EPC2000 Programmable Controller User Guide

HA033210ENG Issue 5

April 2023

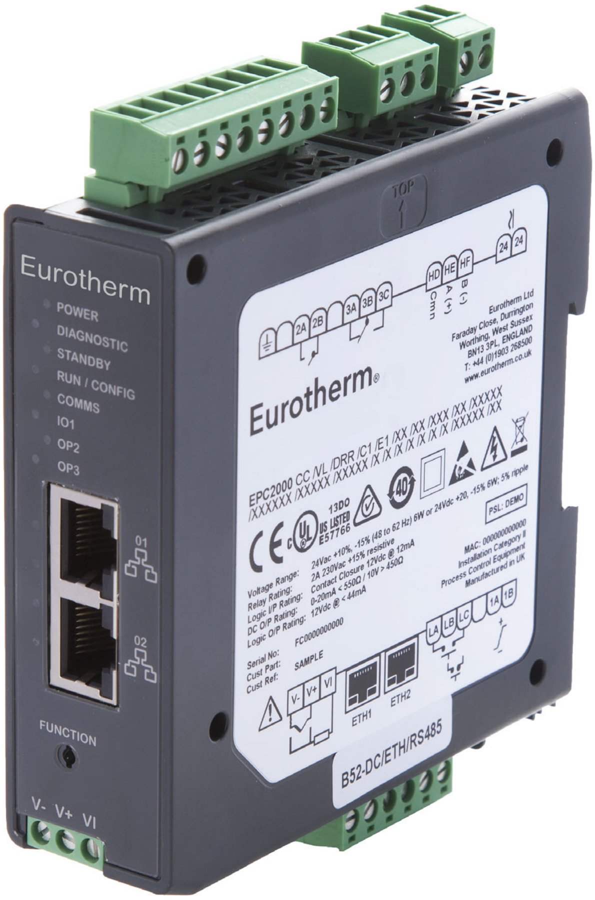
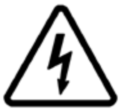


Table of Contents

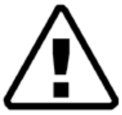
Safety Information

Important Information

Read these instructions carefully and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this manual or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of either symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates a hazardous situation which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a hazardous situation which, if not avoided, **could result in** death or serious injury.

⚠ CAUTION

CAUTION indicates a hazardous situation which, if not avoided, **could result in** minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to physical injury. The safety alert symbol shall not be used with this signal word.

Note: Electrical equipment must be installed, operated, serviced and maintained only by qualified personnel. No responsibility is assumed by Watlow for any consequences arising out of the use of this material.

Note: A qualified person is one who has skills and knowledge related to the construction, installation, and operation of electrical equipment, and has received safety training to recognize and avoid the hazards involved.

Before you Start

Important Information

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Turn off all power to all equipment before starting the installation, removal, wiring, maintenance or inspection of the product.

Use a properly rated voltage sensing device to confirm the power is off.

Power line and output circuits must be wired and fused in compliance with local and national regulatory requirements for the rated current and voltage of the particular equipment. i.e. UK, the latest IEE wiring regulations, (BS7671), and USA, NEC Class 1 wiring methods.

Failure to follow these instructions will result in death or serious injury.

Reasonable use and responsibility

The safety of any system incorporating this product is the responsibility of the assembler/installer of the system.

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

This programmable controller is intended for industrial temperature and process control applications, which meet the requirements of the European Directives on Safety and EMC.

Use in other applications, or failure to observe the installation instructions of this manual may compromise safety or EMC. The installer must ensure the safety and EMC of any particular installation.

To comply with the European EMC directive certain installation precautions are necessary:

- General guidance. Refer to EMC Installation Guide, Part no. HA025464
- Relay outputs. It may be necessary to fit a suitable filter to suppress conducted emissions
- Table top installation. If using a standard power socket, compliance with commercial and light industrial emissions standard is required. To comply with conducted emissions standard, a suitable mains filter must be installed.

Failure to use approved software/hardware with our hardware products may result in injury, harm, or improper operating results.

Please Note

Electrical equipment must be installed, operated, serviced, and maintained only by qualified personnel.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

No responsibility is assumed by Watlow for any consequences arising out of the use of this material.

Qualification Of Personnel

Only appropriately trained persons who are familiar with and understand the contents of this manual and all other pertinent product documentation are authorized to work on and with this product.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical, electrical, or electronic equipment.

The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

Intended Use

The product described or affected by this document, together with software and options, is the EPC2000 Programmable Controller (referred to herein as “programmable controller”, "controller" or "EPC2000"), intended for industrial use according to the instructions, directions, examples, and safety information contained in the present document and other supporting documentation.

The product may only be used in compliance with all applicable safety regulations and directives, the specified requirements, and the technical data.

Prior to using the product, a risk assessment must be performed in respect of the planned application. Based on the results, the appropriate safety-related measures must be implemented.

Since the product is used as a component within a machine or process, you must ensure the safety of this overall system.

Operate the product only with the specified cables and accessories. Use only genuine accessories and spare parts.

Any use other than the use explicitly permitted is prohibited and can result in unanticipated hazards.

Dangers and Warnings

The following Dangers and Warnings should be observed before installing, operating or servicing this controller. Further safety information is provided at appropriate places in this manual.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Do not insert anything through the case apertures.

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to all equipment before starting the installation, removal, wiring, maintenance or inspection of the product.

Power line and output circuits must be wired and fused in compliance with local and national regulatory requirements for the rated current and voltage of the particular equipment, i.e. UK, the latest IEE wiring regulations, (BS7671), and USA, NEC class 1 wiring methods.

The unit must be installed in an enclosure or cabinet.

Do not exceed the device's ratings.

This product must be installed, connected and used in compliance with prevailing standards and/or installation regulations. If this product is used in a manner not specified by the manufacturer, the protection provided by the product will be impaired.

All connections must be tightened in conformance with the specified torque specifications.

Ensure a maximum of two wires, identical in type and cross sectional size are used for each controller terminal.

Do not exceed the maximum exposed conductor length.

Ensure only a suitable insulated tool that fits the aperture is used to depress the function button when required.

Failure to follow these instructions will result in death or serious injury.

⚠ DANGER

FIRE HAZARD

Do not install if the unit, or any part of the unit is damaged.

Do not allow anything to fall through the case apertures and ingress the controller.

Ensure the correct wire gauge size is used and it is rated for the current capacity of the specific circuit.

Ensure when using ferrules (cable ends) that the correct size is selected and each is securely fixed to the wire using a crimping tool.

Ensure only the originally supplied connectors are used.

Do not connect the controller directly to line voltage. Use only isolating PELV or SELV power supplies to supply power to the equipment.

Failure to follow these instructions will result in death or serious injury.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not use, or implement a controller configuration (control strategy) into service without ensuring the configuration has completed all operational tests, been commissioned and approved for service.

During commissioning ensure all operating states and potential fault conditions are carefully tested.

It is the responsibility of the person commissioning the controller to ensure the configuration is correct.

Do not use the product for critical control or protection applications where human or equipment safety relies on the operation of the control circuit or the triggering of an alarm.

Ensure all electrostatic discharge precautions are taken before handling the unit.

Ensure electrically conductive pollution is be excluded from the cabinet in which the controller is mounted.

Ensure appropriate safety interlocks are used where personnel and/or equipment hazards exist.

The unit must be installed in an enclosure or cabinet.

Power line and output circuits must be wired and fused in compliance with local and national regulatory requirements for the rated current and voltage of the particular equipment, i.e. UK, the latest IEE wiring regulations, (BS7671), and USA, NEC class 1 wiring methods.

Ensure all cables and wiring harness are secured using a relevant strain relief mechanism.

Use Copper cables only (except for thermocouple wiring).

Only connect wires to identified terminals shown on the product wiring label, the wiring section of the product User guide or Installation sheet.

Ensure before connecting wires to any connectors that the connector’s orientation is correct

Modifying, disassembling or repairing the product beyond what is stated in the User guide is strictly forbidden. Contact your supplier for repair.

If the output is not wired, but written to by communications, it will continue to be controlled by the communications messages. In this case take care to allow for the loss of communications.

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise must be allowed to program, install, alter and commission this product.

The controller must not be configured while it is connected to a live process as entering Configuration Mode pauses all outputs. The controller remains in Standby until Configuration Mode is exited.

Failure to follow these instructions can result in death, serious injury or equipment damage.

⚠ WARNING

UNANTICIPATED EQUIPMENT OPERATION

Ensure wires are routed to minimize the pick-up of EMI (Electromagnetic interference) keeping cable lengths to a minimum.

Failure to follow these instructions can result in death, serious injury or equipment damage.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

If on receipt, the unit or any part within is damaged, do not install but contact your supplier.

If being stored before use, store the controller within the specified environmental conditions.

To minimize any potential loss of control or controller status when communicating across a network or being controlled via a third party master (i.e. another controller, PLC or HMI) ensure all system hardware, software, network design, configuration and cybersecurity robustness have been correctly configured, commissioned and approved for operation.

Failure to follow these instructions can result in injury or equipment damage.

Symbols

Various symbols may be used on the controller. They have the following meaning:

⚠ Risk of electric shock.

⚠ Take precautions against static.

N1981 Regulatory compliance mark for Australia (ACA) and New Zealand (RSM).

40 Complies with the 40 year Environment Friendly Usage Period.

Hazardous Substances

This product conforms to European **R**estriction **o**f **H**azardous **S**ubstances (RoHS) (using exemptions) and **R**egistration, **E**valuation, **A**uthorisation and Restriction of **Ch**emicals (REACH) Legislation.

RoHS Exemptions used in this product involve the use of lead. China RoHS legislation does not include exemptions and so lead is declared as present in the China RoHS Declaration.

Californian law requires the following notice:

⚠WARNING: This product can expose you to chemicals including lead and lead compounds which are known to the State of California to cause cancer and birth defects or other reproductive harm. For more information go to: http://www.P65Warnings.ca.gov

Cybersecurity

What's in this Chapter?

This chapter outlines some good practice approaches to cybersecurity as they relate to the use of the EPC2000 controller, and draws attention to several EPC2000 features that could assist in implementing robust cybersecurity.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

To minimize any potential loss of control or controller status when communicating across a network or being controlled via a third party master (i.e. another controller, PLC or HMI) ensure all system hardware, software, network design, configuration and cybersecurity robustness have been correctly configured, commissioned and approved for operation.

Failure to follow these instructions can result in injury or equipment damage.

Introduction

When utilizing a Eurotherm EPC2000 controller in an industrial environment, it is important to take 'cybersecurity' into consideration: in other words, the installation's design should help prevent unauthorized and malicious access. This includes both physical access to the control equipment and associated devices, and electronic access (via network connections and digital communications).

Cybersecurity Good Practices

Overall design of a site network is outside the scope of this manual. The Cybersecurity Good Practices Guide, Part Number HA032968 provides an overview of principles to consider. This is available from www.eurotherm.com.

Typically, an industrial controller such as the EPC2000 Programmable Controller together with any associated HMI screens and controlled devices should *not* be placed on a network with direct access to the public Internet. Rather, good practice involves locating these devices on a firewalled network segment, separated from the public Internet by a so-called 'demilitarized zone' (DMZ).

Security Features

The sections below draw attention to some of the cybersecurity features of the EPC2000 controller.

Principle of Secure by Default

Some of the digital communication features on the EPC2000 can provide greater convenience and ease-of-use (particularly in regards to initial configuration), but also can potentially make the controller more vulnerable. For this reason, the following feature is turned off by default:

Bonjour auto-discovery disabled by default

Ethernet connectivity is supplied as default on the EPC2000 controller, including the Bonjour service discovery protocol (see "Bonjour" on page 229). Bonjour enables the controller to be automatically discovered by other devices on the network without the need for manual intervention. However, for cybersecurity reasons, it is disabled by default, as it could be exploited by a malicious user to gain information about the controller.

See also section "Auto Discovery" on page 229 and information on how to turn it on, if required.

Port Use

The following ports are being used:

Port	Protocol
44818 TCP/UDP	EtherNet/IP (see below)
22112 UDP	EtherNet/IP (see below)
2222 UDP	EtherNet/IP (see below)
502 TCP	Modbus (Master and Slave)
5353 UDP	Zeroconf

The following should be noted about the EtherNet/IP ports:

- Ports are always closed by default and are only opened when the corresponding comms protocol is set.
- UDP Port 5353 (Auto-discovery/ZeroConf/Bonjour, open only when Comms.Option.Network.AutoDiscovery parameter is ON.

Access Control

The EPC2000 controller has two levels of access - Operator mode and Configuration mode. Operator mode provides basic functionality required on a day-to-day basis whereas Configuration mode provides full functionality for Initial set-up and process configuration. Passwords are supported by default to control access to configuration mode. Strong passwords should be used (see below). After five unsuccessful login attempts, password entry is blocked for the configured Instrument>Security>PassLockTime, set to 30 minutes by default. A powercycle resets the elapsed time back to 0. This helps protect against ‘brute force’ attempts to guess a password.

Strong Passwords

It is recommended that a strong password is used for the Configuration password and OEM Security password. By ‘strong’, we recommend a password that is:

- At least eight characters in length.
- Has a mixture of both upper and lower case characters.
- Has a minimum of one special punctuation character (#, %, or @ for example).
- Has at least one numeric digit.

NOTICE

POTENTIAL LOSS OF INTELLECTUAL PROPERTY OR CONFIGURATION

Ensure all passwords configured in the programmable controller are ‘strong’ to help prevent the loss of intellectual property or unauthorized configuration changes.

Failure to follow these instructions can result in equipment damage.

OEM Security

An optional OEM security feature is provided to give Original Equipment Manufacturers (OEMs) a layer of protection against theft of their intellectual property, and is designed to help prevent unauthorized cloning of controller configurations. This protection includes application-specific internal (soft) wiring and limited access to certain parameters via comms (by iTools or a third party comms package).

Comms Configuration Level Password

The password for Configuration Level access via iTools has the following features to help protect against unauthorized access (see "Instrument.Security" on page 95 for more details):

- There is no default password for comms configuration level.
- User needs to set the comms configuration password on first connect from iTools.
- If password is not set, serial and Ethernet comms will be in Comms Lockdown mode (see below).
- Comms configuration password is encrypted before sending via comms.
- Passwords are salted and hashed before being stored.
- Number of password attempts is 5. If more than 5 unsuccessful attempts are made, the Password Lock function is triggered.
- iTools will enforce a minimum password length of 8 characters.

Comms Lockdown mode

In Comms Lockdown mode, serial and Ethernet comms will only have read/write access to a limited set of parameters, see the following table.

Table 1: Comms Lockdown Limited Parameter Set

Parameter	Modbus Address	Access	String length
CNOMO Manufacturing ID	0x0079(121)	Read-only	-
CNOMO Instrument ID	0x007A(122)	Read-only	-
Instrument firmware version	0x006B(107)	Read-only	-
CommsPasswordIsSet	0x0080(128)	Read-only	-
KeyExchange	0x53F4(21492)	Read/Write	35
CommsPassword	0x5621(22049)	Write-only	96

Ethernet security features

Ethernet connectivity is available on the EPC2000 controller. The following security features are specific to Ethernet.

Ethernet rate protection

One form of cyberattack is to try to make a controller process so much Ethernet traffic that it drains systems resources and useful control is compromised. For this reason, the EPC2000 Programmable Controller includes an Ethernet rate protection algorithm, which will detect excessive network activity and help to ensure the controller’s resources are prioritized on the control strategy rather than servicing the Ethernet traffic. If this algorithm is active, the RateProtectionActive parameter will be set to ON (see "Comms.Serial.Network and Comms.Ethernet.Network" on page 129).

Broadcast Storm protection

A ‘broadcast storm’ is a condition which may be created by cyberattack, whereby spurious network messages are sent to devices which cause them to respond with further network messages, creating a chain reaction that escalates until the network is unable to transport normal traffic. The EPC2000 includes a broadcast storm protection algorithm, which will automatically detect this condition, stopping the controller from responding to the spurious traffic. If this algorithm is activated, the BroadcastStormActive parameter will be set to ON (see "Comms.Serial.Network and Comms.Ethernet.Network" on page 129).

Communications watchdog

The EPC2000 controller includes a ‘comms watchdog’ feature. This can be configured to raise an alert if any of the supported digital communications are not received for a specified period of time. See the watchdog parameters in "Comms.Serial.Main and Comms.Ethernet.Main" on page 127. These provide a way to configure appropriate action if malicious action interrupts the controller’s digital communications.

Note: This watchdog may not function as expected for multiple Ethernet connections, due to the shared timer and flag for this interface. If the device is configured to receive a setpoint from a remote master via Ethernet connection, it should be routed through the ‘Remote Input’ block ("RemoteInput" on page 113). This block has an independent timeout (default to 1s), allowing the loss of comms to this parameter to be flagged independently of any other Ethernet connections.

Configuration backup and recovery

Using Eurotherm’s iTools software, you can ‘clone’ an EPC2000 controller, saving all its configuration and parameter settings to a file. This can then be copied onto another controller, or used to restore the original controller’s settings—see "Cloning" on page 88.

For cybersecurity reasons, password-restricted parameters are not saved in the clone file.

Clone files include a cryptographic integrity hash, meaning that if the file contents are tampered with, the file will not load back into a controller.

A clone file cannot be generated if the OEM Security feature option is configured and active (see "OEM Security" on page 290).

User Sessions

Communication connections only have two permission levels - an ‘Operator mode’ and a ‘Configuration mode’. Any connection via comms (Ethernet or serial) is separated into its own unique session. One connected user will not share permissions with another. Likewise, a user logged in on the serial connection does not share permissions with anyone logged in via Ethernet and vice-versa.

In addition, only a single user can be logged in to an EPC2000 Programmable Controller in Configuration mode at any one time. If another user attempts to connect and select Configuration mode, the request will be denied until the other user exits the Configuration mode.

User sessions are not persistent across power cycles.

Data Integrity

Flash Integrity

When an EPC2000 controller powers up, it automatically performs an integrity check on the entire contents of its internal flash . If the main application is detected as being corrupt, then the internal firmware update application runs and waits for the Eurotherm Firmware Management Tool to upgrade the firmware. Refer to "Firmware Upgrade" on page 295. The LEDs from Diagnostic to OP3 all flash. If the internal firmware upgrade application is also corrupt, the red Diagnostic LED will be lit and advice must be sought from the manufacturer.

Periodic integrity checks are also performed in 256 byte blocks during normal runtime. If any integrity check detects a difference from what is expected, the controller will stop running and restart.

Non-volatile Data Integrity

When an EPC2000 controller powers up, it automatically performs an integrity check on the contents of its internal non-volatile devices. If the controller is unable to load the parameter database due to corruption, it will attempt to reset the device and cold start.

Additional periodic integrity checks are performed during normal runtime and when non-volatile data is being written. If any integrity check detects a difference from what is expected, the controller enters Standby mode and sets bit 1 or bit 2 in the Instrument.Diagnostics function block, Standby Status Word parameter (refer to "Standby Status Word Bitmap" on page 100 and "Instrument.Diagnostics" on page 97).

Cryptography Usage

Cryptography usage is employed in the following areas:

- Clone files.
- Custom linearization tables.
- Firmware signing.
- OEM security password.
- Comms Configuration password.

Firmware

From time to time, to provide new functionality or address known issues, Eurotherm may make new versions of the EPC2000 firmware available via the Eurotherm Firmware Management Tool.

⚠ CAUTION

NON-EUROTHERM FIRMWARE

The EPC2000 Programmable Controller uses cryptographic digital signing technology to prevent an attacker from loading non-genuine firmware onto the device. Deliberate attempts to force an upgrade to unofficial firmware will, at worst, result in a non-operational device.

In addition, the Eurotherm Firmware Management Tool is digitally signed. Do not use this tool if it has not been signed.

Failure to follow these instructions can result in injury or equipment damage.

Achilles® Communications Certification

The EPC2000 has been certified to Level 1 under the Achilles® Communications Robustness Test Certification scheme. This is an established industry benchmark for the deployment of robust industrial devices recognized by the major automation vendors and operators.

Note: Achilles® Communications Robustness Testing Level 1 is not applicable to the following version; EPC2000 PROFINET.

Decommissioning

When an EPC2000 controller is at the end of its life and being decommissioned, Eurotherm advises reverting all parameters to their default settings (see "Cold Start" on page 89 for instructions). This can help to protect against subsequent data and intellectual property theft if the controller is then acquired by another party.

General EPC2000 Cybersecurity Considerations

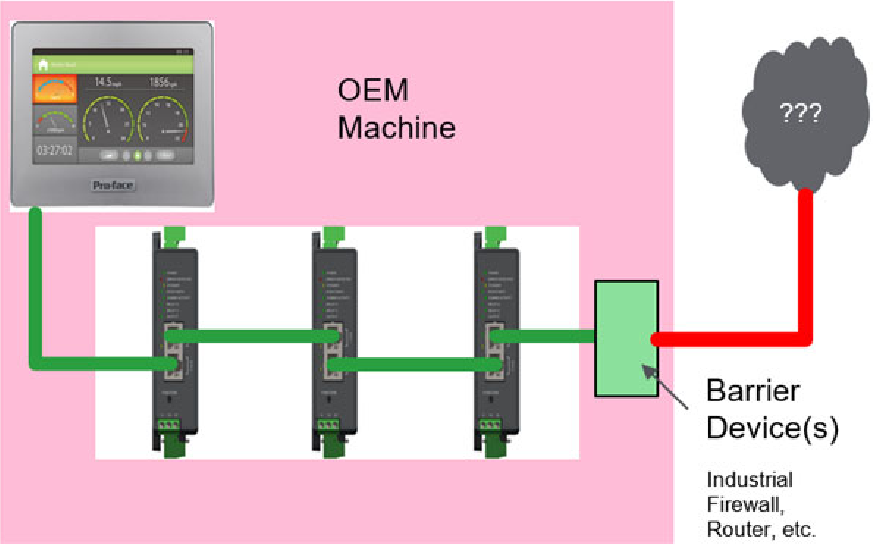
Network Topology for EPC2000/External HMI

The EPC2000 Programmable Controller is a 'blind' back of panel instrument (i.e. it does not have an integrated HMI display). However, an external HMI panel (for example, a Proface GP-4100 series) can be connected to the EPC2000 Programmable Controller via one of the digital communications channels.

When an external HMI panel is connected, cybersecurity implications should be considered. Specifically, to mitigate the risk of a denial of service on the communication channel connecting the two devices, otherwise operator actions performed on the HMI maybe prevented from being actioned by the EPC2000 Programmable Controller. The following two EPC2000-to-HMI network topologies would aid in mitigating this risk.

EPC2000-to-HMI Ethernet Network Segmentation

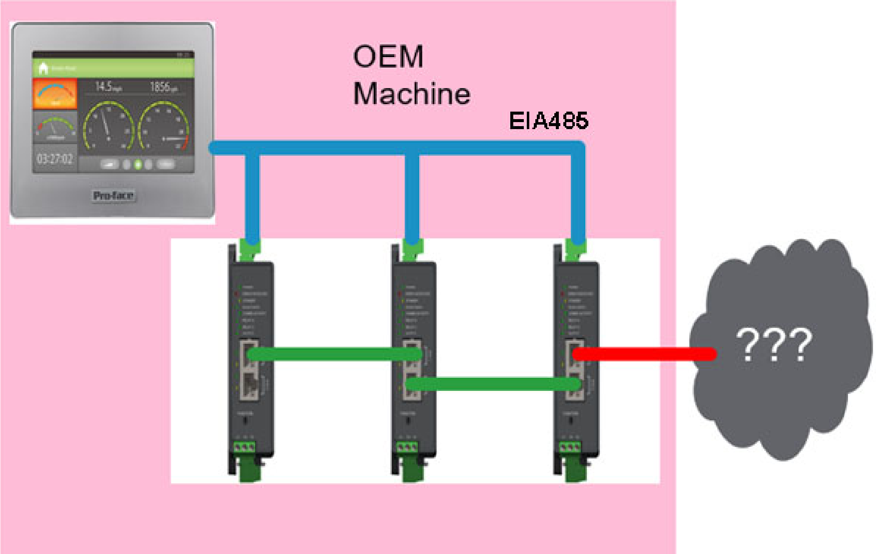
The use of Ethernet network barrier devices (for example, Industrial Firewall, Router, and so on) is required to segment the internal machine network from other external networked devices and connections.



Additionally, it is recommended that the EPC2000 Programmable Controller 'preferred master' parameters are configured with the IP address of the HMI panel to help ensure that the HMI is able to connect to the EPC2000 Programmable Controller even if the other TCP sessions are currently active.

EPC2000-to-HMI via EIA485 Communications

Alternatively, dedicate the EIA485 communications channel to the EPC2000-to-HMI network and use the Ethernet communication channel to connect to other networked devices. This will prevent a denial of service attack OR network misconfiguration from disconnecting the HMI from the EPC2000 Programmable Controller(s).



It should be noted that EIA485 communications is given priority over Ethernet communications, however, EIA485 is slow in comparison and therefore consideration should be given to the latency between the EPC2000 Programmable Controller and the HMI via EIA485.

External HMI Application Security Considerations

The EPC2000 Programmable Controller security functionality, as detailed in the above sections, provides several mechanisms which should be considered when developing an external HMI application. Points to consider are as follows:

- The EPC2000 Programmable Controller has two modes of operation, Operator and Configuration. If an external HMI is being utilized, additional access levels and user management could be implemented in the HMI application which restricts specific operations depending upon roles and user permissions.
- Comms configuration password is encrypted before transmission via comms in EPC2000 V4.01 onwards. However, in order to continue support for external HMIs, a new parameter Instrument>Security>EnableUnencryptedLogin is made available. It is important to note that transmission of plain text passwords is a cybersecurity risk that will need to be mitigated by the user.
- Access to the EPC2000 Programmable Controller configuration mode is via a password. There are a number of diagnostic parameter values which are used to indicate:
  - that the Comms Configuration password has not been set.
  - the Comms Configuration password has expired.
  - the number of successful/unsuccessful login attempts.
- If an external HMI is being utilized, these password diagnostics could be periodically read from the device and visualized in the HMI application.

Legal Information

The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Eurotherm Limited, Watlow or any of its affiliates or subsidiaries shall not be responsible or liable for misuse of the information contained herein.

If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, non-commercial use, all or part of this document on any medium whatsoever without permission of Eurotherm Limited, given in writing. You also agree not to establish any hypertext links to this document or its content. Eurotherm Limited does not grant any right or license for the personal and non-commercial use of the document or its content, except for a non-exclusive license to consult it on an ‘as-is’ basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Eurotherm Limited software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

Watlow, Eurotherm, EurothermSuite, EFit, EPack, EPower, Eycon, Chessell, Mini8, nanodac, piccolo and versadac are trademarks and property of Watlow, its subsidiaries and affiliated companies. All other trademarks are the property of their respective owners.

Introduction

Controller Concept

The EPC2000 Programmable Controller is a programmable single loop process controller which is certified for cybersecurity communications robustness. A range of math, logic, totalizer and specialized functions are also available.

Simple ‘Quick Codes’ can be used to configure standard applications quickly for controlling specific processes. Applications include heat and heat/cool temperature control. These applications are pre-configured providing the user with a starting point for customizing to an individual process.

Eurotherm iTools is a software package which is designed for this purpose by providing user function block wiring in addition to a range of other features. It is available as a free download from www.eurotherm.com or can be ordered on a DVD.

User Guide Concept

This guide is generally laid out in the following manner:

- The first part explains mechanical and electrical installation and covers the same topics as in the Installation and wiring sheet supplied with each instrument but in more detail.
- Operation of the instrument, including start-up operation. In general the descriptions in the user guide assume that the controller is configured with no application loaded or with a heat or heat cool controller application loaded.
- Configuration of the instrument using Eurotherm iTools configuration package.
- Description of different function blocks in the instrument, such as Control Loop, Programmer, Digital Communications.
- Calibration procedure.
- OEM Security feature description.
- Updating the EPC2000 Programmable Controller firmware.
- Technical specification.

Installation

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to product and all I/O circuitry (alarms, control I/O, etc.) before starting the installation, removal, wiring, maintenance or inspection of the product.

Failure to follow these instructions will result in death or serious injury.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise are allowed to program, install, and commission this product.

During commissioning, carefully test for all operating states and potential fault conditions.

Failure to follow these instructions can result in death, serious injury or equipment damage.

In typical temperature control processes there may be issues when heating is constantly on. Reasons why the heating might remain constantly on include:

- The temperature sensor becoming detached from the process.
- Thermocouple wiring becoming short circuit.
- The controller heating being constantly on.
- An external valve or contactor sticking in the heating condition.
- The controller setpoint too high.
- Loss of communications.

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

Alarm relays do not provide protection under all failure conditions and should not be relied on.

What’s in this Chapter?

- A general description of the instrument.
- What is in the package.
- Order codes
- Instrument dimensions and mechanical mounting

What Instrument Do I Have?

Thank you for choosing this controller. The EPC2000 Programmable Controller provides precise control of industrial processes.

The EPC2000 Programmable Controller is powered by a separate power supply, for further details see "Technical Specification".

Input and output options

All controllers are supplied with Ethernet connectivity via a two-port switch on the front panel allowing daisy-chaining if required. Additionally, controllers can be supplied either with or without EIA-485 serial digital communications.

The following inputs and outputs are supplied as standard:

- Sensor input to accept various thermocouples, RTDs, voltage or current inputs.
- Two digital contact inputs.
- Normally-open relay output.
- Changeover relay output.

The following inputs and outputs can be specified when ordering:

- Analog output.

  OR

- An I/O connection that can be configured to be either Logic (SSR drive) output OR contact input.

Labels fitted to the sleeve show the ordering code, serial number, date of manufacture, and terminal connections for the hardware fitted.

Unpacking Your Controller

The controller is supplied with:

- A 2.49Ω resistor for a current input (see "Linear Input (mA, mV or V)" on page 40).
- Installation sheet Part Number HA033209 in English, French, Italian, German, Spanish, Chinese and Russian.

⚠ **DANGER**

FIRE HAZARD

If on receipt, the unit or any part within is damaged, do not install but contact your supplier.

Only use the terminal harness connectors originally supplied with the controller.

Ensure the correct wire gauge size is used per circuit and it is rated for the current capacity of the circuit.

Failure to follow these instructions will result in death or serious injury.

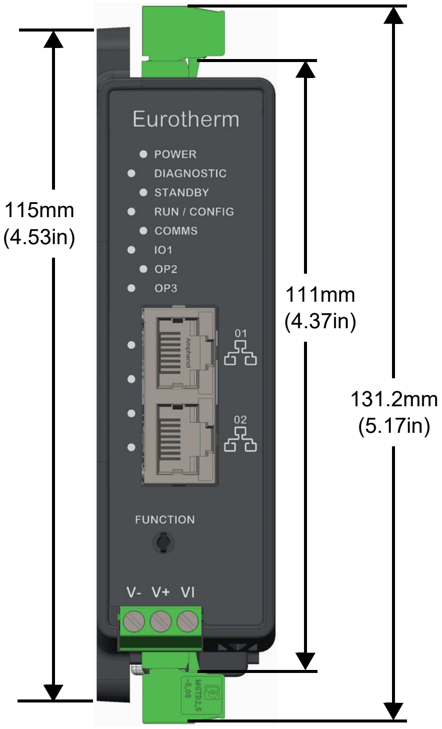
Order Codes

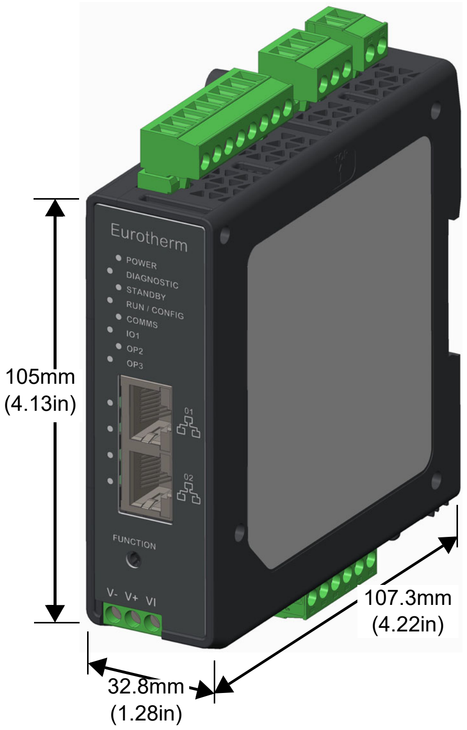
For the latest order codes please refer to the EPC2000 Programmable Controller Data Sheet (HA033270) which can be found at https://www.eurotherm.com > Support > Downloads (https://www.eurotherm.com/en/eurotherm-downloads/).

Dimensions

General views of the controller are shown below together with overall dimensions.

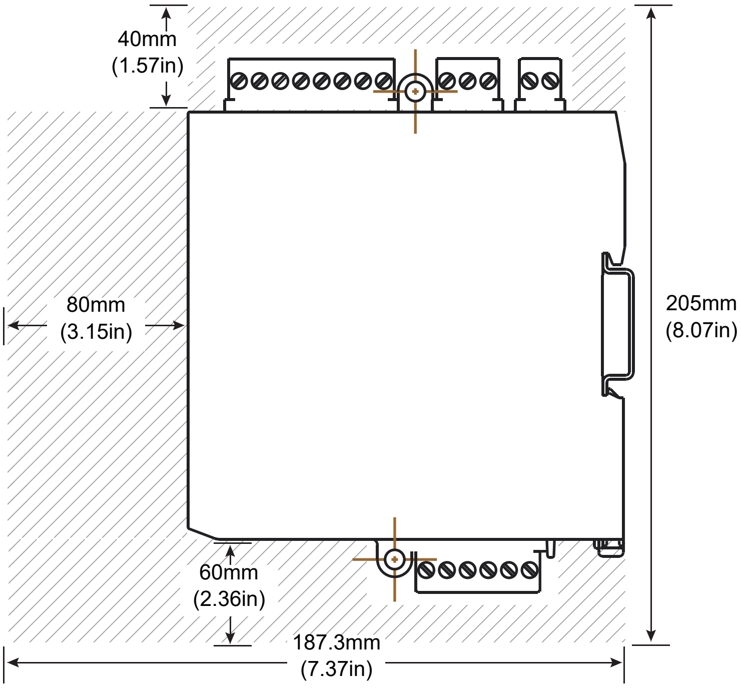
The dimensions of the EPC2000 Programmable Controller are shown in the following figures. The height shown is with the factory-fitted default connectors attached.





Service Access

To enable operator access to the controller and its connectors, space should be made available around the controller as shown in the figure below.



Location

This controller is intended for permanent installation, for indoor use only, and in an enclosure or cabinet.

Select a location which is subject to minimum vibrations, the ambient operating temperature is within 0 and 55°C (32 - 131°F) and operating humidity of 5 to 90% RH non-condensing.

The controller can be either:

- Mounted on a DIN rail.
- Surface mounted.

Please read the safety information in "Safety and EMC" on page 11 before proceeding.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

The unit must be installed in an enclosure or cabinet. Failure to do this impairs the safety of the unit.

Failure to follow these instructions will result in death or serious injury.

General mounting instructions

The product can be mounted using the surface mount fixing holes, or using DIN rail. Refer to the Installation and Wiring Sheet (HA033209) for details.

- Ensure the controller is mounted on a vertical flat surface.
- Mount vertically in an upright direction, so the release clip is located on the bottom side.
- Additional space above and below the controller should be provided to allow cable and connector access.
- Process variable terminals are sensitive to surrounding temperatures. See "Technical Specification" on page 298 for further details.
- Ensure all cables and wiring harnesses are secured using a relevant strain relief mechanism.

Surface mounting the controller

There are two M4 lugs on the top and bottom of the controller. These are to be used for surface mounting.

Mount vertically in an upright direction, on a flat surface so the release clip is positioned on the underside.

Mounting the Controller on a DIN rail

Mount using standard DIN rail, EN50022 (TH 35x7.5) or EN50022 (TH 35x15) and stoppers at each end.

1. Position the top groove of the controller on the top edge of the DIN rail and press the assembly against the DIN rail until you hear or feel the release clip click into place.
2. Confirm the controller is securely attached.

Spacing for controllers

The primary measurement input (IP1) is sensitive to surrounding temperatures. Therefore, sufficient space should be provided between the EPC2000 Programmable Controller and any potential heat sources.

Removal of the Controller

DIN Rail

To remove the controller from a DIN rail:

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to product and all I/O circuitry (alarms, control I/O, etc.) before starting the installation, removal, wiring, maintenance or inspection of the product.

Failure to follow these instructions will result in death or serious injury.

1. Remove power from the controller and from any connected I/O.
2. Remove the bottom terminal block (to gain access to the release clip).
3. Using a flat-bladed screwdriver, lever the release clip downward.
4. Tilt the bottom of the controller towards you.
5. Remove the controller from the DIN rail.

Surface Mount

To remove the controller from a surface-mount installation:

⚠ ⚠ **DANGER**

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to product and all I/O circuitry (alarms, control I/O, etc.) before starting the installation, removal, wiring, maintenance or inspection of the product.

Failure to follow these instructions will result in death or serious injury.

1. Remove power from the controller and from any connected I/O.
2. Unscrew the two M4 fixings from the top and bottom of the controller and remove the product.

Terminal Wiring and Connections

What's in this Chapter?

This chapter describes terminal connections and wiring.

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

Ensure all cables and wiring harnesses are secured using a relevant strain relief mechanism.

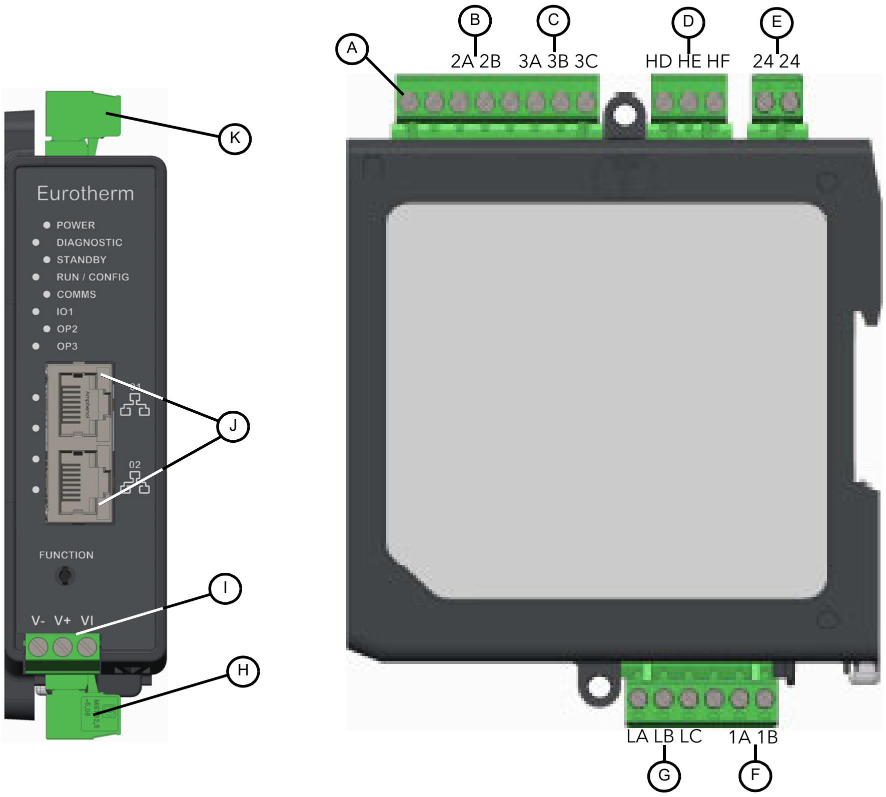
Avoid ingress of conductive materials during installation.

Only connect wires to identified terminals shown on the product label, the Wiring section of the product user guide or Installation sheet.

Ensure before connecting wires to any terminal harness connector that the harness connector's orientation is correct - especially if a connector is disconnected from the unit.

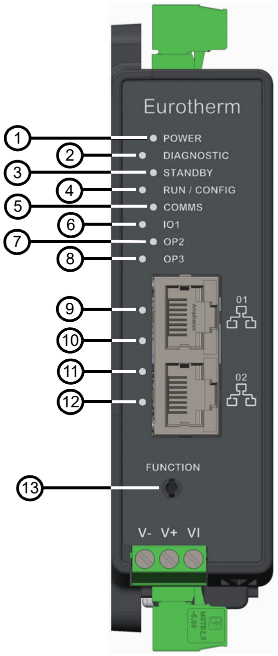
Failure to follow these instructions can result in death, serious injury or equipment damage.

EPC2000 Programmable Controller Terminal Layout



Key	Title	Terminals	Function
A	Functional ground connection		Functional ground connection point
B	OP2 (Output 2) Normally Open (NO) Common (C)	2A 2B	Relay Form A (Normally Open)
C	OP3 (Output 3) Normally Closed (NC) Common (C) Normally Open (NO)	3A 3B 3C	Relay Form C (Change Over Relay)
D	COMMS (Serial Communication) COM: A(+) RX: B(-) TX:	HD HE HF	EIA-485
E	Power Input (Low voltage only)	24 24	24Vac/Vdc
	Fuses should be provided externally. Recommended fuse type, Slow-blow/Time delay rated 2A 250V. • Use copper conductors only. • A switch or circuit breaker must be included in the building installation. It shall be in close proximity to the equipment and within easy reach of the operator. It shall be marked as the disconnecting device for the equipment. Note: A single switch or circuit breaker can drive more than one instrument.		
F	IO1 (Input/Output 1) - Option 1	1A (+) 1B(-)	Analog output
	IO1 (Input/Output 1) - Option 2 Note: Multiple output/input functions depending on controller configuration purchased.	1A (+) 1B(-)	Logic (SSR drive), output Or Contact Input, connected to Logic output
G	DI x2 (Digital Input)	LA, LB, LC	
	Digital Input 1:	LA, LC	Contact Input
	Digital Input 2:	LB.LC	Contact Input
H	Bottom harness connectors 6-pin	see F & G	Various connections
I	IP1 (Input 1), Sensor Measuring Analog Input • Do not run input wires with power cables. • Ground shielded cable at one point only. • Sensor input not isolated from the logic output & digital inputs. • Use appropriate compensating cable to extend thermocouple cabling. Note: Fixed connector - not removable.	V- V+ Or	Thermocouple (TC)
		V- V+ VI Or	Resistance Temperature Detector (RTD)
		V- V+ Or	Current (mA)
		V- V+	Voltage (mV/V)
J	Ethernet port (x2) RJ45 connector		Ethernet connections
K	Top harness connectors (x3) 8-pin, 3-pin, and 2-pin	see A to E	Various connections

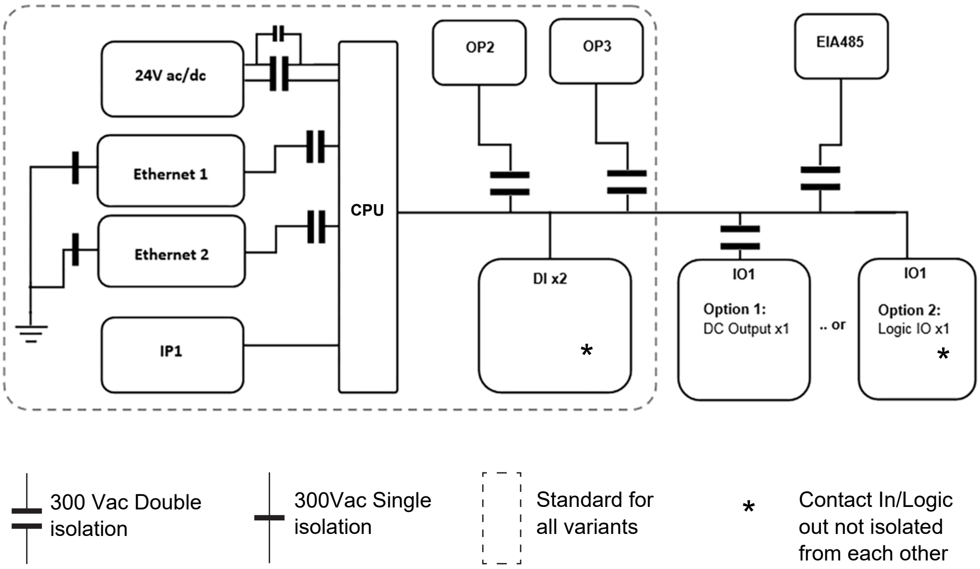
EPC2000 Programmable Controller Indicator Layout



Key	Title	Function
1	Power LED	Illuminates green when power is applied to the controller.
2	Diagnostic LED	Illuminates red if the controller detects that the firmware may not be valid or has been tampered with. Contact your local support. Otherwise it is OFF.
3	Standby LED	Illuminates amber if the controller is not running and is instead in standby mode. Refer to "Standby" on page 62 for information. This LED flashes amber if the controller starts up and is in Manual mode and thus not controlling a process. Refer to "Start-up Modes" on page 62.
4	Run/Config LED	Illuminates solid green when the controller is running. Illuminates flashing green when the controller is in configuration mode.
5	Comms LED	Flashes green when there is any Ethernet or serial communication activity targeted at the controller, otherwise it is OFF.
6	IO1 (Input/Output 1) LED	Illuminates green when the IO1 (if configured as a logic or DC output) is driven.
7	OP2 (Output 2) LED	Illuminates green when the Form A (normally open) relay OP2 is energized.
8	OP3 (Output 3) LED	Illuminates green when the Form C (changeover) relay OP3 is energized.
9	Ethernet Port 1 Network Speed LED	Illuminates green when a 100Mbps connection is established. It does not illuminate when a 10Mbps connection is established.
10	Ethernet Port 1 Network Activity LED	Illuminates amber when Ethernet link established; blinks when activity is detected.
11	Ethernet Port 2 Network Speed LED	Illuminates green when a 100Mbps connection is established. It does not illuminate when a 10Mbps connection is established.
12	Ethernet Port 2 Network Activity LED	Illuminates amber when Ethernet link established; blinks when activity is detected.
13	Function button (Ethernet initialization)	Enables Bonjour Auto Discovery or resets the IP configuration, depending on when the button is pressed. Refer to "Ethernet Configuration" on page 227 for further details.

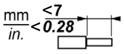
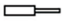
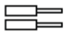
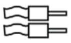
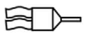
Isolation Boundaries

The drawing shows double and basic isolation boundaries.



Wire Sizes

The table below shows the wire sizes for different methods of terminating cables at the EPC2000 Programmable Controller. Although solid and multi-strand cable can be used in the terminals, it is recommended to use a metal ferrule where possible. Do not insert more than two cables into any single terminal connection.

	MAXIMUM exposed conductor length 7mm (0.28")	**Solid cable**	**Multi-strand cable**	**Multi-strand cable with ferrule**	**Multi-strand cable with ferrule and skirt**	**2x Solid cables**	**2x Multi-strand cables**	**2x Multi-strand cables with 2x ferrules**	**2x Multi-strand cables with twin ferrules**
									
mm^2	0.25 – 2.5	0.20 – 2.5		0.25 – 2.5		2x 0.20 – 1.0	2x 0.20 – 1.5	2x 0.25 – 1	0.5 – 1.5
AWG	24 – 13	24 – 14		23 – 13		2x 24 – 17	2x 24 – 16	2x 23 – 17	20 – 16

All terminal screws should be tightened to a torque of between 0.5 and 0.6Nm (4.4 and 5.3lb inch).

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Tighten terminal screws in conformance with the torque specifications.

A maximum of two wires, identical in type and cross sectional size can be inserted per terminal of a harness connector.

Ensure all wires that are connected to the controller terminals without a ferrule, do not exceed the maximum exposed cable conductor length of 7mm (0.28").

When using ferrules (cable ends) ensure the correct size is selected and each is securely fixed using a crimping tool.

Failure to follow these instructions will result in death or serious injury.

Fuse Protection

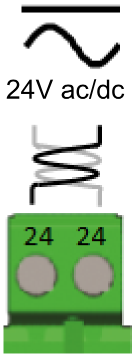
External fuse protection must be provided to the power supply input to the EPC2000 Programmable Controller.

Recommended external fuse rating is as follows:

For 24V ac/dc, fuse type: T rated 2A 250V.

For wiring, use only copper cables.

Low Voltage Power Supply



- 24Vac, −15%, +10% at 42-62Hz.
- 24Vdc, −15%, +20% ± 5% ripple voltage.
- The polarity is not important.
- Power rating: 6W.

⚠ DANGER

FIRE HAZARD

Do not connect the controller directly to line voltage.

Use only isolating PELV or SELV power supplies to supply power to the equipment.

Failure to follow these instructions will result in death or serious injury.

Input 1 Sensor Measuring Analog Input (IP1)

This input is available in all models.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Digital Inputs (DI) and IO1 terminals are not isolated from IP1 Sensor Measuring Input. If IP1 is not at ground or a safe potential then Digital Inputs and IO1 will be at the same potential and care needs to be taken in component ratings and instruction to personnel to ensure safety.

Failure to follow these instructions will result in death or serious injury.

⚠ CAUTION

POTENTIAL INJURY OR EQUIPMENT DAMAGE

Do not run input wires together with power cables.

When shielded cable is used, it must be grounded at one point only.

Failure to follow these instructions can result in injury or equipment damage

NOTICE

MEASUREMENT INACCURACIES

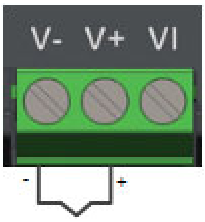
There are several factors that can potentially cause measurement inaccuracies.

Failure to follow these instructions can result in equipment damage.

To mitigate these factors:

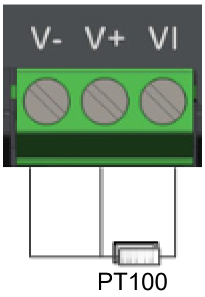
- Do not run input wires together with power cables.
- When shielded cable is used, it must be grounded at one point only.
- Any external components (such as zener barriers, etc) connected between sensor and input terminals may cause incorrect measurement due to excessive and/or un-balanced line resistance or possible leakage currents.
- The sensor input is not isolated from the logic outputs & digital inputs.
- Pay attention to line resistance; a high line resistance may cause measurement inaccuracies.
- Do not connect a single sensor to more than one instrument. Sensor break operation could be severely compromised.

Thermocouple Input



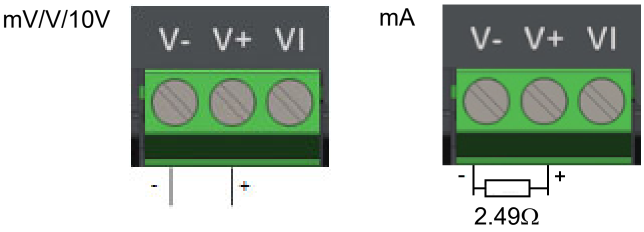
- Use the correct compensating cable (preferably shielded) to extend thermocouple cabling, ensure that polarity is strictly followed throughout and that thermal junctions are avoided in any intermediate connections.

RTD Input



- The resistance of the three wires must be the same. The line resistance may cause measurement inaccuracies if it is greater than 22Ω.

Linear Input (mA, mV or V)



- If shielded cable is used it must only be grounded at one end.
- For a mA input connect the 2.49Ω burden resistor (R) supplied between the + and − input terminals as shown. The resistor supplied is 1% accuracy, 50ppm.

Input/Output 1 (IO1)

IO1 is available as standard. It may be ordered as:

- Option 1 - Analog Output.
- Option 2 - Logic (SSR drive) output OR Contact Input (connected to Logic output).

The function of the I/O is pre-configured by selection of an application, either as part of the order code or through iTools. The function can subsequently be changed through iTools ("IO.IO1" on page 114).

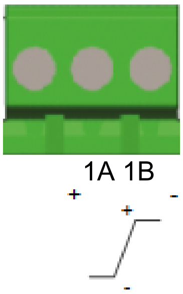
⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Digital Inputs (DI) and IO1 terminals are not isolated from IP1 Sensor Measuring Input. If IP1 is not at ground or a safe potential then Digital Inputs and IO1 will be at the same potential and care needs to be taken in component ratings and instruction to personnel to ensure safety.

Failure to follow these instructions will result in death or serious injury.

Analog Output

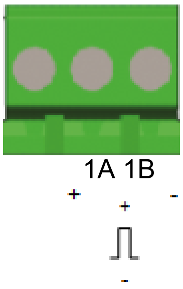


- Output isolated 300Vac
- Software configurable: 0–10Vdc, 0–20mA or 4–20mA.
- Max load resistance: Voltage >450Ω; Current <550Ω
- Calibration accuracy: % of reading + Offset

  Voltage better than ±(0.5% + 50mV)
  Current better than ±(0.5% + 100µA)
- May also be configured as an isolated contact input

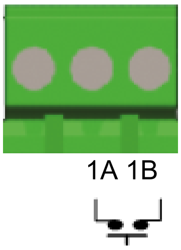
  Open state >365Ω
  Closed state <135Ω

Logic (SSR drive) Output



- Not isolated from the sensor input, the current transformer input or the digital inputs
- Output ON state: 12Vdc at 44mA max
- Output OFF state: <300mV, <100µA
- The output switching rate must be set to help prevent damage to the output device in use. See "Cycle Time and Minimum OnTime Algorithms" on page 120.

Contact Input - connected to Logic output



- Not isolated from the sensor input or logic outputs
- Switching: 12Vdc at 44mA max
- Contact open > 500Ω. Contact closed < 150Ω

Output 2 (OP2) - Form A, Normally Open Relay

⚠ ⚠ DANGER

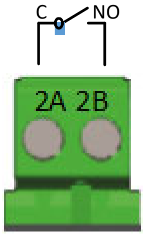
HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to product and all I/O circuitry (alarms, control I/O, etc.) before starting the installation, removal, wiring, maintenance or inspection of the product.

Failure to follow these instructions will result in death or serious injury.

Output 2 is available in all models. It is a Form A (normally-open) Relay.



- Isolated output 300Vac CAT II
- Contact rating: 2A 230Vac +15% resistive
- Contact rating minimum: 100mA 12V
- The output switching rate must be set to help prevent damage to the output device in use. See "Cycle Time and Minimum OnTime Algorithms" on page 120.

Output 3 (OP3) - Form C, Changeover Relay

⚠ ⚠ DANGER

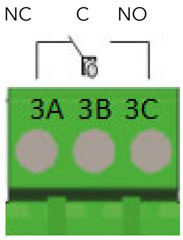
HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to product and all I/O circuitry (alarms, control I/O, etc.) before starting the installation, removal, wiring, maintenance or inspection of the product.

Failure to follow these instructions will result in death or serious injury.

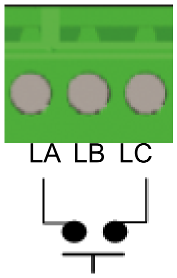
Output 3 is available in all models. It is a Form C (changeover) Relay.



- Isolated output 300Vac CAT II
- Contact rating: 2A 230Vac +15% resistive
- The output switching rate must be set to help prevent damage to the output device in use. See "Cycle Time and Minimum OnTime Algorithms" on page 120.

Digital Input (DI1)

Digital Input 1 is available in all models. It is a contact input.



- Contact open >400Ω
- Contact closed <100Ω
- Not isolated from sensor input.

⚠ ⚠ DANGER

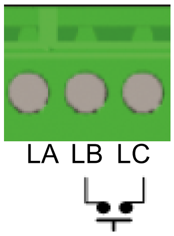
HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Digital Inputs (DI) and IO1 terminals are not isolated from IP1 Sensor Measuring Input. If IP1 is not at ground or a safe potential then Digital Inputs and IO1 will be at the same potential and care needs to be taken in component ratings and instruction to personnel to ensure safety.

Failure to follow these instructions will result in death or serious injury.

Digital Input (DI2)

Digital Input 2 is available in all models. It is a contact input.



- Contact open >400Ω
- Contact closed <100Ω
- Not isolated from sensor input.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Digital Inputs (DI) and IO1 terminals are not isolated from IP1 Sensor Measuring Input. If IP1 is not at ground or a safe potential then Digital Inputs and IO1 will be at the same potential and care needs to be taken in component ratings and instruction to personnel to ensure safety.

Failure to follow these instructions will result in death or serious injury.

General Information About Relays and Inductive Loads

High voltage transients may occur when switching inductive loads such as contactors or solenoid valves. Through the internal contacts, these transients may introduce disturbances which could affect the performance of the controller.

The relays in the EPC2000 Programmable Controller are fitted with a varistor which reduces the need to use snubbers when switching inductive loads up to 0.5A.

Digital Communications Connections

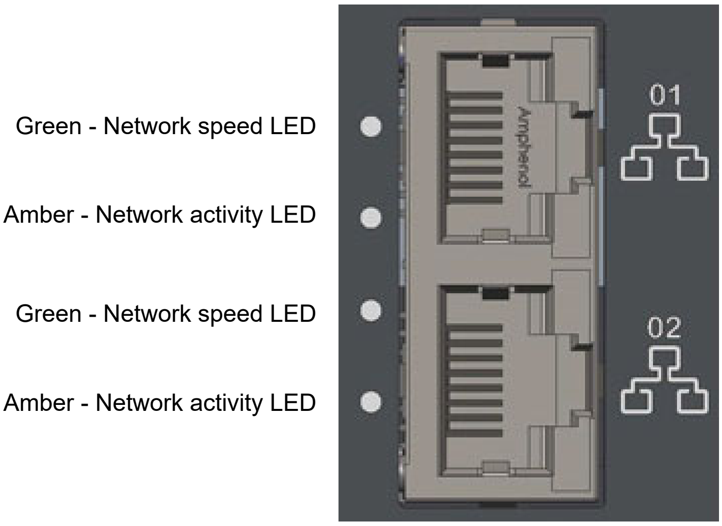
On the EPC2000 Programmable Controller, Ethernet (Modbus TCP) is supplied as standard. Serial communications (EIA-485) is available as an option. Modbus RTU protocol is used for compatibility with existing controllers.

When a shielded cable is used for serial communications (EIA-485), connect the cable shield only to the COM (HD) input of the controller. The recommended maximum cable length is 1500m (4921.26ft) at 19200 baud.

The digital communication ports are isolated to 300Vac CAT II.

Ethernet Wiring

An Ethernet networking capability is provided by two RJ45 connectors, fitted to the front panel.


Green - Network speed LED
Amber - Network activity LED
Green - Network speed LED
Amber - Network activity LED
Amphenol
01
02

Each connector has a pair of LED indicators.

- Green (network speed indication). On = 100Mbps link; off = 10Mbps link (or no link)
- Amber (link activity). On = link established; blinking = Ethernet activity

The connection is 10/100BASE-T, autosensing.

Serial Communication (EIA-485)

The EIA-485 Modbus RTU function of an EPC2000 Programmable Controller provides an alternative digital communications method to Ethernet. It is independent of Ethernet and can be used at the same time that Ethernet communications are active. Data transmission is slower than Ethernet, but it is an effective communications method in some situations.

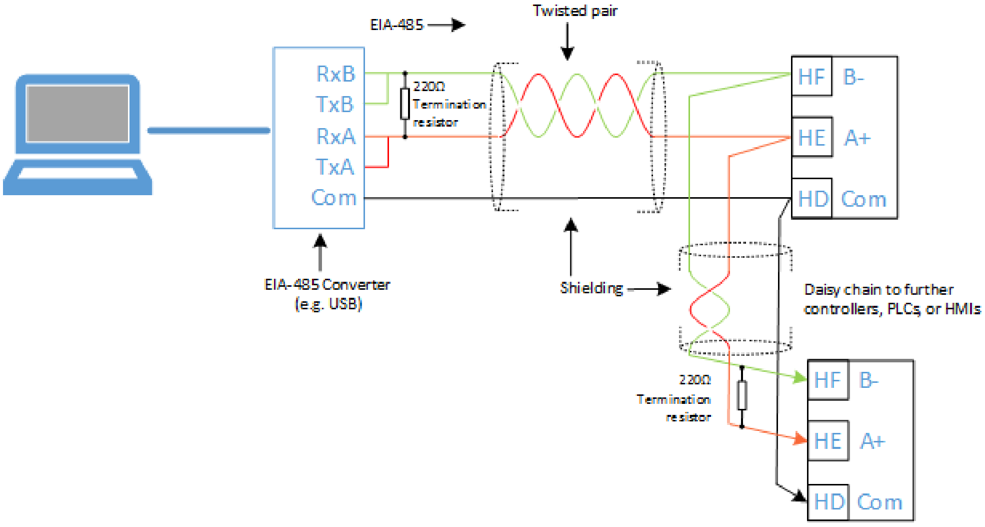
It may be used in the following example contexts:

1. Connection to legacy EIA-485 Automation networks for SCADA or data acquisition.
2. Direct connection to Programmable Logic Controllers using a serial network.
3. To connect to a low cost panel HMI that does not have an Ethernet connection.
4. To interconnect an EPC2000 Programmable Controller, for example to use the broadcast master function to send a digital master setpoint profile to downstream slave devices.
5. To connect Eurotherm iTools, typically in situations where older types of instruments such as Series 3000 are being replaced and EIA-485 infrastructure is already present. Ethernet will typically be a better connection method for new installations.

When connecting a computer to EIA-485, a USB adaptor is typically used. It is good practice to use electrically isolated adaptors, since EMI (electromagnetic interference) may otherwise be transmitted to the computer, damaging it.

EIA-485 supports up to 32 devices per network segment. Segment repeaters may be used to increase the number of devices in an EIA-485 network. Note that 220Ω termination resistors are required at the start and end of the EIA-485 line. Without these, communications will suffer intermittent faults.

Connections using a suitable converter are shown in the following diagram.



Start Up

What's in this Chapter?

This chapter describes:

- Initial setup.
- Commissioning.
- What to expect when the controller is first switched on from new out of the box.
- Power up after the instrument has been configured or commissioned.

Initial Setup

The following stages describe and provide assistance for when you come to start the EPC2000 Programmable Controller for the first time:

- "Installation".
- "Initial Start up (Power On)".
- "Network and iTools connection".
- "Control application and Configuration".

The information contained within this subject "Initial Setup" assumes the following to provide clarity. The product is being installed (mounted and wired) straight out of the box, for information regarding Use, Mounting location, Instructions and temperature/humidity requirements see "Location" on page 31.

Installation

The EPC2000 Programmable Controller should be installed in accordance with the information in HA033209 Installation sheet, which is supplied with the product.

See also:

- "Installation" on page 27.
- "Location" and "General mounting instructions" on page 31.
- "Dimensions" on page 29.
- "EPC2000 Programmable Controller Terminal Layout" on page 35.

Initial Start up (Power On)

After completing installation, it should be possible to ‘power on’ the EPC2000 Programmable Controller for the first time.

The initial start up refers to the EPC2000 Programmable Controller being switched on for the first time, meaning the product has not been in operation before and therefore requires Configuration (parameter and hardware) and more importantly the final stage Commission.

The EPC2000 Programmable Controller will start and go into Standby mode, which is sufficient for the next stage, see "Network and iTools connection" on page 48.

See also:

- "When First Switched On" on page 60.
- "Start-up Modes" on page 62.
- "Fuse Protection" on page 38.

Network and iTools connection

A network connection is required to:

- communicate with the EPC2000 Programmable Controller.
- add a control application and configure the application parameters using iTools.
- configure hardware options (i.e. IO1).
- enable the controller to be part of a larger control system.

The EPC2000 Programmable Controller can communicate over a network using either of the following methods:

- Ethernet network.
- Serial comms (EIA-485 communications).

There are multiple methods of creating a network connection for the EPC2000 Programmable Controller, choose from the following options:

- "Ethernet Initialization, using the Function button" on page 49.
- "Ethernet Connection, using iTools Control Panel and Scan feature" on page 52.
- "Serial Communications, EIA-485 setup" on page 56.
- "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266

Ethernet Initialization, using the Function button

Scan QR Code for EPC2000 'How To' video tutorials.

Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

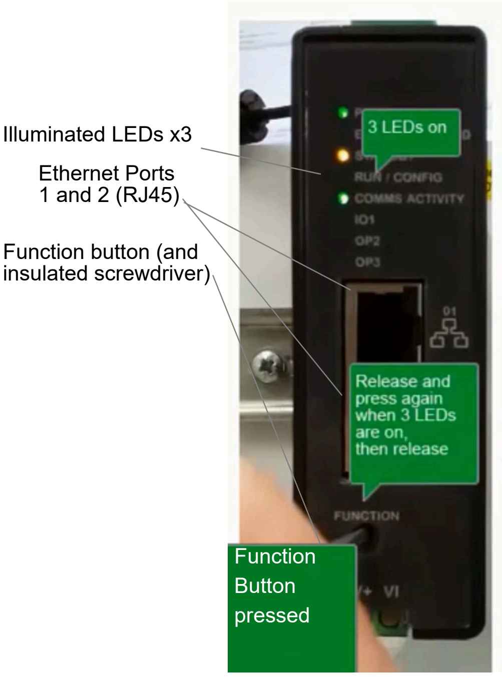
Switching on AutoDiscovery

1. If on, power-off the EPC2000 Programmable Controller and wait for all LEDs to extinguish.
2. Insert a small, suitable insulated tool into the Function button slot to press the recessed button.

> **⚠ WARNING**
>
> **HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH**
>
> Ensure only a suitable insulated tool that fits the aperture is used to depress the function button when required.
>
> **Failure to follow these instructions can result in death, serious injury or equipment damage.**

3. Keep pressing the Function button whilst restoring power to the EPC2000 Programmable Controller. Carefully watch the front-panel LEDs as the timing is important.
4. Once the EPC2000 Programmable Controller's power is restored, all the front panel LEDs illuminate before extinguishing again, as part of a power-up self test.
5. When only three LEDs illuminate (Power, Standby and Comms Activity), quickly release the Function button, before briefly pressing and releasing once again.

   The EPC2000 Programmable Controllers AutoDiscovery function will now be enabled (switched on), allowing iTools to find the device when on the same network.
6. Make sure the EPC2000 Programmable Controller is connected to the Ethernet network it will operate on, using an appropriate Ethernet network cable connected to one of the EPC2000 Programmable Controller Ethernet ports (1 or 2) with an RJ45 connection.

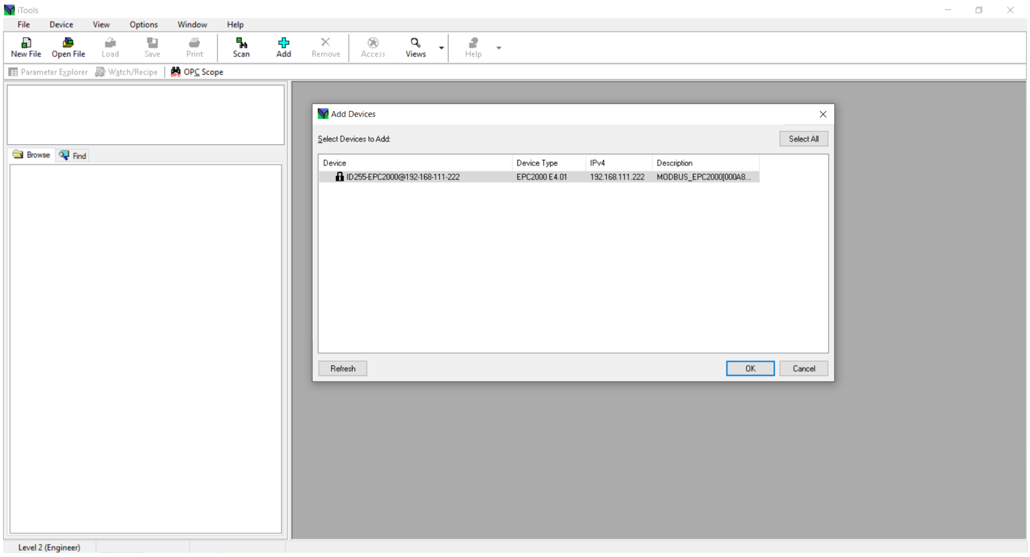

Illuminated LEDs x3
Ethernet Ports 1 and 2 (RJ45)
Function button (and insulated screwdriver)
3 LEDs on
RUN / CONFIG
COMMS ACTIVITY
IO1
OP2
OP3
01
Release and press again when 3 LEDs are on, then release
FUNCTION
Function Button pressed
V+ VI

Note: Make sure that the controller and PC which is running iTools are on the same subnet.

7. Open iTools, Eurotherm's software suite for configuring Programmable Controllers, see "What is iTools?" on page 65 for further details.

8. In iTools select 'Add' Add from the iTools menu bar, the *Add Devices* panel will appear and in the list of devices connected via Ethernet will be the EPC2000 Programmable Controller.

Note: If there is no comms configuration password set for the instrument, the device will have a padlock icon next to it.

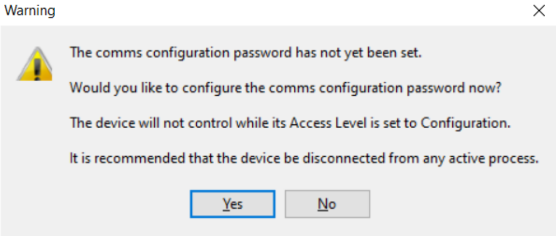

iTools
File Device View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Parameter Explorer Watch/Recipe OPC Scope
Browse Find
Refresh OK Cancel
Level 2 (Engineer)

Add Devices

Select Devices to Add: Select All

Device	Device Type	IPv4	Description
🔒 ID255-EPC2000@192-168-111-222	EPC2000 E4.01	192.168.111.222	MODBUS_EPC2000[000A8...

9. Select discovered controller and click OK. iTools will open a dialog requesting the user to set a comms configuration password.



Warning

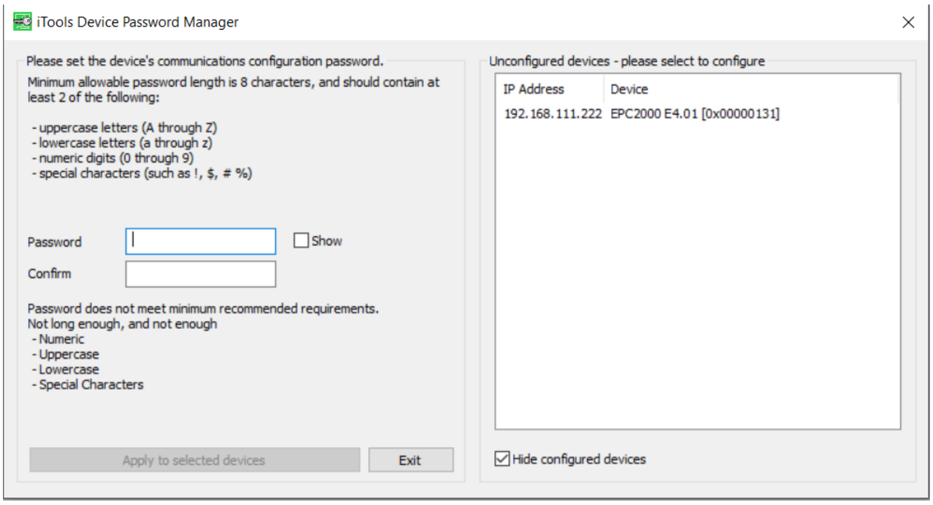
The comms configuration password has not yet been set.

Would you like to configure the comms configuration password now?

The device will not control while its Access Level is set to Configuration.

It is recommended that the device be disconnected from any active process.

10. Select "Yes" to set comms config password and iTools Device Password Manager window will open. Follow instructions to set the comms configuration password.



iTools Device Password Manager

Please set the device's communications configuration password.

Minimum allowable password length is 8 characters, and should contain at least 2 of the following:

- uppercase letters (A through Z)
- lowercase letters (a through z)
- numeric digits (0 through 9)
- special characters (such as !, $, # %)

Password □ Show

Confirm

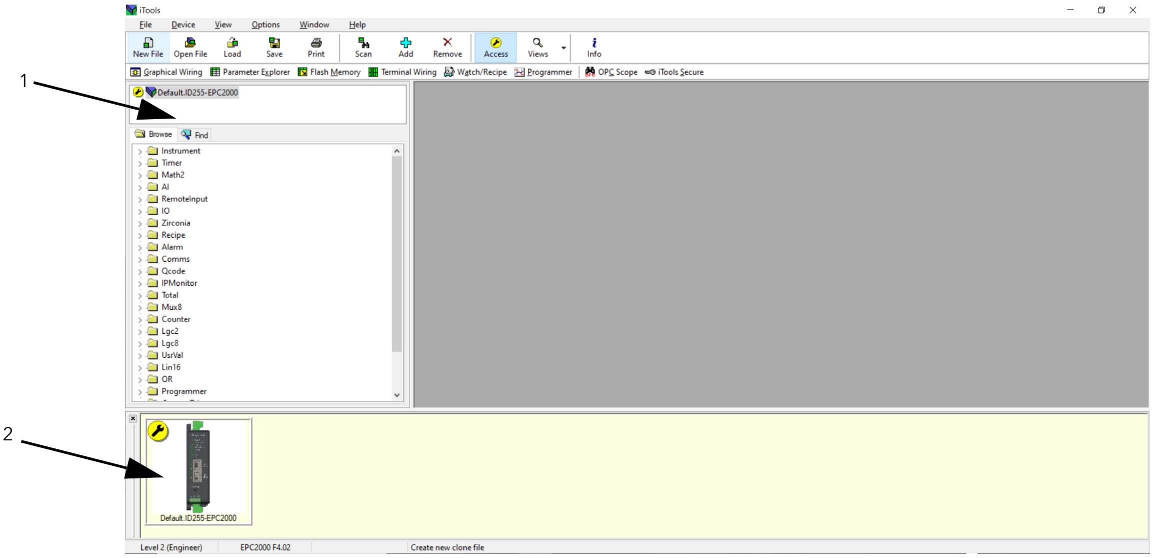
Password does not meet minimum recommended requirements.
Not long enough, and not enough
- Numeric
- Uppercase
- Lowercase
- Special Characters

Unconfigured devices - please select to configure

IP Address	Device
192.168.111.222	EPC2000 E4.01 [0x00000131]

☑ Hide configured devices

11. Once the password has been set, the EPC2000 Programmable Controller connects with the instrument in configuration mode and the following items appear:
    - the device name and number, in top left window (1).
    - an image in view panel window (2).



Note: To maintain Cybersecurity best practice, it is recommended that AutoDiscovery is disabled when not required i.e.after Initial set up deactivate the AutoDiscovery function, for further details, see Auto Discovery parameter in "Comms.Serial.Network and Comms.Ethernet.Network" on page 129.

Ethernet Connection, using iTools Control Panel and Scan feature

For security reasons, however, it may be advisable to keep AutoDiscovery off. In this case, if AutoDiscovery and DHCP is not used, iTools must be set up for Ethernet. This is described in the instructions which follow. iTools configuration package, version V9.79 or later, may be used to configure Ethernet communications.

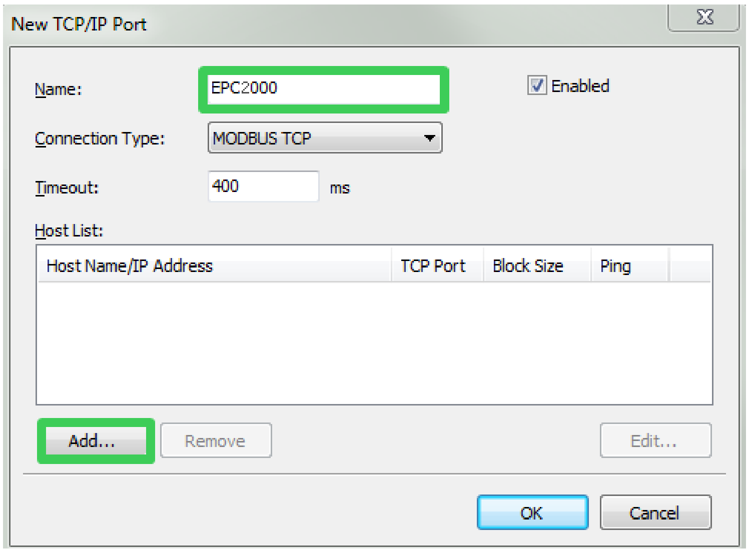
Scan QR Code for EPC2000 ‘How To’ video tutorials.

Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

Adding a Device to iTools Control panel

To include a Host Name/Address within the iTools scan:

1. Ensure iTools is NOT running before taking the following steps.
2. Within Windows, open the 'Control Panels'. If the Control Panels open in ‘Category View’ select Large or Small Icons instead.
3. Double-click on ‘iTools’ to open the iTools control panel, the iTools configuration panel will appear.
4. Within the iTools configuration settings, select the 'TCP/IP' tab.
5. Click the 'Add' button to add a new connection, the New TCP/IP Port panel will appear.
6. Type in a name of your choice, for example “EPC2000 Programmable Controller” and click Add. (Ensure that no duplicate IP address entries are enabled at the same time).



New TCP/IP Port

Name: EPC2000

☑ Enabled

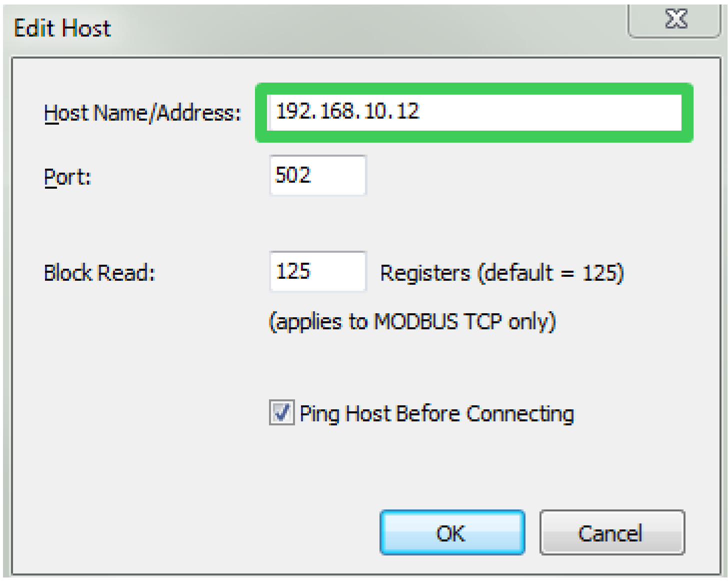
Connection Type: MODBUS TCP

Timeout: 400 ms

Host List:

Host Name/IP Address	TCP Port	Block Size	Ping

7. The Edit Host panel appears, enter the IP address of the device.
   Ensure the PC IP address running iTools is in the same range as the EPC2000 Programmable Controller, then click OK.



Edit Host

Host Name/Address: 192.168.10.12

Port: 502

Block Read: 125 Registers (default = 125)

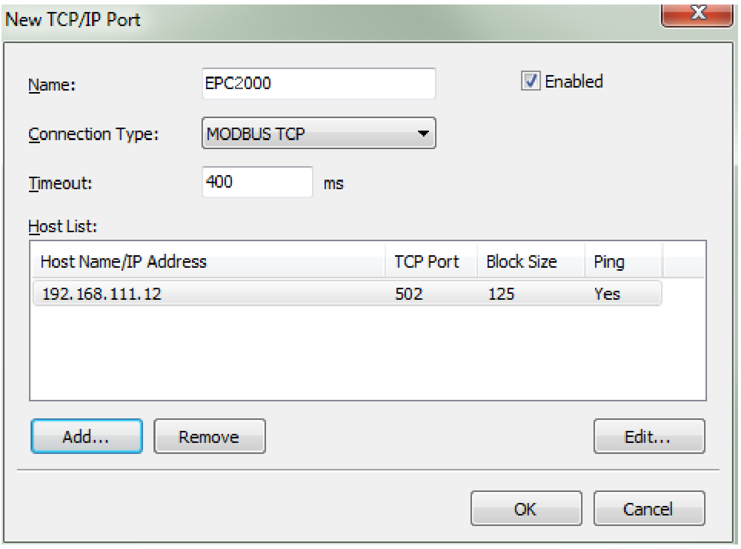
(applies to MODBUS TCP only)

☑ Ping Host Before Connecting

Note: The EPC2000 Programmable Controller can have varying default IP addresses depending on the controllers version (i.e.comms protocol), for further information see "Default IP address, details and Password" on page 57.

Note: The IP address for controllers with the communication protocol PROFINET will require the IP address to be configured during commissioning, see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266 for further information.

8. The New TCP/IP Port panel appears, confirm the IP address is correct, then click OK to commit the new TCP/IP port details into the iTools Control panel.



New TCP/IP Port

Name: EPC2000 ☑ Enabled

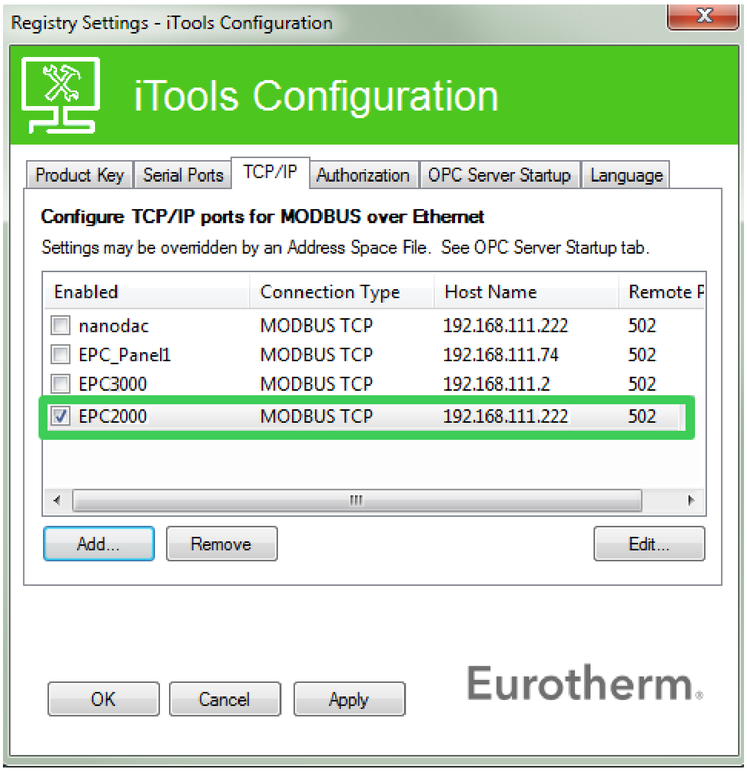
Connection Type: MODBUS TCP

Timeout: 400 ms

Host List:

Host Name/IP Address	TCP Port	Block Size	Ping
192.168.111.12	502	125	Yes

9. The iTools Control panel appears displaying the new TCP/IP port you have just added, select OK to add the new entry.



Registry Settings - iTools Configuration

iTools Configuration

Product Key | Serial Ports | TCP/IP | Authorization | OPC Server Startup | Language

Configure TCP/IP ports for MODBUS over Ethernet

Settings may be overridden by an Address Space File. See OPC Server Startup tab.

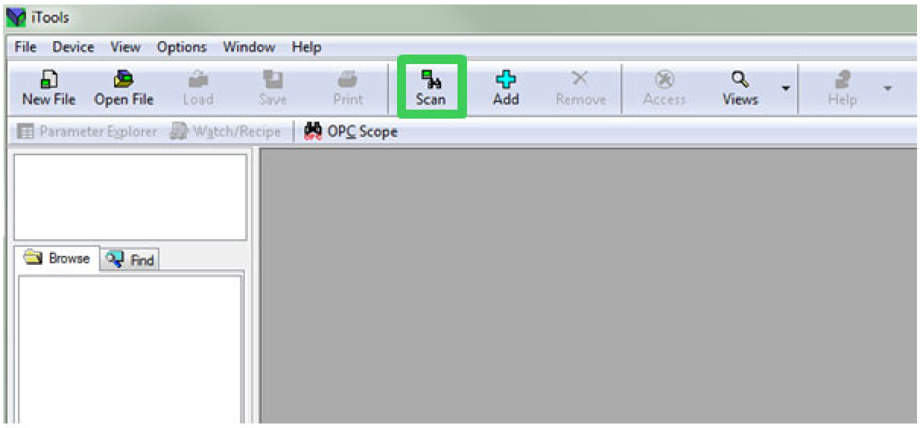
Enabled	Connection Type	Host Name	Remote P
□ nanodac	MODBUS TCP	192.168.111.222	502
□ EPC_Panel1	MODBUS TCP	192.168.111.74	502
□ EPC3000	MODBUS TCP	192.168.111.2	502
☑ EPC2000	MODBUS TCP	192.168.111.222	502

iTools is now ready to communicate with the EPC2000 Programmable Controller at the Host Name/IP Address configured.

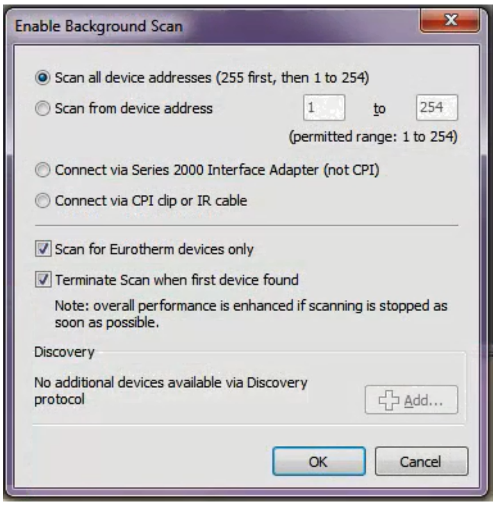
Also included in "Connecting to EPC2000 using iTools" in Chapter "Digital Communications".

iTools: Scan and Connecting to a device

10. Open iTools and click ‘Scan’.



The Enable Background Scan panel appears.



Enable Background Scan

● Scan all device addresses (255 first, then 1 to 254)

○ Scan from device address 1 to 254

(permitted range: 1 to 254)

○ Connect via Series 2000 Interface Adapter (not CPI)

○ Connect via CPI clip or IR cable

☑ Scan for Eurotherm devices only

☑ Terminate Scan when first device found

Note: overall performance is enhanced if scanning is stopped as soon as possible.

Discovery

No additional devices available via Discovery protocol

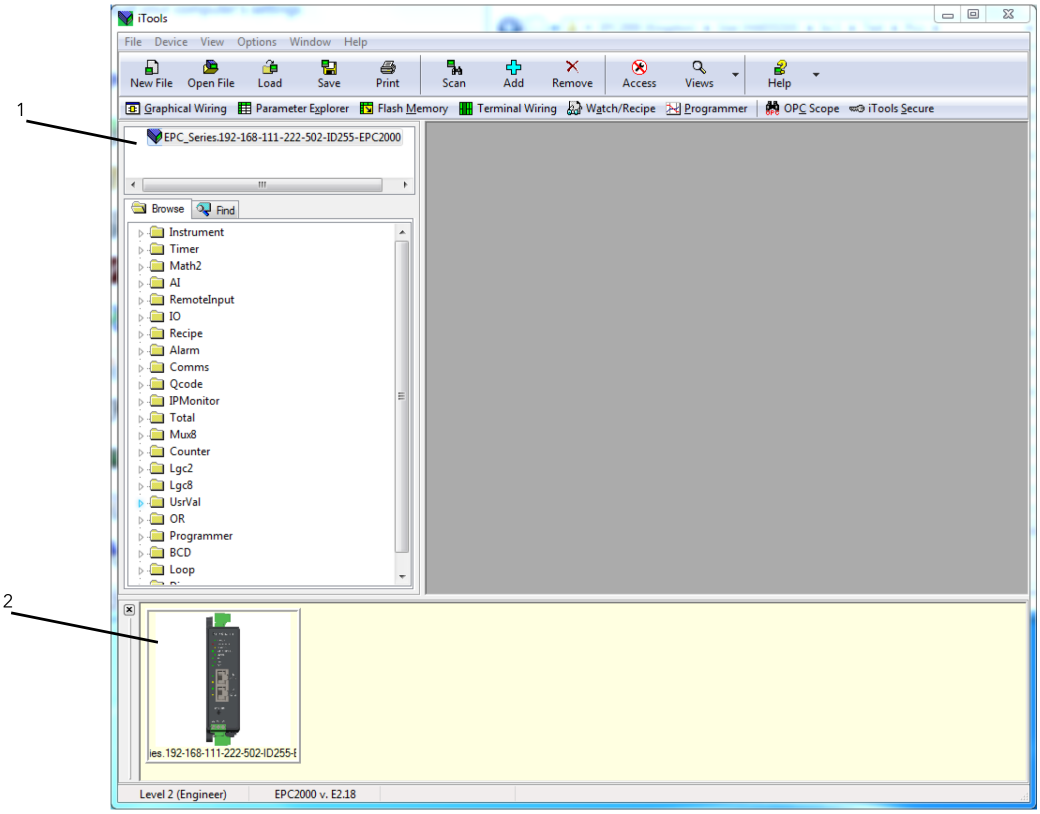
11. If not selected, select Scan all device addresses (255 first, then 1 to 254) option on the Enable background Scan panel, then check the following checkbox options:
    - Scan for Eurotherm devices only.
    - Terminate Scan when first device found.
12. Select OK on the Enable Background Scan panel to start the iTools scan.

    The scan will only find devices if they have been added to the iTools Control Panel. (And if they are in the same range as the IP address of the PC), see "Adding a Device to iTools Control panel" on page 52, for further details.

The EPC2000 Programmable Controller connects and the following items appear in the iTools windows:

- the device name and number, in top left window (1)
- an image in view panel window (2).

If the comms configuration password is not set, iTools will request the user to set the password first via Device Password Manager before iTools connects in configuration mode.


1
2
iTools
File Device View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
EPC_Series.192-168-111-222-502-ID255-EPC2000
Browse Find
Instrument
Timer
Math2
AI
RemoteInput
IO
Recipe
Alarm
Comms
Qcode
IPMonitor
Total
Mux8
Counter
Lgc2
Lgc8
UsrVal
OR
Programmer
BCD
Loop
ies.192-168-111-222-502-ID255-E
Level 2 (Engineer)
EPC2000 v. E2.18

Serial Communications, EIA-485 setup

The Serial communication EIA-485 uses Modbus RTU, which is a payable option on the EPC2000 Programmable Controller and provides an alternative digital communications method to Ethernet. It is independent of Ethernet and can be used at the same time that Ethernet communications are active.

For further information see:

- "Digital Communications Connections" on page 45.
- Description and set up details, see "Serial Communication (EIA-485)" on page 46.
- Modbus RTU and the associated parameters, "Modbus RTU" on page 226.

Additional Network Setup Information/Tasks

Default IP address, details and Password

The default values are listed below for the EPC2000 Programmable Controller:

Setting/Variant	Value
IP Address	
PROFINET:	0.0.0.0
Other:	192.168.111.222
Subnet mask	
PROFINET:	0.0.0.0
Other:	255.255.255.0
Gateway	
All:	0.0.0.0
Comms Configuration password:	No default (set via iTools).

Note: The IP address for controllers with the communication protocol PROFINET will require the IP address to be configured during commissioning, see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266 for further information. The default IP Address and associated details are listed above.

Device IP address and Configuration Password - reset

It is possible to reset the EPC2000 Programmable Controllers' IP address, clear comms configuration password and disable Autodiscovery using the Function button.

For further information see "Reset the Controller’s IP address" in Chapter "Digital Communications".

Control application and Configuration

Once the programmable controller network communications are established - both network and iTools communications, see "Network and iTools connection" on page 48 - it is possible to continue completing initial set up by moving on to the programmable controller application and configuration.

The following items listed require setup, this is only a brief outline listing the fundamentals;

- Control application - create or load (unless pre-configured).
- Configure the following:
  - Controller hardware options (IO1 Analog or Logic Input/Output.
  - Sensor input (IP1 Sensor Measuring analog, Thermocouple/ mA or Ohms input).
  - Controller parameters.
  - Alarm types and their associated Setpoints.
  - Programmer - initial setup (Holdback, holdback type).

For further information see:

- "Controller Concept" on page 26.
- "Quick Start Tables" on page 60.
- "Configuration mode" on page 90.
- "Cloning" and "To Clone a New Controller" on page 88.
- "Quick Start Tables" on page 60 and "Qcode" on page 136.
- "Types of Control" on page 199.

Note: To access the configuration mode the Comms Configuration password will be requested before any configuration access is granted.

See "Default IP address, details and Password" on page 57 for details.

Commissioning

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Electrical equipment must be installed, operated and maintained by only qualified personnel.

Turn off all power to product and all I/O circuitry (alarms, control I/O etc.) before starting the installation, removal, wiring, maintenance or inspection of the product.

Do not use, or implement a controller configuration (control strategy) into service without ensuring the configuration has completed all operational tests, been commissioned and approved for service. It is the responsibility of the person commissioning the controller to ensure the configuration is correct.

Failure to follow these instructions will result in death or serious injury.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Digital Inputs (DI) and IO1 terminals are not isolated from IP1 Sensor Measuring Input. If IP1 is not at ground or a safe potential then Digital Inputs and IO1 will be at the same potential and care needs to be taken in component ratings and instruction to personnel to ensure safety.

Failure to follow these instructions will result in death or serious injury.

⚠ DANGER

FIRE HAZARD

Do not allow anything to fall through the case apertures and ingress the controller. Electrical equipment must be installed, operated and maintained by only qualified personnel.

Failure to follow these instructions will result in death or serious injury.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise must be allowed to program, install, alter and commission this product.

Do not use this product for critical control or protection applications where human or equipment safety relies on the operation of the control circuit.

During commissioning ensure all operating states and potential fault conditions are carefully tested.

It is the responsibility of the person commissioning the controller to ensure the configuration is correct.

Failure to follow these instructions can result in death, serious injury or equipment damage.

When First Switched On

The EPC2000 Programmable Controller is designed to be application based. This chapter describes the different ways in which the controller can be ordered and delivered and how this affects the operation at power up.

1. Controller new 'out of the box' supplied unconfigured.
2. Controller new 'out of the box' supplied pre-configured according to the order code. "Start Up - Pre-configured Programmable Controller" on page 61.
3. Subsequent start ups - Controller previously configured. Go to section "Subsequent Start-ups" on page 62.

In all cases the controller display will perform a diagnostic in which every LED is illuminated. The controller will identify the type of hardware fitted. If a different hardware is detected the instrument will go into standby mode. To clear this condition, change the expected I/O parameter value to match the fitted IO parameter value.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter and commission this product.

It is the responsibility of the person commissioning the controller to ensure the configuration is correct.

Failure to follow these instructions can result in death, serious injury or equipment damage.

More detailed features available in the product may be configured using iTools as explained in "Configuration Using iTools" on page 64. iTools is a configuration package available free of charge from Eurotherm by accessing www.eurotherm.com.

Setpoint

The Setpoint is defined as the value which the process is required to attain. The value of the setpoint may be obtained from a number of sources, for example, via the programmer function block, via an external analog source, via digital communications. The working setpoint is, therefore, defined as the current setpoint derived from any of these sources.

Quick Start Tables

Two Quick Start function blocks can be used to define the application for the EPC2000 Programmable Controller. A third block initiates the controller based on the parameters defined in the first two blocks. For further information on using these function blocks, refer to "Qcode" on page 136.

The 1st character in SET 1 will select an application which automatically configures relevant function block parameters and creates wires between function blocks to make a complete control strategy relevant to that application. Application ‘1’, Heat only controller and Application ‘2’ heat/cool controller are generally covered by this manual. Entering a value of "X" in a field will disable that feature, if appropriate.

Quick Code SET 1

Application	Analog Input 1 Type	Analog Input 1 Range
X = None	X = Use Default	X = Use Default
1 = PID Heat Only Control	**Thermocouple**	1 = 1-100°C
2 = PID Heat/Cool Control	B = Type B	2 = 1-200°C
	J = Type J	3 = 1-400°C
	K = Type K	4 = 1-600°C
	L = Type L	5 = 1-800°C
	N = Type N	6 = 1-1000°C
	R = Type R	7 = 1-1200°C
	S = Type S	8 = 1-1300°C
	T = Type T	9 = 1-1600°C
	RTD	A = 1-1800°C
	P = Pt100	F = Full range
	Linear	
	M = 0-80mV	
	V = 0-10V	
	2 = 0-20mA	
	4 = 4-20mA	

Note: If no application is selected (1st character in SET 1 = X) the controller will exit configuration and will take on a set of default values. Any further configuration may be carried out through iTools configuration software ("Configuration Using iTools" on page 64).

Quick Code SET 2

LA Function	LB Function	Temperature Units
X = Not used	X = Not fitted or not used	X = Use Default
W = Alarm Acknowledge	W = Alarm Acknowledge	C = Celsius
M = Auto/Manual	M = Auto/Manual	F = Fahrenheit
R = Program Run/Hold	R = Program Run/Hold	K = Kelvin
P = Setpoint Select	P = Setpoint Select	
T = Program Reset	T = Program Reset	
U = Remote/Local Select	U = Remote/Local Select	
V = Recipe Load Select	V = Recipe Load Select	
K = Loop Track	K = Loop Track	

Start Up - Pre-configured Programmable Controller

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise must be allowed to program, install, alter and commission this product.

It is the responsibility of the person commissioning the controller to ensure the configuration is correct.

Failure to follow these instructions can result in death, serious injury or equipment damage.

If a product has been ordered with an application it will be pre-configured with a basic wiring for a control loop but must be connected to iTools to set up the application.

The programmable controller should be connected to iTools for commissioning and any further configuration.

For further details see "Initial Setup" on page 47 and "Configuration" on page 90.

Subsequent Start-ups

When the controller is no longer new and has been in normal use it will start up in Operator level. If, however, it was powered down when in Configuration level it will power up in 'Standby' with the Standby LED illuminated. To clear this condition, re-enter configuration level (with password - see "To Access" in Chapter "Configuration"), then, either continue with changes to the configuration or accept existing changes by exiting configuration level. The reason for this is that the controller may have been part configured prior to shut down and either needs completion of the configuration or confirmation that no further changes are required.

Start-up Modes

The controller may start in manual or in automatic mode depending on the setting of the 'Recovery Mode' parameter, see section "Loop.Configuration" on page 163.

If the Recovery Mode has been set to Manual (default) the controller will start in Manual mode.

The Standby LED will flash, showing the controller is in Manual Mode. Initially the output will be at the 'Fallback Value', see section "Loop.Main" on page 161.

If the Recovery Mode has been set to 'Last' the controller will start in either in Manual mode or Auto mode depending on the mode it was in prior to being powered off. If the controller is in Auto mode and not in Standby, the Standby LED will be OFF.

For further information on start up modes see section "Start-up and Recovery" on page 214.

Standby

The Standby LED will be illuminated when the controller is in Standby mode. Standby is the term given when the instrument strategy is not controlling due to the following reasons:

- If the controller starts up and the Recovery Mode parameter is set to 'Manual' (refer to "Start-up Modes" above).
- If the controller has detected an unexpected condition (for example, was powered down whilst in configuration mode, or fitted hardware does not match expected hardware). See the table below for more information on unexpected conditions that will place the instrument into standby.
- If the controller is forced into standby via the Instrument.Diagnostics.ForceStandby parameter.

Use iTools to examine the Instrument.Diagnostics.StandbyCondStatus parameter to determine the cause, as follows:

Bit Number	Decimal Value	Description
0	1	Invalid RAM image of NVOL
1	2	NVOL parameter database load/store was unsuccessful
2	4	NVOL region load/store was unsuccessful
3	8	Option NVOL load/store was unsuccessful
4	16	Factory Calibration not detected
5	32	Unexpected CPU condition
6	64	Hardware Ident Unknown
7	128	Fitted hardware differs from expected hardware
8	256	Unexpected Keyboard condition during startup
9	512	Controller was powered down whilst in config mode
10	1024	Unsuccessful recipe load
11	2048	Not used
12	4096	Not used
13	8192	Not used
14	16384	Not used

Note: NVOL - Non-volatile

When the instrument is in standby, the following occurs:

- All outputs are placed into their 'Off' state, unless being used as Valve Raise (Up) / Lower (Down) in which case the Standby Action is configurable (Rest, Up, Down).
- The control loop will be placed into Hold.
- If an alarm has the Standby Inhibit parameter set to On, then the alarm is inhibited (active alarms will be turned Off and new alarm conditions will not be acted upon).
- If the instrument is placed into configuration mode, the running setpoint program will be reset.

