EUROTHERM
DRIVES

RS485 Communications Interface

Technical Manual

HA463560U001 Issue 4

Compatible with Version 2.x Software

Safety Information

Please read this information BEFORE installing the equipment.

Intended Users

This manual is to be made available to all persons who are required to install, configure or service equipment described herein, or any other associated operation.

The information given is intended to highlight safety issues, and to enable the user to obtain maximum benefit from the equipment.

Application Area

The equipment described is intended for industrial motor speed control utilising AC induction or AC synchronous machines.

Personnel

Installation, operation and maintenance of the equipment should be carried out by qualified personnel. A qualified person is someone who is technically competent and familiar with all safety information and established safety practices; with the installation process, operation and maintenance of this equipment; and with all the hazards involved.

REFER TO YOUR MAIN PRODUCT MANUAL FOR SPECIFIC SAFETY INFORMATION ABOUT THE DEVICE YOU ARE CONTROLLING

Cont.2

Contents

Cont.3

Contents

RS485 TECHNOLOGY OPTION

A System Overview

The RS485 Technology Option provides a serial data port, allowing VSDs (variable speed drives) to be linked to form a network. Using a PLC/SCADA or other intelligent device, this network can be continuously controlled to provide supervision and monitoring for each VSD in the system.

With each unit under local control, the central supervisor performs only periodic setpoint updating, control sequencing and data collection.

In the system, the PLC/SCADA supervisor acts as the Master, and the VSD as the Slave.

The network of VSDs can be set-up using just one unit's MMI/Operator Station, or connection to ConfigEd Lite (or other suitable PC programming tool).

Advantages with this type of control system

1. Multi-wire analog transmission from a central programmable controller is replaced by a bussed digital system using serial data transmission over differential twisted-pair wires.
2. Digital transmission is fundamentally less noise-prone than analog methods, and the accuracy of the transmitted data is unaffected by the transmission medium. The use of intelligent devices at either end of the data link allows error checking to be used. This virtually eliminates the effects of electrical noise on data integrity. It is therefore possible to issue setpoints to drives with much higher accuracy using this method.
3. The communication standard used allows up to 32 drives to be connected to a single link which can be driven from a computer serial port. Additional drives can be readily accommodated through additional ports. Most computers are equipped with RS232 serial ports which can be easily converted to accommodate the RS485 standard. Modules are available from Eurotherm Drives to make this conversion.
4. The chosen standard and protocol are compatible with other Eurotherm Group products. Temperature controls, process controls, data loggers and drives can communicate easily with a common supervisory system.

Protocols

EI Bisynch ASCII/Binary

These communications protocols come under the heading of Binary Synchronous Communications Data Link Control (BSCDLC).

This is all part of an internationally recognised ANSI standard protocol called BISYNCH (Binary Synchronous) and is known by the abbreviation x3.28.

They are widely used by manufacturers of computers, computer peripherals, and communications equipment.

EI BISYNCH, the specific form of communication used, corresponds with the following full American National Standard definition:

- ANSI Standard: x3.28, Revision: 1976
- Establishment and Termination Control Procedures Sub-category 2.5:
  Two-way Alternate, Non-switched Multi-point with Centralised Operation & Fast Select
- Message Transfer Control Procedure Sub-category B1:
  Message Associated Blocking with Longitudinal Checking & Single Acknowledgement

This is known by the abbreviation ANSI - x3.28 - 2.5 - B1.

MODBUS RTU

The MODBUS RTU (Remote Terminal Unit) protocol is an efficient binary protocol in which each eight-bit byte in a message contains two four-bit hexadecimal characters. Each message must be transmitted in a continuous stream.

Further Reading

Manual HP022047C: Eurotherm International BISYNCH Communications Handbook.

Product Features

- Suitable for use with:
  - 584SV software version 4.x onwards
  - 590+ software version 5.x onwards
    - 590+DRV software version 5.x onwards
    - 605A & B software version 4.x onwards
    - 605C software version 4.x onwards
- Hardware self-test
- Connection using shielded, twisted-pair cable
- Configured using Function Block inputs
- Diagnostics using Function Block outputs
- Either 2-wire or 4-wire operation
- Software-selectable Baud Rate
- Software-selectable Slave Address
- Direct tag access for all parameters

Product Code

The Eurotherm Drives' product is fully identified using an alphanumeric code which records how the product was assembled, and its various settings when despatched from the factory.

The Technology Option can be supplied with the drive product, or supplied separately:

Product	Product Code when supplied with the Drive	Product Code when supplied separately
584SV	**584SV**/xxxx/xxx/xxxx/xx/xxx/**EI00**/xx/xxx/xxx	AH463469U001 - plug-in card
590+	**590P**/xxxx/xxx/xxxx/xx/xxx/**EI00**/xxx/xxx	6055/EI00/00 - plug-in Technology Box
590+DRV	**955+**/x/x/xxxx/xxx	6055/EI00/00 - plug-in Technology Box
605A & B	**605**/xxx/xxx/x/x/xxx**2**/xx/xxx	6053/EI00/00 - plug-in Technology Box
605C	**605C**/xxxx/xxx/xxxx/xx/xxx/**EI00**/xx/xxx/xxx	6055/EI00/00 - plug-in Technology Box

Installation

WARNING!

Before installing, ensure that the drive and all wiring is electrically isolated and cannot be made "live" unintentionally by other personnel.

Wait 5 minutes after disconnecting power before working on any part of the system or removing the covers from the Drive.

The RS485 Technology Option is provided in one of two forms:

1. A plug-in Technology Box
2. A board-mounted Technology Card (584SV only)

It can be operated as a 2-wire or 4-wire system.

- A 2-wire system can only be used in a network in which all devices use their tri-state capability. Data flow is restricted, i.e. transmit and receive cannot be simultaneous (half duplex).
- A 4-wire system is suitable for use on a network in which the Master does not have or use its tri-state capability. It permits simultaneous transmit and receive (full duplex).

The driver in an RS485 system has tri-state capability (i.e. its output can be disabled) which allows multiple transmitters to be connected to the same bus. RS485 thus supports "multi-drop" operation. In multi-drop systems there is always one device which is a "Master" and which sends messages to or requests data from the "Slaves". A Slave never initiates a communication.

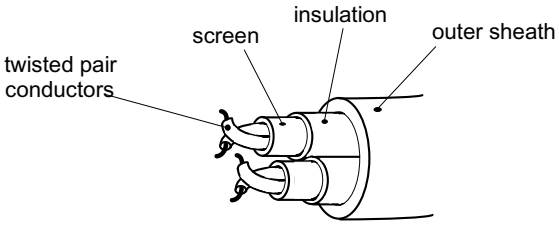
Note: It is possible to make serial communications operate without adhering to the following recommendations, however, the recommendations will promote greater reliability.

PLC/SCADA Supervisor (4-wire only)

If possible, avoid using a PLC/SCADA supervisor which take its transmitter to a high impedance state (tri-state) when idling. If it is unavoidable, then it is essential to use properly screened cable.

Cable Specification

Use cable which has two twisted pairs, with each pair individually screened as shown. The characteristic impedance should be in the range 100 to 165 Ohms.



Recommended Cable Specification	
Characteristic Impedance	100-165Ω at 3-20MHz
Cable Capacitance	<30pF/m
Core Diameter	0.34mm² (22 AWG)
Cable Type	Twisted pair cable
Resistance	<110Ω/km
Shielding	Copper braid, or braid & foil

Note: Belden B3079A cable meets the above specification, but there are others.

Cable Routing

Daisy chain one drive to the next. The supervisor should be at one end of the run. Avoid spurs.

Earthing/Grounding

Connect the screens of both pairs of wires to ground at the supervisor. If possible, connect the supervisor's transmitter/receiver 0V reference to earth. Connect all screens as shown in the following diagrams.

User Connections to the Main Serial Port (P1)

The serial port on the Option allows the following RS485 links to be made.

	RS485	
Electrical Connections	4-wire differential	2-wire differential
Number of transmitters and transceivers allowed per differential pair of wires	32 drivers 32 receivers	32 transceivers
Maximum cable length	4000ft/1200 metres	

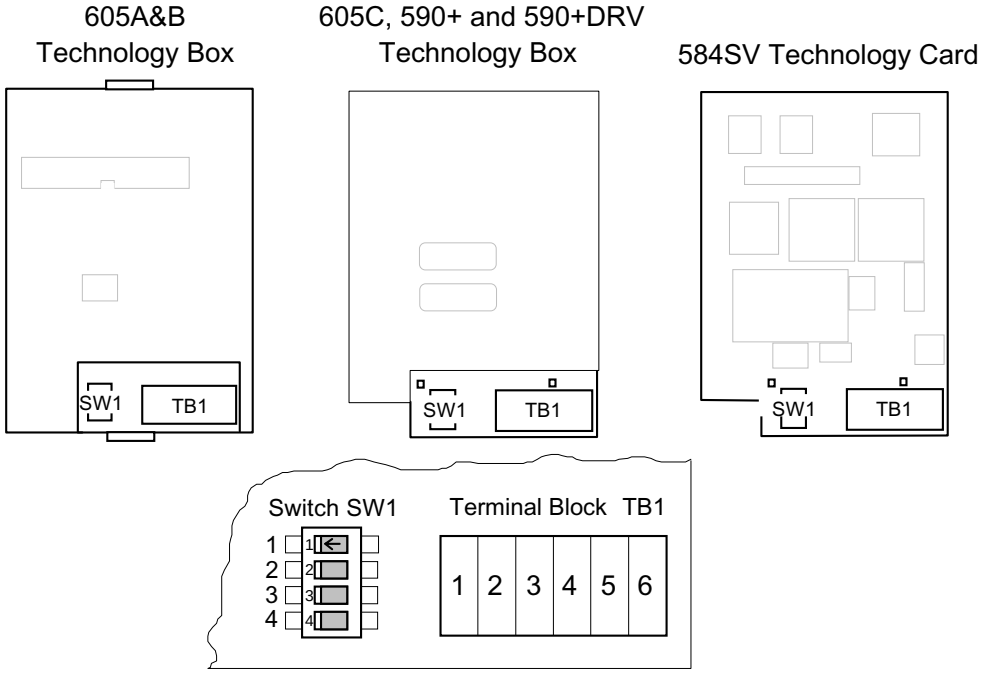
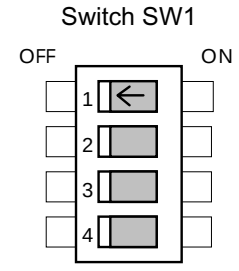


Figure 1 Option showing Terminal Block TB1 and DIL Switch SW1

DIL Switch (SW1) Settings

Set this switch to select 2-wire or 4-wire operation, and to switch in a terminator for the last drive in the system.



Switch	Status	Description
1	OFF ON	4-wire (default) 2-wire
2	OFF ON	Terminator out (default) Terminator in
3 & 4	not used	

Terminators

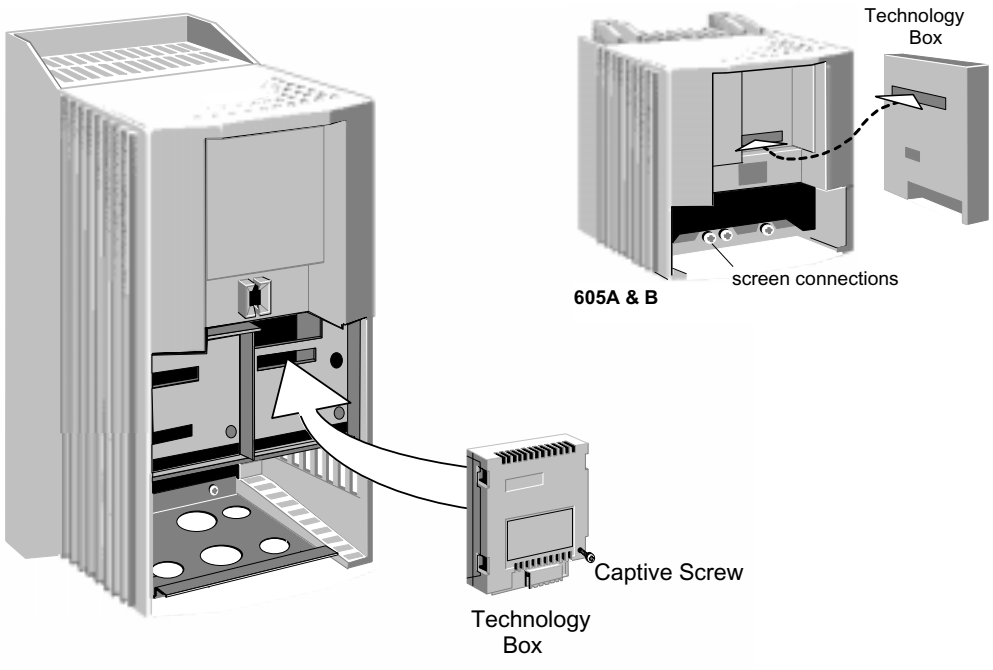
- The unit logically furthest from the supervisor must have switch 1 set to ON.
- All other units in the system must have switch 1 set to OFF.

The supervisor's receiver input should also have a terminating resistor, chosen to match the characteristic impedance of the cable, typically 100 to 165 Ohms.

Terminal Block (TB1) Connections

Terminal No.	2-Wire Designation	4-Wire Designation
1	not used	TXB
2	not used	TXA
3	0V	0V
4	Cable Screen (except 605A & B)	Cable Screen (except 605A & B)
5	RXB/TXB	RXB
6	RXA/TXA	RXA

Fitting and Connecting to the Technology Box



605C, 590+, 590+DRV (590+ 15A unit illustrated)

Figure 2 Plug-in Technology Boxes

> WARNING!
> Ensure that all wiring is isolated.

IMPORTANT: Remember to set the switch positions on the DIL switch, SW1.

The Technology Option plugs into the right-hand position on the front of the drive, or in place of the Operator Station/blank cover (605A & B only).

It can be used with the Operator Station fitted, but for the 605A & B unit you must mount the Operator Station remotely using the Panel Mounting Kit with connecting lead (6052). The connecting lead enters the 605 A & B drive through the gland plate.

- Remove the terminal cover and screws.
- On the 605A & B unit, plug the ribbon cable into the back of the Technology Box and into the socket on the drive.
- Click the Technology Box into place in the recess on the front of the drive. If provided, secure in position by tightening the captive screw on the bottom right hand corner of the Option.
- Make all user wiring connections. Refer to the Wiring Diagrams**.**
- Re-fit the terminal cover securely with the screws.

Connecting to the Technology Card (584SV only)

The option is supplied as a “Technology Card”. This is factory-fitted to the control board inside the drive.

When connecting to the Technology Card, observe static control precautions.

WARNING!
Ensure that all wiring is isolated.

584SV Types 4, 5 & 6

- Remove the terminal cover and screws.

IMPORTANT: Remember to set the switch positions on the DIL switch, SW1.

- Make all user wiring connections. Refer to the Wiring Diagrams**.**
- Re-fit the terminal cover securely with the screws.

584SV Types 7, 8, 9 & 10

- Remove the bottom front cover.

IMPORTANT: Remember to set the switch positions on the DIL switch, SW1.

- Make all user wiring connections. Refer to the Wiring Diagrams**.**
- Re-fit the terminal cover securely with the screws.

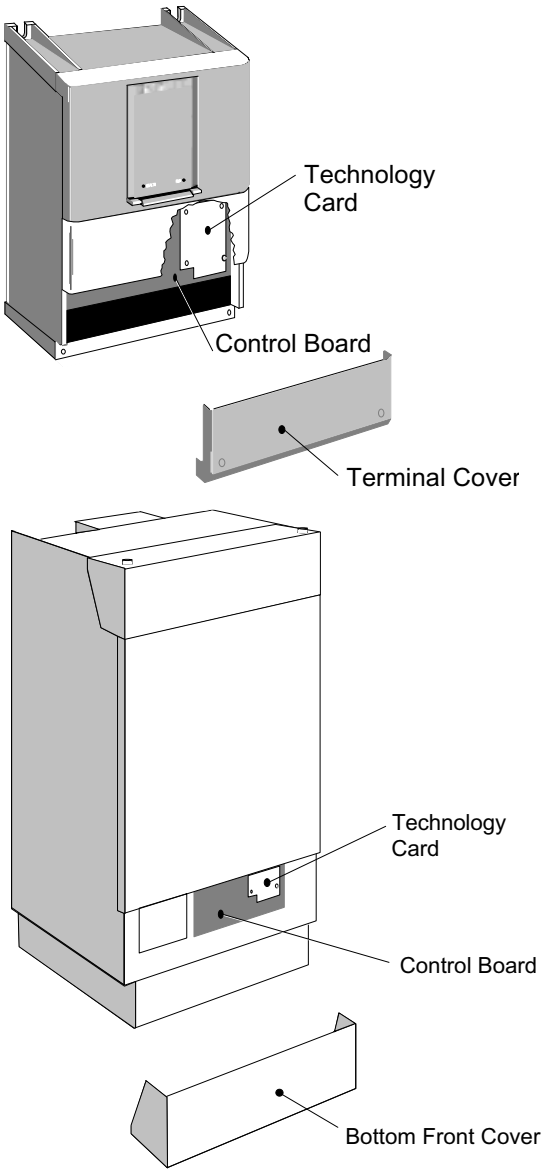


Figure 3 Plug-in Technology Cards

Wiring Diagrams

EI Bisynch ASCII/Binary only

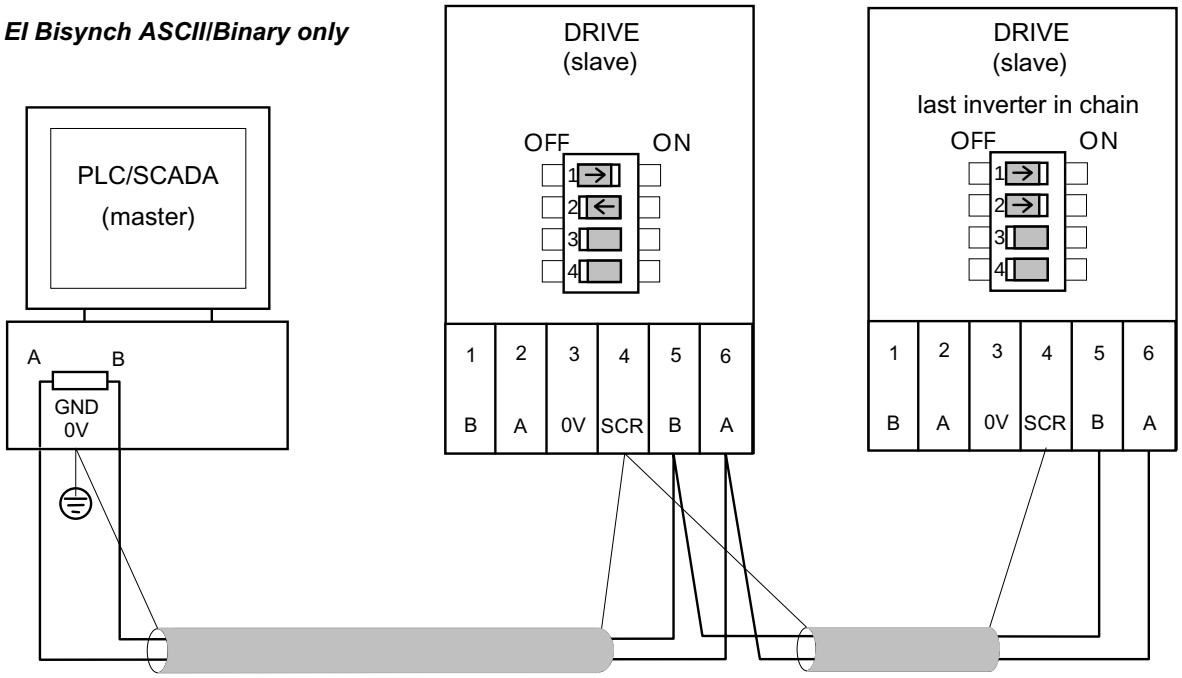


Figure 4 2-Wire Wiring Diagram for the 584SV, 590+, 590+DRV and 605C Drive (EI Bisynch ASCII/Binary only)

