invensys

Foxboro

PAC Systems

# T820 Human Machine Interface

Handbook

HA029131 Issue 5

November 2012

# Declaration of Conformity

| | |
|---|---|
| **Manufacturer's name:** | **Eurotherm Limited** |
| **Manufacturer's address:** | **Faraday Close, Worthing, West Sussex, BN13 3PL, United Kingdom** |
| **Product type:** | **Graphical Display Unit** |
| **Models:** | **T820 (Status level A1 and above)** |
| **Safety specification:** | **BS EN61010-1** |
| **EMC emissions specification:** | **BS EN61326** |
| **EMC immunity specification:** | **BS EN61326** |

Eurotherm Limited hereby declares that the above products conform to the safety and EMC specifications listed. Eurotherm 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: WRDavis Dated 9th June 2006

Signed for and on behalf of Eurotherm Limited
William Davis
(General Manager)

CE

IA249986U710 Issue 1 Jun 06 (CN22344)

# Restriction of Hazardous Substances (RoHS)

**Product group** T820

**Table listing restricted substances**

Chinese

限制使用材料一览表

| 产品 | 有毒有害物质或元素 | | | | | |
|---|---|---|---|---|---|---|
| T820 | 铅 | 汞 | 镉 | 六价铬 | 多溴联苯 | 多溴二苯醚 |
| 印刷线路板组件 | X | O | X | O | O | O |
| 附属物 | O | O | O | X | O | O |
| 显示器 | O | O | O | O | O | O |
| O | 表示该有毒有害物质在该部件所有均质材料中的含量均在SJ/T11363-2006标准规定的限量要求以下。 | | | | | |
| X | 表示该有毒有害物质至少在该部件的某一均质材料中的含量超出SJ/T11363-2006标准规定的限量要求。 | | | | | |

English

Restricted Materials Table

| Product | Toxic and hazardous substances and elements | | | | | |
|---|---|---|---|---|---|---|
| T820 | Pb | Hg | Cd | Cr(VI) | PBB | PBDE |
| PCBA | X | O | X | O | O | O |
| Enclosure | O | O | O | X | O | O |
| Display | O | O | O | O | O | O |
| O | Indicates that this toxic or hazardous substance contained in all of the homogeneous materials for this part is below the limit requirement in SJ/T11363-2006. | | | | | |
| X | Indicates that this toxic or hazardous substance contained in at least one of the homogeneous materials used for this part is above the limit requirement in SJ/T11363-2006. | | | | | |

Approval

| Name: | Position: | Signature: | Date: |
|---|---|---|---|
| Martin Greenhalgh | Quality Manager | Martin Greenhalgh | 22nd March 2007. |

# Contents

T820 Handbook

*Intentionally left blank*

# CHAPTER 1 INTRODUCTION

The T820 Human Machine Interface (HMI) is a monochrome (black and white) graphical 128 x 64 pixel LCD module. It can be supplied as an interface to either,

- send and receive information, as a Control unit, allowing it to control and manage the I/O subsystem
- only receive information, as a Display unit, allowing it to display selected data from the I/O subsystem

*Note. Defined actions and information can be displayed via User generated pages, created using the User Screen Editor, see User Screen Editor Help (Part no. HA260749U005).*

As a Control unit it will hold a control strategy, and all other required files on the Compact Flash Memory Card. It can then operate as a controller, interacting with I/O subsystems and other instruments in the control system. It can also display information from the control system using customer generated User Screens. It is capable of communicating via Ethernet using LIN protocol, Modbus-TCP or RS485/422 Modbus.

As a Display unit it will connect to a T2550. This will allow the T2550 to be configured from the T820 panel interface and display information from the control system in the same manner as a Control unit.

## 1.1 MANUAL CONTENTS

This manual is divided into the following chapters:

| | |
|---|---|
| Chapter 1. | Introduction |
| Chapter 2. | Installation (installation and commissioning of the instrument) |
| Chapter 3. | Getting Started (step-by-step start-up or re-start instructions and User Interface explanation) |
| Chapter 4. | Configuration (overview of the configuration tools) |
| Chapter 5. | Operation (overview of the Operator tasks) |
| Chapter 6. | Control and Automatic Tuning (explaining Control Loop configuration) |
| Chapter 7. | Management (overview of the Engineering tasks, including managing applications, editing setpoint programs, and supervising the day-to-day operation and monitoring of the instrument) |
| Chapter 8. | Task Organisation and Tuning (explaining tasks and tuning) |
| Chapter 9. | Event Log (explaining the Event Log facility) |
| Chapter 10. | Data Management (explaining the data recording and archiving of Data Recording files, .uhh) |
| Chapter 11. | Setpoint Programming (explaining Setpoint Programming tools and files) |
| Chapter 12. | Service (step-by-step Preventive Maintenance instructions for prolonging the life of the instrument) |
| Appendix A. | Technical Specifications (a technical description of the individual parts, hardware and software) |
| Appendix B. | Customisation (step-by-step instructions for customising the Built-in pages and a list of available UNICODE Latin-1 characters) |
| Appendix C. | Terminal Configurator (overview of the instructions for connecting and using the Terminal Configurator) |

## 1.2 OTHER INFORMATION SOURCES

For details of Local Instrument Network (LIN) based Function Blocks, their parameters and input/output connections refer to the *LIN Blocks Reference Manual* (Part no. HA082375U003) which explains how control strategy function blocks are selected and interconnected etc. The creation, monitoring and Online Reconfiguration of LIN Databases and LIN Sequential Function Charts (SFCs) is described in the *LINtools Help* (Part no. RM263001U055). The *ELIN User Guide* (Part no. HA082429) gives full details of installation, and how to configure an ELIN network, including setting the IP address using the instruments internal configurator. Modbus communications, including the the interface type, can be configured, using the Modbus Tools software, see *Modbus Tools OnLine Help* (Part no. HA028988).

Any specifically required T820 PageSets, which includes individual Pages for each required screen, can be designed and generated using the User Screen Editor software. All appropriate information is available in the *User Screen Editor Help* (Part no. HA260749U005).

Defined areas of the T820 can also be restricted by allocating the required access to the appropriate Users via Security Manager. All appropriate information is available in the *Security Manager Help* (Part no. RM028131).

*Note. If you do not possess any document stated, please contact your distributor.*

## 1.3 THE T820

The unit comprises the monochrome (black and white) graphical 128 x 64 pixel LCD screen, and 25 buttons, including cursor, and navigation, numeric and user-definable keys.

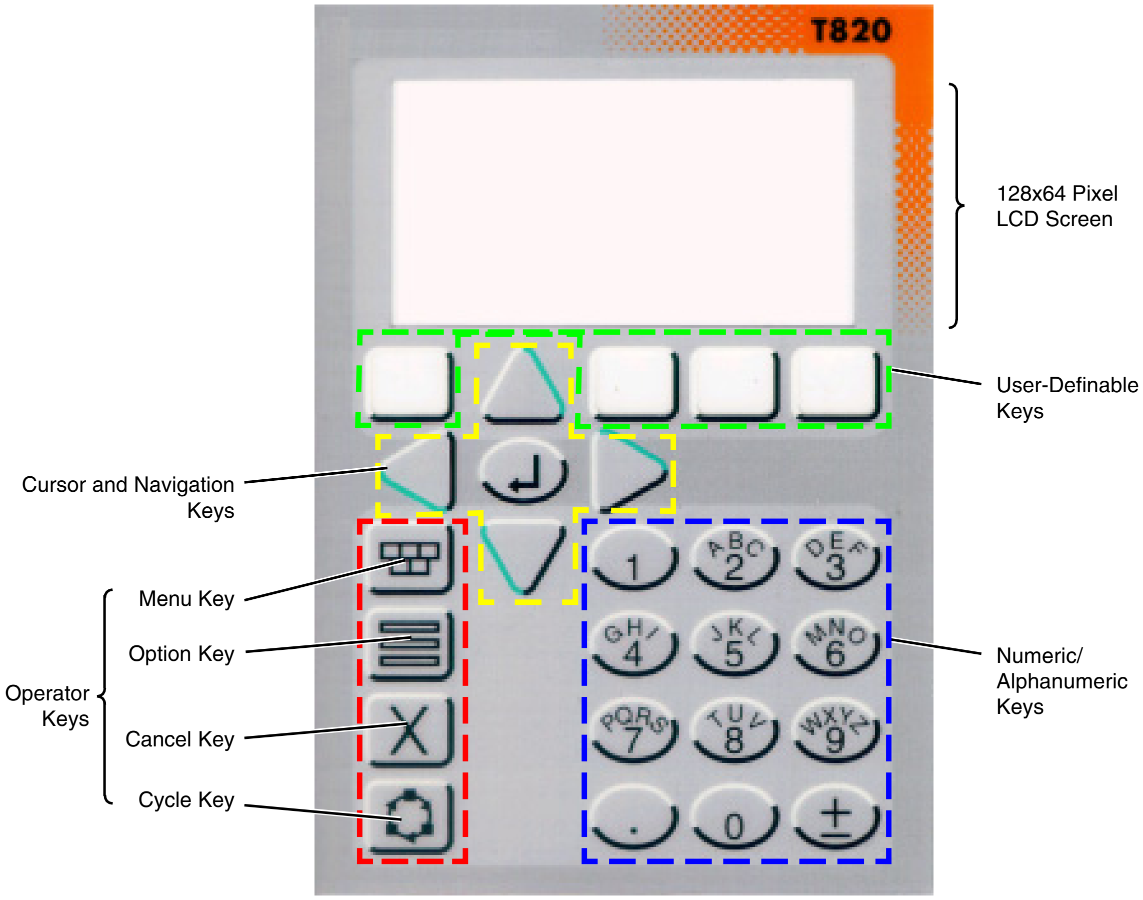



Figure 1.3 T820 Human Machine Interface

### 1.3.1 Main features

The main features of the T820 are as follows:

#### LIN

The Local Instrument Network (LIN) is a collection of LIN instruments, and LIN communications, etc. that together form the control system.

#### LIN COMMUNICATION

The Local Instrument Network (LIN) communications is our proprietry communications system used to connect each LIN instrument in the network.

#### ELIN COMMUNICATION

ELIN communication is the LIN communications system transported via Ethernet. It allows peer-to-peer communications between T820's and the wider network via a standard Ethernet infrastructure, see *Connections and Wiring* section.

#### REDUNDANT POWER SUPPLY CONNECTION

This instrument is capable of supporting a Redundant Power Supply Connection, disabled (default). Remove the wire link to enable Redundant Power Supply Connection.

It supports data in SRAM and the Real-Time Clock for approximately 1.5 years discontinuous use with an internal battery.

#### CONFIGURATION

Continuous strategies and Sequences are configured/downloaded/monitored with LINtools, the recommended LIN configuration tool.

#### ST USER ALGORITHMS

Special ACTION function blocks support user algorithms written in Structured Text (ST) and are well-suited to implement plant logical devices.

#### SEQUENTIAL FUNCTION CHART (SFC)

The Sequential Function Chart (SFC) is the graphical way LINtools (the recommended Configuration tool) represents a LIN Sequence (.sfc). A Sequence is employed when the process being controlled by the LIN Database (.dbf) can adopt several distinct states - e.g. 'Starting Up', 'Full Running', 'Shutting Down', etc. A LIN Sequence is a program that runs in a LIN instrument, in conjunction with a LIN Database. It interacts with its associated LIN Database by writing new values to specified LIN Database fields, in response to changes in the values of other specified LIN Database fields.

#### BLOCK SUPPORT

Most standard LIN function blocks are supported. Special diagnostic blocks are available for hardware and software status reporting, refer to *Diagnostics blocks section* for a list of typically required diagnostic blocks, and the *LIN Blocks Reference Manual* (Part no. HA082375U003) for a full description of each individual function block.

#### DATA MANAGEMENT

Automatic data recording and archiving, providing Data Recording files, .uhh, to store recorded values from defined parameters selected from the database in the instrument. The .uhh files can be automatically archived to defined FTP Servers, and then displayed as a charts using Review.

### 1.3.2 FEATURES (Cont.)

## EVENT LOGGING

Automatic event logging, providing an ASCII text file to record and store individually time stamped events generated in the instrument, and provide a means of indicating the impact of an event on the system.

## MODBUS COMMUNICATIONS

Modbus communication supported via both Serial and TCP interface types, see *Installation* section.

*Note. Always use the Modbus Tools software to configure the T820 Modbus parameters, see Modbus Tools Help (Part no. HA028988).*

## MODBUS GATEWAY FACILITY

The Modbus Gateway facility provides a Modbus interface to the LIN Database. By using the techniques of LIN function block caching, the Modbus Gateway facility can access data in other nodes distributed on the LIN, as well as LIN function blocks in the local LIN Database.

It can support more than one Modbus Gateway facility and is configured by the GW_CON block. Each Modbus Gateway facility is defined in one GW_CON block, e.g. it may require 3 GW_CON function blocks, one for each Modbus Gateway facility.

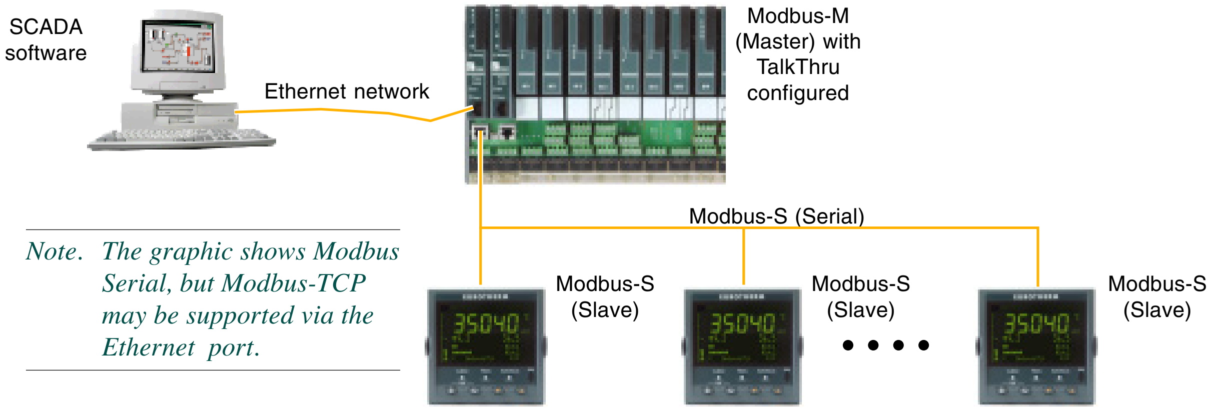
This instrument can be configured to operate in Modbus Master mode. It can read and write values, to and from a third party (Modbus communicating) device operating in Modbus Slave mode. It can support more than one Modbus Gateway facility which is configured using the GW_CON block, i.e. communications with more than one third party device configured to operate in Modbus Slave mode is permitted. Each Modbus Gateway facility is defined in one GW_CON block, e.g. it may require 3 GW_CON function blocks, one for each Modbus Gateway facility (instrument). However, when this instrument is operating as a Modbus Slave, it only allows a PLC or supervisory system configured as a Modbus Master to access values in the LIN Database.

*Note. For details concerning the Modbus communications protocol and interface types, see the Communications Manual (Part no. HA028014).*

## TRANSPARENT MODBUS ACCESS (TMA OR TALKTHRU).

Transparent Modbus Access (TMA), otherwise known as TalkThru. This facility enables the iTools software package, see *iTools User Manual* (Part no. HA026179) or other third party Supervisory Control And Data Acquisition (SCADA) software to inspect and edit specific I/O subsystem parameters while connected to an instrument configured to operate in Modbus Master mode. Talking-through a Modbus Master to access the I/O subsystem parameters can be slower than if actually connected directly because the Modbus Master is interleaving the iTools messages with its own data.

*Note. Specific function codes have been allocated to enable the TalkThru facility.*




*Note. The graphic shows Modbus Serial, but Modbus-TCP may be supported via the Ethernet port.*

Figure 1.3.2 Example of Transparent Modbus Access (TMA or TalkThru) configuration

# CHAPTER 2 INSTALLATION

This chapter is intended for use by those responsible for the installation and commissioning of the T820 and consists of the following sections:

- Safety and Electro-Magnetic Compatibility (EMC) information (section 2.1)
- Unpacking (section 2.2)
- Mechanical layout and installation (section 2.3)
- Setting the communication options (section 2.4)
- Connections and wiring (section 2.5)

## 2.1 SAFETY AND ELECTRO-MAGNETIC COMPATIBILITY (EMC) INFORMATION

Please read this section before installing the T820.

The T820 is designed to meet the requirements of the European Directives on Safety and Electro-Magnetic Compatibility (EMC). 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

The T820 conforms with the essential protection requirements of the EMC Directive 89/336/EEC, amended by 93/68/EEC. It also satisfies the emissions and immunity standards for industrial environments.

To ensure compliance with the European EMC directive, BSEN61326, certain installation precautions are necessary as follows:

- General guidance. For general guidance refer to the *EMC Installation Guide* (Part no. HA025464).
- 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.
- Communications wires. If the communications wires cover a distance greater than 30metres or leave the building make sure they are screened and connected to the connector shell at one or both ends to ensure immunity to surge events.

*It is recommended that...*

*the Instrument Chassis is earthed to provide the best Electro-Magnetic Compatibility (EMC) performance.*

## 2.1.2 Installation safety requirements

Caution

In order to comply with the requirements of BS EN61010, the voltage applied across terminals may not exceed those terminals' isolation voltage. For terminals specified as having no isolation, the maximum permissible voltage is 30V ac or 50 V dc.

### PERSONNEL

Installation must ONLY be carried out by qualified personnel.

### POWER ISOLATION

The installation must include a power isolating switch or circuit breaker. This device should be in close proximity (1 meter) to the controller, within easy reach of the operator and marked as the disconnecting device for the instrument.

### CONDUCTIVE POLLUTION

Electrically conductive pollution (e.g. carbon dust, water condensation) must be excluded from the cabinet in which the controller is mounted. To secure a suitable atmosphere in conditions of conductive pollution, fit an air filter to the air intake of the cabinet. Where condensation is likely, include a thermostatically controlled heater in the cabinet.

### VENTILATION

Ensure that the enclosure or cabinet housing the unit provides adequate ventilation/heating to maintain the operating temperature of the instrument.

### PRECAUTIONS AGAINST ELECTROSTATIC DISCHARGE

Caution

Circuit boards inside the unit contain components which can be damaged by static electrical discharge. Before any circuit board is removed or handled, all electrostatic precautions must be observed, by ensuring that the handler, the instrument and the circuit board are all at the same potential.

## 2.1.3 Keeping the product safe

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

### MISUSE OF EQUIPMENT

If the equipment is used in a manner not specified in this handbook or by the distributor, the protection provided by the equipment may be impaired.

### SERVICE AND REPAIRS

Except for those parts detailed in the Service section, this unit has no user-serviceable parts. Contact the nearest manufacturer's agent for repair.

## 2.2 UNPACKING

The unit 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**

Circuit boards inside the unit contain components which can be damaged by static electrical discharge. Before any circuit board is removed or handled, all electrostatic precautions must be observed, by ensuring that the handler, the instrument and the circuit board are all at the same potential.

### 2.2.2 Package contents

The unit 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.

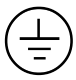
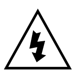
#### PRODUCT LABELLING

Product labelling includes:

- Sleeve label. On the back of the unit, showing all appropriate build information.
- Compact Flash Memory Card label showing firmware version, issue number, initial software licence and Ethernet-Mac Address.
- Safety earth symbol adjacent to safety earth stud.

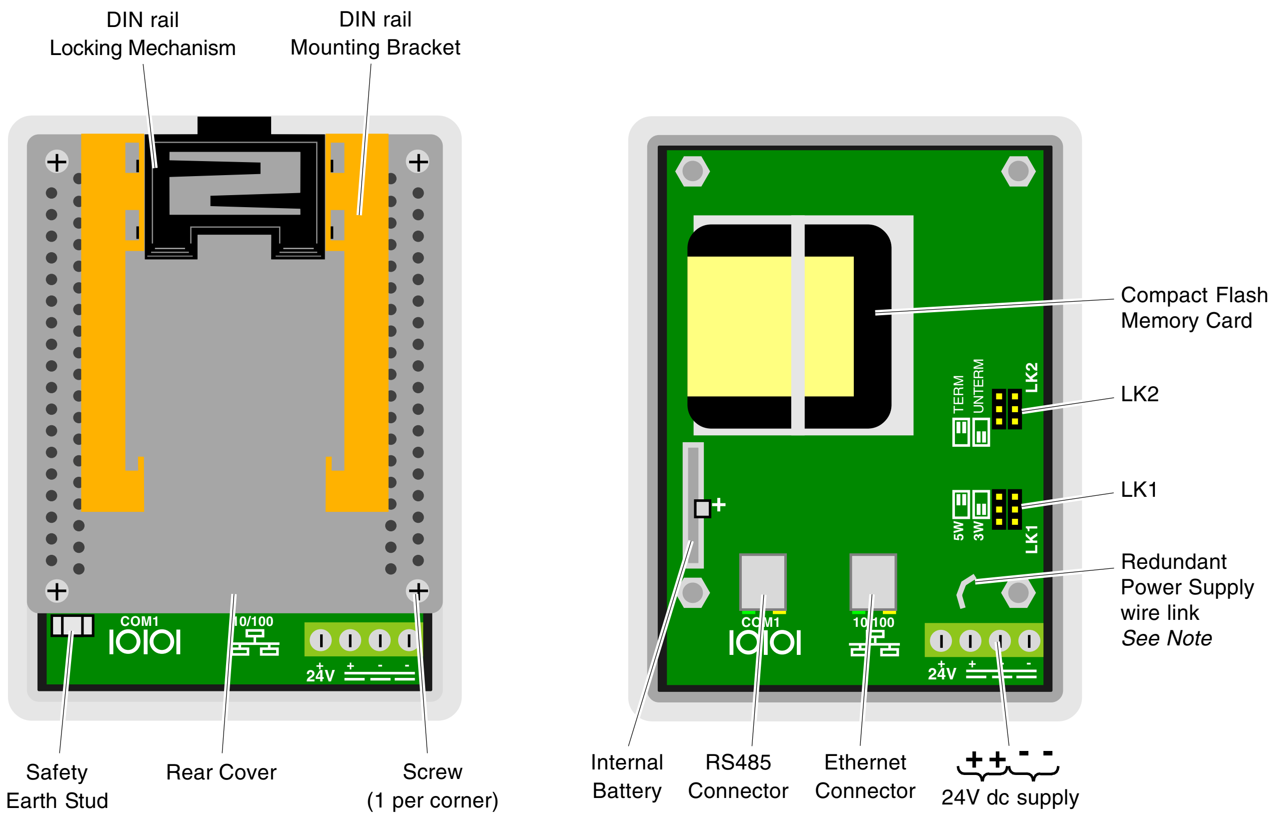
#### SYMBOLS USED IN THE LABELLING

One or more of the symbols in the table below, may appear on the labelling of the unit:

| Label | Function |
|---|---|
|  | Caution, refer to the accompanying documents |
|  | Functional (Ground/Safety) earth |
|  | Protective earth terminal |
|  | Risk of electric shock |
|  | 40 Year Environmently Friendly Usage Period |

## 2.3 MECHANICAL LAYOUT AND INSTALLATION

Figure 2.3 shows the T820 with the cover fitted and removed for clarity.




*Note. Set RS485 Link positions (LK1 and LK2) and, if required, remove the wire link to enable a Redundant Power Supply before installing.*

Figure 2.3 T820 - Rear View

## 2.3.1 Mount a T820

The T820 can be DIN rail or panel mounted.

### DIN RAIL MOUNTING

*Note. Remember to set RS485 Link positions (LK1 and LK2), and if required, remove the wire link to enable the connection of a Redundant Power Supply before installing, see Figure 2.3 parts reference.*

For DIN rail mounting, use symmetrical DIN rail to EN50022-35 X 7.5 or 35 X 15 mounted horizontally.

1. Mount the DIN rail, using suitable bolts.
2. Ensure that the DIN rail makes good electrical contact with the metal base of the enclosure.
3. Guide the T820 to the DIN rail, allowing the lower teeth of the DIN rail Mounting Bracket to rest behind the DIN rail itself.
4. Slowly and firmly, push the top of the T820 back until the DIN rail Locking Mechanism springs back into place. This is confirmed by an audible ‘Click’. The T820 is now mounted to the DIN rail.

*Note. When DIN rail mounted, the unit is rated IP20.*

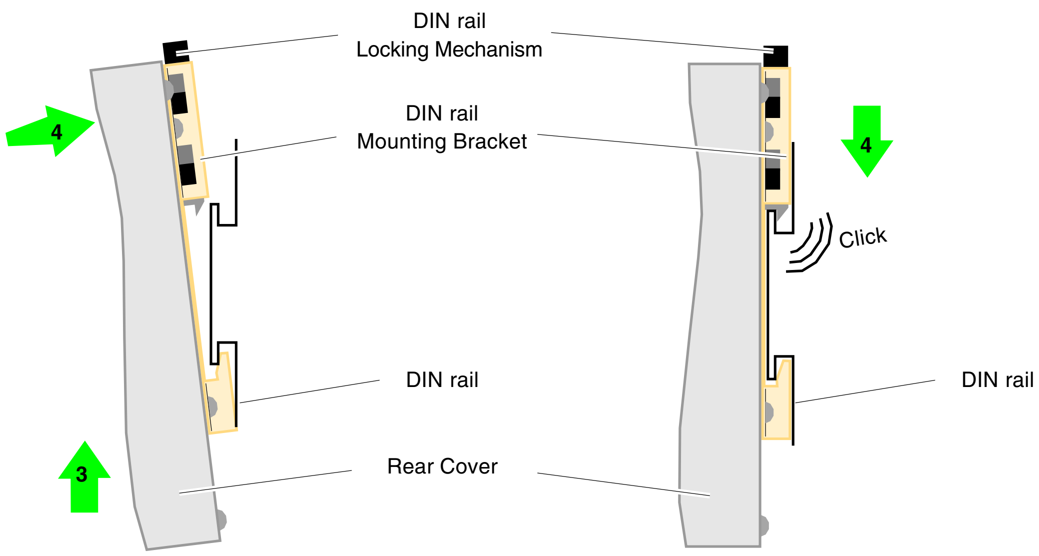



Figure 2.3.1 Mounting T820 to a DIN rail

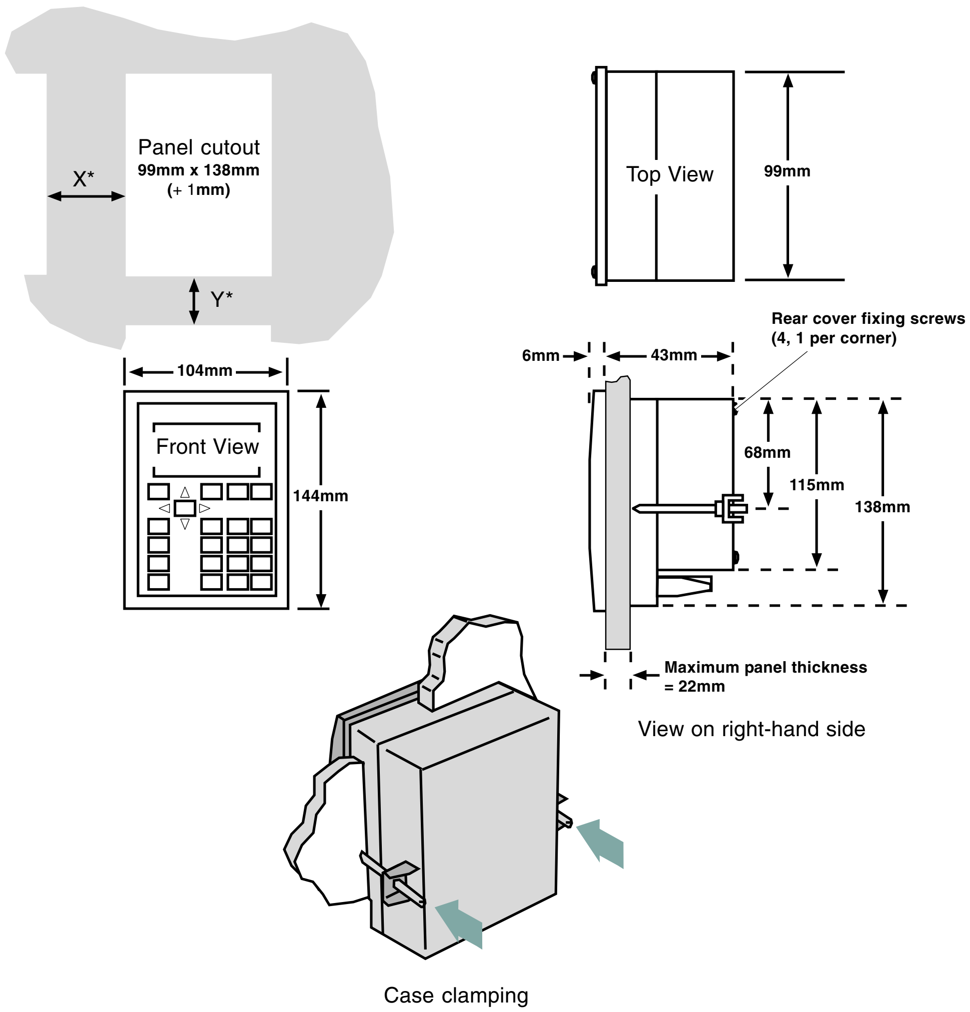
### 2.3.1 Mount a T820 (Cont.)

### DIRECT PANEL MOUNTING

*Note. Remember to set RS485 Link positions (LK1 and LK2), and if required, remove the wire link to enable the connection of a Redundant Power Supply before installing, see* Figure 2.3 *parts reference.*

1. Check that the mounting panel is no thicker than 25mm (typically for wood or plastic) and no thinner than 2mm (for steel).
2. In the mounting panel, cut an aperture 99mm x 138mm (+ 1mm). If more than one T820 is to be mounted in the panel, the recommended minimum spacings are as shown below.
3. From the front side of the mounting panel, insert the instrument (rear end first) through the aperture.
4. Support the rear of the T820 so that the seal is flat against the front of the mounting panel.
5. Insert two panel clamps into the opposing pair of rectangular apertures at the sides of the case.
6. Tighten the screws of the clamps sufficiently to hold the T820 firmly in position.

*Note. When mounted directly in a panel the unit is rated IP66 front of panel, and IP20, rear of panel.*




| Minimum recommended inter-unit spacing | | |
|---|---|---|
| **Clamp Position** | **X*** | **Y*** |
| Left and right | 14 mm | 7.5 mm |

Figure 2.3.1 Mounting the T820 to a Panel

## 2.4 SETTING THE COMMUNICATION OPTIONS

The RS485 communication options are configured using the 2 banks of 6 pin Links, LK1 and LK2, located within the T820. These links will configure either,

- 3-wire or 5-wire RS485 communications, using LK1
- Terminated or Unterminated, using LK2

*Note. Remember to set RS485 Link positions (LK1 and LK2), and if required, remove the wire link to enable the connection of a Redundant Power Supply before installing, see Figure 2.3 parts reference.*

### 2.4.1 Link Location

The links, LK1 and LK2, located within the T820, for setting communication options are only accessible (Figure 2.3) when the Rear Cover is removed.

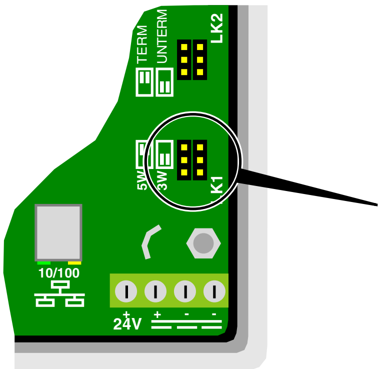
### 2.4.2 Link Functions

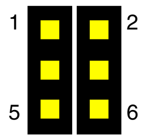
#### 5-WIRE/3-WIRE SELECTION (LK1)

The first bank of 6 pin Links, LK1, specify that RS485 3-wire/5-wire Modbus communications is required.

| Link | Position | Function |
|---|---|---|
| LK1 | 3-5 and 4-6 | RS485 3-wire communications. |
| LK1 | 1-3 and 2-4 | RS485 5-wire communications. |

*Note. Remember when this instrument is operating as a Display unit, LK1 pins 1-3 and 2-4 must be configured.*



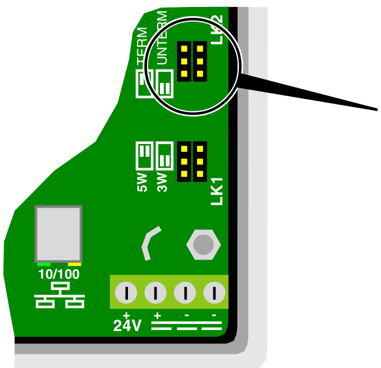



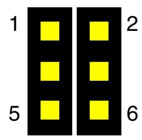

*Note. Pin numbers shown for clarity.*

#### TERMINATED/UNTERMINATED SELECTION (LK2)

The second bank of 6 pin Links, LK2, indicates that the RS485 link is internally terminated or unterminated.

| Link | Position | Function |
|---|---|---|
| LK2 | 3-5 and 4-6 | Unterminated RS485 communications. |
| LK2 | 1-3 and 2-4 | Terminated RS485 communications. |




*Note. Pin numbers shown for clarity.*

## 2.5 CONNECTIONS AND WIRING

If assembling the system yourself, refer to the *LIN/ALIN Installation & User Guide* (Part no. HA082429U005), the *ELIN User Guide* (Part no. HA082429) for advice on LIN connections and wiring and the *Communications Manual* (Part no. HA028014) for advice on Modbus connections and wiring.

Figure 2.5a and Figure 2.5b below, show a simplified overall connection diagram for an RS485 Serial or an Ethernet control system using an Ethernet hub/switch.

Category 5 cables may be used 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.

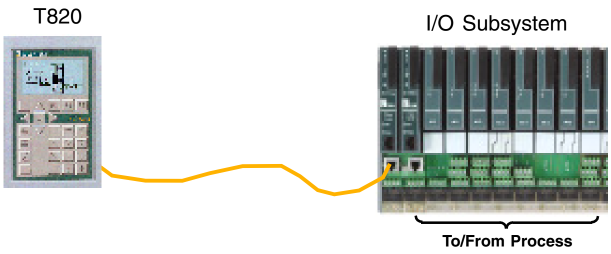



Figure 2.5a Typical RS485 Serial connection diagram

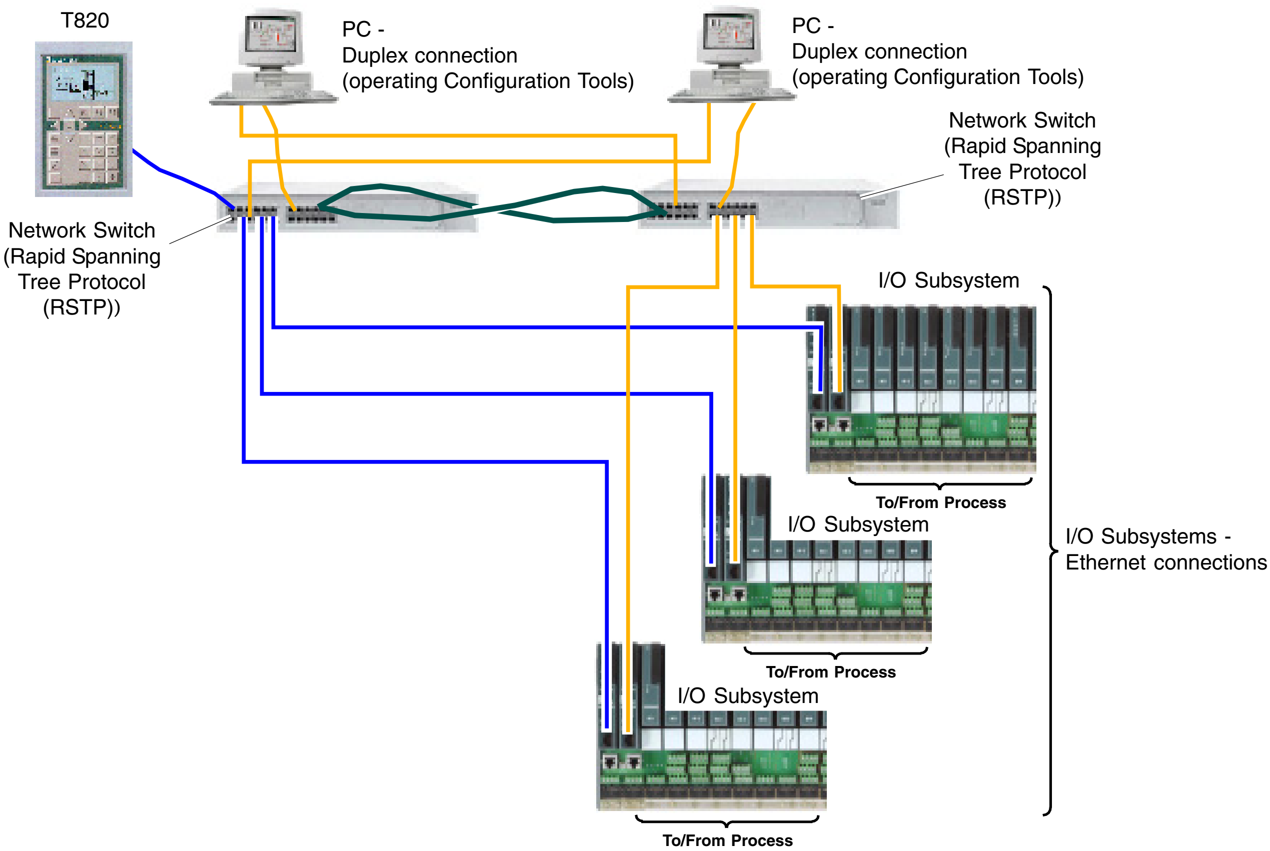



Figure 2.5b Typical Ethernet connection diagram

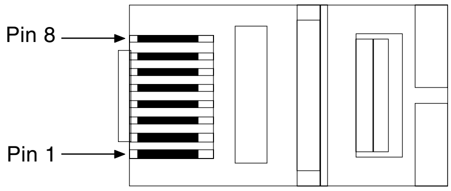
## 2.5.1 Communications

### ETHERNET (100/10 BASE-T) CONNECTOR

*Note. Ethernet communications are only available on Controller units.*

The Ethernet RJ45 connector includes 2 LEDs, a Yellow showing communication activity and a Green LED showing speed (On shows 100Mbps communication speed, Off shows 10Mbps communication speed).

Figure 2.5.1a shows the RS485 Ethernet RJ45 pinout connections.




Ethernet RJ45 plug: View on underside

| | Ethernet |
|---|---|
| Pin | Signal |
| 8 | Not Used |
| 7 | Not Used |
| 6 | RX- |
| 5 | Not Used |
| 4 | Not Used |
| 3 | RX+ |
| 2 | TX- |
| 1 | TX+ |
| Plug shroud to Cable screen | |

*Note. Modbus-TCP Communications will be supported using these pinouts.*

Figure 2.5.1a Pinouts for T820 Ethernet RJ45 type plugs

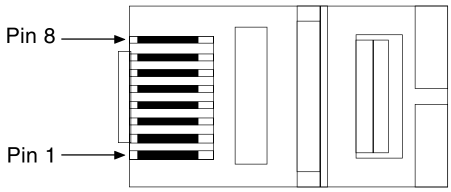
### RS485 SERIAL CONNECTOR

The RS485 Serial RJ45 connector includes 2 LEDs, a Yellow showing communication activity and a Green LED that is not used.

The RS485 Serial RJ45 connector may be used to communicate with a T2550, or to connect to a Modbus network for communicating with a variety of third-party serial devices.

Figure 2.5.1b shows the connector pinouts for Serial communications. These must be configured in conjunction with the Links, LK1 and LK2.

*Note. The T820 Display variant supports the 5-wire pinout only. This allows direct connection to a T2550.*




RS485 RJ45 plug: View on underside

| | EIA485 Serial | |
|---|---|---|
| Pin | 3-wire | 5-wire |
| 8 | N/A | TX+ (TxA) |
| 7 | N/A | TX- (TxB) |
| 6 | Gnd | Gnd |
| 5 | N/A | N/A |
| 4 | N/A | N/A |
| 3 | Gnd | Gnd |
| 2 | A | RX+ (RxA) |
| 1 | B | RX- (RxB) |
| Plug shroud to Cable screen | | |

Figure 2.5.1b Pinouts for T820 RS485 Serial RJ45 type plugs

## 2.5.2 Electrical Installation

*Note. Before carrying out any wiring, please read the Safety and EMC Information.*

Warning

Voltages of greater than 40V (peak), relative to the safety earth potential, must never be applied to any of the dc input terminals (positive or negative), as under such circumstances, the unit may become hazardous to the touch.

It is recommended that external 2 Amp fuses are fitted in each positive supply line.

**Caution**

Always ensure the power is isolated prior to removing the rear cover and before configuring the hardware.

A Polycarbonmonofluoride Lithium battery is also fitted in the unit. When fully charged this will maintain the SRAM and Real-Time Clock (RTC) data for approximately 1.5 years discontinuous use.

### POWER RING CONNECTION

The unit can be connected as part of a Power Ring to a maximum of 16A.

1. Connect the input supply to the + and - terminals, ensuring the correct polarity.
2. Connect the output from the remaining + and - terminals.

### REDUNDANT POWER CONNECTION

Redundant Power can be supplied via two independant sources.

*Note. Remove the Redundant Power Supply wire link, Figure 2.3, to enable Redundant Power Supply facility.*

1. Connect the first input supply to the + and - terminals. Ensure the correct polarity.
2. Connect the second input supply to the + and - terminals. Ensure the correct polarity.

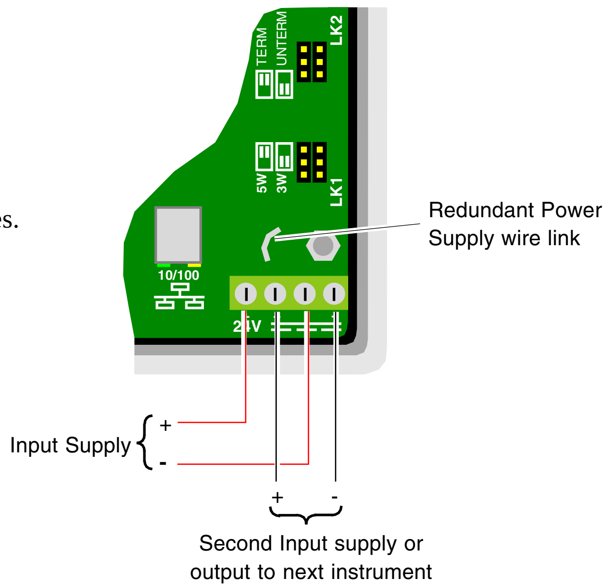



Figure 2.5.2 Power Supply connections

## 2.5.3 Safety earth connection

As shown in Figure 2.3, an earth stud connection is provided. 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.

# CHAPTER 3 GETTING STARTED

This chapter is for all prospective users of the instrument, including those responsible for installing and commissioning it. The chapter consists of the following sections,

- Switch-on, and the opening display, including the navigation keys (section 3.1)
- The Screen: Built-in pages and entry (section 3.2)
- The Pop-up menu (section 3.3)
- The Alarm page (section 3.4)
- The Access page (section 3.5)

## 3.1 SWITCH-ON AND THE OPENING DISPLAY

This section shows the instrument factory set defaults.

### 3.1.1 Switch on

The unit is not fitted with a power switch, so the switch-on arrangements depend upon the particular installation. Approximately 15 seconds, after switch-on an opening display appears that fills most or all of the screen area.

If this opening display fails to appear, check,

- the power supply
- with the Commissioning Engineer that the instrument powered up faultlessly when finally installed and commissioned.

After this, if the problem is still unresolved, recall the Commissioning Engineer.

### 3.1.2 The user interface

The user interface is open to customisation, either before operation or with the instrument taken out of service. For example the opening page (or the Home page) can differ, other pages can be changed, and User pages (User screens generated using the *User Screen Editor* (Part no. HA260749U005)) can be added. The size of the User generated pages can differ, and legends can also differ.

*Note. All built-in menus and pages supplied by the manufacturer have a pre-defined reference number, see Customising section. Built-in menus and pages can be displayed from User generated Pages by specifying the relevant reference number as a Structured Text (ST) action.*

The Operation and Management chapters of this manual describe the system of menus and pages, Built-in pages, supplied by the manufacturer, before any customisation by the user.

## 3.1.3 The standard interface

The Standard Interface consists of a ‘Screen’ and a group of ‘Navigation and Menu’ keys.

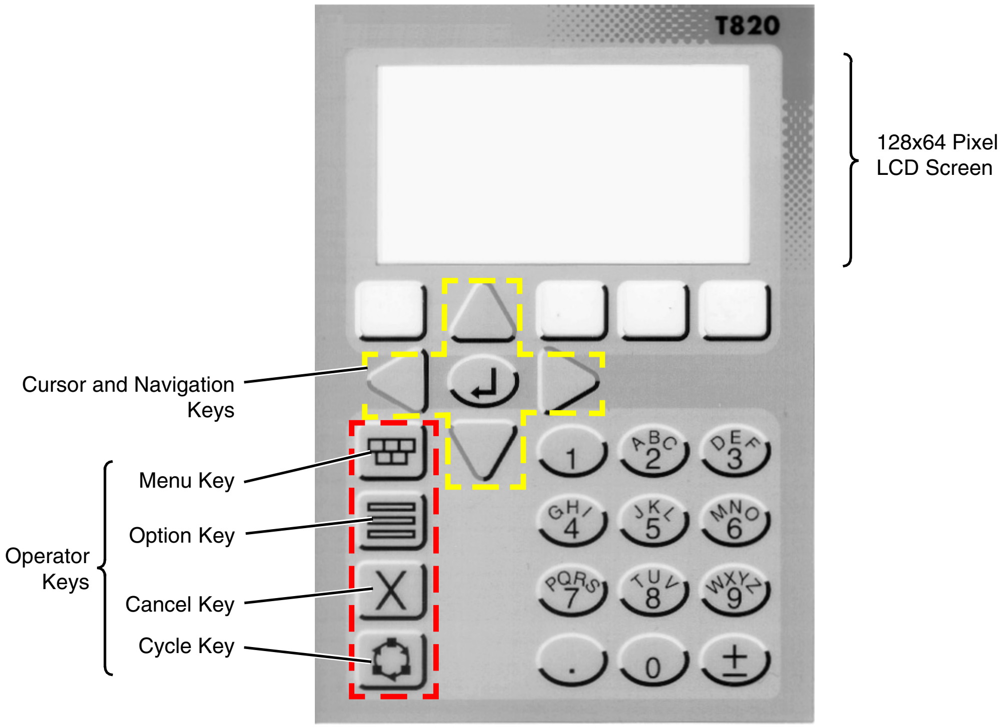



Figure 3.1.3 The Screen, and Navigation and Operator keys

### SCREEN

The Screen is used to display defined system values via User generated Pages created using the *User Screen Editor* (Part no. HA260749U005). It will also allow the User to select specific Menus for displaying the required instrument-specific and application-specific functions from a selection of Built-in pages, i.e. showing active and inactive alarms, and enabling a specific level of security.

### NAVIGATION AND MENU KEYS

The Navigation and Menu keys carry out the following functions:

| | |
|---|---|
| UP | Highlights an option one level above the currently selected item, moving vertically in a menu structure. |
| DOWN | Highlights an option one level below the currently selected item, moving vertically in a menu structure. |
| LEFT | Highlights an option one position to the left of the currently selected item. Also moves forward (left) between successive pages of tabular data. Action depends upon context. |
| RIGHT | Highlights an option one position to the right of the currently selected item. Also moves back (right) between successive pages of tabular data. Action depends upon context. |
| RETURN | Confirms the selection or configuration entry. Action depends upon context. |
| MENU | Displays the Menu (top-level) screen, showing the ACCESS, SYSTEM, and ALARMS pages. |
| OPTION | Displays a menu, or an extra set of keys, for options specific to the page on display. |
| CANCEL | Closes the currently displayed Menu screen |
| CYCLE | Displays the next page in the sequence to be scrolled through, e.g. the System Summary page has two pages, the other is displayed when this key is used. |

## 3.1.4 The opening display

*Note. If the system has been customised and configured with an overview screen, the opening page shows the User selected overview screen.*

The opening display is the System Summary page depicted in Figure 3.1.4, below.

```
System Summary
Instrument type: T820
Serial no: 197
Firmware: 1.1*
Node: 01
```

*Note. The Node number is configured as a Hex number. The 'A', 'B' or 'C' alphanumeric value is input by pressing the '2' numeric key the appropriate number of times, whereas the 'D', 'E', or 'F' alphanumeric value is input by pressing the '3' numeric key the appropriate number of times.*

Figure 3.1.4 System Summary page

The System Summary page confirms the order options that were specified for this instrument. Operation of the 'Down' or 'Cycle Screens' key calls a second page displaying any additional software options that are fitted.