⚠ WARNING

LOSS OF COMMUNICATIONS

If the output is not wired, but written to by communications, it will continue to be controlled by the communications messages. In this case take care to allow for the loss of communications.

Failure to follow these instructions can result in serious injury, or equipment damage

Configuration Using iTools

What's in this Chapter?

This chapter describes how to configure the controller using iTools.

This chapter describes those features which are specific to the EPC2000 Programmable Controller. iTools is generally described in the iTools Help Manual Part No. HA028838 which can be obtained from www.eurotherm.com.

This chapter includes a description of the OEM (Original Equipment Manufacturer) security features and their configuration.

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise must be allowed to program, install, alter and commission this product.

It is the responsibility of the person commissioning the controller to ensure the configuration is correct.

Failure to follow these instructions can result in death, serious injury or equipment damage.

What is iTools?

iTools is a configuration and monitoring package which can be used to edit, store and 'clone' complete controller configurations. It is a free downloadable package available from www.eurotherm.com.

iTools can be used to configure all the functions of the controller already described in this manual. It is also possible using iTools to configure additional functions such as Recipe Storage and to download the configuration into an instrument. These features are described in this chapter.

What is an IDM?

The Instrument Descriptor Module (IDM) is a Windows file that is used by iTools to determine the properties of a specific device. Each version of a device requires its own IDM file. This is normally included with the iTools software and allows iTools to recognize the software version of your instrument.

Connecting a PC to the Controller

This may been done using either of the two Ethernet ports or the optional Serial Comms (EIA-485 only).

Using the Ethernet (Modbus TCP) Ports

Connect the controller to the PC using a standard Ethernet patch cable with RJ45 connectors. If you know the IP address of the controller, you can set up iTools with this known address (refer to "Connecting to EPC2000 using iTools" on page 233).

If you do not know the IP address of the controller, see "Default IP address, details and Password" on page 57 and for some versions you could use the Auto Discovery feature (refer to "Bonjour" on page 229).

Using the Communications Port

Connect the controller to the EIA-485 serial communications port of the PC shown in "Serial Communication (EIA-485)" on page 46.

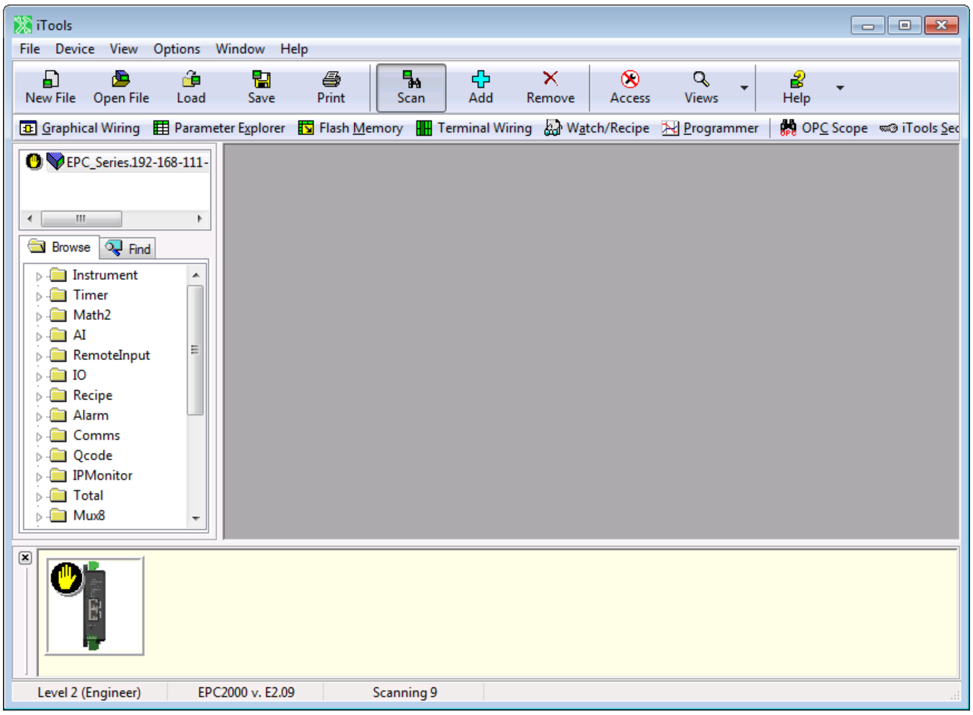
Starting iTools

Scan QR Code for EPC2000 ‘How To’ video tutorials.

Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

Open iTools and, with the controller connected, press ’Scan’ on the iTools menu bar. iTools will search the communications ports and Ethernet connections for recognizable instruments. If you are using Auto Discovery to connect to the controller, refer to "Auto Discovery" on page 229.

When the controller is detected a screen view similar to the one shown below will be displayed. The browser on the left shows the List Headers. To display parameters within a list double click the Header or select 'Parameter Explorer'. Click on a list header to display parameters associated with this list.



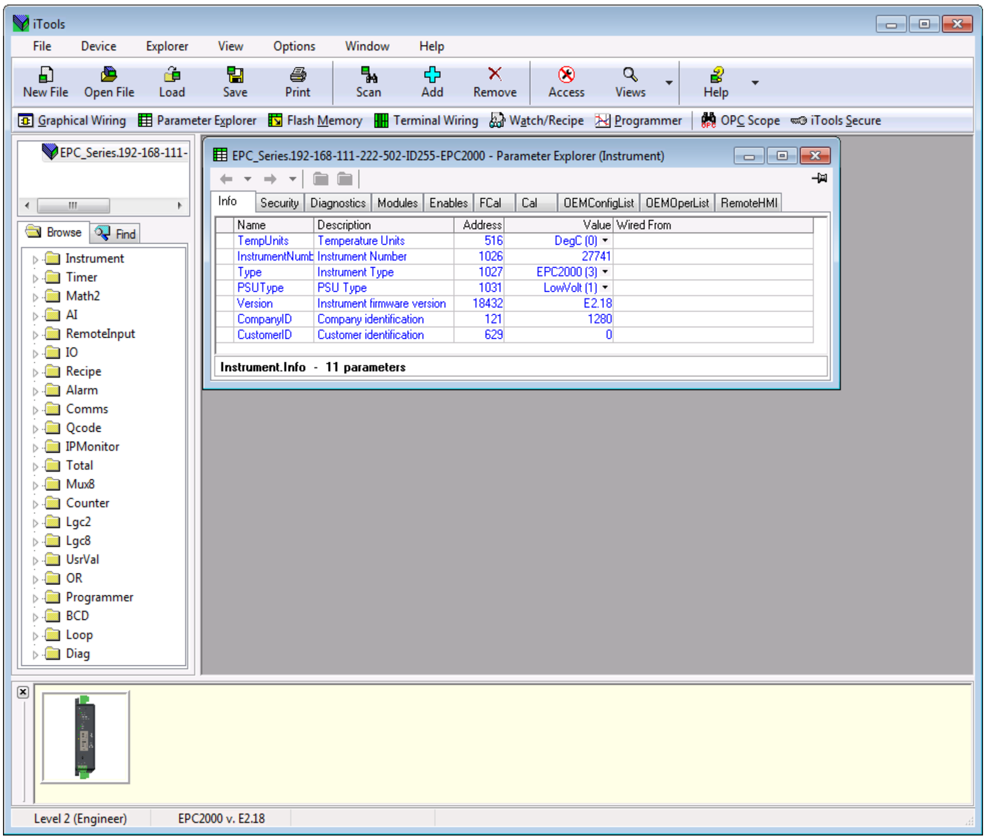
The controller may be configured using the Browser view above. The following pages show a number of examples of how to configure various functions.

It is assumed that the user is generally familiar with iTools and has a general understanding of Windows.

If the controller uses Ethernet comms, iTools needs to be set up to communicate with the controller. This is described in "Ethernet Configuration" on page 227.

The 'Browser' List

All instrument parameters are available in the Browser List. Double-click on a heading to display parameters associated with the selected heating in the right hand side of the iTools view.



Name	Description	Address	Value	Wired From
TempUnits	Temperature Units	516	DegC (0)	
InstrumentNumb	Instrument Number	1026	27741	
Type	Instrument Type	1027	EPC2000 (3)	
PSUType	PSU Type	1031	LowVolt (1)	
Version	Instrument firmware version	18432	E2.18	
CompanyID	Company identification	121	1280	
CustomerID	Customer identification	629	0	

Instrument.Info - 11 parameters

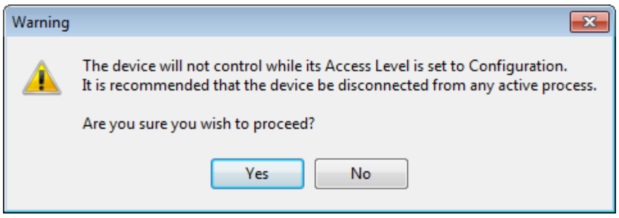
Parameters colored blue are read-only in the selected operator level.

Parameters shown in black may be modified within pre-set limits. Enumerated parameters are selected from a drop-down list and analog parameters may be modified by typing in the new value.

Configuration Access

The controller may be configured over comms using either Ethernet or Serial communications (if ordered). In order to avoid multiple users from writing to the same configuration parameter at the same time, comms connections are separated into sessions: 1x Modbus RTU (serial), 3x Modbus TCP (Ethernet), and 1x Modbus TCP (Ethernet) reserved for a preferred master. When a session is created, it restricts access to another session also being in Configuration mode simultaneously.

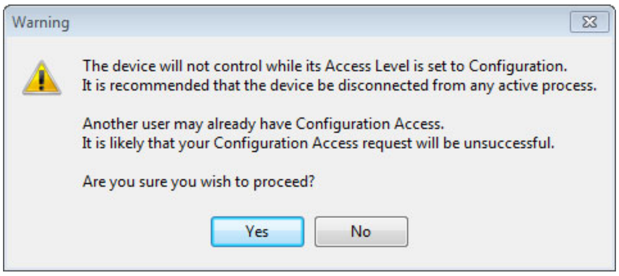
To put the controller into Configuration level, click on Access. A dialog message is displayed as shown.



The device will not control while its Access Level is set to Configuration.
It is recommended that the device be disconnected from any active process.

Are you sure you wish to proceed?

If another session already has the controller in Configuration Mode, a different dialog message is displayed, advising that the request to enter Configuration Mode from this session may not be successful.



The device will not control while its Access Level is set to Configuration.
It is recommended that the device be disconnected from any active process.

Another user may already have Configuration Access.
It is likely that your Configuration Access request will be unsuccessful.

Are you sure you wish to proceed?

⚠ WARNING

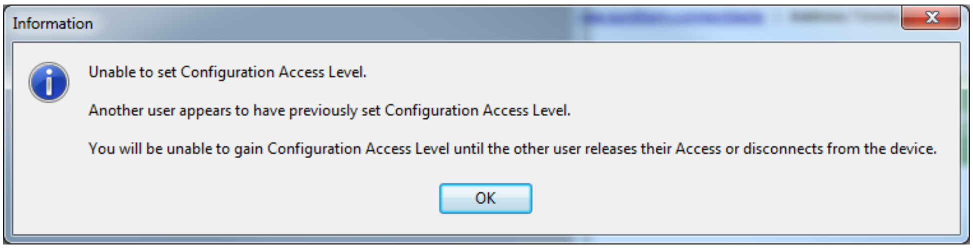
UNINTENDED EQUIPMENT OPERATION

The programmable controller must not be configured while it is connected to a live process as entering Configuration Mode pauses all outputs. The controller remains in Standby until Configuration Mode is exited.

Failure to follow these instructions can result in death, serious injury or equipment damage.

Select ‘Yes’ if the process is not online.

You will be prompted to enter the Comms Config password. If another session already has the controller in Configuration Mode, the following dialog is shown informing you that Configuration Mode is not possible at this time.



Unable to set Configuration Access Level.

Another user appears to have previously set Configuration Access Level.

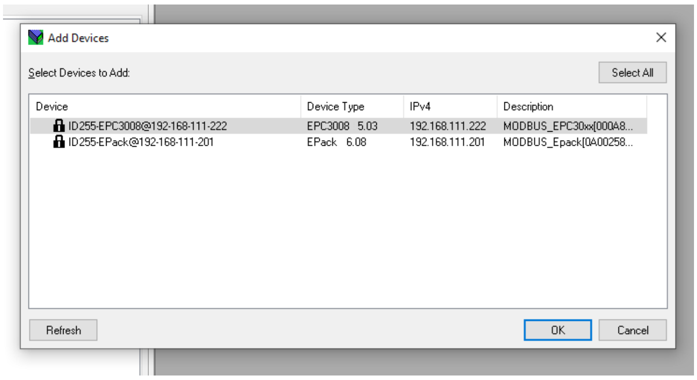
You will be unable to gain Configuration Access Level until the other user releases their Access or disconnects from the device.

The controller may be now configured using iTools.

Setting Comms Configuration Password

For EPC2000 V4.xx and later, the password will require configuration on first connection to iTools via Serial/Ethernet comms. This can be done in two ways:

1. The user selects the “Add Devices” (using the “Add” button), the device will display a padlock icon to the left of the device description.

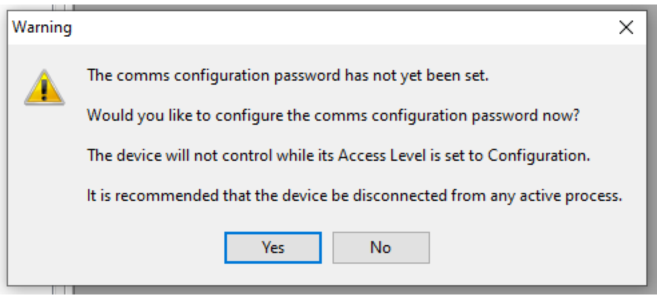


Add Devices

Select Devices to Add:

Device	Device Type	IPv4	Description
🔒 ID255-EPC3008@192-168-111-222	EPC3008 5.03	192.168.111.222	MODBUS_EPC30xx[000A8...
🔒 ID255-EPack@192-168-111-201	EPack 6.08	192.168.111.201	MODBUS_Epack[0A00258...

If they select to add the device, iTools will display a message informing them that they need to configure the Comms Configuration password before being able to proceed.



Warning

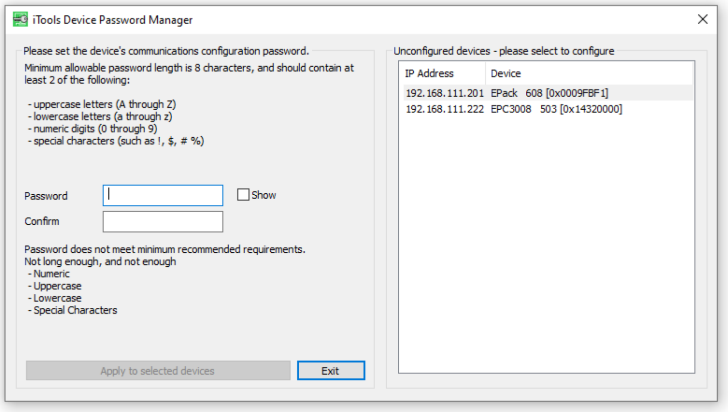
The comms configuration password has not yet been set.

Would you like to configure the comms configuration password now?

The device will not control while its Access Level is set to Configuration.

It is recommended that the device be disconnected from any active process.

Selecting ‘Yes’ will automatically launch the iDevPWSetup tool.



iTools Device Password Manager

Please set the device's communications configuration password.

Minimum allowable password length is 8 characters, and should contain at least 2 of the following:

- uppercase letters (A through Z)
- lowercase letters (a through z)
- numeric digits (0 through 9)
- special characters (such as !, $, # %)

Password

Confirm

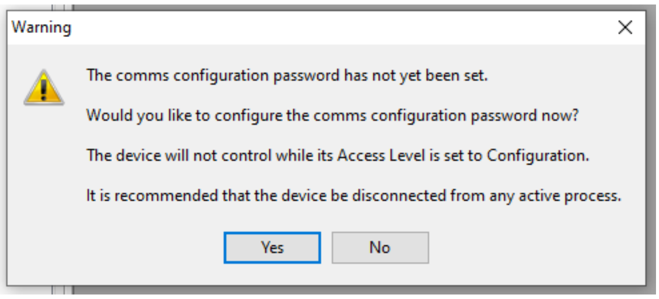
□ Show

Password does not meet minimum recommended requirements.
Not long enough, and not enough
- Numeric
- Uppercase
- Lowercase
- Special Characters

Unconfigured devices - please select to configure

IP Address	Device
192.168.111.201	EPack 608 [0x0009FBF1]
192.168.111.222	EPC3008 503 [0x14320000]

2. The user chooses to perform a network “Scan”, on detecting the device, iTools will display a message:



Warning

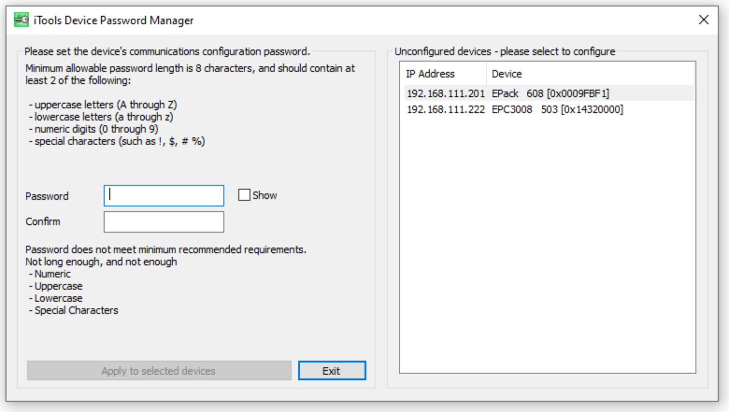
The comms configuration password has not yet been set.

Would you like to configure the comms configuration password now?

The device will not control while its Access Level is set to Configuration.

It is recommended that the device be disconnected from any active process.

Informing the user that they need to configure the Comms Configuration password before being able to proceed. Selecting ‘Yes’ will automatically launch the iDevPWSetup tool.



iTools Device Password Manager

Please set the device's communications configuration password.

Minimum allowable password length is 8 characters, and should contain at least 2 of the following:

- uppercase letters (A through Z)
- lowercase letters (a through z)
- numeric digits (0 through 9)
- special characters (such as !, $, # %)

Password

□ Show

Confirm

Password does not meet minimum recommended requirements.
Not long enough, and not enough
- Numeric
- Uppercase
- Lowercase
- Special Characters

Unconfigured devices - please select to configure

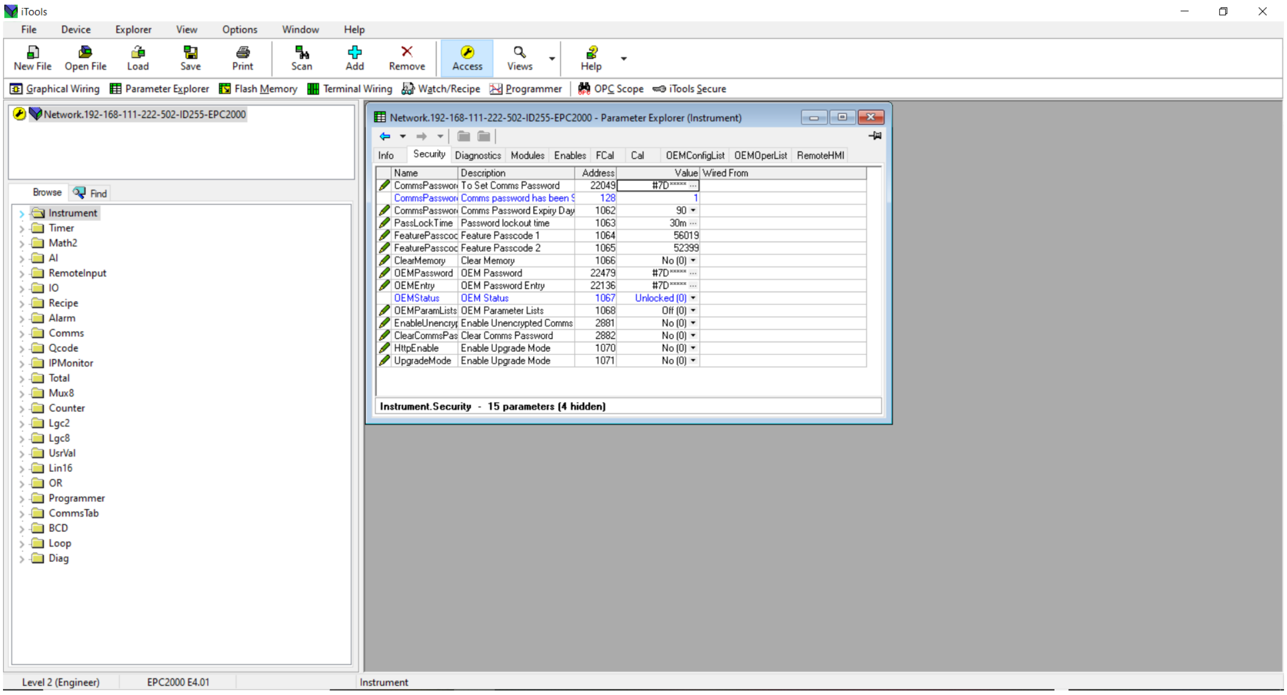
IP Address	Device
192.168.111.201	EPack 608 [0x0009FBF1]
192.168.111.222	EPC3008 503 [0x14320000]

Note the following:

- Passwords are required to be at least eight characters in length
- It is required that the password uses two out of the four following character types:
  - UPPER CASE [A-Z]
  - lower case [a-z]
  - numeric [0-9]
  - special characters [! $ # %, etc]
- Enforce a lockout period after five unsuccessful attempts

Instrument List

The Instrument List is the first list shown in the Browser section of iTools. It allows further features to be set. In particular Security related features including the Comms Config password.



Network.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

Name	Description	Address	Value	Wired From
CommsPasswor	To Set Comms Password	22049	#7D*****	
CommsPasswor	Comms password has been S	128	1	
CommsPasswor	Comms Password Expiry Day	1062	90	
PassLockTime	Password lockout time	1063	30m	
FeaturePasscoc	Feature Passcode 1	1064	56019	
FeaturePasscoc	Feature Passcode 2	1065	52399	
ClearMemory	Clear Memory	1066	No (0)	
OEMPassword	OEM Password	22479	#7D*****	
OEMEntry	OEM Password Entry	22136	#7D*****	
OEMStatus	OEM Status	1067	Unlocked (0)	
OEMParamLists	OEM Parameter Lists	1068	Off (0)	
EnableUnencryp	Enable Unencrypted Comms	2881	No (0)	
ClearCommsPas	Clear Comms Password	2882	No (0)	
HttpEnable	Enable Upgrade Mode	1070	No (0)	
UpgradeMode	Enable Upgrade Mode	1071	No (0)	

Instrument.Security - 15 parameters (4 hidden)

To change the configuration password, click on the ellipsis in the value cell on the password dialog window and enter a new one. The maximum password length is 96 bytes (UTF-8 encoded). The number of characters is therefore dependent upon the character set used. For example:

- For ASCII characters (single byte per character), the limit is 96 characters.
- For Cyrillic (two bytes per character), the limit is 48 characters.
- For Chinese (three bytes per character), the limit is 32 characters.

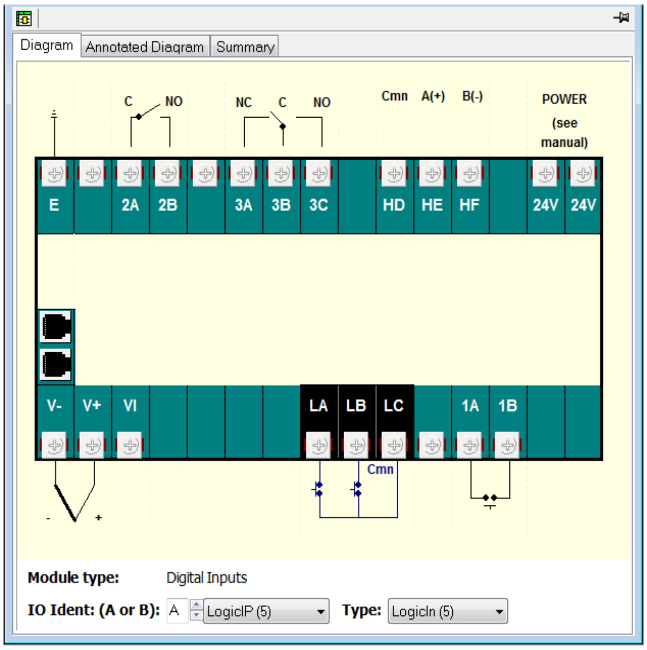
iTools enforces a minimum password length of 8 characters and should contain at least two of the following:

  - uppercase letters (A through Z)
  - lowercase letters (a through z)
  - numeric digits (0 through 9)
  - special characters (such as !, $, #, %)

The parameter ‘Comms Password Expiry Days’ defaults to 90 days. This parameter sets the number of days after which the configuration password will expire. Refer to "Comms Configuration Level Password" on page 19 for details.

Terminal Wiring Editor

Select ‘Terminal Wiring’ on the main toolbar.



Module type: Digital Inputs

IO Ident: (A or B): A LogicIP (5) Type: LogicIn (5)

From this view click on a set of terminals representing an IO module. From the ‘IO Ident’ drop-down, select an IO type. The diagram of the type of IO will be shown against the chosen terminal set.

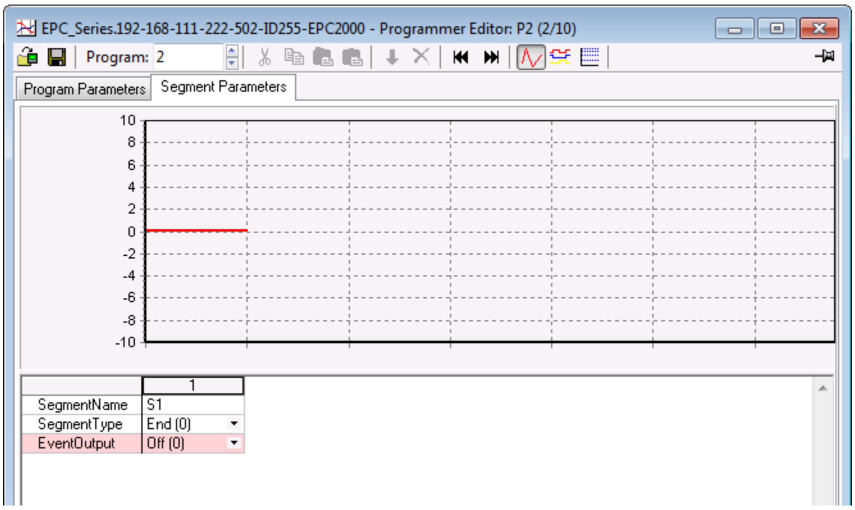
An Annotated Diagram and Summary of wiring can also be viewed.

Programmer

Programs may be configured, run, held or reset in the controller using iTools.

To Set up a Program Using iTools

Press ‘Programmer’ in the menu bar.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Programmer Editor: P2 (2/10)
Program: 2
Program Parameters
Segment Parameters
10
8
6
4
2
0
-2
-4
-6
-8
-10

	1
SegmentName	S1
SegmentType	End (0)
EventOutput	Off (0)

By default, a program will consist of a single End segment as shown above.

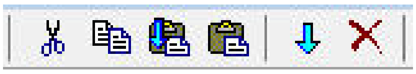
To add segments, change the SegmentType of the End segment to the desired segment type using the SegmentType drop-down menu. A new segment of the required type will be inserted and the End segment will be shifted to the right. Note that changes to the program will be automatically written to the controller.

The EPC2000 Programmable Controller can support up to 20 stored programs; the actual number of programs and segments is dependent upon a software option selected by feature security. The Programmer options are:

- disabled.
- 1 x 8 Basic Programmer (1 program of 8 configurable segments).
- 1 x 24 Advanced Programmer (1 program of 24 configurable segments with up to 8 event outputs).
- 10 x 24 Advanced Programmer (10 programs of 24 configurable segments with up to 8 event outputs).
- 20 x 8 Advanced Programmer (20 programs of 8 configurable segments with up to 8 event outputs).
- For all Programmer options, an additional End segment is provided which may include event outputs if it is an Advanced Programmer.

Programs are identifiable by a program number (1-10, for example). Each program can also be given a program name of up to 20 UTF-8 characters.

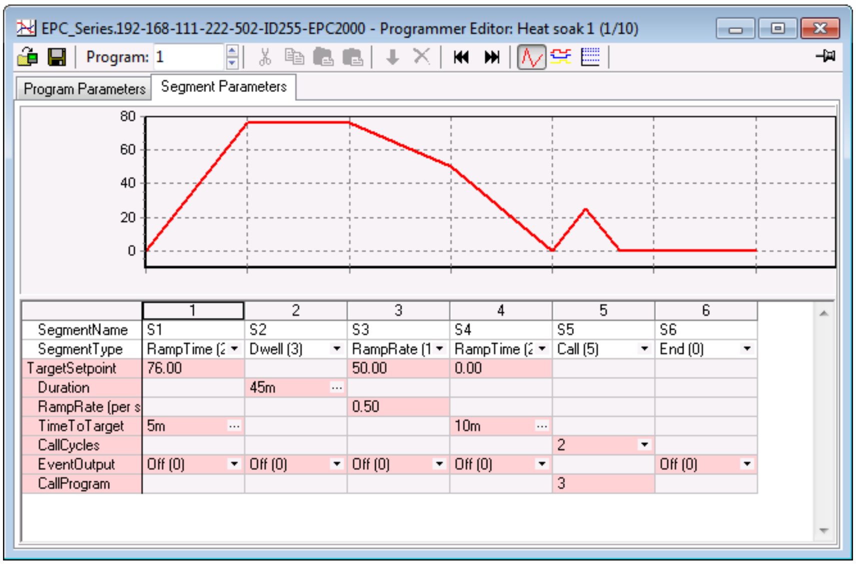
Menu options are shown in the toolbar above the graph and are also available as a context menu by right-clicking in the segment table. They are from left to right:



Select a segment by clicking into the top of the list (segment number or name). Multiple segments may be selected.

- Cut (Ctrl-X):
  Removes the selected segment(s) and copies to the clipboard.
- Copy (Ctrl-C):
  Copies the selected segment(s) to the clipboard.
- Paste (Ctrl-V):
  Pastes segments from the clipboard, inserting them to the right of the selected segment(s).
- Paste Over:
  Replaces the selected segment(s) with segments from the clipboard.
- Insert:
  Inserts a new segment to the right of the selected segment.
- Delete:
  Removes the selected segment(s).

The following diagram shows a program (Program 1) of 5 segments plus an End segment. Segment 5 calls another program (in this case program 3 consisting of a ramp up and a ramp down) to run twice before the program ends. Segment types are described in the Programmer chapter, "Segments" on page 190.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Programmer Editor: Heat soak 1 (1/10)
Program: 1
Program Parameters
Segment Parameters
80
60
40
20
0

	1	2	3	4	5	6
SegmentName	S1	S2	S3	S4	S5	S6
SegmentType	RampTime (2	Dwell (3)	RampRate (1	RampTime (2	Call (5)	End (0)
TargetSetpoint	76.00		50.00	0.00		
Duration		45m				
RampRate (per s			0.50			
TimeToTarget	5m			10m		
CallCycles					2	
EventOutput	Off (0)	Off (0)	Off (0)	Off (0)		Off (0)
CallProgram					3	

⚠ CAUTION

CALL SEGMENTS

If a call segment is selected the controller will default to calling the next program number. This may not necessarily be the correct program so you must ensure that the correct call program number is selected manually.

Failure to follow these instructions can result in injury or equipment damage.

Event Outputs

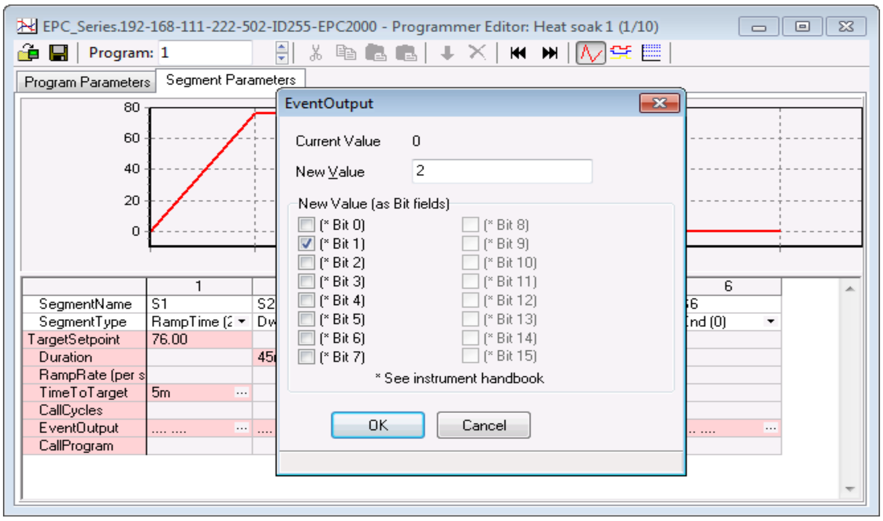
Up to eight event outputs may be enabled using Programmer.Setup.MaxEvents parameter in the iTools browser.

If more than one event is configured, ‘EventOutput’ is shown as an ellipsis, see the diagram below.

If no events are configured then ‘EventOutput’ is not shown in the list.

If one event is configured, ‘EventOutput’ allows the event to be turned on or off directly.

Click on the ellipsis to display a bit map:



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Programmer Editor: Heat soak 1 (1/10)

Program: 1

Program Parameters | Segment Parameters

	1		6
SegmentName	S1	S2	6
SegmentType	RampTime (2	Dw	nd (0)
TargetSetpoint	76.00		
Duration		45	
RampRate (per s			
TimeToTarget	5m ...		
CallCycles			
EventOutput	....	....	.. ..
CallProgram			

EventOutput

Current Value 0

New Value 2

New Value (as Bit fields)

☐ (* Bit 0) ☐ (* Bit 8)
☑ (* Bit 1) ☐ (* Bit 9)
☐ (* Bit 2) ☐ (* Bit 10)
☐ (* Bit 3) ☐ (* Bit 11)
☐ (* Bit 4) ☐ (* Bit 12)
☐ (* Bit 5) ☐ (* Bit 13)
☐ (* Bit 6) ☐ (* Bit 14)
☐ (* Bit 7) ☐ (* Bit 15)

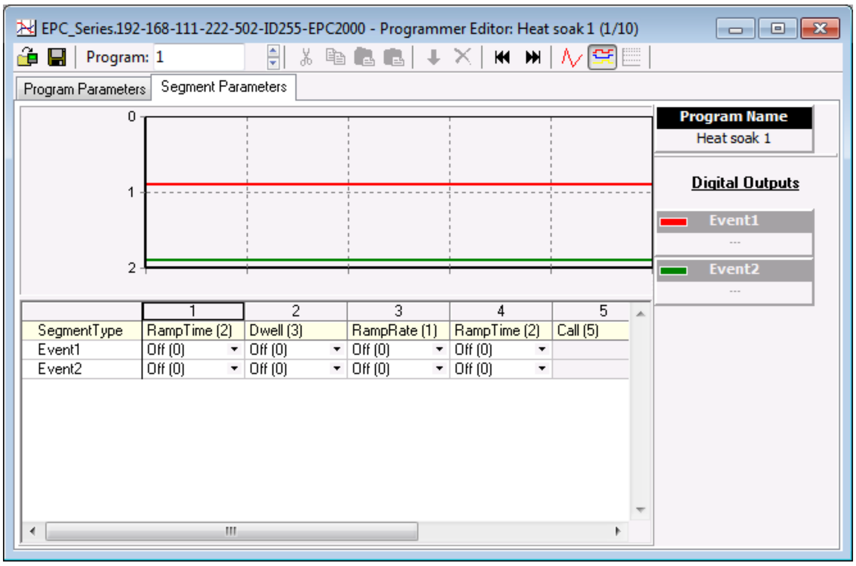
* See instrument handbook

OK Cancel

Tick bit 0 to turn on event 1 in the selected segment.

Tick bit 1 to turn on event 2 in the selected segment.

Alternatively, click on ‘Digital Event Outputs’ (Ctrl+D) to turn the events On or Off directly in each segment including the End segment.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Programmer Editor: Heat soak 1 (1/10)

Program: 1

Program Parameters | Segment Parameters

Program Name
Heat soak 1

Digital Outputs

Event1

Event2

	1	2	3	4	5
SegmentType	RampTime (2)	Dwell (3)	RampRate (1)	RampTime (2)	Call (5)
Event1	Off (0)	Off (0)	Off (0)	Off (0)	
Event2	Off (0)	Off (0)	Off (0)	Off (0)	

The above view shows two events configured.

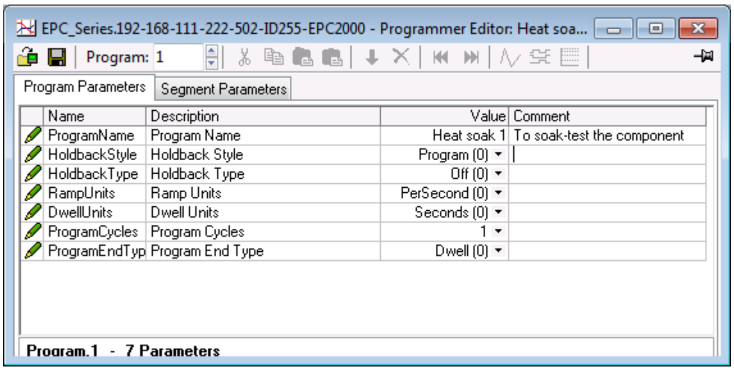
Events may be indication only or may be soft wired to a function block input parameter including an IO block (to operate external devices). This is explained in section "Graphical Wiring" on page 80.

To Name Programs and Segments

Programs and segments can be given alphanumeric names. These are UTF-8 encoded and the number of characters that can be used depends on the character set used. Program names can hold 16 ASCII characters and segment names can hold 40 ASCII characters.

Program Name

1. Select the Program Parameters tab.
2. In ‘ProgramName’ change the default text (P1).
3. A comment may be added in the Comment field as a reminder. This comment does not affect the operation and is not visible on a connected device.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Programmer Editor: Heat soa...

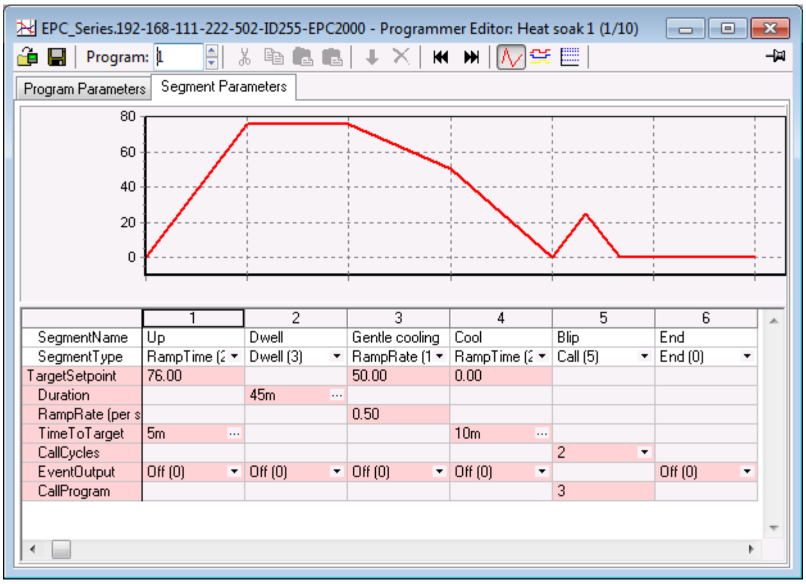
Program: 1

Name	Description	Value	Comment
ProgramName	Program Name	Heat soak 1	To soak-test the component
HoldbackStyle	Holdback Style	Program (0)	
HoldbackType	Holdback Type	Off (0)	
RampUnits	Ramp Units	PerSecond (0)	
DwellUnits	Dwell Units	Seconds (0)	
ProgramCycles	Program Cycles	1	
ProgramEndTyp	Program End Type	Dwell (0)	

Program.1 - 7 Parameters

Segment Name

1. Select the Segment Parameters tab.
2. In ‘SegmentName’ enter a name for each segment.
3. When the program is run, this name can be shown on a remote HMI.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Programmer Editor: Heat soak 1 (1/10)

Program: 1

	1	2	3	4	5	6
SegmentName	Up	Dwell	Gentle cooling	Cool	Blip	End
SegmentType	RampTime (2	Dwell (3)	RampRate (1	RampTime (2	Call (5)	End (0)
TargetSetpoint	76.00		50.00	0.00		
Duration		45m				
RampRate (per s			0.50			
TimeToTarget	5m			10m		
CallCycles					2	
EventOutput	Off (0)	Off (0)	Off (0)	Off (0)		Off (0)
CallProgram					3	

To Save and Load Program Files (*.uip)

A configured Program can be saved to a named file (stored on the local host PC). In a multi-program Programmer, each Program will need to be saved individually. A saved Program may be re-loaded into any Program location in the iTools Programmer Editor. In the case that similar production processes are to be defined, a saved Program may be re-loaded, modified and renamed.

To Save a Program

1. In the Programmer Editor, select the program number to be saved using the program selector.
2. There are two ways to save a program. In the Programmer Editor click on “Save current program to file (Ctrl+S)”. Alternatively, in the main menu click on Programmer and from the drop-down menu select “Save current program to file (Ctrl+S)”.

Do not confuse this with Save on the main toolbar which saves the instrument/device configuration to a Clone file.

To Load a Previously Saved Program

1. In the Programmer Editor, select the stored program number where the saved program is to be loaded using the program selector.
2. There are two ways to load a program.
   a. In the Programmer Editor click on “Load Program (Ctrl+L)”.
   b. Alternatively, in the main menu click on Programmer and from the drop-down menu select “Load...(Ctrl+L)”.

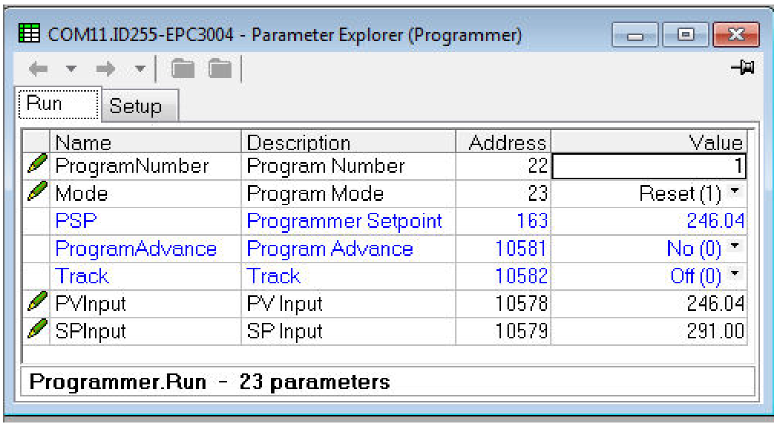
Do not confuse this with Load on the main toolbar as this will load a full configuration from a Clone file.

Note: The following:

1. If an attempt is made to load a program which contains a Call segment into the last stored program (e.g. program 10 or 20, depending on the feature security option) iTools will prohibit the action and report a message as follows, ‘Unable to load: Program 10 (the last program) cannot contain a call segment'.
2. A 1x8 or 1x24 programmer cannot contain any call segments.
3. If an attempt is made to load a program that has a greater number of event outputs (Programmer.Setup.MaxEvents) than the current program, iTools will prohibit the action and report a message as follows, 'Unable to load: EventOutputs used (6) exceeds MaxEvents for device (4). Increase MaxEvents to 6 and reload'.
4. If an attempt is made to load a program that has a greater number of segments than the current program, iTools will prohibit the action and report a message as follows, 'Unable to load: Maximum program size of 8 segments exceeded'.

To Run, Reset and Hold a Program

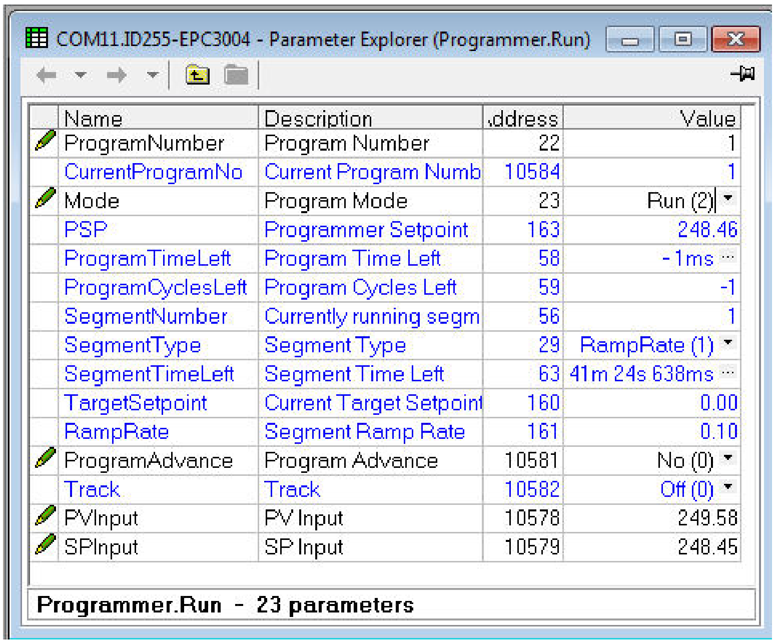
In the browser view, open the Program Run list:



	Name	Description	Address	Value
✎	ProgramNumber	Program Number	22	1
✎	Mode	Program Mode	23	Reset (1)
	PSP	Programmer Setpoint	163	246.04
	ProgramAdvance	Program Advance	10581	No (0)
	Track	Track	10582	Off (0)
✎	PVInput	PV Input	10578	246.04
✎	SPInput	SP Input	10579	291.00

Programmer.Run - 23 parameters

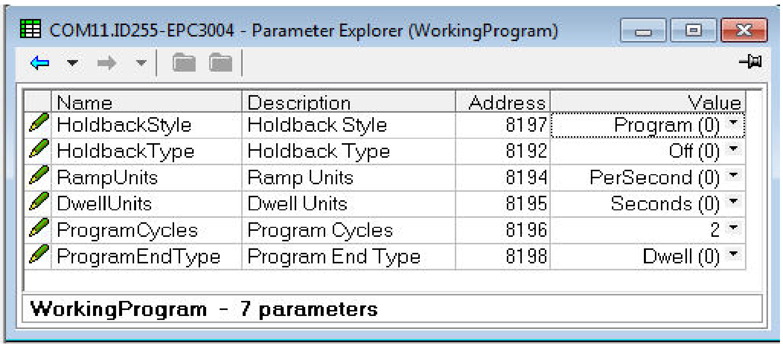
To run a program make sure that the controller is in Operator mode and that the PVInput parameter status is ‘Good’. Select the program number of the program to run and select Run(2) from the Mode parameter drop-down enumeration. The program may also be put into Hold or Reset from the mode Parameter.



	Name	Description	ddress	Value
✎	ProgramNumber	Program Number	22	1
	CurrentProgramNo	Current Program Numb	10584	1
✎	Mode	Program Mode	23	Run (2)
	PSP	Programmer Setpoint	163	248.46
	ProgramTimeLeft	Program Time Left	58	-1ms
	ProgramCyclesLeft	Program Cycles Left	59	-1
	SegmentNumber	Currently running segm	56	1
	SegmentType	Segment Type	29	RampRate (1)
	SegmentTimeLeft	Segment Time Left	63	41m 24s 638ms
	TargetSetpoint	Current Target Setpoint	160	0.00
	RampRate	Segment Ramp Rate	161	0.10
✎	ProgramAdvance	Program Advance	10581	No (0)
	Track	Track	10582	Off (0)
✎	PVInput	PV Input	10578	249.58
✎	SPInput	SP Input	10579	248.45

Programmer.Run - 23 parameters

When one of the programs (Program 1 to 10) is run, the program parameters are copied into the working program. The Working Program and Working Segment parameters are then made available for monitoring and/or editing.



	Name	Description	Address	Value
✎	HoldbackStyle	Holdback Style	8197	Program (0)
✎	HoldbackType	Holdback Type	8192	Off (0)
✎	RampUnits	Ramp Units	8194	PerSecond (0)
✎	DwellUnits	Dwell Units	8195	Seconds (0)
✎	ProgramCycles	Program Cycles	8196	2
✎	ProgramEndType	Program End Type	8198	Dwell (0)

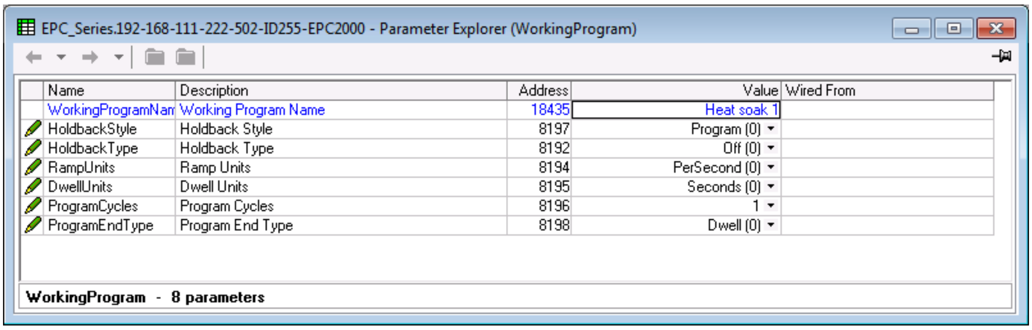
WorkingProgram - 7 parameters

The programmer loads each segment from the working program before it is run. If the programmer is currently running segment 2 of the working program and working segment 3 is edited, then the changes will be executed when working segment 3 is run. If working segment 1 is edited, then the changes will be executed in the next program cycle, (assuming there are any program cycles left). However, if the running program completes or is reset and then run again, the stored program will be copied to the working program thereby overwriting any changes made to the working program. The working program can also be overwritten as a result of running another program or calling another program as a subroutine.

Stored programs are available and configurable via iTools even when a program is currently running. Working program parameters, however, are only available and configurable via iTools when a program is not in reset.

Note: For a running program set for Continuous cycles (using the ProgramCycles parameter in the Programmer Parameters tab) the ‘Program Time Left’ parameter shows -1 in iTools. Similarly, in iTools the ‘Program Cycles Left’ parameter shows -1. If the program cycles is set to repeat for a set number of times, then the ‘Program Time Left’ parameter and the ‘Program Cycles Left’ parameter will count down in iTools.

The Working Program provides read/write access to the program parameters of the currently running program (which can be the main program or a subroutine resulting from a call segment).

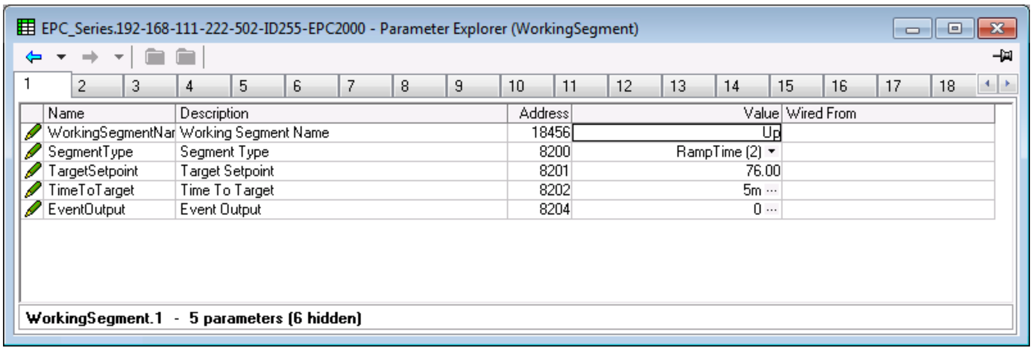


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (WorkingProgram)

Name	Description	Address	Value	Wired From
WorkingProgramNam	Working Program Name	18435	Heat soak 1	
HoldbackStyle	Holdback Style	8197	Program (0)	
HoldbackType	Holdback Type	8192	Off (0)	
RampUnits	Ramp Units	8194	PerSecond (0)	
DwellUnits	Dwell Units	8195	Seconds (0)	
ProgramCycles	Program Cycles	8196	1	
ProgramEndType	Program End Type	8198	Dwell (0)	

WorkingProgram - 8 parameters

The Working Segment provides read/write access to the segment parameters of the currently running program (which can be the main program or a subroutine resulting from a call segment).



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (WorkingSegment)

Name	Description	Address	Value	Wired From
WorkingSegmentNar	Working Segment Name	18456	Up	
SegmentType	Segment Type	8200	RampTime (2)	
TargetSetpoint	Target Setpoint	8201	76.00	
TimeToTarget	Time To Target	8202	5m	
EventOutput	Event Output	8204	0	

WorkingSegment.1 - 5 parameters (6 hidden)

Graphical Wiring

Graphical wiring provides a means of connecting function blocks together to produce a unique process. If the controller has been ordered or configured using the Quick Codes for a particular application then an example of the application has already been produced and is intended as a starting point to modify as required.

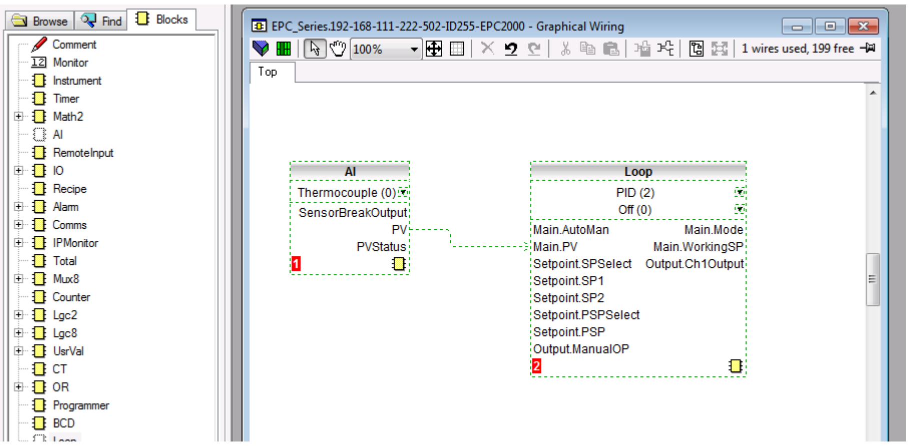
Select ‘Graphical Wiring’ on the main toolbar.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

This operation will require the controller to enter configuration mode. Ensure that the controller is not connected to an active process.

Failure to follow these instructions can result in injury or equipment damage.



A list of function blocks is shown in the left hand side. Blocks are dragged and dropped from the list into the Graphical Wiring section on the right.

They are ‘soft wired’ together to produce the application. The above example shows the Analog Input block wired to the PV input of the loop. This is produced by clicking on the ‘PV’ parameter of the Analog Input block and dragging to the ‘Main PV’ parameter of the Loop block.

Note: The value of a wired parameter cannot be manually changed since it takes on the value of the parameter it is wired from. The blocks and wires are shown dotted until the controller is updated using the ‘Download Wiring to Instrument’ button in the top left corner of the Graphical Wiring section.

For a full description of graphical wiring, refer to the iTools User Manual HA028838.

50 wires are available as standard, with 200 wires available if the Toolkit option has been ordered.

If the controller is ordered unconfigured it will be necessary for the user to wire the function blocks to suit the particular application.

Some examples of graphical wiring are shown in the following sections.

Example 1: To Wire an Alarm

Scan QR Code for EPC2000 'How To' video tutorials.

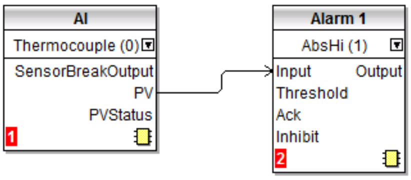
Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

Unless specifically produced in an application, any alarm which is required must be wired by the user.

The example below shows an absolute high alarm monitoring the Process Variable.

This is a 'soft' alarm in that it does not operate a physical output.

1. Drag and drop an alarm function block into the graphical wiring editor.
2. Drag and drop an analog input block into the graphical wiring editor.
3. Click on 'PV' of the input block and drag a wire to 'input' of the alarm block.
4. At this stage the wire is shown dotted and it must be transferred to the controller by clicking the 'Download Wiring to Instrument' button in the top left hand corner of the Graphical Wiring view.

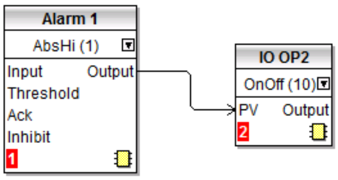


Example 2: To Connect an Alarm to a Physical Output

To make a soft alarm operate an output it must be 'wired'.

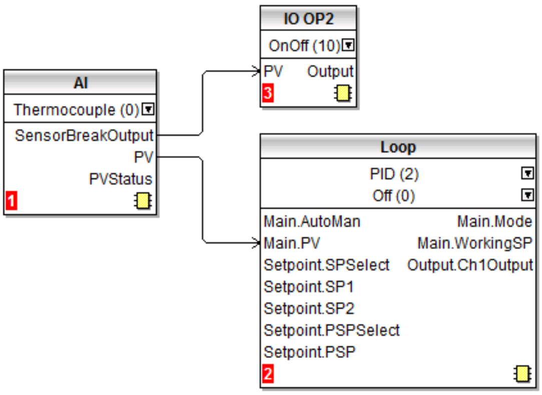
1. Drag and drop an alarm function block into the graphical wiring editor.
2. Drag and drop an output block into the graphical wiring editor.
3. Click on the 'Output' of the Alarm block and drag the wire to the 'PV' input of the output block.
4. At this stage the wire is shown dotted and it must be transferred to the controller by clicking the 'Download Wiring to Instrument' button.

The example shown below uses Alarm 1 and IO2 (configured for On/Off output).



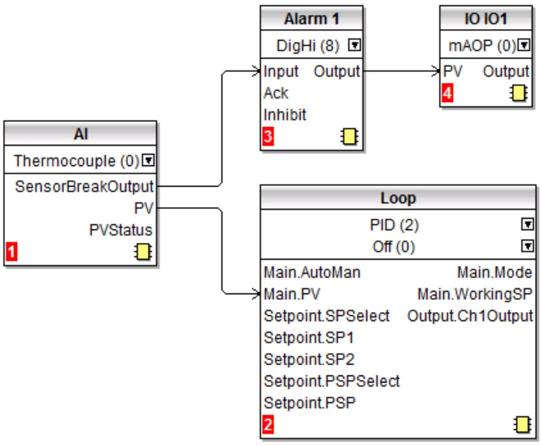
Example 3: To Wire Sensor Break

If a sensor condition is required to operate an output it must be wired as shown in the example below.



Sensor Break Alarm with Latching

In the above example a sensor break alarm has no latching ability. If latching is required the sensor break output can be wired to an alarm function block configured as a digital alarm which can be configured for auto or manual latching. A wiring example is shown below:



Flash Editor

The Flash Editor edits any device data that needs to be saved to the device flash in addition to the OPC Function Block parameter editing mechanism used for most configuration editing.

This includes Recipe Definition and Recipe data sets.

These sets of data are presented on a series of tabs as shown in the following views.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

Any changes made to the controllers flash require the controller to enter configuration mode. The controller will not control the process when in configuration mode. Ensure that the controller is not connected to an active process when in configuration mode.

Failure to follow these instructions can result in injury or equipment damage.

Recipes

A recipe is a list of parameters whose values can be captured and stored in a dataset which can then be loaded at any time to restore the recipe parameters, thus providing a means of altering the configuration of an instrument in a single operation even in Operator mode. Recipes can be set up and loaded using iTools.

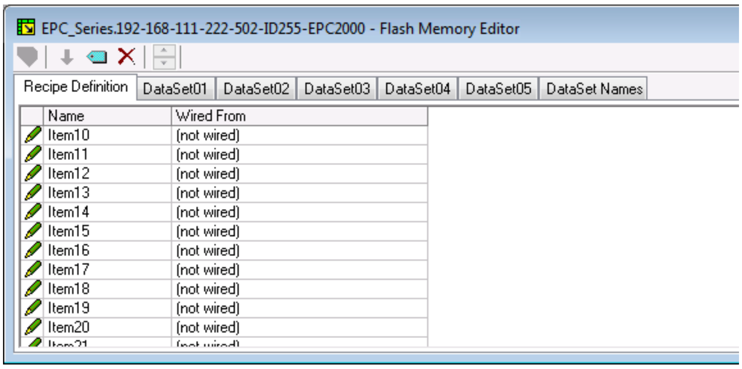
A maximum of five datasets are supported, referenced by name, and defaulted to be the dataset number i.e. 1…5.

By default each dataset consists of 40 parameters which must be populated by the user. A recipe can take a snapshot of the current values and store these into a recipe dataset.

Each dataset can be given a name using iTools configuration software.

Recipe Definitions

To open the Flash Editor, select ‘Flash’ from the main toolbar and then, select the ‘Recipe Definition’ and ‘Recipe Dataset’ tabs as required.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Flash Memory Editor

Name	Wired From
Item10	(not wired)
Item11	(not wired)
Item12	(not wired)
Item13	(not wired)
Item14	(not wired)
Item15	(not wired)
Item16	(not wired)
Item17	(not wired)
Item18	(not wired)
Item19	(not wired)
Item20	(not wired)
Item21	(not wired)

The Recipe Definition table contains a set of 40 parameters. Not all 40 parameters need to be wired.

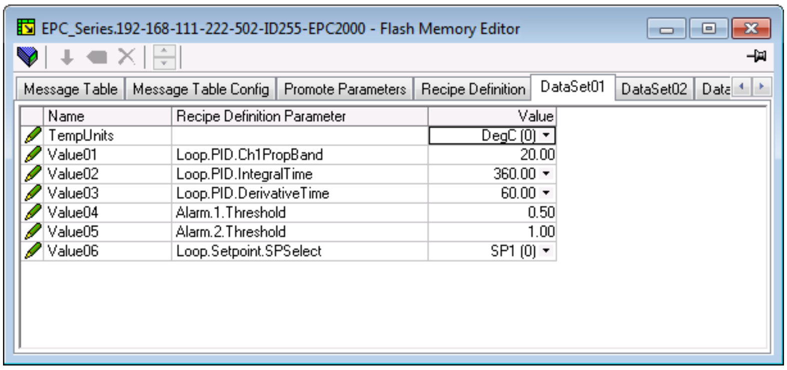
The Recipe Definition tab allows the user to produce a customized list.

To add parameters:

1. Double-click in the next empty item.
2. This opens the parameter list to choose from.
3. Adding a parameter to the list will automatically populate the five datasets with the current value of the added parameter.

Data Sets

Up to five DataSets are available each being a recipe for a particular batch or process.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Flash Memory Editor

Name	Recipe Definition Parameter	Value
TempUnits		DegC (0)
Value01	Loop.PID.Ch1PropBand	20.00
Value02	Loop.PID.IntegralTime	360.00
Value03	Loop.PID.DerivativeTime	60.00
Value04	Alarm.1.Threshold	0.50
Value05	Alarm.2.Threshold	1.00
Value06	Loop.Setpoint.SPSelect	SP1 (0)

To Save the Data Set

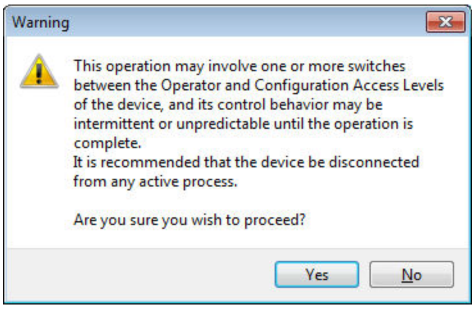
1. Set up the required values in the selected data set—see the example above.

2. Press Enter.

3. Press the ‘Update device flash ’ (Ctrl+F)’ button in the top left of the Flash Editor display to update the controller. This sets the values into all five of the controller datasets.

Note: Saving in the controller will save the current values into one dataset.

Since this operation may involve one or more switches between Operator Level and Configuration Level it is recommended that the controller is disconnected from the process. A warning message is shown.



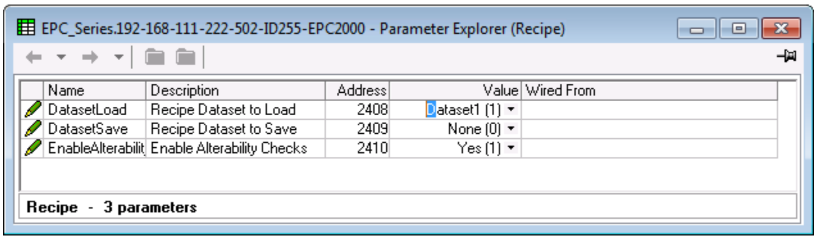
Warning

This operation may involve one or more switches between the Operator and Configuration Access Levels of the device, and its control behavior may be intermittent or unpredictable until the operation is complete.
It is recommended that the device be disconnected from any active process.

Are you sure you wish to proceed?

To Load a Data Set

1. In the browser list select ‘Recipe’.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Recipe)

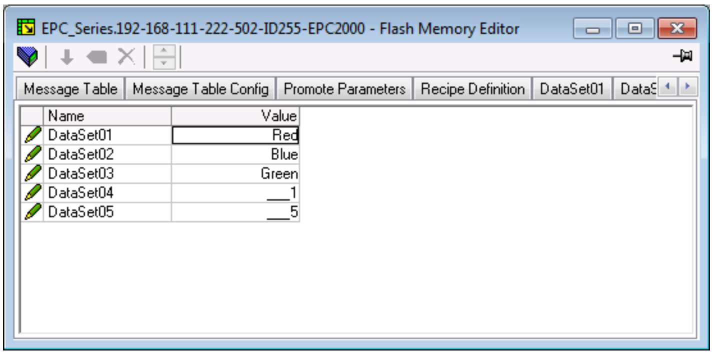
Name	Description	Address	Value	Wired From
DatasetLoad	Recipe Dataset to Load	2408	Dataset1 (1)	
DatasetSave	Recipe Dataset to Save	2409	None (0)	
EnableAlterabilit	Enable Alterability Checks	2410	Yes (1)	

Recipe - 3 parameters

2. Select the required Dataset.

Recipe Names

This tab simply allows a name to be assigned to each of the five Recipe data sets.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Flash Memory Editor

Name	Value
DataSet01	Red
DataSet02	Blue
DataSet03	Green
DataSet04	___1
DataSet05	___5

Watch Recipe Editor

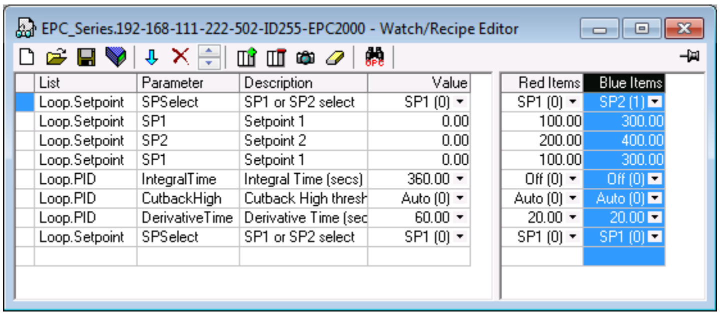
Click on the Watch/Recipe tool button, by selecting 'Watch/Recipe' in the Views menu, or via the shortcut (Alt+A). The window is in two parts: the left part contains the watch list; the right-hand part contains a data set, initially empty and unnamed.

Watch Recipes are run from iTools, and are not stored or run from the device, i.e. iTools must be running and connected to a specific device.

The window is used:

1. To monitor a so-called 'watch list' of parameter values. The watch list can contain parameters from many different lists within the same device.

2. To create 'data sets' of parameter values which can be selected and downloaded to the device, in the sequence defined by the recipe. The same parameter may be used more than once in a recipe.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Watch/Recipe Editor

List	Parameter	Description	Value	Red Items	Blue Items
Loop.Setpoint	SPSelect	SP1 or SP2 select	SP1 (0)	SP1 (0)	SP2 (1)
Loop.Setpoint	SP1	Setpoint 1	0.00	100.00	300.00
Loop.Setpoint	SP2	Setpoint 2	0.00	200.00	400.00
Loop.Setpoint	SP1	Setpoint 1	0.00	100.00	300.00
Loop.PID	IntegralTime	Integral Time (secs)	360.00	Off (0)	Off (0)
Loop.PID	CutbackHigh	Cutback High thresh	Auto (0)	Auto (0)	Auto (0)
Loop.PID	DerivativeTime	Derivative Time (sec	60.00	20.00	20.00
Loop.Setpoint	SPSelect	SP1 or SP2 select	SP1 (0)	SP1 (0)	SP1 (0)

Creating a Watch List

After opening the window parameters can be added to it as described below. Parameters can be added only from the device to which the Watch/Recipe window relates (that is, parameters from more than one device cannot be placed in one Watch list). The values of the parameters update in real time, allowing the user to monitor, simultaneously, a number of parameters which might otherwise be unrelated.

Adding Parameters to the Watch List

1. Parameters can be clicked and dragged into the watch list grid from elsewhere in iTools (for example: the main browse tree, the Parameter Explorer window, the Graphical Wiring Editor (if applicable)). The parameter is placed either in the empty row at the bottom of the list, or 'on top' of an existing parameter, in which case it is inserted above this parameter in the list, the remaining parameters moving down one place.

2. Parameters can be dragged from one position in the list to another. In such a case, a copy of the parameter is produced: the source parameter remains in place. Parameters can be also be copied by using the 'Copy Parameter' item in the Recipe or right mouse-click menu, or by using the shortcut (Ctrl+C). Data set values are not included in the copy.

3. The 'Insert item...' tool button, the 'Insert Parameter' Recipe menu item or the shortcut <Insert> can be used to open a browse window from which a parameter can be selected. The selected parameter is inserted above the currently active parameter.

4. A parameter can be 'copied' from (for example) the Graphical Wiring Editor and subsequently 'pasted' into the watch list using the 'Paste Parameter 'item in the Recipe menu, or the right mouse-click context menu (shortcut = Ctrl+V).

Creating a Data Set

All the parameters required for the recipe should be added to the watch list, described above.

Once this has been done, if the empty data set is selected (by clicking on the column header), the 'Snapshot' tool button (Ctrl+A) can be used to fill the data set with the current values. Alternatively, the 'Snapshot Values' item in the Recipe or context (right-click) menu or the shortcut + can be used to fill the data set.

Individual data values can now be edited by typing directly into the grid cells. Data values can be left blank or cleared, in which case, when the recipe is downloaded, no value will be written for those values. Data values can be cleared by deleting all the characters in the field, then, either moving to a new cell, or typing <Enter>.

The set is called 'Set 1' by default. The name can be edited by using the 'Rename Data Set...' item in the Recipe or right mouse-click menu, or by using the shortcut (Ctrl+R).

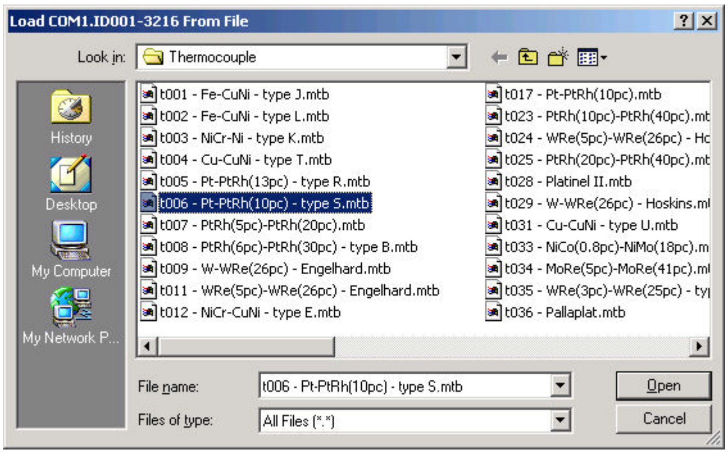
New data sets can be added and edited in the same way, by using the 'Create a new empty....' tool button (Ctrl+W), or by selecting the 'New Data Set' item in the Recipe or right mouse-click menu, or by using the shortcut +.

Once all the data sets required for the Recipe have been created, and saved, they can be downloaded to the device, one at a time, using the download tool (Ctrl+D), or equivalent Recipe/context menu item.

To Load a Custom Linearization Table

In addition to the built-in standard linearization tables, custom tables can be downloaded from files.

1. Press Load
2. Select the linearization table to be loaded from files with the extension .mtb. Linearization files for different sensor types are supplied with iTools and may be found in Program Files (x86) ➞ Eurotherm ➞ iTools ➞ Linearizations ➞ Thermocouple etc.



Load COM1.ID001-3216 From File

Look in: Thermocouple

File name: t006 - Pt-PtRh(10pc) - type S.mtb

Files of type: All Files (*.*)

History
Desktop
My Computer
My Network P...
t001 - Fe-CuNi - type J.mtb
t002 - Fe-CuNi - type L.mtb
t003 - NiCr-Ni - type K.mtb
t004 - Cu-CuNi - type T.mtb
t005 - Pt-PtRh(13pc) - type R.mtb
t006 - Pt-PtRh(10pc) - type S.mtb
t007 - PtRh(5pc)-PtRh(20pc).mtb
t008 - PtRh(6pc)-PtRh(30pc) - type B.mtb
t009 - W-WRe(26pc) - Engelhard.mtb
t011 - WRe(5pc)-WRe(26pc) - Engelhard.mtb
t012 - NiCr-CuNi - type E.mtb
t017 - Pt-PtRh(10pc).mtb
t023 - PtRh(10pc)-PtRh(40pc).mt
t024 - WRe(5pc)-WRe(26pc) - Hc
t025 - PtRh(20pc)-PtRh(40pc).mt
t028 - Platinel II.mtb
t029 - W-WRe(26pc) - Hoskins.m
t031 - Cu-CuNi - type U.mtb
t033 - NiCo(0.8pc)-NiMo(18pc).m
t034 - MoRe(5pc)-MoRe(41pc).m
t035 - WRe(3pc)-WRe(25pc) - ty
t036 - Pallaplat.mtb
Open
Cancel

In this example a Pt-PTRh(10%) thermocouple has been loaded into the controller.

Cloning

The cloning feature allows the configuration and parameter settings of one instrument to be copied into another. Alternatively a configuration may be saved to file and this used to load into connected instruments. The feature allows new instruments to be rapidly set up using a known reference source or standard instrument. Every parameter is downloaded to the new instrument which means that if the new instrument is used as a replacement it will contain exactly the same information as the original. Cloning is generally only possible if the following applies:

- The target instrument has the same hardware configuration as the source instrument.
- The target instrument firmware (i.e. software built into the instrument) is the same as or a later version than that of the source instrument.
- Generally, cloning will copy all operational, engineering and configuration parameters that are writable. The communications address is not copied.

Note: A clone file cannot be generated if the OEM Security feature option is configured and active (see "OEM Security" on page 290).

⚠ **WARNING**

UNINTENDED EQUIPMENT OPERATION

It is the users responsibility to ensure that the information cloned from one instrument to another is correct for the process to be controlled, and that all parameters are correctly replicated into the target instrument.

Failure to follow these instructions can result in death, serious injury or equipment damage

Below is a brief description of how to use this feature. Further details are available in the iTools Manual.

Save to File

The configuration of the controller made in the previous sections may be saved as a clone file. This file can then be used to transfer the configuration to further instruments.

From the File menu use ‘Save to File’ or use the ‘Save’ button on the Toolbar.

To Clone a New Controller

Connect the new controller to iTools and Scan to find this instrument as described at the beginning of this chapter.

From the File menu, select ‘Load Values From File’ or select ‘Load’ from the toolbar. Choose the required file and follow the instruction. The configuration of the original controller will now be transferred to the new controller.

Clone Load Unsuccessful

A Message Log is produced during the cloning process. The log may show a message such as 'Cloning of device completed with 1 unsuccessful entry'. This can be caused by writing a parameter using iTools which is outside of the resolution of a parameter. For example, the parameter Filter Time Constant is stored in the controller to one decimal place (1.6 seconds by default). If it is entered as an IEEE float value, using iTools, as, say, 1.66 it will be rounded up in the controller to 1.7 seconds. Under these circumstances it is possible that 'Clone Load Unsuccessful' can occur because iTools expects a value of 1.66 and the instrument contains 1.7. Values should, therefore, be entered, when using iTools, within the resolution of the parameter.

Cold Start

⚠ CAUTION

COLD START

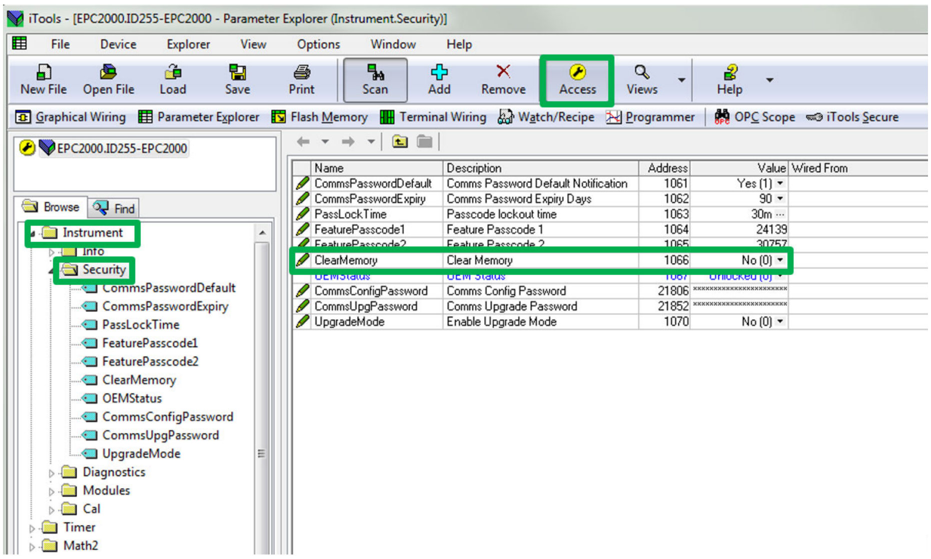
A cold start of the controller must only be carried out under exceptional circumstances as it will erase ALL previous settings and return the controller to its original state.

A controller must not be connected to any equipment when performing a cold start.

Failure to follow these instructions can result in injury or equipment damage.

To Perform a Cold Start

In the Instrument.Security function block, set the 'Clear' parameter to Yes. The controller restarts and clears all user-configured parameters.



iTools - [EPC2000.ID255-EPC2000 - Parameter Explorer (Instrument.Security)]

Name	Description	Address	Value	Wired From
CommsPasswordDefault	Comms Password Default Notification	1061	Yes (1)	
CommsPasswordExpiry	Comms Password Expiry Days	1062	90	
PassLockTime	Passcode lockout time	1063	30m	
FeaturePasscode1	Feature Passcode 1	1064	24139	
FeaturePasscode2	Feature Passcode 2	1065	30757	
ClearMemory	Clear Memory	1066	No (0)	
OEMStatus	OEM Status	1067	Unlocked (0)	
CommsConfigPassword	Comms Config Password	21806	**********************	
CommsUpgPassword	Comms Upgrade Password	21852	**********************	
UpgradeMode	Enable Upgrade Mode	1070	No (0)	

Configuration

Configuration of the instrument may be necessary at the first start up, during commissioning or when relatively small changes are required on site.

To configure the instrument the use of Eurotherm iTools configuration package is needed and this is described in the Configuration Using iTools chapter starting on page 64. iTools is proprietary software designed to configure Eurotherm instruments. It is available as a free download from www.eurotherm.com or can be ordered on a DVD.

What’s in this Chapter?

- How to enter and exit configuration mode.
- Introduction to Function Blocks.
- A complete list of configuration parameters which are available in each function block.

Configuration mode

To Access

To put the EPC2000 Programmable Controller into configuration mode, launch iTools (refer to "Starting iTools" on page 66), locate the instrument and click the Access button in the toolbar. You will be prompted to enter the Comms Config Password. This is set when the instrument is first started and can be changed using the Instrument>Security>CommsPassword parameter in configuration mode.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

Putting the controller into configuration mode stops any active control. Ensure that the controller is not connected to an active process.

Failure to follow these instructions can result in injury or equipment damage.

If the instrument is already in configuration mode using an alternative physical connection (Ethernet or the Serial Communication EIA-485 option), you cannot enter the configuration level from this current session. Exit configuration mode in the other session and then try again.

To Close Access

To exit the configuration level, click the Access button again to deselect it. The instrument exits configuration level.

⚠ CAUTION

UNINTENDED EQUIPMENT OPERATION

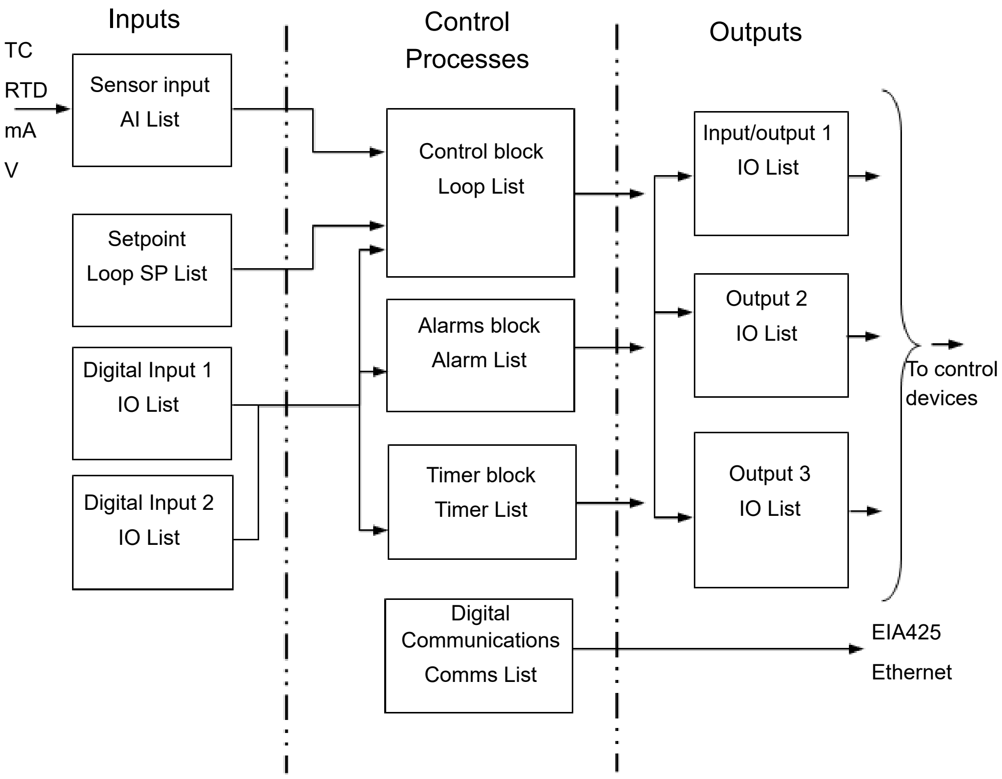
Exiting the configuration level will activate the control application and the controller outputs (IOs). Ensure the complete control process is ready and it is safe for controller to resume operation.

Failure to follow these instructions can result in injury or equipment damage.

Function Blocks

The controller is made up of a number of hardware and software function blocks. Each block has inputs and outputs which are wired together in software (soft wired) to match the application for which the controller is intended.

The following diagram shows an example of function blocks which make up a typical controller.



The Temperature (or Process Value, PV) is measured by the sensor and compared with a Setpoint (SP) set by the user.

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

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

The way in which each block performs is defined by its internal parameters. Some of these parameters are available to the user so that they can be adjusted to suit the characteristics of the process which is to be controlled.

These parameters are found in lists in the iTools browser list.

Configuration Parameters

The following pages list all parameters which are available in the controller in their respective function blocks. Parameters are only shown in the controller if the feature has been supplied and enabled. This section details all available parameters within a function block and is presented in the order that iTools shows them.

Some function blocks have subclasses. For example, the Instrument function block has nine subclasses (Info, Security, Diagnostics, Modules, Enables, Cal, OEMConfigList, OEMOperList and RemoteHMI). The Security subclass is indicated by 'Instrument.Security' (meaning the Security subclass under the Instrument function block).

Some parameters hold analog values between set limits. Other parameters can contain alphanumeric text. Many other parameters are enumerated, which means they have options that can be selected from a list.

Common parameter values

There are some parameters for which the explanation of their meaning is common across the EPC2000 Programmable Controller. Primarily, these are the Units and Status parameters. Below is a summary of these two parameters.

Units

The table below applies to all function blocks which contain units.

Parameter Name	Description	Available values	Value Description
Units	Units	None (0)	No units will be displayed.
		AtmP (1)	Temperature units. °C, °F, K are set in the Instrument.Info function block (see section "Instrument.Info" on page 94).
		V (2)	Volts
		mV (3)	Millivolts
		A (4)	Amps
		mA (5)	Milliamps
		PH (6)	pH
		mmHG (7)	Millimeters of mercury
		PSi (8)	Pounds per square inch
		bAr (9)	Bar
		mBar (10)	millibar
		P.RH (11)	Percent relative humidity
		PErc (12)	Percent
		mmwG (13)	Millimeter Water Gauge
		inwG (14)	Inches Water Gauge
		inWW (15)	Not used
		OhmS (16)	Resistance (ohms)
		PSIG (17)	Pounds per square inch gauge
		P.O2 (18)	Percent O_2
		PPm (19)	Parts per million
		P.CO2 (20)	Percent CO_2
		P.CP (21)	Percent carbon
		P.SEc (22)	Percent per second

Status

The table below applies to all function blocks which contain global status enumeration.

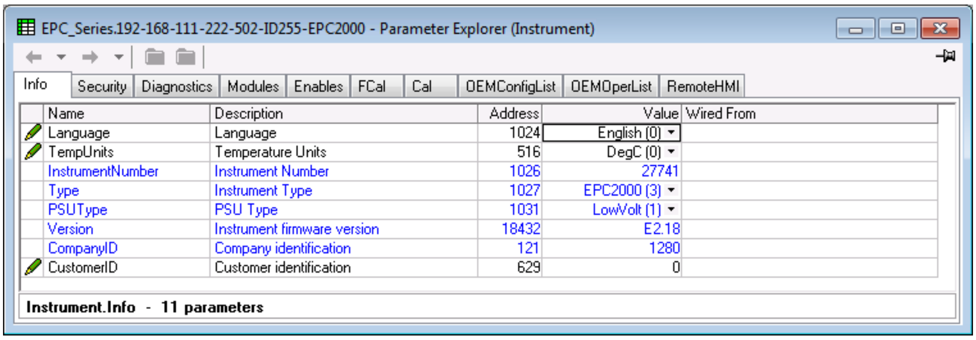
Parameter Name	Description	Available values	Value Description
		Good (0)	The process variable is working correctly.
		Off (1)	Channel is configured to be off.
		O.rng (2)	When the input signal exceeds the upper input limit by more than 5% the PV will flash indicating over range.
		U.rng (3)	When the input signal exceeds the lower input limit by more than 5% the PV will flash indicating under range.
		Hw.s (4)	Input hardware status is unknown.
		Rng (5)	The input status is set to Ranging at the point of an analog input configuration change. It will remain in Ranging until an exit from a configuration-induced instrument restart.
		OFLw (6)	Process variable overflow, possibly due to calculation attempting to divide a number by a relatively small number.
		Bad (7)	The PV is not reading correctly which could be due to an open sensor.
		Hwc (8)	The hardware capabilities have been exceeded at the point of configuration, for example configuration set to 0 to 40V when input hardware is capable of up to 10V.
		Ndat (9)	Insufficient input samples to perform calculation.

Instrument

This category contains nine function blocks: Info, Security, Diagnostics, Modules, Enables, Cal, OEMConfigList, OEMOperList and RemoteHMI. These all control the basic functionality of the instrument.

Instrument.Info

The Instrument.Info function block allows you to read and adjust information such as the Instrument Language, Temperature Units, and Customer ID. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

Name	Description	Address	Value	Wired From
Language	Language	1024	English (0)	
TempUnits	Temperature Units	516	DegC (0)	
InstrumentNumber	Instrument Number	1026	27741	
Type	Instrument Type	1027	EPC2000 (3)	
PSUType	PSU Type	1031	LowVolt (1)	
Version	Instrument firmware version	18432	E2.18	
CompanyID	Company identification	121	1280	
CustomerID	Customer identification	629	0	

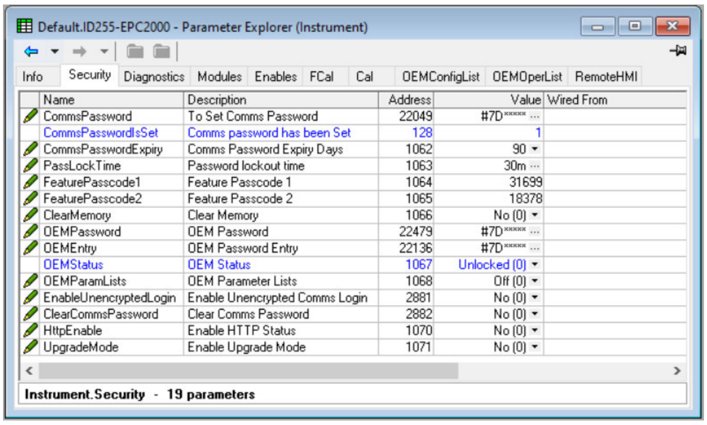
Instrument.Info - 11 parameters

Parameter Name	Description	Available values	Value Description
Language	Language	English (0)	English **Default: English (0)**
		French (1)	French
		German (2)	German
		Italian (3)	Italian
		Spanish (4)	Spanish

Parameter Name	Description	Available values	Value Description
TempUnits	Temperature Units	DegC (0)	Sets the temperature units to Celsius (C). When temperature units is changed, those parameters which are flagged as having a temperature type (Absolute or relative) will have their values converted to reflect the new temperature units. **Default: DegC (0)**
		DegF (1)	Sets the temperature units to Fahrenheit (F).
		DegK (2)	Sets the temperature units to Kelvin (K).
InstrumentNumber	Instrument Number		Unique instrument serial number.
InstrumentType	InstrumentType	EPC2000 (3)	Instrument type: EPC2000 Programmable Controller.
PSUType	PSU Type	HighVolt (0)	100 to 230Vac +/- 15% voltage PSU option (not applicable to the EPC2000 Programmable Controller).
		LowVolt (1)	24Vac/dc voltage PSU option.
Version	Instrument Firmware version	Firmware version number.	
CompanyID	Company identification	Eurotherm CNOMO identifier.	
CustomerID	Customer identification	A non-volatile value for customer use: it has no effect on the instrument functionality. **Default: 0**	

Instrument.Security

The security subclass sets up security settings. The figure below shows the parameters and the table which follows details each parameter.



Name	Description	Address	Value	Wired From
CommsPassword	To Set Comms Password	22049	#7D***** ...	
CommsPasswordIsSet	Comms password has been Set	128	1	
CommsPasswordExpiry	Comms Password Expiry Days	1062	90	
PassLockTime	Password lockout time	1063	30m ...	
FeaturePasscode1	Feature Passcode 1	1064	31699	
FeaturePasscode2	Feature Passcode 2	1065	18378	
ClearMemory	Clear Memory	1066	No (0)	
OEMPassword	OEM Password	22479	#7D***** ...	
OEMEntry	OEM Password Entry	22136	#7D***** ...	
OEMStatus	OEM Status	1067	Unlocked (0)	
OEMParamLists	OEM Parameter Lists	1068	Off (0)	
EnableUnencryptedLogin	Enable Unencrypted Comms Login	2881	No (0)	
ClearCommsPassword	Clear Comms Password	2882	No (0)	
HttpEnable	Enable HTTP Status	1070	No (0)	
UpgradeMode	Enable Upgrade Mode	1071	No (0)	

Instrument.Security - 19 parameters

Parameter Name	Description	Available Values	Value Description
CommsPassword	Comms Configuration Password	The configured password that will be required to place the instrument into comms configuration mode when connected via Serial comms or Ethernet comms. This password is encrypted before being transmitted via comms and is hashed with a random salt before being stored. See also section "Comms Configuration Level Password" on page 19. **Default: Empty (must be set on first connect from iTools)**	
CommsPasswordIsSet	Comms password has been set	No (0) Yes (1)	Flag to indicate if the comms password is set. **Default: No**
CommsPasswordExpiry	Comms Password Expiry Days	The number of days after which the configuration password will expire. When the password expires, a bit in the Instrument Status word is set to inform the user that the password needs changing. It should be noted that a value of 0 will disable the expiry feature. **Default: 90**	
PassLockTime	Password lockout time	0 to 24 hrs	The password entry mechanism will be locked out for this period after the maximum number of invalid attempts to login. This lockout time affects all access level passwords and the comms config password. A value of 0 will disable the lockout mechanism. **Default: 30 minutes**

Parameter Name	Description	Available Values	Value Description
FeaturePasscode1	Feature Passcode 1	Feature passcodes are required to enable chargeable features. These can be added after the controller has been supplied. Examples of chargeable features include Programmer Types, Toolkit Blocks, some Digital Communications protocols, etc. The passcodes can only be added through iTools. To request a feature, contact your supplier and provide the current values of the feature passcodes. Your supplier will provide new values to be entered to update the new feature. Enter these to enable your selected features. Three login attempts are allowed before lockout followed by a lockout period of 30 minutes.	
FeaturePasscode2	Feature Passcode 2		
ClearMemory	Clear Memory	Yes (1) No (0)	See CAUTION table below. Clear memory will force all parameters to be set to their factory default values. **Default: No**
The following four parameters apply to the optional OEM security features. For further information, see "OEM Security" on page 290.			
OEMPassword	OEM Password	These parameters are only shown if the OEM Security feature has been supplied. See Chapter "OEM Security" on page 290 for further information.	
OEMEntry	OEM Password Entry		
OEMStatus	OEM Status		
OEMParamLists	OEM Parameter Lists		
EnableUnencryptedLogin	Enable unencrypted login	No (0) Yes (1)	When this parameter is set, a string parameter (Modbus address 0x5500) becomes available via comms for entering plain text passwords to enter configuration mode. Note: Unlike the encrypted password, the entered string is transmitted as plain text via comms. Upon receipt, instrument hashes the message with the saved random salt before comparing it to the saved hashed password. **Default: No**
ClearCommsPassword	Clear comms configuration password	No (0) Yes (1)	When set, the comms configuration password will be cleared. Note: Clearing the comms configuration password will limit Serial comms and Ethernet comms read/write access to a small set of parameters until a new comms password is set. **Default: No**
HttpEnable	Enable HTTP status	No (0) Yes (1)	Enables HTTP status reporting for upgrades. Can only be activated from config mode. **Default: No**
UpgradeMode	Enable Upgrade Mode	No (0) Yes (1)	Switches the device into upgrade mode. Can only be triggered from config mode. **Default: No**

⚠ CAUTION

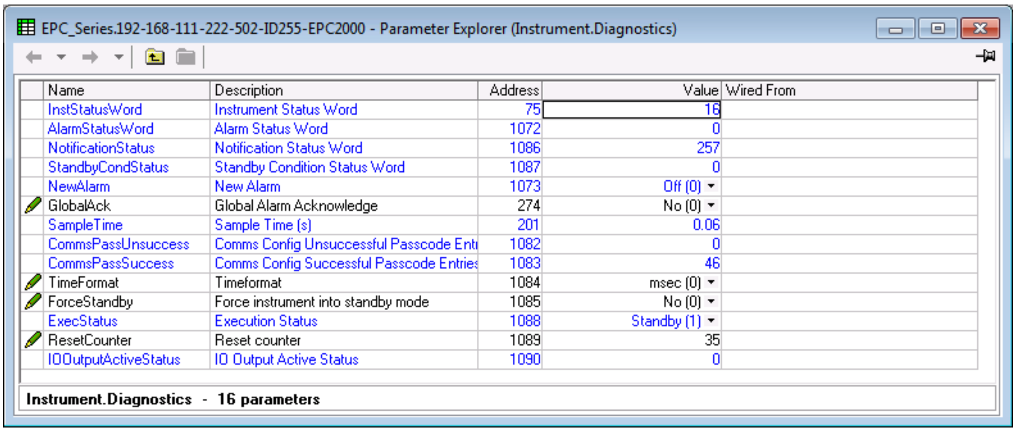
UNINTENDED EQUIPMENT OPERATION

The ClearMemory parameter will force all parameters to be set to their factory default values. This will cause all values previously set by the user to be lost and it should, therefore, only be used in exceptional circumstances. This parameter is only available if the CFG.P parameter has been set to 9999.

Failure to follow these instructions can result in injury or equipment damage.

