



*User Guide*

# **Applications Modules**

## **Solutions Modules for Unidrive SP**

(Incorporating SM-Applications Lite, SM-Applications  
and SM-Applications Plus)

## **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 variable speed drive (drive) with the motor.

The contents of this 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 this guide, without notice.

All rights reserved. No parts of this guide may be reproduced or transmitted in any form or by any means, electrical or mechanical including photocopying, recording or by an information storage or retrieval system, without permission in writing from the publisher.

## **Drive software version**

This product is supplied with the latest version of user-interface and machine control software. If this product is to be used in a new or existing system with other drives, there may be some differences between their software and the software in this product. These differences may cause this product to function differently. This may also apply to drives returned from a Control Techniques Service Centre.

If there is any doubt, contact a Control Techniques Drive Centre.

Copyright © 19 June 2006 Control Techniques Drives Ltd

Issue Code: 2

Firmware: V01.04.05

---

# Contents

---

<b>1</b>	<b>Safety Information .....</b>	<b>6</b>
1.1	Warnings, Cautions and Notes .....	6
1.2	Electrical Safety - General Warning .....	6
1.3	System Design and Safety of Personnel .....	6
1.4	Environmental Limits .....	7
1.5	Compliance with Regulations .....	7
1.6	Motor .....	7
1.7	Adjusting Parameters .....	7
<b>2</b>	<b>Introduction .....</b>	<b>8</b>
2.1	Features .....	8
2.2	Solutions Module identification .....	9
2.3	Conventions used in this guide .....	9
2.4	PC Development Software .....	10
2.5	User Knowledge .....	10
<b>3</b>	<b>Installation .....</b>	<b>11</b>
3.1	Solutions Module slots .....	11
3.2	Installation .....	11
3.3	Electrical Connections <sup>♣♥</sup> .....	12
3.4	CTNet Connections <sup>♣♥</sup> .....	13
3.5	CTNet Cable .....	14
3.6	CTNet Network Termination .....	14
3.7	EIA-RS485 Connections <sup>♣♥</sup> .....	15
3.8	Digital I/O Connections <sup>♣♥</sup> .....	17
3.9	Port isolation <sup>♣♥</sup> .....	17
3.10	SMARTCARD Installation .....	18
<b>4</b>	<b>Getting Started .....</b>	<b>19</b>
4.1	Using SyPTPro and SyPTLite .....	19
4.2	Connecting the PC to the Applications module .....	19
4.3	CTNetAPI routing .....	20
4.4	Configuring Communications within SyPTPro .....	20
4.5	Creating a Unidrive SP node in SyPTPro .....	20
4.6	Porting UD70 programs to the Applications module .....	21
4.7	Using SyPTLite .....	21
4.8	DPL Programming Basics .....	22
4.9	Program Example .....	23
4.10	Downloading Programs .....	24
4.11	SyPTPro Caveats .....	24

<b>5</b>	<b>Parameters .....</b>	<b>25</b>
5.1	Overview .....	25
5.2	Saving Parameters .....	25
5.3	Configuration Parameters .....	26
5.4	Menus 70-75 - PLC Registers .....	37
5.5	Menu 85 - Timer Function Parameters <sup>♣♥</sup> .....	38
5.6	Menu 86 - Digital I/O Parameters .....	40
5.7	Menu 88 - Status Parameters .....	41
5.8	Menu 90 - General Parameters .....	43
5.9	Menu 91 - Fast Access Parameters .....	50
5.10	Menus 18,19 - Application Parameters .....	54
5.11	Menu 20 - Application Menu .....	55
<b>6</b>	<b>DPL Programming .....</b>	<b>56</b>
6.1	Program Header .....	56
6.2	Tasks .....	57
6.3	Variables .....	59
6.4	Parameters .....	61
6.5	Operators .....	62
6.6	Basic DPL Commands .....	63
6.7	User Defined Function Blocks .....	68
<b>7</b>	<b>Communications .....</b>	<b>70</b>
7.1	EIA-RS485 Serial Communications Port <sup>♣♥</sup> .....	70
7.2	CTNet <sup>♣♥</sup> .....	76
7.3	Applications module mapping parameters (fieldbus) .....	77
<b>8</b>	<b>Freeze and Marker .....</b>	<b>79</b>
8.1	Freeze input <sup>♣♥</sup> .....	79
8.2	Freeze input <sup>♦</sup> .....	80
8.3	Marker pulse <sup>♦♣♥</sup> .....	82
<b>9</b>	<b>CTSync <sup>♣♥</sup> .....</b>	<b>83</b>
9.1	Overview .....	83
9.2	Connections .....	83
9.3	Limitations .....	83
9.4	CTSync Function Blocks .....	83
9.5	Motion Engine .....	85
9.6	Virtual Master Example .....	87
<b>10</b>	<b>Inter-option Synchronisation .....</b>	<b>89</b>
10.1	Overview .....	89
10.2	Inter-Option Synchronisation example 1 .....	89
10.3	Inter-option Synchronisation example 2 <sup>♣♥</sup> .....	93
10.4	Position control tasks .....	97

<b>11</b>	<b>Diagnostics .....</b>	<b>98</b>
11.1	Run-time Errors .....	98
11.2	Unidrive SP Display Trip Codes .....	98
11.3	Applications Module Run-time Error Codes .....	99
11.4	Handling Run-Time Errors with the ERROR task .....	101
11.5	Resource monitoring .....	102
11.6	Support .....	103
<b>12</b>	<b>Migration Guide .....</b>	<b>104</b>
12.1	Drive Parameter Differences .....	104
12.2	UD70 Parameters .....	104
12.3	General Features .....	106
12.4	Applications Module Porting aid .....	107
<b>13</b>	<b>Quick Reference .....</b>	<b>109</b>
	<b>Index .....</b>	<b>114</b>

---

# 1 Safety Information

---

## 1.1 Warnings, Cautions and Notes



A **Warning** contains information, which is essential for avoiding a safety hazard.



A **Caution** contains information, which is necessary for avoiding a risk of damage to the product or other equipment.



A **Note** contains information, which helps to ensure correct operation of the product.

## 1.2 Electrical Safety - General Warning

The voltages used in the drive can cause severe electrical shock and/or burns, and could be lethal. Extreme care is necessary at all times when working with or adjacent to the drive. Specific warnings are given at the relevant places in this User Guide.

## 1.3 System Design and Safety of Personnel

The drive is intended as a component for professional incorporation into complete equipment or a system. If installed incorrectly, the drive may present a safety hazard.

The drive uses high voltages and currents, carries a high level of stored electrical energy, and is used to control equipment which can cause injury.

Close attention is required to the electrical installation and the system design to avoid hazards either in normal operation or in the event of equipment malfunction. System design, installation, commissioning and maintenance must be carried out by personnel who have the necessary training and experience. They must read this safety information and this User Guide carefully.

The STOP and SECURE DISABLE functions of the drive do not isolate dangerous voltages from the output of the drive or from any external option unit. The supply must be disconnected by an approved electrical isolation device before gaining access to the electrical connections.

**With the sole exception of the SECURE DISABLE function, none of the drive functions must be used to ensure safety of personnel, i.e. they must not be used for safety-related functions.**

Careful consideration must be given to the functions of the drive which might result in a hazard, either through their intended behaviour or through incorrect operation due to a fault. In any application where a malfunction of the drive or its control system could lead to or allow damage, loss or injury, a risk analysis must be carried out, and where necessary, further measures taken to reduce the risk - for example, an over-speed protection device in case of failure of the speed control, or a fail-safe mechanical brake in case of loss of motor braking.

The SECURE DISABLE function has been independently approved by BGIA as meeting the requirements of EN954-1 category 3 for the prevention of unexpected starting of the drive. It may be used in a safety-related application. **The system designer is responsible for ensuring that the complete system is safe and designed correctly according to the relevant safety standards.**

## 1.4 Environmental Limits

Instructions in the *Unidrive SP User Guide* regarding transport, storage, installation and use of the drive must be complied with, including the specified environmental limits. Drives must not be subjected to excessive physical force.

## 1.5 Compliance with Regulations

The installer is responsible for complying with all relevant regulations, such as national wiring regulations, accident prevention regulations and electromagnetic compatibility (EMC) regulations. Particular attention must be given to the cross-sectional areas of conductors, the selection of fuses or other protection, and protective earth (ground) connections.

The *Unidrive SP User Guide* contains instruction for achieving compliance with specific EMC standards.

Within the European Union, all machinery in which this product is used must comply with the following directives:

98/37/EC: Safety of machinery.

89/336/EEC: Electromagnetic Compatibility.

## 1.6 Motor

Ensure the motor is installed in accordance with the manufacturer's recommendations. Ensure the motor shaft is not exposed.

Standard squirrel cage induction motors are designed for single speed operation. If it is intended to use the capability of the drive to run a motor at speeds above its designed maximum, it is strongly recommended that the manufacturer is consulted first.

Low speeds may cause the motor to overheat because the cooling fan becomes less effective. The motor should be fitted with a protection thermistor. If necessary, an electric forced vent fan should be used.

The values of the motor parameters set in the drive affect the protection of the motor. The default values in the drive should not be relied upon.

It is essential that the correct value is entered in Pr **0.46** motor rated current. This affects the thermal protection of the motor.

## 1.7 Adjusting Parameters

Some parameters have a profound effect on the operation of the drive. They must not be altered without careful consideration of the impact on the controlled system. Measures must be taken to prevent unwanted changes due to error or tampering.

---

## 2 Introduction

---

### 2.1 Features

Modern variable speed drives such as the Unidrive SP offer a multitude of in-built features such as ramp control, PID loops, simple position control, etc. However this functionality is limited. The drive can only do so many things and when it comes to controlling more complex applications, users often have to resort to using external equipment such as PLCs to control the drive from a system point of view.

However the flexibility of the Unidrive SP can be substantially increased by using the Applications modules. These modules are an additional processor for the drive and allow the user to utilise existing, or write their own, application-specific software. They also offer powerful networking capabilities so many drives (and other equipment) can be connected together to communicate process wide information thus offering a complete application solution.

The Applications modules for Unidrive SP are solutions modules that can be fitted to expansion slots in the Unidrive SP.

The Applications modules are powered from the Unidrive SP internal power supply.

#### 2.1.1 Specifications for SM-Applications and SM-Applications Plus

- High speed dedicated microprocessor
- 384kb Flash memory for user program
- 80kb user program memory
- EIA-RS485 port offering ANSI, Modbus-RTU slave and master and Modbus-ASCII slave and master protocols
- CNet high speed network connection offering up to 5Mbit/s data rate.
- Two 24V digital inputs
- Two 24V digital outputs
- Dual-port RAM interface for communicating with the Unidrive SP and other option modules
- Task based programming system allowing for real-time control of drive and process.

**NOTE**

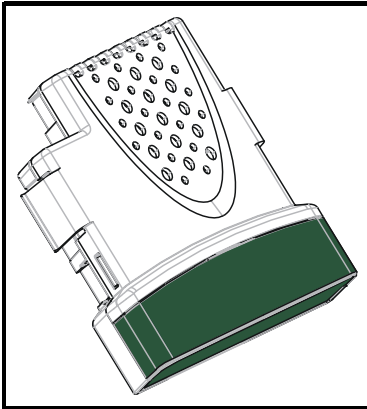
The SM-Applications Plus module has a slightly faster variant of the SM-Applications microprocessor, thereby increasing performance of the module over the SM-Applications.

#### 2.1.2 Specifications for SM-Applications Lite

- High speed dedicated microprocessor
- SyPTLite 10kb executable memory
- SyPTPro 150kb executable memory, 20kb user memory
- Dual-Port RAM interface for communicating with the Unidrive SP and other option modules.
- Task based programming system allowing for real time control of the drive and process

## 2.2 Solutions Module identification

Figure 2-1 Applications Module



The Applications modules can be identified by:

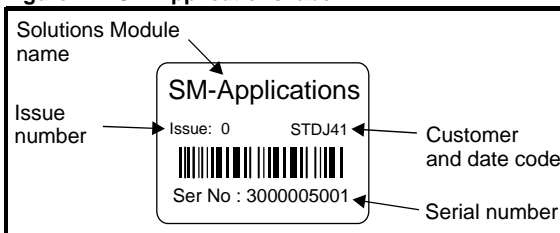
1. The label located on the underside of the Solutions Module.
2. The colour coding across the front of the Solutions Module.

SM-Applications: Dark Green

SM-Applications Lite: White

SM-Applications Plus: Light Moss Green

Figure 2-2 SM-Applications label



## 2.3 Conventions used in this guide

This manual contains information relating to the SM-Applications, SM-Applications Lite and SM-Applications Plus option modules for Unidrive SP.

Throughout the manual these modules will be referred to generically as Applications modules. Should a particular section refer to one or two of these modules only, the reference will be to those modules by their name directly.

Some parts of this manual (such as the parameters section) will contain a superscript icon for each of the three modules. The key is as follows:

- ♦ : **SM-Applications Lite**
- ♥ : **SM-Applications**
- \* : **SM-Applications Plus**

For example, should a description have the superscript keys ♥ and \* superseding it then the description relates to both the SM-Applications Plus and SM-Applications modules.

If a description has the superscript key <sup>♦</sup> superseding it then the description relates to the SM-Applications Lite module only.

Where there is no superscript icon it can be assumed that the section/description relates to all three Applications modules.

The configuration of the host drive and Solutions Module is done using menus and parameters. A menu is a logical collection of parameters that have similar functionality. In the case of a Solutions Module, the parameters will appear in menu 15, 16 or 17 for the Unidrive SP depending on the slot the module is fitted into. The menu is determined by the number before the decimal point.

The method used to determine the menu or parameter is as follows:

- Pr **xx.00** - signifies any menu and parameter number 00.
- Pr **x.XX** - where **x** signifies the menu allocated to the Solutions Module (this could be 15, 16 or 17 on the Unidrive SP) and **XX** signifies the parameter number.

## 2.4 PC Development Software

Application programs for the Applications modules may be developed by the user using the SyPTPro software tool. Users who are familiar with the UD70 product for the Unidrive Classic may already be familiar with this software.

SyPTPro offers various tools to help in developing solutions:

- Configuration editor for configuring drives and connections on CTNet, CT-RTU, CT-TCP and MD29MON networks.
- IEC61131-3 based ladder and function block programming
- Native DPL language programming
- Watch window for monitoring drive and option parameters, and program variables
- Single-stepping and breakpoint debugging facilities

With SyPTPro you may connect to the Applications modules by either:

- Direct connection to the EIA-RS485 port on the front of the Unidrive SP. <sup>♦♦♥</sup>
- Connecting to one or more options on a CTNet network (a CTNet interface card for the PC will be required). <sup>♦♥</sup>

SyPTPro runs under Microsoft Windows™ 98/ME/NT4/2000/XP.


## 2.5 User Knowledge

If developing custom application software it is beneficial to have some understanding of real-time task and event driven programming. A rudimentary understanding of the BASIC programming language is also beneficial but not essential. The ladder diagram (LD) and function block diagram (FBD) facilities of SyPTPro make it much easier for people familiar with PLCs to migrate.


This User Guide assumes the user has at least superficial knowledge of Microsoft Windows™.

# 3 Installation

## 3.1 Solutions Module slots

 Before installing the Solutions Module, refer to Chapter 1 *Safety Information* on page 6.

**WARNING**

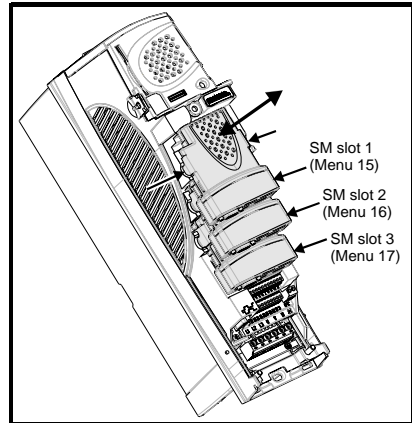
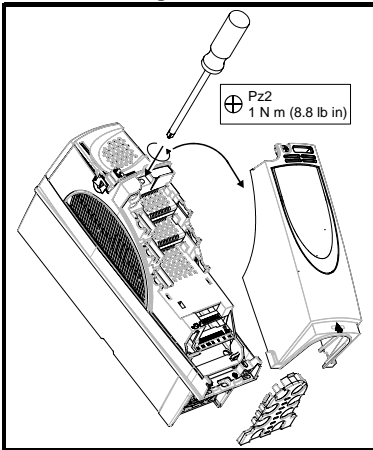
 Do not remove the Solutions Module when the drive is powered-up. This could damage the drive and/or the Solutions Module.

**CAUTION**

## 3.2 Installation

Three Solutions Module slots are available on Unidrive SP. The Solutions Module can be plugged into any of these slots but it is recommended that slot 3 be used for the first Solutions Module, then slot 2 and then slot 1. This ensures the maximum mechanical support for the Solutions Module once fitted. See Figure 3-2.

**Figure 3-1 Removing the Uni SP terminal cover**      **Figure 3-2 Fitting and removing a Solutions Module into the Unidrive SP**



1. Before installing a Solutions Module into the drive, ensure the AC supply has been disconnected for at least 10 minutes.
2. Ensure that any +24V and low voltage DC power supplies (if used) have been disconnected from the drive.
3. Check that the exterior of the Solutions Module is not damaged and the multiway connector on the underside of the module is free from dirt and debris.
4. Do not install a damaged or dirty Solutions Module in the drive.
5. Remove the terminal cover from the drive as shown in Figure 3-1.
6. Position the drive connector of the Solutions Module over the appropriate connector of the drive. Push the Solutions Module towards the rear of the drive until it locks into place. Make any wiring connections as appropriate.
7. Re-fit the terminal cover to the drive by reversing the procedure shown in Figure 3-1.

8. Connect the AC supply or low voltage DC supply (if used) to the drive.
9. When a Solutions Module is fitted to a Unidrive SP for the first time, as the drive is powered-up, the drive will trip on SL1.dF or SL2.dF or SL3.dF depending on which slot the Solutions Module is fitted to. A parameter save must be performed. Set Pr **xx.00** = 1000 (or 1001 in the case of solely using the 24V back-up power supply) and press the Stop/Reset button. If a parameter save is not performed, the drive will trip on the above trips the next time the drive is powered up.

**NOTE** When using the 24V back-up power supply only, the SLx.dF trip will not occur (as the drive is already displaying a UU trip).

**NOTE** When fitting two or more Solutions Modules simultaneously, the SLx.dF trip is only applicable to the module fitted in the lowest numerical slot.

**NOTE** If an SLx.dF trip is not seen after the first power-up, the Solutions Module is not fitted properly to the drive. Power down the drive, remove and re-fit the Solutions Module. Then power can be re-applied to the drive.

10. To access the advanced parameters refer to the *Unidrive SP User Guide*.
11. Check that Menu 15 (slot 1), 16 (slot 2), or 17 (slot 3) parameters are now available (depending on which slot is used).
12. Check that Pr **15.01**, Pr **16.01** or Pr **17.01** shows the correct code for the relevant module, as shown below.
  - SM-Applications: 301
  - SM-Applications Lite: 302
  - SM-Applications Plus: 304
13. Power the drive down and back up. The Solutions Module is now ready for programming.

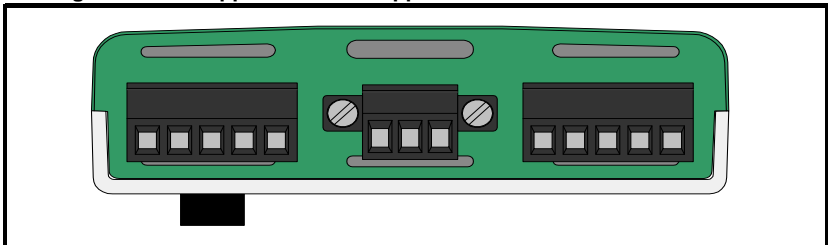
**NOTE** If the Solutions Module is changed for another type of Solutions Module, the drive will trip as in step 9. Follow the above procedure.

14. If the checks in steps 11 and 12 fail, either the Solutions Module is not fully inserted, or it is faulty.
15. If a trip code is now present refer to *Chapter 11 Diagnostics* on page 98.

### 3.3 Electrical Connections<sup>†♥</sup>

The SM-Applications and SM-Applications Plus modules have 2 no. 5-way and 1 no. 3-way screw terminal blocks.

**Figure 3-3 SM-Applications/SM-Applications Plus modules - Front View**



The terminals are numbered from terminal 1 on the left hand side to terminal 13 on the right. The terminal functions are given in the table below:

**Table 3.1 Module Connectors**

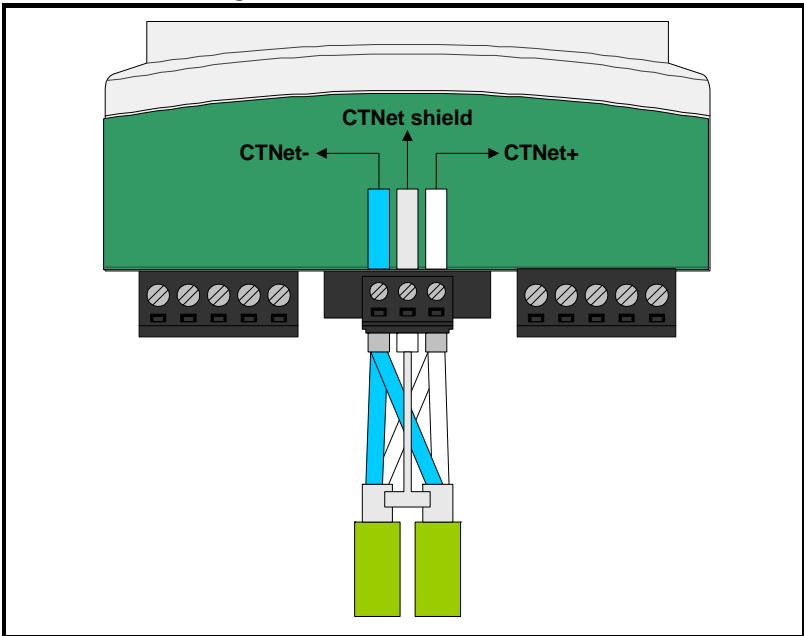
Terminal	Function	Description
1	0V SC	0V connection for EIA-RS485 port
2	/RX	EIA-RS485 Receive line (negative). Incoming.
3	RX	EIA-RS485 Receive line (positive). Incoming.
4	/TX	EIA-RS485 Transmit line (negative). Outgoing.
5	TX	EIA-RS485 Transmit line (positive). Outgoing.
6	CTNet A	CTNet data line
7	CTNet Shield	Shield connection for CTNet
8	CTNet B	CTNet data line
9	0V	0V connection for digital I/O
10	DIO	Digital input 0
11	DI1	Digital input 1
12	DO0	Digital output 0
13	DO1	Digital output 1

### 3.4 CTNet Connections\*♥

**NOTE** This User Guide covers only the basics of connecting a CTNet network. Please refer to the *CTNet User Guide* for full information.

To connect the module to the CTNet network, make the connections as shown in the diagram below.

**Figure 3-4 CTNet Network Connections**



The shields of both cables should be twisted together and connected into the centre terminal of the 3-way terminal block. This “pigtail” should be kept as short as possible. This design guarantees shield continuity.

### 3.5 CTNet Cable

CTNet cable has a single twisted pair plus overall screening. One set of data terminals is provided. This has the advantage that if the terminal block is unplugged the continuity of the CTNet network is not broken.

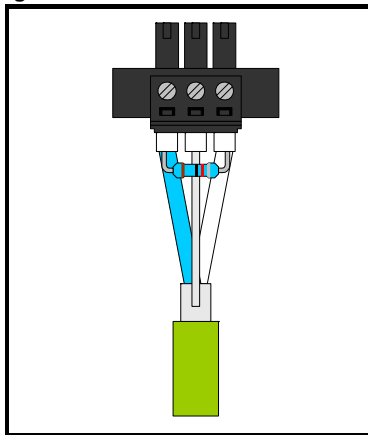
CTNet networks run at high data rates, and require cable specifically designed to carry high frequency signals. Low quality cable will attenuate the signals, and may render the signal unreadable for the other nodes on the network. The only recommended CTNet cable is that supplied by Control Techniques.

### 3.6 CTNet Network Termination

It is very important in high-speed communications networks that the network communications cable is fitted with the specified termination resistor network at each end of the cable. This prevents signals from being reflected back down the cable and causing interference.

The termination resistance should match as closely as possible the impedance of the cable. For the recommended Control Techniques green CTNet cable, an 82Ω 0.25W termination resistor should be fitted across the CTNet+ and CTNet- data lines at BOTH ends of the cable run.

Figure 3-5 CTNet Network Termination



Failure to terminate a network correctly can seriously affect the operation of the network. If the correct termination resistors are not fitted, the noise immunity of the network is greatly reduced.

If too many termination resistors are fitted on a CTNet network, the network will be overloaded, resulting in reduced signal levels. This may cause nodes to miss some bits of information, resulting in transmission errors being reported. If network overload is excessive, the signal levels may be so low that nodes cannot detect any network activity at all.

### 3.6.1 CTNet Cable Screen Connections

The CTNet connection should be wired with the cable shields isolated from earth at each Unidrive SP. The cable shields should be linked together at the point where they emerge from the cable, and formed into a short pigtail to be connected to CTNet shield connection on the terminal block, as already shown.

For safety, the CTNet shield must be connected to earth at one point. This earth connection is to prevent the cable screen from becoming live in the event of catastrophic failure of another device on the CTNet network or along the cable run. For EMC reasons the earth should be connected at one point only.

### 3.6.2 Maximum Network Length and Number of Nodes

The maximum number of nodes that can be connected to a single CTNet network is 255, however a network may need to be split into segments, separated by repeaters. The maximum length of network cable for a CTNet network is dependent on the baud rate.

Repeaters can be purchased from Control Techniques. The table below shows the part numbers for the different repeaters available.

Part Number	Description
4500-0033	AI3-485X - Rev A
4500-0083	AI3-485X-CT - Rev D (Port 1), Rev A (Ports 2 and 3)
4500-0082	AI3-CT - Rev D
4500-0032	AI2-485X/FOG-ST - Rev A (Fibre Optic)
4500-0081	AI2-CT/FOG-ST - Rev D (Fibre-Optic)

Please refer to the *CTNet User Guide* for further guidance on these points.

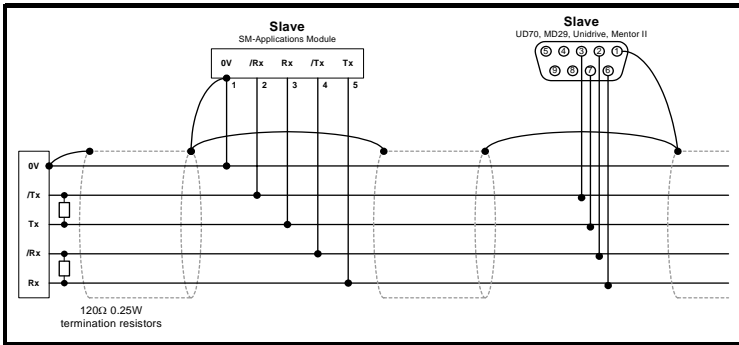
## 3.7 EIA-RS485 Connections\*<sup>†</sup>

The EIA-RS485 port is available for lower-speed communications (up to 115,200bit/s). As standard the port supports the CT-ANSI slave, Modbus-RTU master and slave, and Modbus-ASCII master and slave protocols. Both 2 and 4-wire connections are possible.

More information on the use of the EIA-RS485 port can be found in Chapter 7 *Communications* on page 70.

### 3.7.1 4 Wire EIA-RS485 Network

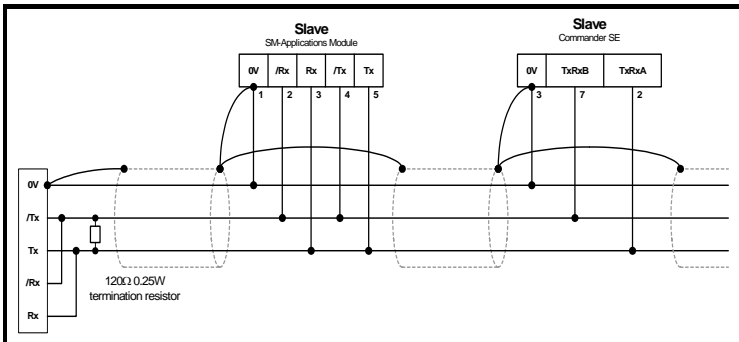
The diagram below shows the connections required for a 4 wire EIA-RS485 network, using a master controller with an EIA-RS485 port. Applications modules can be configured to act as master controllers, but this requires DPL programming to control the network.



An EIA-RS232 to EIA-RS485 converter is required to allow a standard PC serial port to communicate with a 4 wire EIA-RS485 network.

### 3.7.2 2 Wire EIA-RS485 Network

The diagram below shows the connections required for a 2 wire EIA-RS485 network, using a master controller with an EIA-RS485 port. Applications modules can be configured to act as master controllers, but this requires DPL programming to control the network.



An EIA-RS232 to EIA-RS485 converter with “intelligent transceiver switching” (also known as “magic” EIA-RS485 converters) is required to allow a standard PC serial port to communicate with a 2 wire EIA-RS485 network. An example of a “magic” converter is the MA485F converter from Amplicon.

