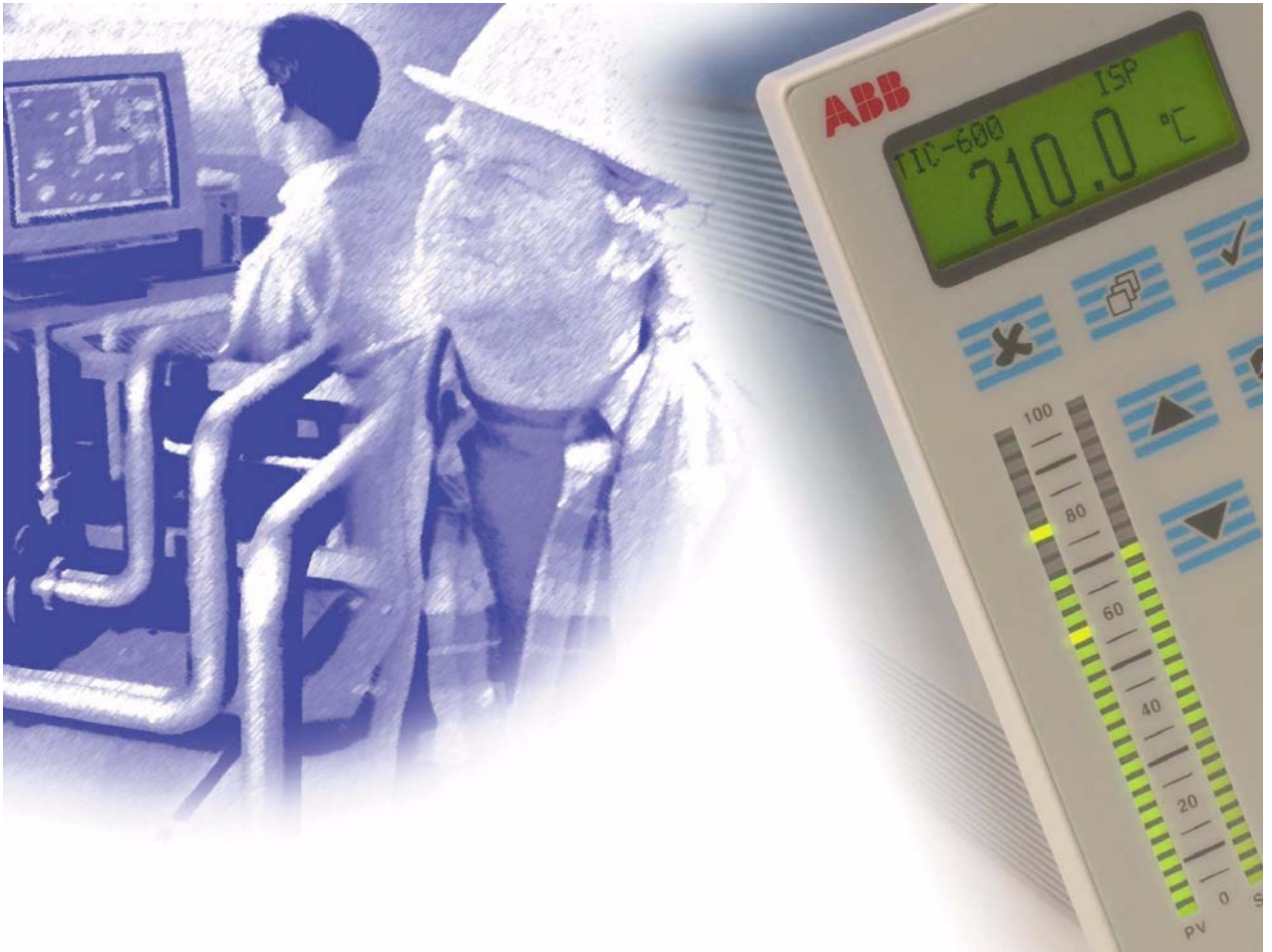


ECA06/60/600 EMA60

Configuration and Installation

Version 1.05



The ABB logo, consisting of the letters 'A', 'B', and 'B' in a bold, red, sans-serif font.

ECA06/60/600 EMA60
Configuration and
Installation

Copyright © 2000 ABB Automation Technology Products AB.

The contents of this document can be changed by ABB Automation Technology Products AB without prior notice and do not constitute any binding undertakings from ABB Automation Technology Products AB. ABB Automation Technology Products AB is not responsible under any circumstances for direct, indirect, unexpected damage or consequent damage that is caused by this document.

All rights reserved.

Release: 0230
Document version: 6-2
Document number: 493-0736-11

Printed in Sweden.

Contents

1	Introduction	11
1.1	Warning and Information Symbols	12
1.2	Process Safety	12
1.3	Electrical Supplies	13
1.4	Operating Environment	13
1.5	Version information	13
	Version 1.00	13
	Version 1.01	13
	Version 1.02	14
	Version 1.03	14
	Version 1.04	14
	Version 1.05	14
2	Installation	17
2.1	Panel Cutout	17
2.2	Installation with Cassette	19
	External Dimensions	19
	Installation and Removal	19
2.3	Installation with Mounting Plates and IP65 Gasket	23
2.4	Electrical Installation	25
	General	25
	Power Wiring	27
	I/O Signal Wiring	31
	Communications signal wiring, RS485	38
	Communications signal wiring, RS232	38
	Connections in an Electrically Noisy Environment	38
3	Configuration	43
3.1	Configuring the Process Value	43
	Configuration Trees	44
	Parameters to Configure	44
	Configuration Step-by-Step	44
3.2	Controls and Indicators	49
	LCD Display	50
	PV Bar Graph Display	50

	SP Bar Graph Display	50
	Output Bar Graph Display	51
	Control Keys	51
3.3	Entering Configuration Mode	53
	No Password Configured	53
	Password Configured	53
3.4	Exiting Configuration Mode	55
3.5	Lost Code Retrieval	55
3.6	Setting the Configuration	56
	Introduction	56
	Navigating the Configuration Menu	56
	Menu Abbreviations	58
	Naming Blocks	60
	Internal Connections	60
	Overview	62
	Input Function Blocks	74
	Arithmetic Function Blocks	74
	Logic Function Blocks	82
	Other Functions	88
	Set Point Block	116
	Alarms	132
	Control	136
	Output Configuration	155
	Output Functions	166
	Outputs	176
	ECA Operator Block	184
	EMA60 Operator Block	190
	System Block	193
3.7	Status Signals	198
3.8	Test Mode	198
4	Maintenance	199
	4.1 General	199
	4.2 Annual Check	199
5	Technical Data	201
	5.1 Type	201
	5.2 Control Functions	201
	5.3 Control Action	201
	5.4 System Functions	201
	Operating Modes	201
	Process Value	202
	Set Point	202
	Output Signals	202
	Alarms	202

Computer Interface	203
Arithmetic Functions	203
Logic Functions	203
Other Functions	203
Miscellaneous	204
5.5 Input Signals	204
5.6 Output Signals	205
5.7 Computer Communication	205
5.8 Safety Functions	205
5.9 Indications	205
5.10 Mains Supply	206
5.11 Transmitter Power Supply	207
5.12 Environment	207
5.13 Installation	207
5.14 Electrical Connections	207
5.15 Relevant Standards	207
5.16 Physical Characteristics	208
A Application Examples	209
A.1 Level Control with Feed Forward and Gain Scheduling	210
Description	210
Mechanical Installation	210
Definition of Control Problem	210
Preset Configurations	211
Selection of I/O	211
Electrical Installation	212
Configuration	214
Startup	219
A.2 Temperature Control and Communication with COMLI	220
Description	220
Definition of the Control Problem	220
Mechanical Installation	221
Preset Configurations	221
Selection of I/O	221
Electrical Installation	221
Configuration	223
Startup	226
A.3 Furnace Pressure Control with 3-State Pulse Output	227
Description	227
Definition of Control Problem	227
Mechanical Installation	227
Preset Configuration	227
Selection of I/O	228
Electrical Installation	228
Configuration	230

Startup	238
B Error, Warning and Information Messages	239
B.1 Message Types	239
System Errors, Errors and Warnings	239
Information Messages	240
B.2 Controller Reaction	240
B.3 Acknowledging Messages	240
B.4 Error and Warning Descriptions	241
HW	241
FLASH Write	241
FLASH Erase	241
FLASH Checksum	241
RAM	241
Sample Time Error	242
AIx Int/AOx Int	242
DO Overload	242
Calibration	242
Arithmetic Error	243
CLPM Osc LoopX	243
XX Source off	243
Tuner Abort LoopX	243
Tuner PVNoise LoopX	243
Adapt Warn LoopX	243
B.5 Information Messages	243
Alarm Messages	243
Tuning	243
B.6 Summary of Messages	244
System Error, Error and Warning Messages	245
Information Messages	247
C Configuration Tree Diagrams	249
C.1 ECA06 Configuration Tree	250
C.2 ECA60 Configuration Tree	254
C.3 ECA600 Configuration Tree	261
C.4 EMA60 Configuration Tree	273
D Configuration Work Sheets	281
ECA06, Default	282
ECA60, Default	283
ECA600, Default	284
EMA60 Default/Triple Ind	285
E Communication	287
E.1 Communication setup	287

E.2	Protocol Description	288
	Modbus Communication	288
	Comli Communication	289
E.3	Communication Facilities	292
	Changing Communication Mode	294
	Changing Control Modes	294
	Changing Loops	295
E.4	Front Panel Usage	295
E.5	Communication Area Mapping	295
	I/O RAM Map	295
	Register Map	297
	Writing K	300
	Writing TI and TD	301
F	Special Modes	303
	F.1 Total Reset	303
	Index	305

Chapter 1

Introduction

This Configuration and Installation Manual provides all the information required by trained installation and maintenance personnel to install the ECA06, ECA60, ECA600 and EMA60 process controllers and to configure them for use by system operators.

Chapter 2, Installation, describes the mechanical and electrical installation procedures for the controllers.

Chapter 3, Configuration, is intended for trained personnel such as Process Control Engineers or Instrument Technicians who configure the controllers. It describes the operator interfaces and features available in the Configuration Mode and describes all the configurable options available for each controller.

Chapter 4, Maintenance, contains maintenance instructions to ensure the continued safe operation of the controllers.

Chapter 5, Technical Data, contains a summary of the features of the controllers.

Appendix A, Application Examples contains detailed configuration instructions for some typical applications.

Appendix B, Error, Warning and Information Codes, lists all the error, warning and information codes that may be encountered and describes the appropriate operator action by maintenance personnel.

Appendix C, Configuration Tree Diagrams, contains complete Configuration Menus for each controller.

Appendix D, Configuration Work Sheet, contains an example worksheet which you can photocopy and use to plan configurations.

Appendix E, Communication, contains details of communication parameters and settings.

Appendix F, Special Modes, contains details of operating modes where you can update the software or reset all the parameters in the configuration to their default settings

- Installation and maintenance personnel should read chapters 3, 4 and 5 of this manual.
- Technicians who are authorised to determine the installation functions and configuration should read Chapter 2 of this manual.

Autotuner patents:

Sweden 8104989-2, Canada 1 205 511, Finland 71435, USA 4 549 123, Denmark 159343, Norway 160632.

1.1 Warning and Information Symbols

Warning and information symbols appear in the left margin adjacent to paragraphs containing important information.

The symbols should be interpreted as follows:



- A warning sign denotes a *danger of injury to personnel* or that caution is required to *avoid damage to the equipment*.



- A sign for *mandatory action* denotes steps necessary to bring the equipment into operation but without immediate safety implications.



- A sign indicating *important information*.

1.2 Process Safety



Read the safety and environmental instructions provided by the supplier of the machinery or process equipment controlled by the ECA controller. Pay special attention to situations in which the mains supply is switched on or off, units are fitted or removed, cables are connected or disconnected, and units are reset or

switched to manual operation. If possible shut down the process before such actions are taken.

1.3 Electrical Supplies



Maintenance of electrical equipment should only be performed by trained personnel. Always switch off the power supply at the mains outlet and disconnect the power supply cables before removing the covers from or attempting to repair any ECA controller.

1.4 Operating Environment



Observe the following:

- The ECA controllers must not be exposed to environmental conditions exceeding the values stated in the technical data in the Configuration and Installation Manual.
- The ECA controllers should not be used in an environment where there is a risk of exposure to strong electrical interference.
- Always follow precautions for handling electrostatic sensitive devices when handling the ECA controllers.

1.5 Version information

Version 1.00

The first version.

- Controller types ECA06, ECA60 and ECA600

Version 1.01

Changes and additions:

- Absolute and Deviation Alarm block; The input connections are freely configurable.
- Auto Mode; No parameters except PID Parameters are changeable.
- New block types; AOC (Analog Output Communication) and DOC (Digital Output Communication).
- Editable texts.

- Configuration via COMLI possible.

Version 1.02

Changes and additions:

- EMA60 is serial approved.
- German menu language is introduced.

Version 1.03

Changes and additions:

- DOX 6 can be used in this version.

Version 1.04

Changes and additions:

- The integrator block is revised completely.
- The integrator block is included in ECA600.
- PID- and Gain Scheduling parameters can be read/written via COMLI.
- A new parameter, COMLIWriteEnable, enables writing via COMLI in Communication Mode Local.
- The parameter ESPLock is added to the Set Point Block, which means that the controller can be locked in ESP mode.
- The “RestartBehaviour” parameter is supplemented with new modes.
- The IncDec Block has improved functions. Two input signals for tracking and two parameters for output signal limitation are added.

Version 1.05

Changes and additions:

- The Rate Limiter function is now also available in ECA60 and ECA600.
- The Time Measurement function is now also available in ECA600.
- Two new function blocks (FO), Timer and Counter, have been added to ECA600 and EMA60.
- A warning does not cause the bar graph to flash, only an error.

- The Three State Pulse function (with feedback) has been revised.
- ISP and output signals are now writeable from a supervisory system via communication.
- Support for the communication protocol MODBUS RTU has been added.
- A Stiction Compensation function has been added to the analog outputs.
- Three decimals are now available in the arithmetic blocks.
- Start/Abort Tuning and Alarm acknowledge are now possible from a supervisory system via communication.

Chapter 2

Installation

This chapter provides all the information required by trained service engineers to install the ECA06, ECA60, ECA600 and EMA60 controllers and to make all the necessary electrical connections.

The ECA06, ECA60, ECA600 and EMA60 controller units fit into a cutout in a vertical panel which allows high packing density. The panel must be between 2 mm and 4 mm thick. For installation on thicker panels, contact your local representative.



Warning – Damage From ESD: To reduce the risk of damage to the controller from ESD (electrostatic discharge) you should discharge yourself by touching a suitable ground point before handling the controller.

2.1 Panel Cutout

The dimensions and spacing of the panel cutouts required for installation of the controller are shown below. All dimensions are in mm.



Note: The panel may need extra support if a large number of controllers are mounted on it and the minimum spacing between the panel cutouts is used.

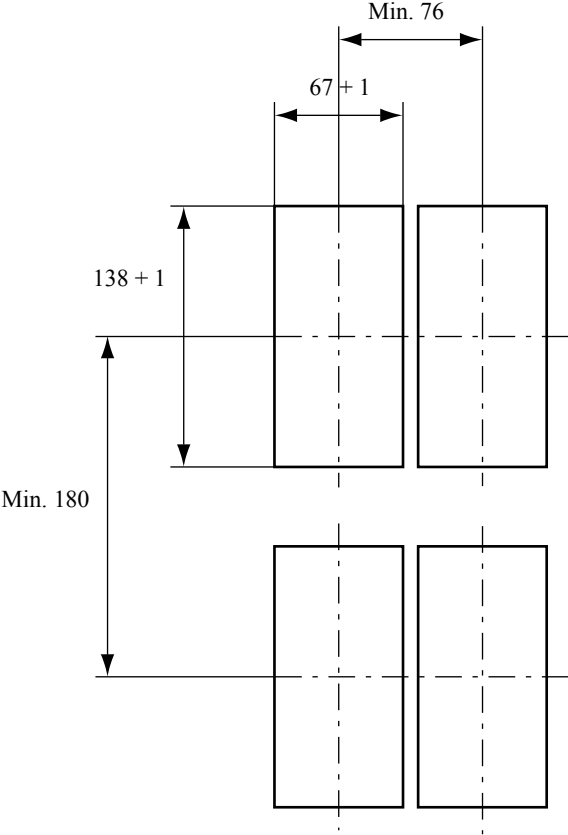


Figure 1 Panel Cutouts.

2.2 Installation with Cassette

External Dimensions

The external dimensions of the controller are shown below

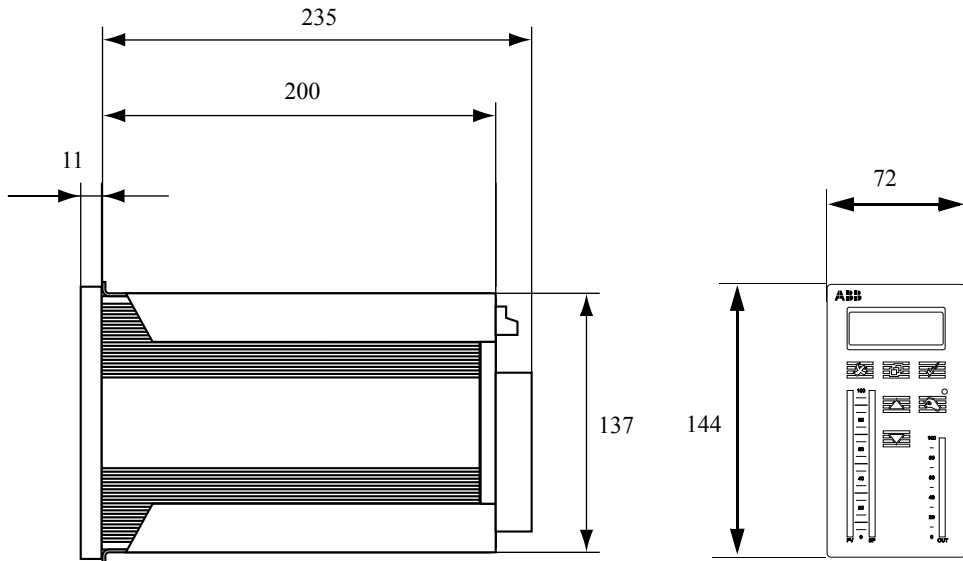


Figure 2 External Dimensions.

Installation and Removal

The mechanical installation is performed in two steps:

- 1 Installation of the cassette in the panel cutout.
- 2 Insertion of the controller unit in the cassette.

The electrical connections can be made after step 1.

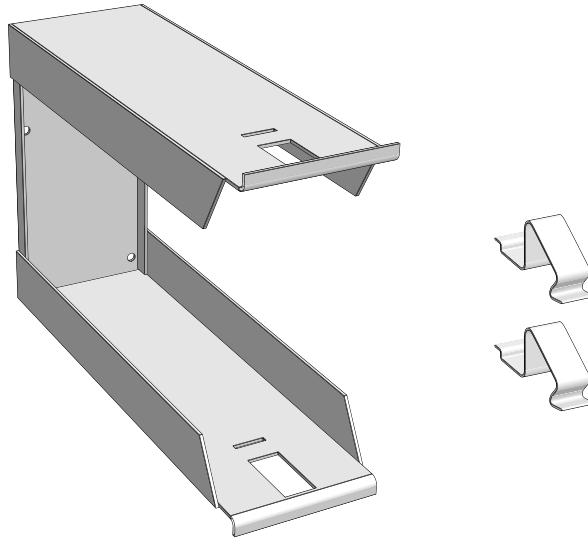


Figure 3 Cassette and Springs.

Installation of the Cassette

- 1 Push the empty cassette into the panel cutout and let it hang down by its own weight with the upper edge pressing against the upper edge of the cutout.

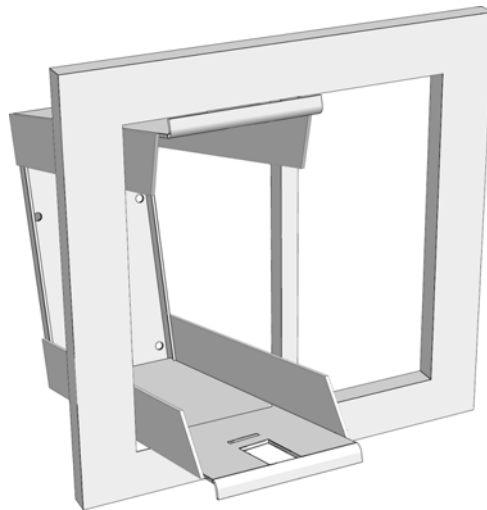


Figure 4 Empty Cassette Hanging in the Cutout.

- 2 Insert the lower retaining spring as shown in the figure below, holding the cassette with your thumb and pressing the spring down with your index finger or a screwdriver if necessary.
- 3 Repeat step 2 for the upper retaining spring.

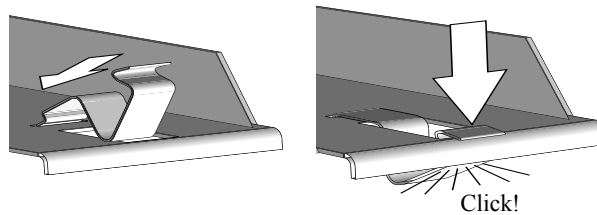


Figure 5 Mounting the Retaining Springs.

Removing the Cassette

Removal is the reverse of installation. Proceed as follows:

- 1 Press the front part of the lower retaining spring back. If the mounting panel is thick it may be necessary to use a screwdriver to do this.
- 2 Lever the retaining spring out.
- 3 Repeat steps 1 and 2 for the upper retaining spring.
- 4 Withdraw the cassette from the panel cutout.



Figure 6 Removing the Controller Unit.

Inserting the Controller Unit

To insert the controller unit in the cassette proceed as follows:

- 1 Gently slide the controller into the cassette unit as far as it will go.
- 2 Press gently on the front until you can hear a distinct click. Take care not to press on the LCD display window or the push buttons.
- 3 Check the mounting force by trying to pull out the controller by hand.

Removing the Controller Unit

To remove the controller unit from the cassette proceed as follows:

- 1 Place the tip of a broad screwdriver (6 – 8 mm) into the small opening on the upper or lower part of the front frame as shown in Figure 6.
- 2 Gently twist the screwdriver until you hear a distinct click.
- 3 Slide the controller out of the cassette.

2.3 Installation with Mounting Plates and IP65 Gasket

To install the unit with mounting plates you need two mounting plates and four fixing nuts as shown in the figure below. For IP65 compliance you also need the gasket.

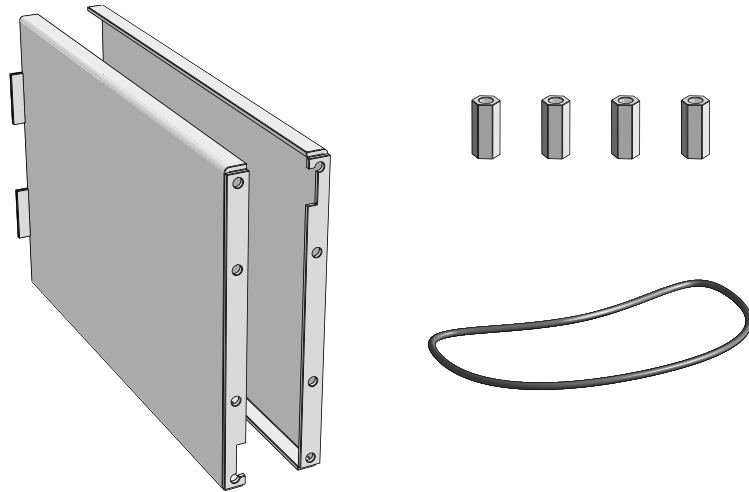


Figure 7 Mounting Plates, Fixing Nuts and IP65 Gasket.

Installation

To install the controller with the mounting plates and IP65 gasket proceed as follows:

- 1 Remove the two mounting plates from the controller unit by removing the four fixing nuts at the rear of the controller.
- 2 If you require IP65 sealing, mount the IP65 gasket by rolling it forward from the rear of the controller to the front frame. Check that the gasket is correctly installed in groove at the front of the frame.
- 3 Insert the controller into the panel cutout from the front.
- 4 Remount the mounting plates and secure in place by tightening the four nuts at the rear of the controller.

Removal

Removal is the reverse of installation.

Note: To ensure IP65 sealing the IP65 gasket must be exchanged for a new one if the unit is reinstalled.

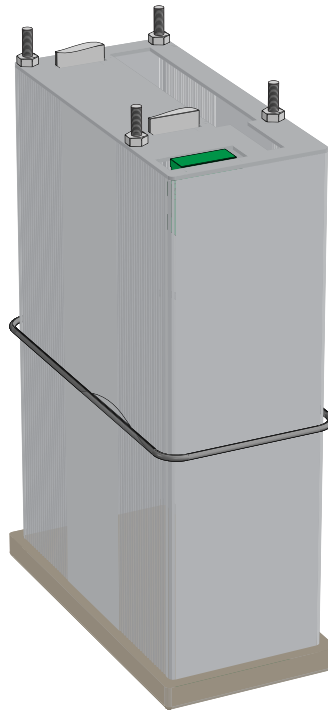


Figure 8 Mounting the Gasket.

2.4 Electrical Installation



Warning – Fuses: To prevent the risk of fire or injury to personnel in the event of a short-circuit the a.c. mains supply must be fitted with a 6 A slow blow fuse. No more than five units should be connected to the same supply. If more controllers are required then each set of five should be connected to a separate, fused mains supply.

General

The electrical installation is performed in four steps:

- 1 Configuring the jumper module.
- 2 Signal connections.
- 3 Connection of data communications link (if applicable).
- 4 Connection of the power supply cabling.

Installation of the controller in an electrically noisy environment is described at the end of this section. The electrical connections can be made either before or after the cassette is mounted in the panel. The terminal blocks at the rear of the cassette are shown in Figure 9.

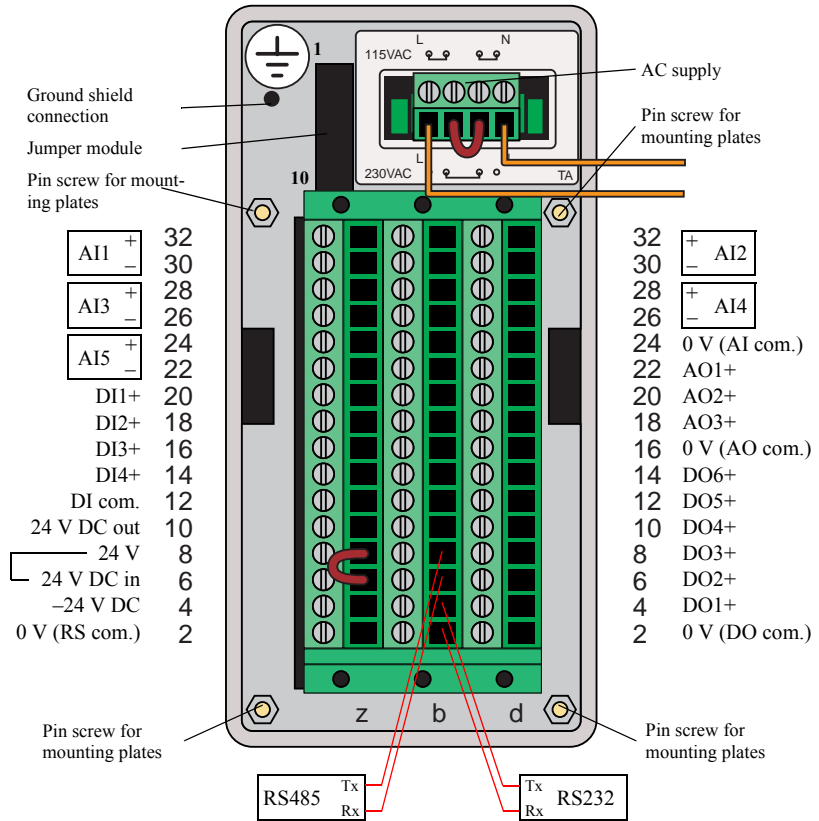


Figure 9 Cassette Terminal Blocks.

Power Wiring

The controller can operate from a direct current supply or an alternating current supply.

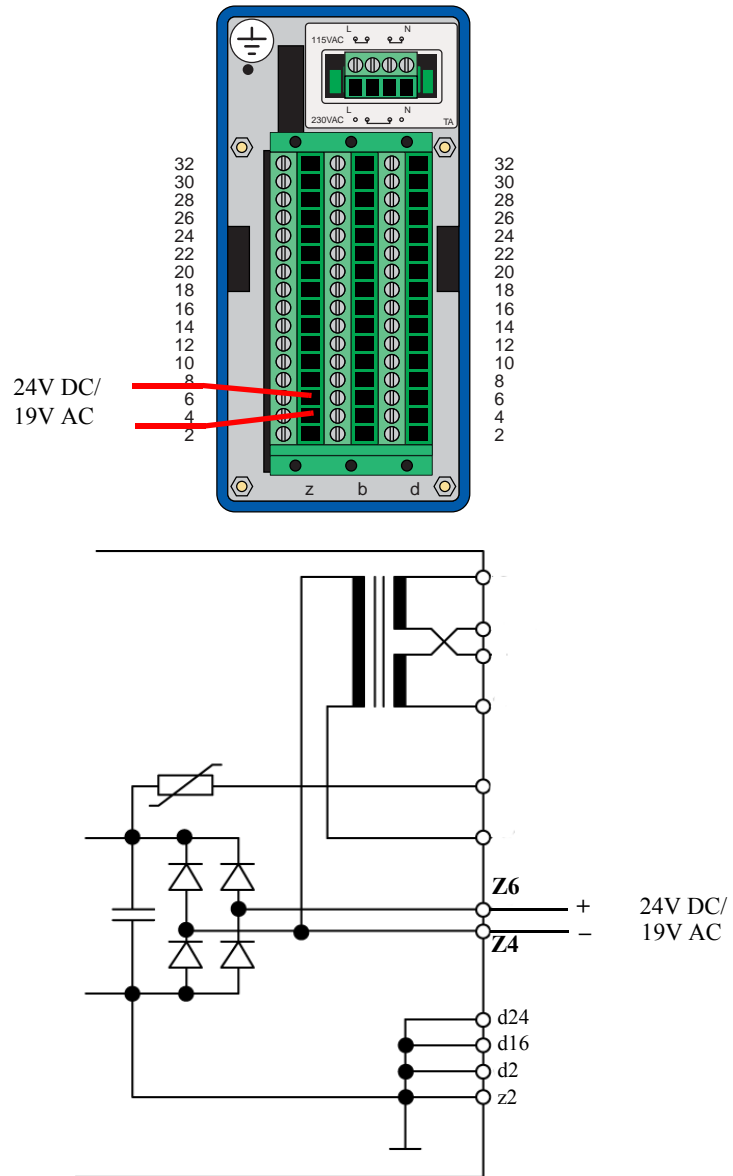


Figure 10 D.C. Power Supply Connections.

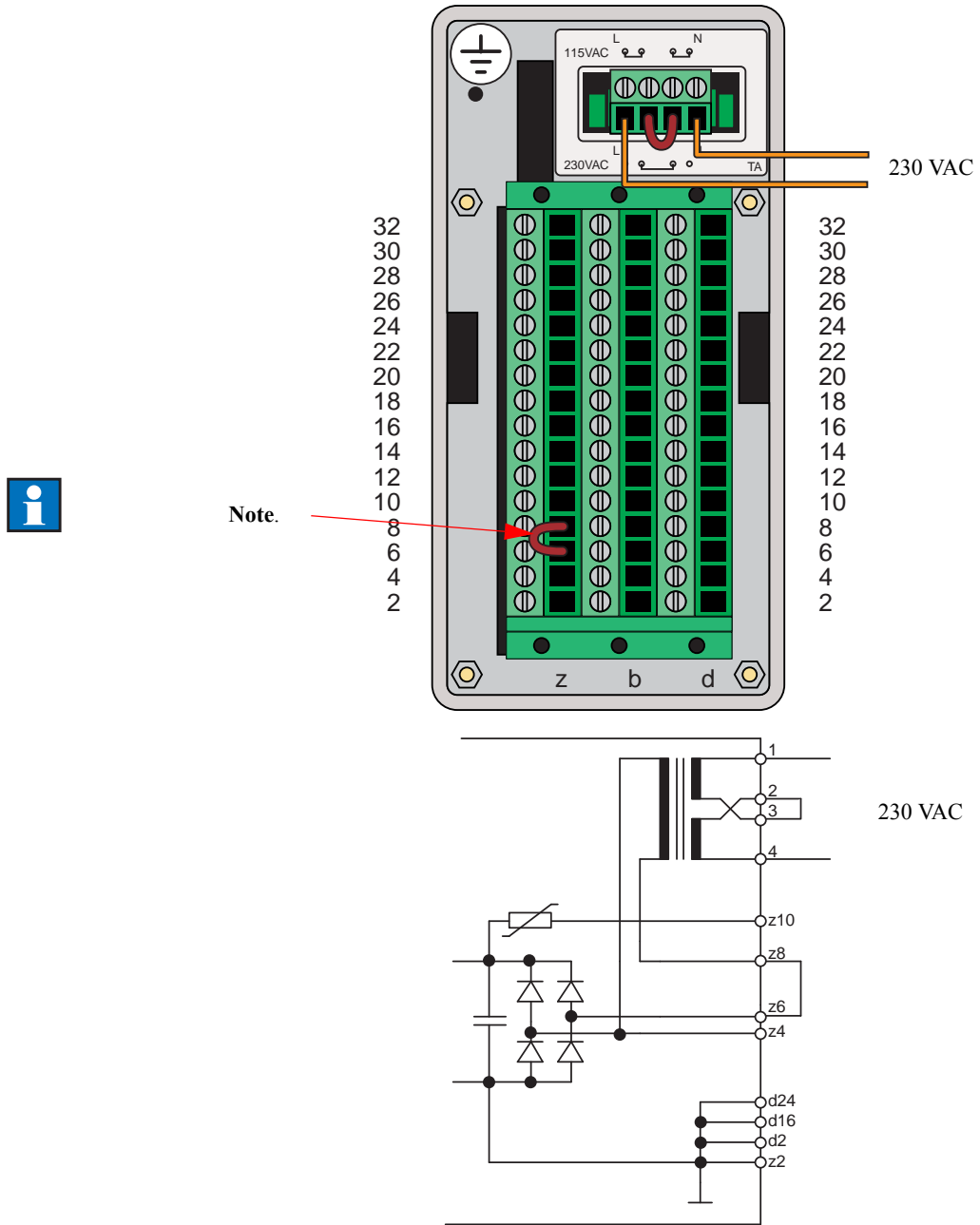


Figure 11 230 V AC Power Input Connections.



Note.

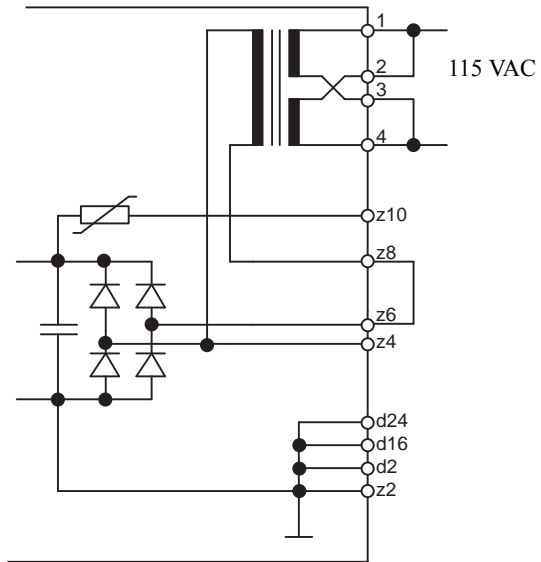
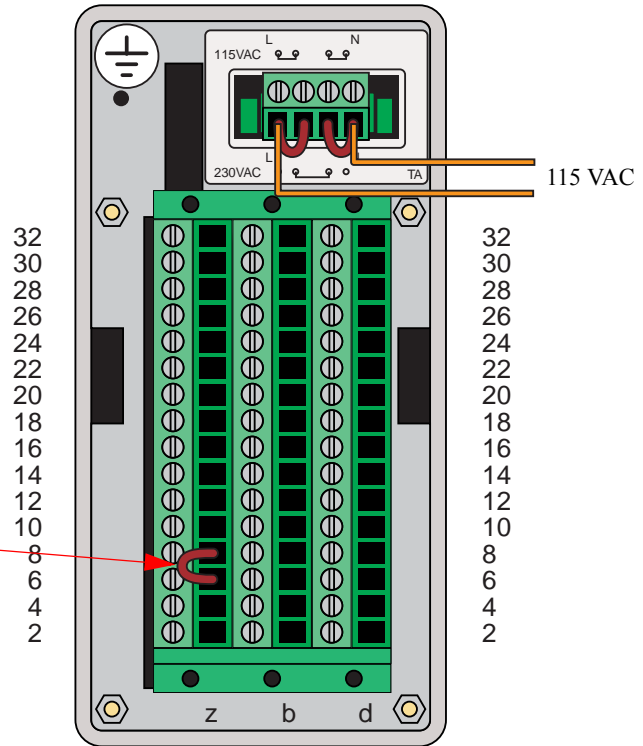


Figure 12 115 V AC Power Input Connections.

Jumper	Open	Closed	Analog Input
1	Differential	Single Ended	1
2	Voltage	Current	1
3	Differential	Single Ended	2
4	Voltage	Current	2
5	Differential	Single Ended	3
6	Voltage	Current	3
7	Differential	Single Ended	4
8	Voltage	Current	4
9	Differential	Single Ended	5
10	Voltage	Current	5

Table 1 Jumper Module Connections.

A.C. Supply

The controller can operate from 115 V, or 230 V, 50/60 Hz, 19 VA maximum power.



Note: Three wire cabling must always be used for connection of the a.c. mains supply. The supply terminals at the rear of the cassette can accommodate wire of maximum size 1.5 mm².

To connect the mains supply proceed as follows:

- 1 Switch the mains supply Off and disconnect the mains supply cable from the mains outlet.



Warning – Lethal Shock Hazard: Always ensure that the mains supply cable is disconnected from the mains outlet before connecting it to the controller otherwise a lethal shock hazard exists.

- 2 Connect jumpers and cable to the supply terminals at the rear of the cassette as follows (see also Figure 9):

For 230 V AC operation connect a jumper between terminals 2 and 3.

For 115 V AC operation connect jumpers between terminals 1 and 2 and between terminals 3 and 4.

Supply ground to ground terminal.
Supply neutral to terminal 4.
Supply live to terminal 1.



Note: The ground connection must always be connected to ground when the controller is connected to the mains supply.

When an AC supply is used a +24 V supply is available from terminals Z10 (+24 V) and Z2 (0 V) to supply external units, e.g. transmitters. The maximum loading on these terminals is 150 mA.

D.C. Supply

The controller can operate from a +24 V DC power supply, maximum power consumption 15 W.



Notes:

- The supply terminals at the rear of the cassette can accommodate wire of maximum size 1.5 mm².
- It is strongly recommended that the d.c. supply should be fused appropriately.

To connect the d.c. supply proceed as follows:

- 1 Switch the d.c. supply Off and disconnect the d.c. supply cable from the d.c. outlet.
- 2 Connect the cable to the supply terminals at the rear of the cassette as follows:

Supply positive to terminal Z6.
Supply negative to terminal Z4.



Note: Remove the jumper between Z8 and Z6.

I/O Signal Wiring

All the input and output signals are connected to the terminal block. The maximum wire size which can be connected to the terminal block is 1.5 mm².

Analog Inputs

AI1 to AI5 have their positive and negative poles connected as shown in Figure 13. Cable screens must be connected to the chassis ground using lug connectors. Connection to the chassis ground is available at the four screws at the rear panel.

Figures 14 to 17 give examples of how to connect different types of analog input to the controller.



Note: For voltage input the appropriate jumper must be removed.

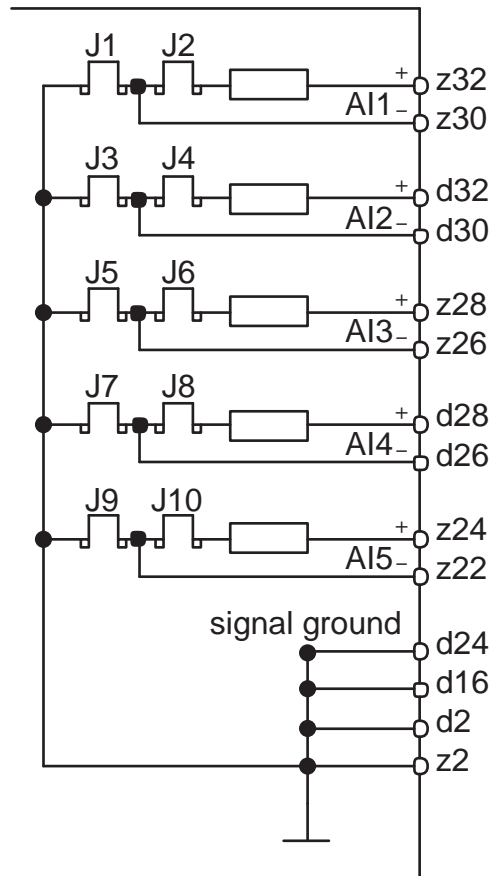


Figure 13 I/O Signal Wiring.

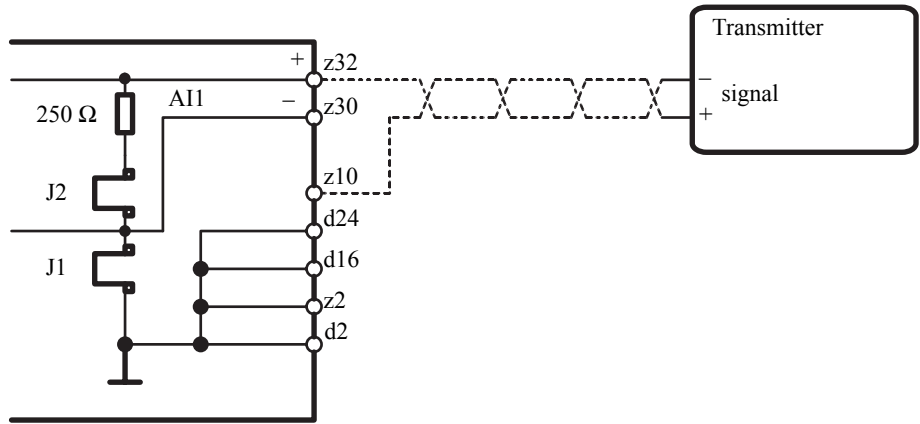


Figure 14 Two-wire connection with 24 v supply from the controller.

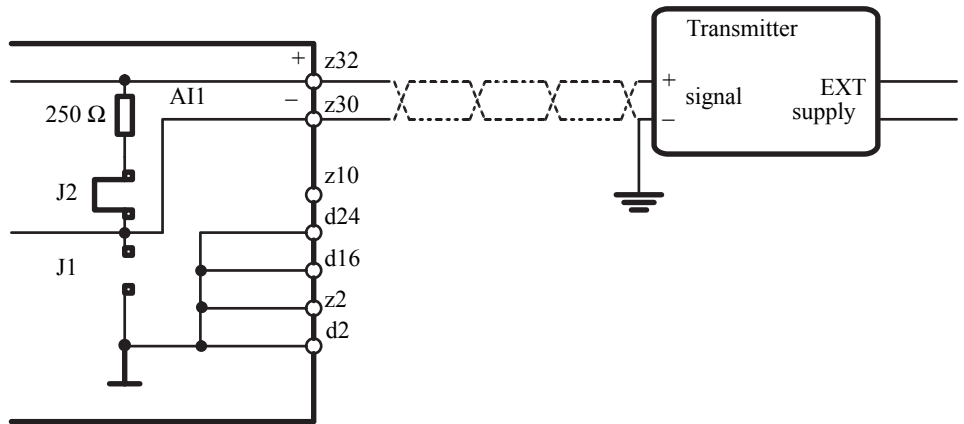


Figure 15 Separately supplied sensor with output signal electrically connected to ground.

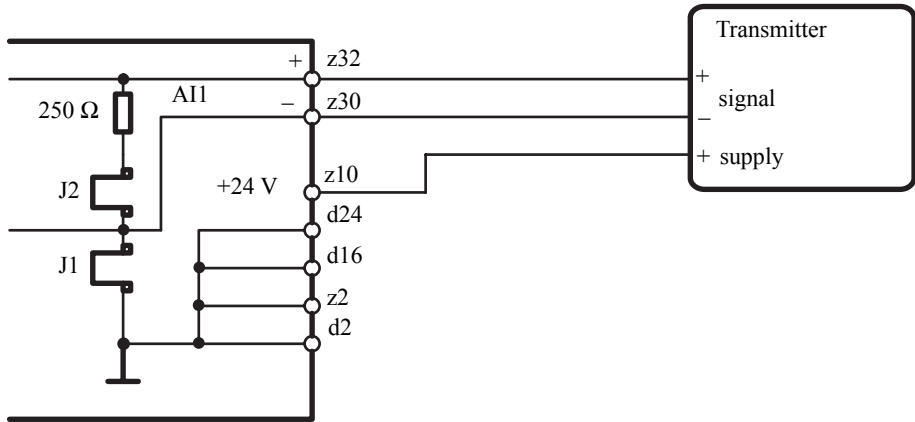


Figure 16 Three-wire connection with 24 V supply from the controller.

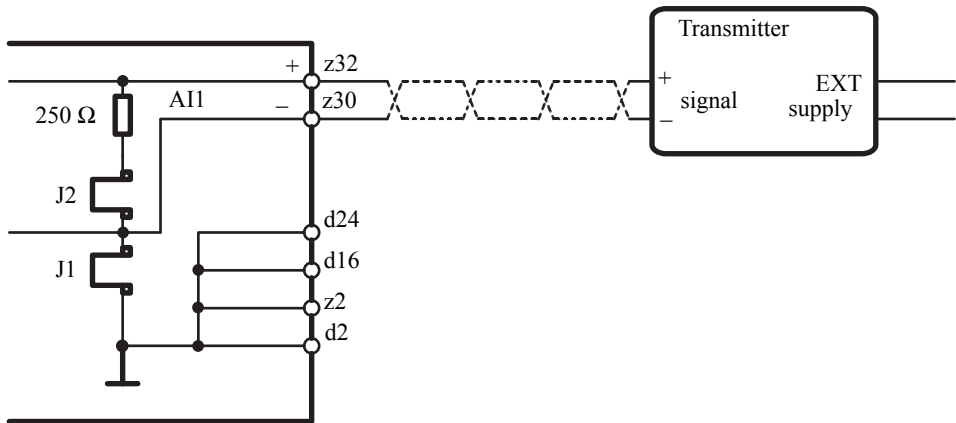


Figure 17 Separately supplied sensor with output signal electrically isolated from ground.

If an actuator potentiometer is used with a three-state pulse output it should be connected as shown in Figure 18. In this case the potentiometer is fed from AO2. The voltage type input has a fixed reference.

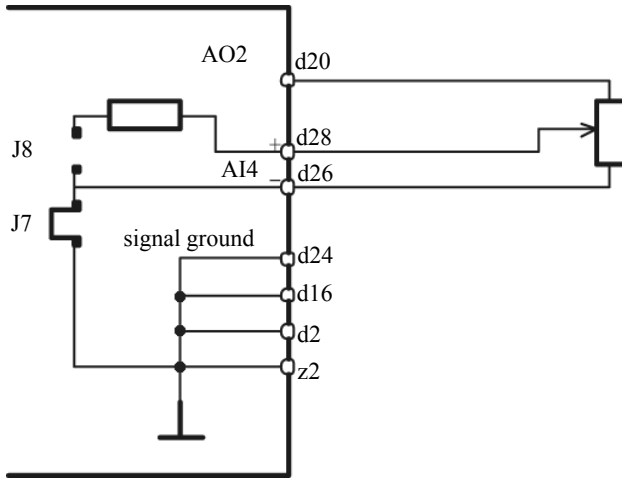


Figure 18 Actuator Potentiometer.

If an analog output is not used the output terminals should be short-circuited, otherwise there is a risk of initiating an error message for open-circuit output.

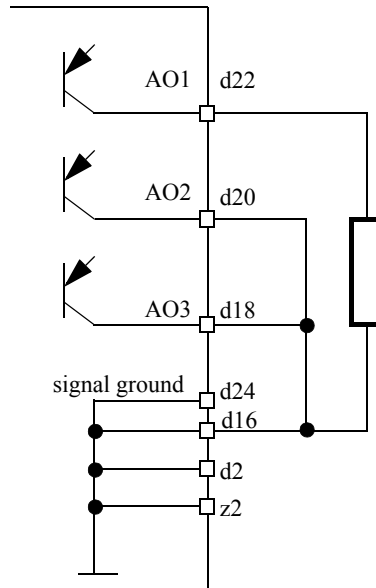


Figure 19 AO1 used / AO2 and AO3 not used.

Digital Inputs

All digital input signals are connected via opto-couplers. The supply may be taken from the controller or from an external current source.

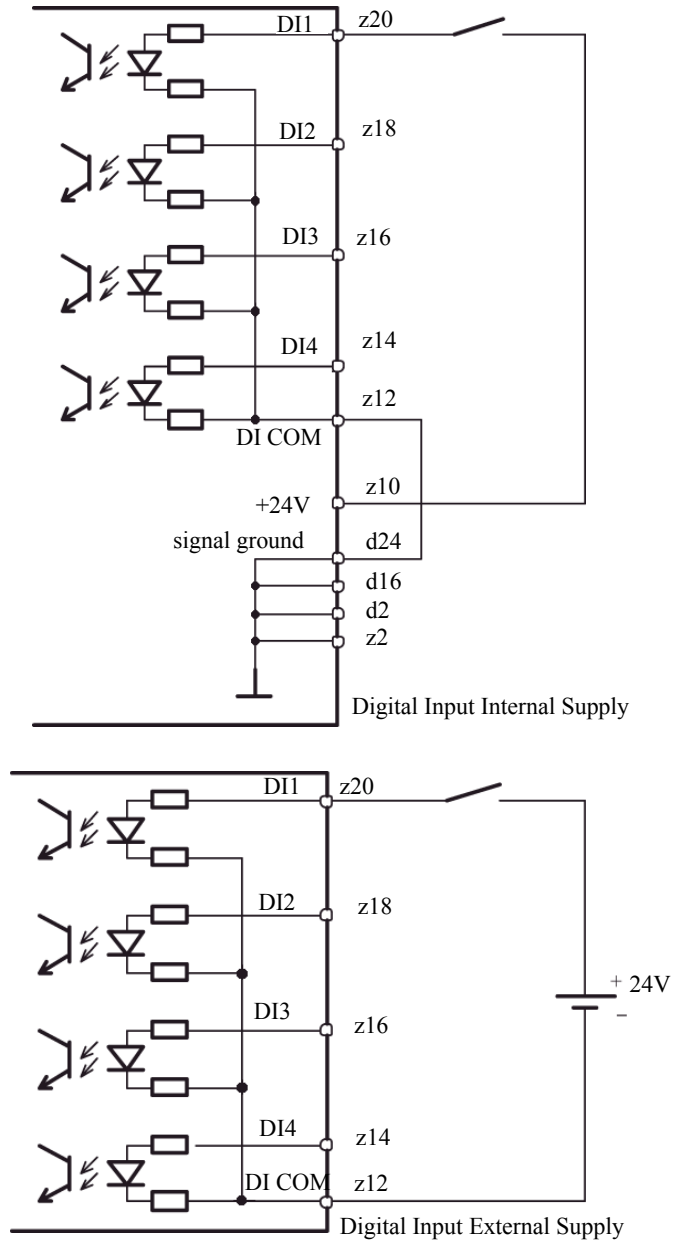


Figure 20 Digital Input Internal and External Supplies.



Digital Outputs

Note: If an inductive load, e.g. a relay coil, is connected to a digital output the coil must be shunted by a diode, such as a 1N4003 or similar. The diode should be wired as close as possible to the coil.

DO1 to DO6 are open-emitter current source transistor outputs. The maximum current is limited to a maximum of 250 mA per output, total maximum load 500mA. The outputs are short circuit protected.