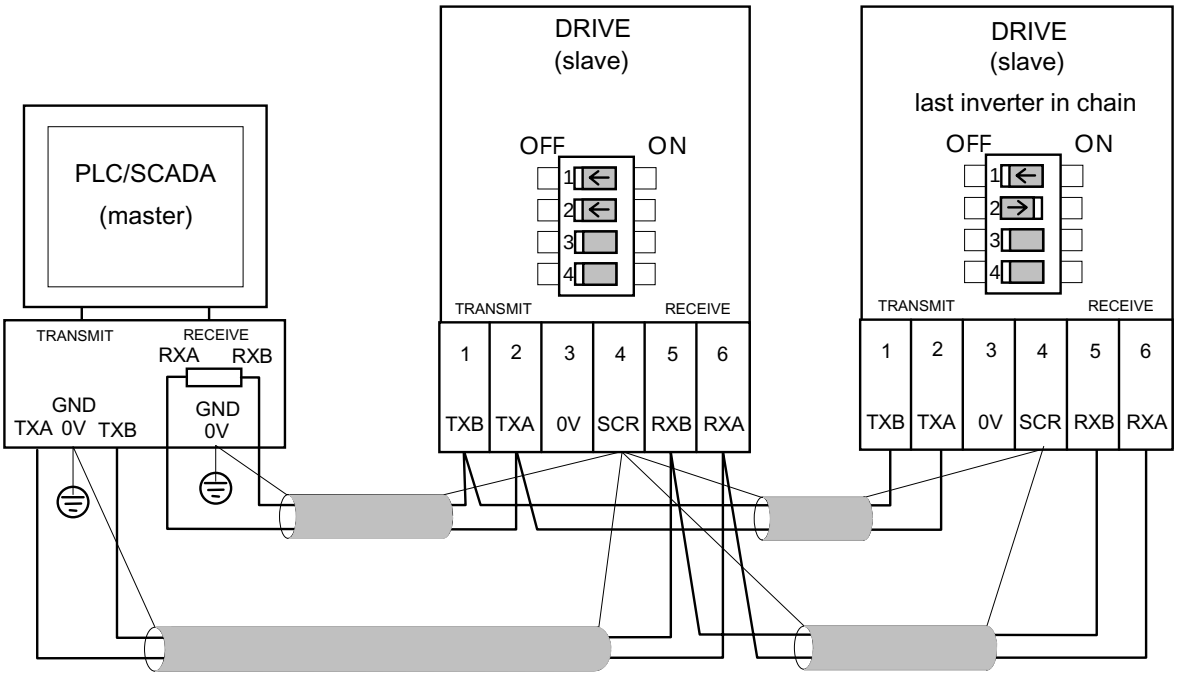


Figure 5 4-Wire Wiring Diagram for the 584SV, 590+, 590+DRV and 605C Drive

EI Bisynch ASCII/Binary only

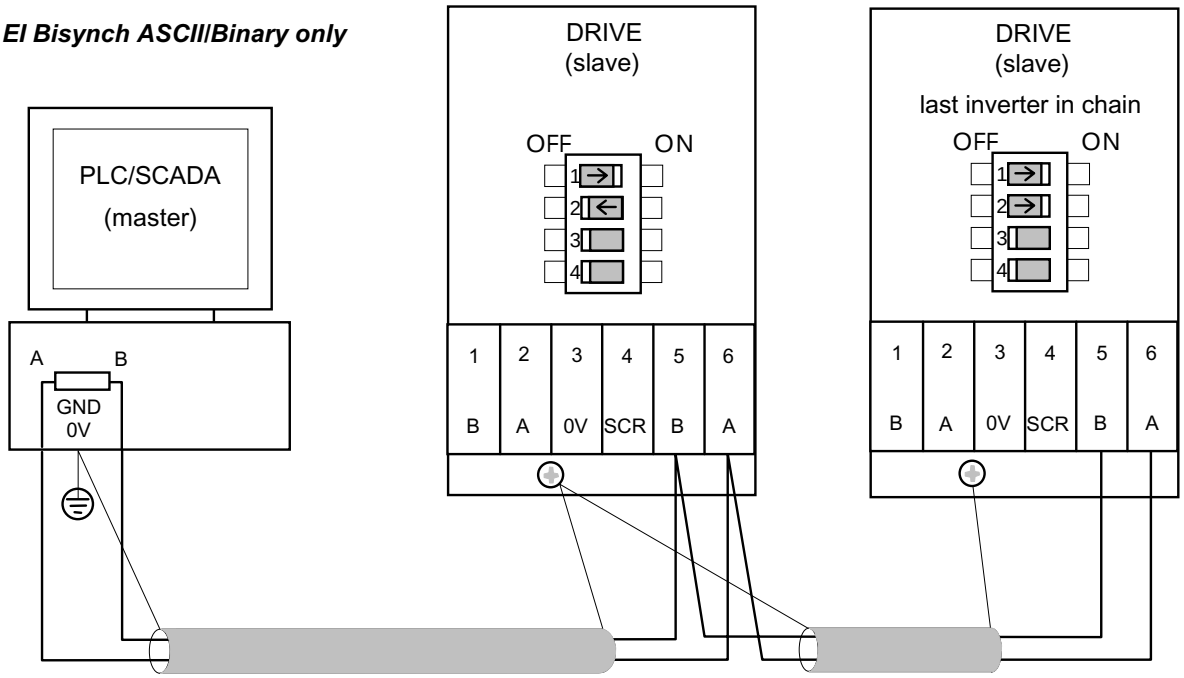


Figure 6 2-Wire Wiring Diagram for the 605A & B Drive (EI Bisynch ASCII/Binary only)

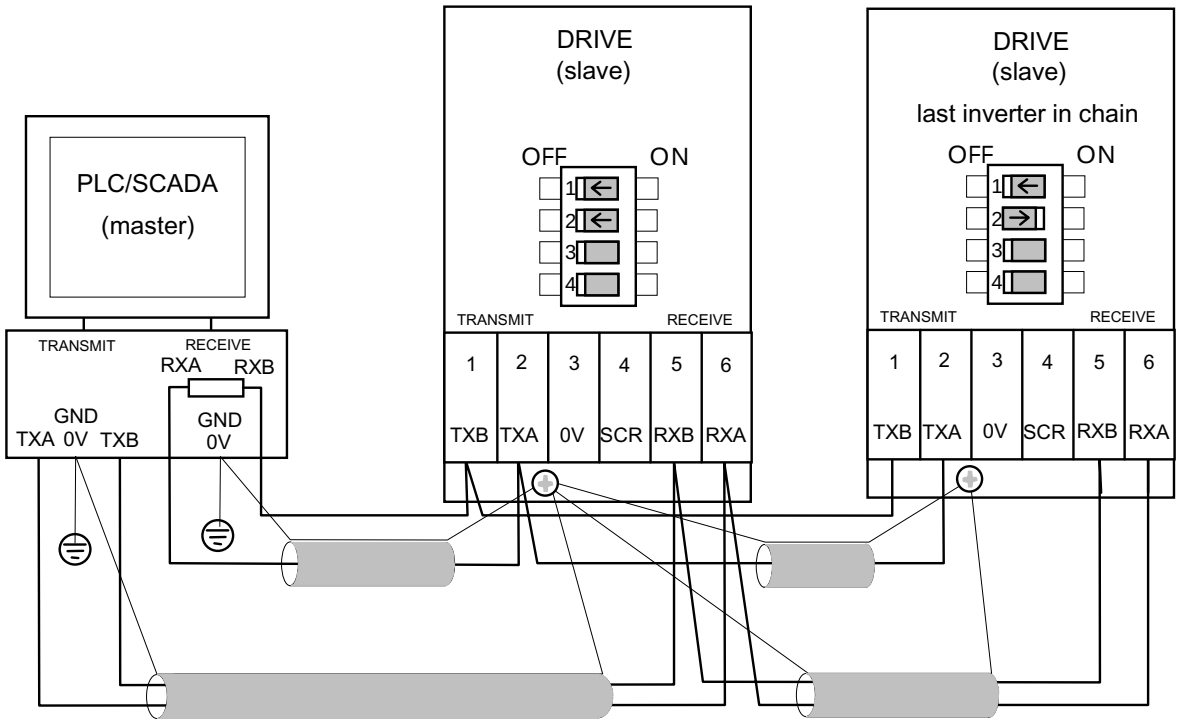


Figure 7 4-Wire Wiring Diagram for the 605A & B Drive

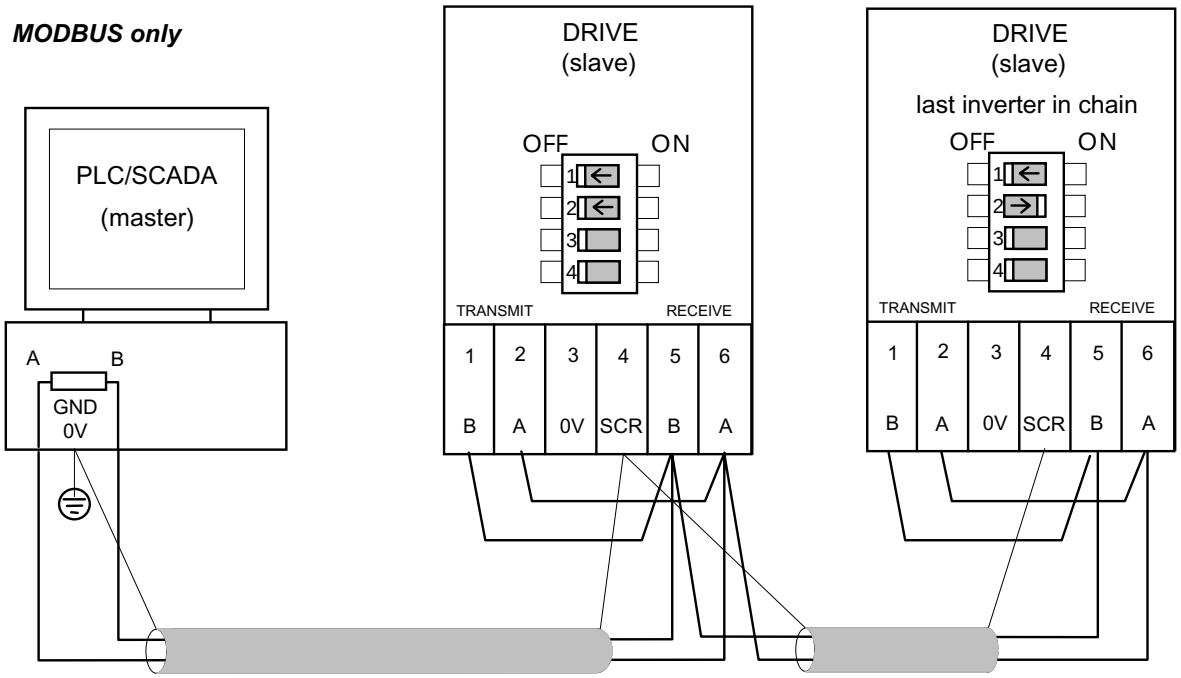


Figure 8 2-Wire Wiring Diagram for the 584SV, 590+, 590+DRV and 605C Drive (Modbus only)

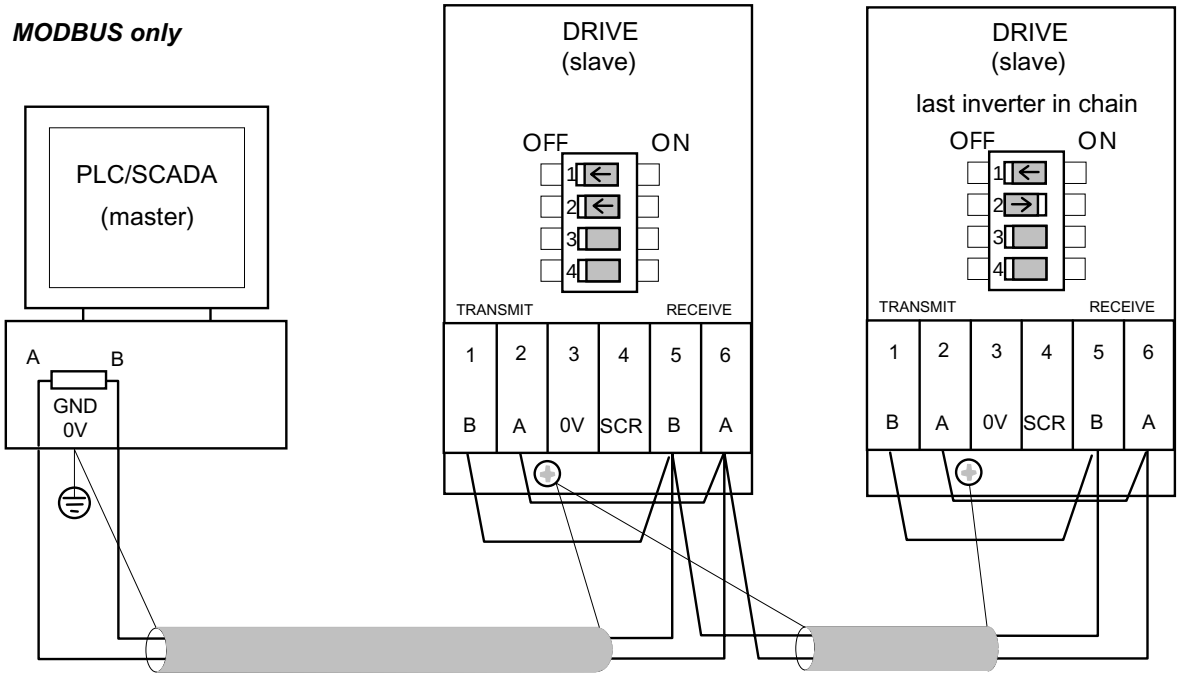


Figure 9 2-Wire Wiring Diagram for the 605A & B Drive (Modbus only)

Initial Check for Connection

With the correct connections to the active PLC/SCADA supervisor, the MODULE LED will be ON continuously and the NETWORK LED will indicate the Idle state with a short flash.

ON	●	MODULE LED
SHORT FLASH	○●	NETWORK LED

Understanding the LED Indications

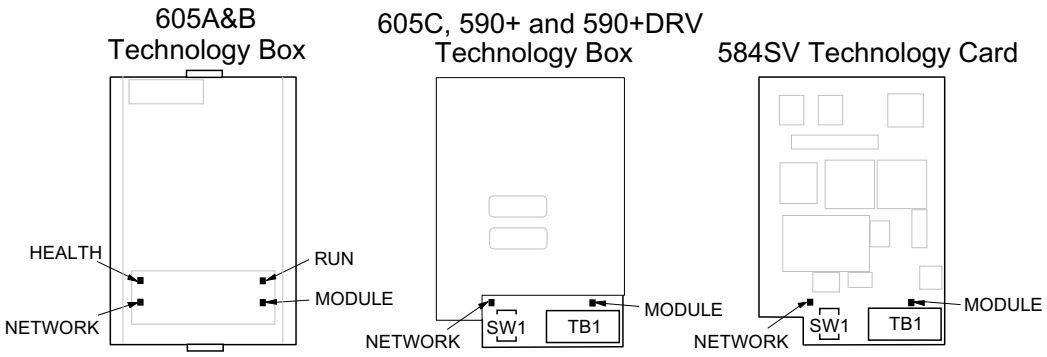


Figure 10 Technology Option LEDs

HINT:

The general rule for LED indications is

"ON IS GOOD, OFF IS BAD"

Health and Run LEDs

605A & B Technology Box

These LEDs reproduce the indications of the LEDs on the 605 that are hidden by the fitting of the Technology Box.

584SV Technology Card and 605C, 590+, 590+DRV Technology Box

The board does not have its own Health or Run LEDs. The LEDs are either on the Operator Station or blank cover.

Module LED

This indicates the set-up state of the Technology Box. The states indicated are those produced by the FAULT parameter of the TEC OPTION function block.

Module LED Indication		FAULT Parameter Indication	Description
OFF	○	SELF TEST	Initialising
SHORT FLASH	○●	HARDWARE	Hardware fault
FLASH	○●	TYPE MISMATCH	Wrong type or disabled
LONG FLASH	●○	PARAMETER	Set-up fault, parameter values out-of-range
ON	●	NONE	Valid set-up, ready for external communications

Network LED

This indicates the state of the connected network.

Network LED Indication	Description
OFF	Not ready for external communications or Idle with inverted RX line
SHORT FLASH	Idle with correct RX line.
FLASH	Activity on RX line (within last second)
LONG FLASH	Valid character received (within last second)
ON	Addressed (within last 5 seconds)

Note: The NETWORK LED can only be in the ON state when the MODULE LED is ON continuously, indicating that the Option is ready for external communications.

Configuring the Drive

MMI Menu Map
Non-specific MMI view

1 SETUP PARAMETERS
2 FUNCTION BLOCKS
3 SERIAL LINKS
4 TEC OPTION
- TEC OPTION TYPE
- TEC OPTION IN 1
- TEC OPTION IN 2
- TEC OPTION IN 3
- TEC OPTION IN 4
- TEC OPTION IN 5
- TEC OPTION FAULT
- TEC OPTION VER
- TEC OPTION OUT 1
- TEC OPTION OUT 2

MMI Menu Map
EI BISYNCH ASCII
MMI view

1 SETUP PARAMETERS
2 FUNCTION BLOCKS
3 SERIAL LINKS
4 TEC OPTION
- TEC OPTION TYPE
- PROTOCOL
- BAUD RATE
- GROUP ID (GID)
- UNIT ID (UID)
- ERROR RESET
- FAULT
- VERSION
- ERROR CODE
- ERROR COUNT

SERIAL LINKS is at Menu Level 1 for the 590+ and 590+DRV.

