User Guide

CTNet

Part No. 0460-0025

Issue Number: 2

Safety Information

Persons supervising and performing the electrical installation or maintenance of a Drive and/or an external Option Unit must be suitably qualified and competent in these duties. They should be given the opportunity to study and if necessary to discuss this User Guide before work is started.

The voltages present in the Drive and external Option Units are capable of inflicting a severe electric shock and may be lethal. The Stop function of the Drive does not remove dangerous voltages from the terminals of the Drive and external Option Unit. Mains supplies should be removed before any servicing work is performed.

The installation instructions should be adhered to. Any questions or doubt should be referred to the supplier of the equipment. It is the responsibility of the owner or user to ensure that the installation of the Drive and external Option Unit, and the way in which they are operated and maintained complies with the requirements of the Health and Safety at Work Act in the United Kingdom and applicable legislation and regulations and codes of practice in the UK or elsewhere.

The Drive software may incorporate an optional Auto-start facility. In order to prevent the risk of injury to personnel working on or near the motor or its driven equipment and to prevent potential damage to equipment, users and operators, all necessary precautions must be taken if operating the Drive in this mode.

The Stop and Start inputs of the Drive should not be relied upon to ensure safety of personnel. If a safety hazard could exist from unexpected starting of the Drive, an interlock should be installed to prevent the motor being inadvertently started.

General Information

The manufacturer accepts no liability for any consequences resulting from inappropriate, negligent or incorrect installation or adjustment of the optional operating parameters of the equipment or from mismatching the Drive with the motor.

The contents of this User Guide are believed to be correct at the time of printing. In the interests of a commitment to a policy of continuous development and improvement, the manufacturer reserves the right to change the specification of the product or its performance, or the contents of the User Guide, without notice.

Copyright	© February 1999 CT SSPD	
Author:	CT SSPD	
Issue Number:	2	
Software Version:	UD70	System File V2.6.0
	MD29AN	System File V2.6.0
Hardware Version:	UD70 Issue 3	
	UD75 Issue 3	
	MD29AN Issue 1	

Contents

1 Introduction

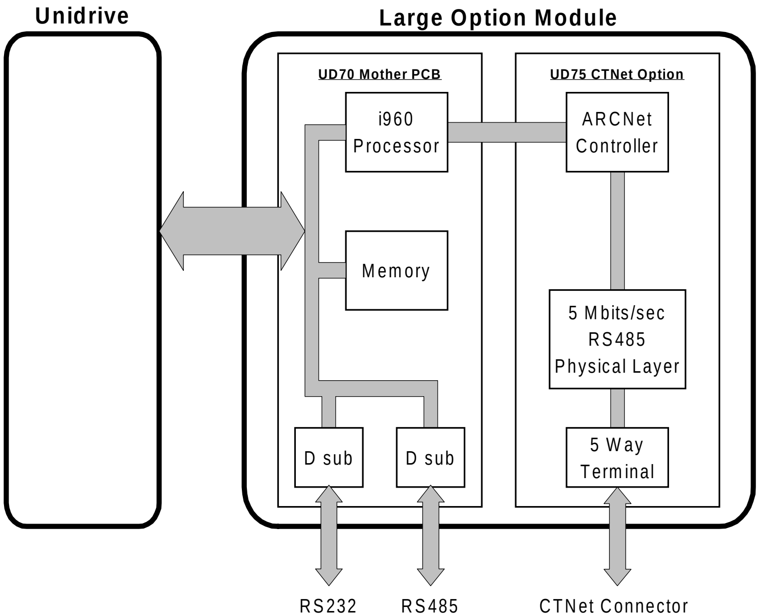
Using this Guide

Drive parameters are denoted in this manual by "# MM.PP", where MM refers to the menu number, and PP refers to the parameter number within that menu. Please refer to the Unidrive and Mentor II manuals for parameter definitions.

1.1 Unidrive - UD75

The UD75 **CTNet** Interface card for Unidrive is supplied in a large option module. It is an add-on card for the UD70 Applications card. The UD70 processor and operating system handles all network activity, and uses a Dual-Port RAM interface to transfer data between itself and the drive.

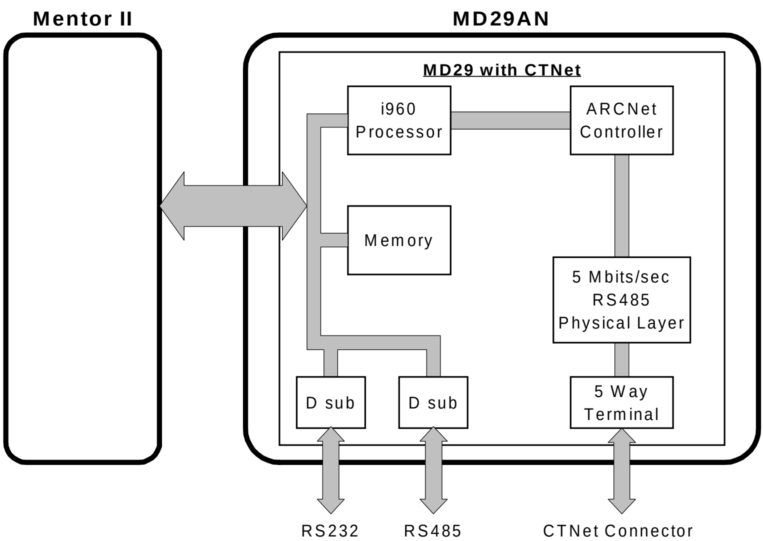
The UD70 retains full functionality, allowing the user to download normal DPL application programs. No program modifications are required to allow existing DPL programs to run. A different UD70 operating system file ("UD70NET.SYS") is used, and the UD70 has this system file pre-loaded.



1.2 Mentor - MD29AN

The MD29AN **CTNet** Interface card for Mentor II is single add-on card. It fits onto the 40 pin header on the MDA-2B card on the Mentor II itself. The MD29AN processor and operating system handles all network activity, and uses a Dual-Port RAM interface to transfer data between itself and the drive.

The MD29AN retains full functionality, allowing the user to download normal DPL application programs. No program modifications are required to allow existing DPL programs to run. A different MD29 operating system file ("M2NET.SYS") is used, and the MD29AN has this system file pre-loaded.

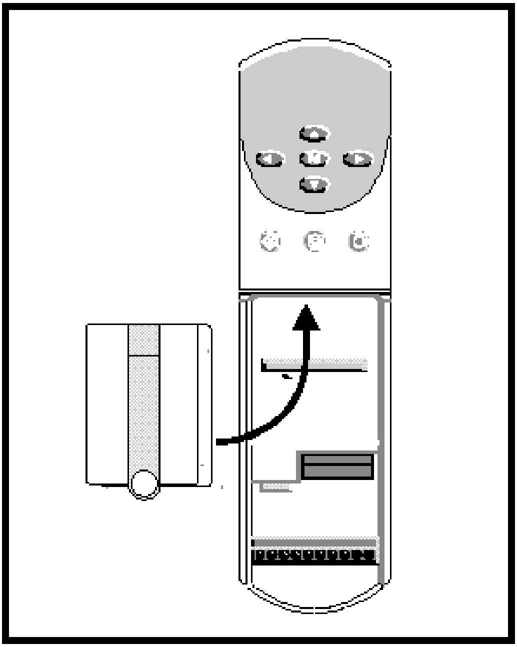


2 Mechanical Installation

Important *The Unidrive or Mentor II must be disconnected from the mains supply before installing or removing an option module.*

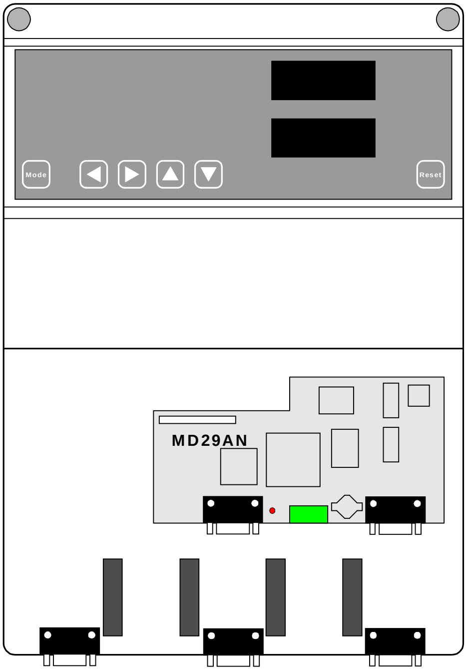
2.1 Unidrive

1. Isolate the drive from the mains supply and allow 5 minutes for the DC Bus capacitors to discharge.
2. Insert Large Option Module as shown below. Ensure that it is correctly inserted. The module will click firmly into place.
3. To remove the module, pull on the black tab, and the module will disengage from the connector and pull out of the drive.



2.2 Mentor II

The MD29AN is to be located upon a 40-way pin header (PL1) on the MDA2B circuit board, as shown below.



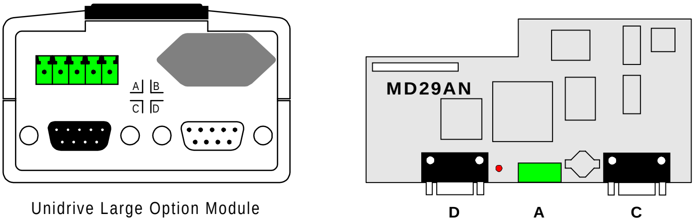
Please take extreme care when locating the board onto this connector - do not force it on. Excessive force may bend and break the pins of the header.

3 Electrical Installation

3.1 CTNet Connectors

The Large Option Module and MD29AN both have the same connectors. The **CTNet** interface uses the green 5 pin terminal block (A), and the 9-way D-type connectors are the RS232 (C) and RS485 (D) ports for the UD70 or MD29.

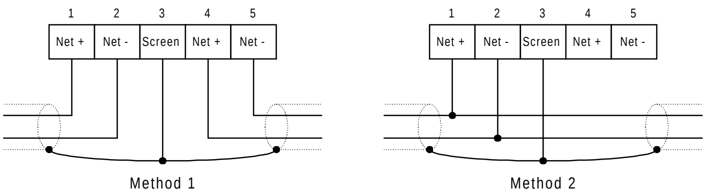
The connections are shown below:



The pin out of the connectors are:-

Function	CTNet	Description
Net +	1,4	Positive data line
Net -	2,5	Negative data line
Screen	3	Cable screen

All three connections **including the cable screen** must be connected using one of the methods shown below:



Pins 1 and 4 are connected via the internal PCB, as are pins 2 and 5. The **CTNet** interface uses Bi-Phase S signalling which is non-phasic, so it does not matter which wire is linked to Net+ and Net-.

Method 1 allows for easy installation by only having a single wire connected to each terminal. Since the Net+ and Net- pins are connected on the internal PCB, it is not necessary to have the drive powered up. If the connector is physically disconnected from the module, this will break the network connection.

Method 2 overcomes the problem of breaking the network in case of disconnection, but it does mean that two wires have to be connected into each terminal. In most cases, this type of connection is not necessary.

3.2 Cable Specification

For maximum noise immunity, **shielded twisted pair** cable must be used for connecting **CTNet** nodes, particularly if high data rates (5.0 or 2.5 Mbits/sec) are to be used.

Characteristic	Min Value	Nominal Value	Max Value	Comments
Impedance		78Ω		
Conductor area	$0.5mm^2$			
Capacitance (line - shield)			70pF/m	
Resistance			40Ω/km	
Line length			200m 400m	@ 5.0 Mbits/sec @ 2.5 Mbits/sec

Customised **CTNet** cable is available from Control Techniques Software and System Products Division.

3.3 Cable Screen

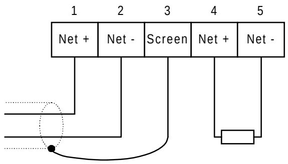
The screen of the cable at every node on the network MUST be connected to the screen terminal (pin 3) on the **CTNet** terminal block. When the screen is stripped back to connect the twisted pair to the **CTNet** terminals, keep the exposed section of the cable as short as possible.

