point State Store	3	Always = 3 (= All Stores) ???
	19	Area Type	0-1, 3	Interested in Points from this Area ???
330	20	Area Number	0-255	
	21	Multi-Area Type	0-1, 3	Multi-Area bitmap Type
331	22	Areas 000-007	0-255	Lsb = Area 0
	23	Areas 008-015	0-255	Lsb = Area 8
	. . .	. . .	. . .	. . .
345	51	Areas 232-239	0-255	Lsb = Area 232
346	52	Area 240	0-1	Lsb = Area 240
347	53	Request Type	0-1	0 = First Point only 1 = All Points, one by one

	54	Search Type	0-10	0 = Points with Input Isolated 1 = Points with Output Isolated 2 = Points with Both Input and Output Isolated 3 = Points with Either Input or Output Isolated 4 = Points with Input Forced 5 = Points with Output Forced 6 = Points with Both Input and Output Forced 7 = Points with Either Input or Output Forced 8 = Points Untested 9 = Points out of Compensation 10 = All Points

Point Information Request

Register	Byte	Usage	Value	Comment
350	10	Reply Status	0-1	0 = Success, i.e. Point(s) found, 1 = Fail – no Point(s) found that match the criteria

	11	Flags	0-31	Bit0 = Input Forced
				Bit1 = Point Untested or Failed (Walk Test)
				Bit2 = Input Isolated
				Bit3 = Output Isolated
				Bit4 = Out of Compensation
				Bit5 = Non-Loop Point (0 = Loop Point)

351	12	Node	1-99	Point Address
	13	Channel	0-6	
352	14	Channel Address	0-15	
	15	Point Category	0-7	
353	16	Point Number	0-255	
	17	Device Type	0-254	As configured (See separate list of Device Types)
354	18	I/O Detail	0-3	0 = Nil 1 = Supports Input 2 = Supports Output 3 = Supports Both Input and Output
355	19	Group	1-65535	Group
	20			
356	21	Area Type	0-1	Area
	22	Area Number	0-255	
357	23	Sector ID	0-255	= 254 if Zone not in Sector
	24	Loop Type	0-2	0 = Thorn 1 = MX Digital 2 = Not Loop (i.e. Local or Remote Digital Points)
358	25	Raw Identity	0-255	Identity code/value returned by the device
	26	Actual Device Type	0-254	Device Type that corresponds to the Raw Identity
359	27	Mode & Sensitivity	0-255	As configured
	28	Raw Analogue Value 1	0-255	Value 1 as provided by the device, without conversion

360	29	Raw Analogue Value 2	0-255	Value 2 as provided by the device, without conversion
	30	Raw Analogue Value 3	0-255	Value 3 as provided by the device, without conversion
361	31	LTA Available Flag	0-1	0 = Not available (some devices don't have it) 1 = LTA available
	32	Raw LTA	0-255	LTA of main Raw Value
362	33	LTA Dirtiness	0-255	
	34	Unit of Measure 1	0-9	0 = Invalid 1 = Degrees C 2 = Degrees F 3 = ppm (parts per million) 4 = %/ft Obscuration 5 = %/m Obscuration 6 = Y Value 7 = 1/10 Amp 8 = 1/10 V 9 = Not Installed
363	35	Unit of Measure 2	0-9	
	36	Unit of Measure 3	0-9	
364	37	Converted Value 1	0-255	Value 1 in Units of Measure 1 (e.g. in %/m obscuration)
	38	Converted Value 2	0-255	Value 2 in Units of Measure 2 (e.g. in °C)
365	39	Converted Value 3	0-255	Value 3 in Units of Measure 3
	40	Instantaneous Active State	0-1	Instant single reading from driver 0 = Normal 1 = Active
366	41	Instantaneous Fault State	0-254	(See separate list of Fault States)
	42	Confirmed Active State	0-1	Confirmed by driver – true physical state of the Point 0 = Normal 1 = Active
367	43	Confirmed Fault State	0-254	(See separate list of Fault States)
	44	Acknowledged Active State	0-1	Internally acknowledged as the driver state for the Point ??? 0 = Normal 1 = Active
368	45	Acknowledged Fault State	0-254	(See separate list of Fault States)
	46	Internal Active State	0-1	Internally used "official" state (may be a Forced Input) ??? 0 = Normal 1 = Active
369	47	Internal Fault State	0-254	(See separate list of Fault States)
	48	Output Forced Mode	0-1	0 = Unforced 1 = Forced

370	49	Unforced State	0-4	0 = OFF 1 = ON 2 = Pulse 1 (pulsing in mode 1) 3 = Pulse 2 (pulsing in mode 2) 4 = Point with no Output
	50	Forced State	0-4	0 = OFF 1 = ON 2 = Pulse 1 (pulsing in mode 1) 3 = Pulse 2 (pulsing in mode 2) 4 = Point with no Output
371	51	Client ID	0-65535	Provided by the Server
	52			

Point Information Reply

Point Isolate/Deisolate Area

Register	Byte	Usage	Value	Comment
372	10	Isolate/De-isolate Sub-command	0-5	0 = Isolate Both Input and Output 1 = Isolate Input 2 = Isolate Output 3 = De-isolate Both Input and Output 4 = De-isolate Input 5 = De-isolate Output
	11	Target Category	0-8	0 = Any Point 1 = Points with Input 2 = Points with Output 3 = Detectors 4 = Callpoints 5 = Sounders 6 = Plant Controls 7 = Extinguishing Controls 8 = Signalling Devices (Tele-communications transmitters)
373	12	Target Type	0-2	0 = Point 1 = Zone 2 = Sector
		If Target Type = Point		
374	13	Channel	0-5	
	14	Channel Address	0-15	
375	15	Point Category	0-6	
	16	Point Number	1-255	
		If Target Type = Zone or Sector		
374	13	Area Number	0-255	

Point isolate de-isolate Request

Register	Byte	Usage	Value	Comment
376	10	Isolate/De-isolate Sub-command	0-5	0 = Isolate Both Input and Output 1 = Isolate Input 2 = Isolate Output 3 = De-isolate Both Input and Output 4 = De-isolate Input 5 = De-isolate Output
	11	Target Category	0-8	0 = Any Point 1 = Points with Input 2 = Points with Output 3 = Detectors 4 = Callpoints 5 = Sounders 6 = Plant Controls 7 = Extinguishing Controls 8 = Signalling Devices (Tele-communications transmitters)
377	12	Target Type	0-2	0 = Point 1 = Zone 2 = Sector
		If Target Type = Point		
378	13	Channel	0-5	
	14	Channel Address	0-15	
379	15	Point Category	0-6	
	16	Point Number	1-255	
		If Target Type = Zone or Sector		
378	13	Area Number	0-255	

Point isolate de-isolate Reply

3.2. Configuring the MODBUS Map

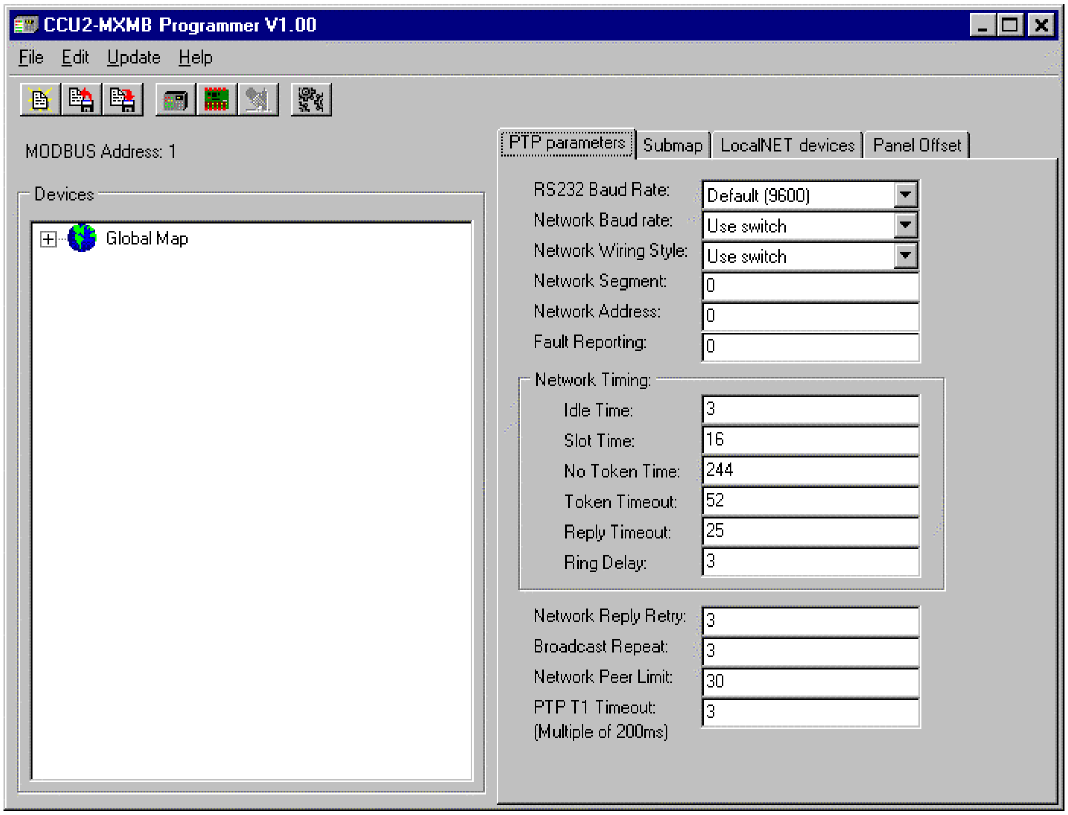
The MODBUS map must be configured using the **CCU2-MXMB Setup** program. The program allows the user to add MX panels and CCUioNET devices into the Global Map. For most applications the most useful feature is the ability to make a submap of up to 1000 bits from the bits in the global map.

The submap allows the user to mirror 1000 bits from the global map in the submap, which is stored from address position 0 to 999. For instance bit 8345 in the global map may be mapped to bit 0 in the submap. This gives the user flexibility in the layout of the MODBUS map.

3.2.1. The CCU2-MXMB Setup program

The **CCU2-MXMB Setup** program is designed for use with the CCU2/C-MXMB. It allows the user to configure the MODBUS map on the CCU2/C-MXMB and download the configuration to the CCU2/C-MXMB.

When the **CCU2-MXMB Setup** program is executed a screen similar to figure 8 (below) should appear.