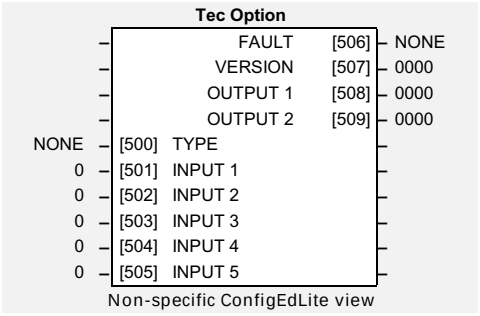
Using the Operator Station (MMI) or other suitable PC programming tool, the TEC OPTION function block requires configuring before the RS485 option can be used.

The parameter names/functions in the TEC OPTION function block are inter-dependant and will change with different parameter values and the various Technology Options that can be fitted.

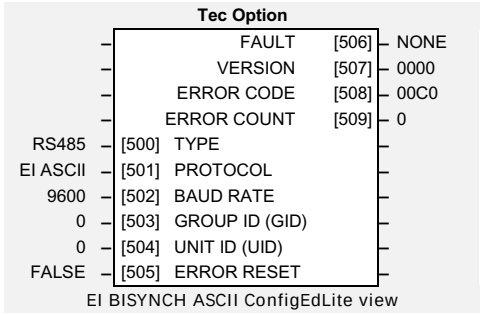
The top function block diagram shows the ConfigEd Lite parameter names, which are also displayed on the MMI if no Technology Option is fitted or an incorrect TYPE is selected for the fitted Technology Option.

ConfigEd Lite is Eurotherm Drives' Windows-based block programming software.

When the TYPE parameter is set to display RS485, the function block parameters take on new identities, as shown in the lower Function Block diagram.



Non-specific ConfigEdLite view



EI BISYNCH ASCII ConfigEdLite view

Selecting EI BISYNCH ASCII

(Select Advanced view level on the Operator Station and view the TEC OPTION function block).

- Select RS485 in the TYPE parameter
- Select EI ASCII in the PROTOCOL parameter
- Select the Baud Rate
- Enter a GID address (if required)
- Enter a UID address (if required)
- Check the FAULT parameter for error messages and rectify if necessary

When setting values for parameters from ConfigEd Lite (or other suitable PC programming tool) you are able to select any value in the parameter's range, i.e. -32768 to 32767. If the value is incorrect, i.e. it doesn't correspond to a value that can be set using the MMI, then the FAULT output parameter will be set to PARAMETER.

MMI Parameter Descriptions for EI Bisynch ASCII

TYPE *Range: Enumerated - see below*

Selects the type of Technology Option. The TYPE parameter is automatically set when defaults are loaded if a Technology Option is present.

Enumerated Value : Technology Option

0 : NONE
1 : RS485
2 : PROFIBUS DP
3 : LINK
4 : DEVICENET
5 : CANOPEN
6 : LONWORKS
7 : TYPE 7

PROTOCOL *Range: Enumerated - see below*

Selects the protocol to be used.

Enumerated Value : Protocol

0 : EI ASCII (default)
1 : EI BINARY
2 : MODBUS RTU

BAUD RATE *Range: Enumerated - see below*

Selects the Baud Rate.

Enumerated Value : Baud Rate

0 : 300
1 : 600
2 : 1200
3 : 2400
4 : 4800
5 : 9600 (default)
6 : 19200

GROUP ID (GID) *Range: 0 to 7*

The Eurotherm protocol group identity address.

UNIT ID (UID) *Range: 0 to 15*

The Eurotherm protocol unit identity address.

ERROR RESET *Range: FALSE/TRUE*

When TRUE, clears the ERROR CODE parameter (setting it to 00C0) and sets the ERROR COUNT parameter to zero.

FAULT *Range: Enumerated - see below*

The fault state of the Technology Option.

0 : NONE	no faults
1 : PARAMETER	parameter out-of-range
2 : TYPE MISMATCH	TYPE parameter not set to RS485
3 : SELF TEST	hardware fault - internal
4 : HARDWARE	hardware fault - external
5 : MISSING	no option fitted

Also refer to “Module LED”, page 10.

VERSION *Range: 0x0000 to 0xFFFF*

The version of the Technology Option card. If no option is fitted then the version is reset to zero.

ERROR CODE *Range: 0x0000 to 0xFFFF*

Displays the last error as a hexadecimal code. Refer to“Last Error Code (EE)”, page 29 for a list of codes.

ERROR COUNT *Range: 0 to 9999*

Increments each time an error is detected.

Note: will stop counting at 9999 (see ERROR RESET).

Configuring the PLC/SCADA Supervisor

By referring to the Parameter Specification Table in the main Product Manual, you can enter the parameter information you require.

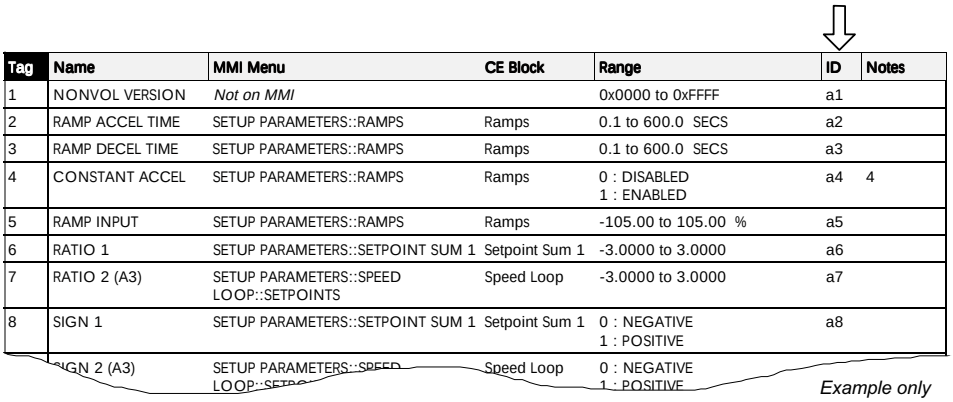
It provides the information in the following way:

Type

The first page of the Parameter Specification Table chapter details parameter types.

ID/MN

The ID or MN column provides the parameter mnemonic (of the tag number).



Tag	Name	MMI Menu	CE Block	Range	ID	Notes
1	NONVOL VERSION	*Not on MMI*		0x0000 to 0xFFFF	a1	
2	RAMP ACCEL TIME	SETUP PARAMETERS::RAMPS	Ramps	0.1 to 600.0 SECS	a2	
3	RAMP DECEL TIME	SETUP PARAMETERS::RAMPS	Ramps	0.1 to 600.0 SECS	a3	
4	CONSTANT ACCEL	SETUP PARAMETERS::RAMPS	Ramps	0 : DISABLED 1 : ENABLED	a4	4
5	RAMP INPUT	SETUP PARAMETERS::RAMPS	Ramps	-105.00 to 105.00 %	a5	
6	RATIO 1	SETUP PARAMETERS::SETPOINT SUM 1	Setpoint Sum 1	-3.0000 to 3.0000	a6	
7	RATIO 2 (A3)	SETUP PARAMETERS::SPEED LOOP::SETPOINTS	Speed Loop	-3.0000 to 3.0000	a7	
8	SIGN 1	SETUP PARAMETERS::SETPOINT SUM 1	Setpoint Sum 1	0 : NEGATIVE 1 : POSITIVE	a8	
	SIGN 2 (A3)	SETUP PARAMETERS::SPEED LOOP::SETPO	Speed Loop	0 : NEGATIVE 1 : POSITIVE		

Example only

ASCII Communications

Data can be transferred in two formats: ASCII or Binary, i.e. a value of 100 is represented by the three ASCII characters 1, 0, 0; or by the Binary equivalent of 100 in 16 bit data format, 0064 Hex.

What Information Can I Transfer?

The data transfer sequence in the ASCII mode offers the following facilities:

i) Parameter enquiry (known as polling)

a. Single Parameter Poll
b. Continuous Polling of a Parameter
c. Sequential Polling (fast polling down the parameter list)

ii) Setting parameters (known as selection)

a. Single Parameter Selection
b. Continuous Selection of a Parameter
c. Sequential Selection (fast selection down the parameter list)

Note: For examples of all the above refer to "Transferring Data - ASCII Example Messages", page 23.

How is the Information Transferred?

There are two types of data transfer message:

1. Reading information from the Drive
2. Writing information to the Drive

In both cases the supervisor must have an established connection with the device, which will then respond. The role of master and slave exchanges during the transfer.

A message consists of a sequence of characters which we identify as

- Control Characters
- Instrument Address
- Parameter Mnemonic
- Data

Note: Refer to "EI Bisynch ASCII Message Protocol" page 18, where these four types of character are discussed in detail.

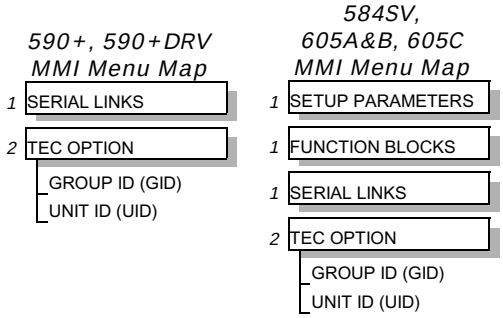
The following events take place in transmitting a successful message:

- Establish Connection
- Enquiry or Set Parameter
- Response
- Further Transmission and/or Termination

Establish Connection

Connection is established with a particular device by sending its two-digit address (i.e. INSTRUMENT ADDRESS as above).

You can set the address in the TEC OPTION menu.



Enquiry or Set Parameter

The message is either an enquiry (reading information from the Drive), or a message to set a parameter (writing information to the Drive).

Response to a `Set Parameter' Message

The Drive will respond to a Set Parameter message in one of three ways:

1. Positive Acknowledgement (ACK)
2. Negative Acknowledgement (NAK)
3. No Reply: Under certain circumstances the supervisor may not receive a reply from the Drive. This could be due to any of the following reasons:

- Group/Unit address identifiers not recognised.
- An error (e.g. parity) is found in one or more of the characters up to and including (ENQ).
- Communications loop failure perhaps due to noise or wrong Baud Rate being selected.
- Hardware failure.
- Serial link is disabled on the Operator Station.

In these cases the supervisor should be programmed to "time-out", i.e. wait for a response for a short time (160 msec minimum) before trying again.

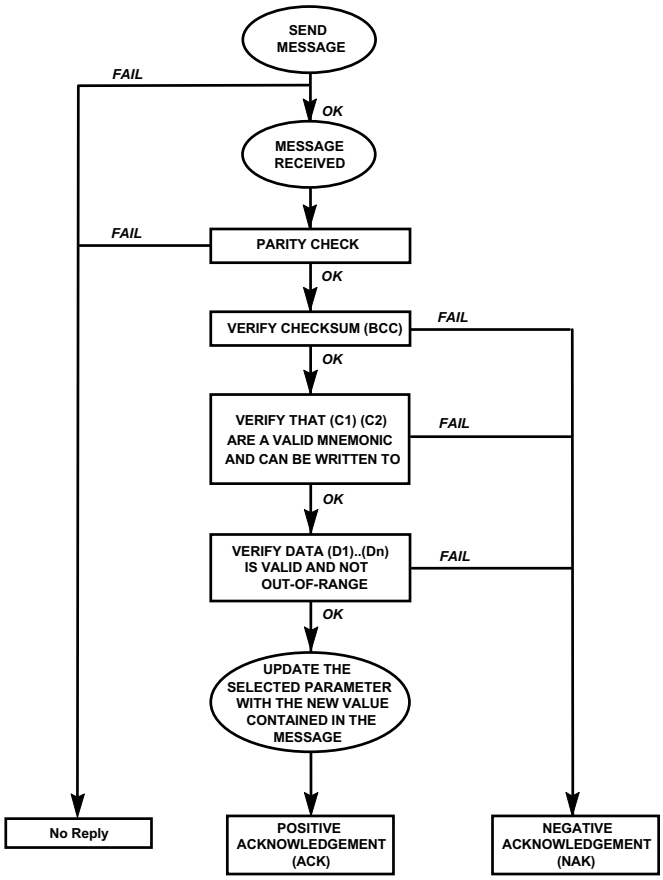


Figure 11 Drive Response Sequence to an ASCII Selection Message

Further Transmission and/or Termination

Further Transmission

If the supervisor still has an established connection with the device, you can repeat the previous message without re-establishing connection.

In both cases, writing to or reading from the device, you can use this to re-select the previous parameter or to select the next parameter in the parameter list. Refer to “Transferring Data - ASCII Example Messages”, page 23 for further explanation.

Termination (EOT)

If you wish to terminate connection with a particular device and establish connection with another, send the 'Establish Connection' sequence preceded by the (EOT) control character, (End Of Transmission).

The (EOT) character resets all devices on the data link to be responsive to the next four characters, i.e. the (GID)(GID)(UID)(UID) address.

- In 4-wire operation, an (EOT) can be sent at any time, including when the device has Master status.
- In 2-wire operation, an (EOT) can only be sent when the supervisor has Master status.

Programmer's Information

ASCII (American Standard Code for Information Interchange)

The RS485 Option communicates using ASCII, a binary code which represents letters, digits, and control signals (collectively called characters).

The code, originated by the American National Standards Institute (ANSI), has become a world-wide standard for information interchange. It uses a seven bit binary word to represent all the letters, digits, punctuation marks and control signals.

Handling of Numerical Data

(Format 21 - Free Format Numeric)

Numerical Data is transferred as a string of characters. The length of the string required to transmit the data value is determined by the value itself, however, no leading zeros are added to pad out the string length and trailing zeros are omitted, i.e.

1.00, 1.0, 1. or 1	is converted to	1
-2.20 or -2.2	is converted to	-2.2

Handling of Status Information

(Format 23 - Hexadecimal)

Status Information is transmitted by first encoding the data into a hexadecimal format. The length of a string is then determined by the number of characters in the encoded data. The hexadecimal data is preceded by a '>' sign to differentiate it from numerical data.

Note: Hexadecimal refers to the common practice of counting to the base of 16 in computing rather than the base of 10. The sixteen `numbers' used being 0 to 9, A to F. Thus an 8 bit byte is represented by two characters in the range 00 to FF, while a 16 bit word is represented by four characters in the range 0000 to FFFF.

Block Check Character (BCC)

This is a checksum value generated by taking the exclusive OR (XOR) of the ASCII values of all the characters transmitted after and excluding (STX) up to and including (ETX). For example, the shaded characters are included in the (BCC) of the following message:

(EOT)	(GID)	(GID)	(UID)	(UID)	(STX)	(C1)	(C2)	(D1)	(D2)	(D3)	(ETX)	(BCC)

Example 5: Set Parameter

For Beginners:

You can calculate this easily by converting the ASCII values to Binary and progressively adding the Binary values together, obeying the following rules:

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +\,1 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \quad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array}$$

Referring to Example 5 on page 27, the calculation of (BCC) becomes:

As Characters	HEX	ASCII	Binary
(C1)	37	7	0 1 1 0 1 1 1
(C2)	31	1	0 1 1 0 0 0 1
			0 0 0 0 1 1 0 (sub-total)
(D1)	33	3	0 1 1 0 0 1 1
			0 1 1 0 1 0 1 (sub-total)
(D2)	30	0	0 1 1 0 0 0 0
			0 0 0 0 1 0 1 (sub-total)
(D3)	2E	.	0 1 0 1 1 1 0
			0 1 0 1 0 1 1 (sub-total)
(ETX)	03	(ETX)	0 0 0 0 0 1 1
(BCC)	28	(	0 1 0 1 0 0 0 (TOTAL)

EI Bisynch ASCII Message Protocol

Transmission Standard	:	RS485
Protocol	:	ANSI-X3.28-2.5-B1
Data Rates	:	300, 600, 1200, 2400, 4800, 9600 or 19200 Baud
Character Format	:	1 start + 7 bit ASCII data + 1 parity + 1 stop bit (10 bits)
Parity	:	Even

The Protocol defines the string or sequence of characters (called a Message) which must be sent between communicating instruments to produce specific responses. The message usually comprises:

- Control Characters
- Instrument Address
- Parameter Mnemonic
- Data

Control Characters

Control Characters are ASCII codes that define actions rather than information. Six ASCII codes are supported:

Keyboard	***HEX***	***ASCII***	
^B	02	(STX)	*Start of Text*
^C	03	(ETX)	*End of Text*
^D	04	(EOT)	*End of Transmission*
^E	05	(ENQ)	*Enquiry*
^F	06	(ACK)	*Positive Acknowledge*
^U	15	(NAK)	*Negative Acknowledge*

Instrument Address

The Drive has a two-digit address, the first digit being the “group” ID number (GID) in the range 0 to 7, the second digit is a “unit” ID number (UID) in the range 0 to F. There are therefore 128 different addresses from 00 to 7F.

The Instrument Address (01 for example) is repeated in the message (i.e. 0011) for security as it is not included in a Checksum.

Parameter Mnemonic

Each parameter in the Drive’s menu system is identified by a unique Tag Number. Information is exchanged across the system by use of a two character Mnemonic that is derived from the Tag Number.

Examples are:

81 : the SETPOINT 1 parameter from the SETPOINTS function block

3b : the I DMD. ISOLATE parameter from the CURRENT LOOP function block

Note: Refer to “ EI Bisynch Binary Parameter Specification Tables”, page 37 for a full list of tag mnemonics. - see the ASCII column.

EI Bisynch ASCII Parameter Mapping

1. EI Bisynch ASCII Prime Set

The following prime set parameters are supported:

Mnemonic	Description	Range (HEX encoding)	Access
II	Instrument Identity	>0605, >0584 or >5900	Read Only
V0	Main Software Version	>0000 to >FFFF	Read Only
V1	Operator Station Software Version	>0000 to >FFFF (>0000 if not fitted)	Read Only
V2	Technology Box Software Version	>0000 to >FFFF	Read Only
EE	Last Error Code	>0000 to >FFFF (Writing any value resets this to >00C0)	Read/Write

2. Command/Status

The following Command/Status parameters are supported:

Mnemonic	Description	Range (Hex encoding)	Access
!1	Command	see below	Write Only
!2	State	see below	Read Only
!3	Save Command	see below	Write Only
!4	Save State	see below	Read Only

!1 : Command

Write-only: used to modify the state of the Inverter and to load configuration data from non-volatile memory.

HEX Value	Description
>7777	Reset Command. Acknowledges failed restore. Loads and saves (590+ does not save) default Product Code and default Configuration (Macro 1).
>0101	Restores Saved Configuration from drive's non-volatile memory.
>0110	Restores Default Configuration (Macro 0) - *not 590+*
>0111	Restores Default Configuration (Macro 1)
>0112	Restores Default Configuration (Macro 2) - *not 590+*
>0113	Restores Default Configuration (Macro 3) - *not 590+*
>0114	Restores Default Configuration (Macro 4) - *not 590+*
>01A9	Restores Default Configuration (Macro 99) - *584SV only*
>4444	Exit Configuration Mode
>5555	Enter Configuration Mode

!2 : State

Read-only: used to determine the major state of the Inverter.

HEX Value	Description
>0000	Initialising. (Powering up)
>0001	Corrupted Product Code and Configuration
>0002	Corrupted Configuration
>0003	Restoring Configuration
>0004	Re-Configuring Mode
>0005	Normal Operation Mode

!3 : Save Command Write-only: used to save the configuration and product code in non-volatile memory.	
HEX Value	**Description**
>0000	Reset Command. Acknowledges (clears) any previous save error.
>0001	Saves Configuration to drive's non-volatile memory.
>0100	Saves Product Code to drive's non-volatile memory.

