User Guide

SI-Applications Plus

Part Number: 0478-0009-02
Issue: 2

Control Techniques™

Leroy-Somer™

Original Instructions

For the purposes of compliance with the EU Machinery Directive 2006/42/EC.

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 with the motor.

Where a malfunction might result in a hazard, it is essential to refer to this guide and the relevant drive documentation. The drive documentation is available for download from:

http://www.emersonindustrial.com/en-EN/controltechniques/downloads/userguidesandsoftware/Pages/downloads.aspx

or

www.emersonindustrial.com/en-EN/leroy-somer-motors-drives/downloads/Pages/manuals.aspx

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 the 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.

Firmware statement

This product is supplied with the latest firmware version. When retro-fitting to an existing system, all firmware versions should be verified to confirm the same functionality as products of the same type already present. This also applies to products returned from a Service Centre or Repair Centre. If there is any doubt please contact the supplier of the product. The firmware version of the product can be identified by looking at Pr **MM.002** where **MM** is the relevant menu number for the module slot being used.

Environmental Statement

We are committed to minimising the environmental impacts of our manufacturing operations and of our products throughout their life cycle. To this end, we operate an Environmental Management System (EMS) which is certified to the International Standard ISO 14001. Further information on the EMS, our Environmental Policy and other relevant information is available on request, or can be found at

http://www.emersonindustrial.com/en-EN/controltechniques/aboutus/environment/Pages/environment.aspx

Electronic variable-speed drives have the potential to save energy and (through increased machine/process efficiency) reduce raw material consumption and scrap throughout their long working lifetime. In typical applications, these positive environmental effects far outweigh the negative impacts of product manufacture and end-of-life disposal.

Nevertheless, when the products eventually reach the end of their useful life, they must not be discarded but should instead be recycled by a specialist recycler of electronic equipment. Recyclers will find the products easy to dismantle into their major component parts for efficient recycling. Many parts snap together and can be separated without the use of tools, while other parts are secured with conventional fasteners. Virtually all parts of the product are suitable for recycling.

Product packaging is of good quality and can be re-used. Large products are packed in wooden crates, while smaller products come in strong cardboard cartons which themselves have a high recycled fibre content. If not re-used, these containers can be recycled. Polythene, used on the protective film and bags for wrapping product, can be recycled in the same way. Our packaging strategy prefers easily-recyclable materials of low environmental impact, and regular reviews identify opportunities for improvement.

When preparing to recycle or dispose of any product or packaging, please observe local legislation and best practice.

REACH Legislation

EC Regulation 1907/2006 on the Registration, Evaluation, Authorisation and restriction of Chemicals (REACH) requires the supplier of an article to inform the recipient if it contains more than a specified proportion of any substance which is considered by the European Chemicals Agency (ECHA) to be a Substance of Very High Concern (SVHC) and is therefore listed by them as a candidate for compulsory authorisation.

For current information on how this requirement applies to our products, please approach your usual contact in the first instance. Our position statement can be viewed at:

www.emersonindustrial.com/en-EN/controltechniques/aboutus/environment/reachregulation/Pages/reachregulation.aspx

Copyright © November 2016

The information contained in this guide is for guidance only and does not form part of any contract. The accuracy cannot be guaranteed as Emerson have an ongoing process of development and reserve the right to change the specification of their products without notice.

Control Techniques Limited. Registered Office: The Gro, Newtown, Powys SY16 3BE. Registered in England and Wales. Company Reg. No. 01236886.

Moteurs Leroy-Somer SAS. Headquarters: Bd Marcellin Leroy, CS 10015, 16915 Angoulême Cedex 9, France. Share Capital: 65 800 512 €, RCS Angoulême 338 567 258.

Issue Number: 2

Firmware: V02.10.00

Contents

1 Safety information 6
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 Access 7
1.6 Compliance with Regulations 7
1.7 Motor 7
1.8 Adjusting Parameters 7
1.9 General safety considerations for remote operation 7

2 Introduction 8
2.1 Features 8
2.2 Option module identification 9
2.3 Conventions used in this guide 10
2.4 PC Development Software 10
2.5 User Knowledge 10

3 Installation 11
3.1 General Installation 11
3.2 Electrical Connections 12
3.3 CTNet connections 13
3.4 CTNet cable 14
3.5 CTNet network termination 14
3.6 EIA-RS485 connections 15
3.7 Digital I/O connections 17
3.8 Port Isolation 17

4 Getting started 18
4.1 Using SyPTPro 18
4.2 Connecting the PC to the SI-Applications Plus 18
4.3 CTNetAPI routing 19
4.4 Configuring communications within SyPTPro 19
4.5 Creating a Node in SyPTPro 20
4.6 DPL Programming basics 20
4.7 Program Example 21
4.8 Downloading programs 22

5 Parameters 23
5.1 Overview 23
5.2 Saving Parameters 23
5.3 Menu 81 - Configuration parameters 24
5.4 Menus 70-79 - PLC Registers 37
5.5 Menu 85 - Timer Function Parameters 38
5.6 Menu 86 - Digital I/O Parameters 41
5.7 Menu 88 - Status Parameters 42
5.8 Menu 90 - General Parameters 44
5.9 Menu 91 - Fast access parameters 52
5.10 Menu 97 - Internal Motion Processor Parameters 56
5.11 Menus 18,19 - Application Parameters 56
5.12 Menu 20 - Application Menu 57

6 Communications 58
6.1 EIA-RS485 serial communications port 58
6.2 CTNet 65
6.3 SI-Applications Plus mapping parameters (fieldbus) 65

7 DPL Programming 66
7.1 Program Header 66
7.2 Tasks 67
7.3 Variables 69
7.4 Parameters 71
7.5 Operators 73
7.6 Basic DPL Commands 73
7.7 User Defined Function Blocks 78

8 Freeze and marker 81
8.1 Freeze input 81
8.2 Marker pulse 83

9 CTSync 85
9.1 Overview 85
9.2 Drive synchronisation source 85
9.3 Connections 85
9.4 Limitations 86
9.5 CTSync Function Blocks 86
9.6 Motion Engine 88
9.7 Virtual Master Example 90

10 Diagnostics 92
10.1 Run-time errors 92
10.2 Drive display trip codes 92
10.3 SI-Applications Plus module run-time error codes 93
10.4 Handling Run-Time Errors with the ERROR task 96
10.5 Resource monitoring 97
10.6 Support 98

11 Migration guide 99
11.1 SyPTPro Porting Tool 99
11.2 Module parameter differences 100
11.3 Drive parameter differences 101
12 Quick reference 102

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.

NOTE 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/start-up and maintenance must be carried out by personnel who have the necessary training and experience. They must read this safety information and this guide carefully.

The STOP and Safe Torque Off 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 Safe Torque Off 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 behavior 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 Safe Torque Off function 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 drive documentation 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 Access

Drive access must be restricted to authorized personnel only. Safety regulations which apply at the place of use must be complied with.

1.6 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 ground (earth) connections.

The drive documentation 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:

2006/42/EC: Safety of machinery.

2014/30/EU: Electromagnetic Compatibility Directive.

1.7 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 installed 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 **00.046** motor rated current. This affects the thermal protection of the motor.

1.8 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.

1.9 General safety considerations for remote operation

SI-Applications Plus enables the possibility of remotely controlling a machine from a distance. It is vital that when connecting to a machine remotely, adequate safety procedures are implemented to prevent damage to the machine or injury to personnel.

Any connection to a live system has the possibility of altering the state of the machine, adequate procedures must be implemented to cover this situation. It is the responsibility of the machine builder to ensure that such a system is safe and complies with current legislation.

2 Introduction

2.1 Features

Modern variable speed drives 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 certain drives can be substantially increased by using a SI-Applications Plus module. The SI-Applications Plus module provides an additional processor for the drive and allow the user to utilise existing, or write their own, application-specific software. It also offers 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 SI-Applications Plus is a System Integration Module that can be installed to Option Module slot 3 on Unidrive M. The module is powered from the drive's internal power supply.

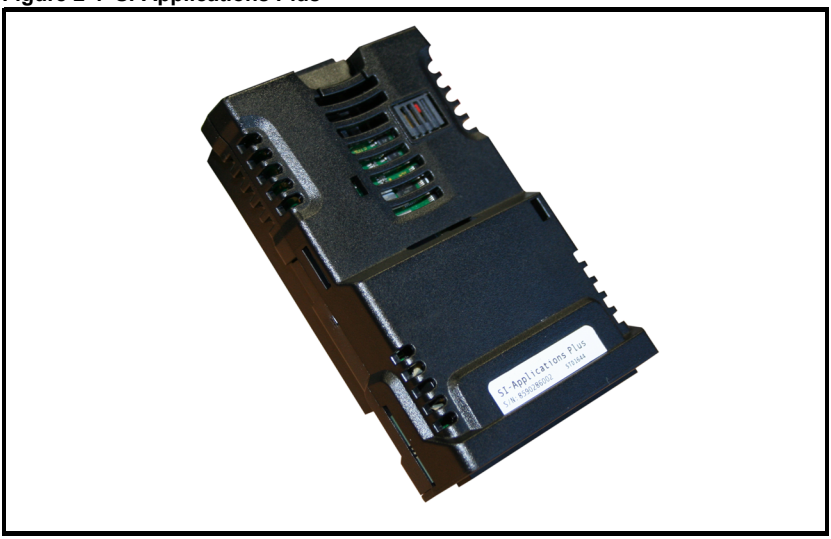
Specifications for SI-Applications Plus

- Enhanced high speed dedicated microprocessor
- 512 kB Flash memory for user program
- 200 kB user program memory
- EIA-RS485 port offering ANSI, Modbus-RTU slave and master and Modbus-ASCII slave and master protocols
- CTNet high speed network connection offering up to 5 Mbit/s data rate.
- Two 24 V digital inputs
- Two 24 V digital outputs
- Task based programming system allowing for real-time control of drive and process
- CTSync.

NOTE SyPTPro V2.6.0 or later is required to use the SI-Applications Plus module.

2.2 Option module identification

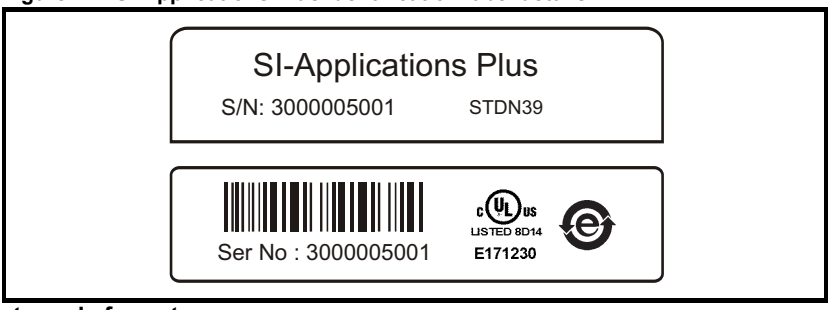
Figure 2-1 SI-Applications Plus



The SI-Applications Plus module can be identified by:

1. The label located on the top of the option module.

Figure 2-2 SI-Applications Plus Identification label details



2.2.1 Date code format

The date code is split into two sections: a letter followed by a number.
The letter indicates the year, and the number indicates the week number (within the year) in which the option module was built. The letters start with A for 1991 (B for 1992, C for 1993 etc.).
Example: A date code of W31 would correspond to week 31 of year 2013.

2.3 Conventions used in this guide

The configuration of the host drive and System Integration Module is done using menus and parameters. A menu is a logical collection of parameters that have similar functionality.

In the case of a System Integration Module, the parameters will appear in menu 15, 16 or 17 for the drive depending on the slot the module is installed into. The SI-Applications Plus module can be physically installed into option slot 3 only, with the parameters for the modules appearing in menu 17. The module parameters can be made to appear in menus 15 or 16 by setting Pr **11.056** (Option Slot Identifiers) on the drive.The method used to determine the menu or parameter is as follows:

- Pr **MM.000** - signifies any menu and parameter number 00.
- Pr **MM.PPP** - where **MM** signifies the the menu allocated to the System Integration Module (this could be 15, 16 or 17) and **PPP** signifies the parameter number.

2.4 PC Development Software

Application programs for the SI-Applications Plus may be developed by the user with the SyPT software tools.

SyPTPro offers various tools to help in developing solutions:

- Configuration editor for configuring drives and connections on CTNet, EtherNet, 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 SI-Applications Plus by either:

- Direct connection to the EIA-RS485 or Ethernet port on the front of the Drive.
- Connecting to one or more options on a CTNet network (a CTNet interface card for the PC will be required). See section *Features* on pages 8, 9 & 10 for availability of CTNet on your SI-Applications Plus module
- It is not possible to connect to the SI-Applications Plus module via the Ethernet ports of the SI-Ethernet, MCi210 or SI-PROFINET V2 modules.

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

Before installing or removing a System Integration Module in any drive, ensure the AC supply has been disconnected for at least 10 minutes and refer to Chapter 1 *Safety information* on page 6. If using a DC bus supply ensure this is fully discharged before working on any drive or System Integration Module.

WARNING

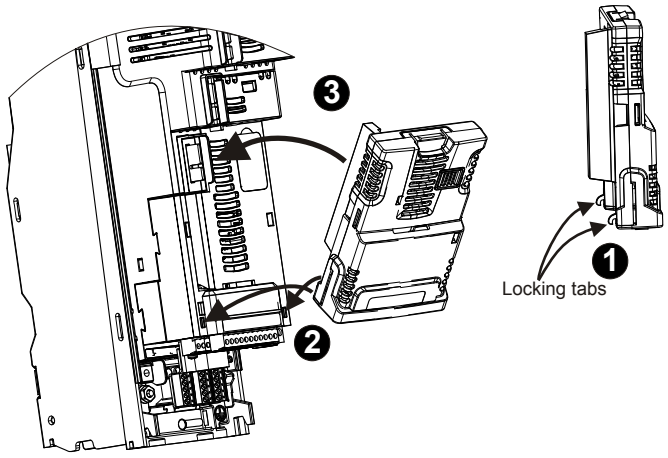
When an SI-Applications Plus module is installed in the drive, the drive must not be used on a corner grounded or centre-grounded delta supply if the supply voltage is above 300 V. If this is required, please contact the supplier of the drive for more information.

WARNING

3.1 General Installation

The installation of a large System Integration Module is illustrated in Figure 3-1.

Figure 3-1 Installing a large System Integration Module

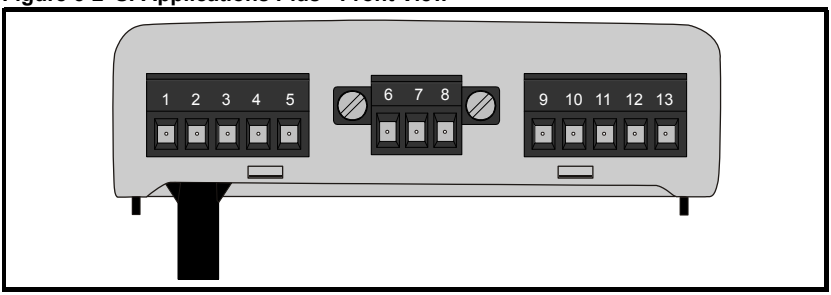


1. Identify the two locking tabs on the rear of the large Option Module
2. Hold the Option Module at a slight angle to the drive, and insert the locking tabs into the slots
3. Push the top of the module into the drive until it locates in the drive body

The System Integration Module connector is located on the underside of the module (1). Push this into the System Integration Module slot located on the drive until it clicks into place (2). For further information, refer to the appropriate drive manual. See the inside front cover of this guide for information on downloading manuals.

3.2 Electrical Connections

Figure 3-2 SI-Applications Plus - Front View



The terminal functions are given in Table 3-1.

NOTE Please ensure that the drive is off before removing any modules.
Please refer to your installation sheet for more information.