Initially the opening display is 'Locked'. In this state, only the Navigation and Menu keys are active. Operation of the MENU key causes the opening 'Pop-up' menu of the Standard Interface to appear. Selecting the ACCESS option calls the Security Access page, see *The Access Page* section. This allows the current Security Access level of the instrument to be changed using an appropriate password.

## 3.2 THE SCREEN

The Screen is used to display system values via Pages created using the *User Screen Editor* (Part no. HA260749U005). It will also allow the User to select a Menu for displaying the required instrument-specific, application-specific and program-specific functions, i.e. displaying active and inactive alarms, configuring a specific level of security, and controlling the application.

*Note. System specific information will appear only if correctly configured Pages are generated using the User Screen Editor (Part no. HA260749U005 ).*

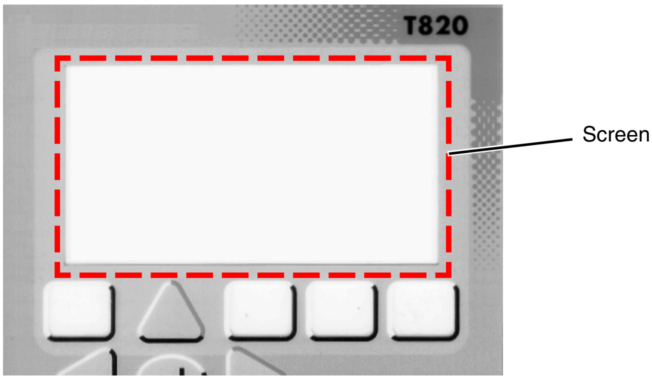



Figure 3.2 The Screen

### 3.2.1 Built-in pages

The Built-in pages are a menu system whose structure is hierarchical like a family tree. At the top is the Pop-up menu which offers a choice of submenus as depicted below.

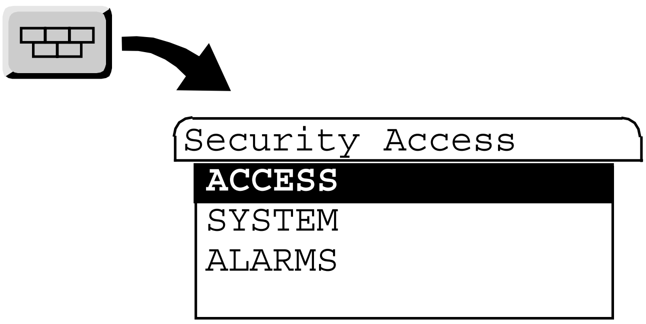



Figure 3.2.1 The Buitl-in page: top level menu

*Note. ACCESS, SYSTEM and ALARMS appear on every instrument, additional Options can appear only if configured using the User Screen Editor(Part no. HA260749U005).*

Below this menu level, there are further levels of functions that give users successively more detailed control of different aspects of programs, applications, and the instrument itself.

### 3.2.2 Built-in page entry

Information entry is via the Navigation keys to select an area on the screen and confirm the selection by using the Return key.

## 3.3 THE POP-UP MENU

*Note. In the following description of the Pop-up menu, and indeed of all the screen displays throughout this manual, it is important to note that almost everything is open to customisation. On any particular instrument the legends can be different from those shown here, or need not exist at all.*

*Throughout this manual what is called the Standard Interface is described. This is generated by the manufacturer, before any customisation by the user. The Minimum Interface, is generated by the minimum configuration necessary for the instrument to function.*

The Standard Interface, or Built-in pages, Pop-up menu consists of three options: ACCESS, SYSTEM and ALARMS, see Figure 3.2.1. With the display 'Locked', i.e. before a password is entered and access to the configuration menus is restricted, only ACCESS and SYSTEM are active. However, most users will operate the Standard Interface with additional User Screens generated using the User Screen Editor, see *User Screen Editor Online help* (Part no. HA260749U005).

In this case, with the display locked, ACCESS, SYSTEM, and HOME/USER SCREENS will be active.

From this menu, without a password, you can explore the menu systems and display information. The SYSTEM, and USER SCREENS displays are view-only at this locked stage; only ACCESS will respond fully to menu and key selections, to grant access, see the Gaining Access section.

The functions of each page:

| | |
|---|---|
| ACCESS | With a valid password, this page is the gateway to the functions-sets below that are needed to do the job. |
| SYSTEM | is the gateway to the system functions of the instrument (that is, the instrument-specific and application-specific functions, as opposed to the program-specific functions). |
| ALARMS | is the gateway to the alarm functions. |
| HOME/USER SCREENS | If configured, this returns you to the Home page. The Home page may be a single page, or it may be the root page of a user-written hierarchy of pages. If HOME/USER SCREENS is not configured, the System Summary page acts as a default Home page, displayed after a timeout. |

*Note. All built-in pages supplied by the manufacturer have a pre-defined reference number, see Customising section. Built-in pages can be displayed from User generated Pages by specifying the relevant reference number as a Structured Text (ST) action.*

Of the two or three further levels in the hierarchy, all are available to Engineers, but only some are available to Operators and Commissioning Engineers. This helps to improve usability, by hiding those facilities which are not currently required.

## 3.4 THE ALARM PAGE

The Alarm Page is used to display any alarm signals (triggered by abnormal conditions detected in the process under control), instrument alarms etc.

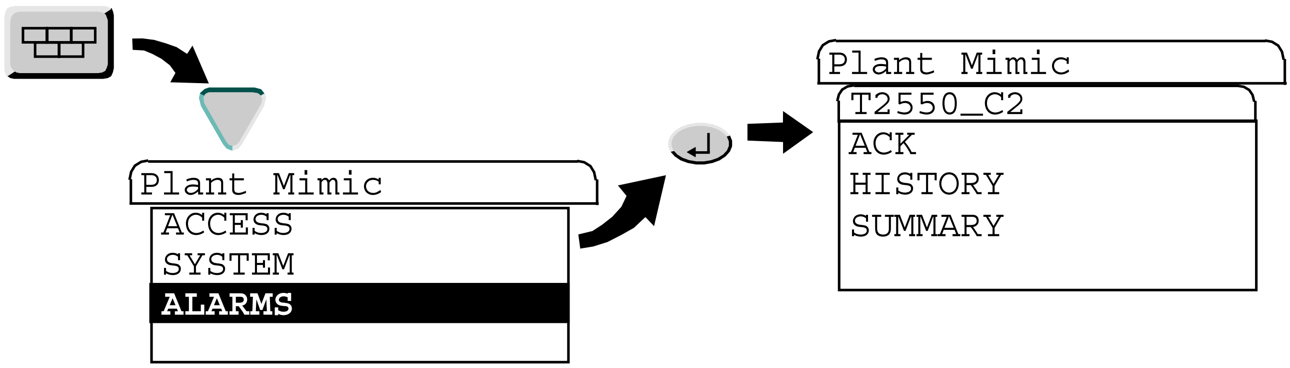



Figure 3.4 The Alarm Page

Alarms can be set to be latching or non-latching (auto acknowledging). Latching alarms are annunciated until acknowledged; auto-acknowledge alarms are annunciated until the alarm trigger returns to a non-alarm state. Decisions on which conditions should trigger an Auto-Ack Alarm rather than an Acknowledge Alarm (latching alarm) are made during configuration.

### 3.4.1 Alarm state indication

Alarm state is active or inactive, acknowledged or not. An alarm is triggered (becomes active) when the value it is monitoring moves outside a pre-set value or range of values. It becomes inactive when the signal returns to within the preset value or range of values. These values are set up during configuration.

To acknowledge an alarm, select the alarm using the Navigation keys and confirm the selection using the Return key. Again using the Navigation and Return keys, select Alarm Summary. This will display a list of all current alarms. Use the Navigation keys to select the Alarm and by pressing the Option key display the types of alarms that require acknowledging. Select the Alarm type and Acknowledge using the Return key. (Access permission needs to be set.)

*Note. If an alarm is selected, pressing the Right arrow displays additional alarm information, e.g. ACTIVE, when the alarm was set, CLEAR, when the alarm cleared, and ACK, when the alarm was acknowledged. Pressing again hides this additional information.In the Alarm Summary page, see Figure. 3.4.1, Alarm indicators flash until the alarm has been acknowledged, at which time they become illuminated steadily.*

In order to gain more details, the Alarm History page can be displayed, see Alarm History section.

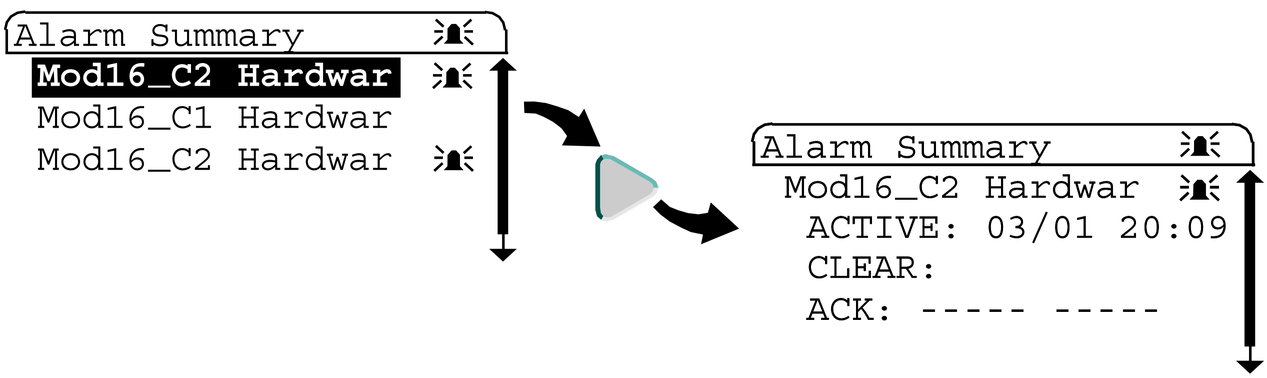



Figure 3.4.1 The Alarm Summary Page

### 3.4.2Responding to alarms

There are four possible responses to an alarm,

- do nothing
- get more information, by bringing up the Alarm History page, which is a list of past and current alarms and events
- acknowledge the alarm, using the Menu, Navigation, Option and Return keys.
- report and then remedy the abnormal condition

#### DO NOTHING

Doing nothing is acceptable when a non-serious abnormal condition did exist but it has now returned to normal, and that it has been acknowledged.

Doing nothing is also acceptable with an Alarm that is not the result of its having been manually acknowledged. In this case, the alarm will have been triggered by an Auto-Ack alarm, which does not require active intervention but will give information about a slightly abnormal condition.

#### ALARM HISTORY PAGE

Figure 3.4.2a shows the Alarm History page of the instrument.

The Alarm History page displays a list of alarm conditions and Events. By pressing the Right arrow, additional alarm information showing when they occurred, and if appropriate, when they were cleared or acknowledged. Events and other items which are not clearable or which cannot be acknowledged display ----- ----- in the CLEAR and ACK fields.

The Alarm History record starts in the first instance from when the instrument is powered up for the first time. Thereafter it is preserved through any automatic restarts called hot starts, see *Setting the Start-up strategy* section.

*Note. Each existing record is lost and a new one started whenever a new application database is loaded.*

The Standard model of the instrument can retain and display a total of 512 alarms or Events. Once these limits are exceeded the oldest item in each case is deleted when a new addition is made to the list.

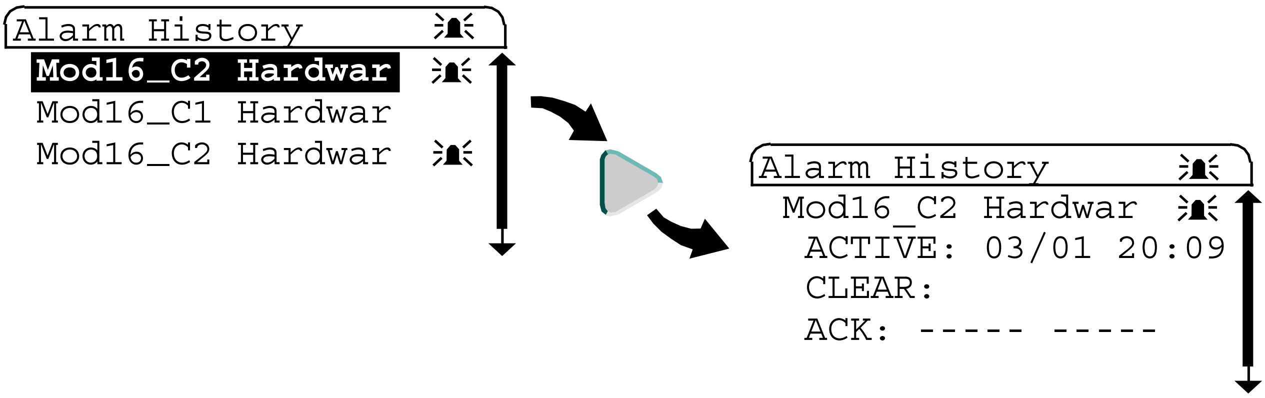



Figure 3.4.2a The Alarm History Page

### 3.4.2 Responding to alarms (Cont.)

#### ALARM ACKNOWLEDGEMENT

Unacknowledged alarms are made evident by the flashing of the alarm bell icon in the Alarm History page and the Alarm Summary page.

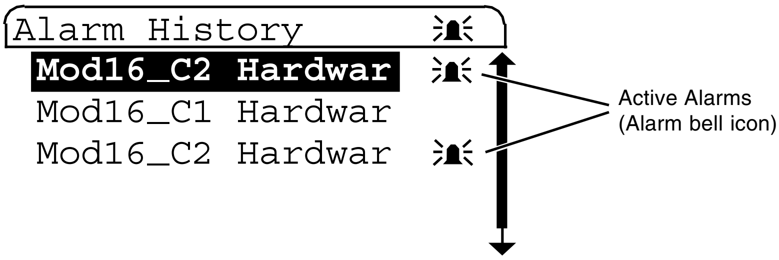



Figure 3.4.2b Alarm Indication

***IMPORTANT It is recommended that an alarm is acknowledged before any attempt is made to rectify the cause of the alarm.***

To acknowledge an alarm, select the alarm using the Navigation keys and Acknowledge this selecting the ACK option and confirm using the Return key.

*Note. If an alarm is selected, pressing the Right arrow displays additional alarm information, e.g. ACTIVE, when the alarm was set, CLEAR, when the alarm cleared, and ACK, when the alarm was acknowledged. Pressing again hides this additional information.*

#### MULTI-LINE DISPLAY

Operation of the Right arrow key toggles between single-line and multi-line working. Single line working is as described above, and as shown in Figure 3.4.2a. When in two-line working, each alarm has a second line showing the User ID of the user who was logged on at the time of the alarm, or event.

## 3.5 THE ACCESS PAGE

This area is displayed to allow a different level of security access to the T820. Each security access level can be password restricted to ensure the security of the T820. Selecting this option from the Built-in pages calls the first of the Access pages described in *Gaining Access* section.

### 3.5.1 Gaining access

The Unit is supplied with the standard access system in operation and is configured with the following types of Users,

- Locked (Disabled User)
- Operator
- Commission (Commissioning Engineers)
- Engineer

Each of these User types has what is known as a level of access to the facilities of the instrument, based upon the needs of the job, and they gain access to that level by typing in a password. The level of access is fixed for each type of User. That is to say, all Operators share the same password to their level of access; all Commissioning Engineers share the same password to their level of access; and all Engineers share the same password to their level of access. The Engineer-level password gives access to every facility in the instrument.

*Note: Only from Engineer-level can passwords be changed.*

The hierarchy of levels is LOCKED (lowest), OPERATOR (next lowest), COMMISSION (middle), ENGINEER (highest). You need a password to change up levels, but not to change down. No password is needed for Locked. What follows are step-by-step instructions on how to navigate the menu system to get access to your level of facilities. The assumption is that you have your password ready. Passwords are set and re-set by the Engineer.

*Note: For first-time access, immediately after commissioning, see the Controlling Access section.*

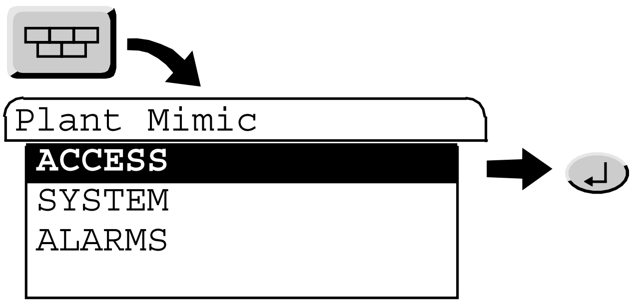
To gain access to a required level of functions,

1. With the Unit powered up, press the Menu key.

   The top level menu of the Built-in pages (figure 3.2.1) appears.

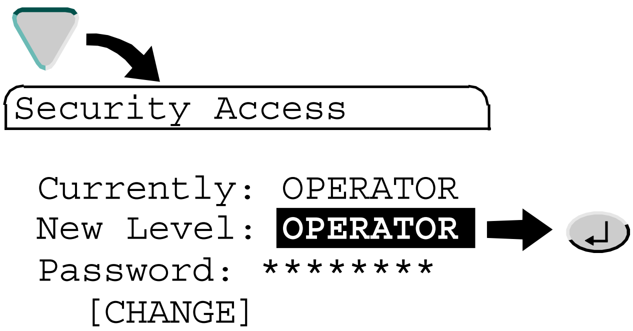
2. Select the ACCESS option and confirm using the Return key.

   The Security Access page appears.




3. Select the 'New Level' field, and again confirm selection using the Return key.

   The Security Access pick-list appears.



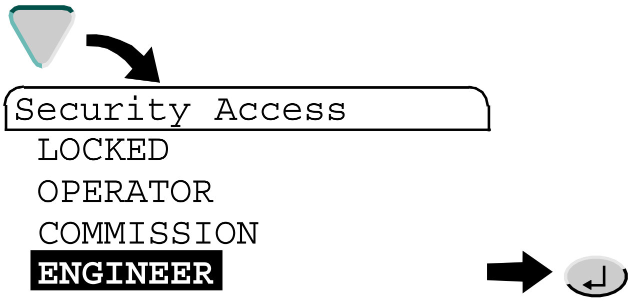
### 3.5.1 Gaining access (Cont.)

4. Using the Navigation keys select the required level of access,

   - LOCKED
   - OPERATOR
   - COMMISSION
   - ENGINEER

   Confirm using the Return key.

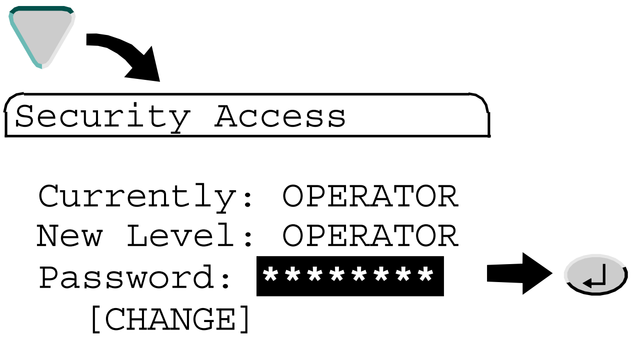
   The pick-list disappears, revealing the Security Access page again, showing the selected level (LOCKED, OPERATOR, COMMISSION or ENGINEER) in the 'New Level' field.




5. Select the Password field (shown as asterisks) using the Navigation keys and confirm using the Return key.

   The asterisks are removed and Password field starts to flash.

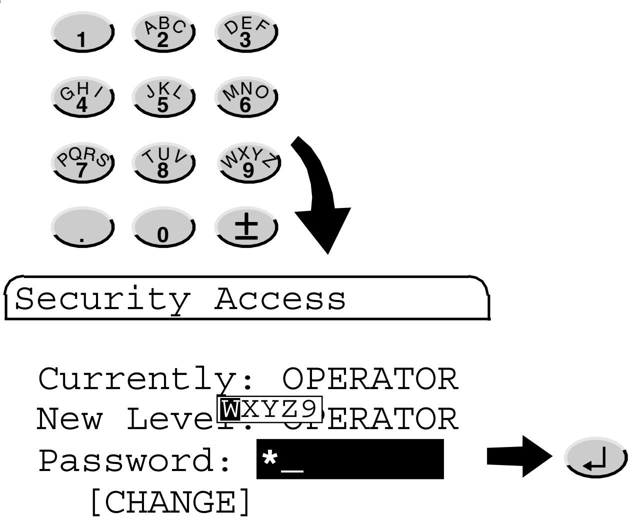
   *Note. A Password is not required when selecting a lower security access level, e.g. ENGINEER to OPERATOR.*




6. Enter each character/numeric of the required Password, using the alphanumeric keys. Each key press selects the next alphanumeric in the list. The selected alphanumeric is only entered in the password when the alphanumeric list disappears.

   *Note. A Password is entered following the same principles as the mobile phone, using multiple keys presses to select the letter or number required.*

   Erroneous characters can be deleted by selecting the relevant character and pressing the appropriate alphanumeric key.



### 3.5.1 Gaining access (Cont.)

7. Select the 'Change' field using the Navigation arrows and confirm using the Return key.

```
Security Access

 Currently: OPERATOR
 New Level: ENGINEER
 Password: ********
   [CHANGE]  [PASSWDS]
```

8. The display blanks momentarily, and returns showing the selected level (OPERATOR, ENGINEER or COMMISSION) in the 'Currently' field, as well as the 'New Level' field.

```
Security Access

 Currently: ENGINEER
 New Level: ENGINEER
 Password: ********
   [CHANGE]  [PASSWDS]
```

*Note. If not, an incorrect password must have been entered. Check that the password is correct for the level selected in step 4, and then repeat from step 3 onward.*

The unit now permits the selected level of configuration.

***IMPORTANT To ensure the security of the unit, it MUST never be unattended while ENGINEER security access has been selected.***

*Intentionally left blank*

# CHAPTER 4 CONFIGURATION

This chapter presents and describes the recommended Configuration Tools and issues concerning the instrument communication protocol.

*Note. Configuration and Ethernet functionality is only applicable to a Control unit. A Display unit will only show the LIN Database of the connected T2550 I/O Subsystem.*

The main topics of this chapter are:

- Tools: The Automatic Build and Configuration Tools (section 4.1)
- Automatic Build (section 4.2)
- LINtools (section 4.3)
- Modbus Configuration Editor (section 4.4)
- Instrument Properties Dialog (section 4.5)

## 4.1 TOOLS: THE AUTOMATIC BUILD AND CONFIGURATION TOOLS

Most LIN Database configuration of a Control unit will be done before despatch, using the LINtools configuration tool. However, at start-up a basic LIN Database and communications parameters can be automatically configured for the T820 instrument using the Automatic Build procedure. A basic Terminal Configurator, see *Terminal Configurator section*, is also resident within the Control unit. The instrument employs the standard LIN block structured approach.

Using the recommended LINtools program also allows the creation of new LIN Databases, including new function blocks, and the editing of existing configurations on-site and on-line, usually to accompany modifications to the processing plant. The *LINtools Help* (Part no. RM263001U055) should be referred to for details of the reconfiguration procedures using the LINtools program.

*Note. Refer to LIN Blocks Reference Manual (Part no. HA082375U003) for full details of the function blocks available for control strategies, and how to configure their parameters*

## 4.2 AUTOMATIC BUILD

An automatically generated LIN Database does not result in a complete, usable control strategy, and is dependant on the hardware configuration of the Instrument. However, it does provide the user with sufficient information, forming a good starting point for the user to begin building a more detailed control strategy.

An instrument operating as a Control unit, can automatically generate a basic range of diagnostic function blocks to assist in identifying and monitoring the instrument and system, generating and configuring appropriate function blocks to replicate the hardware.

The table below shows a list of diagnostic function blocks that are automatically generated as part of the automatically created LIN Database, if a LIN Database is not loaded and the power is isolated, then re-applied.

| Block Type | Block Name | Function |
|---|---|---|
| DB_DIAG | DDIAG_nn | Database diagnostics block. Shows actual and maximum resource levels of the LIN Database by the current software. Displayed parameter values are only valid at runtime. |
| EDB_DIAG | EDIAG_nn | External database diagnostics block. Shows connection information to one external database running in remote instruments and monitors the cached block update rate tuning algorithm. |
| EIO_DIAG | EIODI_nn | Ethernet I/O system diagnostic block. Shows the current state (Healthy/Unhealthy) of the expected and actual I/O modules at each site. It can display a maximum of 16 I/O sites on one screen. |
| ELINDIAG | ELIND_nn | ELIN diagnostics block. Statistics on the operation of the Ethernet Local Instrument Network (ELIN). |
| IDENTITY | IDENT_nn | IDENTITY diagnostics block. Identifies the instrument containing this function block. |
| LIN_DEXT | LDEXT_nn | LIN High-level diagnostics extension block. Statistics on the operation of the Local Instrument Network (LIN). |
| OPT_DIAG | ODIAG_nn | Options/Licence Control System diagnostics block. Shows the user system attributes that may impose some limit of operation, or cause a licence violation alarm. The block is not essential to the LIN Database, and can be added while online. |
| PNL_DIAG | PNLDI_nn | Panel diagnostic block. Shows information about the panel. |
| SFC_DIAG | SFCDI_nn | Sequence diagnostics block. If SFC is enabled, this block shows actual and maximum resource levels of the sequence by the current software. It displays parameter values that are only valid at runtime. |
| TACTTUNE | TTUNE_nn | Tactician tuning block. System task monitoring in priority order. |
| USERTASK | UTASK_nn | User Task diagnostic block. Control strategy task performance monitoring. |

*Note. 'nn' denotes the numeric instance of that specific block type.*

Table 4.2 Automatically generated diagnostic function blocks

### 4.2.1 Preparing for the Automatic Build

Before the Automatic Build process can be started, any active application must be stopped and unloaded. Only then should power be isolated and re-applied, initiating the automatic generation of a LIN Database '_auto.dbf', and corresponding '_auto.run' file. A '*.run' file is automatically generated each time a LIN Database is started, but is always transparent to the user. The automatically generated LIN Database will run automatically, but is not saved to the filesystem.

*Note. A Control unit requires that both Hot Start and Cold Start parameters are set OFF.*

Use the 'Save as' command at the instrument level of the 'manufacturers Network Explorer' or the 'Save' command in the 'Terminal Configurator' to save an automatically generated LIN Database.

*Note. The only constraints on the name of the LIN Database is that it must be a unique 8-character string, although it is recommended that the name of the LIN Database is the Instrument Type and the LIN Address, i.e. T820_0f.*

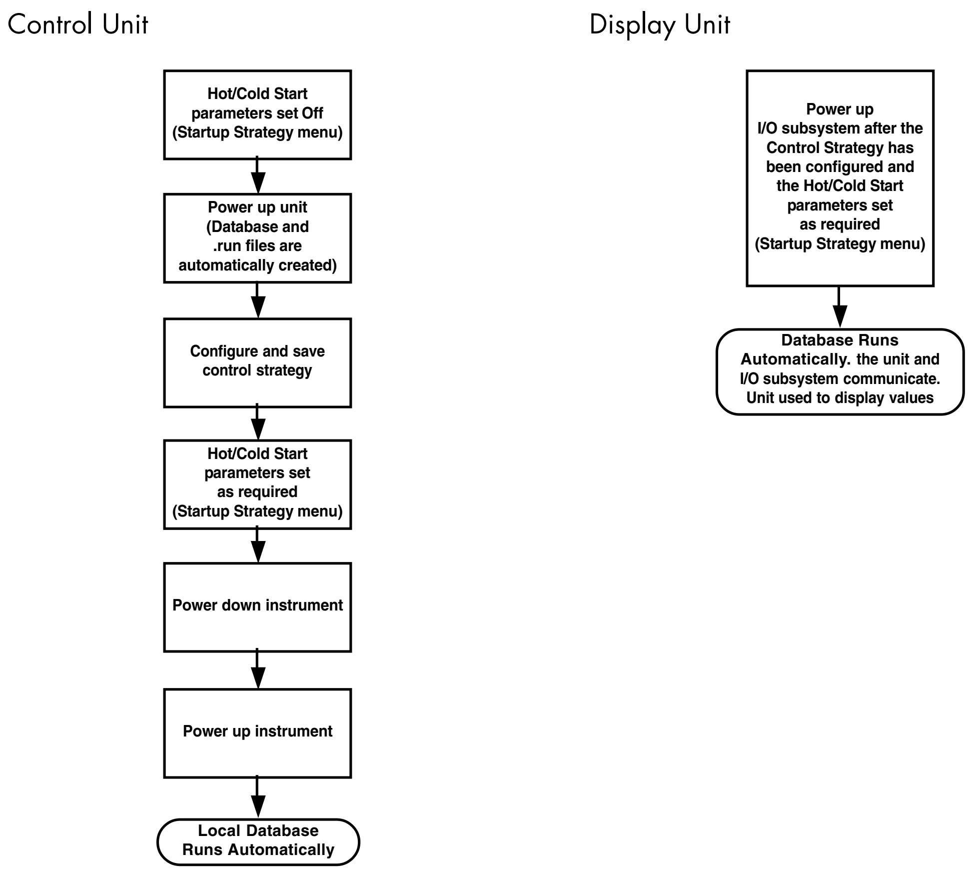



Figure 4.2.1 Automatic LIN Database generation routine

## 4.3 LINTOOLS

The LINtools program provides the user with a view of the control strategy components that compose the configuration of a single device, and an easy way to manage those components. There may be more than one of each component, but not always all component types.

- I/O Modules LIN Database (file extension .dbf)
- LIN Databases (Function Block Diagram - FBD, file extension .dbf)
- Sequences (Sequential Function Chart - SFC, file extension .sdb)
- User Screen PageSets (file extension .uxp and .ofl)
- Action block methods (Structured Text - ST, and Ladder, file extension .stx and .sto)
- Modbus Gateway configurations (file extension .ujg and .gwf)

In summary, LINtools

- Provides a comprehensive view of the instrument configuration
- Provides Build and Download functions
- Assigns LIN names and node addresses to external LIN Databases

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

- Provides On-line Reconfiguration to a running LIN Database only

*Note. On-line Reconfiguration does not apply to other files, i.e. Modbus Gateway file (.gwf), Sequential Function Chart file (.sdb),or User Screen PageSets file (.ofl), etc..*

### 4.3.1 On-line Reconfiguration

On-line Reconfiguration of a unit may involve adding and editing function blocks and wires in a running control strategy. Changes, such as adding new function blocks and wires are automatically made as 'Tentative'. However, when using on-line reconfiguration, LINtools will not permit changes to certain fields of I/O function blocks unless specific conditions are acheived. To ensure that changes made to block fields do not impact on the running control strategy until the user decides, LINtools detaches the block from the control strategy, allowing changes to first be tried using the 'Try' function, and accepted using the 'Apply' function.

On-line Reconfiguration allows the user to make 'Tentative' edits to a running control strategy before applying changes. During on-line reconfiguration, the user can edit a LIN Database loaded in LINtools, and 'Try' changes in the instrument to ensure the changes have the desired affect. The user can then either 'Apply' these changes, making them permanent in both LINtools and the instrument; 'Discard' the changes (restoring the last saved data); or 'Untry' the changes (removing the changes from the live instrument, but retaining them in the PC, so that the changes can be altered in LINtools before again using the 'Try' command).

Caution

Any changes made directly to a running function block cannot be 'Tried/Untried', but are applied immediately (e.g. changing the value of a block's field).

In order to make 'Tentative' changes to a running function block, the user must choose to 'Unlink' that function block in LINtools, so any changes are not directly applied to the function block in a running control strategy. The user can then 'Try' the changes as normal. The instrument creates a new copy of the function block, with all of the changes, and runs it in place of the original. At this point the T820 instrument will be running the altered LIN Database, however, the original function block is still present in the LIN Database, so can be restored if 'Untry' or 'Discard' is selected). The user can also 'Re-link' the function block, discarding all changes made to it, by selecting 'Undo Unlink' on the function block.

During On-line Reconfiguration, using the Apply command will save changes in the running LIN Database, but any other files, i.e. Modbus Gateway file (.gwf), Sequential Function Chart file (.sdb),or User Screen PageSets file (.ofl), etc., that have been edited using the relevant Tools, or are dependant on the LIN Database at load, MUST be downloaded. However, after files have been downloaded and the control strategy is stable, either the application will have to be stopped and then loaded again, or the power to the instrument must be isolated and then re-applied.

***IMPORTANT*** ***On-line Reconfiguration only applies to LIN Database files, .dbf.***

## 4.3.2 Preparing to run LINtools

Getting ready to run LINtools consists of two main topics:

- Connecting the instrument to a PC
- Creating a Project folder

### CONNECTING TO A PC

The instrument LIN Database can be accessed over an Ethernet network via an Ethernet hub/switch. It connects the Ethernet port on the PC, to the Ethernet port (‘TCP’) of a Control unit, or via the T2550 I/O subsystem to a Display unit communicating via the Modbus network.

*Note. Instrument ports can be configured using the Instrument Properties dialog, see Instrument Properties Help (Part no. RM029278).*

### CREATING A PROJECT FOLDER

The use of LINtools is restricted only by the requirement of a Project folder (or Project Database) containing appropriate Network and Instrument folders. A New Project folder is created via the New Project wizard, started from the Start > Programs > ... > Project Folder > New Project command. Thereafter use the context-sensitive menus to create the required Network and Instrument folders.

*Note. ‘...’ indicates the file path of the installed software.*

Each Network folder represents a network and type defined via the New Network wizard and contains all the Instruments within it.

*Note. The Build command can be used at any time, but Networks must be built before Instruments.*

Each Instrument folder represents a type of instrument defined via the New Instrument wizard and contains all the files required for the successful operation of the control strategy by the instrument at the specified address.

*Note. Automatically generated LIN Databases can be saved to the correct Instrument folder using LINtools.*

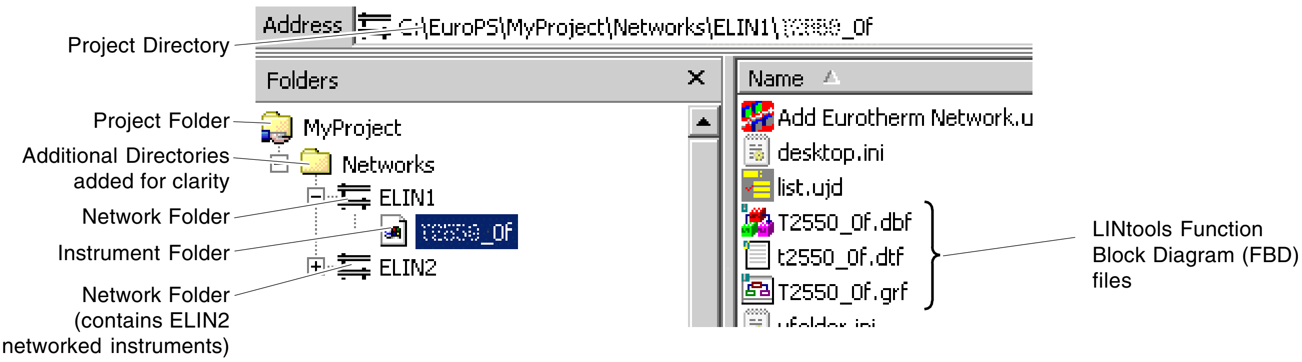



Figure 4.3.2 Project directory structure

## 4.3.3 Running LINtools

An empty LINtools instance can be started via

Start > Programs > ... > LINtools Engineering Studio

*Note. ‘...’ indicates the file path of the installed software.*

Use the ‘Open’ command to locate the required Instrument and then select the required file type and finally open.

- Alternatively, simply double-click the LINtools file from the required Instrument folder.

*Note. Refer to the LINtools Help (Part no. RM263001U055) for details of LIN Database configuration and On-line Reconfiguration procedures using the LINtools software.*

## 4.4 MODBUS CONFIGURATION EDITOR

Any LIN instrument may be configured as a Master communicating to one or more Modbus instruments, and alternatively may be configured as a Modbus Slave instrument, see *Communications Manual* (Part no. HA028014) for Modbus details.

The *Modbus Tools* software application defines the communications between LIN and Modbus instruments.

The Modbus configuration data is defined in a Modbus Gateway File (.gwf). This is downloaded with the LIN Database (.dbf) into a LIN instrument. The data in the .gwf is used to define the transfer of data between LIN and Modbus instruments.

This data defines,

- The operating mode (i.e. Modbus Master or Modbus Slave)
- The serial line set-up (or TCP)
- The mapping between fields in function blocks and the registers of a Modbus instrument
- How field values are transferred between instruments. For example which Modbus functions to use, the addresses of Modbus registers and the format in which data is to be transferred.

### 4.4.1 Preparing to run Modbus Tools

The *Modbus Tools* software application requires the same preparation as LINtools, consisting of:

- Connecting the instrument to a PC.
- Creating a Project folder.

Use the *Modbus Tools Help* (Part no. HA028988) for details of Modbus Configuration procedures using the *Modbus Tools* software application.

### 4.4.2 Running Modbus Tools

An empty Modbus Tools file instance can be started via

Start > Programs > ... > LINtools Advanced > Modbus Tools

*Note. '...' indicates the file path of the installed software.*

Use the 'Open' command to locate the required Instrument and then select the required file type and finally open.

- Alternatively, simply double-click the LIN MODBUS Database file (.ujg) from the required Instrument folder.

# 4.5 THE LIN INSTRUMENT PROPERTIES DIALOG

This dialog provides the simplest method of ensuring the appropriate configuration of the unit is correct. It shows the setup options of a selected Instrument type and Version and permits existing instrument parameters to be uploaded from the instrument, configured and then downloaded to ensure all relevant instrument parameters are correctly defined.

Changing Instrument Options or Network settings of the instrument is achieved by editing parameters, see *Instrument Properties help* (Part no. RM029278).

## 4.5.1 Instrument Options

The Instrument Options page, displays the same editable settings as accessed via the Setup page. In the Instrument Properties dialog, this page contains all the controls used to define the Setup configuration, as follows.

- Startup

  These settings define how the unit will attempt to start after it has been powered down.

- Communications

  These settings define the communication parameters used by each port.

- Internationalise

  These settings define the Language and the Calendar display configurations.

- Panel

  These settings define the screen parameters, such as Time-out periods and Contrast.

## 4.5.2 Network Settings

The Network Settings, held in the network.unh file, display the same editable settings as accessed via the LINOPC Applet available from the PC Control Panel directory.

In the Instrument Properties dialog, this tab contains all the controls used to define the Communication configuration, as follows.

- IP

  These parameters define how the IP configuration will be allocated.

- LIN

  These parameters define the LIN protocol configuration.

- PR

  These parameters define the IP address of each instrument permitted to communicate with this unit.

***IMPORTANT The network.unh is not only used by instruments, but also by LINOPC on the PC for configuring EuroPRP and LIN devices. Configuration of the EuroPRP and LIN devices exceeds the scope of the Properties Dialog.***

## 4.5.3 Running the Instrument Properties dialog

An Instrument Properties dialog can be opened by,

- selecting the properties of an instrument folder from either LINtools or a Project environment
- press the Instrument Options button on the Port Properties page in the Modbus Tools software application, see *Modbus Tools Help* (Part no. HA028988)

  *Note. This will show the Instrument Option parameters only.*

# CHAPTER 5 CONTROL AND AUTOTUNING

The instrument strategy can be configured to control and tune a control loop, via LINtools Engineering Studio. This chapter discusses the use of the LOOP_PID block, but similar Proportional Band, Integral Time, and Derivative Time, PID, principles are also applicable to the 3_Term block and PID block.

*Note Details of each block is described in the LIN Block Reference Manual, Part no. HA 082 375 U003.*

Each single loop of control contains two outputs, Channel 1 and Channel 2, that can be configured for PID, On/Off or Valve Position (bounded or unbounded) control. In a temperature control loop Channel 1 is normally configured for heating and Channel 2 for cooling. Descriptions given in this chapter generally refer to temperature control but can also apply to other process loops.

The main topics covered are:

- What is a Control Loop ? (section 5.1)
- LOOP_PID Function Block (section 5.2)
- Effect of Control Action, Hysteresis and Deadband (section 5.3)

## 5.1 WHAT IS A CONTROL LOOP ?

This is an example of a heat only temperature control loop.

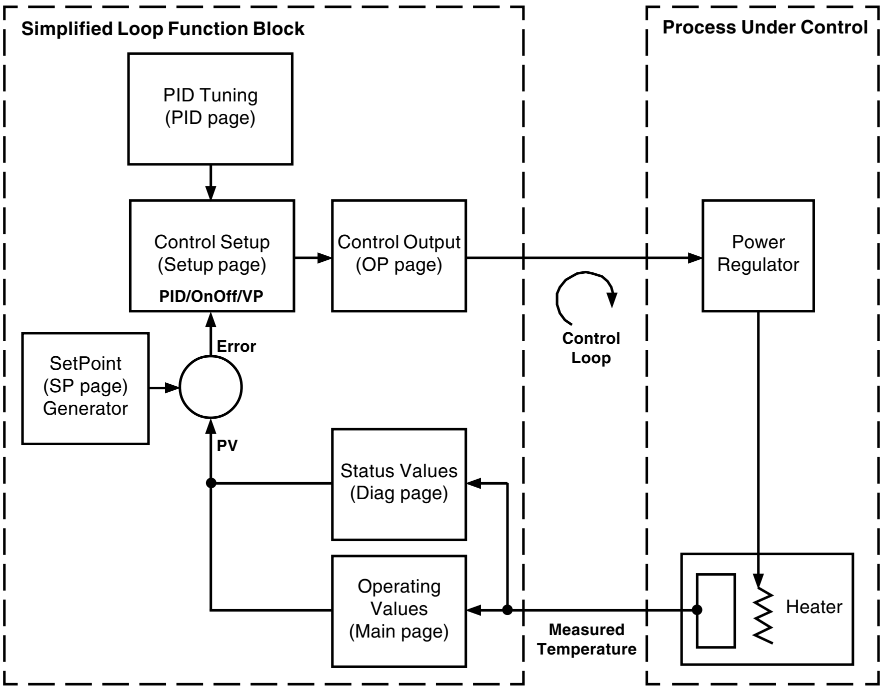



Figure 5-1 Single Loop, Single Channel Control Loop Block schematic

The actual measured temperature, or Process Variable (PV), is connected to the input of the instrument. This PV measurement is compared with a SetPoint (SP, or required temperature). If an error exists between the SP and measured temperature the instrument calculates an output value to call for heating or cooling depending on the process being controlled. In this instrument it is possible to select between a PID, On/Off, Boundless or Bounded Valve Position algorithm. The output(s) from the instrument (OP) are connected to devices in the plant/system, and adjust the heating, or cooling, that results in a change of the PV, that is again measured by the sensor, and the process is repeated. This is referred to as closed loop control.

## 5.2 LOOP_PID FUNCTION BLOCK

The instrument control loop is configured using the Loop function block and up to 7 (seven) additional Tune_Set blocks, allowing a total of eight sets of tuning parameters for an individual control loop.

*Note Each set of PID tuning parameters, one additional set of tuning parameters per Tune_Set block, provides specific tuning for different levels of temperature, particularly useful in control systems where the response to the cooling power is significantly different to that of the heating power.*

The LOOP_PID block consists of the following pages

- Main
  To setup the operating parameters of the Control Loop. These are an overview of the main parameters such as Auto/Manual select, current PV, current output demand, selected SP value and working SP value.
- Setup
  To configure control type for each channel of the selected loop
- Tune
  To set up and run the Auto Tune function
- PID
  To set up 3 term, Proportional Band, Integral Time, and Derivative Time (PID) control parameters
- SP
  To select and adjust different setpoints, setpoint limits, rate of change of setpoint
- OP
  To set up output parameters such as limits, sensor break conditions
- Diag
  To diagnose Control Loop problems, such as sensor break detection, loop break detection
- Alarms
  To setup alarm parameters used to indicate operational extents have been exceeded

*Note Parameters are wired using the LINtools Engineering Studio, as part of a strategy, see LIN Block Reference Manual, Part no. HA 082 375 U003, for full block parameter details.*

### 5.2.1 Main page

The Main page of the Loop block provides an overview of parameters used by the overall control loop. It allows,

- Auto or Manual operation to be selected
- To stop the loop from controlling for commissioning purposes
- To hold the integral action
- Read PV and SP values

#### AUTOMATIC/MANUAL MODE

Each type of control operates differently according to the current operating mode. Automatic indicates that PV is continuously monitored and compared to the SP. The output power is calculated and used to minimise any difference. Manual indicates that the operator controls the output power. The power delivered to the process may be edited directly from the instrument via the User Screen or via the communications network. However, the loop continues to be monitored, allowing a smooth change when Automatic mode is selected.

If On/Off control is configured the output power can be edited by the user but will only allow the power to be set to +100%, 0% or -100%, representing, heat ON/cool OFF, heat OFF/cool OFF, heat OFF/cool ON.

If PID control is configured the output can be edited between +100% and -100%, if cool is configured. The true output value is subject to limiting and output rate limit.

If Valve Position control is configured the raise and lower buttons on a User Screen, configured using the User Screeen Editor, will directly control the raise and lower relay outputs during manual operation. By using digital communications it is possible to control the valve by sending nudge commands. A single nudge command, *OP.NudgeUp* or *OP.NudgeDn*, will move the valve by 1 minimum On-Time. In manual mode the natural state will be rest.

*Note If sensor break occurs while the control loop is in automatic operation, a configured sensor break output power, OP.SbrkOP or OP.SafeOP if Main.Inhibit is configured, can be output. However, the user can now switch to manual control. In this case manual will become active and the user can edit the output power. On leaving manual, i.e. returning to automatic operation control, the controller will again check for sensor break.*

If Auto Tune is enabled, *Tune.Enable* set Yes, while in manual mode, the Auto Tune will remain in a reset state, *Tune.Stage* shows Reset, until the control loop is switched to automatic control, automatically starting the Auto Tune process.

## 5.2.2 Set Up page

Set Up configures the type of control required for each channel.

### TYPES OF CONTROL LOOP

The following control loop types can be configured,

- On/Off Control

  On/Off control is the simplest means of control and simply turns heating power on when the PV, is below SP and off when it is above SP. As a consequence, On/Off control leads to oscillation of the PV. This oscillation can affect the quality of the final product and may be used on non-critical processes. A degree of hysteresis, *Alarms.Hyst*, must be set in On/Off control if the operation of the switching device is to be reduced and relay chatter is to be avoided.

  If cooling is used, cooling power is turned on when the PV is above SP and off when it is below.

  It is suitable for controlling switching devices such as relays, contactors, triacs or digital (logic) devices.

- PID Control

  PID (Proportional Band, Integral time, and Derivative time), or 3 (Three) Term Control, is an algorithm that continuously adjusts the output, according to a set of rules, to compensate for changes in the PV. It provides more stable control but the parameters need to be set up to match the characteristics of the process under control.

  The output from the control is the sum of the contributions from the PID terms. The combined output is a function of the magnitude and duration of the error signal, and the rate of change of the PV.

  It is possible to disable the Integral time and Derivative time terms and control the Proportional Band only (P), or Proportional plus Integral (PI) or Proportional plus Derivative (PD).

  *Note PI control can be used, for example, when the sensor measuring an oven temperature is susceptible to noise or other electrical interference where derivative action could cause the heater power to fluctuate wildly. Whereas, PD control may be used, for example, on servo mechanisms.*

  In addition to the PID terms described above, there are other parameters that determine the control loop performance. These include Cutback terms, Relative Cool Gain, Manual Reset.

- Valve Position Control

  Valve Position (Motorised Valve) Control is an algorithm designed specifically for positioning motorised valves. It operates in boundless, Valve Positioning Unbounded or bounded mode.

  Boundless VP (VPU) control does not require a position feedback potentiometer to operate. It is a velocity mode algorithm that directly controls the direction and velocity of the movement of the valve in order to minimise the error between the SP and the PV. It uses triac or relay outputs to drive the valve motor.

  *Tip! A potentiometer can be used in boundless mode but can only indicate the actual valve position, and is not included in the control algorithm.*

  Bounded VP (VPB) control requires a feedback potentiometer as part of the control algorithm.

  The control is performed by delivering a ‘raise’ pulse, a ‘lower’ pulse or no pulse in response to the control demand signal via relay or triac outputs.

