Process Supervisor

Product Manual

Contents of binder

This binder contains item 1 from the list below, together with items 2 and/or 3 as required.

1	Process Supervisor Handbook	HA261231
2	T2500 I/O Module Product Manual	HA026178
3	LIN/ALIN Installation and User Guide	HA082429U005

Eurotherm Limited pursues a policy of continuous development and product improvement. The specifications in this document may therefore be changed without notice. The information in this document is given in good faith, but is intended for guidance only. Eurotherm Limited will accept no responsibility for any losses arising from errors in this document.

invensys
An Invensys company

Issue 5 Aug 01 Part Number HA261726

Eurotherm Ltd

Head Office
Eurotherm Ltd,
Faraday Close,
Durrington,
Worthing, West Sussex, BN13 3PL
Tel: +44 (0) 1903 695888
Fax: +44 (0) 1903 695666
e-mail: info@eurotherm.co.uk
Web site: http://www.eurotherm.co.uk

Regional Office
GWENT
Units 25-27, Raglan House,
Llantarnam Park, Cwmbran,
Gwent NP44 3AB.
Tel: +44 (0) 1633 876633
Fax: +44 (0) 1633 872911

Overseas Companies

AUSTRALIA
Eurotherm Pty Ltd.
Tel: 61 2 9634 8444
Fax: 61 2 9634 8555

AUSTRIA (Vienna)
Eurotherm GmbH
Tel: 431 798 7601
Fax: 431 798 7605

BELGIUM (Antwerpen)
Eurotherm BV
Tel: 323 320 8550
Fax: 323 321 7363

DENMARK (Fredriksberg)
Eurotherm Danmark A/S
Tel: 45 38-871622
Fax: 45-38-872124

FINLAND (Åbo)
Eurotherm Finland
Tel: 35 8 22 50 60 30/1
Fax: 35 8 22 50 32 01

FRANCE (Lyon)
Eurotherm Automation SA
Tel: 33 4 78 66 4500
Fax: 33 4 78 35 2490

GERMANY (Limburg)
Eurotherm Regler GmbH
Tel: 49 6431 2980
Fax: 49 6431 298119

HONG KONG
Eurotherm Ltd.
Tel: 852 2873 3826
Fax: 852 2870 0148

INDIA (Chennai)
Eurotherm DEL India Ltd
Tel: 91 44 496 1129
Fax: 91 44 496 1831

ITALY (Como)
Eurotherm SpA
Tel: 39 031 975111
Fax: 39 031 977512

JAPAN (Tokyo)
Densei-Lambda KK
Tel: 81 35714 0620
Fax: 81 35714 0621

KOREA (Seoul)
Eurotherm Korea Ltd
Tel: 82 2 478 8507
Fax: 82 2 488 8508

NETHERLANDS
Eurotherm B.V.
Tel: 31 172 411 752
Fax:31 172 417 260

NORWAY (Lysaker)
Eurotherm AS
Tel: 47 67 592170
Fax: 47 67 118301

SPAIN (Madrid)
Eurotherm España SA,
Tel: 34 91 6616001
Fax: 34 91 6619093

SWEDEN (Malmo)
Eurotherm AB
Tel: 46 40 384500
Fax: 46 40 384545

SWITZERLAND (Freienbach)
Eurotherm Produkte (Schweiz) AG,
Tel: 41 55 415 4400
Fax: 4155 415 4415

USA (Leesburg)
Eurotherm Controls Inc.
Tel: 1 703 443 0000
Fax: 1 703 669 1300

Agencies

AFRICA (Egypt)
Industrial Instrumentation Services
Tel: 201 536 2507
Fax: 201 536 8509

CZECH REPUBLIC (Prague)
e-therm s.r.o.
Tel: 422 683 3617
Fax: 422 683 3617

POLAND
Energotechnika
Tel: 4832 135 415
Fax: 4832 135 0790

ROMANIA (Constanta)
Paragon
Tel: 4041 831 293
Fax: 4041 831 298

RUSSIA (Moscow)
TCS Russia Ltd
Tel: 709 524 20801
Fax: 709 524 59982

SOUTH AMERICA (Chile)
Claudio Juppet y Cia Ltda.
Tel: 562 233 6655
Fax: 562 251 9020

TURKEY (Istanbul)
Entek A.S.
Tel: 90-1-3628272/73
Fax: 90-1-3801370

Aug 2001

Process Supervisor

Handbook

invensys

An Invensys company

ISSUE STATUS OF THIS MANUAL

Section	Issue
Title page	5
Preamble	5
Chapter 1	3
Chapter 2	5
Chapter 3	5
Chapter 4	5
Chapter 5	1
Chapter 6	5
Chapter 7	5
Chapter 8	5
Chapter 9	4
Chapter 10	4
Index	5

Notes

1 Sections are up-dated independently and so may be at different issues.

2 The Title page, new or updated sections and the manual as a whole, always take the issue number of the most recently updated section.

3 Within a section, pages may be at different issues. This happens if those pages have been individually up-issued and retro-fitted into an existing manual to bring it up-to-date.

Contents PROCESS SUPERVISOR HANDBOOK

TOP LEVEL CONTENTS LIST

FULL CONTENTS LIST

Chapter 1 INTRODUCTION

Chapter 2 INSTALLATION

Chapter 3 USER INTERFACE

Chapter 4 START-UP

Chapter 5 CONFIGURATION

Chapter 6 ERROR CONDITIONS & DIAGNOSTICS

Chapter 7 MODBUS

Chapter 8 PROFIBUS

Chapter 9 SERVICE

Chapter 10 Specification and Order Codes

GLOSSARY OF TERMS

Items in italics in the descriptions below also appear as glossary items in their own right

2500	I/O sub-system for use with Process Supervisor units
ALIN	Local Instrument Network (*LIN*) protocol on *ARCNET*
ALIN bridge	*LIN* to *ALIN* network link
Application	A *LIN database* and associated *SFCs*
ARCNET	A single non-branching, masterless network, running at 2.5M*Baud* allowing peer-to-peer communications and file transfer up to 100 metres.
Baud	Used to describe transmission speeds over communications links. (9600 baud = approximately 1000 ASCII characters per second)
Brown-out	A brown-out is a transient power variation or partial power failure severe enough to prevent continuation of the process until the process supervisor has been re-initialised.
Cold start	A Cold start is where the instrument starts with the last-loaded database loaded using either default parameters or parameters held in the cold start parameter file. See also *Hot Start*
Cold Start time	The Cold Start time is a pre-set duration, following power off, after which a *Hot Start* is not possible, and a *Cold Start* must be initiated instead.
Configuration	The process of specifying the components of an application.
Control strategy	A control strategy is the overall programmed function of the *LIN database* within an instrument, ready to act upon a real life process.
COSHH	Control of Substances Hazardous to Health legislation
CSP	Cold Start Primary - the left-hand *processor module*. Applies to *redundant mode* systems only.
CSS	Cold Start Secondary - the right-hand *processor module*. Applies to *redundant mode* systems only.
DRAM	Dynamic Random Access Mamory
Duplex	Twin synchronised processors capable of operating in *redundant mode*
EDB	External database
EEPROM	Electrically Erasable Programmable Read Only Memory
EMC	Electro-magnetic compliance

GLOSSARY (Cont.)

Eurotherm Project Studio	A suite of programs for building, testing and configuring programs and systems for process control and I/O.
ε-Suite	A control/monitoring/configuration system for use with process supervisor units.
FB	*Function block.*
FBD	*Function Block* Diagram - a programming language.
Function block	A unit of software that performs a named function. It can be linked to other function blocks to build a *LIN database* and hence a control strategy for an instrument.
GSD file	A GSD (Gerätestammdaten) file contains instrument parameter information, which a Profibus master needs in order to communicate with the instrument.
Hot start	After a power loss, the instrument attempts to re-start with the current database still loaded and with all parameters and values for that application still at the values they held when processing stopped. If the re-start fails the processor enters an *idle* state.
Hot & Cold start:	After a power loss, the instrument attempts to re-start with the current database still loaded and with all parameters and values for that application still at the values they held when processing stopped. If the re-start fails the processor attempts a *cold start*.
ICM	Inter-CPU Messaging for redundancy.
Idle	A state in which the *processor module* is powered up, but with an empty database. This state is entered as a result of '*test*' being selected as start-up mode, or if a *hot start* or *cold start* is not successful.
iTools	A Eurotherm utility for configuring networks of Eurotherm I/O controllers.
LIN	Local Instrument Network, a Eurotherm proprietary system for networking process monitoring and control instruments.
LIN database	The *LIN* database is a set of software function blocks that constitute the control strategy of a LIN instrument.
LIN protocol	The communications protocol employed to control instruments linked by a *LIN*.
LINtools	A Eurotherm utility for configuring networks of LIN instruments.
Modbus®	A proprietary communications protocol (Gould-Modicon Modbus RTU).

GLOSSARY (Cont.)

Non-redundant mode	One or more processors running but not *synchronised*.
PAL	Programmable Logic Array.
Primary	In a *Redundant mode* system, the primary is that processor which is in control. The other processor is called the *secondary* processor.
Processor module	The process supervisor consists of a backplane fitted with one or two Processor Modules and a connection module. 'Processor Module' should not be confused with Central Processor Unit (CPU) which is electronics hardware contained within the Processor module.
Process variable	Characteristics of a process - such as temperature, pressure and valve aperture - that can change value.
Profibus	A communications standard.
PSU	Power supply unit.
Redundant mode	Two *synchronised* processor modules (the *primary* and *secondary)*. The *secondary* processor tracks the *primary* in every respect so that it can take command should the primary (or the supply power to it) fail.
RFI	Radio frequency interference.
Secondary	In a *Redundant mode* system, the *primary* is that processor which is in control. The other processor is called the secondary processor and it continuously tracks the primary, so that it can assume control should the primary fail.
Synchronised	During the start up sequence in *redundant mode*, once the *primary* processor is running, it copies database and *function block* data to the *secondary*. Once this is complete, and the database is running in both *processor modules*, the *processor modules* are said to be synchronised.
SFC	Sequential Function Chart. An SFC monitors key variables and parameters and, on the basis of the values it finds, decides which route through a flowchart the application should follow.
Simplex	A processor working alone i.e in *non-redundant mode.*
SLIN	*LIN protocol* on a serial link (point-to-point).
SRAM	Static Random Access Memory.
Test start	Once started, the *processor module* enters an idle mode, with an empty data base loaded.

EUROTHERM PROCESS AUTOMATION

Declaration of Conformity

Manufacturer's name:	**Eurotherm Recorders Limited**
Manufacturer's address:	**Dominion Way, Worthing, West Sussex, BN14 8QL, United Kingdom.**
Product type:	**Process supervisor**
Models:	**T940 Processor module (Status level F2 or higher)** **T320 Connection module (Status level F2 or higher)** **T310 Backplane (Status level F2 or higher)**
Safety specification:	**EN61010-1: 1993 / A2:1995**
EMC emissions specification:	**EN50081-2 (Group 1; Class A)**
EMC immunity specification:	**EN50082-2**

Eurotherm Recorders Limited hereby declares that the above products conform to the safety and EMC specifications listed. Eurotherm Recorders Limited further declares that the above products comply with the EMC Directive 89 / 336 / EEC amended by 93 / 68 / EEC, and also with the Low Voltage Directive 73 / 23 / EEC

Signed: PR de la Nougerède Dated: 21 Dec. '99

Signed for and on behalf of Eurotherm Recorders Limited
Peter De La Nougerède
(Technical Director)

CE

IA249986U420 Issue 1 Dec 99

Chapter 1 INTRODUCTION

The process supervisor is one part of a complete control system. The entire package is described in the Eurotherm Project Studio User Guide and Tutorial HA261230 which includes a number of tutorial examples, to help users to familiarise themselves with the software and hardware facilities available.

1.1 MANUAL CONTENTS

This manual is divided into the following chapters:

Chapter 1. Introduction

Chapter 2. Installation

Chapter 3. User Interface (explaining the front panel LEDs and switches)

Chapter 4. Start Up (step-by-step instructions on how to start up or re-start the instrument)

Chapter 5. Configuration (how to configure, or more typically re-configure, control strategy and communications protocols on site, usually to match changes in the plant being controlled). (Initial configuration, to Customer Specification, is normally carried out prior to delivery.)

Chapter 6. Diagnostics (how to diagnose faults that could develop in the instrument, by recognising fault indications)

Chapter 7. Modbus communications

Chapter 8. Profibus communications

Chapter 9. Service

Chapter 10. Technical Specification and order codes

The contents of any other manuals in this binder are listed within those manuals.

1.2 OTHER INFORMATION SOURCES

For details of (LIN) based function blocks, their parameters and input/output connections refer to the LIN blocks reference section of the LIN product manual (HA082375U999) which explains how control strategy LIN blocks are selected, interconnected etc. The creation and monitoring of databases and communications configurations is described in the Eurotherm Project studio documentation. The configuration of Sequential Function Charts (SFCs) is described in the T500/550 LINtools User Guide (HA082377U005).

1.3 THE PROCESS SUPERVISOR UNITS

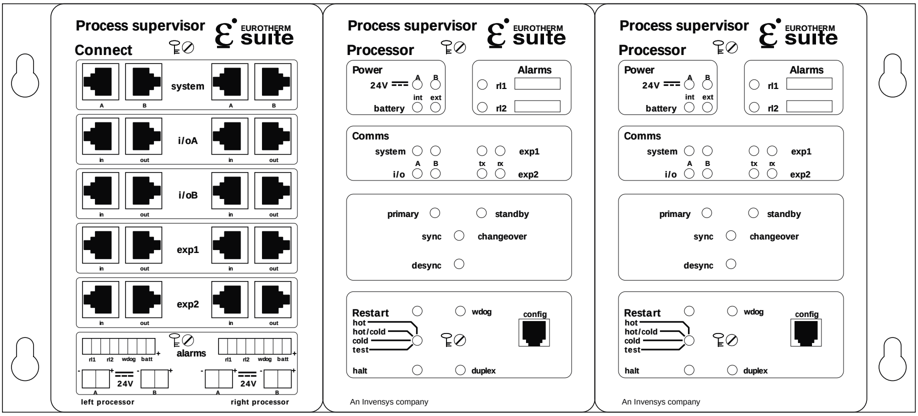


Figure 1.3
Connect module (left) and dual processor modules (centre and right) on the backplane

1.3.1 Typical applications

The process supervisor is designed to control processing plants using distributed input/output modules, interconnected using networks. A number of process supervisors can be networked together, allowing thousands of I/O points to be monitored and controlled.

1.3.2 Features

The main features of the process supervisor are as follows

- **ALIN.** ALIN-based network, allowing communications via a 'daisy-chain' configuration (or via a central ALIN 'hub') with I/O modules and the wider network. See figures 1.3.2a and 1.3.2b.

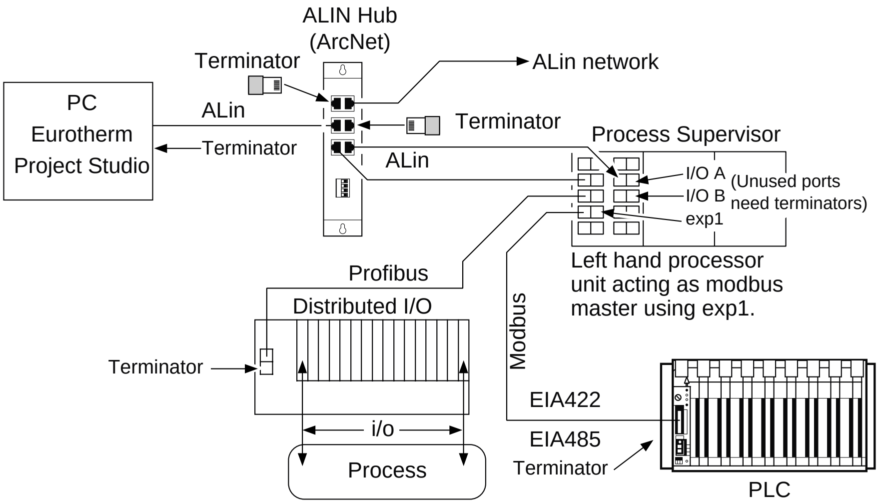


Figure 1.3.2a Typical communications architecture

- **Modbus.** The Unit supports Modbus comms via the connect module exp1 (master) and exp 2 (slave) ports if so configured.
- **Profibus.** The Unit supports Profibus communications via the connect module i/oB port.
- **Redundant processor modules.** The processors can be set up for redundant or non-redundant operation. When operating in redundant (duplex) mode, a high speed data link (ICM) between the primary and secondary processor units provides exact tracking of the control database, allowing bumpless takeover by the secondary unit should the primary processor fail.
- **Automatic takeover.** Takeover of control by the secondary processor in the event of primary failure is automatic, with no loss of I/O states and no need to re-initialise I/O points. Revalidation of all attached ALIN nodes is automatic.
- **Redundant power supply connection** Two independent power connectors for each processor unit, plus external battery for memory backup ensures full redundancy. An internal battery supports the data in SRAM and the real-time clock for a minimum of 72 hours.

1.3.2 FEATURES (Cont.)

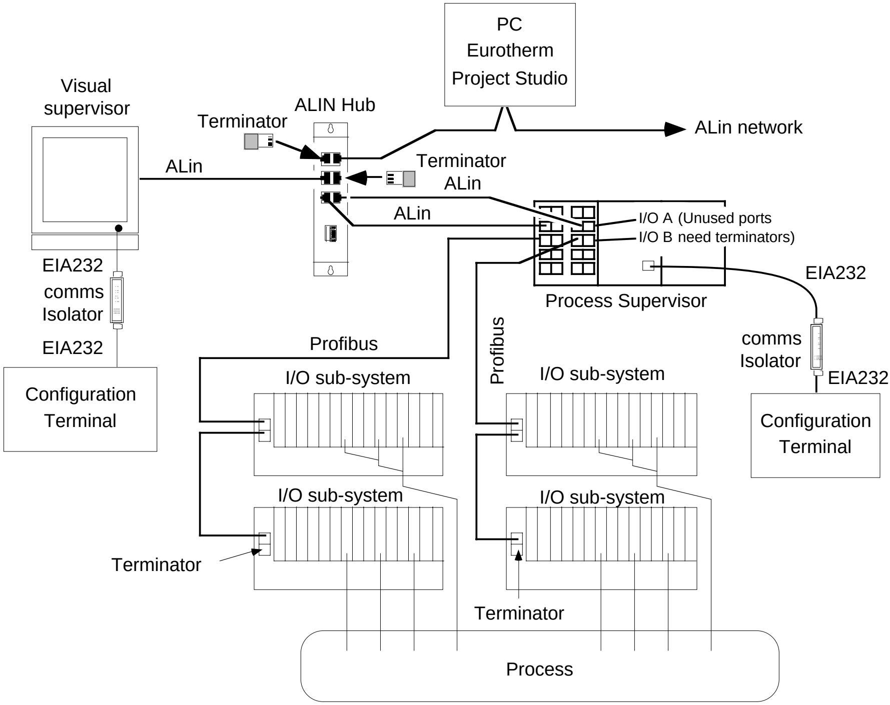


Figure 1.3.2b Extended control network schematic

- **Live processor replacement.** Live replacement of a failed processor can be carried out, with no wiring disconnections. The replacement unit loads its strategy and current status from the active processor. Full hardware and software status indication allows rapid verification and diagnostics.
- **Diagnostics.** Automatic health checks, self-testing, and initialisation on power-up.
- **Front-panel annunciation.** Front panel LEDs are provided for communications and processor status. Control switches are also fitted on each processor module.
- **Continuous health monitoring.** Extensive on-going diagnostics and health monitoring of communications and I/O status.
- **Watchdog.** Watchdog relay for each processor, with Connect module front-panel AND/OR connections.

1.3.2 FEATURES (Cont.)

- **I/O** Distributed I/O is networked using serial communications links.
- **Configuration** Strategies and sequences configured/downloaded/monitored with Eurotherm Project Studio or the resident configurator (needs external terminal).
- **Block structure** Continuous strategies are built up by interconnection of fixed function blocks from a comprehensive library of analogue and logic elements, common to all Eurotherm LIN/ALIN-based instruments.
- **ST user-algorithms** Special ACTION blocks support user-algorithms written in ST (Structured Text) and are well-suited to implement plant logical devices.
- **Block support** All standard LIN data base function blocks are supported in redundant mode. Special diagnostic blocks are available for hardware and software status reporting.
- **Enclosures** (Figure 1.3.2c). Process supervisor units can be supplied in a range of enclosures, both wall-mounted and floor-standing. Power supplies, standard terminations, transmitter power supplies, and I/O modules can all be fitted within these enclosures, and if required, a visual supervisor unit can be door mounted to allow a visual representation of process variables.

1.3.2 FEATURES (Cont.)

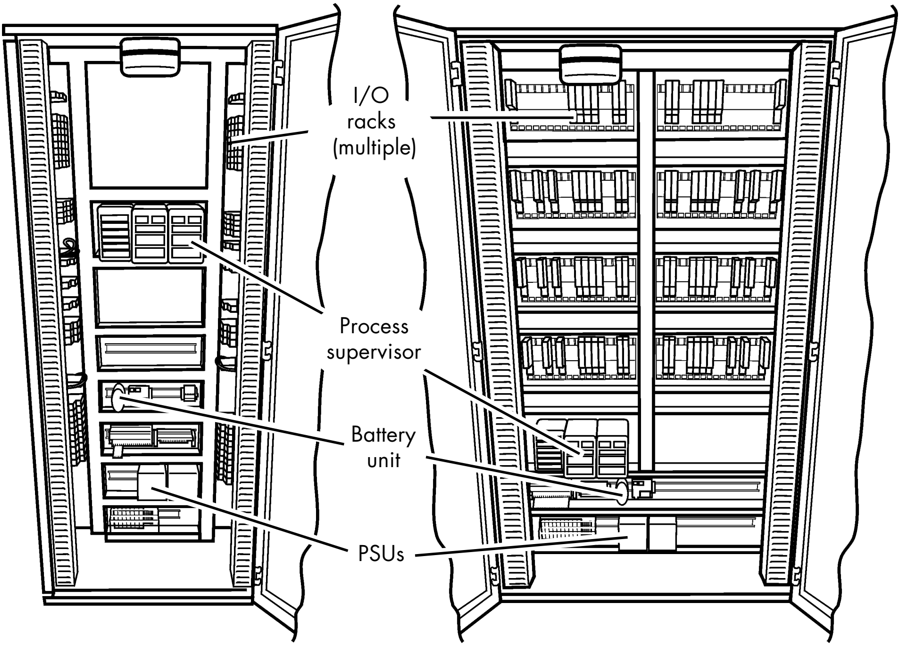


Figure 1.3.2c Typical installations

Note that the process interface i/o modules can be mounted vertically as shown in the sides of the single bay enclosure, or horizontally as shown in the two-bay version.

Chapter 2 INSTALLATION

This chapter presents safety and EMC information and tells you how to install and connect up your instrument.

The main topics covered are:

- Safety & EMC information (§2.1)
- Unpacking (§2.2)
- Mechanical layout (§2.3)
- Set-up switch definition (§2.3)
- Connections and wiring (§2.4)
- Use of configurator terminal (§2.4)

2.1 Safety & EMC information

Please read this section before installing the unit.

This unit meets the requirements of the European Directives on Safety and EMC as detailed on the Declaration of conformity IA249986U420, a copy of which appears in Chapter i of this manual. It is, however, the responsibility of the installer to ensure the safety and EMC compliance of any particular installation.

2.1.1 Installation requirements for EMC

This unit conforms with the essential protection requirements of the EMC Directive 89/336/EEC, amended by 93/68/EEC.

This unit satisfies the emissions and immunity standards for industrial environments.

To ensure compliance with the European EMC directive certain installation precautions are necessary as follows:

- **General guidance.** For general guidance refer to the Eurotherm Process Automation *EMC Installation Guide* (Part No. HG 083 635 U001).
- **Relay outputs.** When using relay outputs it may be necessary to fit a filter suitable for suppressing conducted emissions. The filter requirements will depend on the type of load.
- **Routing of wires.** To minimise the pick-up of electrical noise, low voltage DC connections and sensor input wiring should be routed away from high-current power cables. Where it is impractical to do this, shielded cables should be used, with the shield grounded at both ends.

2.1.2 Installation safety requirements

PERSONNEL

Installation must be carried out only by authorised personnel.

CONDUCTIVE POLLUTION

Electrically conductive pollution (e.g. carbon dust, water condensation) must be excluded from the enclosure in which the unit is mounted. To ensure the atmosphere is suitable, an air filter should be installed in the air intake of the enclosure. Where condensation is likely, a thermostatically controlled heater should be included in the enclosure.

VENTILATION

Ensure that the enclosure or cabinet housing the unit provides adequate ventilation/heating to maintain the operating temperature of the unit within the limits indicated in the Specification (see Chapter 10).

ELECTROSTATIC DISCHARGE HANDLING PRECAUTIONS

Caution

Electrostatic sensitivity. The circuit boards inside the units contain electrostatically sensitive components. To avoid damage, before you remove or handle any board ensure that you, the working area, and the board are electrostatically grounded. Handle boards only by their edges and do not touch the connectors.

2.1.3 Keeping the product safe

To maintain the units in a safe condition, observe the following instructions.

MISUSE OF EQUIPMENT

Note that if the equipment is used in a manner not specified in this handbook or by Eurotherm Process Automation, the protection provided by the equipment may be impaired.

SERVICE AND REPAIRS

Except for those parts detailed in Chapter 9, the Process Supervisor has no user-serviceable parts. Contact the nearest manufacturer’s agent for repair.

2.2 Unpacking

The instrument and accessories should be carefully unpacked and inspected for damage. The original packing materials should be retained in case re-shipment is required. If there is evidence of shipping damage, the supplier or the carrier should be notified within 72 hours and the packaging retained for inspection by the manufacturer's and/or carrier's representative.

2.2.1 Handling precautions

Caution

Electrostatic sensitivity. The circuit boards inside the units contain electrostatically sensitive components. To avoid damage, before you remove or handle any board ensure that you, the working area, and the board are electrostatically grounded. Handle boards only by their edges and do not touch the connectors.

2.2.2 Package contents

Note: The process supervisor may form part of a larger assembly, and/or may be housed in a floor or wall-mounted enclosure. If so, the documentation that accompanied those items should be referred-to.

The package contents should be checked against the order codes, using the labels on the components. Order codes are listed in Chapter 10 of this handbook.

PRODUCT LABELLING

Product labelling includes:

1 Sleeve label. On the outside of the processor and connect module sleeves, showing the model number , serial number, and hardware build level.

2 Backplane label. On the edge of the backplane, showing the model number, serial number, and hardware build level.

3 Software labels showing version and issue numbers.

4 Flash memory card label showing version and issue number.

5 Safety earth symbol adjacent to safety earth stud. 

2.3 MECHANICAL LAYOUT

Figure 2.3.1a shows two processor modules and a connect module mounted on the backplane. Remote I/O modules (described in the *2500 Controller User Manual*, HA026178) are connected to the processor modules using the i/oA and/or i/oB communications sockets of the connection module. Figures 2.3.1b and 2.3.1c show front views of the modules.

When only a single processor is fitted, it is recommended that the blanking plate supplied be fitted to the vacant slot, to maintain EMC emission/immunity specifications.

The processor modules can operate either independently (simplex), or else in ‘redundant’ (duplex) mode in which case one of the processors acts as a primary, backed up by the other processor (the secondary), which can take over from the primary at any time.

Power is supplied to each processor module by one or two external 24V (nom.) power supplies. The two supplies are effectively OR’d together within the processor module, so they can run in parallel, thus ensuring that the processor continues to operate even if one of the supplies fails.

A separate plug is available to allow the connection of an external battery (2.4 to 5.0 V), to maintain the contents of the SRAM and real-time clock during shut down. An internal battery can be fitted to maintain the SRAM/real-time clock for a minimum of 72 hours. Chapter 9 gives installation/replacement procedures for the internal battery, and Chapter 10 gives details of suitable batteries, both internal and external.

2.3.1 Layout drawings

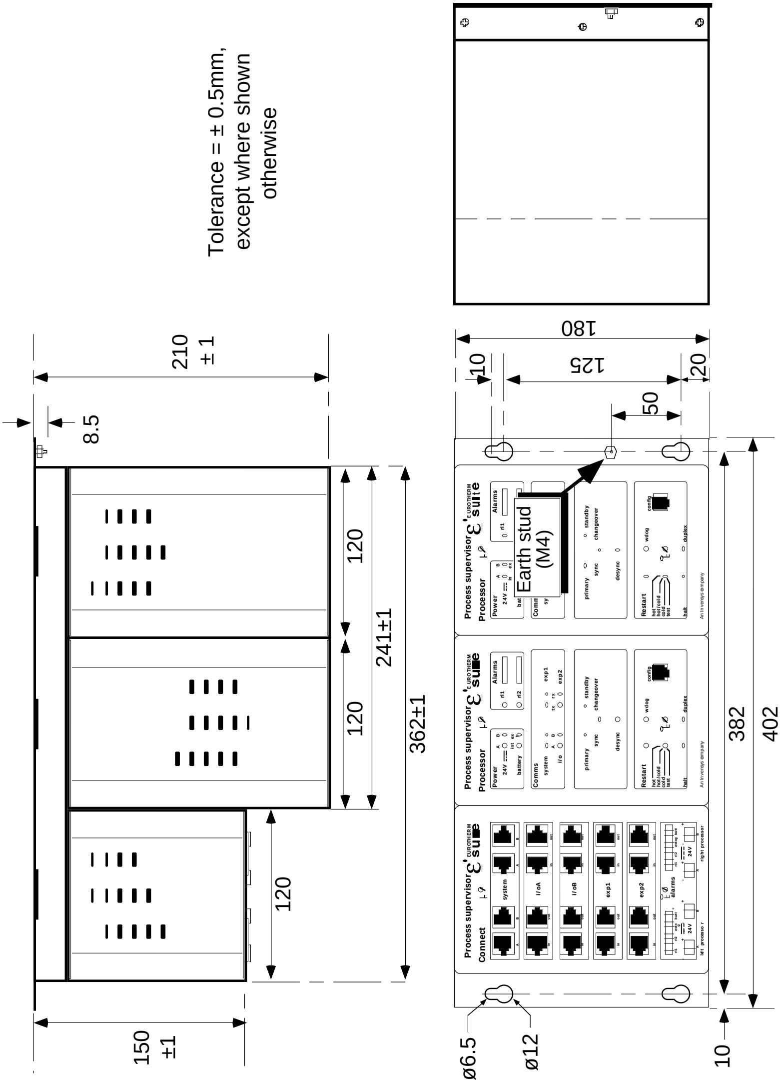


Figure 2.3.1a Dimensions (mm)

2.3.1 LAYOUT DRAWINGS (Cont.)

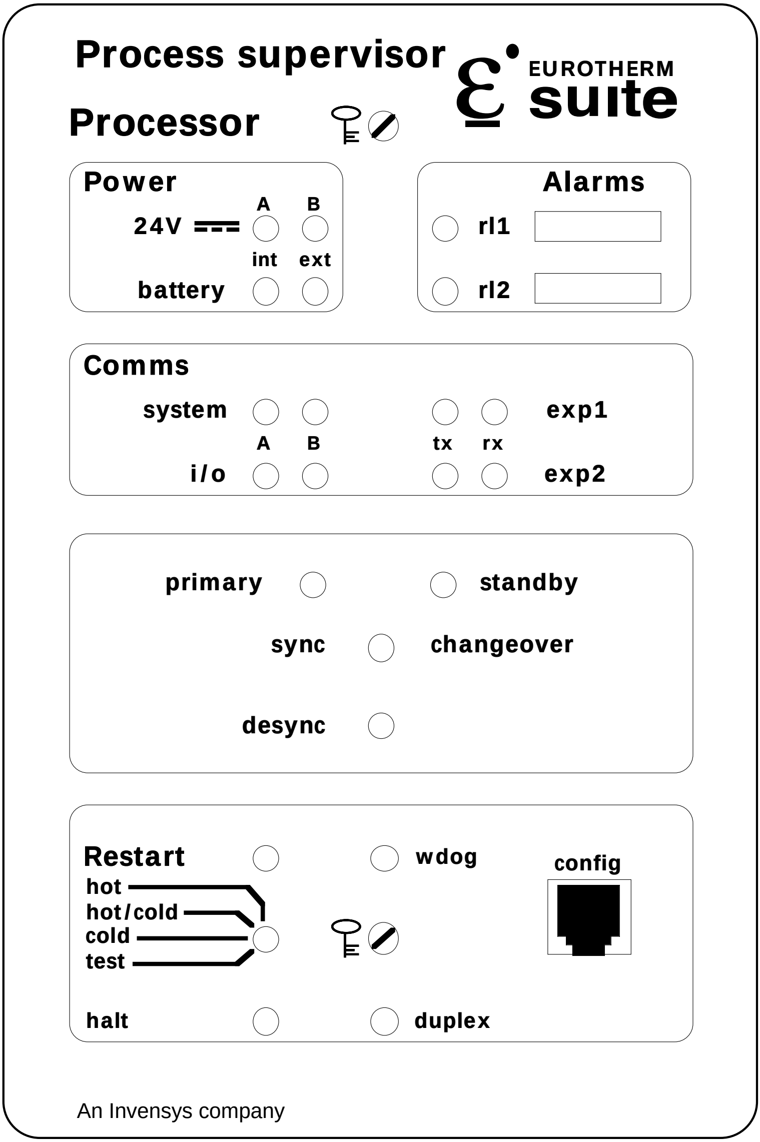


Figure 2.3.1b Processor module front panel layout

2.3.1 LAYOUT DRAWINGS (Cont.)

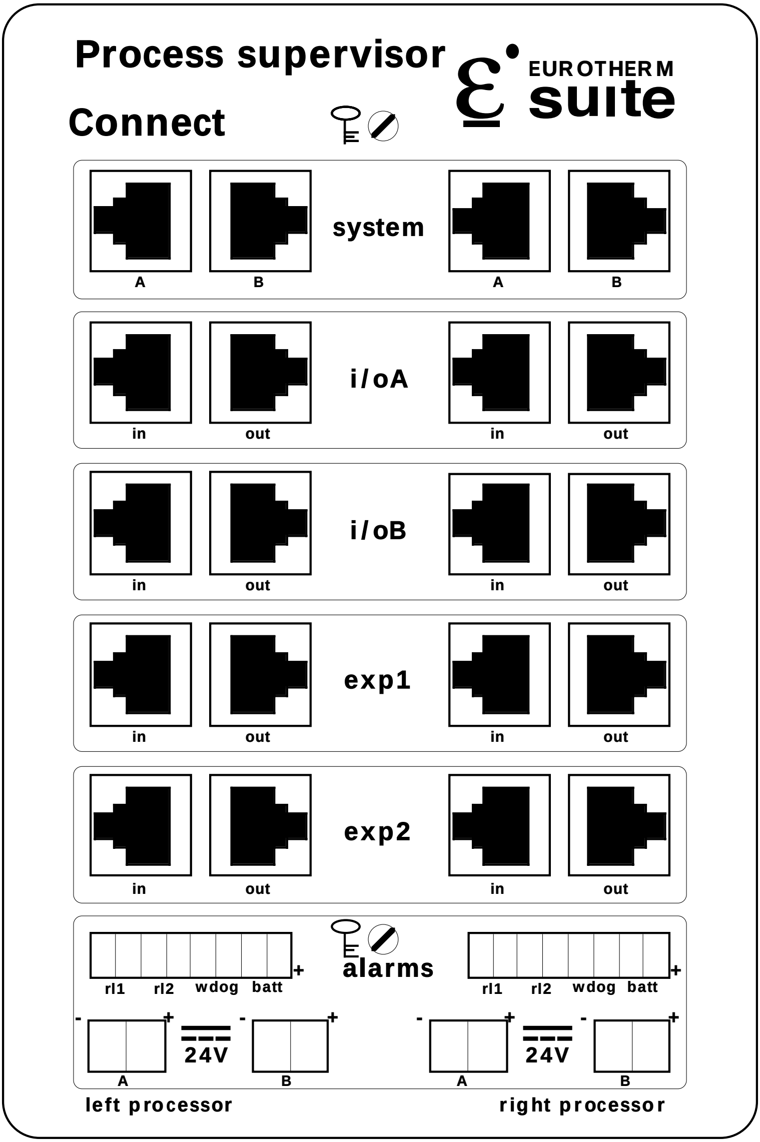


Figure 2.3.1c Connect module front panel layout.

2.3.2 Removal of modules

It is recommended that power be removed and all wiring be disconnected from the connection module, before it is removed from the backplane.

Although Processor Modules are designed to be removed/replaced with power applied, the life of the connector will be maximised if they are removed with power off.

Note: Figure 2.3.2 shows a connection module. The procedure is identical for processor modules.

To remove a module:

1 Remove wiring, by disconnecting connectors

2 Unscrew both retaining screws (anticlockwise) to jack the unit out of its connector.

3 Lift the unit off its retaining catch.

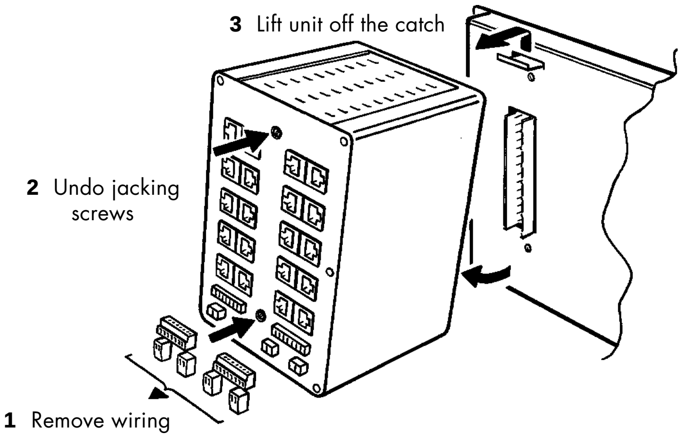


Figure 2.3.2 Module removal

2.3.3 Fitting of modules

1 Lift the module onto its retaining catch, and gently push the module towards the backplane to mate the connector.

Caution

Do not force the unit onto its connector or damage will occur

2 Re-engage and tighten both retaining screws a few turns each at a time, to a maximum torque of 2.5 Nm.

2.3.4 Location of backplane switches

The backplane switches for setting communications addresses and for selecting options on and off are revealed (figure 2.3.4) when the right-hand processor module is removed from the back plane.

The functions of these switches are described in section 2.3.5 below.