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	DI0	Digital input 0
11	DI1	Digital input 1
12	DO0	Digital output 0
13	DO1	Digital output 1

Table 3-2 Digital input specifications

Terminal 10 / Digital Input 0	
Terminal 11 / Digital Input 1	
Type	Positive logic IEC 61131-2
Maximum Input Voltage	+/- 30 V
Switching Threshold	9.5 V +/- 0.3 V
Load	2 mA at +15 V

Table 3-3 Digital output specifications

Terminal 12 / Digital Output 0	
Terminal 13 / Digital Output 1	
Type	Positive logic when active
Output Voltage	0 / 24 V
Nominal max. output	20 mA

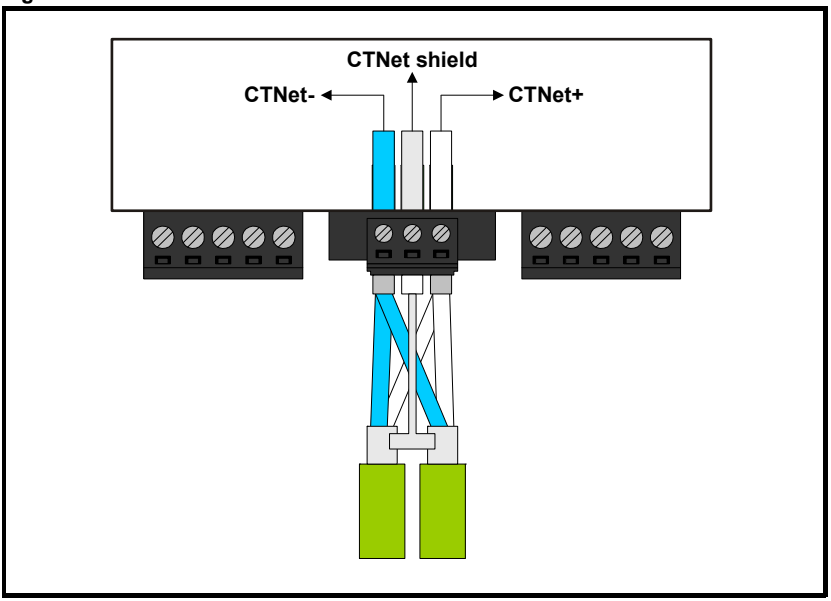
NOTE When inactive they are effectively floating.

3.3 CTNet connections

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 3-3 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.

In order to maintain the immunity levels of the SI-Applications Plus module, the following is recommended:-

- Fit a ferrite ring to each CTNet cable close to the module. One ferrite ring is included with the module. The part number for additional ferrite rings is 4200-7427.
- Connect a 40 mm length of wire between the 0V (terminal 9) on the SI-Applications Plus module, and 0V on the drive control terminals.

3.4 CTNet cable

CTNet cable has a single twisted pair plus overall shielding. 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.

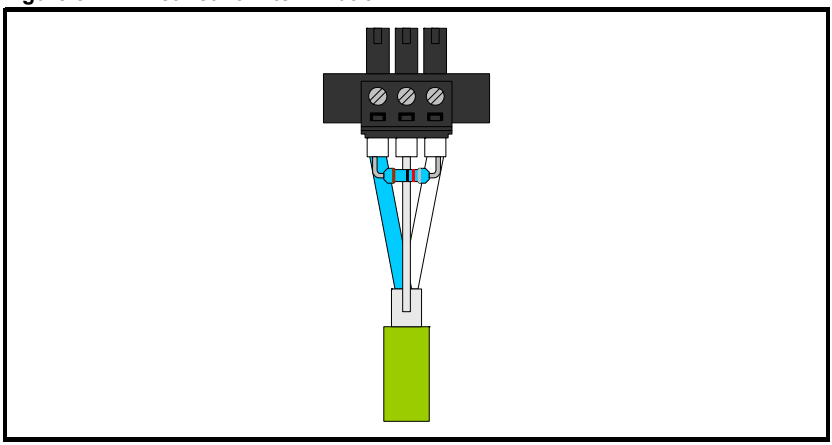
CTNet networks run at high data rates, and require cable specifically designed to carry high frequency signals. Low quality cable will attenuate the signals, and may render the signal unreadable for the other nodes on the network.The only approved CTNet cable is that supplied by Control Techniques/Leroy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

3.5 CTNet network termination

It is very important in high-speed communications networks that the network communications cable is installed 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 green CTNet cable, an 82 Ω, 0.25 W termination resistor should be installed across the CTNet+ and CTNet- data lines at BOTH ends of the cable run.

Figure 3-4 CTNet network termination



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

NOTE If too many termination resistors are installed on a CTNet network, the network will be over-loaded, 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.

3.5.1 CTNet cable shield connections

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 ground at one point. This ground connection is to prevent the cable shield from becoming live in the event of catastrophic failure of another device on the CTNet network or along the cable run.

3.5.2 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 and number of nodes. Refer to *CTNet User Guide*.

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.

3.6 EIA-RS485 connections

The EIA-RS485 port is available for lower-speed communications (up to 115,200 bit/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 6 *Communications* on page 58.

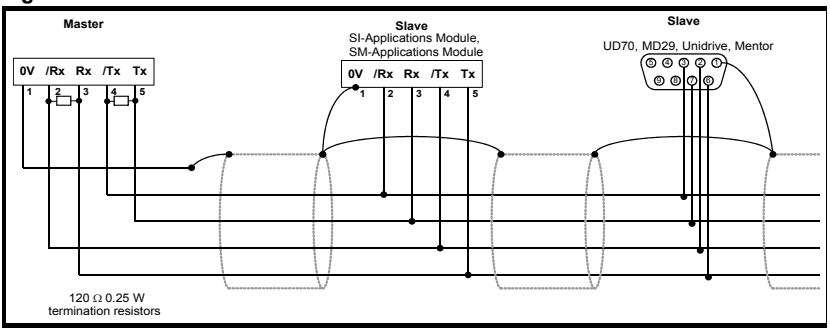
A host controller can operate up to thirty-two EIA RS485 devices with the use of line repeaters. The transmitter and receiver of each device loads the line by 2 unit loads. Therefore in two-wire mode, each device loads the line by 4 unit-loads. This means that no more than a total of seven devices can be connected in a single group, allowing up to 4 unit-loads for the line repeater. Up to 15 devices can be connected if four-wire mode is used.

3.6.1 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. SI-Applications Plus 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.

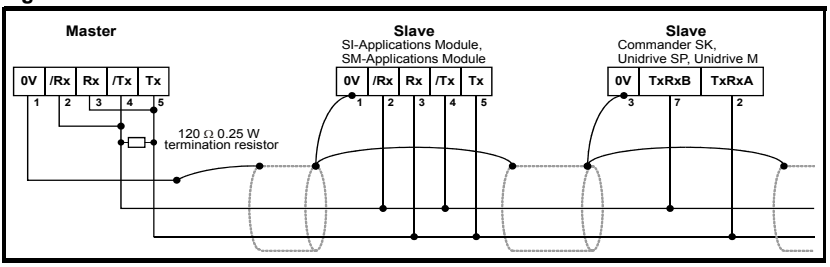
Figure 3-5 4 Wire EIA-RS485 network



3.6.2 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. SI-Applications Plus modules can be configured to act as master controllers, but this requires DPL programming to control the network.

Figure 3-6 2 Wire EIA-RS485 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.

3.6.3 Grounding

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

3.6.4 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 300 mm (1 ft) between the communications cable and the power cable.

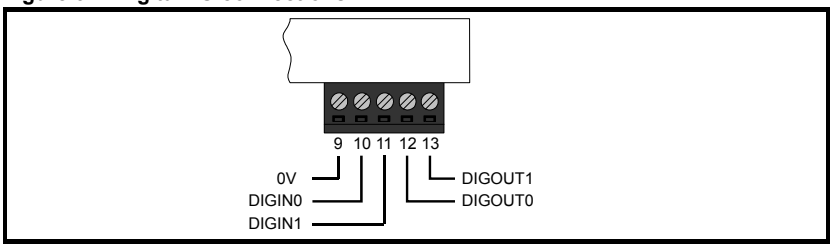
Cables crossing one another at right-angles are unlikely to give trouble. The maximum cable length for a EIA-RS485 jumper (link) is 1200 metres (4,000 ft). This is at low baud rates only. The higher the baud rate the lower the maximum cable length.

3.6.5 Termination

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

3.7 Digital I/O connections

Figure 3-7 Digital I/O connections



The SI-Applications Plus is 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 SI-Applications Plus modules .

The digital outputs are a positive logic arrangement such that they are at +24 V when active and will supply up to 20 mA of current. When inactive they are effectively floating. The digital outputs are protected against short-circuit or overload. The trip threshold is 20 mA, 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 41.

3.8 Port Isolation

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

WARNING

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.

WARNING

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.

4 Getting started

This chapter describes the basics of a user program using the SI-Applications Plus module and some of the aspects of using SyPTPro.

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

Throughout the remainder of this guide, when referring to a specific parameter for any slot the format Pr **81.XXX** will be used. e.g. The *Autorun* parameter will be referred to as Pr **81.013**.

This is also an aid to portability, as SI-Applications Plus modules with code using menu 81 can be installed to any slot, and the code should run as normal.

When the SI-Applications Plus module is installed, the module identification parameter Pr **81.001** will show the following...

SI-Applications Plus
304

The combination of parameters Pr **81.002** and Pr **81.051** provides the firmware version of the module.

NOTE SI-Applications Plus module specific parameters referred to in this guide Menu 70 to 79, 81, 85, 86, 88, 90 and 91 all use the Unidrive M parameter access mechanism, **MM.PPP**. The module is backwards compatible with the Unidrive SP access mechanism, **MM.PP**, This gives legacy compatibility with previously written software

4.1 Using SyPTPro

SyPTPro provides the development platform for the SI-Applications Plus module. This is detailed within the SyPTPro Help file.

4.2 Connecting the PC to the SI-Applications Plus

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

4.2.1 CTNet

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. PCI and USB cards are available for desktop and laptop computers.

Refer to section 3.3 *CTNet connections* on page 13 for details of the CTNet connections on the SI-Applications Plus module.

4.2.2 EIA-RS485 Serial port

You can connect the PC to the RJ45 serial port on the front of the drive. Special pre-made leads are available 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/Leroy Somer products that use a RJ45 EIA-RS485 connector such as the Unidrive SP, Commander SE and Commander SK.

Please refer to your drive documentation for the positioning and pin-out descriptions of the RJ45 connector.

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 SI-Applications Plus module installed these will be seen by SyPT Pro along with any other SI-Applications Plus module connected to this over CTNet. This is part of the routing capability of the CTNetAPI. See section 4.3 *CTNetAPI routing* on page 19.

4.3 CTNetAPI routing

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

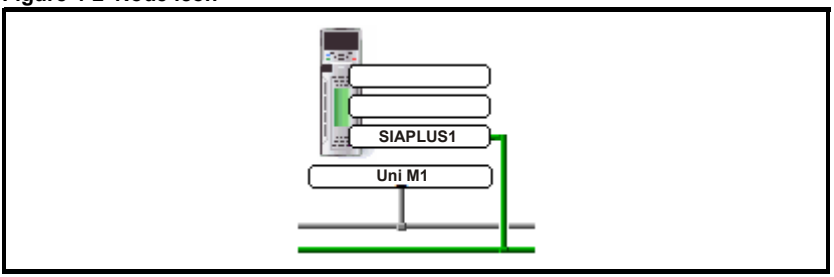
4.4 Configuring communications within SyPTPro

Before attempting to go *on-line* to the SI-Applications Plus 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, 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 parameter Pr **11.025** is set to **19200** (this is the default value).
4. Press OK.

4.5 Creating a Node in SyPTPro

Figure 4-2 Node icon



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 System Integration Modules can be entered using the tabs at the top of the box.
3. Press OK.

4.6 DPL Programming basics

The SI-Applications Plus 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*.

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 8 ms or when the module first powers-up. Each task has a particular name, purpose and priority. Refer to section 7.2 *Tasks* on page 67 for further information.

4.6.1 Function Block Library

SyPTPro comes 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).

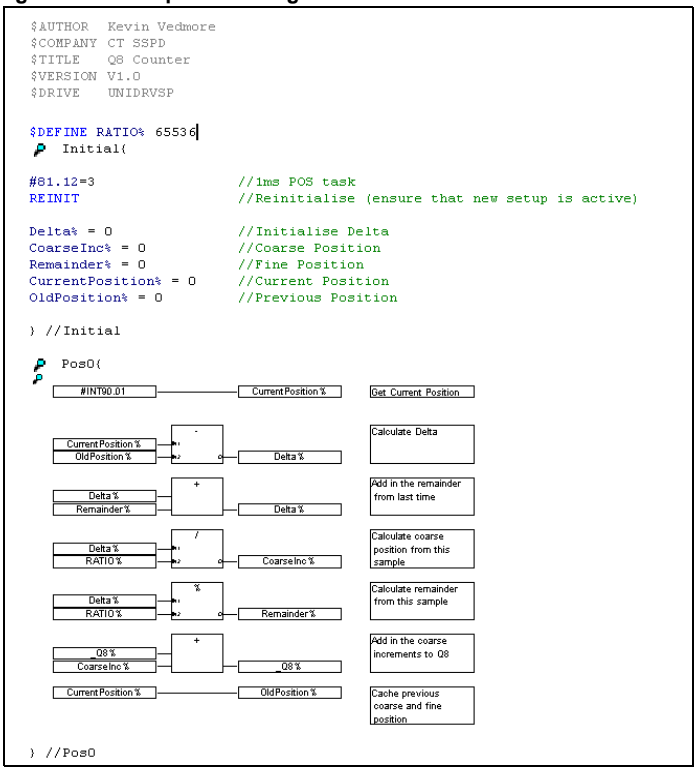
The functions in the FBL are documented in the on-line help.

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 7.7 *User Defined Function Blocks* on page 78 and the on-line help for information.

4.7 Program Example

Figure 4-3 is an example of a DPL program written within SyPTPro:

Figure 4-3 Example DPL Program



This program will take the positional feedback information from the drive (which is scaled to 2^{32}/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 to Unidrive M as 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

4.7.1 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.

4.7.2 Initial Task

As explained later in section 7.2 *Tasks* on page 67, this is a *task* which is executed when the SI-Applications Plus 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 initialize some integer variables (denoted by a trailing % symbol) to zero.

4.7.3 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 79. More information on these can be found in section 5.4 *Menus 70-79 - 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 **81.012** must be set to a non-zero value in the Initial task. After setting this, a REINIT command must be issued (see below).

```
#81.012 = 3 //Pos task schedule period 1 ms
REINIT     //Reinitialize
```

4.8 Downloading programs

By default, programs can only be downloaded to the SI-Applications Plus module when the drive enable signal is not active (Pr **06.015**=0). This behavior can be disabled by setting Pr **81.037** to 0.

5 Parameters

5.1 Overview

The SI-Applications Plus module contains two parameter databases:

- The drive database
  This contains the entire drive parameter set. The 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 "Waiting For Options" will appear for a few seconds on the drive display. This will not occur again unless the module is moved to a different drive with different firmware or the drive firmware is updated.
- The SI-Applications Plus 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 SI-Applications Plus parameters

The parameters that are saved to the SI-Applications Plus module when performing the actions shown below are:

- Menus 70, 71 and 74 to 79 (equivalent to P, Q, T, U, V, W, X and Y register sets)
- Menu 20

NOTE Menu 90 and 91 parameters are not persistent over a reset or power cycle and cannot be saved.

To save the parameters on demand:

- Set Pr **81.019** to 1 (**Save Request)**
  Pr **81.019** will reset to zero automatically and the module and drive will be reset.

