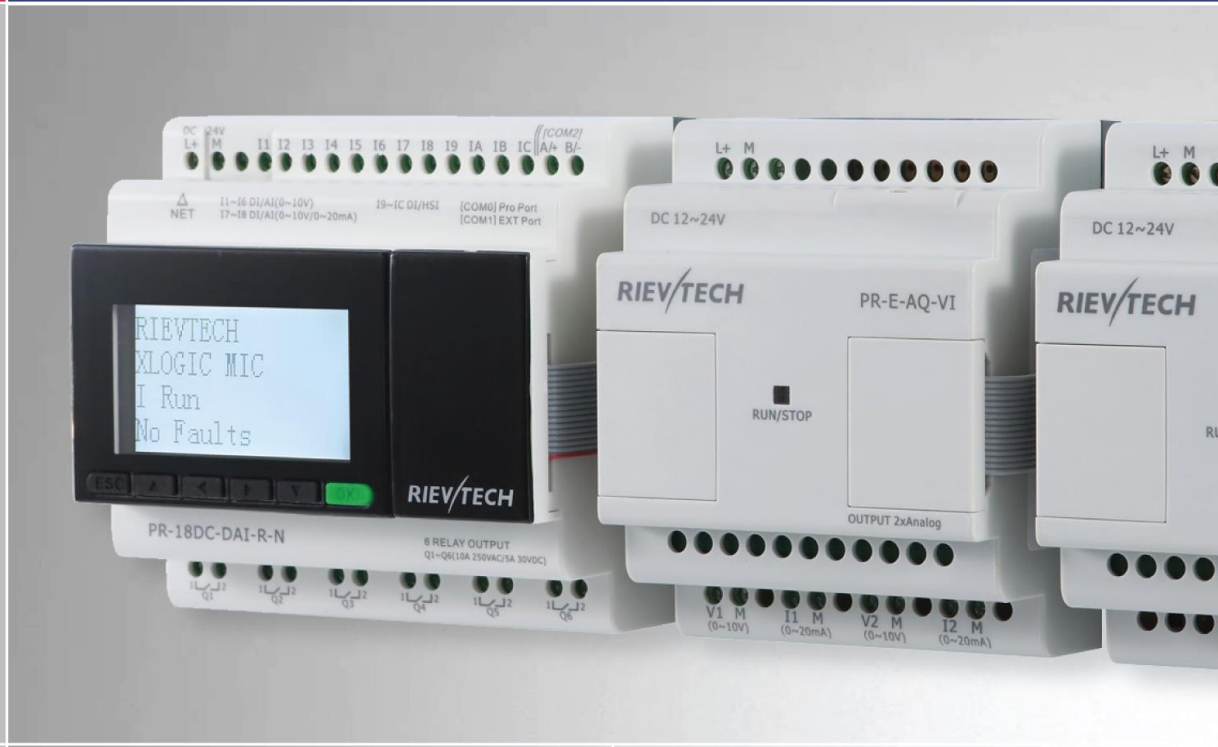
RIEV/TECH
EXPERT / FOR MICRO-AUTOMATION SOLUTIONS

MICRO PLC

User Manual

Applied to ELC, EXM & PR series

INTRODUCTION
GETTING STARTED
INSTALLATION AND WIRING
PROGRAMMING PLC
CONFIGURING & SOFTWARE
APPLICATIONS
TECHNICAL DATA



2018

EN

RIEV/TECH

Programmable Relay User's Manual

November 2019

Version 5.4

Rievtech Electronic Co. Ltd,

Unit 506, Zone A3, No.99 Gao Hu Road, Jiang Ning District., Nanjing, Jiangsu, China (Mainland)

Tel: +86-025-52713690

Document Revisions

Date	Version Number	Document Changes
28/06/2017	4.0	Initial Draft
20/05/2018	4.1	First Revision
02/07/2018	4.2	Addition of MQTT Functions
28/07/2018	4.3	Addition of Library Function
30/09/2018	4.4	Addition of PR Ethernet Series
15/10/2018	4.5	Addition of Webserver
08/12/2018	4.6	Updated LID Save instructions p.126
01/04/2019	4.7	Corrected Modbus Addresses for Middle Coils (M) p.193
27/05/2019	5.0	Addition of User Web Pages – Section 8.2

30/06/2019	5.1	x-Messenger Block Addition
19/08/2019	5.2	Addition of Analogue Table, Free Protocol, General Data Latching Relay and General Arithmetic blocks
07/09/2019	5.3	Added Modbus Read Write block
08/11//2019	5.4	Updated new MQTT Methods

Table of Contents

1. Introduction 14
1. Products Covered 15
1.1. Product Safety 15
1.2. Qualified Personnel 15
1.3. Trademarks 15
1.4. Copyright Rievtech 2015 all rights reserved 15
1.5. Disclaim of Liability 15
1.6. Additional support 16
2. General Introduction to xLogic 17
2.1. Overview 17
2.2. Feature Highlights 17
2.3. Some of the things xLogic can do for you 17
2.4. xLogic Devices 18
2.5. Communication / Network 18
3. Hardware Models and Resources 21
3.1. ELC and PR-Series CPU Naming Convention 21
3.2. ELC, EXM and PR-Series Expansion Unit Naming Convention 23
3.3. Range of Hardware 24
3.3.1. Non-Expandable CPU Range 24
3.3.2. Expandable CPU Range 24
3.4. Expansion Modules 27
3.5. Accessories 29
3.6. Structure & Dimensions 30
3.6.1. PR-6 Series CPU and EXM-E Expansion Modules 30
3.6.2. PR-12 Series with LCD 30
3.6.3. PR-12 Series without LCD 30

3.6.4. PR-14 and PR-18 Series 30

3.6.5. PR-24 Series 30

3.6.6. PR-E Expansion Modules 31

3.7. EXM GSM/GPRS Built-In Module 32

3.7.1. How to Establish Connection between CPU and PC Using GPRS 32

3.7.2. Configuration of E-mail Settings 40

3.7.3. Changing CPU Information via SMS 41

4. Installing and Removing Rievtech Hardware 43

4.1. Dimensions 43

4.2. DIN Rail Mounting 43

4.2.1. Mounting Your Hardware 43

4.2.2. Removing Your Hardware 43

4.3. Wall Mounting 44

4.3.1. Drilling Templates for Wall Mounting 44

4.4. Wiring 45

4.4.1. Connecting Power 45

4.4.2. Connecting Input Signals 45

4.4.3. DC Type Digital Input Connections 47

4.4.4. AC Type Digital Input Connections 47

4.4.5. Analogue Input Connections (0 to 10 VDC) 47

4.4.6. Analogue Input Connections (0 to 20 mA) 47

4.4.7. Analogue Input Connections (PT100) 48

4.4.8. Connecting Output Signals 48

4.5. Communication Port Instructions 49

Expansion port/RS485 (pin definition) 50

5. Parameter Modification using CPU HMI 52

5.1. Overview of CPU Menu Structure 52

5.2. Parameter Assignment Mode 52

5.3. LCD Panel Instructions 52

LCD Menu Structure 53

5.4. Selecting a Function Page **55**

5.4.1. Function Page Options **55**

Run/Stop 55

Set Param 55

5.4.2. How to Switch Between RUN and STOP **55**

5.4.3. Set Parameter **56**

How to modify parameter? 57

5.4.4. Set Password **59**

5.4.5. How to Set the Address of a CPU and Expansion Modules **61**

5.4.6. Setting LCD Backlight and Contrast **63**

5.4.7. Set Communication Parameters **64**

5.4.8. Setting System Date and Time **65**

6. Configuring and Programming Rievtech Hardware **66**

6.1. xLogicsoft Functions **67**

6.1.1. General Input and Output Functions **68**

Digital Input 68

Cursor keys 68

Digital Output 68

Permanent Logical Levels HI and LO 70

Panel Key 70

Shift Register Bits 70

Analogue Inputs 70

F (Digital Flag) 71

AF (Analogue Flag) 72

SMS Message Input 73

SMS Message Output 75

SMS Message Input Output 78

GPRS Data Upload 79

GPRS Connect 82

GPRS Disconnect 83

6.1.2. Basic Functions List – GF 84

AND 84

AND with Edge Evaluation 84

NAND 85

NAND with Edge Evaluation 85

OR 86

NOR 86

XOR 86

NOT 86

Boolean Function 87

6.1.3. Basics of Special Functions - SF 87

Designation of the Inputs to Blocks 88

Time Response 88

Backup of the Real-time Clock 89

Retentivity 89

Parameter Protection 89

Calculating the Gain and Offset of Analogue Values 89

6.1.4. Special functions list – SF 90

On-Delay 93

Off-Delay 95

On/Off-Delay 96

Retentive On-Delay 97

Wiping Relay (pulse output) 98

Edge Triggered Wiping Relay 100

Asynchronous Pulse Generator 101

Random Generator 102

Stairway Lighting Switch 104

Multiple Function Switch 105

Weekly Timer 107

Yearly Timer 110

Astronomical Clock 113

Stopwatch 114

Up/Down Counter 115

Hours Counter 117

Threshold Trigger 119

Analogue Comparator 121

Analogue Threshold Trigger 124

Analogue Amplifier 125
Analogue Watchdog 126
Analogue Differential Trigger 128
Analogue Multiplexer 129
PI Controller 131
Analogue Ramp 138
Analogue Math 140
Long Data Math!! 142
Analogue Math Error Detection 142
Analogue Filter 144
Max/Min 145
Average Value 148
Latching Relay 149
Pulse Relay 150
Message Text 152
CPU Cover 158
Softkey Block 159
Shift Register 160
Pulse Width Modulator (PWM) 162
Modbus Read 164
Modbus Write 168
Modbus Read Write 172
Data Latching Relay 176
Long Data Latching Relay 177
Memory Write 180
Memory Read 182
Word to Bit 183
Bit to Word 185
Device Reset 186
Comport Status 188
RH Math 190

6.1.5. App Functions List – AP 191

Cam Control 191
Angular Cam Timer 192
Pump Management 193

Defrost 195

Comparison of Two Values ... 196

Multi-Compare ... 197

Compare in Zone ... 198

Conversion Word to Bits ... 200

Conversion Bits to Word ... 200

De-Multiplexer ... 200

Multiplexing ... 202

Multiplexer ... 202

Square Root ... 203

Sin & Cos ... 203

Absolute Humidity ... 205

Analogue Table ... 205

General Data Latching Relay ... 212

Free Protocol Communication ... 213

General Arithmetic ... 215

6.1.6. Library Function (LIB) ... 218

Library Function Options ... 218

LID Customisation ... 220

Saving a Library Function ... 221

Updating the Library Function Block ... 222

6.1.7. MQTT Functions ... 223

Configure xLogic for MQTT Server ... 223

Connect to MQTT Server ... 225

Setting Up MQTT Server Example ... 226

MQTT Data Format in xLogicsoft ... 228

VB Block ... 228

VW Block ... 228

VD Block ... 230

String to Integer ... 231

String Comparator ... 232

Publish with MQTT ... 233

MQTT Publish Status Codes ... 238

Subscribe with MQTT ... 239

7. xLogicsoft ... 245

7.1. Installing xLogicsoft ... 245

7.2. Main Functions 248
Editing Function 248
Simulation Function 248
Real-time Monitoring 248
Documentation 248

7.3. Operation Instructions 248
Menu Bar 248
Standard Toolbar 250
Programming Toolbar 251
Selection Tool 251
Text Tool 251
Cut/Join 251
Simulation Tool and Status Window 251
Basic Operation 253
Edit a Function Diagram Program 254
Program Simulation 256
How to Print Your Program File 258
Modifying Password and Transferring a Program 258
Setting Time and Date 259
On-line Monitoring and Program Checking 259

8. How to Configure an Ethernet CPU? 262
Ethernet network 262
Ethernet CPUs with Built-in Webserver 263
How to Distinguish Between New and Old CPU Versions 263
Differences Between Old and New Versions of CPU 263
How to Configure the Network Parameters via Software 264
How to View and Configure the Ethernet parameters via the LCD Panel 266
LCD Menu Structure 268
How to Create Communication between the CPU and a PC via Ethernet? 268
How to Logon to the built-in Web Server 271
How to Establish the Communication between new Ethernet CPUs 277

8.1. Modification of User Web Pages 293
Editing Web Pages 293
Page 1 – login.html 293
Page 2 – cpuinfo.html 295

Page 3 – varscfg.html 295
Page 4 – monitor.html 295
Page 5 – usermanager.html 296
Page 6 – Rievtech.html 297

8.2. Downloading Web Pages to CPU 297

8.3. Configuration with Ethernet Device Manager 299

Basic Settings 300
Network Settings 301
Server Settings 301
Channels 301

8.4. Establish communication between CPU and xLogicsoft/SCADA via Ethernet. 303

Ethernet CPU as Client and xLogicsoft as Server 303
Ethernet CPU as Server and xLogicsoft as Client 304

8.5. How to Establish Communication Between CPUs via Ethernet? 304

Hardware Connection 304

8.6. Resetting to Factory Defaults 309

Resetting Ethernet CPU to Defaults 309
Resetting WIFI CPU to Defaults 310

9. Applications 312

9.1. Dual-function switch 312

Standard solution 312
xLogicsoft solution 312
The scheme of xLogic 313

9.2. Automatic Gate 313

Standard solution 314

9.3. Ventilation System 315

Standard Solution 315
The Scheme of xLogic 316
xLogic Solution 316
The circuit in xLogicsoft 316

9.4. Factory Gate 317

Standard Solution 317
The Scheme of xLogic 317
xLogicsoft Solution 318

9.5. Daylight Lamp System 318
Requirements for Lighting System 318
Standard Solution 318
The Scheme of xLogic 319
Circuit Diagram by xLogicsoft 319
9.6. Rainwater Pump 320
Standard Solution 320
The Scheme of xLogic 320
10. Modbus Function Codes and Register Addresses 322
10.1. xLogic Supported Modbus Function Codes 322
10.2. Register Addresses of xLogic 322

1. Introduction

Congratulations on the purchase of your xLogic Micro PLC provided by Rievtech Electronic Co., Ltd.

The xLogic Micro PLC is a compact and expandable CPU replacing mini PLCs, multiple timers, relays and counters.

The xLogic Micro PLC perfectly fits in the space between timer relays and low-end PLCs. Each CPU incorporates not only a real-time clock and calendar but also provides support for optional expansion I/O modules to enhance control and monitoring applications. Adjustments to programs can easily be performed via the Rievtech easy to use xLogicsoft programming software. The hardware can be either DIN-rail and panel-mounted, offering full flexibility for the various installation needs of your application.

The xLogic Micro PLC is available in 120V/240V AC or 12V/24V DC versions, making it the ideal solution for relay replacement, or simple control applications such as building and parking lot lighting, managing automatic lighting, access control, watering systems, pump control, ventilation systems, home automation and a wide array of other applications which require a low cost as a primary design issue.

We strongly recommended taking the time to read this manual, before putting the xLogic Micro PLC to work. Installation, programming and use of the unit are detailed in this manual. The feature-rich xLogic Micro PLC provides an off-line operation mode, allowing full configuration and testing prior to in-field service commissioning. In reviewing this manual, you will discover many additional advantageous product properties, which will greatly simplify and optimize the use of your xLogic Micro PLC.

1. Products Covered

The manual applies to devices of ELC and EXM series and PR series. Our full range of xLogic Micro PLC's.

1.1. Product Safety

This manual contains notices you have to observe to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol; notices referring to property damage only, have no safety alert symbol. The notices shown below are graded according to the degree of danger.

Table 1 - Safety Warnings

Symbol	Hazard Description
	Indicates that death, severe personal injury or damage to assets **will** result if proper precautions are not taken.
	Indicates that death, severe personal injury or damage to assets **may** result if proper precautions are not taken.
	Indicates personal injury or damage to assets can result if proper precautions are not taken.
	Indicates that an unintended result or situation may occur if the corresponding notice is not taken.

NOTICE:

If more than one warning is present, then the warning with the highest degree of danger shall be used.

1.2. Qualified Personnel

Only persons qualified to correct electrical and safety standards and regulations at the time of use should be allowed to install, commission and operate this equipment.

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by qualified personnel. Within the context of the safety notices in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards. Please read the complete operating instructions before installation and commissioning. Rievtech or its distributors does not accept any liability for possible damage to persons, buildings or machines, which occur due to incorrect use or from not following this manual.

1.3. Trademarks

All names identified by xLogic are registered trademarks of the Rievtech. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

1.4. Copyright Rievtech 2015 all rights reserved

The distribution and duplication of this document or the utilization and transmission of its contents are not permitted without express written permission. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

1.5. Disclaim of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we

cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

1.6. Additional support

We take pride in answering your question as soon as we can. Please consult our website at www.rievtech.com for your closest point of contact or email us at sales@rievtech.com.

2. General Introduction to xLogic

2.1. Overview

xLogic is a universal logic module made by Rievtech.

xLogic is a compact, expandable CPU that can replace mini PLC, multiple timers, relays and counters. Bridging the gap between a timing relay and a low-end PLC. Each CPU has a real-time clock and calendar and supports optional expansion I/O modules to meet your control and monitoring applications needs. Data adjustments can be done via the onboard keypad and LCD display, or with xLogicsoft. It can be either DIN-rail or panel mounted, depending upon the needs of your application, and it is available in 120/240 VAC as well as 12/24 VDC versions, and it is an ideal solution for relay replacement applications, simple control applications such as building and parking lot lighting, managing automatic lighting, access control, watering systems, pump control, or ventilation systems in factory, and home automation and applications in which cost is a primary design issue.

2.2. Feature Highlights

- 4-lines, 16-characters per line, backlight display.
- Multiple value display and input via keypad and LCD display.
- 70 kinds of function Blocks can be used in a circuit program in maximum
- Standard Modbus RTU/ASCII/TCP communication protocol supported.
- It's optional for xLogic to act as a slave or master in a Modbus communication network.
- Easily connect to other factory devices such as touch screens by RS232 cable and/or RS485 module.
- CAN BUS protocol-based expansion modules (PR-18/PR-24 series CPU)
- Expandable up to 16 linked IO expansion modules reaching 282 I/O points in maximum
- Optional RS232, RS485 and Ethernet connectivity
- Multiple channels analogue inputs available with 0-10 VDC, PT100 & 0/4….20mA.
- Default Real Time Clock (RTC) and Summer/Winter timer available
- Real Time Clock (RTC) at 25 °C kept for 20 days
- 4 channels high-speed counting
- Pre-configured standard functions, e.g. on/ off-delays, pulse relay and softkey
- 2 PWM channels (10kHz maximum)
- Retentive memory capability (Not applicable to PR-6 & PR-12-E series CPU)
- Isolated RS232 and USB programming cables
- Program capability up to
  - 1024 function blocks for PR-18, PR-24, PR-12-N, PR-18-N & PR-26-N
  - 512 function blocks for PR-12 & EXM
  - 64 function blocks for PR-6 & PR-12-E
- Mounting via modular 35mm DIN rail or screw fixed mounting plate
- On-line monitor capability (Free SCADA software for all series xLogic)
- Datalogging
- A range of analogue signals available (0 to 10 VDC, 0/4 to 20 mA and PT100 probe inputs and 0 to10VDC and 0/4 to 20mA outputs)
- Low cost

2.3. Some of the things xLogic can do for you

The xLogic Micro PLC provides solutions for commercial, industrial, building and domestic applications such as

lighting, pumping, ventilation, shutter operations or in switching cabinets. The possible uses are diverse, and these are just a few possibilities.

Using the RS485 bus and Ethernet connectivity allows the user to realize various extensive (real-time) monitoring and control applications.

Special versions, without operator LCD and display unit, are available for series production applications in a small machine, installation and cabinet building environments to further reduce cost.

2.4. xLogic Devices

xLogic Basic is available in two voltage ranges.

*DC Range: 12-24 VDC i.e.: PR-6DC, PR-12DC, PR-14DC, PR-18, PR-24DC, PR-26DC and EXM-12DC series.

*AC Range: 110-240 VAC i.e.: PR-6AC, PR-12AC, PR-14AC, PR-18AC, PR-24AC, PR-26AC and EXM-8AC series.

*Without LCD Display: PR-6 series and with "-E" model, such as PR-12DC-DA-R-E.

PR-18, PR-24, PR26 and EXM models all have an LCD display.

Expansion modules:

PR-E (for use with PR-14, ELC-22-N, PR-18, PR-24, PR-26 and all PR-Ethernet CPUs).

EXM-E (for use with all EXM CPUs.

* xLogic digital expansion modules are available in 12/24 VDC and 110/240 VAC versions and have eight inputs and eight outputs.

* xLogic analogue expansion modules are available in 12/24 VDC only and have up to six digital and 4 analogue inputs.

Communication Cable and Expansion Module:

xLogic: RS232 communication cable (Model: ELC-RS232)

It is a serial cable with photoelectric isolation which can be directly connected to a standard 9-pin port of PC. This enables the user's program to be downloaded into xLogic CPU through xLogicsoft.

It can also serve as a connection cable between CPU and third-party devices with an RS232, such as HMIs, PLCs, etc. in a Modbus communication system.

xLogic: USB communication cable (Model: ELC-USB).

It is a communication cable with photoelectric isolation through which PC with USB port can be connected to xLogic CPU module, moreover, it has same features as the ELC-RS232 module, so it is quite convenient for users whose computer has no standard serial port.

xLogic: PRO-RS485 cable (Model: PRO-RS485).

It is a serial converter cable with photoelectric isolation, which converts the CPU RS232 programming port to an isolated RS485 port.

xLogic: RS485 module（Model：PR-RS485）

An isolated RS485 converter providing terminals of RS485 port for PR-14, PR-18, ELC-22-N and PR-24 series CPUs for connection with third-party devices.

2.5. Communication / Network

xLogic offers different ways to communicate within a system.

RS485 port

The RS485 port is used for communication between the CPU and various devices or equipment which have a standard RS485 port. Communication is via the Modbus RTU/ASCII protocol.

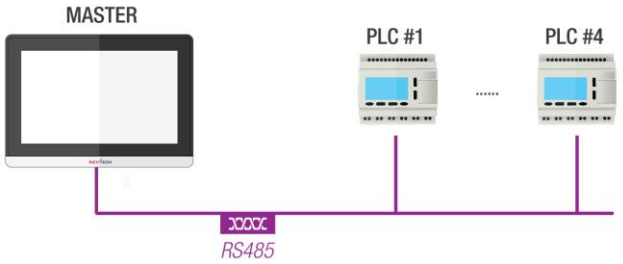


Figure 1 xLogic as RS485 Slave

NOTICE:

PR-E-RS485 module (or PRO-RS485 Cable) is required to connect the CPU to the RS485 bus except for models with built-in RS485 port.

RS232 or USB port (ELC-RS232/ ELC-USB needed)

If the application does not require an RS485 network and only one main module with some expansion modules is needed for the application, then download/upload of the CPU program happens over the standard RS232 or USB port. It allows system maintenance like monitoring too.

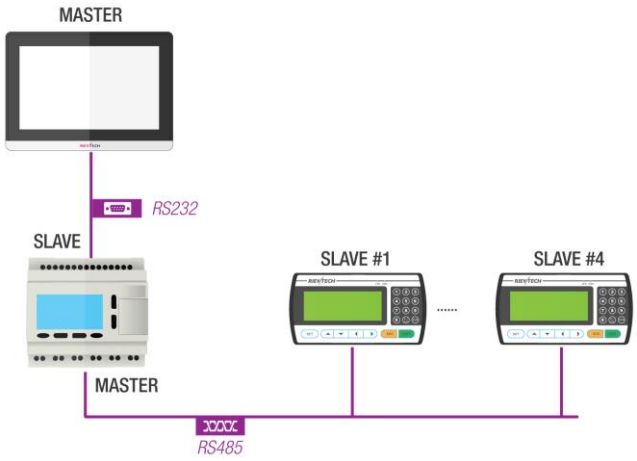


Figure 2 xLogic as RS485 Master

Ethernet network

If the application requires a system where more than one main module is needed and these main modules must communicate with each other, then an Ethernet CPU can be connected directly to an Ethernet network using the built-in LAN port. The CPU programs can also be uploaded/downloaded via Ethernet too. Furthermore, our free of charge SCADA software can be used with our Ethernet CPUs.

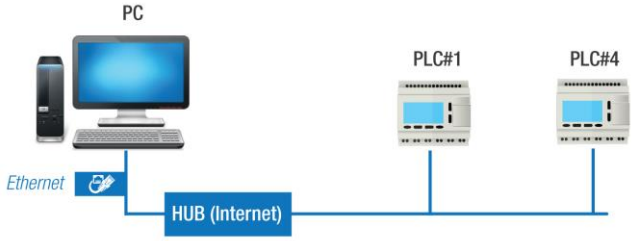


Figure 3 xLogic Ethernet

xLogic CPU can be connected with expansion modules of different supply voltages, but the expansion module must be supplied with the correct voltage corresponding to its type.

Each xLogic CPU provides the following program capabilities regardless of the number of connected expansion units:

- Digital Inputs
  - I1 to I4 (PR-6)
  - I1 to I6 (EXM-8AC)
  - I1 to I8 (PR-12, EXM-12)
  - I1 to IC (PR-18)
  - I1 to IE (PR-24)
  - I1 to IG (PR-26)
  - I11 to I18 (Expansion with address 1)
  - I161 to I168 (Expansion with address 16)
- Analogue Inputs
  - I1 to I4 (PR-6, PR-12, EXM-12)
  - I1 to I6 (PR-14, PR-18, PR-24)

- I1 to I8 and ID to IG (PR-26)
- I11 to I14 (PR-E expansion with address 1)
- AI161 to AI164 (PR-E expansion with address 16)

- Digital Outputs
  - Q1 to Q2 (PR-6)
  - Q1 to Q4 (PR-12, PR-14, EXM-12)
  - Q1 to Q6 (PR-18)
  - Q1 to QA (PR-24, PR-26)
  - Q11 to Q18 (PR-E expansion with address 1)
  - Q161 to Q168 (PR-E expansion with address 16)
- Digital Flag blocks
  - F1-F32 (PR-6 and PR-12-E)
  - F1-F256 (PR-12, PR-18, PR-24, PR-26 and EXM)
  - -F8: Start-up flag
  - -F63: Buzzer of ELC-43TS control bit
  - -F64: Backlight control bit (to control the backlight of LCD and backlight of ELC-43TS)
- Analogue Flag blocks
  - AF1-AF32 (PR-6 and PR-12-E)
  - AF1 to AF256 (PR-12, PR-18, PR-24, PR-26 and EXM)
- Shift register bits S1 to S8

3. Hardware Models and Resources

3.1. ELC and PR-Series CPU Naming Convention

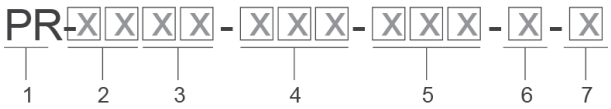


Figure 4 - CPU Naming Convention

1. Series name – ELC-Series or PR-Series.
2. Total number of inputs and outputs for the device.
3. Voltage Supply
    a. AC
    b. DC
4. Types of Inputs:
    a. D – Digital only
    b. DA – Digital and Analogue Voltage
    c. DAI – Digital and Analogue Voltage and Current
5. Type of Outputs:
    a. R – Relay
    b. TN – Transistor PNP
    c. RT – Relay and Transistor PNP
    d. RTA – Relay, Transistor PNP and Analogue Voltage and Current
6. E = Economic model –Non-expandable.
7. N = Ethernet built-in.

NOTICE:

Sometimes additional letters will be used after the product model as follows:

- CAP = CPU has no built-in LCD.
- HMI = CPU has built-in LCD and Panel keys

EXM CPU Naming Convention

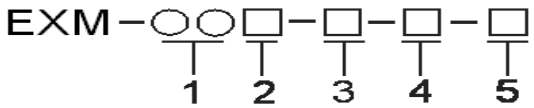


Figure 5 - CPU Naming Convention

1. Total number of inputs and outputs for the device.
2. Voltage Supply
    a. AC
    b. DC
3. Types of Inputs:
    a. D – Digital only
    b. DA – Digital and Analogue Voltage
    c. DAI – Digital and Analogue Voltage and Current
4. Type of Outputs:
    a. R – Relay
    b. TN – Transistor PNP
    c. RT – Relay and Transistor PNP
5. Special Function
    a. N = Ethernet built-in
    b. WIFI = WiFi onnectivity but no GSM/GRS
    c. GWIFI = WiFi connectivity and GSM/GPRS

3.2. ELC, EXM and PR-Series Expansion Unit Naming Convention

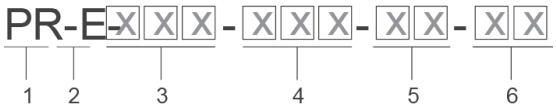


Figure 6 – PR-E or EXM-E Expansion Unit Naming Convention

1. Series name – PR or EXM.
2. Indicates Expansion module.
3. Type of input and output points
4. A number – Total number of inputs and outputs
    a. AI – Analogue input only
    b. AQ – Analogue output only
    c. PT100 – Temperature input only PT100
    d. RS485 – Isolated RS485 communication
5. Voltage Supply:
    a. AC
    b. DC
6. Types of Inputs:
    a. D – Digital only
    b. DA – Digital and Analogue
7. Type of Outputs:
    a. R – Relay
    b. TN – Transistor PNP

3.3. Range of Hardware

The Rievtech range of micro programmable logic controllers provides a mix of expandable and non-expandable CPUs. Packed with powerful features, each CPU in the range can provide a good mix of analogue/digital signals for your control projects.

NOTICE:

Key for understanding lettering used in the following tables:

N = No; Y = Yes;

D = Digital only; DA = Digital or Analogue;

R = Relay 10A max. rating (2A inductive);

T = Transistor PNP 0.3A max. rating.