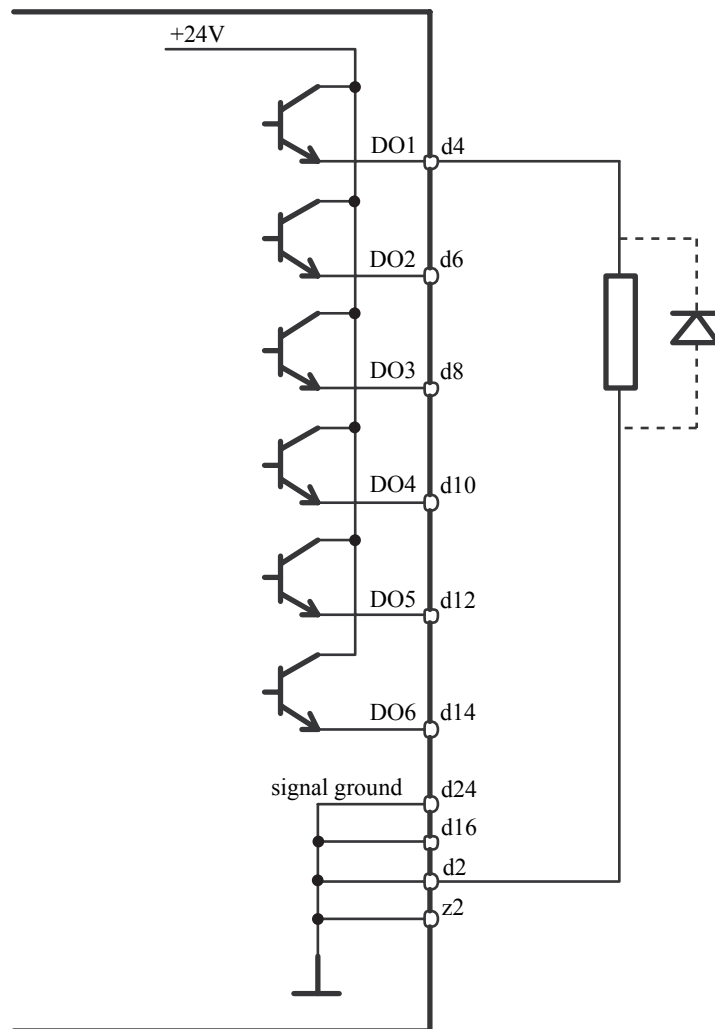


Figure 21 DO1 to DO6 Internal Supply.

Communications signal wiring, RS485

For serial communications (RS485) screened twisted-pair cable must be used with the screen grounded at each device. RS485+ should be connected to terminal b8 and RS485- should be connected to terminal b6. The last device on the data bus must be terminated by a 120 Ω impedance. This is achieved by connecting a 120 Ω resistor between terminals b6 and b8.

Communications signal wiring, RS232

For serial communications (RS232) screened twisted-pair cable must be used with the screen grounded at each device. RS232Tx should be connected to terminal b4 and RS232Rx should be connected to terminal b2.

Connections in an Electrically Noisy Environment

When making connections in an electrically noisy environment the controller should be connected to a phase which suffers the minimum of interference, i.e. contactors or thyristor controlled equipment should not be used on the same line. If this is impossible an isolating transformer or a magnetic voltage stabilizer should be installed. This type of equipment can of course be used to supply a number of controllers, however you must ensure that no electrically noisy equipment is connected to the secondary side – the same side as the controller. Follow the manufacturer's recommendations concerning grounding of this type of transformer.

Screening

The input and output wiring to field mounted equipment, e.g. sensors and input converters, is connected to the controller using screened twisted pair cable.

If the connections are made via an intermediate terminal block in an enclosure the screen should be connected to the grounding terminal with as short a length of free screen as possible, preferably less than 10 cm.



Note: In cases where analog and digital cables share the same cable runs within an enclosure, the digital cables should also be screened. The screening should be grounded at the controller chassis.

All screening on incoming field cables directly connected to the controller must be grounded at the controller chassis. When separate cable runs are used twisted pair cables will suffice. In interference free conditions single wires can be used. However signal wiring should not be run together with power cabling unless the signal wiring is shielded.

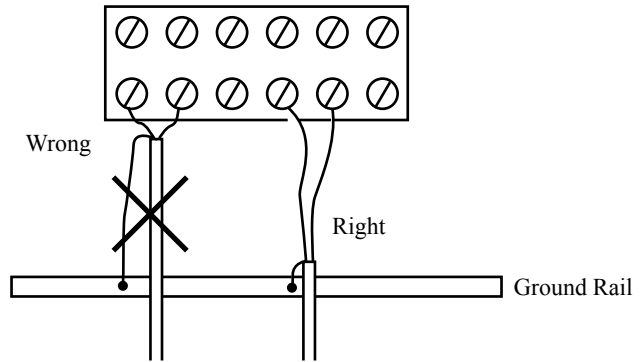


Figure 22 Screening Connections.

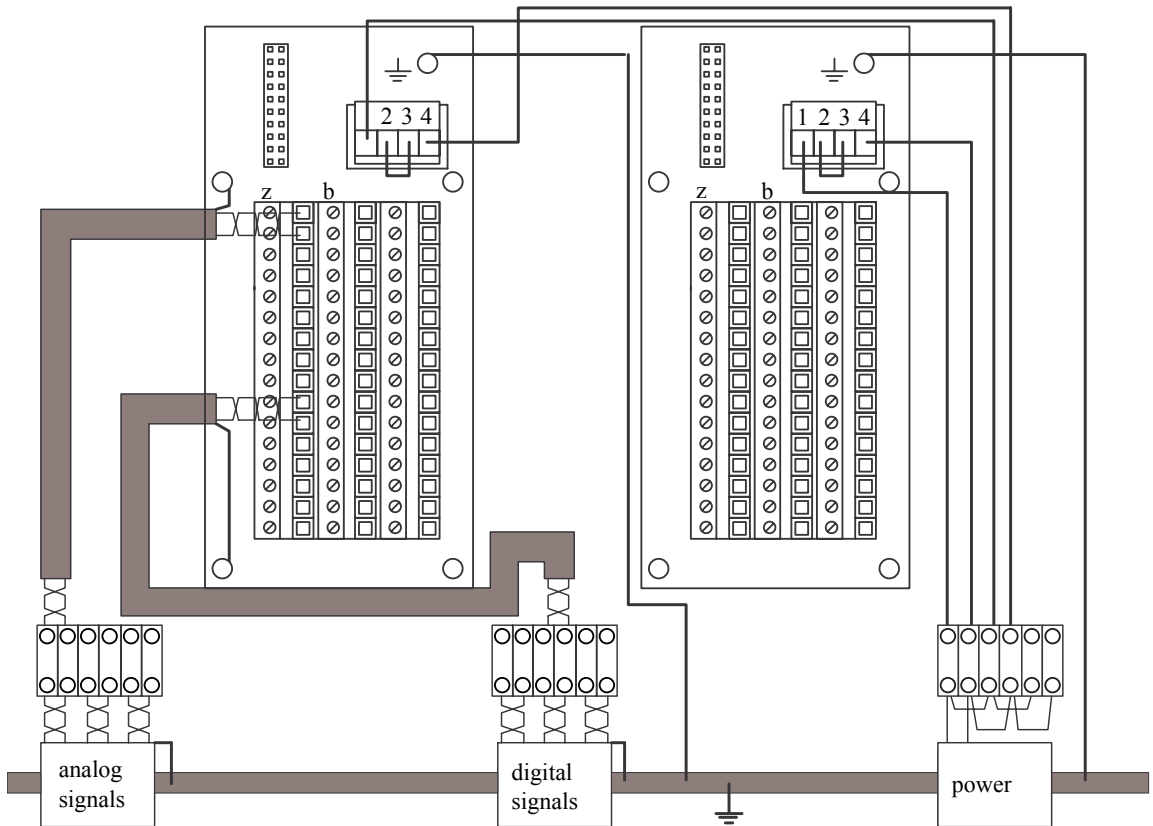


Figure 23 Controller Connections in an Enclosure with an Intermediate Terminal Block.

Grounding and Common Mode Voltage (CMV)

Ground loops can arise when the return wire or the sensor are grounded locally at their point of installation and the analog inputs have fixed reference grounds. The inputs should then be connected as differential inputs, i.e. with floating ground references so that the inputs are floating with respect to the signal ground. Such an input can receive a CMV within the range -5 V to $+17\text{ V}$ measured from the input negative pole to the supply ground. If it is anticipated that the CMV will be outside this range then an isolation amplifier must be used. An example of a ground loop is shown in Figure 24.

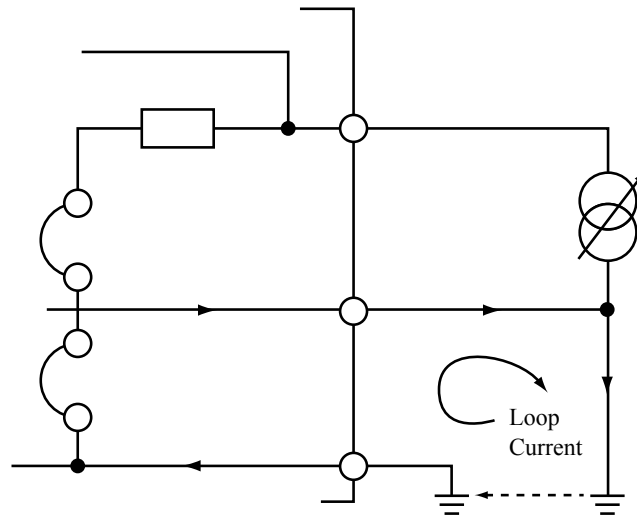


Figure 24 Example of a Ground Loop.

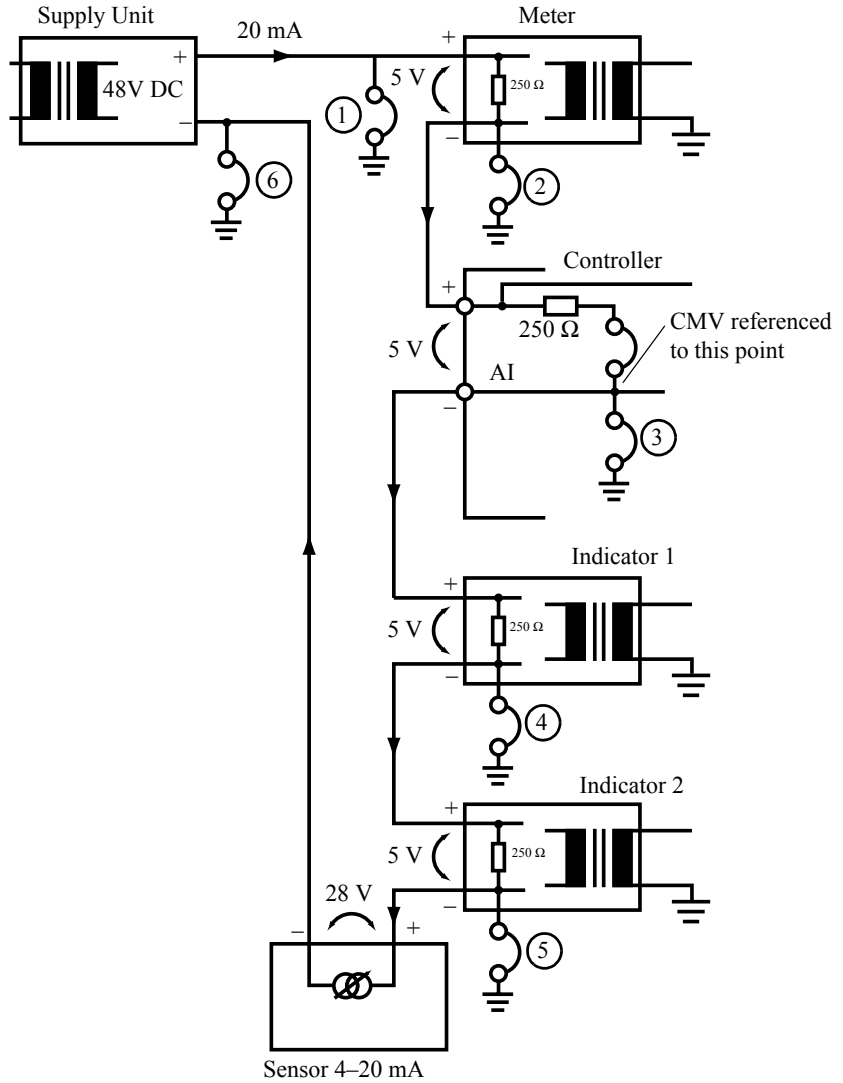


Figure 25 Example of Single Controller Current Loop.

Figure 25 shows an example of a single controller current loop. It is assumed that all the units are isolated from supply ground. This type of current loop must always be grounded at one single point so that it does not float away from ground and introduce a large interference potential. Table 2 shows the potential at points 1 to 6 with respect to ground, and indicates which points are suitable for grounding.

Point	Potential	Grounding Possibility
1	-10 V	Not permitted, max CMV = 0V
2	-5 V	Not permitted, max CMV = 0V
3	0 V	Permitted
4	+5 V	Permitted
5	+10 V	Permitted
6	+38 V	Not permitted, max CMV = +17 V

Table 2 Grounding Points.

Chapter 3

Configuration

This chapter is intended for trained personnel such as Process Control Engineers or Instrument Technicians who configure the ECA06, ECA60, ECA600 and EMA60 controllers. It describes the operator interfaces and features available in the Configuration Mode and includes full instructions for the configuration of the controllers.



When configuring, the regulator must be in Manual mode (ECA600 both loops), in order to change the parameters.



In this mode, neither the operator nor the controller can change the output signal.

3.1 Configuring the Process Value

One of the first things any user will want to do is configure the Process Value – this section shows you how to do this.



Note: To select the display language refer to the section Display Language on page 197.

Configuration Trees

The Configuration trees in Appendix C show the path to take:

Operator

Process Value

- NumberOfDecimals ($0 \leq \text{NumberOfDecimals} \leq 4$)
- MinValue ($-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$)
- MaxValue ($-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$)
- Unit (% , °C , °F , K , W , kW , MW , J , kJ , MJ , rpm , Pa , kPa , MPa , bar , mbar , bara , l , l/s , l/min , l/h , Psi , Psig , Psia , mmWG , "WG , mmvp , mvp , mmH₂O , "H₂O , mmHg , "Hg , kp/cm² , N/m² , kg/cm² , kg/h , t/h , Nm³/h , m³/h , t/24h , m/s , m/min , ms , ms/cm , ppm , pH , mg/l , Feet , inches , meters , tons , lbs , lbs/sec , lbs/min , lbs/hr , ft/sec , ft/hr , BTU/hr , BTU , ft³/sec , ft³/hr , gals , gal/sec , gal/hr , cu.ft , MPPH , cc , cm , GPM)

Parameters to Configure

There are four parameters you can configure:

- *NumberOfDecimals* defines the number of decimal places displayed for the Process Value, e.g. setting *NumberOfDecimals* to 3 will give a displayed Process Value in the form 1.234.
- *MinValue* and *MaxValue* define the lower and upper limits respectively of the acceptable range for the Process Value.
- *Unit* defines the engineering unit in which the Process Value is shown and appears on the display.

Configuration Step-by-Step

The steps listed below describe how to enter the Configuration mode and set each of these parameters in turn. As an example a Process Value within the range -20 to $+120^{\circ}\text{C}$ with two decimal places is used.

- 1 Press and hold the Next key.
- 2 If you are prompted to enter the password, press the Up and Down keys to change the displayed number until it matches your password.
- 3 When the password is correct press the OK key.

```
Config 600 001.05-00
Enter Code
      6845
Cancel      OK
```

- 4 If you enter the correct password you have read/write access to the Configuration mode.

```
Config 600 001.05-00
Code correct
      Full access
Return      OK
```

- 5 Press the OK key.
- 6 Repeatedly press the Up or Down key until you reach the Operator Block.

```
Config 600 001.05-00
Operator
Return      OK
```

- 7 Press the OK key to enter the Operator Block.

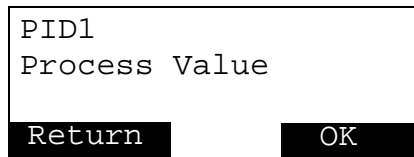


Note: If the unit is an ECA600 the following is displayed:

```
Operator
PID1
Return      OK
```

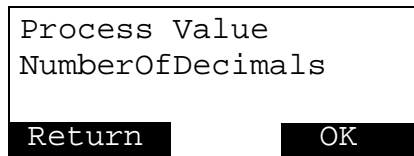
Press the OK key to proceed.

- 8 Repeatedly press the Up or Down key until you reach the Process Value section.



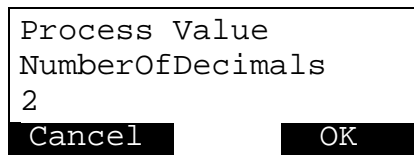
PID1
Process Value
Return OK

- 9 Press the OK key to enter the Process Value section.



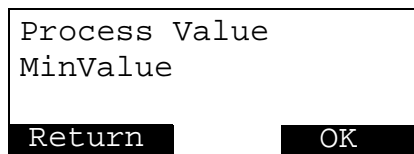
Process Value
NumberOfDecimals
Return OK

- 10 The first parameter displayed is *NumberOfDecimals*. Press the OK key to choose to edit this parameter.
- 11 Press the Up or Down key to scroll through the values 0–4 displayed on the third line of the display.
- 12 When the number of decimal places shown is 2 press the OK key.



Process Value
NumberOfDecimals
2
Cancel OK

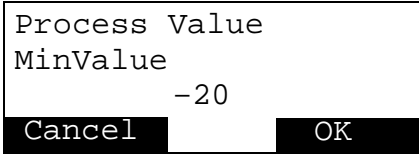
- 13 Press the Up or Down key until the second line of the display shows *MinValue*.



Process Value
MinValue
Return OK

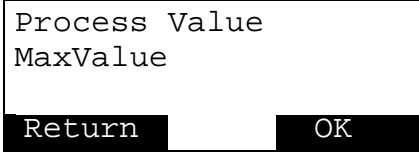
- 14 Press the OK key.
- 15 Press the Up or Down key to scroll through the range of values allowed for *MinValue*.

- 16 When the required *MinValue* -20 is displayed press the OK key.



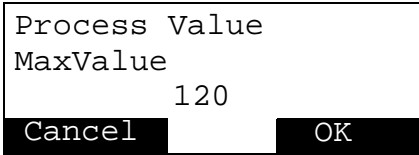
A rectangular display window with a black border. The text inside is as follows: "Process Value" on the first line, "MinValue" on the second line, and "-20" on the third line. At the bottom of the window, there are two black buttons with white text: "Cancel" on the left and "OK" on the right.

- 17 Press the Up or Down key until the second line of the display shows *MaxValue*.



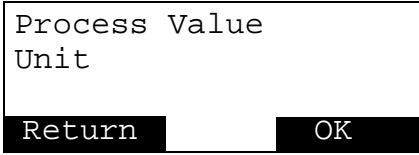
A rectangular display window with a black border. The text inside is as follows: "Process Value" on the first line, "MaxValue" on the second line, and a blank space on the third line. At the bottom of the window, there are two black buttons with white text: "Return" on the left and "OK" on the right.

- 18 Press the OK key.
- 19 Press the Up or Down key to scroll through the range of values allowed for *MaxValue*.
- 20 When the required *MaxValue* 120 is displayed press the OK key.



A rectangular display window with a black border. The text inside is as follows: "Process Value" on the first line, "MaxValue" on the second line, and "120" on the third line. At the bottom of the window, there are two black buttons with white text: "Cancel" on the left and "OK" on the right.

- 21 Press the Up or Down key until the second line of the display shows Unit.



A rectangular display window with a black border. The text inside is as follows: "Process Value" on the first line, "Unit" on the second line, and a blank space on the third line. At the bottom of the window, there are two black buttons with white text: "Return" on the left and "OK" on the right.

- 22 Press the OK key.
- 23 Press the Up or Down key to scroll through the range of units allowed.
- 24 When the required *Unit* °C is displayed press the OK key.

Process Value
Unit
°C
Cancel
OK

- 25 The Configuration is complete. Repeatedly press the Cancel key until you return to Operator Mode.

3.2 Controls and Indicators

In Configuration Mode the functions of certain controls on the front panel are different from in Operator Mode. The controls and indicators are described below.

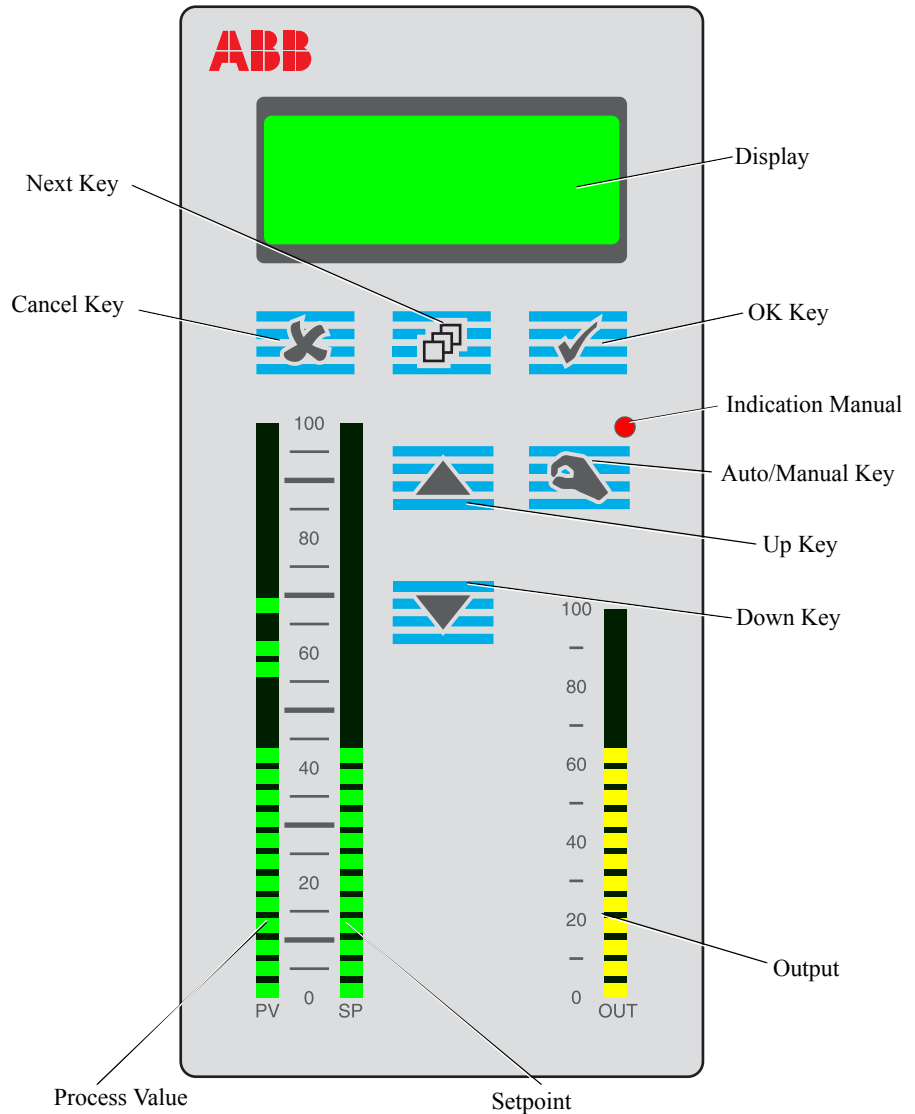


Figure 26 Controls and Indicators.

LCD Display

In Configuration Mode the display consists of four lines of text, each consisting of a maximum of 20 characters.

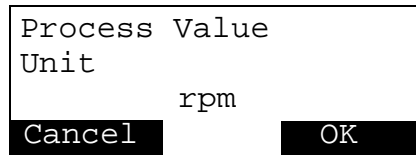


Figure 27 Typical Configuration Mode Display.

- The top line of the display shows the previous selection or current block.
- The second line shows the current selection or current parameter.
- The third line is blank until you start to change a parameter. The third line then shows the current value of the parameter and changes as the Up/Down keys are pressed. When the parameter is at the required value, press the OK key to store the value. The third line is then blank again.
- The bottom line describes the present functions of the Cancel, OK and Next keys. Each key function is labelled in inverse text above the key. If the key has no function at this particular level of the Configuration tree, the space above it on the display is blank.

PV Bar Graph Display

The PV bar graph shows the Process Value. Any alarm levels which have been set are shown as brighter segments.

SP Bar Graph Display

The SP bar graph shows the Set Point value. Any configured Set Point limits are shown as brighter segments.

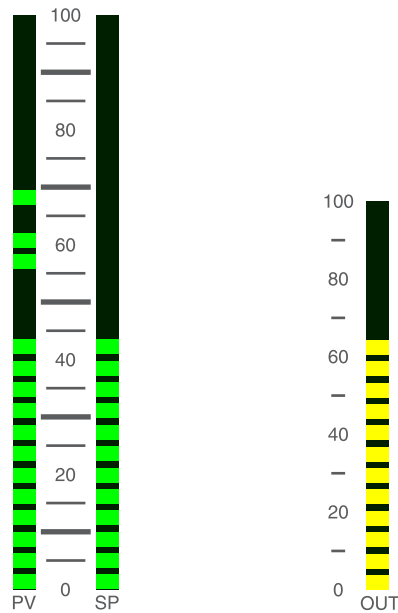


Figure 28 PV, SP and Output Bar Graph Displays.

Output Bar Graph Display

The Output bar graph shows the output level as a percentage of the output signal.

Control Keys

In the Configuration Mode the function of the OK, Cancel and Next keys depends on the present position in the menu structure. By pressing these keys and the Up and Down keys you can navigate the Configuration tree structure to enable and define all the parameters needed for Configuration.

OK Key



Press the OK key (green tick) to descend to the next level in the tree structure or to save a change to a parameter.

Cancel Key



Press the Cancel key (red cross) to ascend one level in the tree structure without making any changes to the displayed parameter.

Next Key



The function of the Next key is shown on the bottom line of the display.

Up Key



Press the Up key to step up through a list of options for the currently selected parameter or to move to the previous parameter in the current branch of the tree.

Down Key



Press the Down key to step down through a list of options on the display or to move to the next parameter in the current branch of the tree.

Hand Key



The Hand key is not enabled in the Configuration Mode.

3.3 Entering Configuration Mode

No Password Configured

To enter Configuration Mode if no password has been configured:

- 1 Press and hold the Next key.
- 2 When the display changes to show that Configuration Mode has been entered release the Next key.

When the controller enters the Configuration mode the display shows the following:

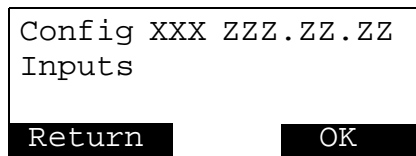


Figure 29 Configuration Mode Start-up Display.

Where:

XXX = Software type (06 for ECA06, 60 for ECA60, 600 for ECA600 and MI for EMA60).

ZZZ.ZZ-ZZ = Software Version e.g. 001.05-00.

Password Configured

To enter Configuration Mode if a password has been configured:

- 1 Press and hold the Next key.
- 2 You are prompted to enter the password. This is a number in the range 1–9999.

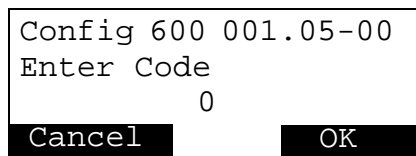


Figure 30 Password Prompt.

- 3 Press the Up and Down keys to change the displayed number. When it is correct press the OK key.

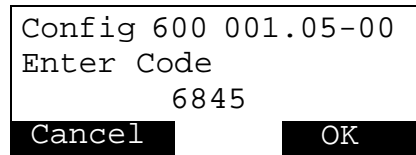


Figure 31 Password Prompt.

- 4 If you enter the correct password you have read/write access to the Configuration mode.

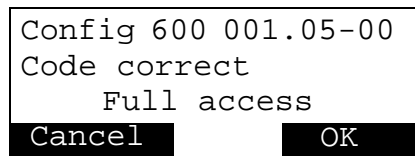


Figure 32 Full Access to Configuration Mode.

- 5 If you do not enter the correct password you have read only access.

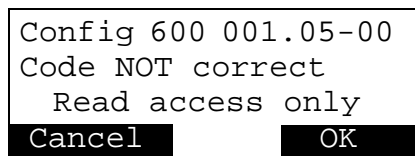


Figure 33 Read Only Access to Configuration Mode.

- 6 In either case press the OK key and you are free to navigate the Configuration tree structure using the keys as described below.
- 7 If you have restricted access pressing the Up/Down keys to change the parameter has no effect, and the OK soft key label does not appear.

3.4 Exiting Configuration Mode

To exit Configuration Mode and return to Operator Mode:

- 1 Repeatedly press the Cancel key to move back up the Configuration Menu.
- 2 When the display changes to show that the Operator Mode has been entered stop pressing the Cancel key.

3.5 Lost Code Retrieval

If you cannot remember the correct password you can retrieve it as follows:

- 1 When the display shows “Code NOT Correct, Read access only” press the Next key and hold it for a few seconds.

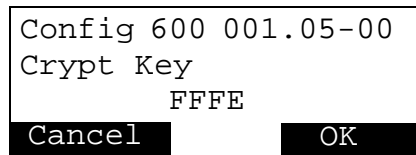


Figure 34 Accessing the Crypt Key.

- 2 The Crypt Key (FFFE in the above example) is shown on the display. This is an encoded version of the password shown in hexadecimal format.
- 3 To retrieve the password first convert the Crypt Key to decimal format and then subtract this number from 65535, i.e.

$$\text{Password} = 65535 - (\text{Crypt Key})_{10}$$

This formula should be accessible only to authorised personnel.

3.6 Setting the Configuration

Introduction

This section describes the Configuration process. Configuration is the process of setting the parameters and making connections. Parameters are configured by means of a multi-level menu system (the Configuration tree) which is navigated by the use of four control keys on the front panel. Each parameter has a unique place in this tree structure.

The full tree structure of the Configuration Menu for each controller is shown in Appendix C.

All parameters have default values. If you wish to use the default value there is no need to configure the parameter.

You can connect the inputs of some blocks to the outputs of others to create more complex functions. Application examples are given in Appendix A.

Navigating the Configuration Menu

Navigation through the Configuration Menus contained in Appendix C is via the following simple steps:

- 1 Access the Configuration Menu as described in section 3.3.
- 2 To move down one level in the menu or to accept a parameter value press the OK key.
- 3 To scroll through a range of permissible values or a list of available parameters at the currently displayed level, press the Up and Down keys.
- 4 To go back one level in the Configuration Menu or to leave the displayed parameter unchanged press the Cancel key.

Example

To configure the *Mode* parameter for analog output AO1 the Configuration tree shows the path to take:

```
Outputs
  Analog Outputs
    AO1
      • Mode
```


The key sequence to set this parameter is given in full below but can be abbreviated to the form:

Outputs → Analog Outputs → AO1 → Mode = Reversed

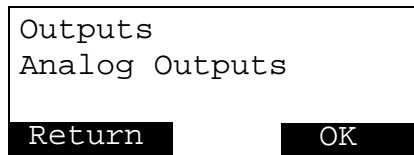
To configure the mode parameter proceed as follows:

- 1 Access the Configuration Menu as described in section 3.3.
- 2 Press the Up or Down key until the second line of the display shows Outputs.



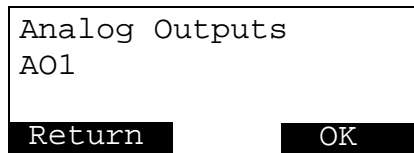
```
Config 600 001.05-00
Outputs
Return      OK
```

- 3 Press the OK key.
- 4 Press the Up or Down key until the second line of the display shows Analog Outputs.



```
Outputs
Analog Outputs
Return      OK
```

- 5 Press the OK key.
- 6 Press the Up or Down key until the second line of the display shows AO1.



```
Analog Outputs
AO1
Return      OK
```

- 7 Press the OK key.
- 8 Press the Up or Down key until the second line of the display shows Mode.

```

AO1
Mode
Return      OK

```

- 9 Press the OK key.
- 10 Press the Up or Down key until the third line of the display shows the required mode, e.g. Reverse.

```

AO1
Mode
          Reversed
Cancel      OK

```

- 11 Press the OK key.
- 12 The Configuration is complete. Repeatedly press the Cancel key until you return to Operator Mode.

Menu Abbreviations

To simplify the instructions in this manual an “x” is used to represent any of the blocks of each type available in each controller. For example, FAX represents any of the numbered Arithmetic Function Blocks.

The specific number of blocks in each type of controller is given in a table in the left hand margin next to the block being described.

Some types of function blocks are available in a limited number. An example is the delay block in the ECA600, where only two instances are available. If you try to configure a third Delay block, a message is displayed on the third row of the display.

```

FO4: Func Other 4
Type
This Type Used Up
          OK

```

The message must be acknowledged by pressing the OK key.

Processing Columns	Blocks	Comment
Input	AI	Analog Inputs
	AIC	Analog Input Communication Buffer
	AU	Analog User
	DI	Digital Input
	DIC	Digital Input Communication Buffer
	DU	Digital User
Input Function	FA	Arithmetic Function
	FL	Logic Function
	FO	Other Function
Set Point	SP	Set Point
Alarm	AA	Absolute Alarm
	DA	Deviation Alarm
Control	CO	Control
	OC	Output Configuration
Output Function	OF	Output Function
Output	AO	Analog Output
	AOC	Analog Output Communication
	DO	Digital Output
	DOC	Digital Output Communications
Operator	OP	Operator
System	SY	System

Table 3 Menu Abbreviations.

Naming Blocks

You can give many of the blocks and parameters which you configure a twenty character name of your own choice. To use this feature proceed as follows:

- 1 Select the option Identity. All names have a default setting.
- 2 Select the character to change by pressing the Next key. The currently selected character is shown in reversed text.
- 3 Press the Up/Down keys to select the displayed character.
- 4 Repeat steps 2 to 3 until the full name has been edited.
- 5 Confirm the completed name by pressing the OK key.

Internal Connections

Types

Internal connections are divided into two different categories:

- Fixed Connections
- Free Connections

Fixed Connections are preset at the factory and are not changeable by the user. The Fixed Connections are shown with thick lines on the Configuration Work Sheet (Appendix D).

The Free Connections may have default settings from the factory, but the user can always change them.

Making Connections

When connecting inputs and outputs, the inputs are always assigned an output. In the transfer of a value from an output to an input, the input takes the initiative. In other terms, the input requests a value from the output it is connected to.

The information about the connections are thus stored in the inputs. When making a connection, the following happens:

- 1 The system locates all available blocks (blocks that are not switched OFF).
- 2 The system locates all valid outputs of these blocks (to an analog input, only analog outputs are valid) and marks them as available.



Notes:

- If an available block has no valid outputs, no outputs of that block are made available for connection.
- If a valid output belongs to a block which is not available, it is not made available for connection.
- The same signal can be shared between several function blocks.
- Blocks may be combined to produce more complex logic functions.

If an output of an Output Function block (OF) is referenced and then the Type parameter setting of the (OF) block is changed, references to outputs of that block will become invalid. The input referencing the invalid output will show the following connection information:

```
A01: Ana Out 1
Signal
OF1[Analog.Invalid]
Cancel      OK
```

Example

Suppose a connection has been made between an output of block A and an input of block B. Then suppose block A is switched OFF. Now this B input is connected to an output whose value is undefined.

In this situation, when block B is executed and the input above requests a value from the A output, the controller will continue to execute, and the following will happen:

- 1 The A output signals to the error handling system that it has been erroneously referenced
- 2 The A output returns a predefined value to the B input enabling the B block to complete its calculations. All analog outputs have the “off value” 0.0, and all digital outputs have the “off value” False.

If the input to a block is OFF, it represents a value to this block that is neutral to the function of that block.



Notes:

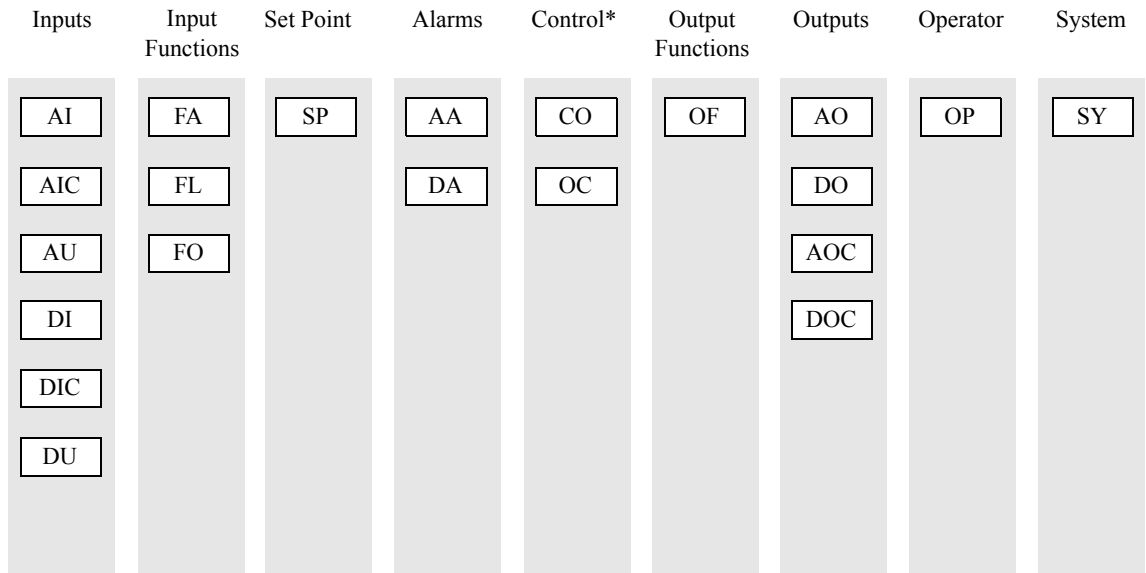
- Any analog input of any block type can be connected to any analog output of any block type.

- Any digital input of any block type can be connected to any digital output of any block type.

Overview

The controllers each have six main configurable sections, called columns:

- The Input column contains blocks that accept signals from direct wiring, or from the communication channels, or from values input by the user on the front panel. These inputs may be analog or digital.
- The Input Functions column contains blocks that mainly manipulate the input signals from the Input Blocks. There are three types – Arithmetic Function Blocks (FA), Logic Function Blocks (FL) and Other Function Blocks (FO). The Other Function Blocks have inputs and outputs that are both analog and digital.
- The Set Point column contains Set Points.
- The Alarm column contains Absolute alarms and Deviation alarms.
- The Control column contains Control and Output Configuration blocks which carry out control and related functions.
- The Output column contains Output blocks that pass the analog or digital control signals either to the communications channels or to directly wired outputs.
- The Operator Block processes all user interaction and scales the Process Value into engineering units.
- The System column contains communication configurations, status and language.

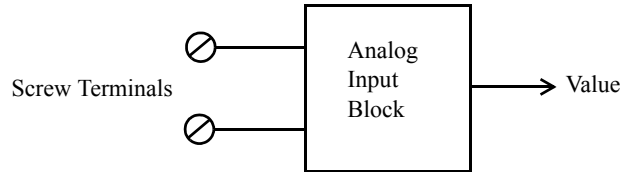


*The EMA60 does not have a Control block

Figure 35 Processing Columns. The inputs and outputs are not shown. Refer to “Configuration Work Sheets”, on page 277.

Available in			
ECA06	ECA60	ECA600	EMA60
2	3	5	3

Analog Inputs



Each Analog Input Block (AI) can accept one external analog signal as an input and produces one analog output, *Value*. The type of input – current or voltage signal and single-ended or differential – is selected by jumpers at the rear of the unit. Each block converts the signal applied to the terminals of the controller to a floating point value. All the required scaling and calculation is done within the block. The output *Value* is in the range 0.0 to 1.0 which corresponds to 0 to 100% of the input. Each block also holds the parameters *Mode*, *RootExtraction*, *FilterTime*, *Alarm*, *Loop* and *Identity*.

Note: The *Loop* parameter is only available on the ECA600.



Parameters		
Name	Setting	Unit
Mode	Off, 0–20 mA, 4–20 mA, 0–5 V, 1–5 V, 0–10 V, 2–10V	None
Root Extraction	Off, On	None
FilterTime	$0.0 \leq \text{FilterTime} \leq 100.0$	seconds
Alarm	Off, Warning, Error	None
Loop	Loop1, Loop2	None
Identity	User defined text	None

Table 4 Analog Input Block Parameters

Outputs	
Name	Type
Value	Analog

Table 5 Analog Input Block Outputs.

To select an analog input for use, configure the parameter *Mode* for the type of input which will be connected to the block. To deselect an analog input, set the parameter *Mode* to Off.

By setting the *Root Extraction* parameter to On the characteristic of the signal on the terminals is changed according to the graph shown below.

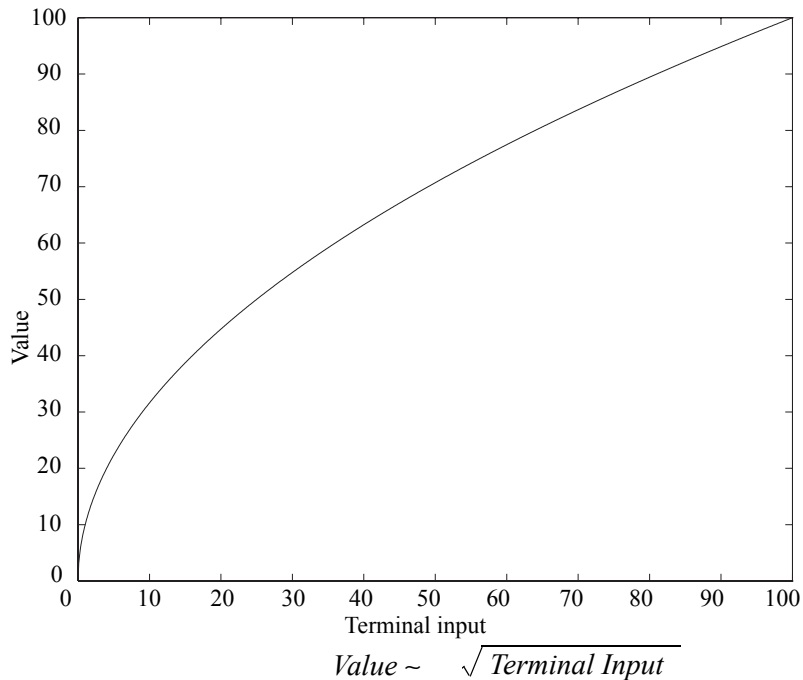


Figure 36 Root Extraction Characteristic.

Example

This function is useful in flow applications where differential pressure is measured to obtain the flow:

Terminal Input = Differential Pressure and Block Output = Flow

$$Flow \sim \sqrt{Differential\ Pressure}$$

The filter function of the analog Input block is activated by setting the parameter *Filter Time* to a non-zero value. The filter characteristic is shown below.

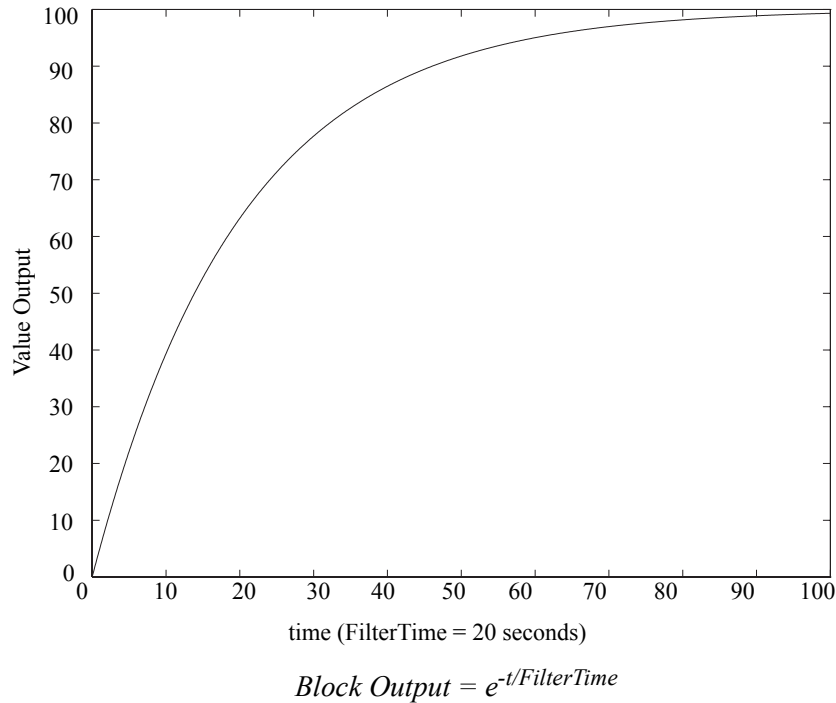


Figure 37 Filter Characteristic.



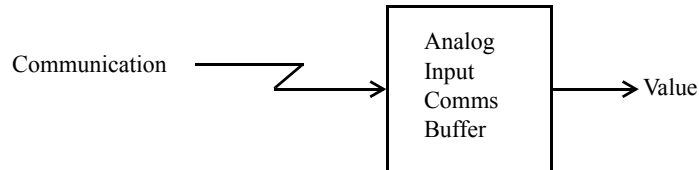
Notes:

- For details of the parameter *Alarm* please refer to Appendix B Error, Warning and Information Messages.
- If a warning is set the controller displays a message while continuing to operate normally.
- The parameter *Loop* is used only on the ECA600. Alarms can be set to act on either loop independently. The control block designated by the *Loop* parameter for alarm will react according to the parameter *ErrorBehavior* in the Control block. The other Control block (loop) will continue to function as normal.

- Each block may be given a twenty letter identification name by configuring the parameter *Identity*.

Available in			
ECA06	ECA60	ECA600	EMA60
4	8	12	4

Analog Input Communication Buffers



Remote analog communication is provided by Analog Input Communications Buffers (AIC) via a two wire RS485 serial channel using a communication protocol. Each Analog Communications Buffer can accept one register value and produces one analog output, *Value*. Each buffer block also holds two parameters.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
Identity	User defined text	None

Table 6 Analog Input Communications Buffer Parameters.

Outputs	
Name	Type
Value	Analog

Table 7 Analog Input Communications Buffer Outputs.

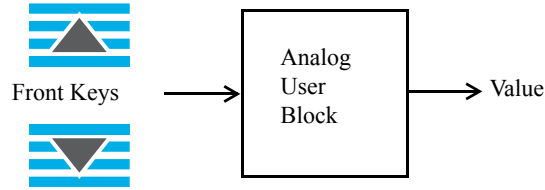
To select an analog input for use, configure the parameter *Mode* to on. To deselect an analog input, set the parameter *Mode* to Off.

Each block may be given a twenty letter identification name by configuring the parameter *Identity*.

Each buffer corresponds to one unique register. See Appendix E for a full list of registers and their addresses.

Available in			
ECA06	ECA60	ECA600	EMA60
-	2	8	4

Analog User Blocks



Analog User Blocks (AU) can be configured so that the operator can enter analog values using the front panel keys. These user defined inputs are added to the operator option list if they are enabled during Configuration. The blocks can be set to Ratio or Bias Values while the controller is in operation.

Each block produces one analog output signal, *Value*. Each block also holds seven parameters.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
NumberOfDecimals	$0 \leq \text{NumberOfDecimals} \leq 4$	None
LoLimit	$-99999 \leq \text{LoLimit} \leq 99999$	None
HiLimit	$-99999 \leq \text{HiLimit} \leq 99999$	None
Identity	User defined text	None
MinValue	$-100.0 \leq \text{MinValue} \leq \text{MaxValue} \leq 100.0$	%
MaxValue	$-100.0 \leq \text{MinValue} \leq \text{MaxValue} \leq 100.0$	%

Table 8 Analog User Block Parameters.

Outputs	
Name	Type
Value	Analog signal

Table 9 Analog User Block Outputs.

To enable an Analog User Block, set the parameter *Mode* to on. To disable the block set *Mode* to Off.

To set the number of decimals displayed for the signal value configure the parameter *NumberOfDecimals*. This also defines the step increase/decrease, i.e. *NumberOfDecimals* = 1, step = 0.1, *NumberOfDecimals* = 2, step = 0.01.

Each user defined input must lie within a specified range. Configure the parameters *LoLimit* and *HiLimit* to set this range.

The output of this type of block is always in the range -100.0% to $+100.0\%$. *LoLimit* always represents the value -1.0 (-100%) and *HiLimit* the value 1.0 (100%).

Each block may be given a twenty letter identification name by configuring the parameter *Identity*.

There are two ranges to keep track of:

- Signal Range
- Displayed Range

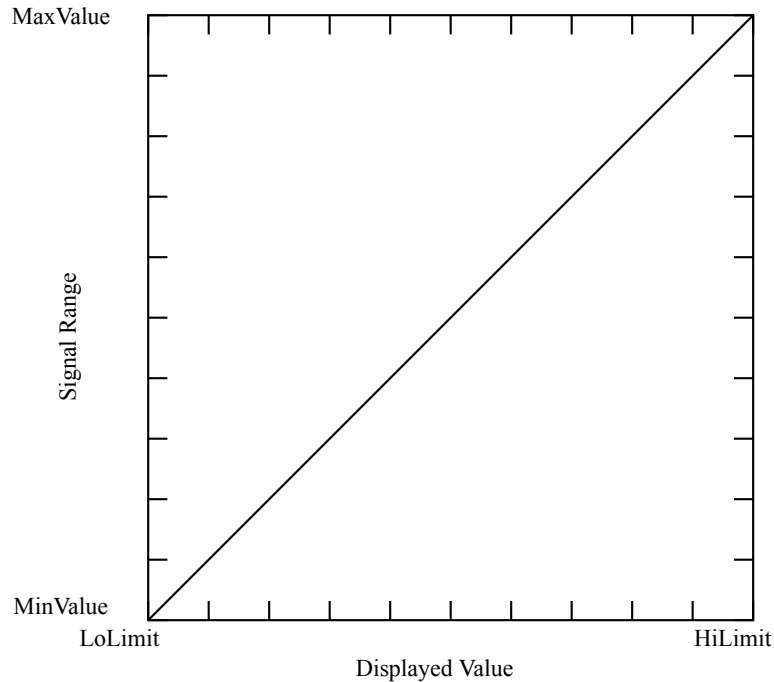


Figure 38 The relationship between Displayed Value and Signal Range.

The Signal Range is the range of the signal available on the output of the block. This range is

$$\text{MinValue} \leq \text{Value} \leq \text{MaxValue}$$

The default values of *MinValue* (0.0%) and *MaxValue* (100.0%) define the default (signal) range

$$-100.0\% \leq \text{Value} \leq 100\%$$

By adjusting *MinValue* and *MaxValue*, the user can limit the (signal) range to positive values only (*MinValue* = 0.0%) or any other permissible range.

The Displayed Range is the range of the value presented on the display. This range is

$$\text{LoLimit} \leq \text{Displayed Value} \leq \text{HiLimit}$$

When an analog user block is activated, it appears in the Operator Option List. Here the operator can adjust the (output) signal by pressing the UP/DOWN keys. It is possible to make the Displayed Range differ from the Signal Range by changing the parameters *LoLimit* and *HiLimit*.

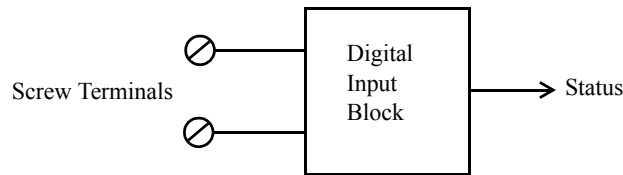
The default values of *LoLimit* (0.0) and *HiLimit* (100.0) define the default (displayed) range

$$-100.0 \leq \text{Displayed Value} \leq 100.0$$

LoLimit determines the value displayed for *MinValue*, and *HiLimit* determines the value displayed for *MaxValue*.

Available in			
ECA06	ECA60	ECA600	EMA60
1	3	4	2

Digital Inputs



Digital Input Blocks (DI) can accept one digital input signal and produce one digital output signal, *Status*. Each block also holds two parameters.

Parameters		
Name	Setting	Unit
Mode	Off, PositiveLogic, NegativeLogic	None
Identity	User defined text	None

Table 10 Digital Input Block Parameters.