To save the parameters on Under Voltage (UU):

- Set Pr **81.020** to 1 (**Enable "UU trip" Save)**

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 **81.021 (Enable menu 20 save and restore)** 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 **81.021 (Enable menu 20 save and restore)** needs to be at 1 at power-up therefore a drive parameter save is required. See section 5.2.3 *Saving drive parameters*.

5.2.3 Saving drive parameters

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

- Menus 1 through 14, 18, 19, 21, 22, 23 and 29 through 41.
- Menus 15, 16 and 17, if a module is present in the relevant slot.

To save the drive parameters:

1. Set Pr **mm.000**=1000 (parameter zero in any menu when using drive keypad)
2. Set Pr **10.038**=100 (simulates pressing the reset button on the drive keypad)

NOTE A drive save can be performed by entering a value of 1000 in any Pr **mm.000** menu and pressing the reset button. For drives that support a 24 V supply 1001 should be used while running on 24 V.

5.3 Menu 81 - Configuration parameters

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

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 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 SI-Applications Plus module is installed in.

Unless otherwise indicated, these parameters are only read when the SI-Applications Plus 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 module from the drive display, enter the value of 1070 in parameter zero of any menu and press the reset button.

NOTE Throughout this *User Guide*, the configuration parameters will be referred to as Pr **81.XXX**. 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.
“Initialization” 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, but menus 15 to 20 and 24 to 28 will be left unchanged.

5.3.1 Parameter Descriptions

Pr 81.001	Module Code		
Access	RO	Range	0 to 499
Default	N/A	Update Rate	N/A

Shows the module ID code. For SI-Applications Plus this is 304.

Pr 81.002	Firmware Version		
Access	RO	Range	00.00 to 99.99.99
Default	N/A	Update Rate	N/A

Specifies the major revision number of the operating system of the module. Use in conjunction with Pr **81.051** to form the complete version number.

Pr 81.003	DPL Program Status		
Access	RO	Range	0 to 3
Default	0	Update Rate	1 ms of change

Provides the run status of the user DPL program in the SI-Applications Plus 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

Pr 81.004	Available System Resource		
Access	RO	Range	0 to 100
Default	N/A	Update Rate	200 ms

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

Pr 81.005	EIA-RS485 Address		
Access	RW	Range	0 to 255
Default	11	Update Rate	Initialization

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).

Pr 81.006	EIA-RS485 Mode		
Access	RW	**Range**	0 to 255
Default	1	**Update Rate**	Initialization

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 - Pr 81.006*

Table 5-1 Serial modes - Pr 81.006

Mode	Description
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.

Table 5-1 Serial modes - Pr 81.006

Mode	Description
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
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 20: 1 start bit, 7 data bits, NO parity, 2 stop bits Mode 50: 1 start bit, 7 data bits, EVEN parity, 1 stop bit Mode 80: 1 start bit, 7 data bits, ODD parity, 1 stop bit
25	CT-Sync Master The baud rate is fixed at 896875bps
26	CT-Sync Slave The baud rate is fixed at 896875 bps

Pr 81.007	EIA-RS485 Baud Rate		
Access	RW	**Range**	0-9 (300-115200 bps)
Default	4 (4800)	**Update Rate**	Initialization

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

NOTE

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

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

Pr 81.008	EIA-RS485 Turn-around Delay		
Access	RW	**Range**	0 to 255 ms
Default	2 ms	**Update Rate**	Initialization

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 1 ms delay and this parameter can be used to extend it.

Pr 81.009	EIA-RS485 Tx Enable Delay		
Access	RW	**Range**	0 to 1 ms
Default	0 ms	**Update Rate**	Initialization

This parameter allows a 1 ms delay to be introduced between the 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.

Pr 81.010	DPL Print Routing		
Access	RW	**Range**	0/1
Default	0	**Update Rate**	Initialization

This parameter controls where the output of the DPL PRINT command is sent. If set to zero (Off), the output is sent to the programming client (SyPTPro) and if set to 1 (On) it will be sent to the EIA-RS485 port.

Pr 81.011	Clock Task Scheduling (ms)		
Access	RW	**Range**	0 to 200 ms
Default	10 ms	**Update Rate**	Initialization

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

Pr 81.012	POS task scheduling rate		
Access	RW	**Range**	0 to 6
Default	0	**Update Rate**	Initialization

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
Disable	0	Disabled
0.25	1	250 µs
0.5	2	500 µs
1	3	1 ms
2	4	2 ms
4	5	4 ms
8	6	8 ms

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.

Pr 81.013	Auto-run Enable		
Access	RW	**Range**	0/1
Default	1	**Update Rate**	Initialization

Setting this parameter to 1 will cause the program in the module to automatically execute whenever the drive is powered up. If set to a zero, a run command must be issued via the programming software for the program to run.

Pr 81.014	Global Run-time Trip Enable		
Access	RW	**Range**	0/1
Default	0	**Update Rate**	Initialization

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

For more information, see section 10.1 *Run-time errors* on page 92.

Pr 81.015	Disable Reset on Trip Cleared		
Access	RW	**Range**	0/1
Default	0	**Update Rate**	Initialization

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).

Pr 81.016	Encoder Data Update Rate		
Access	RW	**Range**	0-3
Default	0	**Update Rate**	Initialization

Display	Description
0	APC data and menu 90 encoder parameters are updated every 250 µs.
1	APC data and menu 90 encoder parameters are updated immediately prior to every POS task.
2	APC data and menu 90 encoder parameters are updated immediately prior to every CLOCK task.
3	APC data and menu 90 encoder parameters are never updated. If these are never updated, more processor resource will become free.

Pr 81.017	Enable Parameter Over-range Trips		
Access	RW	Range	0/1
Default	0	Update Rate	Initialization

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.

Pr 81.018	Watchdog Enable		
Access	RW	Range	0/1
Default	0	Update Rate	Initialization

When set it enables the DPL program watchdog function. The DPL WDOG command must then be executed every 200 ms. This can be used to protect the program against malfunction. If the command is not executed within a 200 ms time period the drive will trip on **Slotx Watchdog**. 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.

Pr 81.019	Save Request		
Access	RW	Range	0/1
Default	0	Update Rate	100 ms

Setting this parameter to 1 will initiate a save of all non-volatile SI-Applications Plus module data. This save may take up to approximately 100 ms before it is actioned. This consists of the menu 70, 71 and 74 to 79 PLC register sets and optionally menu 20 (depending upon the setting of Pr **81.021**).

NOTE This will also cause a reset of the module and this parameter will revert to zero automatically. Also if the drive is tripped it will be reset. Menu 81 will not be saved.

Pr 81.020	Enable “UU trip” Save		
Access	RW	Range	0/1
Default	0	Update Rate	Immediate

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

NOTE Note that when a ‘UU’ save occurs the module will be reset.

Pr 81.021	Enable menu 20 save and restore		
Access	RW	Range	0/1
Default	0	Update Rate	Immediate

If set to 1, menu 20 will be saved/restored along with other non-volatile parameters upon a save request (Pr **81.019**=1) or power-down save (Pr **81.020**=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 module installed to the drive should be used to store and restore menu 20, therefore if more than one module is installed to the drive **only one** should have this parameter set otherwise menu 20 will not be restored correctly on power-up.

NOTE Unlike other set-up parameters, parameters Pr **81.020** and Pr **82.021** are **not cached**, which means a change to the parameter takes immediate effect.

Pr 81.022	CTNet Token Ring ID		
Access	RW	Range	0 to 255
Default	0	Update Rate	Initialization

This parameter allows the user to specify the identity of the CTNet token ring to which a SI-Applications Plus 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 IDs should be set for each ring. The combination of CTNet Token Ring ID and CTNet node address should be unique.

Pr 81.023	CTNet Node Address		
Access	RW	Range	0 to 255
Default	0	Update Rate	Initialization

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.

Pr 81.024	CTNet Baud Rate		
Access	RW	Range	0 to 3
Default	1(2.5)	Update Rate	Initialization

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	5 Mbs
2.500	1	2.5 Mbs
1.250	2	1.25 Mbs
0.625	3	625 kbs

Pr 81.025	CTNet Sync Setup		
Access	RW	Range	0 to 9999
Default	0	Update Rate	Initialization

Specifies the synchronization 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 10 ms and slow every 150 ms. When using easy mode (see below) it is only necessary to set up the FF (fast cyclic rate).

Pr 81.026, Pr 81.028, Pr 81.030	CTNet Easy Mode Setup Parameters		
Access	RW	**Range**	0 to 25503
Default	0	**Update Rate**	Initialization

Pr 81.027, Pr 81.029, Pr 81.031 - Pr 81.034	CTNet Easy Mode Setup Parameters		
Access	RW	**Range**	0 to 9999
Default	0	**Update Rate**	Initialization

NOTE Easy mode cyclic link parameter source and destinations do not accept the **MM.PPP** style mappings for Unidrive M, only **MM.PP**. Where Pr **1.021** is to be accessed remove the first 0 after the decimal point to give a reference of 121, where 121 could be entered in Pr **81.027**, Pr **81.029**, Pr **81.031** to Pr **81.034**.

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

Parameter	Format	Channel	Description
Pr **81.026**	NNNSS	1	Defines the destination node number and slot NNN = Node number (0-255) SS = Slot number (1-3) e.g. A value of 201 means node ID 2, slot 1.
Pr **81.027**	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 Pr **03.002** (speed)
Pr **81.028**	NNNSS	2	Destination node number and slot for channel 2
Pr **81.029**	MMPP	2	Source drive parameter for channel 2
Pr **81.030**	NNNSS	3	Destination node number and slot for channel 3
Pr **81.031**	MMPP	3	Source drive parameter for channel 3
Pr **81.032**	MMPP	1	Slot 1 destination parameter for incoming data
Pr **81.033**	MMPP	2	Slot 2 destination parameter for incoming data
Pr **81.034**	MMPP	3	Slot 3 destination parameter for incoming data

Pr 81.035	CTNet Sync. Event Task ID		
Access	RW	Range	0 to 4
Default	0	Update Rate	Initialization

Identifies which of the EVENT tasks will be scheduled when a CTNet synchronization message is received or generated. Synchronization 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

Pr 81.036	CTNet Diagnostics		
Access	RO	Range	-3 to 32767
Default	N/A	Update Rate	1 s

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

Pr 81.036	Status	Description
>0	Network OK	Indicates the number of messages per second being processed every second.
0	Network OK, 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	Initialization Error	The SI-Applications Plus module was unable to configure the CTNet interface. Check that the node address and data rate are set correctly.
-3	MYRECON	The SI-Applications Plus module forced a CTNet network re-configuration

Pr 81.037	Reject Download if Drive Enabled		
Access	RW	Range	0/1
Default	0	Update Rate	Initialization

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 **81.014**) 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.

Pr 81.038	APC Run-time trip		
Access	RW	Range	0/1
Default	0	Update Rate	Initialization

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 unitialized CAM function. When this parameter is 1 the drive will not trip when an APC non-recoverable error occurs.

Pr 81.039	Inter-module Drive Sync Status		
Access	RW	Range	0/1
Default	0	Update Rate	NA

This parameter displays the current module's synchronization status.

Synchronization	Status
0	The synchronization master request is zero or another System Integration Module is synchronization master.
1	The System Integration Module is synchronization master.
3	The System Integration Module is synchronization master, but the synchronization frequency is out of specification or not present.

Pr 81.041	Indexer Control		
Access	RW	Range	0/3
Default	0	Update Rate	NA

This is used to control the motion sequence user program..

Value	Status
0	Run
1	Stop
2	Pause
3	Step

Pr 81.042	Pass Freeze Through to Drive		
Access	RW	Range	0/1
Default	0	Update Rate	Initialization

When this parameter is ON (1), the voltage on the module digital input 0 (DIGINO) is passed through to the drive's internal common Freeze line. This can be seen by other classes of System Integration Modules. When Pr **81.042** is set to 1, then at power up and on REINIT Pr **03.100** and Pr **03.105** will be set to 4. This sets the common freeze line as the F1 and F2 freeze trigger sources on the drive. For further information on the Freeze Input refer to Chapter 8 *Freeze and marker* on page 81.

Pr 81.043	Freeze Invert		
Access	RW	Range	0/1
Default	0	Update Rate	Initialization

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 81.

Pr 81.044	Task Priority Level		
Access	RW	Range	0 to 255
Default	0	Update Rate	Initialization

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	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	0	Reserved
	1	
2	0	Turbo CTNet Disabled
	1	Turbo CTNet Enabled

Pr 81.045	User Set-Up Parameter 1		
Access	RO	Range	N/A
Default	0	Update Rate	N/A

This parameter is dependent on what is running in the module i.e. Indexer.

Pr 81.046	User Set-Up Parameter 2		
Access	RO	Range	N/A
Default	0	Update Rate	N/A

This parameter is dependent on what is running in the module i.e. Indexer.

Pr 81.047	User Set-Up Parameter 3		
Access	RO	Range	N/A
Default	0	Update Rate	N/A

This parameter is dependent on what is running in the module i.e. Indexer.

Pr 81.048	Line Number of Error		
Access	RO	Range	32 bit
Default	0	Update Rate	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).

Pr 81.049	User program ID		
Access	RO/RW	Range	Signed 16-bit
Default	0	Update Rate	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.

Pr 81.050	Run-time Error Code		
Access	RO	Range	0 to 255
Default	0	Update Rate	On error

When a run-time error occurs the error number is placed into this parameter. See Chapter 10.1 *Run-time errors* on page 92 for further information.

Pr 81.051	Firmware - Minor Version		
Access	RO	Range	0 to 99
Default	N/A	Update Rate	N/A

Specifies the minor revision number of the operating system of the SI-Applications Plus module. Use in conjunction with Pr **81.002** to form the complete version number.

5.4 Menus 70-79 - 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 10 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]		
72	RW	_Rx%, _R%[x]		Used for outgoing CTNet cyclic data links. Non-saveable.
73	RW	_Sx%, _S%[x]		Used to incoming CTNet cyclic data links. Non-saveable
74	RW	_Tx%, _T%[x]		General purpose. Saveable.
75	RW	_Ux%, _U%[x]		
76	RW	_Vx%, _V%[x]		
77	RW	_Wx%, _W%[x]		
78	RW	_Xx%, _X%[x]		
79	RW	_Yx%, _Y%[x]		

You will see from the above table that each parameter within menus 70 to 79 has and equivalent DPL variable. This means that you can use either format for accessing a parameter within these menus.

e.g. Pr **72.001**=1 will do the same as _R01%=1, Pr **75.065**=66 will do the same as _U65%=66 etc.

Menus 70, 71, and 74 to 79 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 23 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.

Pr **72.079** and Pr **73.079** will be reserved if AUTOSYNC is in use.

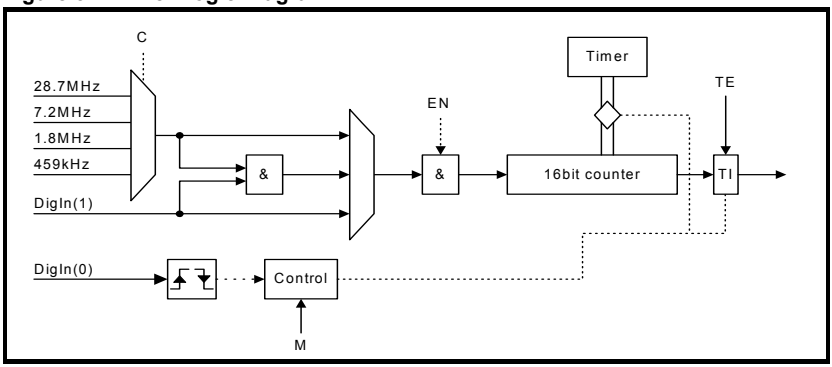
Parameters Pr **71.080** thru Pr **71.099** may be used for the RAM File Recording

5.5 Menu 85 - Timer Function Parameters

A hardware/counter is built into the 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 600 kHz.
- 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 Timer Logic Diagram



Pr 85.001	Timer Unit Control Word		
Access	RW	Range	13 bit
Default	N/A	Update Rate	Immediate

Table 5-2 Control word - Pr 85.001

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 (28.7 MHz) 1=rate /4 2=rate /16 3=rate /64
b7-b8	M	**Timer mode:** **0=Free Running mode** The selected clock drives the counter. The TI flag is set on wrap-around. **1=Capture mode 1** 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). **2=Capture mode 2** 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).