Instrument.Diagnostics

The diagnostics subclass provides general diagnostics information. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument.Diagnostics)

	Name	Description	Address	Value	Wired From
	InstStatusWord	Instrument Status Word	75	16	
	AlarmStatusWord	Alarm Status Word	1072	0	
	NotificationStatus	Notification Status Word	1086	257	
	StandbyCondStatus	Standby Condition Status Word	1087	0	
	NewAlarm	New Alarm	1073	Off (0)	
✎	GlobalAck	Global Alarm Acknowledge	274	No (0)	
	SampleTime	Sample Time (s)	201	0.06	
	CommsPassUnsuccess	Comms Config Unsuccessful Passcode Entı	1082	0	
	CommsPassSuccess	Comms Config Successful Passcode Entries	1083	46	
✎	TimeFormat	Timeformat	1084	msec (0)	
✎	ForceStandby	Force instrument into standby mode	1085	No (0)	
	ExecStatus	Execution Status	1088	Standby (1)	
✎	ResetCounter	Reset counter	1089	35	
	IOOutputActiveStatus	IO Output Active Status	1090	0	

Instrument.Diagnostics - 16 parameters

Parameter Name	Description	Available Values	Value Description
InstStatusWord	Instrument Status Word	Instrument Status Word. This is a 16-bit bitmapped parameter that provides instrument status information. It is mapped as shown in the next section.	
AlarmStatusWord	Alarm Status Word	Alarm Status Word. This is a16-bit bitmapped parameter that provides alarm status information. It is mapped as shown in the following section.	
NotificationStatus	Notification Status Word	Notifications Status Word. This is a 16-bit bitmapped parameter that provides instrument notification status information. It is mapped as shown in the following section.	
StandbyCondStatus	Standby Condition Status Word	Standby Conditions Status Word (include the bit map table).	
NewAlarm	New Alarm	Off (0)	
		On (1)	This shows ON when a process alarm (see Alarm List) becomes active and remains on until the alarm has become inactive (and acknowledged depending upon the latching strategy of the alarm).
GlobalAck	Global Alarm Acknowledge	No (0)	
		Yes (1)	A rising edge will acknowledge all active process alarms (see the Alarm List).
SampleTime	Sample Time (s)	Indicates the sample period (in seconds). This is the period between each execution cycle.	
CommsPassUnsuccess	Comms Config Unsuccessful Password Entries	Number of Communications Configuration Mode unsuccessful login attempts since the last successful login.	
CommsPassSuccess	Comms Config Successful Password Entries	Number of Communications Configuration Mode successful logins.	
TimeFormat	Timeformat	msec (0)	Sets the resolution of time parameters on the config comms channel when read/written via scaled integer comms. **Default: msec(0)**
		sec (1)	
		min (2)	
		hour (3)	
ForceStandby	Force instrument into standby mode	No (0)	**Default: No (0)**
		Yes (1)	Sets the instrument into Standby Mode (see "Standby" on page 62).
ExecStatus	Execution Status	Indicates the status of the execution engine. This parameter can be used to determine if the instrument execution is running, in standby or starting up.	
		Running (0)	Running
		Standby (1)	Standby
		Startup (2)	Start up

Parameter Name	Description	Available Values	Value Description
ResetCounter	Reset counter	This indicates the number of times the instrument has reset due to a power cycle, exit from configuration mode, exit from quick start, or an unexpected software reset. The count value can be reset by writing a value of 0. **Default: 0**	

Instrument Status Word Bitmap

Bit Number	Description
0	Alarm 1 State (0=Off, 1=On).
1	Alarm 2 State (0=Off, 1=On).
2	Alarm 3 State (0=Off, 1=On).
3	Alarm 4 State (0=Off, 1=On).
4	Manual Mode (0=Auto, 1=Manual).
5	Global (PV1) Sensor Break (0=Off, 1=On).
6	Loop Break (0=Good closed loop, 1=Open loop).
7	**Not applicable to the EPC2000 Programmable Controller**
8	Auto Tune (0=Off, 1=On).
9	Program End (0=No, 1=Yes).
10	PV1 out of range (0=No, 1=Yes).
11	**Not applicable to the EPC2000 Programmable Controller**
12	New Alarm (0=No, 1=Yes).
13	Programmer Running (0=No, 1=Yes).
14	**Not applicable to the EPC2000 Programmable Controller**
15	**Not applicable to the EPC2000 Programmable Controller**

Alarm Status Word Bitmap

Bit Number	Description
0	Alarm 1 in active region (0=No,1=Yes).
1	Alarm 1 not acknowledged (0=No,1=Yes).
2	Alarm 2 in active region (0=No,1=Yes).
3	Alarm 1 not acknowledged (0=No,1=Yes).
4	Alarm 3 in active region (0=No,1=Yes).
5	Alarm 3 not acknowledged (0=No,1=Yes).
6	Alarm 4 in active region (0=No,1=Yes).
7	Alarm 4 not acknowledged (0=No,1=Yes).
8	Alarm 5 in active region (0=No,1=Yes).
9	Alarm 5 not acknowledged (0=No,1=Yes).
10	Alarm 6 in active region (0=No,1=Yes).
11	Alarm 6 not acknowledged (0=No,1=Yes).
12	Reserved.
13	**Not applicable to the EPC2000 Programmable Controller**
14	**Not applicable to the EPC2000 Programmable Controller**
15	**Not applicable to the EPC2000 Programmable Controller**

Notification Status Word Bitmap

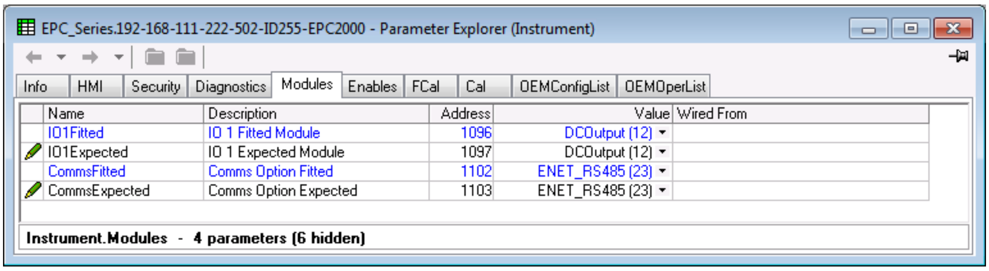
Bit Number	Description
0	Set comms password in iTools.
1	Password has expired.
2	**Not applicable to the EPC2000 Programmable Controller**
3	**Not applicable to the EPC2000 Programmable Controller**
4	**Not applicable to the EPC2000 Programmable Controller**
5	Comms configuration access locked out.
6	Control loop in demo mode.
7	Control loop in auto tune mode.
8	Comms in configuration mode.
9	Loop Autotune requested, but cannot run.
10	Reserved.
11	Reserved.
12	Reserved.
13	Reserved.
14	Reserved.
15	Reserved.

Standby Status Word Bitmap

Bit Number	Description
0	Invalid RAM image of NVOL.
1	NVOL parameter database load/store was unsuccessful.
2	NVOL region load/store was unsuccessful.
3	Option NVOL load/store was unsuccessful.
4	Factory Calibration not detected.
5	Unexpected CPU condition.
6	Hardware Ident Unknown.
7	Fitted hardware differs from expected hardware.
8	**Not applicable to the EPC2000 Programmable Controller**
9	Instrument powered down whilst in Configuration mode.
10	Recipe load was unsuccessful.
11	Reserved.
12	Reserved.
13	Reserved.
14	Reserved.
15	Reserved.

Instrument.Modules

The modules subclass provides information about the modules fitted in the controller. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

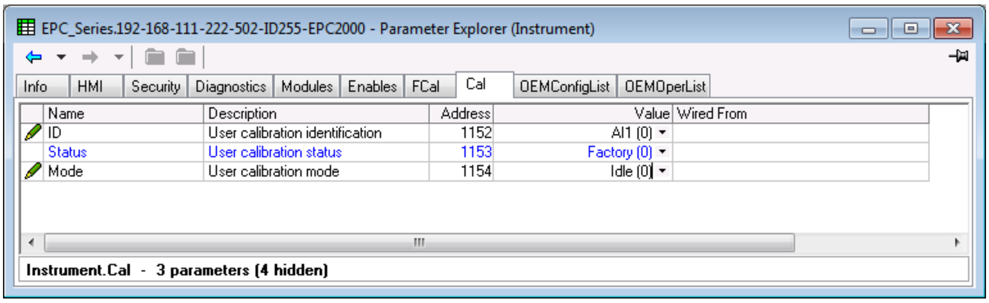
Name	Description	Address	Value	Wired From
IO1Fitted	IO 1 Fitted Module	1096	DCOutput (12)	
IO1Expected	IO 1 Expected Module	1097	DCOutput (12)	
CommsFitted	Comms Option Fitted	1102	ENET_RS485 (23)	
CommsExpected	Comms Option Expected	1103	ENET_RS485 (23)	

Instrument.Modules - 4 parameters (6 hidden)

Parameter Name	Description	Available Values	Value Description
IO1Fitted	IO 1 Fitted Module	LogicIO (11)	The module type that is actually fitted in IO1.
		DCOutput (12)	
IO1Expected	IO 1 Expected Module	LogicIO (11)	The module type that is expected in IO1.
		DCOutput (12)	
CommsFitted	Comms Option Fitted	The comms option that is actually fitted:	
		ENET_RS485 (23)	Ethernet and EIA-485.
		ENET (24)	Ethernet.
CommsExpected	Comms Option Expected	The comms option that is expected:	
		ENET_RS485 (23)	Ethernet and EIA-485.
		ENET (24)	Ethernet.

Instrument.Cal

The Cal subclass provides user calibration status information and a means of input and output calibration. Information and instructions for User Calibration are given in "User Calibration" on page 286. The figure below shows the parameters and the table which follows details each parameter.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)
Info
HMI
Security
Diagnostics
Modules
Enables
FCal
Cal
OEMConfigList
OEMOperList

Name	Description	Address	Value	Wired From
ID	User calibration identification	1152	AI1 (0)	
Status	User calibration status	1153	Factory (0)	
Mode	User calibration mode	1154	Idle (0)	

Instrument.Cal - 3 parameters (4 hidden)

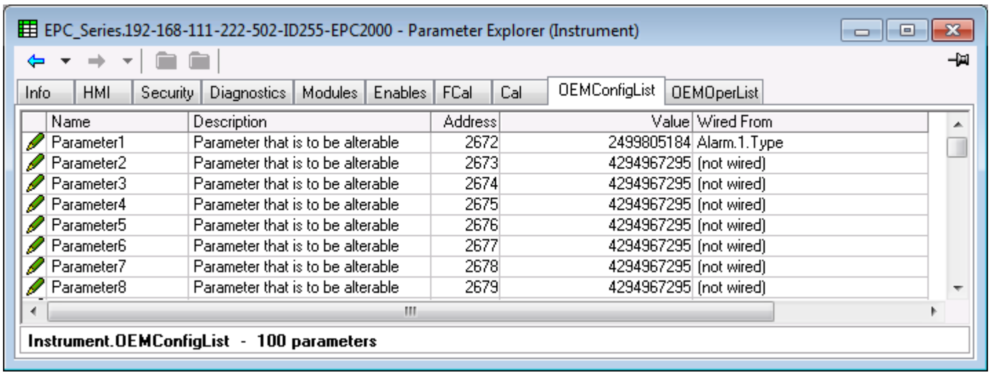
Parameter Name	Description	Available Values	Value Description
ID	User calibration identification	AI1 (0)	Analog input 1.
		AI2 (1)	**Not applicable to the EPC2000 Programmable Controller.**
		DCOP1 (2)	Analog output 1.
		DCOP1 (3)	**Not applicable to the EPC2000 Programmable Controller.**
		DCOP1 (4)	**Not applicable to the EPC2000 Programmable Controller.**
		CT (5)	Current transformer - **not applicable to the EPC2000 Programmable Controller.**
		RSP_MA (6)	**Not applicable to the EPC2000 Programmable Controller.**
		RSP_V (7)	**Not applicable to the EPC2000 Programmable Controller.**
Status	User calibration status	Factory (0)	Factory.
		Adjusted (1)	Adjusted.
Mode	User calibration mode	Idle (0)	Idle.
		Start (1)	Start calibration.
CalVal	User calibration value	This parameter only appears if MODE is equal to Low and High calibration point. For input user calibration this is the value the input is expected to be at the point of calibration. For output user calibration this is the externally measured output value at the point of calibration.	

Instrument.OEMConfigList

The OEMConfigList subclass allows the OEM to choose up to 100 configuration parameters which are to remain Read/Write while in Configuration mode and OEM Security is enabled (locked). In addition to these the following parameters are always writeable in configuration mode:

OEM Security Password Entry, Comms Configuration Password, Controller Coldstart.

The required parameters may be by dragged and dropped from a browser list (on the left hand side) into the Wired From cell in the ‘OEMConfigList’. Alternatively, double-click into the ‘WiredFrom’ cell and select the parameter from the pop-up list. These parameters are those chosen by the OEM which are to remain alterable when OEM Security is enabled and the controller is in Configuration access level.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

Name	Description	Address	Value	Wired From
Parameter1	Parameter that is to be alterable	2672	2499805184	Alarm.1.Type
Parameter2	Parameter that is to be alterable	2673	4294967295	(not wired)
Parameter3	Parameter that is to be alterable	2674	4294967295	(not wired)
Parameter4	Parameter that is to be alterable	2675	4294967295	(not wired)
Parameter5	Parameter that is to be alterable	2676	4294967295	(not wired)
Parameter6	Parameter that is to be alterable	2677	4294967295	(not wired)
Parameter7	Parameter that is to be alterable	2678	4294967295	(not wired)
Parameter8	Parameter that is to be alterable	2679	4294967295	(not wired)

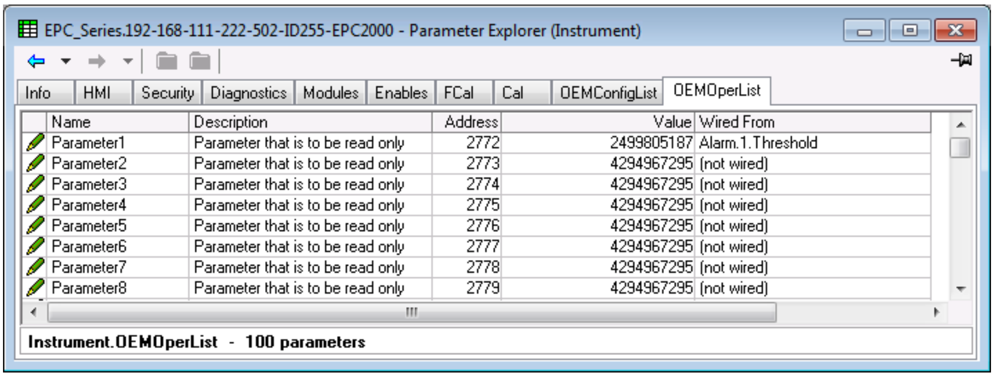
Instrument.OEMConfigList - 100 parameters

The view shows the first eight parameters of which Parameter 1 has been populated with a configuration parameter (Alarm 1 Type). Examples of configuration parameters include Alarm Types, Input Types, Range Hi/Lo, Modules Expected, etc.

When the OEM Status is Locked, this list is not shown. For further information on OEM Security, refer to the chapter, "OEM Security" on page 290 and the parameters "Instrument.Security" on page 95, "Instrument.OEMConfigList" on page 102 and "Instrument.OEMOperList" on page 103.

Instrument.OEMOperList

The OEMOperList subclass operates in the same way as the OEM Configuration list except the parameters selected are those which are available in Operator access level. Examples are programmer mode, alarm setting parameters, etc. The example below shows ‘Alarm 1 Threshold’ which is to be read only in Operator access level.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)
Info
HMI
Security
Diagnostics
Modules
Enables
FCal
Cal
OEMConfigList
OEMOperList

Name	Description	Address	Value	Wired From
Parameter1	Parameter that is to be read only	2772	2499805187	Alarm.1.Threshold
Parameter2	Parameter that is to be read only	2773	4294967295	(not wired)
Parameter3	Parameter that is to be read only	2774	4294967295	(not wired)
Parameter4	Parameter that is to be read only	2775	4294967295	(not wired)
Parameter5	Parameter that is to be read only	2776	4294967295	(not wired)
Parameter6	Parameter that is to be read only	2777	4294967295	(not wired)
Parameter7	Parameter that is to be read only	2778	4294967295	(not wired)
Parameter8	Parameter that is to be read only	2779	4294967295	(not wired)

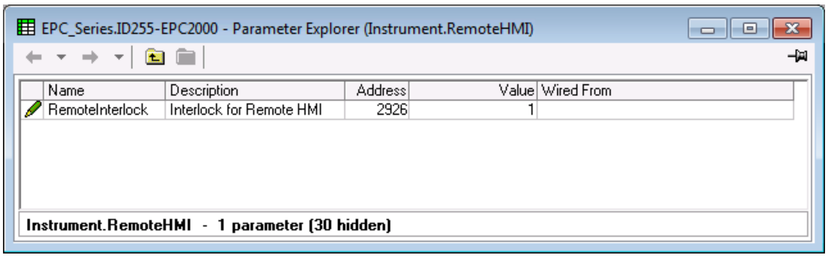
Instrument.OEMOperList - 100 parameters

The example shows the first eight of 100 parameters of which the first has been selected as ‘Alarm 1 Threshold’. This parameter is to be read only when OEM Security is enabled and the controller is in Operator access level.

When the OEM Status is Locked, this list is not shown. For further information on OEM Security, refer to the "OEM Security" on page 290.

Instrument.RemoteHMI

The RemoteHMI subclass provides a method for a remote HMI to bring the controller out of standby. This is useful to help prevent the outputs being driven before a remote HMI has finished starting up. The figure below shows the parameter and the table which follows details the parameter.


EPC_Series.ID255-EPC2000 - Parameter Explorer (Instrument.RemoteHMI)

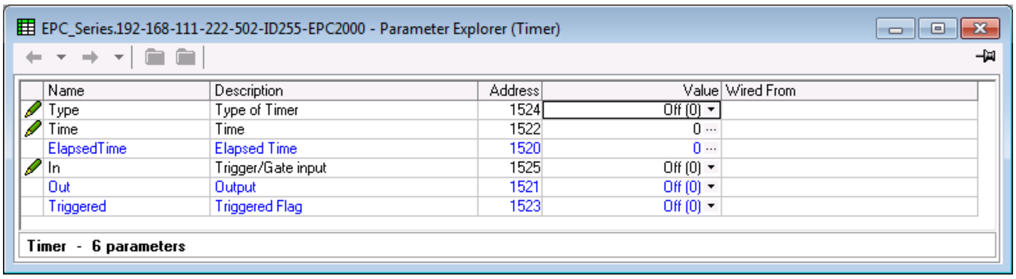
Name	Description	Address	Value	Wired From
RemoteInterlock	Interlock for Remote HMI	2926	1	

Instrument.RemoteHMI - 1 parameter (30 hidden)

Parameter Name	Description	Available Values	Value Description
RemoteInterlock	Interlock for Remote HMI	When wired to Instrument.Diagnostics.ForceStandby, a remote HMI can write to this parameter to bring the instrument out of standby.	

Timer

The EPC2000 Programmable Controller contains one timer function block for use in a user strategy and is only available if the Toolkit option has been ordered. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Timer)

	Name	Description	Address	Value	Wired From
✎	Type	Type of Timer	1524	Off (0)	
✎	Time	Time	1522	0	
	ElapsedTime	Elapsed Time	1520	0	
✎	In	Trigger/Gate input	1525	Off (0)	
	Out	Output	1521	Off (0)	
	Triggered	Triggered Flag	1523	Off (0)	

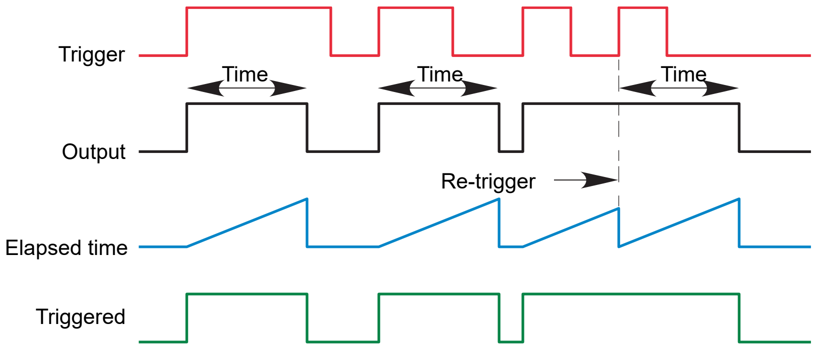
Timer - 6 parameters

Parameter Name	Description	Available Values	Value Description
Type	Type of Timer	Off (0)	Timer not activated. **Default: Off (0)**
		OnPulse (1)	Generates a fixed length pulse from an edge trigger.
		OnDelay (2)	Provides a delay between input trigger event and timer output.
		OneShot (3)	Simple oven timer which reduces to zero before switching off.
		MinOnTime (4)	Compressor timer so that the output remains ON for a time after the input signal has been removed.
Time	Time	Duration of the timer. For re-trigger timers this value is entered once and copied to the time remaining parameter whenever the timer starts. For pulse timers the time value itself is decremented. Range 00:00 to 999:59 minutes. **Default: 0**	
ElapsedTime	Elapsed Time	Elapsed time. Range 00:00 to 999:59 minutes	
In	Trigger/Gate input	Off (0)	Trigger/Gate input. **Default: Off (0)**
		On (1)	Turn On to start timing.
Out	Output	Off (0)	Timer output is off.
		On (1)	Timer output is on.
Triggered	Triggered Flag	This is a status output to indicate that the input to the timer has been detected.	
		Off (0)	Not timing.
		On (1)	Timer has been triggered and is operational.

Timer Modes

On Pulse Timer Mode

Output goes ON as soon as the trigger input goes active, and remains ON until the time period has elapsed. If the timer is re-triggered during the timing period, the timer restarts.



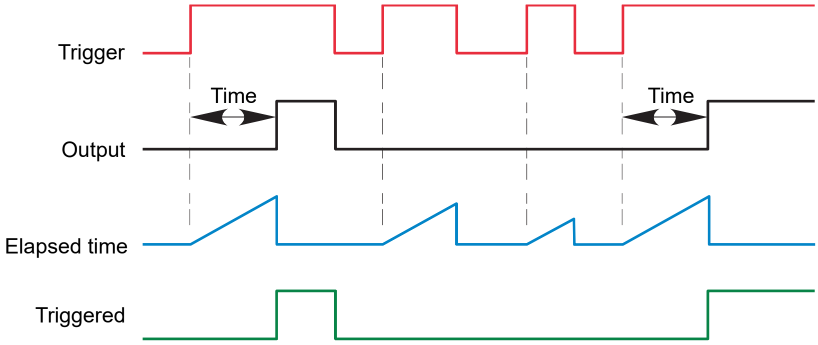
On Delay Timer Mode

Provides a delay between the trigger point and the timer output becoming active.

This type of timer is used to help ensure that the output is not set unless the input has been valid for a pre-determined period of time, thus acting as a kind of input filter.

Rules:

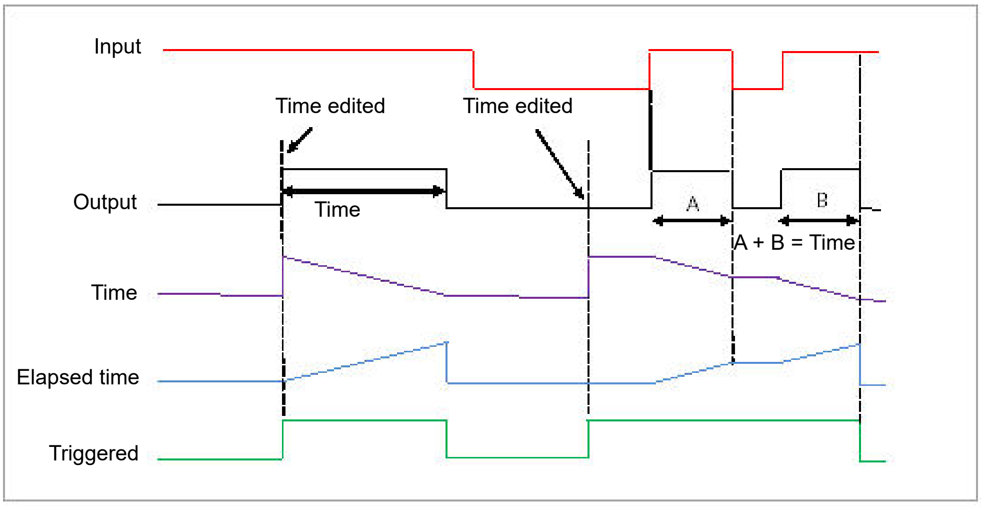
1. After the trigger goes active, the output switches on after the delay time has elapsed, and stays on until the trigger goes inactive.
2. If the trigger goes inactive before the delay time has elapsed, the output does not switch on.



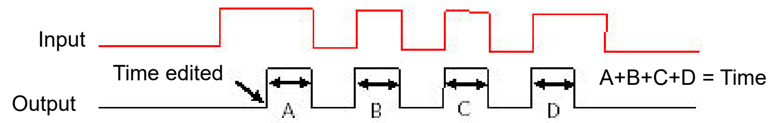
One Shot Timer Mode

- The Time value is decremented on each tick until it reaches zero. When the timer reaches zero the Output is cleared to OFF.
- The Time value can be edited at any instant to increase/decrease the duration of the ON time.
- Once set to zero, the Time is not reset to a previous value, it must be edited by the operator to start the next ON time.
- The input is used to gate the output. If the Input is set, the Time will count down to zero. If the Input is cleared to OFF, then the Time will Hold and the Output will switch OFF until the Input is next set.
- Since the Input is a Digital Wire, it is possible for the operator to NOT wire it, and set the Input value to ON which permanently enables the timer.
- The Triggered variable will be set to ON as soon as the Time is edited. It will reset when the Output is cleared to OFF.

The behavior under different conditions is shown below:



This diagram shows how the Input can be used to gate the Timer as a type of hold.



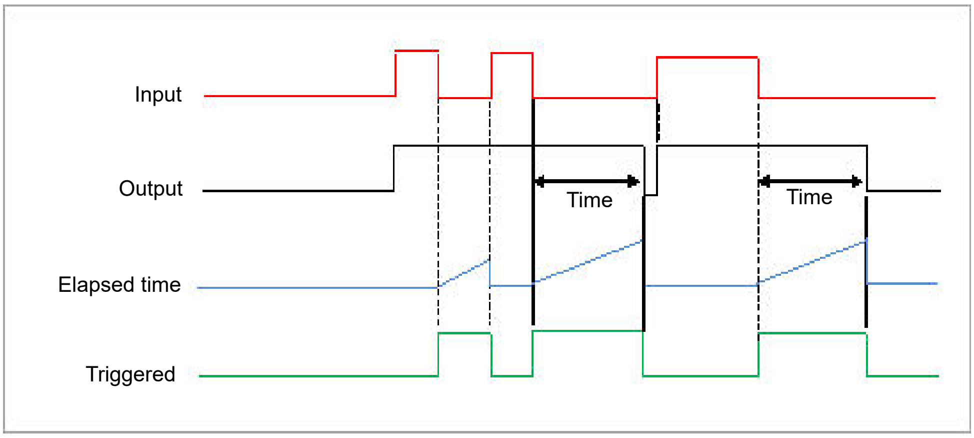
Minimum On Timer Mode or Compressor

The input goes active and remains on for a specified period after the input goes inactive.

It may be used, for example, to help ensure that a compressor is not cycled excessively.

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

The diagram illustrates the behavior of the timer under different input conditions:



Math2

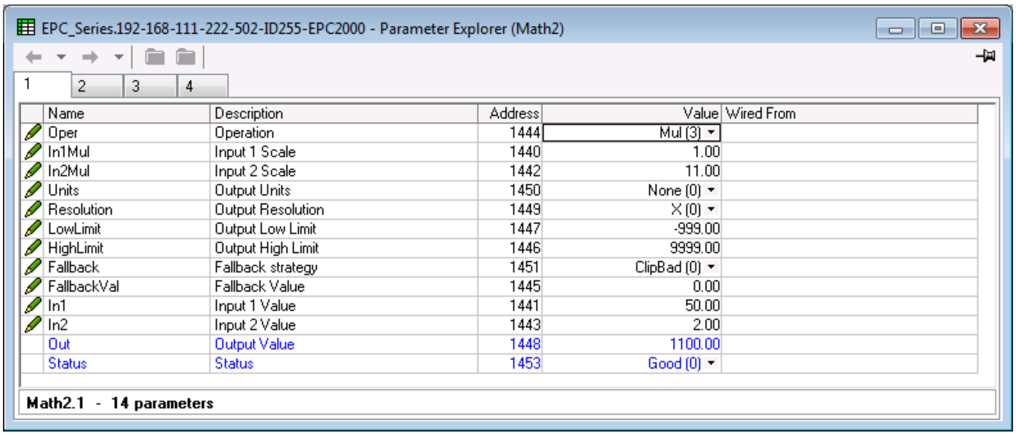
The Math2 category contains four math function blocks and are only available if the Toolkit option has been ordered.

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

The derived two-input control algorithm is as follows:

*Output = (In1Mul * In1) + (InMul2 * In2)*

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Math2)

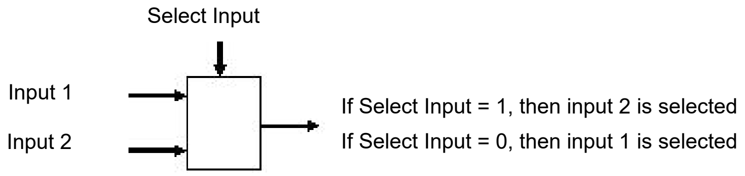
	Name	Description	Address	Value	Wired From
✎	Oper	Operation	1444	Mul (3) ▾	
✎	In1Mul	Input 1 Scale	1440	1.00	
✎	In2Mul	Input 2 Scale	1442	11.00	
✎	Units	Output Units	1450	None (0) ▾	
✎	Resolution	Output Resolution	1449	X (0) ▾	
✎	LowLimit	Output Low Limit	1447	-999.00	
✎	HighLimit	Output High Limit	1446	9999.00	
✎	Fallback	Fallback strategy	1451	ClipBad (0) ▾	
✎	FallbackVal	Fallback Value	1445	0.00	
✎	In1	Input 1 Value	1441	50.00	
✎	In2	Input 2 Value	1443	2.00	
	Out	Output Value	1448	1100.00	
	Status	Status	1453	Good (0) ▾	

Math2.1 - 14 parameters

Parameter Name	Parameter Description	Available Values	Value Description
Oper	Operation	Off (0)	The selected analog operator is turned off. **Default: Off**
		Add (1)	The output result is the addition of Input 1 and Input 2.
		Sub (2)	Subtract. The output result is the difference between Input 1 and Input 2 where Input 1 > Input 2.
		Mul (3)	Multiply. The output result is Input 1 multiplied by Input 2.
		Div (4)	Divide. The output result is Input 1 divided by Input 2.
		AbsDif (5)	Absolute Difference. The output result is the absolute difference between Input 1 and Input 2.
		SelMax (6)	Select Max. The output result is the maximum of Input 1 and Input 2.
		Sel Min (7)	Select Min. The output result is the minimum of Input 1 and Input 2.
		HotSwap (8)	HotSwap. Input 1 appears at the output provided input 1 is 'good'. If input 1 is 'bad' then input 2 value will appear at the output. An example of a bad input occurs during a sensor break condition.
		SmpHld (9)	Sample and Hold. Normally input 1 will be an analog value and input B will be digital. The output tracks input 1 when input 2 = 1 (Sample). The output will remain at the current value when input 2 = 0 (Hold). If input 2 is an analog value then any non zero value will be interpreted as 'Sample'.
		Power (10)	The output is the value at input 1 raised to the power of the value at input 2, i.e. Input $1^{\text{Input 2}}$.
		Sqrt (11)	Square Root. The output result is the square root of Input 1. Input 2 has no effect.
		Log (12)	The output is the logarithm (base 10) of Input 1. Input 2 has no effect.
		Ln (13)	The output is the logarithm (base n) of Input 1. Input 2 has no effect.
		Exp (14)	The output result is the exponential of Input 1. Input 2 has no effect.
		10_x (15)	The output result is 10 raised to the power of Input 1 value. i.e. $10^{\text{input 1}}$. Input 2 has no effect.
		Sel1 (51)	Select input is used to control which Analog Input is switched to the output of the Analog Operator. If the select input is true input 2 is switched through to the output. If false input 1 is switched through to the output. See "Select Input" on page 110.
In1Mul	Input 1 Scale	Input 1 scaling factor. **Default: 1.0**	
In2Mul	Input 2 Scale	Input 2 scaling factor. **Default: 1.0**	
Units	Output Units	**Default: C_F_K_Temp(1)**	
Resolution	Output Resolution	Resolution of the output value.	
		X (0)	No decimal places. **Default: nnnnn**
		X.X (1)	One decimal place.
		X.XX (2)	Two decimal places.
		X.XXX (3)	Three decimal places.
		X.XXXX (4)	Four decimal places.
LowLimit	Output Low Limit	To apply a low limit to the output. **Default: -999**	
HighLimit	Output High Limit	To apply a high limit to the output. **Default: 9999**	

Parameter Name	Parameter Description	Available Values	Value Description
Fallback	Fallback strategy	The fallback strategy will come into effect if the status of the input value is outside its expected range or if the input value is outside the range of Input Hi and Input Lo.	
		ClipBad (0)	If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and 'Status' is set to 'Bad'. If the input signal is within the limits, but its status is bad, the output is set to the 'Fallback' value. **Default: ClipBad (0)**
		ClipGood (1)	If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and 'Status' is set to 'Good'. If the input signal is within the limits, but its status is bad, the output is set to the 'Fallback' value.
		FallBad (2)	If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the 'Fallback' value, and the 'Status' is set to 'Bad'.
		FallGood (3)	If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the 'Fallback' value, and the 'Status' is set to 'Good'
		UpScaleBad (4)	If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the 'High Limit'.
		DownScaleBad (6)	If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the 'Low Limit'.
Fallback Val	Fallback Value	Defines (in accordance with Fallback) the output value when fallback strategy is active. **Default: 0**	
In1	Input 1 Value	Input 1 value (normally wired to an input source). Range −99999 to 99999 (decimal point depends on resolution).	
In2	Input 2 Value	Input 2 value (normally wired to an input source). Range −99999 to 99999 (decimal point depends on resolution).	
Out	Output Value	The analog value of the output, between high and low limits.	
Status	Status	This parameter is used in conjunction with Fallback to indicate the status of the operation. Typically, it is used to flag the status of the operation and used in conjunction with the fallback strategy. It may be used as an interlock for other operations. See section "Status" on page 94 for a list of enumerated values.	

Select Input

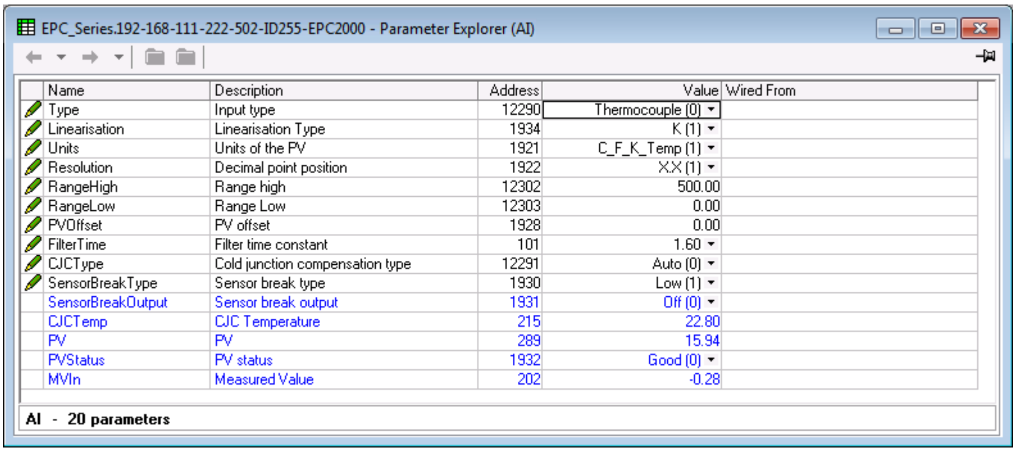


If Select Input = 1, then input 2 is selected

If Select Input = 0, then input 1 is selected

AI

The AI (Analog In) function block provides the ability to configure the input type and other characteristics of the EPC2000 Programmable Controller’s primary sensor input. Other input/output is controlled using the IO function blocks (refer to "IO" on page 114). The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (AI)

Name	Description	Address	Value	Wired From
Type	Input type	12290	Thermocouple (0)	
Linearisation	Linearisation Type	1934	K (1)	
Units	Units of the PV	1921	C_F_K_Temp (1)	
Resolution	Decimal point position	1922	X.X (1)	
RangeHigh	Range high	12302	500.00	
RangeLow	Range Low	12303	0.00	
PVOffset	PV offset	1928	0.00	
FilterTime	Filter time constant	101	1.60	
CJCType	Cold junction compensation type	12291	Auto (0)	
SensorBreakType	Sensor break type	1930	Low (1)	
SensorBreakOutput	Sensor break output	1931	Off (0)	
CJCTemp	CJC Temperature	215	22.80	
PV	PV	289	15.94	
PVStatus	PV status	1932	Good (0)	
MVIn	Measured Value	202	-0.28	

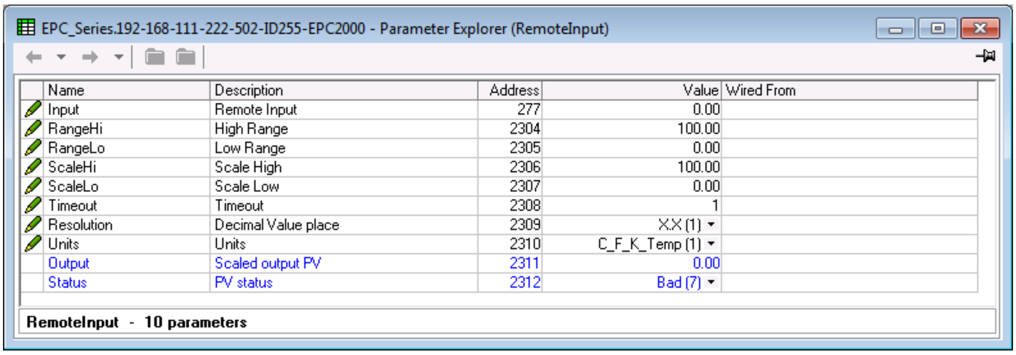
AI - 20 parameters

Parameter Name	Parameter Description	Available Values	Value Description
Type	Input type	Thermocouple (0)	Thermocouple. **Default: Thermocouple (0)**
		mV (1)	millivolts.
		V (2)	Volts.
		mA (3)	milliamps.
		RTD (4)	Platinum resistance thermometer.
		Zirconia (5)	Zirconia.
Linearisation	Linearisation Type	J (0)	Thermocouple type J.
		K (1)	Thermocouple type K. **Default: Type K (1)**
		L (2)	Thermocouple type L.
		R (3)	Thermocouple type R.
		B (4)	Thermocouple type B.
		N (5)	Thermocouple type N.
		T (6)	Thermocouple type T.
		S (7)	Thermocouple type S.
		Custom1 (8)	Custom linearization 1. To download special linearization tables see "To Load a Custom Linearization Table" on page 87.
		Custom2 (9)	Custom linearization 2. Two tables may be downloaded into the EPC2000 Programmable Controller.
Units	Units of the PV	**Default: C_F_K_Temp(1)**	
Resolution	Decimal point position	X (0)	Resolution of the input /output. No decimal places.
		X.X (1)	One decimal place. **Default: X.X (1)**
		X.XX (2)	Two decimal places.
		X.XXX (3)	Three decimal places.
		X.XXXX (4)	Four decimal places.
RangeHigh	Range high	Range high limit. Used to limit ranges of Thermocouple and RTD input types, and scale mV, V and mA inputs. AI2 also include Zirconia. **Default tc 500; mV 40; V 10; mA 20; RTD 500; Zirconia 2000**	

Parameter Name	Parameter Description	Available Values	Value Description
RangeLow	Range Low	Range low limit. Used to limit ranges of Thermocouple and RTD input types, and scale mV, V and mA inputs. AI2 also includes Zirconia. **Default tc 0; mV 0; V 0; mA 4; RTD 0; Zirconia 0**	
PVOffset	PV offset	0.0	A simple offset is provided to adjust the process variable by a fixed amount across its span. This can be used to compensate for known thermocouple and other tolerances which may exist in multi instrument installation so that all instruments read the same value. See also "Calibration using a Dry Block or Equivalent" on page 289 which describes the two point calibration adjust method. This may be used to apply a linear correction to the temperature reading. **Default: 0.0**
FilterTime	Filter time constant	0 to 60	Some industrial installations can cause EMI (Electromagnetic interference) to be introduced into the process measurement. This could be due, for example, to EMC or mechanical linkages. A filter is provided to reduce the frequency of EMI seen by the instrument. The effect of EMI can be reduced by increasing the filter time constant, but a compromise must be achieved as it could affect the closed loop response of the system. The larger the number, the slower the measured temperature will be to respond to fluctuations. **Default: 1.6s**
CJCType	Cold junction compensation type	Auto (0)	A thermocouple measures the temperature difference between the measuring junction (hot junction) and the reference junction (cold junction). Auto uses the measurement of the temperature made by the instrument where the thermocouple is connected to its rear terminals. **Default: Auto (0)**
		0degC (1)	The reference junction is held at a fixed known temperature of 0 degrees usually by an external ice point method.
		50degC (2)	The reference junction is held at a fixed known temperature of 50 degrees usually by an external hot box method.
		Off (3)	CJC is turned off. This could be used, for example, where a thermocouple measurement is made by an external transmitter which does not linearize the thermocouple curve.
SensorBreakType	Sensor break type	Off (0)	The controller continuously monitors the impedance of a transducer or sensor connected to the input. Off means sensor break is not detected.
		Low (1)	Sensor break is detected if impedance at the terminals is above a low threshold (typically between 3 to 5kΩ). **Default: Low (1)**
		High (2)	Sensor break is detected if impedance at the terminals is above a high threshold (typically between 12 to 20kΩ).
SensorBreakOutput	Sensor break output	Off (0)	No sensor break detected.
		On (1)	Sensor break detected. If the sensor break requires to activate a soft alarm the sensor break output parameter can be wired to a Digital High Alarm. (see section "Example 1: To Wire an Alarm" on page 81.
CJCTemp	CJC Temperature	The CJC temperature is a measure of the temperature at the instrument terminals. It is relevant only for thermocouple inputs and is provided as a diagnostic aid.	
PV	PV	Process value is the displayed value on the instrument, usually the measured temperature when the instrument is controlling a temperature loop.	
PVStatus	PV status	The state of the PV is continuously monitored. See section "Status" on page 94 for a list of enumerated values.	
MVIn	Measured Value	This is the measured value in units of mV or ohms, dependent upon input type. The value measured at the rear terminals can be useful as a diagnostic aid to determine if the thermocouple or linear input sensor is wired correctly.	

RemoteInput

The Remote Input function block scales an input from a Modbus remote master between a specified range. The figure below shows the parameters for the first instance of the Remote Input function block and the table which follows details each parameter. Two instances of the Remote Input function block have been implemented.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (RemoteInput)

Name	Description	Address	Value	Wired From
Input	Remote Input	277	0.00	
RangeHi	High Range	2304	100.00	
RangeLo	Low Range	2305	0.00	
ScaleHi	Scale High	2306	100.00	
ScaleLo	Scale Low	2307	0.00	
Timeout	Timeout	2308	1	
Resolution	Decimal Value place	2309	X.X (1)	
Units	Units	2310	C_F_K_Temp (1)	
Output	Scaled output PV	2311	0.00	
Status	PV status	2312	Bad (7)	

RemoteInput - 10 parameters

Parameter Name	Description	Available Values	Value Description
Input	Remote Input	This parameter can be written to via a remote master. The Modbus addresses to be written to by an external master are: RemoteInput.1.Input: 277. RemoteInput.2.Input: 2928.	
RangeHi	High Range	Maximum value of the input. **Default: 100**	
RangeLo	Low Range	Minimum value of the input. **Default: 0**	
ScaleHi	Scale High	The maximum value of the scaled output PV. **Default: 100**	
ScaleLo	Scale Low	The minimum value of the scaled output PV. **Default: 0**	
Timeout	Timeout	This is the period in which the input has to be written to (in seconds). If this period is exceeded the output PV status will be set to Bad. If this period is set to 0, the timeout strategy is disabled. **Default: 1**	
Resolution	Decimal Value place	X (0)	Resolution of the input /output. No decimal places.
		X.X (1)	One decimal place. **Default: X.X (1)**
		X.XX (2)	Two decimal places.
		X.XXX (3)	Three decimal places.
		X.XXXX (4)	Four decimal places.
Units	Units	**Default: C_F_K_Temp(1)**	
Output	Scaled output PV	The output PV that has been linearly scaled Range High to Scale High and Range Low to Scale Low.	
Status	PV status	Status of the output PV. See section "Status" on page 94 for a list of enumerated values.	

IO

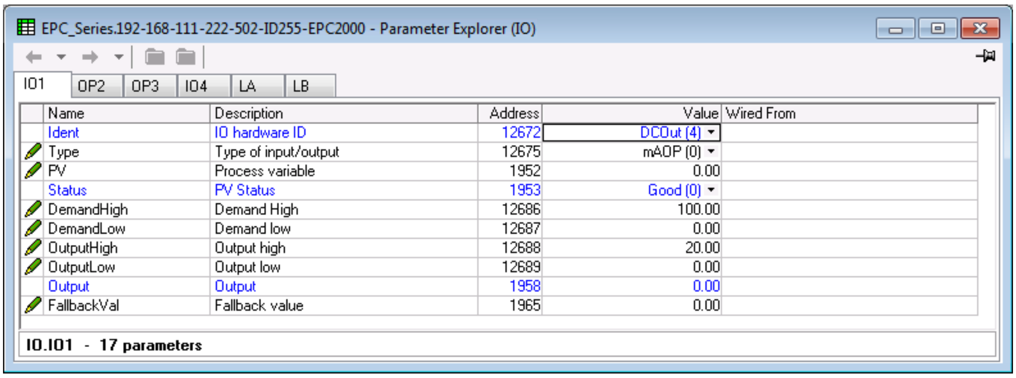
The IO category contains function blocks for the EPC2000 Programmable Controller Input/Output (I/O) hardware. Because there are configurable options at the time of ordering, the actual I/O can be different. The I/O options are as follows:

- IO1 can be either an analog output or a logic output/contact input. This is determined at the point of ordering.
- OP2 is a form A (normally open) relay.
- OP3 is a form C (change-over) relay.
- LA is a digital (contact) input, also known an digital input 1 (DI1).
- LB is a digital (contact) input, also known as digital input 2 (DI2).

The EPC2000 Programmable Controller’s main analog sensor input is controlled using the AI (Analog In) function block (refer "AI" on page 111).

IO.IO1

The IO1 subclass controls the analog output (DC output) or logic/contact input (Logic I/O), a customer option at the time of ordering, at terminal contacts 1A and 1B. The figure below shows the parameters and the table which follows details each parameter. Depending on the hardware configuration and software options, not all the following parameters will be available at any one time.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (IO)

Name	Description	Address	Value	Wired From
Ident	IO hardware ID	12672	DCOut (4)	
Type	Type of input/output	12675	mAOP (0)	
PV	Process variable	1952	0.00	
Status	PV Status	1953	Good (0)	
DemandHigh	Demand High	12686	100.00	
DemandLow	Demand low	12687	0.00	
OutputHigh	Output high	12688	20.00	
OutputLow	Output low	12689	0.00	
Output	Output	1958	0.00	
FallbackVal	Fallback value	1965	0.00	

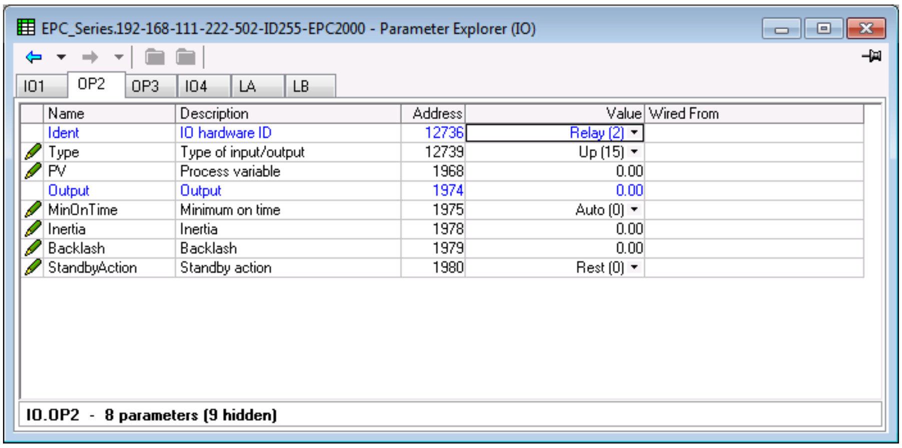
IO.IO1 - 17 parameters

Parameter Name	Description	Available Values	Value Description
Ident	IO hardware ID	This displays the type of IO hardware fitted. Possibilities are:	
		None (0)	No I/O hardware.
		LogicIO (1)	Logic input/output.
		Relay (2)	Relay.
		Triac (3)	Triac (**not applicable to the EPC2000 Programmable Controller**)
		DCOut (4)	DC output.
		LogicIP (5)	Logic input.
Type	Type of input/output	mAOP (0)	mA output (only applicable when ordered as a DC Output).
		VOP (1)	Voltage output (only applicable when ordered as a DC Output).
		LogicIn (5)	Logic input (only applicable when ordered as a Logic I/O).
		OnOff (10)	On/Off output (only applicable when ordered as a Logic I/O).
		TPO (11)	Time proportioning output (only applicable when ordered as a Logic I/O).
		Up (15)	Valve raise (only applicable when ordered with as a Logic I/O).
PV	Process Variable	For an Input type: The measured process variable. For an Output type: The demanded output value.	

Parameter Name	Description	Available Values	Value Description
Status	PV Status	The state of the PV is continuously monitored. Only applicable when ordered as a DC Output. See section "Status" on page 94 for a list of enumerated values.	
DemandHigh	Demand High	Percentage PID demand value giving maximum output - "OUT.H" - Allows for "Output Splitting" Only applicable when ordered as a DC Output. **Default: 100.0**	
DemandLow	Demand low	Percentage PID demand value giving minimum output - "OUT.L" - Allows for "Output Splitting" Only applicable when ordered as a DC Output. **Default: 0.0**	
OutputHigh	Output high	The maximum average output power that can be supplied from this output - Allows for "Output Splitting" Only applicable when ordered as a DC Output. **Default: 100% for TPO; 20 for mA; 10 for V** i.e. the highest possible value for selected type.	
OutputLow	Output low	The minimum average output power that can be supplied from this output - Allows for "Output Splitting" Only applicable when ordered as a DC Output. **Default: 0**	
Output	Output	For Digital Output Types. A value of 0 indicates that the output is low (relay de-energized) A value of 1 indicates that the output is high (relay energized). For DC Output Types. This is the physical output value after the PV has been mapped via the demand range parameters onto the output range.	
FallbackVal	Fallback value	Fallback value which is to be outputted when the status is BAD, **Default: to the value of OUT.L.** Only applicable when ordered as a DC Output.	
Sense	Sense	Sense of the input or output.	
		Normal (0)	Normal (non-inverted) input or output.
		Invert (1)	Inverted input or output.

IO.OP2

The OP2 subclass controls the form A relay (normally open) available at terminal contacts 2A and 2B. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (IO)

Name	Description	Address	Value	Wired From
Ident	IO hardware ID	12736	Relay (2)	
Type	Type of input/output	12739	Up (15)	
PV	Process variable	1968	0.00	
Output	Output	1974	0.00	
MinOnTime	Minimum on time	1975	Auto (0)	
Inertia	Inertia	1978	0.00	
Backlash	Backlash	1979	0.00	
StandbyAction	Standby action	1980	Rest (0)	

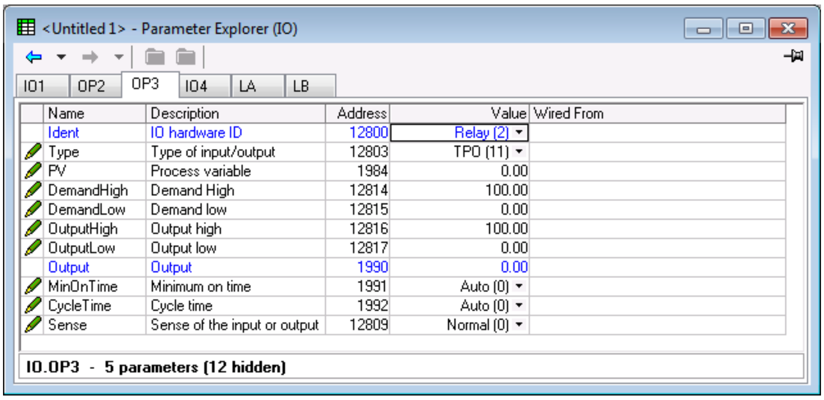
IO.OP2 - 8 parameters (9 hidden)

Parameter Name	Description	Available Values	Value Description
Ident	IO hardware ID	This displays the type of IO hardware fitted. Possibilities are:	
		None (0)	No IO hardware.
		LogicIO (1)	Logic input/output (**not applicable to the EPC2000 Programmable Controller**).
		Relay (2)	Relay.
		Triac (3)	Triac (**not applicable to the EPC2000 Programmable Controller**).
		DCOut (4)	DC output (**not applicable to the EPC2000 Programmable Controller**).
		LogicIP (5)	Logic input (**not applicable to the EPC2000 Programmable Controller**).

Parameter Name	Description	Available Values	Value Description
Type	Type of input/output	OnOff (10)	On/Off output.
		TPO (11)	Time proportioning output.
		Up (15)	Valve raise.
		Down (16)	Valve lower.
PV	Process Variable	The demanded output value.	
Output	Output	A value of 0 indicates that the output is low (relay de-energized). A value of 1 indicates that the output is high (relay energized).	
MinOnTime	Minimum on time	0	Minimum pulse time in seconds. This value sets the minimum duration between any two switching events. Although it is named 'MinOnTime', it applies to both on and off pulses equally. A contactor datasheet will often specify the minimum pulse time that will help to ensure correct energizing and de-energizing of the contactor. This may be the lowest value you should consider using as a MinOnTime. Auto(0) - Automatically sets the minimum on time for the output hardware as 1s. Alternatively, a value may be set manually but it should be noted that this value will be clipped if it is below the minimum permissible value for the relay. **Default: Auto**
Inertia	Inertia	Time taken in seconds for the valve motor to stop after power is removed. 0.0 to 30.0 seconds. Applies to valve position outputs only. IO1+OP2 or OP2+OP3 can be configured as a valve position pair. **Default: 0.0**	
Backlash	Backlash	Time in seconds to take up any backlash in the valve actuator linkage. 0.0 to 30.0 seconds. Applies to valve position outputs only. **Default: 0.0**	
StandbyAction	Standby action		Determines the valve positioning output action (Rest, Raise, Lower) when the instrument is in Standby Mode.
		0	The valve will remain in the current position. **Default: Reset**
		1	The valve will open. Applies to IO1.
		2	The valve will close. Applies to IO2.
			Valve position works across pairs of outputs: If IO1 is UP then OP2 is DOWN. If IO2 is UP then OP3 is DOWN. No further combinations are valid on the EPC2000 Programmable Controller.

IO.OP3

The OP3 subclass controls the form C relay (changeover) available at terminal contacts 3A, 3B and 3C. The figure below shows the parameters and the table which follows details each parameter. The number of parameters is dependent upon the Type parameter.



	Name	Description	Address	Value	Wired From
	Ident	IO hardware ID	12800	Relay (2)	
✎	Type	Type of input/output	12803	TPO (11)	
✎	PV	Process variable	1984	0.00	
✎	DemandHigh	Demand High	12814	100.00	
✎	DemandLow	Demand low	12815	0.00	
✎	OutputHigh	Output high	12816	100.00	
✎	OutputLow	Output low	12817	0.00	
	Output	Output	1990	0.00	
✎	MinOnTime	Minimum on time	1991	Auto (0)	
✎	CycleTime	Cycle time	1992	Auto (0)	
✎	Sense	Sense of the input or output	12809	Normal (0)	

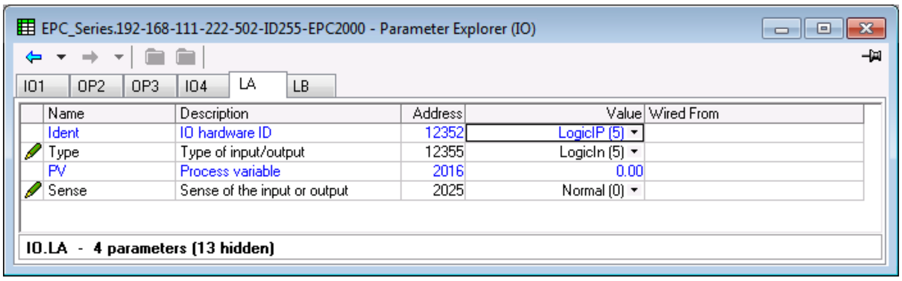
IO.OP3 - 5 parameters (12 hidden)

<table>
<tr><th>Parameter Name</th><th>Description</th><th>Available Values</th><th>Value Description</th></tr>
<tr><td rowspan="7">Ident</td><td rowspan="7">IO hardware ID</td><td colspan="2">This displays the type of IO hardware fitted. Choices are:</td></tr>
<tr><td>None (0)</td><td>No IO hardware.</td></tr>
<tr><td>LogicIO (1)</td><td>Logic input/output (<b>not applicable to the EPC2000 Programmable Controller</b>).</td></tr>
<tr><td>Relay (2)</td><td>Relay.</td></tr>
<tr><td>Triac (3)</td><td>Triac (<b>not applicable to the EPC2000 Programmable Controller</b>).</td></tr>
<tr><td>DCOut (4)</td><td>DC output (<b>not applicable to the EPC2000 Programmable Controller</b>).applicable to</td></tr>
<tr><td>LogicIP (5)</td><td>Logic input (<b>not applicable to the EPC2000 Programmable Controller</b>).</td></tr>
<tr><td rowspan="3">Type</td><td rowspan="3">Type of input/output</td><td>OnOff (10)</td><td>On/Off output.</td></tr>
<tr><td>TPO (11)</td><td>Time proportioning output.</td></tr>
<tr><td>Down (16)</td><td>Valve lower (only if IO.OP2 parameter configured as ‘Up’).</td></tr>
<tr><td>PV</td><td>Process Variable</td><td colspan="2">The demanded output value.</td></tr>
<tr><td>DemandHigh</td><td>Demand High</td><td colspan="2">Percentage PID demand value giving maximum output - "OUT.H" - Allows for "Output Splitting" Only applicable when ordered as a DC Output.
<b>Default: 100.0</b></td></tr>
<tr><td>DemandLow</td><td>Demand low</td><td colspan="2">Percentage PID demand value giving minimum output - "OUT.L" - Allows for "Output Splitting" Only applicable when ordered as a DC Output.
<b>Default: 0.0</b></td></tr>
<tr><td>OutputHigh</td><td>Output high</td><td colspan="2">The maximum average output power that can be supplied from this output - Allows for "Output Splitting" Only applicable when ordered as a DC Output.
<b>Default: 100% for TPO; 20 for mA; 10 for V</b> i.e. the highest possible value for selected type.</td></tr>
<tr><td>OutputLow</td><td>Output low</td><td colspan="2">The minimum average output power that can be supplied from this output - Allows for "Output Splitting" Only applicable when ordered as a DC Output.
<b>Default: 0</b></td></tr>
<tr><td>Output</td><td>Output</td><td colspan="2">A value of 0 indicates that the output is low (relay de-energized) A value of 1 indicates that the output is high (relay energized).</td></tr>
<tr><td>MinOnTime</td><td>Minimum on time</td><td>0</td><td>Minimum pulse time in seconds. This value sets the minimum duration between any two switching events. Although it is named 'MinOnTime', it applies to both on and off pulses equally.
A contactor datasheet will often specify the minimum pulse time that will help to ensure correct energizing and de-energizing of the contactor. This may be the lowest value you should consider using as a MinOnTime.
Auto(0) - Automatically sets the minimum on time for the output hardware as 1s.
Alternatively, a value may be set manually but it should be noted that this value will be clipped if it is below the minimum permissible value for the relay.
<b>Default: Auto</b></td></tr>
<tr><td>CycleTime</td><td>Cycle time</td><td colspan="2">The time proportioning output (TPO) cycle time in seconds. It is defined as the period of time between output repetitions.
When this parameter is Auto(0), which is the default setting, the TPO algorithm will run in a so-called Constant Ripple Mode. Under this regime, the cycle time will be automatically and continuously adjusted depending on the output demand. This is in effort to maintain the amount of ripple in the process at an approximately constant amplitude. The benefit of this is that actuations are reduced on average, which can increase the lifetime of contactors and relays. As suggested, a demand of 50% will produce the shortest cycle time of 4*MinOnTime, and the cycle time is extended the further the demand moves away from 50%. You should therefore choose a MinOnTime that gives an appropriate minimum cycle time.
Alternatively, you can set a value of cycle time directly. When a value is set, the algorithm will run in a so-called Constant Cycle Time Mode. Under this regime, the algorithm will try to keep the cycle time constant, assuming constant demand. Do note that the cycle time will be extended if the demand is such that the cycle time cannot be achieved without violating the MinOnTime. In this case, the effective cycle time will extended allowing the MinOnTime and the demand to be achieved.
<b>Default: Auto (0)</b></td></tr>
</table>

Parameter Name	Description	Available Values	Value Description
Sense	Sense	Sense of the input or output.	
		Normal (0)	Normal (non-inverted) input or output.
		Invert (1)	Inverted input or output.

IO.LA and IO.LB

The LA and LB subclasses control the contact digital input 1 available at terminal contacts LA and LC and contact digital input 2 available at terminal contacts LB and LC respectively. The figure below shows the parameters and the table which follows details each parameter.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (IO)
IO1
OP2
OP3
IO4
LA
LB

Name	Description	Address	Value	Wired From
Ident	IO hardware ID	12352	LogicIP (5)	
Type	Type of input/output	12355	LogicIn (5)	
PV	Process variable	2016	0.00	
Sense	Sense of the input or output	2025	Normal (0)	

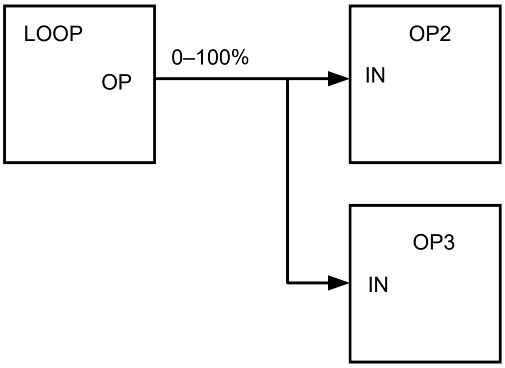
IO.LA - 4 parameters (13 hidden)

Parameter Name	Description	Available Values	Value Description
Ident	IO hardware ID	This displays the type of IO hardware fitted. Choices are:	
		None (0)	No IO hardware.
		LogicIO (1)	Logic input/output (**not applicable to the EPC2000 Programmable Controller**).
		Relay (2)	Relay (**not applicable to the EPC2000 Programmable Controller**).
		Triac (3)	Triac (**not applicable to the EPC2000 Programmable Controller**).
		DCOut (4)	DC output (**not applicable to the EPC2000 Programmable Controller**).
		LogicIP (5)	Logic input.
Type	Type of input/output	LogicIn (5)	Logic input.
PV	Process Variable	The demanded output value.	
Sense	Sense of the input	0	The input is active when the input is = 1. **Default: Normal**
		1	The input is active when the input is = 0.

Output Splitting

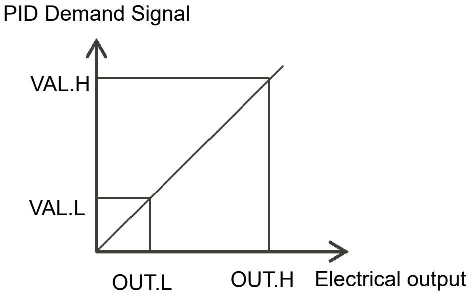
Output splitting is the process of having more than one output being driven from a single control loop. For this to be possible the single loop output signal is divided between two output channels.

This splitting of outputs is not done as a part of the control loop but rather as a part of the output blocks.



Functionality

- The control loop is not affected by the use of output splitting, it will still give its output in the form of a 0–100% value.
- Each output block can be tailored individually in terms of turn on / off points and percentage power output.
- The output from the loop is "wired" to the inputs of two output blocks.
- Each output block has a "ValHigh" and "ValLow" parameter. These values represent the PID demand percentage giving maximum and minimum output power respectively.
- Each output block also has a "OutHigh" and "OutLow" parameter. The values of which determine the percentage limits of output power.
- The relationship between output power and input value can be seen in the graph below:



Cycle Time and Minimum OnTime Algorithms

The ‘Cycle Time’ algorithm and the ‘Min OnTime’ algorithm are mutually exclusive and provide compatibility with existing controller systems. Both algorithms apply to time proportioning outputs only and are not shown for on/off control.

A fixed cycle time allows the output to switch on and off within the time period set by the parameter. For example, for a cycle time of 20 seconds, 25% power demand would turn the output on for 5 seconds and off for 15 seconds, 50% power demand would turn the output on and off for 10 seconds, for 75% power demand the output is on for 15 seconds and off for 5 seconds.

Fixed cycle time may be preferred when driving mechanical devices such as refrigeration compressors.

The ‘Min OnTime’ is described in the IO table in the previous section.

If the control device is a relay or contactor the minimum on time should be set greater than 10 seconds (for example) to prolong relay life. By way of illustration, for a setting of 10 seconds the relay will switch (approximately) as shown in the table below:

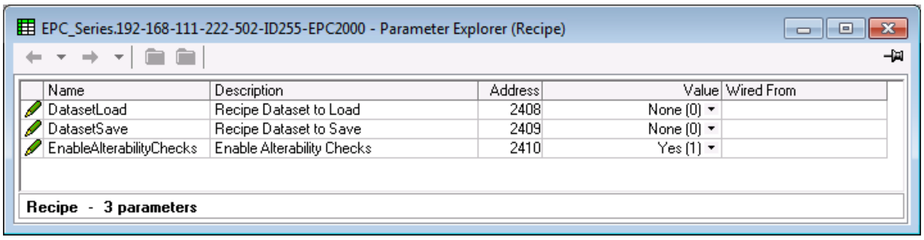
Power demand	Relay ON time	Relay OFF time
10%	10	100
25%	13	39
50%	20	20
75%	39	13
90%	100	10

The Minimum OnTime algorithm is often preferred for control of switching devices using triac, logic or relay outputs in a temperature control application. It also applies to valve position outputs.

Note: Consideration should be given to the number of operations which the relay is expected to endure during its lifetime. See section "Relay Electrical Endurance" on page 303.

Recipe

A recipe is a list of parameters whose values can be captured and stored in a dataset. This dataset can then be loaded into the controller at any time to restore the recipe parameters, thus providing a means of altering the configuration of an instrument in a single operation. A maximum of five datasets are supported, referenced by name, defaulted to be the dataset number i.e. 1…5. The Recipe function block allows the selection of a recipe set to be loaded or saved. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Recipe)

Name	Description	Address	Value	Wired From
DatasetLoad	Recipe Dataset to Load	2408	None (0)	
DatasetSave	Recipe Dataset to Save	2409	None (0)	
EnableAlterabilityChecks	Enable Alterability Checks	2410	Yes (1)	

Recipe - 3 parameters

Parameter Name	Description	Available Values	Value Description
DatasetLoad	Recipe Dataset to Load	None (0)	Selects which recipe dataset to load. Once selected, the values stored in the dataset will be copied back over the active parameters. **Default: None**
		Dataset1 (1) Dataset2 (2) Dataset3 (3) Dataset4 (4) Dataset5 (5)	Dataset 1 to 5.
DatasetSave	Recipe Dataset to Save	None (0)	Selects in which of the five recipe datasets to store the current active parameters. When selected, this parameter initiates a snapshot of the current parameter set into the selected recipe dataset.
		Dataset1 (1) Dataset2 (2) Dataset3 (3) Dataset4 (4) Dataset5 (5)	Dataset 1 to 5.
EnableAlterabilityChecks	Enable Alterability Checks	Yes (1)	Enabled. Set to ‘Yes’ to check all parameters can be written in the current mode before loading a recipe dataset. **Default: Yes (1)**
		No (0)	Disabled. Set to ‘No’ to write all parameters regardless of their ‘config-only’ status. See Note below.

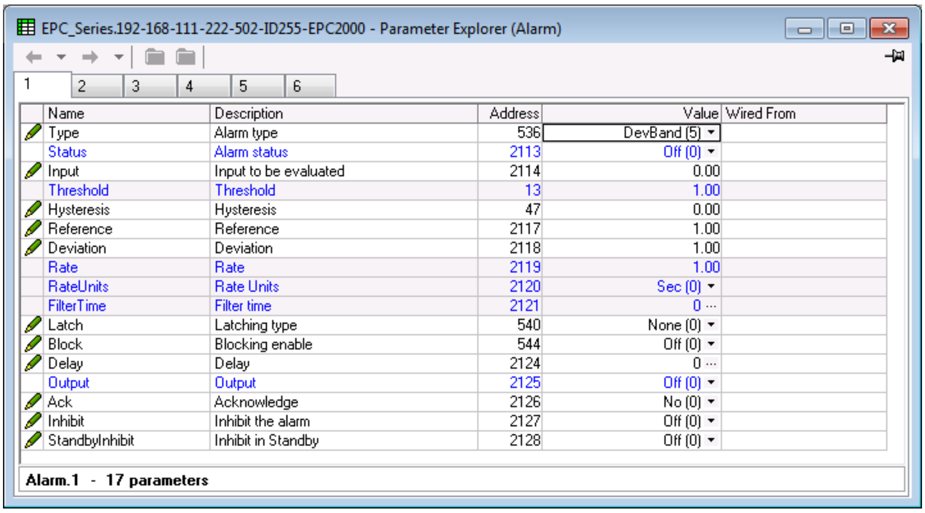
Note: Changing configurations and certain parameters whilst in Operator mode can cause disturbances in the process and, therefore, by default, a dataset will not be loaded (no parameters written to) if a parameter contained in the recipe is not writeable in operator mode. To cater for users who require the loading to operate in a similar manner to the 3200 controller (no parameter checking), this functionality can be disabled. However, to reduce disturbances in the process, whilst loading a dataset which contains configuration parameters the Instrument will be forced into standby whilst the dataset loads.

If the recipe load cannot be completed for any reason (values are invalid or out of range), the instrument will be half configured and will put itself into Standby. This will continue after a power cycle.

There is no default list of parameters for the EPC2000. The parameters required to be held in recipe are defined using iTools, see "Recipes" on page 83.

Alarm

The Alarm category provides access to the configuration of up six alarm function blocks. See also the Chapter "Alarms" on page 179 which describe alarm features. All alarms (1-6) are configured the same way as each other. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Alarm)

	Name	Description	Address	Value	Wired From
	Type	Alarm type	536	DevBand (5)	
	Status	Alarm status	2113	Off (0)	
	Input	Input to be evaluated	2114	0.00	
	Threshold	Threshold	13	1.00	
	Hysteresis	Hysteresis	47	0.00	
	Reference	Reference	2117	1.00	
	Deviation	Deviation	2118	1.00	
	Rate	Rate	2119	1.00	
	RateUnits	Rate Units	2120	Sec (0)	
	FilterTime	Filter time	2121	0 ...	
	Latch	Latching type	540	None (0)	
	Block	Blocking enable	544	Off (0)	
	Delay	Delay	2124	0 ...	
	Output	Output	2125	Off (0)	
	Ack	Acknowledge	2126	No (0)	
	Inhibit	Inhibit the alarm	2127	Off (0)	
	StandbyInhibit	Inhibit in Standby	2128	Off (0)	

Alarm.1 - 17 parameters

Parameter Name	Description	Available Values	Value Description
Type	Alarm type	Off (0)	The alarm is disabled. **Default: Off (0)**
		AbsHi (1)	The alarm is triggered when the input value becomes greater than the threshold.
		AbsLo (2)	The alarm is triggered when the input becomes less than the threshold.
		DevHi (3)3	The alarm is triggered when the input becomes higher than the reference by the amount of the deviation.
		DevLo (4)	The alarm is triggered when the input becomes lower than the reference by the amount of the deviation.
		DevBand (5)	The alarm is triggered when the input differs from the reference by the amount of the deviation.
		RRoC (6)	The alarm is triggered when the input changes positively by more than a specified amount within a specified period (second, minute, hour). It will remain active until the input value's positive rate of change falls below the specified rate.
		FRoC (7)	The alarm is triggered when the input changes negatively by more than a specified amount within a specified period (second, minute, hour). It will remain active until the input value's negative rate of change falls below the specified rate.
		DigHi (8)	The alarm is triggered when the input is equivalent to a Boolean '1' i.e. >=0.5.
		DigLo (9)	The alarm is triggered when the input is equivalent to a Boolean '0' i.e. < 0.5.
Status	Alarm status	This shows that the alarm is Off, Active, InactiveNotAcked or ActiveNotAcked.	
		Off (0)	No alarm. Shows ‘Off’ when the alarm is inhibited.
		Active (1)	Active. The alarm is still present but has been acknowledged.
		InactiveNotAckd (2)	Inactive Not Acknowledged means that the alarm trigger source has returned to a non-alarm state, but the alarm is still active because it has not been acknowledged. applies to ‘Auto’ and ‘Manual’ latching alarms only,
		ActiveNotAckd (3)	Active not acknowledged means that the source is still active and the alarm has not been acknowledged.
Input	Input to be evaluated	The input value being monitored.	

Parameter Name	Description	Available Values	Value Description
Threshold	Threshold	For absolute alarms only, this is the trip point for the alarm. For absolute high alarms, if the input value exceeds the threshold value, then the alarm becomes active, and remains active until the input falls below the value (threshold − hysteresis). For absolute low alarms, if the input falls below the threshold value, then the alarm becomes active and remains active until the input rises above (Threshold + Hysteresis). **Default: 1.0**	
Hysteresis	Hysteresis	Hysteresis is the difference between the point at which the alarm switches 'ON' and the point at which it switches 'OFF'. It is used to provide a definite indication of the alarm condition and to help prevent alarm relay chatter. A value of 0.0 disables hysteresis. **Default: 0.0**	
Reference	Reference	For deviation alarms only, this provides a 'centre point' for the deviation band. For 'deviation high' alarms, the alarm becomes active if the input rises above the value (Reference + Deviation) and remains active until the input falls below (Reference + Deviation - Hysteresis). For 'deviation low' alarms, the alarm becomes active if the input falls below the value (Reference - Deviation) and remains active until the input rises above (Reference - Deviation + Hysteresis). For 'deviation band' alarms, the alarm is active whenever the input lies outside the value (Reference ± Deviation) and remains active until the input returns to within the band, minus or plus Hysteresis as appropriate. **Default: 1.0** Note: If blocking is enabled, changing this parameter will activate alarm blocking. This includes when being wired to. You must ensure that the source value is not noisy, otherwise the alarm will be blocked.Range −19999 to 99999	
Deviation	Deviation	Used in deviation alarms. The deviation value added to or subtracted from the reference value at which the input is evaluated against. Range −19999 to 99999. **Default: 1.0**	
Rate	Rate Units	For rate-of-change alarms only. The alarm becomes active if the input rises (Rising ROC) or falls (Falling ROC) at a rate that is greater than the specified 'Rate' per 'Rate Unit'. The alarm remains active until the rate of change falls below the set 'Rate'. Range −19999 to 99999 **Default: 1.0**	
RateUnits	Rate units	Sec (0)	The rate units, used in rate of change alarms, selects the units for the rate parameter in seconds, minutes or hours. **Default: Seconds**
		Min (1)	
		Hr (2)	
FilterTime	Filter time	For rate-of-change alarms only. This allows a filter period (for the input) to be entered to reduce nuisance trips due to Electromagnetic interference (EMI), or if the rate of change is hovering around the trip value. Range 0.0 to 9999.9 seconds. **Default: 0.0**	
Latch	Latching type	None (0)	No latching methodology i.e. when the alarm condition is removed the alarm will become inactive without being acknowledged. **Default: None (0)**
		Auto (1)	The alarm will remain active until the alarm condition has been removed and the alarm has been acknowledged. The alarm can be acknowledged at any time after the alarm has become active.
		Manual (2)	The alarm will remain active until the alarm condition has been removed and the alarm has been acknowledged. The alarm can only be acknowledged after the alarm condition has been removed.
		Event (3)	Same as a non-latching alarm except the alarm is used as a trigger and therefore will not be announced.
Block	Blocking enable	Off (0)	Blocking disabled. **Default: Off (0)**
		On (1)	Alarms with 'Block' set to 'On' are inhibited until the monitored value has entered the working condition after a start-up. This helps to prevent such alarms from becoming active whilst the process is brought into control. If a latching alarm is not acknowledged then the alarm is reasserted (not blocked), unless the alarm threshold or reference value is changed, in which case the alarm is blocked again.

Parameter Name	Description	Available Values	Value Description
Delay	Delay	Initiates a delay in seconds between the trigger source becoming active, and the alarm becoming active. If the trigger source returns to a non-alarm state before the delay time has elapsed, then the alarm is not triggered and the delay timer is reset. A value of 0 turns off the delay timer. **Default: 0**	
Output	Output	Off (0)	Boolean output which is set to '1' when status is not ‘off’
		On (1)	
Ack	Acknowledge	Off (0)	Not acknowledged.
		On (1)	Select YES to acknowledge the alarm. The display then returns to No.
Inhibit	Inhibit the alarm	Off (0)	Alarm not inhibited.
		On (1)	When ‘Inhibit’ is enabled, the alarm is inhibited and Status is set to ‘Off’. If the alarm is active when inhibit is enabled, then it becomes inactive until inhibit is disabled, when its status depends on its configuration. Similarly if the alarm trigger becomes active when the alarm is inhibited, the alarm remains ‘off’ until inhibit is disabled, when its status depends on its configuration. **Default: Off (0)**
StandbyInhibit	Inhibit in Standby	Off (0)	When the instrument is in standby mode, the alarm will be inhibited if this parameter is On. **Default: Off (0)**
		On (1)	

Comms

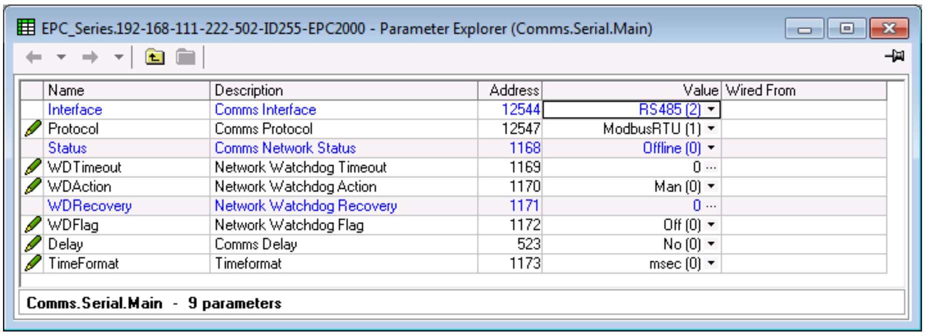
There are two communications options available in the EPC2000 Programmable Controller. These are:

- Ethernet (RJ45) interfaces x2, on the front face.
- and an optional serial communication (EIA-485), terminals HD, HE, HF located on the top of the controller.

Communication settings for Ethernet and Serial communication ports, sometimes referred to as "User Comms", can be configured via iTools using the Comms function blocks. The Ethernet and optional Serial function blocks contain the same parameters, but some parameters may become available/unavailable depending on the interfaces and protocols selected.

Comms.Serial.Main and Comms.Ethernet.Main

The Main subclasses for both the Ethernet and optional serial port allow access to the interface, protocol, and watchdog configuration elements. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Serial.Main)

	Name	Description	Address	Value	Wired From
	Interface	Comms Interface	12544	RS485 (2)	
✎	Protocol	Comms Protocol	12547	ModbusRTU (1)	
	Status	Comms Network Status	1168	Offline (0)	
✎	WDTimeout	Network Watchdog Timeout	1169	0	
✎	WDAction	Network Watchdog Action	1170	Man (0)	
	WDRecovery	Network Watchdog Recovery	1171	0	
✎	WDFlag	Network Watchdog Flag	1172	Off (0)	
✎	Delay	Comms Delay	523	No (0)	
✎	TimeFormat	Timeformat	1173	msec (0)	

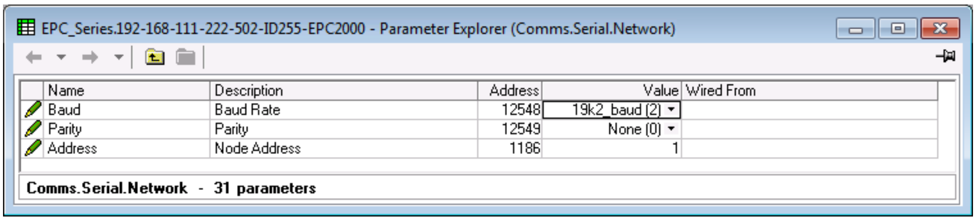
Comms.Serial.Main - 9 parameters

Parameter Name	Description	Available Values	Value Description
Interface	Comms Interface	Communications interface. For the Serial communication port, interface is set according to the hardware fitted. For the Ethernet communication port, it is set according to the configured expected option board in the Instrument function block.	
		None (0)	No interface.
		RS232 (1)	Reserved.
		RS485 (2)	EIA-485 (RS485) - only shown if hardware option ordered.
		RS422 (3)	**Not applicable to EPC2000 Programmable Controller.**
		Ethernet (4)	Ethernet.
		DeviceNet (5)	**Not applicable to EPC2000 Programmable Controller.**
		Profibus (6)	**Not applicable to EPC2000 Programmable Controller.**
		RemoteSP (7)	**Not applicable to EPC2000 Programmable Controller.**
Protocol	Comms Protocol	Protocol running on the comms interface:	
		None (0)	No protocol - when a serial interface is fitted. (No further parameters are shown). **Default: None (0)**
		ModbusRTU (1)	Modbus RTU (serial).
		ModbusSlave (11)	Modbus TCP protocol enabled - only shown if the Ethernet option is fitted. **Default: Ethernet**
		EipAndModSlv (12)	EtherNetIP and Modbus TCP protocol enabled - available in firmware versions V3.xx and above.
		ModMstAndSlv (13)	Modbus Master and Slave communications enabled - available in firmware versions V4.01 and above.
Status	Comms Network Status	Status of the communications used by Modbus TCP:	
		Offline (0)	Offline and not communicating.
		Init (1)	Initializing communications.
		Ready (2)	Ready to accept connection. Not used by Modbus TCP.
		Running (3)	Ready to accept connections or controller communicating.
		Bad_GSD (4)	**Not applicable to EPC2000 Programmable Controller.**
The following four parameters configure the Comms Watchdog Strategy. Used by Modbus RTU and Modbus TCP **Note:** This watchdog may not function as expected for multiple Ethernet connections due to the shared timer and flag for this interface. If the device is configured to receive a setpoint from a remote master via Ethernet connection, it should be routed through the “Remote Input” block ("RemoteInput" on page 113). The remote input block has an independent timeout (default to 1s), allowing the loss of comms to this parameter to be flagged independently of any other Ethernet connections.			
WDTimeout	Network Watchdog Timeout	If the communications stop addressing the instrument longer than this configurable period, the Watchdog Flag will become active. Note: A value of 0 disables the watchdog. All ModbusTCP connections must time-out for the Watchdog flag to be activated. **Default: 0**	

Parameter Name	Description	Available Values	Value Description
WDAction	Network Watchdog Action	Manual (0)	The Watchdog Flag may be cleared Automatically upon reception of valid messages or Manually by clearing the Watchdog Flag parameter. **Default: Manual (0)**
		Auto (1)	
WDRecovery	Network Watchdog Recovery	This parameter is only shown when the Watchdog Action is set to Auto. It is a timer that determines the delay, after reception of valid messages recommences, before the Watchdog Flag is cleared. A value of 0 will reset the Watchdog Flag upon the first valid message received. Other values will wait for at least 2 valid messages to be received within the set time before clearing the Watchdog Flag. **Default: 0**	
WDFlag	Network Watchdog Flag	Off (0)	The Watchdog Flag will become active if the communications stop addressing the instrument longer than the Watchdog Timeout period,
		On (1)	
Delay	Comms Delay	No (0)	Introduces a delay between end of receive and beginning of transmit. This is sometimes necessary if the line transceivers require an extended time to switch to tristate. Comms delay is used by Modbus RTU protocol. **Default: No (0)**
		Yes (1)	
TimeFormat	Time format	msec (0)	Sets the resolution of time parameters on this communications port when read/written via scaled integer comms (Milliseconds, Seconds, Minutes, Hours) **Default: msec (0)**
		sec (1)	
		min (2)	
		hour (3)	

Comms.Serial.Network and Comms.Ethernet.Network

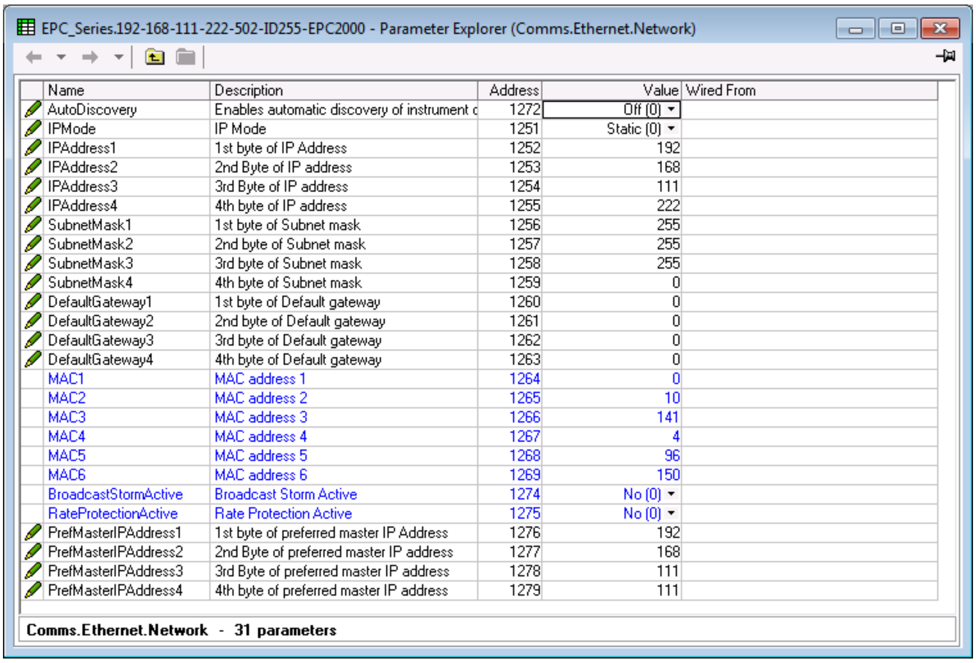
The Network subclasses for both the Ethernet and optional serial port allow access to the essential port configuration. For the Ethernet port, this includes the IP mode, IP address, Subnet mask, and gateway values to be configured, and the MAC address details to be read. For the serial port, this includes the baud rate, parity and Modbus node address to be configured. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Serial.Network)

Name	Description	Address	Value	Wired From
Baud	Baud Rate	12548	19k2_baud (2)	
Parity	Parity	12549	None (0)	
Address	Node Address	1186	1	

Comms.Serial.Network - 31 parameters



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Ethernet.Network)

Name	Description	Address	Value	Wired From
AutoDiscovery	Enables automatic discovery of instrument c	1272	Off (0)	
IPMode	IP Mode	1251	Static (0)	
IPAddress1	1st byte of IP Address	1252	192	
IPAddress2	2nd Byte of IP address	1253	168	
IPAddress3	3rd Byte of IP address	1254	111	
IPAddress4	4th byte of IP address	1255	222	
SubnetMask1	1st byte of Subnet mask	1256	255	
SubnetMask2	2nd byte of Subnet mask	1257	255	
SubnetMask3	3rd byte of Subnet mask	1258	255	
SubnetMask4	4th byte of Subnet mask	1259	0	
DefaultGateway1	1st byte of Default gateway	1260	0	
DefaultGateway2	2nd byte of Default gateway	1261	0	
DefaultGateway3	3rd byte of Default gateway	1262	0	
DefaultGateway4	4th byte of Default gateway	1263	0	
MAC1	MAC address 1	1264	0	
MAC2	MAC address 2	1265	10	
MAC3	MAC address 3	1266	141	
MAC4	MAC address 4	1267	4	
MAC5	MAC address 5	1268	96	
MAC6	MAC address 6	1269	150	
BroadcastStormActive	Broadcast Storm Active	1274	No (0)	
RateProtectionActive	Rate Protection Active	1275	No (0)	
PrefMasterIPAddress1	1st byte of preferred master IP Address	1276	192	
PrefMasterIPAddress2	2nd Byte of preferred master IP address	1277	168	
PrefMasterIPAddress3	3rd Byte of preferred master IP address	1278	111	
PrefMasterIPAddress4	4th byte of preferred master IP address	1279	111	

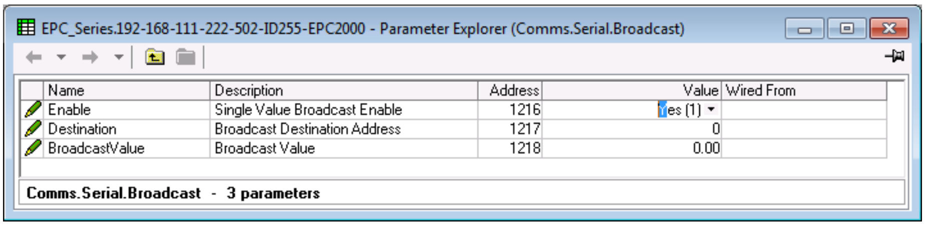
Comms.Ethernet.Network - 31 parameters

Parameter Name	Description	Available Values	Value Description
The first three parameters apply to Modbus Communications protocol.			
Baud	Baud Rate	Baud rate of the network communications:	
		9600_baud(1)	**DO NOT USE**
		19k2_baud(2)	**Default for ModbusRTU**
Parity	Parity	Parity of the network communications: **Default: None (0)**	
		None(0)	No parity
		Even(1)	Even parity
		Odd(2)	Odd parity
Address	Node Address	The address used by the instrument to identify itself on the network. **Default: 1**	
The following parameters apply to Ethernet in the Option Communications sub-list. See also section "Ethernet Configuration" on page 227.			

Parameter Name	Description	Available Values	Value Description
AutoDiscovery	Enables automatic discovery	The controller and iTools software supports automatic discovery of MODBUS TCP enabled instruments. **Default: Off (0)**	
		Off (0)	For cybersecurity reasons the auto discovery feature is turned OFF by default.
		On (1)	To enable this feature set this parameter to ON. Please ensure that your network interface card is set to local. If, for any reason, the controller is not auto-detected and Wi-Fi is enabled on your PC, turn off Wi-Fi and re-start iTools.
IPMode	IP Mode	Static (0)	Static. The IP address, subnet mask and default gateway are set manually. **Default: Static (0)**
		DHCP (1)	DHCP. The IP address, subnet mask and default gateway are supplied by a DHCP server on the network.
IPAddress1	1st byte of IP Address		1st byte of the IP Address: XXX.xxx.xxx.xxx. **Default: 192**
IPAddress2	2nd byte of IP Address		2nd byte of the IP Address: xxx.XXX.xxx.xxx. **Default: 168**
IPAddress3	3rd byte of IP Address		3rd byte of the IP Address: xxx.xxx.XXX.xxx. **Default: 111**
IPAddress4	4th byte of IP Address		4th byte of the IP Address: xxx.xxx.xxx.XXX. **Default: 222**
SubnetMask 1	1st byte of Subnet mask		1st byte of the Subnet Mask: XXX.xxx.xxx.xxx. **Default: 255**
SubnetMask 2	2nd byte of Subnet mask		2nd byte of the Subnet Mask: xxx.XXX.xxx.xxx. **Default: 255**
SubnetMask 3	3rd byte of Subnet mask		3rd byte of the Subnet Mask: xxx.xxx.XXX.xxx. **Default: 255**
SubnetMask 4	4th byte of Subnet mask		4th byte of the Subnet Mask: xxx.xxx.xxx.XXX. **Default: 0**
DefaultGateway1	1st byte of Default gateway		1st byte of the Default Gateway: XXX.xxx.xxx.xxx. **Default: 0**
DefaultGateway2	2nd byte of Default gateway		2nd byte of the Default Gateway: xxx.XXX.xxx.xxx. **Default: 0**
DefaultGateway3	3rd byte of Default gateway		3rd byte of the Default Gateway: xxx.xxx.XXX.xxx. **Default: 0**
DefaultGateway4	4th byte of Default gateway		4th byte of the Default Gateway: xxx.xxx.xxx.XXX. **Default: 0**
MAC1	MAC address 1		1st byte of the MAC address in decimal: XX:xx:xx:xx:xx:xx
MAC2	MAC address 2		2nd byte of the MAC address in decimal: xx:XX:xx:xx:xx:xx
MAC3	MAC address 3		3rd byte of the MAC address in decimal: xx:xx:XX:xx:xx:xx
MAC4	MAC address 4		4th byte of the MAC address in decimal: xx:xx:xx:XX:xx:xx
MAC5	MAC address 5		5th byte of the MAC address in decimal: xx:xx:xx:xx:XX:xx
MAC6	MAC address 6		6th byte of the MAC address in decimal: xx:xx:xx:xx:xx:XX
BroadcastStormActive	Broadcast Storm Active	No (0)	Broadcast storm active. If the reception rate of Ethernet broadcast packets rises too high, broadcast storm mode will become active and the reception of broadcast packets will be disabled until the rate drops.
		Yes (1)	
RateProtectionActive	Rate Protection Active	No (0)	Rate protection active. If rate at which Ethernet unicast packets are received becomes too high, the instrument will enter a special mode that slows down Ethernet processing to preserve core functionality.
		Yes (1)	
PrefMasterIPAddress1	1st byte of preferred master IP Address		1st byte of the preferred master IP Address: XXX.xxx.xxx.xxx. The preferred master IP address is a reserved IP address of a remote client that will be allowed to create a session with the controller, even if the other 3 TCP sessions are currently active. This typically would be a remote HMI to avoid it from being unable to connect to the controller. However, it could equally be a PC running iTools, for example. **Default: 192**
PrefMasterIPAddress2	2nd byte of preferred master IP Address		2nd byte of the preferred master IP Address: xxx.XXX.xxx.xxx. **Default: 168**
PrefMasterIPAddress3	3rd byte of preferred master IP Address		3rd byte of the preferred master IP Address: xxx.xxx.XXX.xxx. **Default: 111**
PrefMasterIPAddress4	4th byte of preferred master IP Address		4th byte of the preferred master IP Address: xxx.xxx.xxx.XXX. **Default: 111**

Comms.Serial.Broadcast

The Broadcast subclass allows configuration of the serial Modbus broadcast parameters. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Serial.Broadcast)

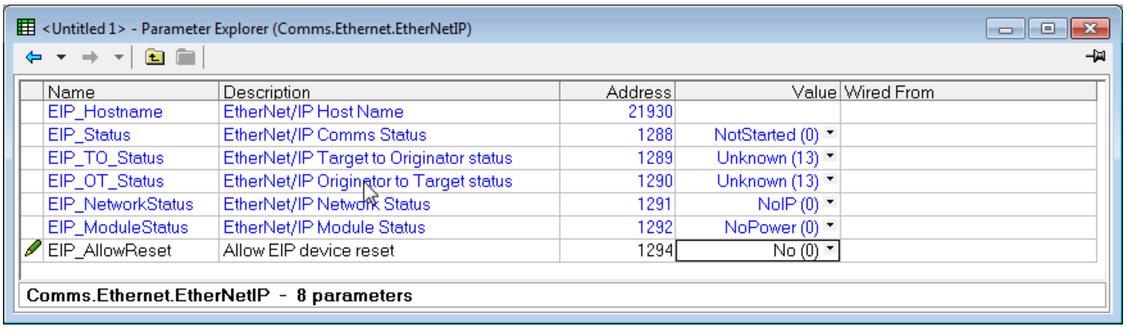
Name	Description	Address	Value	Wired From
Enable	Single Value Broadcast Enable	1216	Yes (1)	
Destination	Broadcast Destination Address	1217	0	
BroadcastValue	Broadcast Value	1218	0.00	

Comms.Serial.Broadcast - 3 parameters

Parameter Name	Description	Available Values	Value Description
Enable	Single Value Broadcast Enable	No (0)	Broadcast comms not enabled **Default: No**
		Yes (1)	Enable single value Modbus broadcast
Destination	Broadcast Destination Address	If the Modbus broadcast facility is enabled, this address will be used as the destination register for the value to be written. For example, if the remote instrument requires a setpoint at register address 26 decimal, the parameter should be set to this value. **Default: 0**	
BroadcastValue	Broadcast Value	If the Modbus broadcast facility is enabled, this value will be sent to the slave devices, after being transformed into a 'scaled integer' 16 bit value. To use the feature, enable broadcast using BroadcastEnable, and wire any instrument value to this parameter. **Default: 0.00**	

Comms.Ethernet.EtherNet/IP

The EtherNet/IP subclass allows configuration of the EtherNet/IP parameters. The figure below shows the parameters and the table which follows details each parameter.



<Untitled 1> - Parameter Explorer (Comms.Ethernet.EtherNetIP)

Name	Description	Address	Value	Wired From
EIP_Hostname	EtherNet/IP Host Name	21930		
EIP_Status	EtherNet/IP Comms Status	1288	NotStarted (0)	
EIP_TO_Status	EtherNet/IP Target to Originator status	1289	Unknown (13)	
EIP_OT_Status	EtherNet/IP Originator to Target status	1290	Unknown (13)	
EIP_NetworkStatus	EtherNet/IP Network Status	1291	NoIP (0)	
EIP_ModuleStatus	EtherNet/IP Module Status	1292	NoPower (0)	
EIP_AllowReset	Allow EIP device reset	1294	No (0)	

Comms.Ethernet.EtherNetIP - 8 parameters

Parameter Name	Description	Available Values	Value Description
EIP_Hostname	EtherNet/IP Host Name		
EIP_Status	EtherNet/IP Comms Status	NotStarted (0)	EtherNet/IP comms not started.
		Ready (1)	EtherNet/IP comms ready.
		Standby (2)	EtherNet/IP comms on standby.
		Running (3)	EtherNet/IP comms running.

Parameter Name	Description	Available Values	Value Description
EIP_TO_Status	EtherNet/IP Target to Originator status	Data Exchanged (0)	Data correctly exchanged.
		InProgress (1)	Connection in progress.
		NoConnection (2)	No connection detected.
		Timeout (3)	Connection timed out.
		NoMacAddress (4)	Unknown MAC address.
		NoConsume (5)	Consumption Timeout.
		ConnectionClosed (6)	Connection Closed.
		ModuleStop (7)	Module stopped.
		EncapsulationErrorDetected (8)	Encapsulation error detected.
		TcpConnectionErrorDetected (9)	TCP connection error detected.
		NoResource (10)	No resource.
		BadFormat (11)	Bad format.
		Idle (12)	Idle mode.
		Unknown (13)	Unknown status.
EIP_OT_Status	EtherNet/IP Originator to Target status	Data Exchanged (0)	Data correctly exchanged.
		InProgress (1)	Connection in progress.
		NoConnection (2)	No connection detected.
		Timeout (3)	Connection timed out.
		NoMacAddress (4)	Unknown MAC address.
		NoConsume (5)	Consumption Timeout.
		ConnectionClosed (6)	Connection Closed.
		ModuleStop (7)	Module stopped.
		EncapsulationErrorDetected (8)	Encapsulation error detected.
		TcpConnectionErrorDetected (9)	TCP connection error detected.
		NoResource (10)	No resource.
		BadFormat (11)	Bad format.
		Idle (12)	Idle mode.
		Unknown (13)	Unknown status.
EIP_NetworkStatus	EtherNet/IP Network Status	NoIP (0)	No IP Address found.
		NoConnection (1)	IP address configured but no connection enabled.
		Connected (2)	IP address configured and connection established.
		Timeout (3)	Connection timed out.
		ErrorDetected (4)	Error detected in network comms.
EIP_ModuleStatus	EtherNet/IP Module Status	NoPower (0)	No power.
		NoConfig (1)	Not configured.
		Run (2)	Running.
		ErrorDetected (3)	Module error detected.
EIP_AllowReset	Allow EIP device reset	No (0)	Device reset not allowed.
		Yes (1)	Device reset allowed.

Input Linearization (LIN16)

The LIN16 list is only available if a Toolkit option has been ordered.

A LIN16 function block converts an input signal into an output PV using a series of up to 14 straight lines (16 points) to characterize the conversion.