  In manual mode operation, Bounded VP controls by the fact that the inner positional loop is still running against the potentiometer feedback, so it is operating as a position loop.

### 5.2.2 Set Up page (Cont.)

In manual mode operation, BoundlessVP control is an algorithm used as a velocity mode positioner. The algorithm predicts where the valve will move to based on the edit of the manual power. Effectively, when the raise or lower command is activated, the raise or lower output is turned on applying +100% or -100% velocity respectively. It is essential that the motor travel time is correct, so the Integral time can be calculated correctly.

*Note Motor travel time is defined as valve fully open – valve fully closed, it is not necessarily the time printed on the motor because if the mechanical stops have been set on the motor, the travel time of the actual valve may be different. Also, if the travel time for the valve is set correctly, the position indicated on the controller will accurately match the actual valve position.*

Every time the valve is driven to its end stops, the algorithm is reset to 0% or 100% to compensate for any changes that have occurred due to wear in linkages or other mechanical parts.

This technique makes boundless VP look like a positional loop in manual even though it is not, and enables combinations of heating and cooling, e.g. PID heat, VPU cool and have the manual mode work as expected.

*Note Motorised Valve Output configuration will automatically configure the second channel after the first has been setup, e.g. if OP.Ch2Outpt is wired and configured as cooling, OP.Ch1Outpt is automatically wired and configured as heating.*

## 5.2.3 PID page

The PID parameters are used to optimize the control of the loop.

*Note If the loop is configured for On/Off Control, only the PID.LBTn is available.*

- Proportional Band, PB

The Proportional Band, PB, or gain, delivers an output that is proportional to the size of the error signal in engineering units or as a percentage of the range. It is the range over which the output power is continuously adjustable in a linear fashion from 0% to 100%, for a heat only control. An error signal below the PB causes an output of 100%, but an error signal above the PB causes an output of 0%.

The width of the PB determines the response to the error signal. If the error signal is too narrow (high gain) the system oscillates by being over responsive, if it is too wide (low gain) the control is sluggish. A control loop is operating at its optimum performance when the PB is as narrow as possible without causing oscillation.

The diagram below shows the effect of narrowing PB to the point of oscillation. A wide PB results in straight line control but with an appreciable initial error between SP and actual temperature. As the PB is narrowed the temperature gets closer to SP until finally becoming unstable.

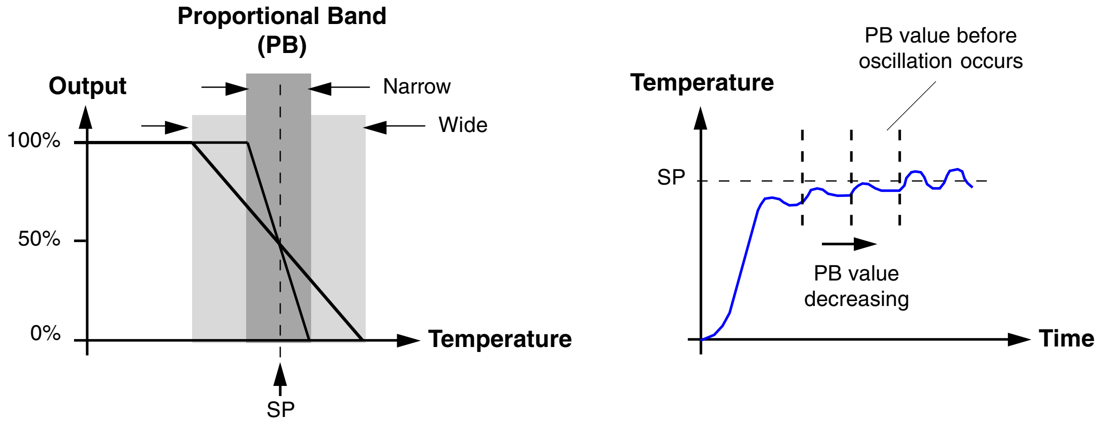



Figure 5-2-3a Proportional Band (PB) configuration

### 5.2.3 PID page (Cont.)

- Integral Time, Ti

In Proportional only control, an error between SP and PV must exist for power to be delivered. Integral time, Ti, is used to achieve zero steady state control error.

The Ti term slowly shifts the output level as a result of an error between SP and measured PV. If the measured PV is below SP the Integral time action gradually increases the output in an attempt to correct the error. If it is above SP the Ti action gradually decreases the output or increases the cooling power to correct the error. The diagram below shows the result of introducing Ti action.

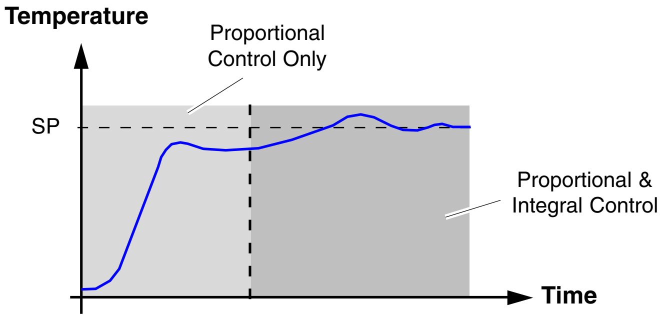



Figure 5-2-3b Integral Time (Ti) configuration

The units for the Ti term are measured in time (1 to 99999 seconds). The longer the Ti constant, the more slowly the output is shifted, resulting in poor response. If the Ti is set too small, it will cause the process to overshoot and even oscillate. The Ti action can be disabled by setting *PID.Tin Off*.

Temporarily disabling the Ti term can be useful when a control loop is expected to open, i.e. it may be necessary to turn heaters off for a short period or switch into manual at low power. In this case it may be an advantage to wire it to a digital input that activates when the heaters are turned off. When the heaters are switched on again the Ti term is already at its previous value minimising overshoot.

In a PID control (3-term control), the Ti term of the PID calculation can be frozen at the current value if *Main.IntHold* is set Yes. It will hold the Ti term at its current value but will not integrate any disturbances in the plant/system. Essentially, this is equivalent to switching to PD control with a manual reset value, Ti term value, preconfigured.

When the control loop is configured to use PID control, changes between manual and automatic can cause abrupt changes to the output value. By configuring Integral Balance, *PID.IntBal*, abrupt changes, bumps, can be prevented, and the output power gradually changed in accordance with the demand from the PID algorithm or by an user via a User Screen.

*Note Output bumps can damage valves and destabilise the process.*

### 5.2.3 PID page (Cont.)

- Derivative Time, Td

  Derivative time, Td action, or rate, provides a sudden shift in output as a result of a rapid change in error, whether or not this is caused by PV alone (derivative on PV) or on SP changes as well (derivative on error selection). If the measured PV falls quickly, the Td provides a large change to the output in an attempt to correct the change in error before it goes too far. It is most beneficial in recovering from small error changes.

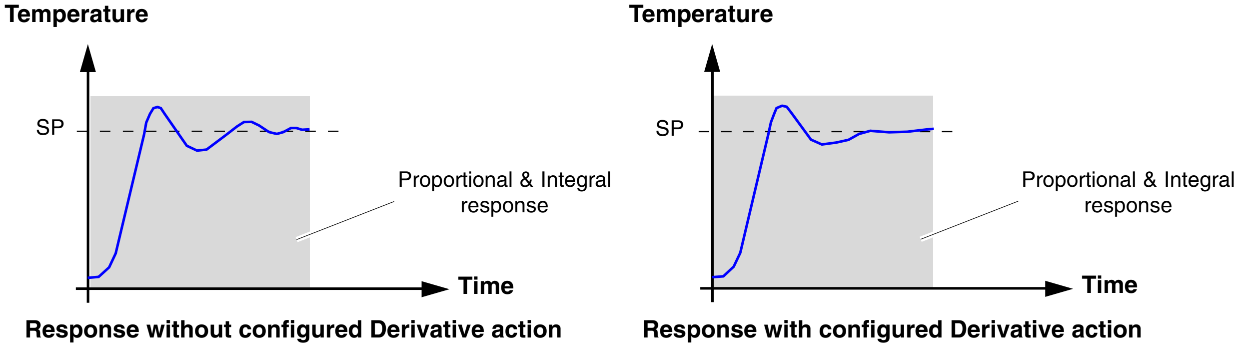



Figure 5-2-3c Derivative Time (Td) configuration

*Note A reduction to wear on valve control can be achieved by configuring Td to react to PV changes, whereas, configuring the Td to react to changes to Error, difference between PV and SP, will redue ramp overshoot, and allows rapid response to small SP changes in temperature control systems.*

The Td modifies the output to reduce the rate of error changes. It reacts to changes in the PV by changing the output to remove the errors . Increasing the Td will reduce the settling time of the loop after a change.

Td is often mistakenly associated with overshoot inhibition rather than error response. In fact, Td should not be used to curb overshoot on start up since this will inevitably degrade the steady state performance of the system. Overshoot should be configured using the High and Low Cutback control parameters.

Td is generally used to increase the stability of the loop, however, there are situations where Td may be the cause of instability, e.g. if the PV is noisy, Td can amplify that noise and cause excessive output changes. In these situations it is often better to disable the Td and re-tune the loop. The Td can be disabled by setting *PID.Tdn Off*.

Td can be calculated on change of PV or change of error. If configured on error, changes in the SP will be transmitted to the output. For applications such as furnace temperature control, it is common practice to select Td on PV to prevent thermal shock caused by a sudden change of output as a result of a change in SP.

- Relative Cool Gain, R2G

  The Relative Cool Gain, R2G, is a tuning parameter corresponding to the gain of channel 2 control output, relative to the channel 1 control output.

  R2G compensates for the different quantities of power available to heat, as opposed to that available to cool, a process, e.g. water cooling applications might require an *R2G* value of 0.25 because cooling is 4 times greater than the heating process at the operating temperature.

*Note This parameter is set automatically when the Autotune process is performed.*

### 5.2.3 PID page (Cont.)

- High and Low Cutback, CBHand CBL

  The CutBack High, *PID.CBH*, and CutBack Low, *PID.CBL*, are values that modify the amount of overshoot, or undershoot, that occurs during large step changes in PV, e.g. under start-up conditions, and are independent of the PID terms. This means that the PID terms can be set for optimal steady state response, while the *PID.CBH* and CBL are used to modify any overshoot that may be present.

  *PID.CBH* and *PID.CBL* involves moving the PB towards a cutback point nearest the measured value whenever the latter is outside the PB and the power is saturated, at 0 or 100% for a heat only controller. The PB moves downscale to the lower cutback point and waits for the measured value to enter it. It then escorts the measured value with full PID control to the SP. In some cases it can cause a 'dip' in the measured value as it approaches SP, see below, but generally decreases the time needed to bring the process into operation.

  The action described above is reversed for falling temperature.

  If *PID.CBH* and *PID.CBL* are set to Auto, the values are automatically configured to 3 x PB.

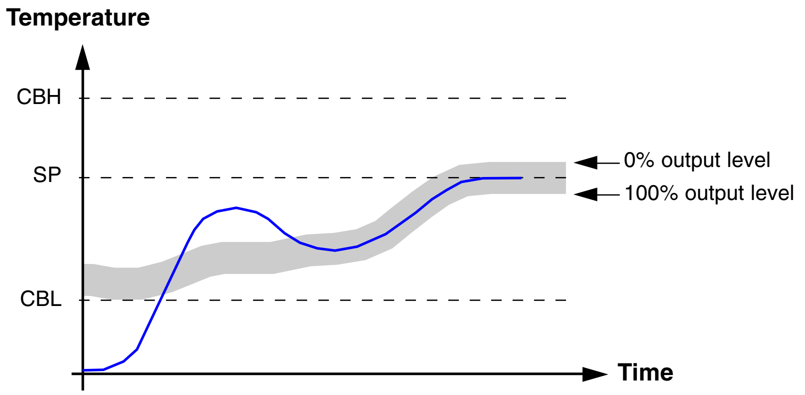



Figure 5-2-3d High and Low Cutback (CBH and CBL) configuration

- Manual Reset, MR

  In a PID control (3 Term control), the Ti term automatically removes the steady state error from the SP. If the PID control is changed to PD control, the Ti term will be set to 'OFF'. Under these conditions the measured value may not settle precisely at SP. The *MR* parameter represents the value of the power output that will be delivered when the error is 0 (zero). To remove the steady state error, the *MR* value must be configured manually.

### 5.2.3 PID page (Cont.)

- Loop Break

The term Loop Break is used to show that the PV has not responded to changes in the output, generally within a configured time. The time of response will usually vary between processes, but by configuring the *LBT* (Loop Break Time) tuning parameter, the *Diag.LpBreak* will only show **Yes** if the PV does not respond before this time limit expires.

The *Diag.LpBreak* attempts to detect loss of restoring action in the control loop by checking the control output, the PV and its rate of change. If the PV has not responded to changes in the output within the configured time limit, *PID.LBTn*, a Loop Break has occurred, setting *Diag.LpBreak* to **Yes**. The control action is not affected unless it is specifically wired, in software or hardware, to the control.

*Note This must not to be confused with Load Failure and Partial Load Failure. The loop break algorithm is purely software detection.*

It is assumed that while the requested output power is operating within the output power limits of a control loop, the control loop is operating in linear control and therefore a Loop Break has not occurred. However, if the output becomes saturated, the control loop is operating outside its linear control region, indicating a Loop Break has occurred.

*Note If the output power remains saturated at the same level for a significant duration, it could indicate a fault in the control loop. The source of the Loop Break is not important, but the loss of control could have serious consequences.*

Since the worst case time constant for a given load is usually known, a worst case time can be calculated using the minimum movement in temperature at the given load. This calculation corresponds to the rate of approach to the SP, and is used to determine that the Loop control will fail at the chosen SP, i.e. if the PV was drifting away from the SP or approaching the SP at a rate less than that calculated, the *Diag.LpBreak* will be set **Yes**.

If an Auto Tune is performed, *LBTn* is automatically set to *Tin* x 2 for PI or PID loop control, and alternatively 12 x *Tdn* for PD loop control. In On/Off control, loop break detection is also based on *LBTn* as 0.1 x SPAN where SPAN = Range High - Range Low. Therefore, if the output is at limit and the PV has not moved by 0.1 x SPAN in the time configured in *LBTn*, the *Diag.LpBreak* will be set **Yes**.

*Note If the time configured in LBTn is 0(off), loop break detection will be disabled.*

If the output is in saturation and the PV has not moved by >0.5 x *Pbn* in the time configured in *LBTn*, the *Diag.LpBreak* will be set **Yes**.

### 5.2.3 PID page (Cont.)

- Gain Scheduling

In some processes the tuned PID set can be very different at low temperatures from that at high temperatures particularly in control systems where the response to the cooling power is significantly different from that of the heating power. Gain Scheduling allows a number of PID sets to be stored and provides automatic transfer of control between one set of PID values and another at different operating points of the process. The Loop block includes one set of PID values, but up to an additional 7 (seven) PID sets, one per Tune_Set block can be used. The total number of PID sets used by the control loop is defined in the *PID.NumSets* parameter.

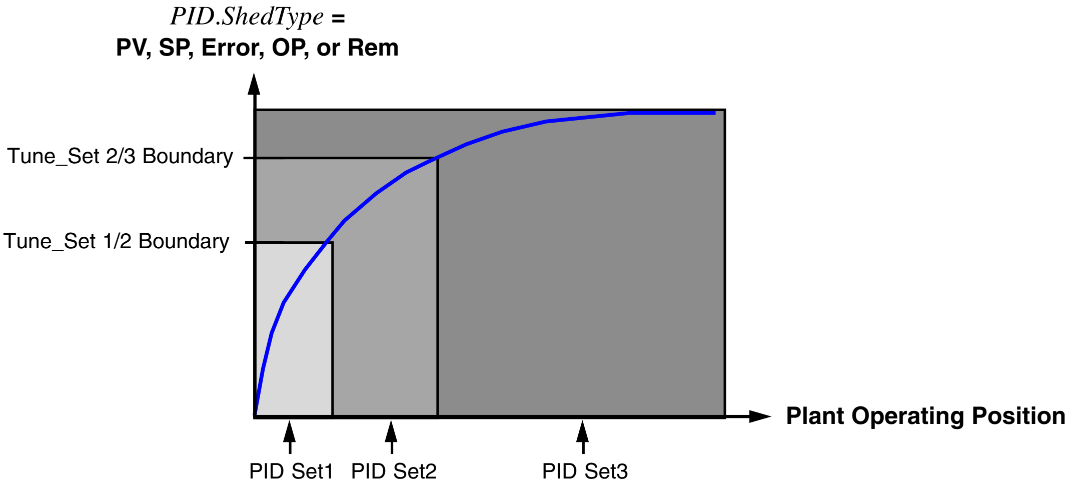



Figure 5-2-3e PID Set boundaries

Gain Scheduling is basically a look up table that can be selected using different strategies or types and provides boundaries, configured in the *Bound* field of each Tune_Set block, that define when the next PID set is used. As the boundary between PID sets is exceeded, under instruction from the Gain Scheduling type defined in *PID.ShedType*, the next PID set is used. The transfer to the next PID set is controlled to stop scheduling oscillation at the boundaries and provides a smooth change between PID sets. The next PID set will start,

- when selected by the operator, if *PID.ShedType* is set to Set. This can also be controlled via the soft wiring within the instrument to allow the operator to select the required PID set remotely.
- when the SP, PV, Error, OP, or Rem value reaches the value configured in the *Bound* field of a Tune_Set block, if *PID.ShedType* is set to SP, PV, Error, OP, or Rem respectively.

*Note Auto tune will tune to the active scheduled PID.*

### 5.2.4 Tuning Page

This page is used to automatically configure parameters that are used to set up and run the Auto Tune function.

Tuning involves configuring the Proportional Band, *PB*, Integral Time, *Ti*, Derivative Time, *Td*, CutBack High, *CBH*, CutBack Low, *CBL*, and Relative Cool Gain, *R2G*, parameters, applicable to heat/cool systems only.

The Loop block is added to the strategy with these parameters set to default values. In many cases the default values will give adequate stable straight line control, however, the response of the loop may not be ideal. This is because the process characteristics are fixed by the design of the process, and therefore it is necessary to adjust the control parameters to achieve best control. To determine the optimum values for any particular loop or process it is necessary to carry out a procedure called Loop Tuning.

---

**Caution**

If changes are made to the process that affect the Control Loop response significantly, it may be necessary to retune the control loop.

---

Users have the choice of tuning the loop automatically or manually. Both procedures require the loop to oscillate to provide a control signal.

#### LOOP RESPONSE

Excluding loop oscillation, loop performance can be described as,

- Under Damped
  In this situation the terms are set to prevent oscillation but do lead to an overshoot of the PV followed by decaying oscillation to finally settle at the SP. This type of response can give a minimum time to SP but overshoot may cause problems in certain situations and the loop may be sensitive to sudden changes in Process Value. This will result in further decaying oscillations before settling once again.
- Critically Damped
  This represents an ideal situation where overshoot to small step changes does not occur and the process responds to changes in control, e.g. PV does not oscillate close to SP.
- Over Damped
  In this situation the loop responds in a controlled but sluggish manner that will result in a loop performance that is not ideal and unnecessarily slow. The balancing of the P, I and D terms depend totally on the nature of the process to be controlled.

Example

In a plastics extruder, a barrel zone will have a different response to a die, casting roll, drive loop, thickness control loop or pressure loop. In order to achieve the best performance from an extrusion line all loop tuning parameters must be set to their optimum values.

### 5.2.4 Tune Page (Cont.)

## INITIAL LOOP BLOCK SETTINGS

In addition to the tuning parameters, there are a number of other parameters that can effect the loop response. Ensure that the following parameters, but not exclusively, are set before any tuning is initiated.

- Setpoint
  Before starting a tuning process, the control loop conditions should be set as closely as practicable to the actual conditions that will be met in normal operation, e.g. in a furnace or oven application a representative load should be included, an extruder should be running, etc.
- Heat/Cool Limits
  The minimum and maximum power delivered to the process can be limited by the parameters *OP.OutputLo* and *OP.OutputHi*. In heat only control the default values are 0 and 100%, but in heat/cool control the defaults are 100 and 100% only. Although it is expected that most processes will be designed to work between these limits it is possible to limit the power delivered to the process, e.g. if driving a 220V heater from a 240V source the heat limit may be set 80% to ensure that the heater does not dissipate more than its maximum power.
- Remote Output Limits
  The *OP.RemOPL* and *OP.RemOPH* parameters must be set within the Heat/Cool Limits, if used.
- Heat/Cool Deadband
  In heat/cool control, use the *OP.Ch2DeadB* to set the distance between the heat and cool PBs. The default value is 0%, indicating that the heating will turn off at the same time as cooling turns on. The deadband must be set to ensure that the heat and cool channels will not run at the same time, particularly when cycling output stages are installed.
- Minimum On-Time
  If either or both of the output channels is fitted with a relay, triac or logic output, the *OP.NudgeUp and OP.NudgeDn* parameters apply the On-Time, for the cycling time of a time proportioning output and should be configured correctly before tuning is started.
- Output Rate limit
  This parameter, *OP.RateOP*, is active during tuning and can affect the tuning results.
- Valve Travel Time
  If an output is connected to a motor valve positioner, *OP.C1TravT* and *OP.C2TravT* must be configured according to the application.

Before the Tuning process begins, it is recommended

- the tuning process is always started when PV and SP are not in close proximity. This allows start up conditions to be measured and CutBack High, *CBH*, and CutBack Low, *CBL* values to be calculated more accurately.
- the tuning should only be attempted during dwell periods and not during ramp stages. If a control loop is tuned automatically, set *Main.IntHold* to **Yes** during each dwell period while Auto Tune is active. It may be worth noting that tuning, carried out in dwell periods that are at different extremes of temperature can give different results owing to non linearity of heating or cooling. This can provide a convenient way to establish values for Gain Scheduling.
- the *OP.OutputHi* and *OP.OutputLo* parameters are configured, as required. These overall output limit parameters apply during tuning and normal operation.
- the *Tune.HiOutput* and *Tune.LoOutput* parameters are configured, as required. These output power limit parameters apply during the Auto Tune function.

*Note The 'tighter' power limit will always apply, e.g. if Tune.HiOutput is set to 80% and OP.OutputHi is set to 70%, the output power will be limited to 70%. The measured value must oscillate to some degree for the tuner to be able to calculate values. The limits must be set to allow oscillation about the SP.*

### 5.2.4 Tune page (Cont.)

## AUTOMATIC TUNING

Automatic tuning operates by switching the output on and off to induce an oscillation in the PV, and calculates the PID tuning parameter values from the amplitude and period of the oscillation. This automatically configures each of the PID parameters with default values.

- Proprtional Band, PB
  This parameter is not tuned using this process.
- Integral time, Ti, and Derivative, Td
  If using PI, PD or P only control, i.e. if Ti and/or Td is set to OFF, disabled, relevant parameters will not be tuned.
- CutBack High, *PID.CBH*, and CutBack Low, *PID.CBL*
  These parameters can only be automatically tuned if a specific value, not AUTO, is configured before Auto Tune is started. If *PID.CBH* and/or *PID.CBL* is set to Auto, these parameters will remain at the default value 3 x PB.

  *Note Auto Tune will never return PID.CBH or PID.CBL values less than 1.6 x PB.*

- Relative Cool Gain, *PID.R2G*
  This parameter can only be automatically tuned if the control is configured as heat/cool. The tuning will always limit the calculated *PID.R2G* value to between 0.1 and 10. If the calculated value is exceeds this limit, *R2G* remains at its previous value but all other tuning parameters are changed.
- Loop Break Time, *PID.LBT*
  The tuning of this parameter depends on the Ti configuration. If Ti is set to OFF, disabled, this parameter is set to 12 x Td, but if Ti is enabled, *PID.LBT* is set to 2 x Ti.

**Caution**

During automatic tuning faults may occur. If a sensor break occurs, *Diag.SensorB* shows On and *Alarms.SensorB* shows TRUE, the Auto Tune will abort and the instrument will deliver the output power configured in *OP.SbrkOP*. Once the fault has been repaired and the fields cleared, the Auto Tune must be re-started.

Automatic tuning can be performed if more than one PID set is used in the control loop. The calculated PID values will be written to the PID set that is active on completion of the tune. Therefore, the user can tune and write the PID values within the boundaries of the appropriate PID set.

*Note If the boundaries are close at the completion of the tune, it is not guaranteed the PID values will be written to the correct set, particularly if PID.ShedType shows PV or OP. In this situation the PID.ShedType should be set to 'Set' and the 'Active Set' chosen manually.*

The Auto Tune algorithm reacts depending on the initial conditions of the plant, i.e. from where PV starts. In a heat/cool, or heat only control loop, automatic tuning can start when PV is,

- below the SP
- at the same value as the SP, i.e. within 0.3% of the range if *Setup.PB_Units* is set to % or ±1 engineering unit, 1 in 1000, if *Setup.PB_Units* is set to Eng.
- outside the *OP.OutputHi* and *OP.OutputLo* or *Tune.HiOutput* and *Tune.LoOutput* as determined by the tightest parameter values.

### 5.2.4 Tune page (Cont.)

Tuning from below SP - Heat/Cool control loop

The point that automatic tuning is performed, Tune Control Point, is designed to operate just below the Target SP, Loop block - *Main.TargetSP*, the expected operating value of the process. Using a Tuning Control Point configured below the Target SP ensures the process is not significantly overheated or overcooled and is calculated as,

**Tune Control Point = Initial PV + 0.75 (Target SP – Initial PV)**

*Note The Initial PV is the PV measured after a settling period of 1 minute.*

Example

If Target SP = 500°C and Initial PV = 20°C, the Tune Control Point is calculated at 380°C.

If Target SP = 500°C and Initial PV = 400°C, the Tune Control Point is calculated at 475°C.

*Note An overshoot is likely to be less in the second example because the process temperature is already close to the Target SP.*

When automatically tuning a heat/cool control loop and the Initial PV is below the SP, a number of cycles are run to calculate the PID tuning parameters.

i Auto Tune is started, *Tune.Enable* set On (A), but both heating and cooling power remain off for 1 minute (A - B) to allow the algorithm to establish steady state condition, then calculate the Initial PV.

ii First heat/cool cycle (B - D) establishes the first overshoot used to calculate *PID.CBL* if it is not set to Auto.

iii Two cycles of oscillation (B - F) are produced to measure the peak to peak response, the true period of oscillation, and calculate the PID terms.

iv An extra heat stage (F - G) is applied and all power is turned off to allow the plant to respond naturally. During this period the *PID.R2G* is calculated, then *PID.CBH* is calculated using the sum *PID.CBL* x *PID.R2G*.

v Auto Tune is complete, *Tune.Enable* set Off (H). The control loop is now operating at the Target SP using the automatically tuned PID term values.

*Note This operation also applies if the Initial PV is above SP, but will start with full cooling applied from (B), and not full heating, as below.*

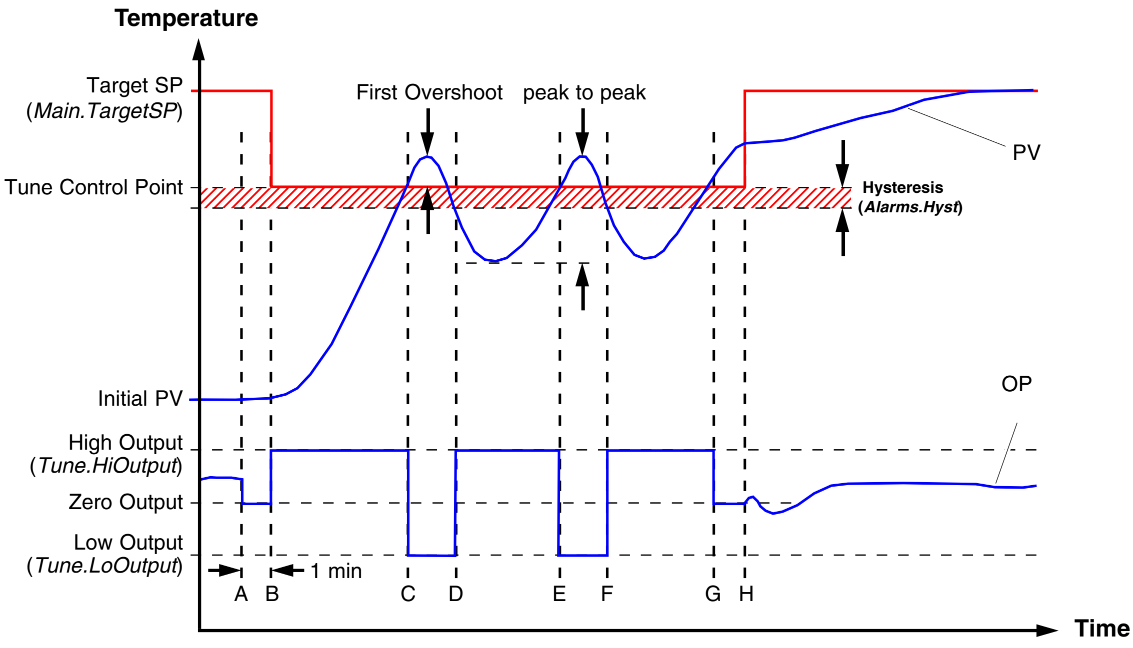



Figure 5-2-4a Tuning from below SP - Heat/Cool control loop

### 5.2.4 Tune page (Cont.)

Tuning from below SP - Heat only control loop

When automatically tuning a heat only control loop and the Initial PV is below the SP, a number of cycles are run to calculate the PID tuning parameters. The operation is similar to the heat/cool control loop, but because a cooling channel does not exist, it completes prematurely, ignoring the *PID.R2G*.

*Note PID.R2G is set to 1.0 for heat only control loop*

i Auto Tune is started, *Tune.Enable* set On (A), the heating power remains off for 1 minute (A - B) to allow the algorithm to establish steady state condition, then calculate the Initial PV.

ii First heat cycle (B - D) establishes the first overshoot used to calculate *PID.CBL* if it is not set to Auto and *PID.CBH* is set to the same value.

iii Two cycles of oscillation (B - F) are produced to measure the peak to peak response, the true period of oscillation, and calculate the PID terms.

iv Auto Tune is complete, *Tune.Enable* set Off (F). The control loop is now operating at the Target SP using the automatically tuned PID term values.

*Note This operation also applies if the Initial PV is above SP, but will start with full cooling applied from (B), and not full heating, PID.CBH is calculated, not PID.CBL, and PID.CBL is set to the same value as PID.CBH.. The operation is similar to the heat/cool control loop, but because a cooling channel does not exist, it completes prematurely, ignoring the PID.R2G.*

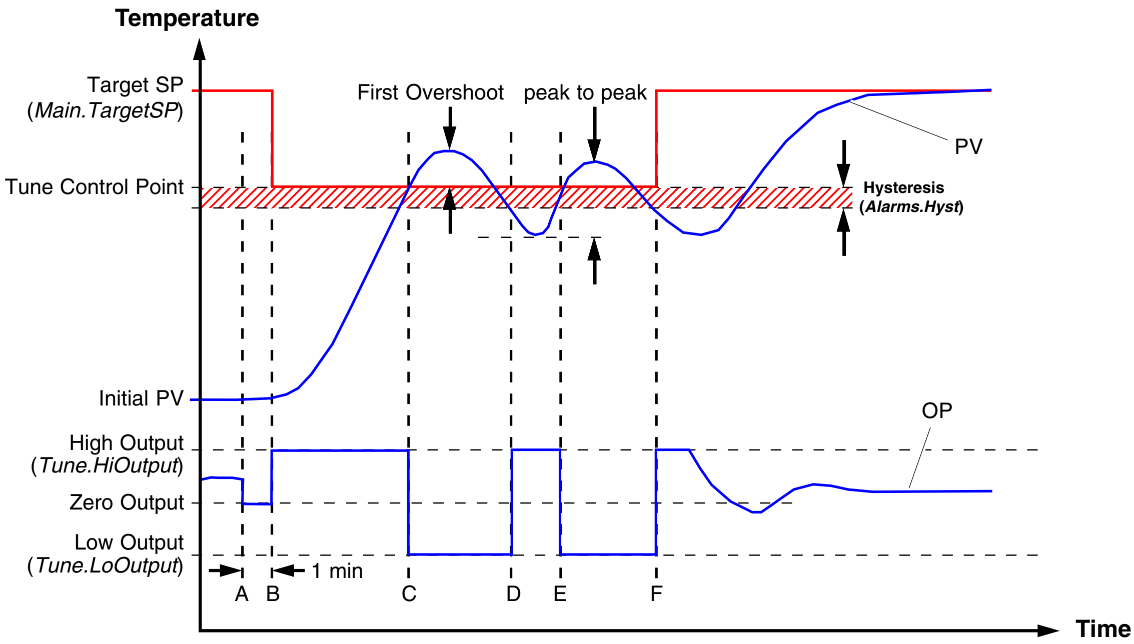



Figure 5-2-4b Tuning from below SP - Heat only control loop

### 5.2.4 Tune page (Cont.)

Tuning at SP - Heat/Cool and Heat only control loop

When automatically tuning either type of control loop and the Initial PV is configured at the same value as the SP, a number of oscillations are produced to calculate the PID tuning parameters. This operation does not calculate *PID.CBH* and *PID.CBL* because there is not an initial start up response to the application of heating or cooling.

*Note PID.CBH and PID.CBL will never return a value less than 1.6 x PB.*

i Auto Tune is started, *Tune.Enable* set On (A). The output is frozen at the current value for 1 minute (A - B), and SP must remain within 0.3% of the range of the control if *Setup.PB_Units* is set to %, Percent, or ±1 engineering unit (1 in 1000) if set to Eng. Range is defined using the *SP.RangeHi*, and *SP.RangeLo* parameters. If during this period the PV drifts outside these conditions Auto Tune will be aborted, and resumed from above or below SP depending on which way the PV has drifted.

*Note A Tune Control Point is not used because the loop is already at SP.*

ii Cycles of oscillation (C - G) are produced by switching the output between the output limits, and are used to measure the peak to peak response, the true period of oscillation, and calculate the PID terms.

iii An extra heat stage (G - H) is applied and all power is turned off (H) to allow the plant to respond naturally. During this period the *PID.R2G* is calculated.

iv Auto Tune is complete, *Tune.Enable* set Off (I). The control loop is now operating at the Target SP using the automatically tuned PID term values.

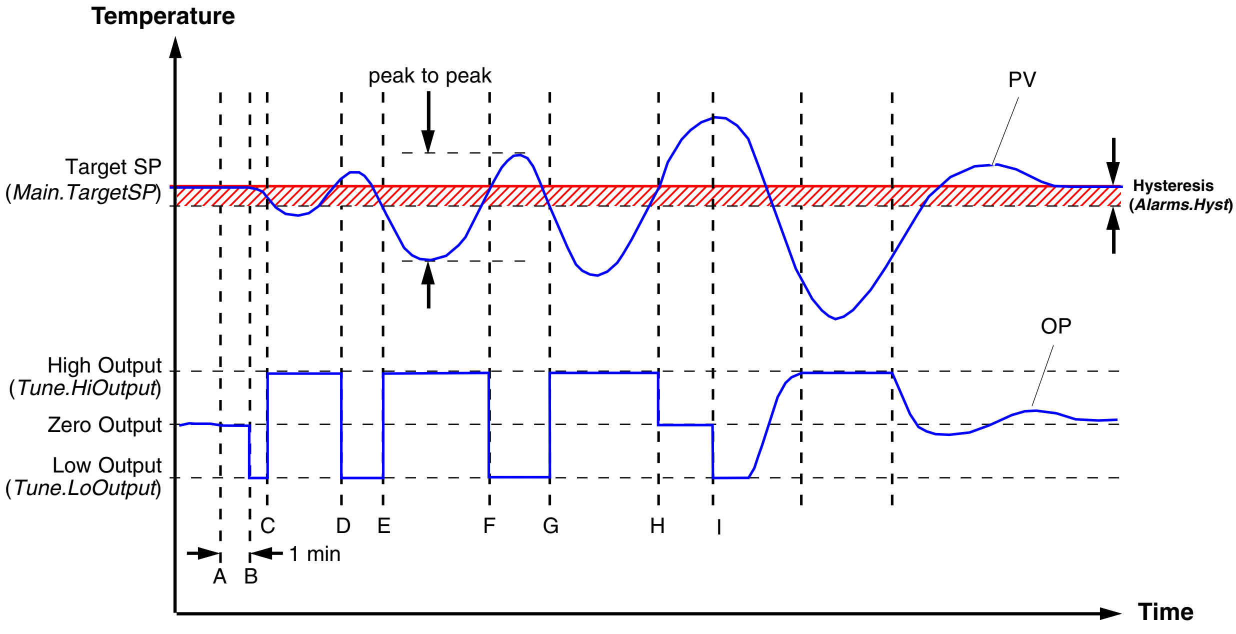



Figure 5-2-4c Tuning from below SP - Heat/Cool and Heat only control loop

### 5.2.4 Tune page (Cont.)

## MANUAL TUNING

If automatic tuning gives unsatisfactory results, the control loop can be tuned manually. There are a number of standard methods for manual tuning, this is the Ziegler-Nichols method.

*Note In a heat/cool control loop, channel 2 must be correctly configured for cooling before tuning is started to allow accurate tuning of the PID.R2G.*

i Adjust SP to the normal operating condition. It is assumed this will be above the PV so that heat only is applied.

ii Set the Integral Time, *PID.Ti*, and the Derivative Time, *PID.Td*, to OFF.

iii Set CutBack High, *PID.CBH*, and CutBack Low, *PID.CBL*, to Auto. These can be changed later, if required.

*Note It is not important that PV does not settle precisely at the SP.*

iv Depending how PV is reacting edit the *PID.PB* value. If PV is stable, reduce and keep reducing *PID.PB* until just before PV starts to oscillate, allowing the loop to settle between each change. Record the *PID.PB* value and the time taken for PV to oscillate. If PV is already oscillating, measure the time taken for PV to oscillate, then increase the *PID.PB* until it just stops oscillating. Record the *PID.PB* value.

*Note The measured time taken for PV to oscillate is used to calculate the PID.Ti and PID.Td values for manually tuning the control loop, see table below.*

v Configure the PID values according to the type of control used, see below.

| Type of Control | Proportional Band *PID.PB* | Integral Time *PID.Ti* | Derivative Time *PID.Td* |
|---|---|---|---|
| Proportional Only | 2 x PB | OFF(Disabled) | OFF(Disabled) |
| Proportional and Integral | 2.2 x PB | 0.8 x measured time | OFF(Disabled) |
| Proportional, Integral and Derivative | 1.7 x PB | 0.5 x measured time | 0.12 x measured time |

Tuning the Relative Cool Gain, *PID.R2G*

The *PID.R2G* parameter is used to compensate for the different quantities of energy needed to heat, as opposed to that needed to cool a process.

i Observe the oscillating PV, an uneven waveform indicates the energy needed for each process is not compensated correctly.

ii Adjust the *PID.R2G* value to produce a symmetrical waveform showing the energy needed for each process is compensated correctly.

iii When the waveform is symmetrical, configure the PID values according to the type of control used, see above.

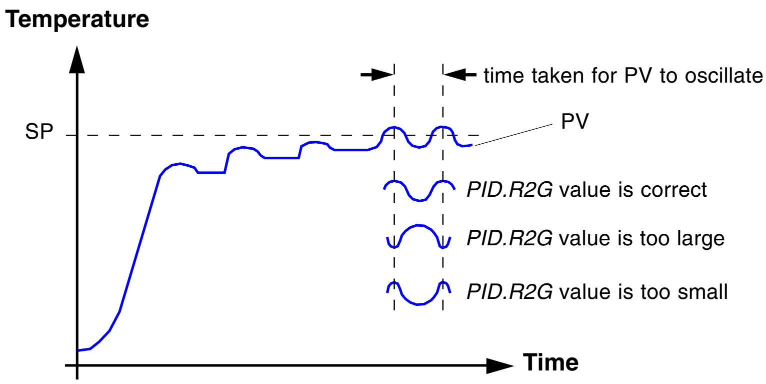



Figure 5-2-4d Relative Cool Gain waveform tuning

### 5.2.4 Tune page (Cont.)

Tuning the CutBack High, *PID.CBH*, and CutBack Low, *PID.CBL*

The *PID.CBH*, and *PID.CBL* parameters are used prevent unacceptable overshoot and undershoot at startup or large step changes in PV.

i Adjust SP to the normal operating condition. It is assumed this will be above the PV so that heat only is applied.

ii Set the Integral Time, *PID.Ti*, and the Derivative Time, *PID.Td*, to provide the optimum steady state control.

iii Set CutBack High, *PID.CBH*, and CutBack Low, *PID.CBL*, to one proportional bandwidth converted into display units. This is calculated using *PID.PB* defined as a %, percentage, value in

$$PID.CBH \text{ and } PID.CBL = \frac{PB}{100} \times \text{Span of control}$$

Example

If PB = 10% and the Span of the control is 0 - 1200°C, then

$$PID.CBH \text{ and } PID.CBL = \frac{10}{100} \times 1200 = 120$$

If overshoot is observed following the correct settings of the PID terms increase the value of *PID.CBL* by the value of the overshoot in display units. If undershoot is observed increase the value of the parameter *PID.CBH* by the value of the undershoot in display units

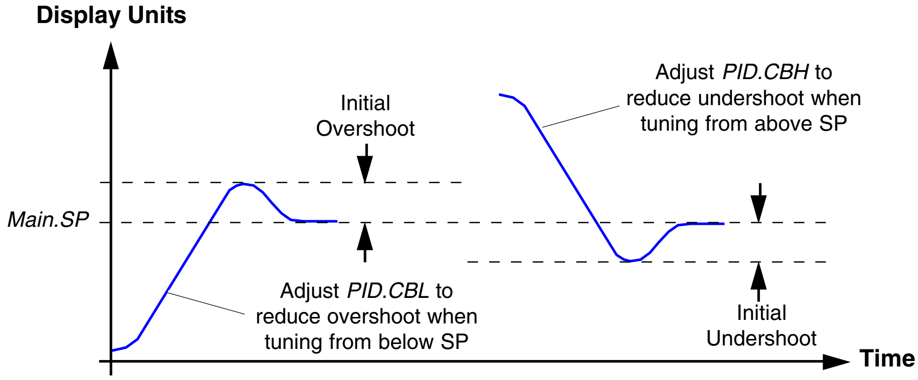



Figure 5-2-4d CutBack High, *PID.CBH*, and CutBack Low, *PID.CBL* waveform tuning

### 5.2.5 SP page

The SetPoint, SP, page of the Loop block provides parameters for configuring the SP used by the control loop.

The control SP, defined as the Working SetPoint (*Main.WSP*), is the value ultimately used to control the PV in a control loop, and can be derived from,

- *SP.SP1* or *SP.SP2*, can be configured by the user and switched into use by an external signal or via a user interface
- *SP.AltSP*, an external (remote) analogue source

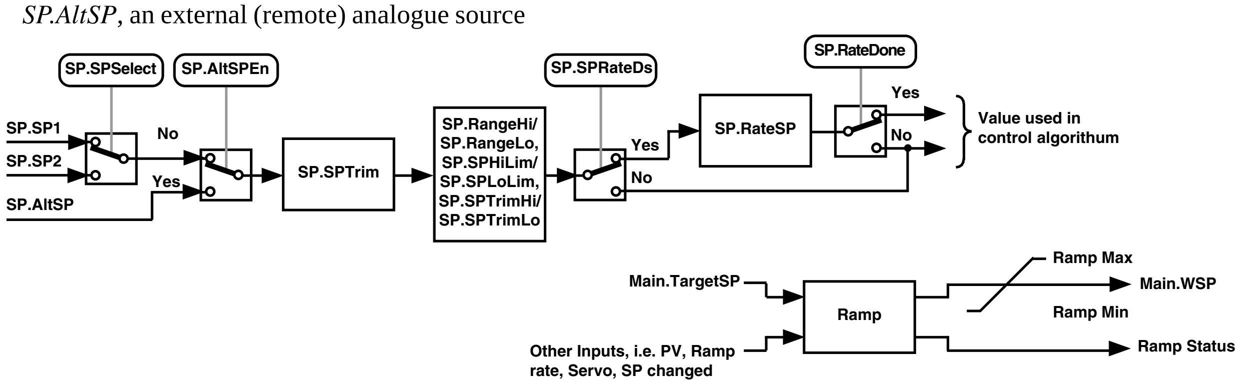



Figure 5-2-5a SetPoint, SP, page block diagram

### 5.2.5 SP page (Cont.)

When the control loop is configured, changes to the *Main.TargetSP* can cause abrupt changes to the output value. By configuring Setpoint Integral Balance, *SP.SPIntBal*, abrupt changes, bumps, can be prevented, and the output power gradually changed in accordance with the demand by an user via a User Screen.

This page also provides the facility to limit the rate of change of the SP before it is applied to the control algorithm. It will also provide upper and lower SP limits, *SP.SPHiLim* and *SP.SPLoLim*, for the local SPs, *SP.SP1* and *SP.SP2*.

*Tip! SP.RangeHi and SP.RangeLo provides range information for the control loop in the control calculation to generate the Proportional Bandwidth, Span = SP.RangeHi - SP.RangeLo. These parameters ultimately affect all SP values.*

User configurable methods for tracking are available, providing smooth transfers between SP values and between operational modes.

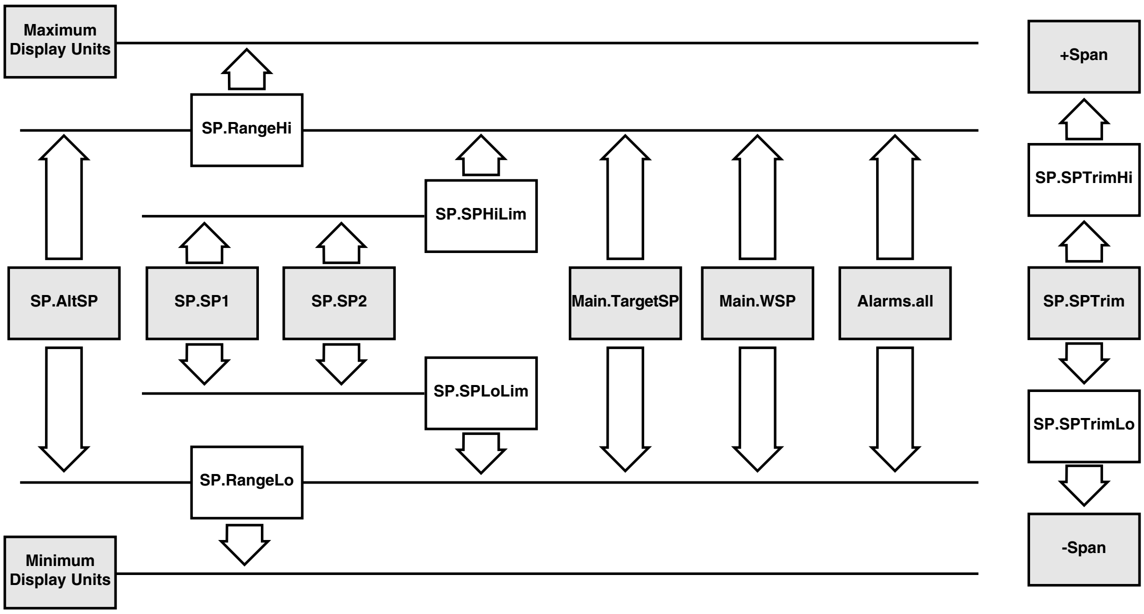



Figure 5-2-5b Setpoint Limits

- Setpoint Rate Limit, SP.RateSP

  The Setpoint Rate Limit, *SP.RateSP*, allows the rate of change of SP to be controlled and prevents step changes in the SP. It is a simple symmetrical rate limiter including any configured Setpoint Trim, *SP.SPTrim*, applied to the Working SP, *Main.WSP*. *SP.RateSP* is controlled by Setpoint Rate Limit Disable, *SP.SPRateDS*. If *SP.SPRate* is set Off, any change made to the SP will be effective immediately, but when a value is set any change in the SP will be effected at the value set in units per minute. *SP.RateSP* applies to *SP.SP1*, *SP.SP2* and *SP.AltSP*.

  When *SP.RateSP* is active *SP.RateDone* will display **No**. When the SP has been reached the value configured in this parameter, *SP.RateDone* will change to **Yes**, but will be cleared if the Target Setpoint, *Main.TargetSP*, is changed.

  When *SP.RateSP* is set to a value, not Off, *SP.SPRateDS* can be used to control, disable and enable, the *SP.RateSP*. This avoids constantly switching this parameter between Off and a value.

