USER'S MANUAL

Maxiflex P3
Process Automation Controller CPU
User's Manual

Date	Revision	Comments
March 2003	1	Initial Issue
March 2003	2	Ethernet configuration updated
April 2003	3	Minor Corrections
April 2003	4	Ethernet Addressing updated
April 2003	5	Battery Link information added
April 2003	6	Typographical corrections
May 2003	7	DITcopy Function Block updated and Network DIP Switch not active on A versions of the product.
April 2004	8	Added Subscriptions, Modbus Master and ISaGraf programming extension. Model 'D'.
September 2004	9	Added more Content for Modbus Master. Added new DIT access function blocks to ISaGraf programming extensions.
March 2005	10	Added remote I/O support & redundant options.
October 2005	11	Added Network Offset screen grab to section 10.4.2
August 2006	12	DIT Numbers in Local & Remote Racks Corrected
November 2006	13	Added M1760/1 32 SOE Module, Queue Service
December 2013	14	Corrected Battery Link Default Connection
September 2014	15	Increased supported number of Remote Racks. Updated hardware revisions.
May 2016	16	Added reference to the Maxiflex A3e CPU.
October 2018	17	Upped the CPU versions due to TSM functionality
February 2019	18	Previous version showed markup
July 2020	19	Rectified incorrect battery link description in Figure 4.1

SOFTWARE COPY AVAILABLE

This manual is available in printed form or in Adobe Acrobat pdf format.

The pdf file is named UMM126XR19.pdf

SCOPE

This User Manual provides information on how to install, configure and use the full range of Maxiflex P3 Process Automation Controller CPU's.

Note : This manual also applies to the Maxiflex A3e CPU (Maxilarm) which is based on the P3 CPU. The standard features are common to both CPU types. For specific Maxilarm functionality, please refer to the Maxiflex A3 User Manual.

This manual does not cover the fundamentals of the IEC61131-3 programming languages, nor the Omniflex ISaGraf Programmer's Workbench. This information is available separately in the manuals supplied with that product, but programming features specific to the P3 CPU's are included in this manual.

This manual covers the following product Models:

Model	Description
M1260F	P3 CPU with RS232/485 Serial Port
M1261F	P3c CPU with RS232/485 Serial Port and Conet/c Twisted Pair Network Port.
M1262G	P3e CPU with RS232/485 Serial Port and Conet/e 10/100 Ethernet Network Port
M1264B	A3c CPU with RS232/485 Serial Port and Conet/c Twisted Pair Network Port .(standard features only)
M1265B	A3e CPU with RS232/485 Serial Port and Conet/e 10/100 Ethernet Network Port (standard features only)
M1267C	P3e-R CPU with RS232/485 Serial Port and 10/100 Ethernet Port.

Introduction

The MAXIFLEX P3 & A3 range of Process Automation Controller (PAC) CPU's is designed for general industrial control applications including process control, PLC, telemetry and remote I/O applications. These CPU's combine powerful industrial network communications capabilities with ease of use and powerful programming features.

All I/O and configuration data variables are automatically accessible through up to 65,000 Data Interchange Registers in a single virtual "Data Interchange Table", allowing the implementation of Remote I/O systems "out of the box" without programming.

For local control and data manipulation, the Maxiflex P3 & A3 PAC CPU's can be programmed in one or more of the standard IEC61131-3 programming languages using the powerful Omniflex ISaGraf Programmer's Workbench.

Many other features such as a built-in real-time clock, battery backup for temporary dynamic data, and a MODBUS (Master or Slave) or Conet/s equipped RS232/485 serial port are standard in these products.

Powerful features such as automatic I/O module recognition and scanning, remote programming, and a versatile Remote Data Subscription Service, all contribute to making the P3 CPU a "plug-and-work" product that dramatically reduces system engineering time.

Following the ISO OSI 7-layer model, these CPU's includes powerful inter-network routing capability for retrieving data from the corners of the factory in very large, geographically spread-out installations. This capability allows many dissimilar network types to be linked to create a seamless factory intranet, quite often without the need to install special network cabling.

The MAXIFLEX P3-R CPU's are designed specifically for redundant remote I/O and control applications, where high availability is required in hot standby configurations.

Table of Contents

Table of Figures

Table of Tables

1. Glossary of Terms

The following terms are used throughout the text of this document:

Active Mode	An operating mode of one of the two dual redundant controllers in a dual redundant control system. The active controller in a dual redundant controller system has control of the redundant communications to the Supervisory System, and is the controller currently in control of the process I/O via its user program.
Base	The Maxiflex Base is the product into which you plug the Power Supply, CPU, I/O Modules and Network Interface Modules in a Maxiflex System. A Maxiflex Base equipped with its modules is referred to as a Maxiflex Rack
Change-over	The term used to describe the moment when one controller in a dual redundant controller system moves into the Active State while the other controller moves out of the Active State.
CPU	Central Processing Unit CPU's are a type of Maxiflex Module. Every Maxiflex Master Base must have one CPU module plugged into the CPU Slot.
DIT	Data Interchange Table The Data Interchange Table is a list of 65,535 data registers of 16 bits each that contain the dynamic and configuration data for the product. Each register is accessed by its DIT Address in the table (0 to 65535). The DIT is used as a focal point for the interchange of data between all the functions in the system.
Failure Mode	When a controller in a dual redundant system is inoperable, then it enters this mode i.e. some condition has occurred that means the unit cannot be used in either Active or Standby mode. This usually means hardware failure of some kind.
Field Network	The Field Network connects the Supervisory System to the Maxiflex Process Automation Controller. Other devices, including other Maxiflex Controllers can be connected to the field network. The field network is not dual redundant. If the field network is Ethernet, then redundancy can be implemented using Ethernet networking techniques. The field network must be connected to both the primary and secondary controllers at all times. The Field Network can be either (or both) of the following: • Serial RS485 using Modbus Slave protocol • Ethernet using the Conet/e or Modbus/TCP protocols (available only on the M1267B P3e-R CPU)
I/O	Inputs and Outputs The I/O of a system is the electrical interface to Input and Output electrical signals in the system.

IOM	I/O Module I/O Modules provide the interface to the electrical signals being monitored or controlled.
LED	Light Emitting Diode LED's are a type of indicator light commonly used in electronic products to display a piece of information to the user. LED's are more popular than older style filament lamps, because they produce far less heat, and never wear out.
NIM	Network Interface Module There is a special class of module that can be installed into any I/O Slot of a Maxiflex base to extend the networking functionality of the CPU. These modules are called Network Interface Modules or NIM's. Different NIM's are required for different types of networks.
Port	Maxiflex products can connect to many different field networks. Each network connects to the product via a Network *Port*.
Primary Controller	The Primary Controller is the Controller in a dual redundant Controller system that will take control of the system whenever it is healthy. When the Primary Controller fails, the Secondary Controller will assume control automatically. You may write a User application in any of the IEC61131 programming languages using the Omniflex ISaGraf Workbench and download it to the Primary Controller.
Process Automation Controller	A Process Automation Controller (PAC) consists of a conventional Maxiflex Base equipped with a Power Supply, P3 CPU and local I/O Modules (if required). If this Controller is required to communicate with Remote I/O racks, then it will also be equipped with an M1592A Remote I/O Network Interface Module (RIO NIM). The Process Automation Controller performs the Main Process and Logic Control for the system and acts as the interface to the Supervisory System. To construct a Maxiflex system with Dual Redundant Process Automation Controllers, two Controllers would be implemented, one acting as Primary Controller, the other as Secondary Controller, both communicating with common I/O in remote I/O Racks. The Process Automation Controller is often just referred to as the Controller or PAC System.
PSU	Power Supply Unit PSU's are a type of Maxiflex module used to power all the other modules in a Maxiflex system. PSU modules plug into the special PSU Slots on the Maxiflex Base.
Rack	A Maxiflex Rack consists of a Maxiflex Base equipped with all its modules.
Remote I/O Link	This link connects the Process Automation Controller to the Remote I/O Racks in a system equipped with Remote I/O. Redundant Controller systems with shared I/O MUST use Remote I/O for their common plant interface.
Remote I/O Rack	A Remote I/O Rack consists of a conventional Maxiflex Base, Power Supply and I/O modules, with a M1248A Remote I/O

	Scanner in the CPU slot. The Remote I/O Scanner connects to the main Controller via a Remote I/O Link. The Remote I/O Scanner automatically scans the I/O on its Maxiflex Base and communicates with the Controller CPU to include the remote I/O in the Controller CPU's I/O list as if it were local to the CPU.
SCADA	Supervisory Control and Data Acquisition The term *SCADA* is often used to identify the supervisory control software that it used to monitor and control a plant or process – hence *SCADA software*
Secondary Controller	The Secondary Controller is the Controller in a dual redundant Controller system that remains on standby while the Primary Controller is in control. The Secondary Controller monitors the health of the Primary Controller and assumes control automatically if the Primary Controller fails. You may write a User application in any of the IEC61131 programming languages using the Omniflex ISaGraf Workbench and download it to the Secondary Controller. The user program in the Secondary Controller can, but does not need to be, the same program as that in the Primary Controller.
Slot	All Maxiflex Modules plug into "Slots" on a Maxiflex Base. There are different types of slots: PSU Slots contain PSU Modules CPU Slots contain CPU Modules I/O Slots contain IOM and NIM Modules
Standby Mode	An operating mode of one of the two dual redundant controllers in a dual redundant control system when it does not have control but is available to take control should it be necessary.
Subscription Service	The subscription service is a common method employed in many Omniflex products to efficiently replicate data between nodes on a network. A block of data in a DIT is replicated in another product DIT across the network by simply configuring a "subscription". The receiving node is said to "subscribe" to the transmitting node's data.
Supervisory System	This is a third party system that is used to monitor and control the plant or process via the Maxiflex system. This system is often referred to as the *SCADA* system
Sync Block	A *Sync Block* is a contiguous block of registers within the *Data Interchange Table (DIT)* of the *controller* that are synchronised between *Primary* and *Secondary Controllers* in a dual redundant controller system. Up to three contiguous blocks of DIT registers can be defined by the user as *Sync Blocks* and will be synchronised between controllers in a dual redundant controller system.
Sync Link	This is a dedicated communications link between the CPU's in the Primary and Secondary Controllers. It is used to synchronise User program registers, I/O registers and program execution.

TCP Port	Unlike real Network ports, TCP ports are virtual interfaces within the product required by the TCP/IP protocols. Each *process* that wants to communicate with another process via TCP/IP identifies itself to the TCP/IP protocol suite by one or more *ports*. A port is a 16-bit number, used by the host-to-host protocol to identify to which higher-level protocol or application program (process) it must deliver incoming messages.

2. General Description of the P3 & A3 CPU's

This manual covers four models of P3 and two kodels of the A3 CPU. The following is a picture of just one of these models:

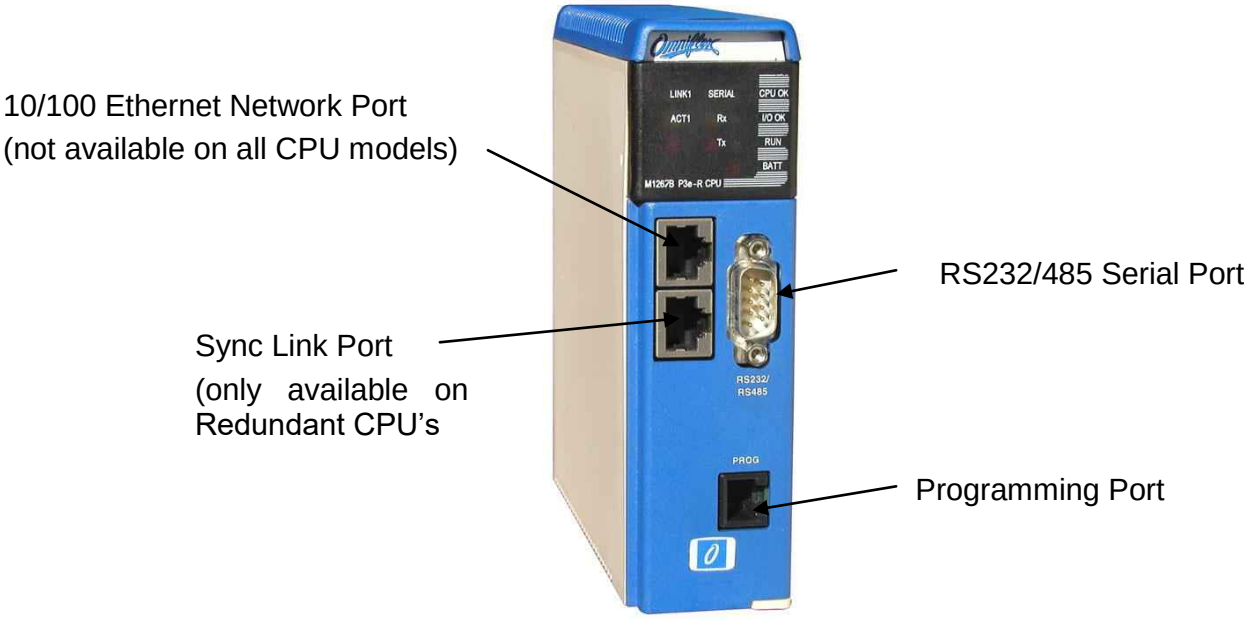


Figure 2.1: View of the M1267B P3e-R Redundant Controller CPU with 10/100 Ethernet

2.1 Front Panel LED Indicators:

The following table identifies which CPU's have which LED Indicators. A full description of the function of each indicator is given in section 4.10

NAME	COLOUR	M1260F P3	M1261F P3c	M1262G P3e	M1267C P3e-R
CPU OK	Green	Y	Y	Y	Y
I/O OK	Green	Y	Y	Y	Y
RUN	Green	Y	Y	Y	Y
BATT	Red	Y	Y	Y	Y
SERIAL Rx	Green	Y	Y	Y	Y
SERIAL Tx	Red	Y	Y	Y	Y
NETWORK Rx	Yellow		Y		
NETWORK Tx	Red		Y		
NETWORK TOKEN	Green		Y		
NET LINK	Green			Y	Y
NET 100	Yellow			Y	Y
SYNC LINK	Green				Y
SYNC 100	Yellow				Y

Table 2.1: Front Panel LED Indicators

2.2 Communication Ports

The following table shows the communications ports available on this range of CPUs.

Model	Description	Prog	Serial	Conet/c	Ethernet	Sync
M1260F	P3	Y	Y			
M1261F	P3c	Y	Y	Y		
M1262G	P3e	Y	Y		Y	
M1264B	A3e	Y	Y			
M1265B	A3e	Y	Y	Y		
M1267C	P3e-R	Y	Y		Y	Y

2.2.1 Programming Port

The programming port is an RS232 port that implements the Conet/s protocol. This port is used for software configuration, interrogation and program downloads using a Microsoft Windows compatible PC, Laptop or Handheld computer. This port is compatible with the Omniset configuration utility, and the Omniflex ISaGraf Programmer's Workbench.

2.2.2 Serial Port

All P3 CPU's are equipped with a serial port that can be used to connect to other serial devices. Connection can be either RS232 or RS485.

The serial port can be configured for one of two protocols:

a) Modbus Slave protocol (ASCII or RTU) is available on this port allowing easy connection to other third party products such as DCS or SCADA systems.

b) Modbus Master protocol (ASCII or RTU) is available on this port allowing easy connection to third party Modbus Slave devices.

c) Conet/s protocol (peer-to-peer) is available on this port for integration into Conet Intranets. Conet/s connection allows seamless network connection between devices with full network capability such as report-by-exception, time-stamped event streams and remote programming. The full-duplex nature of the Conet/s protocol makes efficient use of the serial channel. With the use of modems or other virtual circuits, efficient wide area networks can be easily constructed.

2.2.3 Conet/c Network Port

This port provides connection to a Conet/c industrial network. Conet/c is a true peer-to-peer local area network (LAN) that allows reliable data transfer between multiple nodes over long distances (up to 10km) using conventional twisted pair cabling found in most industrial plants.

2.2.4 Ethernet Network Port

This port provides connection to a 10/100 Ethernet network via a UTP connector on the front of the module. The Ethernet port supports two protocols running under the TCP/IP transport protocol:

- Modbus/TCP (Slave and Master)
- Conet/e (providing full support for all Conet features over Ethernet including remote programming, and time-stamping at source)

These two protocols can co-exist on the Ethernet link

2.2.5 Sync Link Port

This port provides connection to the other CPU in dual redundant applications. Use Link Cable M1812A to connect the Primary and Secondary CPU's via this port.

2.3 I/O Module support

The Maxiflex P3 CPU's can interface to both local and remote I/O. One local Maxiflex Rack and seven Remote I/O Racks can be supported to directly access a total of up to 3808 I/O points (based upon 32 channel I/O modules) on a single CPU.

Initially P3 CPU's supported three remote racks. This was upgraded to allow up to seven remote racks. The earliest versions that support seven remote racks are show in the table below:

Model	Description	Firmware Supporting Seven Remote Racks
M1260F	P3	V12.40 and above
M1261F	P3c	V12.40 and above
M1262G	P3e	V12.29 and above
M1264B	A3e	V12.40 and above
M1265B	A3e	V12.40 and above
M1267C	P3e-R	V12.40 and above

2.4 Dual Redundant Operation

The P3e-R CPU is designed to operate in dual redundant configuration. In this configuration each CPU must be mounted on its own local Master Base with its own Power Supply.

I/O Modules may be fitted to these local Bases, but these I/O modules will only be directly accessible by the CPU on this base.

If shared I/O modules are required, then remote Racks I/O must be used.

2.5 CPU Functions

The P3 CPU contains the following functions:

2.5.1 Real-Time Clock

These CPU's have a real-time clock as a standard function. This clock has the following features:

- Battery-backed to retain real time while power is off for up to 5 years.
- Resolution to 10 milliseconds
- Current Time available in the Data Interchange Table
- Last Power Up Time available in the Data Interchange Table
- Last Power Down Time available in the Data Interchange Table

2.5.2 I/O Scanning

This function is responsible for the automatically scanning of conventional I/O modules (IOM's) installed on the Maxiflex base, reading input data and placing it in the DIT, and writing data from the DIT to the output modules.

If a User program is running in the CPU, then the I/O scanning is synchronised with this User Program. If there is no User Program running, then the I/O is scanned every 10 milliseconds.

This I/O scanning therefore occurs whether there is a User Program running in the CPU or not, allowing data acquisition or remote I/O applications to be implemented without the need to write a User Program for the CPU. This feature significantly reduces system engineering time.

2.5.3 I/O Module Management

This function is responsible for continuously monitoring all slots of the Maxiflex I/O base, keeping track of the currently installed module types. This function also maintains a copy of any I/O module configuration in the CPU, allowing I/O modules to be changed without the need to reconfigure them.

2.5.4 Subscription Service

Central to many applications involving communications across networks is the need to replicate data between nodes on the network. This feature provides an easy to use but powerful data replication service between DIT's in the system, whether they are local or remote. This service provides change-of-state detection and error reporting for optimum performance and reliability.

2.5.5 Queue Service

The P3 CPU has the ability to queue time stamped events that can be generated by modules like the M1760/1 32SOE module or by user programs written specially for the purpose. Once stored in the queue, PC based clients such as the Omniflex Conet OPC Server are able to poll this queue over the network and extract the queued events for analysis. It is also possible for a user application to extract queued events and perform application specific processing such as converting the events to text strings and then send these strings to a printer connected to the serial port.

Typical uses for the Queue Service are Sequence of Event monitoring applications where the P3 provides a buffered interface between real time events detected at the front end (using M1760/1 32SOE modules) and the Conet OPC Server at the backend.

2.5.6 Conet Routing Service

Many systems are constructed of multiple networks to overcome the difficulties of topology or communication protocol conversion. The Routing service provides a means to seamlessly interconnect these networks into an integrated intranet so that any node in the system may be globally addressed from any other with no regard for its physical location.

The P3 CPU can act as a router in these systems, automatically routing data packets seamlessly between nodes on different networks.

2.5.7 Modbus Master Driver

The P3 CPU comes standard with a Modbus Master Driver that allows the CPU to connect to any third party Modbus Slave device. This driver will operate over the RS232/485 serial port that is available on all models of CPU, as well as over the Ethernet port (using Modbus/TCP) of the M1262x and M1267x models.

A combined set of up to 32 queries can be configured for the CPU as a system. There is no fixed allocation of queries per communications port where it is relevant.

2.5.8 IEC61131 User Programming

The P3 CPU's are programmable in all five IEC61131-3 programming languages. Applications range from simple data scaling to analogue control systems and programmable logic control.

The user program has access to the following features in the CPU:

- Direct access to I/O modules in the system through the program I/O
- Direct access to all Data Interchange Tables in the system.
- Remotely programmable through any of the ports configured for the Conet protocol.

3. CPU's and the MAXIFLEX Architecture

This section explains the general architecture of the Maxiflex system. More specific details relating to the configuration of the P3 CPU's can be found in later Chapters.

3.1 General System Architecture

A diagram of the general MAXIFLEX System Architecture is shown in Figure 3.1

This diagram shows the P3 CPU in a system with an Intelligent Module (in this case a Programmable Network Interface Module (NIM)) and a conventional I/O Module (IOM). NIM's and IOM's can be arranged in any order in the available I/O Module slots of the system.

3.1.1 Maxiflex Bases

Two types of Maxiflex Base are available – Master Bases and Expander Bases. Both Master and Expander Bases are available in a range of sizes (number of I/O Modules).

The P3 CPU must be mounted into the CPU Slot of a Master Base. Every Master Base also requires a Power Supply (PSU) Module mounted into the dedicated PSU Slot.

One Expander Base can be connected to a Master Base giving a maximum of 15 local I/O modules supported by a single P3 CPU.

Larger Expander Bases require an additional power supply to support the added I/O Module load.

3.1.2 I/O Modules

The Maxiflex system supports a wide range of conventional Input/Output modules (IOM's). Some of these modules require configuration. This configuration is done in the CPU, and the configuration is stored in the CPU.

The CPU manages the configuration of all IOM's. If an IOM is exchanged, then the CPU reconfigures the replacement module automatically. No additional user intervention is required to replace an IOM.

3.1.3 Data Interchange Table (DIT) Overview

Data Interchange Tables (DIT) are the "crossroads" for data in the Maxiflex system. A DIT is an array of 16 bit registers accessible from any function or communications port in the system.

The P3 CPU has a DIT in which all the configuration settings and dynamic data, including I/O can be accessed. Any exchange of data between functions in the CPU and with the outside world takes place through the CPU's DIT.

Intelligent I/O Modules also each have a DIT. A portion of the intelligent I/O modules' DIT Registers are overlaid on to the CPU DIT, and appear as if they are part of the CPU's DIT. This extended DIT addressing is used to directly access data in any intelligent modules such as Network Interface modules (NIM's) installed on the Maxiflex base as if the data is in the CPU.

A full explanation and layout of the DIT is given in Section 6

3.1.4 Intelligent Modules

Intelligent Modules are more than just I/O modules. Intelligent modules can be considered as an extension of the CPU with their own "co-processors".

Every Intelligent module also has its own Data Interchange Table (DIT). The size of this DIT depends upon the type of intelligent module, but the lowest 2000 registers of this

DIT are mapped into the P3 CPU's Data Interchange Table address space, and so may be directly addressed as if these registers are part of the CPU.

Any programming or configuration for an Intelligent Module is stored in non-volatile memory in the module itself, and not in the CPU. If an Intelligent Module is exchanged, then the replacement module must be re-configured/reprogrammed. This can be done through the CPU's programming port.

An example of an Intelligent Module is any one of the range of Network Interface Modules (NIM's). NIM's can be considered as an extension of the CPU's network ports. Most NIM's are user programmable, so that custom protocols or data manipulation algorithms may be written for the NIM and executed in the NIM to remove processing overhead from the P3 CPU.

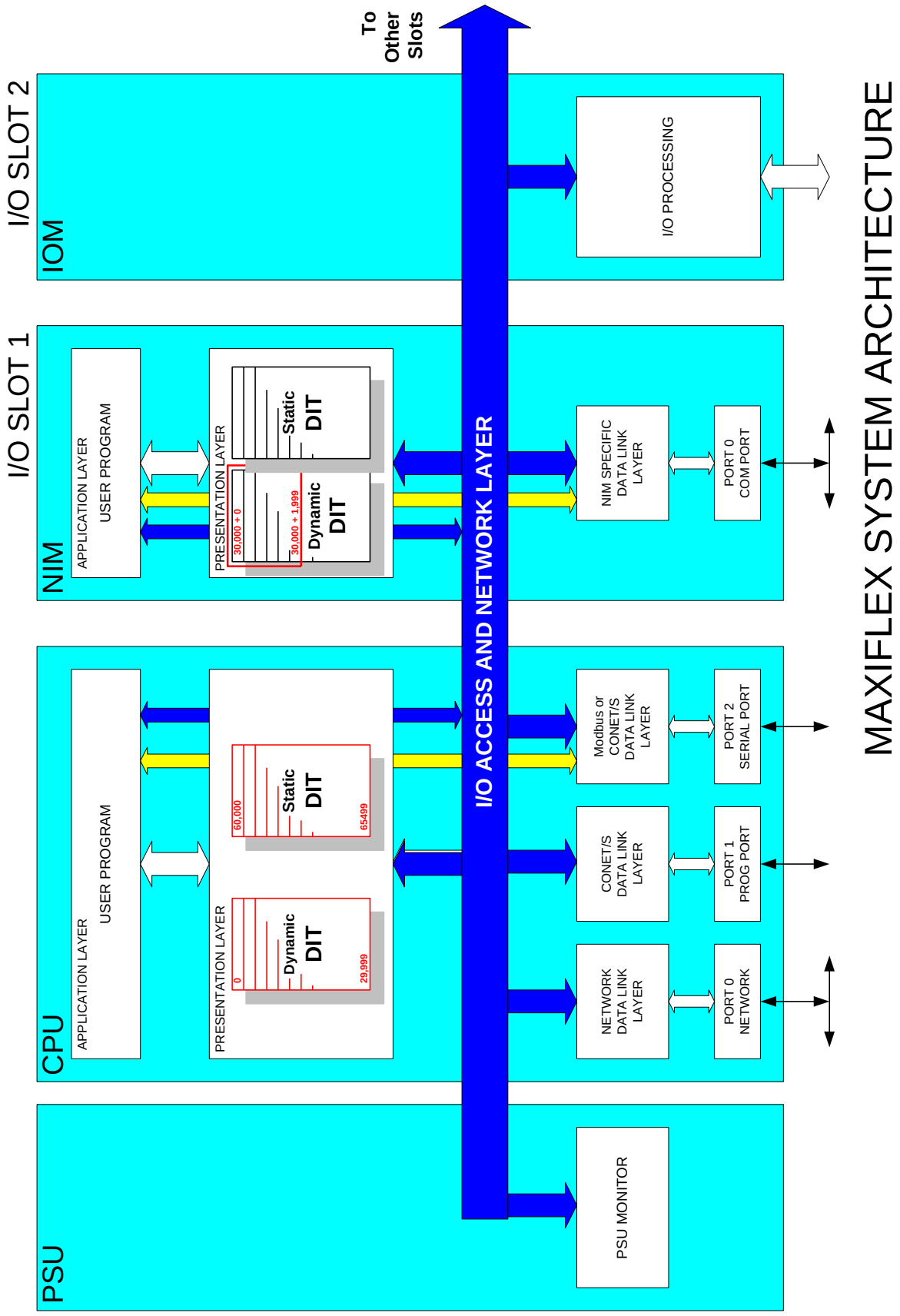


Figure 3.1 P3 CPU System Architecture

4. Installing the CPU

4.1 Connecting the Internal Battery

Before installing the CPU, the internal battery must be connected.