MODBUS Address: 1

RS232 Baud Rate: Default (9600)
Network Baud rate: Use switch
Network Wiring Style: Use switch
Network Segment: 0
Network Address: 0
Fault Reporting: 0

Idle Time: 3
Slot Time: 16
No Token Time: 244
Token Timeout: 52
Reply Timeout: 25
Ring Delay: 3

Network Reply Retry: 3
Broadcast Repeat: 3
Network Peer Limit: 30
PTP T1 Timeout: (Multiple of 200ms) 3

Figure 16: CCU2-MXMB Setup main screen

On the left is a tree which can be expanded or contracted by clicking the '+' and '-' signs. The tree shows all of the MX panels and all CCUioNET devices that are set up in the CCU2/C-MXMB, this list is called the **Global Map Tree**.

The grey bar containing the words **File, Edit, Update** and **Help,** is referred to as the **Main Menu** for the remainder of this document. Most of the program's options can be accessed through the Main Menu. From this point forward options in menus are referred to with the format *Menu* | *Command***.** For instance, the term **File | New** means click on **File** in the Main Menu, wait for a drop down menu to appear and then click on **New**.

The buttons below the Main Menu are equivalent to some items in the Main Menu. Because of their common usage they are available as buttons for easy access.

3.2.2. General configuration

The CCU2/C-MXMB is connected to the MODBUS and therefore must be given an address on the MODBUS. The MODBUS address is changed by selecting **Edit | Change MODBUS Address** from the Main Menu.

3.2.2.1. Opening, Saving and restoring CCU2/C-MXMB configurations

The user may clear the current configuration and start a new configuration by pressing the **New** button or selecting **File | New** from the Main Menu.

A configuration can be saved by pressing the **Save** button or selecting **File | Save** from the Main Menu. All saved files are given the extension *.TNM* by default.

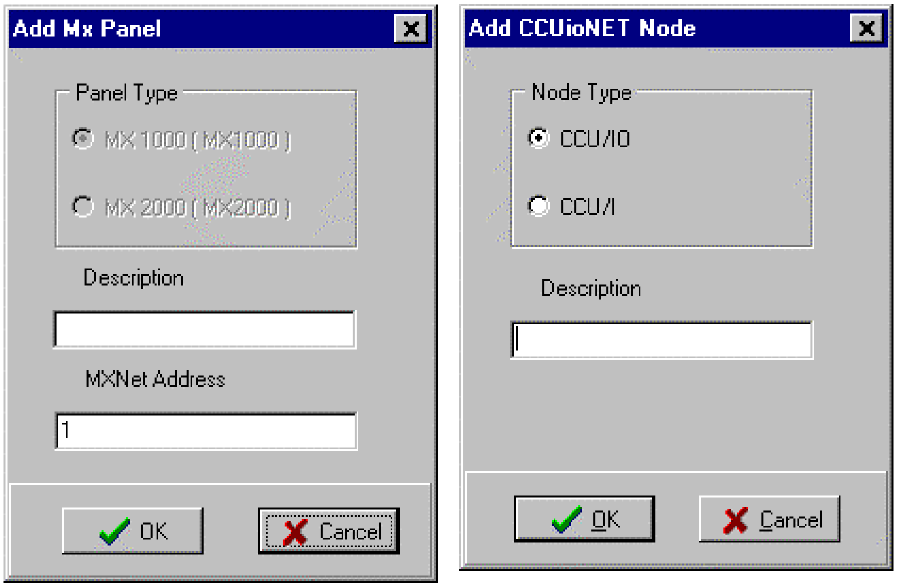
A configuration can be loaded by pressing the **Load** button or selecting **File | Load** from the Main Menu.

3.2.3. Adding to the global map

To add more MX panels to the global map click the **Add MX panel** button.

To add more CCUioNet devices to the global map click the **Add CCUioNET node** button.

For both the above a new window should appear giving a list of devices to choose from. Select the correct device and fill in a brief description for easy identification. When adding a MX panel the user will also have to fill in the MXNet address, which is the address of the MX panel on the MXNET.



Add Mx Panel

Panel Type

● MX 1000 (MX1000)

○ MX 2000 (MX2000)

Description

MXNet Address: 1

Add CCUioNET Node

Node Type

● CCU/IO

○ CCU/I

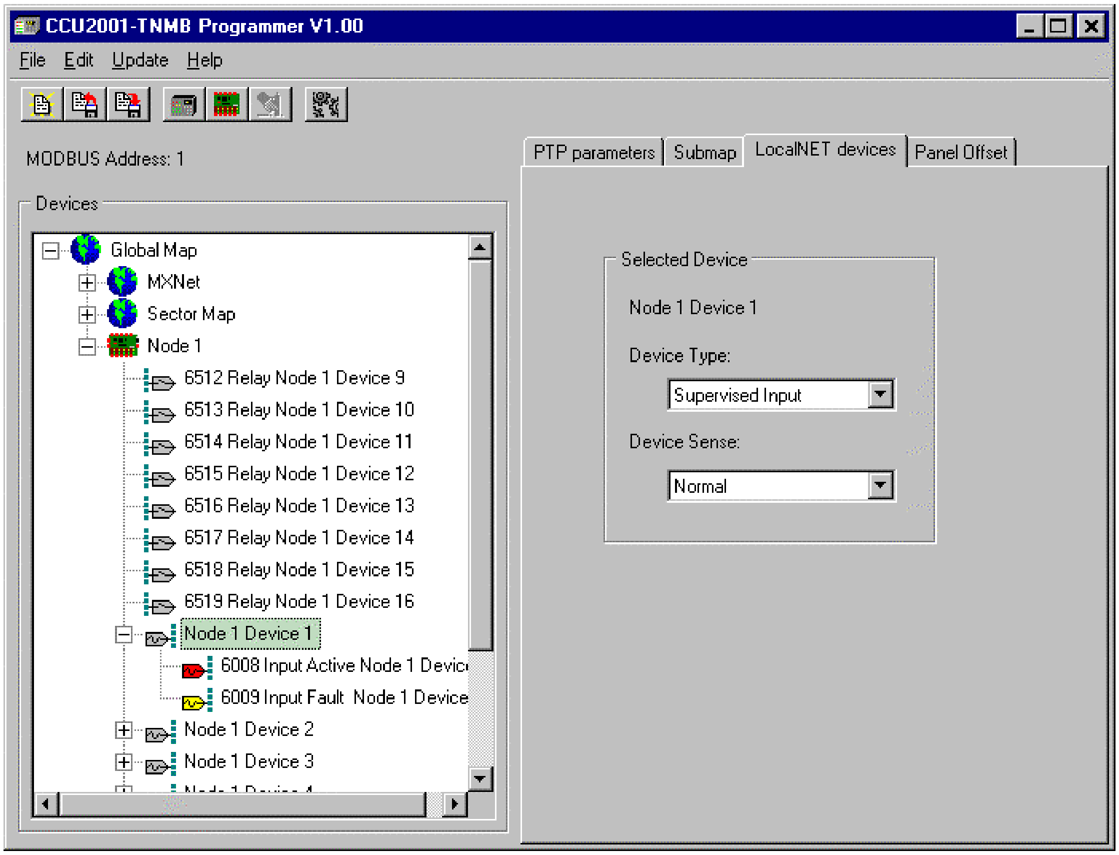
Description

Figure 17: Add new device screens

To remove a panel or CCUioNET node select the panel in the **Global Map Tree** the press the **Delete** button or select **Edit | Remove Panel** from the Main Menu

3.2.4. Configuring CCUioNET nodes

The inputs of the CCUioNET nodes must have their supervision set to either supervised or unsupervised, depending on how the CCU/IO-485s are configured. All of the CCUioNET inputs and outputs are configurable, to either read as **set** or **cleared** under normal conditions. Both the inputs and the outputs can operate in normal or reversed mode. In normal mode the relay outputs read **set** when the relays are energised, in reversed mode they would read **cleared** for this case. In normal mode the 'active' bit is **set** for a normal input, in reversed mode the 'active' bit is **cleared** for a normal input. The 'input fault' bit is not affected by the normal or reverse modes.


CCU2001-TNMB Programmer V1.00
File Edit Update Help
PTP parameters
Submap
LocalNET devices
Panel Offset
Devices
Global Map
MXNet
Sector Map
Node 1
6512 Relay Node 1 Device 9
6513 Relay Node 1 Device 10
6514 Relay Node 1 Device 11
6515 Relay Node 1 Device 12
6516 Relay Node 1 Device 13
6517 Relay Node 1 Device 14
6518 Relay Node 1 Device 15
6519 Relay Node 1 Device 16
Node 1 Device 1
6008 Input Active Node 1 Devic
6009 Input Fault Node 1 Device
Node 1 Device 2
Node 1 Device 3
Node 1 Device 4

MODBUS Address: 1

Selected Device

Node 1 Device 1

Device Type: Supervised Input

Device Sense: Normal

Figure 18: Configuring CCUioNET nodes

3.2.5. Changing the Submap

A submap is a map of up to 1000 bits that mimics 1000 bits from the global map. The user selects which bits to add to the submap from the global map and also the order in which they are arranged. The submap may include any bits from the global map including CCUioNET bits.