*Note SP.RateSP is suspended and Main.WSP is set to 0 (zero) if the PV is in sensor break, Diag.SensorB set Yes and Alarms.SBreak set TRUE. When the sensor break is cleared, Main.WSP returns to the defined SP at the configured SP.RateSP.*

### 5.2.5 SP page (Cont.)

- Setpoint Tracking, SP.SPTrack

  Setpoint Tracking, *SP.SPTrack*, ensures the Local SP, *SP.SP1* or *SP.SP2*, adopts the Alternate Setpoint, *SP.AltSP*, value when switching from *SP.SP1* or *SP.SP2* to *SP.AltSP* to maintain bumpless transfer when returning to *SP.SP1* or *SP.SP2*. Bumpless transfer does not take place when changing from Local to Remote.

  *Note If a SP.RateSP value is configured, the SP will be effected at the value set in units per minute when changing from SP.SP1 or SP.SP2 to SP.AltSP.*

  The SP used by the control can be derived from,

  - local SPs, *SP.SP1* or *SP.SP2*. These can be selected via *SP.SPSelect*, digital communications or by a digital input that selects *SP.SP1* or *SP.SP2*, e.g. to switch between normal running conditions and standby conditions. If *SP.RateSP* is set OFF, the new SP value is adopted immediately when the switch is changed.
  - a Remote analogue source. The source could be an external analogue input into an analogue input module wired to *SP.AltSP* or a User Value wired to *SP.AltSP*. The Alternate Setpoint, *SP.AltSP*, is used when the *SP.AltSPEn* shows **Yes**.

- Manual Tracking

  When the control loop is operating in manual mode the currently selected SP, *SP.SP1* or *SP.SP2*, tracks the PV. When the control loop resumes automatic control there will be no step change in the resolved SP. Manual tracking does not apply to the Alternate Setpoint, *SP.AltSP*.

- Servo to PV

  After power cycling the instrument, the time taken to obtain the *Main.WSP* can be increased by configuring *SP.ServToPV*. When *SP.ServToPV* shows On, the measured PV, *Main.PV*, is used as a start point for the *Main.WSP*. This decreases the time required for the *Main.WSP* to arrive at the *Main.TargetSP*.

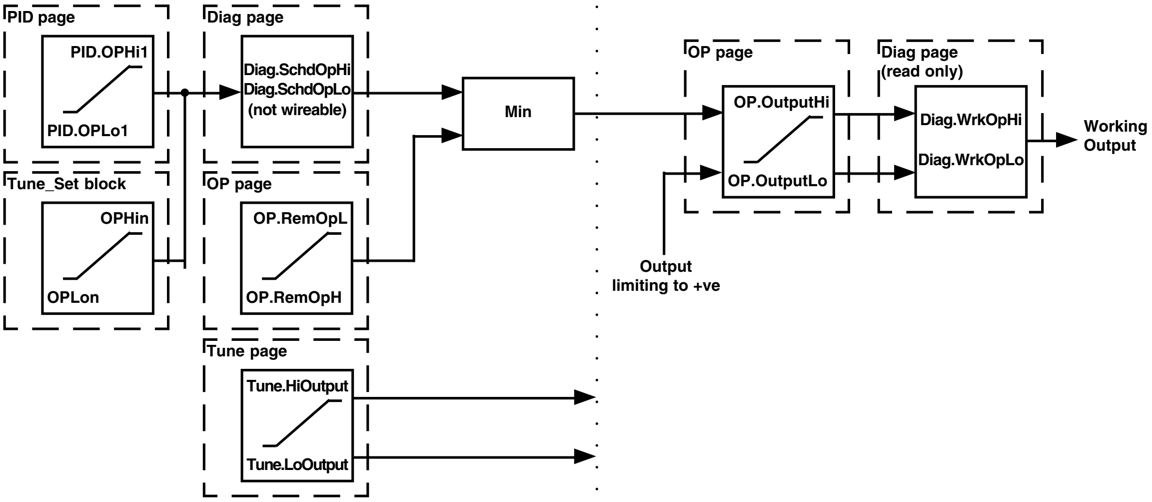
## 5.2.6 OP page

The Output, OP, page of the of the Loop block provides parameters for output control algorithms and manages the output in exception conditions, i.e. start up and sensor break. It selects the correct output sources to be used, determines the heat or cool operation and then applies limits. Power FeedForward and non-linear cooling are also applied. The outputs, *OP.Ch1Outpt* and *OP.Ch2Outpt,* are normally connected to an output module and converted into an analogue or time proportioned signal for electrical heating, cooling or valve movement. These parameters are limited using the upper and lower output limits, *OP.OutputHi* and *OP.OutputLo*. The following additional configuration may also be required,

- Individual output limits can be configured for each set of PID parameters when gain scheduling is used.
- The *Diag.SchdOPHi* and *Diag.SchdOPLo* can be set to values that override the gain scheduling output values.
- A limit can be applied from an external source, derived from *OP.RemOPH* and *OP.RemOPLo,* Remote output high and Remote output low. These parameters are wireable, e.g. they can be wired to an analogue input module so that a limit can be applied through some external strategy. However, if these parameters are not wired, ±100% limit is applied every time the instrument is powered up.
- The tightest set, between Remote and PID, is connected to the output if an overall limit is applied using parameters *OP.OutputHi* and *OP.OutputLo*.
- *Diag.WrkOPHi* and *Diag.WrkOPLo* read only parameters showing the overall working output limits.

  *Note The tune limits are a separate part of the algorithm and are applied to the output during the tuning process. The overall limits OP.OutputHi and OP.OutputLo always have priority.*

### 5.2.6 OP page (Cont.)




*Note Each OPHin and OPLon are derived from a Tune_set block identified by the n, where n equals the PID set number.*

Figure 5-2-5a Output Limits

- Output Rate Limit, OP.RateOP

The Output Rate Limit, *OP.RateOP*, allows the rate of change of OP to be controlled and prevents step changes in the OP. It is a simple symmetrical rate limiter applied to the Working OP, *Main.WrkOP*, and remains active while the control loop is operating in manual mode. The *OP.RateOP* is performed by determining the direction the output is changing, and incrementing or decrementing the Working Output, *Main.WrkOP*, until *Main.WrkOP* is equal to the required Target Output, *Diag.TargetOP*.

The incremental or decremental value is calculated based on the sampling (update) rate of the algorithm, i.e. 110ms, and the configured *OP.RateOP* value. Any change in output less than the rate limit increment will take effect immediately. The direction and increment is calculated on every execution of the rate limit. Therefore, if the rate limit is changed during execution, the new rate of change will take effect immediately. If the output is changed while rate limiting is taking place, the new value will take effect immediately in the direction of the rate limit and in determining whether the rate limit has completed.

*Note The OP.RateOP is self-correcting, i.e. if the increment is small and is lost in the floating point resolution, the increment will be accumulated until it takes effect.*

*OP.RateOP* is controlled by Output Rate Limit Disable, *OP.RateDis*. If *OP.RateOP* shows Off, any change made to the OP will be effective immediately, but when a value is set any change in the OP will be effected at the rate set in %, per cent, per second.

When *OP.RateOP* is set to a value, not Off, *OP.RateDis* can be used to control, disable and enable, the *OP.RateOP*. This avoids constantly switching this parameter between Off and a value.

- Sensor Break Mode, OP.SbrkMode

The Sensor Break Mode, *OP.SbrkMode*, determines the response of the control loop when a Sensor Break occurs. When a Sensor Break is detected by the measurement system, *Diag.SensorB* shows On and *Main.Alarms.Sbreak* set TRUE, the output can be configured to go to a pre-set value, defined by *OP.SbrkOP*, or remain at its current value, *OP.SbrkMode* set Hold.

When *OP.SbrkMode* shows SbrkOP, the output will ramp to the *OP.SbrkOP* value at the rate defined in *OP.RateOP*, unless *OP.RateOP* shows Off causing the output to step to the *OP.SbrkOP* value. When *OP.SbrkMode* shows Hold, the output of the loop will stay at its last good value. If an *OP.RateOP* value, not Off, has been configured a small step may be seen, because the *Main.WrkOP* will limit to the 2 second old value.

When a Sensor Break has been cleared, the power output will ramp from the current value and transfer smoothly to the control value.

### 5.2.6 OP page (Cont.)

- Forced Output, OP.ForcedOP

A Forced Output, *OP.ForcedOP,* is a manually defined control loop output value adopted when switching from automatic control, *Main.AutoMan* shows Auto, to manual control, *Main.AutoMan* shows Man. By default, the output power is maintained and can be edited by the user. An OP value can be automatically applied after power cycling by defining the source using the *OP.ManStart*. When the *OP.ManStart* parameter is set On, *OP.ManMode* is used to define the source of power applied at startup, but if set Off the source of power applied depends on *Main.AutoMan*.

When the control loop output switches to manual mode, *Main.AutoMan* shows Man, the current *Diag.TargetOP* value steps, *OP.ManMode* shows Step, to the output value derived from *OP.ForcedOP*. If *OP.ManMode* shows Track or LastMop, the *OP.ForcedOP* value is not affected.

*Note If OP.ManMode shows Track, and OP.TrackEN shows On, OP.ManOP is derived from a value tracking the Main.WrkOP during automatic control, providing a bumpless transfer to manual mode. Any subsequent edits to the Diag.TargetOP are tracked back into OP.ManOP. If OP.ManMode shows LastMOP, the OP.ManOP value uses the last value configured by the user.*

- Power FeedForward, OP.PwrffEnb

Power FeedForward is used to drive a heating element. It monitors the line voltage and compensates for fluctuations before they affect the process temperature, providing better steady state performance when the line voltage is not stable. It is mainly used for digital type outputs that drive contactors or solid state relays.

Power FeedForward is only applicable to a heating application and can be controlled by Power FeedForward Enable, *OP.PwrffEnb* shows On.

*Note OP.PwrffEnb can be set Off, for any non-electric heating process or when analogue thyristor control is used because compensation for power changes is included in the thyristor driver..*

Example

Consider a process running at 25% power, with zero error and then the line voltage falls by 20%. The heater power would drop by 36% because of the square law dependence of power on voltage. A drop in temperature would result. After a time, the thermocouple and control loop would detect this fall and increase the On-Time of the contactor just enough to bring the temperature back to SP. Meanwhile, the process would be running a bit cooler than optimum that can cause some imperfection in the product.

With Power Feed Forward enabled, *OP.PwrffEnb* shows On, the line voltage is monitored continuously and On-Time increased or decreased to compensate immediately. This prevents any temperature disturbance caused by a line voltage change.

*Note Power FeedForward and Feed Forward are not the same.*

### 5.2.6 OP page (Cont.)

- Cooling Algorithm, *OP.CoolType*

  Cooling Algorithm, *OP.CoolType*, is used to define the method of cooling a system that can vary between applications.

Example

An extruder barrel can be cooled by forced air from a fan, or by circulating water or oil around a jacket. The cooling effect is different depending on the method. The cooling algorithm can be set to linear where the control output changes linearly with the PID demand signal, or it can be set to water, oil or fan where the output changes non-linearly against the PID demand. The algorithm provides optimum performance for these methods of cooling,

- Oil Cooling. Being non-evaporative, oil cooling is pulsed in a linear manner. It is a deep and direct cooling method and needs a lower heat cool gain, *PID.R2G*, than fan cooling
- Water Cooling. Water cooling does not operate well in areas running well above 100°C. The first pulses of water will flash off into steam giving a greatly increased cooling capacity due to the latent heat of evaporation. When the area settles down, less or even no evaporation is possible and the cooling is less severe. The Water cooling algoritm compensates for the transition out of the initial strong evaporative cooling.
- Fan Cooling. This is much gentler than water cooling and not so immediate or decisive because of the long heat transfer path through the finned aluminium cooler and barrel. With fan cooling, a heat cool gain, *PID.R2G*, setting of 3 upwards would be typical and delivery of pulses to the blower would be linear, i.e. the On-Time would increase proportionally with percentage cool demand.

- FeedForward,

  FeedForward is a scaled value that is added to the PID output, before any limiting. It can be used for the implementation of cascade loops or constant heat control. FeedForward is implemented such that the PID output is limited to trim limits, *OP.FFTrimLm*, and acts as a trim on a FeedForward value, *OP.FFOP*. The *OP.FFOP* is derived from the PV or SP, *OP.FFType* shows PV or SP, by scaling the PV or SP by the *OP.FFGain* and *OP.FFOffset*. Alternatively, if *OP.FFOP* shows Remote, a remote value will be used for the *OP.FFOP*, this is not subject to any scaling. The resultant *OP.FFOP* is added to the limited PID OP and becomes the PID output as far as the output algorithm is concerned. The feedback value that is generated must have the *OP.FFOP* removed before being used again by the PID algorithm, as shown below.

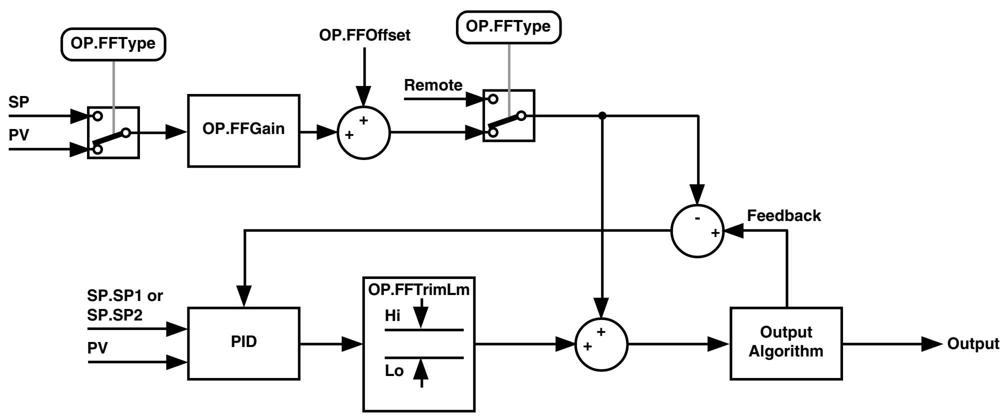



Figure 5-2-5b FeedForward block diagram

### 5.2.7 Diag page

The Diagnostic, Diag, page of the Loop block provides parameters that assist the commissioning of the control loop.

These parameters are generally read only, but can be wired from to produce an application specific strategy, e.g. *Diag.LpBreak* can be wired to an output module to produce a physical output if the Loop Break Time, *PID.LBT*, is exceeded.

Additional gain scheduling parameters are also provided. These display the current values of the control time constants as set by the active PID list and determined by Gain Scheduling.

### 5.2.8 Alarms page

The Alarms page of the Loop block provides parameters that define the alarm limits applied during the operation of the control loop and will help during commissioning.

- High High Absolute, High Absolute, Low Absolute and Low Low Absolute, *Alarms.HiHi,Alarms.Hi, Alarms.Lo, and Alarms.LoLo*

  A High High Absolute, High Absolute, Low Absolute and Low Low Absolute value, displayed in engineering units, define the limits of the process. If the configured value is exceeded the correpsonding alarm field is set TRUE, i.e. *Main.Alarms.Hi* shows TRUE, if PV exceeds an *Alarms.Hi* set at 90. The action of these four multipurpose parameters depends on which type of alarm function is selected (via the *Type* parameter):

| | | |
|---|---|---|
| *HiHighAl* | = | TRUE when *PV* > *HiHigh* |
| *HighAl* | = | TRUE when *PV* > *High* |
| *LowAl* | = | TRUE when *PV* < *Low* |
| *LoLowAl* | = | TRUE when *PV* < *LoLow* |

An alarm is not reset immediately *PV* returns to the alarm level - *PV* must be inside the level by a margin equal to the *Hyst* parameter before the alarm resets. This hysteresis permits clean transitions into and out of the alarm condition. The configured Hysteresis value will be applied.

- High Deviation and Low Deviation, *Alarms.DevHi,and Alarms.DevLo*

  A High Deviation and Low Deviation (*Error*) value, displayed in engineering units, define the limits that PV can deviate from SP before asserting an alarm, *Main.Alarms.DevHi* or *Main.Alarms.DevLo*. The high alarms are set when the positive deviation exceeds the defined levels. The low alarms are set when the negative deviation exceeds the levels:

| | | |
|---|---|---|
| *HiHighAl* | = | TRUE when (*PV–SetPoint*) > *HiHigh* |
| *HighAl* | = | TRUE when (*PV–SetPoint*) > *High* |
| *LowAl* | = | TRUE when (*SetPoint–PV*) > *Low* |
| *LoLowAl* | = | TRUE when (*SetPoint–PV*) > *LoLow*. |

  Hysteresis is applied to deviation values as it is to *PV* in absolute alarms.

- Hysteresis, *Alarms.Hyst*

  A hysteresis value, displayed in engineering units, is applicable to the High Absolute and Low Absolute Alarm limits and the High Deviation, Low Deviation (*Error*) Alarm limits. This value provides a band that defines when the alarm limits are set TRUE. Once an alarm has been annunciated, it is not cleared until the value causing the alarm has returned inside the limit by an amount specified by this parameter.

# 5.3 EFFECT OF CONTROL ACTION, HYSTERESIS AND DEADBAND

## 5.3.1 Control Action, Setup.CtrlAct

When configuring temperature control *Setup.CtrlAct* should be set to Rev. If using PID control this means the heater power decreases as the PV increases, but if using on/off control, output 1, usually heat, will be on, 100%, when PV is below the SP and output 2, usually cool, will be on when PV is above the SP.

## 5.3.2 Hysteresis, Alarms.Hyst

Hysteresis applies to on/off control only and is set in the units of the PV. In heating applications the output will turn off when the PV is at SP. It will turn on again when the PV falls below SP by the hysteresis value, see below.

Hysteresis is used to prevent the OP from oscillating at the control SP. If Hysteresis is set to 0, any change in the PV when operating at SP will change the OP, possibly causing unacceptable oscillations. Hysteresis should be set to a value that provides acceptable life for the output contacts, but does not cause unacceptable oscillations in the PV.

*Note If this performance is unacceptable, it is recommended that you try PID control.*

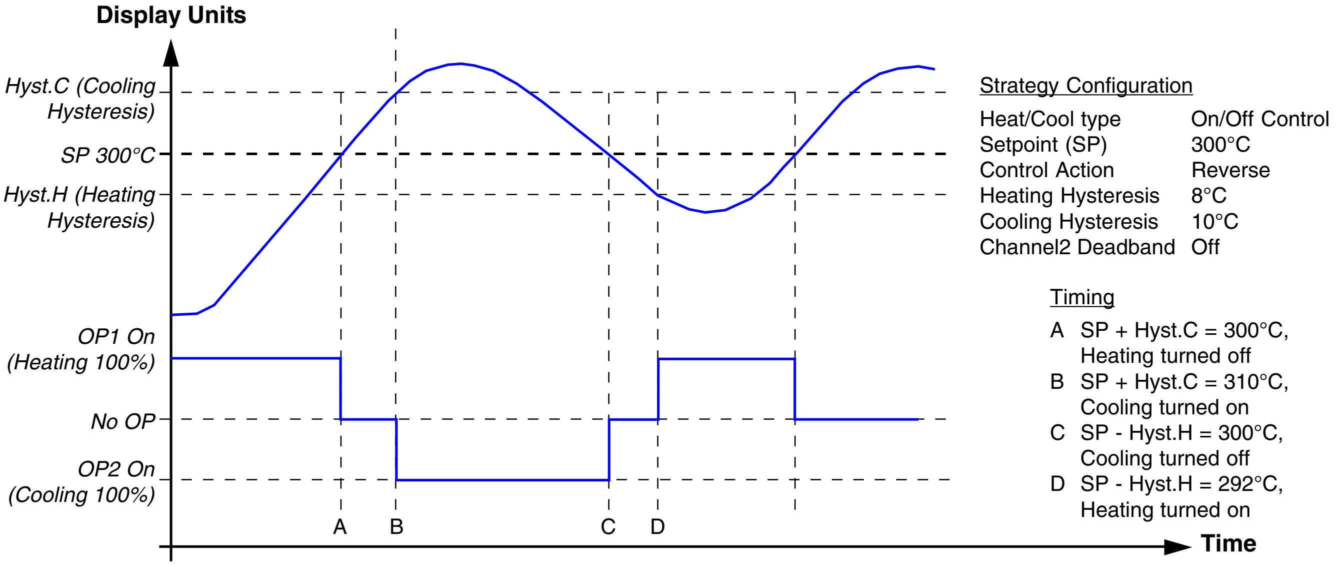



Figure 5-2-6a Hysteresis applied, Deadband not applied

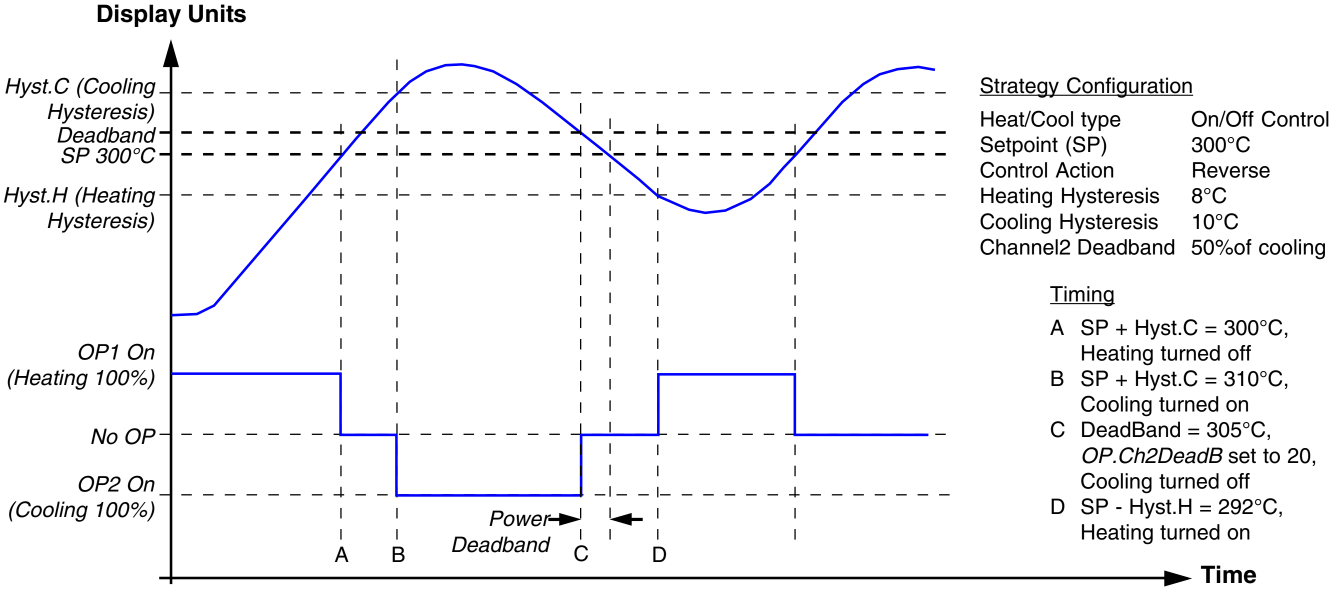



Figure 5-2-6b Hysteresis applied at 5%, Deadband applied at 50%

### 5.3.3 Deadband, OP.CH2DeadB

Channel 2 Deadband, *OP.CH2DeadB*, can operate on both on/off control or PID control. When used in these control types it has the effect of widening the period when no heating or cooling is applied. However, in PID control its effect is modified by both the *PID.Ti* and *PID.Td*.

*OP.CH2DeadB* is expected to be used in on/off control only. However, it can be used in PID control when actuators take time to complete their cycle, ensuring that heating and cooling are not being applied at the same time, see previous diagram.

# CHAPTER 6 OPERATION

This section is for those responsible for the day-to-day operation and monitoring of the instrument. The tasks in this section all require at least the 'Operator' level of access to the instrument. The chapter consists of the following,

- Responding to alarms (section 6.1)

All the descriptions assume that access at Operator level has already been gained, see *Gaining Access* section.

> *Note. If at any point the display differs from what is shown, press the Menu key to return to the top level menu of the Built-in pages allowing the task to be repeated, if required.*

## 6.1 RESPONDING TO ALARMS

This describes the way in which the

- Alarm page is used by the instrument to annunciate alarm, and events occurrence
- Alarm History page is used by the instrument to acknowledge these alarms, and events

### 6.1.1 Time representation

If time synchronisation is configured, the date and time of a cached block are the date and time of alarm/event occurrence at the originating block, and are displayed as DD/MM HH:MM. If it is not certain that the original block's instrument has its own clock synchronised, the date and time of detection of the alarm or event will be used, and these are displayed as DD*MM HH*MM.

### 6.1.2 Filtering options

The following filter options are displayed after pressing the Option key.

| | |
|---|---|
| ALL | Displays all Alarms and Events. |
| = ALARMS | Displays only Alarms. |
| = EVENTS | Displays only Events. |
| = AREA | If an Alarm name is selected, then pressing =AREA causes only those alarms configured to be in the same 'Area' as the highlighted alarm to be displayed. |
| = GROUP | As for =AREA, but for Group. |
| = BLOCK | As for =AREA, but for function block. |

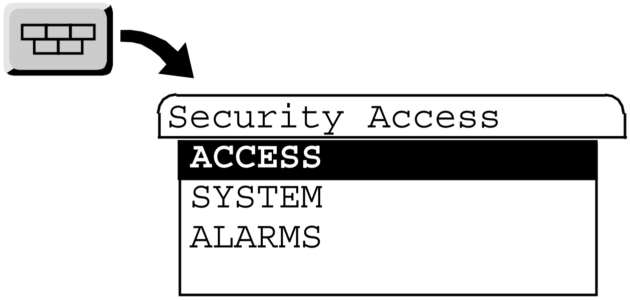
### 6.1.3 Alarm History page

This page displays a list of 'Alarms', and 'Events' that have occurred since the LIN Database was loaded.

*Note. The unit lists 512 'events' before each new event causes the oldest event to be discarded.*

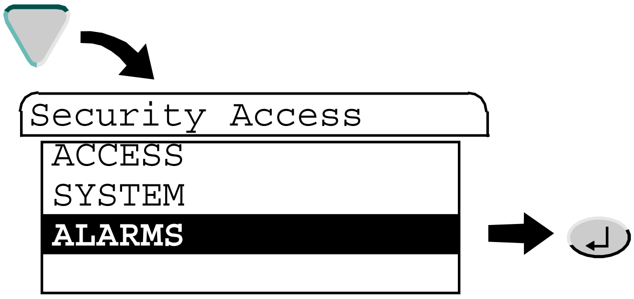
To display the Alarm History page,

1. Press the Menu key to display the top level menu of the Built-in pages.



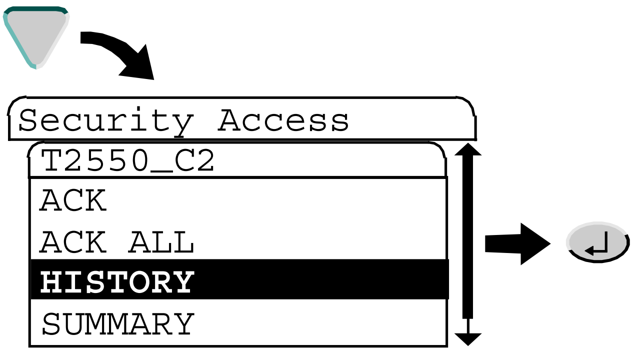

2. Then, using the Navigation keys select ALARMS option from the pop-up menu and confirm using the Return key.

   The Alarms pop-up menu appears.




3. Using the Navigation keys select HISTORY from the Alarm menu and confirm using the Return key.

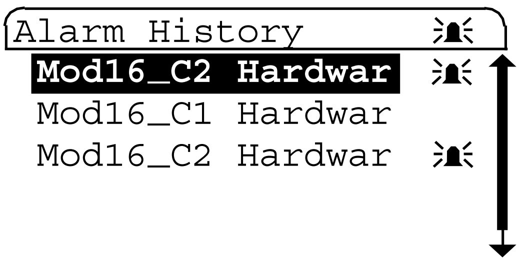
   All previous alarm conditions are displayed in the Alarm History page.




4. The Alarm History page appears.

   Use the Up and Down Navigation keys to select the required alarm record.

   *Note. Use the Right arrow to show or hide additional alarm information, and the Option key to display a menu to allow further filtering of these alarms.*



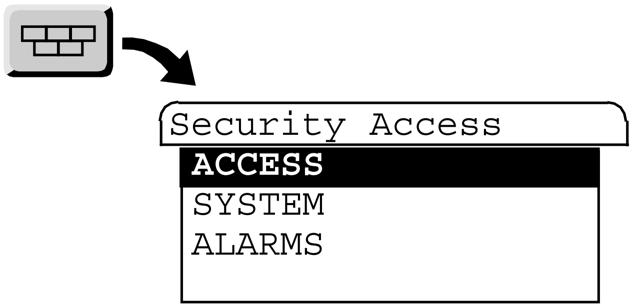

Additional alarm information, giving the date and time of occurrence, and where appropriate, the time of clearing and time of acknowledgement. Where more 'Alarms', and 'Events' have occurred than can be displayed on one page, a 'Scroll bar' appears allowing each 'Alarm', and 'Event' to be displayed in turn.

## 6.1.4 Alarm Acknowledgement

### ALARMS

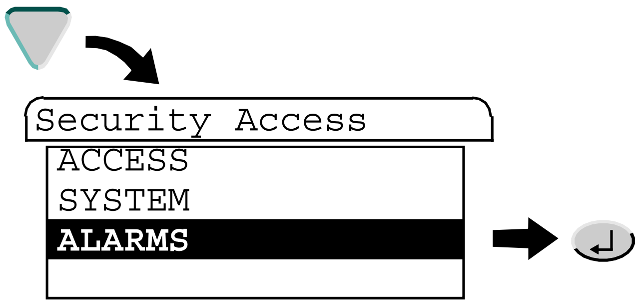
Alarms can be acknowledged by,

1. Press the Menu key to display the top level menu of the Built-in pages.




2. Then, using the Navigation keys select ALARMS option from the pop-up menu and confirm using the Return key.

   The Alarms pop-up menu appears.

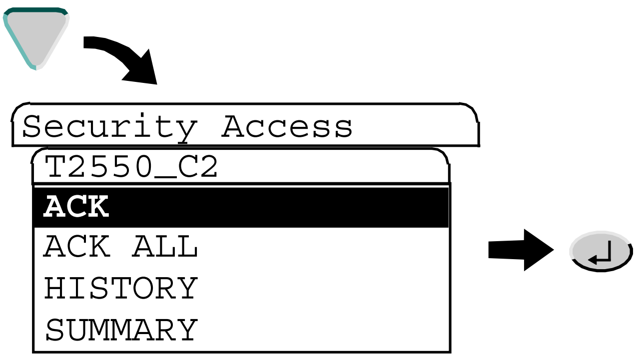



3. Using the Navigation keys select ACK from the Alarm menu and confirm using the Return key.

   *Note. Select ACK ALL to acknowledge all currently active alarms. Always check to ensure it is appropriate to use this command.*

   All previous alarm conditions are displayed in the Alarm Summary page.

   *Note. Use the Navigation key to select the Alarm History page section or the Alarm Summary page section, if required.*



*Intentionally left blank*

# CHAPTER 7 MANAGEMENT

This chapter is for Users responsible for setting up the instrument, for managing application data, and for supervising the day-to-day operation and monitoring of the unit. 'Engineer' level of access to the unit is required, see *Gaining access* section.

This chapter consists of the following sections,

- The Application (section 7.1)
- Displaying the Application Summary page (section 7.2)
- Displaying the Application Manager page (section 7.3)
- Displaying the Function Block Manager page (section 7.4)
- Controlling the Access (section 7.5)
- Setting up the instrument (section 7.6)
- Setting up the Clock (section 7.7)
- Changing Language and Date/Time Formats (section 7.8)
- Setting up the Panel display (section 7.9)
- Displaying the File Manager page (section 7.10)

Most sections consist of several tasks, which are listed in the sub-contents at the start of each section.

*Note. If at any point the display differs from what is shown, press the Menu key to return to the top level menu of the Built-in pages allowing the task to be repeated, if required.*

## 7.1 THE APPLICATION

This section describes the the following functions required by an application, that appear in the Application Manager page.

- *UNLOAD and DELETE:* For many processes, this unit will control one application all the time. It will be loaded and run at commissioning, or soon after, and thereafter will never be unloaded and will never be deleted. For many processes, therefore, UNLOAD and DELETE will not be used.
- *STOP, SAVE, SAVE AS and START:* All processes will use STOP, SAVE, SAVE AS and START because an application has to be stopped to save application data (to preserve cold-start values if they need changing). This is true even if the process runs only one application. START simply restarts an application after a SAVE.
- *LOAD and LD+RUN:* All processes require a LOAD or LD+RUN at least once.

If the very first application is being loaded and run on an instrument, LOAD and then START, or just LD+RUN will be used.

If an application is already running and it is to be replaced by another, the sequence from *Displaying the Application Manager page* to *Loading an application (or Loading and running)* should be referred to.

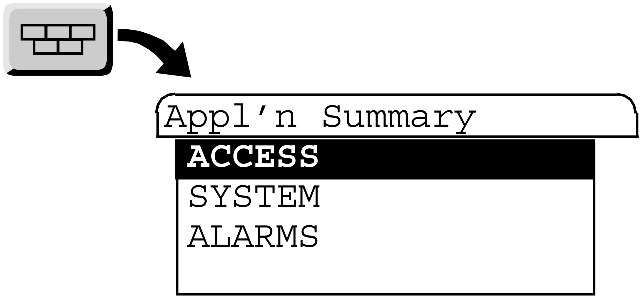
*Note. The Application Summary page and Function Block Manager page provide useful summaries and overviews at any time.*

## 7.2 THE APPLICATION SUMMARY PAGE

This page displays the information about the instrument's various memory resources in percentages (%), that are currently in use.

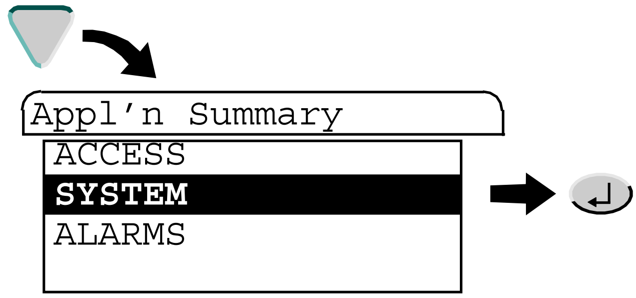
To display the Application Summary page,

1. Press the Menu key to display the top level menu of the Built-in pages.



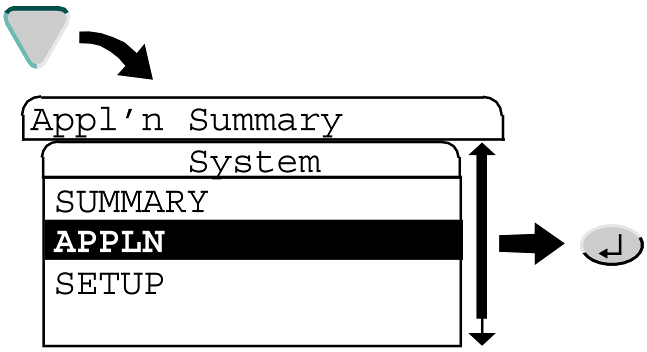

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.

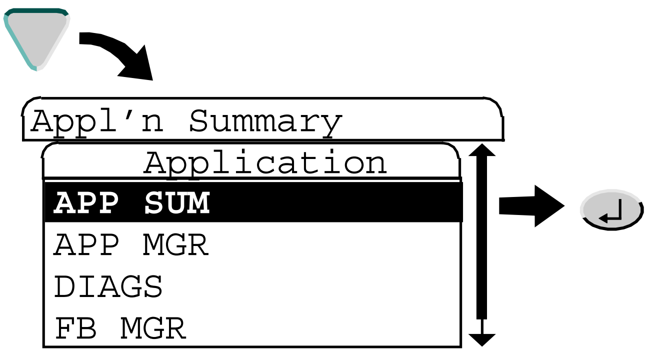



3. Again, using the Navigation keys select the APPLN (Application) option from the pop-up menu and confirm using the Return key.

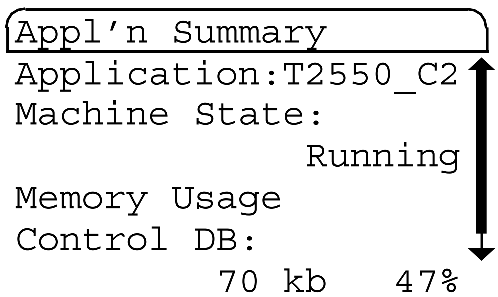
   The Application pop-up menu appears.




4. Finally, using the Navigation keys select the APP SUM option from the pop-up menu and confirm using the Return key.




5. The Application Summary page appears, showing:
   - the name of the loaded application
   - its state (RUNNING, IDLE, or STOPPED)
   - data about memory usage



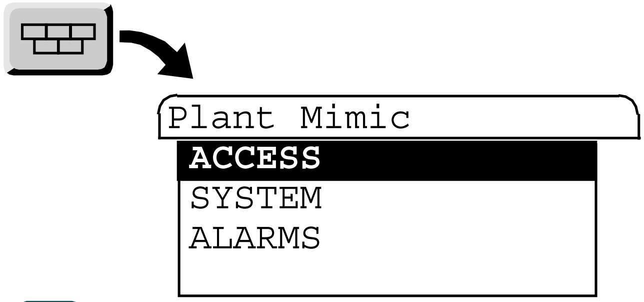
## 7.3 THE APPLICATION MANAGER PAGE

This page controls the operation of the selected application, and consists of the following tasks,

- Stopping an application (section 7.3.1)
- Saving an application (section 7.3.2)
- Unloading an application (section 7.3.3)
- Loading or loading and running an application (section 7.3.4)
- Deleting an application (section 7.3.5)

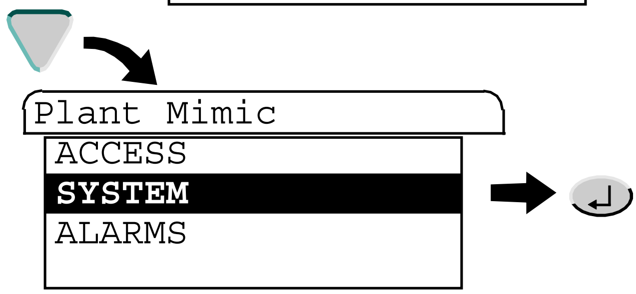
To display the Application Manager page,

1. Press the Menu key to display the top level menu of the Built-in pages.



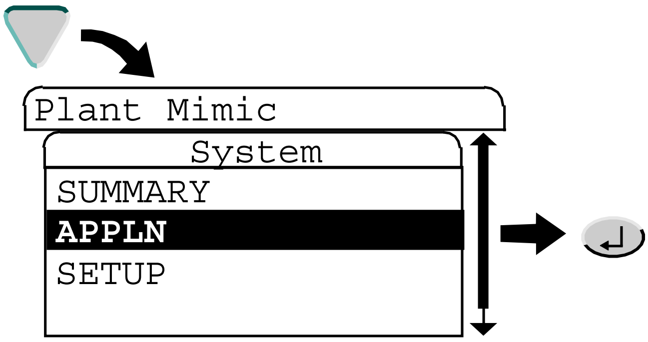

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.

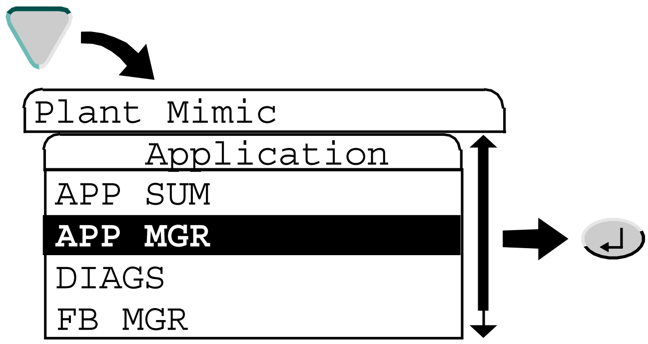



3. Again, using the Navigation keys select the APPLN (Application) option from the pop-up menu and confirm using the Return key.

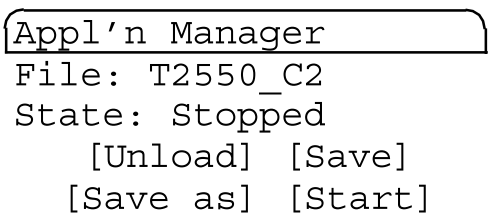
   The Application pop-up menu appears.




4. Finally, using the Navigation keys select the APP MGR option from the pop-up menu and confirm using the Return key.




5. The Application Summary page appears, showing:
   - the name of the loaded application
   - its state (RUNNING, IDLE, or STOPPED)
   - available commands



### 7.3.1 Stopping an application

With an application running, the Appl'n Manager page should look like this (except for the actual file name).

To stop the application,

1. Using the Navigation keys select the STOP option from the pop-up menu and confirm using the Return key.

   *Note. Stopping an application during a critical operation is not recommended.*

```
Appl'n Manager
File: T2550_C2
State: RUNNING
              [STOP]
```

2. The display confirms that the application has STOPPED.

   The commands at the bottom of the display offer the following options:

   UNLOAD the application, without first saving the application data, typically prior to selecting a new application.

```
Appl'n Manager
File: T2550_C2
State: Stopped
   [UNLOAD]  [SAVE]
  [SAVE AS]  [START]
```

   SAVE the application data, not including the current setpoint program, typically because the Cold-start values have changed (usually from the Terminal Configurator).

   SAVE AS a different filename. The application data, not including the current setpoint program, is saved using a different filename.

   START the application again.

   *Note. SAVE, SAVE AS and START are covered in the next section.*

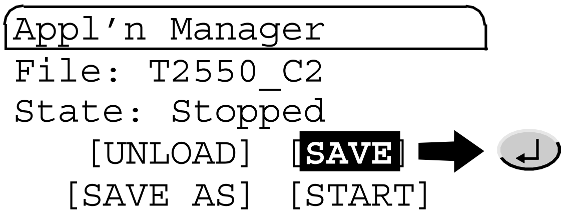
### 7.3.2 Saving an application

With the application stopped, the Appl'n Manager page should look like this (except for the actual file name).

To save the application,

1. Using the Navigation keys select the SAVE option from the pop-up menu and confirm using the Return key.

   The display indicates that the application has STOPPED.

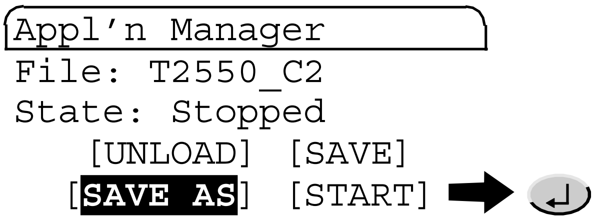



*Note. The progress bar 'Saving' page may be displayed momentarily, as this depends on the size of the LIN Database.*

To save the current application data under a different name.

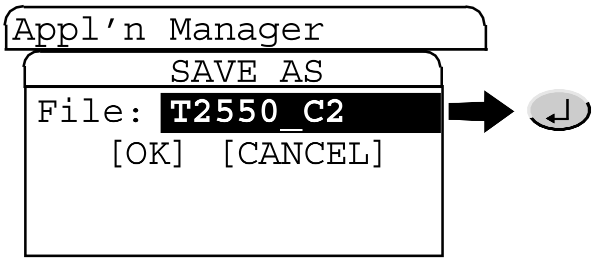
1. Using the Navigation keys select the SAVE AS option from the pop-up menu and confirm using the Return key.

   A 'SAVE AS' page appears




2. Select the 'File' field using the Navigation keys and confirm using the Return key.

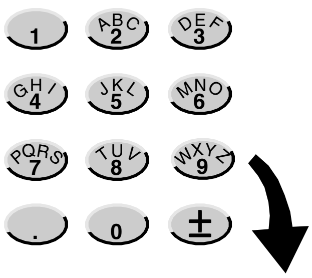
   The 'File' field starts to flash.

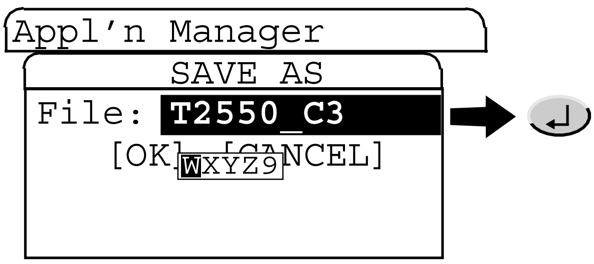



3. Enter each character/numeric required for the new filename, using the alphanumeric keys. Each key press selects the next alphanumeric in the list. The selected alphanumeric is only entered in the filename when the alphanumeric list disappears.

   *Note. A Filename is entered following the same principles as the mobile phone, using multiple keys presses to select the letter or number required.*

   Erroneous characters can be deleted by selecting the relevant character and pressing the appropriate alphanumeric key.




4. When complete, confirm using the Return key.
5. Use the Navigation keys to select OK to accept or CANCEL to ignore the Filename, and confirm using the Return key.

   The File is now saved with a different filename, in the T820.

### 7.3.2 Saving an application (Cont.)

To re-start the application,

*Note. The Appl'n Manager page reverts to its opening display, reporting the current application running.*

1. Using the Navigation keys select the START option from the pop-up menu and confirm using the Return key.
2. The display confirms that the application has STARTED.

*Note. START can be used to start other applications after they have been loaded.*

```
Appl'n Manager
File: T2550_C2
State: Stopped
   [UNLOAD]  [SAVE]
  [SAVE AS]  [START]
```

## 7.3.3 Unloading an application

With an application Stopped, the Appl'n Manager page should look like this (except for the actual file name).

To unload the application,

1. Using the Navigation keys select the UNLOAD option from the pop-up menu and confirm using the Return key.

*Note. The screen might go blank for a few seconds while the application unloads.*

```
Appl'n Manager
File: T2550_C2
State: Stopped
   [UNLOAD]  [SAVE]
  [SAVE AS]  [START]
```

2. The display confirms that the application has UNLOADED by the display of the LOAD, LD+RUN and DELETE options.

   This is referred to as the 'bare' panel.

   At this point choose either to load or load-and-run another application, or to delete an application.

```
Appl'n Manager
File: T2550_C2
   [LOAD]  [LD+RUN]
      [DELETE]
```

*Note. A default LIN Database will be automatically generated, when the Control Unit is powered up, and an application is not loaded.*

## 7.3.4 Loading or loading and running an application

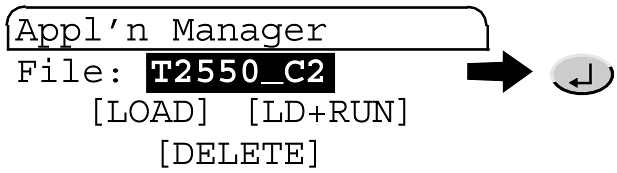
Before an application can be loaded, any previously-loaded application must have been stopped and unloaded. The Appl'n Manager page should look like this (except for the actual file name).

### APPLICATION SELECTION

To select the application,

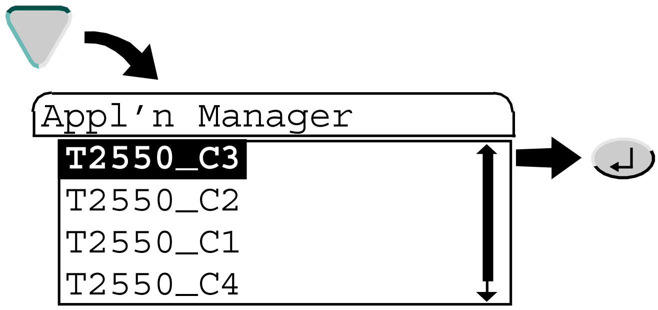
1. Using the Navigation keys select the 'File' field and confirm using the Return key.

   A list of available applications appears.

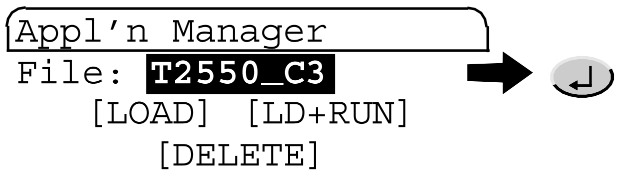



2. Using the Navigation keys select the required application from the pick list and press the Return key.

   The name is displayed in the File field.




3. The selected application is now ready to be loaded.




### APPLICATION LOADING

Before an application can be loaded, the application must be selected. The Appl'n Manager page should look like this (except for the actual file name).