The CPU is shipped from the factory with a “Battery Disconnect Tab” protruding through the top of the unit. Pull on the tab to remove before use.

4.2 Connecting the memory battery backup

The battery in the CPU is used for two purposes:

The battery is always used to retain the real-time clock while the power is off.

The battery can also be optionally used to retain user data in the Data Interchange Table (DIT) when the power is off.

This is not normally required, and if you do not specifically have a need for this feature, then it should not be enabled.

Enable the DIT battery backup feature if required by connecting the battery link, accessible through the front door of the CPU module. The battery link is shipped connected. To activate the battery, pull out the protective tab which says “PULL TO CONNECT BATTERY”.

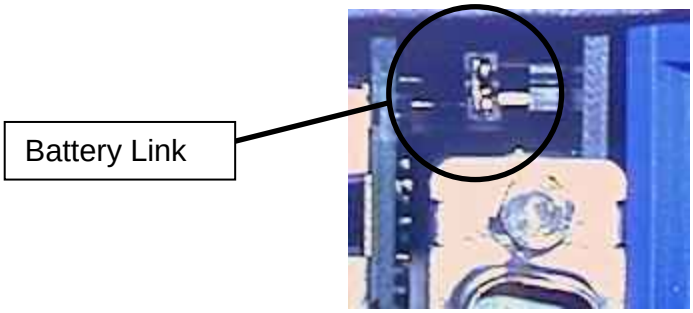


Figure 4.1 P3 Battery Link shown connected (as shipped)

Upon power-up, the CPU checks for valid data in the DIT. If there is no valid data, then the DIT is cleared to zero.

CAUTION – Battery Life:

The Battery used in the P3 CPU is a non-rechargeable Lithium Battery (see section 15 for replacement details).

With power on, there is no drain from the battery, and the life of the battery will be close to its expected shelf life.

With power removed from the CPU, the battery life will be significantly reduced if the DIT battery backup link is installed, due to the extra consumption of the DIT memory.

Therefore only connect the link if this feature is required.

4.3 Installing the CPU on the Maxiflex base

Install the CPU into the CPU position ONLY of any Maxiflex Master Base. CPUs must only be installed and removed from the base with the power off.

Refer to the Maxiflex bases General Instructions for more detail on base layout, module insertion and module removal. Please refer to Figure 2.2 for the CPU position.

See Section 5 for recommended system configuration in dual redundant applications

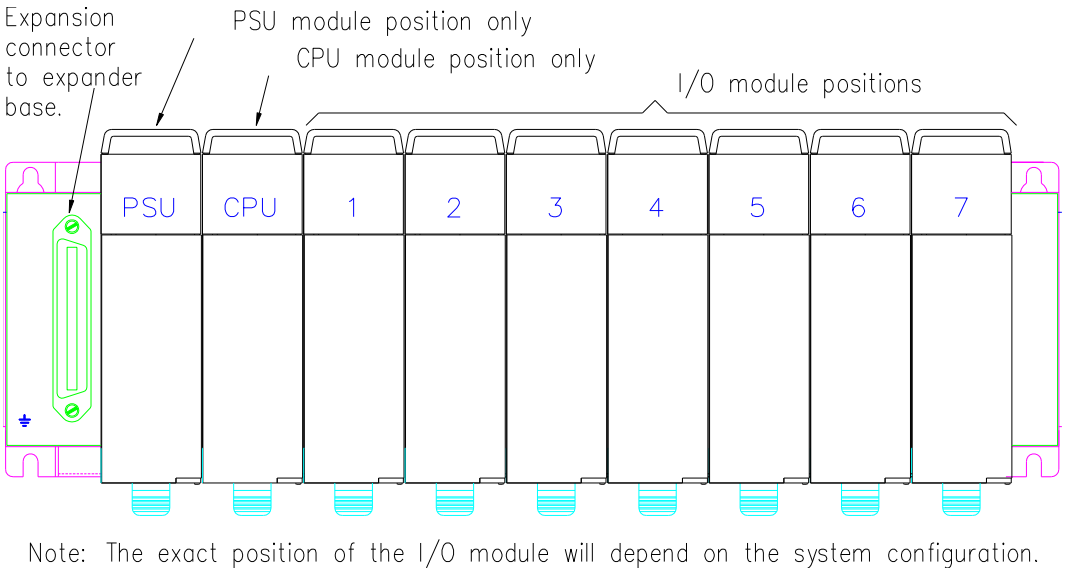


Figure 4.2: Layout of the 7I/O Master Base

4.4 Connecting the Programming port (all models)

Connect the P3/A3 CPU to a standard PC compatible serial port using the OMNIFLEX supplied standard programming cable (Part Number M1831A supplied separately).

If another cable is to be used, the following connections will apply:

Signal Name	DB-9	FCC-68 Pin No.
Rx Data from P3	2	4
Tx Data to P3	3	1
Ground Reference	5	2
All other pins are reserved and must not be connected.		

Table 4.1: Pin allocation of DB-9 connector on program cable and FCC-68 socket on P3 CPU's

4.5 Connecting the Serial port (all models)

The selection of either RS232 or RS485 is accomplished by specific wiring of the serial port connector. No internal links need be changed to select between RS232 and RS422/485.

The serial port of the CPU is not isolated.

Pin number	Communication Standard	
	RS232	RS485
1	Do not connect	Rx Data + (In)
2	Rx Data (In)	Rx Data – (In)
3	Tx Data (Out)	Do not connect
4	Do not connect	Tx Data+ (Out)
5	Ground	Ground
6	Do not connect	Vcc
7	RTS (Out)	Do not connect
8	CTS (In)	Do not connect
9	Do not connect	Tx Data – (Out)

Table 4.2: Pin allocation of serial port connector on P3 CPU's.

NOTE: The RTS and CTS handshaking lines are available for applications that require it. In most applications connecting handshaking lines is not a requirement.

4.6 Connecting the Conet/c network port

The CPU should be connected to the Conet network using the C6169 Conet Termination Board and interconnecting cable. (Refer to C6169 Installation Guide for further details).

Pin number	Description
2	Signal +
5	Cable screen (S)
8	Signal -
1, 3, 4, 6, 7 and 9	No connection

Table 4.3: Pin allocation of Conet port connector on the M1261E CPU

4.7 Connecting the Ethernet network port

This CPU provides a standard UTP interface utilising a RJ45 connector suitable for direct connection to a 10/100 Mbit/s hub/switch in an Ethernet system.

Consult your network administrator/consultant for further installation information for the Ethernet network.

4.8 Connecting the Watchdog Relay Output Contact

The watchdog relay output contact terminals are located below the Programming Port inside the front door of the CPU module.

These terminals provide a potential-free contact output that will be closed (watch-dog relay energised) while power is on and the CPU is operating normally.

4.9 Applying power for the first time

Make sure that the P3 CPU has been installed into the CPU slot and the Power Supply in the PSU slot of the Maxiflex Master Base.

Apply the power to the PSU. All the front-panel indicators on the CPU will light up for a few seconds while the CPU initialises. Thereafter normal operation of the front-panel indicators will resume.

4.10 Front-Panel LED Indicators Explained

NOTE: Not all P3/A3 CPU's have all of these LED Indicators. See section 2.1 for a cross-reference table showing which LED's are supported on your selected P3 CPU.

Legend	Colour	Description
CPU OK	GREEN	**ON** - CPU is healthy **OFF** or **FLASHING** – No power applied or CPU Faulty. Check status of PSU Module, or exchange CPU
I/O OK	GREEN	**ON** - I/O Module status healthy and matches the configured I/O Module List and User Program module configuration, if IEC61131 program is running. **SLOW FLASH** – I/O Module List does not match the installed hardware; or an IEC61131 program is running with I/O Connections that do not match the installed hardware. Solution: Configure the I/O Module List. **FAST FLASH** – IEC61131 program is running with some I/O Forced. Solution: Remove forcing from ISaGraf Workbench.
RUN	GREEN	**ON** – User program is running. **OFF** – No User Program is running **FLASHING** – Indicates the CPU is in redundant configuration and is currently in Standby mode.
BATT	RED	**OFF** – Internal battery is healthy. **ON** – Internal battery is LOW. Solution: Check Battery Disconnect Tab has been removed (see section 4.1), or replace Battery (see section 15.2).
SERIAL Rx	YELLOW	**ON** – data is being received on serial port **OFF** – serial port receiver is idle
SERIAL Tx	RED	**ON** – serial data is being transmitted on serial port **OFF** – serial port transmitter is idle
NETWORK Rx	YELLOW	**ON** – Conet Network data is being received **OFF** – Conet Network receiver is idle
NETWORK Tx	RED	**ON** – Conet Network data is being transmitted **OFF** – Conet Network transmitter is idle
NETWORK TOKEN	GREEN	**ON** – Conet Network not connected or setup incorrectly **OFF** – Conet Network not connected or setup incorrectly **FLASHING EVENLY** – Conet Network is connected (speed of flash indicates token rate) **FLASHING UNEVENLY** – Conet Network is connected but a cable problem is preventing the token from being passed reliably. Check Network Addresses and cable installation and terminations
NET LINK	GREEN	**ON** – Ethernet UTP network link is good. **OFF** – Ethernet Network not connected or setup incorrectly **FLASH OFF** – There is activity on the network segment
NET 100	YELLOW	**ON** – The network is operating at 100Mbit/s. **OFF** – The network is operating at 10Mbit/s.

SYNC LINK	GREEN	**ON** – Sync Link to other CPU is good. **OFF** – Sync Link disconnected or other CPU down. **FLASH OFF** – There is activity on the Sync Link.
SYNC 100	YELLOW	**ON** – The Sync Link is operating at 100Mbit/s. **OFF** – There is a problem with the Sync Link.

Table 4.4 Front Panel LED Indicator Functions Explained

NOTE: On Power up, all LED indicators will light for a few seconds during CPU initialisation. Thereafter, the indicators will resume their normal operation as per the table above.

5. I/O Module Management & Scanning Explained

5.1 Number of I/O Modules

5.1.1 Local I/O Modules

The CPUs can access up to 15 local I/O modules. These modules are located on the same Master Base (plus Expander Base) as the CPU.

In a dual redundant configuration, these modules on the local Base are not accessible by the other Redundant controller (except by using the Sync Blocks). Use Remote I/O modules for shared I/O in redundant controller applications

5.1.2 Remote I/O Modules

The CPU can access up to 7 remote I/O Racks.

A Remote I/O Rack consists of a conventional Master (with Expander Base if required) fitted with a M1248A R2c I/O Scanner module in the CPU position.

To access remote I/O systems from the CPU, install an M1593A Remote I/O Network Interface module (RIO NIM) into Slot 1 of the Master Base in which the CPU is located. This RIO NIM is connected to the R2c I/O Scanners via a dedicated Remote I/O network.

The Remote I/O network is based on the Conet/c rugged industrial LAN allowing remote I/O Racks to be installed up to 10km from the Main Process Automation Controller.

Remote I/O Racks are accessible from both controllers in a dual redundant configuration.

5.2 Automatic I/O Module Identification

The CPU automatically identifies each I/O module installed in a Maxiflex system – both in the local I/O Base and in the remote I/O Bases.

This information is used for two purposes:

1. Checking for the presence of the correct I/O modules in a system.
2. Automatic scanning of these I/O modules.

5.3 The I/O Module List

You can configure a P3 CPU with the required list of I/O modules for the system. This list is called the “I/O Module List”.

The CPU continuously compares the present I/O modules against this I/O Module List, and reports any discrepancies. This monitoring is independent of any User Program downloaded to the P3 CPU.

User programs must also be configured with the I/O modules used by the User Program. It is not necessary to configure ALL of the I/O Modules in your IEC61131 User Program. You need only configure the modules used in the program. This configuration is done in the Omniflex ISaGraf Programmer’s Workbench before downloading the User Program to the P3 PAC CPU. This module status is also continuously monitored while the program is running.

All of this module status information is available in the DIT for monitoring by a remote device, and by indication on the I/O OK LED on the front panel of the CPU (see section 11.7.1).

5.4 Automatic I/O Scanning

When an I/O module is installed into the system, it will be detected by the CPU in seconds and then automatically scanned according to its module type. No special configuration is required for this to occur.

If a User Program is running on the P3 CPU, then the I/O module scanning is synchronised with the user program scan. If no User Program is running on the P3 CPU, then the I/O is scanned every 10ms.

Input status information is automatically read from the Input modules and stored in the CPU's Dynamic Data area of the DIT Table, and output status information is automatically read from Dynamic Data area of the DIT Table and written to the relevant output modules.

The layout of the input and output data in the CPU's DIT is designed to minimise the number of polls required from a supervisory system to read/write data to the CPU, by grouping together like types of data.

It is not necessary to have a User Program installed in the P3 CPU for the I/O data to be accessible by a SCADA/DCS system connected to one of the network ports. User Programs are only required if you wish to perform some local control or internal data manipulation.

6. The Data Interchange Table explained

6.1 DIT Table Overview

Data Interchange Tables (DIT's) are the "crossroads" for data in the Maxiflex system. A DIT is an array of 16 bit registers accessible from any function or communications port in the system.

The P3 CPU has a DIT in which all the configuration settings and dynamic data, including I/O can be accessed. Any exchange of data between functions in the CPU and with the outside world takes place through the CPU's DIT.

Intelligent I/O Modules such as NIM's also each have a DIT. A portion of the intelligent I/O modules' DIT Registers are overlaid on to the CPU DIT, and appear as if they are part of the CPU's DIT. This extended DIT addressing is used to directly access data in any intelligent modules installed on the Maxiflex base as if the data is in the CPU.

6.2 DIT Table Layout

The Data Interchange Table in the P3 CPU provides access to up to 65,500 16-bit data registers used for reading and writing all configuration and dynamic data in the CPU and all of its I/O modules.

Some of these register addresses expose registers in the Intelligent I/O modules of the system as if they are part of the CPU's DIT. This is known as "extended DIT access". There are two modes of operation of extended DIT access:

In Extended Address Mode 1, each of the 15 I/O Module Slots is allocated 2000 Registers in the CPU address space.

In Extended Address Mode 2, only the first 7 I/O Module Slots are allocated 4000 Registers each in the CPU address space.

The following tables show the address map of the DIT table for an entire MAXIFLEX system as viewed from the P3 CPU in each Extended Addressing mode.

This table shows the CPU DIT mapping of the first 2000 registers of any NIM modules installed on the Maxiflex base:

Maxiflex Local Master Rack									
Maxiflex Slot:	**CPU Dynamic Data Space**	**CPU Config Data Space**	**I/O Slot 1**	**I/O Slot 2**	**I/O Slot 3**	**I/O Slot 4**	**I/O Slot 5**	**I/O Slot 6**	**I/O Slot 7**
DIT Start Address:	0	60,000	30,000	32,000	34,000	36,000	38,000	40,000	42,000
DIT End Address:	29,999	65499	31,999	33,999	35,999	37,999	39,999	41,999	43,999
No. of Registers	30,000	5,499	2,000	2,000	2,000	2,000	2,000	2,000	2,000

Table 6.1: DIT Address Map of the P3 CPU and Master Rack in Extended DIT Address Mode 1

Maxiflex Local Expansion Rack								
Maxiflex Slot:	**I/O Slot 8**	**I/O Slot 9**	**I/O Slot 10**	**I/O Slot 11**	**I/O Slot 12**	**I/O Slot 13**	**I/O Slot 14**	**I/O Slot 15**
DIT Start Address:	44,000	46,000	48,000	50,000	52,000	54,000	56000	58,000
DIT End Address:	45,999	47,999	49,999	51,999	53,999	55,999	57,999	59,999
No. of Registers	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000

Table 6.2: DIT Address Map of the Expander Rack in Extended DIT Address Mode 1

In some applications, when Remote I/O is not used it can be advantage to map 4000 DIT registers of a NIM module's DIT to the CPU. In this mode, 4000 DIT registers per I/O Slot are made visible for I/O Slots 1 to 7 only. Hence the table would change as follows:

Maxiflex Local Master Rack									
Maxiflex Slot:	**CPU Dynamic Data Space**	**CPU Config Data Space**	**I/O Slot 1**	**I/O Slot 2**	**I/O Slot 3**	**I/O Slot 4**	**I/O Slot 5**	**I/O Slot 6**	**I/O Slot 7**
DIT Start Address:	0	60,000	30,000	34,000	38,000	42,000	46,000	50,000	54,000
DIT End Address:	29,999	65499	33,999	37,999	41,999	45,999	49,999	53,999	57,999
No. of Registers	30,000	5,499	4,000	4,000	4,000	4,000	4,000	4,000	4,000

Table 6.3: DIT Address Map of the CPU and Master Rack in Extended DIT Address Mode 2

The default setting for the extended DIT map is Mode 1: 2000 DIT registers per I/O slot.

To change this setting using Omniset, use the "Extended DIT Layout" Group under "Configuration/Advanced Settings".

6.3 CPU Dynamic Data Space (0-29,999)

This area of the DIT is typically used to store dynamic, real time data, and provides the fastest access to data from any of the network ports or the user program.

6.4 Intelligent Modules in I/O Slots 1 to 15 (30,000-59,999)

These DIT register ranges provide direct access to the first 2,000 or 4000 DIT registers in any intelligent Modules (such as Network Interface Modules) installed in the I/O slots of the MAXIFLEX rack (as set by the Extended DIT Address Mode – see section 6.2 above.

The separate User Manual for each intelligent Module defines the DIT registers allocated on each module.

These DIT registers for given in each intelligent module User Manual will be shown numbered from 0. To find the CPU's DIT Address for of the first 2,000 (or 4000) DIT registers in an intelligent module as reflected in the CPU DIT, simply add the DIT register number

given in the intelligent module User Manual to the DIT Start Address applicable to the I/O Slot (from the table above).

Alternatively you can calculate the start address of an Intelligent I/O Module DIT as follows:

For Mode 1 (2000 DIT registers per I/O Slot) the calculation is:

Start Address = 30000 + (SlotNumber – 1) x 2000

where SlotNumber can range from 1 to 15

For Mode 2 (4000 DIT registers per I/O Slot) the calculation is:

Start Address = 30000 + (SlotNumber – 1) x 4000

where SlotNumber can range from 1 to 7

CAUTION: Attempts to read data from unallocated areas of the DIT space will give indeterminate results.

CAUTION: The access times from these areas of the DIT are slower than from the CPU Dynamic Data Area (DIT addresses 0 – 30,000). For fastest access from the SCADA/DCS of data in the NIM's, use the CPU User Program to copy the data from the NIM's address space into DIt registers in the address range 0-29,999

Example 1:

A P3 CPU is configured in Extended Addressing mode 1 (2000 DIT registers per I/O Slot).

A M1580A Dual Serial NIM module is present in I/O Slot 2.

The first 2,000 registers of this module will be mapped into the CPU's DIT address space starting at DIT address 32,000.

To read the Alive Counter of the M1585B NIM module in slot 2, which is DIT register 23 in the DIT layout of the NIM module, read CPU DIT address 32,023.

The M1580A Serial NIM has a total of 4000 DIT registers. Only the first 2000 are mapped into the CPU DI space. The balance of the M1580A's DIT space can be accessed through any of the communication ports of the P3 CPU by addressing this I/O module directly as an "Intelligent Module in Slot". See the M1580A User Manual for information on this method of addressing.

6.5 CPU Configuration (Static) Data Space (60,000-65,499)

This DIT area is used to store configuration data for the functions supported in the P3 CPU, such as Serial Port set up etc., and any I/O modules that require configuration (e.g. the M1432 8 way thermocouple module).

Intelligent Modules with their own DIT's, such as the M1585B Modbus Network Interface Module described in 6.4 above are NOT configured in this space. These intelligent modules store their own configuration.

This CPU Configuration/Static Data Space is stored in non-volatile memory in the CPU, (and does not utilise the internal battery). This means that this data will still be valid even when the on-board battery is LOW or is being changed.

In addition to the allocated configuration data in this Static Data space, there is an area available for the user to store application specific configuration that will remain fixed for the life of the application.

Continuous writing of dynamic data to this area during normal system operation should be avoided.

The CPU takes up to 15 seconds to store this data in the non-volatile memory on the CPU, so power should not be removed for at least 15 seconds after Configuration changes have been made.

See the DIT layout in section 15 for details of layout of this section of the DIT.

6.6 DIT Access to I/O Module Data

Space in the P3 CPU's Dynamic Data Area of the DIT is reserved for I/O module scanning, and in the P3 CPU's Static Data Area of the DIT for I/O Module configuration. All access to I/O data in the Maxiflex P3 is done by reading or writing to the relevant DIT Register(s).

The following DIT space is allocated in the CPU to each I/O Slot in the Maxiflex System:

I/O Type	No. of Registers per Slot
Analogue Outputs	16
Analogue Inputs	16
Digital Outputs	4
Digital Inputs	4
Configuration	100

Table 6.4 I/O Slot DIT data space allocation

The Inputs and Outputs from all the modules are grouped together in the DIT according to their I/O Type. Each group is called a "Data Space". This arrangement allows easy reading or writing of multiple modules at the same time, by reading from or writing to relevant I/O Type Data Space in the CPU's Data Interchange Table (DIT). This can be accomplished from any of the CPU's network ports.

The table below identifies the start location and size, in the CPU's DIT, of each of these data spaces, for every I/O slot. If the I/O module in that slot does not utilise a type of data, then that data space in the DIT is not used.

Notes:

1. The specific register contents for each type of I/O Module in each Data Space can be found in Section 7.
2. The Module Configuration Data for I/O modules in Remote Racks is stored in the R2c I/O Scanner on the remote rack. This information is still configured through the main P3 PAC CPU.

	Local Master Rack								Local Expansion Rack							
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs	500	516	532	548	564	580	596	612	628	644	660	676	692	708	724	740
No of Registers	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs	2500	2516	2532	2548	2564	2580	2596	2612	2628	2644	2660	2676	2692	2708	2724	2740
No of Registers	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs	4500	4504	4508	4512	4516	4520	4524	4528	4532	4536	4540	4544	4548	4552	4556	4560
No of Registers	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs	5000	5004	5008	5012	5016	5020	5024	5028	5032	5036	5040	5044	5048	5052	5056	5060
No of Registers	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Config.	61000	61100	61200	61300	61400	61500	61600	61700	61800	61900	62000	62100	62200	62300	62400	62500
No of Registers	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.5- I/O DIT Address Map for P3 & A3 CPUs Local I/O Rack

	Remote Master Rack 1							Remote Expansion Rack 1							
I/O Slot:	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs	756	772	788	804	820	836	852	868	884	900	916	932	948	964	980
No of Registers	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs	2756	2772	2788	2804	2820	2836	2852	2868	2884	2900	2916	2932	2948	2964	2980
No of Registers	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs	4564	4568	4572	4576	4580	4584	4588	4592	4596	4600	4604	4608	4612	4616	4620
No of Registers	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs	5064	5068	5072	5076	5080	5084	5088	5092	5096	5100	5104	5108	5112	5116	5120
No of Registers	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3															
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.6- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 1

		Remote Master Rack 2							Remote Expansion Rack 2							
I/O Slot:		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs		996	1012	1028	1044	1060	1076	1092	1108	1124	1140	1156	1172	1188	1204	1220
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs		2996	3012	3028	3044	3060	3076	3092	3108	3124	3140	3156	3172	3188	3204	3220
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs		4624	4628	4632	4636	4640	4644	4648	4652	4656	4660	4664	4668	4672	4676	4680
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs		5124	5128	5132	5136	5140	5144	5148	5152	5156	5160	5164	5168	5172	5176	5180
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3																
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.7- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 2

		Remote Master Rack 3							Remote Expansion Rack 3							
I/O Slot:		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs		1236	1252	1268	1284	1300	1316	1332	1348	1364	1380	1396	1412	1428	1444	1460
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs		3236	3252	3268	3284	3300	3316	3332	3348	3364	3380	3396	3412	3428	3444	3460
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs		4684	4688	4692	4696	4700	4704	4708	4712	4716	4720	4724	4728	4732	4736	4740
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs		5184	5188	5192	5196	5200	5204	5208	5212	5216	5220	5224	5228	5232	5236	5240
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3																
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.8- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 3

		Remote Master Rack 4							Remote Expansion Rack 4							
I/O Slot:		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs		1476	1492	1508	1524	1540	1556	1572	1588	1604	1620	1636	1652	1668	1684	1700
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs		3476	3492	3508	3524	3540	3556	3572	3588	3604	3620	3636	3652	3668	3684	3700
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs		4744	4748	4752	4756	4760	4764	4768	4772	4776	4780	4784	4788	4792	4796	4800
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs		5244	5248	5252	5256	5260	5264	5268	5272	5276	5280	5284	5288	5292	5296	5300
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3																
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.9- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 4

		Remote Master Rack 5							Remote Expansion Rack 5							
I/O Slot:		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs		1716	1732	1748	1764	1780	1796	1812	1828	1844	1860	1876	1892	1908	1924	1940
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs		3716	3732	3748	3764	3780	3796	3812	3828	3844	3860	3876	3892	3908	3924	3940
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs		4804	4808	4812	4816	4820	4824	4828	4832	4836	4840	4844	4848	4852	4856	4860
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs		5304	5308	5312	5316	5320	5324	5328	5332	5336	5340	5344	5348	5352	5356	5360
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3																
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.10- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 5

		Remote Master Rack 6							Remote Expansion Rack 6							
I/O Slot:		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs		1956	1972	1988	2004	2020	2036	2052	2068	2084	2100	2116	2132	2148	2164	2180
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs		3956	3972	3988	4004	4020	4036	4052	4068	4084	4100	4116	4132	4148	4164	4180
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs		4864	4868	4872	4876	4880	4884	4888	4892	4896	4900	4904	4908	4912	4916	4920
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs		5364	5368	5372	5376	5380	5384	5388	5392	5396	5400	5404	5408	5412	5416	5420
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3																
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.11- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 6

		Remote Master Rack 7							Remote Expansion Rack 7							
I/O Slot:		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Analogue Outputs		2196	2212	2228	2244	2260	2276	2292	2308	2324	2340	2356	2372	2388	2404	2420
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Analogue Inputs		4196	4212	4228	4244	4260	4276	4292	4308	4324	4340	4356	4372	4388	4404	4420
No of Registers		16	16	16	16	16	16	16	16	16	16	16	16	16	16	16
Digital Outputs		4924	4928	4932	4936	4940	4944	4948	4952	4956	4960	4964	4968	4972	4976	4980
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Digital Inputs		5424	5428	5432	5436	5440	5444	5448	5452	5456	5460	5464	5468	5472	5476	5480
No of Registers		4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Module Configuration for Remote Racks is stored in the I/O Scanner on the Remote Rack and not in the P3																
I/O Slot:	**PSU**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Module Config.	64048	4000	8000	12000	16000	20000	24000	28000	32000	36000	40000	44000	48000	52000	56000	60000
No of Registers	2	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Table 6.12- I/O DIT Address Map for P3 & A3 CPUs Remote I/O Rack 7

7. I/O Module DIT Register Reference

This section describes the I/O Data Spaces allocated in the Data Interchange Table for each I/O Module that can be used with the P3 range of CPU's.