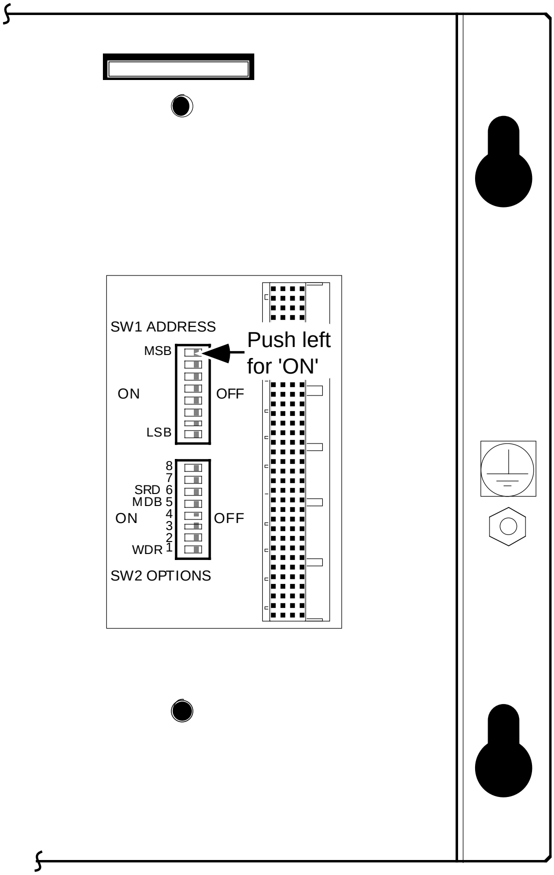


Figure 2.3.4 Location of backplane switches

2.3.5 Backplane switch functions

SW1: ALIN ADDRESS SETTING SWITCH

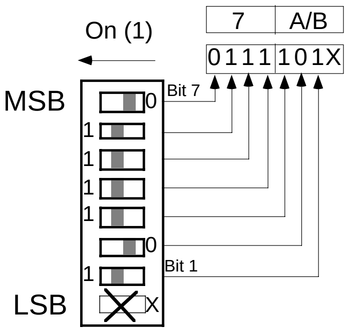
Figure 2.3.5a below shows the ALIN address-setting switch SW1 (located on the backplane as shown in figure 2.3.4 above). The figure shows a sample set up for address pair 7A/7B.

Whenever there are two processor modules fitted to the backplane and working in non-redundant mode, the left-hand processor unit is allocated the even address (Bit 0 = 0) and the right-hand processor is allocated the odd address (Bit 0 = 1).

When working in redundant mode, the primary processor is initially the left-hand (even address) unit and the secondary is initially the right-hand (odd address) unit. Should it prove necessary for the secondary to take over, and become the primary, it will also take over the even address.

NOTE. In redundant mode, a single processor module running on its own in the chassis never adopts the odd address as it is always the primary controller. It is strongly recommended that this odd address be kept ‘spare’ and not allocated to another instrument on the same ALIN segment. This will avoid address clashes if a second processor module is subsequently added to the backplane.

Sw1: ALIN Address



Binary	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

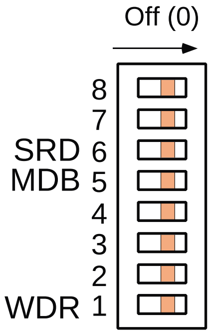
Example of how to set address pair 7A/7B
(Bit 0 automatically set to 0 (thus 7A) for left processor and to 1 (7B) for right processor)
Addresses 00, FE and FF are reserved and MUST NOT be used.

Figure 2.3.5a ALIN address setup example

2.3.5 BACKPLANE SWITCH FUNCTIONS (Cont.)

SW2: OPTIONS SWITCH

Figure 2.3.5b, below, shows the Options switch SW2 (located on the backplane as shown in figure 2.3.4, above).



Sw2 Options

8	Not used
7	Not used
6	Redundancy disable
5	Modbus enable
4	Not used
3	Not used
2	Not used
1	Watchdog retry enable

Figure 2.3.5b Option switch layout

WDR (WATCHDOG RETRY)

Setting this switch segment ‘on’ (slide to the left) causes the processor to try to start again after any watchdog failure. Setting the segment ‘off’ (slide to the right) disables the re-try and the processor will need manual restart after a watchdog failure.

MDB (MODBUS ENABLE)

Setting this switch segment ‘on’ (slide to the left) enables Modbus communications (if fitted). Setting the segment ‘off’ (slide to the right) disables Modbus Communications (see note 2 below).

SRD (REDUNDANCY DISABLE)

Setting this switch segment ‘off’ (slide to the right) selects redundant mode, with two processors defined initially as ‘primary’ (left-hand processor) and ‘secondary’ (right-hand processor). Setting the segment ‘on’ (left) disables redundancy mode, and both processors (if two are fitted) run independently.

Notes:
1. Sequential Flow Chart programs cannot be run in redundant mode.
2. Modbus communications are not available in redundant mode

2.4 CONNECTIONS AND WIRING

Units may be supplied mounted in an enclosure, together with the appropriate termination assemblies — either fitted in the enclosure or supplied in kit form. Please refer to the documentation that was supplied with the enclosure for details of the connections and wiring.

If you are assembling the system yourself, you should refer to the relevant *I/O Modules Reference Manual*, and the *LIN/ALIN Installation & User Guide* (HA082429 U005) for advice on connections and wiring to the I/O modules. For Profibus wiring advice, see section 8.5 of his manual

Figure 2.4 below shows a simplified overall connection diagram for a control system using an ALIN hub with RJ45 connections. Such a hub is useful for individual line lengths of up to 100 metres. For line lengths greater than this one or more pairs of hubs with fibre-optic connections is recommended. As detailed later in this chapter, it is also possible to connect local items together in series, using a daisy-chain technique, rather than in a star layout using a Hub.

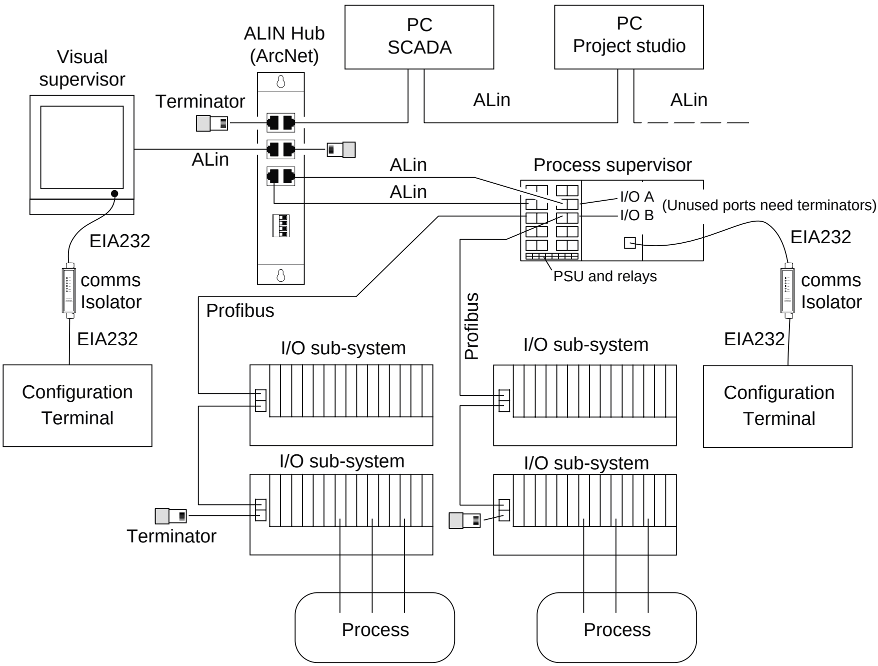


Figure 2.4 Typical overall connection diagram

2.4.1 Connect module

Apart from the connectors labelled ‘system’, which are for manufacturer’s use only, the RJ45 connectors on the front panel can be wired for ALIN, Modbus or Profibus use, according to specification at time of order. The pairs of connectors on the left hand side of the module are assigned to the left hand processor; the right hand connectors to the right-hand module. Each pair of connectors (except system A/B) is wired in parallel to provide for easy daisy chaining.

Plug-in modules to provide biassing components to terminate the transmission line, are available from the manufacturer. Such terminators are required only at the final node of the transmission line.

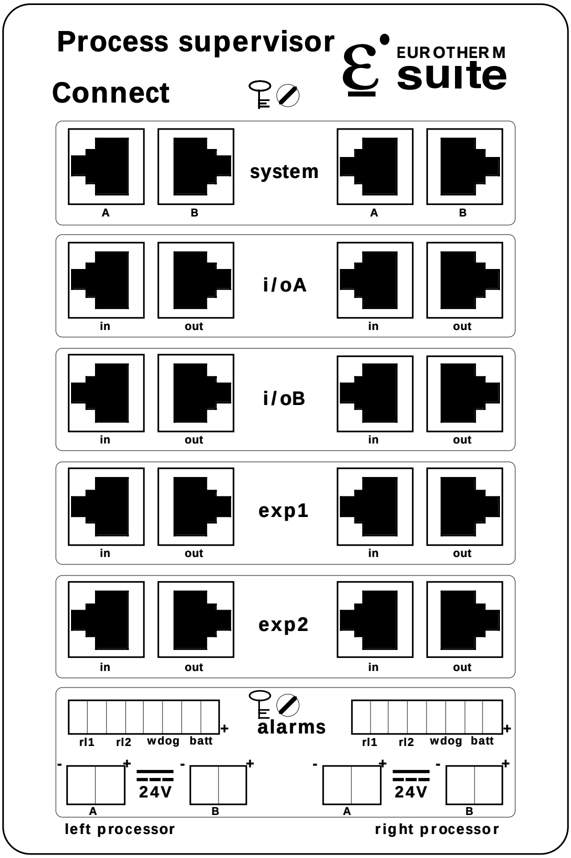
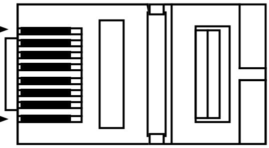


Figure 2.4.1a Connect module front panel

2.4.1 CONNECT MODULE (Cont.)

COMMUNICATIONS CONNECTORS

Figure 2.4.1b shows the connector pinouts for Modbus (EIA422 or EIA485), Profibus and ALIN standards. For the pinout for the processor CONFIG port, see section 2.4.2 below.



RJ 45 plug: View on underside

EIA422/485 (5-wire)	
1	TxB
2	TxA
3	Signal common
4	Not used
5	Not used
6	Signal common
7	RxB
8	RxA
Plug shroud to cable screen	

Slave device
ioB, exp1/2

EIA485 (3-wire)	
1	EIA485 B
2	EIA485 A
3	Signal common
4	Not used
5	Not used
6	Signal common
7	Not used
8	Not used
Plug shroud to cable screen	

Master/slave device
ioB, exp1/2

EIA422/485 (5-wire)	
1	RxB
2	RxA
3	Signal common
4	Not used
5	Not used
6	Signal common
7	TxB
8	TxA
Plug shroud to cable screen	

Master device
ioB, exp1/2

ALIN	
1	Not used
2	Not used
3	Not used
4	ALIN A
5	ALIN B
6	Not used
7	Not used
8	Not used
Plug shroud to cable screen	

ioA, exp1

Profibus	
1	EIA485 B
2	EIA485 A
3	Signal common
4	Not used
5	Not used
6	+5V (for pull-up)
7	Not used
8	Not used
Plug shroud to cable screen	

ioA/B, exp1/2

Figure 2.4.1b Pinout for Connect module RJ45 type plugs

2.4.1 CONNECT MODULE (Cont.)

ALIN CONNECTORS

The Connect module contains two pairs of ALIN RJ45 type connectors called i/oA. The left-hand pair is for the left-hand processor; the right-hand pair for the right-hand processor. The two sockets making up each pair are connected in parallel to allow easy daisy-chaining.

Connection with an ALIN hub, or a PCI ArcNet card (also fitted with 8-way RJ45 connectors) can be made using an RJ45-to-RJ45 cable assembly available from the manufacturer under part number S9508-5/2RJ45. This cable has all eight connections made at both ends, thus making it suitable for all applications, not just ALIN which uses only a single twisted pair). Fig 2.4.1c is a schematic showing the connections.

Notes:
1 The Rx and Tx legends apply to Modbus master connectors. Slave connections have Tx and Rx reversed as shown in figure 2.4.1b above.
2 Wire colours shown may not be correct for your cableform

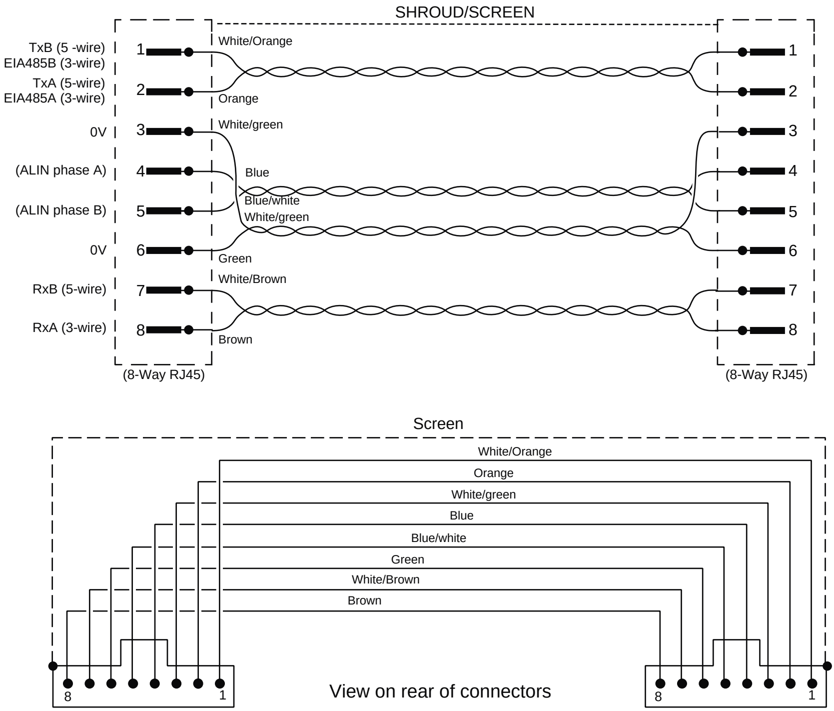


Figure 2.4.1c S9508-5/2RJ45 connection details

2.4.1 CONNECT MODULE (Cont.)

ALIN HUBS (ACTIVE)

Figure 2.4.1d shows a simple ALIN hub layout, and figure 2.4.1e a daisy-chain layout. The hub layout is preferred in cases where the integrity of the ALIN network is considered to be susceptible to lengths of daisy-chain becoming inoperative due to cable breakage or individual hardware faults.

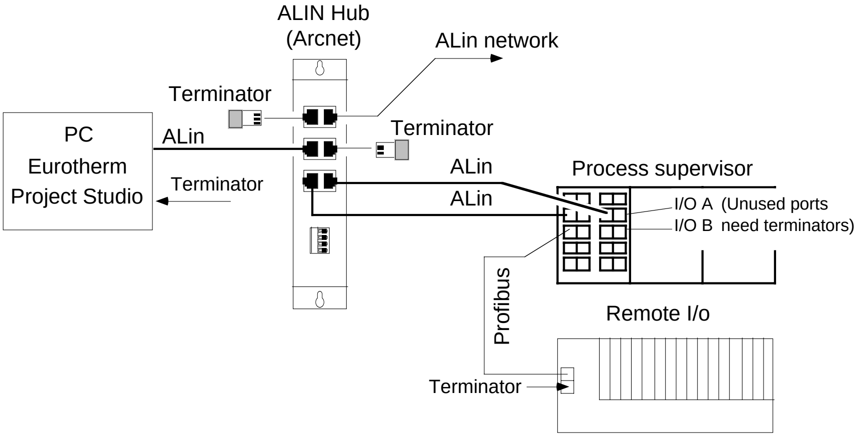


Figure 2.4.1d ALIN Hub layout

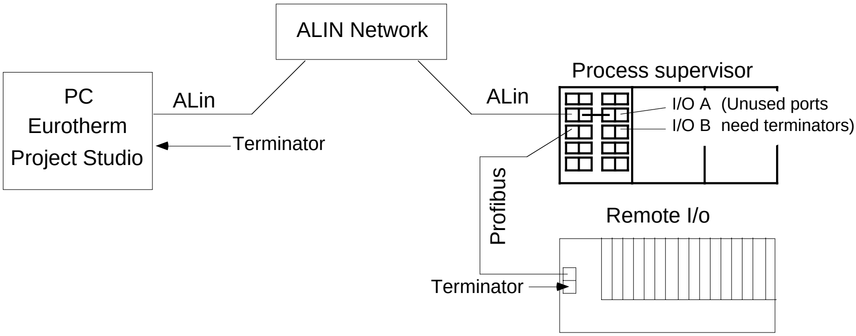


Figure 2.4.1e Daisy-chain layout

2.4.1 CONNECT MODULE (Cont.)

ALIN HUBS (PASSIVE)

Mechanically, a passive ALIN hub consists of a metal box with 12 RJ-45 type connectors and one RJ11 connector (for earlier equipment). Electronically, the hub consists of a resistor network designed to allow each of the 12 ports to be connected to a single unterminated- node, using a cable up to three metres in length. Cable termination is provided by each port, and vacant ports must be left unterminated. This system ensures survival with one port short circuited and any number (up to the maximum) of open-circuit ports.

DAISY-CHAIN LAYOUT

This method of connection is the preferred method where the integrity of the network is certain. Further details are to be found in the LIN/ALIN Installation and user guide HA082429U005.

CABLING

Shielded RJ45 connectors and screened Category 5 cables are widely available. Note, however, that specifications vary and not all components are suitable for reliable ALIN operation. In view of the problems that can arise with inadequate cabling, it is strongly recommended that ready-made interconnecting cables are ordered from the manufacturer.

Note: For instruments with hardware status levels up to and including level 6 it is recommended that Modbus terminators are not fitted, since if they are, operation of the associated serial link may be impaired. This restriction does not apply to instruments with status level 7 or higher.

2.4.1 CONNECT MODULE (Continued)

DC SUPPLY WIRING

Each processor has two 24V supply connections (A and B) near the bottom of the Connect module front panel. The unit will operate on any dc voltage between 18V and 36V at a maximum power requirement of 50W per processor module. In addition to this, a separate connector allows an external battery of between 2.4 and 5.0V to be connected to maintain the real-time clock and data stored in Static Random Access Memory (SRAM). Typical drain currents are 2mA at 2.4V and 3mA at 3.4V.

Recommended power supply units and batteries are listed in Chapter 10.

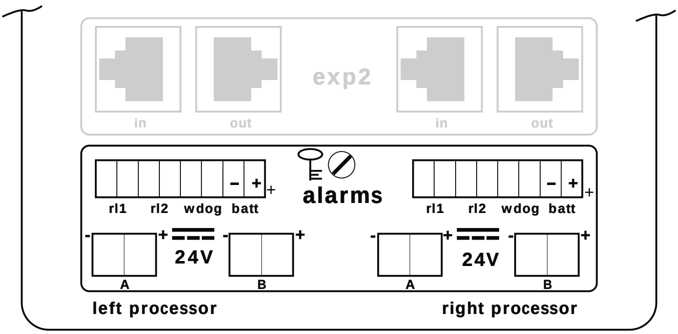
For units with hardware status levels 8 or above, a nickel/metal-hydride battery board can be supplied inside each processor unit. When fully charged this will maintain the SRAM and Real-time clock data for 72 hours, should an external battery not be available during power down, or if the Connect module is removed from the backplane. The battery is supplied partially charged, but it is recommended that the processor unit in which it is fitted be left powered continuously for 48 hours, to ensure full backup capability.

For units with status levels below level 8, the real-time clock is maintained by a .22F 'super cap'. After being charged from flat for 6 hours the capacitor will maintain the clock for12 hours (typ). When fully charged, the clock will be maintained for 24 hours (typ).

Figure 2.4.1f shows the locations of the connectors and gives recommended conductor sizes based on current carrying capability and connector capacity.

FUSES

All positive supply lines must incorporate a fuse. Suitable types are 3A Type T for 24 Volt supplies and 0.5V Type T for each external battery fitted.



Recommended wire sizes

DC supply (single wire): $0.2mm^2$ to $2.5mm^2$ (20 awg to 14 awg)
Relays /battery(single wire): $0.14mm^2$ to $1.5mm^2$ (25 awg to 16 awg)

Fuse types (fuses to be fitted in positive supply line)

24V supply = 3A Type T; External battery = 0.5A Type T

Figure 2.4.1f DC and relay connection details

2.4.1 CONNECT MODULE (Cont.)

RELAY WIRING

There are three relays associated with each processor module, and the common and normally open terminals of these relays are wired to the front of the Connect module as shown in figures 2.4.1f and 2.4.1g. The contact ratings (resistive loads) for the relays are 30V ac/60V dc at 0.5A.

The operational strategy of relays 1 and 2 (rl1 and rl2 respectively) is entirely software controlled and is set up during configuration.

The watchdog relay is under hardware control; this hardware making a number of health checks, before operating the relay. If during operation one of the health check fails, the relay goes into its alarm (power-off) state. See Chapter 3: User Interface for full details of the watchdog system.

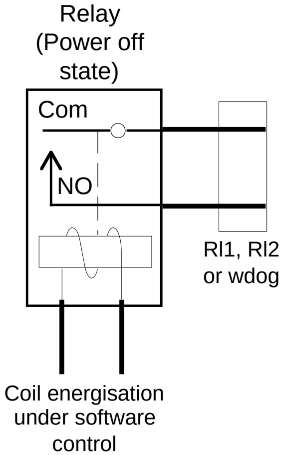


Figure 2.4.1g Relay Wiring

The relays can be wired in series or in parallel. When in parallel, both Processors have to fail, before the alarm becomes valid. When in series, the alarm becomes valid if either processor fails. Figure 2.4.1h shows the relays wired in series to a 24V dc 'healthy' lamp. Figure 2.4.1i shows a parallel configuration, using an auxiliary relay to display both healthy and warning states.

Note: For all relays, the common and normally open contacts are open circuit during power-off, and remain so for some seconds at power-up, until software control has become established. After that, the contacts are short circuit when the relay coil is energised, and open-circuit when the coil is not energised.

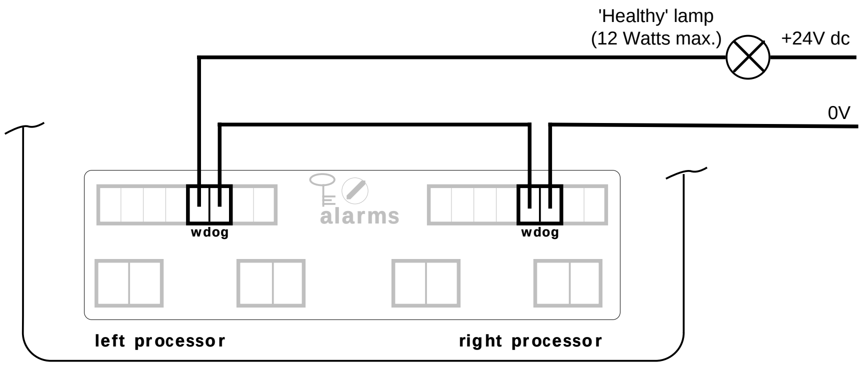


Figure 2.4.1h Example wiring for watchdog relays in series

2.4.1 CONNECT MODULE (Cont.)

RELAY WIRING (Cont.)

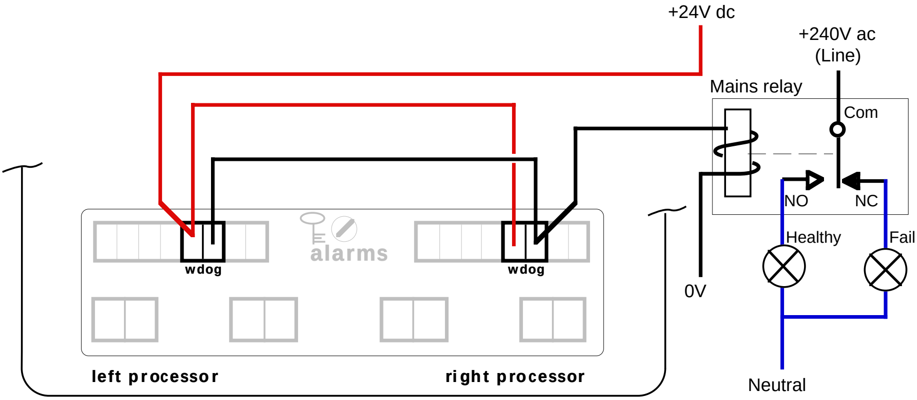


Figure 2.4.1i Example wiring for watchdog relays in parallel

2.4.2 Procssor module

The processor module contains one user connection, an RJ11-type plug for the connection of a configuration terminal for on-line monitoring and minor configuration editing. It is possible to configure a whole system from such a terminal, but is not recommended because of the complexity of most systems.

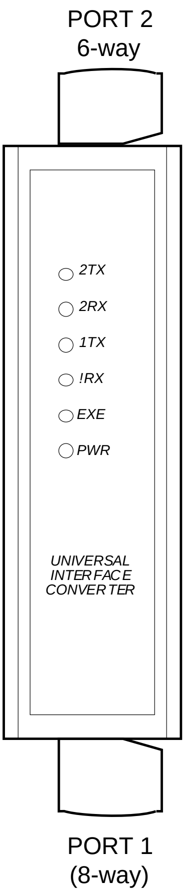
It is recommended that a Communications isolator be connected between the processor module and the terminal. Figure 2.4.2a gives a pinout for such a unit. Figure 2.4.2b shows cables for connecting the processor config port directly to a PC's EIA232 port, both 9-way and 25-way versions.

Communications parameters should be set up as listed below, from the Properties/Connect-to/Configure connection menu:

Parameter	Value
Baud Rate	9600
N° of data bits	7
N° of start bits	1
Parity	Even
N° of stop bits	1

Table 2.4.2 Communications parameters

2.4.2 PROCESSOR MODULE (Cont.)



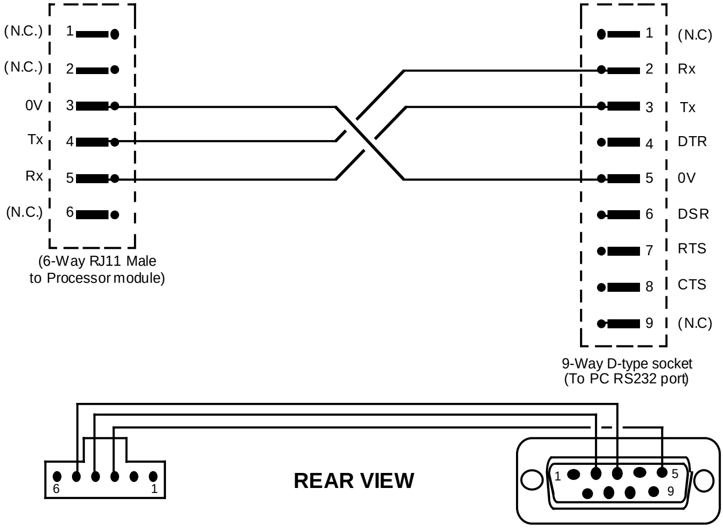
EIA232 Connections

Signal etc.	Communications isolator			Processor
	Port 1 (8-way)	Port 2 (6-way)	Port 1 RJ11	Config Port RJ11
RX Input	3	3	5	5
TX Output	1	1	4	4
Signal ground	6	6	3	3
RTS	5 (input)			
Supply +	7			
Supply +	8			

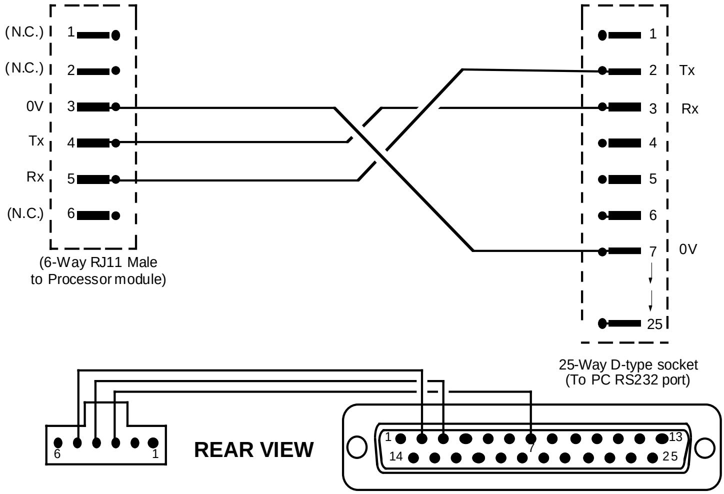
Notes: 1 For isolators with a DIP switch adjacent to Port 2, set all elements Off for RS232 Comms.
2 The RJ11 connector is in parallel with Port 1 (Signals only - not power)
3 For recommended isolators and suitable made-up cables, refer to the ordering guide in Chapter 10.
4 Supply power range is 7 to 35Vdc. Worst case inrush current = 660mA at 4V.

Figure 2.4.2a Isolator wiring details

2.4.2 PROCESSOR MODULE (Cont.)



Connection using 9-way D-type



Connection using 25 - way D-type

Figure 2.4.2b Direct connection between CONFIG port and PC

2.4.2 PROCESSOR MODULE (Cont.)

CONFIGURING CONTROL STRATEGIES & SEQUENCES

You can configure control strategies and sequences for the Process Supervisor using either an external PC-based graphical software package (Eurotherm Project Studio), or with the simpler inbuilt configurator and a dumb terminal.

EUROTHERM PROJECT STUDIO

Control strategies and sequences to be run in a Process Supervisor may be configured and downloaded using the Eurotherm Project Studio, which is fully described in the documentation supplied with it. Information is also available via the Eurotherm network explorer. The *LIN Product Manual* (Part no. HA082375U999) should be consulted for details of the function blocks that can run in the Process Supervisor.

TERMINAL CONFIGURATOR RESTRICTIONS

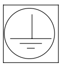
The use of the configurator is restricted according to the operating mode of the Process Supervisor in the following ways:

- The terminal configurator can be used only on the primary processor module.
- The database must not be running if you want the full capability to create blocks, databases, edit field values, and modify pool data (e.g. engineering units). If it is running, the configurator can write only to the normally runtime-writeable fields: e.g. block names cannot be edited, but new blocks may be added and new ‘wires’ can be made on-line.

These restrictions prevent files or edits occurring in the primary database that cannot be tracked by the secondary database.

Note: When the database is started after the terminal configurator has been used, an automatic database save is performed. This ensures that any changes are notified to the secondary CPU during synchronisation.

2.4.3 Safety earth connection



As shown in figure 2.3.1, an M4 earth stud connection is provided on the back plane metalwork. This stud should be bonded to a good local earth using multistrand tri-rated 1.5mm^2 (21A) green/yellow earth cable, with ring terminals for security.

This page is deliberately left blank

Chapter 3 USER INTERFACE

3.1 INTRODUCTION

This chapter describes the functions of the processor module LEDs and switches.

As shown in figure 3.1, the items are arranged in groups on the processor module front panel, and each group is described in turn below. Table 3.1 (below) is a concise list of the LEDs and their functions.

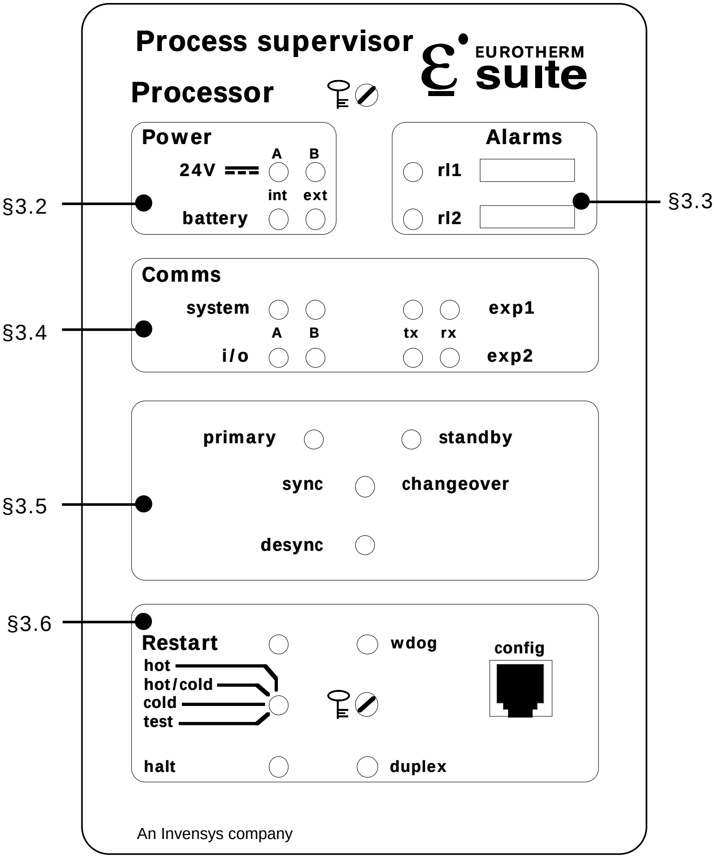


Figure 3-1 Processor module front panel

Group	LED	Function	Diagnostic Value (Ch. 6)
Power	Power A	Green..................... Main power input valid Off.......................... Main power input failed	
Power	Power B	Green..................... Auxiliary power input valid Off.......................... Auixiliary power input failed	
Power	backup ext	Green..................... External battery power valid Off.......................... External battery power failed (Off until start-up complete)	
Power	backup int	Green..................... Internal battery power valid Off.......................... Internal battery power failed (Off until start-up complete)	
Alarms	rl1	Yellow.................... Alarm active Off.......................... Alarm not active	08
Alarms	rl2	Yellow.................... Alarm active Off.......................... Alarm not active	04
Comms	System A	Green..................... System A communications valid Red........................ System A communications hardware failure Flashing Red/Off..... System A communications cable fault	
Comms	System B	Green..................... System B communications valid Red........................ System B communications hardware failure Flashing Red/Off..... System B communications cable fault	
Comms	I/O A	Green..................... System A communications valid Red........................ System A communications hardware failure Flashing Red/Off..... System A communications cable fault	
Comms	I/O B	Green..................... System B communications valid Flashing Green/off.. Remote unit fault (Profibus comms. only) Red........................ System B communications hardware failure Flashing Red/Off..... System B communications cable fault	
Comms	Exp1 Tx / Rx	Intermittent yellow....Communications taking place	Rx = 20 Tx = 10
Comms	Exp2 Tx / Rx	Intermittent yellow... Communications taking place	Rx = 80 Tx = 40
Startup	Primary	Green..................... This CPU is primary Off.......................... This CPU is not primary Flashing................. Powered up but no database is running	02
Startup	Standby	Yellow.................... This CPU is secondary and synchronised Off.......................... This CPU is not secondary synchronised Flashing................. Synchronisation process in progress.	01
Startup	wdog	Green............................... CPU not in reset Red.................................. CPU in reset Red/Green alternating....... Power up sequence in progress	
Startup	Duplex	Green............................... Redundancy communications valid Off.................................... System in non-redundant mode Red/Green alternating....... Inter CPU communications failed	

Table 3.1 LED functions

3.2 Power Monitoring LEDs

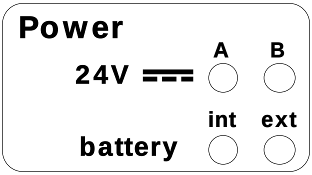


Figure 3.2 Power monitoring LED group

This group of LEDs, located near the top left of the processor module front panel, shows the status of the power inputs wired to the Connect module, and of the internal battery.

3.2.1 A and B

For each processor, two independent sources of 24V power (A and B) can be wired to the Connect module. The LEDs labelled A and B, illuminate green if power supply inputs A and B respectively are greater than 14V.

All sources must be fused (3 A type T) in the positive supply line.

3.2.2 ext

Each processor unit can be backed-up by an external battery wired to the 8-way terminal blocks on the Connect module. This maintains the data in the real-time clock and in the Static Random Access Memory (SRAM), which holds the system's volatile data, for a period that depends on the Ampere-hour (Ah) rating of the battery. Typical load currents are 2mA at battery voltage of 2.4V and 3mA at 3.4V. Once the start-up sequence is complete, the 'ext' LED illuminates continuously green if the battery voltage is greater than approximately 2.6V, and 24V supply power is available. When 24V power is not available the LED is not illuminated.

External battery supplies must be fused (0.5 A type T) in the positive supply line.

3.2.3 int

A further, short-term (72 hours minimum), backup for the SRAM and real-time clock can be provided by an optional internal battery. The 'int' LED operates for this battery in the same way as described above for 'ext', except in that the internal battery voltage must be greater than approximately 3.8V for the LED to be illuminated.
In order to achieve the minimum time duration specified above, 24V power must have been applied to the processor module for 48 hours to ensure that the battery has been fully charged.

Note: Recommended power supply and battery units are listed in chapter 10 of this manual.

3.3 ALARM LEDS

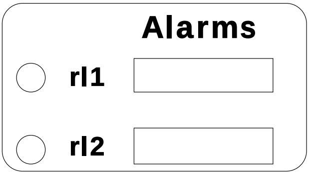


Figure 3.3 Alarm Relay LEDs

This pair of LEDs is located near the top right of the processor module front panel, and indicates the status of the relay outputs ‘rl1’ and ‘rl2’ available at the 8-way terminal blocks on the Connect module. Each LED illuminates yellow if its associated relay is in alarm state (coil not energised). This happens both in alarm and during start-up.

3.4 Comms LEDs

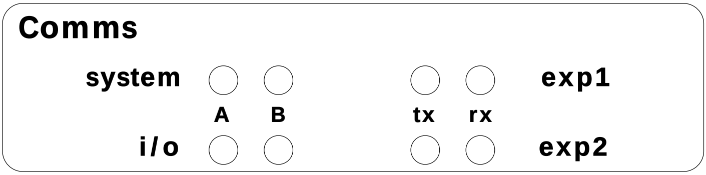


Figure 3.4 Communications LEDs

This group of eight LEDs is located just above the centre of the control panel.

3.4.1 System A/B

This pair of LEDs indicates that system communications are working correctly (illuminated green) or not working correctly (red). If illuminated continuously red, a hardware (electronics) fault is present. If flashing Red/Off, a fault has been detected in the relevant cable or associated connectors.

3.4.2 i/oA and i/oB

GENERAL

This pair of LEDs indicates that the I/O communications systems are working correctly (illuminated green) or not working correctly (red). If illuminated continuously red, a hardware (electronics) fault is present. If flashing Red/Off, a fault has been detected in the relevant cable or associated connectors.

PROFIBUS (I/OB)

When running a Profibus link, the i/oB LED indicates the following:

Off	Profibus not running. (Database not running, no profibus port configured, configuration error preventing Profibus start-up). Error weight 0*.	
Red	Steady:	Local hardware failure - e.g. Profibus board failure. Error weight 1*.
	Flashing:	Local access failure - e.g. cable not plugged in at one or both ends. Error weight 1*.
Green	Steady:	Profibus is working successfully. For a redundant pair secondary, this means only that the secondary can communicate with the primary. Error weight 2*.
	Flashing:	Remote access failure. One or more remote units have failed. This state cannot occur in a redundant pair secondary unit. Error weight 2*.

* Note: Error weights are used in redundant systems to determine what action to take in case of error - see section 8.10.

3.4.3 Exp1 tx/rx

This pair of LEDs indicates communications activity at the ‘exp1’ (expansion 1) port of the Connect module. When working correctly, the LEDs flicker yellow at varying rates according to processor receive (rx) and transmit (tx) activity.

3.4.4 Exp2 tx/rx

This pair of LEDs indicates communications activity at the ‘exp2’ (expansion 2) port of the Connect module. When working correctly, the LEDs flicker yellow at varying rates according to processor receive (rx) and transmit (tx) activity.