Outputs	
Name	Type
Status	Digital

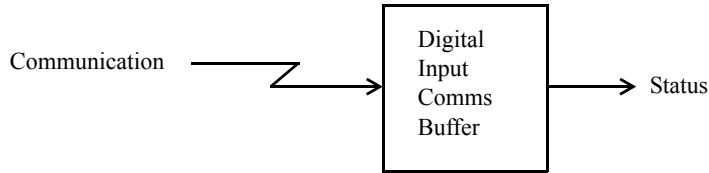
Table 11 Digital Input Block Outputs.

To select a digital input for use, set the parameter *Mode* to the required logic mode. To deselect a digital input, set the parameter *Mode* to Off.

Each block may be given a twenty letter identification name by configuring the parameter *Identity*

Available in			
ECA06	ECA60	ECA600	EMA60
4	8	12	4

Digital Input Communications Buffers



Remote digital communication is provided by Digital Input Communications Buffers (DIC). Via communication a master can feed digital values to the controller. This is done by writing a bit value into the I/O bit area.

Each Digital Communications Buffer represents a I/O bit.

To configure a Digital Communications Buffer for use, set the parameter *Mode* to on. To deselect a digital buffer, set the parameter *Mode* to Off.

Each block may be given a twenty letter identification name by configuring the parameter *Identity*.

Each buffer corresponds to one unique I/O bit.



Note. See Appendix E for a full list of I/O bits and their addresses.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
Identity	User defined text	None

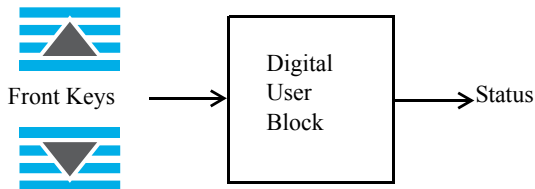
Table 12 Digital Input Communications Buffers Parameters.

Outputs	
Name	Type
Status	Digital

Table 13 Digital Input Communications Buffers Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	2	8	4

Digital User Blocks



Digital User Blocks (DU) can be configured so that the operator can enter digital levels using the front panel keys. These user defined inputs are added to the operator option list if they are enabled during Configuration. When a block represents a False value the text “False” is shown on the display. When the block represents a True value the text “True” is shown on the display.

Each block can accept one digital input and produces one digital output signal, *Status*. Each block also holds two parameters, *Mode* and *Identity*.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
Identity	User defined text	None

Table 14 Digital User Block Parameters.

Outputs	
Name	Type
Status	Digital

Table 15 Digital User Block Outputs.

To enable a Digital User Block, set the parameter *Mode* to on. To disable the block set *Mode* to Off. Each block may be given a twenty letter identification name by configuring the parameter *Identity*.

Input Function Blocks

There are three kinds of input function blocks:

- Arithmetic Function Blocks (FA) which perform arithmetic operations on analog input signals and produce analog outputs.
- Logical Function Blocks (FL) which perform logic operations on digital input signals and produce digital outputs.
- Other Function Blocks (FO) which perform operations on a mix of analog and digital inputs and produce a mix of analog and digital outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
2	4	16	8

Arithmetic Function Blocks

The input to any block may be connected to the output of any other block provided that they are of the same type, i.e. analog to analog or digital to digital. Each output may be connected to several inputs if required. The Arithmetic Function Blocks (FA) can be interconnected to allow more complex functions to be constructed from basic operators. The available arithmetic function blocks are summarised in Table 16.

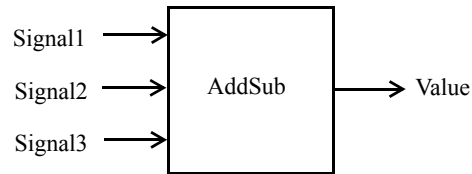
Parameters		
Name	Setting	Unit
Type	Off, AddSub, MulDiv, Exp, Comp	None
Identity	User defined text	None

Table 16 Arithmetic Function Blocks.

To configure an Arithmetic Function Block for use:

- 1 First select the block number to be used.
- 2 Select the required type.
- 3 Connect the required inputs to their respective outputs and define any parameters required for the operation of the arithmetic function, or use default values.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

AddSub

$$Value = Factor1 \times Signal1 + Factor2 \times Signal2 + Factor3 \times Signal3 + Bias$$

The AddSub function has three analog inputs, *Signal1*, *Signal2* and *Signal3*, and one analog output, *Value*. There is one gain factor associated with each input, *Factor1*, *Factor2* and *Factor3*, and one bias value, *Bias*.

The *Value* output is equal to the sum of the three gain adjusted inputs added to the bias.

Inputs	
Name	Connection
Signal1	Off, Analog
Signal2	Off, Analog
Signal3	Off, Analog

Table 17 AddSub Inputs.

Parameters		
Name	Setting	Unit
Factor1	$-10.00 \leq Factor1 \leq 10.00$	None
Factor2	$-10.00 \leq Factor2 \leq 10.00$	None
Factor3	$-10.00 \leq Factor3 \leq 10.00$	None
Bias	$-10.00 \leq Bias \leq 10.00$	None

Table 18 AddSub Parameters.

Outputs	
Name	Type
Value	Analog

Table 19 AddSub Outputs.

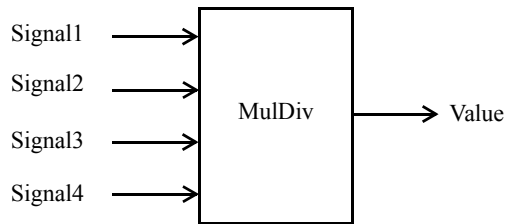


Note: The AddSub block in the ECA06 does not have the input Signal3 or the parameter Factor3 and hence the function is defined as:

$$Value = Factor1 \times Signal1 + Factor2 \times Signal2 + Bias$$

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

MulDiv



$$Value = Factor \times \frac{Signal1 \times Signal2}{Signal3 \times Signal4} + Bias$$

The MulDiv function has four analog inputs, *Signal1*, *Signal2*, *Signal3* and *Signal4*, and one analog output, *Value*. The block holds two parameters; a gain *Factor*, and *Bias*.



Note: If any input is unconnected, it represents the value 1.0.

Inputs	
Name	Connection
Signal1	Off, Analog
Signal2	Off, Analog
Signal3	Off, Analog
Signal4	Off, Analog

Table 20 MulDiv Inputs.

Parameters		
Name	Setting	Unit
Factor	$-10.00 \leq \text{Factor} \leq 10.00$	None
Bias	$-10.00 \leq \text{Bias} \leq 10.00$	None

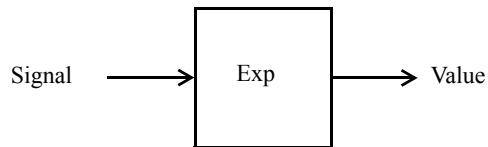
Table 21 MulDiv Parameters.

Outputs	
Name	Type
Value	Analog signal

Table 22 MulDiv Block Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Exp



$$Value = Factor \times Signal^{Power} + Bias$$

The Exp function has one analog input, *Signal*, and one analog output, *Value*. It also holds three analog parameters – the exponential *Power*, a gain *Factor* and a *Bias*.

Inputs	
Name	Connection
Signal	Off, Analog

Table 23 Exp Inputs.

Parameters		
Name	Setting	Unit
Power	$0.00 \leq \text{Power} \leq 10.00$	None
Factor	$-10.00 \leq \text{Factor} \leq 10.00$	None
Bias	$-10.00 \leq \text{Bias} \leq 10.00$	None

Table 24 Exp Parameters.

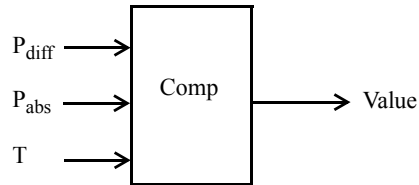
Outputs	
Name	Type
Value	Analog signal

Table 25 Exp Outputs.



Note: If the input is not connected it represents the value $0.0 + \text{Bias}$ in the calculation.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Comp

$$Value = Factor \times \sqrt{P_{diff} \times \left[\frac{P_{abs} \times (P_{absMax} - P_{absMin}) + P_{absMin}}{P_{orifice}} \right]} \times \left[\frac{T_{orifice} + 273}{T \times (T_{max} - T_{min}) + T_{min} + 273} \right]$$

The Comp block is used for flow measurement with pressure and/or temperature compensation. The block has three analog inputs, P_{diff} , P_{abs} and T , and one analog output, $Value$. It also holds seven analog parameters.

Inputs	
Name	Connection
P_{diff}	Off, Analog
P_{abs}	Off, Analog
T	Off, Analog

Table 26 Comp Inputs.

Parameters		
Name	Setting	Unit
Factor	$-10.0 \leq K \leq 10.0$	none
P _{absMax}	$0.0 \leq P_{absMax} \leq 10000.0$	see note
P _{absMin}	$0.0 \leq P_{absMin} \leq 10000.0$	see note
P _{orifice}	$0.0 \leq P_{orifice} \leq 10000.0$	see note
T _{orifice}	$-273.0 \leq T_{orifice} \leq 10000.0$	°C
T _{max}	$-273.0 \leq T_{max} \leq 10000.0$	°C
T _{min}	$-273.0 \leq T_{min} \leq 10000.0$	°C

Table 27 Comp Parameters.

Outputs	
Name	Type
Value	Analog signal

Table 28 Comp Outputs.

If only absolute pressure compensation is required set $T = \text{Off}$:

$$Value = Factor \times \sqrt{P_{diff} \times \left[\frac{P_{abs} \times (P_{absMax} - P_{absMin}) + P_{absMin}}{P_{orifice}} \right]}$$

If only temperature compensation is required set $P_{abs} = \text{Off}$:

$$Value = Factor \times \sqrt{P_{diff} \times \left[\frac{T_{orifice} + 273}{T \times (T_{max} - T_{min}) + T_{min}} + 273 \right]}$$

Summary

Term	Type	Unit	Comment
Factor	Constant	None	-
P _{diff}	Analog input	None	Differential pressure signal from differential pressure sensor. Signal not linearized.
P _{abs}	Analog input	None	Absolute pressure sensor signal. Set to Off if no absolute pressure compensation is required.
P _{absMax}	Constant	see note	Pressure sensor upper measuring range limit (absolute pressure).
P _{absMin}	Constant	see note	Pressure sensor lower measuring range limit (absolute pressure).
P _{orifice}	Constant	see note	Orifice plate design pressure.
T _{orifice}	Constant	°C	Orifice plate design temperature.
T	Analog input	None	Temperature signal (linearised) from transmitter. Set to Off if no temperature compensation is required.
T _{max}	Constant	°C	Temperature sensor upper measuring range limit.
T _{min}	Constant	°C	Temperature sensor lower measuring range limit.

Table 29 Summary of Comp Parameters.



Note: All pressure parameters must be given the same engineering unit.

Available in			
ECA06	ECA60	ECA600	EMA60
-	2	16	8

Logic Function Blocks

The Logic Function Blocks (FL) perform logical operations on the digital input signals before their values are used in the control system.

The available logic function blocks are summarised below.

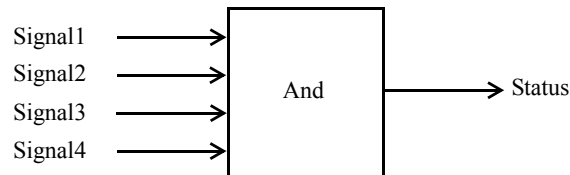
Parameters		
Name	Setting	Unit
Type	Off, And, Or, Xor, S/R, Not	none
Identity	User defined text	none

Table 30 Available Logic Operations.

The Logic Function Blocks can be interconnected to allow more complex logic functions to be constructed from basic operators. For example connecting a digital function block set to NOT to the output of a block that is set to AND will give the logical NAND function.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

And



$$\text{Status} = \text{Signal1 AND Signal2 AND Signal3 AND Signal4}$$

The And function has four digital inputs, *Signal1*, *Signal2*, *Signal3*, *Signal4* and one digital output, *Status*.

The output, *Status*, is True whenever all of the inputs are True. Otherwise *Status* is False.

Inputs	
Name	Connection
Signal1	Off, Digital
Signal2	Off, Digital
Signal3	Off, Digital
Signal4	Off, Digital

Table 31 And Inputs.

Outputs	
Name	Type
Status	Digital

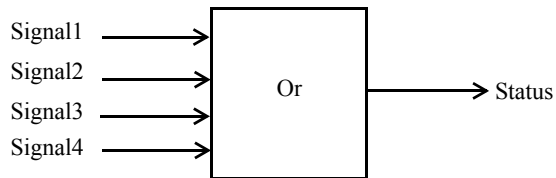
Table 32 And Outputs.



Note: If an input is not connected it represents the value *True* in the calculation.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Or



$$Status = Signal1 \text{ OR } Signal2 \text{ OR } Signal3 \text{ OR } Signal4$$

The Or block has four digital inputs, *Signal1*, *Signal2*, *Signal3* and *Signal4*, and one digital output, *Status*.

The output, *Status*, is True whenever one or more of the inputs are True. Otherwise *Status* is False.

Inputs	
Name	Connection
Signal1	Off, Digital
Signal2	Off, Digital
Signal3	Off, Digital
Signal4	Off, Digital

Table 33 Or Inputs.

Outputs	
Name	Type
Status	Digital

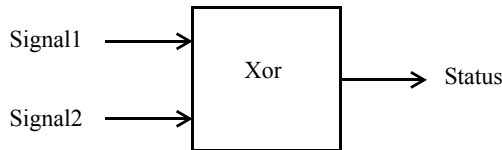
Table 34 Or Outputs.



Note: If an input is not connected it represents the value False in the calculation.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Xor



$$Status = Signal1 \text{ XOR } Signal2$$

The Xor function has two digital inputs, *Signal1* and *Signal2*, and one digital output, *Status*. The function is summarised in the table below.

Inputs		Output
Signal1	Signal2	Status
False	False	False
False	True	True
True	False	True
True	True	False

Table 35 Xor Function.

Inputs	
Name	Connection
Signal1	Off, Digital
Signal2	Off, Digital

Table 36 Xor Inputs.

Outputs	
Name	Type
Status	Digital

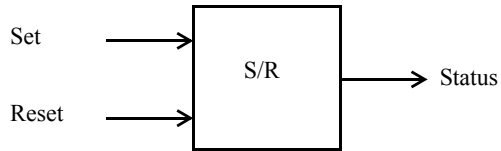
Table 37 Xor Outputs.



Note: If an input is not connected it represents the value False in the calculation.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

S/R



The S/R function has two digital inputs, *Set* and *Reset*, and one digital output, *Status*.

The block performs the function of an S/R flip-flop as summarised in the table below.

Input		Output
Set	Reset	Status
False	False	Unchanged
False	True	False
True	False	True
True	True	Unchanged

Table 38 S/R Function.

When *Set* changes state from False to True, the output, *Status*, is activated. The output then remains True whatever happens to *Set*, provided *Reset* does not change.

When *Reset* changes state from False to True, the output, *Status*, changes state to False. The output remains False whatever happens to *Reset*, provided that *Set* does not change.

Inputs	
Name	Connection
Set	Off, Digital
Reset	Off, Digital

Table 39 S/R Inputs.

Outputs	
Name	Type
Status	Digital

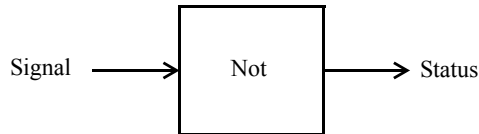
Table 40 S/R Outputs.



Note: If an input is not connected it represents the value False in the calculation.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Not



$$Status = NOT Signal$$

The Not function has one digital inputs, *Signal*, and one digital output, *Status*.

The output, *Status*, is True whenever *Signal* is False, and False when *Signal* is True.

Inputs	
Name	Connection
Signal	Off, Digital

Table 41 Not Inputs.

Outputs	
Name	Type
Status	Digital

Table 42 Not Outputs.



Note: If the input is not connected the output is given the value True.

Other Functions

Available in			
ECA06	ECA60	ECA600	EMA60
-	2	16	8

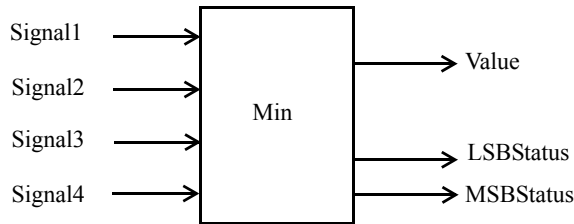
The Other Function Blocks (FO) perform logical and arithmetic operations on digital or analog input signals. Their outputs to the control system are also mixed analog and digital signals.

The configuration options for the mixed function blocks are summarised below.

Parameters		
Name	Setting	Unit
Type	Off, Min, Max, Select, Level Detector, Delay, IncDec, Linear, Rate Limiter, Integrator, Time Measurement	None
Identity	User defined text	None

Table 43 Mixed Function Options.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Min

$$Value = \text{Min}(\text{Signal1}, \text{Signal2}, \text{Signal3}, \text{Signal4})$$

The Min function has four analog inputs *Signal1*, *Signal2*, *Signal3*, *Signal4*, one analog output, *Value*, and two digital outputs *LSBStatus* and *MSBStatus*.

The block selects the input with the smallest value and copies this value to the output *Value*. The input selected by the block can be read from *LSBStatus* and *MSBStatus*. Examples are given in the table below.

Selected Input	MSBStatus	LSBStatus
Signal1	False	False
Signal2	False	True
Signal3	True	False
Signal4	True	True

Table 44 Min Operation.



Note: If two inputs have the same value, *LSBStatus* and *MSBStatus* indicate the input with the smallest ordinal number.

Inputs	
Name	Connection
Signal1	Off, Analog
Signal2	Off, Analog
Signal3	Off, Analog
Signal4	Off, Analog

Table 45 Min Inputs.



Note: If any input is set to Off it is not taken into account when making the selection.

Outputs	
Name	Type
Value	Analog signal
LSBStatus	Digital
MSBStatus	Digital

Table 46 Min Outputs.

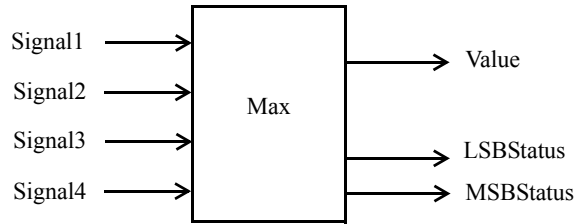
Min Example

Signal1	Signal2	Signal3	Signal4	Value	MSBStatus	LSBStatus
0.5	1.2	2.6	0.1	0.1	True	True
0.1	1.2	2.6	0.1	0.1	False	False

Table 47 Min Example.

LSBStatus and *MSBStatus* form a binary number that identifies the input with the smallest value. The Signal1 input is denoted 0 (zero), Signal2 is denoted 1 (one), Signal3 is denoted 2 (two) and Signal 4 is denoted 3 (three).

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Max

$$Value = \text{Max}(\text{Signal1}, \text{Signal2}, \text{Signal3}, \text{Signal4})$$

The Max function has four analog inputs *Signal1*, *Signal2*, *Signal3*, *Signal4*, one analog output, *Value*, and two digital outputs *LSBStatus* and *MSBStatus*.

This function selects the input with the largest value and copies this value to the output *Value*. The input selected by the block can be read from *LSBStatus* and *MSBStatus*. Examples are given in the table below.

Selected Input	MSBStatus	LSBStatus
Signal1	False	False
Signal2	False	True
Signal3	True	False
Signal4	True	True

Table 48 Max Operation.



Note: If two inputs have the same value, *LSBStatus* and *MSBStatus* indicate the input with the smallest ordinal number.

Inputs	
Name	Connection
Signal1	Off, Analog
Signal2	Off, Analog
Signal3	Off, Analog
Signal4	Off, Analog

Table 49 Max Inputs.



Note: If any input is set to Off it is not taken into account when making the selection.

Outputs	
Name	Type
Out	Analog signal
LSBStatus	Digital
MSBStatus	Digital

Table 50 Max Block Outputs.

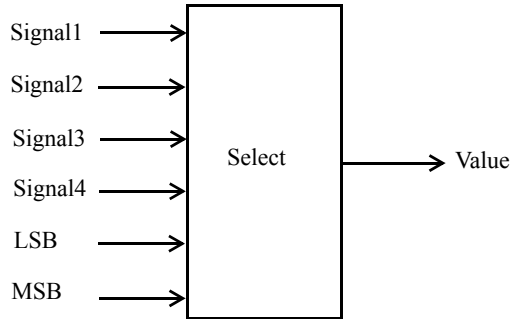
Max Example

Signal1	Signal2	Signal3	Signal4	Value	MSBStatus	LSBStatus
0.5	1.2	2.6	0.1	0.1	True	False
3.1	1.2	2.6	0.1	0.1	False	False

Table 51 Max Example.

MSBStatus and *LSBStatus* form a binary number that identifies the input with the greatest value. The *Signal1* input is denoted 0 (zero), *Signal2* is denoted 1 (one), *Signal3* is denoted 2 (two) and *Signal 4* is denoted 3 (three).

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Select

The Select function has four analog inputs *Signal1*, *Signal2*, *Signal3*, *Signal4*, and two digital inputs *LSB* and *MSB*. The single analog output is the same as the selected input.

This function can be used to select one input out of four or one input out of two to be used as the output. When using two inputs the *LSB* input is used to select the one to be copied to the output. When using four inputs both the *LSB* input and the *MSB* input are used to make the selection. The function is summarised in the table below.

MSB	LSB	Value
False	False	Signal1
False	True	Signal2
True	False	Signal3
True	True	Signal4

Table 52 Select Function.

Inputs	
Name	Connection
Signal1	Off, Analog
Signal2	Off, Analog
Signal3	Off, Analog
Signal4	Off, Analog
LSB	Off, Digital
MSB	Off, Digital

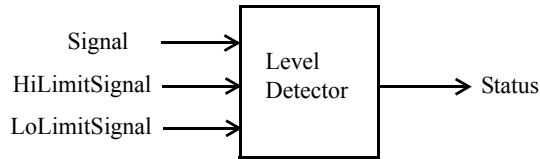
Table 53 Select Inputs.

Outputs	
Name	Type
Value	Analog signal

Table 54 Select Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Level Detector



The Level Detector function has three analog inputs *Signal*, *HiLimitSignal*, *LoLimitSignal* and one digital output *Status*. It holds two parameters, *HiLimit* and *LoLimit*.

The incoming *Signal* is compared with two values, Hi and Lo. When the value of the *Signal* input becomes greater than the value of Hi *Status* is set to True. *Status* remains True until *Signal* becomes less than Lo and then it becomes False. *Status* remains False until *Signal* becomes greater than Hi again and then it becomes True.

If *LoLimitSignal* = Off, then *LoLimitValue* is used for Lo, otherwise *LoLimitSignal* is used. If *HiLimitSignal* = Off, then *HiLimitValue* is used for Hi, otherwise *HiLimitSignal* is used. In short, if the input is connected, the value of the input is used, otherwise the value of the parameter is used. It is possible to have one input connected and the other unconnected.

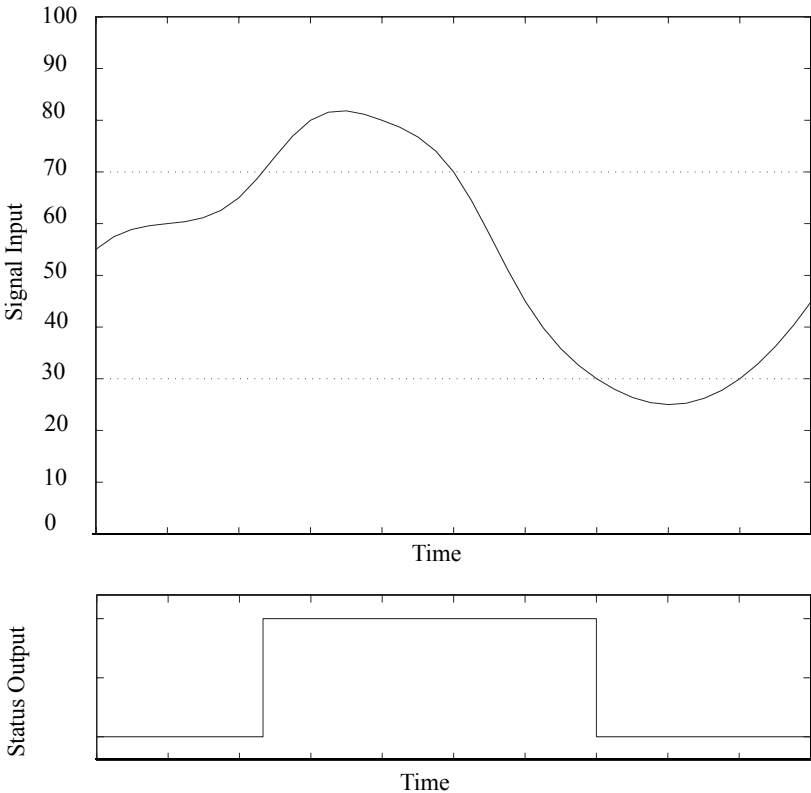


Figure 39 Level Detector Function.

Inputs	
Name	Connection
Signal	Off, Analog
LoLimitSignal	Off, Analog
HiLimitSignal	Off, Analog

Table 55 Level Detector Inputs.

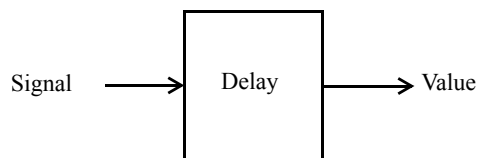
Parameters		
Name	Setting	Unit
LoLimitValue	$0.0 \leq \text{LoLimitValue} \leq \text{HiLimitValue} \leq 100.0$	None
HiLimitValue	$0.0 \leq \text{LoLimitValue} \leq \text{HiLimitValue} \leq 100.0$	None

Table 56 Level Detector Parameters.

Outputs	
Name	Type
Status	Digital

Table 57 Level Detector Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	2	2

Delay

The Delay function has one analog input, *Signal*, and one analog output, *Value*. It also holds one analog parameter, *DelayTime*.

The input *Signal* is transferred to the output *Value* with a delay of *DelayTime* seconds. *DelayTime* can be set to a maximum value of $\text{DelayTime}_{\text{Max}}$ seconds.

The value of $DelayTime_{Max}$ varies with the parameter *SampleTime*:

$$DelayTime_{Max} = 1200 \times SampleTime \text{ seconds}$$

Inputs	
Name	Connection
Signal	Off, Analog

Table 58 Delay Inputs.

Parameters		
Name	Setting	Unit
DelayTime	$0 \leq DelayTime \leq DelayTime_{Max}$	seconds

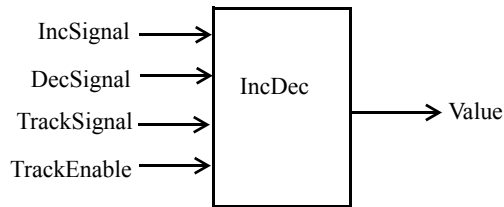
Table 59 Delay Parameters.

Outputs	
Name	Type
Value	Analog

Table 60 Delay Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

IncDec



The IncDec function has two digital inputs, *IncSignal* and *DecSignal*, and one analog output, *Value*. It also holds one analog parameter, *Time*.

The analog output of this block can be controlled by the digital inputs. The parameter *Time* specifies how long it takes for the output *Value* to change by 100%.

As long as *IncSignal* is active the output increases at a rate set by the parameter *Time*. As long as *DecSignal* is active the output decreases at a rate set by the parameter *Time*.

When *TrackEnable* is false, the value of the *Value* output is controlled by the *IncSignal* and the *DecSignal* inputs. When activating *DecSignal*, *Value* will not decrease below *LoLimit*, and when activating *IncSignal*, *Value* will not increase above *HiLimit*.

When *TrackEnable* is true, the value of the *TrackSignal* input is copied to the *Value* output. The *IncSignal* and *DecSignal* inputs are disregarded when *TrackEnable* is true.

The function of *LoLimit* and *HiLimit* always applies.

$$LoLimit \leq Value \leq HiLimit$$



Note: If the inputs (*IncSignal* and *DecSignal*) are equal (both False or both True) the value of the output does not change.

Inputs	
Name	Connection
IncSignal	Off, Digital
DecSignal	Off, Digital
TrackSignal	Analog input
TrackEnable	Digital input

Table 61 IncDec Inputs.

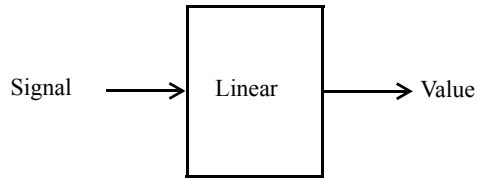
Parameters		
Name	Setting	Unit
Time	$0.1 \leq \text{Time} \leq 1000.0$	seconds/100%
LoLimit	$0.0 \leq \text{LoLimit} \leq \text{HiLimit} \leq 100.0$	%
HiLimit	$0.0 \leq \text{LoLimit} \leq \text{HiLimit} \leq 100.0$	%

Table 62 IncDec Parameters.

Outputs	
Name	Type
Value	$0\% \leq \text{Value} \leq 100\%$

Table 63 IncDec Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Linear

The Linear function can be used to change the characteristic of an analog signal. It has one analog input, *Signal* and one analog output, *Value*. Ten breaking points are available for an accurate characterization. Not all the breaking points have to be used.

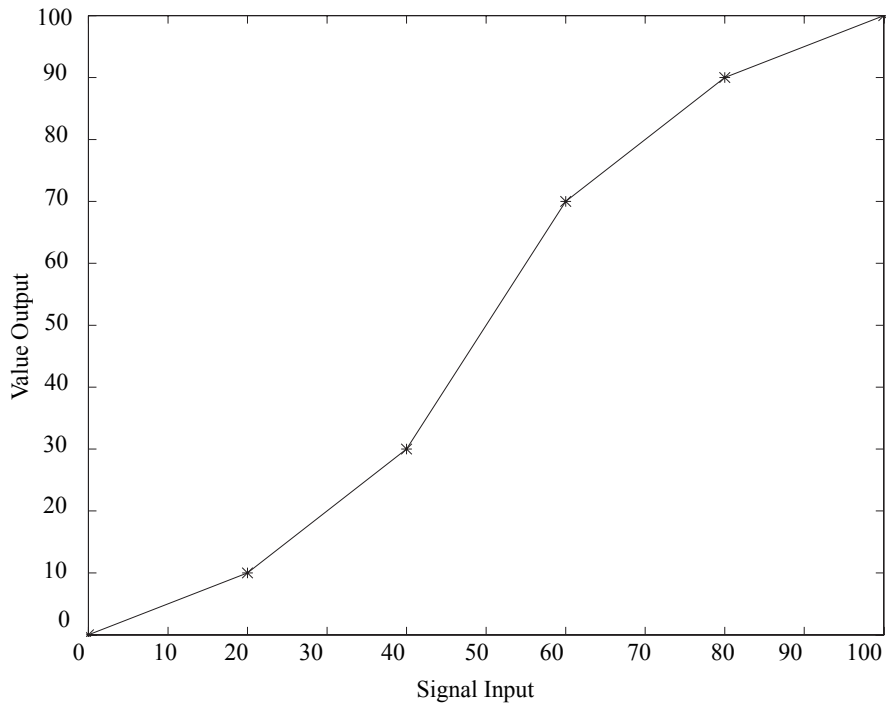


Figure 40 Linear Function.

If $In1 > 0$ (the leftmost end point), and/or $In10 < 100.0$ (the rightmost end point) the function will complete the characteristic so that all values of the input signal (in the range 0.0 to 100.0%) will have defined output values.

In the example below, where $(In1, Out1) = (20.0, 10.0)$, the function extends the characteristic with a horizontal line. If $In10 < 100.0$, the characteristic would also be extended with a horizontal line.

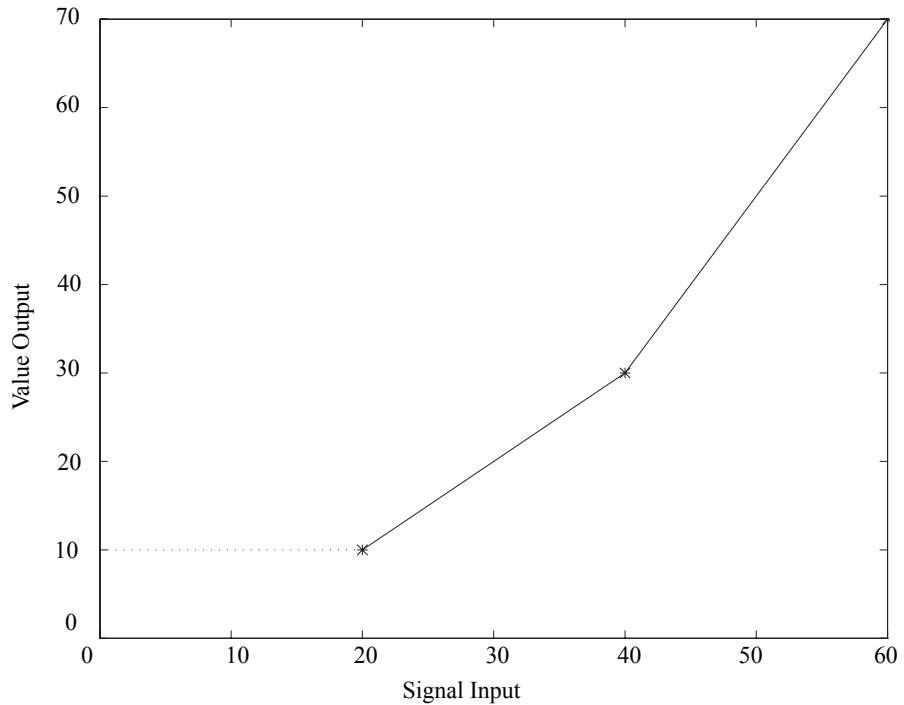


Figure 41 Linear Example.



Note: Only part of the graph is shown in the example above.

It is even possible to have step changes in the breaking points. The output *Value* in the breaking point is defined by the coordinate with the ordinal number. In the example below $Value = 10.0$ for $Signal = 20.0$.

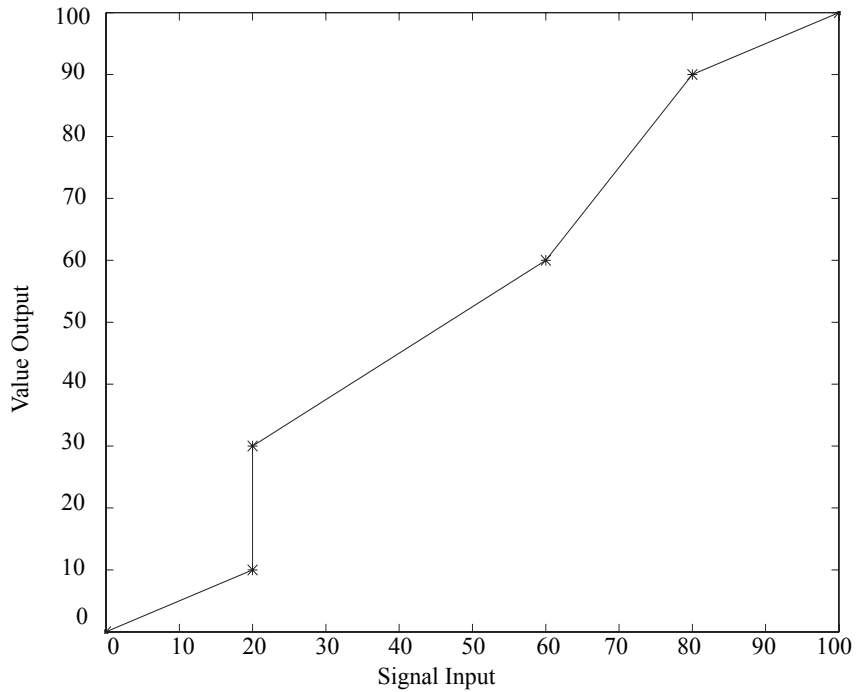


Figure 42 Linear Example with Step Change.



Note: If you set $In_x > In_{x+1}$, then In_{x+1} is adjusted to $In_{x+1} = In_x$. An automatic adjustment is made so that the successive entries are always greater than the preceding ones. For example, suppose $In3 = 15.0$ and $In4 = 20.0$ before you make any changes. When you set $In3 = 30.0$, then automatically $In4 = 30.0$.

Inputs	
Name	Connection
Signal	Off, Analog

Table 64 Linear Inputs

Parameters		
Name	Setting	Unit
In1	$0.0 \leq \text{In1} \leq 100.0$	%
Out1	$0.0 \leq \text{Out1} \leq 100.0$	%
In2	$\text{In1} \leq \text{In2} \leq 100.0$	%
Out2	$0.0 \leq \text{Out2} \leq 100.0$	%
In3	$\text{In2} \leq \text{In3} \leq 100.0$	%
Out3	$0.0 \leq \text{Out3} \leq 100.0$	%
In4	$\text{In3} \leq \text{In4} \leq 100.0$	%
Out4	$0.0 \leq \text{Out4} \leq 100.0$	%
In5	$\text{In4} \leq \text{In5} \leq 100.0$	%
Out5	$0.0 \leq \text{Out5} \leq 100.0$	%
In6	$\text{In5} \leq \text{In6} \leq 100.0$	%
Out6	$0.0 \leq \text{Out6} \leq 100.0$	%
In7	$\text{In6} \leq \text{In7} \leq 100.0$	%
Out7	$0.0 \leq \text{Out7} \leq 100.0$	%
In8	$\text{In7} \leq \text{In8} \leq 100.0$	%
Out8	$0.0 \leq \text{Out8} \leq 100.0$	%
In9	$\text{In8} \leq \text{In9} \leq 100.0$	%
Out9	$0.0 \leq \text{Out9} \leq 100.0$	%
In10	$\text{In9} \leq \text{In10} \leq 100.0$	%
Out10	$0.0 \leq \text{Out10} \leq 100.0$	%

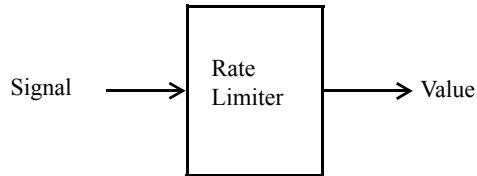
Table 65 Linear Parameters.

Outputs	
Name	Type
Value	Analog

Table 66 Linear Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	X

Rate Limiter



The Rate Limiter function limits the rate of change of the input *Signal* connected to the block. The parameter *Rate* determines the limit.

If *Signal* has a rate of change faster than *Rate* then the output *Value* is in the form of ramp with slope *Rate*.

Inputs	
Name	Connection
Signal	Off, Analog

Table 67 Rate Limiter Inputs

Parameters		
Name	Setting	Unit
Rate	$0.1 \leq \text{Rate} \leq 100.0$	%/second

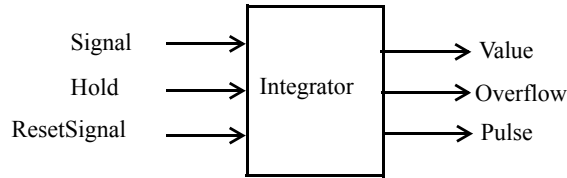
Table 68 Rate Limiter Parameters.

Outputs	
Name	Type
Value	Analog

Table 69 Rate Limiter Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

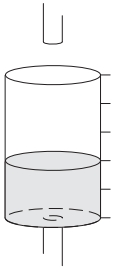
Integrator



This block integrates the input signal over time and produces pulses, the number of which are in direct proportion to the integral. By the means of two inputs, the block can be halted or reset.

Tank Analogy

The integrator block functions like a cylindrical tank standing on one end. It has one inlet and one outlet. On the side of the tank there are gradations. The space between any two gradations is fixed.



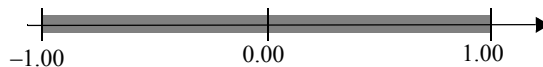
A flow from the inlet makes the water level rise, and pass the gradation marks, one after the other. If instead the bottom outlet is opened, the level will decrease and on its way down, pass the gradation marks.

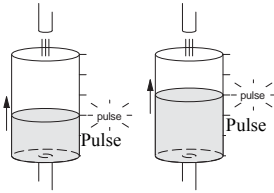
Each time the water level passes a gradation mark on its way up, a pulse is generated. Another pulse is generated when the tank is full.

Signal

Via the input *Signal*, the block can receive values in the range of

$$-1.00 \leq Signal \leq 1.00$$



**Pulse**

The output *Pulse* generates pulses. If the value of *Signal* is great, the pulses come often, and if the value of *Signal* is small, the pulses come seldomly. If the value of *Signal* is less than or equal to zero, no pulses are generated. An internal pulse counter (IPC) keeps track of how many pulses have been generated.

Overflow and OverflowLimit

By setting the parameter *OverflowLimit*, the user can determine how many pulses are generated before the *Overflow* pulse is generated. (The parameter *OverflowLimit* determines the height of the tank.) The relationship between IPC and *OverflowLimit* is:

$$0 \leq \text{IPC} \leq \text{OverflowLimit}$$

When $\text{IPC} = \text{OverflowLimit}$, the *Overflow* pulse is generated.

Pulse Width

The pulse width (of both *Pulse* and *Overflow*) is always equal to the parameter *SampleTime*.

Time and TimeUnit

The parameter *Time* defines the period of time between two consecutive pulses at a *Signal* of 1.00. The parameter *TimeUnit* determines the unit of *Time*: seconds or minutes. (The parameter *Time* corresponds to the radius of the tank. The greater the radius, the slower the water level will rise.)

The permissible range of *Time* is as follows:

$$\text{Time}_{\min} \leq \text{Time} \leq \text{Time}_{\max}$$

Time_{\max} always has the same value

$$\text{Time}_{\max} = 86\,400 \text{ seconds} = 1\,440 \text{ minutes} = 24 \text{ hours} = 1 \text{ day}$$

...while the setting of *SampleTime* affects Time_{\min}

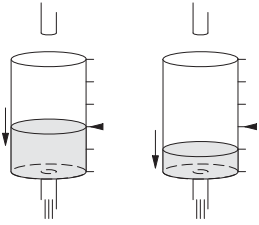
$$\text{Time}_{\min} = 3 \times \text{SampleTime} \text{ seconds} = \frac{3}{60} \times \text{SampleTime} \text{ minutes}$$

Value

Value represents the accumulated value of *Signal* over time. *Value* will increase from 0.00 to 1.00 at a *Signal* of 1.00 in $\text{Time} \times \text{OverflowLimit} \text{ TimeUnit}$ (*Value* is the level in the tank measured in percent of full tank).

Example:

If $\text{TimeUnit} = \text{seconds}$, $\text{Time} = 6.00$ and $\text{OverflowLimit} = 10$, then it will take 60.00 (6.00×10) seconds for *Value* to increase from 0.00 to 1.00 at *Signal* of 1.00.

**Sub-zero Signals**

When *Signal* starts to drop below zero, the value of IPC is stored. As long as *Signal* is negative (or zero), no pulses are generated by the block, but IPC decreases with each passing of a gradation mark.

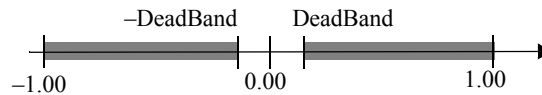
When *Signal* turns positive again, no pulses are generated until IPC reaches the value stored.

Value increases with positive signals and decreases with negative signals.

DeadBand

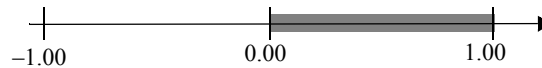
It is possible to make the block disregard *Signals* less than a user defined value by setting the parameter *DeadBand*. The range of *Signal* is then restricted to

$$DeadBand \leq |Signal| \leq 1.00$$

**PositiveOnly**

By setting the parameter *PositiveOnly* to On, the range is restricted to

$$0.0 \leq Signal \leq 1.00$$

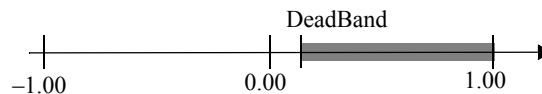


IPC and *Value* remain unchanged as long as *Signal* is less than or equal to zero. When *Signal* rises above zero again, both IPC and *Value* start to increase.

PositiveOnly and DeadBand

The parameters *PositiveOnly* and *DeadBand* can be combined. Then the range is restricted to

$$DeadBand \leq Signal \leq 1.00$$



Hold and Reset

By activating the digital input *Hold*, the block will behave as if *Signal* momentarily dropped to zero. No *Pulses* will be generated, and the *Value* output will freeze. Also, IPC is unchanged while *Hold* is active.

By selecting the parameter *Reset* and pressing OK, the user can reset the block. Activating the digital input *ResetSignal* has the same effect. Resetting the block essentially means setting *Value* = 0.00 and IPC = 0. (*Reset* empties the tank.)

Hold has lower priority than *Reset*.

Inputs	
Name	Connection
Signal	Analog input
Hold	Digital input
ResetSignal	Digital input

Table 70 Integrator Inputs.

Parameters		
Name	Setting	Unit
TimeUnit	seconds, minutes	
Time	$\text{Time}_{\min} \leq \text{Time} \leq \text{Time}_{\max}$	seconds
OverflowLimit	$1 \leq \text{OverflowLimit} \leq 999\ 999$	none
DeadBand	$0.0 \leq \text{DeadBand} \leq 100.0$	%
PositiveOnly	Off, On	none
Reset	Resets IPC and <i>Value</i>	none

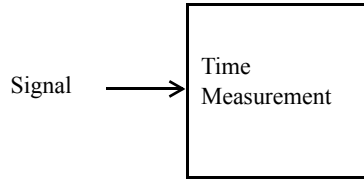
Table 71 Integrator Parameters.

Outputs	
Name	Type
Value	Analog output
Overflow	Digital output
Pulse	Digital output

Table 72 Integrator Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Time Measurement



The Time Measurement block measures the time that the input *Signal* has been active. The block measures time with a resolution of one second. The elapsed time is presented on the display in the form

hours : minutes : seconds.

The internal counter can be reset to zero by selecting *Reset* and pressing the OK key.

When the block is active and selected from the Operator Option List the time is shown on the controller display as below.

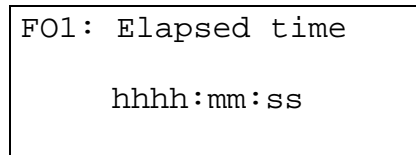


Figure 43 Time Measurement Display.

Inputs	
Name	Connection
Signal	Off, Digital

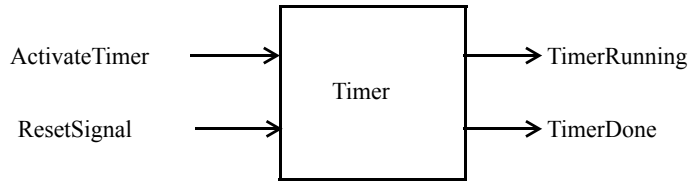
Table 73 Time Measurement Inputs.

Parameters		
Name	Setting	Unit
Reset	OK	None

Table 74 Time Measurement Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Timer



This block provides functions to set the time, to start, to hold and reset a timer. The time can be set to a maximum of 86,400 seconds (1440 minutes = 24 hours). The time unit is seconds or minutes.

The timer is activated by the digital signal *ActivateTimer* and reset by the digital signal *ResetSignal* or from the configuration menu.

The output *TimerRunning* is 1 (running) if the timer function is running and is 0 (not running) if *ActivateTimer* is 0 (not active) or *TimerDone* is 1 (active).

When the preset time has elapsed (equal the parameter *TimerTime*), then *TimerDone* is set to 1 and *TimeRunning* is reset to 0.

When the block is active and selected from the Operator Option List the time is shown on the operator’s display as below.

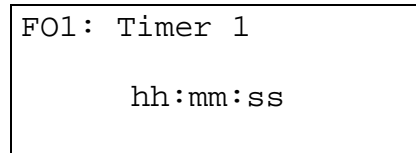


Figure 44 Timer Display.

Inputs	
Name	Connection
ActivateTimer	Off, Digital
ResetSignal	Off, Digital

Table 75 Timer Inputs.

Parameters		
Name	Setting	Unit
TimeUnit	Seconds or Minutes	
TimerTime	$3 \times \text{SampleTime} \leq \text{TimerTime} \leq 86400$	Seconds

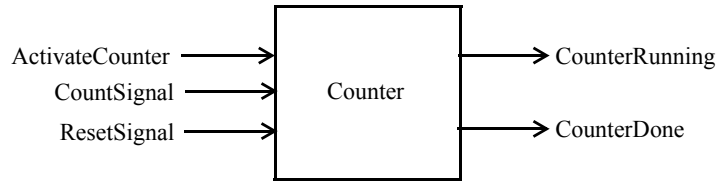
Table 76 Timer Parameters.

Outputs	
Name	Type
TimerRunning	Digital
TimerDone	Digital

Table 77 Timer Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Counter



The block is used to count pulses. When the preset number of pulses on the input *CountSignal* (the value in the parameter *NbrOfPulses*) is reached, the output *CounterDone* is set to 1.

The counter is activated by the digital signal *ActivateCounter* and reset by the digital signal *ResetSignal* or from the configuration menu.

The output *CounterRunning* is 1 (running) if the counter function is running and is 0 (not running) if *ActivateCounter* is 0 (not active) or *CounterDone* is 1 (active).

CounterDone is set to 1 and *CounterRunning* is reset to 0 when the the preset number of pulses is reached.

When the block is active and selected from the Operator Option List the number of pulses counted is shown on the operator's display as below.

F01: Counter 1

23427

Figure 45 Counter Display.

Inputs	
Name	Connection
ActivateCounter	Off, Digital
CountSignal	Off, Digital
ResetSignal	Off, Digital

Table 78 Counter Inputs.

Parameters		
Name	Setting	Unit
NbrOfPulses	$1 \leq \text{NbrOfPulses} \leq 999\,999$	None

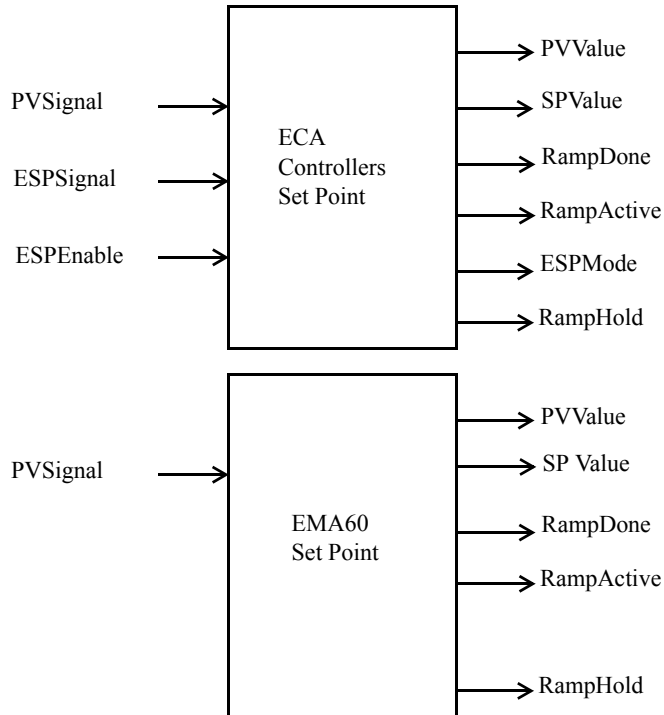
Table 79 Counter Parameters.

Outputs	
Name	Type
CounterRunning	Digital
CounterDone	Digital

Table 80 Counter Outputs.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

Set Point Block



The Set Point block (SP) processes the Set Point data. The Control Block is permanently connected to the Set Point block.



Notes:

- The ECA600 contains two Control blocks and therefore has two Set Point blocks.
- The EMA60 Set Point block does not have an *ESPEnable* input, an *ESPSignal* and associated parameters, an *ESPMode* output or a Simple Set Point Ramp.

- In the text and tables below, any features which are not available in the EMA60 are identified by a “*” character.

Inputs	
Name	Connection
PVSignal	Off, Analog
ESPSignal*	Off, Analog
ESPEnable*	Off, Digital

Table 81 Set Point Block Inputs.