**NOTE**

A “magic” converter is not required if the master controller has an RTS control output. This output is enabled when the master is transmitting, and disabled when the master is not transmitting. Control Techniques software packages (UniSoft, MentorSoft and SystemWise) do NOT switch the RTS line.

### 3.7.3 Grounding

It is recommended for safety that the shield of the communications cable be connected by a low-inductance path to a ‘clean’ earth point. This must only be done at one point.

### 3.7.4 Routing of the cable

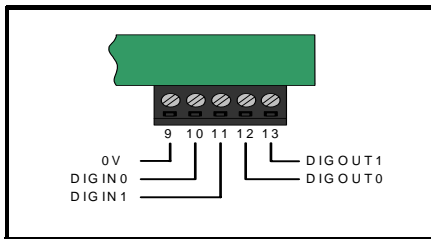
A data communications cable should not run parallel to any power cables, especially ones that connect drives to motors. If parallel runs are unavoidable, ensure a minimum spacing of 300mm (1 foot) between the communications cable and the power cable.

Cables crossing one another at right-angles are unlikely to give trouble. The maximum cable length for a EIA-RS485 link is 1200 metres (4,000 feet). This is at low baud rates only. The higher the baud rate the lower the maximum cable length.

### 3.7.5 Termination

When a long-distance multi-drop EIA-RS485 system is used, the transmit and receive pairs should have a termination resistor of 120Ω fitted across them in order to reduce signal reflections. However, at the lower data rates this is not so critical.

## 3.8 Digital I/O Connections\*♥



The SM-Applications and SM-Applications Plus modules are equipped with 2 digital inputs, known as DIGIN0 and DIGIN1, and 2 digital outputs, known as DIGOUT0 and DIGOUT1. These inputs and outputs can be read/controlled from the user program loaded into the Applications module.

The digital outputs are a positive logic arrangement such that they are at +24V when active and will supply up to 20mA of current. When inactive they are effectively floating. The digital outputs are protected against short-circuit or overload. The trip threshold is 20mA, and if tripped both outputs will be deactivated.

The digital I/O are controlled using menu 86 - refer to section 5.6 *Menu 86 - Digital I/O Parameters* on page 40.

## 3.9 Port isolation\*♥

The digital input/output ports are connected to the main drive control circuits.



The I/O circuits are isolated from the power circuits by basic insulation (single insulation) only. The installer must ensure that external control circuits are insulated from human contact by at least one layer of insulation (supplementary insulation) rated for the AC supply voltage

The CTNet and EIA-RS485 ports have supplementary insulation from the input/output ports, giving overall double insulation from the power circuit. They have simple separation (functional insulation) from each other.



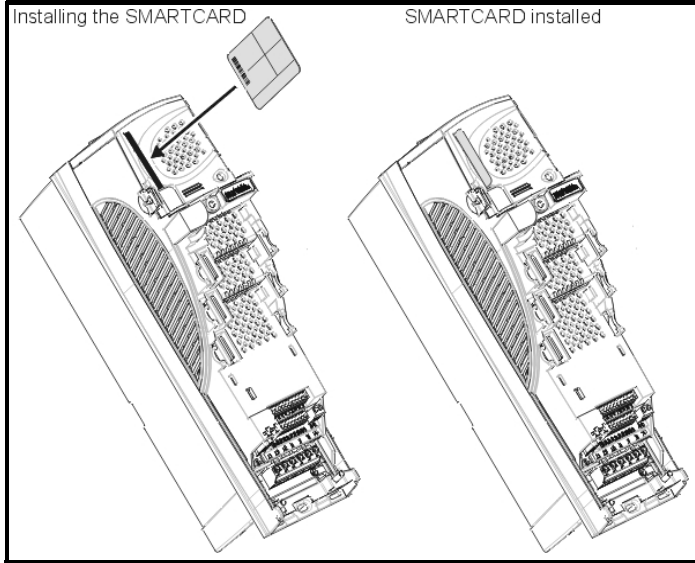
To maintain the double insulation status of the data ports:

- All circuits to which either port is connected must have protective separation (i.e. double insulation or single insulation with grounding)
- All circuits to which the drive control circuits are connected must have at least basic insulation from live parts.

### 3.10 SMARTCARD Installation

The Unidrive SP can be fitted with a SMARTCARD (see Figure 3-6 *Installing the SMARTCARD* below) which allows the user to read and write data. This is useful for storing parameter sets and other data values. For more information on using the SMARTCARD from the keypad, refer to the *Unidrive SP User Guide*. Refer to section 6.6.1 *New Commands for Applications Modules* on page 63 for information on using DPL commands with the SMARTCARD.

**Figure 3-6 Installing the SMARTCARD**



# 4 Getting Started

This chapter describes the basics of a user program using the Applications modules and some of the aspects of using SyPTPro.

Within the Applications modules, the current slot menu is aliased as menu 81. Therefore when connected to the module via a communications link or from the user program, it is easiest if configuration parameters are referenced as menu 81.

Throughout the remainder of this User Guide, when referring to a specific parameter for any slot the format **Pr 81.XX** will be used. e.g. The *Autorun* parameter will be referred to as **Pr 81.13**.

This is also an aid to portability, as an Applications module with code using menu 81 can be fitted to any slot, and the code should run as normal.

When the Applications modules are fitted, the module identification parameter **Pr 81.01** will show the value 301 for the SM-Applications, the value 302 for the SM-Applications Lite and the value 304 for the SM-Applications Plus. The combination of parameters **Pr 81.02** and **Pr 81.51** provides the firmware version of the module. This User Guide is written for firmware version V01.04.05.

## 4.1 Using SyPTPro and SyPTLite

SyPTPro provides the development platform for the Applications module. For the SM-Applications Lite module, SyPTLite is available. It is possible to write a program within SyPTPro for an SM-Applications Lite module but there are restrictions with the available memory. This is detailed within the SyPTPro Help file.

## 4.2 Connecting the PC to the Applications module

There are two methods of connecting the programming PC to the Applications module and these are outlined below:

### 4.2.1 CTNet<sup>♦♥</sup>

With a CTNet connection you may connect the PC to a network of drives thereby allowing you to program and control all the drives directly from the PC. However you will need to have a CTNet interface card in your PC. There are PCI and PCMCIA cards available from Control Techniques for desktop and laptop computers.

CTNet offers the quickest option for downloading and fast update rates for on-line monitoring.

Refer to section 3.4 *CTNet Connections<sup>♦♥</sup>* on page 13 for details of the CTNet connections on the Applications module.

### 4.2.2 EIA-RS485 Serial port<sup>♦♦♥</sup>

You can connect the PC to the RJ45 serial port on the front of the drive. Special pre-made leads are available from Control Techniques for this purpose. These leads are to connect from your pc by either EIA-RS232 to EIA-RS485 or USB to EIA-RS485 - these leads are used with other Control Techniques products that use a RJ45 EIA-RS485 connector such as the Commander SE and Commander SK.

The RJ45 connector is located under a small flap on the front of the Unidrive SP just below the keypad slot. The pin-outs of this connector are described in the *Unidrive SP User Guide*.

Safety Information
Introduction
Installation
<b>Getting Started</b>
Parameters
Programming
DPL
Communications
Freeze and Marker
CTSync <sup>♦♥</sup>
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

**Figure 4-1 Communications Cable**



A number of drives may be connected via their EIA-RS485 ports on an RTU network, thus allowing the user to control any of those drives. In addition, if a drive has an Applications module fitted these will be seen by SyPT Pro along with any other Applications modules connected to this over CTNet. This is part of the routing capability of the CTNetAPI. See section 4.3 *CTNetAPI routing* on page 20.

### 4.3 CTNetAPI routing

The CTNetAPI offers a routing capability that allows access to various drives or Applications modules in a system. The user is able to download to, and upload from, these from within SyPTPro.

### 4.4 Configuring Communications within SyPTPro

Before attempting to go *on-line* to the Applications module you must set SyPTPro up to use the correct communications protocol:

1. In the SyPTPro Configuration Editor select **PC Communications Settings** from the Run menu
2. If connecting via CTNet (SM-Applications only), select CTNet as the protocol and make sure the baud rate is correct (pressing the Help button will show details of the other settings).
3. If connecting via RS232/485 to the front of the drive select **CT-RTU** as the protocol and choose the appropriate RS232 COM port. Also ensure drive parameters **Pr 11.24** and **Pr 11.25** are set to **rtU** and **19200** respectively (these are the default values).
4. Press OK.

### 4.5 Creating a Unidrive SP node in SyPTPro

**Figure 4-2 Node icon**



A Unidrive SP node is created in the same way as other node (drive) types:

1. Insert a new node by selecting **Node** from the Insert menu or by double-clicking on the **Double-click to Insert Node** icon.

2. The **Node** properties will now be displayed. Enter in the relevant details for Node ID and Network. Then any information on Solutions Modules can be entered using the tabs at the top of the box.
3. Press OK.

The new node will now be a Unidrive SP.

## 4.6 Porting UD70 programs to the Applications module

If you intend to convert programs from the UD70 platform for Unidrive Classic to the Unidrive SP please read Figure 12 *Migration Guide* on page 104 for important information.

## 4.7 Using SyPTLite

The SyPTLite program provides an alternative development platform to SyPTPro for the Applications Lite module.

SyPTLite is the 'entrance' level programming tool which is delivered free-of-charge with the Unidrive SP and can be used straight out of the box for programming in quick ladder logic.

Connectivity to the PC is the same as mentioned in section 4.2 *Connecting the PC to the Applications module* on page 19.

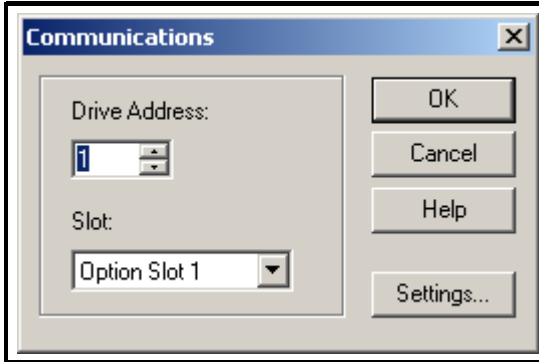
When using SyPTLite there is a choice of 3 targets when you create a new ladder diagram, Unidrive SP Onboard, SM-Applications Lite or Commander SK.

**Figure 4-3 Node Properties**

Upon choosing SM-Applications Lite, you have the option of 3 tasks to choose from. Background, Clock or Initial. You will only be able to use **one** of these tasks for the whole program.

In the 'Communications' settings from the drop-down menu the following box will appear.

**Figure 4-4 Communications Settings Dialogue**



All of the programming in SyPTLite is in ladder and you are limited to 10kb of executable memory.

For further information see Help in SyPTLite.

## 4.8 DPL Programming Basics

The Applications module can be programmed using a mixture of ladder diagrams (LD), function block diagrams (FBD) and DPL code (Drive Programming Language). Collectively they are known as a *DPL Program*. Only the ladder programming is available with SyPTLite.

At the very top level a program consists of:

- Program Header - giving the program title, author, version, etc. This is configured using the node properties dialogue box in SyPTPro.
- Program Body - comprised of *task* sections containing LD, FBD and DPL sections. This is created in the DPL Editor within SyPTPro.

Task sections encapsulate blocks of instructions that are to be executed by the microprocessor at a particular time, for example every 8ms or when the module first powers-up. Each task has a particular name, purpose and priority. Refer to section 6.2 *Tasks* on page 57 for further information.

### 4.8.1 Function Block Library

SyPTPro and SyPTLite come with an extensive library of pre-made function blocks. These perform tasks from simple things like a counter to more complex things such as PID loops or S-Ramp profile generators. These pre-supplied blocks are known collectively as the Function Block Library (FBL).

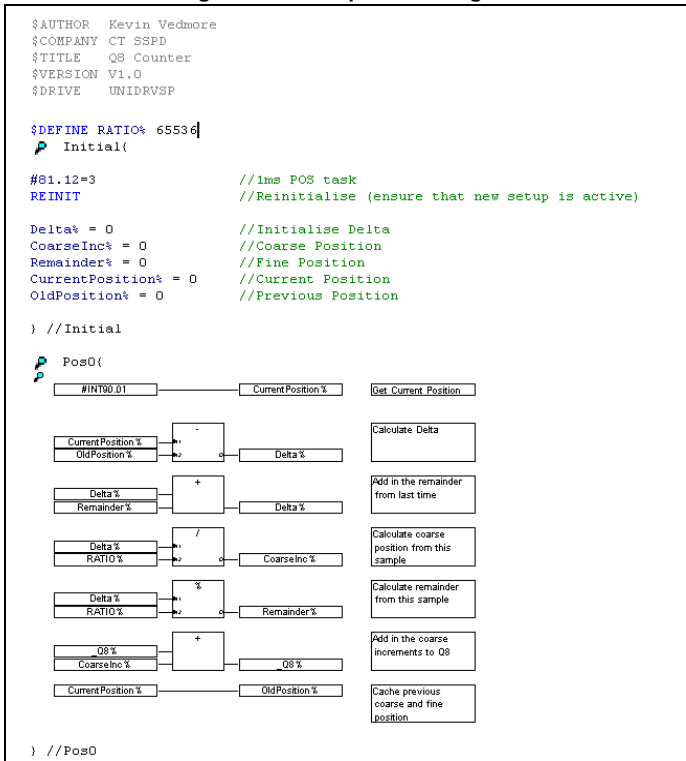
The functions in the FBL are documented in the on-line help.

You can also create your own function blocks within your program. So if you've created a new profile generator, you may encapsulate it within a user-defined function block (UDFB) and incorporate into your main DPL program. See section 6.7 *User Defined Function Blocks* on page 68 and the on-line help for information.

## 4.9 Program Example

Shown below is an example of a DPL program written within SyPTPro:

Figure 4-5 Example DPL Program



This program will take the positional feedback information from the drive (which is scaled to  $2^{32}/\text{rev}$ ), work out the delta (which will be proportional to speed) and convert to encoder counts (based on a standard quadrature encoder) and add this to an accumulator.

This example shows the basic concepts of accessing parameters and using mathematical functions. It may be useful to people migrating from the UD70 platform for Unidrive Classic since it will show how to re-create the `_Q8%` accumulative encoder position value as was available on that product.

Running through the program there are four distinct sections:

- Header section
- An **Initial** task
- A **Pos0** task
- A Function block diagram

### 4.9.1 Header Section

This section is automatically generated by SyPTPro from the details in the node properties dialogue box. It contains information such as the program's title, author and version.

## 4.9.2 Initial Task

As explained later in section 6.2 *Tasks* on page 57, this is a *task* which is executed when the Applications module is first powered up or is reset, providing the Autorun parameter is set (refer to the section on saving parameters). In this task are some DPL statements that initialise some integer variables (denoted by a trailing % symbol) to zero.

## 4.9.3 Pos0 Task and the Function Block Diagram

Because this program will be dealing with position feedback information, the bulk of the work will be done in the POS0 task. Any operations involving speed, position or torque control are usually done in the POS0, and POS1 or the CLOCK task which is now synchronised to the drive. In this case there is a single function block diagram which does all the calculations we need to work out the incremental encoder position.

The basic steps taken are:

1. Read the current encoder feedback value
2. Subtract the previously read encoder feedback value to give us the delta.
3. Re-scale the value to actual encoder counts, assuming a standard incremental (rather than SinCos type) encoder.
4. Add this delta to an accumulator
5. Remember the current encoder position for next time.

In this example program a variable, `_Q8%`, is used. This is a 32-bit value just like any other variable, but is part of a special set of *registers* known as the PLC register set. These PLC registers have the advantage of being able to be saved into non-volatile memory and also are accessible via parameters in menu 70 through to 76. More information on these can be found in section 5.4 *Menus 70-75 - PLC Registers* on page 37.

**NOTE** If you wish to create and try this program yourself and you have not used SyPTPro software before then it is advised to read through the remainder of this chapter first, then read the **Getting Started** section of *SyPTPro Help* which explains how to create such a program.

In order to ensure that the POS0 task is executed, parameter **Pr 81.12** must be set to a non-zero value in the Initial task. After setting this, a REINIT command must be issued (see below).

```
#81.12 = 3 //Pos task schedule period 1ms
REINIT //Reinitialise
```

## 4.10 Downloading Programs

By default, programs can only be downloaded to the Applications module when the drive enable signal is not active (**Pr 6.15**=0). This behaviour can be disabled by setting **Pr 81.37** to 0.

## 4.11 SyPTPro Caveats

The following are known issues in using SyPTPro with the Applications module:

- **Breakpoints in POS0/1 tasks**  
If you set a breakpoint in either of these tasks, the breakpoint will occur but the flag will not be shown and it is only possible to clear the breakpoint by resetting the option module.
- **EVENTx tasks**  
It is not possible to single-step or place breakpoints in any of the EVENT tasks

---

# 5 Parameters

---

## 5.1 Overview

The Applications modules contains two parameter databases:

- The Unidrive SP database  
This contains the entire drive parameter set. The Applications module caches this database in its own non-volatile *flash* memory. At power-up the module will check to see if this cache matches that of the drive. If it doesn't the database will be loaded from the drive, during which time the word "Loading" will appear for a few seconds on the drive display. This will not occur again unless the Applications module is moved to a different drive with different firmware or the drive firmware is updated.
- The Applications module database  
This database contains all parameters held locally to the module such as PLC registers as well as any other short-cut parameters (menus 90, 91, etc.)

## 5.2 Saving Parameters

There are different ways of saving parameters depending on the type of parameter that needs to be saved. These are explained in detail in the following sections:

### 5.2.1 Saving Applications module Parameters

The parameters that are saved to the Applications modules when performing the actions shown below are:

- Menus 70, 71, 74 and 75 (equivalent to P, Q, T and U register sets)
- Menu 20

To save the parameters on demand:

1. Set **Pr 81.19** to 1
2. Press the reset button  
Parameter **Pr 81.19** will reset to zero automatically and the module and drive will be reset.

To save the parameters on Under Voltage (UU):

- Set **Pr 81.20** to 1

Note that simply performing the above operations will not save menu 20. To save menu 20 you will need to perform the above operations but ensure that parameter **Pr 81.21** is set to 1 before doing so. This parameter does not need a module reset for the changes to become active.

### 5.2.2 Restoring menu 20 parameters

To restore menu 20 parameters on power-up, parameter **Pr 81.21** needs to be at 1 at power-up therefore a drive parameter save is required. See section 5.2.3 *Saving Drive Parameters*.

Safety Information
Introduction
Installation
Getting Started
<b>Parameters</b>
DPL Programming
Communications
Freeze and Marker
CT Sync
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

### 5.2.3 Saving Drive Parameters

The parameters that are saved to the drive when performing the actions shown below are:

- Menus 1 thru 14, 18, 19, 21 and 22.
- Menus 15, 16 and 17, if a module is present in the relevant slot.

To save the drive parameters:

1. Set **Pr 0.00**=1000 (parameter zero in any menu when using drive keypad)
2. Set **Pr 10.38**=100 (simulates pressing the reset button on the drive keypad)

## 5.3 Configuration Parameters

The basic configuration (or setup) parameters are held in the appropriate menu for the slot where the Applications module is fitted.

Slot	Menu
1	15
2	16
3	17

In addition to these menus, an alias of the appropriate menu is available as local menu 81 within the Applications module. This menu can be accessed from the user DPL program or via communications (CTNet/CT-RTU/EIA-RS485) and provides a convenient way to read or change the setup parameters without having to know which slot the Applications/Lite module is fitted in.



Unless otherwise indicated, these parameters are only read when the Applications module is first powered up, on a reset or on a *REINIT* DPL command. Changing one of these parameters on the fly will have no immediate effect.

To reset the Applications module from the drive display, enter the value of 1070 in parameter zero of any menu and press the reset button.

#### NOTE

A mark next to the following parameters will indicate whether that parameter is available for a particular variant of the Applications module. The mark for each variant is as follows:

- ♦SM-Applications Lite
- ▼SM-Applications
- \*SM-Applications Plus

#### NOTE

Throughout this User Guide, the configuration parameters will be referred to as **Pr 81.XX**. When setting parameters directly on the drive keypad use the appropriate menu 15, 16 or 17 instead.

#### NOTE

The update rate specified for any parameter refers to the rate at which the parameter is updated for reading or when writing, when the new value takes effect. "Initialisation" means that the parameter is read only on module reset or REINIT DPL command.



Changing the drive mode will clear all configuration and application parameters back to their default value as well as drive parameters. This can be avoided by using the code **1255** in parameter zero rather than the usual **1253**. Only drive parameters will be defaulted, and menus 15 to 20 will be left unchanged.

### 5.3.1 Parameter Descriptions

<b>Pr 81.01</b>	<b>Module Code</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	0 to 499
<b>Default</b>	N/A	<b>Update Rate</b>	N/A

The Module Code indicates the type of module that is fitted in the corresponding slot. For an SM-Applications module the code is **301**, for SM-Applications Lite module the code is **302**, and for the SM-Applications Plus module the code is 304.

<b>Pr 81.02</b>	<b>Firmware Version - Major</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	00.00 to 99.99
<b>Default</b>	N/A	<b>Update Rate</b>	N/A

Specifies the major revision number of the operating system of the Applications module. Use in conjunction with **Pr 81.51** to form the complete version number.

<b>Pr 81.03</b>	<b>DPL Program Status</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	0 to 3
<b>Default</b>	0	<b>Update Rate</b>	1ms of change

Provides the run status of the user DPL program in the Applications module. The following values are defined:

Display	Value	Description
nonE	0	No DPL program present
StoP	1	DPL program is stopped
run	2	DPL program is running
triP	3	Run-time error. ERROR task running or DPL program stopped

<b>Pr 81.04</b>	<b>Available System Resource</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	0 to 100
<b>Default</b>	N/A	<b>Update Rate</b>	200ms

Displays the free CPU resource as a percentage of the current background execution time calculated over 200ms.

<b>Pr 81.05</b>	<b>EIA-RS485 Address</b> ♦♥		
<b>Access</b>	RW	<b>Range</b>	0 to 255
<b>Default</b>	11	<b>Update Rate</b>	Initialisation

Defines the address of this node for ANSI and Modbus communications protocols.

For the ANSI protocol the address range is 11 to 99 where the first digit is the group address and the second digit is the unit number. Both digits must be in the range of 1-9. Zero is not permitted since it is used by the master for addressing groups of nodes.

This parameter has no effect if the EIA-RS485 mode is 25 (CTSync) or 26 (CTSync)

<b>Pr 81.06</b>	<b>EIA-RS485 Mode<sup>♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 255
<b>Default</b>	1	<b>Update Rate</b>	Initialisation

Defines the mode of operation (or protocol) for the on-board EIA-RS485 port. For details of these modes, refer to Table 5.1 *Serial modes - parameter Pr 81.06* ..

**Table 5.1 Serial modes - parameter Pr 81.06**

<b>Mode</b>	<b>Description</b>
1	4-wire CT-ANSI Slave The port is set to 1 start bit, 7 data bits, even parity and 1 stop bit.
2	Reserved
3	Reserved
4	Reserved
5	2-wire CT-ANSI Slave The port is set to 1 start bit, 7 data bits, even parity and 1 stop bit.
6	User mode. 1 start bit, 7 data bits, EVEN parity, 1 stop bit (10 bits total)
7	User mode. 1 start bit, 8 data bits, EVEN parity, 1 stop bit (11 bits total)
8	User mode. 1 start bit, 8 data bits, NO parity, 1 stop bit (10 bits total)
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13,43,73	4-wire Modbus RTU slave The EIA-RS485 port is set for: Mode 13: 1 start bit, 8 data bits, NO parity, 2 stop bits. Mode 43: 1 start bit, 8 data bits, EVEN parity, 1 stop bit. Mode 73: 1 start bit, 8 data bits, ODD parity, 1 stop bit.
14, 44, 74	4-wire Modbus ASCII slave The EIA-RS485 is set for: Mode 14: 1 start bit, 7 data bits, NO parity, 2 stop bits Mode 44: 1 start bit, 7 data bits, EVEN parity, 1 stop bit Mode 74: 1 start bit, 7 data bits, ODD parity, 1 stop bit
15, 45, 75	2-wire Modbus RTU slave The EIA-RS485 port is set for: Mode 15: 1 start bit, 8 data bits, NO parity, 2 stop bits. Mode 45: 1 start bit, 8 data bits, EVEN parity, 1 stop bit. Mode 75: 1 start bit, 8 data bits, ODD parity, 1 stop bit.
16, 46, 76	2-wire Modbus ASCII slave The EIA-RS485 is set for: Mode 16: 1 start bit, 7 data bits, NO parity, 2 stop bits Mode 46: 1 start bit, 7 data bits, EVEN parity, 1 stop bit Mode 76: 1 start bit, 7 data bits, ODD parity, 1 stop bit
17, 47, 77	4-wire Modbus RTU master The EIA-RS485 port is set for: Mode 17: 1 start bit, 8 data bits, NO parity, 2 stop bits. Mode 47: 1 start bit, 8 data bits, EVEN parity, 1 stop bit. Mode 77: 1 start bit, 8 data bits, ODD parity, 1 stop bit.
18, 48, 78	4-wire Modbus ASCII master The EIA-RS485 is set for: Mode 18: 1 start bit, 7 data bits, NO parity, 2 stop bits Mode 48: 1 start bit, 7 data bits, EVEN parity, 1 stop bit Mode 78: 1 start bit, 7 data bits, ODD parity, 1 stop bit

**Table 5.1 Serial modes - parameter Pr 81.06**

Mode	Description
19, 49, 79	2-wire Modbus RTU master The EIA-RS485 port is set for: Mode 19: 1 start bit, 8 data bits, NO parity, 2 stop bits. Mode 49: 1 start bit, 8 data bits, EVEN parity, 1 stop bit. Mode 79: 1 start bit, 8 data bits, ODD parity, 1 stop bit.
20, 50, 80	2-wire Modbus ASCII master The EIA-RS485 is set for: Mode 18: 1 start bit, 7 data bits, NO parity, 2 stop bits Mode 48: 1 start bit, 7 data bits, EVEN parity, 1 stop bit Mode 78: 1 start bit, 7 data bits, ODD parity, 1 stop bit
25	Applications CT-Sync Master The baud rate is fixed at 896875bps
26	Applications CT-Sync Slave The baud rate is fixed at 896875bps

Pr 81.07	EIA-RS485 Baud Rate <sup>✱♥</sup>		
<b>Access</b>	RW	<b>Range</b>	0-9 (300-115200bps)
<b>Default</b>	4 (4800)	<b>Update Rate</b>	Initialisation

Defines the baud-rate (or bits-per-second) for the on-board EIA-RS485 port. The following are supported:

Display	Value	Description
300	0	300bps
600	1	600bps
1200	2	1200bps
2400	3	2400bps
4800	4	4800bps
9600	5	9600bps
19200	6	19200bps
38400	7	38400bps
57600	8	57600bps
115200	9	115200bps

Note that this parameter is not relevant when the EIA-RS485 port mode is set to 25 (CTSync Master) or 26 (CTSync Slave).

Pr 81.08	EIA-RS485 Turn-around Delay <sup>✱♥</sup>		
<b>Access</b>	RW	<b>Range</b>	0 to 255ms
<b>Default</b>	2ms	<b>Update Rate</b>	Initialisation

Defines a fixed delay between receiving a message on the EIA-RS485 port and the response being transmitted. This can be useful in 2-wire configurations where it takes a finite time for the master (host) to switch from transmit mode to receive mode. There is always at least a 1ms delay and this parameter can be used to extend it.

Safety Information
Introduction
Installation
Getting Started
Parameters Programming
DPL
Communications
Freeze and Marker
CTSync <sup>✱♥</sup>
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

<b>Pr 81.09</b>	<b>EIA-RS485 Tx Enable Delay<sup>▲▼</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 1ms
<b>Default</b>	0ms	<b>Update Rate</b>	Initialisation

This parameter allows a 1ms delay to be introduced between the Applications module enabling the EIA-RS485 transmitter and actually commencing the transmission. This should only be required if it is found that the recipient of the transmission is receiving a corrupted start of message.

<b>Pr 81.10</b>	<b>DPL Print Routing<sup>◆▲▼</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 <sup>◆▲▼</sup> /1 <sup>▲▼</sup>
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

This parameter is available on SM-Applications Lite module but is not allowed to be set to a 1 (On). Controls where the output of the DPL PRINT command is sent. If set to zero (Off), the output is sent to the programming client (SyTPPro) and if set to 1 (On) it will be sent to the EIA-RS485 port

<b>Pr 81.11</b>	<b>Clock Task Scheduling (ms)<sup>◆▲▼</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 200ms
<b>Default</b>	10ms	<b>Update Rate</b>	Initialisation

Defines the scheduling period (tick-time), in milliseconds, for the DPL CLOCK task. A value of zero will disable the CLOCK task.

**NOTE** Prior to version 01.05.00 Unidrive SP the default for this parameter was 0ms (disabled).TODO

<b>Pr 81.12</b>	<b>POS task scheduling rate<sup>◆▲▼</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 6
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

Defines the scheduling rate for the POS tasks to suit the application performance and the resource needed to run the user DPL program. The following values are defined::

Display	Value	Description
diSAbled	0	Disabled
0.25	1	250µs
0.5	2	500µs
1	3	1ms
2	4	2ms
4	5	4ms
8	6	8ms

Set this parameter in order for the user DPL program to automatically run at power-on/reset. If this is changed and the new setting needs to be used on power-up ensure that a **drive** parameter save is performed.