!4 : Save State Read only: used to determine the progress of a non-volatile saving operation.	
HEX Value	**Description**
>0000	Idle
>0001	Saving
>0002	Failed

3. Tag Access

Each parameter in the Inverter's menu system is identified by a unique Tag Number. Information is exchanged across the system by use of a two character Mnemonic that is derived from the Tag Number.

Note: Refer to the Parameter Specification Table in the main Product Manual for a full list of tag mnemonics - see the ID/MN column. Refer to the Notes column which gives access information about each parameter.

Parameter Mapping

605A&B/605C/584SV Algorithm

m = INT (TagNo/36)	*(INT: the integer part)*
n = TagNo MOD 36	*(MOD: the remainder)*

590+/590+DRV Algorithm

m = INT ((TagNo + 360)/36)	*(INT: the integer part)*
n = (TagNo + 360) MOD 36	*(MOD: the remainder)*

The algorithm to convert between tag number and 2 character mnemonics is:

```
if  m > 9 then
        char_1 = 'a' + (m - 10)
else
        char_1 = '0' + m
end_if
if  n > 9 then
        char_2 = 'a' + (n - 10)
else
        char_2 = '0' + n
end_if
```

The algorithm generates mnemonics containing only the characters '0' to '9' and 'a' to 'z'.

4. PNO Access (590+ and 590+DRV only)

For compatibility with the earlier 590 product, parameters may also be accessed using the ASCII PNO listed in the " EI Bisynch Binary Parameter Specification Tables", page 37. For example, PNO 39 can be accessed with the mnemonic "27".

5. Encoding

Type	Description	Encoding	Comments
BOOL	Boolean	FALSE >00 TRUE >01	Will accept >0 and >1
WORD	16-bit Bitstring	>0000 to >FFFF	Will accept leading zero suppression, except >0
INT	16-bit Signed Integer	-XXXXX. to XXXXX. -XXXX.X to XXXX.X -XXX.XX to XXX.XX -XX.XXX to XX.XXX -X.XXXX to X.XXXX	Leading zeroes suppressed up to digit before decimal point. Trailing zeroes suppressed after decimal point.
ENUM	Enumerated Value (0 to 99)	XX.	Leading zeroes suppressed, except 0.
STRING	Printable characters.	'SSSSSSSSSSSSSSS where S is a printable character	Maximum number of characters is parameter specific.
STAG	Link Source Tag No.	-XXXX. to XXXX.	As INT above.
DTAG	Link Destination Tag No.	XXXX.	As INT above.

Note: *The "." in the above formats is not optional. It must be sent to conform to the EI-BISYNCH standard.*

EI Bisynch ASCII Sequence Diagrams

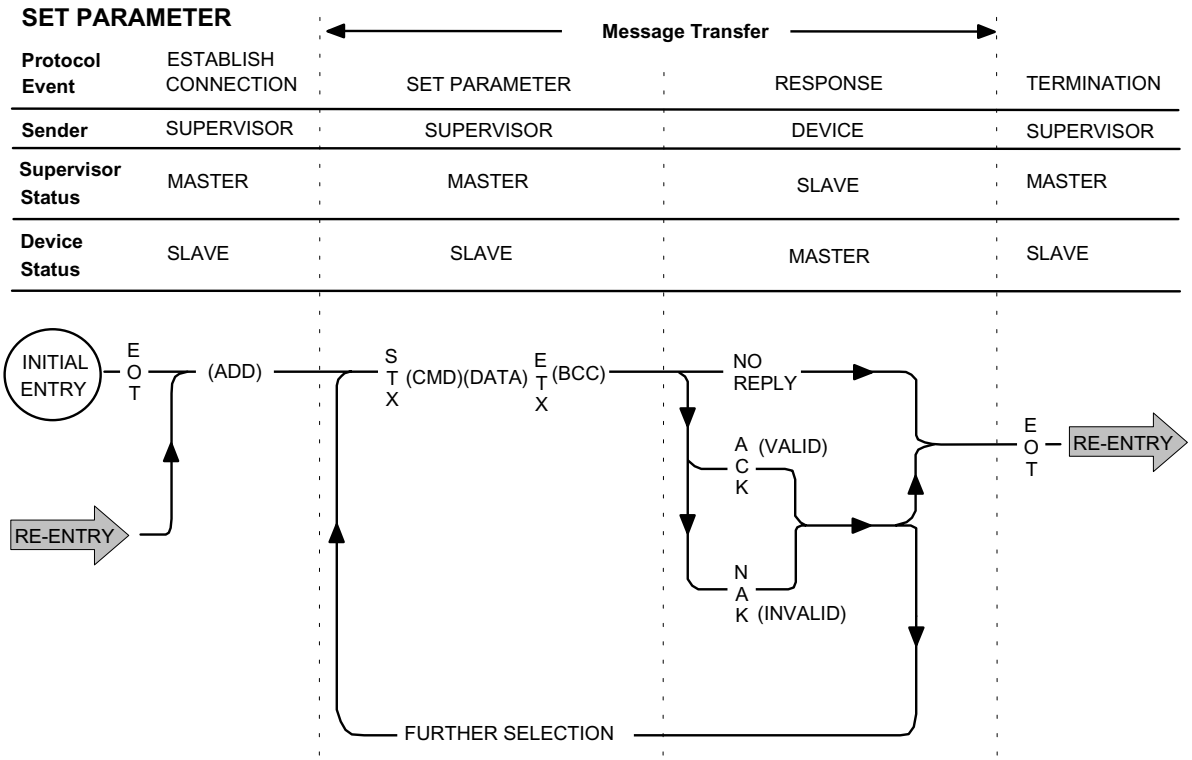


Figure 12 Selection Sequence for Writing Information to the Drive

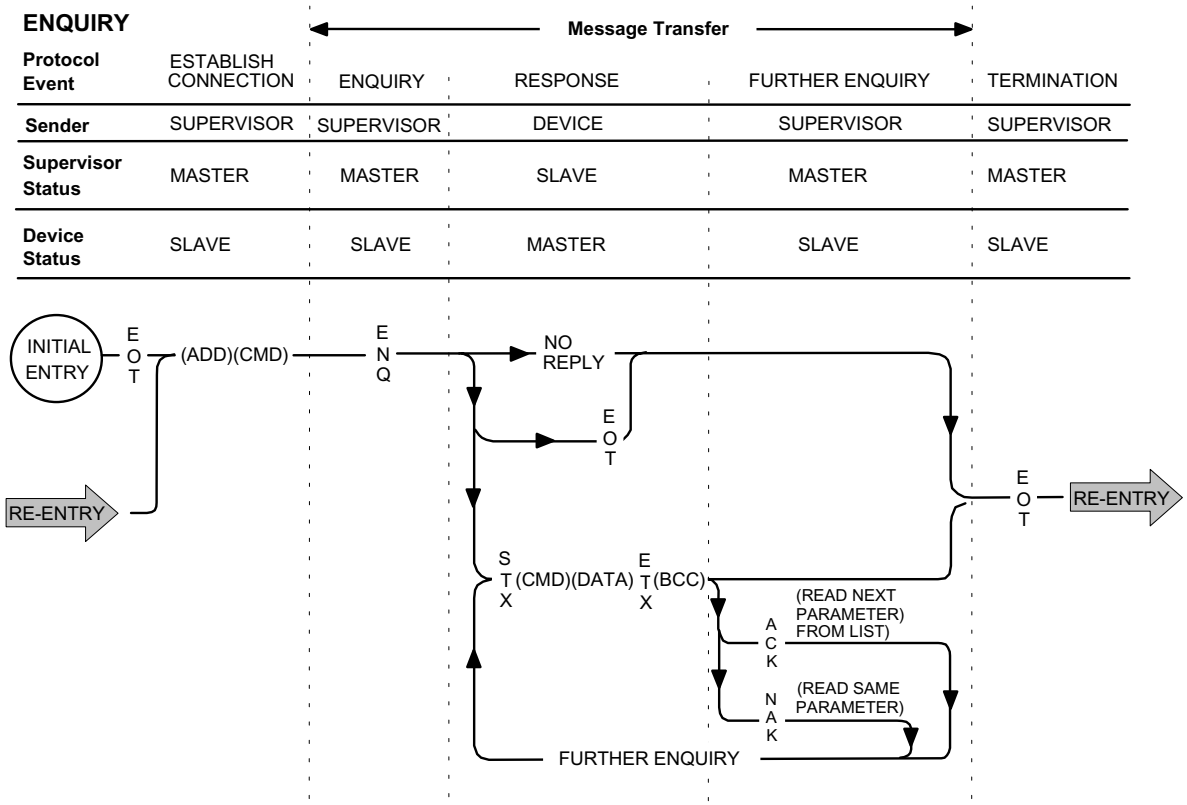


Figure 13 Poll Sequence for Reading Information from the Drive

Transferring Data - ASCII Example Messages

The following examples show how data transfer takes place using the network, they will also help to verify your communications if you have just finished installing the COMMS Option. Many users will not become involved in generating low-level code, but for those experienced in programming, the examples include ASCII, HEX and Control Character information.

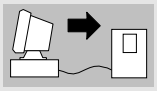
Note: Refer to "Control Character Definitions", page 28 for a more detailed explanation of all control characters.

Example 1: EI Bisynch Prime Set

Note: Refer to "EI Bisynch Binary Parameter Specification Tables", page 37 for a full list of EI BISYNCH Prime Set mnemonics supported.

Using this set of mnemonics, you can enquire about the Drive. For instance, you could enquire about the Instrument Identity:

ENQUIRY



- ***For software users:***

Enter the known address of the Drive (say 01), II, and that it is an enquiry.

- ***For programmers, in ASCII:***

(EOT)	0	0	1	1	I	I	(ENQ)

- ***For programmers, in HEX:***

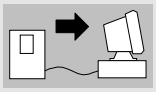
04	30	30	31	31	49	49	05

- ***As Characters - Establish Connection / Ask Question:***

(EOT)	(GID)	(GID)	(UID)	(UID)	(C1)	(C2)	(ENQ)

Note: The (GID)(UID) address is always entered twice.
Refer to "Instrument Address", page 18 for a more detailed explanation.

RESPONSE



- ***For software users:***

The Instrument Identity will be returned, in our case 5900 (representing a 590+ Drive)

- ***For programmers, in ASCII:***

(STX)	I	I	>	5	9	0	0	(ETX)	1

- ***For programmers, in HEX:***

02	49	49	3E	35	39	30	30	03	31

- ***As Characters - Valid Response:***

(STX)	(C1)	(C2)	(D1)	(D2)	(D3)	(D4)	(D5)	(ETX)	(BCC)

*Note: The BCC checksum (XOR) of the data after and excluding (STX) up to and including (ETX) is "**1**" and >**31**. Refer to "Block Check Character (BCC)", page 17 for a more detailed explanation.*

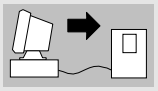
In Example 1, connection to a new device is being made, i.e. the "Establish Connection" information is transmitted. However, these examples can be transmitted without the "Establish Connection" information if connection to the correct device is already established. This is shown by Examples 3, 5 & 6.

Example 2: Tag Access (Single Parameter Poll)

Here we ask a question of a single parameter: *what is the value of SETPOINT 1?*. The example below is for a 590+ product.

(Tag 289, SETPOINT 1, ID 81, Type INT - see the Parameter Specification Table in the Product Manual for this information)

ENQUIRY



- ***For software users**:*

Enter the known address of the Drive (say 01), 81, and that it is an enquiry.

- ***For programmers, in ASCII:***

(EOT)	0	0	1	1	8	1	(ENQ)

- ***For programmers, in HEX:***

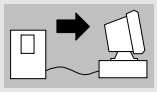
04	30	30	31	31	38	31	05

- ***As Characters - Establish Connection / Ask Question:***

(EOT	(GID)	GID)	(UID)	(UID)	(C1)	(C2)	(ENQ)

Note: The (GID)(UID) address is always entered twice.
Refer to "Instrument Address", page 18 for a more detailed explanation.

RESPONSE



- ***For software users**:*

The SETPOINT 1 value will be returned, say 30. (representing 30.00%)

- ***For programmers, in ASCII:***

(STX)	8	1	3	0	.	(ETX)	`

- ***For programmers, in HEX:***

02	38	31	33	30	2E	03	27

- ***As Characters - Valid Response:***

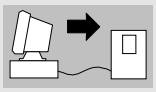
(STX)	(C1)	(C2)	(D1)	(D2)	(D3)	(ETX)	(BCC)

*Note: The BCC checksum (XOR) of the data after and excluding (STX) up to and including (ETX) is "`" and >**27**. Refer to "Block Check Character (BCC))", page 17 for a more detailed explanation.*

Example 3: Tag Access (Continuous Polling of a Parameter)

After receiving a valid response (from Example 2), you can cause the Drive to repeat that response without having to re-establish the connection. You can use this to continuously monitor a parameter.

ENQUIRY



- ***For software users***:
  Send (NAK).
- ***For programmers, in ASCII:***

  (NAK)

- ***For programmers, in HEX:***

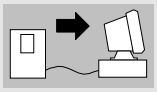
  15

- ***As Characters - Repeat Parameter:***

  (NAK)

RESPONSE

The response will be as for Example 2, however the returned data will be an updated value, i.e. SETPOINT 1 may now be 32. (representing 32.00%).

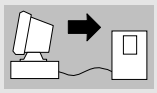


Example 4: Tag Access (Single Parameter Selection)

Here we are writing a value to a single parameter: *the value of TAKE UP 1 is 30.00%.* The example below is for a 590+ product.

SET PARAMETER

(Tag 253, TAKE UP 1, ID 71, Type INT - see the Parameter Specification Table for this information)



- ***For software users**:*
  Enter the known address of the Drive (say 01), (STX), 71, 30. and (ETX).
- ***For programmers, in ASCII:***

(EOT)	0	0	1	1	(STX)	7	1	3	0	.	(ETX)	(

- ***For programmers, in HEX:***

04	30	30	31	31	02	37	31	33	30	2E	03	28

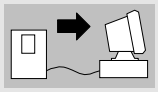
- ***As Characters - Establish Connection / Data Transfer:***

(EOT)	(GID)	(GID)	(UID)	(UID)	(STX)	(C1)	(C2)	(D1)	(D2)	(D3)	(ETX)	(BCC)

Note: The (GID)(UID) address is always entered twice.
Refer to "Instrument Address", page 18 for a more detailed explanation.

*The BCC checksum (XOR) of the data after and excluding (STX) up to and including (ETX) is "**(**" and >**28**. Refer to "Block Check Character (BCC)", page 17 for a more detailed explanation.*

RESPONSE



- ***For software users**:*
  The response will be either (ACK), (NAK) or no reply. If (ACK), the parameter value will be updated at the Drive.
- ***For programmers, in ASCII:***

either (ACK), (NAK) or no reply

- ***For programmers, in HEX:***

either 06, 15 or no reply

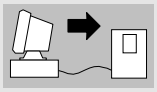
- ***As Characters:***

either (ACK), (NAK) or no reply

Example 5: Tag Access (Continuous Selection of a Parameter)

You can repeat a valid selection (from Example 4) without having to re-establish connection to the Drive. You can use this to continuously update a parameter. Lets say the new value is 35. (representing 35.00%).

SET PARAMETER



- ***For software users***:
  Send (STX), 71, 35. and (ETX).
- ***For programmers, in ASCII:***

(STX)	7	1	3	5	.	(ETX)	-

- ***For programmers, in HEX:***

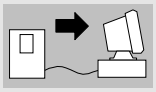
02	37	31	33	35	2E	03	2D

- ***As Characters - Data Transfer:***

(STX)	(C1)	(C2)	(D1)	(D2)	(D3)	(ETX)	(BCC)

Note: The BCC Checksum is the result of the new value you are sending to the Drive. Refer to "Block Check Character (BCC)", page 17 for a more detailed explanation.

RESPONSE



- ***For software users***:
  The response will be either (ACK), (NAK) or no reply. If (ACK), the parameter value will be updated at the Drive.
- ***For programmers, in ASCII:***

either (ACK), (NAK) or no reply

- ***For programmers, in HEX:***

either 06, 15 or no reply

- ***As Characters:***

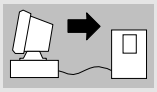
either (ACK), (NAK) or no reply

Example 6: Tag Access (Sequential Selection)

You can also repeat a valid selection (as Example 5) without having to re-establish the connection to the Drive to update any other specified parameter. Lets say the next parameter you want to update is I DMD. ISOLATE whose new value is to be ENABLED (1). The example below is for a 590+ product.

(Tag 119, I DMD. ISOLATE , ID 3b, Type BOOL - see the Parameter Specification Table for this information)

SET PARAMETER



- ***For software users:***
  Send (STX), 3b, 1 and (ETX).
- ***For programmers, in ASCII:***

(STX)	3	b	>	0	1	(ETX)	m

- ***For programmers, in HEX:***

02	33	62	3E	30	31	03	6D

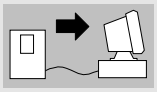
- ***As Characters - Data Transfer:***

(STX)	(C1)	(C2)	>	(D1)	(D2)	(ETX)	(BCC)

Note: The BCC Checksum is the result of the new information you are sending to the Drive.

RESPONSE

The response will be as for Example 5.

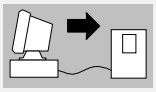


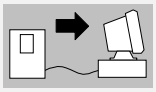
Character Definitions

Standard Character Definitions	
(GID)	The Group address Identifier (repeated for security)
(UID)	The Unit address identifier (repeated for security)
(C1) (C2)	The two characters of the parameter mnemonic (from the Tag number)
(D1)..(Dn)	The value of the requested parameter (string may be any length, determined by the data).
(BCC)	Block Check Character: a character generated by taking the exclusive OR (XOR) of the ASCII values of all the characters transmitted after and excluding (STX) up to and including (ETX)

Control Character Definitions

Standard Control Character Definitions	
(STX)	Start of text
(ETX)	End of text
(EOT)	End of Transmission: resets all instruments on the link and causes them to examine the next four transmitted characters to see if they correspond with their Group/Unit address identifiers Also sent to terminate communication with a particular device.

Control Character Definitions when Reading Information	
(ENQ)	Indicates the end of the message, and that it is an enquiry
(ACK)	Sequential Polling: when transmitted after a valid response, this fetches data from the next parameter in the parameter list
(NAK)	Continuous Polling: when transmitted after a valid response, this fetches data from the previously requested parameter
(EOT)	The information received contained an error

Control Character Definitions when Writing Information	
(ACK)	Positive Acknowledgement: the message was correctly received and the parameter updated
(NAK)	Negative Acknowledgement: the message received by the drive contained an error and the parameter was not updated

Last Error Code (EE)

The EI-BISYNCH Prime Set contains the EE mnemonic. This is also an output parameter in the TEC OPTION function block, where the parameter value can be read and reset. Refer to “Configuring the Drive”, page 12.