Parameters		
Name	Setting	Unit
SPLimitMode *	Off, On	None
SPLoLimit *	MinValue ≤ SPLoLimit ≤ SPHiLimit ≤ MaxValue	None
SPHiLimit *	MinValue ≤ SPLoLimit ≤ SPHiLimit ≤ MaxValue	None
SP TrackMode *	Off, TrackESP, TrackPVInNonAuto	None
ESPLock *	Off, On	None

Table 82 Set Point Block Parameters.

Outputs	
Name	Type
PVValue	$0 \leq PVValue \leq 100$
SPValue	$0 \leq SPValue \leq 100$
ESPMode *	Digital
RampDone	Digital
RampActive	Digital
RampHold	Digital

Table 83 Set Point Block Outputs

Process Value

PVSignal should be connected to the process value. This is usually an Analog Input block (AI), but can also be a Function block (FA, FO) or an Analog Input Communication Buffer block (AIC).

Set Point

There are two different Set Points:

- ISP – Internal Set point (set via the front panel).
- ESP – External Set Point (block input).

The Set Point Block has two analog inputs, *PVSignal* and *ESPSignal* *, and one digital input, *ESPEnable* *. The block has two analog outputs, *PVValue* and *SPValue* and four digital outputs, *RampDone*, *RampActive*, *ESPMode* * and *RampHold*.

ESP

The *ESPSignal* input to the block allows you to specify an External Set Point from different analog sources for use with the controller. Depending on the analog source type, different indicators are displayed on the top row of the LCD display.

Source Type	External Set Point Type	Indicator
AI (analog input)	Remote Set Point	RSP
AIC (analog communication)	Computer Set Point	CSP
AU (analog user)	Local Set Point	LSP
Other than AI, AIC, AU	External Set Point	ESP

Table 84 Set Point Indicators

If the input *ESPSignal* = *Off*, the system disregards all External Set Point parameters completely:

- the system will use ISP for Set Point,
- the setting of the *ESPLock* parameter is disregarded,
- the value of the *ESPStatus* output is invalid (if any inputs are connected to this output, the SP block will signal a warning on the display),
- the ‘External Set Point’ and ‘ISP/ESP’ entries are removed from the Operator Option List.

The object of the *ESPLock* parameter is to lock the controller in ESP mode. Neither the operator nor the *ESPEnable* input must be able to switch into ISP mode.

When *ESPLock* is On, the Set Point block permanently uses *ESPSignal* for Set Point. The input *ESPEnable* is deactivated. Furthermore, the entries ‘Internal Set Point’ and ‘ISP/ESP’ are removed from the Operator Option List, thus preventing the operator from switching into ISP mode.

ESPEnable *

The *ESPEnable* input to the block enables switching between ESP and ISP and can be connected to any digital source. Whenever the *ESPEnable* input is connected to a source switching between ESP and ISP can be made either from the front panel of the controller or by using the *ESPEnable* input.



Note: If the *ESPEnable* input is set to Off, switching between ESP and ISP can only be made from the front panel. Refer to the Operator's Manual, section 4.5.

ESPEnable and the front switch affect ESPMode as shown in the diagram below.

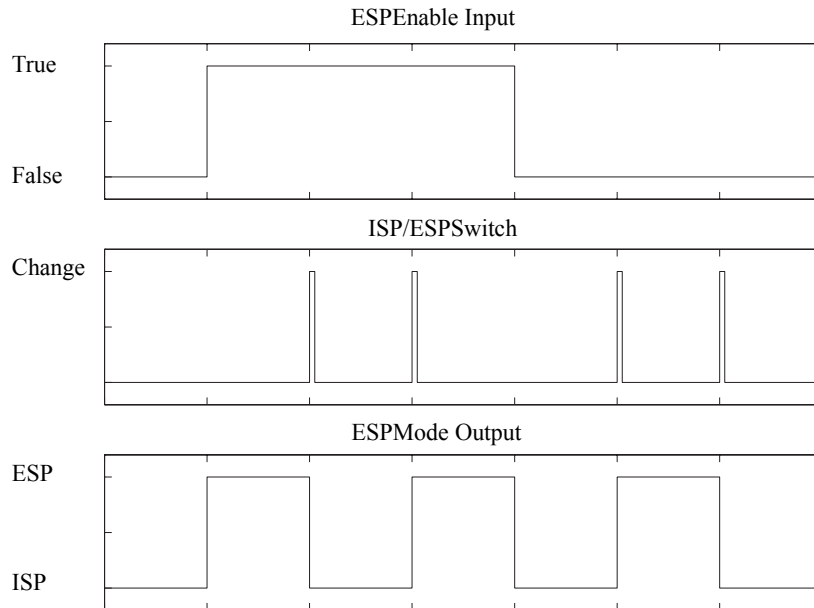


Figure 46 ISP/ESP Switching.

SPValue *

The *SPValue* output from the block is the Set Point used by the Control block. The Control Block has a fixed connection to the *Set Point* output, but other blocks are also allowed to connect to this output.

SPLimitMode *

The parameter *SPLimit Mode* allows you to specify a limitation which ensures that the Set Point lies within an acceptable range, defined by setting upper and lower limits via the parameters *SPLoLimit* and *SPHiLimit*.

For example, setting *SPLoLimit* to 2.00m³/min and *SPHiLimit* to 7.00m³/min limits the Set Point to values within the range $2.00 < SP < 7.00$. The limits are indicated on the SP bar graph by brighter segments. It is not possible to set these parameters beyond the following limitations:

- *SPLoLimit* > *ISP*
- *SPHiLimit* < *ISP*
- *SPLoLimit* > *ESP*
- *SPHiLimit* < *ESP*

Set Point Tracking *

The internal set point can track another value by means of the parameter *SPTrackMode*. There are two methods of tracking available.

If TrackESP is configured the value of the signal is copied to ISP. TrackESP cannot be configured if *ESPSignal* is set to Off; if it is set to Off after TrackESP has been configured, the result is the same as if no tracking had been configured.

If the tracking is configured to TrackPVInNonAuto then ISP tracks the Process Value whenever the control mode is not Auto (i.e. Manual, Tracking, Forced Output, Tuning).

Set Point Ramps

There are two kinds of Set Point ramps:

- Simple *
- Advanced

The following parameters apply to both the Simple Set Point Ramp and the Advanced Set Point Ramp.

Parameters		
Name	Setting	Unit
RampMode	Off, Simple, Advanced	None
Hold	Off, On	None
TimeUnit	Minutes, Seconds	None
RampStart	StartFromPV, StartFromStartValue	None
StartValue	$\text{MinValue} \leq \text{StartValue} \leq \text{MaxValue}$	TimeUnit

Table 85 Set Point Ramp Parameters.

Ramp Start Condition

The following Control Mode transitions, which represent a change from False to True on the *AutoStatus* input, start the Ramp function:

- Manual → Auto
- Forced Output → Auto
- Tracking → Auto

The following Control Mode transitions which represent a change from True to False on the *AutoStatus* input, stop the Ramp function:

- Auto → Manual
- Auto → Forced Output
- Auto → Tracking



Note: The Tracking modes InAutoWDeltaU and InAutoBumpless are both treated as Auto modes by the Control block and the Set Point block. Therefore the Simple Set Point Ramp is triggered according to the table below. The term TrackMode in the table is used for either TrackOneMode or TrackTwoMode.

TrackMode	Triggers Simple Set Point Ramp
InAutoWDeltaU	No
WhenEnabled	Yes
WhenEnabledWDeltaU	No
InAutoBumpless	Yes

Table 86 Simple Set Point Ramp Triggering*.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	-

Simple Set Point Ramp *

Parameters		
Name	Setting	Unit
RampSpeed	$\text{RampSpeed}_{\text{Min}} \leq \text{RampSpeed} \leq \text{RampSpeed}_{\text{Max}}$	TimeUnit
TimerTime	$0 < \text{Timertime} < \text{TimerTime}_{\text{Max}}$	TimeUnit

Table 87 Simple Set Point Ramp Parameters.



Note: $\text{TimerTime}_{\text{Max}}$ always represents 24 hours. When TimeUnit is set to Minutes, $\text{TimerTime}_{\text{Max}} = 1440$ and when TimeUnit is set to Seconds, $\text{TimerTime}_{\text{Max}} = 86400$.

The Set Point block is permanently connected to the controller Output Configuration Block via the digital signal *AutoStatus*. When the controller Output Configuration Block is in Auto Control mode the Auto Status Output is active.



Note: The parameter *RampSpeed* is entered in Engineering Units per TimeUnit. This means that the setting of this parameter depends on the setting of the *MinValue* and *MaxValue* parameters in the Operator block.

The difference between *MaxValue* and *MinValue* parameters makes the span. The value of span determines the resolution of *RampSpeed*.

$$\text{span} = \text{MaxValue} - \text{MinValue}$$

The lowest speed makes the Set Point travel the whole span in 86 400 seconds (24 hours). $RampSpeed_{Min}$ is hence defined by (if TimeUnit = seconds):

$$RampSpeed_{Min} = \frac{span}{86400} \quad \text{Unit/second}$$

or if TimeUnit = minutes:

$$RampSpeed_{Min} = \frac{span}{1440} \quad \text{Unit/minute}$$

The highest speed makes the Set Point travel the whole span in one second. $RampSpeed_{Max}$ is hence defined by (if TimeUnit = seconds):

$$RampSpeed_{Max} = \frac{span}{1} \quad \text{Unit/second}$$

or if TimeUnit = Minutes:

$$RampSpeed_{Max} = span \times 60 \quad \text{Unit/minute}$$

The $RampSpeed$ parameter uses the same number of decimals as defined by the parameter $NumberOfDecimals$ in the Operator Block.

Example

If $MinValue = 10.0$, $MaxValue = 210.0$, $Unit = ^\circ\text{C}$ and $TimeUnit = \text{Minutes}$ then:

$$span = 210.0 - 10.0 = 200.0$$

and:

$$RampSpeed_{Min} = \frac{200}{1440} = 0.14 \text{ } ^\circ\text{C/minute}$$

$$RampSpeed_{Max} = 200.0 \times 60 = 12000 \text{ } ^\circ\text{C/minute}$$

The $RampSpeed$ parameter in this example can be set between 0.14 ($RampSpeed_{Min}$) and 12000 ($RampSpeed_{Max}$).

Hold

Both the Simple and Advanced Set Point Ramp functions offer a hold feature. It is activated by setting the parameter *Hold* to On. As described below, the ramps are activated by a Non-Auto to Auto transition. When the ramp is running (*RampDone* = False) an Auto to Non-Auto transition can either reset the ramp and put it on hold (*Hold* = Off), or just put it on hold (*Hold* = On). A subsequent Non-Auto to Auto transition will then either start the ramp from the beginning (*Hold* = Off) or make it continue from where it was interrupted (*Hold* = On). The function of the *RampHold* output may be summarized:

$$\text{RampHold} = (\text{Hold} = \text{On}) \text{ OR } \text{ManualStatus} \text{ OR } \text{NOT RampDone}$$

As long as the value of *SPValue* is changing the output *RampActive* is True; while *SPValue* is constant *RampActive* is False.

The value of *SPValue* must be equal to the value of ISP/ESP for a period of time defined by the parameter *TimerTime* before the ramp is complete and the SP Block output *RampDone* is set to True.

The time unit used for *TimerTime* is set by the parameter *TimeUnit*.

The figure below shows the timing and function of a typical Simple Set Point Ramp.

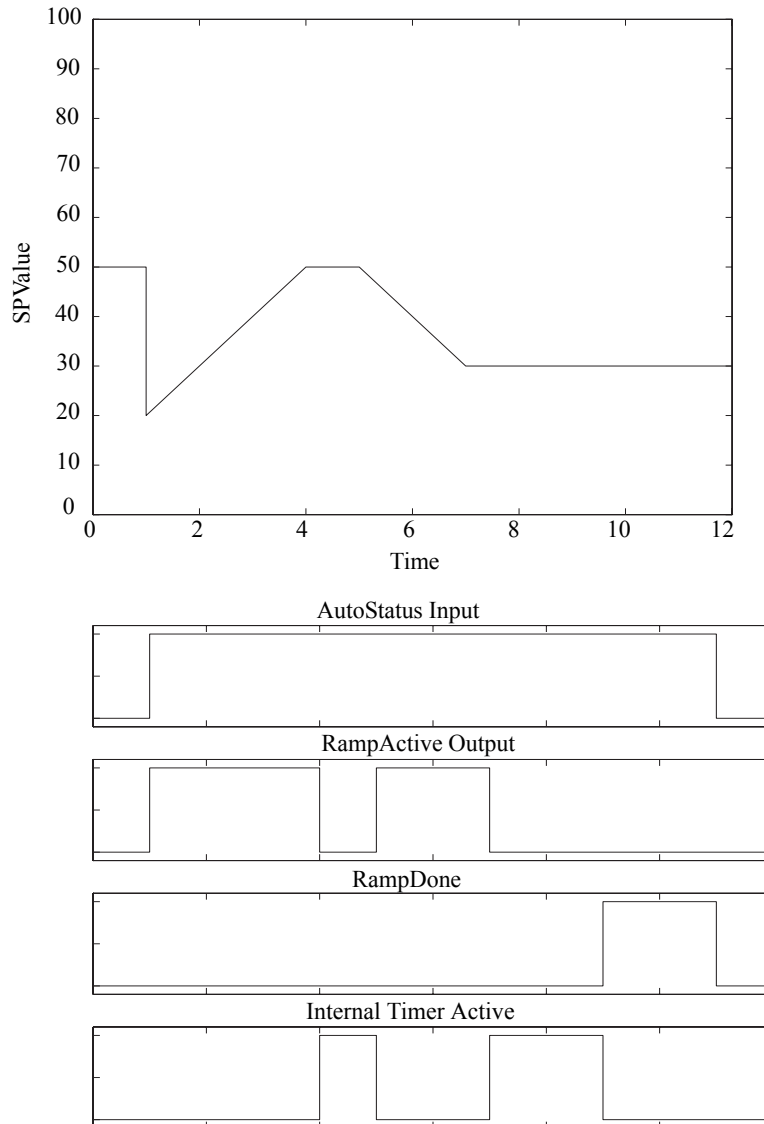


Figure 47 Simple Set Point Ramp Timing and Function Example.

At $t = 1$ the control Mode changes from non Auto (Manual, Tracking, or Forced Output) to Auto and the ramp starts.

In the example above with $TimerTime = 2$.

The time $5 - (\text{minus}) 4 < TimerTime$ and therefore the ramp is not complete.

The time $9 - (\text{minus}) 7 \geq \text{TimerTime}$ and therefore the ramp is complete.



Note: It is not possible to toggle between ISP and ESP when *RampDone* is False.

If *ESPEnable* = True and *RampMode* = On then ESP is the target value of the Simple Set Point Ramp.

If *Tracking* = TrackESP and *RampDone* = True, then the value of ESP is copied over to ISP.

Whenever the control Mode is Manual then the *RampDone* output is True.

Here is how the simple ramp function works when the Set Point Ramp function is active:

- 1 The Control Mode is Auto (*RampActive* = False, *RampDone* = True).
- 2 The Control Mode changes to a non-Auto Mode (usually Manual). When this happens, one segment of the Set Point bar graph is highlighted to show that it is possible to adjust the Set Point Ramp Target Value. The position of this highlighted segment indicates the last target value used.
- 3 The user presses the Next key to make Ramp Set Point the currently selected signal.
- 4 The user presses the UP/Down keys, and sets the target value of the Set Point Ramp. (The actual Set Point remains unchanged, as does the Set Point Bar Graph). The highlighted segment changes its position as the user changes the target value. The target value is also presented in numerical form on the display.
- 5 The Control Mode is changed from non-Auto (usually Manual) to Auto (*RampActive* = True, *RampDone* = False).
- 6 The ramp block memorizes the target value of the Set Point.
- 7 The Set Point is set to the value of the Process Value, or set to the value of the *StartValue* parameter depending on the setting of the setting of the *RampStart* parameter. The Set Point is ramped from this value to the value memorized at a rate of change defined by the parameter *RampSpeed*.
- 8 When the ramp is active, the user can change the target value by pressing the Next key to make Ramp Set Point the currently selected signal and then pressing the Up/Down pushbuttons. This also changes the highlighted segment on the bar graph.

- 9 When the ramp reaches the target value, it stops (*RampActive* = False) and the internal timer starts.
- 10 If the target value changes, the timer is reset, and *RampActive* = True during the change.
- 11 When the timer has run uninterrupted for a period equal to the time defined by the parameter *TimerTime*, the highlighted segment is turned Off (*RampDone* = True).



Notes:

- The values of the outputs *RampActive* and *RampDone* change according to the text above. On the lines where there is no text enclosed in parenthesis, the value of these outputs remain unchanged.
- Changing the controller mode from Auto to non-Auto when the ramp is active terminates the ramp and freezes the Set Point at the current value. At the same time *RampActive* is set to False, and *RampDone* is set to True.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	X

Advanced Set Point Ramp

The Advanced Set Point Ramp function is only available on the ECA600 and EMA60 controllers.

The Advanced Set Point Ramp block has two digital outputs, *RampDone* and *RampActive*. The block holds nine parameters, *RampMode*, *TimeUnit*, *RampSpeed*, *RampStart* and *StartValue*, and five *Time* and *Value* parameters.

The Advanced Set Point Ramp works as follows:

- 1 The Control Mode is non-Auto (usually Manual) (*RampActive* = False, *RampDone* = True).
- 2 The Control Mode is changed to Auto (*RampActive* = True, *RampDone* = False).
- 3 The Set Point is set to the value of the Process Value, or set to the value of the *StartValue* parameter depending on the setting of the setting of the *RampStart* parameter.
- 4 If the Advanced Set Point Ramp is configured to start from the Process Value, the rate of change of the Set Point is the same as the one defined by the *StartValue* and the first Breaking Point.

- 5 When the ramp reaches the first Breaking Point, it immediately starts to ramp towards to second Breaking Point, and so on.



Notes:

- Pressing the Up/Down keys while *RampDone* is False has no effect.
- The values of the outputs *RampActive* and *RampDone* change according to the text above. On the lines where there is no text enclosed in parenthesis, the value of these outputs remain unchanged.
- The output *RampDone* is True whenever the controller is in Manual Mode.
- Changing the controller mode from Auto to non-Auto when the ramp is active, terminates the ramp and freezes the Set Point at the current value. At the same time *RampActive* is set to False, and *RampDone* is set to True.

Parameters		
Name	Setting	Unit
Time1	$0 \leq \text{Time1} \leq 10000$	TimeUnit
Value1	$\text{MinValue} \leq \text{Value1} \leq \text{MaxValue}$	None
Time2	$0 \leq \text{Time2} \leq 1440$	TimeUnit
Value2	$\text{MinValue} \leq \text{Value2} \leq \text{MaxValue}$	None
Time3	$0 \leq \text{Time3} \leq 1440$	TimeUnit
Value3	$\text{MinValue} \leq \text{Value3} \leq \text{MaxValue}$	None
Time4	$0 \leq \text{Time4} \leq 1440$	TimeUnit
Value4	$\text{MinValue} \leq \text{Value4} \leq \text{MaxValue}$	None
Time5	$0 \leq \text{Time5} \leq 1440$	TimeUnit
Value5	$\text{MinValue} \leq \text{Value5} \leq \text{MaxValue}$	None

Table 88 Advanced Set Point Ramp Parameters.



Note: *MinValue* and *MaxValue* are set in the Operator block.

Note: $\text{TimerTime}_{\text{Max}}$ always represents 24 hours. When *TimeUnit* is set to Minutes, $\text{TimerTime}_{\text{Max}} = 1440$ and when *TimeUnit* is set to Seconds, $\text{TimerTime}_{\text{Max}} = 86400$.

Advanced Set Point Ramp Example

In the example below the *Time* and *Value* parameters are set according to the table below.



Note: Each *Time* parameter is relative to the previous *Time* parameter. All *Value* parameters are absolute.

Name	Value
StartValue	0.0
Time1	2
Value1	20
Time2	2
Value2	30
Time3	1
Value3	60
Time4	2
Value4	70
Time5	5
Value5	50

Table 89 Advanced Set Point Ramp Example.

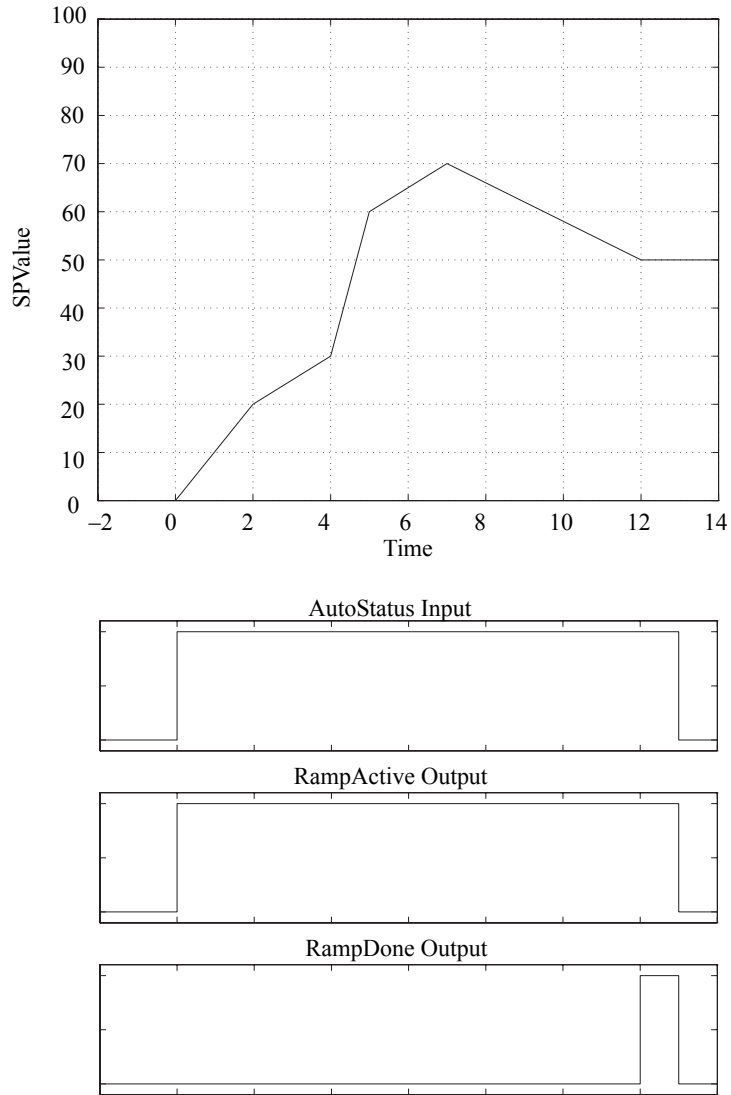


Figure 48 Advanced Set Point Ramp Example.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

Alarms

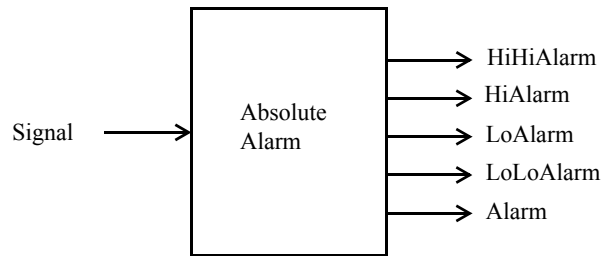
The Alarm blocks holds alarm limits and generates alarm signals when these limits are transgressed.



Note: The ECA600 is configurable as PID1/Master or as PID2/Slave. The functions described here are duplicated for PID1 and PID2.

Available in			
ECA06	ECA60	ECA600	EMA60
1	1	2	3

Absolute Alarm Block



This block handles Absolute Alarms (AA). The outputs of this block are activated when the input *Signal* reaches predefined limits. Four parameters define the alarm limits – *LoLimit*, *HiLimit*, *LoLoLimit* and *HiHiLimit*.

For each alarm limit there is a digital output which indicates if that limit has been reached – *LoAlarm*, *HiAlarm*, *LoLoAlarm* and *HiHiAlarm*. There is also one output which is active if any alarm is active – *Alarm*.

When *Signal* is between *LoLimit* and *HiLimit*, no outputs are active. If *Signal* decreases and reaches *LoLimit* the corresponding *LoAlarm* output is activated. If signal decreases further and reaches *LoLoLimit* then the *LoLoAlarm* output is also activated. If *Signal* increases and reaches *HiLimit* the corresponding *HiAlarm* output is activated. If signal increases further and reaches *HiHiLimit* then the *HiHiAlarm* output is also activated. The alarm outputs may be summarized:

$$HiHiAlarm = Signal > HiHiLimit$$

$$HiAlarm = Signal > HiLimit$$

$$LoAlarm = Signal < LoLimit$$

$$LoLoAlarm = Signal < LoLoLimit$$

$$Alarm = HiHiAlarm \text{ OR } HiAlarm \text{ OR } LoAlarm \text{ OR } LoLoAlarm$$

Disabling Selected Alarms

You may selectively disable these alarms as follows:

LoLoAlarm: Set the *LoLoLimit* to 0.0.

LoAlarm and *LoLoAlarm*: Set the *LoLimit* and the *LoLoLimit* to 0.0.

HiHiAlarm: Set the *HiHiLimit* to 100.0.

HiHiAlarm and *HiAlarm*: Set the *HiLimit* and the *HiHiLimit* to 100.0.

Inputs	
Name	Connection
Signal	Analog

Table 90 Absolute Alarm Inputs

Parameters		
Name	Setting	Unit
Mode	Off, On	None
LoLimit	$0.0 \leq \text{LoLoLimit} \leq \text{LoLimit} \leq \text{HiLimit} \leq \text{HiHiLimit} \leq 100.0$	%
HiLimit	$0.0 \leq \text{LoLoLimit} \leq \text{LoLimit} \leq \text{HiLimit} \leq \text{HiHiLimit} \leq 100.0$	%
LoLoLimit	$0.0 \leq \text{LoLoLimit} \leq \text{LoLimit} \leq \text{HiLimit} \leq \text{HiHiLimit} \leq 100.0$	%
HiHiLimit	$0.0 \leq \text{LoLoLimit} \leq \text{LoLimit} \leq \text{HiLimit} \leq \text{HiHiLimit} \leq 100.0$	%
Identity	User defined text	None

Table 91 Absolute Alarm Parameters.

Outputs	
Name	Type
LoLo Alarm	Digital
Lo Alarm	Digital
Hi Alarm	Digital
HiHiAlarm	Digital
Alarm	Digital

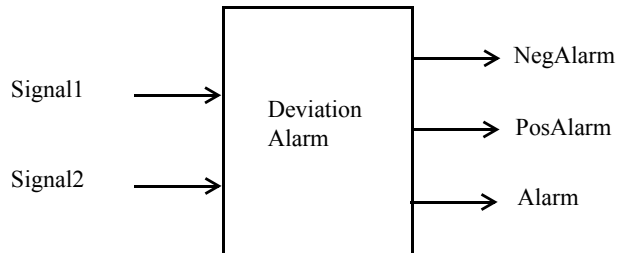
Table 92 Absolute Alarm Outputs.



Note: The Absolute Alarm block is freely configurable.

Available in			
ECA06	ECA60	ECA600	EMA60
-	1	2	3

Deviation Alarm Block



The Deviation Alarm block handles relative alarms. The block observes the difference between the *Signal1* and *Signal2* input values:

$$Diff = Signal2 - Signal1$$

The parameters *NegDeviationLimit* and *PosDeviationLimit* define the alarm limits. For each alarm limit there is a corresponding alarm output – *NegAlarm* and *PosAlarm*. The *Alarm* output is active if any alarm output is active. When *Diff* is between *NegDeviationLimit* and *PosDeviationLimit* no output is active. If *Diff* decreases and reaches *NegDeviationLimit* the corresponding *NegAlarm* output is activated. If *Diff* increases and reaches *PosDeviationLimit* the corresponding *PosAlarm* output is activated. The deviation alarms may be summarized:

$$PosAlarm = Diff > PosDeviationLimit$$

$$NegAlarm = Diff < NegDeviationLimit$$

$$Alarm = PosAlarm \text{ OR } NegAlarm.$$

Disabling Alarms

You may selectively disable the deviation alarms as follows:

NegAlarm: Set *NegDeviationLimit* to -100.0 .

PosAlarm: Set *PosDeviationLimit* to 100.0 .

Inputs	
Name	Connection
Signal1	Analog
Signal2	Analog

Table 93 Deviation Alarm Inputs.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
NegDeviationLimit	$-100.0 \leq NegDeviationLimit \leq 0.0$	%
PosDeviationLimit	$0.0 \leq PosDeviationLimit \leq 100.0$	%
Identity	User defined text	None

Table 94 Deviation Alarm Parameters.

Outputs	
Name	Type
NegAlarm	Digital
PosAlarm	Digital
Alarm	Digital

Table 95 Deviation Alarm Outputs.



Note: The Deviation Alarm block is freely configurable.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	-

Control

Combination

Setting combination to Cascade or DualLoop opens a second Control block for configuration. Select Cascade when you want to connect the Control blocks in a Master/Slave configuration. Select DualLoop when you want the Control blocks to work independently.

Parameters		
Name	Setting	Unit
Combination	SingleLoop, Cascade, DualLoop	None

Table 96 Control Parameters



Notes:

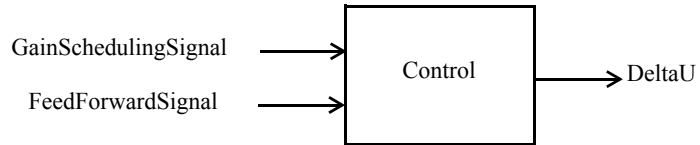
- For information about how to switch between indicated loops, refer to the Operator's Manual, section 4.8.
- The ECA600 comprises two loops. Hence, it has two sets of the parameters described in the following section.

Available in			
ECA06	ECA60	ECA600	EMA60
1	1	2	-

Parameters

The Control block (CO) encapsulates PID control and related functions. The following connections are permanently made to the Set Point block – they cannot be changed:

- PVSignal – Process Value signal.
- SPSignal – Set Point signal.
- ESPMode – indicates whether an internal or an external Set Point is used.



Inputs	
Name	Connection
GainSchedulingSignal	Off, Analog
FeedForwardSignal	Off, Analog

Table 97 Control Inputs.

Outputs	
Name	Type
DeltaU	Analog

Table 98 Control Block Outputs.

The controller can be configured to perform different types of control:

- PID control with fixed parameters. If the process has dynamic characteristics which are independent of the working point (a linear process) and do not change with time, PID control with fixed parameters may be used.
- Gain Scheduling PID control. If the process involves a non-linear valve or other type of non-linear device the process dynamics will change according to the working range. The controller should therefore also have different parameters in different working ranges. If the non-linearity can be related to a reference signal, Gain Scheduling PID control may be used. This can also be combined with adaptive control.
- pPI controller, predicting PI (ECA600 only), which is a dead-time compensating controller. The pPI controller gives short damping-time at a load or set point change. Instead of using the Autotuner, the pPI controller has three parameters to be set: gain, integration time and dead-time. This controller should not be used for integrating processes.
- Feed Forward control. To eliminate a disturbance that could be measured before it affects the Process Value use feed forward via the selected input. In order to eliminate measurable levels of disturbance, adaptive or static feed-forward can be used.
- Adaptive control. The pPI controller cannot be used with adaptive control.

PID control with adaptive PID settings. If the process has variable dynamic characteristics which cannot be related to a reference signal, PID control with adaptive PID settings should be used. (ECA600 only)

The Autotuner function can be used to set the PID parameters for all types of control but not for the pPI controller. If adaptive control is selected, the PID parameters are only used as starting values.

The selected control type can be used with direct or reversed control action.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	-

PID Parameters

The PID parameters are defined by three parameters:

K = controller gain, TI = integral time, TD = derivative time.

Three sets of these parameters are used to facilitate Gain Scheduling as described below. If the TI parameters are set to Off the *Offset* parameter is activated. When TI has a value other than Off the *Offset* parameter is not used.

Parameters		
Name	Setting	Unit
K1	$0.01 \leq K1 \leq 100.0$	None
TI1	$1.0 \leq TI1 \leq 9999.9$, Off	seconds
TD1	$0.0 \leq TD1 \leq 10000.0$	seconds
K2	$0.01 \leq K2 \leq 100.0$	None
TI2	$1.0 \leq TI2 \leq 9999.9$, Off	seconds
TD2	$0.0 \leq TD2 \leq 10000.0$	seconds
K3	$0.01 \leq K3 \leq 100.0$	None
TI3	$1.0 \leq TI3 \leq 9999.9$, Off	seconds
TD3	$0.0 \leq TD3 \leq 10000.0$	seconds
Offset	$0.0 \leq \text{Offset} \leq 100.0$	%

Table 99 PID Parameters.



Note: When Gain Scheduling is disabled and the parameter TI is set to Off, the *Offset* parameter is activated. When the control deviation is 0, the controller output signal is equal to the value of the *Offset* parameter.

The control parameters should be commissioned via the Autotuner function on installation. However the parameters can also be set manually.

The ECA06 has only one set of PID parameters – K1, TI1 and TD1.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	-

Gain Scheduling

Gain Scheduling is used to improve the control of non-linear processes where the non-linearity is fixed, i.e. not time-dependent. Processes with time-dependent non-linearity must be controlled by a PID controller with adaptive PID settings.

In Gain Scheduling the controller uses different PID parameters according to the value of a user selected reference signal, *GainSchedulingSignal*.

Three different settings for PID parameters can be used, within three ranges defined by the user defined parameters *Limit1* and *Limit2* which are expressed as a percentage of *GainSchedulingSignal*. When the value of *GainSchedulingSignal* passes one limit the next set of PID parameters is switched in.

If the Gain Scheduler is reset, the next time the Autotuner is used all three parameter sets in the PID Parameters section will be filled in the same way as when Gain Scheduling was first selected.

The three sets of PID parameters can be inserted into the parameter table in two different ways:

- Via autotuning (normal method).
- Manually.

Inputs	
Name	Connection
GainSchedulingSignal	off, Analog

Table 100 Gain Scheduling Inputs.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
Limit1	$0.0 \leq \text{Limit1} \leq \text{Limit2} \leq 100.0$	%
Limit2	$0.0 \leq \text{Limit1} \leq \text{Limit2} \leq 100.0$	%
Reset	OK	None

Table 101 Gain Scheduling Parameters.

GainSchedulingSignal	PID Parameter Set
$0.0 \leq \text{GainSchedulingSignal} \leq \text{Limit1}$	K1, TI1, TD1
$\text{Limit1} \leq \text{GainSchedulingSignal} \leq \text{Limit2}$	K2, TI2, TD2
$\text{Limit2} \leq \text{GainSchedulingSignal} \leq 100.0$	K3, TI3, TD3

Table 102 Gain Scheduling Parameter Sets

You can set the parameters either manually or by using the Autotuner. If the parameter *Mode* is Off setting a parameter (K, TI or TD) in one set automatically copies the parameter setting to the other sets. This applies to both manual and Autotuner settings.

The input signal *GainSchedulingSignal* together with the parameters *Limit1* and *Limit2* determine which parameter set is to be used by the control algorithm. Depending on which set is selected by *GainSchedulingSignal* the Autotuner adjusts different parameter sets.

The tuning order below is just an example.

When tuning is first performed, or after the Autotuner has been reset, not only the set selected by *GainSchedulingSignal* (here 0.0–Limit1) is filled but also the rest of the table.

GainSchedulingSignal	K	TI	TD
Limit2–100.0	Tuning 1	Tuning 1	Tuning 1
Limit1–Limit2	Tuning 1	Tuning 1	Tuning 1
0.0–Limit1	Tuning 1	Tuning 1	Tuning 1

Table 103 Gain Scheduling First Tuning.

When tuning the second time (and the next set is selected) the first set is left untouched but the one selected (here Limit2–100.0) is filled along with the rest of the table.

GainSchedulingSignal	K	TI	TD
Limit2–100.0	Tuning 2	Tuning 2	Tuning 2
Limit1–Limit2	Tuning 2	Tuning 2	Tuning 2
0.0–Limit1	Tuning 1	Tuning 1	Tuning 1

Table 104 Gain Scheduling Second Tuning.

When tuning the third time (and the remaining set is pointed out) the only set affected is the one pointed out (here Limit1–Limit2).

GainSchedulingSignal	K	TI	TD
Limit2–100.0	Tuning 2	Tuning 2	Tuning 2
Limit1–Limit2	Tuning 3	Tuning 3	Tuning 3
0.0–Limit1	Tuning 1	Tuning 1	Tuning 1

Table 105 Gain Scheduling Third Tuning.

After this, for every tuning performed only the set pointed out is filled.

Note: If you reset the Gain Scheduler, the Autotuner fills in the whole table as described in Table 103, not just the selected parameter set.



Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	-

Feed Forward

The Feed Forward function block has one analog input, *FeedForwardSignal*, and three parameters – *Mode*, *Gain* and *Reset*.

Inputs	
Name	Connection
FeedForwardSignal	off, Analog

Table 106 Feed Forward Inputs.

The parameter *Mode* sets the mode of operation, *Off*, *Static Gain* or *Adaptive Gain*.

The *Gain* parameter can be configured within the range -10.0 to $+10.0$, but is obsolete when Adaptive Feed Forward is used.

When the *Reset* parameter is set to OK the internal parameters for the Adaptive Feed Forward function are reset.

Parameters		
Name	Setting	Unit
Mode	Off, StaticGain, AdaptiveGain	none
Gain	$-10.0 \leq \text{Gain} \leq 10.0$	none
Reset	OK	none

Table 107 Feed Forward Parameters.



Note: If the controller is moved to another control application you must reset the Adaptive Feed Forward.

When using Feed Forward, a term (u_{FF}) is added to the signal produced by the PID algorithm (u_{PID}), and these two make the ControlSignal (u):

$$u = u_{PID} + u_{FF}$$

The term, u_{FF} , is a product where the *FeedForwardSignal* is multiplied by a factor (*Gain*):

$$u_{FF} = \text{FeedForwardSignal} \times \text{Gain}$$

The Feed Forward block in the figure below holds the factor *Gain*. This factor is either static (does not change over time, *Mode* = *StaticGain*), or adaptive (changes with Control Loop dynamics, *Mode* = *AdaptiveGain*). Select the desired behaviour by setting the *Mode* parameter.

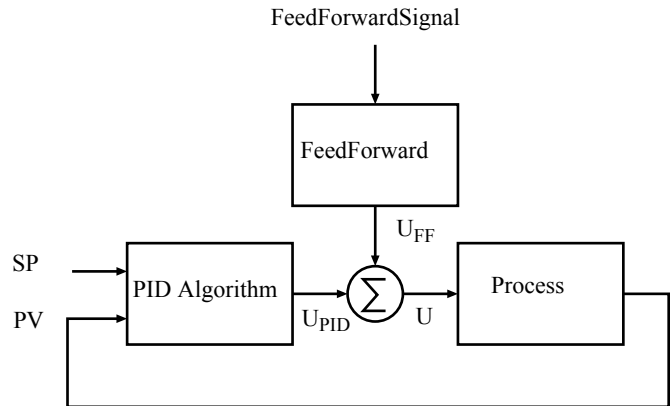


Figure 49 Feed Forward Signal.

Static Gain

If the parameter *Mode* is set to *StaticGain*, you can determine the value of *Gain* using the procedure below:

- 1 Disable the Feed Forward function by setting *Mode* to *Off*.
- 2 Bring the system into a steady state.
- 3 Take note of the values of the *FeedForwardSignal* and *ControlSignal*. Call them v_1 and u_1 respectively.
- 4 Introduce a process disturbance, e.g. by changing the process load, and then wait until the system is in a steady state.
- 5 Take note of the values of the *FeedForwardSignal* and *ControlSignal*. Call them v_2 and u_2 respectively.
- 6 Use the formula below to acquire the value of the *Gain* parameter:

$$Gain = \frac{u_1 - u_2}{v_1 - v_2}$$

- 7 Enable the Feed Forward function by setting *Mode* to *StaticGain*.

Adaptive Gain

Adaptive Feed Forward will always work even if the controller has not been configured for Adaptive Control.

To enable Adaptive Feed Forward set the parameter *Mode* to AdaptiveGain. For the Adaptive Feed Forward function to work, you must tune the system using the Autotuner. The Autotuner gives the Feed Forward function start information.

Notes:



- Adaptive Feed Forward cannot be used with output signal Tracking, or with Three State Pulse output with no actuator feedback
- Feed Forward (*Mode* = StaticGain) can be used with Three State Pulse with no actuator feedback.
- Adaptive Gain will not work together with pPI control.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	-

Autotuner

Autotuning is an aid which permits automatic setting of the controller PID parameters. It uses a relay-type function with hysteresis which initiates a controlled oscillation in the process.

When the function is enabled the autotuning procedure starts by inserting a step in the output signal so that the Process Value begins to diverge from the Set Point. Because the relay function is controlling the process during autotuning, self-oscillation is induced so that the output signal alternately exceeds and falls below the original value by a defined amplitude. The Autotuner itself adjusts this amplitude so that the Process Value will not be greater than a level which is necessary for the process response to be isolated from process and measuring noise (the oscillation amplitude is roughly comparable to the process noise level).

By measuring the process response the Autotuner can identify the process dynamic characteristics during autotuning. When the parameters are considered reliable the autotuning process stops. New PID parameters are calculated and stored in the controller.

The autotuning function has certain similarities to the manual Ziegler Nichols method. However, the Autotuner has control over the process oscillation amplitude which results in considerably less disturbance to the process.

The Autotuner even incorporates a form of intelligence. In situations where the D element should not be used, for example in level control, the Autotuner automatically identifies the process and eliminates the D element.

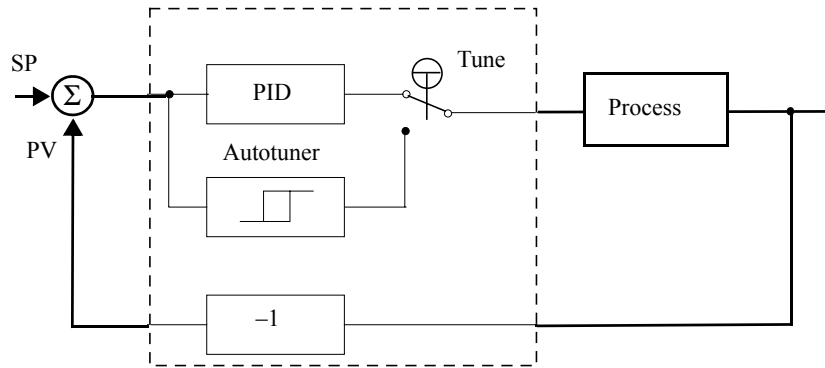


Figure 50 Autotuner Block Diagram.

The Autotuner block holds four parameters – *Mode*, *FirstStep*, *ControllerDynamics* and *Reset*.

The parameter *Mode* switches Autotuning Off or On.

The parameter *FirstStep* defines the maximum size of the first step in the Autotuning process.

The parameter *ControllerDynamics* is used to specify the control process used.

When the *Reset* parameter is set to OK the internal parameters for the Autotuner function are reset.

Parameters		
Name	Setting	Unit
Mode	Off, On	none
FirstStep	3, 5, 10	%
ControllerDynamics	Normal, DeadTime, PI, pPI	none
Reset	OK	none

Table 108 Autotuner Parameters.

pPi Controller

The pPi (predicting PI) controller is a specially designed dead time controller. It should be used when the dead time is longer than twice the dominant process time constant. If the dead time is varying, for example with flow or conveyor speed, the pPi controller can be combined with gain scheduling. The Autotuner function is not needed and is automatically disconnected when pPi control is configured.

The pPi controller is easy to tune, first by making a step response test on the process and then setting the controller parameters by observation. Thereafter the pPi controller can be adjusted manually. In most cases an increase in K (Gain) or a decrease in TI (Integral Time) gives faster control but poorer damping.

In the figure below:

$$\Delta y = y1 - y0$$

$$\Delta u = u1 - u0$$

$$T = t2 - t1$$

$$L = t1 - t0$$

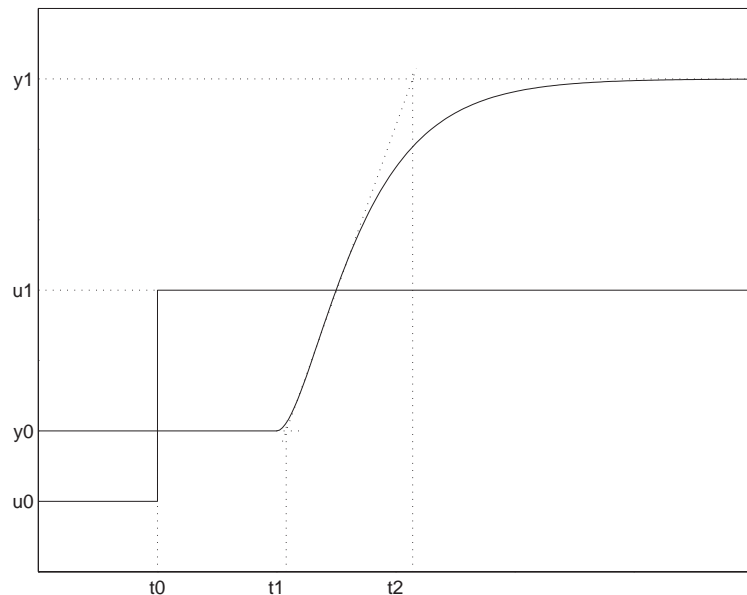


Figure 51 Dead Time Process Step Response.

By determining the four key parameters L , T , Δy and Δu from a step response, the three process parameters static gain K_p , dominant time constant T_p and dead-time L_p can be determined according to the expressions below:

$$K_p = \frac{\Delta y}{\Delta u}$$

$$T_p = T$$

$$L_p = L$$

Knowing these, the pPI's parameters can easily be calculated:

$$K = \frac{1}{K_p}$$

$$T_p = T$$

$$L_p = L$$

Example

If the four key parameters are found to have the following values:

$$\Delta y = y1 - y0 = 80 - 20 = 60\%$$

$$\Delta u = u1 - u0 = 20 - 10 = 10\%$$

$$T = t2 - t1 = 60 - 40 = 20 \text{ s}$$

$$L = t1 - t0 = 40 - 20 = 20 \text{ s}$$

... then we get ...

$$K_p = \frac{\Delta y}{\Delta u} = \frac{60}{10} = 6$$

$$T_p = T = 20 \text{ s}$$

$$L_p = L = 20 \text{ s}$$

... and the controller should be set up according to ...

$$K = \frac{1}{K_p} = \frac{1}{6} \approx 0.17$$

$$TI = T_p = 20 \text{ s}$$

$$TD = L_p = 20 \text{ s}$$



Note: When setting the controller parameter TD you actually set the value of the L_p (Dead-time) parameter, not the Derivative Time. In the menu system however, the notation TD is used for the dead-time of the pPI controller.



Notes:

- With three state pulse output positional feedback must be used for the autotuning process to work. Because the pPI controller does not use the Autotuner, this function will be disconnected when pPI controller is configured.
- If output signal tracking is used, precautions must be taken to ensure that the output signal does not vary because of the tracking function while autotuning is in progress.
- pPI Control is available only in the ECA600.
- When *ControllerDynamics* is set to pPI the Autotuning feature is not available.
- Dead-time design. This is a special PID controller where the Autotuner adapts the gain and integration time for the dead-time process, i.e. low gain and short integration time. This controller should not be used for integrating processes.

Starting the Tuning Process

If tuning has been made available in the ECA controller configuration it may be started at any time. Proceed as follows:

- 1 Press the Next key until the top line of the display shows `To tune press √`.
- 2 To start the Tuning process, press the OK key as instructed by the text in the display.

During Autotuning the bottom line of the display indicates each stage of the process by showing the text `Tuning phase 1`, then `Tuning phase 2`, then `Tuning phase 3`, etc. This is shown as white text on a black background across the width of the bottom line. On completion of the tuning the display returns to normal.



Notes:

- Before you activate Tuning, ensure that the difference between the set point and the process value is small, otherwise the tuning may fail.

- The Tuning process can be interrupted by pressing the Hand key. In this case no changes are made to the controller's parameters.

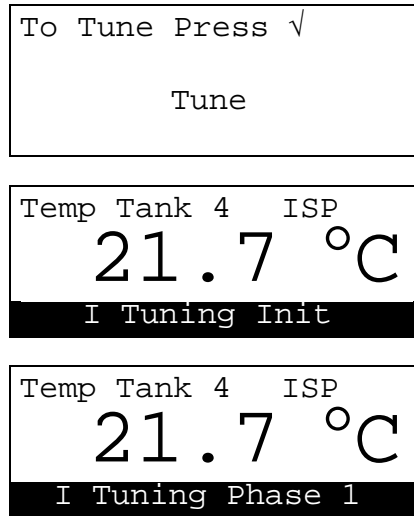


Figure 52 Display – Tuning Process.

After an autotuning sequence the Control Mode is always Auto.

Tuning Conditions

The Tuning conditions are summarised below.

State	Tuning Possible
Manual Control Mode (and Set Point RampMode ≠ Off)	Yes
Auto Control Mode (and set Point RampDone = True)	Yes
Using Feed Forward with Mode = Static Gain	No
Using Three State Pulse output without feedback	No
Forced Output Control Mode	No
Tracking Control Mode	No
Set Point Ramp not done	No

Table 109 Tuning Conditions.



Note: If the controller is to be connected to another control circuit you must reset the Autotuner so that the controller retains its old PID parameters but at the next autotuning the memory is cleared ready for a completely new process with different characteristics.

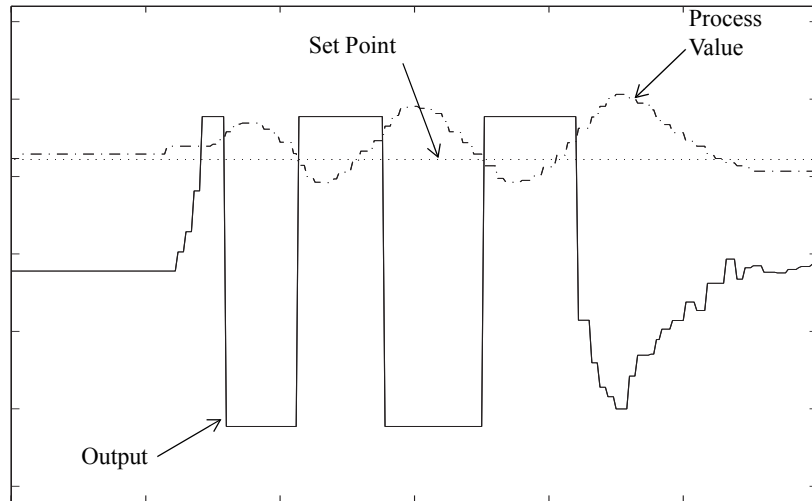


Figure 53 Autotuning Graphs Showing SP and PV During Output Signal Changes.

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	-

Adapt

Adaptive control automatically alters the PID parameters if the process changes. The pPI controller cannot be used with adaptive control.

The Adapt block contains four parameters – *Mode*, *Critical Gain*, *Critical Period* and *Reset*.

The parameter *Mode* sets the adapt function On or Off.

The parameters *Critical Gain* and *Critical Period* are usually determined by the Autotuner, and are used by the adaption algorithm as starting values for the adaptive control.

When the *Reset* parameter is set to OK the internal parameters for the Adapt function are reset.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
CriticalGain	$0.0 < \text{CriticalGain} \leq 100.0$	None
CriticalPeriod	$0.0 \leq \text{CriticalPeriod} < 10000.0$	seconds
Reset	OK	None

Table 110 Adapt Block Parameters.

The controller start value is determined by carrying out the Autotune procedure.



Warning – Large Parameter Changes: Adaptive PID control continuously adjusts itself to the process by changing the PID parameters, but for safety reasons the parameters should not be altered in an unlimited way. Gain, for example, might increase due to a clogged valve (the controller interprets this fault as reduced process gain). The controller thus provides a warning by means of a warning message on the display once the parameters have changed sufficiently to exceed the permitted value (maximum four times the change in gain).