This DIT Register reference can be used when configuring supervisory systems to access I/O from the P3 CPU, or for configuring I/O modules programmatically.

The DIT register addressing given in this section is always based from 0. This is an address "offset" from the start of the relevant Data Space (see Section 6.6 for more information on I/O Data Spaces). The actual DIT address depends upon the I/O Slot in which the module is installed. To find the actual address, add the DIT offset defined in this section to the DIT Start Address for the relevant mode as defined in section 6.2.

7.1 M1101A –24VDC/12V Solar 3A Charger/Power Supply

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs where individual inputs are allocated as follows: Bit 0 (LSB) = First Scan Bit 1 = Primary Supply Status (1= Program in First Scan) Bit 2 = Temperature Sensor Status (1 = Sensor is connected) Bit 3 = Battery Level Warning (1 = Battery Level Low Warning Alarm) Bit 4 = Battery Level Critical (1 = Battery Level Low Critical Alarm) Bits 5 to 16 will read as 0	0
ANALOGUE INPUT DATA SPACE	DIT (Offset)
Battery Voltage (in Volts x 0.1)	0
Digital and Analogue Output Data Spaces are unused by this module	
CONFIGURATION DATA SPACE	DIT (Offset)
Critical Battery Level (in Volts x 0.1)	0
Alarm Battery Level (in Volts x 0.1)	1
Battery Test Interval (in minutes)	2

7.2 M1102B – DC L - 24Vdc Logic Power Supply

No Data Spaces are used by this module

This module has no data readable in the DIT.

When this module is allocated in an IEC61131 program, then a single digital input is available to the program that will energise on the first scan of the program.

7.3 M1104A – DC LF - 48Vdc Logic/Field Power Supply

No Data Spaces are used by this module

This module has no data readable in the DIT.

When this module is allocated in an IEC61131 program, then a single digital input is available to the program that will energise on the first scan of the program.

7.4 M1151C – AC LF - 115/230Vac Logic/Field Power Supply

No Data Spaces are used by this module

This module has no data readable in the DIT.

When this module is allocated in an IEC61131 program, then a single digital input is available to the program that will energise on the first scan of the program.

7.5 M1152B – AC LC - 115/230Vac Logic Power Supply/Charger.

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs where individual inputs are allocated as follows: Bit 0 (LSB) = First Scan Bit 1 = Primary Supply Status (1= Program in First Scan) Bit 2 = Temperature Sensor Status (1 = Sensor is connected) Bit 3 = Battery Level Warning (1 = Battery Level Low Warning Alarm) Bit 4 = Battery Level Critical (1 = Battery Level Low Critical Alarm) Bits 5 to 16 will read as 0	0
ANALOGUE INPUT DATA SPACE	DIT (Offset)
Battery Voltage (in Volts x 0.1)	0
Digital and Analogue Output Data Spaces are unused by this module	
CONFIGURATION DATA SPACE	DIT (Offset)
Critical Battery Level (in Volts x 0.1)	0
Alarm Battery Level (in Volts x 0.1)	1
Battery Test Interval (in minutes)	2

7.6 M1321A – 8DI-C - 8 way Individually Isolated Contact Input Module

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs 1-8 where individual inputs are allocated as follows: Input 1 = Bit 0 (LSB) Input 2 = Bit 1 Input 3 = Bit 2 Input 4 = Bit 3 Input 5 = Bit 4 Input 6 = Bit 5 Input 7 = Bit 6 Input 8 = Bit 7 Bits 9 to 16 will read as 0	0
All other Data Spaces are unused by this module	

Example:
Input 6 of an M1321A module installed in Local Slot 3 can be read from DIT Address 5012, Bit 5. Refer to section 6.6.

7.7 M1322A - 16DI - 16 way Digital Input Module (9-30Vdc)
M1323A - 16DI - 16 way Digital Input Module (18-60Vdc)

DIGITAL INPUT DATA SPACE		DIT (Offset)
Digital Inputs 1-16 where individual inputs are allocated as follows:		0
Input 1 = Bit 0 (LSB) Input 2 = Bit 1 Input 3 = Bit 2 Input 4 = Bit 3 Input 5 = Bit 4 Input 6 = Bit 5 Input 7 = Bit 6 Input 8 = Bit 7	Input 9 = Bit 8 Input 10 = Bit 9 Input 11 = Bit 10 Input 12 = Bit 11 Input 13 = Bit 12 Input 14 = Bit 13 Input 15 = Bit 14 Input 16 = Bit 15 (MSB)	
All other Data Spaces are unused by this module		

Example:
Input 3 of an M1322A module installed in Local Slot 6 can be read from DIT Address 5024, Bit 2

7.8 M1326A – 32DI – 32 way Digital Input Module

DIGITAL INPUT DATA SPACE		DIT (Offset)
Digital Inputs 1-16 where individual inputs are allocated as follows:		0
Input 1 = Bit 0 (LSB) Input 2 = Bit 1 Input 3 = Bit 2 Input 4 = Bit 3 Input 5 = Bit 4 Input 6 = Bit 5 Input 7 = Bit 6 Input 8 = Bit 7	Input 9 = Bit 8 Input 10 = Bit 9 Input 11 = Bit 10 Input 12 = Bit 11 Input 13 = Bit 12 Input 14 = Bit 13 Input 15 = Bit 14 Input 16 = Bit 15 (MSB)	
Digital Inputs 17-32 where individual inputs are allocated as follows:		1
Input 17 = Bit 0 (LSB) Input 18 = Bit 1 Input 19 = Bit 2 Input 20 = Bit 3 Input 21 = Bit 4 Input 22 = Bit 5 Input 23 = Bit 6 Input 24 = Bit 7	Input 25 = Bit 8 Input 26 = Bit 9 Input 27 = Bit 10 Input 28 = Bit 11 Input 29 = Bit 12 Input 30 = Bit 13 Input 31 = Bit 14 Input 32 = Bit 15 (MSB)	
All other Data Spaces are unused by this module		

Example:
Input 28 of an M1326A module installed in Local Slot 12 can be read from DIT Address 5049, Bit 11.

7.9 M1330A – 8DI8RO 8way Digital Input /8 way Relay Output Module

DIGITAL INPUT DATA SPACE		DIT (Offset)
Digital Inputs 1-8 where individual inputs are allocated as follows: Input 1 = Bit 0 (LSB) Input 2 = Bit 1 Input 3 = Bit 2 Input 4 = Bit 3 Input 5 = Bit 4 Input 6 = Bit 5 Input 7 = Bit 6 Input 8 = Bit 7	Bits 9 to 16 will read as 0	0
DIGITAL OUTPUT DATA SPACE		DIT (Offset)
Digital Outputs 1-8 where individual inputs are allocated as follows: Output 1 = Bit 0 (LSB) Output 2 = Bit 1 Output 3 = Bit 2 Output 4 = Bit 3 Output 5 = Bit 4 Output 6 = Bit 5 Output 7 = Bit 6 Output 8 = Bit 7	Bits 9 to 16 are unused.	1
All other Data Spaces are unused by this module		

Example:
Input 5 of an M1330A module installed in Local Slot 3 can be read from DIT Address 5012, Bit 4

Output 5 of an M1330A module installed in Local Slot 3 can be energised by writing a '1' to Bit 4 of DIT Register 4513

7.10 M1341B – 16DO – 16way Digital Output Module

DIGITAL OUTPUT DATA SPACE		DIT (Offset)
Digital Outputs 1-16 where individual outputs are allocated as follows: Output 1 = Bit 0 (LSB) Output 2 = Bit 1 Output 3 = Bit 2 Output 4 = Bit 3 Output 5 = Bit 4 Output 6 = Bit 5 Output 7 = Bit 6 Output 8 = Bit 7	Output 9 = Bit 8 Output 10 = Bit 9 Output 11 = Bit 10 Output 12 = Bit 11 Output 13 = Bit 12 Output 14 = Bit 13 Output 15 = Bit 14 Output 16 = Bit 15 (MSB)	0
All other Data Spaces are unused by this module		

Example:
Output 3 of an M1341B module installed in Slot 4 can be energised by writing to DIT Address 4516, Bit 2

7.11 M1342A – 32DO – 32way Digital Output Module

DIGITAL OUTPUT DATA SPACE		DIT (Offset)
Digital Outputs 1-16 where individual outputs are allocated as follows:		0
Output 1 = Bit 0 (LSB) Output 2 = Bit 1 Output 3 = Bit 2 Output 4 = Bit 3 Output 5 = Bit 4 Output 6 = Bit 5 Output 7 = Bit 6 Output 8 = Bit 7	Output 9 = Bit 8 Output 10 = Bit 9 Output 11 = Bit 10 Output 12 = Bit 11 Output 13 = Bit 12 Output 14 = Bit 13 Output 15 = Bit 14 Output 16 = Bit 15 (MSB)	
Digital Outputs 17-32 where individual outputs are allocated as follows:		1
Output 17 = Bit 0 (LSB) Output 18 = Bit 1 Output 19 = Bit 2 Output 20 = Bit 3 Output 21 = Bit 4 Output 22 = Bit 5 Output 23 = Bit 6 Output 24 = Bit 7	Output 25 = Bit 8 Output 26 = Bit 9 Output 27 = Bit 10 Output 28 = Bit 11 Output 29 = Bit 12 Output 30 = Bit 13 Output 31 = Bit 14 Output 32 = Bit 15 (MSB)	
All other Data Spaces are unused by this module		

Example:
Output 23 of an M1342A module installed in Local Slot 4 can be energised by writing to DIT Address 4517, Bit 6

7.12 M1372A – 8RO – 8way Relay Output Module

DIGITAL OUTPUT DATA SPACE		DIT (Offset)
Digital Outputs 1-8 where individual outputs are allocated as follows:		0
Output 1 = Bit 0 (LSB) Output 2 = Bit 1 Output 3 = Bit 2 Output 4 = Bit 3 Output 5 = Bit 4 Output 6 = Bit 5 Output 7 = Bit 6 Output 8 = Bit 7	Other bits are unused	
All other Data Spaces are unused by this module		

Example:
Output 1 of an M1372A module installed in Local Slot 1 can be energised by writing to DIT Address 4504, Bit 0.

7.13 M1403A – 16AI – 16way Analogue Input Module

ANALOGUE INPUT DATA SPACE	DIT (Offset)
Analogue Input 1	0
Analogue Input 2	1
Analogue Input 3	2
Analogue Input 4	3
Analogue Input 5	4
Analogue Input 6	5
Analogue Input 7	6
Analogue Input 8	7
Analogue Input 9	8
Analogue Input 10	9
Analogue Input 11	10
Analogue Input 12	11
Analogue Input 13	12
Analogue Input 14	13
Analogue Input 15	14
Analogue Input 16	15
Analogue Data is read as an integer in the range 0 to 10000 representing the input range 0 to 100%.	
All other Data Spaces are unused by this module	

Example:
Input 3 of an M1403A module installed in Local Slot 6 can be read from DIT Address 2598.

7.14 M1412A – 8AO – 8 way Analogue Output Module

ANALOGUE OUTPUT DATA SPACE	DIT (Offset)
Analogue Output 1	0
Analogue Output 2	1
Analogue Output 3	2
Analogue Output 4	3
Analogue Output 5	4
Analogue Output 6	5
Analogue Output 7	6
Analogue Output 8	7
Analogue Data must be written as an integer in the range 0 to 10000 representing the output range 0 to 100%.	
All other Data Spaces are unused by this module	

Example:
To set Output 3 of an M1412A module installed in Local Slot 6 to 12mA, write the value 5000 to the DIT register 598 based on a range of 4-20mA.

7.15 M1431B – 8VC ISO – 8 way Isolated Voltage/Current Input Module

ANALOGUE INPUT DATA SPACE	DIT (Offset)
Analogue Input 1	0
Analogue Input 2	1
Analogue Input 3	2
Analogue Input 4	3
Analogue Input 5	4
Analogue Input 6	5
Analogue Input 7	6
Analogue Input 8	7

Analogue Data is read as an integer in the range 0 to 10000 representing the input range 0 to 100%.

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs 1-16 where individual inputs are allocated as follows: Bit 0: Input 1 >= High Limit Bit 1: Input 1 >= High-High Limit Bit 2: Input 1 <= Low Limit Bit 3: Input 1 <= Low-Low Limit Bit 4: Input 1 Calibration Error Bit 5: Input 1 Configuration Corrupted Bit 6: Input 1 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 2 >= High Limit Bit 9: Input 2 >= High-High Limit Bit 10: Input 2 <= Low Limit Bit 11: Input 2 <= Low-Low Limit Bit 12: Input 2 Calibration Error Bit 13: Input 2 Configuration Corrupted Bit 14: Input 2 Incorrect Input Type Bit 15: Unused (will read as 0)	0
Digital Inputs 17-32 where individual inputs are allocated as follows: Bit 0: Input 3 >= High Limit Bit 1: Input 3 >= High-High Limit Bit 2: Input 3 <= Low Limit Bit 3: Input 3 <= Low-Low Limit Bit 4: Input 3 Calibration Error Bit 5: Input 3 Configuration Corrupted Bit 6: Input 3 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 4 >= High Limit Bit 9: Input 4 >= High-High Limit Bit 10: Input 4 <= Low Limit Bit 11: Input 4 <= Low-Low Limit Bit 12: Input 4 Calibration Error Bit 13: Input 4 Configuration Corrupted Bit 14: Input 4 Incorrect Input Type Bit 15: Unused (will read as 0)	1
Digital Inputs 33-48 where individual inputs are allocated as follows: Bit 0: Input 5 >= High Limit Bit 1: Input 5 >= High-High Limit Bit 2: Input 5 <= Low Limit Bit 3: Input 5 <= Low-Low Limit Bit 4: Input 5 Calibration Error Bit 5: Input 5 Configuration Corrupted Bit 6: Input 5 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 6 >= High Limit Bit 9: Input 6 >= High-High Limit Bit 10: Input 6 <= Low Limit Bit 11: Input 6 <= Low-Low Limit Bit 12: Input 6 Calibration Error Bit 13: Input 6 Configuration Corrupted Bit 14: Input 6 Incorrect Input Type Bit 15: Unused (will read as 0)	2

Digital Inputs 49-64 where individual inputs are allocated as follows: Bit 0: Input 7 >= High Limit Bit 1: Input 7 >= High-High Limit Bit 2: Input 7 <= Low Limit Bit 3: Input 7 <= Low-Low Limit Bit 4: Input 7 Calibration Error Bit 5: Input 7 Configuration Corrupted Bit 6: Input 7 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 8 >= High Limit Bit 9: Input 8 >= High-High Limit Bit 10: Input 8 <= Low Limit Bit 11: Input 8 <= Low-Low Limit Bit 12: Input 8 Calibration Error Bit 13: Input 8 Configuration Corrupted Bit 14: Input 8 Incorrect Input Type Bit 15: Unused (will read as 0)	3
Digital Output and Analogue Output Data Spaces are unused by this module	
CONFIGURATION DATA SPACE	DIT (Offset)
Configuration Registers Reserved	0 – 16
Input 1 Type	17
Input 2 Type	18
Input 3 Type	19
Input 4 Type	20
Input 5 Type	21
Input 6 Type	22
Input 7 Type	23
Input 8 Type	24
where Input Type is chosen from one of the following: 0 = No Input Type Selected 40 = Input is 0 – 10 Volts 41 = Input is 2 – 10 Volts 42 = Input is 0 – 5 Volts 43 = Input is 1 – 5 Volts 44 = Input is 0 – 1 Volt 60 = Input is 0 – 20 mA 61 = Input is 4 – 20 mA 62 = Input is 0 – 50 mA 63 = Input is 10 – 50 mA *NB: Set input resistor value for current inputs*	
Display Format for each Input These bits set the data format for each analogue input either in physical units (Volts/milliAmps) or as a percentage of full scale. Bit = 0: Input reads as an integer representing milliVolts or microAmps (dependent upon type selected) Bit = 1: Input reads as an integer representing % of full scale x 100. (i.e. 0 to 10000 = 0 to 100% of range of selected Input Type) Individual bits are allocated as follows: Bit 0: Input 1 Display Format Bit 1: Input 2 Display Format Bit 2: Input 3 Display Format Bit 3: Input 4 Display Format Bit 4: Input 5 Display Format Bit 5: Input 6 Display Format Bit 6: Input 7 Display Format Bit 7: Input 8 Display Format Bits 8 to 15 must be set to 0	25
Configuration Registers Reserved (set to 0)	26 – 29
Input 1 High-High Limit (set in display format)	30
Input 1 High Limit	31
Input 1 Low Limit	32
Input 1 Low-Low Limit	33
Input 1 Deadband	34

Input 2 High-High Limit	35
Input 2 High Limit	36
Input 2 Low Limit	37
Input 2 Low-Low Limit	38
Input 2 Deadband	39
Input 3 High-High Limit	40
Input 3 High Limit	41
Input 3 Low Limit	42
Input 3 Low-Low Limit	43
Input 3 Deadband	44
Input 4 High-High Limit	45
Input 4 High Limit	46
Input 4 Low Limit	47
Input 4 Low-Low Limit	48
Input 4 Deadband	49
Input 5 High-High Limit	50
Input 5 High Limit	51
Input 5 Low Limit	52
Input 5 Low-Low Limit	53
Input 5 Deadband	54
Input 6 High-High Limit	55
Input 6 High Limit	56
Input 6 Low Limit	57
Input 6 Low-Low Limit	58
Input 6 Deadband	59
Input 7 High-High Limit	60
Input 7 High Limit	61
Input 7 Low Limit	62
Input 7 Low-Low Limit	63
Input 7 Deadband	64
Input 8 High-High Limit	65
Input 8 High Limit	66
Input 8 Low Limit	67
Input 8 Low-Low Limit	68
Input 8 Deadband	69
Input 1 Resistor Value in ohms (for current inputs only)	70
Input 2 Resistor Value in ohms (for current inputs only)	71
Input 3 Resistor Value in ohms (for current inputs only)	72
Input 4 Resistor Value in ohms (for current inputs only)	73
Input 5 Resistor Value in ohms (for current inputs only)	74
Input 6 Resistor Value in ohms (for current inputs only)	75
Input 7 Resistor Value in ohms (for current inputs only)	76
Input 8 Resistor Value in ohms (for current inputs only)	77

7.16 M1432C – 8TC ISO – 8 way Isolated Thermocouple/milliVolt Input Module

ANALOGUE INPUT DATA SPACE	DIT (Offset)
Analogue Input 1	0
Analogue Input 2	1
Analogue Input 3	2
Analogue Input 4	3
Analogue Input 5	4
Analogue Input 6	5
Analogue Input 7	6
Analogue Input 8	7

Analogue Data is read as a signed integer as specified in “milliVolt Display Format” or “Temperature Scale” configuration registers as appropriate.

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs 1-16 where individual inputs are allocated as follows: Bit 0: Input 1 >= High Limit Bit 1: Input 1 >= High-High Limit Bit 2: Input 1 <= Low Limit Bit 3: Input 1 <= Low-Low Limit Bit 4: Input 1 Calibration Error Bit 5: Input 1 Configuration Corrupted Bit 6: Input 1 Incorrect Input Type Bit 7: Input Burnout Bit 8: Input 2 >= High Limit Bit 9: Input 2 >= High-High Limit Bit 10: Input 2 <= Low Limit Bit 11: Input 2 <= Low-Low Limit Bit 12: Input 2 Calibration Error Bit 13: Input 2 Configuration Corrupted Bit 14: Input 2 Incorrect Input Type Bit 15: Input Burnout	0
Digital Inputs 17-32 where individual inputs are allocated as follows: Bit 0: Input 3 >= High Limit Bit 1: Input 3 >= High-High Limit Bit 2: Input 3 <= Low Limit Bit 3: Input 3 <= Low-Low Limit Bit 4: Input 3 Calibration Error Bit 5: Input 3 Configuration Corrupted Bit 6: Input 3 Incorrect Input Type Bit 7: Input Burnout Bit 8: Input 4 >= High Limit Bit 9: Input 4 >= High-High Limit Bit 10: Input 4 <= Low Limit Bit 11: Input 4 <= Low-Low Limit Bit 12: Input 4 Calibration Error Bit 13: Input 4 Configuration Corrupted Bit 14: Input 4 Incorrect Input Type Bit 15: Input Burnout	1
Digital Inputs 33-48 where individual inputs are allocated as follows: Bit 0: Input 5 >= High Limit Bit 1: Input 5 >= High-High Limit Bit 2: Input 5 <= Low Limit Bit 3: Input 5 <= Low-Low Limit Bit 4: Input 5 Calibration Error Bit 5: Input 5 Configuration Corrupted Bit 6: Input 5 Incorrect Input Type Bit 7: Input Burnout Bit 8: Input 6 >= High Limit Bit 9: Input 6 >= High-High Limit Bit 10: Input 6 <= Low Limit Bit 11: Input 6 <= Low-Low Limit Bit 12: Input 6 Calibration Error Bit 13: Input 6 Configuration Corrupted Bit 14: Input 6 Incorrect Input Type Bit 15: Input Burnout	2

Digital Inputs 49-64 where individual inputs are allocated as follows: Bit 0: Input 7 >= High Limit Bit 1: Input 7 >= High-High Limit Bit 2: Input 7 <= Low Limit Bit 3: Input 7 <= Low-Low Limit Bit 4: Input 7 Calibration Error Bit 5: Input 7 Configuration Corrupted Bit 6: Input 7 Incorrect Input Type Bit 7: Input Burnout Bit 8: Input 8 >= High Limit Bit 9: Input 8 >= High-High Limit Bit 10: Input 8 <= Low Limit Bit 11: Input 8 <= Low-Low Limit Bit 12: Input 8 Calibration Error Bit 13: Input 8 Configuration Corrupted Bit 14: Input 8 Incorrect Input Type Bit 15: Input Burnout	3
Digital Output and Analogue Output Data Spaces are unused by this module	
CONFIGURATION DATA SPACE	DIT (Offset)
Input 1 Type	17
Input 2 Type	18
Input 3 Type	19
Input 4 Type	20
Input 5 Type	21
Input 6 Type	22
Input 7 Type	23
Input 8 Type	24
where Input Type is chosen from one of the following: 0 = No Input Type Selected 1 = Thermocouple Type K 2 = Thermocouple Type J 3 = Thermocouple Type E 4 = Thermocouple Type N 5 = Thermocouple Type T 6 = Thermocouple Type R 7 = Thermocouple Type S 8 = Thermocouple Type B 20 = milliVolts	
MilliVolt Display Format for each Input These bits set the data format for each analogue input either in physical units (Volts/milliAmps) or as a percentage of full scale when the milliVolt range is selected. Bit = 0: Input reads as an integer representing milliVolts. Bit = 1: Input reads as an integer representing % of full scale x 100. (i.e. 0 to 10000 = 0 to 100% of range of range Input Type) Individual bits are allocated as follows: Bit 0: Input 1 Display Format Bit 1: Input 2 Display Format Bit 2: Input 3 Display Format Bit 3: Input 4 Display Format Bit 4: Input 5 Display Format Bit 5: Input 6 Display Format Bit 6: Input 7 Display Format Bit 7: Input 8 Display Format Bits 8 to 15 must be set to 0	25
Temperature Scale (degrees C or F) Bit = 0: All temperature Inputs read as a signed integer representing tenths of degrees C (e.g. 1000 = 100.0 degrees C) Bit = 1: All temperature inputs read as signed integer representing tenths of degrees F	26

Upscale/Downscale Burnout for each Input These bits specify whether the input reading should go upscale, or downscale when a break in the input is detected. Bit = 0: Input reads +32767 when no input signal is connected. Bit = 1: Input reads –32768 when no input signal is connected. Individual bits are allocated as follows: Bit 0: Input 1 Burnout setting Bit 1: Input 2 Burnout setting Bit 2: Input 3 Burnout setting Bit 3: Input 4 Burnout setting Bit 4: Input 5 Burnout setting Bit 5: Input 6 Burnout setting Bit 6: Input 7 Burnout setting Bit 7: Input 8 Burnout setting Bits 8 to 15 must be set to 0	27
CJC Type (Internal or External Cold Junction Compensation) 0 = Internal CJC; 1 = External CJC	28
External CJC Temperature When external CJC is selected, this register must be set to the external CJC temperature in tenths of a degree C or F (as set in Temperature Scale).	29
Input 1 High-High Limit (set in display format)	30
Input 1 High Limit	31
Input 1 Low Limit	32
Input 1 Low-Low Limit	33
Input 1 Deadband	34
Input 2 High-High Limit	35
Input 2 High Limit	36
Input 2 Low Limit	37
Input 2 Low-Low Limit	38
Input 2 Deadband	39
Input 3 High-High Limit	40
Input 3 High Limit	41
Input 3 Low Limit	42
Input 3 Low-Low Limit	43
Input 3 Deadband	44
Input 4 High-High Limit	45
Input 4 High Limit	46
Input 4 Low Limit	47
Input 4 Low-Low Limit	48
Input 4 Deadband	49
Input 5 High-High Limit	50
Input 5 High Limit	51
Input 5 Low Limit	52
Input 5 Low-Low Limit	53
Input 5 Deadband	54
Input 6 High-High Limit	55
Input 6 High Limit	56
Input 6 Low Limit	57
Input 6 Low-Low Limit	58
Input 6 Deadband	59
Input 7 High-High Limit	60
Input 7 High Limit	61
Input 7 Low Limit	62
Input 7 Low-Low Limit	63
Input 7 Deadband	64

Input 8 High-High Limit	65
Input 8 High Limit	66
Input 8 Low Limit	67
Input 8 Low-Low Limit	68
Input 8 Deadband	69

7.17 M1433B – 6RTD ISO – 6 way Isolated Resistance Bulb Input Module

ANALOGUE INPUT DATA SPACE	DIT (Offset)
Analogue Input 1	0
Analogue Input 2	1
Analogue Input 3	2
Analogue Input 4	3
Analogue Input 5	4
Analogue Input 6	5
Analogue Data is read as a signed integer as specified in the “Temperature Scale” configuration register.	