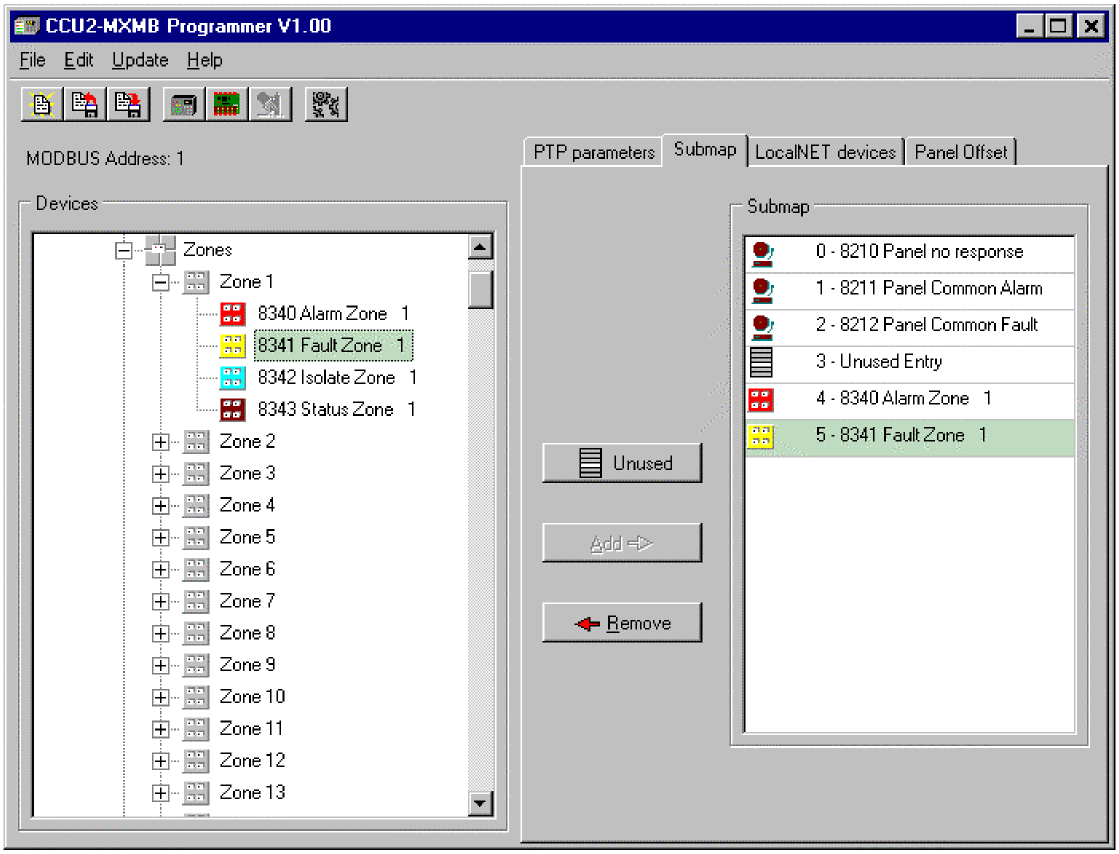
To add bits to a submap go to the Submap page, by clicking on the Submap tab at the top of the window (see figure 11). To add a bit to a submap find the bit that you wish to add in the Global map on the left hand side of the screen. Remember you can expand and withdraw branches by clicking the '+' and '-' signs. Select the bit then the click **Add** button. The bit should appear in the submap on the right-hand side of the screen. If the **Add** button is 'greyed out' then a valid bit has not been selected.

Only one individual bit may be added to the submap at once, for instance a Point in a MX panel may not be added to the submap, because a point is made up of an Alarm bit, a Fault bit and an Isolate Bit.

Blank bits, or unused addresses may be added to the submap by adding the **Unused Entry** in the **Global Map Tree** to the submap or clicking the **Unused** button.

To remove a bit from the submap select the bit from the submap on the right-hand side of the screen and then press the **Remove** button. The bit should disappear from the submap.

It is possible to add the same bit to the submap more than once.



- Zones
  - Zone 1
    - 8340 Alarm Zone 1
    - 8341 Fault Zone 1
    - 8342 Isolate Zone 1
    - 8343 Status Zone 1
  - Zone 2
  - Zone 3
  - Zone 4
  - Zone 5
  - Zone 6
  - Zone 7
  - Zone 8
  - Zone 9
  - Zone 10
  - Zone 11
  - Zone 12
  - Zone 13

Submap:

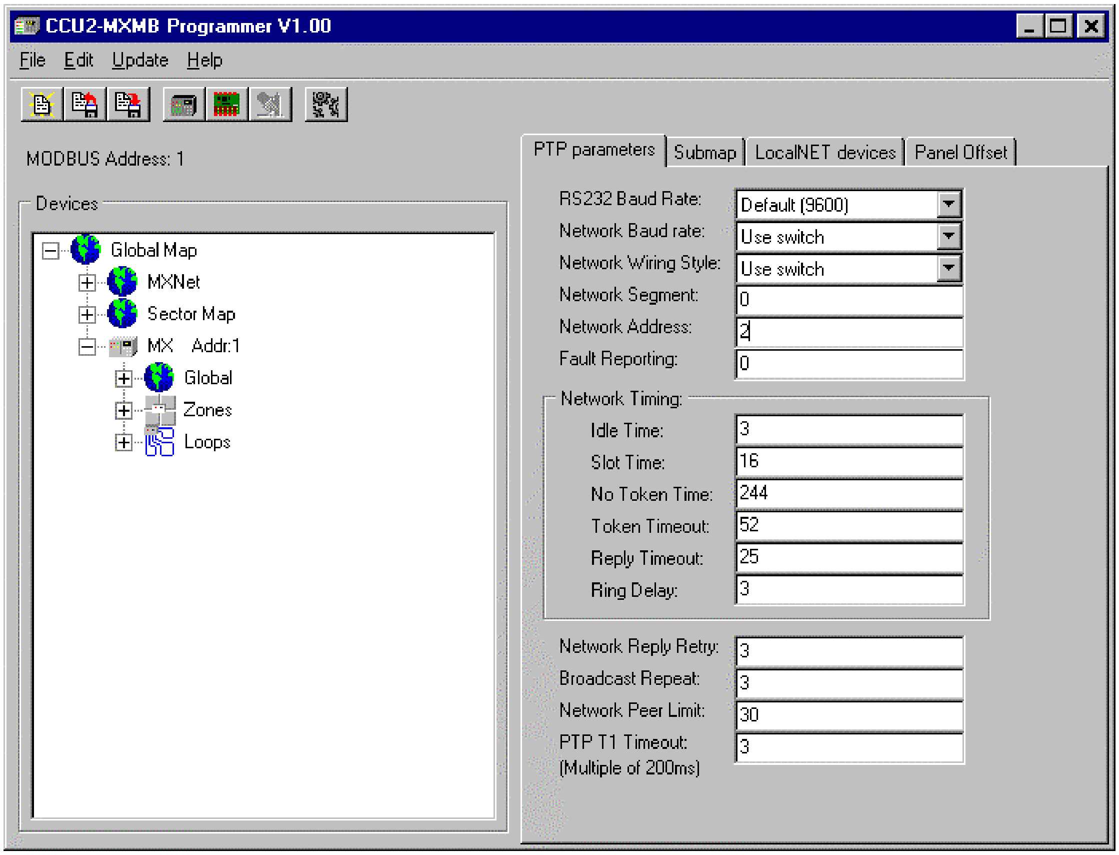
0 - 8210 Panel no response
1 - 8211 Panel Common Alarm
2 - 8212 Panel Common Fault
3 - Unused Entry
4 - 8340 Alarm Zone 1
5 - 8341 Fault Zone 1

Figure 19: Adding to a Submap

3.2.6. Setting the PTP parameters

These parameters reflect the settings for the communication between the MX panel and the CCU2/C-MXMB. The **Network Address** must be set to the jumper address of the TLI-800 and the rest of the settings set to match the network. For further information and help with these settings refer to publication number 19700344 **Network Interface Module** and to **MXSpeak Reference Guide for Advanced Third Party Interface Applications.**
It is recommended that the default settings displayed when the program is first run should be used. These are the default settings used in MXConsys.

ATTN: These parameters must be identical for all nodes on the MXNet. Also the Network Peer Limit value must be set to equal or exceed the number of nodes on the network.



RS232 Baud Rate: Default (9600)
Network Baud rate: Use switch
Network Wiring Style: Use switch
Network Segment: 0
Network Address: 2
Fault Reporting: 0

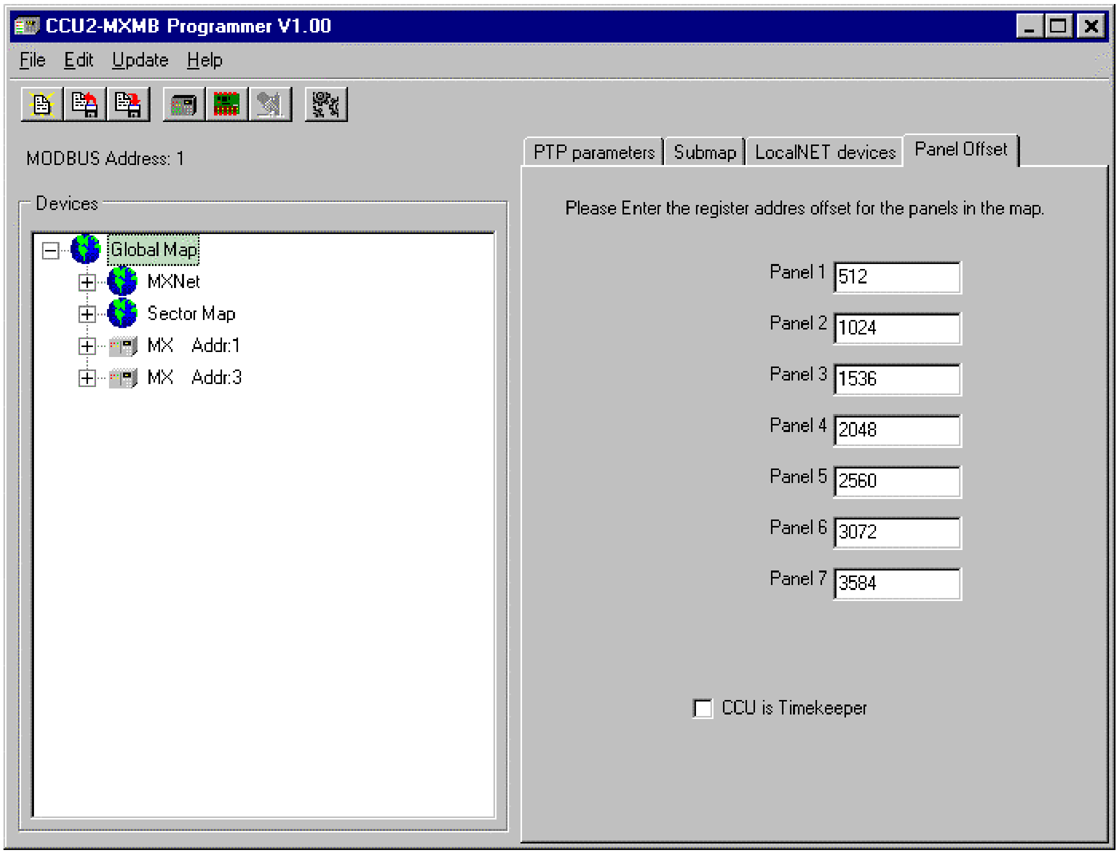
Network Timing:
Idle Time: 3
Slot Time: 16
No Token Time: 244
Token Timeout: 52
Reply Timeout: 25
Ring Delay: 3

Network Reply Retry: 3
Broadcast Repeat: 3
Network Peer Limit: 30
PTP T1 Timeout: (Multiple of 200ms) 3

Figure 20: Setting PTP Parameters

3.2.7. Setting the Panel address offset

The starting address of each of the panels added into the MODBUS map can be individually offset to allow better memory management of MODBUS master devices. The ability to change a panel's starting address is determined by the number of panels added into the Global map. Since the address field in the MODBUS commands, Read Inputs registers and Preset Multiple registers, is of word length then the range of the offset address that can be specified is from 512 to 65024. However in order to use the other two MODBUS commands, of bit orientation, the top offset address cannot be above 3584 (as the highest bit address that can be used in these commands is 65536). Please refer to the MODBUS Commands in section 4 for full command description.