3.5 CHANGEOVER LEDs AND SWITCHES

The sync and desync switches are set behind the panel, and should be operated, when necessary, by a blunt, plastic tool such as the recessed end of a trim-pot adjuster.

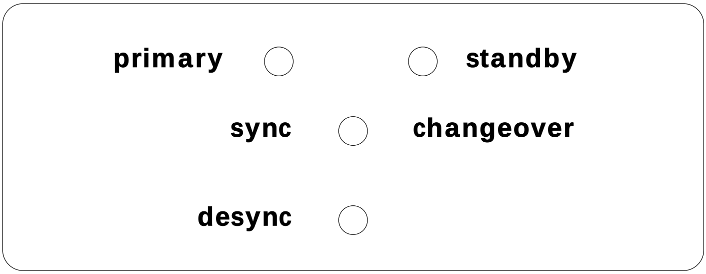


Figure 3.5 Changeover LEDs and switches

This group of components is located slightly below the centre of the processor front panel, and is used to monitor and control the redundant / non-redundant mode selection. The group consists of two LEDs ‘Primary’ and ‘Standby’ and two membrane switches ‘sync’ and ‘desync’. Section 3.5.5, below, gives a brief description of synchronisation.

3.5.1 Primary LED

This LED is illuminated green if this processor module is currently the primary processor. During start up, this LED flashes on and off, until a database has been loaded and is running successfully.

The LED is off, if this processor is not the primary.

3.5.2 Standby LED

This LED is illuminated yellow continuously if this processor module is currently the secondary module of a synchronised redundant system, and is thus able to take over from the primary if required

If this processor is the secondary, the LED will flash whilst the processors are synchronising. This normally happens during start up, but can be forced by operation of the primary ‘sync’ push switch.

3.5.3 Sync/changeover switch

Operation of the primary processor’s ‘sync/changeover’ switch causes the secondary module to start synchronising with the primary module. The secondary’s ‘Standby’ LED flashes during this synchronising process. Once synchronisation is complete, operation of the secondary processor’s ‘sync’ switch causes primary/secondary changeover.

3.5.4 Desync switch

Operation of the desync push-switch causes synchronised processors to de-synchronise.

3.5.5 Processor module Synchronisation

Applicable only to redundant systems, synchronisation means the bulk transfer of all relevant data from that processor which is designated the primary processor to that which is designated the secondary, followed by continuous maintenance of this copied data. This allows the secondary processor to take over from the primary should the primary fail.

This synchronisation process takes place automatically, if both processor modules are powered-up together, and have previously been run as a redundant synchronous pair. Should either of the above conditions not be met, then, at power-up the primary and secondary processors adopt unsynchronised states (Primary unsynch and Secondary unsynch). In such a case, the secondary module cannot take over from the primary in the event of failure.

To synchronise the processors, the primary's 'sync' push switch must be operated.

Once synchronisation has been achieved, the processors are said to be in primary synch state and secondary synch state, and the secondary is able to take over from the primary if required.

Note: With some peripherals, comms failures may be reported during the synchronising process.

TIME TO SYNCHRONISE

The time taken to complete the synchronisation process varies according to the complexity of the control strategy and on how heavily the Flash file system is used. Typically, the 'Load and Run' part of the procedure takes a number of seconds, and the file transfer can take some minutes. During this period, the primary processor runs the control process as normal.

3.6 Startup LEDs and switches

The Restart and halt switches are set behind the panel, and should be operated, when necessary, by a blunt, plastic tool such as the recessed end of a trim-pot adjuster.

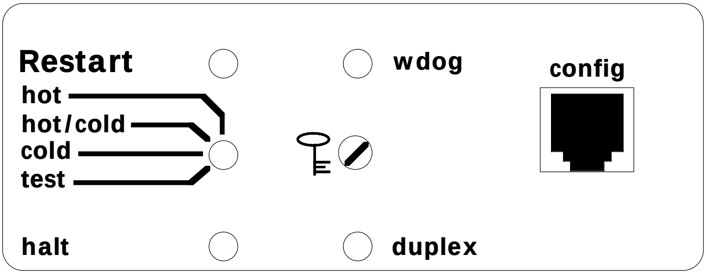


Figure 3.6 Startup control and monitoring

This group of components is located near the bottom of the processor module front panel and consists of two LEDs, two push switches and a four-position rotary switch.

3.6.1 wdog LED

This LED gives information about how start-up is progressing (see chapter 4 for details) and, about what faults might have occurred to cause a processor module to stop running the data base.

If the LED is flashing red/green, the module is in start-up mode.

If the LED is illuminated continuously green, the processor module is running normally as far as the software is concerned, and there are no detectable hardware errors.

If the LED is illuminated continuously red, the processor module is in reset, as a result of one or more of the following errors having been detected:

1. One of the cooling fans has failed during start up. (If a fan fails at any other time, an alarm is set in the header block)
2. The processor circuits have overheated.
3. The processor clock is not running.
4. Halt switch has been operated.
5. A software fault has forced a ‘Halt’.

As shown in Chapter 2, (figure 2.3.5b) sliding segment 1 of SW2 to the left will cause the processor repeatedly to try to re-start after a watchdog failure. Sliding the segment to the right disables this re-try facility, and the processor has to be re-started by the user.

3.6.2 duplex LED

This LED is illuminated green, if the inter-processor communications are valid, and successful data transfers are taking place between the two processor modules. Applies only to redundant systems.

The LED flashes red/green if the interprocessor communications have failed.

The LED is off if the system is not running in redundant mode.

3.6.3 RESTART switch

Operation of this push switch causes the relevant processor to restart in the mode selected on the rotary mode switch immediately below it.

Used after a watchdog failure has occurred.

3.6.4 Halt switch

Operation of this switch for more than four seconds causes a watchdog failure to stop the processor. In a redundant system, ‘Halt’ on the primary processor causes the secondary to take over. This function is normally used only during commissioning or servicing.

3.6.5 Start up mode

This is selected at an eight-position rotary switch, with its positions labelled: Hot, Hot/cold, Cold and Test. (180˚ switch actuator positions are wired identically such that position 1 = position 5, position 2 = position 6 and so on.) A full discussion of start-up modes is given in chapter 4, but a simple overview is included here to complete the user interface description. Figure 3.6.5, below, is a simplified flow diagram showing the basic elements of the start-up sequence.

HOT

A time-out period can be configured by the user for hot start, and this period varies from application to application. The time out period is defined as ‘that period (after the database has stopped running) within which the data base can be re-started without noticeably degrading or endangering the control process. If a restart is requested within this time, and the database is still valid, the processor will restart, using the last known data base. If the data is not valid, or if the timeout has been exceeded, the processor will not attempt to restart, but will clear the memory and create an empty data base. This is called the ‘idle’ state.

COLD

If cold start is requested, the processor will attempt to start from the default data base. If this is not possible, the processor will enter ‘idle’ mode.

HOT/COLD

With the switch set to this position, if a hot start is not possible, a cold start will be attempted.

TEST

This is normally used only during commissioning or servicing, for example under the following conditions:

1. First-time start up.
2. Start-up after a new version of system software has been installed.
3. Memory configuration has been changed.
4. If the processor is to start-up, but is not yet to run a data base.

The memory is cleared, and a blank data base is created.

Note: If the 8 MByte DRAM option is fitted, then, for software versions 1.6 onwards, optimum operating performance of the database is achieved with ‘Cold’ or ‘Test’ selected as start-up mode.

3.6.5 START UP MODE (Cont.)

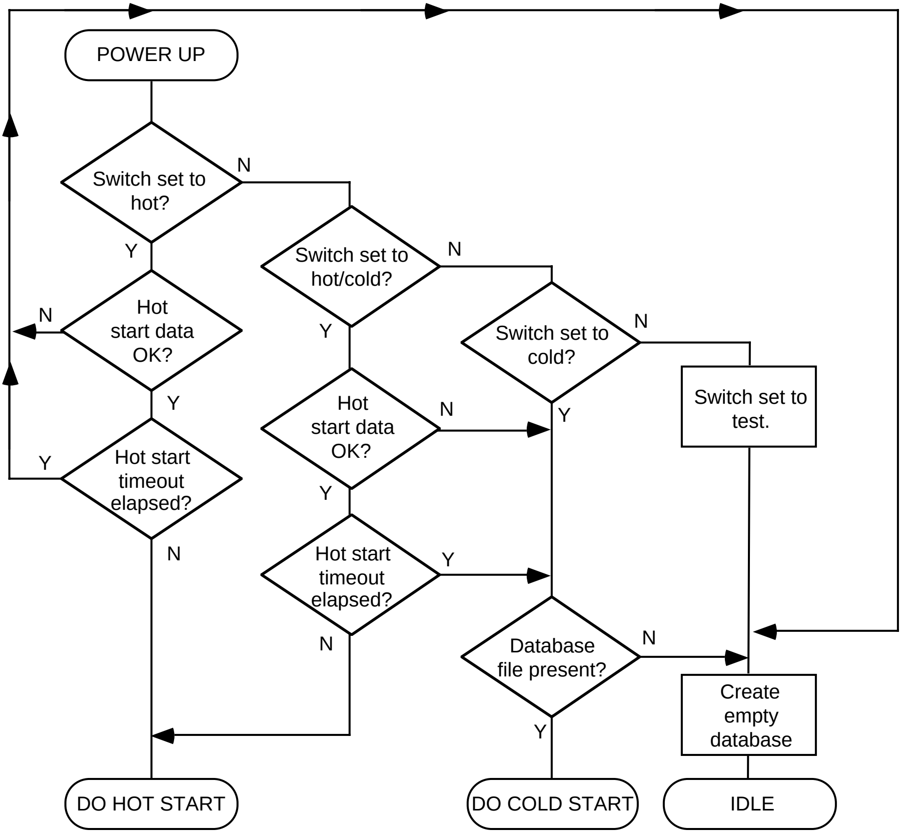


Figure 3.6.5 Simplified start-up flow diagram

Chapter 4 START-UP

This chapter describes the start-up sequence for the unit. Topics covered include the differences between redundant and non-redundant systems and start-mode (hot/cold etc.).

4.1 REDUNDANCY MODES

Redundant (duplex) mode is where two processor units are fitted and are required to act in such a way that one can take over from the other in case of failure. In such a case, one processor unit (normally the left-hand one) is called the 'primary' and the other the 'secondary'. The secondary tracks the primary so that it can take over with minimum disturbance to the controlled system.

Non-Redundant (simplex) mode is where a) there is only one processor or b) there are two processors fitted which act independently of one-another (either intentionally, or because one has failed).

Redundant/non-redundant mode is selected using the options switch (SW2) on the back plane, as shown in Chapter two of this manual (section 2.3.5).

4.2 START-UP MODES

The required start-up mode is selected using the eight-way rotary switch located near the bottom left of the processor front panel. This allows 'Hot', 'Hot/cold', 'Cold' or 'Test' to be selected. (Each start mode has two positions on the switch, 180˚ apart.) Figure 4.2.1, below, shows a simplified flow diagram for the different modes.

4.2.1 Hot start

Hot start means that the instrument re-starts from where it left off (i.e. with the current database still loaded, and with all the variables for the application still at the values they held when database processing stopped). A suitable time period (Cold Start Time) is configured in the root block of the control data base, and if this period is exceeded after the data base stops running, then a hot start is not permissible. The Cold Start Time for any process can be defined as: That period, after the database stops running, within which the database can be restarted from where it left off without unacceptably degrading or endangering the control process.

A brownout time can be set in the root block, and if power to the unit is lost for this duration or longer, the brownout alarm will be set (also in the root block). This brownout time can be defined as the maximum duration of power loss that can be tolerated by the process without reinitialisation being necessary.

For this unit, if the Hot start fails (because the data base is corrupted or because the Cold Start Time has been exceeded) the data base will be cleared and the processors will enter an 'Idle' state and remain there until physically restarted. (See also Hot/Cold start.)

4.2.2 Cold start

Cold start means that the instrument re-starts with the previous database loaded, but with all parameters and values set to starting values appropriate to the process (that is, re-initialised). If the cold start fails the data base will be cleared and the processors enter an 'Idle' state and remain there until physically restarted.

4.2.3 Hot/cold start

This setting causes the unit to attempt a hot start. If the hot start fails, however, instead of going straight into idle state as with 'hot start', the unit attempts to carry out a cold start. If the cold start fails the data base will be cleared and the processors enter an 'Idle' state and remain there until physically restarted.

4.2.4 Test start

A Test start is where the instrument starts up with that part of the memory which holds the database (Static RAM, or SRAM) cleared (set to zero throughout).

A Test start is normally requested:

1 at the very first start up in the life of an instrument
2 when no automatic re-start is required
3 when the start-up preconditions for a redundant system are to be modified
4 when a new version of software has been loaded
5 when there have been modifications in the instrument hardware.
6 in order to clear redundancy start-up data from memory

4.2 START-UP MODES (Cont.)

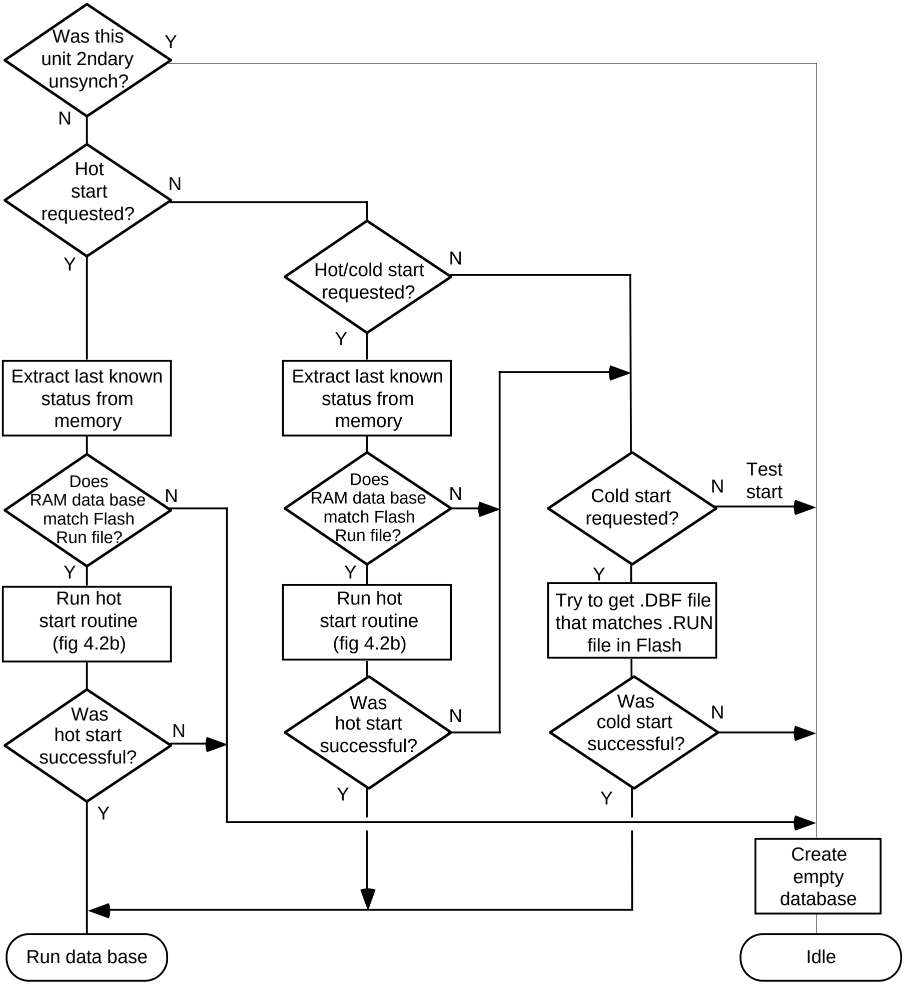


Figure 4.2.1 Simplified start-up flow diagram

4.2 START-UP MODES (Cont.)

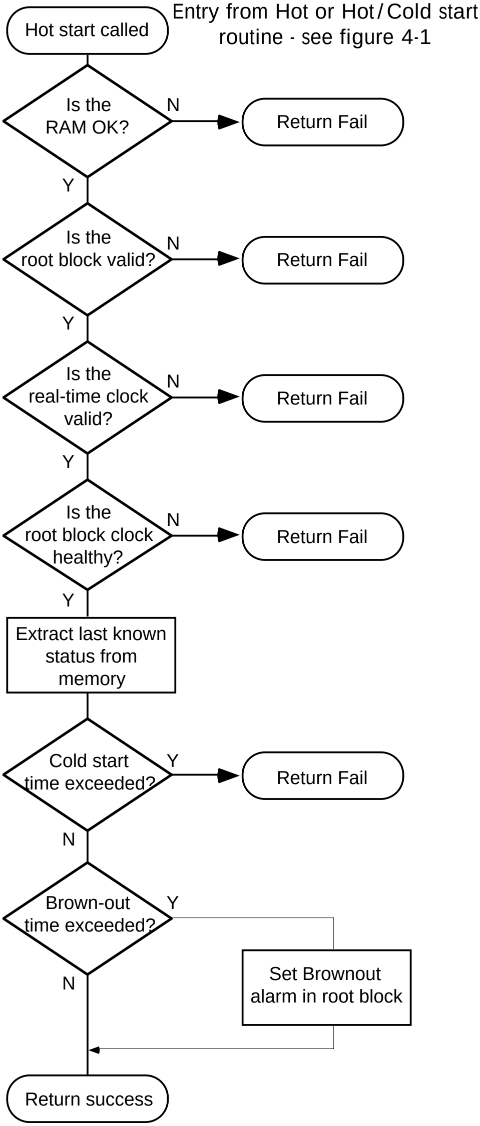


Figure 4-2 Hot-start flowchart

4.3 STARTING A SINGLE (NON REDUNDANT) PROCESSOR

4.3.1 Start-up sequence

Refer to figure 4.3.1 for locations of the various LEDs.

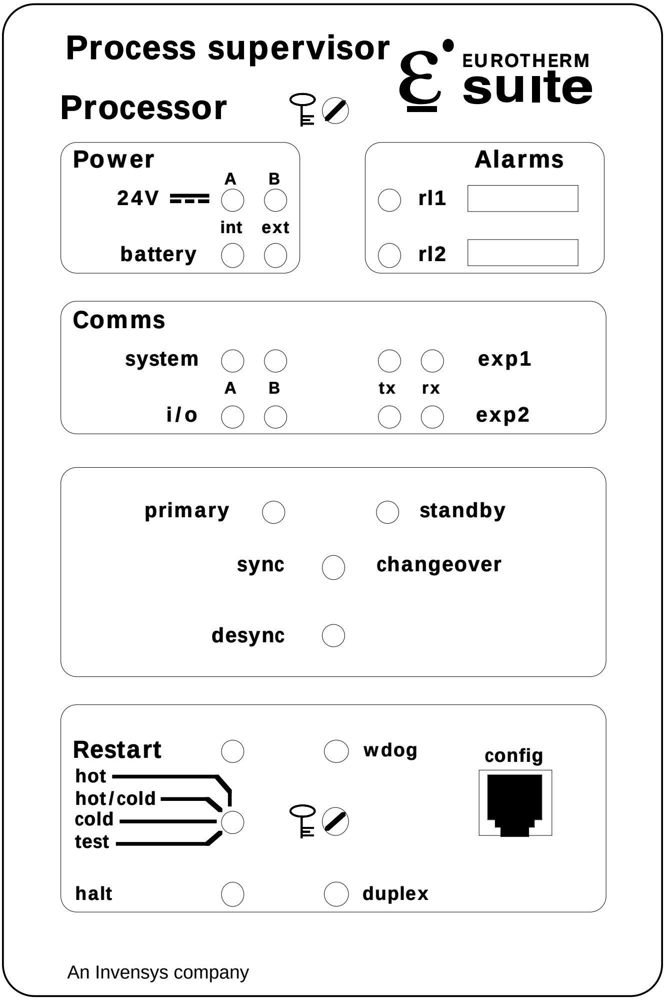


Figure 4.3.1 Processor module front panel

4.3.1 START UP SEQUENCE (Cont.)

OFF STATE

In the Off state, all LEDs are extinguished.

STARTING STATE

When power is applied, the relevant Power LED(s) illuminate green immediately.

The Basic I/O System diagnostic LEDs (rl1, rl2, exp1 rx, exp1 tx, exp2 rx, exp2 tx, Primary and Standby) flash intermittently until the processor is initialised, at which point they all switch off. (See chapter 6 for more details of these LEDs).

‘Wdog’ flashes green/red until the sequence is complete and the CPU has started running the applications software, after which it is illuminated continuously green. See section 4.3.2 below, for further details.

The start-up procedure concludes with the processor attempting to establish Arcnet (ALIN) comms. During this period, the Primary LED flashes on and off.

PRIMARY UNSYNCH STATE

When the start-up sequence is complete, then as a minimum, the Power and wdog LEDs are illuminated continuously green. The primary LED will be illuminated continuously if a data base is running, or it will flash if a data base is not running

The COMMS system LEDs will also be illuminated green if the associated comms links are operating correctly, or Red (steady or flashing) if not.

In addition, if any other communications are in progress, the relevant LEDs will be illuminated, either continuously or intermittently. See Section 3.4 for more details of the communications LEDs.

If back-up batteries are fitted, the ‘int’ and ‘ext’ LEDs are illuminated as appropriate.

4.3.2 Watchdog indications

The watchdog LED has four modes of operation:

<u>Steady green</u> In this state, either the processor is running with no detectable hardware or software faults, the cooling fans are working and the processor temperature is within its working range, or one of the Monitors has been accessed - see section 4.6.

<u>Steady Red</u> When continuously red, a hardware or software fault has developed - see section 4.5, below

<u>Long red/short green flash</u>. This occurs at the beginning of the start-up procedure, whilst the status of the fans and the temperature of the central processing unit are checked.

<u>Long green/short red flash</u>. This indicates that the fan status and temperature measurement were acceptable, and initialisation is continuing correctly. This mode remains active until the start-up process is complete, after which the LED stops flashing and is illuminated continuously green.

4.3.3 Watchdog relay

For primary or simplex units, the watchdog relay is in its alarm state until the Primary LED is continuously illuminated. For secondary units, the watchdog relay comes out of alarm when the database is started, part way through the synchronisation sequence.

4.4 STARTING UP A PAIR OF PROCESSORS

4.4.1 Redundant mode

This start-up sequence is similar to that described for a single processor (section 4.3 above) except in the control and action of the Standby and Duplex LEDs. Figure 4.3.1 (above) shows the locations of the various LEDs.

POWERING UP DECISIONS

Figure 4.4.1 shows the states possible with a pair of processor units in redundant mode.

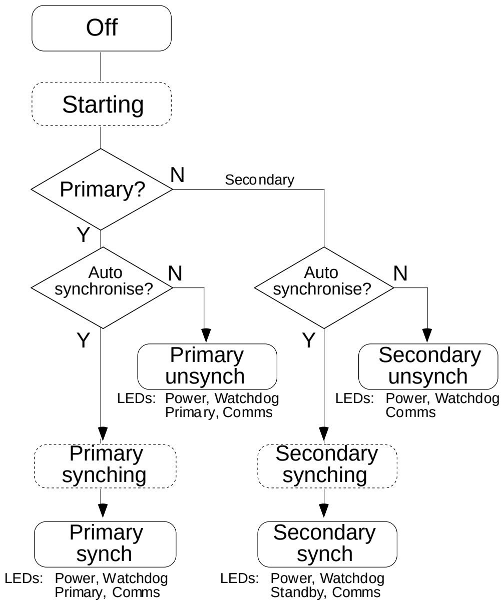


Figure 4.4.1 Power-up redundancy states for a pair of processor units in redundant mode

4.4.1 REDUNDANT MODE (Cont.)

PRIMARY/SECONDARY CRITERIA

With processor units in redundant mode, it is necessary that one be defined as the primary; the other as the secondary. As described above in section 4.1 of this chapter, the primary unit initially assumes control and the secondary tracks the primary such that it can assume control should the primary unit fail. Which of the processors powers-up as the primary is determined as follows:

1 If both processors are powered-up ‘simultaneously’ from their as-delivered default states, the left hand PROCESSOR MODULE (as viewed from the front) attempts to assume primary status.

2 If both processors are powered-up ‘simultaneously’ from other than default state, further tests must be made on the basis of ‘last time’s’ information held in battery-backed memory. This information contains data relating to whether this processor was primary or secondary prior to the last power off. If both processors were primary or both secondary last time, or if the data is inconclusive, then the left hand processor will attempt to assume primary status this time, otherwise they will power-up according to last-time’s status.

3 If the processors are powered-up sequentially, then the first-powered will attempt to assume primary status.

AUTOSYNCHRONISATION

Once the primary/secondary status of the processors has been determined, the system must decide whether synchronisation of the primary and secondary should be automatic or whether it should proceed only after a request from the operator (sync switch). This decision is made as follows:

If the processors are powered-up within 1 second of one another, AND they were running as a synchronised pair prior to power-down (data held in battery-backed memory), then synchronisation will take place without operator intervention.

If either of the above conditions is not met (or if the battery-backed data is not available) then both units will enter unsynchronised states in which case the secondary cannot take over from the primary. This state will continue until the ‘sync’ switch on the primary processor is operated.

SYNCHRONISATION

During synchronisation (automatic or manual), the primary processor carries out the following:

1. The transfer of any cold or hot start data base files to the secondary.
2. It instructs the secondary to load the relevant database.
3. Once this is complete, transfers current block data to the secondary.

4.4.1 REDUNDANT MODE (Cont.)

SYNCHRONISATION (Cont.)

During the synchronisation process, the 'Standby' LED on the secondary processor front panel flashes. Once synchronisation is complete, the Standby LED is continuously illuminated yellow, the 'Duplex' LED is illuminated green and redundant operation starts, with the processors in their synchronised states. In these states, the secondary continuously tracks the primary by receiving data from it including attachments, input reads, block execute synchronisation commands, check sums, block data and health data.

Notes:
1. During synchronisation, some peripherals may report a comms failure. Such failures are transient and clear within approximately two seconds.
2. In redundant operating mode, the secondary refuses any ALIN messages other than identity requests. All database related comms and file system comms are handled by the primary processor.

TIME TO SYNCHRONISE

The time taken to complete the synchronisation process varies according to the complexity of the control strategy and on how heavily the Flash file system is used. Typically, the 'Load and Run' part of the procedure takes a number of seconds, and the file transfer can take some minutes.

Where primary and secondary databases have substantial differences (e.g. when attempting synchronisation for the first time), multiple syncs may be required to copy all the files to the secondary. When such is the case, it can be detected from the Red_Ctrl block sync fields.

4.4.2 Non-redundant mode

Starting a pair of processors in non-redundant (simplex) mode is the same as starting a single processor. Whether the units power up in redundant or non-redundant mode depends on the setting of the SRD element of the Options switch (SW2) on the back plane - see Section 2.3.5 in Chapter 2 of this manual.

4.5 LED FAULT INDICATIONS

The Alarm, Comms and Primary and Standby LEDs are illuminated in various patterns during the first part of the start-up sequence. Should the sequence stop at this time, the pattern of these LEDs give diagnostic information as follows:

POWER A/B LEDS

If either of the Power LEDs fails to light green on power-up there is a fault in the relevant power supply or the Connect module has been removed. If the Connect module is correctly fitted, isolate the power supply unit and remedy the fault.

WATCHDOG LED

If the Watchdog LED changes from short green/long red to steady red, the hardware has failed the temperature and fan tests. Switch off and remedy the fault.

If the Watchdog LED changes from long green/short red to steady red, one or more components of the software have not loaded properly. Switch off, switch on, and if still unsuccessful, contact a service engineer.

If the Watchdog LED changes from steady green to red, an operational fault has developed.

PRIMARY LED

If this LED is off, power to the processor module is off or the processor module is not the primary. If the LED is flashing green/off, there is no database running, either because the unit is still starting up, or because a database has not been loaded or has failed to start.

COMMS LEDS

The comms protocol (e.g. Modbus, ALIN etc.) associated with any one comms connector is configurable, and it is therefore not possible to be more specific about failure indication than the following:

SYSTEM AND I/O LEDS.

If a system or I/O comms LED does not light green, the processor module has not yet established communications. If the LED is illuminated continuously red, there is a hardware fault. If the LED is flashing between red/off, there is a cable or connector fault. See also section 3.4.

EXP1, EXP2 LEDS.

These LEDs remain off until the processor module has established communications, at which time the LEDs flicker, thereby indicating communications activity.

4.5 LED FAULT INDICATIONS (Cont.)

DUPLEX LED

If this LED is illuminated green, inter-processor communications have been successfully established and are running.

If the LED is flashing green/red communications have been established, but are not running (usually after a de-synch. request).

If the LED is off, no interprocessor communications have been established, usually because the system is non-redundant.

4.6 START-UP WITH A CONFIG TERMINAL

The configuration port towards the bottom right of the front panel can be used to monitor the start-up sequence, display fault messages etc. The start-up sequence is unchanged from that described in sections 4.3 and 4.4 of this chapter, above, with the following exceptions:

SYSTEM MONITOR

The System monitor is intended as a diagnostic tool for commissioning and or service engineers, and it is not recommended that it be accessed by other personnel.

The operating boot-up program includes a diagnostic facility called the System Monitor (‘S Mon’). During the first stages of start-up, a message “Press ‘s’ key to stop auto-boot” appears for about 1 second. If this message is ignored, the Start-up sequence will continue as normal; If the ‘s’ monitor is accessed, it allows the user to perform hardware diagnostics such as exercising the I/O ports, testing the status of the flash memory etc. from the terminal. Whilst S Mon is in use, the watchdog LED is illuminated continuously green, but the relay remains in its alarm state.

To quit the system monitor, the unit can be powered off, and then on again to restart the power-up sequence, or ‘Quit’ can be used followed by operation of the ‘Restart’ switch once the watchdog LED goes red.

M MONITOR

The M Monitor is intended as a diagnostic tool for commissioning and or service engineers, and it is not recommended that it be accessed by other personnel.

If the System monitor is ignored, then some time later, a similar message appears asking if the M Monitor is to be accessed. If the message is ignored, the Start-up sequence continues as normal; If it is accessed (within 1 second), it allows the user to configure communications ports, exercise the LEDs re-format the user disk etc. from the terminal. It is also possible to initiate a full Power-on Self-Test (POST) from the M Monitor. This will corrupt the current memory contents, thus making it unlikely that a hot start can be initiated. Whilst the M Monitor is in use, the watchdog LED is illuminated continuously green, but the relay remains in its alarm state.

To exit from the M Monitor ‘Quit’ is used and the system continues with Start up.

Note: for more details, refer to Chapter 9 (Service)

4.7 START-UP WITH SERVER STALL

The use of the Server Stall option of the config block slows the data base start and synchronisation, because it forces a complete read of all DCM block read fields for the lines on which it is applied. This takes, typically, 12 seconds per I/O controller slave node.

For details of the Server Stall option, please refer to the T940 Configuration block section of Chapter 10 of the Linblocks Reference Manual (HA082375U003)

Chapter 5 CONFIGURATION

The main topics of this chapter are:

- Tools: The Configurator and LINtools
- What Configuring consists of
- Preparing to run the Configurator
- Connecting to a PC
- Setting the control efficiency for the instrument
- Running the Configurator
- Configuring databases
- Configuring Communications (Modbus only).

5.1 TOOLS: THE CONFIGURATOR AND LINTOOLS

Most configuration will be done before despatch, using the LINtools component of the Eurotherm Project Suite. This chapter explains how databases and communications parameters are configured for the unit using the Configurator program resident within the processor unit(s). (In redundant mode the program is resident only on the Primary processor).

The Configurator program is mainly for adjusting configurations on site, usually to accompany modifications to the processing plant. The T500/550 LINtools Product Manual (HA082377U999) should be referred to for details of the configuration procedure using the LINtools package.

The Configurator employs the standard LIN block-structured approach. The LIN Product Manual gives full details of the software function blocks available for strategies, and how to configure their parameters.

The Configurator can also be used to load, start, stop, save and monitor databases, and to perform various filing operations.

5.2 CONFIGURABLE ITEMS

The configurable items depend on whether the database or Modbus communications is to be configured. In either case, it is a menu/item selection procedure. The LINtools package, by comparison, offers an icon-based 'drawing' capability.

Configuration of the database consists of carrying out one or more of the following:

1. Installing function blocks in the control strategy (MAKE)
2. Creating duplicates of existing blocks (COPY)
3. Deleting blocks (DELETE)
4. Inspecting and updating blocks (INSPECT)
5. Assigning LIN names and node addresses to external databases* (NETWORK)
6. Accessing the Utilities menu (UTILITIES), from which you can START and STOP programs, SAVE and LOAD databases and FILE pages and APPLY or UNDO changes.

*Note: External databases (EDBs) are databases running in other LIN instruments.

Configuration of Serial Comms consists of carrying out one or more of the following

1. Setting the operating mode of the instrument to either Master or Slave (MODE)**.
2. Selecting the characteristics for serial communications (SETUP). The characteristics are Baud rate, Parity, Stop bits, and Time out
3. Accessing the Tables list, which sets register mapping, and allows tables to be viewed (TABLE)
4. Accessing the Utilities menu (UTILITIES), from which you can SAVE or LOAD protocol configurations.

**Note: Master mode is not supported in software version 1.0

5.2.1 Configuration Access

The Configurator is accessed by connecting the instrument from its Configuration port on the front panel, to a 'VT100' compatible terminal (for example, an IBM-compatible PC running a terminal emulation package).

5.3 PREPARING TO RUN THE CONFIGURATOR

Getting ready to run the Configurator consists of two main steps:

1. Connecting the processor unit to a PC
2. Setting the control efficiency of the instrument.

5.3.1 Connecting to a PC.

The CONFIG port on the primary processor front panel should be connected to the PC RS232 port using a cable fitted with an RJ11 connector at one end and (typically) a 9-way 'D-type' connector at the other (Eurotherm part no. DN026484). The connector pinouts are detailed in Chapter 2, Installation. If further details are required, refer to your PC manual.

Notes:

1. To configure a redundant-mode instrument (dual synchronised processors), the terminal PC must be linked to the primary processor, not the secondary.
2. It is recommended that if a mains-powered PC is to be used, that it be isolated from the Processor Module by a Comms isolator. (For details, Chapter 2 section 2.4.2 Processor Module, should be referred to).

5.3.2 Setting the control efficiency

If the Configurator is to be used without the database running, these next few paragraphs may be skipped and the description rejoined at section 5.4.

Running the Configurator with the database running can affect the control efficiency of the instrument. The control efficiency is the percentage of CPU time spent on control tasks (i.e. updating function blocks.) Any diversion from this task will cause a fall in control efficiency.

100% efficiency can never be attained because there will always be minor ancillary tasks occupying the CPU's time but, under normal control activity with no major diversion, typical control efficiency will range from 80% to 95%.

How much the efficiency falls due to diversion to configuration tasks depends upon whether the Supervisor is set up as non-redundant or redundant, and upon whether the Configuration speed options (Options.CONFspd) bit has been set TRUE or FALSE in the header block of the control strategy (see 'NON REDUNDANT (SIMPLEX) SYSTEM, below).

The Options.CONFspd bit can be set using LINtools when creating a database, or more slowly using the Configurator.

5.3.2 CONTROL EFFICIENCY (Cont.)

NON-REDUNDANT (SIMPLEX) SYSTEM

With *CONFspd* set to TRUE, the processor will always spend 80% of its time updating blocks in the control strategy, leaving a fixed 20% available for configuration tasks. However, if the Configurator is not running, this 20% will be unused and thus wasted. This setting is used for fast-changing processes that are never far from some critical point, where control tasks must always take priority over temporary configuration tasks.

With *CONFspd* set to FALSE (the default state), and the Configurator not running, the processor can spend up to 95% of its time updating blocks. But with the Configurator running, this can fall to 50%, leaving up to 50% available for configuration tasks. This setting is used for slow-changing processes in which a temporary downgrade of control can be tolerated occasionally.

REDUNDANT (DUPLEX) SYSTEM

In duplex mode, up to 30% of processor time can be needed for the high priority task of keeping the secondary and primary processors synchronised. This effectively puts an upper limit of about 70% on the time available for block updates, overriding any higher limit set by the *CONFspd* bit. The low priority task of running the Configurator has to take place in the gaps between higher priority tasks. The Configurator therefore tends to be slow, especially where large strategies are involved. Thus:

- With *CONFspd* set to TRUE, the primary processor spends up to 70% of its time updating blocks, whether or not the Configurator is being run. The synchronisation task leaves the configurator with a small percentage of time in which to run.
- With *CONFspd* set to FALSE, and the Configurator *not* in use, the primary processor is still limited to about 70% on updating blocks, but whenever the Configurator *is* run this is limited to 50%, as for non-redundant mode, and the Configurator can speed up by using the extra spare processing time available.

5.4 Running the configurator

This section describes accessing and quitting the Configurator using HyperTerminal®. If a different terminal program is being used, its user documentation should be consulted (if necessary) for the equivalent procedures.

5.4.1 Accessing the configurator initial menu

1 Power up all components and start Hyperterminal® (Programs/Accessories/ Hyperter-minal®). After entering a name for the link (if necessary) enter the 'Properties' menu and select 'VT100'. In Properties/Connect-to/Configure Connection, set the comms parameters as follows:

Baud rate = 9600, Data bits = 7, Stop bits = 1, Parity = 'Even'.

2 When the hyperterminal starts, the message

```
1  ANSI-CRT
>>>
```

appears. Type <**1**> for the *ANSI-CRT* option. If Modbus is enabled, the configurator *Initial menu* appears, as shown in figure 5.4.1. If Modbus is disabled, the *Main menu* appears instead, as shown in figure 5.5. (Modbus in enabled/disabled by means of the Options switch (SW2) on the backplane — see Chapter 2 section 2.3.5.)

Note: Modbus is not supported if redundant mode is selected (SW2). If redundant mode is selected, the main menu always appears.

```
      INIT    Choose option


                   >DATABASE - General configuration
                    GATEWAY  - MODBUS configuration
```

Figure 5.4.1 Configurator initial menu

Note: The appearing of the Initial or Main menus indicates that the Processor module has entered *configuration mode*.

Locate the cursor (>) at a menu item using the cursor keys, then press <Enter> to display the next level in the menu hierarchy. This is called *selecting* an item. In general, to access the next lower level of the menu hierarchy <Enter> is pressed. To return to the next higher level menu or close a ‘pop-up’ options menu the <Escape> key is pressed. <PageUp> and <PageDown> allow hidden pages in long tables to be accessed.

® Hyperterminal is a trademark of Hilgraeve Inc.

5.4.1 ACCESSING THE CONFIGURATOR INITIAL MENU (Cont.)

For keyboards without cursor-control keys, equivalent ‘control’ character combinations may be used, as indicated in Table 5.4.1. To use these, the <Ctrl> key is held down and the specified character typed.