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs 1-16 where individual inputs are allocated as follows: Bit 0: Input 1 >= High Limit Bit 1: Input 1 >= High-High Limit Bit 2: Input 1 <= Low Limit Bit 3: Input 1 <= Low-Low Limit Bit 4: Input 1 Calibration Error Bit 5: Input 1 Configuration Corrupted Bit 6: Input 1 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 2 >= High Limit Bit 9: Input 2 >= High-High Limit Bit 10: Input 2 <= Low Limit Bit 11: Input 2 <= Low-Low Limit Bit 12: Input 2 Calibration Error Bit 13: Input 2 Configuration Corrupted Bit 14: Input 2 Incorrect Input Type Bit 15: Unused (will read as 0)	0
Digital Inputs 17-32 where individual inputs are allocated as follows: Bit 0: Input 3 >= High Limit Bit 1: Input 3 >= High-High Limit Bit 2: Input 3 <= Low Limit Bit 3: Input 3 <= Low-Low Limit Bit 4: Input 3 Calibration Error Bit 5: Input 3 Configuration Corrupted Bit 6: Input 3 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 4 >= High Limit Bit 9: Input 4 >= High-High Limit Bit 10: Input 4 <= Low Limit Bit 11: Input 4 <= Low-Low Limit Bit 12: Input 4 Calibration Error Bit 13: Input 4 Configuration Corrupted Bit 14: Input 4 Incorrect Input Type Bit 15: Unused (will read as 0)	1
Digital Inputs 33-48 where individual inputs are allocated as follows: Bit 0: Input 5 >= High Limit Bit 1: Input 5 >= High-High Limit Bit 2: Input 5 <= Low Limit Bit 3: Input 5 <= Low-Low Limit Bit 4: Input 5 Calibration Error Bit 5: Input 5 Configuration Corrupted Bit 6: Input 5 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 6 >= High Limit Bit 9: Input 6 >= High-High Limit Bit 10: Input 6 <= Low Limit Bit 11: Input 6 <= Low-Low Limit Bit 12: Input 6 Calibration Error Bit 13: Input 6 Configuration Corrupted Bit 14: Input 6 Incorrect Input Type Bit 15: Unused (will read as 0)	2

Digital Inputs 49-64 where individual inputs are allocated as follows: Bit 0: Input 7 >= High Limit Bit 1: Input 7 >= High-High Limit Bit 2: Input 7 <= Low Limit Bit 3: Input 7 <= Low-Low Limit Bit 4: Input 7 Calibration Error Bit 5: Input 7 Configuration Corrupted Bit 6: Input 7 Incorrect Input Type Bit 7: Unused (will read as 0) Bit 8: Input 8 >= High Limit Bit 9: Input 8 >= High-High Limit Bit 10: Input 8 <= Low Limit Bit 11: Input 8 <= Low-Low Limit Bit 12: Input 8 Calibration Error Bit 13: Input 8 Configuration Corrupted Bit 14: Input 8 Incorrect Input Type Bit 15: Unused (will read as 0)	3
Digital Output and Analogue Output Data Spaces are unused by this module	
CONFIGURATION DATA SPACE	DIT (Offset)
Input 1 Type	17
Input 2 Type	18
Input 3 Type	19
Input 4 Type	20
Input 5 Type	21
Input 6 Type	22
Configuration Registers Reserved (set to 0)	23 - 25
where Input Type is chosen from one of the following: 0 = No Input Type Selected 100 = Input is PT-100 101 = Input is Ni-120	
Temperature Scale (degrees C or F) Bit = 0: All temperature Inputs read as a signed integer representing tenths of degrees C (e.g. 1000 = 100.0 degrees C) Bit = 1: All temperature inputs read as signed integer representing tenths of degrees F	26
Configuration Registers Reserved (set to 0)	27 – 29
Input 1 High-High Limit (set in display format)	30
Input 1 High Limit	31
Input 1 Low Limit	32
Input 1 Low-Low Limit	33
Input 1 Deadband	34
Input 2 High-High Limit	35
Input 2 High Limit	36
Input 2 Low Limit	37
Input 2 Low-Low Limit	38
Input 2 Deadband	39
Input 3 High-High Limit	40
Input 3 High Limit	41
Input 3 Low Limit	42
Input 3 Low-Low Limit	43
Input 3 Deadband	44
Input 4 High-High Limit	45
Input 4 High Limit	46
Input 4 Low Limit	47
Input 4 Low-Low Limit	48
Input 4 Deadband	49

Input 5 High-High Limit	50
Input 5 High Limit	51
Input 5 Low Limit	52
Input 5 Low-Low Limit	53
Input 5 Deadband	54
Input 6 High-High Limit	55
Input 6 High Limit	56
Input 6 Low Limit	57
Input 6 Low-Low Limit	58
Input 6 Deadband	59

7.18 M1760A – 32SOE – 32 way Sequence of Events Input Module - 24V Input M1761A – 32SOE – 32 way Sequence of Events Input Module - 48V Input

DIGITAL INPUT DATA SPACE	DIT (Offset)
Digital Inputs 1-16 where individual inputs are allocated as follows: Input 1 = Bit 0 (LSB) … Input 16 = Bit 15 (MSB)	0
Digital Inputs 17-32 where individual inputs are allocated as follows: Input 17 = Bit 0 (LSB) … Input 32 = Bit 15 (MSB)	1
Digital Inputs 33-48 where individual inputs are allocated as follows: Input Queue > 95% = Bit 0 (LSB) Reserved = Bits 1 to 16	2
Digital Inputs 49-64 Reserved = Bits 0 to 16	3
DIGITAL OUTPUT DATA SPACE	**DIT (Offset)**
Digital Outputs 1-16 where individual inputs are allocated as follows: Inhibit Input 1 = Bit 0 (LSB) … Inhibit Input 16 = Bit 15 (MSB)	0

Digital Outputs 17-32 where individual inputs are allocated as follows: Inhibit Input 17 = Bit 0 (LSB) Inhibit Input 18 = Bit 1 Inhibit Input 19 = Bit 2 Inhibit Input 20 = Bit 3 Inhibit Input 21 = Bit 4 Inhibit Input 22 = Bit 5 Inhibit Input 23 = Bit 6 Inhibit Input 24 = Bit 7 Inhibit Input 25 = Bit 8 Inhibit Input 26 = Bit 9 Inhibit Input 27 = Bit 10 Inhibit Input 28 = Bit 11 Inhibit Input 29 = Bit 12 Inhibit Input 30 = Bit 13 Inhibit Input 31 = Bit 14 Inhibit Input 32 = Bit 15 (MSB)	1
Digital Outputs 33-48 where individual inputs are allocated as follows: Flush Input Queue = Bit 0 (LSB) Reserved = Bits 1 to 16	2
Digital Outputs 49-64 Reserved = Bits 0 to 16	3
Analogue Input and Analogue Output Data Spaces are unused by this module	

CONFIGURATION DATA SPACE	DIT (Offset)
Reserved	0
Input Sense – Inputs 1 to 16. Bit 0 maps to input 1	1
Input Sense – Inputs 17 to 32. Bit 0 maps to input 17	2
Debounce Time	3
Input 1 On Delay Time	4
Input 2 On Delay Time	5
Input 3 On Delay Time	6
Input 4 On Delay Time	7
Input 5 On Delay Time	8
Input 6 On Delay Time	9
Input 7 On Delay Time	10
Input 8 On Delay Time	11
Input 9 On Delay Time	12
Input 10 On Delay Time	13
Input 11 On Delay Time	14
Input 12 On Delay Time	15
Input 13 On Delay Time	16
Input 14 On Delay Time	17
Input 15 On Delay Time	18
Input 16 On Delay Time	19
Input 17 On Delay Time	20
Input 18 On Delay Time	21
Input 19 On Delay Time	22
Input 20 On Delay Time	23
Input 21 On Delay Time	24
Input 22 On Delay Time	25
Input 23 On Delay Time	26
Input 24 On Delay Time	27
Input 25 On Delay Time	28
Input 26 On Delay Time	29

Input 27 On Delay Time	30
Input 28 On Delay Time	31
Input 29 On Delay Time	32
Input 30 On Delay Time	33
Input 31 On Delay Time	34
Input 32 On Delay Time	35
Input 1 Off Delay Time	36
Input 2 Off Delay Time	37
Input 3 Off n Delay Time	38
Input 4 Off Delay Time	39
Input 5 Off Delay Time	40
Input 6 Off Delay Time	41
Input 7 Off Delay Time	42
Input 8 Off Delay Time	43
Input 9 Off Delay Time	44
Input 10 Off Delay Time	45
Input 11 Off Delay Time	46
Input 12 Off Delay Time	47
Input 13 Off Delay Time	48
Input 14 Off Delay Time	49
Input 15 Off Delay Time	50
Input 16 Off Delay Time	51
Input 17 Off Delay Time	52
Input 18 Off Delay Time	53
Input 19 Off Delay Time	54
Input 20 Off Delay Time	55
Input 21 Off Delay Time	56
Input 22 Off Delay Time	57
Input 23 Off Delay Time	58
Input 24 Off Delay Time	59
Input 25 Off Delay Time	60
Input 26 Off Delay Time	61
Input 27 Off Delay Time	62
Input 28 Off Delay Time	63
Input 29 Off Delay Time	64
Input 30 Off Delay Time	65
Input 31 Off Delay Time	66
Input 32 Off Delay Time	67
Chatter Time	68
Queue Head Type	69

Example:
Input 29 of an M1760A module installed in Local Slot 12 can be read from DIT Address 5049, Bit 12.

8. The Subscription Service Explained

8.1 Introduction to Subscriptions

Central to many applications involving communications across networks is the need to replicate data between nodes on the network.

Examples include a SCADA system acquiring data from remote telemetry units in the field; or a point-to-point telemetry application, where inputs are transmitted from one location to outputs at another location.

In all these cases, the traditional method is for a controlling master node to poll the slave nodes regularly for data in case something has changed. This crude method is an inefficient use of the limited network bandwidth, and is inherently slow in typical and worst case update times. It also limits the number of master nodes in the system to one.

The Maxiflex P3 CPU provides a far superior mechanism to accomplish this commonly used function through its Subscription Service. This Service operates as follows:

The node requiring the data sets up a subscription with the source node, very much like you would subscribe to a magazine through your newsagent. You establish a magazine subscription by telling the newsagent which magazine you want, your home address, and how often you want it, and then the newsagent takes the responsibility on himself to send you the magazine whenever a new issue becomes available.

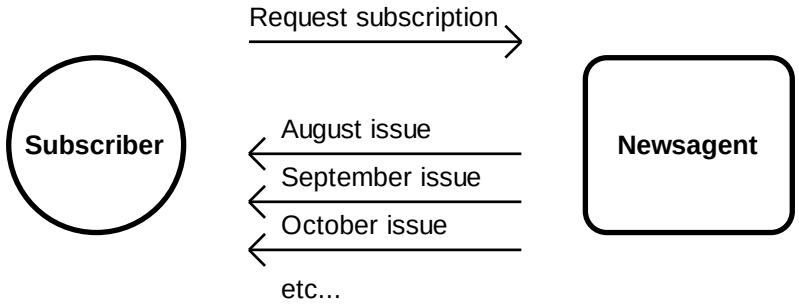


Figure 8.1 The Magazine Subscription Analogy.

In the same way, the P3 CPU's Subscription Service allows the CPU (acting as a node on the network) to subscribe to a range of DIT registers on a remote node.

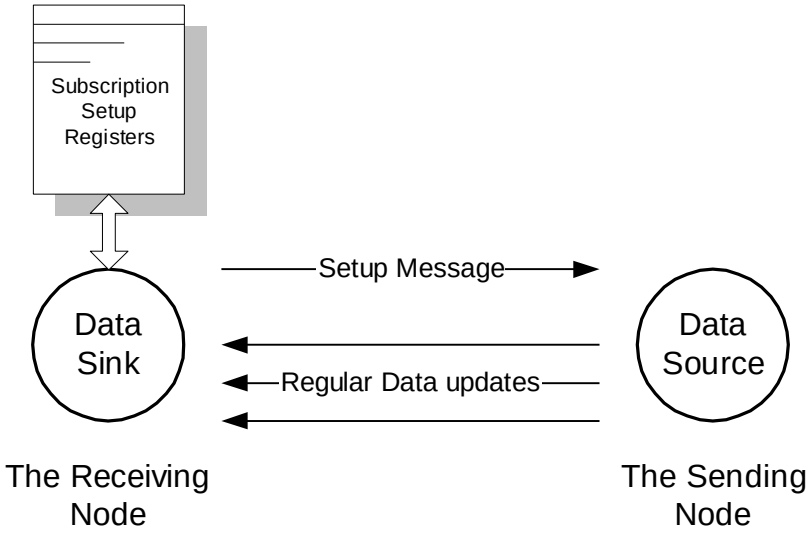
The remote node will then send the data at an agreed time interval, plus, if required, when the data changes state. The receiving node expects these regular updates, and will flag an error if the subscribed data is interrupted for any reason.

8.2 Setting up subscriptions

Just like the magazine subscription, the receiving CPU is responsible for setting up and maintaining subscriptions with other nodes to replicate data across the network. The advantages of using subscriptions over regular polling mechanisms are as follows:

1. Only one message is required on the network for a data update as opposed to two in a Request/Reply polling method. This reduces network overhead allowing more data throughput on the network.

2. The regular data updates can be much slower than the response time required for the system by using change-of-state detection. The Source node will send data immediately there is a change of state, providing the optimum system response, without the need to have a fast regular update time. This reduces network overhead allowing more/faster data throughput on the network when something does change.
3. On peer-to-peer networks multiple subscriptions can be configured between nodes in different directions, each of which would operate independently of any other. This provides far more flexibility than typical Poll/Response Master/Slave methods.



The node acting as a data “source” (the sending node) requires no user configuration to participate in a subscription contract with another node. The node acting as data “sink” (the receiving node) is configured by the user for the required data. The “sink” node then automatically manages the subscription including sending the requests to the data source, and managing errors in the process.

The following table shows the information required to configure a subscription:

ITEM	DESCRIPTION	VALID RANGE
Change-of-State Required	This sets whether the subscription data block will be sent when any data in the block changes.	Yes/No
Update time	This is the time between regular updates of data that will be sent whether the data has changed or not.	1 – 120 seconds or 1 – 120 minutes
Local Port Number	This is the Network Port number on the network module in the local (receiving) system to which the network containing the remote (source) node is connected. This item is only required if a local network ID is specified for the Source Node Address, otherwise it should be set to 0.	0 – 3 On the P3 CPU: 0 = CPU network port 2 = CPU Serial Port Refer to the relevant module user manual for port numbers on modules in other I/O slots.
Destination DIT Start Address	This is the DIT register address where the data will start to be written in this CPU acting as the data sink.	0 – 65535
Source Node Address	This is the network address of the sending or source node from which the required data originates. This can be expressed as a local network address plus the local slot and port to which the network is connected, or it can be expressed as a global network address if network routing is configured in the CPU.	Local network addresses: 1 – 125 Global network addresses: 128 – 254
Source DIT Start Address	This is the DIT register address of the first register in the block of registers to be sent from the source node.	0 – 65535
Register Range	This is the number of 16 bit DIT registers that will be transferred in the subscription	1 – 120

Table 8.1 Subscription Configuration Information

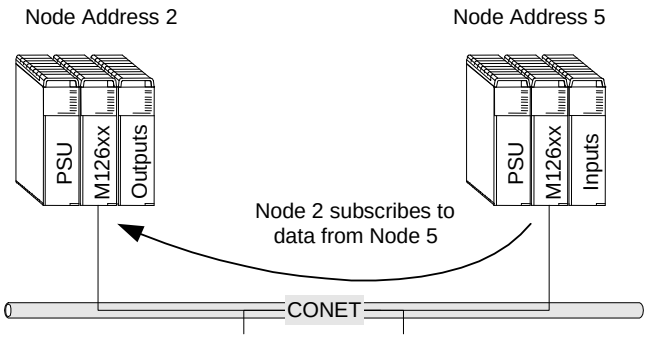
8.3 Number of subscriptions allowed

There is a limit to the number of subscriptions that a P3 CPU can receive, and a limit to the number of subscriptions that it can send. The P3 CPU can be configured to receive subscriptions from a maximum of 64 other nodes and can send subscriptions to a maximum of 16 other receiving nodes.

In other words, each P3 CPU can act as data sink for 64 subscriptions, and data source for 16 subscriptions simultaneously.

8.4 Subscription Application Example

Refer to the diagram of a simple network below:



Remote node 5 on the CONET network has digital information in DIT registers 5000 to 5007. The Central Node 2 needs to monitor these digital inputs. A regular update time of every 10s is quite satisfactory to determine that the remote system is still connected and functioning if nothing changes, but the data should be sent immediately if any of the digital inputs changes state.

Node 2 is the Destination node (it sets up the subscription and receives the data). Node 5 is the sending node, and requires no user configuration to participate in the subscription process.

Fill in the following data into one of the eight subscription blocks in the Subscription Table in the P3 CPU of Node 2 to configure this function:

NAME	VALUE	DESCRIPTION
Change of State Required	Yes	The data will also be sent whenever any bit the DIT registers in the range 620-627 changes.
Update Time	10	The regular updates can be 10 seconds apart.
Local Port Number	0	The CONET network is Port 0 on the CPU
Destination DIT Start Address	6000	This is the DIT Address where we want to start writing the received data.
Source Node Address	5	The local CONET Node Address on the Conet network
Source DIT Start Address	5000	The Source Data start at DIT address 620
Data Range	8	We require 8 DIT registers to be sent

Table 8.2 Subscription Block Data Example

When configured, Node 2 automatically requests a subscription with the following data from Node 5 on the Conet network connected to the CPU:

Node 5 undertakes to send the contents of its own DIT registers 5000 to 5007 over the network to node 2 whenever any of the registers changes or every 10s if no change occurs in that time. The information will be placed in node 2's DIT registers 6000 to 6007.

Node 2 can monitor the status of the subscription by monitoring the relevant bit in the Subscription Status DIT register. If this bit is clear then the subscription is operating successfully. If this bit is set, then the subscription has failed and received data is not valid, or the subscription has not been correctly set.

9. Queue Service Explained

9.1 Introduction to the Queue Service

The Queue Service is best described as a circular buffer of memory that contains blocks of formatted data referred to as queue records. The Queue Service provides a buffering interface between processes that produce queue records in real time and processes that consume those queue records in non-real time. A typical example of a SCADA system on PC, collecting time stamped event information in non-real time from a Maxiflex System configured as a Sequence of Events monitor, that is detecting events in Real Time.

The Maxiflex System makes use of the M176xA 32SOE modules to perform this function. These modules are scanning digital inputs for any change in their state (either '1' or '0'). When such an input does change state, this occurrence is called an **event**. Events are time stamped with the Date and Time of the event and a queue record is created that is then sent to the Queue Service in the P3 CPU. The SCADA system, by means of the Omniflex Conet OPC Server, is able to retrieve these events from the Queue Service for analysis at the SCADA.

10. Modbus Master Operation Explained

10.1 Introduction to Modbus Master Driver

The Modbus Master Driver supports up to 32 Queries to read and write data from third party devices. The same driver is able to send queries using the serial port (i.e. Modbus ASCII and RTU serial protocol) and the Ethernet port (i.e. Modbus TCP protocol) in the case of the M1262F and M1267B. The other CPUs only provide support for the serial protocol. These queries can be any combination of One-shot Queries and Cyclic Queries. In order to use the Modbus Master Driver over the serial port, the serial port must be configured for this use; otherwise none of the Modbus Master configuration changes for the serial port will take effect. Please refer to Section 11.9.2, MODBUS Master Protocol to set up the serial port in this way. In the case of Modbus TCP, no special protocol setting is required for the Ethernet port.

The Modbus Master Driver is extremely flexible to adapt to the many variances found in the Modbus Slave protocols found in third party devices. These variances include query response times and general performance i.e. how often a device can be polled for data. It is possible to adjust the poll rate per query, the delay between queries as well as the delay between the entire polling cycle.

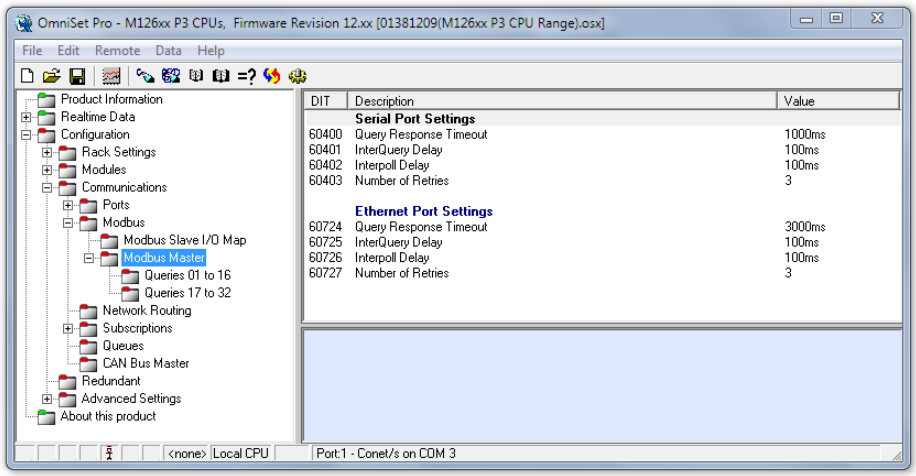
The Modbus Master Driver is also easily adaptable to application demands as it is possible to interleave One-shot and Cyclic queries as desired. One-shot queries are triggered by setting a bit in the DIT table, which is easily achieved from the local application program or remotely through any of the network ports. This remote device could, for example, be a PC running a SCADA application or another Omniflex device.

Data throughput can be controlled by setting the Update Times for Cyclic queries on a per query basis. This allows users to prioritise faster changing data over slower or less important data by setting a larger update time for less important data while keeping the fast data on a shorter update time. This will refresh the fast data more frequently than the slow.

The status of each query is stored in the DIT in bit format, 1 bit for each query, thereby providing open diagnostics to both application program and any remotely connected devices.

10.2 Modbus Master Parameters

These are some general parameters that control the manner in which queries are processed. They are configured via Omniset in the "Modbus Master" group as shown below:



Query Response Timeout

When a query has been sent to a slave device, the slave device must respond before this time period. If not, the Modbus Master Driver will assume an error and will either re-transmit the query or flag an error for that device.

The timeout period must be specified in milliseconds. E.g. if a timeout of 2 seconds is required then the timeout must be set to 2000. This parameter is common to one-shot polling.

Inter-query Delay

A delay may be configured if it is necessary to pause between each query to the Modbus Slave devices. The Inter-query delay enables the user to slow down the rate at which the Master polls the Slaves between each query. This is sometimes essential if the Slave device cannot cope with queries sent at full rate.

This delay must be specified in milliseconds. E.g. if a delay of 100ms is required then the register must be set to 100.

Inter-poll Delay

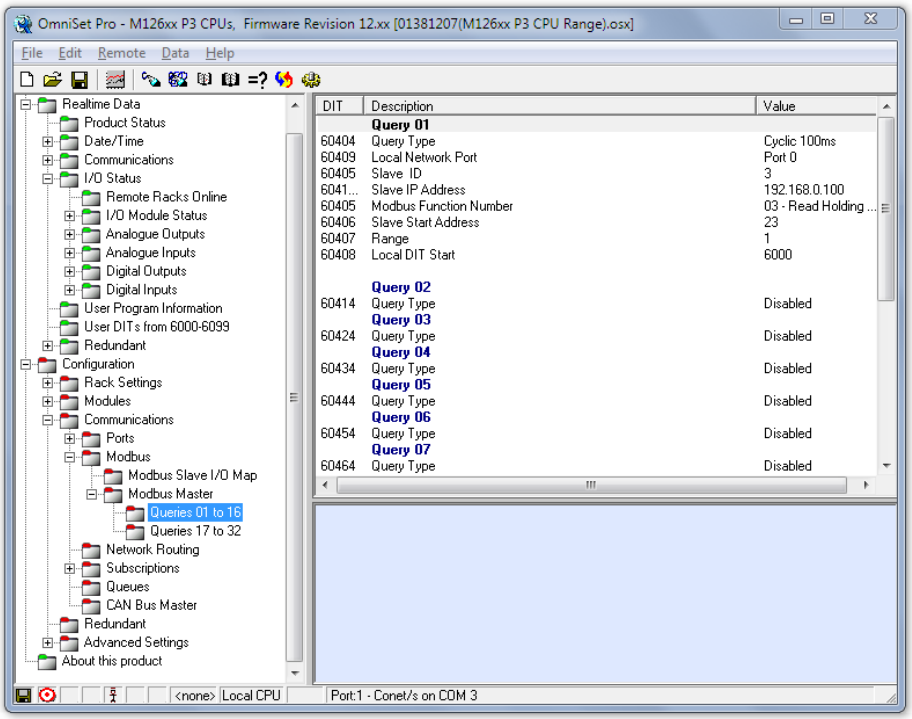
After a round of Cyclic and One-shot queries has been completed, a delay may be configured before the next round resumes. This delay is specified in milliseconds. E.g. if a delay of 2 seconds is required then the register must be set to 2000.

Number of Retries

When a query has failed i.e. the Modbus Master driver did not receive a reply within the Query Response Timeout period specified, the driver will check the number of retries setup and will re-transmit the query according to the number of retries configured. This feature is extremely useful in overcoming spurious transmission line interference as it allows the driver to recover a lost query before flagging an error. The error is flagged only if all retry attempts have failed.

10.3 Query Configuration

Up to 32 query messages may be configured. These Queries are configured using the "Setup Modbus Master Queries…" group using Omniset. There are two groups, 16 queries per group. The figure below shows the group for queries 1 to 16.