3.3.1. Non-Expandable CPU Range

Table 1 - PR-6 Series

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	PWM	HMI	RTC
PR-6AC-R	No	110-240 VAC/DC	4 Digital	2 Relays (10A)	No	No	No	Yes
PR-6DC-DA-R	No	12-24 VDC	4DA 0-10V	2 Relays (10A)	No	No	No	Yes

Table 2 - PR-12 Series

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	PWM	HMI	RTC
PR-12AC-R-E	No	110-240 VAC/DC	8 Digital	4 Relays (10A)	No	No	No	Yes
PR-12DC-DA-R-E	No	12-24 VDC	8D (4DA 0-10V)	4 Relays (10A)	No	No	No	Yes
PR-12AC-R	No	110-240 VAC/DC	8 Digital	4 Relays (10A)	No	No	Yes	Yes
PR-12DC-DA-R	No	12-24 VDC	8D (4DA 0-10V)	4 Relays (10A)	4x60kHz	No	Yes	Yes
PR-12DC-DA-TN	No	12-24 VDC	8D (4DA 0-10V)	4 T (PNP)	4x60kHz	4x10kHz	Yes	Yes

3.3.2. Expandable CPU Range

Table 3 - PR-14 Series

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	PWM	HMI	RTC
PR-14AC-R	Yes	110-240 VAC/DC	10 Digital	4 Relays (10A)	No	No	Yes	Yes

PR-14DC-DA-R	Yes	12-24 VDC	10D (6DA 0-10V)	4 Relays (10A)	4x60kHz	No	Yes	Yes

Table 4 - PR-18 Series

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	PWM	HMI	RTC
PR-18AC-R	Yes	110-240 VAC/DC	12 Digital	6 Relays (10A)	No	No	Yes	Yes
PR-18DC-DA-R	Yes	12-24 VDC	.2D (6DA 0-10V)	6 Relays (10A)	4x60kHz	No	Yes	Yes
PR-18DC-DA-RT	Yes	12-24 VDC	.2D (6DA 0-10V)	4R + 2T (PNP)	4x60kHz	2x10kHz	Yes	Yes

Table 5 - PR-24 Series CPU Range – Built-In RS485 Port

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	PWM	HMI	RTC
PR-24AC-R	Yes	110-240 VAC/DC	14 Digital	6 Relays (10A)	No	No	Yes	Yes
PR-24DC-DA-R	Yes	12-24 VDC	.4D (6DA 0-10V)	6 Relays (10A)	4x60kHz	No	Yes	Yes
PR-24DC-DAI-RTA	Yes	12-24 VDC	.2D (6DA 0-10V) + 2A (0-20mA)	6R + 2T (PNP) + 1A (0-10V/0-20mA)	4x60kHz	2x10kHz	Yes	Yes

Table 6 - EXM Series CPU Range

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	G=GSM E=Ethernet	HMI	RTC
EXM-8AC-R	Yes	110-240 VAC/DC	6 Digital	2 Relays (10A)	No	G	Yes	Yes
EXM-12DC-DA-R	Yes	12-24 VDC	8D (4DA 0-10V)	4 Relays (10A)	2x60kHz	G	Yes	Yes
EXM-12DC-DA-R-N	Yes	12-24 VDC	8D (4DA 0-10V)	4 Relays (10A)	2x60kHz	G/E	Yes	Yes
EXM-8DC-PT100-R	Yes	12-24 VDC	2D + 2 PT100	4 Relays (10A)	No	G	Yes	Yes
EXM-12DC-DAI-R	Yes	12-24 VDC	6D (2DA 0-10V) + 2A (0-20mA)	4 Relays (10A)	2x60kHz	G	Yes	Yes
EXM-12DC-DA-RT-WIFI	Yes	12-24 VDC	8D (4DA 0-10V)	2 Relays (10A) 2 Trans (PNP)	2x60kHz	WIFI	Yes	Yes
EXM-12DC-DA-RT-GWIFI	Yes	12-24 VDC	8D (4DA 0-10V)	2 Relays (10A) 2 Trans (PNP)	2x60kHz	G/WIFI	Yes	Yes

Table 7 - Ethernet Remote Series CPU Units – Built-In RS485 Port and Web Server

Model	Expansion	Supply	Inputs	Outputs	High Speed Count	PWM	HMI	RTC
PR-12AC-R-N	Yes	110-240 VAC/DC	8 Digital	4 Relays (10A)	No	No	Yes	Yes
PR-12DC-DA-R-N	Yes	24 VDC	8D (4DA 0-10V)	4 Relays (10A)	4x60kHz	No	Yes	Yes
PR-18AC-R-N (No Built-in RS485)	Yes	110-240 VAC/DC	12 Digital	6 Relays (10A)	No	No	Yes	Yes
PR-18DC-DAI-R-N	Yes	24 VDC	12D (6DA 0-10V + 2DA 0-20mA)	6 Relays (10A)	4x60kHz	No	Yes	Yes
PR-18DC-DAI-TN-N	Yes	12-24 VDC	12D (6DA 0-10V) + 2A (0-20mA)	6T (PNP)	4x60kHz	2x10kHz	Yes	Yes
PR-26AC-R-N	Yes	110-240 VAC/DC	16 Digital	10 Relays (10/5A)	No	No	Yes	Yes
PR-26DC-DAI-RA-N	Yes	24 VDC	16D (8DA 0-10V + 4DA 0-20mA)	8R + 2A (0-10V/0-20mA)	4x60kHz	No	Yes	Yes
PR-26DC-DAI-RT-N	Yes	24 VDC	16D (8DA 0-10V + 4DA 0-20mA)	8R + 2T (PNP)	4x60kHz	2x10kHz	Yes	Yes
PR-26DC-DAI-RT-2G	Yes	24 VDC	16D (8DA 0-10V + 4DA 0-20mA)	8R + 2T (PNP)	4x60kHz	2x10kHz	Yes	Yes
PR-26DC-DAI-RT-4G	Yes	24 VDC	16D (8DA 0-10V + 4DA 0-20mA)	8R + 2T (PNP)	4x60kHz	2x10kHz	Yes	Yes
PR-26DC-DAI-RT-WIFI	Yes	24 VDC	16D (8DA 0-10V + 4DA 0-20mA)	8R + 2T (PNP)	4x60kHz	2x10kHz	Yes	Yes
PR-26DC-DA-RT-4GWIFI	Yes	24 VDC	16D (8DA 0-10V + 4DA 0-20mA)	8R + 2T (PNP)	4x60kHz	2x10kHz	Yes	Yes

3.4. Expansion Modules

Table 8 - Expansion Modules for PR-14, PR-18, PR-24 and Remote PLC CPUs (Max. 16 Units)

Model	Supply	Inputs	Outputs
PR-E-16AC-R	110-240 VAC/DC	8 Digital	4 Relays (10A) + 4 Relays (3A)
PR-E-16DC-DA-R	12-24 VDC	8Digital (4DA 0-10V)	4 Relays (10A) + 4 Relays (3A)
PR-E-16DC-DA-TN	12-24 VDC	8Digital (4DA 0-10V)	8 Transistor (0.3A)
PR-E-AI-I	12-24 VDC	4x0/4-20mA	-
PR-E-PT100	12-24 VDC	3 x Thermistor PT100	-

PR-E-AQ-VI	12-24 VDC	-	2 x 0-10 V/0-20 mA
PR-RS485	12-24 VDC	Additional Isolated RS485 Port	

Table 9 - Expansion Modules for EXM PLC CPUs (Max. 8 Units)

Model	Supply	Inputs	Outputs
EXM-E-8AC-R	110-240 VAC/DC	4 Digital	2 Relays (10A) + 2 Relays (3A)
EXM-E-8DC-DA-R	12-24 VDC	4Digital (4DA 0-10V)	2 Relays (10A) + 2 Relays (3A)
EXM-E-AI-I	12-24 VDC	4 x 0/4-20mA	-
EXM-E-PT100	12-24 VDC	2 x Thermistor PT100	-
EXM-E-AQ-V	12-24 VDC	-	2 x 0-10 V
EXM-E-AQ-I	12-24 VDC	-	2 x 0-20 mA
EXM-E-RS485	12-24 VDC	Additional Isolated RS485 Port	

3.5. Accessories

Table 10 - xLogic Accessories

Model	Description
ELC-RS232	RS232 Programming and communication lead between CPU and PC
ELC-USB	USB Programming lead between CPU and PC. Needed for firmware updates.
ELC-COPIER	A device used to copy a program from one CPU to another same model CPU
ELC-MEMORY	Real-time data logging module, which can log IO and program data with time and date stamp
PRO-RS485	Optically isolated RS232 to RS485 converter cable
ELC-BATTERY	A standard number of days for RTC retentivity is 20-days. This unit extends that to 1-year.
PR-FLAT	Expansion PR-E Ribbon Cable 1-metre

3.6. Structure & Dimensions

3.6.1. PR-6 Series CPU and EXM-E Expansion Modules

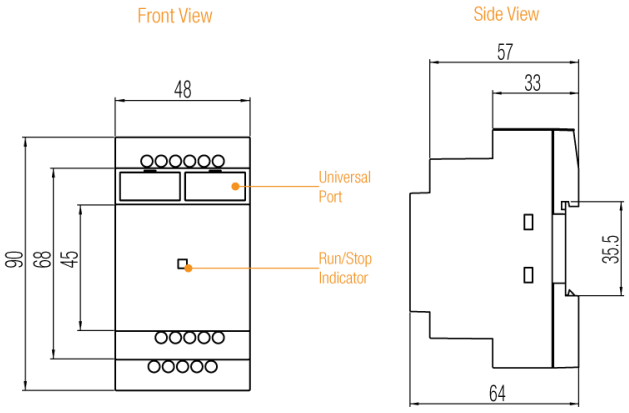


Figure 4 PR-6 Dimensions

3.6.2. PR-12 Series with LCD

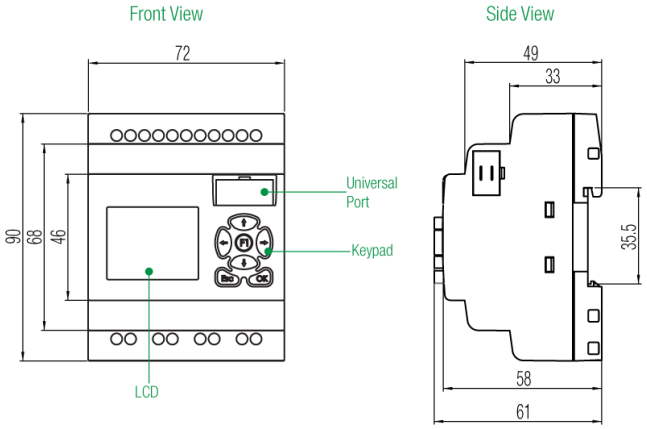


Figure 5 PR-12 Dimensions

3.6.3. PR-12 Series without LCD

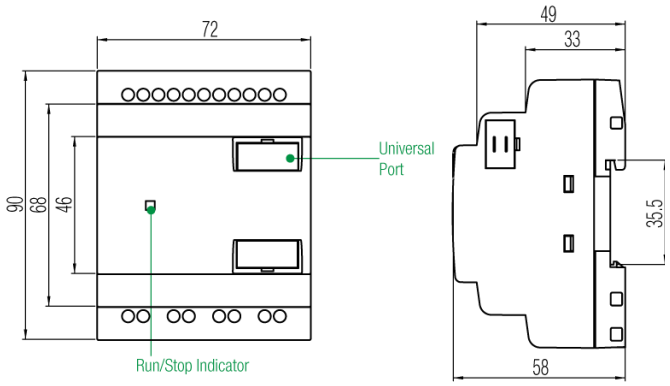


Figure 6 PR-12-E Dimensions

3.6.4. PR-14 and PR-18 Series

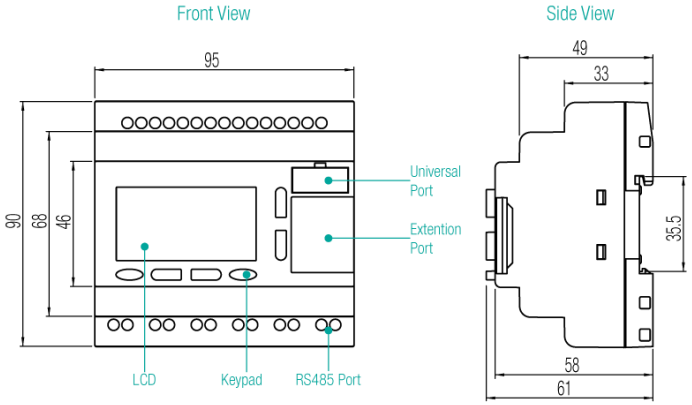


Figure 7 PR-14 Dimensions

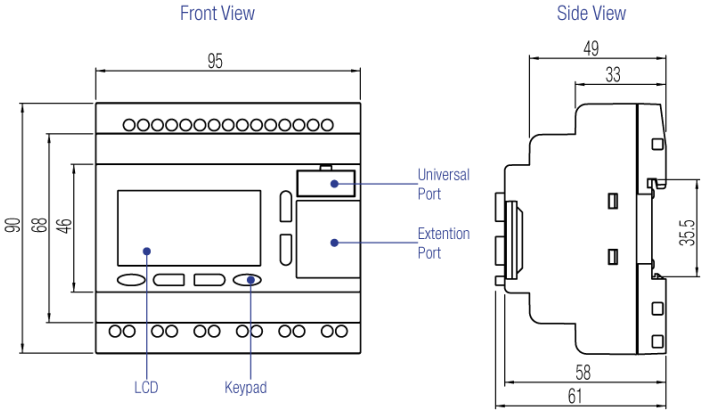


Figure 10 PR-18 Dimensions

3.6.5. PR-24 Series

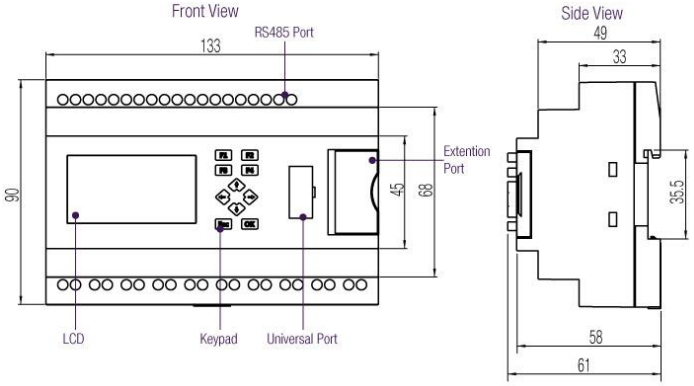


Figure 8 PR-24 Dimensions

3.6.6. PR-E Expansion Modules

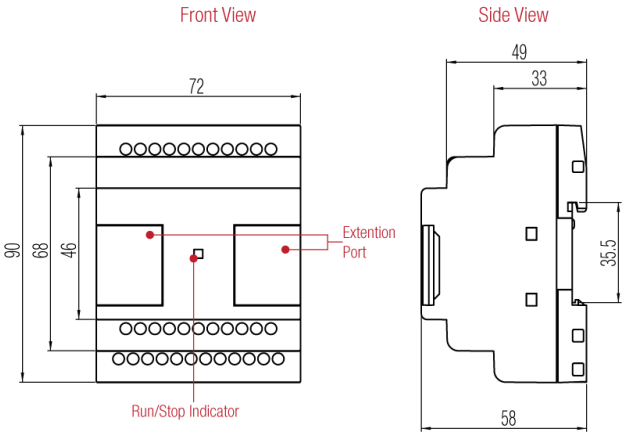


Figure 12 PR-E Dimensions

3.7. EXM GSM/GPRS Built-in Module

Each x-Messenger CPU has an integrated industrial grade Quad band GSM module inside. The x-Messenger will work with the following GSM network frequencies:

- 850 MHz
- 900 MHz
- 1800 MHz
- 1900 MHz

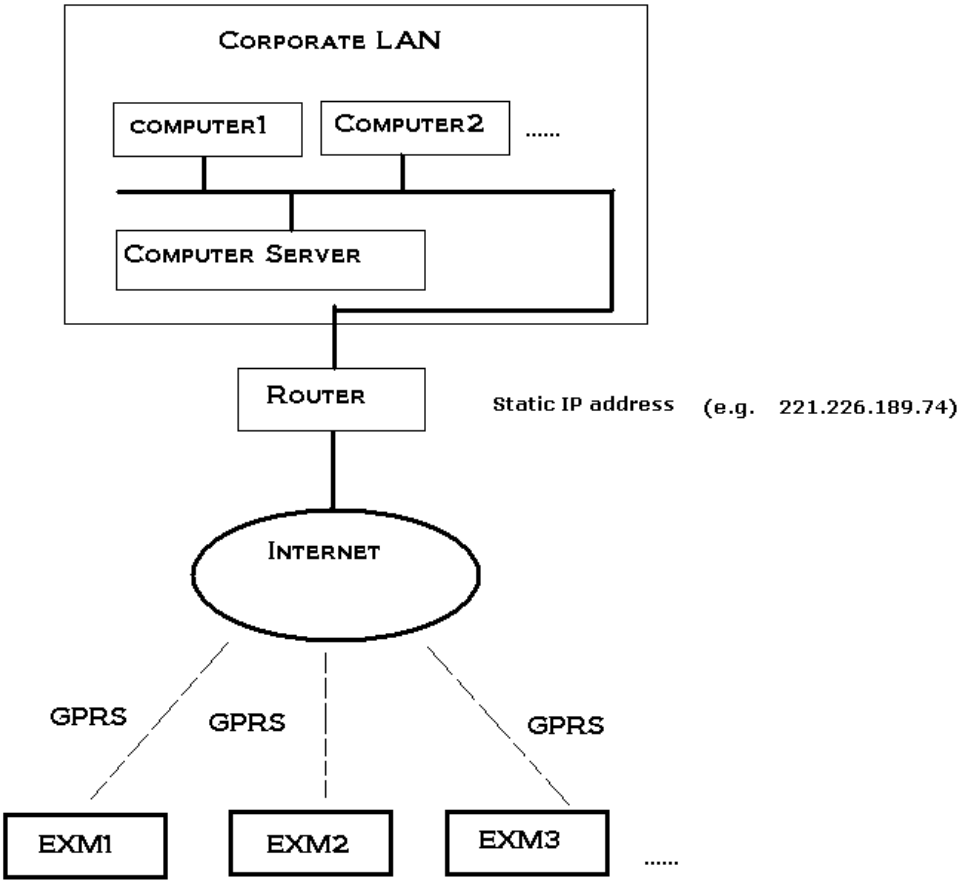
and will automatically search for a frequency to which it can establish a connection when power is applied and a SIM card installed.

NOTICE:

All x-Messenger CPUs support a GSM module except the EXM-12DC-DA-RT-WIFI.

3.7.1. How to Establish Connection between CPU and PC Using GPRS

The following connection diagram is a typical configuration when a CPU to PC connection using GPRS is required.



The EXM CPU's above will work as Clients (Masters) and the PCs will work as Servers (Slaves) in this example.

The SIM cards installed in each of the EXM CPUs will be assigned IP addresses dynamically by the mobile SIM provider each time they establish a connection to the mobile network.

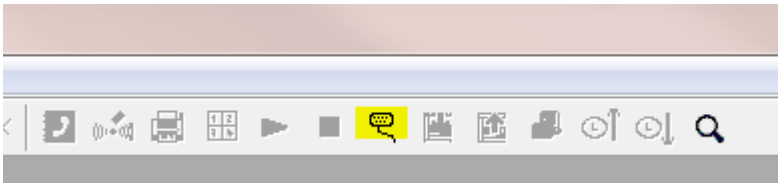
NOTICE:
The router in this example MUST have a fixed Internet IP address, which would be provided by the Internet Service Provider (ISP) used.

In our diagram example above, we will use Computer1 and EXM1 as examples in how to communicate between them using GPRS. The following configurations are applied to each device:

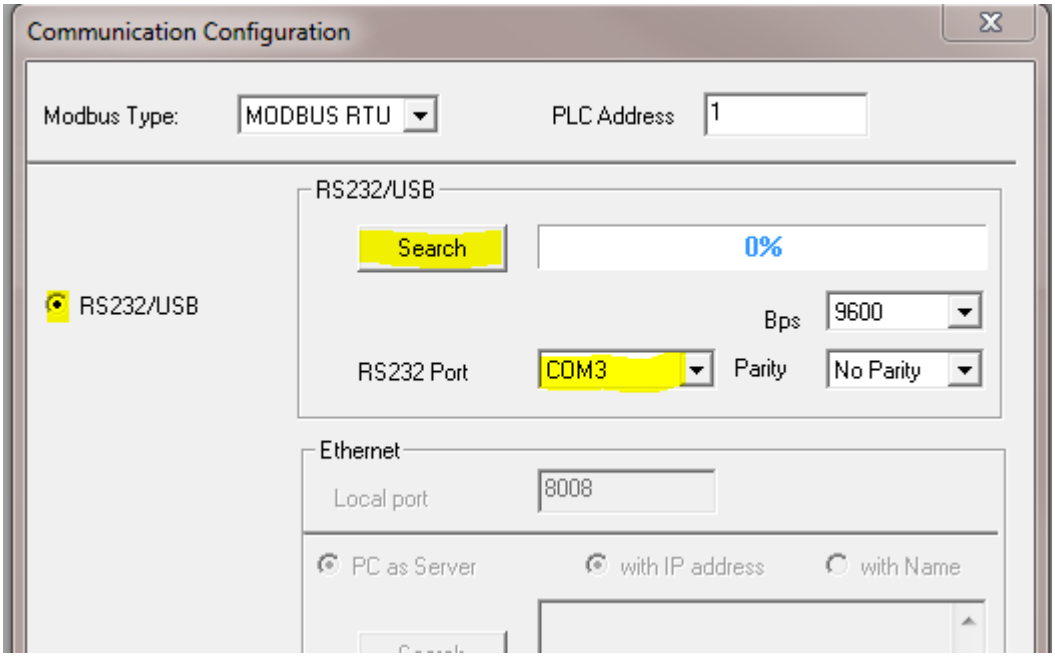
- Computer1
  - IP Address: 192.168.0.100
  - Subnet Mask: 255.255.255.0
  - Gateway: 192.168.0.1
- Router
  - LAN IP address: 192.168.0.1
  - WAN IP Address (Internet Fixed by ISP): 221.226.189.74
- EXM1 CPU
  - IP Address: Dynamically assigned by mobile network
  - Port number: 5002

Configure EXM CPU with xLogicsoft

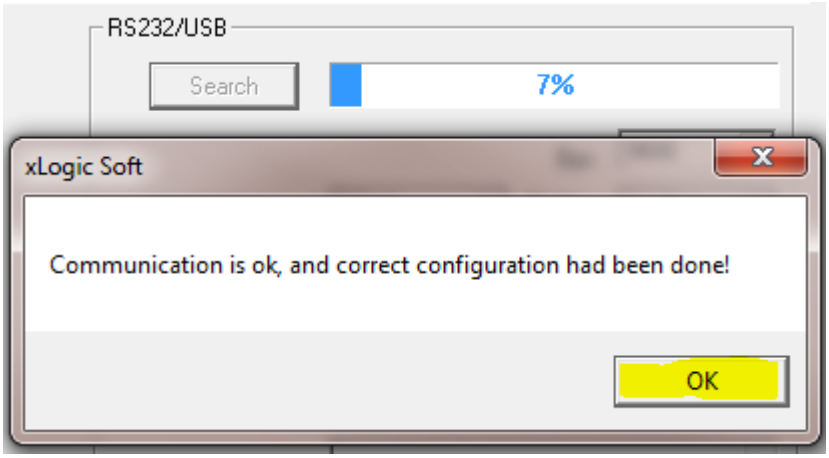
1. Connect EXM CPU to PC using a USB programming cable and power up CPU.
2. Run xLogicsoft software and click on the Open/Close Comm Port icon on the tool bar



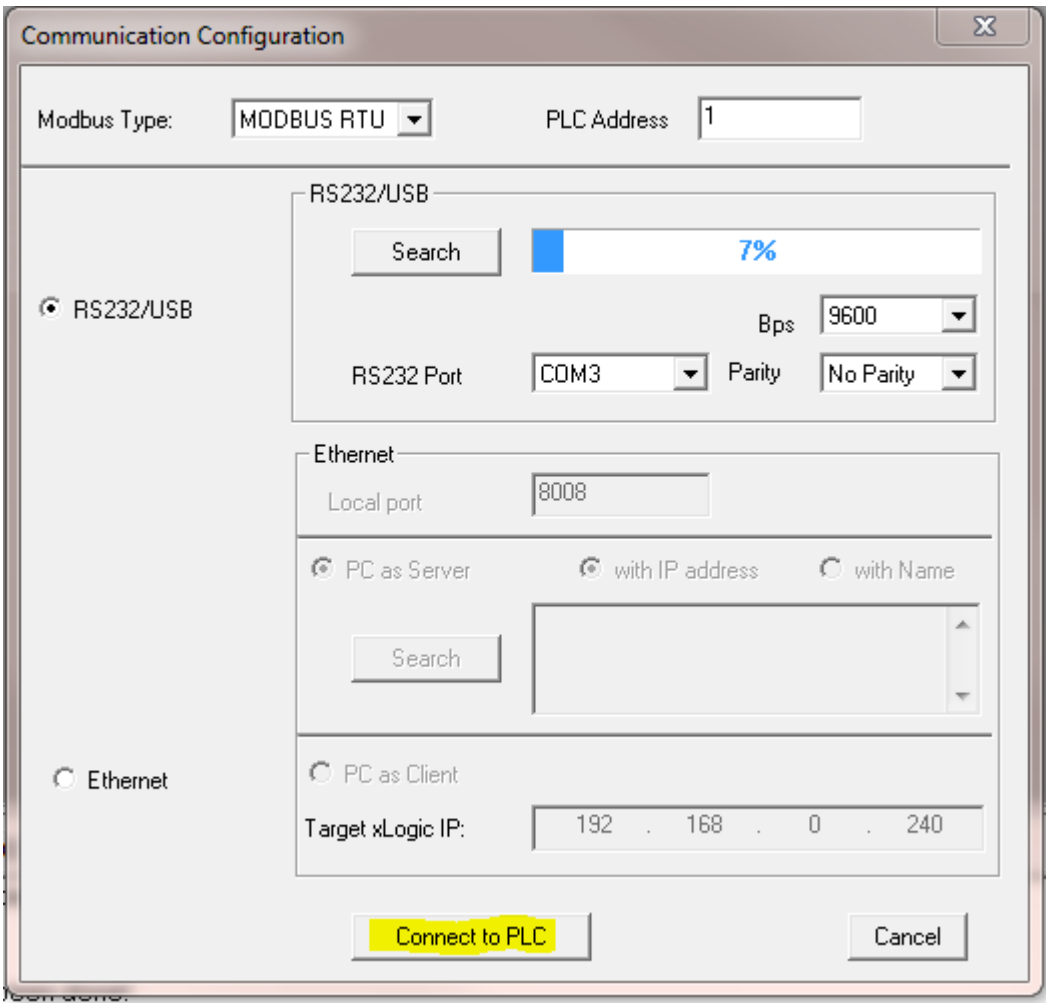
3. Ensure radio button for RS232/USB is selected, the RS232 Port is set to the correct port for the USB cable connection and press the Search button


Communication Configuration
Modbus Type:
MODBUS RTU
PLC Address
1
RS232/USB
Search
0%
RS232/USB
Bps
9600
RS232 Port
COM3
Parity
No Parity
Ethernet
Local port
8008
PC as Server
with IP address
with Name

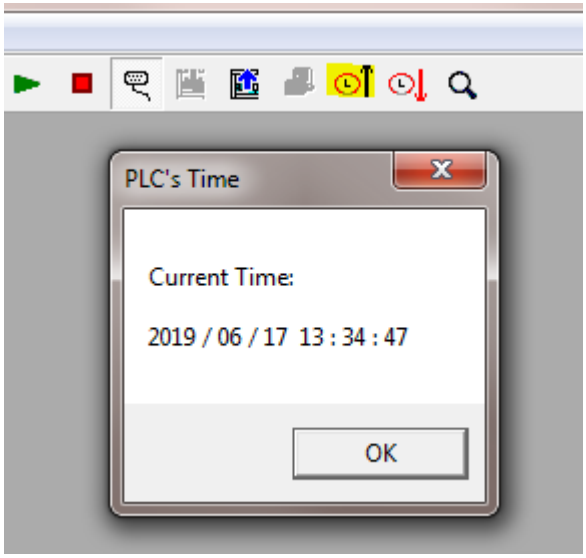
4. After a short while a message confirming communication has been established should appear. If there is a timeout or error message, check you USB cable connection and port settings in the software and repeat this step. Click OK on the communication confirmation window



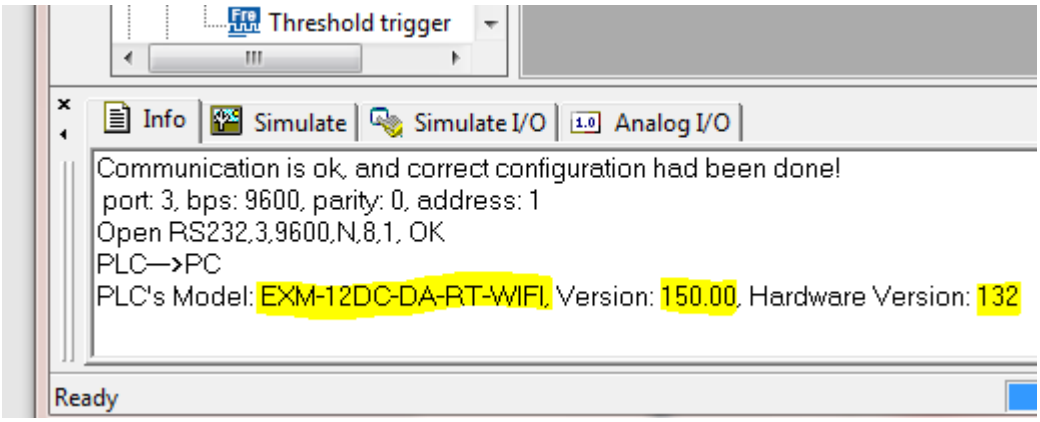
5. Click Connect to PLC button to establish connection


Communication Configuration
Modbus Type:
MODBUS RTU
PLC Address
1
RS232/USB
Search
7%
RS232/USB
Bps
9600
RS232 Port
COM3
Parity
No Parity
Ethernet
Local port
8008
PC as Server
with IP address
with Name
Search
Ethernet
PC as Client
Target xLogic IP:
192 . 168 . 0 . 240
Connect to PLC
Cancel

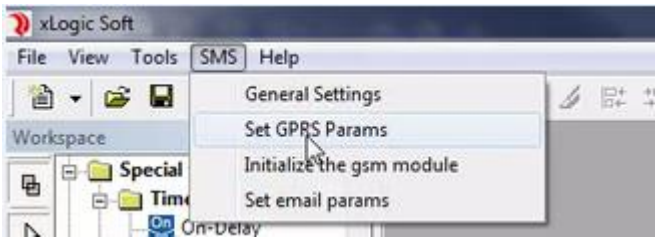
6. Confirm connection has been successful by requesting date and time from the CPU by clicking on the Get PLC Time icon



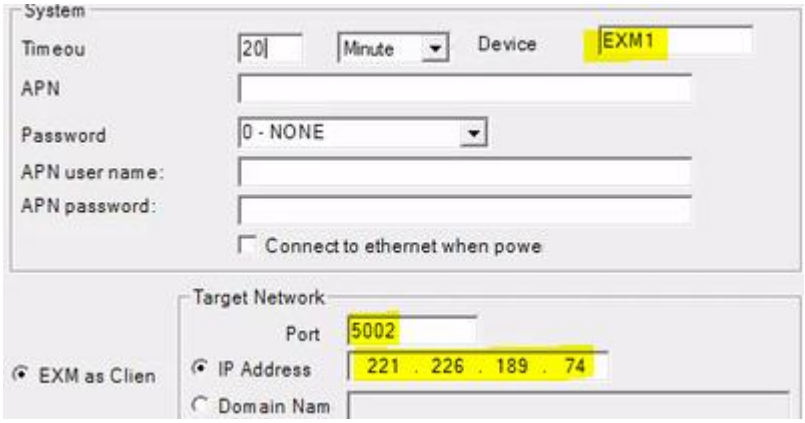
You should also see a message in the Info tab section of the software, which should display the Model, hardware and firmware version of the EXM CPU too



7. Configure GPRS by selecting SMS...Set GPRS Params from the menu



8. Enter the GPRS information in the window that appears. Using our example, the data entered will be as shown below

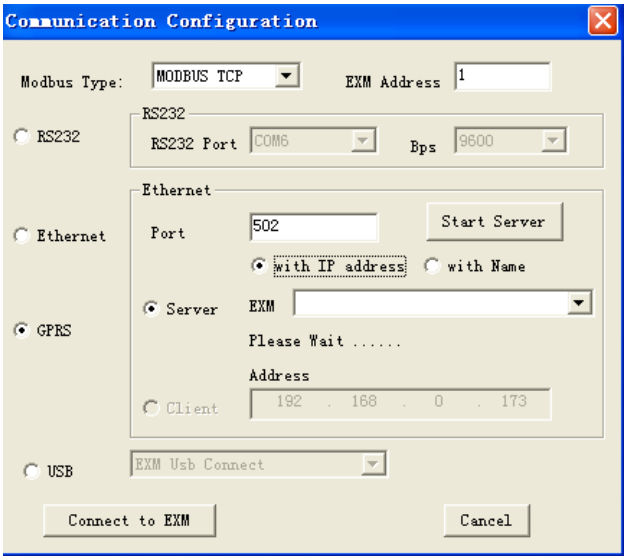


9. If you know them, you may also enter the mobile providers SIM details here at this time too, otherwise click the Write button to save your settings to the EXM CPU.

NOTICE:

The router in this example MUST have a fixed Internet IP address and the Port selected MUST allow traffic through the Router's firewall.

10. Once a SIM card has been added to the EXM CPU and the providers SIM settings have been entered and saved to the CPU, you will then be able to communicate with it via GPRS by using xLogicsoft. Close the communication link you used to configure the EXM CPU via USB cable and re-start the communication and select GPRS rather than RS232

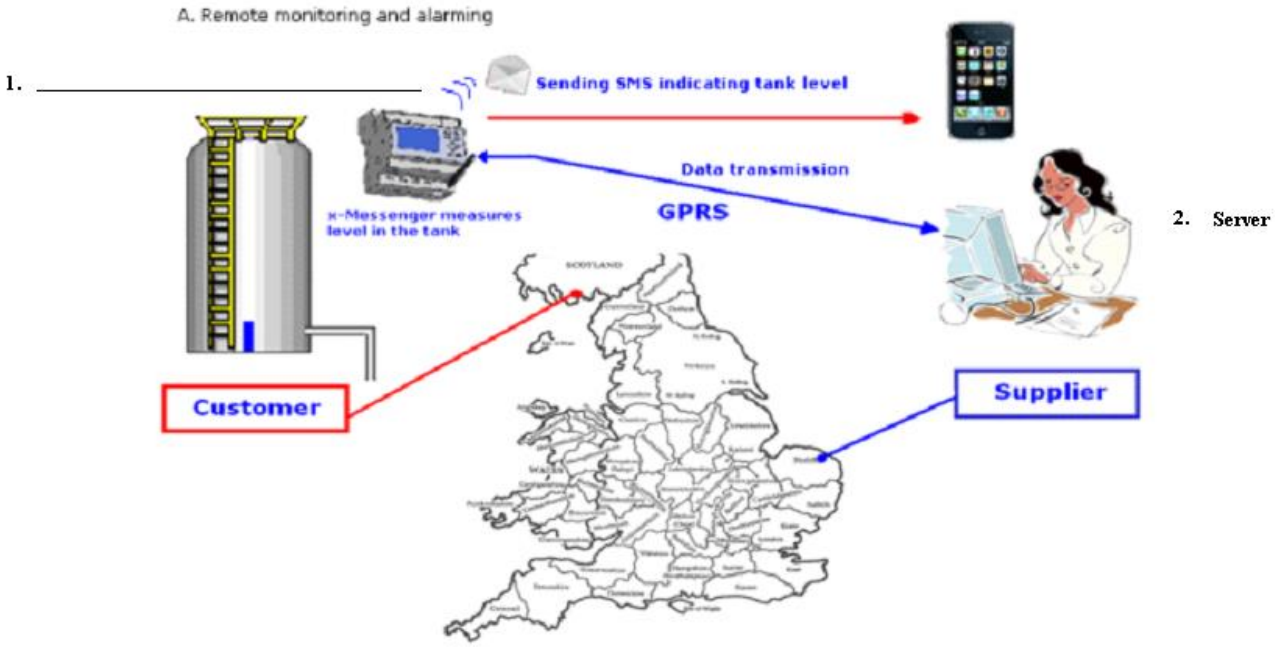


Ensure the port number is set to 5002 if this was used as in our example and press the 'Start Server' button to create a GPRS connection between the PC and the EXM CPU. After connection is established, you are able to upload/download programs and remotely monitor the CPU from anywhere in the world!!!

NOTICE:

At least one end of the GPRS connection needs to be a FIXED IP.

Example of CPU Sending SMS and Central Office Requesting Data via GPR



When EXM CPU is acting as Server with a FIXED IP address and office Client software requesting information such as level, temperature, digital states, etc.
The client software would use Modbus/TCP commands to access the remote data.
More information on the Modbus/TCP protocol in our Rievtech CPUs including register addresses can be found in our Modbus TCP communication document.

If you use the Modbus programming blocks in the xLogicsoft in your EXM CPU program, you can get the CPU to send data to a PC software server, such as analogue and digital information. Use the two GPRS function blocks, GPRS Connect and GPRS Data Upload in the xLogicsoft software, you can configure the CPU to send data periodically to the server software.

3.7.2. Configuration of E-mail Settings

To configure the EXM CPU to send emails, you need to first configure the GPRS parameters as detailed in previous section including any APN settings provided by your mobile SIM provider.

When using the email sending features of the EXM CPU there is no need to check the tick box 'Connect to Ethernet when power on' option.

Once you are connected with xLogicsoft to the EXM CPU use the menu option SMS...Set email params. Enter data into the window shown:

- SMTP: - Enter the IP address of the SMTP Server you are using
- Port: - Enter the port number of the SMTP Server you are using
- User: - Enter the username for SMTP Server email account authentication

- Password: - Enter the password for the SMTP Server email account authentication
- Sender Name: - Enter a name that the receiver will recognise such as Tank Farm Monitor
- Sender Email: - Enter an email address for the EXM

Click the Write button to save the settings to the EXM CPU.