On the **CTNet** PC Interface Adapter Card, there is no screen terminal. The screen should be cut and taped back to prevent it coming into contact with any other surfaces. The on-board terminal resistor should not be connected.

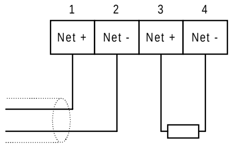
3.4 Network Termination

The network **MUST** be fitted with terminating resistors **AT BOTH ENDS!!!** If resistors are not fitted, the network appear to work OK, but the noise immunity of the network will be drastically reduced. The terminating resistor value should match the nominal characteristic impedance value for the cable; in the case of the customised **CTNet** cable, the terminating resistors used should be nominally 78Ω.

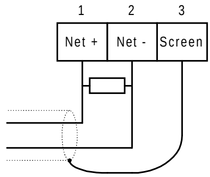
PC ISA and PCMCIA cards for interfacing a PC to **CTNet** are available from CT SSPD. See para 3.5.2. The diagrams below show where to connect the terminating resistor on a UD75 or MD29AN node, the PC ISA card, and the PCMCIA Media Access Unit.



Control Techniques Node



PC Card



PCMCIA Card

3.5 Network Limitations

Networks are capable of driving a maximum of 15 nodes over a maximum network cable length of 100m at 5.0 Mbits/sec. If the network data rate is reduced to 2.5 Mbits/sec, the maximum network length is increased to 200m, but the limit of 15 nodes remains the same.

3.5.1 Network Repeaters

If the combination of network length and number of nodes for a given network data rate cannot be met with a single trunk connection, line repeaters can be used to extend the network. (See Chapter 12, Network Repeaters).

With Issue 3 hardware, a line repeater will allow another section of cable, but it is only capable of driving 10 nodes. Any section of the network that will be driven by a repeater is limited to 9 nodes plus repeater, or 8 nodes plus 2 repeaters. Line termination resistors must be fitted at the end of each section of cable.

3.5.2 PC ISA and PCMCIA Cards

PC ISA and PCMCIA cards are limited to driving 6 nodes. Any section containing a link to a PC interface is limited to 5 nodes plus PC card, or 4 nodes, PC card plus a repeater.

3.5.3 Mixed Networks

Issue 3 and Issue 1 hardware layers are compatible, and can be mixed on a single network with no problems.

In a mixed network, the network limitations for any network segment are determined by the module with the lowest network capabilities. This especially applies when network repeaters, PC ISA and PCMCIA cards are used.

4 Getting Started

There are no hardware links on the UD75 and MD29AN to be set to configure the node for network use. All network specifications are entered using parameters within the Unidrive or Mentor II.

4.1 Network Node Address

Set the desired node address in Unidrive parameter #20.01 and Mentor II parameter #14.05. Every node must have a unique network address, and valid addresses are between 1 and 255. If an invalid address is set, the UD75 or MD29AN will fall back to the default address of 1. Node address changes only take effect after the UD75 or MD29AN has been reset.

On Unidrive, a “soft” reset can be generated by setting #MM.00 to 1070 and pressing the red **RESET** button on the keypad. This only resets the UD70/UD75 interface, not the Unidrive. The MD29AN can be reset using the **RESET** button on the Mentor II keypad. (Note: if the drive is enabled then a reset sequence will not be generated).

If the Unidrive or Mentor II is tripped, resetting the trip will also cause a full reset of the UD70/UD75 or MD29AN.

4.2 Network Data Rate

All nodes on the network must be configured to run at the same data rate. The data rate is set by programming the appropriate code into #20.02 on Unidrive, and #11.01 on Mentor II. If a node has a different data rate compared to the rest of the network, it may prevent the network from initialising properly. The data rate can range from 5.0 Mbits/sec to 312.5 Kbits/sec, depending on the total cable length of the network.

The maximum data rate that can be supported depends on the total length of cable in the network. (See Section 3.5, Network Limitations). The UD75 and MD29AN support the following data rates up to the specified total network cable length:

Data Rate (bits/sec)	Unidrive (#20.02)	Mentor II (#11.01)	Max Network Length (m)
5.0M	0	0	100
2.5M	1	1	200
1.25M	2	2	400
625K	3	3	800
312.5K	4	4	1200

4.3 Synchronisation Message

The cyclic data synchronisation message in specified in #20.03 on Unidrive and #11.02 on Mentor II. When the cyclic data feature is used, each node can transmit data to other nodes when it receives the network token. Since **CTNet** is a peer-to-peer system, with no overall master controller, one node has to be given the task of generating the "synchronisation message". This message is generated on a programmed time-base, and is transmitted to all other nodes at the start of each cyclic data period.

There are 2 cyclic data channels available on **CTNet**: fast cyclic data and slow cyclic data. The fast cyclic data channel is transmitted on synchronisation message cycle, while the slow cyclic channel is transmitted every nth cycle.

The synchronisation message is entered in the form NNCC, where CC is the fast cyclic data time-base in milliseconds, and NN specifies the integer multiple for the slow cyclic data. Setting a value of 0 disables the synchronisation message from the node.

For example, setting a value of 510 would produce a fast cyclic channel transmission every 10ms, and a slow cyclic channel transmission every 50ms.

NOTE *Only ONE node on the network must be set to generate the synchronisation message.*

4.4 Initialising Set-up Changes

When all network configuration parameters have been set, the new values only take effect after the UD70 or MD29AN has been reset. On Unidrive, a soft reset can be generated by setting #MM.00 to 1070 and pressing the red **RESET** button on the keypad. The MD29AN can be reset using the **RESET** button on the Mentor II keypad.

NOTE *if the drive is enabled then a reset sequence will not be generated.*

Function	Unidrive	Mentor II	Range	Comments
Node Address	#20.01	#14.05	1 to 255	Set node address to 0 to disable the **CTNet** driver.
Network Data Rate	#20.02	#11.01	0 to 4	All nodes on the network must be programmed to run at the same data rate.
Synchronisation Message	#20.03	#11.02	NNCC	CC = fast cyclic data rate in ms. NN = slow cyclic rate as multiple of fast cyclic rate. NN ranges from 2 to 19.

4.5 Network Initialisation Failure

There are several rules that must be satisfied for the network to initialise correctly.

- All nodes must have a unique node address.

- All nodes must have the same data rate setting.
- Only one node must be set to generate the synchronisation message.

If any of the above conditions are not set correctly, the node may trip on one of the following, indicating the error.

Trip Code (Unidrive)	Trip Code (Mentor II)	Error
tr52	A29 (52)	Bit 4 in the control word has been set to 1.
tr56	A29 (56)	The UD70 or MD29AN contains the wrong operating system. Download “UD70NET.SYS” or “M2NET.SYS”.
tr57	A29 (57)	An illegal operating system call has been made, e.g. command does not exist, or is not being used correctly.
tr60	A29 (60)	There is a hardware failure on the UD75 or MD29AN.
tr61	A29 (61)	The data rate or node address settings are out of range.
tr62	A29 (62)	The data rate does not match the data rate of the rest of the network.
tr63	A29 (63)	The node address already exists on the network.
tr64	A29 (64)	The cyclic data rate setting is invalid.

NOTE *The Mentor II will trip and display “A29”. The reason for the trip can be found by checking the trip code value in #10.35.*

4.6 Network Loss

If the network connection to a node is lost for any reason, the drive will not trip automatically. DPL code must be written to cause the drive to trip. Several network-monitoring variables are provided to for the DPL programmer to use.

DPL Variable	Description
NOFMESSAGES	Total number of messages sent and received. This value is used to generate the messages per second displayed in #20.50 on Unidrive, and #16.62 on Mentor II.
NOFOVERRUNS	Number of cyclic data overruns. An overrun will occur if a new synchronisation message is issued before a cyclic data messages have been transmitted. Increase the cyclic data period in #20.03 on Unidrive or #11.02 on Mentor II.
NOFLOSTMESSAGES	Number of receive messages lost. If messages are coming in too fast for the UD70/MD29AN to handle, this variable indicates how many have been lost. UD70 and MD29AN can usually process up to 4 or 5 messages per millisecond.
NOFRECONS	Number of network re-configurations. At start-up, the network may re-configure itself several times, but an increasing number of re-configurations may indicate wiring or noise problems on the network.
NOFEXENAKS	Number of excessive NAKS. This would indicate that a node is having trouble accepting messages from this node.
NOFDUPSYNCS	Number of duplicate synchronisation messages. A non-zero value indicates that two or more nodes have been configured to generate the synchronisation message.
NOFRETRIES	Number of retries. This indicates that outgoing messages are being re-transmitted. A high figure would suggest excessive network noise.

The variable NOFMESSAGES can be used to implement a network loss trip in DPL code. If the network connection is lost completely (a broken wire or connector unplugged), the program will see the change in messages received drop to 0.

After each calculation, if the number of messages received is 0, a counter is incremented by 1. After a 20 consecutive 10ms CLOCK tasks with the number of received messages at 0, the drive will trip, but the DPL program in the UD70 or MD29AN will continue to run as normal. (An ERROR task is not raised.) If the network recovers before the trip occurs, the counter will be reset to zero.

```
CLOCK   {
; read total number of messages
new_messages% = NOFMESSAGES
; if network has been lost
IF new_messages% - old_messages% = 0 THEN
  network_loss% = network_loss% + 1; increment counter
  ; if time-out period has elapsed, trip drive on tr65
  IF network_loss% = 20 THEN #10.38 = 65; use #10.35 on Mentor
ELSE
  network_loss% = 0; network OK, reset counter
ENDIF
; store the last read number of messages
old_messages% = new_messages%
}
```

4.7 Network Interruptions

If a Unidrive or Mentor II trips, then provided it is not due to a **CTNet** network error, the whole network will continue to operate. Cyclic data (such as speed or torque references) received by the UD75 or MD29AN will be passed to the drive, however, the data will have no effect while the drive is tripped. Care must be taken if the reference forms part of a closed loop system. It may be possible for the reference to saturate, and cause problems when the trip is reset. Data being transmitted from the tripped node will be frozen at the value when the trip occurred. The important point to note is that the rest of the network will continue to function normally.

If power is lost to a node, this does not prevent the network from continuing, or being re-started without the “dead” node. (Other nodes may need to be re-configured to ignore the “dead” node.) The NET+ are linked via the UD75 and MD29AN, as are the NET- pins, so the network is not physically broken. If a drive is physically removed from the system, one of two precautions must be taken:

- The network lead can be left connected to the UD75 or MD29AN.
- The Net+ and Net- wires for each cable must be wired into the same terminal on the **CTNet** connector.

5 Cyclic Data - “Easy Mode”

A straightforward method of setting up data transfer using **CTNet** is to use the “Easy Mode” set-up. “Easy Mode” uses only drive parameters to configure data transfer and parameter mapping at each node, and can be configured either from the drive keypad, or via the RS485 port using UniSoft or MentorSoft. Data transfer capabilities are limited, but no DPL code is required.

With “Easy Mode”, each drive can be considered to have IN and OUT data slots. Unlike systems such as Interbus-S, Profibus-DP and DeviceNet, **CTNet** does not have an overall master controller. Therefore, IN data is defined as data coming IN to a node, while OUT data is transmitted OUT from the node. The UD75 provides 3 IN and 3 OUT slots, while the MD29AN only provides 2 IN and OUT slots.

5.1 Parameter Mapping

The OUT data slots for a particular node are configured by specifying the source parameter within the source node, the target node address, and also the IN data slot to write to within the target node. This data is collectively known as the “slot mapping data”. The IN data slots receive data from other nodes on the network, and they only require mapping information to specify the target parameter for the incoming data. The mapping cannot be changed dynamically, as a reset of the UD70 or MD29AN must be performed to make the changes active.