MODBUS Address: 1

Please Enter the register addres offset for the panels in the map.

Panel 1	512
Panel 2	1024
Panel 3	1536
Panel 4	2048
Panel 5	2560
Panel 6	3072
Panel 7	3584

□ CCU is Timekeeper

Figure 21: Setting Panel address offsets Parameters

3.2.8. Setting the CCU as Timekeeper

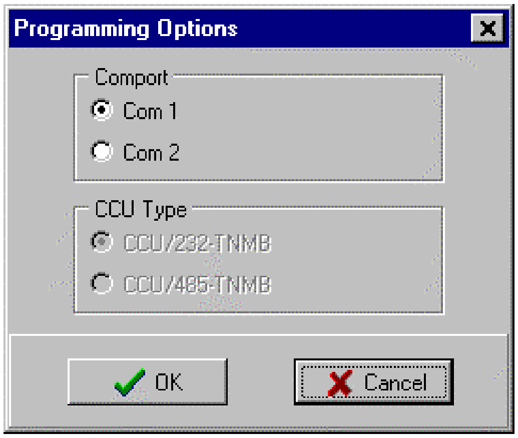
As seen in the figure on the previous page the Panel Offset tab window is also used to enable the CCU as timekeeper. Place a tick in the checkbox beside the text ‘CCU Timekeeper’ and after downloading the configuration data to the CCU the CCU will be enabled as timekeeper on the MXNet. Please see section 4.3 to use the timekeeping.

3.2.9. Programming the CCU2/C-MXMB

When the configuring process is complete the user should save the work and then program the CCU2/C-MXMB.

To plug the computer into the CCU2/C-MXMB refer to APPENDIX C listing the pinout descriptions for the necessary cable. Connect this cable from the desired COM port of the computer to Port 1 on the CCU2/C-MXMB.

Before programming is commenced the PC’s comport must be set in the **CCU2-MXMB Setup** program to the port that the CCU2/C-MXMB is connected to. To do this select **Update | Programming Option** from the Main Menu.



Programming Options

Comport
● Com 1
○ Com 2

CCU Type
● CCU/232-TNMB
○ CCU/485-TNMB

Figure 22: Programming Options

To program:
The CCU can be powered and working during programming.
Press the **Update Data** button or select **Update | Data only** from the Main Menu. Programming should commence and a success message will be displayed when finished.

4. Communication

4.1. MODBUS Communication Parameters

19200,n, 8,1

4.2. MODBUS Commands

The CCU2/C-MXMB connects to a MX MODBUS as a slave. The MODBUS master may use any of the following 4 MODBUS commands:

Code	Description
2	Read Input Status
4	Read Input Registers
15	Force Multiple Coils
16	Preset Multiple Registers

4.2.1. Read Input Status

Read Input Status reads a range of bits from the MODBUS map. This command can read up to 65535 bits starting at any address range from 0 to 65535 in the MODBUS map. This command is used to read from the Global MODBUS on the CCU2/C-MXMB.
The byte fields of this command type are as follows:

Request
```
1. MODBUS Address
2. MODBUS Function
3. Starting MODBUS bit address (High byte)
4. Starting MODBUS bit address (Low byte)
5. Number of bits (High byte)
6. Number of bits (Low byte)
7. CRC16 (Low byte)
8. CRC16 (High byte)
```

Reply
```
1. MODBUS Address
2. MODBUS Function
3. Number of bytes returned
4. Data (number of data bytes is ‘number of bytes returned’)
5. CRC16 (Low byte)
6. CRC16 (High byte)
```

The ‘Starting MODBUS bit address’ is always rounded down to its byte boundary and so the number of bytes returned will encompass the number of bits requested. Hence the bits are returned on their byte boundary (i.e. bit positions within the bytes are as their appear in the MODBUS map).
The status of the returned bits is indicated as: 1 = ON; 0 = OFF.
Note: If the number of bits returned does not divide evenly into eight then the remaining bits, front and back padding, will reflect the status of those bits with the same remaining valid addresses in the Global MODBUS map.

Example
Request:

Field	Example
MODBUS Address	5
MODBUS Function	2
Starting MODBUS Address (High)	1
Starting MODBUS Address (Low)	BF
Number of Bits (High)	0

Number of Bits (Low)	12
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 must be calculated using the appropriate formula by the MODBUS master and added in the last word position (it takes up 2 bytes).

This command will read 0012hex (18 decimal) bits from MODBUS device 5 starting at MODBUS address 01BFhex (447 decimal).

The CCU2/C-MXMB should return the bits from address 01BFhex to 01D1hex. Address 01BF is eighth bit in the 56th byte hence the starting byte returned is the 56th byte therefore the address of the first bit returned is 01B8hex. The bits are packed into bytes in the reply message. The first data byte of the reply contains the bits of the lowest 8 addresses, with the lowest or first address being represented by the LSB (Least Significant Bit).

Reply:
Notice the start address in this example is rounded down to the byte boundary and the number of bytes returned is rounded to include the top bit address.

Field	Example
MODBUS Address	5
MODBUS Function	2
Bytes Returned	4
Data (Address 01B8 – 01BF)	E5
Data (Address 01C0 – 01C7)	4D
Data (Address 01C8 – 01CF)	02
Data (Address 01D0 – 01D7)	78
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 is calculated by the CCU2/C-MXMB and takes up 2 bytes.

4.2.2. Read Input Registers

The Read Input Registers command allows the MODBUS master to read from the Global MODBUS map using the register address of the bits to be read. This command returns word wide bytes from the MODBUS map.

The byte fields of this command type are as follows:

Request

```
1. MODBUS Address
2. MODBUS Function
3. Starting MODBUS register/word address (High byte)
4. Starting MODBUS register/word address (Low byte)
5. Number of registers/words (High byte)
6. Number of registers/words (Low byte)
7. CRC16 (Low byte)
8. CRC16 (High byte)
```

Reply

```
1. MODBUS Address
2. MODBUS Function
3. Number of bytes returned
4. Data (number of data bytes is ‘number of bytes returned’, low byte
   followed by high byte)
5. CRC16 (Low byte)
6. CRC16 (High byte)
```

The status of the returned bits is indicated as: 1 = ON; 0 = OFF.
The first data byte of the reply contains the lower byte of the first register/word to be returned.

Example

Request:

Field	Example
MODBUS Address	5
MODBUS Function	4
Starting register (High)	0
Starting register (Low)	2
Number of Registers (High)	0
Number of Registers (Low)	1
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 must be calculated using the appropriate formula by the MODBUS master and added in the last word position (it takes up 2 bytes).

This command will read register 0002hex from MODBUS device 5. This register is part of the SUBMAP. The number of registers to be returned is 0001hex.

The CCU2/C-MXMB should return the 16 bits from the SUBMAP.

Reply:

Field	Example
MODBUS Address	5
MODBUS Function	4
Bytes Returned	2
Data (Register 2 (Low))	C0
Data (Register 2 (High))	4E
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 is calculated by the CCU2/C-MXMB and takes up 2 bytes.

4.2.3. Force Multiple Coils

Force Multiple Coils sets a range of bits in the MODBUS map. This command can set up to 65535 bits starting at any address range from 0 to 65535 in the MODBUS map. This command is used to set bits in the Global map of the CCU2/C-MXMB. The value of this bit will be reported back until the bit is rewritten by another device or by the CCU.

The byte fields of this command type are as follows:

Request
```
1. MODBUS Address
2. MODBUS Function
3. Starting MODBUS bit address (High byte)
4. Starting MODBUS bit address (Low byte)
5. Number of bits (High byte)
6. Number of bits (Low byte)
7. Data (number of bytes needed to write the appropriate data)
8. CRC16 (Low byte)
9. CRC16 (High byte)
```

Reply
```
1. MODBUS Address
2. MODBUS Function
3. Number of bytes written
4. CRC16 (Low byte)
5. CRC16 (High byte)
```

The 'Starting MODBUS bit address' is always rounded down to its byte boundary and so the number of bytes returned will encompass the number of bits requested. Hence the bits are written on their byte boundary (i.e. bit positions within the bytes are as their appear in the MODBUS Global map).
The status of the written bits is expected as: 1 = ON; 0 = OFF.
Note: If the number of bits to be written does not divide evenly into eight then the remaining bits, front and back padding bits, will reflect the status of those bits with the same remaining valid addresses in the Global MODBUS map.
Also note that the data to be written is expected

Example

Request:

Field	Example
MODBUS Address	5
MODBUS Function	0F
Starting MODBUS Address (High)	1
Starting MODBUS Address (Low)	51
Number of Bits (High)	0
Number of Bits (Low)	9
Data (Address 0150 – 0157)	C2
Data (Address 0158 – 015F)	01
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 must be calculated using the appropriate formula by the MODBUS master and added in the last word position (it takes up 2 bytes).

The data to be written is expected as byte-aligned bits as they would be read in the Global MODBUS map.
The first data byte contains the bits of the lowest 8 addresses, with the lowest or first address being represented by the LSB (Least Significant Bit). Bits of no interest should reflect the status that they should appear in the Global MODBUS map.

The CCU2/C-MXMB should return the number of coils successfully set.

Reply:

Field	Example
MODBUS Address	5
MODBUS Function	0F
Number of bytes written	0
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 is calculated by the CCU2/C-MXMB and takes up 2 bytes.

4.2.4. Preset Multiple Registers

The Preset Multiple Registers command allows the MODBUS master to write to the MODBUS map using the register address of the bits to be set.

The byte fields of this command type are as follows:

Request

10. MODBUS Address
11. MODBUS Function
12. Starting MODBUS bit address (High byte)
13. Starting MODBUS bit address (Low byte)
14. Number of registers/words (High byte)
15. Number of registers/words (Low byte)
16. Data (number of bytes needed to write the appropriate data)
17. CRC16 (Low byte)
18. CRC16 (High byte)

Reply

6. MODBUS Address
7. MODBUS Function
8. Number of bytes written
9. CRC16 (Low byte)
10. CRC16 (High byte)

Note: Data to be written is expected as low byte – high byte order.