Notes:

- In a few cases (about 1% of the total controller applications) the Autotuner may not work very well, e.g. in extremely noisy processes. In these cases the controller should be given a start value by inserting the *CriticalPeriod* and the *CriticalGain*. Analyse the process to determine these values. See the section Critical Gain and Critical Period below.
- If the process has a long deadtime, or a noisy measuring signal, select deadtime or PI design. The adaptive controller then automatically selects the PID parameters which are particularly suited to this type of process.
- If Three State Pulse output is used actuator feedback must be employed.
- Adaptive functions cannot be configured if the controller is set for tracking in Auto since the adaptive function would receive false information.

- Ringing in the process (disturbances with the same time constant as the process) can cause problems with adaptive controlling. If these disturbances cannot be eliminated the controller must be set for adaptive feed forward.

Critical Gain and Critical Period

In a very few cases (like very noisy processes), the AutoTuner may not be able to determine the control loop characteristics (*CriticalGain* and *CriticalPeriod*) properly. The adaptive function does not work properly without values for these parameters. In this case you must set them manually.

Ziegler-Nichols developed a method called the Calculated Sensitivity Method. The *CriticalGain* and *CriticalPeriod* parameters can be determined using this method.

- 1 Put the unit in Manual mode.
- 2 Adjust the controller's parameters so that a pure P-controller is configured.
- 3 Set the minimum value for the Gain parameter ($K = 0.01$).
- 4 Adjust the controller Output signal manually until $PV = SP$.
- 5 Read the current Output signal.
- 6 Set the controller's Offset parameter the value of the output signal.
- 7 Put the unit in Auto mode.
- 8 Change SP by a few percent. (As you change SP, try to keep the value relatively constant by alternating increases with decreases.)
- 9 Observe PV and the Output signal to see if the system oscillates.
- 10 If the system oscillates with a decreasing amplitude, increase Gain, and go to step 8.
- 11 If the system oscillates with a increasing amplitude, decrease *Gain*, and go to step 8.
- 12 If the system oscillates with a constant amplitude, you are done!

Read the controller's *Gain* setting. This is the setting of the *CriticalGain* parameter. Read the oscillation period time. This is the setting of the *CriticalPeriod* parameter.

Miscellaneous Functions

The Miscellaneous section contains two parameters – *ControllerAction*, and *ESPStepBehavior*.

If the controller action is known you can set it using the parameter *ControllerAction* parameter. Otherwise the Autotuner will determine and set the correct value.

If *ESPStepBehavior* = P-AndD-Contr and an External Set Point (ESP) is used, and it changes rapidly, the Proportional (P) and Derivative (D) parts of the PID calculation algorithm are added to the Integral (I) part to form the Control Block output signal, *DeltaU*.

However, if *ESPStepBehavior* = NoP-AndNoD-Contr the Proportional (P) and Derivative (D) parts are not used when calculating *DeltaU*.

When an Internal Set Point is used (i.e. *ESPMode* is False), *ESPStepBehavior* is always NoP-AndNoD-Contr and the actual setting of *ESPStepBehavior* is ignored.

The Control Loop Performance Monitor (CLPM) monitors the control loop for abnormal oscillations and signals an alarm when any occur. The function is completely automatic and requires no user action or parameters to be set after the function is activated.



Note: CLPM is only available in the ECA600 controller.

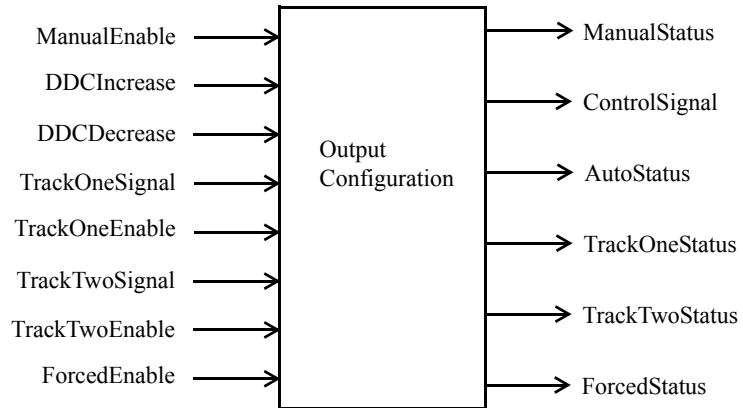
When the block detects an oscillation outside the limits of its detection algorithm a message is indicated on the display. See Appendix B Error, Warning and Information Messages. The most common cause of abnormal oscillations is high valve friction caused by wear and insufficient maintenance.

Parameters		
Name	Setting	Unit
ControllerAction	unknown, direct, reversed	None
ESPStepBehavior	P-AndD-Contr, NoP-AndNoD-Contr	None
CLPM	Off, On	None

Table 111 Miscellaneous Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

Output Configuration



The Output Configuration (OC) block supervises the output of the controller. The block has four different modes of operation:

- Auto.
- Manual.
- Tracking.
- Forced control.

The Manual Mode LED is only lit in the Manual Mode.

Inputs	
Name	Connection
ManualEnable	Off, digital
DDCIncrease	Off, digital
DDCDecrease	Off, digital
TrackOneSignal	Off, Analog
TrackOneEnable	Off, Digital
TrackTwoSignal	Off, Analog
TrackTwoEnable	Off, Digital
ForcedEnable	Off, Digital

Table 112 Output Configuration Block Inputs.

Outputs	
Name	Type
ManualStatus	Digital
AutoStatus	Digital
TrackOneStatus	Digital
TrackTwoStatus	Digital
ForcedStatus	Digital
ControlSignal	Analog

Table 113 Output Configuration Block Outputs.

The Control Mode priorities in order are:

- 1 Manual.
- 2 Forced Output.

- 3 TrackOne.
- 4 TrackTwo.
- 5 Auto.

Manual Mode

The *ManualBehavior* parameter, the *ManualEnable* input and the Hand key at the front panel determine the status of the output *ManualStatus*. When *ManualBehavior* is set to Normal the *ManualEnable* input and the Hand key signal have equal priority.

Parameters		
Name	Setting	Unit
ManualBehavior	Normal, TrueManual	None

Table 114 Manual Mode Parameters.

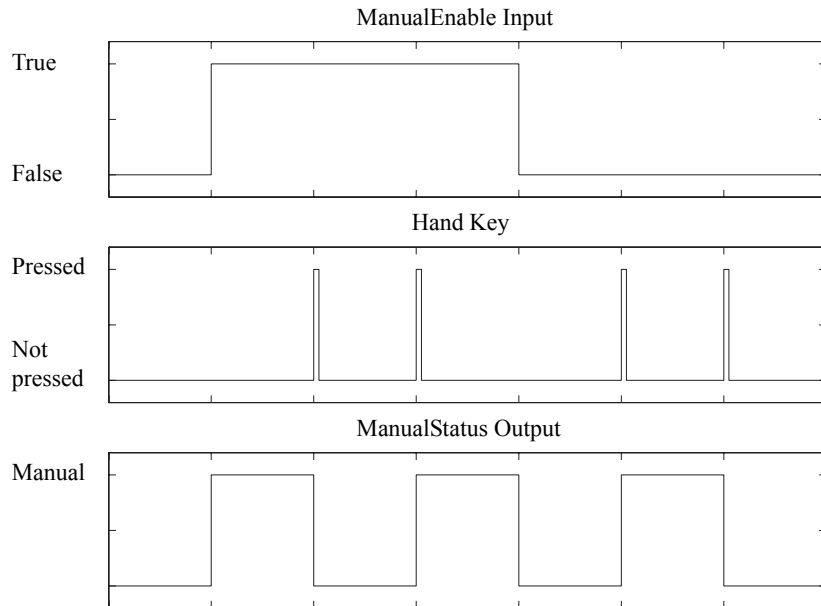


Figure 54 ManualBehavior = Normal.

When *ManualBehavior* is set to *TrueManual* and the *ManualEnable* input has the value *True*, the Hand key is disabled. If the *ManualEnable* input is *False*, the Hand key is enabled and can toggle the Control Mode.

The transition from *Auto* to *Manual* can be made by either the Hand key or the *ManualEnable* input. However, the transition from *Manual* to *Auto* can be made only by the Hand key.

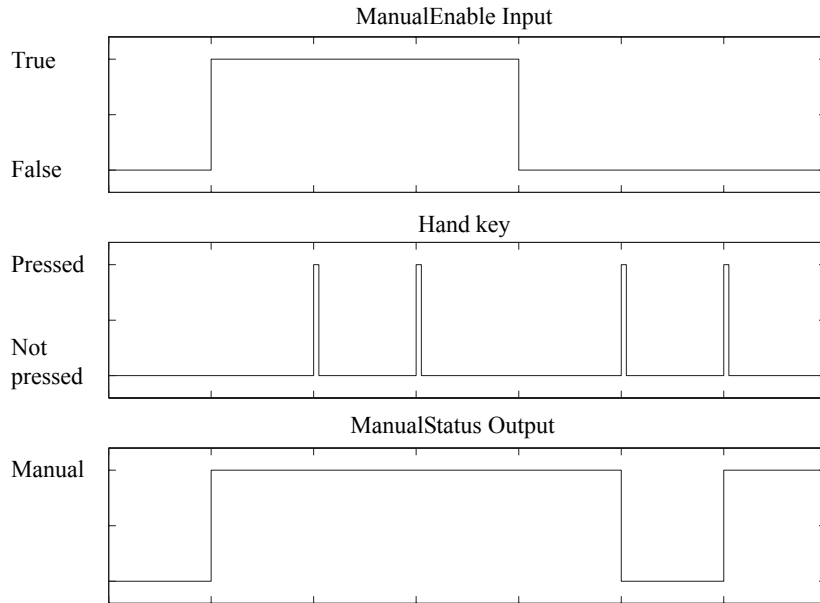


Figure 55 ManualBehavior = TrueManual.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	-

DDC

The DDC function has two digital inputs, *DDCIncrease* and *DDCDecrease*, and one parameter *DDCTime*. The *DDCIncrease* and *DDCDecrease* inputs only control the *ControlSignal* when the *ControlMode* is *Manual*.

Activating *DDCIncrease* makes the *ControlSignal* increase; activating *DDCDecrease* makes the *ControlSignal* decrease. The rate at which the *ControlSignal* changes when one of these inputs is activated is defined by the parameter *DDCTime*.

Activating both *DDCIncrease* and *DDCDecrease* has the same effect as having both inputs deactivated.



Note: In Manual Control Mode the *ControlSignal* is the sum of the competing signals – *DDCIncrease*, *DDCDecrease* and the signals from the keys on the front panel.

Parameters		
Name	Setting	Unit
DDCTime	$1 \leq \text{DDCTime} \leq 1000$	seconds

Table 115 DDC Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
-	1	2	1

Tracking

The ECA60 and EMA60 have the function *TrackOne*; the ECA600 has the functions *TrackOne* and *TrackTwo*. *TrackOne* and *TrackTwo* perform identical functions but *TrackOne* has a higher priority than *TrackTwo*.



Note: The Tracking function is not available when using a Three State Pulse Output.

In Tracking Mode the value of the Controller Output is determined by an external signal. The mode is initiated by digital signals *TrackOneEnable* and *TrackTwoEnable*. For each tracking function there are four different tracking modes:

- InAutoWDeltaU
- WhenEnabled
- WhenEnabledWDeltaU
- InAutoBumpless



Note:

The Control Block's output signal is called *DeltaU*. This signal is used by the Output Configuration Block to form the *ControlSignal*, which is used for analog outputs. When selecting a mode where *DeltaU* is used (InAutoWDeltaU or WhenEnabledWDeltaU), *ControlSignal* is calculated according to the formula below:

$$\text{ControlSignal} = \text{TrackSignal} + \text{DeltaU}$$

Note:

The *DeltaU* signal can take negative or positive values. The EMA60 does not have a Control Block and therefore the signal *DeltaU* does not exist. This means that the modes *InAutoWDeltaU* and *WhenEnabledWDeltaU* do not exist in the EMA60. In the text and tables below, any features which are not available in the EMA60 are identified by a “*” character.

In the modes *InAutoWDeltaU*, *WhenEnabled* and *WhenEnabledWDeltaU* the switch from Manual to Auto is mostly not bumpless. If the manually adjusted *ControlSignal* differs from *TrackSignal + DeltaU* when switching from Manual to Auto, the *ControlSignal* makes a jump. In some applications where this is not acceptable the mode *InAutoBumpless* should be used. It is active whenever Mode is *InAutoBumpless* and the Control Mode is Auto. In practice, the Output Configuration block ramps the *ControlSignal* from the manually adjusted value to the value defined by *TrackSignal + DeltaU*. The speed of this ramp is defined by the parameter *BumplessSpeed*.

The Tracking function is summarised in the table below.

Mode	Active When...	ControlSignal = ...
<i>InAutoBumpless</i>	ControlMode = Auto	TrackSignal
<i>InAutoWDeltaU*</i>	ControlMode = Auto	TrackSignal + DeltaU
<i>WhenEnabled</i>	TrackEnable = True	TrackSignal
<i>WhenEnabledWDeltaU*</i>	TrackEnable = True	TrackSignal + DeltaU

Table 116 Tracking Function.

Note: In the table above *TrackEnable* means *TrackOneEnable* or *TrackTwoEnable* and *TrackSignal* means *TrackOneSignal* or *TrackTwoSignal*.



Parameters		
Name	Setting	Unit
TrackOneMode	InAutoBumpless, InAutoWDeltaU, WhenEnabled, WhenEnabledWDeltaU	None
TrackOneBumpSpeed	$1 \leq \text{TrackOneBumpSpeed} \leq 1440$	s/100%
TrackTwoMode	InAutoBumpless, InAutoWDeltaU, WhenEnabled, WhenEnabledWDeltaU	None
TrackTwoBumpSpeed	$1 \leq \text{TrackTwoBumpSpeed} \leq 1440$	s/100%

Table 117 Tracking Parameters.

If both *TrackOneEnable* and *TrackTwoEnable* are active, the Controller Output will track *TrackOneSignal*.

InAutoWDeltaU*

With *TrackMode* set to *InAutoWDeltaU*, the Output Configuration block tracks the *TrackSignal* whenever *ControlMode* is Auto. The Control block is active, and PID-algorithm contributes to the *ControlSignal*.

$$\text{ControlSignal} = \text{TrackSignal} + \text{DeltaU}$$

When *ControlMode* is Auto, both the *AutoStatus* and the *TrackStatus* outputs of the Output Configuration block are True. When *ControlMode* is Manual, the *ManualStatus* output of the Output Configuration block is True.

WhenEnabledWDeltaU*

With *TrackMode* set to *WhenEnabledWDeltaU*, the Output Configuration block tracks the *TrackSignal* whenever *TrackEnable* is True. The Control block is active, and PID-algorithm contributes to the *ControlSignal*.

$$\text{ControlSignal} = \text{TrackSignal} + \text{DeltaU}$$

When *ControlMode* is Auto, both the *AutoStatus* and the *TrackStatus* outputs of the Output Configuration block are True. When *ControlMode* is Manual, the *ManualStatus* output of the Output Configuration block is True.

WhenEnabled

With *TrackMode* set to WhenEnabled, the Output Configuration block tracks the *TrackSignal* whenever *TrackEnable* is True.

$$\text{ControlSignal} = \text{TrackSignal}$$

When *ControlMode* is Auto, the *TrackStatus* output of the Output Configuration block is True. When *ControlMode* is Manual, the *ManualStatus* output of the Output Configuration block is True.

InAutoBumpless

With *TrackMode* set to InAutoBumpless, the Output Configuration block tracks the *TrackSignal* whenever *ControlMode* is Auto.

$$\text{ControlSignal} = \text{TrackSignal}$$

When *ControlMode* is Auto, the *TrackStatus* output of the Output Configuration block is True. When *ControlMode* is Manual, the *ManualStatus* output of the Output Configuration block is True. In this mode the transfer from Manual to Tracking is bumpless. When switching from Manual to Auto, the *ControlSignal* is ramped from the value set manually to the value of the *TrackSignal* input. The speed of the ramp is determined by the parameter *TrackOneBumpSpeed* or *TrackTwoBumpSpeed*.

When the controller is tracking, regardless of the mode, the Control Mode can be changed to Manual, either by pressing the Hand key or via the *ManualEnable* input. In this state the user has full control over the Control Signal.

When the controller is tracking, the word “Tracking” is displayed on the top line of the display.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

Forced Output

The PID output can be forced to a preset value by activating the *ForcedEnable* input. When the *Mode* parameter is set to Off, forced control is disabled. The value to which the control output is forced is defined by the parameter *ForcedValue*.

Whatever mode is selected the control output signal starts from the *ForcedValue* when the *ForcedEnable* input is dropped.

When *Mode* is set to *AutoToAny* it is possible for the operator to change the control mode to *Manual*. If the *ForcedEnable* input changes from *True* to *False* when the controller is in *Manual Control* mode, the *Control Mode* remains unchanged.

Mode Name	Entry Mode	Exit Mode	Manual Mode Possible
<i>AutoToAny</i>	<i>Auto</i>	<i>Auto or Manual</i>	<i>Yes</i>
<i>AutoToManual</i>	<i>Auto</i>	<i>Manual</i>	<i>Yes</i>
<i>AnyToPrevious</i>	<i>Auto or Manual</i>	<i>Previous mode</i>	<i>No</i>
<i>AnyToManual</i>	<i>Auto or Manual</i>	<i>Manual</i>	<i>No</i>

Table 118 Manual Mode Selection.

When the controller enters *Forced Output Mode*, i.e. *ForcedEnable* is active, the message “*Forced*” is shown at the left of the top line of the operator mode display. If the operator then decides to put the controller into *Manual Mode* by pressing the *Hand* key, the *Hand* key LED illuminates.

Parameters		
Name	Setting	Unit
<i>Mode</i>	<i>Off, AutoToAny, AutoToManual, AnyToManual, AnyToPrevious</i>	<i>n/a</i>
<i>ForcedValue</i>	$0.0 \leq \text{ForcedValue} \leq 100.0$	<i>%</i>

Table 119 Forced Output Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

Miscellaneous

A group of parameters affecting the output are grouped together under the heading miscellaneous.

Parameters		
Name	Value	Unit
RestartBehavior	PrevValPrevMode, PrevValManMode, PrevValAutoMode RestartValManMode RestartValAutoMode RestartValPrevMode	None
RestartValue	$0.0 \leq \text{ReStartvalue} \leq 100.0$	%
ErrorBehavior	HoldValue, ErrorValue	None
ErrorValue	$0.0 \leq \text{ErrorValue} \leq 100.0$	%
LimitMode	Off, InAutoAndTracking, AlwaysExceptForced	None
LoLimit	$0.0 \leq \text{LoLimit} \leq \text{HiLimit} \leq 100.0$	%
HiLimit	$0.0 \leq \text{LoLimit} \leq \text{HiLimit} \leq 100.0$	%

Table 120 Miscellaneous Output Configuration Parameters.

The configuration of the parameter *RestartBehavior* determine how the controller behaves when it is powered up:

- *PrevValPrevMode* – start up using the Control Mode at the time of power down and the same Control Signal as at power down.
- *PrevValManMode* – start up using the Manual Control Mode and the same Control Signal as at power down.
- *PrevValAutoMode* – at power up, start up using the ControlSignal at the time of power down, and Auto Control mode.
- *RestartValManMode* – start up using the Manual Control Mode and the Control Signal defined by the parameter *RestartValue*.
- *RestartValAutoMode* – at power up, start up using the ControlSignal as defined by the parameter *RestartValue*, and Auto Control mode.
- *RestartValPrevMode* – at power up, start up using the ControlSignal as defined by the parameter *RestartValue*, and the Control mode active at the time of power down.

Note. In EMA60, the setting of the parameter *HandKey* (when set to *LockIn-Manual*) has priority over *RestartBehavior*.

The behaviour of the controller output value when an error condition occurs that affects the behaviour of the PID algorithm is determined by the parameter *ErrorBehavior*:

- *HoldValue* – Hold the current signal and change the Control Mode to Manual.
- *ErrorValue* – Set the control signal to the value defined by the parameter *ErrorValue* and change the Control Mode to Manual. The user must set the parameter *ErrorValue*.

ControlSignal can be limited to a certain range of valid values. This feature is activated by setting the parameter *LimitMode*. The actual limits are stored in the parameters *LoLimit* and *HiLimit*. Even if *LimitMode* is set to *InAutoAndTracking* or *AlwaysExceptForced*, the Control Signal limiting is not functional if *ForcedEnable* is True.



Note: The ECA06 does not have the Control Signal Limit function and hence does not have the parameters related to this function (*LimitMode*, *LoLimit* and *HiLimit*).

Output Functions

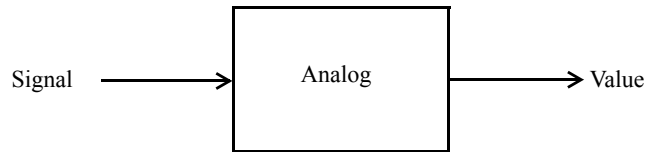
The available Output Functions (OF) are shown in the table below.

Parameters		
Name	Value	Unit
Type	Analog ThreeStatePulse TwoStatePulse SplitRange	None

Table 121 Available Output Functions.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	-

Analog



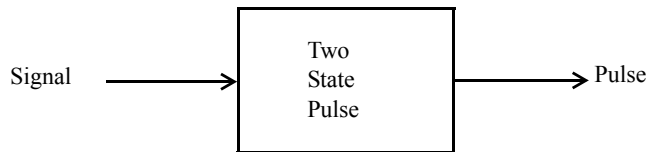
$$Value = Signal$$

This block has no actual functionality but merely transports the signal at the input to the output.

To connect this analog signal to the system under control, one of the Analog Outputs must be used in the Output column.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	-

Two State Pulse



$$PulseLength = CycleTime \times \frac{Signal}{100}$$

The Two State Pulse output is a pulse length modulated digital output. It has one analog input, *Signal*, and one digital output *Pulse*. It also holds one parameter, *CycleTime*.

The active time of the output *Pulse* is proportional to the value of the input *Signal*. When the input is 0%, the output signal is active for the minimum time period, which is equal to the parameter *SampleTime* found in the System Block settings. When the input signal is 100%, the output signal is equal to the parameter *CycleTime*.

To connect the output pulse signal to the system under control one of the Digital Outputs must be used.

Inputs	
Name	Connection
Signal	Fixed connection

Table 122 Two State Pulse Input.



Note: The inputs to the Output Functions block identified as fixed connections cannot be changed. They are listed for information only.

Parameters		
Name	Setting	Unit
CycleTime	$0 \leq \text{CycleTime} \leq 1000$	seconds

Table 123 Two State Pulse Parameters.

Outputs	
Name	Type
Pulse	Digital

Table 124 Two State Pulse Output.

For example, suppose the parameter *CycleTime* is set to 50 seconds. With a *Signal* value of 60%, the *Pulse* output of the Two State Pulse block is active for 60% of the *CycleTime*:

$$60.0\% \times 50 = 30$$

The Pulse output is active for 30 seconds and then inactive for 20 (50–30) seconds.

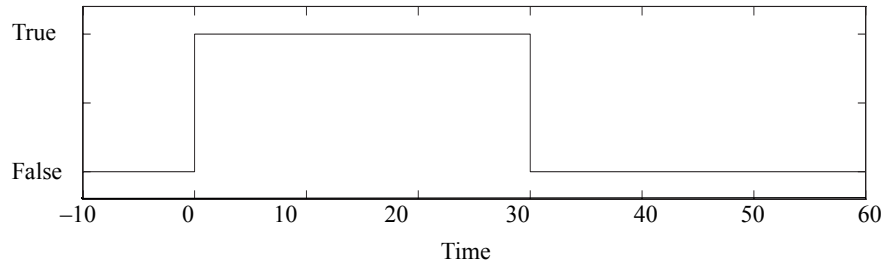
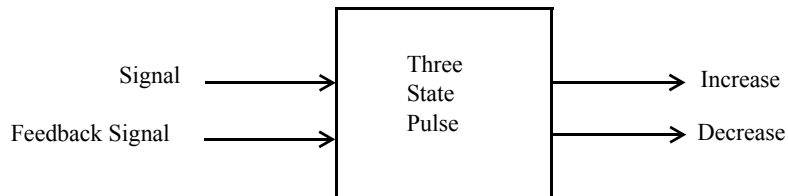


Figure 56 Two State Pulse Example.

Available in			
ECA06	ECA60	ECA600	EMA60
-	X	X	-

Three State Pulse



The Three State Pulse output has two analog inputs, *FeedbackSignal* and *Signal*, two digital outputs, *Increase* and *Decrease*. It also holds three analog parameters, *DeadBand*, *ActuatorPosTime*, and *ActuatorMinPulseLen*.

As long as the input signal (which is the output from the control block) increases, the *Increase* output is active. As long as the input signal decreases, the *Decrease* output is active. When the input signal is stable within the deadband, neither output is active.

To connect the output *Increase* and *Decrease* pulse signals to the system under control, two of the Digital Outputs must be used.

The parameter *ActuatorPosTime* represents the time it takes for the actuator to travel its entire span (from 0.0% to 100.0%, or vice versa). The parameter *ActuatorMinPulseLen* (value in percent) represents a buffer threshold (a value in percent) used for determining when to activate any output.

The *ControlSignal* changes (*DeltaU*) are buffered internally in the Three State Pulse block. When the sum of the accumulated changes becomes greater than the buffer threshold mentioned above, the block activates one of the outputs (*Increase* or *Decrease*).

As long as the control error is less than the value defined by *DeadBand*, the internal buffers are emptied, and no output is activated.

There is an internal interlock that prevents the block from changing active outputs more often than once every two seconds.

When a Three State Pulse output is configured, the following conditions apply:

- If feedback is used the *FeedbackSignal* input is used for indication as the controller Output signal on the Output bar graph and on the display.
- If feedback is not used and the controller is in manual mode, an Inc/Dec indication is shown in the Output menu.
- The ControlSignal *Limit* function is not available.



Note: If a Three State Pulse output with no actuator feedback is configured, no signal is displayed on the Output bar graph. If a Three State Pulse with output with actuator feedback is configured, the actuator feedback signal is displayed on the Output bar graph.

Inputs	
Name	Connection
FeedbackSignal	Off, Analog
Signal	Fixed connection

Table 125 Three State Pulse Inputs.



Note: The inputs identified as fixed connections cannot be changed. They are listed for information only.

Parameters		
Name	Setting	Unit
DeadBand	$0.0 < \text{Deadband} \leq 10.0$	%
Hysteresis	$0.5 \leq \text{Hysteresis} \leq 3.0$	%
ActuatorPosTime	$15 \leq \text{ActuatorPosTime} \leq 1000$	s
ActuatorMinPulseLen	$0.3 \leq \text{ActuatorMinPulseLen} \leq 3.0$	%

Table 126 Three State Pulse Parameters.

Outputs	
Name	Type
Increase	Digital
Decrease	Digital

Table 127 Three State Pulse Outputs.

The parameter *Deadband* sets the sensitivity of the Three State Pulse output in conjunction with the PID algorithm. As long as the difference between the Process Value and Set Point signals is less than the configured deadband no change is made to the Three State Pulse output signals. This applies to Auto Control mode. A similar parameter, *Hysteresis* is described below.



Note: The Three State Pulse output cannot be used with a P-controller ($TI_x = \text{Off}$ and $TD_x = 0.0$). If you configure such a set up the signal *DeltaU* will remain constant at zero.

When using the Three State Pulse output without feedback the following functions are not available:

- RestartBehavior
- Forced Output
- ErrorBehavior

- Adaptive Control
- Tuning
- Control Signal Limit

Non Auto Mode

Non-Auto mode comprises the following Control modes:

- Manual
- Forced Output

No Feedback

In Non-Auto mode with no feedback, the Out bar graph is 0.0% and the display shows Inc/Dec signal. The internal variable ΔU_{3P} is calculated as follows:

$$\Delta U_{3P} = ControlSignal_k - ControlSignal_{k-1}$$

where $ControlSignal_k$ is the control signal of the current sample and $ControlSignal_{k-1}$ is the control signal of the previous sample.

In Auto-mode with no feedback, the Out bar graph shows 0.0% for the $ControlSignal$ value. The internal variable ΔU_{3P} is calculated as follows:

$$\Delta U_{3P} = \Delta U_{COx}$$

Using Feedback

When using Three State Pulse with feedback, there are five different control modes, see table below.

Manual mode	In Manual Mode you can increase/decrease the output signal by pressing the Up/Down buttons. The digital outputs are activated as long as the button is pressed.
Auto-mode	In Auto-Mode the controller works as normal and the feedback signal is shown on the Output bar graph.
Forced* mode	If Forced Mode is used (and activated), the controller tries to force the feedback signal to the selected Forced Value by activating the Increase/Decrease outputs. This continues until the Forced Mode is left (not activated).

Restart Behavior* (to manual mode)	When the controller is restarted and the restart mode is Manual, the controller tries to force the feedback signal to the selected Restart Value by activating the Increase/Decrease outputs. This continues until the feedback signal is within the selected Hysteresis. The controller will then go to Manual mode.
Error Behavior* mode	If an Error arises (for the selected loop), the controller will try to force the feedback signal to the selected Error Value by activating the Increase/Decrease outputs. This will continue until the error is acknowledged.

*. See Output Configuration on page 155.

With Feedback and when in Non-Auto mode the internal variable ΔU_{3P} is calculated as follows:

$$\Delta U_{3P} = \text{ControlSignal} - \text{FeedbackSignal}$$

When in Auto-mode the Out bar graph shows the *FeedbackSignal* and the internal variable ΔU_{3P} is calculated as follows:

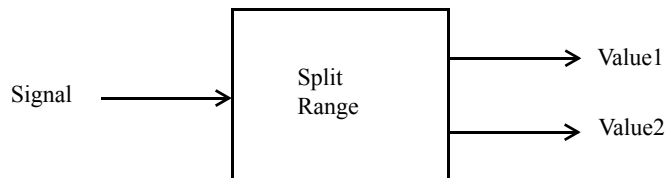
$$\Delta U_{3P} = \Delta U_{COx}$$

When switching from Auto to Non-Auto control modes a bumpless transfer is ensured by setting:

$$\text{ControlSignal} = \text{FeedbackSignal}$$

Available in			
ECA06	ECA60	ECA600	EMA60
-	-	X	-

Split Range



The Split Range output has one analog input, *Signal*, and two analog outputs, *Value1* and *Value2*. It also holds eight analog parameters, *MinIn1*, *MaxIn1*, *MinOut1*, *MaxOut1*, *MinIn2*, *MaxIn2*, *MinOut2* and *MaxOut2*.

This type of output splits the Control Block output signal into two signals, *Value1* and *Value2*. The linear relationships between the input and the two outputs *Value1* and *Value2* can be configured via the parameters.

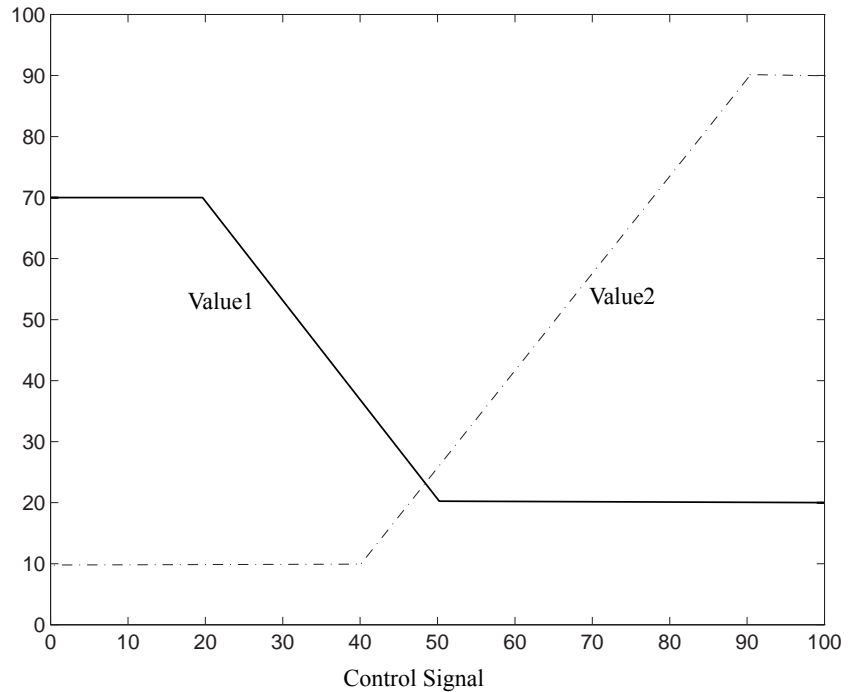


Figure 57 Split Range Output.

In the example above the coordinate pairs are:

Value1: (20.0, 70.0) and (50.0, 20.0) and Value2: (40.0, 10.0) and (90.0, 90.0).

The coordinates are entered according to the table below.

Parameter			
Name	Value	Name	Value
MinIn1	20.0	MinIn2	40.0
MaxIn1	50.0	MaxIn2	90.0
MinOut1	70.0	MinOut2	10.0
MaxOut1	20.0	MaxOut2	90.0

Table 128 Split Range Example.

For values that fall outside the range defined by the coordinate pairs, the function extends the characteristic with a horizontal line.

In the example above, for *ControlSignal* < 20.0%, the value of *Value1* is 70.0%, and for *ControlSignal* > 50.0%, the value of *Value1* is 20.0%.

Equivalently, for *ControlSignal* < 40.0%, the value of *Value2* is 10.0%, and for *ControlSignal* > 90.0%, the value of *Value2* is 90.0%.

Using this block, the user can have one “increase to close” signal, and one “increase to open” signal. This is useful when there is one valve that controls cooling media and one valve that controls heating media in a temperature control application.

Inputs	
Name	Connection
Signal	ControlSignal

Table 129 Split Range Inputs.



Note: The input is a fixed connection and cannot be changed. It is listed for information only.

Parameters		
Name	Setting	Unit
MinIn1	$0.0 \leq \text{MinIn1} \leq 100.0$	%
MaxIn1	$0.0 \leq \text{MinIn1} \leq \text{MaxIn1} \leq 100.0$	%
MinOut1	$0.0 \leq \text{MinOut1} \leq 100.0$	%
MaxOut1	$0.0 \leq \text{MaxOut1} \leq 100.0$	%
MinIn2	$0.0 \leq \text{MinIn2} \leq 100.0$	%
MaxIn2	$0.0 \leq \text{MinIn2} \leq \text{MaxIn2} \leq 100.0$	%
MinOut2	$0.0 \leq \text{MinOut2} \leq 100.0$	%
MaxOut2	$0.0 \leq \text{MaxOut2} \leq 100.0$	%

Table 130 Split Range Parameters.

Outputs	
Name	Type
Value1	$0\% \leq \text{Value1} < 100\%$
Value2	$0\% \leq \text{Value2} < 100\%$

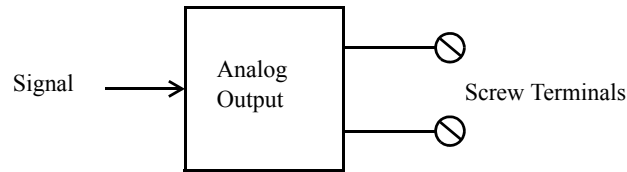
Table 131 Split Range Outputs.

Outputs

The Output block can be configured to provide analog and/or digital output signals as required.

Available in			
ECA06	ECA60	ECA600	EMA60
1	2	3	1

Analog Outputs



Each Analog Output Block (AO) has one analog input, *Signal* and produces a current signal at the external terminals of the controller. Each block also holds seven parameters – *Mode*, *MinCurrent*, *MaxCurrent*, *FixedCurrent*, *Alarm*, *Loop* and *Identity*.

The *Mode* parameter is used to select the block for operation and whether it operates in the Direct or Reverse modes.



Note: The ECA06 and EMA60 do not have the Fixed mode. In the text and tables below, any features which are not available in the ECA06 and EMA60 are identified by a “*” character.

With the block configured to *Mode* = Direct the block has the characteristic shown below:

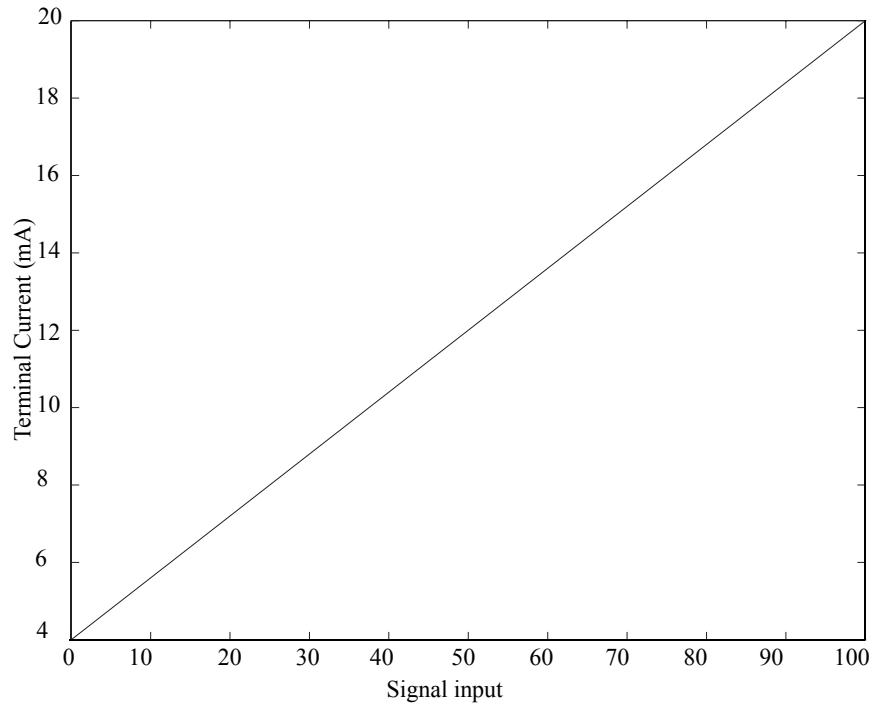


Figure 58 Direct Mode Characteristic.

With the block configured to *Mode = Reverse*, the block has the characteristic shown in Figure 59.

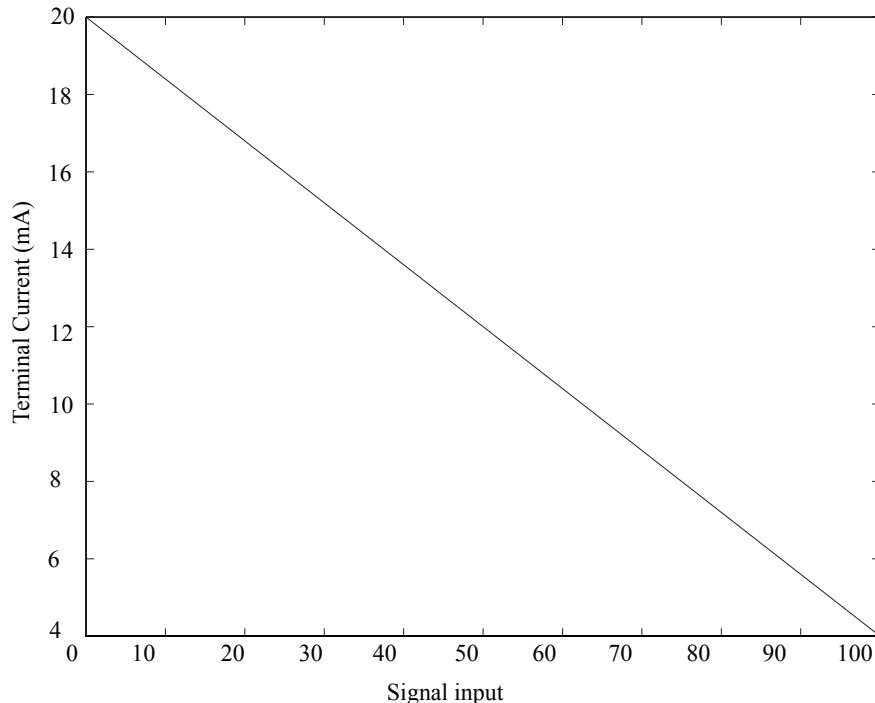


Figure 59 Reverse Mode Characteristic.

With the block configured to *Mode* = Fixed, the block produces a constant current on the terminal outputs, regardless of the value on the *Signal* input. The constant current is defined by the parameter *FixedCurrent*. This feature is useful when powering feedback potentiometers with current when using Three State Pulse outputs with feedback. The *FixedCurrent* parameter setting overrides the settings of *MinCurrent* and *MaxCurrent*.

If the output is configured as a Current output the current range can be defined by setting the parameters *MinCurrent* and *MaxCurrent*.

The input *Signal* is in the range 0–100%. In Direct mode *MinCurrent* corresponds to 0%, *MaxCurrent* to 100%. In Reverse mode *MinCurrent* corresponds to 100%, *MaxCurrent* to 0%.

For more information concerning the *Alarm* parameter and related functions please refer to Appendix B Error, Warning and Information Messages.

The *Loop* parameter applies only to the ECA600 and is used to define the control loop used.

The block can be given a twenty letter name by setting the parameter *Identity*.

Inputs	
Name	Connection
Signal	Off, Analog

Table 132 Analog Output Block Inputs.

Parameters		
Name	Setting	Unit
Mode	Off, Direct, Reverse, Fixed *	None
MinCurrent	$0.0 \leq \text{MinCurrent} \leq 4.0$	mA
MaxCurrent	$20.0 \leq \text{MaxCurrent} \leq 22.0$	mA
FixedCurrent*	$0.0 \leq \text{FixedCurrent} \leq 22.0$	mA
Alarm	Off, Warning, Error	None
Loop	Loop1, Loop2	None
Identity	User defined text	None

Table 133 Analog Output Block Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

Stiction Compensation

This function is to be used with analog output signals which control pneumatic regulating valves with high friction components. The stiction compensation adds short pulses of equal amplitude and duration to the analog output signal. This significantly reduces the control error during stick-slip motion compared with standard control without stiction compensation.

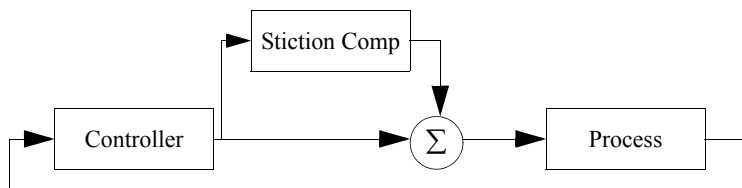


Figure 60 A stiction compensation function connected to the analog output signal.

The stiction compensation function detects the signal direction of the analog output signal and adds positive or negative pulses, depending on the signal direction, see figure below.

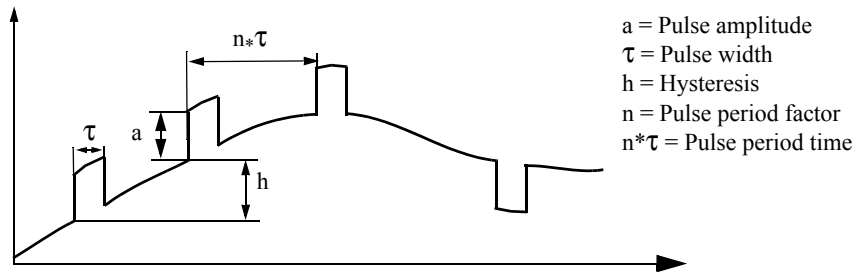


Figure 61 The stiction compensation function adds positive or negative pulses to the output signal.

The pulse period time will be the Pulse period factor (n) x the pulse width (τ).

The Pulse width value must be greater than or equal to the parameter *SampleTime* (TS). Example. If $T_s = 0.050$ s, then select $\tau = 0.050, 0.100, 0.150, \dots$

If $T_s = 0.100$ s, then select $\tau = 0.100, 0.200, 0.300, \dots$



Note. If the *SampleTime* value is greater than the pulse width τ , then τ will automatically be set to the *SampleTime* value (when the stiction compensation function is activated).

The hysteresis factor prevents the pulses from being superimposed and limits the number of pulses when the signal variation is low.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
Amplitude	1–5	%
PulseWidth	0.030–0.500	seconds
PulsePeriodFactor	2–5	-
Hysteresis	0.00–1.00	%

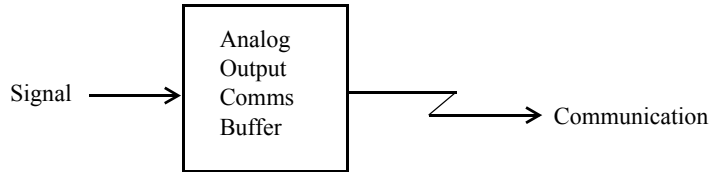
Table 134 Stiction Compensation parameters.



Note. All signal types can be used together with the Stiction Compensation function. This means, for example, that the function can be added to an external input signal and then be distributed as an output signal with the stiction compensation pulses superimposed.

Available in			
ECA06	ECA60	ECA600	EMA60
4	8	12	4

Analog Output Communication Buffers



Remote analog communication is provided by Analog Output Communications Buffers (AOC) via a two wire RS485 serial channel using the COMLI protocol.

Each Analog Output Communications Buffer can accept one analog input and produces one analog output. Each buffer block also holds two parameters, *Mode* and *Identity*. Each block may be given a twenty letter identification name by configuring the parameter *Identity*.



Note. See Appendix E for a full list of communication I/O bits and registers.

Inputs	
Name	Connection
Signal	Off, Analog

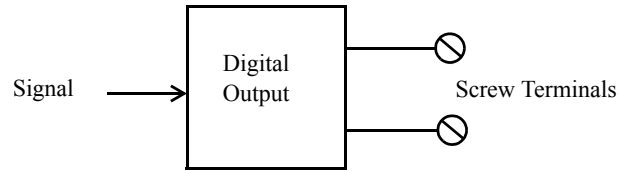
Table 135 Analog Output Communications Buffer Inputs.

Parameters		
Name	Setting	Unit
Mode	Off, On	None
Identity	User defined text	None

Table 136 Analog Output Communications Buffer Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
2	4	6	6

Digital Outputs



Each Digital Output Block (DO) has one digital input, *Signal* and provides one digital signal to the external terminals of the controller. Each block also holds two parameters – *Mode* and *Identity*.

The *Mode* parameter is used to select the block for operation and whether it operates in the Positive Logic or Negative Logic modes.

The block can be given a twenty letter name by setting the parameter *Identity*.

Inputs	
Name	Connection
Signal	Off, Digital

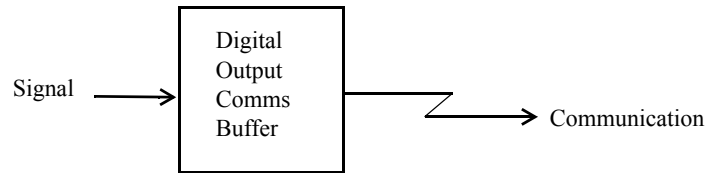
Table 137 Digital Output Block Inputs.

Parameters		
Name	Setting	Unit
Mode	Off, PositiveLogic, NegativeLogic	None
Identity	User defined	None

Table 138 Digital Output Block Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
4	8	12	4

Digital Output Communication Buffers



Remote digital communication is provided by Digital Output Communications Buffers (DOC) via a two wire RS485 serial channel using the COMLI protocol.

Each Digital Output Communications Buffer can accept one analog input and produces one analog output. Each buffer block also holds two parameters, *Mode* and *Identity*. Each block may be given a twenty letter identification name by configuring the parameter *Identity*.



Note. See Appendix E for a full list of communication I/O bits and registers

Inputs	
Name	Value
Signal	Off, digital

Table 139 Digital Output Communications Buffer Inputs.

Parameters		
Name	Value	Unit
Mode	Off, On	None
Identity	User defined text	None

Table 140 Digital Output Communications Buffer Parameters.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	-

ECA Operator Block

The Operator Block (OP) handles the presentation of all information on the controller front panel. The displayed process value can be scaled using the function “ProcessValue” and a selected signal value can be shown on the display using the function “AltDisplay”. Finally, the bar graph display can be changed using the function “bar graphs”.

Process Value

A transmitter generates either a 0–20 mA/0–5 VDC/0–10 VDC signal, or a 4–20 mA/1–5 VDC/2–10 VDC signal. This signal represents a flow, temperature, level or any other physical unit of measurement. To present the signal in Engineering Units (flow, temperature, etc.), a conversion must be made.

By setting the parameters *MinValue* and *MaxValue* to suitable values, the conversion is made automatically.

The user must first enter the number of decimals in the Engineering Unit presentation and then the minimum and maximum values. Five significant digits are available.

For example:

- Setting the *NumberOfDecimals* parameter to two (2) makes *MinValue* and *MaxValue* range from –999.99 to 999.99.
- Setting the *NumberOfDecimals* parameter to four (4) makes *MinValue* and *MaxValue* range from –9.9999 to 9.9999.

Note: If the value to be displayed exceeds five digits at any time, the rightmost digit is dropped.



Parameters		
Name	Setting	Unit
NumberOfDecimals	$0 \leq \text{NumberOfDecimals} \leq 4$	None
MinValue	$-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$	None
MaxValue	$-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$	None
Unit	%, °C, °F, K, W, kW, MW, J, kJ, MJ, rpm, Pa, kPa, MPa, bar, mbar, bara, l, l/s, l/min, l/h, Psi, Psig, Psia, mmWG, "WG, mmvp, mvp, mmH ₂ O, mmHg, "Hg, kp/cm ² , N/m ² , kg/cm ² , kg/h, t/h, Nm ³ /h, m ³ /h, t/24h, m/s, m/min, μs, μs/cm, ppm, pH, mg/l, feet, inches, meters, tons, lbs, lbs/sec, lbs/min, lbs/hr, ft/sec, ft/hr, BTU/hr, BTU, ft ³ /sec, ft ³ /hr, gals, gal/sec, gal/hr, cu.ft, MPPH, cc, cm, GPM.	None
Tag	User defined text.	None

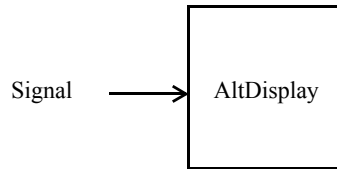
Table 141 Process Value Parameters.

Example

A sensor provides a 4–20 mA output for a calibrated flow range of 0 to 9 m³/min. How should *MinValue* and *MaxValue* be set if the user wants to present the Process Value and Set Point with two decimals?

4 mA represents a flow of 0 m³/min, and 20 mA represents a flow of 9 m³/min. Set *NumberOfDecimals* to 2, then set *MinValue* to 0.00 and *MaxValue* to 9.00.

Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	-

AltDisplay

The AltDisplay function is used to select a signal, scale and present its value on the display. The selected signal value is shown using “Next Key” (in Operator Mode). For setting of parameter values, see the example in function “ProcessValue”.

Inputs	
Name	Connection
Signal	Off, Analog

Table 142 AltDisplay Inputs

Parameters		
Name	Setting	Unit
NumberOfDecimals	$0 \leq \text{NumberOfDecimals} \leq 4$	None
MinValue	$-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$	None
MaxValue	$-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$	None

Parameters		
Unit	%, °C, °F, K, W, kW, MW, J, kJ, MJ, rpm, Pa, kPa, MPa, bar, mbar, bara, l, l/s, l/min, l/h, Psi, Psig, Psia, mmWG, "WG, mmvp,.mvp, mmH ₂ O, mmHg, "Hg, kp/cm ² , N/m ² , kg/cm ² , kg/h, t/h, Nm ³ /h, m ³ /h, t/24h, m/s, m/min, μs, μs/cm, ppm, pH, mg/l, feet, inches, meters, tons, lbs, lbs/sec, lbs/min, lbs/hr, ft/sec, ft/hr, BTU/hr, BTU, ft ³ /sec, ft ³ /hr, gals, gal/sec, gal/hr, cu.ft, MPPH, cc, cm, GPM.	None
Tag	User defined text. <i>Tag</i> is a text string displayed to the left on the top line of the unit in Operator Mode. <i>Tag</i> can hold twenty (20) characters, but the top line cannot hold all of them. If the length of <i>Tag</i> is less than or equal to 12, the full length is displayed. If the length of <i>Tag</i> is greater than twelve (12), eleven (11) characters are displayed followed by the characters "...".	None

Table 143 AltDisplay Parameters.



Note. The MinValue is the value shown for the internal signal value 0 and the MaxValue is the value shown for the internal signal value 1. If the Signal into the AltDisplay block is greater than 1 or less than 0, the value shown in the display is:

$$\text{Shown Value} = ((\text{MaxValue} - \text{MinValue}) \times \text{Signal} + \text{MinValue}).$$

Bar Graphs

The Bar Graphs section contains four parameters, *LeftBargraphMode*, *CenterBargraphMode*, *RightBargraphMode* and *LeftBargraphAlarm*.

