USER'S MANUAL

Maxiflex Dual Serial NIM
M1580A
User's Manual

DATE	REVISION	COMMENTS
August 2004	1	Initial Publication.
February 2005	2	Updated Modbus Master explanation

SOFTWARE COPY AVAILABLE

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

The pdf file is named UM1580AR02.pdf

SCOPE

This User Manual provides information on how to install, configure and use the Maxiflex Dual Serial NIM module.

This manual does not cover the fundamentals of the EziForth programming language.

This manual covers the following product Models:

Model	Description
M1580A	NIM with Dual Serial RS232/485 Serial Ports

Introduction

The MAXIFLEX Dual Serial NIM is designed specifically for industrial telemetry and remote I/O applications in conjunction with the M124x, M125x, M126x range of CPUs (i.e. T2, T3 and P3 CPUs respectively), combining powerful industrial network communications features with ease of use. In addition, the NIM module is programmable (using the EziForth programming language) allowing applications to be run on the NIM device completely separate from the Maxiflex CPU.

Following the ISO OSI 7-layer model, the NIM module, combined with the Maxiflex CPU, includes a powerful inter-network routing capability for communicating over multiple networks, in very large geographically spread installations. This capability allows many dissimilar network types to be linked to create a seamless factory “intranet”, quite often without the need to lay special network cabling.

In addition to the standard propriety Conet/s protocol, the Dual Serial NIM module also supports the MODBUS communications protocol (Master and Slave devices) as standard. Alternatively, if user’s require it, the serial ports can also be controlled by Forth program thereby allowing it to be programmed to communicate to any serial device.

Table of Contents

1. GENERAL ARRANGEMENT 7
1.1 LED Indicators: 7
1.2 Serial Ports: 7
1.3 Programming Port: 8
2. Installing the Dual Serial NIM 9
2.1 Installing the Dual Serial NIM on the Maxiflex base 9
2.2 Connecting the Serial ports: 9
2.3 Applying power for the first time 10
3. Dual Serial NIM and the MAXIFLEX Architecture 11
3.1 Dual Serial NIM Functions 11
3.1.1 FORTH User Program 11
3.1.2 Data Interchange Table 11
3.1.3 Subscription Service 11
3.1.4 Network Routing Service 11
3.1.5 Communications Status Monitoring 11
3.1.6 Serial Port Protocol Selection 11
3.1.7 Remote Programming 12
4. Configuring a Dual Serial NIM 13
4.1 Using OMNISET or DITview to configure the Dual Serial NIM 13
4.2 Configuring the Serial Ports 13
4.2.1 MODBUS Slave Protocol 14
4.2.1.1. Default MODBUS (ASCII 9600 baud Slave) Mode 14
4.2.1.2. Standard MODBUS Slave Protocol Mode 14
4.2.2 MODBUS Master Protocol 15
4.2.2.1. Standard MODBUS Master Protocol Mode 15
4.2.3 Conet/s (Peer-to-peer) Protocol 15
4.2.3.1. Default Conet/s Mode (19200 baud) 15
4.2.3.2. Standard Conet/s Protocol Mode 16
4.2.4 Custom User Protocol Mode 16
4.3 Addressing a NIM Module through the Programming Port of the Maxiflex CPU 17
4.3.1 No Address set 17
4.3.2 Local CPU 18
4.3.3 Intelligent Module in I/O Slot 18
4.3.4 Local Network ID (using Offset) 18
4.3.5 Direct/Global Network ID 18
4.4 Programming the Dual Serial NIM 18
5. The Data Interchange Table explained 19
5.1 DIT Table Layout 19

6. The Subscription Service Explained 21
6.1 Introduction to Subscriptions 21
6.2 Setting up subscriptions 21
6.3 Number of subscriptions allowed 23
6.4 Subscription Application Example 23
7. Modbus Master Operation Explained 25
7.1 Introduction to Modbus Master Driver 25
7.2 Modbus Master Parameters 25
7.3 Query Configuration 27
7.4 Query Triggers 28
7.5 Status DIT Registers 29
Table 7.2 Modbus Master Query Error Codes 29
8. Appendix: Dual Serial NIM Detailed DIT Layout: 30
8.1 Dual Serial NIM Dynamic Data Area DIT Register Assignment 30
8.2 Dual Serial NIM Configuration Data Area DIT Register Assignment 33

1. GENERAL ARRANGEMENT

1.1 LED Indicators:

LED Legend	LED Colour	Description
NIM OK	GREEN	ON - NIM is healthy OFF or Flashing – CPU Faulty or no power applied
RUN	GREEN	ON – User application software program is running. OFF – No User application software is running Flashing – Terminal interaction with CPU
PORT0 Rx	YELLOW	ON – data is being received on serial port0 OFF – serial port0 receiver is idle
PORT0 Tx	RED	ON – serial data is being transmitted on serial port0 OFF – serial port0 transmitter is idle
PORT1 Rx	YELLOW	ON – data is being received on serial port1 OFF – serial port1 receiver is idle
PORT1 Tx	RED	ON – serial data is being transmitted on serial port1 OFF – serial port1 transmitter is idle

1.2 Serial Ports:

The serial ports connect the Dual Serial NIM to other serial devices, depending on the application. Connection can be either RS232 or RS485.

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

a) Conet/s protocol (peer-to-peer) is available on both serial ports 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.

b) Modbus protocol (Master and Slave) is available on both ports allowing easy connection to other third party products such as DCS systems, SCADA systems and PLC's.

c) User Protocol. This setting gives full control of the serial ports to an application program running on the Dual Serial NIM. This allows custom protocols to be written in EZIFORTH and downloaded to the NIM for maximum flexibility. Applications such as creating output for a local printer, communicating with an operator display panel, or communicating with a third party device using a proprietary protocol, are possible.

1.3 Programming Port:

The Dual Serial NIM does not include a built-in programming port but is accessed via the programming port of the Maxiflex CPU module fitted on the same rack. Details on configuring the NIM module via the CPU programming port are provided later.

2. Installing the Dual Serial NIM

2.1 Installing the Dual Serial NIM on the Maxiflex base

Install the Dual Serial NIMs into the **I/O** position of a Maxiflex Base ONLY.

Refer to the Maxiflex bases General Instructions (PN 98-8952-930-XXX) for more detail on base layout, module insertion and module removal. Please refer to Figure 2.1 for the I/O Module position.