**NOTE** With the Unidrive SP version 01.03.00 and earlier, the display will not show the actual rate, but its aliased number. For instance 250µs will be shown on the display as 1.

<b>Pr 81.13</b>	<b>Auto-run Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	1	<b>Update Rate</b>	Initialisation

<b>Pr 81.14</b>	<b>Global Run-time Trip Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

Setting this parameter to 1 will cause the Unidrive SP to trip when certain run-time errors occur within the Applications module user DPL program.

For more information, see section 11.1 *Run-time Errors* on page 98.

<b>Pr 81.15</b>	<b>Disable Reset on Trip Cleared</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

When this parameter is 0, the module will be reset when a drive trip is cleared. When set to 1 the module will be unaffected by a drive trip reset (i.e. continue running).

<b>Pr 81.16</b>	<b>Encoder Data Update Rate</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0-3
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

When this parameter is 0 the APC data and menu 90 encoder parameters are updated every 250µs.

When this parameter is 1 the APC data and menu 90 encoder parameters are updated immediately prior to every POS task.

When this parameter is 2 the APC data and menu 90 encoder parameters are updated immediately prior to every CLOCK task.

When 3 the APC data and menu 90 encoder parameters are never updated. If these are never updated, more processor resource will become free.

<b>Pr 81.17</b>	<b>Enable Parameter Over-range Trips</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

Defines the action taken if a user DPL program attempts to write an out of range value to a parameter. When set at 1, a run-time trip will occur (number 44); when set at zero the value will automatically be limited to the maximum/minimum of that parameter.

<b>Pr 81.18</b>	<b>Watchdog Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

When set it enables the DPL program watchdog function. The DPL WDOG command must then be executed every 200ms. This can be used to protect the program against malfunction. If the command is not executed within a 200ms time period the drive will trip on **SLx.tO**. Please note that the WDOG command must also be executed once for the watchdog to be enabled. This is normally executed at the end of the Initial task.

<b>Pr 81.19</b>	<b>Save Request</b> ♦ ♦ ♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	100ms

Setting this parameter to 1 will initiate an immediate save of all non-volatile Applications module data. This consists of the P/Q/T/U PLC register sets and optionally menu 20 (depending upon the setting of **Pr 81.21**).

**NOTE** Note that this will also cause a reset of the module and this parameter will return back to zero automatically. Also if the Unidrive SP is tripped, it will be reset. Menu 81 will not be saved.

<b>Pr 81.20</b>	<b>Enable "UU trip" Save</b> ♦ ♦ ♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Immediate

Setting this parameter to 1 signals that all non-volatile data of the Applications module will be automatically saved upon an under voltage (UU) trip of the drive.

**NOTE** Note that when a 'UU' save occurs the Applications module will be reset.

<b>Pr 81.21</b>	<b>Enable menu 20 save and restore</b> ♦ ♦ ♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Immediate

If set to 1, menu 20 will be saved/restored along with other non-volatile parameters upon a save request (**Pr xx.19=1**) or power-down save (**Pr xx.20=1**). If menu 20 is to be restored on power-up the user must ensure that this parameter is saved in the drive before powering down.

Since menu 20 is a global drive menu, only one option fitted to the Unidrive SP should be used to store and restore menu 20, therefore if more than one Applications module is fitted to the drive **only one** should have this parameter set otherwise menu 20 will not be restored correctly on power-up.

**NOTE** Unlike other setup parameters, parameters **Pr 81.20** and **Pr 82.21** are **not cached**, which means a change to the parameter takes immediate effect.

<b>Pr 81.22</b>	<b>CTNet Token Ring ID</b> ♦ ♥		
<b>Access</b>	RW	<b>Range</b>	0 to 255
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

This parameter allows the user to specify the identity of the CTNet token ring to which a Applications module is connected. In a system incorporating a single token ring this parameter can be left at its default value. In a system incorporating multiple token rings, separate ID's should be set for each ring. The combination of CTNet Token Ring ID and CTNet node address should be unique.

<b>Pr 81.23</b>	<b>CTNet Node Address<sup>✱♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 255
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

Defines the node address for CTNet. Every node on a CTNet network must have a unique address. Setting this to zero will disable CTNet on this node.

<b>Pr 81.24</b>	<b>CTNet Baud Rate<sup>✱♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 3
<b>Default</b>	1(2.5)	<b>Update Rate</b>	Initialisation

Specifies the data rate for CTNet. All nodes on the network must be set to the same data-rate. The rates are defined as follows:

Display	Value	Description
5.000	0	5Mbit/s
2.500	1	2.5Mbit/s
1.250	2	1.25Mbit/s
0.625	3	625kbit/s

<b>Pr 81.25</b>	<b>CTNet Sync Setup<sup>✱♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 9999
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

Specifies the synchronisation message generation rate for CTNet. This message is used to tell all nodes when to transmit cyclic data. Only one node on the CTNet network should have this parameter set.

The format of the update parameter is SSFF, where FF defines the Fast Cyclic data channel update rate, and SS defines the slow cyclic data rate in multiples of FF. So if the parameter value is 1510, fast cyclic data is set every 10ms and slow every 150ms. When using easy mode (see below) it is only necessary to set up the FF (fast cyclic rate).

<b>Pr 81.26, Pr 81.28, Pr 81.30</b>	<b>CTNet Easy Mode Setup Parameters<sup>♣♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 25503
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

<b>Pr 81.27, Pr 81.29, Pr 81.31- Pr 81.34</b>	<b>CTNet Easy Mode Setup Parameters<sup>♣♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 9999
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

These parameters define the source and destinations for CTNet easy-mode cyclic data.

Parameter	Format	Channel	Description
Pr 81.26	NNSS	1	Defines the destination node number and slot NNN = Node number (1-255) SS = Slot number (1-3) e.g. A value of 201 means node ID 2, slot 1.
Pr 81.27	MMPP	1	Defines the source drive parameter which is to be transmitted MM = Menu number PP = Parameter number e.g. A value of 302 means <b>Pr 3.02</b> (speed)
Pr 81.28	NNSS	2	Destination node number and slot for channel 2
Pr 81.29	MMPP	2	Source drive parameter for channel 2
Pr 81.30	NNSS	3	Destination node number and slot for channel 3
Pr 81.31	MMPP	3	Source drive parameter for channel 3
Pr 81.32	MMPP	1	Slot 1 destination parameter for incoming data
Pr 81.33	MMPP	2	Slot 2 destination parameter for incoming data
Pr 81.34	MMPP	3	Slot 3 destination parameter for incoming data

More information on CTNet easy mode data can be found in the *CTNet User Guide*.

<b>Pr 81.35</b>	<b>CTNet Sync. Event Task ID<sup>♣♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 4
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

Identifies which of the EVENT tasks will be scheduled when a CTNet synchronization message is received or generated. This sync. is generated by a *master* node (which can be this node) on the CTNet network at a fixed time-base. The following values are defined:

Display	Value	Description
Disabled	0	No event task scheduled
Event	1	EVENT task scheduled
Event1	2	EVENT1 task scheduled
Event2	3	EVENT2 task scheduled
Event3	4	EVENT3 task scheduled

<b>Pr 81.36</b>	<b>CTNet Diagnostics<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	-3 to 32767
<b>Default</b>	N/A	<b>Update Rate</b>	1 second

The status of the CTNet network is displayed in the CTNet Diagnostic parameter. When the Applications module is communicating successfully on the CTNet network the number of messages per second is displayed

**Table 5.2 CTNet Diagnostics**

#MM.36	Status	Description
>0	Network Healthy	Indicates the number of messages per second be processed every second.
0	Network Healthy, No Data Transfer	The low-level token ring has been established and is active, but the node is not receiving any CTNet data messages.
-1	RECON	A network reconfiguration has been detected.
-2	Initialisation Error	The Applications module was unable to configure the CTNet interface. Check that the node address and data rate are set correctly.
-3	MYRECON	The Applications module forced a CTNet network reconfiguration

<b>Pr 81.37</b>	<b>Reject Download if Drive Enabled<sup>♦♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

If this parameter is set, then if the user attempts to download a new user DPL program or operating system to this module and the drive is enabled the download will be rejected and a run-time trip 70 will occur, if the global run-time trip parameter (**Pr 81.14**) is set.

Since downloading stops normal operations of the module it may be considered unsafe to do this if the drive system is running, therefore setting this parameter will prevent downloading under this condition.

<b>Pr 81.38</b>	<b>APC Run-time trip<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

When this parameter is 0 the drive will trip with runtime error 81 if an APC non-recoverable error occurs, such as use of an uninitialised CAM function. When this parameter is 1 the drive will not trip when an APC non-recoverable error occurs.

<b>Pr 81.39</b>	<b>Inter-module Drive Sync Status<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	NA

This parameter displays the current module's synchronisation status.

Synchronisation Status	Status
0	The synchronisation master request is zero or another option module is synchronisation master.
1	The option module is synchronisation master.
3	The option module is synchronisation master, but the synchronisation frequency is out of specification or not present.

<b>Pr 81.42</b>	<b>Pass Freeze Through to Drive<sup>♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

When this parameter is ON (1), the voltage on the Applications module's digital input 0 (zero) is passed through to the drive's internal Freeze line. This can be seen by other classes of SM-Solutions modules. For further information on the Freeze Input refer to Chapter 8 *Freeze and Marker* on page 79.

<b>Pr 81.43</b>	<b>Freeze Invert<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

♦♥ When this parameter is set to zero a freeze occurs on the rising edge of the module's DIGIN0. When it is set to 1 a freeze occurs on the falling edge of the module's DIGIN0. For further information on the Freeze Input refer to Chapter 8 *Freeze and Marker* on page 79.

♦ When this parameter is set to zero the module will freeze data when it sees a rising edge of a freeze pulse from the drive. When it is set to a 1 the module will freeze data when it sees a falling edge of a freeze pulse from the drive. For further information on the Freeze Input refer to Chapter 8 *Freeze and Marker* on page 79.

<b>Pr 81.44</b>	<b>Task Priority Level<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 255
<b>Default</b>	0	<b>Update Rate</b>	Initialisation

The priority levels of different tasks may be changed with this parameter. The parameter is accessed in a bit-wise manner:

Bit	Value	Meaning
0 <sup>♦♥</sup>	0	CTNet task priority is higher than Pos tasks priority.
	1	CTNet task priority is lower than Pos tasks priority. This will reduce the jitter of the POS tasks but could lead to the CTNet task being starved
1 <sup>♦♦♥</sup>	0	Inter-option communication task priority is higher than the POS tasks.
	1	Inter option communication task priority lower than the POS tasks.
2 <sup>♦♥</sup>	0	Turbo CTNet Disabled
	1	Turbo CTNet Enabled

<b>Pr 81.48</b>	<b>Line Number of Error<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	32 bit
<b>Default</b>	0	<b>Update Rate</b>	On error

Specifies the DPL program line number that caused a run-time error. This is valid only when:

- The user program has been compiled with the *debug* option set
- The error is one that can be generated by user code, for example divide by zero (50) or parameter does not exist (41).

If both of these conditions are not met, the line number parameter will display zero (0).

<b>Pr 81.49</b>	<b>User program ID ♦♦♥</b>		
<b>Access</b>	RO/RW	<b>Range</b>	Signed 16-bit
<b>Default</b>	0	<b>Update Rate</b>	See Note

This parameter is available for the user to put in an ID code of their program. This may, for example, be the software version number. Use the function block SETUSERID() to write to this parameter.

<b>Pr 81.50</b>	<b>Run-time Error Code ♦♦♥</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 255
<b>Default</b>	0	<b>Update Rate</b>	On error

When a run-time error occurs the error number is placed into this parameter. See Chapter 11.1 *Run-time Errors* on page 98 for further information.

<b>Pr 81.51</b>	<b>Firmware - Minor Version ♦♦♥</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 99
<b>Default</b>	N/A	<b>Update Rate</b>	N/A

Specifies the minor revision number of the operating system of the Applications module. Use in conjunction with **Pr 81.02** to form the complete version number.

## 5.4 Menus 70-75 - PLC Registers

These menus provide access to the PLC registers. The PLC registers are Signed 32-bit integers available for user programs and CTNet communications.

The PLC registers are split into 6 sets of 100 parameters numbered 00 to 99. The registers can also be accessed from within a user DPL program by a special variable name or array name.

Menu Number	Access	DPL variable (x=register number)	Number of Registers	Description
70 ♦♦♥	RW	_Px%, _P[x]	100	General purpose. Saveable.
71 ♦♦♥	RW	_Qx%, _Q[x]	100	General purpose. Saveable.
72 ♦♦♥	RW	_Rx%, _R[x]	100	Used for outgoing CTNet cyclic data links. Non-saveable.
73 ♦♦♥	RW	_Sx%, _S[x]	100	Used to incoming CTNet cyclic data links. Non-saveable.
74 ♦♦♥	RW	_Tx%, _T[x]	100	General purpose. Saveable.
75 ♦♦♥	RW	_Ux%, _U[x]	100	General purpose. Saveable.

You will see from the above table that each parameter within menus 70 to 75 has an equivalent DPL variable. This means that you can use either format for accessing a parameter within these menus.

e.g. **Pr 72.01=1** will do the same as **\_R01%=1**, **Pr 75.65=66** will do the same as **\_U65%=66** etc.

Menus 70, 71, 74 and 75 can all be saved into the non-volatile flash memory upon request or automatically when the drive goes into under-voltage (Refer to section 5.2 *Saving Parameters* on page 25 for more information).

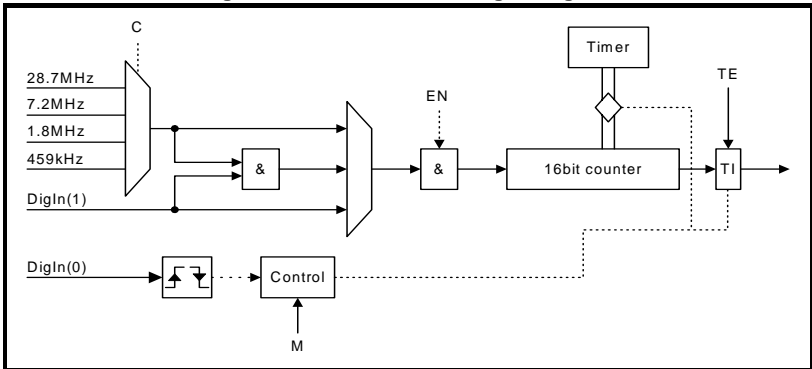
Menus 72 and 73 are used for CTNet cyclic data transfer but if this feature is not being used the registers may be used for any other purpose. However this should be avoided if possible in case cyclic data is used in the future.

## 5.5 Menu 85 - Timer Function Parameters<sup>♣♥</sup>

A hardware/counter is built into the Applications module which has the following features:

- A 16 bit incremental counter.
- Count rate selectable from the internal clock. The clock rate divisor is selectable from rate/ 1, rate/ 4, rate/ 16, rate/ 64.
- Count rate selectable from an external clock via the DIGIN1 digital input. The maximum clock rate is 600kHz.
- The timer can be used to schedule one of the 4 DPL Event tasks on wrap-around or an input capture on DIGIN1.
- Counter overflow can be selected by the user up to the full 16 bit range available for the counter.
- The timer can be set to cache the count on a DIGIN0 rising or falling edge transition.

**Figure 5-1 Counter/Timer Logic Diagram**



Pr 85.01	Timer Unit Control Word <sup>♣♥</sup>		
Access	RW	Range	13 bit
Default	N/A	Update Rate	Immediate

**Table 5.3 Control Word - parameter Pr 85.01**

Bit	Symbol	Function
b0-b2	TE	Timer EVENT task schedule when the TI flag is set: 0=No Event task scheduled 1=Schedule Event task 2=Schedule Event1 task 3=Schedule Event2 task 4=Schedule Event3 task
b3	EN	Enable Timer: 0=Timer is disabled 1=Timer is enabled
b4	CS	Clock source: 0=internal clock 1=external clock provided on DIGIN1.
b5-b6	C	Internal clock pre-scale select (ignored if external clock selected): 0=rate /1 (28.7MHz) 1=rate /4 2=rate /16 3=rate /64

**Table 5.3 Control Word - parameter Pr 85.01**

Bit	Symbol	Function
b7-b8	M	<p>Timer mode:</p> <p><u>Mode 0 (00): Free Running mode</u> The selected clock drives the counter. The TI flag is set on wrap-around.</p> <p><u>Mode 1 (01): Capture mode 1</u> The selected clock drives the counter. A rising edge transition on DIGIN0 causes the current counter value to be latched into the TIMER CAPTURE CACHE parameter and the TI flag is set. The counter then continues incrementing (TI is not set on wrap-around).</p> <p><u>Mode 2 (10): Capture mode 2</u> The selected clock drives the counter. A falling edge transition on DIGIN0 causes the current counter value to be latched into the TIMER CAPTURE CACHE parameter and the TI flag is set. The counter then continues incrementing (TI is not set on wrap-around).</p>

Pr 85.02	Timer Unit Status Word <sup>⚡♥</sup>		
<b>Access</b>	RO	<b>Range</b>	0 to 3
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

**Table 5.4 Status Word - parameter Pr 85.02**


Bit	Symbol	Function
b0	TI	<p>Timer event flag: 0=No event has occurred 1=An event has occurred (see description for parameter <b>Pr 85.01</b>)</p> <p><b>Note:</b> The TI bit is automatically cleared by the operating system if TE flag in <b>Pr 85.01</b> has a non-zero value. Otherwise it is cleared when the status word is read.</p>
b1	OV	<p>Wrap-around flag 0=Wrap-around has not occurred 1=Counter wrap-around has occurred</p> <p><b>Note:</b> This flag is valid for ALL timer modes and is automatically cleared when the status register is read.</p>

Pr 85.03	Timer Unit 16-bit Timer Count <sup>⚡♥</sup>		
<b>Access</b>	RW	<b>Range</b>	16 bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

The current timer value can be read and written at any time using this parameter.

Pr 85.04	Timer Unit Wrap-around limit <sup>⚡♥</sup>		
<b>Access</b>	RW	<b>Range</b>	16 bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

This parameter specifies the value at which parameter **Pr 85.03** will wrap-around.



Setting a small wrap-around figure with the timer operating at fast rates, could result in the Applications module appearing to lock up. This is caused by the wrap-around interrupt being continually serviced and starving the other product features of processor resource. If this occurs the user can reset the Applications module by performing a 1070 reset on the drive. If a DPL program sets the figures for the timer then, before resetting, the program auto-run should be disabled (**Pr xx.13=0**). After the reset the user can reinitialise the timer with a more reasonable wrap-around number.

<b>Pr 85.05</b>	<b>Timer Unit Timer Capture Cache<sup>±♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	16 bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

This parameter stores the cached value when using the timer in modes 1 or 2 (Capture modes)

## 5.6 Menu 86 - Digital I/O Parameters

The SM-Applications and SM-Applications Plus modules have two digital outputs and two digital inputs on-board. These outputs and inputs are controlled through this menu.

<b>Pr 86.01</b>	<b>Digital Input 0<sup>±♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

<b>Pr 86.02</b>	<b>Digital Input 1<sup>±♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

These two parameters read the state of digital inputs 0 and 1. Inactive input (low) will give the value 0 and active input (high) will give 1.

<b>Pr 86.03</b>	<b>Digital Output 0<sup>±♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

<b>Pr 86.04</b>	<b>Digital Output 1<sup>±♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

These two parameters control digital outputs 0 and 1. Setting to 0 will place the output low and setting to 1 will place the input high (+24V).

<b>Pr 86.05</b>	<b>Digital Outputs 0 and 1<sup>±♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 3
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

This parameter provides control for both digital outputs and is an alternative to controlling each output individually with **Pr 86.03** and **Pr 86.04**. Bit0 of this parameter controls digital output 0 (**Pr 86.03**) and Bit1 controls digital output 1 (**Pr 86.04**).

## 5.7 Menu 88 - Status Parameters

Pr 88.01	Error code / Reset <sup>♦**▼</sup>		
<b>Access</b>	RW	<b>Range</b>	0 to 9999
<b>Default</b>	N/A	<b>Update Rate</b>	On error

This parameter has two purposes - when read it will return the identical run-time error as **Pr 81.50** (note - it will not return drive trip codes). The parameter is cleared to zero on reset and when the user program execution is started.

When the parameter is written to with a value of 1070 the Applications module will initiate a warm-restart of the drive and any other options. This can be used to restart the user program (providing auto-run **Pr 81.13**=1) and clear any drive trip. This reset action can be performed at any time, not just after a run-time error or in an ERROR task.



Writing 1070 to parameter **Pr 88.01** will result in any drive trip being automatically cleared as well as resetting all fitted options in the Unidrive SP. This behaviour is different to the UD70 product on Unidrive 1 where the drive was not reset.

Pr 88.02	Task in Error <sup>♦**▼</sup>		
<b>Access</b>	RO	<b>Range</b>	0 to 50
<b>Default</b>	N/A	<b>Update Rate</b>	On Error

The Task in Error parameter can be used to identify which task the error was generated in. This parameter is only valid if it is read from the ERROR task after a run-time trip has occurred. The values will have the following meanings:

Value	Task
50	System
1	Initial
2	Background
3	Clock
4	Error
5	Pos0
6	Pos1
7	Event
8	Event1
9	Event2
10	Event3

A value of zero will be returned if there is no error condition.

For more information on these parameters refer to Chapter 11 *Diagnostics* on page 98.

Pr 88.03	POS Resource Monitoring <sup>♦</sup>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Immediate

This parameter allows the user to enable or disable monitoring of the motion engine tasks free resource. When set to 1, parameters **Pr 88.04** and **Pr 88.05** become active. If set to zero, parameters **Pr 88.04** and **Pr 88.05** will read zero.

<b>Pr 88.04</b>	<b>Free Resource for Motion Engine Tasks*</b>		
<b>Access</b>	RW	<b>Range</b>	0 to 95
<b>Default</b>	0	<b>Update Rate</b>	See Pr 81.12

This parameter indicates the percentage resource available for running the motion engine tasks. These tasks are CTSync, CTSync Output Channels, POS0, PLCopen, APC, APC Output Channel and POS1. If this parameter value reaches zero a task overrun will occur. It is calculated every motion engine period and is displayed for the previous motion engine period.

<b>Pr 88.05</b>	<b>Motion Engine Peak Resource Detect*</b>		
<b>Access</b>	RW	<b>Range</b>	0 to 95
<b>Default</b>	0	<b>Update Rate</b>	See Pr 88.04

This parameter displays the lowest value (i.e. highest resource usage) that parameter **Pr 88.04** reaches since the monitoring was enabled (parameter **Pr 88.03**). It will give a realistic indication of the worst case available resources for the motion engine tasks so that the user can see how close the module has been to a motion engine task overrun.

<b>Pr 88.06</b>	<b>CLOCK Task Resource Monitoring*</b>		
<b>Access</b>	RO	<b>Range</b>	0/1
<b>Default</b>	NA	<b>Update Rate</b>	Immediate

This parameter allows the user to enable or disable monitoring of the CLOCK task free resource. When set to 1, parameters **Pr 88.07** and **Pr 88.08** become active. If set to zero, parameters **Pr 88.07** and **Pr 88.08** will read zero.

<b>Pr 88.07</b>	<b>Free Resource for Clock Task*</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 95
<b>Default</b>	NA	<b>Update Rate</b>	See Pr 81.11

This parameter indicates the percentage resource available for running the Clock task. If this parameter value reaches zero a task overrun will occur. It is calculated every Clock period and is displayed for the previous Clock period.

<b>Pr 88.08</b>	<b>Clock Task Peak Resource Detect*</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 95
<b>Default</b>	NA	<b>Update Rate</b>	See Pr 81.11

This parameter displays the lowest value (i.e. highest resource usage) that parameter **Pr 88.07** reaches since the monitoring was enabled (parameter **Pr 88.06**). It will give a realistic indication of the worst case available resources for the Clock task so that the user can see how close the module has been to a Clock task overrun.

## 5.8 Menu 90 - General Parameters

This menu contains the reference and feedback values from the drive as well as other status information.

**NOTE**

When porting UD70 programs to Applications module special care must be taken as these parameters are different than those on UD70.

Pr 90.01	Feedback Encoder Position ( $2^{32}/\text{rev}$ ) <sup>♦ ♣ ♥</sup>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

Contains the feedback encoder position.

The top 16-bits are scaled to 65536 counts per rev regardless of the type of feedback device or scaling configured in the drive. The lower 16-bits give the fine position as available from the feedback device scaled to 65536. For standard encoders this will typically be zero, but for higher precision devices such as SinCos encoders, this extra precision will be available.

Marker pulses, etc. have no influence on this parameter.



More information on the use of these feedback parameters can be found in the on-line help of SyPTPro.

Pr 90.02	Feedback Encoder Revolution Count <sup>♦ ♣ ♥</sup>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

Contains the feedback encoder revolution count.

Pr 90.03	Reference Encoder Position ( $2^{32}/\text{rev}$ ) <sup>♦ ♣ ♥</sup>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

Contains the reference encoder position.

The top 16-bits are scaled to 65536 counts per rev regardless of the type of feedback device or scaling configured in the drive. The lower 16-bits give the fine position as available from the feedback device scaled to 65536. For standard encoders this will typically be zero, but for higher precision devices such as SinCos encoders, this extra precision will be available.

Marker pulses, etc. have no influence on this parameter.

Pr 90.04	Reference Encoder Revolution Count <sup>♦ ♣ ♥</sup>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

Contains the reference encoder revolution count.

Safety Information
Introduction
Installation
Getting Started
<b>Parameters</b>
DPL Programming
Communications
Freeze and Marker
CT/Sync <sup>♦ ♥</sup>
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

<b>Pr 90.10</b>	<b>Drive Mode</b> ♦♣♥		
<b>Access</b>	RO	<b>Range</b>	Signed 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

Provides a definitive method of identifying the mode the Unidrive SP is in. It is recommended that this parameter is used instead of **Pr 11.31** or **Pr 00.48** since those parameters indicate the requested, not the actual, mode.

The values are defined as follows:

Value	Mode
26	Open-loop
27	Closed-loop vector
28	Servo
29	Regen
20	Commander SE

In order to programmatically change the drive mode, use the MODEXFER or CMODEXFER function blocks.

<b>Pr 90.11</b>	<b>Drive Status and Control Word</b> ♦♣♥		
<b>Access</b>	RW	<b>Range</b>	Signed 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

Writing to this parameter updates the control word. Reading from this parameter reads the status word (same as parameter **Pr 10.40**).

**Table 5.5 Control Word**

Bit	Description
b15	If set, the value of <b>Pr 01.46</b> is set from b6
b14	If set, the value of <b>Pr 01.45</b> is set from b5
b13	Sets the value of <b>Pr 18.33</b> (Application menu 1, bit 3)
b12	If set, the value of <b>Pr 06.32</b> is set from b3
b11	If set, the value of <b>Pr 06.31</b> is set from b2
b10	If set, the value of <b>Pr 06.30</b> is set from b1
b9	If set, the value of <b>Pr 06.15</b> is set from b0
b8	Sets the value of <b>Pr 18.32</b> (Application menu 1, bit 2)
b7	Sets the value of <b>Pr 18.31</b> (Application menu 1, bit 1)
b6	Sets the value of <b>Pr 01.46</b> (Preset select bit 1)
b5	Sets the value of <b>Pr 01.45</b> (Preset select bit 0)
b4	User trip. Trips drive immediately if set.
b3	Sets the value of <b>Pr 06.32</b> (Sequencing bit 2: Run Reverse)
b2	Sets the value of <b>Pr 06.31</b> (Sequencing bit 1: Jog)
b1	Sets the value of <b>Pr 06.30</b> (Sequencing bit 0: Run forward)
b0	Sets the value of <b>Pr 06.15</b> (Drive enable)

**Table 5.6 Status Word**

Bit	Description
b15	Not used
b14	<b>Pr 10.15</b> (Mains Loss)
b13	<b>Pr 10.14</b> (Direction running)
b12	<b>Pr 10.13</b> (Direction commanded)
b11	<b>Pr 10.12</b> (Braking resistor alarm)
b10	<b>Pr 10.11</b> (Braking IGBT active)
b9	<b>Pr 10.10</b> (Regenerating)
b8	<b>Pr 10.09</b> (Drive output is at current limit)
b7	<b>Pr 10.08</b> (Load reached)
b6	<b>Pr 10.07</b> (Above set speed)
b5	<b>Pr 10.06</b> (At speed)
b4	<b>Pr 10.05</b> (Below set speed)
b3	<b>Pr 10.04</b> (Running at or below min speed)
b2	<b>Pr 10.03</b> (Zero speed)
b1	<b>Pr 10.02</b> (Drive running)
b0	<b>Pr 10.01</b> (Drive healthy)

Pr 90.12	Event task schedule reason ♦ ♣ ♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	On Event

Pr 90.13	Event1 task schedule reason ♦ ♣ ♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	On Event1

Pr 90.14	Event2 task schedule reason ♦ ♣ ♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	On Event2

Pr 90.15	Event3 task schedule reason ♦ ♣ ♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	On Event3

These four parameters give the reason why the particular EVENT task was scheduled. The value only has meaning when the particular EVENT task is running.

The value is bitmapped and defined as follows:

Bits	Description	Value
0-1	Slot triggering the task	0 = Local slot 1 = Slot 1 2 = Slot 2 3 = Slot 3
2-7	Reason for trigger.	0-31 = Other option module initiated 32 = CTNet Sync 33 = Timer Unit 34-63 = User-defined reason via the DPL command SCHEDULEEVENT.