The following values are returned if an enquiry (reading information from the drive) is performed on this Read/Write parameter.

Writing any value to this parameter will set the value to >00C0. Clearing the last error value may be useful in seeing a repetitive error re-occurring.

Value	Description
>00C0	No error
>01C7	Invalid Mnemonic
>02C2	Checksum (BCC) error
>04C8	Attempt to read from a write-only parameter
>05C8	Attempt to write to a read-only parameter
>07C8	Invalid Data (Encoding error)
>08C8	Data out of range

Configuring the Drive

MMI Menu Map
Non-specific MMI view

1 SETUP PARAMETERS
2 FUNCTION BLOCKS
3 SERIAL LINKS
4 TEC OPTION
- TEC OPTION TYPE
- TEC OPTION IN 1
- TEC OPTION IN 2
- TEC OPTION IN 3
- TEC OPTION IN 4
- TEC OPTION IN 5
- TEC OPTION FAULT
- TEC OPTION VER
- TEC OPTION OUT 1
- TEC OPTION OUT 2

MMI Menu Map
EI BISYNCH Binary
MMI view

1 SERIAL LINKS
2 TEC OPTION
3 SERIAL LINKS
4 TEC OPTION
- TEC OPTION TYPE
- PROTOCOL
- BAUD RATE
- GROUP ID (GID)
- UNIT ID (UID)
- ERROR RESET
- FAULT
- VERSION
- ERROR CODE
- ERROR COUNT

SERIAL LINKS is at Menu Level 1 for the 590+ and 590+DRV.

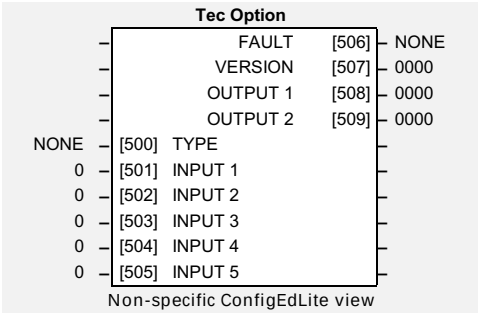
Using the Operator Station (MMI) or other suitable PC programming tool, the TEC OPTION function block requires configuring before the RS485 option can be used.

The parameter names/functions in the TEC OPTION function block are inter-dependant and will change with different parameter values and the various Technology Options that can be fitted.

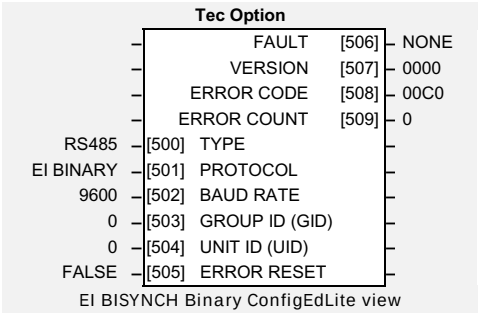
The top function block diagram shows the ConfigEd Lite parameter names, which are also displayed on the MMI if no Technology Option is fitted or an incorrect TYPE is selected for the fitted Technology Option.

ConfigEd Lite is Eurotherm Drives' Windows-based block programming software.

When the TYPE parameter is set to display RS485, the function block parameters take on new identities, as shown in the lower Function Block diagram.



Non-specific ConfigEdLite view



EI BISYNCH Binary ConfigEdLite view

Selecting EI Bisynch Binary

(Select Advanced view level on the Operator Station and view the TEC OPTION function block).

- Select RS485 in the TYPE parameter
- Select EI BINARY in the PROTOCOL parameter
- Select the Baud Rate
- Enter a GID address (if required)
- Enter a UID address (if required)
- Check the FAULT parameter for error messages and rectify if necessary

When setting values for parameters from ConfigEd Lite (or other suitable PC programming tool) you are able to select any value in the parameter's range, i.e. -32768 to 32767. If the value is incorrect, i.e. it doesn't correspond to a value that can be set using the MMI, then the FAULT output parameter will be set to PARAMETER.

MMI Parameter Descriptions for EI Bisynch Binary

TYPE *Range: Enumerated - see below*

Selects the type of Technology Option. The TYPE parameter is automatically set when defaults are loaded if a Technology Option is present.

Enumerated Value : Technology Option

0 : NONE
1 : RS485
2 : PROFIBUS DP
3 : LINK
4 : DEVICENET
5 : CANOPEN
6 : LONWORKS
7 : TYPE 7

PROTOCOL *Range: Enumerated - see below*

Selects the protocol to be used.

Enumerated Value : Protocol

0 : EI ASCII (default)
1 : EI BINARY
2 : MODBUS RTU

BAUD RATE *Range: Enumerated - see below*

Selects the Baud Rate.

Enumerated Value : Baud Rate

0 : 300
1 : 600
2 : 1200
3 : 2400
4 : 4800
5 : 9600 (default)
6 : 19200

GROUP ID (GID) *Range: 0 to 7*

The Eurotherm protocol group identity address.

UNIT ID (UID) *Range: 0 to 15*

The Eurotherm protocol unit identity address.

ERROR RESET *Range: FALSE/TRUE*

When TRUE, clears the ERROR CODE parameter (setting it to 00C0) and sets the ERROR COUNT parameter to zero.

FAULT *Range: Enumerated - see below*

The fault state of the Technology Option.

0 : NONE	no faults
1 : PARAMETER	parameter out-of-range
2 : TYPE MISMATCH	TYPE parameter not set to RS485
3 : SELF TEST	hardware fault - internal
4 : HARDWARE	hardware fault - external
5 : MISSING	no option fitted

Also refer to “Module LED”, page 10.

VERSION *Range: 0x0000 to 0xFFFF*

The version of the Technology Option card. If no option is fitted then the version is reset to zero.

ERROR CODE *Range: 0x0000 to 0xFFFF*

Displays the last error as a hexadecimal code. Refer to“Last Error Code (EE)”, page 29 for a list of codes.

ERROR COUNT *Range: 0 to 9999*

Increments each time an error is detected.

Note: will stop counting at 9999 (see ERROR RESET).

Configuring the PLC/SCADA Supervisor

By referring to the Parameter Specification Table in the main Product Manual, you can enter the parameter information you require.

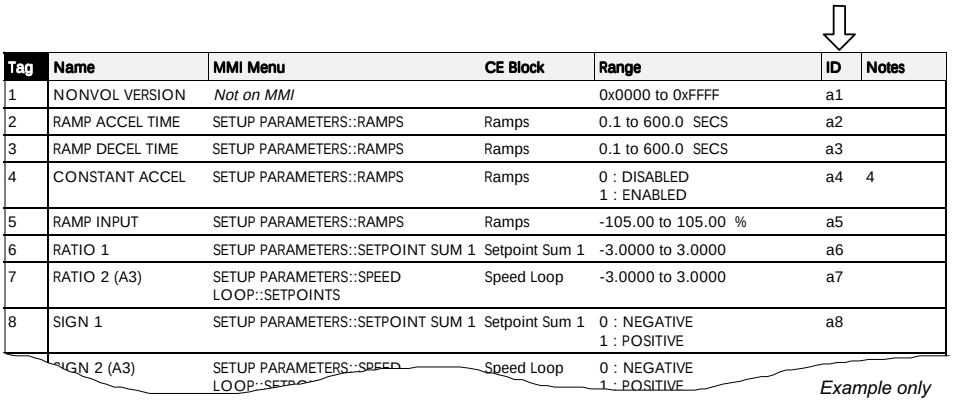
It provides the information in the following way:

Type

The first page of the Parameter Specification Table details parameter types.

ID/MN

The ID or MN column provides the parameter mnemonic (of the tag number).



Tag	Name	MMI Menu	CE Block	Range	ID	Notes
1	NONVOL VERSION	*Not on MMI*		0x0000 to 0xFFFF	a1	
2	RAMP ACCEL TIME	SETUP PARAMETERS::RAMPS	Ramps	0.1 to 600.0 SECS	a2	
3	RAMP DECEL TIME	SETUP PARAMETERS::RAMPS	Ramps	0.1 to 600.0 SECS	a3	
4	CONSTANT ACCEL	SETUP PARAMETERS::RAMPS	Ramps	0 : DISABLED 1 : ENABLED	a4	4
5	RAMP INPUT	SETUP PARAMETERS::RAMPS	Ramps	-105.00 to 105.00 %	a5	
6	RATIO 1	SETUP PARAMETERS::SETPOINT SUM 1	Setpoint Sum 1	-3.0000 to 3.0000	a6	
7	RATIO 2 (A3)	SETUP PARAMETERS::SPEED LOOP::SETPOINTS	Speed Loop	-3.0000 to 3.0000	a7	
8	SIGN 1	SETUP PARAMETERS::SETPOINT SUM 1	Setpoint Sum 1	0 : NEGATIVE 1 : POSITIVE	a8	
	SIGN 2 (A3)	SETUP PARAMETERS::SPEED LOOP::SETPO	Speed Loop	0 : NEGATIVE 1 : POSITIVE		

Example only

Binary Communications

This mode has many similarities with the ASCII mode, and so what follows is a summary of the differences to the ASCII mode.

Character Format
Each byte is transmitted as 11 bits rather than adapting the 10-bit format used by the ASCII mode. The format is represented by the following:-

1	Start bit (low)	
7	Data bits (LSB first)	
1	Control bit *	
1	Even parity bit	
1	Stop bit (high)	* 0 = Control character, 1 = Data character

How is the Information Transferred?

During serial communications, Drive acts as a slave and responds to messages sent from the Supervisor. Messages received from the Supervisor are categorised into Main Messages and Continuation Messages.

The Binary mode introduces several different Control and Data Characters. Refer to “EI Bisynch Binary Message Protocol”, page 34.

Response to a `Selection’ Message

The response is very similar to the ASCII mode but differs in that the ASCII (GID)/(UID) address is replaced by the Binary (INO), Instrument Number. Also, the ASCII parameter mnemonic (C1)(C2) is replaced by the Binary (PNO) character.

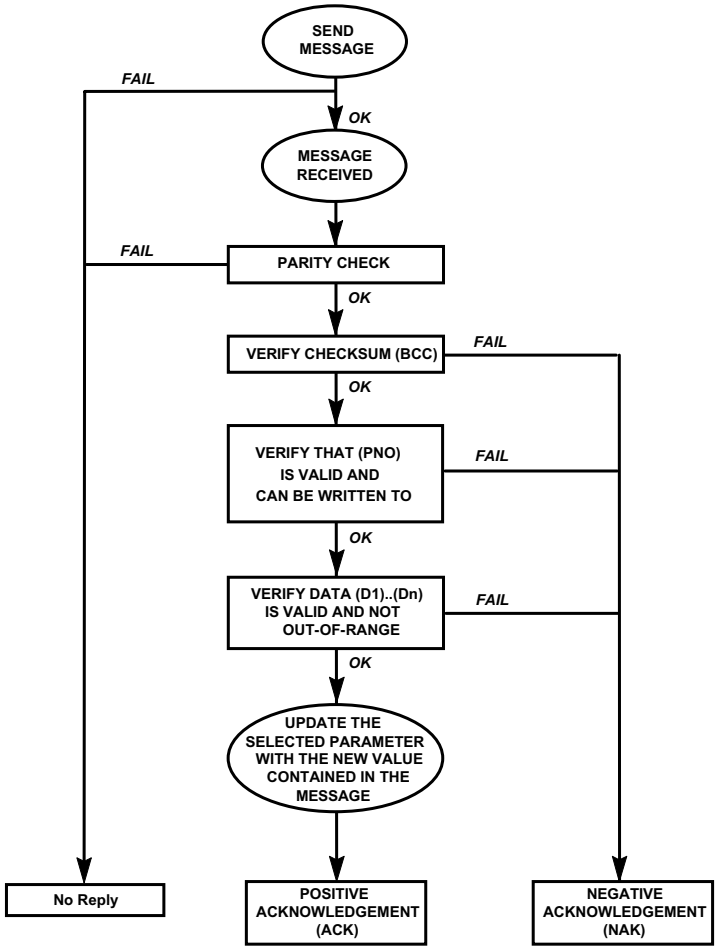


Figure 14 Converter Response Sequence to a Binary Selection Message

EI Bisynch Binary Message Protocol

Transmission Standard	:	RS485 (RS422 bi-directional)
Protocol	:	ANSI-X3.28-2.5-B1
Data Rates	:	300, 600, 1200, 2400, 4800, 9600 or 19200 Baud
Character Format	:	1 start + 8 bit ASCII data + 1 parity + 1 stop bit (11 bits)
Parity	:	Even

Transferring Data - Binary Example Messages

There are two message types:

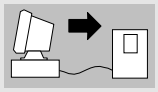
1. Main Messages
2. Continuation Messages

Main Messages

The main messages are in four types:

SELECTION

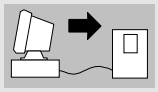
The Supervisor writes to one parameter. The (BCC) character contains the checksum of all characters following the (STX).



(EOT)	(INO)	(CCC)	(STX)	(PNO)	(D1)	(D2)	(D3)	(ETX)	(BCC)

POLLING

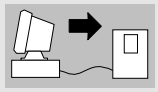
The Supervisor requests to read the value of one parameter.



(EOT)	(INO)	(PNO)	(CCC)	(ENQ)

ENQUIRY POLLING

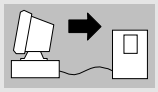
The Supervisor requests to read all parameters in block 1.



(EOT)	(INO)	(CCC)	(ENQ)

MULTI-PARAMETER POLLING

The Supervisor requests to read a given number of parameters. That number is referred to as the count number (CNO), it is included in the request message and the reply will be sent by the drive, in blocks of up to 8 parameters.



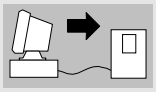
(EOT)	(INO)	(PNO)	(CNO)	(CCC)	(ENQ)

Note: The (CCC) is the checksum of the characters following an (EOT) and is therefore equal to (INO) in Selection and Enquiry Polling messages.

Continuation Messages

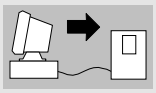
There are two types of continuation messages sent by the Supervisor:

NEXT (send next item from a list)
Only valid if sent following a multi-parameter poll.



(ACK)

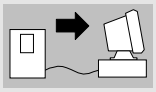
REPEAT (repeat last response)
Only valid if sent following any type of poll. It requests a repetition of the previous response.



(NAK)

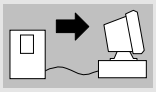
Serial Transmission Responses

SELECTION MESSAGE RESPONSE (one character)
Sent after the correct reception of a Selection message.



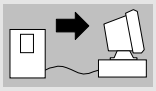
(ACK)

FAULT DETECTION RESPONSE (one character)
Sent in the case of detecting a fault.



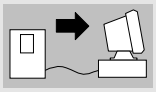
(NAK)	or	(EOT)

POLLING MESSAGE RESPONSE (more than one character)



(STX)	(PNO)	(D1)	(D2)	(D3)	(ETX)	(BCC)

MULTI-POLLING MESSAGE RESPONSE (more than one character)
The response can consist of a group of messages (blocks). The (ETX) character is only sent at the end of the last block (as for Polling Message Response above). For other blocks, the (ETX) is replaced by an (ETB) to indicate an end of a block rather than the end of the response.



Control Character Definitions

Standard Control Character Definitions	
(EOT)	End of Transmission (commands the slave to stop transmitting and wait to be addressed)
(STX)	Start of Text.
(ENQ)	Enquiry (sent by the master as the last character of any type of polling message)
(ETX)	End of Text (is followed by the checksum)
(ETB)	End of Block (sent instead of (ETX) when replying to a multi parameter enquiry). It indicates the end of a block, but not the end of a message.
(ACK)	Positive Acknowledgement
(NAK)	Negative Acknowledgement

Data Character Definitions

Standard Data Character Definitions	
(INO)	Instrument Number (contains the address of the slave drive and is equivalent to the combination of the GID, UID characters of the ASCII mode)
(PNO)	Parameter Number (equivalent to the combination of the (C1) and (C2) characters of the ASCII mode and is sent as a hexadecimal number rather than two ASCII characters)
(D1), (D2) and (D3)	These characters include the mode name and value read from, or to be written to, one of the parameters. A data character is represented by setting its MSB (bit 7). The contents of these characters are as follows: D1 : bits 2 [→] 6 mode number *Number format is:* 0 = XXXX 1 = XXX.X 2 = XX.XX 3 = X.XXX 4 = .XXXX bits 0 and 1 bits 14 and 15 of the value. D2 : bits 0 [→] 6 bits 7 to 13 of the value. D3 : bits 0 [→] 6 bits 0 to 6 of the value.
(CCC)	Connection Check Control (contains the checksum of all the characters following the (EOT) character in the message)
(BCC)	Block Check Charceter (checksum value generated by taking the exclusive OR (XOR) of the ASCII values of all characters transmitted after and excluding (STX) up to and including (ETX).

List of PNO Assignments

The serial link parameter numbers (PNO) include dedicated parameters, and also 16 configurable parameters. These vary with each Drive type.

590+ and 590+DRV

The 16 configurable parameters have PNO's 112 to 127. These can be made to point to any TAG number, either via the MMI (PNO CONFIG), or via the serial link.

PNO's 96 to 111 are pointers associated with PNO's 112 to 127.

For example:

If PNO 96 = 123, then PNO 112 will access TAG number 123.

If PNO 100 = 234, then PNO 116 will access TAG number 234

605A & B, 605C and 584SV

The PRESET 7 and PRESET 8 function blocks INPUT parameters are used to specify the tag. For example:

If PRESET 7::INPUT 0 = 1.23%, then PNO 112 will access tag number 123

If PRESET 7::INPUT 4 = 2.34%, then PNO 116 will access tag number 234

Enquiry Polling

In Enquiry Polling mode, block 1 is polled.

EI Bisynch Binary Parameter Specification Tables

Block 0 (590+)

PNO	ACCESS	DESCRIPTION
0	R/O	Instrument Identifier. Same as ASCII mnemonic II.
1	R/W	Error report. Same as ASCII mnemonic EE
2	R/O	Drive Software Version Number.