Parameters		
Name	Value	Unit
LeftBargraphMode	Bar, ReversedBar, Point, ReversedPoint	None
CenterBargraphMode	Bar, ReversedBar, Point, ReversedPoint	None
RightBargraphMode	Bar, ReversedBar, Point, ReversedPoint	None
LeftBargraphAlarm	Absolute Alarm1, Absolute Alarm2, Deviation Alarm1, Deviation Alarm2	None

Table 144 Bar Graph Block Parameters.

The left and centre bar graphs each have 30 LED segments. The right bar graph has 20 LED segments.

By default the signals on the bar graphs are shown as traditional bars, with all segments lit from the bottom up to the segment that corresponds to the value of the signal. The indication can be reversed by setting the relevant parameter *Left/Center/Right/BargraphMode* to ReversedBar.

A bar is an indication where all segments are lit from the bottom up to a certain level.

A point is an indication where only two segments are lit. The topmost of the two segments represents the value indicated.

If ReversedBar or ReversedPoint is chosen the value indicated is one hundred (per cent) minus the value of the signal connected to the bar graph.

For example, suppose the Process Value is 40%. The left bar graph which indicates Process Value has 30 segments therefore the number of segments lit is given by:

$$\frac{40}{100} \times 30 = 12 \text{ segments.}$$

For reversed indication the number of segments lit is given by:

$$\frac{100-40}{100} \times 30 = 18 \text{ segments.}$$

This is shown in the figures below for *LeftBargraph* set to Bar, ReversedBar, Point and ReversedPoint modes.

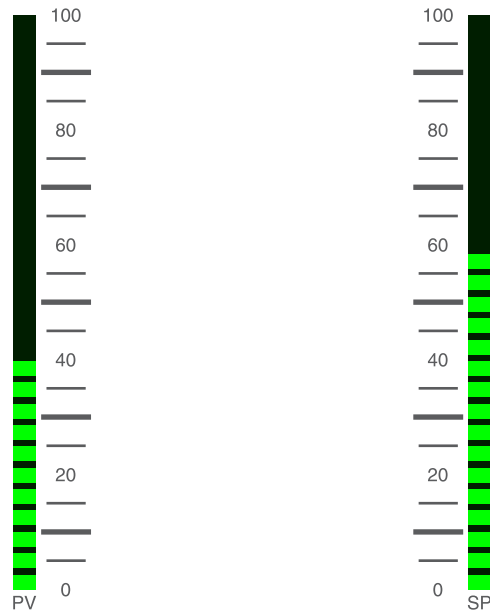


Figure 62 Segments Lit in Bar Mode (Left) and in Reversed Bar Mode (Right).

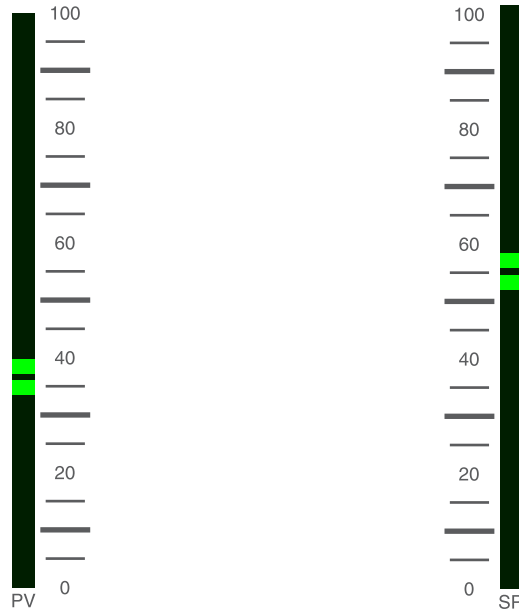
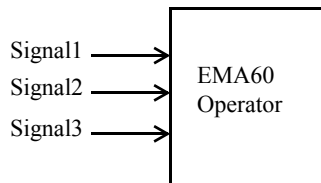


Figure 63 Segments Lit in Point Mode (Left) and Reversed Point Modes (Right).

Available in **EMA60 Operator Block**

ECA06	ECA60	ECA600	EMA60
-	-	-	X



The Operator Block (OP) handles the presentation of all information on the controller front panel.

The Operator block in EMA60 has three analog inputs: *Signal1* is indicated on the left bar graph (PV), *Signal2* is indicated on the center bar graph (SP) and *Signal3* is indicated on the right bar graph (OUT).

Connection to the analog source is made via the input *Signalx* (where $x = 1, 2$ or 3). If *Signalx* is off then the corresponding bar graph is turned off. The text “Off” is shown on the display instead of a numerical value when the signal is accessed from the Operator option List.

Inputs	
Name	Connection
Signal1	Off, Analog
Signal2	Off, Analog
Signal3	Off, Analog

Table 145 EMA60 Operator Block Inputs.

Display

By pressing the next key on the front panel the value of *Signal1*, *Signal2* or *Signal3* can be presented on the display, scaled in the selected engineering units (EGU). The value of the *Signal* input is always truncated internally in the range 0.0% to 100.0%. The scaling into EGU is made using the truncated range.

The EGU scaling is determined by the parameters *MinValue* and *MaxValue*. Five significant digits are available.

For example:

- Setting the *NumberOfDecimals* parameter to two (2) makes *MinValue* and *MaxValue* range from -999.99 to 999.99 .
- Setting the *NumberOfDecimals* parameter to four (4) makes *MinValue* and *MaxValue* range from -9.9999 to 9.9999 .



Note: If the value to be displayed exceeds five digits at any time, the rightmost digit is dropped.

The parameter *Unit* adds a unit label on the display.

Tag is a text string displayed on the top line of the display in Operator Mode when the section is selected for display. *Tag* can hold twenty (20) characters, but the top line cannot display all of them. If the length of *Tag* is less than or equal to 16, the full length is displayed. If the length of *Tag* is greater than 16, then 15 characters are displayed followed by the character “...”.

Bar Graphs

Each bar graph displays the corresponding input value. Alarm indicators can also be shown on the bar graphs. The parameter *BarGraphAlarm* determines which alarm indication is presented on the bar graph. This is selectable from the available Absolute Alarms and Deviation Alarms.

The parameter *BarGraphMode* allows you to select the same modes for each bar graph as described for the ECA controllers.

Hand Key

On the EMA60 it is possible to lock the Hand key so that the control mode is always manual by means of the parameter *HandKey*.

Parameters		
Name	Setting	Unit
xBarGraph Mode	Bar, ReversedBar, Point, ReversedPoint	None
NumberOfDecimals	$0 \leq \text{NumberOfDecimals} \leq 4$	None
MinValue	$-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$	None
MaxValue	$-99999 \leq \text{MinValue} \leq \text{MaxValue} \leq 99999$	None
Unit	%, °C, °F, K, W, kW, MW, J, kJ, MJ, rpm, Pa, kPa, MPa, bar, mbar, bara, l, l/s, l/min, l/h, Psi, Psig, Psia, mmWG, "WG, mmvp, mvp, mmH ₂ O, mmHg, "Hg, kp/cm ² , N/m ² , kg/cm ² , kg/h, t/h, Nm ³ /h, m ³ /h, t/24h, m/s, m/min, μs, μs/cm, ppm, pH, mg/l, feet, inches, meters, tons, lbs, lbs/sec, lbs/min, lbs/hr, ft/sec, ft/hr, BTU/hr, BTU, ft ³ /sec, ft ³ /hr, gals, gal/sec, gal/hr, cu.ft, MPPH, cc, cm, GPM.	None
xBarGraph Alarm	off, AbsoluteAlarm1, AbsoluteAlarm2, AbsoluteAlarm3, DeviationAlarm1, DeviationAlarm2, DeviationAlarm3	None
Tag	Text	None

Parameters		
Name	Setting	Unit
HandKey	Normal, LockInManual	None

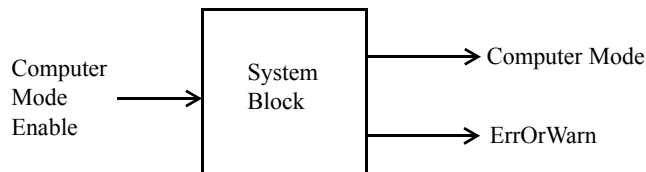
Table 146 EMA60 Operator Block Parameters.

Note: The “x” character in the BargraphMode and BargraphAlarm parameters represents “Left”, “Center” or “Right”.



Available in			
ECA06	ECA60	ECA600	EMA60
X	X	X	X

System Block



The System Block (SY) monitors the controller communications and status.

Status

The status block has one digital input, *ComputerModeEnable* and two digital outputs, *ComputerMode* and *ErrOrWarn*. The block does not hold any user definable parameters. *ComputerMode* can be set to give Local or Computer control. For details of *ErrOrWarn* please refer to Appendix B Error, Warning and Information Messages.

Inputs		
Name	Connection	Default
Computer Mode Enable	Off, Digital	Off

Table 147 System Status Input

Outputs	
Name	Type
Computer Mode	False = Local, True = Computer
ErrOr Warn	Refer to Appendix B for more information

Table 148 System Status Outputs.

COM Setup

There are five parameters associated with communications:

- *ComProtocol* which may be set to COMLI or MODBUS RTU.
- *Identity* which may be set to any number within the range 1 to 247
- *RS-485BaudRate* which may be set to a maximum value of 38,400 baud.
- *RS-232BaudRate* which may be set to a maximum value of 38,400 baud
- *WriteEnable* which may be set to On or Off.

Communication modes can be switched both from the front and via the *ComputerModeEnable* input (described above). The Front Switch (via the Operator Option List) and *ComputerModeEnable* have equal priorities.

If *WriteEnable* is set to Off, a supervisory system can not write to the controller's registers when the Communication Mode is Local.

Communication Mode	Write Operations	Operator Front Panel Access
Local	Impossible	Unrestricted
Computer	Possible	Restricted

Table 149 WriteEnable is Off.



Note. See Appendix E for a full list of communication I/O bits and registers

If *WriteEnable* is set to On, the Communication Mode has no impact on a supervisory system's ability to write to the controller's registers. The Communication Mode simply locks the front panel.

Communication Mode	Write Operations	Operator Front Panel Access
Local	Possible	Unrestricted
Computer	Possible	Restricted

Table 150 WriteEnable is On.

Refer to Appendix E for further details on communication.

Parameters		
Name	Setting	Unit
Identity	$1 \leq \text{Identity} \leq 247$	None
RS-485BaudRate	Off, 1200, 2400, 4800, 9600, 19200, 38400	Baud
RS-232BaudRate	Off, 1200, 2400, 4800, 9600, 19200, 38400	Baud
ComProtocol	COMLI, MODBUS RTU	None
WriteEnable	Off, On	None

Table 151 Communications Parameters.

Miscellaneous

The system block contains a number of miscellaneous parameters whose functions are summarised in the table below.

Parameters		
Name	Setting	Unit
Setup	Please refer to the booklet “Preset Configurations” for details.	None
SampleTime	$0.030 \leq \text{SampleTime} \leq 0.500$	Seconds
Password	Off, $1 \leq \text{Code} \leq 9999$	None

Table 152 System Block Miscellaneous Parameters.

Setup gives you access to the Preset Configurations programmed into the unit. Please refer to the separate booklet for details of the Preset Configurations available in each controller. The factory setting is *Default*.



Note: When selecting the *SetUp* parameter from the available Preset Configurations the entire configuration of the controller unit is set according to predefined parameters. The values in this list vary from model to model.

The *SampleTime* parameter determines how often the blocks in the columns are to be updated. The system scans these once every *SampleTime* seconds. If the time it takes to complete a full cycle of processing is greater than *SampleTime*, the ‘SampleTime Error’ is set. The *SampleTime* is automatically increased so that the system can complete a full cycle of processing without errors.

It is advisable to tune the controller (using the Autotuner) after an occurrence of the ‘SampleTime Error’, as an adjustment of the *SampleTime* affects loop dynamics.

Password allows you to set an access code which must be entered before Configuration mode can be selected. If the code is entered correctly the user has read/write access to the Configuration mode; if the code is entered incorrectly the user has read access only.

Display Language

The Display Language section contains two parameters, *Select* and *Edit*.

Parameters		
Name	Setting	Unit
Select	English, Svenska, Deutsch	None
Edit	Please refer to the Edit section below	None

Table 153 System Block Miscellaneous Parameters.

The language in which all messages appear on the display is selected by setting the parameter *Select*.

Edit

The *Edit* parameter allows you to edit all text strings in the selected language. When this parameter is selected and you press the OK key, the second line of the display shows an editable text string. For example:

```

Edit
Process Value
Process Value
Cancel Next OK

```

Figure 64 Using the Edit Parameter.

In the example above, the string “Process Value” is being edited. The second line shows the current setting of the string, while the third line shows the edited version. The dark background of the letter “P” on the third line indicates that this is the letter whose value will change when you press the Up/Down keys. To edit the next character, press the Next key.



Warning – User Edited Text: As the you press the Up/Down keys to browse the vocabulary, you can choose to edit any string. The edited texts are stored as a fourth language and retained in permanent storage during power loss, but can not be selected via the *Select* parameter. Once you have selected a language from the *Select* list, the edited texts are lost!

3.7 Status Signals

The various status signals and their locations within the controller are listed in the table below.

Status Signal	Block Location
Ramp	Set Point Block
Absolute Alarms	Absolute Alarm Block
Deviation Alarms	Deviation Alarm Block
Computer/Local Mode	System Block
ErrorOrWarningActive	System Block
Control Mode	Output Configuration Block

Table 154 Status Signal Locations.

3.8 Test Mode

To enter Test Mode press the OK key for a few seconds.

In Test mode, you can select from a list of choices using the Next key:

- Display test (Display and bar graph test. All pixels and LEDs are lit).
- Working Time (Time elapsed since power up. Days, hours, minutes, seconds).
In the example below, the unit has been powered up for 76 days, 14 hours, 32 minutes and 10 seconds.

Working Time
76d 14:32:10

Figure 65 Working Time Example.

To exit Test mode press the Cancel key. The unit returns to Operator mode.

Chapter 4

Maintenance

This chapter contains maintenance instructions to ensure the continued safe operation of the ECA06, ECA60, ECA600 and EMA60 controllers.

4.1 General

The operation of the controller should be checked once a year.

The controllers require no other special maintenance or any periodic calibration.

4.2 Annual Check

Once a year perform the following checks:

- 1 Visually inspect the unit for any signs of physical damage. Return any damaged unit to ABB for repair.
- 2 Review the configuration to ensure that it is still optimal for the control application.

Chapter 5

Technical Data

5.1 Type

- Electronic controller with configurable functions.

5.2 Control Functions

Selectable from:

- PID controller with fixed parameters (conventional controller).
- PID controller with continuously adaptive parameters settings.
- PID controller with gain scheduling (combined with adaptive control).
- pPI controller (for dead-time processes).

5.3 Control Action

- Direct or reversed.

5.4 System Functions

Selected by configuration, some functions can be selected during operation.

Operating Modes

- Normal Operation or Configuration, selected via the front panel.
- Auto/Manual/Forced Output/Tracking control selected via the front panel, an external digital signal, or a remote computer.
- Computer/Local control, selected via the front panel or an external digital signal.

- Internal/External Set Point, selected via the front panel or an external digital signal.

All operating mode switching is normally bumpless.

Process Value

- Selectable signal range.
- Arithmetic functions can be configured.

Set Point

- Two types of Set Point: Internal (ISP) or External (ESP).
- Set Point limiting.
- Set Point tracking.
- Set Point ramp (simple and advanced).
- Remote reading of Set Point.

Output Signals

- Four types of output signal – Simple analog, Double analog (split range), two-state or three-state pulse output (with or without potentiometer feedback).
- Output signal limiting.
- Restart value.
- Forced control.
- Output signal on AI/AO signal break.
- Output signal tracking.
- Remote reading of output signal.
- Reversible analog display.
- Digital output signal control (DDC)

Alarms

- Absolute alarms with up to four alarm points or deviation alarms.

Computer Interface

- Can be used for control and/or data logging.
- Can be configured remotely.

Arithmetic Functions

- Addition/Subtraction (AddSub).
- Multiplication/Division (MulDiv)
- Exponent (Exp).
- Pressure/Temperature compensation (COM).
- Selection of minimum signal (MIN).
- Selection of maximum signal (MAX).
- Signal selection (SEL).
- Delay (DEL).

Logic Functions

- AND.
- OR.
- XOR.
- NOT.
- R/S flip-flop.

Other Functions

- Min.
- Max
- Select.
- Delay.
- Inc/Dec.
- Level Detector.
- Linear.

- Integrator
- Time Measurement
- Rate Limiter
- Timer
- Counter

Miscellaneous

- Remote reading of selectable internal signals.
- Password to protect configuration.
- Fault Indications.
- Controller and/or configuration resetting.
- Alarms, fault signals and operating modes can be indicated via digital outputs.

5.5 Input Signals

AI1, AI2, AI3, AI4, AI5

- Five separate analog inputs.
- Fixed or floating reference ground.
- Floating inputs between -5 V and $+17\text{ V}$ relative to supply ground.
- Signal ranges $0\text{--}20\text{ mA}$ or $4\text{--}20\text{ mA}$, alternatively $0\text{--}5\text{ V}$, $1\text{--}5\text{ V}$, $0\text{--}10\text{ V}$ or $2\text{--}10\text{ V}$.
- Input resistance for current inputs $250\ \Omega$ with voltage inputs $>500\text{ k}\Omega$.

Inaccuracy

- $< 0.2\%$ of the measuring range.

Temperature Dependency

- $< 0.01\%$ of the measuring range per $^{\circ}\text{C}$ within the range 0 to 50°C .

DI1, DI2, DI3, DI4

- Four opto-isolated digital inputs for 24 V .

- Inactive = 0 to 5 V, active = 16 to 30 V

5.6 Output Signals

AO1, AO2, AO3

- Three analog outputs, 0–20 mA or 4–20 mA. Loading 0 to 650 Ω .
- Load dependence < 0.2 % for a change from 0 to 60 Ω .

DO1, DO2, DO3, DO4, DO5, DO6

- Six digital outputs, 24 V 250 mA.

5.7 Computer Communication

- RS485 serial interface, COMLI or MODBUS RTU protocol.
- RS232 serial interface, COMLI or MODBUS RTU protocol.

5.8 Safety Functions

Supply Failure

- Restart from present position (3-state pulse output) or a preset position.

Input/Output Signal Failure

- Failure of the analog input/output signal initiates an alarm.

5.9 Indications

Process and Set Point Values

- Analog bar graph display with 3.33 % resolution.
- Calibrated 0 to 100 %.
- Display normally shows Process Value. Automatically reverts to showing the Process Value 5 seconds after the user has finished examining other values.

Output Signal

- Analog bar graph display with 5 % resolution.
- Calibrated 0 to 100 %.

- Bar graph also shows actuator position with increase/decrease output.

Alarms

- Flashing segment on bar graph and message on display.

Set Point Indication

- Digital Set Point value indicated on the display.

Cascade Control (Internal)

- Slave controller indicated by S in the display.
- Master controller indicated by M in the display.

Function Keys

- Fixed function keys in Normal operating mode, variable function keys in Configuration Mode.
- Manual On/Off key includes a red light emitting diode to indicate manual control is On.

Configuration

- Position in the Configuration tree and parameter values indicated on the display.

5.10 Mains Supply

A.C.

- 110 V, 120 V, 220 V or 240 V a.c. $\pm 10\%$, 50/60 Hz.
- Power consumption 17 VA maximum.

D.C.

- 24 V d.c. $\pm 10\%$
- Power consumption 10 W maximum.
- Reverse polarity protection via rectifier (refer to on page 27).
- It is possible to supply the controller with 19 V a.c. rms via d.c. terminals (z4 and z6).
- D.C. supply should be fused appropriately.

5.11 Transmitter Power Supply

- 24 V d.c., 100 mA.
- Potentiometer feedback when three-state pulse output, maximum 20 mA at 650 Ω .

5.12 Environment

Ambient Operating Temperature

- +5 to +55°C.

Sealing Classification

- In accordance with IEC 529
- IP20 in all directions.
- Kit available for IP65 protection from the front in sealed panel.

5.13 Installation

Two options available:

- Flush mounting in panel with mounting plates IP65 with gasket.
- Front-inserted plug-in cassette available for connections with front inserted controller unit.

5.14 Electrical Connections

- Terminal blocks to accommodate wire of maximum size 1.5 mm².

5.15 Relevant Standards

The Controller meets the Low Voltage Directive LVD 73/23/EEC according to the standard EN 61131-2, and meets the EMC directive EMC89/336/EEC according to Immunity EN 50082-2 and Emission EN 50081-1.

5.16 Physical Characteristics

- Width 72 mm.
- Height 144 mm.
- Length 235 mm.
- Weight 2.5 kg.

Appendix A

Application Examples

This appendix contains application examples which demonstrate how to configure the controllers for a number of typical applications. The examples are:

- 1 Level Control with Feed Forward and Gain Scheduling.
- 2 Temperature Control and Communication with COMLI.
- 3 Furnace Pressure Control with Three State Pulse Output and Position Potentiometer.

In the step by step instructions, each parameter setting is shown according to the following example.

Analog Inputs → AI1 → Mode = 0–20 mA

Here the user first presses the Up and Down keys to select “Analog Inputs” where he presses the OK key to accept the selection. Then he presses the Up and Down keys to select “AI1”, and so on. The actual parameter setting is made in the last step where “Mode” is set to the value “0–20 mA” in the example above.

The configuration steps can be performed in any order but are presented in the general logical order inputs/processing/outputs.

Each configuration step is described in full, but remember that you do not have to go back to the first level of the tree each time.

When you are analysing your own controller application you may find the following procedure useful:

- 1 Make a rough sketch on ordinary paper.
- 2 Identify which blocks are needed to solve the problem.
- 3 Use a Configuration Work Sheet and connect the blocks by drawing lines between the outputs and inputs (see Appendix E, Configuration Work Sheet).

- 4 Inspect the default parameter settings of each block, and determine what changes need to be made.
- 5 Write down all parameter changes on paper.
- 6 Configure the controller using your filled out Configuration Work Sheet and parameter changes sheet. These two papers will be a useful record of your solution to the control problem.

A.1 Level Control with Feed Forward and Gain Scheduling

Description

This example describes level control of a tank with two inlets and one outlet with pump. One of the inlets is controlled by a control valve from the process controller, while the other inlet is not controlled but measured by an analogue transmitter.

An uncontrolled but measurable disturbance is thus affecting the level in the tank. If the disturbance is constant, it can be compensated for without any special functions. But if it is a varying disturbance as in this case, the feed forward function can be utilized.

The control valve is non-linear which means it will give different gain at different flow rates. To optimize the controller parameters to the different flow rates, gain scheduling based on the output signal is used.

Mechanical Installation

Please refer to chapter 2 of this manual for details of the mechanical installation.

Definition of Control Problem

The object is to control a fluid flow according to the diagram below. The flow is measured with a differential pressure transmitter using an orifice flange and regulated by a control valve.

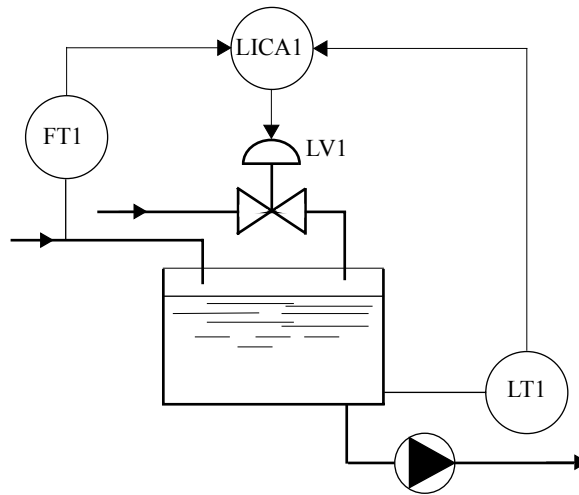


Figure 1 Example of Level Control with Feed Forward and Gain Scheduling.

To solve this problem an ECA60 or an ECA600 controller may be used.

The Set Point is to be set at the front panel of the controller.

Preset Configurations

Is there a preset configuration available that solves this application? Look in the Preset Configuration booklet.

The default Configuration for any of the controllers almost provides the solution – one input for the Process Value and one output for the Control Output. A second input can easily be configured.

Identify the terminals used for the different signals in the Default Configuration – using the same terminals saves work and time.

Selection of I/O

The transmitters provide 4–20 mA outputs. For this example we will assume that the flow transmitter is calibrated for a flow range of 0–9 m³/min. They are powered with 24 V DC from the controller unit.

Three I/O terminals are required:

- One analogue input from the level transmitter (Process Value).

- One analogue input from the flow transmitter.
- One analogue output to control the valve (Control Output).

For the level transmitter choose AI1 (as in the Default Configuration) and for LV1 choose AO1 (as in the Default Configuration). For the flow transmitter choose AI2.

Electrical Installation

Power Wiring

For details of the power installation please refer to chapter 2 of this manual.

Signal Wiring

Connect the transmitter and I/P converter to the controller unit according to the figure below, using screened cable.

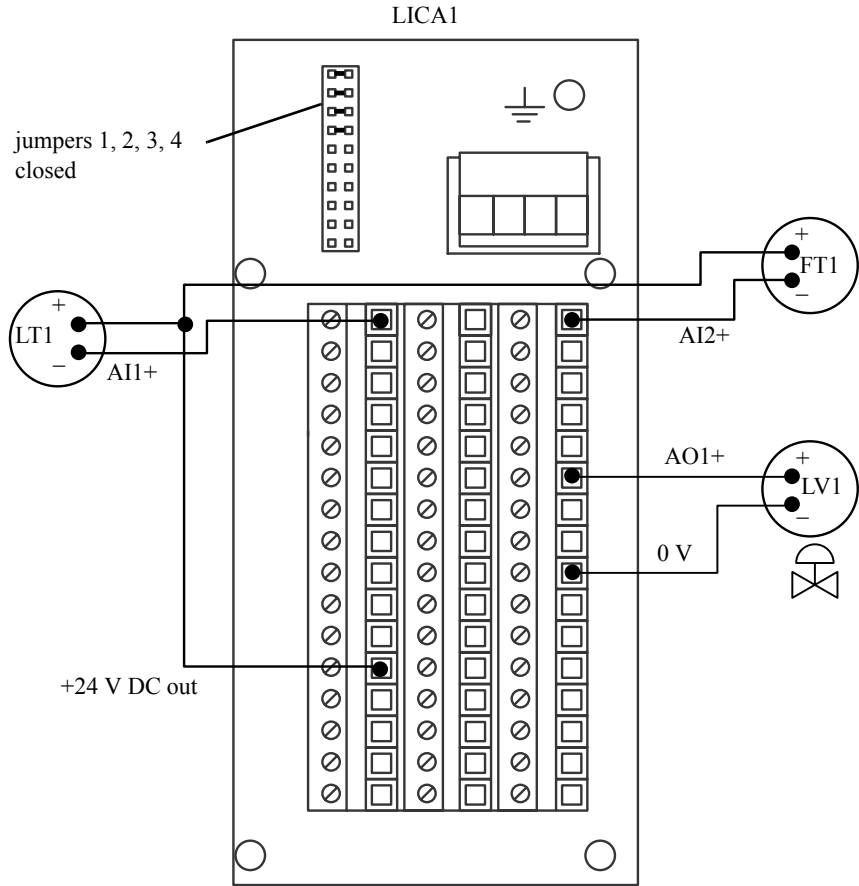


Figure 2 Connection diagram.

Configuration

Preset Configuration

Select the default setup from the Preset Configurations.

- 1 System → Miscellaneous → Setup = Default.

Miscellaneous SetUp Default	
Cancel	OK

Inputs

AI1 is set by default to 4–20 mA current input so there is no need to change it, but it can be given a suitable name. However, AI2 must be configured:

- 1 Inputs → AI1 → Identity = Level

AI1 Identity Level	
Cancel	OK

- 2 Inputs → AI2 → Mode = 0–20mA

AI2 Mode 0–20 mA	
Cancel	OK

- 3 Inputs → AI2 → Identity = Flow

AI2	
Identity	
Flow	
Cancel	OK

Filtering

Set the time constant for AI1 to 12 seconds and AI2 to 13 seconds:

- 1 Inputs → Analog Inputs → AI1 → FilterTime = 12 seconds.

AI1: Level	
FilterTime	
12	
Cancel	OK

- 2 Inputs → Analog Inputs → AI2 → FilterTime = 13 seconds.

AI2: Flow	
FilterTime	
13	
Cancel	OK

Process Value

To configure the range and units and define which signal is the Process Value:

- 1 Operator → (PID1 – ECA600 only →) Process Value → NumberOfDecimals = 1.

Process Value	
NumberOfDecimals	
1	
Cancel	OK

- 2 Operator → (PID1 – ECA600 only →) Process Value → MaxValue = 10.0.

Process Value	
MaxValue	
10.0	
Cancel	OK

- 3 Operator → (PID1 – ECA600 only →) Process Value → Unit = mvp.

Process Value	
Unit	
mvp	
Cancel	OK

Absolute Alarms

Set the absolute low alarm to 2 mvp (i.e. 20%) and the high alarm to 8 mvp (i.e. 80%).

- 1 Alarms → Absolute Alarms → Absolute Alarm 1 → Mode = On.

Absolute Alarm 1	
Mode	
On	
Cancel	OK

- 2 Alarms → Absolute Alarm → Absolute Alarm 1 → LoLimit = 20.

Absolute Alarm 1	
LoLimit	
20%	
Cancel	OK

- 3 Alarms → Absolute Alarm → Absolute Alarm 1 → HiLimit = 80.

Absolute Alarm 1	
HiLimit	
80%	
Cancel	OK

Feed Forward

The Feed Forward mode must be set to StaticGain. The signal that is fed forward is the flow measured by the transmitter FT1, which is connected to AI2. The Gain must be tested by trial and error but in this example it is set to 2.

- Control → Parameters → Feed Forward → Mode = StaticGain.

Feed Forward	
Mode	
Static Gain	
Cancel	OK

- Control → Parameters → Feed Forward → FeedForwardSignal = AI2[Value].

AI2: Level	
FeedForwardSignal	
AI2[Value]	
Cancel	OK

- Control → Parameters → Feed Forward → Gain = 2.

Feed Forward	
Gain	
2	
Cancel	OK

Gain Scheduling

The Gain Scheduling must be enabled and the limits set. The non-linearity in the control valve in this example is divided into three ranges with limits at 40% and 60%.

- Control → Parameters → Gain Scheduling → Mode = On.

Gain Scheduling	
Mode	
On	
Cancel	OK

- Control → Parameters → Gain Scheduling → Limit1 = 40%.

Gain Scheduling	
Limit1	
40%	
Cancel	OK

- Control → Parameters → Gain Scheduling → Limit2 = 60%.

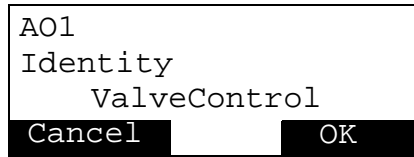
Gain Scheduling	
Limit2	
60%	
Cancel	OK

- Control → Parameters → Gain Scheduling → GainSchedulingSignal = AI1[Value].

Gain Scheduling	
GainSchedulingSignal	
AI1[Value]	
Cancel	OK

Outputs

- Output → Analog Outputs → AO1 → Identity = ValveControl.



Configuration Complete

When Configuration is complete repeatedly press the Cancel key until you return to Operator Mode.



Startup

The controller starts automatically as soon as you leave the configuration mode, using the values you have just configured.

The configuration task is always a part of the installation process. This task can be significantly simplified if a Preset Configuration is used. Look at the preset Configurations at the planning stage. They may give you ideas and help you find the most efficient solution to your control problem. If you let the preset Configurations influence your choice of solution and I/O you may save some valuable time!

You may not find the perfect configuration for your application among the preset configurations but choose the one closest to the solution you require and make the necessary adjustments.

A.2 Temperature Control and Communication with COMLI

Description

This example describes a control loop for temperature control of a medium which is heated by a hot water heat exchanger.

Definition of the Control Problem

The object is to regulate the temperature of one medium by controlling the flow of another. The temperature controller communicates with a host computer or a control system via COMLI.

The hot water pump is started and stopped from the computer via a digital signal. When the pump is running it sends a digital acknowledge signal.

The flow of the heated medium is measured by the flow transmitter FT1 which produces an analogue signal. The temperature of the medium is measured via the temperature transmitter TT1 which also produces an analogue signal.

The computer sets the set point for the controller and registers the temperature on the outgoing heated medium via the COMLI communication.

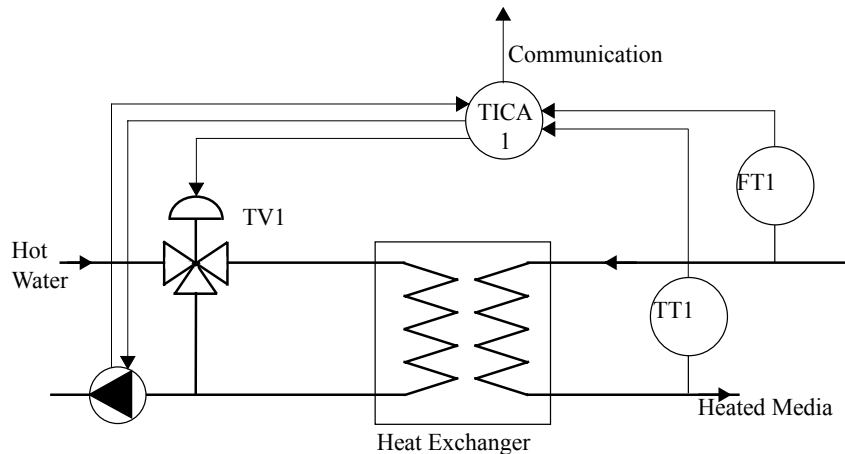


Figure 3 Example of Temperature Control with COMLI

Mechanical Installation

Please refer to chapter 2 of this manual for details of the mechanical installation.

Preset Configurations

Is there a Preset Configuration available that solves this application? Look in the Preset Configuration booklet.

We can adapt the default configuration for this application.

Selection of I/O

We need two analogue inputs, one digital input, one digital output and one analog output.

The temperature transmitter TT1 has a 0–5 V output; the flow transmitter FT1 has a 0–20 mA output.

The inputs and outputs should be connected as shown in the figure below.

The temperature transmitter TT1 is connected to AI1 and has its own independent power supply.

The flow transmitter FT1 connected to AI2 is to take its power supply from the controller. To configure AI2 as a single ended current input, jumpers 3 and 4 must be closed.

Electrical Installation

Power Wiring

For details of the power wiring please refer to chapter 2 of this manual.

Signal Wiring

Connect the signal wiring as shown in the figure below.

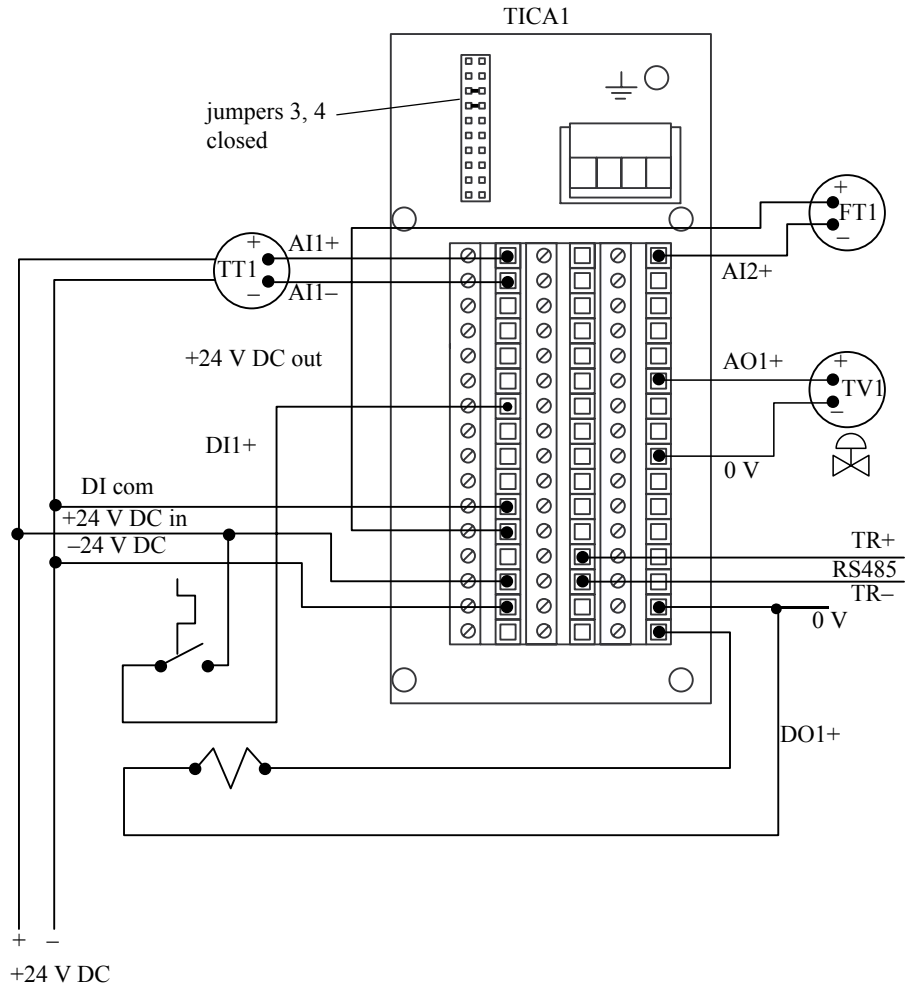


Figure 4 Connection of Transmitter, Actuator and COMLI Communication.

Configuration

Preset Configuration

Select the Default Setup from the Preset Configurations.

- 1 System → Miscellaneous → Setup = Default.

Miscellaneous SetUp Default
Cancel OK

Inputs

The TT1 output signal range is 0–5 V (for 0–400°C) and the FT1 output signal range is 0–20 mA:

- 1 Inputs → Analogue Inputs → AI1 → Mode = 0–5V.

AI1: Ana In 1 Mode 0–5V
Cancel OK

- 2 Inputs → Analogue Inputs → AI1 → Identity = Temp.

AI1: Ana In 1 Identity Temp
Cancel OK

- 3 Inputs → Analogue Inputs → AI2 → Mode = 0–20mA.

AI2: Ana In 2 Mode 0–20 mA
Cancel OK

- 4 Inputs → Analogue Inputs → AI2 → Identity = Flow.

AI2: Ana In 2
Identity
Flow
Cancel <input type="checkbox"/> OK

Process Value

The required temperature range is 0–400°C, with no decimal places. There is no need to change PVMinValue as its default value is zero:

- 1 Operator → (PID1 – ECA600 only →) Process Value → NumberOfDecimals = 0

Process Value
NumberOfDecimals
0
Cancel <input type="checkbox"/> OK

- 2 Operator → (PID1 – ECA600 only →) Process Value → MaxValue = 400

Process Value
MaxValue
400
Cancel <input type="checkbox"/> OK

- 3 Operator → (PID1 – ECA600 only →) Process Value → Unit = °C

Process Value
Unit
°C
Cancel <input type="checkbox"/> OK

Computer Communications

To configure the COMLI communications, give the signal an identity number and select a suitable communications protocol and Baud rate for the transmission:

- 1 System → Com Setup → Com Protocol → COMLI

Com Setup Com Protocol COMLI	
Cancel	OK

- 2 System → Com Setup → Identity = 1

Com Setup Identity 1	
Cancel	OK

- 3 System → Com Setup → RS-485 BaudRate = 9600

Com Setup RS-485 BaudRate 9600	
Cancel	OK

Autotuner

The Autotuner is to be used to set the PID parameters so it must be enabled:

- 1 Control → Parameters → Tuner → Mode = On.

Tuner Mode On	
Cancel	OK

Outputs

- 1 Outputs → Analog Outputs → AO1 → Identity = ValveControl.

```

AO1: Ana Out 1
Identity
      ValveControl
Cancel      OK

```

- 2 Outputs → Digital Outputs → DO1 → Mode = PositiveLogic.

```

DO1: Dig Out 1
Mode
      PositiveLogic
Cancel      OK

```

- 3 Outputs → Digital Outputs → DO1 → Signal = DIC1[Status].

```

DO1: Dig Out 1
Signal
      DIC1[Status]
Cancel      OK

```

- 4 Outputs → Digital Outputs → DO1 → Identity = Switch.

```

DO1: Dig Out 1
Identity
      Switch
Cancel      OK

```

Configuration Complete

When Configuration is complete repeatedly press the Cancel key until you return to Operator Mode.

```

          ISP
0.0 °C

```

Startup

The controller starts automatically as soon as you leave the configuration mode, using the values you have just configured.

A.3 Furnace Pressure Control with 3-State Pulse Output

Description

This example shows how to control the pressure in a steam boiler furnace with a draught valve motor at the flue duct. High and low pressure alarms are required at 0.000 kPa and +0.500 kPa respectively.

Definition of Control Problem

The object is to control the furnace pressure as shown in the figure below. The valve motor M is of the increase/decrease type and has position feedback in the form of a potentiometer (Pot) on the valve. The pressure is measured by the transducer PT1. An ECA60 or an ECA600 can be used to solve this problem.

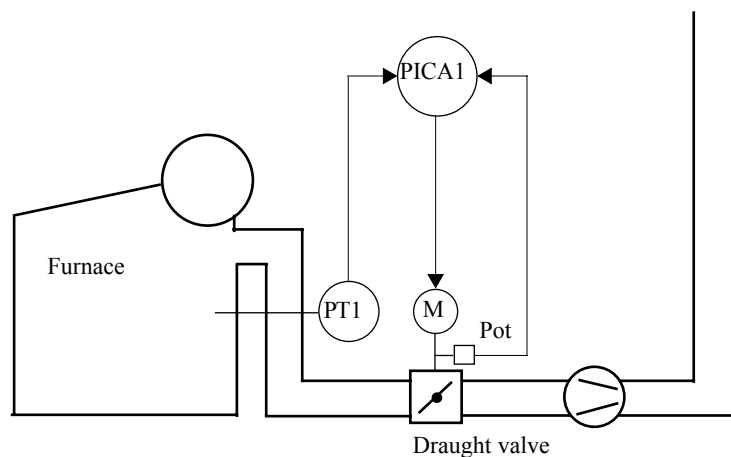


Figure 5 Furnace Pressure Controlled by a Draught Valve.

Mechanical Installation

Please refer to chapter 2 of this manual for details of the mechanical installation.

Preset Configuration

None of the preset configurations match our requirements exactly so we will start with the Default Setup and modify that.

Selection of I/O

The potentiometer can be connected to AI1 and AO1. Jumper 1 must therefore be closed.

The pressure transmitter PT1 does not have its own power supply and therefore must be fed from the controller. The pressure signal is connected to AI2.

The high and low alarms is configured to DO1 and DO2.

The motor is controlled by means of the three state pulse output on DO3, DO4, increase and decrease.

The Autotuner is used to set the PID parameters for this application.

Electrical Installation

Power Wiring

For details of the power wiring please refer to chapter 2 of this manual.

Signal Wiring

Connect the signal wiring as shown in the figure below.

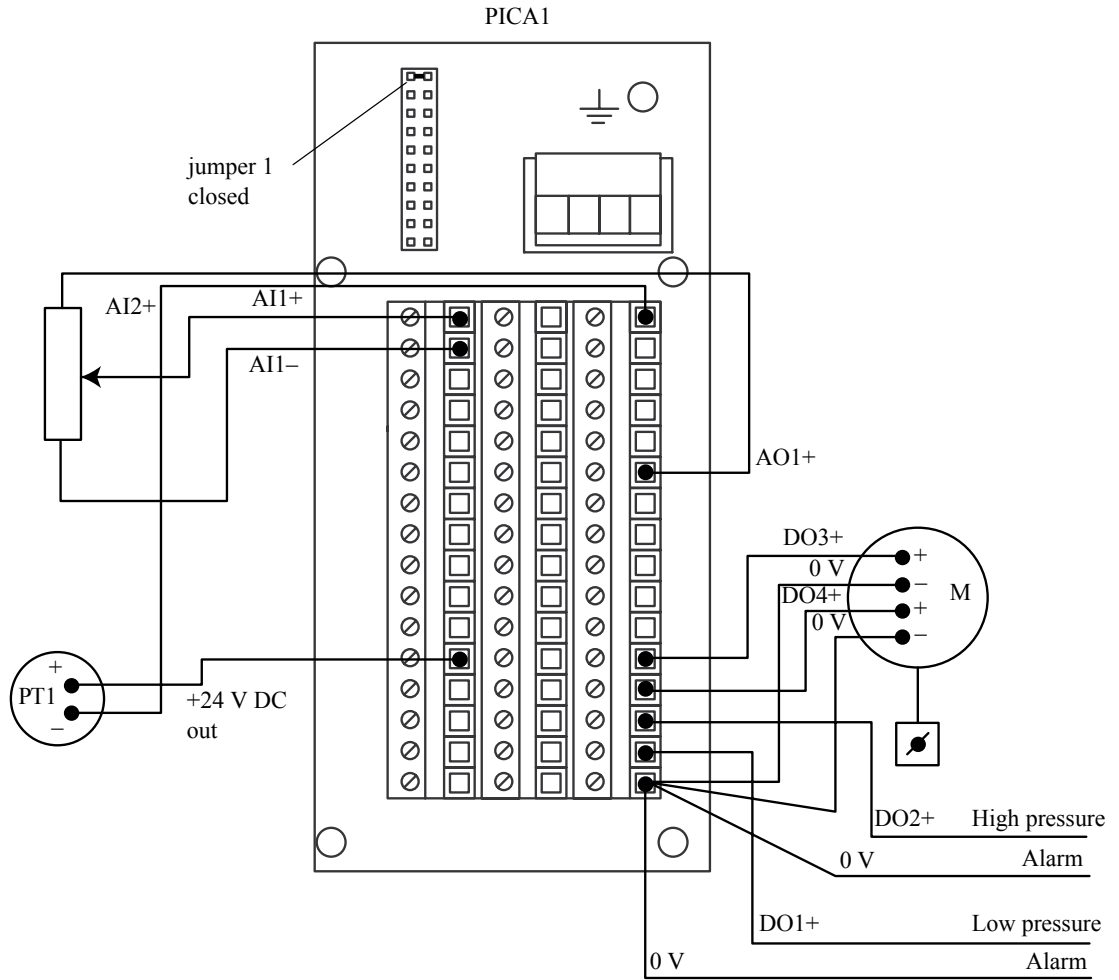


Figure 6 Connection of Transmitter, Actuator Potentiometer, and Increase/Decrease Motor.

Configuration

When configuring the functions only those parameters with unsuitable default values have to be changed, therefore only these points are described below.

Preset Configuration

Select the default setup from the Preset Configurations.

- 1 System → Miscellaneous → Setup = Default.

Miscellaneous	
SetUp	
Default	
Cancel	OK

Inputs

Configure analogue inputs AI1 to accept the potentiometer feedback signal and AI2 to accept the pressure signal:

- 1 Inputs → Analog Inputs → AI1 → Mode = 0–5V.

AI1: Ana In 1	
Mode	
0–5V	
Cancel	OK

- 2 Inputs → Analog Inputs → AI1 → Identity = Feedback.

AI1: Ana In 1	
Identity	
Feedback	
Cancel	OK

- 3 Inputs → Analog Inputs → AI2 → Mode = 4–20mA.

AI2: Ana In 2
Mode
4-20mA
Cancel OK

- 4 Inputs → Analog Inputs → AI2 → Identity = Pressure.

AI2: Ana In 2
Identity
Pressure
Cancel OK

Filtering

Set the time constant for AI2 to 2seconds:

- 1 Inputs → Analog Inputs → AI2 → FilterTime = 2 seconds.

AI2: Pressure
FilterTime
2
Cancel OK

Process Value

The pressure signal is to be used as the Process Value and is in the range -1.0 to +1.0kPa:

- 1 Operator → (PID1 – ECA600 only →) Process Value → NumberOfDecimals = 1.

Process Value
NumberOfDecimals
1
Cancel OK

- 2 Operator → (PID1 – ECA600 only →) Process Value → PVMinValue = -1.

Process Value	
PVMinValue	
-1.0	
Cancel	OK

- 3 Operator → (PID1 – ECA600 only →) Process Value → PVMaxValue = 1.

Process Value	
PVMaxValue	
1.0	
Cancel	OK

- 4 Operator → (PID1 – ECA600 only →) Process Value → Unit = kPa.

Process Value	
Unit	
kPa	
Cancel	OK

- 5 SetPoint → Process Value → PVSignal = AI2[Value].

Process Value	
PVSignal	
AI2[Value]	
Cancel	OK

Absolute Alarms

Set the absolute low alarm to 0.000 kPa (i.e. 0%) and the high alarm to +0.500 kPa (i.e. 50%).

- 1 Alarms → Absolute Alarms → Absolute Alarm 1 → Mode = On.

Absolute Alarm 1 Mode	
On	
Cancel	OK

- 2 Alarms → Absolute Alarm → Absolute Alarm 1 → LoLimit = 0.

Absolute Alarm 1 LoLimit	
0%	
Cancel	OK

- 3 Alarms → Absolute Alarm → Absolute Alarm 1 → HiLimit = 50.

Absolute Alarm 1 HiLimit	
50%	
Cancel	OK

Autotuner

The Autotuner is to be used to set the PID parameters so it must be enabled:

- 1 Control → Parameters → Tuner → Mode = On.

Tuner Mode	
On	
Cancel	OK

Output Functions

Configure the Three State Pulse output function to provide motor running time 120 s, the minimum pulse length 1% and the dead band for the control error 2%:

- 1 Output Functions → (PID1, ECA600 only) → Type = ThreeStatePulse.

```

Output Functions
Type
  ThreeStatePulse
Cancel          OK
  
```

- 2 Output Functions → (PID1, ECA600 only) → ThreeStatePulse → ActuatorPosTime = 120s.

```

ThreeStatePulse
ActuatorPosTime
  120 s
Cancel          OK
  
```

- 3 Output Functions → (PID1/Master, ECA600 only) → ThreeStatePulse → ActuatorMinPulseLen = 1%.

```

ThreeStatePulse
ActuatorMinPulseLen
  1%
Cancel          OK
  
```

- 4 Output Functions → (PID1/Master, ECA600 only) → ThreeStatePulse → DeadBand = 2%.

```

ThreeStatePulse
DeadBand
  2%
Cancel          OK
  
```

- 5 Output Functions → (PID1/Master, ECA600 only) → ThreeStatePulse → Feedback = AI1[Value].

```

ThreeStatePulse
Feedback
  AI1[Value]
Cancel      OK

```

Outputs

The feed to the potentiometer should be set to 5.0mA which gives 0–5V output since the potentiometer resistance is 1000Ω:

- 1 Outputs → Analog Outputs → AO1 → Mode = Direct.

```

AO1: Ana Out 1
Mode
  Direct
Cancel      OK

```

- 2 Outputs → Analog Outputs → AO1 → FixedCurrent = 5 mA.

```

AO1: Ana Out 1
Fixed Current
  5mA
Cancel      OK

```

- 3 Outputs → Analog Outputs → AO1 → Identity = PotFeed.

```

AO1: Ana Out 1
Identity
  PotFeed
Cancel      OK

```

Configure DO1 and DO2 to the low and high pressure alarms:

- 1 Outputs → Digital Outputs → DO1 → Mode = PositiveLogic.

```

DO1: Dig Out 1
Mode
PositiveLogic
Cancel      OK

```

- 2 Outputs → Digital Outputs → DO1 → Signal = AA1 [LoAlarm].

```

DO1:Dig Out 1
Signal
  AA1 [LoAlarm]
Cancel      OK

```

- 3 Outputs → Digital Outputs → DO1 → Identity = LoAlarm.

```

DO1:Dig Out 1
Identity
  LoAlarm
Cancel      OK

```

- 4 Outputs → Digital Outputs → DO2 → Mode = PositiveLogic.

```

DO2:Dig Out 2
Mode
  PositiveLogic
Cancel      OK

```

- 5 Outputs → Digital Outputs → DO2 → Signal = AA1 [HiAlarm].

```

DO2:Dig Out 2
Signal
  AA1[HiAlarm]
Cancel      OK

```

- 6 Outputs → Digital Outputs → DO2 → Identity = HiAlarm.

```

DO2:Dig Out 2
Identity
      HiAlarm
Cancel      OK

```

Configure the controller to provide a three state pulse output from DO3 and DO4 for the motor increase/decrease signal:

- 1 Outputs → Digital Outputs → DO3 → Mode = PositiveLogic.

```

DO3:Dig Out 3
Mode
      PositiveLogic
Cancel      OK

```

- 2 Outputs → Digital Outputs → DO3 → Signal = OF1 [Increase].

```

DO3:Dig Out 3
Signal
      OF1 [Increase]
Cancel      OK

```

- 3 Outputs → Digital Outputs → DO4 → Mode = PositiveLogic.

```

DO4:Dig Out 4
Mode
      PositiveLogic
Cancel      OK

```

- 4 Outputs → Digital Outputs → DO4 → Signal = OF1[Decrease].

```
DO4:Dig Out 4  
Signal  
    OF1[Decrease]  
Cancel      OK
```

Configuration Complete

When Configuration is complete repeatedly press the Cancel key until you return to Operator Mode.

```
          ISP  
0 . 0    kPa
```

Startup

The controller starts automatically as soon as you leave the configuration mode, using the values you have just configured.