<b>Pr 90.18</b>	<b>Feedback Encoder Freeze Flag</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This parameter needs to be set to zero for the freeze position to be captured. Once the freeze has occurred this parameter is set to 1. To reactivate it simply set it to zero.

<b>Pr 90.19</b>	<b>Feedback Encoder Freeze Position</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

<b>Pr 90.20</b>	<b>Feedback Encoder Freeze Turns</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

These 2 parameters store the position and turns of the feedback encoder at the time the freeze input has been activated.

<b>Pr 90.21</b>	<b>Disable Drive Encoder Position Check</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	Immediate

The drive regularly checks the position derived with the sine and cosine waveforms from a SINCOS encoder via serial communications. Set this parameter to 1 to disable this.

<b>Pr 90.22</b>	<b>Drive Encoder Comms Transmit Register</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

When the Drive Encoder Position Check parameter is disabled (**Pr 90.21**=1) this parameter can be used to communicate with the encoder connected via serial comms with the drive.

<b>Pr 90.23</b>	<b>Drive Encoder Comms Receive Register</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

When the Drive Encoder Position Check parameter is disabled (**Pr 90.21**=1) this parameter can be used to communicate with the encoder connected via serial comms with the drive.

<b>Pr 90.24</b>	<b>Module slot number</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 8-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Initialisation

This parameter reports the slot number into which the module is fitted.

<b>Pr 90.25</b>	<b>Feedback encoder marker position (<math>2^{32}/rev</math>)</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

The top 16-bits are scaled to 65536 counts per revolution regardless of the type of feedback device or scaling configured in the drive.

<b>Pr 90.26</b>	<b>Feedback encoder marker turns (<math>2^{16}/\text{rev}</math>) <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

This parameter gives the feedback encoder marker revolution count.

<b>Pr 90.27</b>	<b>Applications module database version number <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Initialisation

The database version number is read from the database after power-up.

<b>Pr 90.28</b>	<b>Reference Encoder Freeze flag <sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	250 $\mu$ s

This parameter needs to be set to zero for the freeze position to be captured. Once the freeze has occurred this parameter is set to 1. To reactivate it simply set it to zero.

<b>Pr 90.29</b>	<b>Reference Encoder Freeze Position <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250 $\mu$ s

<b>Pr 90.30</b>	<b>Reference Encoder Freeze Turns <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250 $\mu$ s

These 2 parameters store the position and turns respectively of the reference encoder at the time the freeze input was activated.

<b>Pr 90.31</b>	<b>Feedback Encoder Turns and Coarse Position <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

<b>Pr 90.32</b>	<b>Reference Encoder Turns and Coarse Position <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

These 2 parameters store the 16-bit turns in the upper word and 16-bit position in the lower word, of the feedback (**Pr 90.31**) and reference (**Pr 90.32**) encoders.

<b>Pr 90.33</b>	<b>Feedback Encoder Freeze Turns and Coarse Position <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250 $\mu$ s

<b>Pr 90.34</b>	<b>Reference Encoder Freeze Turns and Coarse Position <sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250 $\mu$ s

These 2 parameters store the 16-bit turns in the upper word and the 16-bit position in the lower word, at the time the freeze input was activated.

<b>Pr 90.35</b>	<b>Reference Encoder Marker Position (<math>2^{32}/rev</math>)<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

This parameter stores the reference encoder position at the time the marker pulse was activated.

<b>Pr 90.36</b>	<b>Reference Encoder Marker turns (<math>2^{16}/rev</math>)<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

This parameter stores the reference encoder revolution count at the time the marker pulse was activated.

<b>Pr 90.37</b>	<b>Feedback Encoder Marker Turns and Coarse Position<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

<b>Pr 90.38</b>	<b>Reference Encoder Marker Turns and Coarse Position<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

These 2 parameters store the 16-bit turns in the upper word and 16-bit position in the lower word, of the feedback (**Pr 90.37**) and reference (**Pr 90.38**) encoders at the time the marker pulse was activated.

<b>Pr 90.39</b>	<b>Drive Keypad Button Status<sup>♦♦♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	> 40ms

The state of the Reverse, Run and Stop keys can be read using this parameter. The keys are represented by bits as follows:

Value	Description
b0	Rev
b1	Run
b2	Stop

Button presses from both LCD and LED Keypads will be seen.

<b>Pr 90.40</b>	<b>Event Task Trigger<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Unsigned 16-bit
<b>Default</b>	0	<b>Update Rate</b>	Immediate

Upon setting this parameter to a value it will execute one of the Applications module's Event tasks.

Value	Action
0	Do not trigger Event task
1	Trigger Event task
2	Trigger Event1 task
3	Trigger Event2 task
4	Trigger Event3 task

<b>Pr 90.41</b>	<b>Reference Encoder Marker Flag</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

<b>Pr 90.42</b>	<b>Feedback Encoder Marker Flag</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	See Pr 81.16

These 2 parameters are set to 1 if a relevant encoder marker pulse is activated, but only if the marker flag enable parameters have been set (parameters **Pr 90.45** and **Pr 90.46**). To re-arm the marker these parameters must be set to zero by the user. They cannot be set to 1 by the user.

<b>Pr 90.43</b>	<b>Reference Encoder Source</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Unsigned 8-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

<b>Pr 90.44</b>	<b>Feedback Encoder Source</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Unsigned 8-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

These 2 parameters define the source for the reference and feedback data. See the table below for the valid sources.

Value	Description
0	Drive encoder
1	Slot 1
2	Slot 2
3	Slot 3
4	User program
5	Unconfigured

<b>Pr 90.45</b>	<b>Reference Marker Flag Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

<b>Pr 90.46</b>	<b>Feedback Marker Flag Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

These 2 parameters must be set to 1 to allow the marker flags (parameters **Pr 90.41** and **Pr 90.42**) to be set when the marker pulse is activated.

<b>Pr 90.47</b>	<b>Reference Freeze Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

<b>Pr 90.48</b>	<b>Feedback Freeze Enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	0/1
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

These 2 parameters must be set to 1 allow the freeze flags (parameters **Pr 90.18** and **Pr 90.28**) to be set when the freeze input is activated.

<b>Pr 90.49</b>	<b>APC Runtime Error ID</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	32bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

This parameter shows the error ID of an APC runtime error. This will be set when module trip 81 occurs. Brief descriptions of the error codes are shown in the following table. For more information, refer to the *Advanced Position Control User Guide*.

Value	Description
0	No error or failure to set up encoder object
1	CAM array too small
2	CAM segment overrun
3	CAM size has been specified as zero
4	CAM absolute reset failure

## 5.9 Menu 91 - Fast Access Parameters

The parameters in this menu are Applications module virtual parameters which provide a faster update rate or enhanced resolution than drive parameters.

<b>Pr 91.01</b>	<b>Short-cut enable</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Unsigned 8-bit
<b>Default</b>	0	<b>Update Rate</b>	Immediate

This parameter enables the short-cut parameters detailed later in this section. You must set the appropriate bit in this parameter. See the following table.

Bit	Function	Related Parameter
0	Speed reference shortcut enable	<b>Pr 91.02</b>
1	Hard-speed reference shortcut enable	<b>Pr 91.03</b>
2	Torque reference shortcut enable	<b>Pr 91.04</b>
3	Analogue Output 1	<b>Pr 91.11</b>
4	Analogue Output 2	<b>Pr 91.12</b>
5	Reserved	-
6	Reserved	-
7	Reserved	-

<b>Pr 91.02</b>	<b>Speed set-point (Pr 01.21) <sup>♦</sup><sup>**</sup><sup>♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

Sets the speed reference in units of **0.001RPM**. This value is mirrored in drive parameter **Pr 01.21** (preset speed 1), therefore in order to control the drive speed with this parameter ensure preset speed 1 is selected on the drive (**Pr 01.14=3**, **Pr 01.15=1**).

Ensure bit 0 of **Pr 91.01** is set and the full-scale speed in **Pr 91.05** is set accordingly when using this parameter.

<b>Pr 91.03</b>	<b>Hard-speed reference (Pr 03.22) <sup>♦</sup><sup>**</sup><sup>♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

Controls the hard-speed reference on the drive in units of **0.001RPM**.

Ensure bit 1 of **Pr 91.01** is set and the full-scale speed in **Pr 91.05** is set accordingly when using this parameter. Note that this parameter is valid only in Closed Loop Vector and Servo modes only.

<b>Pr 91.04</b>	<b>Torque setpoint (Pr 04.08) <sup>♦</sup><sup>**</sup><sup>♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

Specifies the torque setpoint (drive parameter **Pr 04.08**) in units of 0.01%.

Ensure bit 2 of **Pr 91.01** is set in order to use this parameter.

<b>Pr 91.05</b>	<b>Full scale speed (RPM) <sup>♦</sup><sup>**</sup><sup>♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	1500	<b>Update Rate</b>	N/A

Set this to the maximum (absolute) speed that will ever be written to with **Pr 91.02** or **Pr 91.03**. This is in units of 1 RPM.

This determines the resolution for the speed values sent to the drive. Attempting to write speed values to **Pr 91.02** or **Pr 91.03** greater than the RPM value specified in **Pr 91.05** will result in the value being limited or a value overrange run-time error.

<b>Pr 91.06</b>	<b>Speed feedback <sup>♦</sup><sup>**</sup><sup>♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This parameter returns the value of the Unidrive SP speed feedback in units of 0.01RPM in closed loop modes. This parameter will only be updated if the speed feedback is derived from the drive's encoder input, not a slot number. This can only happen if **Pr 03.26** on the Unidrive SP is set to a zero. However, if a low resolution encoder is used there may be some jitter at low speed. For example, at 10rpm with a 1024ppr encoder this parameter may jump between 0 and 14.65rpm. This is similar to the Unidrive SP parameter **Pr 03.02**.

<b>Pr 91.07</b>	<b>Current feedback (Pr 04.02) <sup>♦</sup><sup>**</sup><sup>♥</sup></b>		
<b>Access</b>	RO	<b>Range</b>	Signed 16-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This value is taken from parameter **Pr 04.02** and is in 0.01A units (i.e. 150=1.5 Amps)

<b>Pr 91.08</b>	<b>Drive analogue input 1 value</b> ♦ ♦ ♥		
<b>Access</b>	RO	<b>Range</b>	±4000
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This value will be taken from the drive's analogue input 1 and is scaled for ±4000 to represent the +/- full scale signal at the input. Refer to the *Unidrive SP User Guide* for information on the sampling rate of analogue inputs.

<b>Pr 91.09</b>	<b>Drive analogue Input 2 value</b> ♦ ♦ ♥		
<b>Access</b>	RO	<b>Range</b>	±1023
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This value will be taken from the drive's analogue input 2 and is scaled for ±1023 to represent the +/- full scale signal at the input. Refer to the *Unidrive SP User Guide* for information on the sampling rate of analogue inputs.

<b>Pr 91.10</b>	<b>Drive analogue input 3 value</b> ♦ ♦ ♥		
<b>Access</b>	RO	<b>Range</b>	±1023
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This value will be taken from the drive's analogue input 3 and is scaled for ±1023 to represent the +/- full scale signal at the input. Refer to the *Unidrive SP User Guide* for information on the sampling rate of analogue inputs.

<b>Pr 91.11</b>	<b>Drive analogue output 1</b> ♦ ♦ ♥		
<b>Access</b>	RW	<b>Range</b>	±1023
<b>Default</b>	N/A	<b>Update Rate</b>	NA

This parameter sets the value of analogue output 1. Refer to parameter **Pr 91.01** for information on enabling this parameter.

<b>Pr 91.12</b>	<b>Drive analogue output 2</b> ♦ ♦ ♥		
<b>Access</b>	RW	<b>Range</b>	±1023
<b>Default</b>	N/A	<b>Update Rate</b>	NA

This parameter sets the value of analogue output 2. Refer to parameter **Pr 91.01** for information on enabling this parameter.

<b>Pr 91.16</b>	<b>Drive digital inputs</b> ♦ ♦ ♥		
<b>Access</b>	RO	<b>Range</b>	Unsigned 8-bit
<b>Default</b>	N/A	<b>Update Rate</b>	250µs

This parameter is similar to drive parameter **Pr 08.20** in providing the status of 7 digital inputs in one single parameter. Logic polarity and inversions are taken into account.

The bits are assigned as follows:

Bit	Digital Input
0	F1
1	F2
2	F3
3	F4
4	F5

Bit	Digital Input
5	F6
6	Enable
7	Reserved - Read as zero

<b>Pr 91.17</b>	<b>Number of Valid CTSync Messages Received<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	

This parameter will increment every time a good CTSync message is received with a valid checksum.

<b>Pr 91.18</b>	<b>Number of Bad CTSync Messages Received<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	

This parameter will increment every time a CTSync message is received with a bad checksum.

<b>Pr 91.19</b>	<b>Number of Missing CTSync Messages<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	

This parameter will increment every time a message has not been received when the module is expecting one.

<b>Pr 91.20</b>	<b>CTSync Synchronisation Signal Width too Short<sup>♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Synchronisation

This parameter will increment every time the synchronisation signal is the incorrect width. During synchronisation this parameter is likely to increase but should stabilise once synchronisation is complete. After synchronisation, if this parameter is incrementing then there is likely to be excessive noise on the EIA-RS485 network. Check the connections.

<b>Pr 91.21</b>	<b>Inter-option Synchronisation Control<sup>♦♦♦♥</sup></b>		
<b>Access</b>	RW	<b>Range</b>	0 to 2
<b>Default</b>	0	<b>Update Rate</b>	Immediate

This parameter allows the user to set up the Applications module in the Inter-Option module Synchronisation scheme. For more information refer to Chapter 10 *Inter-option Synchronisation* on page 89.

Bit	Description
0	Set this bit for the Applications module to participate in the Inter-Option Module Synchronisation scheme as a Producer. Refer to section 10.1 <i>Overview</i> on page 89 for details of the term Producer.
1	Set this bit for the Applications module to participate in the Inter-Option Module Synchronisation scheme as a Consumer. Refer to section 10.1 <i>Overview</i> on page 89 for details of the term Consumer.

<b>Pr 91.22</b>	<b>Inter-option Synchronisation Status ♦♦♥</b>		
<b>Access</b>	RO	<b>Range</b>	Unsigned 8-bit
<b>Default</b>	N/A	<b>Update Rate</b>	Immediate

This parameter shows the status of the Applications module in the Inter-option module synchronisation scheme. For more information refer to Chapter 10 *Inter-option Synchronisation* on page 89.

Bit	Meaning	Description
0	Requested Inter-Module Synchronisation Role	This is identical to bit 1 of the Inter-Option Module Synchronisation Control Parameter as described above.
1	Requested Inter-Module Synchronisation Role	This is identical to bit 0 of the Inter-Option Module Synchronisation Control Parameter as described above.
2	Inter-Module Synchronisation Role Achieved	This bit indicates that for a module attempting to become inter-module synchronisation Producer it has achieved that role. In the event of more than one module on a given drive attempting to become the synchronisation Producer at least one of these modules will have this bit clear. For a module attempting to become inter-module synchronisation Consumer it indicates that it has not requested to become Producer, and that a Producer has been located in another slot and is being used as the source of synchronisation data. (If a Producer has been located in another slot but the rate of the synchronisation data is not compatible with the Consumer then this bit will be cleared because the Producer - although located - is not being used as the source of synchronisation data.)
3	Synchronisation Producer Output within Specification	This bit is relevant only for modules which have been designated as the inter-option module synchronisation Producer and CTSync Slave. This bit indicates that the signal being provided by the inter-option module synchronisation Producer is within the specified tolerance of the drive, and that the drive is now locked to the CTSync Slave's frequency. Note that this bit may only be set if bits 2 to 0 of this parameter are 1,0,1, indicating that the module is Producer (0,1) and that the Producer role has been achieved (1).
7 to 4	Reserved	Read as zero

## 5.10 Menus 18,19 - Application Parameters

These two menus are designated as application parameters since they are all free to be used for whatever purpose the user wants.

Both menus are identical in their layout. All parameters are read/write access to the Applications module (and via comms), but may be read-only on the drive's keypad.

<b>Pr 1x.01</b>	<b>Integer, read-write saved on power-down ♦♦♥</b>		
<b>Access</b>	RW	<b>Range</b>	Signed 16-bit
<b>Default</b>	0	<b>Update Rate</b>	N/A

This parameter is automatically saved by the drive on power-down.

<b>Pr 1x.02- Pr 1x.10</b>	<b>Integer, read-only</b> ♦♦♥		
<b>Access</b>	RW (RO drive)	<b>Range</b>	Signed 16-bit
<b>Default</b>	0	<b>Update Rate</b>	N/A

<b>Pr 1x.11- Pr 1x.30</b>	<b>Integer, read-write</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Signed 16-bit
<b>Default</b>	0	<b>Update Rate</b>	N/A

<b>Pr 1x.31- Pr 1x.50</b>	<b>Bit, read-write</b> ♦♦♥		
<b>Access</b>	RO	<b>Range</b>	0/1
<b>Default</b>	0	<b>Update Rate</b>	N/A

Parameters **Pr 1x.11-Pr 1x.50** are saveable in the drive's non-volatile memory.

## 5.11 Menu 20 - Application Menu

This menu, like menus 18 and 19, contains parameters that do not affect the operation of the drive and therefore can be used for general purpose.

**NOTE** This menu is NOT saved in the drive's non-volatile memory. Instead it can be stored in the Applications module's flash memory upon request. If more than one Applications module is fitted, only one should be configured to store and restore this menu for obvious reasons.

If parameter **Pr 81.21** is set, this menu will be saved and restored by the Applications module..

<b>Pr 20.01- Pr 20.20</b>	<b>Integer, read-write</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Signed 16-bit
<b>Default</b>	0	<b>Update Rate</b>	N/A

<b>Pr 20.21 Pr 20.40</b>	<b>Long integer, read-write</b> ♦♦♥		
<b>Access</b>	RW	<b>Range</b>	Signed 32-bit
<b>Default</b>	0	<b>Update Rate</b>	N/A

On the drive display/keypad if the value exceeds the maximum that can be displayed (9,999,999), "-----" will be shown. It is not possible to enter values larger than the maximum display value on the keypad.

---

## 6 DPL Programming

---

This chapter covers:

- Basic DPL program structure and syntax
- Basic DPL commands
- New features offered by the Applications module

**NOTE** This is by no means a complete guide to the DPL language. The full reference for all DPL commands and function blocks is provided by the on-line help guides.

### 6.1 Program Header

Every DPL program starts with a header section. SyPTPro creates this section for the user. It basically consists of:

- Program title
- Program author
- Program version number

#### 6.1.1 Aliases

Immediately below the header the user may enter a section of *aliases*. Aliases are used to 'replace' various expressions or constants:

- a numerical constant expression
- the address of a register or parameter
- a DPL expression or statement

Aliases are created with the `$DEFINE` statement.

```
$DEFINE name value
```

For example it is good practice to use aliases to give names to all drive parameters used in a program.

```
$DEFINE PRESET_REF_1 #1. 21  
$DEFINE PRESET_REF_2 #1. 22  
$DEFINE SPEED_FB #3. 02
```

It is also recommended to have the alias name in UPPER-case letters in order to help distinguish them from normal variables.

**NOTE** It is recommended that aliases representing integer values have a '%' symbol appended to the alias name. In graphical programming tools (QLD/FBD), SyPTPro will treat all aliases without a % symbol as floating-point values. Hence they will be rejected on LD or integer only inputs.

The `$DEFINE` directive does NOT produce any code, nor does it speed up the execution time of your program - it simply allows you to refer to something with a different name.

## 6.2 Tasks

A DPL program is separated into separate sections called tasks. Within the tasks a user writes the program instructions that will be executed by the microprocessor under certain conditions or on a particular time-base. Each task has a particular name, purpose and priority and only one of each task can be present in the DPL program. The common tasks are outlined below:

**Table 6.1 Common Tasks**

Task Name	Priority	Purpose
INITIAL	3	The very first task that is run after a power-up or reset. This task is commonly used to initialise drive parameters and program variables. No other tasks can run until this task is completed.
BACKGROUND	1	Low priority task used to do non-time critical functions. This task closely resembles the scan loop of a PLC in the way that it works. It is usual that this task section is created as one big loop, with a command at the end of the task to jump back to the start. If the task is allowed to finish, it will not execute again.
CLOCK	2	Task executed on a fixed timebase (between 1-200ms) used for some time related operations, for example to generate a ramp profile. This task is now synchronised to the drive's level 2 control loop and can be used in place of the old Encoder task.
POS0 POS1	4	Two real-time tasks that run synchronously to a multiple of the drive control loops (range from 250µs to 8ms). These tasks are commonly used to control the drive speed and/or current loop in applications such as positioning. The POS0 task is first to run, followed immediately after by the POS1 task.
EVENT	5	Event tasks only run when a certain event occurs. Events can be raised from various sources such as CTNet, other option modules in the Unidrive SP or the user program. EVENT tasks are the highest priority so usually only have a very small number of instructions. They can be likened to interrupt service routines.
EVENT1	5	See above description.
EVENT2	5	See above description.
EVENT3	5	See above description.
ERROR	1	A task that runs only when a run-time error occurs within the user DPL program (for example a divide by zero). This can be used to safely handle abnormal program behaviour situations. All other tasks will be halted prior to the ERROR task running.

**NOTE** When using the CLOCK, POS0 and POS1 tasks, it is advisable to avoid using code such as the FOR and DO WHILE loops. This may cause a DPL over-run error (tr54).

**NOTE** The UD70 ENCODER and SPEED tasks can still be used. These are now aliases for the POS0 and POS1 tasks respectively (i.e. if the program has an ENCODER task this is the same as if it contained a POS0 task). The timebase for both tasks are not fixed as in the UD70 but specified by the user. The CLOCK task in the Applications module can be used in place of the ENCODER task in the UD70, thus giving a timebase closer to that of the UD70 ENCODER task than is available with the POS0 and POS1 tasks.

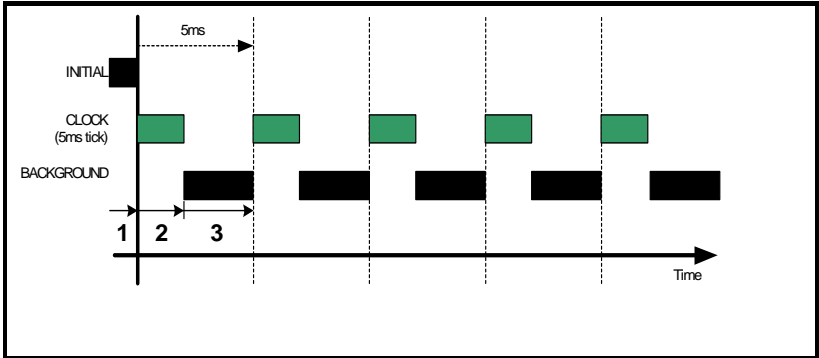
All program instructions **must** live within a task. For time-based tasks like POS0, POS1 and CLOCK the instructions within the task have only a finite time in which to complete, therefore only time critical functions should be performed within them.

The position tasks consist of POS0, APC and POS1 and are executed in this order, if applicable (i.e. if they are set to run).

Safety Information
Introduction
Installation
Getting Started
Parameters
<b>DPL Programming</b>
Communications
Freeze and Marker
CTSync*▼
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

Tasks have different priority levels, therefore it is possible for one task to interrupt another task. In the above table, the higher the priority number the higher the priority is. Therefore a POS0 task can interrupt a CLOCK task which can interrupt the BACKGROUND task.

The following simple diagram illustrates the concept of tasks interrupting each other:

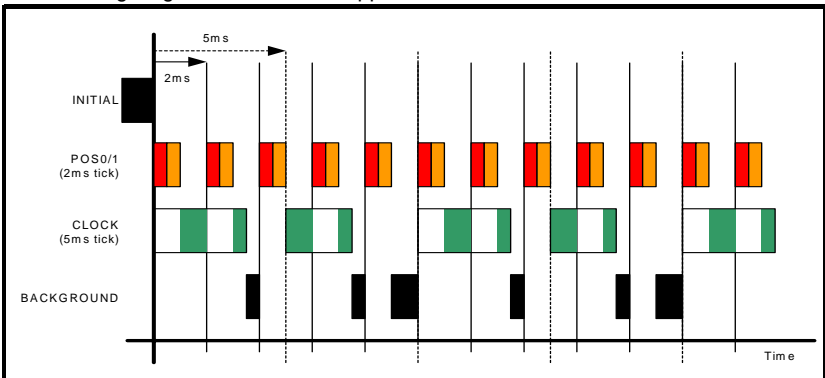


Key:

1. INITIAL task has exclusive control. No other tasks can run
2. CLOCK task which has a higher priority than the BACKGROUND task runs.
3. CLOCK task has finished and now the BACKGROUND task can run - until the next clock tick occurs.

Take particular note that the CLOCK task is run on a fixed timebase (in the diagram above it is 5ms). This means that the instructions within the CLOCK task MUST take less than 5ms to complete otherwise the BACKGROUND task will not get a look-in, or a processor overload trip will occur.

The following diagram shows what happens when the POS tasks are used as well:



This shows the POS0 and POS1 tasks interrupting the CLOCK task which in turn interrupts the BACKGROUND task. As can be seen, this is quite a heavily loaded program since the background task is only executed once in a while. The processor free resource parameter **Pr 81.04** can be used to determine how heavily loaded the Applications module is.

**NOTE**

The SM-Applications Plus module provides parameters **Pr 88.03** thru **Pr 88.08** which will give a greater accuracy on the available resources.

## 6.2.1 EVENT tasks

There are four event tasks provided. The event tasks can be triggered on:

- CTNet SYNC frame received (configured via **Pr 81.35**)
- User program initiated  
New DPL command SCHEDULEEVENT. See on-line help for information.

## 6.3 Variables

### 6.3.1 Types

There are three basic types of variables:

1. Integer Variable
2. Double-precision Floating Point Variable
3. Single-precision Floating Point Variables

An Integer variable is denoted by a % symbol after the variable name. A Floating Point variable is denoted by the lack of a % symbol.

**Table 6.2 Variable Types**

Type	Representation	Range
Integer	32-bit signed.	-2147483648 to 2147483647
Single float	32-bit, 1 sign bit, 8 exponent and 23 mantissa.	±3.40282e+038
Double float	64-bits: 1 sign bit, 52 bit mantissa, 11 bit exponent	±1.79769e+308

Example of variables:

```
Speed% = 1234 // a integer variable  
Value = 55.6 // a floating point variable
```

A special statement is placed at the start of the program to declare what type of floating point variable is used throughout the program - either single or double precision. By default double-precision variables will be used. By including the following line immediately below the program header region (with \$TITLE, etc.) the float type will be single-precision:

```
$float single
```

### 6.3.2 Variable Names

The first character of a variable name must be a letter. Subsequent characters may include letters, numbers and the underscore (\_) character.

#### NOTE

- Variable names are case sensitive (e.g. The variable names speed%, SPEED% and Speed% are different variables).
- SyPTPro QuickLD and FBD editors will only allow the use of variables no longer than 16 characters including any % sign.

### 6.3.3 Initialisation of Variables

All variables must be given an initial value before they can be used. This is typically done within the INITIAL task. For example,

```
Initial {  
Speed_SP% = 0  
Ramp% = 0  
}
```

### 6.3.4 Scope and Lifetime of Variables

Variables can either be global or local. All variables declared in DPL programs are global. i.e. they can be accessed and altered by any task, with the exception of variables within a User Defined Function Block which are local (i.e. cannot be accessed from outside the user defined function block).

No DPL variables survive a reset of the Applications module. Remember that resetting the drive from a tripped condition will also cause a reset (depending on the value in **Pr xx.15**).

### 6.3.5 Fixed-size Variable Arrays

A DPL program may contain arrays of either integer or floating-point variables. Only fixed-size (single-dimension) arrays are allowed.

An array must first be declared using the DIM statement (usually in the Initial task), and the number of elements given in square brackets after the variable name, e.g:

```
DIM MyArray[20] // Integer array having 20 elements
DIM Array2[30] // Floating point array having 30 elements
```

The elements in an array are numbered 0 to number\_of\_elements - 1. So from the above example, the first element of myarray[] is:

myarray[0]

and the last is:

myarray[19]

Two functions are provided that can be used at run-time to determine the upper and lower bounds of an array. These are UPPER and LOWER. for myarray[], UPPER will return 19 and LOWER will return 0.

### 6.3.6 Constant Arrays

Constant arrays, as the name suggests, contain fixed pre-defined values. The values of the constant array are defined within the DPL program by using a special section (see CONST in the on-line help). Only integer values can be defined.

The advantage of constant arrays is that the size of the array is only limited by the amount of available program space - and not variable RAM. The program space is 384kb - it is used to store the compiled DPL file, constant array data, and optionally, the DPL file itself.

### 6.3.7 Storage Space - Number of variables

All variables, fixed-size variable arrays and PLC registers live in an 80kbytes memory space. Each integer variable and single-precision floating point variable consumes 4-bytes (32-bit), and double-precision floating point variables consume 8-bytes (64-bit). There are other things that consume memory as well, such as parameter accesses.

The DPL compiler will notify you if you reach the limit of available memory.