With the settings correctly applied, your EXM CPU is now able to send emails using the SMS Message Output function block within your xLogicsoft program.

3.7.3. Changing CPU Information via SMS

You can change all sorts of information in your CPU by sending an SMS to it. The SMS structure is important and needs to be configured in the following manner:

- Each message MUST start with the characters: **##****
- Each instruction/parameter change MUST be terminated with the characters: **;**

Example of a formatted SMS message:

```
##**
AF1=2;
AQ02=100;
Q01=1;Q12=0;
F1=1;F2=0;
AM1=10;
REG1=100;
REG100=300;
TEL,B002,1,B002,2,B003,1,B003,5:13851448223;
RTC=19-07-09,01:32:09,0;
```

The above message once received by the CPU will carry out the following changes:

Set AF1 flag to value of 2

Set AQ02 to 100

Set Q01 to ON and Q12 to OFF

Set F1 flag to ON and F2 flag to OFF

Set register AM1 to 10

Set REG1 to a value of 100

Set REG100 to a value of 300

Set Receiver Telephone number configured in block B002 1st and 2nd receiver and B003 1st and 5th receiver to 13851448223

Set real time clock to date (yy-mm-dd) 9th July 2019 and to time (hh:mm:ss) 01:32:09 and week begins day 0 (0= Sunday, 1=Monday, etc.)

Using the ALL Function

4. Installing and Removing Rievtech Hardware

4.1. Dimensions

The dimensions of the Rievtech xLogic hardware is compliant with DIN 43880.

xLogic can be "snap-mounted" to 35mm (EN 50022) DIN rail or wall mounted.

Table 9 xLogic Dimensions

Model	Width (mm)
EXM-E Expansion module	48
PR-6 CPU	48
PR-E Expansion module	72
PR-12 CPU	72
PR-14 CPU	95
PR-18 CPU	95
ELC-22 CPU	133
PR-24 CPU	133

Always remove power whilst adding/removing an expansion unit.

4.2. DIN Rail Mounting

The following procedure should be used to install/uninstall your xLogic CPUs and expansion units on DIN rail.

4.2.1. Mounting Your Hardware

1. Hook the top of the xLogic CPU onto the DIN rail.
2. Push down the lower end to snap it into place. The mounting catch MUST engage.
3. Hook the top of the xLogic expansion unit onto the DIN rail.
4. Slide the expansion unit toward the CPU until they are touching. Make sure the ribbon cable does not get caught between it and the CPU.
5. Push down on the lower part of the expansion unit to snap it in place, ensuring the mounting catch engages.
6. Lift-up the expansion port cover on the CPU.
7. Push the connector of the ribbon cable into the socket on the CPU ensuring the correct alignment of the pins and close the cover when done.
8. Repeat the procedure to add further expansion units if required.

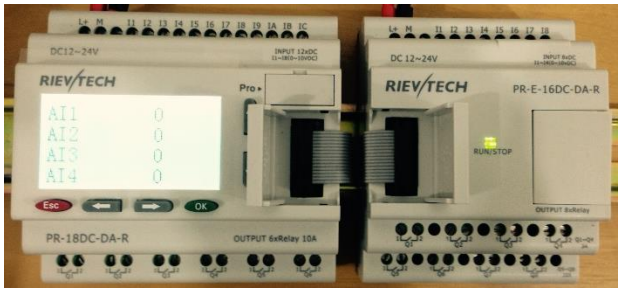


Figure 13 Connection the CPU with expansion

NOTICE:

If you need to install the CPU and expansion units on different DIN rails, then you will need a PR-FLAT expansion cable. The maximum distance between CPU and expansion unit is 200 metres.

4.2.2. Removing Your Hardware

1. Insert a screwdriver into the eyelet at the bottom of the sliding interlock and move the latch downwards.
2. Slide the expansion module to the Right.

3. Using a screwdriver in the eyelet, carefully lever it downward.
4. Lift the expansion unit off the DIN rail swinging the unit up from the bottom.
5. Repeat steps 1 to 4 for all other units.

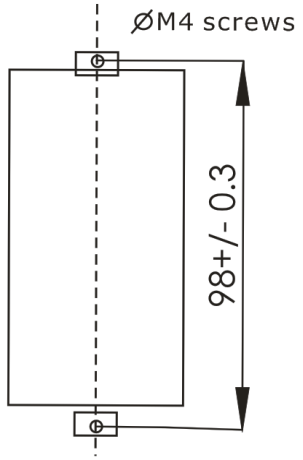


Figure 9 PR-6 Wall Mounting

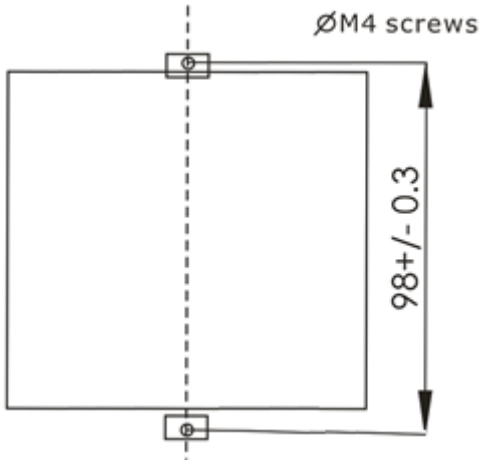
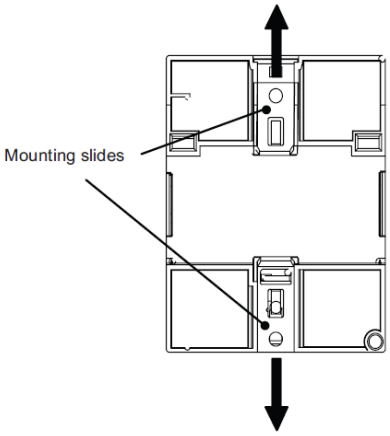


Figure 10 PR-12 Wall Mounting

4.3. Wall Mounting

For wall-mounting, first, slide the mounting slides on the rear side of the devices towards the outside. You can now wall-mount xLogic by means of two mounting slides and two M4 screws (tightening torque 0.8 to 1.2 Nm).



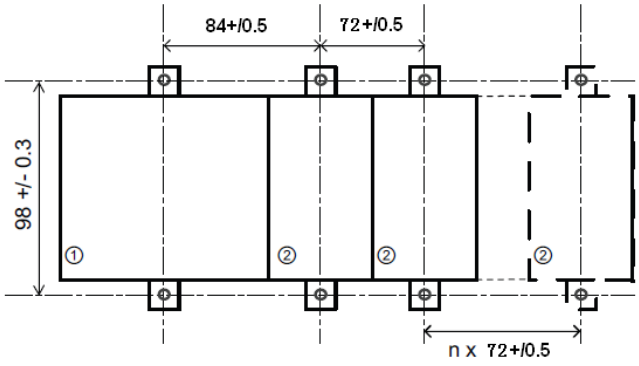


Figure 11 PR-18 & Expansions Wall Mounts

4.3.1. Drilling Templates for Wall Mounting

Use the templates below for mounting your Rievtech hardware on a wall.

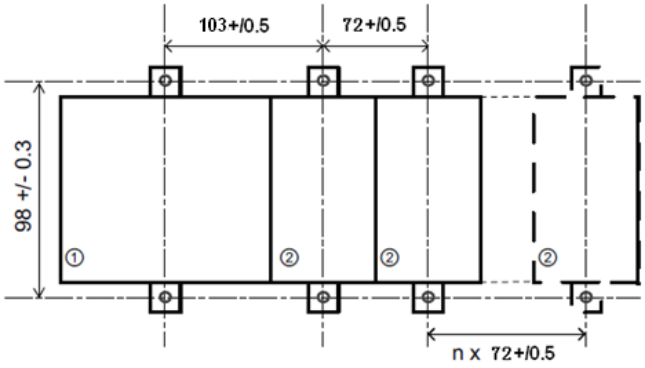


Figure 12 PR-24 Wall Mountings

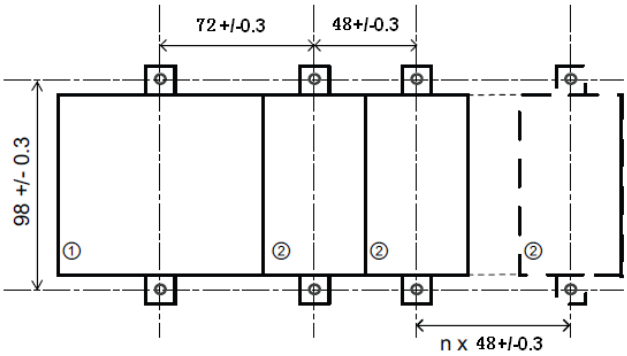


Figure 13 PR-12-N Wall Mountings

4.4. Wiring

Use a screwdriver with a 3-mm blade or adjusting Rievtech hardware terminals. You do not need wire ferrules for the terminals. You can use conductors with cross-sections of up to the following thicknesses:

- 1 x 2.5 mm^2
- 2 x 1.5 mm^2 for each second terminal chamber
- Tightening torque: 0.4... .0.5 N/m (3. .4 lbs/in)

NOTICE:

Always cover the terminals after you have completed the installation. To protect against accidental contact with live parts. Local electrical standards must be complied with.

4.4.1. Connecting Power

The following hardware is rated for operation with 110/240 VAC supply:

- PR-6AC, PR-12AC, PR-14AC, PR-18AC, PR-24AC and PR-E-16AC

The following hardware is rated for operation with 12/24 VDC supply:

- PR-6DC, PR-12DC, PR-14DC, PR-18DC, PR-24DC and PR-E-16DC

NOTICE:

A power failure may cause an additional edge triggering signal. Only data from the last uninterrupted cycle will be stored.

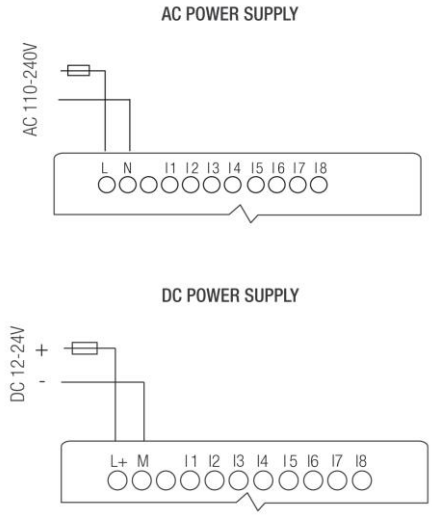


Figure 14 Power Connection

4.4.2. Connecting Input Signals

Inputs can be connected to various sensor elements such as momentary switches, switches, light barriers, daylight control switches, etc.

Table 10 Input Status

Signal state	AC Type	DC Type
OFF/0 **Input Current**	< 40 VAC < 0.03 mA	< 5 VDC < 0.1 mA
ON/1 **Input Current**	> 79 VAC Typical 0.06 mA	> 10 VDC Typical 0.3 mA
Analogue Input	NOT APPLICABLE	0 to 10 VDC

NOTICE:

For PR-6DC-DA-R, PR-12DC-DA, PR-14DC-DA, PR-18DC-DA, PR-24DC-DA Series and versions can accept analogue inputs. The type of signal connection can be set within the user program by selecting ether Analogue Input or Digital Input for that channel. They will be recognized as analogue inputs when the input terminal is connected with an analogue function block, otherwise, they will be recognised as digital inputs.

NOTICE:

The analogue inputs require DC 0V ~ +10V voltage signals. These are divided equally into 0.01 VDC increments. In programming, all the block parameters related to the analogue inputs are based on the minimum increment of 0.01 VDC.

NOTICE:

If the input voltage is greater than 10 VDC, they will NOT be recognised as analogue inputs.

NOTICE:

The time taken for a change of digital state, 0 to 1 or 1 to 0, must be greater than 50ms.

4.4.3. DC Type Digital Input Connections

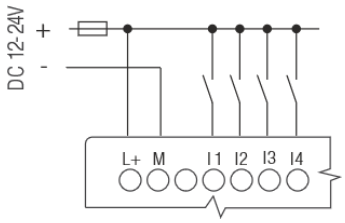


Figure 15 DC Inputs

4.4.4. AC Type Digital Input Connections

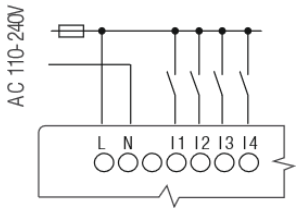


Figure 16 AC Inputs

4.4.5. Analogue Input Connections (0 to 10 VDC)

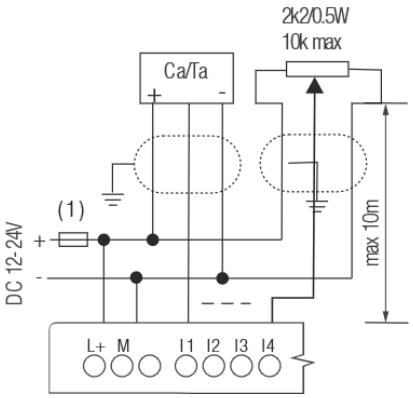


Figure 17 Analogue Voltage Inputs

4.4.6. Analogue Input Connections (0 to 20 mA)

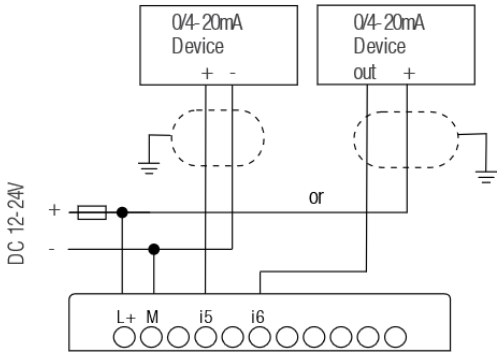


Figure 18 Analogue Current Inputs

4.4.7. Analogue Input Connections (PT100)

Either 2 or 3-wire PT100 sensors can be connected.

With a 2-wire sensor, connect to terminals M1+ and M1-. Short between M1+ and IC1. There is no compensation for any impedance in the wire. A measurement error of 1 Ω is equivalent to +2.5 °C.

Using a 3-wire sensor can inhibit any influence caused by cable length.

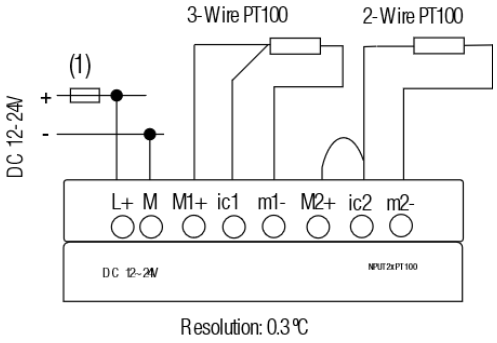


Figure 19 Analogue PT100 2-, 3-Wire Inputs

4.4.8. Connecting Output Signals

Using a Digital Output – Relay Type

Various loads such as a lamp, fluorescent tube, motor, contactor, etc., can be connected to the relay outputs of the Rievtech hardware. The maximum output current that can be handled by the relay is 10A for the resistance load and 3A for an inductive load. The relay output connection should be wired as per the following figure:

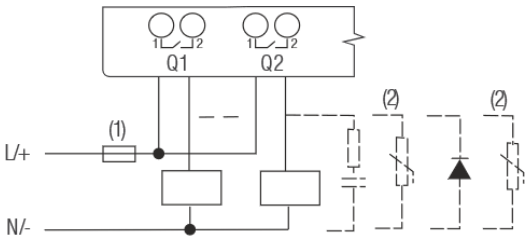


Figure 20 Relay Digital Output

Using a Digital Output – Transistor Type

The load connected to a transistor output must have the following characteristics:

***** The maximum switching current cannot exceed 0.3A.

***** When the switch is ON (Q=1), the maximum current is 0.3A.

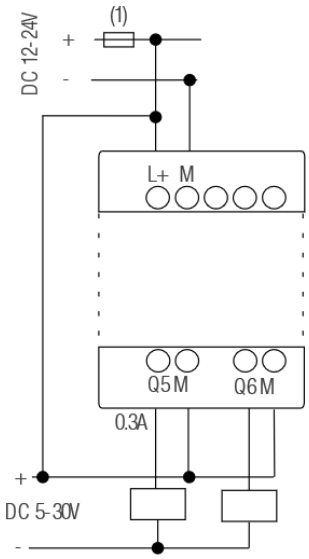


Figure 21 Transistor Digital Output

NOTICE:

* The load connecting voltage must be ≤60 VDC and it must be DC.

* The "+" terminal of the output wiring must be connected to the DC positive voltage, and it must be connected with the "L+" terminal of the CPU power terminal. The load must be connected with the "-ve" terminal of the DC negative voltage supply.

Using an Analogue Output

PR-E-AQ-VI (0 to 10 VDC)

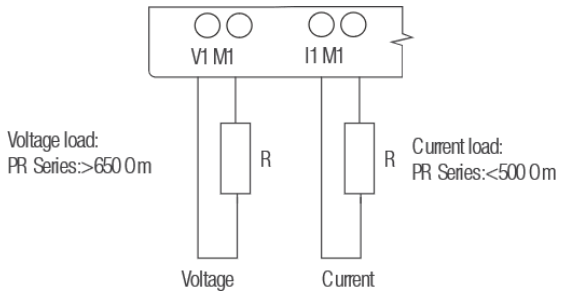


Figure 22 Analogue 0-10V and 0-20mA Output

EXM-E-RS485/PR-RS485

The RS485 expansions are just converters with photo isolation providing 3-sets of wiring terminals (although there are 3-sets of terminals, they are connected within the unit, so only one channel RS485 is available) of RS485 port (2x8pin) of CPU(PR-18/ELC-22/PR-24) for easy connection with other devices.

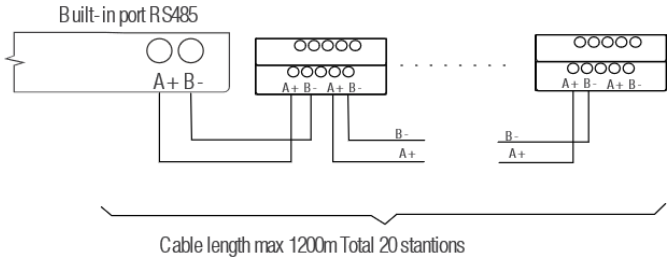


Figure 23 RS485 Connection

If "RT1", RT2" terminals are short connected, one 120R resistor will be connected between A/+ and B/- providing a termination resistor for end-of-line purposes.

4.5. Communication Port Instructions

PR-6 CPUs

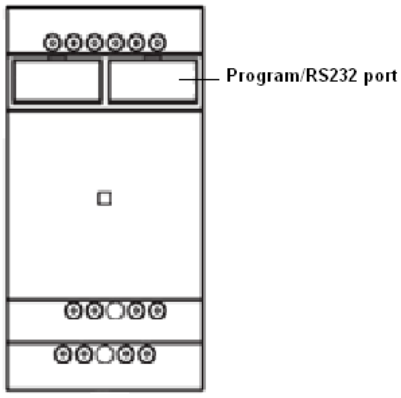
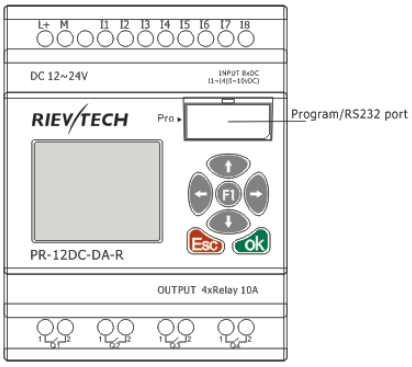
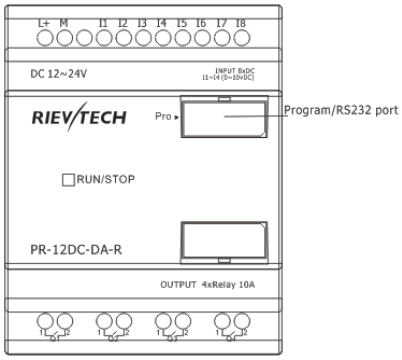


Figure 24 PR-6 CPU Ports

PR-12 CPUs

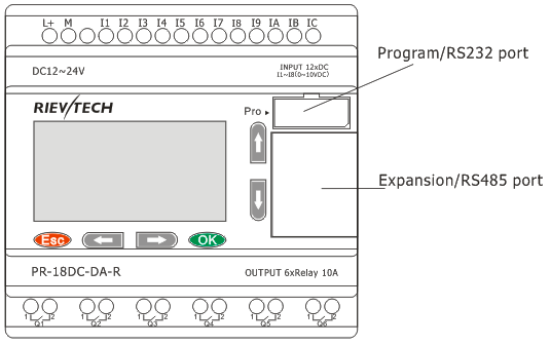


With HMI



Without HMI

PR-14, PR-18, ELC-22 and PR-24 CPUs



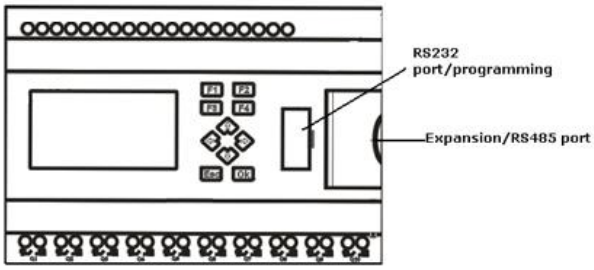


Figure 25 Other CPU Ports

The programming port/RS232 port can be used with any of the following Rievtech devices. ELC-RS232, ELC-USB, ELC-Copier, ELC-MEMORY, ELC-BATTERY, and PRO-RS485. This is COM 0.

When the programming port is being used as the standard RS232 port (D-shape 9 pin header). The ELC-RS232 cable is needed. Below shows you the pinouts for the device:

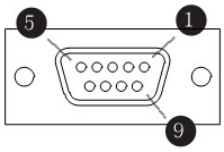


Figure 26 RS232 Pinouts

Expansion port/RS485 (pin definition)

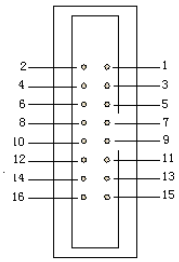


Figure 34 RS485 Pinouts

3------RS485 A 7------CANL

5------RS485 B 9------CANH

4------GND 15------+5V

6------GND 16------+5V

Communication between CPU and expansion module will use pins 4, 7, 9 and 15.

PR-RS485 module is required when either the PR-18/ELC-22/PR-24 CPU is required to communicate with a third-party device via RS485. Note the PR-24 has a built-in RS485 too.

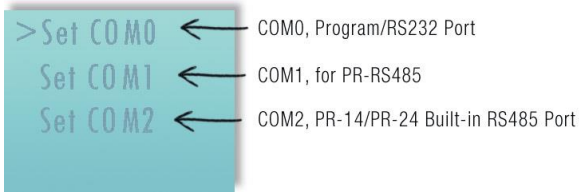


Figure 27 COM Port Details on LCD

The programming port/RS232 port can be used with any of the following Rievtech devices. ELC-RS232, ELC-USB, ELC-Copier, ELC-MEMORY, ELC-BATTERY, and PRO-RS485. This is COM 0.

PIN	Function
2	RXD
3	TXD
5	GND
others	NULL

When the programming port is being used as the standard RS232 port (D-shape 9 pin header). The ELC-RS232 cable is needed.

5. Parameter Modification using CPU HMI

5.1. Overview of CPU Menu Structure

The PR-12, PR-14, PR-18, PR-24, PR-26, EXM-12, ELC-12-N and ELC-22-N have the same menu structure. We will use a PR-18 in the following examples.

5.2. Parameter Assignment Mode

Parameter assignment menu :

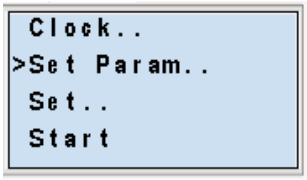
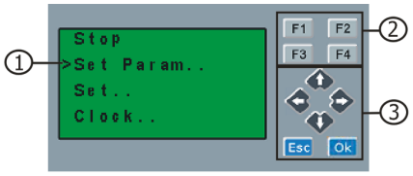


Figure 28 CPU Parameter Menu

5.3. LCD Panel Instructions

PR-12 Series CPU

PR-18 Series CPU



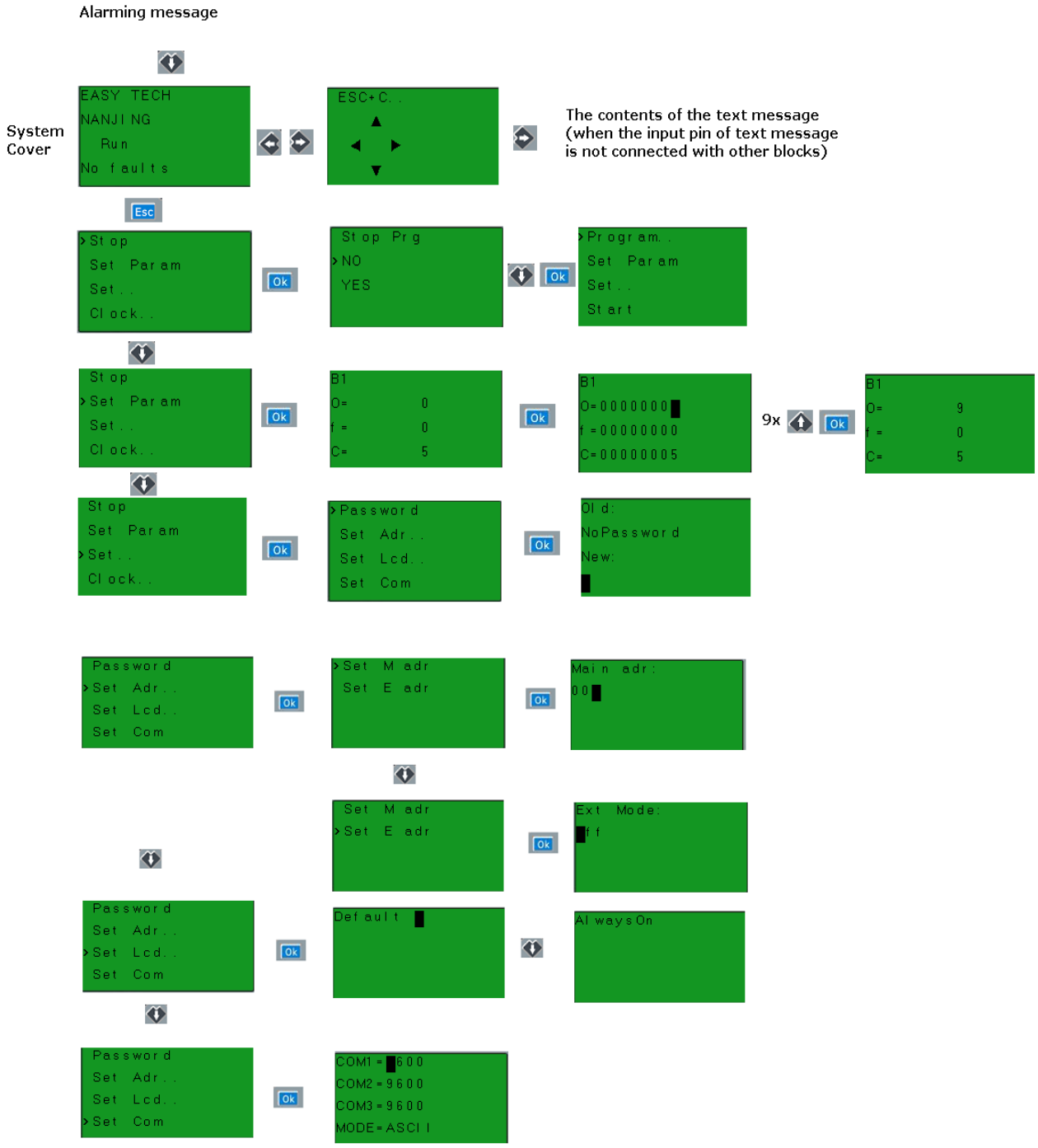
ELC-22 and PR-24 series CPU

1. **Display Area** – 4 x 16 characters can be displayed
2. **4 Function Keys** – Only on models ELC-22-N and PR-24 Series CPUs.
3. **6 Panel Keys** – Used to navigate menu and view alarm messages.

This panel allows the user to carry out the following tasks:

- Display the RUN/STOP status of the CPU
- Display/modify the Clock
- Display IO status of CPU and expansion units
- Display alarm messages
- Display program register values
- Modify some types of register parameters
- Control of backlight using the program, Light on Alarm for example
- Initial system display message can be customized
- Up to 64 different alarm messages can be displayed
- Modify the CPU ID address
- Set password protection
- Change communication parameter for COM ports

LCD Menu Structure



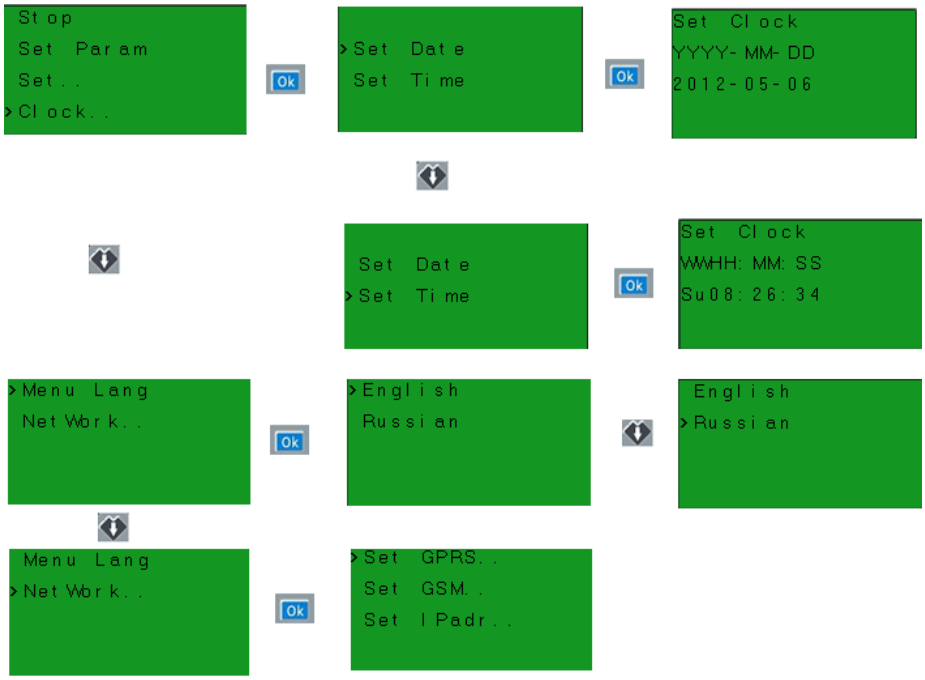
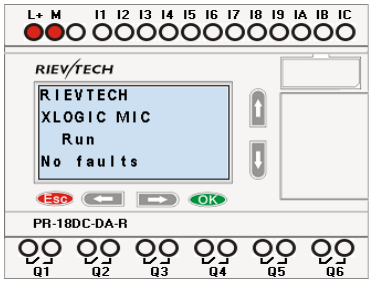


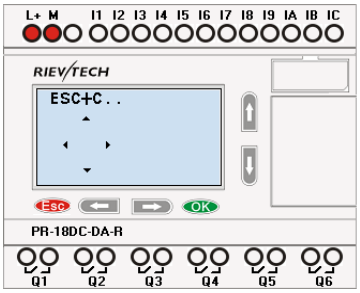
Figure 29 LCD Menu Structure

After being powered on, the CPU will check to see if a valid program is stored within it.

If the program is accurate, then the CPU will be running, meanwhile, the system cover will show as follows, this can be customised within xLogicsoft, this is the default initialization screen.



If there are several parameter pages, users can press ◀ or ▶ keys to go to the page they require. The last page is the cursor mode:



Cursor keys can be controlled in this page by pressing arrow keys and ESC key at the same time.

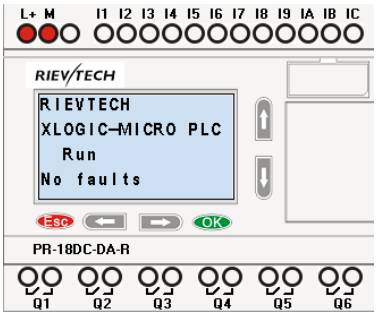
If xLogic has several alarms in the same period, it only displays the message with the highest priority in the function block. You may cycle through all current alarm messages by pressing ▲ or ▼ key.

NOTICE:

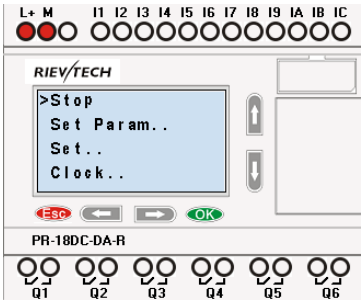
The message text block within xLogicsoft is treated as a parameter page only when it has no input connected to it, otherwise, it will be regarded as an alarm page. When the input pin sees a rising edge, the LCD will display an alarm message.

5.4. Selecting a Function Page

Press ESC key to change from running mode to function page mode.



After pressing ESC key, xLogic switches to function page and the function menu will be shown as below.



5.4.1. Function Page Options

Run/Stop

Select this menu to switch the xLogic status between RUN and STOP. Refer to section 4.3.1 for details.

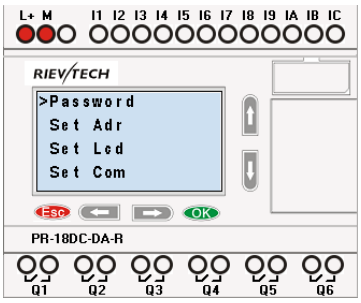
Set Param

To set/change a function block parameter. Refer to section Set Parameter for details.

Set...

Used to set/modify a password (see Set Password), set an address of CPU or expansion unit (see Expansion Addressing), LCD settings (see LCD Settings) and communication parameters settings (see Communications) for details.

