Mini8 - 8 Loop Process Controller

Issue Status of This Manual

Issue A of this manual applies to software version 1.04.

Mini8 8 Loop Process Controller

1. CHAPTER 1   INSTALLATION AND OPERATION

1.1 What Instrument Do I Have?

Thank you for choosing this Mini 8 Controller.

The Mini8 is a compact DIN rail mounting 8 loop PID controller and data acquisition unit. It offers a choice of I/O and a choice of field communications.

The Mini8 mounts on 35mm Top Hat DIN Rail. It is intended for permanent installation, for indoor use only, and to be enclosed in an electrical panel.

The Mini8 is pre-assembled in the factory to give the I/O required for the application as specified in the order code. With standard applications the Mini8 is also supplied configured. Alternatively the Mini8 is configured using Eurotherm's iTools configuration suite running on a personal computer.

Whenever the symbol ☺ appears in this handbook it indicates a helpful hint

1.2 Mini8 Ordering Code

1. Mini8	2. Loops	3. Programs	4. PSU	5. Comms	6. Units	7. I/O Slot1	8. I/O Slot2	9. I/O Slot3	10. I/O Slot4
11 Application	12 . Recipe	13. Wires	14. Manual	15. Config					

1	
MINI8	Mini 8 controller
2	Control Loops
ACQ	IO Acquisition only
8LP	8 Control loops
3	Programs
0PRG	No Programs
1PRG	1 Profile - 50 programs
4	PSU
VL	24Vdc
5	Communications
MODBUS	Non Isolated Modbus RTU slave
ISOLMBUS	Isolated Modbus RTU slave
DEVICENET	Devicenet Slave
6	Temperature Units
C	Centigrade
F	Fahrenheit

7-10	IO Slots 1-4
XXX	No module fitted
TC8	8 Channel TC Input
AO8	8 Channel 4-20mA output
DO8	8 Channel logic output
CT3	3 Channel CT input
11	Application
STD	No configuration
EC8	8 Loop Extrusion Controller
12	Wires
30	30 User Wires
60	60 User Wires
120	120 User Wires
250	250 User Wires
13	Recipes
None	No Recipes
RCP	8 Recipes
14	Manual
ENG	English
GER	German
FRA	French
SPA	Spanish
ITA	Italian
15	Configuration Software
NONE	No CD
ITOOLS	Itools CD & Mini8 documentation

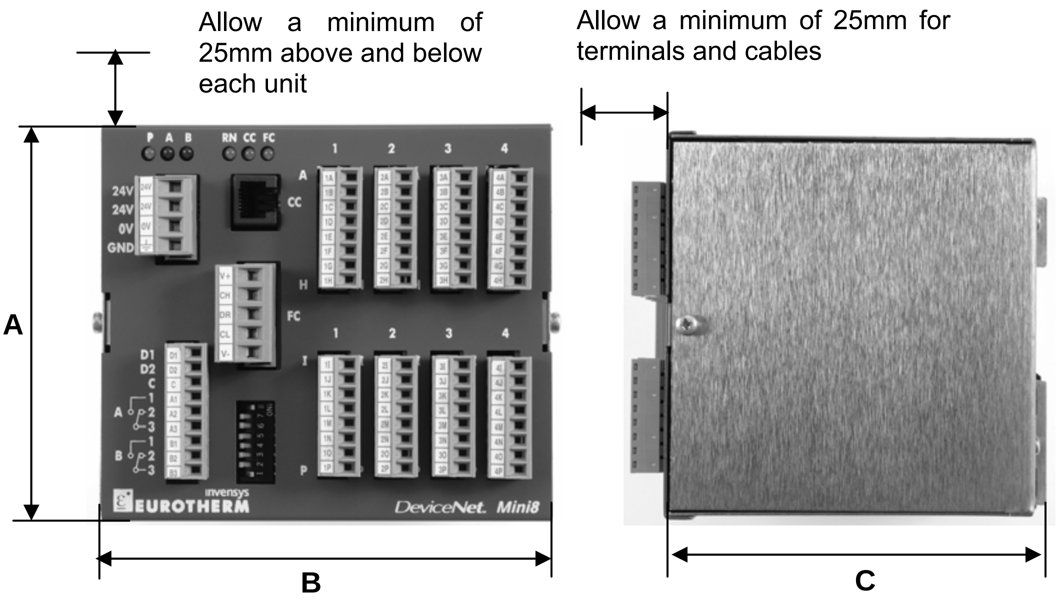
1.3 How to Install the Controller

This instrument is intended for permanent installation, for indoor use only, and to be enclosed in an electrical panel.

Select a location where minimum vibrations are present and the ambient temperature is within 0 and 50°C (32 and 122°F).

Please read the safety information, Appendix C at the end of this guide, before proceeding and refer to the EMC Booklet part number HA025497 for further information.

1.3.1 Dimensions



Dimension	mm
A	108
B	124
C	115

Figure 1-1: Mini8 Dimensions

1.3.2 To Install the Controller

1. Use 35mm symmetrical DIN Rail to EN50022-35 x 7.5 or 35 x 15,
2. Mount the DIN Rail horizontally as indicated in Figure 1.1. The Mini8 is NOT designed to be mounted in other orientations.
3. Hook the upper edge of the DIN rail clip on the instrument on the top of the DIN rail and push.
4. To remove use a screwdriver to lever down the lower DIN rail clip and lift forward when the clip has released.
5. A second unit on the same DIN rail may be mounted adjacent to the unit.
6. A second unit mounted above or below the unit requires a gap of at least 25mm between the top of the lower one and the bottom of the higher one.

1.3.3 Environmental Requirements

Mini8	Minimum	Maximum
Temperature	0°C	55°C
Humidity (non condensing)	5% RH	95% RH
Altitude		2000m

1.4 Electrical Connections

The Mini8 is intended for operation at safe low voltage levels. No Voltage above 42 volts should be applied to the system on any terminal.

A protective earth connection is not required but a good earth connection is required to provide a ground for EMC purposes.

Do not replace the battery. Return to factory if replacement battery is required.

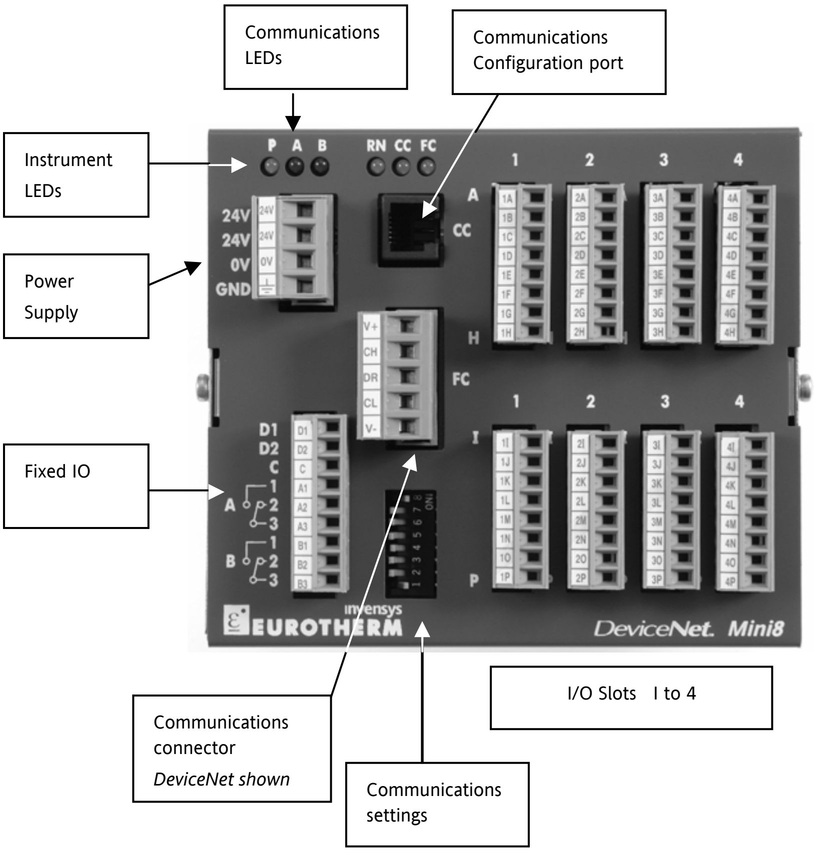
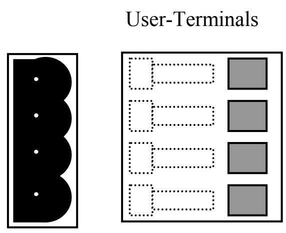


Figure 1-2: Terminal Layout for Mini8 Controller

1.4.1 Power Supply

The power supply requires a supply between 17.8 to 28.8 V dc, 15 watts maximum

24V	Ø	24 V dc
24V	Ø	24 V dc
0V	Ø	0 V dc
GND	Ø	Ground



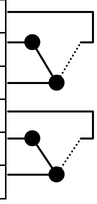
Connector terminals will accept wire sizes from 0.5 to 2.5, 24 to 12 awg.

Note: If the Min8 is used with the VT505 panel ensure that the power supply connectors **cannot** be mistakenly changed over. The connectors are physically the same, but the electrical connections are not compatible. Plugging the VT505 connector into the Mini8 will short-circuit the 24 volt supply.

1.4.2 Fixed IO Connections

These I/O are part of the power supply board and are always fitted.

D1	Ø	Digital Input 1
D2	Ø	Digital Input 2
C	Ø	Digital Input common
A1	Ø	Relay A n/open
A2	Ø	Relay A n/closed
A3	Ø	Relay A common
B1	Ø	Relay B n/open
B2	Ø	Relay B n/closed
B3	Ø	Relay B common



Digital Inputs : ON requires > 10.2V with 2mA drive, 30V max.

Relays contacts: 1 amp max, 42Vdc.

1.4.3 Digital Communications Connections

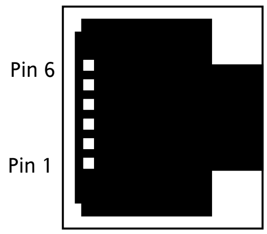
Two communications connections are fitted – a Modbus Configuration port (RJ11) and a Fieldbus port.

The Fieldbus is either Modbus (2 x RJ45) or DeviceNet

1.4.4 Configuration Port

The configuration port (Modbus) is on an RJ11 socket, just to the right of the power supply connections. It is a point to point RS232 connection. Eurotherm supply a standard cable to connect a serial COM port on a computer to the RJ11 socket, part no. **SubMin8/cable/config**.

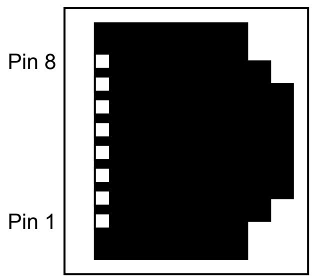
9 pin DF to PC COM port (RS232)	RJ11 Pin	Function
-	6	N/c
3 (Tx)	5	Rx
2 (Rx)	4	Tx
5 (0v)	3	0v (gnd)
	2	N/c
	1	Reserved



1.4.5 Modbus

For a full description of the installation of a communications link, including line matching resistors, see Eurotherm 2000 series communications handbook, part no. HA026230.

RJ45 pin	3 wire	5 wire
8	Receive (RX+)	RxA
7	Transmit (TX+)	RxB
6	Common	Ground
5		
4		
3	Ground	Ground
2	D+	TxA
1	D-	TxB



Two RJ45 sockets are provided – one for the incoming connection, the second to loop onto the next instrument or for a line terminator.

For the address switch see Chapter 10.3.2

The RS485 standard allows one or more instruments to be connected (multi dropped) using a two wire connection, with cable length of less than 1200m. 31 instruments and one master may be connected.

To use RS485, buffer the RS232 port of the PC with a suitable RS232/RS485 converter. The Eurotherm Controls KD485 Communications Adapter unit is recommended for this purpose. The use of a RS485 board built into the computer is not recommended since this board may not be isolated, which may cause noise problems or damage to the computer, and the RX terminals may not be biased correctly for this application.

Either cut a patch cable and connect the open end to the KD485 converter or, using twin screened cable, crimp an RJ45 plug on the Mini8 end.

The communication line must be daisy chained from device to device and, if the communications line is more than a metre or two long, it must be correctly terminated. A Modbus terminator containing the correct termination resistors is available from Eurotherm, order code:

SubMin8/RESISTOR/MODBUS/RJ45. The Modbus terminator is BLACK.

See also the 2000 series Communications Handbook, part number HA026230, available on www.eurotherm.co.uk. for further information on digital communications.

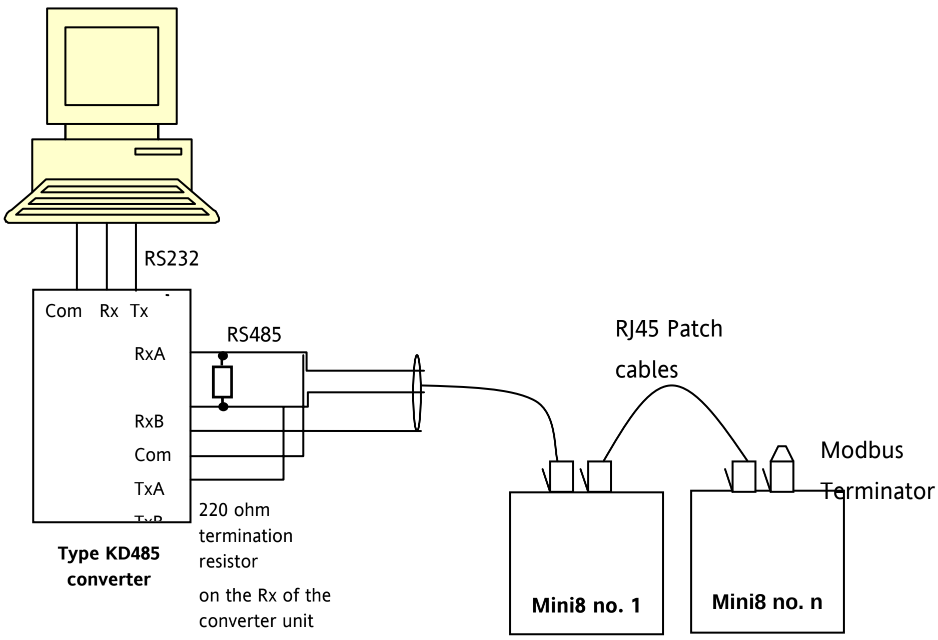


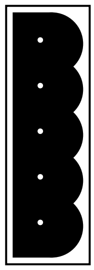
Figure 1-3: RS485 two-wire Connections

For the 4 wire connection the TxA and TxB are not connected to RxA and RxB but connected separately through another twisted pair.

1.4.6 DeviceNet

DeviceNet uses the CAN open connector screw terminal, 5 way with 5.08mm pitch. The mating DeviceNet connector (female Open Connector) is supplied to facilitate screw-in user wiring. The DeviceNet bus is powered (24V) from the system network, not from the instrument. The Mini8 requirement is a load of around 100mA. For the address switch see Chapter 10.5

Legend	Function
V+	V+
CH	CAN HIGH
DR	DRAIN
CL	CAN LOW
V-	V-



The DeviceNet specification states that the bus terminators (121 ohm) should not be included as any part of a master or slave. They are not supplied but should be included in the cabling where required.

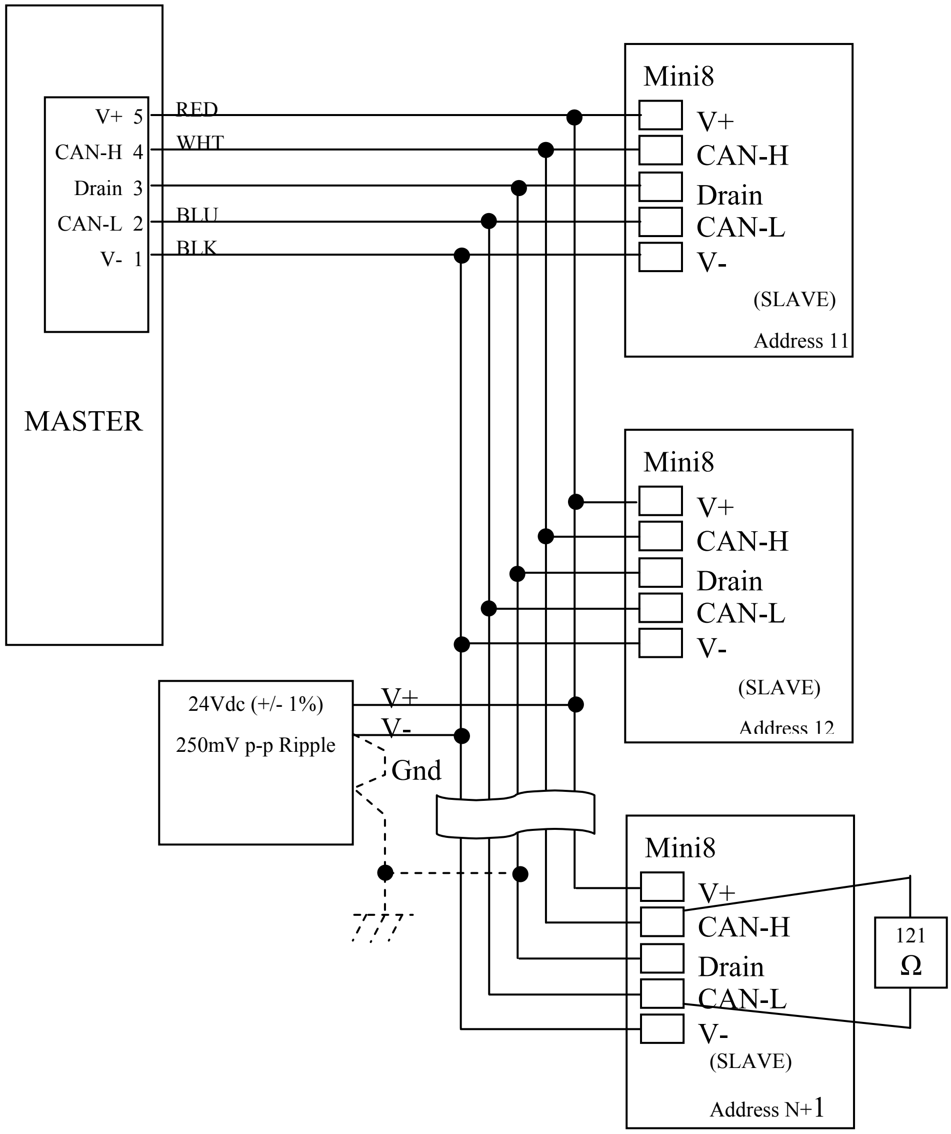
Mini8 Label	Colour	Description
V+	Red	DeviceNet network power positive terminal. Connect the red wire of the DeviceNet cable here. If the DeviceNet network does not supply the power, connect the positive terminal of an external 11-25 Vdc power supply.
CAN_H	White	DeviceNet CAN_H data bus terminal. Connect the white wire of the DeviceNet cable here.
SHIELD	None	Shield/Drain wire connection. Connect the DeviceNet cable shield here. To prevent ground loops, the DeviceNet network should be grounded in only one location.
CAN_L	Blue	DeviceNet CAN_L data bus terminal. Connect the blue wire of the DeviceNet cable here.
V-	Black	DeviceNet network power negative terminal. Connect the black wire of the DeviceNet cable here. If the DeviceNet network does not supply the power, connect the negative terminal of an external 11-25 Vdc power supply.

Network length depends on Baud rate.

Network Length	Varies w/speed, up to 4000m possible w/repeaters		
Baud Rate	**125**	**250**	**500**
Thin trunk	100m (328ft)	100m (328ft)	100m (328ft)
Max drop	6m (20ft)	6m (20ft)	6m (20ft)
Cumulative drop	156m (512ft)	78m (256ft)	39m (128ft)

See the DeviceNet Communications Handbook HA027506

1.4.7 Typical DeviceNet Wiring Diagram



1.4.8 Thermocouple Input TC8

The thermocouple module takes 8 thermocouples. It may be placed in any slot in the Mini8. Up to 4 may be fitted in a Mini8. Each input can be configured to any thermocouple type or a linear mV input. In version 1.04 the linear input can be scaled to engineering units using an Analogue Operator block.

A	Ø	TC1 +
B	Ø	TC1 -
C	Ø	TC2 +
D	Ø	TC2 -
E	Ø	TC3 +
F	Ø	TC3 -
G	Ø	TC4 +
H	Ø	TC4 -

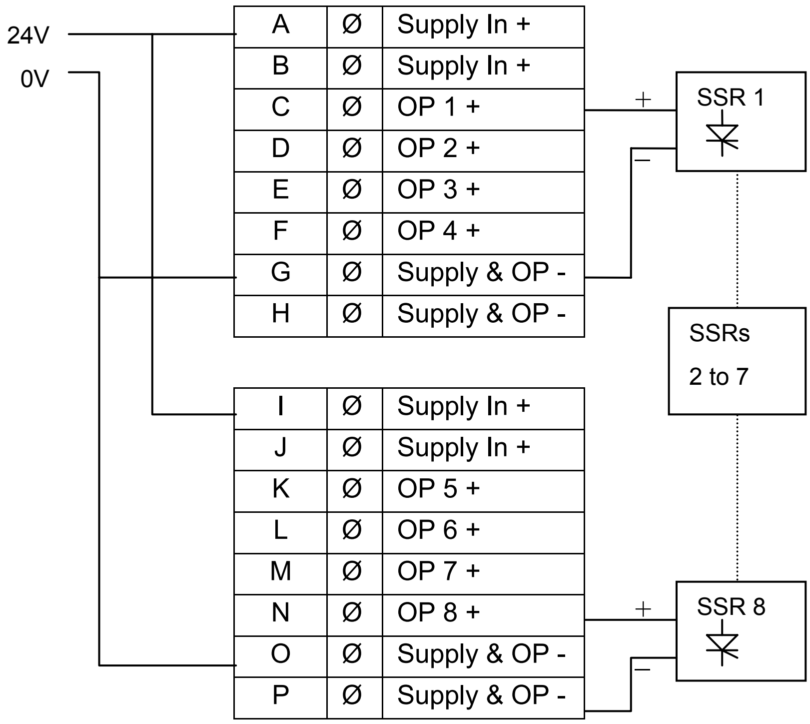
I	Ø	TC5 +
J	Ø	TC5 -
K	Ø	TC6 +
L	Ø	TC6 -
M	Ø	TC7 +
N	Ø	TC7 -
O	Ø	TC8 +
P	Ø	TC8 -

Channel to channel isolation is 42V.

Channel to system isolation is 42V.

1.4.9 Logic Output DO8

The DO8 module provides 8 logic outputs.



Channel to channel isolation – none.

Channel to system isolation – none if power supply shared with Mini8.

42V if independent isolated 24V power supply is used.

Supply In + (A,B,I,J) are all linked internally.

Supply In – (G,H,O,P) are all linked internally.

1.4.10 Analogue Output AO8

The AO8 modules provides 8 analogue outputs of 0 to 20 mA , max load 360 ohm.

Only one module may be fitted and in slot 4 only.

A	Ø	OP 1 +
B	Ø	OP 1 -
C	Ø	OP 2 +
D	Ø	OP 2 -
E	Ø	OP 3 +
F	Ø	OP 3 -
G	Ø	OP 4 +
H	Ø	OP 4 -

I	Ø	OP 5 +
J	Ø	OP 5 -
K	Ø	OP 6 +
L	Ø	OP 6 -
M	Ø	OP 7 +
N	Ø	OP 7 -
O	Ø	OP 8 +
P	Ø	OP 8 -

Channel to channel isolation is 42V.

Channel to system isolation is 42V.

1.4.11 Current Transformer input Module CT3

This provides inputs for 3 current transformers. The heater load cables are threaded through the transformers. Each input is 50mA max into 5 ohms.

A	Ø	Reserved
B	Ø	Reserved
C	Ø	Reserved
D	Ø	Reserved
E	Ø	Reserved
F	Ø	Reserved
G	Ø	Reserved
H	Ø	Reserved

I	Ø	In 1 A
J	Ø	In 1 B
K	Ø	no connection
L	Ø	In 2 A
M	Ø	In 2 B
N	Ø	no connection
o	Ø	In 3 A
P	Ø	In 3 B

The current transformers provide channel isolation; there is no channel to channel isolation in the module.

It is recommended that the current transformer is fitted with a voltage limiting device such as two back to back zener diodes between 3 and 10 volts, rated for 50mA.

There are 3 CT inputs, one for each phase. Up to a maximum of 16 heaters may be threaded through the CTs but with a further limit of 6 heater wires through each individual CT.

See Chapter 7.6 for typical circuit arrangements.

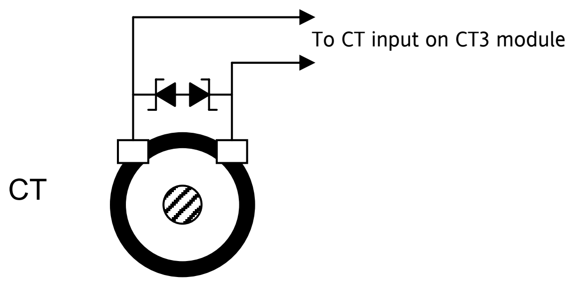


Figure 1-4: CT Input Protection

1.5 Adding or replacing an IO module.

The modules are not covered leaving antistatic sensitive electronic devices exposed. Take full antistatic protection when replacing modules by working on an earthed mat with an earthed wrist strap. Avoiding touching components, keep fingers on the green connectors or the edge of the printed circuit boards.

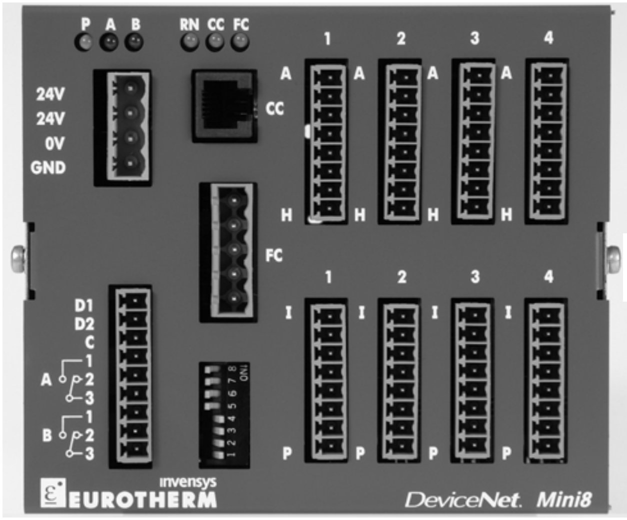


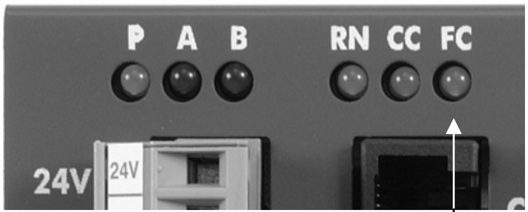
Figure 1-5: Mini8 Cover Retaining Screws

1. Remove all connectors.
2. Remove the 2 screws indicated above
3. Remove the cover.
4. If removing a module gently prise it out using the green connectors.
5. Insert the new module carefully using the guides on the side of the case to help to line up the lower connector with its mate on the motherboard. This requires great care as the guides provide mechanical support rather than being plug in guides.
6. Once you are certain the two connectors are lined up, push the module **gently** into place. Do NOT force.
7. Replace cover and the 2 cover screws.
8. Replace all connectors onto their correct modules.

1.6 Mini 8 LED Indicators

Two sets of 3 LEDs on the front panel indicate the power, the output relays and the status of the Mini8 and communications activity.

Legend	**P**	**A**	**B**
LED	Green	Red	Red
Function	Indicates 24V	Relay A state	Relay B state
OFF	No power	De-energised	De-energised
ON	Powered	Energised	Energised



Legend	**RN**	**CC**	**FC (Modbus)**	**FC (DeviceNet)**
LED	Green	Green	Green	Green
Function	Indicates run mode	Indicates Configuration	Field comms activity	Status
OFF	Not running	--	Offline	Offline
Blinking	Standby	Config traffic	Traffic	Ready
ON	Running	--	Running	Connected

The Mini8 is controlling normally ONLY if the green RN LED is permanently ON.

2. CHAPTER 2 USING THE MINI8

The Mini8 Controller does not have a display. The only means of configuring it, and of interfacing with it during normal operation is via communications.

The auxiliary communications port (RJ11) gives a Modbus interface, usually connected to iTools for configuration and commissioning.

The main configuration port offers Modbus or DeviceNet, normally connected to the system of which the Mini8 is part, and is the means by which the Mini8 is operated.

Here are ways the Mini8 may be used in a system. iTools is the best PC based solution. The Modbus single register addressing is best for Operator panels, PLCs where floating point may not be available or necessary. Use the Modbus floating point addressing with care.

2.1 iTools

iTools offers a pc based solution. The iTools suite allows configuration, commissioning, trend graphs and logging with OPC Scope, Program Editing, Recipes and User pages with View Builder.

2.1.1 iTools OPC Open server

With an OPEN OPC server running on a PC all the Mini8 parameters are available to any third party package with an OPC client. The advantage of this is that all the parameters are addressed by name – the iTools OPC server handles all the physical communication addresses. An example would be with Wonderware inTouch using OPCLink. In this situation the user would not have to know any of the parameter addresses, and would just select a parameter by browsing through the namespace.

e.g. Eurotherm.ModbusServer.1.COM1.ID001-Mini8.Loop.1.Main.PV

2.2 Modbus, single register, SCADA addressing

The key parameters of the Mini8 are available at a fixed address, independent of its configuration. This can be used with any device with a serial Modbus master (Modbus function 4). The parameters are listed in full with their addresses in Appendix A.

This area does not have all the parameters within the instrument. If other parameters are required they can be obtained by using the **Commstab** folder. This allows up to 250 other parameters to be made available using indirection addressing. This is explained in Appendix A.

Also note that in this area the resolution (number of decimal points) has to be configured and the serial Master has to scale the parameter correctly. Again the **Commstab** folder offers an alternative solution where a parameter can be indirectly addressed and configured as a floating point or double register value.

2.3 Modbus (Floating Point)

During configuration the Modbus addresses of the parameters required by the system can be obtained from iTools in decimal or in hex format. This can be used with any device e.g. PC or plc, with a serial Modbus master, able to decode double register for floating point numbers (Modbus function 7) and long integers (Modbus function 8). These parameters are displayed on the iTools parameter lists.

Use this method with care. These parameter addresses are not fixed – a common parameter on two differently configured Mini8s may be at a different address. Similarly if the configuration of a Mini8 is changed, the addresses of some of the parameters may also have changed.

2.4 Fieldbus

The Mini8 may be ordered with the option of a DeviceNet slave. This comes pre-configured with the key parameters of the 8 PID loops and alarms (60 input parameters process variables, alarm status etc and 60 output parameters – setpoints etc.). See Appendix B.

2.5 Mini8 Execution

The nominal update of all inputs and function blocks is 110ms. However, in complex applications the Mini8 will automatically extend this time in multiples of 110ms.

For example, eight simple heat/cool loops each with two alarms (40 wires) will run at 110ms, while the full EC8 configuration will run at 220ms because of the extra wiring and functionality.

The communications traffic will also have some effect on the update rate.

For example, an application using every function block and all 250 wires will run at 220ms with light communications traffic but may be slowed to 330ms with heavy traffic.

Note that as loading changes, the sample rate may increase or decrease automatically. In order to recover to a faster sample rate, the Mini8 must be running consistently with processing power to spare for at least 30s.

2.6 The iTools Operator Interface

Much of this manual is about configuring the Mini8 with iTools. However iTools also provides an excellent commissioning tool and can be used as a long-term operator view if convenient.

First it is necessary to go 'on-line' to the Mini8(s). This assumes the communication ports have been wired up to the COM port on the iTools computer (Chapter 10).

2.6.1 Scanning

Open iTools and, with the controller connected, press 

on the iTools menu bar. iTools will search the communications ports for recognisable instruments. Controllers connected using the RJ11 configuration port or with the configuration clip (CPI), may be found at address 255 (as a single point to point connection) or on a multidrop RS485 or RS422 network will be found at the address configured in the controller.

The iTools handbook, part no. HA026179, provides further step by step instructions on the general operation of iTools. This and the iTools software may be downloaded from www.eurotherm.co.uk.

When an instrument is found on the network it will be shown as, for example

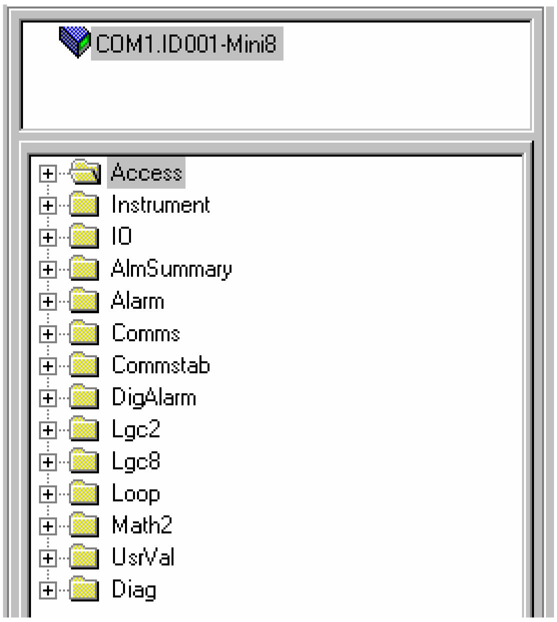
'COM1.ID001-Min8' which represents <computer com port>.ID<instrument address>-<Instrument type>

Stop the scan once all the instruments have been found.

Once an instrument is found on the network a message ''sync pending' or synchronizing' is displayed next to it whilst iTools extracts the exact configuration from the instrument. Wait until this message disappears.

2.6.2 Browsing and Changing Parameter Values

Once the instrument is synchronized the parameter navigation tree is displayed. The contents of this tree will vary depending on the actual configuration of the instrument.



The folders shown will be some of those which are always present –

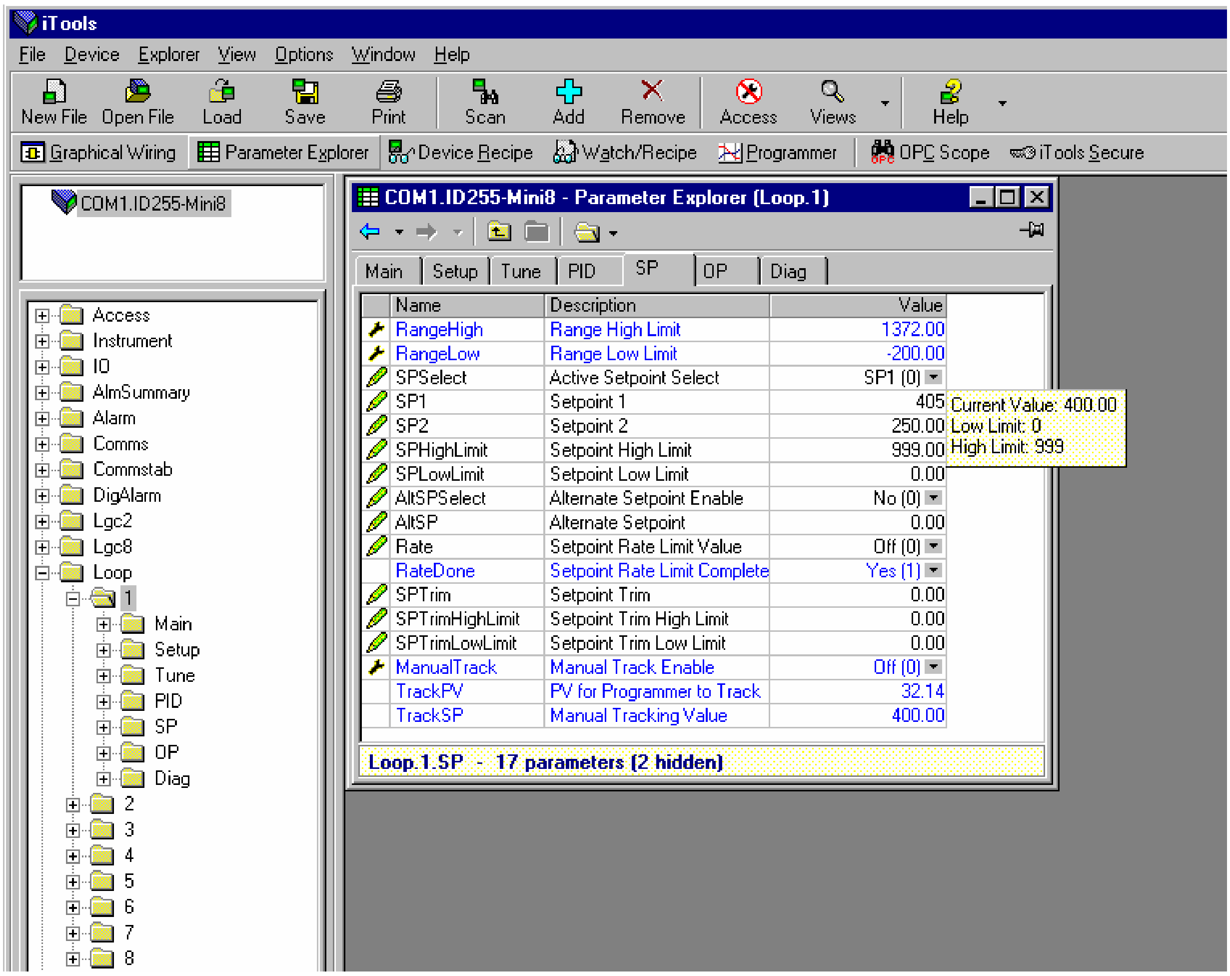
e.g Instrument, IO, Comms, Access

as well as the configuration dependent ones-

e.g. Loops, Alarm, Lgc2 etc. which have been configured.

To view or change a parameter:

1. Highlight the folder
2. Press Parameter Explorer to get the parameter window or open up the parameter list by clicking on the required folder. Right click in the parameter list to reveal or hide columns.
3. To change the value of a parameter,
   a. click the parameter value,
   b. write in the new value. Note a pop-up window indicates the current value, and the high and low limits.
   c. Hit <Enter> to enter the new value or <Escape> to cancel.



The 'Access' button puts the controller into configuration mode. In this mode the controller can be set up without its outputs being active. Press 'Access' again to return to operating level.

To find a parameter use the 'Find' tab at the bottom of the folder list.

☺ Tip: In parameter lists: Parameters in BLUE are read only

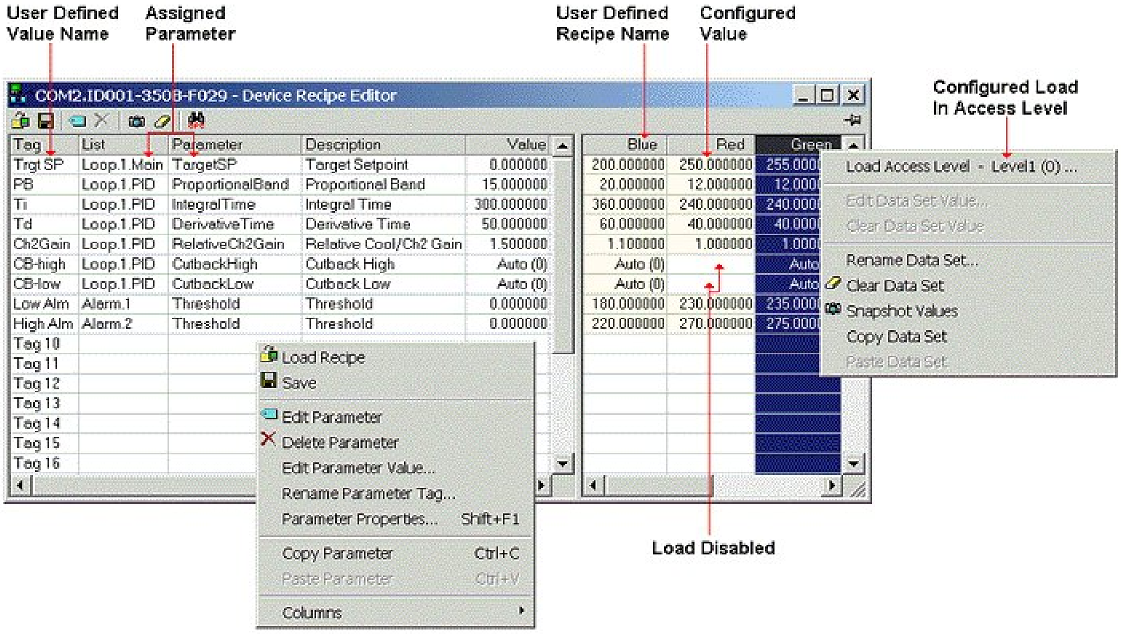
Parameters in BLACK are read/write.

☺ Tip: Every parameter in the parameter lists has a detailed description in the help file – just click on a parameter and hit Shift-F1 on the keyboard or right click and select parameter help.

2.7 Recipe Editor

Press Device Recipe for this feature. Up to 8 recipes can be stored. They can also be named by the user. Recipes allow the operator to change the operating values of up to 24 parameters in an instrument for different batch items/processes by simply selecting a particular recipe to load. Recipes are important for reducing error in setup and they remove the need for operator instructions fixed to the panel next to the instrument.

The Recipe Editor is used during configuration to assign the required parameters and to set up the values to be loaded for each recipe.



2.7.1 Recipe Menu Commands

Command	Description
Load Recipe	Used to load a recipe file into the instrument
Save	Used to save the current recipe configuration into a file
Edit Parameter	Used to assign a parameter to a Tag. Parameters can also be assigned by 'drag and drop' from the iTools parameter list
Delete Parameter	Used to delete an assigned parameter from the recipes
Edit Parameter Value	Used to edit the current value of the assigned parameter
Rename Parameter Tag	Allows the user to rename the Tag of the associated parameter. This tag is used on the instrument to identify assigned parameters (default Value1 - Value24)
Parameter Properties	Used to find the properties and help information of the selected parameter
Copy Parameter	Used to copy the currently selected parameter
Paste Parameter	Used to assign a previously copied parameter to the selected Tag
Columns	Used to hide/show the Description and Comment Columns
Load Access Level	Used to configure the lowest access level in which the selected recipe is allowed to load
Level1	Permitted to load when the instrument is in any of the access levels
Config	Permitted to load when the instrument is in the Config access level
Never	Never permitted to load
Edit Data Set	Used to edit the value of the selected assigned parameter within the selected recipe.

Command	Description
Value	Values can also be edited via double left clicking the value itself
Clear Data Set Value	Used to clear the value of the selected assigned parameter within the selected recipe, thus disabling it from loading when the recipe is selected to load
Rename Data Set	Allows the user to rename the selected recipe. This name is used to identify individual recipes (default Set1 - Set8). Note: Number of recipes dependent upon features
Clear Data Set	Used to clear all values in the selected recipe, thus disabling all from loading when the recipe is selected to load
Snapshot Values 	Used to copy all of the assigned parameters current values into the selected recipe
Copy Data Set	Used to copy all values of the selected recipe
Paste Data Set	Used to paste all values of a previously copied recipe into the selected recipe

2.8 OPCScope

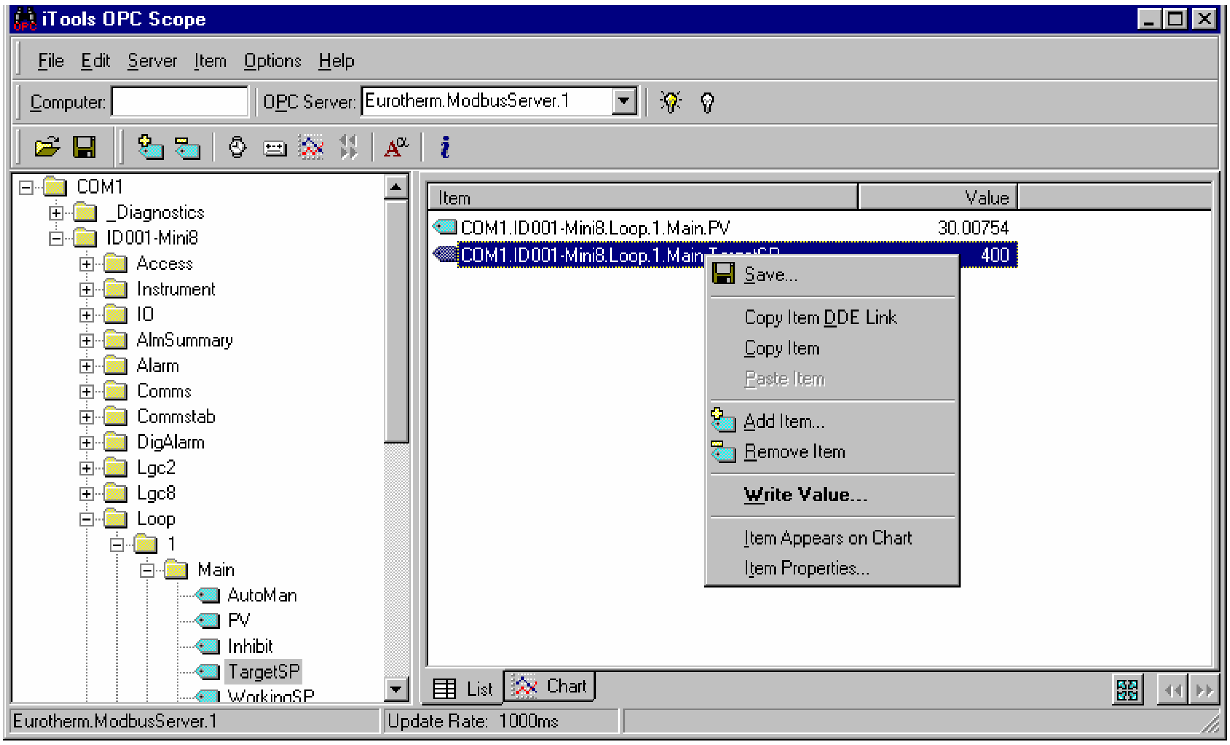
OPC scope is a standalone OPC client that can be used to attach to the iTools OPCserver. It offers real time trend charts and data logging to disc in a .csv (comma separated variable) format which can easily be opened by a spreadsheet such as Excel.

With iTools open OPC Scope can be started using the icon 

.

But it can also be started on its own using the Windows Start/Programs/Eurotherm iTools/OPC Scope

Select Server/Connect or click the icon  and the OPC server will start up (if it is not running) and will display the active ports on the computer. Opening the COM port will show the attached instruments as shown below.



The 'ID001-Mini8' folder will contain all the same folders for the instrument that would have been seen in iTools itself.

Expand the folder and double click on the blue item tag to add to the List Window. The List Window shows all the selected parameters and their current value.

Right click on a parameter to get the context menu.

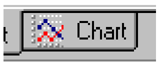
2.8.1 OPC Scope List Window Context Menu

Command	Description
Save	Saves the OPC Scope configuration as <filename>.uix See Section 2.8.3
Copy Item DDE link	Saves the DDE path to the clipboard. 'Paste Special' in an Excel cell and select 'Paste Link' and the current parameter value will be displayed in the cell.
Copy/Paste Item	Copy & Paste
Add Item	Add a new variable by name (easier to browse the navigation tree)
Remove Item	Remove the selected item.
Write Value	Write a new value (not if the item is Read Only).
Item appears on Chart	Up to 8 items can be tended on the Chart Window
Item Properties	Gives the item properties as seen by OPC

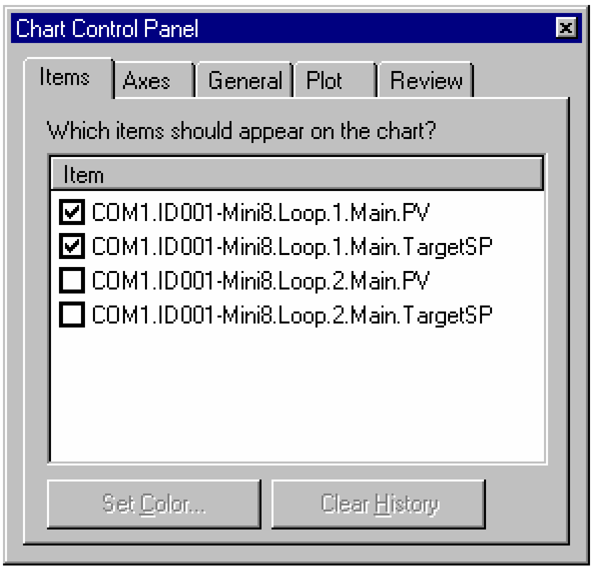
The OPC List can contain parameters from any instrument attached to the Modbus network.

If you have iTools Open (not iTools Standard) then OPC Scope can run on a remote networked computer. Enter the name of the server computer (attached to the instruments) the 'Computer' window and browse for the 'Eurotherm.ModbusServer1'.

2.8.2 OPC Scope Chart Window

Click the Chart tab 

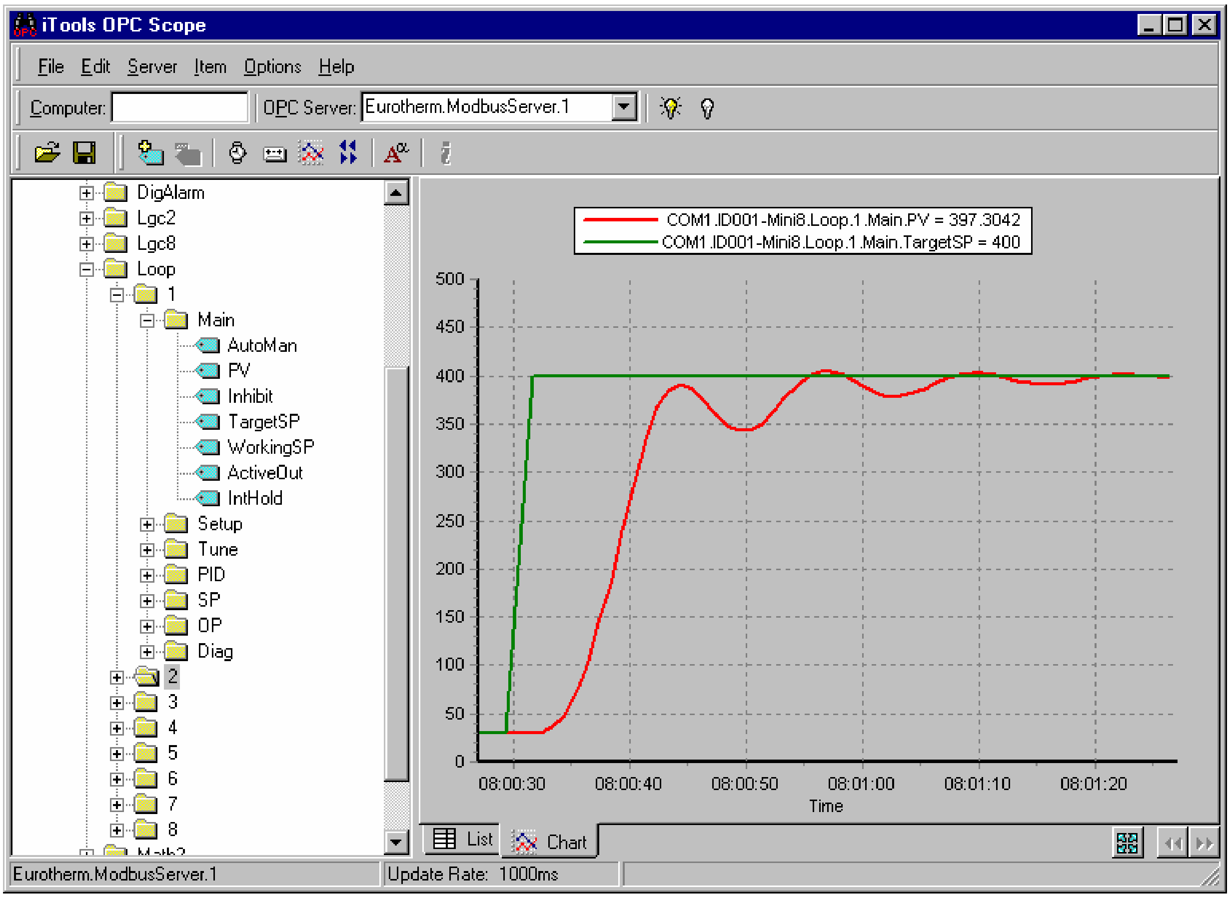
at the bottom of the display window and select Chart Control Panel.

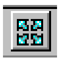


1. **Items.** Includes all the items in the list window. Those items ticked (up to 8) will appear on the chart.
2. **Axes.** Allows time intervals from 1 minute to 1 month. Vertical axes can be 'auto' scaled or a fixed range may be entered.
3. **General.** Allows selection of colours, grid, legends and a data box.
4. **Plot.** Allows selection of line thickness and printing
5. **Review.** Allows review of early history charts.

These are also available on the toolbar.

iTools Trend Graph showing Loop1 SP and PV



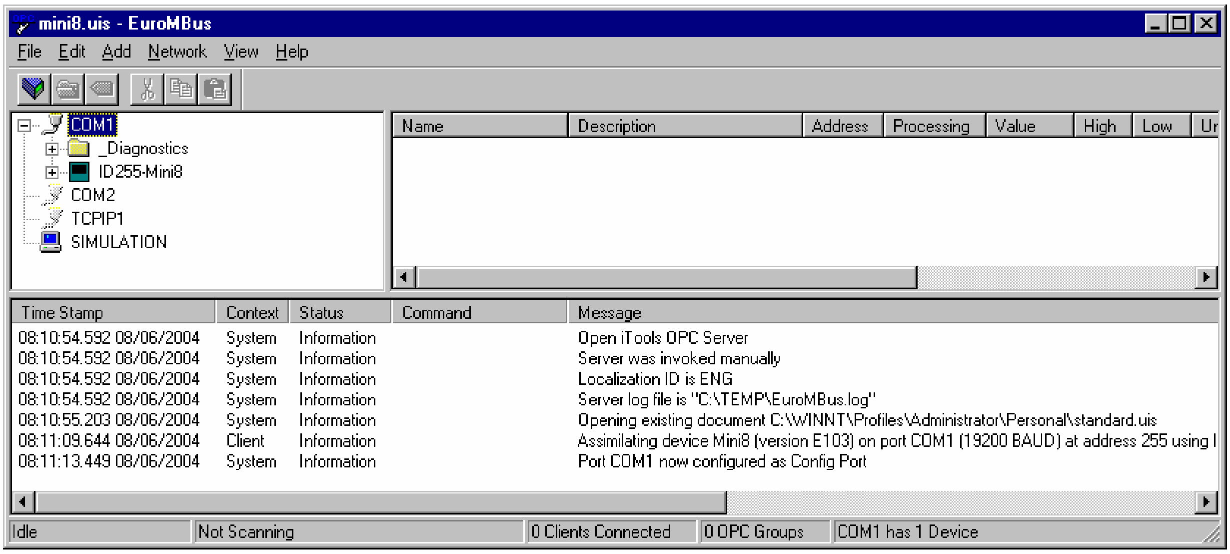
The  icon allows the chart to occupy all the window space.

2.8.3 OPC Server

ITools and OPC Scope all use the Eurotherm OPC Server to provide the connection between the instruments and the computer displays. When you 'scan' for instruments on iTools it is in fact the OPC Server that is actually doing the work in background (the window is not usually displayed).

OPC Scope can run on its own but for it to find the instruments on the network it is necessary to tell the server where they are.

1. Start OPC Server (Windows Start/Programs/Eurotherm iTools/OPC Server)
2. On the menu go to 'Network' and select 'Start One-Shot Scan'
3. Stop the scan when all the instruments have been found.



4. On the menu go to 'File' and select 'Save As' and save the file with a suitable name.
5. Once saved you will be asked 'Would you like to make this file the default start server address file?' – select 'Yes'.
6. Close the server.

Now if you double click on an OPC Scope file e.g. Mini8 Project.uix then this file will open OPC Scope and in turn, in background, OPC scope will open the OPC Server with this instrument file loaded. OPC Scope will then be active with live data from the instrument(s).

3. CHAPTER 3 CONFIGURATION USING iTOOLS

WARNING

Configuration level gives access to a wide range of parameters that match the controller to the process. Incorrect configuration could result in damage to the process being controlled and/or personal injury. It is the responsibility of the person commissioning the process to ensure that the configuration is correct.

In configuration level the controller is not controlling the process or providing alarm indication. Do not select configuration level on a live process.

3.1 Configuration

The Mini8 is supplied unconfigured, unless ordered preconfigured, e.g. EC8. An unconfigured Mini8 has to be configured for use in an application. This is performed using iTools.

The iTools handbook, part no. HA026179 provides further step by step instructions on the general operation of iTools. This and the iTools software may be downloaded from www.eurotherm.co.uk.

3.1.1 On-Line/Off-line Configuration

If iTools is connected to a real Mini8 then all the parameter changes made will be written to the device immediately. Once the Mini8 is configured and working as required, its final configuration can be saved to disk as a 'clone' file of the format <name>.uic.

Alternatively iTools can be used 'off-line' without a real Mini8 connected at all. This virtual Mini8 can be created in iTools and again saved to disk as a clone file. This file can later be loaded into a real Mini8 to create the required real application.

3.2 Connecting a PC to the Mini8 Controller

3.2.1 Configuration Cable and Clip

The controller may be connected to the PC running iTools using the Eurotherm cable **SubMin8/Cable/Config** from the RJ11 port connecting to a serial port on the PC.

Alternatively a Configuration Clip is available from Eurotherm that can be fitted into the rear of the controller.

The benefit of using this arrangement is that it is not necessary to power the controller, since the clip provides the power to the internal memory of the controller.

3.2.2 Scanning

Open iTools and, with the controller connected, press 

on the iTools menu bar. iTools will search the communications ports and TCPIP connections for recognisable instruments. Controllers connected using the RJ11 configuration port or with the configuration clip (CPI), will be found at address 255 regardless of the address configured in the controller. These connections only work from iTools to a single controller.

The iTools handbook, part no. HA026179, provides further step by step instructions on the general operation of iTools. This and the iTools software may be downloaded from www.eurotherm.co.uk.

In the following pages it is assumed that the user is familiar with iTools and has a general understanding of Windows.

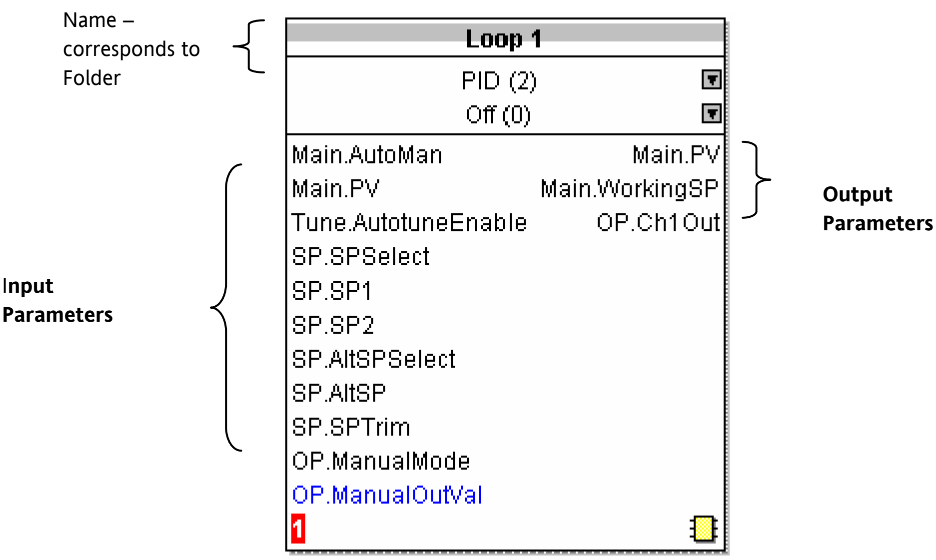
3.3 Configuring the Mini8

Once iTools is successfully connected to a Mini8, it can be configured for the application in hand. Configuration involves selection of the required elements of functionality called 'function blocks' and setting their parameters to the correct values. The next stage is to connect all the function blocks together to create the required strategy of control for the application.

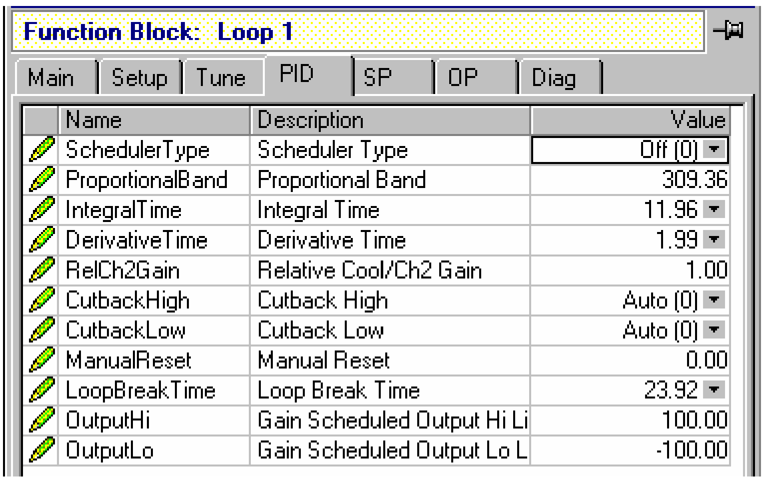
3.3.1 Function Blocks

The controller software is constructed from a number of 'function blocks'. A function block is a software device that performs a particular duty within the controller. It may be represented as a 'box' that takes data in at one side (as inputs), manipulates the data internally (using internal parameter values) and 'outputs' the results. Some of these internal parameters are available to the user so that they can be adjusted to suit the characteristics of the process that is to be controlled.

A representation of a function block is shown below.



Internal Parameters



Function Block: Loop 1

Main | Setup | Tune | PID | SP | OP | Diag

Name	Description	Value
SchedulerType	Scheduler Type	Off (0)
ProportionalBand	Proportional Band	309.36
IntegralTime	Integral Time	11.96
DerivativeTime	Derivative Time	1.99
RelCh2Gain	Relative Cool/Ch2 Gain	1.00
CutbackHigh	Cutback High	Auto (0)
CutbackLow	Cutback Low	Auto (0)
ManualReset	Manual Reset	0.00
LoopBreakTime	Loop Break Time	23.92
OutputHi	Gain Scheduled Output Hi Li	100.00
OutputLo	Gain Scheduled Output Lo L	-100.00

Figure 3-1: Example of a Function Block

In the controller, parameters are organised in simple lists. The top of the list shows the list header. This corresponds to the name of the function block and is generally presented in alphabetical order. This name describes the generic function of the parameters within the list. For example, the list header 'AnAlm' contains parameters that enable you to set up analogue alarm conditions.

3.3.2 Soft Wiring

Soft Wiring (sometimes known as User Wiring) refers to the connections that are made in software between function blocks. Soft wiring, which will generally be referred to as 'Wiring' from now on is created during the instrument configuration using the iTools configuration package.

In general every function block has at least one input and one output. Input parameters are used to specify where a function block reads its incoming data (the 'Input Source'). The input source is usually wired to the output from a preceding function block. Output parameters are usually wired to the input source of subsequent function blocks.

All parameters shown in the function block diagrams are also shown in the parameter tables, in the relevant chapters, in the order in which they appear in iTools.

Figure 3.2 shows an example of how the thermocouple is wired to the PID Loop input and the PID Loop channel 1 (heat) output is wired to the time proportioning logic output.