Below is a table of information required to setup Modbus Master Queries

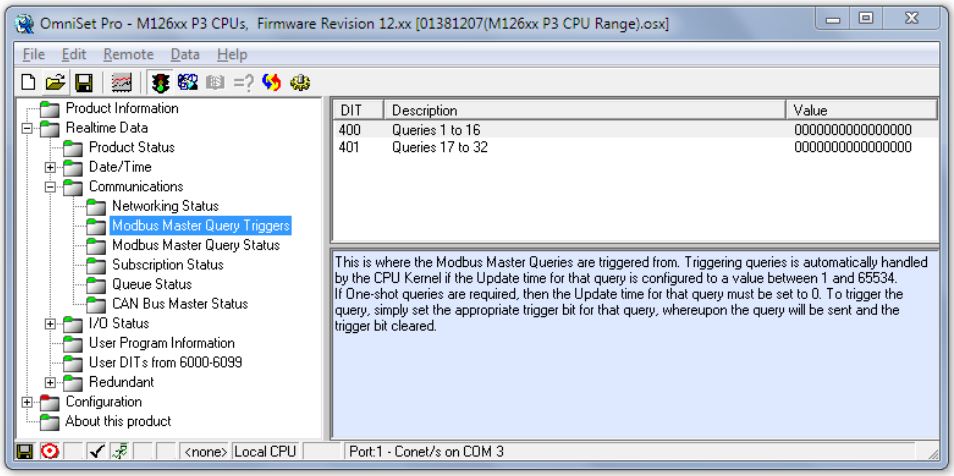
Name	Value(s)	Description
Query Type	Disabled One-shot As fast as possible Cyclic 100ms Cyclic 250ms Cyclic 500ms Cyclic 1000ms Cyclic 2000ms Cyclic 5000ms	Set disable to ensure the query is not executed at all. Set One-shot if its to be triggered manually Set As fast as possible or any other Cyclic option for Cyclic queries. If any of the predefined Cyclic times are not suitable, a unique time can be set in Update Time. The Cyclic query will not be resent until the time has expired.
Update Time	1 - 60000ms	Any suitable Cyclic period that may be required for this particular slave device.
Slave Address	1 - 32	The Modbus Slave address to whom this query is sent.
Modbus Function	1, 2, 3, 4, 5, 6, 15 and 16	The Modbus function to be performed needs to be specified here.

Slave Start Address	Any legal address in the slave address map.	Modbus Slave start register address (referenced to zero). E.g. if the desired register address of the Slave started at 30101, then the value entered here would be 100.
Range	1 (5, 6) 1 to 120 (3,4,16) 1 to 480 (1,2,15)	The number of coils/registers read or written. Legal values vary according to the Functions shown in the Value(s) column.
DIT Start register	500 to 65500	Where the CPU must either start retrieving data from or start saving data to in the DIT
DIT Start Bit	0 to 15	The DIT start bit specifies where in the DIT start register the digital functions (1, 2, 5 and 15) begin accessing the desired bits.

Table 10.1 Modbus Master Query Settings

10.4 Query Triggers

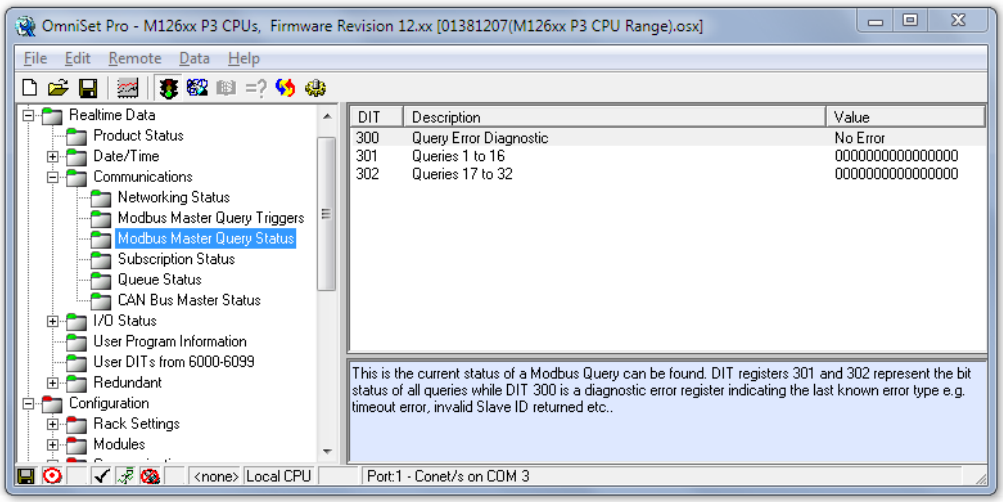
Query Triggers are the mechanism by which all queries are sent. One-shot queries are triggered manually, either by user application or by another device or PC connected remotely via one of the network ports on the CPU. Cyclic queries are triggered automatically by the operating system of the CPU. A single bit in the DIT table is allocated per query and when this bit is set, the query is triggered. The query trigger mechanism can be tested by setting the One-shot query bits using the "Modbus Master Query Triggers" group in Omniset.



Please refer to the detailed DIT layout for more information.

10.5 Status DIT Registers

Each Modbus query, whether Cyclic or One-shot has a status bit associated with it. This allows the user to quickly debug any problems with a particular query. These status bits are available in the DIT table.



There are a number of error responses for queries that allow accurate diagnosis of query problems. Many of which include the exception responses returned by a Slave device when the query message is received without communication errors but cannot be handled by the Slave device for some reason. This will be reflected in the Query Error Diagnostic register for the Last Failed Query. Refer to the Detailed DIT Layout for the DIT Location of this status register.

The Table below lists the various status codes for any given query.

Status	Description
0	Query Message successful – no errors
1-8	Modbus exception code as returned by Slave device- summary follows: 1 – Illegal Function Code 2 – Illegal Data Address 3 – Illegal Data Value 4 – Slave Device Failure 5 – Acknowledge 6 – Slave Device Busy 7 – Negative Acknowledge 8 – Memory Parity Error in Slave device
1000	Timed out waiting for response
1001	Node Address in response doesn't match Node Address in query
1002	Modbus function in response doesn't match function in query
1003	Received different number of coils/registers to what expected
1004	Invalid response to write query (functions 5,6,15 and 16)
65535	Invalid Configuration

Table 10.2 Modbus Master Query Error Codes

11. Configuring the P3 CPU

11.1 Overview

If you follow all of the instructions in this chapter you will have configured all the essential parameters required for most uses of the Maxiflex P3 CPU.

There are three places that configuration must be performed on a P3 CPU:

11.1.1 Communication Port Address Switches

The P3 CPU is equipped with a number of DIP switches inside the front door of the module that are used to set various network communications parameters.

See paragraphs 11.2 and 11.3 for further information.

11.1.2 User Configuration

This configuration is normally done using the Omniset Configuration Utility. See paragraph 11.3 onwards for more detailed information.

11.1.3 IEC61131 User Program

It is not necessary to have a User Program in the P3 CPU to use this product for conventional remote I/O purposes. A User Program is only required if you wish to implement some additional calculation or control.

The P3 CPU can be programmed in any of the IEC61131-3 programming languages using the Omniflex ISaGraf Programmer's Workbench.

Refer to the separate Omniflex ISaGraf Programmer's Workbench User Guide for information on programming the P3 CPU in the IEC61131 programming languages.

11.2 Setting the Serial Port Address Switch

If you are using the serial port on the front of the P3, you will need to choose a serial protocol (such as Modbus ASCII). For most serial protocols, the serial port requires an address to be set. If you are not using the serial port, then it is not necessary to set this switch.

This address is set on the address switch inside the front door of the module.

Figure 11.1: Serial Port Address Switch

The serial port on the P3 CPU comes equipped with the Modbus Master, Modbus Slave and Conet/s protocols. There are two ways to select the required protocol:

Using the DIP Switch, or in the configuration file. The DIP switch provides a quick and simple method of achieving communications using default settings in Modbus Slave and Conet/s.

For more control over the various settings, or for configuring Modbus Master, you must set the serial port in the configuration file using Omniset.

The required protocol including any address selection is made by the correct selection on the Serial Port Switch.

Follow the instructions below to set the required protocol mode:

Example of Serial Port DIP switch setting

[White square indicates position of switch lever]

Switches 1-5 = Node Address

This is set as a binary number with Switch 1 the least significant bit, and Switch 5 the Most significant bit.

To calculate the address setting, add the following weightings of the switches that are ON:

1 ON = +1

2 ON = +2

3 ON = +4

4 ON = +8

5 ON = +16

In this example an address of 2 is selected.

Switch 6 reserved.

(always leave switched off)

Switch 7 Default Conet/s (19200 baud) Protocol Select.

Switch 7 ON = Conet/s default mode of operation

In this example this switch is shown OFF.

Switch 8: Default Modbus ASCII Slave Protocol Selection

Switch 8 ON = Modbus ASCII slave at 9600 baud
(Data bits: 7;Parity: none; Stop bits: 1)

In this example, the Switch 8 is shown in the ON position to set the serial port to its default configuration of MODBUS Slave ASCII at 9600 baud.

Set as shown, (switches 2 and 7 on) the serial port will operate as a Modbus ASCII slave port at 9600 baud, and will respond to the slave address of 2.

Set both Switch 7 and 8 OFF to use internal software settings, set using Omniset. If either switch is on, then the internal software settings will be ignored.

Table 11.1: Serial Port Address switch settings

11.2.1 MODBUS Slave Protocol

11.2.1.1. Default MODBUS Mode - ASCII Slave at 9600 baud

To select this mode, set switch 7 “OFF” and 8 “ON” on the serial port address switch.

This port is then configured as a MODBUS slave port operating in MODBUS ASCII mode at a fixed 9600 baud with 7 data bits, no parity and 2 stop bits.

Switches 1 to 5 of the serial port address switch set the Modbus communications Slave address used to access the P3 CPU through this serial port.

The internal software settable parameters for this function have no influence on the operation of the serial port when switch 8 is on.

This mode allows foolproof communications to be established quickly and easily.

11.2.2 Conet/s (Peer-to-peer) Protocol

11.2.2.1. Default Conet/s Mode (19200 baud)

To select this mode, set switch 7 to “ON” and 8 to “OFF” on the serial port address selection switch.

This port is then configured as a Conet/s port operating at 19,200 baud (with 8 data bits, no parity and 1 stop bits.)

Switches 1 to 5 of the address switch set the node address used to access the P3 CPU through this serial port.

The internal software settable parameters for this port have no influence on the operation of the port when switch 7 is on, and switch 8 is off.

This mode allows foolproof Conet/s communications to be established quickly and easily, even when the internal settings for this port are unknown.

This mode allows foolproof Conet/s communications to be established quickly and easily, even when the internal settings for this port are unknown.

11.3 Setting the Conet/c Network Port Address Switch

The Conet/c Port selection switch is located on the left, inside the front door of the CPU (under the Conet DB9 connector). Use this switch to configure the Node address of the CPU on the Conet network and the desired CONET baud-rate (Normal or Slow).

Communications Protocol	Conet Address switch setting
Conet (Normal mode): Baud rate: 62,500 baud	**Switches 1-7: Conet ID**  (Set in binary Switch 1 = LSB Switch 7 = MSB) **Switch 8: Baud Rate** Switch 8 OFF = 62.5 kBaud [Switch shown set to Address 2, Normal baud rate]
Conet (Slow mode): Baud rate: 7,800 baud	Switches 1-7: Conet ID  Switch 8: Baud Switch 8 ON = 7.8 kBaud [Switch shown set to Address 2, Slow baud rate]

Table 11.2: Conet Port Address switch settings (M1261D P3c)

Each node on the Conet/c network should be allocated a unique address in sequence, starting at 1.

Please refer to the Conet Installation Guide and Conet Protocol Datasheet for more information on the CONET network.

11.4 Preparing the Omniset configuration software to configure the P3 CPU

Most of the features available on the P3 CPU can be configured by writing to Registers in the Data Interchange Table (DIT) of the CPU. (This excludes IEC61131 programming that is done using the Omniflex ISaGraf Programming Workbench.)

The recommended method for configuring a P3 CPU is to use the Microsoft Windows95/98/NT/2000/XP/7 compatible Omniset software configuration utility.

It is also possible to configure a P3 CPU through any of its communications ports that have access to the DIT, using any software capable of writing to the DIT registers of the CPU.

The full DIT register layout is available separately from Omniflex Technical Support, or it can be found in the Product Template File, viewed with Omniset.

11.4.1 Omniset

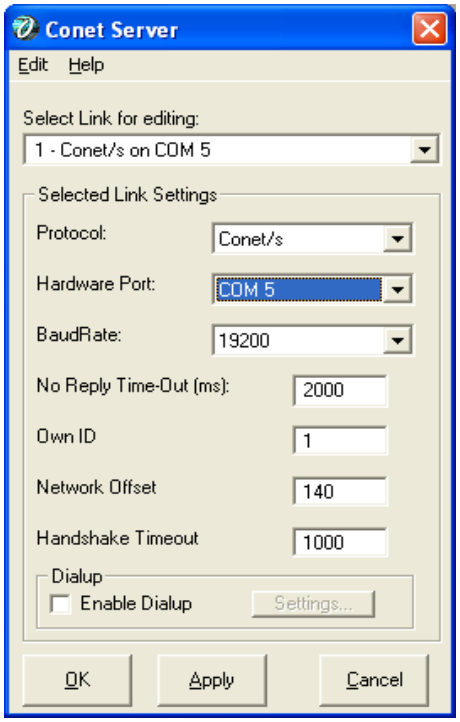
The Omniset configuration utility is available free of charge for the purpose of configuring a wide range of OMNIFLEX products, including this range of CPU's. A Template File compatible with the product is required, and is normally supplied with Omniset. Check for the latest available versions on the OMNIFLEX web site (www.omniflex.com) or with the Omniflex technical support department on techsupport@omniflex.com .

Using Omniset, you can configure the P3 CPU locally through the programming port. If you wish to configure the P3 CPU remotely over a network, then you must purchase Omniset Pro, the full featured version of this software.

11.4.2 Omniset Pro

1. The Omniset Pro software utility is an enhanced version of Omniset, and allows configuration of the P3 CPU remotely over a network. Omniset Pro also provides the ability to edit and create custom Template files.
2. For Remote I/O access you will need to configure the Network Offset of the Conet/s Link in the Conet Server. The Network Offset must be set to **140** as shown in the screen grab

below. This will allow access to the Remote Rack through the Programming Port of the Controller.



Wherever reference to Omniset is made throughout this text, Omniset Pro may be used.

11.4.3 Connecting Omniset to the CPU

To setup the P3 CPU through the programming port using Omniset or Omniset Pro, follow this procedure:

Plug the M1831A Maxiflex Programming Cable into the programming port of the CPU and into an unused RS232 serial port on your PC. If you are using a laptop that is not equipped with a serial port, most USB to RS232 converter products available can be used.

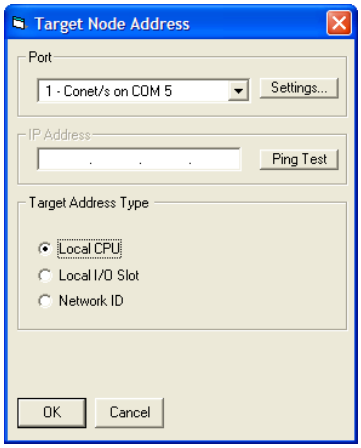
Figure 11.2: Location of Programming Port

Ensure that the Omniset software utility is running on your Laptop or PC.
(For more information on the installation and operation of OMNISET, please consult the Help File shipped with the Omniset software.)

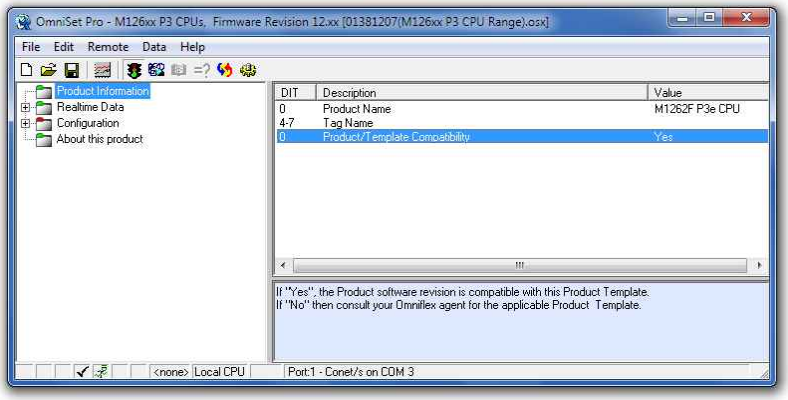
If you are using Omniset (not Omniset Pro), ensure that the correct COM port is selected in the dialogue box accessible from the “Options”->”Port Settings…” menu. (Omniset Pro uses a different method of communicating, via the CONET server. See the help supplied with Omniset Pro for further information to configure Omniset Pro for your computer.)

Open the template file named “013812*nn*(M126xx P3 CPU Range).osx” supplied with Omniset. (The *nn* is replaced with the revision number of the latest available template file).

1. Check the status bar (at the bottom of the Omniset Main Window) and confirm that the Target Address is set to “Local CPU”, and that the serial port displayed is the same serial port to which your programming cable is connected. If not, then this can be changed from the “File Target Address…” item in the “Remote” menu. Check that the settings are as per the window below. (The COM Port should reflect the selected COM port in your PC).



2. The “Connection Information” Group should now be properly displayed. If you are not connected, or the product type is not recognised, then a warning message will appear



If you can see the above information, and you see Omniset polling the target CPU regularly (check for the black tick flashing in the status bar), then you are communicating successfully with the P3 CPU, and you are ready to view or change any of the configuration parameters in the P3 CPU, and to view the internal dynamic data in the Maxiflex System as described below:

11.5 Synchronise Omniset and the P3 CPU

Your configuration data is stored in three places:

1. In the Omniset Configuration Program (held in memory on your computer while Omniset is running).

2. In a configuration file on your hard drive once you save your configuration to file from Omniset.
3. In your target P3 once you download the configuration to the target P3 CPU.

If you are configuring on-line (i.e. connected to the target P3 CPU), then you can synchronise the target P3 CPU and Omniset so that any changes you make to your configuration in Omniset are automatically downloaded to the target P3 CPU the moment you change them.

To synchronise Omniset with your target P3, you must perform one of the following actions:

1. Read All Groups from your target P3 into Omniset.
2. Write All Groups from Omniset to your target P3.
3. Verify All Groups to confirm that the configuration in your P3 target matches the configuration in the configuration file loaded in Omniset.

At the end any of these actions, you will be given the opportunity to elect to stay in Synchronism with your target P3 CPU. This is recommended if you will be changing quite a number of settings.

All future configuration instructions in this chapter assume that your Omniset is synchronised with your target P3 CPU.

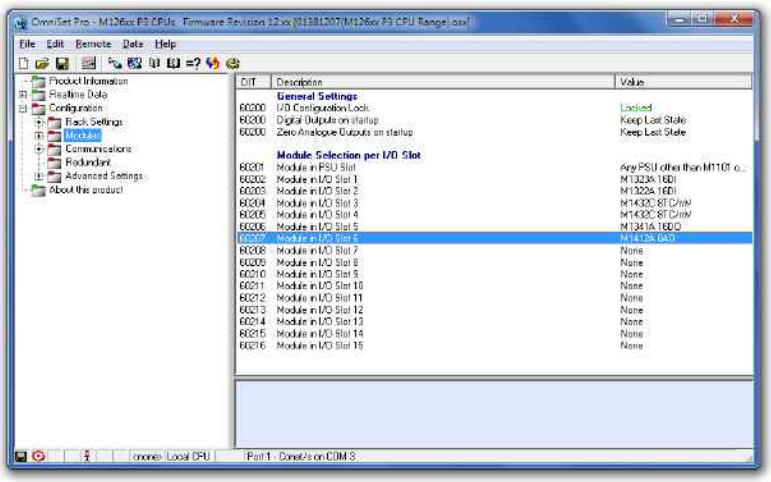
11.6 Configuring the Programming Port

The P3 CPU programming port is dedicated to the Programming and Configuration of the P3 CPU and requires no configuration.[1]

11.7 Configuring the I/O Module List using Omniset

To configure the P3 I/O Module List, follow this procedure:

1. Select the “Modules” Group under “Configuration”.



2. Select the required module type in each Slot in the Rack.

[1] This port is permanently configured for the Conet/s protocol (operating at 19,200 baud, 8 data bits, no parity, and 1 stop bit.)

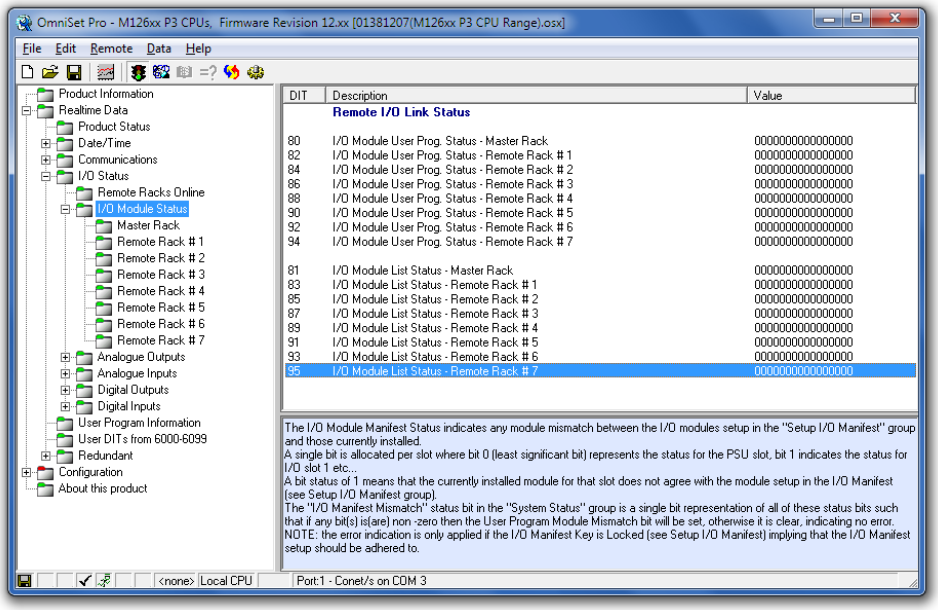
3. When setting up the Module List in the Remote Rack remember to set the Network Offset of the Conet/s Link to **156** as shown below. This will allow you access to the Remote Rack through the Programming Port of the Controller.

11.7.1 Check the Status of installed modules

Once the I/O Module List has been set, you can check if the modules currently installed match this list.

To check whether the Maxiflex modules actually installed on your Maxiflex Racks match your configured I/O Module List, follow this procedure:

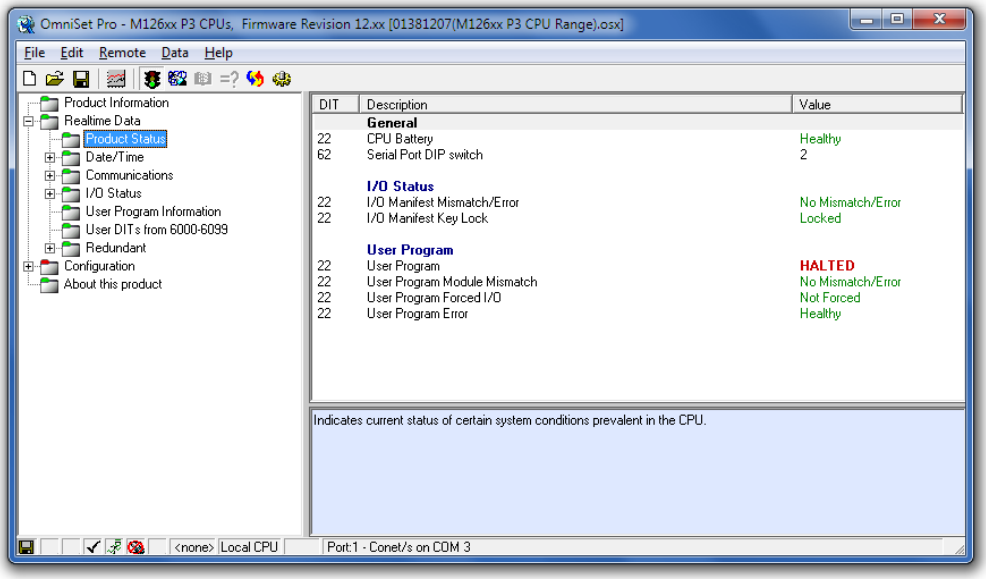
1. Change to the “I/O Module Status” Group:



2. The eight “I/O Module List Status” registers identify any mismatches between the four I/O Module Lists and the actual modules installed in each rack of the system. The eight “I/O Module User Prog.Status” registers identify the mismatches between the I/O module configuration as set within an IEC61131 program, if any, running on the CPU.

   Each bit set to 1 indicates a mismatch in the corresponding module Slot. The Least Significant (Right Most) Bit, Bit 0, represents the module in the PSU Slot; the next bit, Bit 1, represents the module in I/O Slot 1; up to the Most Significant Bit (Left Most) Bit 15 in the register represents the module in I/O Slot 15.

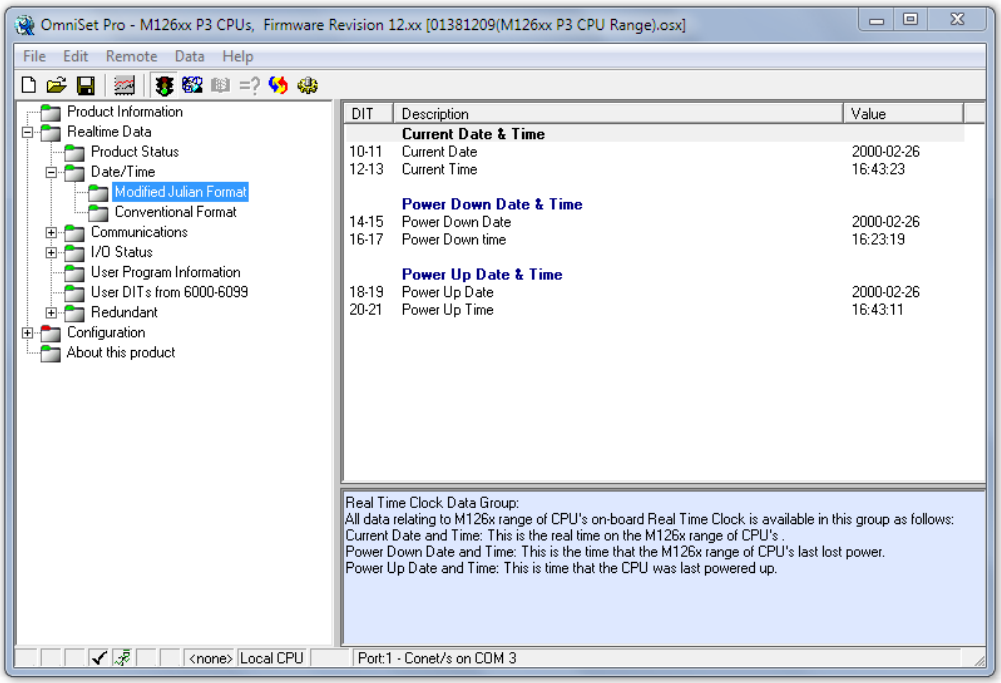
   If all modules match, (=0), then the I/O Module List Status and the I/O User Prog Status bits in the overall system status register will be set to 0 (shown as “OK” below). If any of the modules do not match, then this bit will be set to 1.