1. Press "UP" or "DOWN" key to move the cursor to "**Set...**"
2. Then press the **OK** key, xLogic will display as follows:



Clock

To set and modify date and time. Refer to section Set Time & Date for details.

Network (This only for Ethernet CPU)

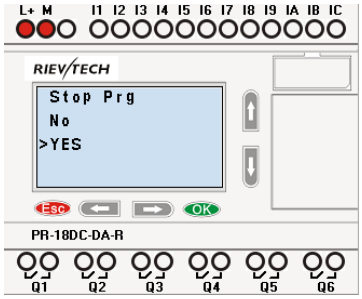
Recovery ET: Pressing OK whilst the cursor is on this option will be reset the CPU to default settings.

5.4.2. How to Switch Between RUN and STOP

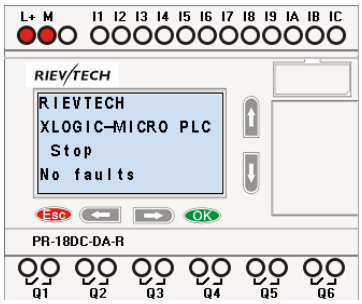
You should first select a FUNCTION PAGE. (Read Select Function)

1. Move the cursor to “Run/stop”: Press “UP” or “DOWN” key.

2. Move the cursor to "Yes": Press OK key.

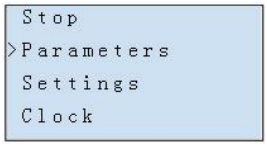


After pressing ESC key, you’ll find out your program has changed to “stop” status as shown below:



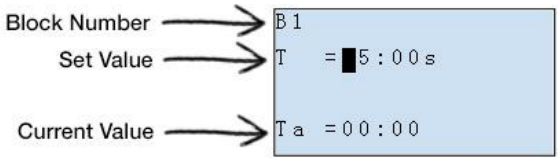
5.4.3. Set Parameter

1. Under the FUNCTION PAGE, select “Parameters”: Press ▲ or ▼ key



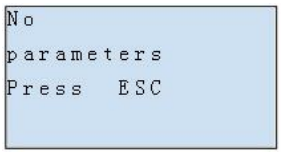
2. Confirm by pressing the **OK** key.

Then xLogic displays the first parameter, so you can modify as you like. If there is no parameter to set/modify, you can press ESC key to return.



No parameter to edit, press ESC key to return

3. Select the parameter you intend to modify.
4. Select a specific value of that parameter which you want to edit, then press the OK key.

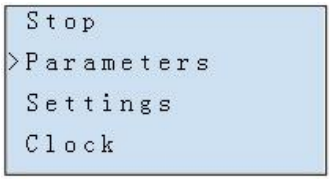


How to modify parameter?

A. First, select a parameter which you need to edit by following the procedures below:

1. Under the FUNCTION PAGE, select

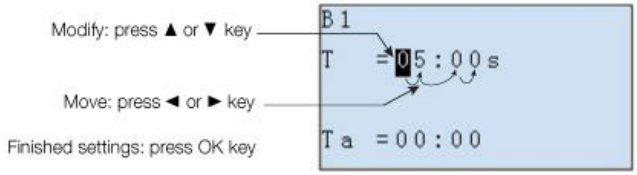
"Parameters": Press ▲ or ▼ key



2. Confirm by pressing the **OK** key.

B. Move to the parameter, modify value and press OK to confirm:

1. Move the cursor to the parameter to be modified: press ◄ or ► key.
2. Modify value: press ▲ or ▼ key.
3. Confirm the modification by pressing the OK key.

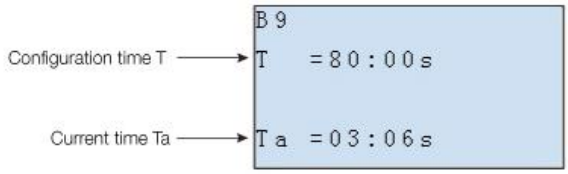


NOTICE:

When xLogic is running, not only the time value b lso the units of time (S, M, H) can be altered. You can alter the time base (s=second, m=minute, h=hour) during runtime.

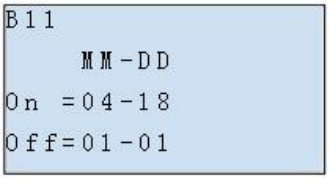
The current value of time T

View time T in parameter mode:



You can modify the configuration time or switch on/off time for a particular time segment.

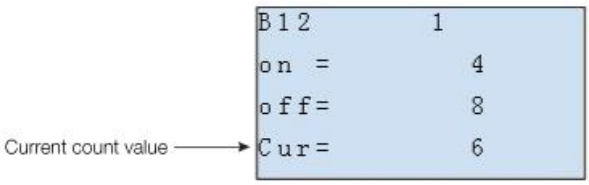
In parameter mode, time segment figure of a timer:



You can alter the time and date of the on/off switch.

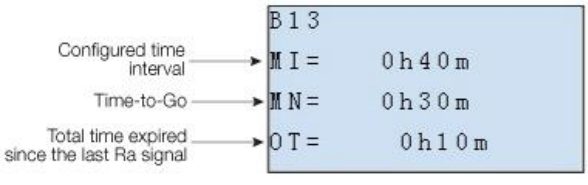
The current value of the counter

In parameter mode, the parameter view of a counter:



The current value of hour counter

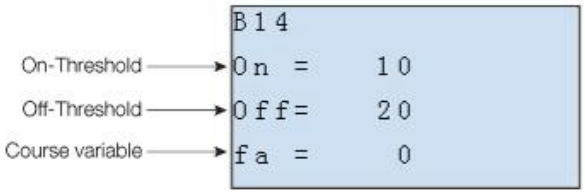
In parameter mode, the view of the hour counters:



You can edit configured time interval (MI).

The current value of the threshold trigger

In parameter mode, the view of threshold trigger:



You can alter the threshold values for on/off.

5.4.4. Set Password

You can set a password to protect your program within the xLogic CPU.

Set a new password:

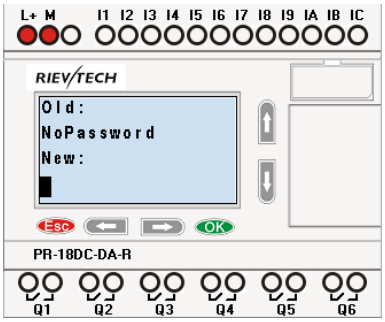
A password contains up to 4 characters and each character is a number from 0 to 9. You can specify, edit or remove a password directly in xLogic using the "Password" menu of the function page:

You should first select a FUNCTION PAGE. (Read 4.3)

1. Move the cursor to "Password" by pressing the "UP" or "DOWN" key.

2. Confirm "Password": Press OK key.

Example: Let's set "1234" as the password for a program. Now the LCD displays the following interface:



Select "New" option, and then edit it.

3. Select "1": press "UP" ▲ key once.

4. Move the cursor to the next character: press "Right" ▶ key.

5. Select "2": press "UP" ▲ key twice.

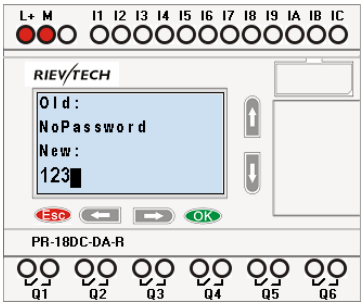
6. Move the cursor to the next character: press "Right" ▶ key.

7. Select"3": press "UP" ▲ key three times.

8. Move the cursor to the next character: press "Right" ▶ key.

9. Select "4": press "UP" ▲ key four times.

Now display:



10. Confirm password: press OK key.

Now, the program is protected by the password of "1234", and then you can return to the main menu.

NOTICE:

You can cancel a password newly-set by pressing the ESC key. In this instance, xLogic will return to the main menu without saving the password. This password protects unauthorised changes to ay parameters, which means that to make any changes via the LCD, the password is required. The program password should be set in xLogicsoft by the menu File->Property-> Parameter to protect the program itself from unauthorised changes/copying.

Modify password:

To modify the password, you are required to enter the current password first.

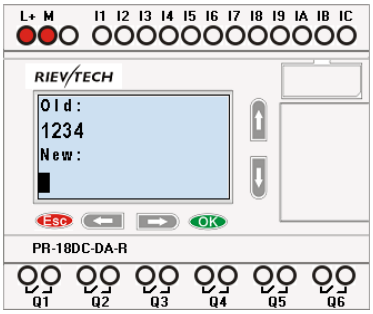
In the menu of the FUNCTION PAGE:

1. Move the cursor to "Password": Press "UP" or "DOWN" key.

2. Confirm "Password": Press OK key.

Select "Old" and input current password (in our instance is "1234"), the process is the same as the step 3 to step 10 mentioned above.

LCD displays:



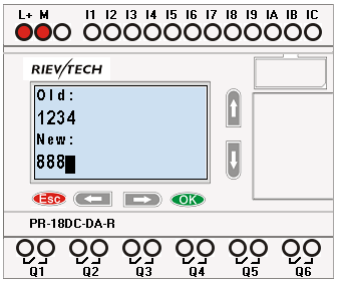
Select "New" to input new password such as "8888":

3. Select "8": press "UP" key.

4. Move the cursor to the next character: press "Right" key.

Repeat the step 3 and 4 to enter the third and fourth characters.

LCD displays:



5. Confirm new password by pressing the OK key.

How to remove the password:

If you need to remove the password e.g. allow the other users to edit your program, then you must know the current password. The process for removing a password is the same as that for modifying it.

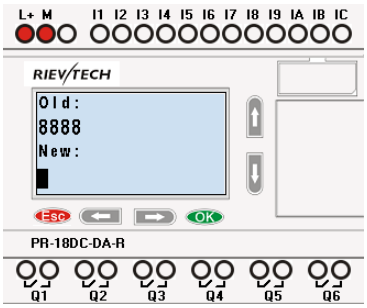
In the menu of the FUNCTION PAGE:

1. Move the cursor to "Password": Press "UP" or "DOWN" key.

2. Confirm "Password": Press OK key.

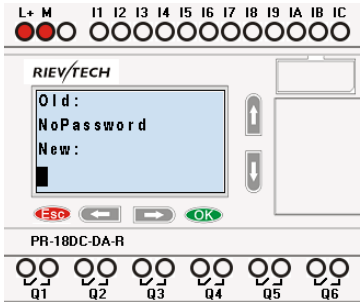
Select "Old" and input current password (in our instance is "8888"), the process is the same as the step 3 to step 10 mentioned above.

LCD displays as follows:

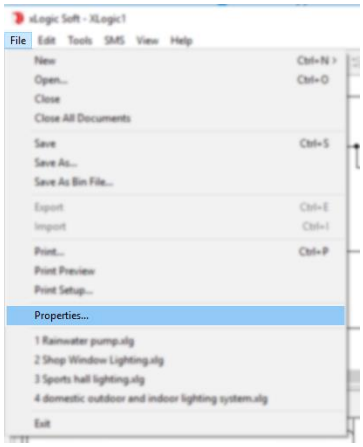


Input nothing under the "New" and press OK to clear the password.

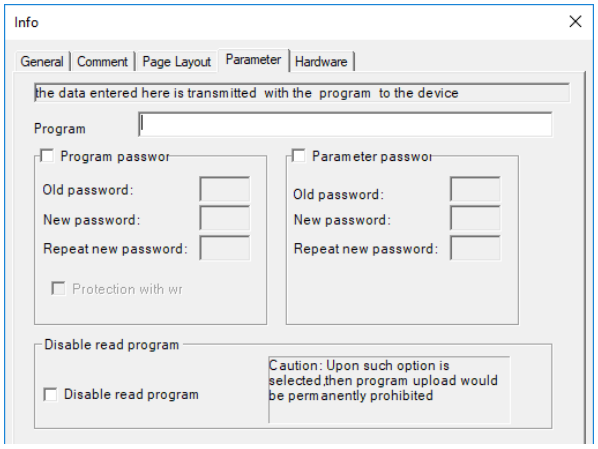
4. Confirm "blank" password: press OK key. Now you have cleared the password and can return to the main menu. If you want to set a password next time, repeat steps above in 4.3.3.



Password settings in xLogicsoft.



The menu File->Properties (Parameter)



There are 2 password settings which can be applied to the program, one is the program password and the other is the parameter password.

Program password:

If you set the program password, then the password is required when you want to upload the program to PC.

Parameter password:

If you set the parameter password, then the password is required when you want to change parameters via the LCD.

NOTICE:

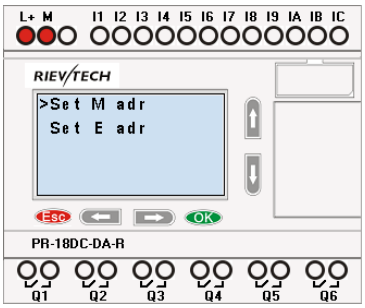
1.The password which can be set in the LCD menu is the parameter password.

2.The program password cannot be set in the LCD menu, it can only be set in xLogicsoft.

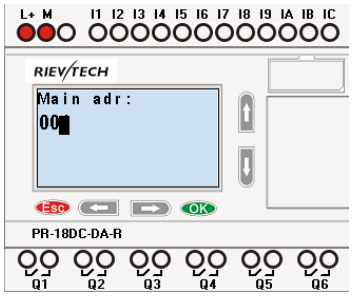
5.4.5. How to Set the Address of a CPU and Expansion Modules

Setting the CPU Address

If there is more than one CPU in a communication network, then the address of each CPU must be different. You can change the address of CPU via xLogicsoft or via the panel key if the CPU with LCD panel. The address range is from 1 to 247 for ELC and PR series CPUs.



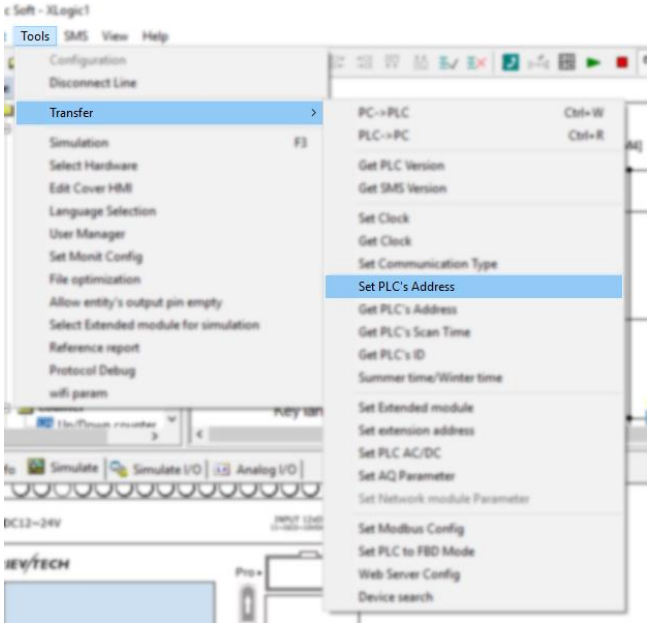
Confirm with "OK"



Change the address with arrow keys and confirm with "OK".

The CPU address can also be set by xLogicsoft if the CPU is without HMI.

You can set the CPU address by the menu Tools-> transfer-> set PLC's address in xLogicsoft.



Set Address of PR-Expansions. (PR-E-16DC-DA-R, etc.)

Plastic Cover

Using a screwdriver, remove the plastic cover and to reveal dip switches.

Set the dip switches to achieve the address you require as detailed below.

Up to 16 expansion units (includes IO, AQ, AI, PT100 modules) can be connected to the CPU.

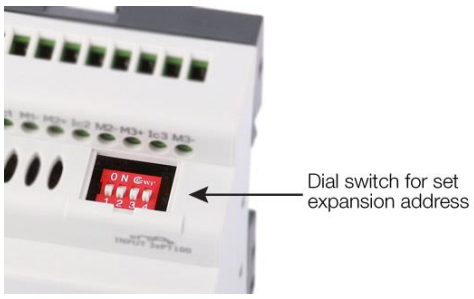
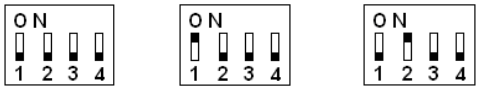


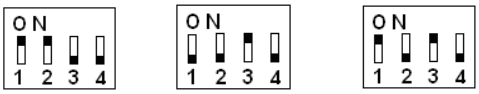
Figure 30 Expansion Address Dip Switches

The default address of PR-E expansions is 1 and the dip switches are set as below (■ means the switch position):

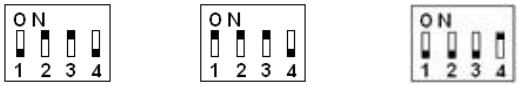
Address 1: Address 2: Address 3:



Address 4: Address 5: Address 6:

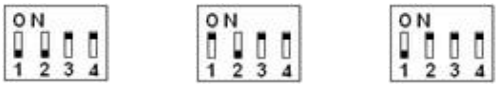


Address 7: Address 8: Address 9:

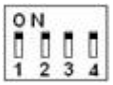


Address 10: Address 11: Address 12:

Address 13: Address 14: Address 15:



Address 16:



NOTICE:

1. The expansion unit address MUST be set before power is applied otherwise the settings will not take effect.

2. You can freely mix both AC and DC CPUs and expansion units.

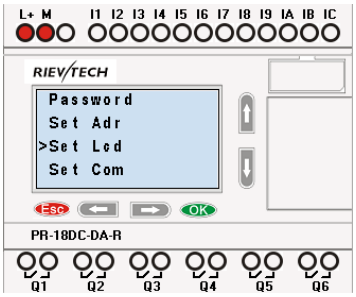
3. If the communication is established between CPU and expansion units, the LED indicator on the top of the expansion units will turn to RUN (Green). (If the LED is red, it means the CPU cannot communicate with it).

4. If more than one expansion module is connected to CPU at the same time, the address of each expansion module must be different from each other, otherwise, the system (CPU + expansions) would run abnormally. As long as each expansion unit has a unique address, they can be connected in any order whatsoever.

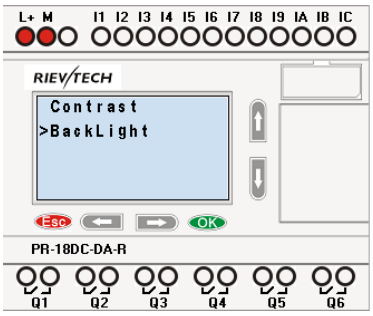
5.4.6. Setting LCD Backlight and Contrast

The backlight of CPU can be set with an "ON" time of 10 seconds or "ON" all the time. This is set as follows:

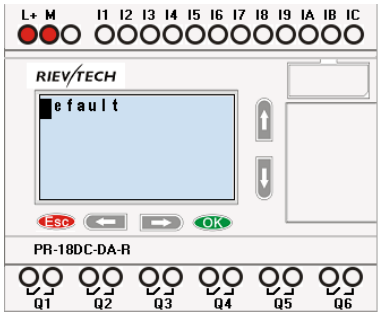
1. Select "Set..." menu and click OK.



2. Select "Set LCD" menu and click "OK"

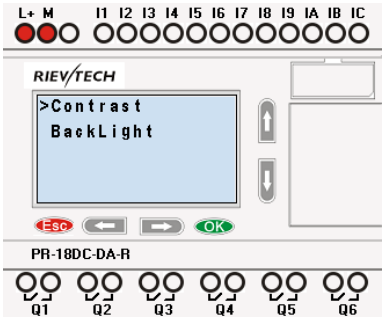


3.Select "Backlight" menu and click "OK"

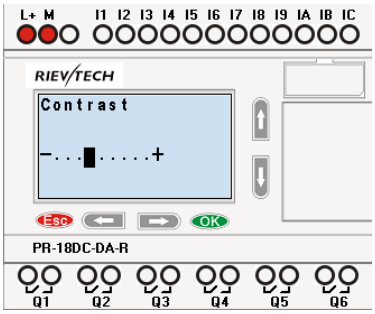


4. The default is 10 seconds, and another option is "Always On". Confirm with "OK"

Modify the contrast



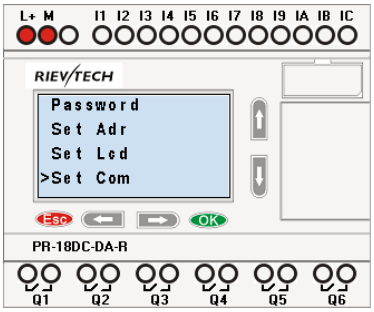
Confirm with "OK"



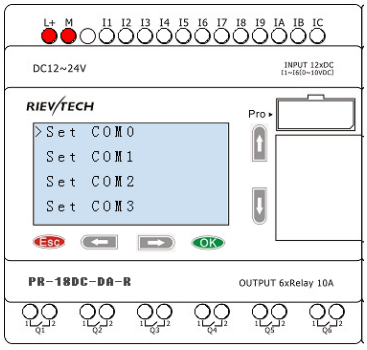
Modify the contrast with Left or Right key and confirm with "OK".

5.4.7. Set Communication Parameters

Select "Set Com" menu from the menu "Set...".



Confirm with "Ok"

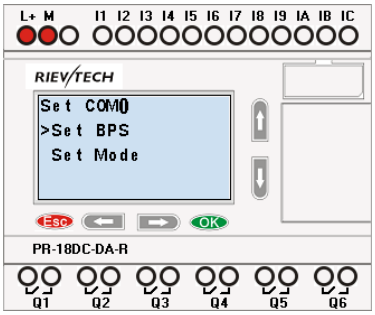


COM0: RS232 port or programming port.

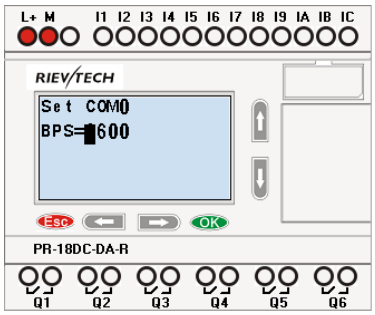
COM1: RS485 port (For the PR-RS485 expansion module when connected to PR-14, PR-18, PR-24, ELC-18, ELC-22, ELC-26 CPU)

COM2: Built-in RS485 port for PR-14 and PR-24 CPU.

COM3: Ethernet port for ELC-22-N CPU, reserved for other series CPU.



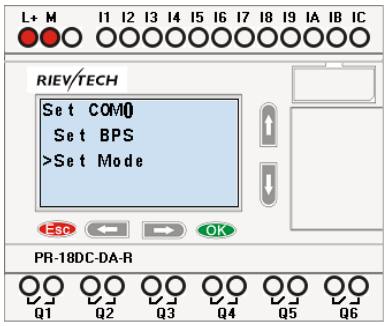
Confirm with "OK"



Change it by pressing Up and Down keys.

And confirm with "OK"

Set Modbus Protocol



Confirm with "OK"

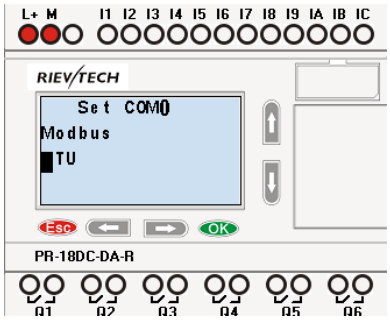


Figure 31 Set Modbus Port

Change with the Up and Down keys and confirm with "OK"

There are 4 options available: RTU, ASCII, TCP RTU, TCP ASCII.

NOTICE:
If you want to use the Modbus TCP protocol, select the "TCP RTU".

5.4.8. Setting System Date and Time

You should first select the FUNCTION PAGE. (see 15.1)

1. Move the cursor to "Clock" by pressing "UP" or "DOWN" key.

2. Confirm "Clock": Press OK key.

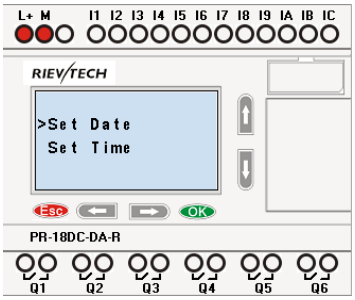
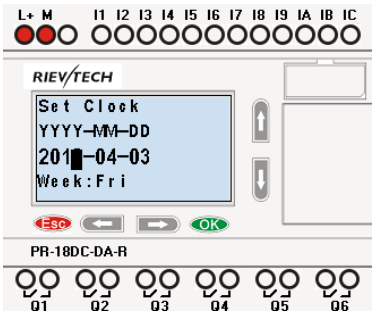
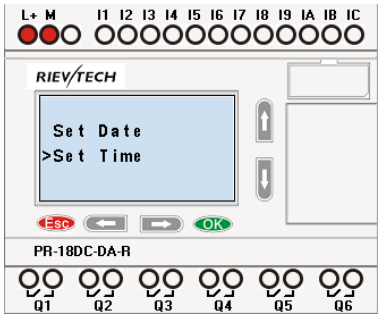


Figure 32 Set Date and Time

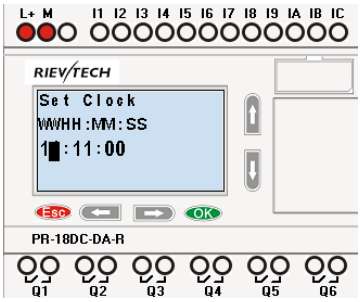
Press OK key to set and modify date.



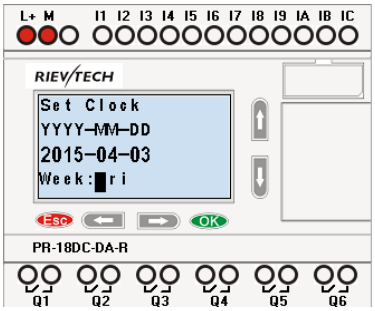
Press "UP" or "DOWN" key to change the date to what you want to set. After you have finished your setting, press OK key to return to Date/Time menu.



To set the time, move the cursor to" Set Time" menu, then press the OK key.



Here you can set the weekday (From Monday to Sunday) and the clock. The method is setting the time. After completion of your setup, press OK key:



Press ESC key and return to FUNCTION PAGE.

6. Configuring and Programming Rievtech Hardware

Programming your hardware is done using the xLogicsoft software package, which is FREE and designed to be installed on a Windows PC. The software utilises a method of programming called Function Block Diagrams (FBD). FBD is an easy to use and user-friendly programming environment. A program is created by dropping pre-programmed blocks onto a workspace area and connecting the blocks together to create your control program.

The xLogicsoft programming software allows the user to create a self-documenting, user-friendly program and to simulate your program on your PC without the need for a Rievtech PLC. Here are some of the features available in xLogicsoft software:

- Modern graphical block representation.
- A friendly graphical user interface for the creation of your circuit program by use of FBD blocks.
- Off-line simulation of your logic on your PC.
- Automatic generation of chart documentation for printing.
- Ability to add comments to your circuit program.
- Download and upload your program to/from your Rievtech PLC CPU.
- On-line debug mode allowing you to view the status of inputs and outputs, block values and timer values.
- Ability to Start and Stop your program via the PC.
- Ability to create your own blocks from the standard blocks available.

6.1. xLogicsoft Functions

The ELC and PR-Series of Rievtech CPUs can be programmed using FBD, which includes a variety of pre-configured function blocks:

- 16 Co - Constant Blocks
- 9 GF - Basic Function Blocks
- 49 SF - Special Function Blocks
- 15 AP – App Function Blocks
- MQ – MQTT Function Blocks
- Add Your Own Library Blocks

By adding these blocks to your program and linking them together, you can create relatively complex control programs. This manual will describe each available block in detail in the following sections.

Blocks are placed into your work area by selecting them from the left-hand block list in xLogicsoft and clicking anywhere in your workspace to the right of the block list.

Blocks are connected to each other by clicking on either the input or output pin of the block (a chain icon will appear) and without releasing the mouse button, drag the connector to the block-pin you wish to link to (a chain icon will appear), then let go of the mouse button. We are continuingly adding features to our software and should you need a new special function block which currently does not exist in xLogicsoft, then let us know and we will look at adding it. Contact: technical@rievtech.com.

6.1.1. General Input and Output Functions

Digital Input

Digital input blocks represent the physical digital input terminals of your Rievtech CPU and expansion units if fitted. The following table list the maximum number of digital inputs that a CPU can support.

Table 11 Maximum Digital Inputs per CPU

CPU	MAXIMUM DIGITAL INPUTS
PR-6	4
PR-12	8
PR-14	74
PR-18	76
ELC-22	78
PR-24	78

To configure a digital input block which has been placed in your workspace, double-click on it to open the blocks configuration options.

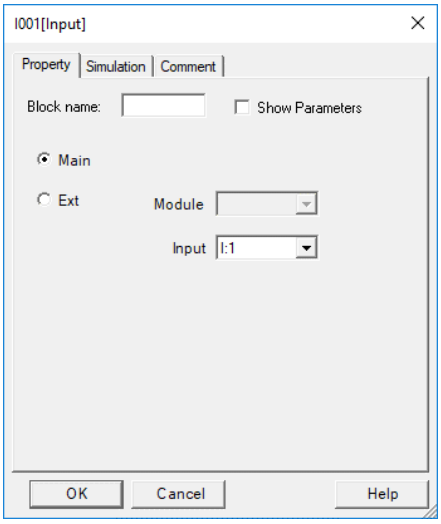


Figure 40 - Digital Input Block Configuration

If the input is on the main CPU then you can select it using the Input drop-down box. Only available inputs will be shown.

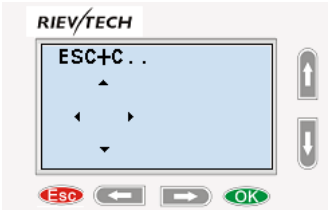
If the digital input is in an expansion module, then select the radio button for Ext. Module and use the drop-down box to select which expansion block number, followed by the digital input number.

NOTICE:

I12 means digital input 2 on expansion address 1, I81 means digital input 1 on expansion address 8.

Cursor keys

Up to four cursor keys are available to you. Cursor keys are programmed for the circuit program in the same way as other inputs. Cursor keys can save switches and inputs and allow operator control of the circuit program.



Switch the screen to the current page (above shows) by pressing the Left or Right key, and press ESC key and arrow keys at the same time, then the corresponding cursor keys will turn on and give off a high trigger!

Digital Output

Output blocks represent the output terminals of xLogic. You can use up to 2 outputs (ELC-6), 4 outputs (PR-12),70 outputs (PR-18), 72 outputs (PR-22), 74 outputs (PR-26). In your block configuration, you can assign an output block to a new terminal, provided this terminal is not already used in your circuit program.

The output always carries the signal of the previous program cycle. This value does not change within the current program cycle.

NOTICE:

Q11 means the Q1 output of the expansion with address 1.

Q85 means the Q5 output of the expansion with address 8.

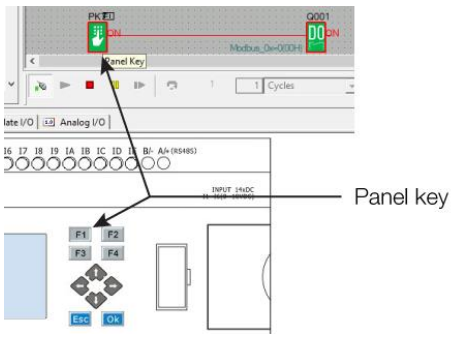
Permanent Logical Levels HI and LO

Set the block input to logical **hi** (hi = high) to set it permanently to logical '1' or 'H' state.

Set the block input to logical lo (lo = low) to set it permanently to logical '0' or 'L' state.

Panel Key

It is the symbol for the Function key on the CPU panel (F1—F4) where fitted. If one of the function keys is pressed down, the status of the corresponding symbol in the program shall turn from 0 to 1. This will give off one high trigger.



NOTICE:

1. Only one function key can be pressed down at any one time. If you press down two or more at the same time, xLogic will not process the keystrokes.

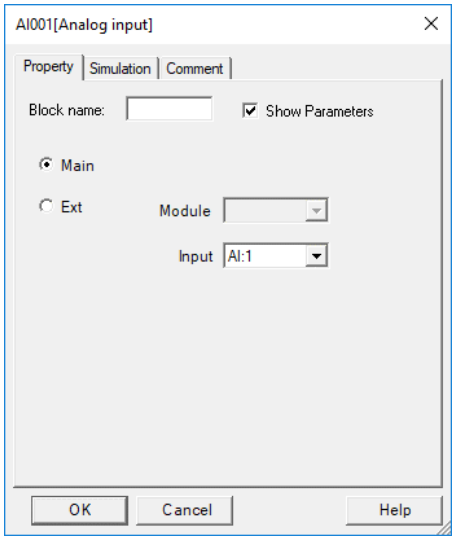
2. If the arrow keys (UP, DOWN, LEFT and RIGHT), Esc and OK had been applied to the program, then they would be invalid for menu operation (e.g. manual programming, parameters modification and view alarming message etc).

Shift Register Bits

xLogic provides the shift register bits S1 to S8, which are assigned the read-only attribute in the circuit program. The content of shift register bits can only be modified by means of the Shift register special function

Analogue Inputs

You can use up to 36 analogue inputs. In your block configuration, you can assign a new input terminal to an input block, provided this terminal is not already used in the circuit program.



For help on analogue block parameter, refer to Information on analogue value processing.

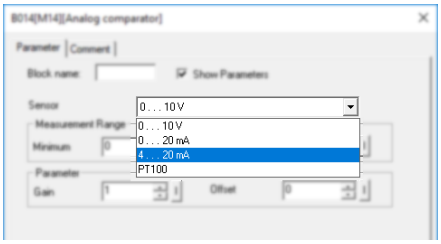
NOTICE:

AI11 means the AI1 input of the expansion with address 1.

AI82 means the AI2 input of the expansion with address 8.

Based on different analogue input signals, you need select a different sensor type in the analogue blocks, you need to use an analogue amplifier with the AI input,

and select the corresponding sensor type for DC0.10V, 0/4.20mA or PT100 input.



F (Digital Flag)

Flags are used when xLogic works in a communication system. F is a digital flag which is used to save /transfer signal 1 or 0 (data format is Bit) and AF is an analogue flag which is used to save /transfer analogue values (data format is Signed short) between the master and slave devices. Up to 32 (ELC-6), 128 (PR-12/ELC-22/ELC-26) can be used when programming. In your block configuration, you can assign a new number to the flag, provided this flag number does not already exist in your circuit program.