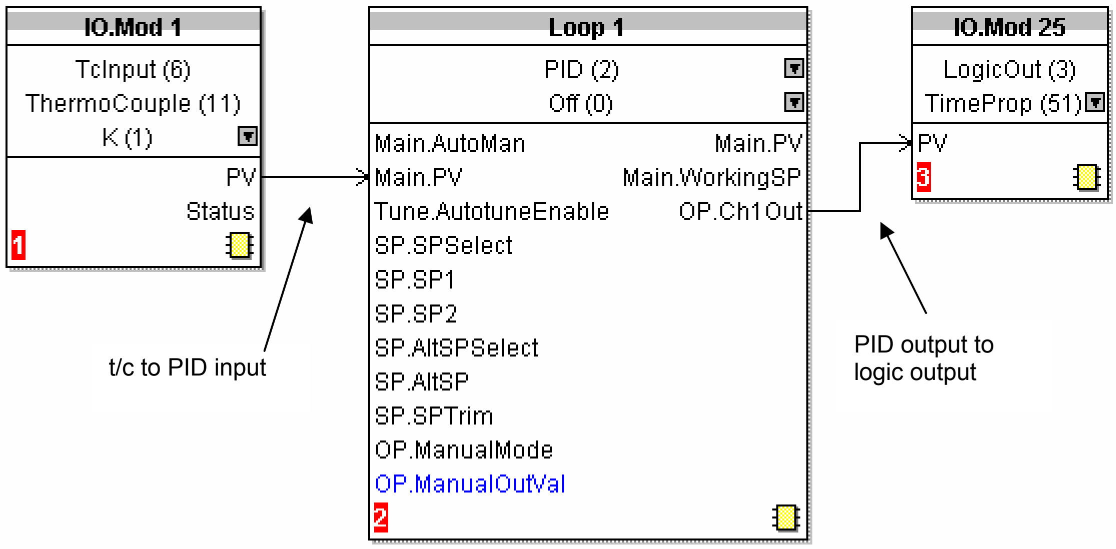


Figure 3-2: Function Block Wiring

3.4 Simple Worked Example

Using function blocks and wiring the following sections will show a blank Mini8 being configured to have one PID loop.

3.4.1 The I/O

With the Mini8 successfully connected to iTools configuration can begin.

☺ Tip: In parameter lists:

Parameters in BLUE are read only

Parameters in BLACK are read/write.

☺ Tip: Every parameter in the parameter lists has a detailed description in the help file – just click on a parameter and hit Shift-F1 on the keyboard or right click and select parameter help.

The I/O will already have been installed in the Mini8 and can be checked in iTools.

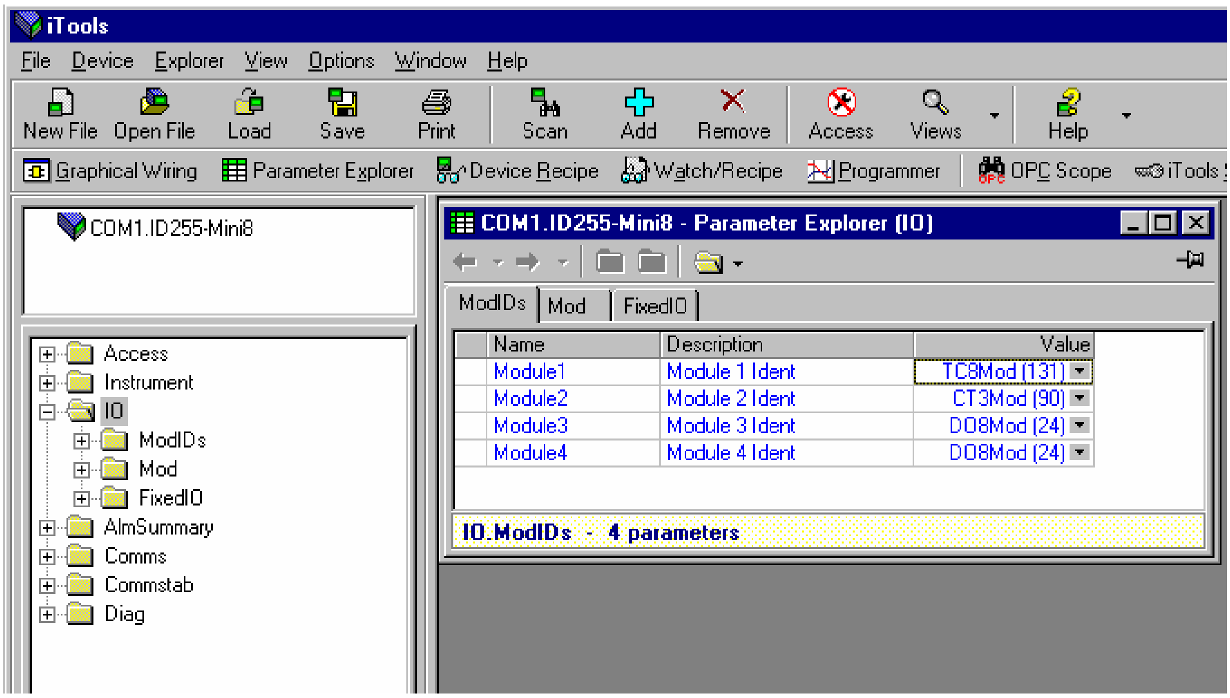


Figure 3-3: Mini8 I/O Modules

This unit has an 8 thermocouple input board in slot 1, a CT3 input card in slot 2, and 2 DO8 output cards in slot 3 and slot 4.. Clicking on the 'Mod' tab will enable the first channel of the thermocouple card to be configured. Firstly the Mini8 has to be put into configuration mode. Go to Device/Access/Configuration or click on the Access button:

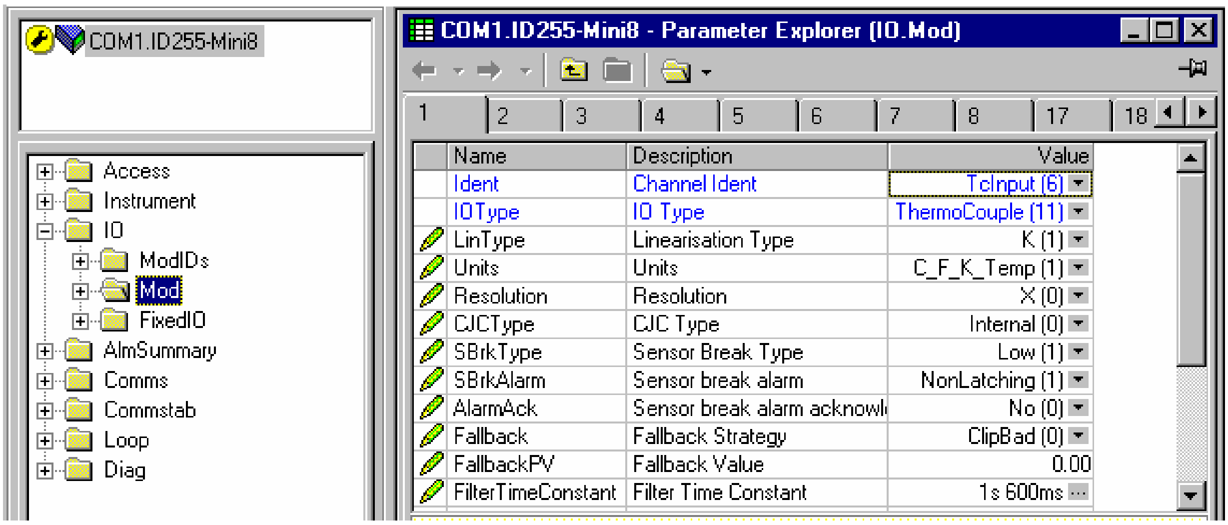


Figure 3-4: Thermocouple Input

Select the I/O type, linearisation, units, resolution etc. required. Parameter details are in Section 7.3.

The other thermocouple channels can be found by using the 2,3,4...7,8 tabs on the top of the parameter window.

Slot 2 in the Mini8 has a CT3 input card and this is configured elsewhere so the Tabs 9 to 16 are not shown.

Slot 3 has a DO8 output card and the first channel of this will be on tab 17 (to 24)

Slot 4 has a DO8 output card and the first channel of this will be on tab 25 (to 32)

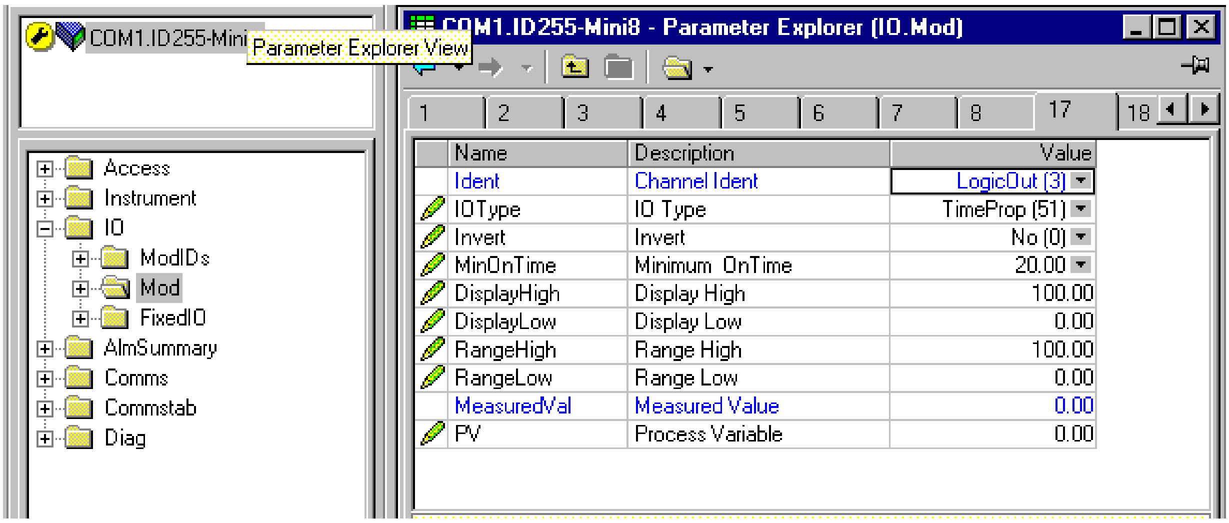


Figure 3-5: Digital Output Channel

Set this channel up as required, IOType, MinOnTime etc. as required. The parameters are detailed in Section 7.2.

The remaining channels on this slot will be found under the tabs 18 to 24.

Slot 4 also contains a DO8 output card with outputs under tabs 25 to 32.

The fixed I/O is always there and there is nothing that has to be configured.

The Current Monitor is covered in Chapter 7.6.

3.4.2 Wiring

The IO that has been configured now needs to be wired to PID loops and other function blocks.

Select 

(GWE) to create and edit instrument wiring.

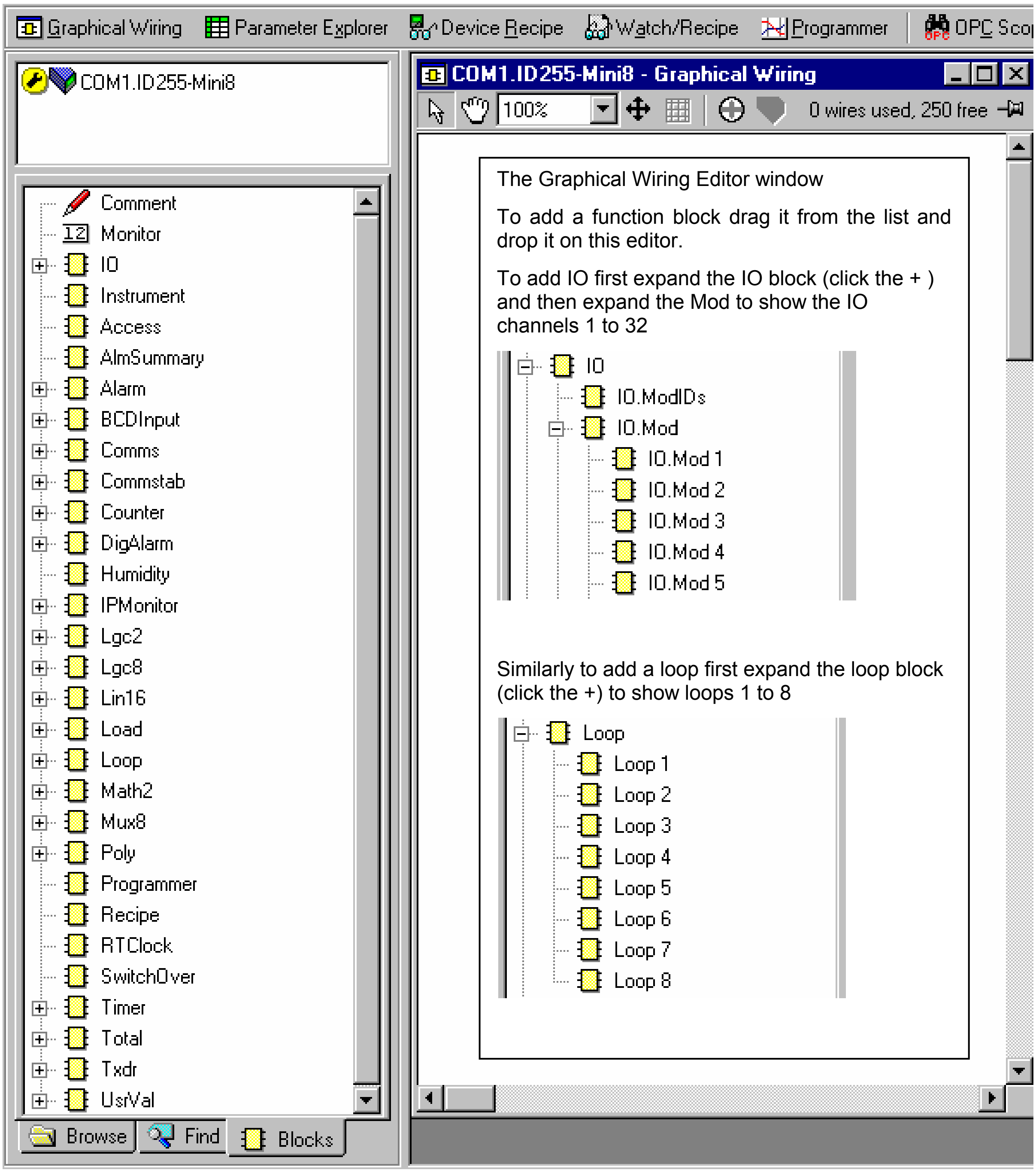


Figure 3-6: List of Function Blocks & Graphical Wiring Window

The left window now contains a list of the function blocks available.

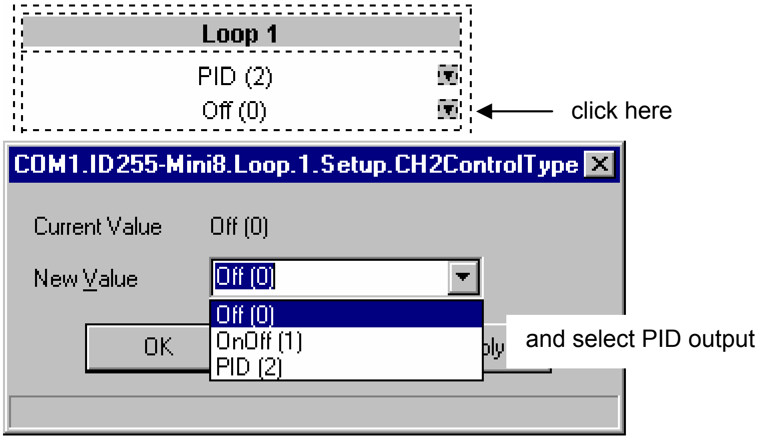
Use drag and drop to select the first thermocouple from IOMod 1, the Cool output from IOMod 17 and the Heat output from IOMod 25 and drop them on the wiring window.

Finally take the first PID block from Loop/Loop 1 and drop it on the wiring window. Note that as each block is used it greys out on the list.

There should now be 4 blocks on the window. Those blocks are shown with dotted lines, as they have not been loaded into the Mini8.

First make the following wire connections.

1. Click on IO.Mod1.PV and move the pointer to Loop 1.MainPV and click again. A dotted wire will have connected the two together.
2. Similarly join Loop1.OP.Ch1Out to IOMod 25.PV (heat output)
3. Enable the Cool output by clicking the select arrow to the top of the loop block:



4. Loop1.OP.Ch2Out to IOMod 17.PV (cool output)

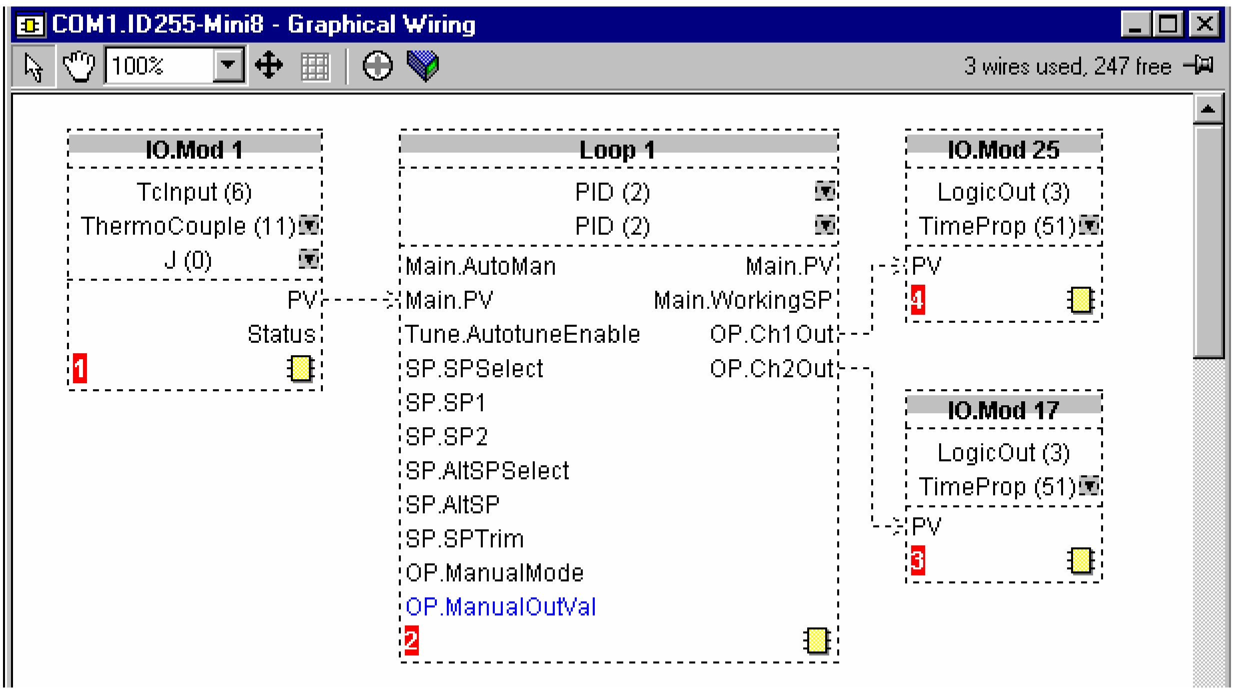


Figure 3-7: Wired Blocks before download

5. Right click on the Loop 1 function Block and select 'Function Block View'. This opens the Loop parameter list on top of the wiring editor.

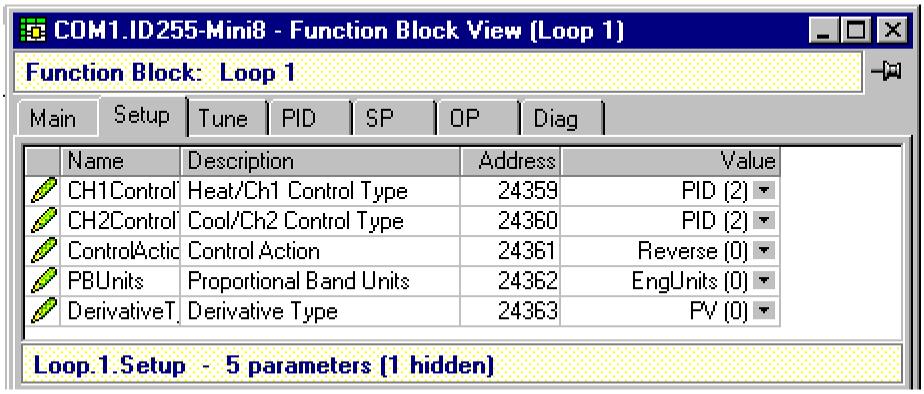


Figure 3-8: PID Function Block

This enables the PID function block to be set up to suit the required application. See Chapter 17 for details.

6. Click on the instrument button to download the application:

7. Once downloaded the dotted lines around the function blocks and the wires will become solid to show that the application is now in the Mini8. The upper status line also shows that 3 of wires have been used out of those available. Max is 250 but quantity depends on number of wires ordered (30, 60, 120 or 250).

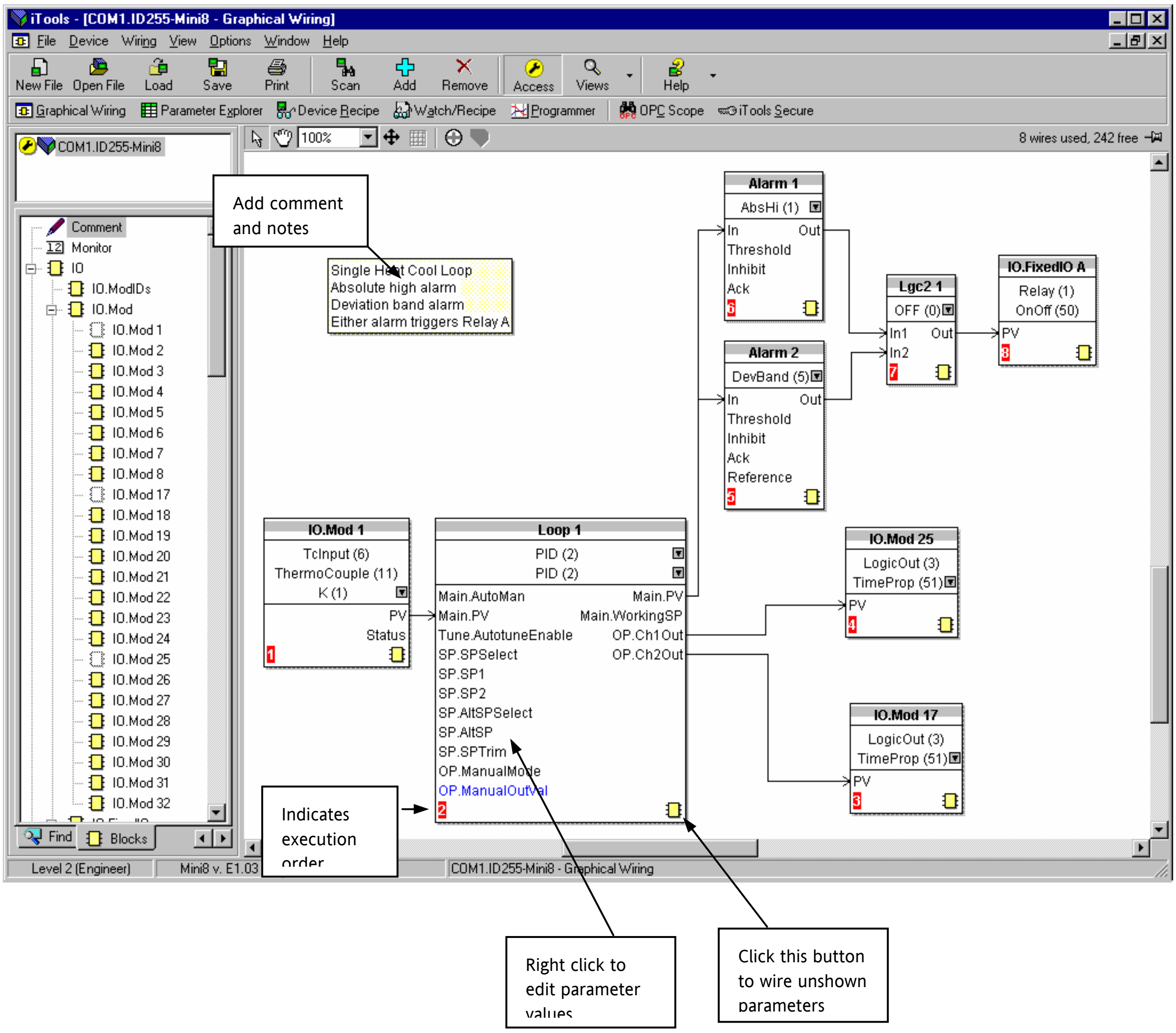
8. Put the Mini8 back into Operating mode by clicking the Access button:

9. The Mini8 will now control the Loop1 as configured.

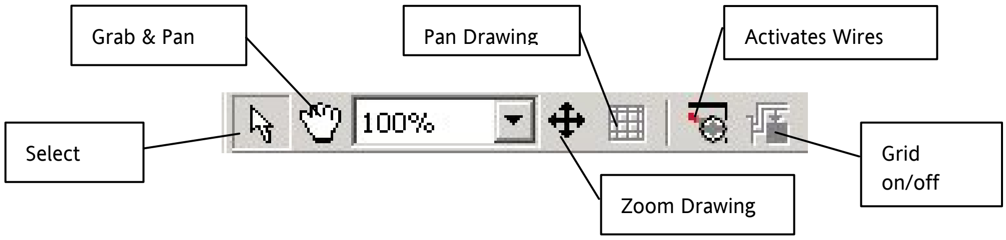
3.5 Graphical Wiring Editor

Select Graphical Wiring (GWE) to view and edit instrument wiring. You can also add comments and monitor parameter values.

1. Drag and drop required function blocks into the graphical wiring from the list in the left pane
2. Click on parameter to be wired from and drag the wire to the parameter to be wired to (do not hold mouse button down)
3. Right click to edit parameter values
4. Select parameter lists and switch between parameter and wiring editors
5. Download to instrument when wiring completed
6. Add comments and notes
7. Dotted lines around a function block show that the application requires downloading



3.5.1 Graphical Wiring Toolbar



3.5.2 Function Block

A Function Block is an algorithm that may be wired to and from other function blocks to make a control strategy. The Graphical Wiring Editor groups the instrument parameters into function blocks. Examples are: a control loop and a mathematical calculation.
Each function block has inputs and outputs. Any parameter may be wired from, but only parameters that are alterable may we wired to.
A function block includes any parameters that are needed to configure or operate the algorithm.

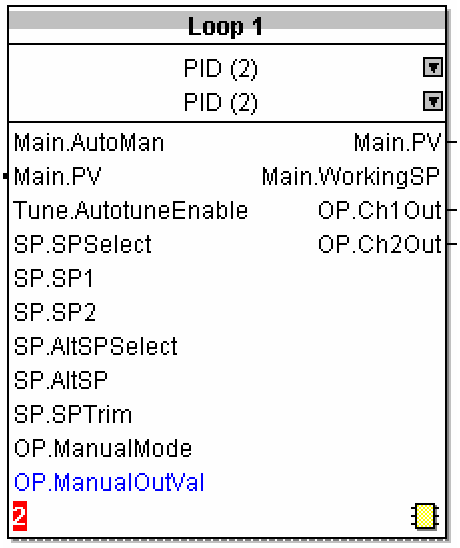
3.5.3 Wire

A wire transfers a value from one parameter to another. They are executed by the instrument once per control cycle.
Wires are made from an output of a function block to an input of a function block. It is possible to create a wiring loop, in this case there will be a single execution cycle delay at some point in the loop. This point is shown on the diagram By a | | symbol and it is possible to choose where that delay will occur.

3.5.4 Block Execution Order

The order in which the blocks are executed by the instrument depends on the way in which they are wired. The order is automatically worked out so that the blocks execute on the most recent data.

3.5.5 Using Function Blocks



If a function block is not faded in the tree then it can be dragged onto the diagram. The block can be dragged around the diagram using the mouse.

A labelled loop block is shown here. The label at the top is the name of the block.

When the block type information is alterable click on the box with the arrow in it on the right to edit that value.

The inputs and outputs that are considered to be of most use are always shown. In most cases all of these will need to be wired up for the block to perform a useful task. There are exceptions to this and the loop is one of those exceptions.

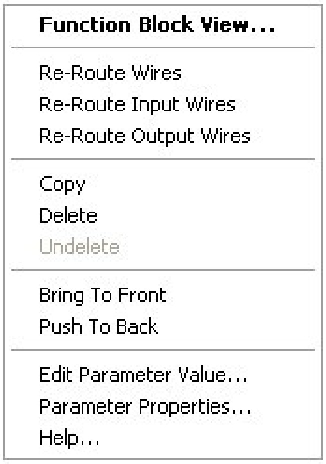
If you wish to wire from a parameter, which is not shown as a recommended output click on the icon in the bottom right, and a full list of parameters in the block will be shown, click on one of these to start a wire.

To start a wire from a recommended output just click on it.

Click the icon in the bottom right hand corner to wire other function block parameters not shown on the list on the right hand side.

3.5.5.1 Function Block Context Menu

Right clicking displays the context menu with the following entries.



Function Block View…	Brings up an iTools parameter list which shows all the parameters in the function block. If the block has sub-lists these are shown in tabs
Re-Route Wires	Throw away current wire route and do an auto-route of all wires connected to this block
Re-Route Input Wires	Only do a re-route on the input wires
Re-Route Output Wires	Only do a re-route on the output wires
Copy	Right click over an input or output and copy will be enabled, this menu item will copy the iTools "url" of the parameter which can then be pasted into a watch window or OPC Scope
Delete	If the block is downloaded mark it for delete, otherwise delete it immediately
Undelete	This menu entry is enabled if the block is marked for delete and unmarks it and any wires connected to it for delete
Bring To Front	Bring the block to the front of the diagram. Moving a block will also bring it to the front
Push To Back	Push the block to the back of the diagram. Useful of there is something underneath it
Edit Parameter Value	This menu entry is enabled when the mouse is over an input or output parameter. When selected it creates a parameter edit dialog so the value of that parameter can be changed
Parameter Properties	Selecting this entry brings up the parameter properties window. The parameter properties window is updated as the mouse is moved over the parameters shown on the function block
Help	Selecting this entry brings up the help window. The help window is updated as the mouse is moved over the parameters shown on the function block. When the mouse is not over a parameter name the help for the block is shown

3.5.6 Tooltips

Hovering over different parts of the block will bring up tooltips describing the part of the block beneath the mouse.
If you hover over the parameter values in the block type information a tooltip showing the parameter description, its OPC name, and, if downloaded, its value will be shown.

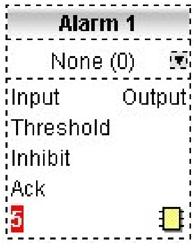
A similar tooltip will be shown when hovering over inputs and outputs.

3.5.7 Function Block State

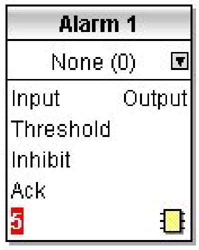
The blocks are enabled by dragging the block onto the diagram, wiring it up, and downloading it to the instrument

When the block is initially dropped onto the diagram it is drawn with dashed lines.

When in this state the parameter list for the block is enabled but the block itself is not executed by the instrument.



Once the download button is pressed the block is added to the instrument function block execution list and it is drawn with solid lines.



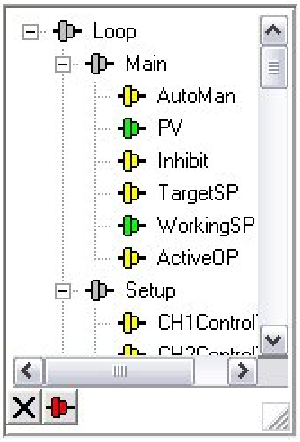
If a block which has been downloaded is deleted, it is shown on the diagram in a ghosted form until the download button is pressed.

This is because it and any wires to/from it are still being executed in the instrument. On download it will be removed from the instrument execution list and the diagram. A ghosted block can be undeleted using the context menu.

When a dashed block is deleted it is removed immediately.

3.5.8 Using Wires

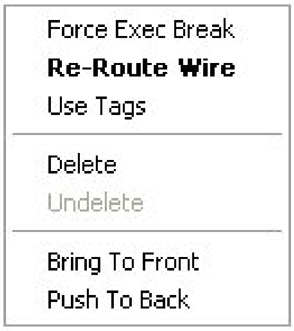
3.5.8.1 Making A Wire Between Two Blocks



- Drag two blocks onto the diagram from the function block tree.
- Start a wire by either clicking on a recommended output or clicking on the icon at the bottom right corner of the block to bring up the connection dialog. The connection dialog shows all the connectable parameters for the block, if the block has sub-lists the parameters are shown in a tree. If you wish to wire a parameter which is not currently available click the red button at the bottom of the connection dialog. Recommended connections are shown with a green plug, other parameters which are available are yellow and if you click the red button the unavailable parameters are shown red. To dismiss the connection dialog either press the escape key on the keyboard or click the cross at the bottom left of the dialog.
- Once the wire has started the cursor will change and a dotted wire will be drawn from the output to the current mouse position.
- To make the wire either click on a recommended input to make a wire to that parameter or click anywhere except on a recommended input to bring up the connection dialog. Choose from the connection dialog as described above.
- The wire will now be auto-routed between the blocks.

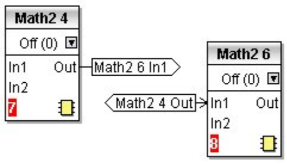
New wires are shown dotted until they are downloaded

3.5.8.2 Wire Context Menu



The wire block context menu has the following entries on it.

Force Exec Break	If wires form a loop a break point has to be found where the value which is written to the block input comes from a block which was last executed during the previous instrument execute cycle thus introducing a delay. This option tells the instrument that if it needs to make a break it should be on this wire
Re-Route Wire	Throw away wire route and generate an automatic route from scratch
Use Tags	If a wire is between blocks which are a long way apart, then, rather than drawing the wire, the name of the wired to/from parameter can be shown in a tag next to the block. Draw the wire first then use this menu to toggle this wire between drawing the whole wire and drawing it as tags
Delete	If the wire is downloaded mark it for delete, otherwise delete it immediately
Undelete	This menu entry is enabled if the wire is marked for delete and unmarks it for delete
Bring To Front	Bring the wire to the front of the diagram. Moving a wire will also bring it to the front
Push To Back	Push the wire to the back of the diagram



3.5.8.3 Wire Colours

Wires can be the following colours:

Black	Normal functioning wire.
Red	The wire is connected to an input which is not alterable when the instrument is in operator mode and so values which travel along that wire will be rejected by the receiving block
Blue	The mouse is hovering over the wire, or the block to which it is connected it selected. Useful for tracing densely packed wires
Purple	The mouse is hovering over a 'red' wire

3.5.8.4 Routing Wires

When a wire is placed it is auto-routed. The auto routing algorithm searches for a clear path between the two blocks. A wire can be auto-routed again using the context menus or by double clicking the wire.

If you click on a wire segment you can drag it to manually route it. Once you have done this it is marked as a manually routed wire and will retain it's current shape. If you move the block to which it is connected the end of the wire will be moved but as much of the path as possible of the wire will be preserved.

If you select a wire by clicking on it, it will be drawn with small boxes on it's corners.

3.5.8.5 Tooltips

Hover the mouse over a wire and a tooltip showing the names of the parameters which are wired and, if downloaded, their current values will also be shown.

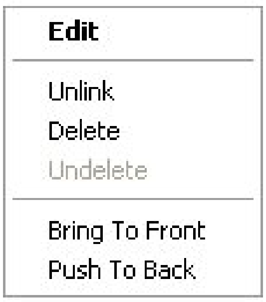
3.5.9 Using Comments

Drag a comment onto the diagram and the comment edit dialog will appear.

Type in a comment. Use new lines to control the width of the comment, it is shown on the diagram as typed into the dialog. Click OK and the comment text will appear on the diagram. There are no restrictions on the size of a comment. Comments are saved to the instrument along with the diagram layout information.

Comments can be linked to function blocks and wires. Hover the mouse over the bottom right of the comment and a chain icon will appear, click on that icon and then on a block or a wire. A dotted wire will be drawn to the top of the block or the selected wire segment.

3.5.9.1 Comment Context Menu



The comment context menu has the following entries on it.

Edit	Open the comment edit dialog to edit this comment
Unlink	If the comment is linked to a block or wire this will unlink it
Delete	If the comment is downloaded mark it for delete, otherwise delete it immediately
Undelete	This menu entry is enabled if the comment is marked for delete and unmarks it for delete
Bring To Front	Bring the comment to the front of the diagram. Moving a comment will also bring it to the front
Push To Back	Push the comment to the back of the diagram. Useful if there is something underneath it

3.5.10 Using Monitors

Drag a monitor onto the diagram and connect it to a block input or output or a wire as described in 'Using Comments'.

The current value (updated at the iTools parameter list update rate) will be shown in the monitor. By default the name of the parameter is shown, double click or use the context menu to not show the parameter name.

3.5.10.1 Monitor Context Menu

The monitor context menu has the following entries on it.

Show Names	Show parameter names as well as values
Unlink	If the monitor is linked to a block or wire this will unlink it
Delete	If the monitor is downloaded mark it for delete, otherwise delete it immediately
Undelete	This menu entry is enabled if the monitor is marked for delete and unmarks it for delete
Bring To Front	Bring the monitor to the front of the diagram. Moving a monitor will also bring it to the front
Push To Back	Push the monitor to the back of the diagram. Useful if there is something underneath it

3.5.11 Downloading

The wires have to be downloaded to the instrument together. When the wiring editor is opened the current wiring and diagram layout is read from the instrument. No changes are made to the instrument function block execution or wiring until the download button is pressed. Any changes made using the instrument front panel after the editor is opened will be lost on download.

When a block is dropped on the diagram instrument parameters are changed to make the parameters for that block available. If you make changes and close the editor without saving them there will be a delay while the editor clears these parameters.

When you download, the wiring is written to the instrument that then calculates the block execution order and starts executing the blocks. The diagram layout including comments and monitors is then written into instrument flash memory along with the current editor settings. When you reopen the editor the diagram will be shown positioned the same as when you last downloaded.

3.5.12 Selections

Wires are shown with small blocks at their corners when selected. All other items have a dotted line drawn round them when they are selected.

3.5.12.1 Selecting Individual Items

Clicking on an item on the drawing will select it.

3.5.12.2 Multiple Selection

Control click an unselected item to add it to the selection, doing the same on a selected item unselects it.

Alternatively, hold the mouse down on the background and wipe it to create a rubber band, anything which isn't a wire inside the rubber band will be selected.

Selecting two function blocks also selects any wires which join them. This means that if you select more than one function block using the rubber band method any wires between them will also be selected.

Pressing Ctrl-A selects all blocks and wires.

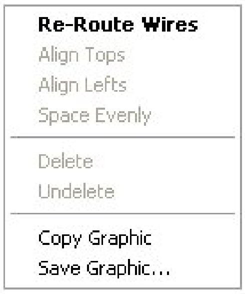
3.5.13 Colours

Items on the diagram are coloured as follows:

Red	Function blocks, comments and monitors which partially obscure or are partially obscured by other items are drawn red. If a large function block like the loop is covering a small one like a math2 the loop will be drawn red to show that it is covering another function block. Wires are drawn red when they are connected to an input which is currently unalterable. Parameters in function blocks are coloured red if they are unalterable and the mouse pointer is over them
Blue	Function blocks, comments and monitors which are not coloured red are coloured blue when the mouse pointer is over them. Wires are coloured blue when a block to which the wire is connected is selected or the mouse pointer is over it. Parameters in function blocks are coloured blue if they are alterable and the mouse pointer is over them
Purple	A wire which is connected to an input which is currently unalterable and a block to which the wire is connected is selected or the mouse pointer is over it is coloured purple (red + blue)

3.6 Diagram Context Menu

The diagram context menu has the following entries on it:-



Re-Route Wires	Throw away current wire route and do an auto-route of all selected wires. If no wires are selected this is done to all wires on the diagram
Align Tops	Line up the tops of all the selected items except wires
Align Lefts	Line up the left hand side of all the selected items except wires
Space Evenly	This will space the selected items such that their top left corners are evenly spaced. Select the first item, then select the rest by control-clicking them in the order you wish them to be spaced, then choose this menu entry
Delete	Delete, or mark for delete (series 3000 instruments) all selected items
Undelete	This menu entry is enabled if any of the selected items are marked for delete and unmarks them when selected
Copy Graphic	If there is a selection it is copied to the clipboard as a Windows metafile, if there is no selection the whole diagram is copied to the clipboard as a Windows metafile. Paste into your favourite documentation tool to document your application. Some programs render metafiles better than others, the diagram may look messy on screen but it should print well
Save Graphic	Same as Copy Graphic but saves to a metafile rather than putting it on the clipboard

3.6.1 Wiring Floats with Status Information

There is a subset of float values which may be derived from an input which may become faulty for some reason, e.g. sensor break, overrange, etc. These values have been provided with an associated status which is automatically inherited through the wiring. The list of parameters which have associated status is as follows:-

Block	Input Parameters	Output Parameters
Loop.Main	PV	PV
Loop.SP		TrackPV
Math2	In1	Out
	In2	
Programmer.Setup	PVIn	
Poly	In	Out
Load		PVOut1
		PVOut2
Lin16	In	Out
Txdr	InVal	OutVal
IPMonitor	In	Out
SwitchOver	In1	
	In2	
Total	In	
Mux8	In1 to 8	Out
Lgc2	In1	
	In2	
UsrVal	Val	Val
Humidity	WetTemp	RelHumid
	DryTemp	DewPoint
	PsychroConst	
	Pressure	
IO.MOD	1.PV to 32.PV	1.PV to 32.PV

Parameters appear in both lists where they can be used as inputs or outputs depending on configuration. The action of the block on detection of a 'Bad' input is dependent upon the block. For example, the loop treats a 'Bad' input as a sensor break and takes appropriate action; the Mux8 simply passes on the status from the selected input to the output, etc.

The Poly, Lin16, SwitchOver, Mux8, IO.Mod.n.PV blocks can be configured to act on bad status in varying ways. The options available are as follows:-

0: Clip Bad

The measurement is clipped to the limit it has exceeded and its status is set to 'BAD', such that any function block using this measurement can operate its own fallback strategy. For example, control loop may hold its output to the current value.

1: Clip Good

The measurement is clipped to the limit it has exceeded and its status is set to 'GOOD', such that any function block using this measurement may continue to calculate and not employ its own fallback strategy.

2: Fallback Bad

The measurement will adopt the configured fallback value that has been set by the user. In addition the status of the measured value will be set to 'BAD', such that any function block using this measurement can operate its own fallback strategy. For example, control loop may hold its output to the current value.

3: Fallback Good

The measurement will adopt the configured fallback value that has been set by the user. In addition the status of the measured value will be set to 'GOOD', such that any function block using this measurement may continue to calculate and not employ its own fallback strategy

4: Up Scale

The measurement will be forced to adopt its high limit. This is like having a resistive pull up on an input circuit. In addition the status of the measured value will be set to 'BAD', such that any function block using this measurement can operate its own fallback strategy. For example, the control loop may hold its output to the current value.

5: Down Scale

The measurement will be forced to adopt its low limit. This is like having a resistive pull down on an input circuit. In addition the status of the measured value will be set to 'BAD', such that any function block using this measurement can operate its own fallback strategy. For example, the control loop may hold its output to the current value.

3.6.2 Edge Wires

If the Loop.Main.AutoMan parameter were wired from a logic input in the conventional manner it would be impossible to put the instrument into manual via communications. Other parameters need to be controlled by wiring but also need to be able to change under other circumstances, e.g. Alarm Acknowledgements. for this reason some Boolean parameters are wired in an alternative way. These are listed as follows:-

SET DOMINANT

When the wired in value is 1 the parameter is always updated. This will have the effect of overriding any changes through digital communications. When the wired in value changes to 0 the parameter is initially changed to 0 but is not continuously updated. This permits the value to be changed through digital communications.

Loop.Main.AutoMan	Programmer.Setup.ProgHold	Access.StandBy

RISING EDGE

When the wired in value changes from 0 to 1, a 1 is written to the parameter. At all other times the wire does not update the parameter. This type of wiring is used for parameters that start an action and when once completed the block clears the parameter. When wired to, these parameters can still be operated via digital communications.

Loop.Tune.AutotuneEnable	Txdr.ClearCal	Alarm.Ack
	Txdr.StartCal	DigAlarm.Ack
Programmer.Setup.ProgRun	Txdr.StartHighCal	AlmSummary.GlobalAck
Programmer.Setup.AdvSeg	Txdr.StartTare	
Programmer.Setup.SkipSeg		Instrument.Diagnostics. ClearStats
IPMonitor.Reset		

BOTH EDGE

This type of edge is used for parameters which may need to be controlled by wiring or but should also be able to be controlled through digital communications. If the wired in value changes then the new value is written to the parameter by the wire. At all other times the parameter is free to be edited through digital communications.

Loop.SP.RateDisable	Loop.OP.RateDisable

4. CHAPTER 4 MINI8 OVERVIEW

Input and output parameters of function blocks are wired together using software wiring to form a particular control strategy within the Mini8. An overview of all the available functions and where to get more detail is shown below.

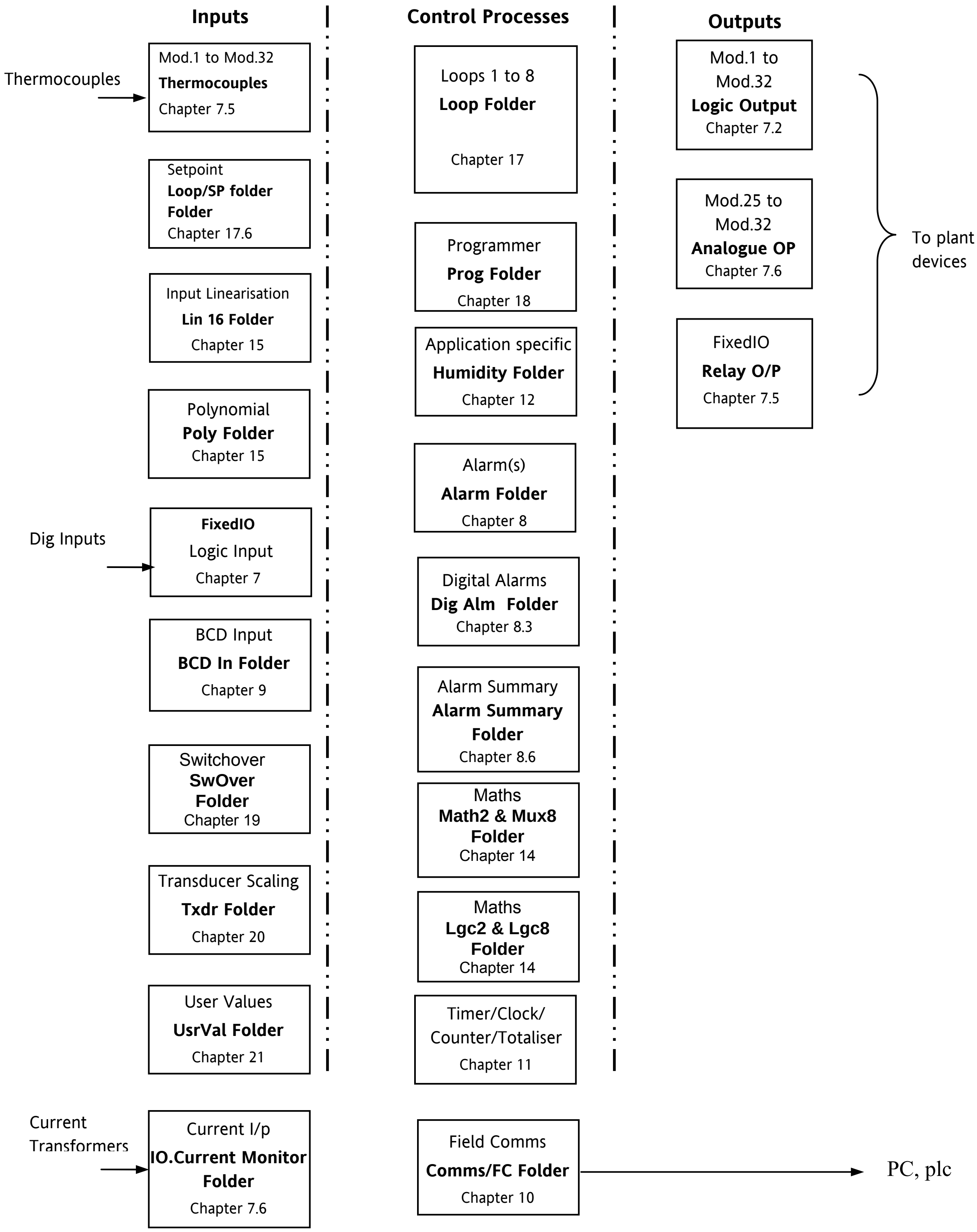


Figure 4-1: Controller Example

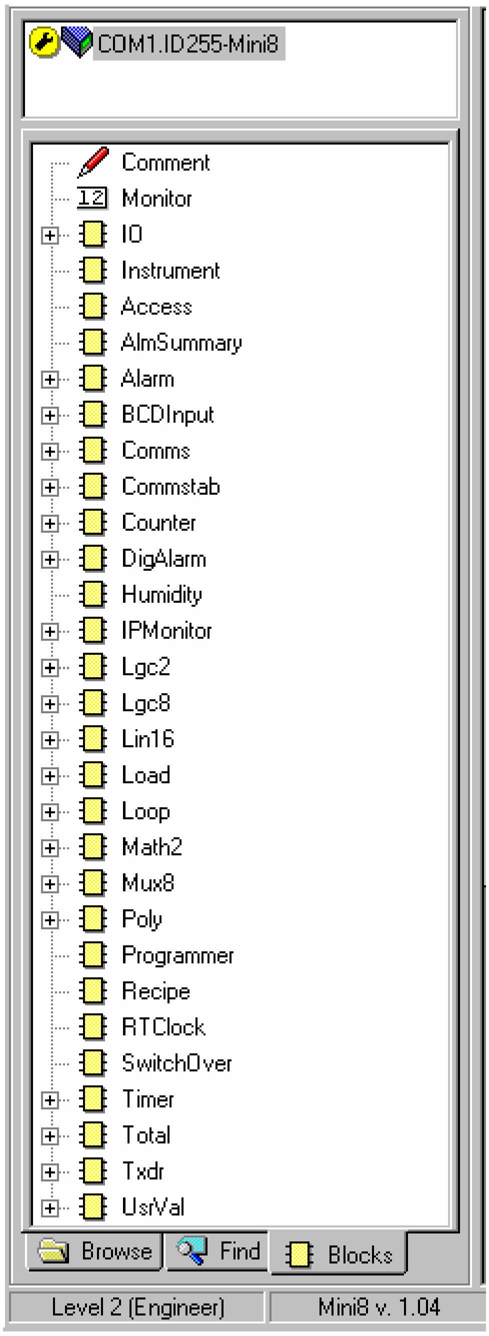
Mini8 series controllers are supplied unconfigured, and with those blocks included in the order code. Option EC8 is supplied with function blocks pre-wired to give an 8 loop heat/cool controller suitable for Extrusion. See data sheet HA028519.

The purpose of the PID control blocks is to reduce the difference between SP and PV (the error signal) to zero by providing a compensating output to the plant via the output driver blocks.

The timer, programmer and alarms blocks may be made to operate on a number of parameters within the controller, and digital communications provides an interface to data collection and control.

The controller can be customised to suit a particular process by 'soft wiring' between function blocks.

4.1 Complete list of Function Blocks.



The list opposite represents an unconfigured Mini8 that has been ordered with all features enabled.

If a particular block or blocks do not appear in your instrument then the option has not been ordered. Check the order code of your instrument and contact Eurotherm.

Examples of features that may not have been enabled are:

Loops

Programmer

Recipe

Humidity

Once a block is dragged and dropped onto the graphical wiring window, the block icon in the block list opposite will be greyed out. At the same time a folder containing the blocks parameters will have been created in the Browse List.

5. CHAPTER 5 ACCESS FOLDER

Folder: Access		Sub-folder: none			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
ClearMemory	Cold start the instrument	No App LinTables InitComms Wires AllMemory Programs NewFware	Disabled Mini8 memory reset but comms and linearisation tables retained Custom Linearisation tables are deleted Comms ports reset to default configurations Clear all wiring All instrument memory is set to default values All Programs cleared Initialise all memory except Linearisation tables after firmware upgrade	No	Conf
CustomerID	Customer Identifier	Reference number for customer use		0	Oper
Standby	Set Instrument to standby	No / Yes		No	Oper

6. CHAPTER 6 INSTRUMENT FOLDER

6.1 Instrument / Options

The following table lists the options that can be enabled in the instrument.

Enable flags are one bit for each item – i.e.Bit (0=1) enables item 1, Bit 1 (=2) item 2, Bit(3=4) item 3 and so on to Bit7(=128) enables Item 8. All 8 items enabled adds up to 255.

☺ Tip: **Features are not normally enabled this way**. Dragging and dropping a function block onto the graphical wiring window automatically sets the required enable flag.

Folder: Instrument		**Sub-folder: Options**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Units	Units	°C,°F or Kelvin scale for all temperature parameters	DegC	Oper
AlarmEn1	Analogue alarms Enable Flags	Alarms 1 to 8 0 (none) to 255 (all 8)	0	Conf
AlarmEn2	Analogue alarms Enable Flags	Alarms 9 to 16 0 (none) to 255 (all 8)	0	Conf
AlarmEn3	Analogue alarms Enable Flags	Alarms 17 to 24 0 (none) to 255 (all 8)	0	Conf
AlarmEn4	Analogue alarms Enable Flags	Alarms 24 to 32 0 (none) to 255 (all 8)	0	Conf
BCDInEn	BCD switch input Enable Flags	BCD input 1 and 2 0 (none) to 3 (both)	0	Conf
CounterEn	Counters Enable Flags	Counters1 and 2 0 (none) to 3 (both)	0	Conf
CurrentMon	Current Monitor Enable Flag	0 = Off 1 = On	0	Conf
DigAlmEn1	Digital alarms Enable Flags	Dig Alarms 1 to 8 0 (none) to 255 (all 8)	0	Conf
DigAlmEn2	Digital alarms Enable Flags	Dig Alarms 9 to 16 0 (none) to 255 (all 8)	0	Conf
DigAlmEn3	Digital alarms Enable Flags	Dig Alarms 17 to 24 0 (none) to 255 (all 8)	0	Conf
DigAlmEn4	Digital alarms Enable Flags	Dig Alarms 24 to 32 0 (none) to 255 (all 8)	0	Conf
HumidityEn	Humidity control Enable Flag	0 = off 1 = On	0	Conf
IP Mon En	Input monitor Enable Flags	Input Monitor 1 and 2 0 (none) to 3 (both)	0	Conf
Lgc2 En1	Logic operators Enable Flags	Logic operators 1 to 8 0 (none) to 255 (all 8)	0	Conf
Lgc2 En2	Logic operators Enable Flags	Logic operators 9 to 16 0 (none) to 255 (all 8)	0	Conf
Lgc2 En3	Logic operators Enable Flags	Logic operators 17 to 24 0 (none) to 255 (all 8)	0	Conf
Lgc8 En	Logic 8 operator Enable Flags	8 input Logic operators 1 & 2 0 (none) to 3 (both)	0	Conf
Lin16Pt En	Input linearisation 16 point	Input Linearisation 1 and 2 0 (none) to 3 (both)	0	Conf
Load En	Load Enable Flags	Loads 1 to 8 0 (none) to 255 (all 8)	As order code	Conf
Loop En	Loop Enable Flags	Loops 1 to 8 0 (none) to 255 (all 8)	As order code	Conf
Math2 En1	Analogue (Maths) Operators Enable Flags	Analogue operators 1 to 8 0 (none) to 255 (all 8)	0	Conf
Math2 En2	Analogue (Maths) Operators Enable Flags	Analogue operators 9 to 16 0 (none) to 255 (all 8)	0	Conf
Math2 En3	Analogue (Maths) Operators Enable Flags	Analogue operators 17 to 24 0 (none) to 255 (all 8)	0	Conf
Mux8 En	Multiplexor Enable Flags	8 input multiplexor 1 and 2 0 (none) to 3 (both)	0	Conf
Poly En	Polynomial linearisation block Enable Flags	Poly Linearisation 1 and 2 0 (none) to 3 (both)	0	Conf
Progr En	Programmer Enable Flags	0 = off 1 = Enabled	0	Conf
RTClock En	Real time clock Enable Flags	0 = off 1 = On	0	Conf

Folder: Instrument		Sub-folder: Options		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
SwOver En	Switch over block Enable Flags	0 = off 1 = On	0	Conf
Timer En	Timers Enable Flags	Timers 1 to 4 0 = none to 15 = 4	0	Conf
Totalise En	Totalisers Enable Flags	Totalisers 1 & 2 0 (none) to 3 (both)	0	Conf
TrScale En	Transducer scaling Enable Flags	Transducer scalers 1 and 2 0 (none) to 3 (both)	0	Conf
UsrVal En1	User values Enable Flags	User Values 1 to 8 0 (none) to 255 (all 8)	0	Conf
UsrVal En2	User values Enable Flags	User Values 9 to 16 0 (none) to 255 (all 8)	0	Conf
UsrVal En3	User values Enable Flags	User Values 17 to 24 0 (none) to 255 (all 8)	0	Conf
UsrVal En4	User values Enable Flags	User Values 24 to 32 0 (none) to 255 (all 8)	0	Conf

6.2 Instrument / InstInfo

Folder: Instrument		Sub-folder: InstInfo		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
InstType	Instrument Type		MINI8	NONE
Version	Version Identifier		-	NONE
Serial No	Serial Number			NONE
Passcode1	Passcode1	0 to 65535		Oper
Passcode2	Passcode2	0 to 65535		Oper
Passcode3	Passcode3	0 to 65535		Oper
CompanyID	CompanyID		1280	NONE

6.3 Instrument / Diagnostics

This list provides fault finding diagnostic information as follows:-

<table>
<tr><th>Folder: Instrument</th><th colspan="2">Sub-folder: Diagnostics</th></tr>
<tr><th>Name</th><th colspan="2">Parameter Description</th></tr>
<tr><td>CPU % Free</td><td colspan="2">This is the amount of free CPU Time left. It shows the percentage of the tasks ticks that are idle.</td></tr>
<tr><td>CPU % Min</td><td colspan="2">A benchmark of the lowest reached value of the CPU free percentage.</td></tr>
<tr><td>Con Ticks</td><td colspan="2">This is the number of ticks that have elapsed while the instrument was performing the control Task.</td></tr>
<tr><td>Max Con Tick</td><td colspan="2">A benchmark of the maximum number of ticks that have elapsed while the instrument was performing the control Task</td></tr>
<tr><td>Clear Stats</td><td colspan="2">Resets the instrument performance benchmarks.</td></tr>
<tr><td>Error Count</td><td colspan="2">The number of errors logged since the last Clear Log. Note: If an error occurs multiple times only the first occurrence will be logged each event will increment the count.</td></tr>
<tr><td>Error1</td><td>The first error to occur</td><td rowspan="8">0 There is no error
1 Bad or unrecognised module ident. A module has been inserted and has a bad or unrecognised ident. Either the module is damaged or the module is unsupported.
3 Factory calibration data bad. The factory calibration data has been read from an I/O module and has not passed the checksum test. Either the module is damaged or has not been initialised.
4 Module changed for one of a different type. A module has been changed for one of a different type. The configuration may now be incorrect
10 Calibration data write error. An error has occurred when attempting to write calibration data back to an I/O module's EE.
11 Calibration data write error. An error occurred when trying to read calibration data back from the EE on an I/O module.
18 Checksum error. The checksum of the NVol Ram has failed. The NVol is considered corrupt and there the instrument configuration may be incorrect.
20 Resistive identifier error. An error occurred when reading the resistive identifier from an i/o module. The module may be damaged.
43 Invalid custom linearisation table. One of the custom linearisation tables is invalid. Either it has failed checksum tests or the table downloaded to the instrument is invalid.
55 The Instrument wiring is either invalid or corrupt.
56 Non Vol write to volatile
58 An attempt was made to perform a checksummed Non Vol write to a non checksummed address.
59 Bad User CT
60 Bad Factory CT
61 QS Error
62 to 65 Slot1 card DFC1 to DFC4 error
66 to 69 Slot2 card DFC1 to DFC4 error
70 to 73 Slot3 card DFC1 to DFC4 error
74 to 77 Slot4 card DFC1 to DFC4 error</td></tr>
<tr><td>Error2</td><td>The second error to occur</td></tr>
<tr><td>Error3</td><td>The third error to occur</td></tr>
<tr><td>Error4</td><td>The fourth error to occur</td></tr>
<tr><td>Error5</td><td>The fifth error to occur</td></tr>
<tr><td>Error6</td><td>The sixth error to occur</td></tr>
<tr><td>Error7</td><td>The seventh error to occur</td></tr>
<tr><td>Error8</td><td>The eight error to occur</td></tr>
<tr><td>Clear Log</td><td colspan="2">Clears the error log entries and count.</td></tr>
<tr><td>String Count</td><td colspan="2">Number of User Strings Defined</td></tr>
<tr><td>String Space</td><td colspan="2">Space Available For User Strings.</td></tr>
<tr><td>Segments Left</td><td colspan="2">Number of Available Program Segments
Gives the number of unused program segments. Each time a segment is allocated to a program, this value is reduced by one.</td></tr>
</table>

Folder: Instrument	Sub-folder: Diagnostics
Name	**Parameter Description**
Ctl Stack Free	Control Stack Free Space (words) The number of words of un-used stack for the control task
Comms Stack Free	Comms Stack Free Space (words) The number of words of un-used stack for the comms task
Idle Stack Free	Idle Stack Free Space (words) The number of words of un-used stack for the idle (background) task.
Max segments	Max number of setpoint programmer segments available
MaxSegsPerProg	Specifies the maximum number of segments that can be configured for a single program
CntrlOverrun	Indicates the amount of control overrun.
PSUident	Shows type of PSU fitted 0 = Mains 1= 24V dc
PwrFailCount	Counts the number of times the instrument power has been switched off.
Cust1Name	Name for custom linearisation table 1
Cust1Name	Name for custom linearisation table 1
Cust1Name	Name for custom linearisation table 1

7. CHAPTER 7 I/O FOLDER

This lists the modules fitted into the instruments, all the IO channels, the fixed IO and the current monitoring.

The IO folder lists all the channels of each of the IO boards in the 4 available slots. Each board has up to 8 inputs or outputs making a maximum of 32 channels. The channels are listed under Mod1 to Mod32.

Slot	Channels
1	IO.Mod.1 to IO.Mod.8
2	IO.Mod.9 to IO.Mod.16
3	IO.Mod.17 to IO.Mod.24
4	IO.Mod.24 to IO.Mod.32

Note that the current transformer input, CT3, is not included in this arrangement. There is a separate folder for current monitoring under IO.CurrentMonitor. If this board is fitted into slot 2 the IO.Mod.9 to Mod.16 would not exist.