11.8 Configuring the Real-time Clock

The Real Time Clock Data Group shows the current time and date, the time and date of the last power down, and the time and date of the last power up.

To set the real-time clock, write the current time and date to the relevant Data Items in this Data Group. The clock will run immediately from this new time when it is written to the CPU.



11.9 Configuring the Serial Port from Omniset

11.9.1 MODBUS Slave Protocol

11.9.1.1. Default MODBUS Mode - ASCII Slave at 9600 baud

The full range of options for the Modbus Slave protocol is settable in Omniset in the “Serial Port” configuration group, including ASCII and RTU mode, Baud Rate, Parity, number of Data bits and Stop bits etc.

To select this mode, set both switches SW7 and SW8 of the serial port address switch OFF.

Switches 1 to 5 of the switch still set the Modbus Slave address used to access the P3 CPU through this serial port.

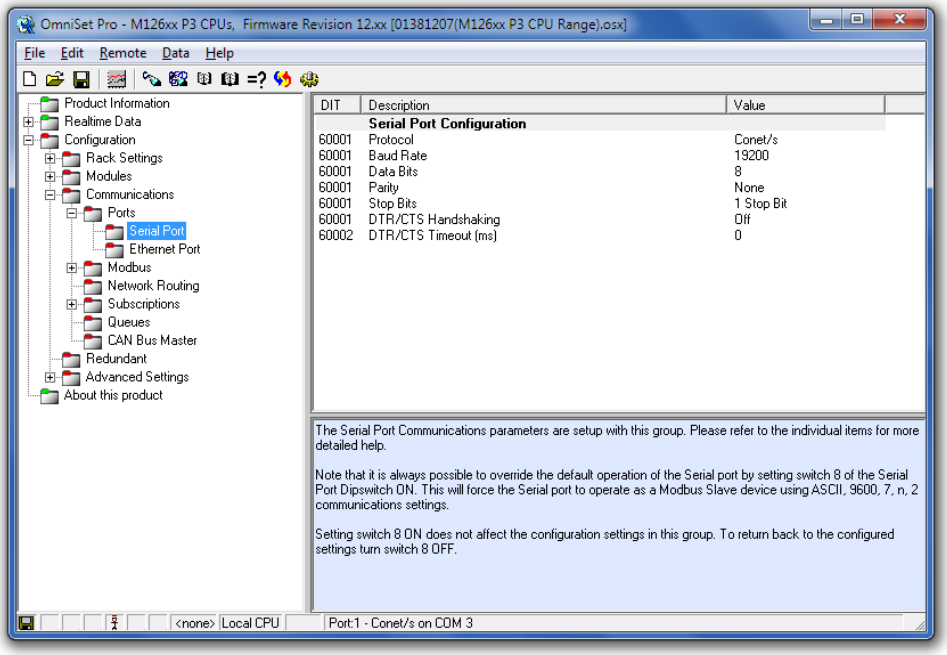


Figure 11.3: Serial Port Configuration Group

11.9.1.2. Modbus Data Register Mapping

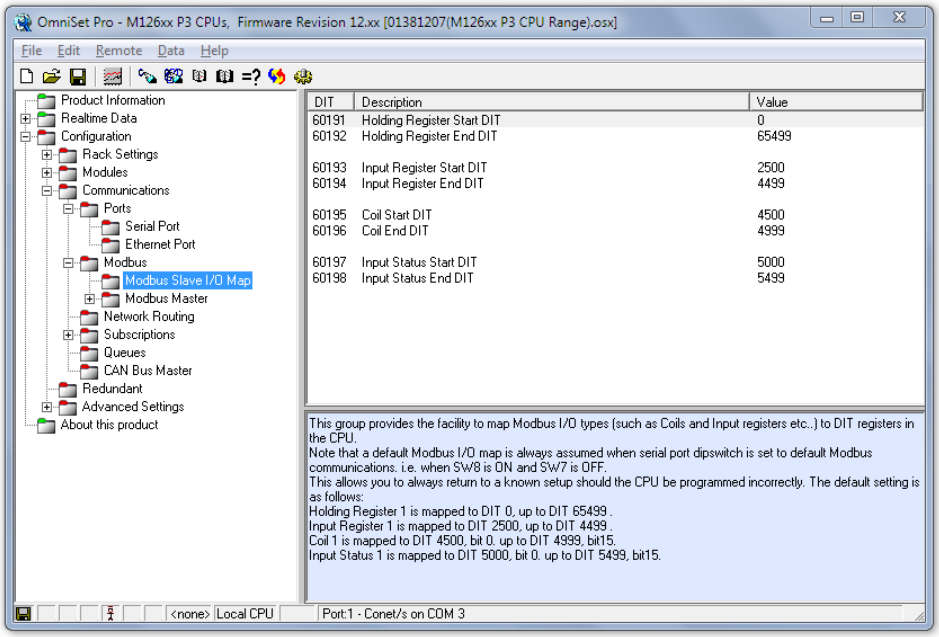
The Modus protocol supports a number of different types of data viz. Digital Status Inputs, Coil Outputs, Input Registers and Holding Registers.

These Modbus Datatypes are mapped to specific areas of the P3 CPU’s DIT.

Select the “Setup Modbus I/O Map” Data Group in Omniset to change these specific areas if required.

As factory default, the Input Statuses are mapped to the DIT Data Space containing the Maxiflex Digital Inputs, the Coils are mapped to the Data Space containing the digital outputs, and the input registers are mapped to the Data Space containing the analogue inputs.

This same Modbus Mapping is applicable to both the Modbus protocol used through the serial port of all CPU types, and the Modbus/TCP protocol used through the Ethernet port of the P3e CPU equipped with Ethernet.



11.9.2 MODBUS Master Protocol

11.9.2.1. Standard MODBUS Master Protocol Mode

Modbus Master Protocol is selected in the Serial ports group as shown in Figure 11.3

Once the protocol selection has been made, the rest of the Modbus Master configuration is done in the Modbus Master set of subgroups.

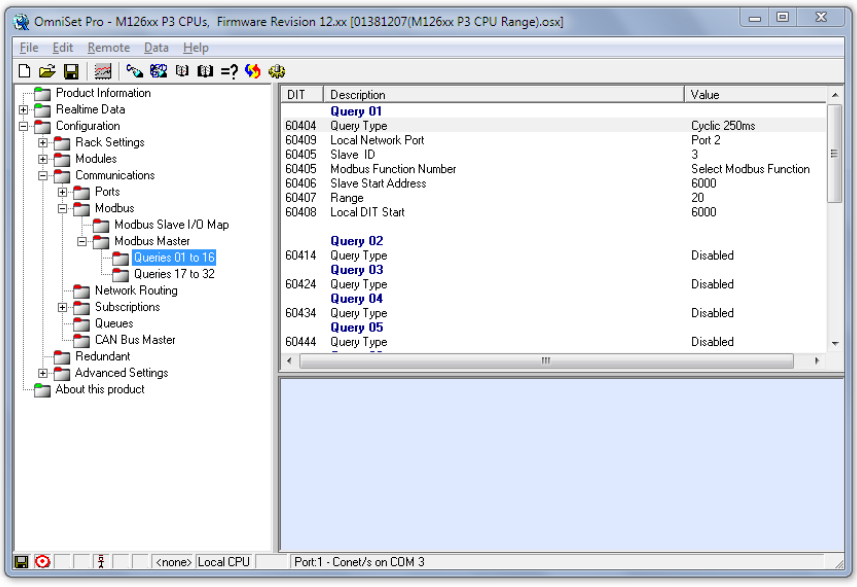


Figure 11.4: Configuring Modbus Master Queries

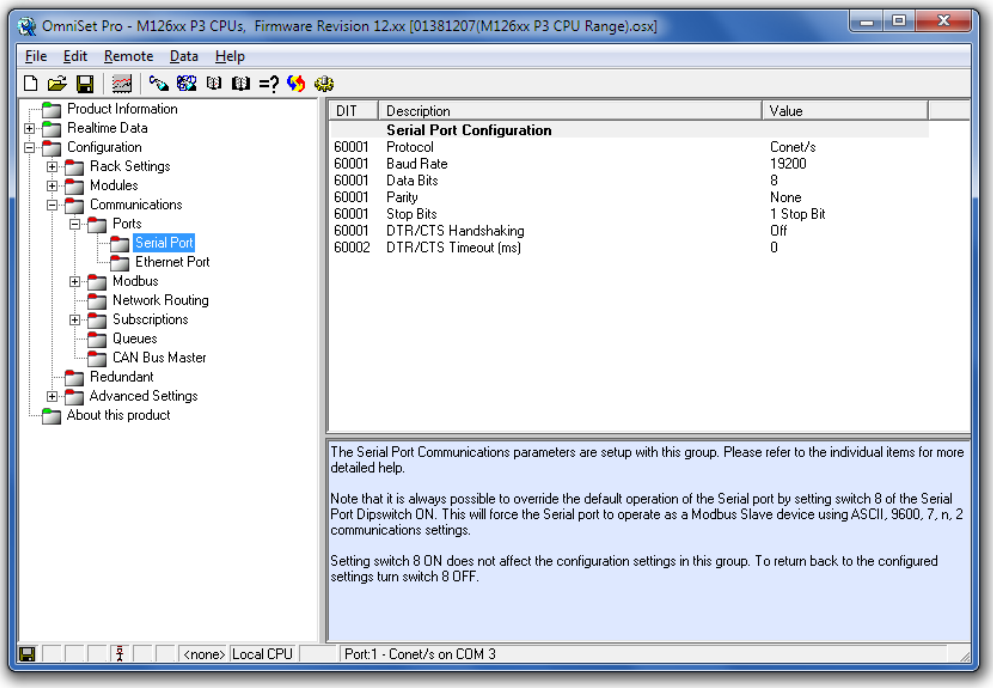
It is necessary to configure some general parameters that control the polling regime required in the "Modbus Master" group. The required Modbus Master Queries can be setup for polling Modbus Slave devices. As shown in the figure above.

11.9.3 Conet/s (Peer-to-peer) Protocol

The Conet/s protocol allows the peer-to-peer networking to be extended across conventional serial links. This provides functionality such as remote programming and configuration and the subscription data service not available using conventional master/slave protocols such as Modbus.

11.9.3.1. Standard Conet/s Protocol Mode

The baud rate of the Conet/s protocol is settable via Omniset in the “Serial Port” group.



In this mode, switch 7 and 8 of the serial port address switch must be OFF.

Switches 1 to 5 of the DIP switch set the communications address used to access the P3 CPU through this serial port. The baud rate etc. for this port is then set in the DIT.

11.10 Configuring the Ethernet Network Port

11.10.1 Protocols supported simultaneously

The Ethernet port on the P3e CPU’s runs the TCP/IP transport protocol.

This Ethernet port supports both the Modbus/TCP data protocol and the Conet/e data protocol, (both of which utilise the TCP/IP or UDP transport protocol).

Up to eight TCP/IP connections may be established simultaneously using either the Modbus/TCP or Conet/e protocols.

This allows, for example, a supervisory computer to be polling data from the P3e CPU using the Modbus/TCP protocol, while downloading a program remotely from an engineering workstation using the Conet/e protocol.

To use either of these protocols with the P3e CPU requires only the setting of the IP address and the Node Address in the CPU, and the same addresses in the supervisory system.

11.10.2 Assigning an IP Address

This port requires an IP address for its operation. The IP address is used to differentiate nodes on the Ethernet network.

Set this address in the “Ethernet Port” Configuration Group to suit your Ethernet network. If you are unsure of the Ethernet IP address to assign, ask your IT person responsible for the network.

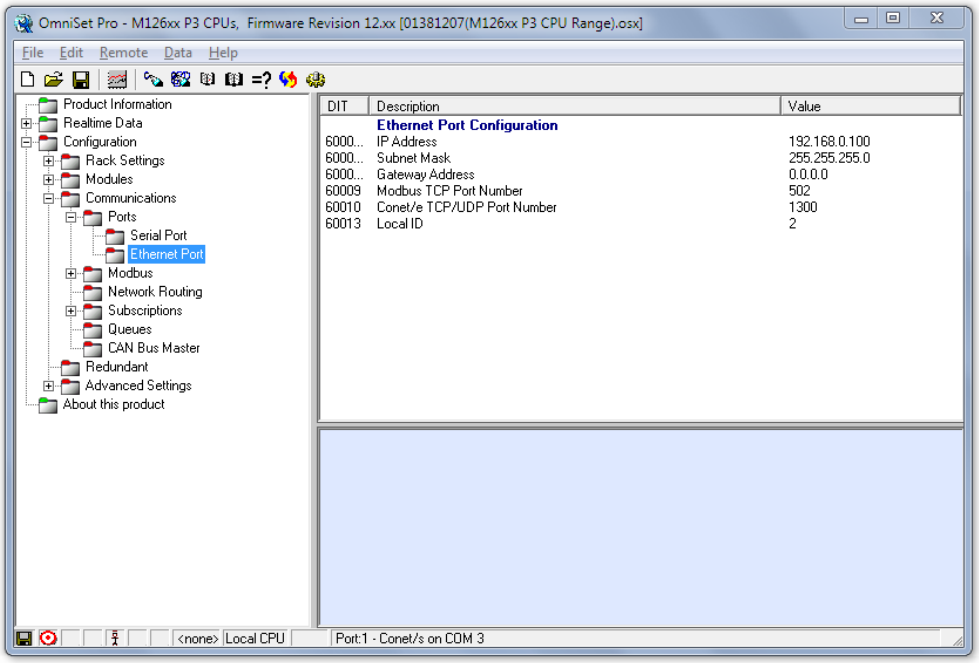


Figure 11.5: Setting Ethernet IP Address etc. in Omniset

11.10.3 Assigning a Node Address

Both the Modbus/TCP and the Conet/e protocols also require a Node address to be specified (sometimes called Unit ID or Slave Address). This address is preset to 2 in this product as a default but may be changed to suit application needs by assigning the value in this “Ethernet Port” Configuration Group as shown above.

11.10.4 TCP Port Number

Both Modbus and Conet/e are assigned TCP Port numbers. These can be changed if required, but this is not normally necessary. Do not change the default values unless you have a reason to do so. Changing these values can cause network access to fail.

12. Programming the P3 CPU

12.1 Introduction to IEC61131-3 Programming

12.1.1 The IEC61131 Standard

The IEC61131 standard was created to fulfil the need for standardisation in PLC programming languages.

The IEC61131 standard is divided into a number of parts:

Part 1 General information Definition of basic terminology and concepts.

Part 2 Equipment requirements and tests Electronic and mechanical construction and verification tests. - published 1992

Part 3 Programmable languages PLC software structure, languages and program execution.

Part 4 User guidelines Guidance on selection, installation, maintenance of PLC's.

Part 5 Messaging service specification Software facilities to communicate with other devices using communications based on MAP Manufacturing Messaging Services.

Part 6 Communications via fieldbus Software facilities of PLC communications using IEC fieldbus

Part 7 Fuzzy control programming Software facilities, including standard function blocks for handling fuzzy logic within PLC's - published 1997

Part 8 Guidelines for the implementation of languages for programmable controllers Application and implementation guidelines for the IEC61131-3 languages.

The part applicable to PLC programming is IEC61131-3. The Maxiflex P3 CPU together with the Application Workbench conforms to this IEC standard for programming languages.

12.1.2 The IEC61131-3 Programming Languages

The IEC61131-3 standard defines 5 programming languages:

- Sequential Flow Chart (SFC)
  A graphical language for depicting sequential behaviour of a control system. It is used for defining control sequences that are time and event driven.
  Sequential Function Chart (SFC), the core language of the IEC 61131-3 standard, divides the process cycle into a number of well-defined steps, separated by transitions. The other languages are used to describe the actions performed within the steps and the logical conditions for the transitions. Parallel processes can easily be described using SFC.
- Function Block Diagram (FBD)
  A graphical language for depicting signal and data flows through function blocks - re-usable software elements. FBD is very useful for expressing the interconnection of control system algorithms and logic.
- Ladder Diagram (LD)
  A graphical language that is based on relay ladder logic - a technique commonly used to program current generation PLC's. However, the IEC Ladder Diagram language also allows the connection of user defined function blocks and functions and so can be used in a hierarchical design.
- Structured Text (ST)
  A high level textual language that encourages structured programming. It has a language structure (syntax) that strongly resembles PASCAL and supports a wide

range of standard functions and operators.
This language is primarily used to implement complex procedures that cannot be easily expressed with graphical languages (e.g. IF / THEN / ELSE, FOR, WHILE...).

- Instruction List (IL)
  A low level 'assembler like' textual language that is based on similar instruction list languages found in a wide range of today's PLC's.

The Maxiflex P3 CPU implements all 5 of these IEC61131-3 languages as well as a sixth language called "Flow Chart".

- Flow Chart (FC)
  Recognizing that virtually every engineer graduating from college today has programmed in Flow Chart, the Workbench fully supports graphical Flow Chart programming. The Flow Chart is an easy to read decision diagram where actions are organized in a graphic flow. Binary decisions are used to control the flow. The Flow Chart Editor has full support for connectors and sub-programs. Actions and tests can be programmed in LD, ST or IL.

12.2 Programming the Maxiflex P3 CPU

The Maxiflex P3 CPU can be programmed in any of the supported languages with the aid of the Omniflex ISaGraf Programmer's Workbench.

For program development, the Application Workbench provides powerful and intuitive Windows based graphical and textual editors with drag-and-drop, and cut-and-paste to enhance ease of use.

The Application Workbench offers the following features:

- Project Management
- I/O Definition
- Modular Programming
- Simulation
- Real-time on-line debugging
- Document Generation

Full instructions on the use of the Application Workbench are available in a separate manual.

This manual is restricted to instructions specific to the Maxiflex P3 CPU.

12.3 Installing the Omniflex ISaGraf Application Workbench for the Maxiflex P3 CPU.

The Omniflex ISaGraf Application Workbench installs together with the Omniset Pro configuration utility. This software suite allows remote programming as well as providing simultaneous access to the CPU by Omniset for remote configuration.

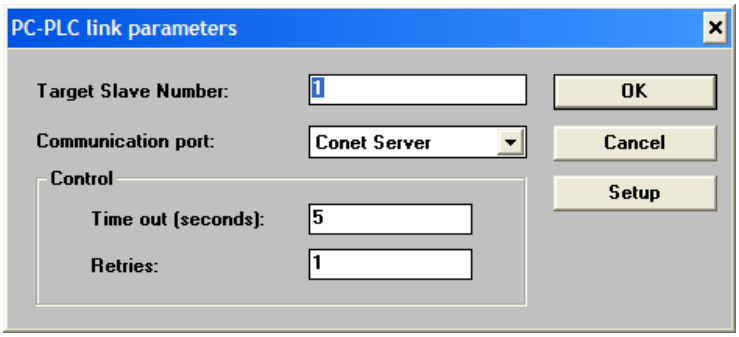
See the Application Workbench installation instructions supplied with the software to install the Application Workbench with Omniset Pro.

12.4 Setting up the Application Workbench for the Maxiflex P3 CPU.

The following settings in the workbench are applicable for the Maxiflex P3 CPU's:

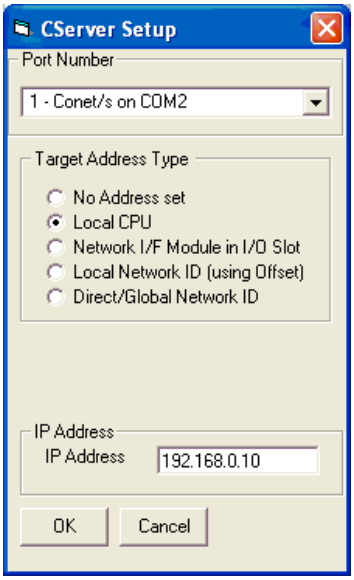
12.4.1 PC-PLC Link Parameters

1. Select "ISaGraf Projects" from the Start menu.
2. Start a new Project or open an existing Project.
3. Select "Link Setup" on the "Debug" menu.
4. Select all parameters as shown below. For remote programming over slow links, the Time out value may need to be increased if timeouts occur:



5. Select "Setup" and then choose the desired Conet Port and Target Address for communicating with the P3 CPU. Programming can be performed over Conet/s (through a serial port on the computer), Conet/c (through a Conet/c PCI Card if installed on the computer and Conet/e (through an Ethernet port if installed on the computer).

   For communicating with the P3 CPU through the CPU programming port via a serial port on your computer, select the following settings (Note: The actual COM port number may vary on your computer, depending upon which COM port you selected when installing the ISaGraf Workbench:



   An IP Address is only required if communicating using Conet/e over Ethernet. Select "Local CPU" to connect with this CPU over any of the ports.

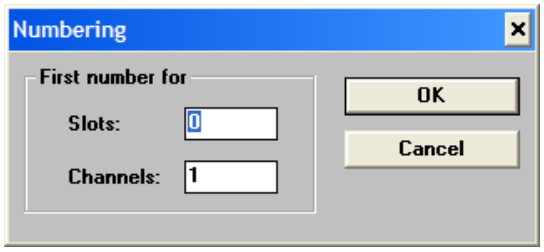
12.4.2 I/O Slot and Channel Numbering

You now need to ensure that the Slot Numbering starts from 0, and the Channel Numbering starts from 1 as follows:
In your open Project:

Open the “I/O Connection” Window from the “Project” menu.

Select “Numbering” from the “Options” menu in this window

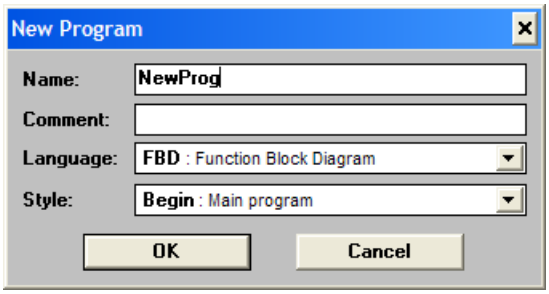
Check that your settings match the following:



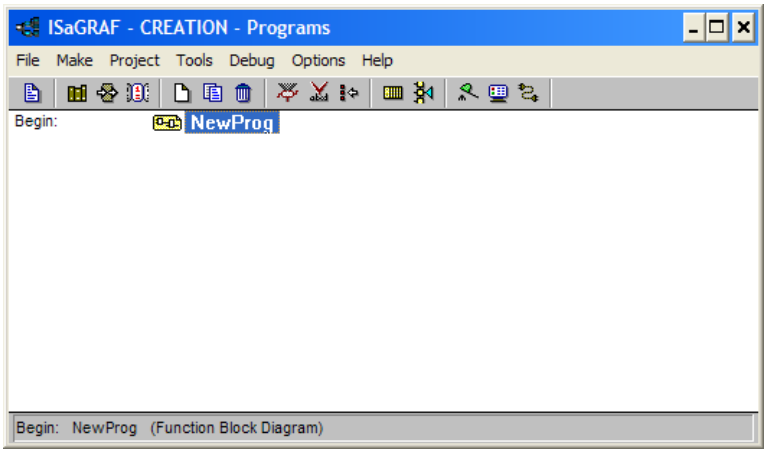
12.4.3 Compiler Options

Ensure that the following compiler options are selected in your Application Workbench for use with your Maxiflex P3 CPU:

1. From within your open project, start a new Program or open an existing program in the project (any language) as follows:
   To start a new program select File New. You will get the following dialogue box:



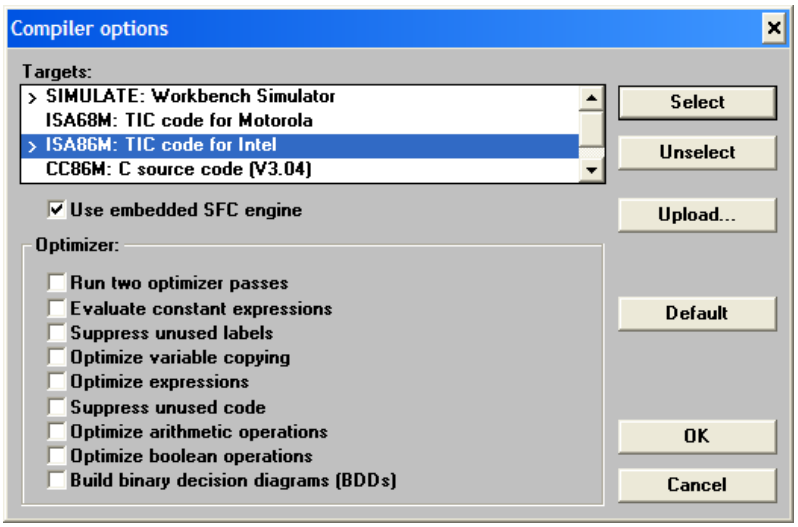
Fill in a program, select a language, and then press OK.
The program will appear in the project window



To open a program, select the program in the window and then select File Open,

(or double click the program name in the window)

2. In the Program Window, select “Compiler options” from the “Options” menu



Ensure that the “SIMULATE” and “ISA86M” options are both selected in the “Targets” list. To select an option in the list. Click on the item and press the “Select” button.

You can also select any compile optimisations that you require at this stage. (These can also be selected at any time in he future in the same way).

12.5 Creating I/O Connections

You must define the Maxiflex I/O Modules in your project in accordance with the physical layout of modules installed on the Maxiflex Bases of your target system.

If the modules in the project do not match the modules on the Maxiflex base, then your program cannot be started.