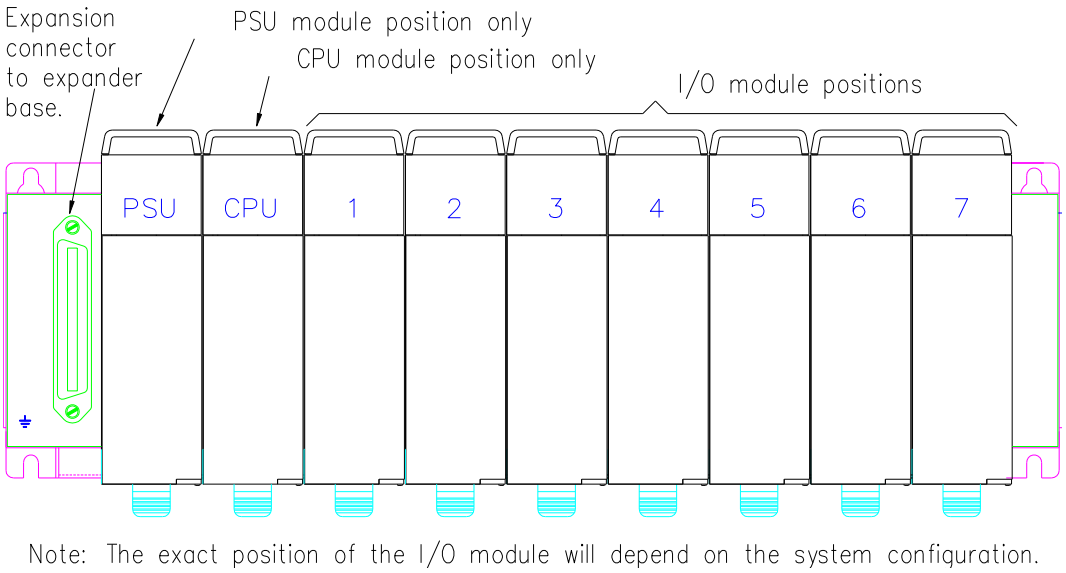


***Figure 2.1*: Layout of the 7I/O Master Base**

2.2 Connecting the Serial ports:

The M1580A DUAL SERIAL NIM allows jumperless conversion between RS232 and RS485 communications. By simply following the appropriate pinout of the serial port connector below it is possible to connect the NIM in either RS232 or RS485 mode without making any hardware or software changes to the NIM.

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 2.1: Pin allocation of serial port connectors on M1580A Dual Serial NIM.

2.3 Applying power for the first time

Once the Dual Serial NIM has been installed into a Maxiflex rack along with a Maxiflex CPU module, apply power and the NIM OK LED will turn ON and remain steady ON until power is removed. If the NIM OK LED remains OFF, then check the power connections to the PSU module and the LED indication of the PSU module. Refer to the Installation Guide of the PSU module to verify it's correct.

The RUN LED will only turn ON at first power up if this NIM was shipped with a user application set to begin operating at start-up. Normally, there is no user application installed and the LED will remain OFF.

3. Dual Serial NIM and the MAXIFLEX Architecture

3.1 Dual Serial NIM Functions

The Dual Serial NIM contains the following functions:

3.1.1 FORTH User Program

While not necessary in the majority of applications due to the rich set of other features in this Dual Serial NIM, the FORTH programming language can be used to customise any part of the system to suit the application, from simple data scaling, to complex data manipulation.

3.1.2 Data Interchange Table

The Data Interchange Table (or DIT) in the Dual Serial NIM is the crossroads for data. Any exchange of data between functions in the Dual Serial NIM and with the outside world takes place through the DIT. The DIT is an array of 16 bit registers accessible from any function or communications port in the system for the purpose of interchanging data.

3.1.3 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.

3.1.4 Network 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.

3.1.5 Communications Status Monitoring

The Data Interchange Table contains a group of communication status registers that may be used for diagnostic purposes. These registers monitor communications activity on each port. The local and global addresses for each port may also be accessed from this group of registers.

3.1.6 Serial Port Protocol Selection

The in-built serial port on the Dual Serial NIM comes equipped with three protocol options as standard:

1. Conet/s protocol can be selected to interconnect Maxiflex systems over wide areas using the RS232 port. This full duplex protocol provides full peer-to-peer communications capability to allow multiple local networks systems to be interconnected over a wide area into a single intranet. This efficient protocol retains the full capability of the Conet networking technology including remote programming and the ability to run data subscriptions in both directions simultaneously

2. Modbus Slave can be selected for easy interconnection of Maxiflex I/O to third party systems such as Distributed Control System's, SCADA software, or Master Programmable Logic Controllers.
3. Modbus Master can be selected for easy interconnection of third party modbus slave devices. Upto 32 Queries can be configured on each port.
4. User defined protocol. The serial ports can be used directly by a Forth application program to accomplish any serial communication required by the User.

3.1.7 Remote Programming

Every CPU is equipped with a dedicated programming port that is equipped with the a version of the Conet protocol. Using the network routing function and convenient table configuration, it is possible to configure/program every node in a Maxiflex intranet from a single programming port. This function significantly reduces system downtime and improves maintenance efficiency and therefore life-cycle costs.

4. Configuring a Dual Serial NIM

4.1 Using OMNISET or DITview to configure the Dual Serial NIM

Most of the features available on the Dual Serial NIM are easily accessible and can be setup with the minimum of effort by writing to Data Interchange Registers in the NIM, as well as dipswitch selection of the Dual Serial NIM hardware.

The best method to manually configure a Dual Serial NIM is to use either the Microsoft Windows95/98/NT/2000 compatible OMNISET or DITView utility via a Maxiflex CPU, although it is possible to configure a Dual Serial NIM through its Conet/c port that also has access to the NIM's "Data Interchange Table".

The OMNISET utility is supplied free of charge for this purpose. The DITView utility is supplied separately or as part of the EziForth Programmer's Workbench, which is required for programming of the NIM using EziForth.

For simplicity, when the OMNISET is referred to below, DITView can be equally applied.

To setup the Dual Serial NIM through the Maxiflex CPU's programming port using OMNISET, follow this procedure (NOTE: you will need a Maxiflex CPU installed on the rack as well):

1. 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 software.)
2. Open the template file named "015601*nn*(M1580A).dvx" supplied with OMNISET. (The *nn* is replaced with the version number of the latest available dvx file).
3. If you are using DITView, ensure that the Logical Port Selected is setup for the Conet/s protocol. OMNISET users will not need to worry as the port is automatically set by the template file.
4. Ensure that the Target Address is set to "Local Slot n" where n is the I/O Slot number of the NIM. This is shown in the centre of the status bar of OMNISET. To change this address refer to Section 4.3 Addressing the NIM using the Programming Port.
5. The Product Information Group should now be properly displayed, showing the Product Name as "M1580 Dual Serial NIM". The Alive Counter in DIT Register 23 should be counting up to indicate that the NIM is "Alive" and connected to the PC, via the CPU's programming port.
6. You are now ready to view or change any of the parameters in the Dual Serial NIM, or to view the internal dynamic data in the product.

4.2 Configuring the Serial Ports

The serial ports on the Dual Serial NIM comes equipped with the four protocol modes installed. The required protocol mode including any address selection is made by the correct selection of the switches on the Serial Port Switches. Follow the instructions below to set the required protocol mode:

4.2.1 MODBUS Slave Protocol

4.2.1.1. Default MODBUS (ASCII 9600 baud Slave) Mode

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

This port is then configured as a MODBUS slave port operating in MODBUS ASCII mode at 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 Dual Serial NIM through this serial port.

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

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

4.2.1.2. Standard MODBUS Slave Protocol Mode

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

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

Switches 1 to 5 of the DIP switch set the communications address used to access the Dual Serial NIM through this serial port.

Example of Serial Port DIP switch setting

[White square indicates position of switch lever]

Switches 1-5 = ID

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

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 on to set Conet/s default mode of operation

In this example this switch is shown OFF.

Switch 8: Default Modbus ASCII (9600 baud) Protocol Selection

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

Switch 8 OFF = Internal protocol configuration

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.