### 6.3.8 Bit addressing of variables

All integer variables and arrays may be bit-addressed. This means that each individual bit within the variable may be separately read or written. After the variable name, place a decimal point (.) followed by the bit number between 0 and 31.

Example 1 (simple variable):

```
Flags% = 0 // initialise all 32 bits to 0
Flags%.0 = 1 // set bit 0 to 1

// now test to see if bit 0 AND bit 1 are set to 1.
IF Flags%.0 & Flags%.1 = 1 THEN
    PRINT "Test satisfied."
ENDIF
```

Example 2 (arrays):

```
DIM MyArray%[10]
...
IF MyArray%.1[4] = 1 THEN; test bit 1 of element #4.
    PRINT "Test satisfied."
ENDIF
```

Note: The bit number must be a constant number - variables are not allowed.

### 6.3.9 PLC Registers

The 'PLC' area is a special range of pre-defined 32-bit registers. The PLC registers are split into 6 sets of 100 parameters numbered 00 to 99. The registers can also be accessed from within a user DPL program by a special variable name or array name. Four of the register sets are also saveable in the Applications module *Flash* memory. See section 5.4 *Menus 70-75 - PLC Registers* on page 37 for further information on PLC registers.

#### 6.3.10 RAM files

RAM files enable the user to store 'files' in the user RAM of the Applications module. These can be uploaded and downloaded using DPL commands. They have an advantage in that you can retrieve or write an array of numbers in one go rather than each element of the array individually.



For further information on RAM files, including example programs, please refer to the on-line help.

## 6.4 Parameters

Parameters are categorised into two sets:

- Drive Parameters
- Applications module Parameters

Drive parameters are ones which reside in the host drive. The majority of them affect the operation of the drive, however a few are set aside as "application parameters". These are menus 18, 19 and 20.

The Applications module parameters are local and accessible only to the Applications module. These parameters provide access to additional features of the Applications module, and give faster access to some drive parameters.

**NOTE** The Applications module always guarantees that the drive parameter database it uses matches that of the host Unidrive SP. When a Applications module is fitted to a Unidrive SP for the first time and powered up the word "Loading" may appear on the drive display for a few seconds. This indicates the Applications module is synchronising databases. This will only occur the first time the module is fitted to the drive. Subsequent power-ups will show "Loading" only for a very short time.

### 6.4.1 Reading and Writing Parameters

Reading and writing parameters is achieved by using the # command. Parameters are accessed in a Pr xx.XX format in the same way as they are accessed on the drive's keypad.

For example, to read the speed feedback parameter (parameter **Pr 03.02**), use:

```
Speed% = #3.02
```

to write to a speed reference parameter (eg. **Pr 01.22**), use:

```
#01.22 = 1500
```

Note that the leading zero in the menu/parameter field is optional. For example #3.02, #03.02, #03.2 and #3.2 will access exactly the same parameter.

### 6.4.2 Fixed-point integer parameters

Dealing with fixed-point integer parameters can be quite a bit slower than for integer parameters. In order to speed this up, a special command #INT can be used when reading and writing parameters. When using this command with fixed-point integer the decimal places will be automatically removed.

For example, parameter **Pr 1.19** has a range of 0.000 - 0.099. Reading the parameter using:

```
Speed_Fine% = #INT1.19
```

will return integer values between 0 and 99. When writing, the command:

```
#INT1.19 = 45
```

will set the parameter to 0.045 (same as #1.19=0.045). The benefit of this is that the DPL program can use integer variables (%) instead of floating-point, thus providing a speed advantage.

## 6.5 Operators

DPL offers all the standard operators as follows:

**Table 6.3 Standard operators in precedence order**

Operator	Meaning
-	Arithmetic negation
!	Boolean negation (unary)
!(..., nbit)	Negation of <i>nbit</i> bits
*	Multiplication
/	Division
%	Modulo (remainder)
+	Addition

**Table 6.3 Standard operators in precedence order**

Operator	Meaning
-	Subtraction
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR (XOR)

**Table 6.4 Conditional operators in precedence order**

Operator	Meaning
=	Equality
<	Less than
>	Greater than
<=	Less then or equals
>=	Greater than or equals
<>	Inequality
AND	Boolean AND
OR	Boolean OR
NOT	Boolean NOT

## 6.6 Basic DPL Commands

The DPL language implemented for the Applications module is backwardly compatible with the UD70 product for Unidrive Classic, but does have a few new extensions.



Refer to the on-line help for full reference on the DPL language and function block library.

### 6.6.1 New Commands for Applications Modules

These are new commands that have been introduced into the DPL language for the Applications module and were not present in the UD70 for Unidrive 1.

#### FOR loop

This is new to the Applications module.

```
FOR variable = integer_expression to integer_expression [STEP constant]
  statements
LOOP
```

#### CASE

This provides an alternative to the IF-ELSEIF-ENDIF construct.

```
SELECT integer_expression
CASE integer_constant
  statements
[CASE integer_constant, integer_constant ...
  [statements]]
[ELSE
  [statements]]
ENDSELECT
```

This construct provides a convenient way to test for multiple constant values. Any number of CASE statements can be included.

**NOTE** In the two examples shown above some sections are within square brackets ([ and ]). This section of code within the square brackets is optional.

**NOTE** The CASE statements operate in the same way as programs like Visual Basic in that the program flow will NOT drop through to the next CASE as it does in the C programming language.

### MAX\_INT, MIN\_INT, MIN\_FLOAT, MAX\_FLOAT

These are special predefined keywords that are recognised by the DPL compiler and replaced by the appropriate numeric value.

**Table 6.5 Min/Max**

Keyword	Value
MIN_INT	-2147483648
MAX_INT	2147483647
MIN_FLOAT	-3.40282e+038 (single precision model) -1.79769e+308 (double precision model)
MAX_FLOAT	3.40282e+038 (single precision model) 1.79769e+308 (double precision model)

### UPPER/LOWER

These functions will take an array as a parameter and will return the upper and lower array index respectively. For example:

```
// Create an array of 1000 elements
DIM Array%(1000)

// now,

l% = LOWER(Array%) // will return the value 0
u% = UPPER(Array%) // will return the value 999.

// get the sum of all values in array%
Total%=0
FOR i% = LOWER(Array%) TO UPPER(Array%)
    Total% = Total% + Array%(i%) //add array element value to total
LOOP
```

### TRUNC

This is used to convert a floating point value to integer, truncating rather than rounding. For example:

```
// Initialise floating point variable
FloatVal = 1.56

Int1% = FloatVal // auto-cast rounds to 2.
Int2% = INT(FloatVal) // explicit cast with INT rounds to 2
Int3% = TRUNC(FloatVal) // explicit cast with TRUNC gives 1
```

## SCHEDULEEVENT

This function block is used to schedule an EVENT task. The arguments are:

- Slot number  
*Specifies which slot to schedule the event task in. Currently only 0 is allowed here which means the local slot.*
- Task ID  
*Range 0-3 to specify which EVENT task to trigger*
- Reason  
*A user defined reason. Must be a value of 34 or higher. This value can be accessed in the EVENT task by looking at parameter **Pr 90.12-Pr 90.15**.*

```
BACKGROUND {
... some code
// Schedule local event1 task with reason code of 45.
a% = SCHEDULEEVENT(0, 1, 45)
... some more code
}

EVENT1 {
IF #90.13 = 45 THEN
    // task scheduled from DPL
ENDIF
}
```

## CTNETDIAGNOSTICS\*♥

This returns diagnostic information for CTNet. Refer to on-line help. This replaces the special variables such as NOFMESSAGES that was used on the UD70 product.

This command takes no input and returns 10 outputs. The table below details the outputs and the equivalent UD70 variable.

**Table 6.6 CTNetDiagnostics Output Values**

Output	Value	UD70 Variable
1	Total number of messages handled by this node	NOFMESSAGES
2	Number of cyclic data overruns	NOFOVERRUNS
3	Lost RX messages	NOFLOSTMESSAGES
4	Number of retries	NOFRETRIES
5	Number of recons	NOFRECONS
6	Number of excessive NAKs	NOFEXENAKS
7	Duplicate sync	NOFDUPSYNCS
8	Number of locally generated recons	NOFMYRECONS
9	Number of non-cyclic messages	NOFNONCYCLICMESSAGES
10	Number of lost routing messages	N/A

## GETPARATTR

This is used to get parameter attributes such as maximum and minimum values, read-only flag, etc.

```
(Max%, Mi n%, Fl ags%) = GETPARATTR(Menu%, Par%)
```

## CModeXfer

This allows the user to change the drive mode without any fitted option modules getting a hard reset. It allows for a smoother drive mode change. While the drive mode change occurs fieldbuses will not be able to write to parameters and this is handled at system level. They will NOT get a 'write to parameter failed' message during this period.

**NOTE** Menus 15 to 20 will not be defaulted when using this command.

**RtuReadRegs<sup>^v</sup>**  
**RtuReadParas<sup>^v</sup>**  
**RtuReadInputRegs<sup>^v</sup>**  
**RtuPresetRegs<sup>^v</sup>**  
**RtuPresetParas<sup>^v</sup>**  
**RtuMasterReply<sup>^v</sup>**  
**RtuMasterStatus<sup>^v</sup>**

These commands are implemented to allow the user to utilise the SM-Applications and SM-Applications Plus modules' Modbus RTU master functionality. Refer to the on-line help for further details.

## PFIXREAD6/PFIXWRITE6

These blocks provide reading and writing of drive parameters in a fixed precision of 6 decimal places.

## SETUSERID

This command is used to set the User ID parameter **Pr 81.49**.

```
SETUSERID(101) // set #81.49 to 101.
```

## ANSIREPLY<sup>^v</sup>

This command existed in the UD70, however the syntax has changed for the Applications module.

```
(Status%, Reply%) = ANSIREPLY()
```

It is a EIA-RS485 port function which is used in conjunction with the ANSIREAD and ANSIWRITE functions.

ANSIREPLY examples:

```
Result% = ANSIREADN(12, 1811) //perform ansi read

//has message been sent successfully
IF Result% = 0 THEN
    //message not sent successfully
    goto top:
ENDIF

Timeout% = 0 //message sent successfully so initialise timeout
DO
    (status%, reply%) = ANSIREPLY() // get status and value of read
    DELAY(1) // delay 100ms
LOOP WHILE Status% = -65536 AND Timeout% < 50 //Timeout=50xclock timebase
```

```

Background{
top:
Value% = #18.11
Result% = ANSIRWITEN(12, 1811, Value%, 1); write value to remote drive
IF Result% = 0 THEN
    //message not sent successfully
    goto top:
ENDIF

CALL get_reply: //get reply

GOTO top:
} //Background

get_reply: {
Timeout% = 0
DO
    (Status%, Reply%) = ANSIREPLY()
LOOP WHILE Status% = -65536 AND Timeout% < 50
} //get_reply:

```

The first output argument returns the status of the ANSIREPLY command and can be one of the following values:

- 65536 = No reply received yet
- 65537 = Reply received, but with bad checksum
- 65538 = EOT received (i.e. parameter does not exist)
- 65539 = NAK received
- 65540 = ACK received

### AssRAM

### UnassRAM

### RamLength

### SetRamLength

These commands allow the programmer to use the RAM files within the Applications module. RAM files provide a means of accessing user program arrays via the CMP file services. For more information on these commands and RAM files refer to the on-line help.

### SMARTCARD functions

New functions are available to utilise the SMARTCARD for reading and storage. There are too many to list here. Refer to the SyPTPro help for more information on using these functions.

## 6.6.2 DPL Commands and Function Blocks

There is a rich list of commands and functions that may be used in a DPL program. This list is too large to go into in this User Guide, so please refer to the on-line help.

Safety Information
Introduction
Installation
Getting Started
Parameters
<b>DPL Programming</b>
Communications
Freeze and Marker
CTSync
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

## 6.7 User Defined Function Blocks

### 6.7.1 Overview

SyPTPro comes as standard with a pre-defined library of function blocks that can be used in the graphical programming tools (LD and FBD) as well as in raw DPL.

The User Defined Function Block system allows the user to create their own function blocks that will automatically become available in the graphical programming tools (Function Block Diagrams and QuickLD diagrams) in addition to the standard library functions.

A UDFB itself is like a self-contained DPL program section in its own right and therefore can consist of a mixture of raw DPL commands, FBD and QLD diagrams and other UDFB's. Note however that you cannot create standard task sections (such as POS0) with a UDFB.

### 6.7.2 Scope of a UDFB

Each UDFB is local to the node's DPL program in which it is defined. To make a UDFB available in other node programs, it is simply a matter of copying and pasting the UDFB section into the other node program.

A UDFB appears within the DPL Editor of SyPTPro in a similar manner to a task - i.e. a collapsible section - and it is recommended practice to place all UDFB's at the top of a program due to the fact that a UDFB must be defined before it is used.

### 6.7.3 Encapsulation and Data Storage

Unlike any task of a DPL program, UDFB's are self-contained units (i.e. encapsulated). This means that each UDFB has its own unique set of variables (local variables).

A UDFB interfaces to the node's DPL program though its input and output arguments alone. It is not possible for a UDFB to access the global DPL variables in the DPL program, or variables in other UDFB's.

A UDFB can of course access drive parameters and Applications module registers that are considered global, however this is to be discouraged especially for blocks that could be re-used in other programs or applications. The only times where a block may need to access parameters or registers directly would be in application / product specific situations.

Each time a UDFB is used in a DPL program, a separate instance is made that is a copy of the UDFB with unique local variables.

Note: The local variables of a UDFB cannot be watched in the SyPTPro Watch Window

### 6.7.4 UDFB Naming

In order to keep UDFBs unique and to avoid any naming collisions between UDFBs and the standard library function blocks, a UDFB name must start with the underscore character (\_). The name is also limited to 16 characters, however it is recommended for the name be kept short so that it displays neatly within the SyPTPro FBD and QuickLD editors, e.g.

`_MyFunc`, `_PI D1` and `_My_Func`

These are examples of illegal names:

`MyFunc`, `UDFB1`

### 6.7.5 Input and Output Arguments

A UDFB can have the following data types passed into and out of it:

- Integer variables
- Floating point variables
- Integer arrays
- Floating point arrays

The input and output arguments are standard DPL variables - i.e. case-sensitive and must start with a letter not a number. The length of input argument names is not limited, however the FBD and QuickLD editors within SyPTPro will only show the first 5 characters of the argument.

The quantity of inputs and outputs is limited only by available memory unlike the UD70 product which was limited to 10 integer inputs and 10 integer outputs.

### 6.7.6 UDFB Code Sections

The code within a UDFB is split into two sections:

- The initial code section
- The body code section

The initial section is used for declaring and initialising any local variables that the UDFB will use. The initial section is run for every instance of a UDFB at start-up or reset (this occurs prior to the DPL Initial task).

**NOTE** The input and output arguments of a UDFB cannot be used in the initial section of the UDFB.

The body section is where the actual code of the function block exists the part that does the work of the function. Input and output arguments only have context within the body section.

The two sections are separated by the keyword FBbody. Initial code goes before this keyword, body code after.

Remember that the actual code can consist of a mixture of DPL, FBD diagrams and QLD diagrams.

Below is an example of a simple UDFB that adds two numbers and scales by a pre-defined amount (0.5):

```
(Output%) = _simplefb(Input1%, Input2%) {
// initialisation code:

Scale% = 500    // initialise a local variable

FBbody
// main body code:

Output% = Input1% + Input2% * Scale% / 1000
}
```

---

# 7 Communications

---

## 7.1 EIA-RS485 Serial Communications Port<sup>♣♥</sup>

The Applications module comes with a EIA-RS485 serial communications port on-board. Refer to section 3 *Installation* on page 11 for information on the hardware connections and wiring.

This port supports a number of built-in protocols: CT-ANSI slave, Modbus RTU in master and slave modes, Modbus ASCII in master and slave modes and 3 user modes. Both two and four wire configurations are possible

If an invalid or unsupported mode is selected, the mode will be defaulted back to 1 (4-wire CT-ANSI) and a run-time error 49 may occur.

The baud-rate is specified in parameter **Pr 81.07**.

The address of this unit is specified in parameter **Pr 81.05**.

### 7.1.1 CT-ANSI

The Applications module supports the ANSIx3.28 protocol.

**NOTE** The CT-ANSI protocol supports reading and writing of full 32-bit values. The UD70 product for Unidrive 1 does not support such large values.

All drive parameters as well as Applications module parameters can be accessed via the EIA-RS485 port on the Applications module.

#### 7.1.1.1 Reading a parameter

The following tables show the message constructs for reading a parameter.

**Table 7.1 Master request**

Character	Description
EOT	End of transmission (Ctrl & D)
A1	Applications module address: 1 <sup>st</sup> digit
A1	Applications module address: 1 <sup>st</sup> digit
A2	Applications module address: 2 <sup>nd</sup> digit
A2	Applications module address: 2 <sup>nd</sup> digit
M1	Menu number: 1 <sup>st</sup> digit
M2	Menu number: 2 <sup>nd</sup> digit
P1	Parameter number: 1 <sup>st</sup> digit
P2	Parameter number: 2 <sup>nd</sup> digit
ENQ	Enquiry (Ctrl & E)

**Table 7.2 Slave response (if request was correct and parameter exists)**

Character	Description
STX	Start of text (Ctrl & B)
M1	Menu number: 1 <sup>st</sup> digit
M2	Menu number: 2 <sup>nd</sup> digit
P1	Parameter number: 1 <sup>st</sup> digit
P2	Parameter number: 2 <sup>nd</sup> digit

**Table 7.2 Slave response (if request was correct and parameter exists)**

Character	Description
D1	Data: 1 <sup>st</sup> digit
D2	Data: 2 <sup>nd</sup> digit
-	
-	
Dn	Data: n <sup>th</sup> digit
ETX	End of text (Ctrl & C)
	Checksum

If the parameter to be read does not exist, the End of transmission character (Ctrl & D) is returned.

The checksum is derived by exclusive ORing the message bytes (characters) together excluding the STX character and the checksum, i.e.  $\text{Checksum} = M1 \wedge M2 \wedge P1 \wedge P2 \wedge D1 \wedge D2 \wedge \dots \wedge Dn \wedge \text{ETX}$ . The checksum is an unsigned 8bit value and if this value is less than 32 then 32 is added to it.

### 7.1.1.2 Writing to a parameter

The following tables show the message constructs for writing to a parameter.

**Table 7.3 Master request**

Character	Description
EOT	End of transmission (Ctrl & D)
A1	Applications module address: 1 <sup>st</sup> digit
A1	Applications module address: 1 <sup>st</sup> digit
A2	Applications module address: 2 <sup>nd</sup> digit
A2	Applications module address: 2 <sup>nd</sup> digit
STX	Start of text (Ctrl & B)
M1	Menu number: 1 <sup>st</sup> digit
M2	Menu number: 2 <sup>nd</sup> digit
P1	Parameter number: 1 <sup>st</sup> digit
P2	Parameter number: 2 <sup>nd</sup> digit
D1	Data: 1st digit
D2	Data: 2nd digit
-	
-	
Dn	Data: nth digit
ETX	End of text (Ctrl & C)
	Checksum

The following rules apply to the data field:

1. The maximum length is 12 characters.
2. The field may contain leading spaces, but not after any other character.
3. A sign character is optional. No sign indicates positive.
4. A decimal point is optional. This can appear at any point in the data field, but not before the sign or before 10 numbers (i.e. the value written should not have more than 9 decimal places). If the decimal point is not in the same position as used by the parameter, some accuracy may be lost or extra decimal places added (e.g. if

+1.2345 is written to a parameter with one decimal place the result is 1.2, if +1.2 is written to a parameter with three decimal places the result is 1.200). It should be noted that parameters can only have 0, 1, 2, 3, 4, 5, or 6 decimal places.

- The data field can contain up to 10 numbers, but the value (even ignoring decimal points) must not exceed the range  $-2^{31}$  to  $2^{31}-1$ .

If the parameter is written successfully an Acknowledge character (Ctrl & F) is returned. If the parameter does not exist, the value written exceeds the range for that parameter or the data field rules are not obeyed, a Not acknowledge character (Ctrl & U) is returned.

The checksum is derived by exclusive ORing the message bytes (characters) together excluding the STX character and the checksum, i.e. Checksum =  $M1 \wedge M2 \wedge P1 \wedge P2 \wedge D1 \wedge D2 \wedge \dots \wedge Dn \wedge ETX$ . The checksum is an unsigned 8bit value and if this value is less than 32 then 32 is added to it.

### 7.1.1.3 Applications module address

The Applications module will only act on messages received where the full address matches the address of the Applications module or the group address in the message (1<sup>st</sup> digit) matches the 1<sup>st</sup> digit of the address of Applications module or the address in the message is a broadcast (0). Broadcast messages are used to write data to multiple nodes.

### 7.1.1.4 Control Characters

**Table 7.4 Summary of control characters**

Character	Description	ASCII code	Ctrl code
STX	Start of text	02	B
ETX	End of text	03	C
EOT	End of transmission	04	D
ENQ	Enquiry	05	E
ACK	Acknowledge	06	F
BS	Backspace	08	H
NAK	Not acknowledge	15	U

### 7.1.2 Modbus-RTU

Both slave and master modes of the Modicon Modbus-RTU protocol are supported.

In RTU slave mode, the following function codes are supported:

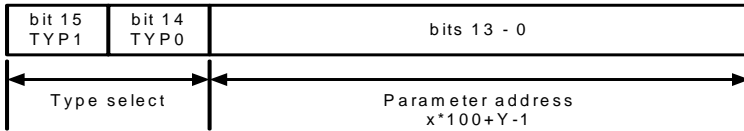
Function	Description
FC3	Read multiple registers
FC6	Preset single registers
FC16	Preset multiple registers
FC23	Read/write multiple registers

The maximum number of registers that can be read/written in one go is 20.

Drive parameters are mapped to Modbus registers as **40000 + Menu×100 + Parameter**.

For example, parameter **Pr 01.21** will be register number 40121.

Parameter data can be accessed in either 16-bit or 32-bit mode. The mode is selected using the upper 2 bits of the register address, as follows:



Type field bits 15-14	Access
00	16-bit. Backwards compatible
01	32-bit.
10	Reserved
11	Reserved

Therefore, to access parameter **Pr 70.01** in 32-bit mode, the register number will be  $40000 + (0 \times 4000 + 70 \times 100 + 01) = 63385$ .

If a 32-bit parameter is read with 16-bit access mode, the least significant 16-bits will be returned.

Note that the actual register number transmitted at the protocol level is one less than the one asked for and does not include the 40000 offset. Most Modbus masters handle this -1 difference automatically, but some don't.

For master mode the following commands are used in the user DPL program:

- RtuReadHoldingRegs
- RtuReadHoldingParas
- RtuReadInputRegs
- RtuPresetMultipleRegs
- RtuPresetMultipleParas
- RtuMasterReply
- RtuMasterStatus

### 7.1.2.5 FC03 Read multiple registers

Read a contiguous array of 16bit registers. The slave imposes an upper limit on the number of registers which can be read. If this is exceeded the slave will issue an exception code 2.

The following tables show the message constructs for Modbus RTU Function Code 03.

**Table 7.5 Master request**

Byte	Description
0	Slave destination node address 1 through 247, 0 is broadcast
1	Function code 0x03
2	Start register address MSB
3	Start register address LSB
4	Number of 16bit registers MSB
5	Number of 16bit registers LSB
6	CRC LSB
7	CRC MSB

**Table 7.6 Slave response**

Byte	Description
0	Slave source node address
1	Function code 0x03
2	Length of register data in read block (in bytes)
3	Register data 0 MSB
4	Register data 0 LSB
3 + byte count	CRC LSB
4 + byte count	CRC MSB

**7.1.2.6 FC06 Preset single register**

Writes a value to a single 16bit register. The normal response is an echo of the request, returned after the register contents have been written. The register address can correspond to a 32bit parameter but only 16bits of data can be sent.

The following tables show the message constructs for Modbus RTU Function Code 06.

**Table 7.7 Master request**

Byte	Description
0	Slave destination node address 1 through 247, 0 is broadcast
1	Function code 0x06
2	Register address MSB
3	Register address LSB
4	Register data MSB
5	Register data LSB
6	CRC LSB
7	CRC MSB

**Table 7.8 Slave response**

Byte	Description
0	Slave source node address
1	Function code 0x06
2	Register address MSB
3	Register address LSB
4	Register data MSB
5	Register data LSB
6	CRC LSB
7	CRC MSB

**7.1.2.7 FC16 Preset multiple registers**

Writes a contiguous array of registers. The slave imposes an upper limit on the number of registers which can be written. If this is exceeded the slave will discard the request and the master will timeout.

The following tables show the message constructs for Modbus RTU Function Code 16.

**Table 7.9 Master request**

Byte	Description
0	Slave destination node address 1 through 247, 0 is broadcast
1	Function code 0x10
2	Start register address MSB
3	Start register address LSB
4	Number of 16bit registers MSB
5	Number of 16bit registers LSB
6	Length of register data to write (in bytes)
7	Register data 0 MSB
8	Register data 0 LSB
7 + byte count	CRC LSB
8 + byte count	CRC MSB

**Table 7.10 Slave response**

Byte	Description
0	Slave source node address
1	Function code 0x10
2	Start register address MSB
3	Start register address LSB
4	Number of 16bit registers written MSB
5	Number of 16bit registers written LSB
6	CRC LSB
7	CRC MSB

### 7.1.2.8 FC23 Read/Presets multiple registers

Writes and reads two continuous arrays of registers. The slave imposes an upper limit on the number of registers which can be written. If this is exceeded the slave will discard the request and the master will timeout.

The following tables show the message constructs for Modbus RTU Function Code 23.

**Table 7.11 Master request**

Byte	Description
0	Slave destination node address 1 through 247, 0 is broadcast
1	Function code 0x17
2	Start register address to read MSB
3	Start register address to read LSB
4	Number of 16bit registers to read MSB
5	Number of 16bit registers to read LSB
6	Start register address to write MSB
7	Start register address to write LSB
8	Number of 16bit registers to write MSB
9	Number of 16bit registers to write LSB
10	Length of register data to write (in bytes)
11	Register data 0 MSB

Safety Information
Introduction
Installation
Getting Started
Parameters
Programming
DPL
Communications
Freeze and Marker
CTSync
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

**Table 7.11 Master request**

Byte	Description
12	Register data 0 LSB
11 + byte count	CRC LSB
12 + byte count	CRC MSB

**Table 7.12 Slave response**

Byte	Description
0	Slave source node address
1	Function code 0x17
2	Length of register data in read block (in bytes)
3	Register data 0 MSB
4	Register data 0 LSB
3 + byte count	CRC LSB
4 + byte count	CRC MSB

### 7.1.3 Modbus ASCII

Both slave and master modes of the Modicon Modbus ASCII protocol are supported. In Modbus ASCII slave mode, the following function codes are supported.:

Function	Description
FC3	Read multiple registers
FC6	Preset single registers
FC16	Preset multiple registers
FC23	Read/write multiple registers



Refer to on-line help for further information.

Full details of the Modbus protocol can be found on the Modicon web site at [www.modicon.com](http://www.modicon.com). Note that the 32-bit access mode is specific to Control Techniques.

### 7.1.4 User Modes

These modes turn off all internal protocols and allow the user to access the EIA-RS485 port directly from the DPL program. They can be used in conjunction with the DPL ANSI commands - ANSIREAD, ANSIWRITE etc. User defined protocols can also be implemented using the DPL PUTCHAR and GETCHAR commands.

## 7.2 CNet\*♥

Full details of CNet are outside the scope of this User Guide and can be found in the separate *CNet User Guide*.

## 7.3 Applications module mapping parameters (fieldbus)

The Applications module has internal parameters that can be written to or read from by fieldbus options also fitted to the Unidrive SP. This can provide a convenient way to communicate between 2 fieldbuses. These parameters are shown in the table below.

**Table 7.13 Applications Module Internal Parameters**

Applications Module Parameters	Parameter Reference	Direct to Slot 1	Direct to Slot 2	Direct to Slot 3
_Pxx% PLC Registers	Pr 70.XX	Pr 100.XX	Pr 130.XX	Pr 160.XX
_Qxx% PLC Registers	Pr 71.XX	Pr 101.XX	Pr 131.XX	Pr 161.XX
_Rxx% PLC Registers	Pr 72.XX	Pr 102.XX	Pr 132.XX	Pr 162.XX
_Sxx% PLC Registers	Pr 73.XX	Pr 103.XX	Pr 133.XX	Pr 163.XX
_Txx% PLC Registers	Pr 74.XX	Pr 104.XX	Pr 134.XX	Pr 164.XX
_Uxx% PLC Registers	Pr 75.XX	Pr 105.XX	Pr 135.XX	Pr 165.XX
Local Configuration Parameters	Pr 81.XX	Pr 111.XX	Pr 141.XX	Pr 171.XX
Timer Function Parameters**	Pr 85.XX	Pr 115.XX	Pr 145.XX	Pr 175.XX
Digital I/O Parameters**	Pr 86.XX	Pr 116.XX	Pr 146.XX	Pr 176.XX
Status Parameters	Pr 88.XX	Pr 118.XX	Pr 148.XX	Pr 178.XX
General Parameters	Pr 90.XX	Pr 120.XX	Pr 150.XX	Pr 180.XX
Fast Access Parameters	Pr 91.XX	Pr 121.XX	Pr 151.XX	Pr 181.XX

The fieldbus interface module reads and writes data directly to and from the internal registers in an Applications module. The fieldbus interface module can read data from and write data to an Applications module fitted in any slot in the Unidrive SP, simply by specifying the target parameter as shown in Table 7.13.