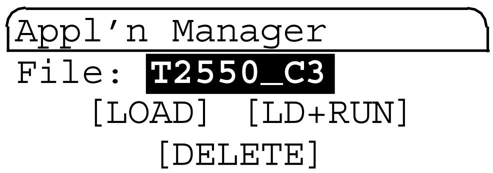
To load, or load and run the application,

1. Using the Navigation keys select the required option and confirm using the Return key.

   If LOAD is selected, there is a short delay before the page shows the name of the application and its state, IDLE.

   *Note. A loaded application must now be started.*

   If LD+RUN is selected, there is a short delay before the page shows the name of the application and its state, RUNNING. The same state can be achieved using LOAD, then START.




### 7.3.5 Deleting an application

Before an application can be deleted, the application must have been stopped and unloaded. The Appl'n Manager page should look like this (except for the actual file name).

*Note. To ensure the correct application is selected refer to Loading an Application.*

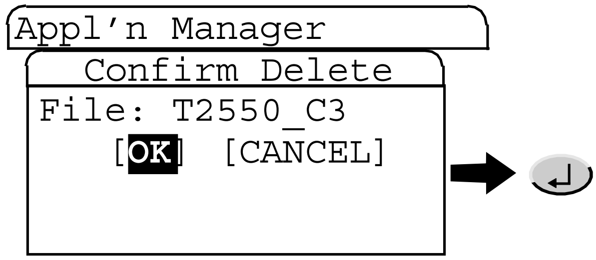
To delete the application,

1. Using the Navigation keys select the DELETE option and confirm using the Return key.

   A Confirm Delete page appears.


2. Using the Navigation keys select OK to accept or CANCEL to ignore the operation, and confirm using the Return key.

   The Appl'n Manager page shows that an application must be loaded.




3. Select the required application, see Application selection, and load, or load and run it, see Loading or loading and running an application.

## 7.4 THE FUNCTION BLOCK MANAGER (FB MGR) PAGE

The function blocks set up in LINtools for an application can be viewed by selecting the FB MGR option from the Application pop-up. Individual function block details can be displayed and edited.

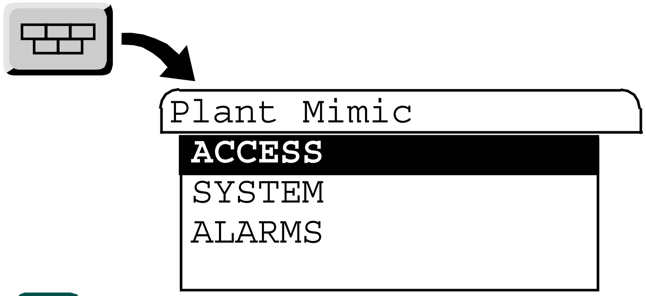
*Note. See the LINtools On-line Help file (Part no. RM263001U055) for configuring an application details and the LIN Blocks Reference Manual (Part no. HA082375U003) for function block specifics.*

This page controls the display of the selected function block, and consists of the following tasks,

- Editing a Function Block (section 7.4.1)

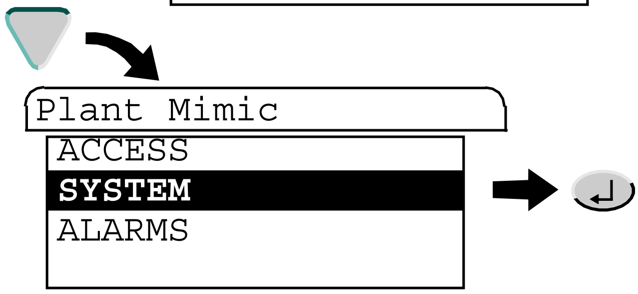
To display the Function Block Manager page,

1. Press the Menu key to display the top level menu of the Built-in pages.



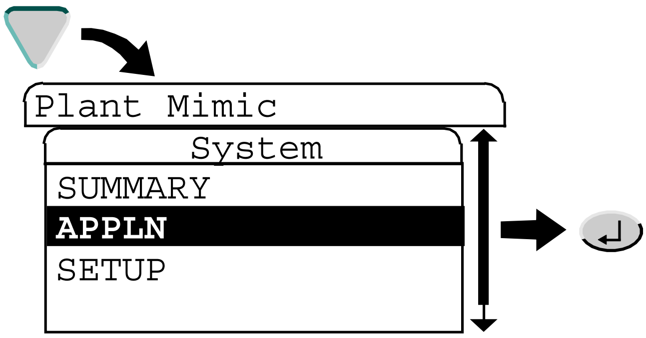

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.



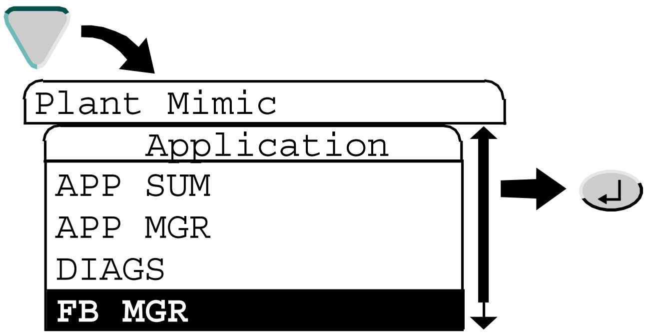

3. Again, using the Navigation keys select the APPLN (Application) option from the pop-up menu and confirm using the Return key.

   The Application pop-up menu appears.



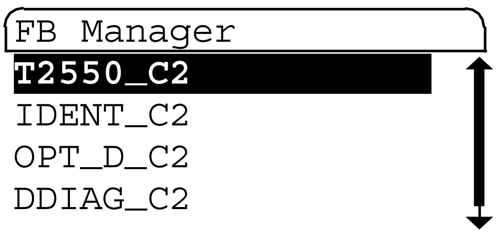

4. Finally, using the Navigation keys select the FB MGR option from the pop-up menu and confirm using the Return key.

   A list of all the Function Blocks in the application is displayed.




5. Function block field information can be displayed by selecting the required function block and confirming the selection.

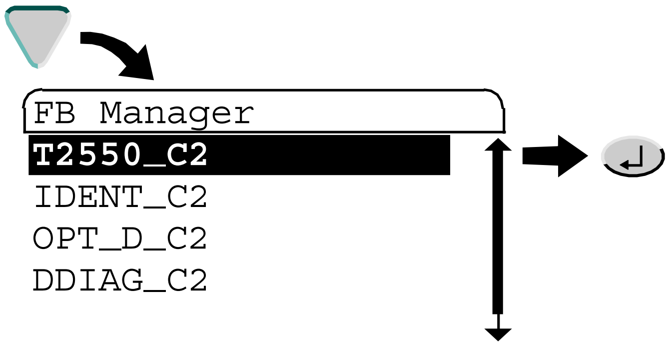
*Note. Use the Option key to toggle the list between LIN Database order and alphabetical order.*



### 7.4.1 Editing a Function Block

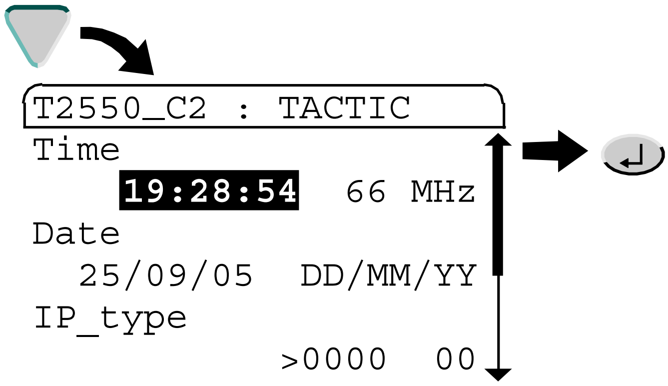
1. Using the Navigation keys select the required function block from the list and confirm using the Return key.

   The function block fields appear.




2. Using the Navigation keys select the required function block field and confirm using the Return key.

   The selected field starts to flash.



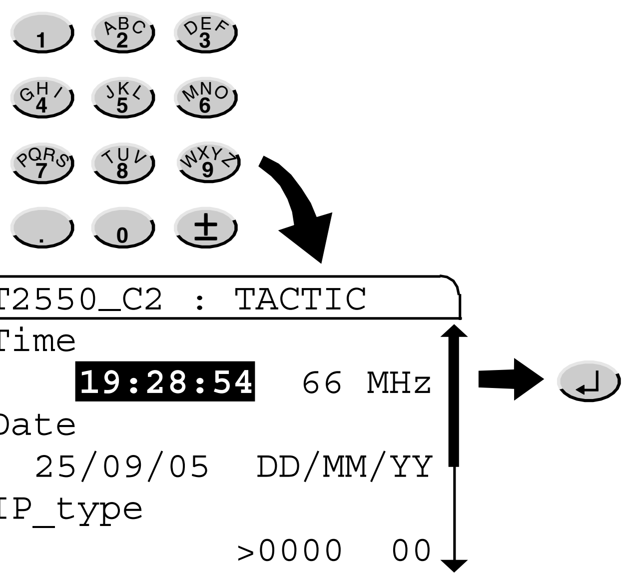

3. Enter the required values inputting each character/ numeric in turn, using the alphanumeric keys.

   Each key press selects the next alphanumeric in the list. The selected alphanumeric is only entered when the alphanumeric list disappears.

   *Note. A value is entered following the same principles as the mobile phone, using multiple keys presses to select the letter or number required.*

   Erroneous characters can be deleted by selecting the relevant character and pressing the appropriate alphanumeric key.

   If subfields exist, further pages are displayed. These can be edited using the same selection and confirmation methods as described in these steps. The page is closed by pressing the X key.




4. When complete, confirm using the Return key.

   The value is now entered in to the LIN Database.

# 7.5 CONTROLLING ACCESS

Access control consists of setting up (and changing, if necessary) the passwords for each of the following types of users:

- Operators
- Commissioning Engineers
- Engineers

The Gaining Access section, contains details of how the passwords are used to gain access to various parts of the instrument configuration. This section consists of the following tasks,

- First-time access (section 7.5.1)
- Editing the Passwords (section 7.5.2)

Using the instructions as described in Setting up the Panel display section, it is possible to set a time period (time-out) after which the access level returns to ‘Locked’.

## 7.5.1 First-time access

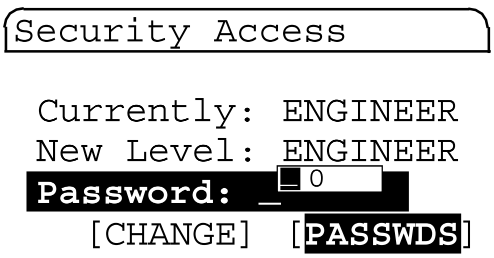
**LOCKED level:** For access to the LOCKED level at first-time or at any other time, no password is required.

**OPERATOR and COMMISSION (Commissioning Engineer) levels:** For first-time access, no password is required.

**ENGINEER level:** For first-time access, immediately after the instrument has been commissioned, the factory-set default password for Engineer-level access should be entered. This password is:

<space>default

i.e. a space character followed immediately by d e f a u l t (eight characters in all). The space is the first option on the 0 (zero) key.



### 7.5.2 Editing the passwords

Before a password can be changed, an Engineer access level must be entered.

*Note. Passwords can only be changed at ENGINEER access level.*

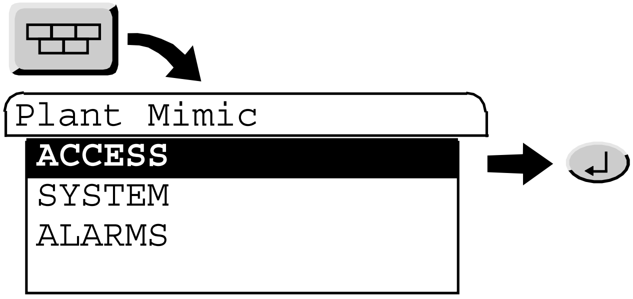
To change the passwords,

1. With the T820 powered up, press the Menu key to display the top level menu of the Built-in pages.

2. Using the Navigation keys select the ACCESS option and confirm using the Return key.

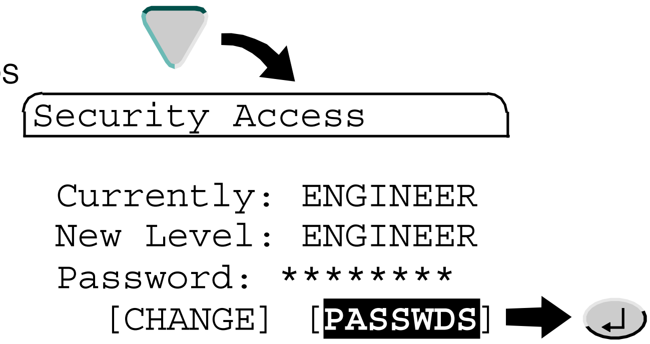
   The Security Access page appears, displaying the current access level (ENGINEER) and two options: CHANGE and PASSWDS.

   *Note. Gaining ENGINEER Access Level is required in order to change Passwords. The PASSWDS option is not available at any other Access Level.*



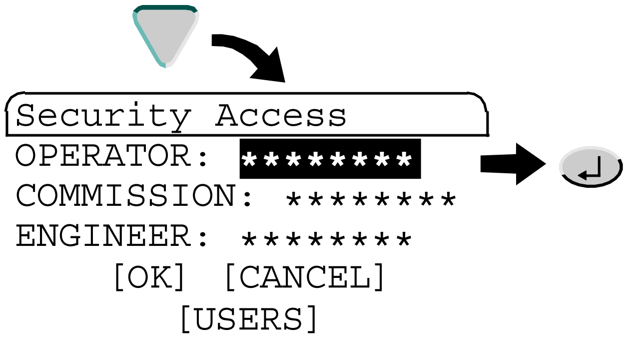

3. Again, using the Navigation keys select the PASSWDS option and confirm using the Return key.

   The Passwords page appears.




4. Using the Navigation keys select the Password field (shown as asterisks) you want to change (e.g. Operator) and confirm using the Return key.

   The asterisks are removed and Password field starts to flash.



### 7.5.2 EDITING PASSWORDS (Cont.)

5. Enter each character/numeric required for the new Password, using the alphanumeric keys.

   Each key press selects the next alphanumeric in the list. The selected alphanumeric is only entered in the password when the alphanumeric list disappears.

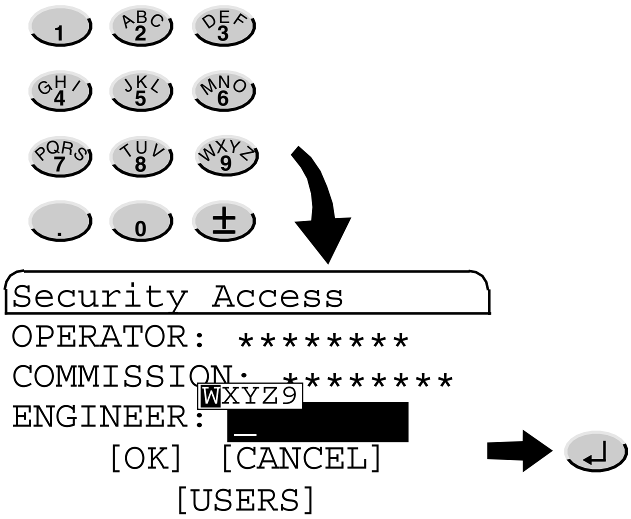
   *Note. A Password is entered following the same principles as the mobile phone, using multiple keys presses to select the letter or number required.*

   Erroneous characters can be deleted by selecting the relevant character and pressing the appropriate alphanumeric key.

   Confirm using the Return key.

   The Security Access page appears if changing the COMMISSION or OPERATOR passwords.

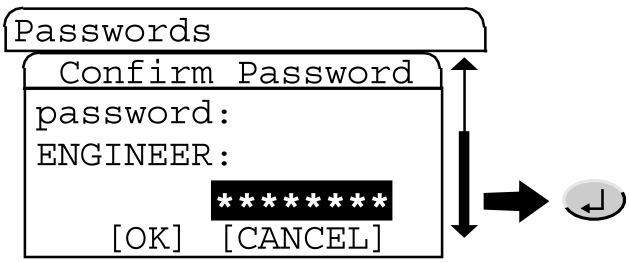
   *Note. If changing the ENGINEER password additional confirmation is required via the Confirm Password page.*



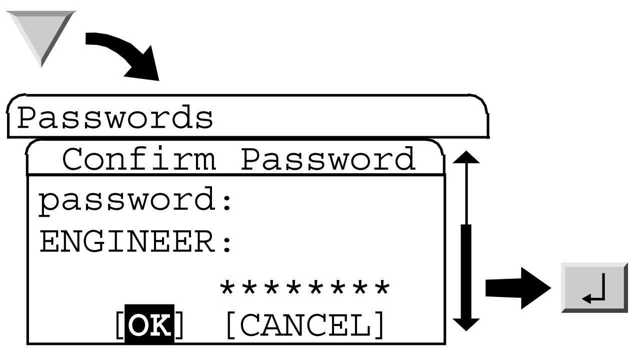

### EDITING ENGINEER ACCESS LEVEL PASSWORD

This requires further confirmation to ensure the security of the T820.

6. Again, using the Navigation keys select the Password field and, as previously described, use the alphanumeric keys to enter the new Password.




7. Finally, using the Navigation keys select OK to accept or CANCEL to ignore the operation, and confirm using the Return key.



## 7.6 SETTING UP THE INSTRUMENT

This section describes the operation required for setting up a unit, and consists of the following tasks,

- Editing Communications parameters (section 7.6.1)
- Setting the Startup strategy (section 7.6.2)
- Setting startup values (section 7.6.3)

*Note. Parameters displayed in these pages can also be edited using the Instrument Properties dialog, see Instrument Properties Online help (Part no. RM029278). The LIN Instrument Properties dialog can only be used if a PC is connected, as the LIN Instrument Properties dialog, is displayed from within an application, such as LINtools, or from the Project Explorer Properties command.*

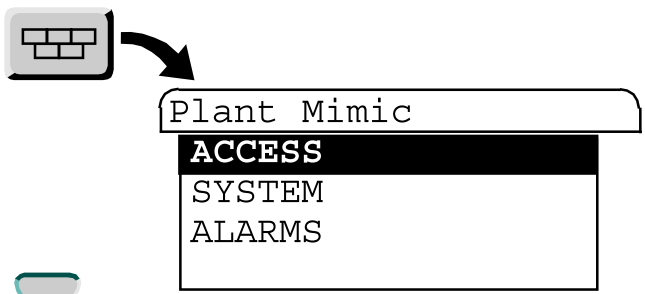
### 7.6.1 Editing communications parameters

Editing Communications parameters consists of displaying the Comms Setup page and setting up or editing the parameters for each port fitted.

*Note. Before any changes can take effect, the application must be stopped and then restarted, or the unit must be powered off and on again. Generally, 'parameter' value changes, i.e. baud rate, require only a stop and restart of the application, whereas 'hardware' changes, i.e. changing a Modbus master port to a slave port, require a power down and up operation.*

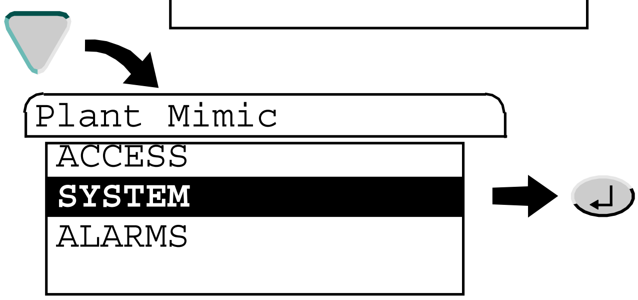
To edit the communications parameters,

1. Press the Menu key to display the top level menu of the Built-in pages.



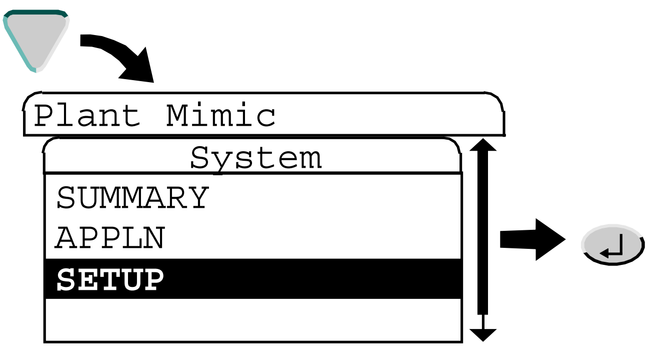

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.



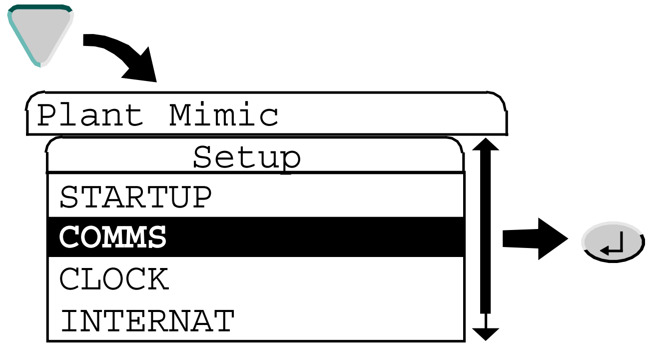

3. Again, using the Navigation keys select the SETUP option from the pop-up menu and confirm using the Return key.

   The Setup page appears.



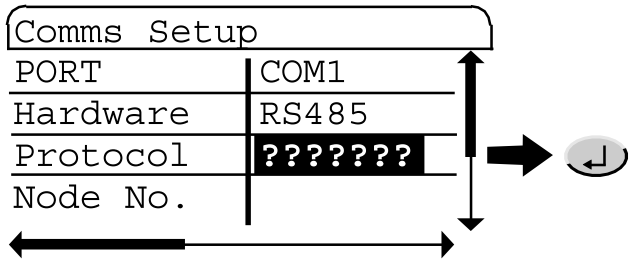
### 7.6.1 Editing communications parameters (Cont.)

4. Finally, using the Navigation keys select the COMMS option from the pop-up menu and confirm using the Return key.




5. The Comms Setup page appears.

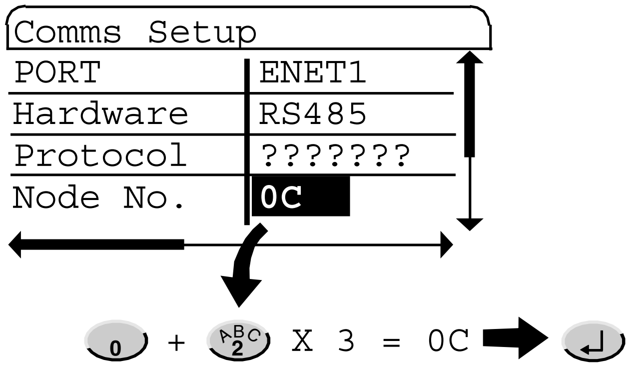
   Use the left and right Navigation keys to show each Port in turn (ENET1, …, ENET5, COM1, COM2). Use the up and down Navigation keys to display each of the Port parameters (Hardware, Protocol, Node No…).




The full parameter list is:

- Hardware Standard (for example, RS232)
- Protocol (for example, Modbus Slave, ELIN, FTP)
- Node Number (hex)

*Note. The 'A', 'B' or 'C' alphanumeric value is input by pressing the '2' numeric key the appropriate number of times, 'D', 'E', or 'F' alphanumeric value by pressing the '3' numeric key the appropriate number of times.*




Editing the Node Number Example

- Baud (rate)
- Parity
- Data bits (number of)
- Stop bits (number of)
- Timeout (Modbus Master only, in milliseconds)
- Talk Thru (Modbus Slave only - see Transparent Modbus Access (TMA/TalkThru) section)

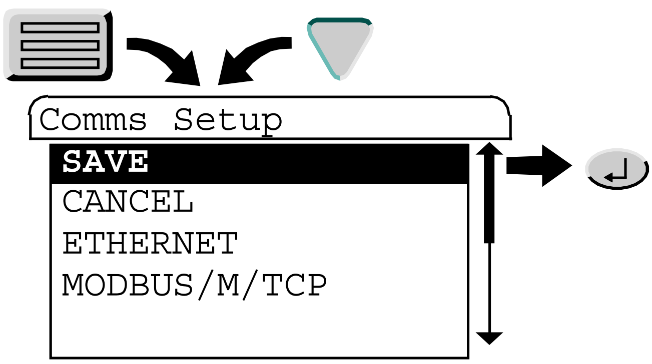
*Note. A blank cell indicates the parameter does not apply to the protocol selected for that port, and a cell containing '???????' indicates further configuration is required.*

6. Select the parameter that must be edited and confirm using the Return key.

   The selected field will either, flash to indicate a numeric value is required, or display a popup menu of available values. Edt the field as appropriate.

7. Press the Option key to display the Protocol pop-up menu.

8. Then, using the Navigation keys select SAVE from the pop-up menu and confirm using the Return key.



### 7.6.1 Editing communications parameters (Cont.)

#### EDIT THE ETHERNET PROTOCOL

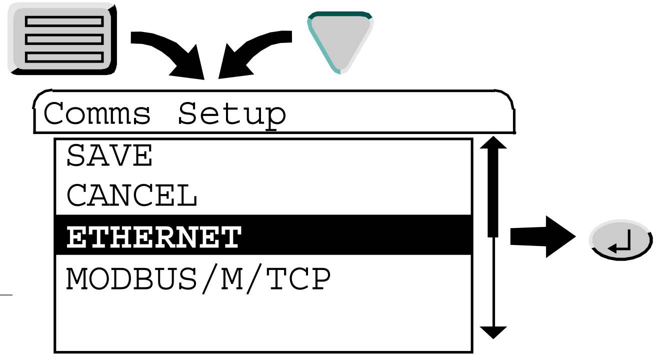
The Protocol for the selected Port can be changed by pressing the 'Option' key.

1. Press the Option key to display the Protocol pop-up menu.

2. Then, using the Navigation keys select the required Protocol from the pop-up menu and confirm using the Return key.

   An additional Comms - Protocol page appears displaying the parameters relevant to the selected Protocol.

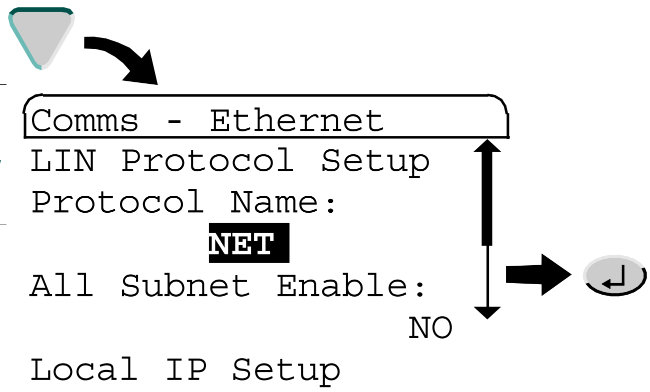
   *Note. The displayed Protocols depend on the current configuration of the instrument.*



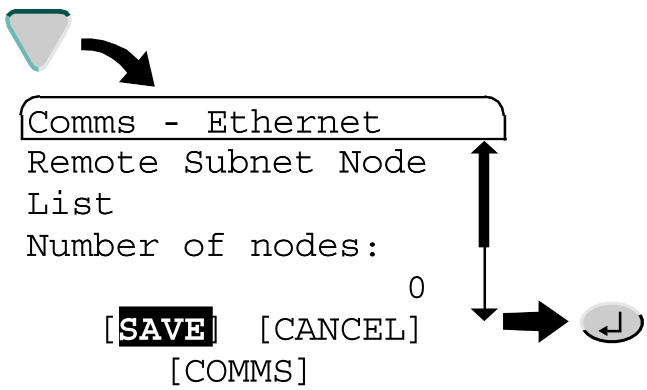

3. Again, using the Navigation keys select the required parameter and confirm using the Return key.

   *Note. Parameters that require numeric or alphanumeric input will flash to indicate that it can be edited. Some parameters will display a pop-up menu.*

   Use the appropriate method to configure the selected parameter and confirm using the Return key.




4. Finally, using the Navigation keys select the appropriate option to Save, Cancel or return to the page.



### 7.6.1 Editing communications parameters (Cont.)

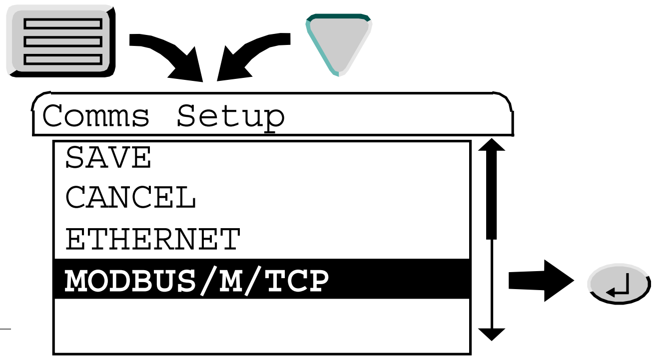
## EDIT THE MODBUS PROTOCOL

The Protocol for the selected Port can be changed by pressing the 'Option' key.

1. Press the Option key to display the Protocol pop-up menu.
2. Then, using the Navigation keys select the required Protocol from the pop-up menu and confirm using the Return key.

   An additional Comms - Protocol page appears displaying the parameters relevant to the selected Protocol.

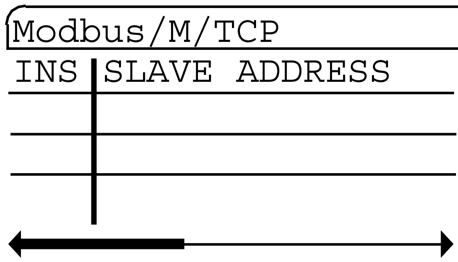
   *Note. The displayed Protocols depend on the current configuration of the instrument.*




3. Again, using the Navigation keys select the required parameter and confirm using the Return key.

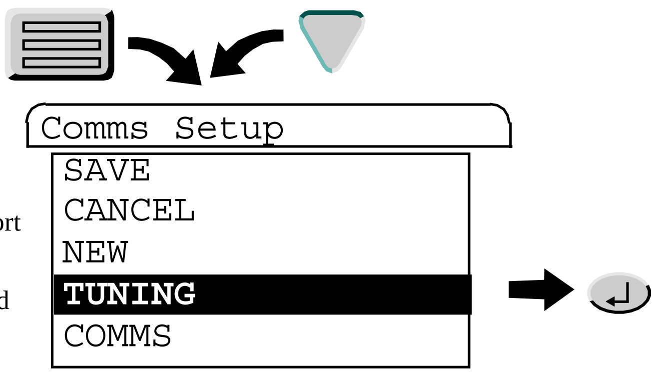
   *Note. The displayed parameters depend on the current configuration of the instrument.*

   Use the appropriate method to configure the selected parameter and confirm using the Return key.




- INS (Displays number of this Instrument)
- SLAVE ADDRESS (Displays the Slave Address)
- HOST (Displays the Host IP Address)
- TCP PORT (Displays the TCP Port configured for this instrument)

4. Press the Option key again, to display a Command pop-up menu with a list of commands. Use the Navigation keys select the required Command and configure as appropriate, confirm using the Return key.




- Save (Save the current changes to the selected Port and return to the Comms - Protocol page)
- Cancel (Ignore the current changes to the selected Port and return the Comms - Protocol page)
- New (Displays a further page that enables a new slave, including all relevant parameters to be configured)
- Tuning (Displays a further page showing the TCP Port Properties, also configurable via the Instrument Properties dialog)

  *Note. It is recommended that these fields are not edited, however, selecting the DEFAULTS command will reset the parameters to the factory setting.*

- Comms (Causes the screen to return to the Comms Setup page)

### 7.6.1 Editing communications parameters (Cont.)

#### SOFTWARE PARAMETER EDITING

1. Using the Navigation keys select the required parameter and confirm using the Return key.

   Confirmation will display either numeric values or a pick-list.

2. Select an option or enter the new value as required, using the appropriate method of configuration and confirm using the Return key.

#### HARDWARE CHANGES

1. Change the position of the Communications jumpers within the unit, see *Link Functions*.
2. With no application loaded, press the Option key while the Comms Setup page is displayed. This displays the Comms Setup menu.
3. Using the Navigation keys select the required Hardware option.

   *Note. Read and observe the instructions displayed before continuing with the Hardware check operation.*

   The screen redraws and shows the new settings. If these do not correspond to the required values, check that the jumpers have been set to the correct positions.

#### PROTOCOLS AVAILABLE

The communications Protocol can be configured for each Port fitted on the unit. It can also changed at any time, however, these changes will only become effective when specific criteria are met, as follows,

- ENET1, 2, 3, 4 On power up
- Serial On application start

*Notes.*
1. *Serial ports are always EIA422/485.*
2. *All Modbus protocols use 8 data bits.*
3. *The Node Number (Node No. - where presented) must be non-zero to enable the port.*

| Port | Protocol | Notes |
|---|---|---|
| COM1 | None, Termcfg, MODBUS/S, MODBUS/M | Used to provide serial communications support to the Termcfg (Telnet Terminal Configurator), Modbus/M (Modbus Master) or Modbus/S (Modbus Slave). |
| ENET1 | None, ELIN | Used to connect a Local Instrument Network (LIN) across Ethernet, only if the instrument can operate as a Control unit, see *Introduction*. |
| ENET2 | None, FTP | Used to provide an FTP server, only if the instrument can operate as a Control unit, see *Introduction*. |
| ENET3 | None, Modbus-S | The fullname of the protocol is 'Modbus RTU slave'. It is used for Modbus/TCP, but only if the instrument can operate as a Control unit, see *Introduction*. Supports the 'TalkThru' facility, and direct connection with a Modbus master. |
| ENET3 | None, Modbus-M | The full name of the protocol is 'Modbus RTU master'. |
| | | *Note. An additional Comms Setup configuration page is displayed for MODBUS/TCP master configuration, to assist with setting up IP Addresses for MODBUS/TCP slaves.* |
| ENET5 | Termcfg | Used to provide an ethernet communications to the Terminal Configurator via a Telnet session, only if the instrument can operate as a Control unit, see *Introduction*. |
| | | *Note. This will always be enabled.* |

Table 7.6.1 Available Protocols


HA029131
Issue 5 Nov 12

### 7.6.1 Editing communications parameters (Cont.)

#### ETHERNET

This only applies if the instrument is used to operate as a Control unit, see *Introduction*.

*Notes.*

1. *The user must have suitable access permission in order to edit the Ethernet setup.*
2. *Communications between Ethernet and Serial networks can be established using the Transparent Modbus Access (TMA or Talk Thru) facility, see Communications Manual (Part no. HA028014).*

Node location, including IP address configuration, is setup on the Network Settings page in the Instrument Properties dialog, see *Instrument Properties Online help* (Part no. RM029278) for details.

### 7.6.2 Setting the startup strategy

Hot-Start and Cold-Start define the automatic start parameters after a power failure or a power variation large enough to trigger an alarm (a 'Brown-out'). The control strategy is configured by specifying a Hot-start, Cold-start, both, or neither, and by setting periods for Hot-start and Brown-out time-outs.

#### HOT-START

A Hot-start uses data about the current application that is automatically saved in case of power variation or failure. Using this information, which is preserved through a power loss, the process can be restarted at any time after normal power returns.

In the fields on the Startup Strategy page, the following items need to be configured,

- whether the process is to start automatically after a power loss
- a maximum time period (time-out), after the expiry of which, a hot-start is inappropriate

#### COLD-START

Cold-start indicates that the T820 re-starts with the default LIN Database loaded, including all parameters and values set to starting values appropriate to the process (re-initialised). If the Cold-start fails the LIN Database will be cleared while the instrument indicates an 'Idle' state, remaining until physically restarted.

Parameter File

In the event of a cold-start, the instrument loads the .dbf file and searches for a .CPF file with the same name, and if found, the file is executed. The .CPF file consists of a list of parameters being processed in Structured Text (ST) style assignment statements (one complete statement per line of text) that,

- allocate the current Cold-start parameter values to function block fields.
- define the Reset Data Set values.

*Note. A .CPF file is a parameter overlay text file storing values that are initialised when a Cold-start is requested. If the it exceeds 10kB, the power hold-up may not be long enough to ensure all the data gets stored. LINtools can allow this file to be interrogated to determine the Cold-start parameters.*

Fields that are normally 'read only' can be written to from the .CPF file by adding the '>' character to the beginning of the assignment statement. A .CPF file can also include ST comment lines, e.g. (* Comment *). The header block includes a 'CPF alarm' to indicate if any problems were encountered whilst executing the .CPF file.

Sample .CPF file:

```
(* Production plant Cold Start Initialisation --- .CPF file *)
(* Ensure no automatic control until started *)
PIC-023.Mode := "Manual";
XCV-124.Mode := "Manual";

(* Ensure vent valves open *)
XCV-124.Demand := "False"; (* Open *)
XCV-123.Demand := "False"; (* Open *)

(* Reset profile to default *)
Profile.A0 := 23.4; (* Start temp Deg C *)
Profile.A1 := 34.5; (* First target temp Deg C *)
Profile.A2 := 2.0; (* Ramp rate Deg C / min *)

(* Initialise totalisation block*)
>COUNT-01.NTotal := 10;
>COUNT-01.NTotFrac := 0.5;
```

### 7.6.2 Setting the startup strategy (Cont.)

Reset Data Set

The .CPF file is also used to define the Reset Data Set. It can include parameters both with and without assignments, however only those with assigments cause values to be written on Cold-start.

```
PIC-023.Mode
XCV-124
Profile.A0
```

The Reset Data Set also includes the mode (MODE), local setpoint (SL) and output (OP) values of the control loop function blocks (PID, MODE, SETPOINT, MAN_STAT). The Reset Data Set defines the parameters which are unaltered in the .dbf file when performing a runtime save from the header block 'Options.SaveDBF'.

In the Reset Data package are a number of explicitly defined fields, specified in a .CPF file. The maximum total number of fields (including the SL, OP and MODE fields) is 2560.

## HOT/COLD START CRITERIA

The type of start depends upon the process, and on the operational policy of the user, e.g. some processes are so sensitive that a power-loss of any duration will cause the process plant, the load, or both, to need manual attention before restarting. Automatic Hot-start or Cold-start would be inappropriate.

Hot-start and Cold-start selected

The most common control strategy is to set both the Hot-start and the Cold-start to 'YES' so, if power returns before the time-out has expired, a Hot-start is attempted. If the time-out has expired, then a Cold-start power up is attempted.

Hot-start only

A power loss, or a 'brown-out' lasting long enough to trigger an automatic restart, which returns to normal before the Hot Start time-out, causes the T820 to attempt a Hot-start. If the power does not return to normal within the time-out period, a manual restart will be required, see *Saving an application* section

Cold-start only

If a power loss, or a 'brown-out' lasting long enough to trigger an automatic restart occurs, provided that power returns before the Hot-start time-out interval, the instrument will attempt a Cold-start power up.

Neither Hot-start nor Cold-start

If both Hot-start and Cold-start are set to 'No' the T820 will automatically generate a LIN Database including a limited number of function blocks.

## TIME-OUT PERIOD SELECTION

Hot-start time

This depends upon the process under control. If the process can tolerate only a short time without normal power before either the plant or the load requires manual attention, then a short time-out needs to be set. If, however, the process is robust enough to regain normal processing conditions even after a lengthy power-outage, then a longer Hot-start time-out may be set.

***IMPORTANT*** ***Actual Hot-start times are process-dependent, but as a general rule, the process must not restart automatically beyond the time when manual attention is required.***

Brown-out time

This sets an alarm when a power-variation has persisted for longer than a preset time. Unless the alarm is set up to take some action, the Brown-out time acts only as a warning, in case some special control strategies exist that may need implementing in those circumstances, or that have been set up to run automatically.

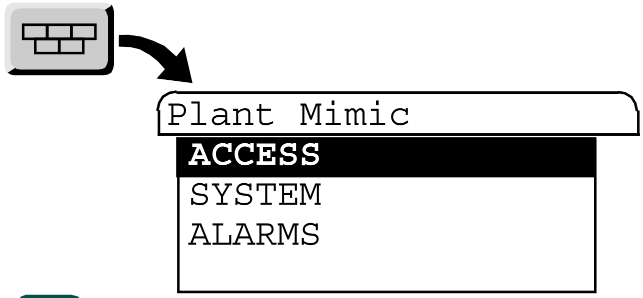
Any total power failure, that returns within the period specified as Brown-out time is treated as such. If it returns after the Brown-out time, a restart is either possible or certain, depending on how soon after the time limit it returns.

***IMPORTANT*** ***The type of restart attempted depends on the programmed control strategy.***

### 7.6.2 Setting the startup strategy (Cont.)

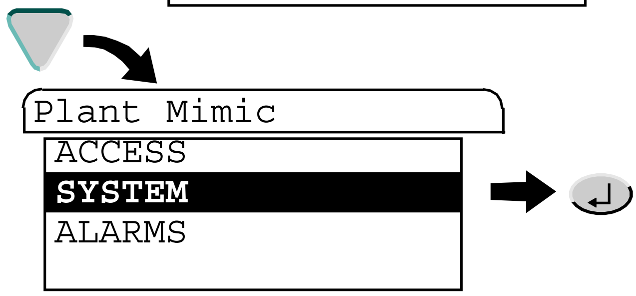
**STARTUP STRATEGY PAGE**

1. Press the Menu key to display the top level menu of the Built-in pages.



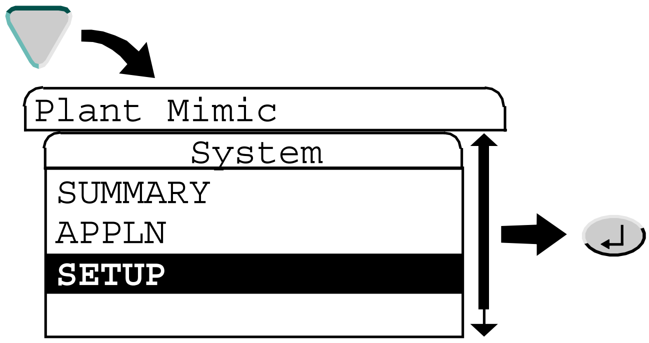

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.



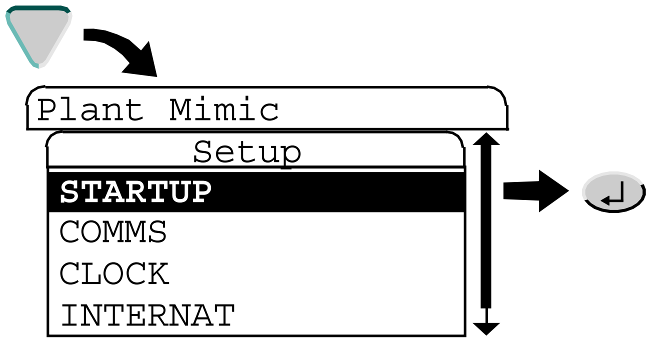

3. Again, using the Navigation keys select the SETUP option from the pop-up menu and confirm using the Return key.

   The Setup page appears.

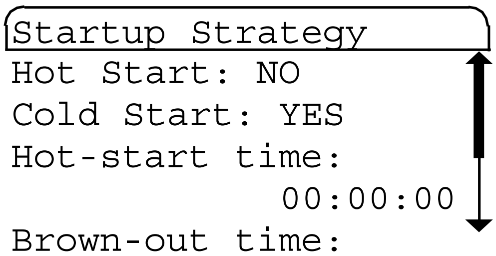



4. Finally, using the Navigation keys select the STARTUP option from the pop-up menu and confirm using the Return key.

   The Startup Strategy page appears.




5. Each specific field can now be edited, as required.

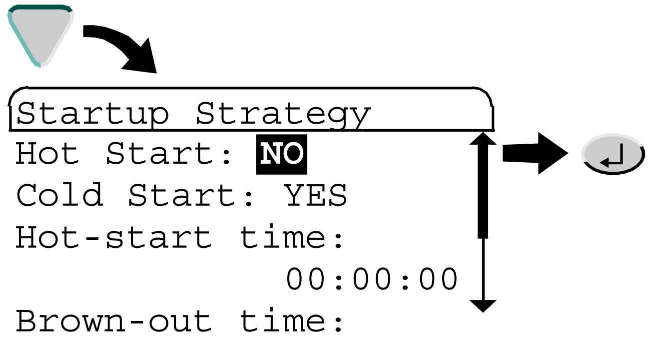


## 7.6.3 Setting startup values

### CONFIGURE HOT-START AND COLD-START SETTINGS

1. Using the Navigation keys select the Hot-start or Cold-start fields, as required and confirm using the Return key.

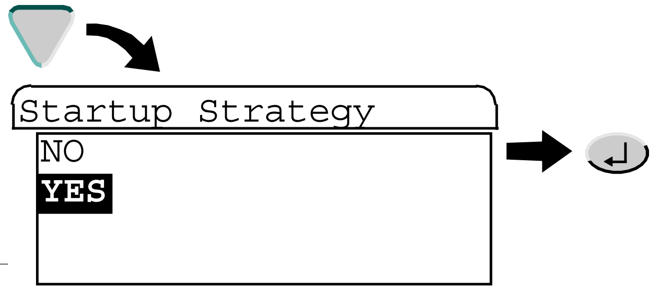
   A Yes/No pick-list appears.




2. Using the Navigation keys select the required option and confirm using the Return key.

   The pick-list disappears and the Startup Strategy page displays the new value.

3. Using the Navigation keys select the 'SAVE' option, and confirm using the Return key.

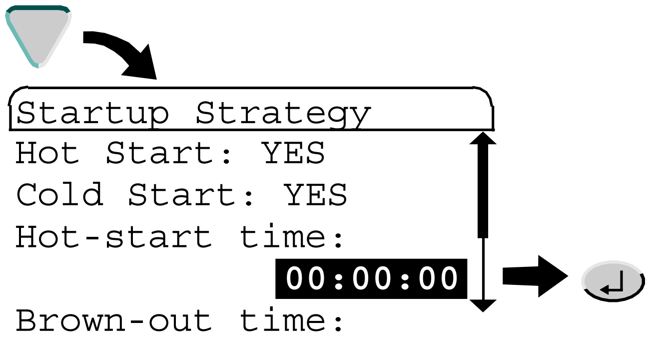



> *Note. Setting both Hot-start and Cold-start parameters to 'NO' will automatically create a LIN Database.*

### CONFIGURE TIME-OUT VALUES

1. Using the Navigation keys select the Hot-start time field, and confirm using the Return key.

   The Hot-start time field starts to flash.

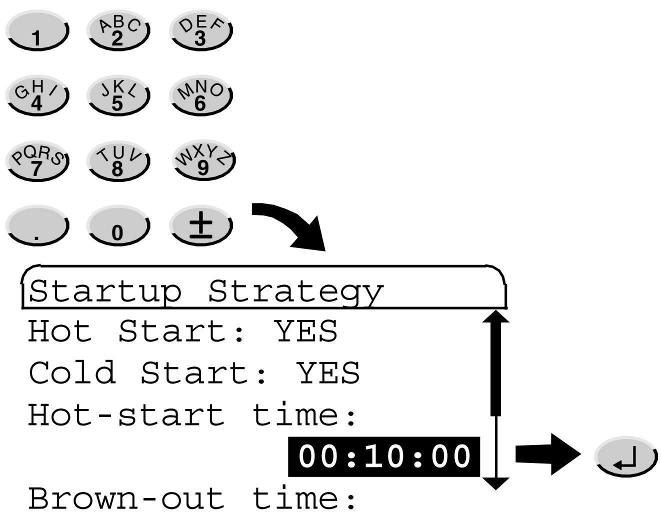



2. Enter each numeric of the required value, using the alphanumeric keys, following the format indicated.

   Confirm using the Return key.

   The pick-list disappears and the Startup Strategy page displays the new value.

3. Using the Navigation keys select the 'SAVE' option, and confirm using the Return key.



## 7.7 THE CLOCK SETUP PAGE

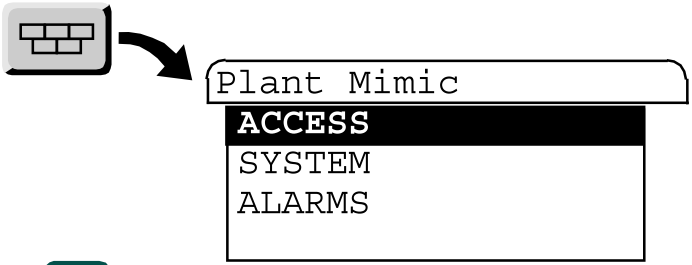
This allows the Real-Time Clock to be configured manually.

If the instument is configured as a Time Of Day slave, it automatically receives the time and date from an external source. By default, instrument LIN Databases that do not contain the TOD_DIAG block operate in TOD (time-of-day) slave mode. An instrument can act as a TOD slave to a cross-subnet master, by mapping to the cross-subnet master, see *The ELIN User Guide* (Part no. HA082429).