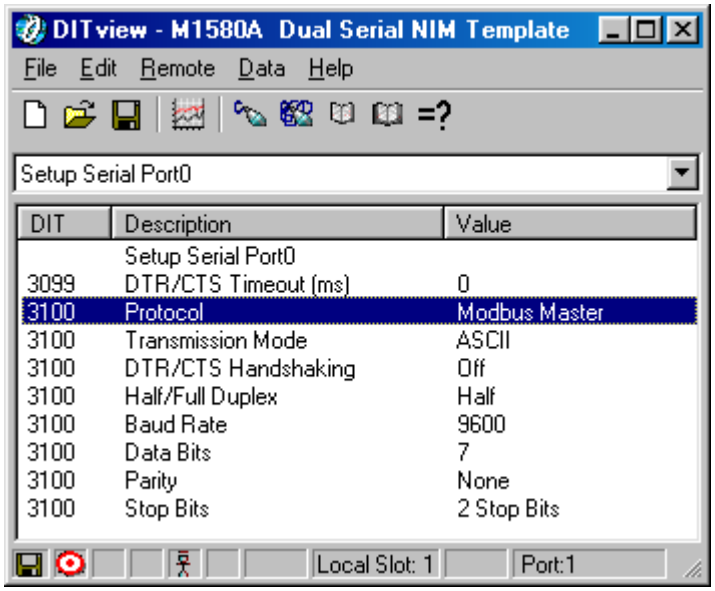
Set as shown, (switches 2 and 7 on) the serial port will operate as a Modbus ASCII slave port at 9600 baud, and will respond to the slave address of 2.

Table 4.1: Serial Port default Address switch settings

4.2.2 MODBUS Master Protocol

4.2.2.1. Standard MODBUS Master Protocol Mode

Modbus Master Protocol is not available as a default communications option but is setup through OMNISET using the "Setup Serial Port" group, including ASCII and RTU mode, Baud Rate, Parity, number of Data bits and Stop bits etc.



Once the protocol selection has been made, it is necessary to configure some general parameters that control the polling regime required in a different group called "Setup Modbus Master Parameters", whereafter the required Modbus Master Queries can be setup for polling Modbus Slave devices.

Please refer to Section 7, Modbus Master Operation Explained for details on how to setup these particular options.

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

4.2.3.1. Default Conet/s Mode (19200 baud)

To select this mode 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 19200 baud (with 8 data bits, no parity and 1 stop bits.)

Switches 1 to 5 of the address switch set the communications address used to access the Dual Serial NIM 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.

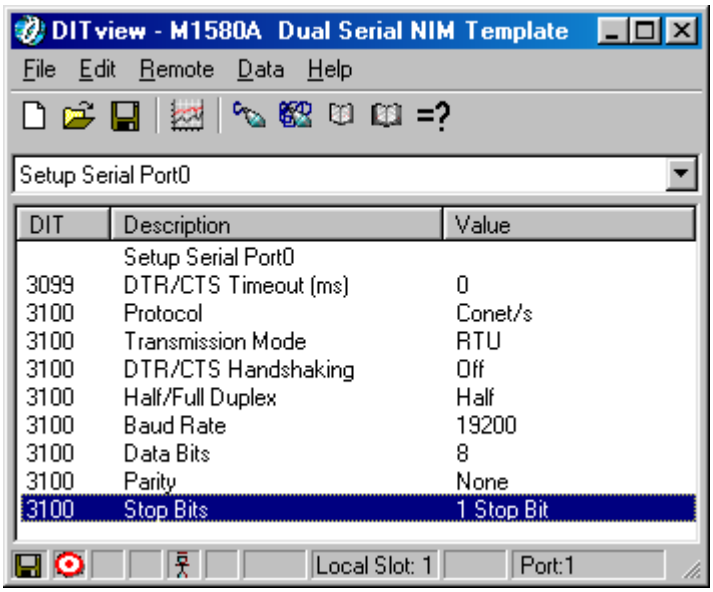
This mode allows foolproof Conet/s communications to be established quickly and easily.

4.2.3.2. Standard Conet/s Protocol Mode

The baud rate of the Conet/s protocol is settable through OMNISET/DITview in the “Setup Serial Port0” group for Serial Port0 and “Setup Serial Port1” group for Serial Portt1 in this mode.

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 Dual Serial NIM through this serial port.

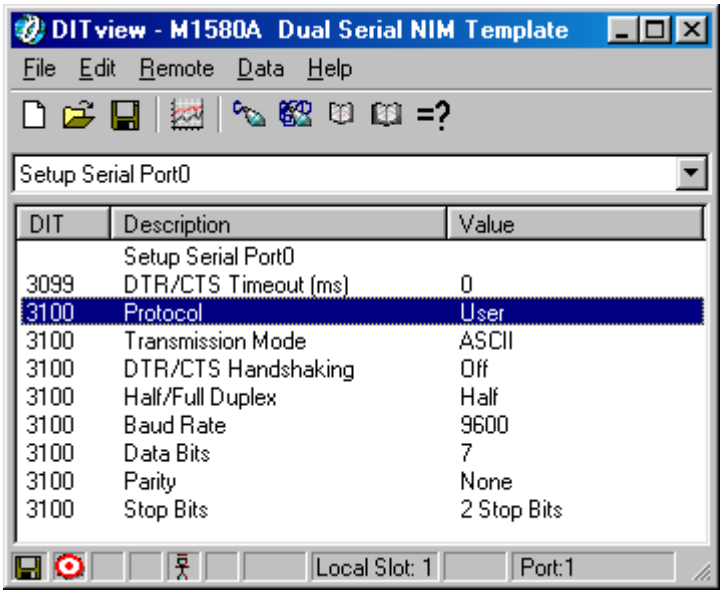


4.2.4 Custom User Protocol Mode

In order to use this advanced feature of the Dual Serial NIM, the “User” Protocol type must be configured in the “Setup Serial Port0” configuration group for Serial Port0 or “Setup Serial Port1” configuration group for Serial Port1, and a custom protocol driver must have been downloaded to the Dual Serial NIM.

Consult the factory for available protocols, or for assistance in writing your own protocol in EZIFORTH.

In this mode, the DIP switch is unassigned by the Dual Serial NIM, but may be used by the serial protocol downloaded.



4.3 Addressing a NIM Module through the Programming Port of the Maxiflex CPU

The first step when starting to configure a Dual Serial NIM in a Maxiflex System with a Maxiflex CPU is to select the address of the module. In Omniset and DITView, select the module to be addressed using the pull-down menus as follows:

From the pull-down menu select “Remote”, then “File Target Address…”, “Group Target Address” or “Target Address…” whichever is displayed on the menu. (These variations are explained in the Omniset and DITView Help files, and are not relevant to the selection of the Target Address as discussed here.)

A Selection Window as shown below will be displayed:

The different options are described below:

4.3.1 No Address set

This mode is used in Omniset or DITview, when the Group Selected is required to follow the File settings. ie. All Groups use the same Target Address setting as set in

the File Header. If another setting is chosen for the Group, then this setting will override the File setting.

4.3.2 Local CPU

Select this option to communicate directly with the Maxiflex CPU to which the programming tool is connected.

4.3.3 Intelligent Module in I/O Slot

Select this option to communicate with an intelligent I/O module (typically a NIM) on the same base as the CPU to which the programming tool is connected. Enter the Slot number in the text box provided. USE THIS OPTION TO ACCESS THE Dual Serial NIM MODULE IN THE RELEVANT SLOT.