7.1 Module ID

Folder: IO		**Sub-folder: ModIDs**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Module1	Module1Ident	0 NoMod – No Module 24 DO8Mod – 8 logic outputs 90 CT3Mod – 3 current transformer inputs 131 TC8Mod – 8 thermocouple inputs 201 AO8Mod – 8 mA outputs (Module 4 only)	0	Read Only
Module2	Module2Ident		0	Read Only
Module3	Module3Ident		0	Read Only
Module4	Module4Ident		0	Read Only

7.1.1 Modules

The content of the Mod folders depends on the type of IO module fitted in each slot. These will be covered in sections 7.2 to 7.5.

7.2 Logic Output

If a slot is fitted with a DO8 board then 8 channels will be available to be configured and connected to Loop outputs, alarms or other logic signals.

7.2.1 Logic Out Parameters

Folder – IO		Sub-folder Mod.1 to .32			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Ident	Channel Identity	LogicOut			Read Only
IOType	IO Type	OnOff	On off output		Conf
		Time Prop	Time proportioning output		
Invert	Sets the sense of the logic input or output	No Yes	No inversion Inverted	No	Conf
The next five parameters are only shown when ‘IO Type’ = ‘Time Prop’ outputs					
Min OnTime	Minimum output on/off time. Prevents relays from switching too rapidly	Auto 0.01 to 150.00 seconds	Auto = 20ms. This is the fastest allowable update rate for the output	Auto	Oper
Disp Hi	The maximum displayable reading	0.00 to 100.00		100.00	Oper
Disp Lo	The minimum displayable reading	0.00 to 100.00		0.00	Oper
Range Hi	The maximum (electrical) input/output level	0.00 to 100.00			Oper
Range Lo	The minimum (electrical) input/output level	0.00 to 100.00			Oper
Always displayed					
Meas Val	The current value of the output demand signal to the hardware	0 1	On (unless Invert = Yes) Off (unless Invert = Yes)		Read only
PV	When configured as an output, this is the desired output value; when configured as an input the current state of the digital input is displayed	0 to 100 or 0 to 1 (OnOff)			Oper

PV can be wired from the output of a function block. For example if it is used for control it may be wired from the control loop output (Ch1 Output).

7.2.2 Logic Output Scaling

If the output is configured for time proportioning control, it can be scaled such that a lower and upper level of PID demand signal can limit the operation of the output value.

By default, the output will be fully off for 0% power demand, fully on for 100% power demand and equal on/off times at 50% power demand. You can change these limits to suit the process. It is important to note, however, that these limits are set to safe values for the process. For example, for a heating process it may be required to maintain a minimum level of temperature. This can be achieved by applying an offset at 0% power demand which will maintain the output on for a period of time. Care must be taken to ensure that this minimum on period does not cause the process to overheat.

If Range Hi is set to a value <100% the time proportioning output will switch at a rate depending on the value - it will not switch fully on.

Similarly, if Range Lo is set to a value >0% it will not switch fully off.

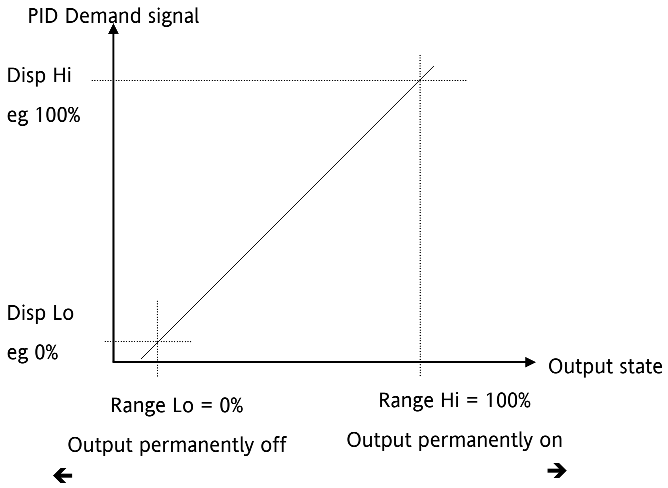


Figure 7-1: Time Proportioning Output

7.2.3 Example: To Scale a Proportioning Logic Output

Access level must be configuration.

COM1.ID255-Mini8 - Parameter Explorer (IO.Mod.17)

Name	Description	Address	Value
Ident	Channel Ident	21462	LogicOut (3)
IOType	IO Type	21412	TimeProp (51)
Invert	Invert	21447	No (0)
MinOnTime	Minimum OnTime	21441	Auto (0)
DisplayHigh	Display High	21419	100.00
DisplayLow	Display Low	21421	0.00
RangeHigh	Range High	21423	90.00
RangeLow	Range Low	21425	8.00
MeasuredVal	Measured Value	21431	1.00
PV	Process Variable	21433	0.00

IO.Mod.17 - 10 parameters (29 hidden)

In this example the output will switch on for 8% of the time when the PID demand wired to ‘PV’ signal is at 0%.

Similarly, it will remain on for 90% of the time when the demand signal is at 100%

7.3 Thermocouple Input

If a TC8 board is fitted then 8 thermocouple input channels will be available.

7.3.1 Thermocouple Input Parameters

Folder – IO		Sub-headers: Mod .1 to .32			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Ident	Channel Ident	TCinput			Read Only
IO Type	IO Type	Thermocouple			Conf
Lin Type	Input linearisation	see section 7.3.2			Conf
Units	Display units used for units conversion	see section 7.3.2			Conf
Resolution	Resolution	XXXXX to X.XXXX	Sets scaling for digital communications using the SCADA table		Conf
CJC Type	To select the cold junction compensation method	Internal 0°C 45°C 50°C External Off	See description in section 6.2.2. for further details	Internal	Conf
SBrk Type	Sensor break type	Low	Sensor break will be detected when its impedance is greater than a ‘low’ value		Conf
		High	Sensor break will be detected when its impedance is greater than a ‘high’ value		
		Off	No sensor break		
SBrk Alarm	Sets the alarm action when a sensor break condition is detected	ManLatch	Manual latching (see also the alarm Chapter 11 Alarms)		Oper
		NonLatch	No latching		
		Off	No sensor break alarm		
AlarmAck	Sensor Break alarm acknowledge	No Yes		No	Oper
Fallback	Fallback Strategy See also section 6.2.5.	Downscale	Meas Value = Input range lo - 5%		Conf
		Upscale	Meas Value = Input range Hi + 5%		
		Fall Good	Meas Value = Fallback PV		
		Fall Bad	Meas Value = Fallback PV		
		Clip Good	Meas Value = Input range Hi/lo +/- 5%		
		Clip Bad	Meas Value = Input range Hi/lo +/- 5%		
Fallback PV	Fallback value See also section 6.2.5.		Instrument range		Conf
Filter Time	Input filter time. An input filter provides damping of the input signal. This may be necessary to prevent the effects of excessive noise on the PV input.		Off to 500:00 (hhh:mm) m:ss.s to hh:mm:ss to hhh:mm	0:00.4	Oper
Measured Value	The current electrical value of the PV input				R/O

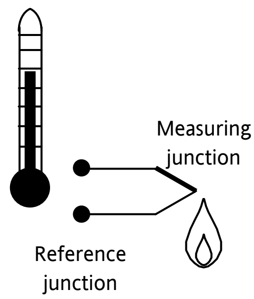
Folder – IO		Sub-headers: Mod .1 to .32			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
PV	The current value of the PV input after linearisation		Instrument range		R/O
Offset	Used to add a constant offset to the PV see section 6.2.6.		Instrument range		Oper
CJC Temp	Reads the temperature of the rear terminals at the thermocouple connection				R/O
SBrk Value	Sensor break Value Used for diagnostics only, and displays the sensor break trip value				R/O
Cal State	Calibration state Calibration of the PV Input is described in Chapter 22. 3	Idle			Conf
Status	PV Status The current status of the PV.	0 1 2 3 4 5 6	Normal operation Initial startup mode Input in sensor break PV outside operating limits Saturated input Uncalibrated channel No Module		R/O

7.3.2 Linearisation Types and Ranges

Input Type		Min Range	Max Range	Units	Min Range	Max Range	Units
J	Thermocouple type J	-210	1200	°C	-238	2192	°F
K	Thermocouple type K	-200	1372	°C	-238	2498	°F
L	Thermocouple type L	-200	900	°C	-238	1652	°F
R	Thermocouple type R	-50	1700	°C	-58	3124	°F
B	Thermocouple type B	0	1820	°C	32	3308	°F
N	Thermocouple type N	-200	1300	°C	-238	2372	°F
T	Thermocouple type T	-200	400	°C	-238	752	°F
S	Thermocouple type S	-50	1768	°C	-58	3214	°F
PL2	Thermocouple Platinel II	0	1369	°C	32	2466	°F
C	Custom						
Linear	mV or mA linear input	-10.00	80.00				
SqRoot	Square root						
Custom	Customised linearisation tables						

7.3.3 CJC Type

A thermocouple measures the temperature difference between the measuring junction and the reference junction. The reference junction, therefore, must either be held at a fixed known temperature or accurate compensation be used for any temperature variations of the junction.



7.3.3.1 Internal Compensation

The controller is provided with a temperature sensing device which senses the temperature at the point where the thermocouple is joined to the copper wiring of the instrument and applies a corrective signal.

Where very high accuracy is needed and to accommodate multi-thermocouple installations, larger reference units are used which can achieve an accuracy of ±0.1°C or better. These units also allow the cables to the instrumentation to be run in copper. The reference units are contained basically under three techniques, Ice-Point, Hot Box and Isothermal.

7.3.3.2 The Ice-Point

There are usually two methods of feeding the EMF from the thermocouple to the measuring instrumentation via the ice-point reference, the bellows type and the temperature sensor type.

The bellows type utilises the precise volumetric increase which occurs when a known quantity of ultra pure water changes state from liquid to solid. A precision cylinder actuates expansion bellows which control power to a thermoelectric cooling device. The temperature sensor type uses a metal block of high thermal conductance and mass, which is thermally insulated from ambient temperatures. The block temperature is lowered to 0°C by a cooling element, and maintained there by a temperature sensing device.

Special thermometers are obtainable for checking the 0°C reference units and alarm circuits that detect any movement from the zero position can be fitted.

7.3.3.3 The Hot Box

Thermocouples are calibrated in terms of EMF generated by the measuring junctions relative to the reference junction at 0°C. Different reference points can produce different characteristics of thermocouples, therefore referencing at another temperature does present problems. However, the ability of the hot box to work at very high ambient temperatures, plus a good reliability factor has led to an increase in its usage. The unit can consist of a thermally insulated solid aluminium block in which the reference junctions are embedded.

The block temperature is controlled by a closed loop system, and a heater is used as a booster when initially switching on. This booster drops out before the reference temperature, usually between 55°C and 65°C, is reached, but the stability of the hot box temperature is now important. Measurements cannot be taken until the hot box reaches the correct temperature.

7.3.3.4 Isothermal Systems

The thermocouple junctions being referenced are contained in a block which is heavily thermally insulated. The junctions are allowed to follow the mean ambient temperature, which varies slowly. This variation is accurately sensed by electronic means, and a signal is produced for the associated instrumentation. The high reliability factor of this method has favoured its use for long term monitoring.

7.3.3.5 CJC Options in Mini8 Series

0: CJC measurement at instrument terminals
1: CJC based on external junctions kept at 0C (Ice Point)
2: CJC based on external junctions kept at 45C (Hot Box)
3: CJC based on external junctions kept at 50C (Hot Box)
4: CJC based on independent external measurement
5: CJC switched off

7.3.4 Sensor Break Value

The controller continuously monitors the impedance of a transducer or sensor connected to any analogue input. This impedance, expressed as a % of the impedance which causes the sensor break flag to trip, is a parameter called ‘SBrk Trip Imp’ and is available in the parameter lists associated with analogue inputs. The table below shows the typical impedance which causes sensor break to trip for various types of input and high and low SBrk Impedance parameter settings. The impedance values are only approximate (±25%) as they are not factory calibrated.

TC8 Input	SBrk Impedance – High	~ 12KΩ
Range -77 to +77mV	SBrk Impedance – Low	~ 3KΩ

7.3.5 Fallback

A Fallback strategy may be used to configure the default value for the PV in case of an error condition. The error may be due an out of range value, a sensor break, lack of calibration or a saturated input.

The Status parameter would indicate the error condition and could be used to diagnose the problem.

Fallback has several modes and may be associated with the Fallback PV parameter

The Fallback PV may be used to configure the value assigned to the PV in case of an error condition. The Fallback parameter should be configured accordingly.

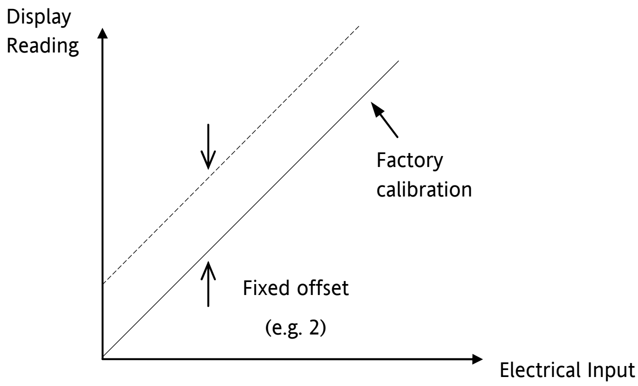
The fallback parameter may be configured so as to force a Good or Bad status when in operation. This in turn allows the user to choose to override or allow error conditions to affect the process.

7.3.6 PV Offset

All ranges of the controller have been calibrated against traceable reference standards. This means that if the input type is changed it is not necessary to calibrate the controller. There may be occasions, however, when you wish to apply an offset to the standard calibration to take account of known errors within the process, for example, a known sensor error or a known error due to the positioning of the sensor. In these instances it is not advisable to change the reference calibration, but to apply a user defined offset.

It is also possible to apply a two point offset and this is described in the next section.

PV Offset applies a single offset over the full display range of the controller and can be adjusted in Level 3. It has the effect of moving the curve up a down about a central point as shown in the example below:-



7.3.6.1 Example: To Apply an Offset:-

- Connect the input of the controller to the source device which you wish to calibrate to
- Set the source to the desired calibration value
- The controller will show the current measurement of the value
- If the value is correct, the controller is correctly calibrated and no further action is necessary. If you wish to offset the reading use the Offset parameter where

Corrected value (PV) = input value + Offset.

7.3.7 Using TC8 input as a mV input

Leave the IO type as Thermocouple

1. Set the Linearisation Type as Linear.
2. Use a Maths2 block to scale the mV into engineering units.

Maximum input range is ± 77 mV

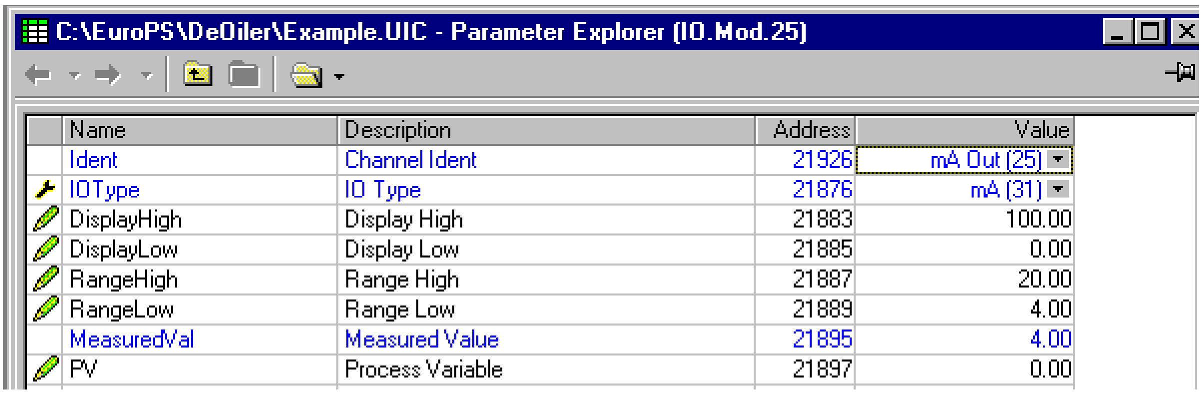
7.4 Analogue Output

If an AO8 module is fitted in Slot 4 then 8 analogue output channels are available.

Folder – IO		**Sub-folder: Mod.25 to Mod.32**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Ident	Channel ident	mAout			R/O
IO Type	To configure the output drive signal	mA	milli-amps dc		Conf
Resolution	Display resolution	XXXXX to X.XXXX	Determiones scaling for SCADA communications		Conf
Disp Hi	Display high reading	-99999 to 99999 decimal points depend on resolution		100	Oper
Disp Lo	Display low reading			0	Oper
Range Hi	Electrical high input level	0 to 20		20	Oper
Range Lo	Electrical low input level			4	Oper
Meas Value	The current output value				R/O
PV					Oper
Cal State	Calibration state	Idle Lo Hi Confirm Go Abort Busy Passed Failed Accept	Non calibrating state Select calibration of the low position Select calibration of the high position Confirm the position to calibrate Start calibration Abort calibration Controller automatically calibrating Calibration OK Calibration bad To store the new values	Idle	Conf
Status	PV Status The current status of the PV.	0 1 2 3 4 5 6	Normal operation Initial startup mode Input in sensor break PV outside operating limits Saturated input Uncalibrated channel No Module		R/O

7.4.1 Example – 4 to 20mA Analogue Output

In this example 0% (=Display Low) to 100% (=Display High) from a Loop PID Output wired to this PV input will give a 4mA (=Range Low) to 20mA (=Range High) control signal.



Name	Description	Address	Value
Ident	Channel Ident	21926	mA Out (25)
IOType	IO Type	21876	mA (31)
DisplayHigh	Display High	21883	100.00
DisplayLow	Display Low	21885	0.00
RangeHigh	Range High	21887	20.00
RangeLow	Range Low	21889	4.00
MeasuredVal	Measured Value	21895	4.00
PV	Process Variable	21897	0.00

7.5 Fixed IO

There are two digital inputs, designated D1 and D2.

Folder: IO		**Sub-folder: Fixed IO.D1 and .D2**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Ident	Channel Ident	LogicIn	LogicIn	Read Only
IO Type	IO Type	Input	Input	Read Only
Invert	Invert	No/Yes – input sense is inverted	No	Conf
Measured Val	Measured Value	On/Off	Off	Read Only
PV	Process Variable	On/Off	Off	Read Only

There are two fixed relay outputs, designated A and B

Folder: IO		**Sub-folder: Fixed IO.A and .B**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Ident	Channel Ident	Relay	Relay	Read Only
IO Type	IO Type	OnOff	OnOff	Read Only
Invert	Invert	No/Yes = output sense is inverted.	No	Conf
Measured Val	Measured Value	On/Off	Off	Read Only
PV	Process Variable	On/Off	Off	Oper

7.6 Current Monitor

The Mini8, with a CT3 card, has the capability of detecting failures of up to 16 heater loads by measuring the current flowing through them via 3 current transformer inputs. The failures that can be detected are:

SSR Fault

If current is detected flowing through the heater when the controller is requesting it to be off then this indicates that the SSR has a short circuit fault. If current is not detected when the controller is requesting the heater to be on it indicates that the SSR has an open circuit fault.

Partial Load Fault (PLF)

If less current is detected flowing through the heater than the PLF threshold then this indicates that the heater has a fault; in applications that use multiple heater elements in parallel then it indicates that one or more of the elements has an open circuit fault.

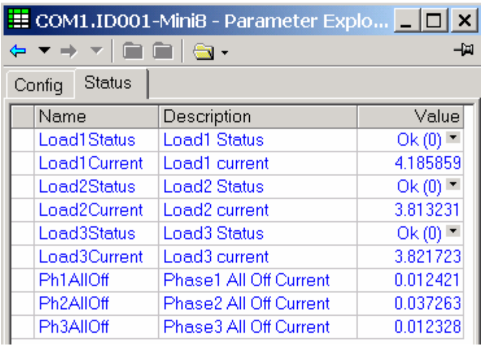
Over Current Fault (OCF)

If more current is detected flowing through the heater than the OCF threshold then this indicates that the heater has a fault; in applications that use multiple heater elements in parallel then it indicates that one or more of the elements has short circuit fault.

Heater failures are indicated via individual load status parameters and via four status words. In addition, a global alarm parameter will indicate when a new CT alarm has been detected, which, will also be registered in the alarm log.

7.6.1.1 Current Measurement

Individual LoadCurrent parameters indicate the current measured for each heater. The Current Monitor function block utilises a cycling algorithm to measure the current flowing through one heater per measurement interval (default 10s, user alterable). Compensation within the control loop minimises the disturbance to the PV when current through a load is being measured.



Name	Description	Value
Load1Status	Load1 Status	Ok (0)
Load1Current	Load1 current	4.185859
Load2Status	Load2 Status	Ok (0)
Load2Current	Load2 current	3.813231
Load3Status	Load3 Status	Ok (0)
Load3Current	Load3 current	3.821723
Ph1AllOff	Phase1 All Off Current	0.012421
Ph2AllOff	Phase2 All Off Current	0.037263
Ph3AllOff	Phase3 All Off Current	0.012328

The interval between successive measurements is dependent upon the average output power required to maintain SP. The recommended absolute minimum interval can be calculated as follows:

*Minimum interval (s) > 0.25 * (100/average output power to maintain SP).*

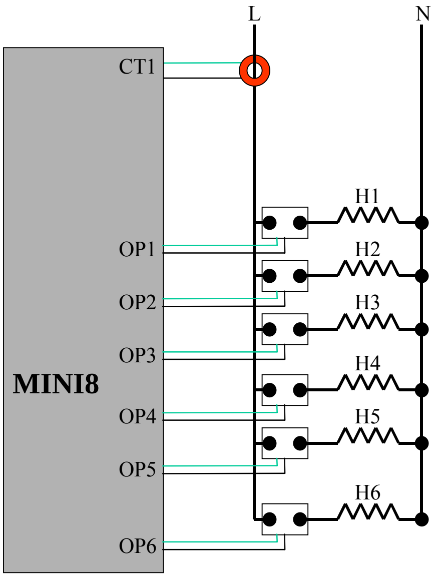
For example, if average output power to maintain SP is 10%, using the above rule, the recommended minimum interval is 2.5 seconds. The interval may need to be adjusted depending upon the response of the heaters being used.

7.6.2 Single Phase Configurations

7.6.2.1 Single SSR triggering

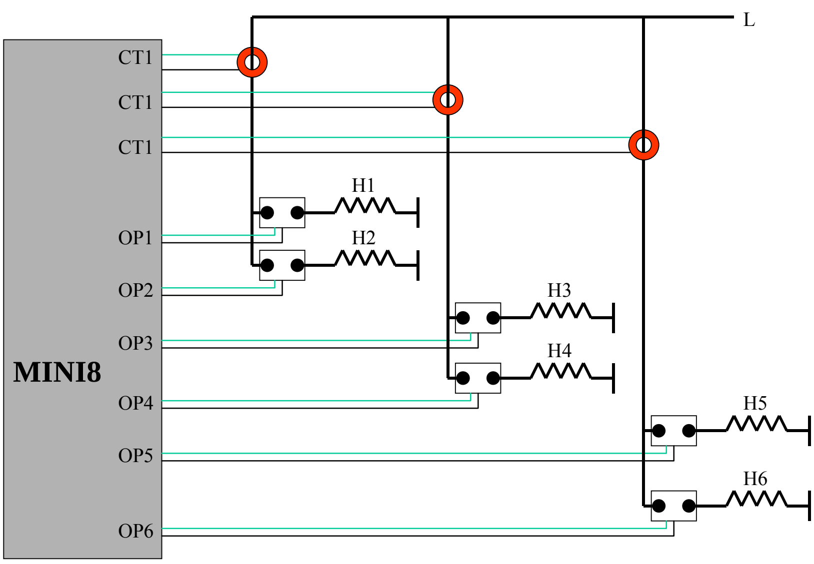
With this configuration, failures of individual heater loads can be detected. For example, if the current detected flowing through Heater 3 is less than its PLF threshold then this will be indicated as Load3PLF

Example1 – Using one CT input



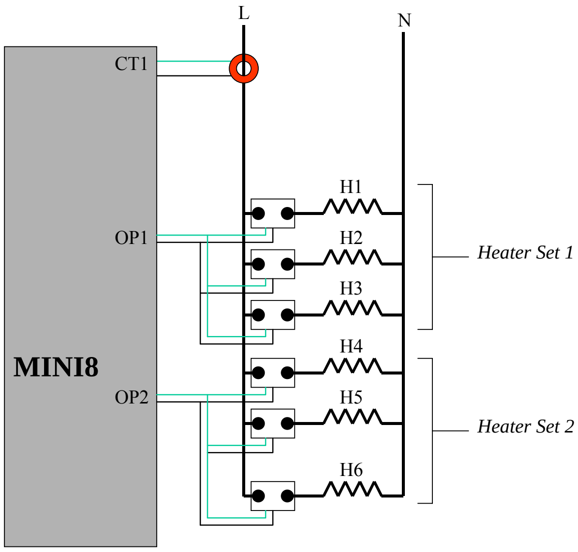
Note: Maximum of 6 Heaters can be connected to one CT input

Example2 – Using three CT inputs



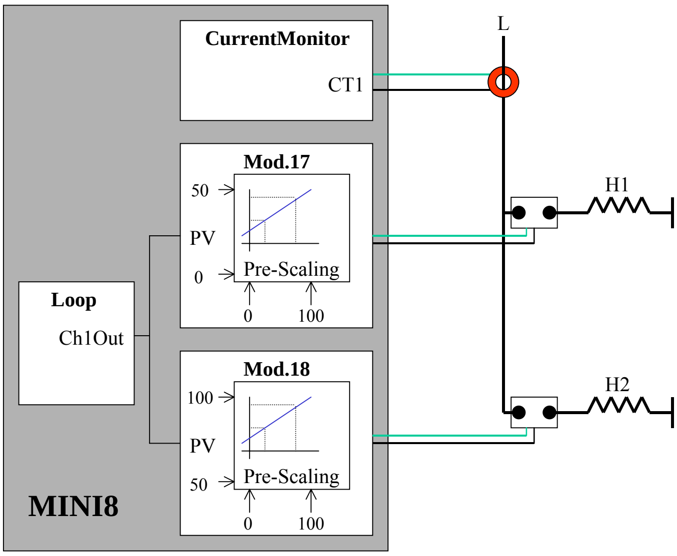
7.6.2.2 Multiple SSR triggering

With this configuration, failure of a set of heater loads can be detected. For example, if the current detected flowing through Heater Set 1 is less than Load1's PLF threshold then this will be indicated as Load1PLF. Further investigation will then be required to determine which heater within set1 has failed.



7.6.2.3 Split Time Proportioning Outputs

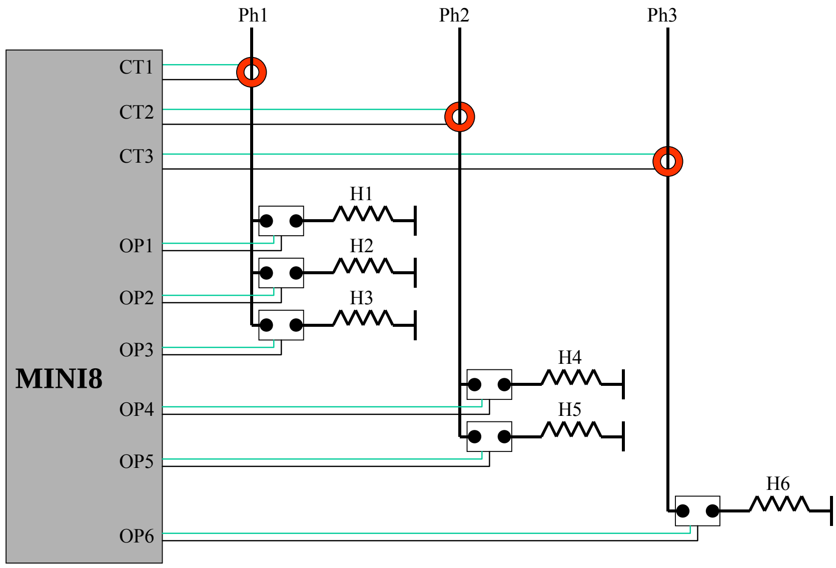
This is where a single power demand is split and applied to two time proportioning outputs, that have been scaled, allowing the loads to switch on incrementally as the output power increases. For example, Heater1 will deliver any demand from 0-50%, and Heater2 will deliver any demand from 50-100% (with Heater1 fully on).



As the Mini8 has the capability of detecting faults with up to 16 heater loads it can handle this type of application even if all 8 loops have split time proportioning outputs.

7.6.3 Three Phase Configuration

Configuration for Three Phase supply applications is similar to that for Single phase using three CT inputs.



Note: Maximum of 6 Heaters can be connected to one CT input

7.6.4 Parameter Configuration

If Current Monitor is enabled in the folder Instrument/Options/Current Monitor then the current monitor configuration folder appears as a subfolder in IO.

Folder: IO		**Sub-folder: CurrentMonitor/Config**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Commission	Commission CT	No Auto Manual Accept Abort	See Chapter 7.11.2 and 7.11.4	No	Oper
CommissionStatus	Commission Status	Not commissioned Commissioning NoDO8 card fitted NoloopTPouts SSRfault MaxLoadsCT1/2/3 NotAccepted Passed ManuallyConfigured	Not commissioned Commissioning in progress There are no DO8 cards installed in the instrument. The digital outputs are either not configured as time proportioning or are not wired from loop heater channels. Either a SSR short circuit or open circuit fault is present. More than 6 heaters have been connected to CT input 1or 2 or 3. Commissioning failed Successfully auto commissioned Configured manually	0	Read Only
Interval	Measurement Interval	1s to 1m		10s	Oper
Inhibit	Inhibit	No – current is measured Yes –current measurement is inhibited		No	Oper
MaxLeakPh1	Max Leakage Current Phase 1	0.25 to 1 amp		0.25	Oper
MaxLeakPh2	Max Leakage Current Phase 2	0.25 to 1 amp		0.25	Oper
MaxLeakPh3	Max Leakage Current Phase 3	0.25 to 1 amp		0.25	Oper
CT1Range	CT input 1 range	10 to 1000 amps		10	Oper
CT2Range	CT input 2 range	10 to 1000 amps		10	Oper
CT3Range	CT input 3 range	10 to 1000 amps		10	Oper
CalibrateCT1	Calibrate CT1	Idle 0mA -70mA LoadFactorCal SaveUserCal	See Chapter 7.12	Idle	Oper
CalibrateCT2	Calibrate CT2	As CT1		Idle	Oper
CalibrateCT3	Calibrate CT3	As CT1		Idle	Oper

7.6.5 Commissioning

7.6.5.1 Auto Commission

Auto commissioning of the Current Monitor is a feature that automatically detects which time proportioning outputs drive individual heaters (or heater sets), detects which CT input individual heaters are associated with and determines the Partial Load and Over Current thresholds using a 1:8 ratio. If auto commissioning fails, a status parameter indicates the reason why.

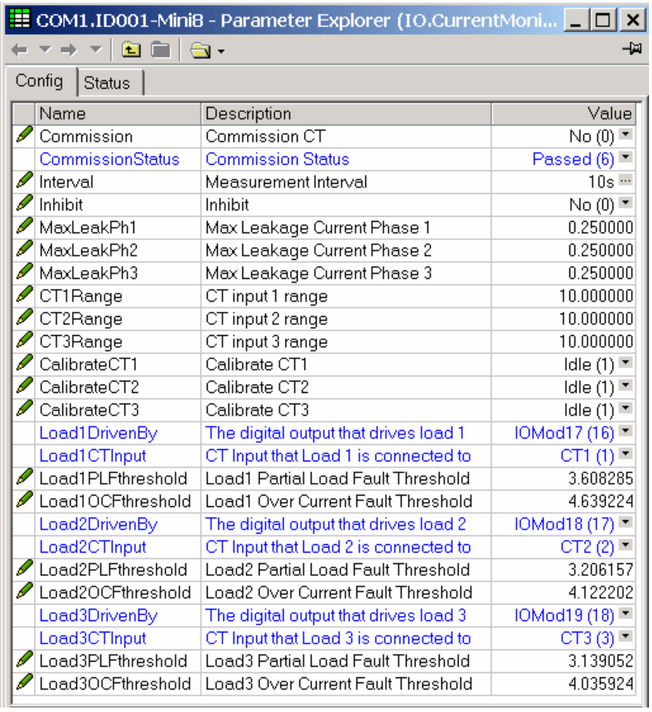
How to Auto Commission

1. Ensure the process is enabled and powered for full operation of the heating circuit with the digital outputs configured as Time Proportioning and 'soft' wired to the appropriate loop heater channels. During auto commissioning digital outputs will switch on and off.

2. Put instrument into Operator Mode.

3. Set Commission to Auto and CommissionStatus will display 'Commissioning'.

4. If successful, CommissionStatus will display Passed and configured load parameters will become available. If unsuccessful, CommissionStatus displays the offending fault.



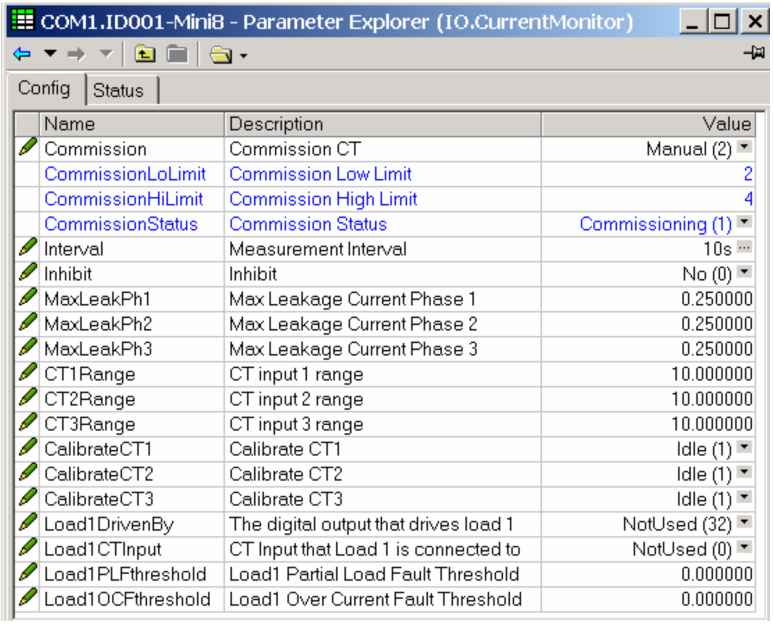
Name	Description	Value
Commission	Commission CT	No (0)
CommissionStatus	Commission Status	Passed (6)
Interval	Measurement Interval	10s
Inhibit	Inhibit	No (0)
MaxLeakPh1	Max Leakage Current Phase 1	0.250000
MaxLeakPh2	Max Leakage Current Phase 2	0.250000
MaxLeakPh3	Max Leakage Current Phase 3	0.250000
CT1Range	CT input 1 range	10.000000
CT2Range	CT input 2 range	10.000000
CT3Range	CT input 3 range	10.000000
CalibrateCT1	Calibrate CT1	Idle (1)
CalibrateCT2	Calibrate CT2	Idle (1)
CalibrateCT3	Calibrate CT3	Idle (1)
Load1DrivenBy	The digital output that drives load 1	IOMod17 (16)
Load1CTInput	CT Input that Load 1 is connected to	CT1 (1)
Load1PLFthreshold	Load1 Partial Load Fault Threshold	3.608285
Load1OCFthreshold	Load1 Over Current Fault Threshold	4.639224
Load2DrivenBy	The digital output that drives load 2	IOMod18 (17)
Load2CTInput	CT Input that Load 2 is connected to	CT2 (2)
Load2PLFthreshold	Load2 Partial Load Fault Threshold	3.206157
Load2OCFthreshold	Load2 Over Current Fault Threshold	4.122202
Load3DrivenBy	The digital output that drives load 3	IOMod19 (18)
Load3CTInput	CT Input that Load 3 is connected to	CT3 (3)
Load3PLFthreshold	Load3 Partial Load Fault Threshold	3.139052
Load3OCFthreshold	Load3 Over Current Fault Threshold	4.035924

7.6.5.2 Manual Commission

Manual Commissioning is also available and is intended for those users who want to commission the Current Monitor off-line or do not want to accept auto commissioned settings.

How to Manual Commission

.



Name	Description	Value
Commission	Commission CT	Manual (2)
CommissionLoLimit	Commission Low Limit	2
CommissionHiLimit	Commission High Limit	4
CommissionStatus	Commission Status	Commissioning (1)
Interval	Measurement Interval	10s
Inhibit	Inhibit	No (0)
MaxLeakPh1	Max Leakage Current Phase 1	0.250000
MaxLeakPh2	Max Leakage Current Phase 2	0.250000
MaxLeakPh3	Max Leakage Current Phase 3	0.250000
CT1Range	CT input 1 range	10.000000
CT2Range	CT input 2 range	10.000000
CT3Range	CT input 3 range	10.000000
CalibrateCT1	Calibrate CT1	Idle (1)
CalibrateCT2	Calibrate CT2	Idle (1)
CalibrateCT3	Calibrate CT3	Idle (1)
Load1DrivenBy	The digital output that drives load 1	NotUsed (32)
Load1CTInput	CT Input that Load 1 is connected to	NotUsed (0)
Load1PLFthreshold	Load1 Partial Load Fault Threshold	0.000000
Load1OCFthreshold	Load1 Over Current Fault Threshold	0.000000

1. Set Commission to Manual. CommissionStatus will display Commissioning and Load1 configuration parameters will become available
2. Set Load1DrivenBy to the IO Module that is connected to the heater load.
3. Set Load1CTInput to the CT input number that is connected to the heater load.
4. Set Load1PLFthreshold and Load1OCFthreshold to appropriate values for the heater load.
5. Repeat for other loads.
6. To use the commissioned settings set Commission to ‘Accept’. CommissionStatus will display ManuallyConfigured.
7. To stop manual commissioning set Commission to ‘Abort’. CommissionStatus will display NotCommissioned.

7.6.6 Calibration

A Mini8 supplied from factory with the CT3 card already installed the CT inputs will have been factory calibrated. If the CT3 card is installed at a later date then default calibration values are automatically loaded into the instrument. However, three calibration parameters, one for each CT input, are provided to allow the inputs to be calibrated in the field.

Note: DC Current Source, capable of outputting a –70mA signal, is required to calibrate the inputs.

The CT inputs are calibrated individually.

How to Calibrate

1. Apply the stimulus (0mA or –70mA) from the DC current source to the CT input to be calibrated.
2. Set CalibrateCT1, to reflect the stimulus being applied to the input.
3. CalibrateCT1 displays ‘Confirm’. Select ‘Go’ to proceed with the calibration process.
4. After selecting Go, CalibrateCT1 displays ‘Calibrating’.
5. If calibration was successful, CalibrateCT1 displays ‘Passed’. Select ‘Accept’ to keep the calibration values.
6. If calibration was unsuccessful, CalibrateCT1 displays ‘Failed’. Select ‘Abort’ to reject the calibration.
7. Select ‘SaveUserCal’ to save the calibration values into non-volatile memory.
8. Select ‘LoadFactCal’ to restore calibration values to the factory calibrated or default settings.
9. Note: It is possible to stop the calibration process at anytime by selecting ‘Abort’.

Follow the same procedure for CT2 and CT3.

8. CHAPTER 8 ALARMS

Alarms are used to alert the system when a pre-set level has been exceeded or a particular condition has changed state. As the Mini8 has no display to show alarms the alarm flags are all available over communications in status words See Alarm Summary (Section 8.7) . They may also be wired directly or via logic to an output such as a relay.

Alarms can be divided into two main types. These are:-

Analogue alarms - operate by monitoring an analogue variable such as the process variable and comparing it with a set threshold.

Digital alarms – operate when the state of a boolean variable changes, for example, sensor break.

Number of Alarms - up to 32 analogue and 32 digital alarms may be configured.

8.1 Further Alarm Definitions

Hysteresis is the difference between the point at which the alarm switches 'ON' and the point at which it switches 'OFF'. It is used to provide a definite indication of the alarm condition and to prevent alarm relay chatter.

Latching Alarm used to hold the alarm condition once an alarm has been detected. It may be configured as:-

None	Non latching	A non latching alarm will reset itself when the alarm condition is removed
Auto	Automatic	An auto latching alarm requires acknowledgement before it is reset. The acknowledgement can occur BEFORE the condition causing the alarm is removed.
Manual	Manual	The alarm continues to be active until both the alarm condition is removed AND the alarm is acknowledged. The acknowledgement can only occur AFTER the condition causing the alarm is removed.
Event	Event	Alarm output will activate.

Blocking Alarms The alarm may be masked during start up. Blocking prevents the alarm from being activated until the process has first achieved a safe state. It is used, for example, to ignore start up conditions which are not representative of running conditions. A blocking alarm is re-initiated after a setpoint change.

Delay A short time can be set for each alarm which prevents the output from going into the alarm state. The alarm is still detected as soon as it occurs, but if it cancels before the end of the delay period then no output is triggered. The timer for the delay is then reset. It is also reset if an alarm is changed from being inhibited to uninhibited.

8.2 Analogue Alarms

Analogue alarms operate on variables such as PV, output levels, etc. They can be soft wired to these variables to suit the process.

8.2.1 Analogue Alarm Types

Absolute High - an alarm occurs when the PV exceeds a set high threshold.

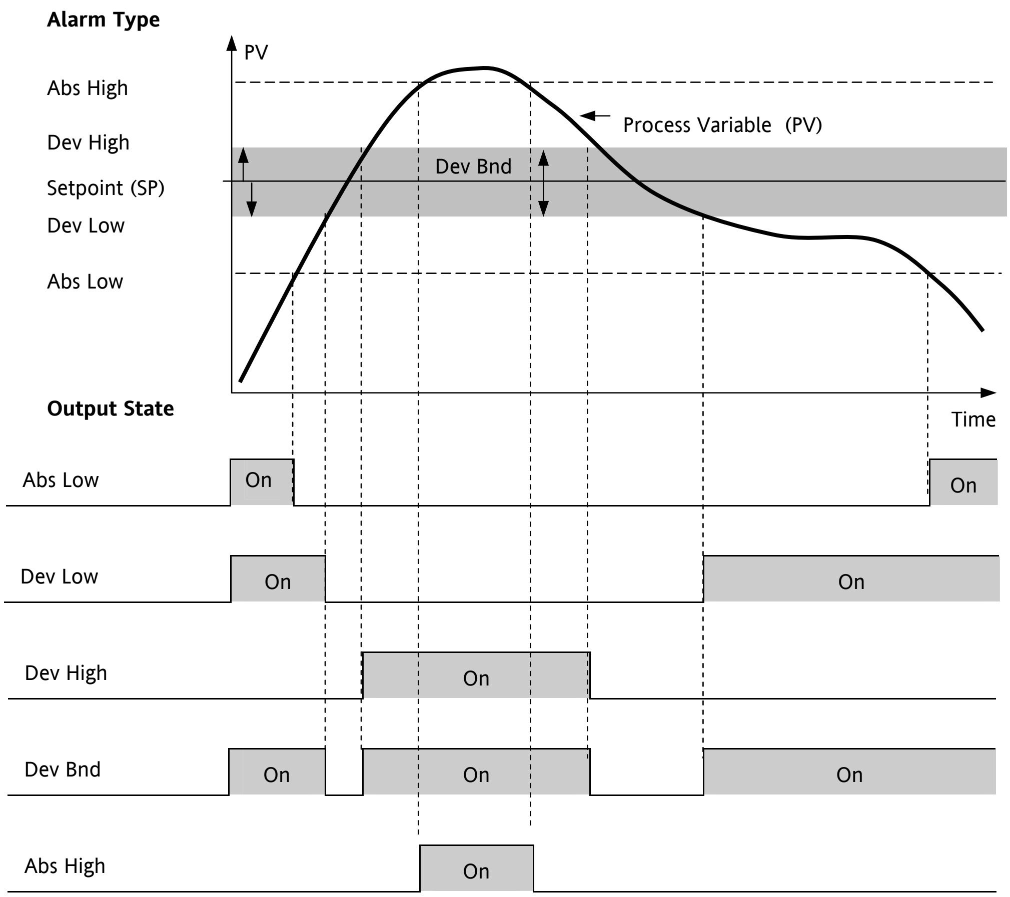
Absolute Low - an alarm occurs when the PV exceeds a set low threshold.

Deviation High - an alarm occurs when the PV is higher than the setpoint by a set threshold

Deviation Low - an alarm occurs when the PV is lower than the setpoint by a set threshold

Deviation Band - an alarm occurs when the PV is higher or lower than the setpoint by a set threshold

These are shown graphically below for changes in PV plotted against time. (Hysteresis set to zero)



8.3 Digital Alarms

Digital alarms operate on Boolean variables. They can be soft wired to any suitable Boolean parameter such as digital inputs or outputs.

8.3.1 Digital Alarm Types

Pos Edge The alarm will trigger when the input changes from a low to high condition

Neg Edge The alarm will trigger when the input changes from a high to low condition

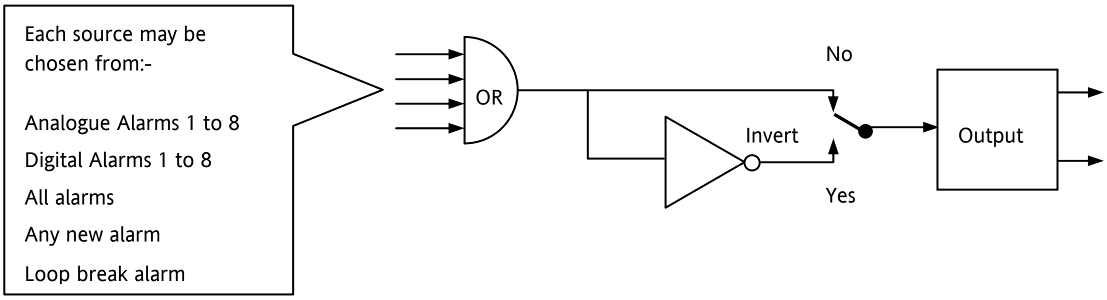
Edge The alarm will trigger on any change of state of the input signal

High The alarm will trigger when the input signal is high

Low The alarm will trigger when the input signal is low

8.4 Alarm Outputs

Alarms can operate a specific output (usually a relay). Any individual alarm can operate an individual output or any combination of alarms, up to four, can operate an individual output. They are wired as required in configuration level.



8.4.1 How Alarms are Indicated

Alarm states are all embedded in 16 bit status words. See Alarm Summary in Section 8.7

8.4.2 To Acknowledge an Alarm

Set the appropriate alarm acknowledge flag to acknowledge that particular alarm. Alternatively the GlobalAck in the AlmSummary folder can be used to acknowledge ALL alarms that require acknowledging in the instrument.

The action, which now takes place, will depend on the type of latching, which has been configured

8.4.2.1 Non Latched Alarms

If the alarm condition is present when the alarm is acknowledged, the alarm output will be continuously active. This state will continue for as long as the alarm condition remains. When the alarm condition clears the output will go off..
If the alarm condition clears before it is acknowledged the alarm output goes off as soon as the condition disappears.

8.4.2.2 Automatic Latched Alarms

The alarm continues to be active until both the alarm condition is removed AND the alarm is acknowledged. The acknowledgement can occur **BEFORE** the condition causing the alarm is removed.

8.4.2.3 Manual Latched Alarms

The alarm continues to be active until both the alarm condition is removed AND the alarm is acknowledged. The acknowledgement can only occur **AFTER** the condition causing the alarm is removed.

8.5 Alarm Parameters

Four groups of eight **analogue** alarms are available. The following table shows the parameters to set up and configure alarms.

Folder: Alarm	**Sub-folders: 1 to 32**				
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Type	Selects the type of alarm	None Abs Hi Abs Lo Dev Hi Dev Lo Dv Bnd	Alarm not configured Full Scale High Full Scale Low Deviation High Deviation Low Deviation band		Conf
Input	This is the parameter that will be monitored and compared against the threshold value to see if an alarm condition has occurred	Instrument range			Oper
Reference	The reference value is used in deviation alarms and the threshold is measured from this reference and not from its absolute value.	Instrument range			Oper
Threshold	The threshold is the value that the input is compared against to determine if an alarm has occurred.	Instrument range			Oper
Out	The output indicates whether the alarm is on or off depending on: the alarm condition, latching and acknowledge, inhibiting and blocking.	Off On	Alarm output deactivated Alarm output activated		R/O
Inhibit	Inhibit is an input to the Alarm function. It allows the alarm to be switched OFF. Typically the Inhibit is connected to a digital input or event so that during a phase of the process alarms do not activate. For Example, if the door to a furnace is opened the alarms may be inhibited until the door is closed again.	No Yes	Alarm not inhibited Inhibit function active	As order code	Oper
Hysteresis	Hysteresis is used to prevent signal noise from causing the Alarm output to oscillate. Alarm outputs become active as soon as the PV exceeds the Alarm Setpoint. They return to inactive after the PV has returned to the safe region by more than the hysterisis value. Typically the Alarm hysterisis is set to a value that is greater than the oscillations seen on the instrument display	Instrument range			Oper
Latch	Determine the type of latching the alarm will use, if any. Auto latching allows acknowledgement while the alarm condition is still active, whereas manual latching needs the condition to revert back to safe before the alarm can be acknowledged. See also the description at the beginning of this chapter	None Auto Manual Event	No latching is used Automatic Manual Event		Oper
Ack	Used in conjunction with the latching parameter. It is set when the user responds to an alarm.	No Yes	Not acknowledged Acknowledged		Oper

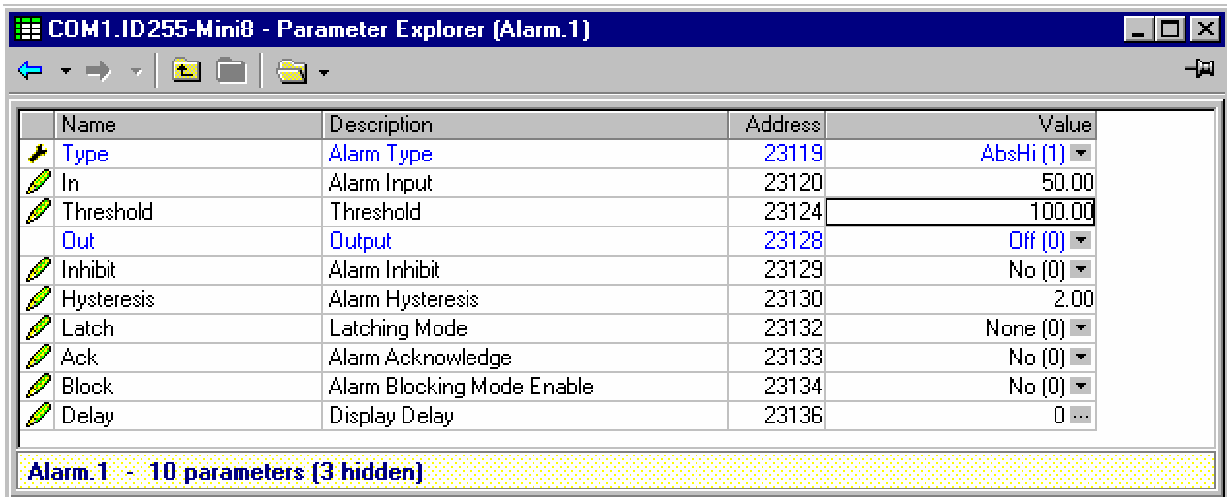
Folder: Alarm		Sub-folders: 1 to 32			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Block	Alarm Blocking is used to prevent alarms from activating during start-up. In some applications, the measurement at start-up is in an alarm condition until the system has come under control. Blocking causes the alarms to be ignored until the system is under control (in the safe state), after this any deviations trigger the alarm	No Yes	No blocking Blocking		Oper
Delay	This is a small delay between sensing the alarm condition and displaying it. If in the time between the two, the alarm goes safe, then no alarm is shown and the delay timer is reset. It can be used on systems that are prone to noise.	0:00.0 to 500 mm:ss.s hh:mm:ss hhh:mm		0:00.0	Oper

8.5.1 Example: To Configure Alarm 1

Change Access level to configuration.

In this example the high alarm will be detected when the measured value exceeds 100.00.

The current measured value is 50.00 as measured by the 'Input' parameter. This parameter will normally be wired to an internal source such as a thermocouple input. In this example the alarm will cancel when the measured value decreases 2 units below the trip level (at 98 units)



Name	Description	Address	Value
Type	Alarm Type	23119	AbsHi (1)
In	Alarm Input	23120	50.00
Threshold	Threshold	23124	100.00
Out	Output	23128	Off (0)
Inhibit	Alarm Inhibit	23129	No (0)
Hysteresis	Alarm Hysteresis	23130	2.00
Latch	Latching Mode	23132	None (0)
Ack	Alarm Acknowledge	23133	No (0)
Block	Alarm Blocking Mode Enable	23134	No (0)
Delay	Display Delay	23136	0

8.6 Digital Alarm Parameters

Four groups of eight **digital** alarms are available. The following table shows the parameters to set up and configure alarms.

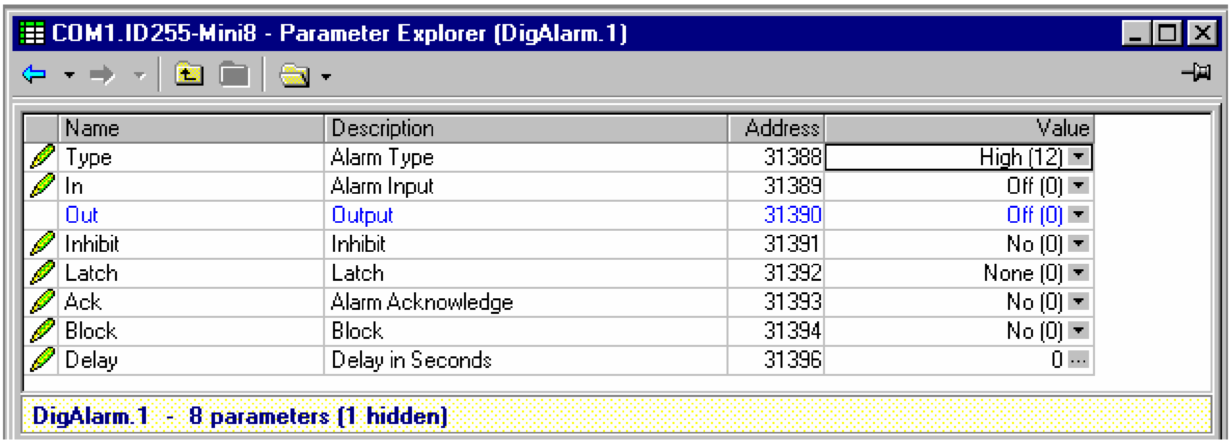
Folder: DigAlarm	**Sub-folders: 1 to 32**				
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Type	Selects the type of alarm	None PosEdge NegEdge Edge High Low	Alarm not configured On rising edge On falling edge On change High (1) Low (0)		Conf
In	This is the parameter that will be monitored and checked according to the AlarmType to see if an alarm condition has occurred	0 to 1			Oper
Out	The output indicates whether the alarm is on or off depending on: the alarm condition, latching and acknowledge, inhibiting and blocking.	Off On	Alarm output deactivated Alarm output activated		R/O
Inhibit	Inhibit is an input to the Alarm function. It allows the alarm to be switched OFF. Typically the Inhibit is connected to a digital input or event so that during a phase of the process alarms do not activate. For Example, if the door to a furnace is opened the alarms may be inhibited until the door is closed again.	No Yes	Alarm not inhibited Inhibit function active	As order code	Oper
Latch	Determine the type of latching the alarm will use, if any. Auto latching allows acknowledgement while the alarm condition is still active, whereas manual latching needs the condition to revert back to safe before the alarm can be acknowledged. See also the description at the beginning of this chapter	None Auto Manual Event	No latching is used Automatic Manual Event		Oper
Ack	Used in conjunction with the latching parameter. It is set when the user responds to an alarm.	No Yes	Not acknowledged Acknowledged		Oper
Block	Alarm Blocking is used to prevent alarms from activating during start-up. In some applications, the measurement at start-up is in an alarm condition until the system has come under control. Blocking causes the alarms to be ignored until the system is under control (in the safe state), after this any deviations trigger the alarm	No Yes	No blocking Blocking		Oper
Delay	This is a small delay between sensing the alarm condition and displaying it. If in the time between the two, the alarm goes safe, then no alarm is shown and the delay timer is reset. It can be used on systems that are prone to noise.	0:00.0 to 500 mm:ss.s hh:mm:ss hhh:mm		0:00.0	Oper

8.6.1 Example: To Configure DigAlarm 1

Change Access level to configuration.

In this example the digital alarm will come on if Timer 1 expires.

Timer.1.Out is wired to the alarm input. The DigAlarm.1.Out will turn on if the timer expires.



Name	Description	Address	Value
Type	Alarm Type	31388	High (12)
In	Alarm Input	31389	Off (0)
Out	Output	31390	Off (0)
Inhibit	Inhibit	31391	No (0)
Latch	Latch	31392	None (0)
Ack	Alarm Acknowledge	31393	No (0)
Block	Block	31394	No (0)
Delay	Delay in Seconds	31396	0

8.7 Alarm Summary

This is a summary of all the alarms in the Mini8. It provides global alarm and acknowledge flags as well as 16 bit status words which can be read over communications by the supervisory system.

Folder: AlmSummary Sub-folders: General

Name	Parameter Description	Value		Default	Access Level
NewAlarm	A new alarm has occurred since the last reset (excludes CT alarms)	Off/On			R/O
RstNewAlarm	Resets the NewAlarm flag	Yes / No		No	Oper
NewCTAlarm	A new Current alarm has occurred since the last reset	Off/On			R/O
RstNewCTAlarm	Resets the NewCTAlarm flag	Yes / No		No	Oper
AnyAlarm	Any new alarm since the last reset	Off/On			R/O
GlobalAck	Acknowledges every alarm in the Mini8 requiring acknowledgement.	No Yes	Not acknowledged Acknowledged		Oper
AnAlarmStatus1	16 bit word for analogue alarms 1 to 8	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 9 Bit 10 Bit 11 Bit 12 Bit 13 Bit 14 Bit 15	Set if Alarm 1 active Alarm 1 ack'd Set if Alarm 2 active Alarm 2 ack'd Set if Alarm 3 active Alarm 3 ack'd Set if Alarm 4 active Alarm 4 ack'd Set if Alarm 5 active Alarm 5 ack'd Set if Alarm 6 active Alarm 6 ack'd Set if Alarm 7 active Alarm 7 ack'd Set if Alarm 8 active Alarm 8 ack'd		R/O
AnAlarmStatus2	16 bit word for analogue alarms 9 to 16	Same format as above			R/O
AnAlarmStatus3	16 bit word for analogue alarms 17 to 24	Same format as above			R/O
AnAlarmStatus4	16 bit word for analogue alarms 25 to 32	Same format as above			R/O
DigAlarmStatus1	16 bit word for digital alarms 1 to 8	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 9 Bit 10 Bit 11 Bit 12 Bit 13 Bit 14 Bit 15	Set if Alarm 1 active Alarm 1 ack'd Set if Alarm 2 active Alarm 2 ack'd Set if Alarm 3 active Alarm 3 ack'd Set if Alarm 4 active Alarm 4 ack'd Set if Alarm 5 active Alarm 5 ack'd Set if Alarm 6 active Alarm 6 ack'd Set if Alarm 7 active Alarm 7 ack'd Set if Alarm 8 active Alarm 8 ack'd		R/O
DigAlarmStatus2	16 bit word for digital alarms 9 to 16	Same format as above			R/O

Folder: AlmSummary Sub-folders: General					
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
DigAlarmStatus3	16 bit word for digital alarms 17 to 24	Same format as above			R/O
DigAlarmStatus4	16 bit word for digital alarms 25 to 32	Same format as above			R/O
SBrkAlarmStatus1	16 bit word for IO channels Mod.1 to 8	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 9 Bit 10 Bit 11 Bit 12 Bit 13 Bit 14 Bit 15	Set if Mod.1 fault Alarm 1 ack'd Set if Mod.2 fault Alarm 2 ack'd Set if Mod.3 fault Alarm 3 ack'd Set if Mod.4 fault Alarm 4 ack'd Set if Mod.5 fault Alarm 5 ack'd Set if Mod.6 fault Alarm 6 ack'd Set if Mod.7 fault Alarm 7 ack'd Set if Mod.8 fault Alarm 8 ack'd		R/O
SbrkAlarmStatus2	16 bit word for IO channels Mod.9 to 16	Same format as above			R/O
SbrkAlarmStatus3	16 bit word for IO channels Mod.17 to 24	Same format as above			R/O
SbrkAlarmStatus4	16 bit word for IO channels Mod.25 to 32	Same format as above			R/O
CTAlarmStatus1	16 bit word for CT alarms 1 to 5	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 9 Bit 10 Bit 11 Bit 12 Bit 13 Bit 14 Bit 15	Set if Load1 SSR fail Set if Load1 PLF Set If Load1 OCF Set if Load2 SSR fail Set if Load2 PLF Set If Load2 OCF Set if Load3 SSR fail Set if Load3 PLF Set If Load3 OCF Set if Load4 SSR fail Set if Load4 PLF Set If Load4 OCF Set if Load5 SSR fail Set if Load5 PLF Set If Load5 OCF -		R/O
CTAlarmStatus2	16 bit word for CT alarms 6 to 10	Bit 0 Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 Bit 7 Bit 8 Bit 9 Bit 10 Bit 11 Bit 12 Bit 13 Bit 14 Bit 15	Set if Load6 SSR fail Set if Load6 PLF Set If Load6 OCF Set if Load7 SSR fail Set if Load7 PLF Set If Load7 OCF Set if Load8 SSR fail Set if Load8 PLF Set If Load8 OCF Set if Load9 SSR fail Set if Load9 PLF Set If Load9 OCF Set if Load10 SSR fail Set if Load10 PLF Set If Load10 OCF -		R/O
CTAlarmStatus3	16 bit word for CT alarms 11 to 15	Same format as CTAlarmStatus1			R/O
CTAlarmStatus4	16 bit word for CT alarm 16	Same format as CTAlarmStatus1			R/O

8.8 Alarm Log

A list of the last 32 alarms to have occurred is maintained in an Alarm Log.

Folder: AlmSummary	**Sub-folder: AlmLog**			
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
ClearLog	Clear Alarm Log	Yes/No	No	Oper
Entry1Ident	Most recent alarm activation	All analogue alarms All digital alarms All sensor break alarms All current alarms	NoEntry	R/O
Entry1Day	The day the first entry activated	NoEntry, Monday/Tuesday…Sunday.	NoEntry	R/O
Entry1Time	The time the first entry activated	hh:mm:ss	0	R/O
Entry2Ident	2nd most recent alarm activation	All analogue alarms All digital alarms All sensor break alarms All current alarms	NoEntry	R/O
Entry2Day	The day the second entry activated	NoEntry, Monday/Tuesday…Sunday.	NoEntry	R/O
Entry2Time	The time the second entry activated	hh:mm:ss	0	R/O
…etc				
Entry32Ident	32nd most recent alarm activation	All analogue alarms All digital alarms All sensor break alarms All current alarms	NoEntry	R/O
Entry32Day	The day the 32nd entry activated	NoEntry, Monday/Tuesday…Sunday.	NoEntry	R/O
Entry32Time	The time the 32nd entry activated	hh:mm:ss	0	R/O

9. CHAPTER 9 BCD INPUT

The Binary Coded Decimal (BCD) input function block uses a number of digital inputs and combines them to make a numeric value. A very common use for this feature is to select a setpoint program number from panel mounted BCD decade switches.

The block uses 4 bits to generate a single digit.