Appendix B

Error, Warning and Information Messages

B.1 Message Types

Four types of messages can be shown on the display. In order of priority these are:

- System Errors, Errors and Warnings.
- Information.

The two types are handled separately by the system.

System Errors, Errors and Warnings

These categories of messages appear flashing on the *top line* of the display. When a System Error or an Error message appears, the bargraphs also flash.

In order of priority these messages are:

- 1 System Error.
- 2 Error.
- 3 Warning.

A higher priority message replaces a lower priority message on the display. When the higher priority message disappears (when the user acknowledges the message) the lower priority message reappears.



Figure 7 Display – System Error.

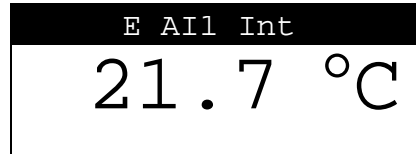


Figure 8 Display – Error Message.

Information Messages

This category of messages appears flashing on the *bottom line* of the display.

If the Information Message is an Absolute or Deviation Alarm, the associated alarm limit indication on the bargraphs also flashes.

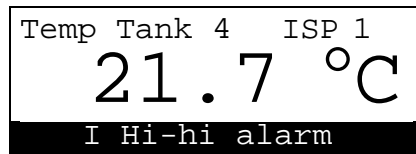


Figure 9 Display – Information Message.

B.2 Controller Reaction

System Errors and Errors force the Control Mode into manual. At the same time the Control Output is set according to the parameter *ErrorBehavior*.

It is not possible to change the Control Mode as long as the System Error or Error condition is valid.

B.3 Acknowledging Messages

Error and Warning Messages must be acknowledged to make the message disappear. If the condition that caused the message is still valid when it is acknowledged, the message will reappear.

Errors and Warnings can only be acknowledged in Operator Mode.

System Errors and Information Messages cannot be acknowledged. However, in some cases actions can be taken to make the message disappear. Refer to section B.4 on page cxxxxvii for details. System Errors are irreversible and Information Messages disappear when the condition causing the message disappears.

To acknowledge an Error or Warning proceed as follows;

- 1 Make sure you are in Operator Mode.
- 2 Access the Message List by multiple pressing of the Next key.
- 3 Acknowledge the Error or Warning by pressing the OK key.

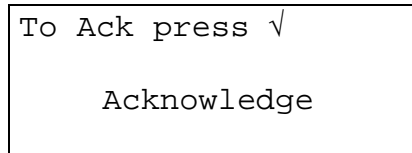


Figure 10 Display – Acknowledging a Message.

B.4 Error and Warning Descriptions

HW

If the system detects a hardware (HW) error, this error is set. If the signal wires of an analog input are connected with reverse polarity, the following actions should be taken:

- 1 Turn off power of controller unit
- 2 Install wires properly
- 3 Turn on power of controller unit again

FLASH Write

If a Write Error occurs in the FLASH memory, this error is set.

FLASH Erase

If an Erase Error occurs in the FLASH memory, this error is set.

FLASH Checksum

If a Checksum Error occurs in the FLASH memory, this error is set.

RAM

If a Checksum Error is found in the Configuration RAM memory, this error is set.

Sample Time Error

If the time it takes to complete a full cycle of processing is greater than the Sample Time configured, this error is set.

The Sample Time is automatically increased so that the system can complete a full cycle of processing without errors.

It is advisable to tune the controller (using the Autotuner) after an occurrence of this error as an adjustment of the Sample Time affects loop dynamics.

AIx Int/AOx Int

All Analog Inputs and Outputs can be configured to cause either a Warning or an Error when the connection is broken. If a warning is configured, a message is displayed according to the general procedure. If an Error is configured, the message is displayed and the Loop that is using the I/O takes action as defined by the parameter *ErrorBehavior*.

If an analog input or analog output configured for 4–20 mA (or 1–5 V, or 1–10 V) detects a current (or voltage) less than 2 mA (or 0.5 V, or 1.0 V), this is indicated as a warning or error. The analog output reacts at once, while the analog input reacts after five seconds.

If the signal wires of an analog input are connected with reverse polarity, a "Sys Err HW" error is set. For actions, refer to section B.4.1.

DO Overload

If the Digital Outputs are overloaded all Digital Outputs are shut and this error is set. When the user acknowledges this error, the Digital Outputs return to their normal function.

If the function of the digital outputs does not return to normal after acknowledgement, turn off the power, let the unit cool off for a few minutes, and then turn on the power again.

Calibration

If a Checksum Error is found in the Calibration factors, this error is set. The Calibration factors are used to produce accurate values from the Analog Inputs and to the Analog Outputs. If this condition is valid, an alternate set of factors is used, resulting in reduced accuracy for Analog I/O.

Arithmetic Error

If the calculations done in the Input Function Blocks result in a divide by zero or an arithmetic overflow, this error is set.

CLPM Osc LoopX

If the CLPM (Control Loop Performance Monitor) detects abnormal oscillation in the control loop, this warning is set. Please refer to page cxxxix for more information.

XX Source off

If any source or output is referenced when its mode is off, this warning is set.

Tuner Abort LoopX

If the Deviation is too great when Tuning is initiated, this warning is set and the Tuning procedure is interrupted.

Tuner PVNoise LoopX

If the Process Value is noisy, and threatens to make the Tuner render invalid PID Parameters, this warning is set.

Adapt Warn LoopX

If the Adaption algorithm detects abnormal PID Parameters, this warning is set. Control continues as usual. For further details refer to the warning concerning large parameter changes in the Autotuner section in the main part of this manual.

B.5 Information Messages

Alarm Messages

If a Process Value Alarm or a Deviation Alarm is configured, and the corresponding block detects an alarm condition, an information message is displayed.

Tuning

When tuning, the tuner informs the user about the current tuning phase as the tuning progresses.

B.6 Summary of Messages

The messages which appear on the display are summarised in the tables below.

System Error, Error and Warning Messages

Prior.	Condition	Sys	Err	Warn	Display	Comment
1	HW	X			Sys Err HW	Hardware Error
2	FLASH Write	X			Sys Err FLASH Write	Write Error in FLASH memory
3	FLASH Erase	X			Sys Err FLASH Erase	Erase Error in FLASH memory
4	FLASH Checksum	X			Sys Err FLASH Checksum	Checksum Error in FLASH memory
5	RAM	X			Sys Err RAM	Checksum Error in RAM memory
6	Sample Time Error		X		E Sample Time Error	Sample time overrun
7	AO1 Int		X	X	E AO1 Int	AO1 interrupted
8	AO2 Int		X	X	E AO2 Int	AO2 interrupted
9	AO3 Int		X	X	E AO3 Int	AO3 interrupted
10	AI1 Int		X	X	E AI1 Int	AI1 interrupted
11	AI2 Int		X	X	E AI2 Int	AI2 interrupted
12	AI3 Int		X	X	E AI3 Int	AI3 interrupted
13	AI4 Int		X	X	E AI4 Int	AI4 interrupted
14	AI5 Int		X	X	E AI5 Int	AI5 interrupted
15	DO overload		X		E DO Overload	DO overload
16	Calibration			X	W Calibration	Checksum Error in Correction Factors
17	Arithmetic error			X	W Arithmetic error	Divide by zero or overflow
18	CLPM Osc Loop1			X	W CLPM Osc Loop1	Please refer to page cxxxix.
19	CLPM Osc Loop2			X	W CLPM Osc Loop2	Please refer to page cxxxix.
20	OF Source off			X	W OF Source off	Output or Source off when called
21	OC Source off			X	W OC Source off	Output or Source off when called

Prior.	Condition	Sys	Err	Warn	Display	Comment
22	DA Source off			X	W DA Source off	Output or Source off when called
23	AA Source off			X	W AA Source off	Output or Source off when called
24	SP Source off			X	W SP Source off	Output or Source off when called
25	FO Source off			X	W FO Source off	Output or Source off when called
26	FL Source off			X	W FL Source off	Output or Source off when called
27	FA Source off			X	W FA Source off	Output or Source off when called
28	DU Source off			X	W DU Source off	Output or Source off when called
29	DIC Source off			X	W DIC Source off	Output or Source off when called
30	DI Source off			X	W DI Source off	Output or Source off when called
31	AU Source off			X	W AU Source off	Output or Source off when called
32	AIC Source off			X	W AIC Source off	Output or Source off when called
33	AI Source off			X	W AI Source off	Output or Source off when called
34	Tune abort Loop1			X	W Tune abort Loop1	Deviation too great to complete tuning
35	Tune abort Loop2			X	W Tune abort Loop2	Deviation too great to complete tuning
36	Tune PVNoise Loop1			X	W Tune PVNoise Loop1	PV noise disturbing tuner
37	Tune PVNoise Loop2			X	W Tune PVNoise Loop2	PV noise disturbing tuner
38	Adapt warn Loop1			X	W Adapt warn Loop1	Adaption problems
39	Adapt warn Loop2			X	W Adapt warn Loop2	Adaption problems

Table 1 System Error, Error and Warning Messages.



Note: For Errors/Warnings numbers 7 to 14, the leading character in the display is “E” for Errors and “W” for Warnings. In the table above, the “E” is chosen as an example.

Information Messages

Message	Display	Comment
Hi-hi alarm from block AA1	I Hi-hi alarm 1	Absolute Value Alarm
Hi-hi alarm from block AA2	I Hi-hi alarm 2	Absolute Value Alarm
Hi-hi alarm from block AA3	I Hi-hi alarm 3	Absolute Value Alarm
Lo-lo alarm from block AA1	I Lo-lo alarm 1	Absolute Value Alarm
Lo-lo alarm from block AA2	I Lo-lo alarm 2	Absolute Value Alarm
Lo-lo alarm from block AA3	I Lo-lo alarm 3	Absolute Value Alarm
Hi alarm from block AA1	I Hi alarm 1	Absolute Value Alarm
Hi alarm from block AA2	I Hi alarm 2	Absolute Value Alarm
Hi alarm from block AA3	I Hi alarm 3	Absolute Value Alarm
Lo alarm from block AA1	I Lo alarm 1	Absolute Value Alarm
Lo alarm from block AA2	I Lo alarm 2	Absolute Value Alarm
Lo alarm from block AA3	I Lo alarm 3	Absolute Value Alarm
Neg dev alarm from block DA1	I Neg dev 1	Deviation Alarm
Neg dev alarm from block DA2	I Neg dev 2	Deviation Alarm
Neg dev alarm from block DA3	I Neg dev 3	Deviation Alarm
Pos dev alarm from block DA1	I Pos dev 1	Deviation Alarm
Pos dev alarm from block DA2	I Pos dev 2	Deviation Alarm
Pos dev alarm from block DA3	I Pos dev 3	Deviation Alarm
Tuning Phase 8	I Tuning Phase 8	Self explanatory
Tuning Phase 7	I Tuning Phase 7	Self explanatory
Tuning Phase 6	I Tuning Phase 6	Self explanatory

Tuning Phase 5	I Tuning Phase 5	Self explanatory
Tuning Phase 4	I Tuning Phase 4	Self explanatory
Tuning Phase 3	I Tuning Phase 3	Self explanatory
Tuning Phase 2	I Tuning Phase 2	Self explanatory
Tuning Phase 1	I Tuning Phase 1	Self explanatory
Tuning Init	I Tuning Init	Self explanatory

Table 2 Information Messages.

Appendix A

Configuration Tree Diagrams

The following pages show the Configuration menu tree structures for each of the controllers: ECA06, ECA60, ECA600 and EMA60.

The different tabulation levels indicate the different levels in the configuration tree. This means that all items on the same tabulation level are on the same tree level.

Items preceded by a bullet (• item) are parameters.

Items preceded by a right double-arrow (\Rightarrow item) are inputs.

A term in square brackets, e.g. FA1[AddSub Value], means the value of the output from the block FA1 which is of the type AddSub.



Note: Outputs are not available to the user for configuration. When making a connection between an input and an output, the input is configured to point to an output.

A.1 ECA06 Configuration Tree

Inputs

Analog Inputs

A11–A12

- Mode
- RootExtraction
- FilterTime
- Alarm
- Identity

Analog In Com

AIC1–AIC4

- Mode
- Identity

Digital Inputs

D11

- Mode
- Identity

Digital In Com

DIC1–DIC4

- Mode
- Identity

Input Functions

Arith Func Blocks

FA01–02

AddSub

- Identity
- ⇒ Signal1
- ⇒ Signal2
- Factor1
- Factor2
- Bias

Set Point

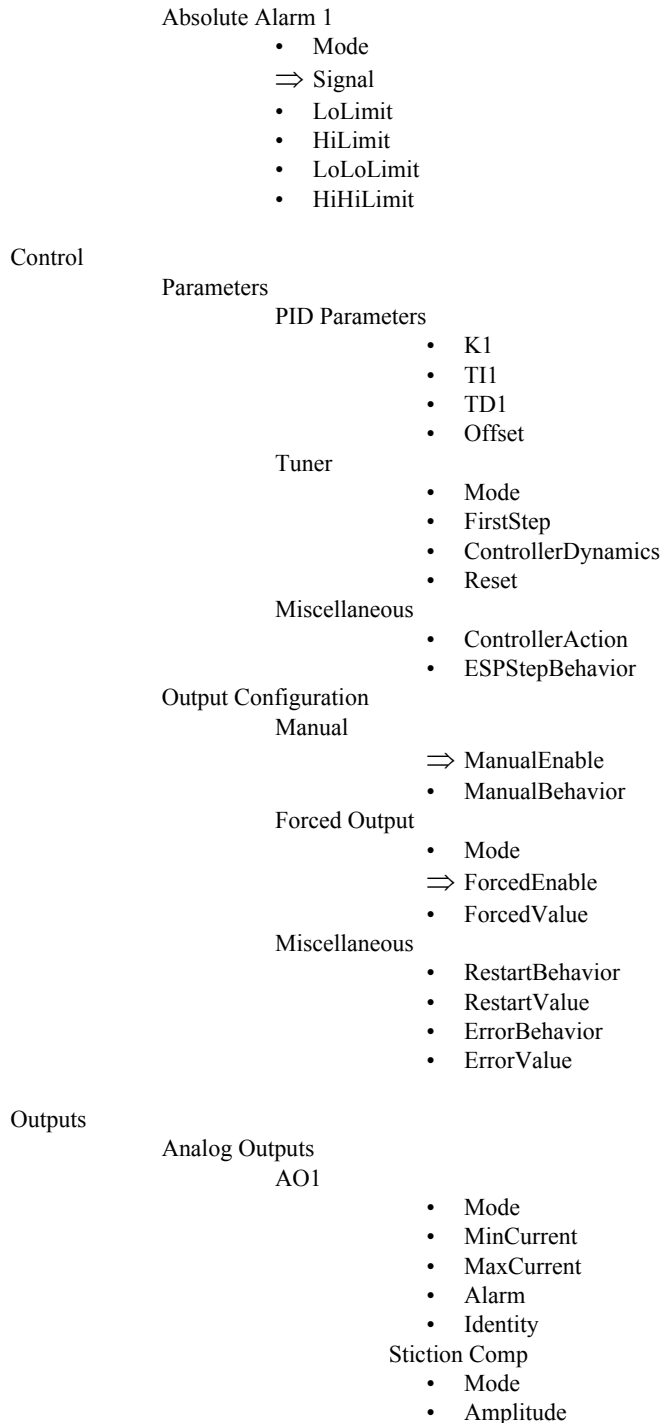
Process Value

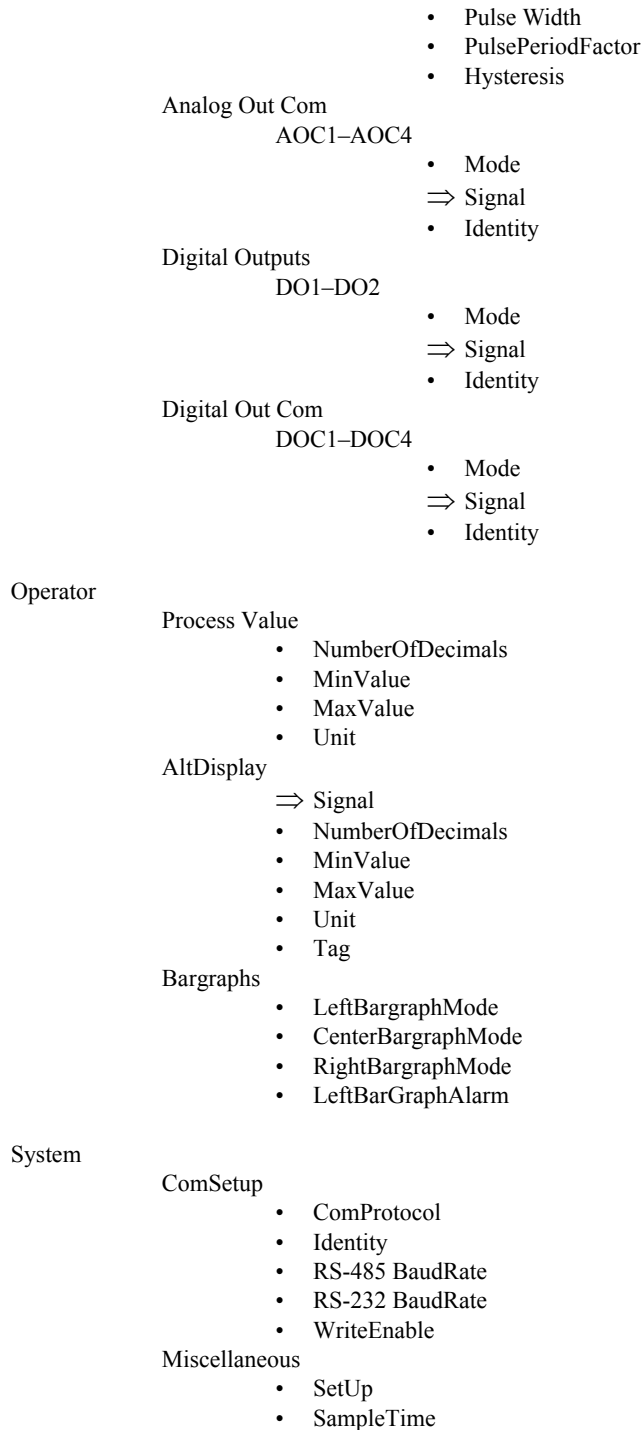
⇒ PVSignal

Set Point

- ⇒ ESPSignal
- ⇒ ESPEnable
- ESPLock
- SPLimitMode
- SPLoLimit
- SPHiLimit
- SPTrackMode

Alarms





- DisplayLanguage
 - Select
 - Edit
- Password

A.2 ECA60 Configuration Tree

Inputs

Analog Inputs

AI1–AI3

- Mode
- RootExtraction
- FilterTime
- Alarm
- Identity

Analog In Com

AIC1–AIC8

- Mode
- Identity

Analog User Blocks

AU1–AU2

- Mode
- NumberOfDecimals
- LoLimit
- HiLimit
- MinValue
- MaxValue
- Identity

Digital Inputs

DI1–DI3

- Mode
- Identity

Digital In Com

DIC1–DIC8

- Mode
- Identity

Digital User Blocks

DU1–DU2

- Mode
- Identity

Input Functions

Arith Func Blocks

FA01–FA04

- Type

AddSub

- Identity
- ⇒ Signal1
- ⇒ Signal2
- ⇒ Signal3
- Factor1
- Factor2

- Factor3
- Bias
- MulDiv
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Factor
 - Bias
- Logic Func Blocks
 - FL01–FL02
 - Type
 - And
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Or
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Xor
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - Not
 - Identity
 - ⇒ Signal
- Other Func Blocks
 - FO01–FO02
 - Type
 - Min
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Max
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2

- ⇒ Signal3
- ⇒ Signal4
- Select
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - ⇒ LSBSignal
 - ⇒ MSBSignal
- IncDec
 - Identity
 - ⇒ IncSignal
 - ⇒ DecSignal
 - Time
 - ⇒ TrackSignal
 - ⇒ TrackEnable
 - LoLimit
 - HiLimit
- Linear
 - Identity
 - ⇒ Signal
 - In1
 - Out1
 - In2
 - Out2
 - In3
 - Out3
 - In4
 - Out4
 - In5
 - Out5
 - In6
 - Out6
 - In7
 - Out7
 - In8
 - Out8
 - In9
 - Out9
 - In10
 - Out10
- RateLimiter
 - Identity
 - Signal
 - Rate

Set Point

Process Value

⇒ PVSignal

Set Point

⇒ ESPSignal

⇒ ESPEnable

⇒ ESPLock

- SPLimitMode (Off, On)
- SPLoLimit
- SPHiLimit
- SPTrackMode

Set Point Ramp

- RampMode
- Hold
- TimeUnit
- RampStart
- StartValue

Simple Ramp

- RampSpeed
- TimerTime

Alarms

Absolute Alarm 1

- Mode
- ⇒ Signal
- LoLimit
 - HiLimit
 - LoLoLimit
 - HiHiLimit

Deviation Alarm 1

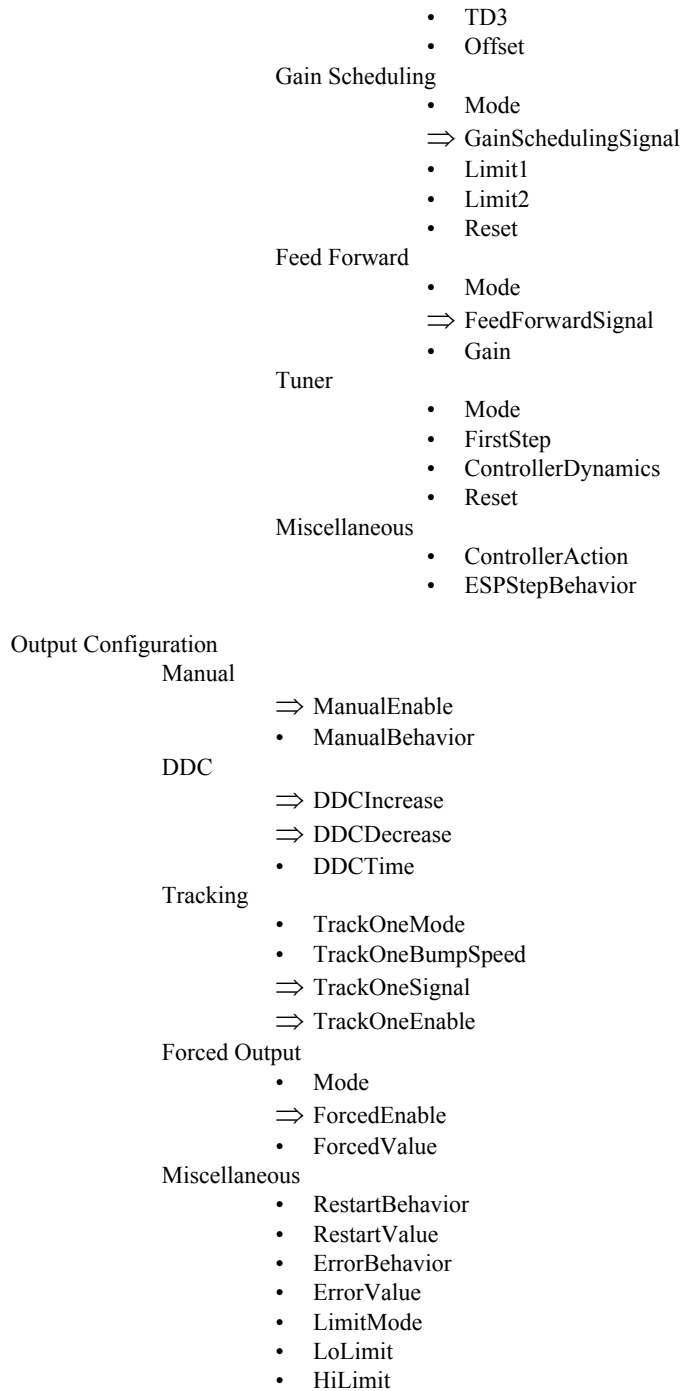
- Mode
- ⇒ Signal1⇒ Signal2
- NegDeviationLimit
 - PosDeviationLimit

Control

Parameters

PID Parameters

- K1
- T11
- TD1
- K2
- T12
- TD2
- K3
- T13



Outputs Functions

- Type
 - Analog
 - ThreeStatePulse
- DeadBand
- Hysteresis
- ActuatorPosTime
- ActuatorMinPulseLen
- ⇒ FeedbackSignal
- TwoStatePulse
- CycleTime

Outputs

Analog Outputs

AO1–AO2

- Mode
- ⇒ Signal
- MinCurrent
- MaxCurrent
- FixedCurrent
- Alarm
- Identity

Stiction Comp

- Mode
- Amplitude
- Pulse Width
- PulsePeriodFactor
- Hysteresis

Analog Out Com

AOC1–AOC8

- Mode
- ⇒ Signal
- Identity

Digital Outputs

DO1–DO4

- Mode
- ⇒ Signal
- Identity

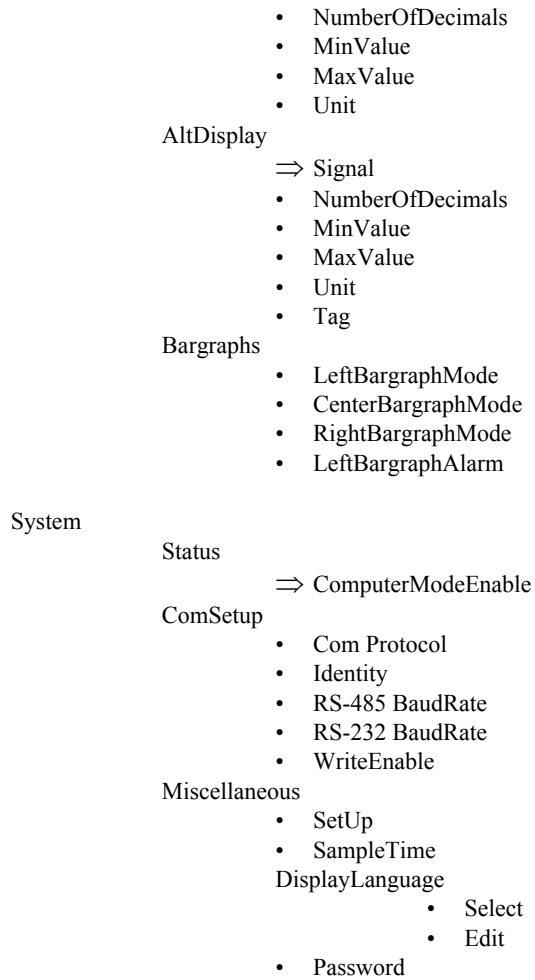
Digital Out Com

DOC1–DOC8

- Mode
- ⇒ Signal
- Identity

Operator

Process Value



A.3 ECA600 Configuration Tree

Inputs

Analog Inputs

AI1–AI5

- Mode
- RootExtraction
- FilterTime
- Alarm
- Loop
- Identity

Analog In Com

AIC1–AIC12

- Mode
- Identity

Analog User Blocks

AU1–AU8

- Mode
- NumberOfDecimals
- LoLimit
- HiLimit
- MinValue
- MaxValue
- Identity

Digital Inputs

DI1–DI4

- Mode
- Identity

Digital In Com

DIC1–DIC12

- Mode
- Identity

Digital User Blocks

DU1–DU8

- Mode
- Identity

Input Functions

Arith Func Blocks

FA01–FA16

- Type

AddSub

- Identity
- ⇒ Signal1
- ⇒ Signal2
- ⇒ Signal3
- Factor1

- Factor2
- Factor3
- Bias

MulDiv

- Identity
- ⇒ Signal1
- ⇒ Signal2
- ⇒ Signal3
- ⇒ Signal4
- Factor
- Bias

Exp

- Identity
- ⇒ Signal
- Factor
- Power
- Bias

Comp

- Identity
- ⇒ P_{diff}
- ⇒ P_{abs}
- ⇒ T
- Factor
- P_{absMax}
- P_{absMin}
- $P_{orifice}$
- $T_{orifice}$
- T_{max}
- T_{min}

Logic Func Blocks

FL01–FL16

- Type
- And
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
- Or
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3

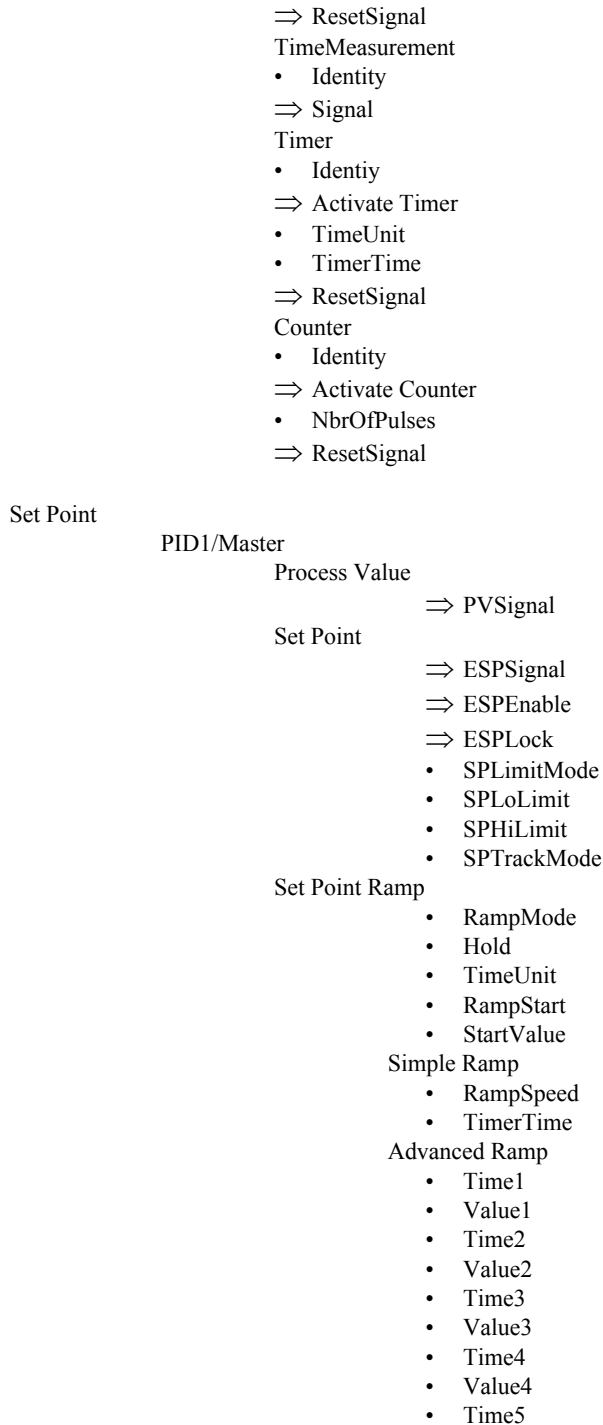
⇒ Signal4
 Xor
 • Identity
 • Signal1
 • Signal2
 S/R
 • Identity
 ⇒ Set
 ⇒ Reset
 Not
 • Identity
 ⇒ Signal

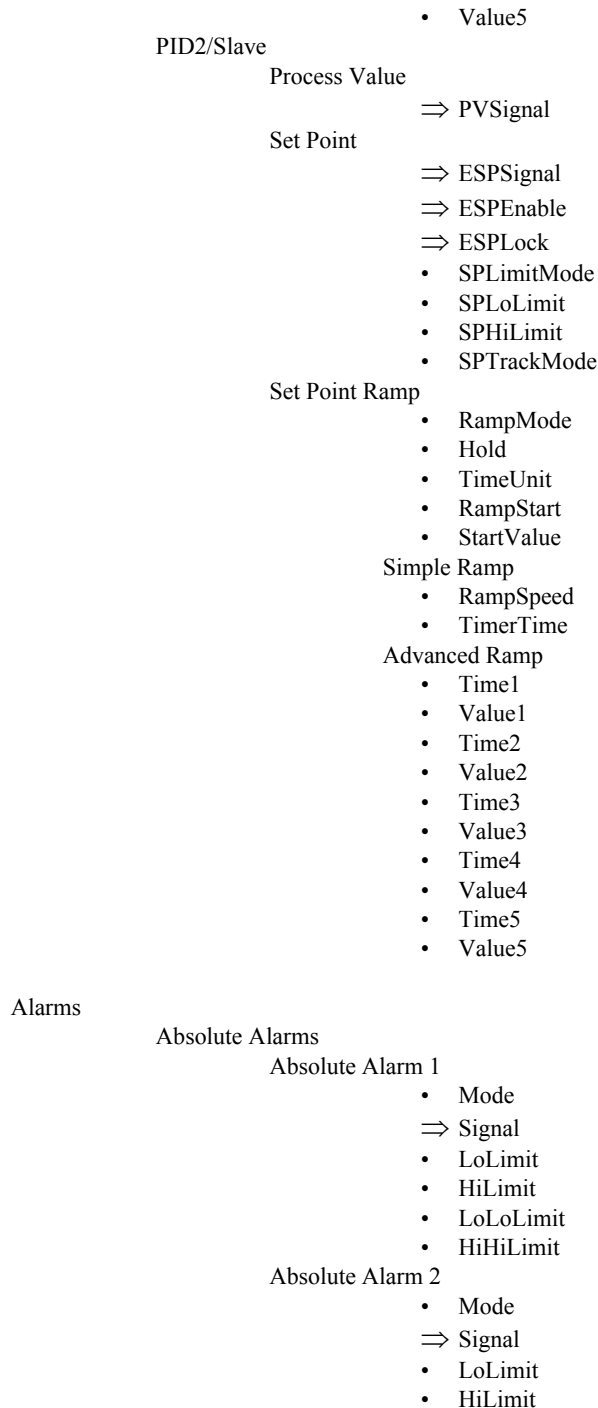
Other Func Blocks

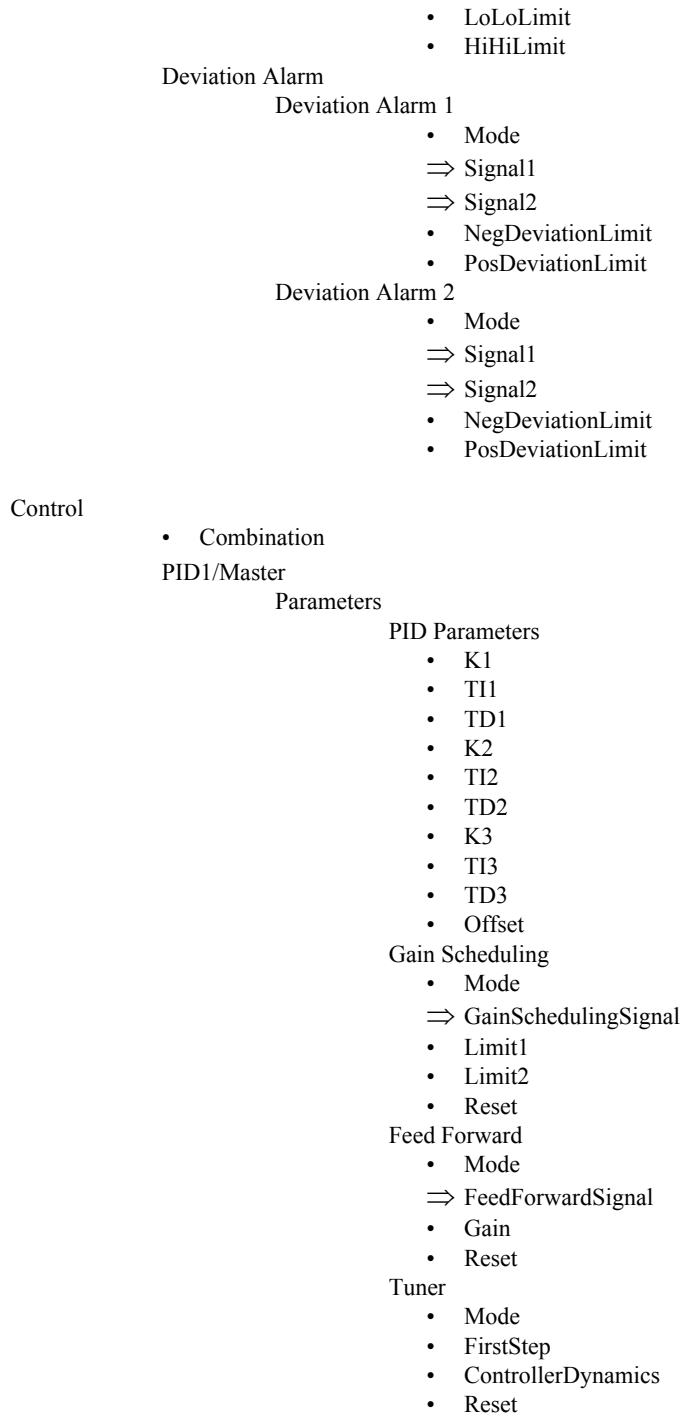
FO01–FO16

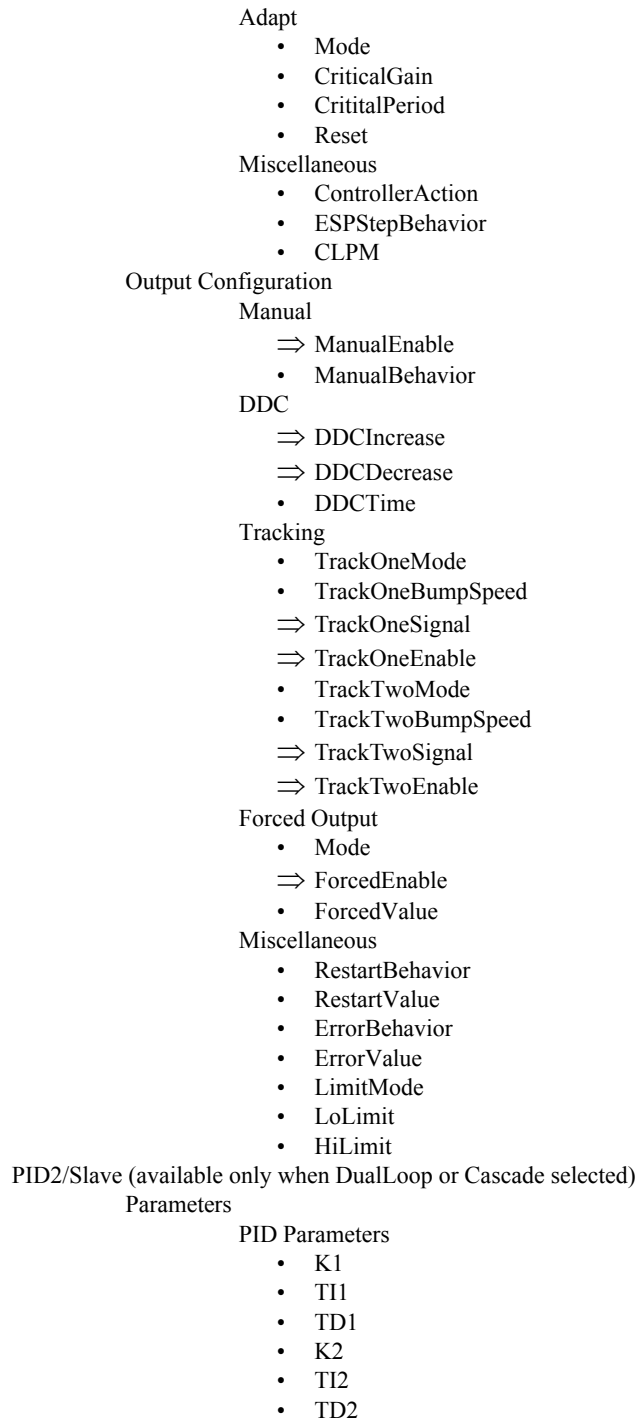
• Type
 Min
 • Identity
 ⇒ Signal1
 ⇒ Signal2
 ⇒ Signal3
 ⇒ Signal4
 Max
 • Identity
 ⇒ Signal1
 ⇒ Signal2
 ⇒ Signal3
 ⇒ Signal4
 Select
 • Identity
 ⇒ Signal1
 ⇒ Signal2
 ⇒ Signal3
 ⇒ Signal4
 ⇒ LSBSignal
 ⇒ MSBSignal
 Delay
 • Identity
 ⇒ Signal
 • DelayTime
 IncDec
 • Identity
 ⇒ IncSignal
 ⇒ DecSignal
 • Time

- ⇒ TrackSignal
- ⇒ TrackEnable
 - LoLimit
 - HiLimit
- LevelDetector
 - Identity
- ⇒ Signal
- ⇒ LoLimitSignal
- ⇒ HiLimitSignal
 - LoLimit Value
 - HiLimitValue
- Linear
 - Identity
- ⇒ Signal
 - In1
 - Out1
 - In2
 - Out2
 - In3
 - Out3
 - In4
 - Out4
 - In5
 - Out5
 - In6
 - Out6
 - In7
 - Out7
 - In8
 - Out8
 - In9
 - Out9
 - In10
 - Out10
- RateLimiter
 - Identity
- ⇒ Signal
 - Rate
- Integrator
 - Identity
- ⇒ Signal
- ⇒ Hold
 - TimeUnit
 - Time
 - OverflowLimit
 - DeadBand
 - PositiveOnly









- K3
- TI3
- TD3
- Offset
- Gain Scheduling
 - Mode
 - ⇒ GainSchedulingSignal
 - Limit1
 - Limit2
 - Reset
- Feed Forward
 - Mode
 - ⇒ FeedForwardSignal
 - Gain
 - Reset
- Tuner
 - Mode
 - FirstStep
 - ControllerDynamics
 - Reset
- Adapt
 - Mode
 - CriticalGain
 - CrititalPeriod
 - Reset
- Miscellaneous
 - ControllerAction
 - ESPStepBehavior
 - CLPM
- Output Configuration
 - Manual
 - ⇒ ManualEnable
 - ManualBehavior
 - DDC
 - ⇒ DDCIncrease
 - ⇒ DDCDecrease
 - DDCTime
 - Tracking
 - TrackOneMode
 - TrackOneBumpSpeed
 - ⇒ TrackOneSignal
 - ⇒ TrackOneEnable
 - TrackTwoMode
 - TrackTwoBumpSpeed
 - ⇒ TrackTwoSignal
 - ⇒ TrackTwoEnable
 - Forced Output

- Mode
- ⇒ ForcedEnable
- ForcedValue

Miscellaneous

- RestartBehavior
- RestartValue
- ErrorBehavior
- ErrorValue
- LimitMode
- LoLimit
- HiLimit

Output Functions

PID1/Master

- Type
- Analog
- ThreeStatePulse
- DeadBand
- Hysteresis
- ActuatorPosTime
- ActuatorMinPulseLen
- ⇒ FeedbackSignal
- TwoStatePulse
- CycleTime
- SplitRange
- MinIn1
- MaxIn1
- MinOut1
- MaxOut1
- MinIn2
- MaxIn2
- MinOut2
- MaxOut2

PID2/Slave

- Type
- Analog
- ThreeStatePulse
- DeadBand
- Hysteresis
- ActuatorPosTime
- ActuatorMinPulseLen
- ⇒ FeedbackSignal
- TwoStatePulse
- CycleTime
- SplitRange
- MinIn1
- MaxIn1
- MinOut1

- MaxOut1
- MinIn2
- MaxIn2
- MinOut2
- MaxOut2

Outputs

Analog Outputs

AO1–AO3

- Mode
- ⇒ Signal
- MinCurrent
- MaxCurrent
- FixedCurrent
- Alarm
- Loop
- Identity

Stiction Comp

- Mode
- Amplitude
- Pulse Width
- PulsePeriodFactor
- Hysteresis

Analog Out Com

AOC1–AOC12

- Mode
- ⇒ Signal
- Identity

Digital Outputs

DO1–DO6

- Mode
- ⇒ Signal
- Identity

Digital Out Com

DOC1–DOC12

- Mode
- ⇒ Signal
- Identity

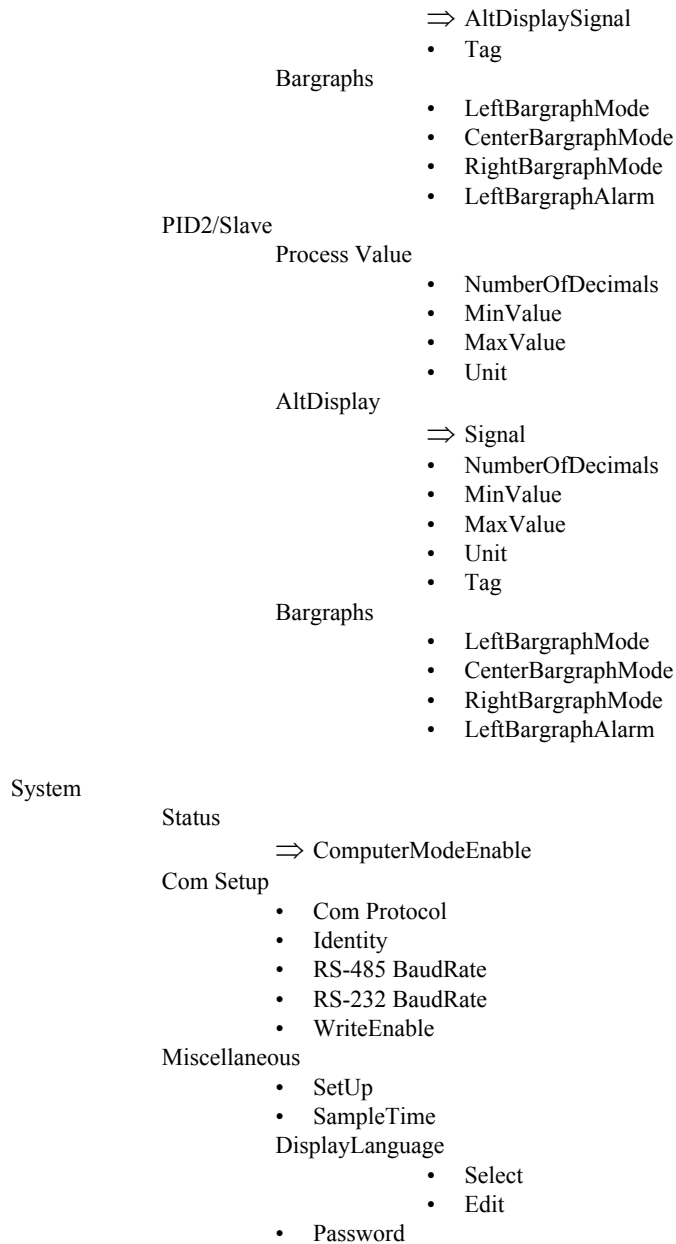
Operator

PID1/Master

Process Value

- NumberOfDecimals
- MinValue
- MaxValue
- Unit

Display



A.4 EMA60 Configuration Tree

Inputs

Analog Inputs

AI1–AI3

- Mode
- RootExtraction
- FilterTime
- Alarm
- Identity

Analog In Com

AIC1–AIC4

- Mode
- Identity

Analog User Blocks

AU1–AU4

- Mode
- NumberOfDecimals
- LoLimit
- HiLimit
- MinValue
- MaxValue
- Identity

Digital Inputs

DI1–DI2

- Mode
- Identity

Digital In Com

DIC1–DIC4

- Mode
- Identity

Digital User Blocks

DU1–DU4

- Mode
- Identity

Input Functions

Arith Func Blocks

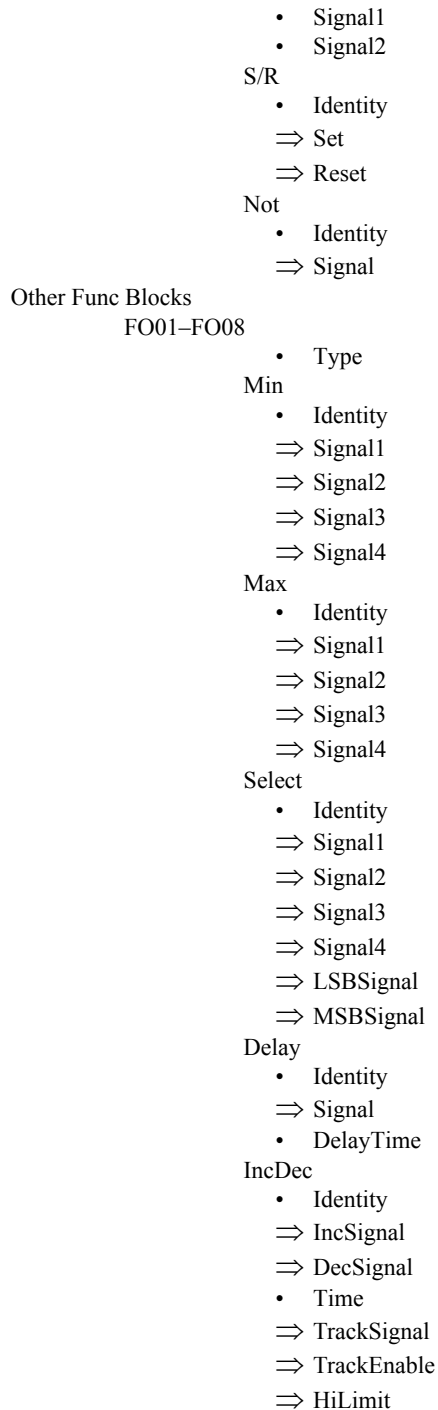
FA01–FA8

- Type

AddSub

- Identity
- ⇒ Signal1
- ⇒ Signal2
- ⇒ Signal3
- Factor1
- Factor2

- Factor3
- Bias
- MulDiv
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Factor
 - Bias
- Exp
 - Identity
 - ⇒ Signal
 - Factor
 - Power
 - Bias
- Comp
 - Identity
 - ⇒ Pdiff
 - ⇒ Pabs
 - ⇒ T
 - Factor
 - PabsMax
 - PabsMin
 - Porifice
 - Torifice
 - Tmax
 - Tmin
- Logic Func Blocks
 - FL01–FL08
 - Type
 - And
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Or
 - Identity
 - ⇒ Signal1
 - ⇒ Signal2
 - ⇒ Signal3
 - ⇒ Signal4
 - Xor
 - Identity



LevelDetector

- Identity

⇒ Signal

⇒ LoLimitSignal

⇒ HiLimitSignal

- LoLimit Value
- HiLimitValue

Linear

- Identity

⇒ Signal

- In1
- Out1
- In2
- Out2
- In3
- Out3
- In4
- Out4
- In5
- Out5
- In6
- Out6
- In7
- Out7
- In8
- Out8
- In9
- Out9
- In10
- Out10