4.3.4 Local Network ID (using Offset)

This option uses the network routing facility in the CPU to communicate with remote nodes through the programming port of the CPU.

Select this option to communicate with another node connected to a network that is connected to this NIM. For further information consult the relevant CPU User Guide on how to setup Routing.

4.3.5 Direct/Global Network ID

If the CPU to which you are connected has had its Network Routing Table configured, then all nodes in the system will have been allocated Global Addresses (in the range 127 to 254). Use this option to select any target node in the system using its Global Address. Consult the CPU User Guide on Conet Routing.

4.4 Programming the Dual Serial NIM

Please refer to the separate EziForth Programmer's Guide.

5. The Data Interchange Table explained

5.1 DIT Table Layout

The Data Interchange Table (or DIT) in the Dual Serial NIM is a table of up to 4000 16-bit data registers used for reading and writing all configuration and dynamic data to the NIM.

NIM Dynamic DIT

DIT registers 0 to 2999 comprise the Dynamic Data Space and this memory is non-volatile. This means that data stored in this area will be lost on power down. These DIT registers are automatically cleared to zero on power up. This area is intended for run-time dynamic data only.

NIM Configuration DIT

DIT registers 3000 to 3999 comprise the Configuration Data Space and this memory non-volatile and data stored in this section is retained during power fail. NIM configuration such as Subscription setup is stored in this area.

There is also a “scratch pad” area available in this section to the User for application specific configuration that will remains fixed for the life of the application.

Writing to this area of the DIT is slow, and continuous writing to this area during normal system operation should be avoided.

The T2 CPU now supports up to 65500 DIT registers to cater for I/O module access. The following table shows the address map of the DIT table for an entire MAXIFLEX system as viewed from the T2 CPU:

Maxiflex 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	64000	4000	8000	12000	16000	20000	24000	28000
DIT End Address:	3999	65499	7999	11999	15999	19999	23999	27999	31999

Maxiflex 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:	32000	36000	40000	44000	48000	52000	56000	60000
DIT End Address:	35999	39999	43999	47999	51999	55999	59999	63999

Table 5.1: DIT Address Map of the T2 CPU:

The map is different for the M125x T3 CPUs and the M126x P3 CPUs as these CPUs have more dynamic data. This narrows the address space available for NIM modules. The T3 and P3 CPUs can view the DIT tables of NIMs in I/O slots in blocks of 2000 DITs per slot or blocks of 4000 DITs per slot. Please consult the relevant CPU User Guide for further details on this.

6. The Subscription Service Explained

6.1 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 update times. It also limits the number of master nodes in the system to one.

The Maxiflex Dual Serial NIM 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.

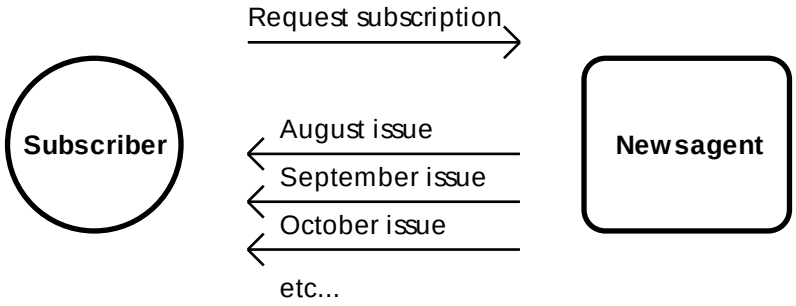


Figure 6-2 The Magazine Subscription Analogy.

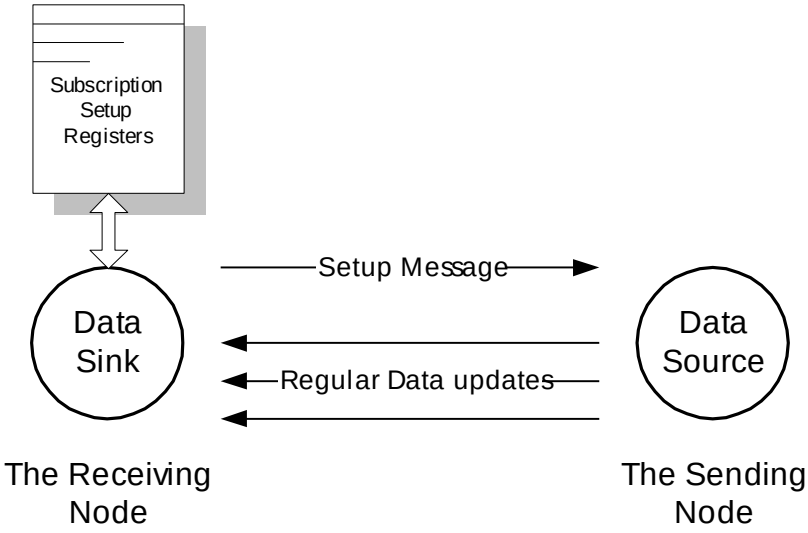
In the same way, the Dual Serial NIM's Subscription Service allows the NIM (acting as a node on the network) to subscribe to a range of DIT registers on a remote node.

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.

6.2 Setting up subscriptions

Just like the magazine subscription, the receiving NIM (or any other Conet/c device supporting the Subscription service) 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 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 setup between nodes in different directions, each of which would operate independently of any other. This provides far more flexibility than typical Poll/Response methods.



The node acting as a data “source” (the transmitting node) requires no user setup to participate in a subscription contract with another node. The node acting as data “sink” (the receiving node) is setup by the user with the required data. Thereafter the “sink” node automatically manages the subscription including setting up the data source, and managing errors in the process.

The following table shows the configuration required to setup the subscription:

ITEM	DESCRIPTION	VALID RANGE
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 – 126 Global network addresses: 127 – 254
Local Port Number	This is the Network Port number on the network module in the local system to which the network containing the 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 only for the Conet NIM
Destination DIT Start Address	This is the DIT register address where the data will start to be written in this NIM acting as the data sink.	0 – 65535
Data Range	This is the number of 16 bit DIT registers that will be transferred in the subscription	1 – 120
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
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
Change-of-State Required	This sets whether the subscription data block will be sent when any data in the block changes.	Yes/No

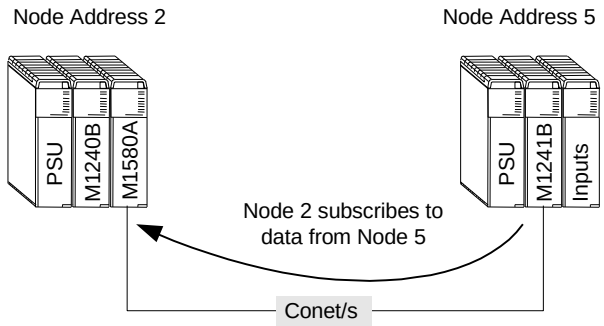
6.3 Number of subscriptions allowed

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

In other words each NIM can act as data sink for 8 subscriptions, and data source for 8 subscriptions simultaneously.