**NOTE** If a single Applications module is fitted to the Unidrive SP, normal Applications module parameter references can be used without problem, as the SM-PROFIBUS-DP will automatically divert them to the Applications module.

### 7.3.1 Example configuration 1

Consider a Unidrive SP with the following configuration:

- Slot 1 - Vacant
- Slot 2 - Applications module
- Slot 3 - SM-PROFIBUS-DP module

If a parameter read request comes over the PROFIBUS-DP network to read **Pr 71.08**, this will be re-directed to the Applications module in the lowest slot number, i.e. slot 2. The value in \_Q08% from slot 2 will be returned.

If a parameter read request comes over the PROFIBUS-DP network to read **Pr 131.08**, this will be sent straight to the Applications module in slot 2. The value in \_Q08% from slot 2 will be returned.

If a parameter read request comes over the PROFIBUS-DP network to read **Pr 101.08**, this will be sent straight to the Applications module in slot 1. As there is no Applications module fitted in slot 1, an error message will be returned, indicating that the parameter does not exist.

If, for example, an SM-DeviceNet module is fitted to slot 1, you could use the direct slot parameter reference to read or write data, giving a simple communications gateway between DeviceNet and Profibus-DP.

### 7.3.2 Example configuration 2

Consider a Unidrive SP with the following configuration:

- Slot 1 - Applications module
- Slot 2 - Applications module
- Slot 3 - SM-PROFIBUS-DP module

If a parameter read request comes over the PROFIBUS-DP network to read **Pr 71.08**, this will be re-directed to the Applications module in the lowest slot number, i.e. slot 1. The value in `_Q08%` from slot 1 will be returned.

If a parameter read request comes over the PROFIBUS-DP network to read **Pr 131.08**, this will be sent straight to the Applications module in slot 2. The value in `_Q08%` from slot 2 will be returned.

If a parameter read request comes over the PROFIBUS-DP network to read **Pr 101.08**, this will be sent straight to the Applications module in slot 1. The value in `_Q08%` from slot 1 will be returned.

**NOTE**

If more than one Applications module is fitted to the Unidrive SP, it is best to access the module parameters using the direct slot parameter references. If normal Applications module parameters are used, and the Applications module is removed from slot 1, these parameter references will be re-directed to slot 2 instead.

### 7.3.3 Example configuration 3

Consider a Unidrive SP with the following configuration:

- Slot 1 - Applications module
- Slot 2 - Applications module
- Slot 3 - Applications module

The Applications module does not allow reading from or writing to Direct slot parameter references, so data transfer must be done another way.

If the Applications module in slot 1 wanted to read register **Pr 71.08** from the Applications module in slot 3, this must be done through one of the 32bit application parameters (**Pr 20.21-Pr 20.40**). The Applications module in slot 1 must write the register data to the application parameter. The Applications module in slot 3 can then read that parameter.

Code in slot 1 :

```
#20. 21 = #71. 08
```

Code in slot 3 :

```
#71. 08 = #20. 21
```

This data can also be accessed by the Applications module in slot 2.

**NOTE**

The Applications module cannot access the **Pr 1xx.XX** internal menus. It must access them directly by the parameter reference number. e.g. if you wanted to read parameter **Pr 104.35** you would need to use **Pr 74.35**. Therefore, it is not possible to access these parameters/registers via inter-option communications.

\*\*The example code above will not synchronise the updating of parameter **Pr 71.08** in the 2 modules. For more information refer to section 10 *Inter-option Synchronisation* on page 89.

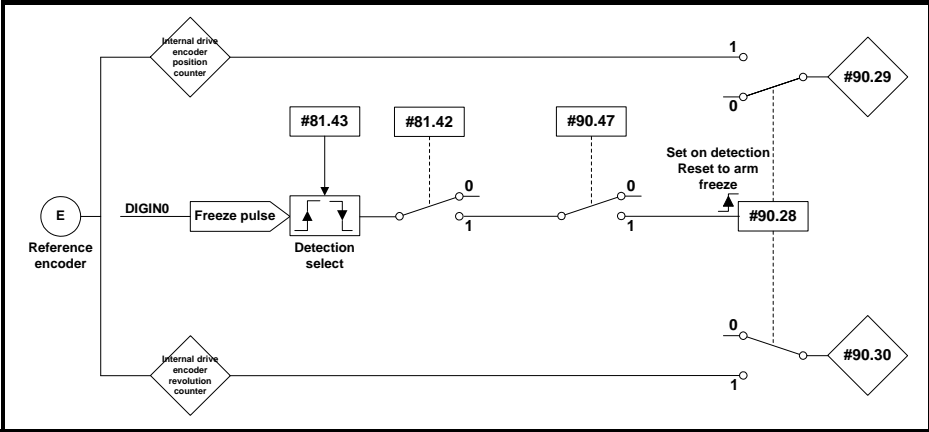
If the Applications\*\* modules are connected on the same CTNet network then this can be done through cyclic data. For more information on CTNet refer to the *CTNet User Guide*.

# 8 Freeze and Marker

## 8.1 Freeze input<sup>†</sup>

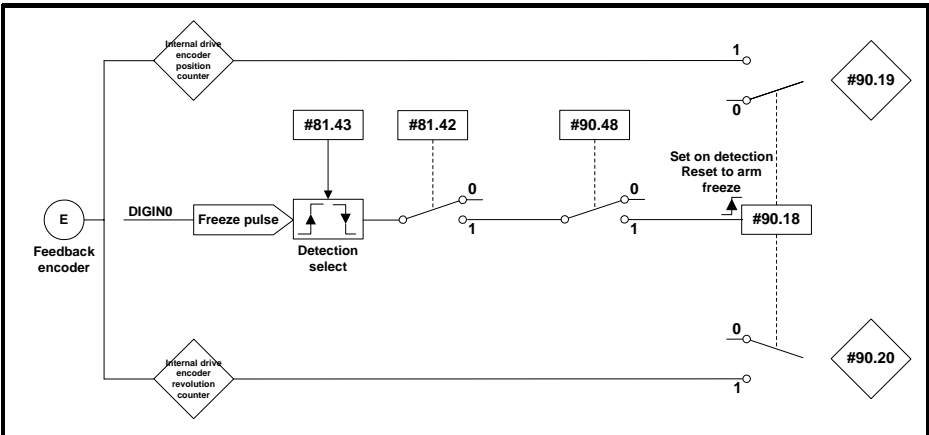
Digital Input 0 (DIGIN0) can be used to 'freeze' the reference and feedback encoder counters.

**Figure 8-1 SM-Applications/SM-Applications Plus Reference Freeze Input**



The encoder revolution counter is cached into parameter **Pr 90.30** and the encoder position is cached into **Pr 90.29**.

**Figure 8-2 SM-Applications/SM-Applications Plus Feedback Freeze Input**



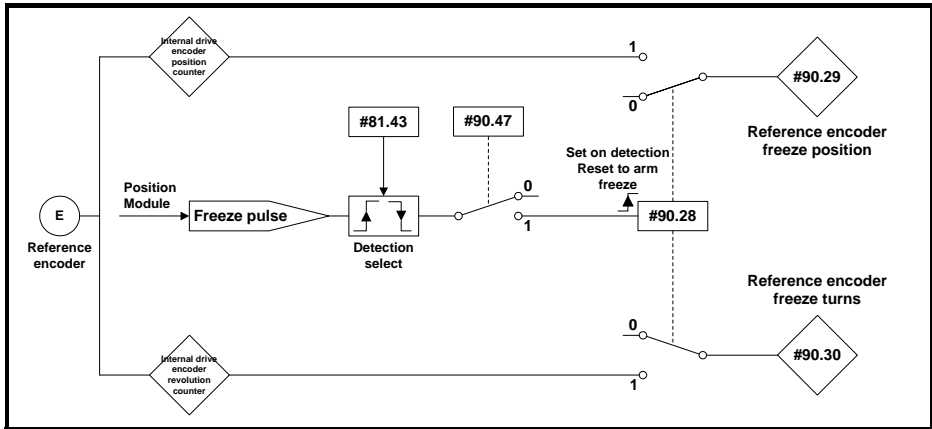
The encoder revolution counter is cached into parameter **Pr 90.20** and the encoder position is cached into **Pr 90.19**.

**NOTE** When freezing the SM-Universal Encoder Plus data, the Applications module's freeze input cannot be used. The freeze should come from the SM-Universal Encoder Plus module. Refer to the *SM-Universal Encoder Plus User Guide* for more information on using its freeze with the Applications modules.

## 8.2 Freeze input<sup>♦</sup>

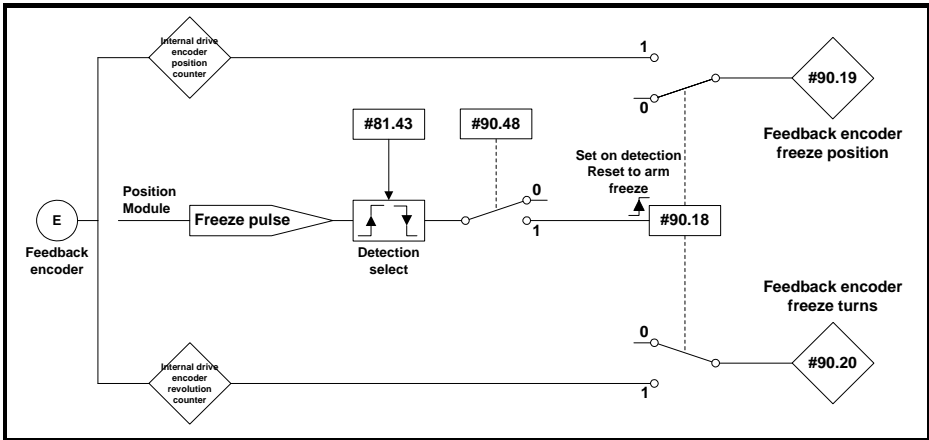
The SM-Applications Lite is unlike the SM-Applications and SM-Applications Plus modules in that it does not have digital I/O which can be used to 'freeze' the reference and feedback encoder counters. However, another module in the same drive can be used to freeze the encoder data in the Applications Lite. For instance, setting parameter x.40 for an SM-Universal Encoder Plus module to 1 allows other slots in the same drive to freeze data.

**Figure 8-3 SM-Applications Lite Reference Freeze Input**



The encoder revolution counter is cached into parameter **Pr 90.29** and the encoder position is cached into **Pr 90.30**.

Figure 8-4 SM-Applications Lite Feedback Freeze Input



The encoder revolution counter is cached into parameter **Pr 90.19** and the encoder position is cached into **Pr 90.20**.

The freeze position of both the reference and feedback encoders can be captured on the rising or the falling edge of the freeze pulse. This is determined by setting parameter **Pr 81.43** to either 0 (rising edge) or 1 (falling edge). Parameter **Pr 81.42** enables the position to be written to parameters **Pr 90.19** and **Pr 90.29** and the revolution counters to be written to parameters **Pr 90.20** and **Pr 90.30**.

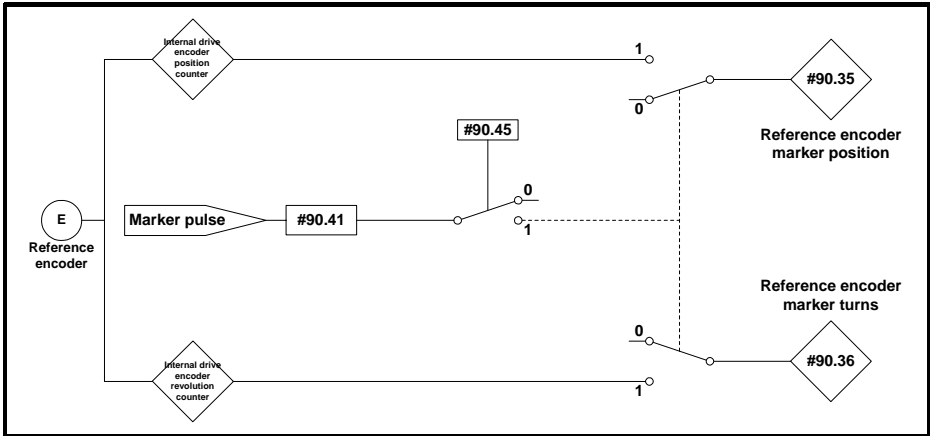
When a freeze input is seen parameter **Pr 90.18** and **Pr 90.28** are set to 1 automatically so that the position can be written to **Pr 90.19** and **Pr 90.29** and the revolution counters can be written to **Pr 90.20** and **Pr 90.30**. Parameters **Pr 90.18** and **Pr 90.28** must be reset to zero if the user wants to update the data again on the next freeze pulse.

Safety Information
Introduction
Installation
Getting Started
Parameters
Programming
DPL
Communications
Freeze and Marker
CTSync*
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

### 8.3 Marker pulse ♦♣♥

The Applications module is able to cache the position and revolution count at the point when a Marker pulse is seen on the reference or feedback encoders.

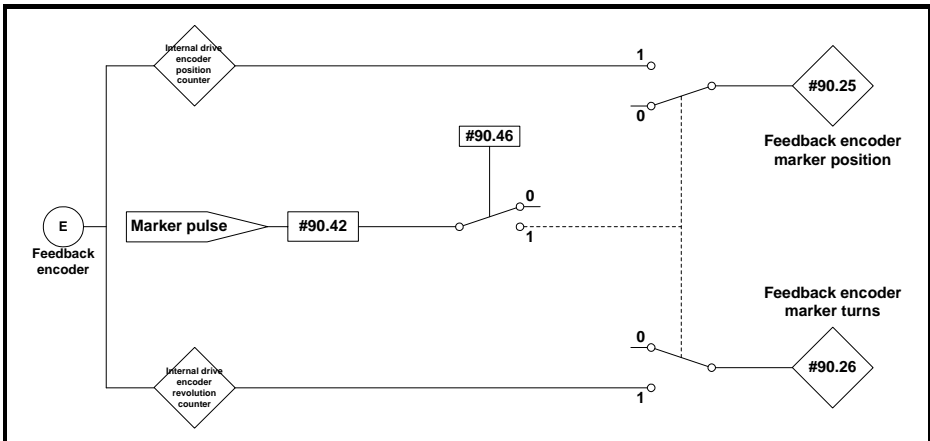
**Figure 8-5 Applications Module Reference Marker**



The marker position is cached into parameter **Pr 90.35** and the marker revolution counter is cached into parameter **Pr 90.36**.

The user sets **Pr 90.41** to a zero and the drive sets **Pr 90.41** to a 1 EVERY time a marker is detected. The marker data must be consumed before the next marker pulse.

**Figure 8-6 Applications Module Feedback Marker**



The marker position is cached into parameter **Pr 90.25** and the marker revolution counter is cached into parameter **Pr 90.26**.

The user sets **Pr 90.42** to a zero and the drive sets **Pr 90.42** to a 1 EVERY time a marker is detected. The marker data must be consumed before the next marker pulse.

## 9.1 Overview

The Applications module may be used to synchronise two or more Unidrive SP drives. This will ensure that the drives run their internal functions at exactly the same frequency and time meaning all actions are performed at the same instant.

Also, 3 data values can be passed from one module (the Master) to others (Slaves) on the CTSync network. This comprises 2 x signed 32bit integers and 1 x Unsigned 8bit integer.

Only one Applications module should be configured as the Master and all others configured as Slaves if they need to participate in the CTSync scheme. The Master generates reference data which is transmitted to all Slaves on the network. The Master can be set to operate as a Slave, if for instance two drives need to be synchronised. In this case the Master will be generating the reference data as well as following that reference data. The slave will also be following that reference data.

## 9.2 Connections

CTSync operates via a connection between the EIA-RS485 ports of the Applications modules on the network in either 2-wire or 4-wire. Refer to section 3.7 *EIA-RS485 Connections\*♥* on page 15 for information on how to connect the Applications' RS-485 ports.

To simplify wiring the Slave transmit and Master receive signal line connections can be omitted in 4-wire mode. This is because the Master does not receive a response from the Slave.

## 9.3 Limitations

- Only one CTSync Master is permitted on the Network
- All CTSync Master and Slave Pos tasks must be set to the same update time (parameter **Pr 81.12**)
- 8 nodes maximum for 2-wire and 16 nodes maximum for 4-wire before line repeaters are required.
- Maximum cable length of EIA-RS485 network is 1200 metres.

## 9.4 CTSync Function Blocks

Six Function Blocks are available when using the CTSync functionality of the Applications module. These are listed below.

### 9.4.1 CTSYNCSetMasterReferences

```
CTSYNCSetMasterReferences(Reference1%, Reference2%, AuxiliaryReference%)
```

Input Arguments	
Argument name	Range
Reference1	Signed 32-bit
Reference2	Signed 32-bit
AuxiliaryReference	Unsigned 8-bit

This function block allows the CTSync Master to write reference data to all CTSync Slaves on the network. This command will have no effect if used in a CTSync Slave.

## 9.4.2 CTSYNCGetSlaveReferences

```
(Reference1%, Reference2%, AuxiliaryReference%, Status%) = CTSYNCGetSlaveReferences()
```

Output Arguments	
Argument name	Range
Reference1%	Signed 32-bit
Reference2%	Signed 32-bit
AuxiliaryReference%	Unsigned 8-bit

This function block allows the CTSync to read the reference data created by the CTSync Master. It can be used on both the Master and Slaves.

The block returns the values of the last Master references received.

### Output Arguments:

Status%: 1: OK  
0: Zero or too few data received.  
-1: More bytes than expected received.  
-2: Checksum error in received data.  
-3: Not in CTSync mode.

If Status does not contain OK then Reference1%, Reference2% and AuxiliaryReference% are not modified.

## 9.4.3 CTSYNCSetupOutputChannel

```
Result% = CTSYNCSetupOutputChannel (Channel%, Menu%, Parameter%)
```

This function block configures one of the 3 channels to route any data passed to it to a specified drive parameter.

### Input Arguments:

Channel%: 1,2 or 3 for the 3 available "channels".  
Menu%: Drive menu number to write to.  
Parameter%: Drive parameter number to write to.

### Output Arguments:

Result%: 1: Operation succeeded.  
-1: Invalid channel specified.  
-3: Channel configuration being run by another task.  
-4: Parameter does not exist or is read-only

## 9.4.4 CTSYNCEnableOutputChannel

```
Result% = CTSYNCEnableOutputChannel (Channel%)
```

This function block allows the specified channel to write it's data to the Unidrive SP at the start of each motion engine sample period.

### Input Arguments:

Channel%: 1,2 or 3 for the 3 available "channels".

### Output Arguments:

Result%: 1: Operation succeeded.  
0: Channel is not set up correctly.  
-1: Invalid channel specified.  
-3: Channel configuration being run by another task.

## 9.4.5 CTSYNCDisableOutputChannel

Result % = CTSYNCDisableOutputChannel (Channel %)

This function block causes the specified channel to cease writing it's data to the Unidrive SP.

### Input Arguments:

Channel%: 1,2 or 3 for the 3 available "channels".

### Output Arguments:

Result%: 1: Operation succeeded.  
 -1: Invalid channel specified.  
 -3: Channel configuration being run by another task.

## 9.4.6 CTSYNCWriteOutputChannel

Result % = CTSYNCWriteOutputChannel (Channel %, value%)

This function block writes a value to the specified channel.

### Input Arguments:

Channel%: 1,2 or 3 for the 3 available "channels".  
 Value%: Value to write.

### Output Arguments:

Result%: 1: Operation succeeded.  
 0: Channel is not set up correctly.  
 -1: Invalid channel specified.  
 -3: Channel configuration being run by another task.

If the value to be written is over-range for the parameter, the drive will be tripped (tr44) if **Pr 81.14=1** and **Pr 81.17=1**, or the value will be clamped if either of these is set to zero.

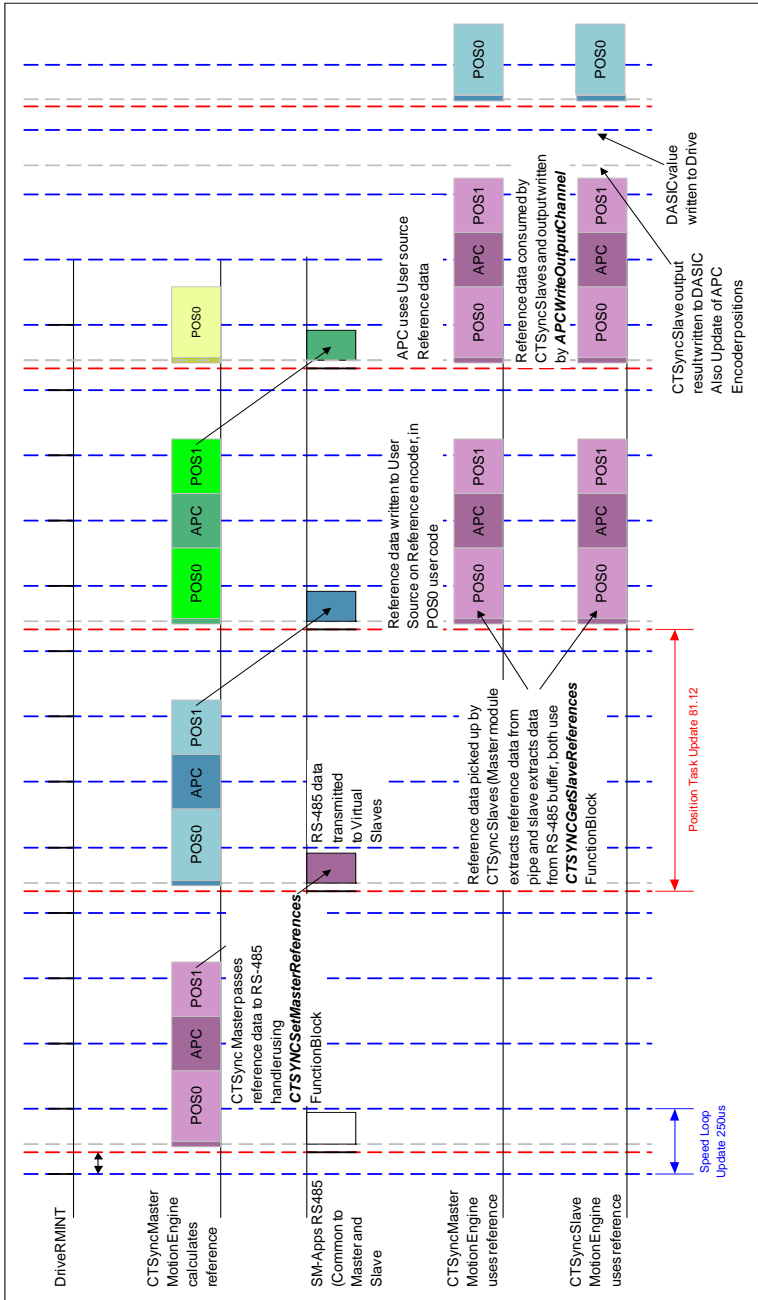
## 9.5 Motion Engine

The motion engine tasks for each motion engine sample are shown below:

1. The Master motion engine calculates the reference.
2. The Master passes the reference data to the RS-485 handler by using the **CTSYNCSetMasterReferences** function block.
3. The RS-485 data is transmitted to the Slave(s).
4. The reference data is retrieved by the Slave(s) using the **CTSYNCGetSlaveReferences** function block.
5. The reference data is output by the **CTSYNCWriteOutputChannel** function block (if required) to the channels specified.
6. The Slave output values are written to the Drive parameters via the Drive's ASIC.

For more information on the timings refer to Figure 9-1 *Motion Engine timing*

Figure 9-1 Motion Engine timing



## 9.6 Virtual Master Example

### 9.6.1 Example Master Code

The following code shows how the master would be set up to generate reference data and send the data to the slaves. It also shows that the master acts as a slave by using the reference data it has generated. This is done with the **CTSYNCGetSlaveReferences** function block.

The example generates Virtual Master data as a triangular ramp and also implements a Virtual Slave which outputs the data to the drive.

```
Initial
{
    #81.06=25 //CTSync Master
    #81.12=2 //Pos task at 500us
    REINIT

    //Set up Virtual Master Ramp Reference.
    Ramp% = 0

    //Configure Slave output channel 1 to menu 18, parameter 11.
    CTSYNCSetupOutputChannel (1, 18, 11)

    //Enable the configured output channel.
    CTSYNCEnableOutputChannel (1)
}

POSO
{
    //Perform Slave function first, so timing is identical between master and slave:
    (SlaveRef1%, SlaveRef2%, Auxref%, Status%) = CTSYNCGetSlaveReferences()

    //Status% should be checked here to ensure the data was received correctly.
    if Status% = USR_VALUE_OK then

        //Output to the drive at the start of next motion period.
        CTSYNCWriteOutputChannel (1, SlaveRef1%)

    endif

    //Perform Master function, in this case a ramp:
    Ramp% = Ramp% + 1

    //Apply reset to ramp profile if required
    if Ramp% > 999 then

        //Wrap back to zero
        Ramp% = 0
    endif

    //Pass Reference Data to Slaves. Note only reference1 used.
    CTSYNCSetMasterReferences(Ramp%, 0, 0)
}
```

## 9.6.2 Example Slave code

The following code shows how the slave(s) would read the reference data generated by the Virtual Master.

```
Initial
{
    #81.06=26 //CTSync Slave
    #81.12=2 //Pos task at 500us
    REINIT

    //Set up Virtual Master Ramp Reference.
    Ramp% = 0

    //Configure Slave output channel 1 to menu 18, parameter 11.
    CTSYNCSetupOutputChannel (1, 18, 11)

    //Enable the configured output channel.
    CTSYNCEnableOutputChannel (1)
}

POSO
{
    //Perform Slave function first, so timing is identical between master and slave:
    (Slaveref1%, Slaveref2%, Auxref%, Status%) = CTSYNCGetSlaveReferences()

    // Check the data was received in good order.
    if Status% = USR_VALUE_OK then

        //Output to the drive at the start of next motion period.
        CTSYNCWriteOutputChannel (1, Slaveref1%)

    endif
}
```

# 10 Inter-option Synchronisation

## 10.1 Overview

The Inter-option Module Synchronisation scheme provides a mechanism to allow position control tasks on one or more modules to be synchronised to position control tasks on another module on the same Unidrive SP without the need for external wiring. Synchronisation is achieved by one module producing a trigger signal and one or more modules consuming the trigger signal. These modules are referred to as Producers and Consumers. The trigger signal is used to alter the scheduling of the POS engine.

If the module is a Producer the trigger may be provided at a rate of 250µs, 500µs, 1ms, 2ms, 4ms or 8ms i.e. Producer modules generate the trigger every POS period, as specified in parameter *Pr 81.12*.

♦♥This scheme is in addition to the CTSync scheme (refer to section 9 *CTSync*♦♥ on page 83 for more information), although it can be used in conjunction with CTSync to synchronise modules in more than one Unidrive SP (see section 10.3 *Inter-option Synchronisation example 2*♦♥ on page 93).

The module, when a Consumer, can run its Position Control tasks at the same rate or quicker than the Synchronisation Signal (trigger) from the Producer.

If a module is a Consumer and its position control task is set to run slower than that of the Synchronisation Signal (trigger) from the Producer, then it will be impossible for it to know the phase of the Producer, so it will not attempt to synchronise to the Producer's signal (see section 10.3 *Inter-option Synchronisation example 2*♦♥ on page 93).

## 10.2 Inter-Option Synchronisation example 1

This example shows a scenario where the POS0 tasks of the modules in a Unidrive SP need to be synchronised.