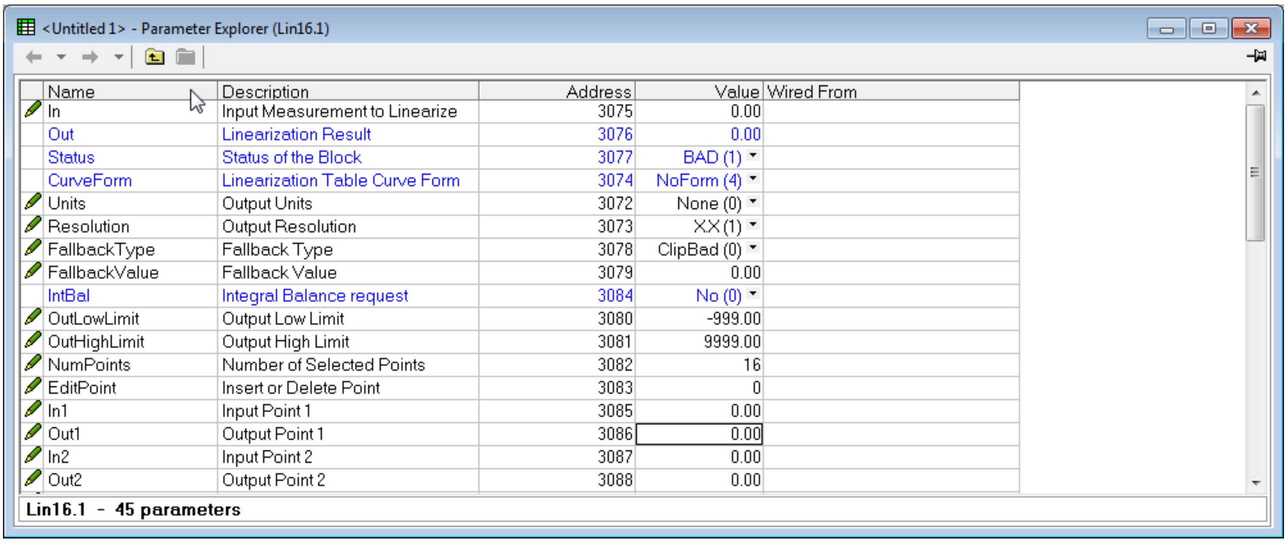
In EPC2000 series controllers, from firmware versions V4.01 and above, two instances of the linearization function block have been added. It is an orderable option protected by Feature Security.

The LIN16 function block allows a user to create a their own linearization to match the characteristics of a particular sensor not covered by any of the standard inputs. It can also be used for the adjustment of the process variable to account for differences introduced by the overall measurement system or to derive a different process variable. These may be set up using iTools. For this reason the configuration of the LIN16 block is described in the iTools section.

The parameter descriptions for the LIN16 block are shown in the following section:

Linearization Block Parameters

The Linearization Block subclass allows configuration of the Linearization parameters. The figure below shows the parameters and the table which follows details each parameter.



<Untitled 1> - Parameter Explorer (Lin16.1)

	Name	Description	Address	Value	Wired From
✎	In	Input Measurement to Linearize	3075	0.00	
	Out	Linearization Result	3076	0.00	
	Status	Status of the Block	3077	BAD (1)	
	CurveForm	Linearization Table Curve Form	3074	NoForm (4)	
✎	Units	Output Units	3072	None (0)	
✎	Resolution	Output Resolution	3073	XX (1)	
✎	FallbackType	Fallback Type	3078	ClipBad (0)	
✎	FallbackValue	Fallback Value	3079	0.00	
	IntBal	Integral Balance request	3084	No (0)	
✎	OutLowLimit	Output Low Limit	3080	-999.00	
✎	OutHighLimit	Output High Limit	3081	9999.00	
✎	NumPoints	Number of Selected Points	3082	16	
✎	EditPoint	Insert or Delete Point	3083	0	
✎	In1	Input Point 1	3085	0.00	
✎	Out1	Output Point 1	3086	0.00	
✎	In2	Input Point 2	3087	0.00	
✎	Out2	Output Point 2	3088	0.00	

Lin16.1 - 45 parameters

Parameter Name	Description	Available Values	Value Description
In	Input Measurement to Linearize	The Input Value to linearize through the Linearization Table.	
Out	Linearization Result	The Output Value that is the result of the linearization of the Input Value through the Linearization Table.	
Status	Status of the Block	GOOD (0)	Good status indicates a correct linearization of the Input.
		BAD (1)	Bad status may be caused by bad input signal (e.g. sensor break), output out of range or invalid series of points.

Parameter Name	Description	Available Values	Value Description
CurveForm	Linearization Table Curve Form	FreeForm (0)	All the selected input points are used to generate a freeform curve.
		Increasing (1)	All the selected input points are used to generate an increasing curve.
		Decreasing (2)	All the selected input points are used to generate a decreasing curve.
		SkippedPoints (3)	At least one input point has been skipped because of not expected order with respect to the previous ones.
		NoForm (4)	No valid pair of points has been found that has strictly monotonically increasing input values.
Units	Output Units	None (0)	
		C F K Temp (1)	The parameter associated with this units definition is an absolute temperature and hence will adopt the global temperature units of the instrument. In addition, if the global units are changed the parameter, will be converted to the new units. For example degC to degF
		V (2)	Volts.
		mV (3)	millivolts.
		A (4)	Amps.
		mA (5)	milliamps.
		PH (6)	Measurement of acidity or alkalinity.
		mmHg (7)	Measurement of pressure.
		psi (8)	Measurement of pressure.
		Bar (9)	Measurement of pressure.
		mBar (10)	Measurement of pressure.
		PercentRH (11)	Percentage Relative Humidity.
		Percent (12)	Percentage.
		mmWG (13)	millimetres water gauge.
		inWG (14)	inches water gauge.
		inWW (15)	inches water.
		Ohms (16)	Ohms.
		PSIG (17)	Pounds per square inch gauge.
		PercentO2 (18)	Percentage 0xygen.
		PPM (19)	Parts per Milliion.
		PercentCO2 (20)	Percentage Carbon Dioxide.
		PercentCarb (21)	Percentage Carbon.
		PercentPerSec (22)	Percentage per Second.
		RelTemperature (24)	Relative Temperature.
		Vacuum (25)	Measurement of vacuum in mBar / pascals or Torr. If configured a parameter will used the overall instrument units of vacuum.
		Secs (26)	seconds.
		Mins (27)	minutes.
		Hours (28)	hours.
		Days (29)	days.
		Mb (30)	
		Mb (31)	
		ms (32)	milliseconds.
Resolution	Output Resolution	X (0)	No decimal places.
		XX (1)	One decimal place.
		XXX (2)	Two decimal places.
		XXXX (3)	Three decimal places.
		XXXXX (4)	Four decimal places.

Parameter Name	Description	Available Values	Value Description
FallbackType	Fallback Type	Clip Bad (0)	The measurement is clipped to the limit it has exceeded and its status is set to BAD, such that any function block using this measurement can operate its own fallback strategy. For example, the control loop may hold its output.
		Clip Good (1)	The measurement is clipped to the limit it has exceeded and its status is set to GOOD, such that any function block using this measurement may continue to calculate and not employ its own fallback strategy.
		Fallback Bad (2)	The measurement will adopt the configured fallback value. Which has been set by the user. In addition the status of the measured value will be set to BAD, such that any function block using this measurement can operate its own fallback strategy. For example, the control loop may hold its output.
		Fallback Good (3)	The measurement will adopt the configured fallback value, which has been set by the user. In addition the status of the measured value will be set to GOOD, such that any function block using this measurement may continue to calculate and not employ its own fallback strategy.
		Up Scale (4)	The measurement will be forced to adopt its high limit, this is like having a resistive pull up on an input circuit. In addition the status of the measurement is set to BAD, such that any function block using this measurement can operate its own fallback strategy. For example, the control loop may hold its output.
		Down Scale (6)	The measurement will be forced to adopt it's low limit, this is like having a resistive pull down on an input circuit. In addition the status of the measurement is set to BAD, such that any function block using this measurement can operate its own fallback strategy. For example, the control loop may hold its output.
FallbackValue	Fallback Value	In the event of a bad status, the output may be configured to adopt the fallback value. This allows the strategy to dictate a known output value.	
IntBal	Integral Balance request	No (0)	
		Yes (1)	
OutLow Limit	Output Low Limit	Minimum value allowed for the output. If the linearization table would result in an output value that is less than the low limit, then the Fallback strategy will be actuated.	
OutHighLimit	Output High Limit	Maximum value allowed for the output. If the linearization table would result in an output value that is greater than the high limit, then the Fallback strategy will be actuated.	
NumPoints	Number of Selected Points	Number of points selected to define the linearization table. It can be set between 2 and 16.	
EditPoint	Insert or Delete Points	A point can be added or deleted by specifying the desired position. Set EditPoint equal to 1, 2, ..., 16 to insert a point at the associated position; each following point will be moved into the next position. Set EditPoint equal to -1, -2, ..., -16 to remove the point at the associated position; each following point will be moved into the previous position and the last one is maintained.	
In1	Input Point 1	Input Coordinate of Point 1 of the linearization table.	
Out1	Output Point 1	Output Coordinate of Point 1 of the linearization table.	
Up to 16 Input and output points are available depending on the setting of the Number of Points parameter.			
In16	Input Point 16	Input Coordinate of Point 16 of the linearization table.	
Out16	Output Point 16	Output Coordinate of Point 16 of the linearization table.	

Qcode

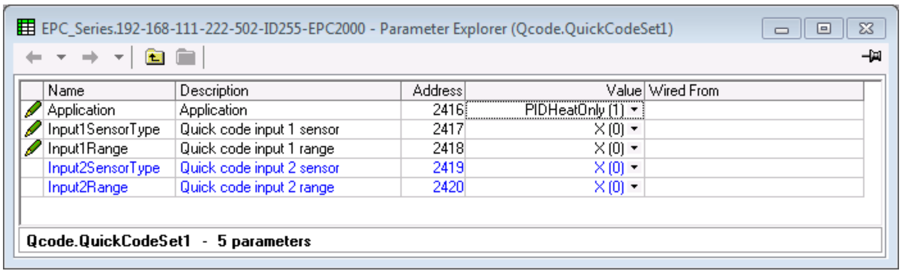
Quick Codes (Qcode) allow the EPC2000 Programmable Controller to be automatically configured for commonly used functions such heat-only or heat-and-cool type of applications. The Quick Codes configure the instrument’s parameters, input type, range, digital input functions and graphical wiring.

There are two Qcode function blocks which are used to configure the basic operation desired, and then a third Qcode function block to automatically load the configuration into the instrument.

For further information on Quick Codes, refer to the "Quick Start Tables" on page 60.

Qcode.QuickCodeSet1 and Qcode.QuickCodeSet2

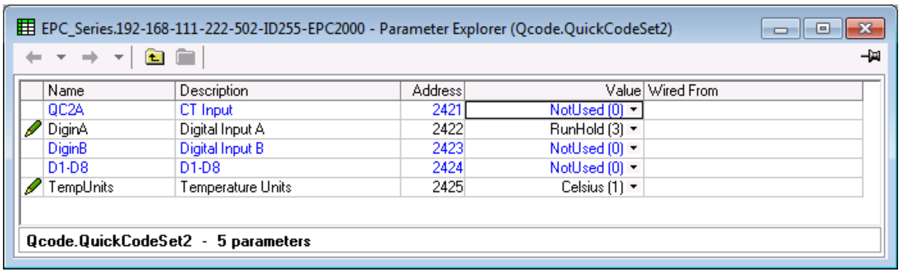
The QuickCodeSet1 function block allows you to choose the specific application for the controller to be automatically configured for, and to specify the type of thermocouple to be used and the temperature range. The QuickCodeSet2 function block builds on set 1, and allows configuration of the digital input function and the temperature units. To apply the configuration, the single parameter in the QuickCodeExit function block needs to set. The figure below shows the parameters in both function blocks and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Qcode.QuickCodeSet1)

	Name	Description	Address	Value	Wired From
✎	Application	Application	2416	PIDHeatOnly (1)	
✎	Input1SensorType	Quick code input 1 sensor	2417	X (0)	
✎	Input1Range	Quick code input 1 range	2418	X (0)	
	Input2SensorType	Quick code input 2 sensor	2419	X (0)	
	Input2Range	Quick code input 2 range	2420	X (0)	

Qcode.QuickCodeSet1 - 5 parameters



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Qcode.QuickCodeSet2)

	Name	Description	Address	Value	Wired From
	QC2A	CT Input	2421	NotUsed (0)	
✎	DiginA	Digital Input A	2422	RunHold (3)	
	DiginB	Digital Input B	2423	NotUsed (0)	
	D1-D8	D1-D8	2424	NotUsed (0)	
✎	TempUnits	Temperature Units	2425	Celsius (1)	

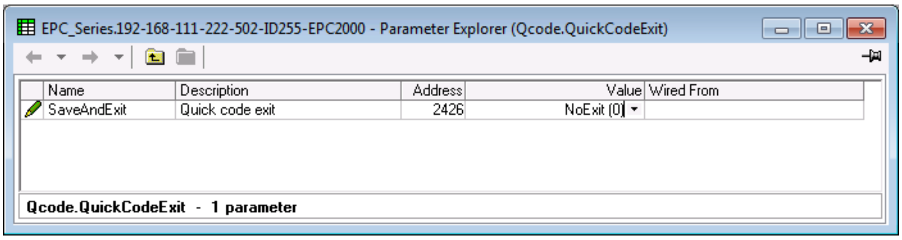
Qcode.QuickCodeSet2 - 5 parameters

Parameter Name	Value		Description
QuickCode Set 1			
Application			Defines the application.
	None	0	No application configured. The controller has no soft wiring.
	PIDHeatOnly	1	PID heat only controller.
	PIDHeatCool	2	PID heat/cool controller.

Parameter Name	Value		Description
Input 1 Sensor Type			Defines the type of the input sensor connected to the sensor input.
	X	0	Use Default.
	B	1	Type B.
	J	2	Type J.
	K	3	Type K.
	L	4	Type L.
	N	5	Type N.
	R	6	Type R.
	S	7	Type S.
	T	8	Type T.
	Pt100	20	PT100.
	80mV	30	0-80mV.
	10V	31	0-10V.
	20mA	32	0-20mA.
	4-20mA	33	4-20mA.
Input 1 Range			Defines the range of the sensor input.
	X	0	Use Default.
	1	1	0-100°C (32-212°F).
	2	2	0-200°C (32-392°F).
	3	3	0-400°C (32-752°F).
	4	4	0-600°C (32-1112°F).
	5	5	0-800°C (32-1472°F).
	6	6	0-1000°C (32-1832°F).
	7	7	0-1200°C (32-2192°F).
	8	8	0-1300°C (32-2372°F).
	9	9	0-1600°C (32-2912°F).
	A	10	0-1800°C (32-3272°F).
	F	11	Full Range.
Quick Code Set2			
Digital input A	Not used	0	Defines the functionality of digital input A.
	Alarm acknowledge	1	
	Loop Auto/Manual	2	
	Programmer Run/Hold	3	
	Keylock	4	
	Setpoint Select	5	
	Programmer Run/Reset	6	
	Loop Remote/Local	7	
	Recipe Select	8	
	Loop Track	9	
Digital input B	As for Digital Input A		
D1-D8			**Not applicable to the EPC2000 Programmable Controller.**
Temperature units	Default	0	Default temperature units.
	Celsius	1	Degrees Celsius.
	Fahrenheit	2	Degrees Fahrenheit.
	Kelvin	3	Kelvin.
QuickCode Exit			
	NoExit	0	Do not exit quick start mode.
	Save	1	Save quick start settings.
	Discard	2	Discard quick start settings.

Qcode.QuickCodeExit

The QuickCodeExit function block allows you to apply the configuration to the EPC2000 Programmable Controller using the settings defined in the QuickCodeSet1 and QuickCodeSet2 function blocks (Save option). You can also discard the configuration settings defined (Discard option). The figure below shows the parameter in the function block and the table which follows details the parameter options.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Qcode.QuickCodeExit)

Name	Description	Address	Value	Wired From
SaveAndExit	Quick code exit	2426	NoExit (0)	

Qcode.QuickCodeExit - 1 parameter

Parameter Name	Value		Description
SaveAndExit	NoExit	0	Do not exit quick start mode.
	Save	1	Save quick start settings and restart instrument.
	Discard	2	Discard quick start settings and restart instrument.

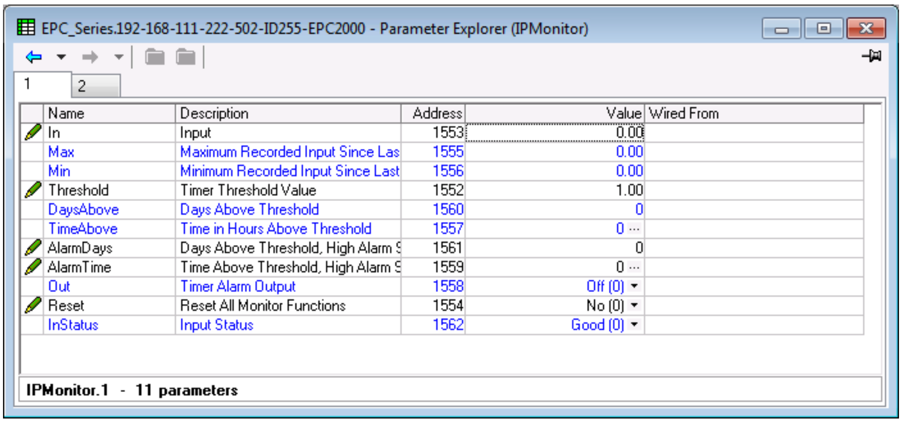
IPMonitor

The Input Monitor (IPMonitor) category contains two function blocks (IPMonitor.1 and IPMonitor.2) which allow any variable in the controller to be monitored. The function blocks then provide three functions:

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

There can be up to two IPMonitor blocks utilized and these are only available if the Toolkit option has been ordered.

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (IPMonitor)

Name	Description	Address	Value	Wired From
In	Input	1553	0.00	
Max	Maximum Recorded Input Since Las	1555	0.00	
Min	Minimum Recorded Input Since Last	1556	0.00	
Threshold	Timer Threshold Value	1552	1.00	
DaysAbove	Days Above Threshold	1560	0	
TimeAbove	Time in Hours Above Threshold	1557	0 ···	
AlarmDays	Days Above Threshold, High Alarm S	1561	0	
AlarmTime	Time Above Threshold, High Alarm S	1559	0 ···	
Out	Timer Alarm Output	1558	Off (0)	
Reset	Reset All Monitor Functions	1554	No (0)	
InStatus	Input Status	1562	Good (0)	

IPMonitor.1 - 11 parameters

Parameter Name	Parameter Description	Available Values	Value Description
In	Input	Monitored input value.	
Max	Maximum Recorded Input Since Last Reset	This function continuously monitors the input value. If the value is higher than the previously recorded maximum, it becomes the new maximum. This value is retained following a power outage.	
Min	Minimum Recorded Input Since Last Reset	This function continuously monitors the input value. If the value is lower than the previously recorded minimum, it becomes the new minimum. This value is retained following a power outage.	
Threshold	Timer Threshold Value	The input timer accumulates the time the input PV spends above this trigger value. **Default: 1.0**	
DaysAbove	Days Above Threshold	Accumulated days the input has spent above threshold since the last reset. Days is an integer count of the 24 hour periods only. The Days value should be combined with the Time value to make the total time above threshold.	
TimeAbove	Time in Hours Above Threshold	Accumulated time above the timer threshold since last reset. The time value accumulates from 00:00.0 to 23:59.59. Overflows are added to the days value.	
AlarmDays	Days Above Threshold, High Alarm Setpoint	Days threshold for the monitors time alarm. Used in combination with the TimeAbove parameter. The AlmOut is set to true if the inputs accumulated time above threshold is higher than the timer high parameters. **Default: 0**	
AlarmTime	Time Above Setpoint, High Alarm Setpoint	Time threshold for the monitors time alarm. Used in combination with the AlmDay parameter. The AlmOut is set to true if the inputs accumulated time above threshold is higher than the timer high parameters. **Default: 0**	
Out	Timer Alarm Output	Off (0)	
		On (1)	Set true if the accumulated time that the input spends above the threshold value is higher than the alarm setpoint.
Reset	Reset All Monitor Functions	No (0)	**Default: No (0)**
		Yes (1)	Resets the Max and Min values and resets the time above threshold to zero.

Parameter Name	Parameter Description	Available Values	Value Description
InStatus	Input Status	Displays the status of the input. See section "Status" on page 94 for a list of enumerated values.	

Total

A totalizer is an electronic integrator, primarily used to record the numeric total over time of a measured value that is expressed as a rate. For example, the number of liters/gallons (since reset), based on a flow rate in liters (gallons) per minute.

One totalizer function block is available in the EPC2000 and is only available if the Toolkit option has been ordered. A totalizer can, by soft wiring, be connected to any measured value. The outputs from the totalizer are its integrated value and an alarm state. The user may set a setpoint which causes the alarm to activate once the integration exceeds the setpoint.

The totalizer has the following attributes:

1. Run/Hold/Reset

    In **Run**, the totalizer will integrate its input and continuously test against an alarm setpoint. The higher the value of the input the faster the integrator will run.

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

    In **Reset**, the totalizer will be zeroed, and alarms will be reset.

2. Alarm Setpoint

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

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

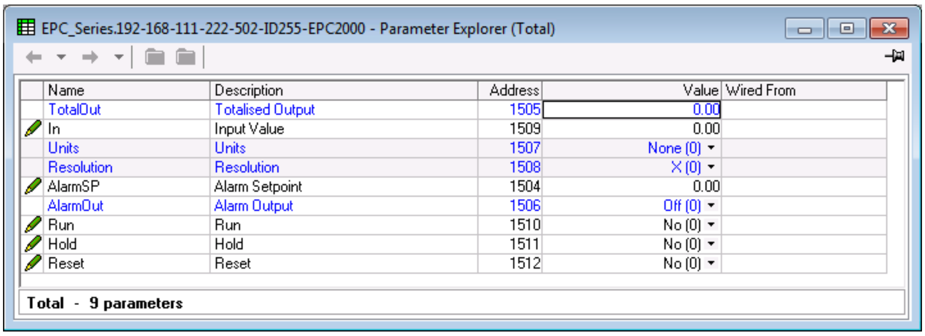
    If the totalizer alarm setpoint is set to 0.0, the alarm will be off. It will not detect values above or below.

    The alarm output is a single state output. It may be cleared by resetting the totalizer, stopping the Run condition, or by changing the alarm setpoint.

3. The total is limited to max and min 32-bit floating point values.

4. The totalizer helps ensure that resolution is maintained when integrating small values onto a large total. However, very small values will not be integrated into a large value, i.e. 0.000001 will not be integrated into 455500.0 this is due to the limitations of 32-bit floating point resolution.

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Total)

	Name	Description	Address	Value	Wired From
	TotalOut	Totalised Output	1505	0.00	
✎	In	Input Value	1509	0.00	
	Units	Units	1507	None (0)	
	Resolution	Resolution	1508	X (0)	
✎	AlarmSP	Alarm Setpoint	1504	0.00	
	AlarmOut	Alarm Output	1506	Off (0)	
✎	Run	Run	1510	No (0)	
✎	Hold	Hold	1511	No (0)	
✎	Reset	Reset	1512	No (0)	

Total - 9 parameters

Parameter Name	Description	Available Values	Value Description
TotalOut	Totalized Output	The totalized value.	
In	Input Value	The value to be totalized. The totalizer stops accumulating if the input is 'Bad'.	
Units	Units	See section "Units" on page 93 for a list of units used throughout.	
Resolution	Resolution	X (0)	Totalizer resolution. **Default: X (0) - no decimal places**
		X.X (1)	One decimal place.
		X.XX (2)	Two decimal places.
		X.XXX (3)	Three decimal places.
		X.XXXX (4)	Four decimal places.
AlarmSP	Alarm Setpoint	Sets the totalized value at which an alarm will occur.	
AlarmOut	Alarm Output	This is a read-only value which indicates the alarm output On or Off. The totalized value can be a positive number or a negative number. If the number is positive the alarm occurs when: Total > Alarm Setpoint If the number is negative the alarm occurs when: Total < Alarm Setpoint	
		Off (0)	Off.
		On (1)	On.
Run	Run	No (0)	Totalizer not running. See note below.
		Yes (1)	Select to run the totalizer.
Hold	Hold	No (0)	Totalizer not in hold. See note below.
		Yes (1)	Holds the totalizer at its current value.
Reset	Reset	No (0)	Totalizer not in reset.
		Yes (1)	Resets the totalizer.

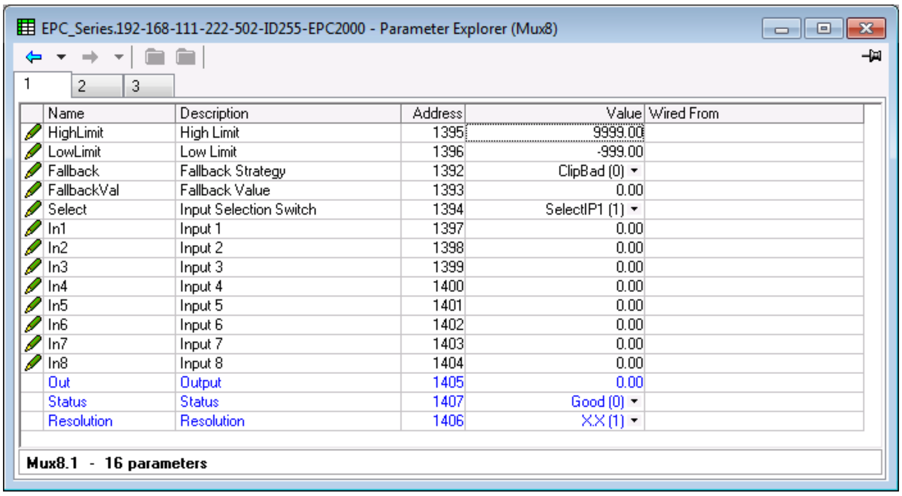
Note: The Run & Hold parameters are designed to be wired to (for example) digital inputs. Run must be 'on' and Hold must be 'off' for the totalizer to operate.

Mux8

The 8-input analog multiplexers (Mux8) function blocks may be used to switch one of eight inputs to an output. It is usual to wire inputs to a source within the controller which selects that input at the appropriate time or event.

There are up to three instances of 8-Input analog multiplexer (switch) in the EPC2000 Programmable Controller and these are only available if the Toolkit option has been ordered.

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Mux8)

Name	Description	Address	Value	Wired From
HighLimit	High Limit	1395	9999.00	
LowLimit	Low Limit	1396	-999.00	
Fallback	Fallback Strategy	1392	ClipBad (0)	
FallbackVal	Fallback Value	1393	0.00	
Select	Input Selection Switch	1394	SelectIP1 (1)	
In1	Input 1	1397	0.00	
In2	Input 2	1398	0.00	
In3	Input 3	1399	0.00	
In4	Input 4	1400	0.00	
In5	Input 5	1401	0.00	
In6	Input 6	1402	0.00	
In7	Input 7	1403	0.00	
In8	Input 8	1404	0.00	
Out	Output	1405	0.00	
Status	Status	1407	Good (0)	
Resolution	Resolution	1406	X.X (1)	

Mux8.1 - 16 parameters

Parameter Name	Description	Available Values	Value Description
HighLimit	High Limit	The high limit for all inputs and the fall back value. Range Low Limit to Max 32-bit floating point value (decimal point depends on resolution). **Default: 9999**	
LowLimit	Low Limit	The low limit for all inputs and the fall back value. Range Min 32-bit floating point value to High Limit (decimal point depends on resolution). **Default: -999**	
Fallback	Fallback Strategy	The state of the Output and Status parameters when either input is bad or operation cannot be completed. This parameter could be used in conjunction with fallback value.	
		ClipBad (0)	Clip Bad. If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and 'Status' is set to 'Good'. If the input signal is within the limits, but its status is bad, the output is set to the 'Fallback' value. **Default: ClipBad (0)**
		ClipGood (1)	Clip Good. If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and 'Status' is set to 'Bad'. If the input signal is within the limits, but its status is bad, the output is set to the 'Fallback' value.
		FallBad (2)	Fall Bad. If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the 'Fallback' value, and the 'Status' is set to 'Bad'.
		FallGood (3)	Fall Good. If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the 'Fallback' value, and the 'Status' is set to 'Good'.
		UpScaleBad (4)	Upscale. If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the 'High Limit'.
		DownScaleBad (6)	Downscale. If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the 'Low Limit'.
FallbackVal	Fallback Value	Used (in accordance with Fallback Strategy) to define the output value when fallback strategy is active. Range Low Limit to High Limit (decimal point depends on resolution).	

Parameter Name	Description	Available Values	Value Description
Select	Input Selection Switch	Input values (normally wired to an input source). **Default: SelectP1 (1)**	
		SelectP1 (1)	
		SelectP2 (2)	
		SelectP3 (3)	
		SelectP4 (4)	
		SelectP5 (5)	
		SelectP6 (6)	
		SelectP7 (7)	
		SelectP8 (8)	
In1	Input 1	0.00	To input values if not wired. Range Min 32-bit floating point value to Max 32-bit floating point value.
In2	Input 2	0.00	
In3	Input 3	0.00	
In4	Input 4	0.00	
In5	Input 5	0.00	
In6	Input 6	0.00	
In7	Input 7	0.00	
In8	Input 8	0.00	
Out	output	Indicates the analog value of the output between high and low limits.	
Status	Status	Used in conjunction with Fallback to indicate the status of the operation. Typically, status is used to flag the status of the operation and used in conjunction with fallback strategy. May be used as an interlock for other operations. See section "Status" on page 94 for a list of enumerated values.	
Resolution	Resolution	Indicates the resolution of the output. The resolution of the output is taken from the selected input. If the selected input is not wired, or if its status is 'bad' then the resolution will be set to one decimal place.	
		X (0)	No decimal places **Default: X (0)**
		X.X (1)	One decimal place
		X.XX (2)	Two decimal places
		X.XXX (3)	Three decimal places
		X.XXXX (4)	Four decimal places

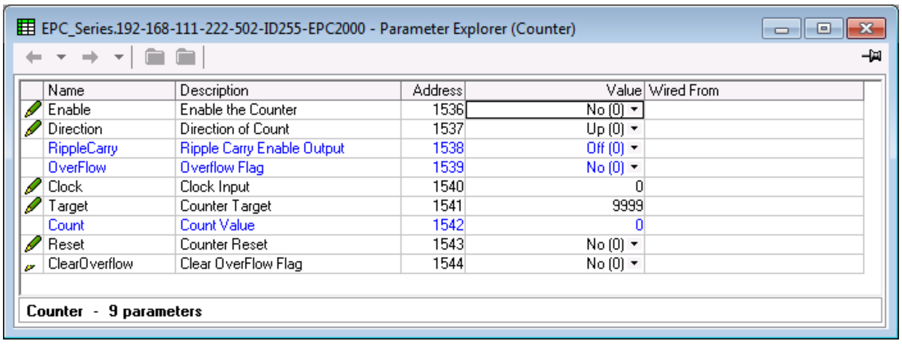
Counter

The counter function block is only available if the Toolkit option has been ordered.

One counter function block is available in the EPC2000.

Each time the ‘Clock’ input is triggered the ‘Count’ output is incremented by 1 for an Up Counter and decremented by 1 for a Down Counter. A target value can be set and when this is reached the Ripple Carry flag is set. This flag can be wired to operate an event or other output.

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Counter)

Name	Description	Address	Value	Wired From
Enable	Enable the Counter	1536	No (0)	
Direction	Direction of Count	1537	Up (0)	
RippleCarry	Ripple Carry Enable Output	1538	Off (0)	
OverFlow	Overflow Flag	1539	No (0)	
Clock	Clock Input	1540	0	
Target	Counter Target	1541	9999	
Count	Count Value	1542	0	
Reset	Counter Reset	1543	No (0)	
ClearOverflow	Clear OverFlow Flag	1544	No (0)	

Counter - 9 parameters

Parameter Name	Description	Available Values	Value Description
Enable	Enable the Counter	No (0)	Count is frozen while Enable is FALSE. **Default: No (0)**
		Yes (1)	Count responds to Clock events when Enable is TRUE.
Direction	Direction of Count	Up (0)	Up counter. See note (1) below. **Default: Up (0)**
		Down (1)	Down counter. See note (2) below.
RippleCarry	Ripple Carry Enable Output	Ripple carry is normally used to act as an enabling input to the next counter. However, in EPC2000, only one counter is available. Ripple carry is turned On when the counter reaches the target set. This can be wired to operate an event or alarm or other function as required.	
		Off (0)	Off.
		On (1)	On.
OverFlow	Overflow Flag	No (00	Overflow flag is held true (Yes) when the counter reaches zero (Down) or passes target (Up).
		Yes (1)	
Clock	Clock Input	Clock input to counter. The counter will increment (for an UP Counter) on a positive going edge (FALSE to TRUE). This is normally wired to an input source such as a digital input.	
Target	Counter Target	Level count to which the counter is aiming. **Default: 9999**	
Count	Count Value	Counts each time a clock input occurs until the target is reached. Range 0 to 99999.	
Reset	Counter Reset	No (0)	Counter not reset.
		Yes (1)	When the Reset is set TRUE, the Count is set to 0 in 'up' mode or to Target in 'down' mode. The Reset also clears the overflow flag.
ClearOverflow	Clear OverFlow Flag	No (0)	Not cleared.
		Yes (1)	Clears the overflow flag.

Notes:

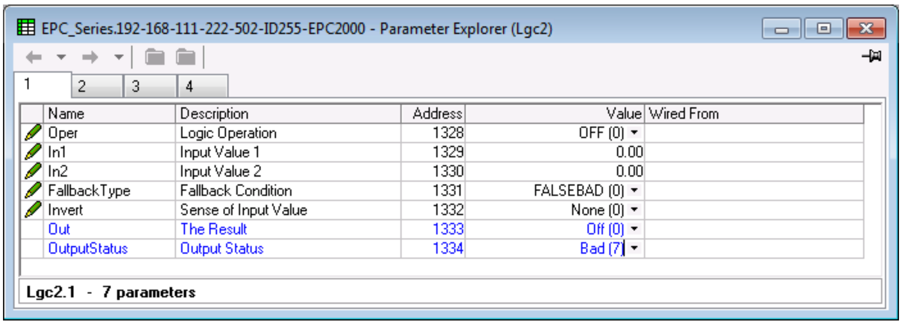
1. When configured as an Up counter, Clock events increment Count until reaching the Target. On reaching Target RippleCarry is set true. At the next clock pulse, Count returns to zero. Overflow is latched true and RippleCarry is returned false.
2. When configured as a down counter, Clock events decrement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

Lgc2

The two-input Logic Operator (Lgc2) function block allows the controller to perform logical calculations on two input values. These values can be sourced from any available parameter including Analog Values, User Values and Digital Values.

There can be up to four LGC2 operator blocks utilized and these are only available if the Toolkit option has been ordered.

Up to four different Lgc2 function blocks may be configured. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Lgc2)

Name	Description	Address	Value	Wired From
Oper	Logic Operation	1328	OFF (0)	
In1	Input Value 1	1329	0.00	
In2	Input Value 2	1330	0.00	
FallbackType	Fallback Condition	1331	FALSEBAD (0)	
Invert	Sense of Input Value	1332	None (0)	
Out	The Result	1333	Off (0)	
OutputStatus	Output Status	1334	Bad (7)	

Lgc2.1 - 7 parameters

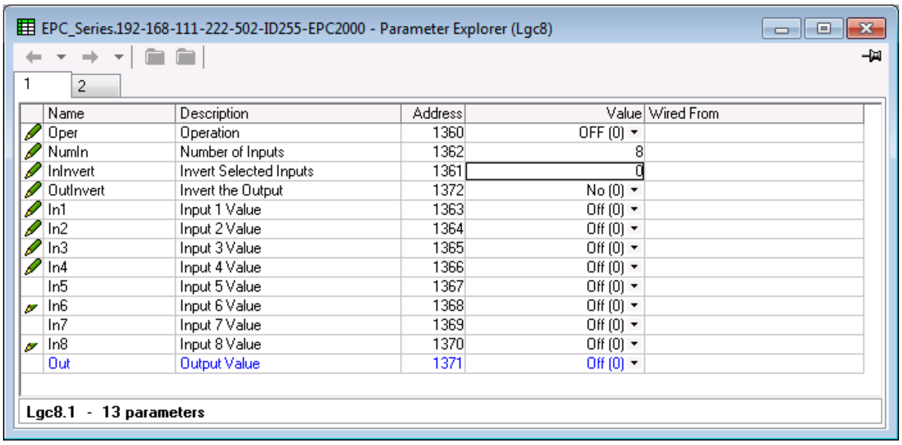
Parameter Name	Description	Available Values	Value Description
Oper	Logic operation	OFF (0)	The selected logic operator is turned off. **Default: OFF (0)**
		AND (1)	The output result is ON when both Input 1 and Input 2 are ON.
		OR 2()	The output result is ON when either Input 1 or Input 2 is ON.
		XOR (3)	Exclusive OR. The output result is true when one and only one input is ON. If both inputs are ON the output is OFF.
		LATCH (4)	Input 1 sets the latch, Input 2 resets the latch.
		EQUAL (5)	Equal. The output result is ON when Input 1 = Input 2.
		NOTEQUAL (6)	Not equal.The output result is ON when Input 1 ≠ Input 2.
		GREATERTHAN (7)	Greater than. The output result is ON when Input 1 > Input 2.
In1 In2	Input Value 1 Input Value 2	Normally wired to a logic, analog or user value. May be set to a constant value if not wired.	
FallBackType	Fallback Condition	FALSEBAD (0)	The output value is FALSE and the status is BAD. **Default: FALSEBAD (0)**
		TRUEBAD (1)	The output value is TRUE and the status is BAD.
		FALSEGOOD (2)	The output value is FALSE and the status is GOOD.
		TRUEGOOD (3)	The output value is TRUE and the status is GOOD.
Invert	Sense of Input Value	None (0)	The sense of the input value, may be used to invert one or both of the inputs. **Default: None (0)**
		Input1 (1)	Invert input 1.
		Input (2)	Invert input 2.
		Both (3)	Invert both inputs.
Out	The Result	On (1)	The output from the operation is a boolean (true/false) value.
		Off (0)	
OutputStatus	Output Status	The status of the result value (good/bad). See section "Status" on page 94 for a list of enumerated values.	

Lgc8

The eight-input Logic Operator (Lgc8) function block only appears if the function has been enabled, and allows the controller to perform logical calculations on up to eight input values. These values can be sourced from any available parameter including Analog Values, User Values and Digital Values. Up to two eight-input logic operators are available.

There can be up to two Lgc8 operator blocks utilized and these are only available if the Toolkit option has been ordered.

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Lgc8)

Name	Description	Address	Value	Wired From
Oper	Operation	1360	OFF (0)	
NumIn	Number of Inputs	1362	8	
InInvert	Invert Selected Inputs	1361	0	
OutInvert	Invert the Output	1372	No (0)	
In1	Input 1 Value	1363	Off (0)	
In2	Input 2 Value	1364	Off (0)	
In3	Input 3 Value	1365	Off (0)	
In4	Input 4 Value	1366	Off (0)	
In5	Input 5 Value	1367	Off (0)	
In6	Input 6 Value	1368	Off (0)	
In7	Input 7 Value	1369	Off (0)	
In8	Input 8 Value	1370	Off (0)	
Out	Output Value	1371	Off (0)	

Lgc8.1 - 13 parameters

Parameter Name	Description	Available Values	Value Description
Oper	Operation	OFF (0)	Operator is turned off. **Default: Off**
		AND (1)	Output is ON when ALL inputs are ON.
		OR (2)	Output is ON when one or more of the 8 inputs are ON
		XOR (3)	Exclusive OR. The output is based on the inputs being cascaded XOR'd together (True logical XOR equation) that is, Cascading XOR performs an odd parity function, so if an even number of inputs are on, the output is off. If and odd number of inputs are on, the output is on.
NumIn	Number of Inputs	This parameter is used to configure the number of inputs for the operation. **Default: 2**	
InInvert	Invert Selected Inputs		Invert selected inputs. This is a status word with one bit per input. 0x1 - input 1 0x2 - input 2 0x4 - input 3 0x8 - input 4 0x10 - input 5 0x20 - input 6 0x40 - input 7 0x80 - input 8
OutInvert	Invert the Output	No (0)	Output not inverted. **Default: No(0)**
		Yes (1)	Output inverted.

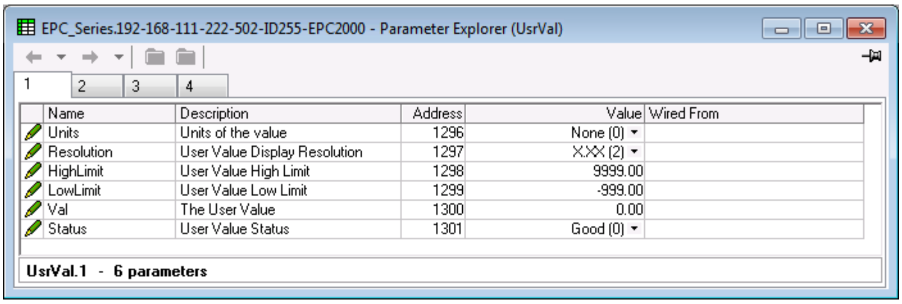
Parameter Name	Description	Available Values	Value Description
In1 **to** In8	Input1 **to** Input8 Value	Normally wired to a logic, analog or user value. All values are interpreted as follows: <0.5 = Off, >=0.5 = On May be set to a constant value if not wired.	
		Off (0)	Input is false.
		On (1)	Input is true.
Out	Output Value	Off (0)	Output result of the operator (output not activated).
		On (1)	Output result of the operator (output activated).

UsrVal

User values (UsrVal) function blocks are registers provided for use in calculations. They may be used as constants in equations or temporary storage in extended calculations.

User Values are only available if the Toolkit option has been ordered.

There are four instances of User Values available. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (UsrVal)

Name	Description	Address	Value	Wired From
Units	Units of the value	1296	None (0)	
Resolution	User Value Display Resolution	1297	X.XX (2)	
HighLimit	User Value High Limit	1298	9999.00	
LowLimit	User Value Low Limit	1299	-999.00	
Val	The User Value	1300	0.00	
Status	User Value Status	1301	Good (0)	

UsrVal.1 - 6 parameters

Parameter Name	Description	Available Values	Value Description
Units	Units of the value	See section "Units" on page 93 for a list of units used throughout.	
Resolution	User Value Display Resolution	X (0)	User value resolution.
		X.X (1)	One decimal place.
		X.XX (2)	Two decimal places. **Default: X.XX (2)**
		X.XX (3)	Three decimal places.
		X.XXX (4)	Four decimal places.
HighLimit	User Value High Limit	The high limit may be set for each user value to help prevent the value being set to an out-of-bounds value. Range Low Limit to Max 32-bit floating point value (decimal point depends on resolution). **Default: 99999**	
LowLimit	User Value Low Limit	The low limit of the user value may be set to help prevent the user value from being edited to an illegal value. This is important if the user value is to be used as a setpoint. Range Min 32-bit floating point value to High Limit (decimal point depends on resolution). **Default: -99999**	
Val	The User Value	To set the value within the range limits. See note below.	
Status	User Value Status	Can be used to force a good or bad status onto a user value. This is useful for testing status inheritance and fallback strategies. See note below. See section "Status" on page 94 for a list of enumerated values.	

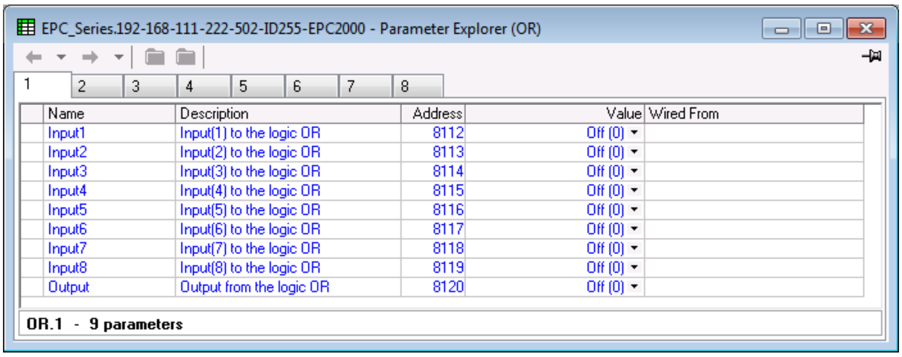
Note: If ‘Value’ parameter is wired but ‘Status’ parameter is not wired, then, instead of being used for force the Status, it will indicate the status of the value which is inherited from the wired connection to the ‘Value’ parameter.

OR (Logic OR)

The Logic OR function block allows multiple parameters to be wired to a single Boolean parameter without the need to enable toolkit blocks for the LGC2 or LGC8 'OR' functionality.

There are eight Logic OR blocks available.

Each block consists of eight inputs which are OR’d together into one output. It may be used, for example, to take the outputs from a number of alarm blocks and OR them together to operate a single general alarm output. The figure below shows the parameters and the table which follows details each parameter.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (OR)
1
2
3
4
5
6
7
8

Name	Description	Address	Value	Wired From
Input1	Input(1) to the logic OR	8112	Off (0)	
Input2	Input(2) to the logic OR	8113	Off (0)	
Input3	Input(3) to the logic OR	8114	Off (0)	
Input4	Input(4) to the logic OR	8115	Off (0)	
Input5	Input(5) to the logic OR	8116	Off (0)	
Input6	Input(6) to the logic OR	8117	Off (0)	
Input7	Input(7) to the logic OR	8118	Off (0)	
Input8	Input(8) to the logic OR	8119	Off (0)	
Output	Output from the logic OR	8120	Off (0)	

OR.1 - 9 parameters

Parameter Name	Description	Available Values	Value Description
Input1	Input 1 to the logic OR	Off (0)	Input 1 to the OR block. **Default: Off**
		On (1)	
Input2	Input 2 to the logic OR	Off (0)	Input 2 to the OR block. **Default: Off**
		On (1)	
Input3	Input 3 to the logic OR	Off (0)	Input 3 to the OR block. **Default: Off**
		On (1)	
Input4	Input 4 to the logic OR	Off (0)	Input 4 to the OR block. **Default: Off**
		On (1)	
Input5	Input 5 to the logic OR	Off (0)	Input 5 to the OR block. **Default: Off**
		On (1)	
Input6	Input 6 to the logic OR	Off (0)	Input 6 to the OR block. **Default: Off**
		On (1)	
Input7	Input 7 to the logic OR	Off (0)	Input 7 to the OR block. **Default: Off**
		On (1)	
Input8	Input 8 to the logic OR	Off (0)	Input 8 to the OR block. **Default: Off**
		On (1)	
Output	Output from the logic OR	Off (0)	Output result
		On (1)	

Programmer

A programmer provides a means of varying the setpoint in a controlled manner over a set period of time. This varying setpoint can then be used in the control process.

The EPC2000 Programmable Controller can support up to 20 stored programs; the actual number of programs is dependent upon the software option purchased, which is protected by feature security. The Programmer options are:

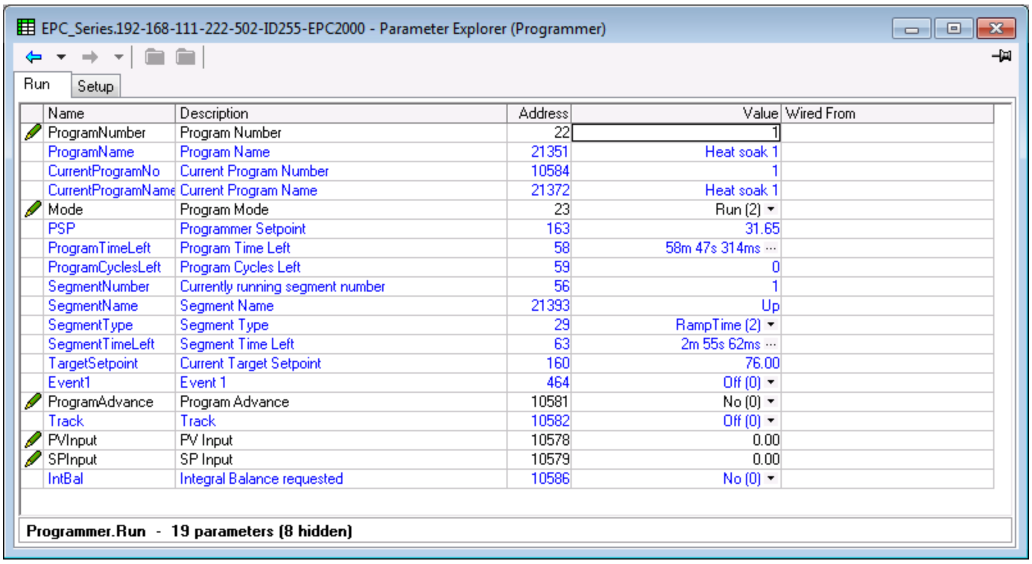
- disabled.
- 1 x 8 Basic Programmer (One program of eight configurable segments).
- 1 x 24 Advanced Programmer (One program of 24 configurable segments with up to eight event outputs).
- 10 x 24 Advanced Programmer (Ten programs of 24 configurable segments with up to eight event outputs).
- 20 x 8 Advanced Programmer (Twenty programs of eight configurable segments with up to eight event outputs).
- For all options, an additional End segment is provided which may also have event outputs if it is an Advanced Programmer.

For further information about setting up the Programmer using iTools, refer to "Programmer" on page 73. For full details on the Programmer functionality, refer to the chapter, "Programmer" on page 189.

There are two sets of parameters that can be used to control and monitor Programmer behavior: Run and Setup parameter lists.

Programmer.Run

The Run parameters are used to monitor and control the running program and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Programmer)

	Name	Description	Address	Value	Wired From
✎	ProgramNumber	Program Number	22	1	
	ProgramName	Program Name	21351	Heat soak 1	
	CurrentProgramNo	Current Program Number	10584	1	
	CurrentProgramName	Current Program Name	21372	Heat soak 1	
✎	Mode	Program Mode	23	Run (2)	
	PSP	Programmer Setpoint	163	31.65	
	ProgramTimeLeft	Program Time Left	58	58m 47s 314ms ···	
	ProgramCyclesLeft	Program Cycles Left	59	0	
	SegmentNumber	Currently running segment number	56	1	
	SegmentName	Segment Name	21393	Up	
	SegmentType	Segment Type	29	RampTime (2)	
	SegmentTimeLeft	Segment Time Left	63	2m 55s 62ms ···	
	TargetSetpoint	Current Target Setpoint	160	76.00	
	Event1	Event 1	464	Off (0)	
✎	ProgramAdvance	Program Advance	10581	No (0)	
	Track	Track	10582	Off (0)	
✎	PVInput	PV Input	10578	0.00	
✎	SPInput	SP Input	10579	0.00	
	IntBal	Integral Balance requested	10586	No (0)	

Programmer.Run - 19 parameters (8 hidden)

Parameter Name	Description	Available Values	Value Description
ProgramNumber	Program Number		The number of the program to be run.
ProgramName	Program Name		The name of the program to be run.
CurrentProgramNo	Current Program Number		The number of the program currently running.
CurrentProgramName	Current Program Name		The name of the program currently running.
Mode	Program Mode	Allows users to perform actions to change the current program state (Run, Hold, Reset – additionally indicates when a program is in Holdback or has Completed).	
		Reset (1)	**Default: Reset (1)**
		Run (2)	
		Hold (4)	
		Holdback (8)	
		Complete (16)	
PSP	Programmer Setpoint		Current setpoint in the program.
ProgramTimeLeft	Program Time Left		The amount of time left in the current program or -1 if the program cycles is set to ‘Continuous’.
ProgramCyclesLeft	Program Cycles Left		The number of cycles left in the current program or -1 if the program cycles is to set ‘Continuous’.
SegmentNumber	Currently running segment number		The number of the currently running segment.
SegmentName	Segment Name		The name of the currently running segment.

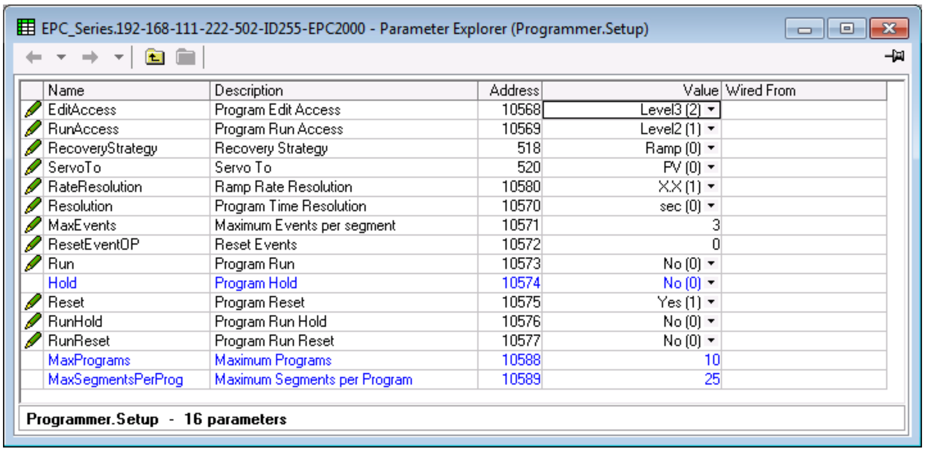
Parameter Name	Description	Available Values	Value Description
SegmentType	Segment Type	The type of the currently running segment:	
		End (0)	The very last segment in a program.
		RampRate (1)	This is specified by a target setpoint and a rate at which to ascend/descend to the setpoint.
		RampTime (2)	This is specified by a target setpoint and a time in which to achieve the ramp to setpoint.
		Dwell (3)	This is specified by how long the setpoint is to be maintained.
		Step (4)	This allows a step change in the target setpoint. **Note**: The step will occur immediately followed by a 1 second dwell period to allow event outputs to be set.
		Call (5)	This allows the main program to call another program as a subroutine. The number of times the program is called will be configurable, 1…9999. A program can only call other programs with a program number greater than itself, this prevents cyclic programs from being created. This segment type will only be available if multiple programs are enabled via Feature Security, and it should be noted that all configurable segments (1-24) can be configured as a call segment.
SegmentTimeLeft	Segment Time Left		The remaining time for the segment to complete.
TargetSetpoint	Current Target Setpoint		The target setpoint for the current segment.
RampRate	Segment Ramp Rate		The current ramp rate to achieve the target setpoint.
Event (n)	Event (n)	Event Output (n) value for the current segment.	
		Off (0)	The event is Off.
		On (1)	The event is On.
ProgramAdvance	Program Advance	Sets the Programmer Setpoint to the target setpoint of the current segment and advances to the next segment in the program.	
		No (0)	Default value.
		Yes (1)	Advances segment to the next one with the Programmer Setpoint assuming the Target Setpoint of the original segment.
Track	Track	Output parameter which is typically wired to the Loop Track parameter which is used to force the Loop into Track mode when the program completes and the program end type has been configured to Track.	
		Off (0)	Default value. Program has not completed.
		On (1)	Program has completed.
PVInput	PV Input		The PV Input used for Servo to PV, typically wired from the Loop’s Track PV parameter.
SPInput	SP Input		The SP Input used for Servo to SP, typically wired from the Loop’s Track SP parameter.
IntBal	Integral Balance Requested		This flag is set briefly when the Programmer performs a servo to PV, which would require the Loop to perform an integral balance in order to stop the working output from reacting to the change in setpoint. This parameter should be wired into the Loop.Main.IntBal parameter.
		No (0)	Integral balance is not requested.
		Yes (1)	Integral balance is requested.

Programmer.Setup

The Setup parameters are used to configure the Programmer behavior that is not likely to change. In addition to this, the Setup list also contains digital parameters that can be wired to Run, Reset and Hold a program.

Refer to the chapter, "Programmer" on page 189, for further details of the Programmer function.

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Programmer.Setup)

Name	Description	Address	Value	Wired From
EditAccess	Program Edit Access	10568	Level3 (2)	
RunAccess	Program Run Access	10569	Level2 (1)	
RecoveryStrategy	Recovery Strategy	518	Ramp (0)	
ServoTo	Servo To	520	PV (0)	
RateResolution	Ramp Rate Resolution	10580	X.X (1)	
Resolution	Program Time Resolution	10570	sec (0)	
MaxEvents	Maximum Events per segment	10571	3	
ResetEventOP	Reset Events	10572	0	
Run	Program Run	10573	No (0)	
Hold	Program Hold	10574	No (0)	
Reset	Program Reset	10575	Yes (1)	
RunHold	Program Run Hold	10576	No (0)	
RunReset	Program Run Reset	10577	No (0)	
MaxPrograms	Maximum Programs	10588	10	
MaxSegmentsPerProg	Maximum Segments per Program	10589	25	

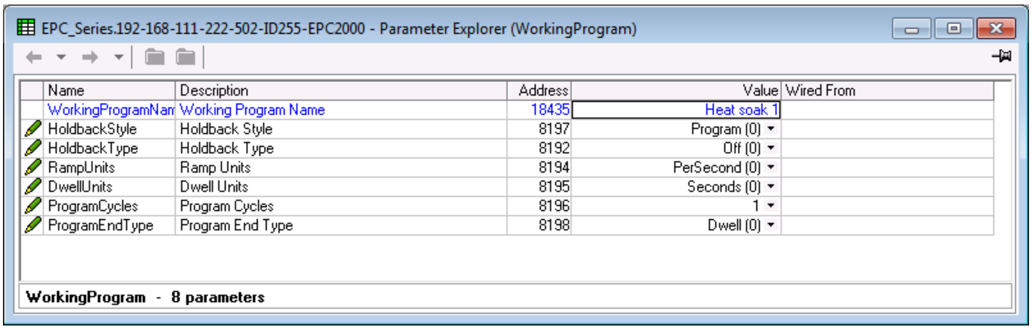
Programmer.Setup - 16 parameters

Parameter Name	Description	Available Values	Value Description
ProgrammerType	Programmer Type	The type of programmer:	
		Disabled (0)	
		1x8 (1)	Single program of up to eight segments.
		1x24 (2)	Single program of up to 24 segments.
		10x24 (3)	Up to ten programs of up to 24 segments.
		20x8 (4)	Up to 20 programs of up to eight segments.
EditAccess	Program Edit Access	Sets the minimum user access level allowed to edit programs. Not applicable to the EPC2000 Programmable Controller.	
		Level1 (0)	
		Level2 (1)	**Default: Level2 (1)**
		Level3 (2)	
		Config (4)	
RunAccess	Program Run Access	Sets the minimum user access level allowed to run programs. Not applicable to the EPC2000 Programmable Controller.	
		Level1 (0)	
		Level2 (1)	**Default: Level2 (1)**
		Level3 (2)	
RecoveryStrategy	Recovery Strategy	Configures the Power Fail and Sensor Break recovery strategy.	
		Ramp (0)	
		Reset (1)	**Default: Reset (1)**
		Track (2)	
ServoTo	Servo To	Configures the programmer to start from either the PV Input or the SP Input.	
		PV (0)	**Default: PV (0)**
		SP (1)	

Parameter Name	Description	Available Values	Value Description
RateResolution	Ramp Rate Resolution	Configures the resolution (number of decimal places) of the rate used in ramp rate segments. **Not applicable to the EPC2000 Programmable Controller**.	
		X (0)	
		X.X (1)	**Default: X.X (1)**
		X.XX (2)	
		X.XXX (3)	
		X.XXXX (4)	
Resolution	Program Time Resolution	Configures the resolution of the Program and Segment Time Left when read via comms as a Scaled Integer value.	
		sec (0)	**Default: sec (0)**
		min (1)	
		hour (2)	
MaxEvents	Maximum Events per segment	Range (0 to 8)	**Default: 1** **Note**: This parameter is not available if Programmer Type is 1x8.
ResetEventOP	Reset Events	Range (0 to 8)	Defines the event output states when the program is in the reset state.
Run	Program Run	The digital input to start the program running.	
		No (0)	
		Yes (1)	
Hold	Program Hold	The digital input to hold the running program.	
		No (0)	
		Yes (1)	
Reset	Program Reset	The digital input to reset (abort) the running program.	
		No (0)	
		Yes (1)	
RunHold	Program Run Hold	Dual functionality digital input, going from LOW to HIGH will start the program, whilst LOW the program is in Hold.	
		No (0)	
		Yes (1)	
RunReset	Program Run Reset	Dual functionality digital input, going from LOW to HIGH will start the program, whilst LOW the program is in Reset.	
		No (0)	
		Yes (1)	
MaxPrograms	Maximum Programs	Range (1 to 20)	The maximum number of programs allowed. This is defined by ProgrammerType parameter.
MaxSegmentsPerProg	Maximum Segments per Program	Range (1 to 24)	The maximum number of programs allowed. This is defined by ProgrammerType parameter.

WorkingProgram

The WorkingProject function block is only visible when the controller is in Operator level and a program is actually running. The function block is used to define parameters that are global to the program. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (WorkingProgram)

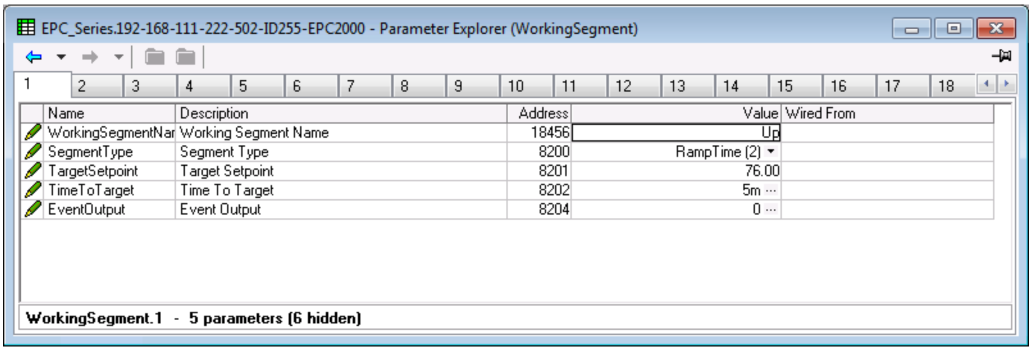
Name	Description	Address	Value	Wired From
WorkingProgramNam	Working Program Name	18435	Heat soak 1	
HoldbackStyle	Holdback Style	8197	Program (0)	
HoldbackType	Holdback Type	8192	Off (0)	
RampUnits	Ramp Units	8194	PerSecond (0)	
DwellUnits	Dwell Units	8195	Seconds (0)	
ProgramCycles	Program Cycles	8196	1	
ProgramEndType	Program End Type	8198	Dwell (0)	

WorkingProgram - 8 parameters

Parameter Name	Description	Available Values	Value Description
WorkingProgramName	Working Program Name	A text field which contains the name of the currently running program. The default name is the character ‘P’ followed by the program number. If you have renamed the program, that name is displayed here instead.	
HoldbackStyle	Holdback Style	Holdback is when the PV deviates from the setpoint by more than the holdback value, the program will temporarily hold until the PV returns to within the specified value. It is possible to set holdback either for the entire program or on a per-segment basis.	
		Program (0)	Default: Holdback set for the entire program
		Segment (1)	Holdback set for the segment only
HoldbackType	Holdback Type	Holdback prohibits the program from advancing faster than the load can react. Holdback continuously monitors the difference between the PV and the programmer setpoint. The holdback type specifies whether holdback tests for deviations above, below, or above and below the setpoint.	
		Off (0)	**Default: Off. No holdback tests are performed**
		Low (1)	Holdback tests for deviations below the setpoint.
		High (2)	Holdback tests for deviations above the setpoint.
		Band (3)	Holdback tests for deviations above and below the setpoint.
HoldbackValue	Holdback Value	A holdback value can be entered so that if the program setpoint differs from the PV value by the entered holdback value, the program will pause until the PV has caught up. This feature is useful for achieving the soak times of dwell segments. i.e, the Dwell does not start until the PV has reached the target setpoint. In the Programmer, the holdback value can be set once per program or for every segment depending on the holdback style set. It is possible to select whether the holdback is disabled or applied from above, from below, or both directions.	
RampUnits	Ramp Units	Ramp units can be defined as per second, per minute, or per hour. Ramp units are set for the entire program. Changing ramp units will convert the ramp rate parameter values of all Ramp Rate segments in the program.	
		PerSecond (0)	**Default: PerSecond(0).** Ramp units are defined as per second.
		PerMinute (1)	Ramp units are defined as per minute.
		PerHour (2)	Ramp units are defined as per hour.
DwellUnits	Dwell Units	Dwell units can be defined as per second, per minute, or per hour. Dwell units are set for the entire program.	
		PerSecond (0)	**Default: PerSecond(0).** Dwell units are defined as per second.
		PerMinute (1)	Dwell units are defined as per minute.
		PerHour (2)	Dwell units are defined as per hour.
ProgramCycles	Program Cycles	If a program is called from another program, then this value is ignored and the Call segment ‘Call Cycles’ parameter defines the number of sub-program loops.	
		Continuous (-1)	The program cycles continuously.
		1-9999	**Default.** The program cycles this number of times.
ProgramEndType	Program End Type	Defines the action to be taken after the final segment.	
		Dwell (0)	The programmer setpoint is maintained indefinitely and the event outputs remain at the states configured for the end segment.
		Reset (1)	The program is reset at the programmer setpoint will servo either to the PVInput value or the SPInput value as configured by the Programmer.Setup.ServoTo parameter. The event outputs will return to the states specified by the Programmer.Setup.ResetEventOP parameter.
		Track (2)	The programmer setpoint is maintained indefinitely and event outputs remain at the states configured for the end segment. If the programmer is wired to the Loop, the Loop will be forced into Track mode.

WorkingSegment

The WorkingSegment function block is only visible when the controller is in Operator level and a program is actually running. The function block is used to define the behavior of the working segments. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (WorkingSegment)

Name	Description	Address	Value	Wired From
WorkingSegmentNar	Working Segment Name	18456	Up	
SegmentType	Segment Type	8200	RampTime (2)	
TargetSetpoint	Target Setpoint	8201	76.00	
TimeToTarget	Time To Target	8202	5m ···	
EventOutput	Event Output	8204	0 ···	

WorkingSegment.1 - 5 parameters (6 hidden)

Parameter Name	Description	Available Values	Value Description
WorkingSegmentName	Working Segment Name	A text field which contains the name of the working segment. The default name is the character ‘S’ followed by the working segment number. If you have named the segments, that name is displayed here instead.	
SegmentType	Segment Type	Specifies the type of the current segment.	
		End (0)	**Default:** The current segment is of type ‘End’.
		Ramp Rate(1)	The current segment is of type ‘Ramp Rate’
		Ramp Time (2)	The current segment is of type ‘Ramp Time’
		Dwell (3)	The current segment is of type ‘Dwell’
		Step (4)	The current segment is of type ‘Step’
		Call (5)	The current segment is of type ‘Call’
TargetSetpoint	Target SetPoint	The define the desired setpoint to be achieved by the end of the segment.	
Duration	Dwell Duration	A dwell segment is specified by a duration, the length of time that the setpoint (inherited from the previous segment) is to be maintained.	
RampRate	Ramp Rate	Specifies the rate at which the setpoint is to be achieved. The units of the ramp rate (per second, per minute, or per hour) is specified by the Program parameter ‘RampUnits’.	
TimeToTarget	Time to Target	For Time to Target ramp segments, this parameter specifies the time to achieve the setpoint.	
CallCycles	Call Cycles	Defines the number of times the sub-program is run. To loop continuously, set cycles to 0 (Continuous).	
		Continuous (0)	The sub-program runs continuously.
		1-9999	**Default: 1.** The number of times the sub-program is to run.
EventOutput	Event Output	Defines the event output states. These event states can be wired to physical outputs for driving external events.	
HoldbackType	Holdback Type	Holdback prohibits the program from advancing faster than the load can react. Holdback continuously monitors the difference between the PV and the programmer setpoint. The holdback type specifies which type of deviation to check.	
		Off (0)	**Default: Off.** No holdback tests are performed.
		Low (1)	Holdback tests for deviations below the setpoint.
		High (2)	Holdback tests for deviations above the setpoint.
		Band (3)	Holdback tests for deviations above and below the setpoint.

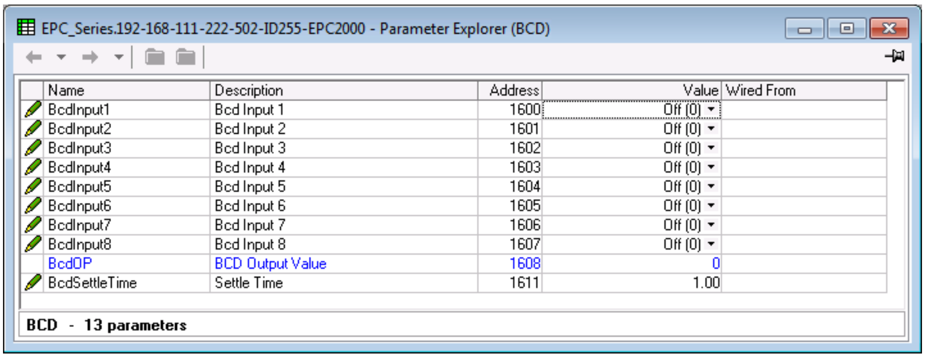
Parameter Name	Description	Available Values	Value Description
HoldbackValue	Holdback Value	A holdback value can be entered so that if the program setpoint differs from the PV by the entered holdback value, the program will pause until the PV has caught up. This feature is useful for achieving the soak times of dwell segments. i.e, the Dwell does not start until the PV has reached the target setpoint. In the Programmer, the holdback value can be set once per program or for every segment depending on the holdback style set.	
CallProgram	Call Program	The sub-program to call. This only applies to call segments. Only program numbers greater than the calling program can be called.	

BCD

The Binary Coded Decimal (BCD) Input Function block takes eight digital inputs and combines them to make a single numeric value, typically used to select a program or a recipe. The value generated is restricted to the decimal range of 0-9, any combinations which result in a value greater than 0 are truncated to 9.

The block uses four bits to generate a single digit. Two groups of four bits are used to generate a two digit value (0 to 99)

The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (BCD)

Name	Description	Address	Value	Wired From
BcdInput1	Bcd Input 1	1600	Off (0)	
BcdInput2	Bcd Input 2	1601	Off (0)	
BcdInput3	Bcd Input 3	1602	Off (0)	
BcdInput4	Bcd Input 4	1603	Off (0)	
BcdInput5	Bcd Input 5	1604	Off (0)	
BcdInput6	Bcd Input 6	1605	Off (0)	
BcdInput7	Bcd Input 7	1606	Off (0)	
BcdInput8	Bcd Input 8	1607	Off (0)	
BcdOP	BCD Output Value	1608	0	
BcdSettleTime	Settle Time	1611	1.00	

BCD - 13 parameters

Parameter Name	Description	Available Values	Value Description
BcdInput1	Bcd Input 1	Off (0)	
		On (1)	Digital input 1
BcdInput2	Bcd input 2	Off (0)	
		On (1)	Digital input 2
BcdInput3	Bcd input 3	Off (0)	
		On (1)	Digital input 3
BcdInput4	Bcd input 4	Off (0)	
		On (1)	Digital input 4
BcdInput5	Bcd input 5	Off (0)	
		On (1)	Digital input 5
BcdInput6	Bcd input 6	Off (0)	
		On (1)	Digital input 6
BcdInput7	Bcd input 7	Off (0)	
		On (1)	Digital input 7
BcdInput8	Bcd input 8	Off (0)	
		On (1)	Digital input 8
BcdOP	BCD Output Value	Reads the value (in BCD) of the switch as it appears on the digital inputs. See the examples in the table below.	
BcdSettleTime	Settle Time	As a BCD switch is turned from the current value to another, intermediate values may be seen on the output parameters of the block. These could cause issues in some applications. The Settle Time can be used to filter out these intermediate values by applying a settling period between the inputs changing and the converted values appearing at the outputs. **Default: 1s**	

in1	In2	In3	In4	In5	In6	In7	In8	BCD.OP
1	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	9
0	0	0	0	1	0	0	1	90
1	0	0	0	1	0	0	1	91
1	0	0	1	1	0	0	1	99

Loop

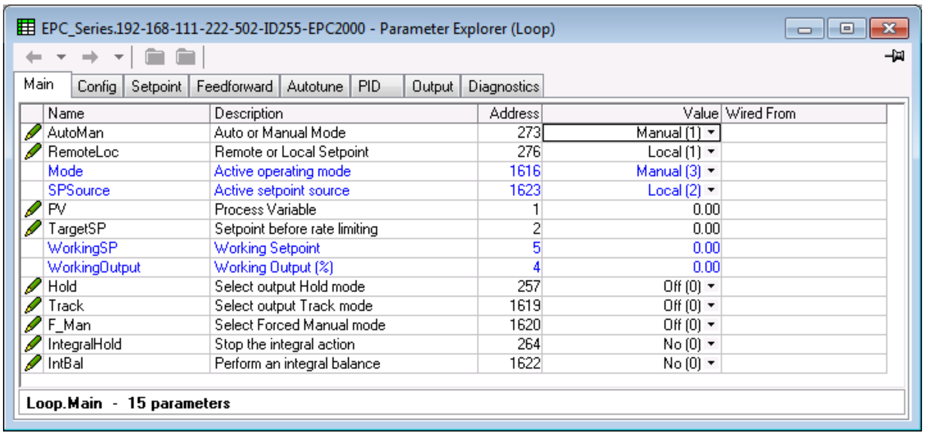
The Loop category contains and coordinates the various control and output algorithms, often to control the temperature of a process. The actual temperature measured at the process (PV) is connected to the input of the controller. This is compared with a setpoint (or required) temperature (SP). The controller calculates an output value to call for heating or cooling so that the difference between set and measured temperature is minimized. The calculation depends on the process being controlled but normally uses a PID algorithm. The output(s) from the controller are connected to devices within the plant which deliver the demanded heating (or cooling). This, in turn, is detected by the temperature sensor. This is referred to as the control loop or closed loop control.

For detailed information of how the loop operates and further descriptions of parameters please refer to the chapter, "Control" on page 198.

The Loop category contains eight function blocks: Main, Configuration, Setpoint, Feedforward, Autotune, PID, Output, Diagnostics, and are described below.

Loop.Main

The Main function block defines how the control loop behaves under different modes. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)

	Name	Description	Address	Value	Wired From
✎	AutoMan	Auto or Manual Mode	273	Manual (1) ▾	
✎	RemoteLoc	Remote or Local Setpoint	276	Local (1) ▾	
	Mode	Active operating mode	1616	Manual (3) ▾	
	SPSource	Active setpoint source	1623	Local (2) ▾	
✎	PV	Process Variable	1	0.00	
✎	TargetSP	Setpoint before rate limiting	2	0.00	
	WorkingSP	Working Setpoint	5	0.00	
	WorkingOutput	Working Output (%)	4	0.00	
✎	Hold	Select output Hold mode	257	Off (0) ▾	
✎	Track	Select output Track mode	1619	Off (0) ▾	
✎	F_Man	Select Forced Manual mode	1620	Off (0) ▾	
✎	IntegralHold	Stop the integral action	264	No (0) ▾	
✎	IntBal	Perform an integral balance	1622	No (0) ▾	

Loop.Main - 15 parameters

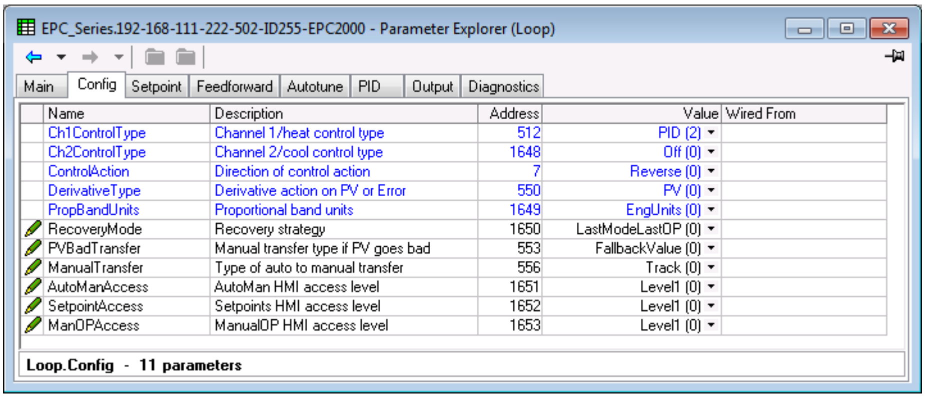
Parameter Name	Description	Available Values	Value Description
AutoMan	Auto or Manual Mode	Auto (0)	Select automatic (closed loop) control.
		Manual (1)	Select manual operation (output power adjusted by the user). **Default: Manual (1)**
RemoteLoc	Remote or Local Setpoint	Local (1)	Local Setpoint. In Auto mode, the loop uses one of its local setpoints (SP1/SP2) which is alterable over comms. **Default: Local (1)**
		Remote (0)	Remote Setpoint. This selects the remote setpoint source. This mode is commonly used, for example, in a cascade topology or with a multi-zone furnace. Although this parameter is used to select the remote setpoint, it will not necessarily become active. The RSP_En input must be true and the RSP must have a good status before it will become active. If either of these conditions are not met, the loop will fallback to using the local setpoint.

Parameter Name	Description	Available Values	Value Description
Mode	Active operating mode	Reports the currently active operating mode. The loop has a number of possible operating modes which can be selected by the application. It is possible for the application to request several modes at once, so the active mode is determined by a priority model, whereby the mode with the highest priority wins. The modes shown below are listed in their priority order.	
		Hold (0)	Hold. Priority 0: The controller Working Output will be maintained at its current value.
		Track (1)	Track. Priority 1: The controller output will follow the track output parameter. The track output can either be a constant value or be derived from an external source (e.g. an analog input).
		F_Man (2)	Forced Manual. Priority 2: This mode behaves in the same way as Manual but it indicates that Auto mode cannot currently be selected. This mode is selected if the PV status is not good (e.g. sensor break) and, optionally, if a process alarm has triggered. When transferring to Forced Manual from Auto mode, the output will go to the Fallback Value (unless the hold action has been selected). Transferring to Forced Manual from any other mode will be bumpless. This is used in a number conditions, further described in "Operating Modes" on page 214.
		Manual (3)	Manual. Priority 3: In manual mode, the controller passes authority over the output to the operator. The output is alterable over comms.
		Tune (4)	Tune. Priority 4: This mode indicates that the autotuner is running and has authority over the output.
		Auto (5)	Auto mode. Priority 5 (lowest): In Auto mode, the automatic control algorithm has authority over the output.
SPSource	Active setpoint source	Indicates the currently active setpoint source.	
		F_Local (0)	Forced local setpoint. The setpoint has fallen back to the local source because the remote setpoint is not being accessed correctly.
		Remote (1)	The setpoint is derived from a remote source.
		Local (2)	The setpoint is derived locally.
PV	Process Variable	The process variable. This is typically wired from an analog input.	
TargetSP	Setpoint before rate limiting	Adjust and display the current target setpoint. The target setpoint is the value before rate limiting.	
WorkingSP	Working Setpoint	Displays the current working setpoint. This setpoint may be derived from a number of sources depending on the application. Examples are from the programmer function block or a remote setpoint source.	
WorkingOutput	Working Output %	The current output demand in %.	
Hold	Select output Hold mode	Off (0)	When On is selected, the controller output will maintain its current value.
		On (1)	
Track	Select output Track mode	Off (0)	Used to select Track mode. In this mode, the controller output will follow the Track Output value. The Track output may be a constant value or come from an external source (e.g. an analog input). Track has priority 1 and so overrides all other modes apart from HOLD.
		On (1)	
F_Man	Select Forced Manual mode	Off (0)	When On is selected this mode behaves in the same way as Manual but whilst active it indicates that Auto cannot currently be selected. When transferring to this mode from Auto, and this input is asserted, the output will jump to the Fallback Value. This input can be wired to alarms or digital inputs and used during abnormal process conditions. This mode has priority 2 and so will override all modes, except for Hold and Track.
		On (1)	
When any of the above modes are selected they will be indicated by the Mode parameter above.			

Parameter Name	Description	Available Values	Value Description
IntegralHold	Stop the integral action	No (0)	
		Yes (1)	If asserted the integral component of the PID calculation will be frozen.
IntBal	Perform an integral balance	No (0) Yes (1)	This rising edge-triggered input can be used to force an integral balance. This recalculates the integral term in the controller so that the previous output is maintained, balancing out any change in the other terms. This can be used to minimize bumps in output when it is known that, for example, an artificial step change in PV will occur. For example, a compensation factor has just changed in an oxygen probe calculation. The integral balance is intended to help prevent any proportional or derivative kick, instead allowing the output to be smoothly adjusted under integral action.

Loop.Configuration

The configuration function block defines control type and how certain parameters behave for particular conditions. These parameters are unlikely to require changing once the application has been configured. The figure below shows the parameters and the table which follows details each parameter.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)
Main
Config
Setpoint
Feedforward
Autotune
PID
Output
Diagnostics

Name	Description	Address	Value	Wired From
Ch1ControlType	Channel 1/heat control type	512	PID (2)	
Ch2ControlType	Channel 2/cool control type	1648	Off (0)	
ControlAction	Direction of control action	7	Reverse (0)	
DerivativeType	Derivative action on PV or Error	550	PV (0)	
PropBandUnits	Proportional band units	1649	EngUnits (0)	
RecoveryMode	Recovery strategy	1650	LastModeLastOP (0)	
PVBadTransfer	Manual transfer type if PV goes bad	553	FallbackValue (0)	
ManualTransfer	Type of auto to manual transfer	556	Track (0)	
AutoManAccess	AutoMan HMI access level	1651	Level1 (0)	
SetpointAccess	Setpoints HMI access level	1652	Level1 (0)	
ManOPAccess	ManualOP HMI access level	1653	Level1 (0)	

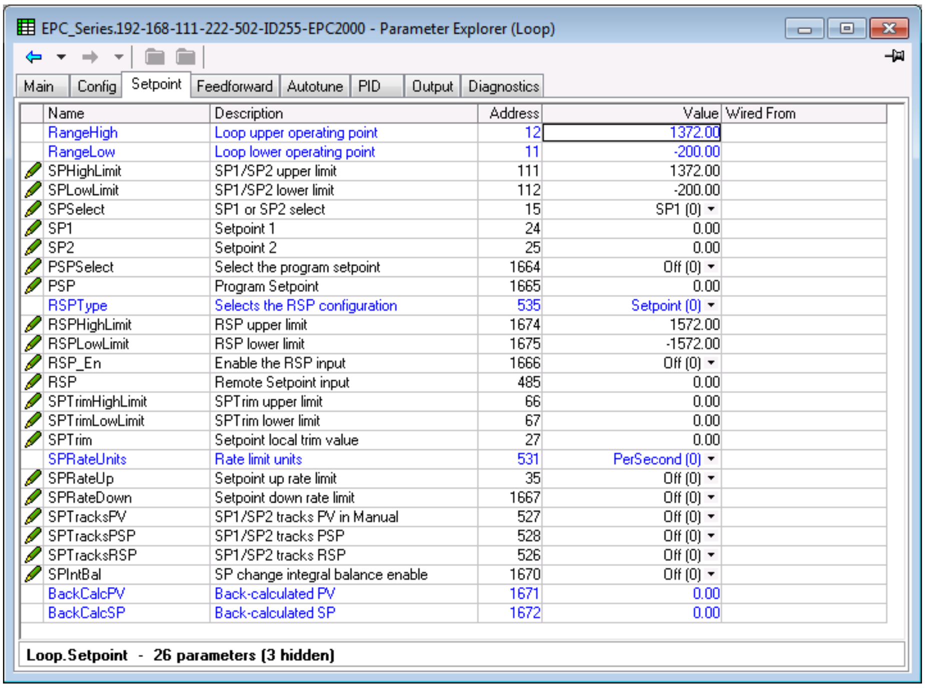
Loop.Config - 11 parameters

Parameter Name	Description	Available Values	Value Description
Ch1ControlType	Channel 1/heat control type	Off (0)	Control loop channel inoperative.
		OnOff (1)	On/Off control.
		PID (2)	PID Proportional, Integral, derivative, full three term control. **Default: PID (2)**
		VPU (3)	Valve position unbounded (no feedback potentiometer is required).
Ch2ControlType	Channel 1/cool control type	Off (0)	Control loop channel inoperative. **Default: Off**
		OnOff (1)	On/Off control.
		PID (2)	PID Proportional, Integral, derivative, full three term control.
		VPU (3)	Valve position unbounded (no feedback potentiometer is required).
ControlAction	Direction of control action	Reverse (0)	Reverse acting. Output decreases as PV increases. This is the normal setting for heating processes. Not applicable to on/off control. **Default: Reverse (0)**
		Direct (1)	Direct acting. Output increases as PV increases.

Parameter Name	Description	Available Values	Value Description
DerivativeType	Derivative action on PV or Error	PV (0)	Only changes in PV cause a derivative output. Generally used for process systems particularly using valve control where it reduces wear on valve mechanics. Not applicable to on/off control. **Default: PV (0)**
		Error (1)	Changes to either PV or SP will cause a derivative output. The derivative term will responds to the rate of change of the difference between PV and the setpoint. Not applicable to on/off control.
PropBandUnits		Proportional band units	EngUnits (0).
			The proportional band is set in percent of loop span (RangeHigh minus RangeLow).
RecoveryMode	Recovery strategy	This parameter configures the loop recovery strategy. This strategy is followed in the following circumstances: • Upon instrument start-up, after a power cycle or power outage. • Upon exit from Instrument Configuration or Standby conditions. • Upon exit from Forced Manual mode to a lower priority mode (e.g. when the PV recovers from a bad status or an alarm condition goes away).	
		LastModeLastOP (0)	Last Mode with Last Output. The loop will assume the last mode with the last output value. **Default: LastModeLastOP (0)**
		ManModeFallbackOP (1)	Manual Mode with Fallback Output. The loop will assume MANUAL mode with the fallback output value, unless exiting from Forced Manual in which case the current output will be maintained.
PVBadTransfer	Manual transfer type if PV goes bad	If the PV goes ’bad’ (e.g. due to a sensor break) this parameter configures the type of transfer to Forced Manual. Note that this is only followed if transitioning to Forced Manual from Auto. Transitioning from any other mode will be bumpless and transitioning due to the F_Man input being asserted will always go to the Fallback Value.	
		FallbackValue (0)	The FallbackValue will be applied to the output. **Default: FallbackValue (0)**
		Hold (1)	The last good output will be applied. This will be an output value from approximately 1 second before the transition.
ManualTransfer	Type of auto to manual transfer	Type of auto/manual transfer	
		Track (0)	The Manual Output will track the Working Output while the mode is not MANUAL. This helps to ensure a bumpless transfer when the mode does go into MANUAL. **Default: Track (0)**
		Step (1)	The Manual Output will be set to the Manual Step Value while the mode is not MANUAL.
		LastValue (2)	The Manual Output will remain at the last value used.
AutoManAccess	Automan HMI access level	**Not applicable to EPC2000 Programmable Controller.**	
SetpointAccess	Setpoints HMI access level	**Not applicable to EPC2000 Programmable Controller.**	
ManOPAccess	ManualOP HMI access level	**Not applicable to EPC2000 Programmable Controller.**	

Loop.Setpoint

The setpoint function block defines setpoint parameters such as limits, rates of change, trims and tracking strategies. The figure below shows the parameters and the table which follows details each parameter.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)
Main
Config
Setpoint
Feedforward
Autotune
PID
Output
Diagnostics

Name	Description	Address	Value	Wired From
RangeHigh	Loop upper operating point	12	1372.00	
RangeLow	Loop lower operating point	11	-200.00	
SPHighLimit	SP1/SP2 upper limit	111	1372.00	
SPLowLimit	SP1/SP2 lower limit	112	-200.00	
SPSelect	SP1 or SP2 select	15	SP1 (0)	
SP1	Setpoint 1	24	0.00	
SP2	Setpoint 2	25	0.00	
PSPSelect	Select the program setpoint	1664	Off (0)	
PSP	Program Setpoint	1665	0.00	
RSPType	Selects the RSP configuration	535	Setpoint (0)	
RSPHighLimit	RSP upper limit	1674	1572.00	
RSPLowLimit	RSP lower limit	1675	-1572.00	
RSP_En	Enable the RSP input	1666	Off (0)	
RSP	Remote Setpoint input	485	0.00	
SPTrimHighLimit	SPTrim upper limit	66	0.00	
SPTrimLowLimit	SPTrim lower limit	67	0.00	
SPTrim	Setpoint local trim value	27	0.00	
SPRateUnits	Rate limit units	531	PerSecond (0)	
SPRateUp	Setpoint up rate limit	35	Off (0)	
SPRateDown	Setpoint down rate limit	1667	Off (0)	
SPTracksPV	SP1/SP2 tracks PV in Manual	527	Off (0)	
SPTracksPSP	SP1/SP2 tracks PSP	528	Off (0)	
SPTracksRSP	SP1/SP2 tracks RSP	526	Off (0)	
SPIntBal	SP change integral balance enable	1670	Off (0)	
BackCalcPV	Back-calculated PV	1671	0.00	
BackCalcSP	Back-calculated SP	1672	0.00	

Loop.Setpoint - 26 parameters (3 hidden)

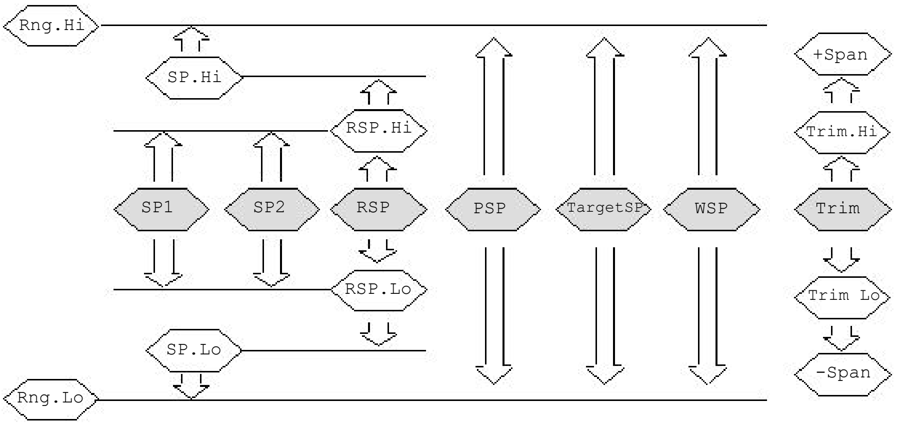
Parameter Name	Description	Available Values	Value Description
RangeHigh	Loop upper operating point	Range high limit. Selectable between the high limit of the selected input type to the ‘RangeLow’ limit parameter. **Default: 1372.0**	
RangeLow	Loop lower operating point	Range low limit. Selectable between the low limit of the selected input type to the ‘RangeHigh’ limit parameter.	
SPHighLimit	SP1/SP2 upper limit	Maximum allowable setpoint setting. Range is between ’RangeHigh’ limit and RangeLow’ limit. **Default: 1372.0**	
SPLowLimit	SP1/SP2 lower limit	Minimum allowable setpoint setting. Range is between ’RangeHigh’ limit and RangeLow’ limit.	
SPSelect	SP1 or SP2 select	SP1 (0)	Select setpoint 1. **Default: SP1 (0)**
		SP2 (1)	Select setpoint 2.
SP1	Setpoint 1	The current value of setpoint 1. Range from Low to High setpoint limits.	
SP2	Setpoint 2	The current value of setpoint 2. Range from Low to High setpoint limits.	
PSPSelect	Select the program setpoint	Off (0)	Program setpoint not selected.
		On (1)	Program setpoint selected.
PSP	Program Setpoint	The current value of the programmer setpoint.	
RSPType	Selects the RSP configuration	This parameter configures the remote setpoint topology.	
		Setpoint (0)	The remote setpoint (RSP) is used as a setpoint for the control algorithm. If required, a local trim can be applied. **Default: Setpoint (0)**
		Trim (1)	The local setpoint (SP1/SP2) is used as the setpoint for the control algorithm. The remote setpoint (RSP) acts as a remote trim on this local setpoint.
RSPHighLImit	RSP upper limit	Sets the maximum range limit for the remote setpoint. **Default: 1572.0**	
RSPLowLimit	RSP lower limit	Sets the minimum range limit for the remote setpoint. **Default: -1572.0**	

Parameter Name	Description	Available Values	Value Description
RSP_En	Enable the RSP input	On (1)	This input is used to enable the remote setpoint (RSP). The remote setpoint cannot become active unless this input is asserted. This is typically used in a cascade arrangement and allows the master to signal to the slave that it is providing a valid output. I.e. the master controller's Loop.Diagnostics.MasterReady parameter should be wired into here.
		Off (0)	Disable the remote setpoint.
RSP	Remote Setpoint input	The remote setpoint (RSP) is typically used in a cascade control arrangement or in a multi-zone process, where a master controller is transmitting a setpoint to the slave. For the remote setpoint to become active, the RSP status must be ‘Good’, the RSP_En input must be ‘TRUE’ and RemLocal must be set to ‘Remote’. The RSP can either be used as a setpoint itself (with a local trim if required) or as a remote trim on a local setpoint.	
SPTrimHighLimit	SPTrim upper limit	Local setpoint trim upper limit. Lower range limit is set by SPTrimLowLimit.	
SPTrimLowLimit	SPTrim lower limit	Local setpoint trim lower limit. Upper range limit is set by SPTrimHighLimit.	
SPTrim	Setpoint local trim value	To adjust the value by which the setpoint is being trimmed between SPTrimHighLimit and SPTrimLowLimit.	
SPRateUnits	Rate limit units	PerSecond (0)	Set the setpoint rate limit to units per second, units per minute or units per hour. **Default: PerSecond (0)**
		PerMinute (1)	
		PerHour (2)	
SPRateUp	Setpoint up rate limit	Off (0)	Limits the rate at which the setpoint can increase when setpoint ramp rate is used. OFF means no rate limit is applied. **Default: Off**
SPRateDown	Setpoint down rate limit	Off (0)	Limits the rate at which the setpoint can decrease when setpoint ramp rate is used. OFF means no rate limit is applied. **Default: Off**
The next three parameters are only shown if either of the setpoint rate limit parameters above are set to a value.			
SPRateDisable	Disable setpoint rate limits	No (0)	Setpoint rate limit enabled.
		Yes (1)	Disable setpoint rate limit.
SPRateDone	Setpoint ramp complete	No (0) Yes (1)	Indicates that the working setpoint has reached the target setpoint. If the setpoint is subsequently changed it will ramp at the set rate until the new value is reached.
SPRateServo	Rate limit servo to PV enabled	When the setpoint is being rate limited and servo-to-PV is enabled, changing the target SP will cause the working SP to servo (step) to the current PV before ramping to the new target. This feature is only applied to SP1 and SP2 and not to the program or remote setpoints.	
		Off (0)	Disabled.
		On (1)	The selected setpoint will servo to the current value of the PV.
SPTracksPV	SP1/SP2 tracks PV in Manual	Off (0)	No setpoint tracking when in manual.
		On (1)	When the controller is operating in manual mode the currently selected SP (SP1 or SP2) tracks the PV. When the controller resumes automatic control there will be no step change in the working SP. Manual tracking does not apply to the remote setpoint or programmer setpoint.
SPTracksPSP	SP1/SP2 tracks PSP	Off (0)	No setpoint tracking of the programmer
		On (1)	SP1/SP2 tracks the programmer setpoint while the program is running, so that there will no step change in the Working SP when the program ends and the programmer is reset. This is sometimes referred to as ‘Program Tracking’.

Parameter Name	Description	Available Values	Value Description
SPTracksRSP	SP1/SP2 tracks RSP	On (1)	When the remote setpoint is selected SP1/SP2 tracks the remote setpoint, so that there will be no step change in the Working SP when transitioning to the local setpoint source. The selected setpoint returns to its set value at the rate set by the SPRateUp and SPRateDown parameters.
		Off (0)	Disabled.
SPIntBal	SP change integral balance enable	When enabled, this causes the control algorithm to perform an integral balance whenever the target setpoint is changed. It only applies when the local setpoint is in use. The effect of this option is to suppress proportional and derivative kicks whenever the setpoint changes, so that the output moves smoothly to its new value under integral action. This option is similar to having both proportional and derivative terms act on PV only, and not error.	
		Off (0)	Disabled
		On (1)	Enable. To suppress proportional and derivative kick.
BackCalcPV	Back-calculated PV	This output is the back-calculated PV. It is the value of PV minus the setpoint trim. This is typically wired to the PV input of a setpoint programmer. Wiring this input, rather than the PV itself, helps to ensure that the holdback feature can take account of any setpoint trim that may be applied and also allows setpoint programs to start smoothly with the working setpoint equal to the PV, if configured.	
BackCalcSP	Back-calculated SP	This output is the back-calculated SP. It is the working setpoint minus the setpoint trim. This is typically wired to the servo input of a setpoint programmer, so that it can start smoothly without bumping the working setpoint, if configured.	

Setpoint Limits

The figure below shows a pictorial overview of setpoint limits

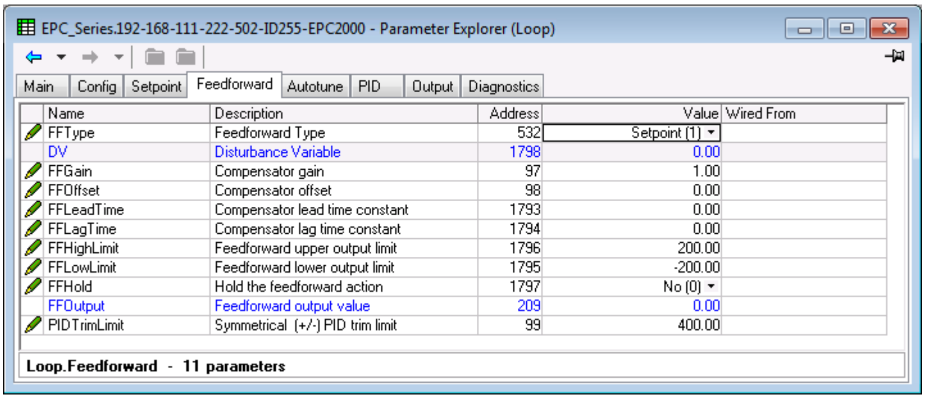


Span is taken to be the value given by Range High Limit - Range Low Limit.

Note: Whilst it is possible to set the RSP Limits outside the Range Limits, the RSP value will still be clipped to the Range Limits.