6.4 Subscription Application Example

Refer to the diagram of a simple network below:



Remote node 5 (CPU module) on the Conet/s network has digital information in DIT registers 620 to 627. The Central Node 2 (NIM module) 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 NIM to configure this function:

NAME	VALUE	DESCRIPTION
Source Node Address	5	The local CONET/S ID on the Conet/s network
Local Port Number	0	The Conet/s network is Port 0 when using Serial Port0 and Port 1 when using Serial Port1 on the NIM
Destination DIT Start Address	100	This is the DIT Address where we want to start writing the received data.
Data Range	8	We require 8 DIT registers to be sent
Source DIT Start Address	620	The Source Data start at DIT address 620
Update Time	10	The regular updates can be 10 seconds apart.
Change of State Required	Yes	The data will also be sent whenever any bit the DIT registers in the range 620-627 changes.

When configured, Node 2 automatically requests a subscription with the following data from Node 5 on the Conet/s network connected to the NIM.

Node 5 undertakes to send the contents of its own DIT registers 620 to 627 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 100 to 107.

7. Modbus Master Operation Explained

7.1 Introduction to Modbus Master Driver

The Modbus Master Driver supports up to 32 Queries to read and write data from third party devices. These queries can be any combination of One-shot Queries and Cyclic Queries. In order to use the Modbus Master Driver, the serial port must be configured for this use, otherwise none of the Modbus Master configuration changes will take effect. Please refer to Section 4.2.2, MODBUS Master Protocol to set up the serial port in this way.

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.

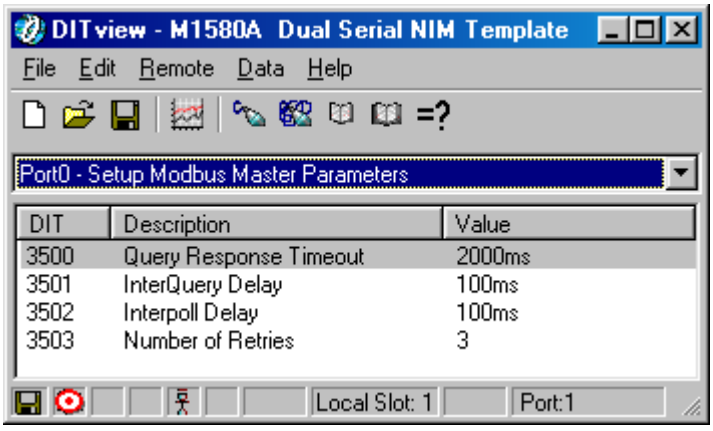
The Modbus Master Driver is also easily adaptable to application demands as it is possible to interleave One-shot and Cyclic queries as desired. One-shot queries are triggered by setting a bit in the DIT table, which is easily achieved from the local application program or remotely through any of the 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.

7.2 Modbus Master Parameters

These are some general parameters that control the manner in which queries are processed. They are configured via Omniset using the "Setup Modbus Master Parameters" 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.

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.

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.

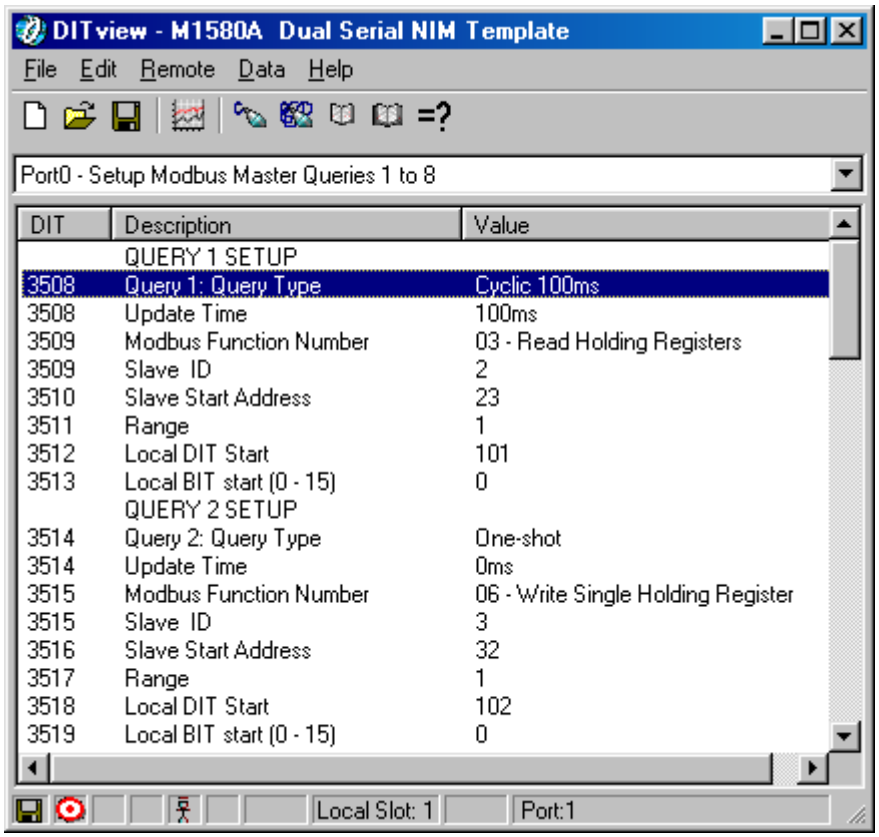
This delay must be specified in milliseconds. E.g. if a delay of 100ms is required then the register must be set to 100.

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.

7.3 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 four groups, 8 queries per group. The figure below shows the group for queries 1 to 8.



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

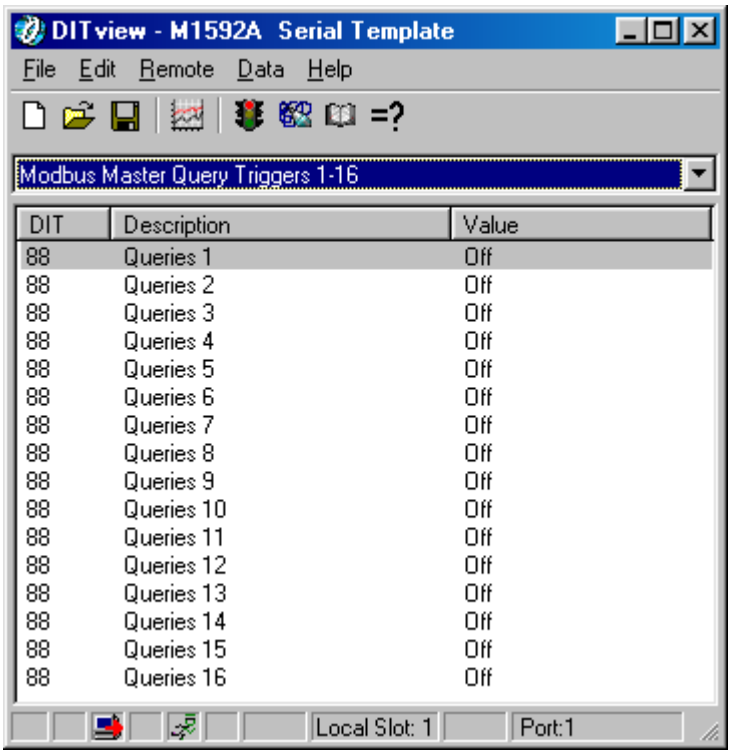
Name	Value(s)	Description
Query Type	Disabled One-shot As fast as possible Cyclic 100ms Cyclic 250ms Cyclic 500ms Cyclic 1000ms	Set disable to ensure the query is not executed at all. Set One-shot if its to be triggered manually Set As fast as possible, or any other Cyclic option for Cyclic queries. 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 ID	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	100 to 2999	Where the NIM 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 7.1 Modbus Master Query Settings