The following table shows the settings for each of the 3 modules:

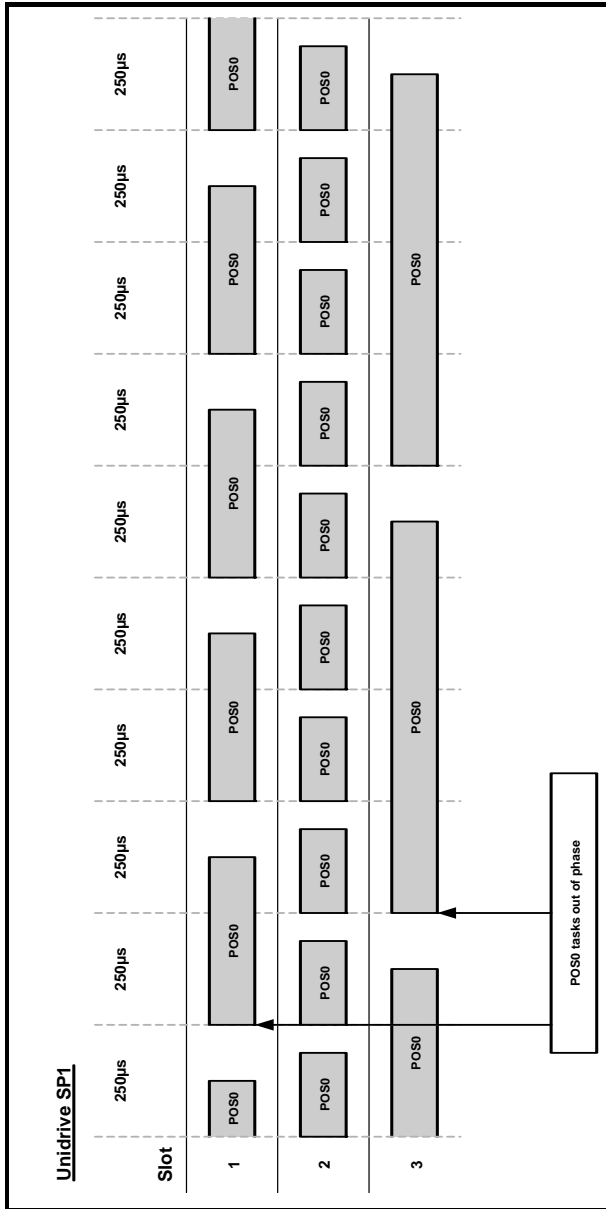
**Table 10.1 Parameter Settings**

	Parameter	Value	Description
Slot 1	Pr 81.06	1 (default)	Not in CTSync scheme.♦♥
	Pr 81.12	2	500µs position control task period
	Pr 91.21	2	Inter-Option Sync Consumer
	Pr 91.22	6	<i>Inter-Option Sync Consumer status achieved</i>
Slot 2	Pr 81.06	1 (default)	Not in CTSync scheme.♦♥
	Pr 81.12	1	250µs position control task period
	Pr 91.21	2	Inter-Option Sync Consumer
	Pr 91.22	6	<i>Inter-Option Sync Consumer status achieved</i>
Slot 3	Pr 81.06	1 (default)	Not in CTSync scheme.♦♥
	Pr 81.12	3	1ms position control task period
	Pr 91.21	1	Inter-Option Sync Producer
	Pr 91.22	5	<i>Inter-Option Sync Producer status achieved</i>

The italicised parameter **Pr 91.22** is a status parameter and does not require setting. It shows whether the module has achieved the Inter-Option functionality specified in parameter **Pr 91.21**. For more information refer to parameter *Pr 91.22* .

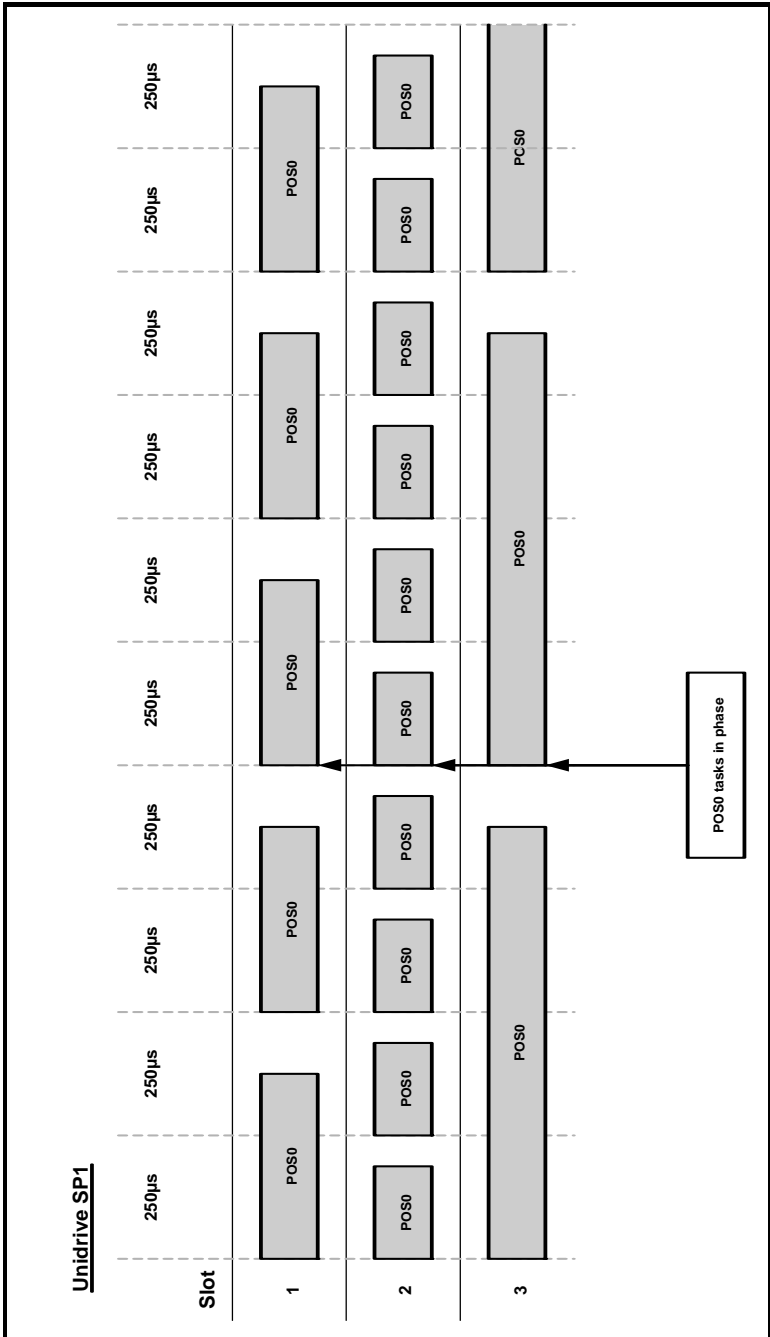
Before setting the parameters above, the POS tasks may be executing out of phase with each other as shown in Figure 10-1 *POS Task Execution BEFORE Inter-option Synchronisation* . After setting the parameters to the values shown above the POS tasks will be executing in phase with each other as shown in Figure 10-2 *POS Task Execution AFTER Inter-option Synchronisation* .

Figure 10-1 POS Task Execution BEFORE Inter-option Synchronisation



Safety Information	Introduction	Installation	Getting Started	Parameters Programming	DPL	Communications	Freeze and Marker	CTSync**	<b>Inter-option Synchronisation</b>	Diagnostics	Migration Guide	Quick Reference	Index
--------------------	--------------	--------------	-----------------	------------------------	-----	----------------	-------------------	----------	-------------------------------------	-------------	-----------------	-----------------	-------

Figure 10-2 POS Task Execution AFTER Inter-option Synchronisation



## 10.3 Inter-option Synchronisation example 2<sup>★♥</sup>

This example shows a scenario where the POS0 tasks of the modules in 2 Unidrive SPs need to be synchronised.

For this to work a module on each drive needs to be configured to participate in the CTSync scheme (refer to section 9 *CTSync*<sup>★♥</sup> on page 83 for more information). One module must be configured as the CTSync Master and one module configured as the CTSync Slave.

The following tables show the settings for each of the 6 modules:

**Table 10.2 Parameter Settings for Unidrive SP 1**

	Parameter	Value	Description
Slot 1	Pr 81.06	1 (default)	Not in CTSync scheme
	Pr 81.12	2	500µs position control task period
	Pr 91.21	2	Inter-Option Sync Consumer
	<i>Pr 91.22</i>	6	<i>Inter-Option Sync Consumer status achieved</i>
Slot 2	Pr 81.06	1 (default)	Not in CTSync
	Pr 81.12	1	250µs position control task period
	Pr 91.21	2	Inter-Option Sync Consumer
	<i>Pr 91.22</i>	6	<i>Inter-Option Sync Consumer status achieved</i>
Slot 3	Pr 81.06	25 (default)	CTSync Master
	Pr 81.12	3	1ms position control task period
	Pr 91.21	1	Inter-Option Sync Producer
	<i>Pr 91.22</i>	5	<i>Inter-Option Sync Producer status achieved</i>

**Table 10.3 Parameter Settings for Unidrive SP 2**

	Parameter	Value	Description
Slot 1	Pr 81.06	1 (default)	Not in CTSync scheme
	Pr 81.12	2	500µs position control task period
	Pr 91.21	2	Inter-Option Sync Consumer
	<i>Pr 91.22</i>	6	<i>Inter-Option Sync Consumer status achieved</i>
Slot 2	Pr 81.06	26	CTSync Slave
	Pr 81.12	3	1ms position control task period
	Pr 91.21	1	Inter-Option Sync Producer
	<i>Pr 91.22</i>	13	<i>Inter-Option Sync Producer status achieved with DINT signal</i>
Slot 3	Pr 81.06	1 (default)	Not in CTSync scheme
	Pr 81.12	4	2ms position control task period
	Pr 91.21	2	Inter-Option Sync Consumer
	<i>Pr 91.22</i>	2	<i>Inter-Option Sync Consumer status not achieved (POS task too slow)</i>

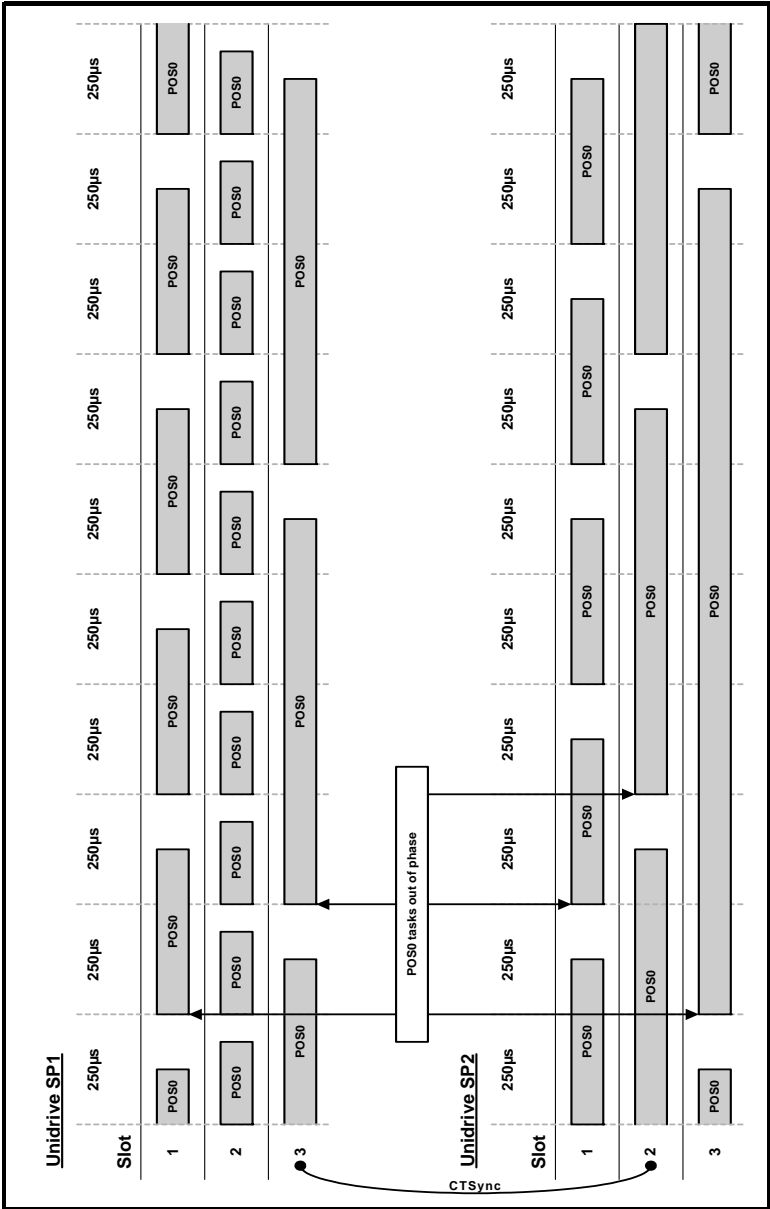
The italicised parameter Pr 91.22 is a status parameter and does not require setting. It shows whether the module has achieved the Inter-Option functionality specified in parameter Pr 91.21. For more information refer to parameter *Pr 91.22* .

**NOTE**

When a module is a CTSync Slave, it has to be the Inter-Option Synchronisation Producer on the drive it is fitted to therefore parameter Pr 91.21 will automatically be set to a value of 1.

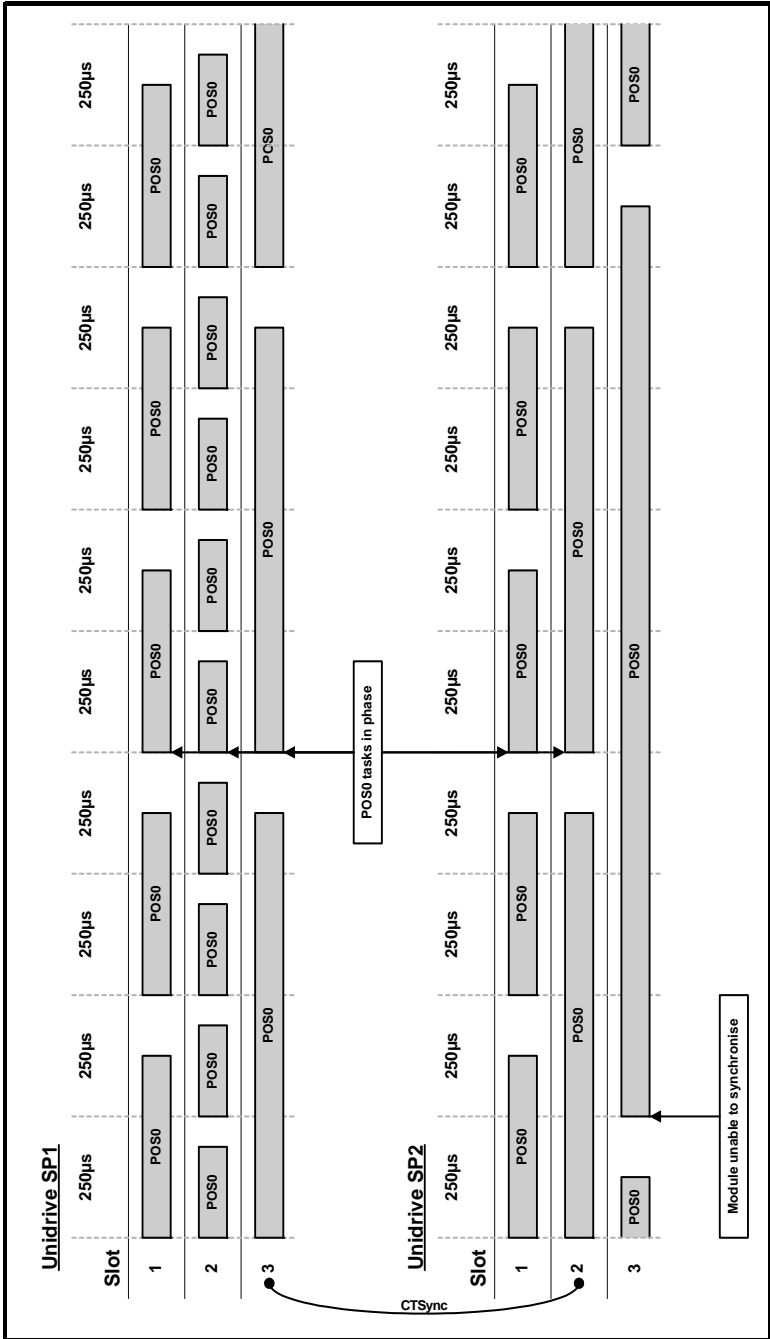
Before setting the parameters above, the POS tasks may be executing out of phase with each other as shown in Figure 10-3 *POS Task Execution BEFORE Inter-option Synchronisation with CTSync* . After setting the parameters to the values shown above the POS tasks should be executing in phase with each other as shown in Figure 10-4 *POS Task Execution AFTER Inter-option Synchronisation with CTSync* .

Figure 10-3 POS Task Execution BEFORE Inter-option Synchronisation with CTSync



Safety Information	Introduction	Installation	Getting Started	Parameters Programming	DPL Communications	Freeze and Marker	CTSync**	<b>Inter-option Synchronisation</b>	Diagnostics	Migration Guide	Quick Reference	Index
--------------------	--------------	--------------	-----------------	------------------------	--------------------	-------------------	----------	-------------------------------------	-------------	-----------------	-----------------	-------

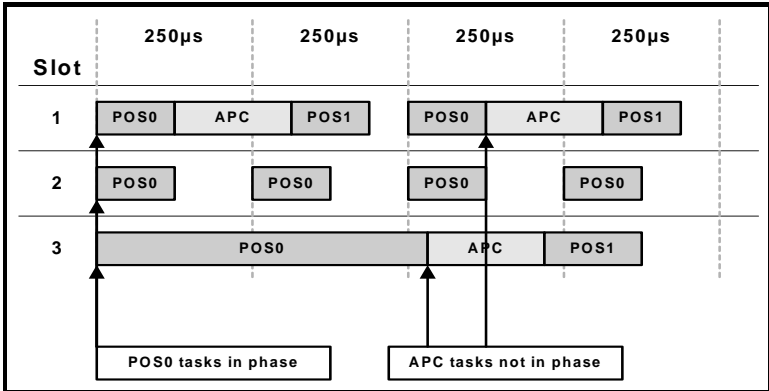
Figure 10-4 POS Task Execution AFTER Inter-option Synchronisation with CTSync



## 10.4 Position control tasks

The position control tasks consist of POS0, APC and POS1. The tasks are run in this order (POS0 runs first followed by the APC which is followed by POS1). Because the POS0 and APC tasks may not take the same time to execute on different modules the APC and POS1 tasks cannot be phased. Therefore when using Inter-option synchronisation the user must be aware that the position tasks as a whole for each module will start in phase, not the individual tasks. Figure 10-5 *Position control tasks* shows an example of this.

Figure 10-5 Position control tasks



# 11 Diagnostics

This chapter details the following:

- Run-time errors and trip codes
- Handling of run-time errors
- CTNet network status
- Support

## 11.1 Run-time Errors

A run-time error is an error which occurs in a specific operation of the Applications module. It can happen as a result of an error in the execution of the user's DPL program (such as trying to write to a parameter that doesn't exist, or trying to divide a value by zero), a misconfiguration (such as incorrect CTNet setup<sup>★♥</sup>) or a system error such as processor overload or *watchdog* time-out.

The action taken when an error occurs may be one or more of the following:

- User program may be halted or prevented from starting
- Drive may be tripped on **SLx.Er** (where x is the slot number) with the run-time error code placed into **Pr xx.50**
- Drive may be tripped on another **SLx.\*\*\*** code.
- The DPL **ERROR** task may be executed (if it exists).

Which of these occurs depends upon the type of error and the setting of the global run-time trip enable parameter **Pr xx.14**. This is detailed in section 11.3, below.

## 11.2 Unidrive SP Display Trip Codes

The table below shows the possible trip codes that will be displayed on the Unidrive SP when an error is detected in the Applications module which instigates a drive trip. Remember, not all run-time errors instigate a drive trip.

**Table 11.1 Unidrive SP Trip Display Codes**

Slot Where Trip Was Initiated			Fault	Description
Slot 1	Slot 2	Slot 3		
SL1.HF	SL2.HF	SL3.HF	Hardware Fault	Unidrive SP has detected that an option module is present, but is unable to communicate with it.
SL1.tO	SL2.tO	SL3.tO	Watchdog Timeout	Indicates a user program which has utilised the <i>watchdog</i> feature has failed to issue the WDOG command within 200ms.
SL1.Er	SL2.Er	SL3.Er	Error	Run-time trip generated by the Applications module either due to a user DPL program error or some other event. The actual error code is placed into <b>Pr xx.50</b> .
SL1.nF	SL2.nF	SL3.nF	Not Fitted	Module was disconnected while operational, or module has crashed. This trip will also occur if a Unidrive SP slot is configured for a Applications module, but the module is not fitted in the slot.
SL1.dF	SL2.dF	SL3.dF	Different Fitted	This trip will occur when an Applications module is fitted to a slot previously occupied by another solutions module, or is fitted to a previously unused slot.

## 11.3 Applications Module Run-time Error Codes

If the Applications module detects an error during operation the error code will be placed in the following parameter:

Pr 81.50	Applications Module Error Code		
Access	RO	Range	0 to 255
Default	N/A	Update Rate	On error

For certain errors the user may select if the Unidrive SP should trip as well. This is configured with the Global Run-time Trip enable parameter:

Pr 81.14	Global Run-time Trip Enable		
Access	RW	Range	0/1
Default	0	Update Rate	N/A

If set to 1 (On), the drive will trip on ALL run-time errors.

The table below shows the error codes and their meaning as well as if the drive will trip, the User program will stop and whether the DPL ERROR task will run.

### Notes:

- “**May**” under Drive Trip indicates that the drive will only trip if the global run-time trip enable parameter is set
- “**Not Run**” under Program Halted indicates that the error occurs at initialisation and the program will not be started.

**Table 11.2 Applications Module Error Codes**

Error Code	Reason	Trip Drive?	ERROR task?	Prog Halted?
39	User program stack overflow	Yes	No	Yes
40	Unknown error - please contact supplier	Yes	No	Yes
41	Parameter does not exist. User has attempted to read or write a non-existent parameter in the DPL program.	May	Yes	Yes
42	Attempt to write to a read-only parameter.	May	Yes	Yes
43	Attempt to read from a write-only parameter.	May	Yes	Yes
44	Parameter value out of range. (User has written an illegal value to a parameter within a DPL program.) If parameter #MM.17=0 the value written will be automatically limited and no error will occur.	May	Yes	Yes
45	Invalid synchronisation modes	Yes	No	Not Run
46	Unused	N/A	N/A	N/A
47	Synchronisation lost with CTSync Master. ♣♥	Yes	Yes	Yes
48	RS485 not in user mode. Occurs if user attempts to use a user-mode RS485 DPL command but the RS485 port is not in a user-mode. ♣♥	Yes	Yes	Yes
49	Invalid RS485 configuration. For example, invalid mode. ♣♥	Yes	Yes	Yes
50	Maths error - divide by zero or overflow.	May	Yes	Yes
51	Array index out of range. E.g. arr%[20] where arr% has only been DIMensioned to 19 elements.	May	Yes	Yes
52	Control word user trip. Instigated by setting the trip bit in the control word #90.11	Yes	No	No

**Table 11.2 Applications Module Error Codes (Continued)**

Error Code	Reason	Trip Drive?	ERROR task?	Prog Halted?
53	DPL program incompatible with target. For example, downloading a program compiled for UD70.	Yes	N/A	N/A
54	DPL task overrun. This occurs if the DPL code within a real-time task (e.g. POS0) cannot be completed in time. Use parameter #88.02 to identify the task in which this error occurred. Check that the task scheduling rate is correct and that there are no loops in the task. This can also occur as a result of external influences such as a large burst of data coming in over CTNet. This problem may be overcome by changing the CTNet priority so that it is lower than the POS tasks. This, however, may cause the CTNet task to be starved. Refer to parameter <i>Pr 81.44</i> on page 36 for further information.	May	Yes	Yes
55	Invalid encoder configuration. Only applies to system file V01.02.01 or earlier.	Yes	N/A	N/A
56	Invalid timer unit configuration ♣♥	Yes	Yes	Yes
57	Function block does not exist.	Yes	Yes	Not Run
58	Flash PLC Storage corrupt. Occurs at startup and will mean that the PLC register set (P/Q/T/U) and menu 20 will not have been restored. If this problem persists it may indicate a hardware failure so contact your supplier.	Yes	Yes	Not Run
59	Drive rejected application module as Sync master	Yes	Yes	Yes
60	CTNet hardware failure. Please contact your supplier. ♣♥	May	No	No
61	CTNet invalid configuration. Check all configuration parameters. ♣♥	May	No	No
62	CTNet invalid baud-rate. Check #MM.24 and network connections. ♣♥	May	No	No
63	CTNet invalid node ID. Check #MM.23. ♣♥	May	No	No
64	Digital Output overload. Both digital outputs will be taken inactive when this occurs and will remain inactive until the error condition is cleared. The trip threshold is 20mA. ♣♥	Yes	Yes	Yes
65	Invalid function block parameter(s). You have called a FB within a DPL program but one or more of the inputs are invalid.	Yes	Yes	Yes
66	User heap too large. The program has been compiled for a target that has more RAM than this one has. Occurs at startup.	Yes	No	Not Run
67	RAM file does not exist or a non-RAM file id has been specified.	Yes	Yes	Yes
68	The RAM file specified is not associated to an array.	Yes	Yes	Yes
69	Failed to update drive parameter database cache in Flash memory.	Yes	No	Not Run
70	User program downloaded while drive enabled. Will occur if #MM.37 = 1 and a program is downloaded.	May	No	Yes
71	Failed to change drive mode	Yes	No	Yes
72	Invalid CTNet buffer operation. ♣♥	Yes	Yes	Yes
73	Fast parameter initialisation failure	Yes	No	No

**Table 11.2 Applications Module Error Codes (Continued)**

Error Code	Reason	Trip Drive?	ERROR task?	Prog Halted?
74	Over-temperature	Yes	Yes	Yes
75	Hardware unavailable. The user program attempted to access unavailable hardware. e.g. if access is made to digital I/O, RS485 port or CTNet on SM-Applications Lite module.	Yes	Yes	Yes
76	Module type cannot be resolved. Module is not recognised.	Yes	No	Not Run
77	Inter-Option module comms error with module in slot 1.	Yes	Yes	Yes
78	Inter-Option module comms error with module in slot 2.	Yes	Yes	Yes
79	Inter-Option module comms error with module in slot 3.	Yes	Yes	Yes
80	Inter-Option module comms error with module unknown slot.	Yes	Yes	Yes
81	<ul style="list-style-type: none"> <li>• Slot selected as the reference or feedback does not contain a position option module</li> <li>• Attempt to change the Reference source or the Feedback source in more than one task.</li> </ul> APC internal error. See parameter #81.38. This may be caused by one of the following: <ul style="list-style-type: none"> <li>• CAM table too small</li> <li>• A change of too many CAM segments has occurred at the CAM table input</li> <li>• CAM is selected but size is zero</li> <li>• CAM absolute mode selected and Reset Index or Reset Position in segment is out of range</li> </ul>	May	Yes	Yes
82	Communications to drive faulty.	May	Yes	Yes

## 11.4 Handling Run-Time Errors with the ERROR task

Certain run-time errors will cause the DPL ERROR task to be invoked if it exists. This provides a convenient way to safely handle the error condition and take whatever action is necessary, such as a controlled stop of the system or signalling of an alarm.

When an ERROR task runs, all other DPL tasks will have been stopped. Therefore the ERROR task has exclusive execution rights. Once the ERROR task has completed, the DPL program ends and no other DPL tasks operate (though it is possible to reset and restart the program - more details on this below).

**NOTE** Drive trips do not cause the ERROR task to run. Only certain DPL program errors do.

Within the ERROR task all standard DPL commands may be used as well as most function blocks. All Unidrive SP and Applications module parameters can be accessed. The run-error code can be determined using this parameter:

Pr 88.01	Error Status / Reset		
<b>Access</b>	RW	<b>Range</b>	0 to 9999
<b>Default</b>	N/A	<b>Update Rate</b>	On error

This parameter has two purposes - when read it will return the run-time error code the same as **Pr 81.50** (note - it will not return drive trip codes). The parameter is cleared to zero on reset and when the user program execution is started.

Safety Information
Introduction
Installation
Getting Started
Parameters
DPL Programming
Communications
Freeze and Marker
CTSync
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

When the parameter is written to with a value of 1070 the Applications module will initiate a warm-restart of the drive and any other options. This can be used to restart the user program (providing auto-run **Pr 81.13=1**) and clear any drive trip. This reset action can be performed at any time, not just after a run-time error or in an ERROR task.



Writing 1070 to parameter Pr 88.01 will result in any drive trip being automatically cleared as well as resetting all fitted options in the Unidrive SP. This behaviour is different to the UD70 product on Unidrive Classic where the drive was not reset.

The task that caused a run-time error can be determined by reading parameter **Pr 88.02**, as previously described.

If the user wishes to trip the drive (if it hasn't already been tripped) then write the appropriate trip code into parameter **Pr 10.38**.

## 11.5 Resource monitoring

The SM-Applications Plus module provides more realistic resource data than is available from parameter **Pr 81.04** for SM-Applications Lite and SM-Applications. These are shown below.

Pr 88.03	POS Resource Monitoring*		
Access	RW	Range	0/1
Default	0	Update Rate	Immediate

This parameter allows the user to enable or disable monitoring of the motion engine tasks free resource. When set to 1, parameters **Pr 88.04** and **Pr 88.05** become active. If set to zero, parameters **Pr 88.04** and **Pr 88.05** will read zero.

Pr 88.04	Free Resource for Motion Engine Tasks*		
Access	RW	Range	0 to 95
Default	0	Update Rate	See Pr 81.12

This parameter indicates the percentage resource available for running the motion engine tasks. These tasks are CTSync, CTSync Output Channels, POS0, PLCopen, APC, APC Output Channel and POS1. If this parameter value reaches zero a task overrun will occur. It is calculated every motion engine period and is displayed for the previous motion engine period.

Pr 88.05	Motion Engine Peak Resource Detect*		
Access	RW	Range	0 to 95
Default	0	Update Rate	See Pr 88.04

This parameter shows the peak that parameter **Pr 88.04** reaches. It will show the lowest value that parameter **Pr 88.04** reaches since the monitoring was enabled (parameter **Pr 88.03**). It will give a realistic indication of the worst case available resources for the motion engine tasks so that the user can see how close the module has been to a motion engine task overrun.

Pr 88.06	CLOCK Task Resource Monitoring*		
Access	RO	Range	0/1
Default	NA	Update Rate	Immediate

This parameter allows the user to enable or disable monitoring of the CLOCK task free resource. When set to 1, parameters **Pr 88.07** and **Pr 88.08** become active. If set to zero, parameters **Pr 88.07** and **Pr 88.08** will read zero.