Two groups of four bits are used to generate a two digit value (0 to 99)

The block outputs four results

1. Units Value: The BCD value taken from the first four bits (range 0 – 9)
2. Tens Value: The BCD value taken from the second four bits (range 0 – 9)
3. BCD Value: The combined BCD value taken from all 8 bits (range 0 – 99)
4. Decimal Value: The decimal numeric equivalent of Hexadecimal bits (range 0 – 255)

The following table shows how the input bits combine to make the output values.

<table>
<tr><td>Input 1</td><td rowspan="4">Units value (0 – 9)</td><td rowspan="8">BCD value (0 – 99)</td><td rowspan="8">Decimal value (0 – 255)</td></tr>
<tr><td>Input 2</td></tr>
<tr><td>Input 3</td></tr>
<tr><td>Input 4</td></tr>
<tr><td>Input 5</td><td rowspan="4">Tens value (0 – 9)</td></tr>
<tr><td>Input 6</td></tr>
<tr><td>Input 7</td></tr>
<tr><td>Input 8</td></tr>
</table>

Since the inputs cannot all be guaranteed to change simultaneously, the output will only update after all the inputs have been stable for two samples.

9.1 BCD Parameters

Folder – BCDInput		**Sub-Folders: 1 and 2**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
In 1	Digital Input 1	On or Off	Alterable from the operator interface if not wired	Off	Oper
In 2	Digital Input 2	On or Off		Off	Oper
In 3	Digital Input 3	On or Off		Off	Oper
In 4	Digital Input 4	On or Off		Off	Oper
In 5	Digital Input 5	On or Off		Off	Oper
In 6	Digital Input 6	On or Off		Off	Oper
In 7	Digital Input 7	On or Off		Off	Oper
In 8	Digital Input 8	On or Off		Off	Oper
Dec Value	Decimal value of the inputs	0 – 255	See examples below		R/O
BCD Value	Reads the value (in BCD) of the switch as it appears on the digital inputs	0 – 99	See examples below		
Units	Units value of the first switch	0 – 9	See examples below		R/O
Tens	Units value of the second switch	0 – 9	See examples below		R/O

In 1	In 2	In 3	In 4	In 5	In 6	In 7	In 8	Dec	BCD	Units	Tens
1	0	0	0	0	0	0	0	1	1	1	0
1	1	1	1	0	0	0	0	15	9	9	0
0	0	0	0	1	1	1	1	240	90	0	9
1	1	1	1	1	1	1	1	255	99	9	9

9.1.1 Example: To wire a BCD Input

The BCD digital input parameters may be wired to digital input terminals of the controller. There are two standard digital input terminals which may be used, D1 and D2.

10. CHAPTER 10 DIGITAL COMMUNICATIONS

Digital Communications (or 'comms' for short) allows the Mini8 to be part of a system by communicating with a PC or a programmable logic controller (PLC).

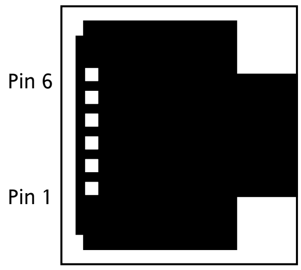
The Mini8 also has a configuration port for 'cloning' or saving/loading instrument configurations for future expansion of the plant or to allow you to recover a system after a fault.

10.1 Configuration Port

The configuration port is on an RJ11 socket, just to the right of the power supply connections. This will normally be connected to a personal computer running iTools. Eurotherm supply a standard cable to connect a serial COM port on a computer to the RJ11 socket, part no. **SubMini8/cable/config**.

This port conforms to MODBUS RTU ® protocol a full description of which can be found on www.modbus.org.

9 pin DF to PC COM port (RS232)	RJ11 Pin	Function
-	6	N/c
3 (Tx)	5	Rx
2 (Rx)	4	Tx
5 (0v)	3	0v (gnd)
	2	N/c
	1	Reserved



10.1.1 Configuration Communications Parameters

Folder - Comms		Sub-folders: CC (Config Comms)			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Ident	Identification of the module fitted.	Modbus always.		Modbus	R/O
Protocol	Digital communications protocol	MODBUS		MODBUS	R/O
Baud Rate	Communications baud rate	4800 9600 19k2 (19200)		19200	Conf
Parity	Communications parity	None Even Odd	No parity Even parity Odd parity	None	Conf
Address	Instrument address	1 to 254		1	Oper
Wait	Rx/tx wait states	No Yes	No delay Fixed delay. This inserts a delay between Rx and Tx to ensure that the drivers used by intelligent RS232/RS485 converters have sufficient time to switch over.	No	Conf

When connecting to iTools the instrument on this port will be found at address 255. iTools will also optimise the baud rate to suit the conditions.

This port can be used as a 'permanent' connection but it is limited to one instrument, it is a RS232 point to point connection.

Configuration is also possible through the Field Communications port but ONLY if that port is Modbus. In that situation the Mini8s can be multi-dropped to iTools.

10.2 Field Communications Port

The Min8 controller has a number of communication options. These have to be ordered from the factory as part of the instrument build. A change of protocol is not usually possible in the field. The physical port and the connections will vary depending on the field communications protocol. Mini8 version 1.04 offers Modbus and DeviceNet.

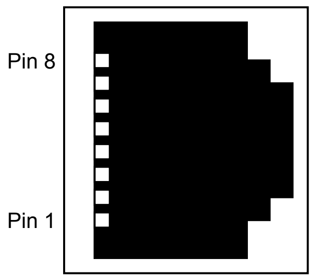
10.3 Modbus

This port conforms to MODBUS RTU ® protocol a full description of which can be found on www.modbus.org.

10.3.1 Modbus Connections

This uses two parallel RJ45 connectors for use with screened Cat5e patch cables. The connection is usually 2 wire but 4 wire is also available. This is selected by the top switch of the address switches below the RJ45 ports – OFF (to the left) 2 wire, ON (to the right) 4 wire.

RJ45 pin	3 wire	5 wire
8	Receive (RX+)	RxA
7	Transmit (TX+)	RxB
6	Common	Ground
5		
4		
3	Ground	Ground
2	D+	TxA
1	D-	TxB



10.3.2 Modbus Communications Parameters

The following table shows the parameters available.

Folder – Comms		**Sub-folder: FC (Field Communications**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Ident	Identification of the module fitted.	Comms		Modbus	R/O
Protocol	Digital communications protocol	MODBUS		MODBUS	
Baud Rate	Communications baud rate	4800 9600 19k2 (19200)		9600	Conf
Parity	Communications parity	None Even Odd	No parity Even parity Odd parity	None	Conf
Address	Instrument address	1 to 254		1	Oper
Resolution	Communications resolution	Full Int	Full Integer	Full	Oper
Network Status	Network Status	For Profibus and DeviceNet only. Displays status of the network and connection			R/O
Comms Delay	Rx/tx delay time	No Yes	No delay Fixed delay. This inserts a delay between Rx and Tx to ensure that the drivers used by intelligent RS232/RS485 convertors have sufficient time to switch over.	No	Conf
Broadcast Enabled	To enable broadcast master communications. (See 10.4)	No Yes	Not enabled Enabled	No	
Broadcast Address ⚠	Address of the parameter being written to slaves.	0 to 32767	See Appendix A for addresses of all Mini8 parameters.		
Broadcast Value	Value to be sent to instruments on the network. This would normally be wired to a parameter within the master	Range of the parameter wired. In the case of a Boolean the value will be 0 or 1.			

10.3.3 Communications Identity

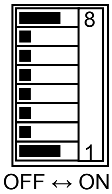
The instrument recognizes the type of communication board fitted. The identity **'id'** displayed so that you can verify that the instrument is built to your requirement.

10.3.4 Modbus Address Switch

On a network of instruments an address is used to specify a particular instrument. Each instrument on a network MUST have a unique address. Address 255 is reserved for configuration using the configuration port or the configuration clip

The switch is situated at the bottom of the Comms module. The switch gives addresses from 1 to 31. If Address 0 is set the Mini8 will then take the address and parity settings entered in the configuration of the instrument, see folder above. This allows for addresses above 31.

Sw	OFF	ON
8	3 wire	5 wire
7	NO Parity	Parity
6	Even	Odd
5	-	Address 16
4	-	Address 8
3	-	Address 4
2	-	Address 2
1	-	Address 1



Example shows 5 wire and address 1

10.3.5 Baud Rate

The baud rate of a communications network specifies the speed that data is transferred between instrument and master. A baud rate of 9600 equates to 9600 Bits per second. Since a single character requires 8 bits of data plus start, stop, and optional parity, up to 11 bits per byte may be transmitted. 9600 baud equates approximately to 1000 Bytes per second. 4800 baud is half the speed – approx. 500 Bytes per second.

In calculating the speed of communications in your system it is often the Latency between a message being sent and a reply being started that dominates the speed of the network.

For example, if a message consists of 10 characters (10msec at 9600 Baud) and the reply consists of 10 characters, then the transmission time would be 20 msec. However, if the Latency is 20msec, then the transmission time has become 40msec.

10.3.6 Parity

Parity is a method of ensuring that the data transferred between devices has not been corrupted.

Parity is the lowest form of integrity in the message. It ensures that a single byte contains either an even or an odd number of ones or zero in the data.

In industrial protocols, there are usually layers of checking to ensure that the first byte transmitted is good. Modbus applies a CRC (Cyclic Redundancy Check) to the data to ensure that the package is correct.

10.3.7 RX/TX Delay Time

In some systems it is necessary to introduce a delay between the instrument receiving a message and its reply. This is sometimes caused by communications converter boxes which require a period of silence on the transmission to switch over the direction of their drivers.

10.4 Modbus Broadcast Master Communications

Broadcast master communications will to allow the Mini8 series controllers to send a single value to any slave instruments using a Modbus broadcast using function code 6 (Write single value). This allows the Mini8 to link through digital communications with other products without the need for a supervisory PC to create a small system solution.

Example applications include multi-zone profiling applications or cascade control using a second controller. The facility provides a simple and precise alternative to analogue retransmission.

⚠ **Warning**

When using broadcast master communications, be aware that updated values are sent many times a second. Before using this facility, check that the instrument to which you wish to send values can accept continuous writes. ***Note that in common with many third party lower cost units, the Eurotherm 2200 series and the 3200 series prior to version V1.10 do not accept continuous writes to the temperature setpoint. Damage to the internal non-volatile memory could result from the use of this function. If in any doubt, contact the manufacturer of the device in question for advice.***

When using the 3200 series fitted software version 1.10 and greater, use the Remote Setpoint variable at Modbus address 26 if you need to write to a temperature setpoint. This has no write restrictions and may also have a local trim value applied. There is no restriction on writing to the 2400 or Mini8 series.

10.4.1 Mini8 Broadcast Master

The Mini8 broadcast master can be connected to up to 31 slaves if no segment repeaters are used. If repeaters are used to provide additional segments, 32 slaves are permitted in each new segment. The master is configured by selecting a Modbus register address to which a value is to be sent. The value to send is selected by wiring it to the Broadcast Value. Once the function has been enabled, the instrument will send this value out over the communications link every control cycle typically every 110ms.

Notes:-

1. The parameter being broadcast must be set to the same decimal point resolution in both master and slave instruments.
2. If iTools, or any other Modbus master, is connected to the port on which the broadcast master is enabled, then the broadcast is temporarily inhibited. It will restart approximately 30 seconds after iTools is removed. This is to allow reconfiguration of the instrument using iTools even when broadcast master communications is operating.

A typical example might be a multi zone application where the setpoint of each zone is required to follow, with digital accuracy, the setpoint of a master.

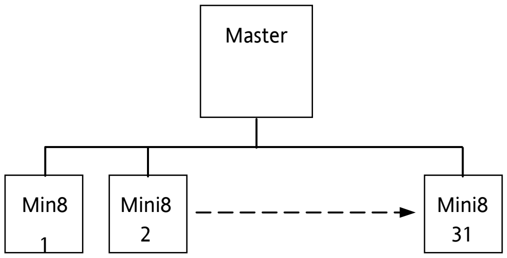


Figure 10-1: Broadcast Comms

10.4.2 Wiring Connections

The Digital Communications module for the master must be the Field Comms and is only RS485/RS422. RS232 is not available.

The Digital Communications module for the slave can be the Config port (RS232 only) or the Field Comms port (Not RS232)..

Standard patch cables cannot be used, as the connections do not 'cross over.' Wire using twisted pair(s) cable and crimp on the appropriate RJ45 or RJ11 plug.

RS485 2-wire

Connect A (+) in the master to A (+) of the slave

Connect B (-) in the master to B (-) of the slave

This is shown diagrammatically below

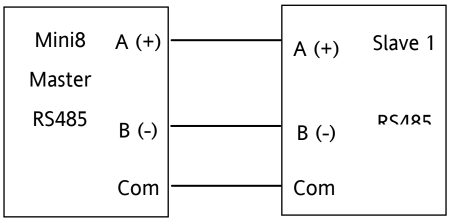


Figure 10-2: Rx/Tx Connections RS485 2-wire

RS422, RS485 4-wire

Rx connections in the master are wired to Tx connections of the slave

Tx connections in the master are wired to Rx connections of the slave

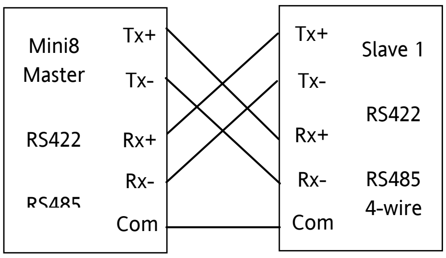
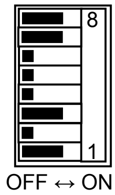


Figure 10-3: Rx/Tx Connections for RS422, RS485 4-wire

10.5 DeviceNet

Only 2 parameters have to be set on the Mini8 for use with DeviceNet, baud rate and address. Both can be set on the hardware address switch situated under the DeviceNet connector. Each Mini8 must have a unique address on the DeviceNet network and all units must be set to the same Baud rate. The switch gives addresses from 0 to 63.

Sw	OFF	ON
8	Baud rate	Baud rate
7	Baud rate	Baud rate
6	-	Address 32
5	-	Address 16
4	-	Address 8
3	-	Address 4
2	-	Address 2
1	-	Address 1



Example shows 500k baud rate and address 5

Sw	Baud rate		
	125k	250k	500k
8	OFF	OFF	ON
7	OFF	ON	ON

Use 500k unless the total length of the DeviceNet network is longer than 100m.

In iTools the DeviceNet Network Status is available and will return the following status:

Offline: No DeviceNet traffic detected

Ready: DeviceNet traffic detected but not for this address

Running: DeviceNet traffic detected addressing this instrument.

11. CH. 11 COUNTERS, TIMERS, TOTALISERS, RT CLOCK

A series of function blocks are available which are based on time/date information. These may be used as part of the control process.

11.1 Counters

Up to two counters are available. They provide a synchronous edge triggered event counter.

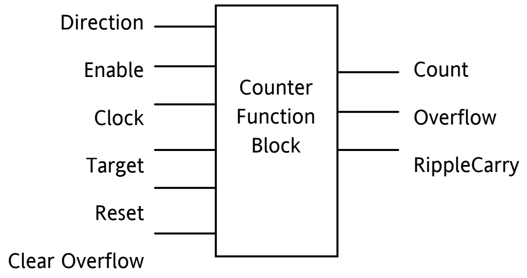


Figure 11-1: Counter Function Block

When configured as an Up counter, Clock events increment Count until reaching the Target. On reaching Target RippleCarry is set true. At the next clock pulse, Count returns to zero. Overflow is latched true and RippleCarry is returned false.

When configured as a down counter, Clock events decrement Count until it reaches zero. On reaching zero RippleCarry is set true. At the next clock pulse, Count returns to the Target count. Overflow is latched true and RippleCarry is reset false

Counter blocks can be cascaded as shown in the diagram below

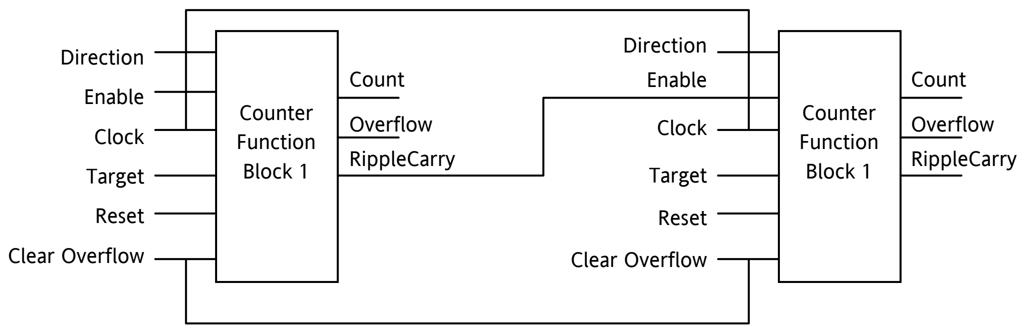


Figure 11-2: Cascading Counters

The RippleCarry output of one counter acts as an enabling input for the next counter. In this respect the next counter in sequence can only detect a clock edge if it was enabled on the previous clock edge. This means that the Carry output from a counter must lead its Overflow output by one clock cycle. The Carry output is, therefore, called a RippleCarry as it is NOT generated on an Overflow (i.e. Count $\geq$ Target) but rather when the count reaches the target (i.e. Count = Target). The timing diagram below illustrates the principle for the Up Counter.

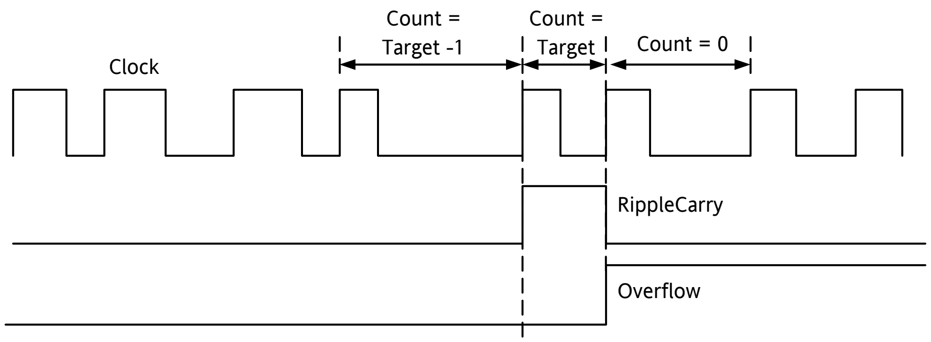


Figure 11-3: Timing Diagram for an Up Counter

11.1.1 Counter Parameters

Folder - Counter		**Sub-folders: 1 to 2**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Enable	Counter enable. Counter 1 or 2 is enabled in the Instrument Options folder but they can also be turned on or off in this list	Yes No	Enabled Disabled	No	Oper
Direction	Defines count up or count down. This is not intended for dynamic operation (i.e. subject to change during counting). It can only be set in configuration level.	Up Down	Up counter Down counter	Up	Conf
Ripple Carry	Ripple carry to act as an enabling input to the next counter. It is turned On when the counter reaches the target set	Off			R/O
Overflow	Overflow flag is turned on when the counter reaches zero				R/O
Clock	Tick period to increment or decrement the count. This is normally wired to an input source such as a digital input.	0 1	No clock input Clock input present	0	R/O if wired
Target	Level to which the counter is aiming	0 to 99999		9999	Oper
Count	Counts each time a clock input occurs until the target is reached.	0 to 99999			R/O
Reset	Resets the counter	No Yes	Not in reset Reset	No	Oper
Clear Overflow	Clear overflow flag	No Yes	Not cleared Cleared	No	Oper

11.2 Timers

Up to four timers can be configured. Each one can be configured to a different type and can operate independently of one another.

11.2.1 Timer Types

Each timer block can be configured to operate in four different modes. These modes are explained below

11.2.2 On Pulse Timer Mode

This timer is used to generate a fixed length pulse from an edge trigger.

- The output is set to On when the input changes from Off to On.
- The output remains On until the time has elapsed
- If the 'Trigger' input parameter recurs while the Output is On, the Elapsed Time will reset to zero and the Output will remain On
- The triggered variable will follow the state of the output

The diagram illustrates the behaviour of the timer under different input conditions.

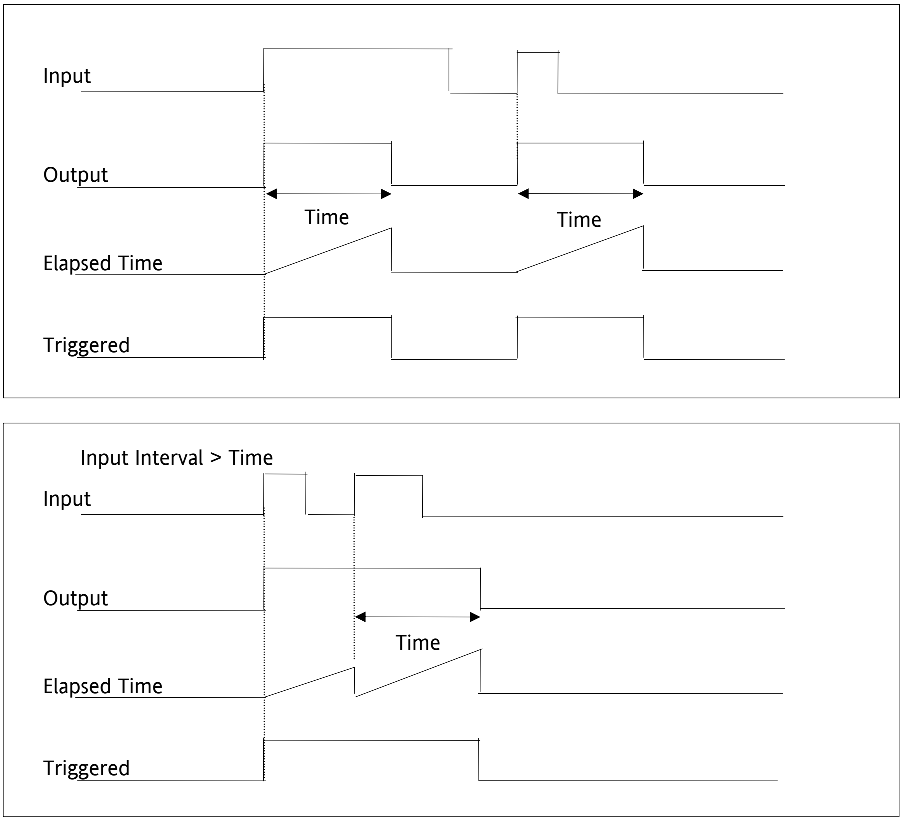


Figure 11-4: On Pulse Timer Under Different Input Conditions

11.2.3 Off Delay Timer Mode

This timer provides a delay between the trigger event and the Timer output. If a short pulse triggers the Timer, then a pulse of one sample time (110ms) will be generated after the delay time.

- The Output is set to Off when the Input changes from Off to On.
- The Output remains Off until the Time has elapsed.
- If the Input returns to Off before the time has elapsed, the Timer will continue until the Elapsed Time equals the Time. It will then generate a pulse of one Sample Time duration.
- Once the Time has elapsed, the Output will be set to On.
- The Output will remain On until the Input is cleared to Off.
- The Triggered variable will be set to On by the Input changing from Off to On. It will remain On until both the Time has elapsed and the Output has reset to Off.

The diagram illustrates the behaviour of the timer under different input conditions.

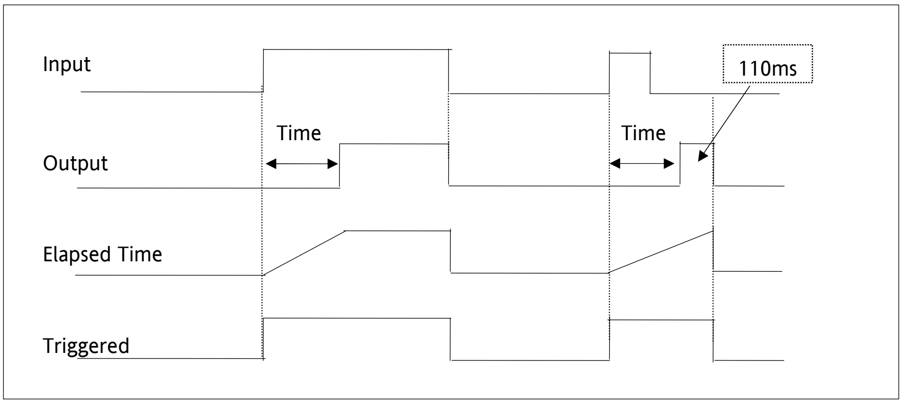


Figure 11-5: Off Delay Timer Under Different Input Conditions

11.2.4 One Shot Timer Mode

This timer behaves like a simple oven timer.

- When the Time is edited to a non-zero value the Output is set to On
- The Time value is decremented until it reaches zero. The Output is then cleared to Off
- The Time value can be edited at any point to increase or decrease the duration of the On time
- Once set to zero, the Time is not reset to a previous value, it must be edited by the operator to start the next On-Time
- The Input is used to gate the Output. If the Input is set, the time will count down to zero. If the Input is cleared to Off, then the Time will hold and the Output will switch Off until the Input is next set.

Note: since the Input is a digital wire, it is possible for the operator to NOT wire it, and set the Input value to On which permanently enables the timer.

- The Triggered variable will be set to On as soon as the Time is edited. It will reset when the Output is cleared to Off.

The behaviour of the timer under different input conditions is shown below.

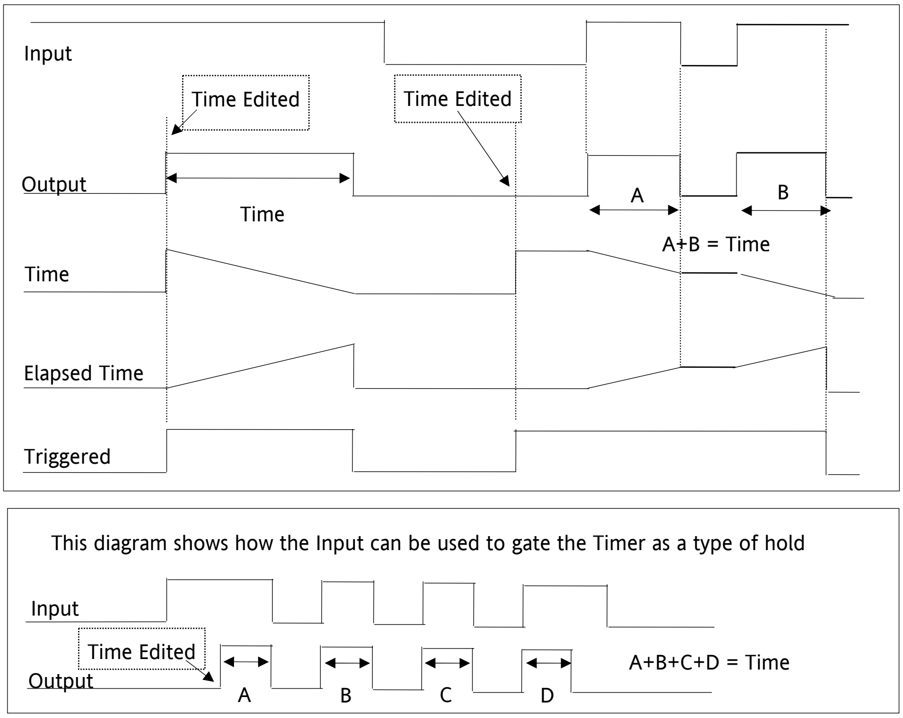


Figure 11-6: One Shot Timer

11.2.5 Compressor or Minimum On Timer Mode

This timer has been targeted at guaranteeing that the output remains On for a duration after the input signal has been removed. It may be used, for example, to ensure that a compressor is not cycled excessively.

- The output will be set to On when the Input changes from Off to On.
- When the Input changes from On to Off, the elapsed time will start incrementing towards the set Time.
- The Output will remain On until the elapsed time has reached the set Time. The Output will then switch Off.
- If the Input signal returns to On while the Output is On, the elapsed time will reset to 0, ready to begin incrementing when the Input switches Off.
- The Triggered variable will be set while the elapsed time is >0. It will indicate that the timer is counting.

The diagram illustrates the behaviour of the timer under different input conditions.

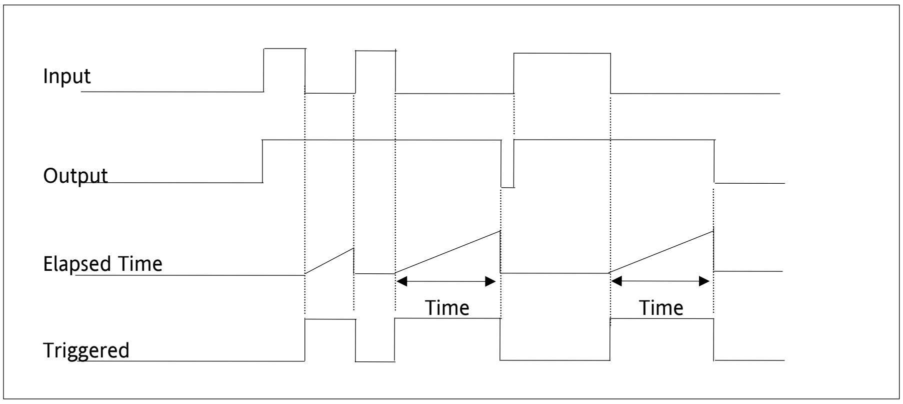


Figure 11-7: Minimum On Timer Under Different Input Conditions

11.2.6 Timer Parameters

Folder – Timer		Sub-folders: 1 to 4			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Type	Timer type	Off	Timer not configured	Off	Conf
		On Pulse	Generates a fixed length pulse from an edge trigger		
		Off Delay	Provides a delay between input trigger event and timer putput		
		One Shot	Simple oven timer which reduces to zero before switching off		
		Min-On Ti	Compressor timer guaranteeing that the output remains ON for a time after the input signal has been removed		
Time	Duration of the timer. For re-trigger timers this value is entered once and copied to the time remaining parameter whenever the timer starts. For pulse timers the time value itself is decremented.	0:00.0 to 99:59:59		0:00.0	Oper
Elapsed Time	Timer elapsed time	0:00.0 to 99:59:59			R/O
In	Trigger/Gate input. Turn On to start timing	Off On	Off Start timing	Off	Oper
Out	Timer output	Off On	Output off Timer has timed out		R/O
Triggered	Timer triggered (timing). This is a status output to indicate that the timers input has been detected	Off On	Not timing Timer timing		R/O

The above table is repeated for Timers 2 to 4.

11.3 Totalisers

There are two totaliser function blocks which are used to measure the total quantity of a measurement integrated over time. A totaliser can, by soft wiring, be connected to any measured value. The outputs from the totaliser are its integrated value and an alarm state. The user may set a setpoint which causes the alarm to activate once the integration exceeds the setpoint.

The totaliser has the following attributes:-

1. Run/Hold/Reset

In Run the totaliser will integrate its input and continuously test against an alarm setpoint.

In Hold the totaliser will stop integrating its input but will continue to test for alarm conditions.

In Reset the totaliser will be zeroed, and alarms will be reset.

2. Alarm Setpoint

If the setpoint is a positive number, the alarm will activate when the total is greater than the setpoint.

If the setpoint is a negative number, the alarm will activate when the total is lower (more negative) than the setpoint.

If the totaliser alarm setpoint is set to 0.0, the alarm will be off. It will not detect values above or below.

The alarm output is a single state output. It may be cleared by resetting the totaliser, or by changing the alarm setpoint.

3. Limits

The total is limited to a maximum of 99999 and a minimum of -19999.

4. Resolution

The totaliser ensures that resolution is maintained when integrating small values onto a large total.

11.3.1 Totaliser Parameters

Folder – Total		Sub-Folders: 1 to 2			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
TotalOut	The totalised value	99999 t o-19999			R/O
In	The value to be totalised	-9999.9 to 9999.9. Note:- the totaliser stops accumulating if the input is 'Bad'.			Oper
Units	Totaliser units	None AbsTemp V, mV, A, mA, PH, mmHg, psi, Bar, mBar, %RH, %, mmWG, inWG, inWW, Ohms, PSIG, %O2, PPM, %CO2, %CP, %/sec, RelTemp mBar/Pa/T sec, min, hrs,			Conf
Resolution	Totaliser resolution	XXXXX XXXX.X XXX.XX XX.XXX X.XXXX		XXXXX	Conf
Alarm SP	Sets the totalised value at which an alarm will occur	-99999 to 99999			Oper
AlarmOut	This is a read only value which indicates the alarm output On or Off. The totalised value can be a positive number or a negative number. If the number is positive the alarm occurs when Total > + Alarm Setpoint If the number is negative the alarm occurs when Total > - Alarm Setpoint	Off On	Alarm inactive Alarm output active	Off	Oper
Run	Runs the totaliser	No Yes	Timer not running Select Yes to run the timer	No	Oper
Hold	Holds the totaliser at its current value Note: The Run & Hold parameters are designed to be wired to (for example) digital inputs. Run must be 'on' and Hold must be 'off' for the totaliser to operate.	No Yes	Timer not in hold Hold timer	No	Oper
Reset	Resets the totaliser	No Yes	Timer not in reset Timer in reset	No	Oper

11.4 Real Time Clock

A real time clock (day of week and time only) is used to provide a daily and weekly scheduling facility and provides two corresponding outputs. The configuration for an output is an On-Day and an On-Time and an Off-Day and an Off-Time.

The day options supported are:-

Day Option	Description
Never	Disables the output feature
Monday	Output will only be available on a Monday
Tuesday	Output will only be available on a Tuesday
Wednesday	Output will only be available on a Wednesday
Thursday	Output will only be available on a Thursday
Friday	Output will only be available on a Friday
Saturday	Output will only be available on a Saturday
Sunday	Output will only be available on a Sunday
Mon-Fri	Output will only be available between Monday to Friday
Mon-Sat	Output will only be available on between Monday to Saturday
Sat-Sun	Output will only be available on between Saturday to Sunday
Everyday	Output always available

For example, it is possible to configure an output to be activated at 07:30 on Monday and deactivated at 17:15 on Friday

The output from the Real Time Clock outputs may be used to place the instrument in standby or to sequence a batch process.

The Real Time Clock function will set/clear the output outputs only at the time of the output. Therefore, it is possible to manually override the outputs by editing the output to On/Off between output activations.

The Real Time Clock does not display date or year.

11.4.1 Real Time Clock Parameters

Folder – RTClock		Sub Folders: None			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Mode	This parameter can be used to set the clock	Running Edit Stopped	Normal operation Allows the clock to be set Clock stopped (saves battery life)	Stopped	Oper
Day	Displays the day or allows the day to be set when in Edit mode	Monday to Sunday			Oper
Time	Displays the time or allows the time to be set when in Edit mode	00:00:00 to 23:59:59			Oper
On Day1 On Day2	Days when output 1 and 2 are activated	See table above			Oper
On Time1 On Time2	Time of day when output 1 and 2 are activated	00:00:00 to 23:59:59			Oper
Off Day1 Off Day2	Days when output 1 and 2 are de-activated	See table above			Oper
Off Time1 Off Time2	Time of day when output 1 and 2 are de-activated	00:00:00 to 23:59:59			Oper
Out1 Out2	Output 1 and 2	Off On	Output not activated Output activated		Oper

12. CHAPTER 12 HUMIDITY CONTROL

12.1.1 Overview

Humidity (and altitude) control is a standard feature of the Mini8 controller. In these applications the controller may be configured to generate a setpoint profile (see Chapter 18 'Programmer Operation').

Also the controller may be configured to measure humidity using either the traditional Wet/Dry bulb method or it may be interfaced to a solid state sensor.

The controller output may be configured to turn a refrigeration compressor on and off, operate a bypass valve, and possibly operate two stages of heating and/or cooling

12.1.2 Temperature Control of an Environmental Chamber

The temperature of an environmental chamber is controlled as a single loop with two control outputs. The heating output time proportions electric heaters, usually via a solid state relay. The cooling output operates a refrigerant valve which introduces cooling into the chamber. The controller automatically calculates when heating or cooling is required.

12.1.3 Humidity Control of an Environmental Chamber

Humidity in a chamber is controlled by adding or removing water vapour. Like the temperature control loop two control outputs are required, i.e. Humidify and Dehumidify.

To humidify the chamber water vapour may be added by a boiler, an evaporating pan or by direct injection of atomised water.

If a boiler is being used adding steam increases the humidity level. The humidify output from the controller regulates the amount of steam from the boiler that is allowed into the chamber.

An evaporating pan is a pan of water warmed by a heater. The humidify output from the controller humidity regulates the temperature of the water.

An atomisation system uses compressed air to spray water vapour directly into the chamber. The humidify output of the controller turns on or off a solenoid valve.

Dehumidification may be accomplished by using the same compressor used for cooling the chamber. The dehumidify output from the controller may control a separate control valve connected to a set of heat exchanger coils.

12.2 Humidity Parameters

List Folder - Humidity		Sub-folder: None			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Resolution	Resolution of the relative humidity	XXXXX XXXX.X XXX.XX XX.XXX X.XXXX			Conf
Psychro Const	The psychrometric constant at a given pressure (6.66E-4 at standard atmospheric pressure). The value is dependent on the speed of air-flow across the wet bulb, and hence the rate of evaporation. 6.66E-4 is for the ASSMANN ventilated Psychrometer.	0.0 to 10.0		6.66	Oper
Pressure	Atmospheric Pressure	0.0 to 2000.0		1013.0 mbar	Oper
WetTemp	Wet Bulb Temperature	Range units			
WetOffset	Wet bulb temperature offset	-100.0 to 100.0		0.0	Oper
DryTemp	Dry Bulb Temperature	Range units			
RelHumid	Relative Humidity is the ratio of actual water vapour pressure (AVP) to the saturated water vapour pressure (SVP) at a particular temperature and pressure	0.0 to 100.0		100	R/O
DewPoint	The dew point is the temperature to which air would need to cool (at constant pressure and water vapour content) in order to reach saturation	-999.9 to 999.9			R/O
Sbrk	Indicates that one of the probes is broken.	No Yes	No sensor break detection Sensor break detection enabled		Conf

13. CHAPTER 13 INPUT MONITOR

There are two Input monitors. Each input monitor may be wired to any variable in the controller. It then provides three functions:-

1. Maximum detect
2. Minimum detect
3. Time above threshold

13.1.1 Maximum Detect

This function continuously monitors the input value. If the value is higher than the previously recorded maximum, it becomes the new maximum.

This value is retained following a power fail.

13.1.2 Minimum Detect

This function continuously monitors the input value. If the value is lower than the previously recorded minimum, it becomes the new minimum.

This value is retained following a power fail.

13.1.3 Time Above Threshold

This function increments a timer whenever the input is above a threshold value. If the timer exceeds 24 hours per day, a counter is incremented. The maximum number of days is limited to 255. A time alarm can be set on the timer so that once the input has been above a threshold for a period, an alarm output is given.

Applications include:-

- Service interval alarms. This sets an output when the system has been running for a number of days (up to 90 years)
- Material stress alarms - if the process cannot tolerate being above a level for a period. This is a style of 'policeman' for processes where the high operating point degrades the life of the machine.
- In internal wiring applications in the controller

13.2 Input Monitor Parameters

Folder - IPMonitor		**Sub-Folders: 1 or 2**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
In	The input value to be monitored	May be wired to an input source. The range will depend on the source		Oper R/O if wired
Max	The maximum measured value recorded since the last reset	As above		R/O
Min	The minimum measured value recorded since the last reset	As above		R/O
Threshold	The input timer accumulates the time the input PV spends above this trigger value.	As above		Oper
Days Above	Accumulated days the input has spent above threshold since the last reset.	Days is an integer count of the 24 hour periods only. The Days value should be combined with the Time value to make the total time above threshold.		R/O
Time Above	Accumulated time above the 'Threshold' since last reset.	The time value accumulates from 00:00.0 to 23:59.9. Overflows are added to the days value		R/O
AlarmDays	Days threshold for the monitors time alarm. Used in combination with the Alarm Time parameter. The 'Out' is set to true if the inputs accumulated time above threshold is higher than the timer high parameters.	0 to 255	0	Oper
AlarmTime	Time threshold for the monitors time alarm. Used in combination with the Alarm Days parameter. The 'Out' is set to true if the inputs accumulated time above threshold is higher than the timer high parameters.	0:00.0 to 99:59:59	0:00.0	Oper
Out	Set true if the accumulated time that the input spends above the trigger value is higher than the alarm threshold.	Off — Normal operation On — time above setpoint exceeded		R/O
Reset	Resets the Max and Min values and resets the time above threshold to zero.	No — Normal operation Yes — Reset values	No	Oper
In Status	Monitors the status of the input	Good — Normal operation Bad — The input may be incorrectly wired		R/O Oper

14. CHAPTER 14 LOGIC AND MATHS OPERATORS.

14.1 Logic Operators

Logic Operators allow the controller to perform logical calculations on **two** input values. These values can be sourced from any available parameter including Analogue Values, User Values and Digital Values.

The parameters to use, the type of calculation to be performed, input value inversion and 'fallback' value are determined in Configuration level.

The Logic Operators folder is only available if the operators have been enabled in **'Instrument'** folder sub-folder **'Options'**.

There are 24 separate calculations – they do not have to be in sequence. When logic operators are enabled a Folder **'Lgc2'** exists where the 2 denotes two input logic operators.

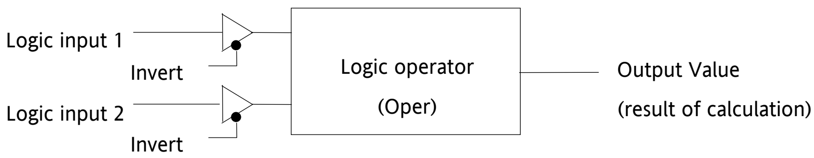


Figure 14-1: 2 Input Logic Operators

Logic Operators are found under the folder **'Lgc2'**. Note that the logic operators can also be enable by dragging a block onto the graphical wiring screen in iTools.

14.1.1 Logic 8

Logic 8 operators can perform logic calculations on up to **eight** inputs. The calculations are limited to AND,OR,XOR. Up to two 8 input operators can be enabled. The folder is labelled **'Lgc8'** to denote eight input logic operators.

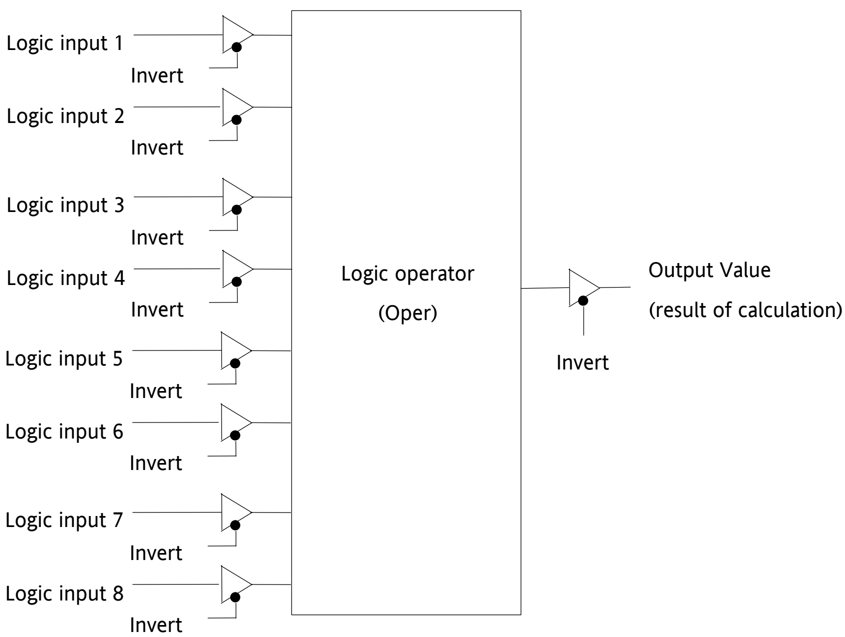


Figure 14-2: 8 Input Logic Operators

14.1.2 2 input Logic Operations

The following calculations can be performed:

Oper	Operator description	Input 1	Input 2	Output Invert = None
0: OFF	The selected logic operator is turned off			
1: AND	The output result is ON when both Input 1 and Input 2 are ON	0 1 0 1	0 0 1 1	Off Off Off On
2: OR	The output result is ON when either Input 1 or Input 2 is ON	0 1 0 1	0 0 1 1	Off On On Off
3: XOR	Exclusive OR. The output result is true when one and only one input is ON. If both inputs are ON the output is OFF.	0 1 0 1	0 0 1 1	Off On On Off
4: Latch	Input 1 sets the latch, Input 2 resets the latch.	0 1 0 1	0 0 1 1	
5: Equal (==)	The output result is ON when Input 1 = Input 2	0 1 0 1	0 0 1 1	On Off Off On
6: Not equal (<>)	The output result is ON when Input 1 = Input 2	0 1 0 1	0 0 1 1	Off On Off On
7: Greater than (>)	The output result is ON when Input 1 > Input 2	0 1 0 1	0 0 1 1	Off On Off Off
8: Less than (<)	The output result is ON when Input 1 < Input 2	0 1 0 1	0 0 1 1	Off Off On Off
9: Equal to or Greater than (=>)	The output result is ON when Input 1 ≥ Input 2	0 1 0 1	0 0 1 1	On On Off On
10: Less than or Equal to (<=)	The output result is ON when Input 1 ≤ Input 2	0 1 0 1	0 0 1 1	On Off On On

Note 1: The numerical value is the value of the enumeration

Note 2: For options 1 to 4 an input value of less than 0.5 is considered false and greater than or equal to 0.5 as true.

14.1.3 Logic Operator Parameters

Folder – Lgc2 (2 Input Operators)		Sub-Folders: 1 to 24			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Oper	To select the type of operator	See previous table		None	Conf
In1	Input 1	Normally wired to a logic, analogue or user value. May be set to a constant value if not wired.		0	OPER
In2	Input 2				
FallbackType	The fallback state of the output if one or both of the inputs is bad	0: FalseBad 1: TrueBad 2: FalseGood 3: TrueGood	The output value is FALSE and the status is GOOD. The output value is FALSE and the status is BAD The output value is TRUE and the status is GOOD The output value is TRUE and the status is BAD.		Conf
Invert	The sense of the input value, may be used to invert one or both of the inputs	0: None 1: Input1 2: Input2 3: Both	Neither input inverted Invert input 1 Invert input 2 Invert both inputs		Conf
Out	The output from the operation is a boolean (true/false) value.	On Off	Output activated Output not activated		R/O
Status	The status of the result value	Good Bad			R/O

14.2 Eight Input Logic Operators

The eight input logic operator may be used to perform the following operations on eight inputs.

Oper	Operator description	
0: OFF	The selected logic operator is turned off	
1: AND	The output result is ON when ALL eight inputs are ON	
2: OR	The output result is ON when one or more of the 8 inputs are ON	
3: XOR	Exclusive OR. The output result is true when one and only one of the 8 inputs is ON.	

Eight Input Logic Operator Parameters

Folder – Lgc8 (8 Input Operators)		Sub-Folders: 1 to 2			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Oper	To select the type of operator	0: OFF 1: AND 2: OR 3: XOR	Operator turned off Output ON when all inputs are ON Output ON when one input is ON Exclusive OR	OFF	Conf
NumIn	This parameter is used to configure the number of inputs for the operation	1 to 8		2	Conf
InInvert	Used to invert selected inputs prior to operation. This is a status word with one bit per input, the left hand bit inverts input 1.	The invert parameter is interpreted as a bitfield where: 1 (0x1) - input 1 2 (0x2) - input 2 4 (0x4) - input 3 8 (0x8) - input 4 16 (0x10) - input 5 32 (0x20) - input 6 64 (0x40)- input 7 128 (0x80)- input 8 (e.g. 255 = all eight)		0	Oper
Out Invert	Invert the output	No Yes	Output not inverted Output inverted	No	Oper
In1 to In8	Input state 1 to 8	Normally wired to a logic, analogue or user value. When wired to a floating point, values less than or equal to –0.5 or greater than or equal to 1.5 will be rejected (e.g. the value of the lgc8 block will not change). Values between –0.5 and 1.5 will be interpreted as ON when greater than or equal to 0.5 and OFF when less than 0.5. May be set to a constant value if not wired.		Off	Oper
Out	Output result of the operator	On Off	Output activated Output not activated		R/O

14.3 Maths Operators

Maths Operators (sometimes known as Analogue Operators) allow the controller to perform mathematical operations on two input values. These values can be sourced from any available parameter including Analogue Values, User Values and Digital Values. Each input value can be scaled using a multiplying factor or scalar.

The parameters to use, the type of calculation to be performed and the acceptable limits of the calculation are determined in Configuration level. In normal operation the values of each of the scalars may be changed via communications or iTools.

There are 24 separate calculations – they do not have to be in sequence. When maths operators are enabled (in Instrument/Options folder) a Folder **'Math2'** exists (where the 2 denotes two input maths operators).

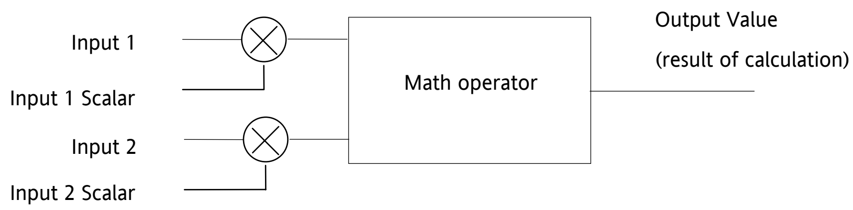
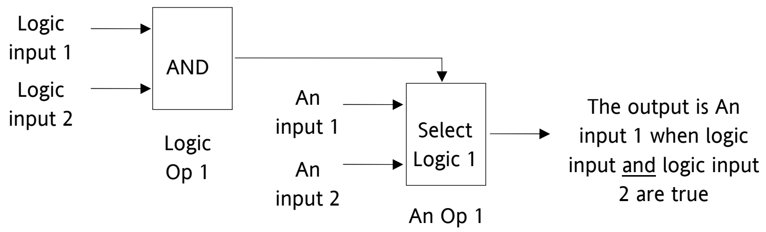


Figure 14-3: 2 Input Math Operators

8 input multiplexers are also available and are described in the next section.

14.3.1 Math Operations

The following operations can be performed:

0: Off	The selected analogue operator is turned off
1: Add	The output result is the addition of Input 1 and Input 2
2: Subtract (Sub)	The output result is the difference between Input 1 and Input 2 where Input 1 > Input 2
3: Multiply (Mul)	The output result is the Input 1 multiplied by Input 2
4: Divide (Div)	The output result is Input 1 divided by Input 2
5: Absolute Difference (AbsDif)	The output result is the absolute difference between Input 1 and 2
6: Select Max (SelMax)	The output result is the maximum of Input 1 and Input 2
7: Select Min (SelMin)	The output result is the minimum of Input 1 and Input 2
8: Hot Swap (HotSwp)	Input 1 appears at the output provided input 1 is 'good'. If input 1 is 'bad' then input 2 value will appear at the output. An example of a bad input occurs during a sensor break condition.
9: Sample and Hold (SmpHld)	Normally input 1 will be an analogue value and input B will be digital. The output tracks input 1 when input 2 = 1 (Sample). The output will remain at the current value when input 2 = 0 (Hold). If input 2 is an analogue value then any non zero value will be interpreted as 'Sample'.
10: Power	The output is the value at input 1 raised to the power of the value at input 2. I.e. input $1^{input\ 2}$
11: Square Root (Sqrt)	The output result is the square root of Input 1. Input 2 has no effect.
12: Log	The output is the logarithm (base 10) of Input 1. Input 2 has no effect
13: Ln	The output is the logarithm (base n) of Input 1. Input 2 has no effect
14: Exp	The output result is the exponential of Input 1. Input 2 has no effect
15: 10 x	The output result is 10 raised to the power of Input 1 value. I.e. $10^{input\ 1}$. Input 2 has no effect
51: Select	Any logic value may be used to control which Analogue Input is switched to the output of the Analogue Operator. If the output from the logic operator is true input 1 is switched through to the output. If false input 2 is switched through to the output. See example below:- 

When Boolean parameters are used as inputs to analogue wiring, they will be cast to 0.0 or 1.0 as appropriate. Values <= -0.5 or >= 1.5 will not be wired. This provides a way to stop a Boolean updating. Analogue wiring (whether simple re-routing or involving calculations) will always output a real type result, whether the inputs were booleans, integers or reals.

Note: The numerical value is the value of the enumeration

14.3.2 Math Operator Parameters

Folder – Math2 (2 Input Operators)		Sub-Folders: 1 to 24			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Oper	To select the type of operator	See previous table		None	Conf
In1Mul	Scaling factor on input 1	Limited to max float *		1.0	Oper
In2 Mul	Scaling factor on input 2	Limited to max float *		1.0	Oper
Units	Units applicable to the output value	None AbsTemp V, mV, A, mA, PH, mmHg, psi, Bar, mBar, %RH, %, mmWG, inWG, inWW, Ohms, PSIG, %O2, PPM, %CO2, %CP, %/sec, RelTemp mBar/Pa/T sec, min, hrs,		None	Conf
Resolution	Resolution of the output value	XXXXX. XXXX.X, XXX.XX, XX.XXX, X.XXXX			Conf
LowLimit	To apply a low limit to the output	Max float* to High limit (decimal point depends on resolution)			Conf
HighLimit	To apply a high limit to the output	Low limit to Max float* (decimal point depends on resolution)			Conf
Fallback	The state of the Output and Status parameters in case of a fault condition. This parameter could be used in conjunction with fallback value	Clip Bad Clip Good Fall Bad Fall Good Upscale DownScale	Descriptions, see section 3.4.5.		Conf
Fallback Val	Defines (in accordance with Fallback) the output value during fault conditions.	Limited to max float * (decimal point depends on resolution)			Conf
In1	Input 1 value (normally wired to an input source – could be a User Value)	Limited to max float * (decimal point depends on resolution)			Oper
In2	Input 2 value (normally wired to an input source – could be a User Value)	Limited to max float * (decimal point depends on resolution)			Oper
Out	Indicates the analogue value of the output	Between high and low limits			R/O
Status	This parameter is used in conjunction with Fallback to indicate the status of the operation. Typically, status is used to flag fault conditions and may be used as an interlock for other operations.	Good Bad			R/O

* Max float in this instrument is ±9,999,999,999

14.3.3 Sample and Hold Operation

The diagram below shows the operation of the sample and hold feature.

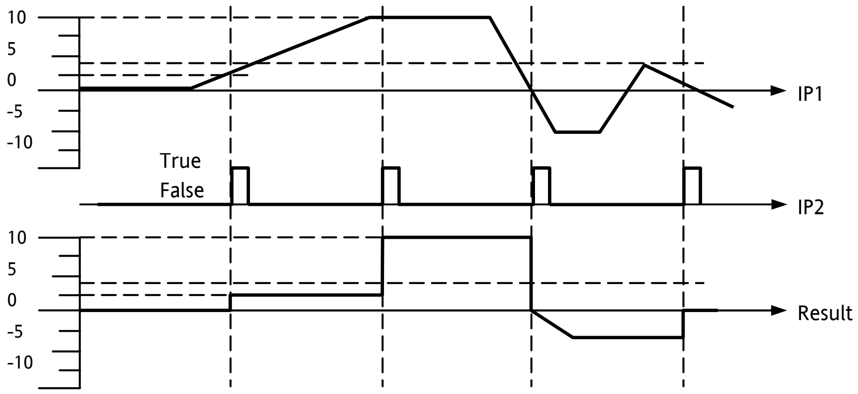


Figure 14-4: Sample and Hold

14.4 Eight Input Analog Multiplexers

The eight Input analogue multiplexers may be used to switch one of eight inputs to an output. It is usual to wire inputs to a source within the controller that selects that input at the appropriate time or event. It is possible to enable four multiplexers from the **'Instrument/Options'** folder.

Multiple Input Operator Parameters

Folder – Mux8 (8 Input Multiplexers)		**Sub-folders: 1 to 4**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
LowLimit	The high limit for all inputs and the fall back value.	-99999 to High limit (decimal point depends on resolution)			Conf
HighLimit	The low limit for all inputs and the fall back value.	Low limit to 99999 (decimal point depends on resolution)			Conf
Fallback	The state of the Output and Status parameters in case of a fault condition. This parameter could be used in conjunction with Fallback Val.	Clip Bad Clip Good Fall Bad Fall Good Upscale DownScale	Descriptions see section 3.4.5		Conf
Fallback Val	Used (in accordance with Fallback) to define the output value during fault conditions	-99999 to 99999 (decimal point depends on resolution)			Conf
Select	Used to select which input value is assigned to the output.	Input1 to Input8			Oper
Input1 to 8	Input values (normally wired to an input source)	-99999 to 99999 (decimal point depends on resolution)			Oper
Out	Indicates the analogue value of the output	Between high and low limits			R/O
Status	Used in conjunction with Fallback to indicate the status of the operation. Typically, status is used to flag fault conditions and may be used as an interlock for other operations.	Good Bad			R/O

14.4.1 Fallback

The fallback strategy will come into effect if the status of the input value is bad or if the input value is outside the range of Input Hi and Input Lo.

In this case the fallback strategy may be configured as:-

Fallback Good – the output value will be the fallback value and the output status will be 'Good'.

Fallback Bad – the output value will be the fallback value and the output status will be 'Bad'.

Clip Good – If the input is outside a limit the output will be clipped to the limit and the status will be 'Good'.

Clip Bad – If the input is outside a limit the output will be clipped to the limit and the status will be 'Bad'.

Upscale – the output value will be Output Hi and the output status will be 'Bad'.

Downscale – the output value will be Output Lo and the output status will be 'Bad'.

15. CHAPTER 15 INPUT CHARACTERISATION

15.1 Input Linearisation

The Lin16 function block converts an input signal into an output PV using a series of up to 15 straight lines to characterise the conversion.

The function block provides the following behaviour.

1. The Input values must be monotonic and constantly rising.
2. To convert the MV to the PV, the algorithm will search the table of inputs until the matching segment is found. Once found, the points either side will be used to interpolate the output value.
3. If during the search, a point is found which is not above the previous (below for inverted) then the search will be terminated and the segment taken from the last good point to the extreme (In Hi-Out Hi) see following diagram.

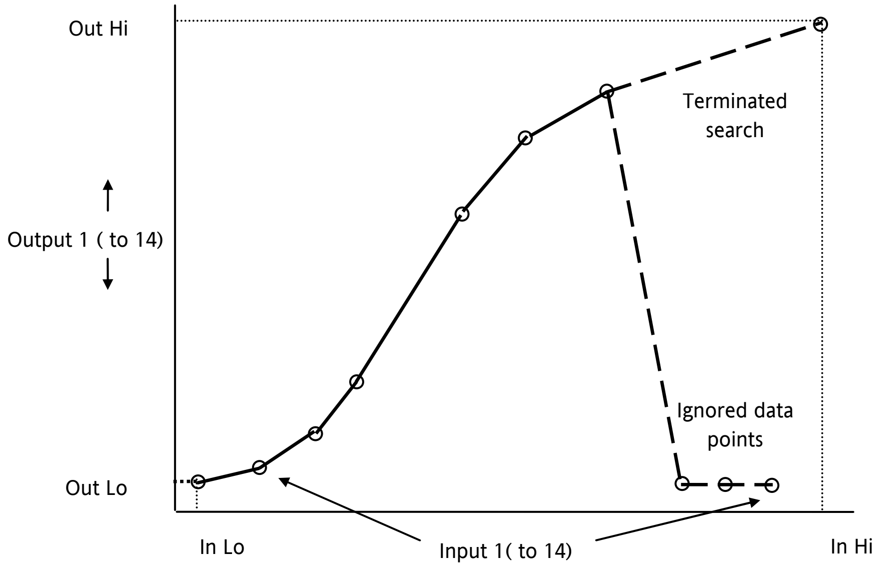


Figure 15-1: Linearisation Example

Notes:

1. The linearisation block works on rising inputs/rising outputs or rising inputs/falling outputs. It is not suitable for outputs which rise and fall on the same curve.
2. Input Lo/Output Lo and Input Hi/Output Hi are entered first to define the low and high points of the curve. It is not necessary to define all 15 intermediate points if the accuracy is not required. Points not defined will be ignored and a straight line fit will apply between the last point defined and the Input Hi/Output Hi point. If the input source has a bad status (sensor break, or over-range) then the output value will also have a bad status.

1. If the input value is outside the translated range then the output status will indicate Bad, and the value will be limited to the nearest output limit.
2. The units and resolution parameters will be used for the output values. The input values resolution and units will be specified by the source of the wire.
3. If the 'Out Low' is higher than the 'Out High' then the translation will be inverted.

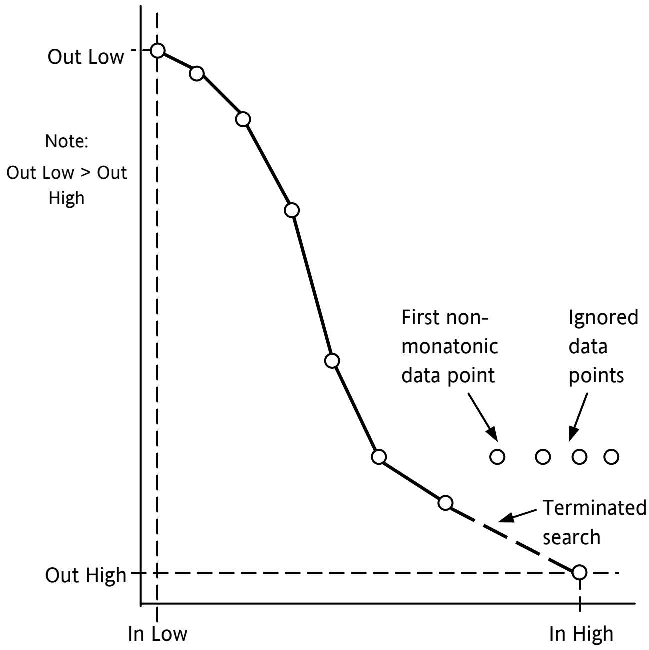


Figure 15-2: How an Inverted Curve will Terminate its search when it detects non-monatonic data

15.1.1 Compensation for Sensor Non-Linearities

The custom linearisation feature can also be used to compensate for errors in the sensor or measurement system. The intermediate points are, therefore, available in Level 1 so that known discontinuities in the curve can be calibrated out. The diagram below shows an example of the type of discontinuity which can occur in the linearisation of a temperature sensor.