5.1.1 Unidrive

The mapping for Unidrive and UD75 is set using menu 20 parameters. The mapping control parameters are shown in the table below.

Slot	Source/Destination Parameter	Destination Node
IN Slot 1	#20.10 (MMPP)	
IN Slot 2	#20.11 (MMPP)	
IN Slot 3	#20.12 (MMPP)	
OUT Slot 1	#20.05 (MMPP)	#20.04 (NNNSS)
OUT Slot 2	#20.07 (MMPP)	#20.06 (NNNSS)
OUT Slot 3	#20.09 (MMPP)	#20.08 (NNNSS)

The source and destination parameters are entered in the form MMPP, where MM is the menu number and PP is the parameter number. The destination node and slot is entered in the form NNNSS, where NNN is the destination node address, and SS is the IN slot to write to.

If any mapping parameter is set to an invalid value (target parameter is read-only or does not exist), the mapping will revert to 0 when the UD70 is reset. If a slot is not being used, it can be disabled by setting the mapping to 0.

5.1.2 Mentor II

The mapping for Mentor II and MD29AN set using menu 11 parameters. The mapping control parameters are shown in the table below.

Slot	Source/Destination Parameter	Destination Node
IN Slot 1	#11.07 (MMPP)	
IN Slot 2	#11.08 (MMPP)	
OUT Slot 1	#11.04 (MMPP)	#11.03 (NNNS)
OUT Slot 2	#11.06 (MMPP)	#11.05 (NNNS)

The source and destination parameters are entered in the form MMPP, where MM is the menu number and PP is the parameter number. The destination node and slot is entered in the form NNNS, where NNN is the destination node address, and S is the IN slot to write to.

If any mapping parameter is set to an invalid value (target parameter is read-only or does not exist), the mapping will revert to 0 the MD29AN is reset. If a slot is not being used, it can be disabled by setting the mapping to 0.

NOTE *For MD29AN with system file V2.5.4 or earlier, the format for destination node and slot is NNSS.*

5.2 Changing The Parameter Mapping

If any mapping parameters are changed, the new values only take effect after the UD70/UD75 or MD29AN has been reset. On Unidrive, a soft reset can be generated by setting #MM.00 to 1070 and pressing the red RESET button on the keypad. The MD29AN can be reset using the RESET button on the Mentor II keypad.

NOTE *If the drive is enabled then a reset sequence will not be generated.*

To keep any mapping changes permanently, (i.e. retained after power-down) the parameters must be stored.

5.3 Saving Unidrive Parameters

5.3.1 Menu 1 through 19

All parameters in these menus are saved in the EEPROM in the Unidrive. To initiate the non-volatile save sequence set #MM.00 (parameter 0 in any menu) to 1000 and press the red RESET button on the keypad.

5.3.2 Menu 20 and PLC Parameters

All menu 20 parameters and PLC parameters (_Pxx% and _Qxx%) are stored in the flash memory of the UD70 in the large option module. To initiate the non-volatile save sequence for these parameters, set #17.19

to 1. The UD70 will then save menu 20, clear #17.19 to zero, and reset. The UD75 **CTNet** interface will also be reset. These parameters can also be stored automatically at power-down, by setting #17.20 to 1. (This parameter change must be stored and the drive reset before it will take effect.)

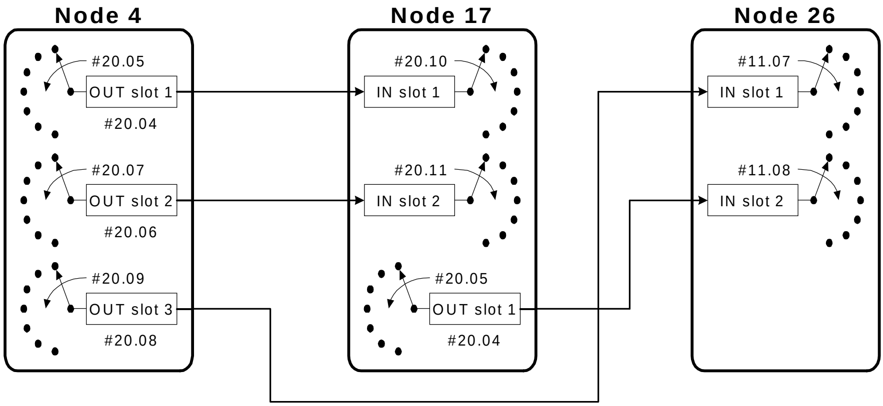
5.4 Saving Mentor II Parameters

To initiate the non-volatile save sequence, set #MM.00 to 1 and press RESET.

5.5 "Easy Mode" Example Network

The network below shows an example application which requires the post ramp speed reference from node 4 to be transmitted to node 17, parameter #1.22. Both drives are Unidrive. Torque reference #4.08 is also to be transmitted to node 17 to control the torque demand limit, #4.07. The feedback signals from two 4-20mA sensors connected to node 4 and node 17 are routed through to #18.11 in each drive, and both are transmitted to node 26, which is a Mentor II. The Mentor II should receive these signals in #15.06 and #15.07 respectively.

The network must transfer the data every 4ms, with node 26 generating the synchronisation message. The data rate must be 5.0 Mbits/sec.



The diagram above shows parameter links that need to be set up, and the table below shows the parameter settings required in each node.

Parameter		Function	Node 4	Node 17	Node 26
Unidrive	Mentor II				
#20.01	#14.05	Node address	4	17	26
#20.02	#11.01	Data rate	0	0	0
#20.03	#11.02	Synchronisation message	0	0	504
#20.04	#11.03	Output slot 1 destination	1701	2602	----
#20.05	#11.04	Output slot 1 source	122	1811	----
#20.06	#11.05	Output slot 2 destination	1702	----	----
#20.07	#11.06	Output slot 2 source	408	----	----
#20.08		Output slot 3 destination	2601	----	----
#20.09		Output slot 3 source	1811	----	----
#20.10	#11.07	Input slot 1 destination	----	121	1506
#20.11	#11.08	Input slot 2 destination	----	407	1507
#20.12		Input slot 3 destination	----	---	----

---- means that the parameter should be set to 0.

5.6 Mapping Conflicts

When the mapping parameters are set, care must be taken to ensure that there are no clashes with the mapping of the analogue and digital inputs within the drive. Neither the UD75 or MD29AN will indicate if there is a conflict of mapping parameters. This only applies to analogue and digital inputs, and incoming data from the **CTNet**.

If a parameter is written to from two different sources, the value of this parameter will depend entirely upon the scan times for the analogue or digital input and the **CTNet**. Further confusion may be caused due to the update rate of the display. A parameter may appear to be steady at a particular value, but occasionally glitch in the value will be seen. Internally, this value may be continuously changing between the two supplied incoming data values, leading to erratic drive behaviour.

Drive Input	**Unidrive Mapping Parameter**	**Mentor II Mapping Parameter**	**Drive Input**	**Unidrive Mapping Parameter**	**Mentor II Mapping Parameter**
Analogue 1	#7.10	#7.11	Digital 2	#8.13	#8.12
Analogue 2	#7.14	#7.12	Digital 3	#8.16	#8.13
Analogue 3	#7.18	#7.13	Digital 4	#8.19	#8.14
Analogue 4		#7.14	Digital 5	#8.21	#8.15
Speed		#7.15	Digital 6	#8.23	#8.16
EM Slot 1	#20.10	#11.07	Digital 7		#8.17
EM Slot 2	#20.11	#11.08	Digital 8		#8.18
EM Slot 3	#20.12		Digital 9		#8.19
Digital 1	#8.10		Digital 10		#8.20

To ensure that there are no mapping conflicts, check that each Unidrive mapping parameter and each Mentor II mapping parameter has a different value programmed. Analogue and digital inputs can be de-programmed by setting the mapping to 0.

6 Cyclic Data - “Advanced Mode”

A more powerful mode of configuring cyclic data transfer is the “Advanced Mode”. “Advanced Mode” requires DPL code to configure and control the transfer of data between nodes. Data transfer can be up to 80 parameters per cycle using both fast and slow cyclic data channels.

“Advanced Mode” uses the 32 bit _Rxx% and _Sxx% registers in the UD70 and MD29AN as source and destination locations respectively for transmitting data. To set up fast and slow cyclic data transfer using the _Rxx% and _Sxx% registers, “data links” must first be defined for both the fast and slow cyclic data channels. Each link is configured to transmit a defined set of consecutive _Rxx% registers in the source node to a defined set of consecutive _Sxx% registers in the target node. This is known as “binding” registers together.

A System Programming Tool (SYPT) for configuring **CTNet** networks from a PC is available from Control Techniques Software and System Products Division. (See Chapter 12, Network Repeaters, “**CTNet Software and Drivers SYPT Workbench**”).

6.1 Fast and Slow Cyclic Data Channels

The fast cyclic data links are defined using the CYCLICDATA FAST command and the slow cyclic data links with the CYCLICDATA SLOW command. The data link structures must be placed before the INITIAL task when compiling the DPL program.

The cyclic data rate is set using #20.03 on Unidrive, and #11.02 on Mentor II. The slow cyclic data rate is set as an integer multiple of the fast cyclic data rate.

```
CYCLICDATA FAST:   {
num_reg%,  node%,  source_reg%,  dest_reg%;  data link 1
num_reg%,  node%,  source_reg%,  dest_reg%;  data link 2
.....
num_reg%,  node%,  source_reg%,  dest_reg%;  data link 10
}
CYCLICDATA SLOW:   {
num_reg%,  node%,  source_reg%,  dest_reg%;  data link 1
num_reg%,  node%,  source_reg%,  dest_reg%;  data link 2
.....
num_reg%,  node%,  source_reg%,  dest_reg%;  data link 10
}
```

Object	Range	Description
num_reg%	1 to 20	The number of consecutive _Rxx% registers to be transmitted to the remote node.
node%	1 to 255	Target node address. A “Global Address” of 0 allows a node to transmit to all other nodes on the network. The allocation of _Sxx% registers in all nodes must be carefully thought out before using global addressing.
source_reg%	_R00% to _R79%	Defines the first _Rxx% register in the source node to be transmitted.
dest_reg%	_S00% to _S79%	Defines the first _Sxx% register in the target node to be written to.

There are three limitations in defining the links between nodes, and the number of registers to be transmitted.

1. A maximum of 10 separate links can be defined for a single node. This total includes both fast and slow data links.
2. No individual link may transfer more than 20 consecutive registers.
3. A maximum of 80 registers may be transferred from a single node. This total includes registers being transmitted on both the fast and slow data links.

Each link runs on successive token rotations. The first defined link is transmitted on the first token rotation after the “synchronisation message”, the next link on the next token rotation and so on.

6.2 Overloading Network Nodes

Care must be taken when configuring a network to ensure that no node on the network gets overloaded with incoming messages. UD70 and MD29AN nodes are capable of processing up to 4 or 5 messages every millisecond, and may produce very strange behaviour if there are too many incoming messages.

A good example where overloading a node with messages can occur is when one node is assigned to monitor the network. All other nodes will continuously increment an internal counter, and use cyclic data to transmit the counter value to the “monitoring” node, to indicate that they are still operating on the network. On a large network, say 50 drives, this can mean that the monitoring node suddenly receives 50 messages in very quick succession, and is unable to process them quickly enough. The effect of the lost messages could indicate to the monitoring node that several drives are not functioning, when this is not actually the case, and all nodes are operating OK.

The correct way to implement a network monitoring node is to use the CHECKNODE() command in the BACKGROUND task, and use the monitoring node to interrogate each node in turn, and get the operational status of each node. (Refer to Section on **Non-Cyclic Data** - "Network Cycle Time ").

6.3 Editing The Fast and Slow Cyclic Data Structures

It is possible with **CTNet** to dynamically change the fast and slow data link structures from within the DPL program using the EDITFASTLINK() and EDITSLOWLINK() commands. These commands can edit one data link within a data structure at a time. This could be useful in an application on a production line where different grades of a material are produced. The network may need to re-configure itself to bypass or include certain sections of the plant, depending on the material type selected.