Block 1 (590+)

PNO (ID)	(MN)	TAG	DATA FORMAT			LIMITS	ACCESS	DESCRIPTION
BINARY	ASCII		BIT	ASCII	BINARY	MIN TO MAX		
8	08	063	-	21	xxx.xx		R/O	Speed Setpoint
9	09	089	-	21	xxx.xx		R/O	Speed Demand
10	0A	062	-	21	xxx.xx		R/O	Speed Feedback
11	0B	066	-	21	xxx.xx		R/O	Current Demand
12	0C	065	-	21	xxx.xx		R/O	Current Feedback
13	0D	183	-	21	xxx.xx		R/O	Field Demand
14	0E	181	-	21	xxx.xx		R/O	Field Feedback
15	0F	115	-	23	xxxxx		R/O	Health Word
			0			0/1		OVERSPEED
			1			0/1		MISSING PULSE
			2			0/1		FIELD OVER I
			3			0/1		Fin Over Temperature
			4			0/1		Motor Over Temperature
			5			0/1		OVER VOLTS (VA)
			6			0/1		Speed Feedback
			7			0/1		Encoder Failed
			8			0/1		Field Failed
			9			0/1		Three Phase Failed
			10			0/1		Phase Lock Loop
			11			0/1		5703 Receive Error
			12			0/1		Stall Trip
			13			0/1		Over Current Trip
			14			0/1		Cal. Card
			15			0/1		ACCTS Failed

Block 2 (590+)								
PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
16	10	050	-	21	xxx.xx		R/O	Anin 1 (A2)
17	11	051	-	21	xxx.xx		R/O	Anin 2 (A3)
18	12	052	-	21	xxx.xx		R/O	Anin 3 (A4)
19	13	053	-	21	xxx.xx		R/O	Anin 4 (A5)
20	14	054	-	21	xxx.xx		R/O	Anin 5 (A6)
21	15	067	-	21	xxx.xx		R/O	Actual Pos I Lim
22	16	061	-	21	xxx.xx		R/O	Actual Neg I Lim
23	17	040	-	23	xxxxx		R/O	
-		068	0			0/1		Start Input
-		069	1			0/1		Jog Input
-		070	2			0/1		Enable Input
-		071	3			0/1		Digital Input 1
-		072	4			0/1		Digital Input 2
-		073	5			0/1		Digital Input 3
-		-	6			0/1		Program Stop Input
-		-	7			0/1		Coast Stop Input
-		074	8			0/1		Digital Output 1
-		075	9			0/1		Digital Output 2
-		076	10			0/1		Digital Output 3
-		-	11-15			0/1		Reserved

Block 3 (590+)								
PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
24	18	030		21	xxx.xx	-200.00/200.00	R/W	Additional Current Demand
25	19	015		21	xxx.xx	0/200.00	R/W	Main Current Limit
26	1A	087		21	xxx.xx	0/200.00	R/O	+ve Current Clamp
27	1B	088		21	xxx.xx	0/200.00	R/O	-ve Current Clamp
28	1C	016		21	xxx.xx	0/200.00	R/W	Current Loop P Gain
29	1D	017		21	xxx.xx	0/200.00	R/W	Current Loop I Gain
30	1E	171		21	xxx.xx	0/100.00	R/W	Field Current Setpoint
31	1F	116		23	xxxxx		R/O	Health Store
			0			0/1		Over Speed
			1			0/1		Missing Pulse
			2			0/1		Field Over Current
			3			0/1		Fin Over Temperature
			4			0/1		Motor Over Temperature
			5			0/1		Field Over Volts
			6			0/1		Speed Feedback
			7			0/1		Encoder Fail
			8			0/1		Field Fail
			9			0/1		Three Phase

Block 3 (590+)								
PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
			10			0/1		Phase Lock Loop
			11			0/1		5703 Receive Error
			12			0/1		Stall Trip
			13			0/1		Over Current Trip
			14			0/1		Cal. Card
			15			0/1		ACCTS Failed.

Block 4 (590+)								
PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
32	20	060		21	xxx.xx		R/O	Back EMF
33	21	058		21	xxx.xx		R/O	Analogue Tach
34	22	059		21	xxxxx		R/O	Encoder
35	23	064		21	xxx.xx		R/O	Speed Error
36	24	132		21	x.xxxx	-3.0000/3.0000	R/W	P3 Setpoint Ratio
37	25	014		21	xxx.xx	0/200.00	R/W	Speed Loop P Gain
38	26	013		21	xx.xxx	0.001/ 30.000	R/W	Speed Loop Time Constant (SEC)
39 *	27			23	xxxxx			
		161	0			0/1	R/W	Aux. Start
		168	1			0/1	R/W	Aux. Enable
			2.7				-	Reserved
		288	8			0/1	R/W	External Ramp Reset
		287	9			0/1	R/W	Auto Reset
		113	10				R/O	Ramping
		303	11			0/1	R/W	Reset Ramp to Speed Feedback

Block 5 (590+)								
PNO (ID)	(MN)	TAG	DATA FORMAT			LIMITS	ACCESS	DESCRIPTION
BINARY	ASCII		BIT	ASCII	BINARY	MIN TO MAX		
40	28	006		21	x.xxxx	-3.0000/3.0000	R/W	Ratio 1
41	29	007		21	x.xxxx	-3.0000/3.0000	R/W	Ratio 2
42	2A	086		21	xxx.xx		R/O	Set Point Sum Output
43	2B	002		21	xxx.x	0.1/600.0	R/W	Ramp Accel. Time
44	2C	003		21	xxx.x	0.1/600.0	R/W	Ramp Decel. Time
45	2D	085		21	xxx.xx	-	R/O	Ramp Output
46	2E	041		21	xxx.xx	-100.00/100.00	R/W	Speed Setpoint 4
47	2F			23	xxxxx			
		082	0				R/O	Drive Start
		084	1				R/O	Drive Enable
		122	2				R/O	Health Flag
		125	3				R/O	Ready
			4 - 7					Reserved
		079	8				R/O	At Standstill
		112	9				R/O	Stall Trip Warning
			10 - 15					Reserved

Block 6 (590+)								
PNO (ID)	(MN)	TAG	DATA FORMAT			LIMITS	ACCESS	DESCRIPTION
BINARY	ASCII		BIT	ASCII	BINARY	MIN TO MAX		
48	30	027		21	xxx.x	0.1/600.0	R/W	Stop time
49	31	026		21	xxx.x	0.1/600.0	R/W	P-Stop time
50	32	091		21	xxx.xx	0/200.00	R/W	P-Stop Current Limit
51	33	029		21	xxx.xx	0/100.00	R/W	Stop Zero Speed Threshold
52	34	005		21	xxx.xx	-100.00/100.00	R/W	Ramp Input
53	35	100		21	xxx.xx	-200.00/200.00	R/O	Setpoint Sum Input 1
54	36	309		21	xxx.xx	-200.00/200.00	R/W	Setpoint Sum Input 0
55	37			23	xxxxx			
		94	0			0/1	R/W	Aux. Digital Output 1
		95	1			0/1	R/W	Aux. Digital Output 2
		96	2			0/1	R/W	Aux. Digital Output 3
		-	3 - 7					Reserved
		292	8			0/1	R/W	Sign 0
		8	9			0/1	R/W	Sign 1
		9	10			0/1	R/W	Sign 2
			11 - 15					Reserved

Block 7 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
56	38	055		21	xxx.xx		R/O	Analogue Output 1
57	39	056		21	xxx.xx		R/O	Analogue Output 2
58	3A	128		21	xxx.xx	-100.00/100.00	R/W	Aux. Analogue Output 1
59	3B	129		21	xxx.xx	-100.00/100.00	R/W	Aux. Analogue Output 2
60	3C	266		21	xxx.xx	0/100.00	R/W	% S-Ramp
61	3D	264		21	xxx.xx		R/O	Raise / Lower Output
62	3E	255		21	xxx.xx	-300.00/300.00	R/W	Raise / Lower Reset Value
63	3F	-		23	xxxxx			
-		261	0			0 1	R/W	Raise / Lower Raise Input
-		262	1			0 1	R/W	Raise/Lower Lower Input
-		307	2			0 1	R/W	Raise / Lower Reset

Block 8 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
64	40	218		21	xxx.xx	-100.00/100.00	R/W	Jog Speed 1
65	41	219		21	xxx.xx	-100.00/100.00	R/W	Jog Speed 2
66	42	253		21	xxx.xx	-100.00/100.00	R/W	Take Up 1
67	43	254		21	xxx.xx	-100.00/100.00	R/W	Take Up 2
68	44	225		21	xxx.xx	-100.00/100.00	R/W	Crawl Speed
71	47	-		23	xxxxx			
-		228	0			0 1	R/W	Jog Mode
-		227	1			0 1	R/W	Auxiliary Jog

Block 9 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
72	48	208		21	x.xxxx	-3.0000/+3.0000	R/W	Ratio 0
73	49	309		21	xxx.xx	-100.00/+100.00	R/W	Input 0
74	4A	48		21	xxx.xx	-100.00/+100.00	R/W	Pre-set -ve Current Limit
75	4B	301		21	xxx.xx	-100.00/+100.00	R/W	Pre-set +ve Current Limit

Block 10 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
80	50	103		21	xxx.xx	-300.00 /+300.00	R/W	Value for TRUE Digital Input 1
81	51	104		21	xxx.xx	-300.00 /+300.00	R/W	Value for FALSE Digital Input 1
82	52	106		21	xxx.xx	-300.00 /+300.00	R/W	Value for TRUE Digital Input 2
83	53	107		21	xxx.xx	-300.00 /+300.00	R/W	Value for FALSE Digital Input 2
84	54	109		21	xxx.xx	-300.00 /+300.00	R/W	Value for TRUE Digital Input 3
85	55	110		21	xxx.xx	-300.00/+300.00	R/W	Value for FALSE Digital Input 3

Block 11 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
88	58	339		21	xxx.xx	-300.00/+300.00	R/W	Value 1
89	59	340		21	xxx.xx	-300.00/+300.00	R/W	Value 2
90	5A	341		21	xxx.xx	-300.00/+300.00	R/W	Value 3
91	5B	342		21	xxx.xx	-300.00/+300.00	R/W	Value 4
92	5C	343		21	xxx.xx	-300.00/+300.00	R/W	Value 5
93	5D	344		21	xxx.xx	-300.00/+300.00	R/W	Value 6
94	5E	345		21	xxx.xx	-300.00/+300.00	R/W	Value 7
95	5F	-		23	xxxxx			
-		346	0			0 1	R/W	Logic 1
-		347	1			0 1	R/W	Logic 2
-		348	2			0 1	R/W	Logic 3
-		349	3			0 1	R/W	Logic 4
-		350	4			0 1	R/W	Logic 5
-		351	5			0 1	R/W	Logic 6
-		352	6			0 1	R/W	Logic 7
-		353	7			0 1	R/W	Logic 8

Block 12 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
96	60	312		21	xxxxx		R/W	Pointer for PNO 112
97	61	313		21	xxxxx		R/W	Pointer for PNO 113
98	62	314		21	xxxxx		R/W	Pointer for PNO 114
99	63	315		21	xxxxx		R/W	Pointer for PNO 115
100	64	316		21	xxxxx		R/W	Pointer for PNO 116
101	65	317		21	xxxxx		R/W	Pointer for PNO 117
102	66	318		21	xxxxx		R/W	Pointer for PNO 118
103	67	319		21	xxxxx		R/W	Pointer for PNO 119

Block 13 (590+)

PNO (ID) BINARY	(MN) ASCII	TAG	DATA FORMAT BIT	ASCII	BINARY	LIMITS MIN TO MAX	ACCESS	DESCRIPTION
104	68	320		21	xxxxx		R/W	Pointer for PNO 120
105	69	321		21	xxxxx		R/W	Pointer for PNO 121
106	6A	322		21	xxxxx		R/W	Pointer for PNO 122
107	6B	323		21	xxxxx		R/W	Pointer for PNO 123
108	6C	324		21	xxxxx		R/W	Pointer for PNO 124
109	6D	325		21	xxxxx		R/W	Pointer for PNO 125
110	6E	326		21	xxxxx		R/W	Pointer for PNO 126
111	6F	327		21	xxxxx		R/W	Pointer for PNO 127

Block 14 (590+)								
PNO (ID)	(MN)	TAG	DATA FORMAT			LIMITS	ACCESS	DESCRIPTION
BINARY	ASCII		BIT	ASCII	BINARY	MIN TO MAX		
112	70	PNO 96		*	*	*	*	Configurable PNO 0
113	71	PNO 97		*	*	*	*	Configurable PNO 1
114	72	PNO 98		*	*	*	*	Configurable PNO 2
115	73	PNO 99		*	*	*	*	Configurable PNO 3
116	74	PNO 100		*	*	*	*	Configurable PNO 4
117	75	PNO 101		*	*	*	*	Configurable PNO 5
118	76	PNO 102		*	*	*	*	Configurable PNO 6
119	77	PNO 103		*	*	*	*	Configurable PNO 7

Block 15 (590+)								
PNO (ID)	(MN)	TAG	DATA FORMAT			LIMITS	ACCESS	DESCRIPTION
BINARY	ASCII		BIT	ASCII	BINARY	MIN TO MAX		
120	78	PNO 104		*	*	*	*	Configurable PNO 8
121	79	PNO 105		*	*	*	*	Configurable PNO 9
122	7A	PNO 106		*	*	*	*	Configurable PNO 10
123	7B	PNO 107		*	*	*	*	Configurable PNO 11
124	7C	PNO 108		*	*	*	*	Configurable PNO 12
125	7D	PNO 109		*	*	*	*	Configurable PNO 13
126	7E	PNO 110		*	*	*	*	Configurable PNO 14
127	7F	PNO 111		*	*	*	*	Configurable PNO 15

** = These fields depend upon the destination TAG number*

Block 0 (605)			
PNO	**TAG**	**DESCRIPTION**	**EQUIVALENT ASCII COMMAND**
0		instument identity	(II)
1		error	(EE)
2		main version	(V0)
3		comms version	(V3)
4		system command	(!1)
5		system state	(!2)
6		save command	(!3)
7		save state	(!4)

Block 1 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
8	066	motor current	(xxx.xx%)
9	072	load	(xxx.xx%)
10	073	field	(xxx.xx%)
11	370	current limiting	(bool)
12	255	speed demand	(xxx.xx%)
13	591	drive frequency	(xxx.xHz)
14	006	first trip	(enum)
15	272	comms status	(word)

Block 2 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
16	568	encoder speed Hz	(xxx.xHz)
17	569	encoder speed RPM	(xxxxxn/min)
18	749	encoder speed %	(xxx.xx%)
19	748	encoder position	(xxxxx)
20	360	at zero speed	(bool)
21	004	active trips	(word)
22	005	trips warning	(word)
23	598	multiplexer output	(word)

Block 3 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
24	365	motor limit	(xxx.xx%)
25	623	regen limit	(xxx.xx%)
26	258	ramp up time	(xxx.xs)
27	259	ramp down time	(xxx.xs)
28	057	max speed	(xxx.xHz)
29	337	min speed	(xxx.xx%)
30	104	V/F shape	(enum)
31	106	base frequency	(xxx.xHz)

Block 4 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
32	107	fixed boost	(xx.xx%)
33	108	auto boost	(xx.xx%)
34	064	full load calib	(xxxx.xA)
35	065	no load calib	(xxxx.xA)
36	242	power factor	(x.xx)
37	237	I*t threshold	(xxx.xx%)
38	239	I*t upper limit	(xxx.xx%)
39	238	I*t time	(xxs)

Block 5 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
40	347	preset 1 input 0	(xxx.xx%)
41	348	preset 1 input 1	(xxx.xx%)
42	349	preset 1 input 2	(xxx.xx%)
43	350	preset 1 input 3	(xxx.xx%)
44	351	preset 1 input 4	(xxx.xx%)
45	352	preset 1 input 5	(xxx.xx%)
46	353	preset 1 input 6	(xxx.xx%)
47	354	preset 1 input 7	(xxx.xx%)

Block 6 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
48	380	preset 2 input 0	(xxx.xx%)
49	381	preset 2 input 1	(xxx.xx%)
50	382	preset 2 input 2	(xxx.xx%)
51	383	preset 2 input 3	(xxx.xx%)
52	384	preset 2 input 4	(xxx.xx%)
53	385	preset 2 input 5	(xxx.xx%)
54	386	preset 2 input 6	(xxx.xx%)
55	387	preset 2 input 7	(xxx.xx%)

Block 7 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
56	390	preset 3 input 0	(xxx.xx%)
57	391	preset 3 input 1	(xxx.xx%)
58	392	preset 3 input 2	(xxx.xx%)
59	393	preset 3 input 3	(xxx.xx%)
60	394	preset 3 input 4	(xxx.xx%)
61	395	preset 3 input 5	(xxx.xx%)
62	396	preset 3 input 6	(xxx.xx%)
63	397	preset 3 input 7	(xxx.xx%)

Block 8 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
64	342	skip frequency 1	(xxx.xHz)
65	343	skip frequency 2	(xxx.xHx)
66	344	skip frequency 3	(xxx.xHz)
67	345	skip frequency 4	(xxx.xHz)
68	341	skip band 1	(xxx.xHz)
69	680	skip band 2	(xxx.xHx)
70	681	skip band 3	(xxx.xHz)
71	682	skip band 4	(xxx.xHz)

Block 9 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
72	271	comms command	(word)
73	269	comms setpoint	(xxx.xx%)
74	355	preset 1 select	(enum)
75	388	preset 2 select	(enum)
76	398	preset 3 select	(enum)
77	279	run stopping mode	(enum)
78	304	fast stopping mode	(enum)
79	599	demultiplexer input	(word)

Block 10 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
80	311	pid enable	(bool)
81	313	pid p gain	(xxxx.x)
82	314	pid integral tc	(xx.xxs)
83	315	pid derivative tc	(xxxx.xs)
84	316	pid filter tc	(xxxx.xs)
85	317	pid pos out limit	(xxx.xx)
86	318	pid neg out limit	(xxx.xx)
87	319	pid out scaling	(x.xxxx%)

Block 11 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
88	130	value func 1 input a	(xxx.xx%)
99	131	value func 1 input b	(xxx.xx%)
90	132	value func 1 input c	(xxx.xx%)
91	133	value func 1 output	(xxx.xx%)
92	135	value func 2 input a	(xxx.xx%)
93	136	value func 2 input b	(xxx.xx%)
94	137	value func 2 input c	(xxx.xx%)
95	138	value func 2 output	(xxx.xx%)

Block 12 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
96	543	preset 7 input 0	(pointer for PNO 112)
97	544	preset 7 input 1	(pointer for PNO 113)
98	545	preset 7 input 2	(pointer for PNO 114)
99	546	preset 7 input 3	(pointer for PNO 115)
100	547	preset 7 input 4	(pointer for PNO 116)
101	548	preset 7 input 5	(pointer for PNO 117)
102	549	preset 7 input 6	(pointer for PNO 118)
103	550	preset 7 input 7	(pointer for PNO 119)

Block 13 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
104	554	preset 8 input 0	(pointer for PNO 120)
105	555	preset 8 input 1	(pointer for PNO 121)
106	556	preset 8 input 2	(pointer for PNO 122)
107	557	preset 8 input 3	(pointer for PNO 123)
108	558	preset 8 input 4	(pointer for PNO 124)
109	559	preset 8 input 5	(pointer for PNO 125)
110	560	preset 8 input 6	(pointer for PNO 126)
111	561	preset 8 input 7	(pointer for PNO 127)

Block 14 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
112		indirect access parameter 1	
113		indirect access parameter 2	
114		indirect access parameter 3	
115		indirect access parameter 4	
116		indirect access parameter 5	
117		indirect access parameter 6	
118		indirect access parameter 7	
119		indirect access parameter 8	

Block 15 (605)			
PNO	**TAG**	**DESCRIPTION**	**DATA FORMAT**
120		indirect access parameter 9	
121		indirect access parameter 10	
122		indirect access parameter 11	
123		indirect access parameter 12	
124		indirect access parameter 13	
125		indirect access parameter 14	
126		indirect access parameter 15	
127		indirect access parameter 16	

Configuring the Drive

MMI Menu Map
Non-specific MMI view

- 1 SETUP PARAMETERS
- 2 FUNCTION BLOCKS
- 3 SERIAL LINKS
- 4 TEC OPTION
  - TEC OPTION TYPE
  - TEC OPTION IN 1
  - TEC OPTION IN 2
  - TEC OPTION IN 3
  - TEC OPTION IN 4
  - TEC OPTION IN 5
  - TEC OPTION FAULT
  - TEC OPTION VER
  - TEC OPTION OUT 1
  - TEC OPTION OUT 2

MMI Menu Map
MODBUS RTU
MMI view

- 1 SETUP PARAMETERS
- 2 FUNCTION BLOCKS
- 3 SERIAL LINKS
- 4 TEC OPTION
  - TEC OPTION TYPE
  - PROTOCOL
  - BAUD RATE
  - PARITY
  - ADDRESS
  - ERROR RESET
  - FAULT
  - VERSION
  - ERROR CODE
  - ERROR COUNT

SERIAL LINKS is at Menu Level 1 for the 590+ and 590+DRV.