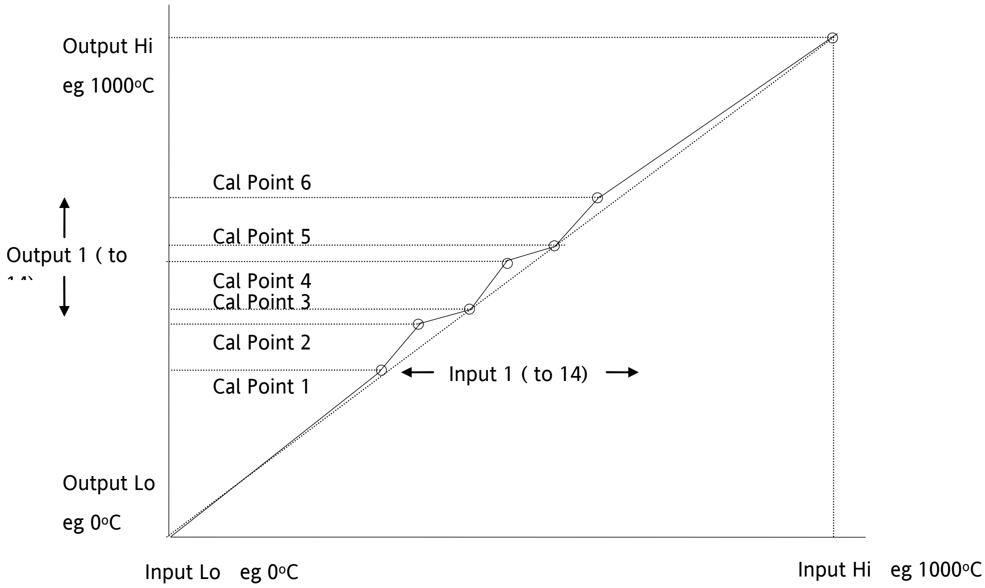


Figure 15-3: Compensation for Sensor Discontinuities

The calibration of the sensor uses the same procedure as described above. Adjust the output (displayed) value against the corresponding input value to compensate for any errors in the standard linearisation of the sensor.

15.1.2 Input Linearisation Parameters

List Folder – Lin16		**Sub-folders: 1 to 2**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Units	Units of the linearised output	None AbsTemp V, mV, A, mA, PH, mmHg, psi, Bar, mBar, %RH, %, mmWG, inWG, inWW, Ohms, PSIG, %O2, PPM, %CO2, %CP, %/sec, RelTemp mBar/Pa/T sec, min, hrs,			Conf
Resolution	Resolution of the output value	XXXXX. XXXX.X, XXX.XX, XX.XXX, X.XXXX			Conf
In	Input measurement to linearise. Wire to the source for the custom linearisation	Between InLowLimit and InHighLimit		0	Oper
FallbackType	Fallback Type The fallback strategy will come into effect if the status of the input value is bad or if the input value is outside the range of input high scale and input low scale. In this case the fallback strategy may be configured as:	Clip Bad	If the input is outside a limit the output will be clipped to the limit and the status will be BAD	ClipBad	Oper
		Clip Good	If the input is outside a limit the output will be clipped to the limit and the status will be GOOD		
		Fall Bad	The output value will be the fallback value and the output status will be BAD		
		Fall Good	The output value will be the fallback value and the output status will be GOOD		
		Upscale	The output value will be output high scale and the output status will be BAD		
		DownScale	The output value will be the output low scale and the output status will be BAD		
Fallback Value	In the event of a bad status, the output may be configured to adopt the fallback value. This allows the strategy to dictate a safe output in the event of a fault being detected.			0	Oper
Out	Linearisation Result	Between OutLowLimit and OutHighLimit			R/O
InLowLimit	Adjust to the low input value	-99999 to InHighLimit		0	Conf
OutLowLimit	Adjust to correspond to the low input value	-99999 to OutHighLimit		0	Conf
InHighLimit	Adjust to the high input value	InLowLimit to 99999		0	Conf
OutHighLimit	Adjust to correspond to the high input value	OutLowLimit to 99999		0	Conf
In1	Adjust to the first break point			0	Oper
Out1	Adjust to correspond to input 1			0	Oper
…etc up to				0	
In14	Adjust to the last break point			0	Oper
Out14	Adjust to correspond to input 14			0	Oper
Status	Status of the block. A value of zero indicates a healthy conversion.	Good Bad	Within operating limits A bad output may be caused by a bad input signal (perhaps the input is in sensor break) or an output which is out of range		R/O

The 16 point linearisation does not require you to use all 16 points. If fewer points are required, then the curve can be terminated by setting the first unwanted value to be less than the previous point.

Conversely if the curve is a continuously decreasing one, then it may be terminated by setting the first unwanted point above the previous one.

15.2 Polynomial

Folder – Poly		**Sub-Folders: 1 to 2**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
LinType	To select the input type. The linearisation type selects which of the instruments linearisation curves is applied to the input signal. The instrument contains a number of thermocouple and RTD linearisations as standard. In addition there are a number of custom linearisations that may be downloaded using iTools to provide linearisations of non-temperature sensors.	J , K, L, R, B, N, T, S, PL2, C, PT100, Linear, SqRoot	J	Conf
Units	Units of the output	None AbsTemp V, mV, A, mA, PH, mmHg, psi, Bar, mBar, %RH, %, mmWG, inWG, inWW, Ohms, PSIG, %O2, PPM, %CO2, %CP, %/sec, RelTemp mBar/Pa/T sec, min, hrs,	None	Conf
Resolution	Resolution of the output value	XXXXX. XXXX.X, XXX.XX, XX.XXX, X.XXXX	XXXXX	Conf
In	Input Value The input to the linearisation block	Range of the input wired from		Oper
Out	Output value	Between Out Low and Out High		R/O
InHighScale	Input high scale	In Low to99999	0	Oper
InLowScale	Input low scale	-99999 to In High	0	Oper
OutHighScale	Output high scale	Out Low to 99999	0	Oper
OutLowScale	Output low scale	-99999 to Out High	0	Oper
Fallback Type	Fallback Type The fallback strategy will come into effect if the status of the input value is bad or if the input value is outside the range of input high scale and input low scale. In this case the fallback strategy may be configured as:	Clip Bad: If the input is outside a limit the output will be clipped to the limit and the status will be BAD Clip Good: If the input is outside a limit the output will be clipped to the limit and the status will be GOOD Fall Bad: The output value will be the fallback value and the output status will be BAD Fall Good: The output value will be the fallback value and the output status will be GOOD		Conf

Folder – Poly		Sub-Folders: 1 to 2			
Name	**Parameter Description**	**Value**		**Defaul t**	**Access Level**
		Upscale	The output value will be output high scale and the output status will be BAD		
		Down-Scale	The output value will be the output low scale and the output status will be BAD		
FallbackValue	Value to be adopted by the output in the event of Status = Bad				Oper
Status	Indicates the status of the linearised output:	Good	Good indicates the value is within range and the input is not in sensor break.		R/O
		Bad	Indicates the Value is out of range or the input is in sensor break. Note: This is also effected by the configured fallback strategy		

16. CHAPTER 16 LOAD

The load simulation block provides styles of load which can be used to allow an instrument configuration to be tested before connection to the process plant. In the current issue of firmware the simulated loads available are Oven and Furnace.

16.1 Load Parameters

Folder – Load		Sub-Folders: None			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Type	The type of load simulation to use. Oven is a simple load of 3 first order lags, providing a single process value for connection to the control loop. Furnace consists of 12 interactive first order lags giving a slave PV, followed by 6 interactive first order lags giving a master PV.	Oven Furnace	Simulates the characteristics of a typical oven Simulates the characteristics of a typical furnace	Oven	Conf
Resolution	The display resolution of the resultant PV Out.				Conf
Units	The Units of the resultant PV.				Conf
Gain	The gain of the load, the input power is multiplied by gain, before use by the load.				Oper
TimeConst1	The time constant of lag 1 in the Oven load and slave lags (1-12) of the Furnace load. The time constant has units of seconds.				Oper
TimeConst2	The time constant of lag 2/3 of the Oven load and master lags (13-18) of the furnace load.				Oper
Attenuation (Furnace load only)	Attenuation Between PV1 and PV2 Stages. Used in the advanced furnace load and defines an attenuation factor between the slave and master lags				Oper
Ch 2 Gain	Defines the relative gain when cooling is requested, applied to the input power when the power requested is < 0.				Oper
PVFault	The load function block provides 2 PV outputs, sensor fault can be used to generate a fault condition on these PV's such that the bad status is passed along a wire to be consumed by another block such as the loop. The sensor fault can be confiured as:	None PVOut1 PVOut2 Both	No fault conditions. Fault on the first output (slave). Fault on the second output (master). A fault on first and second outputs (master and slave).		Oper
PV Out1	First Process Value The PV in Process Value an Oven load or the Slave PV in a furnace load.				R/O
PV Out2 (Furnace load only)	Second Process Value Second process value, lagged from PVOut1, used as a cascade master input. The Master PV in the Furnace load.				R/O

Folder – Load		Sub-Folders: None			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
LoopOutCh1	Loop output channel 1 input. The output of the loop as wired to the load simulation, this is the power requested of the load. This can be used as the heat demand.				Oper
LoopOutCh2	Loop output channel 2 input. The output of the loop as wired to the load simulation, this is the power requested of the load. This can be used as the cool demand.				Oper
Noise	Noise Added to PV This is used to make the PV of the load appear noisy, and hence more like a real measurement.	Off 1 to 99999	The amount of noise is specified in engineering units.	Off	Oper
Offset	Process offset Used to configure an offset in the process. In a temperature application this could represent the ambient operating temperature of the plant.				Oper

17. CHAPTER 17 CONTROL LOOP SET UP

The Mini8 has eight loops of control. Each Loop has two outputs, Channel 1 and Channel 2, each of which can be configured for PID or On/Off.

The control function block is divided into a number of sections the parameters of which are all listed under the Folder **'Loop'**.

The 'Loop' folder contains sub-folders for each section as shown diagrammatically below.

17.1 What is a Control Loop?

An example of a heat only temperature control loop is shown below:-

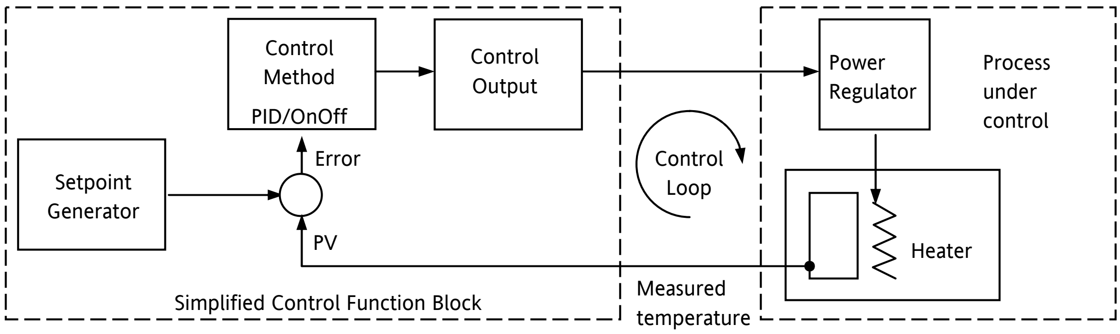


Figure 17-1: Single Loop Single Channel

The actual temperature measured at the process (PV) is connected to the input of the controller. This is compared with a setpoint (or required) temperature (SP). If there is an error between the set and measured temperature the controller calculates an output value to call for heating or cooling. The calculation depends on the process being controlled but normally uses a PID algorithm. The output(s) from the controller are connected to devices on the plant which cause the heating (or cooling) demand to be adjusted which in turn is detected by the temperature sensor. This is referred to as the control loop.

17.2 Loop Parameters - Main

Folder – Loop.1 to Loop.8		Sub-Folder: Main			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
AutoMan	To select Auto or Manual operation.	Auto Man	Automatic (closed loop) operation Manual (output power adjusted by the user) operation	Auto	Oper
PV	The process variable input value. This is typically wired from an analogue input.	Range of the input source			Oper
Inhibit	Used to stop the loop controlling. If enabled the loop will stop control and the output of the loop will be set to the safe output value. On exit from inhibit the transfer will be bumpless. This may be wired to an external source	No Yes	Inhibit disabled Inhibit enabled	No	Oper
TargetSP	The value of setpoint at which the control loop is aiming. It may come from a number of different sources, such as internal SP and remote SP.	Between setpoint limits			Oper
WorkingSP	The current value of the setpoint being used by the control loop. It may come from a number of different sources, such as internal SP and Remote SP. The working setpoint is always read-only as it is derived from other sources.	Between setpoint limits			R/O
ActiveOut	The actual output of the loop before it is split into the channel 1 and channel 2 outputs.				R/O
IntHold	Stops Integral action			No	Oper

17.3 Loop Set up

These parameters configure the type of control.

Folder – Loop.1 to Loop.8		Sub-folder: Setup			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Ch1 ControlType	Selects the channel 1 control algorithm. You may select different algorithms for channels 1 and 2. In temperature control applications, Ch1 is usually the heating channel, Ch2 is the cooling channel.	Off OnOff PID	Channel turned off On/off control 3 term or PID control	PID	Conf
Ch2 ControlType	Control type for channel 2				
Control Action	Control Action	Rev Dir	Reverse acting. The output increases when the PV is below SP. This is the best setting for heating control. Direct acting. The output increases when the PV is above SP. This is the best setting for cooling control	Rev	Conf
PB Units	Proportional band units.	Eng Percent	Engineering units eg C or F Per cent of loop span (range Hi - Range Lo)	Eng	Conf

Derivative Type	Selects whether the derivative acts only on PV changes or on Error (either PV or Setpoint changes).	PV Error	Only changes in PV cause changes to the derivative output. Changes to either PV or SP will cause a derivative output.	PV	Conf
The above two parameters do not appear if either Ch1 or Ch2 are configured for Off or OnOff control					

17.3.1 Types of Control Loop

17.3.1.1 On/Off Control

On/Off control simply turns heating power on when the PV is below setpoint and off when it is above setpoint. If cooling is used, cooling power is turned on when the PV is above setpoint and off when it is below. The outputs of such a controller will normally be connected to relays – hysteresis may be set as described in the Alarms section to prevent relay chatter or to provide a delay in the control output action.

17.3.1.2 PID Control

PID control, also referred to as 'Three Term Control', is a technique used to achieve stable straight line control at the required setpoint. The three terms are:

P = Proportional band

I = Integral time

D = Derivative time

The output from the controller is the sum of the contributions from these three terms. The combined output is a function of the magnitude and duration of the error signal, and the rate of change of the process value. It is possible to turn off integral and derivative terms and control on only proportional, proportional plus integral or proportional plus derivative.

17.4 PID Control

The PID controller consists of the following parameters:-

Parameter	Meaning or Function
Proportional Band 'PB'	The proportional term, in display units or %, delivers an output that is proportional to the size of the error signal.
Integral Time 'Ti'	Removes steady state control offsets by ramping the output up or down in proportion to the amplitude and duration of the error signal.
Derivative Time 'Td'	Determines how strongly the controller will react to the rate of change in the measured value. It is used to prevent overshoot and undershoot and to restore the PV rapidly if there is a sudden change in demand.
High Cutback 'CBH'	The number of display units, above setpoint, at which the controller will increase the output power, in order to prevent undershoot on cool down.
Low Cutback 'CBL'	The number of display units, below setpoint, at which the controller will cutback the output power, in order to prevent overshoot on heat up.
Relative Cool Gain 'R2G'	Only present if cooling has been configured. Sets the cooling proportional band, which equals the heat proportional band value divided by the cool gain value.

17.4.1 Proportional Term

The proportional term delivers an output which is proportional to the size of the error signal. An example of this is shown below, for a temperature control loop, where the proportional band is 10ºC and an error of 3ºC will produce an output of 30%.

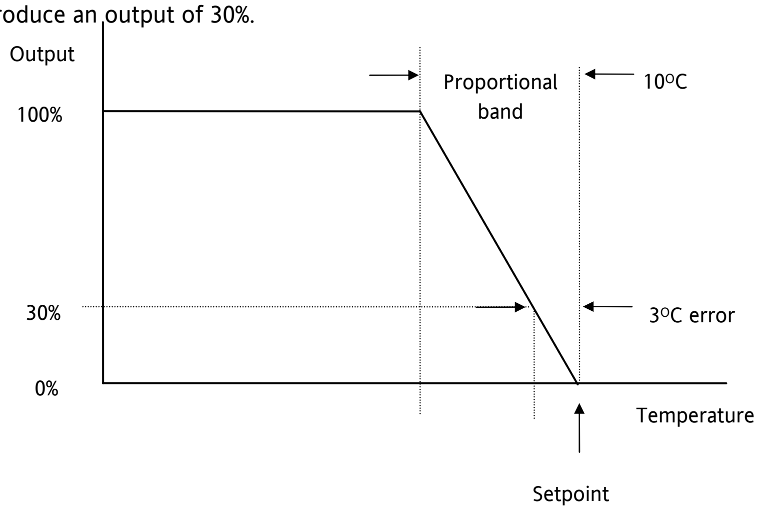


Figure 17-2: Proportional Action

Proportional only controllers will, in general, provide stable straight line control, but with an offset corresponding to the point at which the output power equals the heat loss from the system.

The proportional term may be set in engineering units, as shown in the above example, or as a percentage of the controller range. In the above example, if the inout range is 0 to 1000ºC the proportional band is set to 1%.

17.4.2 Integral Term

The integral term removes steady state control offset by ramping the output up or down in proportion to the amplitude and duration of the error signal. The ramp rate (reset rate) is the integral time constant, and must be longer than the time constant of the process to avoid oscillations.

17.4.3 Derivative Term

The derivative term is proportional to the rate of change of the temperature or process value. It is used to prevent overshoot and undershoot of the setpoint by introducing an anticipatory action. The derivative term has another beneficial effect. If the process value falls rapidly, due, for example, an oven door being opened during operation, and a wide proportional band is set the response of a PI controller can be quite slow. The derivative term modifies the proportional band according to this rate of change having the effect of narrowing the proportional band. Derivative action, therefore, improves the recovery time of a process automatically when the process value changes rapidly.

Derivative can be calculated on change of PV or change of Error. For applications such as furnace control, it is common practice to select Derivative on PV to prevent thermal shock caused by a sudden change of output following a change in setpoint.

17.4.4 High and Low Cutback

While the PID parameters are optimised for steady state control at or near the setpoint, high and low cutback parameters are used to reduce overshoot and undershoot for large step changes in the process. They respectively set the number of degrees above and below setpoint at which the controller will start to increase or cutback the output power.

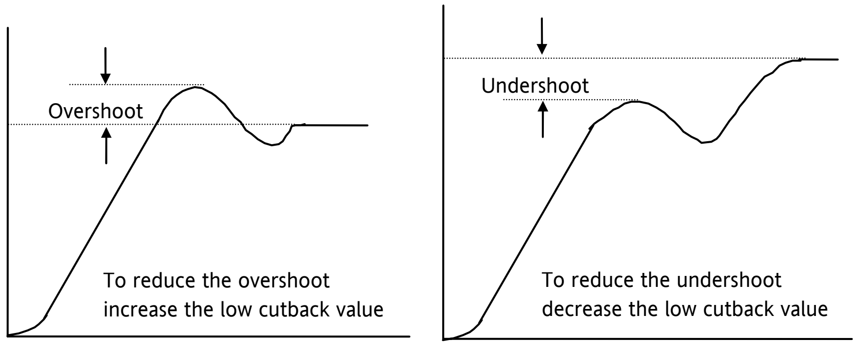


Figure 17-3: High and Low Cutback

17.4.5 Integral action and manual reset

In a full three-term controller (that is, a PID controller), the integral term automatically removes steady state errors from the setpoint. If the controller is set as a PD controller, the integral term will be set to 'OFF'. Under these conditions the measured value may not settle precisely at setpoint. The Manual Reset parameter (MR) represents the value of the power output that will be delivered when the error is zero. You must set this value manually in order to remove the steady state error.

17.4.6 Relative Cool Gain

The gain of channel 2 control output, relative to the channel 1 control output.
Relative Ch2 Gain compensates for the different quantities of energy needed to heat, as opposed to that needed to cool, a process. For example: water cooling applications might require a relative cool gain of 4 (cooling is 4 times faster than the heat-up process).
(This parameter is set automatically when Autotune is used). A nominal setting of around 4 is often used.

17.4.7 Loop Break Time

The loop is considered to be broken if the PV does not respond to a change in the output. Since the time of response will vary from process to process the Loop Break Time parameter allows a time to be set before a loop break alarm is initiated. In these circumstances the output power will drive to high or low limit. For a PID controller, if the PV has not moved by 0.5 x Pb in the loop break time the loop is considered to be in break. The loop break time is set by the Autoune, a typical value is 12 x Td. For an On/Off controller Loop Break Time is not shown and loop break alarm is inhibited.

17.4.8 Cooling Algorithm

The method of cooling may vary from application to application.
For example, an extruder barrel may be cooled by forced air (from a fan), or by circulating water or oil around a jacket. The cooling effect will be different depending on the method. The cooling algorithm may be set to linear where the controller output changes linearly with the PID demand signal, or it may 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.

17.4.9 Gain Scheduling

Gain scheduling is the automatic transfer of control between one set of PID values and another. It may be used in very non-linear systems where the control process exhibits large changes in response time or sensitivity, see diagram below. This may occur, for example, over a wide range of PV, or between heating and cooling where the rates of response may be significantly different. The number of sets depends on the non-linearity of the system. Each PID set is chosen to operate over a limited (approximately linear) range.

In the Mini8 controller, this is done at a preset strategy defined by the parameter 'Scheduler Type'. The choices are:

No.	Type	Description
0	Off	Just one fixed set of PID values
1	Set	The PID set can be selected manually or from a digital input
2	SP	The transfer between one set and the next depends on the value of the SP
3	PV	The transfer between one set and the next depends on the value of the PV
4	Error	The transfer between one set and the next depends on the value of the error
5	OP	The transfer between one set and the next depends on the value of the OP demand
6	Rem Sched IP	The transfer between one set and the next depends on the value from a remote source for example, a digital input

The Mini8 controller has three sets of PID values for each loop – the maximum number, which you may wish to use, is set by 'Num Sets' parameter.

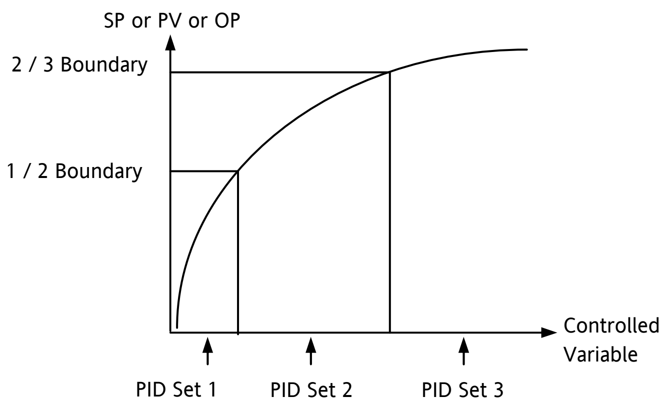


Figure 17-4: Gain Scheduling in a Non-Linear System

17.4.10 PID Parameters

Control loops must be specifically ordered – Order Code MINI8 - LP8. To enable a loop place one of the Loop function blocks on the graphical wiring page.

Folder – Loop		**Sub-folders: Loop1.PID to Loop8.PID**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
SchedulerType	To choose the type of gain scheduling	Off Set SP PV Error OP Rem	See above for explanation Parameters displayed will vary depending on type of scheduling selected.	Off	Oper
Num Sets	Selects the number of PID sets to present. Allows the lists to be reduced if the process does not require the full range of PID sets.	1 to 3		1	Oper
Active Set	Currently working set	Set1 Set2 Set3		Set1	R/O except type 'Set'
Boundary 1-2	Sets the level at which PID set 1 changes to PID set 2	Range units		0	Oper
Boundary 3-4	Sets the level at which PID set 2 changes to PID set 3	Range units		0	Oper
ProportionalBand 1, 2, 3	Proportional band Set1/Set2/Set3	0 to 99999 Eng units		300	Oper
IntegralTime 1, 2, 3	Integral term Set1/Set2/Set3			360s	Oper
DerivativeTime 1, 2, 3	Derivative term Set1/Set2/Set3			60s	Oper
RelCh2Gain 1, 2, 3	Relative cool gain Set1/Set2/Set3			1	Oper
CutbackHigh 1, 2, 3	Cutback high Set1/Set2/Set3			Auto	Oper
CutbackLow 1, 2, 3	Cutback low Set1/Set2/Set3			Auto	Oper
ManualReset 1, 2, 3	Manual reset Set1/Set2/Set3. This must be set to 0.0 when the integral term is set to a value			0.0	Oper
LoopBreakTime 1, 2, 3	Loop break time Set1/Set2/Set3			100	Oper
OutputHi 1, 2, 3	Output High Limit Set1/Set2/Set3			100	Oper
OutputLo 1, 2, 3	Output Low Limit Set1/Set2/Set3			-100	

17.5 Tuning

In tuning, you match the characteristics (PID parameters) of the controller to those of the process being controlled in order to obtain good control. Good control means:

Stable, 'straight-line' control of the PV at setpoint without fluctuation

No overshoot, or undershoot, of the PV setpoint

Quick response to deviations from the setpoint caused by external disturbances, thereby rapidly restoring the PV to the setpoint value.

Tuning involves calculating and setting the value of the parameters listed in the above table.

17.5.1 Automatic Tuning

This controller uses a one-shot tuner that automatically sets up the initial values of the parameters listed in the table on the previous page.

17.5.2 One-shot Tuning

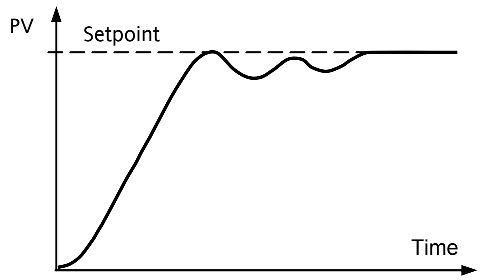
The 'one-shot' tuner works by switching the output on and off to induce an oscillation in the measured value. From the amplitude and period of the oscillation, it calculates the tuning parameter values.

If the process cannot tolerate full heating or cooling being applied, then the levels can be restricted by setting the high power limit ('Output Hi') and low power limit ('Output Lo'). However, the measured value *must* oscillate to some degree for the tuner to be able to calculate values.

A One-shot Tune can be performed at any time, but normally it is performed only once during the initial commissioning of the process. However, if the process under control subsequently becomes unstable (because its characteristics have changed), you can re-tune again for the new conditions.

It is best to start tuning with the process at ambient conditions and with the SP close to the normal operating level. This allows the tuner to calculate more accurately the low cutback and high cutback values which restrict the amount of overshoot, or undershoot.

Typical automatic tuning cycle



Autotune starts 1 minute after being turned on to determine steady state conditions.

Tuning normally takes place at a PV which has a value of setpoint x 0.7.

The power is automatically turned on and off to cause oscillations.

From the results the values shown in the table are calculated

17.5.3 Calculation of the cutback values

Low cutback and *High cutback* are values that restrict the amount of overshoot, or undershoot, that occurs during large step changes in PV (for example, under start-up conditions).

If either low cutback, or high cutback, is set to 'Auto' the values are fixed at three times the proportional band, and are not changed during automatic tuning.

To tune the cutback values, first set them to values other than Auto, then perform a tune as usual.

17.5.4 Manual Tuning

If for any reason automatic tuning gives unsatisfactory results, you can tune the controller manually. There are a number of standard methods for manual tuning. The one described here is the Ziegler-Nichols method.

With the process at its normal running conditions:

Set the Integral Time and the Derivative Time to OFF.

Set High Cutback and Low Cutback to 'Auto'.

Ignore the fact that the PV may not settle precisely at the setpoint.

If the PV is stable, reduce the proportional band so that the PV just starts to oscillate. If PV is already oscillating, increase the proportional band until it just stops oscillating. Allow enough time between each adjustment for the loop to stabilise. Make a note of the proportional band value 'PB' and the period of oscillation 'T'.

Set the proportional band, integral time and derivative time parameter values according to the calculations given in the table below:-

Type of control	Proportional band (PB)	Integral time (Ti) seconds	Derivative time (Td) seconds
Proportional only	2xPB	OFF	OFF
P + I control	2.2xPB	0.8xT	OFF
P + I + D control	1.7xPB	0.5xT	0.12xT

17.5.5 Setting the Cutback Values

The above procedure sets up the parameters for optimum steady state control. If unacceptable levels of overshoot or undershoot occur during start-up, or for large step changes in PV, then manually set the cutback parameters.

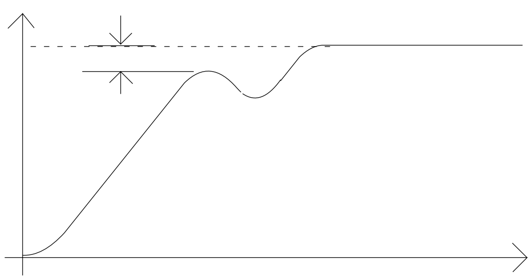
Proceed as follows:

Set the low and high cutback values to three proportional bandwidths (that is to say, 'CBH'= 'CBL' = 3 x PB).

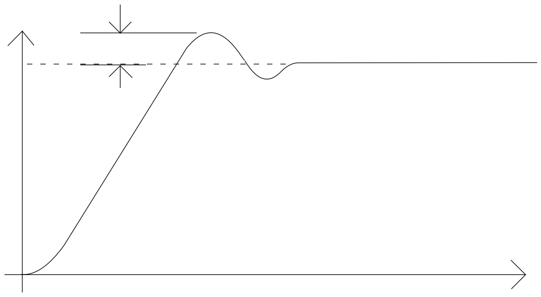
Note the level of overshoot, or undershoot, that occurs for large PV changes (see the diagrams below).

In example (a) increase Low Cutback by the undershoot value. In example (b) reduce Low Cutback by the overshoot value.

Example (a)



Example (b)



Where the PV approaches setpoint from above, you can set High Cutback in a similar manner.

17.5.6 Multi-zone applications.

The tuning of one loop can be unduly influenced by the controlling effect of adjacent zone(s). Ideally the zone either side of the one being tuned should be turned OFF, or put in manual with the power level set to keep its temperature at about the usual operating level.

17.5.7 Tune Parameters

Folder – Loop.Loop.1 to Loop.8		**Sub-folder: Tune**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
AutoTune	To start self tuning	Off On	Stop Start	Stop	Oper
OutputHigh Limit	Set this to limit the maximum output power level which the controller will supply during the tuning process. If the high output power limit set in the output list is lower the autotune high limit will be clipped to this value.	Between Low Output and 100.0		!00.0	Oper
OutputLow Limit	Set this to limit the minimum % output power level which the controller will supply during the tuning process. If the low output power limit set in the output list is higher the autotune low limit will be clipped to this value.	Between High Output and 0.0		0.0	Oper
State	Shows if self tuning is in progress	OFF			R/O
Stage	Shows the progress of the self tuning	Reset			R/O
Stage Time	Time in the particular stage				R/O

17.6 Setpoint Function Block

For each of the 8 loops, the controller setpoint is the **Working Setpoint** that may come from a number of alternative sources. This is the value ultimately used to control the process variable in each loop.

The working setpoint may be derived from:-

1. SP1 or SP2, both of which are individually set, can be selected by an external signal or via the SPSelect parameter over communications.
2. From an external (remote) analogue source
3. The output of a programmer function block and will, therefore, vary in accordance with the program in use.

The setpoint function block also provides the facility to limit the rate of change of the setpoint before it is applied to the control algorithm. It will also provide upper and lower limits. These are defined as setpoint limits for the local setpoints and instrument range high and low for other setpoint sources. All setpoints are ultimately subject to a limit of range hi and range lo.

User configurable methods for tracking are available, such that the transfer between setpoints and between operational modes will not cause a bump in the setpoint.

17.6.1 Setpoint Function Block

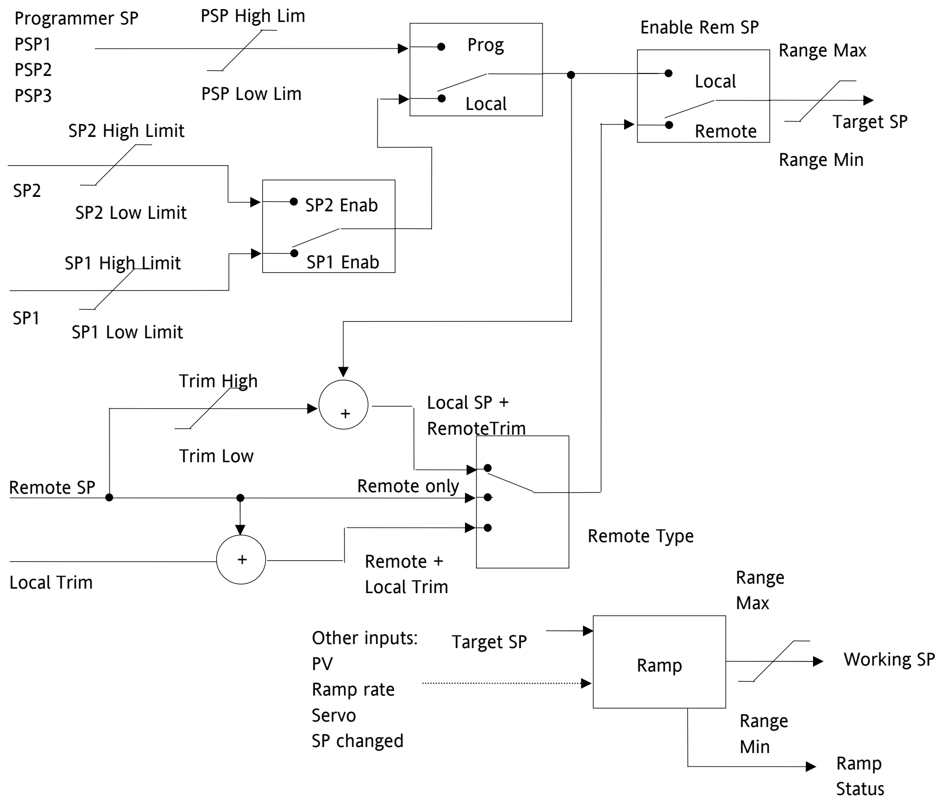


Figure 17-5: Setpoint Function Block

17.6.2 SP Tracking

When setpoint tracking is enabled and the local setpoint is selected, the local setpoint is copied to ‘TrackSP’. Tracking now ensures that the alternate SP follows or tracks this value. When the alternate setpoint is selected it initially takes on the tracked value thus ensuring that no bump takes place. The new setpoint is then adopted gradually. A similar action takes place when returning to the local setpoint.

17.6.3 Manual Tracking

When the controller is operating in manual mode the currently selected SP tracks the PV. When the controller resumes automatic control there will be no step change in the resolved SP.

17.6.4 Rate Limit

Rate limit will control the rate of change of setpoint. It is enabled by the **‘Rate’** parameter. If this is set to Off then any change made to the setpoint will be effective immediately. If it is set to a value then any change in the setpoint will be effected at the value set in units per minute. Rate limit also acts on SP2 and when switching between SP1 and SP2.

When rate limit is active the ‘**RateDone**’ parameter will display ‘**No**’. When the setpoint has been reached this parameter will change to ‘**Yes**’.

When ‘**Rate’** is set to a value (other than Off) an additional parameter ‘**SPRate Disable’** is displayed which allows the setpoint rate limit to be turned off and on without the need to adjust the ‘Rate’ parameter between Off and a value.

17.6.5 Setpoint Parameters

Folder – Loop.1 to Loop.8		**Sub-folder: SP**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Range High	The Range limits provide a set of absolute maximums and minimums for setpoints within the control loop. Any derived setpoints are ultimately clipped to be within the Range limits. If the Proportional Band is configured as % of Span, the span is derived from the Range limits.	Full range of the input type			Conf
Range Low					Conf
SP Select	Select local or alternate setpoint	SP1 SP2	Setpoint 1 Setpoint 2	SP1	Oper
SP1	Primary setpoint for the controller	Between SP high and SP low limits			Oper
SP2	Setpoint 2 is the secondary setpoint of the controller. It is often used as a standby setpoint.				Oper
SP HighLimit	Maximum limit allowed for the local setpoints	Between Range Hi and Range Lo			Oper
SP LowLimit	Minimum limit allowed for the local setpoints				Oper
Alt SP Select	To enable the alternative setpoint to be used. This may be wired to a source such as the programmer Run input.	No Yes	Alternative setpoint disabled Alternative setpoint enabled		Oper
Alt SP	This may be wired to an alternative source such as the programmer or remote setpoint				Oper

Folder – Loop.1 to Loop.8		Sub-folder: SP			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Rate	Limits the maximum rate at which the working setpoint can change. The rate limit may be used to protect the load from thermal shock which may be caused by large step changes in setpoint.	Off or 0.1 to 9999.9 engineering units per minute		Off	Oper
RateDone	Flag which indicates when the setpoint is changing or completed	No Yes	Setpoint changing Complete		R/O
Rate Disable	Setpoint rate disable	No Yes	Enabled Disabled		Oper
SP Trim	Trim is an offset added to the setpoint. The trim may be either positive or negative, the range of the trim may be restricted by the trim limits Setpoint trims may be used in a retransmission system. A master zone may retransmit the setpoint to the other zones, a local trim may be applied to each zone to produce a profile along the length of the machine	Between SP Trim Hi and SP Trim Lo			Oper
SPTrim HighLimit	Setpoint trim high limit				Oper
SPTrim LowLimit	Setpoint trim low limit				Oper
ManualTrack	To enable manual tracking. When the loop is switched from Manual to Auto, the Setpoint is set to the current PV. This is useful if the load is started in Manual Mode, then later switched to Auto to maintain the operating point.	Off On	Manual tracking disabled Manual tracking enabled		R/O
SP Track	Setpoint tracking ensures bumpless transfer in setpoint when switching between a local and an alternate setpoint such as the programmer. This enables the tracking interface provided by TrackPV and TrackVal, which is used by the programmer and other setpoint providers external to the control loop	Off On	Setpoint tracking disabled Setpoint tracking enabled		Conf
Track PV	The programmer tracks the PV when it is servoing or tracking.				R/O
Track SP	Manual Tracking Value. The SP to track for manual tracking.				R/O

17.7 Output Function Block

The output function block allows you to set up output conditions from the control block, such as output limits, hysteresis, output feedforward, behaviour in sensor break, etc.

Folder – Loop.1 to Loop.8		Sub-folder: OP		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Output High Limit	Maximum output power delivered by channels 1 and 2. By reducing the high power limit, it is possible to reduce the rate of change of the process, however, care should be taken as reducing the power limit will reduce the controllers ability to react to disturbance.	Between Output Lo and 100.0%	100.0	Oper
Output Low Limit	Minimum (or maximum negative) output power delivered by channels 1 and 2	Between Output Hi and -100.0%	-100.0	
Ch1 Out	Channel 1 (Heat) output. The Ch1 output is the positive power values (0 to Output Hi) used by the heat output. Typically this is wired to the control output (time proportioning or DC output).	Between output Hi and Output Lo		R/O
Ch2 Out	The Ch2 output is negative portion of the control output (0 – Output Lo) for heat/cool applications. It is inverted to be a positive number so that it can be wired into one of the outputs (time proportioning or DC outputs).	Between output Hi and Output Lo		R/O
Ch2 DeadBand	Ch1/Ch2 Deadband is a gap in percent between output 1 going off and output 2 coming on and vice versa. For on/off control this is taken as a percentage of the hysteresis.	Off to 100.0%	Off	Oper
Rate	Limits the rate at which the output from the PID can change in % change per second. Output rate limit is useful in preventing rapid changes in output from damaging the process or the heater elements.	Off to 9999.9 engineering units per minute	Off	Oper
Rate Disable	Output rate disable	No: Enabled Yes: Disabled		Oper
Ch1 OnOff Hysteresis	Channel hysteresis only shown when channel 1 is configured as OnOff. Hysteresis sets the difference between output on and output off to prevent (relay) chatter.	0.0 to 200.0	10.0	Oper
Ch2 OnOff Hysteresis		0.0 to 200.0	10.0	Oper
SensorBreak Mode	Defines the action taken if the Process Variable is bad, i.e. the sensor has failed. This can be configured as hold, in which case the output of the loop is held at its last good value. Alternately the output can switch to a safe output power defined at configuration.	Safe: To select the level set by 'Safe OP' Hold: To hold the current output level at the point when sensor break occurs	Safe	Oper
Safe OP Val	Sets the output level to be adopted when loop is inhibited	Between output Hi and Output Lo	0	Oper
SbrkOp	Sets the output level to be adopted when in sensor break condition.	Between output Hi and Output Lo	0	Oper

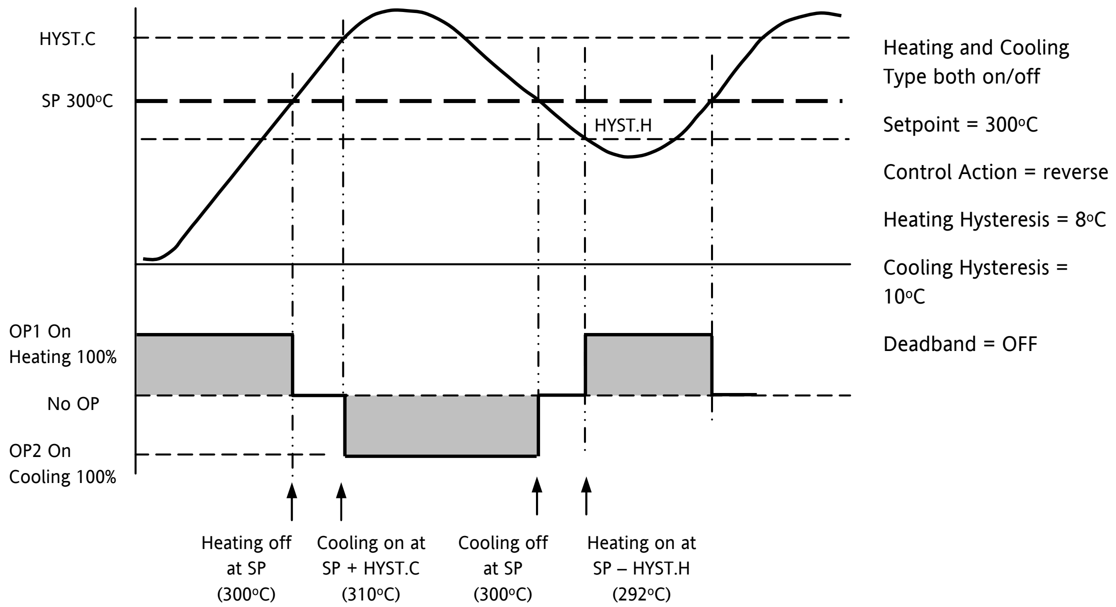
Folder – Loop.1 to Loop.8		Sub-folder: OP			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Manual Mode	Selects the mode of manual operation.	Track Step	In auto the manual output tracks the control output such that a change to manual mode will not result in a bump in the output. on transition to manual the output will be the manual op value as last set by the operator.		Oper
ManualOutVal	The output when the loop is in manual. Note: In manual mode the controller will still limit the maximum power to the power limits, however, it could be dangerous if the instrument is left unattended at a high power setting. It is important that the over range alarms are configured to protect your process. *We recommend that all processes are fitted with an independent over range "policeman"*	Between output Hi and Output Lo			R/O
Cool Type	Selects the type of cooling channel characterisation to be used. Can be configured as water, oil or fan cooling.	Linear Oil Water Fan	These are set to match the type of cooling medium applicable to the process		Conf
FeedForward Type	Feedforward type The following four parameters appear if FF Type ≠ None	None	No signal fed forward	None	Conf
		Remote	A remote signal fed forward		
		SP	Setpoint fed forward		
		PV	PV fed forward		
FeedForward Gain	Defines the gain of the feedforward value, the feed forward value is multiplied by the gain				Conf
FeedForward Offset	Defines the offset of the feedforward value this is added to the scaled feedforward.				Oper
FeedForward Trim Limit	Feedforward trim limits the effect of the PID output. Defines symmetrical limits around the PID output, such that this value is applied to the feedforward signal as a trim.				Oper
FF_Rem	Remote Feedforward signal. Allows an another signal to be used as Feedforward.	This is not affected by FeedForward Gain or Offset			R/O
FeedForward Val	The calculated Feedforward Value.				R/O
TrackOutVal	Value for the loop output to track when OP Track is Enabled.				
Track Enable	When enabled, the output of the loop will follow the track output value. The loop will bumplessly return to control when tracking is turned off.	Off On	Disabled Enabled		Oper
RemOPL	Remote output low limit. Can be used to limit the output of the loop from a remote source or calculation. This must always be within the main limits.	-100.0 to 100.0			Oper
RemOPH	Remote output high limit	-100.0 to 100.0			Oper

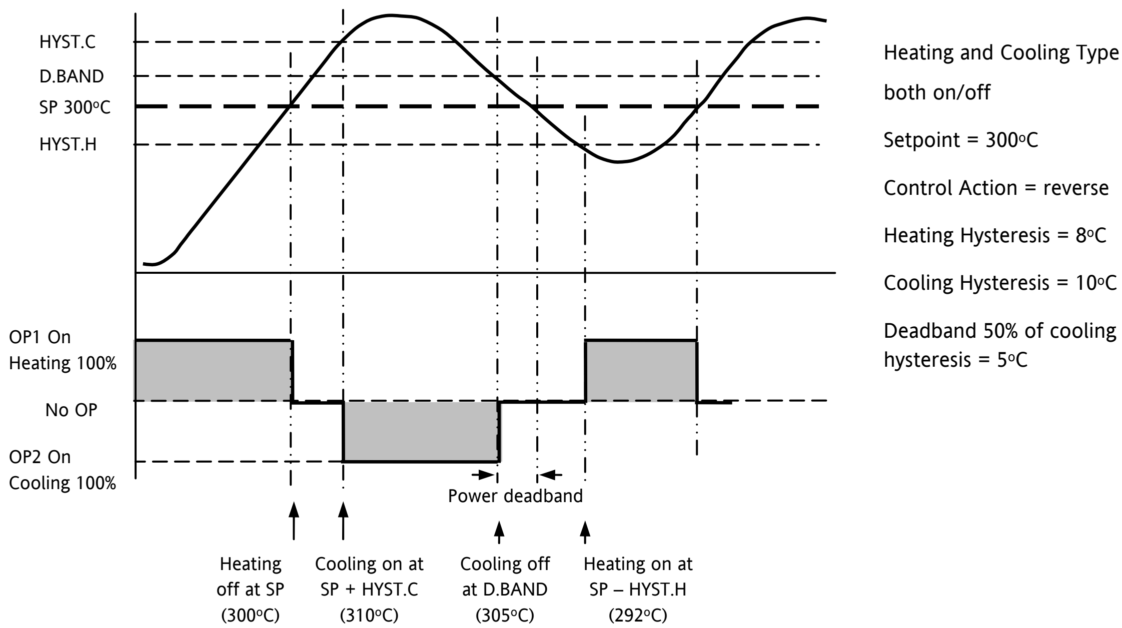
17.7.1 Effect of Control Action, Hysteresis and Deadband

For temperature control '**Loop.1.Control Action'** will be set to '**Reverse'**. For a PID controller this means that the heater power decreases as the PV increases. For an on/off controller output 1 (usually heat) will be on (100%) when PV is below the setpoint and output 2 (usually cool) will be on when PV is above the setpoint

Hysteresis applies to on/off control only. It defines the difference in temperature between the output switching off and switching back on again. The examples below shows the effect in a heat/cool controller.

Deadband (**Ch2 DeadB**) can operate on both on/off control or PID control where 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 integral and derivative terms. Deadband might be used in PID control, for example, where actuators take time to complete their cycle thus ensuring that heating and cooling are not being applied at the same time. Deadband is likely to be used, therefore, in on/off control only. The second example below adds a deadband of 20 to the above example.





18. SETPOINT PROGRAMMER

In a setpoint programmer you can set up a profile in the controller in which the setpoint varies in a pre-determined way over a period of time. Temperature is a very common application where it is required to 'ramp' the process value from one level to another over a set period of time.

The **Program** is divided into a flexible number of **Segments** - each being a single time duration. The total number of segments available is **200** or **50 per program** and it is possible to store up to **50 separate programs.**

It is often necessary to switch external devices at particular times during the program. Up to eight digital 'event' outputs can be programmed to operate during those segments.

An example of a program and two event outputs is shown below.

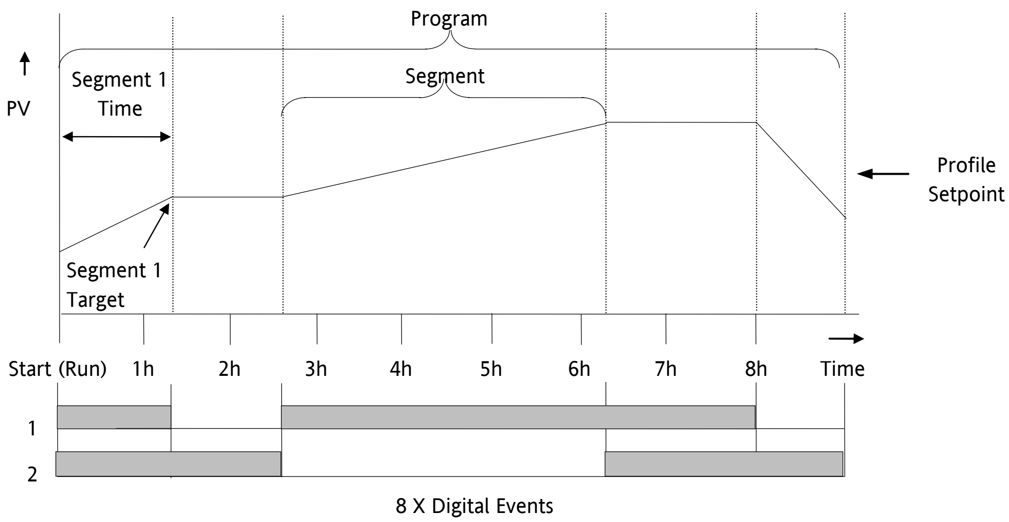


Figure 18-1: A Setpoint Program

Each individual segment can be configured as **Time-to-Target** or **Ramp-Rate**. A program with all segments configured as Time-to-Target is shown below.

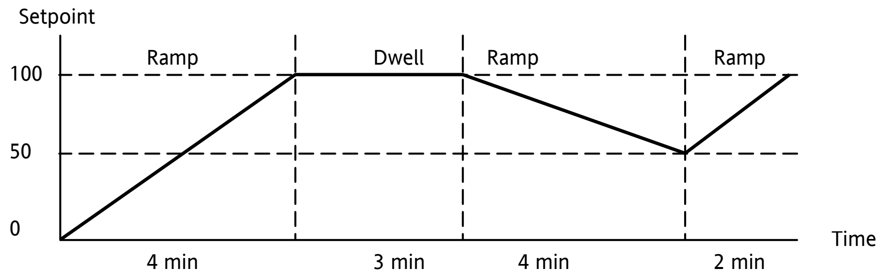


Figure 18-2: Time to Target Programmer

A ramp rate programmer specifies ithe ramp segments as maximum setpoint changes per time unit. The diagram below demonstrates a ramp rate programmer.

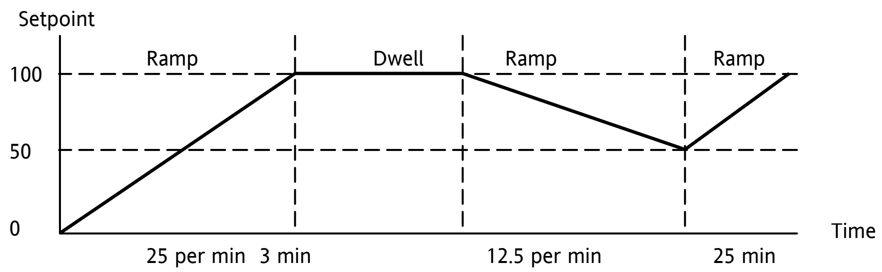


Figure 18-3: Ramp Rate Programmer

18.1 Programmer Operating States

18.1.1 Reset

In reset the programmer is inactive and the controller behaves as a standard controller. It will:-

1. Continue to control with the setpoint determined by the next available source, SP1, SP2, Alternative Setpoint.
2. Allow edits to all segments
3. Return all controlled outputs to the configured reset state.

18.1.2 Run

In run the programmer working setpoint varies in accordance with the profile set in the active program. A program will always run – non configured programs will default to a single Dwell end segment.

18.1.3 Hold

A programmer may only be placed in Hold from the Run or Holdback state. In hold the setpoint is frozen at the current programmer setpoint and the time remaining parameter frozen at its last value. In this state you can make temporary changes to program parameters such as a target setpoint, ramp rates and times. These changes will only remain effective until the end of the currently running segment, when they will be overwritten by the stored program values.

18.1.4 Program Cycles

If the Program Cycles parameter is chosen as greater than 1, the program will execute all its segments (including calls to other programs) then repeat from the beginning. The number of cycles is determined by the parameter value. The Program Cycles parameter has a range of 0 to 999 where 0 is enumerated to CONTinuous.

18.1.5 Servo

Servo can be set in configuration so that when a program is run the setpoint can start from the initial controller setpoint or from the current process value. Whichever it is, the starting point is called the servo point. This can be set in the program.

Servo to PV will produce a smooth and bumpless start to the process.

Servo to SP may be used in a Ramp Rate programmer to guarantee the time period of the first segment. (Note: in a Time to Target programmer the segment duration will always be determined by the setting of the Segment Duration parameter.)

18.1.6 Skip Segment

Moves immediately to the next segment and starts the segment from the current setpoint value.

18.1.7 Advance Segment

Sets the program setpoint equal to the target setpoint and moves to the next segment.

18.1.8 Fast x10 mode

Executes the program at 10x the normal speed. It is provided so that programs can be tested **but the process should not be run in this state**.

18.1.9 Sensor break recovery

On sensor break, the program state changed to HOLD if the current state is RUN or HOLDBACK. Sensor break is defined as status bad on the PV Input parameter. If the program state is in HOLD when PV input status returns to OK, the program state is automatically set back to RUN.

18.1.10 Holdback (Guaranteed Soak)

Holdback freezes the program if the process value (PV) does not track the setpoint (SP) by more than a user defined amount. The instrument will remain in HOLDBACK until the PV returns to within the requested deviation from setpoint.

In a **Ramp** it indicates that the PV is lagging the SP by more than the set amount and that the program is waiting for the process to catch up.

In a **Dwell** it freezes the dwell time if the difference between the SP and PV exceeds the set limits.

In both cases it guarantees the correct soak period for the product.

Each program can be configured with a holdback value. Each segment determines the holdback function.

Holdback will cause the execution time of the program to extend, if the process cannot match the demanded profile.

Holdback state will not change the user's access to the parameters. The parameters will behave as if in the RUN state.

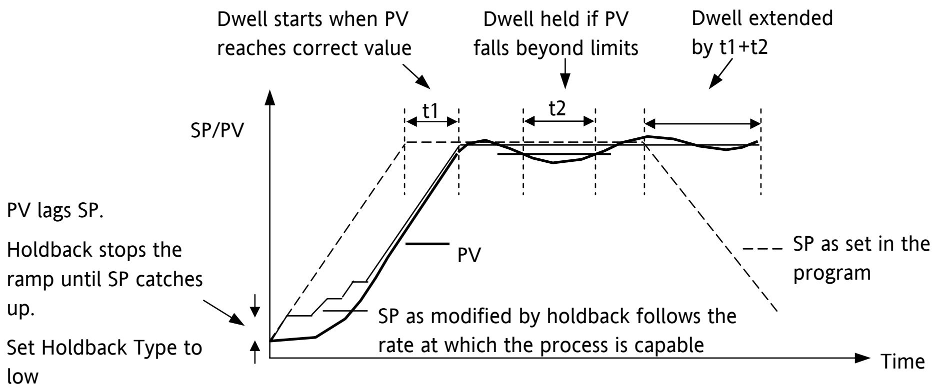


Figure 18-4: Effect of Holdback to Produce Guaranteed Soak

The above diagram demonstrates that the demanded setpoint (SP) will only change at the rate specified by the program when the PV's deviation is less than the holdback value. When the Deviation between the setpoint and PV is greater than the holdback value (Holdback Val) the setpoint ramp will pause until the deviation returns to within the band.

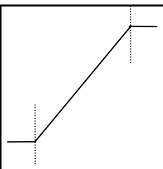
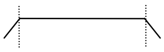
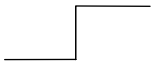
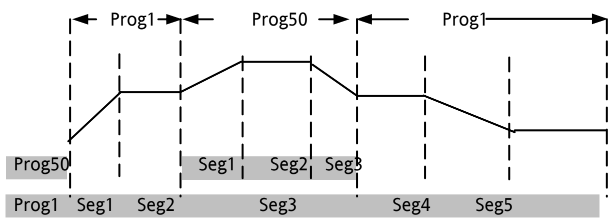
The next segment will not start until the deviation between Setpoint and PV is less than the holdback value.

Four types of Holdback are available:-

None	Holdback is disabled for this segment.
High	Holdback is entered when the PV is greater than the Setpoint **plus** Holdback Val.
Low	Holdback is entered when the PV is lower than the Setpoint **minus** Holdback Val.
Band	Holdback is entered when the PV is **either** greater than the Setpoint **plus** Holdback Val **or** lower than the Setpoint **minus** Holdback Val

18.1.11 Segment Types

A segment may be set as:-

Ramp		A Ramp segment provides a controlled change of setpoint from an original to a target setpoint. The duration of the ramp is determined by the rate of change specified. Two styles of ramp are possible in the range, Ramp-Rate or Time-To-Target. The ramp is specified by the target setpoint and the desired ramp rate. The ramp rate parameter is presented in engineering units (°C, °F, Eng.) per real time units (Seconds, Minutes or Hours). If the units are changed, all ramp rate are re-calculated to the new units and clipped if necessary.
Dwell		The setpoint remains constant for a specified period at the specified target. The operating setpoint of a dwell is inherited from the previous segment.
Step		The setpoint changes instantaneously from its current value to a new value at the beginning of a segment. A Step segment has a minimum duration of 1 second.
Call		A CALL segment may only be selected in instruments offering multiple programs. The segment allows programs to be nested within each other. To prevent re-entrant programs from being specified, only higher number programs may be called from a lower program. i.e. program 1 may call programs 2 through 50, but program 49 may only call program 50. When a CALL segment is selected the operator may specify how many cycles the called program will execute. The number of cycles is specified in the calling program. If a called program has a number of cycles specified locally, they will be ignored. A CALL segment will not have a duration, a CALL segment will immediately transfer execution to the called program and execute its first segment. Called programs do not require any modification, the calling program treats any END segments as return instructions. The example shows Prog 50 (Ramp/Dwell/Ramp) inserted in place of segment 3/Program1. Prog 50 can be made to repeat using the 'Cycles' parameter. 
End		A program may contain one End segment. This allows the program to be truncated to the number of segments required. The end segment can be configured to have an indefinite dwell or to reset the program. This is selectable by the user. If a number of program cycles are specified for the program, then the End segment is not executed until the last cycle has completed.

18.1.12 Power Fail Recovery

In the event of power fail to the controller, a strategy may be set in configuration level, which defines how the controller behaves on restoration of the power. These strategies include:

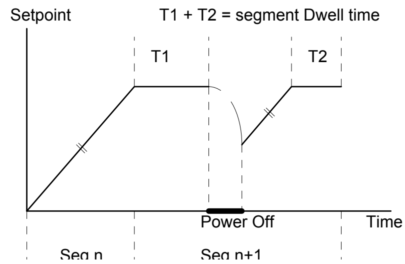
Continue	The program setpoint returns immediately to its last value prior to the power down. This may cause full power to be applied to the process for a short period to heat the process back to its value prior to the power failure.
Ramp back	This will servo the program setpoint to the measured value (the PV Input parameter value), then return to the target setpoint at the current (or previous) ramp rate. The setpoint is not allowed to step change the program setpoint. The outputs will take the state of the segment which was active before power was interrupted.
Reset	The process is aborted by resetting the program. All event outputs will take the reset state.

18.1.12.1 Ramp back (Power fail during Dwell segments.)

If the interrupted segment was a Dwell, then the ramp rate will be determined by the previous ramp segment.

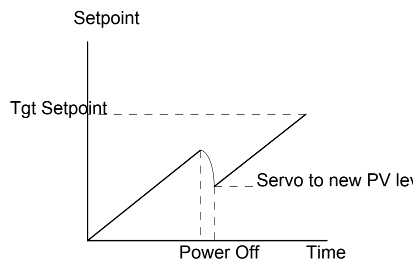
On achieving the Dwell setpoint, the dwell will continue from the point at which the power was interrupted.

Note: If a previous ramp segment does not exist, i.e. the first segment of a program is a dwell, then the Dwell will continue at the "servo to PV" setpoint.



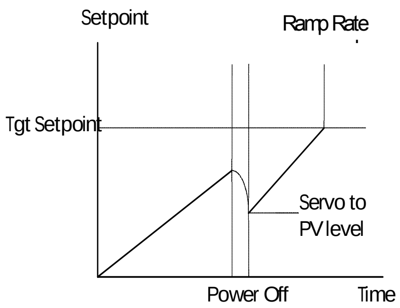
18.1.12.2 Ramp back (power fail during Ramp segments)

If the interrupted segment was a ramp, then the programmer will servo the program setpoint to the PV, then ramp towards the target setpoint at the previous ramp rate. Previous ramp rate is the ramp rate at power fail.



18.1.12.3 Ramp back (power fail during Time-to-target segments)

If the programmer was defined as a Time-to-Target programmer then when the power is returned the previous ramp rate will be recovered. The Time remaining will be recalculated. The rule is to maintain RAMP RATE, but alter TIME REMAINING.



18.1.13 Sync mode

This mode will allow two or more programmers to by synchronised together. This means that the start of each segment (excluding the first) will begin at the same time. Two or more instruments may be synchronised by wiring the "end of segment" and "sync input" parameters between units. (see diagram below).

Set "SyncMode" to Yes

Wire instruments as follows :-

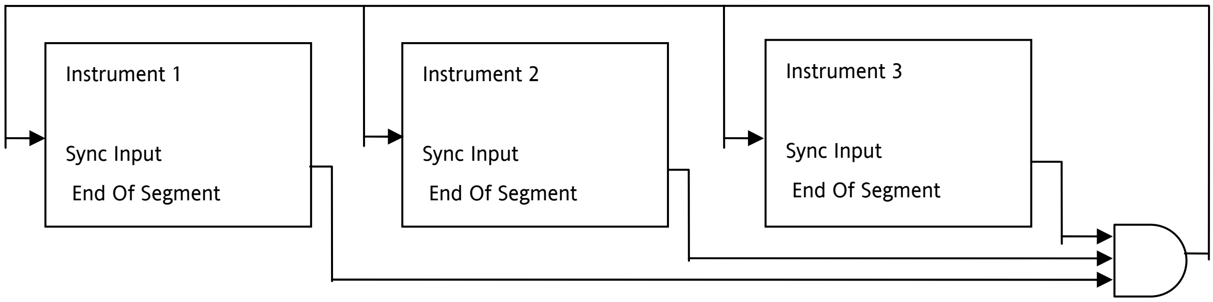


Figure 18-5: Synchronisation of three controllers

At the end of a segment, the program will be put into a temporary hold state (program status will continue to show that the program is running) and the End_of_Segment parameter will be true. Once all segments have completed, the SyncInput goes high and the next segment is started.

If the "SyncMode" is disabled, the "End_of_Segment" parameter is guaranteed to be true for 1 tick at the end of every segment.

18.2 Configuring the Programmer

Programmer.1.Setup contains the general configuration settings for the Programmer Block. Programs are created and stored in the **Program** Folder. Once a Program exists it can be run using the parameters in the **Programmer.1.Run** folder..