The output always carries the signal of the previous program cycle. This value does not change if the communication has failed.

Start-Up Flag: F8

The F8 flag is set in the first cycle of the user program and can thus be used in your circuit program as a start-up flag. It is reset after the first program execution cycle.

In the subsequent cycles, the F8 flag reacts in the same way as the F1 to F64 flags.

Backlight Flag: F64

The F64 flag controls the backlight of the xLogic display or the ELC-43T backlight. You can use the outputs of timers, message texts, or other function blocks to activate the backlight flags. To enable multiple conditions to control the backlight of the devices, you can use multiple function blocks in parallel or in sequence. Buzzer of ELC-43TS control flag: F63

NOTICE:

1. The address of "F" can be found in the Modbus communication protocol file. The F block figure must have the input pin in the xLogic showing.

2. Keep the input pin of F NULL (do not connect with other blocks) if you want to use the write property.

AF (Analogue Flag)

Flags are used when xLogic works in a communication system. AF is an analogue flag which is used to save /transfer analogue values (data format is Signed short) between the master and slave devices. Up to 32 (PR-6) and 256 (PR-12/PR-18/PR-24) for xLogic CPU can be used when programming. In your block configuration, you can assign a new number to the analogue flag, provided this flag number does not already exist in your circuit program.

The output always carries the signal of the previous program cycle. This value does not change if the communication were to fail.

NOTICE:

1. The address of "AF" can be found in the Modbus communication protocol file.

2. Keep the input pin of AF NULL (do not connect with other blocks) if you want to use the Write property.

You can set a start value for the AF1—AF64 (PR-12/PR-18/PR-24), and the value does not get lost if the power was a failure, so you can use such AF in your program as a number input from the touchscreen or the panel key of the CPU.

NOTICE:

1. AF65--AF256 does not support a Start Value function, and the values of AF65 to AF256 will be lost if the power fails.

2. If you enable the "start value option" of AF, then you are not allowed to connect any other function to the input leg of the AF.

3. If you connect the input leg of the AF block to other function blocks, the "start value" will not be available anymore.

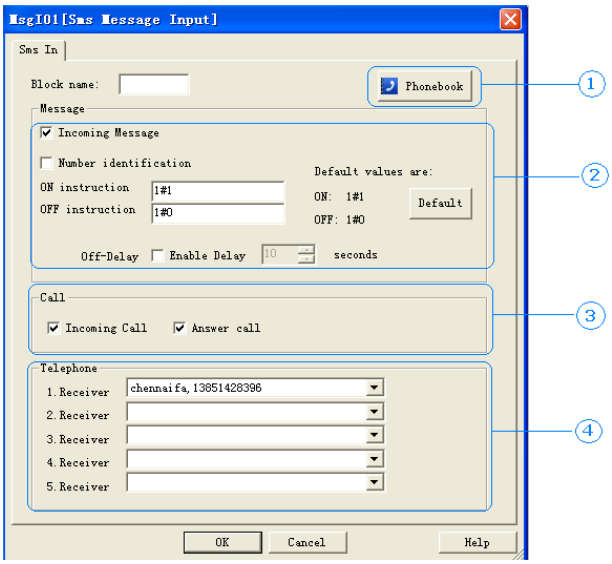
SMS Message Input

Description of Function

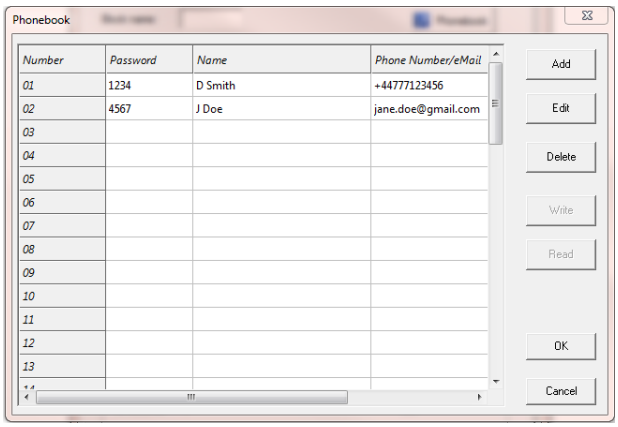
Up to 32 SMS Message Input blocks are available for use in your program. They are used in a similar way to other digital input blocks. But they enable the switching of their state via a SMS message sent to the CPU from a phone.

Users can send a pre-set message to the CPU to change the state of the inputs, from "0/OFF" to "1/ON" or vice versa. The ability to define other commands is also provided such as "Switch Pump On", "Close Drain Valve", etc.

Configuration of Function



1. Phonebook – Clicking this button will display the phonebook and allow editing of it.



There are two control options for the SMS Message Input, Incoming Message and Incoming Call, which can be used individually or both together.

2. Message Configuration
    a. Incoming Message – If this option is selected, then the input state can be switched using a SMS message, which MUST match the On and Off instructions configured in this section. Default values are "1#1" for ON and "1#0" for OFF.
    b. Number Identification – If this box is ticked, then only SMS received from those configure in the Telephone section will be acted upon.
    c. On Instruction – Define the text that will switch this input TRUE. Default is 1#1.
    d. OFF Instruction – Define the text that will switch this input FALSE. Default is 1#0.
    e. Off-Delay – If this box is ticked and a time in seconds configured, then once a SMS "On" message is received and accepted, the input will switch to TRUE and reset to FALSE once the time has expired.
3. Incoming Call Configuration
    a. By ticking the box for this feature, then the input will become TRUE when a call is received. Once the call is stopped or Off-Delay timer expires, the input will revert to its FALSE state.
    b. Answer Call – If this is selected, then the CPU will answer the call made to it and sets the input to TRUE for 10-seconds or until you hang up if sooner. This is a good way to check the CPU recognises your number. If this option is not selected, because the CPU would not answer your call, then no charge should result on your phone bill.
4. Telephone Receiver Configuration

In this section you can choose up to a maximum of five contacts from the phonebook for which this block will accept instructions for changing state.

SMS Message Output

Description of Function

You are allowed up to a maximum of 64 SMS Message Output blocks. Each one can send a SMS message or Call (without sound) to all contacts configured in the block. This block is only triggered when there is a LOW to HIGH trigger.

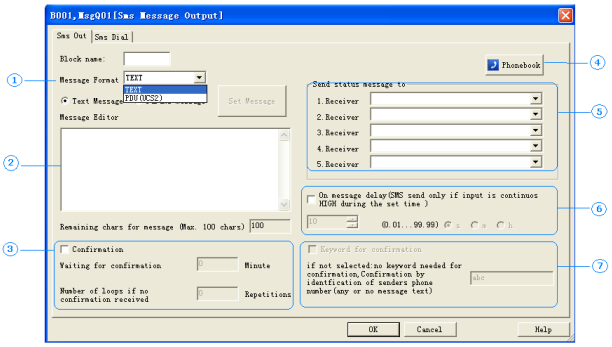
NOTICE:

1. At least ONE number MUST be configured.
2. Only use standard ASCII characters in the SMS.
3. Any 'special' characters will cause an error.

The output pin of this block will only become high once ALL messages have been successfully sent.

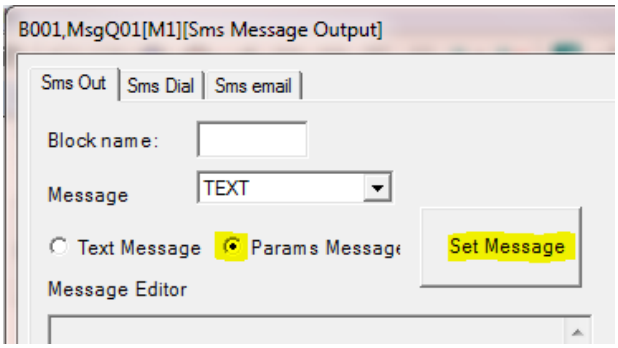
Connection	Description
Input **Trg**	Trigger to initiate SMS out message
Input **R**	Reset output pin to 0/False
Output Q	Set TRUE after ALL SMS successfully sent

Configuration of Function

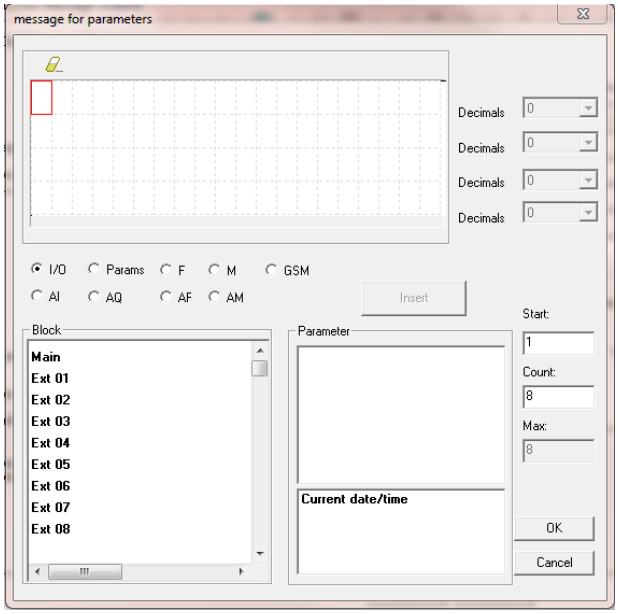


1. Message Format
   a. Text – Support for ASCII characters
   b. PDU – Support for foreign language characters
2. Message Editor
   a. Text – Up to 100 ASCII characters can be used
   b. PDU – Up to 70 characters can be used

If you wish to send parameter information such as IO states, then select the Params Message radio button and click the Set Message button.

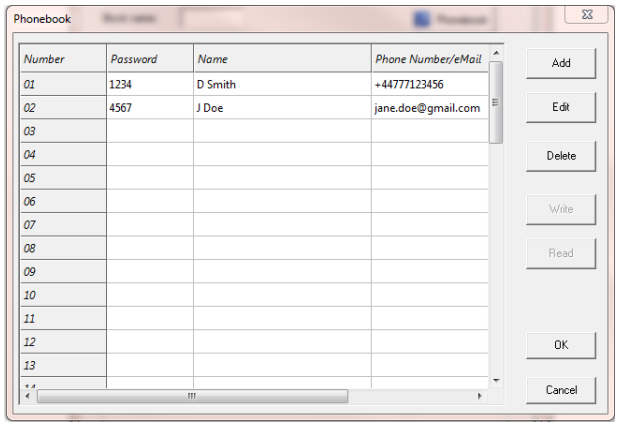


The message for parameters window should appear:



See the Message Text configuration section for details on configuring the settings here.

3. Confirmation – If ticked, the CPU will expect a SMS confirmation. If, unticked, each number configured in the list will be sent the SMS.
    a. Confirmation Time in Minutes – Maximum amount of time the CPU will wait for a confirmation message. If timer expires, then the message will be sent to the next number configured in the Receiver list.
    b. No. of Loops – The number of times the CPU must try to send SMS to the numbers configured.
4. Phonebook – Clicking this button will display the phonebook and allow editing of it.



5. Send Message To – Configure up to a maximum of 5 numbers who are to receive the message.
6. On Message Delay – When the trigger state goes TRUE, then the time delay configured here, will cause the CPU to wait until the timer has expired before sending the SMS. Should the input become FALSE during this delay, the timer will be reset. Time delay is configured between 0.1 seconds and 99.9 hours.

NOTICE:

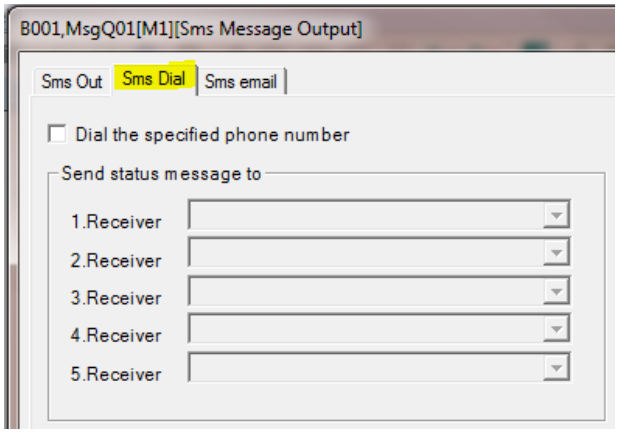
It is advisable to Enable this feature so as to prevent false triggers such as liquid in a tank, moving over a sensor and then uncovering it again, whilst the tank may be moving.

7. Keyword for Confirmation – If this option is NOT selected, then it is sufficient to send SMS confirmation to the CPU with no text. If selected, then the confirmation sent MUST be the text entered in this section.

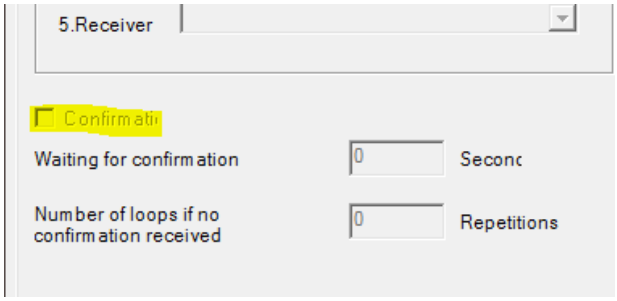
If preferred, this block may be used to send an email or call a contact instead of sending an SMS.

SMS Dial – Call Contact(s)

When the EN pin has been activated, the CPU will dial the phone number(s) stored in the pre-set receivers:



With confirmation function activated:



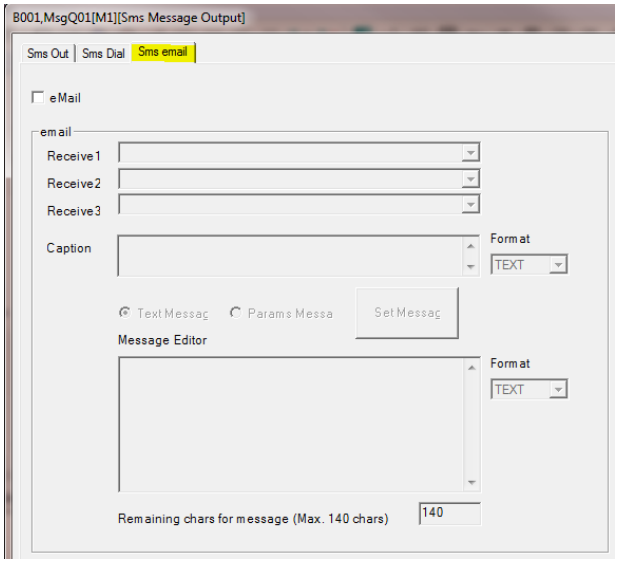
If the receiver does not answer the call from the CPU within a certain pre-set time, it will dial the next number.

The 'Waiting for confirmation' setting determines the length of time before the next number is attempted. This process is repeated until either a call is answered or the end of the list is reached.

If you require the CPU to try a number of times, then by configuring the Repetitions field with the number of retires required will make the CPU repeat the loop for the set number.

SMS Email – Email Contact(s)

When the EN pin has been activated, the CPU will email the address listed in the pre-set receivers:



See the Message Text configuration section for details on configuring the settings here.

NOTICE:

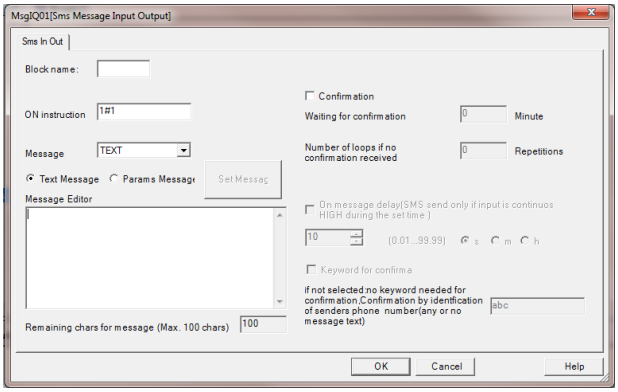
The email settings MUST be configured before the CPU is able to send emails. See 'Configuration of Email Settings'.

SMS Message Input Output

Description of Function

This block can be used in a program where the sending of a SMS message requires the receiver to provide an acknowledgement SMS response.

Up to 12 of these blocks can be used in a program.



This block can be used when you require a SMS message to be sent on an event AND you require an acknowledgement from the receiver.

See SMS Message Input and SMS Message Output for instructions on configuring this block.

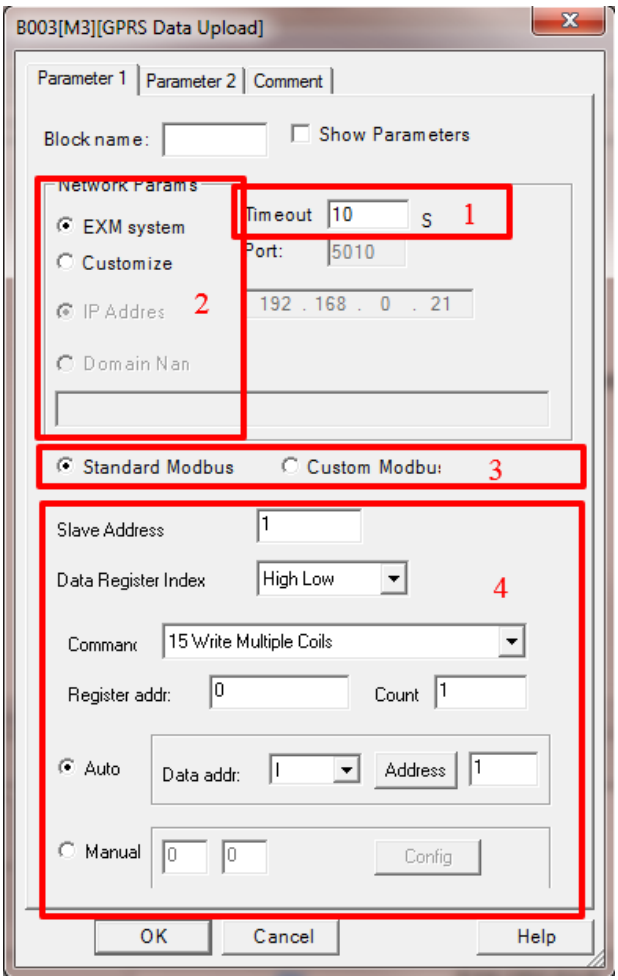
GPRS Data Upload

Description of Function

This block cab be used to enable a GPRS connection and then transfer some information to the network server.

Connection Pin	Description
Input Trg	GPRS connection enable when set to High/True.
Input R	Reset the GPRS connection and output pin even if Trg input is High.
Output Q	Q is switched On when a GPRS connection is established with the server software and the correct Modbus/TCP response is received. If the GPRS is disconnected, the output will be Off.

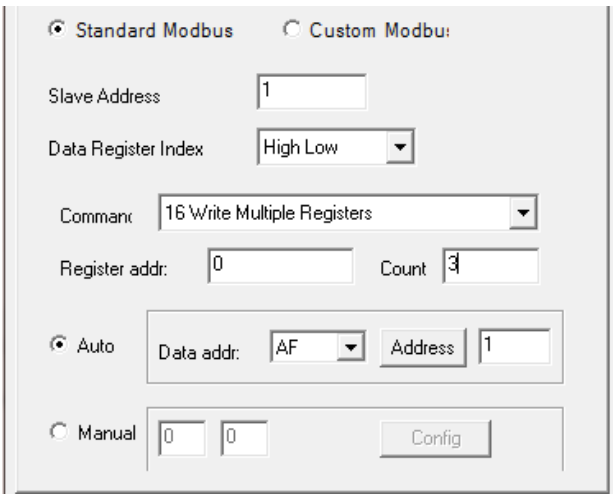
Configuration of Function



1) Timeout – If GPRS server does not respond within the set timeout period, the CPU will attempt the data transmission up to a maximum of three times after which the GPRS connection will be automatically disconnected.
2) Network Params – There are two options, see GPRS Connect for further information.
3) Modbus Type – Standard or Custom
   a. Standard Modbus

If your program has three flags with the following values:

- AF1 = 100
- AF2 = 1000
- AF3 = 10000



The CPU will upload the AF1 to AF3 values to the server (Slave ID 1) as follows:

00 01 00 00 00 0D 01 10 00 00 00 03 06 00 64 03 E8 27 10

The request and response are prefixed by six bytes as follows:
byte 0: transaction identifier - copied by server
byte 1: transaction identifier - copied by server
byte 2: protocol identifier = 0
byte 3: protocol identifier = 0
byte 4: length field (upper byte) = 0 (since all messages are smaller than 256)
byte 5: length field (lower byte) = number of bytes following
byte 6: unit identifier (previously ‘slave address’)
byte 7: MODBUS function code
byte 8: Register of the slave start address
byte 9: Register of the slave start address
byte 10: number of registers
byte 11: number of registers
byte 12: data length field (lower byte) = number of bytes following

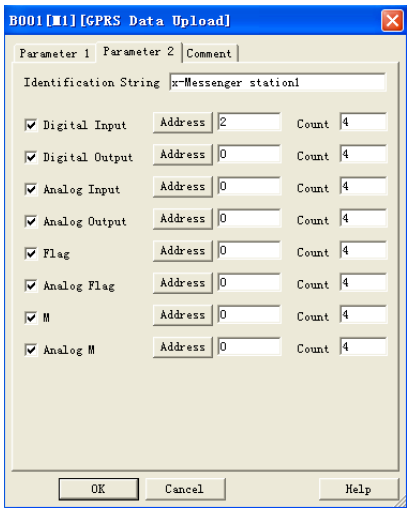
So, AF1 = 00 64 (DECIMAL 100)
AF2 =03 E8 (DECIMAL 1000)
AF3 = 27 10 (DECIMAL 10000)

The server end(slave1) responds
00 01 00 00 00 06 01 10 00 00 00 03
So, if the "Trg" input remains high, then the output will be high also after the CPU gets the above, correct response.

NOTICE:
The Modbus Function Code 16 is used to upload analogue values (inputs, outputs, flags and REG) and Modbus Function Code 15 is used to upload digital values (inputs, outputs and flags).

b. Custom Modbus

Using this setting enables you to upload a mix of data types as configured under the Parameter 2 tab of this block.



The above example would result in the following data being uploaded to the server:

00 09 00 00 00 4c 01 8b 49 14 78 2d 4d 65 73 73 65 6e 67 65 72 20 73 74 61 74 69 6f 6e 31 01 01 01 02 01 0d 0b 08 00 80 00 88 00 00 00 00 0c 08 00 00 00 00 00 80 00 88 16 01 02 17 08 01 c2 00 00 02 26 00 00 07 01 04 0d 08 00 00 00 00 00 00 01 c2

Explanation of the data message:

00 09 00 00 00 4c – Modbus/TCP prefixed data of six bytes

01 – CPU address (Default is 1)

8b – Upload code (Fixed if GPRS Data Upload used)

49 – Length field (Number of bytes following)

14 – Length field (Number of bytes of identification string)

78 2d 4d 65 73 73 65 6e 67 65 72 20 73 74 61 74 69 6f 6e 31 – CPU Station 1 (Identification string)

01 – Type Code (01=Digital Inputs)

01 – Length field (number of bytes of digital inputs)

01 – Status of digital inputs (I3=1, I4=0, I5=0, I6=0)

01 – Type Code (02=Digital Outputs)

01 – Length field (number of bytes of digital inputs)

0d – Status of digital outputs (Q1=1, Q2=0, Q3=1, Q4=1)

0c – Type code (0c means the Analogue output)

08 – Length field (number of bytes of the analogue outputs)

00 00 00 00 00 80 00 88 – Analogue output values (AQ1= 00 00, AQ2 = 00 00, AQ11= 00 80, AQ12= 00 88)

16 – Type code (16 means the digital flag)

01 – Length field (number of bytes of the digital flag)

02 – Status of the digital flag (F1=0, F2=1, F3=0, F4=0)

17 – Type code (17 means the Analogue flag)

08 – Length field (number of bytes of the analogue flag)

01 c2 00 00 02 26 00 00 – Analogue flag values (AF1= 01 c2, AF2 = 00 00, AF3= 02 26, AF4= 00 00)

07 – Type code (07 means the M status)

01 – Length field (number of bytes of M(Function block status(1/0))

04 – Status of the M Flags (M1=0, M2=10, M3=1, M4=0)

0d – Type code (0d means the AM value)

08 – Length field = number of bytes of AM

00 00 00 00 00 00 01 c2 – AM values (AM1=00 00, AM2=00 00, AM3=00 00, AM4=01 c2)

The expected server response would be:

00 00 00 00 00 02 01 8b

So, if the Trg input is kept High, then the output, Q, will be kept High is the above correct response is received.

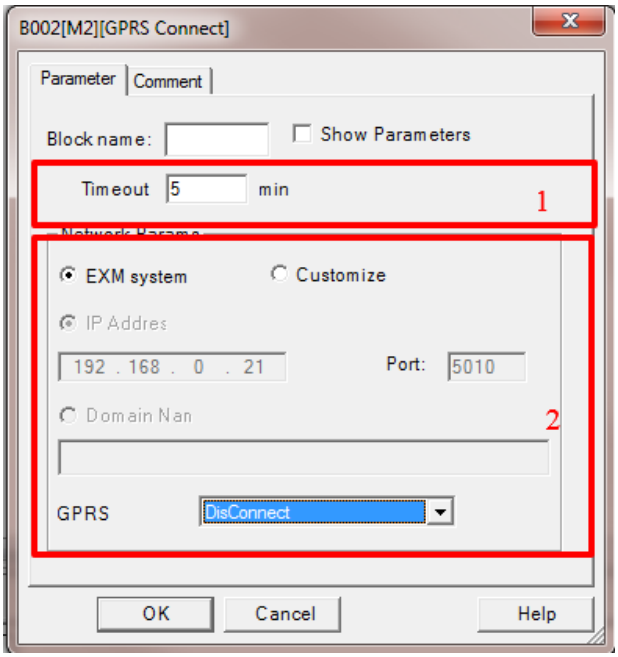
GPRS Connect

Description of Function

The block can be used to enable/disable the GPRS connection of the CPU.

Pin Connection	Description
Input **Trg**	When the trigger input is enabled, the GPRS connection is enabled.
Output **Q**	The output Q is switched On once a GPRS connection is established. It will be set to Off if the connection is disconnected.

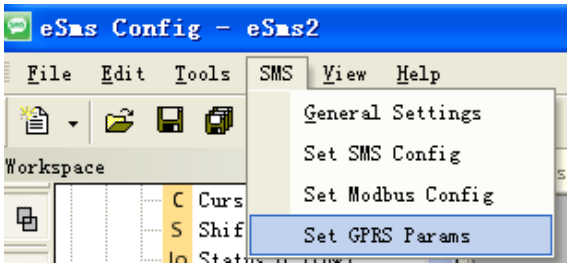
Configuration of Function

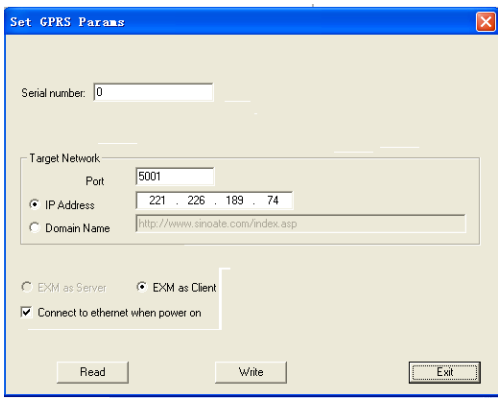


1. Timeout Setting – This is the length of time for the timeout if there is no data transmission during this period. The GPRS will disconnect automatically.
2. Network Params – There are two options to choose from.

a. EXM System

If this setting is selected, the IP and port number of the server shall be the same as configure in the menu option... "SMS...Set GPRS Params".





b. Customize

This option allows you to select another server.

NOTICE:

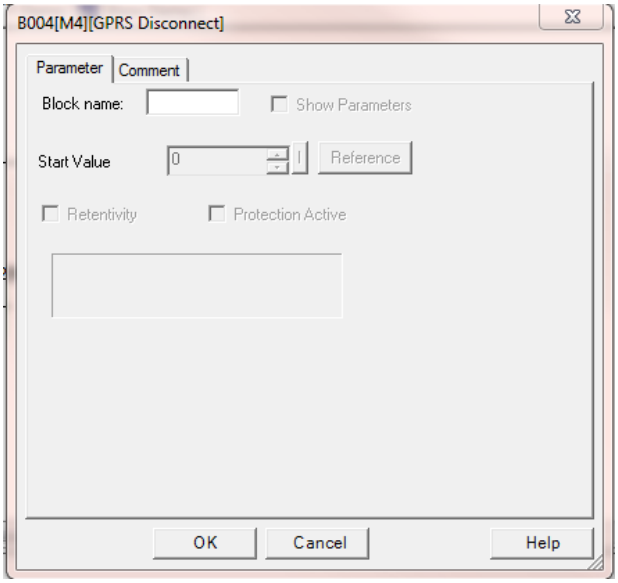
1) The configuration of this block has a HIGHER priority than the settings set using the menu option. If the block is activated, the GRPS connection configured via the menu will be disconnected automatically and these settings used instead.
2) The data transmission between the CPU and the server is based on standard Modbus/TCP protocol.

GPRS Disconnect

Description of Function

This function block can be used to force the GPRS connection to be disconnected.

Configuration of Function



When the input pin of this block is set High, the output pin will be set High too.

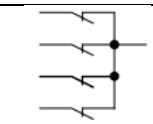
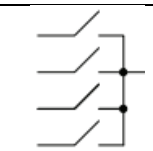
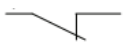
6.1.2. Basic Functions List – GF

Basic functions represent simple logical elements of Boolean algebra.

You can invert the inputs of individual basic functions, i.e. The circuit program inverts a logical "1" at a relevant input to a logical "0"; if "0" is set at the input, the program sets a logical "1" and vice versa.

The GF list contains the basic function blocks you can use for your circuit program. The following basic functions are available:

Table 12 Basic functions

View in the circuit diagram	View in xLogicsoft	Name of the basic function
		AND
		AND with edge evaluation
		NAND (Not AND)
		NAND with edge evaluation
		OR
		NOR (Not OR)
		XOR (exclusive OR)
		NOT (negation, inverter)

BOOLEAN FUNCTION		BOOLEAN FUNCTION

AND

Description of function

The output of an AND function is only 1 if all inputs are 1, i.e. when they are closed.

A block input that is not used (x) is assigned: x = 1.

Table 13 AND Block Logic

Input1	Input2	Input 3	Input 4	Output
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

AND with Edge Evaluation

Description of function

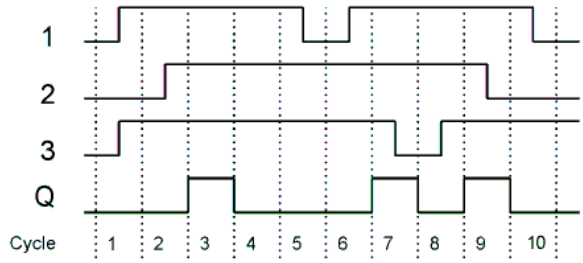
The output of an AND with edge evaluation is only 1 if all inputs are 1 and at least one input was 0 during the last cycle.

The output is set to 1 for the duration of one cycle and must be reset to 0 for the duration of the next cycle before it can be set to 1 again.

A block input that is not used (x) is assigned: x = 1.

0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

The timing diagram of an AND with edge evaluation



NAND

Description of function

The output of a NAND function is only 0 if all inputs are 1, i.e. when they are closed.

A block input that is not used (x) is assigned: x = 1.

Table 14 NAND Block Function

Input1	Input2	Input 3	Input 4	Output
0	0	0	0	1
0	0	0	1	1

NAND with Edge Evaluation

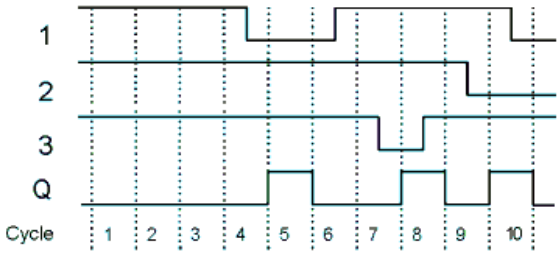
Description of function

The output of a NAND with edge evaluation is only 1 at least one input is 0 and **all** inputs were 1 during the last cycle.

The output is set to 1 for the duration of one cycle and must be reset to 0 at least for the duration of the next cycle before it can be set to 1 again.

A block input that is not used (x) is assigned: x = 1.

The timing diagram of a NAND with edge evaluation



OR

Description of function

The output of an OR is 1 if at least one input is 1 (closed).

A block input that is not used (x) is assigned: x = 0.

Table 15 OR Block Logic

Input1	Input2	Input 3	Input 4	Output
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

NOR

Description of function

The output of a NOR (NOT OR) is only 1 if all inputs are 0, i.e. when they are open. When one of the inputs is switched on (logical 1 state), the output is switched off.

A block input that is not used (x) is assigned: x = 0.

Table 16 NOR Block Logic

Input1	Input2	Input 3	Input 4	Output
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

XOR

Description of function

The XOR (exclusive OR) output is 1 if the signal status of the inputs is different.

A block input that is not used (x) is assigned: x = 0.

Table 17 XOR Function Logic

Input1	Input2	Output
0	0	0
0	1	1
1	0	1
1	1	0

NOT

Description of function

The output is 1 if the input is 0. The NOT block inverts the input status.

The advantage of the NOT, for example, xLogic no longer requires break contacts. You simply use a make contact and convert it into a break contact with the help of the NOT function.

Table 17 NOT Function Logic

Input1	Output
0	1
0	0

Boolean Function

Description of function

The **BOOLEAN** function gives the value of the output according to the combination of inputs.

The function has four inputs, and therefore 16 combinations. These combinations can be found in a truth table; for each of these, the output value can be adjusted. The number of configurable combinations depends on the number of inputs connected to the function.

Non-connected inputs are set to 0.