Function	Key combination
Clear screen	<Ctrl> + W
Cursor Up	<Ctrl> + U
Cursor Down	<Ctrl> + D
Cursor Left	<Ctrl> + L
Cursor Right	<Ctrl> + R
Page Up	<Ctrl> + P
Page Down	<Ctrl> + N
Stop automatic update	<Ctrl> + V

Table 5.4.1 Cursor-control — equivalent key combinations

Some tables allow a value to be entered directly, or via a called-up menu. For direct entry, the first character(s) of the chosen option is (are) typed, followed by <Enter>. Alternatively, the menu can be accessed with <Enter> or <Tab> as the first character after the field is selected.

5.4.2 The Initial menu

The Initial menu (figure 5.4.1 above) lists two options — *Database* and *Gateway*. Database allows access to the Main menu for configuring a LIN database. This is described in section 5.5. Gateway allows access to the Gateway menu, for setting up a Modbus configuration, described in section 5.6.

5.4.3 Quitting the terminal emulation program

Exit from configuration mode must be done from the terminal by pressing <Escape> repeatedly until the main menu screen appears, then again to clear the screen. The processor is now out of configuration mode.

Note: You cannot stop/start/download/upload files via LINfiler (in the LINtools package) in a processor if it still in configuration mode. If you try, error 8333 (‘*Configurator in use*’) is reported. You must quit processor configuration mode before attempting these operations.

Caution

Always quit the primary processor from configurator mode after use. If you do not, an operator unaware that the Processor module is still in configurator mode might subsequently plug in a terminal and type <Enter> <Enter> — hoping to see the version and power-up/shutdown messages. The result could be totally unexpected because the configurator would react according to where it was left, e.g. if last used to start a database it would execute the start sequence (twice).

5.5 DATABASE CONFIGURATION

Figure 5.5 shows the Main menu, and 5.5.1 to 5.5.7 describe its items.

```
MAIN MENU  Select option

              >MAKE        - Create block
               COPY        - Copy block
               DELETE      - Delete block
               INSPECT     - Inspect block
               NETWORK     - Network setup
               UTILITIES   - Engineering utilities
               ALARMS      - Current Alarms
```

Figure 5.5 Configurator Main menu

5.5.1 MAKE

Installs function blocks in the control strategy. Select MAKE to display the SET MENU — the processor's resident library of block categories, detailed in the *LIN Product Manual* (Part number HA 082 375 U999). Note that every strategy must contain a 'header' block — a *T940* block — the only block initially available for a new strategy. Select a category to list its blocks. Figure 5.5.1a shows part of the screen display when LOGIC is selected, as an example.

```
LOGIC      Select type

               >PULSE
                AND4
                OR4
                XOR4
```

Figure 5.5.1a Logic category menu (upper part)

Select the block to be installed. The block *Overview* appears listing the block parameters, default values and units in a double 3-column format. Figure 5.5.1b shows the (default) overview for the PID block as an example.

Note: Blocks may not be configured if processor modules are synchronised

5.5.1 MAKE (Cont.)

BLOCK OVERVIEW

Refer to Figure 5-4 which shows the main features of a typical block overview, used to monitor and update block parameters. (Overviews can also be accessed via the COPY and INSPECT main menu options.) The overview is equivalent to a LINtools *Specification menu* and its fields have the same meanings, although data entry is different. Note that parameters being updated by incoming connections from other blocks are not specially indicated in a block overview.

```
  OVERVIEW Block: “NoName”   Type: PID   Compound:
— — — — — — — — — — — — — — — — — — — — — — — — — —
 Mode       AUTO                | Alarms
 FallBack   AUTO                |
                                | HAA      100.0     Eng
 PV           0.0       Eng     | LAA        0.0     Eng
 SP           0.0       Eng     | HDA      100.0     Eng
 OP           0.0       %       | LDA      100.0     Eng
 SL           0.0       Eng     |
 TrimSP       0.0       Eng     | TimeBase Secs
 RemoteSP     0.0       Eng     | XP       100.0     %
 Track        0.0       %       | TI       0.000
                                | TD       0.000
 HR_SP      100.0       Eng     |
 LR_SP        0.0       Eng     | Options    00001100
 HL_SP      100.0       Eng     | SelMode    00000000
 LL_SP        0.0       Eng     |
                                | ModeSel    00000000
 HR_OP      100.0       %       | ModeAct    00000000
 LR_OP        0.0       %       |
 HL_OP      100.0       %       | FF_PID     50.0     %
 LL_OP        0.0       %       | FB_OP       0.0     %
```

Figure 5.5.1b Overview — PID block

Title bar. Contains fields common to all overviews: *Block*, *Type*, and *Compound*. *Block* and *Type* have their usual LIN meanings; *Compound* is equivalent to *Dbase*. Please refer to the *LIN Blocks Reference Manual* (in the *LIN Product Manual*) for details of these fields. A blank *Compound* field denotes that the block database is local.

Note that the block is not installed into the control strategy until (at the minimum) its *Block* field has been assigned a value — i.e. tagname — and the database has been restarted.

5.5.1 MAKE (Cont.)

Overview data field entry. To update a parameter field, locate the flashing 'underline' cursor (_) at the field using the arrow keys, then proceed as described next for the different data field types. Some data fields display further nested levels of data when entered, as detailed in the following sections. Press <Enter> to access a deeper level; press <Escape> to return to a higher level.

Note that editing a database during runtime is possible but is not recommended. (Stopping the database is described in section 5.5.6.)

- **User-defined names.** Type in a name (8 characters max.) and press <Enter> to overwrite existing data. To insert characters, locate the cursor at the character to follow and type the insertions. A 'beep' warns that excess characters have been typed. To abort the current entry and leave the database unchanged, move the cursor to a field above or below the current field *before* pressing <Enter>, or press the <Escape> key.

  Note that, remote database names entered in the *Compound* field must be prefixed by an 'equals' sign (=) which is included in the character count.

  Pressing <Enter> with the cursor on the first character of the *Block* or *Compound* fields (before starting to type) accesses a *Full Description* page (Figure 5.5.1c shows an example). This page gives general information about the block and has a common format.

```
     FULL DESCRIPTION    Block: PID_1      Type: PID
─────────────────────────────────────────────────────────
     Refresh rate                      Ø.1Ø4Ø
     Server number                     2
     Compound:                         =Alpha
     Rate  ms                          10
     Execute time                      1234
```

Figure 5.5.1c FULL DESCRIPTION page for block (example)

Block. (Read/write). Block tagname.

Type. (Read-only). Block type.

Refresh rate. (Read-only). Time (secs) since the block was last scheduled to run. Note that for a control block the PID algorithm is not necessarily recalculated every time the block is scheduled.

Server number. (Read-only). Block's timescheduled task priority. Server 2 (top priority) executes all local blocks, and Server 3 (next priority) executes all remote blocks.

Compound. (Read/write). Name of the block's parameter database. A blank field means the block database is **local**, i.e. in the current processor. (Database names and their LIN addresses are specified via the main menu NETWORK option, described in section 5.5.5.)

5.5.1 MAKE (Cont.)

Rate ms. Rate is the minimum update period (i.e. maximum rate) at which an individual cached block is transmitted across the Local Instrument Network (LIN). The default is 10ms minimum, i.e. 100Hz maximum. Rate can be set between 10ms and 64s. Note that rate values are minimum update times only, and heavily loaded networks may not be able to reach the faster update rates.

Execution time This is the time taken in microseconds to execute a block (including connections etc.)

- **Parameter values.** Type in a value and press <Enter> to update the database. (Read-only parameters do not accept new values.) The processor module automatically adds a following decimal point and padding zeros if needed, but before a decimal point a zero must always be typed, e.g. 0.5, not .5.

  Pressing <Enter> with the field selected, before starting to type, accesses a *Full Description* page for the parameter (Figure 5.5.1d shows an example).

```
FULL DESCRIPTION   Field: PV      Block: PID_1     Type: PID
— — — — — — — — — — — — — — — — — — — — — — — — —
   Value           8Ø.1                          Real32
   Input           SIM 1.OP
```

Figure 5.5.1d FULL DESCRIPTION page for parameter (example)

Field, Block, Type. Read-only fields.

Value. (Read/write) Parameter value, editable as for the Overview.

Real32. (Read-only) Value *type* (Real32 = floating point number)

Input. (Read/write) Defines the source of any connection to the parameter from another block, as **Block Tagname.Output Mnemonic**. A blank field means no connection. To make or edit a connection, type in the source block tagname and output mnemonic (e.g. **SIM 1.OP**, or **SEQ.DIGOUT.BIT3**), then press <Enter>. Invalid data is ‘beeped’ and is not accepted. The field is not case sensitive. To delete a connection, type <space> then press <Enter>.

Note: See CONNECTION TYPES (below) for information and advice on types of database connections.

- **Parameter units.** Type in a value and press <Enter>. All other related units in the database automatically copy the edited unit. Pressing <Enter> with the field selected, before starting to type, accesses the parameter *Full Description* page (as for the value field).

5.5.1 MAKE (Cont.)

- **Options menu fields.** Press <Enter> to display a pop-up menu of options for the field. Figure 5.5.1e shows an example (PID Mode) in part of an overview page.

```
   OVERVIEW   Block: PID_1   Type: PID      Compound:
----------------------------------------------------------------
                                       |
        Mode                           |     Alarms
        Fallback +-------+             |
                 |>HOLD  |             |
                 | TRACK |             |        HAA      100.0    Eng
        PV       | MANUAL| g           |        LAA      0.0      Eng
        SP       | AUTO  | g           |        HDA      100.0    Eng
        OP       | REMOTE|             |        LDA      100.0    Eng
        SL       | F_MAN | g           |
        TrimSP   | F_AUTO| g           |        TimeBase          Secs
        RemoteSP +-------+ g           |        XP       100.0    %
        Track                          |        TI       0.000
                                       |        TD       0.000
```

Figure 5.5.1e Pop-up options menu (example)

Using the ‘arrow’ keys, move the cursor (>) to a menu option and select it by pressing <Enter>. (Disabled options may not respond to selection.)

A quicker alternative to accessing the pop-up options menu is to type the required option, or enough of its *initial letters* to uniquely specify it, directly into the selected field and then press <Enter>. E.g. entering just **H** selects HOLD; entering **F_M** selects F_MAN (Forced Manual).

- **Alarms field**. Press <Enter> to display a 4-column *Alarms* page listing alarm *name* (e.g. HighAbs), *acknowledgement* (e.g. Unackd), *status* (e.g. Active), and *priority* (0 to 15). Update the acknowledgement or priority fields (the only editable ones) by typing in a value and pressing <Enter>. (Any single letter can be used for the acknowledgement field.) Figure 5.5.1f shows an example Alarms page.

```
    Alarms                    Block: PID_1     Type: PID
--------------------------------------------------------------
    Software                  Unackd         Active      15
    HighAbs                   Unackd         Active      15
    LowAbs                                               0
    HighDev                                  Active      10
    LowDev                                               2
    Combined                  Unackd         Active      15
```

Figure 5.5.1f Alarms page (example)

5.5.1 MAKE (Cont.)

- **Bitfields.** Contain eight (or sixteen) binary digits showing the logic states of a corresponding set of up to eight (or sixteen) parameters. To edit the bitfield directly, type in a bit-pattern then <Enter> it. Alternatively, press <Enter> to display a *Full Description* page listing the parameter TRUE/FALSE or HIGH/LOW states (in the same format used for LINtools Specification Menu bitfields). Figure 5.5.1g shows an example. Alter a logic state by locating the cursor on the state, typing in T(rue) or F(alse), and pressing <Enter>. (A bit may be read-only.)

```
 FULL DESCRIPTION  Field: ModeAct   Block: PID_1    Type:PID

 NotRem      TRUE
 HoldAct     FALSE
 TrackAct    FALSE
 RemAct      FALSE
 AutoAct     TRUE
 ManAct      FALSE
 FAutoAct    FALSE
 FManAct     FALSE
```

Figure 5.5.1g FULL DESCRIPTION page for bitfield (example)

  To connect an input to a bitfield, press the → key and type in the block name/field name from which the connection is to be made.

  Note: See CONNECTION TYPES... (below) for information and advice on types of database connections.

- **Two- and four-digit ‘combined’ hexadecimal status fields.** Hex fields are marked with a ‘>’ sign and have the same format and significance as those found in LINtools specification menus. The digits show the logic states of a corresponding set of parameters, up to four per hex digit. To edit the field directly, type in new values then press <Enter>. Alternatively, press <Enter> to display a *Full Description* page listing the parameter TRUE/FALSE states and edit this list (as described for Bitfields, above).

CONNECTION TYPES IN A PROCESSOR MODULE DATABASE

There are three types of connection used in a processor module database: local connections, connections writing to a cached block, and connections from a cached block to a local block. The following explains how and when they are evaluated.

- **Local connections.** These are connections between two blocks that are both local to the processor module database. The connection is always evaluated immediately prior to the execution of the destination block’s update procedure, regardless of whether the source data has changed between iterations. With this sort of connection, any attempt to write to the connection destination is immediately ‘corrected’ by the next connection evaluation.

CONNECTION TYPES IN A PROCESSOR MODULE DATABASE (CONT.)

- **Connections writing to cached block.** These are connections whose destination block is a cached copy of a block in another instrument. The source of the connection can be either a local database block or another cached block. Such connections are evaluated only if the source and destination data do not match. All cached blocks in the database are processed at regular intervals, and whenever a change is detected a single field write is performed over the communications link.
- **Connections from cached block to local block.** These are connections where the source block is a cached copy of a block in another instrument, and the destination block is local to the processor module database. All cached blocks in the database are tested at regular intervals, and if a change in the block data is detected, then all such connections out of the cached block into local blocks are evaluated. The connections are not evaluated if the source data has not changed.

  This third type of connection is unique to redundant processor systems (duplex processors). Such connections are evaluated in this way to minimise the load involved in synchronising the databases of a duplex pair, whilst ensuring the coherence of the data between the primary and secondary processor units.

Caution

With this third type of connection, if the source block is unchanging — as can easily happen, particularly if all block fields are digital values — then the connections are not continually re-evaluated. This allows other tasks to write to the connection destination, leaving the source and destination of the connection with different values. You should ensure that your strategy does not write to connection destinations.

5.5.2 COPY

Creates duplicates of existing blocks. Select COPY from the main menu to display all the blocks in the control strategy, in semi-graphical format as shown in Figure 5.5.2. The blocks are displayed from left to right in order of creation. Move the cursor (>) to a block and press <Enter>. The block is duplicated and added to the strategy, and its Overview page automatically appears ready for parameterising. The duplicate retains all the original parameter values except for the *Block* field, which has the default tagname **"NoName"**. Input connections are not copied; nor are I/O block site numbers.

```
   COPY Select block
 >T100    SIM_1     TIC_100   PID_1     FIC_101
```

Figure 5.5.2 COPY display (example)

Pressing <Escape> returns the COPY display, where the copied block can be seen added to the list. Press <Escape> again to return to the top level menu.

5.5.3 DELETE

Deletes blocks from the control strategy.

Select DELETE from the main menu to display all the blocks in the control strategy, in the same format as for the COPY option described in section 5.5.2. Select a block and press <Enter>. The block and any connections *from it* are deleted, and the main menu returns to the screen.

Notes

1. The control database must be stopped, otherwise selecting DELETE results in an audible warning and no action. Stopping the database is described in the UTILITIES option described in section 5.5.6
2. A block cannot be deleted unless its input connections have been cleared.

5.5.4 INSPECT

Allows blocks in the control strategy to be inspected and updated. Select INSPECT from the main menu to display all the blocks in the control strategy, in the same format as for the COPY and DELETE options already described. Select a block and press <Enter> to display its overview page, ready for monitoring/updating.

Pressing <Escape> returns the INSPECT display, where other blocks can be selected for inspection. Press <Escape> again to return to the top level menu.

5.5.5 NETWORK

Allows block databases to be assigned *names* and node *addresses* on the LIN (Local Instrument Network) so that they can be configured as 'cached' blocks and run in a remote instrument. (The overview page of the cached block *Compound* field specifies the remote database name.)

Note: It is good practice when using cached blocks, to cache at least one block in each direction. This allows the status of the comms link between the nodes to be monitored from both ends via the cached blocks' software alarms. This 'bidirectional caching' also eliminates the fleeting software alarms that may otherwise be seen during processor changeover in a redundant mode system.

Select NETWORK from the main menu to display the *Network setup* page (initially blank). Figure 5.5.5 shows the top part of an example page with several databases already assigned.

```
      Network setup
— — — — — — — — — — — — — — — — — — — — — — — — — — —
      Alpha              >Ø1   |
      Beta               >Ø2
      dBase_1            >Ø3   |
```

Figure 5.5.5 NETWORK setup page (example)

To assign a new database name and address, locate the underline cursor at the left hand column of a blank row, type in a unique name (7 characters max.) and press <Enter>. The name appears added to the list together with a default node address **>ØØ**. (Non-unique or invalid names are 'beeped' and not accepted. Do not use ØØ or FF as node addresses). Move the cursor to the default address and type in the required node address (two hex digits). Press <Enter> to complete the assignation.

To edit an existing name or address, locate the cursor at a field, type in the new value, and press <Enter>. Invalid entries are not accepted.

To delete a complete name and address entry, edit its name field to a *space* character. Configurations downloaded from LINtools will have a Network page set up automatically.

5.5.6 UTILITIES

Allows program control, I/O calibration, and filing. Select UTILITIES from the main menu to display the Utilities options, shown in Figure 5.5.6.

```
      UTILITIES  Select option
                    >START      - Start runtime system
                     STOP       - Stop runtime system
                     SAVE       - Save database
                     LOAD       - Load database
                     FILE       - File page
                     APPLY      - Apply Changes
                     UNDO       - Undo Changes
```

Figure 5.5.6 UTILITIES options menu

START, STOP UTILITIES

Select START or STOP from the UTILITIES options menu and press <Enter> to start or stop the control program running in the processor.

Note: When you START a database in RAM it is automatically saved to the file in E: drive called *filename*.DBF, where *filename* is indicated in the *filename*.RUN file. It is then reloaded and started.

SAVE UTILITY

Names and saves a control program to a specified memory area. Select SAVE from the UTILITIES options menu — the default filename specification, **E:T940.DBF** is displayed. (The prefix **E:** directs the save to the local E: drive area of the processor; this is the only available memory area. To save a database to a remote instrument, prefix the filename specification by the node address of the instrument separated by a double colon, e.g. **FC::E:T940.DBF**).

Type in a new specification if needed, then press <Enter> to execute the save. After a short pause the processor module signals completion with the message: **'Type a key to continue'**. Typing any key returns the UTILITIES menu.

An invalid filename specification aborts the save, and an error message is sent, e.g. **'Save failed — Invalid device'**.

Notes:

1. When you START a database in RAM it is automatically saved to the file in E: drive called *filename*.DBF, where *filename* is indicated in the *filename*.RUN file. It is then reloaded and started.
2. Modifications to a control database are carried out on the RAM image only, not directly to the .DBF file in E: drive. They are copied to E: drive (overwriting the existing .DBF file) automatically as you restart the database, or when you do a SAVE operation.

5.5.6 UTILITIES (Cont.)

LOAD UTILITY

Retrieves a control program from a specified memory area and loads it to the processor RAM. Note that LOAD cannot be performed during runtime. Select LOAD from the UTILITIES options menu — the default filename specification, **E:T940.DBF** is displayed. Edit the specification if needed (to alter the filename or its source, as described in 'SAVE utility' above), then press <Enter>. After a short pause the processor signals completion as described for the SAVE option. Typing any key returns the UTILITIES menu.

An invalid filename specification aborts the load, and an error message is sent, e.g. **'Load failed — File not found'**.

FILE UTILITY

Permits access to the processor file page, allowing files to be deleted or copied, and the E: device to be formatted. The file page displays files in the **E**-device and also in a configurable remote **??::?:** device. To access a remote device, move the cursor to the **??::?:** field and type in the required node and device letter, e.g. **FA::M:**. Press <Enter> to display its files (up to a maximum of 20).

Move the cursor up and down the file list and tag files with an asterisk (*****) using <Enter>. Then move the cursor to the top column-head field and press <Enter> to display the function menu: *Copy*, *Delete*, *Find*, and — for E-device only — *Format*. Finally, select a function and press <Enter> to carry it out. (Note that the Find function has wildcard characters (**?**) to help you locate filenames containing known character strings.) Press <Escape> to return to the UTILITIES menu.

APPLY/UNDO UTILITIES

Limited database changes can be executed on-line from the terminal configurator. These changes include making and setting the parameters for blocks and creating and deleting connections. Any such changes made whilst the data base is running are 'provisional' and are not applied until APPLY is selected. These provisional changes can be discarded by selecting UNDO, before APPLY has been selected. UNDO has no effect once APPLY has been used.

> Note; If changes have been applied,, and a sync. is attempted, it will fail unless the primary database has been saved using either the root block's full save option, or it is stopped, saved and started from the terminal.

5.5.7 ALARMS

Select ALARMS to view the currently active alarms in the instrument. Move the cursor up and down the list; press <Enter> to acknowledge an individual alarm. Press **I** to inspect the block containing the alarm.

5.6 Modbus configuration

Figure 5.6 shows the Gateway menu, and paragraphs 5.6.1 to 5.6.4 describe its four items.

Note: The resident Modbus configurator works in a similar way to the Modbus configurator in the T500 LINtools package. Refer to the *T500/T550 LINtools Product Manual* (Part No. HA082377U999) for more information.

```
    GATEWAY    MODBUS configuration


                   >MODE       - Operating mode
                    SETUP      - Serial line
                    TABLES     - Register & bit configuration
                    UTILITIES  - File Load & Save
```

Figure 5.6 Gateway menu

5.6.1 MODE

Sets the operating state of the instrument to *Slave* or *Master**. Selecting MODE pops up a menu showing the current mode (Slave by default), as shown in Figure 5.6.1. Select a different mode if required.

```
    MODE    Operating mode
---------------------------------------------------------------
                    Mode      +--------+
                              |>Slave  |
                              | Master*|
                              +--------+
```

Figure 5.6.1 MODE menu

---

*Note: Master mode is not available with this release of software

---

5.6.2 SETUP

Selects characteristics for the serial line operation. Select SETUP to see a menu of four items — Baud rate, Parity, Stop bits, and Time out — in master mode, plus a fifth item — Slave No — if slave mode has been configured.

- Baud rate. Highlight and enter this item to see a menu of the available baud rates — 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200. Select and enter the required baud rate, which updates the SETUP display.
- Parity. Entering this item displays a menu of options — None, Odd, and Even. Select and enter the required parity.
- Stop bits. Enter this item, type in the required number of stop bits, and press <Enter> to update the SETUP menu. (*Only 1 or 2 stop bits are permitted*).
- **Time out.** Enter a *Time out* value, in the range 0 to 65.5 seconds. In slave mode, this parameter specifies a watchdog period for all tables. That is, if a table has not been accessed for *Time out* seconds, the *Online* bit in the slave mode diagnostic register for that particular table resets to zero. In master mode, *Time out* specifies a maximum period between the end of a master's request for data to the start of the slave's response. If this time is exceeded, the *Online* bit in the master mode diagnostic register for the particular table concerned resets to zero.
- **Slave No.** (Slave operating mode only). Input a 'slave number', i.e. the address on the Modbus serial link of the slave device being configured. Slave addresses are in the range 01 to FF hexadecimal, but note that for some equipment FF is invalid.

When you have finished configuring the SETUP menu, press <Escape> or the right-hand mouse button to return to the Gateway menu.

5.6.3 Tables

This item accesses the **Tables list**, described below. To view the tables list, highlight TABLES and press <Enter>.

TABLES LIST

The tables list provides an overview of the sixteen tables in the Modbus configuration, through which you create tables and specify their types, offsets, sizes, and — for master mode — function codes, scan counts and slave numbers. The tables list also accesses individual table menus for detailed configuration (database mapping) — see 'TABLE MENUS' below.

Figure 5.6.3a shows an example tables list with Table 1 configured as a register table. The first four columns — Table, Type, Offset, and Count — are common to both the slave and master modes of operation. The remaining three — Functions, Scan count, and Slave No — appear only in a master mode configuration (not shown in the figure).

Table	Type	Offset	Count
1	Register	0	16
2	Unused	0	0
3	Unused	0	0
4	Unused	0	0
5	Unused	0	0
6	Unused	0	0
7	Unused	0	0
8	Unused	0	0
9	Unused	0	0
10	Unused	0	0
11	Unused	0	0
12	Unused	0	0
13	Unused	0	0
14	Unused	0	0
15	Unused	0	0
16	Unused	0	0

Figure 5.6.3a Modbus tables list — slave mode

The functions of tables list column headings are described next.

Table. This is the table number, which is not editable. If you highlight and <Enter> a table number field — for a table with a *Type* other than **Unused** — you access the *table menu* for that table. This is described in 'TABLE MENUS' below.

5.6.3 TABLES (Cont.)

Type. This field, which defaults to **Unused**, lets you create or change the table type. Enter a *Type* field to see a menu of four options. Select one and press <Enter> to create a new table or convert an existing one to a new type. Note that other fields in the tables list associated with your selection automatically adopt default values.

The *Type* options are:

- **Unused.** The table is deleted.
- **Register.** This type of table maps LIN database parameters onto standard 16-bit Modbus registers.
- **Digital.** This type of table maps LIN digital, boolean or alarm values onto bits in the Modbus address space.
- **Diagnostic.** This is a special table, similar to a register table, but the values in the table have pre-defined values that are used to control the Modbus operation, or present diagnostic information to the database.

Offset. This field selects the start address of the table on the Modbus network. The values used here are the actual values used in the address field of the Modbus messages, i.e. the 'protocol addresses' (see Chapter 7). Note that PLCs differ in the correspondence between their register or bit addresses and the protocol addresses.

Count. This field lets you specify the number of registers or bits in a table. It allows the size of register and digital tables to be changed from their default values of 64 registers or bits, respectively, to optimise the use of memory. Diagnostic tables are fixed at 32 registers.

Functions. (*Master mode only*). This field allows you to enable or disable the default Modbus function codes that can be used with a particular Modbus table type. Modbus function codes define the type of data exchange permitted between master and slave instruments via a particular table.

To disable a default function code, highlight it with the mouse and press <Enter> to see a menu of '–' and the default code number. Selecting and entering '–' disables that code for the table concerned. Select the code number again to re-enable it if required.

Scan count. (*Master mode only*). This sets the maximum number of registers (register table) or bits (digital table) that can be read or written in a single Modbus transmission. *Scan count* defaults to the same value as *Count*, i.e. as the table size, which results in the whole table being updated each polling cycle. If *Scan count* is made less than *Count* for a particular table, it takes more than one cycle to be updated but the overall polling cycle speeds up. This may be required for Modbus devices with limited buffer sizes.

Slave No. (*Master mode only*). This specifies the hexadecimal slave number value of the instrument on the Modbus network in which the data registers or bits associated with this master table are located.

5.6.3 TABLES (Cont.)

TABLE MENUS

Individual table menus are accessed from the tables list by highlighting its table number (in the first column headed *Table*) and pressing <Enter>. To highlight fields you can move the arrow cursor around a table menu using the mouse, or the PC's <Home>, <End>, and cursor keys.

Table menus allow you to configure the mapping between the LIN database fields and the Modbus addresses. Figure 5.6.3b shows the default table menu for a register (or diagnostic) table. Note that table headings differ for register and digital tables, but that some fields are common to both — *Field, DB Write*, and *MOD Write*. The functions and usage of all the table menu field types are described next.

Register	Field	DP	Format	DB Write	MOD Write	Value
0		0	Normal	Enable	Enable	>0000
1		0	Normal	Enable	Enable	>0000
2		0	Normal	Enable	Enable	>0000
3		0	Normal	Enable	Enable	>0000
4		0	Normal	Enable	Enable	>0000
5		0	Normal	Enable	Enable	>0000
6		0	Normal	Enable	Enable	>0000
7		0	Normal	Enable	Enable	>0000
8		0	Normal	Enable	Enable	>0000
9		0	Normal	Enable	Enable	>0000
10		0	Normal	Enable	Enable	>0000
11		0	Normal	Enable	Enable	>0000
12		0	Normal	Enable	Enable	>0000
13		0	Normal	Enable	Enable	>0000
14		0	Normal	Enable	Enable	>0000
15		0	Normal	Enable	Enable	>0000

Figure 5.6.3b Register table menu — default

Register. (*Register and diagnostic tables only*) This column shows the Modbus address of the particular register. The first register in the table takes its address from the *Offset* value given to the table via the tables list (described above). The remaining (read-only) addresses follow on consecutively.

5.6.3 TABLES (Cont.)

Digital. (*Digital tables only*) This column shows the Modbus address of the digital bit on that particular line of the table. If the line contains a bitfield rather than a single bit, the address shown is that of the first bit in the bitfield. Mappings may be made for a single bit, or for an 8- or 16-bit field, according to the value defined in the *Width* parameter (see later). The very first bit address in the table takes its value from the *Offset* given to the table via the tables list. The remaining (read-only) addresses follow on according to the numbers of bits on each successive line of the table (1, 8, or 16).

Field. This is the LIN database field that you can map onto the Modbus address, or leave blank. Select a field with the cursor and type in and enter a block name plus parameter (and subfield if needed), separated by full stops (periods), e.g. **PV1.Alarms.Software**. Note that if you try to enter an analogue parameter into a digital table *Field*, your entry is ignored. You can however enter any type of parameter into a register (or diagnostic) table. Note also that in a digital table you cannot enter or overwrite a database parameter if this would force an entry lower down the table to change its address (*Digital* value).

DP. (*Register and diagnostic tables only*) This column can be used for either of two functions: specifying a decimal point position, or creating a 32-bit register.

- **Decimal point position.** *DP* can store a decimal point scaling factor that is used when converting floating point numbers to 16-bit Modbus registers. For this purpose, enter an integer from 0 to 4; the *DP*-value represents the number of decimal places in the converted number.
- **32-bit register.** (*Register tables only*) A 32-bit register is created by 'joining' a consecutive pair of 16-bit registers, as described below. Note the restrictions that are applied to ensure that the 32-bit value created is transferred indivisibly:

  1. The multiread function (3) and multiwrite function (16) must both be enabled.
  2. The scan count must be even.
  3. The first register of the pair must be at an even offset within the table.
  4. The first register of the pair must not be the last register in the table.
  5. The second register of the pair must not already be assigned to a database field.
  6. The field type of the 32-bit register pair must be 32-bit long signed or unsigned, 32-
     bit real or a string. For a string, only the first four characters are transferred.

  To create a 32-bit register pair, enter 'd' (or 'D' — the case does not matter) in the *DP* field of the first register of the pair. This causes the register's *DP* to adopt the value '**D**', and the following register the value '**d**'. If any of the above restrictions are violated, your entry will be rejected.

  When the first register of the 32-bit pair is assigned to a database field, the second register automatically copies the same field name; assigning the name and the *DP* can be done in either order. You can restore a 32-bit register pair to individual 16-bit registers by changing the first register's *DP* to 0-4.

5.6.3 TABLES (Cont.)

Format. (*Register and diagnostic tables only*) This column specifies the format of the data in the register — normal or BCD (binary coded decimal). Normal format means that the data is a simple 16-bit integer. In BCD format the value is first limited to the range 0-9999, and then stored as four 4-bit nibbles in the register. The units are stored in the low order nibble, the tens in the second nibble, the hundreds in the third, and the thousands in the high-order nibble. BCD format allows the data to be used with certain devices such as displays.

Note: *Format* is ignored in 32-bit registers.

Width. (*Digital tables only*) This column indicates the number of bits contained in the associated field. The default *Width* is 16, but it automatically updates when you allocate a parameter to the field. Allocated field 'widths' are read-only, but you can specify the width of an unallocated field by highlighting its *Width* value and entering a valid number — in the range 1 to 16, but normally only 1, 8, or 16. Note that you are not able to edit a *Width* value if this would force an entry lower down the table to change its address (*Digital* value).

DB Write. This column allows you to prevent selected values in the LIN database from being overwritten by values received across the serial link. Highlight the required *DB Write* field and press <Enter> to see a menu of options — Enable and Protect. Select *Protect* to write-protect the LIN database parameter, or *Enable* to allow overwriting.

Note: For a 32-bit register pair, *DB Write* applies only to the first register. The *DB Write*-value of the second register is ignored.

MOD Write. This column allows you to prevent selected values in the LIN database from being written to their associated Modbus registers or bits. Highlight the required *MOD Write* field and press <Enter> to see a menu of options — Enable and Protect. Select *Protect* to write-protect the Modbus register/bit(s), or *Enable* to allow overwriting.

Notes:
1. The easiest way to globally protect an entire table — in a gateway operating in master mode — is to disable its write function codes (5 and 15, or 6 and 16) in the tables list.
2. For a 32-bit register pair, *MOD Write* applies only to the first register. The *MOD Write*-value of the second register is ignored.

Value. This column shows the current 16-bit value of the field in 4-digit hexadecimal representation. 'Value' is read-only.

5.6.4 Utilities

The Utilities menu lets you save and load Modbus configurations. Files may be copied to and retrieved from the local processor module, or from a remote instrument across the LIN. The Modbus configuration is stored in a file with extension **.GWF**, and the root filename should be the same as that of the corresponding database **.DBF** file.

Select UTILITIES in the Gateway menu to see the options shown in Figure 5.6.4.

```
UTILITIES    File Load & Save
                  >SAVE       - MODBUS Configuration
                   LOAD       - MODBUS Configuration
```

Figure 5.6.4 UTILITIES menu

- **SAVE.** Select SAVE and press <Enter> to see the default filename specification **E:T940.GWF** (for the T940 processor). To save the current Modbus configuration under the default filename press <Enter> again. If you want to save it under a different filename, edit the default one first before carrying out the save operation.

  Note: An existing file with the same filename is ***overwritten without warning***.

- **LOAD.** Select LOAD and edit the default **E:T940.GWF** if required to the filename to be loaded. Press <Enter> to load the specified configuration.

  An error message appears if the specified file cannot be found.

  Note: The current Modbus configuration is ***overwritten without warning***.

This page is deliberately left blank

Chapter 6
ERROR CONDITIONS & DIAGNOSTICS

This chapter describes the various ways to tell if a fault has occurred in the process supervisor, (not in the process being supervised).

The main topics covered are:

- Error indication types (§6.1)
- Processor module front-panel error displays (§6.2)
- Power-up failures (§6.3)
- Power-on self-tests (§6.4)
- Diagnostic blocks (§6.5)
- Error numbers (§6.6)

6.1 ERROR INDICATION TYPES

Error indications include:

- **LEDs.** The processor module LEDs are the most immediate source of error and instrument status information concerning basic I/O system (BIOS) start, watchdog functions and normal running. During BIOS start, a number of the front panel LEDs are intermittently illuminated to indicate the BIOS status. If a processor start fails, the pattern that these LEDs adopt prior to the failure is helpful to service engineers, so it is recommended that this pattern is recorded (along with the unit serial number) before a service call is made.
- **Error messages.** A large number of highly specific error messages are transmitted (mainly during start-up) by the processor modules, which can be viewed if a VDU terminal is attached via the EIA232 front-panel CONFIG port (or the other comms ports).
- **POSTs.** The results of power-on self-tests (POSTs) can be used to pinpoint error conditions in the instrument.
- **Diagnostic blocks.** A range of function blocks can be included in the running strategy database to provide diagnostic information on various topics, including the redundancy mechanism, the ICM (inter-processor communications), the I/O interface, and others.

6.2 PROCESSOR MODULE FRONT PANEL ERROR DISPLAYS

6.2.1 LEDs

Figure 6.2.1 shows the processor module front-panel LEDs. Table 6.2.1 specifies their functions.

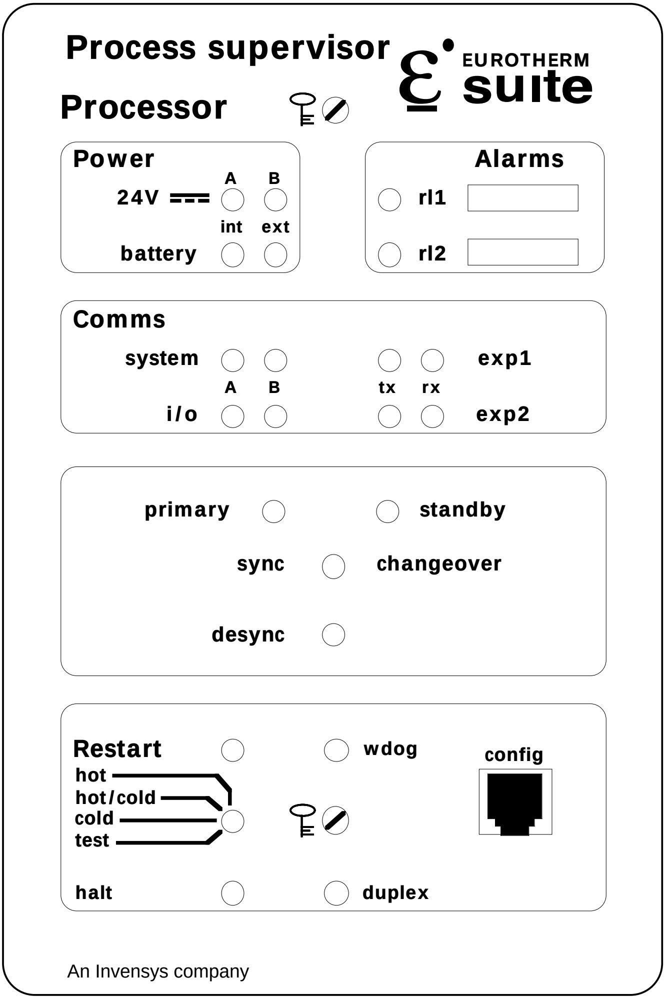


Figure 6.2.1 Processor module front panel

6.2.1 LEDs (Cont.)

Group	LED	Function		Diagnostic Value
Power	Power A	Green...................... Main power input valid Off........................... Main power input failed		
	Power B	Green...................... Auxiliary power input valid Off........................... Auixiliary power input failed		
	backup ext	Green...................... External battery power valid Off........................... External battery power failed	Off until start-up complete	
	backup int	Green...................... Internal battery power valid Off........................... Internal battery power failed	Off until start-up complete	
Alarms	rl1	Yellow..................... Alarm active Off........................... Alarm not active		08
	rl2	Yellow..................... Alarm active Off........................... Alarm not active		04
Comms	System A	Green...................... System A communications valid Red......................... System A communications hardware failure Flashing Red/Off..... System A communications cable fault		
	System B	Green...................... System B communications valid Red......................... System B communications hardware failure Flashing Red/Off..... System B communications cable fault		
	I/O A	Green...................... System A communications valid Red......................... System A communications hardware failure Flashing Red/Off..... System A communications cable fault		
	I/O B	Green...................... System B communications valid Flashing Green/off.. Remote unit fault (Profibus comms. only) Red......................... System B communications hardware failure Flashing Red/Off..... System B communications cable fault		
	Exp1 Tx / Rx	Intermittent yellow....Communications taking place		Rx = 20 Tx = 10
	Exp2 Tx / Rx	Intermittent yellow... Communications taking place		Rx = 80 Tx = 40
Startup	Primary	Green...................... This CPU is primary Off........................... This CPU is not primary Flashing.................. Powered up but no database is running		02
	Standby	Yellow..................... This CPU is secondary and synchronised Off........................... This CPU is not secondary synchronised Flashing.................. Synchronisation process in progress.		01
	wdog	Green................................CPU not in reset Red...................................CPU in reset Red/Green alternating....... Power up sequence in progress		
	Duplex	Green.................................Redundancy communications valid Off.....................................System in non-redundant mode Red/Green alternating....... Inter CPU communications failed		

Table 6.2.1 Processor LED functions

6.2.2 Processor failure modes

The front-panel LEDs can indicate directly the following processor module failure or potential failure modes: power loss, watchdog failure, desynchronisation, loss of primary status, database halt, ALIN or I/O communications failure, and ICM failure.