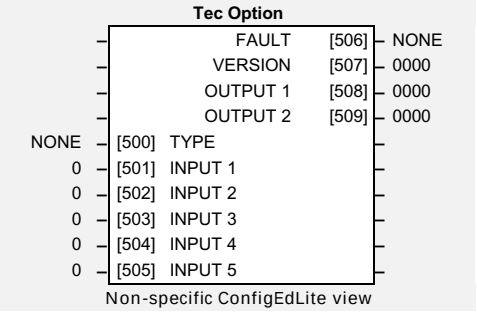
Using the Operator Station (MMI) or other suitable PC programming tool, the TEC OPTION function block requires configuring before the RS485 option can be used.

The parameter names/functions in the TEC OPTION function block are inter-dependant and will change with different parameter values and the various Technology Options that can be fitted.

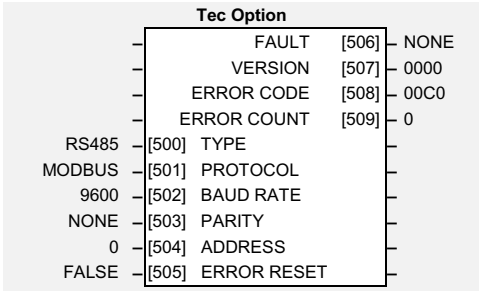
The top function block diagram shows the ConfigEd Lite parameter names, which are also displayed on the MMI if no Technology Option is fitted or an incorrect TYPE is selected for the fitted Technology Option.

ConfigEd Lite is Eurotherm Drives' Windows-based block programming software.

When the TYPE parameter is set to display RS485, the function block parameters take on new identities, as shown in the lower Function Block diagram.



Non-specific ConfigEdLite view



MODBUS RTU ConfigEdLite view

Selecting MODBUS RTU

(Select Advanced view level on the Operator Station and view the TEC OPTION function block).

- Select RS485 in the TYPE parameter
- Select MODBUS RTU in the PROTOCOL parameter
- Select the Baud Rate
- Select the Parity
- Enter the MODBUS slave address
- Check the FAULT parameter for error messages, rectify if necessary

When setting values for parameters from ConfigEd Lite (or other suitable PC programming tool) you are able to select any value in the parameter's range, i.e. -32768 to 32767. If the value is incorrect, i.e. it doesn't correspond to a value that can be set using the MMI, then the FAULT output parameter will be set to PARAMETER.

MMI Parameter Descriptions for MODBUS RTU

TYPE *Range: Enumerated - see below*

Selects the type of Technology Option. The TYPE parameter is automatically set when defaults are loaded if a Technology Option is present.

Enumerated Value : Technology Option

0 : NONE
1 : RS485
2 : PROFIBUS DP
3 : LINK
4 : DEVICENET
5 : CANOPEN
6 : LONWORKS
7 : TYPE 7

PROTOCOL *Range: Enumerated - see below*

Selects the protocol to be used.

Enumerated Value : Protocol

0 : EI ASCII (default)
1 : EI BINARY
2 : MODBUS RTU

BAUD RATE *Range: Enumerated - see below*

Selects the Baud Rate.

Enumerated Value : Baud Rate

0 : 300
1 : 600
2 : 1200
3 : 2400
4 : 4800
5 : 9600 (default)
6 : 19200

PARITY *Range: 0 to 7*

Selects the Parity

Enumerated Value : Parity

0 : NONE
1 : ODD
2 : EVEN

ADDRESS *Range: 0 to 247*

The MODBUS protocol unit identity address.
Note: if set to 0, it will only respond to broadcast messages.

ERROR RESET *Range: FALSE/TRUE*

When TRUE, clears the ERROR CODE and sets the ERROR COUNT parameter to zero.

FAULT *Range: Enumerated - see below*

The fault state of the Technology Option.

0 : NONE	no faults
1 : PARAMETER	parameter out-of-range
2 : TYPE MISMATCH	TYPE parameter not set to RS485
3 : SELF TEST	hardware fault - internal
4 : HARDWARE	hardware fault - external
5 : MISSING	no option fitted

Also refer to “Module LED”, page 10.

VERSION *Range: 0x0000 to 0xFFFF*

The version of the Technology Option card. If no option is fitted then the version is reset to zero.

ERROR CODE *Range: 0x0000 to 0xFFFF*

Displays the last error as a code. Refer to “Error Response”, page 62.

ERROR COUNT *Range: 0 to 9999*

Increments each time an error is detected.

Note: will stop counting at 9999 (see ERROR RESET).

Configuring the PLC/SCADA Supervisor

By referring to the Parameter Specification Table in the main Product Manual, you can enter the parameter information you require.

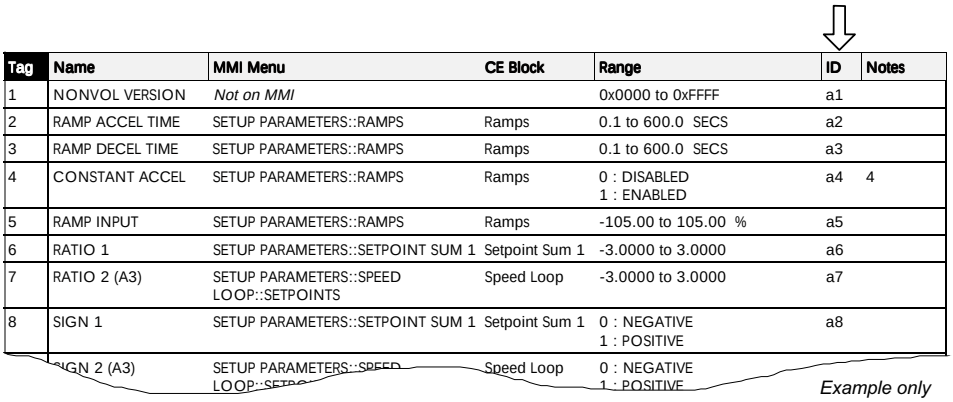
It provides the information in the following way:

Type

The first page of the Parameter Specification Table chapter details parameter types.

ID/MN

The ID or MN column provides the parameter mnemonic (of the tag number).



Tag	Name	MMI Menu	CE Block	Range	ID	Notes
1	NONVOL VERSION	*Not on MMI*		0x0000 to 0xFFFF	a1	
2	RAMP ACCEL TIME	SETUP PARAMETERS::RAMPS	Ramps	0.1 to 600.0 SECS	a2	
3	RAMP DECEL TIME	SETUP PARAMETERS::RAMPS	Ramps	0.1 to 600.0 SECS	a3	
4	CONSTANT ACCEL	SETUP PARAMETERS::RAMPS	Ramps	0 : DISABLED 1 : ENABLED	a4	4
5	RAMP INPUT	SETUP PARAMETERS::RAMPS	Ramps	-105.00 to 105.00 %	a5	
6	RATIO 1	SETUP PARAMETERS::SETPOINT SUM 1	Setpoint Sum 1	-3.0000 to 3.0000	a6	
7	RATIO 2 (A3)	SETUP PARAMETERS::SPEED LOOP::SETPOINTS	Speed Loop	-3.0000 to 3.0000	a7	
8	SIGN 1	SETUP PARAMETERS::SETPOINT SUM 1	Setpoint Sum 1	0 : NEGATIVE 1 : POSITIVE	a8	
	SIGN 2 (A3)	SETUP PARAMETERS::SPEED LOOP::SETPO	Speed Loop	0 : NEGATIVE 1 : POSITIVE		

Example only

MODBUS RTU Communications

A MODBUS RTU communication network can have only one Master, and one or more Slave devices.

- Each Slave has a unique “device address”
- The device address “0” is a special case and is used for messages that are broadcast to all Slaves. This is restricted to parameter write operations.
- The unit supports a subset of MODBUS RTU function codes.
- The data includes parameters referenced by a “parameter address”.
- Sending a communication with a unique device address causes only the device with that address to respond. That device will check for errors, perform the requested task and then reply with its own address, data and check sum.
- Sending a communication with the device address “0” is a broadcast communication that sends information to all devices on the network. Each device performs the required action but does not transmit a reply.

How is the Information Transferred?

A typical transaction consists of a request sent from the Master followed by a response from the Slave.

A message consists of a sequence of characters which we identify as:

- Device Address
- Function Code
- Data
- Error Check Data
- End of Transmission

Device Address

Each Slave has a unique 8-bit device address. The Gould MODBUS Protocol defines the address range limits as 1 to 247 (device address 0 is the broadcast message to all slaves simultaneously).

Parameter Address

Data bits or data words exchange information between Master and Slave devices. This data consists of parameters. All parameters communicated between Master and Slaves have a 16-bit parameter address.

The MODBUS parameter address range is 0001 to FFFF.

RTU Mode of Transmission

The MODBUS RTU definition of the mode of transmission for a single character is:

A start bit, eight data bits, a parity bit, one or two stop bits

All Eurotherm Drives' units use one stop bit.

Parity may be configured to be NONE, ODD or EVEN (if NONE, no parity bit is transmitted)

The RTU mode of transmission for a single character is represented as follows:

Start	d7	d6	d5	d4	d3	d2	d1	d0	Parity	Stop

Message Frame Format

A message frame format consists of a number of correctly sequenced characters, as shown below.

Frame Start	Device Address	Function Code	Data	CRC	EOT
3 bytes	1 byte	1 byte	n bytes	2 bytes	3 bytes

Frame Start

The frame start is a period of inactivity at least 3.5 times the single character transmission time. For example, at 9600 baud a character with a 1 start, 1 stop and 8 data bits will require 3.5ms frame start. This period is the implied EOT of a previous transmission.

Device Address

The device address is a single byte (8-bits), unique to each device on the network.

Function Code

Function codes are a single byte instruction to the Slave describing the action to perform.

Data

The Data segment of a message will depend on the function code and the number of bytes will vary accordingly. Typically, the data segment will contain a parameter address and the number of parameters to read or write.

CRC

The CRC (Cyclic Redundancy Check) is an error code and is 2 bytes (16-bits) long.

EOT

The EOT (End Of Transmission) segment is a period of inactivity 3.5 times the single character transmission time. The EOT segment at the end of a message indicates to the listening device that the next transmission will be a new message and therefore a device address character.

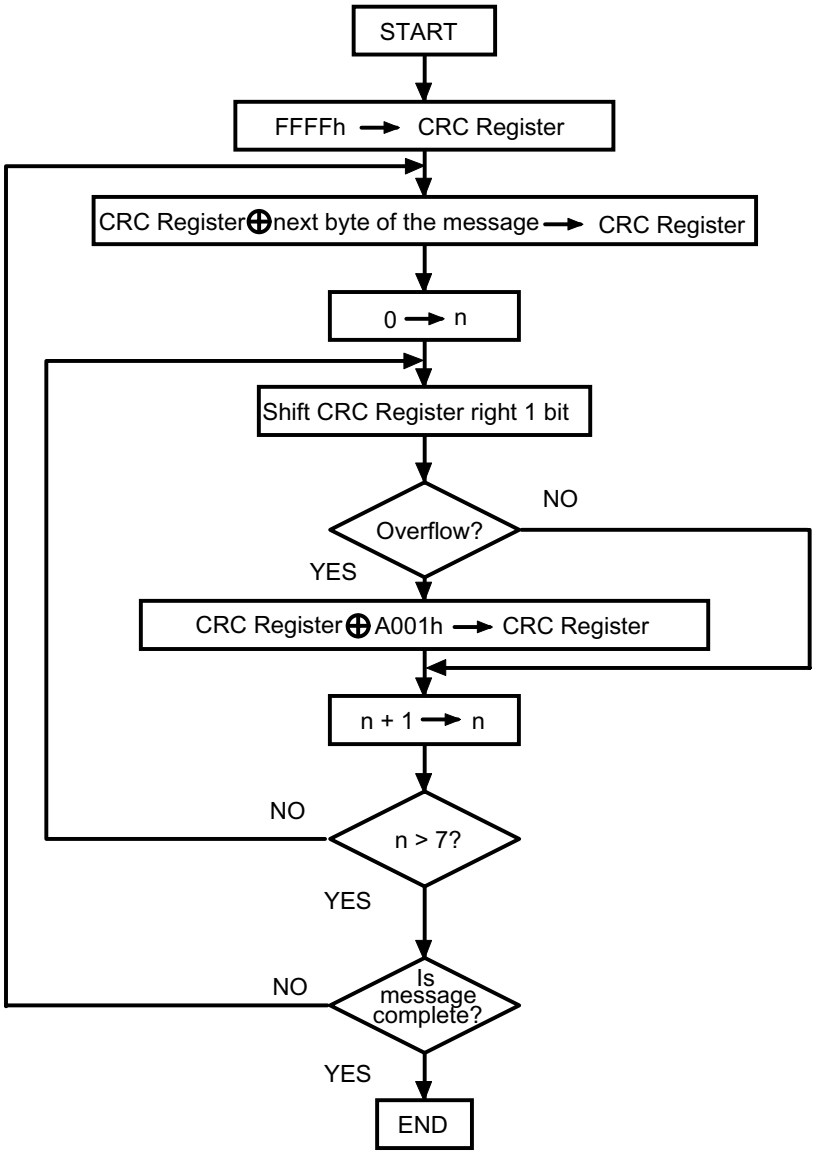
Cyclic Redundancy Check

This is an error check code and is 2 bytes (16-bits) long. After constructing a message (data only - no start, stop or parity bits), the transmitting device calculates a CRC code and appends this to the end of the message. The receiving device also calculates a CRC code from the received message. If this CRC code is not the same as the transmitted CRC there has been a communication error. Units do not reply if they detect a CRC error in messages sent to them.

The CRC code is formed by the following steps:

1. Load a 16-bit CRC register with FFFFh.
2. Exclusive OR (⊕) the first 8-bit byte of the message with the high order byte of the CRC register. Return the result to the CRC register.
3. Shift the CRC register one bit to the right.
4. If the overflow bit (or flag) is 1, exclusive OR the CRC register with A001 hex and return the result to the CRC register.
5. Repeat steps 3 & 4 seven times (8 in total).
6. Exclusive OR the next 8-bit byte of the message with the high order byte of the CRC register.
7. Repeat step 3 through 6 until all bytes of the message have been exclusive OR'd with the CRC register and shifted 8 times.
8. The contents of the CRC register are the 2 byte CRC error code and are added to the message with the most significant bits first.

The flow chart below illustrates this CRC error check algorithm.



Example of a CRC Calculation

This example is a request to read from the Slave unit at address 02, the fast read of the status (07).

Function	16 Bit Register				Carry
	LSB			MSB	Flag
Load register with FFFF hex	1111	1111	1111	1111	0
First byte of the message (02)			0000	0010	
Exclusive OR	1111	1111	1111	1101	
1st shift right	0111	1111	1111	1110	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1101	1111	1111	1111	
2nd shift right	0110	1111	1111	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1100	1111	1111	1110	
3rd shift right	0110	0111	1111	1111	0
4th shift right (carry = 0)	0011	0011	1111	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1001	0011	1111	1110	
5th shift right	0100	1001	1111	1111	0
6th shift right (carry = 0)	0010	0100	1111	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1000	0100	1111	1110	
7th shift right	0100	0010	0111	1111	0
8th shift right (carry = 0)	0010	0001	0011	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1000	0001	0011	1110	
Next byte of the message (07)			0000	0111	
Exclusive OR (shift = 8)	1000	0001	0011	1001	
1st shift right	0100	0000	1001	1100	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1110	0000	1001	1101	
2nd shift right	0111	0000	0100	1110	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1101	0000	0100	1111	
3rd shift right	0110	1000	0010	0111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1100	1000	0010	0110	
4th shift right	0110	0100	0001	0011	0
5th shift right (carry = 0)	0011	0010	0000	1001	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1001	0010	0000	1000	
6th shift right	0100	1001	0000	0100	0
7th shift right (carry = 0)	0010	0100	1000	0010	0
8th shift right (carry = 0)	0001	0010	0100	0001	0
CRC error check code	**12h**		**41h**		

The final message transmitted including the CRC code is:

Device Address		Function Code		CRC MSB		CRC LSB	
02h		07h		41h		12h	
0000	0010	0000	0111	0100	0001	0001	0010

↑ First bit Transmission order Last bit ↑

Example of a CRC Calculation in the "C" Language

This routine assumes that the data types "uint16" and "uint8" exist. These are unsigned 16 bit integer (usually an "unsigned short int" for most compiler types) and unsigned 8 bit integer (unsigned char).

"z_p" is a pointer to a Modbus message, and z_message_length is its length, excluding the CRC.

Note that the Modbus message will probably contain "NULL" characters and so normal C string handling techniques will not work.

```
uint16 calculate_crc (uint8 *z_p, uint16 z_message_length)

/* CRC runs cyclic Redundancy Check Algorithm on input z_p */
/* Returns value of 16 bit CRC after completion and        */
/* always adds 2 crc bytes to message                      */
/* returns 0 if incoming message has correct CRC           */

{
   uint16 CRC = 0xffff;
   uint16 next;
   uint16 carry;
   uint16 n;
   uint8 crch, crcl;

   while (z_message_length--) {
      next = (uint16)*z_p;
      CRC ^= next;
      for (n = 0; n < 8; n++) {
            carry = CRC & 1;
            CRC >> = 1;
            if (carry)  {
                  CRC ^= 0xa001;
            }

      }
      z_p++;

   }
   crch = CRC / 256;
   crcl = CRC % 256;
   *z_p++ = crcl;
   *z_p = crch;
   return CRC;
}
```

Example of a CRC Calculation in Basic Language

```
Function CRC (messages) as long
'' CRC runs Cyclic Redundancy Check Algorithm on input message$
'' Returns value of 16 bit CRC after completion and
'' always adds 2 crc bytes to message
'' returns 0 if incoming message has correct CRC

   '' Must use double word for CRC and decimal constants

   crc16& = 65535
   FOR c% = 1 to LEN(message$)
      crc16& = crc16& XOR ASC(MID$(message$, c%, 1))
      FOR bit% = 1 to 8
            IF crc16& MOD 2 THEN
               crc16& = (crc16& \ 2) XOR 40961
            ELSE
               crc16& = crc16& \ 2
            END IF
      NEXT BIT%
   NEXT c%
   crch% = CRC16& \ 256: crcl% = CRC16& MOD 256
   message$ = message$ + CHR$(crcl%) + CHR$(crch%)
   CRC = CRC16&
END FUNCTION CRC
```

Function Codes

Function codes are a single byte instruction to the Slave describing the action to perform.