Folder – Programmer.1		**Sub-folder: Setup**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Units	Units of the Output			None	Conf
Resolution	Programmer Output resolution	X to X.XXXX			Conf
PVIn	The programmer uses the PV input for a number of functions In holdback, the PV is monitored against the setpoint, and if a deviation occurs the program is paused. The programmer can be configured to start its profile from the current PV value (servo to PV). The programmer monitors the PV value for Sensor Break. The programmer holds in sensor break.	The PV Input is normally wired from the loop TrackPV parameter. Note: This input is automatically wired when the programmer and loop are enabled and there are no existing wires to track interface parameters. Track interface parameters are Programmer.Setup, PVInput, SPInput, Loop.SP, AltSP, Loop.SP, AltSPSelect.			Conf
SPIn	The programmer needs to know the working setpoint of the loop it is trying to control. The SP input is used in the servo to setpoint start type.	SP Input is normally wired from the loop Track SP parameter as the PV input.			Conf
Servo	The transfer of program setpoint to PV Input (normally the Loop PV) or the SP Input (normally the Loop setpoint).	PV SP	See also section 18.1.5.		Conf
PowerFailAct	Power fail recovery strategy	Ramp Reset Cont	See section 18.1.12.		Conf
SyncIn	The synchronise input is a way of synchronising programs. At the end of a segment the programmer will inspect the sync. input, if it is True (1) then the programmer will advance to the next segment. It is typically wired from the end of segment output of another programmer. Only appears if 'SyncMode' = 'Yes'	0 1	This will normally be wired to the 'End of Seg' parameter as shown in section 18.1.13.		Oper
Max Events	To set the maximum number of output events required for the program. This is for convenience to avoid having to scroll through unwanted events when setting up each segment	1 to 8			Conf
SyncMode	Allows multiple controllers to be synchronised at the end of each segment	No Yes	Sync output disabled Sync output enabled		Conf
Prog Reset	Flag showing reset state	No/Yes	Can be wired to logic inputs to provide remote program control		Oper
Prog Run	Flag showing run state	No/Yes			Oper
Prog Hold	Flag showing hold state	No/Yes			Oper
AdvSeg	Set output to target setpoint and advance to next segment	No/Yes			Oper
SkipSeg	Skip to the next setpoint and start the segment at the current output value.	No/Yes			Oper
EventOut1 to 8	Flags showing event states	No/Yes			R/O
End of Seg	Flag showing end of segment state	No/Yes			R/O

18.3 To Select, Run, Hold or Reset a Program

The 'Run' folder allows an existing program to be selected and run. The folder also shows the current program status

Folder – Programmer.1		Sub-folder: Run		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
CurProg	Current Program Number	0 to 50. Change only when Programmer is in Reset.	0	Oper R/O
CurrSeg	Current Running Segment	1 to 255	1	R/O
ProgStatus	Program Status	Reset – Run – Hold – Holdback – End –		Oper
PSP	Programmer Setpoint		0	R/O
CyclesLeft	Number of Cycles Remaining	0 to 1000	0	R/O
CurSegType	Current Segment type	End Rate Time Dwell Step Call	End	R/O
SegTimeLeft	Segment Time Remaining	Hr Min Sec Millisec	0	R/O
ResetEventOP	Reset Event Outputs	0 to 255, each bit resets its corresponding output	0	Oper
SegTarget	Current Target Setpoint Value			R/O
SegRate	Segment Ramp Rate	0.1 to 9999.9	0	R/O
ProgTimeLeft	Program Time Remaining	Hrs Min Sec Millisec	0	R/O
FastRun	Fast Run	No (0) Normal Yes (1) Program executes at 10 times real time	No	Conf
EndOutput	End Output	Off (0) Program not in End On (1) Program at End	Off	R/O
EventsOut	Event Outputs	0 to 255, each bit represents an output.	0	R/O

18.4 Creating a Program

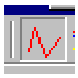
A folder exists for each Program containing a few key parameters listed below. This folder would normally be viewed via the iTools Program Editor under the Program Parameters tab. The Program Editor is used to create the segments of Program itself using the Segment Editor tab.

Folder – Program		**Sub-folder: 1 to 50**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Name	Program Name	Up to 8 characters		Null	Oper
Holdback Value	Deviation between SP and PV at which holdback is applied. This value applies to the whole program.	Minimum setting 0		0	Oper
Ramp Units	Time units applied to the segments	Sec Min Hour	Seconds Minutes Hours	sec	Oper
Cycles	Number of times the whole program repeats	Cont (0) 1 to 999	Repeats continuously Program executes once to 999 times	1	Oper

18.5 Program Editor

The Program Editor in iTools provides the method of entering and editing programs directly in the controller. Setpoint programs can be created graphically, stored and downloaded into the controller. From the iTools menu select 'Program Editor OR Press Programmer to create/edit a Program.

18.5.1 Analog View

This view is used for editing the analogue setpoints. 

1. Select a program number using Program: 1 - 2 in this example.
2. Double click Program Name and enter a name for the program - "Example"
3. Right click in the blank area and choose 'Add Segment'

Segment Type	Description	Parameters	Values
End	Ends Program	Reset	Reset – returns to Loop setpoint Dwell – remains at final setpoint
Rate	Ramp at a rate	Target SP Ramp rate	SP range 0.1 – 9999.9
Time	Ramp to a target over an interval	Target SP Duration	SP range hh:mm:ss
Dwell	Soak at a fixed SP	Duration	hh:mm:ss
Call	Call another Program	Program Number No. of cycles	1 to 50 1 to 999

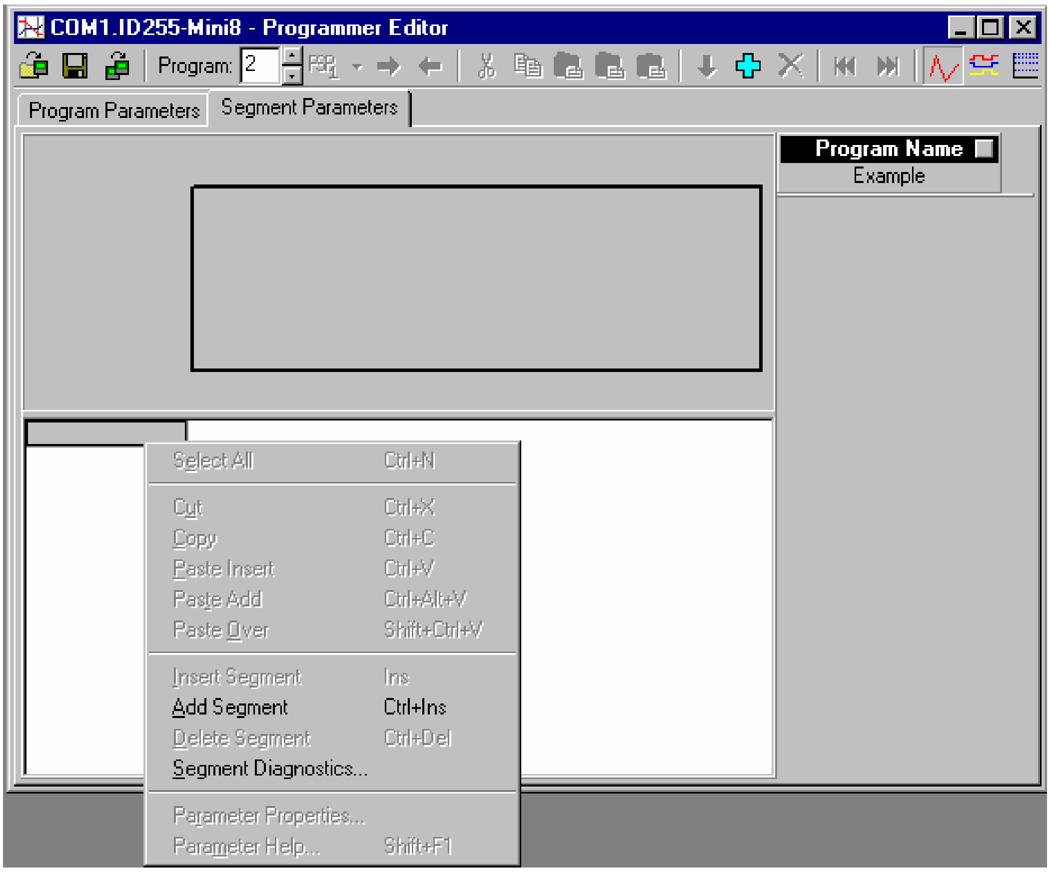


Figure 18-6: Blank Programmer editor – Right Click to add segment

4. Use the drop down to select segment type. Each segment type has the necessary parameters to suit.

5. Right click to insert more segments. End with an 'End' segment.

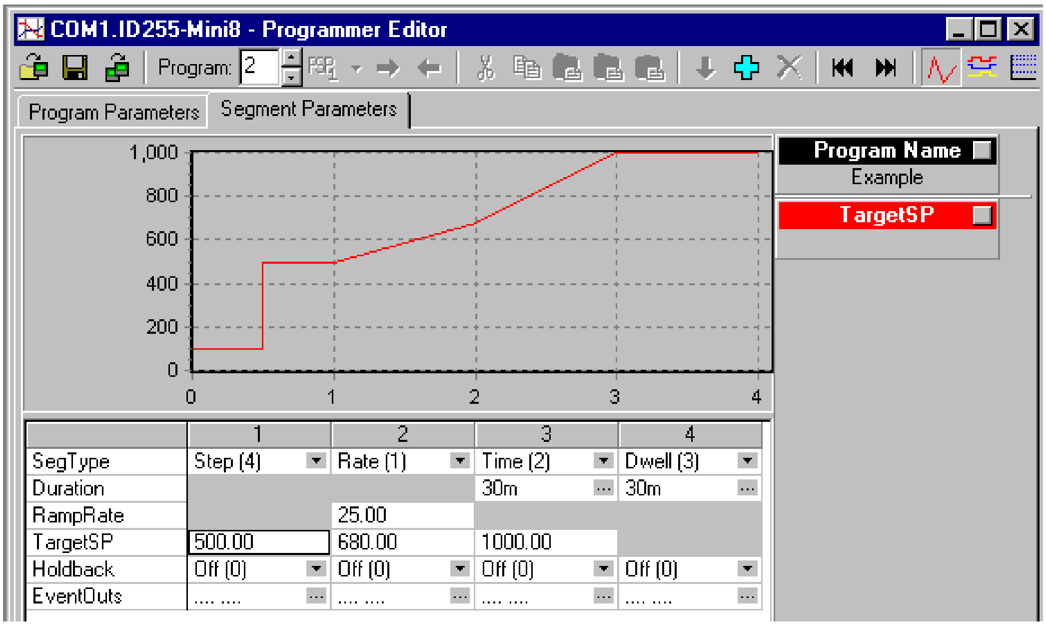
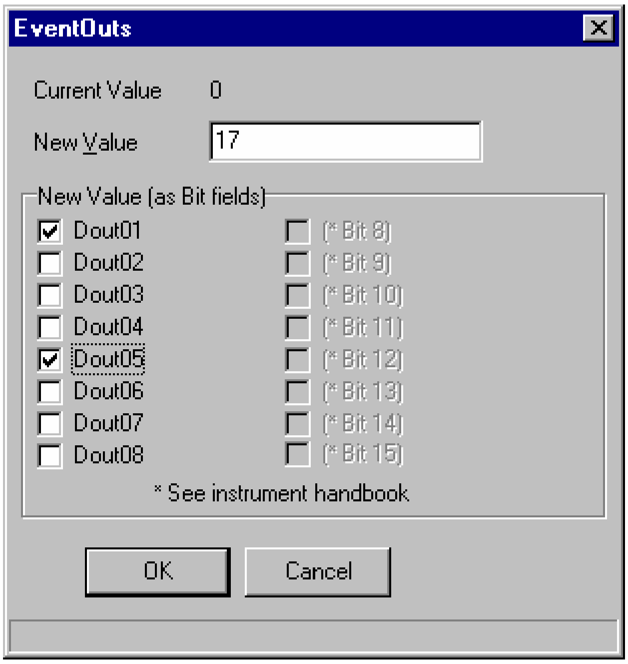


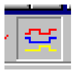
Figure 18-7: Spreadsheet Editor with 4 different segment types

The dots in EventsOut show which outputs are on in each segment.

6. Click on 'EventsOuts' to set up the event outputs for each segment.



18.5.2 Digital View

Alternatively click the icon and the Digital Editor is shown (or hit Cntrl D) 

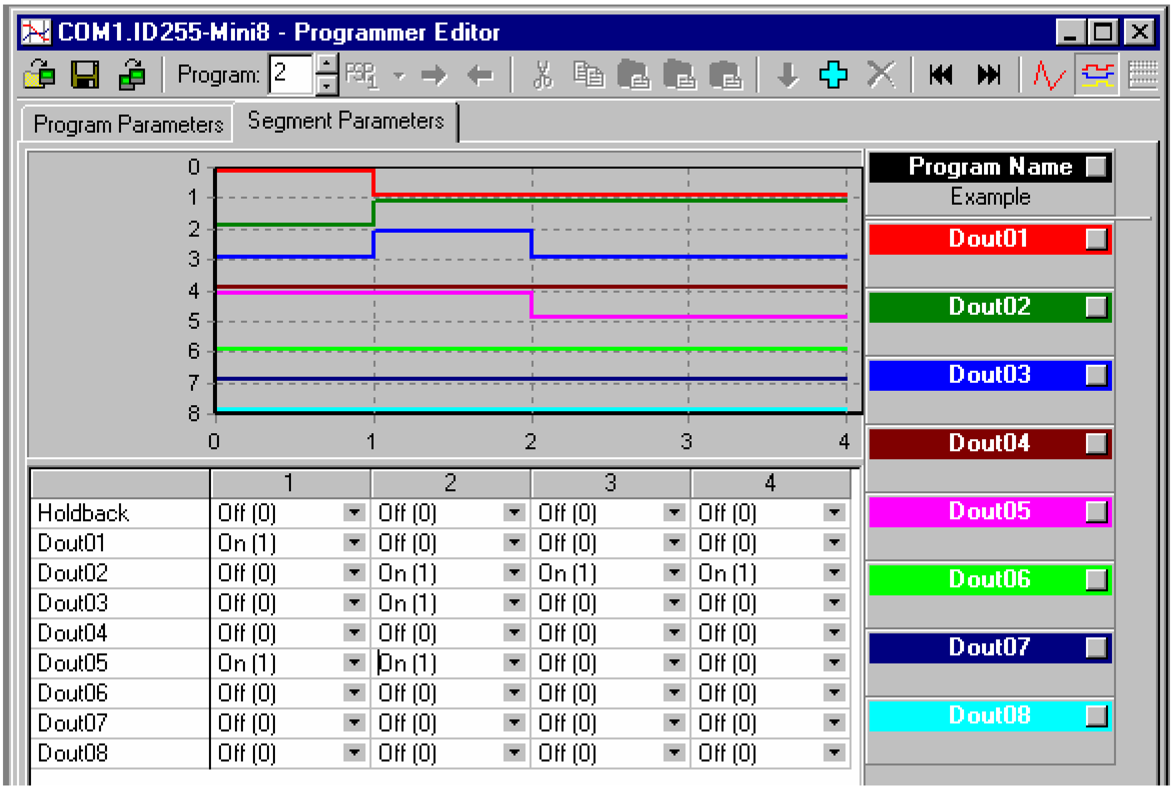
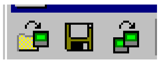


Figure 18-8: Digital Editor showing event outputs

7. Once the program is complete it may be saved to file, or loaded to an instrument.



If you are online to an instrument the program is already 'loaded'. The only option is to save it to file. This example would be saved as 'Example.uip'

If you are working offline a program can be loaded from or saved to disk. If there are instruments connected the program can be sent to them one by one.

18.5.3 Printing a Program

☺ If you select all segments, Cntrl N (or right click 'Select All') and copy spreadsheet cells they are put on the clipboard as tab separated values which can be pasted into Microsoft Excel.

There is no direct printing support in the Programmer Editor, but you can generate a report using Microsoft Excel as follows:

- Right click on the graph and choose 'Copy Chart'.
- Open a new spreadsheet in Excel and paste the chart, position to taste.
- Go back to the Programmer Editor and Choose 'Edit|Select All' followed by 'Edit|Copy'.
- Switch to Excel, choose the top left cell for the segment data and then choose 'Edit|Paste'.
- Optionally delete any columns that have no settings and format the cells.
- Print the spreadsheet.

The program is listed down rather than across the page so long programs can be printed.

18.6 Wiring the Programmer Function Block.

The Programmer block is invariably used with the Loop blocks. When a programmer block is placed on the graphical wiring editor it will **automatically** make the essential connections between itself and the Loop1 block.

These connections ensure that the program setpoint goes to the loop and that 'servo' and other program options work correctly.

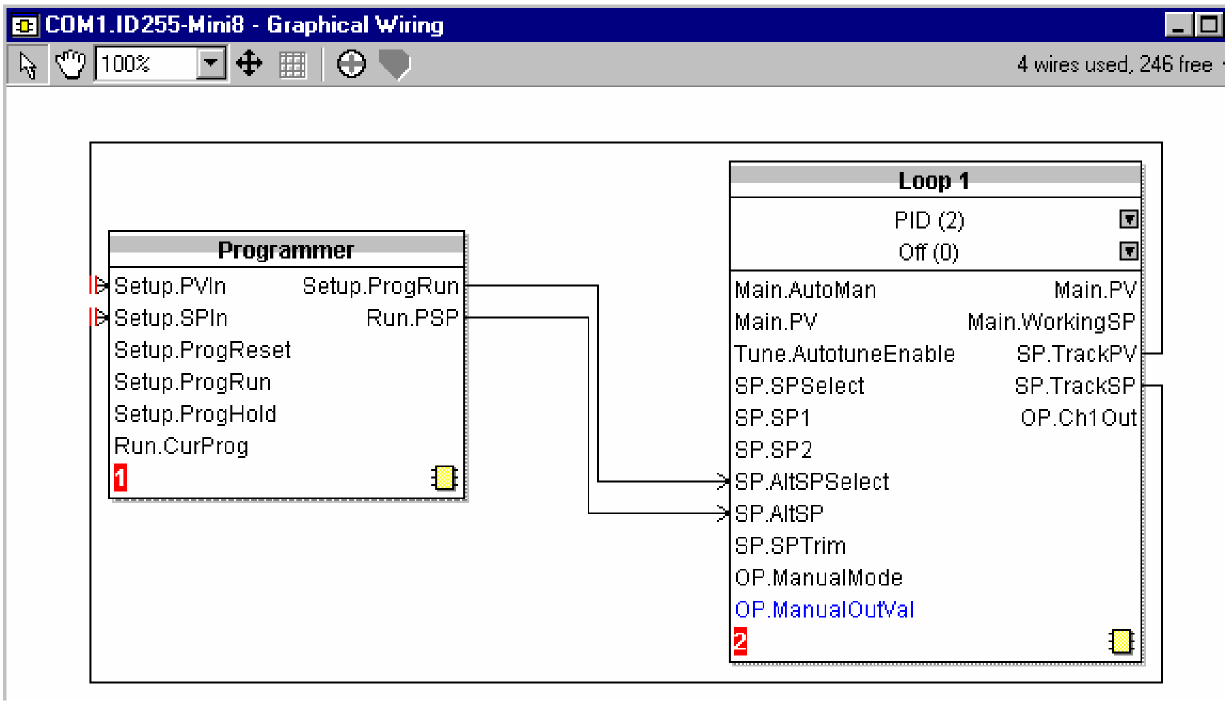


Figure 18-9: Wiring Programmer to Loop Block

19. CHAPTER 19 SWITCH OVER

This facility is commonly used in temperature applications which operate over a wide temperature range. A thermocouple may be used to control at lower temperatures and a pyrometer then controls at very high temperatures. Alternatively two thermocouples of different types may be used.

The diagram below shows a process heating over time with boundaries which define the switching points between the two devices. The higher boundary (2 to 3) is normally set towards the top end of the thermocouple range and this is determined by the 'Switch Hi' parameter. The lower boundary (1 to 2) is set towards the lower end of the pyrometer (or second thermocouple) range using the parameter 'Switch Lo'. The controller calculates a smooth transition between the two devices.

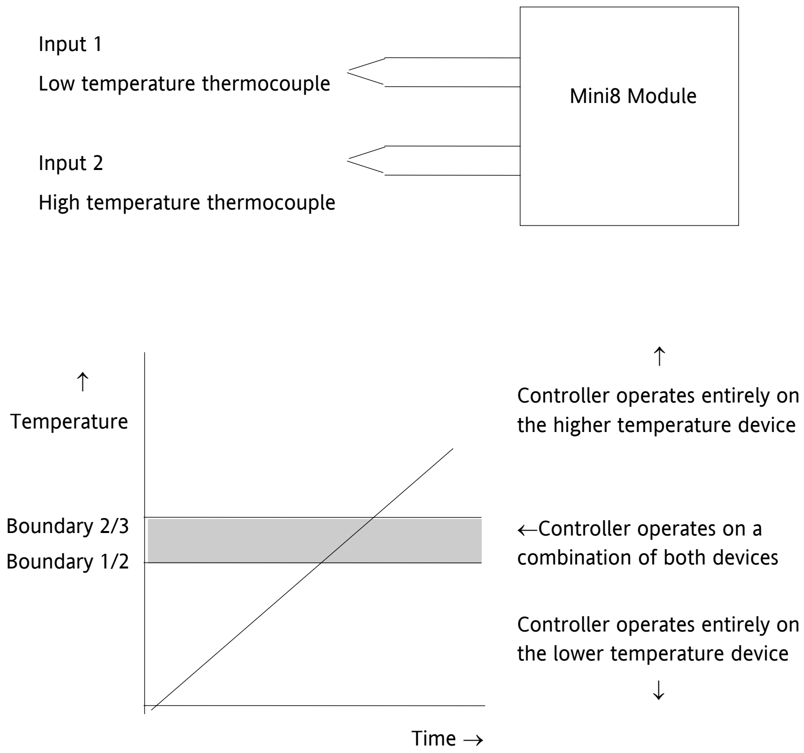


Figure 19-1: Thermocouple to Pyrometer Switching

19.1.1 Example: To Set the Switch Over Levels

Set Access to configuration level

1. Open the 'SwOver' Folder
2. Set 'Switch Hi' to a value which is suitable for the pyrometer (or high temperature thermocouple) to take over the control of the process
3. Set 'Switch Lo' to a value which is suitable for the low temperature thermocouple to control the process

19.1.2 Switch Over Parameters

Folder – SwitchOver		Sub-folders: .1			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
InHigh	Sets the high limit for the switch over block. It is the highest reading from input 2 since it is the high range input sensor.	Input range			Oper
InLow	Sets the low limit for the switch over block. It is the lowest reading from input 1 since it is the low range input sensor				Oper
Switch High	Defines the high boundary of the switchover region	Between Input Hi and Input Lo			Oper
Switch Low	Defines the low boundary of the switchover region.				Oper
In1	The first input value. This must be the low range sensor.	These will normally be wired to the thermocouple/pyrometer input sources via the PV Input or Analogue Input Module. The range will be the range of the input chosen.			R/O if wired
In2	The second input value. This must be the high range sensor				R/O if wired
Fallback Value	In the event of a bad status, the output may be configured to adopt the fallback value. This allows the strategy to dictate a safe output in the event of a fault being detected	Between Input Hi and Input Lo		0.0	Oper
Fallback Type	Fall back type	Clip Bad Clip Good Fall Bad Fall Good Upscale Downscale		Clip Bad	Conf
SelectIn	Indicates which input is currently selected	Input 1 Input 2	0: Input 1 has been selected 1: Input 2 has been selected 2: Both inputs are used to calculate the output		R/O
ErrMode	The action taken if the selected input is BAD	UseGood ShowBad	0: Assumes the value of a good input If the currently selected input is BAD the output will assume the value of the other input if it is GOOD 1: If selected input is BAD the output is BAD	Use Good	Conf
Out	Output produced from the 2 input measurements				R/O
Status	Status of the switchover block	Good Bad			R/O

20. CHAPTER 20 TRANSDUCER SCALING

The Mini8 controller includes two transducer calibration function blocks that may be enabled in configuration level in the '**Instrument/Options**' folder. These are a software function blocks that provide a method of offsetting the calibration of the input when compared to a known input source. Transducer scaling is often performed as a routine operation on a machine to take out system errors. For this reason it can be carried out in operator mode.

Transducer scaling can be applied to any input or derived input, i.e. the PV Input or Analogue Input fitted in one of the module slots. These can be wired in configuration level to the above inputs.

Three types of calibration are explained in this chapter:-

- Auto-tare
- Load Cell Calibration
- Comparison Calibration

20.1 Auto-Tare Calibration

The auto-tare function is used, for example, when it is required to weigh the contents of a container but not the container itself.

The procedure is to place the empty container on the weigh bridge and 'zero' the controller. Since it is likely that following containers may have different tare weights the auto-tare feature is always available.

Further parameters are available which are used to pre-configure the tare measurement or for interrogation purposes. Tare calibration may be carried out no matter what type of transducer is in use.

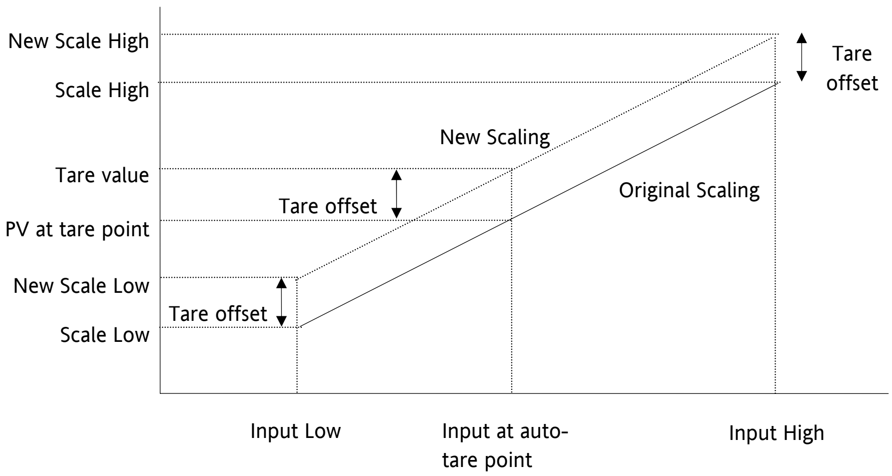


Figure 20-1: Effect of Auto Tare

20.2 Load Cell

A load cell provides an analogue output which can be in Volts, milli-Volts or milli-Amps. This may be connected to the PV Input or Analogue Input.

When no load is placed on the cell the output is normally zero. However, in practice there may be a residual output and this can be calibrated out in the controller.

The high end is calibrated by placing a reference weight on the load cell and performing a high end calibration in the controller.

20.3 Comparison Calibration

Comparison calibration is used to calibrate the controller against a second reference instrument.

The load is removed (or taken to a minimum) from the reference device. The controller low end calibration is done using the 'Cal Enable' parameter and entering the reading from the reference instrument.

Add a weight and when the reading has become stable select the 'Cal Hi Enable' parameter then enter the new reading from the reference instrument.

20.4 Transducer Scaling Parameters

Folder – Txdr		**Sub-folders: .1 or .2**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Cal Type	Used to select the type of transducer calibration to perform See descriptions at the beginning of this chapter.	1: Off 1: Shunt 2: Load Cell 3: Compare	Transducer type unconfigured Shunt calibration Load Cell Comparison	Off	Conf
Cal Enable	To make the transducer ready for calibration. Must be set to Yes to allow calibration to be done at L1. This includes Tare Cal.	No Yes	Not ready Ready	No	Conf
Range Max	The maximum permissible range of the scaling block	Range min to 99999		1000	Conf
Range Min	The minimum permissible range of the scaling block	-19999 to Range max		0	Conf
Start Tare	Begin tare calibration	No Yes	 Start tare calibration	No	Oper if 'Cal Enable' = 'Yes'
Start Cal	Starts the Calibration process. Note: for Load Cell and Comparison calibration 'Start Cal' starts the first calibration point.	No Yes	 Start calibration	No	Oper if 'Cal Enable' = 'Yes'
Start HighCal	For Load Cell and Comparison calibration the 'Start High Cal' must be used to start the second calibration point.	No Yes	 Start high calibration	No	Oper if 'Cal Enable' = 'Yes'

Folder – Txdr		**Sub-folders: .1 or .2**			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Clear Cal	Clears the current calibration constants. This returns the calibration to unity gain	No Yes	 To delete previous calibration values	No	Oper
Tare Value	Enter the tare value of the container				Conf
InHigh	Sets the scaling input high point				Oper
InLow	Sets the scaling input low point				Oper
Scale High	Sets the scaling output high point. Usually the same as the 'Input Lo'				Oper
Scale Low	Sets the scaling output low point. Usually 80% of 'Input Hi'				Oper
Cal Band	The calibration algorithms use the threshold to determine if the value has settled. When switching in the shunt resistor, the algorithm waits for the value to settle to within the threshold before starting the high calibration point.				Conf
CalAdjust	The adjust is used in the Comparison Calibration method.	When edited, the Adjust parameter can be set to the desired value. On confirm, the new adjust value is used to set the scaling constants			Oper
ShuntOut	Indicates when the internal shunt calibration resistor is switched in. Only appears if 'Cal Type' = 'Shunt'	Off On	Resistor not switched in Resistor switched in		Oper
Cal Active	Indicates calibration taking place	Off On	Inactive Active		R/O
InVal	The input value to be scaled.	-9999.9 to 9999.9			Oper
OutVal	The Input Value is scaled by the block to produce the Output Value				Oper
Status	The status of the output accounting for sensor fail signals passed to the block and the state of the scaling.	Good Bad			Conf
Cal Status	Indicates the progress of calibration	0: Idle 1: Active 2: Passed 3: Failed	No calibration in progress Calibration in progress Calibration Passed Calibration Failed		L1 R/O

20.4.1 Parameter Notes

Enable Cal	This may be wired to a digital input for an external switch. If not wired, then the value may be changed. When enabled the transducer parameters may be altered as described in the previous sections. When the parameter has been turned On it will remain on until turned off manually even if the controller is powered cycled.
Start Tare	This may be wired to a digital input for an external switch. If not wired, then the value may be changed.
Start Cal	This may be wired to a digital input for an external switch. If not wired, then the value may be changed. It starts the calibration procedure for: Shunt Calibration The low point for Load Cell Calibration The low point for Comparison Calibration
Start Hi Cal	This may be wired to a digital input for an external switch. If not wired, then the value may be changed. It starts:- The high point for Load Cell Calibration The high point for Comparison Calibration
Clear Cal	This may be wired to a digital input for an external switch. If not wired, then the value may be changed. When enabled the input will reset to default values. A new calibration will overwrite the previous calibration values if Clear Cal is not enabled between calibrations.

20.4.2 Tare Calibration

The Mini8 controller has an auto-tare function that is used, for example, when it is required to weigh the contents of a container but not the container itself.

The procedure is to place the empty container on the weighbridge and 'zero' the controller. The procedure is as follows:-

1. Place container on weighbridge
2. Go to Txdr.1 (or .2) Folder.
3. Transducer calibration Type must be 'Load Cell'.
4. CalEnable must be set to 'Yes'.
5. Set StartTare to 'yes'
6. The controller automatically calibrates the to the tare weight which is measured by the transducer and stores this value.
7. During this measurement Cal Status will show progress. If the cal fails it is probably an 'out of range' problem.

20.4.3 Load Cell

A load cell output must be within the range 0 to 77 mV to go into a TC8 input. Use a shunt for mA inputs, mV can possibly go direct, Volt inputs must use a potential divider.

To calibrate a load cell.

1. Remove all load from the transducer to establish a zero reference.
2. Go to Txdr.1 (or .2) Folder.
3. Transducer calibration Type must be 'Load Cell'.
4. CalEnable must be set to 'Yes'.
5. Set Start Cal to 'yes'
6. The controller will calibrate the low point.
7. Set StartHighCal to 'yes'
8. The controller will calibrate the high point.

Cal Status advises progress and result.

20.4.4 Comparison Calibration

Comparison calibration is used to calibrate the input against a second reference instrument. Typically this might be a local display on the weighing device itself.

To calibrate against a known reference source:-

1. Add a load at the low end of the scale range
2. Go to Txdr.1 (or .2) Folder.
3. Transducer calibration Type must be 'Comparision'.
4. CalEnable must be set to 'Yes'.
5. Enter the reading from the reference instrument into 'Cal Adjust'.
6. Add a load at the high end of the scale.
7. Set StartHighCal to 'yes'
8. The controller will calibrate the high point.

Cal Status advises progress and result.

21. CHAPTER 21 USER VALUES

User values are registers provided for use in calculations. They may be used as constants in equations or temporary storage in extended calculations. Up to 32 User Values are available provided they have been enabled in the ''Instrument/Options' folder. They are arranged in 4 groups of 8. Each User Value can then be set up in the **'UserVal'** folder.

21.1 User Value Parameters

Folder – UsrVal		**Sub-Folders: .1 to .32**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Units	Units assigned to the User Value	None Abs Temp °C/°F/°K, V, mV, A, mA, PH, mmHg, psi, Bar, mBar, %RH, %, mmWG, inWG, inWW, Ohms, PSIG, %O2, PPM, %CO2, %CP, %/sec, RelTemp °C\°F\°K(rel), Custom 1, Custom 2, Custom 3, Custom 4, Custom 5, Custom 6, sec, min, hrs,		Conf
Resolution	Resolution of the User Value	XXXXX to X.XXXX		Conf
High Limit	The high limit may be set for each user value to prevent the value being set to an out-of-bounds value.			Oper
Low Limit	The low limit of the user value may be set to prevent the user value from being edited to an illegal value. This is important if the user value is to be used as a setpoint.			Oper
Val	To set the value within the range limits	See note 1		Oper
Status	Can be used to force a good or bad status onto a user value. This is useful for testing status inheritance and fallback strategies.	Good Bad — See note 1		Oper

Note 1.

If 'Val' is wired into but 'Status' is not, then, instead of being used to force the Status it will indicate the status of the value as inherited form the wired connection to 'Val'.

22. CHAPTER 22 CALIBRATION

In this chapter calibration refers to calibration of the TC input of the TC8 module. Calibration is accessed using the 'Cal State' parameter that is only available in configuration level. Since the controller is calibrated during manufacture to traceable standards for every input range, it is not necessary to calibrate the controller when changing ranges. Furthermore, a continuous automatic check and correction of the calibration during the controllers' normal operation means that it is calibrated for life.
However, it is recognised that, for operational reasons, it may be a requirement to check or re-calibrate the controller. This new calibration is saved as a User Calibration. It is always possible to revert to the factory calibration if necessary.

☺ Tip: Consider using the '**Offset**' parameter for User Cal (e.g. Mod.1.Offset). This can be set to correct any measured difference between the Mini8 given PV and a calibration value obtained from another source.

22.1 User calibration

22.1.1 Set Up

No pre-calibration warm-up is required.
As calibration is a single-point on 8 channels, quick enough (a few minutes) to avoid self-heating effects, there are no special environmental, mounting position or ventilation requirements for calibration.
Calibration should be performed at a reasonable ambient temperature (15°C to 35°C). Calibration outside these limits will compromise the expected working accuracy.
Every channel of every TC8 card must be individually connected to the calibrator source using thick copper wire (so the sensor-break voltage drop in the wires and source impedance is minimal).
The voltage source, monitor DVM and the target Mini8 should be at the same temperature (to eliminate added series e.m.f. due to thermocouple effects).
The Mini8 must be in Configuration Mode.

22.1.2 Zero Calibration

No "zero" calibration point is required.

22.1.3 Voltage Calibration

1. Set the Calibrator voltage source to an accurate 50.005mV. (The extra 5uV is to compensate for self-heating tempco effect).
2. Connect the 50mV to channel 1
3. Set Mod.1.CalState to HiCal (23) and then select 'Confirm'
4. When complete set CalState to SaveUser(25)

22.1.4 CJC Calibration

No CJC calibration required; the sampled values are ratiometric, providing uncalibrated uncertainty of ±1°C.

22.1.5 Sensor-Break Limit Check

Apply a 900Ω resistor to each channel in turn, Sensor Break Type to 'Low', filter to off (0). Verify the SBrkValue is greater than 24.0 and less than 61.0

22.2 To Return to Factory Calibration

To clear the User calibration and restore the calibration from the factory.

1. Put Mini8 into Configuration Mode
2. Set the 'Mod.1.Calibration State' to LoadFact (25).
3. Return Instrument to Operating Mode.

22.3 Calibration Parameters

List Header - IO		Sub-headers: Mod.1 to Mod.32			
Name	**Parameter Description**	**Value**		**Default**	**Access Level**
Cal State	Calibration state of the input	Idle Hi-50mV Load Fact Save User Confirm Go Busy Passed Failed	Normal operation High input calibration for mV ranges Restore factory calibration values Save the new calibration values To start the calibration procedure when one of the above has been selected Starting the automatic calibration procedure Calibration in progress Calibration successful Calibration unsuccessful	Idle	Conf
Status	PV Status The current status of the PV.	0 1 2 3 4 5 6	Normal operation Initial startup mode Input in sensor break PV outside operating limits Saturated input Uncalibrated channel No Module		R/O

The above list shows the values of CalState, which appear during a normal calibration procedure. The full list of possible values follows – the number is the enumeration for the parameter.

1: Idle
2: Low calibration point for Volts range
3: High calibration point for Volts range
4: Calibration restored to factory default values
5: User calibration stored
6: Factory calibration stored
11: Idle
12: Low calibration point for HZ input
13: High calibration point for the HZ input
14: Calibration restored to factory default values
15: User calibration stored
16: Factory calibration stored
20: Calibration point for factory rough calibration
21: Idle
22: Low calibration point for the mV range
23: Hi calibration point for the mV range
24: Calibration restored to factory default values
25: User calibration stored
26: Factory calibration stored
30: Calibration point for factory rough calibration
31: Idle
32: Low calibration point for the mV range
33: High calibration point for the mV range
34: Calibration restored to factory default values
35: User calibration stored
36: Factory calibration stored
41: Idle
42: Low calibration point for RTD calibration (150 ohms)
43: Low calibration point for RTD calibration (400 ohms)
44: Calibration restored to factory default values
45: User calibration stored
46: Factory calibration stored
51: Idle
52: CJC calibration used in conjunction with Term Temp parameter
54: Calibration restored to factory default values
55: User calibration stored
56: Factory calibration stored
200: Confirmation of request to calibrate
201: Used to start the calibration procedure
202: Used to abort the calibration procedure
210: Calibration point for factory rough calibration
212: Indication that calibration is in progress
213: Used to abort the calibration procedure
220: Indication that calibration completed successfully
221: Calibration accepted but not stored
222: Used to abort the calibration procedure
223: Indication that calibration failed

23. APPENDIX A MODBUS SCADA TABLE

These parameters are single register Modbus values for use with Third Party Modbus masters in SCADA packages or plcs. Scaling of the parameters has to be configured – the Modbus master scaling has to match the Mini8 parameter resolution to ensure the decimal point is in the correct position.

23.1 Comms Table

The tables that follow do not include every parameter in the Mini8. The Comms Table is used to make most parameters available at any SCADA address.

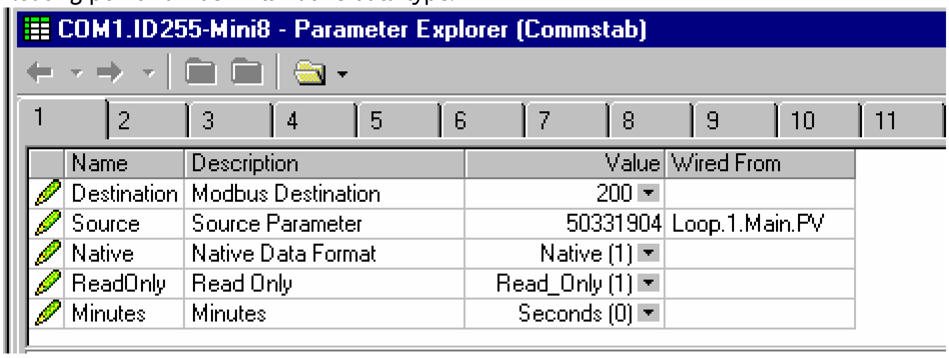
Folder – Commstab		**Sub-folders: .1 to .250**		
Name	**Parameter Description**	**Value**	**Default**	**Access Level**
Destination	Modbus Destination	Not Used 1 to 16011	Not used	Conf
Source	Source Parameter	Taken from source parameter		Conf
Native	Native Data Format	0 Integer 1 Native	Integer	Conf
ReadOnly	Read Only Read/Write only if source is R/W	0 Read/Write 1 Read Only	R/W	Conf
Minutes	Minutes Units in which time is scaled.	0 Seconds 1 Minutes	Seconds.	Conf

Entering a value in the Source parameter may be done in two ways:

1 - drag the required parameter into the Source

2 - right click the Source parameter, select Edit Wire and browse to the required parameter.

In the Example below the PV of Loop 1 would be available at addresses 200 and 201 as a two register floating point number - its native data type.



There are 250 comms tables entries available.

23.2 SCADA Table

The parameters in the tables following are available in scaled integer format, accessed via their associated Modbus address.

Wherever possible use an OPC client with the iTools OPCserver as the server. In this arrangement the parameters are all referenced by name and the values are floating point so the decimal point for all parameters is inherited.

Description	ModBus	HEX
Access.CustomerID	4739	1283
Access.InstrumentMode	199	00c7
Alarm.1.Ack	10250	280a
Alarm.1.Block	10246	2806
Alarm.1.Delay	10248	2808
Alarm.1.Hysteresis	10242	2802
Alarm.1.Inhibit	10247	2807
Alarm.1.Latch	10244	2804
Alarm.1.Out	10249	2809
Alarm.1.Reference	10243	2803
Alarm.1.Threshold	10241	2801
Alarm.1.Type	10240	2800
Alarm.2.Ack	10266	281a
Alarm.2.Block	10262	2816
Alarm.2.Delay	10264	2818
Alarm.2.Hysteresis	10258	2812
Alarm.2.Inhibit	10263	2817
Alarm.2.Latch	10260	2814
Alarm.2.Out	10265	2819
Alarm.2.Reference	10259	2813
Alarm.2.Threshold	10257	2811
Alarm.2.Type	10256	2810
Alarm.3.Ack	10282	282a
Alarm.3.Block	10278	2826
Alarm.3.Delay	10280	2828
Alarm.3.Hysteresis	10274	2822
Alarm.3.Inhibit	10279	2827
Alarm.3.Latch	10276	2824
Alarm.3.Out	10281	2829
Alarm.3.Reference	10275	2823
Alarm.3.Threshold	10273	2821
Alarm.3.Type	10272	2820
Alarm.4.Ack	10298	283a
Alarm.4.Block	10294	2836
Alarm.4.Delay	10296	2838
Alarm.4.Hysteresis	10290	2832
Alarm.4.Inhibit	10295	2837
Alarm.4.Latch	10292	2834
Alarm.4.Out	10297	2839
Alarm.4.Reference	10291	2833
Alarm.4.Threshold	10289	2831
Alarm.4.Type	10288	2830
Alarm.5.Ack	10314	284a
Alarm.5.Block	10310	2846
Alarm.5.Delay	10312	2848
Alarm.5.Hysteresis	10306	2842
Alarm.5.Inhibit	10311	2847
Alarm.5.Latch	10308	2844
Alarm.5.Out	10313	2849
Alarm.5.Reference	10307	2843
Alarm.5.Threshold	10305	2841
Alarm.5.Type	10304	2840
Alarm.6.Ack	10330	285a
Alarm.6.Block	10326	2856
Alarm.6.Delay	10328	2858
Alarm.6.Hysteresis	10322	2852
Alarm.6.Inhibit	10327	2857
Alarm.6.Latch	10324	2854

Description	ModBus	HEX
Alarm.6.Out	10329	2859
Alarm.6.Reference	10323	2853
Alarm.6.Threshold	10321	2851
Alarm.6.Type	10320	2850
Alarm.7.Ack	10346	286a
Alarm.7.Block	10342	2866
Alarm.7.Delay	10344	2868
Alarm.7.Hysteresis	10338	2862
Alarm.7.Inhibit	10343	2867
Alarm.7.Latch	10340	2864
Alarm.7.Out	10345	2869
Alarm.7.Reference	10339	2863
Alarm.7.Threshold	10337	2861
Alarm.7.Type	10336	2860
Alarm.8.Ack	10362	287a
Alarm.8.Block	10358	2876
Alarm.8.Delay	10360	2878
Alarm.8.Hysteresis	10354	2872
Alarm.8.Inhibit	10359	2877
Alarm.8.Latch	10356	2874
Alarm.8.Out	10361	2879
Alarm.8.Reference	10355	2873
Alarm.8.Threshold	10353	2871
Alarm.8.Type	10352	2870
Alarm.9.Ack	10378	288a
Alarm.9.Block	10374	2886
Alarm.9.Delay	10376	2888
Alarm.9.Hysteresis	10370	2882
Alarm.9.Inhibit	10375	2887
Alarm.9.Latch	10372	2884
Alarm.9.Out	10377	2889
Alarm.9.Reference	10371	2883
Alarm.9.Threshold	10369	2881
Alarm.9.Type	10368	2880
Alarm.10.Ack	10394	289a
Alarm.10.Block	10390	2896
Alarm.10.Delay	10392	2898
Alarm.10.Hysteresis	10386	2892
Alarm.10.Inhibit	10391	2897
Alarm.10.Latch	10388	2894
Alarm.10.Out	10393	2899
Alarm.10.Reference	10387	2893
Alarm.10.Threshold	10385	2891
Alarm.10.Type	10384	2890
Alarm.11.Ack	10410	28aa
Alarm.11.Block	10406	28a6
Alarm.11.Delay	10408	28a8
Alarm.11.Hysteresis	10402	28a2
Alarm.11.Inhibit	10407	28a7
Alarm.11.Latch	10404	28a4
Alarm.11.Out	10409	28a9
Alarm.11.Reference	10403	28a3
Alarm.11.Threshold	10401	28a1
Alarm.11.Type	10400	28a0
Alarm.12.Ack	10426	28ba
Alarm.12.Block	10422	28b6
Alarm.12.Delay	10424	28b8
Alarm.12.Hysteresis	10418	28b2

Description	ModBus	HEX
Alarm.12.Inhibit	10423	28b7
Alarm.12.Latch	10420	28b4
Alarm.12.Out	10425	28b9
Alarm.12.Reference	10419	28b3
Alarm.12.Threshold	10417	28b1
Alarm.12.Type	10416	28b0
Alarm.13.Ack	10442	28ca
Alarm.13.Block	10438	28c6
Alarm.13.Delay	10440	28c8
Alarm.13.Hysteresis	10434	28c2
Alarm.13.Inhibit	10439	28c7
Alarm.13.Latch	10436	28c4
Alarm.13.Out	10441	28c9
Alarm.13.Reference	10435	28c3
Alarm.13.Threshold	10433	28c1
Alarm.13.Type	10432	28c0
Alarm.14.Ack	10458	28da
Alarm.14.Block	10454	28d6
Alarm.14.Delay	10456	28d8
Alarm.14.Hysteresis	10450	28d2
Alarm.14.Inhibit	10455	28d7
Alarm.14.Latch	10452	28d4
Alarm.14.Out	10457	28d9
Alarm.14.Reference	10451	28d3
Alarm.14.Threshold	10449	28d1
Alarm.14.Type	10448	28d0
Alarm.15.Ack	10474	28ea
Alarm.15.Block	10470	28e6
Alarm.15.Delay	10472	28e8
Alarm.15.Hysteresis	10466	28e2
Alarm.15.Inhibit	10471	28e7
Alarm.15.Latch	10468	28e4
Alarm.15.Out	10473	28e9
Alarm.15.Reference	10467	28e3
Alarm.15.Threshold	10465	28e1
Alarm.15.Type	10464	28e0
Alarm.16.Ack	10490	28fa
Alarm.16.Block	10486	28f6
Alarm.16.Delay	10488	28f8
Alarm.16.Hysteresis	10482	28f2
Alarm.16.Inhibit	10487	28f7
Alarm.16.Latch	10484	28f4
Alarm.16.Out	10489	28f9
Alarm.16.Reference	10483	28f3
Alarm.16.Threshold	10481	28f1
Alarm.16.Type	10480	28f0
Alarm.17.Ack	10506	290a
Alarm.17.Block	10502	2906
Alarm.17.Delay	10504	2908
Alarm.17.Hysteresis	10498	2902
Alarm.17.Inhibit	10503	2907
Alarm.17.Latch	10500	2904
Alarm.17.Out	10505	2909
Alarm.17.Reference	10499	2903
Alarm.17.Threshold	10497	2901
Alarm.17.Type	10496	2900
Alarm.18.Ack	10522	291a
Alarm.18.Block	10518	2916
Alarm.18.Delay	10520	2918
Alarm.18.Hysteresis	10514	2912
Alarm.18.Inhibit	10519	2917
Alarm.18.Latch	10516	2914
Alarm.18.Out	10521	2919
Alarm.18.Reference	10515	2913
Alarm.18.Threshold	10513	2911
Alarm.18.Type	10512	2910
Alarm.19.Ack	10538	292a
Alarm.19.Block	10534	2926
Alarm.19.Delay	10536	2928
Alarm.19.Hysteresis	10530	2922
Alarm.19.Inhibit	10535	2927
Alarm.19.Latch	10532	2924
Alarm.19.Out	10537	2929
Alarm.19.Reference	10531	2923
Alarm.19.Threshold	10529	2921
Alarm.19.Type	10528	2920
Alarm.20.Ack	10554	293a
Alarm.20.Block	10550	2936
Alarm.20.Delay	10552	2938
Alarm.20.Hysteresis	10546	2932
Alarm.20.Inhibit	10551	2937
Alarm.20.Latch	10548	2934
Alarm.20.Out	10553	2939
Alarm.20.Reference	10547	2933
Alarm.20.Threshold	10545	2931
Alarm.20.Type	10544	2930
Alarm.21.Ack	10570	294a
Alarm.21.Block	10566	2946
Alarm.21.Delay	10568	2948
Alarm.21.Hysteresis	10562	2942
Alarm.21.Inhibit	10567	2947
Alarm.21.Latch	10564	2944
Alarm.21.Out	10569	2949
Alarm.21.Reference	10563	2943
Alarm.21.Threshold	10561	2941
Alarm.21.Type	10560	2940
Alarm.22.Ack	10586	295a
Alarm.22.Block	10582	2956
Alarm.22.Delay	10584	2958
Alarm.22.Hysteresis	10578	2952
Alarm.22.Inhibit	10583	2957
Alarm.22.Latch	10580	2954
Alarm.22.Out	10585	2959
Alarm.22.Reference	10579	2953
Alarm.22.Threshold	10577	2951
Alarm.22.Type	10576	2950
Alarm.23.Ack	10602	296a
Alarm.23.Block	10598	2966
Alarm.23.Delay	10600	2968
Alarm.23.Hysteresis	10594	2962
Alarm.23.Inhibit	10599	2967
Alarm.23.Latch	10596	2964
Alarm.23.Out	10601	2969
Alarm.23.Reference	10595	2963
Alarm.23.Threshold	10593	2961
Alarm.23.Type	10592	2960

Description	ModBus	HEX
Alarm.24.Ack	10618	297a
Alarm.24.Block	10614	2976
Alarm.24.Delay	10616	2978
Alarm.24.Hysteresis	10610	2972
Alarm.24.Inhibit	10615	2977
Alarm.24.Latch	10612	2974
Alarm.24.Out	10617	2979
Alarm.24.Reference	10611	2973
Alarm.24.Threshold	10609	2971
Alarm.24.Type	10608	2970
Alarm.25.Ack	10634	298a
Alarm.25.Block	10630	2986
Alarm.25.Delay	10632	2988
Alarm.25.Hysteresis	10626	2982
Alarm.25.Inhibit	10631	2987
Alarm.25.Latch	10628	2984
Alarm.25.Out	10633	2989
Alarm.25.Reference	10627	2983
Alarm.25.Threshold	10625	2981
Alarm.25.Type	10624	2980
Alarm.26.Ack	10650	299a
Alarm.26.Block	10646	2996
Alarm.26.Delay	10648	2998
Alarm.26.Hysteresis	10642	2992
Alarm.26.Inhibit	10647	2997
Alarm.26.Latch	10644	2994
Alarm.26.Out	10649	2999
Alarm.26.Reference	10643	2993
Alarm.26.Threshold	10641	2991
Alarm.26.Type	10640	2990
Alarm.27.Ack	10666	29aa
Alarm.27.Block	10662	29a6
Alarm.27.Delay	10664	29a8
Alarm.27.Hysteresis	10658	29a2
Alarm.27.Inhibit	10663	29a7
Alarm.27.Latch	10660	29a4
Alarm.27.Out	10665	29a9
Alarm.27.Reference	10659	29a3
Alarm.27.Threshold	10657	29a1
Alarm.27.Type	10656	29a0
Alarm.28.Ack	10682	29ba
Alarm.28.Block	10678	29b6
Alarm.28.Delay	10680	29b8
Alarm.28.Hysteresis	10674	29b2
Alarm.28.Inhibit	10679	29b7
Alarm.28.Latch	10676	29b4
Alarm.28.Out	10681	29b9
Alarm.28.Reference	10675	29b3
Alarm.28.Threshold	10673	29b1
Alarm.28.Type	10672	29b0
Alarm.29.Ack	10698	29ca
Alarm.29.Block	10694	29c6
Alarm.29.Delay	10696	29c8
Alarm.29.Hysteresis	10690	29c2
Alarm.29.Inhibit	10695	29c7
Alarm.29.Latch	10692	29c4
Alarm.29.Out	10697	29c9
Alarm.29.Reference	10691	29c3

Description	ModBus	HEX
Alarm.29.Threshold	10689	29c1
Alarm.29.Type	10688	29c0
Alarm.30.Ack	10714	29da
Alarm.30.Block	10710	29d6
Alarm.30.Delay	10712	29d8
Alarm.30.Hysteresis	10706	29d2
Alarm.30.Inhibit	10711	29d7
Alarm.30.Latch	10708	29d4
Alarm.30.Out	10713	29d9
Alarm.30.Reference	10707	29d3
Alarm.30.Threshold	10705	29d1
Alarm.30.Type	10704	29d0
Alarm.31.Ack	10730	29ea
Alarm.31.Block	10726	29e6
Alarm.31.Delay	10728	29e8
Alarm.31.Hysteresis	10722	29e2
Alarm.31.Inhibit	10727	29e7
Alarm.31.Latch	10724	29e4
Alarm.31.Out	10729	29e9
Alarm.31.Reference	10723	29e3
Alarm.31.Threshold	10721	29e1
Alarm.31.Type	10720	29e0
Alarm.32.Ack	10746	29fa
Alarm.32.Block	10742	29f6
Alarm.32.Delay	10744	29f8
Alarm.32.Hysteresis	10738	29f2
Alarm.32.Inhibit	10743	29f7
Alarm.32.Latch	10740	29f4
Alarm.32.Out	10745	29f9
Alarm.32.Reference	10739	29f3
Alarm.32.Threshold	10737	29f1
Alarm.32.Type	10736	29f0
AlmSummary.General.AnAlarmStatus1	10176	27c0
AlmSummary.General.AnAlarmStatus2	10177	27c1
AlmSummary.General.AnAlarmStatus3	10178	27c2
AlmSummary.General.AnAlarmStatus4	10179	27c3
AlmSummary.General.AnyAlarm	10213	27e5
AlmSummary.General.CTAlarmStatus1	4192	1060
AlmSummary.General.CTAlarmStatus2	4193	1061
AlmSummary.General.CTAlarmStatus3	4194	1062
AlmSummary.General.CTAlarmStatus4	4195	1063
AlmSummary.General.DigAlarmStatus1	10188	27cc
AlmSummary.General.DigAlarmStatus2	10189	27cd
AlmSummary.General.DigAlarmStatus3	10190	27ce
AlmSummary.General.DigAlarmStatus4	10191	27cf
AlmSummary.General.GlobalAck	10214	27e6
AlmSummary.General.NewAlarm	10212	27e4
AlmSummary.General.NewCTAlarm	4196	1064
AlmSummary.General.RstNewAlarm	10215	27e7
AlmSummary.General.RstNewCTAlarm	4197	1065
AlmSummary.General.SBrkAlarmStatus1	10200	27d8
AlmSummary.General.SBrkAlarmStatus2	10201	27d9
AlmSummary.General.SBrkAlarmStatus3	10202	27da
AlmSummary.General.SBrkAlarmStatus4	10203	27db
Comms.FC.Ident	12963	32a3
DigAlarm.1.Ack	11274	2c0a
DigAlarm.1.Block	11270	2c06
DigAlarm.1.Delay	11272	2c08

Description	ModBus	HEX
DigAlarm.1.Inhibit	11271	2c07
DigAlarm.1.Latch	11268	2c04
DigAlarm.1.Out	11273	2c09
DigAlarm.1.Type	11264	2c00
DigAlarm.2.Ack	11290	2c1a
DigAlarm.2.Block	11286	2c16
DigAlarm.2.Delay	11288	2c18
DigAlarm.2.Inhibit	11287	2c17
DigAlarm.2.Latch	11284	2c14
DigAlarm.2.Out	11289	2c19
DigAlarm.2.Type	11280	2c10
DigAlarm.3.Ack	11306	2c2a
DigAlarm.3.Block	11302	2c26
DigAlarm.3.Delay	11304	2c28
DigAlarm.3.Inhibit	11303	2c27
DigAlarm.3.Latch	11300	2c24
DigAlarm.3.Out	11305	2c29
DigAlarm.3.Type	11296	2c20
DigAlarm.4.Ack	11322	2c3a
DigAlarm.4.Block	11318	2c36
DigAlarm.4.Delay	11320	2c38
DigAlarm.4.Inhibit	11319	2c37
DigAlarm.4.Latch	11316	2c34
DigAlarm.4.Out	11321	2c39
DigAlarm.4.Type	11312	2c30
DigAlarm.5.Ack	11338	2c4a
DigAlarm.5.Block	11334	2c46
DigAlarm.5.Delay	11336	2c48
DigAlarm.5.Inhibit	11335	2c47
DigAlarm.5.Latch	11332	2c44
DigAlarm.5.Out	11337	2c49
DigAlarm.5.Type	11328	2c40
DigAlarm.6.Ack	11354	2c5a
DigAlarm.6.Block	11350	2c56
DigAlarm.6.Delay	11352	2c58
DigAlarm.6.Inhibit	11351	2c57
DigAlarm.6.Latch	11348	2c54
DigAlarm.6.Out	11353	2c59
DigAlarm.6.Type	11344	2c50
DigAlarm.7.Ack	11370	2c6a
DigAlarm.7.Block	11366	2c66
DigAlarm.7.Delay	11368	2c68
DigAlarm.7.Inhibit	11367	2c67
DigAlarm.7.Latch	11364	2c64
DigAlarm.7.Out	11369	2c69
DigAlarm.7.Type	11360	2c60
DigAlarm.8.Ack	11386	2c7a
DigAlarm.8.Block	11382	2c76
DigAlarm.8.Delay	11384	2c78
DigAlarm.8.Inhibit	11383	2c77
DigAlarm.8.Latch	11380	2c74
DigAlarm.8.Out	11385	2c79
DigAlarm.8.Type	11376	2c70
DigAlarm.9.Ack	11402	2c8a
DigAlarm.9.Block	11398	2c86
DigAlarm.9.Delay	11400	2c88
DigAlarm.9.Inhibit	11399	2c87
DigAlarm.9.Latch	11396	2c84

Description	ModBus	HEX
DigAlarm.9.Out	11401	2c89
DigAlarm.9.Type	11392	2c80
DigAlarm.10.Ack	11418	2c9a
DigAlarm.10.Block	11414	2c96
DigAlarm.10.Delay	11416	2c98
DigAlarm.10.Inhibit	11415	2c97
DigAlarm.10.Latch	11412	2c94
DigAlarm.10.Out	11417	2c99
DigAlarm.10.Type	11408	2c90
DigAlarm.11.Ack	11434	2caa
DigAlarm.11.Block	11430	2ca6
DigAlarm.11.Delay	11432	2ca8
DigAlarm.11.Inhibit	11431	2ca7
DigAlarm.11.Latch	11428	2ca4
DigAlarm.11.Out	11433	2ca9
DigAlarm.11.Type	11424	2ca0
DigAlarm.12.Ack	11450	2cba
DigAlarm.12.Block	11446	2cb6
DigAlarm.12.Delay	11448	2cb8
DigAlarm.12.Inhibit	11447	2cb7
DigAlarm.12.Latch	11444	2cb4
DigAlarm.12.Out	11449	2cb9
DigAlarm.12.Type	11440	2cb0
DigAlarm.13.Ack	11466	2cca
DigAlarm.13.Block	11462	2cc6
DigAlarm.13.Delay	11464	2cc8
DigAlarm.13.Inhibit	11463	2cc7
DigAlarm.13.Latch	11460	2cc4
DigAlarm.13.Out	11465	2cc9
DigAlarm.13.Type	11456	2cc0
DigAlarm.14.Ack	11482	2cda
DigAlarm.14.Block	11478	2cd6
DigAlarm.14.Delay	11480	2cd8
DigAlarm.14.Inhibit	11479	2cd7
DigAlarm.14.Latch	11476	2cd4
DigAlarm.14.Out	11481	2cd9
DigAlarm.14.Type	11472	2cd0
DigAlarm.15.Ack	11498	2cea
DigAlarm.15.Block	11494	2ce6
DigAlarm.15.Delay	11496	2ce8
DigAlarm.15.Inhibit	11495	2ce7
DigAlarm.15.Latch	11492	2ce4
DigAlarm.15.Out	11497	2ce9
DigAlarm.15.Type	11488	2ce0
DigAlarm.16.Ack	11514	2cfa
DigAlarm.16.Block	11510	2cf6
DigAlarm.16.Delay	11512	2cf8
DigAlarm.16.Inhibit	11511	2cf7
DigAlarm.16.Latch	11508	2cf4
DigAlarm.16.Out	11513	2cf9
DigAlarm.16.Type	11504	2cf0
DigAlarm.17.Ack	11530	2d0a
DigAlarm.17.Block	11526	2d06
DigAlarm.17.Delay	11528	2d08
DigAlarm.17.Inhibit	11527	2d07
DigAlarm.17.Latch	11524	2d04
DigAlarm.17.Out	11529	2d09
DigAlarm.17.Type	11520	2d00