Example

Request:

Field	**Example**
MODBUS Address	5
MODBUS Function	10(hex)
Starting register (High)	0
Starting register (Low)	3
Number of Registers (High)	0
Number of Registers (Low)	2
Data (Register 3 (Low))	1D
Data (Register 3 (High))	F0
Data (Register 4 (Low))	12
Data (Register 4 (High))	69
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 must be calculated using the appropriate formula by the MODBUS master and added in the last word position (it takes up 2 bytes).

This command will write register 3 and register 4 on MODBUS device 5.

The CCU2/C-MXMB will return the number of bytes successfully written.

Reply:

Field	**Example**
MODBUS Address	5
MODBUS Function	10(hex)
Number of bytes written (High)	0
Number of bytes written (Low)	4
CRC16 (Low)	???
CRC16 (High)	???

NB. CRC16 is calculated by the CCU2/C-MXMB and takes up 2 bytes.

4.2.5. Calculating CRC16

Step 1 - Start with a 16-bit register set to FFFF hex. Call this the CRC16 register.

Step 2 - Exclusive OR the first eight-bit byte of the message with the low byte of the CRC16 register, putting the result in the CRC16 register.

Step 3 - Shift the CRC16 register one bit to the right, shifting in a zero for the MSB.

Step 4 - If the LSB in the CRC16 register is 0, repeat Step 3. If the LSB is 1, Exclusive OR the CRC16 register with A001 hex (1010 0000 0000 0001).

Step 5 - Repeat Steps 3 and 4 until eight shifts have been performed. When this is done, a complete eight-bit byte will have been processed.

Step 6 - Repeat Steps 2 ... 5 for the every byte in the message.

Step 7 – Place the CRC16 value in the message.

4.3. Point Information Request

The ‘Point Information Request’ registers in the CCU are located at the address positions as specified in the MX MODBUS Specification V1.0. That is, at offset address 325 in each of the panel bits the ’Point Information Request’ bits start.
NOTE: When performing a write to the MODBUS the data sent to the CCU must be in Lo-Hi byte format. Hence when writing to the ‘Point Information Request’ addresses be sure to send the data to be written in the CCU as it would appear in the registers (from register 325 to register 348, Lo-Hi).

4.4. Isolate De-isolate Request

As with the ‘Point Information Request’ the ‘Isolate De-isolate Request’ is written to the CCU in the same fashion, Lo-Hi byte. However the starting address of the ‘Isolate De-isolate Request’ data is at 372.

4.5. CCU Timekeeping

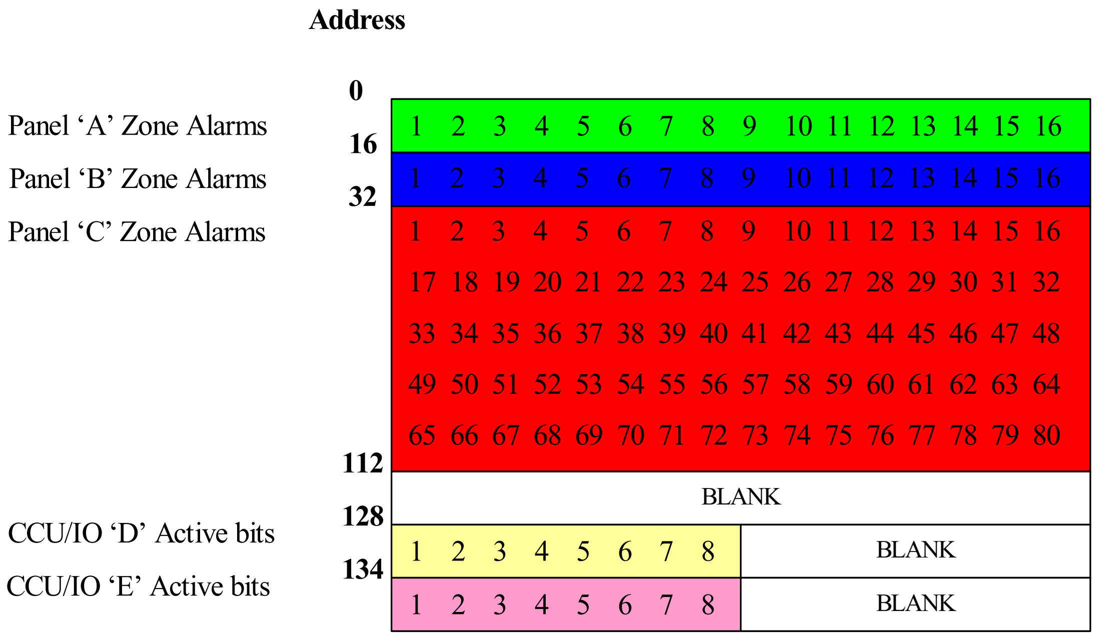
There are two ways to configure the MXNet timekeeping with reference to the CCU2/C-MXMB. The first is to configure one of the panels on the MXNet to be timekeeper. Once the time has been set correctly on this panel it will broadcast the time and date to the MXNet and all nodes on the net will use its time to keep time (including the CCU). The second way is to enable the CCU as timekeeper. On power up the CCU has a default time and date that will be used if configured as the timekeeper. In order to set the correct time and date set the appropriate bits in the time and date registers, in the MODBUS map, and the CCU will set its own time to this and use this time to perform timekeeping on the MXNet. Please see section 4.1 MX CCU Interface document for date and time format.

5. *Example CCU2/C-MXMB Setup*

This section will step through an example configuration. The example configuration will map five devices (labelled ‘A’, ‘B’, ‘C’, ‘D’ and ‘E’):

- 3 x MX panels connected to the MXNET (‘A’, ‘B’ and ‘C’).
- 2 x CCU/IO boards on the CCUioNET (‘D’ and ‘E’)

The MODBUS map that we will require for this example project will list all of the zone alarms (excluding zone 0, the system zone) of the three MX panels in the start of the address map. Leave some blank addresses and then have all of the active bits from CCU/IO ‘D’. Next we will have 8 more blank addresses then active bits from CCU/IO from CCU/IO ‘E’.



	Address																
Panel ‘A’ Zone Alarms	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Panel ‘B’ Zone Alarms	16	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Panel ‘C’ Zone Alarms	32	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
		33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
		49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
		65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
	112	BLANK															
CCU/IO ‘D’ Active bits	128	1	2	3	4	5	6	7	8	BLANK							
CCU/IO ‘E’ Active bits	134	1	2	3	4	5	6	7	8	BLANK							

Figure 23: The desired MODBUS Map

5.0.1. Set the MODBUS address

The first task is to set the MODBUS address of the CCU2/C-MXMB. To do this select **Edit | Change MODBUS address** and enter the address.

5.1. Set up the Global Map

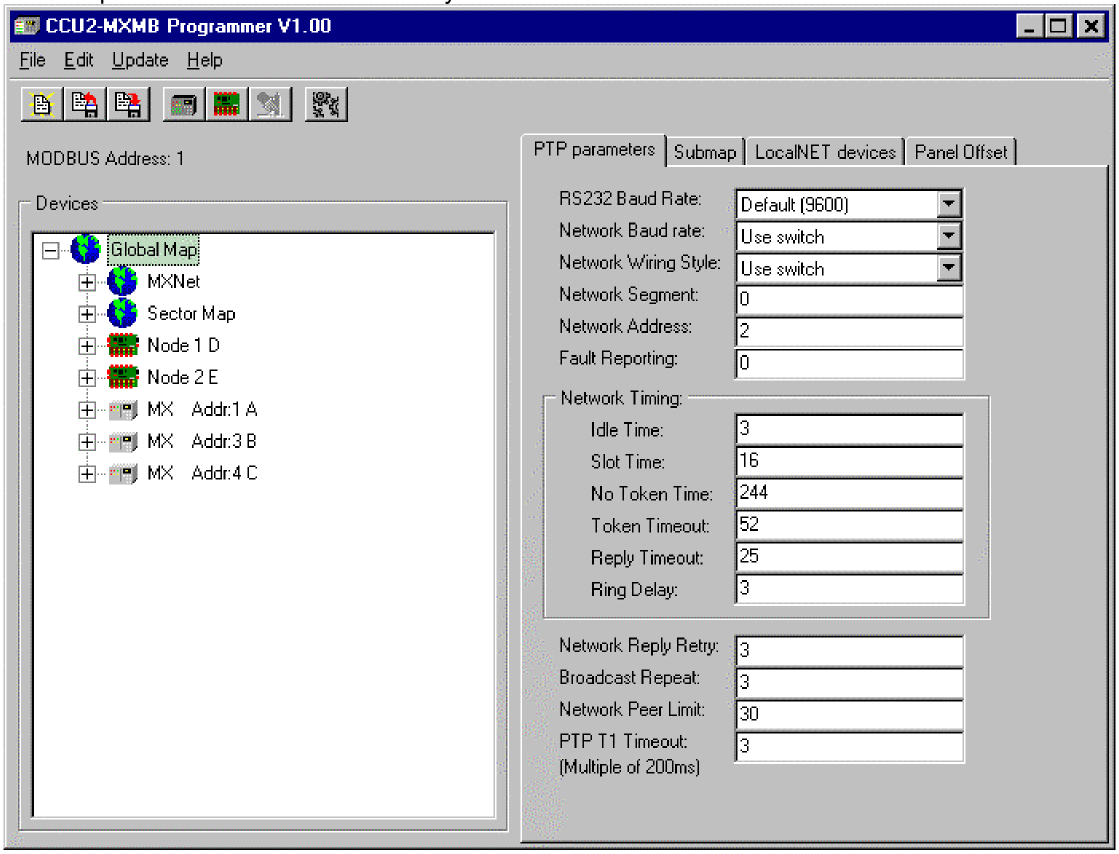
5.1.1. Add the CCUioNET nodes

Our example system has two CCUioNET nodes- the CCU/IO-485 boards ‘D’ and ‘E’. To add a node press the **Add CCUioNET Node** button or select **Edit | Add CCUioNET node** from the Main Menu. Select CCU/IO from the list of CCUioNET devices. Press **OK**. The new CCUioNET node should be listed in the **Global Map Tree** on the left hand side of the screen. You may need to expand the tree to see the node, to do this press the ‘+’ next to ‘Global Map’ and also the ‘+’ next to ‘CCUioNET nodes’. To add a description to this node so it is not confused with other nodes double-click on the node in the **Global Map Tree**. The node number in the **Global Map Tree** should match the address on the CCU/IO board. So CCU/IO board ‘D’ has address 1 on the CCUioNET.
Add CCU/IO board ‘E’ in a similar way.