The command works by completely redefining the set-up for the data link within the fast or slow cyclic data structure.

```
status% = EDITFASTLINK(index%, num_reg%, node%, source_reg%, dest_reg%)
status% = EDITSLOWLINK(index%, num_reg%, node%, source_reg%, dest_reg%)
```

Object	Range	Description
status%	1 - Link OK 5 - Error	Indicates whether the data structure was modified successfully or not.
index%	1 to 10	Specifies the line number in the data structure to be modified.
num_reg%	1 to 20	Specifies the number of consecutive _Rxx% registers to be transmitted.
node%	1 to 255	Target node address.
source_reg%	_R00% to _R79%	Specifies the first _Rxx% register in source node to be transmitted.
dest_reg%	_S00% to _S79%	Specifies the first _Sxx% register in target node to be written to.

6.4 Control Word (Unidrive Only)

The Control Word is an efficient way of remotely controlling the motion of a drive. Each bit in the control word has a particular function, and provides a method of controlling the function of the drive (RUN, JOG, etc.) with a single data word.

The control word is addressed as #90.11, and is a virtual parameter within the UD70.

b15	b14	b13	b12	b11	b10	b9	b8
M6	M5	#18.33	M3	M2	M1	M0	#18.32

b7	b6	b5	b4	b3	b2	b1	b0
#18.31	#1.46	#1.45	Trip	#6.32	#6.31	#6.30	#6.15

The bits shown as “Mx” are individual mask bits which allow the corresponding "bx” to be masked. The “Trip” bit will cause a “tr52” trip when set to 1. Parameters #18.31 to #18.33 are general user parameters and do not have mask bits.

NOTE *If an Mx bit is reset to 0, the bit parameter that it masks will remain at the previous value set.*

All direct control of the sequencing bits (#6.30 - #6.32) from digital inputs must be disabled before the control word will can be used. (Set #8.16, #8.19 and #8.21 to another value or 0.) The sequencing bits have the following functions when set to 1.

Parameter	Sequencing bit	PLC Mode (#6.04 = 3)	Wire-proof PLC Mode (#6.04 = 4)
#6.15	Enable	Enable	Enable
#6.30	0	Run	Run Forward
#6.31	1	Jog	Jog
#6.32	2	Reverse	Run Reverse

ENABLE - the display will show "**Inh**" when set at 0, and depends on #6.30 and #6.32 when set to 1. Setting #6.15 to 0 overrides #6.30 and #6.32, and immediately disables the drive. The motor will coast to rest if it is running when the drive is disabled.

JOG - the jog bit must be set, along with the appropriate run and direction signals.

It is not possible to transmit the control word directly from one drive to other drives on the network. If #90.11 is used as a source parameter, it will return the status word, not the control word. To copy the control signals of one drive to another via **CTNet**, DPL code must be written to construct the control word in a temporary register within the master node. When complete, the control word is moved to register _Rxx%, and transmitted directly to #90.11 in other drives on the next transmission cycle.

In the example below, the control word is constructed using the enable, run, direction and trip status parameters in the drive. This can be transmitted either by defining a CYCLICDATA FAST or CYCLICDATA SLOW link, or using “Easy Mode” set-up with the source parameter as #72.01.

```
ENCODER   {
_P01%.0   = #6.15  ; check the enable bit
_P01%.1   = #6.30  ; check the run/stop bit
_P01%.3   = #6.32  ; check the direction bit
_P01%.9   = 1      ; set mask bit M0
_P01%.10  = 1      ; set mask bit M1
_P01%.12  = 1      ; set mask bit M3
IF #10.0  = 0 THEN ; if the drive is tripped
  _P01%.4 = 1      ; set the trip bit 4
ELSE
  _P01%.4 = 0      ; else reset the trip bit 4
ENDIF              ; if drive is enabled, healthy and running
                   ; in reverse, _P01% = 0x0E0B or 3595
_R01% = _P01%      ; move control word to output register
}
```

6.4.1 Control Word Bit Mapping Functions

Bit	Unidrive - UD75	
	Parameter	**Description**
0	#6.15	Drive enable
1	#6.30	Sequencing bit 0
2	#6.31	Sequencing bit 1
3	#6.32	Sequencing bit 2
4	TRIP	Drive Trip (tr52)
5	#1.45	Preset select 0
6	#1.46	Preset select 1
7	#18.31	Application bit
8	#18.32	Application bit
9	M0	Mask bit 0
10	M1	Mask bit 1
11	M2	Mask bit 2
12	M3	Mask bit 3
13	#18.33	Application bit
14	M5	Mask bit 5
15	M6	Mask bit 6

The bits shown as “Mx” are individual mask bits which allow the corresponding "bx” to be masked. The “Trip” bit will cause a “tr52” trip when set to 1. Parameters #18.31 to #18.33 are general user parameters and do not have mask bits.

6.4.2 Example Unidrive Control Word Values (PLC Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Value	Action
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0x0200	Drive disable
0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0x1E01	Enabled + stopped
0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0x1E03	Enabled + run fwd
0	0	0	1	1	1	1	0	0	0	0	0	1	0	1	1	0x1E0B	Enabled + run rev
0	0	0	1	1	1	1	0	0	0	0	0	1	1	1	1	0x1E07	Enabled + jog rev

6.4.3 Example Unidrive Control Word Values (Wire Proof PLC Mode)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Value	Action
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0x0200	Drive disable
0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0x1E01	Enabled + stopped
0	0	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0x1E03	Enabled + run fwd
0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	1	0x1E09	Enabled + run rev
0	0	0	1	1	1	1	0	0	0	0	0	1	1	0	1	0x1E0C	Enabled + jog rev

6.5 Mapping Conflicts

When the mapping parameters for the **CTNet** polled data channels are set, care must be taken to ensure that there are no clashes with the mapping of the analogue and digital inputs within the drive. Neither the UD75 or MD29AN will indicate if there is a conflict of mapping parameters.

See Section 5.6, Mapping Conflicts.

6.6 Status Word

The status word is an efficient way of determining the status of the drive. Each bit in the status word indicates the status of a particular function of the drive, e.g. at speed, tripped, etc., and provides a quick method of transferring several bits of data using a single data word.---Unidrive.

The status word on Unidrive is #10.40, and can be mapped to any OUT slot by setting the appropriate mapping parameter to 1040.

b15	b14	b13	b12	b11	b10	b9	b8
X	#10.15	#10.14	#10.13	#10.12	#10.11	#10.10	#10.09

b7	b6	b5	b4	b3	b2	b1	b0
#10.08	#10.07	#10.06	#10.05	#10.04	#10.03	#10.02	#10.01

Bit	Unidrive - UD75	
	Parameter	Description
0	#10.01	Drive healthy
1	#10.02	Drive running
2	#10.03	Zero speed
3	#10.04	Running at or below minimum speed
4	#10.05	Below set speed
5	#10.06	At speed
6	#10.07	Above set speed
7	#10.08	Load reached
8	#10.09	In current limit
9	#10.10	Regenerating
10	#10.11	Dynamic brake active
11	#10.12	Dynamic brake alarm
12	#10.13	Direction commanded
13	#10.14	Direction running
14	#10.15	Mains Loss
15	X	Not used

6.6.1 Mentor II

The status word on Mentor II can be read from #90.11. It cannot be mapped to an output slot, as Mentor II mapping parameters are limited to a value of 1999. The value can only be transferred to another node using "Advanced Mode" data links or non-cyclic data transfers.

b15	b14	b13	b12	b11	b10	b9	b8
#11.21							

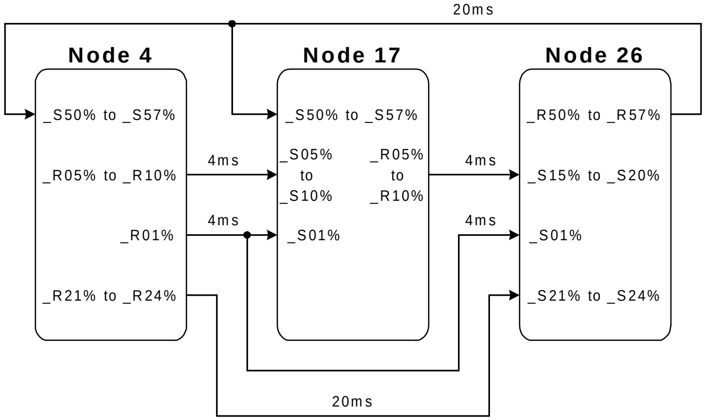
b7	b6	b5	b4	b3	b2	b1	b0
#10.25							

Bit	Mentor II - MD29AN	
	Parameter	Description
0 to 7	#11.21	LED status
8 to 15	#10.25	Current trip code

6.7 "Advanced Mode" Example Network

In the example given below, it is assumed that each UD70 has a program installed to update the _Rxx% registers, and read from the _Sxx% registers.

The diagram below shows the information transfer requirements of an application. "Advanced Mode" must be used to transfer the register groups, as the number of registers exceeds 3 from any node, and two cyclic data rates are required.



The network configuration will use node 26 to generate the synchronisation message, with a fast cyclic data rate of 4ms and a slow cyclic data rate of 20ms. The data rate will be 5.0 Mbits/sec.

Parameter	Function	Node 4	Node 17	Node 26
#20.01	Node address	4	17	26
#20.02	Data rate	0	0	0
#20.03	Synchronisation message	0	0	504

The cyclic data links required for this application are shown in the table below.

Source node	Target node(s)	First source register	Number of registers	First target register	Cyclic data rate
4	17	_R05%	6	_S05%	Fast
17	26	_R05%	6	_S15%	Fast
4	17 and 26	_R01%	1	_S01%	Fast
4	26	_R21%	4	_S21%	Slow
26	4 and 17	_R50%	8	_S50%	Slow

Each link has to be defined in the DPL code in the source node. Global addressing will be used to transmit from node 26 to nodes 4 and 17. As there are no other nodes on the network this will not cause a problem. The code for each link must be declared in the DPL program for each node.

Node 4	CYCLICDATA FAST:
	{
	6, 17, 5, 5
	1, 0, 1, 1
	}
		CYCLICDATA SLOW:
		{
		4, 26, 21, 21
		}

Node 17	CYCLICDATA FAST:
	{
	6, 26, 5, 15
	}	

Node 26		CYCLICDATA SLOW:
		{
		8, 0, 50, 50
		}

NOTE

If the network is configured using the SYPT Workbench software (See Chapter 12, Network Repeaters.), the cyclic data links are configured graphically, and added automatically to DPL programs when the programs are compiled. Consequently, when editing a DPL program with SYPT, the CYCLICDATA FAST and CYCLICDATA SLOW commands are not be included with the DPL program code. If the DPL Toolkit is being used, these commands MUST be included at the top of the program.

7 Non-Cyclic Data

The non-cyclic data channel allows true “peer-to-peer” communications, where any node can access any parameter or virtual parameter from any other network node. This function is particularly useful for transmitting infrequent events around the network.

A typical example might be using a switch to engage a pinch roller when the line is ready to start a production run. Response time to changing the switch is not critical, since it may take several seconds for the roller to engage. Once the roller is engaged, there is no need to transmit any information about the switch over the network until the switch changes. This provides a large reduction in the network bandwidth used for this signal, and increases the bandwidth available for other data signals. This offers a clear advantage over PLC-based systems where the main processor has to scan the switch continuously.