The following diagram shows an example of part of the Boolean function truth table:

Table 18 Boolean Function Logic (Example of Output states)

Index	Input1	Input2	Input 3	Input 4	Output
1	0	0	0	0	1
2	0	0	0	1	1
3	0	0	1	0	1
4	0	0	1	1	0
5	0	1	0	0	1
6	0	1	0	1	1
7	0	1	1	0	1
8	0	1	1	1	1
9	1	0	0	0	1
10	1	0	0	1	0
11	1	0	1	0	0
12	1	0	1	1	0
13	1	1	0	0	0
14	1	1	0	1	1
15	1	1	1	0	0
16	1	1	1	1	0

Parameters

Having connected at least one input, you can configure the value of the output in the truth table, in the Parameters window.

The output values can be 0 for the Inactive state, and 1 for the Active state (Double click to change the 0 or 1).

By selecting the Output ON if the result is TRUE, the output takes the value configured in the truth table.

By selecting the Output OFF if the result is TRUE, the output takes the inverse value of the value configured in the truth table.

6.1.3. Basics of Special Functions - SF

Because of their different input designation, you can see right away that there is a difference between the special functions (SF) and basic functions. SF's contain timer functions, retentive functions and various parameter assignment options, which allow you to adapt the circuit program to suit your own requirements.

This section provides you with a brief overview of input designations and with some particular background information on SF's. The SF's, in particular, are described in Chapter 5.5

Designation of the Inputs to Blocks

Logical Inputs

Here, you will find the description of the connectors you can use to create a logical link to other blocks or to the inputs of the xLogic unit.

Table 19 Logical Input types

Name	Description	Image
S (Set)	A signal at input S sets the output to logical "1".	▶
R (Reset)	The reset input R takes priority over all other inputs and resets the outputs.	○
Trg (Trigger)	This input is used to trigger the start of a function.	●
Cnt (Count):	This input is used for counting pulses.	●
Fre (Frequency)	Frequency signals to be evaluated are applied to this input.	▶
Dir (Direction)	This input determines the direction of count.	-
En (Enable) or Validation	This input enables a block function. When this input is "0", other signals to the block will be ignored.	▶
Inv (Invert)	A signal at this input inverts the output signal of the block.	-
Ral (Reset all):	All internal values are reset.	●
Value	Integer value	▶

Parameter inputs

For some of the inputs, you do not apply any signals. You configure the relevant block values instead. Examples:

Par (Parameter):

This input will not be connected. Here, you set the relevant block parameters (times, on/off thresholds etc.).

No (Cam):

This input will not be connected. Here, you configure the time patterns.

P (Priority):

This is an open input. Here, you define priorities and specify whether a message is to be acknowledged during RUN time.

Time Response

Parameter T

In some of the SFs it is possible to configure a time value T. When you pre-set this time, your input values are based on the time base set:

Time base	–:–
s (seconds)	Seconds : 1/100 seconds
m (minutes)	Minutes : seconds
h (hours)	Hours : minutes

Accuracy of T

Because of slight tolerances in the characteristics of electronic components, the set time T may deviate. The

xLogic has a maximum tolerance of ± 0.02 %. When 0.02 % of the time T is smaller than 0.02 seconds, the maximum deviation is 0.02 seconds.

Example: The maximum tolerance per hour (3600 seconds) is ±0.02%, which is proportional to ± 0.72 seconds. The maximum tolerance per minute (60 seconds) is ± 0.02 seconds.

The accuracy of the timer (weekly/yearly timer)

The maximum timing inaccuracy is ± 5 s/day.

Backup of the Real-time Clock

Because the internal real-time clock of a CPU is backed up, it continues operation after a power failure. The ambient temperature influences the backup time. At an ambient temperature of 25°C, the typical backup time of a CPU is 100 hours.

Retentivity

The switching states and counter values of SFs can be set to be retentive. This means that current data is retained after a power failure and that the block resumes operation at the point at which power was lost. The timer is not reset but resumes operation until the time-to-go has expired, for example, to enable this response, however, the relevant functions must be set retentive.

SFs hours counter, weekly timer, yearly timer and PI controller are always retentive.

Parameter Protection

In the parameter protection settings, you can determine whether or not the parameters can be displayed and edited in xLogic parameter assignment mode. Two options are available:

- The parameter attribute permits read/write access in parameter assignment mode(default).

- The parameter settings are read–/write–protected in parameter assignment mode and can be edited only via xLogicsoft.

Calculating the Gain and Offset of Analogue Values

A sensor is connected to the analogue input and converts a process variable into an electrical signal. This value of signal lies within the typical range of this sensor. xLogic always converts the electrical signals at the analogue input into digital values from 0 to 1000. A voltage of 0 to 10 V (or current signal 0/4…20mA) at input AI is transformed internally into a range of values from 0 to 1000. An input voltage exceeding 10 V is shown as internal value 1000.

Because you cannot always process the range of values from 0 to 1000 as predetermined by xLogic, you can multiply the digital values by a gain factor and then shift the zero of the range of values (offset). This allows you to output an analogue value to the xLogic display, which is proportional to the actual process variable.

Table 20 Gain and Offset

Parameter	Minimum	Maximum
Input voltage (in V)	0	≤10
Input current (in mA)	0/4	≤20
Internal value	0	1000
Gain	–10.00	+10.00
Offset	–10000	+10000

Mathematical rule

Equation 1 Offset and Gain

*Actual value Ax = (internal value at input Ax * gain) + offset*

Gain and offset calculation

The gain and offset are calculated based on the relevant high

and low values of the function.

Example 1:

The available thermocouples have the following technical

data:

–30 to +70 ℃, 0 to 10 V DC (i.e. 0 to 1000 in xLogic).

Actual value = (internal value ·gain) + offset, thus

–30 = (0 ·A) + B, i.e. offset B = –30

+70 = (1000 ·A) –30, i.e. gain A = 0.1

Example 2:

A pressure sensor converts a pressure of 1000 mbar into a voltage of 0 V, and a pressure of 5000 mbar into a voltage of 10 V.

Actual value = (internal value. gain) + offset, thus

1000 = (0·A) + B, i.e. offset B = 1000

5000 = (1000 ·A) +1000, i.e. gain A = 4

Example of analogue values

Process variable	Voltage (V)	Internal value	Gain	Offset	Value shown (Ax)
–30⁰ C	0	0	0.1	–30	–30
0⁰ C	3	300	0.1	–30	0
+70⁰ C	10	1000	0.1	–30	70
1000 mbar	0	0	4	1000	1000
3700 mbar	6.75	675	4	1000	3700
5000 mbar	10	1000	4	1000	5000
	0	0	0.01	0	0
	5	500	0.01	0	5
	10	1000	0.01	0	10
	0	0	1	0	0
	5	500	1	0	500
	10	1000	1	0	1000
	0	0	10	0	0
	5	500	10	0	5000
	10	1000	10	0	10000
	0	0	0.01	5	5
	5	500	0.01	5	10
	10	1000	0.01	5	15
	0	0	1	500	500
	5	500	1	500	1000
	10	1000	1	500	1500
	0	0	1	–200	–200
	5	500	1	–200	300
	10	1000	1	–200	800
	0	0	10	–10000	–10000
	10	1000	10	–10000	0
	0.02	2	0.01	0	0
	0.02	2	0.1	0	0
	0.02	2	1	0	2
	0.02	2	10	0	20

6.1.4. Special functions list – SF

When you create your circuit program in xLogicsoft, you find the special function blocks in the SF list.

You can invert the inputs of SF's individually, i.e. the circuit program converts a logical "1" at the input into a logical "0"; a logical "0" converts to logical "1". The table also specifies whether the relevant function can be set retentive (Rem). The following SF's are available:

Table 21 SF - Special Functions List

View in xLogic	Name of the special function	Rem
Timer		
	On-delay	REM
	Off-delay	REM
	On-/Off-delay	REM
	Retentive on-delay	REM
	Wiping relay (pulse out)	REM
	Edge triggered wiping relay	REM
	Asynchronous pulse generator	REM
	Random generator	
	Stairway lighting switch	REM
	Multiple function switch	REM
	Weekly Timer	
	Yearly timer	
Counter		
	Up/down counter	REM
	Hours counter	REM
	Threshold trigger	

Analogue		
	Analogue Threshold Trigger	
	Analogue Differential Trigger	
	Analogue Comparator	
	Analogue Watchdog	
	Analogue Amplifier	
	Analogue Multiplexer	
	Average Value	
	Analogue Math	
	Long Datas Math	
	Analogue Ramp	
	PI Controller	
	Max/Min	
	Analogue Filter	
	Analogue Math Error Detection	
Miscellaneous		
	Latching Relay	
	Pulse Relay	
	Message Texts	
	Softkey	

	Shift Register	
	PWM (Pulse-Width Modulation)	
	Data Latching Relay	REM
	Long Data Latching Relay	REM
	Modbus Read	
	Modbus Write	
	Modbus Read Write	
	Memory Write	
	Memory Read	
	Word to Bit	REM
	Bit to Word	REM
	Device Reset	
	Comport Status	
	RH Math	
APP		
	Cam Control Block	REM
	Angular Cam Timer	

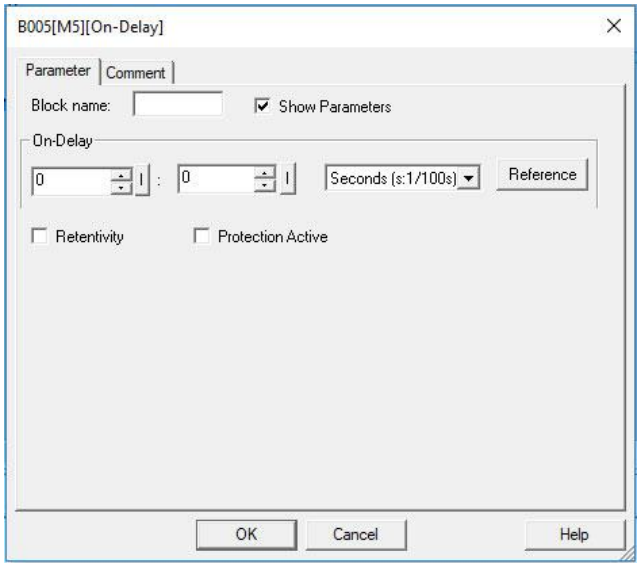
	Pumps Management	
	Defrost	
	Comparison of 2 values	
	Multicompare	
	Compare in Zone	
	Convert Word to Bits	
	Convert Bits to Word	
	Demultiplexer	
	Multiplexing	
	Multiplexer	
	Square root	
	Sin Cos	
	Absolute Humidity	

On-Delay

Description of function

The output is not switched on until a configured delay time has expired.

Connection	Description
Trg input (—●)	The on-delay time is triggered via the Trg (Trigger) input
Parameter	**T** represents the on-delay time after which the output is switched on (output signal transition 0 to 1). **Retentivity** on = the status is retentive in memory.
Output Q	Q switches on after a specified time T has expired, provided Trg is still set.



Parameter T

The time for parameter T can also be pre-set based on the actual value of another, already-configured function. You can use the actual values of the following functions:

Analog comparator (Ax – Ay)　　Analog threshold trigger (Ax)

Analog amplifier (Ax)　　Analog multiplexer (AQ)

Analog ramp (AQ)　　Mathematic instruction (AQ)

PI controller (AQ)　　Up/down counter (Cnt)

Analog filter (AQ)　　Average value (AQ)

Max/Min (AQ)　　On-delay (Ta)

Off-delay (Ta)　　On-/off-delay (Ta)

Retentive on-delay (Ta)　　Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta)　　Multiple function switch (Ta)

Stopwatch (Ta)

You select the required function via the block number. The time-base can be adjusted.

The value of "T" can be set/modified in parameter mode. For information about how to modify，refer to chapter 5.2.2, please.

For information on the validity and accuracy of the time base, refer to the xLogic time-base list as follows:

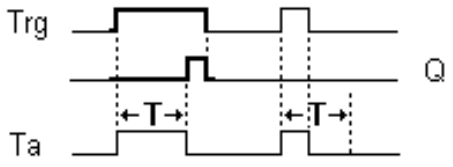
Valid ranges of the time-base, if T = parameter

Time-base	Max. value	Min. resolution	Accuracy
s (seconds)	99:99	10 ms	± 10 ms
m (minutes)	99:59	1s	± 1 s
h (hours)	99:59	1 min	± 1 min

Valid ranges of the time base, if T = Actual value of an already-programmed function

Time-base	Max. value	Meaning	Accuracy
ms	99990	Number of ms	± 10 ms
s	5999	Number of s	± 1 s
m	5999	Number of min	± 1 min

Timing diagram



The time Ta (the current time in xLogic) is triggered with the 0 to 1 transition at input Trg.

If the status at input Trg stays 1 at least for the duration of the configured time T, the output is set to 1 when this

time has expired (the on-signal of the output follows the on-signal of the input with delay).

The time is reset if the status at input Trg changes to 0 again before the time T has expired.

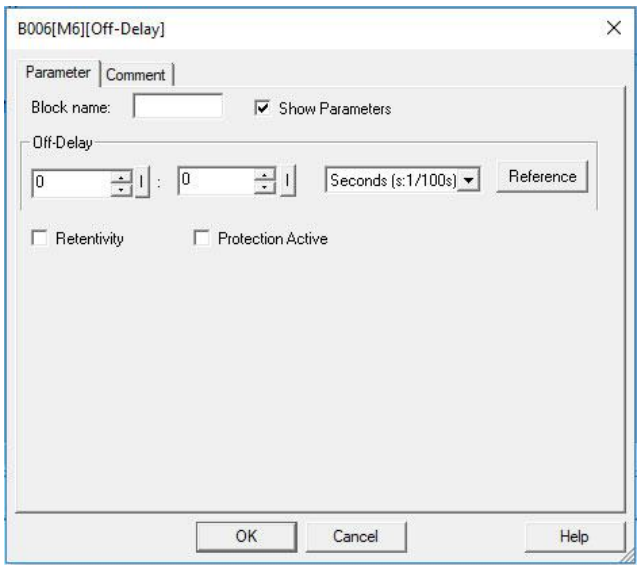
The output is reset to 0 when input Trg is 0.

Off-Delay

Description of function

The output with off delay is not reset until a defined time has expired.

Connection	Description
Trg input (—●)	Start the off-delay time with a negative edge (1 to 0 transition) at input Trg (Trigger)
Input R (—○)	Reset the off-delay time and set the output to 0 via the R (Reset) input. Reset has priority over Trg
Parameter	**T:** The output is switched off on the expiration of the delay time T (output signal transition 1 to 0). **Retentivity** on = the status is retentive in memory.
Output Q	Q is switched on for the duration of the time T after a trigger at input Trg.



Parameter

The time set in parameter T can be supplied by the value of another already-programmed function:

Analog comparator (Ax – Ay)　Analog threshold trigger (Ax)

Analog amplifier (Ax)　Analog multiplexer (AQ)

Analog ramp (AQ)　Mathematic instruction (AQ)

PI controller (AQ)　Up/down counter (Cnt)

Analog filter (AQ)　Average value (AQ)

Max/Min (AQ)　On-delay (Ta)

Off-delay (Ta)　On-/off-delay (Ta)

Retentive on-delay (Ta)　Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta)　Multiple function switch (Ta)

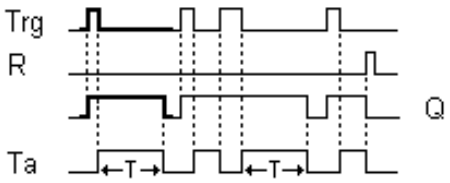
Stopwatch (Ta)

The value of "T" is set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Select the required function by the block number. The time base is configurable.

For information on valid time base ranges and parameter pre-set, refer to chapter 4.4.1 the On-delay topic.

Timing diagram



Output Q is set to 1 momentarily with a 0 to 1 transition at input Trg. At the 1 to 0 transition at input Trg, xLogic retriggers the current time T, and the output remains set. The output Q is reset to 0 when T_a reaches the value specified in T (T_a=T) (off delay).

A one-shot at input Trg retriggers the time Ta.

You can reset the time Ta and the output via the input R (Reset) before the time Ta has expired.

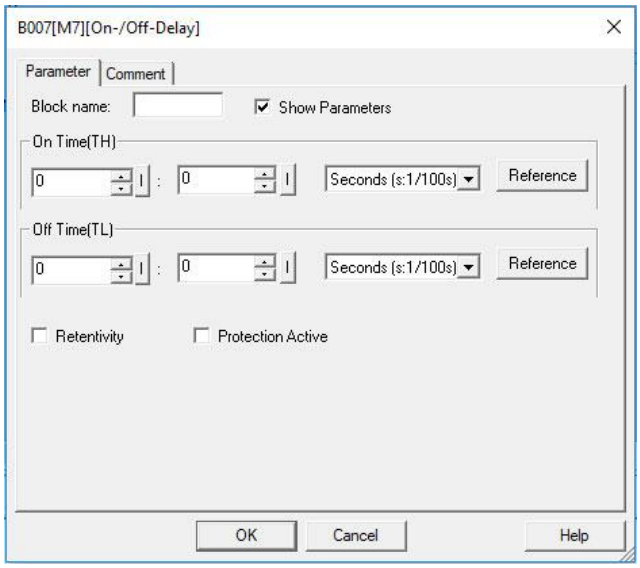
If retentivity is not set, output Q and the expired time are reset after a power failure.

On/Off-Delay

Description of function

The on/off delay function is used to set an output after a configured-on delay time and then reset it again upon expiration of a second configured time.

Connection	Description
Trg input (—●)	You trigger the on delay with a positive edge (0 to 1 transition) at input Trg (Trigger).
Parameter	**T_H** is the on-delay time for the output (output signal transition 0 to 1). **T_L** is the off-delay time for the output (output signal transition 1 to 0). **Retentivity** on = the status is retentive in memory.
Output Q	Q is switched on upon expiration of a configured time TH if Trg is still set. It is switched off again upon expiration of the time TL and if Trg has not been set again.



Parameter

The on-delay time and off-delay time set in parameter TH and TL can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) (Ax)

Analog threshold trigger

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ) (AQ)

Mathematic instruction

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta) (Ta)

Wiping relay(pulse output)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta)

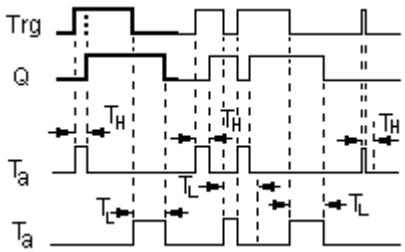
Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T_H" and "T_L" is set/modified in parameter mode. For information about how to modify, refer to chapter 7.2.2, please.

For information on the validity and accuracy of the time base, refer to 7.4.2

Timing diagram



The time T_H is triggered with a 0 to 1 transition at input Trg.

If the status at input Trg is 1 at least for the duration of the configured time T_H, the output is set to logical 1 upon expiration of this time (output is on delayed to the input signal).

The time T_H is reset if the status at input Trg is reset to 0 before this time has expired.

The time T_L is triggered with the 1 to 0 transition at the output.

If the status at input Trg remains 0 at least for the duration of a configured time T_L, the output is reset to 0 upon expiration of this time (output is off delayed to the input signal).

The time T_L is reset if the status at input Trg returns to 1 before this time has expired.

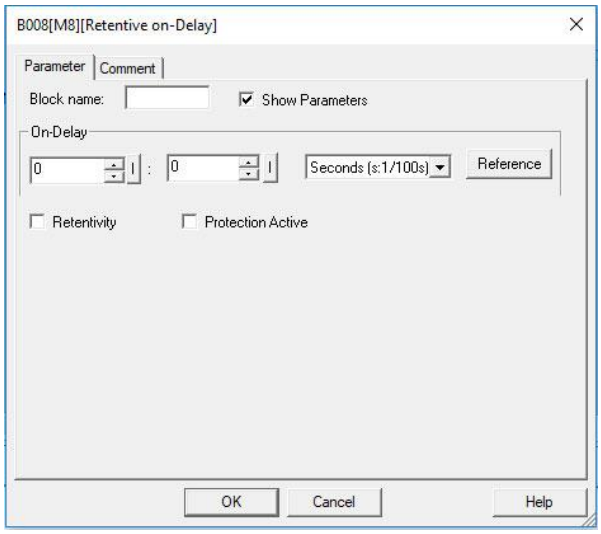
Retentive On-Delay

Description of function

A one-shot at the input triggers a configurable time. The output is set upon expiration of this time.

Connection	Description
Trg input (—●)	Trigger the on-delay time via the Trg (Trigger) input.

Input R (—○)	Reset the time on delay time and reset the output to 0 via input R (Reset). The reset takes priority over Trg.
Parameter	**T** is the on-delay time for the output (output signal transition 0 to 1). **Retentivity** on = the status is retentive in memory.
Output Q	Q is switched on upon expiration of the time T.



Parameter

The time in parameter T can be provided by the value of another already-programmed function:

Analog comparator (Ax – Ay) Analog threshold trigger (Ax)

Analog amplifier (Ax) Analog multiplexer (AQ)

Analog ramp (AQ) Mathematic instruction (AQ)

PI controller (AQ) Up/down counter (Cnt)

Analog filter (AQ) Average value (AQ)

Max/Min (AQ) On-delay (Ta)

Off-delay (Ta) On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

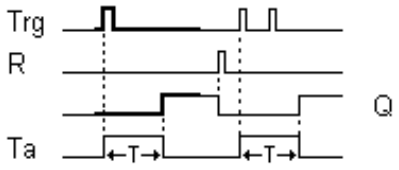
Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T" can be set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Timing diagram



The current time Ta is triggered with a 0 to 1 signal transition at input Trg. Output Q is set to 1 when Ta reaches the time T. A further pulse at input Trg does not affect Ta.

The output and the time Ta are only reset to 0 with a '1' signal at input R.

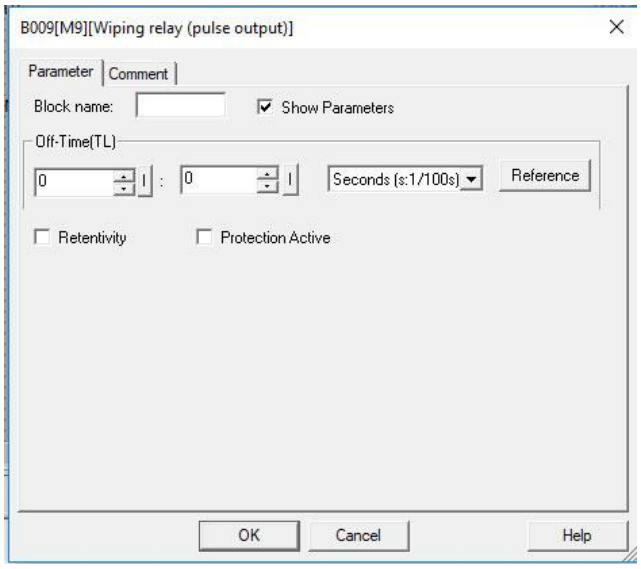
If retentivity is not set, output Q and the expired time are reset after a power failure.

Wiping Relay (pulse output)

Description of function

An input signal generates an output signal of a configurable length.

Connection	Description
Trg input (—●)	You trigger the time for the wiping relay with a signal at input Trg (Trigger)
Parameter	**T_L** represents the time after which the output is reset (output signal transition 1 to 0). **Retentivity** set (on) = the status is retentive in memory.
Output Q	A pulse at Trg sets Q. The output stays set until the time T has expired and if Trg = 1 for the duration of this time. A 1 to 0 transition at Trg prior to the expiration of T also resets the output to 0.



Parameter

The off-time T can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) Analog threshold trigger (Ax)

Analog amplifier (Ax) Analog multiplexer (AQ)

Analog ramp (AQ) Mathematic instruction (AQ)

PI controller (AQ) Up/down counter (Cnt)

Analog filter (AQ) Average value (AQ)

Max/Min (AQ) On-delay (Ta)

Off-delay (Ta) On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

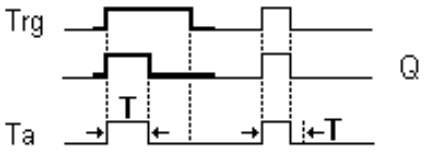
Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T_L" can be set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Timing diagram



With the input signal Trg = 1, output Q is set to 1. The signal also triggers the time Ta, while the output remains set.

When Ta reaches the value defined at T (Ta=T), the output Q is reset to 0 state (pulse output).

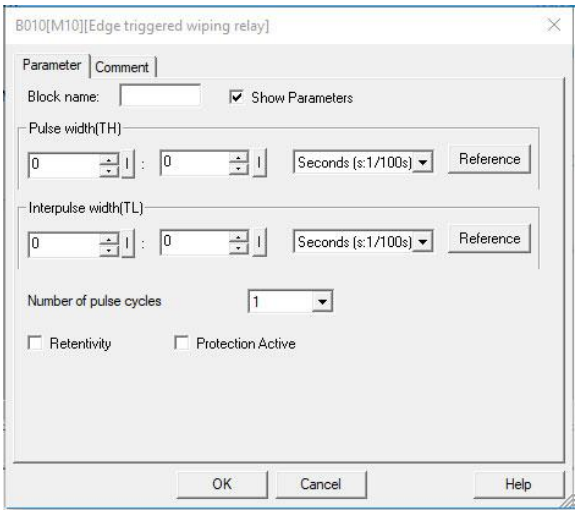
If the signal at input Trg changes from 1 to 0 before this time has expired, the output is immediately reset from 1 to 0.

Edge Triggered Wiping Relay

Description of function

An input pulse generates a pre-set number of output pulses with a defined pulse/pause ratio (retriggerable) after a configured delay time has expired.

Connection	Description
Trg input (—●)	You trigger the times for the Edge-triggered wiping relay with a signal at input Trg (Trigger).
Input R (—○)	The output and the current time Ta are reset to 0 with a signal at input R.
Parameter	**T_L, T_H**: The inter-pulse period TL and the pulse period TH are adjustable. **N** determines the number of pulse/pause cycles TL / TH: Value range: 1...9. **Retentivity** set (on) = the status is retentive in memory
Output Q	Output Q is set when the time T_L has expired and is reset when T_H has expired.



Parameter

The pulse width TH and the inter-pulse width TL can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)	Analog threshold trigger (Ax)
Analog amplifier (Ax)	Analog multiplexer (AQ)
Analog ramp (AQ)	Mathematic instruction (AQ)
PI controller (AQ)	Up/down counter (Cnt)
Analog filter (AQ)	Average value (AQ)
Max/Min (AQ)	On-delay (Ta)
Off-delay (Ta)	On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

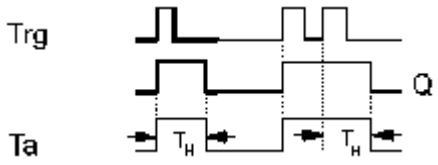
Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T_H" and "T_L" can be set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Timing diagram



T_L = 0; N = 1

With the change at input Trg to 1, the time T_L (time low) is triggered. After the time T_L has expired, output Q is set to 1 for the duration of the time T_H (time high).

If input Trg is retriggered prior to the expiration of the preset time (T_L + T_H), the time Ta is reset, and the pulse/pause period is restarted.

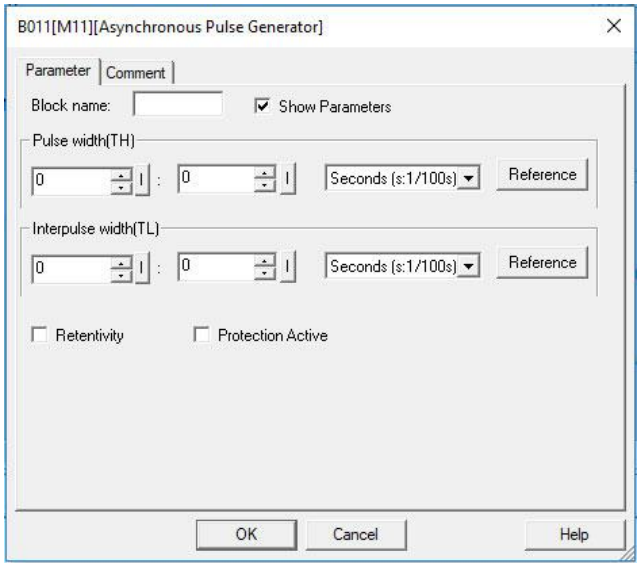
If retentivity is not set, output Q and the time are reset after a power failure.

Asynchronous Pulse Generator

Description of function

The pulse shape at the output can be modified via a configurable pulse/pause ratio.

Connection	Description
Input En (—▸)	You enable/disable the asynchronous pulse generator with the signal at input En.
Input Inv	The Inv input can be used to invert the output signal of the active asynchronous pulse generator.
Parameter	**T_L, T_H**: You can customize the pulse (T_L)/ pause (T_H) ratio. **Retentivity** set (on) = the status is retentive in memory.
Output Q	Q is toggled on and off cyclically with the pulse times T_H and T_L.



Parameter

The pulse width T_H and the inter-pulse width T_L can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) Analog threshold trigger (Ax)

Analog amplifier (Ax) Analog multiplexer (AQ)

Analog ramp (AQ) Mathematic instruction (AQ)

PI controller (AQ) Up/down counter (Cnt)

Analog filter (AQ) Average value (AQ)

Max/Min (AQ) On-delay (Ta)

Off-delay (Ta) On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

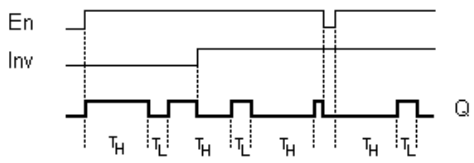
Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T_H" and "T_L" can be set/modified in parameter mode. For information about how to modify, please refer to chapter 5.2.2.

Timing diagram



You can set the pulse/pause ratio at the T_H (Time High) and T_L (Time Low) parameters.

The INV input can be used to invert the output signal. The input block INV only inverts the output signal if the block is enabled via EN.

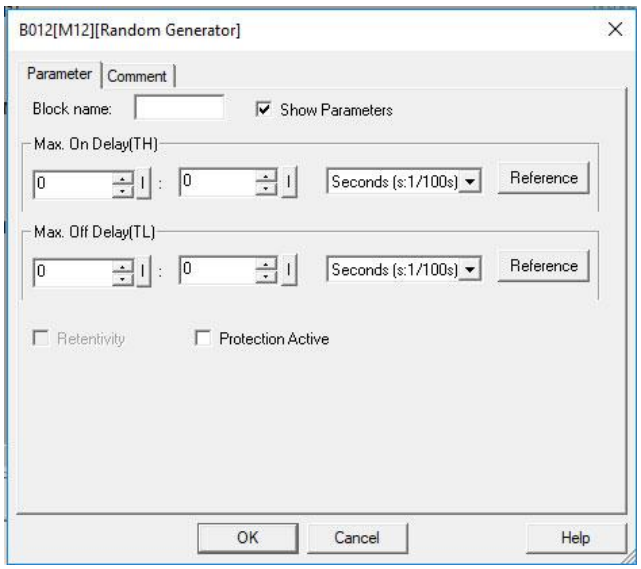
If retentivity is not set, output Q and the expired time are reset after a power failure.

Random Generator

Description of function

The output of a random generator is toggled within a configurable time.

Connection	Description
Input En (—▸)	The positive edge (0 to 1 transition) at the enable input En (Enable) triggers the on delay for the random generator. The negative edge (1 to 0 transition) triggers the off delay for the random generator.
Parameter	**T_H**: The on delay is determined at random and lies between 0 s and T_H. **T_L**: The off delay is determined at random and lies between 0 s and T_L.
Output Q	Q is set on, expiration of the on delay if En is still set. It is reset when the off-delay timer has expired and if En has not been set again.



Parameter

The on-delay time T_H and the off-delay time T_L can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)　　Analog threshold trigger (Ax)

Analog amplifier (Ax)　　Analog multiplexer (AQ)

Analog ramp (AQ)　　Mathematic instruction (AQ)

PI controller (AQ)　　Up/down counter (Cnt)

Analog filter (AQ)　　Average value (AQ)

Max/Min (AQ)　　On-delay (Ta)

Off-delay (Ta)　　On-/off-delay (Ta)

Retentive on-delay (Ta)　　Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

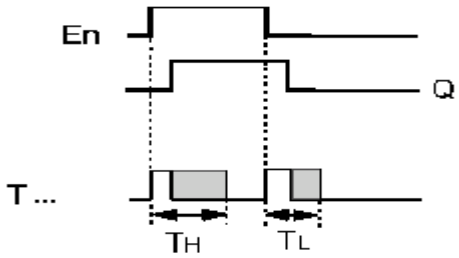
Asynchronous pulse generator (Ta)

Stairway light switch (Ta)　　Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T_H" and "T_L" can be set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Timing diagram



With the 0 to 1 transition at input En, a random time (on delay time) between 0 s and T_H is set and triggered. If the status at input En is 1 at least for the duration of the On delay, the output is set to 1 when this On delay time has expired.

The time is reset if the status at input En is reset to 0 before the On-delay time has expired.

When input En is reset 0, a random time (off delay time) between 0 s and T_L is set and triggered.

If the status at input En is 0 at least for the duration of the off-delay time, the output Q is reset to 0 when the off-delay time has expired.

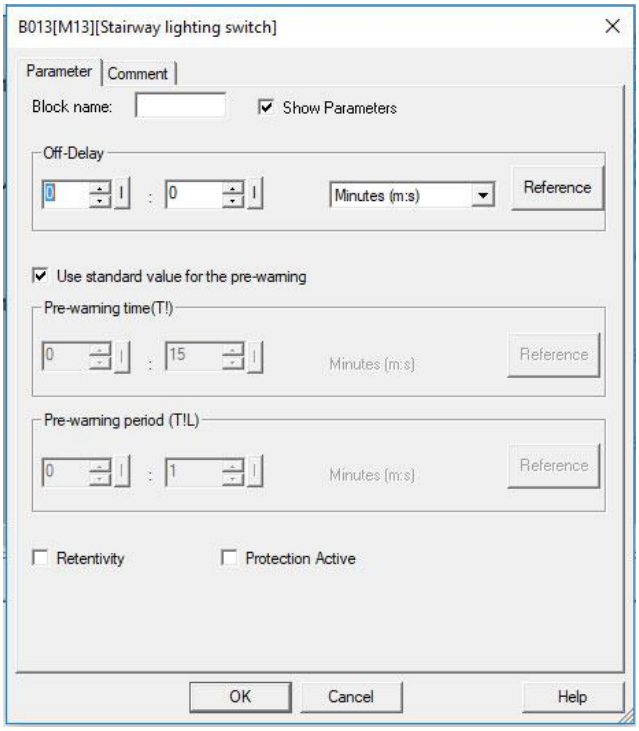
The time is reset if the status at input En returns to 1 before the on-delay time has expired.