<b>Pr 88.07</b>	<b>Free Resource for Clock Task*</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 95
<b>Default</b>	NA	<b>Update Rate</b>	See Pr 81.11

This parameter indicates the percentage resource available for running the Clock task. If this parameter value reaches zero a task overrun will occur. It is calculated every Clock period and is displayed for the previous motion engine period.

<b>Pr 88.08</b>	<b>Clock Task Peak Resource Detect*</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 95
<b>Default</b>	NA	<b>Update Rate</b>	See Pr 81.11

This parameter shows the peak that parameter **Pr 88.07** reaches. It will show the lowest value that parameter **Pr 88.07** reaches since the monitoring was enabled (parameter **Pr 88.06**). It will give a realistic indication of the worst case available resources for the Clock task so that the user can see how close the module has been to a Clock task overrun.

## 11.6 Support

The information from the parameters described below should always be noted before contacting your supplier for technical support.

### 11.6.1 Module Firmware

<b>Pr 81.02</b>	<b>Firmware - Major Version</b>		
<b>Access</b>	RO	<b>Range</b>	00.00 to 99.99
<b>Default</b>	N/A	<b>Update Rate</b>	N/A

<b>Pr 81.51</b>	<b>Firmware - Minor Version</b>		
<b>Access</b>	RO	<b>Range</b>	0 to 99
<b>Default</b>	N/A	<b>Update Rate</b>	N/A

The full version of the Applications module firmware version can be read for the corresponding slot. This manual was written for Applications modules fitted with V01.04.05 firmware. The table below shows how to construct the full firmware version from these values.

**Table 11.3 Firmware Version**

Major Version	Minor Version	Firmware Version
1.04	3	V01.04.03

---

# 12 Migration Guide

---

This chapter outlines some of the major differences between the UD70 product for Unidrive Classic and the Applications module for Unidrive SP that may be helpful in converting (porting) user DPL programs.

## 12.1 Drive Parameter Differences

The Unidrive SP parameter set has quite a few differences compared to the Unidrive Classic. All parameters accessed in a DPL program should ideally have been configured as #Define's at the top of the program, thereby making conversions easier. So in porting, you will have to go through the program by hand and locate all drive parameter references. The first step is to search for the # symbol, and then go through the program looking for any function blocks that may take parameter pointer inputs (e.g. PFixRead or WRNET).

When parameters are located, make sure they are the same in the Unidrive SP and ensure that they are in the same units. Adjust the parameter number or scaling as appropriate.

## 12.2 UD70 Parameters

### 12.2.1 Setup Parameters

The set-up parameters can now be located in menu 15,16 or 17 whereas with UD70 they were always located in menu 17. It is strongly advised that new user DPL programs that change setup parameters are altered to use menu 81 which will eliminate the need to determine which of menu 15/16/17 is being used.

The table below outlines some of the major differences between the two setup menus:

**Table 12.1 Setup Parameter Changes**

Parameter	Description	Changes
Pr 17.03	Line number of error	This is now <b>Pr 81.48</b> .
Pr 17.06	EIA-RS485 mode	Modes 2,3,4,9,10,11,12 are no longer supported. The user modes 6,7,8 are not supported with the current version of software.
Pr 17.08	EIA-RS485 pointer	Not supported. Parameter is now turnaround delay
Pr 17.09	EIA-RS485 pointer 2	Not supported. Parameter is now TX enable delay
Pr 17.10	EIA-RS485 scale factor	Not supported. Parameter is now DPL print routing
Pr 17.11	Clock tick time	With drive version < 01.05.00 the default is zero. With drive versions >=01.05.00 the default is 10ms.
Pr 17.12	Position controller	This parameter now controls the scheduling rate for the new POS0/1 which are different to those of UD70 Speed and Encoder tasks.
Pr 17.15	EIA-RS485 trip mode	This parameter now allows the user to specify whether an Applications module will reset upon resetting the Unidrive SP from a trip condition..
Pr 17.16	IO Link sync. source	IO link is not supported. Parameter is now used to select the encoder update rate.
Pr 17.20	Flash power-down store	This parameter now takes immediate effect.
Pr 17.21	Disable MD29MON	This parameter now controls the saving of menu 20.
Pr 17.22	RS232 drive-drive	Not supported. Parameter now defines CTNet token ring ID.

**Table 12.1 Setup Parameter Changes**

Parameter	Description	Changes
Pr 17.23	EVENT trigger control	See Pr 81.35. This parameter is now CTNet ID.
Pr 17.24	ANSI 2-wire turn-around	See parameter Pr 81.08. Parameter now used by CTNet.
Pr 17.25+	Unused	There are now 51 set-up parameters.

**12.2.2 Menus 18/19**

These menus remain unchanged.

**12.2.3 Menu 20**

- Menu 20 now consists of 40 parameters rather than 50. Parameters **Pr 20.21 to Pr 20.40** are now full signed 32-bit parameters.
- Previously the first 20 parameters of menu 20 were considered to be reserved for communications options. This is now no longer the case since communications parameters are set up in menu 15/16/17, therefore all of menu 20 is free for use.
- The saving of menu 20 has also changed. In order to save the parameters you must set parameter **Pr 81.21** to a 1 and then set **Pr 81.19** or **Pr 81.20**. The reason for this change is that you can now potentially have two or three Applications modules fitted into the drive and since menu 20 is saved and restored from the flash memory of the module, only one module can be configured to look after menu 20 otherwise it will be incorrectly restored.

**12.2.4 PLC Registers**

- There are now an extra 200 **saveable** registers available. These are in register banks T and U (or menus 74 and 75). The P/Q/R/S register sets remain as they were, however since there is no built-in position controller, the Q register set is completely free for use.
- On the UD70 with CTNet, some registers in the R and S sets were reserved. This is no longer the case - all 100 registers of both sets can be used for CTNet cyclic data links (note that 3 registers in R and S are still reserved for use with easy mode cyclic data links).

**12.2.5 Menu 90 Parameters**

Substantial changes have occurred to parts of this menu and programs WILL have to be modified to take them into account. The encoder position parameters **Pr 90.01/Pr 90.03** now provide the full position information (including fine position) scaled such that the full 32-bit value represents one revolution. Separate revolution counters are available in **Pr 90.02** and **Pr 90.04**.

Safety Information
Introduction
Installation
Getting Started
Parameters
Programming
DPL
Communications
Freeze and Marker
CTSync**
Inter-option Synchronisation
Diagnosis
Migration Guide
Quick Reference
Index

## 12.3 General Features

### 12.3.1 Hardware

- CTNet on-board as standard\*\*
- RS232 programming port is no longer available. Programming is now done via the CT-RTU protocol through the RJ45 serial port on the front of the drive, or via CTNet\*\*. (The old DPL Toolkit software is not supported.)
- \*\*Digital I/O is now 24V positive logic rather than TTL. There are now two digital outputs instead of just the one. Also the digital I/O is now much more accessible via screw-terminals.

### 12.3.2 DPL Language

DPL remains backwardly compatible. The following enhancements have been made though:

- Addition of new constructs such as FOR...LOOP and SELECT...CASE.
- Nesting: The stack size is now allocated on a per-task basis and is larger than that of the UD70, so more nesting is allowed.
- New data type of single-precision floating point has been added that offers a small execution time advantage over double-precision (which remain the default). Single or double precision is specified as a global program option using the new \$FLT SINGLE directive in the program header.

#### NOTE

The single-precision data type is not a direct replacement for double-precision and users should make sure that the single precision type offers sufficient precision for their application.

### 12.3.3 Maths

- Expression complexity is now greatly improved in that on the UD70 you were limited to how much you could do in one expression. Now you can create expressions with much greater complexity.
- A new casting operator TRUNC has been added. This provides a float to integer conversion, truncating rather than rounding the result.
- Floating point comparison: With the UD70 a rather imprecise and unpredictable fuzzy compare was used for comparing floating point values. Now the Applications module uses a standard comparison method, as used in other programming languages, that is not fuzzy. However this may lead to an unexpected side effect. For example,

```
f = 1.2 * 3
IF f = 3.6 THEN
    // This would not yield true.
ENDIF
```

If checking for equality or not equality, use a range. For example:

```
f = 1.2 * 3
IF f > 3.59 or f < 3.61 THEN
    // This would yield true.
ENDIF
```

### 12.3.4 Tasks

The tasks have been changed.

- The ENCODER and SPEED tasks are no longer used. Instead the POS0 and POS1 tasks have been added. Also, the CLOCK task can be used as a replacement for the ENCODER task. The POS0, POS1 and CLOCK tasks run synchronously to the

drive (like the SPEED task did), but you can now specify a multiple of that time - from 250µs to 8ms for the POS0 and POS1 and 5ms to 200ms for the CLOCK task. Users may still use the names SPEED and ENCODER since they are aliases to POS0 and POS1, but be aware that the timing **is** different and code may have to be altered to take this into account. Note also that the timing does not alter depending on drive switching frequency, as it did on UD70.

- Three new EVENT tasks have been added, but this should not cause any problems migrating programs.
- EVENT tasks can now be scheduled from DPL using the new SCHEDULEEVENT function block.

### 12.3.5 User Defined Function Blocks

- The UD70 had a limit of 10 integer inputs and 10 integer output. There were also alignment restrictions. The Applications module has none of the restrictions. Number of inputs and outputs are limited only by memory. There should be no migration problems.

### 12.3.6 Communications

#### CTNet\*\*

- CTNet communications is now standard and no longer an option. All configuration parameters have changed. Sync generation is slightly different in that it is now in true milliseconds rather than the closest multiple of 1.38ms as UD70. Also EVENT task triggering will also occur on the master node as well as slave nodes - UD70 didn't do this.
- All set-up parameters have moved from menu 20 to the option menu (15/16/17/81).
- The special diagnostic variables such as NOFMESSAGES are no longer supported. Instead a new function block CTNETDIAGS returns the diagnostic information. Old programs using these diagnostic variables will not compile and will have to be changed to use the new function block.
- The EVENT task triggering on SYNC message now also occurs on the node that is actually generating the SYNC message as well as slave nodes.

#### EIA-RS485 Port\*\*

- CT-ANSI enhanced to now support full 32-bit read/write.
- IO box mode 10 is no longer supported.
- Drive to drive and cascade modes 2,3 and 4 are not supported.
- RS232 drive to drive mode is not supported.
- Mode 9 is not supported due to hardware limitations.
- Modes 11-12 are not supported. Mode 11 which bypassed the UD70's software buffers is no longer required since software buffers will not be used for user modes.
- New Modbus-RTU master modes and associated commands are now available.

## 12.4 Applications Module Porting aid

### 12.4.1 Overview

SyPTPro includes a Porting aid which will warn users of any features that are different on the Applications modules compared with the UD7X modules. The warnings will be displayed by SyPTPro in an error window. This error window will include the line number that the error/porting information is related to, so by double-clicking the line in the error

window SyPTPro will show the DPL source line containing the changed item. This porting aid can be activated by using the new \$PORTING directive after the main header information:

```
$AUTHOR    Kevi n Vedmore
$COMPANY   Control Techni ques
$TITLE     Test Program
$VERSION   V1.0
$DRIVE     UNI DRVSP
$PORTING
```

## 12.4.2 Reported Differences

The following kinds of information will be reported by SyPTPro when the \$PORTING directive is used:

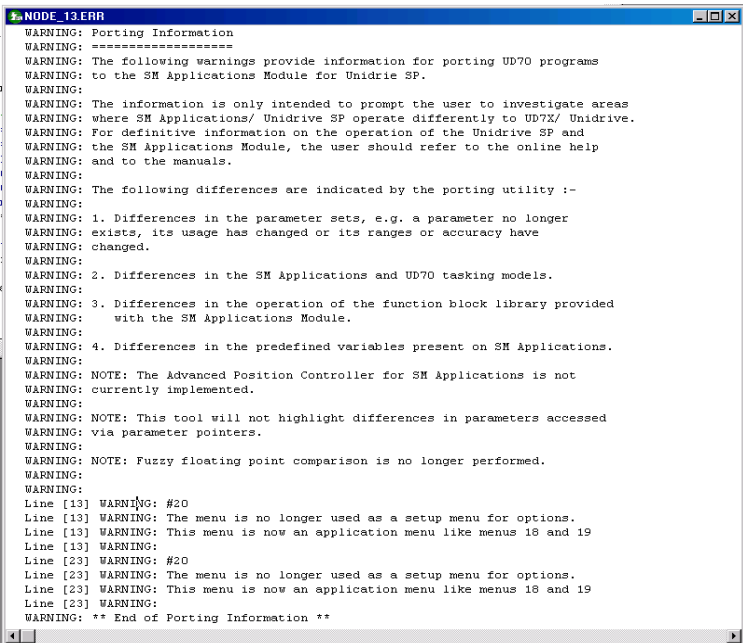
1. POS0 and POS1 tasks will be recommended instead of the older Speed, Torque and Encoder tasks. The Clock task will be recommended as an alternative to using a Pos task in place of an Encoder task.
2. Differences in the Unidrive SP or Applications module parameter sets, e.g. the number of decimal places has changed, names and usage has changed and the parameter no longer exists or has moved.

Note: Specific warnings about the number of decimal places changing will only be shown if a parameter is accessed using the #INT notation. An overall warning about decimal place changes will be shown for non #INT parameter accesses.

3. Different operation of RTL and OS function blocks.

An example of the type of messages given by the Porting aid is shown below.

Figure 12-1 Porting window information



```

WARNING: Porting Information
WARNING: =====
WARNING: The following warnings provide information for porting UD70 programs
WARNING: to the SM Applications Module for Unidrie SP.
WARNING:
WARNING: The information is only intended to prompt the user to investigate areas
WARNING: where SM Applications/ Unidrive SP operate differently to UD7X Unidrive.
WARNING: For definitive information on the operation of the Unidrive SP and
WARNING: the SM Applications Module, the user should refer to the online help
WARNING: and to the manuals.
WARNING:
WARNING: The following differences are indicated by the porting utility :-
WARNING:
WARNING: 1. Differences in the parameter sets, e.g. a parameter no longer
WARNING: exists, its usage has changed or its ranges or accuracy have
WARNING: changed.
WARNING:
WARNING: 2. Differences in the SM Applications and UD70 tasking models.
WARNING:
WARNING: 3. Differences in the operation of the function block library provided
WARNING: with the SM Applications Module.
WARNING:
WARNING: 4. Differences in the predefined variables present on SM Applications.
WARNING:
WARNING: NOTE: The Advanced Position Controller for SM Applications is not
WARNING: currently implemented.
WARNING:
WARNING: NOTE: This tool will not highlight differences in parameters accessed
WARNING: via parameter pointers.
WARNING:
WARNING: NOTE: Fuzzy floating point comparison is no longer performed.
WARNING:
WARNING:
Line [13] WARNING: #20
Line [13] WARNING: The menu is no longer used as a setup menu for options.
Line [13] WARNING: This menu is now an application menu like menus 18 and 19
Line [13] WARNING:
Line [23] WARNING: #20
Line [23] WARNING: The menu is no longer used as a setup menu for options.
Line [23] WARNING: This menu is now an application menu like menus 18 and 19
Line [23] WARNING:
WARNING: ** End of Porting Information **
```

# 13 Quick Reference

Refer to section 5 *Parameters* on page 25 for full details of these parameters.

**Table 13.1 Setup Parameters**

Parameter	Description	Range	Default
Pr 81.01	Module option code <sup>♦♦♦</sup>	0-499	N/A
Pr 81.02	Module firmware version <sup>♦♦♦</sup>	0-99.99	N/A
Pr 81.03	DPL program status <sup>♦♦♦</sup>	0-3	N/A
Pr 81.04	Available system resource % <sup>♦♦♦</sup>	0-100%	N/A
Pr 81.05	EIA-RS485 address <sup>♦♦</sup>	0-255	11
Pr 81.06	EIA-RS485 Mode <sup>♦♦</sup>	0-255	1
Pr 81.07	EIA-RS485 Baud rate <sup>♦♦</sup>	0-9	4
Pr 81.08	EIA-RS485 Turn-around delay <sup>♦♦</sup>	0-255ms	2ms
Pr 81.09	EIA-RS485 Tx Enable Delay <sup>♦♦</sup>	0-1ms	0ms
Pr 81.10	DPL print routing <sup>♦♦♦</sup>	0-1	0
Pr 81.11	Clock tick time (ms) <sup>♦♦♦</sup>	0-200ms	0
Pr 81.12	POS task schedule rate <sup>♦♦♦</sup>	0-6	0
Pr 81.13	Enable autorun <sup>♦♦♦</sup>	0-1	1
Pr 81.14	Global run-time trip enable <sup>♦♦♦</sup>	0-1	0
Pr 81.15	Disable Reset on Trip Cleared <sup>♦♦♦</sup>	0-1	0
Pr 81.16	Encoder Data Update Rate <sup>♦♦♦</sup>	0-3	0
Pr 81.17	Enable parameter over-range trips <sup>♦♦♦</sup>	0-1	0
Pr 81.18	Watchdog enable <sup>♦♦♦</sup>	0-1	0
Pr 81.19	Save request <sup>♦♦♦</sup>	0-1	0
Pr 81.20	Save on 'UU' trip <sup>♦♦♦</sup>	0-1	0
Pr 81.21	Include menu 20 for save/restore <sup>♦♦♦</sup>	0-1	0
Pr 81.22	CTNet token ring ID <sup>♦♦</sup>	0-255	0
Pr 81.23	CTNet node address <sup>♦♦</sup>	0-255	0
Pr 81.24	CTNet baud-rate <sup>♦♦</sup>	0-3	1
Pr 81.25	CTNet sync setup (SSFF) <sup>♦♦</sup>	0-9999	0
Pr 81.26	CTNet Easy Mode 1 destination node (NNSS) <sup>♦♦</sup>	0-25503	0
Pr 81.27	CTNet Easy Mode 1 source parameter (MMPP) <sup>♦♦</sup>	0-9999	0
Pr 81.28	CTNet Easy Mode 2 destination node (NNSS) <sup>♦♦</sup>	0-25503	0
Pr 81.29	CTNet Easy Mode 2 source parameter (MMPP) <sup>♦♦♦</sup>	0-9999	0
Pr 81.30	CTNet Easy Mode 3 destination node (NNSS) <sup>♦♦</sup>	0-25503	0
Pr 81.31	CTNet Easy Mode 3 source parameter (MMPP) <sup>♦♦</sup>	0-9999	0
Pr 81.32	CTNet Easy Mode Slot 1 destination (MMPP) <sup>♦♦</sup>	0-9999	0
Pr 81.33	CTNet Easy Mode Slot 2 destination (MMPP) <sup>♦♦</sup>	0-9999	0
Pr 81.34	CTNet Easy Mode Slot 3 destination (MMPP) <sup>♦♦</sup>	0-9999	0
Pr 81.35	CTNet Sync EVENT task ID <sup>♦♦</sup>	0-4	0

Safety Information
Introduction
Installation
Getting Started
Parameters
DPL Programming
Communications
Freeze and Marker
CTSync <sup>♦♦</sup>
Inter-option Synchronisation
Diagnostics
Migration Guide
<b>Quick Reference</b>
Index

**Table 13.1 Setup Parameters (Continued)**

Parameter	Description	Range	Default
Pr 81.36	CTNet Diagnostic parameter**	-3-32767	N/A
Pr 81.37	Reject download if drive enabled**	0-1	0
Pr 81.38	APC Run-time trip**	0-1	0
Pr 81.39	Inter-option module Drive Synchronisation Status**	0-3	0
Pr 81.42	Freeze feedback encoder position**	0-1	0
Pr 81.43	Freeze invert**	0-1	0
Pr 81.44	Task priority level**	0-255	0
Pr 81.48	DPL line number of error**	32bit	0
Pr 81.49	User program ID**	16bit	0
Pr 81.50	Run-time error code**	0-255	0
Pr 81.51	Minor software revision**	0-99	N/A

**Table 13.2 Timer Unit Parameters\*\***

Parameter	Description	Range	Default
Pr 85.01	Timer Unit Control Word**	13bit	N/A
Pr 85.02	Timer Unit Status Word**	0-3	N/A
Pr 85.03	Timer Unit 16-bit Timer Count**	16bit	N/A
Pr 85.04	Timer Unit Wrap-around Limit**	16bit	N/A
Pr 85.05	Timer Unit Timer Capture Cache**	16bit	N/A

**Table 13.3 Digital I/O Parameters\*\***

Parameter	Description	Range	Default
Pr 86.01	Digital Input 0**	0-1	N/A
Pr 86.02	Digital Input 1**	0-1	N/A
Pr 86.03	Digital Output 0**	0-1	N/A
Pr 86.04	Digital Output 1**	0-1	N/A
Pr 86.05	Digital Outputs 0 and 1**	0-3	N/A

**Table 13.4 Status Parameters**

Parameter	Description	Range	Default
Pr 88.01	Error Code / Reset**	0-9999	N/A
Pr 88.02	Task In Error**	0-50	N/A
Pr 88.03	POS Resource Monitoring*	0-1	0
Pr 88.04	Free Resource for Motion Engine Tasks*	0-95	N/A
Pr 88.05	Motion Engine Peak Resource Detect*	0-95	N/A
Pr 88.06	CLOCK Task Resource Monitoring*	0-1	0
Pr 88.07	Free Resource for Clock Task*	0-95	N/A
Pr 88.08	Clock task peak resource detect*	0-95	N/A

**Table 13.5 Applications Module Error Codes**

Error Code	Reason	Trip Drive?	ERROR task?	Prog Halted?
39	User program stack overflow	Yes	No	Yes
40	Unknown error - please contact supplier	Yes	No	Yes
41	Parameter does not exist. User has attempted to read or write a non-existent parameter in the DPL program.	May	Yes	Yes
42	Attempt to write to a read-only parameter.	May	Yes	Yes
43	Attempt to read from a write-only parameter.	May	Yes	Yes
44	Parameter value out of range. (User has written an illegal value to a parameter within a DPL program.) If parameter #MM.17=0 the value written will be automatically limited and no error will occur.	May	Yes	Yes
45	Invalid synchronisation modes	Yes	No	Not Run
46	Unused	N/A	N/A	N/A
47	Synchronisation lost with CTSync Master. ♣♥	Yes	Yes	Yes
48	RS485 not in user mode. Occurs if user attempts to use a user-mode RS485 DPL command but the RS485 port is not in a user-mode. ♣♥	Yes	Yes	Yes
49	Invalid RS485 configuration. For example, invalid mode. ♣♥	Yes	Yes	Yes
50	Maths error - divide by zero or overflow.	May	Yes	Yes
51	Array index out of range. E.g. arr%[20] where arr% has only been DIMensioned to 19 elements.	May	Yes	Yes
52	Control word user trip. Instigated by setting the trip bit in the control word #90.11	Yes	No	No
53	DPL program incompatible with target. For example, downloading a program compiled for UD70.	Yes	N/A	N/A
54	DPL task overrun. This occurs if the DPL code within a real-time task (e.g. POS0) cannot be completed in time. Use parameter #88.02 to identify the task in which this error occurred. Check that the task scheduling rate is correct and that there are no loops in the task. This can also occur as a result of external influences such as a large burst of data coming in over CTNet. This problem may be overcome by changing the CTNet priority so that it is lower than the POS tasks. This, however, may cause the CTNet task to be starved. Refer to parameter <i>Pr 81.44</i> on page 36 for further information.	May	Yes	Yes
55	Invalid encoder configuration. Only applies to system file V01.02.01 or earlier.	Yes	N/A	N/A
56	Invalid timer unit configuration ♣♥	Yes	Yes	Yes
57	Function block does not exist.	Yes	Yes	Not Run
58	Flash PLC Storage corrupt. Occurs at startup and will mean that the PLC register set (P/Q/T/U) and menu 20 will not have been restored. If this problem persists it may indicate a hardware failure so contact your supplier.	Yes	Yes	Not Run
59	Drive rejected application module as Sync master	Yes	Yes	Yes
60	CTNet hardware failure. Please contact your supplier. ♣♥	May	No	No

Safety Information
Introduction
Installation
Getting Started
Parameters
DPL Programming
Communications
Freeze and Marker
CTSync
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

**Table 13.5 Applications Module Error Codes (Continued)**

Error Code	Reason	Trip Drive?	ERROR task?	Prog Halted?
61	CTNet invalid configuration. Check all configuration parameters. ♣♥	May	No	No
62	CTNet invalid baud-rate. Check #MM.24 and network connections. ♣♥	May	No	No
63	CTNet invalid node ID. Check #MM.23. ♣♥	May	No	No
64	Digital Output overload. Both digital outputs will be taken inactive when this occurs and will remain inactive until the error condition is cleared. The trip threshold is 20mA. ♣♥	Yes	Yes	Yes
65	Invalid function block parameter(s). You have called a FB within a DPL program but one or more of the inputs are invalid.	Yes	Yes	Yes
66	User heap too large. The program has been compiled for a target that has more RAM than this one has. Occurs at startup.	Yes	No	Not Run
67	RAM file does not exist or a non-RAM file id has been specified.	Yes	Yes	Yes
68	The RAM file specified is not associated to an array.	Yes	Yes	Yes
69	Failed to update drive parameter database cache in Flash memory.	Yes	No	Not Run
70	User program downloaded while drive enabled. Will occur if #MM.37 = 1 and a program is downloaded.	May	No	Yes
71	Failed to change drive mode	Yes	No	Yes
72	Invalid CTNet buffer operation. ♣♥	Yes	Yes	Yes
73	Fast parameter initialisation failure	Yes	No	No
74	Over-temperature	Yes	Yes	Yes
75	Hardware unavailable. The user program attempted to access unavailable hardware. e.g. if access is made to digital I/O, RS485 port or CTNet on SM-Applications Lite module.	Yes	Yes	Yes
76	Module type cannot be resolved. Module is not recognised.	Yes	No	Not Run
77	Inter-Option module comms error with module in slot 1.	Yes	Yes	Yes
78	Inter-Option module comms error with module in slot 2.	Yes	Yes	Yes
79	Inter-Option module comms error with module in slot 3.	Yes	Yes	Yes
80	Inter-Option module comms error with module unknown slot.	Yes	Yes	Yes
81	<ul style="list-style-type: none"> <li>• Slot selected as the reference or feedback does not contain a position option module</li> <li>• Attempt to change the Reference source or the Feedback source in more than one task.</li> </ul> APC internal error. See parameter #81.38. This may be caused by one of the following: <ul style="list-style-type: none"> <li>• CAM table too small</li> <li>• A change of too many CAM segments has occurred at the CAM table input</li> <li>• CAM is selected but size is zero</li> <li>• CAM absolute mode selected and Reset Index or Reset Position in segment is out of range</li> </ul>	May	Yes	Yes
82	Communications to drive faulty.	May	Yes	Yes

**Table 13.6 Terminals\*\***

Terminal	Function	Description
1	0V SC	0V connection for EIA-RS485 port
2	/RX	EIA-RS485 Receive line (negative). Incoming.
3	RX	EIA-RS485 Receive line (positive). Incoming.
4	/TX	EIA-RS485 Transmit line (negative). Outgoing.
5	TX	EIA-RS485 Transmit line (positive). Outgoing.
6	CTNet A	CTNet data line
7	CTNet Shield	Shield connection for CTNet
8	CTNet B	CTNet data line
9	0V	0V connection for digital I/O
10	DIGIN0	Digital input 0
11	DIGIN1	Digital input 1
12	DIGOUT0	Digital output 0
13	DIGOUT1	Digital output 1

Safety Information
Introduction
Installation
Getting Started
Parameters Programming
DPL Communications
Freeze and Marker
CTSync**
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
Index

---

# Index

---

## C

Communications .....	28, 70, 107
Connections .....	13, 15
CTNet .....	13, 14, 19, 107
CTSync .....	83

## D

Diagnostics .....	98
Digital I/O .....	40, 110
Downloading .....	24
DPL .....	22, 56, 63, 106

## E

Error Codes .....	99
-------------------	----

## F

Firmware .....	103
Firmware Version .....	103
Freeze .....	79, 80

## H

Hardware .....	106
----------------	-----

## I

Installation .....	11, 18
Inter-option .....	89

## M

Mapping parameters (fieldbus) .....	77
Marker .....	79, 82
Maths .....	106
Migration Guide .....	104
Modbus ASCII .....	76
Modbus-RTU .....	72
Motion Engine .....	85

## P

Parameters .....	25, 61
PLC Registers .....	37, 61, 105
Porting .....	107

## Q

Quick Reference .....	109
-----------------------	-----

## R

RS485 .....	15, 19
Run-time Errors .....	98

## S

Safety .....	6
Setup Parameter Changes .....	104
Setup Parameters .....	109

SMARTCARD .....	18
Specifications .....	8
Status .....	110
SyPTLite .....	19, 21
SyPTPro .....	19, 20, 24
<b>T</b>	
Tasks .....	57, 106
Terminals .....	113
Termination .....	14, 17
Timer .....	38
Timer Unit .....	110
Trip Codes .....	98
<b>U</b>	
UDFB .....	68
<b>V</b>	
Variables .....	59
Virtual Master .....	87

Safety Information
Introduction
Installation
Getting Started
Parameters
DPL Programming
Communications
Freeze and Marker
CTSync <sup>▲▼</sup>
Inter-option Synchronisation
Diagnostics
Migration Guide
Quick Reference
<b>Index</b>



**0471-0062-02**