7.4 Query Triggers

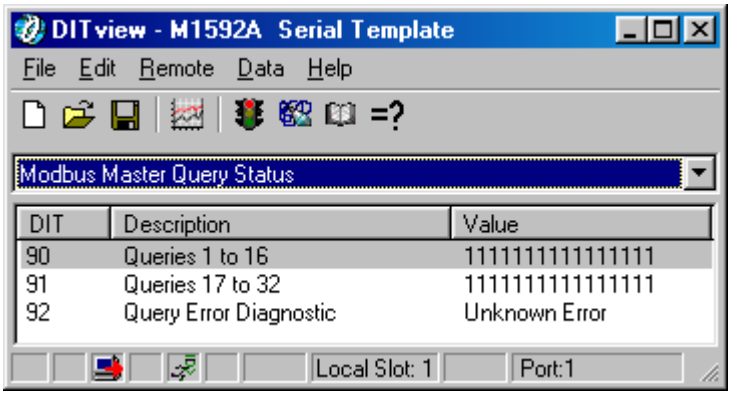
Query Triggers are the mechanism by which all queries are sent. One-shot queries are triggered manually, either by user application or by another device or PC connected remotely via one of the network ports on the CPU. Cyclic queries are triggered automatically by the operating system of the CPU. A single 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.

7.5 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	ID in response doesn't match ID 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 7.2 Modbus Master Query Error Codes

8. Appendix: Dual Serial NIM Detailed DIT Layout:

8.1 Dual Serial NIM Dynamic Data Area DIT Register Assignment

(DIT Registers 0 – 2999)

DIT Number	Description
0 – 23	**Dual Serial NIM System Information**
0	**Product Code** Unique Number in BCD format that reflects the type of product as follows: 150 – M1580A Dual Serial NIM
1	**DIT Revision Number** Version Number of the DIT Layout supported by the NIM.
2	**Kernel Version Number** Version number of the NIM Kernel. BCD format.
3	**Supported Services Flags** Network related services supported by the Kernel: Bit 0: DIT service Bit 1: Subscription service Bit 2: Queue service Bit 3: Datagram service Bit 4: Routing service Bit 5: Reserved Bit 6: Programming service Bit 7: Reserved Bits 8-15: Reserved Bits set indicate which service is supported.
4-7	**User Tag** User configurable string of DIT registers in ASCII format. The user is able to write up to 8 ASCII characters into these DIT registers (two characters per register) to uniquely identify each CPU.
8-21	RESERVED
22	**System Register** Indicates the following conditions current in the NIM: Bit 0: Application program HALTED (0) or RUNNING (1) Bits 1-15: Reserved
23	**Alive Counter** Free running counter in the NIM. Useful for communications diagnostics. While this register is incrementing, the Operating System is running. If this register is static, the NIM has halted or communications has failed.
24 – 68	RESERVED
69 – 78	**Networking Information**

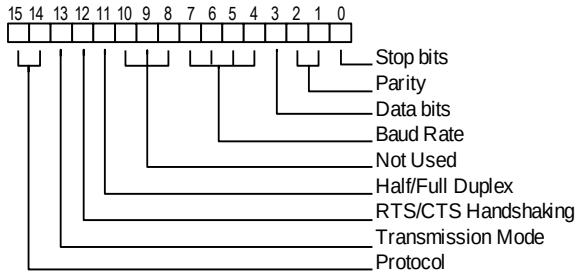
DIT Number	Description
	Status Information for the available network ports on the CPU hardware.
69	**Subscriber Status** A bit map indicating the status of 8 receive subscriptions in the system as follows: Bit 0: link healthy (0) or unhealthy (1) or not configured (1) for Subscription 1 Bit 1: link healthy (0) or unhealthy (1) or not configured (1) for Subscription 2 Bit n: link healthy (0) or unhealthy (1) or not configured (1) for Subscription n+1 Bit 7: link healthy (0) or unhealthy (1) or not configured (1) for Subscription 8 Bits 8-15: Reserved Note: Link Healthy means that the receiving node is happy that subscriptions are being received from the transmitting node at the required intervals. Link Unhealthy means that the receiving node has not received subscriptions from the transmitter and flags an error condition. Should the transmitting node resume transmissions, the flag will be cleared to indicate healthy link dstatus
70	RESERVED
71	**Serial Port0 Inactivity** A seconds counter that is reset to zero every time there is active communications on Serial Port0. If there is no valid communications, the counter is incremented every second.
72	**Serial Port1 Inactivity** A seconds counter that is reset to zero every time there is active communications on Serial Port1. If there is no valid communications, the counter is incremented every second.
73	RESERVED
74	**Serial Port0 Local Address** The local ID setting for the Serial Port0 as set on the Serial Port0 dipswitch, swtiches 1 to 6.
75	**Serial Port0 Global Address** Global ID of the NIM on Serial Port0. Only applies if the Network Routing Table is configured to include this port. Note that as the default Network Routing Table does not include this port, the default Global ID setting is set to 255 which is an invalid Global ID.
76	**Serial Port1 Local Address** The local ID setting for the Serial Port1 as set on the Serial Port1 dipswitch, swtiches 1 to 6.
77	**Serial Port1 Global Address** Global ID of the NIM on Serial Port1. Only applies if the Network Routing Table is configured to include this port. Note that as the default Network Routing Table does not include this port, the default Global ID setting is set to 255 which is an invalid Global ID.
78	**Serial Port0 dipswitch** Displays the current Serial Port0 diswitch setting.
79	**Serial Port1 dipswitch** Displays the current Serial Port1 diswitch setting.
80 - 87	**System Timers**

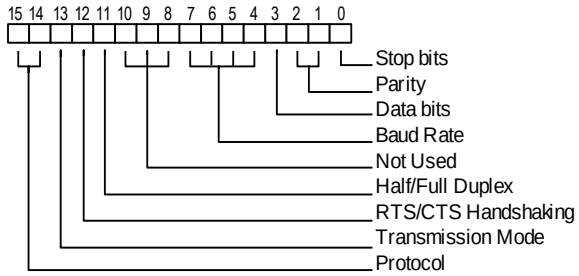
DIT Number	Description
	System count down timers decremented by the Operating System until zero is reached. Values entered are decremented by 1 every 10 milliseconds. For example, enter 100 to time for 1 second. The largest timer available is 655350 ms or about 11 minutes.
80	System Timer 0
81	System Timer 1
82	System Timer 2
83	System Timer 3
84	System Timer 4
85	System Timer 5
86	System Timer 6
87	System Timer 7
88	**Port0 - Modbus Master Query Triggers 1 to 16** Set this trigger register only when One Shot Queries are been used. Refer to the 0150xxxx(M1580A).dvx file for further information.
89	**Port0 - Modbus Master Query Triggers 17 to 32** Set this trigger register only when One Shot Queries are been used. Refer to the 0150xxxx(M1580A).dvx file for further information.
90 - 92	**Port0 - Modbus Master Query Status** When a configured Modbus Master query has had a successful poll then the appropriate bit is cleared for that Query. If the Query was unsuccessful or if no queries are configured then the bit will be set .
90	Port0 - Query 1 to 16 Bit Status
91	Port0 - Query 17 to 32 Bit Status
92	Port0 - Query Error Diagnostic Register.
93	**Port1 - Modbus Master Query Triggers 1 to 16** Set this trigger register only when One Shot Queries are been used. Refer to the 0150xxxx(M1580A).dvx file for further information.
94	**Port1 - Modbus Master Query Triggers 17 to 32** Set this trigger register only when One Shot Queries are been used. Refer to the 0150xxxx(M1580A).dvx file for further information.
95 - 97	**Port1 - Modbus Master Query Status** When a configured Modbus Master query has had a successful poll then the appropriate bit is cleared for that Query. If the Query was unsuccessful or if no queries are configured then the bit will be set .
95	Port1 - Query 1 to 16 Bit Status
96	Port1 - Query 17 to 32 Bit Status
97	Port1 - Query Error Diagnostic Register.
110 – 2999	**User Data Area**