Loop.Feedforward

This function block defines the strategy to be adopted for a particular application. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)

Name	Description	Address	Value	Wired From
FFType	Feedforward Type	532	Setpoint (1)	
DV	Disturbance Variable	1798	0.00	
FFGain	Compensator gain	97	1.00	
FFOffset	Compensator offset	98	0.00	
FFLeadTime	Compensator lead time constant	1793	0.00	
FFLagTime	Compensator lag time constant	1794	0.00	
FFHighLimit	Feedforward upper output limit	1796	200.00	
FFLowLimit	Feedforward lower output limit	1795	-200.00	
FFHold	Hold the feedforward action	1797	No (0)	
FFOutput	Feedforward output value	209	0.00	
PIDTrimLimit	Symmetrical (+/-) PID trim limit	99	400.00	

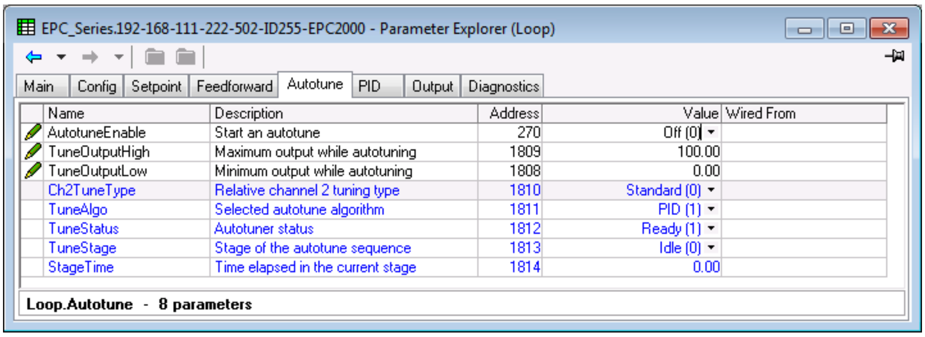
Loop.Feedforward - 11 parameters

Parameter Name	Description	Available Values	Value Description
FFType	Feedforward Type	Off (0)	No signal fed forward.
		Setpoint (1)	The working setpoint is used as the input to the feedforward compensator.
		PV (2)	The PV is used as the input to the feedforward compensator. This is sometimes used as an alternative to 'Delta-T' control.
		RemoteDV (3)	The remote Disturbance Variable (DV) is used as the input to the feedforward compensator. This is usually a secondary process variable that can be used to head-off disturbances in the PV before they have a chance to occur.
FFOutput	Feedforward output value	Feedforward compensator output in percent.	
The following parameters are available if FFType is not set to OFF (0)			
FFGain	Compensator gain	Defines the gain of the feedforward value, the feed forward value is multiplied by the gain. **Default: 1.0**	
FFOffset	Compensator offset	The feedforward compensator bias/offset. This value is added to the feedforward input. Note that the bias is applied after the gain.	
FFLeadTime	Compensator lead time constant	The feedforward compensator lead time constant in seconds can be used to 'speed-up' the feedforward action. Set to 0 to disable the lead component. In general, the lead component should not be used on its own without any lag. The lead and lag time constants allow dynamic compensation of the feedforward signal. The values are usually determined by characterizing the effect of the input on the process (e.g. by a bump test). In the case of a Disturbance Variable, the values are chosen so that the disturbance and the correction 'arrive' at the process variable at the same instant, thereby minimizing any perturbation. As a rule of thumb, the lead time should be set equal to the lag between the controller output and the PV, while the lag time should be set equal to the lag between the DV and the PV.	

Parameter Name	Description	Available Values	Value Description
FFLagTime	Compensator lag time constant	The feedforward compensator lag time constant can be used to slow down the feedforward action. Set to 0 to disable the lag component. The lead and lag time constants allow dynamic compensation of the feedforward signal. The values are usually determined by characterizing the effect of the input on the process (e.g. by a bump test). In the case of a Disturbance Variable, the values are chosen so that the disturbance and the correction 'arrive' at the process variable at the same instant, thereby minimizing any perturbation. As a rule of thumb, the lead time should be set equal to the lag between the controller output and the PV, while the lag time should be set equal to the lag between the DV and the PV.	
FFHighLimit	Feedforward upper output limit	The maximum allowed value of feedforward output. This limit is applied to the feedforward output prior to it being added to the PID output. **Default: 200.0%**	
FFLowLimit	Feedforward lower output limit	The minimum allowed value of feedforward output. This limit is applied to the feedforward output prior to it being added to the PID output. **Default: -200%**	
FFHold	Hold the feedforward action	No (0)	When true, the feedforward output will maintain its current value. This can be used to temporarily halt feedforward action.
		Yes (1)	
PIDTrimLimit	Symmetrical (+/-) PID trim limit	The PID trim limit limits the effect of the PID output. The implementation of Feedforward allows the Feedforward component to make the dominant contribution to the control output. The PID contribution can then be used as a trim on the Feedforward value. This arrangement is sometimes known as "Feedforward with Feedback Trim". This parameter defines symmetrical limits (expressed as a percentage of output) around the PID output, to limit the magnitude of the PID contribution. If it is required to allow PID contribution to dominate, set a large value for this parameter (400.0). **Default: 400.0**	
If FFType is set to Remote the following additional parameter is available			
DV	Disturbance Variable	The remote Disturbance Variable is typically a secondary measured process variable. This is usually a secondary process variable that can be used to head-off disturbances in the PV before they have a chance to occur.	

Loop.Autotune

The Autotune function block is used to automatically tune the PID loop to match the characteristics of the process. See also "Autotuning" on page 220. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)

Name	Description	Address	Value	Wired From
AutotuneEnable	Start an autotune	270	Off (0)	
TuneOutputHigh	Maximum output while autotuning	1809	100.00	
TuneOutputLow	Minimum output while autotuning	1808	0.00	
Ch2TuneType	Relative channel 2 tuning type	1810	Standard (0)	
TuneAlgo	Selected autotune algorithm	1811	PID (1)	
TuneStatus	Autotuner status	1812	Ready (1)	
TuneStage	Stage of the autotune sequence	1813	Idle (0)	
StageTime	Time elapsed in the current stage	1814	0.00	

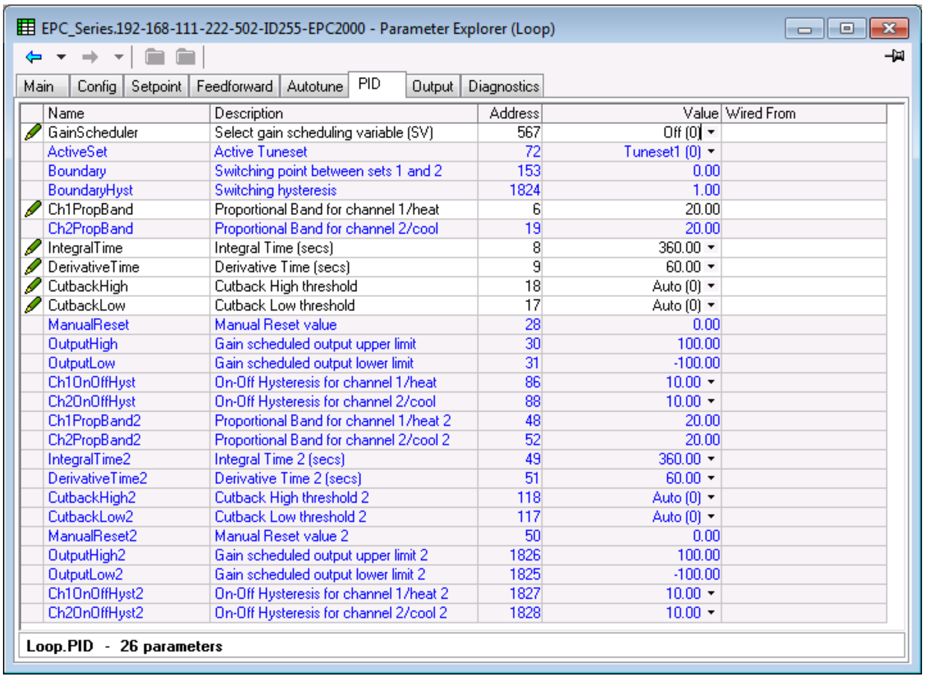
Loop.Autotune - 8 parameters

Parameter Name	Description	Available Values	Value Description
AutotuneEnable	Start an autotune	Off (0)	Autotune not enabled or abort an autotune.
		On (1)	Enable autotune.
TuneOutputHigh	Maximum output while autotuning	−100 to +100%	To set a maximum limit on the output during tuning. **Default: 100**
TuneOutputLow	Minimum output while autotuning	−100 to +100%	To set a minimum limit on the output during tuning. **Default: -100**
CH2TuneType	Relative channel 2 tuning type	Configures which experiment will be used to determine the relationship between channel 1 and channel 2 proportional bands.	
		Standard (0)	Standard. Tunes the ch2 proportional band using the standard relative ch2 tuning algorithm. **Default: Std**
		Alternative (1)	Alternative relative ch2 tuning. Uses a model-based tuning algorithm that has been shown to give enhanced results with higher-order, low-loss plants. In particular, it performs well with heavily-lagged temperature processes.
		KeepPBRatio (2)	Do not try to determine the relative gain. This option can be used to help prevent the autotune trying to determine the ch2 proportional band. Instead, it will maintain the existing ratio between the ch1 and ch2 proportional bands. In general, this option is not recommended unless there is a known reason to select it (e.g. the relative gain is already known and the tuner gives an incorrect value).
TuneAlgo	Selected autotune algorithm	This parameter reports which autotuning algorithm is available for the current control configuration. The appropriate tuning algorithm is automatically determined. See also "Autotuning" on page 220 for further information on self tuning.	
		None (0)	There is no autotuner available for the current control configuration.
		PID (1)	The standard autotuner based on a modified relay method. It requires two cycles to complete (not including the relative ch2 tune). This is used for PID-only configurations and where there is no output rate limiting configured.
		Fourier (2)	This algorithm uses the same modified relay method but uses a more complex analysis based on the work of Joseph Fourier. It requires three cycles to complete (not including the relative ch2 tune). This algorithm is used for VP or mixed channel configurations and is also used whenever there is an output rate limit set.

Parameter Name	Description	Available Values	Value Description
TuneStatus	Autotuner status	This parameter displays the current state of the autotune.	
		Unavailable (0)	Unavailable.
		Ready (1)	Ready to run an autotune.
		Triggered (2)	An autotune has been triggered but a higher priority mode is preventing it from starting. When the higher priority mode is no longer active the tune will start.
		Running (3)	The autotuner is running and currently has authority over the controller outputs.
		Complete (4)	The autotune successfully completed and has updated the tuneset parameters.
		Aborted (5)	Autotune aborted.
		Timeout (6)	If any stage of the autotune sequence exceeds two hours in duration, the sequence will time out and be aborted. It could be due to the loop being open or not responding to the demands from the controller. Very heavily lagged systems may produce a timeout if the cooling rate is very slow. The Stage Time parameter counts up the time in each stage.
		Overflow (7)	A buffer overflow occurred while collecting process data. Contact your supplier for support.
TuneStage	Stage of the autotune sequence	This reports the stage of the current autotuning sequence.	
		Idle (0)	Not autotuning.
		Monitor (1)	The process is being monitored. This stage lasts one minute. The setpoint can be changed during this stage.
		Initial (2)	An initial oscillation is being established.
		Max (3)	Maximum output applied
		Min (4)	Minimum output applied
		R2G (5)	Relative channel 2 gain test is running. If the calculated proportional band ratio is outside the range 0.1 and 10.0, The Ch1/Ch2 Proportional Band ratio will be clipped to these limits but all other PID parameters are updated. R2G limit may occur if the gain difference between heating and cooling is too large. This could also occur if the controller is configured for heat/cool but the cooling medium is turned off or not working correctly. It could similarly occur if the cooling medium is on but heating is off or not working correctly.
		PD (6)	The autotuner is trying to control to setpoint and is examining the response.
		Analysis (7)	The autotuner is calculating the new tuning parameters.
StageTime	Time elapsed in the current stage	The time elapsed in the current autotune stage. This is reset each time the autotuner advances a stage. If this exceeds two hours, a timeout will occur.	

Loop.PID

The PID function block is used to display and set the current PID values. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)
Main
Config
Setpoint
Feedforward
Autotune
PID
Output
Diagnostics

Name	Description	Address	Value	Wired From
GainScheduler	Select gain scheduling variable (SV)	567	Off (0)	
ActiveSet	Active Tuneset	72	Tuneset1 (0)	
Boundary	Switching point between sets 1 and 2	153	0.00	
BoundaryHyst	Switching hysteresis	1824	1.00	
Ch1PropBand	Proportional Band for channel 1/heat	6	20.00	
Ch2PropBand	Proportional Band for channel 2/cool	19	20.00	
IntegralTime	Integral Time (secs)	8	360.00	
DerivativeTime	Derivative Time (secs)	9	60.00	
CutbackHigh	Cutback High threshold	18	Auto (0)	
CutbackLow	Cutback Low threshold	17	Auto (0)	
ManualReset	Manual Reset value	28	0.00	
OutputHigh	Gain scheduled output upper limit	30	100.00	
OutputLow	Gain scheduled output lower limit	31	-100.00	
Ch1OnOffHyst	On-Off Hysteresis for channel 1/heat	86	10.00	
Ch2OnOffHyst	On-Off Hysteresis for channel 2/cool	88	10.00	
Ch1PropBand2	Proportional Band for channel 1/heat 2	48	20.00	
Ch2PropBand2	Proportional Band for channel 2/cool 2	52	20.00	
IntegralTime2	Integral Time 2 (secs)	49	360.00	
DerivativeTime2	Derivative Time 2 (secs)	51	60.00	
CutbackHigh2	Cutback High threshold 2	118	Auto (0)	
CutbackLow2	Cutback Low threshold 2	117	Auto (0)	
ManualReset2	Manual Reset value 2	50	0.00	
OutputHigh2	Gain scheduled output upper limit 2	1826	100.00	
OutputLow2	Gain scheduled output lower limit 2	1825	-100.00	
Ch1OnOffHyst2	On-Off Hysteresis for channel 1/heat 2	1827	10.00	
Ch2OnOffHyst2	On-Off Hysteresis for channel 2/cool 2	1828	10.00	

Loop.PID - 26 parameters

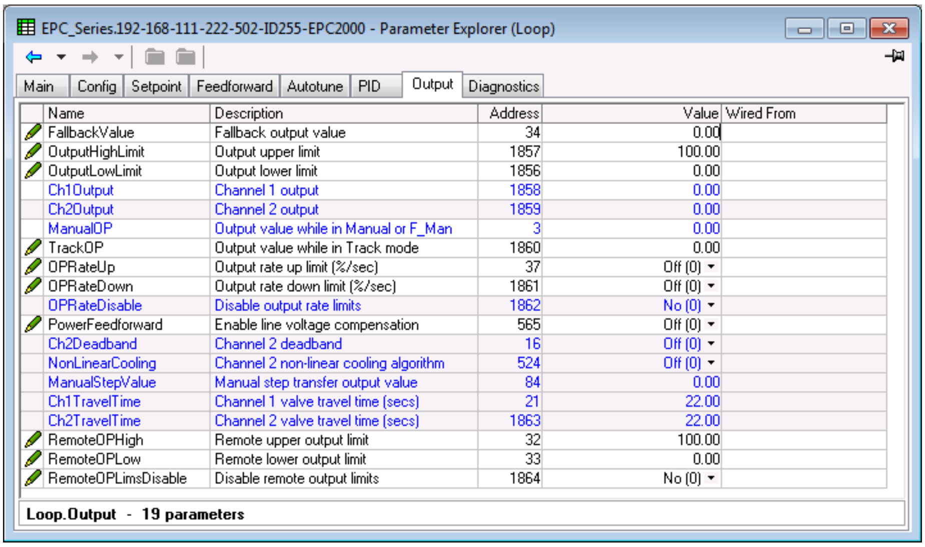
Parameter Name	Description	Available Values	Value Description
GainScheduler	Select gain scheduling variable (SV)	Gain scheduling is provided so that processes which change their characteristics can be controlled. For example, in some temperature processes, the dynamical response may be very different at low temperatures from that at high temperatures. Gain Scheduling typically uses one of the loop's parameters to select the active PID set - this parameter is known as the scheduling variable (SV). Two sets are available and a boundary is provided which defines the switching point.	
		Off (0)	Gain scheduling not active
		Set (1)	The PID set can be selected by the operator. It is possible to use soft wiring to control the selection of the gain sets. This could be linked to the programmer segment, changing the PID settings for individual segments or it could be wired to a digital input so that the working PID set can be set remotely.
		PV (2)	The transfer between one set and the next depends on the value of the process variable.
		Setpoint (3)	The transfer between one set and the next depends on the value of the working setpoint.
		Output (4)	The transfer between one set and the next depends on the value of the output.
		Deviation (5)	The transfer between one set and the next depends on the value of the difference between the SP and PV.
		Mode (6)	This parameter selects set 2 when the remote setpoint is active, and set 1 when the local setpoint is active.
ActiveSet	Active Tuneset	Tuneset1 (0)	This shows the set being tuned and is displayed if GainScheduler = Set, PV, Setpoint, Output or Deviation.
		Tuneset2 (1)	
Boundary	Switching point between sets 1 and 2	Sets the level at which PID set 1 changes to PID set 2. This only applies when the schedule type = PV, Setpoint, Output, or Deviation. **Default: 1.0**	

Parameter Name	Description	Available Values	Value Description
BoundaryHyst	Switching hysteresis	This specifies the amount of hysteresis around the gain scheduling boundary. This is used to avoid continuous switching as the scheduling variable passes through the boundary.	
Ch1PropBand	Proportional Band for channel 1/heat	The proportional band for channel 1. This may be in % or engineering units as set by the parameter PropBandUnits (in the Config function block). **Default: 20.0%**	
Ch2PropBand	Proportional Band for channel 2/cool	The proportional band for channel 2. This may be in % or engineering units as set by the parameter PropBandUnits (in the Config function block). **Default: 20.0%**	
IntegralTime	Integral Time (secs)	The integral time in seconds for channel 1. Set to 0 to disable integral action. **Default: 360 seconds**	
DerivativeTime	Derivative Time (secs)	The derivative time in seconds for channel 1. Set to 0 to disable derivative action. **Default: 60 seconds**	
CutbackHigh	Cutback High threshold	0	Defines a High Cutback threshold in the same units as the proportional band (either engineering units or percent of span, depending on configuration).
CutbackLow	Cutback Low threshold	0	Defines a Low Cutback threshold in the same units as the proportional band (either engineering units or percent of span, depending on configuration).
ManualReset	Manual reset value	Manual reset. This parameter only appears if the control algorithm is PID or VPU, AND the integral time is set to 0 (Off). It is used to manually adjust the output power to offset for any difference between SP and PV. See also "Manual Reset (PD Control)" on page 202.	
OutputHigh	Gain scheduled output upper limit	Gain scheduled output upper limit. **Default: 100**	
OutputLow	Gain scheduled output lower limit	Gain scheduled output lower limit. **Default: -100**	
Ch1OnOffHyst	On-Off Hysteresis for channel 1/heat	0	This parameter is only available if channel 1(heat) is configured for On/OFF control. It sets the hysteresis between the output being on and the output being off. **Default: 10**
Ch2OnOffHyst	On-Off Hysteresis for channel 2/cool	0	This parameter is only available if channel 2 (cool) is configured for On/OFF control. It sets the hysteresis between the output being on and the output being off. **Default: 10**
Ch1PropBand2	Proportional Band for channel 1/heat 2	The proportional band for channel 1, for tuneset 2. This may be in % or engineering units as set by the parameter PB.UNT. **Default: 20.0%**	
Ch2PropBand2	Proportional Band for channel 2/cool 2	The proportional band for channel 2, for tuneset 2. This may be in % or engineering units as set by the parameter PB.UNT. **Default: 20.0%**	
IntegralTime2	Integral Time 2 (secs)	The integral time in seconds for tuneset 2. Set to 0 to disable integral action. **Default: 360 seconds**	
DerivativeTime2	Derivative Time 2 (secs)	The derivative time in seconds for tuneset 2. Set to 0 to disable derivative action. **Default: 60 seconds**	
CutbackHigh2	Cutback High threshold 2	0	Defines a High Cutback threshold, for tuneset 2, in the same units as the proportional band (either engineering units or percent of span, depending on configuration).
CutbackLow2	Cutback Low threshold 2	0	Defines a Low Cutback threshold, for tuneset 2, in the same units as the proportional band (either engineering units or percent of span, depending on configuration).
ManualReset2	Manual Reset value 2	Manual reset for tuneset 2. This parameter only appears if the control algorithm is PID or VPU, AND the integral time is set to 0 (Off). It is used to manually adjust the output power to offset any for any difference between SP and PV. See also "Manual Reset (PD Control)" on page 202.	

Parameter Name	Description	Available Values	Value Description
OutputHigh2	Gain scheduled output upper limit 2	Gain scheduled output upper limit for tuneset 2. Range between +100.0% to OutputLow2.	
OutputLow2	Gain scheduled output lower limit 2	Gain scheduled output lower limit for tuneset 2. Range between −100.0% and OutputHigh2.	
Ch1OnOffHyst2	On-Off Hysteresis for channel 1/heat 2	0	On-Off Hysteresis for channel 1/heat, for tuneset 2. This is set in the units of the PV. It defines the point below setpoint where the channel 1 output will turn on. The output will turn off when the PV is at setpoint. The hysteresis is used to minimize the chattering of the output at the control setpoint. If the hysteresis is set to 0 then even the smallest change in the PV, when at setpoint, will cause the output to switch. The hysteresis should be set to a value which provides an acceptable life for the output contacts, but which does not cause unacceptable oscillations in the PV. If this performance is unacceptable, it is recommended that PID control with a time proportioning output is used. **Default: 10**
Ch2OnOffHyst2	On-Off Hysteresis for channel 2/cool 2	0	On-Off Hysteresis for channel 2/cool, for tuneset 2. This parameter is only available if channel 2 (cool) is configured for On/OFF control. It sets a second value of the hysteresis, for tuneset 2, between the output being on and the output being off. The comments above are also applicable to this parameter. **Default: 10**

Loop.Output

The Output function block is used to display and set the output parameters. The figure below shows the parameters and the table which follows details each parameter.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)

	Name	Description	Address	Value	Wired From
✎	FallbackValue	Fallback output value	34	0.00	
✎	OutputHighLimit	Output upper limit	1857	100.00	
✎	OutputLowLimit	Output lower limit	1856	0.00	
	Ch1Output	Channel 1 output	1858	0.00	
	Ch2Output	Channel 2 output	1859	0.00	
	ManualOP	Output value while in Manual or F_Man	3	0.00	
✎	TrackOP	Output value while in Track mode	1860	0.00	
✎	OPRateUp	Output rate up limit (%/sec)	37	Off (0)	
✎	OPRateDown	Output rate down limit (%/sec)	1861	Off (0)	
	OPRateDisable	Disable output rate limits	1862	No (0)	
✎	PowerFeedforward	Enable line voltage compensation	565	Off (0)	
	Ch2Deadband	Channel 2 deadband	16	Off (0)	
	NonLinearCooling	Channel 2 non-linear cooling algorithm	524	Off (0)	
	ManualStepValue	Manual step transfer output value	84	0.00	
	Ch1TravelTime	Channel 1 valve travel time (secs)	21	22.00	
	Ch2TravelTime	Channel 2 valve travel time (secs)	1863	22.00	
✎	RemoteOPHigh	Remote upper output limit	32	100.00	
✎	RemoteOPLow	Remote lower output limit	33	0.00	
✎	RemoteOPLimsDisable	Disable remote output limits	1864	No (0)	

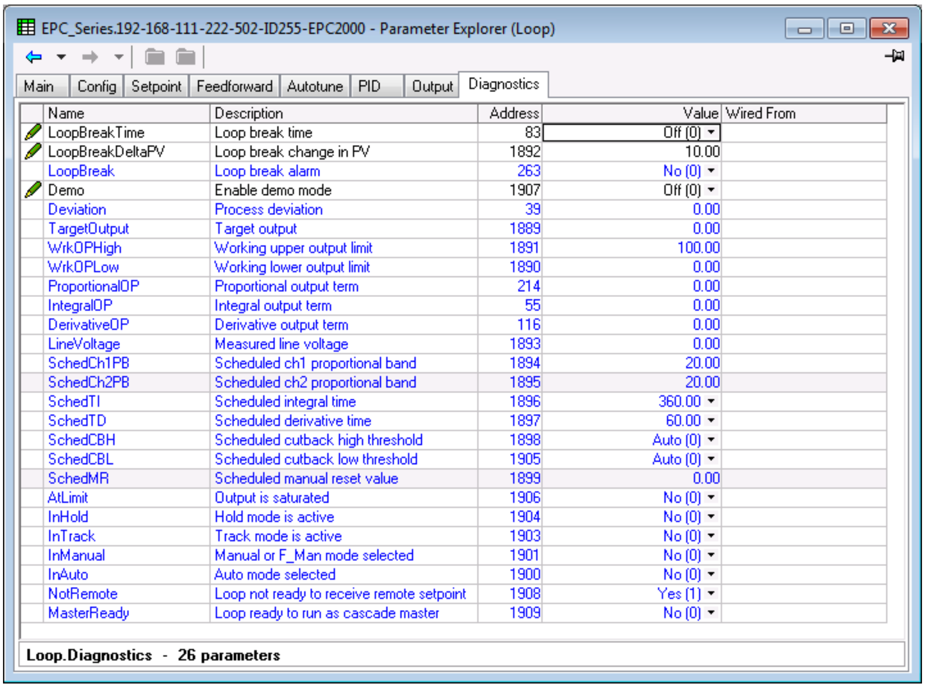
Loop.Output - 19 parameters

Parameter Name	Description	Available Values	Description
FallbackValue	Fallback output value	The Fallback Output Value is used in a number of circumstances: • If the PV status goes bad (e.g. sensor break), the loop will enter Forced Manual mode (F_Man) with either the fallback value or the last good output. This depends on the configured PV Bad Transfer type. • If Forced Manual (F_Man) mode is activated by an external signal (e.g. a process alarm) then the fallback output value is applied. • If the Recovery Mode is configured as 'ManualModeFallbackOP', then the controller will always start up in Manual mode with the fallback output value. This is also true when exiting Instrument Config or Standby modes.	
OutputHighLimit	Output upper limit	Maximum output power delivered by channels 1 and 2. By reducing the high power limit, it is possible to reduce the rate of change of the process, however, care should be taken as reducing the power limit will reduce the controllers ability to react to disturbances. Range between OutputLowLimit and 100.0%. This parameter does not affect the Fallback Value being achieved in Manual mode. **Default: 100**	
OutputLowLimit	Output lower limit	Minimum (or maximum negative) output power delivered by channels 1 and 2. Range between OutputHighLimit and −100.0% **Default: 0**	
Ch1Output	Channel 1 output	The current value of channel 1 output demand. Channel 1 (Heat) output. The Ch1 output is the positive power values (0 to Output Hi) used by the heat output. Typically this is wired to the control output (time proportioning or DC output). Range between OutputHighLimit and OutputLowLimit.	
Ch2Output	Channel 2 output	The current value of channel 2 output demand. The Ch2 output is negative portion of the control output (0 – Output Lo) for heat/cool applications. It is inverted to be a positive number so that it can be wired into one of the outputs (time proportioning or DC outputs). Range between OutputHighLimit and OutputLowLimit	
ManualOP	Output value when in Manual or F_Man	The output value when in manual or forced manual modes	
TrackOP	Output value when in Track mode	This value will be used as the output when in track mode	
OPRateUp	Output rate up limit (%/sec)	0	Increasing output rate limit %/second. Limits the rate at which the output from the PID can change. Output rate limit can be useful in helping to prevent rapid changes in output from damaging the process or the heater elements. However, it should be used with care since the higher setting could significantly affect the process performance. Range OFF or 0.1%/Sec to display range.
OPRateDown	Output rate down limit (%/sec)	0	Decreasing output rate limit %/second. Comments listed for OPRateUp apply.
OPRateDisable	Disable output rate limits	When an output rate limit has been configured, this input can be used as part of the strategy to temporarily disable rate limiting.	
		No (0)	Enable
		Yes (1)	Disable
PowerFeedforward	Enable line voltage compensation	No (0)	Power Feedforward is a feature which monitors the line voltage and adjusts the output signal to compensate for fluctuations before they affect the process temperature. It is assumed that the supply to the controller is the same as the supply to the load.
		Yes (1)	
Ch2Deadband	Channel 2 deadband	Ch1/Ch2 De22.0adband is a gap in percent between output 1 going off and output 2 coming on and vice versa. For on/off control this is taken as a percentage of the hysteresis.	

Parameter Name	Description	Available Values	Description
NonLInearCooling	Channel 2 non-linear cooling algorithm	Channel 2 non-linear cooling algorithm. Selects the type of cooling channel characterization to be used.	
		Off (0)	No non-linear cooling algorithm used. Ch2 output will be linear.
		Oil (1)	Often used in an extruder to provide cooling using oil.
		Water (2)	Often used in an extruder to provide flash cooling using water.
		Fan (3)	Often used in an extruder to provide on/off cooling using air or an analog output to a VFD driving fan.
ManualStepValue	Manual step transfer output value	If the Manual Transfer type has been configured as 'Step' then this value will be applied to the output on the transition from Auto to Manual.	
Ch1TravelTime	Channel 1 valve travel time (secs)	The valve travel time in seconds for the channel 1 output. This parameter must be set if the Ch1 Control Type is set to VP. The valve travel time is the time taken for the valve to move from fully-closed to fully-open. This must be the measured time to move from endstop to endstop. This not necessarily the time printed on the valve label. In a Heat/Cool application Channel 1 is the heat valve. **Default: 22.0**	
Ch2TravelTime	Channel 2 valve travel time (secs)	The valve travel time in seconds for the channel 2 output. This parameter must be set if the Ch2 Control Type is set to VP. The valve travel time is the time taken for the valve to move from fully-closed to fully-open. This must be the measured time to move from endstop to endstop. This not necessarily the time printed on the valve label. In a Heat/Cool application, Channel 2 is the cool valve. **Default: 22.0**	
RemoteOPHigh	Remote upper output limit	Can be used to limit the output of the loop from a remote source or calculation. **Default: 100.0**	
RemoteOPLow	Remote lower output limit	Can be used to limit the output of the loop from a remote source or calculation. **Default: 0.0**	
RemoteOPLimsDisable	Disable remote output limits	No (0)	
		Yes (1)	Disable remote output limits.

Loop.Diagnostics

The diagnostic function block contains parameters which may be used for troubleshooting or may be soft wired as part of a control strategy. The figure below shows the parameters and the table which follows details each parameter.


EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Loop)
Main
Config
Setpoint
Feedforward
Autotune
PID
Output
Diagnostics

Name	Description	Address	Value	Wired From
LoopBreakTime	Loop break time	83	Off (0)	
LoopBreakDeltaPV	Loop break change in PV	1892	10.00	
LoopBreak	Loop break alarm	263	No (0)	
Demo	Enable demo mode	1907	Off (0)	
Deviation	Process deviation	39	0.00	
TargetOutput	Target output	1889	0.00	
WrkOPHigh	Working upper output limit	1891	100.00	
WrkOPLow	Working lower output limit	1890	0.00	
ProportionalOP	Proportional output term	214	0.00	
IntegralOP	Integral output term	55	0.00	
DerivativeOP	Derivative output term	116	0.00	
LineVoltage	Measured line voltage	1893	0.00	
SchedCh1PB	Scheduled ch1 proportional band	1894	20.00	
SchedCh2PB	Scheduled ch2 proportional band	1895	20.00	
SchedTI	Scheduled integral time	1896	360.00	
SchedTD	Scheduled derivative time	1897	60.00	
SchedCBH	Scheduled cutback high threshold	1898	Auto (0)	
SchedCBL	Scheduled cutback low threshold	1905	Auto (0)	
SchedMR	Scheduled manual reset value	1899	0.00	
AtLimit	Output is saturated	1906	No (0)	
InHold	Hold mode is active	1904	No (0)	
InTrack	Track mode is active	1903	No (0)	
InManual	Manual or F_Man mode selected	1901	No (0)	
InAuto	Auto mode selected	1900	No (0)	
NotRemote	Loop not ready to receive remote setpoint	1908	Yes (1)	
MasterReady	Loop ready to run as cascade master	1909	No (0)	

Loop.Diagnostics - 26 parameters

Parameter Name	Description	Available Values	Value Description
LoopBreakTime	Loop break time	0	Sets the loop break time. This parameter, along with LoopBreakDeltaPV, sets the condition for loop break detection. The loop break alarm attempts to detect loss of control in the control loop by checking the control output, the process value and its rate of change. Loop break detection works for all control algorithms: PID, VP and ON-OFF. Note: This is not to be confused with load failure and partial load failure.
LoopBreakDeltaPV	Loop break change in PV	If the controller output is saturated, this is the minimum change in PV that the system would expect to see in 2× loop break times. If the output is saturated and the PV has not moved by this amount in 2×LoopBreakTime then the loop break alarm will be activated. **Default: 10.0**	
LoopBreak	Loop break alarm	No (0)	
		Yes (1)	This flag indicates the a loop break has been detected.
Demo	Enable demo mode	Off (0)	
		On (1)	Turns on the simulated plant for demonstration purposes.
Deviation	Process deviation	This is the process deviation (sometimes called “error”). It is calculated as PV minus SP. Therefore, a positive deviation implies that the PV is above Setpoint, while a negative deviation implies that the PV is below Setpoint.	
TargetOutput	Target output	The requested control output. This is the output taken before rate any limiting.	
WrkOPHigh	Working upper output limit	This is the resolved upper output limit that is currently in use. It is derived from the gain scheduled limit, the remote limits and the global limits	
WrkOPLow	Working lower output limit	This is the resolved lower output limit that is currently in use. It is derived from the gain scheduled limit, the remote limits and the global limits	

Parameter Name	Description	Available Values	Value Description
ProportionalOP	Proportional output term	This is the output contribution from the proportional term. This diagnostic is not available for VP.	
IntegralOP	Integral output term	This is the output contribution from the integral term. This diagnostic is not available for VP.	
DerivativeOP	Derivative output term	This is the output contribution from the derivative term. This diagnostic is not available for VP.	
LineVoltage	Measured line voltage	This is the line voltage measured by the instrument (in volts). This is the value used for Power Feedforward if enabled.	
SchedCh1PB	Scheduled ch1 proportional band	The currently active channel 1 proportional band.	
SchedCh2PB	Scheduled ch2 proportional band	The currently active channel 2 proportional band.	
SchedTI	Scheduled integral time	0	The currently active integral time.
SchedTD	Scheduled derivative term	0	The currently active derivative time
SchedCBH	Scheduled cutback high threshold	0	The currently active cutback high threshold.
SchedCBL	Scheduled cutback low threshold	0	The currently active cutback low threshold.
SchedMR	Scheduled manual reset value	0	The currently active manual reset value
AtLimit	Output is saturated	No (0)	
		Yes (1)	This flag will be asserted whenever the controller output is saturated (has hit a limit). This may be useful for a cascade strategy.
InHold	Hold mode is active	No (0)	
		Yes (1)	Hold mode is active.
InTrack	Track mode is active	No (0)	
		Yes (1)	Track mode is active.
InManual	Manual or F_Man mode selected	No (0)	
		Yes (1)	Manual or F-mode selected.
InAuto	Auto mode selected	No (0)	
		Yes (1)	Auto mode is selected.
NotRemote	Loop not ready to receive remote setpoint	No (0)	
		Yes (1)	When true (Yes), this flag indicates that the controller is not ready to receive a remote setpoint. Typically this is wired back to the Track output value of a cascade master, such that the master can track the slave SP if the slave is switched to local setpoint.
MasterReady	Loop ready to run as cascade master	No (0)	
		Yes (1)	When true (Yes), this flag indicates that the controller is not able to run as a cascade master. Typically this is wired to the RSP_En input of a cascade slave, such that the slave can control to a local setpoint if the master is taken out of Auto mode.

Alarms

What’s in this Chapter?

- This chapter provides a description of alarm types used in the controllers.
- Definitions of alarm parameters.

What are Alarms?

For the purposes of this section alarms alert an operator when a pre-set threshold, determined by the user as applicable to their particular process, has been exceeded.

Unless they have been produced in a particular application in EPC2000 Programmable Controller, there are no specific alarms. It is then necessary to wire alarm blocks using iTools (see "Graphical Wiring" on page 80.

Alarms may also switch an output, usually a relay, to allow external devices to be activated when an alarm is active (see "Example 2: To Connect an Alarm to a Physical Output" on page 81).

Up to six process related alarms can be configured in all models.

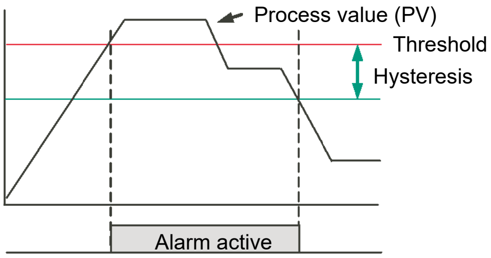
Alarms can also be configured as ‘Events’. Events may be used to operate an output.

Alarm Types

There are four distinct types of alarm; Absolute, Deviation, Rate of Change and Digital. These are split into the following nine alarm types. The descriptions for these nine alarm types are for the algorithms only, blocking and latching is applied separately, after the active/working state has been determined (See "Blocking" on page 186).

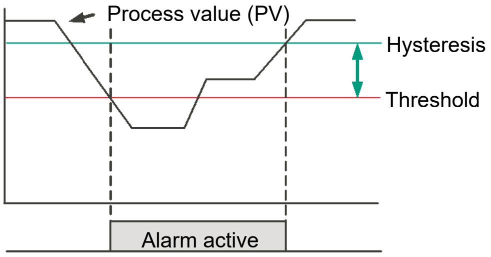
Absolute High

The Absolute High alarm is active when the input is greater than the threshold. It remains active until the input falls below the threshold minus the hysteresis value.



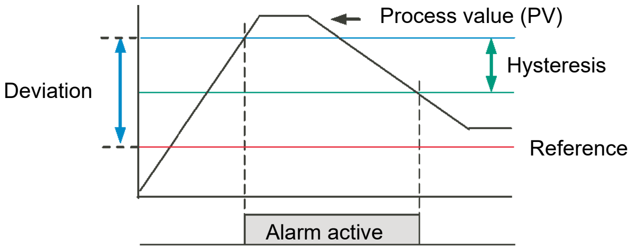
Absolute Low

The Absolute Low alarm is active when the input is less than the threshold. It remains active until the input increases above the threshold plus the hysteresis value.



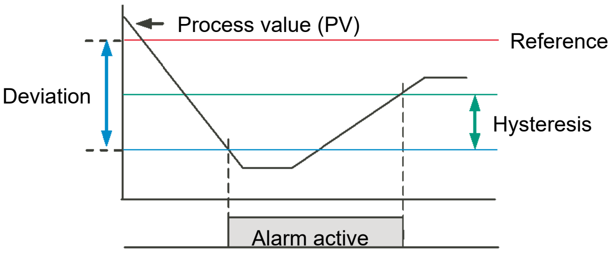
Deviation High

The alarm is triggered when the input becomes higher than the reference by the amount of the deviation. It remains active until the input falls below the hysteresis value.



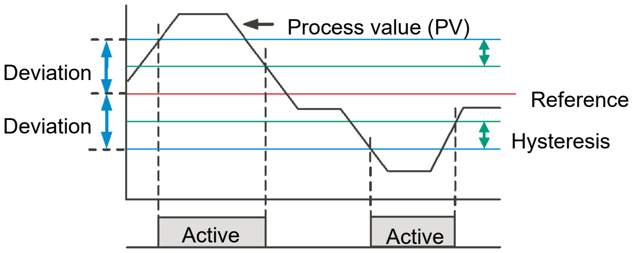
Deviation Low

The alarm is triggered when the input becomes lower than the reference by the amount of the deviation. It remains active until the input increases above the hysteresis value.



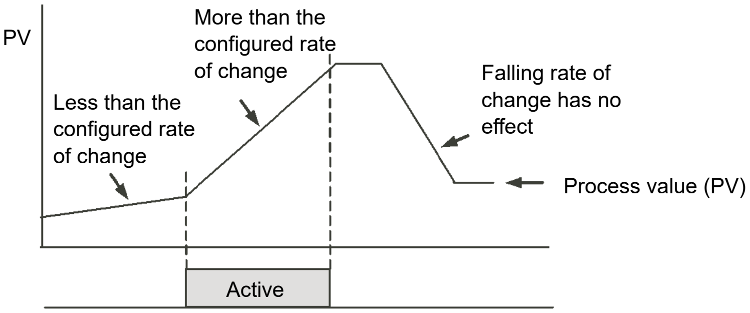
Deviation Band

The Deviation Band alarm is a combination of Deviation High and Deviation Low alarms. The alarm will be active when the input goes outside the deviation band i.e. is greater than the reference plus the deviation OR is less than the reference minus the deviation. It remains active until the input returns to within the reference value, plus/minus the deviation, minus/plus the hysteresis value.



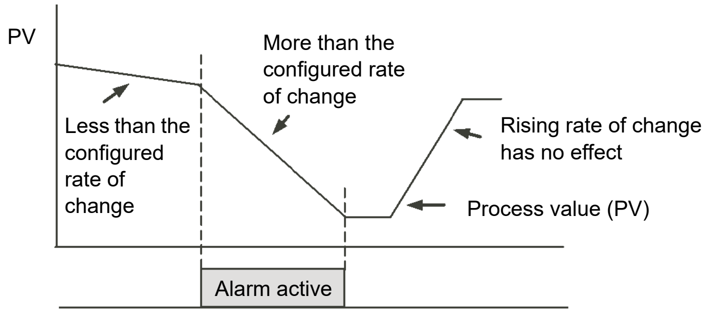
Rising Rate of Change

The Rising Rate of Change alarm sets the alarm active when the rate at which the input increases exceeds the configured maximum rate of change (per change time). It will remain active until the rising rate of the input falls below the configured rate of change.


PV
More than the configured rate of change
Less than the configured rate of change
Falling rate of change has no effect
Process value (PV)
Active

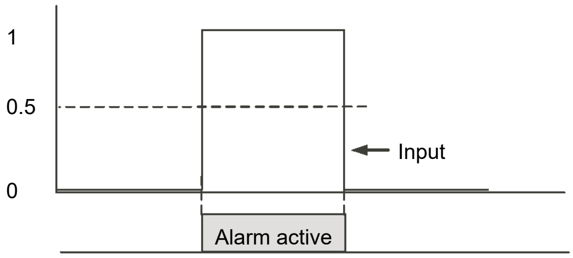
Falling Rate of Change

The Falling Rate of Change alarm sets the alarm active when the rate at which the input decreases exceeds the configured maximum rate of change (per change time). It will remain active until the falling rate of the input falls below the configured rate of change.


PV
More than the configured rate of change
Less than the configured rate of change
Rising rate of change has no effect
Process value (PV)
Alarm active

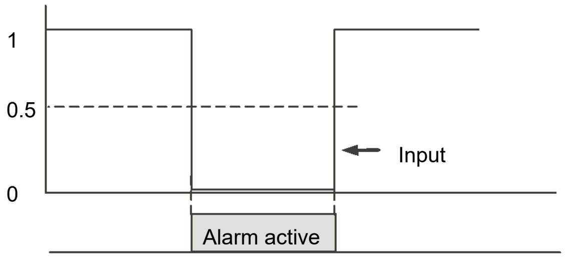
Digital High

The Digital High alarm is effectively an Absolute High alarm with a fixed threshold of 0.5 and 0 hysteresis. It sets the alarm to active when the input is greater than 0.5 (HIGH/TRUE for a digital/boolean input).



Digital Low

The Digital Low alarm is effectively an Absolute Low alarm with a fixed threshold of 0.5 and 0 hysteresis. It sets the alarm to active when the input is less than 0.5 (LOW/FALSE for a digital/boolean input).

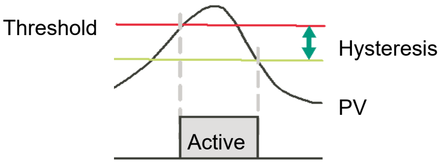


Sensor Break

If the process sensor becomes open circuit an alarm can be generated. The chosen application may already do this, but, if not it must be wired. This is shown in section "Example 3: To Wire Sensor Break" on page 82.

Hysteresis

Hysteresis helps to prevent an alarm output from oscillating (rapidly switching between active and non active) due to electrical noise (such as EMI) on the monitored parameter. As illustrated in the diagram below, the alarm will become active as soon as the alarm condition has been met (that is, the monitored parameter crosses the threshold value), however, it will only become inactive when the monitored parameter has gone into the region defined by the hysteresis amount.

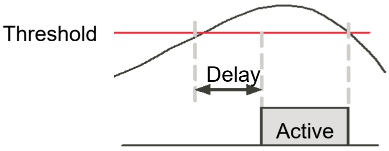


Hysteresis can be disabled by setting a value of 0.0, this is the default value.

Hysteresis is supported for the following analog alarm types: AbsHi, AbsLo, DevHi, DevLo, DevBand.

Delay

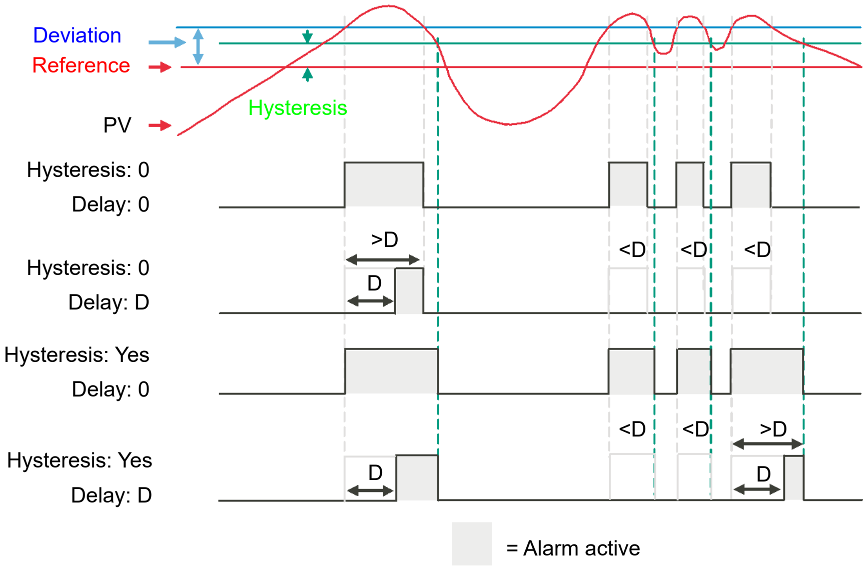
Alarm Delay is supported for all alarm types. This is a small delay between sensing the alarm condition and acting upon it. If in the time between the two, the measured value returns to below the threshold, then the alarm will not be activated and the delay timer is reset.



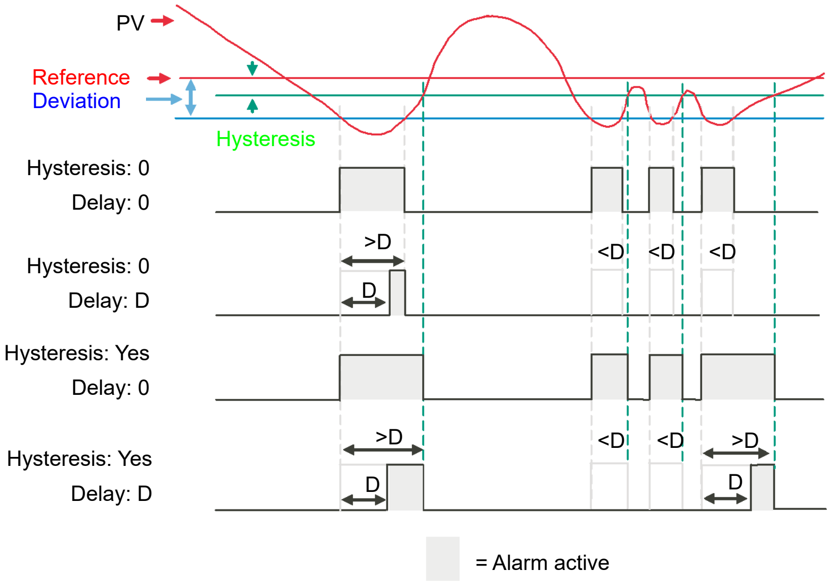
Effects of Delay and Hysteresis

The following diagrams show the effect of delay on hysteresis (for a very out of control process!).

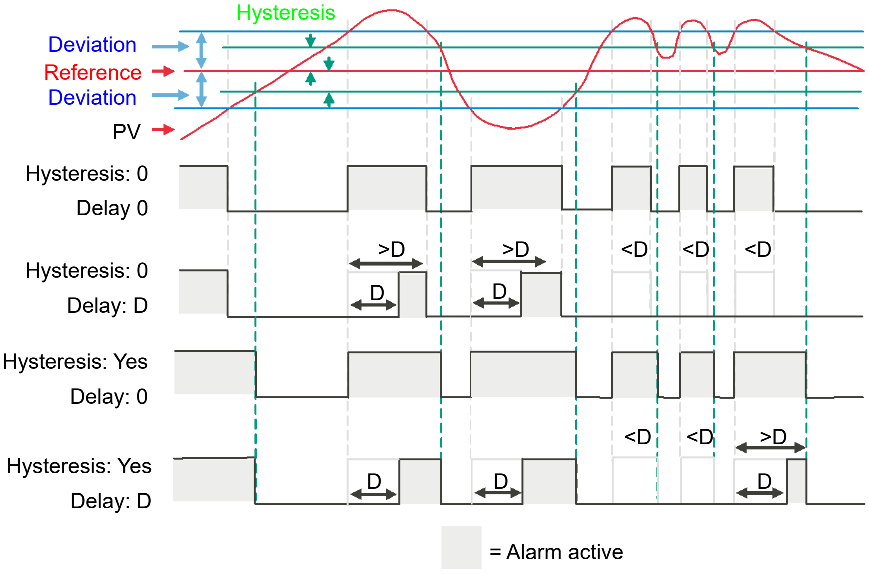
Deviation High



Deviation Low



Deviation Band



Inhibit

Inhibit helps to prevent an alarm from activating when the Alarm Inhibit input is held High. Alarm Inhibit is supported for all alarm types.

Standby Inhibit

Standby inhibit helps to prevent an alarm from activating when the instrument is in Standby "Standby" on page 62. This includes when the instrument is in configuration mode. Alarm Standby Inhibit is supported for all alarm types.

Latching

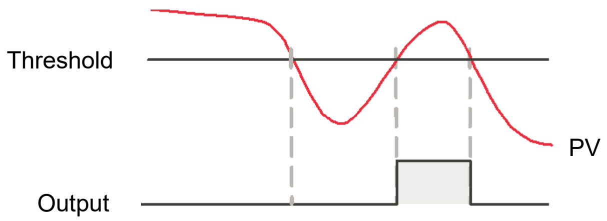
Alarm latching is used to hold the alarm condition active once an alarm has been detected.

The following latching types are supported for all alarm types:

Type	Description
None	No latching methodology i.e. when the alarm condition is removed the alarm will become inactive without being acknowledged.
Auto	The alarm will remain active until the alarm condition has been removed and the alarm has been acknowledged. The alarm can be acknowledged **at any time** after the alarm has become active.
Manual	The alarm will remain active until the alarm condition has been removed and the alarm has been acknowledged. The alarm can only be acknowledged **after** the alarm condition has been removed.
Event	Same as a non-latching alarm except the alarm is used as a trigger and therefore will not be displayed.

Blocking

Blocking stops an alarm from being activated until the value of the monitored parameter (for example PV) has first achieved the desired working state. It is typically used to ignore start-up conditions which are not representative of running conditions. Blocking of alarms is supported for all alarm types.



Blocking will be enforced after a power-cycle or after exit from configuration depending upon the latching status of the alarm as follows:

- For a non-latching alarm or an event alarm, blocking will be enforced.
- For an auto-latching alarm, blocking will be enforced only if the alarm had been acknowledged prior to the power cycle or exit from configuration level.
- For a manual-latching alarm, blocking will not be enforced.
- Blocking will be enforced for a Deviation alarm if the reference value is changed. It should be noted that if the reference value is wired from an electrically 'noisy' input then blocking should be disabled otherwise the alarm will continually be blocked.
- Blocking will be enforced, regardless of current active state and latching method, if the alarm is inhibited (either inhibit or standby inhibit).

To Set Alarm Threshold

The levels at which absolute high and absolute low process alarms operate are adjusted by the Threshold parameter, whilst in Configuration mode. Refer to "Alarm" on page 123 for details of the Alarm parameters.

Alarm Indication

Any output (usually a relay) attached to an alarm will operate. To attach an output to an alarm see "Example 2: To Connect an Alarm to a Physical Output" on page 81.

It is normal to configure the relay to be de-energized in alarm so that an alarm can be indicated externally if power to the controller is removed.

To Acknowledge an Alarm

There are a number of ways in which an alarm can be acknowledged. They include:

1. Using iTools in Configuration mode, select the correct Alarm function block and change the ‘Ack’ parameter to Yes. This acknowledges the alarm. The value of ‘Ack’ reverts to ‘No’ as soon as the alarm acknowledgement is confirmed by the controller.
2. A digital input can be wired using iTools to alarm acknowledge. The procedure is the same as described in section "Example 2: To Connect an Alarm to a Physical Output" on page 81.
3. Use the GlobalAck (global acknowledge) parameter in the Instrument.Dianostics function block to acknowledge all alarms. This may also be wired in the same way as other parameters (for example, to a digital input) and is used to acknowledge all alarms.

The action which takes place depends on the latching type of the alarm configured. By default alarms are configured as non-latching, de-energized during an alarm.

Alarms Advanced

Behavior of Alarms after a Power Cycle

The response of an alarm after a power cycle depends upon the latching type, whether it has been configured to be a blocking alarm, the state of the alarm and the acknowledge status of the alarm.

The response of active alarms after a power cycle is as follows:

- For a non-latching alarm, blocking, if configured, will be re-instated. If blocking is not configured the active alarm will remain ‘active’. If the alarm condition has returned to within the threshold value during the down time, then the alarm will return ‘inactive’.
- For an auto-latching alarm, blocking, if configured, will be re-instated, only if the alarm had been acknowledged prior to the power cycle. If blocking is not configured or the alarm had not been acknowledged the active alarm will remain ‘active’. If the alarm condition has returned to within the threshold value during the downtime, then the alarm will return ‘inactive’ if it had been acknowledged prior to the power cycle else it will return ‘inactive but not acknowledged’. If the alarm was ‘inactive but not acknowledged’ prior to the power cycle the alarm will return ‘inactive but not acknowledged’.
- For a manual-latching alarm, blocking will not be re-instated and the active alarm will remain ‘active’. If the alarm condition has returned to within the threshold value during the downtime, then the alarm will return ‘inactive but not acknowledged’. If the alarm was ‘inactive but not acknowledged’ prior to the power cycle the alarm will return ‘inactive but not acknowledged’.

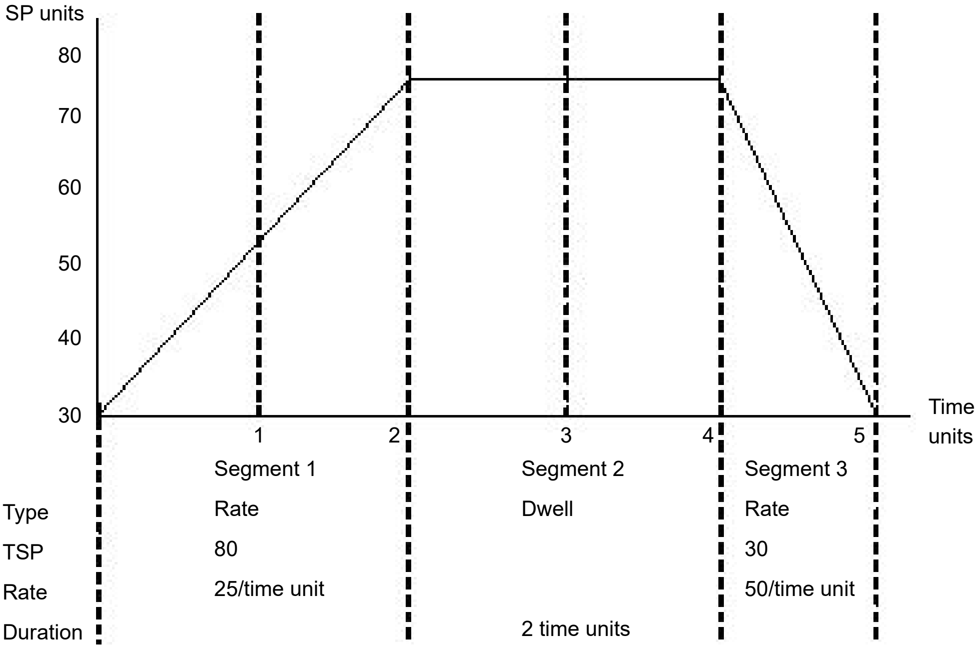
Programmer

What's in this Chapter?

This chapter describes the functionality of a setpoint programmer.

What is a Programmer?

A programmer provides a means of being able to vary the setpoint in a controlled manner over a set period of time. This varying setpoint can then be used in the control process.



The example above shows a simple three segment program in which the Target Setpoint (TSP) increases at a controlled rate of 25/time unit to a value of 75. It then dwells at that setpoint for two time units before decreasing to 30 at a controlled rate of 50/time unit.

The programmer in the EPC2000 Programmable Controller is a single channel programmer and can be ordered in four different options. These are:

- 1 x 8 Basic Programmer (1 program of 8 configurable segments, no event outputs).
- 1 x 24 Advanced Programmer (1 program of 24 configurable segments with up to 8 event outputs).
- 10 x 24 Advanced Programmer (10 programs of 24 configurable segments with up to 8 event outputs).
- 20 x 8 Advanced Programmer (20 programs of 8 configurable segments with up to 8 event outputs).

For all options, an additional End segment is provided which may also have event outputs if it is an Advanced Programmer.

The above programmer types are orderable options. They may be upgraded using the feature codes described in section "Instrument.Security" on page 95.

⚠ **CAUTION**

UNINTENDED EQUIPMENT OPERATION

If the programmer option is changed from 24-segment programs to 8-segment programs or vice versa, then previously stored programs will be lost. All segments will be defaulted to End type segments. It is recommended to clone the controller prior to upgrade so that a copy of the stored programs are retained before the Feature Security change is implemented.

Failure to follow these instructions can result in injury or equipment damage

Programs

A program is a sequence of varying setpoints which executes with reference to time. Up to a maximum of 20 programs are supported; the actual number of programs depends on the programmer type ordered, and is set via Feature Security (see "Instrument.Security" on page 95).

Programs are identifiable by a program number i.e. 1…20 and a configurable program name.

Segments

A segment is a single step within a program, typically it has a specified target setpoint and either a duration to maintain that setpoint, or a ramp rate (or time) to achieve that setpoint, however other segment types instruct the programmer to perform additional tasks.

Up to 24 configurable segments are supported, plus a fixed end segment, in each program. Each segment (in a program) is identifiable by a segment number (1 to 25) and can also be given an alphanumeric name.

The following types of segments are supported:

Ramp Time

A ramp time segment is specified by a target setpoint and a time in which to achieve the ramp to target setpoint.

Ramp Rate

A ramp rate segment specifies a target setpoint and a rate at which to ascend/descend to this setpoint.

Dwell

A dwell segment specifies how long the setpoint is to be maintained.

Step

A step segment causes the programmer setpoint to change to the target setpoint in a single execution cycle.

Note: The step will occur immediately followed by a 1 second dwell period to allow event outputs to be set.

Call

A call segment allows the main program to call another program as a subroutine. The number of times the program is called is configurable, 1 to 9999 or continuous.

Note: A program can only call other programs which have a program number greater than itself, this helps to prevent cyclic programs from being created.

This segment type is only available if multiple programs are enabled via Feature Security and the program is not the last program (i.e. Program 20). All configurable segments (1–24) can be configured as call segments.

⚠ **CAUTION**

CALL SEGMENTS

If a call segment is selected the controller will default to calling the next program number. This may not necessarily be the correct program so you must ensure that the correct call program number is selected manually.

Failure to follow these instructions can result in injury or equipment damage

End

An End segment is the very last segment in a program and using the Program.ProgramEndType parameter, the user can specify the programmer behavior when the program ends, as follows:

- Dwell—the programmer setpoint (PSP) is maintained indefinitely and event outputs remain at the states configured for the End segment.
- Reset—the program is reset and the programmer setpoint (PSP) will servo either to PVInput value or SPInput value as configured by the Programmer.Setup.ServoTo parameter. The event outputs will return to the states specified by the Programmer.Setup.ResetEventOP parameter.
- Track - the programmer setpoint (PSP) is maintained indefinitely and event outputs remain at the states configured for the End segment. If the programmer is wired to the Loop, the Loop will be forced into Track mode.

Note: The first End segment will terminate the program in the configured manner if there no more program cycles left to run.

Standard Functionality

The EPC2000 Programmable Controller supports the following standard functionality:

Recovery Strategy

The recovery strategy after an instrument reset or a power outage can be configured to be:

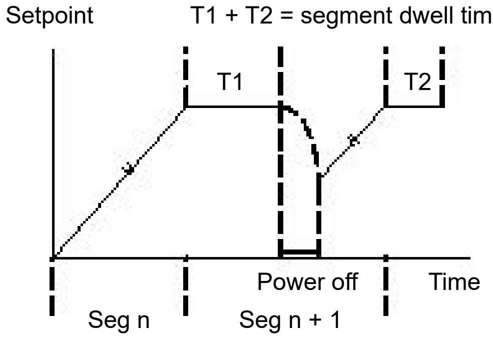
- Ramp Back—the programmer setpoint will servo to the input Process Value (PV), and ramp to the target setpoint at the rate prior to the power outage.
- Reset—the programmer will reset the program.
- Continue—the programmer setpoint will return immediately to its last value prior to the reset and the program will continue to run.

This is shown diagrammatically in the following sections.

Ramp back (Power outage during Dwell segments)

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

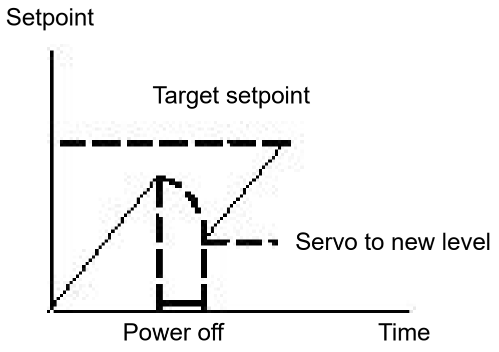
On achieving the Dwell setpoint, the dwell time will continue from the point at which the power was interrupted.



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

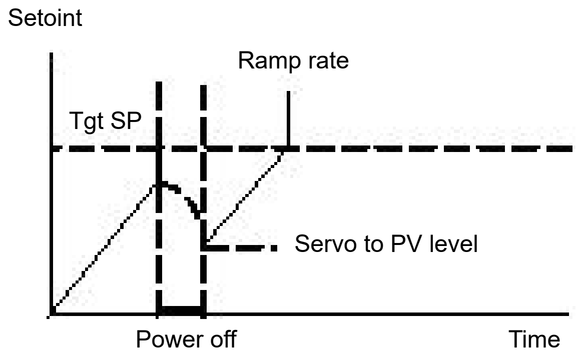
Ramp back (Power outage during Ramp segments)

If the interrupted segment was a Ramp, then the programmer will servo the programmer setpoint to the PV, then ramp towards the target setpoint using the ramp rate prior to the power outage.



Ramp back (Power outage during Ramp Time segments)

If the programmer was interrupted while a Ramp Time segment is running then, when the power is returned, the previous ramp rate will be recovered. The time remaining will be recalculated. The rule is to maintain RAMP RATE, but alter TIME REMAINING.



Sensor Break Recovery

If the recovery strategy is set to Reset, then, upon sensor break of the PV input, the program will be reset. If the recovery strategy is other than Reset, then the program will be placed in Hold. When the PV input goes out of sensor break, the programmer will apply the recovery strategy described above.

Holdback

When the PV deviates from the Programmer Setpoint (PSP) by more than a specified amount the program will temporally hold until the PV catches up to within the specified deviation.

Holdback Style configures Holdback to operate over the entire program or on a per segment basis (mutually exclusive).

Holdback Type may be set to Off, Low, High or Band.

- Off: Holdback disabled.

- Low: Active if PV < (PSP − Holdback Value)
- High: Active if PV > (PSP + Holdback Value)
- Band: Active if (PV < (PSP − Holdback Value)) OR (PV > (PSP + Holdback Value))

Servo To PV/SP

The programmer can be set to servo to (jump) the PV input or the Setpoint input at the start of a program.

Event Outputs

Up to eight digital 'event' outputs can be configured for each segment within a program. These event outputs will remain at their configured value for the entire duration of the segment.

Digital Inputs

The following digital inputs are supported:

- Run - start the current program on the rising edge of this input.
- Hold - hold the current program whilst this input is High.
- Reset - current program is in reset whilst this input signal is High.
- Run/Hold - This is a dual action input. A rising edge will run the current program, but hold the current running program whilst the input is Low.
- Run/Reset - This is a dual action input. A rising edge starts the current program, but resets the current program whilst input is Low.
- Advance - a rising edge initiates the following sequence of actions:
  - go to the end of the current segment.
  - set the programmer setpoint to the target setpoint.
  - start the next segment.

Program Cycles

A program can be configured to repeat 1 to 9999 times or run continuously.

Configuration Mode Reset

It is not permissible to run a program whilst the instrument is in configuration mode. If a program is running and the instrument is placed into configuration mode (by comms) the running program will be reset.

Program Selection

When multiple programs are configured, selecting the program to be run is done by setting the Programmer.ProgramNumber parameter to the required program number. This selection can be performed via Comms.

It is often convenient to use a BCD switch physically connected to digital inputs as shown in the "Digital Communications Connections" on page 45.

The program selected can then be run using Mode parameter or any of the Run digital input parameters, i.e., Run, RunHold or RunReset digital input.

Rules for Program Creation / Editing

It is possible to create and edit a stored program, i.e., Programs 1-20, (via Comms) even when the programmer is in RUN, HOLD or RESET mode, with the changes being retained.

When running one of the stored programs it will first be copied into the ‘working’ program which will then be run. It is NOT be possible to edit the working program when the programmer is in RESET, but it can be edited when the programmer is in RUN or HOLD, however, the changes will be overwritten when a different program is loaded to be run. Edits to the working program will not change the stored programs. The working program will be overwritten when the next stored program is copied into it, either as a result of running a new program or calling another program as a subroutine.

A Programmer Run list is provided (via Comms) which can edit a copy of the working program segment that is currently running when the programmer is in HOLD mode, however the changes will be overwritten when the next segment is loaded and run.

Program & Segment Times

Segment Time Left is available whilst a program is running.

The programmer will attempt to calculate Program Time left when the program is running or when the working program is edited while the program is on Hold. If the calculation takes too long, it will be aborted and the Program Time Left parameter will not be available.

Resolution

When read/written via scaled integer comms, the units of the following segment parameters can be configured as follows:

- Segment.Duration (sec/min/hour) configured by Program.DwellUnitsTime (sec/min/hour).
- Segment.TimeToTarget (sec/min/hour) configured by Program.RampUnits.
- Segment.RampRate (per sec/per min/per hour) configured by Program.RampUnits.

In addition, when read/written via scaled integer comms, it is possible to configure the units for the following time remaining parameters:

- Programmer.Run.ProgramTimeLeft (sec/min/hour) configured by Programmer.Setup.Resolution.
- Programmer.Run.SegmentTimeLeft (sec/min/hour) configured by Programmer.Setup.Resolution (sec/min/hour).

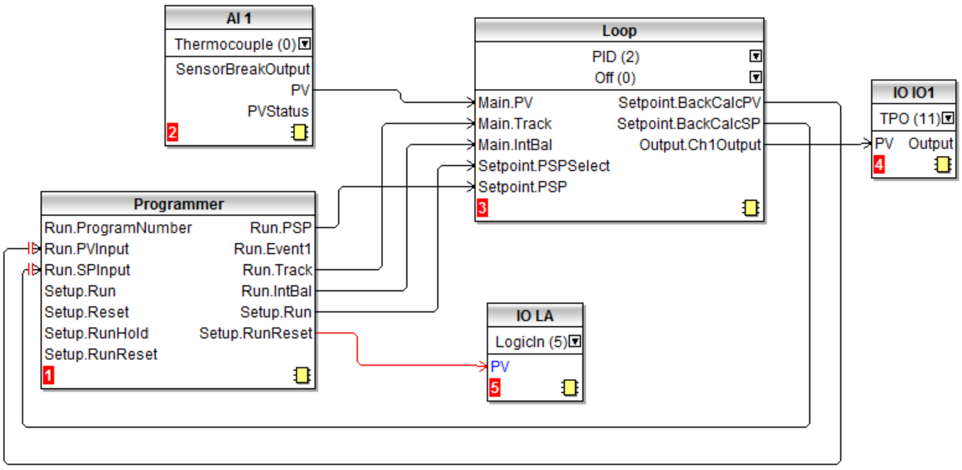
Times are stored as 32-bit integer millisecond values and as such, times will be capped at 500hrs i.e. 1,800,000,000ms. When a program exceeds this value, program time left will remain at 500hrs until such time that the accumulative segment times is 500hrs or less and then the program time left will start to count down.

Programmer Time Base Accuracy

The programmer timebase accuracy is dependent upon the timebase accuracy of the microcontroller which is specified as <±50ppm at 25°C (77°F). This equates to a worst case of ±4.3s in 24hrs.

Typical Loop to Programmer Graphical Wiring

The figure below shows a simple soft wiring diagram for a programmer.



Soft wiring is carried out using iTools and is described in section "Graphical Wiring" on page 80.

In the figure a thermocouple is connected to Analog Input AI1. The PV output from AI1 provides the input to the control loop. The setpoint for the control loop is provided by the programmer block using the parameter Run.PSP. The programmer will run when the Setup.Run parameter changes to true. In this example the LA digital input may be used to Run/Reset the programmer from an external source.

Integral balance is required so that there is no sudden change of output when the programmer is operated.

The loop heat output is connected to output IO1.

Communications

Programs can be configured and run via Modbus communications.

The Modbus parameter addresses for the programmer parameters, program parameters, and the segment parameters (for the first 16 segments) are compatible with the 2400 series controllers. Several parameters within segments are mutually exclusive and are accessed via comms using the same Modbus Address.

Modbus Address Ranges

1x8, 1x24 and 10x24 programmers are 2400 compatible.

2400 Compatible - Program General Data & Segments 1…16 Parameters

Area	Base Address - Decimal	Base address - HEX
Program 0 (Currently running program)	8192	2000
Program 1	8328	2088
Program 2	8464	2110
Program 3	8600	2198
Program 4	8736	2220
Program 5	8872	22A8
Program 6	9008	2330
Program 7	9144	23B8
Program 8	9280	2440
Program 9	9416	24C8
Program 10	9552	2550
Non-compatible - Segments 17…26 & Additional Programmer Parameters		
Area	Base Address - Decimal	Base address - HEX
Program 0	9688	25D8
Program 1	9768	2628
Program 2	9848	2678
Program 3	9928	26C8
Program 4	10008	2718
Program 5	10088	2768
Program 6	10168	27B8
Program 7	10248	2808
Program 8	10328	2858
Program 9	10408	28A8
Program 10	10488	28F8
Programmer (additional parameters)	10568 - 11007	2948 - 2AFF

Note: In 20x8 programmer the number of segments is fixed as is the assignment of Modbus Addresses. The mapping from segment instance to program/segment is different from all existing EPC2000 Programmable Controller programmer types. Modbus addresses do not match 2400 series.

Controlling the Programmer through iTools

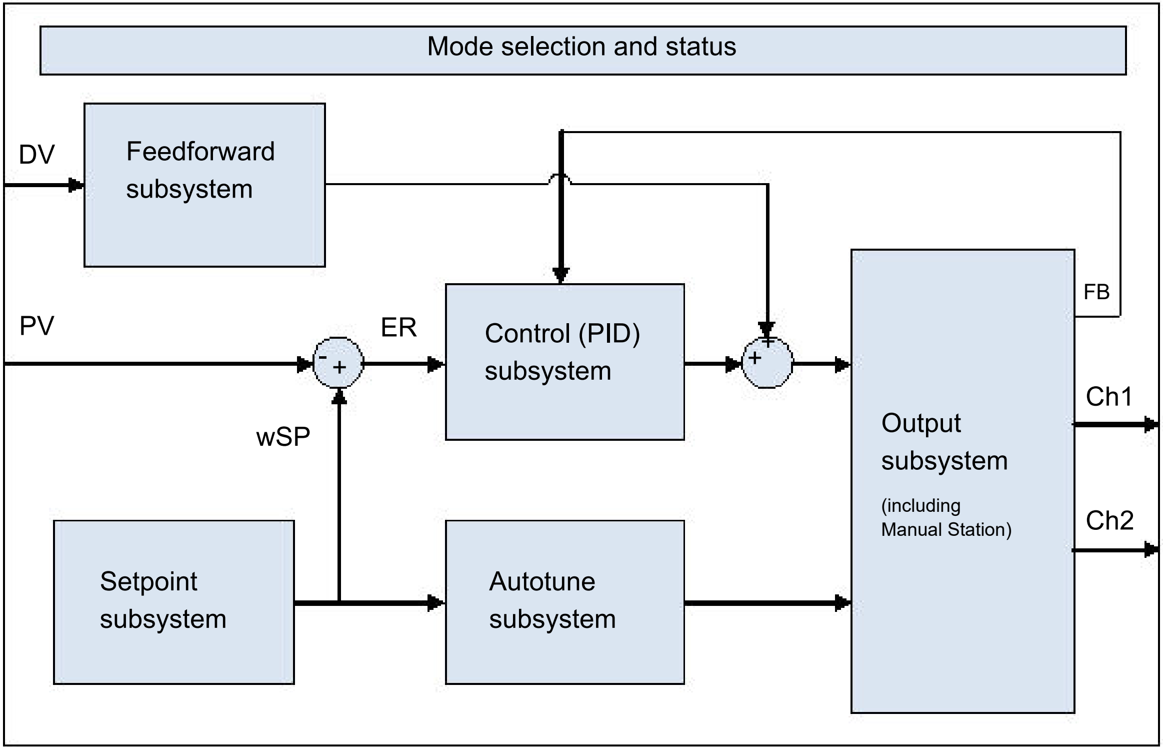
To run, reset and hold a program using iTools, refer to "To Run, Reset and Hold a Program" on page 78. For further details of configuring the Programmer using iTools, refer to "Programmer" on page 73.

Control

The ‘Loop’ function block contains and coordinates the various control and output algorithms. The diagram below shows the top-level structure of the Loop function block for a heat only or heat/cool temperature controller.

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

Loop Function Block



Types of Control

Three types of control loop may be configured. These are PID control, On/Off control, or control of motorized valves

PID Control

PID, also referred to as 'Three Term Control', is an algorithm which continuously adjusts the output, according to a set of rules, to compensate for changes in the process variable. It provides more stable control but the parameters need to be set up to match the characteristics of the process under control.

The three terms are:

- Proportional term (PB).
- Integral term (TI).
- Derivative term (TD).

The Eurotherm PID algorithm is based upon an ISA type algorithm in its positional (non-incremental) form. The output from the controller is the sum of the contributions from these three terms.The simplified Laplace transform is:

$$OP/ER = (100/PB)\ (1 + 1/sTI + sTD)$$

The combined output is a function of the magnitude and duration of the error signal, and the rate of change of the process value.

It is possible to turn off integral and derivative terms and control on proportional only (P), proportional plus integral (PI) or proportional plus derivative (PD).

An example of where PI control might be used i.e. D is turned off, is process plants (flows, pressures, liquid levels), which are inherently turbulent and noisy, causing valves to fluctuate wildly.

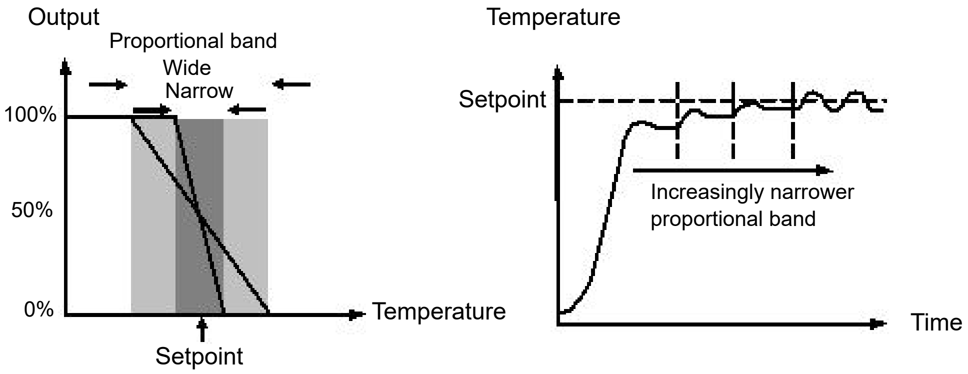
PD control may be used, for example, on servo mechanisms.

In addition to the three terms described above, there are other parameters which determine how well the control loop performs. These include High and Low Cutback and Manual Reset and are described in detail in subsequent sections.

Proportional Term 'PB'

The proportional term, or gain, delivers an output which is proportional to the size of the difference between SP and PV. It is the range over which the output power is continuously adjustable in a linear fashion from 0% to 100% (for a heat only controller). Below the proportional band the output is full on (100%), above the proportional band the output is full off (0%) as shown in the diagram below.

The width of the proportional band determines the magnitude of the response to the error. If it too narrow (high gain) the system oscillates by being over responsive. If it is too wide (low gain) the control is sluggish. The ideal situation is when the proportional band is as narrow as possible without causing oscillation.



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

The proportional band can be specified in engineering units or in percent of span (RangeHigh – RangeLow). Engineering Units is recommended for its ease of use.

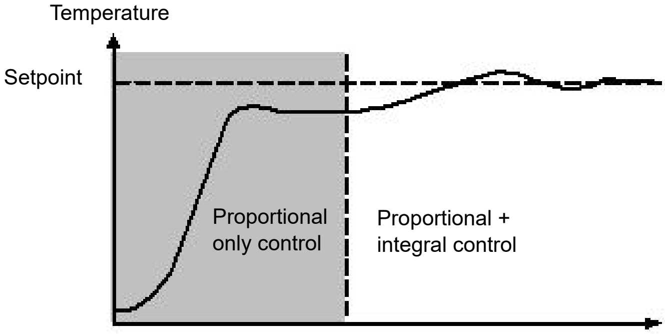
Previous controllers had the parameter Relative Cool Gain (R2G) to adjust the cool proportional band relative to the heat. This has been replaced by separate proportional bands for Channel 1 (Heat) and Channel 2 (Cool).

Integral Term 'TI'

In a proportional only controller, a difference between setpoint and PV must exist for the controller to deliver power. Integral is used to reduce this to a zero steady state control.

The integral term slowly shifts the output level as a result of a difference between setpoint and measured value. If the measured value is below setpoint the integral action gradually increases the output in an attempt to correct the difference. If it is above setpoint integral action gradually decreases the output or increases the cooling power to correct the difference.

The diagram below shows the result of introducing integral action.



The units for integral are measured in time. The longer the integral time constant, the more slowly the output is shifted and results in a sluggish response. Too small an integral time will cause the process to overshoot and even oscillate. The integral action may be disabled by setting its value to Off(0), in which case manual reset will be made available.

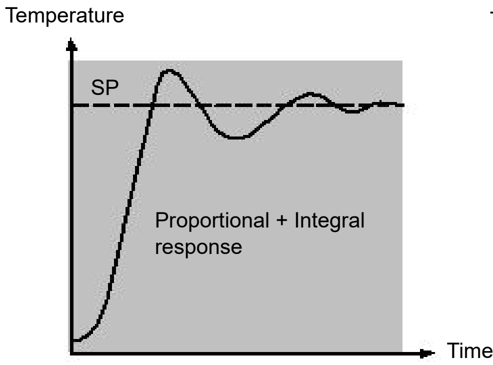
The integral time is specified in seconds. In US nomenclature, the integral time is equivalent to ‘seconds per repeat’.

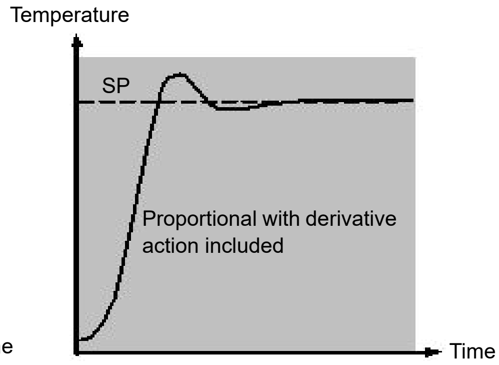
Integral Hold

When the IntegralHold parameter is turned on, the output value contained in the integrator will be frozen. It will be maintained even through mode changes. This can sometimes be useful e.g. in a cascade to stop the master integral winding up when the slave is saturated.

Derivative Term 'TD'

Derivative action, or rate, provides a sudden shift in output as a result of a rapid change in error. If the measured value falls quickly derivative provides a large change in output in an attempt to correct the perturbation before it goes too far. It is most beneficial in recovering from small perturbations.





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

Derivative is often mistakenly associated with overshoot inhibition rather than transient response. In fact, derivative should not be used to curb overshoot on start up since this will inevitably affect the steady state performance of the system. Overshoot inhibition is best left to the approach control parameters, High and Low Cutback, described below.

Derivative is generally used to increase the stability of the loop, however, there are situations where derivative may be the cause of instability. For example, if the PV is electrically noisy, then derivative can amplify that noise and cause excessive output changes, in these situations it is often better to disable the derivative and re-tune the loop.

Derivative time is specified in seconds. Derivative action can be turned off by setting the derivative time to Off(0).

Derivative on PV or Error (SP - PV)

By default, derivative action is applied to the PV only and not to the error (SP - PV). This helps to prevent large derivative kicks when the setpoint is changed.

If required, derivative can be switched to error using the DerivativeType parameter. This is not usually recommended but can, for example, reduce overshoot at the end of SP ramps.

Manual Reset (PD Control)

In a full three-term controller (that is, a PID controller), the integral term automatically removes the steady state error from the setpoint. Turn off the integral term to set the controller to PD. Under these conditions the measured value may not settle precisely at setpoint. The Manual Reset parameter (MR) represents the value of the power output that will be delivered when the error is zero.

This value must be set manually in order to remove the steady state error.

Cutback

Cutback is a system of approach control for process start-up and for large setpoint changes. It allows the response to be tuned independently of the PID controller, thereby allowing optimum performance for both large and small setpoint changes and disturbances. It is available for all control types except OnOff.

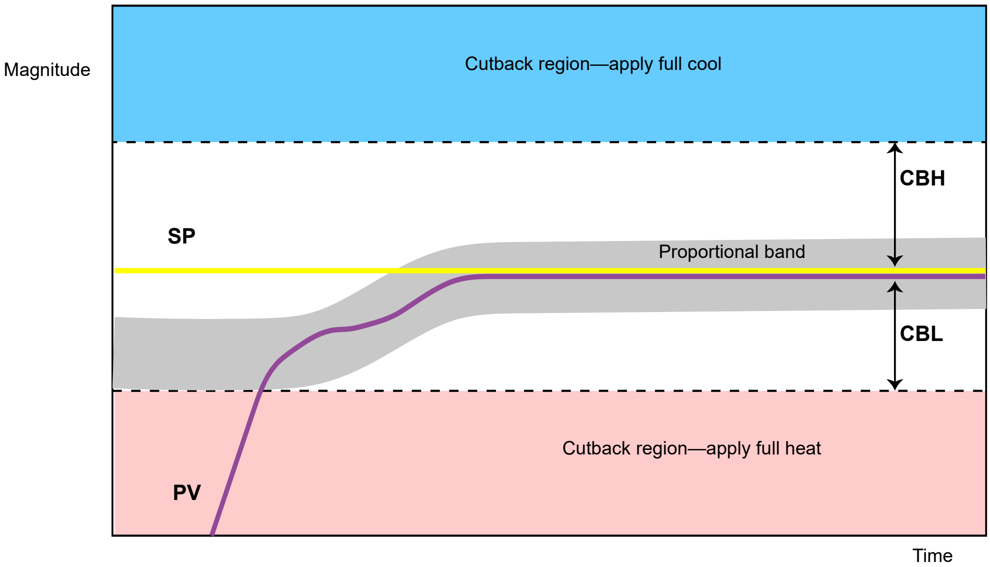
The cutback high and low thresholds, CBH and CBL, define two regions above and below the working setpoint (WSP). They are specified in the same units as the proportional band. Operation can be explained in three rules:

1. When the PV is more than *CBL* units *below* WSP, *maximum* output is applied.
2. When the PV is more than *CBH* units *above* WSP, *minimum* output is applied.
3. When the PV exits a cutback region, the output is returned *bumplessly* to the PID algorithm.

The effect of rule 1 and 2 is to bring the PV towards the WSP as rapidly as possible whenever there exists a significant deviation, just as an experienced operator might do manually.

The effect of 3 is to allow the PID algorithm to immediately start 'cutting back' the power from maximum or minimum when the PV passes the cutback threshold. Remember that, due to 1 and 2, the PV should be moving rapidly towards WSP, and it is this that causes the PID algorithm to start cutting back the output.

By default, CBH and CBL are set to *Auto (0)*, which means that they are automatically taken to be three times the proportional band. This is a reasonable starting point for most processes, but rise time to setpoint on start-up or large setpoint changes may be improved by tuning them manually.


Magnitude
Cutback region—apply full cool
CBH
SP
Proportional band
CBL
Cutback region—apply full heat
PV
Time

Note: Because cutback is a type of non-linear controller, a set of CBH and CBL values that are tuned for one particular operating point may not be satisfactory for another operating point. It is advisable therefore not to try to tune the cutback values tightly, or otherwise to use gain scheduling to schedule different values of CBH and CBL at different operating points. All the PID tuning parameters can be gain scheduled.

Reverse/Direct Action

For single-channel loops, the concept of reverse and direct action is important.

The ControlAction parameter should be set appropriately:

1. If an increase in control output causes a corresponding increase in PV, such as in a heating process, then set ControlAction to Reverse.
2. If an increase in control output causes a corresponding decrease in PV, such as in a refrigeration process, then set ControlAction to Direct.

The ControlAction parameter is not available for split-range configurations, where channel 1 is reverse acting and channel 2 is direct acting.

Loop Break

The loop is considered to be broken if the PV does not respond to a change in the output. An alarm may be initiated but in the EPC2000 Programmable Controller this must be explicitly wired using the ‘LoopBreak’ parameter. Since the time of response will vary from process to process the Loop Break Time parameter allows a time to be set before a Loop Break Alarm is initiated. In these circumstances the output power will drive to high or low limit. For a PID controller, two parameters under diagnostics are used to determine if the loop is broken, ‘Loop Break Time’ and ‘Loop Break Delta PV’.

If the control loop is broken, the output will tend to wind up and eventually hit a limit.

Once the output is at the limit, the loop break detection algorithm will monitor the PV. If the PV has not moved by a specified amount (LoopBreakDeltaPV) in twice the specified time (LoopBreakTime), then a loop break will be flagged.

Motorized Valve Positioning Control

Valve Positioning control is used for ‘three-step’ motorized valve actuators that are driven with a digital ‘raise’ and ‘lower’ signal. A common example is a valve modulating the firing rate of a gas-fired furnace or oven. Some valves are already fitted with positioners, in which case these algorithms are not suitable and PID should be used.

The EPC2000 Programmable Controller contains the Boundless (VPU) (or Unbounded) algorithm which does not require a feedback potentiometer.

This type of valve has an inherent travel time – that is the time needed to slew from end-stop to end-stop. This time should be measured as accurately as possible in both directions and the average entered into the appropriate travel time parameter.

Boundless (VPU)

The Valve Positioner boundless (VPU) algorithm operates *without knowledge* of the actual valve position. Therefore, it *does not* require a potentiometer on the valve.

VPU contains a special incremental form of the PID algorithm. It uses the valve itself as an accumulator, to ‘add up’ the increments calculated by the algorithm. Because of this special formulation, it can be treated as a positional algorithm, just like PID itself.

It contains a simple software model of the valve, based on the entered Travel Time, which estimates the valve position (the Working Output). It is important to realize that this estimation is just that, and that over time, particularly long cycles, the displayed Working Output and the real valve position may be different. This has no effect on control performance—it is purely a display problem. This model is also used in non-auto modes such as Manual.

With VPU, it is important that the valve travel time is measured and set as accurately as possible. This helps to ensure that the tuning parameters retain their real physical meanings and also helps to ensure proper autotuning, which may otherwise give an unsatisfactory tune. Motor travel time is defined as valve fully open - valve fully closed - it is not necessarily the time printed on the motor since, if mechanical stops have been set on the motor, the travel time of the actual valve may be different.

Motorized Valve Control in Manual mode

When manual is selected, the algorithm predicts where the valve will move to based on the value of the manual power. The manual output is set as normal, and the controller will position the valve according to the internal estimated position.

Every time the valve is driven to its end stops the estimated and real positions will tend to realign themselves.

Note: Parameters shown in this section are relevant to the subject being described. Further information is available in the Configuration chapter.

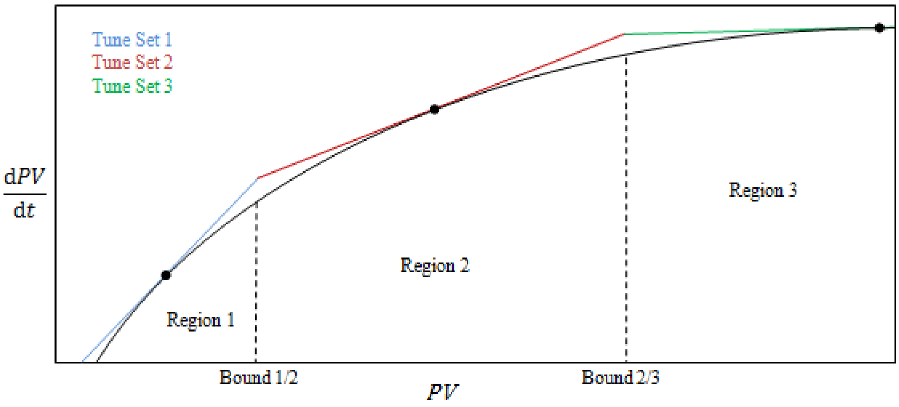
Gain scheduling

Some processes exhibit non-linear dynamics. For example, a heat treatment furnace may behave quite differently at low temperatures than at high temperatures. This is commonly due to the effects of radiant heat transfer, which start to appear above about 700℃ (1292°F). This is illustrated in the diagram below.

It is often unfeasible, then, for a single set of PID tuning constants to perform well over the entire process operating range. To combat this, several sets of tuning constants can be used and 'scheduled' according to the process operating point.

Each set of constants is called a 'gain set' or a 'tune set'. The gain scheduler selects the active gain set by comparing the value of the Scheduling Variable (SV) against a set of boundaries.

An integral balance is issued whenever the active gain set changes. This helps to prevent discontinuities ('bumps') in the controller output.



On-Off Control

Each of the two control channels can be configured for On-Off control. This is a simple type of control often found in basic thermostats.

The control algorithm takes the form of a simple hysteretic relay.

For channel 1 (heat):

1. When PV > WSP, OP = 0%
2. When *PV* < (*WSP* – *Ch1OnOffHyst*), OP = 100%

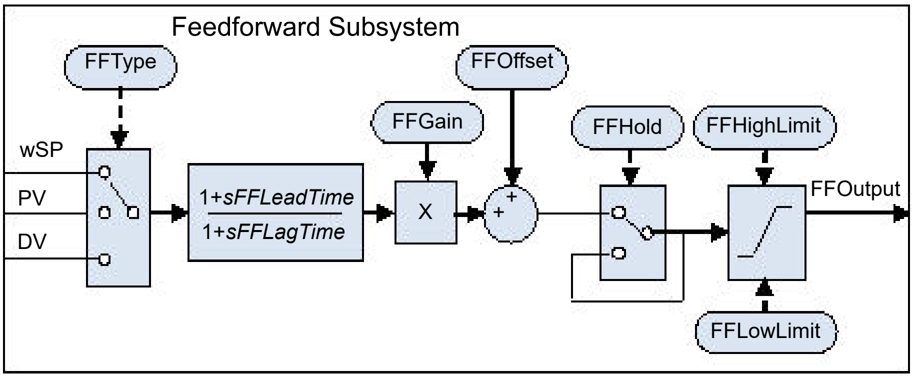
For channel 2 (cool):

1. When *PV* > (*WSP* + *Ch2OnOffHyst*), OP = 100%
2. When *PV* < *WSP*, OP = 0%

This form of control will lead to oscillation about setpoint but it is by far the easier to tune. The hysteresis should be set according to the trade-off between oscillation amplitude and actuator switching frequency. The two hysteresis values can be gain scheduled.

Feedforward

The block diagram for the feedforward subsystem structure is shown below.



The Loop incorporates a feedforward controller in addition to the normal feedback (PID) controller; it is capable of static or dynamic feedforward compensation. Broadly, there are three common uses for feedforward in these instruments and are described in turn below.

Disturbance feedforward

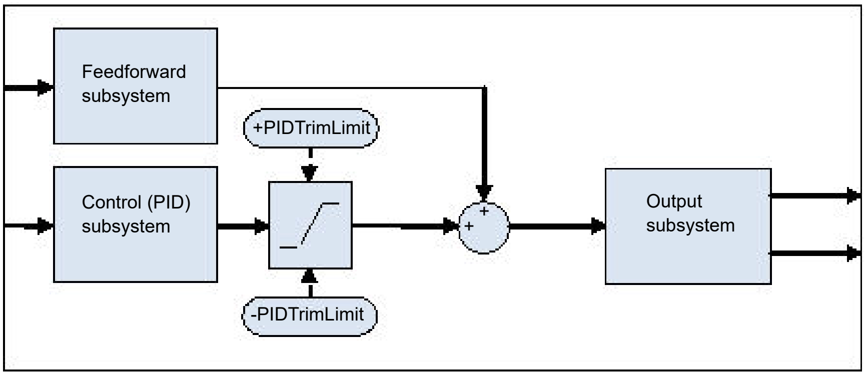
A disadvantage of a feedback (PID) controller is that it responds only to deviations between PV and SP. By the time a PID controller first starts to react to a process disturbance, it is already too late and the disturbance is in progress; all that can be done is to try to minimize the extent of the disruption as much as possible.

Feedforward control is often used to overcome this disadvantage. It uses a measurement of the disturbance variable itself and *a priori* knowledge of the process to predict the controller output that will exactly counter the disturbance *before* it has a chance to affect the PV.

Feedforward on its own also has a major disadvantage. It is an open-loop strategy that relies entirely on a model of the process. Modeling error, uncertainty and process variation all help to prevent zero tracking error being achieved in practice. Further, the feedforward controller can only respond to disturbances that are explicitly measured and modeled.

To counter the relative disadvantages, the Loop combines both types of control in an arrangement known as “Feedforward with Feedback Trim”. The Feedforward controller gives the principal control output and the Feedback controller can trim this output appropriately to give zero tracking error.

The diagram below shows the feedforward with feedback trim structure.



A symmetrical trim limit is provided around the PID component such that the influence of the feedback trim can be restricted.

Setpoint feedforward

Setpoint feedforward is arguably the type most often seen in instrument applications. A signal that is proportional to the working setpoint is fed-forward directly to the controller output. The most common scenario is dead-time dominant processes.

Dead times are common in process control. Flow lines, packaging lines, food processing lines and similar can all involve some amount of transport delay; that is to say that there is a finite period of time between an actuation being made by the final control element and a change being observed by the sensor.

Where this time delay is large with respect to other process dynamics, stable feedback control becomes increasingly difficult. A solution to this is often to detune the gain of the controller. Whilst this may well achieve stability, it also results in a sluggish system response to setpoint changes.

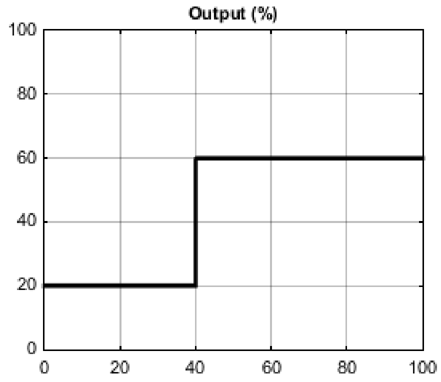
The ‘Feedforward with Feedback Trim’ arrangement shown above can be used to significantly improve the situation. The feedforward controller immediately gives an output value close to the final value and the PID controller can then trim this to give zero tracking error. The maximum amount of trim can be limited to help prevent the PID component having too much influence.

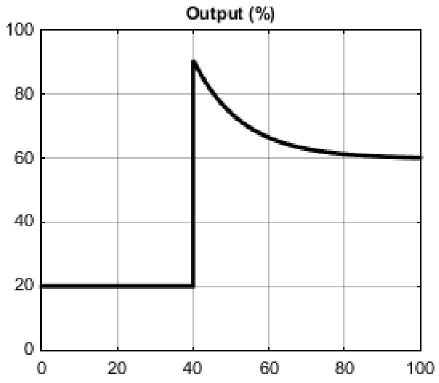
First, obtain the static characteristics of the plant. This can be achieved by putting the controller into Manual and, at a number of output values, record the final steady-state PV. Determine values of Gain and Bias that approximate the relationship, such that *OP = Gain*PV + Bias*.

If required, dynamic compensation can be used to change the feedforward output response. For example, it may speed things up even more if the output gives an initial ‘kick’ *in excess* of its final value before settling back down. A lead compensation can achieve this as discussed later.

Static or dynamic compensation

An example of Feedforward output response to SPchange with Static (left) and Dynamic (right) compensation is shown below.



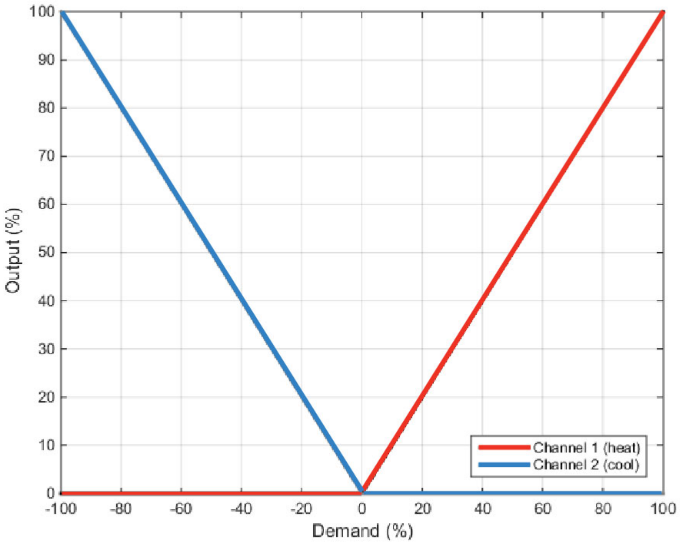


Split Range (heat/cool)

Inherent in the Loop is the concept of split-range for heat/cool.

Each Loop has a single setpoint and single PV, but can have *two* outputs. These two outputs operate in opposite directions. For example, consider a chamber with both a heater and a chiller. Both of these actuators are used to influence the temperature (the ‘process variable’, PV), but they operate in different directions: increasing heat output causes increasing PV, whereas increasing chiller output causes decreasing PV. Another example might be a gas carburizing furnace where the atmosphere is either enriched with methane (channel 1) or diluted with air (channel 2).

The way that the loop implements this is to allow the control output to extend over the span −100 to +100%. In this way, the range is split so that 0 to +100% is output on channel 1 (heat) and −100 to 0% is output on channel 2 (cool). The diagram below shows Split Range Outputs (Heat/Cool).



In addition, the loop allows each of the two channels to use different control types. The available control algorithm types are:

1. PID with an absolute output.
2. PID with valve positioning (without measured position and VPU).
3. Hysteretic On-Off (‘bang-bang’) control.

For example, a process may have an electrical heater on channel 1, controlled with the PID algorithm, whereas the flow of coolant through a jacket is modulated by a valve which is controlled by the VPU algorithm on channel 2. Transfer between the different algorithms is handled automatically.

In addition, different actuator gains are handled by having a separate proportional band for each channel.

Cooling Algorithm

The method of cooling may vary from application to application.

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

Non-linear cooling

The loop provides a set of curves that can be applied to the cooling (ch2) output. These can be used to compensate for cooling non-linearities thereby making the process 'look' linear to the PID algorithm. Curves for *Oil, Fan* and *Water* cooling are provided.