Non-cyclic network transactions are serviced by the network controller every 8ms, thus limiting the number of non-cyclic messages that be handled by a node to 125 per second. Transmission of data will occur when the node next receives the token, and the network is available for transmission. Non-cyclic data transfers are implemented using the following DPL system commands:

```
RDNET()  WRNET()  NETSTATUS()
NETREPLY()  CHECKNODE()
```

7.1 Non-Cyclic Parameter Read

This command allows a node to read in a parameter from any other node on the network. This is the **CTNet** equivalent to the ANSIREAD command for the RS485 port on UD70 and MD29.

```
status%  =  RDNET(node%, menu%, param%, waitcode%)
```

Object	Range	Description
status%	-6 to +6	Returns a value to indicate the nature of the response to the RDNET() command. Response codes are different, depending on whether automatic or manual time-out is used. “**status%**” does NOT return the data value from an RDNET() command.
node%	1 to 255	Defines the target node address.
menu%	0 to 99	Defines the menu number in the target node. Range depends on the drive type of the target node.

Object	Range	Description
param%	0 to 99	Defines the parameter number in target node. Range depends on the drive type and menu number being interrogated.
waitcode%		Defines the number of milliseconds to wait for confirmation that the message has been completed. The time-out period must be > 16ms.

The operating system of the UD70 and MD29AN only polls the network controller chip for incoming messages every 8ms. In the worst case, there may be up to 16ms delay before the response is received:

- Up to 8ms between the transmission of an RDNET() command, and the remote node recognising the message.
- Up to 8ms before the source node re-polls the network controller and sees the response.

7.1.1 Automatic Message Time-out

With automatic message time-out, the value set for **waitcode%** sets the maximum number of milliseconds that the system will wait for a response to arrive from a remote node, before timing out, and indicating an error.

```
status%  =  RDNET(node%, menu%, param%, waitcode%)
```

NOTE

*When **waitcode% > 0**, the DPL task will stop and wait for a response, or the specified time-out period to elapse, before continuing execution. For this reason, RDNET() commands with **waitcode% > 0** should **ONLY** be used in the INITIAL or BACKGROUND tasks.*

Node Status	status %	Description
Command Complete	1	The previous RDNET() or WRNET() command has been completed successfully. Use the NETREPLY() command to return the node to the Node Ready state.
Node Time-out	4	No response to the previous RDNET() or WRNET() was received from the network before the time-out period elapsed.
Global Address Error	6	Global addressing cannot be used with the RDNET() command.
Parameter Does Not Exist	-2	The specified parameter does not exist in the target node.
Parameter is Write-Only	-4	The specified parameter in the target node is a write-only parameter, and cannot be read.

Node Status	status %	Description
I/O Box Comms Error	-6	The communications link to the remote I/O box on the target node is not running.

7.1.2 Manual Message Time-out

To use manual message time-out, **waitcode%** should be set to 0. Message time-out monitoring MUST be implemented within the DPL program, or a node may experience "communications lock-out" if an error occurs.

```
status%  =  RDNET(node%, menu%, param%, 0)
```

The main advantage of disabling the automatic time-out feature is that the DPL task execution does not stop when the RDNET() command is issued. Execution time is not wasted while waiting for responses, and consequently, this allows the RDNET() command to be used in the CLOCK and ENCODER tasks.

Node Status	status %	Description
Message Transferred	2	The previous RDNET() or WRNET() command was transferred successfully to the network controller. Use the NETSTATUS() command to check for Command Complete status.
Node Busy	3	The previous RDNET() or WRNET() command is still in progress. The network is not ready to accept the new RDNET() or WRNET() command.
Global Address Error	6	Global addressing cannot be used with the RDNET() command.

7.2 Non-Cyclic Parameter Write

This command allows a node to write data to a parameter in any other node on the network. This is the **CTNet** equivalent to the ANSIWRITE command for the RS485 port on UD70 and MD29.

```
status%  =  WRNET(node%,  menu%,  param%,  value%,dpos%,  waitcode%)
```

Object	Range	Description
status%	-6 to +6	Returns a value to indicate the response received to the WRNET command. (See table below.)
node%	0 to 255	Defines the target node address. 0 is used as the "global address", where all nodes will receive the transmitted data

Object	Range	Description
menu%	1 to 20	Defines the menu number in the target node. Range depends on the drive type of the target node.
param%	0 to 99	Defines the parameter number in target node. Range depends on the remote drive type and menu number.
value%	$\pm 2^{31}$	Data value to be written to destination parameter in the target node. Any program variable or node parameter can be defined as the source of the data to be transmitted. Only integer values are allowed.
dpos%	0 to 3	Defines the position of the decimal point in the target parameter for scaled data.
waitcode%		Defines the number of milliseconds to wait for confirmation that the message has been completed. The time-out period must be >16ms.

The value returned in the variable "status%" depends on the method of message time-out being used in the program.

7.2.1 Automatic Message Time-out

With automatic message time-out, the value set for **waitcode%** sets the maximum number of milliseconds that the system will wait for a response to arrive from a remote node, before timing out, and indicating an error.

```
status% = WRNET(node%, menu%, param%, value%, dpos%, waitcode%)
```

NOTE *when **waitcode% > 0**, the DPL task will stop and wait for a response, or the specified time-out period to elapse, before continuing execution. For this reason, WRNET() commands with **waitcode% > 0** should **ONLY** be used in the INITIAL or BACKGROUND tasks.*

Node Status	status%	Description
Command Complete	1	The previous RDNET() or WRNET() command has been completed successfully. Use the NETREPLY() command to return the node to the Node Ready state.
Node Time-out	4	No response to the previous RDNET() or WRNET() was received from the network before the time-out period elapsed.
Global Address Error	6	Global addressing cannot be used with the RDNET() command.

Node Status	status%	Description
Parameter Does Not Exist	-2	The specified parameter does not exist in the target node.
Parameter is Read-Only	-3	The specified parameter in the target node is a read-only parameter, and cannot be written to.
Parameter Over-Range	-5	The data value being written to the specified parameter in the target node out of range. No data value will be transferred to the target paramter.
I/O Box Comms Error	-6	The communications link to the remote I/O box on the target node is not running.

7.2.2 Manual Message Time-out

To use manual message time-out, waitcode% should be set to 0. Message time-out monitoring MUST be implemented within the DPL program, or a node may experience “communications lock-out” if an error occurs.

```
status% = WRNET(node%, menu%, param%, value%, dpos%, 0)
```

The main advantage of disabling the automatic time-out feature is that the DPL task execution does not stop when the RDNET() command is issued. Execution time is not wasted while waiting for responses, and consequently, this allows the RDNET() command to be used in the CLOCK and ENCODER tasks.

Node Status	status %	Description
Message Transferred	2	The previous RDNET() or WRNET() command was transferred successfully to the network controller. Use the NETSTATUS() command to check for Command Complete status.
Node Busy	3	The previous RDNET() or WRNET() command is still in progress. The network is not ready to accept the new RDNET() or WRNET() command.
Global Address Error	6	Global addressing cannot be used with the RDNET() command.

7.3 Network Status

The NETSTATUS command returns an integer value to indicate the status of the network. This command MUST be used if automatic time-out is not being used, i.e. when **waitcode%** = 0.

```
status%  =  NETSTATUS(0)
```

7.3.1 Automatic Message Time-out

The NETSTATUS(0) command is not required with the RDNET() and WRNET() commands when using the automatic time-out mode. The success or otherwise of the message will be returned in the status% variable from the RDNET() or WRNET() commands.

7.3.2 Manual Message Time-out

The NETSTATUS(0) command is used to check for incoming messages from other nodes on the network, in response to RDNET() and WRNET() commands. If an error has occurred, it will indicate what the error was, and the user can use this value to determine future network activity.

Node Status	status %	Description
Node Idle	0	The node status has been returned to idle, and is available for the next RDNET() or WRNET() command.
Command Complete	1	The previous RDNET() or WRNET() command has been completed successfully. Use the NETREPLY() command to retrieve data, and return the node to the NODE_IDLE status.
Node Busy	3	The previous RDNET() or WRNET() command is still in progress. The network is not ready to accept a new RDNET() or WRNET() command.
Node Time-out	4	No response to the previous RDNET() or WRNET() was received from the network before the time-out period elapsed. (Automatic time-out only.)
Internal Error	5	No internal network controller buffers are free.
Parameter Does Not Exist	-2	The specified parameter does not exist in the target node.
Parameter is Read-Only	-3	The specified parameter in the target node is a read-only parameter, and cannot be written to.
Parameter is Write-Only	-4	The specified parameter in the target node is a write-only parameter, and cannot be read.
Parameter Out Of Range	-5	The data value being written to the specified parameter in the target node out of range. No data value will be transferred to the target parameter.

Node Status	status %	Description
I/O Box Comms Error	-6	The communications link to the remote I/O box on the target node is not running.

7.4 Return Value From Non-Cyclic Parameter Read

The NETREPLY(0) command is used after the RDNET() to receive the data value from the target node. This command is the **CTNet** equivalent to the ANSIREPLY command for the RS485 port on UD70 and MD29.

```
value%  =  NETREPLY(0)
```

where **value%** = 32-bit integer data from previous RDNET() operation.

7.4.1 Automatic Message Time-out

The DPL program should ensure that the NETREPLY(0) command is called immediately after a successful RDNET() command. If the data is not retrieved before the next RDNET() command is issued, the value will be over-written by the new command.

7.4.2 Manual Message Time-out

The NETREPLY(0) command MUST be used after an RDNET() or WRNET() command to return the node to the idle state. If the RDNET() command was successful, (**status%** = 1) **value%** will contain the data value received.

If an error occurred during the message transmission, (**status%** <> 1) **value%** will return the previous data value. The NETREPLY(0) command must still be called before the next RDNET() or WRNET() commands can be issued.

7.5 Check If Node Exists

This allows a DPL program to determine whether a target node is present on the network or not. If problems are experienced reading from or writing to a remote node, this can be used to check that the remote node is present on the network.

```
status%  =  CHECKNODE(node%)
```

Object	Range	Description
status%	0 or 1	Returns a value to indicate the presence or not of a node with a particular node address. 0 - Node address is not present on the network. 1 - Node address is present on the network.
node%	1 to 255	Specifies the node address to check for on the network.

7.6 Handling Message Time-outs (Examples)

7.6.1 Automatic Message Time-out

The example code below shows a simple method of implementing an RDNET() command using automatic time-out. The BACKGROUND task ensures that there will be no problems with task over-run errors.

```
BACKGROUND   {
; issue the RDNET command
message_status% = RDNET(node%,menu%,par%,waitcode%)
; message was completed successfully
IF message_status% = 1 THEN
  value% = NETREPLY(0)  ; get returned data value
  #18.11 = value%       ; transfer to #18.11
; error with message
ELSE
  value% = NETREPLY(0)  ; return node to idle
ENDIF
}
```

To use the above code, the user must ensure that the variables **node%, menu%, par%** and **waitcode%** are loaded with the required values before the code is executed. Once the RDNET() command is issued, the BACKGROUND task waits until either a response is received, or the time-out period elapses.

If the Command Complete signal is received **(message_status% = 1)**, the value is retrieved using the NETREPLY(0) command, and the node returned to the idle state. If an error is returned, or the time-out period elapses, the NETREPLY(0) command is issued to return the node to the idle state.

Additional error-handling routines can be added to trap and analyse errors in **message_status%** to make the code more tolerant of network errors.

7.6.2 Manual Message Time-out

The example code below shows a simple method of implementing an RDNET() command using manual time-out. The CLOCK task ensures that the code is executed regularly.

```
CLOCK   {
; check the status of the network
net_status% = NETSTATUS(0)
; if node is NET_IDLE, issue a new RDNET command
IF net_status% = 0 THEN
  status% = RDNET(17,18,11,0) ; issue the RDNET command
  time_out% = TIME + wait_time% ; add time-out period to
  ; current run-time clock
  ; check to see if time_out% has rolled, but TIME has not
  IF SGN(time_out%) = -1 AND SGN(TIME) = 1 THEN
    roll_over% = 1 ; set roll_over% flag
  ELSE
    roll_over% = 0 ; reset roll_over% flag
  ENDIF
; if node is NET_OK, get returned data value from remote node
ELSEIF net_status% = 1 THEN

  value% = NETREPLY(0)
  status% = 0
  read_messages% = read_messages% + 1

; if TIME roll-over is about to occur, prevent execution of
; time-out branch
ELSEIF roll_over% = 1 THEN
  ; when TIME has also rolled over, reset roll_over% flag
  ; and allow time-out calculation branch to be executed
  IF SGN(time_out%) = SGN(TIME) THEN roll_over% = 0

; if the node is NET_WAITING or the time-out period has elapsed
ELSEIF net_status% <> 3 or TIME > time_out% THEN
  x% = NETREPLY(0) ; reset the node to NET_IDLE
  read_errors% = read_errors% + 1
  value% = 0 ; set data value to 0
ENDIF
}
```

By only using this section of code once in the program, it is not possible to issue a new RDNET() command until the previous command is complete. The advantage is that the CLOCK task will not pause while waiting for a response.