This page controls the display of the clock, and consists of the following tasks,

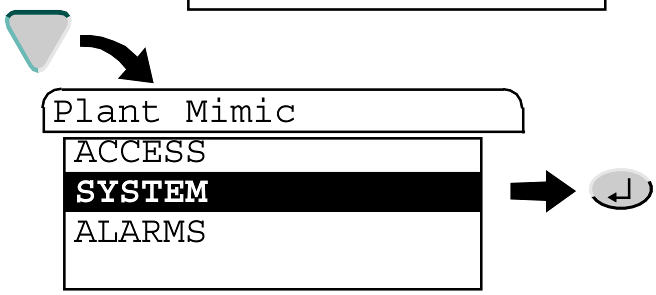
- Changing Date and Time (section 7.7.1)

1. Press the Menu key to display the top level menu of the Built-in pages.



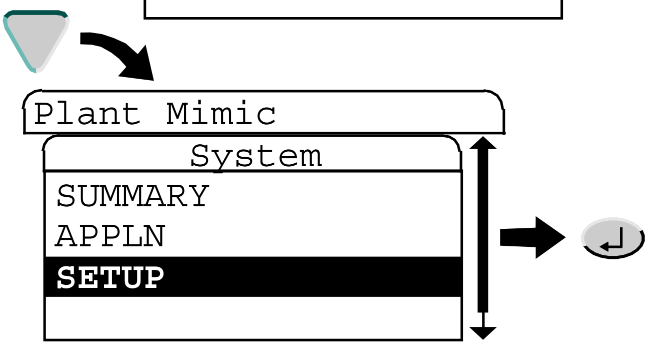

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System page appears.

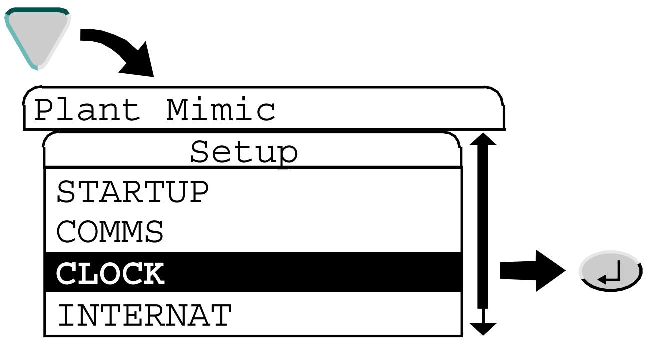



3. Again, using the Navigation keys select the SETUP option from the pop-up menu and confirm using the Return key.

   The Setup page appears.

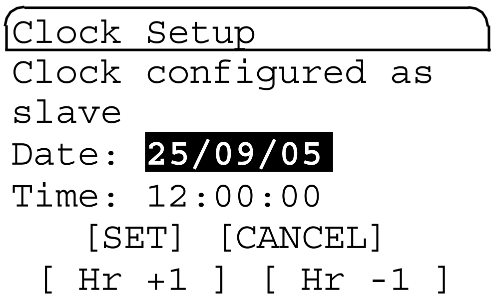



4. Finally, using the Navigation keys select the CLOCK option from the pop-up menu and confirm using the Return key.




5. The Clock Setup page appears.

   Each specific field can now be edited, as required.



### 7.7.1 Changing Date and Time

To increment or decrement the hours value, press the Hr+1 or the Hr-1 key respectively.

The change takes place immediately, and changes the date if appropriate.

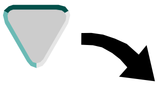
*Notes.*

1. *Operating the SET key enters the time and date displayed on the page. These values are not updated in real-time, but show the values obtaining when the page was called to the screen. As it is not possible to SET the date separately from the time, it is recommended that the date be changed first, then the time.*
2. *The clock re-starts when the SET is selected. This happens after the time has been keyed in and after the Return key has been pressed. It is therefore recommended that the keyed-in time is at least 20 seconds ahead of real time, so that SET can subsequently be operated (to start the clock) when real-time equals the keyed-in time.*
3. *For systems configured to have their clocks synchronised by another network node, it is not possible to edit the time or date if the master clock is running.*

**DATE CHANGING**

1. Using the Navigation keys select the DATE field and confirm using the Return key.

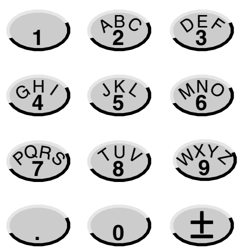
   The Date field starts to flash.



```
Clock Setup
Clock configured as
slave
Date: 18/01/01
Time: 12:00:00
    [SET] [CANCEL]
 [ Hr +1 ] [ Hr -1 ]
```

2. Enter each numeric of the required value, using the alphanumeric keys, following the format indicated.

   Confirm using the Return key.



3. Using the Navigation keys select the 'SET' option, and confirm using the Return key when the actual time is the same as the time just entered to re-start the clock.

```
Clock Setup
Clock configured as
slave
Date: 25/09/05
Time: 12:00:00
    [SET] [CANCEL]
 [ Hr +1 ] [ Hr -1 ]
```

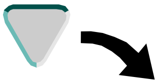
### 7.7.1 CHANGING DATE AND TIME (Cont.)

#### TIME CHANGING

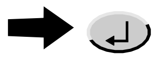
Before starting, please see note 3 on the previous page.

1. Using the Navigation keys select the TIME field and confirm using the Return key.

   The Time field starts to flash.

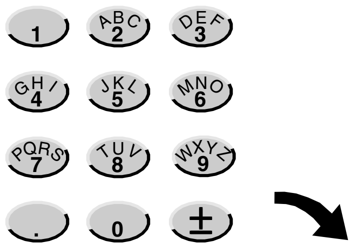


```
Clock Setup
Clock configured as
slave
Date: 18/01/01
Time: 12:00:00
    [SET] [CANCEL]
  [ Hr +1 ] [ Hr -1 ]
```



2. Enter each numeric of the required value, using the alphanumeric keys, following the format indicated.

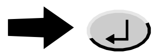
   Confirm using the Return key.





3. Using the Navigation keys select the 'SET' option, and confirm using the Return key when the actual time is the same as the time just entered to re-start the clock.

```
Clock Setup
Clock configured as
slave
Date: 18/01/01
Time: 23:59:00
    [SET] [CANCEL]
  [ Hr +1 ] [ Hr -1 ]
```



## 7.8 CHANGING LANGUAGE AND DATE/TIME FORMATS

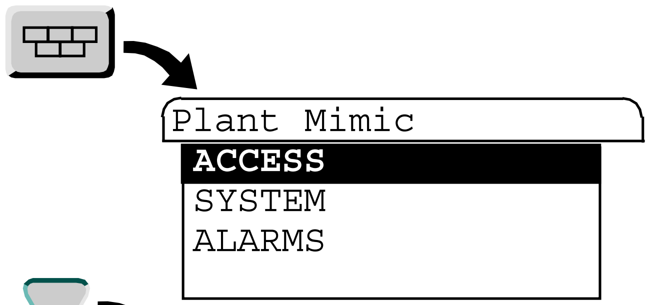
A different language can be selected only if the instrument holds the appropriate language dictionary file.

*Note. Parameters displayed in these pages can also be edited using the Instrument Properties dialog, see Instrument Properties Online help (Part no. RM029278).*

This page controls the T820 display, and consists of the following tasks,

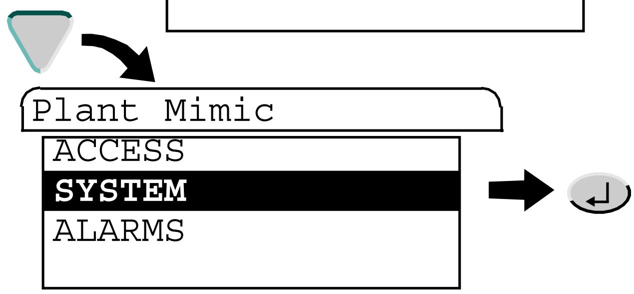
- Changing the language (section 7.8.1)
- Changing the Date Format (section 7.8.2)
- Changing the Time Format (section 7.8.3)
- Changing the Duration Fmt (section 7.8.4)

1. Press the Menu key to display the top level menu of the Built-in pages.



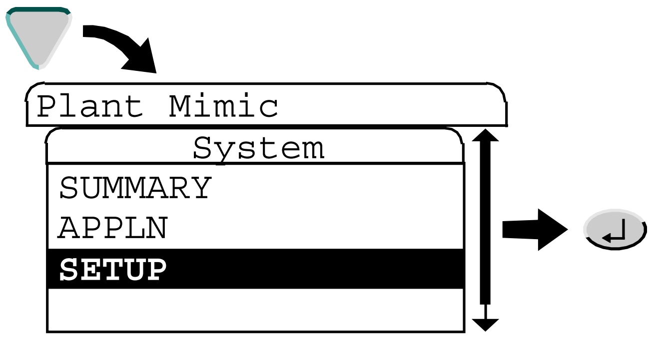

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.

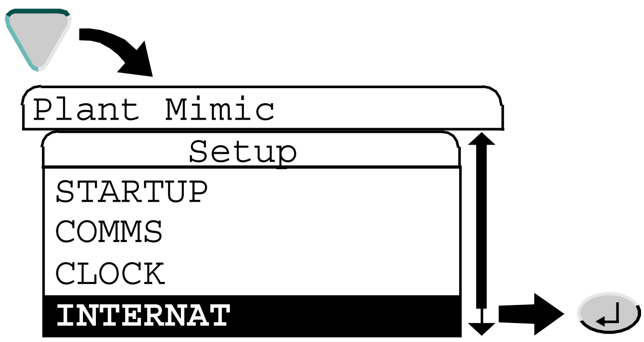



3. Again, using the Navigation keys select the SETUP option from the pop-up menu and confirm using the Return key.

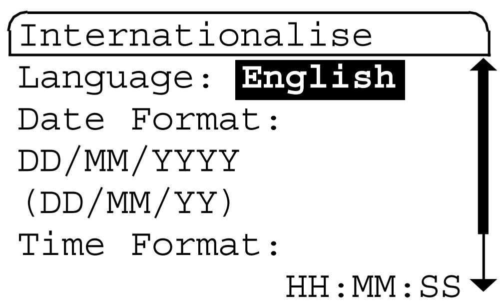
   The Setup page appears.




4. Finally, using the Navigation keys select the INTERNAT option from the pop-up menu and confirm using the Return key.



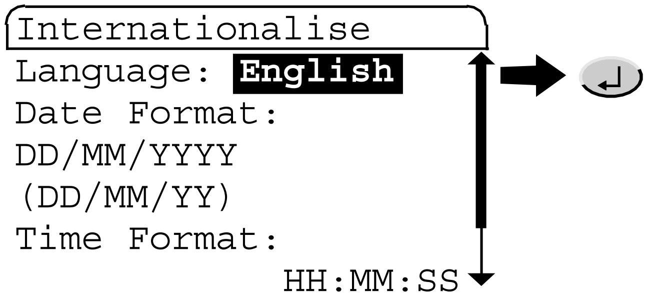

5. The Internationalise page appears.



### 7.8.1 Changing the language

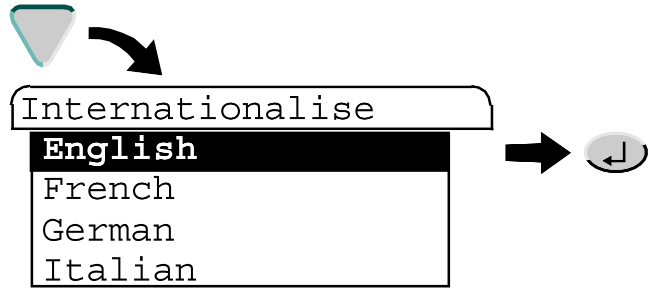
1. Using the Navigation keys select the Language option from the page and confirm using the Return key.

   The pick-list showing all available Languages appears.



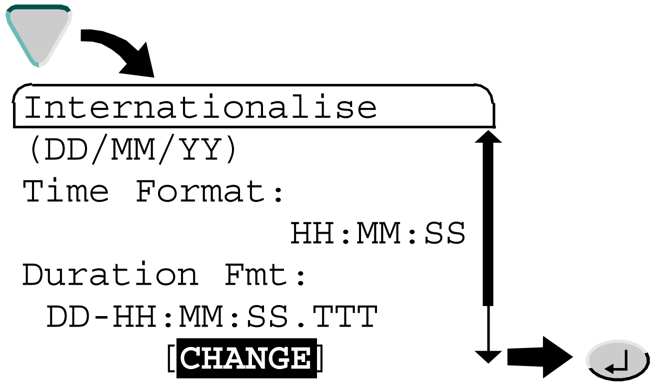

2. Using the Navigation keys select the Language required and confirm using the Return key.

   The Internationalise page appears.




3. Finally, using the Navigation keys select the CHANGE field and confirm using the Return key.

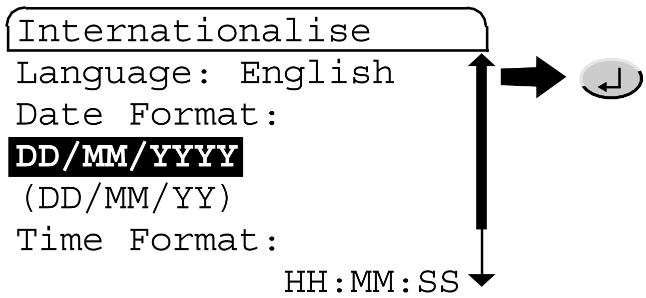
   The language is now selected.



## 7.8.2 Changing the Date Format

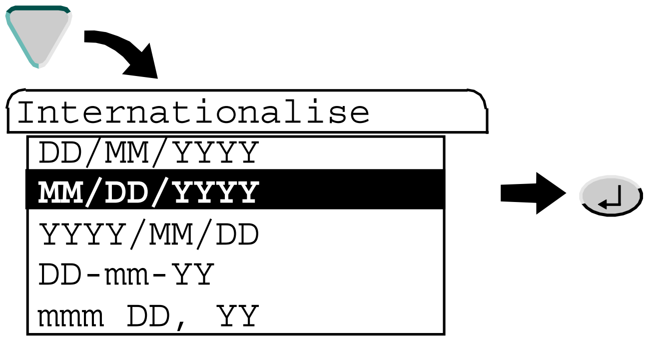
1. Using the Navigation keys select the Date Format option from the page and confirm using the Return key.

   The Date Format pick-list appears.



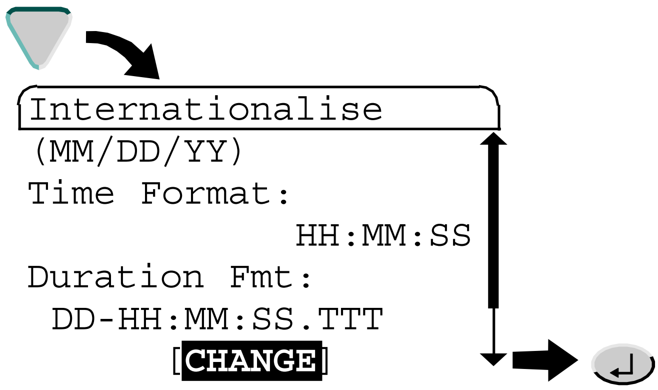

2. Using the Navigation keys select the Date Format required and confirm using the Return key.

   The Internationalise page appears.




3. Finally, using the Navigation keys select the CHANGE field and confirm using the Return key.

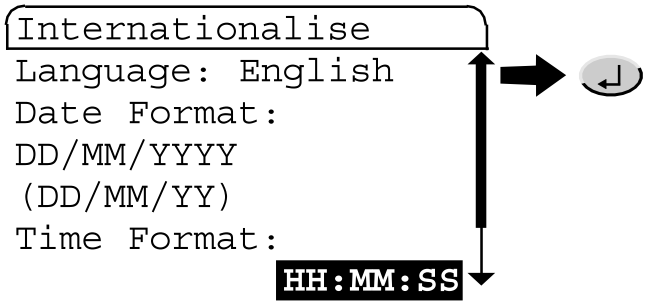
   The Date Format field shows the selected format.



### 7.8.3 Changing the Time Format

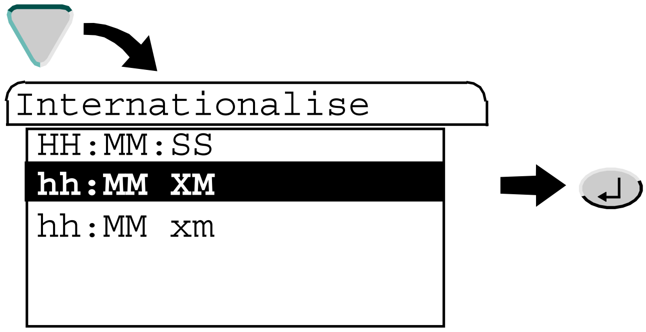
1. Using the Navigation keys select the Time Format option from the page and confirm using the Return key.

   The Time Format pick-list appears.



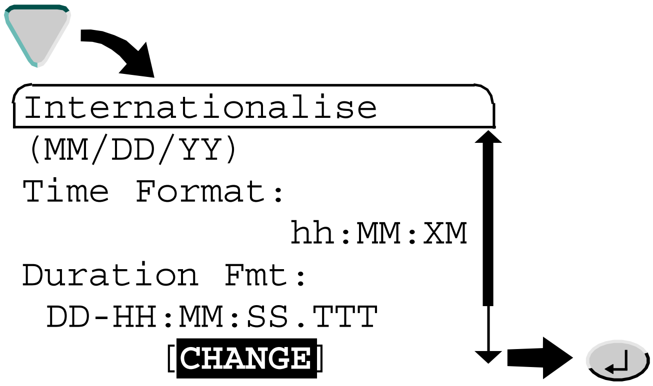

2. Using the Navigation keys select the Time Format required and confirm using the Return key.

   The Internationalise page appears.




3. Finally, using the Navigation keys select the CHANGE field and confirm using the Return key.

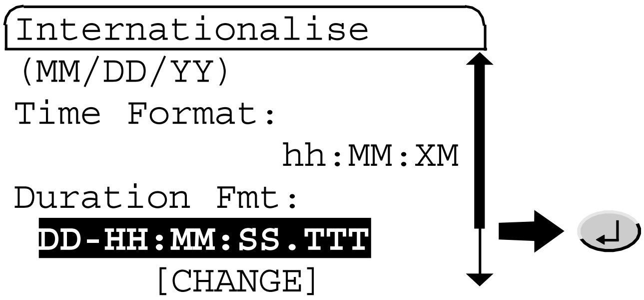
   The Time Format field shows the selected format.



### 7.8.4 Changing the Duration Fmt

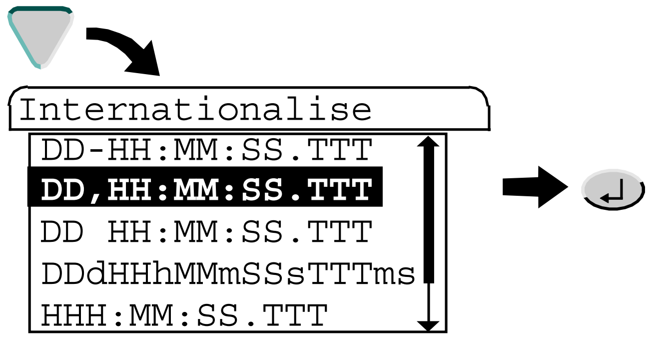
1. Using the Navigation keys select the Duration Format option from the page and confirm using the Return key.

   The Duration Format pick-list appears.



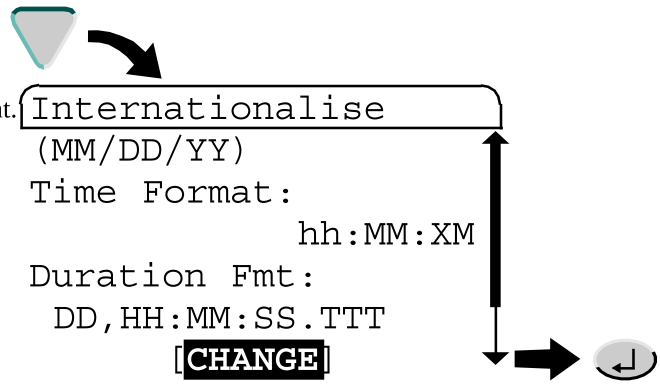

2. Using the Navigation keys select the Duration Format required and confirm using the Return key.

   The Internationalise page appears.




3. Finally, using the Navigation keys select the CHANGE field and confirm using the Return key.

   The Duration Format field shows the selected format.



## 7.9 SETTING UP THE PANEL DISPLAY

This page displays the T820's current screen and time-out setup.

*Note. Parameters displayed in these pages can also be edited using the Instrument Properties dialog, see Instrument Properties Online help (Part no. RM029278).*

From the Panel Setup page you can change the,

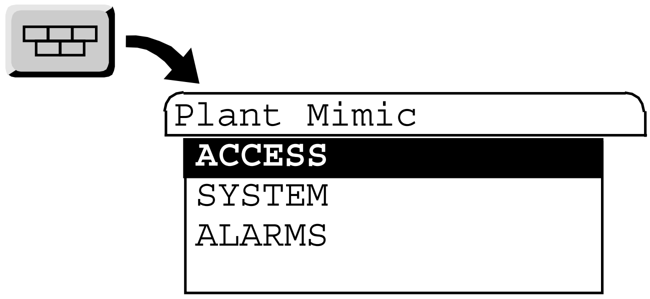
- Contrast properties (section 7.9.1)

  Contrast. This has the effect of scaling the relative colour strength of the screen.

- Page time-out values (0 = no time-out) (section 7.9.2)

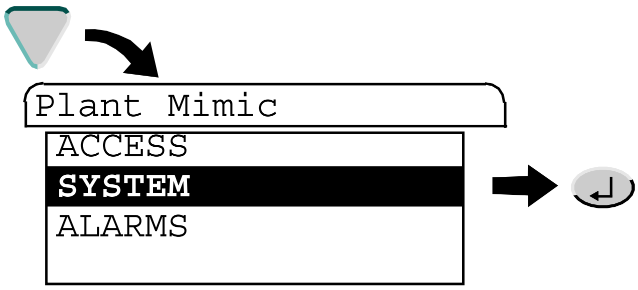
  Home (for any Home pages)
  Pop-up (for the Pop-up menu)
  Data Entry (for the pick-lists)

1. Press the Menu key to display the top level menu of the Built-in pages.



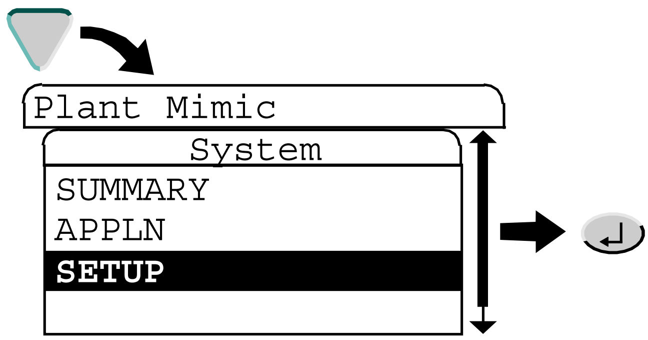

2. Then, using the Navigation keys select the SYSTEM option from the pop-up menu and confirm using the Return key.

   The System pop-up menu appears.

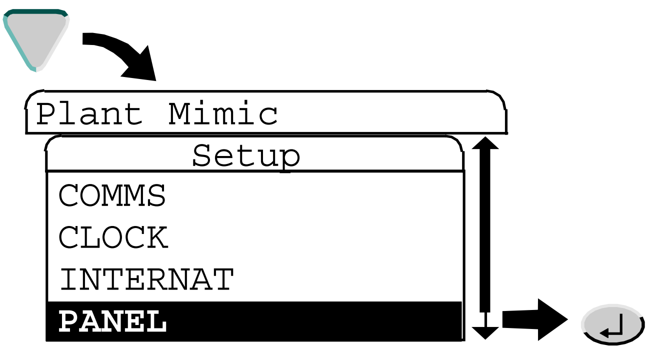



3. Again, using the Navigation keys select the SETUP option from the pop-up menu and confirm using the Return key.

   The Setup page appears.




4. Finally, using the Navigation keys select the PANEL option from the pop-up menu and confirm using the Return key.

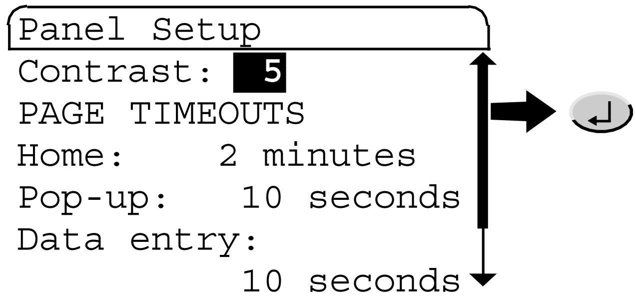



5. The Panel Setup page appears.

### 7.9.1 Contrast settings

1. Using the Navigation keys select the Contrast field and confirm using the Return key.

   The Contrast field begins to flash

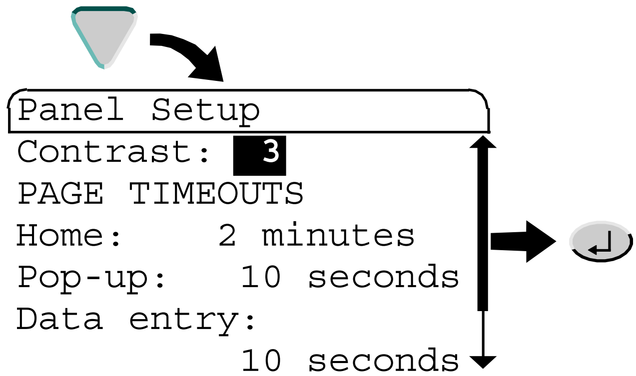



2. Using the Up and Down Navigation keys to increase (up to 10) or decrease (down to 1) the relative colour strength of the screen and confirm using the Return key.

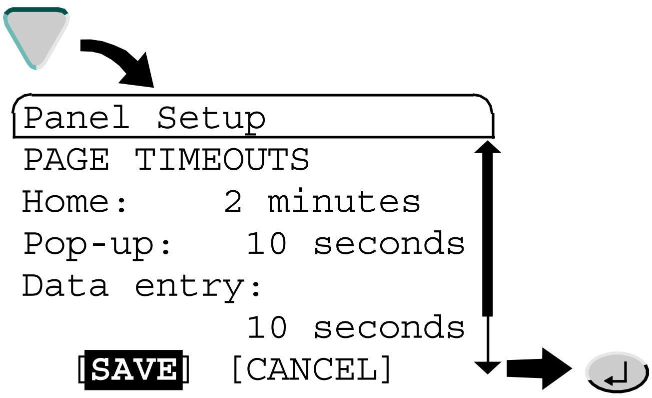
   ***IMPORTANT The screen text may become illegible when setting the Contrast to the extents of the limit.***

   > *Note. A value may also be entered using the alphanumeric keypad.*

   The Panel Setup page appears.



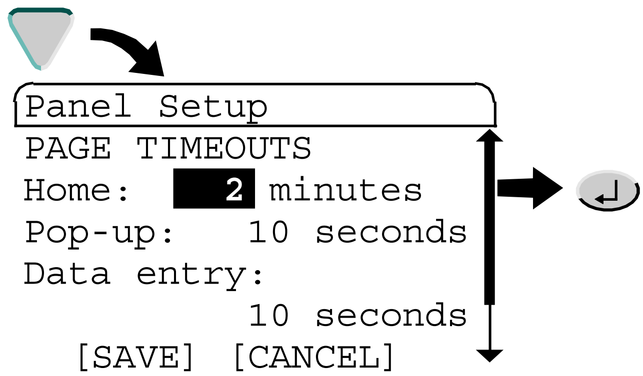

3. Finally, using the Navigation keys select the SAVE field and confirm using the Return key.



### 7.9.2Time-outs

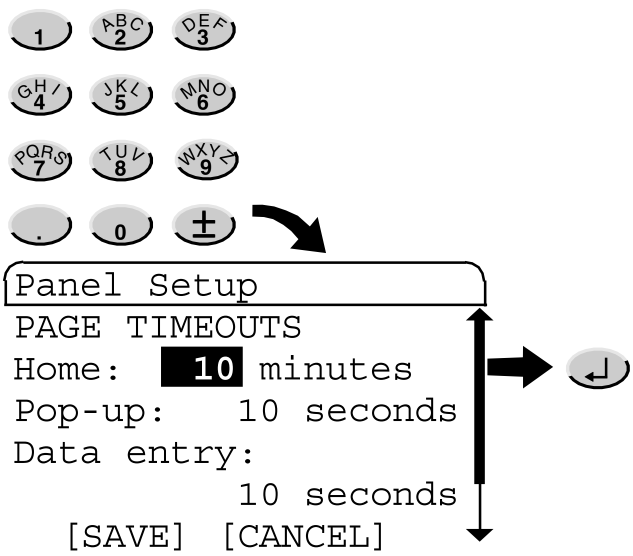
1. Using the Navigation keys select the required Timeout field and confirm using the Return key.

   The selected Timeout field starts to flash.



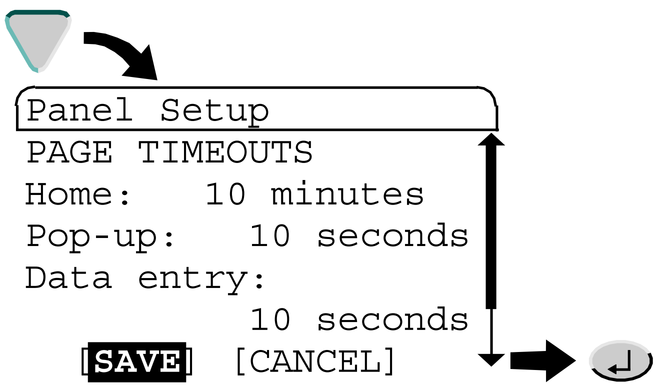

2. Enter each numeric of the required value, using the alphanumeric keys, observing the indicated.

   Confirm using the Return key.




3. Finally, using the Navigation keys select the SAVE field and confirm using the Return key.

   The Timeout value is saved.



## 7.10 THE FILE MANAGER PAGE

The File Manager allows the copying of files in the internal Flash memory. It also allows files to be deleted from the internal memory and consists of the following tasks,

- Copy and Delete a File (section 7.10.1)

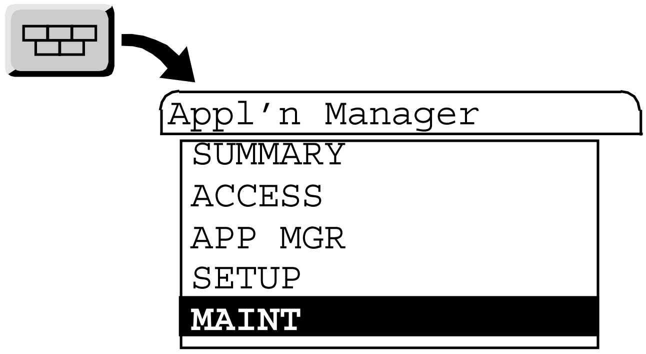
Before the File Manager page can be displayed, any application must be stopped and unloaded. Once the application has been stopped and unloaded, the File Manager can be displayed via the 'Maint' (Maintenance) option.

1. Stop and unload the application, see the Stopping an application and Unloading an application sections.

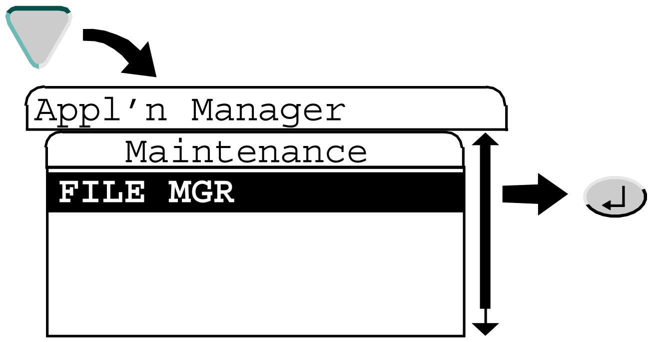
   The Appl'n Manager page appears.


2. Press the Menu key to display a pick-list. Using the Navigation keys select the MAINT option from the pick-list and confirm using the Return key.

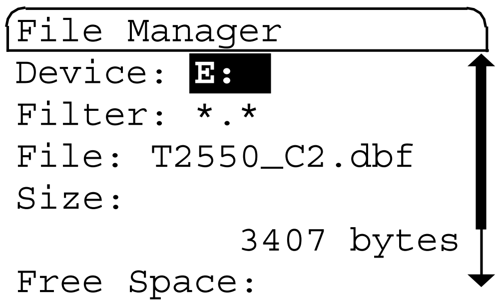
   The Maintenance page appears.




3. Using the Navigation keys select the FILE MGR option from the Maintenance page and confirm using the Return key.




4. The File Manager page appears.




| | |
|---|---|
| Device | E: (internal flash memory) only. |
| Filter | Allows file display to be limited to certain types of files. For example, *.* shows all files, whereas, an entry of *.dbf allows only files with .dbf extension to appear. See 'File'. |
| File | Selecting this option causes a scroll list of files to be displayed. Any file can be selected (one at a time) for copying or deleting. The range of files displayed can be limited by entering a display 'filter' to limit the scroll list to certain file names or file types. |
| Size | Shows the size of the selected file. |
| Free space | Shows the remaining capacity of the drive selected. |

### 7.10.1 Copy and Delete a File

Selecting the FILE MGR option in the 'Maintenance' pop-up calls the File Manager page. Once this page is on display, a file name has to be selected from the relevant drive, then the copy or delete option can be selected.

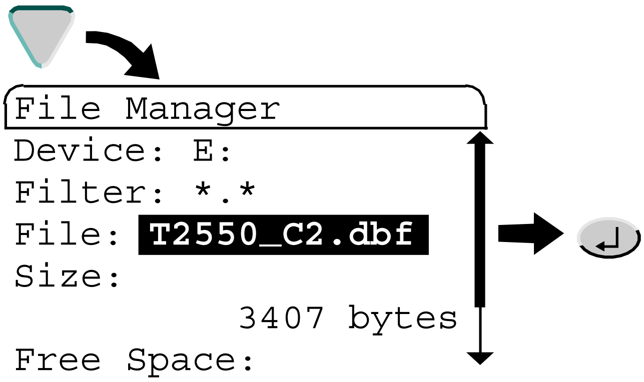
CAUTION

1. Files of the form _SYSTEM.xyz* must not be deleted or the instrument will not operate correctly and revert to a factory configuration.
2. The file _DEFAULT.ofl must not be deleted or the faceplates in the overview page will fail to operate correctly,
   * .xyz is any three character extension.

1. Using the Navigation keys select the FILE MGR option from the Maintenance page and confirm using the Return key.

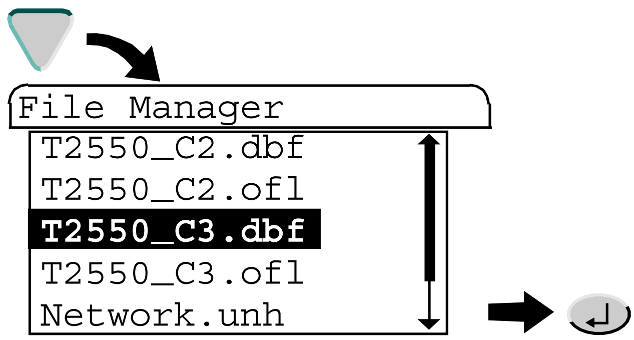
   A pick-list of all available files appears.

   *Note. Use the Filter field to show only the selected file types.*



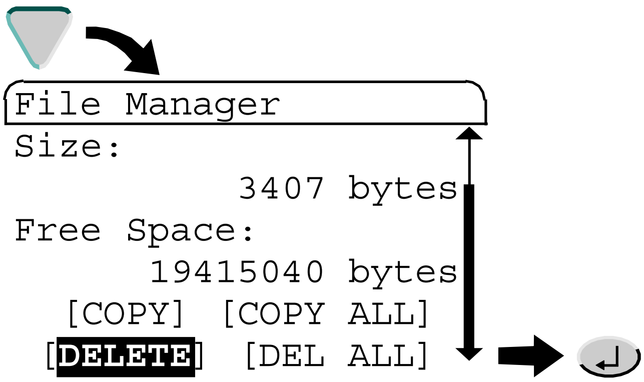

2. Again, using the Navigation keys select the required file and confirm using the Return key.

   A File Manager page appears.

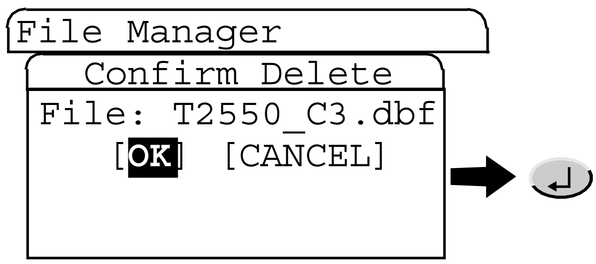



3. Finally, using the Navigation keys select the required option (COPY, COPY ALL, DELETE, or DEL ALL) and confirm using the Return key.

   A Confirm Delete page appears.




4. Using the Navigation keys select OK to accept or CANCEL to ignore the operation, and confirm using the Return key.



# CHAPTER 8 TASK ORGANISATION AND TUNING

The first section of this chapter describes these various software functions (tasks) and their scheduling within the instrument. The next section describes user tasks and their associated block servers. User Task software structure and block server operation is also outlined, as is User Task Tuning, by varying minimum repeat rates, is described.

The main topics covered are:

- Task Scheduling (section 8.1)
- User Tasks (section 8.2)
- User Task Tuning (section 8.3)
- Data Coherence (section 8.4)

## 8.1 TASK SCHEDULING

All in-built and user-programmed instructions are performed serially, i.e. one at a time.

### 8.1.1 Tasks

A Task is a unit of software that is responsible for carrying out particular duties at certain times, usually while the Database is running. There are 24 recognisable Tasks in the instrument. Most Tasks are fixed and cannot be varied by the user. Others, the user tasks, are programmable, see User Tasks.

### 8.1.2 Priorities

Each task has a priority based on its importance to efficient and safe operation. Priorities are numbered from 1 (highest) to 24 (lowest). A task, once started, will run to completion unless it is interrupted at any time by a task of higher priority. In this case the lower priority task suspends activities until the higher priority Task has finished, at which point it resumes running. These interruptions are hierarchical; several Tasks may be held in suspension by higher priority Tasks at any one time.

### 8.1.3 Functions

A list of Task functions is given in Table 8.1.3, below.

The following 6 tasks are the block servers and are under the control of the configuration engineer.

#### USER TASKS 1 TO 4

These are responsible for running up to four user tasks. User Task 1, Fast I/O task (10ms) and User Task 3, Slow I/O Task (110ms) are synchronised to the I/O modules and are module type specific, see Table 1.3.1. The associated I/O blocks can be assigned to User Task 1 or User Task 3, as applicable.

*Note Any blocks added to the database are automatically assigned to User Task 3 by default. However, the SFC_CON block and any blocks interfacing with a Sequence must always operate on User Task 4.*

#### CACHE SYNC SERVER

This Task is used to maintain synchronisation of cached blocks. The task is repeat driven every 110 msec, but this may be extended depending on the available CPU time available after servicing User Tasks.

#### CACHE CONN SERVER

This Task is responsible for processing LIN field writes into and out of cached blocks. The task is repeat driven every 110 msec, but this may be extended depending on the available CPU time available after servicing User Tasks.

### 8.1.3 TASK FUNCTIONS (Cont.)

| | Task | Schedule | Function |
|---|---|---|---|
| 1 | Tick | Every 5 msec | Provides system check. |
| 2 | Rx_ICM | Event driven | Processes messages received over the ICM. |
| 3 | Rx_LIN | Event driven | Processes messages received over the LIN. |
| 4 | ICM_Mgr | Every 50 msec | Monitors ICM link low level status.<br>Applies timeouts to transmitted messages.<br>Reprograms ICM hardware if errors are detected |
| 5 | PRMT | Event driven (<100msec) | Process Redundancy Management Task.<br>Responsible for effecting and maintaining synchronisation between redundant processors. |
| 6 | Pr_Rx | Every 100 msec (approx.) | Processes message received using ELIN via Port Resolution Protocol (PRP). |
| 7 | EDBserv (X2) | Every 10 msec (approx.) | Manages ELIN communications with external databases via cached blocks. |
| 8 | Network | Event driven | 'Housekeeping' for all transactions over the LIN. |
| 9 | File Sync | Event driven | Responsible for maintaining synchronisation of filing systems on redundant systems. |
| 10 | Mod_Rx | Event driven | Processes messages received via GW Modbus. |
| 11 | ModServ | Periodic | Modbus database management. |
| 12 | User Task (x4) | Every TaskRptn secs | Runs User Task 1 and User Task 3 synchronised to the fast and slow I/O task modules respectively. Both User Tasks run at an integer multiple (³1) of the repeat rate, i.e. User Task 1 runs at N * 10ms, and User Task 3 runs at M * 110ms, where N and M are ³1. User tasks 2 to 4 run at a repeat rate set in header block. |
| 13 | Cache Sync Server | Min. default 100 msec | Responsible for maintaining synchronisation of cached blocks. |
| 14 | Cache Conn Server | Min. default 100 msec | Responsible for connections into cached blocks (i.e. LIN network field writes) |
| 15 | LLC | Every 100 msec (approx.) | Monitors LIN link low level status.<br>Applies timeouts to transmitted messages.<br>Re-programs LIN hardware if errors are detected. |
| 16 | NFS | Event driven | Network Filing system. Processes LIN filing requests. |
| 17 | TTermcfg | Event driven | Runs the Terminal Configurator accessed via a Telnet session. |
| 18 | Pr_Maint | Every 500 msec (approx.) | PRP database management. |
| 19 | Load | Event driven | Loads a database on remote request. |
| 20 | Panel | Event driven | Runs the Human Machine Interface. |
| 21 | Config | Event driven | Runs the Terminal Configurator via the serial port |
| 22 | BatLoad | Event driven | Responsible for batch load operations (e.g. loading or unloading an SFC). |
| 23 | Bgnd (scan) | Event driven | Collates alarm information. Performs database checksum testing. |
| 24 | Idle | Event driven | 'Null task'. Provides environment for CPU execution, whilst no other tasks run. |

Table 8.1.3 Task scheduling

# 8.2 USER TASKS

## 8.2.1 Terminology

### USER TASK

A User Task is a defined set of function blocks in a database that are updated at a specific tick rate. This is normally associated with instrument control.

### BLOCK SERVER

A Block Server is a fixed software task, within this instrument, that executes a User Task, or processes cached blocks.

## 8.2.2 Execution times

User Task execution times are repeat driven, User Task 1, and User Task 3 are synchronised to the fast and slow I/O task modules respectively. Both running at an integer multiple ($^{3}1$) of the repeat rate, i.e. User Task 1 runs at N * 10ms, and User Task 3 runs at M * 110ms, where N and M are $^{3}1$.

User task 1 has the highest priority, followed (in descending order) by User Task 2, User Task 3 and User Task 4 (lowest priority).

*Note All I/O blocks for any I/O module must be configured to User Task 1 or User Task 3.*

Each of the 4 User Tasks has a ‘requested repeat rate’. This can be configured using LINtools (Task *n* Period dialog) or the Terminal Configurator (Block Full Description page).

All function blocks have a Task field, used to allocate each function block to one of the four available User Tasks. This field may also be used to configure the ‘requested repeat rate’ of the User Tasks. If the ‘requested repeat rate’ is changed via a function block allocated to a particular User Task, this change is made to the User Task, NOT the function block, and affects all other function blocks assigned to that User Task.

If using the LINtools Database Editor, selecting the Task field from the function block Object Properties Pane reveals the Task dialog. This dialog permits changes to the Task No. allocated to the function block. To enable changes to the Task Period, which is changes to the ‘requested repeat rate’, click the right (next) arrow button to display the Task Period dialog.

If the requested repeat rate is not configured (LINtools Task *n* Period dialog or Terminal Configurator Rate ms field set to 0) the default request repeat rate is applied, 10ms for User Task 1 and User Task 2, and 110ms for User Task 3 and User Task 4.

*Note Do not configure any Task to a faster requested repeat rate than any higher priority task. Any such configuration will be ignored by the instrument, but will be run according to the rules stated in* Initiating repeat rates section.

## 8.2.3 User task block servers

### BLOCK SERVER INTERACTIONS

There are six block servers in this instrument, one for each of the User Tasks, and two for the cached blocks (see Table 8.1.3). The block servers are prioritised, repeat-rate driven, and fully coherent, see Data Coherence section. The instrument's block structured LIN Database supports cached blocks by showing local 'image' of a remote function block, i.e. a function block running in another instrument on the LIN. The cached function block allows interaction with the remote function block. In a cached function block, the DBase field specifies the name of the remote LIN Database containing the 'real' function block.

Block Server 1 has the highest priority, and block server 6 the lowest. Interruption of one block server by another of higher priority, see Priorities section. The User Task block servers will only start at intervals specified by the corresponding Task repeat rate. If the task continues beyond the task repeat time, it will be suspended until the next task repeat time, e.g. User Task 1 is set to repeat every 10 ms, but lasts 10.25 ms, it will start again at the next scheduled repeat time.

*Note User Task 1 is reserved for future use. User Task 3 is synchronised with the I/O modules and will start every 110ms. Refer to Table 1.3.1.*

Figure 8.2.2a shows schematically how the block servers interact with each other according to their priorities. The darker bars represent running tasks and the paler bars represent suspended tasks.

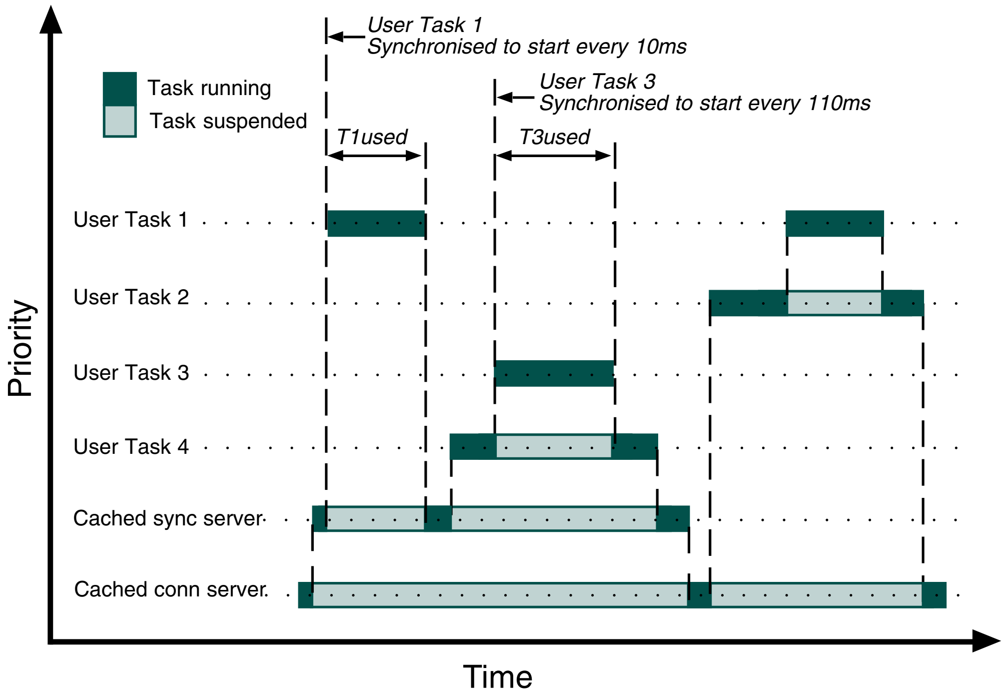



Figure 8.2.2a User task block server interactions

### 8.2.3 User task block servers (Cont.)

## USER TASK BLOCK SERVER OPERATION

A higher priority user task block server always interrupts the running of a lower priority user task block server. Thus, whenever a given user task is running, all higher priority user tasks must have run to completion.

Figure 8.2.2b shows, schematically, the sequence of events that occurs during the running of a user task block server. These are as follows:

1. The user task is marked as ‘busy’. During this ‘busy’ period lower priority tasks are suspended.
2. All connections sourced from higher priority tasks are copied into their destination blocks in this user task. This occurs as a single, indivisible, operation.
3. The blocks and their associated intra-task connections are then executed in order.
4. All connections sourced from this user task are now copied into their destination blocks in all higher priority user tasks, as a single, indivisible, operation.
5. The task ‘busy’ flag is removed.