To define the modules in your project:

1. Start a new Project or open an existing Project.
2. Select “I/O Connection” from the “Project” menu (or use “I/O Connection” the button on the toolbar).
3. Select the I/O Module position to be defined.
   The following table gives the physical module positions on the Local and Remote Maxiflex bases vs. the Slot Numbers in this dialogue box:

ISaGraf Slot No.	Local Slot No.	Remote 1 Slot No.	Remote 2 Slot No.	Remote 3 Slot No	Remote 4 Slot No	Remote 5 Slot No	Remote 6 Slot No	Remote 7 Slot No
0	PSU							
1	1							
2	2							
3	3							
4	4							
5	5							

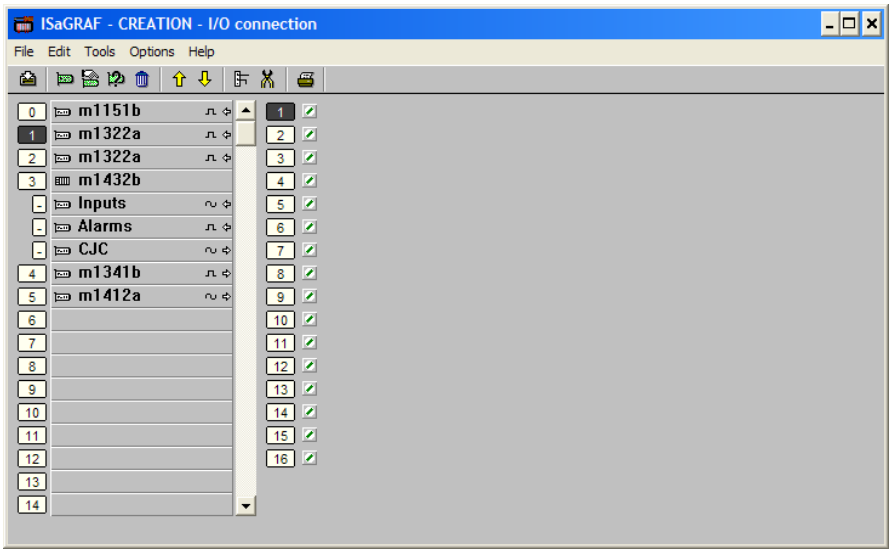
ISaGraf Slot No.	Local Slot No.	Remote 1 Slot No.	Remote 2 Slot No.	Remote 3 Slot No	Remote 4 Slot No	Remote 5 Slot No	Remote 6 Slot No	Remote 7 Slot No
6	6							
7	7							
8	8							
9	9							
10	10							
11	11							
12	12							
13	13							
14	14							
15	15							
16		PSU						
17		1						
18		2						
19		3						
20		4						
21		5						
22		6						
23		7						
24		8						
25		9						
26		10						
27		11						
28		12						
29		13						
30		14						
31		15						
32			PSU					
33			1					
34			2					
35			3					
36			4					
37			5					
38			6					
39			7					
40			8					
41			9					
42			10					
43			11					
44			12					
45			13					
46			14					
47			15					
48				PSU				
49				1				
50				2				
51				3				
52				4				
53				5				
54				6				
55				7				
56				8				
57				9				

ISaGraf Slot No.	Local Slot No.	Remote 1 Slot No.	Remote 2 Slot No.	Remote 3 Slot No	Remote 4 Slot No	Remote 5 Slot No	Remote 6 Slot No	Remote 7 Slot No
58				10				
59				11				
60				12				
61				13				
62				14				
63				15				
64					PSU			
65					1			
66					2			
67					3			
68					4			
69					5			
70					6			
71					7			
72					8			
73					9			
74					10			
75					11			
76					12			
77					13			
78					14			
79					15			
80						PSU		
81						1		
82						2		
83						3		
84						4		
85						5		
86						6		
87						7		
88						8		
89						9		
90						10		
91						11		
92						12		
93						13		
94						14		
95						15		
96							PSU	
97							1	
98							2	
99							3	
100							4	
101							5	
102							6	
103							7	
104							8	
105							9	
106							10	
107							11	
108							12	
109							13	

ISaGraf Slot No.	Local Slot No.	Remote 1 Slot No.	Remote 2 Slot No.	Remote 3 Slot No	Remote 4 Slot No	Remote 5 Slot No	Remote 6 Slot No	Remote 7 Slot No
110							14	
111							15	
112								PSU
113								1
114								2
115								3
116								4
117								5
118								6
119								7
120								8
121								9
122								10
123								11
124								12
125								13
126								14
127								15
128 onwards					Can be used for virtual I/O Modules (see section 12.6)			

Maxiflex Virtual Modules can be installed in any unused slots.

4. Select “Set Board/Equipment” from the “Edit” menu. (This selection can also be made by double clicking the Slot Number).
5. The Application Workbench separates I/O modules that have a single type of Input/Output from those with multiple Input/Output Types.
   Select “Boards” in the Library box to display all Maxiflex Modules with a single Input/Output type, or “Equipments” to display all Maxiflex Modules with multiple Input/Output types.
   You can view more information about each module using the “Note” button.
6. Select the desired Maxiflex module and press “Ok” to insert the module in the current slot.



7. Once Input and Output variables have been defined in your project dictionary, you can connect these variables to the appropriate Maxiflex Module Input or Output in this window by selecting “Select Channel/Parameter” from the “Edit” menu or double clicking the desired Channel number on the selected Slot.
   NOTE:
   Some Maxiflex I/O modules have “parameters” which must be set for the correct operation of those modules. See a full explanation in the Note attached to the Maxiflex Module.

12.6 Programming with the DIT

The Data Interchange Table provides a versatile repository for all data used within the Maxiflex P3 CPU. This data may be accessed in a User Program in a number of ways:

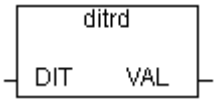
12.6.1 Using DIT Functions to access data in the DIT

A number of functions are available in the Application Workbench to access Data in the DIT. These can be found in the pull-down list of Functions available when editing a program in the FBD language, or by entering these functions as text in the ST or IL languages.

NOTE: These functions are implemented immediately, and are not synchronised to the program scan. This has the following consequences:

1. Data is written to the DIT when the function is implemented, and this data is available immediately for use in the same program scan.
2. DIT access to extended DIT areas (see section 6.4) take longer to execute than accesses to the local CPU DIT. If many such functions are included in the User Program, then this will have a significant effect on the scan rate of the program. Instead, in these circumstances, use the MxDIT_CPY Virtual I/O module to copy the data from the extended DIT area to the local DIT in blocks, where the data can be accessed at full speed.

12.6.1.1. DITRD - Read a DIT register



Arguments:

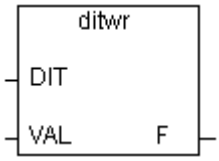
Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address to be read
VAL	OUT	Integer	The 16-bit value read from the DIT Register

Description:

Use this function to read an integer from a local DIT register.

This function cannot be used to access the extended DIT Range

12.6.1.2. DITWR - Write a value to a DIT register



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address to be written
VAL	IN	Integer	The 16-bit value to write to the DIT Register
F	OUT	Boolean	“True” if write is successful

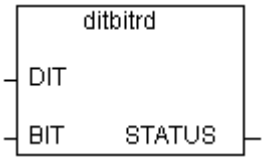
Description:

Use this function to write a 16 bit integer to a local DIT register.

The integer must be in the range –32768 to 32767

This function cannot be used to access the extended DIT Range. (Use the MxDIT_CPY function to copy extended DIT’s to/from an unused local DIT area).

12.6.1.3. DITBITRD - Read a bit from a DIT



Arguments:

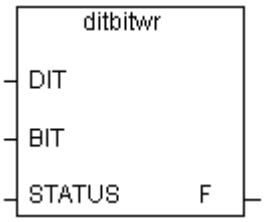
Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address to be read
BIT	IN	Integer	The BIT of the DIT register to be read. BIT must be in the range 0 to 15. BIT 0 is the least significant bit. BIT 15 is the most significant bit.
STATUS	OUT	Boolean	The Bit value read from the DIT Register

Description:

Use this function to read a BIT from a local DIT register as a Boolean.

This function cannot be used to access the extended DIT Range. (Use the MxDIT_CPY function to copy extended DIT’s to/from an unused local DIT area).

12.6.1.4. DITBTWR - Write a bit in a DIT



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address to be written to
BIT	IN	Integer	The BIT of the DIT register to be written to. BIT must be in the range 0 to 15. BIT 0 is the least significant bit. BIT 15 is the most significant bit.
STATUS	IN	Boolean	The Bit value to be written
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a Boolean value to a BIT in a local DIT register.

This function cannot be used to access the extended DIT Range (Use the MxDIT_CPY function to copy extended DIT’s to/from an unused local DIT area).

12.6.1.5. DITRDD - Read a double word from the DIT

Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address from which to read the double word.
DVAL	OUT	Integer	The 32-bit double word read from the register 'DIT' and 'DIT +1'

Description:

Use this function to read a double word from a local DIT register.

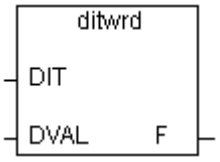
NOTE: the word order for a double word read from DIT register *n* is as follows:

DIT *n* = Most significant word

DIT *n* +1 = Least significant word

This function cannot be used to access the extended DIT Range.

12.6.1.6. DITWRD - Write a double word to the DIT



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address at which to write the double word.
DVAL	IN	Integer	The 32-bit double word to be written to register 'DIT' and 'DIT +1'
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a double word to a local DIT register.

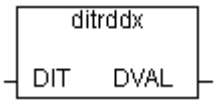
NOTE: the word order in the double word written to DIT register *n* is as follows:

DIT *n* = Most significant word

DIT *n* +1 = Least significant word

This function cannot be used to access the extended DIT Range.

12.6.1.7. DITRDDX - Read a double word from the DIT, swapped



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address from which to read the double word.
DVAL	OUT	Integer	The 32-bit double word read, in reverse word order, from the register 'DIT' and 'DIT +1'

Description:

Use this function to read a double word from a local DIT register.

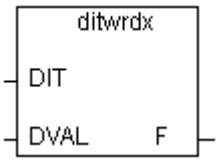
NOTE: the word order for a double word read from DIT register *n* is as follows:

DIT *n* = Least significant word

DIT *n* +1 = Most significant word

This function cannot be used to access the extended DIT Range.

12.6.1.8. DITWRDX - Write a double word to the DIT, swapped



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address at which to write the double word.
DVAL	IN	Integer	The 32-bit double word to be written, in reverse word order, to register 'DIT' and 'DIT +1'

F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a double word to a local DIT register.

NOTE: the word order in the double word written to DIT register *n* is as follows:

DIT *n* = Least significant word

DIT *n* +1 = Most significant word

This function cannot be used to access the extended DIT Range.

12.6.1.9. DITRDR - Read a real number from the DIT

Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address from which to read the real number.
RVAL	OUT	Real	The 32-bit IEEE floating point real number read from register 'DIT' and 'DIT +1'

Description:

Use this function to read a 32-bit IEEE floating point real number from a local DIT register.

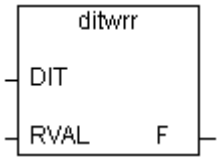
NOTE: the word order for a real number read from DIT register *n* is as follows:

DIT *n* = Most significant word

DIT *n* +1 = Least significant word

This function cannot be used to access the extended DIT Range.

12.6.1.10. DITWRR - Write a real number to the DIT



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address at which to write the real number.
RVAL	IN	Real	The 32-bit IEEE floating point real number to be written to register 'DIT' and 'DIT +1'
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a 32-bit IEEE floating point real number to a local DIT register.

NOTE: the word order in the real number written to DIT register *n* is as follows:

DIT *n* = Most significant word

DIT *n* +1 = Least significant word

This function cannot be used to access the extended DIT Range.

12.6.1.11. DITRDRX - Read a real number from the DIT, swapped

Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address from which to read the real number.
RVAL	OUT	Real	The 32-bit IEEE floating point real number read, in reverse word order, from the register 'DIT' and 'DIT +1'

Description:

Use this function to read a 32-bit IEEE floating point real number from a local DIT register.

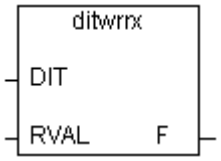
NOTE: the word order for a real number read from DIT register *n* is as follows:

DIT *n* = Least significant word

DIT *n* +1 = Most significant word

This function cannot be used to access the extended DIT Range.

12.6.1.12. DITWRRX - Write a real number to the DIT, swapped



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address at which to write the double word.
RVAL	IN	Real	The 32-bit IEEE floating point real number to be written, in reverse word order, to register 'DIT' and 'DIT +1'
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a 32-bit IEEE floating point real number to a local DIT register.

NOTE: the word order in the real number written to DIT register *n* is as follows:

DIT *n* = Least significant word

DIT *n* +1 = Most significant word

This function cannot be used to access the extended DIT Range.

12.6.1.13. DITRDT - Read a timer from the DIT

Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address from which to read the timer.
TVAL	OUT	Timer	The timer value read from the register 'DIT' and 'DIT +1'

Description:

Use this function to read a timer from a local DIT register.

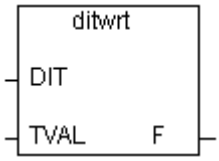
NOTE: the word order for a timer read from DIT register *n* is as follows:

DIT *n* = Most significant word

DIT *n* +1 = Least significant word

This function cannot be used to access the extended DIT Range.

12.6.1.14. DITWRT - Write a timer to the DIT



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address at which to write the timer.
TVAL	IN	Timer	The timer to be written to register 'DIT' and 'DIT +1'
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a timer to a local DIT register.

NOTE: the word order in the timer written to DIT register *n* is as follows:

DIT *n* = Most significant word

DIT *n* +1 = Least significant word

This function cannot be used to access the extended DIT Range.

12.6.1.15. DITRDM - Read a message from the DIT

Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address from which the message will be read.

LEN	IN	Integer	The length of the message string or number of characters.
MESS	OUT	Message	Message read from the DIT table starting at 'DIT'.

Description:

Use this function to read a message from the local DIT registers. Each DIT contains two characters. Messages are read from the DIT in increasing DIT order. For example, the message 'user', stored at DIT *n* would be read from DIT *n* and DIT *n*+1 as follows:

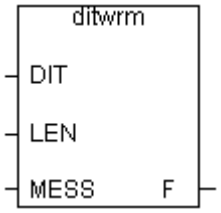
DIT *n* = 'u' (msb), 's' (lsb)

DIT *n*+1 = 'e' (msb), 'r' (lsb)

Always ensure that the variable connected to MESS is greater than or equal to LEN.

This function cannot be used to access the extended DIT Range.

12.6.1.16. DITWRM - Write a message to the DIT



Arguments:

Name	In/Out	Type	Description
DIT	IN	Integer	The DIT Register Address to which the message will be written.
LEN	IN	Integer	The length of the message string or number of characters.
MESS	IN	Message	Message written to the DIT table starting at 'DIT'.
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a message to the local DIT registers. Each DIT contains two characters. Messages are to the DIT in increasing DIT order. For example, the message 'user', stored at DIT *n* would be written to DIT *n* and DIT *n*+1 as follows:

DIT *n* = 'u' (msb), 's' (lsb)

DIT *n*+1 = 'e' (msb), 'r' (lsb)

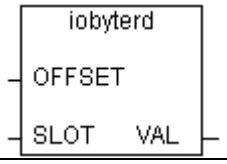
The number of DIT registers that will be written can be calculated as follows:

If LEN is an even number then the number of DIT’s written equals LEN/2

If LEN is an odd number then the number of DIT’s written equals LEN/2+1

This function cannot be used to access the extended DIT Range.

12.6.1.17. IOBYTERD - Read a byte from an I/O Module



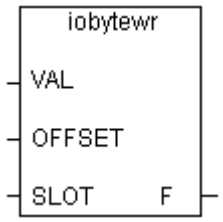
Arguments:

Name	In/Out	Type	Description
OFFSET	IN	Integer	Byte offset in the memory map of the module.
SLOT	IN	Integer	I/O slot where the module is installed.
VAL	OUT	Integer	Value of byte (0 - 255)

Description:

Use this function to read a byte of memory directly from an I/O module. This is an advanced function and is available for use with specialised I/O modules. In most applications, this function is not necessary.

12.6.1.18. IOBYTEWR - Write a byte to an I/O Module



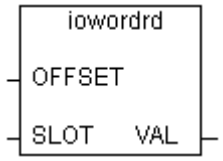
Arguments:

Name	In/Out	Type	Description
VAL	IN	Integer	Value of byte (0 - 255)
OFFSET	IN	Integer	Byte offset in the memory map of the module.
SLOT	IN	Integer	I/O slot where the module is installed.
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a byte of memory directly to an I/O module. This is an advanced function and is available for use with specialised I/O modules. In most applications, this function is not necessary.

12.6.1.19. IOWORDRD - Read a word from an I/O Module



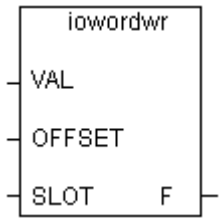
Arguments:

Name	In/Out	Type	Description
OFFSET	IN	Integer	Word offset in the memory map of the module.
SLOT	IN	Integer	I/O slot where the module is installed.
VAL	OUT	Integer	Value of word (0 - 65535)

Description:

Use this function to read a word of memory directly from an I/O module. This is an advanced function and is available for use with specialised I/O modules. In most applications, this function is not necessary.

12.6.1.20. IOWORDWR - Write a word to an I/O Module



Arguments:

Name	In/Out	Type	Description
VAL	IN	Integer	Value of word (0 - 65535)
OFFSET	IN	Integer	Word offset in the memory map of the module.
SLOT	IN	Integer	I/O slot where the module is installed.
F	OUT	Boolean	“True” if write is successful

Description:

Use this function to write a word of memory directly to an I/O module. This is an advanced function and is available for use with specialised I/O modules. In most applications, this function is not necessary.

12.6.2 Using “DIT” Virtual I/O Modules to access data in the DIT

In the library of Maxiflex I/O Modules are a number of “virtual” modules that do not represent actual Maxiflex Modules, but do represent data in the DIT as if it were an actual module.

These virtual Modules can be used to read/write data to/from the DIT just as if it was data from the outside world.

As I/O modules these inputs/outputs will be updated once per scan.

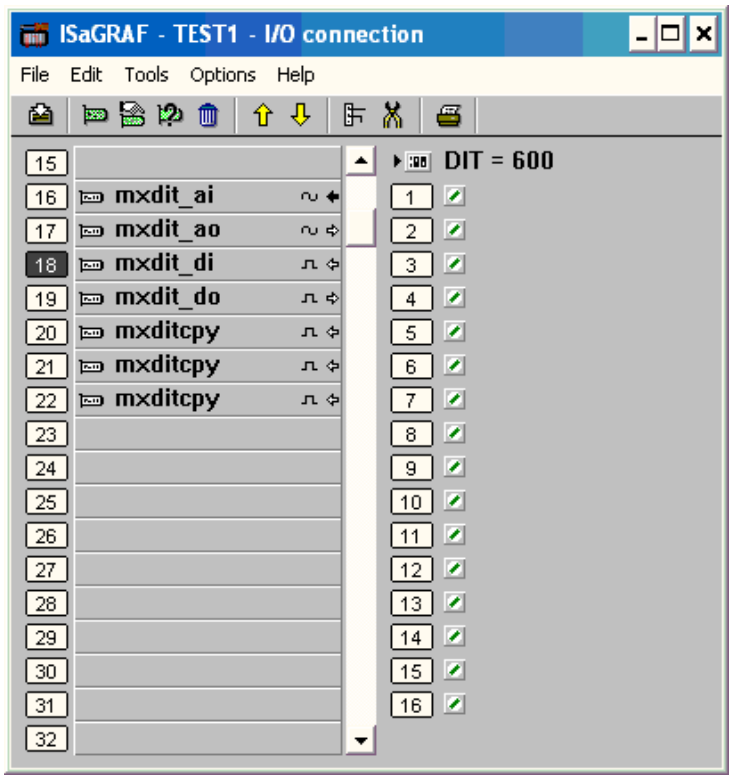
To use these modules, assign them to an I/O slot in the normal way. I/O slots 64 to 255 can be used for this purpose without encroaching on the real I/O Slots available in the system, although any I/O slot can be used for a virtual module.

Set the DIT parameter in the module to the DIT(s) to be addressed.

See the Descriptive Note attached to these modules for more detailed information.

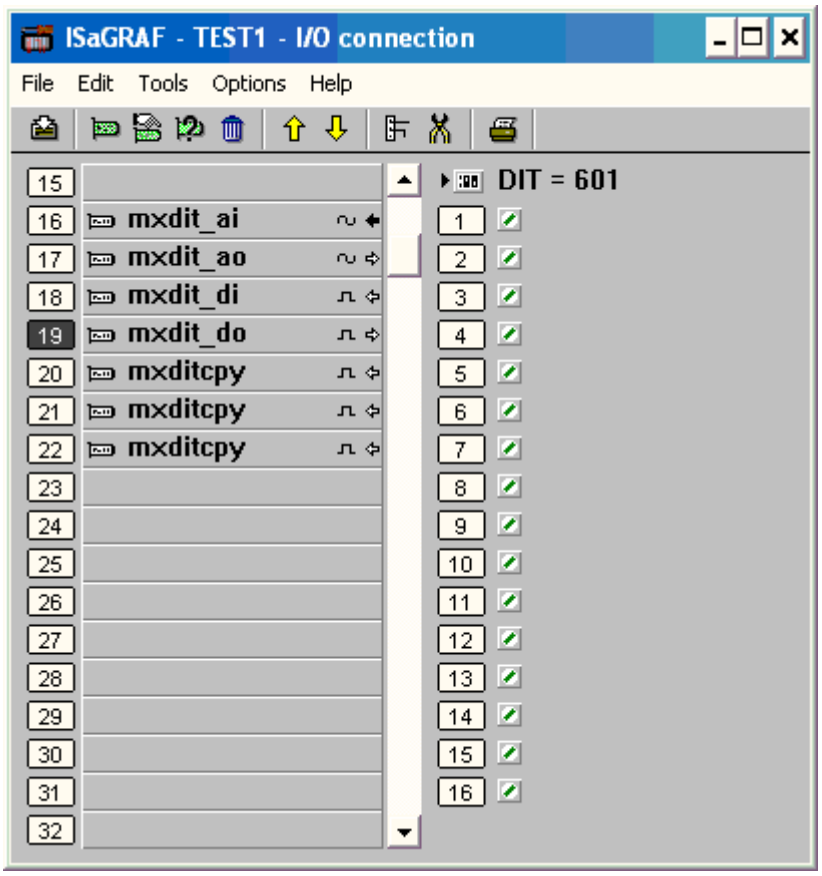
12.6.2.1. MxDIT_DI

Use this Virtual I/O Module to read the 16 bits of a single local DIT register as 16 Boolean inputs.



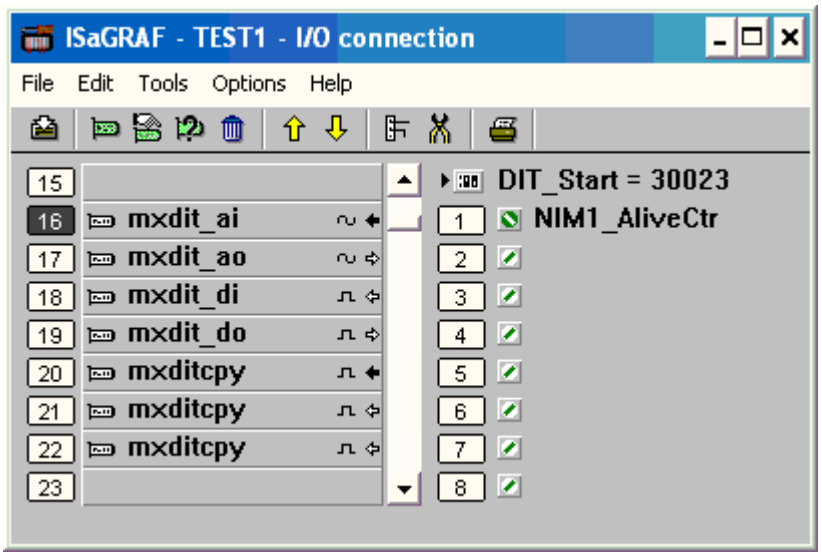
12.6.2.2. MxDIT_DO

Use this Virtual I/O Module to write 16 Boolean outputs to the individual bits of a single local DIT register.



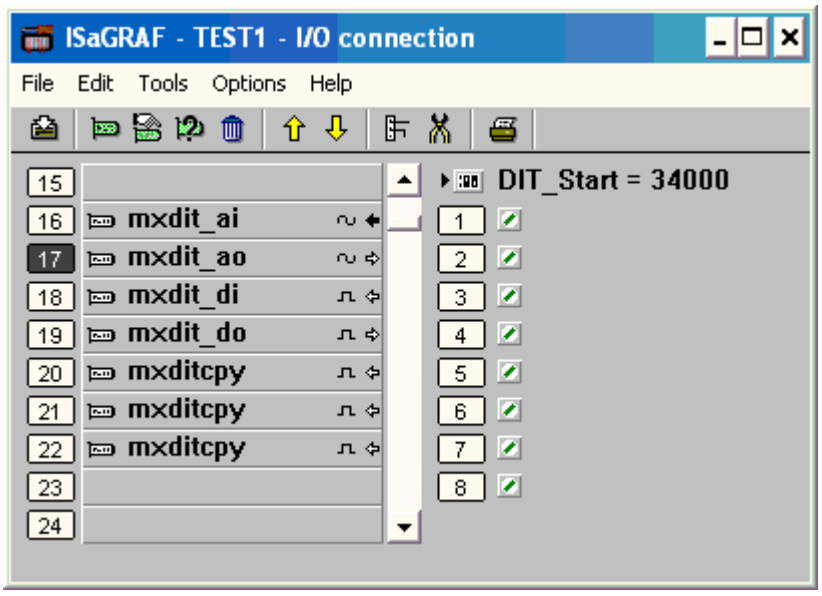
12.6.2.3. MxDIT_AI

Use this Virtual I/O Module to read 8 integers from a contiguous block of 8 local DIT Registers.