To use the code, the user must ensure that the variables **node%, menu%, par% and wait_time%** are loaded with the required values before the code is executed. Once the RDNET() command is issued, the node status is checked every cycle. As long as the Node Busy state **(node_status% = 3)** is returned, nothing is done.

If the Command Complete signal is received (**node_status% = 1**), the value is retrieved using the NETREPLY(0) command, and the node returned to the idle state, ready for the next CLOCK task execution. If an error is returned, or the time-out period elapses, the NETREPLY(0) command is issued to return the node to the idle state.

Additional error-handling routines can be added to trap errors in **message_status%** or **node_status%** to make the code more tolerant of network errors.

8 Network Timing

The timing of the network depends partly on the propagation delay (T_p) of the signal along the network cable. This can be calculated as:

$$T_p = \frac{LineLength * 10^6}{Propagation\ Speed}$$

Variable	Units	Description
T_p	µs	Signal propagation delay in the transmission line.
Line length	metres	Length of network cable.
Propagation speed	metres/ sec	Signal propagation speed in the cable.

For the CT Net cable, the propagation speed is $1.98*10^8$ metres/second. For a network with a cable length of 200m, the cable propagation delay would be approximately 1µs.

8.1 Cyclic Data: Node-to-Node

The time taken to transmit a cyclic data message between nodes can be calculated using the equation below:

Data Rate	2.5 Mbits/sec	5 Mbits/sec
Time T_{cd}(µs)	$167.4 + 17.6*R + 5*T_p$	$83.7 + 8.8R + 5T_p$

where: R = number of contiguous registers transferred

T_p = propagation delay for the transmission line

8.2 Cyclic Data: Network Broadcast

The time taken to transmit a broadcast message from one node to all other nodes can be calculated using the equation below:

Data Rate	2.5 Mbits/sec	5 Mbits/sec
Time T_{bc}(µs)	$103.4 + 17.6*R + T_p$	$65.3 + 8.8R + T_p$

8.3 Non-Cyclic Data: RDNET and WRNET

The time taken to transmit a message between nodes using the non-cyclic data commands can be calculated from the equation below:

Data Rate	2.5 Mbits/sec	5 Mbits/sec
Time Tnc(µs)	345.6 + 10Tp	174 + 10Tp

NOTE *Non-cyclic data commands are serviced every 8ms by the ARCNET ASIC. The time delay before they are transmitted is indeterminate.*

8.4 Synchronisation Telegram

The synchronisation telegram is transmitted at the start of each cyclic data period. This takes a finite period of time to be transmitted.

Data Rate	2.5 Mbits/sec	5 Mbits/sec
Time $T_{st}(\mu s)$	92	59.2

8.5 Network Cycle Time

The total cycle time to transmit all the fast rate cyclic data can be calculated as:

Cycle Time (μs) = $T_{st} + N * T_{cd}$

where:

T_{st} = time to transmit the synchronisation telegram in μsecs.

N = number of nodes on the network.

T_{cd} = time to transmit each node-to-node cyclic message in μsecs.

8.6 Example Network Calculations

A **CTNet** network has 50 nodes, and is running at 5 Mbits/sec. Four registers are transmitted from each node to another node on the network using the fast cyclic data channel. Calculate the total cycle time, and set the cyclic data rate accordingly.

T_{cd} = 83.7 + 8.8 * 4 + 5 * 1 = 242.8 μs

Cycle Time = 92 + (123.9 * 50) = 6287 μs

= 6.287 ms

Typically, the network should be configured to run at approximately 80% load with only the cyclic data being transmitted, leaving 20% for the non-cyclic data. Therefore, the synchronisation telegram could be set to run as fast as 8ms.

9 General Information

9.1 Network Cycle Time

Consider a network with 50 nodes, with each node transmitting four registers to the next node on the line. The network data rate is 5 Mbits/sec, and the network length is 50m. The theoretical minimum network cycle time is 6.3ms.

9.2 System Propagation Delays

The worst case propagation delay for an **CTNet** network can be calculated by considering all delays through each stage of the system. Consider a system with 50 nodes running at 5 Mbits/sec, each node having four data words, with one node having an I/O Box connected to the UD70. What delay may be seen between a switch being operated at the I/O Box on one drive to a change being reflected within another drive?

The delays present within the system are:

- I/O Box update delay
- **CTNet** cyclic data time
- **CTNet** module to drive update time

The update rate for the I/O Box can be synchronised to the ENCODER or CLOCK tasks. This means that the fastest possible update time is 5ms. (For this example, the I/O Box is synchronised to the ENCODER task, running every 5.52ms.) In the worst case, the buffer will change just after the I/O Box has scanned the inputs and sent the data to the UD70, giving a maximum delay of 5.5ms.

The minimum cyclic data rate for the network is 6.3ms, so the fast cyclic rate would be set to 8ms. The **CTNet** network is not synchronised to the internal drive control loops. In the worst case, the updated value from the I/O Box will arrive just after the cyclic data is sent, resulting in up to 8ms delay.

When the data arrives at the target drive, it can take up to 2ms for data to be transferred to the UD70. For the example network, worst case propagation delay would be 5.5 + 8 + 2 = 15.5ms.

10 Diagnostics

10.1 Determining If Network Is Active On Unidrive

To determine if the **CTNet** network is active, check the value being displayed in #20.50 on Unidrive, and #16.62 on Mentor II. If the network is healthy and the node is communicating, the number of messages being processed per second wil be displayed.

If the network has stopped communicating with the node, or the network connection has been lost, a value of -1 will be displayed.

If the value displayed is -2, this indicates a problem that the node was unable to initialise itself.

10.2 Cannot Establish CTNet Connection

1. Ensure that the correct type of cable is used. **CTNet** runs at very high data rates, and consequently requires high quality cable. Performance cannot be guaranteed if the specified Control Techniques **CTNet** cable is not used.
2. 78Ω terminating resistors MUST be fitted. Unpredictable network behaviour may result if resistors are not fitted, due to signal reflections. The noise immunity of the network will also be severely reduced without proper line termination resistors.
3. Ensure that the correct system file has been downloaded. On Unidrive and UD70, this should be “UD70NET.SYS”, while on Mentor II and MD29AN, it should be “M2NET.SYS”.
4. If SYPT is being used to configure the network, ensure that V2.6.3 or later system files have been downloaded to the UD70 and MD29AN.
5. Check the node configuration settings. Every node must have a unique node address (#20.01 and #14.05), all node data rates must be identical (#20.02 and #11.01), and only one drive must be set to generate the synchronisation message (#20.03 and #11.02).
6. Ensure that the baud rate selected is OK for the length of network cable installed. If necessary, try reducing the network data rate AT ALL NODES to establish communications. Repeaters may be required to extend the network length for a given data rate. See Section 3.5, Network Limitations.
7. **CTNet** nodes are only capable of driving up to 15 nodes each. If more than 15 nodes are connected to a single section of cable, network repeaters will be required. See Chapter 12, Network Repeaters.
8. On Mentor II, parameters #11.01 to #11.10 are used to configure **CTNet**, and cannot be used to define Menu 0. These parameters should all be set to 0 before the MD29AN card is installed. Existing

parameter values may cause unexpected mapping of data to drive parameters.

10.3 Drive Trip Codes

If certain errors occur, the Unidrive or Mentor II will trip and show the trip code in the upper window.

Unidrive	Mentor II (#10.35)	Error
tr52		Indicates that the trip originated from the setting of bit 4 in the Unidrive control word to 1.
tr56	A29 (56)	The UD70 or MD29AN does not contain the correct operating system. Download the “UD70NET.SYS” or “M2NET.SYS” from the MD29 DPL Toolkit
tr57	A29 (57)	An illegal operating system call has been made. The command does not exist, or is not being used correctly.
tr60	A29 (60)	There is a hardware failure on the UD75 or MD29AN.
tr61	A29 (61)	The data rate or node address settings are out of range.
tr62	A29 (62)	The data rate does not match the data rate of the rest of the network.
tr63	A29 (63)	The node address already exists on the network.
tr64	A29 (64)	The cyclic data rate setting is invalid.

To reset the Unidrive using the **CTNet** network, write a value of 100 to #10.38 using the non-cyclic data channel. Digital input control of this parameter must be disabled. (Refer to Unidrive Manual).

To reset the Mentor II using the **CTNet** network, write a value of 255 to #10.35. (Refer to Mentor II Manual).

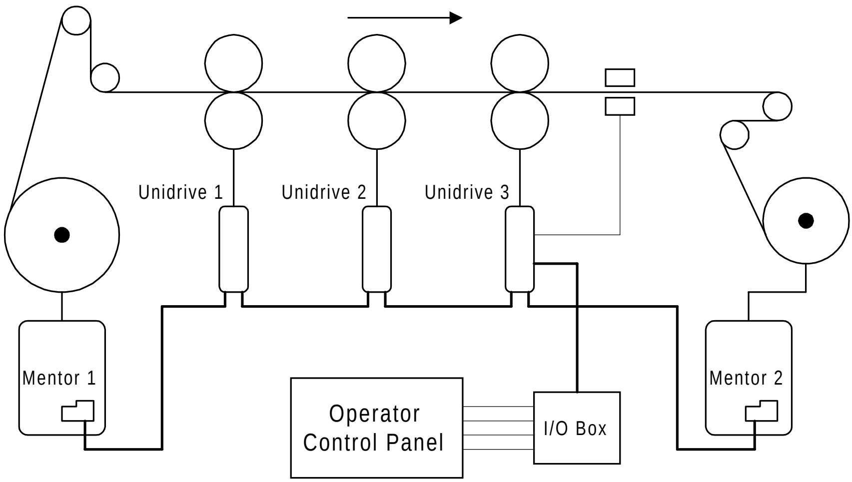
11 Example Application

11.1 System Overview

Consider a drawing process as shown below. The material is unwound under constant tension, and passed between 3 sets of grip rolls. The material is “drawn” to the required thickness by introducing a ratio between each set of grip rolls. The machine operator has a control panel next to the machine, while the controlling PLC is located some 100m in a another building.

A typical control system using **CTNet** is shown, consisting of 2 * Mentor II drives with tacho feedback and MD29AN cards, 3 * Unidrives with encoder feedback and UD70 with **CTNet** interfaces and 1 * I/O Box. The operator panel connects to the digital and analogue I/O on the I/O Box. There is no master PLC to control the system. The total network cable length is 20m. A transducer to measure the material thickness provides a feedback signal to Unidrive 3.

The required data rate of the system will be determined when the data transfer requirements are known.



Unidrive 3 is the master drive in the system. This receives the master speed reference from the Operator Control Panel via the I/O Box, and broadcasts the post-ramp speed reference to the other drives on the network. The material thickness feedback is used to trim the ratio settings to , which is used to trim the draw ratios to maintain material thickness.

Unidrive 3 also reads all the control signals from the control panel, decodes them ,and transmits the necessary control signals the other nodes via the cyclic channel.

Unidrive 2 is running in digital lock with Unidrive 3. It uses the encoder feedback from Unidrive 3 to generate its own speed reference, and stay locked to it. The draw ratio is read by Unidrive 3, and the digital lock ratio is transmitted from Unidrive 3. Unidrive 1 is locked to Unidrive 2, and also receives a digital lock ratio from Unidrive 3.