Stairway Lighting Switch

Description of function

The edge of an input pulse triggers a configurable time. The output is reset when this time has expired. An off warning can be output prior to the expiration of this time.

Connection	Description
Trg input (—●)	You trigger the time (off delay) for the stairway switch with a signal at input Trg (Trigger).
Parameter	**T**: The output is reset (1 to 0 transition when the time T has expired. **T_I**: Determines the triggering time for the pre-warning. **T_{IL}**: Determines the length of the pre-warning time. **Retentivity** set (on) = the status is retentive in memory.
Output Q	Q is reset after the time T has expired. A warning signal can be output before this time has expired.



Parameter

The off-delay time T, the pre-warning time T! and the pre-warning period T!L can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)

Analog threshold trigger (Ax)

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta)

Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

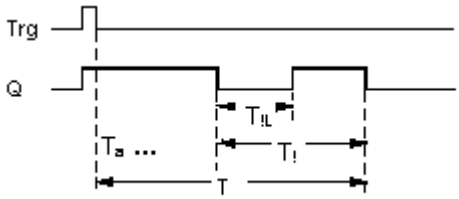
Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "T" can be set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Timing diagram



Changing the time base

You can change the pre-warning time base and the period.

Time base T	Pre-warning time	Pre-warning period
Seconds	750 ms	50 ms
Minutes	15 s	1 s
Hours	15 min	1 min

*** makes sense only for programs with a cycle time of < 25 ms**

Output Q is set to 1 with a 0 to 1 signal transition at input Trg. The 1 to 0 transition at input Trg triggers the current time and output Q remains set.

Output Q is reset to 0 when Ta reaches the time T. Before the off-delay time (T - $T_!$) has expired, you can output a pre-warning that resets Q for the duration of the off-pre-warning time $T_{!L}$.

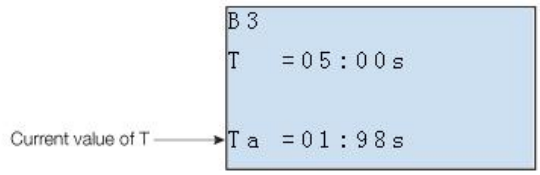
Ta is retriggered (optional) at the next high/low transition at input Trg and if Ta is expiring.

If retentivity is not set, output Q and the expired time are reset after a power failure.

Setting the Par parameter

NOTICE:
All times must have the same time base.

View in parameter assignment mode (example):



Multiple Function Switch

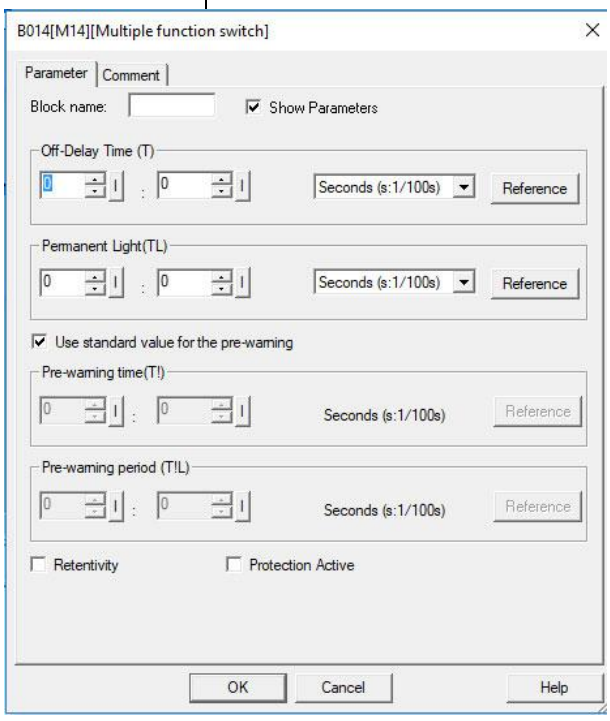
Description of function

Switched with two different functions:

1. Pulse switch with off delay
2. Switch (continuous light)

Connection	Description
Trg input (—●)	With a signal at input Trg (Trigger), you set output Q (continuous light), or reset Q with off delay. Output Q can be reset with a signal at the Trg input.
Input R (—○)	You set the current time T_a, and reset the output to 0, with a signal at input R.
Parameter	**T**: The output is reset (1 to 0 transition) when the time T has expired.

	T_L: Determines the period during which the input must be set in order to enable the permanent light function. **$T_!$**: Determines the on delay for the pre-warning time. **$T_{!L}$**: Determines the length of the pre-warning time. **Retentivity** set (on) = the status is retentive in memory.
Output Q	Output Q is set with a signal at input Trg, and it is reset again after a configured time has expired and depending on the pulse width at input Trg, or it is reset with another signal at input Trg.



Parameter

The off-delay time T, the permanent light time TL, the on-delay pre-warning time T!, and the pre-warning time period T!L can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) (Ax)

Analog threshold trigger

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta)

Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

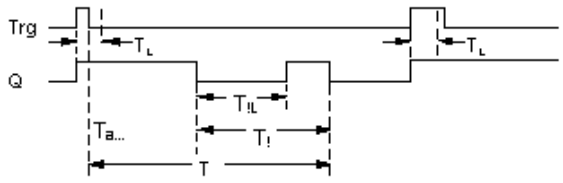
Stairway light switch (Ta)

Multiple function switch (Ta)

Stopwatch (Ta)

The value of "TL", "T"can be set/modified in parameter mode. For information about how to modify, refer to chapter 5.2.2, please.

Timing diagram



Output Q is set to 1 with a 0 to 1 signal transition at Trg.

If output Q = 0, and input Trg is set hi at least for the duration of TL, the permanent lighting function is enabled, and output Q is set accordingly.

The off-delay time T is triggered when the status at input Trg changes to 0 before the time T_L has expired.

Output Q is reset when the Ta = T.

Before the off-delay time (T - T_I) has expired, you can output an off pre-warning that resets Q for the duration of the off-pre-warning time T_{IL}.

A further signal at input Trg always resets T and output Q.

Caution

The time base for the T, T_I and T_{IL} must be identical.

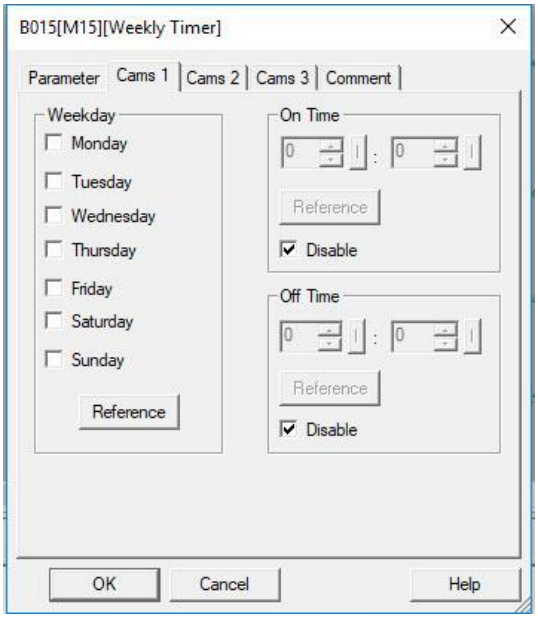
If retentivity is not set, output Q and the expired time are reset after a power failure.

Weekly Timer

Description of function

The output is controlled by means of a configurable on/off date. The function supports any combination of weekdays.

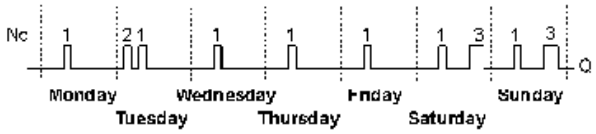
Connection	Description
Parameter	At the **No1**, **No2**, **No3** (cam) parameters you set the on and off triggers for each cam of the weekly timer. The parameter units are the days and the time-of-day.
Output Q	Q is set when the configured cam is actuated.



Parameter

You can configure a time hysteresis for each individual cam in parameter mode. For information about how to modify, refer to chapter 4.2.2, please.

Timing diagram (three practical examples)



No1: Daily: 06:30 h to 08:00 h

No2: Tuesday: 03:10 h to 04:15 h

No3: Saturday and Sunday: 16:30 h to 23:10 h

Description of the function

Each weekly timer is equipped with three cams. You can configure a time hysteresis for each individual cam. For the cams, you set the on and off hysteresis. The weekly timer sets the output at a certain time, provided it is not already set.

The output is reset at a certain time, provided it is not already reset. A conflict is generated in the weekly timer when the set on time and the set off time at another cam are identical. In this case, cam 3 takes priority over cam 2, while cam 2 takes priority over cam 1.

The switching status of the weekly timer is determined by the status at the No1, No2 and No3 cams.

Days of the week

The prefix "D=" (Day) has the following meaning:

- M: Monday
- T: Tuesday
- W: Wednesday
- T: Thursday
- F: Friday

On-/Off-times

Any time between 00:00h and 23:59h is possible. You can also configure the on time to be a pulse signal. The timer block will be activated at the specified time for one cycle and then the output is reset.

- -:- - means: No on/off-times set.

Setting the weekly timer

To set the on-/off-times:

1. Move the cursor to one of the Cam parameters of the timer (e.g. No1).

2. Press OK. xLogic opens the Cam parameter assignment screen form. The cursor is positioned on the day of the week.

3. Press ▲ and ▼ to select one or several days of the week.

4. Press ▶ to move the cursor to the first position of the on-time.

5. Set the on-time.

Modify the value at the respective position, using the keys ▲ and ▼. Move to the cursor to the various positions, using the keys ◀ and ▶. At the first position, you can only select the value - -:- - (- -:- - means: No on/off-times set).

6. Press ▶ to move the cursor to the first position of the off time.

7. Set the off time (in the same way as in step 5).

8. Confirm your entries with **OK**.

The cursor is now positioned on the No2 parameter (Cam2) and you can configure a further cam.

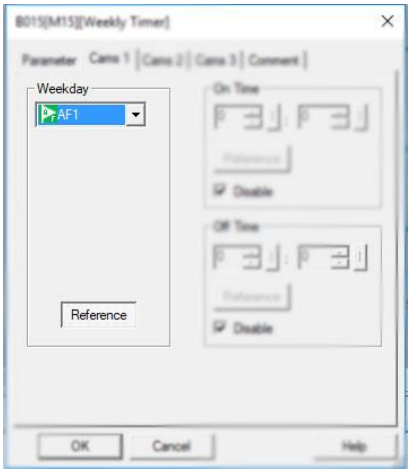
Special characteristics to note when configuring

The block properties window offers a tab for each one of the three cams. Here you can set the weekly on times for the cams. Each tab offers you, in addition, an option of defining the on and off times for each cam in hour and minute units. Hence, the shortest switching cycle is one minute.

You can disable the on and off times individually, i.e. you can achieve switching cycles extending across more than one day, for example, by setting the on time for cam 1 to Monday 7:00h and the off time of cam 2 to Wednesday 13:07h, while disabling the on time for cam 2.

Backup of the real-time clock

The internal real-time clock of xLogic is buffered against power failure. The buffering time is influenced by the ambient temperature and is typically 100 hours at an ambient temperature of 25°C.



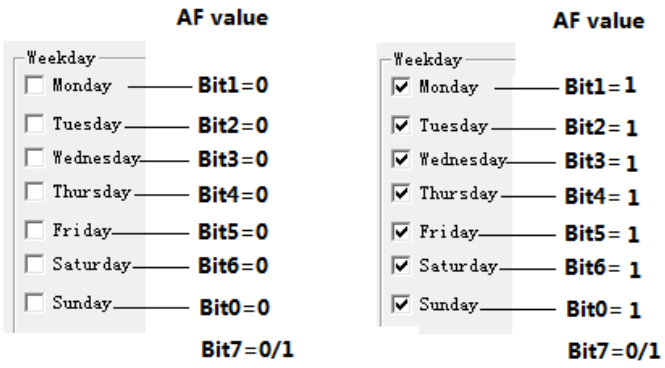
In order to set the weekly timer parameters via MODBUS, they can be assigned to an Analogue flag, AF, which would hold a value to represent the bits for each day of the week.

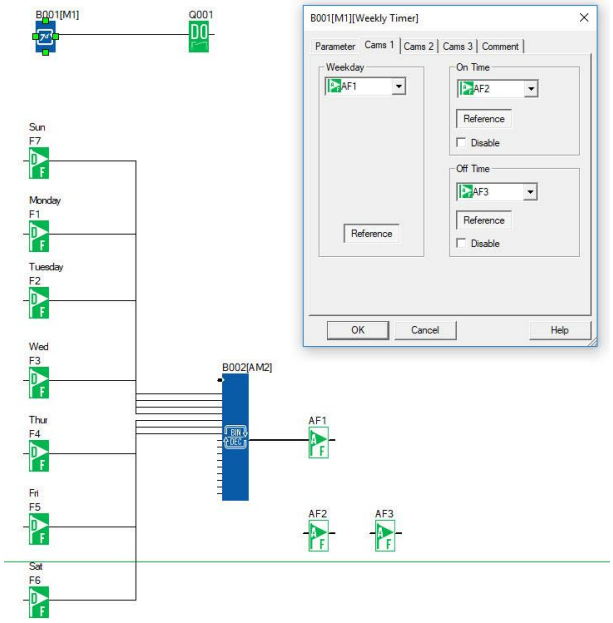
Weekday parameter.

For example, use AF1

AF has a 16-bit type value, the weekday uses the low 8-bits status as the parameters:

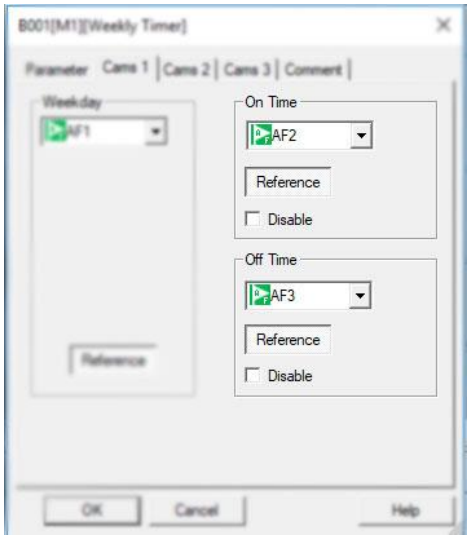
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0 (Reserved)	SAT	FRI	THU	WED	TUE	MON	SUN





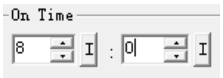
You can program as above, use one Bit to word block and then, you can set the F1--F7 at the touch screen to change the weekday settings.

On time/Off Time settings--Reference AF value.

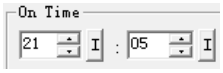


For example,

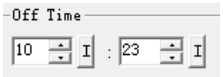
- If AF2 =800, then the On time would be 08:00;



- If AF2=2105, then the On time would be 21:05;



- If AF3=1023, then the OFF time would be 10:23;



NOTICE:

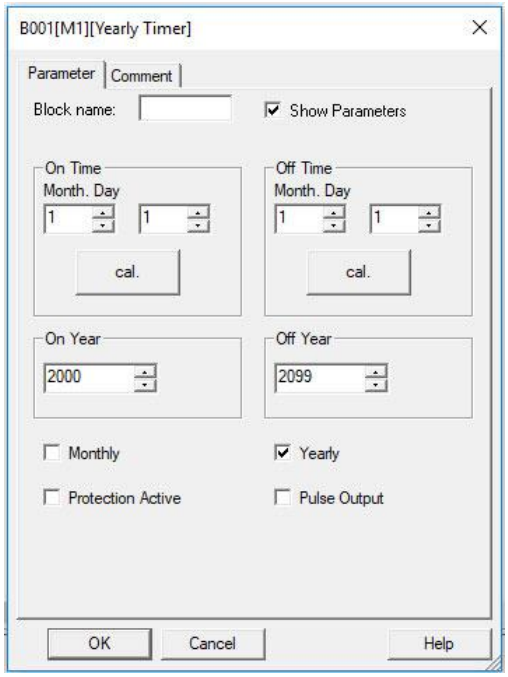
If the AF value is over 2323, it will be not available for the On/Off time, the output of the weekly timer will be 0.

Yearly Timer

Description of function

The output is controlled by means of a configurable on/off date

Connection	Description
Parameter	At the **No** (cam) parameter you set the on and off trigger for the cam of the yearly timer.
Output Q	Q is set on when the configured cam is switched on.

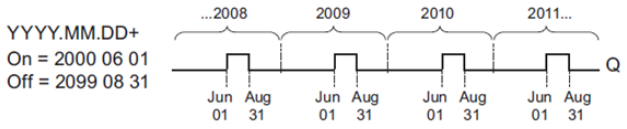


Parameter

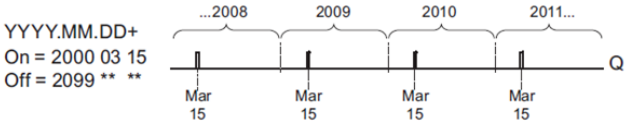
The on and off trigger for the cam of the yearly timer can be set/modified in parameter mode and you can configure what you need. For information about how to modify, please refer to chapter 5.2.2.

Timing diagrams

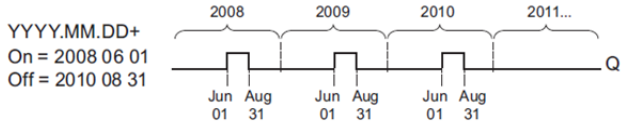
Example 1: Yearly mode on, Monthly mode off, Pulse Off, On Time = 2000-06-01, Off Time = 2099-08-31: Every year on June 1 the timer output switches on and remains on until August 31.



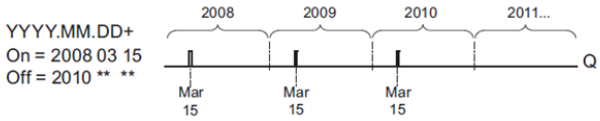
Example 2: Yearly mode on, Monthly mode off, Pulse on, On Time = 2000-03-15, Off Time = 2099-**-**: Every year on March 15, the timer switches on for one cycle.



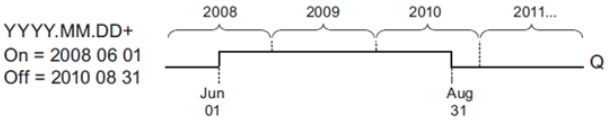
Example 3: Yearly mode on, Monthly mode off, Pulse off, On Time = 2008-06-01, Off Time = 2010-08-31: On June 1 of 2008, 2009, and 2010 the timer output switches on and remains on until August 31.



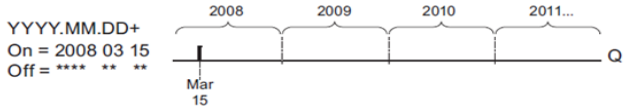
Example 4: Yearly mode on, Monthly mode off, Pulse on, On Time = 2008-03-15, Off Time = 2010-**-**: On March 15 of 2008, 2009, and 2010, the timer output switches on for one cycle.



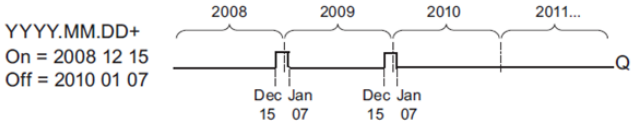
Example 5: Yearly mode off, Monthly mode off, Pulse off, On Time = 2008-06-01, Off Time = 2008-08-31: On June 1, 2008, the timer output switches on and remains on until August 31, 2010.



Example 6: Yearly mode off, Monthly mode off, Pulse selected, On Time = 2008-03-15, Off Time = ****-**-**: On March 15, 2008, the timer output switches on for one cycle. Because the timer does not have a monthly or yearly action, the timer output pulses only one time at the specified-On Time.

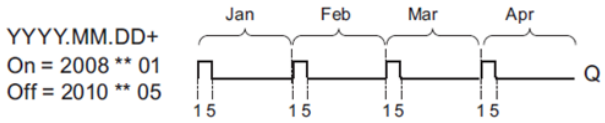


Example 7: Yearly mode on, Monthly mode off, Pulse off, On Time = 2008-12-15, Off Time = 2010-01-07: On December 15 of 2008 and 2009, the timer output switches on and remains on until January 7 of the following year. When the timer output turns off on January 7, 2010, it does NOT turn on again the following December 15.



Example 8: Yearly mode on, Monthly mode on, On Time = 2008-**-01, Off Time = 2010-**-05: Starting in 2008, on the first day of each month the timer output switches on

and switches off on the fifth day of the month. The timer continues in this pattern through the last month of 2010.



Description of the function

The yearly timer sets and resets the output at specific on and off times. The off date identifies the day on which the output is reset again. The first value defines the month, the second the day.

When you select the every month check box, the yearly clock switches on or off at a certain day of every month.

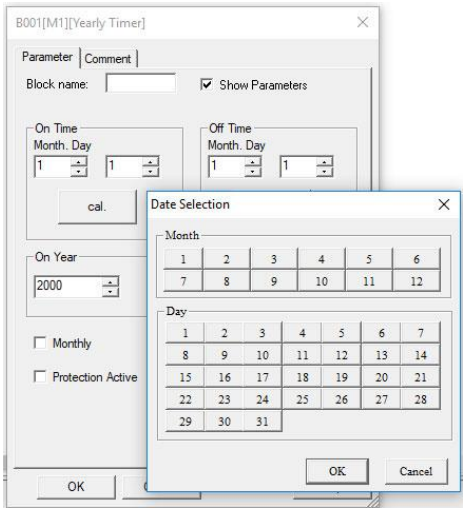
Backup of the real-time clock

The internal real-time clock of xLogic is buffered against power failure. The buffering time is influenced by the ambient temperature and is typically 100 hours at an ambient temperature of 25°C.

Special characteristics to note when configuring

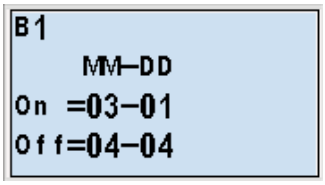
A click on the dialogue box enables direct keyboard input of the month and day values. The values entered may not exceed the logical maximum of the relevant input boxes; otherwise xLogicsoft returns an error message.

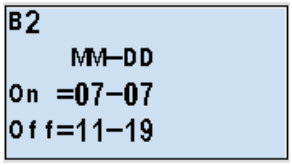
The calendar icon offers you a Rievtech way of setting the date. It opens a window where you can set the days and months by clicking the relevant buttons.



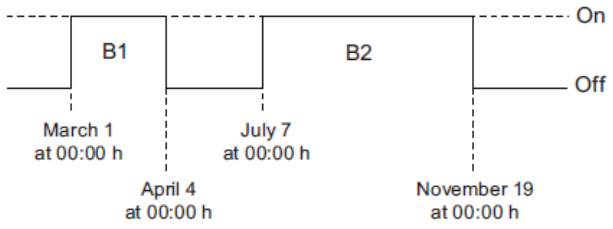
Sample configuration

The output of an xLogic is to be set annually on March 1, reset on April 4, set again on July 7, and reset again on November 19. You need to configure two yearly timers with corresponding on-times. Then logically link the outputs by means of an OR block.

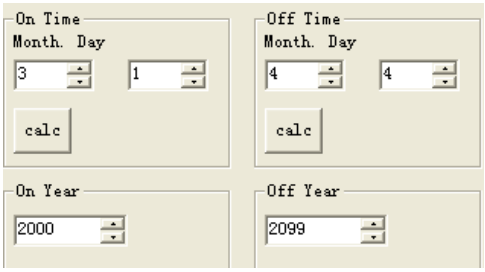


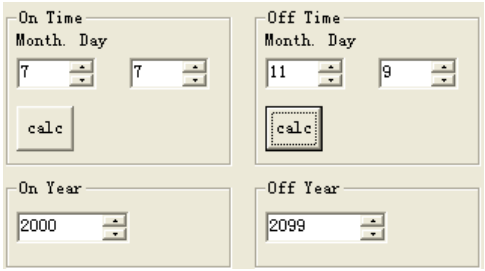


Result

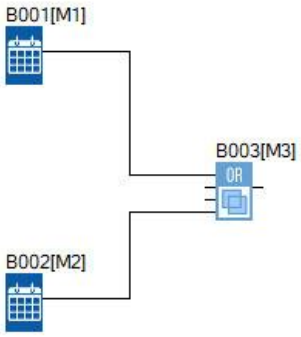


Place two yearly timer switch SFBs on your programming interface and configure the blocks as specified.





Create a logical link of the blocks via a standard OR block. The OR output is 1 if at least one of the yearly timer switches is set.

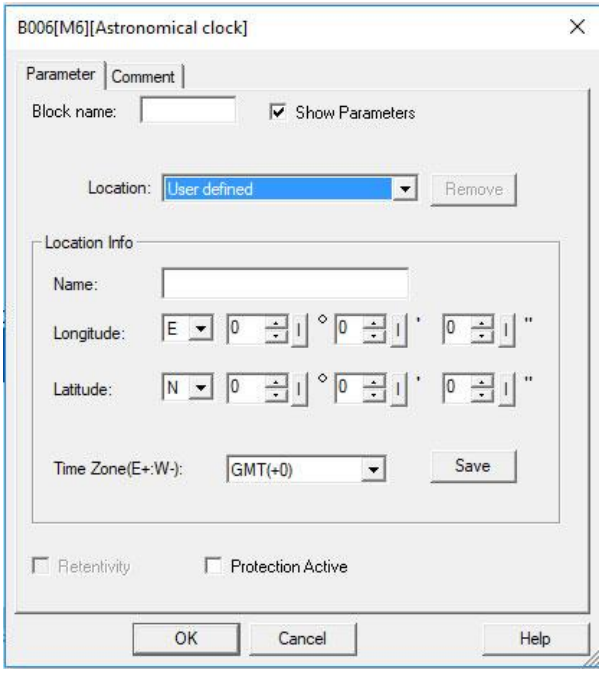


Astronomical Clock

Description of function

The astronomical clock SFB is used to set an output high between sunrise and sunset based on the local time at the geographical location of the xLogic device. The output status of this function block also depends on the configuration of Summertime/Winter time or Daylight Savings settings.

Connection	Description
Parameter	The location info including longitude, latitude and time zone.
Output Q	Q is set to Hi when the sunrise time is reached. It holds this state until sunset time is reached.



Parameters

In the astronomical clock dialogue, you can select the location of the xLogic device. You can select one of the following pre-defined time zone locations:

Beijing Berlin London Rome Moscow Tokyo

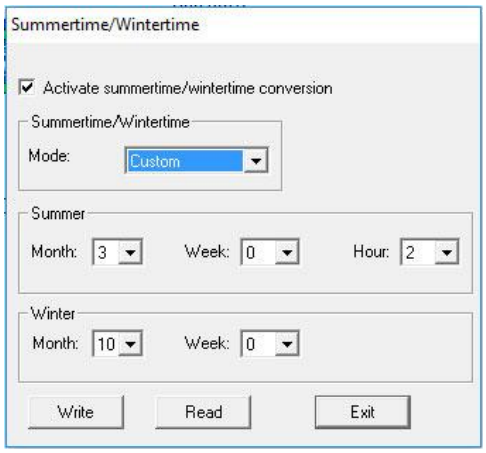
Washington Ankara Madrid Amsterdam

If you select one of these locations, xLogicsoft uses the latitude, longitude, and time zone of your selection.

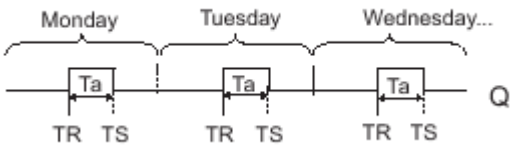
Alternatively, you can configure a specific latitude, longitude, and time zone for your location, and provide a name for this custom location.

Based on the location and time zone, xLogic calculates the absolute sunrise and sunset time for the current day. The block also takes Summertime/Winter Daylight Savings time into consideration, if it is configured on the computer where xLogicsoft is installed. To do such configuration, you should select the checkbox of "Automatically adjust clock for daylight for saving changes" in the "Date and Time Properties" dialogue.

You can also set this up in the PLC using the Transfer.....Summertime/Winter Time menu option:



Timing diagram



Description of function

The function calculates the value at the input and sets or resets Q depending on the sunrise time and sunset time at the configured location and time zone of the module.

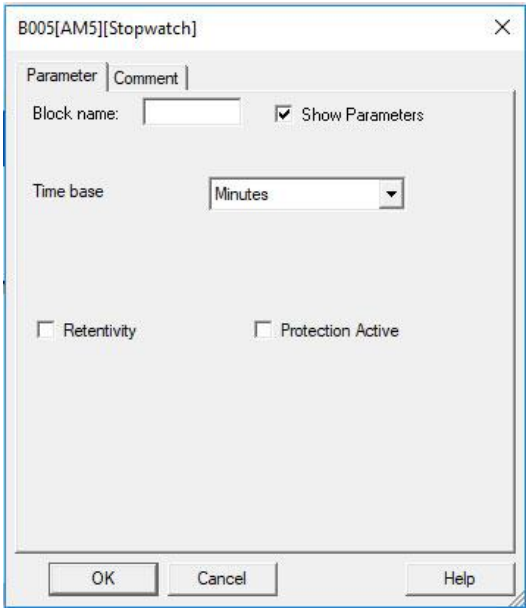
Stopwatch

Description of function

The stopwatch records the time elapsed since it was enabled.

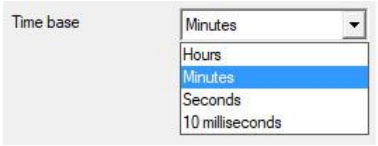
Connection	Description
Input En (—▷)	En (Enable) is the monitoring input. xLogic sets the current elapsed time to 0 and begins counting elapsed time when En transitions from 0 to 1. When En transitions from 1 to 0, the elapsed time is frozen.
Input Lap	A positive edge (0 to 1 transition) at input Lap pauses the stopwatch and sets the output to lap time. A negative edge (1 to 0 transition) at input Lap resumes the stopwatch and set the output to current elapsed time.
Input R (—○)	A signal at input R (Reset) clears the current elapsed time and lap time.
Parameter	The time base for the elapsed time can be set to hours, minutes, seconds, or 1/100ths of seconds.
Output AQ	The output AQ outputs value of the current elapsed time when it is a negative edge (1 to 0 transition) at the input Lap, and outputs value of the Lap time when it is a positive

	edge (0 to 1 transition) at the input Lap. A positive edge (0 to 1 transition) resets the value at output AQ to 0.



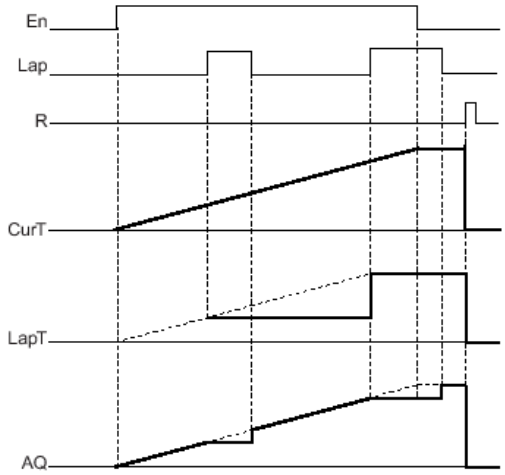
Parameters Time Base

You can configure the time base for the analogue output:



The time base for the elapsed time can be in hours, minutes, seconds, or 1/100ths of seconds (units of 10 milliseconds). The smallest time base, and therefore the resolution, is 10 milliseconds, or 1/100ths of a second.

Timing diagram



Description of the function

When En = 1, the current time increases.

When En = 0, the current time counting pauses.

When En = 1 and Lap = 0, the output AQ outputs the value of the current elapsed time.

When En = 1 and Lap = 1, the current time continue increasing, but the output AQ outputs the value of the Lap time.

When En = 0 and Lap =1, the output AQ outputs the value of the Lap time.

When En = 0 and Lap = 0, the output AQ outputs the value of the latest current time.

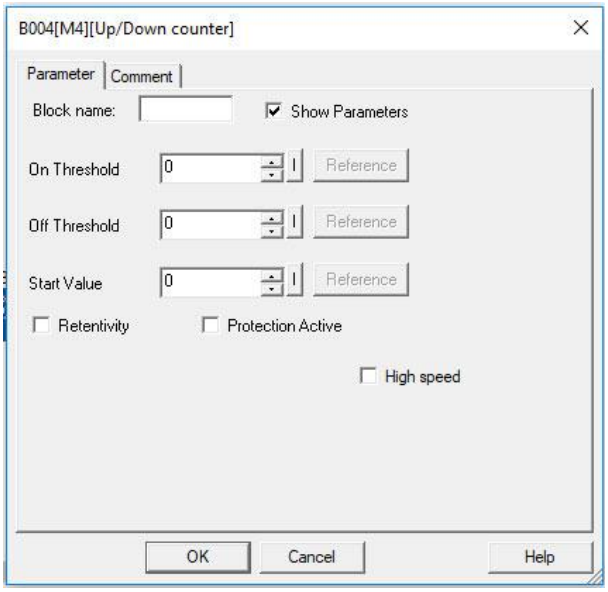
When R = 1, both the current time and the Lap time are reset.

Up/Down Counter

Description of function

An input pulse increments or decrements an internal value, depending on the parameter setting. The output is set or reset when a configured threshold is reached. The direction of count can be changed with a signal at input Dir

Connection	Description
Input R (—○)	You reset the output and the internal counter value to zero with a signal at input R (Reset).
Input Cnt (—●)	This function counts the 0 to 1 transition at input Cnt. It does not count 1 to 0 transitions.
Input Dir	Input Dir (Direction) determines the direction of count: Dir = 0: Up Dir = 1: Down
Parameter	**On**: On threshold Value range: 0...99999999 **Off**: Off threshold Value range: 0...99999999 **Start Val**: Initial value from which to begin counting either down or up. **Retentivity** set (on) = the status is retentive in memory.
Output Q	Q is set and reset according to the actual value at Cnt and the set thresholds.