*Note This structure results in the least work being carried out by the highest priority task.*

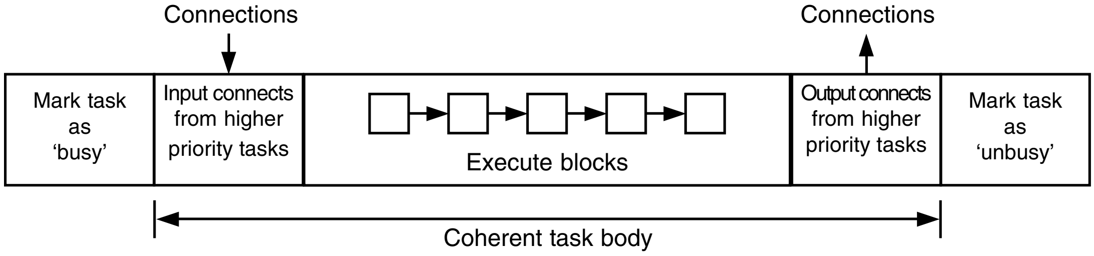



Figure 8.2.2b User task block server operation

## 8.3 USER TASK TUNING

### 8.3.1 Initiating repeat rates

At Database start-up, various checks are performed on the requested task repeat rates. Starting with the highest priority task, each block server initiates the following regulations,

- Firstly, ensuring any requested repeat rate is NOT faster than any higher priority block server task. Any lower priority block server task configured with a higher repeat rate is adjusted to match the next highest priority task.
- Then, ensuring that the requested repeat rate of the I/O synchronised block servers (User Task 1 and User Task 3 only), is an integer multiple of the I/O repeat rate value.

### 8.3.2 Automatic dynamic tuning

To compensate for the variable nature of user task execution times, the spread of CPU loading across user tasks and system tasks is monitored and user task repeat rates are altered, dynamically, to ensure a fair spread of CPU allocation.

The USERTASK block allows execution times and repeat times for all the user tasks and the cached block server to be monitored. The TACTTUNE block shows the percentage CPU usage by the various user and system tasks in the instrument. The prioritised nature of the user tasks should be allowed for when adjusting repeat rates (1 is the highest priority, 4 the lowest). The reported execution time for a user task may include a period of suspension whilst higher priority tasks execute.

Rapidly fluctuating repeat times for the lower priority tasks or a higher or unstable value of Stretch usually indicates an attempt to allocate too much total CPU time to the user tasks.

### 8.3.3 Manual tuning

The Stretch value in the USERTASK block indicates how much the dynamic tuning has had to slow the Database. During normal operation the Stretch value should show a value of ‘1’, indicating that the User Tasks are running at the requested rate, as configured in the Task Period field. Any small increases in the Stretch value, or occasional fluctuations caused by the abnormal loading of the User Tasks are acceptable. However, if the Stretch value increases significantly or is fluctuating erratically, the requested repeat rate will not be attained, causing the dynamic tuning to be inadequate. Examination of the USERTASK block and TACTTUNE block will allow the identification of the CPU with the highest usage. Simply adjust the requested repeat rate to reduce the CPU load.

*Note Some aspects of Database execution can dynamically change the CPU usage, e.g. enabling an ACTION block or starting a Sequential Function Chart (SFC). These do not run continuously but affect the CPU loading when starting and while running.*

# 8.4 DATA COHERENCE

## 8.4.1 Data flow between tasks

Data coherence is an important aspect of control strategies involving more than one user task. Data flow is defined as being coherent if during any single execution of a task the data input into it from outside the task is a ‘snapshot’ - unchanging during the execution of the task - and represents the values output from other tasks that have completed their execution.

Data coherence, by definition, refers to connections that are ‘remote’ (i.e. linking different tasks). Connections that are limited to within a task (i.e. ‘local’), are simply dealt with by being copied from source to destination immediately before executing the destination function block.

For any task, there are three important types of remote connection. These types, and the way in which data coherence is ensured, are as follows.

### CONNECTIONS INTO TASKS (FROM OTHER TASKS IN THE SAME INSTRUMENT (NODE))

In order to ensure that multiple uses (in this task) of the same value (from another task) always use the same iteration of the value, such values are copied prior to the execution of all the executable blocks of this task - i.e. a ‘snapshot’ is taken of all values external to this task.

Two types of connection apply - those from higher priority tasks to lower priority tasks, and those from lower priority tasks to higher priority tasks:

- Higher to lower priority. For coherence, whenever connections out of a task are used, all their values must result from the same iteration of that task. Owing to the priority structuring of the tasks, any connections from a higher priority task into a lower priority task meet this requirement. This is because a lower priority task cannot interrupt a higher priority task, which therefore always runs to completion. Hence, these connections are dealt with by a ‘snapshot’ copying at the start of the lower priority task.
- Lower to higher priority. A low priority task may be interrupted by a higher priority task before completion, and so be ‘caught’ with an incoherent set of output values. To avoid such invalid values being passed on, the last action of task execution is for the lower priority task to copy its set of coherent connections as a ‘snapshot’ to the higher priority task. In this way, the values passed on are always the last set of coherent values from a complete task execution.

### CONNECTIONS INTO THIS TASK (FROM OTHER TASKS IN ANOTHER INSTRUMENT)

Connections between nodes are effected by the use of cached blocks. The process of cached block transmission, and reception at the destination end, is coherent for all the data within that function block.

At the destination end, the cached block exists on a cached block server. Connections from this cached block to other blocks effectively become inter-server connections within the same node, the coherence of which is guaranteed (as described in 'Connections into tasks...', immediately above).

### 8.4.1 DATAFLOW BETWEEN TASKS (Cont.)

## CONNECTIONS OUT OF THIS TASK TO ANOTHER NODE

This type of connection results in data flow that is not coherent, because the data is transmitted across the network as individual field writes, rather than complete block updates. If coherence is required, the block(s) can be cached in the opposite direction, via an AN_CONN block for example. This is illustrated in Figure 8.4, where block A coherently connects to block B across the LIN via the AN_CONN block (bold lines), but the connection is non-coherent when routed via cached block B.

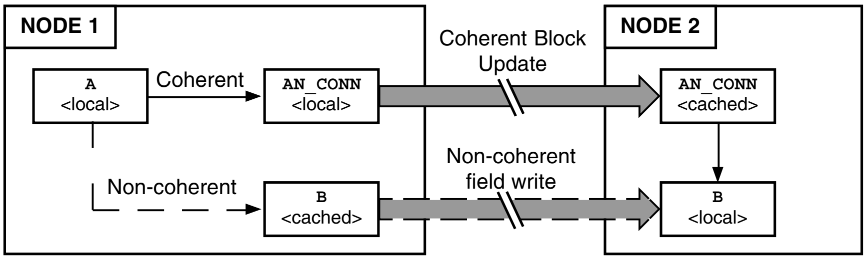



Figure 8.4 Coherent and non-coherent data flow across network

# CHAPTER 9 EVENT LOG

This section describes the Event Log facility supported by this instrument.

The purpose of the Event Log is to record and store individually time stamped, Real-Time Clock, RTC, and instrument internal time, events generated in the instrument, and provide a means of indicating the impact of an event on the system.

*Note This file is used to assist with diagnosing problems in the system.*

## 9.1 THE EVENT LOG

Each event record is stored in a ASCII text file, using a single line for each event record. An I/O Subsystem with provision for two Processors use two Event Log files, 'event_l.udz' and 'event_r.udz' for left and right processors respectively. The 'event_l.udz' file is also used in Simplex I/O Subsystem. Eventually, as more event records are automatically added, the oldest event records are removed from the file. The file indicates the impact of the event on the system using the '!' character. Status, Warning, Error, and Major Error, are represented by 0, 1, 2 or 3 '!' characters respectively.

The following example shows a typical file resulting from the power-up and start-up of a database on the primary module of a duplex pair containing two GateWay instances.

Event Impact level | Real-Time Clock time stamp | Instrument Internal time stamp | Event Record Message

```
      08/01/00 01:42:11 (0x0000028A) Power On / Reset
      08/01/00 01:42:11 (0x000002DC) PRMT initial mode PRIMARY
      08/01/00 01:42:34 (0x000014B8) Attempting Hot Start
!     08/01/00 01:42:34 (0x00001528) GW System failed to Hot Start
      08/01/00 01:42:34 (0x00001528) GW System started from GWF file
      08/01/00 01:42:34 (0x00001528) GW Load from file = TMC.GWF
      08/01/00 01:42:34 (0x0000152C) GW Load from file = TMCM.GWF
      08/01/00 01:42:35 (0x0000153C) Database Started
      08/01/00 01:42:35 (0x00001543) GW Start GWCON block = gwcon1
      08/01/00 01:42:35 (0x00001547) GW Start GWCON block = gwcon2
      08/01/00 01:42:35 (0x0000156D) PRMT block servers to PRIMARY
      08/01/00 01:42:35 (0x00001579) Changeover state machine complete
      08/01/00 01:43:52 (0x00005166) Locally generated Sync Request
      08/01/00 01:43:57 (0x000055F9) Sync Completed Successfully
!!    08/01/00 01:44:22 (0x0000695D) Secondary has inferior LIN status
      08/01/00 01:44:22 (0x0000695D) Locally generated Desync Request
      08/01/00 01:44:22 (0x0000695D) Desync Completed Successfully
      08/01/00 01:44:42 (0x000078FD) GW Reload GWCON block = gwcon2
      08/01/00 01:44:42 (0x0000790B) GW Stop GWCON block = gwcon2
      08/01/00 01:44:42 (0x00007910) GW Load from file = TMCM.GWF
      08/01/00 01:44:42 (0x00007915) GW Start GWCON block = gwcon2
```

*Note The failure to Hot Start event record is a 'Warning', but Desync event record due to the disconnected LIN cable is an 'Error'.*

Figure 9.1 Typical EventLog file, .udz, - example

### 9.1 The Event Log (Cont.)

The Event Log file, supports the recording of the following events,

- Status

Status records, no '!' characters, indicate normal operation events, e.g. power up, database start (hot start, cold start Hot/Cold start, Terminal Configurator, Network), database stop, Online Reconfiguration operations, normal synchronisation of a duplex pair, etc.

- Warning

Warning records, one '!' character, indicate minor abnormalities, e.g. hot start fails due to power off time exceeded, controlled changeover of a duplex pair, etc.

- Error

Error records, two '!' characters, indicate real faults on the system, e.g. automated changeover of a duplex pair due to a detected fault, running serial communications on unsupported versions of this instrument causing corruption of communications bus on power-up.  If any Error is written to the Event Log file, the *Alarms.EventLog* and *Status.EventLog* fields of the database Tactician header block are setTRUE.  This offers an output that can be linked a display to provide immediate identification of a problem that can have an effect on the system.

- Major Error

Major Error records, three '!' characters, indicate real faults in the execution of the instrument that must be investigated before continuing.  If any Major Error is written to the Event Log file, the *Alarms.EventLog* and *Status.EventLog* fields of the database header block are setTRUE.  This offers an output that can be linked a display to provide immediate identification of a problem that can have a serious effect on the system.

# CHAPTER 10 DATA MANAGEMENT

This section describes the Data Management functionality supported by the Control version of this instrument only. Data Management functionality will only operate on hardware status level 4 or later, e.g. Hardware J4, Software Version 4, and a valid Licence, D10 to D90.

***IMPORTANT*** ***Hardware status level 4 and later contains flash memory that supports Data Management, but this will only be functional if the relevant licence is present. Use the Tactician Licence Tool to request a licence upgrade.***

The purpose of Data Management is to record, archive and visualise data values derived from a strategy during run-time. The data values are recorded to a file stored in the flash memory of the instrument, and can then be archived to a maximum of three FTP (File Transfer Protocol) Servers.

The main topics covered are as follows:

- Data Recording (section 10.1)
- Data Archiving (section 10.2)
- Data Management Configuration (section 10.3)

*Note Refer to the T2550 PAC Tutorial/User Guide, Part no. HA 029 723, for full details about Data Management configuration.*

## 10.1 DATA RECORDING

Data recording is the process of writing data values derived from selected parameters in the instrument strategy to a Data Recording file, .uhh, see Data Recording File, .uhh. To simplify the organisation of the recorded fields, they are configured in to groups, see Data Recording Groups, and held in the internal flash memory of the instrument. The instrument can be configured to automatically push the .uhh files via the network to a defined FTP Server for archiving, see Data Archiving.

Data recording is configured using LINtools, see *LINtools help file (Part no. RM 263 001 U055)*, and downloaded to the instrument with the database file, .dbf.

*Note Instrument flash memory problems can be resolved by inspecting the RMEMDIAG block , see LIN Block Reference Manual, Part no. HA 082 375 U003.*

### 10.1.1 Data Recording File, .uhh

The Data Recording file, .uhh, is an electronic tamper-resistant file that is used to record the values derived from the instrument during run-time. It is a non human-readable file format, that can only be interpreted by **Review** software.

### 10.1.2 Data Recording Groups

A Data Recording Group is a set of LIN block fields that are recorded to one sequence of files providing a method of organising recorded data, e.g. a single group can be created for each area of the plant/system. This provides the ability to group fields to best suit the process requirements. Each field is assigned to a group, identified by an RGROUP block. Each group records the configured field value at a specified rate. Fields may be assigned to multiple groups simultaneously, allowing the defined field to be recorded at different rates.

It is possible to record up to eight groups simultaneously, i.e. one RGROUP block per recording group, with a maximum of 127 data values per group.

## 10.2 DATA ARCHIVING

Data archiving applies to the process of copying recorded data from the instruments' internal flash memory to .uhh files on a defined FTP Server across a network via FTP, see File Transfer Protocol (FTP). The archived .uhh file can then be replayed using an off-line tool, **Review** software.

A maximum of three FTP servers can be defined in the **Instrument Options** page in the **Instrument Properties** dialog to provide a back-up service for archiving the .uhh files. When multiple FTP Servers are configured the .uhh files are archived to all defined FTP Servers.

*Note Archiving problems can be resolved by inspecting the RARCDIAG block, see LIN Block Reference Manual, Part no. HA 082 375 U003.*

### 10.2.1 File Transfer Protocol (FTP)

File Transfer Protocol (FTP) is a commonly used Server/Client transfer mechanism. It allows the instrument to act as a FTP client to up to three FTP Servers for the purpose of transferring recorded files from the flash memory to a remote computer.

## 10.3 DATA MANAGEMENT CONFIGURATION

Data Management is configured using LINtools. Groups of recorded fields are defined in the instrument database, and can be individually customised using the Data Recording Configurator. Configuring individual fields provides a clear identification of each recorded field when displayed in **Review**. LINtools also provides the facility to define the FTP Servers used to archive the .uhh files, via the Instrument Properties dialog. Once the files are archived to the defined FTP Servers, **Review** can be configured to display .uhh files from the different groups and instruments.

To configure data management,

1 Define the data recording configuration using LINtools.

2 Define the data archiving configuration using the Instrument Properties in LINtools.

3 Define the data visualisation configuration using Review.

*Note Review can pull files directly from the instrument. It is not recommended but can be configured using the **Auto-Backup + Transfer** dialog in Review, and requires a **User Name**, 'history', and a **Password**, 'history'.*

4 Configure the FTP Servers.

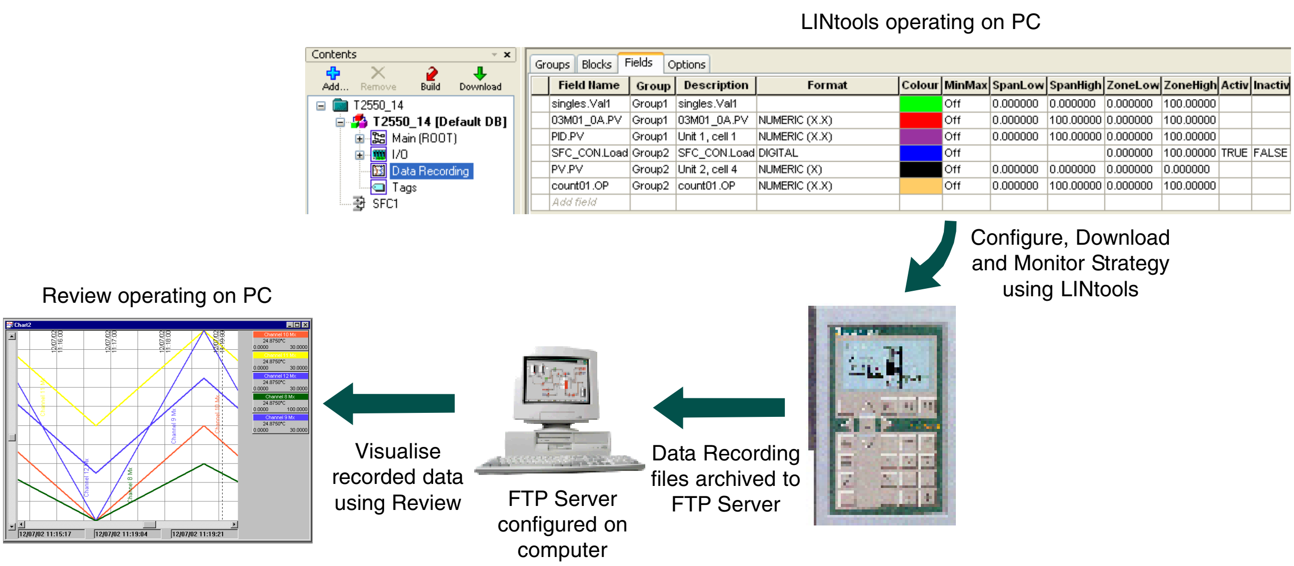



Figure 10.3 Data Management Configuration

# CHAPTER 11 SETPOINT PROGRAMMER

This section describes the Setpoint Programmer facility supported by this instrument.

The purpose of a Setpoint Programmer is to create a program that will control and manage the changing target values that an automatic control system, e.g. PID controller, will aim to reach.

Example

A boiler control system might have a temperature Setpoint, that is a temperature the control system aims to attain in the system.

The main topics covered are as follows:

- Setpoint Programming (section 11.1)
- Program Configuration (section 11.2)

*Note Refer to the T2550 PAC Tutorial/User Guide, Part no. HA 029 723, for full details about Setpoint Programming configuration.*

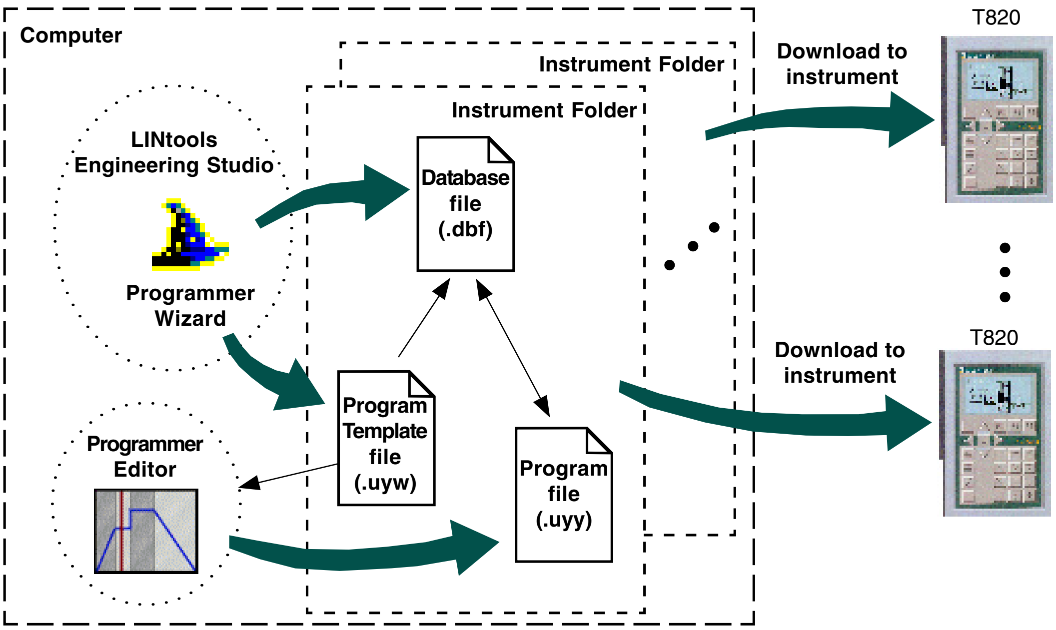
## 11.1 SETPOINT PROGRAMMING

The Program Template file (.uyw) decribes the structure of a Programmer, the number of Channels and corresponding names, the number of Digital Events, Wait conditions, Exit Conditions, and User Values, and how each Channel is presented in the Programmer Editor. The description contained in the Program Template file is then used to ensure the Program file (.uyy) corresponds to each Programmer's structure. The names of individual process variables (Profiled Channels), digital event outputs and user values defined in the Program Template file are all shown in the Programmer Editor used to generate the Program.

The Program Template file must be modified using the Programmer Wizard in LINtools.

The Program Template file can be referenced by a local instrument or any other instrument on the same network. This allows the same Program Template file to apply to multiple instruments.

The following diagram shows all system configuration components required for a Programmer application.



### 11.1 SETPOINT PROGRAMMING (Cont.)

The Setpoint Program is a set a values stored in a Program file that is used to control a specified process variable over a defined period of time. The configured Program values produce a pattern of control for a single wired process variable value (Profiled Channel) typically derived from an AI_UIO block connected to the plant/system. The output or current setpoint (*PROGCHAN.Monitor.CurrSP*) of the channel is the demand value, and should be wired to the setpoint of a control loop, i.e. *LOOP_PID.SP.AltSP*, together with the loop PV itself, so the loop can control an output, typically via an AO_UIO block, to achieve the desired process value. The Program file is generated by the Programmer Editor within the constraints of a Program Template file generated using the Programmer Wizard.

The Programmer Wizard is launched from LINtools Engineering Studio and simplifies the generating or editing of a Program Template file. It also automatically creates a PROG_WIZ compound in the Database file (.dbf). This compound contains

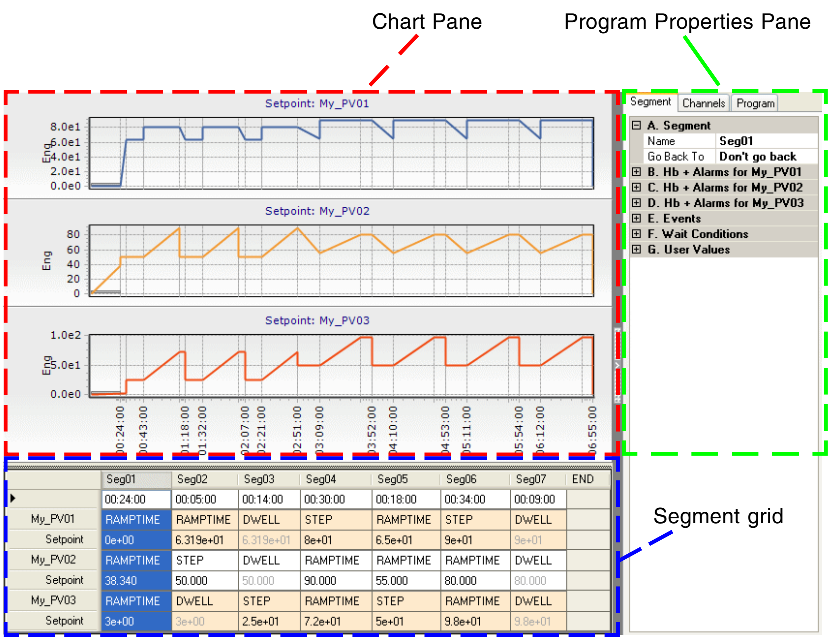
- 1 PROGCTRL block, used to control the overall execution of the Setpoint Program
- up to 8 PROGCHAN blocks, one for each profiled setpoint per PROG_WIZ compound,
- up to 8 SEGMENT blocks per channel maximum, each SEGMENT block offering 4 program segments

While using the Programmer Wizard to configure the Program Template file, the individual process variables (Profiled Channels) will be titled for identification in the Programmer Editor. The wizard can also be used to specify the maximum number of digital event outputs, user values and segments allowed in the Program. The total number of digital event outputs, user values and Wait/Exit conditions is only limited by the size of the Database file and the remaining number of PROGCHAN blocks available. Additional PROGCHAN blocks will be automatically created if more than 16 digital event outputs and 4 user values are requested, but only up to 8 PROGCHAN blocks can exist in a PROG_WIZ compound.

## 11.1.1 Programs

This instrument supports single and multi-channel Programs. This is defined by the number of Name entries in the Profiled Channels page of the Programmer Wizard. The Chart pane in the Programmer Editor shows a maximum of three Profiled Channels, the first two Name entries in the Programmer Wizard correspond to the two most upper Profiled Channels the other Profiled Channel displayed in the lowest Chart position is the one selected in the Segment grid. The lowest Chart position can also show a Digital Event Output, or User Values by selecting it from the Properties pane, see below.

- A single channel Program, i.e. the control of one input value from the plant/system, supports Step, Dwell, RampRate, RampTime, and End Segment types. The Profiled Channel appears in the upper most Chart position, allowing Digital Event Output, and User Values to be displayed in the remaining Chart positions.
- A multi-channel Program, i.e. the control of more than one input value from the plant/system over an identical time period, supports Step, Dwell, RampTime, and End Segment types, but does not support RampRate Segment type. The first two Profiled Channels always appear in the upper most Chart positions in the order defined in the Programmer Wizard, and in the lowest Chart position displays the selected information as stated above.



## 11.2 PROGRAM CONFIGURATION

A Setpoint Program is configured using LINtools and the Programmer Editor. LINtools provides the Programmer Wizard to generate and/or edit a Program Template file and create a PROG_WIZ compound containing the required PROGCTRL block, PROGCHAN blocks and SEGMENT blocks. The Programmer Editor is used to configure a Program, the pattern of control for each profiled setpoint, Digital Event Outputs, User Values, Wait conditions and Exit conditions. Any Program Template file can be used to construct many different Programs that can be run by each Programmer instance.

The PROGCTRL block is an interface between the Programmer Editor and the Database file. It provides control and management of the Program. The number of PROGCHAN blocks is equal to the number of Profiled Channels plus sufficient blocks to support the requested number of Digital Event Outputs and User Values. Any PROGCHAN blocks that have been automatically created simply to add further Digital Event Outputs or User Values have their Profiled Channels disabled (*PROGCHAN.Config.Options.DisChan* set TRUE). Each SEGMENT block supports 4 segments. It uses pages to distinguish between segments and each page shows a segment configured in the Programmer Editor.

To configure a Setpoint Program,

1 In LINtools, create (edit) the instrument Program Template file using the Programmer Wizard on the Tools menu.

***IMPORTANT To prevent erroneous Program Template file configuration always use the wizard to edit the blocks in the PROG_WIZ compound. Changing the number of Profiled Channels, Digital Events, or User Values will invalidate any Program file created with the previous version.***

2 Wire the control loop configuration (LOOP_PID block) to the Programmer configuration (PROGCHAN block) and return the current setpoint from the Programmer configuration (PROGCHAN block) to the control loop configuration configuration (LOOP_PID block). This will provide the setpoint control for the control loop configuration.Wire the input values (*AI_UIO.PV*) from the plant/system to the control loop (*LOOP_PID.Main.PV*).

- Wire the Digital Events and User Values defined using the Programmer Wizard to appropriate output blocks.
- Wire to the required Wait conditions and Exit conditions defined using the Programmer Wizard from appropriate input blocks.

When wiring is complete, save the Database file. Add the Program Template file and the Program file to the List of files to be Downloaded.

*Note Refer to the T2550 PAC Tutorial/User Guide, Part no. HA 029 723, for full details about setpoint control wiring.*

3 Create and/or open a Program file. This can be done by using the context menu available when selecting the *PROGCTRL.File.ProgFile* (*block.page.field*) in the Object Properties pane in LINtools after providing the Program name, or by opening the Programmer Editor , and selecting File > New (Open), and choose the Program Template file that matches the blocks of a PROG_WIZ compound in the database.

4 Configure the Program, setting each Segment type, Duration, and Target Setpoint in the Segment grid as required. Then configure the Digital Event Outputs, User Values, Exit and/or Wait conditions in the Program Properties Pane.

*Note Refer to LINtools and Programmer Editor for full details.*

5 Download all relevant files to the instrument.

*Note Connect to the instrument from the Programmer Editor to control the running Program.*

## 11.2 PROGRAM CONFIGURATION (Cont.)

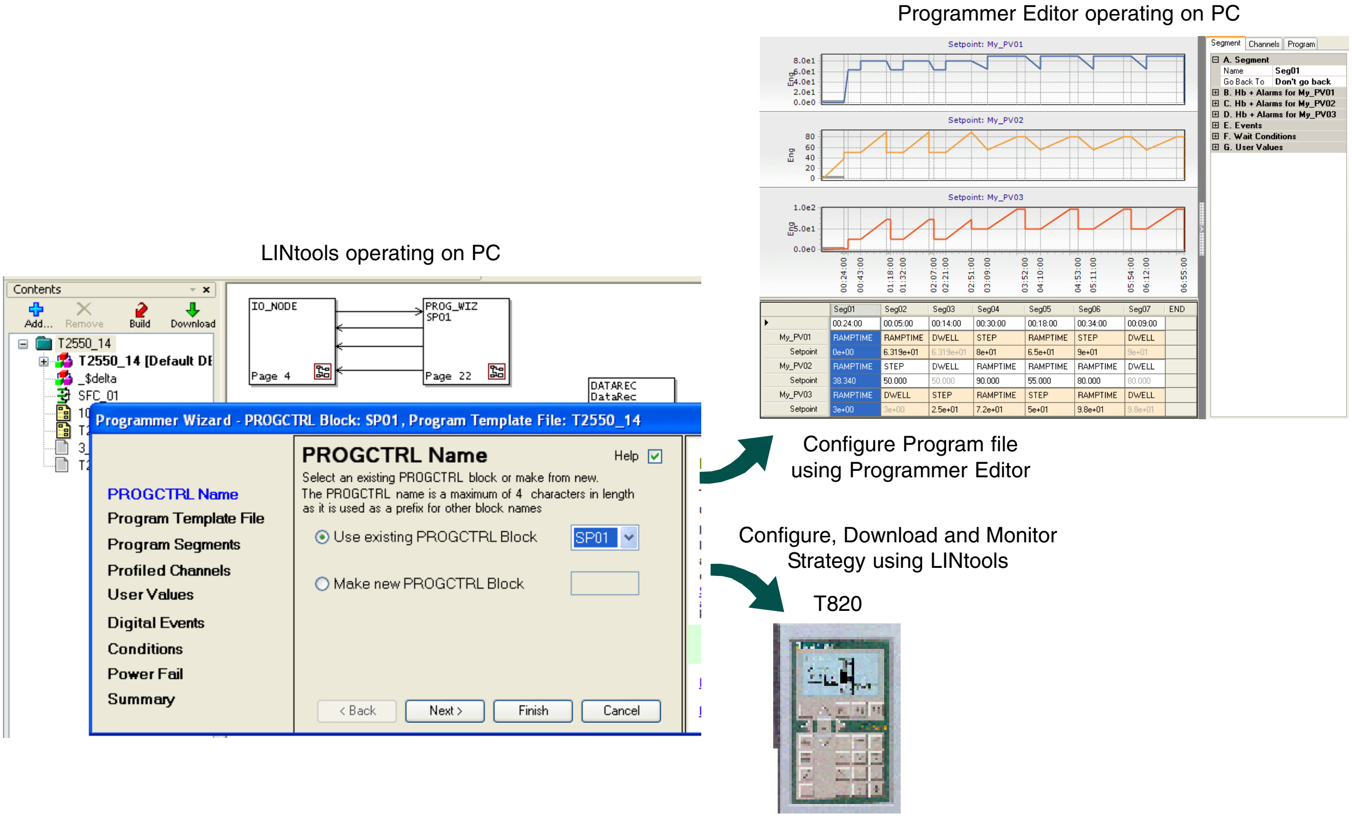



Figure 11.3 Setpoint Program Configuration

# CHAPTER 12 SERVICE

This section describes the regular service procedures, such as changing of back-up batteries etc., and shows how to replace the Compact Flash Memory Card.

The main topics covered are:

- Preventive Maintenance Schedule (section 12.1)
- Replacement Procedures (section 12.2), including Compact Flash Memory Card, and Battery Replacement.

For details of how to update and change the instrument'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.

## 12.1 PREVENTIVE MAINTENANCE SCHEDULE

The following periods are recommended to guarantee maximum availability of the instrument, 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.

The following are recommended:

1. Every one to three years, the service consumables listed below should be replaced. The recommended replacement period is a function of the average ambient temperature in which the instrument operates. At an ambient of 50 degrees Celsius, the recommended replacement period is two years. For an ambient of 20 degrees Celsius the recommended period is four years.

   Service consumables are:

   a) Battery - Part no. PA261095.

Whenever preventive maintenance is performed, it is recommended that a visual inspection of the instrument 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.

## 12.2 REPLACEMENT PROCEDURES

### 12.2.1 Firmware upgrade

The manufacturer can supply replacement Compact Flash Memory Card pre-programmed with the latest firmware version. This allows the user to upgrade the unit just by replacing the card. In such cases, the user is responsible for reloading configuration files.

- Alternatively, the manufacturer's agents can upgrade the firmware version with the card *in situ* thus retaining the user configuration.

#### COMPACT FLASH CARD REPLACMENT PROCEDURE

Figure 11.2.1 shows the replacement of the 'Compact Flash Memory Card' fitted to current T820s. This procedure allows LIN Databases, user configurations, IP address and Network name to be transferred from one module to another, allowing the 'Mean Time to Replace' to be reduced to a minimum.

**Caution**

Always ensure the power is isolated prior to removing the unit and before configuring the hardware.

*Note. It is recommended that the unit is removed from the DIN Rail or panel for ease of use.*

1. Disconnect all connections that restrict access to the unit.
2. Remove the rear cover, by extracting the 4 screws, and carefully guide the rear cover away from the T820.
3. Once the cover has been safely removed, carefully withdraw the Compact Flash Memory Card from its connector.
4. Slide the replacement Compact Flash Memory Card until secured firmly in position.
5. Re-assemble the T820.

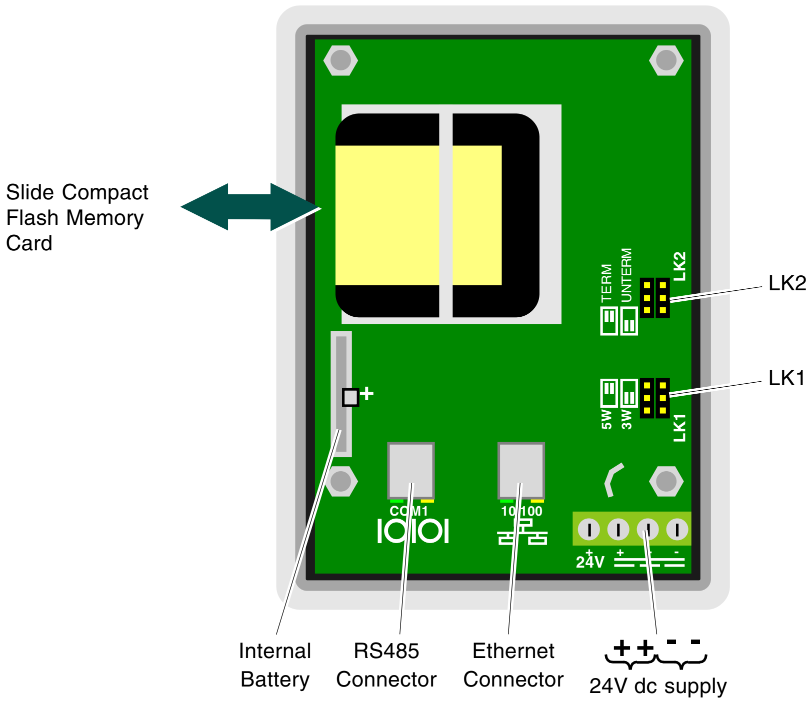



Figure 12.2.1 Replacing the Compact Flash Memory Card

### 12.2.2 Battery replacement

WARNING

Care must be taken to ensure that neither the exhausted battery or its replacement are shorted out, otherwise an explosion may occur, resulting in the emission of hazardous products. Do not use a metal object to remove the battery from its holder: if necessary, use a plastic or wooden item instead.

WARNING

Before the Polycarbonmonofluoride Lithium battery can be disposed as a non-hazardous product, ensure it is fully or mostly discharged.

*Note. Although the circuitry and the battery are protected against incorrect battery insertion, the clock, hot start data etc. held in the T820 SRAM will not be protected against power loss, should the battery be inserted 'back-to-front'.*

**Caution**

Always ensure the power is isolated prior to removing the unit and before configuring the hardware.

*Note. It is recommended that the unit is removed from the DIN Rail or panel for ease of use.*

1. Disconnect all connections that may restrict access to the rear cover.
2. Remove the rear cover, by extracting the 4 screws, and carefully guide the rear cover away from the T820.
3. Once the cover has been safely removed, the exhausted battery, can be extracted from its holder, and the replacement fitted.
4. Ensure that positive side of the battery is towards the raised arm. (There is usually a + sign on the battery sleeve.)
5. Re-assemble the T820.

*Intentionally left blank*

# APPENDIX A SPECIFICATIONS

## A.1 INSTALLATION CATEGORY AND POLLUTION DEGREE

This product has been designed to conform to BS EN61010 installation category II and pollution degree 2. These are defined as follows:

### A.1.1 Installation category II

The rated impulse voltage is 2500V.

### A.1.2 Pollution degree 2

Normally, only non-conductive pollution occurs. Occasionally however, a temporary conductivity caused by condensation shall be expected.

# A.2 SPECIFICATION

This specification defines the T820 instrument components:

- General specification (see A.2.1)
- Hardware specification (see A.2.2)
- Software specification (see A.2.3)

## A.2.1 General specification

**Physical**

| | | |
|---|---|---|
| Dimensions: | | |
| | T820 HMI: | 104 mm wide x 144 mm high |
| | Cut out: | 98 mm wide x 138 mm high (+1mm) |
| Weight: | | |
| | T820 HMI: | 1kg max |
| Safety earth connections: | | |
| | T820 HMI: | An M4 earth stud on rear cover. |
| Screen: | | |
| | Type: | Monochrome 128x64 pixel LCD |

**Environmental**

| | | |
|---|---|---|
| Temperature: | | |
| | Storage: | -10 to +85°C |
| | Operation: | 0 to + 50°C |
| Humidity: | | |
| | Storage: | 5 to 95% RH (non-condensing) |
| | Operation: | 5 to 85% RH (non-condensing) |
| RFI: | | |
| | EMC emissions: | BS EN68136 |
| | Susceptibility: | BS EN68136 |
| Safety Specification: | | BS EN61010-1:2001 |
| | | Installation Category II, Pollution degree 2 |
| Vibration: | | BS EN60873, section 2.1.3 |
| Shock: | | BS EN60068-2-31; BS EN60873, section 2.1.3 |
| Protection: | | |
| | Front panel: | IP66 |
| | Rear panel: | IP20 |

**Power Requirements**

| | | |
|---|---|---|
| Main supply: | | |
| | Voltage Range: | 18 to 28 Vdc. |
| | Supply Ripple: | 2Vp–p max |
| | Consumption: | 3.5W max |
| Surge Current: | | 200mA max. |
| Backup supplies | | |
| | Internal Battery: | Polycarbonmonoflouride Lithium battery. Maintains the Real-Time Clock for 1.5 years discontinuous use. |

Warning

Before the Polycarbonmonofluoride Lithium battery can be disposed as a non-hazardous product, ensure it is fully or mostly discharged.

## A.2.2 Hardware specification

### General

Compact Flash Card:

| | |
|---|---|
| Memory Card: | 32 MByte removable |

### Communications port(s)

Ethernet Port (Controller Unit Only):

| | |
|---|---|
| Connectors: | RJ45 |
| Network medium: | Ethernet Category 5 cables |
| Protocols: | LIN over Ethernet / IP (ELIN), Modbus-TCP RTU slave, FTP. |
| Speed: | 10/100Hz |
| Network Topology: | Star connection to a hub |
| Line length (max): | 100 metres, extendable by repeater |
| Allocation of IP address: | Manual, DHCP, Link-Local or BootP |
| Isolation: | 50V dc. |

Serial Port:

| | |
|---|---|
| Connector: | RJ45 |
| Network medium: | 3-wire or 5-wire RS422/RS485 communications |
| Protocols: | Modbus (Controller Unit Only)<br>Proprietary (Display Unit Only) |
| Speed: | 38k4 max. |
| Network Topology: | Point to Point or Multidrop |
| Line length (max): | 100 metres |
| Isolation: | 50V dc. |

## A.2.3 Software specification

### LIN Block libraries (continuous LIN Database Function Block categories)

Function Block categories:

| | |
|---|---|
| Batch: | Sequencing recipe/record and discrepancy checking |
| Communication: | Control and/or interface to the communications subsystem |
| Condition: | Dynamic signal-processing and alarm collection |
| Configuration: | Instrument identity blocks |
| Control: | Analogue control, simulation and communications |
| Diagnostic: | Diagnostics |
| I/O: | Analogue and digital input output |
| Logic: | Boolean, latching, counting and comparison |
| Maths: | Mathematical functions and free-format expressions |
| Organisation: | Display navigation and flash memory data recording |
| Programmer: | Setpoint Programmer control |
| Record: | Recording of analogue and digital points |
| Selector: | Selection, switching, alarm and display page management |
| Timing: | Timing, sequencing, totalisation and events |

## A.2.3 Software specification (Cont.)

### LIN Blocks Software Licence categories

*Note The Tactician Configuration (Header) block supports T2550 and T820 product configuration, but this instrument hardware does not support Profibus communications.*

| Block Type | Category: Foundation | Standard | Control | Advanced | Description |
|---|---|---|---|---|---|
| **Batch** | | | | | Batch control and management |
| RECORD, DISCREP, SFC_DISP, SFC_MON | | ✓ | | | |
| SFC_CON | | | ✓ | | |
| **Communications** | | | | | Communications control |
| GW_CON, GW_TBL, GWPROFS_CON | ✓ | | | | |
| **Conditioning** | | | | | Signal processing control |
| AN_ALARM, DIGALARM CHAR, UCHAR, FILECHAR | ✓ | | | | |
| FIITER, LEAD_LAG, LEADLAG, FLOWCOMP, INVERT, RANGE, TC_SEL | | ✓ | | | |
| TC_LIFE | | | ✓ | | |
| AGA8DATA, GASCONC, ZIRCONIA | | | | ✓ | |
| **Control** | | | | | Loop control and management |
| AN_CONN, DG_CONN | ✓ | | | | |
| ANMS, DGMS, MAN_STAT, MODE, SETPOINT, PID_LINK, SIM | | ✓ | | | |
| 3_TERM, LOOP_PID, PID, TUNE_SET | | | ✓ | | |
| **Convert** | | | | | Conversion control and management |
| REALTIME | | | ✓ | | |
| **Diagnostic** | | | | | Fault control and management |
| All blocks | ✓ | | | | |
| **I/O** | | | | | I/O control and management |
| AI_UIO, AO_UIO, DI_UIO, DO_UIO, FI_UIO, MOD_UIO, TPO_UIO MOD_DI_UIO, MOD_DO_UIO CALIB_UIO | ✓ | | | | |
| **Logic** | | | | | Logical calculation control |
| AND4, OR4, XOR4, NOT COMPARE, COUNT LATCH, PULSE | | ✓ | | | |
| **Maths** | | | | | Mathematical calculation control |
| ACT_2A23WT, ACT15A3W, ACTION, DIGACT, ACTUI818 | | | ✓ | | |
| ADD2, SUB2, MUL2, DIV2, EXPR | | ✓ | | | |
| **Organise** | | | | | Screen and Data management |
| AREA, GROUP | ✓ | | | | |
| **Programmer** | | | | | Setpoint control and management |
| PROGCTRL, SEGMENT | ✓ | | | | |
| SPP_RAMP, PROGCHAN | | ✓ | | | |
| **Recorder** | | | | | Data recording control and management |
| RGROUP | ✓ | | | | |
| DR_ANCHP, DR_DGCHP | | ✓ | | | |
| **Selector** | | | | | Signal selection control and management |
| ALC | ✓ | | | | |
| 2OF3VOTE, SELECT, SWITCH | | ✓ | | | |
| **Timing** | | | | | Time control and management |
| SEQ, TIMER, TIMEDATE, TPO | ✓ | | | | |
| DELAY, DTIME, RATE_ALM, RATE_LMT, SEQE, TOT_CON, TOTAL, TOTAL2 | | ✓ | | | |
| **Control Module** | | | | | Hardware represented block |
| VLV1IN, VLV2IN, VLV3WAY, MTR3IN, DUTYSTBY, AN_ALM_2 | ✓ | | | | |

### A.2.3 Software specification (Cont.)

**Continuous database resources**

| | |
|---|---|
| Number of function blocks (maximum) | 256 |
| Number of templates (maximum) | 50 |
| Number of libraries (maximum) | 28 |
| Number of EDBs (maximum) | 32 |
| Number of FEATTs (maximum) | 256 |
| Number of TEATTs (maximum) | 128 |
| Number of Servers (maximum) | 6 |
| Number of connections | 256 |
| Control database size (maximum) | 85 kByte |

*Notes*

1. *Apart from database memory sizes, these figures 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 or allow Online Reconfiguration. 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. *If the EDB maximum is exceeded, some EDBs will malfunction. This is likely to affect the LINtools facility.*

**Sequence Control Resources**

Sequence memory:

| | |
|---|---|
| Program data: | 59 kBytes |
| SFC Resources: | 53 kBytes |
| N° of independent sequence tasks: | 68 simultaneously active |
| SFC Roots: | 15 |
| Steps: | 212 |
| Action associations: | 848 |
| Actions: | 424 |
| Transitions: | 318 |

**Alarms and events**

| | |
|---|---|
| Number of records in history: | 512 lines maximum |
| History line format: | Name - Type - Date - Time |

**Modbus**

Modbus communications support:

| | |
|---|---|
| Configuration Tools: | The Serial parameters of the instrument must be configured using the PC based Modbus Tools software. The instrument parameters can be configured using the PC based Instrument Properties dialog |
| Memory Size: | 14 kBytes |
| Maximum Tables: | 80 Diagnostics Registers = 16 general purpose registers + 1 register for each table |
| Operating Mode: | Master, Slave |
| Transparent Modbus Access (TMA/TalkThru): | Yes |
| Format: | Direct 32 bit, Reverse 32 bit (D, and S) |
| Tick Rate: | 5ms |
| Number of facilities: | 3 Modbus Gateway facilities |
| Redundancy: | Full control |
| Interface: | 2 (1 Serial (COM1) + 1 TCP/IP (TCP)) The Serial interface is electrically limited to communicate with a maximum of 64 slave devices, 1 per register in the .gwf. TCP can communicate with 16 slave devices and 16 additional master devices, via the ENET3 and ENET4 ports. |

*Intentionally left blank*

# APPENDIX B CUSTOMISING

This section is for users who wish to customise the Built-in pages to their own requirements.

It consists of the following sections:

- Introduction (section B.1)
- Dictionaries (section B.2)
- Built-in page identification (section B.3)
- Database names (section B.4)
- Database monitoring (section B.5)
- Character set reference (section B.6)

## B.1 INTRODUCTION

The most important form of customisation is the creation of User Pages, i.e. display pages created for the user's specific application. These are created using the User Screen Editor, see *User Screen Editor Online Help* (Part no. RM260749U005). Text created on these pages can be specified directly, or via user page dictionaries, see Dictionaries section, to allow internationalisation.

When constructing user pages, it is possible to define buttons jumping to built-in pages, for example a button jumping directly to the Alarm History page. A list of the built-in page identification numbers for the Unit is supplied, see Built-in page identification section.

Certain built-in pages refer to items in the LIN Database, containing names that may be obscure to the user, e.g. block names and alarm types appearing in the Alarm History page. A suitable alternative, or 'Alias', can be created for block names and alarm types, see Database names section.

In some applications, a high importance may be attached to changes of specific parameters in the LIN Database. Such parameter changes can be configured to appear in the Alarm History page, see Database monitoring section.

Finally, an ASCII-based character set, see Character set reference section, is supported on the Unit.

## B.2 DICTIONARIES

Dictionaries can be used to hold text appearing on user pages. This can be either a single dictionary, *_user.uyl* , or, for an internationalised system, multiple dictionaries, *_user<n>.uyl* , where *<n>* is 0 for English, 1 for French, etc. These files should be set up in the 'E:' drive of the Unit.