The Mentor II drives are running in torque control, with Coiler software installed in the MD29AN cards. They receive the line speed reference and the material tension reference from Unidrive 3, and the torque output is varied with as roll diameter changes.

Machine control is implemented using the I/O Box. References are set using potentiometers and switches into the I/O Box. Displays are controlled using the analogue and digital outputs from the I/O Box, with display information being transmitted to Unidrive 3 from other nodes using the slow cyclic data channel. The draw ratio is read in from four thumb wheel switches in BCD format, providing an accuracy of four decimcal places.

Node	Device	Function
1	Mentor II 1	Unwind drive, with Coiler software to provide tension control.
2	Unidrive 1	The first slave drive is running in digital lock with Unidrive 2. Digital lock ratio is passed over network.
3	Unidrive 2	The second slave drive is running in digital lock with Unidrive 3, the master drive. Digital lock ratio is passed over network.
4	Unidrive 3	The master drive is running in speed control, with the post-ramp speed reference used as the line speed reference.
5	Mentor II 2	Rewind drive, with Coiler software to provide tension control.

11.2 IN Cyclic Data

To specify the network requirements, it is necessary to analyse each node, and identify the data that must be transferred between nodes. This will determine the requirements of the network, and the maximum theoretical performance.

IN cyclic data means data that is coming IN to a particular node. As Unidrive 3 is monitoring and controlling the operator panel and line speed, it could be considered to be the master. Hence the IN and OUT data compared to the other nodes in reversed.

Node	Device	Fast Cyclic Data			Slow Cyclic Data	
1	Mentor II 1	Control word node 1	Line speed reference	Unwind tension reference		
2	Unidrive 1	Control word node2			Digital lock ratio 1	
3	Unidrive 2	Control word node 3			Digital lock ratio 2	
4	Unidrive 3	Status word node 1	Status word node 2	Status word node 3	Unwind roll diameter	Unwind final tension ref
		Status word node 5			Rewind roll diameter	Rewind final tension ref
5	Mentor II 2	Control word node 5	Line speed reference	Rewind tension reference		

The roll diameter calculation within the Mentor II gives the diameter relative to the initial reel diameter, so the MD29AN can calculate the actual roll diameter and pass it to Unidrive 3. This allows the line speed to be ramped up and down automatically as required when rolls are replaced.

11.3 OUT Cyclic Data

OUT cyclic data means data that in going OUT of the particular node. As Unidrive 3 is controlling the Operator Panel and line speed, it could be considered to be the master. Hence the IN and OUT data compared to the other nodes in reversed.

Node	Device	Fast Cyclic Data			Slow Cyclic Data	
1	Mentor II 1	Status word node 1			Unwind roll diameter	Unwind final tension ref
2	Unidrive 1	Status word node 2				
3	Unidrive 2	Status word node 3				

Node	Device	Fast Cyclic Data			Slow Cyclic Data	
4	Unidrive 3	Control word node 1	Control word node 2	Control word node 3		
4	cont.	Control word node 5	Rewind tension reference	Unwind tension reference		
4	cont.	Line speed reference			Digital lock ratio 1	Digital lock ratio 2
5	Mentor II 2	Status word node 5			Rewind roll diameter	Rewind final tension ref

The control words are written to each drive, thus making it a fully remote controlled system. For safety reasons, the ENABLE terminal on all drives would have to be hardwired into an emergency stop circuit. This would ensure that all drives are disabled instantly if the emergency stop is pressed.

The Mentor II does not have #90.11 as a control word, but the MD29AN can decode the data transmitted and modify the controlling bit parameters as required.

11.4 Network Set-up

A summary of the data transmission links can now be set as shown below.

Node	OUT registers		IN registers	
	Fast	Slow	Fast	Slow
Mentor II 1	1	2	3	---
Unidrive 1	1	---	1	1
Unidrive 2	1	---	1	1
Unidrive 3	2	2	4	4
Unidrive 3 (Broadcast)	2	---	---	
Mentor II 2	1	2	3	---

The amount of data that must be transmitted via the fast cyclic data channel is the main factor determining the fast cyclic data rate that may be set. For the above network, the minimum cycle times different data rates are as follows:

Data Rate	321.5K	625K	1.25M	2.5M	5M
Min Cycle Time (ms)	9.4	4.7	2.4	1.2	0.6

NOTE *For 1.25 Mbits/sec and below, the times are approximate.*

11.5 Network Data Rate

The following points need to be considered before the final data rate and cyclic data timings are chosen.

Network length	cable length is 50m. 5 Mbits/sec is OK up to 100m, so this does not impose any restrictions.
Drive operations	the ENCODER task in Unidrive which is controlling the operator panel runs every 5.52ms. For this reason, the Mentor II ENCODER task would be set to 5ms, so there is no advantage in updating cyclic data faster than 5ms.
Network loading	the load on the network with only the cyclic data being transmitted will help determine the data rate required. If 50% loading is chosen, this leaves 50% for non-cyclic data operations, and a SCADA node for future expansion.
Slow cyclic data	data that is for panel indication purposes only does not need to be updated every 5ms, as the operator would not be able to respond that quickly anyway. Updating the displays on the control panel every 50ms would be more than sufficient.

With 50% loading for the cyclic data, the following network set-up can be used.

Function		Unidrive	Mentor II	Value
Data rate	1.25 Mbits/sec	#20.02	#11.01	2
Synchronisation telegram	Fast data - 5ms Slow data - 50ms	#20.03	#11.02	1005

11.6 Digital I/O

The I/O Box has 32 inputs and 32 outputs. Utilisation is as follows:

- 4 * 4 inputs   draw ratio in BCD format. Provides the overall draw ratio of 0.xxxx.
- 8 inputs   inputs for run, jog, enable and emergency stop signals.

The digital outputs are used to control display indication lamps, etc. on the operator's control panel.

11.7 Analogue I/O

The I/O Box has 5 inputs and 3 outputs. Utilisation is as follows:

- 1 input — feedback from material thickness transducer.
- 1 input — master line speed reference.
- 2 inputs — material tension references for unwind and rewind sections.

The analogue outputs can be scaled to produce real unit readings on analogue meters.

- 1 output — material thickness, displayed in mm.
- 2 outputs — material tension, displayed in Newtons.

The line speed output can be taken from one of the drive analogue outputs.

12 Network Repeaters

Network repeaters allow the a **CTNet** network to be extended to accommodate more nodes or a longer length of network cable. The 3-port active hub version can also be used to create branches off the main trunk cable run.

	Part Number
2 Port Repeater	3000-0003
3 Port Active Hub	3000-0004

12.1 Repeater

12.1.1 Power Supply

The repeaters do not draw power from the network itself, and require a separate power supply to operate.

	DC	AC
Voltage	10 to 30 Volts	8 to 24 Volts
Power	4 Watts	4 VA
Frequency		47 to 63 Hz

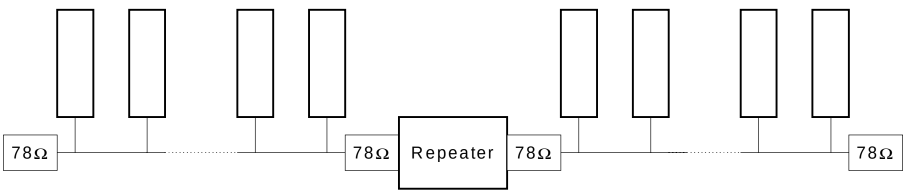
12.1.2 Data Rate

The network data rate should also be set on the selector switch on the front of the module. The repeaters support the following network data rates:

Data Rate (Mbits/sec)	Switch Setting
1.25	5
2.5	6
5.0	7

12.1.3 Network Termination

It is important to ensure that the cable is correctly terminated. Ensure that the internal 120Ω terminating resistors are disconnected by checking that the internal jumper link E1 is removed. 78Ω termination resistors should be fitted to the external connector.



12.1.4 Status LEDs

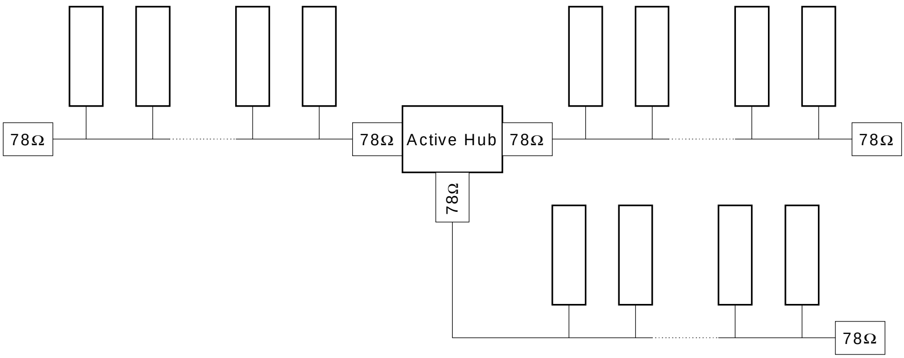
The operational status of the module is indicated by several LEDs.

Function	Indication	Status
STATUS	On	The unit is actively receiving network signals from **CTNet** at one port, and transmitting via another port.
STATUS	Flashing	The unit is healthy, but there is no network activity
ACTIVITY	On	One LED for each port. Each LED will light when data is being received at the port. Intensity of the LED will increase as the amount of received data increases.
RECON	On	The network is reconfiguring. This can occur when a node is switched off or on to the network. Repeated reconfiguration activity can indicate there are problems with the network. (See "NOFOVERRUNS", Section 4.6, Network Loss.)

12.2 Active Hub

The 3-port version can also be used as an active hub. This allows a branch off the main trunk cable to be connected into the line. Each section of cable must be terminated at both ends. If an active hub is used as a simple repeater, i.e. the third port is not used, there is no need to terminate the unused port.

Apart from the additional port, the 3-port version is otherwise identical in specification to the 2-port repeater.



13 CTNet-to-PC Interface Cards

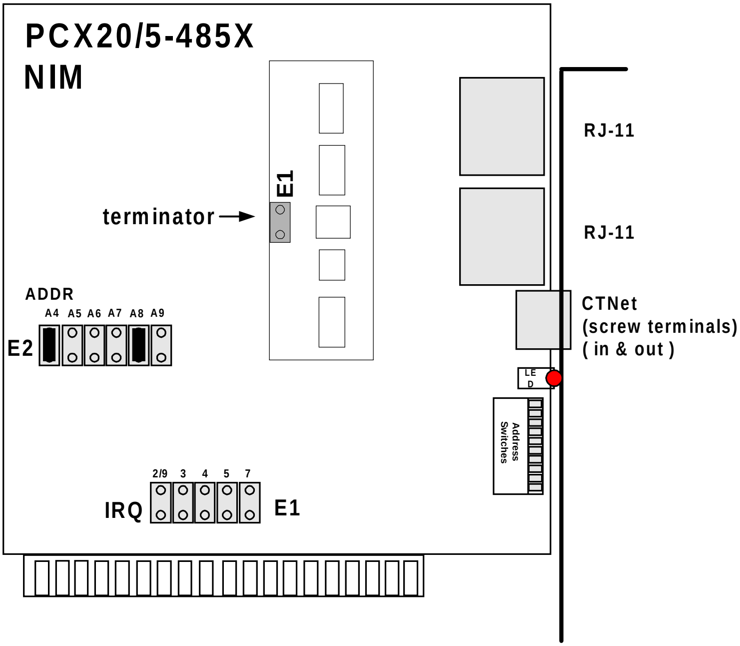
To attach a computer to a **CTNet** network, you will need a Network Interface Module (NIM). A NIM allows the software on a PC to interact with any **CTNet** node on the network. NIMs can be purchased through any Control Techniques Drive Centre. The following table lists the appropriate part numbers for the computer platforms supported:

Hardware Platform	Part Number
Desktop PC (ISA bus)	3000-0001
Laptop PC (PCMCIA)	3000-0002

NOTE *Ensure that anti-static handling precautions are followed when handling the interface card.*

13.1 PC ISA Card

The card has three sets of jumper links, which must be set before installation into the PC. The locations of these links are shown on the diagram below,:

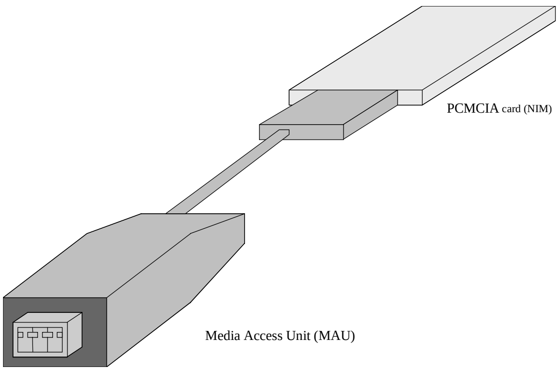


1. Remove E5. Jumper E5 connects the internal 120Ω terminating resistor, but as **CTNet** requires a 75Ω terminating resistor, this must be disconnected.
2. Ensure that the links A4 and A8 on jumper block E2 are fitted as shown. Jumper block E2 is used to set the I/O Base Address of the card. The factory setting selects address 2E0, which is usually appropriate for most PC applications. If this address is already used on your computer, refer to the printed documentation that is supplied with the ISA Card for a chart of the alternative settings.
3. Jumper E1 selects the interrupt request level (IRQ). SYPT currently does not require the use of an IRQ. Either remove all links (remember to leave a spare link on one pin only, so it can be used in the future if required), or fit the link in appropriate location of a free IRQ on your computer.