When a processor, which is running as one of a redundant pair, fails, it usually changes its redundancy state in response to the failure, e.g. from primary to secondary, or from synchronised to unsynchronised. Figure 6.2.2 maps out various ways in which a pair of processor modules might fail, and shows how they change redundancy state as a consequence.

In the figure, the boxes represent possible processor module states, and the arrowed lines between boxes represent transitions from one state to another. Arrows are labelled with the fault conditions causing the transition. ‘Primary processor module’ and ‘secondary processor module’ are abbreviated as ‘#1’ and #2’ respectively. The front panel LEDs help to identify what state each processor is in, as well as the nature of any failure. (The 'Comms' LEDs will be on, off or flashing as indicated in table 6.1.)

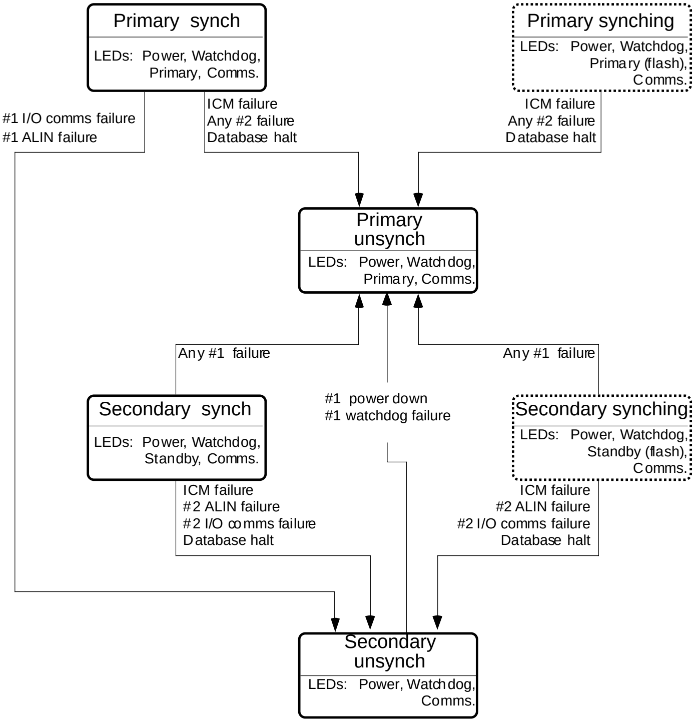


Figure 6.2.2 Processor unit failure modes

6.2.3 Power failure

In the event of a power failure all the LEDs associated with the affected processor are extinguished and the processor adopts the OFF state.

PRIMARY PROCESSOR MODULE

A power failure in the primary unit of a redundant pair, will cause the secondary unit to adopt the Primary Unsynch state.

If the secondary was not synchronised at the time, the database halts. The new primary's PRIMARY LED flashes to show that the database is not running.

If the secondary was synchronised at the time of takeover, the database continues to run in the new primary (PRIMARY LED on continuously).

SECONDARY PROCESSOR MODULE

A power failure in the secondary unit of a redundant pair, will cause the primary unit to enter the Primary Unsynch state.

6.2.4 Watchdog failure

In the event of a watchdog failure of a processor module, the green watchdog LED is illuminated red and the affected processor module enters a 'Watchdog fail' state.

In this state the indications given by the standby, primary, and comms LEDs are unreliable and should be ignored. Operation of the RESTART button resets the watchdog and restarts the CPU if this is possible.

On watchdog failure of a processor module in redundant mode, the surviving processor module adopts (or maintains) the PRIMARY UNSYNCH state. And as in the case of power failures, the survivor runs the database only if it was synchronised before takeover, halting it otherwise.

6.2.5 ICM (Inter-CPU Messaging for redundancy) failure

Note: An ICM failure is not associated with any single processor module, and so is not classed as either primary or secondary in figure 6.2.2.

An ICM failure is indicated by the standy and duplex LEDs when the primary and secondary processors can no longer communicate with each other across the internal high-speed link, making database synchronisation impossible to maintain. Figure 6-2 shows that an ICM fail causes desynchronisation of the processor modules, but no primary/secondary changeovers.

STANDBY LED:	Off
Duplex LED off:	No communications link established
Duplex LED flashing red/green:	Communications possible, but not taking place (usually due to a 'desynch' request.)

6.2.5 ICM FAILURE (Cont.)

ACTION IN THE EVENT OF ICM FAILURE

In the event of an ICM failure the processors desynchronise. The control strategy must be designed to present the supervisory system with an appropriate alarm to annunciate this state. (E.G. use the RED_CTRL block's *ICM_Ok* status bit).

If the ICM does fail, the secondary processor module should be replaced. If this solves the problem re-synchronise the processors. If the fault persists, the running, primary processor module is the most likely cause and should be replaced. Initially the original secondary should be re-fitted as it is unlikely to be faulty and will still retain the current database in memory, with the parameter values existing at the time of desynchronisation. The faulty primary, should now be removed, this causing the secondary to take over as sole primary but with a stopped database. If appropriate, restart the existing database by powering down and then up again. Otherwise, reload a 'default' database and restart it in the new primary. ***This last option is a cold start and requires manual supervision of the plant during the transition***.

Note: A fault in the backplane is a possible but unlikely cause of ICM failure.

6.2.6 ALIN failure

This occurs when a processor is not communicating over the ALIN, due to a damaged or disconnected ALIN cable or to a hardware (electronics) failure.

An interconnection failure causes the relevant comms LEDs associated with the affected processor module to flash on and off. A hardware fault is indicated by the relevant comms LEDs either being illuminated continuously red (system or i/o ports) or being off (exp1/2 ports).

An ALIN failure in a synchronised primary processor module causes primary/secondary changeover and loss of synchronisation, i.e. Primary synch adopts Secondary unsynch, and Secondary synch adopts Primary unsynch.

Figure 6-2 shows that if an unsynchronised primary processor module suffers an ALIN failure no changes of state occur (there are no arrows leading *out* of the PRIMARY UNSYNCH box).

In the event of an ALIN fail in a synchronised secondary processor, it adopts the Secondary unsynch state (yellow standby LED off), and the primary processor module correspondingly desynchronises to the Primary unsynch state). If the secondary processor module was unsynchronised at the time of the failure, no change of state occurs.

EFFECT OF ALIN FAILURE ON REDUNDANCY MODE CONTROL

ALIN failure affects the ability to synchronise processor modules. An ALIN-failed secondary processor cannot successfully be synchronised with the primary by pressing the primary's synch switch, for example. Attempts to do this are inhibited by the redundancy control software, and this is indicated by the yellow standby LED's lack of response.

6.2.7 Database stop

If the database in the primary processor stops running for any reason, the green primary LED starts to flash and the processor modules desynchronise. Attempts to resynchronise are inhibited by the redundancy control software. The yellow standby LED of the secondary processor is extinguished.

6.2.8 I/O Comms failure

This occurs if a processor detects a hardware or interconnection fault in the link with the I/O system(s) it is attempting to communicate with. If a fault is detected, this is indicated by the relevant i/oA, i/oB, exp1 or exp2 LED's going red; either continuous red (hardware fault) or flashing red/off (connection fault).

As shown in figure 6.2, if an I/O failure is detected with redundant processors in any state other than 'unsync', the two processors will go into their unsync states. If the failure is in the primary, then the secondary will become the new primary and assume control, and the original primary will become the secondary. If the failure is in the secondary, no change-over occurs.

6.3 POWER-UP FAILURE

6.3.1 Processor unit power-up routine

A number of error conditions can occur during the power-up phase of a processor module. This power-up routine is described in Chapter 4, and this should be referred-to for detailed information. Various messages are generated by the processor module during power-up, and these can be displayed on a VDU terminal attached to the front-panel EIA232 CONFIG connector (see chapter 2). These messages appear when configuration mode is accessed. Full lists of error messages are given in section 6.6 of this chapter.

Figure 6.3.1a charts the power-up routine in a simplified schematic form, and figure 6.3.1b shows the hot start ‘subroutine’ that may be called by the main power-up routine. The two flow diagrams also show various error conditions.

6.3.1 PROCESSOR UNIT POWER-UP ROUTINE (Cont.)

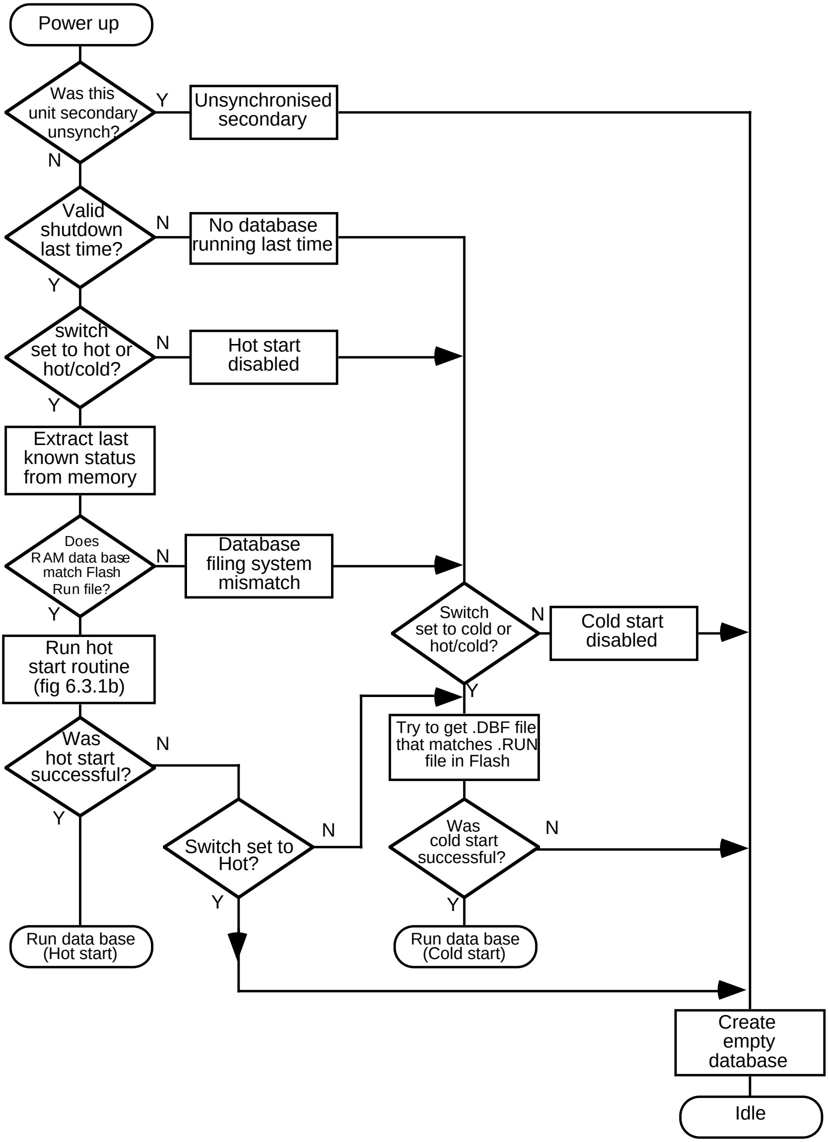


Figure 6.3.1a Processor unit power-up routine flowchart — simplified

6.3.1 PROCESSOR UNIT POWER-UP ROUTINE (Cont.)

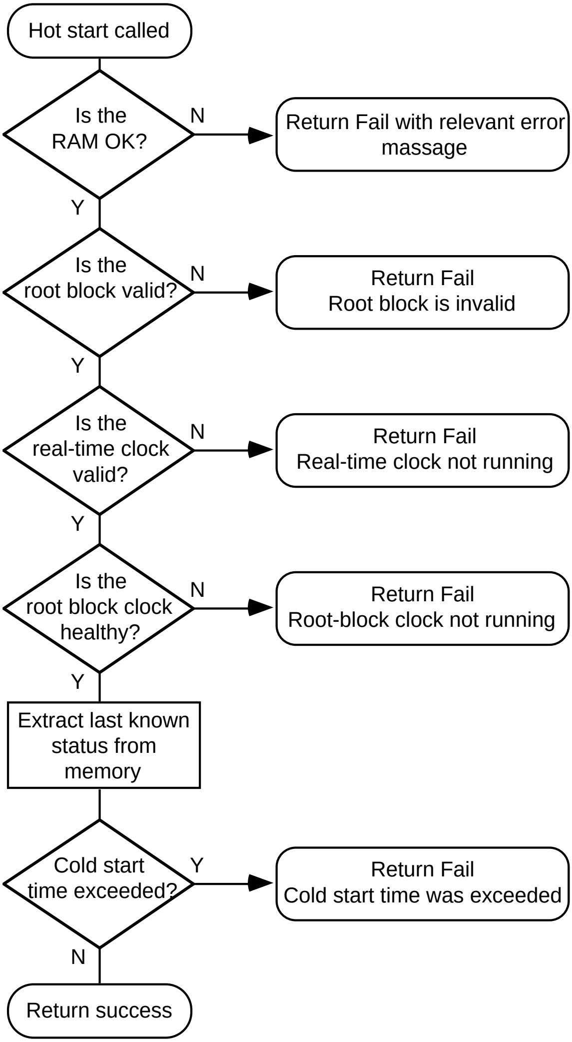


Figure 6.3.1b Processor unit warm start routine flowchart — simplified

6.4 POSTs (power-on self-tests)

Whenever a processor module is powered-up, it automatically performs a series of diagnostic tests. For these tests to work, a considerable amount of hardware must function correctly, in particular the PROMs and processor module.

POST results can be displayed on a VDU terminal plugged into the EIA232 CONFIG port in the processor front panel as described in Chapter 5 of this manual.

Note: It is recommended that the terminal screen be cleared <Ctrl>+<W> prior to use. If the screen has not been cleared the POST output may merge with the existing display and be unreadable.

At switch on, the Basic I/O system (BIOS) starts running and checks that the Central Processor Unit (CPU)* is operating correctly. This stage of power-up is apparent by the intermittent lighting of what are called the 'BIOS LEDs' shown in figure 6.4, below. Should the CPU fail to initialise fully, the final pattern of these LEDs may be of use to service engineers, but is not interpretable by the user.

---

* Note: This CPU is a part of the internal electronics of the 'Processor Module' and the two terms should not be confused.

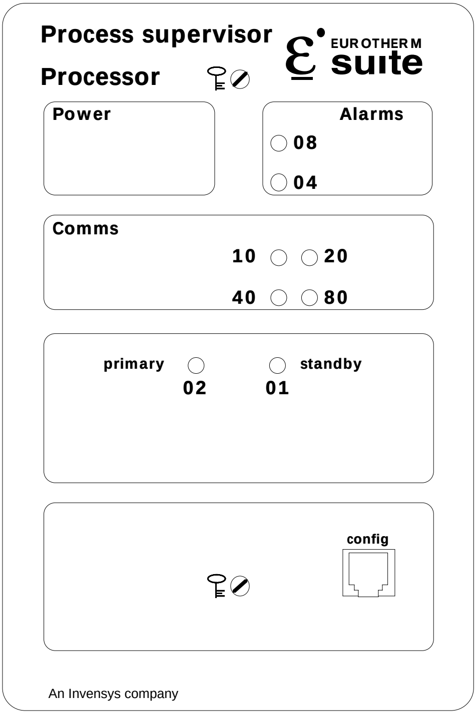


Figure 6.4 BIOS LEDs and their code values

6.4 POWER-ON SELF TESTS (Cont.)

Once the CPU is running, it runs the Boot ROM (Flash disk) which enables the system monitor (SMON). If the system monitor is not entered within one second, the start-up process continues with the loading of the application and system code from the FLASH ROM (accessible at the rear of the unit). At this point, a 1 second entry point for a second monitor (the 'M' monitor) appears. Refer to Chapter 9, Section 9,4 for details of the monitors.

The POST now checks

1. that all the electronic systems hardware is available for the Communications protocols required by the software. If not, this is deemed to be a 'Serious Hardware Fault', and although the power-up sequence continues, the processor will not load a data base, and will stop with its Primary LED flashing on and off, and the relevant Comms LED in its fault condition.
2. the Static RAM (SRAM) signature† checksum is valid. If not the signature is re-written to the default state.

† Note: The signature contains details such as what the processor status was at the last power down (e.g synched secondary of a redundant system).

If the start up mode switch is set to 'test' the SRAM is completely rewritten. This means that a cold start will be required before a data base can be run.

The System now attempts to start the software, determining first whether the options switch (SW2) on the backplane is set for redundant or non-redundant mode.

If redundant working is required, the primary/secondary status of each processor module is decided, according to the criteria in Chapter 4, §4.4.1, if necessary, using 'signature' data relating to last-times power down, auto synchronise states and so on.

A check is made to ensure that the ICM (inter-processor) communications are valid, and if so, the primary processor continues-its power up sequence, according to the mode selected at the front panel switch. The STANDBY LED starts flashing on and off when the primary starts to download data to the secondary.

If the ICM test fails, or if non-redundant working is required, the processor(s) continue the power-up sequence, according to the mode selected at the front panel switch .

A diagnostic test result code appears at the bottom of the screen, with a value of 0000 (check successfully completed), 0001 (only minor problems reported) or 0002 (major problem(s) reported). If the code is 0002, the processor fails to power-up.

6.4 POWER-ON SELF TESTS (Cont.)

ERROR TYPES

SERIOUS ERROR. Serious errors are reported if the unit's operation is impaired, but still capable of running. These errors are:

1. ALIN/Profibus hardware failure. Results in an inability to communicate with those systems using the particular protocol. The relevant front panel LEDs are continuously red.
2. Less than 8 MB of memory in dynamic RAM (see Note, below).
3. No config.txt file (normally installed on the system FLASH device, or created via the monitor)

FATAL ERROR. A fatal error is one where the unit's operation is impaired to the extent that it cannot continue to operate, or cannot start up. In a redundant system, the processor modules will desynchronise. This type of error is caused by:

1. No Flash memory available (normally because of a hardware fault).
2. SRAM check failed (normally due to a hardware fault).

Note: Less than 4 MB for versions prior to Version 2/2.

6.5 DIAGNOSTIC BLOCKS

Several diagnostic function blocks are available from the DIAG category, that can be installed in the control database at configuration time to help in diagnosing any error conditions that may arise in the running strategy. The VIEW facility in the LINtools package can then be used, via the ALIN network, to look at the fields in these blocks to find out what is happening. Alternatively a terminal emulation program running in a PC can be used to access the processor module's resident configurator, via the EIA232 CONFIG connector, to allow the diagnostic block parameters to be viewed in inspection mode.

These diagnostic blocks are described in the *LIN Product Manual* (part number HA082375U999) and you should refer there for details. Table 6.5 provides a brief summary of these blocks.

Block	Function
XX_XXXX	Block server tasks timing information, in priority order.
ICM_DIAG	ICM (Inter-CPU Messaging for Redundancy) diagnostics. Statistics on numbers and types of message passing between redundant processor modules.
RED_CTRL	Redundancy control block. Shows PRMT (processor redundancy management task) parameters. Can also be used to trigger processor module synchronisation, desynchronisation, and primary/secondary processor swap.
FTQ_DIAG	Low-level statistics on the queues maintained by the PRMT for interfacing with the various processes occurring in the Unit controller/supervisor.
MDBDIAG	Modbus diagnostics.
TOD_DIAG	Time-of-day synchronisation diagnostic block. Statistics on broadcasts, requests, receipts, rejections, etc.
SFC_DIAG	Sequence-related diagnostics and resource statistics. Number of configured and available resources.

Table 6.5 Unit controller/supervisor diagnostic block types

6.6 ERROR NUMBERS

This section lists the error messages that may be seen during the running of the Unit controller/supervisor connected to a terminal — either via the EIA232 port or over the ALIN or other serial ports.

6.6.1 Error number structure

All error conditions have an associated 4-digit number, and usually a corresponding text message as well. Error numbers are hexadecimal 4-digit groups. The first two digits show the 'package' that was running when the error occurred, and the last two specify the particular error associated with that package.

RUNNING PACKAGES

Packages are defined as:

82	File system (table 6.6.3a)	8D	Structured text system (table 6.6.2i)
83	Database system (table 6.6.2b)	8F	PCLIN/PC I/F package (table 6.6.2j)
85	Objects system (table 6.6.2c)	90	T1000 menu system (table 6.6.2k)
86	Trend system (table 6.6.2d)	91	Configuration files (table 6.6.2l)
87	Control config (table 6.6.2e)	92	Kernel items (table 6.6.2m)
89	Network error (table 6.6.2f)	9A	MODBUS codes (table 6.6.2n)
8B	Sequence database system (table 6.6.2g)	9B	Xec codes (table 6.6.2p)
8C	Sequence runtime system (table 6.6.2h)	A6	Asynchronous I/O (table 6.6.2q)

6.6.2 Error messages

Table 6.6.2 lists error messages package by package. Note that this is a complete list of all error messages generated by LIN-based systems, and therefore includes errors that are additional to those which can be generated by the processor module.

The error code FFFF means "unknown".

8201	Not mounted	8208	File not found
8202	Invalid device	8209	No handle
8203	Physical error	820A	Bad filename
8204	Not implemented	820B	Verify error
8205	Format error	820C	File locked
8206	Not present	820D	File read-only or No key fitted
8207	Device full		

Table 6.6.2a File system error codes (82xx)

6.2 ERROR MESSAGES (Cont.)

8301	Bad template	8333	Configurator in use or device busy
8302	Bad block number	8340	.DBF file write failed
8303	No free blocks	8341	More than one .RUN file found
8304	No free database memory	8342	.RUN file not found
8305	Not allowed by block create	834A	Connection Source is not an O/P
8306	In use	834B	Multiple connection to same I/P
8307	Database already exists	834C	Connection Destination not I/P
8308	No spare databases	834D	No free connection resources
8309	Not enough memory	834E	Bad conn. src/dest block/field
8320	Bad library file	834F	Invalid connection destination
8321	Bad template in library	8350	Warmstart switch is disabled
8322	Bad server	8351	No database was running
8323	Cannot create EDB entry	8352	Real-time clock is not running
8324	Bad file version	8353	Root block clock is not running
8325	Bad template spec	8354	Coldstart time was exceeded
8326	Unable to make block remote	8355	Root block is invalid
8327	Bad parent	8356	Too many control loops
8328	Corrupt data in .DBF file	8357	Coldstart switch is disabled
8329	Corrupt block spec	8360	Unsynchronised Block Types
832A	Corrupt block data	8361	DB/Filing system mismatch
832B	Corrupt pool data	8362	Unsynchronised Secondary
832C	No free resources	8363	Operation forbidden whilst CPUs synchronising/changing over
832D	Template not found	8364	Pwr-up data inhibits run
832E	Template resource fault	8365	POST hardware failure
8330	Cannot start	8366	Not fixed function strategy
8331	Cannot stop	8367	Default strategy missing
8332	Empty database		

Table 6.6.2b Database system error codes (83xx)

6.6.2 ERROR MESSAGES (Cont.)

8501	Out of F RAM - DO NOT save file
8502	Out of N RAM - DO NOT save file

Table 6.6.2c Objects system error codes (85xx)

8602	Bad channel number
8603	Bad type code
8611	Bad handle or not hist
8613	File exists
8614	Exceeded global limit
8615	Unexpected end of file
8616	Read error
8617	Write error
8619	Bad filename
861A	Bad timestamp

Table 6.6.2d Trend system error codes (86xx)

8701	Unnamed blocks
8702	Cannot save compounds
8703	No root block
8704	.GRF file write failed
8705	Compounds too deep
8706	Unused GRF block - deleted
8707	Unused GRF connection - deleted
8708	Missing GRF block - added
8709	Missing GRF connection - added
870A	Unknown DBF/GRF block mismatch
870B	Unknown DBF/GRF connect mismatch
870C	DBF/GRF file mismatch - use FIX

Table 6.6.2e Control config error codes (87xx)

8901	Network timeout
8902	Rejected by local node
8903	Rejected by remote node
8904	Not implemented
8905	Not active on local node
8906	Not active on remote node
8907	Transmit failure
8908	Failed to get memory
8909	Decode packet

Table 6.6.2f Network error codes (89xx)

(Continued)

890A	Remote file system busy
890B	Illegal TEATT
890C	Wrong TEATT
890D	NServer is busy
890E	TEATT not owned
890F	Duplicate block
8910	TEATT rejected
8911	Port disabled
8912	No port configuration
8913	Bad network filename
8999	Network node invalid

Table 6.6.2f Network error codes (89xx) (Concluded)

8B01	Object Overload
8B02	Text Overload
8B03	No Matching Step Name
8B04	No Matching Action Name
8B05	Step already Exists
8B06	Action already Exists
8B07	Link already Exists
8B08	Leave a Bigger Gap
8B09	Bad Time Format
8B0A	File Read Error
8B0B	File Write Error
8B0C	File doesn't Exist
8B0D	File not Open
8B0E	Create Action ?
8B0F	No Match with string
8B10	No More Matches
8B11	Match found in Transition
8B12	Match found in Action
8B13	Changed - Are you sure ?
8B14	Link Already Exists
8B15	Illegal Chars in Name
8B16	Action Did Not Compile
8B17	Fatal Memory Overflow - Quit Now!
8B18	Out of memory when compiling
8B19	Root action must be SFC
8B1A	Invalid actions found during compilation
8B1B	Invalid DB name
8B1C	No database loaded
8B1D	Map is invalid

Table 6.6.2g Sequence data base system error codes (8Bxx)

6.6.2 ERROR MESSAGES (Cont.)

8C01	Database not Running
8C02	No Sequences Loaded
8C03	Sequence is being displayed
8C04	Cannot find an SFC_DISP block
8C05	Cannot find Source File
8C06	Sequence Not Loaded

Table 6.6.2h Sequence runtime error codes (8Cxx)

8D01	Syntax Error
8D02	Statement expected
8D03	Assignment expected
8D04	THEN expected
8D05	no ELSE or END_IF
8D06	END_IF expected
8D07	“;”expected
8D08	Bad bracket matching
8D09	Identifier too long
8D0A	Bad identifier
8D0B	Unrecognised symbol
8D0C	Code Buffer Full
8D0D	Expression expected
8D0E	Can’t find this name
8D0F	“String” > 8 chars
8D10	End quotes expected
8D11	Bad Number.

Table 6.6.2i Structured text error codes (8Dxx)

8F01	PCLIN Card not responding
8F02	PCLIN Request failed
8F04	EDB not known or not external
8F07	Unknown EDB
8F0A	Unable to delete ED
8F14	Bad block number
8F15	Template mismatch
8F16	Block failed to attach
8F17	Block failed to detach

Table 6.6.2j PCLIN/PC I/F package error codes (8Fxx)

9001	Invalid PIN
9002	PINs do not match - unchanged
9003	Invalid PIN - reset to 1234
9004	Access denied
9005	Invalid default security info
9006	Invalid DTU A security info
9007	Invalid DTU B security info

Table 6.6.2k T1000 menu system error codes (90xx)

9100	Couldn’t open config file
9101	Section not found
9102	Parameter not found
9103	Argument not found
9104	Config area too small
9105	Config file syntax error
9106	Config header corrupted
9107	Not a number
9108	Out of memory

Table 6.6.2l Configuration files error codes (91xx)

9201	Already registered
9202	To many kernel users
9203	Could not allocate required local storage
9204	Error changing priority
9205	Need to supply an instance name
9206	Failed to get platform info
9207	Platform not known

Table 6.6.2m Kernel error codes (92xx).

9A01	Invalid Second Register
9A02	Not a 32 bit field type
9A03	Invalid Scan Count
9A04	Incorrect Modbus function types
9A05	Invalid register position
9A06	Second register of 32 bit pair
9A07	Invalid register type

Table 6.6.2n MODBUS error codes (94xx)

6.6.2 ERROR MESSAGES (Cont.)

9B01	Illegal unique task id
9B02	Task id already being used
9B03	No more task control blocks
9B04	Out of XEC memory
9B64	Task aborted
9B65	Task timeout

Table 6.6.2p Xec error codes (9Bxx)

A601	Asynchronous I/O in progress
A602	No asynchronous I/O in progress
A603	Not yet implemented
A604	Tx operation complete but not all characters transferred
A605	Rx operation complete, but not all characters received
A606	Event not unique
A607	General CIO error
A608	No asynch. operation fetched
A609	Out of serial lines
A60A	Unable to allocate the requested line
A60B	Failed to submit asynchronous I/O
A60C	Input/output timed out
A60D	Indeterminate error during fetch
A60E	I/O timed out but failed to cancel operation in progress

Table 6.6.2q Asynchronous I/o error codes (A6xx)

Chapter 7 MODBUS

This chapter describes the two implementations of Modbus available on this instrument. The two versions are called Modbus Gateway (section 7.1), and Modbus DCM (Devolved Control Module) (section 7.2). See also Chapter 5, Section 5.6 for Modbus configuration and Chapter 2, Section 2.4.1 for cabling details.

Note: Modbus Gateway does not support Modbus Master mode at this release.

7.1 MODBUS GATEWAY

This section describes the implementation of the Modbus gateway in the Unit controller/ supervisor.

The main topics covered are:

1 Overview of the Modbus gateway (7.1.1)
2 Principles of operation (7.1.2)
3 Using the diagnostic table (7.1.3)
4 Modbus diagnostic function codes (7.1.4)
5 Modbus exception responses (7.1.5)
6 Notes on Modbus/JBUS implementation (7.1.6)
7 Modbus/JBUS interface performance figures (7.1.7)

7.1.1 Overview of the Modbus gateway

The Modbus/JBUS gateway provides a serial interface to the LIN database. By using the techniques of block caching, the gateway can access data in other nodes distributed on the LIN as well as blocks in the local database.

The product operates as a Modbus slave, allowing a PLC or supervisory system configured as a Modbus master to access data in the LIN database.

7.1.1 OVERVIEW (Cont.)

MAIN FEATURES

1. The mapping between the database and the Modbus address space is entirely user-configurable for both digitals and registers.
2. Digitals may be mapped as single bits, 8 bit bytes or 16 bit words.
3. Analogue values map to single 16 bit registers with definable decimal point (Floating-point numbers as well as Integers.)
4. 32-bit values (floating point or long integer) may be mapped to a pair of registers.
5. Configuration can be done via an ANSI standard terminal attached to the gateway local configurator port, by filling in tables using prompts and menus to simplify the task. Refer to Chapter 5, *The control configurator*, for details. The resident configurator checks the validity of the entries during configuration time to minimise errors. Alternatively, the LINtools Modbus configurator running on a PC can be used — see the *LINtools Product Manual*, Part No. HA082377U999.
6. Diagnostic and status registers allow the database to control the Modbus interface.
7. The gateway supports the Modbus RTU (8-bit) transmission mode. ASCII (7-bit) mode is not supported.

FUNCTIONAL DESCRIPTION

The gateway functions by keeping a copy of relevant parameters in Modbus tables which may be individually configured for either digital or register data. This copy is updated from the LIN database by a scanner task running in the gateway.

The gateway supports 16 separate tables, whose size is configurable. The Modbus data area does not reduce the space available for the continuous database.

The default Modbus interface is implemented as RJ45 sockets on the connection module, labelled exp1 (master) and exp2 (slave) as discussed in chapter 2.

7.1.1 OVERVIEW (Cont.)

MODBUS/JBUS FUNCTION CODES SUPPORTED

Table 7.1.1 lists the Modbus function codes supported by the gateway, together with their maximum scan counts, i.e. the maximum number of registers or bits that can be read or written in a single Modbus transmission of this type. For full details on Modbus messages and functions please refer to the Gould *Modicon Modbus Protocol Reference Guide*.

Code	Function
1	Read digital output status
2	Read digital input status
3	Read output registers
4	Read input registers
5	Write single digital output
6	Write single output register
7	Fast read of single byte
8	Diagnostics
	(supports subcodes 0, 1, 2, 3, 4, A, C, D, E, F, 10, 11, 12 — see Table 7.1.4)
15	Write multiple digital outputs
16	Write multiple output registers

Table 7.1.1 Modbus function codes supported

Note that the gateway makes no distinction between inputs and outputs. Thus any register or bit assigned in the gateway can be accessed as both an input or an output as required. This follows the JBUS implementation of Modbus.

7.1.2 Principles of operation

The LIN database groups related data into blocks, such blocks representing an input, an output, a controller etc. The LIN configurators and display packages recognise different types of block, and handle them appropriately. By contrast, the Modbus registers and bits are simply lists of data points. In general there is no predefined structuring of these points into blocks or loops, etc., and most implementations define the allocation of registers differently.

Any gateway involves the mapping of data from the instrument's database to Modbus registers and digitals.

7.1.2 PRINCIPLES OF OPERATION (Cont.)

The gateway has two main purposes:

1. To allow a Modbus master to read from and to write to fields within standard blocks in the LIN system. The slave is passive and can transfer data, with the master only.
2. To allow the master to translate data into a LIN format.

The mapping between registers and blocks is bidirectional; it is up to the master to manage how it interacts with a particular register or point.

Figure 7.1.2 shows a possible mapping of Modbus registers to points in a LIN database. The mapping between the two is configured by the user.

Gaps can be left in the Modbus data areas for future expansion and these gaps can be written to and read from if required, allowing a system of ‘letterboxes’ to be set up that can be exploited by some systems. Data in the gaps does not interact with the standard Process Supervisor database.

The gateway functions by keeping a copy of the relevant parameters in Modbus format. This copy is updated from the slave’s database by a ’scanner’ task running in the gateway. This task regularly examines each value in both database and copy. If it finds that a database value has changed since last time, it transfers the new value to the copy ready to be read by the master at the next poll request. If the scanner finds that a value in the copy has been updated by the master, it writes that value to the database. When a master reads a value across the Modbus, the data is transmitted from the copy.

Note: To maximise communications efficiency, dynamic data should be grouped so that it is available in contiguous table entries for a multi-parameter read.

7.1.2 PRINCIPLES OF OPERATION (Cont.)

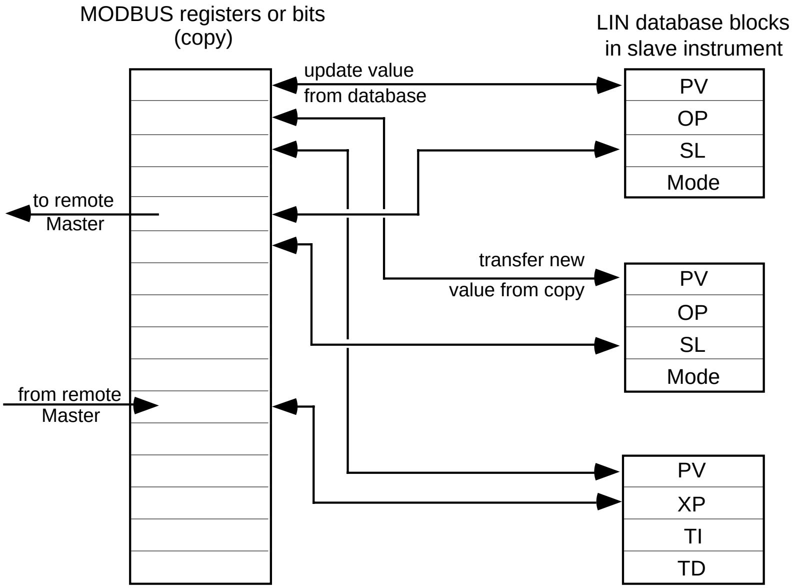


Figure 7.1.2 Slave mode operation

7.1.2 PRINCIPLES OF OPERATION (Cont.)

REFRESH RATES AND TIMING INFORMATION

This section describes the calculations used for determining refresh rates. Refer to section 7.1.7 for typical figures.

Slave mode timing

Response times. Time from end of command until first character of the response:

Minimum	3.5 character periods
Normal	12msec (9600 baud)
Maximum	50msec (9600 baud)

The cycle time depends on a) the slave response time b) the transit time on the serial link (about 14msec plus 1.15msec per byte at 9600 baud) and c) the execution time of the master.

Scan period. The scan period is the time for all the data in the copy areas of all the tables to be updated. This depends on the number of parameters mapped onto the Modbus address space, and on the number of writes made from the master to blocks that are cached within the slave.

Writing to local blocks does not affect this figure, but the data is updated in only one direction each scan, so that if data is written from the copy to the database, it is not updated from the database to the copy until the following scan.

Data is transferred from the Modbus image to the database only if the value has been changed by the master.

The scan period is calculated from the following formula, with a minimum value of 100msec:

$$\text{scan period} = (m \times nt) + (r \times 3.5) + (d \times 3.5) + (w \times 100)\ \text{msec}$$

where m = minimum period (100msec)
nt = number of tables
r = number of registers
d = number of digitals (or sets of digitals)
w = number of writes to remote (cached) blocks per scan period.

Example:

For a system with a table of 16 registers and a table with 16 digital descriptors, but no values connected to cached blocks, the scan period is:

$$(100 \times 2) + (16 \times 3.5) + (16 \times 3.5) + (0 \times 100) = 312\text{msec}$$

7.1.2 PRINCIPLES OF OPERATION (Cont.)

MEMORY USE AND REQUIREMENTS

An area of memory is allocated to map the database parameters to the Modbus address space. This memory is allocated to tables, each table representing a series of consecutive registers or bits in the Modbus address space. The table contains an image of the data in the Modbus address space, and a descriptor for each register, bit, or set of bits mapped onto that address space.

Current configuration sizes and limits

Memory for tables	6000 bytes
Maximum number of tables	16
Minimum entries per table	1
Maximum entries per table (limited by memory usage)	Digital bits: 999. Registers: 2000

Memory requirements for the tables

Overhead	18 bytes per table
Image data — registers	2 bytes per register
Image data — digitals	1 bit per digital (*rounded up - see below*)
Descriptors — registers	6 bytes/entry (*whether connected or not*)
Descriptors — digitals	8 bytes/entry (*whether connected or not*)

7.1.2 PRINCIPLES OF OPERATION (Cont.)

MEMORY USE AND REQUIREMENTS (Cont.)

Digital image data. The storage requirement of digital image data is calculated by converting the total number of bits in the table to 8-bit bytes, then rounding this number of bytes up to the nearest **2-byte boundary**, i.e. the nearest even number. This means that total bit-counts of from 1 to 16 need 2 bytes of storage space, from 17 to 32 bits need 4 bytes, from 33 to 48 bits need 6 bytes, and so on.

The calculation can be done using the following formula, assuming truncation and integer arithmetic:

$$2 \times \mathrm{INT}((\mathit{bitcount} + 15)/16) \text{ bytes.}$$

Examples.

1 A register table with 40 values occupies:

18[overhead] + (40 × 2)[data] + (40 × 6)[descriptors] = 338 bytes.