RateLimiter

- Identity

⇒ Signal

- Rate

Integrator

- Identity

⇒ Signal

⇒ Hold

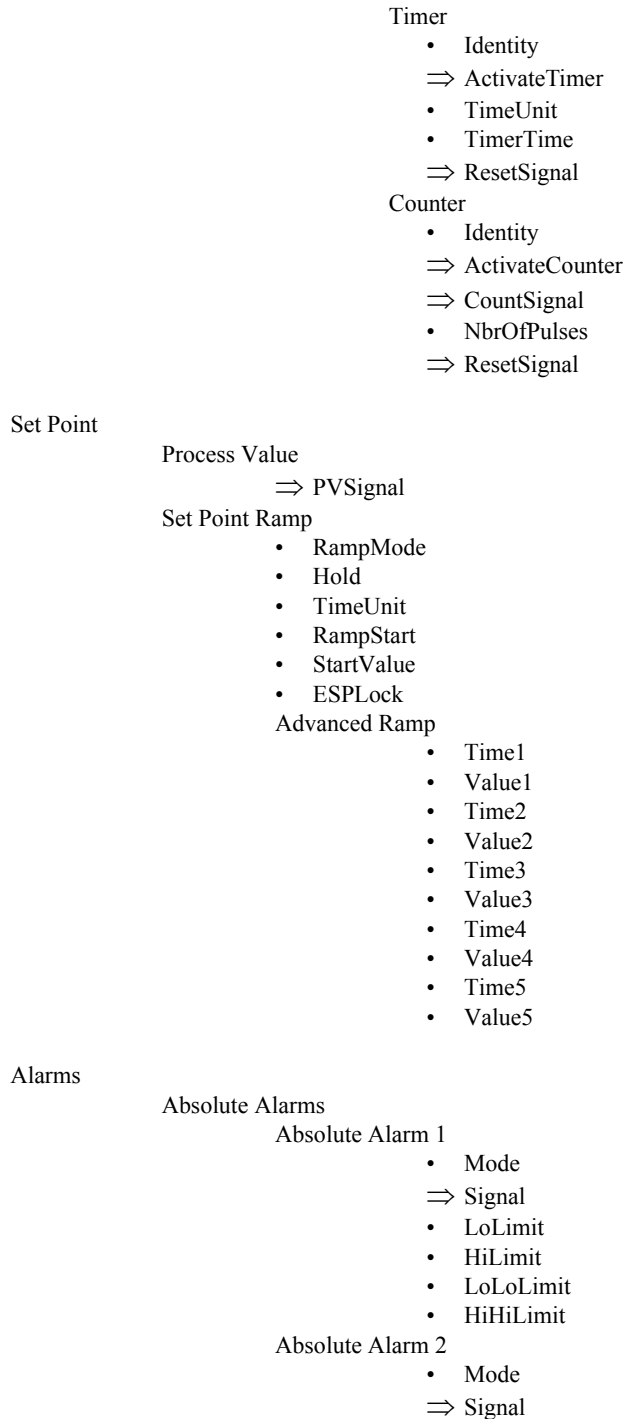
- TimeUnit
- Time
- OverflowLimit
- DeadBand
- PositiveOnly

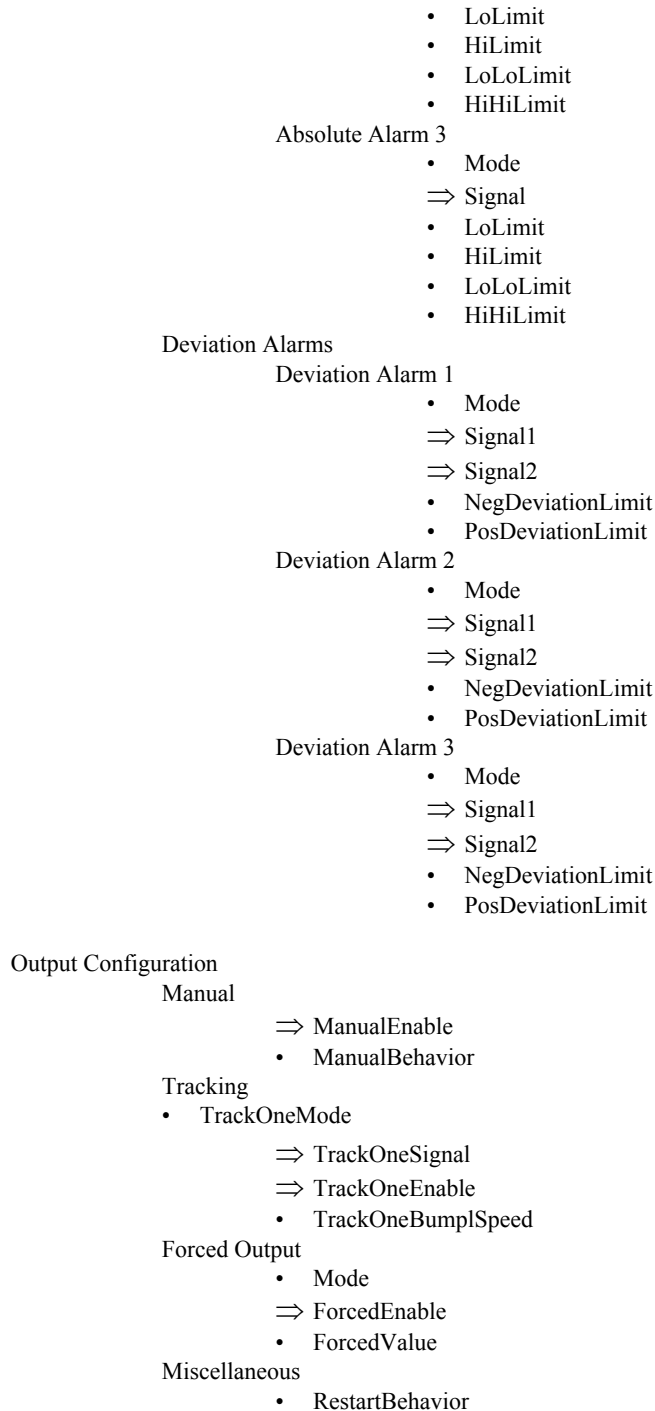
⇒ ResetSignal

TimeMeasurement

- Identity

⇒ Signal





- RestartValue
- ErrorBehavior
- ErrorValue
- LimitMode
- LoLimit
- HiLimit

Outputs

Analog Outputs
AO1

- Mode
- ⇒ Signal
- MinCurrent
- MaxCurrent
- Alarm
- Identity

Stiction Comp

- Mode
- Amplitude
- Pulse Width
- PulsePeriodFactor
- Hysteresis

Analog Out Com
AOC1–AOC4

- Mode
- ⇒ Signal
- Identity

Digital Outputs
DO1–DO6

- Mode
- ⇒ Signal
- Identity

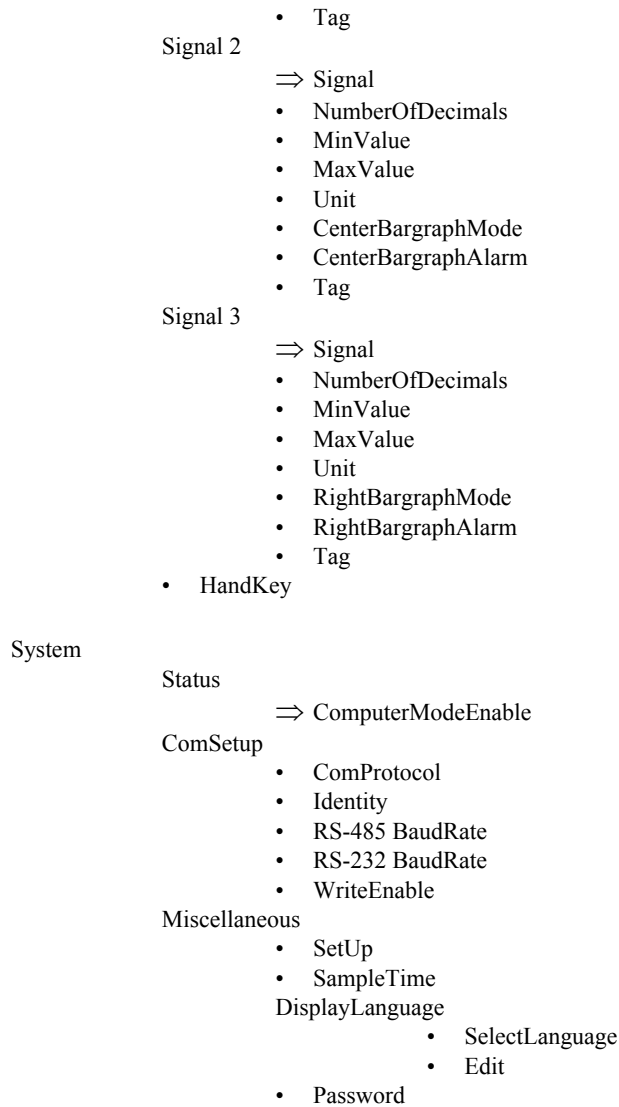
Digital Out Com
DOC1–DOC4

- Mode
- ⇒ Signal
- Identity

Operator

Signal 1

- ⇒ Signal
- NumberOfDecimals
- MinValue
- MaxValue
- Unit
- LeftBargraphMode
- LeftBargraphAlarm



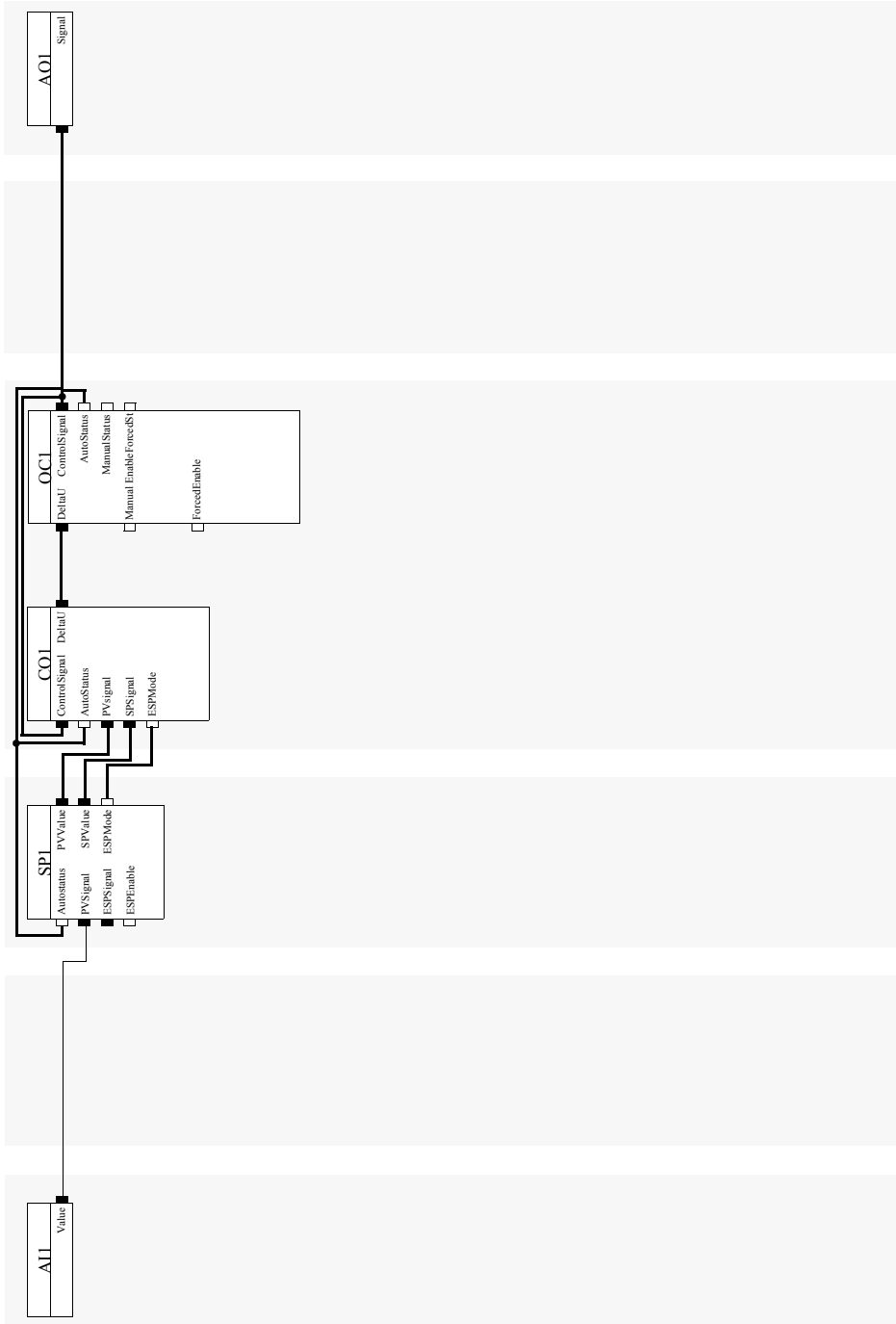
Appendix A

Configuration Work Sheets

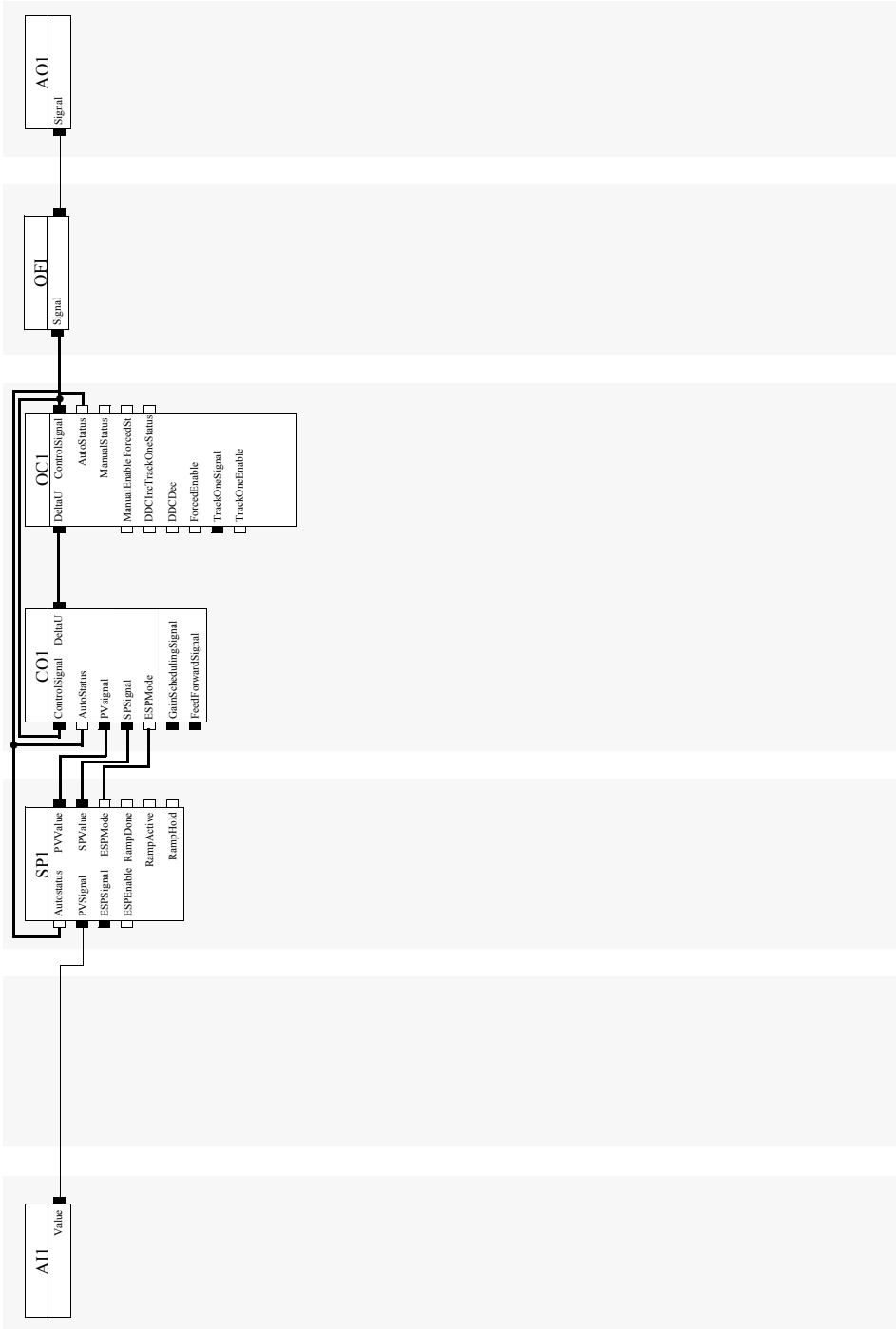
This appendix contains a Configuration Work Sheet for each controller to assist you in configuration planning.

Make a photocopy of one of the default sheets and add the configuration required for the control application.

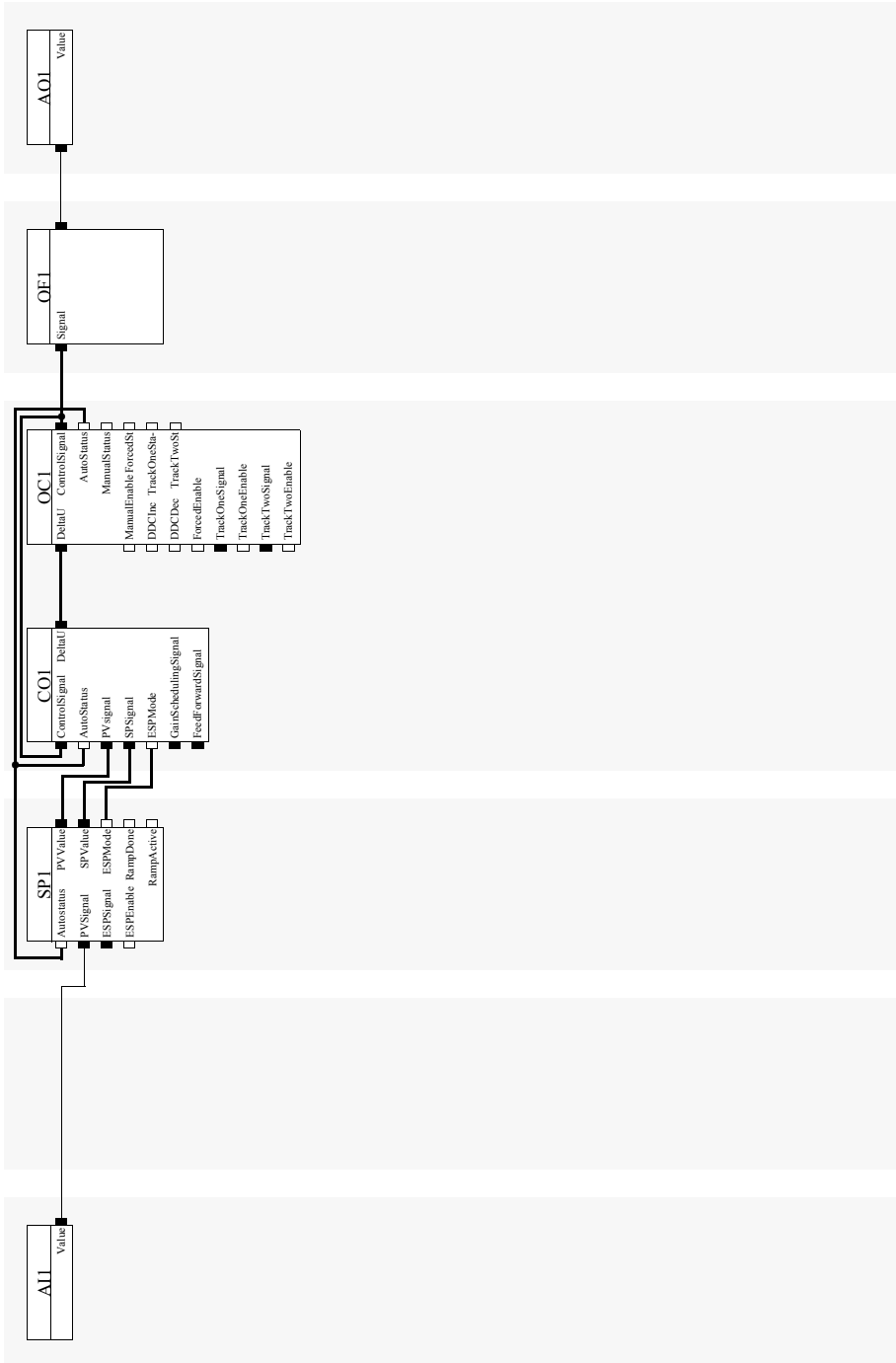
ECA06, Default



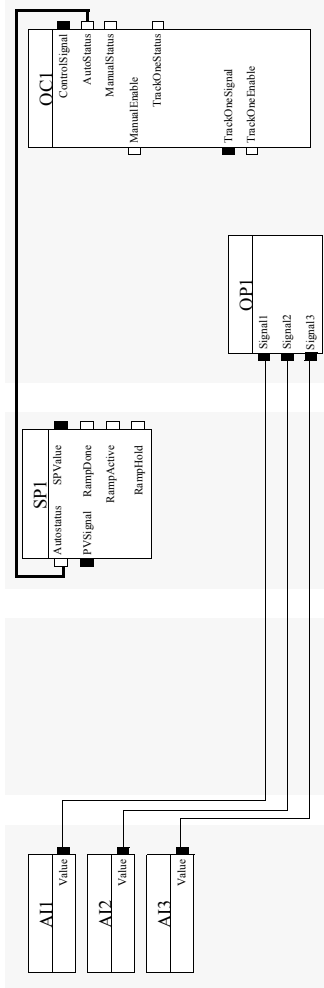
ECA60, Default



ECA600, Default



EMA60 Default/Triple Ind



Appendix A

Communication

This appendix describes how to establish ECA communication with a supervisory control system.

A.1 Communication setup

The information below facilitates the communication setup.

- Serial communication interface RS485 (point to point or multidrop) or RS232 (point to point) is available. A screened twisted-pair cable is to be used. See "Communications signal wiring, RS485", on page xxxiv.
- The available protocols are COMLI and MODBUS RTU, see "COM Setup", on page cxc.
- The parity must be set to Odd (fixed) in the supervisory system.
- The baud rate may be set between 1200 and 38400 baud, see "COM Setup", on page cxc.
- All ECA controllers connected to the supervisory control system (Master) are Slaves.
- Each ECA controller in a network must be set to a unique communication identity number between #1 and #247, see "COM Setup", on page cxc.

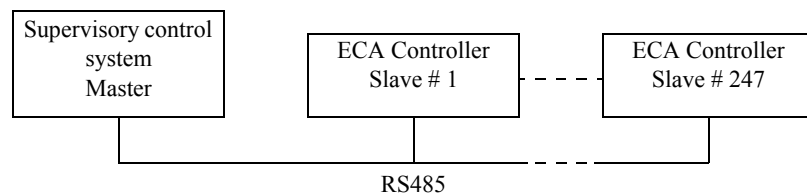


Figure 1 ECA controllers connected in multidrop configuration to a supervisory system.

Recommended max number of units per master system is 10 slaves. This is due to the response time when cyclically requesting information from all slaves.

A.2 Protocol Description

Modbus Communication

Function Codes

ECA can only work as a MODBUS slave and supports the following MODBUS function codes.

Code	Designation	Function
01	Read Coil Status	Read binary values
02	Read Input Status	Read binary values
03	Read Holding Registers	Read registers (16 bits Integer)
04	Read Input Registers	Read registers (16 bits Integer)
05	Force Single Coil	Write/Set a single register value (1 bit)
06	Preset Single Register	Write/Set a single register value (16 bits Integer)

Message codes [01] and [02] are treated as a read operation in the "I/O RAM Map" in the ECA.

Message codes [03] and [04] are treated as a read operation in the "Register Map" in the ECA.

Message code [05] is a write operation in the "I/O RAM Map" (write single bit).

Message code [06] is a write operation in the "Register Map" (write single register).

Response Codes

ECA supports the following response codes:

Code	Name
01	Illegal function

Code	Name
02	Illegal data address
03	Illegal data value

Examples of MODBUS Master Requests

Here are some examples of read/write operations in a MODBUS Master system. The syntax is an example and differs between different master systems.

Operation	Syntax in Master	Operation in ECA
Read Coil Status	C3	Read I/O RAM Map bit 3
Read Input Status	I1076 or I10076	Read I/O RAM Map bit 76
Read Holding Registers	HR4006, HR40006	Read Register Map R6 (AIC6)
Read Input Registers	IR3033, IR30033	Read Register Map R33 (AI1)
Force Single Coil	C208	Write I/O RAM Map bit 208
Preset Single Register	IR3052, HR4052, IR30052, HR40052	Write Register Map R52 (ISP1)

Comli Communication

Comli is a general asynchronous, serial communication protocol, utilizing the master/slave relationship and communication in only one direction at the time (Half-duplex).

ECA can only work as a Comli slave and supports the following message types.

Type 0 – transfer of I/O RAM Status or Register value.

Type 1 – acknowledge correctly received data.

Type 2 – request of I/O RAM Status or Register value

Type 3 – transfer of status for one I/O bit.

Type 4 – request of status for one I/O bit.

Message Structure

There are five message types recognised by the controller. Each message type sent has a given number of characters, or more precisely, a given number of bytes of

information. The basic format always starts with the STX (start of text) as shown, followed by bytes for:

- The destination address
- A stamp (indicating if the message is a retransmission)
- The message type
- The message
- End of text (ETX) character
- A checksum block check count (BCC)

0 STX	1 2 Destination	3 Stamp	4 Message type	5 6 7 8 Start address	9 10 Number of data bytes	11 N-2 Data ASCII/binary	N-1 ETX	N BCC
----------	--------------------	------------	----------------------	--------------------------	---------------------------------	--------------------------------	------------	----------

Explanation of Terms

Terms used in the explanation of the message types and the ASCII characters used are as follows.

STX – Start of text. This is an ASCII character with the value 02H (the symbol H is used to indicate hexadecimal value) and indicates the start of a message.

Destination Address – This address indicates which controller the message is intended for. The address consists of two characters that have values between 00 and FF. Since the master identity is always 00, the slave station identities begin with 01.

Example: To address station 24:

Decimal 24 equivalent is 18H. The ASCII code for 18 is:

1=31H

8=38H

The destination address would appear as:

Message byte number	
1	2
24=18H=	31
	38

Stamp – A character that changes value between message so that the slave station can determine if it is a retransmission. The first message after starting must have the value 30H. After the initial message, STAMP must alternate between 31H and 32H.

Stamp from the controller is always assigned the same value as Stamp from the master (STAMPM) in the correctly received message. This means that Stamp and Stamps are always different for each new message.

If at any time they are the same, the controller knows that the message type is a retransmission and does not process the data.

Message type – Message position 4 reflects the message type being sent.

Start Address – Four characters indicating the starting address of the memory location, or registers in the controller where data retrieval is to begin. Each controller register holds two bytes (16 bits). Therefore, register addresses must be evenly divisible by 16.

Formula: Address = (16 × register number)H + 4000H (register 0).

Example: Calculate the address for register 16.

$$16 \times 16 = 256(\text{Decimal}) = 100$$

$$100\text{H} + 4000\text{H} = 4100\text{H}$$

$$4100\text{H} = \text{ASCII } 34\text{H}, 31\text{H}, 30\text{H}, 30\text{H}$$

5	6	7	8
34	31	30	30

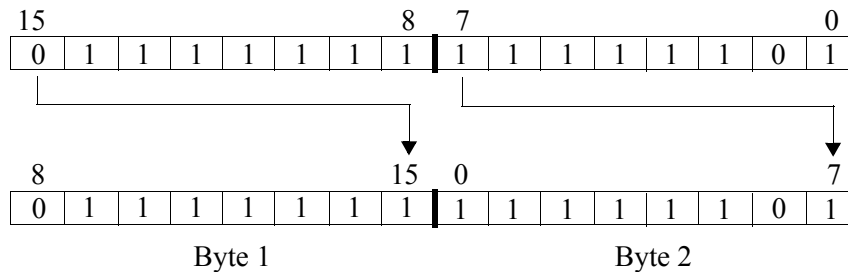
Number of Characters – This part of the message is a count of the number of data bytes or characters sent in the message. The example below shows the ASCII code for 32 bytes.

$$24=18H= \begin{array}{|c|c|} \hline & \begin{array}{c} 9 \quad 10 \\ \hline 32 \quad 30 \end{array} \\ \hline \end{array}$$

Data Characters – This section of the message contains the transmitted characters specified by the Number of Characters. The value in the 16 bit registers are sent in two bytes. Byte 1 which is sent second is the data character with the lowest number. The bits in the register are sent in reverse order. The most significant bit in the register is sent as the least significant bit of byte 2, as shown in the example. The least significant bit in the register is the most significant bit of byte 1.

Example: The value loaded in register A is:

7FFD (50%)



ETX – End of text (03H), indicates the end of message.

BCC – block check count. This is a checksum calculated as the module 2 addition (XOR) of each character in the message, except the initial STX-character.

A.3 Communication Facilities

A supervisory control system can read and write bits or register words located in the communication memory area of the ECA controller (see "Communication Area Mapping", on page cclxxxv). The communication facilities are listed below.

- All available analog and digital signal values can be read by a supervisory system. The signals must be connected via communications buffers of type AOC (Analog Output Communication) and DOC (Digital Output Communication).

- Analog and digital signal values can be written to the controller. The signals must be connected via communications buffers of type AIC (Analog Input Communication) and DIC (Digital Input Communication).
- All status bits, such as Alarm, Tuning phase, Auto/Manual mode, etc., can be read by a supervisory system.
- All register values, such as PV, SP, ISP, ESP, Control Signals and PID parameters can be read by a supervisory system.
- ISP and the Control signals can be set in manual mode.
- New PID parameters can be written to the controller.
- The controller can be switched to Manual or Auto-Mode using the parameter *ManualEnable* in the Control block (the switch must be connected via a DIC block).
- The Control signal can be increased or decreased (in manual mode) by use of a bit operation.
- Tuning can be started/stopped individually for each loop.
- An alarm can be acknowledged from a supervisory system.



Note. Switching between Computer and Local mode is not possible. This must be performed from the front panel.

Example

The example below illustrates the interconnections for switching the controller setpoint from internal mode (ISP) to external mode (ESP) on command from the supervisory control system. The setpoint is then determined by the control system.

The external analog setpoint signal is connected to communications buffer AIC3, the select signal (LSBSignal) is connected to DIC2 and the ESPEnable signal to DIC4. The analog SP value is connected to AOC3 and the parameter *RampActive* is connected to DOC4, so that the control system can read these signals. The bits and registers used for the AIC, DIC, AOC and DOC blocks appear in the tables in "Communication Area Mapping", on page cclxxxv.

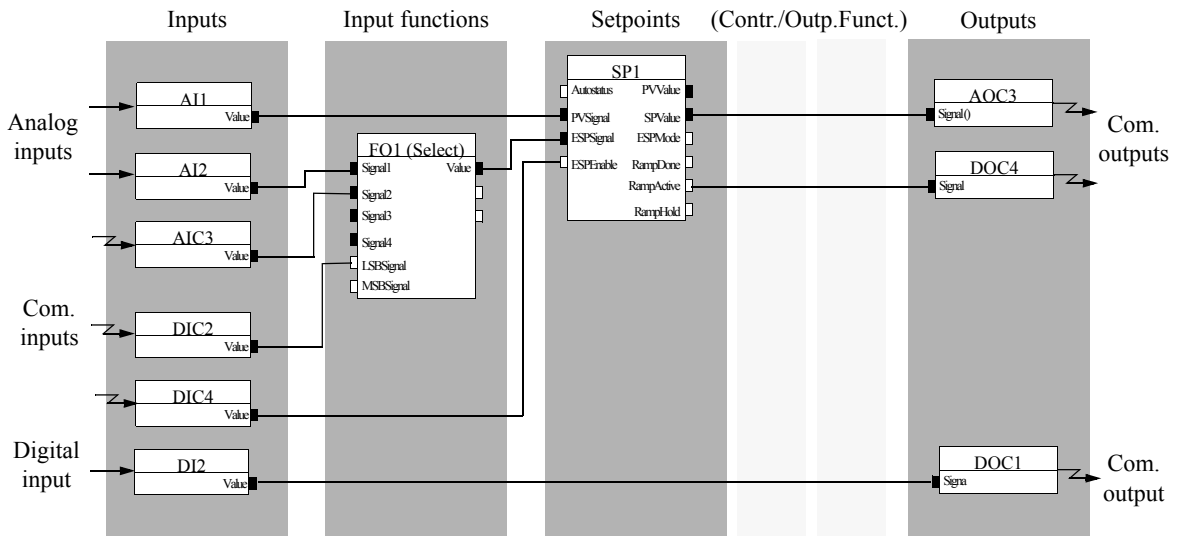


Figure 2 Communication example.

Changing Communication Mode

You can regain control over all front panel functions by pressing the Next key until the display shows "Communication Mode", and then selecting Local.

Changing Control Modes

You can toggle Control Modes between Auto and Manual by pressing the Hand key while in Computer Mode. The Communication Mode remains in Computer however.

Changing Loops

You can select loops by pressing the Next key until the display shows “Loop”, and then selecting the desired loop.

A.4 Front Panel Usage

If the Communication Mode is Local, the front panel is operable as usual.

If the Communication Mode is Computer, the operator can inspect all posts in the list of operator choices. But there are just four posts that can be operated:

- Communication Mode (in order to switch back to Local Mode)
- Control Mode (in order to switch between Automatic and Manual Mode)
- Output Signal (when the Control Mode is Manual)
- Displayed Loop (in order to switch between display of Loop1 and Loop2 when controlling multiple loops)

All other functions are disabled.

A.5 Communication Area Mapping

A supervisory system gains access to the controller’s parameters via I/O bits and registers. The description below are common both for Comli and Modbus.

I/O RAM Map

Variable	Access	Address			Bit Position							
		Hex	Oct	Dec	7	6	5	4	3	2	1	0
DICx	Read/Write	0000 ₁₆	0000 ₈	0000 ₁₀	DIC8	DIC7	DIC6	DIC5	DIC4	DIC3	DIC2	DIC1
DICx	Read/Write	0008 ₁₆	0010 ₈	0008 ₁₀					DIC12	DIC11	DIC10	DIC9
DOCx	Read	0010 ₁₆	0020 ₈	0016 ₁₀	DOC8	DOC7	DOC6	DOC5	DOC4	DOC3	DOC2	DOC1
DOCx	Read	0018 ₁₆	0030 ₈	0024 ₁₀					DOC12	DOC11	DOC10	DOC9
DIx	Read	0020 ₁₆	0040 ₈	0032 ₁₀					DI4	DI3	DI2	DI1
DUx	Read	0028 ₁₆	0050 ₈	0040 ₁₀	DU8	DU7	DU6	DU5	DU4	DU3	DU2	DU1
DOx	Read	0030 ₁₆	0060 ₈	0048 ₁₀			DO6	DO5	DO4	DO3	DO2	DO1
Control Mode	Read	0038 ₁₆	0070 ₈	0056 ₁₀				Tuning Loop1	Tracking Loop1	Forced Loop1	Manual Loop1	Auto Loop1
Control Mode	Read	0040 ₁₆	0100 ₈	0064 ₁₀				Tuning Loop2	Tracking Loop2	Forced Loop2	Manual Loop2	Auto Loop2
System Errors	Read	0048 ₁₆	0110 ₈	0072 ₁₀				Sys Err RAM	Sys Err FLASH Chks	Sys Err FLASH Erase	Sys Err FLASH Write	Sys Err HW
Errors	Read	0050 ₁₆	0120 ₈	0080 ₁₀							DO o.load	SampT Err

Variable	Access	Address			Bit Position							
		Hex	Oct	Dec	7	6	5	4	3	2	1	0
Errors	Read	0058 ₁₆	0130 ₈	0088 ₁₀	A15 Int Err	A14 Int Err	A13 Int Err	A12 Int Err	A11 Int Err	AO3 Int Err	AO2 Int Err	AO1 Int Err
Warnings	Read	0060 ₁₆	0140 ₈	0096 ₁₀	A15 Int Warn	A14 Int Warn	A13 Int Warn	A12 Int Warn	A11 Int Warn	AO3 Int Warn	AO2 Int Warn	AO1 Int Warn
Warnings	Read	0068 ₁₆	0150 ₈	0104 ₁₀	AA Src Off	DA Src Off	OC Src Off	OF Src Off	CLPM L2	CLPM L1	Arith Err	Calibr Err
Warnings	Read	0070 ₁₆	0160 ₈	0112 ₁₀	AU Src Off	DI Src Off	DIC Src Off	DU Src Off	FA Src Off	FL Src Off	FO Src Off	SP Src Off
Warnings	Read	0078 ₁₆	0170 ₈	0120 ₁₀	Adapt Warn L2	Adapt Warn L1	Tun Warn L2	Tun Warn L1	Tun Abort L2	Tun Abort L1	AI Src Off	A IC Src Off
Information	Read	0080 ₁₆	0200 ₈	0128 ₁₀	Hi 2	Hi 1	LoLo 3	LoLo 2	LoLo 1	HiHi 3	HiHi 2	HiHi 1
Information	Read	0088 ₁₆	0210 ₈	0136 ₁₀	Pos Dev 1	Neg Dev 3	Neg Dev 2	Neg Dev 1	Lo 3	Lo 2	Lo 1	Hi 3
Information	Read	0090 ₁₆	0220 ₈	0144 ₁₀					Tun Init L2	Tun Init L1	Pos Dev 3	Pos Dev 2
Information	Read	0098 ₁₆	0230 ₈	0152 ₁₀	Tun Phase 8 L1	Tun Phase 7 L1	Tun Phase 6 L1	Tun Phase 5 L1	Tun Phase 4 L1	Tun Phase 3 L1	Tun Phase 2 L1	Tun Phase 1 L1
Information	Read	00A0 ₁₆	0240 ₈	0160 ₁₀	Tun Phase 8 L2	Tun Phase 7 L2	Tun Phase 6 L2	Tun Phase 5 L2	Tun Phase 4 L2	Tun Phase 3 L2	Tun Phase 2 L2	Tun Phase 1 L2
Control	Read/Write	00B0 ₁₆	260 ₈	176 ₁₀							Controller Action, Loop 1	GS Mode, Loop 1
Control	Read/Write	00B8 ₁₆	270 ₈	186 ₁₀							Controller Action, Loop 2	GS Mode, Loop 2
Control	Read/Write	00C0 ₁₆	300 ₈	192 ₁₀								Tuner Strt/Abort Loop 1 ^a
Control	Read/Write	00C8 ₁₆	310 ₈	200 ₁₀								Tuner Strt/Abort Loop 2 ^a
Control	Read/Write	00D0 ₁₆	320 ₈	208 ₁₀				Decrease Loop 2 (see note 3)	Decrease Loop 1 (see note 3)	Increase Loop 2 (see note 3)	Increase Loop 1 (see note 3)	Alarm acknowl.

a. The value 1 (one) denotes start and the value 0 (zero) denotes abort

Table 1 I/O RAM Map.



Note 1. For ‘GS Mode’, the value 0 (zero) denotes Off, and the value 1 (one) denotes On.

Note 2. For ‘Controller Action’, the value 0 (zero) denotes Reversed, and the value 1 (one) denotes Direct. The value 0 (zero) is also used for Unknown (Read only).

Note 3. A supervisory system, e.g. SattGraph 5000, can conveniently increase or decrease the output value of the ECA controller by use of two bits in the variable Control 00D0₁₆. The analog output signal value (loop 1) will increase as long as bit 1 is active (bit 1 = 1 and bit 3 = 0) and will decrease if bit 3 is active (bit 3 = 1 and bit 1 = 0). The speed of increase/decrease depends on the controller sample time.

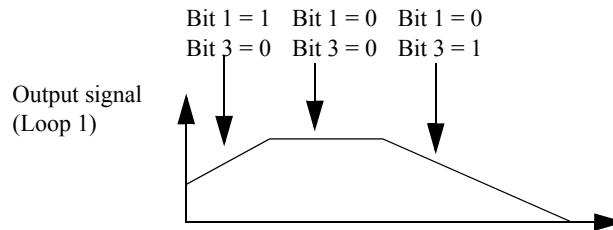


Figure 3 External increase/decrease of the analog output signal using two bits in the variable Control 00D016.

Register Map

Each register consists of a 16-bit integer containing a value between 0 and 65535.

Variable	Access	Register
AIC1	Read/Write	R1
AIC2	Read/Write	R2
AIC3	Read/Write	R3
AIC4	Read/Write	R4
AIC5	Read/Write	R5
AIC6	Read/Write	R6
AIC7	Read/Write	R7
AIC8	Read/Write	R8
AIC9	Read/Write	R9
AIC10	Read/Write	R10
AIC11	Read/Write	R11
AIC12	Read/Write	R12

Variable	Access	Register
AOC1	Read	R17
AOC2	Read	R18
AOC3	Read	R19
AOC4	Read	R20
AOC5	Read	R21
AOC6	Read	R22
AOC7	Read	R23
AOC8	Read	R24
AOC9	Read	R25
AOC10	Read	R26
AOC11	Read	R27
AOC12	Read	R28
Control Signal of OC1 block	Read/Write	R29 ^a
Control Signal of OC2 block	Read/Write	R30 ^b
AI1	Read	R33
AI2	Read	R34
AI3	Read	R35
AI4	Read	R36
AI5	Read	R37
AU1	Read	R38
AU2	Read	R39
AU3	Read	R40
AU4	Read	R41
AU5	Read	R42
AU6	Read	R43
AU7	Read	R44
AU8	Read	R45
AO1	Read	R46
AO2	Read	R47
AO3	Read	R48

Variable	Access	Register
Sample Time	Read	R49
Process Value of SP1block	Read	R50
Set Point of SP1block	Read	R51
Internal Set Point of SP1 block	Read/Write	R52
External Set Point of SP1 block	Read	R53
Control Signal of OC1 block	Read/Write	R54 ^a
Process Value of SP2 block	Read	R55
Set Point of SP2 block	Read	R56
Internal Set Point of SP2 block	Read/Write	R57
External Set Point of SP2 block	Read	R58
Control Signal of OC2 block	Read/Write	R59 ^b
Model	Read	R60
SW Version, part 1	Read	R61
SW Version, part 2	Read	R62
K1, Loop 1	Read/Write	R96
TI1, Loop 1 (hi ^c)	Read/Write	R97
TI1, Loop 1 (lo ^c)	Read/Write	R98
TD1, Loop 1 (hi ^c)	Read/Write	R99
TD1, Loop 1 (lo ^c)	Read/Write	R100
K2, Loop 1	Read/Write	R101
TI2, Loop 1 (hi ^c)	Read/Write	R102
TI2, Loop 1 (lo ^c)	Read/Write	R103
TD2, Loop 1 (hi ^c)	Read/Write	R104
TD2, Loop 1 (lo ^c)	Read/Write	R105
K3, Loop 1	Read/Write	R106
TI3, Loop 1 (hi ^c)	Read/Write	R107
TI3, Loop 1 (lo ^c)	Read/Write	R108
TD3, Loop 1 (hi ^c)	Read/Write	R109
TD3, Loop 1 (lo ^c)	Read/Write	R110
Offset, Loop 1	Read/Write	R111

Variable	Access	Register
Limit1, Loop 1	Read/Write	R112
Limit2, Loop 1	Read/Write	R113
K1, Loop 2	Read/Write	R128
TI1, Loop 2 (hi ^c)	Read/Write	R129
TI1, Loop 2 (lo ^c)	Read/Write	R130
TD1, Loop 2 (hi ^c)	Read/Write	R131
TD1, Loop 2 (lo ^c)	Read/Write	R132
K2, Loop 2	Read/Write	R133
TI2, Loop 2 (hi ^c)	Read/Write	R134
TI2, Loop 2 (lo ^c)	Read/Write	R135
TD2, Loop 2 (hi ^c)	Read/Write	R136
TD2, Loop 2 (lo ^c)	Read/Write	R137
K3, Loop 2	Read/Write	R138
TI3, Loop 2 (hi ^c)	Read/Write	R139
TI3, Loop 2 (lo ^c)	Read/Write	R140
TD3, Loop 2 (hi ^c)	Read/Write	R141
TD3, Loop 2 (lo ^c)	Read/Write	R142
Offset, Loop 2	Read/Write	R143
Limit1, Loop 2	Read/Write	R144
Limit2, Loop 2	Read/Write	R145

- a. R29 and R54 have the same content.
- b. R30 and R59 have the same content.
- c. The parameters TI_x/TD_x are measured in milliseconds (ms) and stored in two registers. The least significant word (lower 16 bits) is stored in the register denoted lo, and the most significant word (upper 16 bits) is stored in the register denoted hi.

Table 2 Register Map.

Writing K

Permissible values of K are

$$0.01 \leq K \leq 100.00$$

The registers only hold integer values. To be able to write and read K using communication, a simple conversion must be made:

$$K_{\text{COMLI}} = K * 100$$

which means that

$$1 \leq K_{\text{COMLI}} \leq 10000$$

Example

Send the value $K1 = 0.87$ to loop one.

Procedure:

- 1 Convert the value into integer format by multiplying by 100:
 $K1_{\text{COMLI}} = K1 * 100 = 0.87 * 100 = 87$
- 2 Send the converted value to R96.

Writing TI and TD

TI and TD are stored in two registers each. When writing either TI or TD, the value is not stored in the controller's PID-table until the lower part of the value is written to the unit. This means that when writing a two-register value, the most significant part must be written first, and the the least significant part must be written last.

Example 1

Send the value $TI1 = 125.7$ seconds to loop one.

Procedure:

1. Convert the value in milliseconds into hexadecimal format:
 $(125\ 700_{10} = 0001\ \text{EB}04_{16})$
2. Split the hexadecimal value into two 16-bit words:
 $(0001_{16}$ and $\text{EB}04_{16})$
3. Write the MSW (most significant word) to the first register of TI1...
 $(\text{Write } 0001_{16}, \text{ or } 1_{10} \text{ to R97})$
4. ... and write the LSW (Least Significant Word) to the second register of TI1
 $(\text{Write } \text{EB}04_{16}, \text{ or } 60\ 164_{10} \text{ to R98})$

First write the value $100\ 000_{10}$ to register R97, and then the value $25\ 700_{10}$ to register R98.

Example 2:

Send the value $TD2 = 2.6$ seconds to loop two.

Procedure:

1. Convert the value in milliseconds into hexadecimal format:
($2\,600_{10} = 0000\,0A28_{16}$)
2. Split the hexadecimal value into two 16-bit words:
(0000_{16} and $0A28_{16}$)
3. Write the MSW (most significant word) to the first register of TD2... (Write 0000_{16} , or 0_{10} to R136)
4. ... and write the LSW (Least Significant Word) to the second register of TD2
(Write $0A28_{16}$, or $2\,600_{10}$ to R137)

Gain scheduling

When the Gain Scheduling is Off, the parameter sets K2/TI2/TD2 and K3/TI3/TD3 are mirrors of the K1/TI1/TD1 set. This means that data written to a parameter in one set will be copied automatically to the other sets. The same applies to reading. The controller makes sure that the three sets are identical when reading and writing.

EMA60

EMA60 has no control function, and therefore no PID-parameter sets. Reading from registers R96–113 and 128–145 hence yields non-sense information. Also, writing to these registers has no impact on the function of this unit.

Appendix B

Special Modes

B.1 Total Reset

To reset all parameters in the configuration to factory default settings pressing the Cancel and Down keys simultaneously on the front panel of the controller when powering the unit up.

Index

A

AA 132
AA Source off 246
Absolute Alarm 132
Acknowledge 241
Adapt warn Loop1 246
Adapt warn Loop2 246
Adaptive control 138, 151
AI 64
AI Source off 246
AI1 Int 245
AI2 Int 245
AI3 Int 245
AI4 Int 245
AI5 Int 245
AIC 67
AIC (Analog Input Communication) 293
AIC Source off 246
Alarm acknowledge 293
Analog 181
Analog input 64
Analog Input Communications 67
Analog Inputs 32
Analog Output 176
Analog Output Communications 181
Analog User 68
AO 176
AO1 Int 245
AO2 Int 245
AO3 Int 245
AOC 181
AOC (Analog Output Communication) 292
Arithmetic error 245
Arithmetic Function Blocks 74

AddSub 75

Comp 79

Exp 78

MulDiv 76

AU 68

AU Source off 246

Autotuning 145, 149

B

Bar graphs

Output bar graph 51

Parameters 188

Process Value bar graph 50

Set Point bar graph 50

Bias 68

C

Calibration 245

CLPM Osc Loop1 245

CLPM Osc Loop2 245

CO 137

Columns 62

COMLI 287

Baud Rate 194

Communication 194

Communication 287

Communication memory area 292

Communication Mode (change) 294

Communication Requirements 287

Computer/Local mode 293

Configuration Mode

Entering 53

Exiting 55

Navigating 56

Configuration Tree Diagrams 249

ECA60 Configuration Tree 254

ECA600 Configuration Tree 261

EMA60 Configuration Tree 273

Configuration Work Sheet 281

Control Block 137

Control Keys 51

Cancel Key 52

Down Key 52

Hand Key 52

Next Key 52

OK Key 51

Up Key 52

Control Modes (change) 294

Controls 49

Critical Gain 153

Critical Period 153

CSP 119

D

DA Source off 246

DDC 158

Deviation Alarm 134

DI 71

DI Source off 246

DIC 72

DIC (Digital Input Communication) 293

DIC Source off 246

Digital Input 36, 71

Digital Input Communications 72

Digital Output 37, 182

Digital Output Communications 183

Digital User 73
Display 50, 186, 191, 239
DO 182
DO overload 245
DOC 183
DOC (Digital Output Communication) 292
DU 73
DU Source off 246

E

Engineering Units 184, 191
Error Behavior 66
ESP 119
Example (communication) 294

F

FA 74
FA Source off 246
Feed Forward 138, 143
 Adaptive 145
 Static 144
FL 82
FL Source off 246
FLASH Checksum 245
FLASH Erase 245
FLASH Write 245
FO 88
FO Source off 246
Forced Output 162
Front panel 49
Front Panel Usage 295

G

Gain Scheduling 138, 140

H

Hand key lock 192
Hi alarm ... 247
Hi-hi alarm ... 247
HW 245

I

Indicators 49
Input Function Blocks 74
Installation 17
 Cassette 19

Communications signal wiring 38

Electrical 25
gasket 23
I/O Signal Wiring 31
In an Electrically Noisy Environment 38
IP65 23
Mounting Plates 23
Panel Cutout 17
Power Wiring 27

Internal connections 60

J

Jumper Module 30

L

Language 197
Lo alarm ... 247
Logic Function Blocks 82
 And 82
 Not 87
 S/R 86
 Xor 84
Lo-lo alarm ... 247
Loops (change) 295
LSP 119

M

Manual Mode 157
Manual mode 293
Master 287
Miscellaneous Block 154
ModBus RTU 287

N

Naming blocks 60
Neg dev ... 247

O

OC 155
OC Source off 245
OF 166
OF Source off 245
OP 184, 190
Operator Block controllers 184

EMA60 190

Other Function Blocks 88

Delay 97
IncDec 98
Level Detector 95
Linear 101
Max 91
Min 89
Rate Limiter 105
Select 93
Time Measurement 111

Output Configuration 155

Output Functions 166
 Analog 166
 Split Range 172
 Three State Pulse 168
 Two State Pulse 166
Outputs 176

P

Password 53, 196
 Lost code retrieval 55
PID 293
PID parameters 139
 derivative time 139
 gain 139
 integral time 139
 K 139
 Offset 139
 TD 139
 TI 139
Pos dev ... 247
pPI control 138
pPi control 147
Preset Configurations 196
Protocol (communication) 287
PV 293

R

RAM 245
Ratio 68
RS232 287
RS485 287
RSP 119

S

Sample Time 196
Sample Time Error 245
SattGraph 5000 297
Set Point Block 116
Slave 287
SP 116
SP Source off 246
Special Modes 303
Status signals 198
Supervisory control system
 287
SY 193
System Block 193

T

Tag 187, 191
Technical Data 201
Test Mode 198
Total Reset 303
Tracking 159
True Manual 158
Tune abort Loop1 246
Tune abort Loop2 246
Tune PVNoise Loop1 246
Tune PVNoise Loop2 246
Tuner 145
Tuning Init 248
Tuning Phase ... 247, 248



ABB Automation Technology Products AB
Instrumentation
SE-191 85 Sollentuna, Sweden

Phone: +46 8 658 80 00
Fax: +46 8 658 80 21
e-mail: instrumentation@se.abb.com
www.abb.se

Specifications subject to change without notice.
Printed in Sweden. © 2000 ABB Automation Technology Products AB.

493-0736-11
0230