Description	ModBus	HEX
DigAlarm.18.Ack	11546	2d1a
DigAlarm.18.Block	11542	2d16
DigAlarm.18.Delay	11544	2d18
DigAlarm.18.Inhibit	11543	2d17
DigAlarm.18.Latch	11540	2d14
DigAlarm.18.Out	11545	2d19
DigAlarm.18.Type	11536	2d10
DigAlarm.19.Ack	11562	2d2a
DigAlarm.19.Block	11558	2d26
DigAlarm.19.Delay	11560	2d28
DigAlarm.19.Inhibit	11559	2d27
DigAlarm.19.Latch	11556	2d24
DigAlarm.19.Out	11561	2d29
DigAlarm.19.Type	11552	2d20
DigAlarm.20.Ack	11578	2d3a
DigAlarm.20.Block	11574	2d36
DigAlarm.20.Delay	11576	2d38
DigAlarm.20.Inhibit	11575	2d37
DigAlarm.20.Latch	11572	2d34
DigAlarm.20.Out	11577	2d39
DigAlarm.20.Type	11568	2d30
DigAlarm.21.Ack	11594	2d4a
DigAlarm.21.Block	11590	2d46
DigAlarm.21.Delay	11592	2d48
DigAlarm.21.Inhibit	11591	2d47
DigAlarm.21.Latch	11588	2d44
DigAlarm.21.Out	11593	2d49
DigAlarm.21.Type	11584	2d40
DigAlarm.22.Ack	11610	2d5a
DigAlarm.22.Block	11606	2d56
DigAlarm.22.Delay	11608	2d58
DigAlarm.22.Inhibit	11607	2d57
DigAlarm.22.Latch	11604	2d54
DigAlarm.22.Out	11609	2d59
DigAlarm.22.Type	11600	2d50
DigAlarm.23.Ack	11626	2d6a
DigAlarm.23.Block	11622	2d66
DigAlarm.23.Delay	11624	2d68
DigAlarm.23.Inhibit	11623	2d67
DigAlarm.23.Latch	11620	2d64
DigAlarm.23.Out	11625	2d69
DigAlarm.23.Type	11616	2d60
DigAlarm.24.Ack	11642	2d7a
DigAlarm.24.Block	11638	2d76
DigAlarm.24.Delay	11640	2d78
DigAlarm.24.Inhibit	11639	2d77
DigAlarm.24.Latch	11636	2d74
DigAlarm.24.Out	11641	2d79
DigAlarm.24.Type	11632	2d70
DigAlarm.25.Ack	11658	2d8a
DigAlarm.25.Block	11654	2d86
DigAlarm.25.Delay	11656	2d88
DigAlarm.25.Inhibit	11655	2d87
DigAlarm.25.Latch	11652	2d84
DigAlarm.25.Out	11657	2d89
DigAlarm.25.Type	11648	2d80
DigAlarm.26.Ack	11674	2d9a
DigAlarm.26.Block	11670	2d96

Description	ModBus	HEX
DigAlarm.26.Delay	11672	2d98
DigAlarm.26.Inhibit	11671	2d97
DigAlarm.26.Latch	11668	2d94
DigAlarm.26.Out	11673	2d99
DigAlarm.26.Type	11664	2d90
DigAlarm.27.Ack	11690	2daa
DigAlarm.27.Block	11686	2da6
DigAlarm.27.Delay	11688	2da8
DigAlarm.27.Inhibit	11687	2da7
DigAlarm.27.Latch	11684	2da4
DigAlarm.27.Out	11689	2da9
DigAlarm.27.Type	11680	2da0
DigAlarm.28.Ack	11706	2dba
DigAlarm.28.Block	11702	2db6
DigAlarm.28.Delay	11704	2db8
DigAlarm.28.Inhibit	11703	2db7
DigAlarm.28.Latch	11700	2db4
DigAlarm.28.Out	11705	2db9
DigAlarm.28.Type	11696	2db0
DigAlarm.29.Ack	11722	2dca
DigAlarm.29.Block	11718	2dc6
DigAlarm.29.Delay	11720	2dc8
DigAlarm.29.Inhibit	11719	2dc7
DigAlarm.29.Latch	11716	2dc4
DigAlarm.29.Out	11721	2dc9
DigAlarm.29.Type	11712	2dc0
DigAlarm.30.Ack	11738	2dda
DigAlarm.30.Block	11734	2dd6
DigAlarm.30.Delay	11736	2dd8
DigAlarm.30.Inhibit	11735	2dd7
DigAlarm.30.Latch	11732	2dd4
DigAlarm.30.Out	11737	2dd9
DigAlarm.30.Type	11728	2dd0
DigAlarm.31.Ack	11754	2dea
DigAlarm.31.Block	11750	2de6
DigAlarm.31.Delay	11752	2de8
DigAlarm.31.Inhibit	11751	2de7
DigAlarm.31.Latch	11748	2de4
DigAlarm.31.Out	11753	2de9
DigAlarm.31.Type	11744	2de0
DigAlarm.32.Ack	11770	2dfa
DigAlarm.32.Block	11766	2df6
DigAlarm.32.Delay	11768	2df8
DigAlarm.32.Inhibit	11767	2df7
DigAlarm.32.Latch	11764	2df4
DigAlarm.32.Out	11769	2df9
DigAlarm.32.Type	11760	2df0
Humidity.DewPoint	13317	3405
Humidity.DryTemp	13318	3406
Humidity.Pressure	13313	3401
Humidity.PsychroConst	13315	3403
Humidity.RelHumid	13316	3404
Humidity.Resolution	13320	3408
Humidity.SBrk	13314	3402
Humidity.WetOffset	13312	3400
Humidity.WetTemp	13319	3407
Instrument.Diagnostics.CntrlOverrun	4737	1281
Instrument.Diagnostics.ErrCount	4736	1280

Description	ModBus	HEX
Instrument.Diagnostics.PSUident	13027	32e3
Instrument.InstInfo.CompanyID	121	0079
Instrument.InstInfo.InstType	122	007a
Instrument.InstInfo.Version	107	006b
Instrument.Options.Units	4738	1282
IO.CurrentMonitor.Config.CalibrateCT1	4170	104a
IO.CurrentMonitor.Config.CalibrateCT2	4171	104b
IO.CurrentMonitor.Config.CalibrateCT3	4172	104c
IO.CurrentMonitor.Config.Commission	4096	1000
IO.CurrentMonitor.Config.CommissionStatus	4097	1001
IO.CurrentMonitor.Config.CT1Range	4103	1007
IO.CurrentMonitor.Config.CT1Resolution	4198	1066
IO.CurrentMonitor.Config.CT2Range	4104	1008
IO.CurrentMonitor.Config.CT2Resolution	4199	1067
IO.CurrentMonitor.Config.CT3Range	4105	1009
IO.CurrentMonitor.Config.CT3Resolution	4200	1068
IO.CurrentMonitor.Config.Inhibit	4099	1003
IO.CurrentMonitor.Config.Interval	4098	1002
IO.CurrentMonitor.Config.Load1CTInput	4107	100b
IO.CurrentMonitor.Config.Load1DrivenBy	4106	100a
IO.CurrentMonitor.Config.Load1OCFthreshold	4109	100d
IO.CurrentMonitor.Config.Load1PLFthreshold	4108	100c
IO.CurrentMonitor.Config.Load1Resolution	4201	1069
IO.CurrentMonitor.Config.Load2CTInput	4111	100f
IO.CurrentMonitor.Config.Load2DrivenBy	4110	100e
IO.CurrentMonitor.Config.Load2OCFthreshold	4113	1011
IO.CurrentMonitor.Config.Load2PLFthreshold	4112	1010
IO.CurrentMonitor.Config.Load2Resolution	4202	106a
IO.CurrentMonitor.Config.Load3CTInput	4115	1013
IO.CurrentMonitor.Config.Load3DrivenBy	4114	1012
IO.CurrentMonitor.Config.Load3OCFthreshold	4117	1015
IO.CurrentMonitor.Config.Load3PLFthreshold	4116	1014
IO.CurrentMonitor.Config.Load3Resolution	4203	106b
IO.CurrentMonitor.Config.Load4CTInput	4119	1017
IO.CurrentMonitor.Config.Load4DrivenBy	4118	1016
IO.CurrentMonitor.Config.Load4OCFthreshold	4121	1019
IO.CurrentMonitor.Config.Load4PLFthreshold	4120	1018
IO.CurrentMonitor.Config.Load4Resolution	4204	106c
IO.CurrentMonitor.Config.Load5CTInput	4123	101b
IO.CurrentMonitor.Config.Load5DrivenBy	4122	101a
IO.CurrentMonitor.Config.Load5OCFthreshold	4125	101d
IO.CurrentMonitor.Config.Load5PLFthreshold	4124	101c
IO.CurrentMonitor.Config.Load5Resolution	4205	106d
IO.CurrentMonitor.Config.Load6CTInput	4127	101f
IO.CurrentMonitor.Config.Load6DrivenBy	4126	101e
IO.CurrentMonitor.Config.Load6OCFthreshold	4129	1021
IO.CurrentMonitor.Config.Load6PLFthreshold	4128	1020
IO.CurrentMonitor.Config.Load6Resolution	4206	106e
IO.CurrentMonitor.Config.Load7CTInput	4131	1023
IO.CurrentMonitor.Config.Load7DrivenBy	4130	1022
IO.CurrentMonitor.Config.Load7OCFthreshold	4133	1025
IO.CurrentMonitor.Config.Load7PLFthreshold	4132	1024
IO.CurrentMonitor.Config.Load7Resolution	4207	106f
IO.CurrentMonitor.Config.Load8CTInput	4135	1027
IO.CurrentMonitor.Config.Load8DrivenBy	4134	1026
IO.CurrentMonitor.Config.Load8OCFthreshold	4137	1029
IO.CurrentMonitor.Config.Load8PLFthreshold	4136	1028
IO.CurrentMonitor.Config.Load8Resolution	4208	1070
IO.CurrentMonitor.Config.Load9CTInput	4139	102b
IO.CurrentMonitor.Config.Load9DrivenBy	4138	102a
IO.CurrentMonitor.Config.Load9OCFthreshold	4141	102d
IO.CurrentMonitor.Config.Load9PLFthreshold	4140	102c
IO.CurrentMonitor.Config.Load9Resolution	4209	1071
IO.CurrentMonitor.Config.Load10CTInput	4143	102f
IO.CurrentMonitor.Config.Load10DrivenBy	4142	102e
IO.CurrentMonitor.Config.Load10OCFthreshold	4145	1031
IO.CurrentMonitor.Config.Load10PLFthreshold	4144	1030
IO.CurrentMonitor.Config.Load10Resolution	4210	1072
IO.CurrentMonitor.Config.Load11CTInput	4147	1033
IO.CurrentMonitor.Config.Load11DrivenBy	4146	1032
IO.CurrentMonitor.Config.Load11OCFthreshold	4149	1035
IO.CurrentMonitor.Config.Load11PLFthreshold	4148	1034
IO.CurrentMonitor.Config.Load11Resolution	4211	1073
IO.CurrentMonitor.Config.Load12CTInput	4151	1037
IO.CurrentMonitor.Config.Load12DrivenBy	4150	1036
IO.CurrentMonitor.Config.Load12OCFthreshold	4153	1039
IO.CurrentMonitor.Config.Load12PLFthreshold	4152	1038
IO.CurrentMonitor.Config.Load12Resolution	4212	1074
IO.CurrentMonitor.Config.Load13CTInput	4155	103b
IO.CurrentMonitor.Config.Load13DrivenBy	4154	103a
IO.CurrentMonitor.Config.Load13OCFthreshold	4157	103d
IO.CurrentMonitor.Config.Load13PLFthreshold	4156	103c
IO.CurrentMonitor.Config.Load13Resolution	4213	1075
IO.CurrentMonitor.Config.Load14CTInput	4159	103f
IO.CurrentMonitor.Config.Load14DrivenBy	4158	103e
IO.CurrentMonitor.Config.Load14OCFthreshold	4161	1041
IO.CurrentMonitor.Config.Load14PLFthreshold	4160	1040
IO.CurrentMonitor.Config.Load14Resolution	4214	1076
IO.CurrentMonitor.Config.Load15CTInput	4163	1043
IO.CurrentMonitor.Config.Load15DrivenBy	4162	1042
IO.CurrentMonitor.Config.Load15OCFthreshold	4165	1045
IO.CurrentMonitor.Config.Load15PLFthreshold	4164	1044
IO.CurrentMonitor.Config.Load15Resolution	4215	1077
IO.CurrentMonitor.Config.Load16CTInput	4167	1047
IO.CurrentMonitor.Config.Load16DrivenBy	4166	1046
IO.CurrentMonitor.Config.Load16OCFthreshold	4169	1049
IO.CurrentMonitor.Config.Load16PLFthreshold	4168	1048
IO.CurrentMonitor.Config.Load16Resolution	4216	1078
IO.CurrentMonitor.Config.MaxLeakPh1	4100	1004
IO.CurrentMonitor.Config.MaxLeakPh2	4101	1005
IO.CurrentMonitor.Config.MaxLeakPh3	4102	1006
IO.CurrentMonitor.Status.Load1Current	4173	104d
IO.CurrentMonitor.Status.Load2Current	4174	104e
IO.CurrentMonitor.Status.Load3Current	4175	104f
IO.CurrentMonitor.Status.Load4Current	4176	1050
IO.CurrentMonitor.Status.Load5Current	4177	1051
IO.CurrentMonitor.Status.Load6Current	4178	1052
IO.CurrentMonitor.Status.Load7Current	4179	1053
IO.CurrentMonitor.Status.Load8Current	4180	1054
IO.CurrentMonitor.Status.Load9Current	4181	1055
IO.CurrentMonitor.Status.Load10Current	4182	1056
IO.CurrentMonitor.Status.Load11Current	4183	1057
IO.CurrentMonitor.Status.Load12Current	4184	1058
IO.CurrentMonitor.Status.Load13Current	4185	1059
IO.CurrentMonitor.Status.Load14Current	4186	105a
IO.CurrentMonitor.Status.Load15Current	4187	105b

Description	ModBus	HEX
IO.CurrentMonitor.Status.Load16Current	4188	105c
IO.CurrentMonitor.Status.Ph1AllOff	4189	105d
IO.CurrentMonitor.Status.Ph2AllOff	4190	105e
IO.CurrentMonitor.Status.Ph3AllOff	4191	105f
IO.FixedIO.A.PV	4226	1082
IO.FixedIO.B.PV	4227	1083
IO.FixedIO.D1.PV	4224	1080
IO.FixedIO.D2.PV	4225	1081
IO.Mod.1.AlarmAck	4260	10a4
IO.Mod.1.MinOnTime	4292	10c4
IO.Mod.1.PV	4228	1084
IO.Mod.2.AlarmAck	4261	10a5
IO.Mod.2.MinOnTime	4293	10c5
IO.Mod.2.PV	4229	1085
IO.Mod.3.AlarmAck	4262	10a6
IO.Mod.3.MinOnTime	4294	10c6
IO.Mod.3.PV	4230	1086
IO.Mod.4.AlarmAck	4263	10a7
IO.Mod.4.MinOnTime	4295	10c7
IO.Mod.4.PV	4231	1087
IO.Mod.5.AlarmAck	4264	10a8
IO.Mod.5.MinOnTime	4296	10c8
IO.Mod.5.PV	4232	1088
IO.Mod.6.AlarmAck	4265	10a9
IO.Mod.6.MinOnTime	4297	10c9
IO.Mod.6.PV	4233	1089
IO.Mod.7.AlarmAck	4266	10aa
IO.Mod.7.MinOnTime	4298	10ca
IO.Mod.7.PV	4234	108a
IO.Mod.8.AlarmAck	4267	10ab
IO.Mod.8.MinOnTime	4299	10cb
IO.Mod.8.PV	4235	108b
IO.Mod.9.AlarmAck	4268	10ac
IO.Mod.9.MinOnTime	4300	10cc
IO.Mod.9.PV	4236	108c
IO.Mod.10.AlarmAck	4269	10ad
IO.Mod.10.MinOnTime	4301	10cd
IO.Mod.10.PV	4237	108d
IO.Mod.11.AlarmAck	4270	10ae
IO.Mod.11.MinOnTime	4302	10ce
IO.Mod.11.PV	4238	108e
IO.Mod.12.AlarmAck	4271	10af
IO.Mod.12.MinOnTime	4303	10cf
IO.Mod.12.PV	4239	108f
IO.Mod.13.AlarmAck	4272	10b0
IO.Mod.13.MinOnTime	4304	10d0
IO.Mod.13.PV	4240	1090
IO.Mod.14.AlarmAck	4273	10b1
IO.Mod.14.MinOnTime	4305	10d1
IO.Mod.14.PV	4241	1091
IO.Mod.15.AlarmAck	4274	10b2
IO.Mod.15.MinOnTime	4306	10d2
IO.Mod.15.PV	4242	1092
IO.Mod.16.AlarmAck	4275	10b3
IO.Mod.16.MinOnTime	4307	10d3
IO.Mod.16.PV	4243	1093
IO.Mod.17.AlarmAck	4276	10b4
IO.Mod.17.MinOnTime	4308	10d4

Description	ModBus	HEX
IO.Mod.17.PV	4244	1094
IO.Mod.18.AlarmAck	4277	10b5
IO.Mod.18.MinOnTime	4309	10d5
IO.Mod.18.PV	4245	1095
IO.Mod.19.AlarmAck	4278	10b6
IO.Mod.19.MinOnTime	4310	10d6
IO.Mod.19.PV	4246	1096
IO.Mod.20.AlarmAck	4279	10b7
IO.Mod.20.MinOnTime	4311	10d7
IO.Mod.20.PV	4247	1097
IO.Mod.21.AlarmAck	4280	10b8
IO.Mod.21.MinOnTime	4312	10d8
IO.Mod.21.PV	4248	1098
IO.Mod.22.AlarmAck	4281	10b9
IO.Mod.22.MinOnTime	4313	10d9
IO.Mod.22.PV	4249	1099
IO.Mod.23.AlarmAck	4282	10ba
IO.Mod.23.MinOnTime	4314	10da
IO.Mod.23.PV	4250	109a
IO.Mod.24.AlarmAck	4283	10bb
IO.Mod.24.MinOnTime	4315	10db
IO.Mod.24.PV	4251	109b
IO.Mod.25.AlarmAck	4284	10bc
IO.Mod.25.MinOnTime	4316	10dc
IO.Mod.25.PV	4252	109c
IO.Mod.26.AlarmAck	4285	10bd
IO.Mod.26.MinOnTime	4317	10dd
IO.Mod.26.PV	4253	109d
IO.Mod.27.AlarmAck	4286	10be
IO.Mod.27.MinOnTime	4318	10de
IO.Mod.27.PV	4254	109e
IO.Mod.28.AlarmAck	4287	10bf
IO.Mod.28.MinOnTime	4319	10df
IO.Mod.28.PV	4255	109f
IO.Mod.29.AlarmAck	4288	10c0
IO.Mod.29.MinOnTime	4320	10e0
IO.Mod.29.PV	4256	10a0
IO.Mod.30.AlarmAck	4289	10c1
IO.Mod.30.MinOnTime	4321	10e1
IO.Mod.30.PV	4257	10a1
IO.Mod.31.AlarmAck	4290	10c2
IO.Mod.31.MinOnTime	4322	10e2
IO.Mod.31.PV	4258	10a2
IO.Mod.32.AlarmAck	4291	10c3
IO.Mod.32.MinOnTime	4323	10e3
IO.Mod.32.PV	4259	10a3
IO.ModIDs.Module1	12707	31a3
IO.ModIDs.Module2	12771	31e3
IO.ModIDs.Module3	12835	3223
IO.ModIDs.Module4	12899	3263
IPMonitor.1.Max	4915	1333
IPMonitor.1.Min	4916	1334
IPMonitor.1.Reset	4919	1337
IPMonitor.1.Threshold	4917	1335
IPMonitor.1.TimeAbove	4918	1336
IPMonitor.2.Max	4920	1338
IPMonitor.2.Min	4921	1339
IPMonitor.2.Reset	4924	133c

Description	ModBus	HEX
IPMonitor.2.Threshold	4922	133a
IPMonitor.2.TimeAbove	4923	133b
Lgc2.1.In1	4822	12d6
Lgc2.1.In2	4823	12d7
Lgc2.1.Out	4824	12d8
Lgc2.2.In1	4825	12d9
Lgc2.2.In2	4826	12da
Lgc2.2.Out	4827	12db
Lgc2.3.In1	4828	12dc
Lgc2.3.In2	4829	12dd
Lgc2.3.Out	4830	12de
Lgc2.4.In1	4831	12df
Lgc2.4.In2	4832	12e0
Lgc2.4.Out	4833	12e1
Lgc2.5.In1	4834	12e2
Lgc2.5.In2	4835	12e3
Lgc2.5.Out	4836	12e4
Lgc2.6.In1	4837	12e5
Lgc2.6.In2	4838	12e6
Lgc2.6.Out	4839	12e7
Lgc2.7.In1	4840	12e8
Lgc2.7.In2	4841	12e9
Lgc2.7.Out	4842	12ea
Lgc2.8.In1	4843	12eb
Lgc2.8.In2	4844	12ec
Lgc2.8.Out	4845	12ed
Lgc2.9.In1	4846	12ee
Lgc2.9.In2	4847	12ef
Lgc2.9.Out	4848	12f0
Lgc2.10.In1	4849	12f1
Lgc2.10.In2	4850	12f2
Lgc2.10.Out	4851	12f3
Lgc2.11.In1	4852	12f4
Lgc2.11.In2	4853	12f5
Lgc2.11.Out	4854	12f6
Lgc2.12.In1	4855	12f7
Lgc2.12.In2	4856	12f8
Lgc2.12.Out	4857	12f9
Lgc2.13.In1	4858	12fa
Lgc2.13.In2	4859	12fb
Lgc2.13.Out	4860	12fc
Lgc2.14.In1	4861	12fd
Lgc2.14.In2	4862	12fe
Lgc2.14.Out	4863	12ff
Lgc2.15.In1	4864	1300
Lgc2.15.In2	4865	1301
Lgc2.15.Out	4866	1302
Lgc2.16.In1	4867	1303
Lgc2.16.In2	4868	1304
Lgc2.16.Out	4869	1305
Lgc2.17.In1	4870	1306
Lgc2.17.In2	4871	1307
Lgc2.17.Out	4872	1308
Lgc2.18.In1	4873	1309
Lgc2.18.In2	4874	130a
Lgc2.18.Out	4875	130b
Lgc2.19.In1	4876	130c
Lgc2.19.In2	4877	130d

Description	ModBus	HEX
Lgc2.19.Out	4878	130e
Lgc2.20.In1	4879	130f
Lgc2.20.In2	4880	1310
Lgc2.20.Out	4881	1311
Lgc2.21.In1	4882	1312
Lgc2.21.In2	4883	1313
Lgc2.21.Out	4884	1314
Lgc2.22.In1	4885	1315
Lgc2.22.In2	4886	1316
Lgc2.22.Out	4887	1317
Lgc2.23.In1	4888	1318
Lgc2.23.In2	4889	1319
Lgc2.23.Out	4890	131a
Lgc2.24.In1	4891	131b
Lgc2.24.In2	4892	131c
Lgc2.24.Out	4893	131d
Lgc8.1.In1	4894	131e
Lgc8.1.In2	4895	131f
Lgc8.1.In3	4896	1320
Lgc8.1.In4	4897	1321
Lgc8.1.In5	4898	1322
Lgc8.1.In6	4899	1323
Lgc8.1.In7	4900	1324
Lgc8.1.In8	4901	1325
Lgc8.1.Out	4902	1326
Lgc8.2.In1	4903	1327
Lgc8.2.In2	4904	1328
Lgc8.2.In3	4905	1329
Lgc8.2.In4	4906	132a
Lgc8.2.In5	4907	132b
Lgc8.2.In6	4908	132c
Lgc8.2.In7	4909	132d
Lgc8.2.In8	4910	132e
Lgc8.2.Out	4911	132f
Lin16.In	4960	1360
Lin16.In1	4929	1341
Lin16.In2	4930	1342
Lin16.In3	4931	1343
Lin16.In4	4932	1344
Lin16.In5	4933	1345
Lin16.In6	4934	1346
Lin16.In7	4935	1347
Lin16.In8	4936	1348
Lin16.In9	4937	1349
Lin16.In10	4938	134a
Lin16.In11	4939	134b
Lin16.In12	4940	134c
Lin16.In13	4941	134d
Lin16.In14	4942	134e
Lin16.InHighLimit	4943	134f
Lin16.InLowLimit	4928	1340
Lin16.Out	4961	1361
Lin16.Out1	4945	1351
Lin16.Out2	4946	1352
Lin16.Out3	4947	1353
Lin16.Out4	4948	1354
Lin16.Out5	4949	1355
Lin16.Out6	4950	1356

Description	ModBus	HEX
Lin16.Out7	4951	1357
Lin16.Out8	4952	1358
Lin16.Out9	4953	1359
Lin16.Out10	4954	135a
Lin16.Out11	4955	135b
Lin16.Out12	4956	135c
Lin16.Out13	4957	135d
Lin16.Out14	4958	135e
Lin16.OutHighLimit	4959	135f
Lin16.OutLowLimit	4944	1350
Loop.1.Diag.DerivativeOutContrib	119	0077
Loop.1.Diag.Error	113	0071
Loop.1.Diag.IntegralOutContrib	118	0076
Loop.1.Diag.LoopBreakAlarm	116	0074
Loop.1.Diag.LoopMode	114	0072
Loop.1.Diag.PropOutContrib	117	0075
Loop.1.Diag.SBrk	120	0078
Loop.1.Diag.SchedCBH	32	0020
Loop.1.Diag.SchedCBL	33	0021
Loop.1.Diag.SchedLPBrk	35	0023
Loop.1.Diag.SchedMR	34	0022
Loop.1.Diag.SchedOPHi	37	0025
Loop.1.Diag.SchedOPLo	38	0026
Loop.1.Diag.SchedPB	29	001d
Loop.1.Diag.SchedR2G	36	0024
Loop.1.Diag.SchedTd	31	001f
Loop.1.Diag.SchedTi	30	001e
Loop.1.Diag.TargetOutVal	115	0073
Loop.1.Main.ActiveOut	4	0004
Loop.1.Main.AutoMan	10	000a
Loop.1.Main.Inhibit	20	0014
Loop.1.Main.PV	1	0001
Loop.1.Main.TargetSP	2	0002
Loop.1.Main.WorkingSP	5	0005
Loop.1.OP.Ch1OnOffHysteresis	84	0054
Loop.1.OP.Ch1Out	82	0052
Loop.1.OP.Ch2Deadband	16	0010
Loop.1.OP.Ch2OnOffHysteresis	85	0055
Loop.1.OP.Ch2Out	83	0053
Loop.1.OP.CoolType	93	005d
Loop.1.OP.EnablePowerFeedforward	91	005b
Loop.1.OP.FeedForwardGain	95	005f
Loop.1.OP.FeedForwardOffset	96	0060
Loop.1.OP.FeedForwardTrimLimit	97	0061
Loop.1.OP.FeedForwardType	94	005e
Loop.1.OP.FeedForwardVal	98	0062
Loop.1.OP.FF_Rem	103	0067
Loop.1.OP.ManualMode	90	005a
Loop.1.OP.ManualOutVal	3	0003
Loop.1.OP.MeasuredPower	92	005c
Loop.1.OP.OutputHighLimit	80	0050
Loop.1.OP.OutputLowLimit	81	0051
Loop.1.OP.Rate	86	0056
Loop.1.OP.RateDisable	87	0057
Loop.1.OP.RemOPH	102	0066
Loop.1.OP.RemOPL	101	0065
Loop.1.OP.SafeOutVal	89	0059
Loop.1.OP.SensorBreakMode	88	0058

Description	ModBus	HEX
Loop.1.OP.TrackEnable	100	0064
Loop.1.OP.TrackOutVal	99	0063
Loop.1.PID.ActiveSet	28	001c
Loop.1.PID.Boundary1-2	26	001a
Loop.1.PID.Boundary2-3	27	001b
Loop.1.PID.CutbackHigh	18	0012
Loop.1.PID.CutbackHigh2	46	002e
Loop.1.PID.CutbackHigh3	56	0038
Loop.1.PID.CutbackLow	17	0011
Loop.1.PID.CutbackLow2	47	002f
Loop.1.PID.CutbackLow3	57	0039
Loop.1.PID.DerivativeTime	9	0009
Loop.1.PID.DerivativeTime2	45	002d
Loop.1.PID.DerivativeTime3	55	0037
Loop.1.PID.IntegralTime	8	0008
Loop.1.PID.IntegralTime2	44	002c
Loop.1.PID.IntegralTime3	54	0036
Loop.1.PID.LoopBreakTime	40	0028
Loop.1.PID.LoopBreakTime2	49	0031
Loop.1.PID.LoopBreakTime3	59	003b
Loop.1.PID.ManualReset	39	0027
Loop.1.PID.ManualReset2	48	0030
Loop.1.PID.ManualReset3	58	003a
Loop.1.PID.NumSets	64	0040
Loop.1.PID.OutputHi	41	0029
Loop.1.PID.OutputHi2	51	0033
Loop.1.PID.OutputHi3	61	003d
Loop.1.PID.OutputLo	42	002a
Loop.1.PID.OutputLo2	52	0034
Loop.1.PID.OutputLo3	62	003e
Loop.1.PID.ProportionalBand	6	0006
Loop.1.PID.ProportionalBand2	43	002b
Loop.1.PID.ProportionalBand3	53	0035
Loop.1.PID.RelCh2Gain	19	0013
Loop.1.PID.RelCh2Gain2	50	0032
Loop.1.PID.RelCh2Gain3	60	003c
Loop.1.PID.SchedulerRemoteInput	65	0041
Loop.1.PID.SchedulerType	63	003f
Loop.1.Setup.CH1ControlType	22	0016
Loop.1.Setup.CH2ControlType	23	0017
Loop.1.Setup.ControlAction	7	0007
Loop.1.Setup.DerivativeType	25	0019
Loop.1.Setup.LoopType	21	0015
Loop.1.Setup.PBUnits	24	0018
Loop.1.SP.AltSP	68	0044
Loop.1.SP.AltSPSelect	69	0045
Loop.1.SP.ManualTrack	75	004b
Loop.1.SP.RangeHigh	12	000c
Loop.1.SP.RangeLow	11	000b
Loop.1.SP.Rate	70	0046
Loop.1.SP.RateDisable	71	0047
Loop.1.SP.RateDone	79	004f
Loop.1.SP.SP1	13	000d
Loop.1.SP.SP2	14	000e
Loop.1.SP.SPHighLimit	66	0042
Loop.1.SP.SPLowLimit	67	0043
Loop.1.SP.SPSelect	15	000f
Loop.1.SP.SPTrack	76	004c

Description	ModBus	HEX
Loop.1.SP.SPTrim	72	0048
Loop.1.SP.SPTrimHighLimit	73	0049
Loop.1.SP.SPTrimLowLimit	74	004a
Loop.1.SP.TrackPV	77	004d
Loop.1.SP.TrackSP	78	004e
Loop.1.Tune.AutotuneEnable	108	006c
Loop.1.Tune.OutputHighLimit	105	0069
Loop.1.Tune.OutputLowLimit	106	006a
Loop.1.Tune.Stage	111	006f
Loop.1.Tune.StageTime	112	0070
Loop.1.Tune.State	110	006e
Loop.1.Tune.StepSize	109	006d
Loop.1.Tune.Type	104	0068
Loop.2.Diag.DerivativeOutContrib	375	0177
Loop.2.Diag.Error	369	0171
Loop.2.Diag.IntegralOutContrib	374	0176
Loop.2.Diag.LoopBreakAlarm	372	0174
Loop.2.Diag.LoopMode	370	0172
Loop.2.Diag.PropOutContrib	373	0175
Loop.2.Diag.SBrk	376	0178
Loop.2.Diag.SchedCBH	288	0120
Loop.2.Diag.SchedCBL	289	0121
Loop.2.Diag.SchedLPBrk	291	0123
Loop.2.Diag.SchedMR	290	0122
Loop.2.Diag.SchedOPHi	293	0125
Loop.2.Diag.SchedOPLo	294	0126
Loop.2.Diag.SchedPB	285	011d
Loop.2.Diag.SchedR2G	292	0124
Loop.2.Diag.SchedTd	287	011f
Loop.2.Diag.SchedTi	286	011e
Loop.2.Diag.TargetOutVal	371	0173
Loop.2.Main.ActiveOut	260	0104
Loop.2.Main.AutoMan	266	010a
Loop.2.Main.Inhibit	276	0114
Loop.2.Main.PV	257	0101
Loop.2.Main.TargetSP	258	0102
Loop.2.Main.WorkingSP	261	0105
Loop.2.OP.Ch1OnOffHysteresis	340	0154
Loop.2.OP.Ch1Out	338	0152
Loop.2.OP.Ch2Deadband	272	0110
Loop.2.OP.Ch2OnOffHysteresis	341	0155
Loop.2.OP.Ch2Out	339	0153
Loop.2.OP.CoolType	349	015d
Loop.2.OP.EnablePowerFeedforward	347	015b
Loop.2.OP.FeedForwardGain	351	015f
Loop.2.OP.FeedForwardOffset	352	0160
Loop.2.OP.FeedForwardTrimLimit	353	0161
Loop.2.OP.FeedForwardType	350	015e
Loop.2.OP.FeedForwardVal	354	0162
Loop.2.OP.FF_Rem	359	0167
Loop.2.OP.ManualMode	346	015a
Loop.2.OP.ManualOutVal	259	0103
Loop.2.OP.MeasuredPower	348	015c
Loop.2.OP.OutputHighLimit	336	0150
Loop.2.OP.OutputLowLimit	337	0151
Loop.2.OP.Rate	342	0156
Loop.2.OP.RateDisable	343	0157
Loop.2.OP.RemOPH	358	0166

Description	ModBus	HEX
Loop.2.OP.RemOPL	357	0165
Loop.2.OP.SafeOutVal	345	0159
Loop.2.OP.SensorBreakMode	344	0158
Loop.2.OP.TrackEnable	356	0164
Loop.2.OP.TrackOutVal	355	0163
Loop.2.PID.ActiveSet	284	011c
Loop.2.PID.Boundary1-2	282	011a
Loop.2.PID.Boundary2-3	283	011b
Loop.2.PID.CutbackHigh	274	0112
Loop.2.PID.CutbackHigh2	302	012e
Loop.2.PID.CutbackHigh3	312	0138
Loop.2.PID.CutbackLow	273	0111
Loop.2.PID.CutbackLow2	303	012f
Loop.2.PID.CutbackLow3	313	0139
Loop.2.PID.DerivativeTime	265	0109
Loop.2.PID.DerivativeTime2	301	012d
Loop.2.PID.DerivativeTime3	311	0137
Loop.2.PID.IntegralTime	264	0108
Loop.2.PID.IntegralTime2	300	012c
Loop.2.PID.IntegralTime3	310	0136
Loop.2.PID.LoopBreakTime	296	0128
Loop.2.PID.LoopBreakTime2	305	0131
Loop.2.PID.LoopBreakTime3	315	013b
Loop.2.PID.ManualReset	295	0127
Loop.2.PID.ManualReset2	304	0130
Loop.2.PID.ManualReset3	314	013a
Loop.2.PID.NumSets	320	0140
Loop.2.PID.OutputHi	297	0129
Loop.2.PID.OutputHi2	307	0133
Loop.2.PID.OutputHi3	317	013d
Loop.2.PID.OutputLo	298	012a
Loop.2.PID.OutputLo2	308	0134
Loop.2.PID.OutputLo3	318	013e
Loop.2.PID.ProportionalBand	262	0106
Loop.2.PID.ProportionalBand2	299	012b
Loop.2.PID.ProportionalBand3	309	0135
Loop.2.PID.RelCh2Gain	275	0113
Loop.2.PID.RelCh2Gain2	306	0132
Loop.2.PID.RelCh2Gain3	316	013c
Loop.2.PID.SchedulerRemoteInput	321	0141
Loop.2.PID.SchedulerType	319	013f
Loop.2.Setup.CH1ControlType	278	0116
Loop.2.Setup.CH2ControlType	279	0117
Loop.2.Setup.ControlAction	263	0107
Loop.2.Setup.DerivativeType	281	0119
Loop.2.Setup.LoopType	277	0115
Loop.2.Setup.PBUnits	280	0118
Loop.2.SP.AltSP	324	0144
Loop.2.SP.AltSPSelect	325	0145
Loop.2.SP.ManualTrack	331	014b
Loop.2.SP.RangeHigh	268	010c
Loop.2.SP.RangeLow	267	010b
Loop.2.SP.Rate	326	0146
Loop.2.SP.RateDisable	327	0147
Loop.2.SP.RateDone	335	014f
Loop.2.SP.SP1	269	010d
Loop.2.SP.SP2	270	010e
Loop.2.SP.SPHighLimit	322	0142

Description	ModBus	HEX
Loop.2.SP.SPLowLimit	323	0143
Loop.2.SP.SPSelect	271	010f
Loop.2.SP.SPTrack	332	014c
Loop.2.SP.SPTrim	328	0148
Loop.2.SP.SPTrimHighLimit	329	0149
Loop.2.SP.SPTrimLowLimit	330	014a
Loop.2.SP.TrackPV	333	014d
Loop.2.SP.TrackSP	334	014e
Loop.2.Tune.AutotuneEnable	364	016c
Loop.2.Tune.OutputHighLimit	361	0169
Loop.2.Tune.OutputLowLimit	362	016a
Loop.2.Tune.Stage	367	016f
Loop.2.Tune.StageTime	368	0170
Loop.2.Tune.State	366	016e
Loop.2.Tune.StepSize	365	016d
Loop.2.Tune.Type	360	0168
Loop.3.Diag.DerivativeOutContrib	631	0277
Loop.3.Diag.Error	625	0271
Loop.3.Diag.IntegralOutContrib	630	0276
Loop.3.Diag.LoopBreakAlarm	628	0274
Loop.3.Diag.LoopMode	626	0272
Loop.3.Diag.PropOutContrib	629	0275
Loop.3.Diag.SBrk	632	0278
Loop.3.Diag.SchedCBH	544	0220
Loop.3.Diag.SchedCBL	545	0221
Loop.3.Diag.SchedLPBrk	547	0223
Loop.3.Diag.SchedMR	546	0222
Loop.3.Diag.SchedOPHi	549	0225
Loop.3.Diag.SchedOPLo	550	0226
Loop.3.Diag.SchedPB	541	021d
Loop.3.Diag.SchedR2G	548	0224
Loop.3.Diag.SchedTd	543	021f
Loop.3.Diag.SchedTi	542	021e
Loop.3.Diag.TargetOutVal	627	0273
Loop.3.Main.ActiveOut	516	0204
Loop.3.Main.AutoMan	522	020a
Loop.3.Main.Inhibit	532	0214
Loop.3.Main.PV	513	0201
Loop.3.Main.TargetSP	514	0202
Loop.3.Main.WorkingSP	517	0205
Loop.3.OP.Ch1OnOffHysteresis	596	0254
Loop.3.OP.Ch1Out	594	0252
Loop.3.OP.Ch2Deadband	528	0210
Loop.3.OP.Ch2OnOffHysteresis	597	0255
Loop.3.OP.Ch2Out	595	0253
Loop.3.OP.CoolType	605	025d
Loop.3.OP.EnablePowerFeedforward	603	025b
Loop.3.OP.FeedForwardGain	607	025f
Loop.3.OP.FeedForwardOffset	608	0260
Loop.3.OP.FeedForwardTrimLimit	609	0261
Loop.3.OP.FeedForwardType	606	025e
Loop.3.OP.FeedForwardVal	610	0262
Loop.3.OP.FF_Rem	615	0267
Loop.3.OP.ManualMode	602	025a
Loop.3.OP.ManualOutVal	515	0203
Loop.3.OP.MeasuredPower	604	025c
Loop.3.OP.OutputHighLimit	592	0250
Loop.3.OP.OutputLowLimit	593	0251

Description	ModBus	HEX
Loop.3.OP.Rate	598	0256
Loop.3.OP.RateDisable	599	0257
Loop.3.OP.RemOPH	614	0266
Loop.3.OP.RemOPL	613	0265
Loop.3.OP.SafeOutVal	601	0259
Loop.3.OP.SensorBreakMode	600	0258
Loop.3.OP.TrackEnable	612	0264
Loop.3.OP.TrackOutVal	611	0263
Loop.3.PID.ActiveSet	540	021c
Loop.3.PID.Boundary1-2	538	021a
Loop.3.PID.Boundary2-3	539	021b
Loop.3.PID.CutbackHigh	530	0212
Loop.3.PID.CutbackHigh2	558	022e
Loop.3.PID.CutbackHigh3	568	0238
Loop.3.PID.CutbackLow	529	0211
Loop.3.PID.CutbackLow2	559	022f
Loop.3.PID.CutbackLow3	569	0239
Loop.3.PID.DerivativeTime	521	0209
Loop.3.PID.DerivativeTime2	557	022d
Loop.3.PID.DerivativeTime3	567	0237
Loop.3.PID.IntegralTime	520	0208
Loop.3.PID.IntegralTime2	556	022c
Loop.3.PID.IntegralTime3	566	0236
Loop.3.PID.LoopBreakTime	552	0228
Loop.3.PID.LoopBreakTime2	561	0231
Loop.3.PID.LoopBreakTime3	571	023b
Loop.3.PID.ManualReset	551	0227
Loop.3.PID.ManualReset2	560	0230
Loop.3.PID.ManualReset3	570	023a
Loop.3.PID.NumSets	576	0240
Loop.3.PID.OutputHi	553	0229
Loop.3.PID.OutputHi2	563	0233
Loop.3.PID.OutputHi3	573	023d
Loop.3.PID.OutputLo	554	022a
Loop.3.PID.OutputLo2	564	0234
Loop.3.PID.OutputLo3	574	023e
Loop.3.PID.ProportionalBand	518	0206
Loop.3.PID.ProportionalBand2	555	022b
Loop.3.PID.ProportionalBand3	565	0235
Loop.3.PID.RelCh2Gain	531	0213
Loop.3.PID.RelCh2Gain2	562	0232
Loop.3.PID.RelCh2Gain3	572	023c
Loop.3.PID.SchedulerRemoteInput	577	0241
Loop.3.PID.SchedulerType	575	023f
Loop.3.Setup.CH1ControlType	534	0216
Loop.3.Setup.CH2ControlType	535	0217
Loop.3.Setup.ControlAction	519	0207
Loop.3.Setup.DerivativeType	537	0219
Loop.3.Setup.LoopType	533	0215
Loop.3.Setup.PBUnits	536	0218
Loop.3.SP.AltSP	580	0244
Loop.3.SP.AltSPSelect	581	0245
Loop.3.SP.ManualTrack	587	024b
Loop.3.SP.RangeHigh	524	020c
Loop.3.SP.RangeLow	523	020b
Loop.3.SP.Rate	582	0246
Loop.3.SP.RateDisable	583	0247
Loop.3.SP.RateDone	591	024f

Description	ModBus	HEX
Loop.3.SP.SP1	525	020d
Loop.3.SP.SP2	526	020e
Loop.3.SP.SPHighLimit	578	0242
Loop.3.SP.SPLowLimit	579	0243
Loop.3.SP.SPSelect	527	020f
Loop.3.SP.SPTrack	588	024c
Loop.3.SP.SPTrim	584	0248
Loop.3.SP.SPTrimHighLimit	585	0249
Loop.3.SP.SPTrimLowLimit	586	024a
Loop.3.SP.TrackPV	589	024d
Loop.3.SP.TrackSP	590	024e
Loop.3.Tune.AutotuneEnable	620	026c
Loop.3.Tune.OutputHighLimit	617	0269
Loop.3.Tune.OutputLowLimit	618	026a
Loop.3.Tune.Stage	623	026f
Loop.3.Tune.StageTime	624	0270
Loop.3.Tune.State	622	026e
Loop.3.Tune.StepSize	621	026d
Loop.3.Tune.Type	616	0268
Loop.4.Diag.DerivativeOutContrib	887	0377
Loop.4.Diag.Error	881	0371
Loop.4.Diag.IntegralOutContrib	886	0376
Loop.4.Diag.LoopBreakAlarm	884	0374
Loop.4.Diag.LoopMode	882	0372
Loop.4.Diag.PropOutContrib	885	0375
Loop.4.Diag.SBrk	888	0378
Loop.4.Diag.SchedCBH	800	0320
Loop.4.Diag.SchedCBL	801	0321
Loop.4.Diag.SchedLPBrk	803	0323
Loop.4.Diag.SchedMR	802	0322
Loop.4.Diag.SchedOPHi	805	0325
Loop.4.Diag.SchedOPLo	806	0326
Loop.4.Diag.SchedPB	797	031d
Loop.4.Diag.SchedR2G	804	0324
Loop.4.Diag.SchedTd	799	031f
Loop.4.Diag.SchedTi	798	031e
Loop.4.Diag.TargetOutVal	883	0373
Loop.4.Main.ActiveOut	772	0304
Loop.4.Main.AutoMan	778	030a
Loop.4.Main.Inhibit	788	0314
Loop.4.Main.PV	769	0301
Loop.4.Main.TargetSP	770	0302
Loop.4.Main.WorkingSP	773	0305
Loop.4.OP.Ch1OnOffHysteresis	852	0354
Loop.4.OP.Ch1Out	850	0352
Loop.4.OP.Ch2Deadband	784	0310
Loop.4.OP.Ch2OnOffHysteresis	853	0355
Loop.4.OP.Ch2Out	851	0353
Loop.4.OP.CoolType	861	035d
Loop.4.OP.EnablePowerFeedforward	859	035b
Loop.4.OP.FeedForwardGain	863	035f
Loop.4.OP.FeedForwardOffset	864	0360
Loop.4.OP.FeedForwardTrimLimit	865	0361
Loop.4.OP.FeedForwardType	862	035e
Loop.4.OP.FeedForwardVal	866	0362
Loop.4.OP.FF_Rem	871	0367
Loop.4.OP.ManualMode	858	035a
Loop.4.OP.ManualOutVal	771	0303

Description	ModBus	HEX
Loop.4.OP.MeasuredPower	860	035c
Loop.4.OP.OutputHighLimit	848	0350
Loop.4.OP.OutputLowLimit	849	0351
Loop.4.OP.Rate	854	0356
Loop.4.OP.RateDisable	855	0357
Loop.4.OP.RemOPH	870	0366
Loop.4.OP.RemOPL	869	0365
Loop.4.OP.SafeOutVal	857	0359
Loop.4.OP.SensorBreakMode	856	0358
Loop.4.OP.TrackEnable	868	0364
Loop.4.OP.TrackOutVal	867	0363
Loop.4.PID.ActiveSet	796	031c
Loop.4.PID.Boundary1-2	794	031a
Loop.4.PID.Boundary2-3	795	031b
Loop.4.PID.CutbackHigh	786	0312
Loop.4.PID.CutbackHigh2	814	032e
Loop.4.PID.CutbackHigh3	824	0338
Loop.4.PID.CutbackLow	785	0311
Loop.4.PID.CutbackLow2	815	032f
Loop.4.PID.CutbackLow3	825	0339
Loop.4.PID.DerivativeTime	777	0309
Loop.4.PID.DerivativeTime2	813	032d
Loop.4.PID.DerivativeTime3	823	0337
Loop.4.PID.IntegralTime	776	0308
Loop.4.PID.IntegralTime2	812	032c
Loop.4.PID.IntegralTime3	822	0336
Loop.4.PID.LoopBreakTime	808	0328
Loop.4.PID.LoopBreakTime2	817	0331
Loop.4.PID.LoopBreakTime3	827	033b
Loop.4.PID.ManualReset	807	0327
Loop.4.PID.ManualReset2	816	0330
Loop.4.PID.ManualReset3	826	033a
Loop.4.PID.NumSets	832	0340
Loop.4.PID.OutputHi	809	0329
Loop.4.PID.OutputHi2	819	0333
Loop.4.PID.OutputHi3	829	033d
Loop.4.PID.OutputLo	810	032a
Loop.4.PID.OutputLo2	820	0334
Loop.4.PID.OutputLo3	830	033e
Loop.4.PID.ProportionalBand	774	0306
Loop.4.PID.ProportionalBand2	811	032b
Loop.4.PID.ProportionalBand3	821	0335
Loop.4.PID.RelCh2Gain	787	0313
Loop.4.PID.RelCh2Gain2	818	0332
Loop.4.PID.RelCh2Gain3	828	033c
Loop.4.PID.SchedulerRemoteInput	833	0341
Loop.4.PID.SchedulerType	831	033f
Loop.4.Setup.CH1ControlType	790	0316
Loop.4.Setup.CH2ControlType	791	0317
Loop.4.Setup.ControlAction	775	0307
Loop.4.Setup.DerivativeType	793	0319
Loop.4.Setup.LoopType	789	0315
Loop.4.Setup.PBUnits	792	0318
Loop.4.SP.AltSP	836	0344
Loop.4.SP.AltSPSelect	837	0345
Loop.4.SP.ManualTrack	843	034b
Loop.4.SP.RangeHigh	780	030c
Loop.4.SP.RangeLow	779	030b

Description	ModBus	HEX
Loop.4.SP.Rate	838	0346
Loop.4.SP.RateDisable	839	0347
Loop.4.SP.RateDone	847	034f
Loop.4.SP.SP1	781	030d
Loop.4.SP.SP2	782	030e
Loop.4.SP.SPHighLimit	834	0342
Loop.4.SP.SPLowLimit	835	0343
Loop.4.SP.SPSelect	783	030f
Loop.4.SP.SPTrack	844	034c
Loop.4.SP.SPTrim	840	0348
Loop.4.SP.SPTrimHighLimit	841	0349
Loop.4.SP.SPTrimLowLimit	842	034a
Loop.4.SP.TrackPV	845	034d
Loop.4.SP.TrackSP	846	034e
Loop.4.Tune.AutotuneEnable	876	036c
Loop.4.Tune.OutputHighLimit	873	0369
Loop.4.Tune.OutputLowLimit	874	036a
Loop.4.Tune.Stage	879	036f
Loop.4.Tune.StageTime	880	0370
Loop.4.Tune.State	878	036e
Loop.4.Tune.StepSize	877	036d
Loop.4.Tune.Type	872	0368
Loop.5.Diag.DerivativeOutContrib	1143	0477
Loop.5.Diag.Error	1137	0471
Loop.5.Diag.IntegralOutContrib	1142	0476
Loop.5.Diag.LoopBreakAlarm	1140	0474
Loop.5.Diag.LoopMode	1138	0472
Loop.5.Diag.PropOutContrib	1141	0475
Loop.5.Diag.SBrk	1144	0478
Loop.5.Diag.SchedCBH	1056	0420
Loop.5.Diag.SchedCBL	1057	0421
Loop.5.Diag.SchedLPBrk	1059	0423
Loop.5.Diag.SchedMR	1058	0422
Loop.5.Diag.SchedOPHi	1061	0425
Loop.5.Diag.SchedOPLo	1062	0426
Loop.5.Diag.SchedPB	1053	041d
Loop.5.Diag.SchedR2G	1060	0424
Loop.5.Diag.SchedTd	1055	041f
Loop.5.Diag.SchedTi	1054	041e
Loop.5.Diag.TargetOutVal	1139	0473
Loop.5.Main.ActiveOut	1028	0404
Loop.5.Main.AutoMan	1034	040a
Loop.5.Main.Inhibit	1044	0414
Loop.5.Main.PV	1025	0401
Loop.5.Main.TargetSP	1026	0402
Loop.5.Main.WorkingSP	1029	0405
Loop.5.OP.Ch1OnOffHysteresis	1108	0454
Loop.5.OP.Ch1Out	1106	0452
Loop.5.OP.Ch2Deadband	1040	0410
Loop.5.OP.Ch2OnOffHysteresis	1109	0455
Loop.5.OP.Ch2Out	1107	0453
Loop.5.OP.CoolType	1117	045d
Loop.5.OP.EnablePowerFeedforward	1115	045b
Loop.5.OP.FeedForwardGain	1119	045f
Loop.5.OP.FeedForwardOffset	1120	0460
Loop.5.OP.FeedForwardTrimLimit	1121	0461
Loop.5.OP.FeedForwardType	1118	045e
Loop.5.OP.FeedForwardVal	1122	0462

Description	ModBus	HEX
Loop.5.OP.FF_Rem	1127	0467
Loop.5.OP.ManualMode	1114	045a
Loop.5.OP.ManualOutVal	1027	0403
Loop.5.OP.MeasuredPower	1116	045c
Loop.5.OP.OutputHighLimit	1104	0450
Loop.5.OP.OutputLowLimit	1105	0451
Loop.5.OP.Rate	1110	0456
Loop.5.OP.RateDisable	1111	0457
Loop.5.OP.RemOPH	1126	0466
Loop.5.OP.RemOPL	1125	0465
Loop.5.OP.SafeOutVal	1113	0459
Loop.5.OP.SensorBreakMode	1112	0458
Loop.5.OP.TrackEnable	1124	0464
Loop.5.OP.TrackOutVal	1123	0463
Loop.5.PID.ActiveSet	1052	041c
Loop.5.PID.Boundary1-2	1050	041a
Loop.5.PID.Boundary2-3	1051	041b
Loop.5.PID.CutbackHigh	1042	0412
Loop.5.PID.CutbackHigh2	1070	042e
Loop.5.PID.CutbackHigh3	1080	0438
Loop.5.PID.CutbackLow	1041	0411
Loop.5.PID.CutbackLow2	1071	042f
Loop.5.PID.CutbackLow3	1081	0439
Loop.5.PID.DerivativeTime	1033	0409
Loop.5.PID.DerivativeTime2	1069	042d
Loop.5.PID.DerivativeTime3	1079	0437
Loop.5.PID.IntegralTime	1032	0408
Loop.5.PID.IntegralTime2	1068	042c
Loop.5.PID.IntegralTime3	1078	0436
Loop.5.PID.LoopBreakTime	1064	0428
Loop.5.PID.LoopBreakTime2	1073	0431
Loop.5.PID.LoopBreakTime3	1083	043b
Loop.5.PID.ManualReset	1063	0427
Loop.5.PID.ManualReset2	1072	0430
Loop.5.PID.ManualReset3	1082	043a
Loop.5.PID.NumSets	1088	0440
Loop.5.PID.OutputHi	1065	0429
Loop.5.PID.OutputHi2	1075	0433
Loop.5.PID.OutputHi3	1085	043d
Loop.5.PID.OutputLo	1066	042a
Loop.5.PID.OutputLo2	1076	0434
Loop.5.PID.OutputLo3	1086	043e
Loop.5.PID.ProportionalBand	1030	0406
Loop.5.PID.ProportionalBand2	1067	042b
Loop.5.PID.ProportionalBand3	1077	0435
Loop.5.PID.RelCh2Gain	1043	0413
Loop.5.PID.RelCh2Gain2	1074	0432
Loop.5.PID.RelCh2Gain3	1084	043c
Loop.5.PID.SchedulerRemoteInput	1089	0441
Loop.5.PID.SchedulerType	1087	043f
Loop.5.Setup.CH1ControlType	1046	0416
Loop.5.Setup.CH2ControlType	1047	0417
Loop.5.Setup.ControlAction	1031	0407
Loop.5.Setup.DerivativeType	1049	0419
Loop.5.Setup.LoopType	1045	0415
Loop.5.Setup.PBUnits	1048	0418
Loop.5.SP.AltSP	1092	0444
Loop.5.SP.AltSPSelect	1093	0445

Description	ModBus	HEX
Loop.5.SP.ManualTrack	1099	044b
Loop.5.SP.RangeHigh	1036	040c
Loop.5.SP.RangeLow	1035	040b
Loop.5.SP.Rate	1094	0446
Loop.5.SP.RateDisable	1095	0447
Loop.5.SP.RateDone	1103	044f
Loop.5.SP.SP1	1037	040d
Loop.5.SP.SP2	1038	040e
Loop.5.SP.SPHighLimit	1090	0442
Loop.5.SP.SPLowLimit	1091	0443
Loop.5.SP.SPSelect	1039	040f
Loop.5.SP.SPTrack	1100	044c
Loop.5.SP.SPTrim	1096	0448
Loop.5.SP.SPTrimHighLimit	1097	0449
Loop.5.SP.SPTrimLowLimit	1098	044a
Loop.5.SP.TrackPV	1101	044d
Loop.5.SP.TrackSP	1102	044e
Loop.5.Tune.AutotuneEnable	1132	046c
Loop.5.Tune.OutputHighLimit	1129	0469
Loop.5.Tune.OutputLowLimit	1130	046a
Loop.5.Tune.Stage	1135	046f
Loop.5.Tune.StageTime	1136	0470
Loop.5.Tune.State	1134	046e
Loop.5.Tune.StepSize	1133	046d
Loop.5.Tune.Type	1128	0468
Loop.6.Diag.DerivativeOutContrib	1399	0577
Loop.6.Diag.Error	1393	0571
Loop.6.Diag.IntegralOutContrib	1398	0576
Loop.6.Diag.LoopBreakAlarm	1396	0574
Loop.6.Diag.LoopMode	1394	0572
Loop.6.Diag.PropOutContrib	1397	0575
Loop.6.Diag.SBrk	1400	0578
Loop.6.Diag.SchedCBH	1312	0520
Loop.6.Diag.SchedCBL	1313	0521
Loop.6.Diag.SchedLPBrk	1315	0523
Loop.6.Diag.SchedMR	1314	0522
Loop.6.Diag.SchedOPHi	1317	0525
Loop.6.Diag.SchedOPLo	1318	0526
Loop.6.Diag.SchedPB	1309	051d
Loop.6.Diag.SchedR2G	1316	0524
Loop.6.Diag.SchedTd	1311	051f
Loop.6.Diag.SchedTi	1310	051e
Loop.6.Diag.TargetOutVal	1395	0573
Loop.6.Main.ActiveOut	1284	0504
Loop.6.Main.AutoMan	1290	050a
Loop.6.Main.Inhibit	1300	0514
Loop.6.Main.PV	1281	0501
Loop.6.Main.TargetSP	1282	0502
Loop.6.Main.WorkingSP	1285	0505
Loop.6.OP.Ch1OnOffHysteresis	1364	0554
Loop.6.OP.Ch1Out	1362	0552
Loop.6.OP.Ch2Deadband	1296	0510
Loop.6.OP.Ch2OnOffHysteresis	1365	0555
Loop.6.OP.Ch2Out	1363	0553
Loop.6.OP.CoolType	1373	055d
Loop.6.OP.EnablePowerFeedforward	1371	055b
Loop.6.OP.FeedForwardGain	1375	055f
Loop.6.OP.FeedForwardOffset	1376	0560

Description	ModBus	HEX
Loop.6.OP.FeedForwardTrimLimit	1377	0561
Loop.6.OP.FeedForwardType	1374	055e
Loop.6.OP.FeedForwardVal	1378	0562
Loop.6.OP.FF_Rem	1383	0567
Loop.6.OP.ManualMode	1370	055a
Loop.6.OP.ManualOutVal	1283	0503
Loop.6.OP.MeasuredPower	1372	055c
Loop.6.OP.OutputHighLimit	1360	0550
Loop.6.OP.OutputLowLimit	1361	0551
Loop.6.OP.Rate	1366	0556
Loop.6.OP.RateDisable	1367	0557
Loop.6.OP.RemOPH	1382	0566
Loop.6.OP.RemOPL	1381	0565
Loop.6.OP.SafeOutVal	1369	0559
Loop.6.OP.SensorBreakMode	1368	0558
Loop.6.OP.TrackEnable	1380	0564
Loop.6.OP.TrackOutVal	1379	0563
Loop.6.PID.ActiveSet	1308	051c
Loop.6.PID.Boundary1-2	1306	051a
Loop.6.PID.Boundary2-3	1307	051b
Loop.6.PID.CutbackHigh	1298	0512
Loop.6.PID.CutbackHigh2	1326	052e
Loop.6.PID.CutbackHigh3	1336	0538
Loop.6.PID.CutbackLow	1297	0511
Loop.6.PID.CutbackLow2	1327	052f
Loop.6.PID.CutbackLow3	1337	0539
Loop.6.PID.DerivativeTime	1289	0509
Loop.6.PID.DerivativeTime2	1325	052d
Loop.6.PID.DerivativeTime3	1335	0537
Loop.6.PID.IntegralTime	1288	0508
Loop.6.PID.IntegralTime2	1324	052c
Loop.6.PID.IntegralTime3	1334	0536
Loop.6.PID.LoopBreakTime	1320	0528
Loop.6.PID.LoopBreakTime2	1329	0531
Loop.6.PID.LoopBreakTime3	1339	053b
Loop.6.PID.ManualReset	1319	0527
Loop.6.PID.ManualReset2	1328	0530
Loop.6.PID.ManualReset3	1338	053a
Loop.6.PID.NumSets	1344	0540
Loop.6.PID.OutputHi	1321	0529
Loop.6.PID.OutputHi2	1331	0533
Loop.6.PID.OutputHi3	1341	053d
Loop.6.PID.OutputLo	1322	052a
Loop.6.PID.OutputLo2	1332	0534
Loop.6.PID.OutputLo3	1342	053e
Loop.6.PID.ProportionalBand	1286	0506
Loop.6.PID.ProportionalBand2	1323	052b
Loop.6.PID.ProportionalBand3	1333	0535
Loop.6.PID.RelCh2Gain	1299	0513
Loop.6.PID.RelCh2Gain2	1330	0532
Loop.6.PID.RelCh2Gain3	1340	053c
Loop.6.PID.SchedulerRemoteInput	1345	0541
Loop.6.PID.SchedulerType	1343	053f
Loop.6.Setup.CH1ControlType	1302	0516
Loop.6.Setup.CH2ControlType	1303	0517
Loop.6.Setup.ControlAction	1287	0507
Loop.6.Setup.DerivativeType	1305	0519
Loop.6.Setup.LoopType	1301	0515

Description	ModBus	HEX
Loop.6.Setup.PBUnits	1304	0518
Loop.6.SP.AltSP	1348	0544
Loop.6.SP.AltSPSelect	1349	0545
Loop.6.SP.ManualTrack	1355	054b
Loop.6.SP.RangeHigh	1292	050c
Loop.6.SP.RangeLow	1291	050b
Loop.6.SP.Rate	1350	0546
Loop.6.SP.RateDisable	1351	0547
Loop.6.SP.RateDone	1359	054f
Loop.6.SP.SP1	1293	050d
Loop.6.SP.SP2	1294	050e
Loop.6.SP.SPHighLimit	1346	0542
Loop.6.SP.SPLowLimit	1347	0543
Loop.6.SP.SPSelect	1295	050f
Loop.6.SP.SPTrack	1356	054c
Loop.6.SP.SPTrim	1352	0548
Loop.6.SP.SPTrimHighLimit	1353	0549
Loop.6.SP.SPTrimLowLimit	1354	054a
Loop.6.SP.TrackPV	1357	054d
Loop.6.SP.TrackSP	1358	054e
Loop.6.Tune.AutotuneEnable	1388	056c
Loop.6.Tune.OutputHighLimit	1385	0569
Loop.6.Tune.OutputLowLimit	1386	056a
Loop.6.Tune.Stage	1391	056f
Loop.6.Tune.StageTime	1392	0570
Loop.6.Tune.State	1390	056e
Loop.6.Tune.StepSize	1389	056d
Loop.6.Tune.Type	1384	0568
Loop.7.Diag.DerivativeOutContrib	1655	0677
Loop.7.Diag.Error	1649	0671
Loop.7.Diag.IntegralOutContrib	1654	0676
Loop.7.Diag.LoopBreakAlarm	1652	0674
Loop.7.Diag.LoopMode	1650	0672
Loop.7.Diag.PropOutContrib	1653	0675
Loop.7.Diag.SBrk	1656	0678
Loop.7.Diag.SchedCBH	1568	0620
Loop.7.Diag.SchedCBL	1569	0621
Loop.7.Diag.SchedLPBrk	1571	0623
Loop.7.Diag.SchedMR	1570	0622
Loop.7.Diag.SchedOPHi	1573	0625
Loop.7.Diag.SchedOPLo	1574	0626
Loop.7.Diag.SchedPB	1565	061d
Loop.7.Diag.SchedR2G	1572	0624
Loop.7.Diag.SchedTd	1567	061f
Loop.7.Diag.SchedTi	1566	061e
Loop.7.Diag.TargetOutVal	1651	0673
Loop.7.Main.ActiveOut	1540	0604
Loop.7.Main.AutoMan	1546	060a
Loop.7.Main.Inhibit	1556	0614
Loop.7.Main.PV	1537	0601
Loop.7.Main.TargetSP	1538	0602
Loop.7.Main.WorkingSP	1541	0605
Loop.7.OP.Ch1OnOffHysteresis	1620	0654
Loop.7.OP.Ch1Out	1618	0652
Loop.7.OP.Ch2Deadband	1552	0610
Loop.7.OP.Ch2OnOffHysteresis	1621	0655
Loop.7.OP.Ch2Out	1619	0653
Loop.7.OP.CoolType	1629	065d