5.1.2. Add the MX panels

Our example system has three MX panels, ‘A’, ‘B’ and ‘C’. To add a panel press the **Add MX panel** button or select **Edit | Add MX panel** from the Main Menu. The **MXNet Address** box must be filled in with the address of the MX panel on the MXNet. Press **OK**. If you have already entered a panel with this address an error message will be displayed, otherwise the new MX should be listed in the **Global Map Tree** on the left hand side of the screen. You may need to expand the tree to see the node, to do this press the ‘+’ next to ‘Global Map’. To add a description to this panel double-click on the panel in the **Global Map Tree**.

Add MX panel ‘B’ and ‘C’ in a similar way.



MODBUS Address: 1

- Global Map
  - MXNet
  - Sector Map
  - Node 1 D
  - Node 2 E
  - MX   Addr:1 A
  - MX   Addr:3 B
  - MX   Addr:4 C

RS232 Baud Rate: Default (9600)
Network Baud rate: Use switch
Network Wiring Style: Use switch
Network Segment: 0
Network Address: 2
Fault Reporting: 0

Idle Time: 3
Slot Time: 16
No Token Time: 244
Token Timeout: 52
Reply Timeout: 25
Ring Delay: 3

Network Reply Retry: 3
Broadcast Repeat: 3
Network Peer Limit: 30
PTP T1 Timeout: (Multiple of 200ms) 3

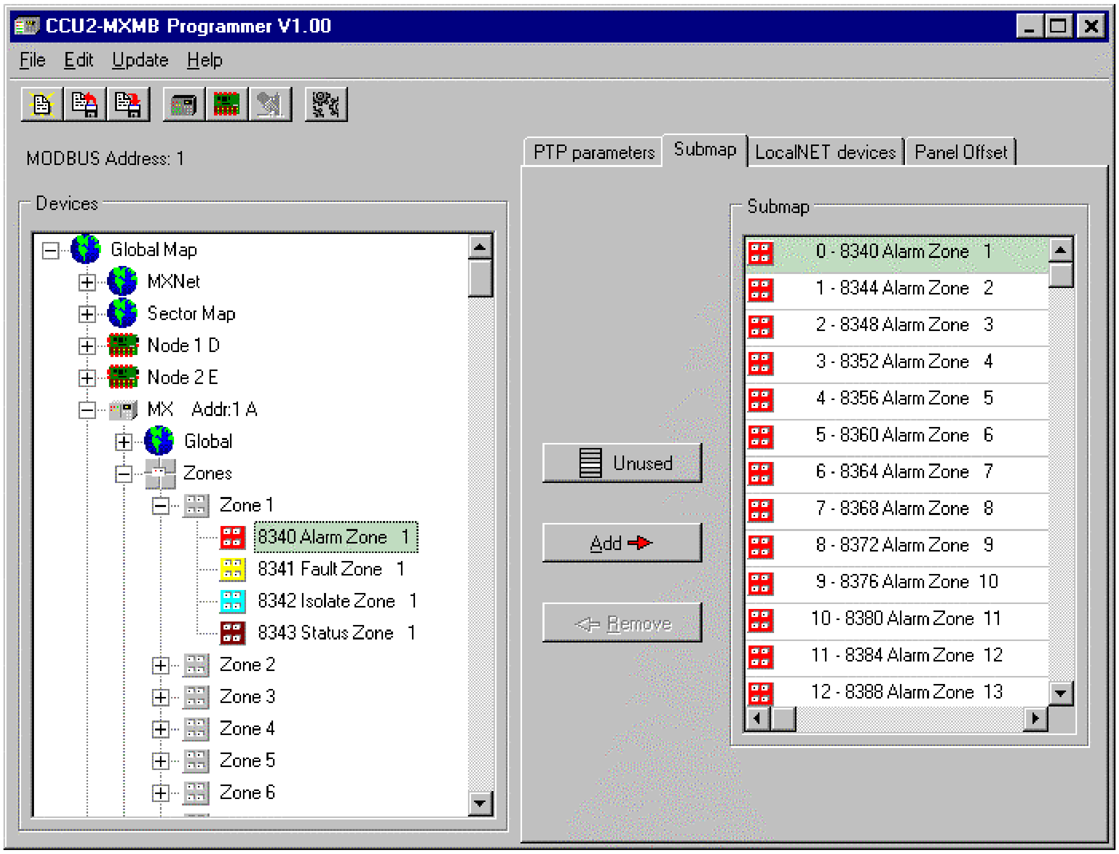
Figure 24: GLOBAL Tree for Sample MXNet configuration MODBUS Map

5.2. Set up the Submap

At this stage the Global Map is completely set up. Now all that is left is to create a submap of points that are useful to us from the global map.

5.2.1. Adding the MX Zone Alarms to the Submap

The appropriate bits from the Global Map must be added to the submap in the desired order. For us, this means that Panel ‘A’ Zone Alarm 1 should be added first. To do this open the **Submap** page and expand the **Global Map Tree** to be able to see the Zone Alarm 1. It should come under Global Map – MX (Device ‘A’) – Zones – Zone 1. Select Alarm Zone 1 by clicking on it. The **Add** button should change so it is no longer ‘greyed out’. Click the **Add** button, the bit should appear in the submap on the right-hand side of the screen. Repeat this for all of the Zone Alarms on all three MX panels, making sure that they stay in order in the submap.



0 - 8340 Alarm Zone 1
1 - 8344 Alarm Zone 2
2 - 8348 Alarm Zone 3
3 - 8352 Alarm Zone 4
4 - 8356 Alarm Zone 5
5 - 8360 Alarm Zone 6
6 - 8364 Alarm Zone 7
7 - 8368 Alarm Zone 8
8 - 8372 Alarm Zone 9
9 - 8376 Alarm Zone 10
10 - 8380 Alarm Zone 11
11 - 8384 Alarm Zone 12
12 - 8388 Alarm Zone 13

Figure 25: Adding the Zone Alarm bits to the submap

5.2.2. Adding the Blank addresses to the Submap

In the coarse of laying out a submap often there are addresses that are not needed or used and so are left blank. To do this select **Unused Entry** from the **Global Map Tree** and press Add to add it to the submap. As many unused addresses as are required may be added to the submap. However, unused bits still take up space in the 1000 bit submap.

Add the desired 16 unused bits to the example project.

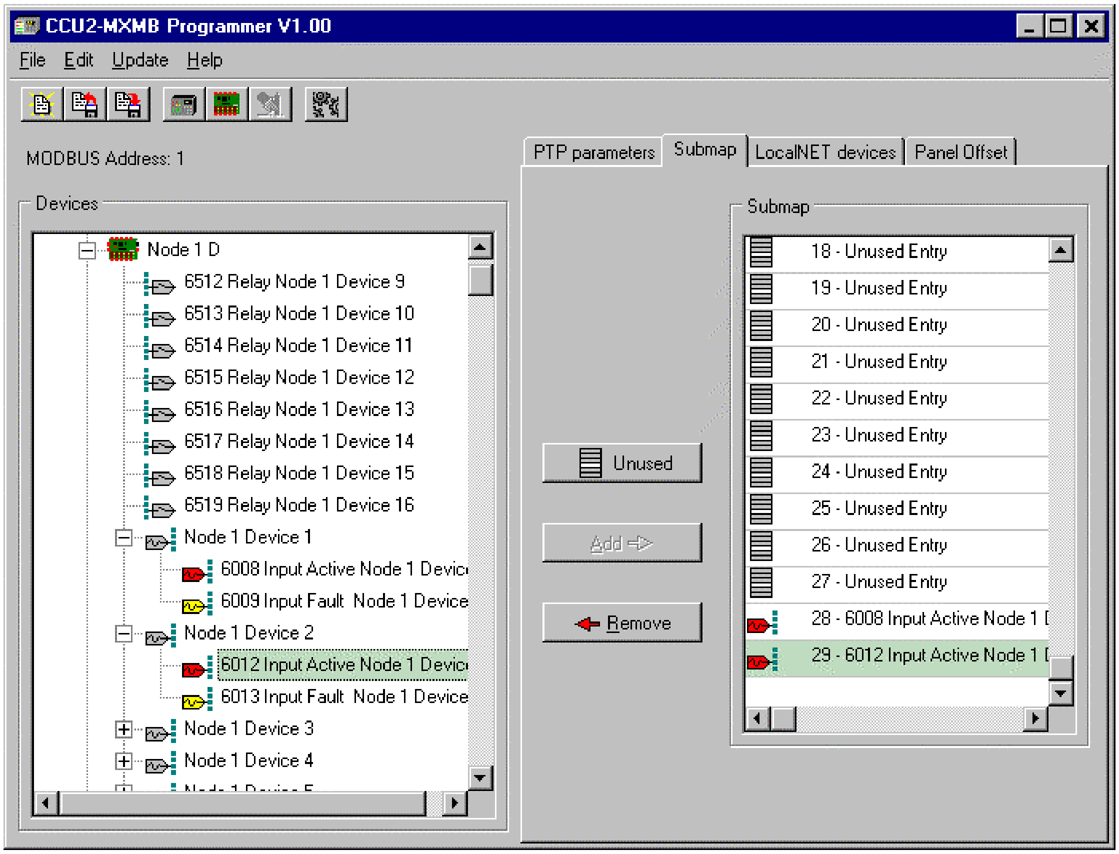
5.2.3. Adding the CCU/IO Active bits to the Submap

The CCU/IOs have two bits for every input on the board. One indicates a supervision fault on the input and the other indicates that the input has been set to its active state. For the example project the eight Active bits of CCU/IO ‘D’ and ‘E’ are added to the submap with eight blank spaces between them.

To add the Active bits to the example project expand the **Global Map Tree** to be able to see the Input Active Node 1 Device. It should come under Global Map – CCUioNET Nodes – Node 1 (Device ‘D’) – Node 1 Device 1 – Input Active Node 1 Device. Select Input Active Node 1 Device by clicking on it. The **Add** button should change so it is no longer ‘greyed out’. Click the **Add** button, the bit should appear in the submap on the right-hand side of the screen. Repeat this for all of the Active bits on the CCU/IO, making sure that they stay in order in the submap.