A dictionary is a simple text file, containing a maximum of 200 entries. Each entry contains a maximum of 32 characters in the form,

```
U<n>,<Text>
```

where <n> is the dictionary index referred to on the User Page, and <Text> is the required text, for example,

```
U1,Boiler Overview
U2,Main Temp
U3,Pressure
etc, etc.
```

*Note Dictionaries can be configured to modify text in the built-in pages, but this exceeds the scope of this manual.*

## B.3 BUILT-IN PAGE INDENTIFICATION

These are the identification numbers that can be associated with user page buttons, used to create a jump to one of the built-in pages.

*Note Where available, the Page Name column includes the name that appears within the built-in pages, shown in bold type.*

| Page Name | Page Id number | Description |
|---|---|---|
| User | 1-999 | User-defined pages, created using the User Screen Editor |
| Root Menu | 1000 | The top level menu |
| **OVERVIEW** | 1500 | Area/Group/Point hierarchical pages |
| **SYSTEM** (submenu) | 2000 | Second level menu for System pages |
| **SUMMARY** | 2010, 2011 | System summary (Main page, Options Page) |
| **APP SUM** | 2020 | Application Summary page |
| **STARTUP** | 2030 | Start-up strategy setup (hot/cold start etc.) |
| **COMMS** | 2040 | Communications setup |
| **CLOCK** | 2050 | Real-time clock setup |
| **INTERNAT** | 2060 | Internationalisation setup (language selection etc.) |
| **PANEL** | 2070 | Panel setup (LCD display contrast, timeouts, etc.) |
| **APPLN** (submenu) | 2200 | Third level menu for application pages |
| **APP MGR** | 2210 | Application Manager page |
| **FB MGR** | 2230 | Function Block Manager pages |
| **SETUP** (submenu) | 2300 | Third level menu for setup pages |
| **ACCESS** | 4000 | Security Access (logging on, etc.) |
| **ALARMS** (submenu) | 9000 | Second level menu for Alarms pages |
| **HISTORY** | 9010 | Alarm History page |
| **SUMMARY** | 9011 | Alarm Summary page |

Table B.3 Built-in Page Identification Numbers

## B.4 DATABASE NAMES

Each name used in the LIN Database can be assigned an `Alias`, and is contained in an application-specific 'database names file', *<appname>.uyn*. An `Alias` is a user defined text string, that replaces the Database Name on the screen. The *<appname>.uyn* files need to be set up on the 'E:' drive of the Unit.

A database names file is a simple text file, with entries for the following types of database names.

- Function block names
- Alarm names
- Enumerations

### B.4.1 Function block names

Function block names are displayed, for example in the Alarm History page, where they are stored as ASCII.

The syntax for replacing an existing function block name is,

`<Block Alias>,<Block Name>`

where

'`Block Alias`' is the replacement text and '`Block Name`' is the original LIN Database name for the function block.

For example:

`First Loop,Lp1` replaces the LIN Database name '`Lp1`' with the new name '`First Loop`'.

*Note. A function block Alias can be up to 16 characters allowing for a more descriptive Block name.*

### B.4.2 Alarm names

These are displayed in the Alarm History page.

The syntax for replacing an existing alarm name is,

`<Alarm Alias>,<Block Name>.Alarms.<Alarm Name>`

where:

'`Alarm Alias`' is the replacement text and '`Alarm Name`' is the original LIN Database name for the block.

For example:

`No Battery,T820.Alarms.BadBat` replaces the current name (text) '`BadBat`' with the new name (text) '`No Battery`'.

### B.4.3 Enumerations

The most commonly used Enumerations are Boolean two-state variables, such as TRUE/FALSE and OPEN/CLOSED. By using the User text dictionary a specified Enumerated variable can be identified using a more descriptive text string.

The syntax for replacing an existing enumeration in a block field is,

```
,<Block Name>.<Block Field>, "<Alias>,<Alias>"
```

For example:

```
,digital.Out,"OPEN,CLOSED"
```

This replaces the existing enumerations in block.field ‘`digital.out`’ with the new enumerations `OPEN,CLOSED`.

The syntax for replacing an existing enumeration in a block field subfield is,

```
,<Block Name>.<Block Field>.<Block SubField>, "<Alias>,<Alias>"
```

For example:

```
,digital.Out.Bit1,"OPEN,CLOSED"
```

This replaces the existing enumerations in block.field.subfield ‘`digital.out.bit1`’ with the new enumerations `OPEN,CLOSED`.

## B.5 DATABASE MONITORING

Each parameter used in the LIN Database can be monitored by identifying the parameter in an application-specific 'database monitoring file', <*appname*>*.uya*. This allows changes to selected parameters to be monitored, providing a certain degree of 'audit-trail' functionality. The <*appname*>*.uya* files need to be set up on the 'E:' drive of the Unit.

The file consists of a special header line, followed by a single line for each parameter to be monitored.

*Note This must not be considered as 21CFR Part11 compliant.*

### B.5.1 Header Line

The syntax for the header line is of the form,

```
UYA,1[,[<burst_threshold>][,[<back_off_period>][,[<dynamic_threshold]]]]
```

The three optional numeric fields are as follows,

| | |
|---|---|
| Burst threshold | The number of consecutive LIN Database cycles that a value has changed before a 'burst' condition is seen to have occurred. Once the burst condition occurs audit trailing will stop to prevent over-filling the log until the value stabilises gain. Default value = 10 |
| Back off period | The number of consecutive LIN Database cycles that a value must remain unchanged before a burst condition is considered to have cleared and normal audit trailing resumes.<br>Default value = 10 |
| Dynamic threshold | The percentage (in integer multiples) of LIN Database cycles that a value must have changed before a 'dynamic' condition is seen to have occurred. Once the dynamic condition occurs audit trailing will stop to prevent over-filling the log until the value stabilises gain. Default = 10 |

### B.5.2 Parameter Lines

Parameter lines are of the form depicted below. Each line identifies a field name to be monitored.

The syntax for identifiying a field name to be monitored is,

```
<BlockName>.<FieldName>
```

## B.6 CHARACTER SET REFERENCE

The following table contains representations of the ASCII character set (UNICODE Latin-1) available.

| Character | Code | Character | Code | Character | Code | Character | Code |
|---|---|---|---|---|---|---|---|
| Space | 20 | P | 50 | | A0 | Ð | D0 |
| ! | 21 | Q | 51 | ¡ | A1 | Ñ | D1 |
| “ | 22 | R | 52 | ¢ | A2 | Ò | D2 |
| # | 23 | S | 53 | £ | A3 | Ó | D3 |
| $ | 24 | T | 54 | ¤ | A4 | Ô | D4 |
| % | 25 | U | 55 | ¥ | A5 | Õ | D5 |
| & | 26 | V | 56 | ¦ | A6 | Ö | D6 |
| ‘ | 27 | W | 57 | § | A7 | × | D7 |
| ( | 28 | X | 58 | ¨ | A8 | Ø | D8 |
| ) | 29 | Y | 59 | © | A9 | Ù | D9 |
| * | 2A | Z | 5A | ª | AA | Ú | DA |
| + | 2B | [ | 5B | « | AB | Û | DB |
| , | 2C | \ | 5C | ¬ | AC | Ü | DC |
| - | 2D | ] | 5D | – | AD | Ý | DD |
| . | 2E | ^ | 5E | ® | AE | Þ | DE |
| / | 2F | _ | 5F | ¯ | AF | ß | DF |
| 0 | 30 | ‘ | 60 | º | B0 | à | E0 |
| 1 | 31 | a | 61 | ± | B1 | á | E1 |
| 2 | 32 | b | 62 | ² | B2 | â | E2 |
| 3 | 33 | c | 63 | ³ | B3 | ã | E3 |
| 4 | 34 | d | 64 | ´ | B4 | ä | E4 |
| 5 | 35 | e | 65 | µ | B5 | å | E5 |
| 6 | 36 | f | 66 | ¶ | B6 | æ | E6 |
| 7 | 37 | g | 67 | · | B7 | ç | E7 |
| 8 | 38 | h | 68 | ç | B8 | è | E8 |
| 9 | 39 | i | 69 | ¹ | B9 | é | E9 |
| : | 3A | j | 6A | º | BA | ê | EA |
| ; | 3B | k | 6B | » | BB | ë | EB |
| < | 3C | l | 6C | ¼ | BC | ì | EC |
| = | 3D | m | 6D | ½ | BD | í | ED |
| > | 3E | n | 6E | ¾ | BE | î | EE |
| ? | 3F | o | 6F | ¿ | BF | ï | EF |
| @ | 40 | p | 70 | À | C0 | ð | F0 |
| A | 41 | q | 71 | Á | C1 | ñ | F1 |
| B | 42 | r | 72 | Â | C2 | ò | F2 |
| C | 43 | s | 73 | Ã | C3 | ó | F3 |
| D | 44 | t | 74 | Ä | C4 | ô | F4 |
| E | 45 | u | 75 | Å | C5 | õ | F5 |
| F | 46 | v | 76 | Æ | C6 | ö | F6 |
| G | 47 | w | 77 | Ç | C7 | ÷ | F7 |
| H | 48 | x | 78 | È | C8 | ø | F8 |
| I | 49 | y | 79 | É | C9 | ù | F9 |
| J | 4A | z | 7A | Ê | CA | ú | FA |
| K | 4B | { | 7B | Ë | CB | û | FB |
| L | 4C | \| | 7C | Ì | CC | ü | FC |
| M | 4D | } | 7D | Í | CD | ý | FD |
| N | 4E | ~ | 7E | Î | CE | þ | FE |
| O | 4F | | 7F | Ï | CF | ÿ | FF |

Table B.8 ASCII Codes

# APPENDIX C TERMINAL CONFIGURATOR

This explains the complexities of using the Terminal Configurator program resident on the instrument.

The main topics of this chapter are:

- The Configurator (section C.1)
- Running the Configurator (section C.2)
- LIN Database configuration (section C.3)
- Modbus configuration (section C.4)

## C.1 THE CONFIGURATOR

Most configuration will be done before despatch, using the LINtools software package. This chapter explains how LIN Databases and communications parameters are configured for the instrument using the Configurator program resident within the instrument.

The Configurator program is mainly for adjusting existing configurations on site, usually to accompany modifications to the processing plant and can also be used to 'Load', 'Start', 'Stop', 'Save' and 'Monitor' LIN Databases, to perform various filing operations and 'Try' and 'Untry' changes to the running control strategy.

It employs the standard LIN function block structured approach. The *LIN Blocks Reference Manual* (Part no. HA082375U003) gives full details of the software LIN function blocks available for the control strategy, and how to configure the parameters.

*Note. Instruments operating in redundant configuration will not allow function blocks to be added or deleted unless Primary and Secondary are synchronised.*

### C.1.1 Configurable Items

The configurable items are configured using a menu/item selection procedure. Configuration of the LIN Database consists of carrying out one or more of the following:

- Installing function blocks in the running control strategy (MAKE)
- Creating duplicates of existing LIN function blocks (COPY)
- Deleting function blocks (DELETE)
- Inspecting and updating function blocks (INSPECT)
- Test changes to the running control strategy (TRY)
- Cancel the test, but keep the changes displayed on the Configurator (UNTRY)
- Accept changes to the running control strategy (APPLY)
- Cancel all changes to the running control strategy and return to last operational LIN Database (UNDO)
- Accessing the Utilities menu (UTILITIES), from which the user can START and STOP programs, SAVE and LOAD LIN Databases, and access the ELIN setup page

## C.2 RUNNING THE CONFIGURATOR

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

*Note HyperTerminal® is the only recommended method of accessing the Configurator. Other methods of accessing the Configurator may result in unforeseen consequences.*

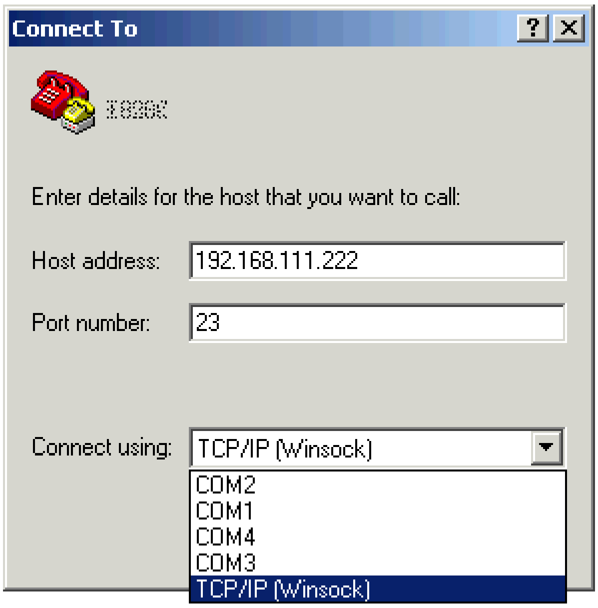
### C.2.1 Initial menu access

Using Windows™ XP as an example,

1. Power up the PC and start HyperTerminal® (Programs > Accessories > ... > HyperTerminal®). A 'new connection' sign-on screen appears.
2. Enter a name for the link and accept using the OK button. This will now reveal a Connect to dialog.
3. In the Connect using drop-down, select the TCP/IP (Winsock) option. After selection the fields above this drop-down now displays a Host and Port number field.

   *Note. The Configurator will only operate correctly if the VT100 is defined in the Emulation field, File > Properties > Settings page.*

4. After entering appropriate values to each of the required fields and confirming the changes, the sign-on screen will appear.




5. Press 1 to display the Initial menu, see Fig C.2.1b.

```
Telnet 149.121.165.188

Total Machine Control  - 1/0 at 66 MHz
(Hardware Build: RS485)
Serial number = 1426
Ethernet (MAC) address = 00:E0:4B:00:45:DA
IP address = 192.168.111.222
Subnet mask = 255.255.255.0
Default gateway = 0.0.0.0
POST result (0000) = SUCCESS
Last shutdown because: Successful Power Down

1  ANSI-CRT
>>>
```

Figure C.2.1a Typical sign-on screen

| | |
|---|---|
| Ethernet (MAC) address | Shows the address of the Ethernet interface. This value is unique and is permanently fixed for an individual instrument. |
| IP address | Gives the IP address currently assigned to this instrument. |
| Subnet Mask | Gives the subnet mask currently assigned to this instrument. An IP host uses the subnet mask, in conjunction with its own IP address, to determine if a remote IP address is on the same subnet (in which case it can talk directly to it), or a different subnet (in which case it must talk to it via the Default Gateway). |
| Default Gateway | Gives the IP address of the Default Gateway. It is the address via which this instrument must talk in order to communicate with IP addresses on other subnets. If undefined (0.0.0.0 or blank) then this instrument can only talk to other IP hosts on this same subnet. |

*Note. Refer to the ELIN User Guide (Part no. HA082429) for full details.*

® Hyperterminal is a trademark of Hilgraeve Inc.

### C.2.1 INITIAL MENU ACCESS (Cont.)

If Modbus is enabled, the Configurator *Initial menu* appears, see Figure C.2.1b. If Modbus is disabled, the *Main menu* appears instead, as shown in Figure C.3.

```
    INIT    Choose option

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

Figure C.2.1b  Initial menu

*Note. If the Initial or Main menu appears, this indicates that the instrument 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.

*Note. The next lower level of menu hierarchy can be accessed directly by simply pressing the initial letter of the menu item, e.g. on the Configurator initial menu above, pressing 'G' will select the GateWay menu item.*

| Function | Key combination |
|---|---|
| Redraw 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 C.2.1a  Cursor-control - equivalent key combinations

For keyboards without cursor-control keys, equivalent 'control' character combinations may be used, as indicated in Table C.2.1a. To use these, the <Ctrl> key is held down and the specified character typed.

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.

## C.2.2 The Initial menu

The Initial menu, Figure C.2.1b, lists two options – *Database* and *Gateway*. Database allows access to the Main menu for configuring a LIN Database, see Database Configuration section. Gateway allows access to the GateWay menu, for setting up a Modbus configuration.

## C.2.3 Quitting the Terminal Configurator

The instrument automatically exits configuration mode when the 'Telnet' session is closed.

*Note. If the Configurator is left running but unused, the user will eventually be locked out of the online operations, including Download, Start and Stop, and Online Reconfiguration.*

## C.3 LIN DATABASE CONFIGURATION

Most LIN Database configuration is completed before despatch, using the LINtools configuration tool. However, this basic Terminal Configurator is resident within the instrument allowing configuration of a LIN database from an appropriately configured PC.

When attempting to edit a control strategy that is running, only limited commands can be used, see Configurable Items section. The commands are accessed from the 'Utilities' menu, and permit 'Tentative' changes in a running control strategy. The 'Tentative' changes can be tested ('TRY' command) and accepted ('APPLY' command) if the required output is received. Continual changes can be attempted or discarded ('UNTRY' command) until the required value is obtained.

### START USING THE TERMINAL CONFIGURATOR

Following the successful start of a 'Telnet' session, and access from the Initial Menu, the Main menu appears.

Figure C.3 shows the Main menu.

```
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 C.3 Configurator Main menu

### C.3.1 MAKE command

Installs function blocks in the control strategy. Select MAKE to display the SET MENU, the instrument resident library of function block categories, as detailed in the *LIN Block Reference Manual* (Part no. HA082375U003). Figure C.3.1a shows part of the screen display when LOGIC is selected, as an example.

*Note. Every control strategy must contain a 'header' block, the only LIN function block initially available for a new control strategy.*

Select a category to list its function blocks. Select the function block to be installed. The function block *Overview* appears listing the function block parameters, default values and units in a double 3-column format. Figure C.3.1b shows the (default) overview for the PID block as an example.

*Note. Any function blocks added while the control strategy is running, online, are made as 'Tentative'. They will not become part of the running control strategy until either 'TRY' or 'APPLY' is selected from the Utilities menu.*

```
LOGIC        Select type
-----------------------------------------------
                                 > PULSE
                                   AND4
                                   OR4
                                   XOR4
                                   LATCH
                                   COUNT
                                   COMPARE
```

Figure C.3.1a Logic category menu

## C.3.1 MAKE command (Cont.)

## BLOCK OVERVIEW

Refer to Figure C.3.1b, which shows the main features of a typical function block overview, used to monitor and update function block parameters. (Overviews can also be accessed via the COPY and INSPECT main menu options.) The overview is equivalent to a LINtools *Object Properties pane* and its fields have the same meanings, although data entry is different.

*Note. Parameters being updated by incoming connections from other function blocks are not specially indicated in a function block overview.*

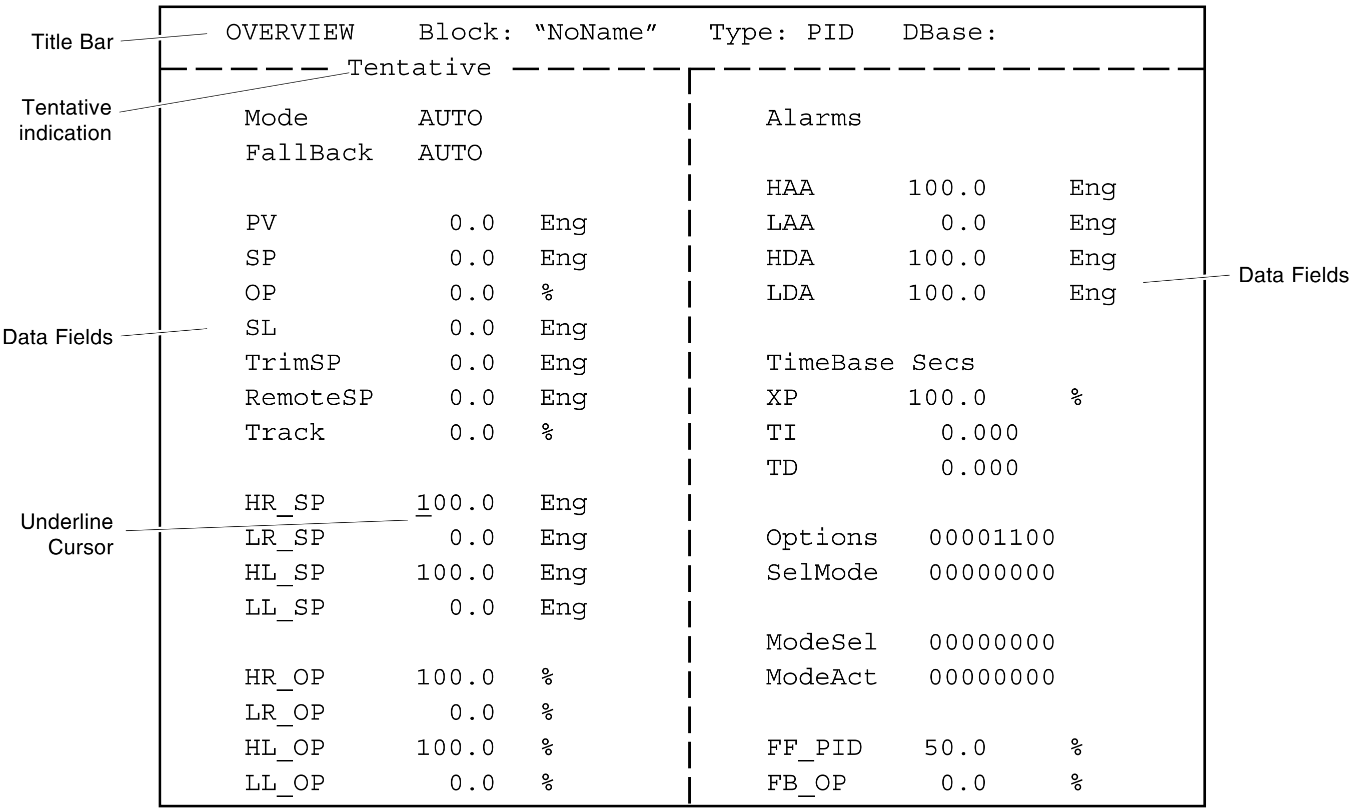



Figure C.3.1b Overview - PID block

TITLE BAR

Contains fields common to all overviews: *Block*, *Type*, and *DBase*. Details of these fields are to be found in the *LIN Blocks Reference Manual* (Part no. HA082375U003). A blank *DBase* field denotes that the LIN Database is local.

*Note. A function block is not added to the control strategy until (at the minimum) a block name has been assigned, i.e. tagname, and either the LIN Database has been restarted or APPLY operated in the Utilites menu. Using the TRY command will temporarily add the function block, until it is cancelled, using the UNTRY command.*

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.

### C.3.1 MAKE command (Cont.)

1 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 LIN Database unchanged, move the cursor to a function block field above or below the current field *before* pressing <Enter>, or press the <Escape> key.

Pressing <Enter> with the cursor on the first character of the *Block* or *DBase* fields (before starting to type) accesses a *Full Description* page (Figure C.3.1c shows an example). This page gives general information about the function block and has a common format.

```
FULL DESCRIPTION    Block: INP01       Type: ANIN
-----------------------------------------------------
         Request refresh                     0.1040
         Actual refresh                      0.105
         Server number                        3
         DBase:                               =Alpha
         Rate ms                             10
         Execute time                        1234
```

Figure C.3.1c FULL DESCRIPTION page for block (example)

| | |
|---|---|
| Block | Block tagname (Read/write) |
| Type | Block type (Read-only). |
| Request refresh | Configured time period (secs) for running the LIN function block. (Read-only). |
| Actual refresh | Time period (secs) since the function block was last run. (Read-only). |
| Server number | Function block's time scheduled task priority (Read/write). There are four User Tasks numbered from User Task 1 (highest priority) to User Task 4 (lowest priority). |
| DBase: | Name of the function block's LIN Database. A blank field denotes the LIN Database is local, i.e. is resident in this instrument. (LIN Database names and their LIN addresses are specified via the main menu NETWORK option, see Network section) (Read/write). |

*Note. Remote LIN Database names entered in the DBase field must be prefixed by an 'equals' sign (=).*

| | |
|---|---|
| Rate ms | Rate is the minimum update period (i.e. maximum rate) at which an individual cached function 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. Rate values are minimum update times only. Heavily loaded networks may not be able to reach the faster update rates.*

| | |
|---|---|
| Execute time | This is the time taken in microseconds to execute a LIN function block (including connections etc.). |

*Note. If the control strategy is running (online), the 'DBase' and 'Rate ms' fields cannot be edited. Only local function blocks can be made.*

### C.3.1 MAKE command (Cont.)

2 Parameter values.
Type in a value and press <Enter> to update the LIN Database. (Read-only parameters do not accept new values.) The instrument 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 C.3.1d shows an example).

```
FULL DESCRIPTION   Field: PV       Block: PID_1   Type: PID
-----------------------------------------------------------
    Value           80.1                          Real32
    Input           SIM 1.OP
```

Figure C.3.1d FULL DESCRIPTION page for parameter (example)

| | |
|---|---|
| Field, Block, Type | Read-only fields |
| Value | Parameter value, editable as for the Overview. (Read/write) |
| Real32 | Value type (Real32 = floating point number) (Read Only) |
| Input | Defines the source of any connection to the parameter from another function block, as Block Tagname.Output Mnemonic. A blank function block field means no connection. To make or edit a connection, type in the source function 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>. (Read/write) |

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

3 Parameter units.
Type in a value and press <Enter>. All other related units in the LIN 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).

### C.3.1 MAKE command (Cont.)

4 Options menu fields
Press <Enter> to display a pop-up menu of options for the field. Figure C.3.1e shows an example (PID Mode) in part of an *Overview* page.

```
OVERVIEW  Block: PID_1      Type:  PID                          DBase:
--------------------------------------------------------------------------------
  Mode     AUTO                               |  Alarms
  FallBack  > MANUAL                          |
              AUTO                            |  HAA        100.0           Eng
  PV          REMOTE          Eng             |  LAA          0.0           Eng
  SP          F_MAN           Eng             |  HDA        100.0           Eng
  OP          F_AUTO          %               |  LDA        100.0           Eng
  SL                          Eng             |
  TrimSP                      Eng             |  TimeBase  Secs
  RemoteSP    0.0             Eng             |  XP        100.0            %
  Track       0.0             %               |  TI        0.00
                                              |  TD        0.00
  HR_SP     100.0             Eng             |
  LR_SP       0.0             Eng             |  Options   00101100
  HL_SP     100.0             Eng             |  SeMode    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 C.3.1e Pop-up options menu (example)

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 **M** selects MANUAL; entering **F_M** selects F_MAN (Forced Manual).

5 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 C.3.1f 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
```

Figure C.3.1f Alarms page (example)

#### C.3.1 MAKE command (Cont.)

6 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 *Object Properties pane* bitfields). Figure C.3.1g 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 C.3.1g Full Description page for bitfield (example)

To connect an input to a bitfield, press the → key and type in the LIN function block name/field name from which the connection is to be made. A connection can be deleted simply by replacing the LIN function block name/field name in the bitfield with a <space>.

**Caution**

Any connections deleted while the control strategy is running (online), are marked as 'DeleteReq'. It can be edited further by adding a different connection to the bitfield. However, this new connection will not be used, and the existing connection remains part of the running control strategy until either 'TRY' or 'APPLY' is selected from the Utilities menu.

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

7 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 LIN INSTRUMENT DATABASE

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

1 Local connections.
These are connections between two function blocks that are both local to the LIN Database. The connection is always evaluated immediately prior to the execution of the destination LIN function 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.

#### C.3.1 CONNECTION TYPES IN A LIN DATABASE (Cont.)

2 Connections to cached function block.
These are connections whose destination function block is a cached copy of a function block in another instrument. The source of the connection can be either a local function block or another cached function block. Such connections are evaluated only if the source and destination data do not match. All cached function blocks in the LIN Database are processed at regular intervals, and whenever a change is detected a single field write is performed over the communications link.

3 Connections from cached function block to local function block
These are connections where the source function block is a cached copy of a function block in another instrument, and the destination function block is local to the LIN Database. All cached function blocks in the LIN Database are tested at regular intervals, and if a change in the function block data is detected, then all such connections out of the cached function block into local function blocks are evaluated. The connections are not evaluated if the source data has not changed. These connections minimise the load involved in synchronising the LIN Databases of a duplex pair, whilst ensuring the coherence of the data between the primary and secondary instruments.

**Caution**

With this third type of connection, tasks are allowed 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.

### C.3.2 COPY command

Creates duplicates of existing function blocks. Select COPY from the main menu to display all the function blocks in the control strategy, in semi-graphical format as shown in Figure C.3.2. The function blocks are displayed from left to right in order of creation. Move the cursor (**>**) to a function block and press <Enter>. The function 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 function block site numbers.

```
COPY  Select block
        +----+----+----+----+----+
          >T2550  SIM_1  TIC_100  PID_1  FIC_101
        +----+----+----+----+----+
```

Figure 8.3.2 COPY display (example)

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

*Note. Any function block copied while the control strategy is running, on-line, are made as ‘Tentative’. They will not become part of the running control strategy until either ‘TRY’ or ‘APPLY’ is selected from the Utilities menu.*

### C.3.3 DELETE command

Deletes function blocks from the control strategy.

*Note. Before deleting a function block all connections to and from it must be cleared. This is achieved simply by clearing the source fields of each affected connection, including the source fields of any input connection.*

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

*Note. Any function blocks deleted while the control strategy is running (online), are marked as 'DeleteReq'. They will not be removed from the control strategy until either 'TRY' or 'APPLY' is selected from the Utilities menu.*

### C.3.4 INSPECT command

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

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

*Note. All function blocks can be inspected while the control strategy is running, online.*

### C.3.5 NETWORK command

Allows a LIN database to be assigned to a specific LIN node address. This permits locally generated function blocks to be configured as 'cached' function blocks by changing the *DBase* field in the function block Title bar, see *Make* command. (The overview page of the cached function block *DBase* field specifies the remote LIN Database name.)

*Note. It is good practice when using cached function blocks, to cache at least one block in each direction. This allows the status of the communications 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 changeover in a redundant mode system.*

Select NETWORK from the main menu to display the *Network setup* page (initially blank). Figure C.3.5 shows the top part of an example page with several LIN Databases already assigned.

```
       Network setup
 ---------------------------------------------------------------
                                     |
                                     |
       Alpha         >  01           |
       Beta          >  02           |
       dBase_1       >  03           |
                                     |
```

Figure C.3.5 Network setup page (example)

To assign a new LIN 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 **>00**. Move the cursor to the default address and type in the required node address (two hex digits). Press <Enter> to assign the LIN Database to the specified node address.

*Note. Non-unique or invalid names are 'beeped' and not accepted. Do not use 00 or FF as node addresses.*

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.

*Note. External Databases (EDBs) cannot be created while the control strategy is running, online.*

## C.3.6 UTILITIES command

Allows program control, I/O calibration, and filing. Select UTILITIES from the main menu to display the Utilities Options page, shown in Figure C.3.6.

```
UTILITIES  Select option
-----------------------------------------------------
              >START     - Start runtime system
               STOP      - Stop runtime system
               SAVE      - Save database
               LOAD      - Load database
               FILE      - File page
               TRY       - Try Changes
               UNTRY     - Untry Changes
               APPLY     - Apply Changes
               UNDO      - Undo Changes
               ELIN      - Elin Setup
```

Figure C.3.6 UTILITIES options menu

### START, STOP COMMAND

Select START or STOP from the UTILITIES options menu and press <Enter> to start or stop the control program running in the instrument. If the control strategy program is in progress, '`Running`' appears below the first line in the Configurator, but will change to '`Stopped`' if the control strategy is halted.

*Note. When you START a LIN 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 COMMAND

Names and saves a control program to a specified memory area. Select SAVE from the UTILITIES options menu - the default filename specification, **E:<filename>.DBF** is displayed. (The prefix **E:** directs the save to the local E: drive area of the instrument; 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:<filename>.DBF**).

Type in a new specification if needed, then press <Enter> to execute the save. After a short pause the T280 instrument 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 LIN 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 LIN Database are carried out on the RAM image only, not directly to the filename.dbf file in E: drive. They are copied to E: drive (overwriting the existing filename.dbf file) automatically as you restart the LIN Database, or when you do a SAVE operation.*

### C.3.6 UTILITIES command (Cont.)

#### LOAD COMMAND

Retrieves a control program from a specified memory area and loads it to the instrument RAM.

*Note. A LOAD operation can be performed using the 'Load' option during online reconfiguration.*

Select LOAD from the UTILITIES options menu - the default filename specification, **E:<filename>.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 instrument 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 COMMAND

Permits access to the instrument 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). Press <Escape> to return to the UTILITIES menu.

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. The Find function has wild-card characters (?) to help you locate filenames containing known character strings).*

#### TRY/UNTRY CHANGES COMMAND

LIN Database changes can be Tried and Untried on a running LIN Database from the Configurator. If the control strategy has '`Tentative`' changes, '`Changes`' appears below the first line in the Configurator, but will change to '`Trying`' when testing the strategy. Any such changes made whilst the LIN Database is running are '`Tentative`', as indicated on the Configurator screen and are not applied until APPLY is selected. These '`Tentative`' changes can be discarded by selecting UNTRY, before APPLY has been selected. UNTRY has no effect once APPLY has been used.

*Note. If changes have been applied, and a synchronisation is attempted, it will fail unless the LIN Database running in the primary instrument has been saved using either the root LIN function block's full save option, or it is stopped, saved and started from the Configurator program.*

Select TRY or UNTRY from the UTILITIES options menu and press <Enter> to try or untry the '`Tentative`' changes to the control strategy running in the instrument.

#### APPLY/UNDO COMMAND

LIN Database changes can be executed online from the Configurator. Any such changes made whilst the LIN Database is running are '`Tentative`' and are not applied until APPLY is selected. These '`Tentative`' 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 synchronisation is attempted, it will fail unless the LIN Database running in the primary instrument has been saved using either the root function block's full save option, or it is stopped, saved and started from the Configurator program.*

### C.3.6 UTILITIES command (Cont.)

### ELIN SETUP PAGE COMMAND

The *ELIN Setup* page allows the instrument's 'network.unh' file to be configured.

*Note. The Network configuration can be edited using the Instrument Properties dialog via the Project Environment or the instrument folder. The 'network.unh' file can also be edited using an appropriate text editor, e.g. 'notepad.exe'.*

```
Elin Setup (network.unh file)
-----------------------------------------------------------------------
                                        |
 LIN PROTOCOL SETUP                     | REMOTE SUBNET NODE LIST
                                        |
  Protocol Name        RKN              |   149.121.173.1
  All Subnet Enable    OFF              |
  Elin Only Enable     ON               |
                                        |
                                        |
 LOCAL IP SETUP                         |
                                        |
 Get Address Method Fixed               |
 IP Address           149.121.128.209   |
 Subnet               255.255.252.0     |
 Default Gateway      149.121.128.138   |
                                        |
                                        | TELNET
                                        |   Login Id
                                        |   Password      ********
```

Figure C.3.6 ELIN Setup page (example)

LIN PROTOCOL SETUP — This area of the screen allows specification of the items in the '[LIN]' section of the 'network.unh' file.

LOCAL IP SETUP — Allows the specification of those items in the '[IP]' section of the 'network.unh' file. The IP address etc. is entered using data obtained from the network administrator.

REMOTE SUBNET NODE LIST — Allows the user to enter the IP addresses of all the nodes with which it is required to communicate. (The '[PR]' section of the 'network.unh' file.)

Once all the required entries have been made, the ESC key should be operated. A confirmation message asks if the 'network.unh' file is to be updated. If 'Y', the file is updated and a power cycle is requested.

CROSS SUBNET WORKING — With 'All Subnet Enable' set 'OFF' (default), the instrument will not communicate ELIN cross subnet. This can be overridden in the network.unh file by setting 'All Subnet Enable' to 'ON'. This defines the behavior when the instrument is powered on. The ability to communicate cross subnet can be modified at run time by using the 'Options.AllSubnt' bit in the instrument's header block. Set to TRUE, this bit enables cross-subnet working. When set to FALSE, cross-subnet working is disabled.

*Note. This bit may be set FALSE, remotely, from a cross-subnet connection. If this is done, communications will be lost, and it will thus not be possible to reset it to TRUE from the cross-subnet connection.*

## C.3.7 ALARMS command

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 LIN function block containing the alarm.

## C.4 MODBUS CONFIGURATION

Most Modbus configuration is completed before despatch, using the Modbus configuration tool.  However, this basic Terminal Configurator is resident within the instrument and permits both offline configuration and online reconfiguration.

Following the successful start of a 'Telnet' session, and access from the Initial Menu, the Gateway Modbus Configuration menu appears, see Figure C.4.

```
GATEWAY  MODBUS Configuration
-------------------------------------------------------------
                       >GWindex       - Select GW index
                        MODE          - Operating mode
                        INTERFACE     - Select interface
                        SETUP         - Configure interface
                        TABLES        - Register and bit configuration
```

Figure C.4    Modbus Configurator Main menu

### C.4.1 GWindex command

This command only appears in products that support multiple GW indices, see Figure C.4.1.

Select the GWindex number to be viewed by the Configurator.  This is limited from 1 to the maximum number of GW indices supported by the instrument, e.g. 3 for the T2550.  The filename from where the GW index number was loaded appears in the Filename field.

```
GWindex Select GWindex
-------------------------------------------------------------
         GWindex    1
         Filename   ABCDEFGH
```

Figure C.4.1    GWindex menu

### C.4.2 MODE command

Sets the operating state of the instrument to *Modbus Slave* or *Modbus Master*.

```
MODE  Operating mode
-------------------------------------------------------------
                 Mode  > Slave
                         Master
```

Figure C.4.2    OPERATING MODE menu

## C.4.3 INTERFACE command

Sets the Interface Type and Interface Instance of the instrument via enumerated lists, see Figure C.4.3.

```
INTERFACE Select interface
--------------------------------------------------------

          Type        Serial  > Serial
          Port        COM1      TCP/IP
```

Figure C.4.3 INTERFACE menu

Select the Interface Type, Serial or TCP/IP, used to communicate with the Modbus instrument and then define the Port it is connected to.

*Note. Individual Modbus specifications are described in the appropriate instrument handbook.*

## C.4.4 SETUP command

Configures the selected Interface Type and Interface Instance of the instrument defined in the INTERFACE menu. Selecting SETUP displays a menu that is dependant on the INTERFACE and MODE configurations.

- Serial master

  If the Serial is selected in the INTERFACE menu and Master is specified in the MODE menu this SETUP menu will show the Baud rate, Parity, Stop bits, and Time out fields.

- Serial slave

  If the Serial is selected in the INTERFACE menu and Slave is specified in the MODE menu this SETUP menu will show the Baud rate, Parity, Stop bits, Time out, and Slave No. fields.

```
SETUP    Configure interface
--------------------------------------------------------

         Baud rate  2400
         Parity     Odd
         Stop bits       2
         Instr No      >63
         Time out   1.000 secs
```

Figure C.4.4 Typical TCP/IP Slave SETUP menu

- TCP master

  If the TCP/IP is selected in the INTERFACE menu and Master is specified in the MODE menu this SETUP menu will show the Time out field only.

- TCP slave

  If the TCP/IP is selected in the INTERFACE menu and Slave is specified in the MODE menu this SETUP menu will show the Port no, Instr No, Time out, and CNOMO fields.

*Note. If the instrument supports CNOMO registers, this field indicates that Register Offset values 121, to 124 will display specific Manufacturer and Product details.*

### C.4.4 SETUP command (Cont.)

This page gives general information about the Interface configuration.

| | |
|---|---|
| Port no | TCP/IP Interface and Slave Operating Mode only. It shows the TCP port via which this modbus-TCP-slave instance communicates. 0 = default = 502. |
| 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. |
| 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*. |
| Line type | Shown only if both Serial Interface is selected and instruments supports software selection of 3-wire/5-wire operation. |

*Note. This is not currently supported.*

| | |
|---|---|
| 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. |
| Instr No | Slave Operating Mode only. Input an 'instrument 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. |

## C.4.5 TABLES command

Shows the Tables List dependant on the MODE configuration. To view the tables list, highlight TABLES and press <Enter>. Individual menus can be displayed by selecting the required Table number, see Table Menus.

### TABLES LIST

The Tables List provides an overview of all the tables in the Modbus configuration. Each instrument will support a maximum number of Tables as defined by the MAX_TABLES field in the instrument Configuration (Header) block. The Tables List offers sixteen tables per page, therefore an instrument supporting 64 Tables, e.g. T2550, will cover 4 pages.

This menu allows tables to be created and the types, offsets, sizes, and for master mode, function codes, scan counts, instrument numbers and tick rate to be specified. The Tables List also accesses individual Table Menus for detailed configuration,i.e. LIN Database mapping, see Table Menus section.

The Tables List menu below, Figure C.4.5a, 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 Master and Slave Operating Modes. The remaining, Functions, Scan count, Instr No, and TickRate appear only when Master Operating Mode is configured.

| Table | Type | Offset | Count | Functions | | | | Scan count | Instr No | TickRate |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Register | 0 | 16 | 3 | 4 | 6 | 16 | 16 | >00 | 100 |
| 2 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 3 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 4 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 5 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 6 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 7 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 8 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 9 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 10 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 11 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 12 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 13 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 14 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 15 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |
| 16 | Unused | 0 | 0 | - | - | - | - | 0 | >00 | 0 |

Figure C.4.5a Typical Master Mode Table menu

### C.4.5 TABLES command (Cont.)

This page gives general information about the Modbus Table configuration.

Table — This is the Table number, which is not editable. Highlight and <Enter> a Table number field to display the information related to the selected Table number. For a table with a Type other than Unused, the table menu for that table is displayed, see Table Menu.

Type — This field, defaults to Unused, allows the Table Type to be created or edited. 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. Other fields in the Tables List associated with the selection automatically adopt default values.*

The Type options are:

Unused — The table does not exist.

Register — This type of table maps LIN Database parameters to standard 16-bit Modbus registers.

Digital — This type of table maps LIN digital, boolean or alarm values to 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 LIN Database.

Offset — This field selects the start address of the table on the Modbus network. These values are the actual values used in the address field of the Modbus messages, i.e. the ‘protocol addresses’.

*Note. PLCs differ in the correspondence between their register or bit addresses and the protocol addresses.*

Count — This field shows 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 the default Modbus function codes that can be used with a particular Modbus table type to be enabled or disabled. 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.

Instr 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.

Tick Rate — Each table of registers is assigned a Tick Rate, a value between 0 and 65535 ms, to define the frequency at which it is scanned. The Tick Rate associated with each table can be configured. If the LIN instrument does not support Tick Rates, and/or if the instrument is configured to operate in Slave mode, the Tick Rate fields are disabled.

### C.4.5 TABLES command (Cont.)

## TABLE MENUS

You access an individual table menu 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 the mapping between the LIN Database fields and the Modbus addresses to be configured.

The Table Menu below, Figure C.4.5b, shows an example of the default Table Menu for a Register (or Diagnostic) Table.

*Note. Table headings differ between Register and Digital Tables, but some fields are common to both, e.g. Field, DB Write, and MOD Write.*

| 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 C.4.5b Typical Master Mode Table menu

This page gives detailed information about the selected table configuration.

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 Table List. The remaining (read-only) addresses follow on consecutively.

Digital — Digital tables only. This column shows the Modbus address of the digital bit on the selected 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. The first bit address in the table takes its value from the *Offset* given to the table via the Table 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).

### C.4.5 TABLES command (Cont.)

Field — This is the LIN Database field that the Modbus address is mapped to. It can remain blank. Select a field with the cursor and type in and enter a LIN finction block name plus parameter (and subfield if needed), separated by periods, e.g. **PV1.Alarms.Software**.

*Note. If attempting to enter an analogue parameter into a digital table Field, the entry is ignored. However any type of parameter can be entered in a register (or diagnostic) table. If attempting to enter or overwrite a LINDatabase parameter that would force an entry lower down the table to change its address (Digital value), the edit is ignored.*

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. 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 LIN 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') 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**'. To create a reverse 32-bit register pair, enter 's' (or 'S') in the *DP* field of the first register of the pair. 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 LIN Database field, the second register automatically copies the same field name; assigning the name and the *DP* can be done in either order. To restore a 32-bit register pair to individual 16-bit registers change the first register's *DP* to 0-4.

### C.4.5 TABLES command (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. Editing a Width value is not permitted if this would force an entry lower down the table to change its address (Digital value).*

DB Write — This column prevents the 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 prevents the 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.

*Note. The easiest way to globally protect an entire table, in a Modbus Gateway facility operating in Modbus Master mode, is to disable its write function codes (5 and 15, or 6 and 16) in the Tables List.*
*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.

*Intentionally left blank*

# Index

T820 Handbook

**S (Cont.)**

# Eurotherm: International sales and service


**AUSTRALIA Sydney**
Eurotherm Pty. Ltd.
Telephone (+61 2) 9838 0099
Fax (+61 2) 9838 9288
E-mail info.eurotherm.au@invensys.com

**AUSTRIA Vienna**
Eurotherm GmbH
Telephone (+43 1) 798 7601
Fax (+43 1) 798 7605
E-mail info.eurotherm.at@invensys.com

**BELGIUM & LUXEMBOURG Moha**
Eurotherm S.A/N.V.
Telephone (+32) 85 274080
Fax (+32 ) 85 274081
E-mail info.eurotherm.be@invensys.com

**BRAZIL Campinas-SP**
Eurotherm Ltda.
Telephone (+5519) 3707 5333
Fax (+5519) 3707 5345
E-mail info.eurotherm.br@invensys.com

**CHINA**
Eurotherm China
**Shanghai Office**
Telephone (+86 21) 6145 1188
Fax (+86 21) 6145 2602
E-mail info.eurotherm.cn@invensys.com
**Beijing Office**
Telephone (+86 10) 5909 5700
Fax (+86 10) 5909 5709 or
Fax (+86 10) 5909 5710
E-mail info.eurotherm.cn@invensys.com

**DENMARK Copenhagen**
Eurotherm Danmark AS
Telephone (+45 70) 234670
Fax (+45 70) 234660
E-mail info.eurotherm.dk@invensys.com

**FINLAND Abo**
Eurotherm Finland
Telephone (+358) 2250 6030
Fax (+358) 2250 3201
E-mail info.eurotherm.fi@invensys.com

**FRANCE Lyon**
Eurotherm Automation SA
Telephone (+33 478) 664500
Fax (+33 478) 352490
E-mail info.eurotherm.fr@invensys.com

**GERMANY Limburg**
Eurotherm Deutschland GmbH
Telephone (+49 6431) 2980
Fax (+49 6431) 298119
E-mail info.eurotherm.de@invensys.com

**INDIA Chennai**
Eurotherm India Limited
Telephone (+91 44) 2496 1129
Fax (+91 44) 2496 1831
E-mail info.eurotherm.in@invensys.com

**IRELAND Dublin**
Eurotherm Ireland Limited
Telephone (+353 1) 469 1800
Fax (+353 1) 469 1300
E-mail info.eurotherm.ie@invensys.com

**ITALY Como**
Eurotherm S.r.l
Telephone (+39 031) 975111
Fax (+39 031) 977512
E-mail info.eurotherm.it@invensys.com

**KOREA Seoul**
Eurotherm Korea Limited
Telephone (+82 31) 2738507
Fax (+82 31) 2738508
E-mail info.eurotherm.kr@invensys.com

**NETHERLANDS Alphen a/d Rijn**
Eurotherm B.V.
Telephone (+31 172) 411752
Fax (+31 172) 417260
E-mail info.eurotherm.nl@invensys.com

**NORWAY Oslo**
Eurotherm A/S
Telephone (+47 67) 592170
Fax (+47 67) 118301
E-mail info.eurotherm.no@invensys.com

**POLAND Katowice**
Invensys Eurotherm Sp z o.o
Telephone (+48 32) 218 5100
Fax (+48 32) 218 5108
E-mail info.eurotherm.pl@invensys.com

**SPAIN Madrid**
Eurotherm España SA
Telephone (+34 91) 661 6001
Fax (+34 91) 661 9093
E-mail info.eurotherm.es@invensys.com

**SWEDEN Malmo**
Eurotherm AB
Telephone (+46 40) 384500
Fax (+46 40) 384545
E-mail info.eurotherm.se@invensys.com

**SWITZERLAND Wollerau**
Eurotherm Produkte (Schweiz) AG
Telephone (+41 44) 787 1040
Fax (+41 44) 787 1044
E-mail info.eurotherm.ch@invensys.com

**UNITED KINGDOM Worthing**
Invensys Eurotherm Limited
Telephone (+44 1903) 268500
Fax (+44 1903) 265982
E-mail info.eurotherm.uk@invensys.com

**U.S.A Ashburn VA**
Eurotherm Inc.
Telephone (+1 703) 724 7300
Fax (+1 703) 724 7301
E-mail info.eurotherm.us@invensys.com


ED60



Represented by:

invensys
Foxboro

HA029131/5 (CN29111)