Curves are scaled to fit between 0 and the output low limit. Tuning the curve to the process is an important step in commissioning and can be achieved by adjusting the output low limit. The low limit should be set to the point at which the cooling effect is maximum, before it starts to drop off again.

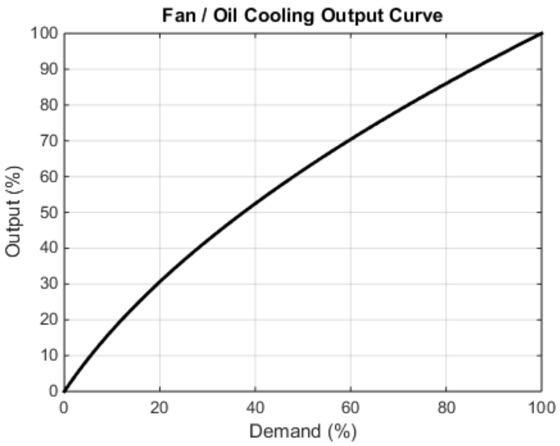
Be aware that any output rate limiting is applied to the output *before* non-linear cooling. Therefore, the actual controller output may change faster than any configured rate limit, but the power being delivered to the process will move at the correct rate, provided the curve has been correctly applied.

Air or Oil cooling

At low temperatures, the rate of heat transfer from one body to another can be considered linear and is proportional to the temperature difference between them. In other words, as the cooling media heats up, the rate of heat transfer slows down. So far, this is linear.

The non-linearity arises when a *flow* of cooling media is introduced. The higher the rate of flow (mass transfer), the less time a given 'unit' of media is in contact with the process and so the greater the average rate of heat transfer.

The air and oil characteristic is shown in the diagram below.

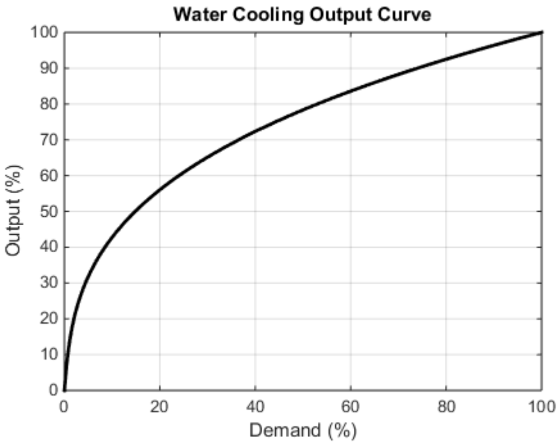


Evaporative Water cooling

Vaporizing water requires about five times as much energy as is required to raise its temperature from 0–100°C (32–212°F). This difference represents a large non-linearity, where at low cooling demands, the principal cooling effect is evaporative, but at higher cooling demands only the first few pulses of water flash off to steam.

To compound this, the mass transfer non-linearity described above for oil and air cooling is also true for water cooling.

Evaporative water cooling is often used in plastics extruder barrels and so this feature is ideal for that application. The evaporative water cooling characteristic is shown below.

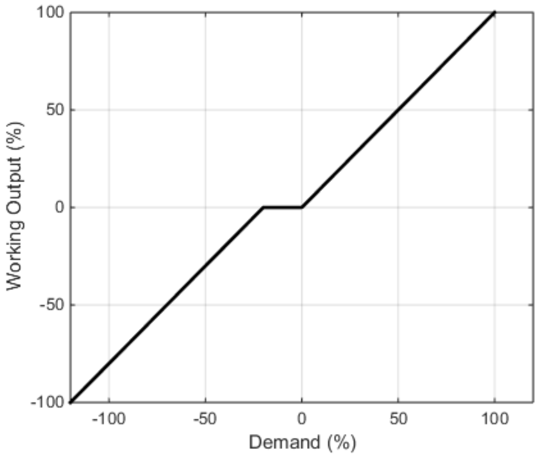


Channel 2 (heat/cool) Deadband

The channel 2 deadband introduces a gap between the point at which channel 1 switches off and the point at which channel 2 switches on, and vice versa. This is sometimes used to help prevent small and fleeting demands for cooling during normal process operation.

For a PID control channel, the deadband is specified in % output. For example, if the deadband is set to 10%, then the PID algorithm must demand −10% before ch2 will begin to switch on.

For an On/Off control channel, the deadband is specified in % of hysteresis. The diagram shows heat/cool with 20% deadband.



Bumpless Transfer

Where possible, the transfer to an Auto control mode from a non-Auto control mode will be ‘bumpless’. This means that the transition will go smoothly without large discontinuities.

Bumpless transfer relies on there being an integral term in the control algorithm to ‘balance out’ the step change. For this reason, it is sometimes called an ‘integral balance’.

The *IntBal* parameter allows the external application to request an integral balance. This is often useful if it is known that a step change in PV is to occur, for example a compensation factor has just changed in an oxygen probe calculation. The integral balance will help to prevent any proportional or derivative kick, instead allowing the output to be smoothly adjusted under integral action.

Sensor Break

‘Sensor Break’ is an instrument condition that occurs when the input sensor is broken or out of range. The Loop reacts to this condition by putting itself in Forced Manual mode (see above description). The type of transfer when entering Forced Manual, when the PV status is not good, can be selected using the *PVBadTransfer* parameter. The options are:

- Enter Forced Manual mode with the output set to the Fallback Value.
- Enter Forced Manual mode with the output held at the last good value (typically a value from about one second ago).

Operating Modes

The Loop has a number of possible operating modes. It is quite possible for several modes to be requested by the application at once. The active mode is therefore determined by a priority model, whereby the mode with the highest priority will ‘win’.

For details of the modes and their priorities, refer to "Loop" on page 160.

Start-up and Recovery

Proper start up is an important consideration and varies depending on the process. The Loop recovery strategy is followed under any of the following circumstances:

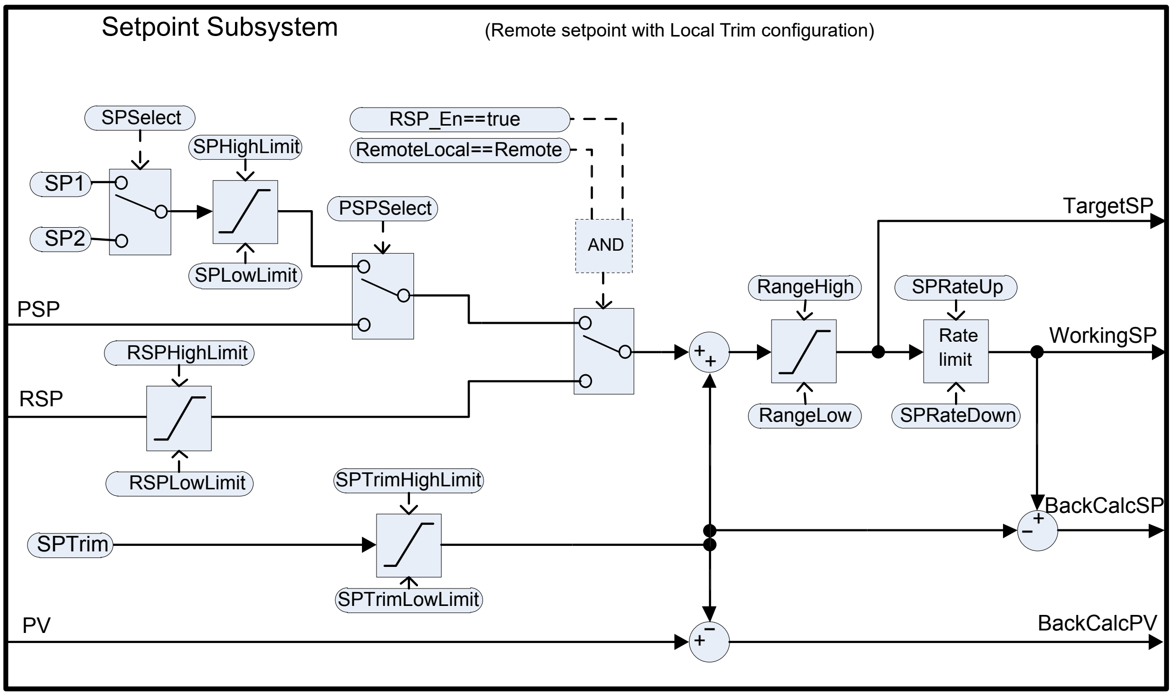
- Upon instrument start-up, after a power cycle, power outage event or power disruption.
- Upon exit from Instrument Configuration or Standby conditions.
- Upon exit from Forced Manual mode to a lower priority mode (e.g. when the PV recovers from a bad status or an alarm condition goes away).

The strategy to follow is configured by the *RecoveryMode* parameter. The two available options are:

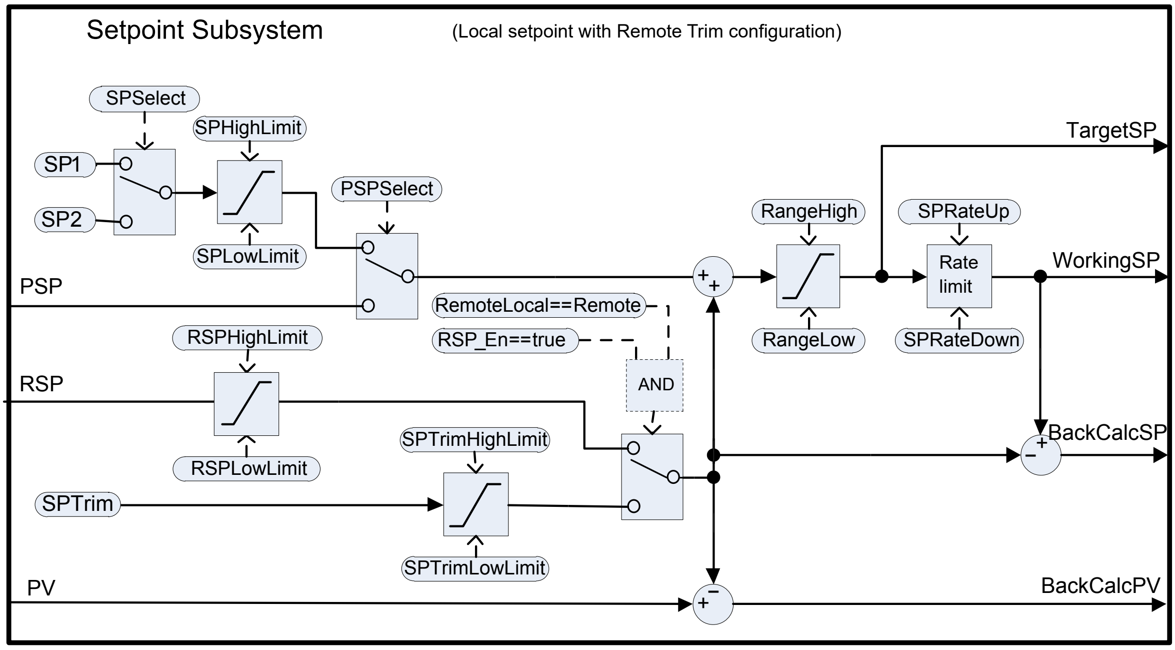
1. Last mode with last output
   The loop will return to Auto or Manual mode, whichever was last active. The working output will be initialized to the last used output value.
2. Manual mode with fallback output
   The loop will enter Manual mode. The initial output will be the configured Fallback Value, unless recovering from Forced Manual where the transfer will be bumpless.

Setpoint subsystem

The diagrams below show Setpoint function block. The first shows the ‘Remote Setpoint with local trim’ configuration.



The second diagram shows Setpoint subsystem in the ‘Local Setpoint with remote trim’ configuration.



The setpoint subsystem resolves and generates the working setpoint for the control algorithms. The working setpoint can ultimately come from several different sources, programmer, local or remote, have local or remote trims applied, and be limited and rate limited.

Remote/Local Setpoint Source Selection

The RemoteLocal parameter selects between the remote or local setpoint source.

The SPSource parameter reports which source is currently active. The three values are:

- Local – the local setpoint source is active.
- Remote – the remote setpoint source is active.
- F_Local – the remote setpoint source has been selected but it cannot become active. The local setpoint source is active until the exceptional condition is resolved.

In order for the remote setpoint source to become active, the following conditions need to be fulfilled:

1. The RemoteLocal parameter has been set to ‘Remote’.
2. The RSP_En input is true.
3. The status of the RSP input is Good.

Local Setpoint Selection

There are three local setpoint sources: the two operator setpoints, SP1 and SP2; and the programme setpoint, PSP. For selection parameters and priorities, refer to the above diagram.

Remote Setpoint

RSP is the remote setpoint source. It can be configured by the *RSPType* parameter in one of two ways:

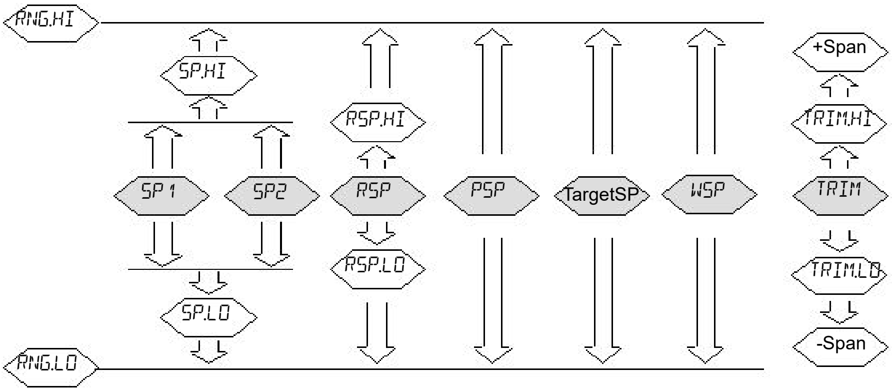
1. Remote setpoint (RSP) with a local trim (SPTrim).
   For example, in a continuous oven with several temperature zones, the master controller can transmit its setpoint to each slave’s RSP, and then a local trim can be applied in each slave to achieve the desired temperature gradient through the oven.
2. Local setpoint (SP1, SP2 or PSP) with a remote trim (RSP).
   For example, in a combustion air/fuel ratio application where the ratio setpoint is fixed, but a remote controller analyses the excess oxygen in the flue gases and is allowed to trim the ratio within a given band.

The remote setpoint is limited by the RSPHighLimit and RSPLowLimit parameters.

If a user wishes to write via MODBUS to the Control Loop RSP parameter via comms, it is strongly recommended to write that value via the RemoteInput block’s Input parameter instead, with the Output of the Remote Input block ("RemoteInput" on page 113) wired to the RSP parameter of the loop. This allows for the detection of unreliable communications, which would allow the loop to fall back to a local setpoint value.

Setpoint limits

The various setpoint parameters are subjected to limits according to the diagram below. Some of the limits themselves are also subjected to limits.



The *Span* is taken to be the value given by (*RangeHigh – RangeLow*).

Note: Whilst it is possible to set the RSP Limits outside the Range Limits, the RSP value will still be clipped to the Range Limits.

Setpoint Rate Limit

Rate limits can be applied to the final setpoint value. This can sometimes be useful to help prevent sudden step changes in controller output, and therefore help to prevent damage to the process or product.

Asymmetric rate limits are available. That is to say that the increasing rate limit can be set independently of the decreasing rate limit. This is often useful, for example in a reactor application whereby a sudden increase in flow should be reduced so that an exothermic event does not overwhelm the cooling control loop. On the other hand, a sudden decrease in flow should be permitted.

The setpoint rate limits may be set in units per hour, per minute or per second, according to the SPRateUnits parameter.

Note: When transitioning into an automatic control mode from a non-automatic control mode such as manual, the WSP will be set equal to the PV whenever a rate limit is set. It will then move towards the target setpoint from there at the configured rate.

In addition, if the SPRateServo parameter is enabled, the WSP will be set equal to the PV whenever the Target SP is changed and will then move towards the target from there. This only applies in Auto (including the transition to Auto) when SP1 or SP2 is active. It does not apply when using a remote or program setpoint.

Target SP

The Target SP is the setpoint value immediately prior to rate limiting (the Working SP is the value immediately after it). In many instruments it is possible to write to the Target SP directly. The effect of this is to trigger a back-calculation, which takes into account the trim value (either a local or remote trim), and then to write the back-calculated value to the selected setpoint source. This is so that the calculated Target SP on the next execution is equal to the entered value.

This usefully allows the target setpoint to be set to a desired value immediately, without having to manually make the calculations and determine which setpoint source is active.

Writing to the TargetSP is not possible when a remote setpoint is active.

Tracking

There are three setpoint tracking modes available. They can each be turned on by enabling the appropriate parameter.

1. SP1/SP2 tracks PV
   Whilst the mode is MANUAL, whichever of SP1 or SP2 is active will track the PV (less the trim). This is so that the operating point is maintained whenever the mode is changed to Auto.
2. SP1/SP2 tracks PSP
   Whilst PSPSelect is enabled, whichever of SP1 or SP2 is active will track the PSP. This is so that the operating point is maintained when the programmer is reset and PSPSelect goes false.
3. SP1/SP2/SPTrim tracks RSP
   When the RSP is active and acting as a Remote Setpoint, whichever of SP1 or SP2 is active will track RSP. If RSP is acting as a Remote Trim, then it is SPTrim that will track RSP. This is so that the operating point is maintained if the setpoint is switched to local.

Back-calculated SP and PV

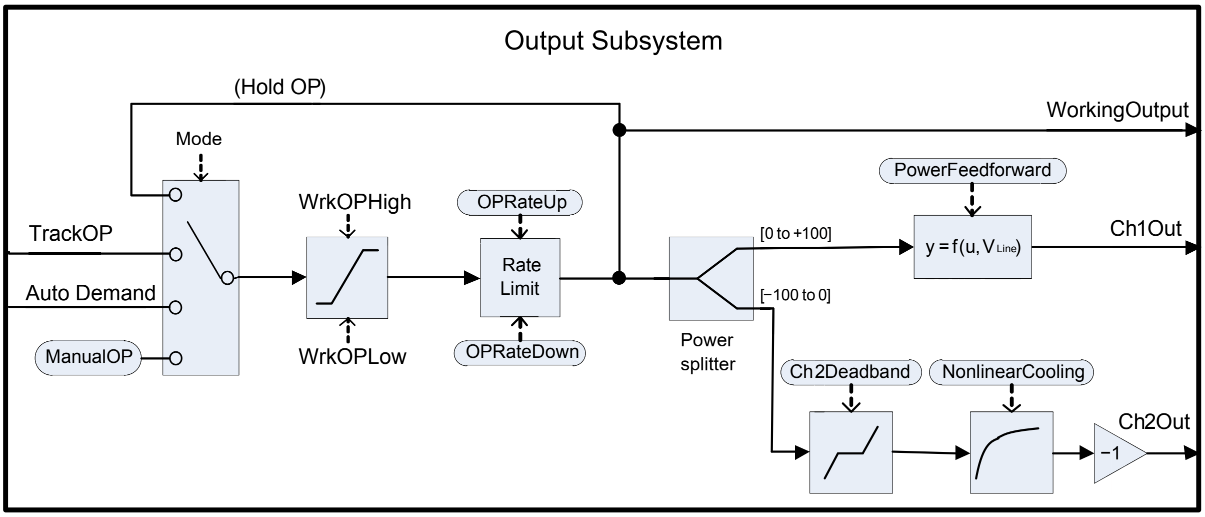
Back-calculated versions of WSP and PV are provided as outputs. These are simply WSP/PV minus the active trim value. These outputs are given so that an external setpoint source (such as a setpoint programmer or a cascade master) can track their output to them as necessary, thus helping to prevent bumps on mode changes and switchovers.

Setpoint Integral Balance

When the SPIntBal parameter is enabled, the setpoint subsystem will issue an integral balance request to the PID/VPU algorithms whenever a step change in SP1 or SP2 occurs. This will cause any proportional or derivative kick to be suppressed and the PV will move smoothly to the new setpoint with the integral as the driving force and with minimum overshoot. The effect is the same as what is sometimes called ‘proportional and derivative on PV’ instead of error, but only applies to step changes in SP1 or SP2 and on transition to local setpoint from remote.

Output Subsystem

The diagram shows the block diagram of the Output subsystem.



Output selection (including Manual Station)

The source of the output demand is resolved according to which controller mode is active. In HOLD, the previous Working Output is held. In TRACK, the output demand is taken from TrackOP. In MANUAL and F_MAN, the output is taken from ManualOP. In other modes, the output is taken from the control subsystems output.

Output Limiting

The resolved demand is subject to position limiting. There are several different sources of position limits:

- The master limits, *OutputHighLimit* and *OutputLowLimit*
- The active gain scheduled limits: *OutputHigh(n)* and *OutputLow(n)*
- The remote limits, *RemoteOPHigh* and *RemoteOPLow*
- The tune limits (only during auto-tuning), *TuneOutputHigh* and *TuneOutputLow*

The most restrictive limits take priority. That is to say, the minimum of the upper limits and the maximum of the lower limits are used. These become the working output limits, *WrkOPHigh* and *WrkOPLow.*

The output limits are applied in Auto modes. In non-Auto modes such as Manual, the *FallbackValue* may override a limit if that limit would help prevent the *FallbackValue* being achieved. For example, if the *OutputLowLimit* is 20% and the *FallbackValue* is 0%, then in Auto the working low limit will be 20%, while in Manual it will be 0%.

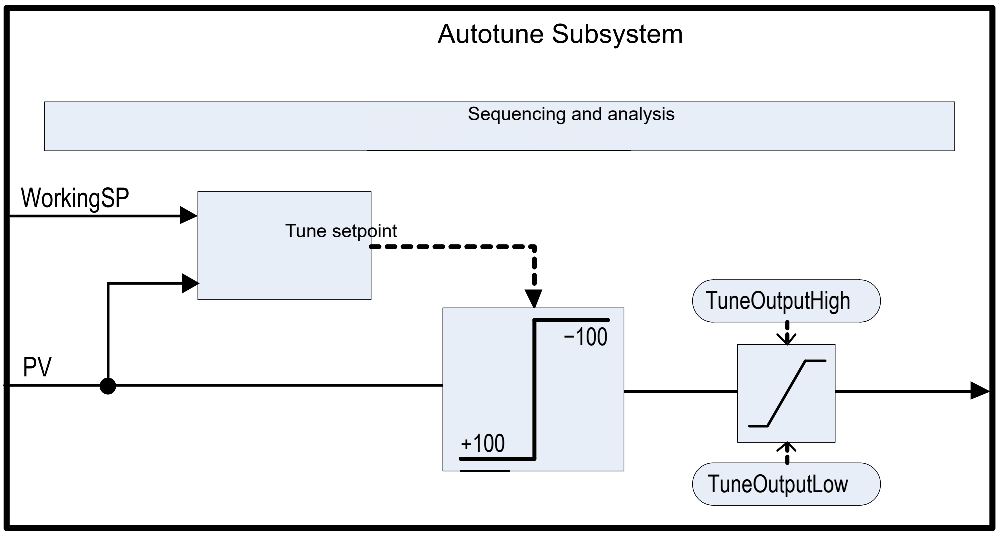
The remote output limits are only applied in Auto mode.

Rate limiting

The working output can be rate limited by setting the two parameters, *OPRateUp* and *OPRateDown*. They are specified in % per second. Output rate limiting is only available for PID control channels and should be used only where necessary since it can significantly impact process performance.

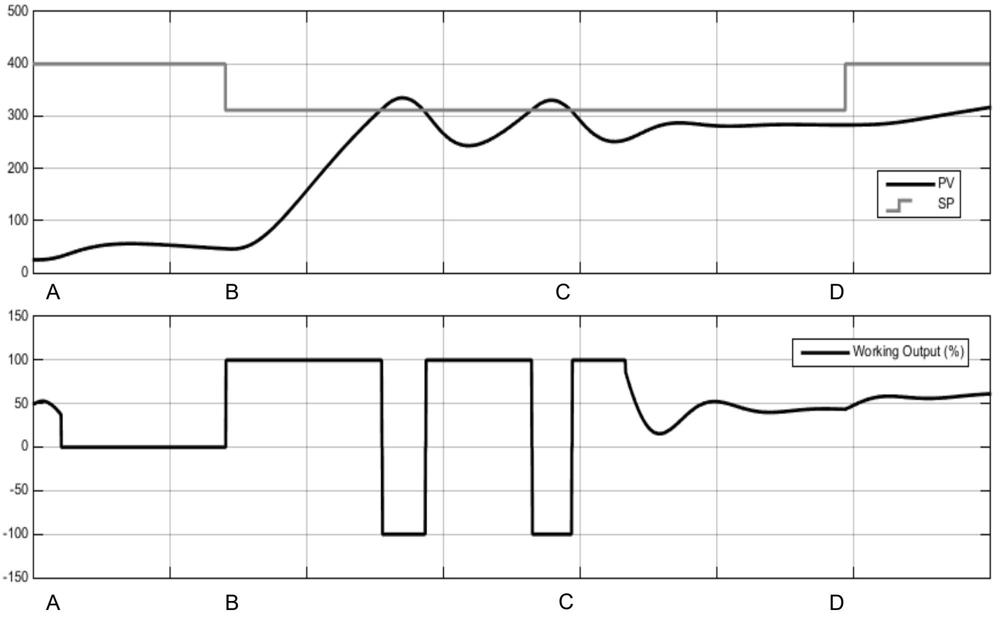
Autotuning

The diagram below shows a simplified structure of a relay-based Autotuner.



The function block contains sophisticated autotuning algorithms that can tune the controller to the process. They work by performing experiments on the plant, by inducing perturbations, and observing and analysing the response. The autotuning sequence is described in detail below.

The diagram shows an example of heat/cool Autotune with 'alternative' CH2 Tune Type.



Time	Description
A	**Autotune Begins** Setting the *AutotuneEnable* parameter to On and the controller mode to Auto will cause the autotune to begin. Before starting an autotune, you should turn off the PID terms that you do not want to use. For example, setting TD to Off will disable derivative action and the autotuner will therefore tune for a PI controller. If you do not want any integral, set TI to Off and the autotuner will tune for a PD controller. If the cutback thresholds, CBH and CBL, are set to Auto then the autotuner will not attempt to tune them. An autotune may be triggered at any time, but it will not begin until the mode goes to Auto. Similarly, autotune will abort if the mode is changed away from Auto at any time during the tune, including reasons such as sensor status is bad. In this case it will be necessary to start autotune again. Note that the PID tuning constants will be written to whichever gain set is active when tuning completes.
A to B	**Initial Delay** This period persists for precisely one minute. If the PV is already at the WSP then the working output will be frozen. Otherwise, the output is set to 0 and the process is allowed to drift while some initial measurements are made. The target setpoint may be changed during this initial delay, but not after it. You should set the target setpoint to the operating point at which you would like to tune. Care should be taken in setting the setpoint, to help to ensure that oscillations of the process will not damage the process or load. For some processes, it may be necessary to use a setpoint for tuning purposes that is below the normal operating point.
B	**Calculate Tune Setpoint** Once the initial delay has elapsed, the tune setpoint is determined. It is calculated by: If PV = Target SP: Tune SP = Target SP If PV < Target SP: Tune SP = PV + 0.75(Target SP – PV) If PV > Target SP: Tune SP = PV − 0.75(PV – Target SP) Once determined, this tune setpoint will be used for the duration of the autotune and any changes to the target setpoint will be ignored until the autotune has completed. If you wish to change the tuning setpoint, abort and restart the autotune.

Time	Description
B to C	**Relay Experiment** The autotuner will now insert a relay into the closed-loop. This establishes the limit-cycle oscillations in PV. The relay operates such that: If PV > SP: OP = minimum If PV < SP: OP = maximum The minimum and maximum outputs are determined by the various limits. There is also a small amount of hysteresis, not described, around the relay switching point to help prevent EMI (Electromagnetic interference) from causing nuisance switching. The number of oscillations required before moving to the next stage depends on the controller configuration: If either channel is configured for VPU, or OnOff control, or if output rate limiting is enabled, then the ‘Fourier’ autotune algorithm will run. This requires three cycles of oscillation. If only PID is configured and there is no output rate limiting, then the ‘PID’ autotune algorithm will run. Only two cycles of oscillation are required. There will be an additional half-cycle of oscillation at the beginning of this stage if the initial PV is above the SP. Once the required number of cycles is achieved, the algorithm moves to the next stage.
C to D	**Relative Channel 2 Tuning Experiment** This stage is only used for dual-channel heat/cool configurations. For heat-only or cool-only, it is skipped. The purpose of this stage is to determine the relative gain between channel 1 and channel 2. This is used to set the correct proportional bands. For example, in a heat/cool process, the heater and the chiller are typically not equally rated, e.g. the heater might be capable of putting much more energy into the process in a given period of time than the chiller is capable of removing. This non-linearity needs accounting for and the purpose of this additional experiment is to collect the necessary information to make that correction. The type of experiment that is used can be selected with the Ch2TuneType parameter: The *Standard* experiment is the default and gives good results for most processes. It will put the process into an additional cycle of oscillation but instead of applying minimum output, it will apply 0 output and allow the PV to drift. This option is not available if TuneAlgo is Fourier. The *Alternative* experiment is recommended for processes that do not exhibit significant losses—for example, a very well-lagged tank or oven. It attempts to control the PV to the SP and collects data about the process input required to do so. The length of this stage is equivalent to between 1.5 and 2 oscillation cycles. The *KeepRatio* option should only be selected when the relative gain of the two channels is well known. It causes this stage to be skipped and instead the existing proportional band ratio will be maintained. So, for example, if you know that the heating channel will deliver a maximum of 20kW and the cooling channel will deliver a maximum of -10kW, then setting the proportional bands such that the ratio Ch2PB/Ch1PB = 2 before autotuning will allow the correct ratio to be maintained.

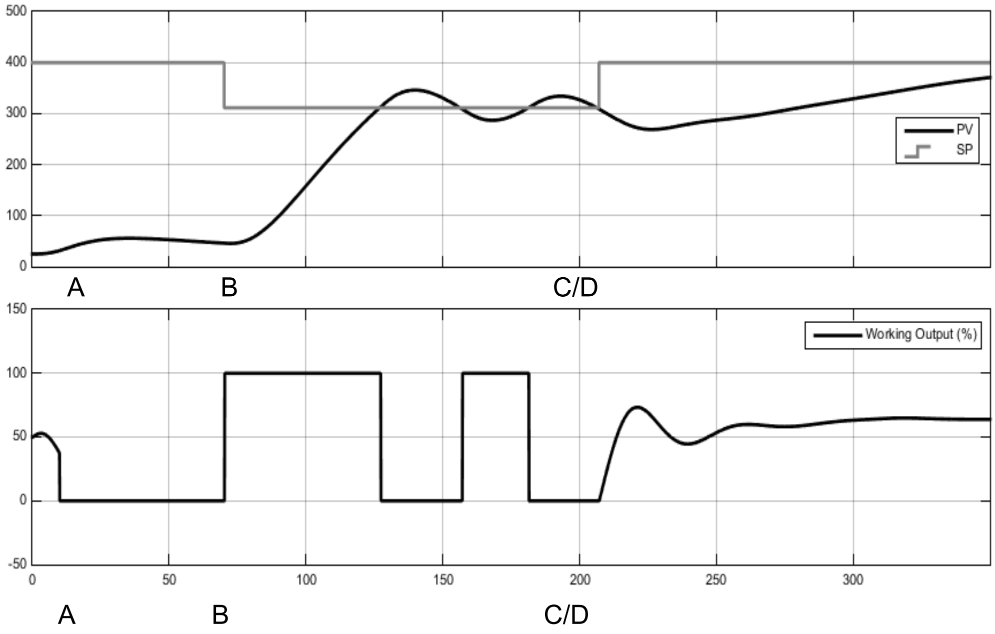
Time	Description
D	**Analysis and Completion** The autotune experiments are now complete. Finally, some analysis will be performed on the collected data and the controller tuning constants will be chosen and written to whichever gain set is active. This analysis may take a number of seconds, typically less than 15, during which the output will be frozen. After the tune has completed, the working setpoint is released and can be modified in the usual manner. Authority over the output is returned bumplessly to the control algorithms.

Notes:

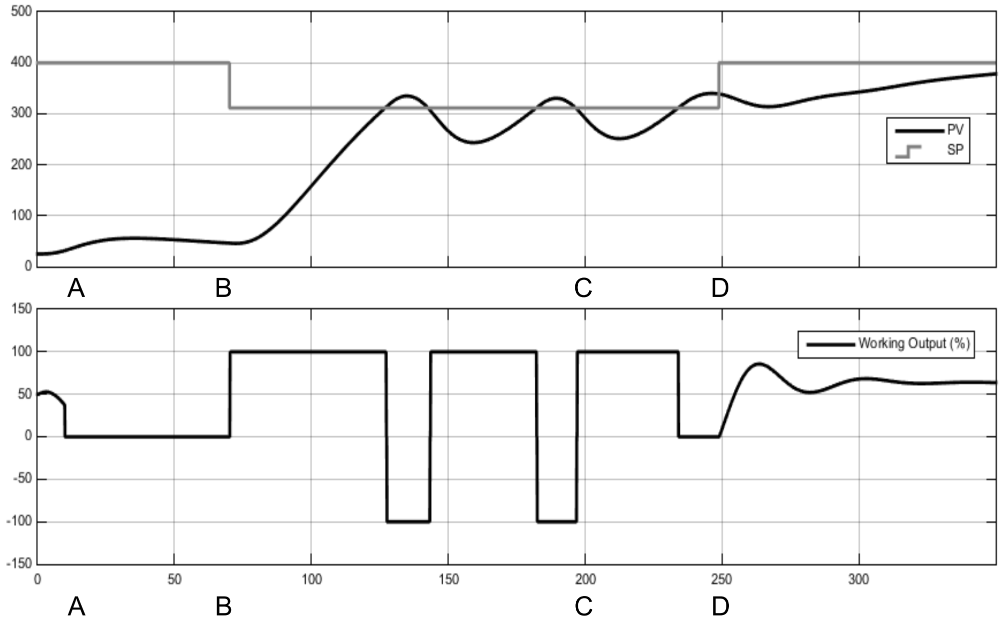
- If any stage of the autotune sequence exceeds two hours in duration, the sequence will timeout and be aborted. The StageTime parameter counts up the time in each stage.
- Channels configured for OnOff control cannot be autotuned but they will be exercised during the experiments if the opposite channel is not OnOff.
- For VPU channels, it is important that the associated Travel Time parameter is set as accurately as possible before commencing the autotune.
- Carbon Potential loops, which have a setpoint in the range 0–2.0% (and other loops with small setpoint ranges), cannot be autotuned if the proportional band type is set to ‘Engineering Units’. For these loops, proportional band type should be set to ‘Percent’ and the RangeHigh and RangeLow set correctly. This allows autotune to work.

A number of further examples under different conditions are pictured below.

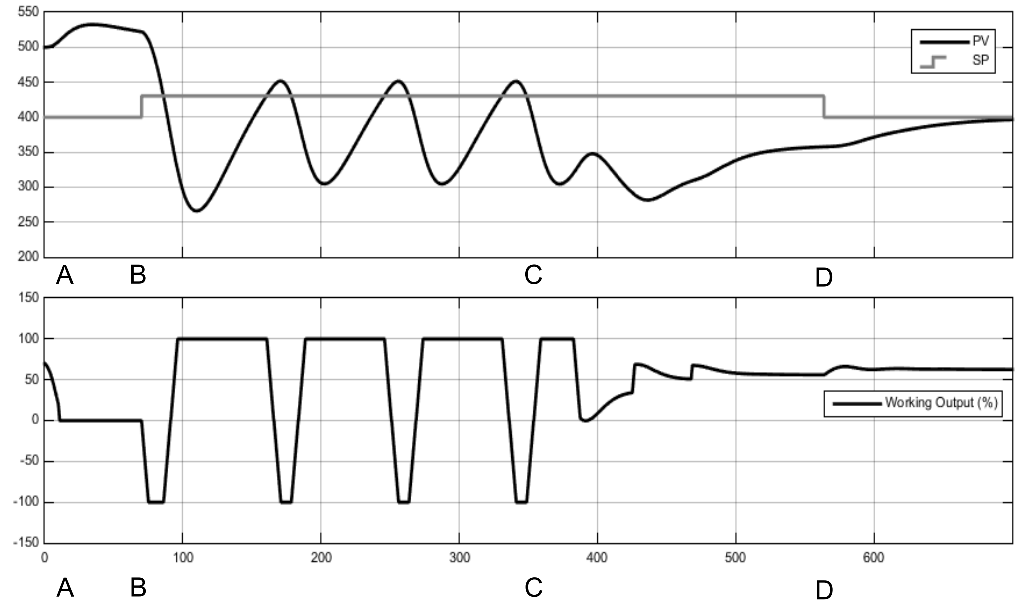
The first shows an example Heat-only Autotune.



The second example shows Heat/Cool Autotune with ‘Standard’ Ch2 tune type.



The third shows an example of Heat/Cool Autotune from above with output rate limiting.



Autotuning multiple zones

Autotune relies on the principle of cause and effect. It perturbs the process and then watches for what the effect will be. It is therefore essential that all external influences and disturbances are minimized as much as possible during an autotune.

When autotuning a process that has multiple interacting loops, for example a furnace with multiple temperature zones, each loop should be autotuned separately. They *should not* under any circumstances be autotuned at the same time, since the algorithms will not be able to ascertain what cause produced what effect. The procedure below should be followed:

1. Place all loops in manual and set the outputs to the approximate steady state value for the desired operating point. Allow the process to settle.
2. Enable autotune on a *single zone.* Allow the tune to complete.
3. After the zone has finished autotuning, allow it to settle out in auto and then place it back into manual.
4. Repeat steps 2 and 3 for each zone.

Digital Communications

EPC2000 Programmable ControllerDigital Communications (or ‘comms’ for short) allows the controller to communicate with a PC, a networked computer system, or any type of communications master using the protocols supplied. Connections to the PC are shown in "Digital Communications Connections" on page 45. A data communication protocol defines the rules and structure of messages used by all devices on a network for data exchange. Communications can be used for many purposes - SCADA packages; PLCs; data logging for archiving and plant diagnostic purposes; cloning for saving instrument set ups for future expansion of the plant or to allow a controller to be replaced by a spare.

Serial Communications

The serial comms protocol available is Modbus RTU.

Modbus RTU

Modbus (JBUS) Protocol defines a digital communication network to have only one master and one or more slave devices. Either a single or multi-drop network is possible. All message transactions are initiated by the master. Eurotherm instruments communicate using the Modbus RTU binary protocol.

The JBUS protocol is identical in most respects to Modbus protocol - the main difference being that Modbus uses a 0-based register addressing whereas JBUS uses 1-based register addressing.

The list of Modbus addresses is available in iTools by opening the browser list.

For further information on the Modbus protocol see www.modbus.org.

Serial Comms Parameters

The parameters which follow are applicable to Modbus RTU.

Baud Rate

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

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

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

Parity

Parity is a method of ensuring that the data transferred between devices has not been corrupted. It ensures that every single byte in the received message contains the same number of ones and zeros when it was received as when it was transmitted.

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

Communication Address

On a network of instruments a comms address is used to identify a particular instrument. Each instrument on a network should have a unique comms address. Address 255 is reserved for the configuration port.

Comms Delay

In some systems it is necessary to introduce a delay between the instrument receiving a message and sending its reply. is sometimes necessary if the line transceivers require an extended time to switch to tristate.

Ethernet Configuration

MAC address display

The two Ethernet ports on the front of the EPC2000 Programmable Controller share a single MAC address, presented as a 12-digit hexadecimal number in the format ‘aa-bb-cc-dd-ee-ff’.

In the EPC2000 Programmable Controller, MAC addresses are shown as six separate decimal values in the ‘COMMS’ list. MAC1 shows the first pair of digits (example ‘170’), MAC2 shows the second pair of digits and so on.

The MAC address is viewable using iTools by looking at the Comms.Ethernet.Network function block.

IP Mode Settings

It will generally be necessary to consult with your network administrator to determine if the IP address, Subnet mask and Default gateway for the instruments should be static or whether they should be dynamically allocated by a DHCP server.

For instruments with Static IP mode, network configuration must be entered manually in the Comms.Option.Network IP address, Subnet mask and Default gateway parameters.

Dynamic IP Addressing

Within the ‘Option Comms’ list of the instrument, set the ‘IP Mode’ parameter to ‘DHCP’. Once connected to the network and powered, the instrument will acquire its IP address, Subnet Mask and Default gateway parameters from the DHCP Server and display this information within a few seconds.

If DHCP is active but the DHCP server cannot be contacted the IP address will fall back to AutoIP addressing in the 169.254.xxx.yyy address range.

If a valid DHCP IP address lease expires and the server is not contactable then the address will revert to AutoIP addressing in the 169.254.xxx.yyy address range.

Static IP Addressing

Within the Comms.Ethernet.Network function block of the instrument, ensure the ‘IP Mode’ parameter is set to ‘Static’, then set the IP address and Subnet Mask and Default gateway as required (and defined by your network administrator).

See section "Comms.Serial.Network and Comms.Ethernet.Network" on page 129.

Network Connection

A RJ45 connector is used to connect to the instrument's 2-port Ethernet interface to a 100BaseT or 10BaseT switch/hub using standard CAT5 cable. The Ethernet interface on the instrument is auto-switching so specific cross-over cables are not required.

Broadcast Storm Protection

Broadcast Storm Protection discards all broadcast packets if the broadcast rate climbs too high. Broadcast Storm and Ethernet Rate Protection are intended to favor maintaining the control strategy in certain high traffic network environments.

The diagnostic parameters Broadcast Storm and Rate Protection, see section "Comms.Serial.Network and Comms.Ethernet.Network" on page 129, indicate when the protection is active.

Ethernet Rate Protection

Certain excessive network loads on embedded products have the potential to impact processor availability to the point that useful control is compromised and the product restarts as there are no further CPU resources available to service the device watchdog.

The EPC2000 Programmable Controller incorporates an Ethernet Rate Protection algorithm which will de-prioritize Ethernet comms in very heavy traffic environments so that the control strategy continues and the instrument does not watchdog reset.

Additional information

The Comms.Option.Network function block also includes configuration settings for ‘Default Gateway’, these parameters will be set automatically when DHCP IP mode is used. When static IP mode is used these settings are only required if the instrument needs to communicate across subnets - see your network administrator for the required setting.

Bonjour

Bonjour™ is an implementation of Zeroconf, which allows a ‘plug and play’ type feel to instrument connectivity by providing a method of auto discovery of a device on an Ethernet network and therefore negates the need for network configuration by the user. It is used to provide an easy route for Ethernet connectivity configuration in the EPC2000 Programmable Controller.

Bonjour™ is released under a terms-of-limited-use license by Apple.

Note: For cybersecurity reasons, the Bonjour™ service is disabled by default, as it makes it easier for a malicious user to discover and access the controller via the network. To turn on Bonjour™ auto-discovery, use the auto discovery parameter as described in the following sections.

Auto Discovery

The ‘Auto Discovery’ parameter set to ‘On’ implements Bonjour™, which means it is not necessary to add the EPC2000 Programmable Controller IP address to the iTools control panel applet.

Note: EPC2000 PROFINET does not have the Auto Discovery feature see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266 for further information.

To turn on Auto Discovery

Scan QR Code for EPC2000 ‘How To’ video tutorials.

Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

There are two methods to turn on Auto Discovery:

- using iTools and whilst in Configuration Mode, modifying the Comms.Ethernet.Network.AutoDiscovery parameter to ’On’, or
- pressing the Function button in a specific sequence (see below) whilst powering-on the EPC2000 Programmable Controller.

To perform method 1, you will already need your PC to be connected to the EPC2000 Programmable Controller over Serial Comms or Ethernet. Then use iTools to turn on the Auto Discovery parameter in the Comms.Ethernet.Network function block whilst in Configuration Mode. To perform the second method, use the following procedure:

Switching on Auto Discovery

1. Ensure no active process is being controlled.
2. If on, power-off the EPC2000 Programmable Controller and wait for all LEDs to extinguish.

3. Insert a small, suitable, insulated tool into the Function button slot to press the recessed button.

⚠ ⚠ DANGER

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Ensure only a suitable insulated tool that fits the aperture is used to depress the function button when required.

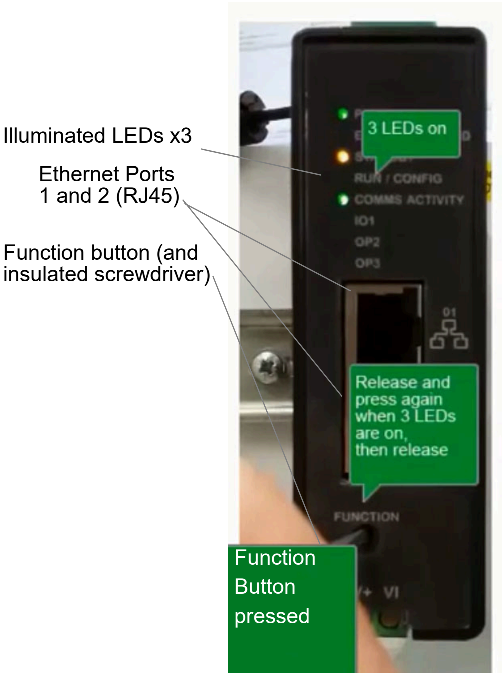
Failure to follow these instructions will result in death or serious injury.

4. Keep pressing the Function button whilst restoring power to the EPC2000 Programmable Controller. Carefully watch the front-panel LEDs as the timing is important.

5. Once the EPC2000 Programmable Controllers power is restored, all the front panel LEDs illuminate before extinguishing again, as part of a power-up self test.

6. When only three LEDs (Power, Standby and Comms Activity) illuminate, quickly release the Function button, before briefly pressing and releasing once again.

   The EPC2000 Programmable Controllers Auto Discovery function will now be enabled (switched on), allowing iTools to find the device when on the same network.


Illuminated LEDs x3
Ethernet Ports 1 and 2 (RJ45)
Function button (and insulated screwdriver)
3 LEDs on
RUN / CONFIG
COMMS ACTIVITY
IO1
OP2
OP3
Release and press again when 3 LEDs are on, then release
FUNCTION
Function Button pressed
VI

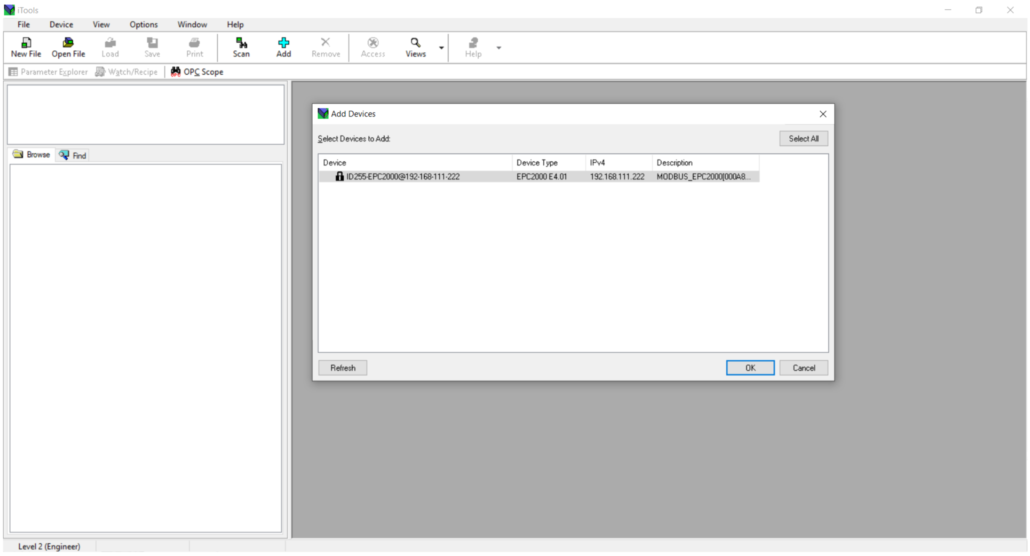
7. Make sure the EPC2000 Programmable Controller is connected to the Ethernet network it will operate on, using an appropriate Ethernet network cable connected to one of the EPC2000 Programmable Controller Ethernet ports (1 or 2) with an RJ45 connection.

   Note: Make sure that the controller and PC which is running iTools are on the same subnet.

8. Open iTools, Eurotherm’s software suite for configuring Programmable Controllers, see "What is iTools?" on page 65 for further details.

9. In iTools select 'Add' Add from the iTools menu bar, the *Add Devices* panel will appear and in the list of devices connected via Ethernet will be the EPC2000 Programmable Controller.

Note: If comms password is not set then iTools will ask the user to set the comms password before connecting to the EPC2000 in configuration mode.

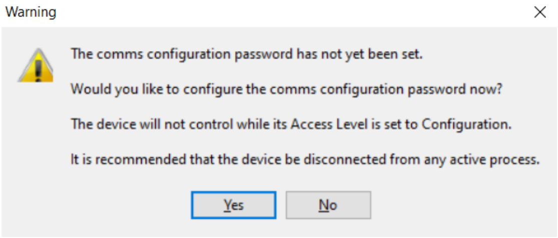


Add Devices

Select Devices to Add:

Device	Device Type	IPv4	Description
ID255-EPC2000@192-168-111-222	EPC2000 E4.01	192.168.111.222	MODBUS_EPC2000[000A8...

10. Select discovered controller and click OK. iTools will open a dialog requesting the user to set a comms configuration password.



Warning

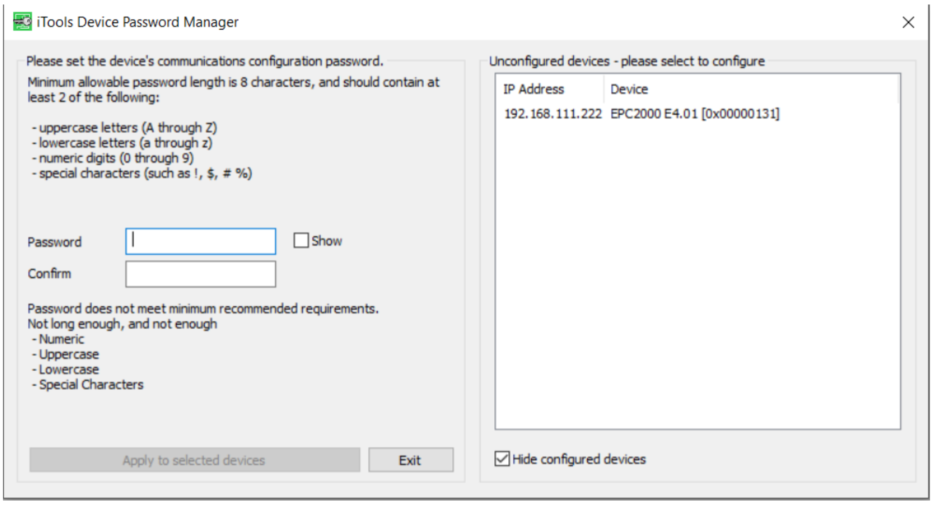
The comms configuration password has not yet been set.

Would you like to configure the comms configuration password now?

The device will not control while its Access Level is set to Configuration.

It is recommended that the device be disconnected from any active process.

11. Select "Yes" to set comms config password and iTools Device Password Manager window will open. Follow instructions to set the comms configuration password.



iTools Device Password Manager

Please set the device's communications configuration password.

Minimum allowable password length is 8 characters, and should contain at least 2 of the following:

- uppercase letters (A through Z)
- lowercase letters (a through z)
- numeric digits (0 through 9)
- special characters (such as !, $, # %)

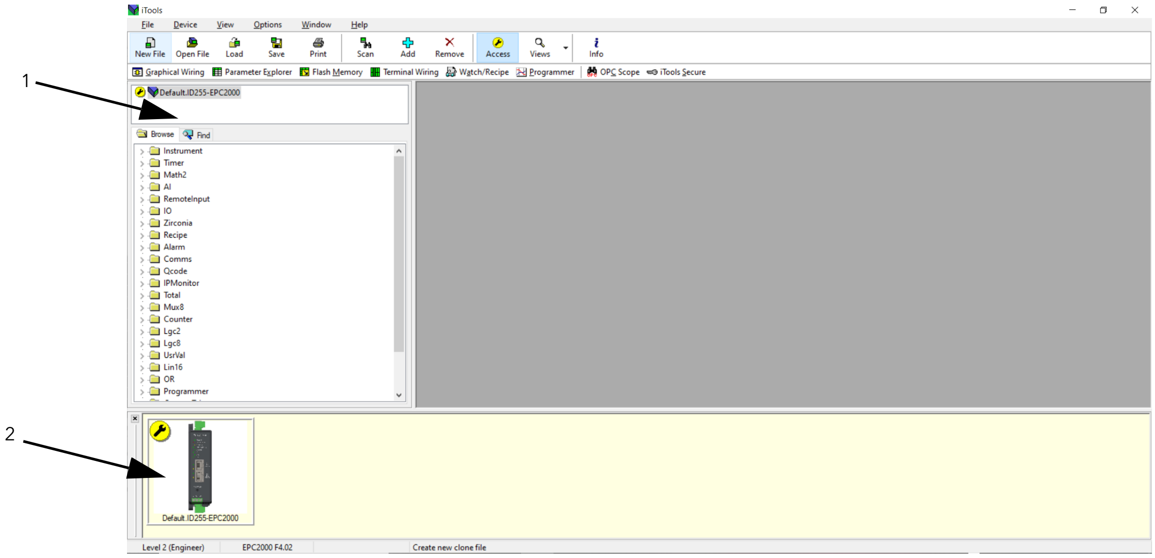
Password does not meet minimum recommended requirements.
Not long enough, and not enough
- Numeric
- Uppercase
- Lowercase
- Special Characters

Unconfigured devices - please select to configure

IP Address	Device
192.168.111.222	EPC2000 E4.01 [0x00000131]

12. Once the password has been set, the EPC2000 Programmable Controller connects with the instrument in configuration mode and the following items appear:
    - the device name and number, in top left window (1).

- an image in view panel window (2).


1
2
iTools
File Device View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Info
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Default.ID255-EPC2000
Browse Find
Instrument
Timer
Math2
AI
RemoteInput
IO
Zirconia
Recipe
Alarm
Comms
Qcode
IPMonitor
Total
Mux8
Counter
Lgc2
Lgc8
UsrVal
Lin16
OR
Programmer
Default.ID255-EPC2000
Level 2 (Engineer) EPC2000 F4.02 Create new clone file

To maintain Cybersecurity best practice, it is recommended that AutoDiscovery is disabled when not required i.e. after Initial set up, deactivate the AutoDiscovery function. For further details, see Auto Discovery parameter in "Comms.Serial.Network and Comms.Ethernet.Network" on page 129.

To turn on DHCP

13. From 7 above, use iTools to locate the Comms.Ethernet.Network function block and change the IPMode parameter to DHCP.

The instrument will receive its address from the network. Note, however, that DHCP servers can assign different addresses over time to the same controller and it can therefore be difficult to work out which address has been given to any specific controller.

Reset the Controller’s IP address

Scan QR Code for EPC2000 ‘How To’ video tutorials.

Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

If the IP address is changed and you are unable to remember the IP address it was changed to, it can be restored to the default settings. For further information regarding the default IP Address and associated details see "Default IP address, details and Password" on page 57.

Note: In performing this procedure, the Auto Discovery parameter is also set to Off and Comms configuration password is cleared.

To restore these default settings, perform the following procedure:

1. Ensure no active process is being controlled.
2. Power-off the EPC2000 Programmable Controller and wait for all LEDs to extinguish.

3. Insert a small, flat-bladed screwdriver into the Function button slot to press the recessed button

⚠ WARNING

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Ensure only an suitable insulated tool that fits the aperture is used to depress the function button when required.

Failure to follow these instructions can result in death, serious injury or equipment damage.

4. Keep pressing the Function button whilst restoring power to the EPC2000 Programmable Controller. Carefully watch the front-panel LEDs as the timing is important.

5. After the EPC2000 Programmable Controller's power is restored, all the front-panel LEDs illuminate before extinguishing again. Then only three LEDs illuminate (Power, Standby and Comms Activity). Finally, all the LEDs on the left flash, followed by all the LEDs on the right - as the right side LEDs flash, quickly release the Function button and briefly press and release a second time.

⚠ WARNING

HAZARD OF ELECTRIC SHOCK, EXPLOSION OR ARC FLASH

Ensure only an suitable insulated tool that fits the aperture is used to depress the function button when required.

Failure to follow these instructions can result in death, serious injury or equipment damage.

6. The EPC2000 Programmable Controller resets, returns the IP address and other associated settings to the default setting - see "Default IP address, details and Password" on page 57. Auto Discovery is disabled and comms configuration password is cleared.

Connecting to EPC2000 using iTools

Scan QR Code for EPC2000 'How To' video tutorials.

Further details at:
https://www.eurotherm.com/temperature-control/epc2000-how-to-tutorials/

iTools configuration package, version V9.78 or later, may be used to configure Ethernet communications. If Auto Discovery is not used, iTools must be set up for Ethernet as described in the following instructions.

Connect the controller to the PC using an Ethernet cable with RJ45 connectors. To establish a connection, the IP address of the EPC2000 Programmable Controller needs to be known. If you do not know the IP address of the controller see "Default IP address, details and Password" on page 57, for some versions you could use the function "Auto Discovery" on page 229.

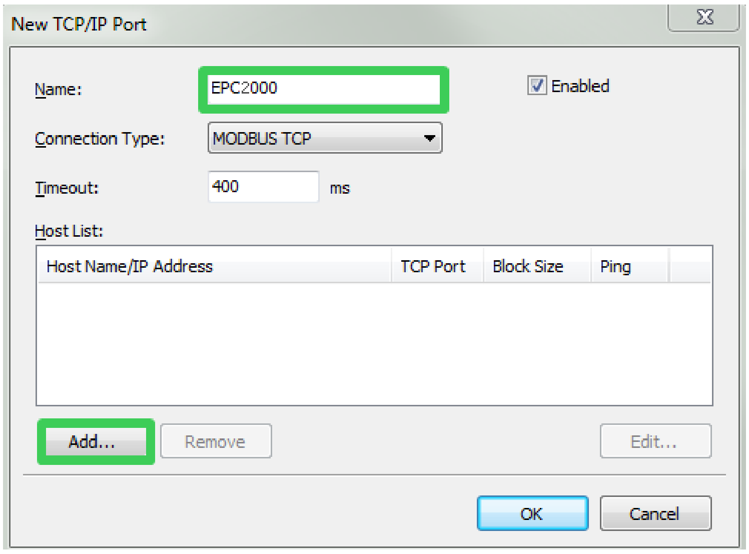
Note: The EPC2000 PROFINET version will require the IP address to be configured using a PROFINET Configuration tool, ensure this has been completed before continuing for further information, see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266.

Adding a Device to iTools Control panel

To include a Host Name/Address within the iTools scan:

Note: Ensure iTools is NOT running before taking the following steps.

1. Within Windows, open the ‘Control Panel’. If the Control Panel is open in ‘Category View’ select Large or Small Icons instead.
2. Double-click on ‘iTools’ to open the iTools control panel, the iTools configuration panel will appear.
3. Within the iTools configuration settings select the TCP/IP tab.
4. Click the 'Add' button to add a new connection, the New TCP/IP Port panel will appear.
5. Type in a name of your choice, for example ‘EPC2000 Programmable Controller’ and click Add. (Ensure that no duplicate IP address entries are enabled at the same time).



New TCP/IP Port

Name: EPC2000

☑ Enabled

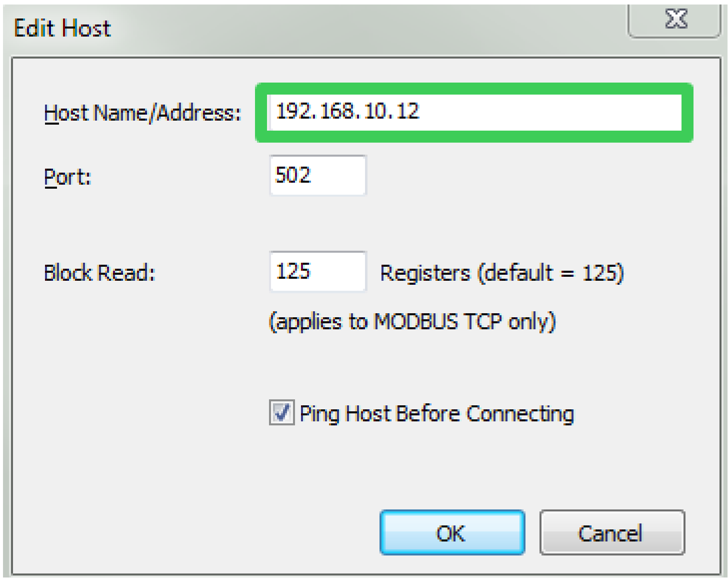
Connection Type: MODBUS TCP

Timeout: 400 ms

Host List:

Host Name/IP Address	TCP Port	Block Size	Ping

6. The Edit Host panel appears, enter the IP address of the device ensuring that the PC IP address is in the same range as the EPC2000 Programmable Controller then click OK.



Edit Host

Host Name/Address: 192.168.10.12

Port: 502

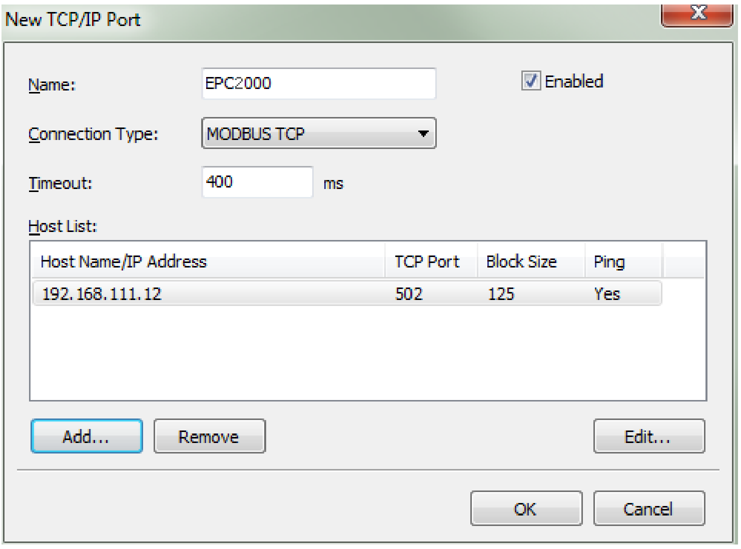
Block Read: 125 Registers (default = 125)

(applies to MODBUS TCP only)

☑ Ping Host Before Connecting

Note: The EPC2000 Programmable Controller's default IP address and associated details are available, see "Default IP address, details and Password" on page 57.

7. The New TCP/IP Port panel appears, confirm the IP address is correct, then click OK to commit the new TCP/IP port details into the iTools Control panel.



New TCP/IP Port

Name: EPC2000

☑ Enabled

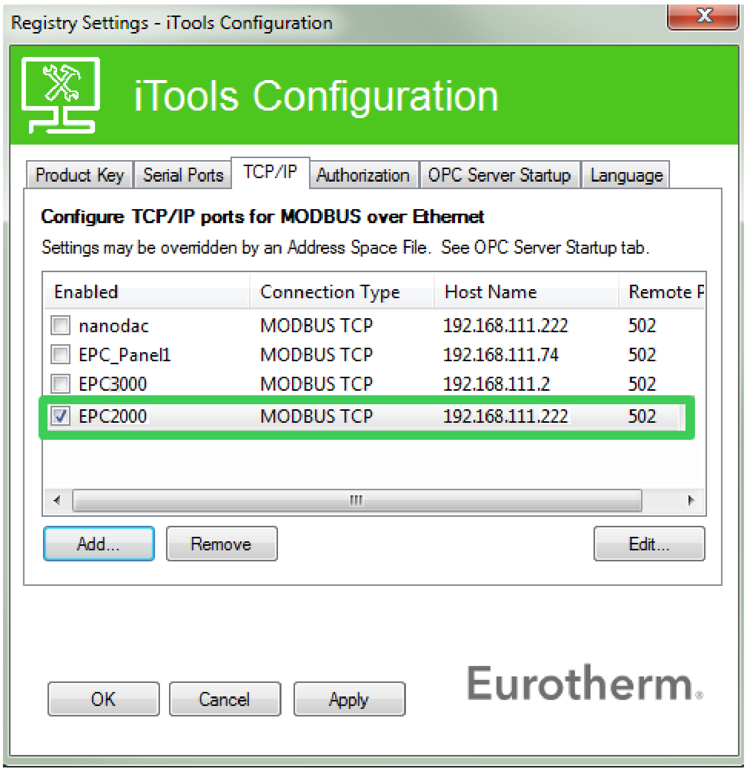
Connection Type: MODBUS TCP

Timeout: 400 ms

Host List:

Host Name/IP Address	TCP Port	Block Size	Ping
192.168.111.12	502	125	Yes

8. The iTools Control panel appears displaying the new TCP/IP port you have just added, select OK to add the new entry.



Registry Settings - iTools Configuration

iTools Configuration

Configure TCP/IP ports for MODBUS over Ethernet

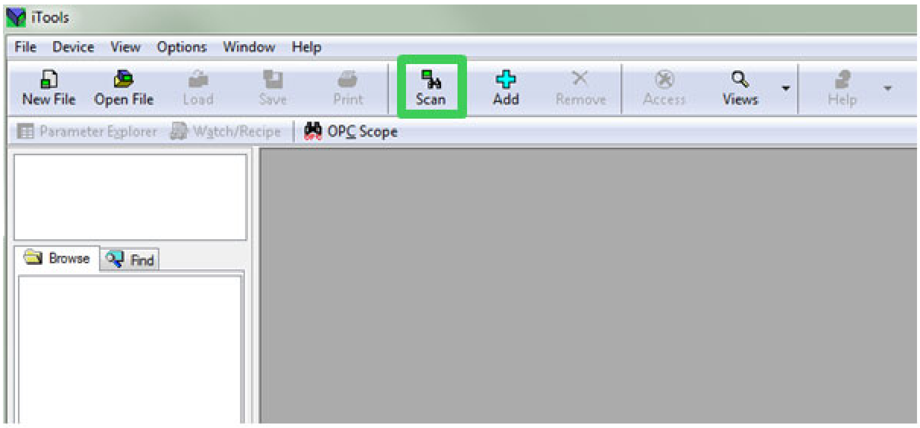
Settings may be overidden by an Address Space File. See OPC Server Startup tab.

Enabled	Connection Type	Host Name	Remote P
□ nanodac	MODBUS TCP	192.168.111.222	502
□ EPC_Panel1	MODBUS TCP	192.168.111.74	502
□ EPC3000	MODBUS TCP	192.168.111.2	502
☑ EPC2000	MODBUS TCP	192.168.111.222	502

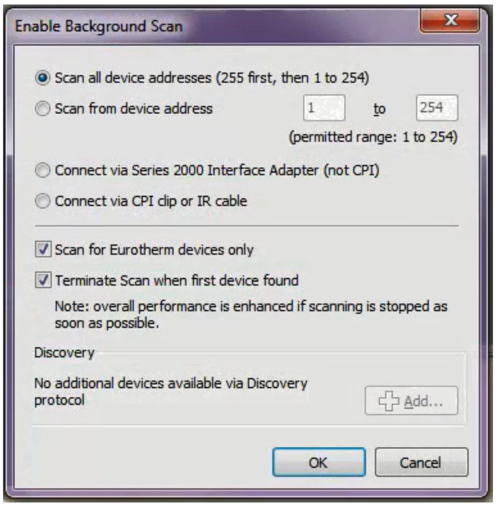
iTools is now ready to communicate with an instrument at the Host Name/IP Address configured.

iTools: Scan and Connecting to a device

9. Open iTools and press Scan.



The Enable Background Scan panel appears.



Enable Background Scan

● Scan all device addresses (255 first, then 1 to 254)

○ Scan from device address 1 to 254

(permitted range: 1 to 254)

○ Connect via Series 2000 Interface Adapter (not CPI)

○ Connect via CPI clip or IR cable

☑ Scan for Eurotherm devices only

☑ Terminate Scan when first device found

Note: overall performance is enhanced if scanning is stopped as soon as possible.

Discovery

No additional devices available via Discovery protocol

10. If not selected, select Scan all device addresses (255 first, then 1 to 254) option on the Enable background Scan panel, then check the following checkbox options:

    - Scan for Eurotherm devices only
    - Terminate Scan when first device found

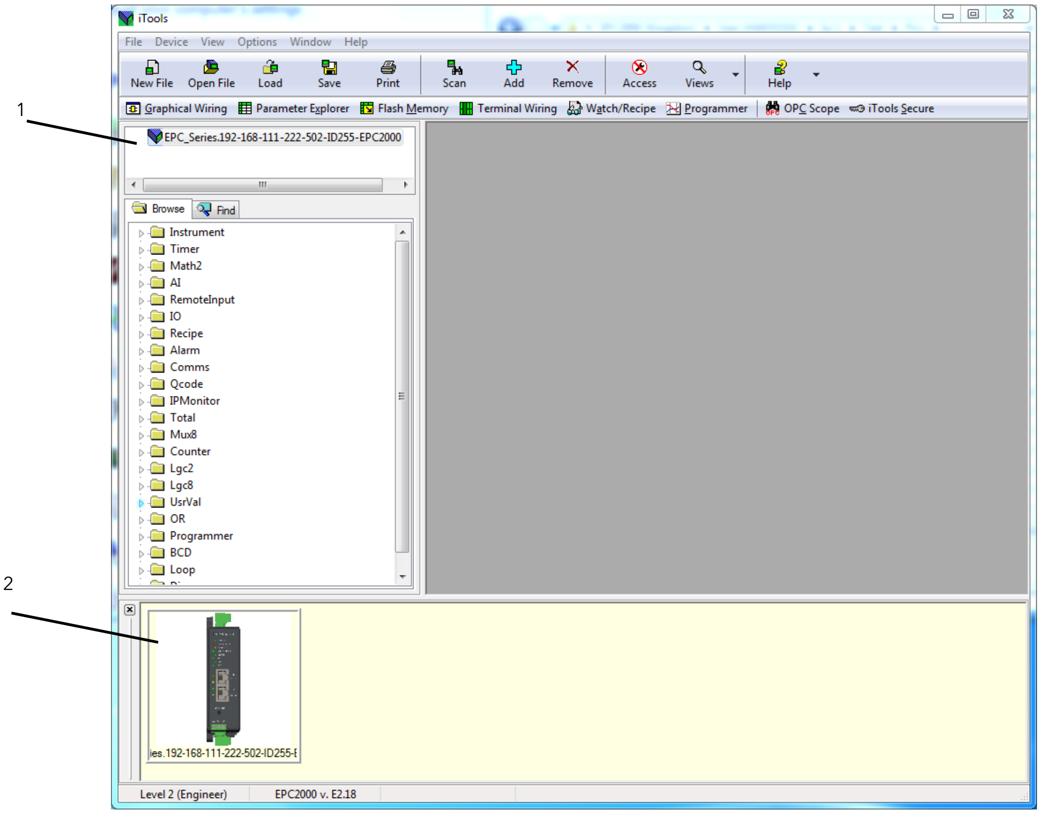
11. Click OK to start the iTools scan.

    The scan will only find devices if they have been added to the iTools Control Panel (and if they are in the same range as the IP address of the PC). See "Adding a Device to iTools Control panel" on page 234, for further details.

    Note: If Comms Configuration password is not set then iTools will ask the user to set the password before iTools connects to the instrument in configuration mode.

The EPC2000 device connects and the following items appears in the iTools windows:

- the device name and number, in top left window (1)
- an image in view panel window (2)



EtherNet/IP

EtherNet/IP Adapter (slave) is available in firmware versions V3.xx and above. The controller has been conformance tested to CT15.

EtherNet/IP (Ethernet/Industrial Protocol) is a ‘producer-consumer’ communication system used to allow industrial devices to exchange time-critical data. Such devices range from simple I/O devices such as sensors/actuators, to complex control devices such as robots and PLCs. The producer-consumer model allows the exchange of information between a single sending device (producer) and a large number of receiving devices (consumers) without having to send data multiple times to multiple destinations.

EtherNet/IP makes use of the CIP (Common Industrial Protocol), common network, transport and application layers currently implemented by DeviceNet and ControlNet. Standard Ethernet and TCP/IP technology is used to transport CIP communications packets. The result is a common, open application layer on top of Ethernet and TCP/IP protocols. With EtherNet/IP option enabled, an EPC2000 controller can act as an EtherNet/IP Adapter (slave) in an EtherNet/IP configured installation. This is a chargeable feature protected by Feature Security.

Note: An EPC2000 controller is NOT available as an EtherNet/IP Scanner client (master).

EPC2000 controllers, in common with other Eurotherm controllers, have a large number of potential parameters available, but practical systems are constrained by the total I/O space available in the EtherNet/IP Scanner client (master) being used and by the amount of traffic permissible on the network. The EPC2000 controller implicit IO exchange communications will be limited to a maximum of 64 configurable input and 64 configurable output parameters. A Fieldbus IO Gateway tool is provided within iTools software to configure the IO exchange parameters.

The EPC2000 controller EtherNet/IP adapter has been conformance tested and certified by ODVA with Declaration of Conformity (DOC) File #11868.01. It is able to communicate to a variety of ODVA approved EtherNet/IP Scanners.

EPC2000 Controller EtherNet/IP Features

The EtherNet/IP implementation features include:

- 10/100Mbit, full / half duplex operation: auto sensing
- A selectable software option, at configuration
- 3x Implicit IO messaging connections available
- 6x Explicit messaging connections available

CIP Object Support

Class (hex)	Name
01	Identity Object
02	Message Router Object
04	Assembly Object (64 inputs / 64 outputs <=> EPC2000 Fieldbus I/O Gateway)
06	Connection Manager Object
F5	TCP/IP Interface Object
F6	Ethernet Link Object
44	Modbus object

Setting Up The EtherNet/IP Scanner

This section is included for guidance only and you should refer to the instructions provided by the manufacturer of the Scanner. The EtherNet/IP Scanner used in the following example is a CompactLogix L23E QB1B PLC from Allen Bradley.

Prerequisites

The following pre-requisites must be satisfied:

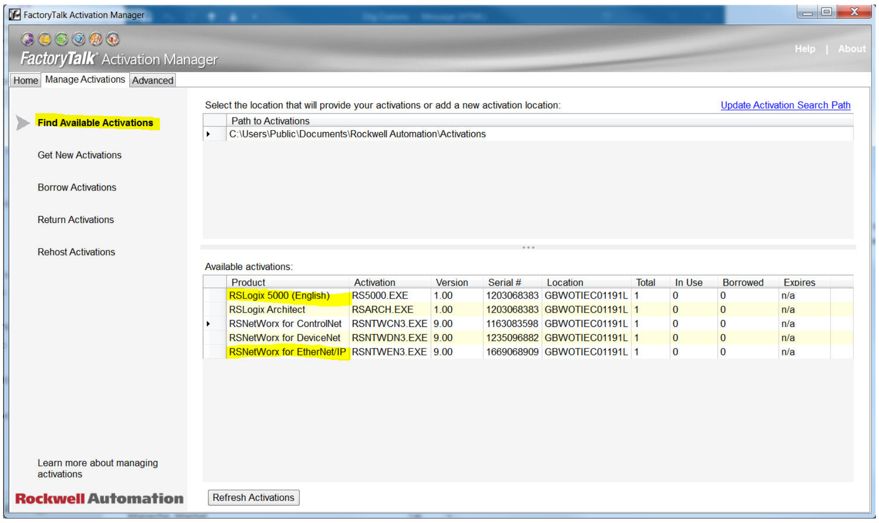
1. FactoryTalk Activation Manager, RSLinx Classic and RSLogix 5000 software must be installed on your PC.
2. Connect an Allen Bradley CompactLogix L23E to the PC via serial port.
3. Connect PC, Allen Bradley CompactLogix L23E and EPC2000 controller on the same local Ethernet network using a hub or a switch.
4. Configure PC and EPC2000 controller to be on the same subnet.
5. Power up CompactLogix L23E with key set to PROG.

Checking Software Licenses

To check software licenses, proceed as follows:

1. Click on Start/All Programs/Rockwell Software/FactoryTalk Activation/FactoryTalk Activation Manager (needs Internet connection to verify activation). The FactoryTalk Activation Manager window opens.

2. Click “Find Available Activations” and make sure that licenses for RSLogix 5000 and RSNetWorx for EtherNet/IP are present in the Available Activations table.



Select the location that will provide your activations or add a new activation location:

Path to Activations: C:\Users\Public\Documents\Rockwell Automation\Activations

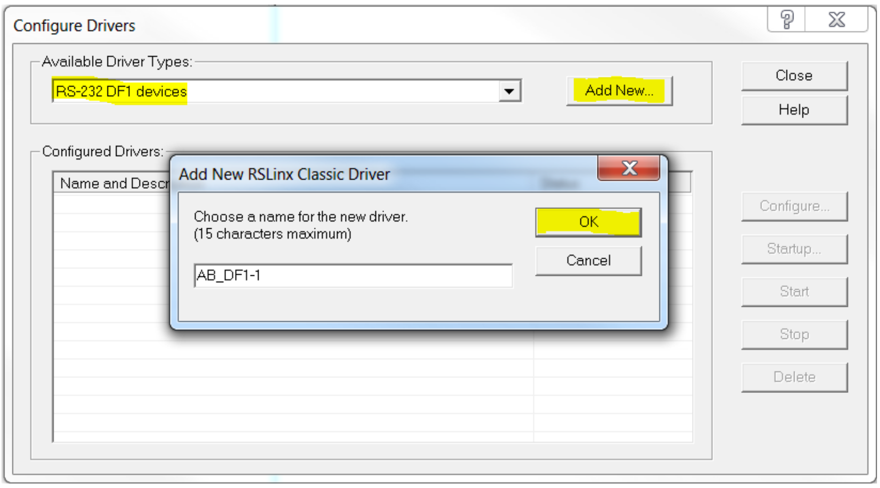
Available activations:

Product	Activation	Version	Serial #	Location	Total	In Use	Borrowed	Expires
RSLogix 5000 (English)	RS5000.EXE	1.00	1203068383	GBWOTIEC01191L	1	0	0	n/a
RSLogix Architect	RSARCH.EXE	1.00	1203068383	GBWOTIEC01191L	1	0	0	n/a
RSNetWorx for ControlNet	RSNTWCN3.EXE	9.00	1163083598	GBWOTIEC01191L	1	0	0	n/a
RSNetWorx for DeviceNet	RSNTWDN3.EXE	9.00	1235096882	GBWOTIEC01191L	1	0	0	n/a
RSNetWorx for EtherNet/IP	RSNTWEN3.EXE	9.00	1669068909	GBWOTIEC01191L	1	0	0	n/a

Configuration of PC Interfaces

1. Click on Start/All Programs/Rockwell Software/RSLinx/RSLinx Classic. The ‘RSLinx Classic’ window opens.

2. Click on ‘Communications’ and select ‘Configure Drivers’. When the ‘Configure Drivers’ window opens, select ‘RS-232 DF1 devices’ in the ‘Available Drive Types’ pull down menu and click “Add New”.



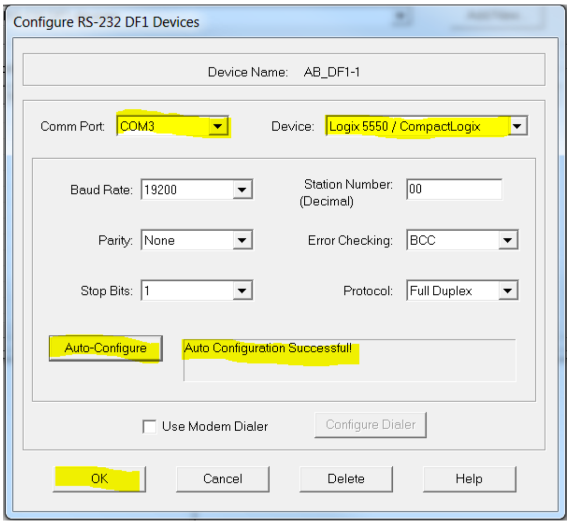
Available Driver Types: RS-232 DF1 devices

Choose a name for the new driver.
(15 characters maximum)

AB_DF1-1

3. Click OK.

4. Select PC Comm Port connection and the EtherNet/IP Scanner device connected to the port and the click Auto-Configure. Make sure that auto-configuration was successful then click OK.



Configure RS-232 DF1 Devices

Device Name: AB_DF1-1

Comm Port: COM3

Device: Logix 5550 / CompactLogix

Baud Rate: 19200

Station Number: (Decimal) 00

Parity: None

Error Checking: BCC

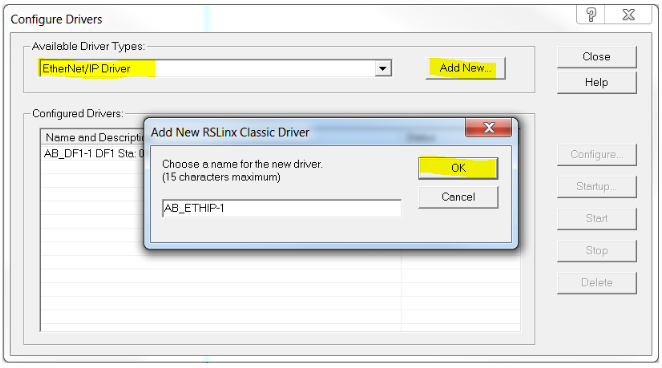
Stop Bits: 1

Protocol: Full Duplex

Auto Configuration Successful!

☐ Use Modem Dialer

5. Select ‘EtherNet/IP driver’ in the ‘Available Drive Types’ pull-down menu and click “Add New”.



Configure Drivers

Available Driver Types: EtherNet/IP Driver

Configured Drivers:

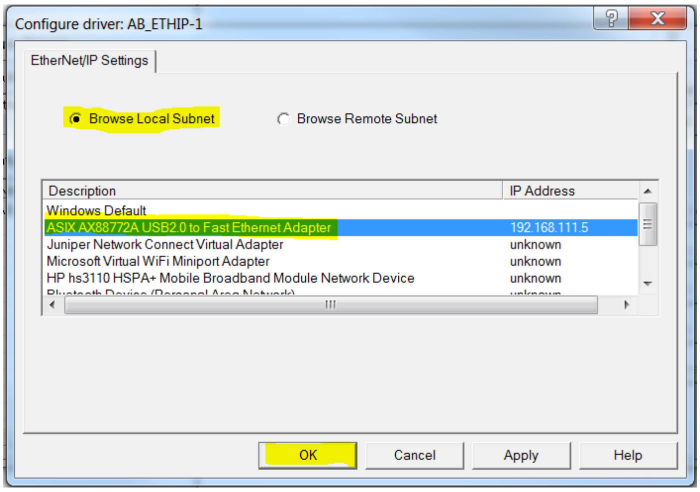
Name and Descripti
AB_DF1-1 DF1 Sta: 0

Add New RSLinx Classic Driver

Choose a name for the new driver.
(15 characters maximum)

AB_ETHIP-1

6. Select “Browse Local Subnet” and select the local PC network card to be used to connect to the EtherNet/IP network then click OK.



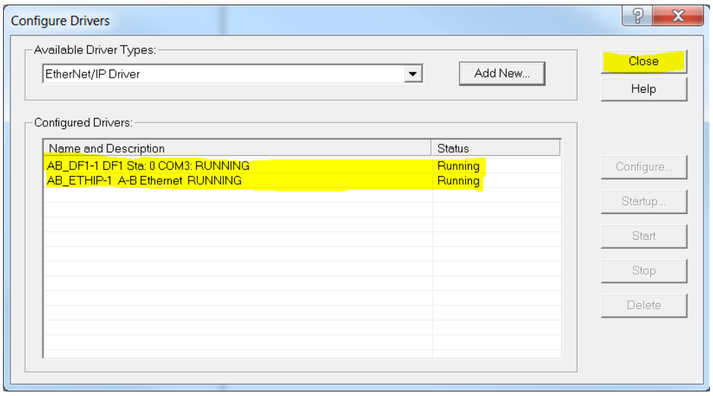
Configure driver: AB_ETHIP-1

EtherNet/IP Settings

● Browse Local Subnet ○ Browse Remote Subnet

Description	IP Address
Windows Default	
ASIX AX88772A USB2.0 to Fast Ethernet Adapter	192.168.111.5
Juniper Network Connect Virtual Adapter	unknown
Microsoft Virtual WiFi Miniport Adapter	unknown
HP hs3110 HSPA+ Mobile Broadband Module Network Device	unknown
Bluetooth Device (Personal Area Network)	unknown

7. PC Serial and EtherNet/IP drivers must now be running. Minimize window.



Configure Drivers

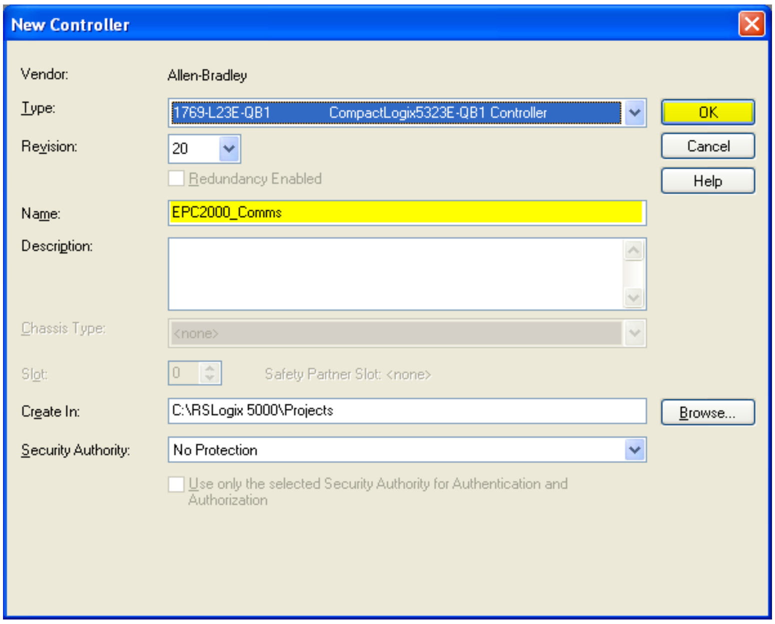
Available Driver Types: EtherNet/IP Driver

Name and Description	Status
AB_DF1-1 DF1 Sta: 0 COM3: RUNNING	Running
AB_ETHIP-1 A-B Ethernet RUNNING	Running

Configuration of RSLOGIX 5000 Application

The following describes the configuration of the Compactlogix L23E EtherNet/IP Scanner network settings using the RXLogix 5000 software:

1. Start the RSLogix 5000 program (from ‘Start/All programs/... /RSLogix 5000). When the ‘Quick Start’ window opens, close it.
2. In the ‘File’ menu select ‘New’, or click the ‘New Tool’ icon. The ‘New Controller’ window opens.
3. Select the relevant PLC from the drop-down menu. Enter a name for the configuration and click ‘OK’. After a few seconds, the window opens for the selected controller.



New Controller

Vendor: Allen-Bradley

Type: 1769-L23E-QB1 CompactLogix5323E-QB1 Controller

Revision: 20

□ Redundancy Enabled

Name: EPC2000_Comms

Description:

Chassis Type: <none>

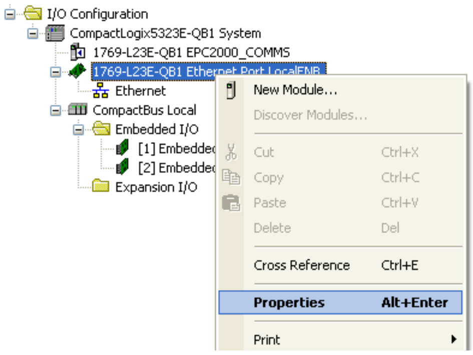
Slot: 0 Safety Partner Slot: <none>

Create In: C:\RSLogix 5000\Projects

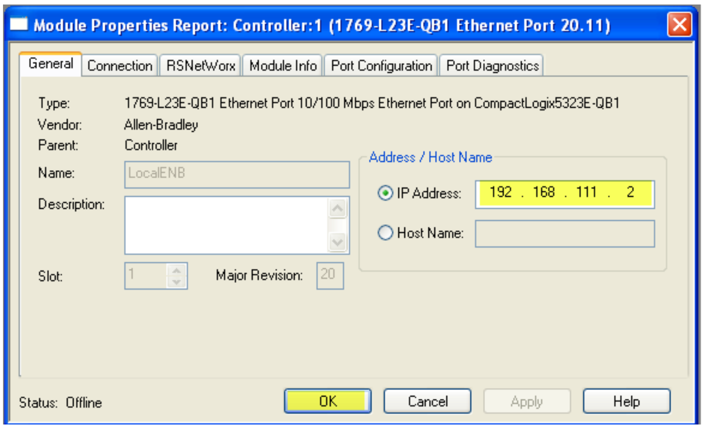
Security Authority: No Protection

□ Use only the selected Security Authority for Authentication and Authorization

4. Configure the Ethernet port settings of the CompactLogix L23E by right-clicking on the relevant Ethernet port in the left pane ‘tree’, and select ‘Properties’.



5. In the Module Properties window, configure the IP address and click OK.


Module Properties Report: Controller:1 (1769-L23E-QB1 Ethernet Port 20.11)

General | Connection | RSNetWorx | Module Info | Port Configuration | Port Diagnostics

Type: 1769-L23E-QB1 Ethernet Port 10/100 Mbps Ethernet Port on CompactLogix5323E-QB1

Vendor: Allen-Bradley

Parent: Controller

Name: LocalENB

Description:

Slot: 1

Major Revision: 20

Address / Host Name

● IP Address: 192 . 168 . 111 . 2

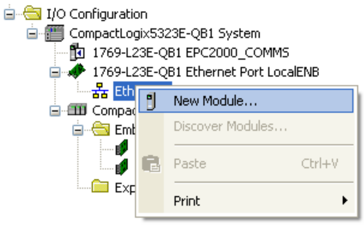
○ Host Name:

Status: Offline

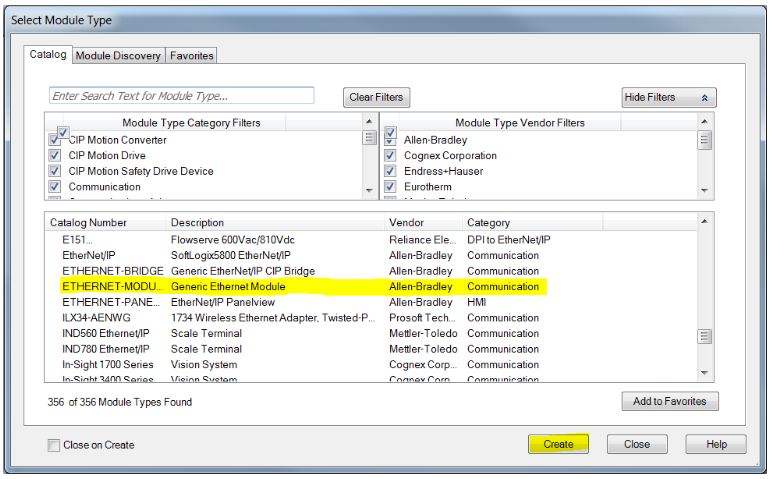
Configure Scanner Connection Settings to the EPC2000 Controller EtherNet/IP Adapter

Method 1 (without EDS file)

1. First, configure the EPC2000 Adapter by creating a new module under the CompactLogix L23E Ethernet node.


I/O Configuration
CompactLogix5323E-QB1 System
1769-L23E-QB1 EPC2000_COMMS
1769-L23E-QB1 Ethernet Port LocalENB
Eth
Compac
Emb
Exp
New Module...
Discover Modules...
Paste Ctrl+V
Print

2. Select “Generic Ethernet Module” as the module type and click Create.



Module Type Category Filters
☑ CIP Motion Converter
☑ CIP Motion Drive
☑ CIP Motion Safety Drive Device
☑ Communication

Module Type Vendor Filters
☑ Allen-Bradley
☑ Cognex Corporation
☑ Endress+Hauser
☑ Eurotherm

Catalog Number	Description	Vendor	Category
E151...	Flowserve 600Vac/810Vdc	Reliance Ele...	DPI to EtherNet/IP
EtherNet/IP	SoftLogix5800 EtherNet/IP	Allen-Bradley	Communication
ETHERNET-BRIDGE	Generic EtherNet/IP CIP Bridge	Allen-Bradley	Communication
ETHERNET-MODU...	Generic Ethernet Module	Allen-Bradley	Communication
ETHERNET-PANE...	EtherNet/IP Panelview	Allen-Bradley	HMI
ILX34-AENWG	1734 Wireless Ethernet Adapter, Twisted-P...	Prosoft Tech...	Communication
IND560 Ethernet/IP	Scale Terminal	Mettler-Toledo	Communication
IND780 Ethernet/IP	Scale Terminal	Mettler-Toledo	Communication
In-Sight 1700 Series	Vision System	Cognex Corp...	Communication
In-Sight 3400 Series	Vision System	Cognex Corp...	Communication

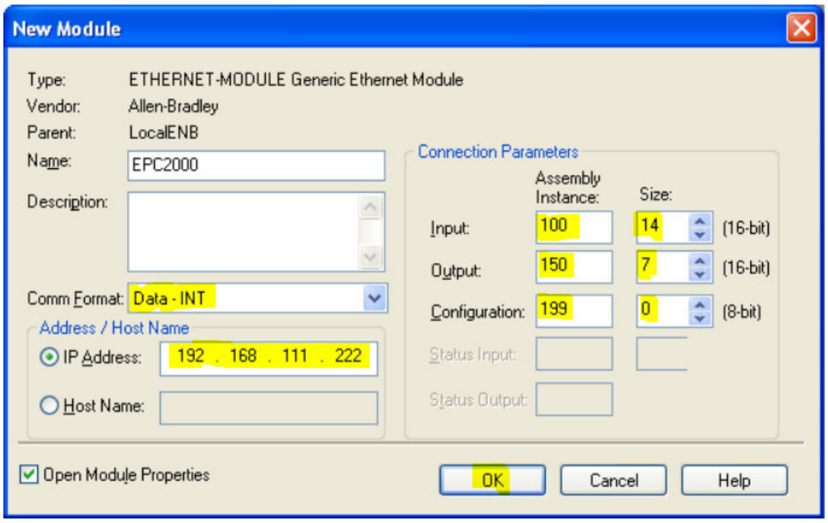
356 of 356 Module Types Found

□ Close on Create

3. Complete the Module properties with the EPC2000 Adapter settings then click OK.

Comm Format	(Data - INT)
IP Address	(xxx.xxx.xxx.xxx)

Description	Assembly Instance	Size
Input	100	14 x 16-bit (EPC2000 default)
Output	150	7 x 16-bit (EPC2000 default)
Configuration	199	0 (EPC2000 default)



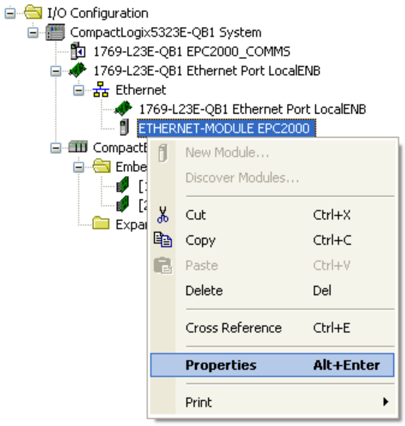
Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: LocalENB
Name: EPC2000
Description:
Comm Format: Data - INT
Address / Host Name
● IP Address: 192 . 168 . 111 . 222
○ Host Name:

Connection Parameters

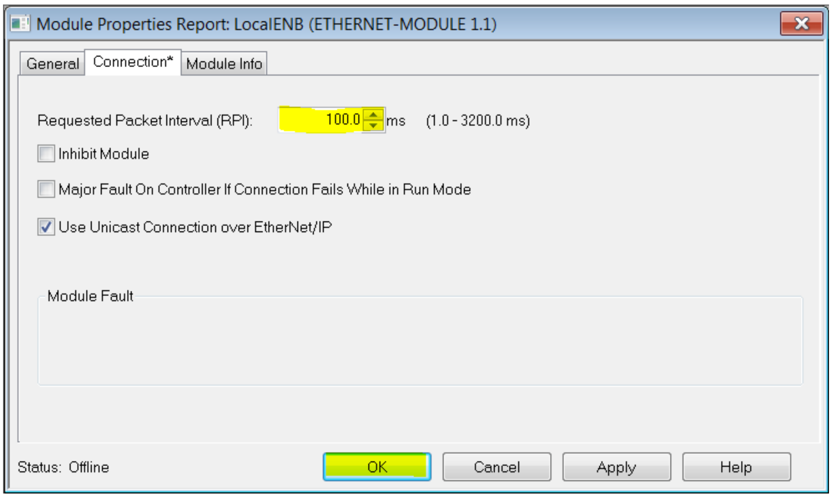
	Assembly Instance:	Size:	
Input:	100	14	(16-bit)
Output:	150	7	(16-bit)
Configuration:	199	0	(8-bit)
Status Input:			
Status Output:			

☑ Open Module Properties

4. Configure connection properties of the newly created module by right-clicking it and selecting “Properties”.



5. Set Requested Packet Interval (RPI) using Module Properties ‘Connection’ tab making sure that it is within 50 to 3200ms then click OK.


Module Properties Report: LocalENB (ETHERNET-MODULE 1.1)

Requested Packet Interval (RPI): 100.0 ms (1.0 - 3200.0 ms)

□ Inhibit Module

□ Major Fault On Controller If Connection Fails While in Run Mode

☑ Use Unicast Connection over EtherNet/IP

Module Fault

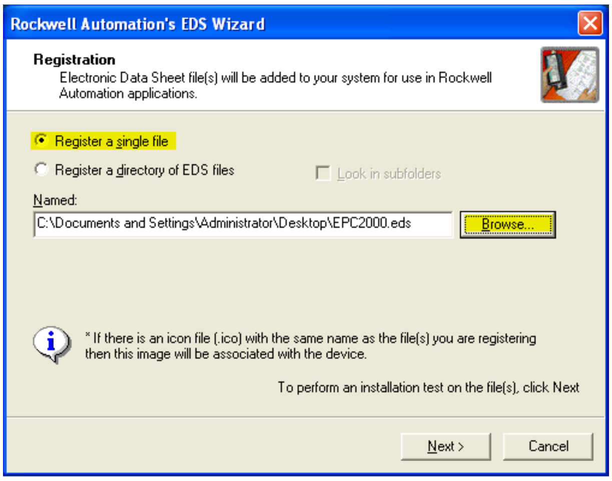
Status: Offline

Method 2 (with EDS file)

EPC2000 EDS Installation

1. Click on Start/All Programs/Rockwell software/RSLinx/Tools/EDS Hardware Installation Tool. The ‘EDS Hardware Installation Tool’ window opens.

2. Click Add to open the EDS Wizard window and then select the ‘Register a single file’ radio button. Browse to the EPC2000 EDS file and then click Next.



Rockwell Automation's EDS Wizard

Registration

Electronic Data Sheet file(s) will be added to your system for use in Rockwell Automation applications.

● Register a single file

○ Register a directory of EDS files □ Look in subfolders

Named: C:\Documents and Settings\Administrator\Desktop\EPC2000.eds

* If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.