2 The requirements for a digital table depend on how the data is mapped between the Modbus and the database. The examples below show the two extremes for mapping 64 bits to the database. In case **a** the bits are mapped onto the database in 16-bit units, needing only 4 descriptors. In case **b** each bit is separately mapped to a different point in the database, needing a total of 64 descriptors.

a 18[overhead] + 8[data] + (4 × 8)[descriptors] = 58 bytes.

b 18[overhead] + 8[data] + (64 × 8)[descriptors] = 538 bytes.

DATA CONVERSION

The conversion of data between standard Modbus format and the LIN database format is described here.

Data conversion of digitals

Modbus digital signals can be mapped onto database bitfields, booleans and alarms. The following rules apply to mapping these types into the Modbus address space.

- Bitfields can be mapped individually or as a complete set of 8 or 16 bits onto the Modbus address space.
- Booleans are mapped onto a single bit in the Modbus address space.
- Alarms are mapped onto a single bit in the Modbus address space. A value of ‘1’ for this bit corresponds to the ‘In alarm’ status.

Data conversion of registers

All data types can be mapped onto single registers in the Modbus address space. However, special care should be taken when mapping database values that require more than 16 bits — in particular 32 bit integers and floating point numbers.

7.1.2 PRINCIPLES OF OPERATION (Cont.)

DATA CONVERSION (Cont.)

- **Values requiring up to 16 bits of storage.** Database values that require up to 16 bits of storage (one or two bytes) are mapped directly onto a single register. This includes 8- and 16-bit integers, booleans, alarms and bitfields.

  Long signed 32-bit integers: When these values are transferred from the database to a Modbus register they are truncated, and only the low order 16 bits are written. When the register is being transferred from the Modbus to the database, the value is sign-extended into the high-order 16 bits.

  Long unsigned 32-bit integers: When these values are transferred from the database to a single Modbus register they are truncated, and only the low-order 16 bits are written. When the register is being transferred from the Modbus to the database, the high-order 16 bits are assumed to be zero.

  Floating-point numbers: When these values are transferred from the database to a Modbus register they are scaled according to the decimal point you specify, converted to an integer with rounding, limited to the range –65536 to +65535, and then truncated to 16 bits. This allows applications to work either with signed numbers (–32768 to +32767) or with unsigned numbers (0 to +65535).

  When the register is being transferred from the Modbus to the database, it is treated as a signed number in the range –32768 to +32767, scaled according to the decimal point specified and then written to the database.

- **Values requiring up to 32 bits of storage.** 32-bit fields representing values where precision must be preserved may be connected to a pair of Modbus registers. The two parts are stored in standard PC format in two consecutive registers, of which the first must be at an even address. This method of linking is enabled by entering D (double precision) in the DP field of the first register — see Chapter 5 Section 5.6.3 for details. The scanner task ensures data coherency.

  32-bit totals: Two-register mapping of long integers is used for the Total and Target fields of the TOTAL and TOT_CONN blocks.

7.1.3 Using the diagnostic table

The diagnostic table is a special set of 32 registers containing status and control bits to allow the database to interact with the Modbus drivers. A diagnostic table allows the user to control the Modbus operation, or present diagnostic information to the database. Generally only one diagnostic table needs to be configured per Modbus configuration.

The registers of a diagnostic table are in two distinct sets. The first sixteen — the *internal diagnostic registers* — at default addresses 0 - 15. The last sixteen — the *Modbus table status and control registers* — are at addresses 16 - 31. These two sets of registers are described below.

7.1.3 USING THE DIAGNOSTIC TABLE (Cont.)

INTERNAL DIAGNOSTIC REGISTERS

The first set of registers (with default addresses 0 to 15) are for internal diagnostic use, and are read-only to the user. They present general information on the operation of the Modbus, and their functions are independent of whether the instrument is operating as a master or a slave. Table 7.1.3 lists these registers and their functions.

Offset	Function
0	(*Unused*)
1	(*Unused*)
2	Diagnostic register, bits currently allocated: Bit 5 — Slave in listen-only mode
3	Query data as transmitted by function code 8 sub code 0
4	Input delimiter as transmitted by function code 8 sub code 3
5	(*Unused*)
6	(*Unused*)
7	Count of error messages sent by slave
8	(*Unused*)
9	(*Unused*)
10	(*Unused*)
11	Master polling task: cycle period in 4 msec ticks
12	Scanner task: time to check all tables in 4 msec ticks
13	Scanner task: time used last time scheduled in 4 msec ticks
14	Scanner task: time used for last delay in 4 msec ticks.
15	(*Unused*)

Table 7.1.3 Internal diagnostic registers 0 - 15

MODBUS TABLE STATUS AND CONTROL REGISTERS

The second set of registers (with default addresses 16 to 31) allows individual tables in the configuration to be monitored and controlled. Each register in the diagnostic table is automatically allocated to an entire table in the configuration. Specifically, the diagnostic register at default address 16 is assigned to table 1, the register at address 17 is assigned to table 2, and so on up to table 16.

The functions of this second set of registers depends on whether the system is working in master or slave mode.

7.1.3 USING THE DIAGNOSTIC TABLE (Cont.)

SLAVE MODE DIAGNOSTIC TABLE REGISTERS

The slave mode diagnostic register includes bits that allow monitoring and control of the associated Modbus table by an application running in the database. Figure 7.1.3a shows the allocation of the bits in the register. The values in the register are used in the following way:

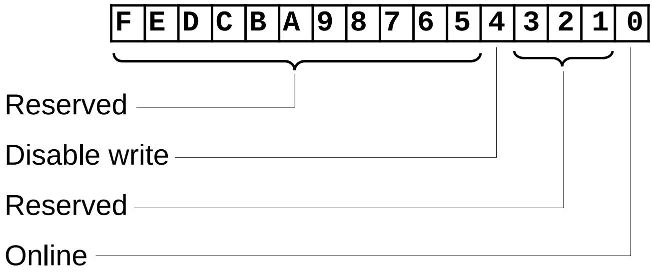


Figure 7.1.3a Slave mode diagnostic registers

- **Disable write** Setting this bit disables writes across the Modbus serial link to the associated table. The slave will return error code 8 (see Table 7.1.5a, *Exception responses*).
- **Online** This bit is set to 1 if the table has been written to or read from in the period defined in *Time out* in the SETUP menu. (See section 5.6.2).

7.1.4 Diagnostic function codes

Table 7.1.4 summarises how the common Modbus diagnostic function codes have been supported by the gateway in slave mode. The diagnostics are accessed via Modbus function code 8.

Diagnostic sub-code	Data sent	Description
0000	xxxx	Echoes the data sent
0001	0000	Restarts
	FF00	Resets the diagnostic counters, and re-enables responses if the slave had been placed in Listen-only mode by sub-code 4.
0002	xxxx	Returns the diagnostic register. (In the current versions, the returned data is always zero.)
0003	ABxx	Changes ASCII delimiter. (This echoes the data sent.)
0004	0000	Forces Listen-only mode. There is NO response to this function.
000A	0000	Resets all counters.
000B		(*Not supported*)
000C	0000	Returns the number of CRC errors detected in messages addressed to this slave.
000D	0000	Returns the number of error messages returned by this slave.
000E	0000	Returns the number of correct messages addressed to this slave.
000F	0000	Returns a count of the number of times the slave has not responded to a valid message (e.g. due to an unsupported function, or a buffering problem in the slave).
0010	0000	Always returns 0.
0011	0000	Always returns 0.
0012	0000	Returns the count of character errors received at the slave, i.e. (overrun + parity + framing) errors.
0013		(*Not supported*)
0014		(*Not supported*)

Table 7.1.4 Modbus diagnostic function codes

7.1.5 Modbus exception responses

SLAVE MODE ERROR CODES

Table 7.1.5 lists the error codes that may be returned in an exception response from a gateway in slave mode.

Code	Name	Meaning (current implementation)
01*	Illegal function	The function is illegal, or not supported within the Modbus gateway.
02*	Illegal data address	The address referenced does not exist in the slave device.
03*	Illegal data value	The value in the data field is invalid.
04	Failure in associated device	
05	Acknowledge	
06	Busy, rejected message	
07	NAK-negative acknowledgement	
08*+	Write error	The data has been write-protected via a bit in the appropriate table diagnostic register.
09+	Zone overlap	
0A+	Header error	
0B+	Slave absent	
0C+	CRC error	
0D+	Transmission blocked	

Table 7.1.5 Exception responses from a slave

**Codes implemented in the controller/supervisor slave mode. +Supplementary codes defined by the JBUS specification.*

7.1.6 Notes on Modbus/JBUS implementation

Although based on the original Modbus specification, different manufacturers' implementations vary slightly in the correspondence between the actual register or bit addresses in a PLC, for example, and the Modbus/JBUS address, i.e. the 'protocol address'. It is this protocol address that is to be configured in the Modbus gateway implementation.

MODBUS (AEG-MODICON)

Read-only ('input') and read/write ('output') registers and bits are assigned to separate tables, each with its own address-offset relative to the Modbus protocol address. Table 7.1.6 summarises this.

Data type	Modbus function codes		PLC address	Protocol address
	Read	Write		
Output bits	01	05, 15	00001+X	X
Input bits	02	—	10001+X	X
Output registers	03	06, 16	40001+X	X
Input registers	04	—	30001+X	X

Table 7.1.6 PLC address offsets for different data types

It is the Modbus function code that determines the value of the offset required, and therefore whether a given Modbus protocol address is directed at an input or output, in a bit or register table.

JBUS

In the JBUS implementation there is a direct correspondence between a register or bit address and the Modbus protocol address, and no distinction is made between input and output (or indeed internal) PLC registers. Thus Modbus function codes 01 and 02 are treated identically, as are codes 03 and 04. All PLC data thus conforms to a single address range.

OTHER PRODUCTS

Other manufacturers' 'gateway' implementations conform to the MODICON principle of separate tables for different types of data exchange, but the correspondence between PLC base address and Modbus protocol address is *user-configurable*.

7.1.7 Modbus/JBUS interface performance figures

UPDATE PERIOD

In general the update period between the database in a master device and the database in a slave device, for continuously polled values, is the sum of the following times:

1. Scanning period between MODBUS table and database in the master
2. Cycle time of serial link communications
3. Scanning period between MODBUS table and database in the slave

SERIAL LINK CYCLE TIME

The cycle time of the serial link communications is itself the sum of the following:

1. Response time at master
2. Transit time (request + response) over serial link
3. Response time at slave

SCANNING PERIOD & RESPONSE TIME

For the Unit controller/supervisor the scanning period and response time depend on the number of 16-bit words to be scanned and may be expressed approximately, for both master and slave versions, as:

Scanning period (msec) $= 200 + 3.5(r + d)$
Response time (msec) $= 10 + 0.08n$

where r = number of register table entries
d = number of digital table entries (1-bit, 8-bit, or 16-bit)
n = number of 16-bit words (registers and bits expressed in multiples of 16)

TRANSIT TIME ON SERIAL LINK

The Transit time over the serial link depends on the baud rate and the volume of information. At 9600 baud (no parity, 2 stop bits) this may be calculated as follows:

Transit time (msec) $= 14 + 2.3n$

PROCESS SUPERVISOR CONTRIBUTION

As a guideline, Table 7.1.7 indicates the contribution of the Process supervisor as measured on an 8MHz unit. The contribution of the other device must be added to these times.

Number of 16-bit words:	16	64	125
Table/database scan time (msec):	256	432	648
Serial link cycle time (msec):	63	177	322
Total minimum update time (msec):	319	609	970

Table 7.1.7 Unit controller/supervisor contribution to timings

This compares with a database execution period typically in the order of 100 to 400 msec depending on complexity.

Write transactions add to these figures according to volume of data. Note that data transfers from a MODBUS table to a cached block imply a write transaction across the LIN, which adds to the overall update time.

7.2 MODBUS DCM

Note: The process supervisor acts as a master when running modbus DCM

7.2.1 Introduction

A Devolved Control Module (DCM) is configured for each item to be accessed via the Modbus link. In addition, an Instrument Block is available for each model of I/O unit produced by the Process Supervisor manufacturer. This contains various instrument and Modbus parameters along with instrument and I/O failure and status indications.

When all blocks have been configured correctly, and are resident in the data base along with the *_system.opt* file and any relevant Universal Map for Modbus (.uym) files*, then the Process Supervisor will start communicating with the I/O unit as soon as it is initialised, without the need to set up mapping tables as is required by the Gateway version.

For ‘third party’ instruments, a .uym* file must be created for each DCM.

*See section 7.2.4 for .uym file description

The DCMs themselves are fully described in Chapter 15 of the LIN Blocks Reference Manual (HA082375U003), but brief details of what is available is given in the list below.

7.2.2 DCMs available

The modules available at time of print are:

Loop Blocks:

D2X_LOOP	Access PID Control loop in the I/O unit.
D2X_TUNE	Tune PID Loop in the I/O unit.
D25_LOOP	Access PID Control loop in the I/O unit.
D25_TUNE	Tune PID Loop in the I/O unit.

Ramp Blocks

D25_RAMP	Ramp remote setpoint

I/O Module Blocks

D25_MOD	Access single I/O Module
D25_AI2	Access two-channel analogue input module
D25_AI3	Access three-channel analogue input module
D25_AO2	Access two-channel analogue output module
D25_AO4	Access four-channel analogue output module
D25_DI4	Access four-channel digital input module
D25_DI8	Access eight-channel digital input module
D25_DO4	Access four-channel digital output module

7.2.2 DCMs AVAILABLE (Cont.)

I/O channel blocks

D25_AI	Access single analogue input channel
D25_AICH	Access single analogue input channel
D25_AO	Access single analogue output channel
D25_AOCH	Access single analogue output channel
D25_DI	Access single digital input channel
D25_DICH	Access single digital input channel
D25_DO	Access single digital output channel
D25_DOCH	Access single digital output channel

User Wiring Blocks

D25_R_CV	Access up to 8 user wiring calculated values (real) in the I/O unit.
D25_B_CV	Access up to 8 user wiring calculated values (boolean) in the I/O unit.
D25_R_UV	Access the 8 user values (real) in the I/O unit.

User Alarm Blocks

D25_UALM	Access the alarms in the I/O unit.

Parameter blocks

DCM_B8	Access up to 8 Boolean parameters in the I/O unit.
DCM_D8	Access up to 8 double precision integer (32-bit) parameters in the I/O unit.
DCM_I8	Access up to 8 signed integer (16-bit) parameters in the I/O unit.
DCM_R8	Access up to 8 signed real number (32-bit) parameters in the I/O unit.
DCM_S8	Access up to 8 signed 8-bit integer values in the I/O unit.
DCM_T8	32-bit I/O unit time period (msec.)
DCM_UI8	Access up to 8 unsigned integer (16-bit) parameters in the I/O unit.
DCM_US8	Access up to 8 unsigned 8-bit integer values in the I/O unit.
DCM_W8	Access up to 8 ABCD hex word (16-bit) parameters in the I/O unit.
DCM_Y8	Access up to 8 AB hex byte (8-bit) parameters in the I/O unit.

Instrument/Diagnostic blocks

D2000	Provides overall view of the I/O unit comms parameters/status indication etc.
D2500	Provides overall view of the I/O unit comms parameters/status indication etc.

7.2.3 _System.opt file

This is a text file, created on a text editor and loaded into the machine's file system once, at power-up. Its format is as follows (example only).

```
COM1,RS232,Termcfg,0,9600,Even,7,1,0,None<Return>
COM2,RS422,Modbus-M,0,9600,None,8,1,2000,None,Five<Return>
COM3,RS422,Modbus-S,1,9600,None,8,1,0,None,Five<Return>
```

Field 1: Port

COM1	'Config' port on Master Processor module(s)
COM2	Relevant 'exp1' port on Connect Module
COM3	Relevant 'exp2' port on Connect Module

Field 2: Communications Standard

RS232	Used at COM1 only. Com1 can use only RS232
RS422	Used at COM2 and/or COM3 port
RS485	Used at COM2 and/or COM3 port

Field 3: Protocol

Termcfg	Used only at processor module Config ports
Modbus-M	Modbus Master
Modbus-S	Modbus Slave

Field 4: Address

Termcfg must always be 0
Modbus master must always be 0
Modbus slave may never be zero but must be between 1 and 247

Field 5: Baud Rate

Enter a suitable Baud Rate. Must match other units on the link

Field 6: Parity

Set to 'None' or 'Even'. Must match other units on the link. Must be 'even' for termcfg.

Field 7: Data bits

7 for configuration terminal, 8 for Modbus instruments. Must match other units on the link

Field 8: Stop bits

Termcfg = 1, Modbus = 1 or 2. Must match other units on the link

Field 9: Timeout in msec. (Time before it is assumed that an error has occurred)

Termcfg = 0, Modbus = 50 to 5000 as required.

Field 10: Transparent Modbus Access (TMA)

Always 'none' (not yet implemented)

Field 11: Three/five wire circuitry

Not applicable to Termcfg.
EIA422 is five-wire only.
EIA485 can be three or five wire.

7.2.4 The .uym file

A .uym file is required for every* DCM which is to communicate with 'third party' equipment, in order that the Process Supervisor may know the address at which a particular point (channel value, alarm threshold value etc.) is to be found. This information has be obtained from the third party documentation.

*Note:
An unlimited number of blocks may use the same .uym file, provided only that their register usage (i.e. the range of addresses) in the target instrument is identical.

The .uym file (fully described in chapter 15 of the LIN Blocks Reference Manual HA083375U003), is a text file, created on a text editor and loaded into the database. The format is as follows:

Field,Register,<u>Type,Function codes</u>

Where:

Underlined items do not need to be included if the defaults are acceptable

Field is the name of the LIN database block being mapped, and

Register is the required Modbus register of the point being accessed.

This field can be a simple decimal number or it can be of the form:

Constant1[Constant2(Field name±Constant3)]*

Where

Constants 1,2 and 3 are a simple decimal numbers,

Field name is any field name in the block which has a 16-bit integer value. A sample expression might be:

200 + 10[(Slot_No +Chan_No)-1]

In which Constants 1, 2 and 3 are 200, 10 and 1 respectively, and the field name is 'Slot_No + Chan_No'.

Type is number type. This field needs to be entered only if the default (Unsigned Integer (UINT)) is not the correct type. See table 7.2.4 for number type entries.

Function codes

Modbus function codes. This field needs to be entered only if the default (3, 4) (read registers) is not acceptable. See table 7.1.1 above for a list of Function codes supported by the Process Supervisor.

7.2.4 THE .UYM FILE (Cont.)

Number type entry	Number type definition
BOOL	Value of 0 or 1 in the LSB
DINT	Signed 32-bit register
DINT_X*	Signed 32-bit register (reverse word order)
INT	Signed 16-bit integer
REAL	32-bit IEEE floating point value in two registers
REAL_X*	32-bit IEEE floating point value in two registers (reverse word order)
REAL_24_STAT8	Real + status
REAL_24_STAT8_X	Real + status (reverse word order)
SINT	Signed 8-bit integer
SREAL_p1	16-bit number in units of 0.1
SREAL_p2	16-bit number in units of 0.01
SREAL_p3	16-bit number in units of 0.001
STIME_dh	16-bit duration in deci hour (0.1 hour) units
STIME_dm	16-bit duration in deciminute (0.1 min) units
STIME_ds	16-bit duration in decisecond (0.1second) units
TIME	Signed 32-bit duration in millisecs
UDINT	Unsigned 32-bit integer
UDINT_X*	Unsigned 32-bit integer (reverse word order)
UINT	Unsigned 16-bit integer
UINT_PK	Packed, unsigned, 16-bit integer
UINT_PK_S	Packed, unsigned, 16-bit integer + status
USINT	Unsigned 8-bit integer
* _X versions must be used when communicating with LIN instruments	

Table 7.2.4 Number types supported by the Process Supervisor

EXAMPLE

To read an analogue input value from channel 17 of a chart recorder.

For a particular chart recorder, the Communications parameters, have been set up in the Configuration: Comms menus, as follows (to match the Process Supervisor settings):

Protocol:	MODBUS
Baud Rate:	9600
Parity:	Even
Data bits:	8 (fixed for MODBUS protocol)
Stop bits:	1
H/W handshake:	Off
Address:	4

From the recorder documentation the analogue input channels are accessed using code 03 and are addressed contiguously, starting with channel 1 at decimal address 0. Thus to read its input value, code 03 and address 16 are required. Thus the .UYM file should contain the following: *MV,16,UINT,03*

7.2.4 THE .UYM FILE (Cont.)

Scaling

The recorder documentation also states that the value (PV) is returned as a 16-bit hex number in the range 0000 (Channel Low range value) to FFFF (Channel High range value), and the calculation:

$$Scaled\ value\ =\ \left((High\ range\ -\ low\ range)\ \times \frac{PV}{FFFF} \right) + low\ range$$

has to be carried out to find the actual scaled value. The recorder Channel Configuration must be accessed to determine the High and Low range values.

Example

High range = 90% for 4V input signal

Low range = 10% for 1V input signal

Current PV = 2.5V (7FFF)

The scaled value is {(90 - 10)% x 7FFF/FFFF} + 10% = 50%

COMMENTING

A comment can be attached to the end of one or more lines in the form:

, , “Comment”

The maximum number of characters for the .UYM file is 60 characters, including delimeters. The comment text string can contain a maximum of (60 minus rest of line) characters.

Thus the sample .UYM file could become:

```
MV,16,UINT,03,,”Recorder 4, channel 17”
```

This page is deliberately left blank

Chapter 8 PROFIBUS

8.1 INTRODUCTION

PROFIBUS DP is an industry standard open network used to interconnect instrumentation and control devices in, for example, a manufacturing or processing plant. It is often used to allow a central Programmable Logic Controller (PLC) or PC based control system to use external 'slave' devices for input/output (I/O) or specialised functions, thus reducing the processing load on the controlling unit so that its other functions can be carried out more efficiently using less memory.

This implementation of the PROFIBUS network uses a high speed version of the EIA485 standard to permit transmission rates of up to 12Mbits/second between the host and multiple 'Stations' otherwise called 'nodes' either within a single section of network or, with EIA485 repeaters, in several separate sections of network. Acceptable node addresses are 3 to 126.

Note: For versions prior to 2/2, the maximum number of Profibus nodes is 16.

It is not within the scope of this document to describe the PROFIBUS standard in detail; more detailed information can be found by reference to the profibus web site:

'http://www.profibus.com'.

8.2 PARAMETER SETUP (_System.opt file)

The _System.opt file is a text file, created on a text editor and loaded into the machine's file system once, at power-up. Its format is as follows (example only).

```
COM1,RS232,Termcfg,0,9600,Even,7,1,0,None
COM2,RS422,Modbus-M,0,9600,None,8,1,2000,None,Five
COM3,RS422,Modbus-S,1,9600,None,8,1,0,None,Five
PROFIBUS1,RS485,ProfibusDpv1-M,2,12000000,,,,250,None,Five
```

Field 1: Port

COM1	'Config' port on Master Processor module(s)
COM2	Relevant 'exp1' port on Connect Module
COM3	Relevant 'exp2' port on Connect Module
PROFIBUS1	Profibus port

Field 2: Communications Standard

RS232	Used at COM1 only. Com1 can use only RS232
RS422	Used at COM2 and/or COM3 port
RS485	Used at COM2 and/or COM3 port

Field 3: Protocol

Termcfg	Used only at processor module Config ports
Modbus-M(S)	Modbus Master (Slave)
ProfibusDpv1-M	Profibus Master

Field 4: Address

Termcfg must always be 0
Modbus Master must always be 0
Modbus Slave may never be zero but must be between 1 and 247 inclusive
Profibus Master. It is recommended that address 2 be allocated to the active master unit. Address 1 will then be automatically allocated to the passive master unit.

Field 5: Baud Rate

Enter a suitable Baud Rate. For Modbus, this must match other units on the link. (Profibus is auto-Baud setting.)

Field 6: Parity

Set to 'None' or 'Even'. Must match other units on the link. Must be 'even' for termcfg.

Field 7: Data bits

Set to seven for configuration terminal, to eight for Modbus or Profibus instruments. Must match other units on the link

Field 8: Stop bits

Termcfg = 1, Modbus/Profibus = 1 or 2. Must match other units on the link

Field 9: Timeout in msec. (Time before it is assumed that an error has occurred)

Termcfg = 0, Modbus/Profibus = as required between 50 and 5000.

Field 10: Transparent Modbus Access (TMA)

Always 'none' (not yet implemented)

Field 11: Three/five wire circuitry

Not applicable to Termcfg.
EIA422 is five-wire only.
EIA485 can be three or five wire

8.3 DEVOLVED CONTROL MODULES

8.3.1 Introduction

A Devolved Control Module (DCM) is configured for each item to be accessed via the link. In addition, an Instrument Block is available for each model of I/O unit produced by the Process Supervisor manufacturer. This contains various instrument and Profibus parameters along with instrument and I/O failure and status indications.

When all blocks have been configured correctly, and are resident in the data base along with the *_system.opt* file and any relevant Universal Map for Profibus (.uyp) files*, then the Process Supervisor will start communicating with the I/O unit as soon as it is initialised.

For ‘third party’ instruments, a .uyp* file must be created for each DCM.

*See section 8.4 for .uyp file description

The DCMs themselves are fully described in Chapter 15 of the LIN Blocks Reference Manual (HA082375U003), but brief details of what is available is given in the list below.

8.3.2 DCMs available

The modules available at time of print are:

Loop Blocks:

D2X_LOOP	Access PID Control loop in the I/O unit.
D2X_TUNE	Tune PID Loop in the I/O unit.
D25_LOOP	Access PID Control loop in the I/O unit.
D25_TUNE	Tune PID Loop in the I/O unit.

Ramp Blocks

D25_RAMP	Ramp remote setpoint

I/O Module Blocks

D25_MOD	Access single I/O Module
D25_AI2	Access two-channel analogue input module
D25_AI3	Access three-channel analogue input module
D25_AO2	Access two-channel analogue output module
D25_AO4	Access four-channel analogue output module
D25_DI4	Access four-channel digital input module
D25_DI8	Access eight-channel digital input module
D25_DO4	Access four-channel digital output module

8.3.2 DCMs AVAILABLE (Cont.)

I/O channel blocks*

D25_AI	Access single analogue input channel
D25_AICH	Access single analogue input channel
D25_AO	Access single analogue output channel
D25_AOCH	Access single analogue output channel
D25_DI	Access single digital input channel
D25_DICH	Access single digital input channel
D25_DO	Access single digital output channel
D25_DOCH	Access single digital output channel

User Wiring Blocks

D25_R_CV	Access up to 8 user wiring calculated values (real) in the I/O unit.
D25_B_CV	Access up to 8 user wiring calculated values (boolean) in the I/O unit.
D25_R_UV	Access the 8 user values (real) in the I/O unit.

User Alarm Blocks

D25_UALM	Access the alarms in the I/O unit.

Parameter blocks

DCM_B8	Access up to 8 Boolean parameters in the I/O unit.
DCM_D8	Access up to 8 double precision integer (32-bit) parameters in the I/O unit.
DCM_I8	Access up to 8 signed integer (16-bit) parameters in the I/O unit.
DCM_R8	Access up to 8 signed real number (32-bit) parameters in the I/O unit.
DCM_S8	Access up to 8 signed 8-bit integer values in the I/O unit.
DCM_T8	32-bit I/O unit time period (msec.)
DCM_UI8	Access up to 8 unsigned integer (16-bit) parameters in the I/O unit.
DCM_US8	Access up to 8 unsigned 8-bit integer values in the I/O unit.
DCM_W8	Access up to 8 ABCD hex word (16-bit) parameters in the I/O unit.
DCM_Y8	Access up to 8 AB hex byte (8-bit) parameters in the I/O unit.

Instrument/Diagnostic blocks

D2000	Provides overall view of the I/O unit comms parameters/status indication etc.
D2500	Provides overall view of the I/O unit comms parameters/status indication etc.

*Note: Refer to the Lin blocks reference manual for details of the difference between 'CH' and non 'CH' versions.

Note: In each 16-way I/O base, a maximum of 12 eight-way digital input modules (DI8s) may be fitted.

8.4 THE .uyp FILE

A .uyp file is required for every* DCM (section 8.3) which is to communicate with ‘third party’ equipment, in order that the Process Supervisor may know the address at which a particular point (channel value, alarm threshold value etc.) is to be found. This information has to be obtained from the third party documentation.

---

*Note:
An unlimited number of blocks may use the same .uyp file, provided only that their register usage (i.e. the range of addresses) in the target instrument is identical.

The .uyp file, is a text file, created on a text editor and loaded into the database. The format is as follows:

Field,Address,<u>Type,”Operations”</u>

Where:

Underlined items do not need to be included if the defaults are acceptable

Field is the name of the LIN database block being mapped, and

Address is the required register of the point being accessed.

This field can be a simple decimal number or it can be of the form:

Constant1[Constant2(Field name±Constant3)]*

Where

Constants 1,2 and 3 are a simple decimal numbers,

Field name is any field name in the block which has a 16-bit integer value. A sample expression might be:

200 + 10[(Slot_No +Chan_No)-1]

In which Constants 1, 2 and 3 are 200, 10 and 1 respectively, and the field name is ‘Slot_No + Chan_No’.

Type is number type. This field needs to be entered only if the default (Unsigned Integer (UINT)) is not the correct type. See table 8.4.4 for number type entries.

Operations One or more of:

RC, WC, RA, WA, where R = read, W = write, C = cyclic, A = acyclic
Default = RC (read cyclic)

8.4 THE .uyp FILE (Cont.)

Number type entry	Number type definition
BOOL	Value of 0 or 1 in the LSB
DINT	Signed 32-bit register
DINT_X*	Signed 32-bit register (reverse word order)
INT	Signed 16-bit integer
REAL	32-bit IEEE floating point value in two registers
REAL_X*	32-bit IEEE floating point value in two registers (reverse word order)
REAL_24_STAT8	Real + status
REAL_24_STAT8_X	Real + status (reverse word order)
SINT	Signed 8-bit integer
SREAL_p1	16-bit number in units of 0.1
SREAL_p2	16-bit number in units of 0.01
SREAL_p3	16-bit number in units of 0.001
STIME_dh	16-bit duration in deci hour (0.1 hour) units
STIME_dm	16-bit duration in deciminute (0.1 min) units
STIME_ds	16-bit duration in decisecond (0.1second) units
TIME	Signed 32-bit duration in millisecs
UDINT	Unsigned 32-bit integer
UDINT_X*	Unsigned 32-bit integer (reverse word order)
UINT	Unsigned 16-bit integer
UINT_PK	Packed, unsigned, 16-bit integer
UINT_PK_S	Packed, unsigned, 16-bit integer + status
USINT	Unsigned 8-bit integer
* _X versions must be used when communicating with LIN instruments	

Table 8.4 Number types supported by the Process Supervisor

COMMENTING

A comment can be attached to the end of one or more lines in the form:

, , “Comment”

The maximum number of characters for the .UYP line is 60 characters, including delimeters. The comment text string can contain a maximum of (60 minus rest of line) characters.

Thus a sample .UYP file might be:

```
MV,16,UINT,"RC,WC",,"Recorder 4, channel 17"
```

8.5 INSTALLATION

8.5.1 Guidelines

GENERAL

1. Profibus specified terminators (390Ω/220Ω/390Ω for Type A; 390Ω/150Ω/390Ω for Type B) must be used at each end of the link (resistors 0.25 W min.). Category 5 terminators (available from the Process Supervisor manufacturer under part number CI026529) should be used with 100Ω Category 5 cable.
2. Cable types within a segment should not be mixed. Wherever possible use cable which complies with Profibus Standard EN50170.
3. Keep stub lengths to a minimum. The total capacitance of all stubs in a network must not exceed 25pF (12Mbit/sec); 200pF (1.5Mbit/sec) or 600pF (500kbit/sec).
4. Always use the lowest data rate consistent with acceptable performance.
5. All site installations must comply with the Profibus Installation Guidelines for Profibus-DP/FMS, available from the local Profibus National Organisation under part number 2.112.

IN CUBICLE

1. 24awg, solid core, low loss Category 5 FTP cable with mutual capacitance <60pF/m. should be used. See section 10.2.2 for suitable cable. The minimum acceptable bend radius is 6mm. Cable idents, cable supports/cleats etc. must not be overtightened, or in anyway distort the cable, as the crushing of the outer jacket can affect the characteristic impedance of the cable and generate refleced signals.
2. The RJ45 plugs must be designed to accept solid core cable. (Suitable connectors are available from the Process Supervisor manufacturer under part number CI250449)
3. The maximum total number of nodes is 18.
4. Total cable runs must not exceed 30 metres (12Mbit/sec) or 70 metres (1.5Mbit/sec) without a repeater. Runs between nodes must be kept as short as possible.
5. When terminating the cable, unshielded lengths must not exceed 30 mm. in length.
6. At each end of each segment, a terminator (part number CI026529) must be used.
7. Test nodes should be used sparingly and never more than one per segment.

8.5.2 Cubicle wiring

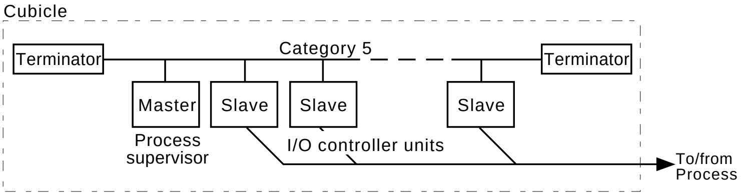
The Profibus link is terminated at the connection module, using an RJ45 connector as described in section 2.4.1 (figure 2.4.1b for pin-out).

The Process Supervisor Profibus Master was designed for use with up to 16 I/O controller units all located within one or more cubicles (figures 8.5.2a and 8.5.2b). For ease of connection, RJ45 connectors and Category 5 cable are used within the cubicle.

For interface with external Profibus networks, using 9-way D-type connectors, a suitable repeater must be used (figure 8.5.2c). Such external profibus networks are normally implemented using Type A cable, described in section 8.5.3 below, and are subject to the Profibus Standard guidelines available from the local Profibus National Organisation (PNO).

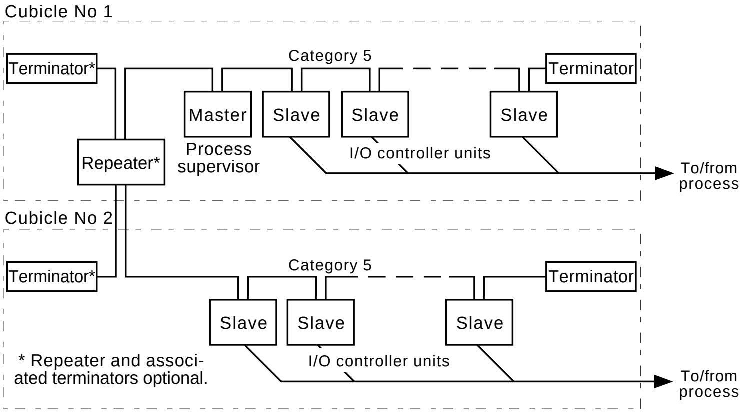
8.5.2 CUBICLE WIRING (Cont.)

The following figures relate to a Profibus Network operating at 12Mbits/second.



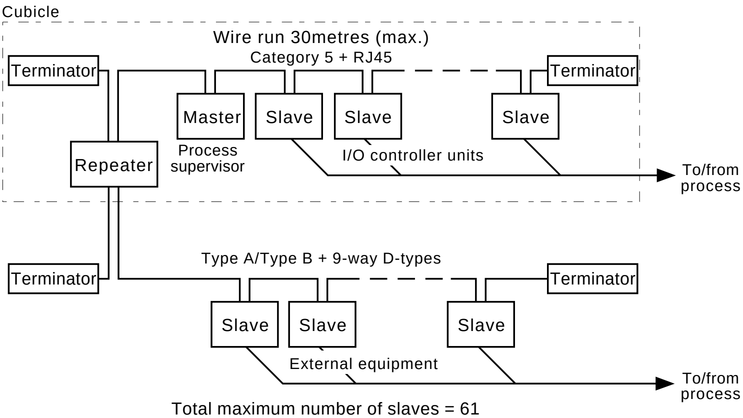
Maximum cable run = 30 metres. Maximum number of slaves = 31

Figure 8.5.2a single cubicle wiring



Wire run 30 metres (max.) per network. Total maximum number of slaves = 61

Figure 8.5.2b Dual cubicle wiring



Total maximum number of slaves = 61
Maximum cable run depends on installation and Baud rate.

Figure 8.5.2c Single cubicle wiring with external equipment

8.5.3 External profibus networks

This section gives general guidelines only. For full details of Profibus installation, refer to Profibus Guidelines available from the local Profibus National Organisation under part number 2.142.

The cable details given below refer to standard 150Ω Profibus cable. Terminators must be Profibus approved for the cable type.

EARTHING THE SHIELD

The PROFIBUS standard suggests that both ends of the transmission line be connected to safety earth. If such a course is followed, care must be taken to ensure that differences in local earth potential do not allow circulating currents to flow, as these can not only induce large common mode signals in the data lines, resulting in communications failure, but can also produce potentially dangerous heating in the cable. Where doubt exists, it is recommended that the shield be earthed at only one point in each section of the network or 100Ω current limiting resustors be fitted in accordance with EIA485-A.

NETWORK WIRING

There are two distinct ways of wiring a network, known as ‘Linear topology’ and ‘Tree topology’. In a linear network (figure 8.5.3a), the maximum number of repeaters is three, giving a total number of stations of 122. In theory the tree set-up (figure 8.5.3b) can have more stations, but the Profibus protocol limits the number of stations to 127 (addresses 0 to 126).

For the process supervisor, the maximum number of nodes per segment is set to 32, where masters and repeaters represent one node each.

CABLE TYPE

Table 8.5.3a below gives the specification for a suitable Type A cable.

Impedance	135 to 165 ohms at 3 to 20 MHz
Resistance	<110 Ohms/km
Cable capacitance	<30 pF/metre
Core diameter	0.34mm^2 max. (22 awg)
Cable type	Shielded twisted pair
Shielding	Cu shielding braid or shielding foil

Table 8.5.3a Cable specification

8.5.3 EXTERNAL PROFIBUS NETWORKS (Cont.)

MAXIMUM TRANSMISSION RATE

The maximum transmission speed depends on the length of the cable run including 'stub' (distance from the bus to a station) lengths. Guaranteed minimum values for Type A cable (assuming maximum attenuation) are given in table 8.5.3b, below. See also section 8.5.1 for cabling guidelines.