Description	ModBus	HEX
Loop.7.OP.EnablePowerFeedforward	1627	065b
Loop.7.OP.FeedForwardGain	1631	065f
Loop.7.OP.FeedForwardOffset	1632	0660
Loop.7.OP.FeedForwardTrimLimit	1633	0661
Loop.7.OP.FeedForwardType	1630	065e
Loop.7.OP.FeedForwardVal	1634	0662
Loop.7.OP.FF_Rem	1639	0667
Loop.7.OP.ManualMode	1626	065a
Loop.7.OP.ManualOutVal	1539	0603
Loop.7.OP.MeasuredPower	1628	065c
Loop.7.OP.OutputHighLimit	1616	0650
Loop.7.OP.OutputLowLimit	1617	0651
Loop.7.OP.Rate	1622	0656
Loop.7.OP.RateDisable	1623	0657
Loop.7.OP.RemOPH	1638	0666
Loop.7.OP.RemOPL	1637	0665
Loop.7.OP.SafeOutVal	1625	0659
Loop.7.OP.SensorBreakMode	1624	0658
Loop.7.OP.TrackEnable	1636	0664
Loop.7.OP.TrackOutVal	1635	0663
Loop.7.PID.ActiveSet	1564	061c
Loop.7.PID.Boundary1-2	1562	061a
Loop.7.PID.Boundary2-3	1563	061b
Loop.7.PID.CutbackHigh	1554	0612
Loop.7.PID.CutbackHigh2	1582	062e
Loop.7.PID.CutbackHigh3	1592	0638
Loop.7.PID.CutbackLow	1553	0611
Loop.7.PID.CutbackLow2	1583	062f
Loop.7.PID.CutbackLow3	1593	0639
Loop.7.PID.DerivativeTime	1545	0609
Loop.7.PID.DerivativeTime2	1581	062d
Loop.7.PID.DerivativeTime3	1591	0637
Loop.7.PID.IntegralTime	1544	0608
Loop.7.PID.IntegralTime2	1580	062c
Loop.7.PID.IntegralTime3	1590	0636
Loop.7.PID.LoopBreakTime	1576	0628
Loop.7.PID.LoopBreakTime2	1585	0631
Loop.7.PID.LoopBreakTime3	1595	063b
Loop.7.PID.ManualReset	1575	0627
Loop.7.PID.ManualReset2	1584	0630
Loop.7.PID.ManualReset3	1594	063a
Loop.7.PID.NumSets	1600	0640
Loop.7.PID.OutputHi	1577	0629
Loop.7.PID.OutputHi2	1587	0633
Loop.7.PID.OutputHi3	1597	063d
Loop.7.PID.OutputLo	1578	062a
Loop.7.PID.OutputLo2	1588	0634
Loop.7.PID.OutputLo3	1598	063e
Loop.7.PID.ProportionalBand	1542	0606
Loop.7.PID.ProportionalBand2	1579	062b
Loop.7.PID.ProportionalBand3	1589	0635
Loop.7.PID.RelCh2Gain	1555	0613
Loop.7.PID.RelCh2Gain2	1586	0632
Loop.7.PID.RelCh2Gain3	1596	063c
Loop.7.PID.SchedulerRemoteInput	1601	0641
Loop.7.PID.SchedulerType	1599	063f
Loop.7.Setup.CH1ControlType	1558	0616
Loop.7.Setup.CH2ControlType	1559	0617

Description	ModBus	HEX
Loop.7.Setup.ControlAction	1543	0607
Loop.7.Setup.DerivativeType	1561	0619
Loop.7.Setup.LoopType	1557	0615
Loop.7.Setup.PBUnits	1560	0618
Loop.7.SP.AltSP	1604	0644
Loop.7.SP.AltSPSelect	1605	0645
Loop.7.SP.ManualTrack	1611	064b
Loop.7.SP.RangeHigh	1548	060c
Loop.7.SP.RangeLow	1547	060b
Loop.7.SP.Rate	1606	0646
Loop.7.SP.RateDisable	1607	0647
Loop.7.SP.RateDone	1615	064f
Loop.7.SP.SP1	1549	060d
Loop.7.SP.SP2	1550	060e
Loop.7.SP.SPHighLimit	1602	0642
Loop.7.SP.SPLowLimit	1603	0643
Loop.7.SP.SPSelect	1551	060f
Loop.7.SP.SPTrack	1612	064c
Loop.7.SP.SPTrim	1608	0648
Loop.7.SP.SPTrimHighLimit	1609	0649
Loop.7.SP.SPTrimLowLimit	1610	064a
Loop.7.SP.TrackPV	1613	064d
Loop.7.SP.TrackSP	1614	064e
Loop.7.Tune.AutotuneEnable	1644	066c
Loop.7.Tune.OutputHighLimit	1641	0669
Loop.7.Tune.OutputLowLimit	1642	066a
Loop.7.Tune.Stage	1647	066f
Loop.7.Tune.StageTime	1648	0670
Loop.7.Tune.State	1646	066e
Loop.7.Tune.StepSize	1645	066d
Loop.7.Tune.Type	1640	0668
Loop.8.Diag.DerivativeOutContrib	1911	0777
Loop.8.Diag.Error	1905	0771
Loop.8.Diag.IntegralOutContrib	1910	0776
Loop.8.Diag.LoopBreakAlarm	1908	0774
Loop.8.Diag.LoopMode	1906	0772
Loop.8.Diag.PropOutContrib	1909	0775
Loop.8.Diag.SBrk	1912	0778
Loop.8.Diag.SchedCBH	1824	0720
Loop.8.Diag.SchedCBL	1825	0721
Loop.8.Diag.SchedLPBrk	1827	0723
Loop.8.Diag.SchedMR	1826	0722
Loop.8.Diag.SchedOPHi	1829	0725
Loop.8.Diag.SchedOPLo	1830	0726
Loop.8.Diag.SchedPB	1821	071d
Loop.8.Diag.SchedR2G	1828	0724
Loop.8.Diag.SchedTd	1823	071f
Loop.8.Diag.SchedTi	1822	071e
Loop.8.Diag.TargetOutVal	1907	0773
Loop.8.Main.ActiveOut	1796	0704
Loop.8.Main.AutoMan	1802	070a
Loop.8.Main.Inhibit	1812	0714
Loop.8.Main.PV	1793	0701
Loop.8.Main.TargetSP	1794	0702
Loop.8.Main.WorkingSP	1797	0705
Loop.8.OP.Ch1OnOffHysteresis	1876	0754
Loop.8.OP.Ch1Out	1874	0752
Loop.8.OP.Ch2Deadband	1808	0710

Description	ModBus	HEX
Loop.8.OP.Ch2OnOffHysteresis	1877	0755
Loop.8.OP.Ch2Out	1875	0753
Loop.8.OP.CoolType	1885	075d
Loop.8.OP.EnablePowerFeedforward	1883	075b
Loop.8.OP.FeedForwardGain	1887	075f
Loop.8.OP.FeedForwardOffset	1888	0760
Loop.8.OP.FeedForwardTrimLimit	1889	0761
Loop.8.OP.FeedForwardType	1886	075e
Loop.8.OP.FeedForwardVal	1890	0762
Loop.8.OP.FF_Rem	1895	0767
Loop.8.OP.ManualMode	1882	075a
Loop.8.OP.ManualOutVal	1795	0703
Loop.8.OP.MeasuredPower	1884	075c
Loop.8.OP.OutputHighLimit	1872	0750
Loop.8.OP.OutputLowLimit	1873	0751
Loop.8.OP.Rate	1878	0756
Loop.8.OP.RateDisable	1879	0757
Loop.8.OP.RemOPH	1894	0766
Loop.8.OP.RemOPL	1893	0765
Loop.8.OP.SafeOutVal	1881	0759
Loop.8.OP.SensorBreakMode	1880	0758
Loop.8.OP.TrackEnable	1892	0764
Loop.8.OP.TrackOutVal	1891	0763
Loop.8.PID.ActiveSet	1820	071c
Loop.8.PID.Boundary1-2	1818	071a
Loop.8.PID.Boundary2-3	1819	071b
Loop.8.PID.CutbackHigh	1810	0712
Loop.8.PID.CutbackHigh2	1838	072e
Loop.8.PID.CutbackHigh3	1848	0738
Loop.8.PID.CutbackLow	1809	0711
Loop.8.PID.CutbackLow2	1839	072f
Loop.8.PID.CutbackLow3	1849	0739
Loop.8.PID.DerivativeTime	1801	0709
Loop.8.PID.DerivativeTime2	1837	072d
Loop.8.PID.DerivativeTime3	1847	0737
Loop.8.PID.IntegralTime	1800	0708
Loop.8.PID.IntegralTime2	1836	072c
Loop.8.PID.IntegralTime3	1846	0736
Loop.8.PID.LoopBreakTime	1832	0728
Loop.8.PID.LoopBreakTime2	1841	0731
Loop.8.PID.LoopBreakTime3	1851	073b
Loop.8.PID.ManualReset	1831	0727
Loop.8.PID.ManualReset2	1840	0730
Loop.8.PID.ManualReset3	1850	073a
Loop.8.PID.NumSets	1856	0740
Loop.8.PID.OutputHi	1833	0729
Loop.8.PID.OutputHi2	1843	0733
Loop.8.PID.OutputHi3	1853	073d
Loop.8.PID.OutputLo	1834	072a
Loop.8.PID.OutputLo2	1844	0734
Loop.8.PID.OutputLo3	1854	073e
Loop.8.PID.ProportionalBand	1798	0706
Loop.8.PID.ProportionalBand2	1835	072b
Loop.8.PID.ProportionalBand3	1845	0735
Loop.8.PID.RelCh2Gain	1811	0713
Loop.8.PID.RelCh2Gain2	1842	0732
Loop.8.PID.RelCh2Gain3	1852	073c
Loop.8.PID.SchedulerRemoteInput	1857	0741

Description	ModBus	HEX
Loop.8.PID.SchedulerType	1855	073f
Loop.8.Setup.CH1ControlType	1814	0716
Loop.8.Setup.CH2ControlType	1815	0717
Loop.8.Setup.ControlAction	1799	0707
Loop.8.Setup.DerivativeType	1817	0719
Loop.8.Setup.LoopType	1813	0715
Loop.8.Setup.PBUnits	1816	0718
Loop.8.SP.AltSP	1860	0744
Loop.8.SP.AltSPSelect	1861	0745
Loop.8.SP.ManualTrack	1867	074b
Loop.8.SP.RangeHigh	1804	070c
Loop.8.SP.RangeLow	1803	070b
Loop.8.SP.Rate	1862	0746
Loop.8.SP.RateDisable	1863	0747
Loop.8.SP.RateDone	1871	074f
Loop.8.SP.SP1	1805	070d
Loop.8.SP.SP2	1806	070e
Loop.8.SP.SPHighLimit	1858	0742
Loop.8.SP.SPLowLimit	1859	0743
Loop.8.SP.SPSelect	1807	070f
Loop.8.SP.SPTrack	1868	074c
Loop.8.SP.SPTrim	1864	0748
Loop.8.SP.SPTrimHighLimit	1865	0749
Loop.8.SP.SPTrimLowLimit	1866	074a
Loop.8.SP.TrackPV	1869	074d
Loop.8.SP.TrackSP	1870	074e
Loop.8.Tune.AutotuneEnable	1900	076c
Loop.8.Tune.OutputHighLimit	1897	0769
Loop.8.Tune.OutputLowLimit	1898	076a
Loop.8.Tune.Stage	1903	076f
Loop.8.Tune.StageTime	1904	0770
Loop.8.Tune.State	1902	076e
Loop.8.Tune.StepSize	1901	076d
Loop.8.Tune.Type	1896	0768
Math2.1.In1	4750	128e
Math2.1.In2	4751	128f
Math2.1.Out	4752	1290
Math2.2.In1	4753	1291
Math2.2.In2	4754	1292
Math2.2.Out	4755	1293
Math2.3.In1	4756	1294
Math2.3.In2	4757	1295
Math2.3.Out	4758	1296
Math2.4.In1	4759	1297
Math2.4.In2	4760	1298
Math2.4.Out	4761	1299
Math2.5.In1	4762	129a
Math2.5.In2	4763	129b
Math2.5.Out	4764	129c
Math2.6.In1	4765	129d
Math2.6.In2	4766	129e
Math2.6.Out	4767	129f
Math2.7.In1	4768	12a0
Math2.7.In2	4769	12a1
Math2.7.Out	4770	12a2
Math2.8.In1	4771	12a3
Math2.8.In2	4772	12a4
Math2.8.Out	4773	12a5

Description	ModBus	HEX
Math2.9.In1	4774	12a6
Math2.9.In2	4775	12a7
Math2.9.Out	4776	12a8
Math2.10.In1	4777	12a9
Math2.10.In2	4778	12aa
Math2.10.Out	4779	12ab
Math2.11.In1	4780	12ac
Math2.11.In2	4781	12ad
Math2.11.Out	4782	12ae
Math2.12.In1	4783	12af
Math2.12.In2	4784	12b0
Math2.12.Out	4785	12b1
Math2.13.In1	4786	12b2
Math2.13.In2	4787	12b3
Math2.13.Out	4788	12b4
Math2.14.In1	4789	12b5
Math2.14.In2	4790	12b6
Math2.14.Out	4791	12b7
Math2.15.In1	4792	12b8
Math2.15.In2	4793	12b9
Math2.15.Out	4794	12ba
Math2.16.In1	4795	12bb
Math2.16.In2	4796	12bc
Math2.16.Out	4797	12bd
Math2.17.In1	4798	12be
Math2.17.In2	4799	12bf
Math2.17.Out	4800	12c0
Math2.18.In1	4801	12c1
Math2.18.In2	4802	12c2
Math2.18.Out	4803	12c3
Math2.19.In1	4804	12c4
Math2.19.In2	4805	12c5
Math2.19.Out	4806	12c6
Math2.20.In1	4807	12c7
Math2.20.In2	4808	12c8
Math2.20.Out	4809	12c9
Math2.21.In1	4810	12ca
Math2.21.In2	4811	12cb
Math2.21.Out	4812	12cc
Math2.22.In1	4813	12cd
Math2.22.In2	4814	12ce
Math2.22.Out	4815	12cf
Math2.23.In1	4816	12d0
Math2.23.In2	4817	12d1
Math2.23.Out	4818	12d2
Math2.24.In1	4819	12d3
Math2.24.In2	4820	12d4
Math2.24.Out	4821	12d5
Program.Cycles	8196	2004
Program.DwellUnits	8195	2003
Program.HoldbackVal	8193	2001
Program.RampUnits	8194	2002
Programmer.CommsProgNum	8192	2000
Programmer.Run.CurProg	8201	2009
Programmer.Run.CurSeg	8202	200a
Programmer.Run.CurSegType	8206	200e
Programmer.Run.CyclesLeft	8205	200d
Programmer.Run.EventOuts	8212	2014

Description	ModBus	HEX
Programmer.Run.FastRun	8216	2018
Programmer.Run.ProgStatus	8203	200b
Programmer.Run.ProgTimeLeft	8209	2011
Programmer.Run.PSP	8204	200c
Programmer.Run.ResetEventOuts	8200	2008
Programmer.Run.SegRate	8208	2010
Programmer.Run.SegTarget	8207	200f
Programmer.Run.SegTimeLeft	8213	2015
Programmer.Setup.AdvSeg	8217	2019
Programmer.Setup.EndOfSeg	8214	2016
Programmer.Setup.PowerFailAct	8197	2005
Programmer.Setup.PVIn	8210	2012
Programmer.Setup.Servo	8198	2006
Programmer.Setup.SkipSeg	8218	201a
Programmer.Setup.SPIn	8211	2013
Programmer.Setup.SyncIn	8215	2017
Programmer.Setup.SyncMode	8199	2007
Recipe.LastDataset	4913	1331
Recipe.LoadingStatus	4914	1332
Recipe.RecipeSelect	4912	1330
Segment.1.CallCycles	8259	2043
Segment.1.CallProg	8258	2042
Segment.1.Duration	8260	2044
Segment.1.EndType	8263	2047
Segment.1.EventOuts	8264	2048
Segment.1.Holdback	8257	2041
Segment.1.RampRate	8261	2045
Segment.1.SegType	8256	2040
Segment.1.TargetSP	8262	2046
Segment.2.CallCycles	8275	2053
Segment.2.CallProg	8274	2052
Segment.2.Duration	8276	2054
Segment.2.EndType	8279	2057
Segment.2.EventOuts	8280	2058
Segment.2.Holdback	8273	2051
Segment.2.RampRate	8277	2055
Segment.2.SegType	8272	2050
Segment.2.TargetSP	8278	2056
Segment.3.CallCycles	8291	2063
Segment.3.CallProg	8290	2062
Segment.3.Duration	8292	2064
Segment.3.EndType	8295	2067
Segment.3.EventOuts	8296	2068
Segment.3.Holdback	8289	2061
Segment.3.RampRate	8293	2065
Segment.3.SegType	8288	2060
Segment.3.TargetSP	8294	2066
Segment.4.CallCycles	8307	2073
Segment.4.CallProg	8306	2072
Segment.4.Duration	8308	2074
Segment.4.EndType	8311	2077
Segment.4.EventOuts	8312	2078
Segment.4.Holdback	8305	2071
Segment.4.RampRate	8309	2075
Segment.4.SegType	8304	2070
Segment.4.TargetSP	8310	2076
Segment.5.CallCycles	8323	2083
Segment.5.CallProg	8322	2082
Segment.5.Duration	8324	2084
Segment.5.EndType	8327	2087
Segment.5.EventOuts	8328	2088
Segment.5.Holdback	8321	2081
Segment.5.RampRate	8325	2085
Segment.5.SegType	8320	2080
Segment.5.TargetSP	8326	2086
Segment.6.CallCycles	8339	2093
Segment.6.CallProg	8338	2092
Segment.6.Duration	8340	2094
Segment.6.EndType	8343	2097
Segment.6.EventOuts	8344	2098
Segment.6.Holdback	8337	2091
Segment.6.RampRate	8341	2095
Segment.6.SegType	8336	2090
Segment.6.TargetSP	8342	2096
Segment.7.CallCycles	8355	20a3
Segment.7.CallProg	8354	20a2
Segment.7.Duration	8356	20a4
Segment.7.EndType	8359	20a7
Segment.7.EventOuts	8360	20a8
Segment.7.Holdback	8353	20a1
Segment.7.RampRate	8357	20a5
Segment.7.SegType	8352	20a0
Segment.7.TargetSP	8358	20a6
Segment.8.CallCycles	8371	20b3
Segment.8.CallProg	8370	20b2
Segment.8.Duration	8372	20b4
Segment.8.EndType	8375	20b7
Segment.8.EventOuts	8376	20b8
Segment.8.Holdback	8369	20b1
Segment.8.RampRate	8373	20b5
Segment.8.SegType	8368	20b0
Segment.8.TargetSP	8374	20b6
Segment.9.CallCycles	8387	20c3
Segment.9.CallProg	8386	20c2
Segment.9.Duration	8388	20c4
Segment.9.EndType	8391	20c7
Segment.9.EventOuts	8392	20c8
Segment.9.Holdback	8385	20c1
Segment.9.RampRate	8389	20c5
Segment.9.SegType	8384	20c0
Segment.9.TargetSP	8390	20c6
Segment.10.CallCycles	8403	20d3
Segment.10.CallProg	8402	20d2
Segment.10.Duration	8404	20d4
Segment.10.EndType	8407	20d7
Segment.10.EventOuts	8408	20d8
Segment.10.Holdback	8401	20d1
Segment.10.RampRate	8405	20d5
Segment.10.SegType	8400	20d0
Segment.10.TargetSP	8406	20d6
Segment.11.CallCycles	8419	20e3
Segment.11.CallProg	8418	20e2
Segment.11.Duration	8420	20e4
Segment.11.EndType	8423	20e7
Segment.11.EventOuts	8424	20e8
Segment.11.Holdback	8417	20e1

Description	ModBus	HEX
Segment.11.RampRate	8421	20e5
Segment.11.SegType	8416	20e0
Segment.11.TargetSP	8422	20e6
Segment.12.CallCycles	8435	20f3
Segment.12.CallProg	8434	20f2
Segment.12.Duration	8436	20f4
Segment.12.EndType	8439	20f7
Segment.12.EventOuts	8440	20f8
Segment.12.Holdback	8433	20f1
Segment.12.RampRate	8437	20f5
Segment.12.SegType	8432	20f0
Segment.12.TargetSP	8438	20f6
Segment.13.CallCycles	8451	2103
Segment.13.CallProg	8450	2102
Segment.13.Duration	8452	2104
Segment.13.EndType	8455	2107
Segment.13.EventOuts	8456	2108
Segment.13.Holdback	8449	2101
Segment.13.RampRate	8453	2105
Segment.13.SegType	8448	2100
Segment.13.TargetSP	8454	2106
Segment.14.CallCycles	8467	2113
Segment.14.CallProg	8466	2112
Segment.14.Duration	8468	2114
Segment.14.EndType	8471	2117
Segment.14.EventOuts	8472	2118
Segment.14.Holdback	8465	2111
Segment.14.RampRate	8469	2115
Segment.14.SegType	8464	2110
Segment.14.TargetSP	8470	2116
Segment.15.CallCycles	8483	2123
Segment.15.CallProg	8482	2122
Segment.15.Duration	8484	2124
Segment.15.EndType	8487	2127
Segment.15.EventOuts	8488	2128
Segment.15.Holdback	8481	2121
Segment.15.RampRate	8485	2125
Segment.15.SegType	8480	2120
Segment.15.TargetSP	8486	2126
Segment.16.CallCycles	8499	2133
Segment.16.CallProg	8498	2132
Segment.16.Duration	8500	2134
Segment.16.EndType	8503	2137
Segment.16.EventOuts	8504	2138
Segment.16.Holdback	8497	2131
Segment.16.RampRate	8501	2135
Segment.16.SegType	8496	2130
Segment.16.TargetSP	8502	2136
Segment.17.CallCycles	8515	2143
Segment.17.CallProg	8514	2142
Segment.17.Duration	8516	2144
Segment.17.EndType	8519	2147
Segment.17.EventOuts	8520	2148
Segment.17.Holdback	8513	2141
Segment.17.RampRate	8517	2145
Segment.17.SegType	8512	2140
Segment.17.TargetSP	8518	2146
Segment.18.CallCycles	8531	2153

Description	ModBus	HEX
Segment.18.CallProg	8530	2152
Segment.18.Duration	8532	2154
Segment.18.EndType	8535	2157
Segment.18.EventOuts	8536	2158
Segment.18.Holdback	8529	2151
Segment.18.RampRate	8533	2155
Segment.18.SegType	8528	2150
Segment.18.TargetSP	8534	2156
Segment.19.CallCycles	8547	2163
Segment.19.CallProg	8546	2162
Segment.19.Duration	8548	2164
Segment.19.EndType	8551	2167
Segment.19.EventOuts	8552	2168
Segment.19.Holdback	8545	2161
Segment.19.RampRate	8549	2165
Segment.19.SegType	8544	2160
Segment.19.TargetSP	8550	2166
Segment.20.CallCycles	8563	2173
Segment.20.CallProg	8562	2172
Segment.20.Duration	8564	2174
Segment.20.EndType	8567	2177
Segment.20.EventOuts	8568	2178
Segment.20.Holdback	8561	2171
Segment.20.RampRate	8565	2175
Segment.20.SegType	8560	2170
Segment.20.TargetSP	8566	2176
Segment.21.CallCycles	8579	2183
Segment.21.CallProg	8578	2182
Segment.21.Duration	8580	2184
Segment.21.EndType	8583	2187
Segment.21.EventOuts	8584	2188
Segment.21.Holdback	8577	2181
Segment.21.RampRate	8581	2185
Segment.21.SegType	8576	2180
Segment.21.TargetSP	8582	2186
Segment.22.CallCycles	8595	2193
Segment.22.CallProg	8594	2192
Segment.22.Duration	8596	2194
Segment.22.EndType	8599	2197
Segment.22.EventOuts	8600	2198
Segment.22.Holdback	8593	2191
Segment.22.RampRate	8597	2195
Segment.22.SegType	8592	2190
Segment.22.TargetSP	8598	2196
Segment.23.CallCycles	8611	21a3
Segment.23.CallProg	8610	21a2
Segment.23.Duration	8612	21a4
Segment.23.EndType	8615	21a7
Segment.23.EventOuts	8616	21a8
Segment.23.Holdback	8609	21a1
Segment.23.RampRate	8613	21a5
Segment.23.SegType	8608	21a0
Segment.23.TargetSP	8614	21a6
Segment.24.CallCycles	8627	21b3
Segment.24.CallProg	8626	21b2
Segment.24.Duration	8628	21b4
Segment.24.EndType	8631	21b7
Segment.24.EventOuts	8632	21b8

Description	ModBus	HEX
Segment.24.Holdback	8625	21b1
Segment.24.RampRate	8629	21b5
Segment.24.SegType	8624	21b0
Segment.24.TargetSP	8630	21b6
Segment.25.CallCycles	8643	21c3
Segment.25.CallProg	8642	21c2
Segment.25.Duration	8644	21c4
Segment.25.EndType	8647	21c7
Segment.25.EventOuts	8648	21c8
Segment.25.Holdback	8641	21c1
Segment.25.RampRate	8645	21c5
Segment.25.SegType	8640	21c0
Segment.25.TargetSP	8646	21c6
Segment.26.CallCycles	8659	21d3
Segment.26.CallProg	8658	21d2
Segment.26.Duration	8660	21d4
Segment.26.EndType	8663	21d7
Segment.26.EventOuts	8664	21d8
Segment.26.Holdback	8657	21d1
Segment.26.RampRate	8661	21d5
Segment.26.SegType	8656	21d0
Segment.26.TargetSP	8662	21d6
Segment.27.CallCycles	8675	21e3
Segment.27.CallProg	8674	21e2
Segment.27.Duration	8676	21e4
Segment.27.EndType	8679	21e7
Segment.27.EventOuts	8680	21e8
Segment.27.Holdback	8673	21e1
Segment.27.RampRate	8677	21e5
Segment.27.SegType	8672	21e0
Segment.27.TargetSP	8678	21e6
Segment.28.CallCycles	8691	21f3
Segment.28.CallProg	8690	21f2
Segment.28.Duration	8692	21f4
Segment.28.EndType	8695	21f7
Segment.28.EventOuts	8696	21f8
Segment.28.Holdback	8689	21f1
Segment.28.RampRate	8693	21f5
Segment.28.SegType	8688	21f0
Segment.28.TargetSP	8694	21f6
Segment.29.CallCycles	8707	2203
Segment.29.CallProg	8706	2202
Segment.29.Duration	8708	2204
Segment.29.EndType	8711	2207
Segment.29.EventOuts	8712	2208
Segment.29.Holdback	8705	2201
Segment.29.RampRate	8709	2205
Segment.29.SegType	8704	2200
Segment.29.TargetSP	8710	2206
Segment.30.CallCycles	8723	2213
Segment.30.CallProg	8722	2212
Segment.30.Duration	8724	2214
Segment.30.EndType	8727	2217
Segment.30.EventOuts	8728	2218
Segment.30.Holdback	8721	2211
Segment.30.RampRate	8725	2215
Segment.30.SegType	8720	2210
Segment.30.TargetSP	8726	2216

Description	ModBus	HEX
Segment.31.CallCycles	8739	2223
Segment.31.CallProg	8738	2222
Segment.31.Duration	8740	2224
Segment.31.EndType	8743	2227
Segment.31.EventOuts	8744	2228
Segment.31.Holdback	8737	2221
Segment.31.RampRate	8741	2225
Segment.31.SegType	8736	2220
Segment.31.TargetSP	8742	2226
Segment.32.CallCycles	8755	2233
Segment.32.CallProg	8754	2232
Segment.32.Duration	8756	2234
Segment.32.EndType	8759	2237
Segment.32.EventOuts	8760	2238
Segment.32.Holdback	8753	2231
Segment.32.RampRate	8757	2235
Segment.32.SegType	8752	2230
Segment.32.TargetSP	8758	2236
Segment.33.CallCycles	8771	2243
Segment.33.CallProg	8770	2242
Segment.33.Duration	8772	2244
Segment.33.EndType	8775	2247
Segment.33.EventOuts	8776	2248
Segment.33.Holdback	8769	2241
Segment.33.RampRate	8773	2245
Segment.33.SegType	8768	2240
Segment.33.TargetSP	8774	2246
Segment.34.CallCycles	8787	2253
Segment.34.CallProg	8786	2252
Segment.34.Duration	8788	2254
Segment.34.EndType	8791	2257
Segment.34.EventOuts	8792	2258
Segment.34.Holdback	8785	2251
Segment.34.RampRate	8789	2255
Segment.34.SegType	8784	2250
Segment.34.TargetSP	8790	2256
Segment.35.CallCycles	8803	2263
Segment.35.CallProg	8802	2262
Segment.35.Duration	8804	2264
Segment.35.EndType	8807	2267
Segment.35.EventOuts	8808	2268
Segment.35.Holdback	8801	2261
Segment.35.RampRate	8805	2265
Segment.35.SegType	8800	2260
Segment.35.TargetSP	8806	2266
Segment.36.CallCycles	8819	2273
Segment.36.CallProg	8818	2272
Segment.36.Duration	8820	2274
Segment.36.EndType	8823	2277
Segment.36.EventOuts	8824	2278
Segment.36.Holdback	8817	2271
Segment.36.RampRate	8821	2275
Segment.36.SegType	8816	2270
Segment.36.TargetSP	8822	2276
Segment.37.CallCycles	8835	2283
Segment.37.CallProg	8834	2282
Segment.37.Duration	8836	2284
Segment.37.EndType	8839	2287

Description	ModBus	HEX
Segment.37.EventOuts	8840	2288
Segment.37.Holdback	8833	2281
Segment.37.RampRate	8837	2285
Segment.37.SegType	8832	2280
Segment.37.TargetSP	8838	2286
Segment.38.CallCycles	8851	2293
Segment.38.CallProg	8850	2292
Segment.38.Duration	8852	2294
Segment.38.EndType	8855	2297
Segment.38.EventOuts	8856	2298
Segment.38.Holdback	8849	2291
Segment.38.RampRate	8853	2295
Segment.38.SegType	8848	2290
Segment.38.TargetSP	8854	2296
Segment.39.CallCycles	8867	22a3
Segment.39.CallProg	8866	22a2
Segment.39.Duration	8868	22a4
Segment.39.EndType	8871	22a7
Segment.39.EventOuts	8872	22a8
Segment.39.Holdback	8865	22a1
Segment.39.RampRate	8869	22a5
Segment.39.SegType	8864	22a0
Segment.39.TargetSP	8870	22a6
Segment.40.CallCycles	8883	22b3
Segment.40.CallProg	8882	22b2
Segment.40.Duration	8884	22b4
Segment.40.EndType	8887	22b7
Segment.40.EventOuts	8888	22b8
Segment.40.Holdback	8881	22b1
Segment.40.RampRate	8885	22b5
Segment.40.SegType	8880	22b0
Segment.40.TargetSP	8886	22b6
Segment.41.CallCycles	8899	22c3
Segment.41.CallProg	8898	22c2
Segment.41.Duration	8900	22c4
Segment.41.EndType	8903	22c7
Segment.41.EventOuts	8904	22c8
Segment.41.Holdback	8897	22c1
Segment.41.RampRate	8901	22c5
Segment.41.SegType	8896	22c0
Segment.41.TargetSP	8902	22c6
Segment.42.CallCycles	8915	22d3
Segment.42.CallProg	8914	22d2
Segment.42.Duration	8916	22d4
Segment.42.EndType	8919	22d7
Segment.42.EventOuts	8920	22d8
Segment.42.Holdback	8913	22d1
Segment.42.RampRate	8917	22d5
Segment.42.SegType	8912	22d0
Segment.42.TargetSP	8918	22d6
Segment.43.CallCycles	8931	22e3
Segment.43.CallProg	8930	22e2
Segment.43.Duration	8932	22e4
Segment.43.EndType	8935	22e7
Segment.43.EventOuts	8936	22e8
Segment.43.Holdback	8929	22e1
Segment.43.RampRate	8933	22e5
Segment.43.SegType	8928	22e0

Description	ModBus	HEX
Segment.43.TargetSP	8934	22e6
Segment.44.CallCycles	8947	22f3
Segment.44.CallProg	8946	22f2
Segment.44.Duration	8948	22f4
Segment.44.EndType	8951	22f7
Segment.44.EventOuts	8952	22f8
Segment.44.Holdback	8945	22f1
Segment.44.RampRate	8949	22f5
Segment.44.SegType	8944	22f0
Segment.44.TargetSP	8950	22f6
Segment.45.CallCycles	8963	2303
Segment.45.CallProg	8962	2302
Segment.45.Duration	8964	2304
Segment.45.EndType	8967	2307
Segment.45.EventOuts	8968	2308
Segment.45.Holdback	8961	2301
Segment.45.RampRate	8965	2305
Segment.45.SegType	8960	2300
Segment.45.TargetSP	8966	2306
Segment.46.CallCycles	8979	2313
Segment.46.CallProg	8978	2312
Segment.46.Duration	8980	2314
Segment.46.EndType	8983	2317
Segment.46.EventOuts	8984	2318
Segment.46.Holdback	8977	2311
Segment.46.RampRate	8981	2315
Segment.46.SegType	8976	2310
Segment.46.TargetSP	8982	2316
Segment.47.CallCycles	8995	2323
Segment.47.CallProg	8994	2322
Segment.47.Duration	8996	2324
Segment.47.EndType	8999	2327
Segment.47.EventOuts	9000	2328
Segment.47.Holdback	8993	2321
Segment.47.RampRate	8997	2325
Segment.47.SegType	8992	2320
Segment.47.TargetSP	8998	2326
Segment.48.CallCycles	9011	2333
Segment.48.CallProg	9010	2332
Segment.48.Duration	9012	2334
Segment.48.EndType	9015	2337
Segment.48.EventOuts	9016	2338
Segment.48.Holdback	9009	2331
Segment.48.RampRate	9013	2335
Segment.48.SegType	9008	2330
Segment.48.TargetSP	9014	2336
Segment.49.CallCycles	9027	2343
Segment.49.CallProg	9026	2342
Segment.49.Duration	9028	2344
Segment.49.EndType	9031	2347
Segment.49.EventOuts	9032	2348
Segment.49.Holdback	9025	2341
Segment.49.RampRate	9029	2345
Segment.49.SegType	9024	2340
Segment.49.TargetSP	9030	2346
Segment.50.CallCycles	9043	2353
Segment.50.CallProg	9042	2352
Segment.50.Duration	9044	2354

Description	ModBus	HEX
Segment.50.EndType	9047	2357
Segment.50.EventOuts	9048	2358
Segment.50.Holdback	9041	2351
Segment.50.RampRate	9045	2355
Segment.50.SegType	9040	2350
Segment.50.TargetSP	9046	2356
SwitchOver.SelectIn	4927	133f
SwitchOver.SwitchHigh	4925	133d
SwitchOver.SwitchLow	4926	133e
UsrVal.1.Val	4962	1362
UsrVal.2.Val	4963	1363
UsrVal.3.Val	4964	1364
UsrVal.4.Val	4965	1365
UsrVal.5.Val	4966	1366
UsrVal.6.Val	4967	1367
UsrVal.7.Val	4968	1368
UsrVal.8.Val	4969	1369
UsrVal.9.Val	4970	136a
UsrVal.10.Val	4971	136b
UsrVal.11.Val	4972	136c
UsrVal.12.Val	4973	136d
UsrVal.13.Val	4974	136e
UsrVal.14.Val	4975	136f
UsrVal.15.Val	4976	1370
UsrVal.16.Val	4977	1371
UsrVal.17.Val	4978	1372
UsrVal.18.Val	4979	1373
UsrVal.19.Val	4980	1374
UsrVal.20.Val	4981	1375
UsrVal.21.Val	4982	1376
UsrVal.22.Val	4983	1377
UsrVal.23.Val	4984	1378
UsrVal.24.Val	4985	1379
UsrVal.25.Val	4986	137a
UsrVal.26.Val	4987	137b
UsrVal.27.Val	4988	137c
UsrVal.28.Val	4989	137d
UsrVal.29.Val	4990	137e
UsrVal.30.Val	4991	137f
UsrVal.31.Val	4992	1380
UsrVal.32.Val	4993	1381

24. APPENDIX B DEVICENET PARAMETER TABLES

24.1 IO Re-Mapping Object

The Mini8 DeviceNet communicates is supplied with a default input assembly table (80 bytes) and output assembly table (48 bytes). The parameters included are listed below.

To modify these tables see the next section.

The default **Input** assembly table

Input Parameter	Offset	Value (Attr ID)
PV – Loop 1	0	0
Working SP – Loop 1	2	1
Working Output – Loop 1	4	2
PV – Loop 2	6	14 (0EH)
Working SP – Loop 2	8	15 (0FH)
Working Output – Loop 2	10	16 (10H)
PV – Loop 3	12	28 (1CH)
Working SP – Loop 3	14	29 (1DH)
Working Output – Loop 3	16	30 (1EH)
PV – Loop 4	18	42 (2AH)
Working SP – Loop 4	20	43 (2BH)
Working Output – Loop 4	22	44 (2CH)
PV – Loop 5	24	56 (38H)
Working SP – Loop 5	26	57 (39H)
Working Output – Loop 5	28	58 (3AH)
PV – Loop 6	30	70 (46H)
Working SP – Loop 6	32	71 (47H)
Working Output – Loop 6	34	72 (48H)
PV – Loop 7	36	84 (54H)
Working SP – Loop 7	38	85 (55H)
Working Output – Loop 7	40	86 (56H)
PV – Loop 8	42	98 (62H)
Working SP – Loop 8	44	99 (63H)
Working Output – Loop 8	46	100 (64H)
Analogue Alarm Status 1	48	144 (90H)
Analogue Alarm Status 2	50	145 (91H)
Analogue Alarm Status 3	52	146 (92H)
Analogue Alarm Status 4	54	147 (93H)
Sensor Break Alarm Status 1	56	148 (94H)
Sensor Break Alarm Status 2	58	149 (95H)
Sensor Break Alarm Status 3	60	150 (96H)
Sensor Break Alarm Status 4	62	151 (97H)
CT Alarm Status 1	64	152 (98H)
CT Alarm Status 2	66	153 (99H)
CT Alarm Status 3	68	154 (9AH)
CT Alarm Status 4	70	155 (9BH)
New Alarm Output	72	156 (9CH)
Any Alarm Output	74	157 (9DH)
New CT Alarm Output	76	158 (9EH)
Program Status	78	184 (B8H)
TOTAL LENGTH	80	

The default **output** assembly table.

Output Parameter	Offset	Value
Target SP – Loop 1	0	3
Auto/Manual – Loop 1	2	7
Manual Output – Loop 1	4	4
Target SP – Loop 2	6	17 (11H)
Auto/Manual – Loop 2	8	21 (15H)
Manual Output – Loop 2	10	18 (12H)
Target SP – Loop 3	12	31 (1FH)
Auto/Manual – Loop 3	14	35 (23H)
Manual Output – Loop 3	16	32 (20H)
Target SP – Loop 4	18	45 (2DH)
Auto/Manual – Loop 4	20	49 (31H)
Manual Output – Loop 4	22	46 (2EH)
Target SP – Loop 5	24	59 (3BH)
Auto/Manual – Loop 5	26	63 (3FH)
Manual Output – Loop 5	28	60 (3CH)
Target SP – Loop 6	30	73 (49H)
Auto/Manual – Loop 6	32	77 (4DH)
Manual Output – Loop 6	34	74 (4AH)
Target SP – Loop 7	36	87 (57H)
Auto/Manual – Loop 7	38	91 (5BH)
Manual Output – Loop 7	40	88 (58H)
Target SP – Loop 8	42	101 (65H)
Auto/Manual – Loop 8	44	105 (69H)
Manual Output – Loop 8	46	102 (66H)
TOTAL LENGTH	48	

24.2 Application Variables Object

This is the list of parameters available to be included in the input and output tables.

Parameter	Attribute ID
Process Variable – Loop 1	0
Working Setpoint – Loop 1	1
Working Output – Loop 1	2
Target Setpoint – Loop 1	3
Manual Output – Loop 1	4
Setpoint 1 – Loop 1	5
Setpoint 2 – Loop 1	6
Auto/Manual Mode – Loop 1	7
Proportional Band – Loop 1 working Set	8
Integral Time – Loop 1 working Set	9
Derivative Time – Loop 1 working Set	10
Cutback Low – Loop 1 working Set	11
Cutback High – Loop 1 working Set	12
Relative Cooling Gain – Loop 1 working Set	13
Process Variable – Loop 2	14
Working Setpoint – Loop 2	15
Working Output – Loop 2	16
Target Setpoint – Loop 2	17
Manual Output – Loop 2	18
Setpoint 1 – Loop 2	19
Setpoint 2 – Loop 2	20
Auto/Manual Mode – Loop 2	21
Proportional Band – Loop 2 working Set	22
Integral Time – Loop 2 working Set	23
Derivative Time – Loop 2 working Set	24
Cutback Low – Loop 2 working Set	25
Cutback High – Loop 2 working Set	26
Relative Cooling Gain – Loop 2 working Set	27
Process Variable – Loop 3	28
Working Setpoint – Loop 3	29
Working Output – Loop 3	30
Target Setpoint – Loop 3	31
Manual Output – Loop 3	32
Setpoint 1 – Loop 3	33

Parameter	Attribute ID
Setpoint 2 – Loop 3	34
Auto/Manual Mode – Loop 3	35
Proportional Band – Loop 3 working Set	36
Integral Time – Loop 3 working Set	37
Derivative Time – Loop 3 working Set	38
Cutback Low – Loop 3 working Set	39
Cutback High – Loop 3 working Set	40
Relative Cooling Gain – Loop 3 working Set	41
Process Variable – Loop 4	42
Working Setpoint – Loop 4	43
Working Output – Loop 4	44
Target Setpoint – Loop 4	45
Manual Output – Loop 4	46
Setpoint 1 – Loop 4	47
Setpoint 2 – Loop 4	48
Auto/Manual Mode – Loop 4	49
Proportional Band – Loop 4 working Set	50
Integral Time – Loop 4 working Set	51
Derivative Time – Loop 4 working Set	52
Cutback Low – Loop 4 working Set	53
Cutback High – Loop 4 working Set	54
Relative Cooling Gain – Loop 4 working Set	55
Process Variable – Loop 5	56
Working Setpoint – Loop 5	57
Working Output – Loop 5	58
Target Setpoint – Loop 5	59
Manual Output – Loop 5	60
Setpoint 1 – Loop 5	61
Setpoint 2 – Loop 5	62
Auto/Manual Mode – Loop 5	63
Proportional Band – Loop 5 working Set	64
Integral Time – Loop 5 working Set	65
Derivative Time – Loop 5 working Set	66
Cutback Low – Loop 5 working Set	67
Cutback High – Loop 5 working Set	68
Relative Cooling Gain – Loop 5 working Set	69
Process Variable – Loop 6	70
Working Setpoint – Loop 6	71
Working Output – Loop 6	72
Target Setpoint – Loop 6	73
Manual Output – Loop 6	74
Setpoint 1 – Loop 6	75
Setpoint 2 – Loop 6	76
Auto/Manual Mode – Loop 6	77
Proportional Band – Loop 6 working Set	78
Integral Time – Loop 6 working Set	79
Derivative Time – Loop 6 working Set	80
Cutback Low – Loop 6 working Set	81
Cutback High – Loop 6 working Set	82
Relative Cooling Gain – Loop 6 working Set	83
Process Variable – Loop 7	84
Working Setpoint – Loop 7	85
Working Output – Loop 7	86
Target Setpoint – Loop 7	87
Manual Output – Loop 7	88
Setpoint 1 – Loop 7	89
Setpoint 2 – Loop 7	90
Auto/Manual Mode – Loop 7	91
Proportional Band – Loop 7 working Set	92
Integral Time – Loop 7 working Set	93
Derivative Time – Loop 7 working Set	94
Cutback Low – Loop 7 working Set	95
Cutback High – Loop 7 working Set	96
Relative Cooling Gain – Loop 7 working Set	97
Process Variable – Loop 8	98
Working Setpoint – Loop 8	99
Working Output – Loop 8	100
Target Setpoint – Loop 8	101
Manual Output – Loop 8	102

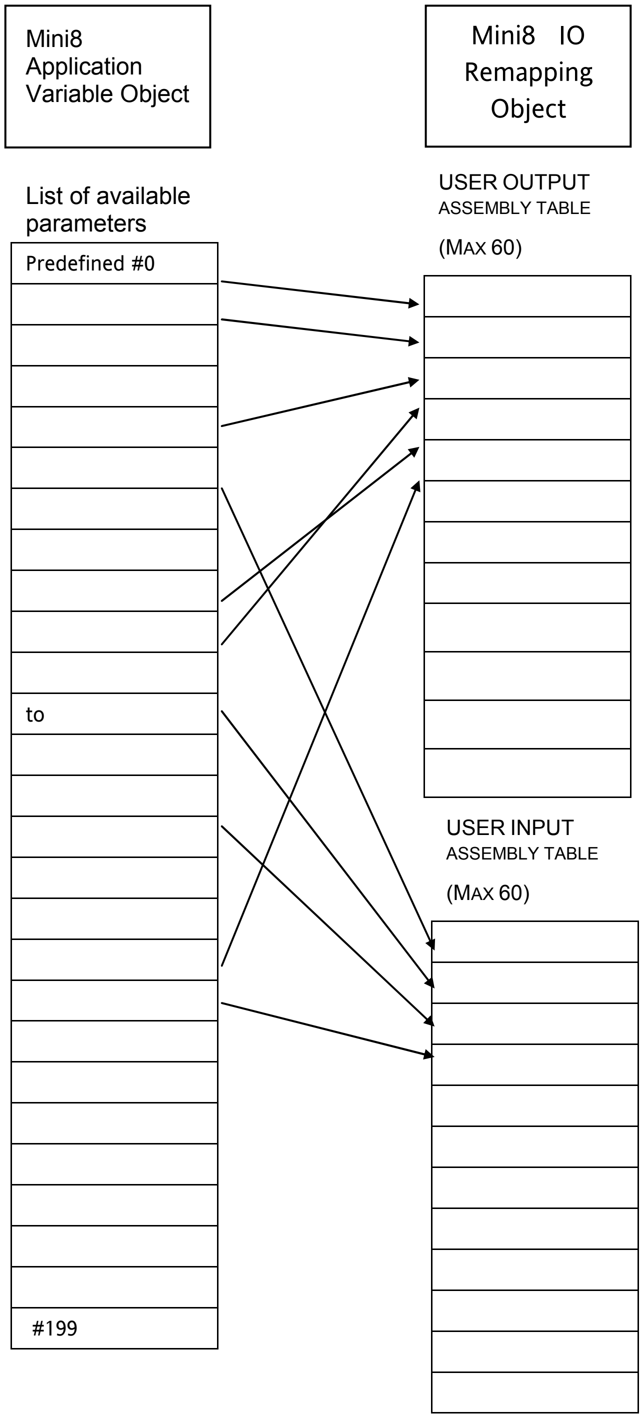
Parameter	Attribute ID
Setpoint 1 – Loop 8	103
Setpoint 2 – Loop 8	104
Auto/Manual Mode – Loop 8	105
Proportional Band – Loop 8 working Set	106
Integral Time – Loop 8 working Set	107
Derivative Time – Loop 8 working Set	108
Cutback Low – Loop 8 working Set	109
Cutback High – Loop 8 working Set	110
Relative Cooling Gain – Loop 8 working Set	111
Module PV – Channel 1	112
Module PV – Channel 2	113
Module PV – Channel 3	114
Module PV – Channel 4	115
Module PV – Channel 5	116
Module PV – Channel 6	117
Module PV – Channel 7	118
Module PV – Channel 8	119
Module PV – Channel 9	120
Module PV – Channel 10	121
Module PV – Channel 11	122
Module PV – Channel 12	123
Module PV – Channel 13	124
Module PV – Channel 14	125
Module PV – Channel 15	126
Module PV – Channel 16	127
Module PV – Channel 17	128
Module PV – Channel 18	129
Module PV – Channel 19	130
Module PV – Channel 20	131
Module PV – Channel 21	132
Module PV – Channel 22	133
Module PV – Channel 23	134
Module PV – Channel 24	135
Module PV – Channel 25	136
Module PV – Channel 26	137
Module PV – Channel 27	138
Module PV – Channel 28	139
Module PV – Channel 29	140
Module PV – Channel 30	141
Module PV – Channel 31	142
Module PV – Channel 32	143
Analogue Alarm Status 1	144
Analogue Alarm Status 2	145
Analogue Alarm Status 3	146
Analogue Alarm Status 4	147
Sensor Break Alarm Status 1	148
Sensor Break Alarm Status 2	149
Sensor Break Alarm Status 3	150
Sensor Break Alarm Status 4	151
CT Alarm Status 1	152
CT Alarm Status 2	153
CT Alarm Status 3	154
CT Alarm Status 4	155
New Alarm Output	156
Any Alarm Output	157
New CT Alarm Output	158
Reset New Alarm	159
Reset New CT Alarm	160
CT Load Current 1	161
CT Load Current 2	162
CT Load Current 3	163
CT Load Current 4	164
CT Load Current 5	165
CT Load Current 6	166
CT Load Current 7	167
CT Load Current 8	168
CT Load Status 1	169
CT Load Status 2	170
CT Load Status 3	171

Parameter	Attribute ID
CT Load Status 4	172
CT Load Status 5	173
CT Load Status 6	174
CT Load Status 7	175
CT Load Status 8	176
PSU Relay 1 Output	177
PSU Relay 2 Output	178
PSU Digital Input 1	179
PSU Digital Input 2	180
Program Run	181
Program Hold	182
Program Reset	183
Program Status	184
Current Program	185
Program Time Left	186
Segment Time Left	187
User Value 1	188
User Value 2	189
User Value 3	190
User Value 4	191
User Value 5	192
User Value 6	193
User Value 7	194
User Value 8	195
User Value 9	196
User Value 10	197
User Value 11	198
User Value 12	199

24.2.1 Table Modification

Make a list of parameters required in the input and output tables to suit the application. If the parameter is listed in the predefined list then use the attribute number of that parameter.

To set up the controller so that the required parameters are available on the network requires setting up the INPUT and OUTPUT data assembly tables with the IDs from the Application Variable Object.



25. APPENDIX C SAFETY AND EMC INFORMATION

Eurotherm Controls Ltd manufactures this controller in the UK.

Please read this section carefully before installing the controller

This controller is intended for industrial temperature and process control applications when it will meet the requirements of the European Directives on Safety and EMC. Use in other applications, or failure to observe the installation instructions of this handbook may impair safety or EMC. The installer must ensure the safety and EMC of any particular installation.

Safety

This controller complies with the European Low Voltage Directive 73/23/EEC, by the application of the safety standard EN 61010.

Electromagnetic compatibility

This controller conforms with the essential protection requirements of the EMC Directive 89/336/EEC, by the application of EMC standard EN61326

GENERAL

The information contained in this manual is subject to change without notice. While every effort has been made to ensure the accuracy of the information, your supplier shall not be held liable for errors contained herein.

Unpacking and storage

The packaging should contain an instrument and an Installation guide. It may contain a CD.

If on receipt, the packaging or the instrument are damaged, do not install the product but contact your supplier. If the instrument is to be stored before use, protect from humidity and dust in an ambient temperature range of -10^{o}C to +70^{o}C.

SERVICE AND REPAIR

This controller has no user serviceable parts. Contact your supplier for repair.

Cleaning

Do not use water or water based products to clean labels or they will become illegible. Isopropyl alcohol may be used to clean labels. A mild soap solution may be used to clean other exterior surfaces of the product.

INSTALLATION SAFETY REQUIREMENTS

Personnel

Installation must only be carried out by suitably qualified personnel.

Wiring

It is important to connect the controller in accordance with the wiring data given in this guide. Take particular care not to connect AC supplies to the low voltage sensor input or other low level inputs and outputs. Only use copper conductors for connections (except thermocouple inputs) and ensure that the wiring of installations comply with all local wiring regulations. For example in the UK use the latest version of the IEE wiring regulations, (BS7671). In the USA use NEC Class 1 wiring methods.

Power Isolation

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

Overcurrent protection

The power supply to the system should be fused appropriately to protect the cabling to the units.

Voltage rating

The maximum continuous voltage applied between any of the following terminals must not exceed 33Vac:

- relay output to logic, dc or sensor connections;
- any connection to ground.

The controller must not be wired to a three phase supply with an unearthed star connection. Under fault conditions such a supply could rise above 264Vac with respect to ground and the product would not be safe.

Conductive pollution

Electrically conductive pollution must be excluded from the cabinet in which the controller is mounted. For example, carbon dust is a form of electrically conductive pollution. To secure a suitable atmosphere, install an air filter to the air intake of the cabinet. Where condensation is likely, for example at low temperatures, include a thermostatically controlled heater in the cabinet.

This product has been designed to conform to BSEN61010 installation category II, pollution degree 2. These are defined as follows:-

Installation Category II

The rated impulse voltage for equipment on nominal 24V dc supply is 800V.

Pollution Degree 2

Normally only non conductive pollution occurs. Occasionally, however, a temporary conductivity caused by condensation shall be expected.

Grounding of the temperature sensor shield

In some installations it is common practice to replace the temperature sensor while the controller is still powered up. Under these conditions, as additional protection against electric shock, we recommend that the shield of the temperature sensor is grounded. Do not rely on grounding through the framework of the machine.

Over-Temperature Protection

When designing any control system it is essential to consider what will happen if any part of the system should fail. In temperature control applications the primary danger is that the heating will remain constantly on. Apart from spoiling the product, this could damage any process machinery being controlled, or even cause a fire.

Reasons why the heating might remain constantly on include:

- the temperature sensor becoming detached from the process
- thermocouple wiring becoming short circuit;
- the controller failing with its heating output constantly on
- an external valve or contactor sticking in the heating condition
- the controller setpoint set too high.

Where damage or injury is possible, we recommend fitting a separate over-temperature protection unit, with an independent temperature sensor, which will isolate the heating circuit.

Please note that the alarm relays within the controller will not give protection under all failure conditions.

INSTALLATION REQUIREMENTS FOR EMC

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

- For general guidance refer to EMC Installation Guide, HA025464.
- When using relay outputs it may be necessary to fit a filter suitable for suppressing the conducted emissions. The filter requirements will depend on the type of load. For typical applications we recommend Schaffner FN321 or FN612.
- If the unit is used in table top equipment which is plugged into a standard power socket, then it is likely that compliance to the commercial and light industrial emissions standard is required. In this case to meet the conducted emissions requirement, a suitable mains filter should be installed. We recommend Schaffner types FN321 and FN612.

Routing of wires

To minimise the pick-up of electrical noise, the low voltage DC connections and the sensor input wiring should be routed away from high-current power cables. Where it is impractical to do this, use shielded cables with the shield grounded at both ends. In general keep cable lengths to a minimum.

26. APPENDIX D TECHNICAL SPECIFICATION

The I/O electrical specifications are quoted as factory calibrated worst-case; for life, over full ambient temperature range and supply voltage. Any "typical" figures quoted are the expected values at 25°C ambient and 24Vdc supply.
The nominal update of all inputs and function blocks is every 110ms. However, in complex applications the Mini8 will automatically extend this time in multiples of 110ms.

26.1 Environmental Specification

Power Supply Voltage:	17.8Vdc min to 28.8Vdc max.
Supply Ripple:	2Vp-p max.
Power Consumption:	15W max.
Operating Temperature:	0 to 55°C
Storage Temperature:	-10°C to +70°C
Operating Humidity:	5% to 95% RH non-condensing
EMC:	EN61326 for Industrial Environments
Safety:	Meets EN61010, installation category II, pollution degree 2.
Max. applied voltage any terminal:	42Vpk.

The Mini8 must be mounted in a protective enclosure.

26.2 Network Communications Support

Modbus RTU: RS485, 2 x RJ45, user select switch for 3-wire or 5-wire.
Baud rates: 4800, 9600, 19200

DeviceNet: CAN, 5-pin standard "open connector" with screw terminals.
Baud rates: 125k, 250k, 500k

Modbus and DeviceNet are mutually exclusive options; refer to the Mini8 order code document.

26.3 Configuration Communications Support

Modbus RTU: 3-wire RS232, through RJ11 configuration port.
Baud rates: 4800, 9600, 19200

All versions of Mini8 support one configuration port.

The configuration port can be used simultaneously with the network link.

26.4 Fixed I/O Resources

The PSU card supports 2 independent and isolated relay contacts

Relay Output Types:	On/Off (C/O contacts, "On" closing the N/O pair)
Contact Current:	<1A (resistive loads)
Terminal Voltage:	<42Vpk
Contact Material:	Gold
Snubbers:	Snubber networks are NOT fitted.
Contact Isolation:	42Vpkmax.

The PSU card supports 2 independent and isolated logic inputs

Input Types:	Logic (24Vdc)
Input Logic 0 (off):	< 5Vdc.
Input Logic 1 (on):	> 10.8Vdc.
Input Operating Range:	-30Vdc to +30Vdc.
Input Current:	2.5mA (approx.) at 10.5V; 10mA max @ 30V supply.
Detectable Pulse Width:	110ms min.
Isolation to system:	42Vpkmax.

26.5 TC8 8-Channel TC Input Card

The TC8 supports 8 independently programmable and electrically isolated channels, catering for all standard and custom thermocouple types.

Channel Types:	TC, mV Input Range: -77mV to +77mV.
Resolution:	20 bit ($\Sigma\Delta$ converter), 1.6µV with 1.6s filter time
Temperature Coefficient:	< ±50ppm (0.005%) of reading/ °C
Cold Junction Range:	-10°C to +70°C
CJ Rejection:	> 30:1
CJ Accuracy:	± 1°C
Linearisation Types:	C, J, K, L, R, B, N, T, S, LINEAR mV, custom.
Total accuracy:	± 1°C ± 0.1% of reading (using internal CJC)
Channel PV Filter:	0.0 seconds (off) to 999.9 seconds, 1st order low-pass.
Sensor Break: AC detector:	Off, Low or High resistance trip levels.
Input Resistance :	>100 M
Input Leakage Current:	<100nA (1nA typical).
Common mode rejection:	>120dB, 47 - 63Hz
Series mode rejection:	>60dB, 47 - 63Hz
Isolation channel-channel:	42Vpkmax
Isolation to system:	42Vpkmax

26.6 DO8 8-Channel Digital Output Card

The DO8 supports 8 independently programmable channels, the output switches requiring external power supply. Each channel is current and temperature protected, holdback limiting occurring at about 100mA.

The supply line is protected to limit total card current to 200mA.

The 8 channels are isolated from the system (but not from each other). To maintain isolation it is essential to use an independent and isolated PSU.

Channel Types:	On/Off, Time Proportioned
Channel Supply (Vcs):	15Vdc to 30Vdc
Logic 1 Voltage Output:	> (Vcs - 3V) (not in power limiting)
Logic 0 Voltage Output:	< 1.2Vdc no-load, 0.9V typical
Logic 1 Current Output:	100mA max. (not in power limiting)
Min. Pulse Time:	20ms
Channel Power Limiting:	Current limiting capable of driving short-circuit load
Terminal Supply Protection:	Card supply is protected by 200mA self-healing fuse
Isolation (channel-channel):	N/A (Channels share common connections)
Isolation to system:	42Vac/dc max.

26.7 Toolkit Blocks

User Wires:	Orderable options of 30, 60 120 or 250
User values:	32 real values
2 Input Maths:	24 blocks
	Add, subtract, multiply, divide, absolute difference, maximum, minimum, hot swap, sample and hold, power, square root, Log, Ln, exponential, switch
2 Input Logic:	24 blocks
	AND, OR, XOR, latch, equal, not equal, greater than, less than, greater than or equal to, less than or equal to
8 Input Logic:	2 blocks
	AND, OR, XOR
8 Input Multiplexer:	4 blocks
	8 sets of 8 values selected by input parameter
BCD Input:	2 blocks
	2 decades (8 inputs giving 0 to 99).
Input monitor:	2 blocks
	Max, min, time above threshold
16 Point Linearisation:	2 blocks
	16-point linearisation fit
Polynomial Fit:	2 blocks
	Characterisation by Poly Fit table
Switchover:	1 block
	Smooth transition between two input values
Timer blocks:	4 blocks
	OnPulse, OnDelay, OneShot, MinOn Time
Counter blocks:	2 blocks
	Up or down, Directional flag
Totaliser blocks:	2 blocks
	Alarm at Threshold value
Real time clock:	1 block
	Day & time, 2 time based alarms

26.8 CT3 3-Channel Current-Transformer Input Card

The CT3 supports 3 independent channels designed for heater current monitoring. A scan block allows periodic test of nominated outputs to detect load (failure) changes.

Channel Types:	A (current)
Factory set accuracy:	better than ±2% of range
Current Input Range :	0mA to 50mA rms
Transformer Ratio:	10/0.05 to 1000/0.05
Input Load Burden:	1W
Isolation:	None (provided by CT)

26.9 Load Failure Detection

Requires CT3 module

Max number of loads:	16 Time Proportioned Outputs
Max loads per CT:	6 loads per CT input
Alarms:	1 in 8 Partial load failure, Over current, SSR short circuit, SSR open circuit
Commissioning:	Automatic or manual
Measurement interval:	1 sec - 60 sec

26.10 AO8 8 Channel 4-20mA Output Card

The AO8 supports 8 independently programmable and electrically isolated mA output channels for 4-20mA current-loop applications.

Channel Types:	mA (current) Output
Output Range :	0-20mA, 360Ω
Setting Accuracy:	±0.1% of reading
Resolution:	1 part in 10000 (1uA typical)
Isolation channel-channel:	42Vpkmax
Isolation to system:	42Vpkmax

26.11 PID Control Loop Blocks

Number of Loops:	0 or 8 Loops (order options)
Control modes:	On/Off, single PID, Dual channel OP
Control Outputs:	Analogue 4-20mA, Time proportioned logic,
Cooling algorithms:	Linear, water, fan, or oil
Tuning:	3 sets PID, One-shot auto-tune.
Auto manual control:	Bumpless transfer or forced manual output available
Setpoint rate limit:	Ramp in units per sec, per min or per hour.
Output rate limit:	Ramp in % change per second
Other features:	Feedforward, Input track, Sensor break OP, Loop break alarm, remote SP, 2 internal loop setpoints

26.12 Process Alarms

Number of alarms:	32 analogue, 32 digital, 32 Sensor break,
Alarm types:	Absolute high, absolute low, deviation high, deviation low, deviation band, sensor break
Alarm modes:	Latching or non-latching, blocking, time delay.

26.13 Setpoint Programmer

The Setpoint Programmer is a software orderable option

Number of programs:	50
Number of segments:	200
Number of event outputs:	8
Digital inputs:	Run, Hold, Reset, Program Advance, Skip, Segment, Sync
Power failure action:	Ramp, Reset, Continue
Servo start:	PV, SP

26.14 Recipes

Recipes are a software orderable option

Number of recipes:	8
Tags:	24 tags in total