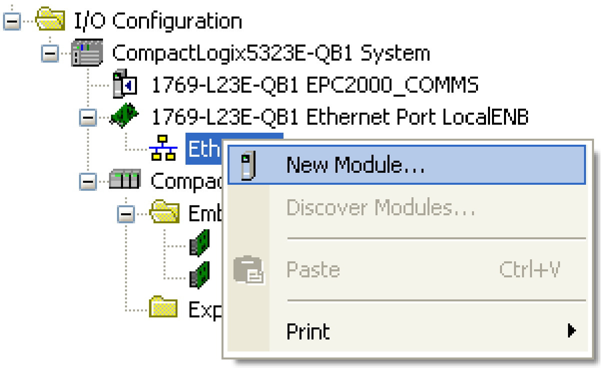
To perform an installation test on the file(s), click Next

3. Click Next on the next three windows and then click Finish on the final window.

Configure Scanner Connection Settings to the EPC2000 Adapter

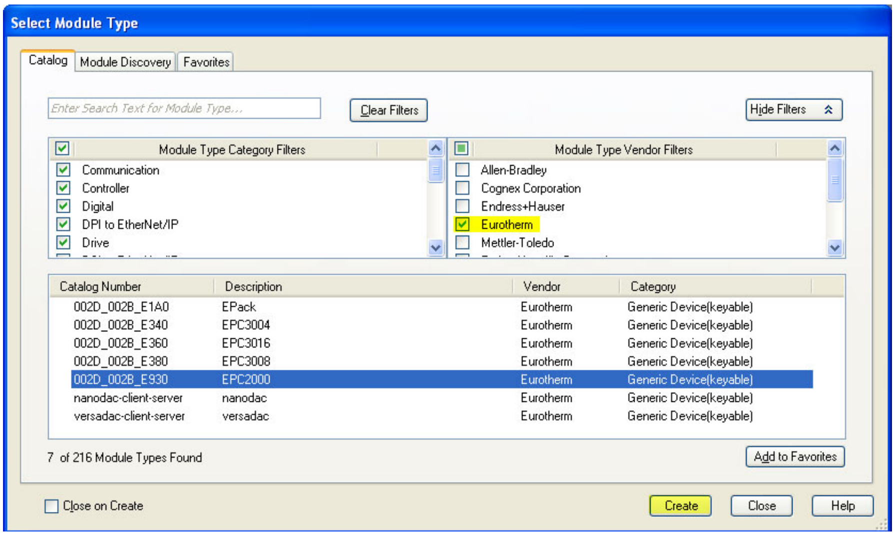
In the RSLogix 5000 Scanner Program, configure EPC2000 Adapter connection settings by creating a new module under the CompactLogix L23E Ethernet node.

1. Right click the Ethernet node and select ‘New Module’ from context menu. On the pop-up window ‘Select Module Type’. Click Show Filters.



I/O Configuration
CompactLogix5323E-QB1 System
1769-L23E-QB1 EPC2000_COMMS
1769-L23E-QB1 Ethernet Port LocalENB
Eth
Compac
Emb
Exp
New Module...
Discover Modules...
Paste Ctrl+V
Print

2. Filter for Eurotherm devices and then select the EPC2000 device module required (module installed in the previous section via EDS file) and then click Create.



Select Module Type

☑ Module Type Category Filters	□ Module Type Vendor Filters
☑ Communication	□ Allen-Bradley
☑ Controller	□ Cognex Corporation
☑ Digital	□ Endress+Hauser
☑ DPI to EtherNet/IP	☑ Eurotherm
☑ Drive	□ Mettler-Toledo

Catalog Number	Description	Vendor	Category
002D_002B_E1A0	EPack	Eurotherm	Generic Device(keyable)
002D_002B_E340	EPC3004	Eurotherm	Generic Device(keyable)
002D_002B_E360	EPC3016	Eurotherm	Generic Device(keyable)
002D_002B_E380	EPC3008	Eurotherm	Generic Device(keyable)
002D_002B_E930	EPC2000	Eurotherm	Generic Device(keyable)
nanodac-client-server	nanodac	Eurotherm	Generic Device(keyable)
versadac-client-server	versadac	Eurotherm	Generic Device(keyable)

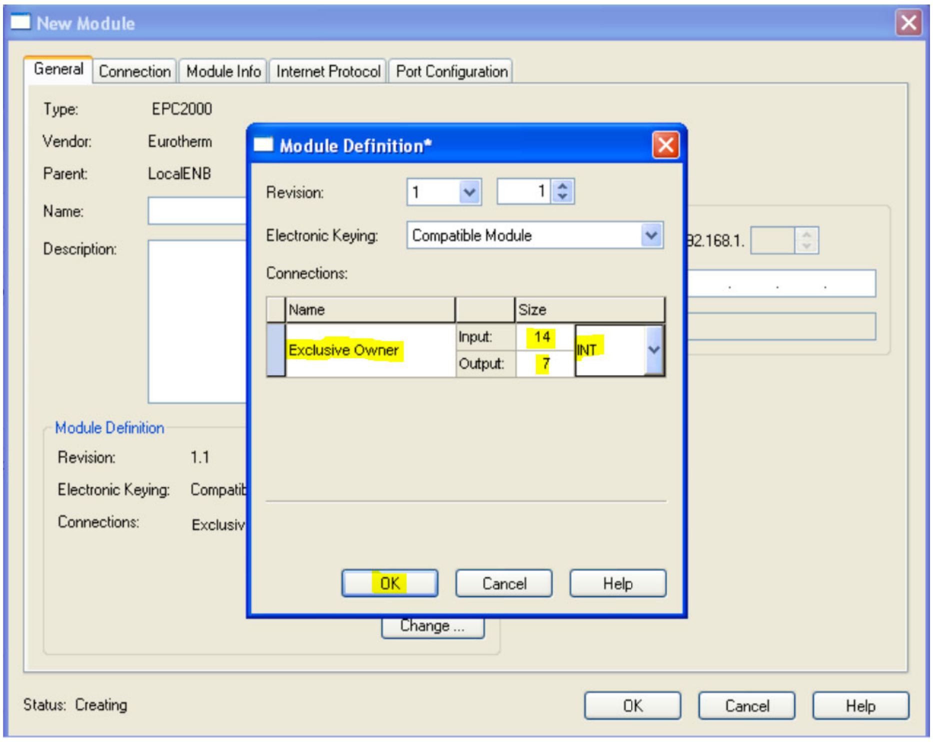
7 of 216 Module Types Found

□ Close on Create

3. A ‘New Module’ window will pop up. Click Change to configure the:

Connection type:	Exclusive Owner / Input Only / Listen Only
Input Size:	Default length of EPC2000 inputs in INT (14 x 16-bit)
Output Size:	Default length of EPC2000 outputs in INT (7 x 16-bit)

Then click OK.



New Module

Type: EPC2000
Vendor: Eurotherm
Parent: LocalENB
Name:
Description:

Module Definition
Revision: 1.1
Electronic Keying: Compatib
Connections: Exclusiv

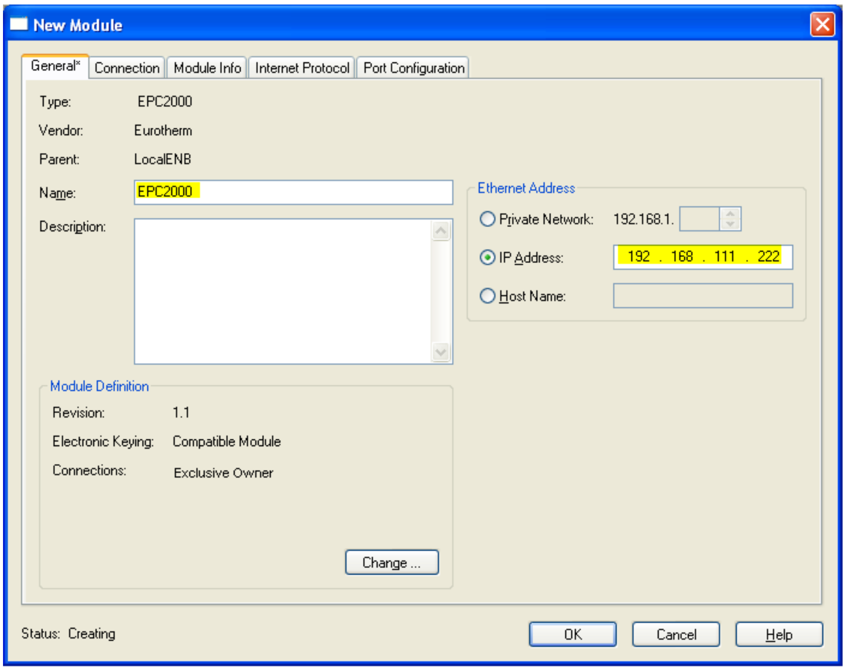
Module Definition*

Revision: 1 1
Electronic Keying: Compatible Module
Connections:

Name		Size	
Exclusive Owner	Input:	14	INT
	Output:	7	

Status: Creating

4. On the ‘New Module’ window, configure the IP address of the EPC2000 EtherNet/IP adapter. Enter a descriptive name and then click OK.



New Module

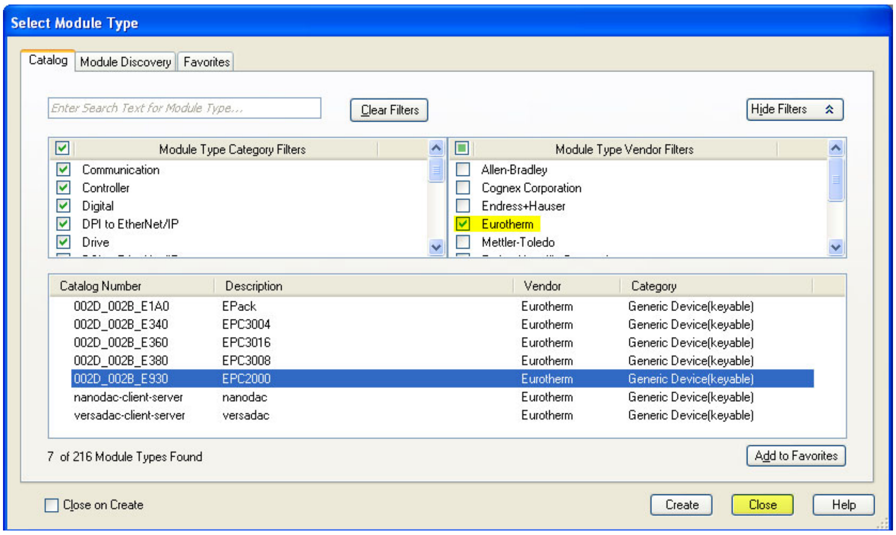
Type: EPC2000
Vendor: Eurotherm
Parent: LocalENB
Name: EPC2000
Description:

Ethernet Address
○ Private Network: 192.168.1.
● IP Address: 192 . 168 . 111 . 222
○ Host Name:

Module Definition
Revision: 1.1
Electronic Keying: Compatible Module
Connections: Exclusive Owner

Status: Creating

5. Close the ‘Select Module Type’ window.



☑	Module Type Category Filters
☑	Communication
☑	Controller
☑	Digital
☑	DPI to EtherNet/IP
☑	Drive

■	Module Type Vendor Filters
□	Allen-Bradley
□	Cognex Corporation
□	Endress+Hauser
☑	Eurotherm
□	Mettler-Toledo

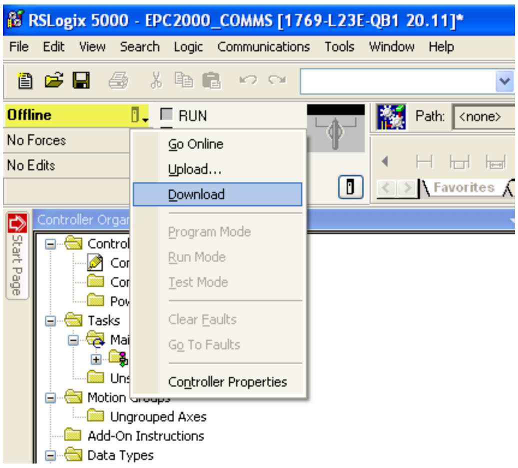
Catalog Number	Description	Vendor	Category
002D_002B_E1A0	EPack	Eurotherm	Generic Device(keyable)
002D_002B_E340	EPC3004	Eurotherm	Generic Device(keyable)
002D_002B_E360	EPC3016	Eurotherm	Generic Device(keyable)
002D_002B_E380	EPC3008	Eurotherm	Generic Device(keyable)
002D_002B_E930	EPC2000	Eurotherm	Generic Device(keyable)
nanodac-client-server	nanodac	Eurotherm	Generic Device(keyable)
versadac-client-server	versadac	Eurotherm	Generic Device(keyable)

7 of 216 Module Types Found

□ Close on Create

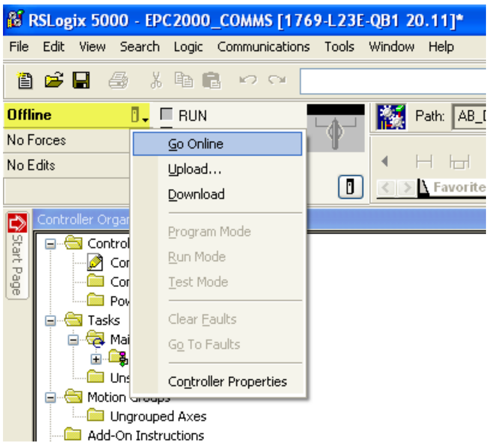
Downloading and Running the RSLOGIX 5000 Application to the Scanner

1. Make sure that the Mode key of the CompactLogix hardware is set to ‘PROG’ and start download by clicking on the drop-down Offline menu and selecting ‘Download’.



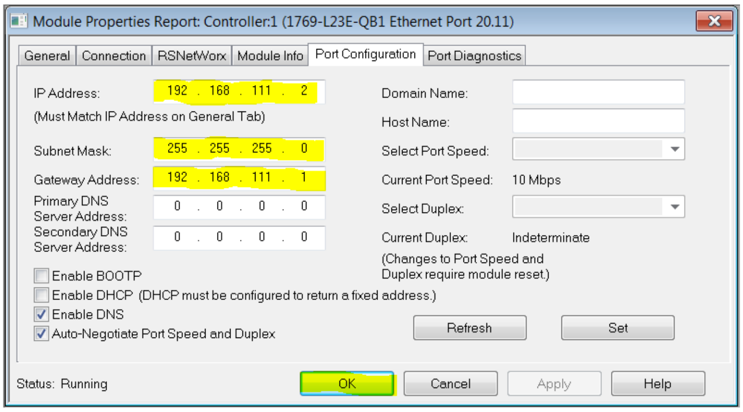
RSLogix 5000 - EPC2000_COMMS [1769-L23E-QB1 20.11]*

2. Go Online to the CompactLogix L23E by clicking on the drop-down Offline menu and selecting ‘Go Online’.



If there is a problem with the path, use RSLogix 5000>Communications > Who Active, select AB_DF1 and select ‘Download’.

3. Select Port Configuration tab and configure CompactLogix L23E port settings making sure that there is no duplication of IP address and it is in the same subnet as the PC and EPC2000. Then click OK.


Module Properties Report: Controller:1 (1769-L23E-QB1 Ethernet Port 20.11)

General | Connection | RSNetWorx | Module Info | Port Configuration | Port Diagnostics

IP Address: 192 . 168 . 111 . 2
(Must Match IP Address on General Tab)

Subnet Mask: 255 . 255 . 255 . 0

Gateway Address: 192 . 168 . 111 . 1

Primary DNS Server Address: 0 . 0 . 0 . 0

Secondary DNS Server Address: 0 . 0 . 0 . 0

Domain Name:

Host Name:

Select Port Speed:

Current Port Speed: 10 Mbps

Select Duplex:

Current Duplex: Indeterminate

(Changes to Port Speed and Duplex require module reset.)

☐ Enable BOOTP
☐ Enable DHCP (DHCP must be configured to return a fixed address.)
☑ Enable DNS
☑ Auto-Negotiate Port Speed and Duplex

Status: Running

4. Switch the CompactLogix L23E mode key to ‘RUN’ and the CompactLogix L23E EtherNet/IP scanner should start connecting to the EPC2000 EtherNet/IP immediately.

Establishing Communications

EtherNet/IP I/O messaging will commence when the EtherNet/IP network is correctly cabled and powered, the EtherNet/IP Scanner and Adapter (EPC2000 controller) are configured with valid and unique IP addresses in the same subnet and the I/O parameter data definitions are setup correctly.

The EPC2000 Input/Output definitions need to be matched with the EtherNet/IP Scanner (for example, PLC) data registers.

Parameters are either INPUT parameters read by the EtherNet/IP Scanner or OUTPUT parameters written by the EtherNet/IP scanner.

Note: EPC2000 V4.xx onwards, comms configuration password will need to be set in order for EtherNet/IP to work.

Data Formats

16-bit data read from the EPC2000 controller EtherNet/IP are ‘scaled integers’ and the value will depend on the resolution of the parameter being read. A 32-bit float value of 12.34 with resolution 2 will be encoded as 1234 while, if the resolution is changed to 1, it will be encoded as 123.

32-bit float and 32-bit time integers can also be written to, and read from, the EPC2000 using I/O exchange when the same parameter is configured in consecutive rows in the Fieldbus I/O Gateway definition table. 32-bit values can also be written to, and read from, the EPC2000 using Explicit messaging via Modbus object when writing to, or reading from, the EPC2000’s IEEE region (Modbus address > 0x8000).

The EDS File

The EtherNet/IP EDS (Electronic Data Sheet) files for EPC2000 Programmable Controller are available by going to web site www.eurotherm.com or from your supplier.

The EDS file is designed to automate the EtherNet/IP network configuration process by defining the required device parameter information. Software configuration tools utilize the EDS file to configure an EtherNet/IP network.

Note: Selected parameters may be configured to exchange Input and Output data across a network. These may be configured using iTools.

Troubleshooting

No Communications:

- Check the cabling carefully and ensure that the RJ45 connectors are fully plugged into the sockets.
- Confirm that EtherNet/IP is available and enabled in the EPC2000 controller by setting the Comms>Option>Main>Protocol to EipAndModTCP(12) in iTools.
- Check that the EPC2000 controller network settings, IP Address, Subnet Mask and Gateway in the 'Comms' list are correct and unique for the network configuration in use, and that the EPC2000 controller and the EtherNet/IP Scanner (master) are in the same subnet.
- Ensure that the configured EtherNet/IP Scanner Input and Output data length matches the data length of the EPC2000 Adapter Input and Output definitions configured using the Fieldbus I/O Gateway Editor. If the master is attempting to read (input) or write (output) more or less data than has been registered on the EPC2000 Adapter, using the iTools Fieldbus I/O Gateway Editor, the EPC2000 controller Adapter will refuse the connection.

Modbus Master

Overview

Modbus Master is available over Ethernet communications (Modbus TCP). It is mutually exclusive with EtherNet/IP but will available in conjunction with Modbus TCP slave.

Modbus TCP Master is protected by Feature Security.

Communication profiles for Eurotherm Products EPCx (EPC3000 & EPC2000 generic), EPack, and EPower devices are supported. A maximum of three Modbus TCP slave devices can be configured with timeouts and retries configurable per slave.

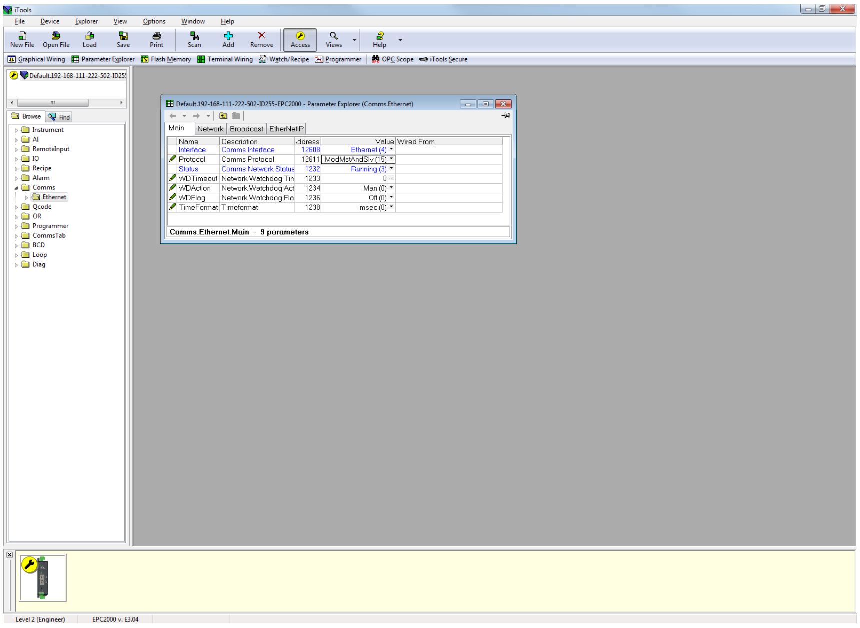
A maximum of 32 data points are supported to be shared among the three slave devices. These data points can be configured for writing to or reading from a Modbus slave.

Note: EPC2000 V4.xx onwards, comms configuration password will need to be set in order for Modbus Master to work.

Setting Modbus Master Protocol

Proceed as follows:

1. Connect iTools to the instrument via an Ethernet cable.
2. From iTools, place the instrument in Configuration mode.
3. If Ethernet Option module is available on the instrument and Modbus Master feature is enabled, select Modbus Master and slave protocol for Ethernet comms interface.


iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Browse Find
Instrument AI RemoteInput IO Recipe Alarm Comms Ethernet Qcode OR Programmer CommsTab BCD Loop Diag
Level 2 (Engineer) EPC2000 v. E3.04

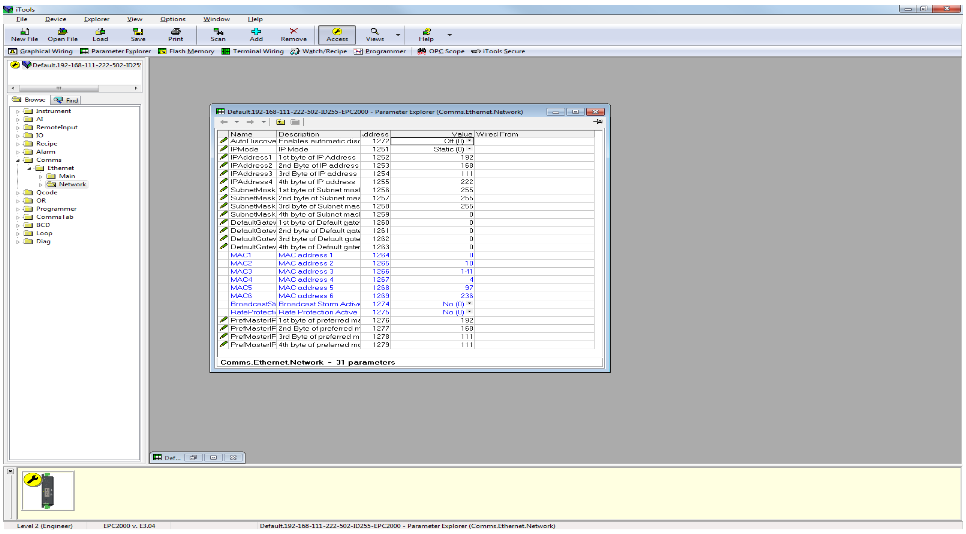
Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Ethernet)

Main | Network | Broadcast | EtherNetIP

Name	Description	ddress	Value	Wired From
Interface	Comms Interface	12608	Ethernet (4)	
Protocol	Comms Protocol	12611	ModMstAndSlv (15)	
Status	Comms Network Status	1232	Running (3)	
WDTimeout	Network Watchdog Tim	1233	0	
WDAction	Network Watchdog Act	1234	Man (0)	
WDFlag	Network Watchdog Fla	1236	Off (0)	
TimeFormat	Timeformat	1238	msec (0)	

Comms.Ethernet.Main - 9 parameters

Note: The Network configuration of the Modbus TCP Master can be found in the Comms function block Network tab. Confirm that the IP address and subnet mask are configured correctly to be able to communicate with Modbus slave devices within the subnet. If the slave device is outside the subnet then the Default gateway must be configured correctly.



Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Ethernet.Network)

Name	Description	ddress	Value	Wired From
AutoDiscove	Enables automatic disc	1272	Off (0)	
IPMode	IP Mode	1251	Static (0)	
IPAddress1	1st byte of IP Address	1252	192	
IPAddress2	2nd Byte of IP address	1253	168	
IPAddress3	3rd Byte of IP address	1254	111	
IPAddress4	4th byte of IP address	1255	222	
SubnetMask	1st byte of Subnet masl	1256	255	
SubnetMask	2nd byte of Subnet mas	1257	255	
SubnetMask	3rd byte of Subnet mas	1258	255	
SubnetMask	4th byte of Subnet masl	1259	0	
DefaultGatev	1st byte of Default gate	1260	0	
DefaultGatev	2nd byte of Default gate	1261	0	
DefaultGatev	3rd byte of Default gate	1262	0	
DefaultGatev	4th byte of Default gate	1263	0	
MAC1	MAC address 1	1264	0	
MAC2	MAC address 2	1265	10	
MAC3	MAC address 3	1266	141	
MAC4	MAC address 4	1267	4	
MAC5	MAC address 5	1268	97	
MAC6	MAC address 6	1269	236	
BroadcastSt	Broadcast Storm Active	1274	No (0)	
RateProtecti	Rate Protection Active	1275	No (0)	
PrefMasterIP	1st byte of preferred ma	1276	192	
PrefMasterIP	2nd Byte of preferred m	1277	168	
PrefMasterIP	3rd Byte of preferred m	1278	111	
PrefMasterIP	4th byte of preferred ma	1279	111	

Comms.Ethernet.Network - 31 parameters

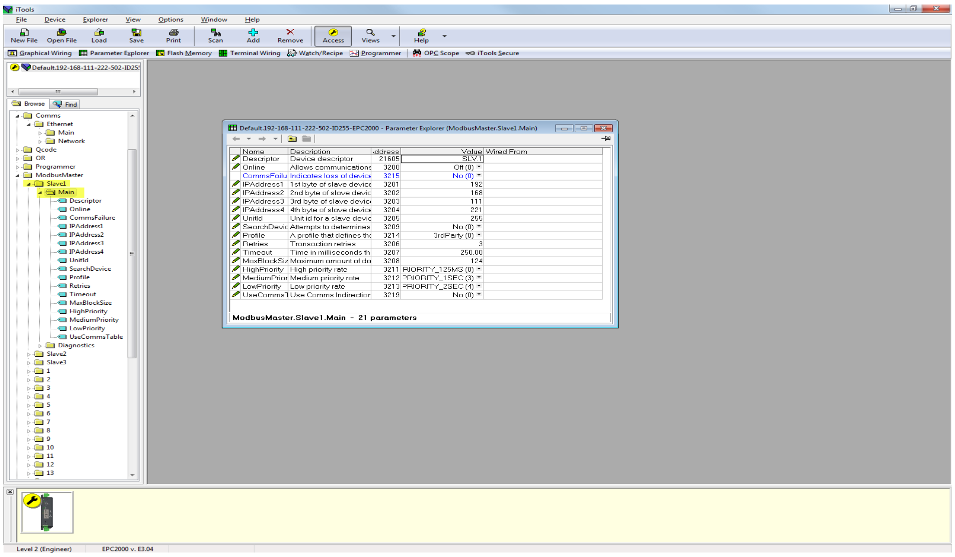
Level 2 (Engineer) EPC2000 v. E3.04 Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Comms.Ethernet.Network)

4. From iTools, place the instrument out of Config to restart the instrument and initialize new comms settings.

Configuring Communications to Modbus Slaves

To configure communications to Modbus slaves, proceed as follows:

1. From iTools, place the instrument in Config mode and open: ModbusMaster>Slave1>Main to configure the first slave.



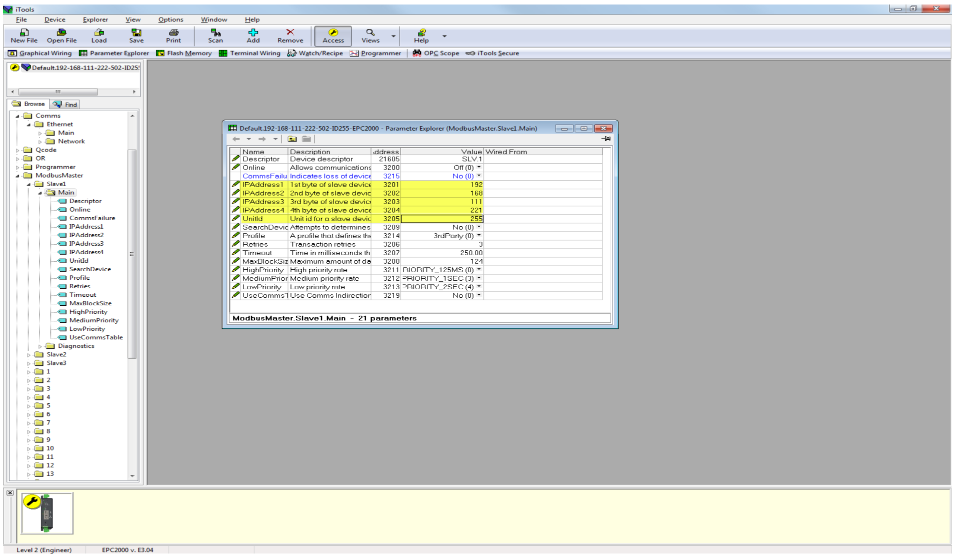
Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave1.Main)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21605	SLV.1	
Online	Allows communications	3200	Off (0)	
CommsFailu	Indicates loss of device	3215	No (0)	
IPAddress1	1st byte of slave device	3201	192	
IPAddress2	2nd byte of slave devic	3202	168	
IPAddress3	3rd byte of slave devic	3203	111	
IPAddress4	4th byte of slave device	3204	221	
UnitId	Unit id for a slave devic	3205	255	
SearchDevic	Attempts to determines	3209	No (0)	
Profile	A profile that defines the	3214	3rdParty (0)	
Retries	Transaction retries	3206	3	
Timeout	Time in milliseconds th	3207	250.00	
MaxBlockSiz	Maximum amount of da	3208	124	
HighPriority	High priority rate	3211	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3212	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3213	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3219	No (0)	

ModbusMaster.Slave1.Main - 21 parameters

Level 2 (Engineer) EPC2000 v. E3.04

2. Configure the slave’s IP Address and unit ID.

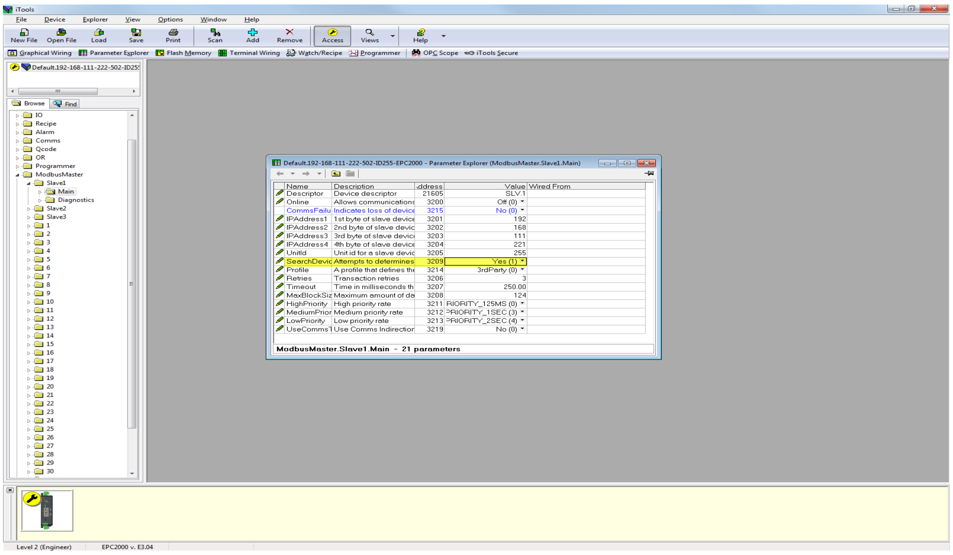

iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Browse Find
Level 2 (Engineer) EPC2000 v. E3.04

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave1.Main)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21605	SLV.1	
Online	Allows communications	3200	Off (0)	
CommsFailu	Indicates loss of device	3215	No (0)	
IPAddress1	1st byte of slave device	3201	192	
IPAddress2	2nd byte of slave devic	3202	168	
IPAddress3	3rd byte of slave devic	3203	111	
IPAddress4	4th byte of slave device	3204	221	
UnitId	Unit id for a slave devic	3205	255	
SearchDevic	Attempts to determines	3209	No (0)	
Profile	A profile that defines the	3214	3rdParty (0)	
Retries	Transaction retries	3206	3	
Timeout	Time in milliseconds th	3207	250.00	
MaxBlockSiz	Maximum amount of da	3208	124	
HighPriority	High priority rate	3211	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3212	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3213	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3219	No (0)	

ModbusMaster.Slave1.Main - 21 parameters

3. You can now check if the device is online via the ‘Search device’ parameter by setting its value to ‘Yes’. The search status should be ‘Searching(0).’

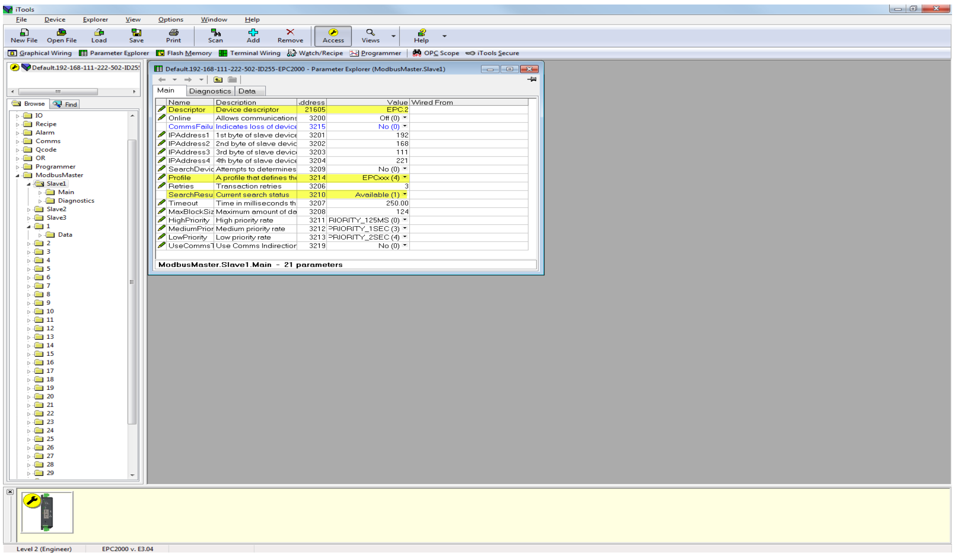

iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Browse Find
Level 2 (Engineer) EPC2000 v. E3.04

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave1.Main)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21605	SLV.1	
Online	Allows communications	3200	Off (0)	
CommsFailu	Indicates loss of device	3215	No (0)	
IPAddress1	1st byte of slave device	3201	192	
IPAddress2	2nd byte of slave devic	3202	168	
IPAddress3	3rd byte of slave devic	3203	111	
IPAddress4	4th byte of slave device	3204	221	
UnitId	Unit id for a slave devic	3205	255	
SearchDevic	Attempts to determines	3209	Yes (1)	
Profile	A profile that defines the	3214	3rdParty (0)	
Retries	Transaction retries	3206	3	
Timeout	Time in milliseconds th	3207	250.00	
MaxBlockSiz	Maximum amount of da	3208	124	
HighPriority	High priority rate	3211	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3212	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3213	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3219	No (0)	

ModbusMaster.Slave1.Main - 21 parameters

4. If the Modbus slave is online then the search result will be ‘Available(1)’, otherwise the result will be ‘Unreachable(3)’. If it is a Eurotherm instrument with a supported profile, the ‘Profile’ parameter will display the Modbus slave’s profile, otherwise it will display ‘3rdParty(0)’.

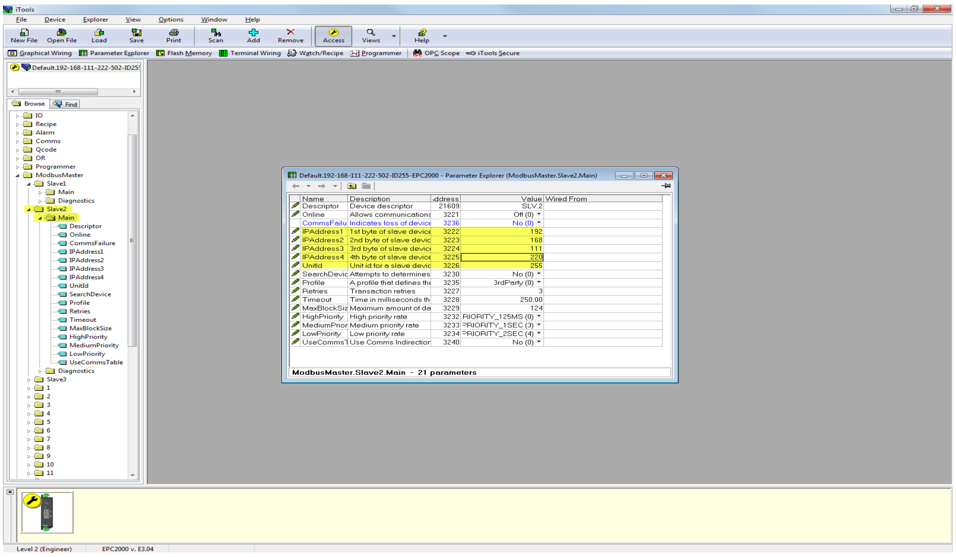


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave1)

Name	Description	Address	Value	Wired From
Descriptor	Device descriptor	21605	EPC.2	
Online	Allows communications	3200	Off (0)	
CommsFailu	Indicates loss of device	3215	No (0)	
IPAddress1	1st byte of slave device	3201	192	
IPAddress2	2nd byte of slave devic	3202	168	
IPAddress3	3rd byte of slave device	3203	111	
IPAddress4	4th byte of slave device	3204	221	
SearchDevic	Attempts to determines	3209	No (0)	
Profile	A profile that defines the	3214	EPCxxx (4)	
Retries	Transaction retries	3206	3	
SearchResu	Current search status	3210	Available (1)	
Timeout	Time in milliseconds th	3207	250.00	
MaxBlockSiz	Maximum amount of da	3208	124	
HighPriority	High priority rate	3211	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3212	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3213	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3219	No (0)	

ModbusMaster.Slave1.Main - 21 parameters

5. Configure a second slave by setting the device's IP Address and unit ID.

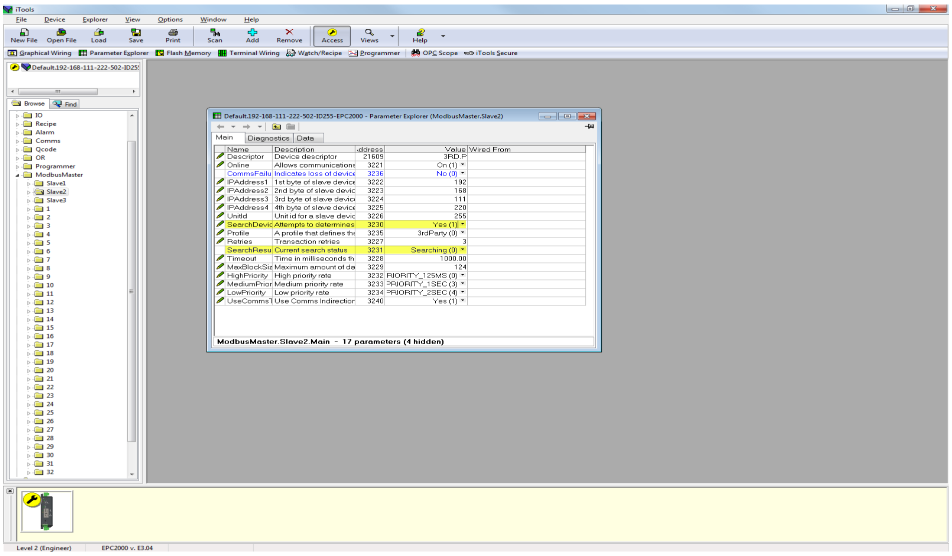


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave2.Main)

Name	Description	Address	Value	Wired From
Descriptor	Device descriptor	21609	SLV.2	
Online	Allows communications	3221	Off (0)	
CommsFailu	Indicates loss of device	3236	No (0)	
IPAddress1	1st byte of slave device	3222	192	
IPAddress2	2nd byte of slave devic	3223	168	
IPAddress3	3rd byte of slave device	3224	111	
IPAddress4	4th byte of slave device	3225	220	
UnitId	Unit id for a slave devic	3226	255	
SearchDevic	Attempts to determines	3230	No (0)	
Profile	A profile that defines the	3235	3rdParty (0)	
Retries	Transaction retries	3227	3	
Timeout	Time in milliseconds th	3228	250.00	
MaxBlockSiz	Maximum amount of da	3229	124	
HighPriority	High priority rate	3232	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3233	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3234	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3240	No (0)	

ModbusMaster.Slave2.Main - 21 parameters

6. Check if the device is online via the 'Search device' parameter by setting its value to 'Yes'. The search status should not be 'Searching(0).'



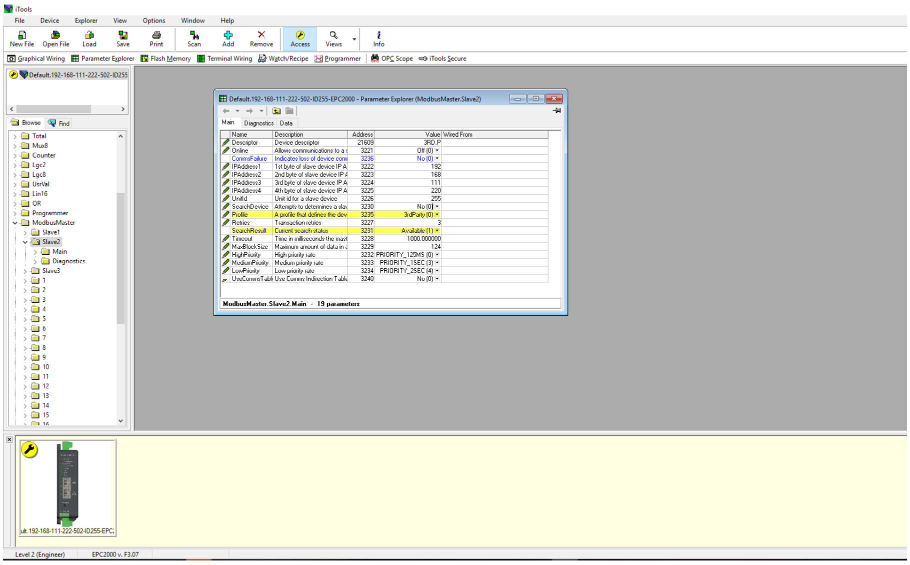
Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave2)

Name	Description	Address	Value	Wired From
Descriptor	Device descriptor	21609	3RD.P	
Online	Allows communications	3221	On (1)	
CommsFailu	Indicates loss of device	3236	No (0)	
IPAddress1	1st byte of slave device	3222	192	
IPAddress2	2nd byte of slave devic	3223	168	
IPAddress3	3rd byte of slave device	3224	111	
IPAddress4	4th byte of slave device	3225	220	
UnitId	Unit id for a slave devic	3226	255	
SearchDevic	Attempts to determines	3230	Yes (1)	
Profile	A profile that defines the	3235	3rdParty (0)	
Retries	Transaction retries	3227	3	
SearchResu	Current search status	3231	Searching (0)	
Timeout	Time in milliseconds th	3228	1000.00	
MaxBlockSiz	Maximum amount of da	3229	124	
HighPriority	High priority rate	3232	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3233	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3234	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3240	Yes (1)	

ModbusMaster.Slave2.Main - 17 parameters (4 hidden)

7. If the Modbus slave is online then the search result will be ‘Available(1)’, otherwise the result will be ‘Unreachable(3)’. If it is a Eurotherm instrument with a supported profile, the ‘Profile’ parameter will display the Modbus slave’s profile, otherwise it will display ‘3rdParty(0)’.

Note: Changes to the slave profile will default previous data configured to be read from or written to the slave.

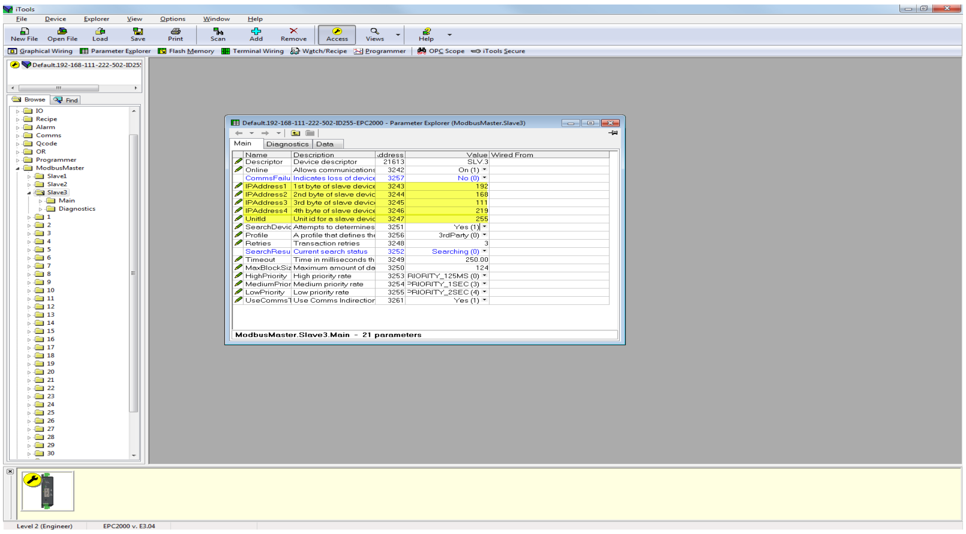

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave2)

Name	Description	Address	Value	Wired From
Descriptor	Device descriptor	21609	3RD.P	
Online	Allows communications to a s	3221	Off (0)	
CommsFailure	Indicates loss of device comı	3236	No (0)	
IPAddress1	1st byte of slave device IP A	3222	192	
IPAddress2	2nd byte of slave device IP A	3223	168	
IPAddress3	3rd byte of slave device IP A	3224	111	
IPAddress4	4th byte of slave device IP A	3225	220	
UnitId	Unit id for a slave device	3226	255	
SearchDevice	Attempts to determines a slav	3230	No (0)	
Profile	A profile that defines the dev	3235	3rdParty (0)	
Retries	Transaction retries	3227	3	
SearchResult	Current search status	3231	Available (1)	
Timeout	Time in milliseconds the mast	3228	1000.000000	
MaxBlockSize	Maximum amount of data in a	3229	124	
HighPriority	High priority rate	3232	PRIORITY_125MS (0)	
MediumPriority	Medium priority rate	3233	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3234	PRIORITY_2SEC (4)	
UseCommsTabl	Use Comms Indirection Table	3240	No (0)	

ModbusMaster.Slave2.Main - 19 parameters

iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Info
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Main Diagnostics Data
Level 2 (Engineer) EPC2000 v. F3.07

8. For the third slave, configure the device’s IP Address and unit ID then trigger ‘SearchDevice’.


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave3)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21613	SLV.3	
Online	Allows communications	3242	On (1)	
CommsFailu	Indicates loss of device	3257	No (0)	
IPAddress1	1st byte of slave device	3243	192	
IPAddress2	2nd byte of slave devic	3244	168	
IPAddress3	3rd byte of slave devic	3245	111	
IPAddress4	4th byte of slave device	3246	219	
UnitId	Unit id for a slave devic	3247	255	
SearchDevic	Attempts to determines	3251	Yes (1)	
Profile	A profile that defines the	3256	3rdParty (0)	
Retries	Transaction retries	3248	3	
SearchResu	Current search status	3252	Searching (0)	
Timeout	Time in milliseconds th	3249	250.00	
MaxBlockSiz	Maximum amount of da	3250	124	
HighPriority	High priority rate	3253	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3254	PRIORITY_1SEC (3)	
LowPriority	Low priority rate	3255	PRIORITY_2SEC (4)	
UseCommsT	Use Comms Indirectior	3261	Yes (1)	

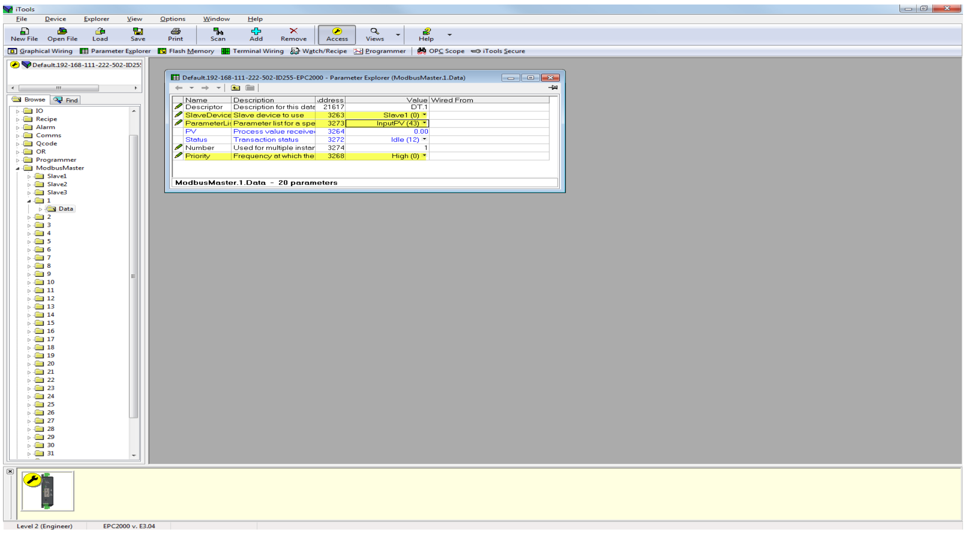
ModbusMaster.Slave3.Main - 21 parameters

iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Main Diagnostics Data
Level 2 (Engineer) EPC2000 v. E3.04

Data Configuration for Cyclic Read/Writes

A maximum of 32 data points can be configured. These data points can be shared among all three slaves or it can be used for a single slave.

For a slave with a known profile, it is possible to configure a data read by selecting the slave and then select the required parameter from the Parameter list drop-down box. The register address, function code, data type and priority for the parameter will be automatically configured. The user still has the option to change the recommended priority.

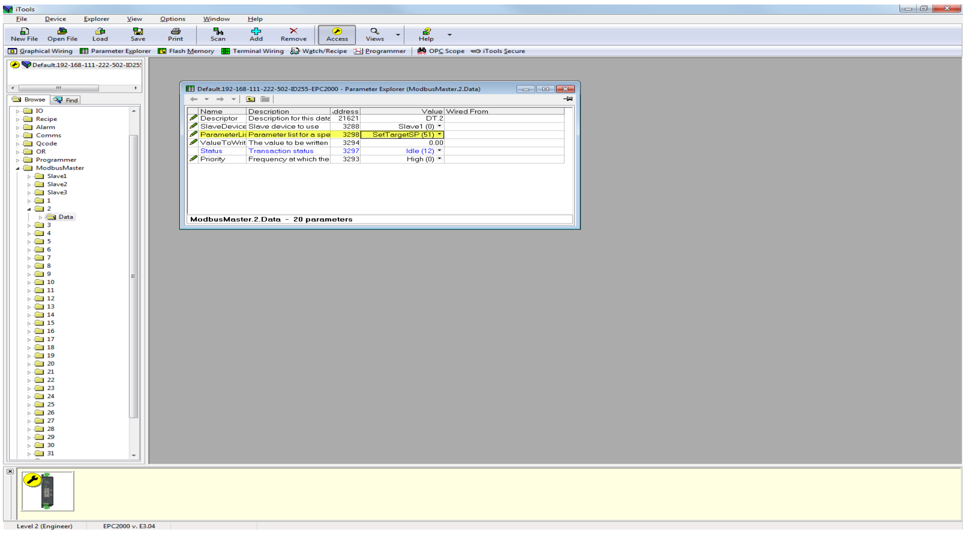


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.1.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21617	DT.1	
SlaveDevice	Slave device to use	3263	Slave1 (0)	
ParameterLis	Parameter list for a spe	3273	InputPV (43)	
PV	Process value receive	3264	0.00	
Status	Transaction status	3272	Idle (12)	
Number	Used for multiple instar	3274	1	
Priority	Frequency at which the	3268	High (0)	

ModbusMaster.1.Data - 20 parameters

1. To configure a write for a known profile, select parameter to write from the Parameter List drop-down box.

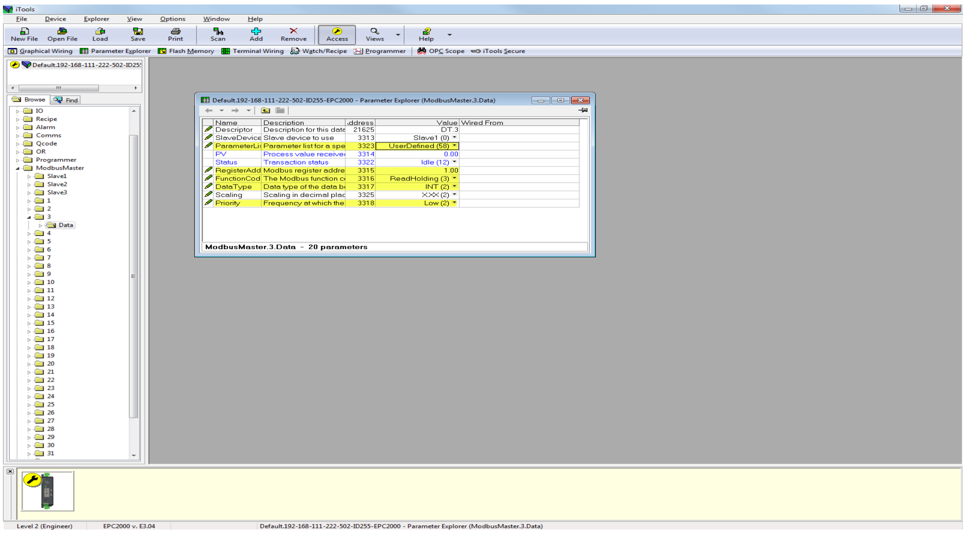


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.2.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21621	DT.2	
SlaveDevice	Slave device to use	3288	Slave1 (0)	
ParameterLis	Parameter list for a spe	3298	SetTargetSP (51)	
ValueToWrit	The value to be written	3294	0.00	
Status	Transaction status	3297	Idle (12)	
Priority	Frequency at which the	3293	High (0)	

ModbusMaster.2.Data - 20 parameters

2. For a parameter that is not on the Parameter List, the data configuration has to be done manually. Select ‘UserDefined’ from the Parameter List and configure the register address, function code, the data type and priority of data read/write.

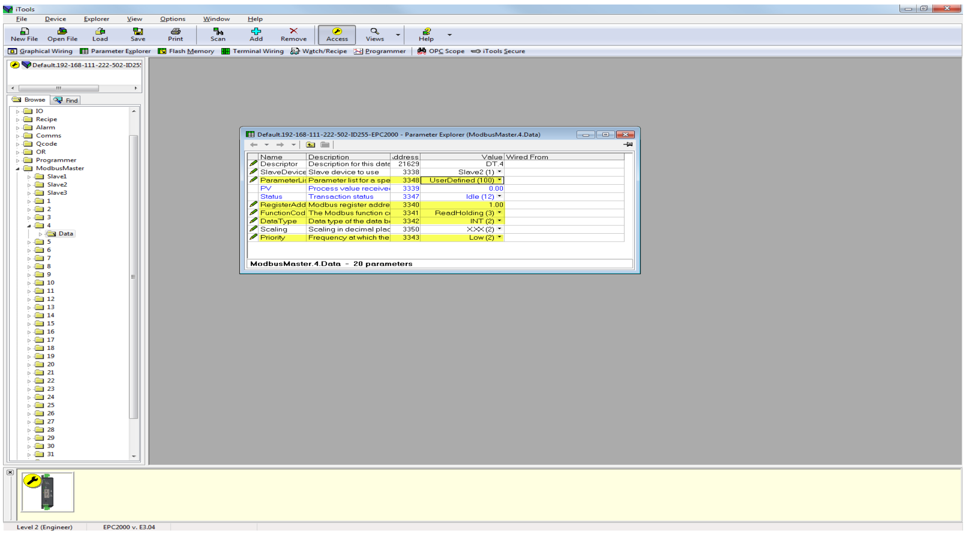


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.3.Data)

Name	Description	Address	Value	Wired From
Descriptor	Description for this data	21625	DT.3	
SlaveDevice	Slave device to use	3313	Slave1 (0)	
ParameterLis	Parameter list for a spe	3323	UserDefined (58)	
PV	Process value receive	3314	0.00	
Status	Transaction status	3322	Idle (12)	
RegisterAdd	Modbus register addre	3315	1.00	
FunctionCod	The Modbus function c	3316	ReadHolding (3)	
DataType	Data type of the data b	3317	INT (2)	
Scaling	Scaling in decimal plac	3325	X.XX (2)	
Priority	Frequency at which the	3318	Low (2)	

ModbusMaster.3.Data - 20 parameters

3. For a third party slave (unsupported profile), select “UserDefined” from the Parameter List drop-down and configure the register address, function code, the data type and priority of data read/write.

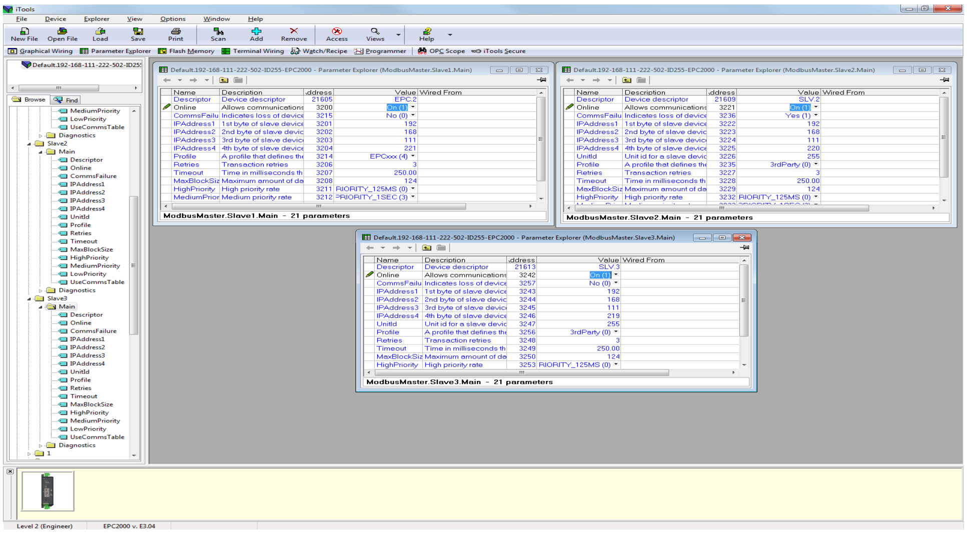


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.4.Data)

Name	Description	Address	Value	Wired From
Descriptor	Description for this data	21629	DT.4	
SlaveDevice	Slave device to use	3338	Slave2 (1)	
ParameterLis	Parameter list for a spe	3348	UserDefined (100)	
PV	Process value receive	3339	0.00	
Status	Transaction status	3347	Idle (12)	
RegisterAdd	Modbus register addre	3340	1.00	
FunctionCod	The Modbus function c	3341	ReadHolding (3)	
DataType	Data type of the data b	3342	INT (2)	
Scaling	Scaling in decimal plac	3350	X.XX (2)	
Priority	Frequency at which the	3343	Low (2)	

ModbusMaster.4.Data - 20 parameters

4. To start cyclic communications to the slaves. Take the Modbus master device out of Config mode and set the Online parameter for each of the slave.


iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Browse Find
Level 2 (Engineer) EPC2000 v. E3.04

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave1.Main)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21605	EPC.2	
Online	Allows communications	3200	On (1)	
CommsFailu	Indicates loss of device	3215	No (0)	
IPAddress1	1st byte of slave device	3201	192	
IPAddress2	2nd byte of slave devic	3202	168	
IPAddress3	3rd byte of slave devic	3203	111	
IPAddress4	4th byte of slave device	3204	221	
Profile	A profile that defines the	3214	EPCxxx (4)	
Retries	Transaction retries	3206	3	
Timeout	Time in milliseconds th	3207	250.00	
MaxBlockSiz	Maximum amount of da	3208	124	
HighPriority	High priority rate	3211	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3212	PRIORITY_1SEC (3)	

ModbusMaster.Slave1.Main - 21 parameters

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave2.Main)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21609	SLV.2	
Online	Allows communications	3221	On (1)	
CommsFailu	Indicates loss of device	3236	Yes (1)	
IPAddress1	1st byte of slave device	3222	192	
IPAddress2	2nd byte of slave devic	3223	168	
IPAddress3	3rd byte of slave devic	3224	111	
IPAddress4	4th byte of slave device	3225	220	
UnitId	Unit id for a slave devic	3226	255	
Profile	A profile that defines the	3235	3rdParty (0)	
Retries	Transaction retries	3227	3	
Timeout	Time in milliseconds th	3228	250.00	
MaxBlockSiz	Maximum amount of da	3229	124	
HighPriority	High priority rate	3232	RIORITY_125MS (0)	

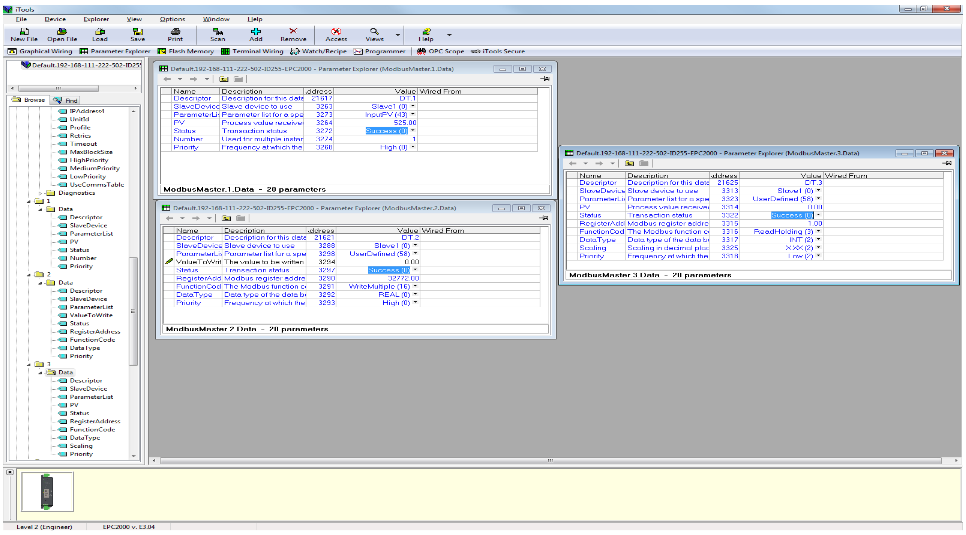
ModbusMaster.Slave2.Main - 21 parameters

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave3.Main)

Name	Description	ddress	Value	Wired From
Descriptor	Device descriptor	21613	SLV.3	
Online	Allows communications	3242	On (1)	
CommsFailu	Indicates loss of device	3257	No (0)	
IPAddress1	1st byte of slave device	3243	192	
IPAddress2	2nd byte of slave devic	3244	168	
IPAddress3	3rd byte of slave devic	3245	111	
IPAddress4	4th byte of slave device	3246	219	
UnitId	Unit id for a slave devic	3247	255	
Profile	A profile that defines the	3256	3rdParty (0)	
Retries	Transaction retries	3248	3	
Timeout	Time in milliseconds th	3249	250.00	
MaxBlockSiz	Maximum amount of da	3250	124	
HighPriority	High priority rate	3253	RIORITY_125MS (0)	

ModbusMaster.Slave3.Main - 21 parameters

The data read and write status should succeed if wiring, comms configuration, slave configuration and data configuration are correct. The PV read will be shown in the Data PV parameter.


iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Browse Find
Level 2 (Engineer) EPC2000 v. E3.04

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.1.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21617	DT.1	
SlaveDevice	Slave device to use	3263	Slave1 (0)	
ParameterLis	Parameter list for a spe	3273	InputPV (43)	
PV	Process value receive	3264	525.00	
Status	Transaction status	3272	Success (0)	
Number	Used for multiple instar	3274	1	
Priority	Frequency at which the	3268	High (0)	

ModbusMaster.1.Data - 20 parameters

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.2.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21621	DT.2	
SlaveDevice	Slave device to use	3288	Slave1 (0)	
ParameterLis	Parameter list for a spe	3298	UserDefined (58)	
ValueToWrit	The value to be written	3294	0.00	
Status	Transaction status	3297	Success (0)	
RegisterAdd	Modbus register addre	3290	32772.00	
FunctionCod	The Modbus function c	3291	WriteMultiple (16)	
DataType	Data type of the data b	3292	REAL (0)	
Priority	Frequency at which the	3293	High (0)	

ModbusMaster.2.Data - 20 parameters

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.3.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21625	DT.3	
SlaveDevice	Slave device to use	3313	Slave1 (0)	
ParameterLis	Parameter list for a spe	3323	UserDefined (58)	
PV	Process value receive	3314	0.00	
Status	Transaction status	3322	Success (0)	
RegisterAdd	Modbus register addre	3315	1.00	
FunctionCod	The Modbus function c	3316	ReadHolding (3)	
DataType	Data type of the data b	3317	INT (2)	
Scaling	Scaling in decimal plac	3325	X.XX (2)	
Priority	Frequency at which the	3318	Low (2)	

ModbusMaster.3.Data - 20 parameters

Data Configuration for Acyclic Data Writes

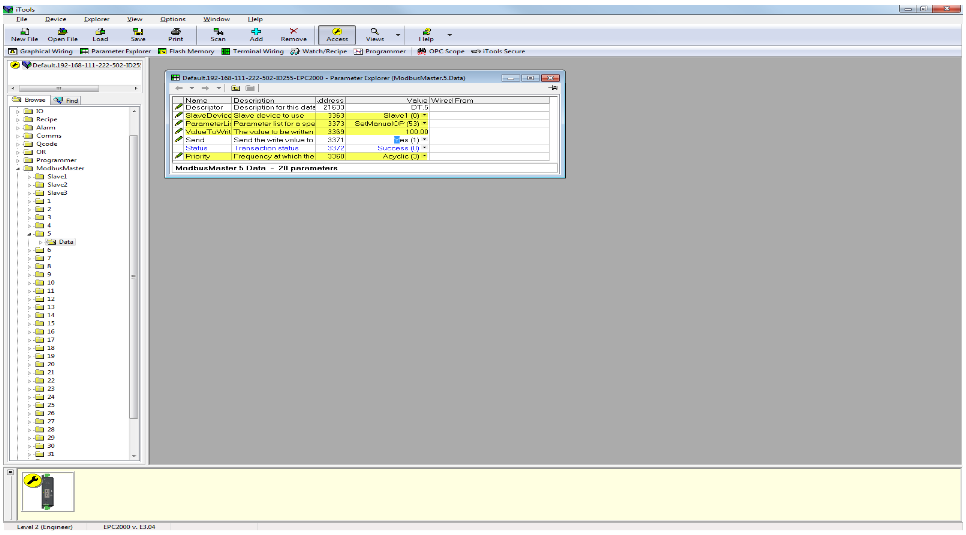
To configure data for Acyclic Data Writes, proceed as follows:

1. Place the Modbus Master device in Configuration mode.

   Note: Cyclic communications to all slaves will stop in configuration mode. The slave online parameter can be set in Operator or Configuration mode but cyclic communications will only run when the device is in Operator mode.

2. For a supported slave profile select the slave and parameter to write to as well as the value to write and then set the Priority to ‘Acyclic(3)’.

Note: Acyclic communications are only available for data writes but they can be triggered in Operator or Configuration mode.

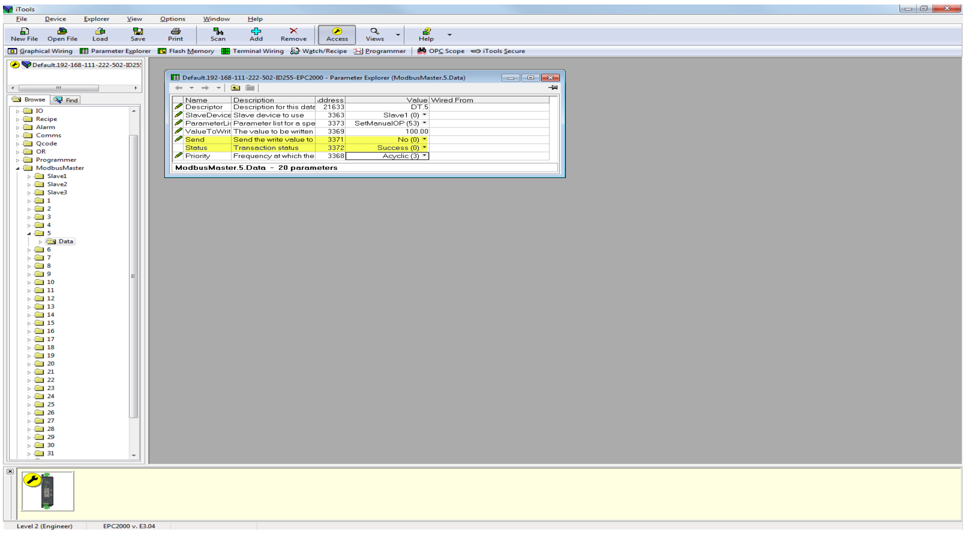

iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Default.192-168-111-222-502-ID255
Browse Find
IO Recipe Alarm Comms Qcode OR Programmer ModbusMaster Slave1 Slave2 Slave3 1 2 3 4 5 Data 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Level 2 (Engineer) EPC2000 v. E3.04

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.5.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21633	DT.5	
SlaveDevice	Slave device to use	3363	Slave1 (0)	
ParameterLis	Parameter list for a spe	3373	SetManualOP (53)	
ValueToWrit	The value to be written	3369	100.00	
Send	Send the write value to	3371	Yes (1)	
Status	Transaction status	3372	Success (0)	
Priority	Frequency at which the	3368	Acyclic (3)	

ModbusMaster.5.Data - 20 parameters

3. To send the write request, set the ‘Send’ parameter. The Status will go to ‘Pending(13)’ briefly before going to ‘Success’ when the parameter has been written. If the write has failed then the Status will show the reason for the failure.

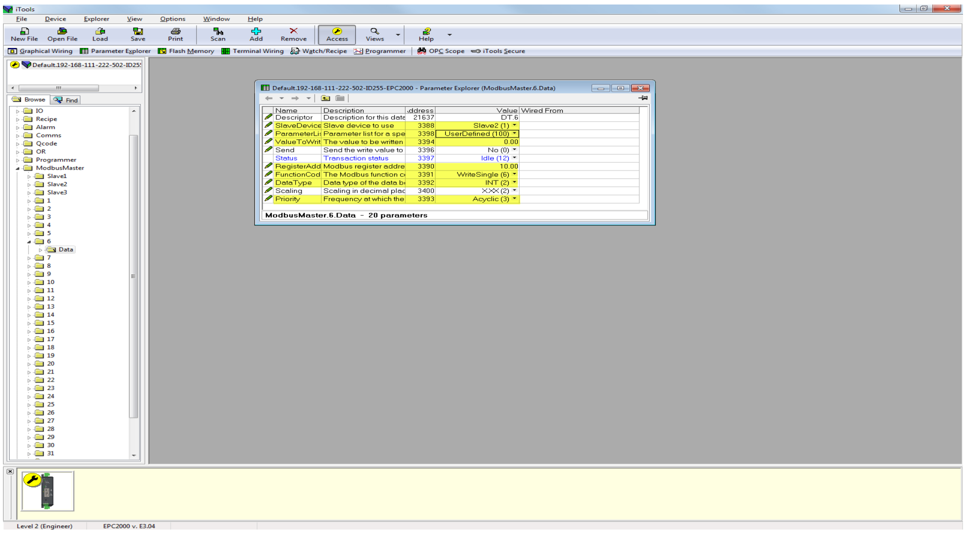

iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Help
Graphical Wiring Parameter Explorer Flash Memory Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
Default.192-168-111-222-502-ID255
Browse Find
IO Recipe Alarm Comms Qcode OR Programmer ModbusMaster Slave1 Slave2 Slave3 1 2 3 4 5 Data 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Level 2 (Engineer) EPC2000 v. E3.04

Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.5.Data)

Name	Description	ddress	Value	Wired From
Descriptor	Description for this data	21633	DT.5	
SlaveDevice	Slave device to use	3363	Slave1 (0)	
ParameterLis	Parameter list for a spe	3373	SetManualOP (53)	
ValueToWrit	The value to be written	3369	100.00	
Send	Send the write value to	3371	No (0)	
Status	Transaction status	3372	Success (0)	
Priority	Frequency at which the	3368	Acyclic (3)	

ModbusMaster.5.Data - 20 parameters

4. For an unsupported slave profile (Third party) select the slave, select ‘UserDefined’ from the Parameter List drop-down and configure the register address, function code (must be a write), the data type, the value to write and then set the Priority to ‘Acyclic(3)’.



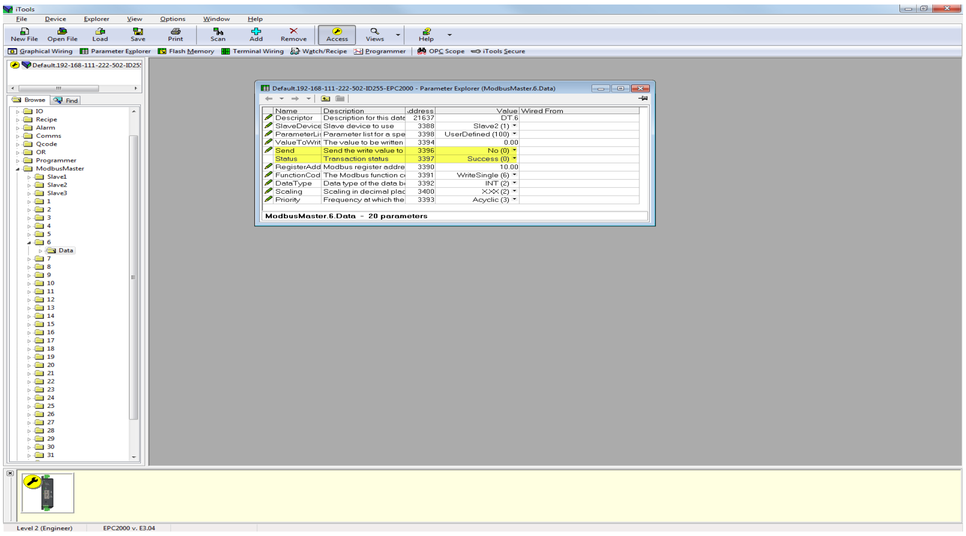
Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.6.Data)

Name	Description	.ddress	Value	Wired From
Descriptor	Description for this data	21637	DT.6	
SlaveDevice	Slave device to use	3388	Slave2 (1)	
ParameterLi	Parameter list for a spe	3398	UserDefined (100)	
ValueToWrit	The value to be written	3394	0.00	
Send	Send the write value to	3396	No (0)	
Status	Transaction status	3397	Idle (12)	
RegisterAdd	Modbus register addre	3390	10.00	
FunctionCod	The Modbus function c	3391	WriteSingle (6)	
DataType	Data type of the data b	3392	INT (2)	
Scaling	Scaling in decimal plac	3400	X.XX (2)	
Priority	Frequency at which the	3393	Acyclic (3)	

ModbusMaster.6.Data - 20 parameters

Level 2 (Engineer) EPC2000 v. E3.04

5. To send the write request, set the 'Send' parameter. The Status will go to 'Pending(13)' briefly before going to 'Success' when the parameter has been written. If the write has failed then the Status will show the reason for the failure.



Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.6.Data)

Name	Description	.ddress	Value	Wired From
Descriptor	Description for this data	21637	DT.6	
SlaveDevice	Slave device to use	3388	Slave2 (1)	
ParameterLi	Parameter list for a spe	3398	UserDefined (100)	
ValueToWrit	The value to be written	3394	0.00	
Send	Send the write value to	3396	No (0)	
Status	Transaction status	3397	Success (0)	
RegisterAdd	Modbus register addre	3390	10.00	
FunctionCod	The Modbus function c	3391	WriteSingle (6)	
DataType	Data type of the data b	3392	INT (2)	
Scaling	Scaling in decimal plac	3400	X.XX (2)	
Priority	Frequency at which the	3393	Acyclic (3)	

ModbusMaster.6.Data - 20 parameters

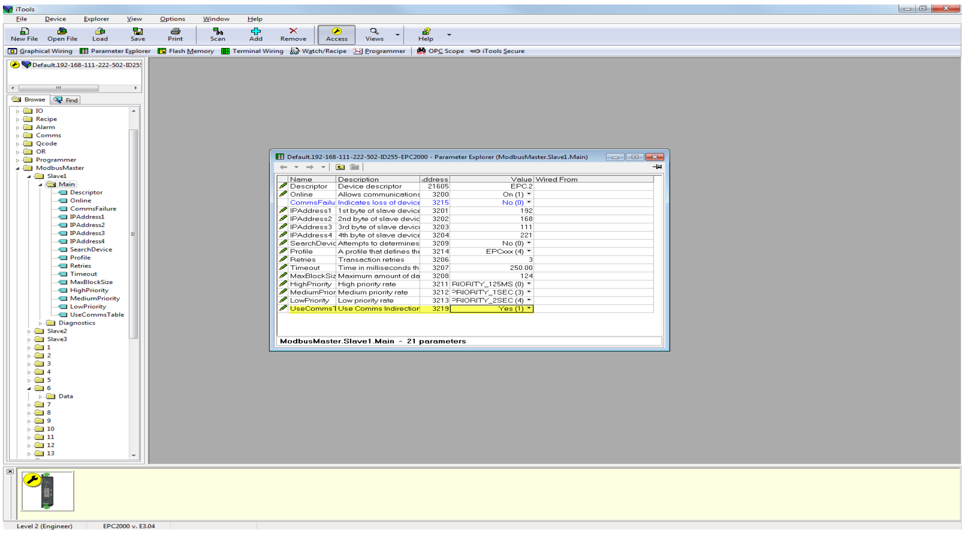
Level 2 (Engineer) EPC2000 v. E3.04

Accessing Modbus Master Data from the Modbus Indirection Table

To allow efficient reads from and writes to Modbus Master data, CommsTab Function Block can be used to map Modbus Master data into a contiguous block of Modbus addresses in the range:

15360(0x3C00) to 15615(0x3CFF)

Modbus Master data can be auto-configured to be accessible from the Modbus Indirection table by placing the Modbus Master device into Configuration mode and setting the UseCommsTable parameter from any one of the slave configuration windows and then taking the Modbus Master device out of Configuration mode to initialize the CommsTab Function Block settings.

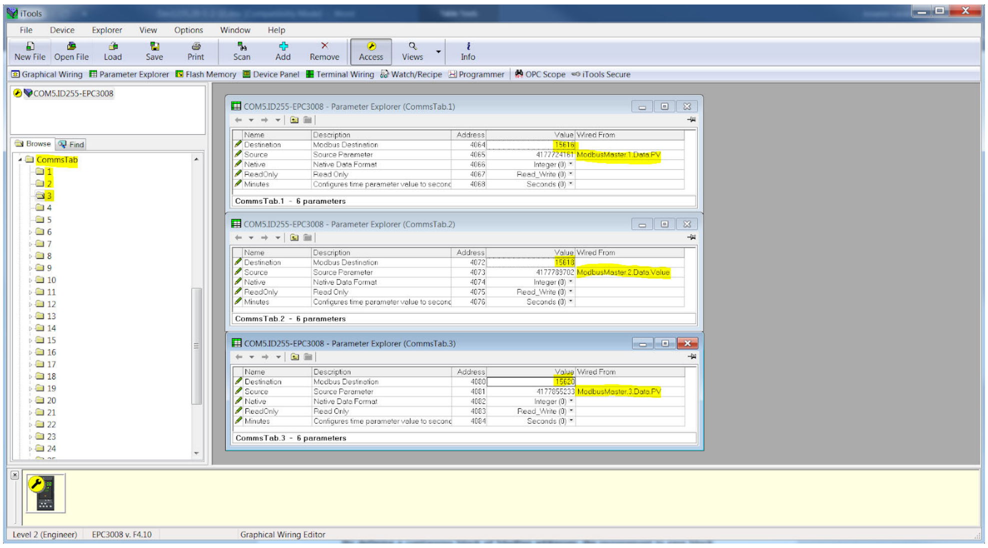


Default.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (ModbusMaster.Slave1.Main)

Name	Description	›ddress	Value	Wired From
Descriptor	Device descriptor	21605	EPC.2	
Online	Allows communications	3200	On (1)	
CommsFailu	Indicates loss of device	3215	No (0)	
IPAddress1	1st byte of slave device	3201	192	
IPAddress2	2nd byte of slave devic	3202	168	
IPAddress3	3rd byte of slave devic	3203	111	
IPAddress4	4th byte of slave device	3204	221	
SearchDevic	Attempts to determines	3209	No (0)	
Profile	A profile that defines the	3214	EPCxxx (4)	
Retries	Transaction retries	3206	3	
Timeout	Time in milliseconds th	3207	250.00	
MaxBlockSiz	Maximum amount of da	3208	124	
HighPriority	High priority rate	3211	RIORITY_125MS (0)	
MediumPrior	Medium priority rate	3212	⊃RIORITY_1SEC (3)	
LowPriority	Low priority rate	3213	⊃RIORITY_2SEC (4)	
UseCommsT	Use Comms Indirection	3219	Yes (1)	

ModbusMaster.Slave1.Main - 21 parameters

1. In Operator mode, the CommsTab Function Block should now show every configured Modbus Master data.
2. The user can then change Native, ReadOnly and Minutes parameters from default to configure how the data is presented from the Modbus indirection table.



COM5.ID255-EPC3008 - Parameter Explorer (CommsTab.1)

Name	Description	Address	Value	Wired From
Destination	Modbus Destination	4064	15616	
Source	Source Parameter	4065	4177724161	ModbusMaster.1.Data.PV
Native	Native Data Format	4066	Integer (0)	
ReadOnly	Read Only	4067	Read_Write (0)	
Minutes	Configures time parameter value to seconc	4068	Seconds (0)	

CommsTab.1 - 6 parameters

COM5.ID255-EPC3008 - Parameter Explorer (CommsTab.2)

Name	Description	Address	Value	Wired From
Destination	Modbus Destination	4072	15618	
Source	Source Parameter	4073	4177789702	ModbusMaster.2.Data.Value
Native	Native Data Format	4074	Integer (0)	
ReadOnly	Read Only	4075	Read_Write (0)	
Minutes	Configures time parameter value to seconc	4076	Seconds (0)	

CommsTab.2 - 6 parameters

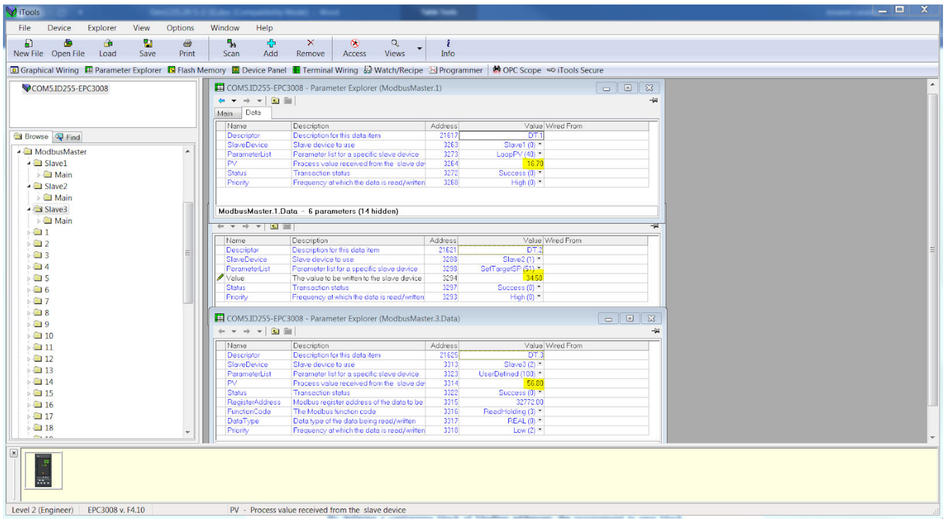
COM5.ID255-EPC3008 - Parameter Explorer (CommsTab.3)

Name	Description	Address	Value	Wired From
Destination	Modbus Destination	4080	15620	
Source	Source Parameter	4081	4177855233	ModbusMaster.3.Data.PV
Native	Native Data Format	4082	Integer (0)	
ReadOnly	Read Only	4083	Read_Write (0)	
Minutes	Configures time parameter value to seconc	4084	Seconds (0)	

CommsTab.3 - 6 parameters

The screenshots below show Modbus Master data auto-configured to appear at the Modbus Indirection table and the values read by a third party Modbus Master from a Eurotherm Modbus Master device:

Third party Modbus TCP Master read data	Modbus Master device data
0x0686	16.70
0x0D7A	34.50
0x1630	56.80


iTools
File Device Explorer View Options Window Help
New File Open File Load Save Print Scan Add Remove Access Views Info
Graphical Wiring Parameter Explorer Flash Memory Device Panel Terminal Wiring Watch/Recipe Programmer OPC Scope iTools Secure
COM5.ID255-EPC3008
Browse Find
ModbusMaster
Slave1
Main
Slave2
Main
Slave3
Main
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Main Data

COM5.ID255-EPC3008 - Parameter Explorer (ModbusMaster.1)

Name	Description	Address	Value	Wired From
Descriptor	Description for this data item	21617	DT.1	
SlaveDevice	Slave device to use	3263	Slave1 (0)	
ParameterList	Parameter list for a specific slave device	3273	LoopPV (40)	
PV	Process value received from the slave de	3264	16.70	
Status	Transaction status	3272	Success (0)	
Priority	Frequency at which the data is read/written	3268	High (0)	

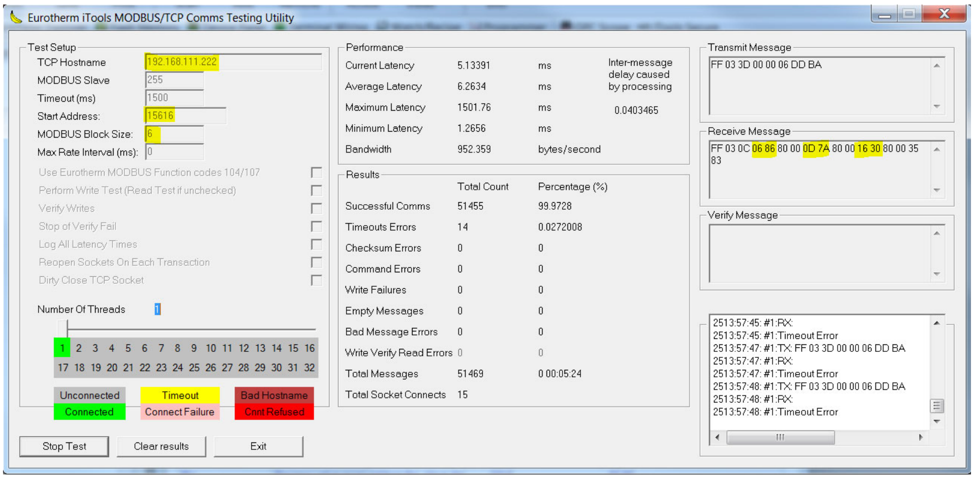
ModbusMaster.1.Data - 6 parameters (14 hidden)

Name	Description	Address	Value	Wired From
Descriptor	Description for this data item	21621	DT.2	
SlaveDevice	Slave device to use	3288	Slave2 (1)	
ParameterList	Parameter list for a specific slave device	3298	SetTargetSP (51)	
Value	The value to be written to the slave device	3294	34.50	
Status	Transaction status	3297	Success (0)	
Priority	Frequency at which the data is read/written	3293	High (0)	

COM5.ID255-EPC3008 - Parameter Explorer (ModbusMaster.3.Data)

Name	Description	Address	Value	Wired From
Descriptor	Description for this data item	21625	DT.3	
SlaveDevice	Slave device to use	3313	Slave3 (2)	
ParameterList	Parameter list for a specific slave device	3323	UserDefined (100)	
PV	Process value received from the slave de	3314	56.80	
Status	Transaction status	3322	Success (0)	
RegisterAddress	Modbus register address of the data to be	3315	32772.00	
FunctionCode	The Modbus function code	3316	ReadHolding (3)	
DataType	Data type of the data being read/written	3317	REAL (0)	
Priority	Frequency at which the data is read/written	3318	Low (2)	

Level 2 (Engineer) EPC3008 v. F4.10 PV - Process value received from the slave device



Test Setup

TCP Hostname 192.168.111.222
MODBUS Slave 255
Timeout (ms) 1500
Start Address: 15616
MODBUS Block Size: 6
Max Rate Interval (ms): 0

Use Eurotherm MODBUS Function codes 104/107 □
Perform Write Test (Read Test if unchecked) □
Verify Writes □
Stop of Verify Fail □
Log All Latency Times □
Reopen Sockets On Each Transaction □
Dirty Close TCP Socket □

Number Of Threads 1

Performance

Current Latency	5.13391	ms	Inter-message delay caused by processing
Average Latency	6.2634	ms	
Maximum Latency	1501.76	ms	0.0403465
Minimum Latency	1.2656	ms	
Bandwidth	952.359	bytes/second	

Results

	Total Count	Percentage (%)
Successful Comms	51455	99.9728
Timeouts Errors	14	0.0272008
Checksum Errors	0	0
Command Errors	0	0
Write Failures	0	0
Empty Messages	0	0
Bad Message Errors	0	0
Write Verify Read Errors	0	0
Total Messages	51469	0 00:05:24
Total Socket Connects	15	

Transmit Message

```
FF 03 3D 00 00 06 DD BA
```

Receive Message

```
FF 03 0C 06 86 80 00 0D 7A 80 00 16 30 80 00 35
83
```

Verify Message

```
2513:57:45: #1:RX:
2513:57:45: #1:Timeout Error
2513:57:47: #1:TX: FF 03 3D 00 00 06 DD BA
2513:57:47: #1:RX:
2513:57:47: #1:Timeout Error
2513:57:48: #1:TX: FF 03 3D 00 00 06 DD BA
2513:57:48: #1:RX:
2513:57:48: #1:Timeout Error
```

Note: There are 32 parameter available for configuration in the CommsTab Function Block, one for each Modbus Master data. Partition the Modbus Indirection table for reads and writes for efficient data access.

Comms Indirection Table

EPC2000 controllers make a fixed set of parameters available over digital communications using Modbus addresses. This is known as the SCADA Table. The SCADA Modbus address area is 0 to 15615 (0x3CFF).

The Commstab function block allows a Source parameter value to be available(read/write) from a Destination Modbus address.

The following parameters however cannot be set as a Destination Modbus address:

- Instrument Number
- Instrument Type
- Instrument Firmware Version
- Company ID
- Feature Security Words

The following contiguous Modbus addresses have been reserved for use by the Commstab function block. By default the addresses have no associated parameters:

Modbus Range (Decimal)	Modbus Range (Hex)
15360 to 15615	3C00 to 3CFF

PROFINET

PROFINET is an open industrial Ethernet based networking solution for automation. It is similar to PROFIBUS in that it enables distributed IO control from a PLC. PROFINET uses TCP/IP and IT standards, and is, in effect, real-time Ethernet.

PROFINET IO was developed for real time (RT) and isochronous real time (IRT) communication with the decentral periphery. The designations RT and IRT merely describe the real-time properties for the communication within PROFINET IO.

There are four stages to setting up a network:

- "PROFINET Wiring" on page 265
- "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266
- "Configuring the Cyclic (IO Data) Data Exchange" on page 270
- "Acyclic Data Exchange (Record Data)" on page 271

NOTICE

UNINTENDED EQUIPMENT OPERATION

When ordering the product, select the appropriate communication protocol.

EPC2000 must be ordered with a compatible firmware version for the PROFINET function to be used. The PROFINET version has to be specifically selected - please check and confirm your order is correct.

Failure to follow these instructions can result in equipment damage.

See "Order Codes" on page 29 for further information.

NOTICE

UNINTENDED EQUIPMENT OPERATION

The EPC2000 PROFINET also supports Modbus/TCP protocol for the device configuration through iTools, but this protocol must not be used when the EPC2000 Programmable Controller is commissioned, operational and especially when the PROFINET cyclic (IO Data) data exchange is active.

Failure to follow these instructions can result in equipment damage.

PROFINET Features

- 10/100Mbps Auto-negotiation
- Galvanically isolated bus electronics
- Field pluggable option
- Polled and Explicit I/O messaging connection
- PROFINET IO Device version: V2.41
- Device Type: Compact field device
- Conformance class: CC-A
- Real Time class: RT-1
- Supported Net Load class: Class 1
- Number of slots: 2 (see description below)
- Minimum Device interval (cycle time): 64ms
- Number of available modules: 2 (Input and Output Fieldbus IO Gateway modules)

PROFINET Wiring

PROFINET capability is provided by the RJ45 Ethernet port, "Network and iTools connection" on page 48.

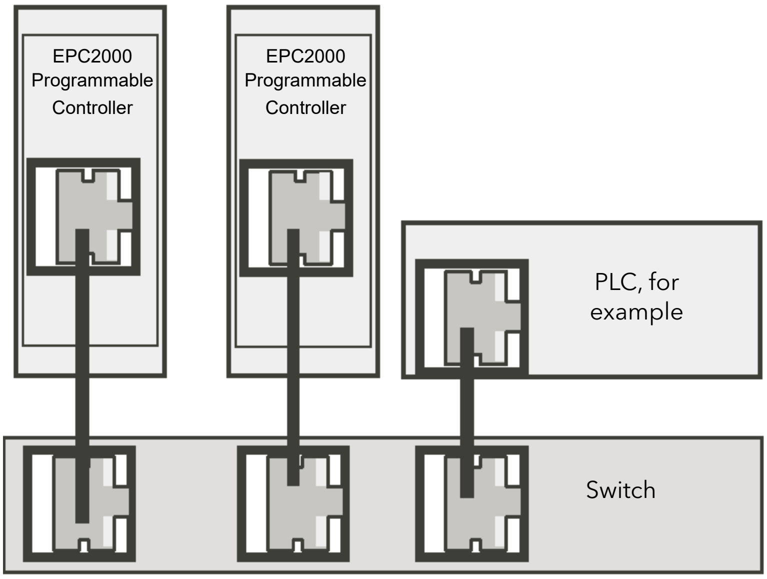
The PROFINET port is a 100Mb, full duplex operation port and should be connected via a PROFINET Ethernet Switch to a IO Controller (e.g. PLC) with Cat5e or higher cable via the standard RJ45 connector (maximum length 100m).

The interconnecting cables should be fitted with plugs provided with an outer metallic shell with the shell connected to the wire screen of the cable.

Notes:

1. Although CC-A requirements can be met by the use of ordinary Ethernet Switches (supporting the VLANs), it is strongly recommended to use PROFINET managed Industrial Switches (Managed Switches, e.g. MOXA EDS-408A-PN, Scalance X204IRT). This will allow future migration to the Conformance Class CC-B without the need to change your Network infrastructure (“Network Diagnostic” with SNMP, LLDP-MIB for the “Device Replacement without Engineering Tool”).

2. The MAC address of the device is stated on the product label. To ensure the 'Neighborhood detection' functionality with LLDP, each physical Ethernet Port requires its own MAC address. Therefore, P1 uses the device MAC address incremented by one and by two for P2.


EPC2000 Programmable Controller
EPC2000 Programmable Controller
PLC, for example
Switch

Setting up the EPC2000 Programmable Controller for PROFINET

The EPC2000 Programmable Controller functions as a PROFINET IO Device once configured and included in a PROFINET PLC. Before installing and commissioning the controller will require configuring using any of the following suggested applications;

- PROFINET Configurations tools;
    - PLC Programming with SIMATIC STEP7 (https://new.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal/software/step7-tia-portal.html)
    - PROFINET Commander (https://profinetcommander.com)
- iTools (used to configure the control application);
    - "What is iTools?" on page 65
    - "Control application and Configuration" on page 58

The following is a guide, using SIEMENS TIA Portal as the PROFINET Configuration tool to commission the PROFINET IO Device.

Note: A new EPC2000 Programmable Controller has a default IP address of 0.0.0.0, before using iTools it is necessary to initially set up the controller using PROFINET configuration tool, for example Siemens Totally Integrated Automation (TIA) Portal.

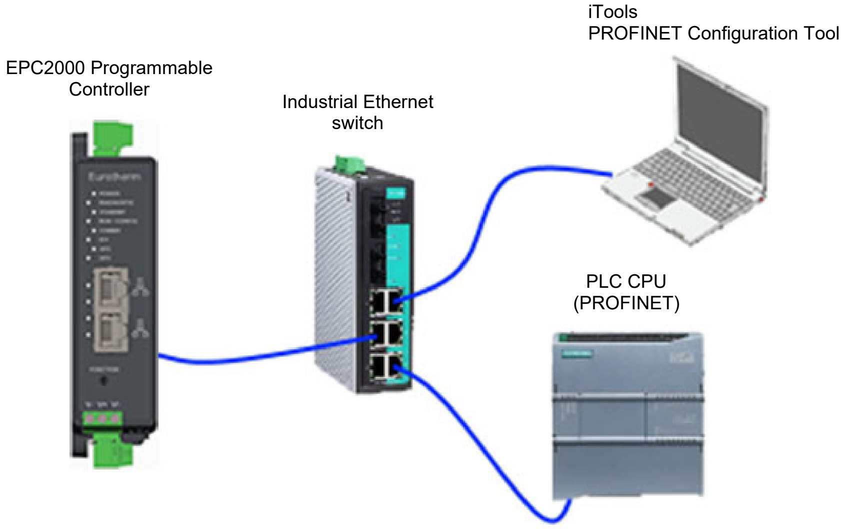
The first thing to do is to identify the PROFINET Device (EPC2000 Programmable Controller) on the network. This is done automatically with the PROFINET tool which uses a specific DCP service for this purpose (DCP Identity Request).

See "iTools & PROFINET Connection" on page 267 for a typical network configuration, containing both EPC2000 Programmable Controller and PROFINET PLC.

Device Name & IP Address - configure
Once connected and identified, you can adjust the “Device Name” and the IP address configuration for the selected EPC2000 Programmable Controller. This is done with a PROFINET configuration tool, for further information see "Commissioning using DCP protocol" on page 267.

iTools & PROFINET Connection

Connect the EPC2000 Programmable Controller to the PROFINET Configuration tool and to iTools - below a single controller is displayed, but multiple controllers can be connected.


iTools
PROFINET Configuration Tool
EPC2000 Programmable Controller
Industrial Ethernet switch
PLC CPU (PROFINET)

Commissioning using DCP protocol

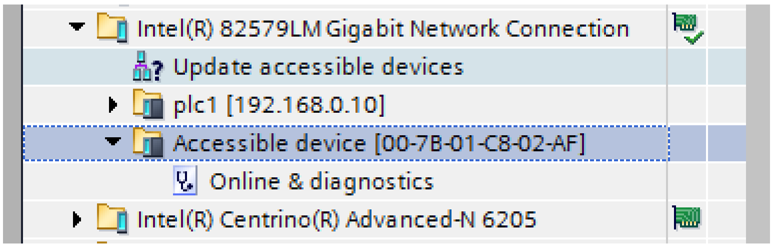
This section describes ‘Device Name’ and ‘IP Configuration’ assignment.