8.2 Dual Serial NIM Configuration Data Area DIT Register Assignment

(DIT Registers 3000 – 3999)

This DIT area is maintained in the NIM as non-volatile memory. All configuration of NIM functions is therefore implemented here so that configuration is not lost during power down. In addition, there are some registers available for the User to place some configuration parameters for application programs.

DIT Number	Description
3000 – 3014	**Application Information** This is a scratch pad of non-volatile memory for the User to place application information.
3000 – 3013	**Application Name** The User may write up to 28 ASCII characters for the name of the application.
3014	**Application Version** The User may write the version number of the application in Hex format.
3015 – 3094	RESERVED
3095 – 3100	**Serial Port0 and Serial Port1 Configuration**
3095	**Serial Port1 RTS/CTS Timeout** If RTS/CTS handshaking is enabled, then the transmitter will assert RTS when it is ready to transmit. If CTS is asserted before the timeout period in this register, then transmission will commence. If CTS is not asserted, then the transmission will be aborted. Handshaking is enabled by setting bit 12 of register 3100.
3096	**Serial Port1 Setup** This register sets up the communications parameters for the serial port on the NIM. The format of this register is as follows:  Stop Bits: 1 = 1 stop bit, 0 = 2 stop bits Parity: 00 = none, 01 = odd, 10 = even Data Bits: 1 = 8 data bits, 0 = 7 data bits Baud: 0000 = 300 baud 0001 = 600 baud 0010 = 1200 baud 0011 = 2400 baud 0100 = 4800 baud 0101 = 9600 baud 0110 = 19200 baud

DIT Number	Description
	0111 = 38400 baud RTSCTS Handshaking: 1 = ON, 0 = OFF Transmission Mode: 1 = ASCII, 0 = RTU(Binary) Protocol: 00 = Reserved 01 = Modbus 10 = User Protocol
3099	**Serial Port0 RTS/CTS Timeout** If RTS/CTS handshaking is enabled, then the transmitter will assert RTS when it is ready to transmit. If CTS is asserted before the timeout period in this register, then transmission will commence. If CTS is not asserted, then the transmission will be aborted. Handshaking is enabled by setting bit 12 of register 3100.
3100	**Serial Port0 Setup** This register sets up the communications parameters for the serial port on the NIM. The format of this register is as follows:  Stop Bits: 1 = 1 stop bit, 0 = 2 stop bits Parity: 00 = none, 01 = odd, 10 = even Data Bits: 1 = 8 data bits, 0 = 7 data bits Baud: 0000 = 300 baud 0001 = 600 baud 0010 = 1200 baud 0011 = 2400 baud 0100 = 4800 baud 0101 = 9600 baud 0110 = 19200 baud 0111 = 38400 baud RTSCTS Handshaking: 1 = ON, 0 = OFF Transmission Mode: 1 = ASCII, 0 = RTU(Binary) Protocol: 00 = Reserved 01 = Modbus 10 = User Protocol
3101 – 3120	**Network Routing Table** These registers contain the data that allows the NIM to route packets across networks e.g. routing a packet from a Modbus network connected to the serial port of the CPU, to

DIT Number	Description
	a Conet/s network connected to the Conet/s port of the PNIM. A MS Excel Spreadsheet is used to generate the data for this table. This spreadsheet is available from Omniflex if the user requires to route messages with the T1 CPU+NIM. NOTE that once the table is updated in the CPU, the equivalent data is setup in the NIM module automatically. There is no need for the user to enter this data repeatedly in the NIM modules fitted the rack.
3101	**Setup for Network 1** The default setting for this register is 64768 when no User routing table has been configured. If a routing table has been configured and the User wishes to revert back to the default setting, he may simply write zero to this register and the default settings will be invoked
3102	Setup for Network 2
3102	Setup for Network 3
3102	Setup for Network 4
3102	Setup for Network 5
3102	Setup for Network 6
3102	Setup for Network 7
3102	Setup for Network 8
3102	Setup for Network 9
3102	Setup for Network 10
3102	Setup for Network 11
3102	Setup for Network 12
3102	Setup for Network 13
3102	Setup for Network 14
3102	Setup for Network 15
3102	Setup for Network 16
3102	Setup for Network 17
3102	Setup for Network 18
3102	Setup for Network 19
3102	Setup for Network 20
3102	Setup for Network 21
3102	Setup for Network 22
3102	Setup for Network 23
3102	Setup for Network 24
3102	Setup for Network 25
3102	Setup for Network 26
3102	Setup for Network 27
3102	Setup for Network 28
3102	Setup for Network 29
3102	Setup for Network 30
3102	Setup for Network 31
3102	Setup for Network 32
3102	Setup for Network 33

DIT Number	Description
3102	Setup for Network 34
3102	Setup for Network 35
3102	Setup for Network 36
3102	Setup for Network 37
3102	Setup for Network 38
3102	Setup for Network 39
3102	Setup for Network 40
3102	Setup for Network 41
3102	Setup for Network 42
3102	Setup for Network 43
3102	Setup for Network 44
3102	Setup for Network 45
3200 – 3247	**Subscription Blocks Table** These registers contain the data that allows the NIM to subscribe to data on another device on the network. The subscription mechanism allows this NIM so "subscribe" to a block of data on another device on the network. That block of data will be written automatically to the specified location on this NIM on a regular basis as well as on a change of state of the remote data. Up to 8 subscriptions may be setup on any NIM as the recipient (and up to 8 subscriptions may be received from other devices on the network, although this requires no setup and operates transparently.) A Status register monitors the performance of subscriptions, and if communication is lost with the remote node, then an alarm can be generated.
3200-3205	**Subscription Block 1**
3200	Remote ID Set the local ID of the remote node from which data is required.
3201	Local Port Number Set the Port Number on this NIM through which the remote node can be found. The Conet/c port number is 0 for the Conet NIM.
3202	Local Start DIT Set the start address of the destination DIT block in this CPU where you would like the remote data to be written.
3203	Range Set the number of DIT registers to be transferred in the subscription (64 maximum)
3204	Remote Start DIT Set the Start Register Number in the remote node where the source data is located.
3205	Update time Set the time interval for regular updates of the data.