The following communication functions are supported by Eurotherm Drives' units:

Function Code	Function
01 or 02	Read n bits
03 or 04	Read n words
05	Write 1 bit
06	Write 1 word
08	Loopback
15	Write n bits
16	Write n words

Read n Bits

Function Code: 01 or 02, (01h or 02h)

Command:

Device Address	Function Code 01 or 02	Address of 1st bit		Number of bits to read		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The maximum number of bits that may be read is 512.

Reply:

Device Address	Function Code 01 or 02	Number of bits to read	First byte of data	....	Last byte of data	CRC	
1 byte	1 byte	1 byte	1 byte	....	1 byte	MSB	LSB

The first data byte contains the status of the first 8 bits, with the least significant bit being the first bit. The second data byte contains the status of the next 8 bits, etc. Unused bits are set to zero.

Example

From the unit at device address 02, read 14 parameters, beginning at Tag 640:

Command:

Device Address	Function Code 01 or 02	Address of 1st bit		Number of bits to read		CRC	
02	01	02	7F	00	0E	8D	97

Reply:

Device Address	Function Code 01 or 02	Number of bytes read	First byte of data	Last byte of data	CRC	
02	01	02	27	03	A6	0D

An expansion of the data bytes illustrates the relationship between data and the parameter addresses.

Data byte	1st byte (27h)								2nd byte (03h)							
Param. address	647	646	645	644	643	642	641	640			653	652	651	650	649	648
Bit values	0	0	1	0	0	1	1	1	0	0	0	0	0	0	1	1

Read n Words

Function Code: 03 or 04, (03h or 04h)

Command:

Device Address	Function Code 03 or 04	Address of 1st word		Number of words to read		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The maximum number of words that may be read is 32.

Reply:

Device Address	Function Code 03 or 04	Number of bytes read	Value of 1st word		....	Value of last word		CRC	
1 byte	1 byte	1 byte	MSB	LSB	....	MSB	LSB	MSB	LSB

Example

For a 605 Inverter at device address 02, read 2 parameters beginning at Tag 254 (Speed Setpoint and Speed Demand). SPEED SETPOINT is 100.00% and SPEED DEMAND is 50.00%.

Command:

Device Address	Function Code 03 or 04	Address of 1st word		Number of words to read		CRC	
02	03	00	FD	00	02	55	C8

Reply:

Device Address	Function Code 03 or 04	Number of bytes read	Value of 1st word		Value of last word		CRC	
02	03	04	27	10	13	88	CF	14

Write 1 Bit

Function Code: 05, (05h)

Command:

Device Address	Function Code 05	Address of bit		Value of bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The LSB of "Value of bit" is always set to 00. The MSB is used to write the value of the addresses bit. To set a bit value of 1, either transmit 01h or FFh. To set a bit value of 0 transmit 00h.

A device address 00 will broadcast the data to all devices on the network.

Reply:

(There will be no reply to a command broadcast to the device address 00.)

Device Address	Function Code 05	Address of bit		Value of bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The reply to function 05 is the same as the command.

Example

Write to the unit at device address 02 setting the parameter with Tag 3 to be TRUE..

Command:

Device Address	Function Code 05	Address of bit		Value of bit		CRC	
02	05	00	02	01	00	6D	A9

Reply:

Device Address	Function Code 05	Address of bit		Value of bit		CRC	
02	05	00	02	01	00	6D	A9

Write 1 Word

Function Code: 06, (06h)

Command:

Device Address	Function Code 06	Address of word		Value of word		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

A device address 00 will broadcast the data to all devices on the network.

Reply:

(There will be no reply to a command broadcast to the device address 00.)

Device Address	Function Code 06	Address of word		Value of word		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The reply to function 06 is the same as the command.

Example

For a 605 Inverter at device address 02, write 20.0 to RAMP ACCEL RATE (Tag 258).

Command:

Device Address	Function Code 06	Address of word		Value of word		CRC	
02	06	01	01	00	C8	D8	53

Reply:

Device Address	Function Code 06	Address of word		Value of word		CRC	
02	06	01	01	00	C8	D8	53

Diagnostic Loopback

Function Code: 08, (08h)

This function provides a means of testing the communications link by means of a “loopback” operation. The data sent to the unit is returned unchanged. Only diagnostic code 0 from the Gould Modicon Specification is supported.

Command:

Device Address	Function Code 08	Diagnostic Code 0000		Loopback Data		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Reply:

The reply to function 08 is the same as the command.

Example

Perform a loopback from the unit at address 02 using a data value of 1234h.

Command:

Device Address	Function Code 08	Diagnostic Code 0000		Loopback Data		CRC	
02	08	00	00	12	34	ED	4F

Reply:

Device Address	Function Code 08	Diagnostic Code 0000		Loopback Data		CRC	
02	08	00	00	12	34	ED	4F

Write n Bits

Function Code: 15, (0Fh)

Command:

Device Address	Function Code 0F	Address of 1st word		Number of bits to write		Number of data bytes (n)	Data	CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	1 byte	n bytes	MSB	LSB

The maximum number of bits that may can be transmitted is 512.

A device address 00 will broadcast the data to all devices on the network.

Reply:
(There will be no reply to a command broadcast to the device address 00).

Device Address	Function Code 0F	Address of 1st word		Number of bits written		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example

Write to the Slave unit, at device address 02, 14 parameters beginning at Tag 640 the values 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0.

Command:

Device Address	Function Code 0F	Address of 1st word		Number of bits to write		Number of data bytes (n)	Data	CRC	
02	0F	02	7F	00	0E	02	see below	83	06

Data byte	1st byte (27h)							
Param. address	647	646	645	644	643	642	641	640
Bit values	0	0	1	0	0	1	1	1

Data byte	2nd byte (03h)							
Param. address			653	652	651	650	649	648
Bit values	0	0	0	0	0	0	1	1

Reply:

Device Address	Function Code 0F	Address of 1st word		Number of bits written		CRC	
02	0F	02	7F	00	0E	E4	5C

Write n Words

Function Code: 16, (10h)

Command:

Device Address	Function Code 10	Address of 1st word		Number of words to write		Number of data bytes (n)	Data	CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	1 byte	n bytes	MSB	LSB

The maximum number of words that may can be transmitted is 32.

The first 2 bytes are data with the required value of the first parameter, MSB first. Following pairs are data for the consecutive parameter addresses.

A device address 00 will broadcast the data to all devices on the network.

Reply:

(There will be no reply to a command broadcast to the device address 00).

Device Address	Function Code 10	Address of 1st word		Number of words written		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example

605 Inverter: write to the Slave unit at device address 02

Tag 258 RAMP ECCEL RATE = 20.0
Tag 259 RAMP DECEL RATE = 15.0

Command:

Device Address	Function Code 10	Address of 1st word		Number of words to write		Number of data bytes (n)	Data	CRC	
02	10	01	01	00	02	04	see below	31	27

Data (200) for Tag 258		Data (150) for Tag 259	
00	C8	00	96

Reply:

Device Address	Function Code 10	Address of 1st word		Number of words written		CRC	
02	10	01	01	00	02	11	C7

Error Response

The MODBUS protocol defines the response to a number of error conditions. A Slave device is able to detect a corrupted command or one that contains an incorrect instruction, and will respond with an error code.

With some errors, the Slave devices on the network are unable to make a response. After a wait period, the Master will interpret the failure to reply as a communications error. The Master should then re-transmit the command.

A Slave device that has detected a corrupted command, or a command that contains in incorrect instruction, will respond with an error message. The error message has the following syntax:

Device Address	Function Code	Error Response Code	CRC	
1 byte	1 byte	1 byte	MSB	LSB

The Function Code byte contains the transmitted function code but with the most significant bit set to 1. (This is the result of adding 128 to the function code.)

The error response code indicates the type of error detected. The following error response codes are supported by Eurotherm Drives' units:

Code	Error	Description
01	Illegal Function	The requested function is not supported by the slave.
02	Illegal Data Address	The address referenced in the data field is not an allowable address for the Slave
03	Illegal Data Value	The value referenced in the data field is not allowable in the addressed Slave location
06	Host Busy	The slave cannot precess the request at this time. Try again later.
07	NAK	Rejected for an unspecified reason.

Wait Period

There are several errors for which the Slave devices on the network are unable to make a response:

- If the Master attempts to use an invalid address then no Slave device will receive the message
- For a message corrupted by interference, the transmitted CRC will not be the same as the internally calculated CRC. The Slave will reject the command and will not reply to the Master.

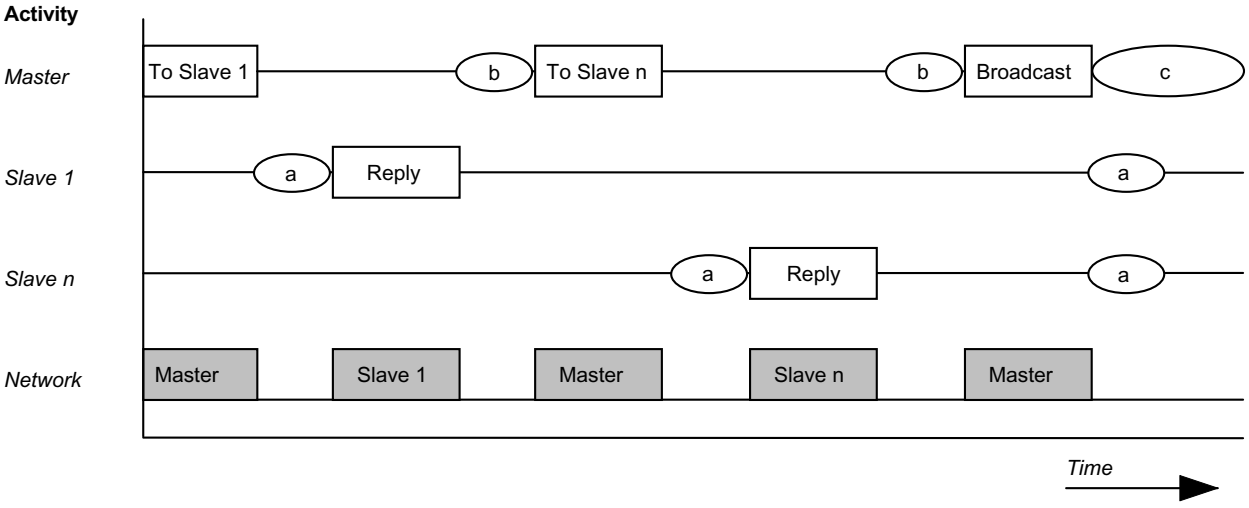
After a wait period, the Master will re-transmit the command.

A wait period is also required after a broadcast communication to device address 0.

IMPORTANT: Failure to observe the wait period after a broadcast will negate the broadcast message.

Typical Transmission Line Activity

This diagram illustrates a typical sequence of events on a Modbus transmission line.



Period “a” The processing time (latency), required by the Slave to complete the command and construct a reply. This is typically 2 milliseconds.

Period “b” The processing time required by the Master to analyse the Slave response and formulate the next command.

Period “c” The wait time calculated by the Master for the Slaves to perform the operation. None of the Slaves will reply to a broadcast message.

MODBUS RTU Parameter Mapping

1. MODBUS RTU Prime Set

Mnemonic	Description	Range (HEX values)	Access
9901	Instrument Identity	0605, 0584 or 5900	Read Only
9902	Main Software Version	0000 to FFFF	Read Only
9903	6051 Software Version	0000 to FFFF (0000 if not fitted)	Read Only
9904	Technology Box 1 Software Version	0000 to FFFF	Read Only
9905	Technology Box 2 Software Version	0000 to FFFF	Read Only
9909	Last Tag Number	0000 to FFFF	Read Only

2. Command/Status

Mnemonic	Description	Range (HEX values)	Access
9911	Command	see below (!1)	Write Only
9912	State	see below (!2)	Read Only
9913	Save Command	see below (!3)	Write Only
9914	Save State	see below (!4)	Read Only

!1 : Command

Write-only: used to modify the state of the Inverter and to load configuration data from non-volatile memory.

HEX Value	Description
7777	Reset Command. Acknowledges failed restore. Loads and saves (590+ does not save) default Product Code and default Configuration (Macro 1).
0101	Restores Saved Configuration from drive's non-volatile memory.
0110	Restores Default Configuration (Macro 0) - *not 590+*
0111	Restores Default Configuration (Macro 1)
0112	Restores Default Configuration (Macro 2) - *not 590+*
0113	Restores Default Configuration (Macro 3) - *not 590+*
0114	Restores Default Configuration (Macro 4) - *not 590+*
01A9	Restores Default Configuration (Macro 99) - *584SV only*
4444	Exit Configuration Mode
5555	Enter Configuration Mode

!2 : State

Read-only: used to determine the major state of the Inverter.

HEX Value	Description
0000	Initialising. (Powering up)
0001	Corrupted Product Code and Configuration
0002	Corrupted Configuration
0003	Restoring Configuration
0004	Re-Configuring Mode
0005	Normal Operation Mode

!3 : Save Command Write-only: used to save the configuration and product code in non-volatile memory.	
HEX Value	**Description**
0000	Reset Command. Acknowledges (clears) any previous save error.
0001	Saves Configuration to drive's non-volatile memory.
0100	Saves Product Code to drive's non-volatile memory.

!4 : Save State Read only: used to determine the progress of a non-volatile saving operation.	
HEX Value	**Description**
0000	Idle
0001	Saving
0002	Failed

3. Tag Access

Each parameter is directly mapped to four MODBUS registers: two of these represent it as a single data bit, and the other two represent it as a 16-bit signal or unsigned data word.

This allows a parameter to be read and written using the MODBUS bit functions (01, 02, 05 and 15) or word functions (03, 04, 06 and 16).

For example, the parameter with Tag 65 in the drive is mapped to register:

Bit Functions		
{0}0065	as a COIL STATUS REGISTER for access using functions :	(01, 05, 15)
{1}0065	as an INPUT STATUS REGISTER for access using function :	(02)
Word Functions		
{4}0065	as a HOLDING REGISTER for access using functions :	(03, 06, 16)
{3}0065	as an INPUT REGISTER for access using function :	(04)

4. PNO Access (590+ and 590+DRV only)

Parameters may also be accessed using the register number derived from the " EI Bisynch Binary Parameter Specification Tables" page 37.

> For example, PNO 39 can be accessed as register X1039 (i.e. PNO + 1000) where {X} is {0}, {1}, {4} or {3}.

5. Encoding

All parameters may be accessed except for those of type STRING.

Reading a parameter which is not of type BOOLEAN using a bit function (01 or 02) will return 1 if the value is non-zero. Writing to parameter which is not of type BOOLEAN using a bit function (05 or 15) will set the value to either 0 or 1 if the limits of the parameter allow this.

Troubleshooting

LED Indications		Cause/Symptom	Remedy
NETWORK	MODULE		
(OFF)		No power at the drive.	Check and apply power to the drive.
		Technology Box/Option not installed correctly.	Check connections between Technology Box/Option and drive. On 605A & B, check the ribbon cable.
		Hardware fault. 605A & B WARNING: Remove the terminal cover and the Technology Box whilst connected to see the drive's HEALTH and RUN LEDs. BEWARE OF ELECTRIC SHOCK.	If HEALTH and RUN LEDS are OFF, replace the drive, else replace the Technology Box/Option.
		The self-test has failed.	Replace the Technology Box/Option.
		Incorrect Technology Box/Option fitted or selected.	Fit the correct Technology Box/Option or select the matching value for the TYPE parameter in the TEC OPTION function block. (TYPE = RS485).
		Set-up fault. A TEC OPTION parameter is out-of-range.	Select the correct value for the parameter in the TEC OPTION function block.
		Wiring to RXA and RXB terminals is transposed.	Correct the wiring to the RXA and RXB terminals.
		No data is being received from PLC/SCADA.	Enable the PLC/SCADA application program.
		No data is being received from PLC/SCADA.	Check power for all equipment on the network, e.g. RS232 to RS485 converter or repeater.
		Baud rate incorrect.	Set the same baud rate on the drive and PLC/SCADA.
		Incorrect data format.	Check the PLC/SCADA has 7 data bits selected.
		Incorrect parity.	Check the PLC/SCADA has even parity selected.
		Drive not being addressed.	Check the GID and UID drive address matches the address sent by the PLC/SCADA.
		ERROR CODE = 00C0 PLC/SCADA receives invalid/corrupted reply.	Check the GID and UID drive address is unique to the network.
*		ERROR CODE = 00C0 Wiring from TXA/TXB incorrect (4-wire only)	Correct the TXA/TXB wiring.
*		ERROR CODE - 00C0 (2-wire only)	Ensure that SW1 is set for 2-wire operation.
*		ERROR CODE = 01C7 Mnemonic from PLC/SCADA not recognised.	Send the correct mnemonic from the PLC/SCADA.
*		ERROR CODE = 02C2 Drive received an incorrect checksum.	Check (BCC) if manually entered, or try sending the message again.

* *If this is an intermittent problem, it may indicate poor wiring and/or poor cable routing in an electrically `noisy' environment. Also check that terminating resistors are present and correctly set.*

LED Indications		Cause/Symptom	Remedy
NETWORK	MODULE		
[redacted]	[redacted]	ERROR CODE = 04C8 PLC/SCADA tried to read from a write-only parameter.	Correct the PLC/SCADA program so that it doesn't try to read from a write-only parameter.
[redacted]	[redacted]	ERROR CODE = 05C8 PLC/SCADA tried to write to a read-only parameter.	Correct the PLC/SCADA program so that it doesn't try to write to a read-only parameter.
[redacted]	[redacted]	ERROR CODE = 07C8 PLC/SCADA sent a message with invalid data format.	Correct the PLC/SCADA to send the correct data format for the parameter in question.
[redacted]	[redacted]	ERROR CODE = 08C8 PLC/SCADA sent a value outside the permissible range of the parameter.	Correct the PLC/SCADA program so that it doesn't send out-of-range parameter values.

ASCII Table

BINARY				b_6	0	0	0	0	1	1	1	1
				b_5	0	0	1	1	0	0	1	1
				b_4	0	1	0	1	0	1	0	1
b_3	b_2	b_1	b_0	HEX	0x	1	2	3	4	5	6	7
0	0	0	0	x0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC_1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC_2	“	2	B	R	b	r
0	0	1	1	3	ETX	DC_3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC_4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	‘	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
1	0	1	1	B	VT	ESC	+	;	K	[	k	{
1	1	0	0	C	FF	FS	,	<	L	\	l	|
1	1	0	1	D	CR	GS	-	=	M	]	m	}
1	1	1	0	E	SO	RS	.	>	N	^	n	~
1	1	1	1	F	SI	US	/	?	O	_	o	DEL