Max. line length/segment (metres)	100	200	400	1000	1200
Baud rate (kbit/sec)	12,000	1,500	500	187.5	93.75

Table 8.5.3b Maximum Baud rate versus line length

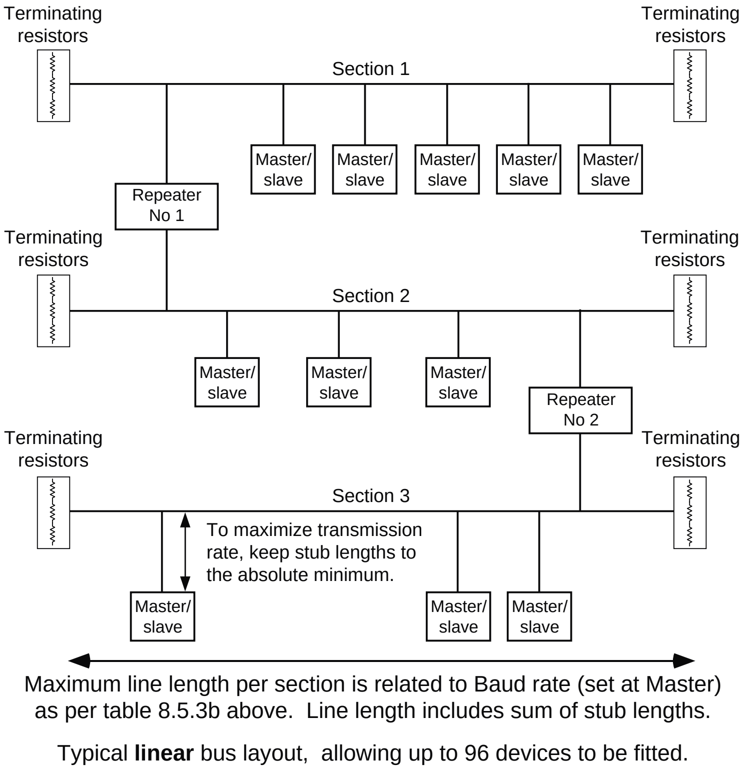


Figure 8.5.3a Typical linear bus layout

8.5.3 EXTERNAL PROFIBUS NETWORKS (Cont.)

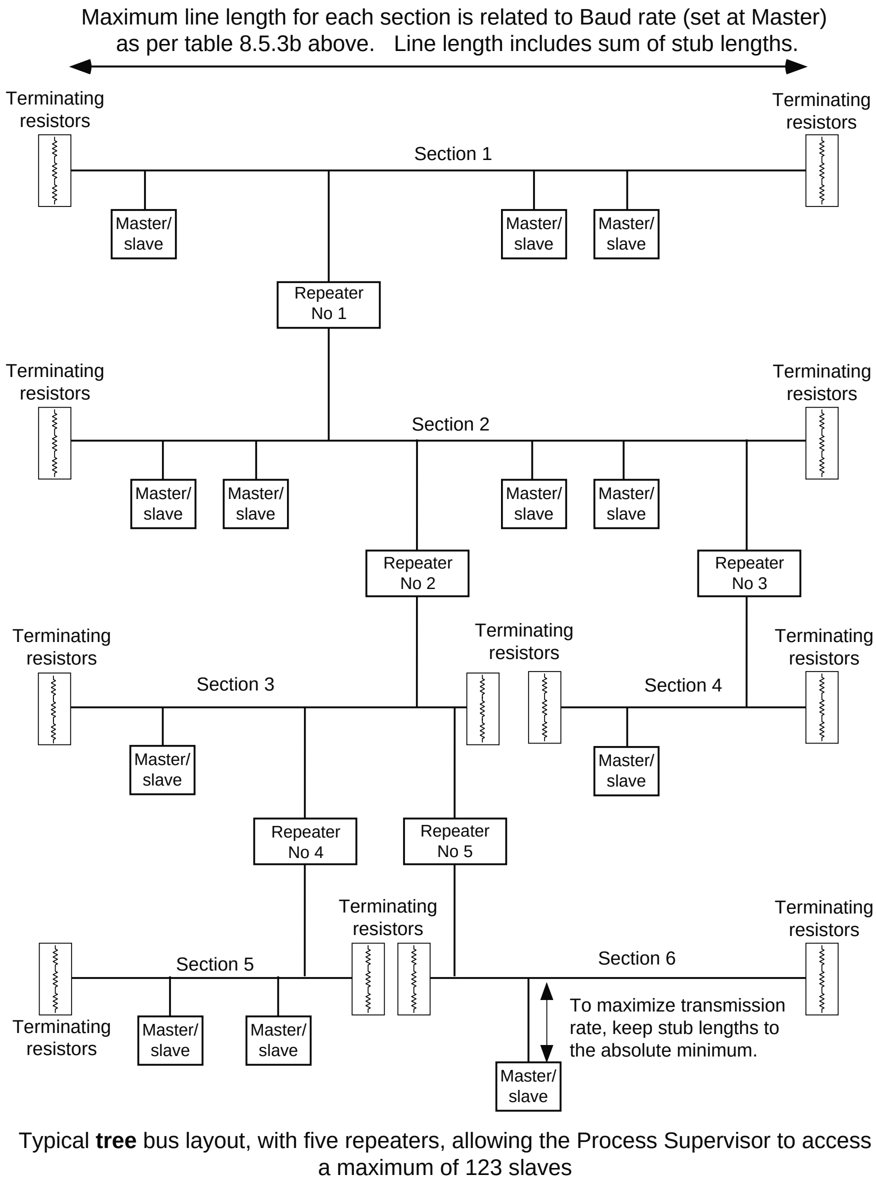


Figure 8.5.3b Typical tree bus layout

8.5.4 Adding the unit to the network

Once the unit has been physically connected, the _system.op and UYP files and any .gsd files for third party equipment can be transferred, and the database started.

8.5.5 Redundant (duplex) mode

The dual processor modules of the process supervisor allow redundant processing to be carried out, so a failure in one processor will not affect control of the I/O units. Redundant wiring to the I/O units is not possible, however, and any break in the daisy chain link will cause all units beyond the break to lose processor control, and may under some circumstances stop the communications link altogether due to reflections from the unterminated break point. Figure 8.5.5, below, shows a typical wiring layout for redundant processing.

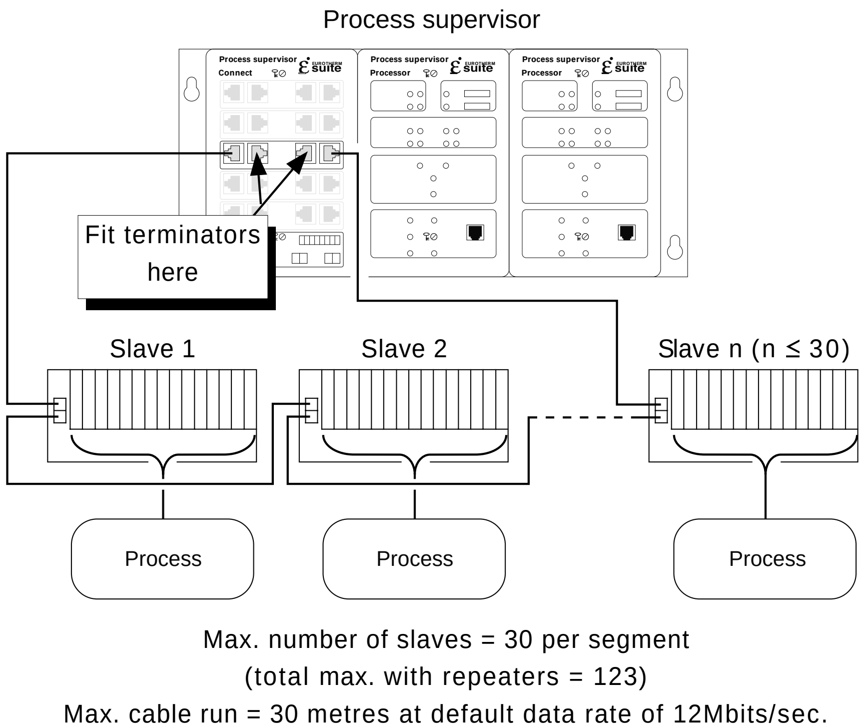


Figure 8.5.5 Redundant Profibus processing typical wiring

8.6 TROUBLESHOOTING

WARNING

Fault finding may affect the network and control system. Ensure that no damage to personnel or equipment can be caused by any fault finding activity.

NO COMMUNICATIONS

1. Check the wiring
2. Check the node address, ensuring that it is unique.
3. Ensure that the network has been correctly configured and that the configuration has been correctly transferred to the master.
4. Verify that the GSD file being used is correct, by loading it into the GSD file configurator to check the format..
5. Ensure that the maximum line length of transmission line has not been exceeded for the Baud rate in use (Table 8.5.3b above).
6. Ensure that the final node on the transmission line (no matter what type of instrument it is) is terminated correctly using a terminator unit. Note that some equipment has built-in pull up and pull down resistors which in some cases can be switched in and out of circuit. Such resistors must be removed or switched out of circuit for all but the instruments at each end of the line.
7. Replace any faulty item(s) and re-test.

INTERMITTENT FAILURE TO COMMUNICATE

This fault is shown by the diagnostic status changing, without alarms being generated in the instrument. The following section details diagnostics information.

1. Check wiring as for ‘No Communications’ above. Pay particular attention to the integrity of the screening and termination
2. Check the number of words in the data exchange against the maximum number the master can support.
3. Ensure that the maximum line length of transmission line has not been exceeded for the Baud rate in use (Table 8.5.3b above).
4. Ensure that the final node on the transmission line (no matter what type of instrument it is) is terminated correctly, and that only the first and final nodes are so terminated. Note that some equipment has built-in pull up and pull down resistors which in some cases can be switched in and out of circuit. Such resistors must be removed or switched out of circuit for all but the instruments at each end of the line.
5. Replace any faulty item(s) and re-test.

8.6 TROUBLESHOOTING (Cont.)

DATA FORMAT OR PARAMETER DATA SEEMS INCORRECT

Verify that the GSD file is correct for the given application by loading it into a GSD file configurator program.

COMMUNICATION SEEMS SLOW

The normal cyclic exchange of data should be very fast. Should so much data be requiring transfer that it cannot be fitted into the cycle rate, then it will be sent acycliclly, and this results in a much slower transfer rate of all data.

To maximise efficiency, module DCMs should be used wherever possible instead of individual channel DCMs. Module DCMs provide a process variable's value and alarm status only. Refer to the Lin Blocks reference manual for further details.

The diagnostic block amc_diag provides information showing any communications 'overflow'.

The diagnostic block pmc_diag provides profibus diagnostics - see the Linblocks reference manual (HA082375U003) for more details.

8.7 Global commands

Freeze and Sync from a PROFIBUS master have no effect

8.8. OPERATION

PROFIBUS DP performs a cyclical scan of the network devices, during which input and output data for each node is exchanged.

Values from each node (input data) are read by the Profibus controller, which then runs its control program, and generates a set of values (output data) to be transmitted to the nodes. This process is called an ‘I/O data exchange’. This process is repeated continuously, to give a cyclical I/O data exchange.

Examples of input data are

a. A set of digital readings for a digital input

b. The measured temperature and alarm status from a PID controller.

Examples of output data are:

a. A setpoint to be sent to a PID controller

The I/O data exchange can be repeated continuously, can be synchronised at given times, or can be repeated at a pre-defined interval, which is asynchronous with the controller. Each node is normally assigned a group of PLC I/O registers, or a single function block, so that the controlling program can deal with each node’s data as though the node is an internal device, without having to be concerned about timing problems. This mapping of node to register or function block is carried out during network configuration, which is usually carried out using a PC based program.

8.8.1 I/O data transfer limits.

The PROFIBUS DP standard allows up to 244 bytes of data, or 116 discrete data items to be transferred in each direction, during each I/O data exchange. Many PLC masters, however, are unable to support more than 32 bytes, and this has become a typical value. Input and output data lengths for a given node are variable, and it is possible to define nodes as read only, write only or read/write.

The I/O data mixture used by a given slave device is defined by what is called a ‘GSD’ file, which can be edited to change the mapping of node parameters to PROFIBUS inputs and outputs. This file is imported into the network configuration before the network is created.

8.8.2. Data format

Data is transmitted in both directions as a single 16-bit integer value (also called a ‘register’). The value is returned as a scaled integer such that 999.9 is returned as 9999, and 1.234 is returned as 1234. The control program in the PROFIBUS master must convert these integers into floating point numbers if required. Alternatively, scaled integer number types can be used in the .uyp file to achieve the same end (see section 8.4 above).

8.9 GSD FILES

Figure 8.9 shows that for each instrument on the communications link, a Device Database File is constructed and loaded into the Profibus configuration terminal. These files (called Gerätestammdaten or GSD files) contain information, relating to the instrument's parameters, which the PROFIBUS master (a PLC in the figure) needs in order to communicate with the device.

When operating as a Profibus slave unit (not supported this version), it is necessary to load a Process Supervisor .gsd file into the master unit before communications can be established. A suitable .gsd file is supplied with the unit.

When operating as a master unit, .gsd files are required for all 'third party' equipment with which the Process supervisor is to communicate. Such .gsd files are normally supplied with the third party equipment. The Process Supervisor will normally come ready loaded with suitable .gsd files for I/O systems (for example) supplied by the Process Supervisor manufacturer

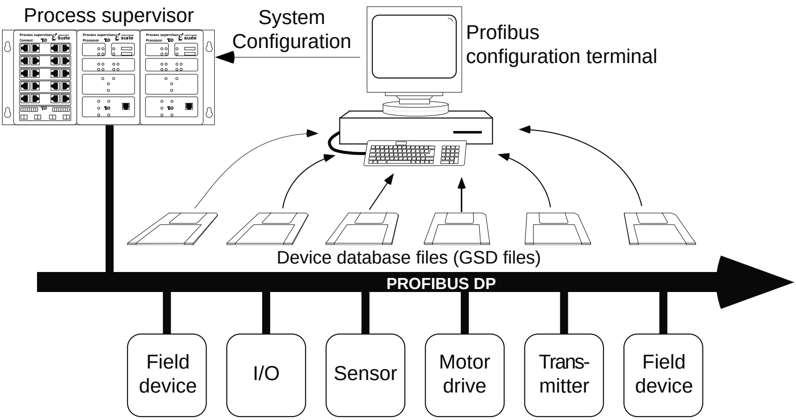


Figure 8.9 Typical PROFIBUS link using a PLC as master
(from section 3.3 of http://www.profibus.com)

8.10 REDUNDANT MODE WORKING

8.10.1 Redundancy decisions

When the units are acting as a duplex (redundant) pair, the primary and secondary units will independently derive the profibus status, and each unit will derive a ‘Profibus error weight’ (see table below).

Normal redundant operation will take place only if the primary processor believes that both processors have an equal view of the Profibus slaves. If the ‘Error weight’ for the primary processor is higher than that of the secondary processor, the units will desynchronise. If both units have an error weight of 1, the units will change over to try to achieve a better result.

The decsion (to remain synchronised, to desynchronise or to change over) is always made by the current primary processor, and then only whilst the two units are synchronised. I.E. An attempt to synchronise will be allowed to complete, and only after completion will the decision be made.

The decision is also deferred if the error weight is unstable. This prevents spurious desynchronise or changeover decisions being made as faults are introduced to or removed from the Profibus network.

I/OB LED	Description	State	Error weight
Steady green	The unit is running Profibus and successfully communicating with all slaves. For a redundant Secondary this means only that it can communicate with the primary, as this is the only node it may communicate with.	All OK	2
Flashing green	The unit is running Profibus and successfully communicating with at least one slave, but other slaves are not responding. This fault cannot appear on a Redundant Secondary as the necessary information is not available to it.	Faulty slave	2
Flashing red	The unit is running Profibus, but it cannot 'see' any slaves (e.g. because of a cable break).	Faulty network	1
Steady red	The unit is unable to start the Profibus because the Profibus circuit board is faulty.	Faulty hardware	1
Off	The Profibus is not running because there is no data base running, no Profibus port is configured or there is a configuration error preventing the Profibus from starting.	Not running	0

8.10.2 Examples

Figure 8.10.2a is a simplified version of the cabling in figure 2.4 above. Potential cable breaks are shown at points 1, 2 and 3, and the resulting reactions are given in table 8.10.2a.

Figure 8.10.2b and table 8.10.2b, below give similar information for an alternative cabling scheme.

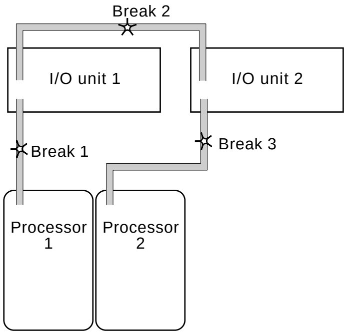


Figure 8.10.2a Break point locations

Break point	Processor reaction
Break 1	Processor 1 (P) can see no slaves. Processor 2 (S) cannot see the primary. Both report 'faulty network' (I/OB LEDs flash red). Processors change over. Processor 2 (P) now sees all slaves and reports 'OK' (LED steady green). Processor 1 (S) cannot see Processor 2 and reports 'Faulty network' (LED flashes red). Units desynchronise, with processor 2 the primary.
Break 2	Processor 1 (P) can see some slaves and reports 'faulty slave'. (LED flashes green). Processor 2 (S) cannot see the primary and reports 'faulty network' (LED flashes red). Units desynchronise, with processor 1 remaining the primary.
Break 3	Processor 1 (P) can see all slaves and reports 'OK' (LED steady green). Processor 2 (S) cannot see the primary and reports 'Faulty network' (LED flashes red). Units desynchronise, with processor 1 remaining the primary.
(P) = primary; (S) = secondary. Processor 1 initially primary; processor 2 initially secondary	

Table 8.10.2a Processor reaction to figure 8.10.2a cable breaks

8.10.2 EXAMPLES (Cont.)

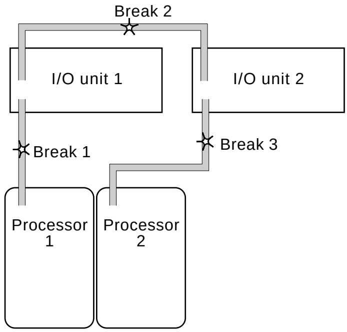


Figure 8.10.2b Alternative cabling

Break point	Processor reaction
Break 1	Processor 1 (P) can see no slaves and reports 'Faulty network'. I/OB LED flashes red. Processor 2 (S) can see the primary and reports 'OK' (LED steady green). Units remain synchronised.
Break 2	Processor 1 (P) can see some slaves and reports 'faulty slave' (LED flashes green). Processor 2 (S) can see the primary and reports 'OK' (LED steady green). Units remain synchronised.
Break 3	If processor 1 is the primary, Processor 1 (P) can see all slaves and reports 'OK' (LED steady green). Processor 2 (S) cannot see the primary and reports 'Faulty network' (LED flashes red). Units desynchronise, with processor 1 remaining the primary. If processor 2 is the primary, processor 2 (P) can see no slaves and processor 1 (S) cannot see the primary. Both report 'faulty network' (LEDs flash red). Processors change over. Processor 1 (P) now sees all slaves and reports 'OK' (LED steady green). Processor 2 (S) cannot see Processor 1 (P) and reports 'Faulty network' (LED flashes red). Units desynchronise, with processor 1 the primary.
(P) = primary; (S) = secondary. Processor 1 initially primary; processor 2 initially secondary	

Table 8.10.2b Processor reaction to figure 8.10.2b cable breaks

Notes:

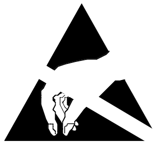
1. Terminators should be used at each end of the cable run, no matter what cabling scheme is used.
2. If a non-cable network fault ocurs (e.g. power loss to i/o units), then the Primry will see no slaves and report ‘faulty network’ (I/OB LED flashes red). The secondary can see the primary and so reports ‘OK’ (LED steady green). Units remain synchronised.

This page is deliberately left blank

Chapter 9 SERVICE

This section deals with the regular changing of filters, fans, and back-up batteries, and shows how to replace or upgrade the file-system flash memory.

For details of how to update the Profibus board, and how to change the unit's system software, boot ROM and libraries, please contact the nearest manufacturer's service centre.



Caution

All circuit boards associated with this unit are susceptible to damage due to static electrical discharges of voltages as low as 60V. All relevant personnel must be aware of correct static handling procedures.

9.1 PREVENTIVE MAINTENANCE SCHEDULE

The following periods are recommended to guarantee maximum availability of the processor unit, for use in what the manufacturer considers to be a normal environment. Should the environment be particularly dirty, or particularly clean, then the relevant parts of the schedule may be adjusted accordingly. For example, the fan filter may need replacing more frequently than every two years, if the unit is located in a dusty area.

The following are recommended, (based on fan manufacturer's figures specified at 60˚C):

6-monthly Make visual inspection of the external fans, and replace filters if clogging is evident

2-yearly Replace the filter associated with the external-fan

2-yearly Replace internal back-up battery board (if fitted).

4-yearly Replace internal and external fans.

Whenever the fan filter is replaced, it is recommended that a visual inspection of the interior of the processor unit be made, and any deposits of dirt or dust removed using a low-pressure compressed 'air duster' such as are available from most electronics distributors.

9.2 REPLACEMENT PROCEDURES

Figure 9.2 is a partial exploded view of the processor unit. The drawing shows one side plate removed, for clarity, but this is not always necessary for the procedures below.

9.2.1 Filter replacement

1. Remove the relevant processor module from the backplane, as shown (for the connection module) in Chapter 2 of this manual.
2. Undo and remove the six front panel securing screws ('A' in figure 9-1).
3. With the processor module vertical - i.e. with its connector on the bench, carefully lift the front panel away, releasing the ribbon cable connection as it becomes accessible. Place the front panel in a static-safe area ready for later re-assembly.
4. Remove the two jacking screws ('B') and place them to one side for later re-assembly.
5. Lay the module on one side, and slide the lower panel (containing the fan) out such that the four 4mm (7mm AF) fan securing nuts ('C') are accessible.
6. Undo these nuts, and ensuring all fixings are retained, remove the nuts and washers, and lift the fan off its studs.
7. Again, retaining all fixings, remove the fan filter and replace it with a new one.
8. Replacing all the washers previously removed, re-fit the fan and secure it using the M4 nuts ('C').
9. Slide the lower panel back into place, carefully ensuring that the fan cable harness is not damaged in the process.
10. With the processor module standing on its connector again, re-fit the two jacking screws ('B').
11. Re-fit the ribbon cable connector to the front panel, and re-fit the front panel to the module, ensuring that the jacking screws are located in their apertures in both back and front panels. Secure the front panel using the six screws ('A') previously removed.

    Note: This final part of the re-assembly is non-trivial. If a non-powered back plane is available, it is easier to secure the processor module onto the back plane before the front panel is fitted. This ensures that the jacking screws do not move during the fixing of the front panel.

12. If not already re-fitted, re-fit the module to the backplane and secure using the jacking screws, ensuring that the connector is correctly mated before tightening the screws, a few turns at a time each, to a final torque of not more than 2.5 Nm.

9.2 REPLACEMENT PROCEDURES (Cont.)

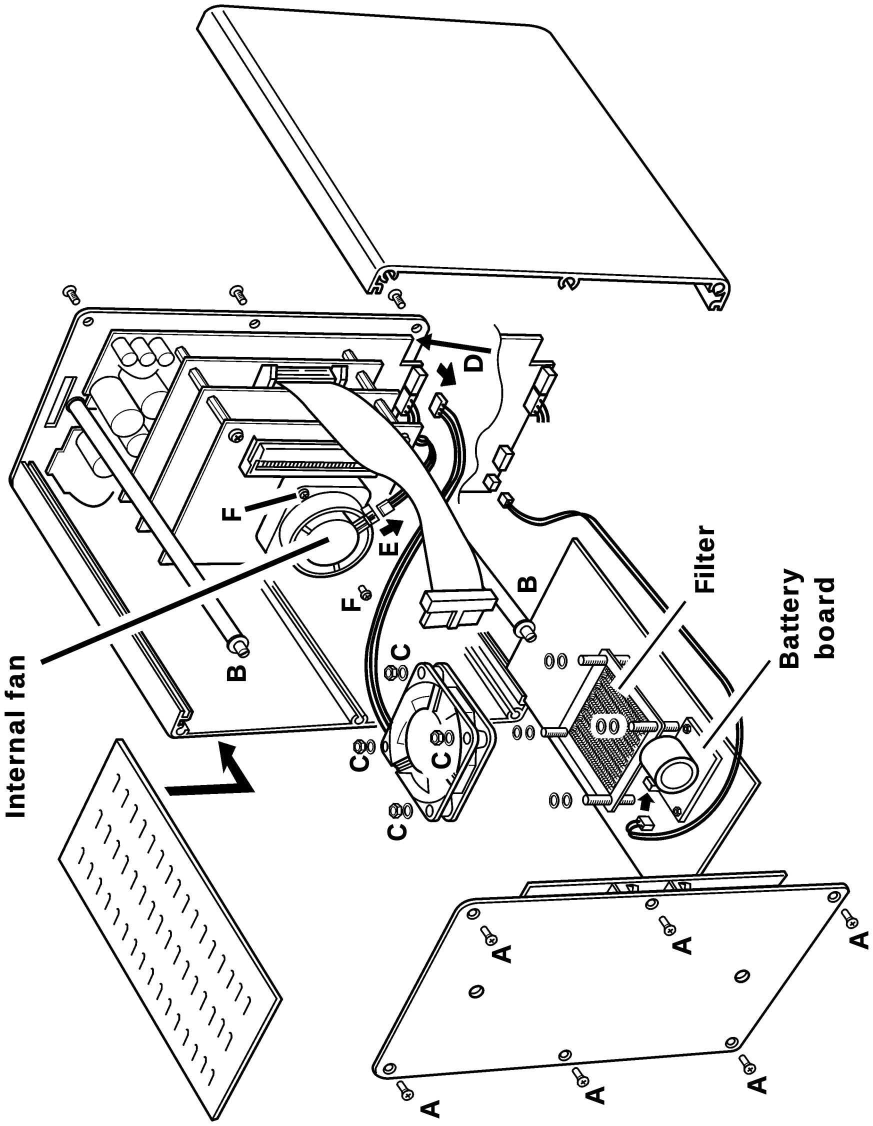


Figure 9.2 Processor module - part explosion

9.2.2 Fan replacement

1. Prepare the module, by carrying out steps 1 to 4 of section 9.2.1 above.
2. Lay the module on one side, and slide the lower panel (containing the external fan) out, disconnecting the fan connector (‘D’ in figure 9.2 above) as it becomes accessible.
3. Undo the four 4mm (7mm AF) fan securing nuts (‘C’), and ensuring all fixings are retained, remove the nuts and washers, and lift the fan off its studs and discard it.
4. Remove the fan filter and replace it with a new one.
5. Replacing all the washers previously removed, fit the new fan and secure it using the M4 nuts (‘C’).
6. Un-plug the internal fan (connector ‘E’ in the figure).
7. Taking care to retain all fixings, remove the fan by undoing its two securing screws (‘F’) (Size zero crosspoint screwdriver).
8. Fit the replacement fan and secure it using the screws (‘F’) previously removed. Re-make connector ‘E’.
9. Slide the lower panel back into place, carefully ensuring that the fan cable harness is not damaged in the process and re-make connector ‘D’.
10. Re-assemble the module as described in section 9.2.1 steps 10 to 12, above.

9.2.3 Battery board replacement

WARNING

The battery being replaced is likely to be partially charged, and must not be short-circuited, intentionally or inadvertently, as to do so carries a risk of explosion with possible emission of dangerous and corrosive materials.

Notes
1. Replacing the internal battery board causes the database to be lost, and a hot start is no longer possible. If the processor module is part of a redundant pair, a Synchronise request will restore the data base.
2. Although the battery on the replacement board is supplied partially charged, it is recommended that in order to achieve its specified back-up performance, it must be left in its processor module, with power applied, for approximately 48 hours. If power is removed before this period, data retention time will be reduced accordingly.

PROCEDURE

1. Prepare the processor module by carrying out section 9.2.1 steps 1 to 4.
2. Lay the module on one side, and slide the lower panel (containing the battery board) out until the board fixings become accessible.
3. Disconnect the battery connector.
4. Retaining all fixings, undo the two M3 screws securing the board to the case.
5. Remove the battery board, and place it in a safe, non-conductive area. Dispose of the battery board according to local regulations regarding Nickel-Metal hydride batteries.
6. Fit the new board, using the fixings previously removed.
7. Re-assemble the unit, following section 9.2.1 steps 9 to 12.

9.2.4 Upgrading the memory

Figures 9.2.4a and 9.2.4b show the replacement of the Flash file system memory card. The procedure also allows a 4MB version to be upgraded to an 8MB version, and data bases to be transferred from one processor module to another.

1. At the rear of the processor module, gently prize the plastic clips apart, to allow the memory circuit board to spring up.
2. Gently pull the board out of its connector, and fit the replacement.
3. Push the new board into place, such that it is retained by the plastic clips.

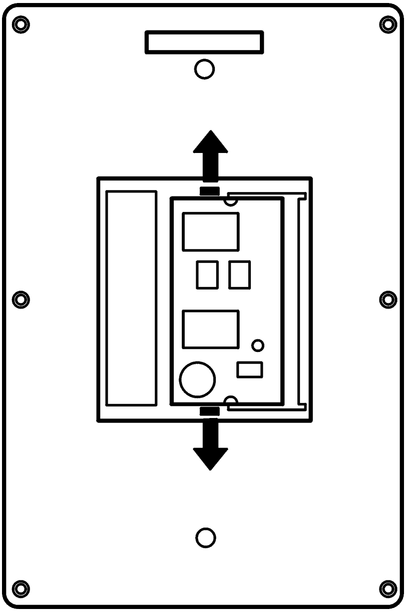


Figure 9.2.4a Separate clips

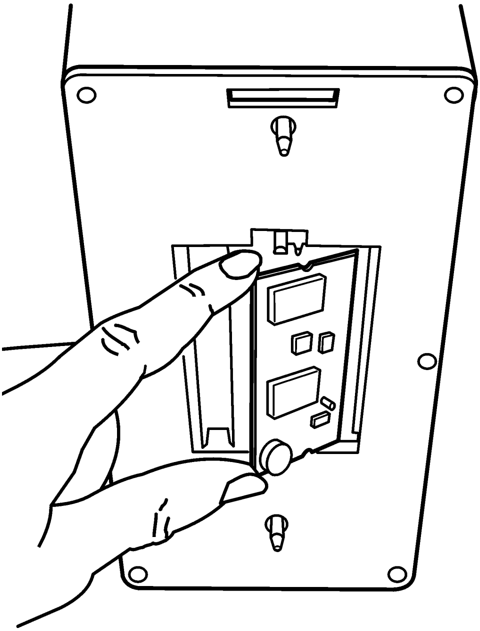


Figure 9.2.4b Remove/refit memory board

9.2.5 Action after firmware upgrade

In order to ensure that the RAM is cleared, the first start-up after any firmware upgrade (e.g. after upgrading the memory), must be a TEST start-up. See section 4.2 above for start-up details.

9.3 PHYSICAL ARRANGEMENT INSIDE PROCESSOR MODULE

Figure 9.3a shows the arrangement of circuit boards etc. inside the processor module, figure 9.3b is an interconnection diagram, and figure 9.3c is a simplified block diagram of the Process supervisor. In order to simplify figures 9.3b and 9.3c, only one processor and its associated power and communications connections are shown.

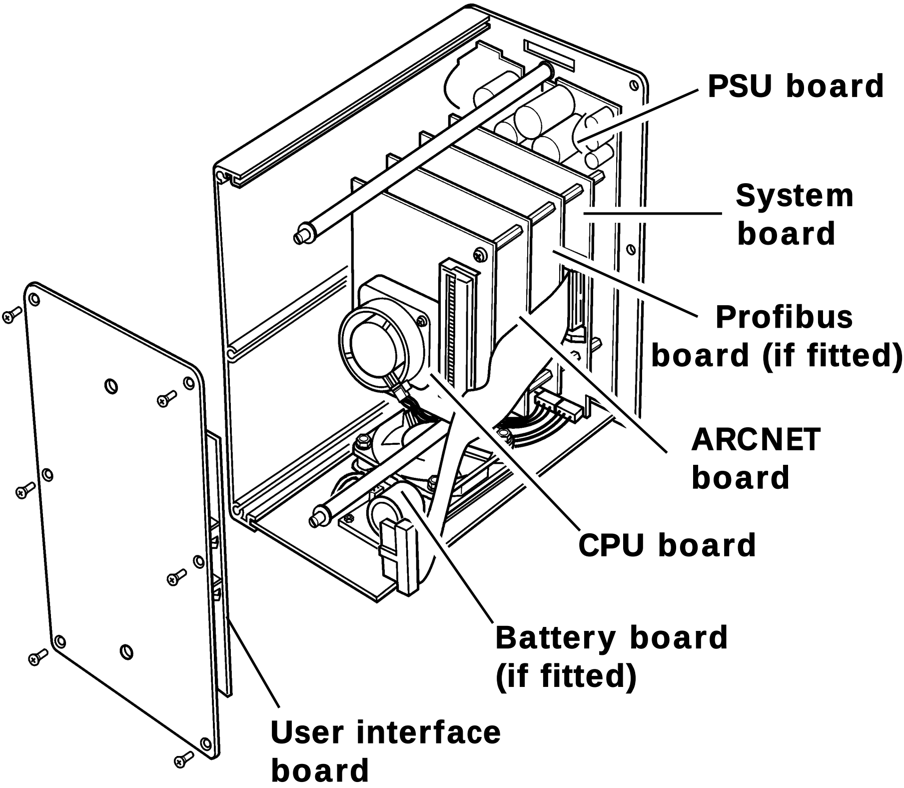


Figure 9.3a Hardware organisation

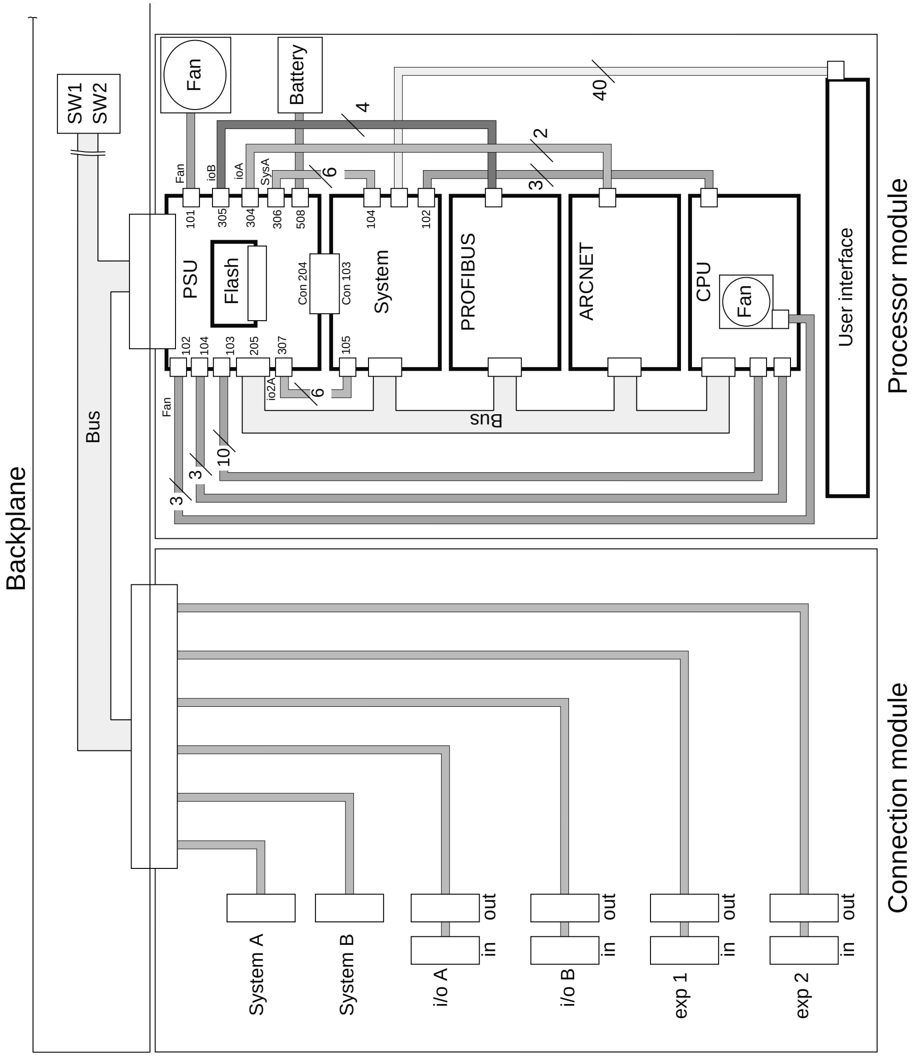


Figure 9.3b Hardware connections

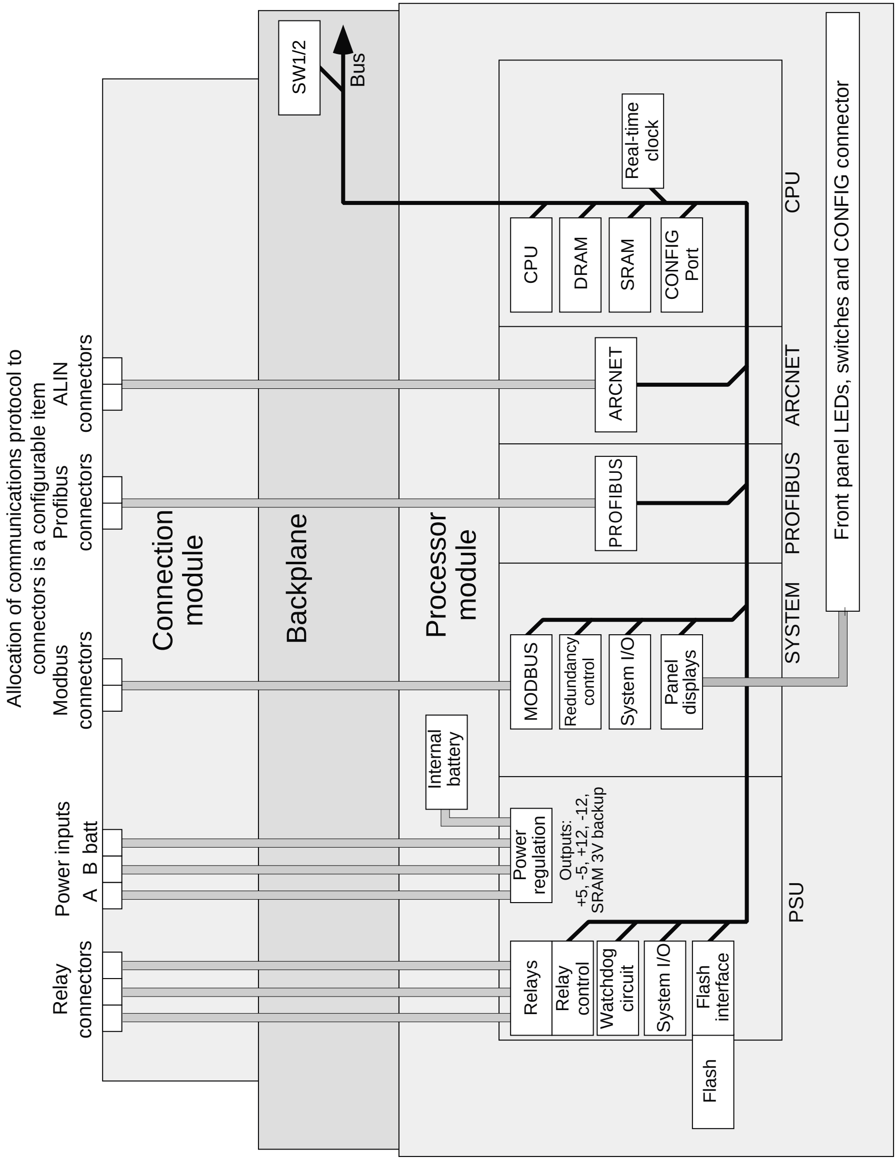


Figure 9.3c Simplified block diagram

9.4 THE MONITORS

9.4.1 The 'M' Monitor

Note: The 'M' monitor is intended as a diagnostic tool for Commissioning and/or Service Engineers. Access by other personnel is not recommended.