Pr 85.002	Timer Unit Status Word		
Access	RO	Range	0 to 3
Default	N/A	Update Rate	Immediate

Table 5-3 Status Word - Pr 85.002

Bit	Symbol	Function
b0	TI	**Timer event flag:** 0=No event has occurred 1=An event has occurred (see description for Pr **85.001**) **Note:** The TI bit is automatically cleared by the operating system if TE flag in Pr **85.001** has a non-zero value. Otherwise it is cleared when the status word is read.
b1	OV	**Wrap-around flag:** 0=Wrap-around has not occurred 1=Counter wrap-around has occurred **Note:** This flag is valid for ALL timer modes and is automatically cleared when the status register is read.

Pr 85.003	Timer Unit 16-bit Timer Count		
Access	RW	Range	16 bit
Default	N/A	Update Rate	Immediate

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

Pr 85.004	Timer Unit Wrap-around limit		
Access	RW	Range	16 bit
Default	N/A	Update Rate	Immediate

This parameter specifies the value at which Pr **85.003** will wrap-around.

Setting a small wrap-around figure with the timer operating at fast rates, could result in the 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 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 **81.013**=0). After the reset the user can re-initialize the timer with a more reasonable wrap-around number.

Pr 85.005	Timer Unit Timer Capture Cache		
Access	RW	Range	16 bit
Default	N/A	Update Rate	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 SI-Applications Plus module has two digital outputs and two digital inputs on-board. These outputs and inputs are controlled though this menu. See Chapter 3 *Installation* on page 11 for detailed specifications.

Pr 86.001	Digital Input 0		
Access	RO	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

Pr 86.002	Digital Input 1		
Access	RO	**Range**	0/1
Default	N/A	**Update Rate**	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.

Pr 86.003	Digital Output 0		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

Pr 86.004	Digital Output 1		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

Pr **86.003** & Pr **86.004** control digital outputs 0 and 1. Setting to 0 will place the output low and setting to 1 will place the input high (+24 V).

Pr 86.005	Digital Outputs 0 and 1		
Access	RW	**Range**	0 to 3
Default	N/A	**Update Rate**	Immediate

This parameter provides control for both digital outputs and is an alternative to controlling each output individually with Pr **86.003** and Pr **86.004**. Bit0 of this parameter controls digital output 0 (Pr **86.003**) and Bit1 controls digital output 1 (Pr **86.004**).

5.7 Menu 88 - Status Parameters

Pr 88.001	Error code / Reset		
Access	RW	**Range**	0 to 9999
Default	N/A	**Update Rate**	On error

This parameter has two purposes - when read it will return the identical run-time error as Pr **81.050** (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 SI-Applications Plus 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 **81.013**=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.

Pr 88.002	Task in Error		
Access	RO	**Range**	0 to 50
Default	N/A	**Update Rate**	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
11	APC

A value of zero will be returned if there is no error condition. For more information on these parameters refer to Chapter 10 *Diagnostics* on page 92.

Pr 88.003	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, Pr **88.004** and Pr **88.005** become active. If set to zero, Pr **88.004** and Pr **88.005** will read zero.

Pr 88.004	Free Resource for Motion Engine Tasks		
Access	RO	Range	0 to 95
Default	N/A	Update Rate	See Pr **81.012**

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 88.005	Motion Engine Peak Resource Detect		
Access	RO	Range	0 to 95
Default	N/A	Update Rate	See Pr **88.004**

This parameter displays the lowest value (i.e. highest resource usage) that Pr **88.004** reaches since the monitoring was enabled (Pr **88.003**). 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 88.006	CLOCK Task Resource Monitoring		
Access	RW	Range	0/1
Default	0	Update Rate	Immediate

This parameter allows the user to enable or disable monitoring of the CLOCK task free resource. When set to 1, Pr **88.007** and Pr **88.008** become active. If set to zero, Pr **88.007** and Pr **88.008** will read zero.

Pr 88.007	Free Resource for Clock Task		
Access	RO	Range	0 to 95
Default	NA	Update Rate	See Pr **81.011**

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.

Pr 88.008	Clock Task Peak Resource Detect		
Access	RO	Range	0 to 95
Default	NA	Update Rate	See Pr **81.011**

This parameter displays the lowest value (i.e. highest resource usage) that Pr **88.007** reaches since the monitoring was enabled (Pr **88.006**). 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.

Pr 90.001	Feedback Encoder Position (2^{32}/rev)		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

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 90.002	Feedback Encoder Revolution Count		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	See Pr **81.016**

Contains the feedback encoder revolution count.

Pr 90.003	Reference Encoder Position (2^{32}/rev)		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

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 90.004	Reference Encoder Revolution Count		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	See Pr **81.016**

Contains the reference encoder revolution count.

Pr 90.010	Drive Mode		
Access	RO	**Range**	Signed 16-bit
Default	N/A	**Update Rate**	Immediate

Provides a definitive method of identifying the mode the drive is in. It is recommended that this parameter is used instead of Pr **11.031** or Pr **00.048** 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

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

Pr 90.011	Drive Status and Control Word		
Access	RW	**Range**	Signed 16-bit
Default	N/A	**Update Rate**	Immediate

Writing to this parameter updates the control word. Reading from this parameter reads the status word (same as Pr **10.040**).

Table 5-4 Control Word

Bit	Description
b15	If set, the value of Pr **01.046** is set from b6
b14	If set, the value of Pr **01.045** is set from b5
b13	Sets the value of Pr **18.033** (Application menu 1, bit 3)
b12	If set, the value of Pr **06.032** is set from b3
b11	If set, the value of Pr **06.031** is set from b2
b10	If set, the value of Pr **06.030** is set from b1
b9	If set, the value of Pr **06.015** is set from b0
b8	Sets the value of Pr **18.032** (Application menu 1, bit 2)
b7	Sets the value of Pr **18.031** (Application menu 1, bit 1)
b6	Sets the value of Pr **01.046** (Preset select bit 1)
b5	Sets the value of Pr **01.045** (Preset select bit 0)
b4	User trip. Trips drive immediately if set.
b3	Sets the value of Pr **06.032** (Sequencing bit 2: Run Reverse)
b2	Sets the value of Pr **06.031** (Sequencing bit 1: Jog)
b1	Sets the value of Pr **06.030** (Sequencing bit 0: Run forward)
b0	Sets the value of Pr **06.015** (Drive enable)

Table 5-5 Status Word

Bit	Description
b15	Not used
b14	Pr **10.015** (Mains Loss)
b13	Pr **10.014** (Direction running)
b12	Pr **10.013** (Direction commanded)
b11	Pr **10.012** (Braking resistor alarm)
b10	Pr **10.011** (Braking IGBT active)
b9	Pr **10.010** (Regenerating)
b8	Pr **10.009** (Drive output is at current limit)
b7	Pr **10.008** (Load reached)
b6	Pr **10.007** (Above set speed)
b5	Pr **10.006** (At speed)
b4	Pr **10.005** (Below set speed)
b3	Pr **10.004** (Running at or below min speed)
b2	Pr **10.003** (Zero speed)
b1	Pr **10.002** (Drive running)
b0	Pr **10.001** (Drive OK)

Pr 90.012	Event task schedule reason		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	On Event

For a description see Pr **90.015**.

Pr 90.013	Event1 task schedule reason		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	On Event1

For description see Pr **90.015**.

Pr 90.014	Event2 task schedule reason		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	On Event2

For description see Pr **90.015**.

Pr 90.015	Event3 task schedule reason		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	On Event3

The four parameters above (Pr **90.012** - Pr **90.015**) 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 or Embedded
2-7	Reason for trigger	0-31 = Other System Integration Module initiated 32 = CTNet Sync 33 = Timer Unit 34-63 = User-defined reason via the DPL command SCHEDULEEVENT.

Pr 90.018	Feedback Encoder Freeze Flag		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	250 µs

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.

Pr 90.019	Feedback Encoder Freeze Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	250 µs

For description see Pr **90.020**

Pr 90.020	Feedback Encoder Freeze Turns		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	250 µs

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

Pr 90.024	Module slot number		
Access	RO	**Range**	Unsigned 8-bit
Default	N/A	**Update Rate**	Initialization

This parameter reports the slot number into which the module is installed. This parameter will reflect any changes made to the option slot menu assignments in Pr **11.056** on the drive.

Pr 90.025	Feedback encoder marker position (2^{32}/rev)		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

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

Pr 90.026	Feedback encoder marker turns (2^{16}/rev)		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	See Pr **81.016**

This parameter gives the feedback encoder marker revolution count.

Pr 90.027	Second Processor database version number		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	Initialization

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

Pr 90.028	Reference Encoder Freeze flag		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	250 µs

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.

Pr 90.029	Reference Encoder Freeze Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	250 µs

See Pr **90.030** for description.

Pr 90.030	Reference Encoder Freeze Turns		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	250 µs

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

Pr 90.031	Feedback Encoder Turns and Coarse Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

See Pr **90.32** for description.

Pr 90.032	Reference Encoder Turns and Coarse Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

These 2 parameters (Pr **90.031** & Pr **90.032**) store the 16-bit turns in the upper word and 16-bit position in the lower word, of the feedback (Pr **90.031**) and reference (Pr **90.032**) encoders.

Pr 90.033	Feedback Encoder Freeze Turns and Coarse Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	250 µs

See Pr **90.034** for description.

Pr 90.034	Reference Encoder Freeze Turns and Coarse Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	250 µs

These 2 parameters (Pr **90.033** & Pr **90.034**) 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.

Pr 90.035	Reference Encoder Marker Position (2^{32}/rev)		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

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

Pr 90.036	Reference Encoder Marker turns (2^{16}/rev)		
Access	RO	**Range**	Unsigned 16-bit
Default	N/A	**Update Rate**	See Pr **81.016**

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

Pr 90.037	Feedback Encoder Marker Turns and Coarse Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

See Pr **90.038** for description.

Pr 90.38	Reference Encoder Marker Turns and Coarse Position		
Access	RO	**Range**	Signed 32-bit
Default	N/A	**Update Rate**	See Pr **81.016**

These 2 parameters (Pr **90.037**,Pr **90.038**) store the 16-bit turns in the upper word and 16-bit position in the lower word, of the feedback (Pr **90.037**) and reference (Pr **90.038**) encoders at the time the marker pulse was activated.

Pr 90.039	Drive Keypad Button Status		
Access	RO	**Range**	Signed 16-bit
Default	N/A	**Update Rate**	> 40 ms

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

Value	Description
b0	Auxiliary
b1	Run
b2	Stop

Pr 90.040	Event Task Trigger		
Access	RW	**Range**	Unsigned 16-bit
Default	0	**Update Rate**	Immediate

Upon setting this parameter to a value it will execute one of the 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

Pr 90.041	Reference Encoder Marker Flag		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	See Pr **81.016**

See Pr **90.042** for description.

Pr 90.042	Feedback Encoder Marker Flag		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	See Pr **81.016**

These 2 parameters (Pr **90.041** & Pr **90.042**) 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 **90.045** and Pr **90.046**). To re-arm the marker these parameters must be set to zero by the user. They cannot be set to 1 by the user.

Pr 90.043	Reference Encoder Source		
Access	RW	**Range**	Unsigned 8-bit
Default	N/A	**Update Rate**	Immediate

See Pr **90.044** for description.

Pr 90.044	Feedback Encoder Source		
Access	RW	**Range**	Unsigned 8-bit
Default	N/A	**Update Rate**	Immediate

Pr **90.043** and Pr **90.044** define the source for the reference and feedback data. See the table below for the valid sources and the configuration of the freeze functions.

Value	Description	Freeze functionality
0	P1 Drive encoder	Freeze operates on drive F1 freeze function
1	P1 Slot 1	Freeze operates on Slot 1 F1 freeze function
2	P1 Slot 2	Freeze operates on Slot 2 F1 freeze function
3	P1 Slot 3	Freeze operates on Slot 3 F1 freeze function
4	User program	
5	Un-configured	
6	P2 Drive encoder	Freeze operates on drive F2 freeze function
7	P2 Slot 1	Freeze operates on Slot 1 F2 freeze function
8	P2 Slot 2	Freeze operates on Slot 2 F2 freeze function
9	P2 Slot 3	Freeze operates on Slot 3 F2 freeze function
10	P1 Slot 4	Freeze operates on Slot 4 F1 freeze function
11	P2 Slot 4	Freeze operates on Slot 4 F2 freeze function

When a P1 interface (drive or option module) is selected as the source for Pr **90.043** or Pr **90.044**, the position source for the F1 freeze function will be set to P1 (in the case of the drive Pr **03.102** will be set to P1).

Firmware V02.10.00 and later

When a P2 interface (drive or option module) is selected as the source for Pr **90.043** or Pr **90.044**, the position source for the F2 freeze function will be set to P2 (in the case of the drive Pr **03.107** will be set to P2).

Firmware prior to V02.10.00

When a P2 interface (drive or option module) is selected as the source for Pr **90.043** or Pr **90.044**, the position source for the F2 freeze function is not automatically set by the module and must be set by the user otherwise the freeze position for the P2 source will not be captured correctly.

Pr 90.045	Reference Marker Flag Enable		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

See Pr **90.046** for description.

Pr 90.046	Feedback Marker Flag Enable		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

These 2 parameters (Pr **90.045** & Pr **90.046**) must be set to 1 to allow the marker flags (Pr **90.041** and Pr **90.042**) to be set when the marker pulse is activated.

Pr 90.047	Reference Freeze Enable		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

See Pr **90.048** for description.

Pr 90.048	Feedback Freeze Enable		
Access	RW	**Range**	0/1
Default	N/A	**Update Rate**	Immediate

These 2 parameters (Pr **90.047** & Pr **90.048**) must be set to 1 allow the freeze flags (parameters Pr **90.018** and Pr **90.028**) to be set when the freeze input is activated.

Pr 90.049	APC Runtime Error ID		
Access	RO	**Range**	32bit
Default	N/A	**Update Rate**	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 virtual parameters which provide a faster update rate or enhanced resolution than drive parameters.

Pr 91.001	Short-cut enable		
Access	RW	**Range**	Unsigned 8-bit
Default	0	**Update Rate**	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	Pr **91.002**
1	Hard-speed reference shortcut enable	Pr **91.003**
2	Reserved	
3		
4		
5		
6		
7		

Pr 91.002	Speed set-point (Pr 01.021)		
Access	RW	Range	Signed 32-bit
Default	N/A	Update Rate	250 µs

This parameter is available when SI-Applications Plus firmware V02.04.00 and later and drive firmware V01.11.00 and later are used.

Sets the speed reference in units of **0.001 rpm**. This value is mirrored in drive Pr **01.021** (preset speed 1), therefore in order to control the drive speed with this parameter ensure preset speed 1 is selected on the drive (Pr **01.014**=3, Pr **01.015**=1).

Ensure bit 0 of Pr **91.001** is set and the full-scale speed in Pr **91.005** is set accordingly when using this parameter.

Pr 91.003	Hard-speed reference (Pr 03.022)		
Access	RW	Range	Signed 32-bit
Default	N/A	Update Rate	250 µs

Controls the hard-speed reference on the drive in units of 0.001 rpm. This value is mirrored in drive Pr **03.022**, therefore in order to control the drive speed with this parameter ensure the hard speed reference is selected on the drive (Pr **03.023**=1).