Add eight unused (blank) bits.

Add the eight Active bits for CCU/IO ‘E’.



MODBUS Address: 1

Node 1 D
6512 Relay Node 1 Device 9
6513 Relay Node 1 Device 10
6514 Relay Node 1 Device 11
6515 Relay Node 1 Device 12
6516 Relay Node 1 Device 13
6517 Relay Node 1 Device 14
6518 Relay Node 1 Device 15
6519 Relay Node 1 Device 16
Node 1 Device 1
6008 Input Active Node 1 Devic
6009 Input Fault Node 1 Device
Node 1 Device 2
6012 Input Active Node 1 Devic
6013 Input Fault Node 1 Device
Node 1 Device 3
Node 1 Device 4
Node 1 Device 5

18 - Unused Entry
19 - Unused Entry
20 - Unused Entry
21 - Unused Entry
22 - Unused Entry
23 - Unused Entry
24 - Unused Entry
25 - Unused Entry
26 - Unused Entry
27 - Unused Entry
28 - 6008 Input Active Node 1
29 - 6012 Input Active Node 1

Figure 26: Adding the Input Active bits to the submap

5.3. Save the project and Program the CCU2/C-MXMB

Now both the global map and the submap are totally set up. Before programming the CCU2/C-MXMB it is a good idea to save your work, so that it can be used again at a later date, or for reference purposes.

5.3.1. Saving

To save all of the information that has been entered so far press the **Save** button or select **File | Save** from the Main Menu. To restore it at a later date press the **Load** button or select **File | Load** from the Main Menu.

5.3.2. Programming

Connect the CCU2/C-MXMB to the computer using a straight through standard communications port cable (see Appendix C) from the desired com port to Port 1 on the CCU2/C-MXMB.

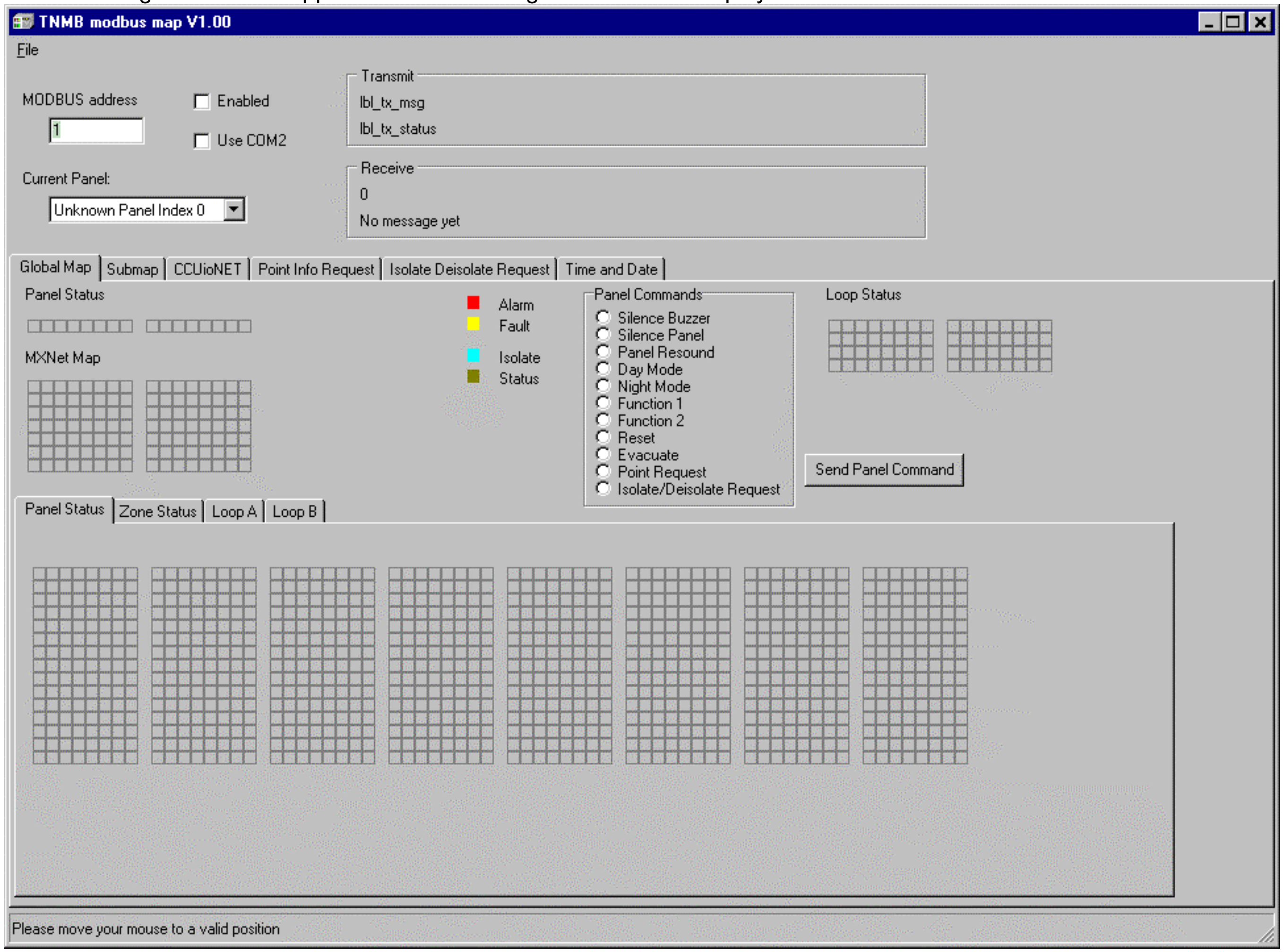
Select the **programming** screen and set the **Comport** to the same port that the CCU2/C-MXMB is connected to.

Before programming is commenced the comport must be set in the **CCU2-MXMB Setup** program to the port that the CCU2/C-MXMB is connected to. To do this select **Update | Select COM Port** from the Main Menu.

Press the **Update Data** button or select **Update | Data only** from the Main Menu. Programming should commence and once all of the lines have been sent a success message should be displayed.

5.4. Testing the Configuration

Once the configuration has been downloaded to the CCU it can be tested using the supplied test application modbus.exe. To use this program the CCU and panels must be on the MXNet with all of the relevant cabling installed.
When running the modbus application the following screen will be displayed.



TNMB modbus map V1.00

MODBUS address: 1

□ Enabled

□ Use COM2

Current Panel: Unknown Panel Index 0

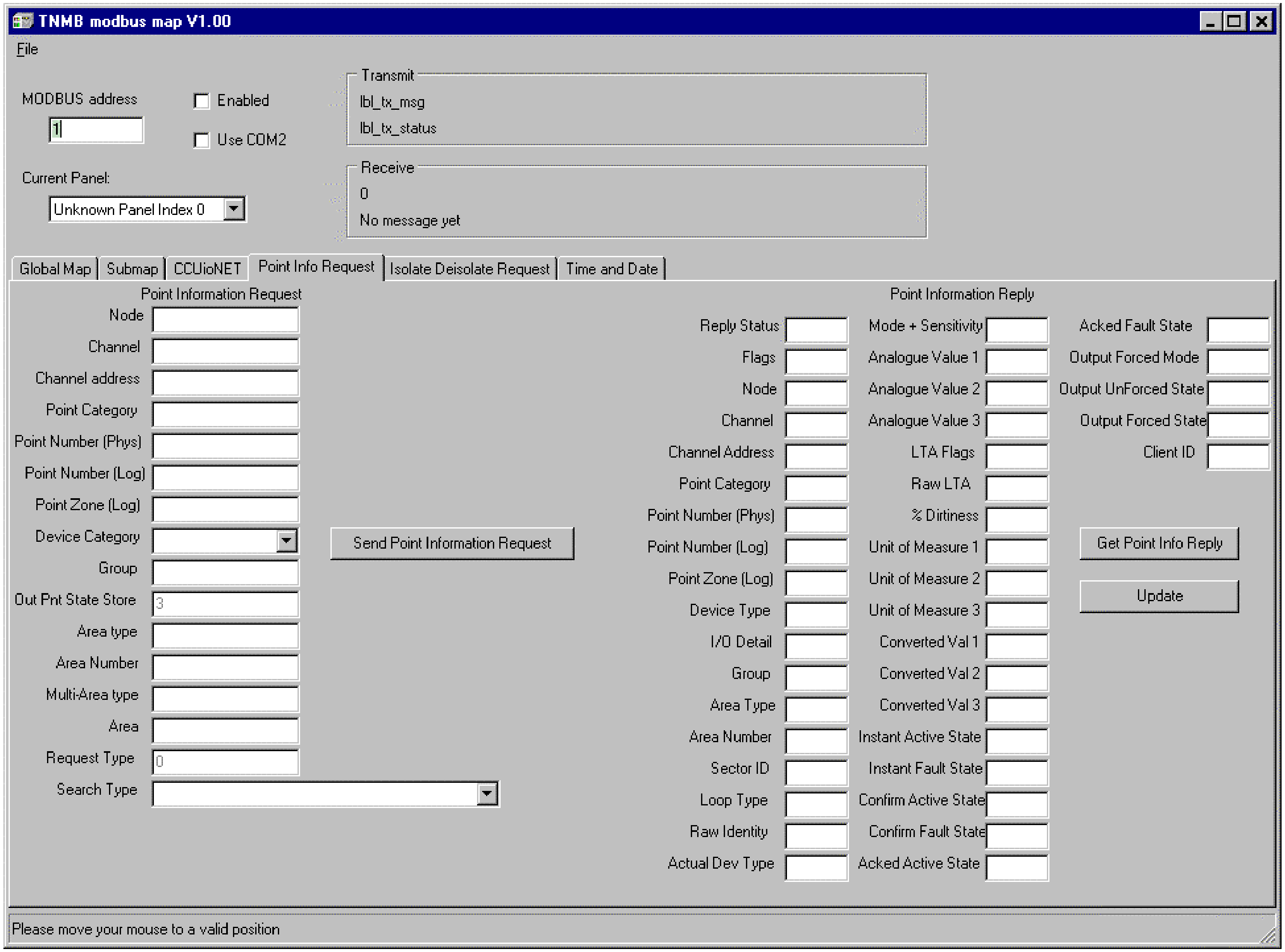
Transmit
lbl_tx_msg
lbl_tx_status

Receive
0
No message yet

Please move your mouse to a valid position

The **File | Open** menu allows the TNM file that was saved in the MXMB configuration program to be opened. Locate the TNM file and open it. The panels that were added into the configuration will be displayed in the *Current Panel* drop-down list box. Select the panel that you wish to monitor with the list box. Place a check in the box beside the 'Use COM2' text if you have connected the CCU to COM2 else COM1 will be used by default. Place a check in the box next to the 'Enabled' text to start the polling of the CCU Global Map. If the cable has been wired and connected correctly there will be packets displayed in the transmitted and received boxes at the top of the screen. Each of the tab sheets display the bits in the map that are being polled at present. The different sections of the map have been separated into different areas in the window. To identify a bit which is activated in the map move your mouse over the bit and the name will be displayed in the status bar on the bottom of the window. This will work with all of the bits in each of the tab windows.
To send commands to the panel ensure that the 'Enabled' check box is ticked. Select the function to be sent to the panel by clicking on the radio buttons in the 'Panel Commands'

section. Then click the ‘Send Panel Command’ button.
The Submap and CCUioNET tab sheets will poll the appropriate bit in the MODBUS map and display them to the screen.