A PROFINET Device is characterised by its ‘Device Name’ (named also ‘Station Name’) and its IP address.

The configuration of a PROFINET device is based on the DCP protocol which is used specifically to assign the ‘Device Name’ or to assign the IP configuration (IP address, Network Mask …).

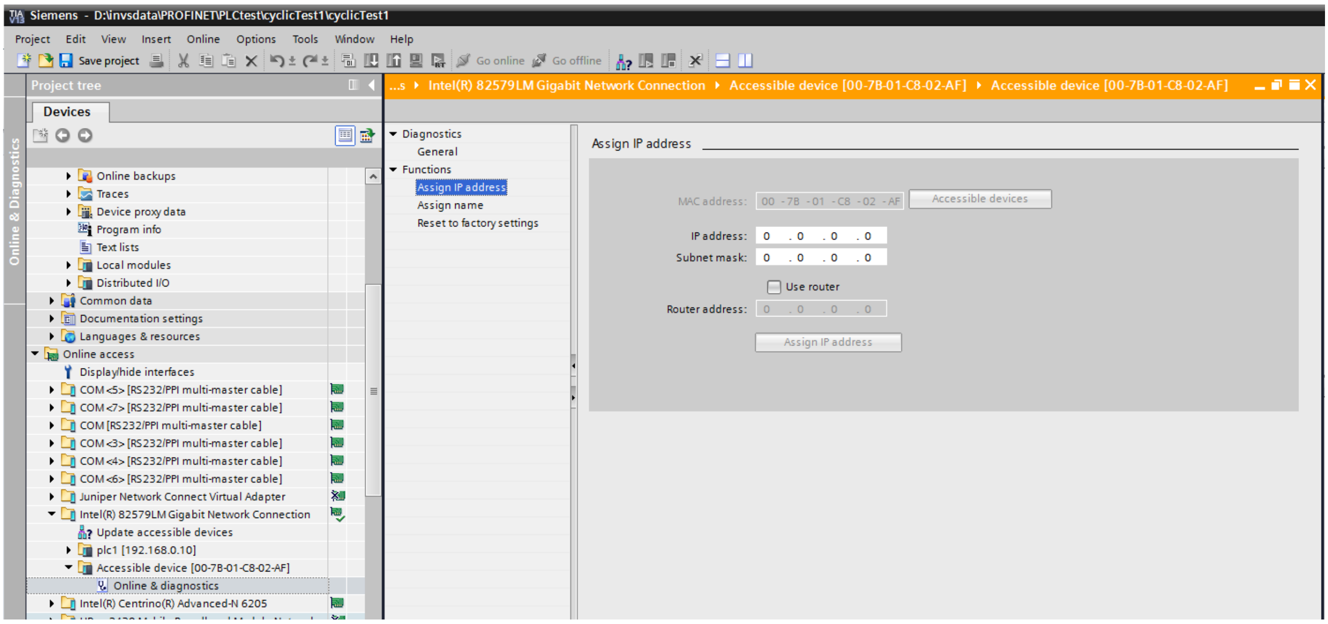
PROFINET Config Tool - Device setup (IP address)
An EPC2000 Programmable Controller which is just ‘Out of the box’ has it’s ‘Device Name’ and the IP configuration address set to Null by default. (The MAC address is used initially by the DCP protocol for setting up the ‘Device Name’).

During the system set up, the PROFINET configuration tool first identifies the existing devices in the system (by sending out a ‘DCPIdentity.req’), shown below. This example uses the ©Siemens TIA Portal / STEP 7 (‘Update accessible devices’ function).



Intel(R) 82579LM Gigabit Network Connection
Update accessible devices
plc1 [192.168.0.10]
Accessible device [00-7B-01-C8-02-AF]
Online & diagnostics
Intel(R) Centrino(R) Advanced-N 6205

The next step assigns the ‘IP Configuration’ and the ‘Device Name’. This operation can be performed by clicking on the ‘Online & diagnostics’.


Siemens - D:\invsdata\PROFINET\PLCtest\cyclicTest1\cyclicTest1
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offline
Project tree
Devices
Online & Diagnostics
Online backups
Traces
Device proxy data
Program info
Text lists
Local modules
Distributed I/O
Common data
Documentation settings
Languages & resources
Online access
Display/hide interfaces
COM <5> [RS232/PPI multi-master cable]
COM <7> [RS232/PPI multi-master cable]
COM [RS232/PPI multi-master cable]
COM <3> [RS232/PPI multi-master cable]
COM <4> [RS232/PPI multi-master cable]
COM <6> [RS232/PPI multi-master cable]
Juniper Network Connect Virtual Adapter
Intel(R) 82579LM Gigabit Network Connection
Update accessible devices
plc1 [192.168.0.10]
Accessible device [00-7B-01-C8-02-AF]
Online & diagnostics
Intel(R) Centrino(R) Advanced-N 6205
...s ▸ Intel(R) 82579LM Gigabit Network Connection ▸ Accessible device [00-7B-01-C8-02-AF] ▸ Accessible device [00-7B-01-C8-02-AF]
Diagnostics
General
Functions
Assign IP address
Assign name
Reset to factory settings
Accessible devices
Assign IP address

Assign IP address

MAC address: 00 -7B -01 -C8 -02 -AF

IP address: 0 . 0 . 0 . 0

Subnet mask: 0 . 0 . 0 . 0

□ Use router

Router address: 0 . 0 . 0 . 0

Note: The Default Gateway can be changed in the same way (named ‘Router address’ in this example).

IP Address - Requirements

The IP address must be configured as a permanent, fixed IP address.

This guide demonstrates the use of © Siemens TIA Portal / STEP 7 which configures a fixed IP address. It is possible various third party PROFINET configurations tools may configure only temporary IP addresses.

> **Note:** At Firmware Upgrade (or Cloning to a device) a permanent fixed IP address is a requirement, see "Firmware Upgrade" on page 295.

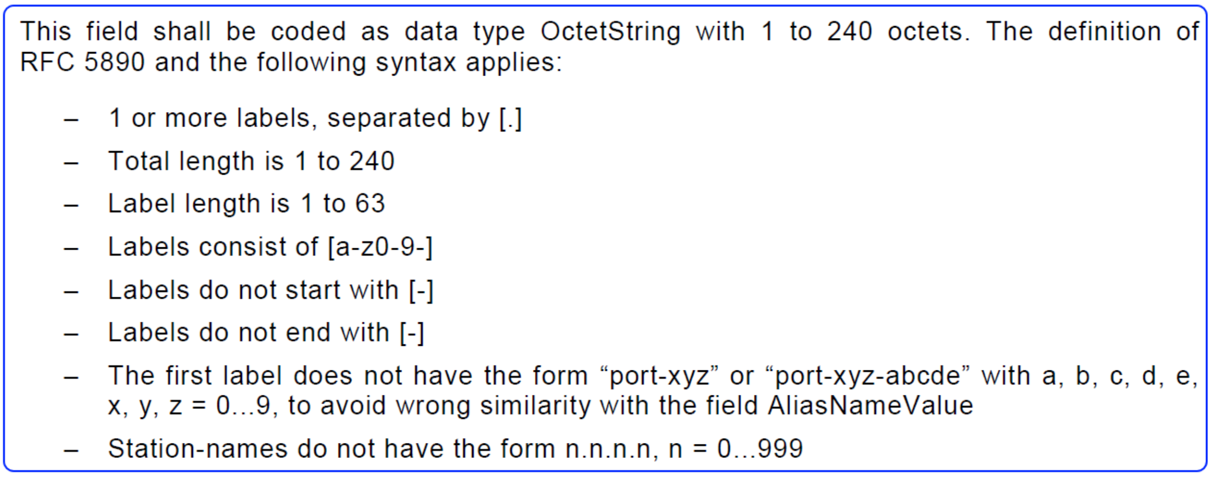
Device Name

The Device Name is used to identify a Device on a PROFINET node.

Device Name via the DCP protocol

The Device Name is written to the Device by the PROFINET configuration tool via the DCP protocol (see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266).

The length can not exceed 240 characters, only the lower case characters are authorized to be used.



This field shall be coded as data type OctetString with 1 to 240 octets. The definition of RFC 5890 and the following syntax applies:

- 1 or more labels, separated by [.]
- Total length is 1 to 240
- Label length is 1 to 63
- Labels consist of [a-z0-9-]
- Labels do not start with [-]
- Labels do not end with [-]
- The first label does not have the form “port-xyz” or “port-xyz-abcde” with a, b, c, d, e, x, y, z = 0...9, to avoid wrong similarity with the field AliasNameValue
- Station-names do not have the form n.n.n.n, n = 0...999

The Device Name respecting these rules can be read or written in EPC2000 Programmable Controller by using the PROFINET tool, for example TIA Portal/STEP 7.

Display of the Device Name in iTools

The last 64 characters of the Device Name are displayed in iTools in the ‘Comms Functional Block’ through the ‘PN_DevName’ parameter (read only).

Other DCP Services

In addition to the assignment of the ‘Device Name’ and the ‘IP Configuration’, the DCP protocol provides the following services for EPC2000 Programmable Controller.

Flash LED (also called “Flash Once”)

The DCP service provides easy visual identification of a Device among a group of Devices. For this, the Ethernet LEDs will flash with a duration of three seconds and a frequency of 1Hz (500ms on, 500ms off).

Reset to factory

The DCP service allows the IP Configuration to be reset (to 0.0.0.0) and the Device Name (reset to “”) to its as-delivered condition, the device returns to its state.

EPC2000 PROFINET - Connecting and Accessing iTools

Once the EPC2000 Programmable Controller has had the IP address etc. configured, change from the PROFINET default and updated to an iTools accessible IP address (and noted for later use) the controller can be connected to iTools, see the following for details;

- "Ethernet Connection, using iTools Control Panel and Scan feature" on page 52
- "Connecting to EPC2000 using iTools" on page 233.

Once connected to iTools you can use iTools to configure the EPC2000 Programmable Controllers control application (i.e. setpoint, pv, alarm functions and Fieldbus gateway).

Other additional information (though not all are necessary for initial setup);

- "Fieldbus I/O Gateway" on page 275 Fieldbus
- "Reset the Controller’s IP address" on page 232

Note: Continue reading to understand the remaining configuration tasks (i.e. Fieldbus Gateway, Cyclic and Acyclic (IO Data) Data Exchange).

NOTICE

UNINTENDED EQUIPMENT OPERATION

The EPC2000 PROFINET also supports Modbus/TCP protocol for the device configuration through iTools, but this protocol must not be used when the EPC2000 Programmable Controller is commissioned, operational and especially when the PROFINET cyclic (IO Data) data exchange is active.

Failure to follow these instructions can result in equipment damage.

EPC2000 Programmable Controller Modules

Fieldbus I/O Gateway IOGW Modules (Generic Modules)

The use of the Fieldbus IO Gateways allows the acquisition of some selected EPC2000 parameters by using the configuration tool named ‘iTools’.

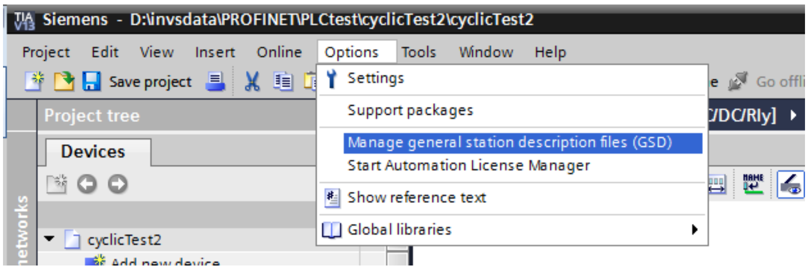
The two IOGW Modules are defined in the following table.

Module name (EPC2000 IOGW)	In / Out attribute (from PLC view)	Size (Bytes/Reg)
Fieldbus Input I/O Gateway	In	128/64
Fieldbus Output I/O Gateway	Out	128/64

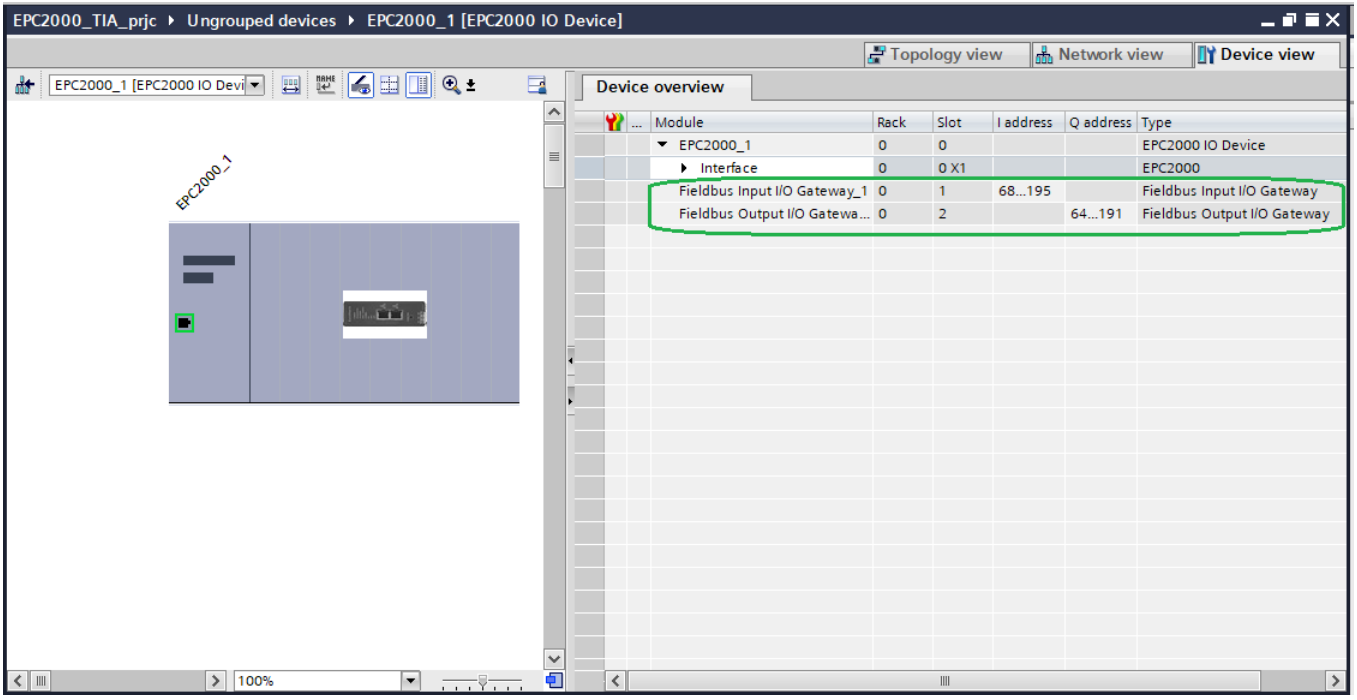
Configuring the Cyclic (IO Data) Data Exchange

During the PROFINET commissioning, the principle is to 'plug' the first module (representing the "Input I/O Gateway") into the slot 1 and the second one (representing the "Output I/O Gateway") into the slot 2 (at this stage, it is assumed that the 'Device Name' and the 'IP Configuration' has already been set up).

This is performed with the PROFINET configuration tool, (for example TIA Portal STEP 7 as displayed) and with an appropriate GSDML file.

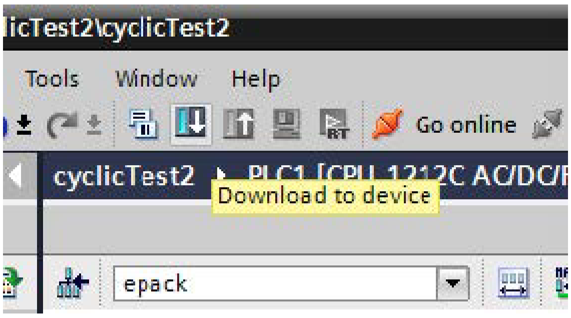

Siemens - D:\invsdata\PROFINET\PLCtest\cyclicTest2\cyclicTest2
Project Edit View Insert Online Options Tools Window Help
Save project
Go offli
Project tree
Devices
Networks
cyclicTest2
Add new device
Settings
Support packages
Manage general station description files (GSD)
Start Automation License Manager
Show reference text
Global libraries
/DC/Rly]

The "Drag and Drop" of the Input and Output Modules (Input and Output I/O Gateway) respectively into Slot 1 and Slot 2 of EPC2000 Programmable Controller.


EPC2000_TIA_prjc ▸ Ungrouped devices ▸ EPC2000_1 [EPC2000 IO Device]
Topology view
Network view
Device view
EPC2000_1 [EPC2000 IO Devi
EPC2000_1
100%
Device overview

Module	Rack	Slot	I address	Q address	Type
▾ EPC2000_1	0	0			EPC2000 IO Device
▸ Interface	0	0 X1			EPC2000
Fieldbus Input I/O Gateway_1	0	1	68...195		Fieldbus Input I/O Gateway
Fieldbus Output I/O Gatewa...	0	2		64...191	Fieldbus Output I/O Gateway

Once the Cyclic (IO Data) Data Exchange settings and Fieldbus gateway have been configured using the configuration tool, it is necessary to compile the configuration and download it to the PROFINET IO Controller (PLC) of the PROFINET network.



The cyclic exchanges will then start between the IO Devices once the PROFINET network is activated.

Note: The IO cycle can be adjusted from 64ms up to 512ms.

Acyclic Data Exchange (Record Data)

Acyclic data exchange (or Record Data) is used to transfer data that does not require continuous updates.

It is possible to access any parameter in the EPC2000 Programmable Controller controller unit by this means, regardless of whether it has been included in the PROFINET input/output data assembly.

Acyclic data are transmitted via UDP/IP with the RPC protocol. For this, PROFINET 0provides “Read” and “Write” services of data.

For addressing Record Data services, the combination of API/Slot/Subslot/Index values is used. The Modbus address of the EPC2000 Programmable Controller parameter to read or write is passed through the Index value.

Modbus addresses are listed in the iTools Parameter Explorer.

PROFINET Acyclic Readings

This section describes how to access a variable using PROFINET in acyclic mode.

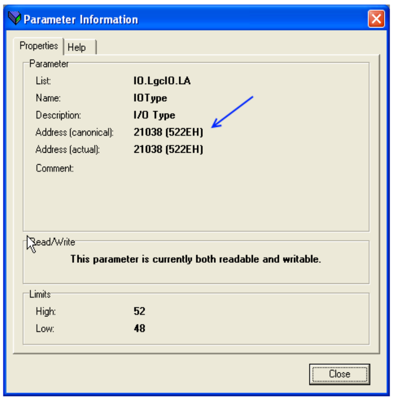
PROFINET uses the following parameter to access a variable in acyclic mode:

- API
- The Slot and Subslot
- Index

To access a parameter in acyclic mode, you first need to know its Modbus address. This may be accessed by selecting the parameter from the Parameter Explorer list shown in Address column.

The figure below shows an alternative way to access a parameter. This uses the Graphical Wiring Editor. The Modbus address is shown in Address column.

Right click on the parameter to open the parameter help window.



List: IO.LgcIO.LA
Name: IOType
Description: I/O Type
Address (canonical): 21038 (522EH)
Address (actual): 21038 (522EH)
Comment:

This parameter is currently both readable and writable.

High: 52
Low: 48

The API is always 0 (Zero):

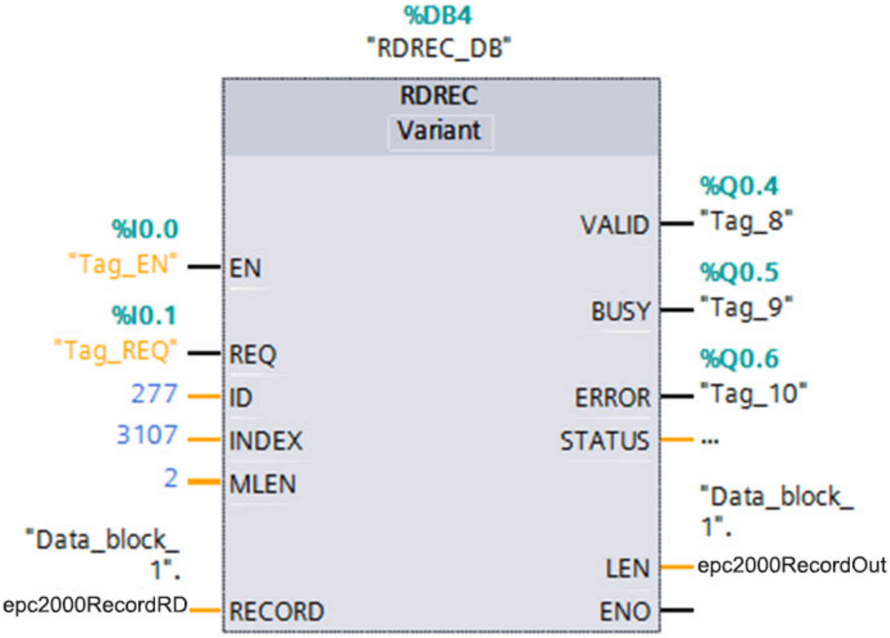
- The Slot is always 1 (One)
- The Subslot is always 1 (One)
- The Index will be the Modbus-address you found previously in iTools

Acyclic data exchanges, Step 7 (TIA Portal) program block

The RDREC and the WRREC function blocks are used to respectively read and write the record data, which allows access to the overall parameters of EPC2000.

The Modbus address of the parameter to be read is set in the INDEX entry, the ID value has to match with the Hardware ID of your device incremented by one.

See example below of the Fieldbus Input IO Gateway slot, where the Modbus address is 3107 and the HW ID is 277 when reading one EPC2000 parameter, using RDREC function block of STEP 7.



Constraints on the Parameters

The parameter in acyclic-mode follows the same limitation as the parameters in the Fieldbus I/O gateway: 16 bits length and they follow the same scaling, "Configuring the Cyclic (IO Data) Data Exchange" on page 270

Data Formats

Data is returned as 'scaled integers', such that 999.9 is returned or sent as 9999; 12.34 is encoded as 1234. The control program in the PROFINET client (master) must convert the numbers into floating point values if required.

The GSD File

The PROFINET GSDML (General Stations Description) file for EPC2000 Programmable Controller controller has the naming rule *GSDML-V2.41-Eurotherm-EPC2000-yyyymmdd.xml* and is available from your supplier, or electronically by going to Web site https://www.eurotherm.com/en.

The GSD file is designed to automate the PROFINET network configuration process by precisely defining the required device parameter information. Software configuration tools utilise the GSD file to configure a PROFINET network.

Alarm Notification

EPC2000 Programmable Controller has the ability to send "Alarm Notification" when an alarm is raised; the I/O Controller acknowledges this Alarm Notification request (e.g. "Instrument in Standby" if the device is in Standby mode).

The Alarms are configured to the Input Fieldbus IO Gateway module, which is connected to Slot 1.

The EPC2000 Programmable Controller uses the “Channel Diagnosis” to transmit its Diagnosis Alarm, which is made of a single 16 Bit “ErrorType” for each alarm, defined in the “Manufacturer Specific” range (0x0100-0x7FFF), which starts at 0x0200 (512d) for EPC2000 (e.g. 512 stands for “Alarm 1 State”, 513 for “Alarm 2 State”, and so on).

The definition of the different “ErrorType” is provided in the “human readable” format in the GSDML file and are summarized in the table below.

When an Alarm Notification is raised, it is managed by the IO Controller which hands it over to its alarm buffer. If this alarm disappears, EPC2000 sends a new request to the IO Controller to remove it from its alarm buffer.

EPC2000 can treat several alarms simultaneously, but the Profinet Alarm Buffer size is limited to two entries, when an entry is freed and another alarm is still present in EPC2000, this second one is sent to the IO Controller.

As described above, the Alarms are explicitly described in the GSDML file, an additional field is used to provide the first level of suggested action to manage this alarm.

Error Type	EPC2000 meaning
512	Alarm 1 Status
513	Alarm 2 Status
514	Alarm 3 Status
515	Alarm 4 Status (Alarms 5 & 6 managed below)
517	Sensor Break (Analog Input)
518	Loop Break (Loop control issue)
520	Auto Tune (EPC2000 in Auto Tune mode)
522	PV1 out of range
528	Standby Mode
529	Alarm 5 Status
530	Alarm 6 Status

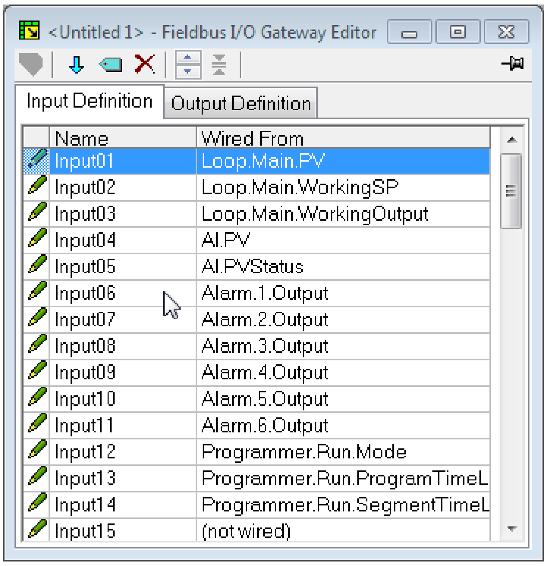
Table 1: EPC2000's Profinet Alarm list (see GSDML file for details)

Fieldbus I/O Gateway

The EPC2000 controller contains a number of parameters and some protocols, such as EtherNet/IP, PROFINET need a way of configuring a few selected parameters to exchange Input and Output data across a network.

The Fieldbus I/O tool available in iTools allows the configuration of an Input and Output table definition which can be used by the relevant protocol for I/O communications.

Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to that shown below:



Name	Wired From
Input01	Loop.Main.PV
Input02	Loop.Main.WorkingSP
Input03	Loop.Main.WorkingOutput
Input04	AI.PV
Input05	AI.PVStatus
Input06	Alarm.1.Output
Input07	Alarm.2.Output
Input08	Alarm.3.Output
Input09	Alarm.4.Output
Input10	Alarm.5.Output
Input11	Alarm.6.Output
Input12	Programmer.Run.Mode
Input13	Programmer.Run.ProgramTimeL
Input14	Programmer.Run.SegmentTimeL
Input15	(not wired)

By default, the Input and Output Definition tables are configured with the most frequently used parameters.

There are two tabs within the editor, one for the definition of Inputs, and the other for Outputs.

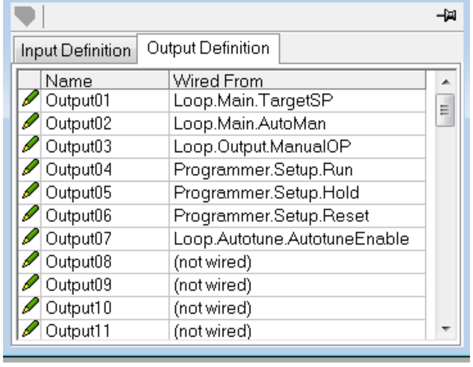
'Inputs' are values read from the EPC2000 controller and sent out for example, alarm status information or measured values, i.e. they are readable values.

Note: The Input and Output buffer should not be empty. At least one parameter should be selected so the data cyclic exchange functions correctly.

'Outputs' are values received from the client (master) and are written to the controller, for example, setpoints written from the client (master) to the controller. Input and Output parameter values are read and written cyclically. The frequency of the I/O data exchange is determined by the Requested Packet Interval (RPI) which is set by the client (master) device. (For details regarding communication protocols and their associated RPI see "I/O and communication types" on page 301).

The procedure for selecting and replacing variables is the same for both input and output tabs. Double click the row to be edited in the input or output table and select the variable to assign to it. A pop-up provides a browser from which a list of parameters can be selected. Double click the parameter to assign it to the selected row.

Note: You should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it.



Name	Wired From
Output01	Loop.Main.TargetSP
Output02	Loop.Main.AutoMan
Output03	Loop.Output.ManualOP
Output04	Programmer.Setup.Run
Output05	Programmer.Setup.Hold
Output06	Programmer.Setup.Reset
Output07	Loop.Autotune.AutotuneEnable
Output08	(not wired)
Output09	(not wired)
Output10	(not wired)
Output11	(not wired)

When the definition tables are populated with the variables required, note how many 'wired' entries are included in the input and output areas as this will be needed when setting up the I/O scanner.

Input and Output Parameters are 16 bits (2 bytes) each. In the example above, there are 16 input parameters (32 bytes) and 7 output parameters (14 bytes), so a total of 46 bytes of data. Make a note of this number, as it is required when setting the I/O length when configuring the I/O scanner.

Notes:

1. 32-bit float and 32-bit time parameters can also be configured on the Input and Output tables by adding the same parameter in consecutive rows.
2. All the parameters in the Input table are assumed to be readable and Output table writeable. If going through the Input/Output tables during I/O messaging a parameter is not readable/writeable, the read/write aborts. Read parameter values are sent along with 0 values for parameters that were not read.

   Note: For EtherNet/IP - if the table read or write is aborted, the EtherneNet/IP diagnostic parameter Comms>Option>EtherNetIP>EIP_ModuleStatus will show a value of ErrorDetected(3).

Once the changes have been made to the Input and Output definitions, they must be downloaded to the EPC2000 Programmable Controller. This is done with the button on the top left of the Fieldbus I/O Gateway Editor marked:

Note: iTools can place the EPC2000 controller in and out of Configuration mode while downloading Fieldbus I/O Gateway changes.

Input Linearization (LIN16)

The linearization block converts an analog input into an analog output through a user-defined table. This linearization table consists of a series of 16 points defined by input breakpoints (In1 to In16) and output values (Out1 to Out16). In other words, the linearization block implements a piecewise linear curve (a connected sequence of line segments) defined by a series of input coordinates (In1 to In16) and associated output coordinates (Out1 to Out16).

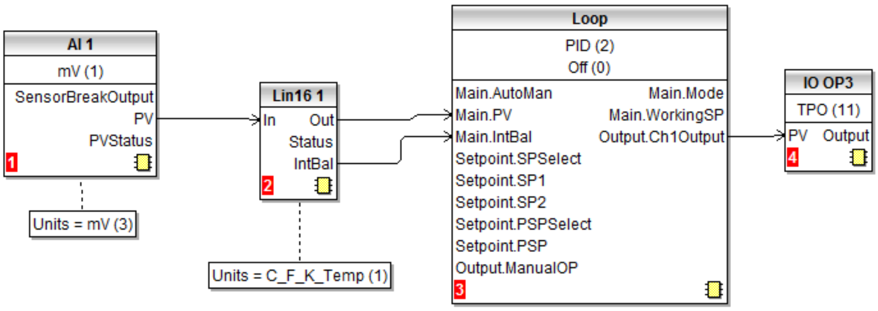
Two of the most typical applications for the LIN16 function block are:

1. Custom linearization of a sensor input.
2. Adjustment of the process variable to account for differences introduced by the overall measurement system or to derive a different process variable.

Custom Linearization

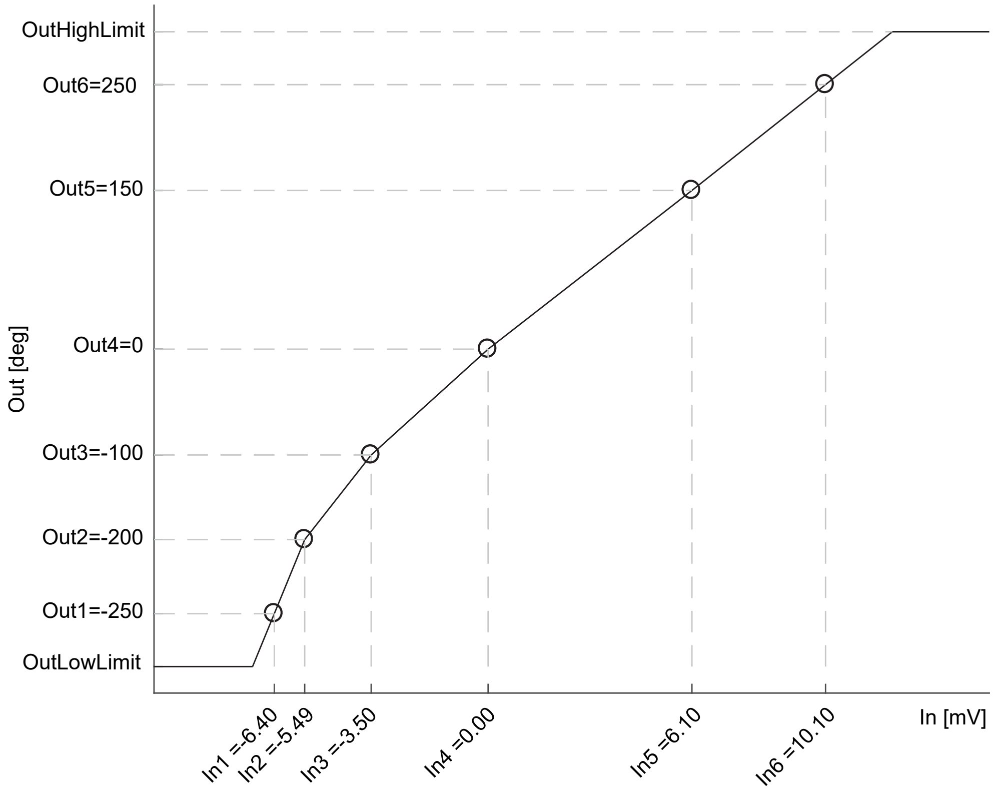
This application allows the user to create their own linearization table.

In the following example the LIN16 block is placed between the Loop block and an Analog Input set to linear and Linearization Type to mV, V, mA, Ohms, etc. In the following example the AI block is set to mV.



The following graph shows a typically increasing linearization curve. The decision of the actual number of points depends of the required accuracy in converting the input electrical signal into the required output value: the higher the number of points, the higher accuracy can be obtained; conversely a lower number of points requires less time to configure the function block. If less than 16 points are used, set the 'NumPoints' parameter to the required number. Points not selected will then be ignored, the curve will continue in a straight line fit to the levels set in 'OutHighLimit' or 'OutLowLimit' and the 'CurveForm' output will be 'Increasing'.

Example 1: Custom Linearization - Increasing Curve

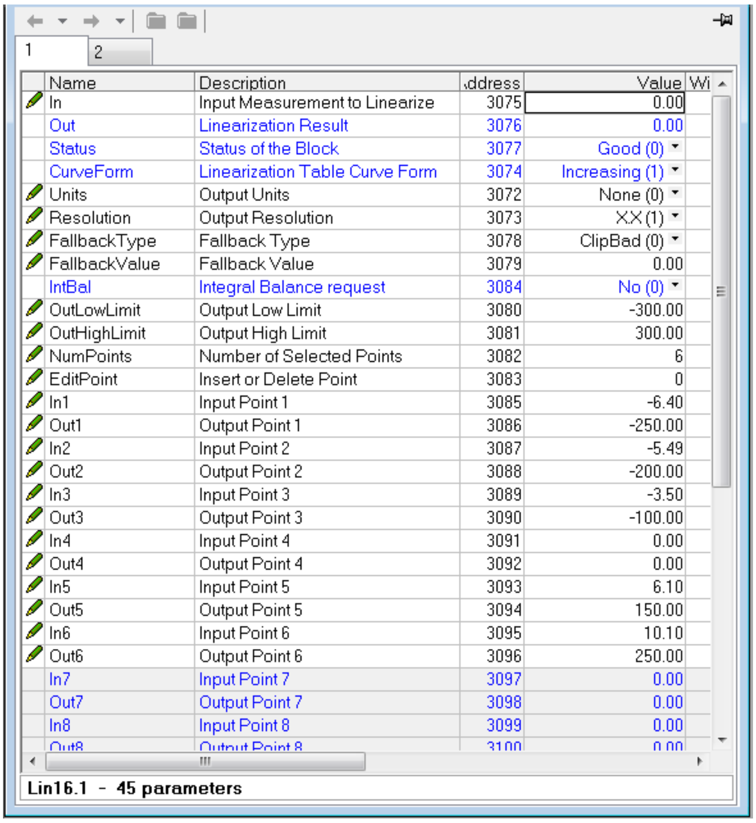


To Setup the Parameters

1. Set the appropriate Fallback type and value, Output units and resolution (editable only in Config mode); Units and resolution of the input and the input breakpoints will be derived by the source wired to 'In'.
2. Set the ‘OutHighLimit’ and ‘OutLowLimit’ to restrict the output of the linearization curve. The ‘OutHighLimit’ must be greater than the ‘OutLowLimit’.
3. Set the 'NumPoints' (6 in this example) to the required number of points for the linearization table. This is an important and required step and the effects of skipping it are reported in the Example 2.
4. Enter values of the first Input breakpoint ‘In1’ and Output value ‘Out1’.
5. Continue with the remaining Input breakpoints and Output values.
6. Wire the ‘IntBal’ parameter to the ‘Loop.Main.IntBal’ parameter. This prevents any proportional or derivative kick in the controller output when any change occurs in the LIN16 configuration parameters.

Points on the linearization curve can be derived from reference tables or can be found by associating the measurements of an external reference (e.g. temperature in degrees Celsius) to the AI electrical readings (e.g. mV or mA).

The iTools view reproduced below shows how the parameters are set up in LIN block 1 for the above example. The list corresponds to the parameters shown in iTools, see section "Linearization Block Parameters" on page 133. Parameter help is also available by right clicking the parameter in the iTools list.



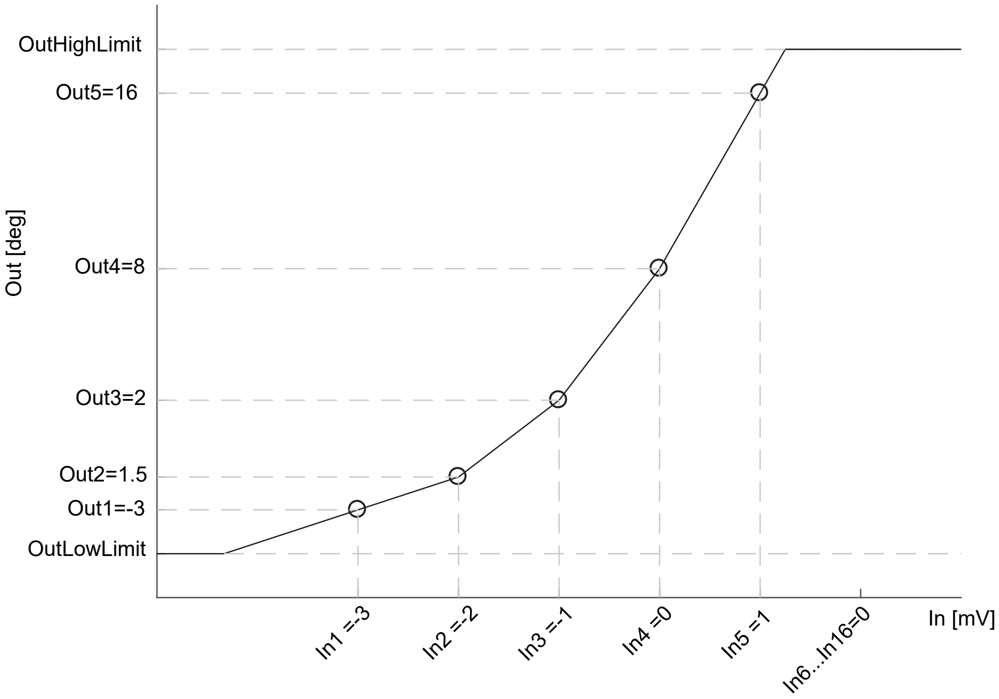
	Name	Description	ddress	Value	Wi
✎	In	Input Measurement to Linearize	3075	0.00	
	Out	Linearization Result	3076	0.00	
	Status	Status of the Block	3077	Good (0)	
	CurveForm	Linearization Table Curve Form	3074	Increasing (1)	
✎	Units	Output Units	3072	None (0)	
✎	Resolution	Output Resolution	3073	X.X (1)	
✎	FallbackType	Fallback Type	3078	ClipBad (0)	
✎	FallbackValue	Fallback Value	3079	0.00	
	IntBal	Integral Balance request	3084	No (0)	
✎	OutLowLimit	Output Low Limit	3080	-300.00	
✎	OutHighLimit	Output High Limit	3081	300.00	
✎	NumPoints	Number of Selected Points	3082	6	
✎	EditPoint	Insert or Delete Point	3083	0	
✎	In1	Input Point 1	3085	-6.40	
✎	Out1	Output Point 1	3086	-250.00	
✎	In2	Input Point 2	3087	-5.49	
✎	Out2	Output Point 2	3088	-200.00	
✎	In3	Input Point 3	3089	-3.50	
✎	Out3	Output Point 3	3090	-100.00	
✎	In4	Input Point 4	3091	0.00	
✎	Out4	Output Point 4	3092	0.00	
✎	In5	Input Point 5	3093	6.10	
✎	Out5	Output Point 5	3094	150.00	
✎	In6	Input Point 6	3095	10.10	
✎	Out6	Output Point 6	3096	250.00	
	In7	Input Point 7	3097	0.00	
	Out7	Output Point 7	3098	0.00	
	In8	Input Point 8	3099	0.00	
	Out8	Output Point 8	3100	0.00	

Lin16.1 - 45 parameters

The function block will automatically skip points that do not follow strictly monotonically increasing order of the 'In' coordinates. If at least one point has been skipped the 'CurveForm' parameter will show 'SkippedPoints'. If no valid interval is found the 'CurveForm' parameter will show 'NoForm' and the Fallback strategy will be applied. Other conditions when the Fallback strategy is applied are input source bad status (for example, sensor break or sensor over-range) and calculated LIN16 output over-range (that is, less than OutLowLimit or greater than InHighLimit).

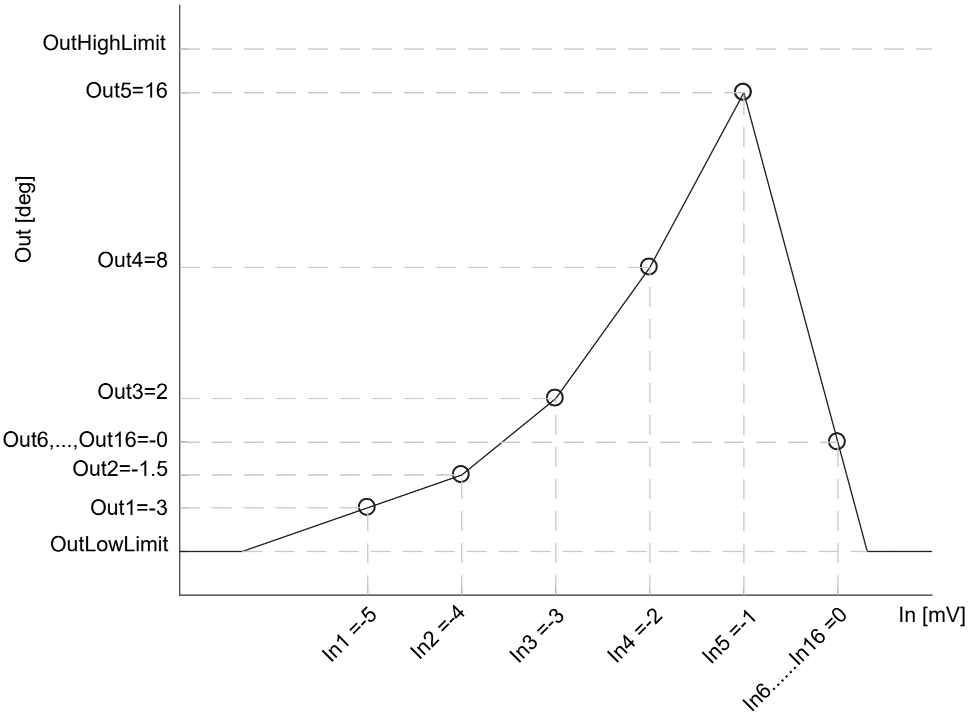
Example 2: Custom Linearization - Skipped Points Curve

If points defaulted to zero have not been disabled, by reducing 'NumPoints', - AND assuming that at least one of the previous input breakpoints is positive (see the curve below) - then those points will be automatically skipped. The output characteristics will be the same as those obtained by disabling the points defaulted to zero but the 'CurveForm' will be 'SkippedPoints'.



In1 to In5 will be used. In6 to In16 will be ignored. 'CurveForm' will be 'SkippedPoints'

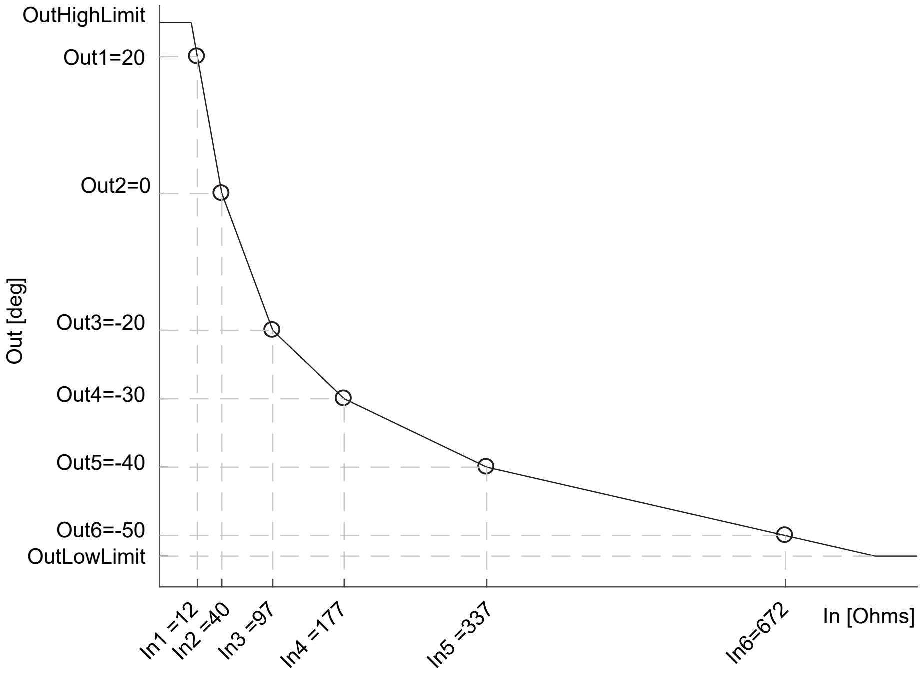
However, when the 'CurveForm' parameter is 'SkippedPoints' (because the number of points 'NumPoints' has not been reduced to the required set) it is not guaranteed that the output characteristics will be increasing or decreasing. In fact, for example, if the input breakpoints are all negative and the final points are zero, then the first "zero" point will be included in the characteristics - see the following curve. Therefore, always set 'NumPoints' to the required value to get the expected sensor linearization curve type - increasing, decreasing or free form.



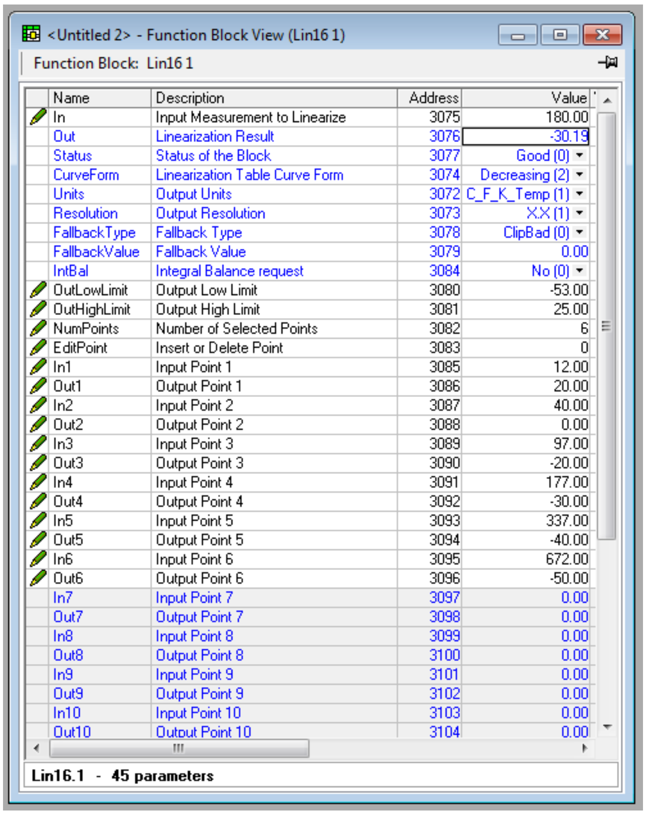
In1 to In5 will be used as well as In6, possibly resulting in a not expected curve. In7, ..., In16 will be ignored. CurveForm will be SkippedPoints.

Example 3: Custom Linearization - Decreasing Curve

The curve may also be a decreasing form as shown below.



The procedure to setup the parameters is the same as in the previous example.



Function Block: Lin16 1

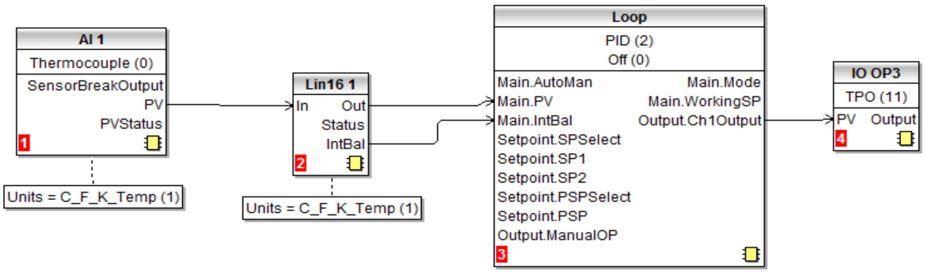
	Name	Description	Address	Value
✎	In	Input Measurement to Linearize	3075	180.00
	Out	Linearization Result	3076	-30.19
	Status	Status of the Block	3077	Good (0)
	CurveForm	Linearization Table Curve Form	3074	Decreasing (2)
	Units	Output Units	3072	C_F_K_Temp (1)
	Resolution	Output Resolution	3073	X.X (1)
	FallbackType	Fallback Type	3078	ClipBad (0)
	FallbackValue	Fallback Value	3079	0.00
	IntBal	Integral Balance request	3084	No (0)
✎	OutLowLimit	Output Low Limit	3080	-53.00
✎	OutHighLimit	Output High Limit	3081	25.00
✎	NumPoints	Number of Selected Points	3082	6
✎	EditPoint	Insert or Delete Point	3083	0
✎	In1	Input Point 1	3085	12.00
✎	Out1	Output Point 1	3086	20.00
✎	In2	Input Point 2	3087	40.00
✎	Out2	Output Point 2	3088	0.00
✎	In3	Input Point 3	3089	97.00
✎	Out3	Output Point 3	3090	-20.00
✎	In4	Input Point 4	3091	177.00
✎	Out4	Output Point 4	3092	-30.00
✎	In5	Input Point 5	3093	337.00
✎	Out5	Output Point 5	3094	-40.00
✎	In6	Input Point 6	3095	672.00
✎	Out6	Output Point 6	3096	-50.00
	In7	Input Point 7	3097	0.00
	Out7	Output Point 7	3098	0.00
	In8	Input Point 8	3099	0.00
	Out8	Output Point 8	3100	0.00
	In9	Input Point 9	3101	0.00
	Out9	Output Point 9	3102	0.00
	In10	Input Point 10	3103	0.00
	Out10	Output Point 10	3104	0.00

Lin16.1 - 45 parameters

Adjustment of the Process Variable

This application allows the user to compensate for known inaccuracies introduced by the overall measurement system. This not only includes the sensor but also the overall measurement chain. Furthermore, this can also be used to derive a different process variable, for instance, a temperature measured in a different place from where the actual sensor is positioned. The adjustment is made directly on the value, and in the units, of the process variable measured by the controller.

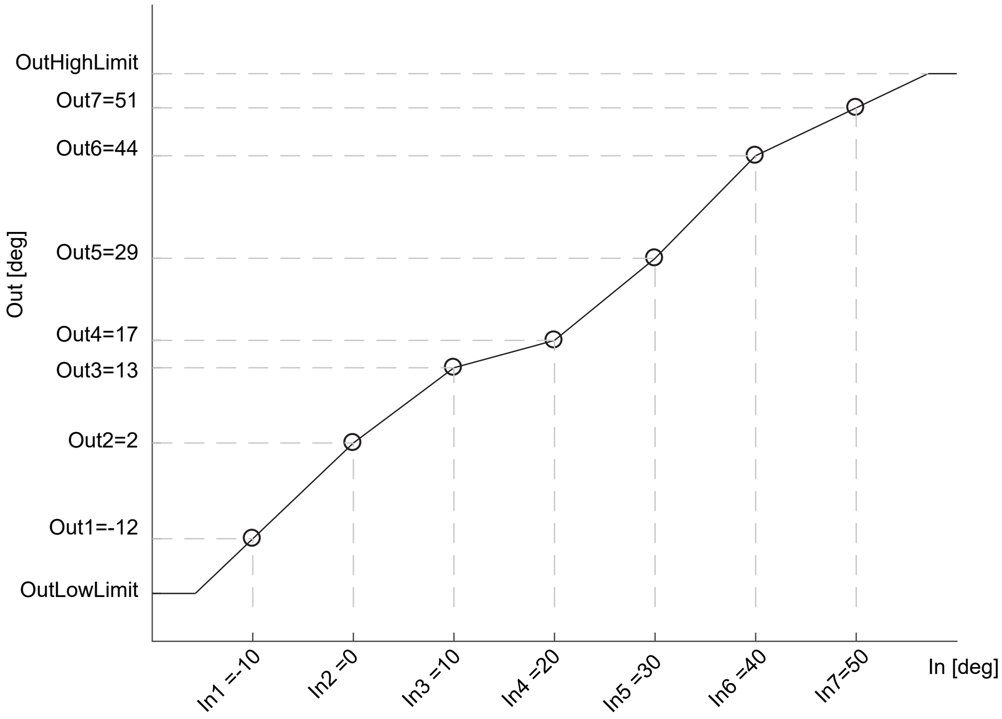
The process variable can be adjusted in different operating conditions (for example, different temperatures), by using the LIN16 multiple point adjustment curve: this extends the simple PV Offset feature present in the AI block, which just adds or subtracts a single value to the measured PV in all operating conditions.



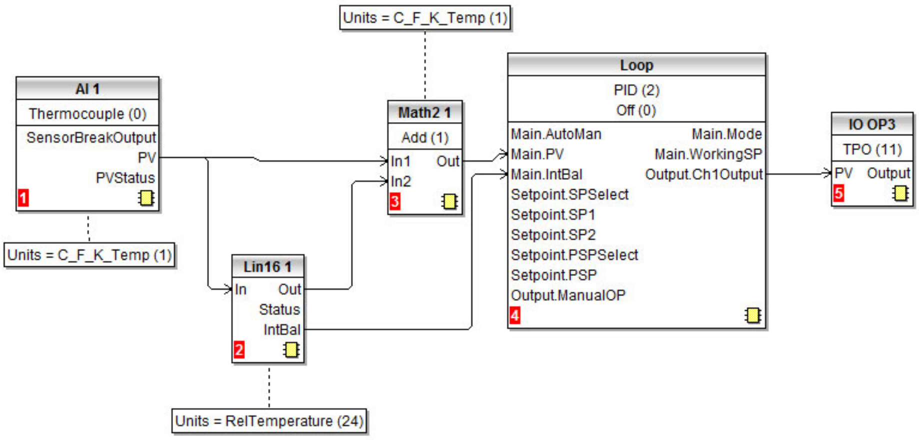
Two alternative configurations can be used:

In the first case the LIN16 table contains the process variable values ‘In1’ to ‘In16’, measured by the controller, and the reference values, ‘Out1’ to ‘Out16’, measured by an external reference.

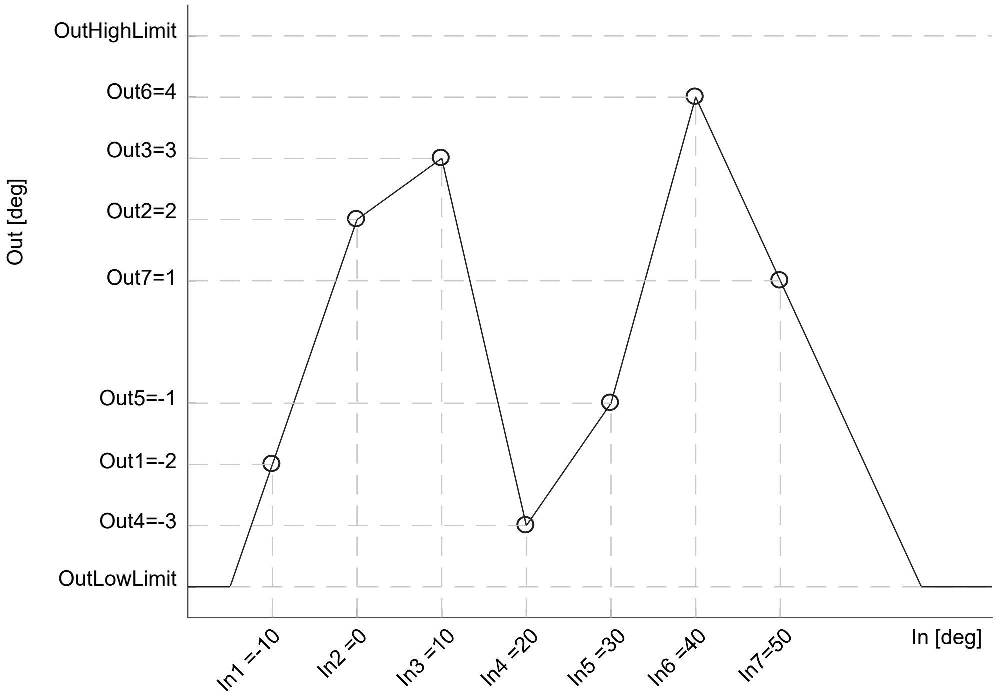
An example is shown below. The same setup procedure detailed before also applies here apart from the different configuration of the AI block. As shown in the graph and in the wiring diagram, the units of both the input and the output of LIN16 are absolute temperatures.



In the second case, for the same application, the LIN16 table stores the offsets between the process variable values measured in the controller and a Math block, set to Add, placed between the Analog Input (AI) and the Loop block. The adjustment is made by adding the offset calculated by the LIN16 block to the measured process variable. In the case of temperature adjustment (and differently from the previous case) the output units of LIN16 should be set to relative temperature. This is in order to select the correct conversion equation when a temperature units change is applied to the offsets (e.g. from degrees Celsius to Fahrenheit).



Because offsets do not follow in general a continuously increasing or decreasing trend, then the ‘CurveForm’ parameter will be ‘FreeForm’, ‘Increasing’ or ‘Decreasing’ depending on their values: see the following graph as an example of a free form offset curve.



Both the two above mentioned configurations provide the control Loop function block with the same adjusted PV. The values are reported in table for the two examples. The high values of the offsets are only to accentuate in the pictures the action of the adjustment.

Input Breakpoints	Output values: absolute temperature	Alternative output values: relative temperature
-10 deg	-12 deg	-2 deg
0 deg	2 deg	2 deg
10 deg	13 deg	3 deg
20 deg	17 deg	-3 deg
30 deg	29 deg	-1 deg
40 deg	44 deg	4 deg
50 deg	51 deg	1 deg

User Calibration

The controller is calibrated during manufacture using traceable standards for every input range. It is, therefore, not necessary to calibrate the controller when changing ranges. Furthermore, the use of a continuous automatic zero correction of the input helps ensure that the calibration of the instrument is optimised during normal operation.

To comply with statutory procedures such as the Heat Treatment Specification AMS2750, the calibration of the instrument can be verified and re-calibrated if considered necessary in accordance with the instructions given in this chapter.

User calibration allows the controller to be calibrated at any part of its range (not just span and zero) or to allow for fixed known measuring offsets such as sensor tolerances.

Factory calibration is stored within the controller and can be returned to at any time.

In some cases it is only required to calibrate the controller itself, however, it is often necessary to compensate for tolerances in the sensor and its connections as well. This is particularly true for temperature measurement which uses, typically, thermocouple or PRT sensors. In the latter case this may be done using an ice cell or hot bath or dry block calibrator. The different methods are described in the following sections.

Controller only Calibration

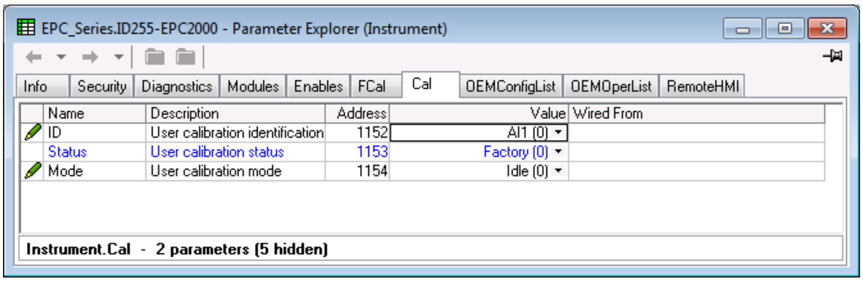
To Calibrate the Analog Input

This can be done using iTools. The following points should be observed:

- Allow at least 10 minutes for the controller to settle after switching on.
- Connect the input of the controller to a millivolt source. If the controller is configured for thermocouple ensure that the millivolt source is set to the correct CJC compensation for the thermocouple in use and that the correct compensating cable is used.
- If the input to be calibrated is mV, mA or volts the measurement will be linear mV, mA or volts. If it is configured for thermocouple or RTD the measurement will be in degrees in accordance with the instrument configuration.

Using iTools

Open the Instrument.Cal function.



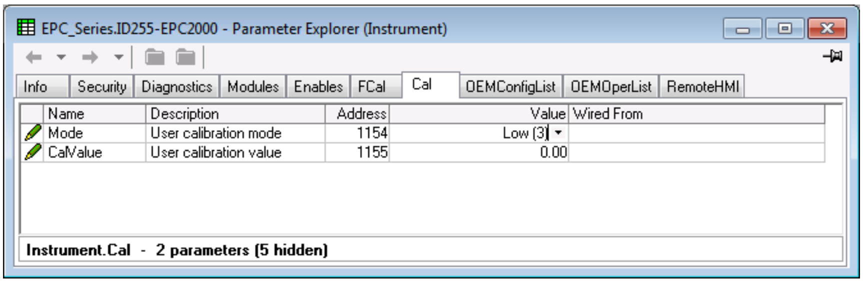
Name	Description	Address	Value	Wired From
ID	User calibration identification	1152	AI1 (0)	
Status	User calibration status	1153	Factory (0)	
Mode	User calibration mode	1154	Idle (0)	

Instrument.Cal - 2 parameters (5 hidden)

The status parameter will show ‘Factory’ if user calibration has not been carried out before.

To Start User Calibration

Click the ‘Mode’ parameter and select ‘Start’.



Name	Description	Address	Value	Wired From
Mode	User calibration mode	1154	Low (3)	
CalValue	User calibration value	1155	0.00	

Instrument.Cal - 2 parameters (5 hidden)

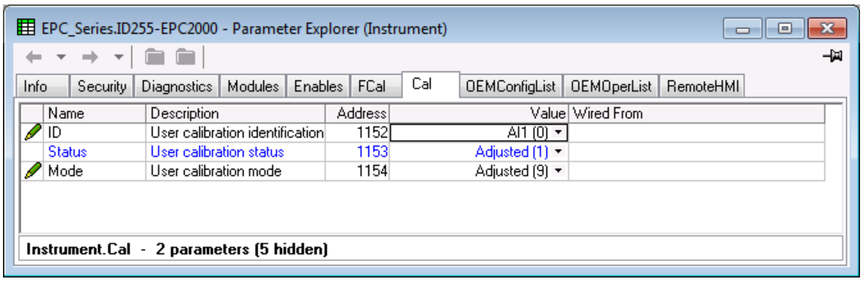
The Mode will change to ‘Low’.

1. In ‘CalValue’ enter a value which represents the low reading required on the controller display, in this case 0.00
2. Set the mV source to 0.00mV. If the input is a thermocouple make sure that the mV source is set to compensate for the type of thermocouple configured. It is not necessary to calibrate for other thermocouple types.
3. In ‘Mode’ select ‘SetLow’. This will calibrate the controller to the selected input mV (0.00). Discard will return to Factory calibration.

The ‘Mode’ will change to ‘High’.

1. In ‘CalValue’ enter a value which represents the high reading required on the controller display, in this case 300.00.
2. Set the mV source to the correct input level. If the input is a thermocouple this will be the mV equivalent to 300.00°C. It is not necessary to calibrate for other thermocouple types.
3. In ‘Mode’ select ‘SetHigh’. This will calibrate the controller to the selected input mV. Discard will return to Factory calibration.

The ‘Status’ and ‘Mode’ will show ‘Adjusted’ which indicates that the controller has been calibrated by the user.



Name	Description	Address	Value	Wired From
ID	User calibration identification	1152	AI1 (0)	
Status	User calibration status	1153	Adjusted (1)	
Mode	User calibration mode	1154	Adjusted (9)	

Instrument.Cal - 2 parameters (5 hidden)

It may be found useful to open the AI function block when carrying out the calibration since the PV can be read directly during the calibration procedure. This also allows the input measurement settling to be viewed during the calibration process.

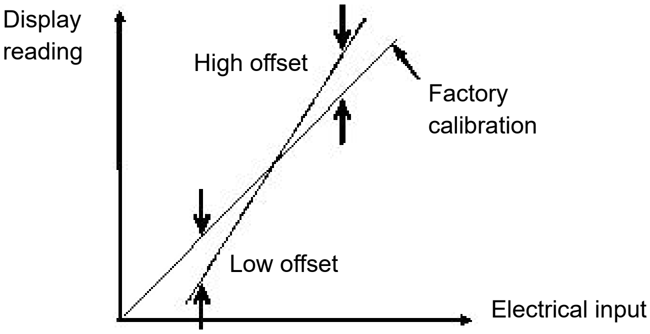
Note: If, at the end of a calibration process, the calibration has been unsuccessful the Status will return to Factory and the Mode will show ‘Unsuccessful’.

To Return to Factory Calibration

In ‘Mode’ drop-down select ‘Discard’.

Two Point Offset Calibration

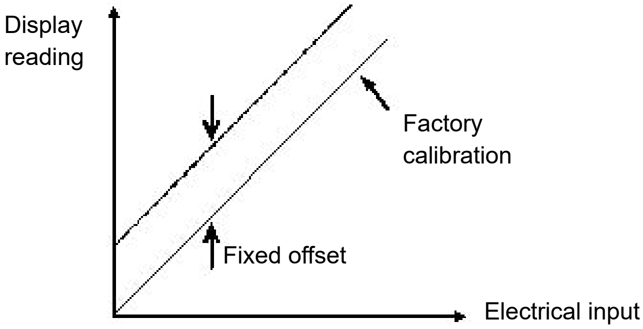
A two point offset enables the controller display to be offset by different amounts at the low end of the scale and at the high end of the scale. The basic calibration of the controller is unaffected but the two point offset provides a compensation for sensor or inter-connection errors. The diagrams below show that a line is drawn between the low and high offsets values. Any readings above and below the calibration points will be an extension of this line. For this reason it is good practice to calibrate with the two points as far apart as possible.



The procedure is exactly the same as shown in the previous section. For the minimum input set the ‘CalValue’ to the reading required on the controller as shown in the Low offset in the diagram above.

Similarly, for the maximum input set the ‘CalValue’ to the reading required on the controller as shown in the High offset in the above diagram.

Note: A ‘PvOffset’ parameter is available in the Analog Input list which provides a fixed value to be added to or subtracted from the process variable. This is not part of the User Calibration procedure but applies a single offset over the full display range of the controller. It has the effect of moving the curve up a down about a central point as shown in the example below:



Calibration using a Dry Block or Equivalent

A dry block, ice cell or hot bath are heated or cooled to a specific temperature and accurately maintained at that temperature. Calibration is a comparison between two devices. The first device is the unit to be calibrated, often called the unit under test. The second device is the standard, which has a known accuracy. Using the standard as a guide, the unit under test is adjusted until both units display the same results while exposed to the same temperature. Using this method the tolerance of the temperature sensor, CJC, etc is included in the calibration.

The procedure is essentially the same as that already described but the millivolt source is replaced by the temperature sensor under test.

OEM Security

OEM security allows users, typically OEMs or distributors, to help protect their intellectual property and is designed to help prevent unauthorized viewing, reverse engineering or cloning of controller configurations. This protection includes application-specific internal (soft) wiring and limited access to certain parameters via comms (by iTools or a third party comms package). OEM Security is available as an orderable option and is enabled through Feature Security ("Instrument.Security" on page 95).

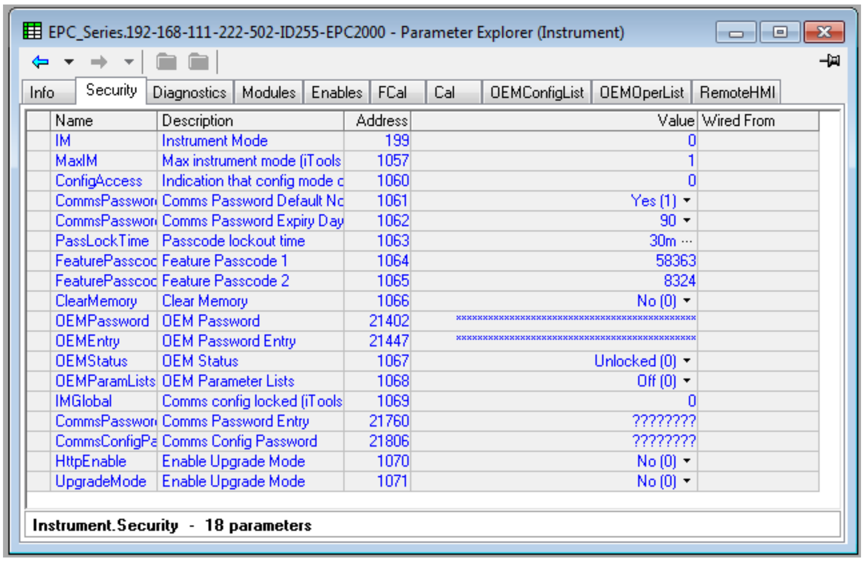
When ‘OEM Security’ is enabled, users are prevented from accessing soft wiring from any source, and it is not possible to Load or Save the configuration of the instrument via iTools or by using the Save/Restore facility.

Altering configuration and/or operator parameters via an external HMI or Comms may also be restricted when OEM security is implemented.

Once the security function has been set up for a particular application it may be cloned into every other identical application without further configuration.

Implementation

OEM Security parameters are displayed in the ‘Instrument - Security’ function block.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

Name	Description	Address	Value	Wired From
IM	Instrument Mode	199	0	
MaxIM	Max instrument mode (iTools	1057	1	
ConfigAccess	Indication that config mode c	1060	0	
CommsPasswor	Comms Password Default Nc	1061	Yes (1) ▾	
CommsPasswor	Comms Password Expiry Day	1062	90 ▾	
PassLockTime	Passcode lockout time	1063	30m ···	
FeaturePasscoc	Feature Passcode 1	1064	58363	
FeaturePasscoc	Feature Passcode 2	1065	8324	
ClearMemory	Clear Memory	1066	No (0) ▾	
OEMPassword	OEM Password	21402	**	
OEMEntry	OEM Password Entry	21447	**	
OEMStatus	OEM Status	1067	Unlocked (0) ▾	
OEMParamLists	OEM Parameter Lists	1068	Off (0) ▾	
IMGlobal	Comms config locked (iTools	1069	0	
CommsPasswor	Comms Password Entry	21760	????????	
CommsConfigPa	Comms Config Password	21806	????????	
HttpEnable	Enable Upgrade Mode	1070	No (0) ▾	
UpgradeMode	Enable Upgrade Mode	1071	No (0) ▾	

Instrument.Security - 18 parameters

OEMPassword This password is selected by the OEM. Any alpha/numeric text can be used and the field is editable whilst the OEM Status is ‘Unlocked’. A minimum of eight characters should be used. It is not possible to clone the OEM Security Password. (Highlight the complete row before entering).

OEMEntry Enter the OEM security password to enable and disable OEM security. The controller must be in configuration level to enter this password. When the correct password is entered the OEM Status will toggle between ‘Locked’ and ‘Unlocked’. (Highlight the complete row before entering). Five login attempts are allowed before lockout which is followed by a 90 minute password lockout period.

OEMStatus Read-only showing ‘Locked’ or ‘Unlocked’.

If Unlocked two lists are available (OEMConfigList and OEMOperList) which allow an OEM to restrict which parameters are alterable when the controller is in Operator and Configuration Access levels.

If the ‘OEMStatus’ is ‘Locked’ these two lists are not shown. The controller configuration cannot be cloned and the internal wiring cannot be accessed via comms.

OEMParameterLists This parameter is only writeable when the ‘OEM Status’ is ‘Unlocked’.

When ‘Off’, Operator type parameters are alterable in Operator access level and Config parameters are alterable in Configuration access level (all within other limitations such as high and low limits).

When ‘On’, parameters added to the OEMConfigList WILL be available to the operator when the controller is in configuration level. Parameters not added in this list will not be available to the operator. Parameters added to the OEMOperList will NOT be available to the operator when the controller is in Operator access level.

The table at the end of this section shows an example for just two parameters ‘Alarm 1 Type’ (configuration type parameter) and ‘Alarm 1 Threshold’ (operator type parameter).

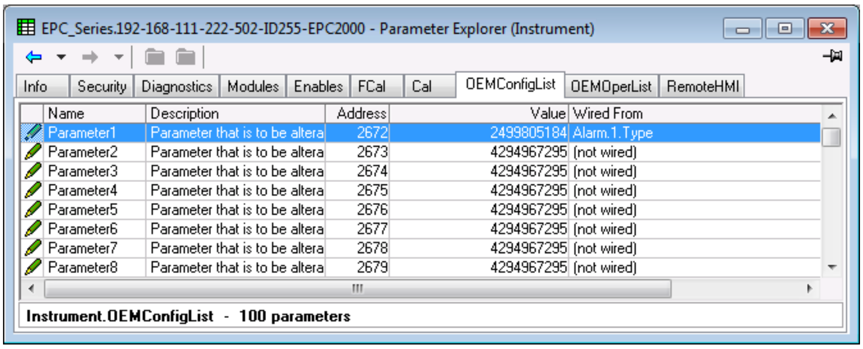
Note: When entering or exiting OEM Security a few seconds should be allowed for iTools to synchronize.

OEM Configuration List

The ‘OEMConfigList’ allows the OEM to choose up to 100 configuration parameters which are to remain Read/Write while in Configuration level and OEM Security is enabled (locked). In addition to these the following parameters are writeable in configuration mode:

OEM Security Password Entry, Comms Configuration Passcode, Controller Coldstart.

The required parameters may be by dragged and dropped from a browser list (on the left hand side) into the Wired From cell in the ‘OEMConfigList’. Alternatively, double-click into the ‘WiredFrom’ cell and select the parameter from the pop-up list. These parameters are those chosen by the OEM which are to remain alterable when OEM Security is enabled and the controller is in Configuration access level.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

Name	Description	Address	Value	Wired From
Parameter1	Parameter that is to be altera	2672	2499805184	Alarm.1.Type
Parameter2	Parameter that is to be altera	2673	4294967295	(not wired)
Parameter3	Parameter that is to be altera	2674	4294967295	(not wired)
Parameter4	Parameter that is to be altera	2675	4294967295	(not wired)
Parameter5	Parameter that is to be altera	2676	4294967295	(not wired)
Parameter6	Parameter that is to be altera	2677	4294967295	(not wired)
Parameter7	Parameter that is to be altera	2678	4294967295	(not wired)
Parameter8	Parameter that is to be altera	2679	4294967295	(not wired)

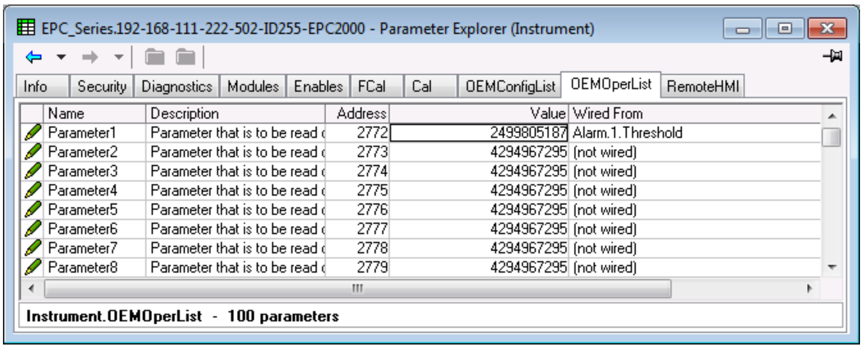
Instrument.OEMConfigList - 100 parameters

The view shows the first eight parameters of which Parameter 1 has been populated with a configuration parameter (Alarm 1 Type). Examples of configuration parameters include Alarm Types, Input Types, Range Hi/Lo, etc.

When the OEM Status is Locked, this list is not shown.

OEM Operator List

The OEM Operator List operates in the same way as the OEM Configuration list except the parameters selected are those which are available in Operator access level. Examples are programmer mode, alarm setting parameters, etc. The example below shows ‘Alarm 1 Threshold’ which is to be read only in Operator access level.



EPC_Series.192-168-111-222-502-ID255-EPC2000 - Parameter Explorer (Instrument)

Name	Description	Address	Value	Wired From
Parameter1	Parameter that is to be read (	2772	2499805187	Alarm.1.Threshold
Parameter2	Parameter that is to be read (	2773	4294967295	(not wired)
Parameter3	Parameter that is to be read (	2774	4294967295	(not wired)
Parameter4	Parameter that is to be read (	2775	4294967295	(not wired)
Parameter5	Parameter that is to be read (	2776	4294967295	(not wired)
Parameter6	Parameter that is to be read (	2777	4294967295	(not wired)
Parameter7	Parameter that is to be read (	2778	4294967295	(not wired)
Parameter8	Parameter that is to be read (	2779	4294967295	(not wired)

Instrument.OEMOperList - 100 parameters

The example shows the first eight of 100 parameters of which the first has been selected as ‘Alarm 1 Threshold’. This parameter is to be read only when OEM Security is enabled and the controller is in Operator access level.

When the OEM Status is Locked, this list is not shown.

Effect of the ‘OEM ParamList’ Parameter

The table below shows the availability of the two ‘Alarm 1’ parameters set up in the previous pages when the ‘OEMParamList’ parameter is turned On or Off.

‘Alarm 2’ is used as an example of all parameters which have not been included in OEM Security.

‘OEMParamLists’	**Parameter**	**Controller in Configuration Access**		**Controller in Operator Access**	
		Alterable	Not alterable	Alterable	Not alterable
On	A1 Type	✓			✓
	A2 Type		✓		✓
	A1 Threshold		✓		✓
	A2 Threshold	✓		✓	
Off	A1 Type	✓			✓
	A2 Type	✓			✓
	A1 Threshold	✓		✓	
	A2 Threshold	✓		✓	

The iTools views shown in the next page show how this example is presented in the iTools browser:

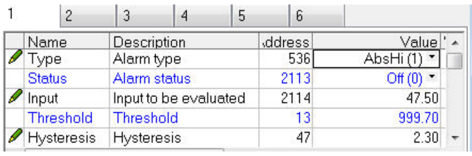
‘OEMParamLists’ On

The iTools views shown below show the alterability of the alarm parameters used in the previous examples. Alarm 1 has been set up in OEM Security. Alarm 2 is used as an example of parameters not set up in OEM Security.

Text in black shows parameters are alterable. Text in blue is not alterable.

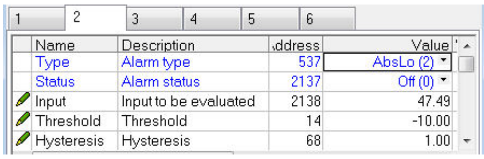
Controller in Configuration Mode

‘Alarm 1 Type’ is alterable
‘Alarm 1 Threshold’ is not alterable



	Name	Description	ddress	Value
✎	Type	Alarm type	536	AbsHi (1)
	Status	Alarm status	2113	Off (0)
✎	Input	Input to be evaluated	2114	47.50
	Threshold	Threshold	13	999.70
✎	Hysteresis	Hysteresis	47	2.30

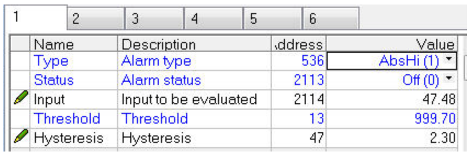
‘Alarm 2 Type’ is not alterable
‘Alarm 2 Threshold’ is alterable



	Name	Description	ddress	Value
	Type	Alarm type	537	AbsLo (2)
	Status	Alarm status	2137	Off (0)
✎	Input	Input to be evaluated	2138	47.49
✎	Threshold	Threshold	14	-10.00
✎	Hysteresis	Hysteresis	68	1.00

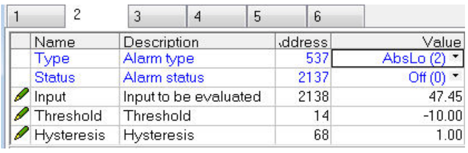
Controller in Operator Mode

‘Alarm 1Type’ is not alterable
‘Alarm 1 Threshold’ is not alterable



	Name	Description	ddress	Value
	Type	Alarm type	536	AbsHi (1)
	Status	Alarm status	2113	Off (0)
✎	Input	Input to be evaluated	2114	47.48
	Threshold	Threshold	13	999.70
✎	Hysteresis	Hysteresis	47	2.30

‘Alarm 2 Type’ is not alterable
‘Alarm 2 Threshold’ is alterable

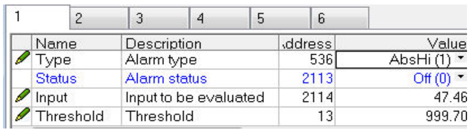


	Name	Description	ddress	Value
	Type	Alarm type	537	AbsLo (2)
	Status	Alarm status	2137	Off (0)
✎	Input	Input to be evaluated	2138	47.45
✎	Threshold	Threshold	14	-10.00
✎	Hysteresis	Hysteresis	68	1.00

‘OEMParaLists’ Off

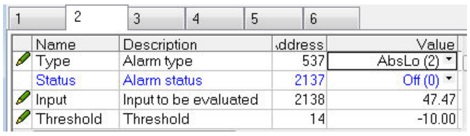
Controller in Configuration Mode

‘Alarm 1 Type’ is alterable
‘Alarm 1 Threshold’ is alterable



	Name	Description	ddress	Value
✎	Type	Alarm type	536	AbsHi (1)
	Status	Alarm status	2113	Off (0)
✎	Input	Input to be evaluated	2114	47.46
✎	Threshold	Threshold	13	999.70

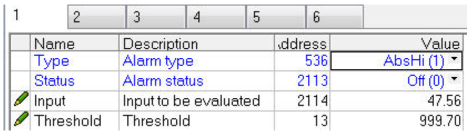
‘Alarm 2 Type’ is alterable
‘Alarm 2 Threshold’ is alterable



	Name	Description	ddress	Value
✎	Type	Alarm type	537	AbsLo (2)
	Status	Alarm status	2137	Off (0)
✎	Input	Input to be evaluated	2138	47.47
✎	Threshold	Threshold	14	-10.00

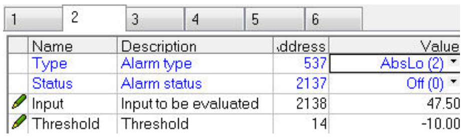
Controller in Operator Mode

‘Alarm 1 Type’ is not alterable
‘Alarm 1 Threshold’ is alterable



	Name	Description	ddress	Value
	Type	Alarm type	536	AbsHi (1)
	Status	Alarm status	2113	Off (0)
✎	Input	Input to be evaluated	2114	47.56
✎	Threshold	Threshold	13	999.70

‘Alarm 2 Type’ is not alterable
‘Alarm 2 Threshold’ is alterable



	Name	Description	ddress	Value
	Type	Alarm type	537	AbsLo (2)
	Status	Alarm status	2137	Off (0)
✎	Input	Input to be evaluated	2138	47.50
✎	Threshold	Threshold	14	-10.00

Note: Parameters are alterable within other set limits.

Firmware Upgrade

The firmware of the EPC2000 Programmable Controller can be upgraded over Ethernet via the PC application Eurotherm Firmware Management Tool. The upgrade utility automatically downloads the appropriate firmware image for the device connected.

To upgrade the firmware:

1. Confirm that the EPC2000 Programmable Controller is not connected to a live process.
2. Ensure iTools, (and any other connections for example Modbus or PROFINET) are disconnected from the EPC2000 Programmable Controller.
3. Launch the Eurotherm Firmware Management Tool either from the Windows Start menu, or through iTools Help menu > Launch Eurotherm Firmware Management Tool.

For detailed information on using the Eurotherm Firmware Management Tool, refer to the online help within the tool.

EPC2000 PROFINET

NOTICE

UNINTENDED EQUIPMENT OPERATION

At firmware upgrade the device may experience connection issues which could affect, or prevent the firmware upgrade procedure.

Observe the firmware upgrade messages and follow any instructions that are displayed during the update process.

Failure to follow these instructions can result in equipment damage.

The Eurotherm Firmware Management Tool initially identifies if a device has a temporary IP address, and if so cancels the upgrade process. The device must have a permanent (fixed) IP address.

If the EPC2000 Programmable Controller (PROFINET) does not connect and appear in the Eurotherm Firmware Management Tool after a minute, start iTools and check if iTools is able to connect to the device.

If iTools does not connect to the device see "EPC2000 PROFINET - Reconfiguring a fixed IP Address" on page 297.

If the EPC2000 PROFINET device has a temporary IP address, the Eurotherm Firmware Management Tool will notify and advise the device must be configured with a fixed IP address to continue.

To configure the EPC2000 PROFINET device with a permanent IP address, select the most relevant option below;

- Reconfigure the IP address using the Siemens TIA Portal application - this configures a permanent fixed IP address, see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266
- Confirm, if using a 3rd party PROFINET configuration tool that it applies a permanent fixed IP address, with the software manufacturer. (If you reconfigure the IP address, then power cycle the device and find it does not connect to iTools it is probably configuring a temporary IP address)
- Temporarily reconfigure a fixed IP address using iTools, see "EPC2000 PROFINET - Reconfiguring a fixed IP Address" on page 297

EPC2000 PROFINET - Reconfiguring a fixed IP Address

Connect the device to iTools and the 3rd party PROFINET Configuration tool, and power up. (iTools will not be able to connect because of the temporary PROFINET IP address being deleted (or defaulting to 0.0.0.0), as the device has been power cycled).

Using the 3rd party PROFINET Configuration tool, configure the same IP address the device had before commencing a firmware update - this will be temporary, but will allow iTools to find and connect to the device in the usual way.

Once connected do not use any other commands in 3rd party PROFINET Configuration tool. (Any other PROFINET comms sent will reset the IP Address to default (0.0.0.0) causing the device to drop off iTools).

From iTools enter configuration mode (enter the Configuration password when requested).

Manually re-enter, apply the same IP address within the following parameters;

- Set IP Mode to Static
- IPAddress1 to IPAddress4
- SubnetMask1 - SubnetMask4

Once updated, exit the configuration mode to commit the changes and complete setting a Fixed IP address for the device. (Power Cycle to test, but do not use any 3rd part PROFINET Configuration tools).

Now return to the Firmware upgrade process whilst the device has a fixed IP address configured.

See "Comms.Serial.Network and Comms.Ethernet.Network" on page 129 for further details regarding the IP Mode, IPAddress parameters.

Once the Firmware upgrade process is complete the EPC2000 PROFINET will need configuring using a suitable 3rd party PROFINET Configuration tool as before, see "Setting up the EPC2000 Programmable Controller for PROFINET" on page 266 for further details.

Note: The same requirements i.e. a temporary iTools fixed IP address, is necessary for Cloning and to clone a new Controller, see "Cloning" on page 88 for further details.

Technical Specification

General

Controller Function	Single loop DIN or Surface mount PID programmable controller with Autotune, ON/OFF, Valve Positioning (no slidewire required). Zirconia probe atmosphere control. Single loop profile/program up to 10 profiles of 24 segments, or 20 profiles of eight segments, (see Program/Profiler, below). 24V AC/DC options.
Measurement Inputs	Single input. +/− 0.1% accuracy
PID Control	Two PID sets are available (Separate proportional band for heat and cool). Enhanced Autotuning control with cutback to minimize overshoot and oscillation. Fast reacting precision control to setpoint changes or after process disturbances. Enhanced Valve Positioning (unbounded) algorithm. Gain scheduling allows PID selection for a wide range of operating situations, including deviation from setpoint, absolute temperature, output level and others. PV and SP feedforward functions.
Program/Profiler	Maximum 20 program sequences of eight segments. Options for 1x8, 1x24, 10x24, with textual program and segment names Holdback (“guaranteed soak”), event outputs, time to target, ramp rate, dwell, step and call segment types. Additional timer functions available.
User function block wiring	Optional Totalizer, Math, Logic and multiplexing, BCD conversion, Counter/Timer, Zirconia.
Additional Functions	Average, min, max, Zirconia. Six configurable alarms with manual, automatic, non-latching and event types plus alarm delay function and blocking. Alarms may be inhibited in standby. Five recipes with 40 selectable parameters switchable from digital input.
Backup and Configuration Tools	Free Eurotherm iTools software for backup and configuration. iTools connects also using Ethernet and serial Modbus RTU.
Ethernet	100BASE-T with integral switch. Certified to Achilles® Communications Robustness Testing Level 1[1].

[1] Achilles® Communications Robustness Testing Level 1 is not applicable to the following version; EPC2000 PROFINET.

Environmental specifications, standards, approvals and certifications

Operating Temperature	0℃ to 55℃ (32°F to 131°F)
Storage Temperature	−20℃ to 70℃ (-4°F to 158°F)
Operating/storage humidity	5% to 90%, non-condensing
Atmosphere	Non-corrosive, non-explosive
Altitude	< 2000 meters (6561.68ft)
Vibration / Shock	EN61131-2 (5 to 11.9Hz @ 7mm (0.275in) peak to peak displacement, 11.9-150Hz @ 2g, 1 octave/min.) EN60068-2-6 Test FC, Vibration. EN60068-2-27 Test Ea and guidance, Shock.
IP protection	EN60529 IP10 (IP20 with connectors in place)
Flammability of plastic materials	UL746C-V0
Electromagnetic compatibility (EMC) EMC Emissions EMC Immunity	 EN61326-1:2013 Class A EN61326-1:2013 Industrial locations
Electrical safety	EN 61010-1:2010/A1:2019, UL 61010-1: 2012. Pollution Degree 2 Insulation Category II
Approvals and Certification Europe & UK	 CE, UKCA, RoHS, REACH, WEEE
USA, Canada	UL, cUL (E57766)
China	RoHS, CCC: Exempt (product not listed in catalog of products subject to China Compulsory Certification)
Global	When subject to the necessary field calibration, EPC2000 Programmable Controllers manufactured by Eurotherm are suitable for use in Nadcap applications in all furnace classes, as defined in AMS2750E clause 3.3.1. Meets accuracy requirements of CQI-9 Achilles® Level 1 CRT Cyber Security Assessment[1]

[1] Achilles® Communications Robustness Testing Level 1 is not applicable to the following version; EPC2000 PROFINET.

EN ISO 13849 Assessment Declaration

The EPC2000 has been assessed against the following standards:

- EN ISO 13849-1:2015 – Safety of Machinery – Safety Related Parts of Control Systems
- EN ISO 13849-2:2012 – Safety of Machinery – Safety Related Parts of Control Systems – Part 2: Validation

The results are shown in the following table.

Key safety values	Value	Standard
Performance Level (PL)[1]	c	EN ISO 13849-1
Diagnostic Coverage$_{avg}$	None	
Mean Time to Dangerous Failure (MTTFd)	100 years[3]	
Category[2]	1	
Maximum service life	10 years	

1. The Performance Level is defined for the safety function of the EPC2000. The process is monitored using the PV input. In the event of an alarm condition, the OP3 change-over relay will be activated.
2. The EN ISO 13849-1 performance level (PL) and safety category (Cat) of the overall system depends on multiple factors, including the selected modules, wiring practices, the physical environment, and the application.
3. For the level of assessment, 100 years is the maximum acceptable MTTFd which is exceeded by all modular variants of the EPC2000.

Mechanical

Dimensions

Dimensions given as width × height.

EPC2000 Programmable Controller	Excluding terminals	32.6mm × 111mm 1.28 inch × 4.37 inch
	Including terminals (factory-fitted default terminals)	32.6mm × 131.2mm 1.28 inch × 5.17 inch
	Fixing centres (between mounting holes) Hole designed for M4 bolts	115mm (holes are 5.08mm offset - see install sheet) 4.53 inch (holes are 0.2 inch offset - see install sheet)
	Depth	107.3mm 4.22 inch

Weight

EPC2000 Programmable Controller	210 grams; 7.4oz

Input and Outputs

I/O and communication types

I/O and Comms	
Analog inputs	1 universal input 20Hz
Form A relay output	1
Form C relay output	1
Logic I/O or DC analog output	1
Contact closure logic input	2
Communications Ethernet	Dual Ethernet switch. Shielded grounded RJ45 connections supporting 10/100BASE-T auto sensing. Fixed IP address or DHCP.
Protocol Type (Requested Packet Interval (RPI) Range)	EtherNet/IP (50 - 3200 milliseconds) PROFINET (64 - 512 milliseconds) Modbus/TCP
Serial	EIA485 Half duplex Baud rates 9600, 19200 Modbus RTU 8 data bits, odd/even/no parity selectable

I/O specifications

Input types	Thermocouples, PT100 RTD, 4-20mA, 0-20mA, 10V, 80mV, 40mV, Zirconia (oxygen probe). Accuracy ±0.1% reading. When subject to the necessary field calibration, “Control, Monitoring and Recording Instruments” manufactured by Eurotherm are suitable for use in Nadcap applications in all furnace classes as defined in AMS2750E clause 3.3.1.
Sample time Process Inputs: Thermocouple: RTD:	50ms (20Hz) 62.5ms (16Hz) 100ms (10Hz)
Mains rejection Series mode rejection: Common mode rejection:	48-62Hz >80dB >150dB
Sensor break	AC sensor break, detected within three seconds worst case
Input filtering	Filter time constant = OFF to 60 seconds
User calibration	User 2-point input adjust (offset/gradient), transducer scaling
Thermocouple	B, J, K, L, N, R, S, T as standard plus 2 downloadable custom curves Linearization accuracy: CJ Calibration: <±1.0C at 25°C (77°F) ambient. CJ Ambient rejection ratio: better than 40:1 from 25°C (77°F) ambient CJ automatic (internal), variable (external fixed 0, 45, 50ºC) (32, 113, 122°F)

Inputs and outputs

Input ranges		**40mV**	**80mV**	**mA**	**10V**	**RTD (PT100)**
Range	Min	-40mV	-80mV	-32mA	-10V	0Ω (-200ºC)
	Max	+40mV	+80mV	+32mA	+10V	400Ω (850ºC)
Thermal stability from 25ºC ambient		±0.4µV/ºC ±13ppm/ºC	±0.4µV/ºC ±13ppm/ºC	±0.16µA/ºC ±113ppm/ºC	±8µV/ºC ±70ppm/ºC	±0.01ºC/ºC ±25ppm/ºC
Resolution		1.0µV unfiltered	1.6µV	0.6µA	250µV	0.05ºC
Measurement noise (pk to pk with 1.6s input filter)		0.8µV	3.2µV	1.3µA	500µV	0.05ºC
Linearity accuracy (best fit straight line)		0.003%	0.003%	0.003%	0.007%	0.0033%
Calibration accuracy at 25ºC ambient		±4.6µV ±0.053%	±7.5µV ±0.052%	±3µA ±1.052%	±1.5mV ±0.063%	±0.31ºC ±0.023%
Input resistance		100MΩ	100MΩ	2.49Ω (1% shunt)	57kΩ	
Bulb current						190µA

Contact closure inputs

Thresholds	Open > 400Ω, Closed < 100Ω
Input functions	Auto/Manual select, SP2 select, Integral Hold/Control inhibit/Program run functions/Recipe, Select/PID, select/BCD Bit/Autotune enable/Standby/PV Select plus other functions using soft wiring

Logic I/O modules

Output Rating	ON 12V DC 44mA max. Minimum control cycle time 50ms (auto)
Output Functions	Time proportioned heat, Time proportioned cool. SSR Drive Alarm and event outputs, interlock outputs, other functions using soft wiring.
Contact Closure (input)	Open > 400Ω, Closed < 100Ω
Input Functions	Auto/Manual select, Integral Hold, Control inhibit, Program run functions, Recipe select, PID select, BCD Bit, Autotune enable, Standby, PV Select plus other functions using soft wiring.

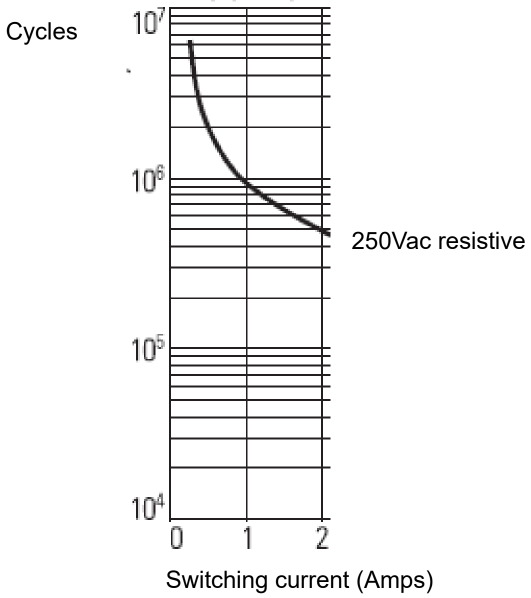
Relays

Types	Form A (normally open) Form C (changeover) Integrated snubbers (MOV type)
Output Functions	Time proportioned heat, time proportioned cool. SSR Drive. Direct Valve raise/lower. Alarm and event outputs, interlock outputs, other functions using soft wiring.
Rating	Min 100mA @ 12V, Max 2A @ 264V AC resistive. 0.5A: @ 264V AC inductive. Internal varistors used to help protect the relay's output contacts.

Relay Electrical Endurance

The number of operations which the relays are expected to endure is limited in accordance with the graph shown below for a resistive load. Typically at a load of 2A, 250Vac resistive at 23°C this is 500,000 operations - see below. Differences in load current, ambient temperature, load type and switching frequency will impact the number of operations.

Electrical Endurance



Isolated DC output module

	Current Output	**Voltage output**
Range	0-20mA	0-10V
Load resistance	<550Ω	>450Ω
Calibration accuracy	< ±(0.5% of reading + 100µA offset)	Calibration accuracy: < ±(0.5% of reading + 50mV offset)
Resolution	13.5 Bits resolution	13.5 Bits resolution
Output Functions	SCR/Power control drive. Proportional valve. Retransmission to chart recorder or other instrumentation. Other functions using soft wiring.	

Power supply

Controller supply voltage	24Vac +10/-15%, 48 to 62Hz 24Vdc +20/-15%, max 5% ripple voltage.
PSU Rating	6W

Eurotherm Ltd
Faraday Close, Worthing
West Sussex, BN13 3PL
Phone: +44 (0) 1903 268500

www.eurotherm.com

Contact your local sales representative

Published April 2023

HA033210ENG Issue 5

Powered by Possibility