As described in chapter 4 above, the start up process can be monitored at a suitable computer terminal connected to the Processor unit ‘Config’ port. During the start-up sequence, the message “`Press 'm' key to stop auto-start`” appears. Operation of the 'm' key stops the start-up sequence and calls the top level menu depicted below. Operation of any other key stops the autostart process for 2 minutes, and repeats the message. If the message is ignored, the booting-up process continues.

```
Press 'm' key to stop auto-start
m
......Main  menu
 0: Quit
 1: Help
 2: Display saved system features
 3: Diagnostic tests
 4: Manual set-up
 5: Automatic set-up
 6: Format the User (E:) flash disk
......Selection: _
```

QUIT

Monitor is quitted, and the unit continues its start-up procedure

HELP

To be issued later

9.4.1 THE 'M' MONITOR (Cont.)

DISPLAY SAVED SYSTEM FEATURES

This screen is accessed by typing '2', then 'Y' or 'y' in response to the <Display?> enquiry. (Typing 'N' or 'n' returns to the main menu.) The display lists the current settings for the communications ports associated with this processor unit.

```
Sys Ethernet                -> Single

I/O Chan A                  -> Single
I/O chan A configuration -> ArcNet

I/O chan B configuration -> Serial

EXP chan A configuration -> Serial

EXP chan B configuration -> Serial
```

Sys Ethernet	Displays either 'Single' or 'Redundant' as appropriate for the internal system communications.
I/O Chan A	Displays either 'Single' or 'Redundant' for I/O A port.
I/O chan A configuration	Currently, always set to ArcNet (ALIN)
I/O chan B configuration	Either 'none' or 'Serial' (Modbus) or Profibus
EXP chan A configuration	Either 'none' or 'Serial' (Modbus) or Profibus
EXP chan B configuration	Either 'none' or 'Serial' (Modbus) or Profibus

To return to the main menu, type 'N' or 'n' in response to the <Display?> enquiry.

9.4.1 THE 'M' MONITOR (Cont.)

DIAGNOSTICS MENU

The diagnostics menu is accessed by typing '3', in the main menu. The menu allows a number of checks to be carried out, as detailed below.

Note: These checks can affect the machine’s ability to restart and should be used only to diagnose faults or to clear the memory.

```
......Diag Menu
......Level 1
 0: Quit
 1: Automatic test sequence
 2: PSE comm menu
 3: Net menu
 4: Profibus test
 5: SRAM menu
 6: Led Test
......Selection: _
```

AUTOMATIC TEST SEQUENCE

Accessed by typing '1' in the diagnostics menu, this carries out a number of tests, and displays the results either as 'OK' or 'ERROR', before returning to the Diagnostics menu.

```
RTC contents check ---> OK
Flash Disks ---> OK
SRAM status/signature ---> OK
Expansion serial comm port 1 ---> OK
Expansion serial comm port 2 ---> OK
Sys ethernet port 1 ---> OK
DRAM status ---> OK
I/O arcNet port 1 ---> OK
I/O Profibus port 2 ---> ERROR
```

9.4.1 THE 'M' MONITOR (Cont.)

DIAGNOSTICS MENU

PSE COMM MENU

Accessed by typing '2', this allows the Serial Communications to be checked. This test requires that a three- or five-wire cable be connected between Exp1 and Exp2 ports, with a cross over between Rx and Tx lines.

```
......PSE Comm Test
......Level 2

 0: Quit
 1: Set 9600 Baud
 2: Set 19200 Baud
 3: Set 38400 Baud
 4: Set 57600 Baud
 5: Set Modbus Ch 1 to Master
 6: Set Modbus Ch 2 to Master
 7: Start loop test 3W
 8: Start loop test 5W
......Selection: _
```

Baud Rate	Select required Baud rate for this test as required, by typing '1', '2', '3' or '4'. Baud rate is reset after the monitor is quitted.
Master/Slave	Typing '5' displays the message <Ch1 master?>... (Y,y,N,n). Typing 'Y' or 'y' sets EXP1 to Master status. Typing 'N' or 'n' sets it to Slave status. Typing '6' is similar, but for EXP2 port.
Start loop test 3W (5W)	Typing '7' or '8' starts the loop test for 3-wire or 5-wire systems respectively, once the number of repeats has been entered. Note that the number of repeats received should be the same as the number of repeats requested before the start of the test. If this is not the case, there is a problem with the communications link.

```
<NrRepeats>... n
Err......  Comm 1 0
Err......  Comm 2 0
Rx ......  Comm 1 n
Rx ......  Comm 2 n
Rx ......  buff 1 Eurotherm
Rx ......  buff 2 Eurotherm
```

- <NrRepeats>... n — Enter Number of repeats
- Err...... Comm 1 0 — No of errors detected in Comm 1
- Err...... Comm 2 0 — No of errors detected in Comm 2
- Rx ...... Comm 1 n — No of repeats received in Comm 1
- Rx ...... Comm 2 n — No of repeats received in Comm 2
- Rx ...... buff 1 / buff 2 Eurotherm — Contents of buffers at end of test

The PSE Comms test menu is re-displayed on the screen, allowing the user to quit the test, or to repeat it, perhaps at a different Baud rate.

9.4.1 THE 'M' MONITOR (DIAGNOSTICS MENU) (Cont.)

NET MENU

Not applicable to this release

PROFIBUS TEST

Not applicable to this release

9.4.1 THE 'M' MONITOR (DIAGNOSTICS MENU) (Cont.)

SRAM/SIGNATURE TEST MENU

Notes:
1 In order to use items 1 to 4 of the following menu successfully, the unit must have its mode switch set to 'Test'.
2 Items 5 to 8 appear only with software versions prior to 1.6

Typing '5' in the level 1 diagnostics menu calls the SRAM/Signature test menu to the screen, as depicted below.

```
......SRAM/Signature test menu
......Level 2
 0: Quit
 1: SRAM Format
 2: SRAM Signature contents
 3: Show the signature contents
 4: SRAM Access loop test
 5: DIMM Access loop test
 6: DIMM Read loop test
 7: DIMM SRAM enable test
 8: DIMM SRAM Write / Read one Address test
......Selection: _
```

SRAM FORMAT

Typing '1' followed by a carriage return clears the Static memory. Such clearing of the Static Memory may be required under circumstances, such as the following:

1. It is necessary to clear the memory of its last redundancy state after transferring a processor module from one system to another.
2. To clear the memory in order to prevent a 'Hot Start' because the previous data base is no longer required at start-up.

SRAM SIGNATURE CONTENTS

Typing '2', followed by a carriage return, initialises the unit's hot start information

SHOW THE SIGNATURE CONTENTS

Selecting '3' Shows the signature contents, e.g.

```
SRAM Signature T940
Signature Checksum 28016
```

9.4.1 THE 'M' MONITOR (DIAGNOSTICS MENU) (Cont.)

SRAM ACCESS LOOP TEST

This test applies to the Static RAM memory located on the memory board at the rear of the unit.

The user enters a start address and a number of repeats. Once initiated, the loop test writes and reads back five incremental values to the start address, then repeats this for start address + 1, then start address + 2 and so on until start address + N is reached ('N' is the specified number of repeats), at which time the test stops.

For example:

```
<Start Address>...d00018<CR>
<Nr Repeats>...4<CR>
Sram addr = d00018 = 00 : Sram addr = d00018 = 01 :
 Sram addr = d00018 = 02 : Sram addr = d00018 = 03 :
 Sram addr = d00018 = 04 :
Sram addr = d00019 = 05 : Sram addr = d00019 = 06 :
 Sram addr = d00019 = 07 : Sram addr = d00019 = 08 :
 Sram addr = d00019 = 09 :
Sram addr = d0001a = 0a : Sram addr = d0001a = 0b :
 Sram addr = d0001a = 0c : Sram addr = d0001a = 0d :
 Sram addr = d0001a = 0e :
Sram addr = d0001b = 0f : Sram addr = d0001b = 10 :
 Sram addr = d0001b = 11 : Sram addr = d0001b = 12 :
 Sram addr = d0001b = 13 :
Sram addr = d0001c = 14 : Sram addr = d0001c = 15 :
 Sram addr = d0001c = 16 : Sram addr = d0001c = 17 :
 Sram addr = d0001c = 18 :
```

Notes:
1. There will normally be more than two addresses per line – the above diagram is designed for clarity rather than exactness.
2. User entries are emphasised in the diagram above by being underlined. Such underlining does not appear on the actual screen.
3. <CR> represents user operation of the [carriage] return or enter key.
4. Address range starts at d00000.
5. Should the end of the memory be reached, the test returns to the specified start address and continues from there.

DIMM ACCESS LOOP TEST

This test applies to memory located on the CPU board located inside the unit and operates in the same way as the SRAM loop test described above, except that the address range is different (lowest address = 400000).

This test does not appear as a menu item with software versions 1.6 onwards.

9.4.1 THE 'M' MONITOR (DIAGNOSTICS MENU) (Cont.)

DIMM READ LOOP TEST

The user enters a start address and a number of repeats. Once initiated, the loop test displays the contents of the start address, followed by (start address + 1) etc. up to (start address +n) where n is the specified number of repeats.

Example:

```
<Start Address>... 401018 <CR>
<Nr Repeats>... 4<CR>

Dimm addr = 401018 = ffffffff :
Dimm addr = 401019 = ffffffff :
Dimm addr = 40101a = ffffffff :
Dimm addr = 40101b = ffffffff :
```

Should the end of the memory be reached, the test returns to the specified start address and continues from there.

This test does not appear as a menu item with software versions 1.6 onwards.

DIMM SRAM ENABLE

Selecting '7' from the menu enables the SRAM area of the DIMM/SRAM board, and returns the following message (or one similar according to SRAM size):

```
DIMM SRAM enable gives SRAM size of 3072 K bytes
```

This test does not appear as a menu item with software versions 1.6 onwards.

DIMM SRAM WRITE/READ ONE ADDRESS TEST

Selecting '8' from the menu allows one particular DIMM address to be exercised, by repeatedly writing the value hex 55 (0x55) to it and displaying the result. The address and the number of repeats are user entered values.

This test does not appear as a menu item with software versions 1.6 onwards.

Example:

```
<Start Address>...400010
<Nr Repeats>...2

Dimm addr = 400010 = 55
Dimm addr = 400010 = 55
Dimm addr = 400010 = 55
```

9.4.1 THE 'M' MONITOR (DIAGNOSTICS MENU) (Cont.)

LED TEST

Selecting '6' from the Diagnostic menu allows an LED test sequence to be initiated, as described below. The test is started by selecting ‘1’ from the LED sub menu.

The test sequence is as follows:

1. Duplex LED switched green, red, off.
2. battery int LED switched green. (Off at step 16)
3. battery ext LED switched green. (Off at step 17)
4. System A LED switched green, red, off.
5. System B LED switched green, red off.
6. i/o A LED switched green, red, off.
7. i/o B LED switched green, red, off
8. Standby LED switched yellow. (Off at step 18)
9. Primary LED switched green. (Off at step 19)
10. Exp1 Tx LED switched yellow. (Off at step 20)
11. Exp1 Rx LED switched yellow. (Off at step 21)
12. Exp2 Tx LED switched yellow. (Off at step 22)
13. Exp2 Rx LED switched yellow. (Off at step 23)
14. rl1 LED switched yellow. (Off at step 24)
15. rl2 LED switched yellow. (Off at step 25)
16. battery int LED switched off.
17. battery ext LED switched off.
18. Standby LED switched off.
19. Primary LED switched off.
20. Exp1 Tx LED switched off.
21. Exp1 Rx LED switched off.
22. Exp2 Tx LED switched off.
23. Exp1 Tx LED switched off.
24. rl1 LED switched off.
25. rl2 LED switched off.

Type 2<CR>
to stop test after step 25

If not stopped, by typing ‘2 <CR>’ during the above sequence, the test continues with the Primary, Standby and Duplex LEDs flashing, and, at the same time, the following LEDs being switched on and off one after the other in an endless loop:

int, sysA, sysB, i/oB, i/o A, ext, rl1, exp1 tx, exp2 tx, exp2 rx, exp1 rx, rl2.

This loop is stopped, and the LED sub-menu redisplayed, by typing '2' <CR> but it should be noted that it will continue until its next conclusion (rl2 off) which can take over 20 seconds.

9.4.2 The 'S' Monitor

Note: The 'S' monitor is intended only as a diagnostic tool for Commissioning and/or Service Engineers. Because of the unit's safety-critical requirements, access by other personnel is not recommended.

The operating boot program includes a diagnostics facility called the 'S' monitor. If a suitable terminal is connected (Chapter 2, section 2.4.2 above) to the Config port on the relevant processor unit, then during the start up process, the message
`Press 's' key to stop auto-boot`
is displayed for about one second. Operation of the 's' key during this period calls the top level menu depicted below. Operation of any other key stops the autoboot process for 2 minutes, and repeats the message. If the message is ignored, the booting-up process continues.

```
Press 's' key to stop auto-boot
s
......Main menu......level 0
 0: Quit
 1: Help
 2: Display basic machine status
 3: Display extended machine status
 4: Diagnostic menu
 5: Memory status
 6: Show boot info
 7: Date/Time set
 8: Format the Primary flash disk
 9: Format the Secondary flash disk
......Selection: _
```

QUIT

Selecting ‘0’ in this menu quits the ‘S’ monitor. If the watchdog retry enable switch is set 'on' (section 2.3.5 above), the unit will re-start. If the switch is set off, the unit must be reset manually, or powered off, then back on again.

HELP

To be issued later

9.4.2 'S' MONITOR (Cont.)

DISPLAY BASIC MACHINE STATUS

Accessed by selecting '2' from the main menu, this page displays the following information:

```
<Display?>...  (Y,y,N,n) Y<CR>

 RTC power                 -> Ok (Real-time clock lost power)
 CMOS checksum             -> Ok (CMOS checksum is bad)
 Memory compare            -> Ok (Memory size compare error)
 CMOS time                 -> Ok (CMOS time invalid)
 Flash file system status -> Ok (E, S disk status)
 SRAM module               -> Ok (Check only the presence)
<Display?>...  (Y,y,N,n) N<CR>
......Main menu......level 0
etc.
```

DISPLAY EXTENDED MACHINE STATUS

Accessed by selecting '3' from the main menu, this page displays the following information:

```
<Display?>...  (Y,y,N,n) Y<CR>

 Register 0x0F = Reason for shutdown     =>>>    0 = 0

 Register 0x10 = diskette0 set-up        =>>>    1.44 M drive
 Register 0x10 = diskette1 set-up        =>>>    None

 Register 0x12 = HD0 disk set-up         =>>>    None type =>>>    0
 Register 0x12 = HD1 disk set-up         =>>>    None type =>>>    0
 Register 0x14,bit5/4 = Primary display  =>>>    EGA/VGA

Date                                     =>>>    DD/MM/YY
Time                                     =>>>    HH:MM:SS

<Display?>...  (Y,y,N,n) N<CR>
......Main menu......level 0
etc.
```

9.4.2 THE 'S' MONITOR (Cont.)

DIAGNOSTICS MENU

The diagnostics menu (depicted below) is called by typing '4' from the top level menu above.

```
......Diag Menu
......Level 1
 0: Quit
 1: Watchdog register
 2: System LED
 3: I/O LED
 4: Serial LED
 5: ILOCK WRO Output
 6: Read input status
 7: Flash driver menu
 8: Connect the interrupts (5,9,11,12,15)
```

WATCHDOG REGISTER

Accessed by selecting ‘1’ from the diagnostics menu, this page displays the following information:

```
......Watchdog menu
......Level 2
 0: Quit
 1: Bit 7 = Enable flash Vpp
 2: Bit 6 = Flash write protection
 3: Bit 5 = Redundancy interrupt
 4: Bit 4 = Watchdog Relay
 5: Bit 3 = Watchdog Pat
 6: Bit 2 = Alarm relay 1
 7: Bit 1 = alarm relay 2
```

Notes:
1. Switching the alarm relays also switches their associated LED
2. Switching the watchdog relay has no effect on the watchdog LED

SYSTEM LED

Accessed by selecting ‘2’ from the diagnostics menu, this page allows the two ‘battery’ LEDs and the ‘Duplex’ LED to be exercised individually.

9.4.2 THE 'S' MONITOR (DIAGNOSTICS MENU) (Cont.)

I/O LED

Accessed by selecting ‘3’ from the diagnostics menu, this page allows the Comms panel ‘system A’, and ‘B’ and i/o ‘A’ and ‘B’ LEDs to be exercised individually.

SERIAL LED

Accessed by selecting ‘4’ from the diagnostics menu, this page allows the Comms panel ‘exp1’ and ‘exp 2’ Rx and Tx LEds, the Alarm panel ‘rl1’ and ‘rl2’ LEDs and the ‘Primary’ and ‘Standby’ LEDs to be exercised individually.

Note:
Setting the rl1 or rl2 LED on does not switch the associated relay on. Switching rl1 or rl2 on in the watchdog menu (above) does cause the associated LED to be illuminated.

ILOCK WR0

Accessed by selecting ‘5’ from the diagnostics menu, this redundancy control monitor page displays the following information:

```
......Ilock wr0
......Level 2
 0: Quit
 1: Bit 5 = Reset minor fault
 2: Bit 2 = A request clocks
 3: Bit 1 = A Ok
 4: Bit 0 = A Req Primary
......Selection: _
```

9.4.2 THE 'S' MONITOR (DIAGNOSTICS MENU) (Cont.)

READ INPUT STATUS

Accessed by selecting ‘6’ from the diagnostics menu, this page displays the following information:

```
<DISPLAY?>...  (Y,y,N,y): Y<CR>
 Byte 1 = ALIN address              =>>>     ffffffa0 = -96
 Byte 2 = ADDR_HIGH register        =>>>     8a = 138
 Byte 2,bit7 = Power Fail           =>>>     1
 Byte 2,bit6 = RTC Battery Failure  =>>>     0
 Byte 2,bit5 = Over temperature     =>>>     0
 Byte 2,bit4 = CPU fan stall        =>>>     0
 Byte 2,bit3 = Main Batt failure    =>>>     1
 Byte 2,bit2 = Main fan stall       =>>>     0
 Byte 2,bit1 = Spare                =>>>     0
 Byte 2,bit0 = Spare                =>>>     0

 Byte 3 = DIL register              =>>>     47 = 71
 Byte 3,bit7 = Spare                =>>>     0
 Byte 3,bit6 = mode 4               =>>>     1
 Byte 3,bit5 = Hardware Build Lev.1 =>>>     0
 Byte 3,bit4 = Hardware Build Lev.0 =>>>     0
 Byte 3,bit3 = Spare                =>>>     0
 Byte 3,bit2 = halt                 =>>>     1
 Byte 3,bit1 = mode 2               =>>>     1
 Byte 3,bit0 = mode 1               =>>>     1

 Byte 4 = ILOCK_RD0 register        =>>>     24 = 36
 Byte 5 = ILOCK_RD1 register        =>>>     8 = 8
<DISPLAY?>...  (Y,y,N,y): N<CR>
......Diag Menu
etc.
```

9.4.2 THE 'S' MONITOR (DIAGNOSTICS MENU) (Cont.)

FLASH DRIVER MENU

The Flash driver file system menu (depicted below) is called by selecting '7' from the diagnostics menu.

```
......Flash driver Menu
......Level 2
 0: Quit
 1: flashFsShow S: disk
 2: flashFsShow E: disk
......Selection: _
```

This allows details of the system (S) and database (E) flash memories to be displayed. The S: disk is depicted below; the display for the E disk is similar.

```
FLASH DEVICE
    pVolDesc            0x9f8d8
    max files num       16
    compression         not installed

  FLASH DRIVER INFO:
    pFlashDev   0x9fd18
Vendor: Intel, Device i28F016SA, sector size 65536 bits
Total number of sectors: 32, Total capacity 2097152 bytes
Sectors being erased: .
  CURRENT FLASH VOLUME:
    long names          NOT enabled
    volume label:       “System     “
    boot record ptr ( 4, 119 ) : root dir ptr ( 4, 104 )
    current sector for reclaim  9
  SECTORS STATUS
( 0: Blk 9 )( 1: Blk 2 )( 2: Blk 1 )( 3: Blk 3 )( 4: Blk 4 )( 5: Blk 5 )( 6: Blk
 6 )( 7: Blk 7 )( 8: Blk 8 )( 9: spare )( 10: Blk 0 )( 11: Blk 10 )( 12: Blk 11
)( 13: Blk 12 )( 14: Blk 13 )( 15: Blk 14 )( 16: Blk 15 )( 17: Blk 16 )( 18: Blk
17 )( 19: Blk 18 )( 20: Blk 19 )( 21: Blk 20 )( 22: Blk 21 )( 23: Blk 22 )( 24:
Blk 23 )( 25: Blk 24 )( 26: Blk 25 )( 27: Blk 26 )( 28: Blk 27 )( 29: Blk 28 )(
30: Blk 29 )( 31: Blk 30 )
   total sectors 32 : blocks 31
   sectors; nSpare 1 : nWaitForErase 0 : nBad 0
      erase count min 3 - max 3 : average 3
   block size                 65536 = 0x10000
   the most extent size       4086 bytes
   free space: total           331728 = 0x050fd0
                without reclaim 29119 = 0x71bf
===========================================
```

9.4.2 THE 'S' MONITOR (DIAGNOSTICS MENU) (Cont.)

CONNECT THE INTERRUPTS

Accessed by selecting ‘8’ in the diagnostics menu, this page is for use only by the Manufacturer.

MEMORY STATUS

This page is accessed by selecting item 5 from the Main menu, and presents memory information as follows:

```
<Display?>...  (Y,y,N,n): Y<CR>

 Register 0x15/16 = Base memory in kbyte      =>>>     280 = 640
 Register 0x33    = Extension memory in kbyte =>>>     80 = 128
 Register 0x17/18 = Extension memory in kbyte =>>>     c00 = 3072
 Total DRAM size in kbyte                     =>>>     f00 = 3840
 Total  SRAM size                             =>>>     100000 = 1048576
 Total flash virtual size                     =>>>     400000 = 4194304
 Primary flash disk size                      =>>>     200000 = 2097152
 Second flash disk size                       =>>>     200000 = 2097152
 Primary empty flash disk size                =>>>     50fd0 = 331728
 Second empty flash disk size                 =>>>     1ef93d = 2029885
<Display?>...  (Y,y,N,n): n<CR>
......Main menu
etc.
```

SHOW BOOT INFO

This page is accessed by selecting item 6 from the Main menu, and presents boot information as follows:

```
<Display?>...  (Y,y,N,n): Y<CR>

 Boot device                -> System A Net
 Boot file                  -> /flash/vxWorks
 Host name                  -> host’s name
 Target name                -> PSE
 Target IP addr             -> 10.1.1.1
 Host IP addr               -> 0.0.0.0
 Gateway IP addr            ->
 FTP user                   -> guest
 FTP password               -> guest
<Display?>...  (Y,y,N,n): n<CR>
......Main menu
etc.
```

9.4.2 THE 'S' MONITOR (Cont.)

DATE / TIME SET

Accessed by selecting item 7 from the main menu, this allows the user to set the date and time.

FORMAT THE PRIMARY (SECONDARY) FLASH DISK

This facility is accessed by selecting item 8 or 9 from the main menu.

CAUTION

Formatting either of these discs destroys its contents, which include data base and set-up files.

Chapter 10 Specification and Order Codes

10.1 SPECIFICATION

This specification defines the Process supervisor components:
T310 Backplane
T320 Connection Module
T940 Processor Module

10.1.1 General specification

Physical

Dimensions

Backplane:	402mm wide x 180mm high x 24mm deep
Connection Module:	120mm wide x 180mm high x 126mm deep
Processor Module:	120mm wide x 180 mm high x 186mm deep
Backplane fixing centres:	382 horizontal x 125 vertical

Weight

Backplane without modules:	2kg. max
Connection module:	1.1kg max.
Processor module:	2.4kg max. (each)

Environmental

Temperature	Storage:	-25 to +85°C
	Operation:	0 to + 50°C
Humidity	Storage/Operation:	5 to 95% RH (non-condensing)
RFI	EMC emissions:	BS EN50081-2 Generic standard (industrial)
	EMC immunity:	BS EN50082-2 Generic standard (industrial)
Safety Specification		BS EN61010-1/A2:1993
Vibration		To IEC1131-2 section 2.1.3 (0.075mm peak amplitude10 to 57 Hz; 1g 57 to 150 Hz)

Power Requirements

Main supply:		24V dc nom. (18 to 36Vdc) at 50W per processor module, maximum. Two supplies can be connected per processor module, to ensure continued operation should one supply fail.
Backup supplies	External (option):	2.4 to 5Volt battery. Typical drain per processor = 3.4mA at 3.4V.
	Internal (option):	Nickel/metal hydride battery board maintains data base and real-time clock for 72 hours (min.).
Fusing	24V supplies	3A Type T in each positive supply line
	External batteries:	0.5A Type T in each positive supply line

10.1.2 Backplane specification

General

Switches	SW1:	ALIN address
	SW2, segment 1:	Watchdog retry (trip and try again mode)
	SW2, segment 6:	Redundant/non-redundant mode select (duplex/simplex)
	SW2, segment 5:	Modbus select
Safety earth connection		By M4 earth stud on right hand flange of the backplane

10.1.3 Connect module specification

ALIN ports

Connectors	Parallel wired pairs of shielded RJ45 connectors per processor unit.
Network medium	ArcNet (screened twisted pair, 100 Ohm)
Network type	Token bus
Speed	2.5 Mbits/sec.
N° of nodes (max)	8, extendable by repeater
Line length (max)	100 metres, extendable by repeater
Isolation	60Vdc / 30V ac; 5.6kΩ to 0V

Modbus/Jbus (EIA422/485)

Connectors	Parallel wired pairs of shielded RJ45 connectors per processor unit.
Protocol	MODBUS/JBUS RTU slave
Data rate	Selectable between 600 and 38,400 Baud
Data format	8-bits, 1 or 2 stop bits, selectable parity
MODBUS data tables	16, configurable as registers or bits
Table length (max.)	200 registers or 999 bits
Memory allocated to tables	6000 bytes
Isolation	60Vdc / 30V ac

Modbus (DCM)

Connectors	Parallel wired pairs of shielded RJ45 connectors per processor unit.
Protocol	MODBUS/JBUS RTU master
Data rate	Selectable between 600 and 38,400 Baud
Data format	8-bits, 1 or 2 stop bits, selectable parity
Isolation	60Vdc / 30V ac

Profibus

Connectors	Parallel wired pairs of shielded RJ45 connectors per processor unit.
Protocol	Profibus DP/DPV1
Data rate	Selectable between 9600 and 12M Baud
Isolation	60Vdc / 30V ac; 1MΩ to Chassis

10.1.3 CONNECT MODULE SPECIFICATION (Cont.)

Other connections

Supply voltage:	Two 2-way connectors per processor module for connection of 24V (nom.) supply.
Safety Earth:	See backplane specification above
Battery backup:	For each processor unit, one external battery can be connected using two terminals of an eight-way (relay) connector block.
Relay connections:	For each processor module there are one watchdog relay and two 'alarm' relays (operation configured by the user). For each relay, only the common and normally open contacts are used, these being short circuit under normal operating conditions, and open circuit under alarm or power-off conditions.

Relay specification

One watchdog and two user configurable relays per processor.

Contact rating (resistive)	30V ac/60V dc at 0.5Amps
Isolation (Contact-to-ground)	30V ac (RMS) or 60V dc.

10.1.4 Processor Module specification

General

CPU type		AMD586; 133MHz
SRAM		1 MByte (requires internal or external battery backup)
Flash memory	System:	2 MByte
	User:	2 MByte
Serial Communications		Non-isolated RS232 terminal configuration port (RJ11 connector)

Panel Indicators

Light emitting diodes (LEDs) for:

- Main supply (24V dc nom)
- External battery (optional)
- Internal battery (optional)
- Alarm relay status
- Serial communications
- ALIN/Profibus status
- Primary processor
- Standby processor
- Watchdog indicator
- Duplex (redundant mode) indicator

Control switches

Push button switches for	Watchdog Halt
	Watchdog Restart
	Processor module synchronisation/changeover
	Processor module desynchronisation
Rotary switch for	Start-up mode selection

10.1 SPECIFICATION (Cont.)

10.1.5 Software specification

LIN Block libraries (continuous database function block categories)

I/O:	Analogue and digital input output manual override
Conditioning:	Dynamic signal-processing and alarm collection
Control:	Analogue control, simulation and communications
Timing:	Timing, sequencing, totalisation and events.
Selector:	Selection, switching, alarm and display page management
Logic:	Boolean, latching, counting and comparison
maths:	Mathematical functions and free-format expressions.
Config:	Unit identity blocks
Diag:	Diagnostics
Batch:	Sequencing recipe/record and discrepancy checking.

Continuous database resources

Number of function blocks	1024
Number of templates (maximum)	50
Number of libraries (maximum)	28
Number of EDBs (maximum)	32
Number of FEATTs (maximum)	1024
Number of TEATTs (maximum)	512
Number of Servers (maximum)	8
Number of connections	1024
Control database size (maximum)	256 kByte
SFC database size (Maximum)	512 kByte

Notes:

1. Apart from database memory sizes, the figures above are default maximums and are the recommended limits for typical situations. Subject to note 2, below, it is possible to exceed some of the above maxima, although if a database with more resources than the default maximum is loaded, then the maximum is set to the new value and there may then be insufficient memory to load the entire database. In such a case, the ‘connections’ disappear first. (FEATTs are not subject to this problem, since when a database is saved, there are not normally any FEATTs present, so the default maximum cannot be overridden.
2. The EDB maximum must not be exceeded. If it is, some EDBs will malfunction, and this is likely to affect the LINtools VIEW facility.

10.1.5 SOFTWARE SPECIFICATION (Cont.)

Continuous database performance

To be issued later

Sequence Control Resources

Sequence memory	Program data:	256 kBytes
	SFC Resources:	512 kBytes
	N° of independent sequence tasks:	40 simultaneously active
	SFC actions:	200, including root SFCs
	Steps:	640
	Action associations:	2400
	Actions:	1200
	Transitions:	900

10.2 ORDER CODES

10.2.1 Instrument order code

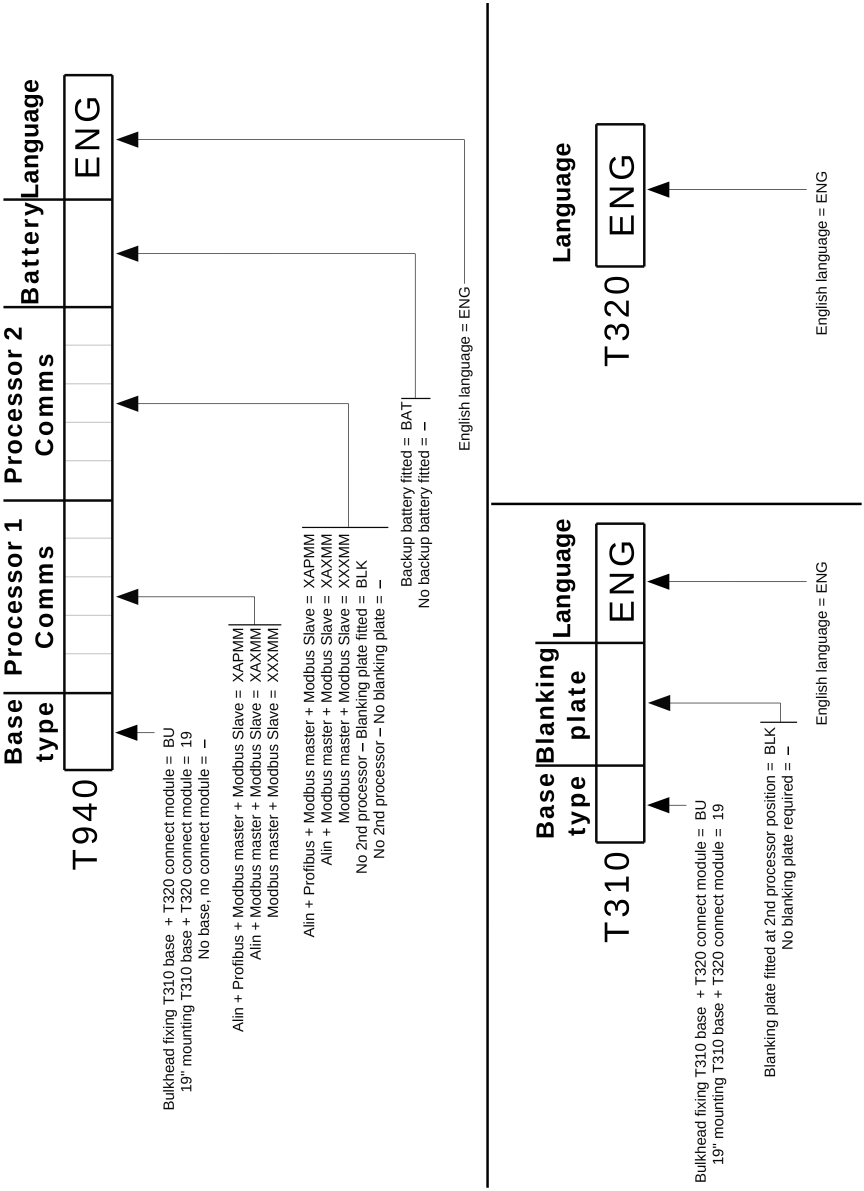


Figure 10.2.1 Instrument ordering guide

10.2.2 Spares and accessories

Internal battery board	AH261438
Cable harness for battery board	DN261448
Main fan assembly (on processor module lower panel)	LA260259
Filter for main fan assembly	BH240476
CPU fan assembly	To be issued later
Cable harness for CPU fan assembly	DN260327
Power supply (Input: mains; Output 24Vdc at 2.5 A. max.)	2500P/2A5
Power supply (Input: mains; Output 24Vdc at 5 A. max.)	2500P/5A0
Power supply (Input: mains; Output 24Vdc at 10 A. max.)	2500P/10A
External 4V battery	S9537
Charger for external battery (Supply voltage = 24Vdc)	S9538/24V
Communications Isolator (EIA232 - EIA232)	To be issued later

ALIN Cables

1x RJ45 connector and ferrules for screw terminals	S9508-5/1RJ45/xxx/-
RJ45 connectors both ends	S9508-5/2RJ45/xxx/-
Ferrules both ends	S9508-5/2FER/xxx/-
RJ11 connector one end; RJ45 connector at other end	S9508-5/RJ11-45/xxx/-
Cable without termination	S9508-5/ - /xxx/-

(xxx = cable length in 10 cm. increments to 100 metres max.)
(Final hyphen is “connector boot colour = default”. Consult factory for other colours)

Profibus in-cubicle cables

1x RJ45 connector and ferrules for screw terminals	S9508-5/1RJ45/xxx/-
RJ45 connectors both ends	S9508-5/2RJ45/xxx/-

(xxx = cable length in 10 cm. increments to 100 metres max.)
(Final hyphen is “connector boot colour = default”. Consult factory for other colours)

Configuration terminal cables

RJ11 to 9-way D-type	DN026484
RJ11 to 25-way D-type	To be issued later

Cable accessories

ALIN terminal-mounted line terminator	LA082586U002
ArcNet/MODBUS line terminator (RJ45)	CI026528
Profibus line terminator (RJ45)	CI026529
Feed-through adapter (RJ45)	CI250407
Shielded RJ45 connector, unassembled	CI250449
RJ45 connector assembly handtool	Consult factory
ALIN (ArcNet) hub	S9576

10.3 COSHH

Nickel-Metal Hydride batteries

Product:	BACK-UP BATTERIES		
Part numbers: PA61437 (mounted on circuit board assembly AH261438)			

HAZARDOUS INGREDIENTS

Name	% Range	TLV	Toxicological data
Nickel hydroxide	10	Not established	Highly toxic if ingested
Nickel metal	20 - 26	Not established	
Misch metal alloy	10-11	Not established	
Potassium hydroxide	8	Not established	Highly toxic, Highly corrosive.

PHYSICAL DATA

Boiling point	Not applicable	Specific gravity	Not applicable
Vapour pressure	Not applicable	Solubility in water	Not applicable
Odour	Not applicable	Colours	Not applicable

FIRE AND EXPLOSION DATA

		FLAMMABLE LIMIT	
Flash point (deg C) (Method used)	Not applicable	LEL	UEL
Extinguishing media	Any	Not applicable	Not applicable
Special fire-fighting procedures	Not applicable		
Unusual fire and explosion hazards	Batteries might explode due to excessive presure build-up which might not be self-venting. Toxic fumes might be generated.		

HEALTH HAZARD DATA

Threshold limit value	Not applicable		
LD 50 Oral	Not applicable	LD 50 Dermal	Not applicable
Skin and eye irritation	Should cells leak, the leak material will be a caustic solution. Avoid contact.		
Over-exposure effects	Not applicable		
Chemical nature	See above. There are no risks in normal use.		

10.3 COSHH (Cont.)

Nickel-Metal Hydride batteries

HEALTH HAZARD DATA (Cont.)			
FIRST AID PROCEDURES			
Eyes and skin	If leakage occurs, wash the affected area withplenty of water and cover with dry gauze. If eyes are affected, wash with plenty of water. Seek medical assistance.		
Ingestion	If ingestion of leak material occurs, DO NOT induce vomiting. Give plenty of milk to drink. Obtain immediate medical assistance, stating 'nickel/metal-hydride battery'.		
Inhalation	Not applicable		
REACTIVITY DATA			
STABILITY			Conditions to avoid
Stable	Yes	Unstable	Mechanical damage, overcharging, short circuiting terminals, charging temperatures outside the range 0 to 65° C Direct soldering
Hazardous decomposition products	None		
Hazardous polymerisation	Will not occur		
SPILL OR LEAK PROCEDURES			
In normal use there is no risk of leakage. If batteries are abused, this may lead to the leaking of a caustic alkaline solution which will corrode aluminium and copper. The leak material should be neutralised using a weak acidic solution such as vinegar, or washed away with copious amounts of water.			
Contact should be avoided			
DISPOSAL			
Batteries must be disposed of according to current local regulations. Batteries should not be discarded with normal refuse.			
SPECIAL PROTECTION INFORMATION			
Respiratory	Not applicable		
Ventilation	Not applicable		
Protective clothing	Not applicable		
Other			

This sheet is deliberately left blank

Index

PROCESSOR SUPERVISOR HANDBOOK

Symbols

A

B

C

D

E

Error codes (Cont.)

F

G

O

P

R

S

S (Cont.)

S (Cont.)

T

U

V

W

W (Cont.)