DIT Number	Description
3206-3211	**Subscription Block 2**
3206	Remote ID
3207	Local Port Number
3208	Local Start DIT
3209	Range
3210	Remote Start DIT
3211	Update time
3212-3217	**Subscription Block 3**
3212	Remote ID
3213	Local Port Number
3214	Local Start DIT
3215	Range
3216	Remote Start DIT
3217	Update time
3218-3223	**Subscription Block 4**
3218	Remote ID
3219	Local Port Number
3220	Local Start DIT
3221	Range
3222	Remote Start DIT
3223	Update time
3224-3229	**Subscription Block 5**
3224	Remote ID
3225	Local Port Number
3226	Local Start DIT
3227	Range
3228	Remote Start DIT
3229	Update time
3230-3235	**Subscription Block 6**
3230	Remote ID
3231	Local Port Number
3232	Local Start DIT
3233	Range
3234	Remote Start DIT
3235	Update time
3236-3241	**Subscription Block 7**
3236	Remote ID
3237	Local Port Number
3238	Local Start DIT
3239	Range

DIT Number	Description
3240	Remote Start DIT
3241	Update time
3242-3243	**Subscription Block 8**
3242	Remote ID
3243	Local Port Number
3244	Local Start DIT
3245	Range
3246	Remote Start DIT
3247	Update time
3248 - 3499	RESERVED
3500 – 3699	**Port0 Setup Modbus Master Parameters** These registers contain data that allows the NIM Port0 to act as a Modbus Master to other Modbus Slave Devices. Upto to 32 Modbus Queries can be configured for this port. ***Note Remember to set Serial Port0 for Modbus Master operation in the Setup Serial Port0 Group when you require the use of this feature**.
3500	**Query Response Timeout** Enter a value in the range of 0 to 65535ms in this register for the response timeout. This configuration is used to allow the Master to wait for this time for a response from the Slave device. In the event that the Slave device does not reply within this timeout then the Modbus Master engine will flag a Timeout error.
3501	**InterQuery Delay** Enter a value in the range of 0 to 65535ms in this register for the InterQuery delay. This will cause the Modbus Master engine to wait for the specified time before sending the next Query.
3502	**Interpoll Delay** Enter a value in the range of 0 to 65535ms in this register for the Interpoll Delay. This will cause the Modbus Master engine to wait for the specified time before sending the first query.
3503	**Number of Retries** Enter a number in the range of 0 to 65535 in this register for the Number of Retries. This will cause the Modbus Master engine to send the same query for the specified number of retries entered in this register until the Slave responds or the Number of retries for that query has elapsed, before sending the next query.
3504 - 3507	Reserved
3508	**Query1 Type/Update Time**
3509	Modbus Function Number/Slave ID ***Note: Refer to the 0150xxxx(M150A).dvx file for further information.**

DIT Number	Description
3510	Slave Start Address
3511	Range
3512	Local DIT Start
3513	Local BIT start (0-15)
3514 - 3519	Port0 Query2 Setup
3520 - 3525	Port0 Query3 Setup
3526 - 3531	Port0 Query4 Setup
3532 - 3537	Port0 Query5 Setup
3538 - 3543	Port0 Query6 Setup
3544 - 3549	Port0 Query7 Setup
3550 - 3555	Port0 Query8 Setup
3556 - 3561	Port0 Query9 Setup
3562 - 3567	Port0 Query10 Setup
3568 - 3573	Port0 Query11 Setup
3574 - 3579	Port0 Query12 Setup
3580 - 3585	Port0 Query13 Setup
3586 - 3591	Port0 Query14 Setup
3592 - 3597	Port0 Query15 Setup
3598 - 3603	Port0 Query16 Setup
3604 - 3609	Port0 Query17 Setup
3610 - 3615	Port0 Query18 Setup
3616 - 3621	Port0 Query19 Setup
3622 - 3627	Port0 Query20 Setup
3628 - 3633	Port0 Query21 Setup
3634 - 3639	Port0 Query22 Setup
3640 - 3645	Port0 Query23 Setup
3646 - 3651	Port0 Query24 Setup
3652 - 3657	Port0 Query25 Setup
3658 - 3663	Port0 Query26 Setup
3664 - 3669	Port0 Query27 Setup

DIT Number	Description
3670 - 3675	Port0 Query28 Setup
3676 - 3681	Port0 Query29 Setup
3682 - 3687	Port0 Query30 Setup
3688 - 3693	Port0 Query31 Setup
3694 - 3699	Port0 Query32 Setup
3700 – 3899	**Port1 Setup Modbus Master Parameters** These registers contain data that allows the NIM Port1 to act as a Modbus Master to other Modbus Slave Devices. Upto to 32 Modbus Queries can be configured for this port. ***Note Remember to set Serial Port1 for Modbus Master operation in the Setup Serial Port1 Group when you require the use of this feature**.
3700	**Query Response Timeout** Enter a value in the range of 0 to 65535ms in this register for the response timeout. This configuration is used to allow the Master to wait for this time for a response from the Slave device. In the event that the Slave device does not reply within this timeout then the Modbus Master engine will flag a Timeout error.
3701	**InterQuery Delay** Enter a value in the range of 0 to 65535ms in this register for the InterQuery delay. This will cause the Modbus Master engine to wait for the specified time before sending the next Query.
3702	**Interpoll Delay** Enter a value in the range of 0 to 65535ms in this register for the Interpoll Delay. This will cause the Modbus Master engine to wait for the specified time before sending the first query.
3703	**Number of Retries** Enter a number in the range of 0 to 65535 in this register for the Number of Retries. This will cause the Modbus Master engine to send the same query for the specified number of retries entered in this register until the Slave responds or the Number of retries for that query has elapsed, before sending the next query.
3704 - 3707	Reserved
3708	**Query1 Type/Update Time**
3709	Modbus Function Number/Slave ID ***Note: Refer to the 0150xxxx(M150A).dvx file for further information.**
3710	Slave Start Address
3711	Range
3712	Local DIT Start

DIT Number	Description
3713	Local BIT start (0-15)
3714 - 3719	Port1 Query2 Setup
3720 - 3725	Port1 Query3 Setup
3726 - 3731	Port1 Query4 Setup
3732 - 3737	Port1 Query5 Setup
3738 - 3743	Port1 Query6 Setup
3744 - 3749	Port1 Query7 Setup
3750 - 3755	Port1 Query8 Setup
3756 - 3761	Port1 Query9 Setup
3762 - 3767	Port1 Query10 Setup
3768 - 3773	Port1 Query11 Setup
3774 - 3779	Port1 Query12 Setup
3780 - 3785	Port1 Query13 Setup
3786 - 3791	Port1 Query14 Setup
3792 - 3797	Port1 Query15 Setup
3798 - 3803	Port1 Query16 Setup
3804 - 3809	Port1 Query17 Setup
3810 - 3815	Port1 Query18 Setup
3816 - 3821	Port1 Query19 Setup
3822 - 3827	Port1 Query20 Setup
3828 - 3833	Port1 Query21 Setup
3834 - 3839	Port1 Query22 Setup
3840 - 3845	Port1 Query23 Setup
3846 - 3851	Port1 Query24 Setup
3852 - 3857	Port1 Query25 Setup
3858 - 3863	Port1 Query26 Setup
3864 - 3869	Port1 Query27 Setup
3870 - 3875	Port1 Query28 Setup
3876 - 3881	Port1 Query29 Setup
3882 - 3887	Port1 Query30 Setup

DIT Number	Description
3888 - 3893	Port1 Query31 Setup
3894 - 3899	Port1 Query32 Setup