Ensure bit 1 of Pr **91.001** is set and the full-scale speed in Pr **91.005** is set accordingly when using this parameter.

NOTE

This parameter is valid only in RFC-A and RFC-S modes only.

Pr 91.005	Full scale speed (rpm)		
Access	RW	Range	Signed 32-bit
Default	1500	Update Rate	N/A

Set this to the maximum (absolute) speed that will ever be written to with Pr **91.002** or Pr **91.003**. 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 **91.002** or Pr **91.003** greater than the rpm value specified in Pr **91.005** will result in the value being limited or a value over range run-time error.

When the drive is in RFC-A, in the event of a module reset (**81.019** = On), Pr **91.005** will default to 3000 rpm.

Pr 91.006	Speed feedback		
Access	RO	Range	Signed 32-bit
Default	N/A	Update Rate	250 µs

This parameter returns the value of the drive speed feedback in units of 0.01 rpm in RFC-A/S 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 **03.026** on the drive is set to a 0 to select the P1 interface or 6 to select the P2 interface on the drive. However, if a low resolution encoder is used there may be some jitter at low speed. For example, at 10 rpm with a 1024 ppr encoder this parameter may jump between 0 and 14.65 rpm. This is similar to the drive parameter Pr **03.002**.

Pr 91.007	Current feedback (Pr 04.002)		
Access	RO	Range	Signed 32-bit
Default	N/A	Update Rate	250 µs

This parameter returns the value of drive current feedback (Pr **04.002**) in units of 0.001 A. For example a value of 1500 in Pr **91.007** would indicate 1.5 A in Pr **04.002**.

Pr 91.008	Drive analog input 1 value		
Access	RO	Range	±4000
Default	N/A	Update Rate	250 µs

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

Pr 91.009	Drive analog Input 2 value		
Access	RO	Range	±2047
Default	N/A	Update Rate	250 µs

This value will be taken from the drive's analog input 2 and is scaled for ±2047 to represent the +/- full scale signal at the input. Refer to the drive documentation for information on the sampling rate of analog inputs.

Pr 91.010	Drive analog input 3 value		
Access	RO	Range	±2047
Default	N/A	Update Rate	250 µs

This value will be taken from the drive's analog input 3 and is scaled for ±2047 to represent the +/- full scale signal at the input. Refer to the drive documentation for information on the sampling rate of analog inputs.

Pr 91.016	Drive digital inputs		
Access	RO	Range	Unsigned 8-bit
Default	N/A	Update Rate	250 µs / 2 ms

This parameter is similar to Pr **08.072** on the drive and provides the status of 7 drive digital inputs in one single parameter. Logic polarity is taken into account. The bits are assigned as shown in the table below.
The update rate is either 250 µs or 2 ms depending on input as shown in the table below.

Bit	Digital Input	Update rate
0	F1	2 ms
1	F2	2 ms
2	F3	2 ms
3	F4	250 µs
4	F5	250 µs
5	F6	2 ms
6	Enable	2 ms
7	Reserved - Read as zero	N/A

Pr 91.017	Number of Valid CTSync Messages Received		
Access	RW	Range	Signed 32-bit
Default	N/A	Update Rate	

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

Pr 91.018	Number of Bad CTSync Messages Received		
Access	RW	Range	Signed 32-bit
Default	N/A	Update Rate	

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

Pr 91.019	Number of Missing CTSync Messages		
Access	RW	Range	Signed 32-bit
Default	N/A	Update Rate	

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

Pr 91.020	CTSync Synchronization Signal Width too Short		
Access	RW	Range	Signed 32-bit
Default	N/A	Update Rate	synchronization

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

Pr 91.021	Inter-option Synchronization Control		
Access	RW	Range	0 to 2
Default	0	Update Rate	Immediate

Reserved.

Pr 91.022	Inter-option synchronization Status		
Access	RO	Range	Unsigned 8-bit
Default	N/A	Update Rate	Immediate

Reserved.

Pr 91.023	Enable CTSync Output Channels		
Access	RW	Range	Unsigned 8-bit
Default	1	Update Rate	Immediate

Enable CTSync output channels on this module with motion engine even if the module is not in CTSync mode. (Note that one of POS0, APC, POS1 must be running for CTSync output channels to be enabled). This parameter has a default value of 1. Setting this parameter to zero can help free up resources if the CTSync output channels are not required.

5.10 Menu 97 - Internal Motion Processor Parameters

Pr **97.000** to Pr **97.099** are 32 bit parameters and are reserved for the internal motion processor. They can be read and written to from the user program, but are not accessible by the keypad.

5.11 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 SI-Applications Plus module (and via comms), but may be read-only on the drive's keypad.

Pr 1x.001	Integer, read-write saved on power-down		
Access	RW	**Range**	Signed 16-bit
Default	0	**Update Rate**	N/A

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

Pr 1x.002- Pr 1x.010	Integer, read-only		
Access	RW (RO drive)	**Range**	Signed 16-bit
Default	0	**Update Rate**	N/A

The SI-Applications Plus module has read/write access to Pr **1x.002** to Pr **1x.010**, but these parameters are read-only when viewed from the drive's keypad. These parameters are not scalable in the drive's non-volatile memory.

Pr 1x.011- Pr 1x.030	Integer, read-write		
Access	RW	**Range**	Signed 16-bit
Default	0	**Update Rate**	N/A

Pr 1x.031- Pr 1x.050	Bit, read-write		
Access	RW	**Range**	0/1
Default	0	**Update Rate**	N/A

Pr **1x.011** to Pr **1x.050** are saveable in the drive's non-volatile memory.

Pr 1x.051 – Pr 1x.054	Long integer, read-write saved on power-down		
Access	RW	**Range**	Signed 32-bit
Default	0	**Update Rate**	N/A

5.12 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 SI-Applications Plus module flash memory upon request. If more than one SI-Applications Plus module is installed, only one should be configured to store and restore this menu for obvious reasons.

If Pr **81.021** is set, this menu will be saved and restored by the module.

Pr 20.001-Pr 20.020	**Integer, read-write**		
Access	RW	**Range**	Signed 16-bit
Default	0	**Update Rate**	N/A

Pr 20.021 Pr 20.040	**Long integer, read-write**		
Access	RW	**Range**	Signed 32-bit
Default	0	**Update Rate**	N/A

6 Communications

6.1 EIA-RS485 serial communications port

SI-Applications Plus comes with a EIA-RS485 serial communications port on-board. Refer to Chapter 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 Pr **81.007**.

The address of this unit is specified in Pr **81.005**.

6.1.1 CT-ANSI

The SI-Applications Plus supports the ANSIx3.28 protocol.

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

6.1.1.1 Reading a parameter

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

Table 6-1 Master request

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

Table 6-2 Slave response (if request was correct and parameter exists)

Character	Description
STX	Start of text (Ctrl & B)
M1	Menu number: 1st digit
M2	Menu number: 2nd digit
P1	Parameter number: 1st digit
P2	Parameter number: 2nd digit
D1 to Dn	Data: 1st digit Data: nth 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. Checksum = M1 ^ M2 ^ P1 ^ P2 ^ D1 ^ D2 ^ ..... Dn ^ ETX. The checksum is an unsigned 8 bit value and if this value is less than 32 then 32 is added to it.

6.1.1.2 Writing to a parameter

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

Table 6-3 Master request

Character	Description
EOT	End of transmission (Ctrl & D)
A1	Module address: 1st digit
A1	Module address: 1st digit
A2	Module address: 2nd digit
A2	Module address: 2nd digit
STX	Start of text (Ctrl & B)
M1	Menu number: 1st digit
M2	Menu number: 2nd digit
P1	Parameter number: 1st digit
P2	Parameter number: 2nd digit
D1 to Dn	Data: 1st digit 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.
5. 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 ^ M2 ^ P1 ^ P2 ^ D1 ^ D2 ^ ..... Dn ^ ETX. The checksum is an unsigned 8bit value and if this value is less than 32 then 32 is added to it.

6.1.1.3 SI-Applications Plus address

The module will only act on messages received where the full address matches the address of the SI-Applications Plus or the group address in the message (1^{st} digit) matches the 1^{st} digit of the address of SI-Applications Plus or the address in the message is a broadcast (0). Broadcast messages are used to write data to multiple nodes.

6.1.1.4 Control characters

Table 6-4 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

6.1.2 Modbus-RTU

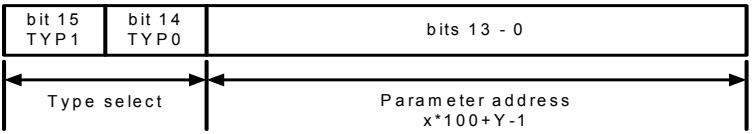
Both slave and master modes of the Modicon Modbus-RTU protocol are supported. In RTU slave mode, the following function codes are supported:

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 simultaneously is 20.

Drive parameters are mapped to Modbus registers as **40000 + Menu×100 + Parameter**. For example, Pr **01.021** will be register number 40121. Only parameters up to **MM.99** are accessible on the Unidrive M target via the RS485 port on SI-Applications Plus.

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 Pr **70.001** in 32-bit mode, the register number will be 40000 + (0x4000 + 70×100 + 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
- RtuPresetMutlipleParas
- RtuMasterReply
- RtuMasterStatus

6.1.2.1 FC03 Read multiple registers

Read a contiguous array of 16 bit 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 6-5 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 16 bit registers MSB
5	Number of 16 bit registers LSB
6	CRC LSB
7	CRC MSB

Table 6-6 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

6.1.2.2 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 32 bit parameter but only 16bits of data can be sent.

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

Table 6-7 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 6-8 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

6.1.2.3 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 6-9 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 6-10 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

6.1.2.4 FC23 Read/Preset 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 6-5 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
12	Register data 0 LSB
11 + byte count	CRC LSB
12 + byte count	CRC MSB

Table 6-11 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

6.1.3 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 Modbus organization web site at www.modbus.org. Note that the 32-bit access mode is specific to Control Techniques/ Leroy Somer.

6.1.4 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.

6.2 CTNet

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

6.3 SI-Applications Plus mapping parameters (fieldbus)

SI-Applications Plus module parameters can be access from other option modules fitted to the drive using the standard Unidrive M parameter notation Pr **S.MM.PPP**. For example, Pr **70.001** on a SI-Applications Plus module in slot 3 can be accessed from another fieldbus module using the notation Pr **3.70.001**.

7 DPL Programming

This chapter covers:

- Basic DPL program structure and syntax
- Basic DPL commands

NOTE The full reference for all DPL commands and function blocks is provided by the on-line help guides.

7.1 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

7.1.1 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 #1.021
$DEFINE PRESET_REF_2 #1.022
$DEFINE SPEED_FB #3.002
```

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.

NOTE To give backward compatibility with the Unidrive SP **MM.PP** style parameter access, the SI-Applications Plus supports both **MM.PP** and **MM.PPP** style parameter access e.g. 01.21, 1.21, 01.021, 1.021 are all valid ways to access Unidrive M Pr **01.021**.

7.2 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 7-1 Common tasks

Task Name	Priority	Purpose
INITIAL	2	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. Only the ERROR task can run before this task is completed.
BACKGROUND	6	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	5	Task executed on a fixed timebase (between 1-200 ms) 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 250 µs to 8 ms). 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	3	Event tasks only run when a certain event occurs. Events can be raised from various sources such as CTNet, other System Integration Modules in the drive or the user program and usually only have a very small number of instructions. They can be likened to interrupt service routines.
EVENT1	3	See above description.
EVENT2	3	See above description.
EVENT3	3	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.

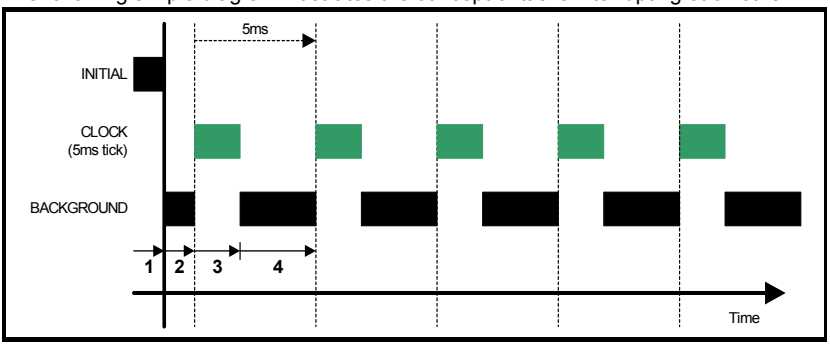
NOTE When using the CLOCK, POS0 and POS1 tasks, it is advisable to avoid using code such as the FOR and DO WHILE loops. This may cause a DPL over-run error (tr54).

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).

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:

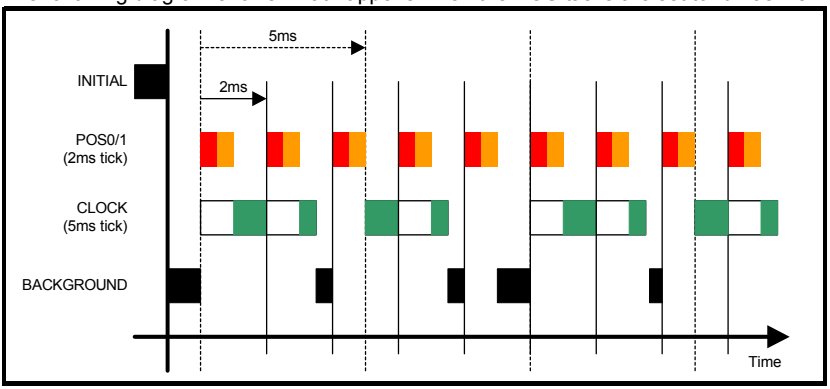


Key:

1. INITIAL task has exclusive control. No other tasks can run.
2. BACKGROUND task runs after INITIAL task has completed.
3. CLOCK task interrupts BACKGROUND task. The drive controls when the CLOCK task will execute. BACKGROUND task is suspended.
4. CLOCK task has finished and now the BACKGROUND task can continue running - 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 5 ms). This means that the instructions within the CLOCK task MUST take less than 5 ms to complete otherwise the BACKGROUND task will not be able to execute, or a processor overload trip will occur.

The following diagram shows what happens when the POS tasks are set to run 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 **81.004** can be used to determine how heavily loaded the module is.

NOTE The SI-Applications Plus module provides Pr **88.003** through to Pr **88.008** which will give a greater accuracy on the available resources.

7.2.1 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.

7.3 Variables

7.3.1 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 7-2 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:

```
$flt single
```

7.3.2 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.

7.3.3 Initialization 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
}
```

7.3.4 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 module. Remember that resetting the drive from a tripped condition will also cause a reset (depending on the value in Pr **81.15**).

7.3.5 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.

7.3.6 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 512 kB - it is used to store the compiled DPL file, constant array data, and optionally, the DPL file itself.

7.3.7 Storage Space - Number of variables

All variables, fixed-size variable arrays and PLC registers live in an 200 kB 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.