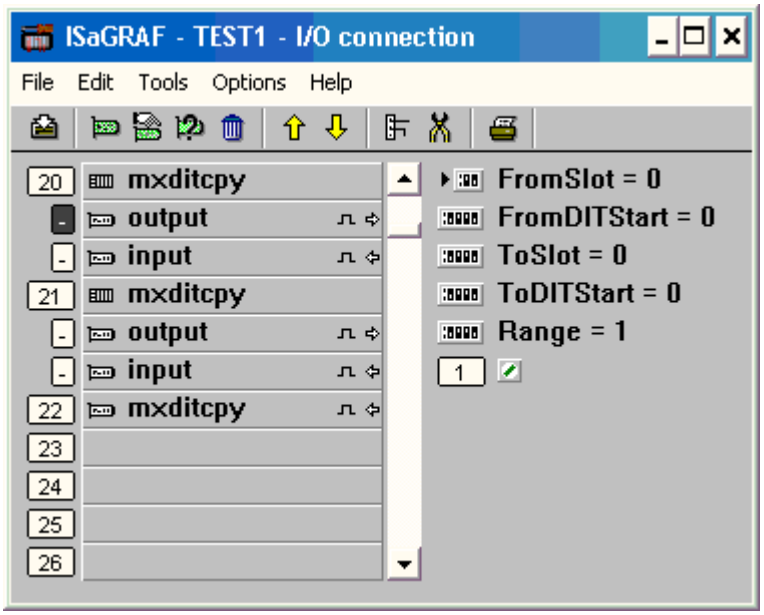
12.6.2.4. MxDIT_AO

Use this Virtual I/O Module to write 8 integer values to a contiguous block of 8 local DIT registers.



12.6.2.5. MxDITCPY

Use this “Complex Equipment” Virtual I/O Module to copy a block of DIT registers from one place to another. This module can be used to copy blocks of DIT registers to/from the slower DIT extended address space from/to the local CPU address space, from where it can be accessed by the program using the DIT read and write function blocks (see section 12.6.1).



12.7 Maxiflex P3 CPU Specific Function Blocks

In addition to the standard library of functions and function blocks available in the Application Workbench, some additional Function Blocks (such as Auto-tuning PID control blocks) are available for the Maxiflex P3 CPU.

Details of these function blocks can be found in the separate Function Block Application Notes distributed with these function blocks. See your Omniflex distributor for availability.

13. Constructing a Redundant Controller System

13.1 Principles of Dual Redundancy

Availability of a system is defined in terms of the Mean Time To Failure (MTTF) of individual components and the Mean Time to Repair(MTTR) of these components (MTBF = MTTF+MTTR). As systems become more complex, the MTBF of these systems decreases due to the increasing number of components. The principle of “dual redundancy” is to duplicate those components identified as “critical” to the system, such that if any one of those duplicated components were to fail, then the system would continue to operate uninterrupted, (but with an alert of the failure).

“Critical” components are considered those components whose operation is central to a significant part of the system, and/or whose MTBF is considered low in relation to other components in the system.

There are number of requirements to be met before dual redundancy will provide the theoretically promised system availability:

1. All failures of redundant components must be self-revealing.
2. Any failure of a redundant component must not be able affect the operation the remaining system.

13.2 Functional Safety

Dual Redundant systems are not necessarily safety-related systems. In order for a control system to be considered for use in a “safety related” role, the system, its design and application must comply with the requirements of a standard such as IEC61508.

In some sense, high availability and Functional Safety are related, in that both concepts are concerned with the detection of failures, and the increase of system MTBF. If the system could be guaranteed to be 100% reliable, then the issue of failure to a safe state would not be applicable.

But in another sense, high availability and Functional Safety are opposing concepts. Safety-related systems are concerned with the safety of the process, and will attempt to put the process into a safe state as an overriding requirement, with no regard for system availability and therefore productivity. High Availability on the other hand is concerned with keeping the equipment and therefore plant running in the face of component failure.

Safety-related systems have to meet defined requirements for safety as specified in standards such as IEC61508, and require the analysis of every failure mode of every component in the system into safe and dangerous failures. Safety-related systems are concerned with the detection of every possible failure that can occur in the safety related parts of the system.

High Availability systems, on the other hand, are concerned mainly with the detection of failure of the redundant parts of the system only, and are not concerned with the categorisation of these failures as “safe” or “dangerous”.

High Availability systems are only concerned with certain component failures, and their affect on the performance of the system, in an attempt to increase the combined system MTBF to acceptable levels.

IMPORTANT NOTICE REGARDING SAFETY

The Maxiflex P3-R CPU's are designed for High Availability applications. Although every attempt is made to design a system with maximum diagnostic fault coverage, this system is not designed for use in "safety related systems" as defined in IEC61508 or equivalent.

Omniflex makes no claim regarding the ability of the system to fail to a safe state, and cannot be held liable for consequential loss resulting from the failure of any component or system supplied by Omniflex.

13.3 Redundant System Architecture Overview

The P3-R Redundant Controller is designed for applications where dual redundant CPU's are required connected to common Inputs and Outputs.

The dual redundant network connection to the supervisory system is transparent to the supervisory system.

This architecture is shown in the following typical system diagram:

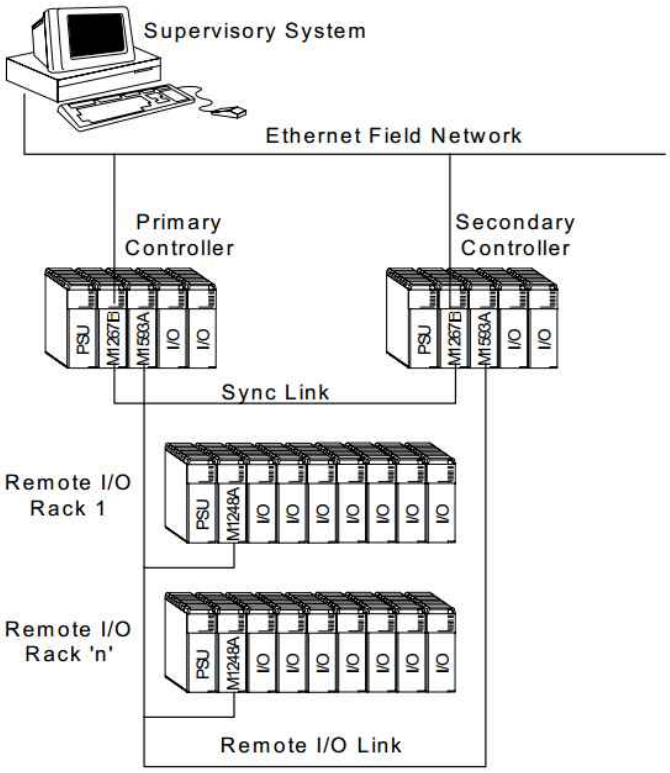


Figure 13.1: Dual Redundant Controller System Architecture

This Dual Redundant Architecture has the following features:

- Two identical Maxiflex "P3-R" CPU's are configured to create a primary and secondary controller pair acting as a single redundant Master Controller.
- Each P3-R CPU is contained on its own master base with power supply and communications capability, eliminating all common points of failure between the two.

- The Primary and Secondary controllers communicate with each other through a dedicated high speed communications channel (called the Sync. Link), used for program synchronisation and health status reporting.
- Each controller is connected to the I/O through an independent high speed remote I/O network.
- The remote I/O network connected to the Primary and Secondary controllers communicate with a Remote I/O processor on each remote I/O base.
- Each Remote I/O Processor can control up to 15 conventional I/O modules.
- Up to 3 Remote I/O bases can be connected to the Remote I/O network giving a total of 45 I/O modules per redundant controller system (up to 1440 I/O using 32 channel I/O modules).
- Both controllers read the same input information from the common I/O via the Remote I/O processors on each remote base in the system.
- Only the “active” controller writes data to the Outputs.
- When the primary controller is no longer able to control due to any internally detected fault, then the secondary controller becomes active and takes over all tasks from the Primary Controller including writing to the outputs.

13.4 Operating Modes of the Dual Redundant System

13.4.1 The Primary Controller

The Primary Controller operates in three distinct modes i.e. Active, Standby and Failure mode.

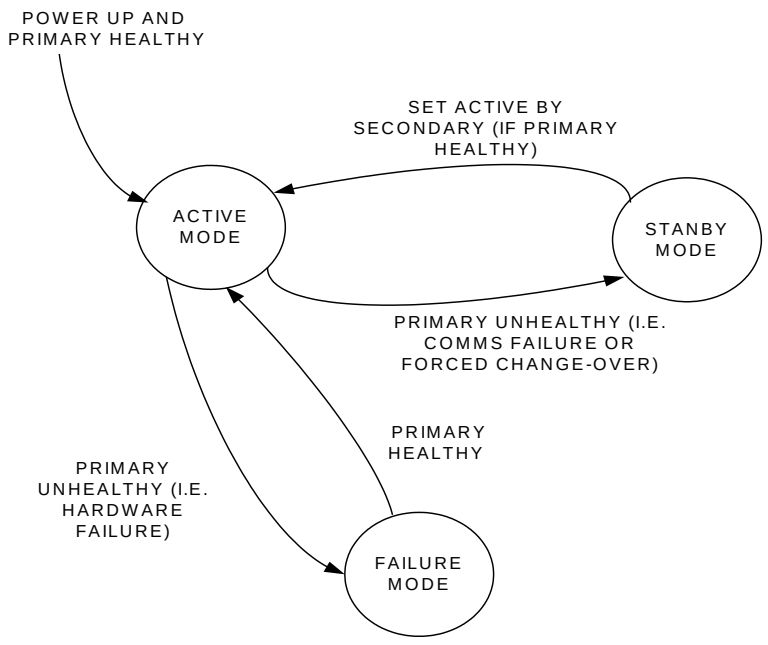


Figure 13.2: Operating Modes of Primary Controller

The following able shows the actions performed by the Primary Controller during its different modes of operation.

Mode	Actions
Active	• Respond to messages addressed to the Redundant Controller Node Address • Execute User program • Read Inputs from Remote I/O Rack • Write Outputs to Remote I/O Rack • Check Health status • Check for change-over instruction to Standby from Secondary Controller • Read Health Status from Secondary Controller • Check for the Force Change-over command
Standby	• Do NOT Respond to messages addressed to Controller Node Address. • Check Health status and make it available to Secondary Controller • Check for change-over instruction to Active from Secondary Controller • Read Health Status from Secondary Controller
Failure	• This mode of operation is a hardware failure and the Controller is not operable therefore no action is expected.

Table 13.1: Action List by Operating Mode – Primary Controller

13.4.2 The Secondary Controller

The Secondary operates in a similar three modes to the primary controller.

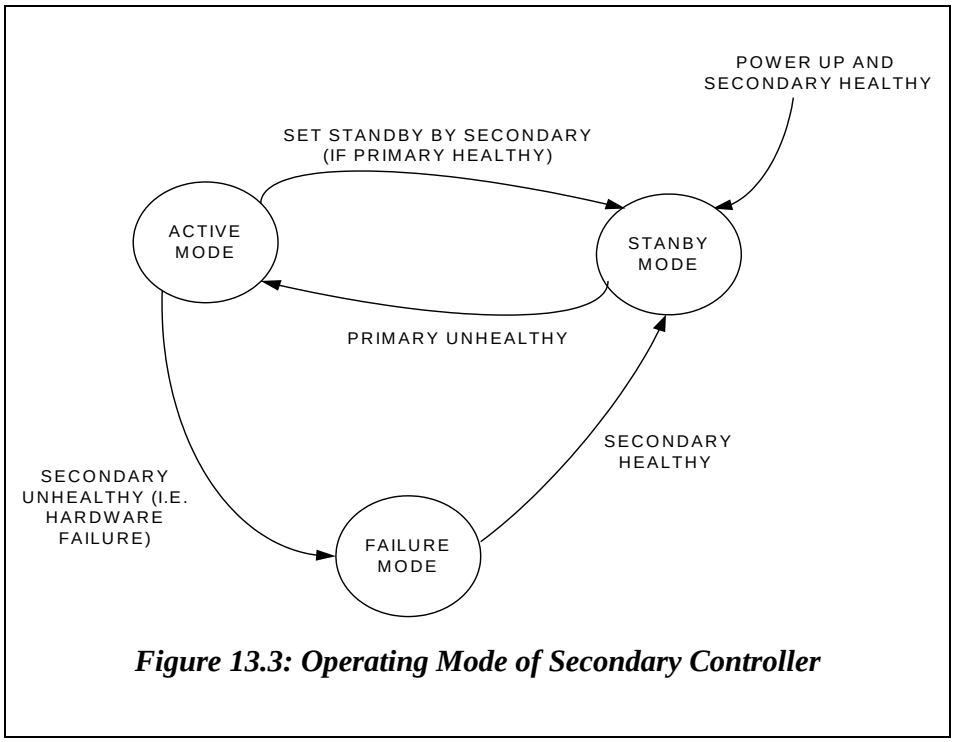


Figure 13.3: Operating Mode of Secondary Controller

Table 2 below shows the actions performed by the Secondary Controllers during its various operating modes.

Mode	Actions
Active	• Respond to messages addressed to Redundant Controller Node Address. • Execute User program • Read Inputs from Remote I/O Rack • Write Outputs to Remote I/O Rack • Read Health Status from Primary Controller • Force Primary to change over to Active Mode if Primary is healthy • Force Secondary to change over to Standby if Primary is healthy
Standby	• Do NOT Respond to messages addressed to Controller Node Address • Read Sync Blocks from Primary Controller • Force Primary to change over to Standby Mode if Primary is unhealthy • Force Secondary to change over to Active if Primary is unhealthy
Failure	• This mode of operation is a hardware failure and the Controller is not operable therefore no action is expected.

Table 13.2: Action List by Operating Mode – Secondary Controller

13.4.3 Controller Health Status

The Health status of the Primary and Secondary Controllers is a key element to the redundant operation of the system. It is this parameter that determines which controller operates in the Active mode.

The health status is dependant upon the following three factors:

13.4.3.1. Status of Supervisory Communications

A controller must always be able to intelligently receive messages from the Supervisory System, irrespective of whether it is operating in the Active mode or Standby mode. The Active controller will respond to these messages, and the Standby controller will not.

13.4.3.2. Status of Controller Hardware

Upon detection of a hardware failure of the primary controller, either by the primary controller itself, or by the secondary controller, the secondary controller will assume control (as long as it is healthy). When the health of the primary controller is restored, control will be automatically restored to the primary controller.

13.4.3.3. Forced change-over by the User (Primary only)

From time to time you may wish to test the change-over ability of the system from primary to secondary control. This is accomplished by setting a “Force change-over” flag in the Primary Controller while in the Active state. This can be done from the supervisory system.

Clearing this flag will cause the primary to resume control.

14. Steps to Set Up a Redundant Controller System

14.1 Step 1: Assemble Hardware

The hardware setup for a redundant CPU system with Common I/O is shown in Figure 13.1.

Each redundant CPU must be installed on its own master base with power supply.

Slot 1 of each redundant base must be equipped with an M1593A RIO NIM.

Connect the Remote I/O Link to the primary and secondary controllers and to the remote I/O racks.

14.2 Step 2: Set Remote I/O DIP Switch Addresses

Set the following address switches for the modules. These switches are located inside the doors of each module:

MODULE	ADDRESS	Switch Setting (0=off; 1=on) 12345678
Primary Controller M1593A RIO NIM	1	10000000
Secondary Controller M1593A RIO NIM	2	01000000
Remote I/O Rack 1: M1248A R2c	3	11000000
Remote I/O Rack 2: M1248A R2c	4	00100000
Remote I/O Rack 3: M1248A R2c	5	10100000

Table 14.1: Remote I/O Module Address Switch Settings

The Primary Controller identifies itself by the Address set on the RIO NIM. If the RIO NIM address is set to 1, then the P3 CPU will act as the primary controller. If the RIO NIM address is set to 2, then the P3 CPU will act as the Secondary Controller.

If less than 3 remote I/O Bases are to be used, then omit higher Base addresses.

14.3 Step 3: Power up the Primary Controller and Remote I/O Racks

Power up the Primary Controller and Remote I/O racks and check that the Primary Controller P3 CPU, RIO NIM and R2c Remote I/O Scanner modules have their CPU OK/NIM OK LED's on steady.

Plug in the Sync Link cable, but leave the Secondary Controller un-powered at this stage.

The I/O LED on the Primary Controller may flash at this point because the I/O Module List has not been configured.

The Battery light on the Primary Controller P3 CPU should be off. If the Battery Light is ON, check that the Battery Disconnect Tab has been removed, and that a battery is fitted to the CPU.

14.4 Step 4: Check Remote I/O Activity

Observe the LED activity on the Primary Controller RIO NIM and the R2c CPU's in the remote racks. The Token LED should be flashing steadily and there should be activity on the Tx and Rx LED's at least once every second.

14.5 Step 5: Configure Primary Controller

Plug a programming cable into the primary controller and configure the Primary Controller and Remote I/O using Omniset, as if it were the only controller in the system. See chapter 11 for full instructions on configuring the primary controller CPU.

If you require any user data DIT to be synchronised between primary and secondary controllers, then you can organise this data in the Data Interchange Table (DIT) into up to three blocks of any length. You then configure the address and size of these Sync Blocks in the primary controller as shown below.

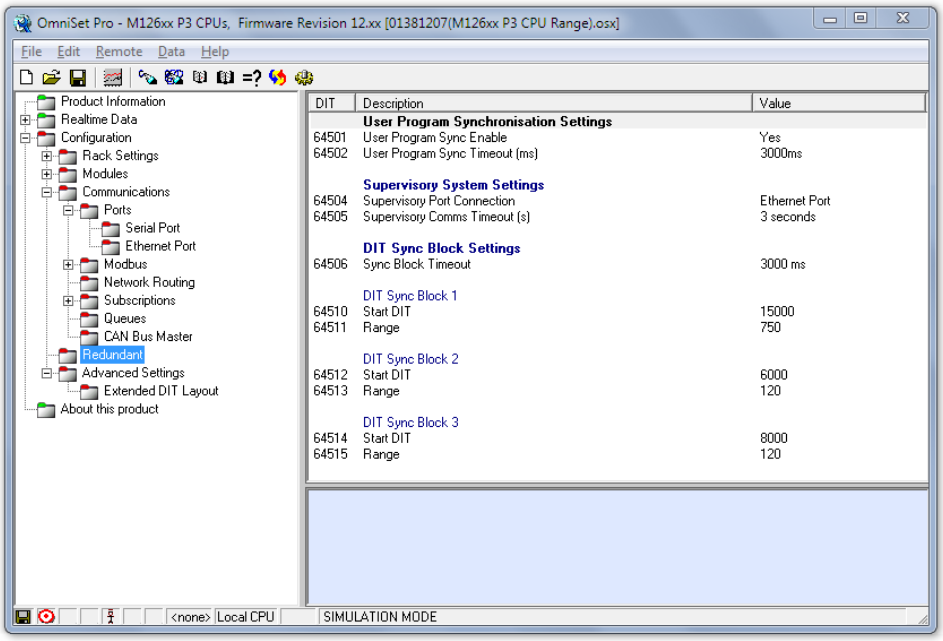


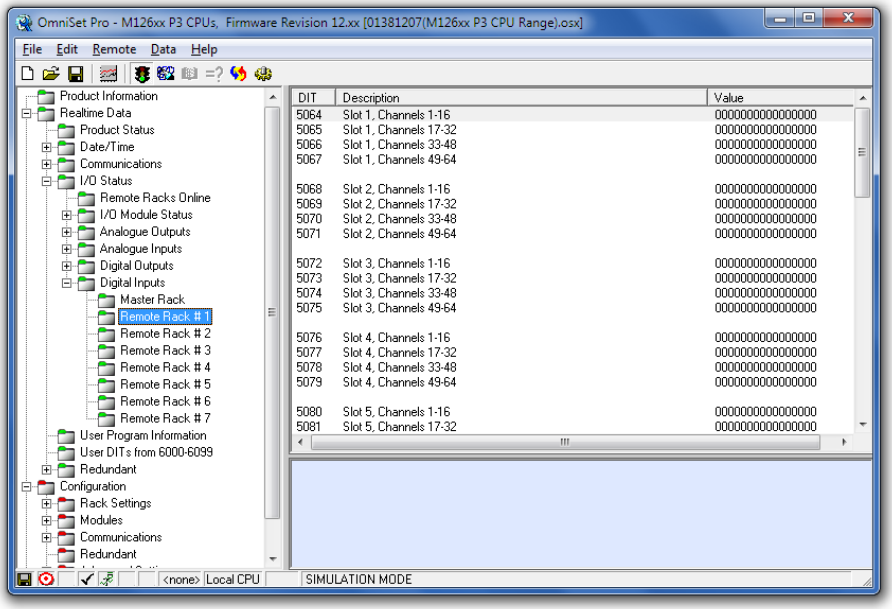
Figure 14.1: Setting the Sync Parameters in Omniset

The system change-over time settings are also in this configuration group. Change these timers from their default settings if required.

Once the Primary Controller is fully configured, the RUN LED should be on steady indicating that it is the controller in control.

14.6 Step 6: Test I/O Access from Primary Controller

From Omniset, view the status of I/O from every I/O rack ensuring that the status is properly indicated in the relevant Omniset group



14.7 Step 7: Save the Primary Controller Configuration

When you are satisfied that the Primary Controller is fully functional as required, then save the Configuration to your hard drive giving the configuration file a name that you will remember.

14.8 Step 8: Power up the Secondary Controller

Ensure that the Sync Link cable is installed, and then power up the secondary controller.

The RUN LED on will be OFF if there is problem with the secondary controller installation or configuration, or FLASHING to indicate that the Secondary Controller is in standby.

14.9 Step 9: Check for Sync Link Activity

The “SYNC 100” LED’s should be ON on both controllers, and their “SYNC LINK” LED’s should be on, flashing OFF periodically to indicate activity on the Sync Link.

14.10 Step 10: Check Remote I/O Activity on Secondary Controller

Check that the Token LED on the RIO NIM on the secondary controller Rack is flashing at a fast steady rate, and that the Tx and Rx LED’s are flashing periodically (at least once per second).

14.11 Step 11: Configure Secondary Controller

Now remove the Programming cable from the Primary Controller’s Programming Port and plug it into the Secondary Controller Programming Port.

Download the identical configuration to the Secondary Controller that is in the Primary Controller, by performing a “Write All” function in Omniset.

14.12 Step 12: Test I/O Access from Secondary Controller

Check the I/O Status in the secondary controller as you did in step 14.6 for the primary controller.

14.13 Step 13: Check Active and Standby Status of the Redundant System

At this stage, the Primary Controller should be active – indicated by the RUN LED on the Primary CPU on steady, and the Secondary Controller should be in Standby, indicated by the RUN LED on the secondary controller flashing.

14.14 Step 14: Test Fail Change-over

You can test the fail change-over from primary to secondary controller in a number of ways:

1. Power down the Primary Controller, OR
2. Unplug the Remote I/O cable from the Primary Controller's RIO NIM.

You can tell always tell which controller is active, because its RUN LED will be on steady. The RUN LED on the standby controller will be OFF if the controller is not ready to run, or flashing if it is in standby – ready to run.

NOTE: Unplugging the Sync Link cable will not cause a change over from primary to secondary.

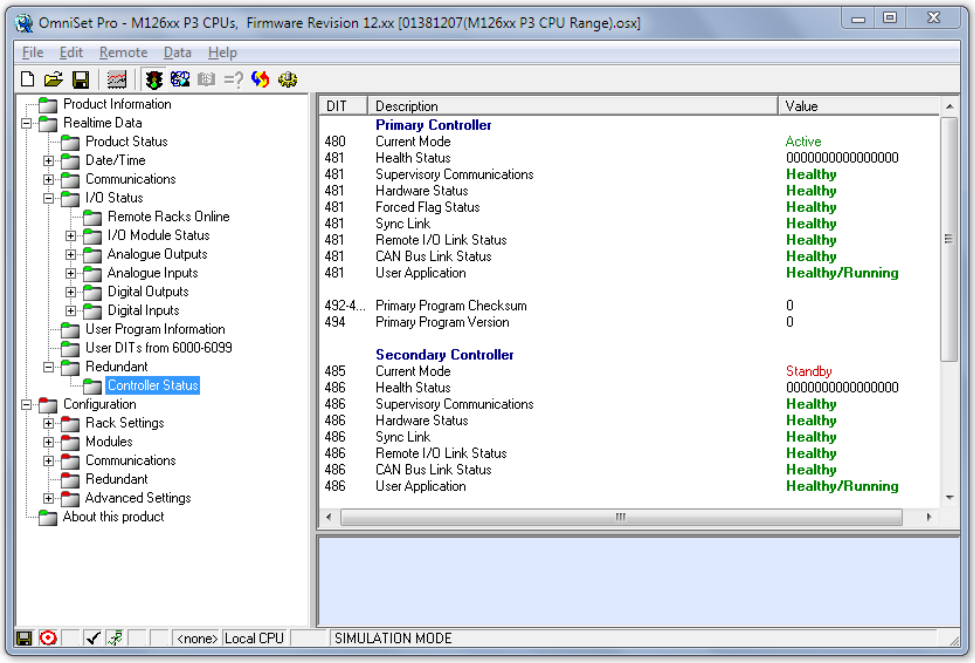
14.15 Step 15: Test Force Change-over

It is possible to force a changeover from software. This allows changeover to be tested externally without interfering with any of the hardware.

This Force function can be invoked through any of the communications ports. This will allow this test function to be included as part of a Supervisory Computer design if necessary.

To test this function using Omniset follow this procedure:

1. Plug the programming cable into the Primary Controller's Prog Port
2. Confirm that the Primary is Active, and the Secondary is in Standby:



3. Change "Force Primary Failure" from False to True.

4. Primary should change to “Standby” and Secondary to “Active”
5. Unplug the Programming Cable from the Primary, and plug it into the Secondary. This is because you are now no longer able to control the Primary changeover, because the Secondary is in control.
6. Change the “Force Primary Failure” setting from True to False.
7. The Primary should once again resume control, and the Secondary will go to “Standby”.

NOTE: If you leave the Force flag set, it will time out after about five minutes and automatically clear. If the primary is in standby, then it will resume control from the secondary controller.

15. Maintenance

15.1 Battery Type

The P3 CPU is equipped with an internal plug-in Lithium Battery.

This battery is a type CR2032 Lithium Battery. This battery can be obtained from Omniflex by specifying Part Number 3.5701.001

The Battery is used only during power outages to maintain the real-time clock, and (optionally) the internal RAM memory.

It is recommended that the battery be replaced at least every three years, or when the Red “BATT” Battery Low indicator on the front of the CPU lights.

15.2 Battery Replacement Procedure

To replace the internal battery proceed as follows:

1. Turn off power from the Maxiflex Base.
2. Remove the CPU from the Base
3. Unscrew the top Vent Cover using a No.0 Pozidrive screwdriver of diameter 3.2mm
4. Lift the front of the Vent Cover slightly and then slide carefully forward to disengage the rear clip.
5. Remove the battery using a small screw driver to lever the clip open to release the old battery.
6. Insert the new battery ensuring that it is properly seated in the holder.
7. Replace the Vent cover.
8. Replace the CPU on the Maxiflex base and restore the power.
9. Check that the BATT indicator on the front of the module is off.