MODBUS address: 1

☐ Enabled

☐ Use COM2

Current Panel: Unknown Panel Index 0

Transmit
lbl_tx_msg
lbl_tx_status

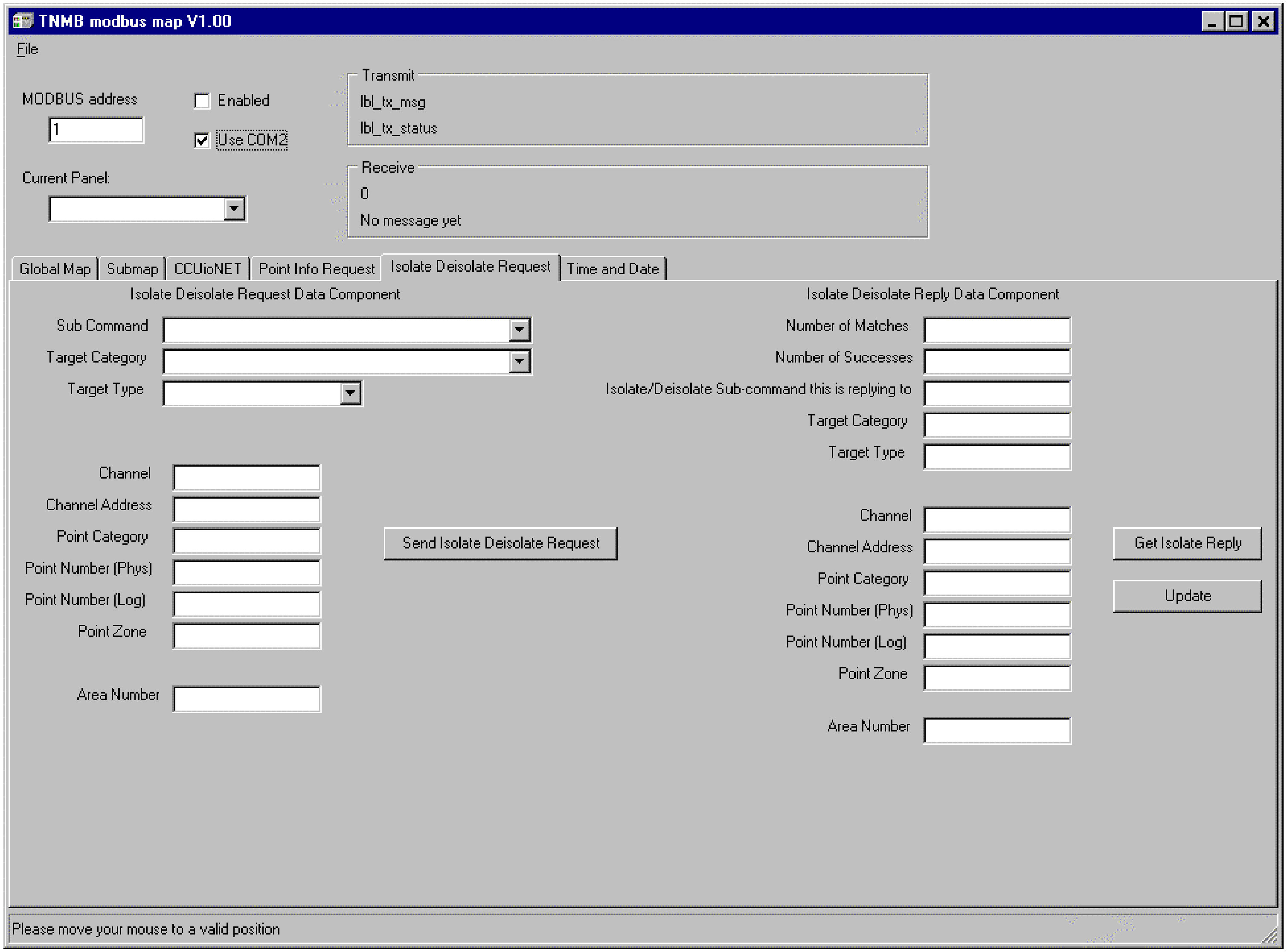
Receive
0
No message yet

Out Pnt State Store: 3

Request Type: 0

Please move your mouse to a valid position

To send a Point Information Request use the Point Info Request tab to enter in the packet data to be sent to the panel. Enter the values in the text edit boxes on the left of the window and click the Send Point Information Request button. This will write those values to the appropriate area in the MODBUS map. Returning to the Global map tab sheet the ‘Point Request’ command can be sent to the CCU to inform it to send the packet to the panel. To get the reply go back to the Point Info Request tab sheet and click the Get Point Info Reply. To display the reply to the screen click the Update button.



TNMB modbus map V1.00

MODBUS address: 1

☐ Enabled

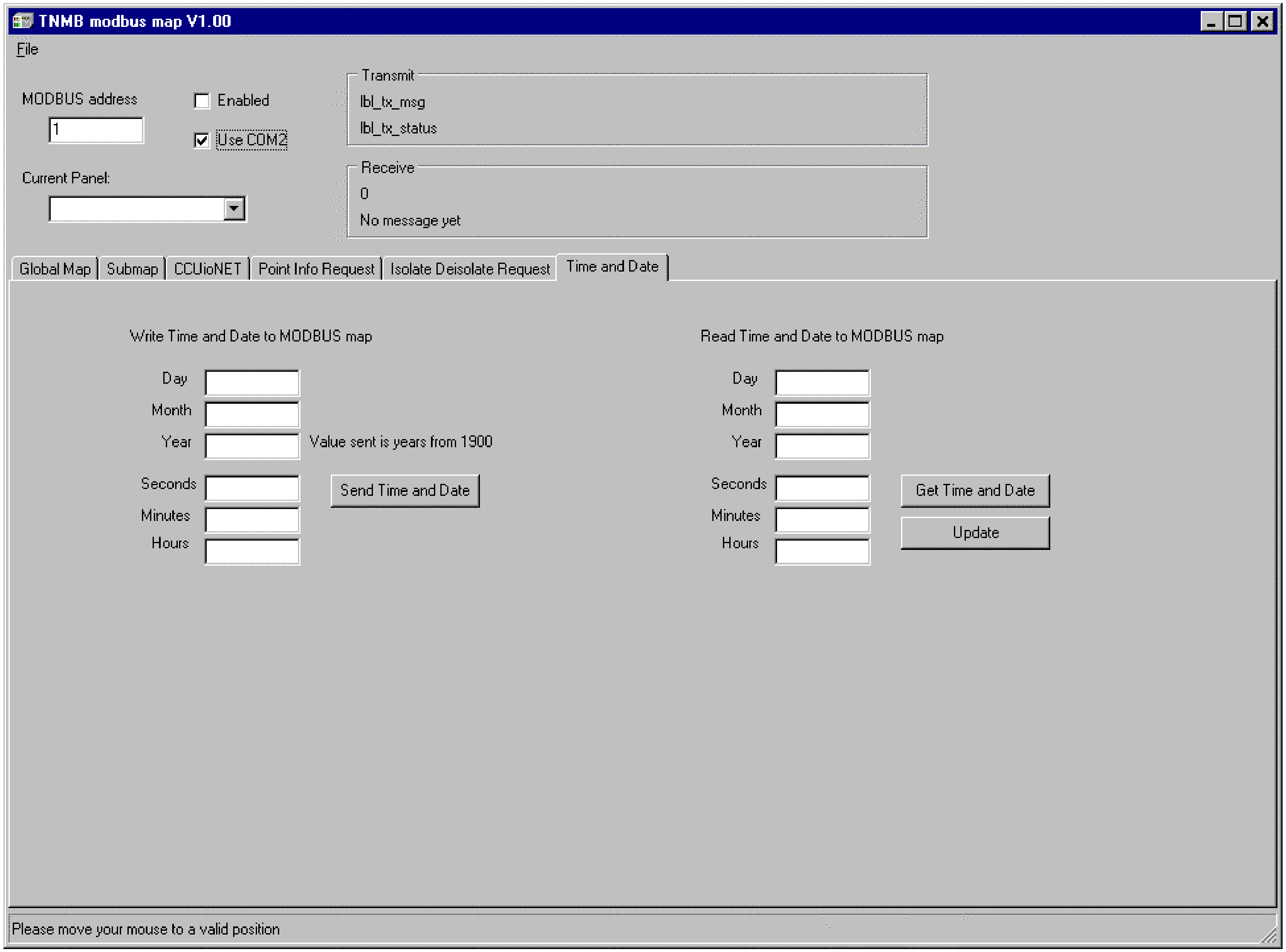
☑ Use COM2

Transmit
lbl_tx_msg
lbl_tx_status

Receive
0
No message yet

Please move your mouse to a valid position

To send an Isolate/Deisolate Request refer to the previous description of how to send a Point Information Request and use the same process for the Isolate/Deisolate Reply.



MODBUS address 1

☐ Enabled

☑ Use COM2

lbl_tx_msg

lbl_tx_status

0

No message yet

Value sent is years from 1900

Please move your mouse to a valid position

For Time and Date information use the Time and Date tab sheet as seen in above.

To set the time and date in the CCU enter the values in the text edit boxes on the left hand side click the Send Time and Date button. NOTE: Enter in the real year (eg. 2001), however the actual value written to the CCU is years from 1900.

To read the time from the CCU click the Get Time and Date button. This will read the appropriate bits from the MODBUS map. Use the Update button to display the time to the screen