7.3.8 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 "Result% = 1"
ENDIF
```

Example 2 (arrays):

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

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

7.3.9 PLC Registers

The 'PLC' area is a special range of pre-defined 32-bit registers. The PLC registers are split into 10 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. Eight of the register sets are also saveable in the SI-Applications Plus *Flash* memory.

See section 5.4 *Menus 70-79 - PLC Registers* on page 37 for further information on PLC registers.

7.3.10 RAM files

RAM files enable the user to store ‘files’ in the user RAM of the SI-Applications Plus 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.

7.4 Parameters

Parameters are categorized into two sets:

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

NOTE The SI-Applications Plus module always guarantees that the drive parameter database it uses matches that of the host drive. When a module is installed to a Unidrive M for the first time and powered up the word “Waiting For Options” may appear on the drive display for a few seconds. This indicates the module is synchronizing databases. This will only occur the first time the module is installed to the drive.

7.4.1 Reading and Writing Parameters

Reading and writing parameters is achieved by using the # command. Parameters may be accessed in Pr **MM.PP** format as they were on Unidrive SP where the leading zero may be removed from the parameter, or in Pr **MM.PPP** format as they are on the Unidrive M keypad.

For example, to read the speed feedback parameter (Pr **03.002**), use:

```
Speed% = #3.002
```

To write to a speed reference parameter (e.g. Pr **01.022**), use:

```
#01.022 = 1500
```

Note that the leading zeros in the menu/parameter field are optional. For example **#3.002**, **#03.002**, **#3.002**, **#03.02**, **#3.02**, **#03.2** and **#3.2** will access exactly the same parameter.

This gives backward compatibility of software written for Unidrive SP, since most Unidrive M parameters directly overlay with those of Unidrive SP.

7.4.2 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 **01.019** has a range of 0.000 - 0.099. Reading the parameter using:

```
Speed_Fine% = #INT1.019
```

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

```
#INT1.019 = 45
```

will set the parameter to 0.045 (same as Pr **01.019** = 0.045). The benefit of this is that the DPL program can use integer variables (%) instead of floating-point, thus providing a speed advantage.

NOTE The #INT directive can be used with **MM.PP** parameter format as they were on Unidrive SP where the leading zero may be removed from the parameter, or in **MM.PPP** format as they are on the Unidrive M keypad.

7.5 Operators

DPL offers all the standard operators as follows:

Table 7-3 Standard operators in order of precedence

Operator	Meaning
-	Arithmetic negation
!	Boolean negation (unary)
!(..., nbit)	Negation of *nbit* bits
*	Multiplication
/	Division
%	Modulo (remainder)
+	Addition
-	Subtraction
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR (XOR)

Table 7-4 Conditional operators in order of precedence

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

7.6 Basic DPL Commands

The DPL language implemented for the SI-Applications Plus is backwards compatible with the SM-Applications family of modules.

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

FOR loop

```
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 recognized by the DPL compiler and replaced by the appropriate numeric value.

Table 7-5 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 ***90.012***-Pr ***90.015****.*

```
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 command takes no input and returns 10 outputs.

CTNetDiagnostics Output Values

Output	Value
1	Total number of messages handled by this node
2	Number of cyclic data overruns
3	Lost RX messages
4	Number of retries
5	Number of recons
6	Number of excessive NAKs
7	Duplicate sync
8	Number of locally generated recons
9	Number of non-cyclic messages
10	Number of lost routing messages

GETPARATTR

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

```
(Max%, Min%, Flags%) = GETPARATTR(Menu%, Par%)
```

NOTE The Par% input to the GETPARATTR function block accepts both **MM.PPP** Unidrive M and **MM.PP** Unidrive SP parameter references e.g. Par% = 001 and Par% = 01 would both access **MM.001** on Unidrive M.

CModeXfer

This allows the user to change the drive mode without any installed System Integration 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 and 24 to 28 will not be defaulted when using this command.

RtuReadRegs

NOTE Only **MM.PP** parameter access is supported for this RS485 communications protocol, therefore to read from or write to a Unidrive M parameter it is necessary to omit the leading zero from the parameter identifier e.g. Pr **1.021** may be identified as 121 or Menu% = 1 Par% 21.

RtuReadParas
RtuReadInputRegs
RtuPresetRegs
RtuPresetParas
RtuMasterReply
RtuMasterStatus

These commands are implemented to allow the user to utilize the SI-Applications Plus 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 Pr **81.049**.

```
SETUSERID(101)    // set #81.049 to 101.
```

ANSIREPLY

```
(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 initialize 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% = ANSIWRITEN(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

NOTE Only **MM.PP** parameter access is supported for this RS485 communications protocol, therefore to read from or write to a Unidrive M parameter it is necessary to omit the leading zero from the parameter identifier e.g. Pr **1.021** may be identified as 121 or Menu% = 1 Par% 21.

AssRAM
UnassRAM
RamLength
SetRamLength

These commands allow the programmer to use the RAM files within the 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.

7.6.1 DPL Commands and Function Blocks

There is a rich list of commands and functions that may be used in a DPL program. Please refer to the on-line help.

7.7 User Defined Function Blocks

7.7.1 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 UDFBs. Note however that you cannot create standard task sections (such as POS0) with a UDFB.

7.7.2 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 UDFBs at the top of a program due to the fact that a UDFB must be defined before it is used.

7.7.3 Encapsulation and Data Storage

Unlike any task of a DPL program, UDFBs 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 nodes 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 UDFBs.

A UDFB can of course access drive parameters and 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

7.7.4 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, _PID1 and _My_Func
```

These are examples of illegal names:

```
MyFunc, UDFB1
```

7.7.5 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.

7.7.6 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):

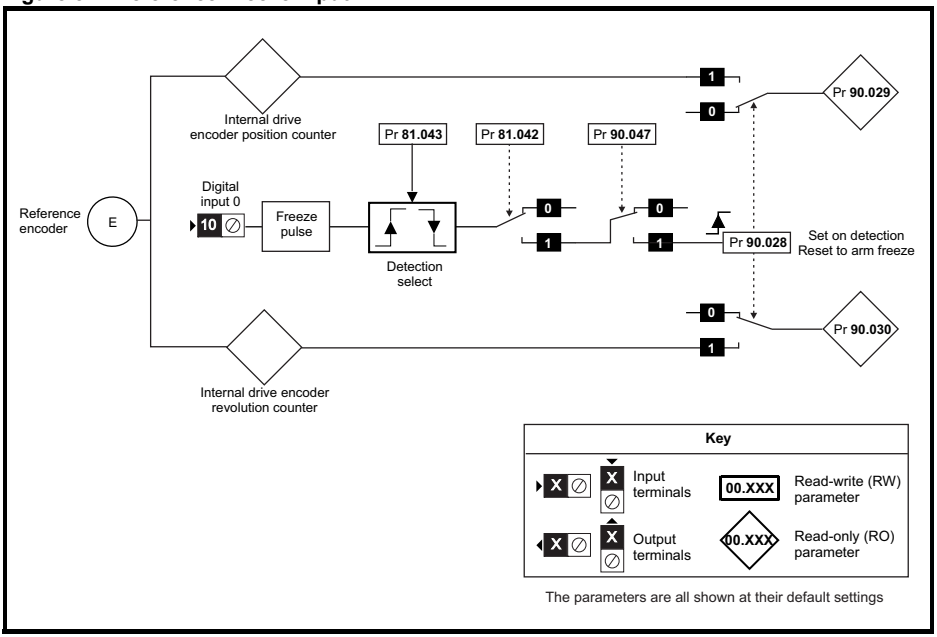
```
(Output%) = _simplefb(Input1%, Input2%) {
// Initialization code:

Scale% = 500     // initialise a local variable

FBbody
// main body code:

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

}
```

8 Freeze and marker

8.1 Freeze input

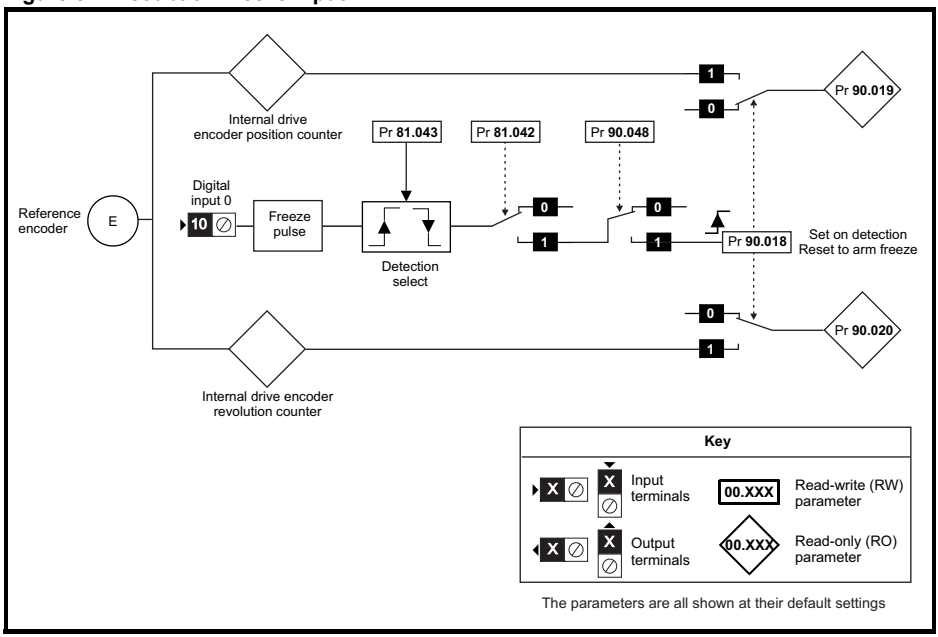
Digital Input 0 (DIGIN0) on the module can be used to 'freeze' the reference and feedback encoder counters.

Figure 8-1 Reference Freeze Input



The encoder revolution counter is cached into Pr **90.030** and the encoder position is cached into Pr **90.029**.

Figure 8-2 Feedback Freeze Input



The encoder revolution counter is cached into parameter Pr **90.020** and the encoder position is cached into Pr **90.019**.

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 **81.043** to either 0 (rising edge) or 1 (falling edge). Pr **81.042** enables the position to be written to Pr **90.019** and Pr **90.029** and the revolution counters to be written to Pr **90.020** and Pr **90.030**.

When a freeze input is seen Pr **90.018** and Pr **90.028** are set to 1 automatically so that the position can be written to Pr **90.019** and Pr **90.029** and the revolution counters can be written to Pr **90.020** and Pr **90.030**. Pr **90.018** and Pr **90.028** must be reset to zero if the user wants to update the data again on the next freeze pulse.

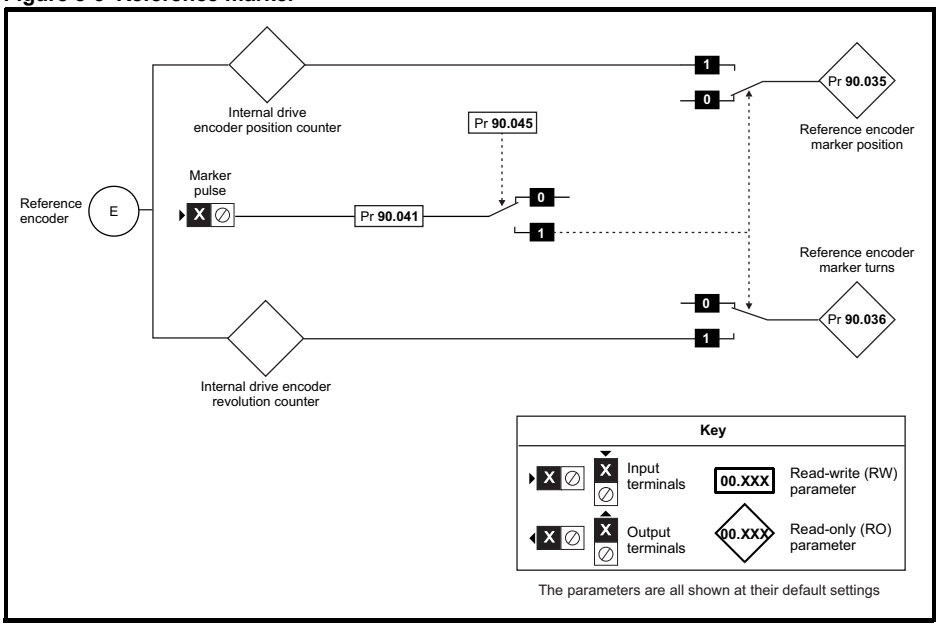
The freeze functions on the SI-Applications Plus module utilise the F1 and F2 freeze functions in the drive or position feedback option module to perform the position capture. The SI-Application module will configure the drive or option module freeze function parameters as follows. If Pr **81.042** is set to 1, then at power up and on REINIT, Pr **03.100** and Pr **03.105** will be set to 4. This sets the common freeze line as the F1 and F2 freeze trigger sources on the drive.

If Pr **90.043** or Pr **90.044** is selected to be a P1 position source on either the drive or an option module, then the F1 freeze function on the selected device will used and P1 will be selected as the F1 freeze position source. If Pr **90.043** or Pr **90.044** is selected to be a P2 position source on either the drive or an option module, then the F2 freeze function on the selected device will used and P2 will be selected as the F2 freeze position source. See the description for Pr **90.044** on page 51 for more information.

8.2 Marker pulse

The SI-Applications Plus 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.

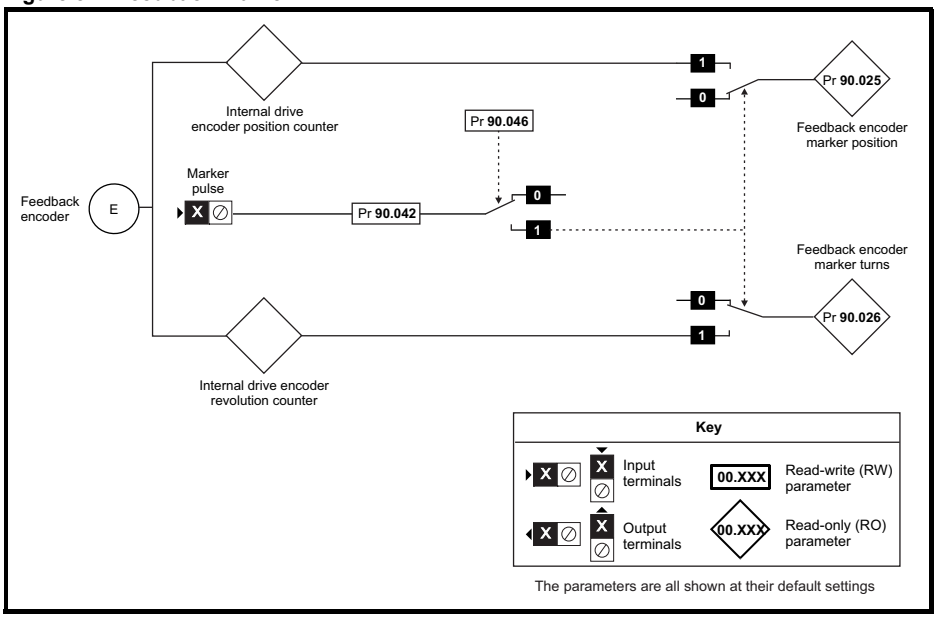
Figure 8-3 Reference marker



The marker position is cached into Pr **90.035** and the marker revolution counter is cached into Pr **90.036**.

The user sets Pr **90.041** to a zero and the drive sets Pr **90.041** to a 1 EVERY time a marker is detected. The marker data must be consumed before the next marker pulse.

Figure 8-4 Feedback marker



The marker position is cached into Pr **90.025** and the marker revolution counter is cached into Pr **90.026**.

The user sets Pr **90.042** to a zero and the drive sets Pr **90.042** to a 1 EVERY time a marker is detected. The marker data must be consumed before the next marker pulse.

9 CTSync

9.1 Overview

The SI-Applications Plus module may be used to synchronize two or more 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 32-bit integers and 1 x Unsigned 8-bit integer.

Only one SI-Applications Plus 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 Drive synchronisation source

To use CTSync to synchronise multiple drives, the drives must be told which option module provides the synchronisation source.

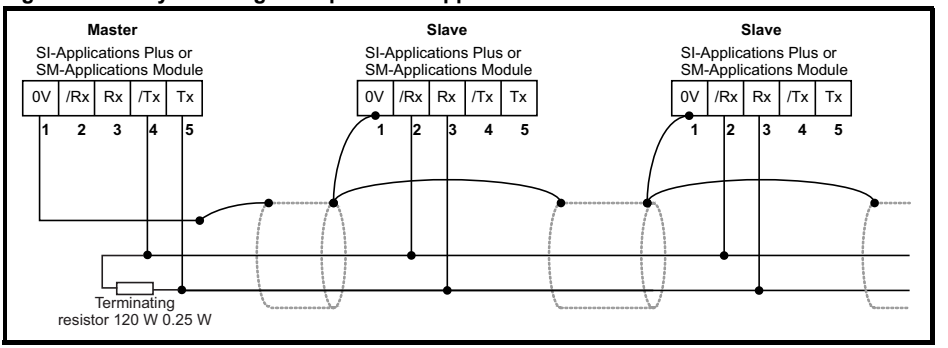
Pr **11.001** on the drive is used to select and enable timing synchronisation between an option module and the drives control system. Therefore, if it is required for CTSync on an SI-Applications Plus module fitted in slot 3 to be the synchronisation source for the drive, then Pr **11.001** should be set to "Slot 3". Pr **11.002** on the drive shows the actual synchronisation source being used by the drive.

9.3 Connections

CTSync operates via a connection between the EIA-RS485 ports of the SI-Applications Plus module on the network in either 2-wire or 4-wire. Refer to section 3.6 *EIA-RS485 connections* on page 15 for information on how to connect the SI-Applications Plus Module RS-485 port.

To simplify wiring the Slave transmit and Master receive signal line connections can be omitted in 4-wire mode (See Figure 9-1). This is because the Master does not receive a response from the Slave.

Figure 9-1 CTSync Wiring Example for SI-Applications Plus



9.4 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 **81.012**)
- 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 m.

9.5 CTSync Function Blocks

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

9.5.1 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.

9.5.2 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.

9.5.3 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

NOTE Par% directive can be used with **MM.PP** parameter format as they were on Unidrive SP where the leading zero may be removed from the parameter, or in **MM.PPP** format as they are on the Unidrive M keypad.

9.5.4 CTSYNCEnableOutputChannel

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

This function block allows the specified channel to write its data to the drive 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.

9.5.5 CTSYNCDisableOutputChannel