NOTE *Remember the settings you've made for the I/O Base Address since these will be needed when you attempt to communicate with **CTNet**.*

13.2 PCMCIA Card

The following instructions relate to installing on a **Windows 95** computer. The PCMCIA card drivers are provided with SYPT Workbench and **CTNet** DDE Server, and are also available on request from CT SSPD.



13.2.1 Installing the PCMCIA driver software:

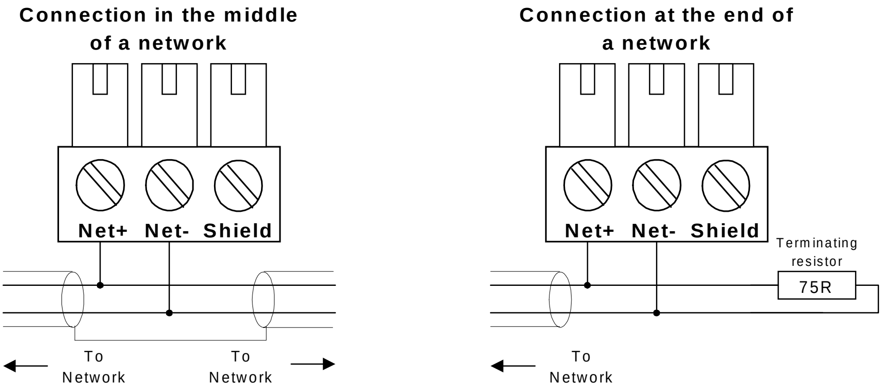
To install the PCMCIA card:

1. Plug the card into an empty PCMCIA type 2 socket.
2. Windows 95 "New Hardware Found" dialog will appear and it will identify the card as 'Contemporary Controls PCM20' and will ask for a disk that contains the drivers for configuring the card.
3. Direct Windows to look in the directory **\DRIVERS\PCMCIA** on the SYPT installation CDROM (or direct to drive A: if you have the drivers on disk.)
4. Select OK.

5. The Windows 95 Plug and Play system will automatically choose a 'base address' for the PCMCIA Card. In order to complete the SYPT installation, you must check to see what value Windows has chosen.
6. Determine the base address of PCMCIA card:
7. Select **Start->Settings->Control Panel**.
8. Double-click on the **System** icon.
9. Choose the **Device Manager** tab.
10. There should be a **PCM20** category displayed. Double-click this category and you should see a PCM20 device listed.
11. Double-click the **PCM20** device.
12. Choose the Resources tab. Listed under the resources tab will be an IO address range, for example **110-11F**. Take a note of the first number (e.g. 110).
13. Press Cancel, and close the **Device Manager** and **Control Panel.**

13.3 Connecting to the Network

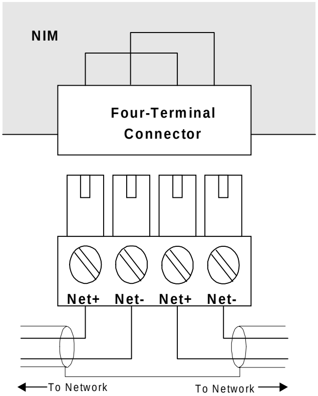
The Network Interface Modules use a three-pin removable screw terminal connector similar to the Control Techniques **CTNet** connector, with the shield tie-point missing. The diagram below shows the connections to make, depending on where the PC is to be placed in the **CTNet** network.



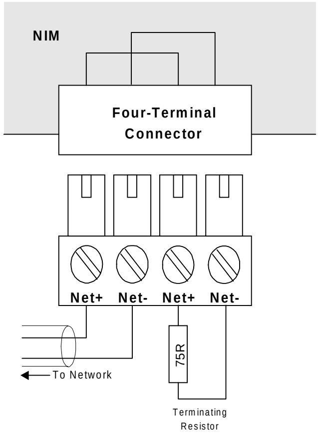
On most PCs, the shield connection is connected to the PC chassis, and thus to power earth. If connected, this can sometimes cause unwanted ground loops, so the normal connection recommendations exclude the shield connection.

NOTE *Older PC ISA cards have a 4-way **CTNet** connector, instead of a 3-way connector. The connections for this type of card are shown below. There are no performance differences between the newer and older cards.*

Connection in the Middle of a Network



Connection at the End of a Network



14 CTNET Software and Drives

14.1 SYPT Workbench

The SYPT Workbench is a complete integrated development for distributed drive applications. It contains several powerful tools, including an Application Configuration Editor for distributed systems, a Program Editor for DPL programs, Ladder Diagram and Function Block Diagram programs, plus a suite of Debugging and Compilation tools.

From the various menus you can:

- Create distributed drive system applications in Drive Programming Language, Ladder Diagram and/or Function Block Diagram Languages.
- Compile and debug applications, off-line or "live" (on-line).
- Graphically represent and simulate applications prior to implementation.
- Download applications to target drives, and upload programs from target drives.
- Download and upload programs via **CTNet**.

14.2 Dynamic Data Exchange

Dynamic Data Exchange (DDE) is a Microsoft standard for sharing data between Windows applications.

A DDE driver is available from Control Techniques Software and System Products Division that allows Windows-based software applications from other companies to interface to **CTNet**, and allow data transfer between the "Client" application and the **CTNet** DDE "server".

14.3 Intellution FIX32 Driver

An special driver is available for use with FIX32 SCADA package from Intellution. This driver allows FIX32 to interface directly to the **CTNet** network, without the need for a DDE server.

Contact your local Drive Centre for further information about any of the above.

15 Appendix

15.1 CTNet Network Operation

CTNet is a full “peer-to-peer” communications system which allows full communications between any nodes, without the need for an overall network master controller. A “peer-to-peer” system means that all nodes have the ability to transmit data to over the network, so a “token rotation" system is used to prevent data collisions. This ensures that only one node can transmit over the network at a time.

The token is a “software token”, and is passed from node to node in the form of a message. When a node is in possession of the token, it has full control of the network, and can transmit a single data message. The token is then passed on to the drive with the next highest node address on the network, and so on. If there is no node with a higher address, the token is passed to the drive with the lowest address, and the token rotation sequence starts again. The token takes 57µs to transmit per node, so the token passage between nodes is very fast.

To control the cyclic data over the network, one node on the network is configured as a “pseudo-master”, and this node is responsible for generating the cyclic data interrupts. After an interrupt has occurred, the “pseudo-master” node will broadcast the “synchronisation telegram” when it is next in possession of the token. This message is received by all nodes, and is the signal to each node start sending cyclic data messages.

During cyclic data transfer, one cyclic data link is transmitted every time a node receives the token. If 10 data links are defined within a single node, the token will have to be passed around the whole system 10 times, before cyclic data transfer from the node is complete.

If a message gets corrupted and the CRC does not check out correctly, or drive goes off the network, the network controller chip on the source node will automatically attempt to re-send the message. If, after 5 re-tries, it is still unable to receive a response to the message, it will discard the message, and increment the “**NOFRETRIES**” register. (See Section 4.6, Network Loss).

15.2 Error Detection

CTNet commands incorporate error detection automatically. Error correction is not provided, as the lifetime of data in industrial applications using real-time communications systems is very short, and transmitting error correction data would only consume additional network bandwidth.

16 Quick Parameter Reference

16.1 Set-up Parameters

Function	Unidrive	Mentor II	Range	Comments
Node Address	#20.01	#14.05	1 to 255	Set node address to 0 to disable the **CTNet** driver.
Network Data Rate	#20.02	#11.01	0 to 4	All nodes on the network must be programmed to run at the same data rate.
Synchronisation Message	#20.03	#11.02	NNCC	CC = fast cyclic data rate in ms. NN = slow cyclic rate as multiple of fast cyclic rate. NN ranges from 2 to 19.

16.2 Unidrive Mapping Parameters

Slot	Source/Destination Parameter	Destination Node
IN Slot 1	#20.10 (MMPP)	
IN Slot 2	#20.11 (MMPP)	
IN Slot 3	#20.12 (MMPP)	
OUT Slot 1	#20.05 (MMPP)	#20.04 (NNNSS)
OUT Slot 2	#20.07 (MMPP)	#20.06 (NNNSS)
OUT Slot 3	#20.09 (MMPP)	#20.08 (NNNSS)

16.3 Mentor II Mapping Parameters

Slot	Source/Destination Parameter	Destination Node
IN Slot 1	#11.07 (MMPP)	
IN Slot 2	#11.08 (MMPP)	
OUT Slot 1	#11.04 (MMPP)	#11.03 (NNNS)
OUT Slot 2	#11.06 (MMPP)	#11.05 (NNNS)

16.4 Activating and Saving Set-up And Mapping Parameters

Action	Unidrive	Mentor II
Activate mapping changes	Set #MM.00 to 1070 and press the red RESET button.	Press RESET button when the drive is disabled.
Save and activate mapping changes	Set #17.19 to 1.	Set #MM.00 to 1 and press RESET.
Remote drive reset	Set #10.38 to 100.	Set bit 11 of the Control Word to 1.

16.5 Control Word

Bit	Unidrive - UD75	
	Parameter	Description
0	#6.15	Drive enable
1	#6.30	Sequencing bit 0
2	#6.31	Sequencing bit 1
3	#6.32	Sequencing bit 2
4	TRIP	Drive Trip (tr 52)
5	#1.45	Preset select 0
6	#1.46	Preset select 1
7	#18.31	Application bit
8	#18.32	Application bit
9	M0	Mask bit 0
10	M1	Mask bit 1
11	M2	Mask bit 2
12	M3	Mask bit 3
13	#18.33	Application bit
14	M5	Mask bit 5
15	M6	Mask bit 6

16.6 Status Word

Bit	Unidrive - UD75	
	Parameter	**Description**
0	#10.01	Drive healthy
1	#10.02	Drive running
2	#10.03	Zero speed
3	#10.04	Running at or below minimum speed
4	#10.05	Below set speed
5	#10.06	At speed
6	#10.07	Above set speed
7	#10.08	Load reached
8	#10.09	In current limit
9	#10.10	Regenerating
10	#10.11	Dynamic brake active
11	#10.12	Dynamic brake alarm
12	#10.13	Direction commanded
13	#10.14	Direction running
14	#10.15	Mains Loss
15	X	Not used