```
Result% = CTSYNCDisableOutputChannel(Channel%)
```

This function block causes the specified channel to cease writing its data to the Unidrive M.

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.

9.5.6 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.

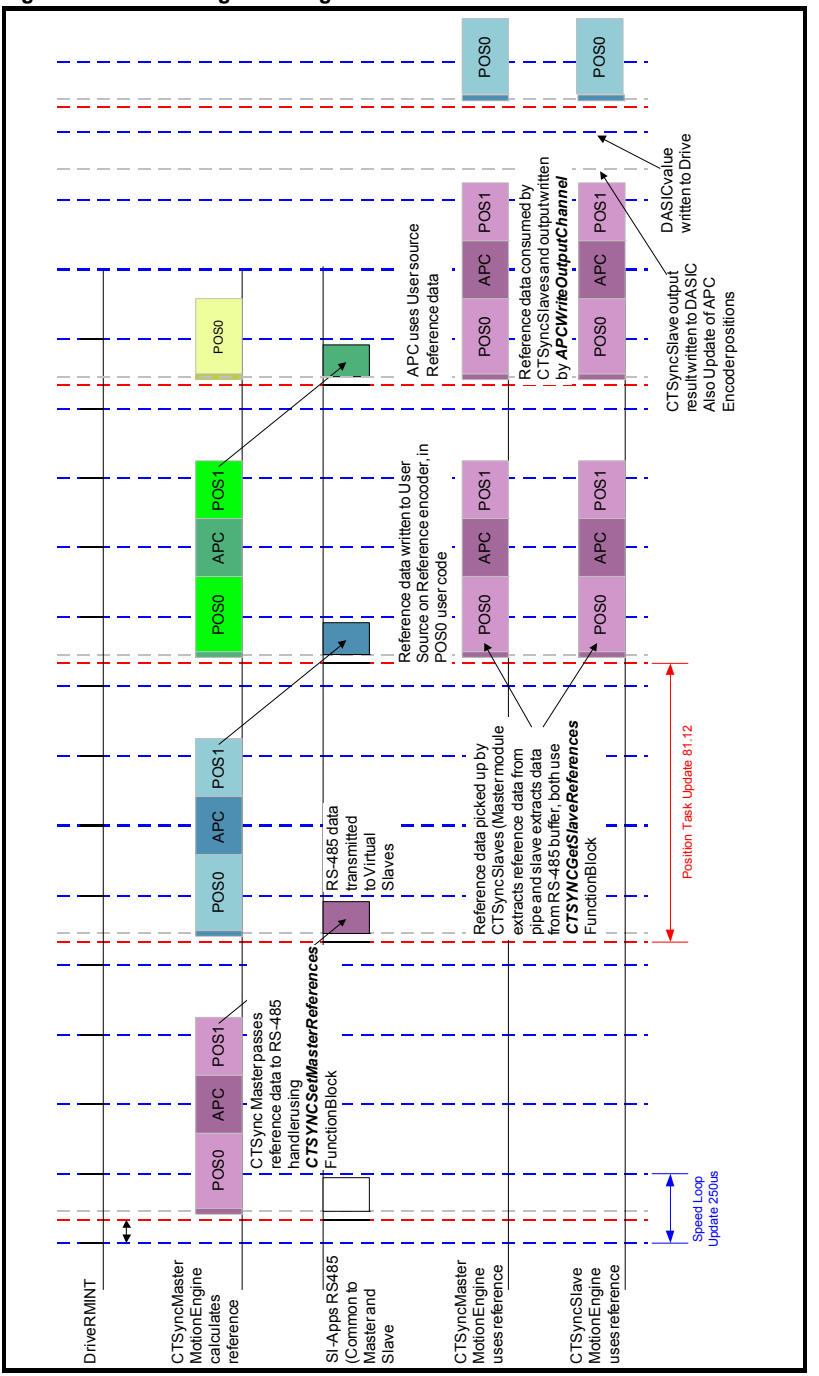
9.6 Motion Engine

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 9-2 *Motion Engine timing*

Figure 9-2 Motion Engine timing



9.7 Virtual Master Example

9.7.1 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)
}


POS0
{

    //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)

}
```

9.7.2 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)
}


POS0
{

    //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 Diagnostics

This chapter details the following:

- Run-time errors and trip codes
- Handling of run-time errors
- CTNet network status
- Support

10.1 Run-time errors

A run-time error is an error which occurs in a specific operation of the SI-Applications Plus 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) 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 **Slotx Error** (where x is the slot number) with the run-time error code placed into Pr **81.050**
- Drive may be tripped on another **Slotx ****** 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 **81.014**. This is detailed in section 10.3, below.

10.2 Drive display trip codes

The table below shows the possible trip codes that will be displayed on the drive when an error is detected in the SI-Applications Plus module which instigates a drive trip. Remember, not all run-time errors instigate a drive trip.

Table 10-1 Drive display trip codes

Drive trip code	Fault	Description
Slotx* HF	Hardware Fault	The drive has detected that an System Integration Module is present, but is unable to communicate with it.
Slotx* Watchdog	Watchdog Timeout	Indicates a user program which has utilised the *watchdog* feature has failed to issue the WDOG command within 200 ms.
Slotx* Error	Error	Run-time trip generated by the SI-Applications Plus module either due to a user DPL program error or some other event. The actual error code is placed into Pr **81.050**.
Slotx* Not Fitted	Not installed	Module was disconnected while operational, or module has crashed. This trip will also occur if a drive slot is configured for a SI-Applications Plus module, but the module is not installed in the slot.
Slotx* Different	Different installed	This trip will occur when an SI-Applications Plus module is installed to a slot previously occupied by another System Integration Module, or is installed to a previously unused slot.

* Where x determines the slot number. For example an error with the module in slot 3 would give a Slot3 Error trip.

10.3 SI-Applications Plus module run-time error codes

If the SI-Applications Plus module detects an error during operation the error code will be placed in the following parameter:

Pr 81.050	Run-time Error Code		
Access	RO	Range	0 to 255
Default	N/A	Update Rate	On error

For certain errors the user may select if the drive should trip as well. This is configured with the Global Run-time Trip enable parameter:

Pr 81.014	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 Initialization and the program will not be started.

Table 10-1 SI-Applications Plus modules & motion processors error codes

Error Code	Reason	Drive trip?	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 Pr **81.017**=0 the value written will be automatically limited and no error will occur.	May	Yes	Yes
45	Invalid synchronization modes	Yes	No	Not Run
46	Unused	N/A	N/A	N/A
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 Pr **90.011**	Yes	No	No

Error Code	Reason	Drive trip?	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 Pr **88.002** 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 *Pr 81.044* on page 35 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/V/W/X/Y) 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 Pr **81.024** and network connections.	May	No	No
63	CTNet invalid node ID. Check Pr **81.023**.	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 20 mA.	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 Pr **81.037** = 1 and a program is downloaded.	May	No	Yes
71	Failed to change drive mode	Yes	No	Yes

Error Code	Reason	Drive trip?	ERROR task?	Prog Halted?
72	Invalid CTNet buffer operation.	Yes	Yes	Yes
73	Fast parameter Initialization failure	Yes	No	No
74	Over-temperature	Yes	Yes	Yes
75	Hardware unavailable. The user program attempted to access unavailable hardware.	Yes	Yes	Yes
76	Module type cannot be resolved. Module is not recognized.	Yes	No	Not Run
77	Inter-System Integration Module comms error with module in slot 1.	Yes	Yes	Yes
78	Inter-System Integration Module comms error with module in slot 2.	Yes	Yes	Yes
79	Inter-System Integration Module comms error with module in slot 3.	Yes	Yes	Yes
80	Inter-System Integration Module comms error with module unknown slot.	Yes	Yes	Yes
81	• Slot selected as the reference or feedback does not contain a position System Integration Module • Attempt to change the Reference source or the Feedback source in more than one task. APC internal error. See Pr **81.038**. This may be caused by one of the following: • CAM table too small • A change of too many CAM segments has occurred at the CAM table input • CAM is selected but size is zero • CAM absolute mode selected and Reset Index or Reset Position in segment is out of range	May	Yes	Yes
82	Communications to drive faulty.	May	Yes	Yes

10.4 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 M and module parameters can be accessed.

The run-error code can be determined using this parameter:

Pr 88.001	Error Status / Reset		
Access	RW	**Range**	0 to 9999
Default	N/A	**Update Rate**	On error

This parameter has two purposes - when read it will return the run-time error code the same as Pr **81.050** (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 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 **81.013=**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 Pr **88.001** will result in any drive trip being automatically cleared as well as resetting all installed options in the drive.

The task which caused a run-time error can be determined by reading Pr **88.002**, 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 Pr **10.038**.

10.5 Resource monitoring

The SI-Applications Plus module provides more realistic resource data than is available from parameter Pr **81.004** as shown below.

Pr 88.003	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 **88.004** and Pr **88.005** become active. If set to zero, parameters Pr **88.004** and Pr **88.005** will read zero.

Pr 88.004	Free Resource for Motion Engine Tasks		
Access	RW	**Range**	0 to 95
Default	0	**Update Rate**	See Pr **81.012**

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 88.005	Motion Engine Peak Resource Detect		
Access	RW	**Range**	0 to 95
Default	0	**Update Rate**	See Pr **88.004**

This parameter shows the peak that parameter Pr **88.004** reaches. It will show the lowest value that parameter Pr **88.004** reaches since the monitoring was enabled (parameter Pr **88.003**). 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 88.006	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 **88.007** and Pr **88.008** become active. If set to zero, parameters Pr **88.007** and Pr **88.008** will read zero.

Pr 88.007	Free Resource for Clock Task		
Access	RO	**Range**	0 to 95
Default	NA	**Update Rate**	See Pr **81.011**

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.

Pr 88.008	Clock Task Peak Resource Detect		
Access	RO	Range	0 to 95
Default	NA	Update Rate	See Pr **81.011**

This parameter shows the peak that parameter Pr **88.007** reaches. It will show the lowest value that parameter Pr **88.007** reaches since the monitoring was enabled (parameter Pr **88.006**). 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.

10.6 Support

The information from the parameters described below should always be noted before contacting your supplier for technical support.

10.6.1 Module Firmware

Pr 81.002	Firmware - Major Version		
Access	RO	Range	00.00 to 99.99
Default	N/A	Update Rate	N/A

Pr 81.051	Firmware - Minor Version		
Access	RO	Range	0 to 99
Default	N/A	Update Rate	N/A

The full version of the module firmware version can be read for the corresponding slot. This manual was written for SI-Applications Plus module installed with V02.00.00 firmware and later. The table below shows how to construct the full firmware version from these values.

Table 10-2 Firmware version

Major Version	Minor Version	Firmware Version
2.00	0	V02.00.00

11 Migration guide

This section provides information on the parameter differences between the SM-Applications family of modules for Unidrive SP and the SI-Applications Plus module for Unidrive M with regards to migration from an SM-Applications family module to SI-Applications Plus.

11.1 SyPTPro Porting Tool

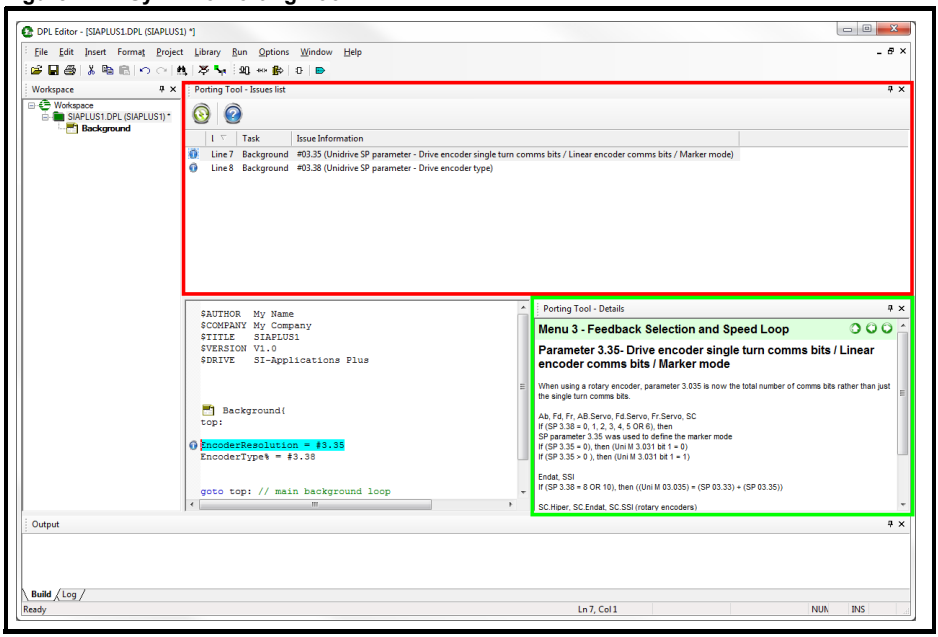
SyPTPro contains a porting tool that can be used to help port SM-Applications Plus (including SM-Aplications Lite V2 and SM-Register) user programs for use on the Unidrive M drive with an SI-Applications Plus module.

The tool aims to provide information targeted at porting the user program. A list of issues will be generated for the user program and detailed information will be provided to allow the user to resolve the issue. No automatic changes will be made. The tool should only be used in an advisory capacity.

The porting tool is available if the selected device is a SI-Applications Plus module and it can be started by clicking the icon on the toolbar or selecting Run > Show Porting Issues from the menus within the DPL editor.

Within the porting tool the issues list (highlighted in red in Figure 11-1) displays possible issues associated with porting the user program, and the details panel (highlighted in green in Figure 11-1) displays information about the issue currently selected in the issues list pane. This then allows the user to make any necessary changes to the user program.

Figure 11-1 SyPTPro Porting Tool



11.2 Module parameter differences

Most SI-Applications Plus module parameters are identical to the SM-Applications family of modules, but there are a small number of differences. These differences are listed in Table 11-1 below, but will also be identified by the SyPTPro Porting Tool.

Table 11-1 SM-Applications module family to SI-Applications Plus parameter differences

Parameter	Description	Change
90.21	Disable Drive Encoder Position Check	These parameters have been removed. An alternative method of communicating with an encoder is provided by parameters Pr **03.067**, Pr **03.068** and Pr **03.068** on the drive.
90.22	Drive Encoder Comms Transmit Register	
90.23	Drive Encoder Comms Receive Register	
91.01	Short-cut Enable	Only bits 0 and 1 of Pr **91.001** can be set. Bits 2, 3 and 4 cannot be used as the associated short-cut parameters are not available. See Pr **91.04**, Pr **91.09** and **91.10** below for more information.
91.04	Torque Setpoint (Pr **04.008**)	This parameter has been removed. The value that was intended to be written to this parameter can be written directly to parameter Pr **04.008** on the drive instead. Pr **04.008** on Unidrive M is read every 250 µs which was the same as Pr **91.04** on the SM-Applications modules. Pr **91.04** was an integer parameter on SM-Applications whereas Pr **04.008** on Unidrive M is a fixed-point integer parameter. Due to it taking longer to access fixed-point integer points than integer parameter, it is recommended to use #INT04.008 when accessing Pr **04.008**.
91.07	Current Feedback (Pr **04.002**)	The units of this parameter have changed. On SI-Applications Plus this parameter has units of 0.001 A.
91.09	Drive Analog Input 2 Value	The range of these parameters has changed from ±1023 to ±2047 due to the increase in resolution of analog inputs 2 and 3 on the drive from 10 bits to 11 bits.
91.10	Drive Analog Input 3 Value	
91.11	Drive Analog Output 1	These parameters have been removed. The value that is required to be written to the analog output short-cut parameters should be written to application parameters such as Pr **18.011** and Pr **18.012**. The value will need to be scaled depending on the range of the applications parameters used. If Pr **18.011** or Pr **18.012** is used then the original value will need to be multiplied by 32 to ensure the correct output level on the analog output. The analog output source parameters Pr **07.019** or Pr **07.022** should be set to the relevant applications parameters (such as Pr **18.011** and Pr **18.012**). The update rate of the analog outputs on Unidrive M is dependent on the update rate of the source parameter with the fastest update rate being 250 µs. Therefore, if the required value is written to the application parameter at 250 µs, then the analog output will also be updated every 250 µs.
91.12	Drive Analog Output 2	

11.3 Drive parameter differences

The Unidrive M parameter set is based on the Unidrive SP parameter set where the Unidrive M parameter set can be considered as a superset of Unidrive SP. However, there are some minor parameter changes which can be identified by using the SyPTPro Porting Tool.

12 Quick reference

Refer to Chapter 5 *Parameters* on page 23 for full details of these parameters.

Table 12-1 Set-up parameters

Parameter	Description	Cross Ref	Range	Default
Pr **81.001**	Module option code	section 5.3.1 on page 25	0-499	N/A
Pr **81.002**	Module firmware version		0-99.99	
Pr **81.003**	DPL program status		0-3	
Pr **81.004**	Available system resource %		0-100 %	
Pr **81.005**	EIA-RS485 address		0-255	11
Pr **81.006**	EIA-RS485 Mode		0-255	1
Pr **81.007**	EIA-RS485 Baud rate		0-9	4
Pr **81.008**	EIA-RS485 Turn-around delay		0-255 ms	2 ms
Pr **81.009**	EIA-RS485 Tx Enable Delay		0-1 ms	0 ms
Pr **81.010**	DPL print routing		0-1	0
Pr **81.011**	Clock tick time (ms)		0-200 ms	0
Pr **81.012**	POS task schedule rate		0-6	0
Pr **81.013**	Enable autorun		0-1	1
Pr **81.014**	Global run-time trip enable		0-1	0
Pr **81.015**	Disable Reset on Trip Cleared		0-1	0
Pr **81.016**	Encoder Data Update Rate		0-3	0
Pr **81.017**	Enable parameter over-range trips		0-1	0
Pr **81.018**	Watchdog enable		0-1	0
Pr **81.019**	Save request		0-1	0
Pr **81.020**	Save on ‘UU’ trip		0-1	0
Pr **81.021**	Include menu 20 for save/restore		0-1	0
Pr **81.022**	CTNet token ring ID		0-255	0
Pr **81.023**	CTNet node address		0-255	0
Pr **81.024**	CTNet baud-rate		0-3	1
Pr **81.025**	CTNet sync setup (SSFF)		0-9999	0
Pr **81.026**	CTNet Easy Mode 1 destination node (NNNSS)		0-25503	0
Pr **81.027**	CTNet Easy Mode 1 source parameter (MMPP)		0-9999	0
Pr **81.028**	CTNet Easy Mode 2 destination node (NNNSS)		0-25503	0
Pr **81.029**	CTNet Easy Mode 2 source parameter (MMPP)		0-9999	0
Pr **81.030**	CTNet Easy Mode 3 destination node (NNNSS)		0-25503	0
Pr **81.031**	CTNet Easy Mode 3 source parameter (MMPP)		0-9999	0

Parameter	Description	Cross Ref	Range	Default
Pr **81.032**	CTNet Easy Mode Slot 1 destination (MMPP)	section 5.3.1 on page 25	0-9999	0
Pr **81.033**	CTNet Easy Mode Slot 2 destination (MMPP)		0-9999	0
Pr **81.034**	CTNet Easy Mode Slot 3 destination (MMPP)		0-9999	0
Pr **81.035**	CTNet Sync EVENT task ID		0-4	0
Pr **81.036**	CTNet Diagnostic parameter		-3-32767	N/A
Pr **81.037**	Reject download if drive enabled		0-1	0
Pr **81.038**	APC Run-time trip		0-1	
Pr **81.039**	Inter-module Drive Synchronization Status		0-3	
Pr **81.041**	Indexer Contol		0-3	
Pr **81.042**	Pass freeze through to drive		0-1	
Pr **81.043**	Freeze invert		0-1	
Pr **81.044**	Task priority level		0-255	
Pr **81.045**	User Setup Parameter 1		N/A	N/A
Pr **81.046**	User Setup Parameter 2		N/A	
Pr **81.047**	User Setup Parameter 3		N/A	
Pr **81.048**	DPL line number of error		32 bit	0
Pr **81.049**	User program ID		16 bit	
Pr **81.050**	Run-time error code		0-255	
Pr **81.051**	Minor software revision		0-99	N/A

Table 12-2 Timer unit parameters

Parameter	Description	Cross Ref	Range	Default
Pr **85.001**	Timer Unit Control Word	section 5.5 on page 38	13bit	N/A
Pr **85.002**	Timer Unit Status Word		0-3	
Pr **85.003**	Timer Unit 16-bit Timer Count		16bit	
Pr **85.004**	Timer Unit Wrap-around Limit		16bit	
Pr **85.005**	Timer Unit Timer Capture Cache		16bit	

Table 12-3 Digital I/O parameters

Parameter	Description	Cross Ref	Range	Default
Pr **86.001**	Digital Input 0	section 5.6 on page 41	0-1	N/A
Pr **86.002**	Digital Input 1		0-1	
Pr **86.003**	Digital Output 0		0-1	
Pr **86.004**	Digital Output 1		0-1	
Pr **86.005**	Digital Outputs 0 and 1		0-3	

Table 12-4 Status parameters

Parameter	Description	Cross Ref	Range	Default
Pr **88.001**	Error Code / Reset	section 5.7 on page 42	0-9999	N/A
Pr **88.002**	Task In Error		0-50	
Pr **88.003**	POS Resource Monitoring		0-1	0
Pr **88.004**	Free Resource for Motion Engine Tasks		0-95	N/A
Pr **88.005**	Motion Engine Peak Resource Detect		0-95	
Pr **88.006**	CLOCK Task Resource Monitoring		0-1	0
Pr **88.007**	Free Resource for Clock Task		0-95	N/A
Pr **88.008**	Clock task peak resource detect		0-95	

Table 12-5 General parameters

Parameter	Description	Cross Ref	Range	Default
Pr **90.001**	Feedback encoder position (2^{32}/rev)	section 5.8 on page 44	Signed 32-bit	N/A
Pr **90.002**	Feedback Encoder Revolution Count		Unsigned 16-bit	
Pr **90.003**	Reference Encoder Position (2^{32}/rev)		Signed 32-bit	
Pr **90.004**	Reference Encoder Revolution Count		Unsigned 16-bit	
Pr **90.010**	Drive Mode		Signed 16-bit	
Pr **90.011**	Drive Status and Control Word		Signed 16-bit	
Pr **90.012**	Event task schedule reason		Unsigned 16-bit	
Pr **90.013**	Event1 task schedule reason		Unsigned 16-bit	
Pr **90.014**	Event2 task schedule reason		Unsigned 16-bit	
Pr **90.015**	Event3 task schedule reason		Unsigned 16-bit	
Pr **90.018**	Feedback Encoder Freeze Flag		0/1	
Pr **90.019**	Feedback Encoder Freeze Position		Signed 32-bit	
Pr **90.020**	Feedback Encoder Freeze Turns		Unsigned 16-bit	
Pr **90.024**	Module slot number		Unsigned 8-bit	
Pr **90.025**	Feedback encoder marker position (2^{32}/rev)		Signed 32-bit	
Pr **90.026**	Feedback encoder marker turns (2^{16}/rev)		Unsigned 16-bit	

Parameter	Description	Cross Ref	Range	Default
Pr **90.027**	Second Processor database version number	section 5.8 on page 44	Unsigned 16-bit	N/A
Pr **90.028**	Reference Encoder Freeze flag		0/1	
Pr **90.029**	Reference Encoder Freeze Position		Signed 32-bit	
Pr **90.030**	Reference Encoder Freeze Turns		Unsigned 16-bit	
Pr **90.031**	Feedback Encoder Turns and Coarse Position		Signed 32-bit	
Pr **90.032**	Reference Encoder Turns and Coarse Position		Signed 32-bit	
Pr **90.033**	Feedback Encoder Freeze Turns and Coarse Position		Signed 32-bit	
Pr **90.034**	Reference Encoder Freeze Turns and Coarse Position		Signed 32-bit	
Pr **90.035**	Reference Encoder Marker Position (2^{32}/rev)		Signed 32-bit	N/A
Pr **90.036**	Reference Encoder Marker turns (2^{16}/rev)		Unsigned 16-bit	
Pr **90.037**	Feedback Encoder Marker Turns and Coarse Position		Signed 32-bit	
Pr **90.038**	Reference Encoder Marker Turns and Coarse Position		Signed 32-bit	
Pr **90.039**	Drive Keypad Button Status		Signed 16-bit	
Pr **90.040**	Event Task Trigger		Unsigned 16-bit	0
Pr **90.041**	Reference Encoder Marker Flag		0/1	N/A
Pr **90.042**	Feedback Encoder Marker Flag		0/1	N/A
Pr **90.43**	Reference Encoder Source		Unsigned 8-bit	N/A
Pr **90.044**	Feedback Encoder Source		Unsigned 8-bit	N/A
Pr **90.045**	Reference Marker Flag Enable		0/1	N/A
Pr **90.046**	Feedback Marker Flag Enable		0/1	N/A
Pr **90.047**	Reference Freeze Enable		0/1	N/A
Pr **90.048**	Feedback Freeze Enable		0/1	N/A
Pr **90.049**	APC Runtime Error ID		32bit	N/A

Table 12-6 Fast access parameters

Parameter	Description	Cross Ref	Range	Default
Pr **91.001**	Short-cut enable	section 5.9 on page 52	Unsigned 8-bit	0
Pr **91.002**	Speed set-point (Pr **01.021**)		Signed 32-bit	N/A
Pr **91.003**	Hard-speed reference (Pr **03.22**)		Signed 32-bit	N/A
Pr **91.005**	Full scale speed (rpm)		Signed 32-bit	1500
Pr **91.006**	Speed feedback		Signed 32-bit	N/A
Pr **91.007**	Current feedback (Pr **04.002**)		Signed 32-bit	N/A
Pr **91.008**	Drive analog input 1 value		±4000	N/A
Pr **91.009**	Drive analog input 2 value		±2047	N/A
Pr **91.010**	Drive analog input 3 value		±2047	N/A
Pr **91.016**	Drive digital inputs		Unsigned 8-bit	N/A
Pr **91.017**	Number of Valid CTSync Messages Received		Signed 32-bit	N/A
Pr **91.018**	Number of Bad CTSync Messages Received		Signed 32-bit	N/A
Pr **91.019**	Number of Missing CTSync Messages		Signed 32-bit	N/A
Pr **91.020**	CTSync Synchronization Signal Width too Short		Signed 32-bit	N/A
Pr **91.021**	Inter-option Synchronization Control		0 to 2	0
Pr **91.022**	Inter-option Synchronization Status		Unsigned 8-bit	N/A
Pr **91.023**	Enable CTSync Output Channels		Unsigned 8-bit	1

Table 12-7 Second processor error codes

Error Code	Reason	Drive trip?	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 Pr **81.017** = 0 the value written will be automatically limited and no error will occur.	May	Yes	Yes
45	Invalid synchronization modes	Yes	No	Not Run
46	Unused	N/A	N/A	N/A
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 Pr **90.011**	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 Pr **88.002** 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 *Pr 81.044* on page 35 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/V/W/X/Y) 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 Second Processor as Sync master	Yes	Yes	Yes

Error Code	Reason	Drive trip?	ERROR task?	Prog Halted?
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 Pr **81.024** and network connections.	May	No	No
63	CTNet invalid node ID. Check Pr **81.023**.	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 20 mA.	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 Pr **81.037** = 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 initialization failure	Yes	No	No
74	Over-temperature	Yes	Yes	Yes
75	Hardware unavailable. The user program attempted to access unavailable hardware.	Yes	Yes	Yes
76	Module type cannot be resolved. Module is not recognized.	Yes	No	Not Run
77	Reserved	Yes	Yes	Yes
78		Yes	Yes	Yes
79		Yes	Yes	Yes
80		Yes	Yes	Yes

Error Code	Reason	Drive trip?	ERROR task?	Prog Halted?
81	• Slot selected as the reference or feedback does not contain a position option module • Attempt to change the Reference source or the Feedback source in more than one task. APC internal error. See parameter Pr **81.038**. This may be caused by one of the following: • CAM table too small • A change of too many CAM segments has occurred at the CAM table input • CAM is selected but size is zero • CAM absolute mode selected and Reset Index or Reset Position in segment is out of range	May	Yes	Yes
82	Communications to drive faulty.	May	Yes	Yes

Table 12-8 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

Index

C

Communications 26, 58
Connections 13, 15
CTNet 13, 14, 18
CTSync 85

D

Date code format 9
Diagnostics 92
Digital I/O 41
Downloading 22
DPL 20, 66, 73

E

Error Codes 93

F

Firmware 98
Freeze 81

I

Installation 11

M

Mapping parameters (fieldbus) 65
Marker 81, 83
Migration Guide 99
Modbus ASCII 64
Modbus-RTU 61
Motion Engine 88

P

Parameters 23, 71
PLC Registers 37, 71

Q

Quick Reference 102

R

RS485 ..... 15, 18
Run-time Errors 92

S

Safety 6
SyPTPro 18, 19, 20

T

Tasks 67
Termination 14, 17
Timer 38
Trip Codes 92

U
UDFB 78

V
Variables 69
Virtual Master 90

0478-0009-02