Parameter

The on threshold On and the off threshold Off can be provided by the value of another already-programmed function:

Analog comparator (Ax – Ay) Analog threshold trigger (Ax)

Analog amplifier (Ax) Analog multiplexer (AQ)

Analog ramp (AQ) Mathematic instruction (AQ)

PI controller (AQ) Up/down counter (Cnt)

Analog filter (AQ) Average value (AQ)

Max/Min (AQ) On-delay (Ta)

Off-delay (Ta) On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

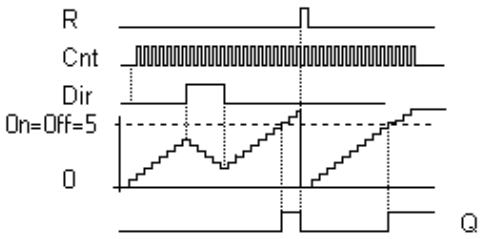
Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "On", "Off" and "Cnt" can be set/modified in parameter mode.

Timing diagram



Description of the function

The function increments (Dir = 0) or decrements (Dir = 1) the internal counter by one count with every positive edge at input Cnt.

You can reset the internal counter value to '000000', with a signal at the reset input R. As long as R=1, the output is 0 and the pulses at input Cnt are not counted.

Output Q is set and reset according to the actual value at Cnt and the set thresholds. See the following rules for calculation.

Calculation rule

If the on threshold >= off threshold, then:
Q = 1, if Cnt >= On
Q = 0, if Cnt < Off.

If the on threshold < off threshold, then:
Q = 1, if On <= Cnt < Off.

Caution

The function polls the limit value of the counter once in each cycle.

Thus, if the pulses at the fast inputs are faster than the scan cycle time, the SFB might not switch until the specified limit has been exceeded.

Example: Up to 100 pulses per cycle can be counted; 900 pulses have been counted so far. On = 950; Off = 10000. The output is set in the next cycle after the value has reached 1000.

The output would not be set at all if the value Off = 980

Hours Counter

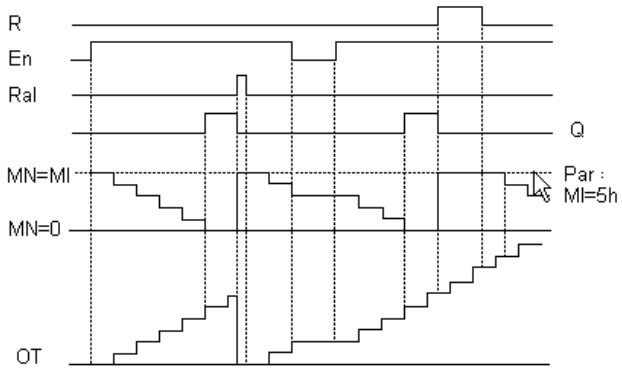
Description of function

A configured time is triggered by a signal at the monitoring input. The output is set when this time has expired.

Connection	Description
Input R (—○)	A positive edge (0 to 1 transition) at input R resets output Q and sets a configured value MI at the counter for the duration of the time-to-go (MN).
Input En (—▷)	En is the monitoring input. xLogic scans the on-time of this input.
Input Ral (—●)	A positive edge at input Ral (Reset all) resets the hours counter (OT) and the output, and sets the time-to-go value (MN) to the maintenance interval MI:

	• Output Q = 0 • Measured operating time OT = 0 • Time-to-go of the maintenance interval MN = MI.
Parameter	**MI**: Maintenance interval to be pre-set in units of hours and minutes Range of values: 0000 h to 9999 h, 0 m to 59 m **OT**: the accumulated total operating time (you can specify an offset in hours and minute) Range of values: 00000 h to 99999 h, 0 m to 59 m Q → 0: • When "R" is selected: Q = 1, if MN = 0; Q = 0, if R = 1 or Ral = 1 • When "R+En" is selected: Q = 1, if MN = 0; Q = 0, if R = 1 or Ral = 1 or En = 0.
Output Q	The output is set when the time-to-go MN = 0. The output is reset: • When "Q→0:R+En", if R = 1 or Ral = 1 or En = 0 • When "Q→0:R", if R = 1 or Ral = 1.

Timing diagram

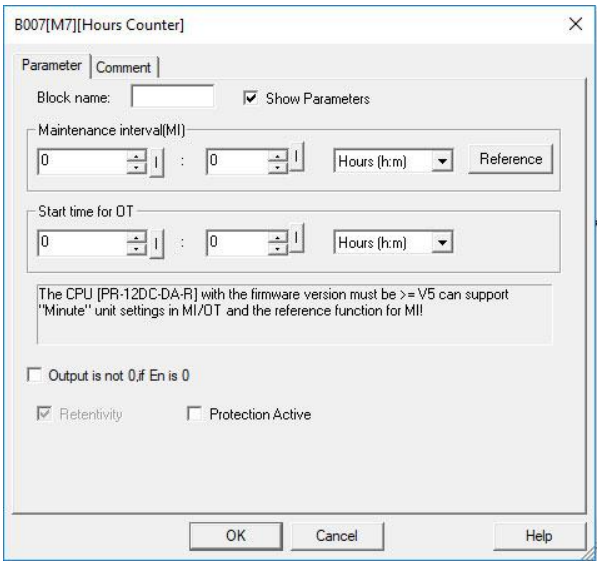


MI = Configured time interval

MN = Time-to-go

OT = Total time expired since the last 1 signal at the Ral input

These values are principally held retentive!



Parameter

The maintenance interval MI can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) Analog threshold trigger (Ax)

Analog amplifier (Ax) Analog multiplexer (AQ)

Analog ramp (AQ) Mathematic instruction (AQ)

PI controller (AQ) Up/down counter (Cnt)

Analog filter (AQ) Average value (AQ)

Max/Min (AQ) On-delay (Ta)

Off-delay (Ta) On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

The value of "MI" can be set and modified in parameter mode.

Description of the function

The hour's counter monitors input En. As long as the status of this input is 1, xLogic calculates the time expired and the time-to-go MN. xLogic displays these times when it is set to configuration mode. The output is set to 1 when the time-to-go is equal to zero.

You reset output Q and the time-to-go counter to the specified value, MI with a signal at input R. The operation hour counter, OT remains unaffected.

You reset output Q and the time-to-go counter to the specified value MI with a signal at input Ral. The operation hour counter, OT is reset to 0.

Depending on your configuration of the Q parameter, the output is either reset with a reset signal at input R or Ral or when the reset signal is 1 or the En signal is 0.

The limit value of OT

The values of the operating hours in OT are retained when you reset the hours counter with a signal at input R. The hours counter OT continues the count as long as En = 1, irrespective of the status at the reset input R. The counter limit of OT is 99999 h. The hour's counter stops when it reaches this value.

Parameter Pre-set

In xLogicsoft, you can define MI and an OT start value.

You can set whether or not the behaviour of Q does or does not depend on En by selecting the corresponding checkbox.

Retentivity with the hour's counter

The hours counter in the xLogic is generally retentive.

However, if the values of the hour's counter are lost after a power failure, then select the respective block in your circuit program. Right mouse click on the hours counter and select Block Properties > Parameters. The option Retentivity must be activated and not changeable (greyed out).

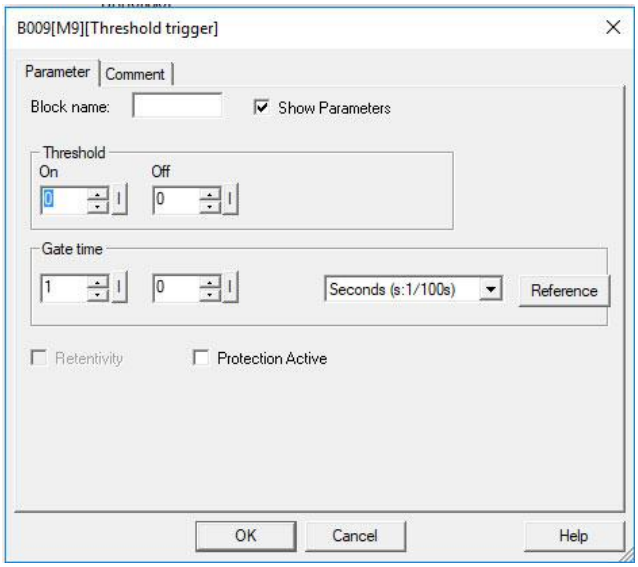
If the Retentivity option is not available, then delete the block and insert a new special function hours counter at the same position.

Threshold Trigger

Description of function

The output is switched on and off, depending on two configurable frequencies.

Connection	Description
Input Fre (—●)	The function counts 0 to 1 transitions at input Fre.1 to 0 transitions are not counted. Use Inputs I5, I6 (14kHz) I7, I8 (60kHz) for ELC-12 CPU, Inputs I9, IA (14kHz) IB, IC (60kHz) for PR-18, ELC-22 and PR-24 CPU Any other input or circuit element can be used for low frequencies (typical 4 Hz).
Parameter	**On**: On threshold The range of values: 0000...9999 **Off**: Off threshold The range of values: 0000...9999 **G_T**: Time interval or gate time during which the input pulses are measured. The range of values: 00:05 s...99:99 s
Output Q	Q is set or reset according to the threshold values.



Parameter

The gate time G_T can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)

Analog threshold trigger (Ax)

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta)

Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

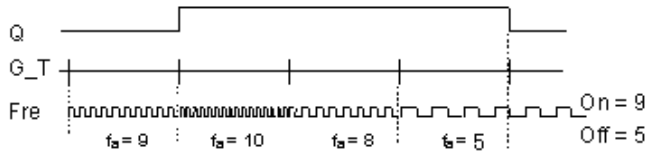
Stairway light switch (Ta)

Multiple function switch (Ta)

Stopwatch (Ta)

The value of "On", "Off" can be set/modified in parameter mode.

Timing diagram



fa = Input frequency

Description of the function

The trigger measures the signals at input Fre. The pulses are captured during a configurable period G_T.

Q is set or reset according to the set thresholds. See the following calculation rule.

Calculation rule

If the threshold (On) > threshold (Off), then:
Q = 1, if fa >= On
Q = 0, if fa < Off.

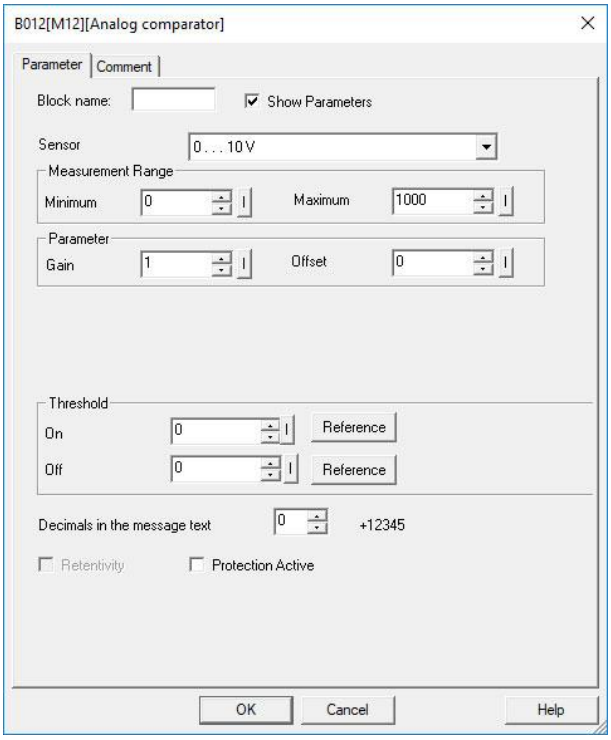
If the threshold (On) < threshold (Off), then Q = 1, if On <= fa < Off.

Connection	Description
Input Ax, Ay	Input the analogue signals of which you want to determine the delta at the inputs Ax and Ay. Use the analogue inputs AI1...AI8, the analogue outputs AQ1 and AQ2. AI1...AI8: 0 - 10 V corresponds with 0 - 1000 (internal value).
Parameter	**A**: Gain Range of values: ± 10.00 **B**: Zero offset The range of values: ± 10,000 **On**: On threshold The range of values: ± 20,000 **Off**: Off threshold The range of values: ± 20,000 **p**: Number of decimals Range of values: 0, 1, 2, 3
Output Q	Q is set or reset depending on the set thresholds.

Analogue Comparator

Description of function

The output is set and reset depending on the difference Ax - Ay and on two configurable thresholds.



Parameter p (number of decimals)

The on threshold On and the off threshold Off can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)

Analog threshold trigger (Ax)

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta)

Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta)

Multiple function switch (Ta)

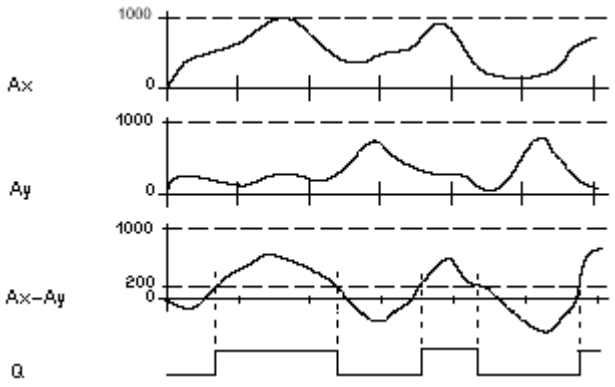
Stopwatch (Ta)

Applies only to Ax, Ay, Delta, On and Off values displayed in a message text.

Does not apply to the comparison of on and off values! (The compare function ignores the decimal point.)

The value of "On", "Off" and "Dec" can be set/modified in parameter mode.

Timing diagram



Q for Ax - Ay > 200, On = Off = 200

Description of function

The function reads the value of the signal at the analogue input Ax.

This value is multiplied by the value of parameter A (gain). Parameter B (offset) is added to the product, hence

(Ax · gain) + offset = Actual value Ax.

(Ay · gain) + offset = Actual value Ay.

Output Q is set or reset depending on the difference of the actual values Ax - Ay and the set thresholds. See the following calculation rule.

Calculation rule

1. If threshold On ≥Threshold Off, then:

Q = 1, if (actual value Ax - actual value Ay) > On

Q = 0, if (actual value Ax - actual value Ay) ≤Off.

2. If threshold On < Threshold Off, then Q = 1, falls:

On ≤ (actual value Ax - actual value Ay) < Off.

Reducing the input sensitivity of the analogue comparator

You can delay the output of the analogue comparator selectively by means of the "on delay" and "off delay" SFBs. By doing so, you determine that output Q is only set if the input trigger length Trg (= output of the analogue comparator) exceeds the defined-on delay time.

This way you can set a virtual hysteresis, which renders the input less sensitive to short changes.

Particular characteristics to be noted when configuring

For help on analogue block parameters, refer to the Analogue value processing section in xLogicsoft.

Setting the Par parameter

The gain and offset parameters are used to adapt the sensors to the relevant application.

Example

In a heating control system, the supply Tv and return line temperatures Tr are to be compared, for example with a sensor at AI2.

A control signal is to be triggered (for example "heater On") when the difference between the supply and return line temperatures is greater than 15 °C. The control signal is reset when the difference is less than 5 °C.

The process variable of the temperature is to be shown in parameter assignment mode.

The thermocouples available have the following technical data: -30 to +70 °C, 0 to 10 VDC.

Application	Internal mapping
-30 °C to 70 °C = 0 VDC to 10 VDC	0 to 1000
0 °C	300 → Offset = -30
Range of values: **-30 °C to 70 °C = 100**	1000 → Gain = 100/1000 = 0.1
On threshold = 15 °C	Threshold = 15
Off threshold = 5 °C	Threshold = 5

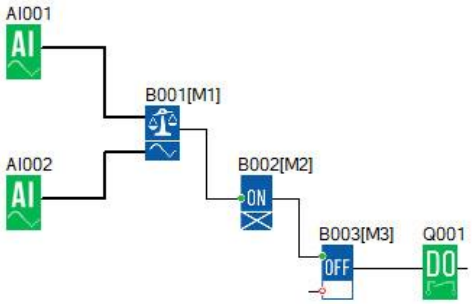
See also topic "Calculating the gain and offset of analogue values

Reducing the input response of the analogue comparator

You can selectively delay the output of an analogue comparator by means of the "On-delay" and "Off-delay" special functions. With on-delay, output Q is only set if the pulse width of the triggering signal at input Trg (=analogue comparator output) is longer than the on-delay time.

Using this method, you will obtain a virtual hysteresis and reduce the input response to short signals.

Function block diagram

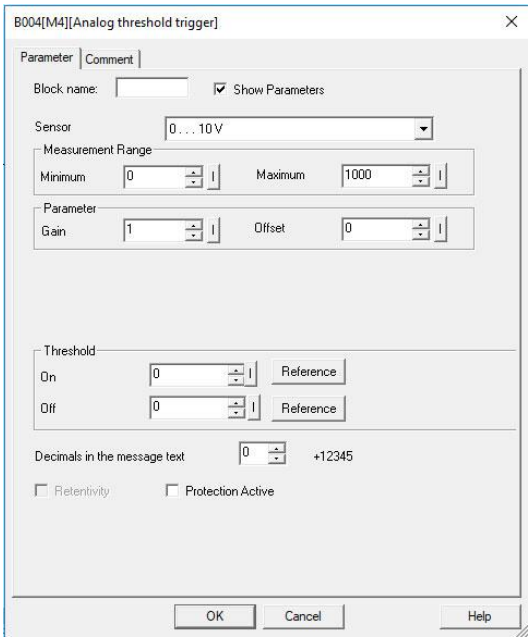


Analogue Threshold Trigger

Description of function

The output is set or reset depending on two configurable thresholds (hysteresis).

Connection	Description
Input Ax	Input the analogue signal to be evaluated at input Ax. Use the analogue inputs AI1...AI8, the analogue outputs AQ1 and AQ2. 0 - 10 V is proportional to 0 - 1000 (internal value).
Parameter	**A**: Gain Range of values: ± 10.00 **B**: Zero offset The range of values: ± 10,000 **On**: On threshold The range of values: ±20,000 **Off**: Off threshold The range of values: ± 20,000 **p**: Number of decimals Range of values: 0, 1, 2, 3
Output Q	Q is set or reset depending on the set thresholds.



Parameter On and Off

The On and Off parameters can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)	Analog threshold trigger (Ax)
Analog amplifier (Ax)	Analog multiplexer (AQ)
Analog ramp (AQ)	Mathematic instruction (AQ)
PI controller (AQ)	Up/down counter (Cnt)
Analog filter (AQ)	Average value (AQ)
Max/Min (AQ)	On-delay (Ta)
Off-delay (Ta)	On-/off-delay (Ta)
Retentive on-delay (Ta)	Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

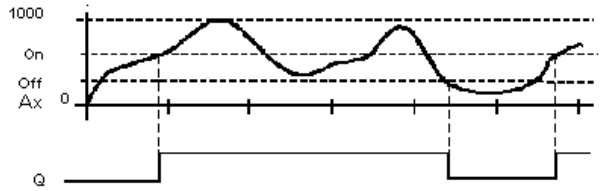
Stopwatch (Ta)

Applies only to the display of On, Off and Ax values in a message text.

Does not apply to the comparison of On and Off values! (The compare function ignores the decimal point.)

The value of "On", "Off" and "Dec" can be set/modified in parameter mode.

Timing diagram



Description of function

The function reads the value of the signal at the analogue input Ax.

This value is multiplied by the value of parameter A (gain). Parameter B (offset) is added to the product, hence

(Ax ·Gain) + Offset = Actual value Ax.

Output Q is set or reset depending on the set threshold values. See the following calculation rule.

Calculation rule

If threshold (On) ≥ threshold (Off), then:

Q = 1, if the actual value Ax > On

Q = 0, if the actual value Ax ≤Off.

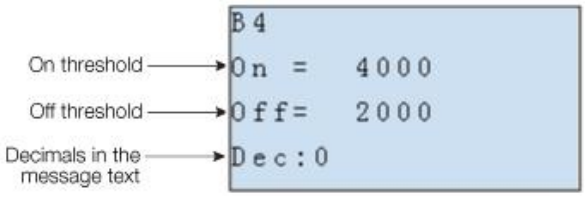
If threshold (On) < threshold (Off), then Q = 1, if

On ≤ the actual value Ax < Off.

NOTICE:

The decimal point setting must be identical in the min. and max. range.

View in parameter assignment mode (example):



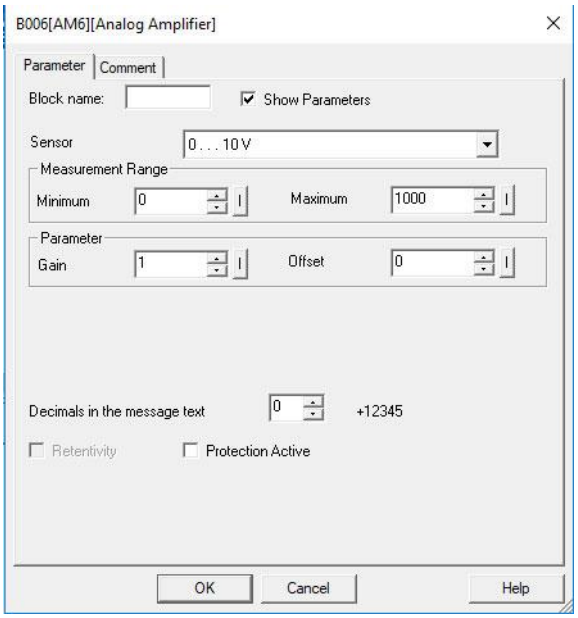
Analogue Amplifier

Description of function

This SFB amplifies an analogue input value and returns it at the analogue output.

Connection	Description
Input Ax	Input the analogue signal to be amplified at input Ax.

	Use the analogue inputs AI1...AI8, the analogue outputs AQ1 and AQ2. AI1...AI8: 0 - 10 V corresponds with 0 - 1000 (internal value).
Parameter	**A**: Gain Range of values: ± 10.00 **B**: Zero offset The range of values: ± 10,000 **p**: Number of decimals Range of values: 0, 1, 2, 3
Output AQ	Analog output Value range for AQ: -32768...+32767



Parameter p (number of decimals)

Applies only to the display of Ax and Ay values in a message text.

Does not apply to the comparison of On and Off values! (The compare function ignores the decimal point.)

Description of function

The function reads the value of an analogue signal at the analogue input Ax.

This value is multiplied by the gain parameter A. Parameter B (offset) is added to the product, i.e.

(Ax ·gain) + offset = Actual value Ax.

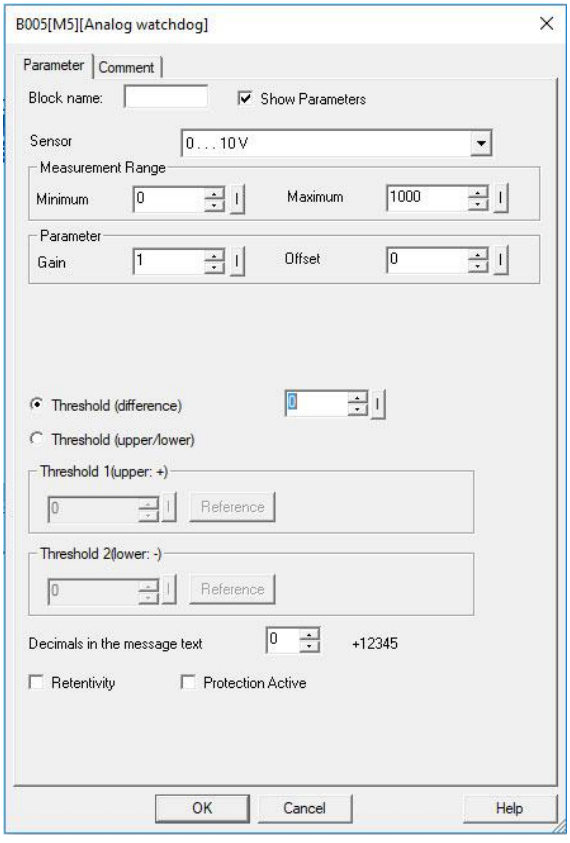
Analogue Watchdog

Description of function

This special function saves the process variable of an analogue input to memory and sets the output when the output variable exceeds or drops below this stored value plus a configurable offset.

Connection	Description
Input En (—▸)	A positive edge (0 to 1 transition) at input En saves the analogue value at input Ax ("Aen") to memory and starts monitoring of the analogue range Aen ± Delta.
Input Ax	You apply the analogue signal to be monitored at input Ax. Use the analogue inputs AI1...AI8, the analogue outputs AQ1 and AQ2. 0 - 10 V is proportional to 0 - 1000 (internal value).
Parameter	**A**: Gain Range of values: ± 10.00 **B**: Zero offset The range of values: ± 10,000 **Delta**: Difference value for the Aen on/off threshold The range of values: ± 20,000

	Δ1: difference value above Aen: on/off threshold Range of values: 0 to 20,000 Δ2: difference value below Aen: on/off threshold Range of values: 0 to 20,000 **p**: Number of decimals Range of values: 0, 1, 2, 3
Output Q	Q is set/reset, depending on the stored analogue value and the offset.



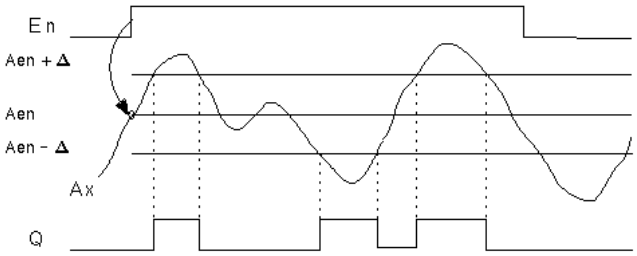
Parameter p (number of decimals)

The two threshold parameters Threshold 1 and Threshold 2 can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) (Ax)	Analog threshold trigger
Analog amplifier (Ax)	Analog multiplexer (AQ)
Analog ramp (AQ) (AQ)	Mathematic instruction
PI controller (AQ)	Up/down counter (Cnt)
Analog filter (AQ)	Average value (AQ)
Max/Min (AQ)	On-delay (Ta)
Off-delay (Ta)	On-/off-delay (Ta)
Retentive on-delay (Ta)	Wiping relay(pulse output) (Ta)
Edge triggered wiping relay (Ta)	
Asynchronous pulse generator (Ta)	
Stairway light switch (Ta)	Multiple function switch (Ta)
Stopwatch (Ta)	

Applies only to the display of Aen, Ax and Delta values in a message text.

Timing diagram



Description of function

A 0 to 1 transition at input En saves the value of the signal at the analogue input Ax. This saved process variable is referred to as "Aen".

Both the analogue actual values Ax and Aen are multiplied by the value of parameter A (gain), and parameter B (offset) is then added to the product, i.e.

(Ax · gain) + offset = Actual value Aen, when input En changes from 0 to 1, or

(Ax · gain) + offset = Actual value Ax.

Output Q is set when the signal at input En = 1 and if the actual value of input Ax is out of range of Aen ± Delta.

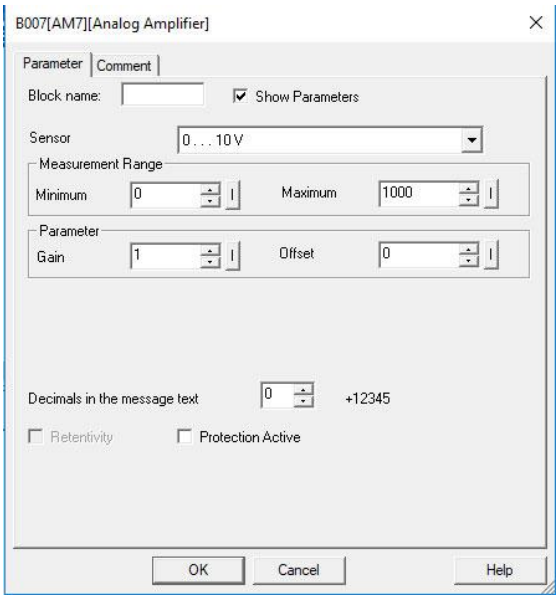
Output Q is reset, when the actual value of input Ax lies within the range of Aen +- Delta, or when the signal at input En changes to lo.

Analogue Differential Trigger

Description of function

The output is set and reset depending on a configurable threshold and a differential value.

Connection	Description
Input Ax	You apply the analogue signal to be analysed at input Ax. Use the analogue inputs AI1...AI8, the analogue outputs AQ1 and AQ2. 0 - 10 V is proportional to 0 - 1000 (internal value).
Parameter	**A**: Gain Range of values: ± 10.00 **B**: Zero offset The range of values: ± 10,000 **On**: On threshold The range of values: ±20,000 **Delta**: Differential value for calculating the off parameter The range of values: ± 20,000 **p**: Number of decimals Range of values: 0, 1, 2, 3
Output Q	Q is set or reset, depending on the threshold and difference values.

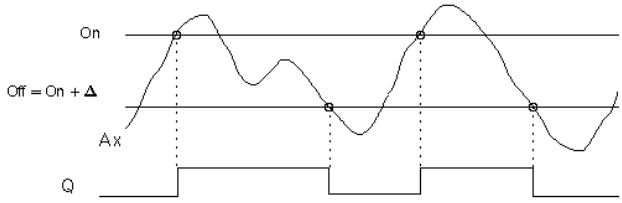


Parameter p (number of decimals)

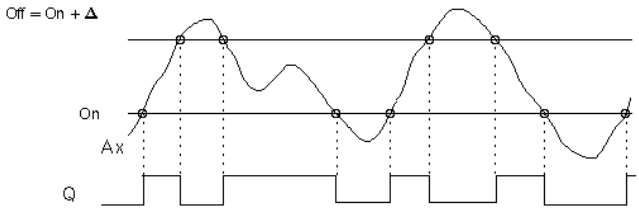
Applies only to the display of On, Off and Ax values in a message text.

Does not apply to the comparison of On and Off values! (The compare function ignores the decimal point.)

Timing diagram A: Function with a negative difference Delta



Timing diagram B: Function with positive difference Delta



Description of the function

The function fetches the analogue signal at input Ax.

Ax is multiplied by the value of the A (gain) parameter, and parameter B (offset) is added to the product, i.e.

(Ax · gain) + offset = actual value of Ax.

Output Q is set or reset, depending on the set (On) threshold and difference value (Delta). The function automatically calculates the Off parameter: Off = On + Delta, whereby Delta may be positive or negative. See the calculation rule below.

Calculation rule

When you set a negative differential value Delta and the On threshold >= Off threshold, then:

Q = 1, if the actual value Ax > On

Q = 0, if the actual value Ax ≤ Off.

See the timing diagram A.

When you set a positive differential value Delta and the On threshold < the Off threshold, then

Q = 1, if:

On threshold, ≤ the actual value Ax < Off threshold.

See the timing diagram B.

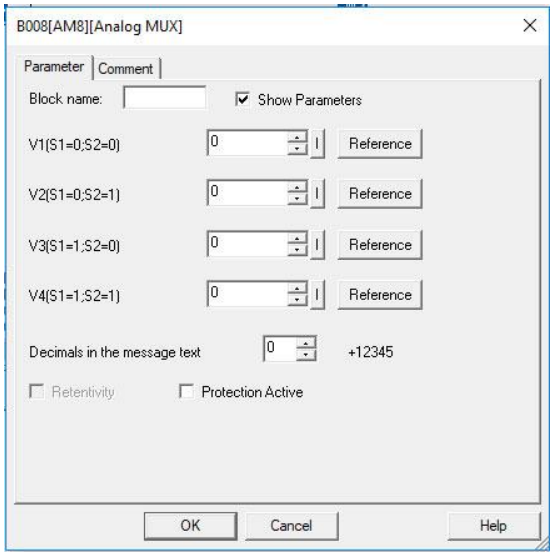
Analogue Multiplexer

Description of function

This special function displays 0 or one of 4 saved analogue values on the analogue output.

Connection	Description
Input En (⟶)	1 on input En (Enable) switches, dependent on S1 and S2, a parameterized analogue value to the output AQ. 0 on input EN switches 0 to the output AQ.
Input S1 and S2	S1 and S2 (selectors) for selecting the analogue value to be issued. S1 = 0 and S2 = 0: The value 1 is issued S1 = 0 and S2 = 1: The value 2 is issued S1 = 0 and S2 = 1: The value 2 is issued S1 = 1 and S2 = 0: The value 3 is issued

	S1 = 1 and S2 = 1: The value 4 is issued
Parameter	**V1...V4**: Analog values (Value) that will be issued. Value range: -32768...+32767 **p**: Number of decimal places value range: 0, 1, 2, 3
Output AQ	Analog output Value range for AQ: -32768...+32767



Parameters V1...V4

The values for V1...V4 can be provided by the value of another already-programmed function such as:

Analog comparator (Ax – Ay)

Analog threshold trigger (Ax)

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta)

Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

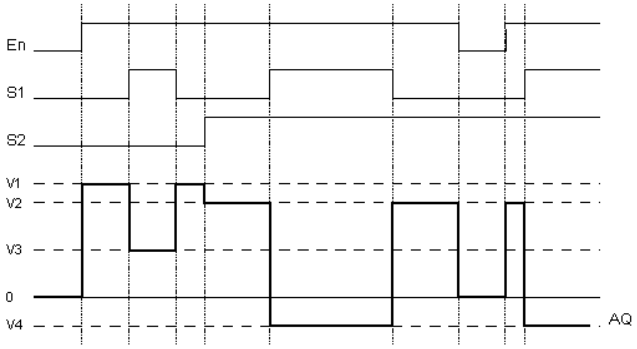
Asynchronous pulse generator (Ta)

Stairway light switch (Ta)

Multiple function switch (Ta)

Stopwatch (Ta)

Timing Diagram



Description of Function

If input En is set, then the function issues one of 4 possible analogue values V1 to V4 at the output AQ, depending on the parameters S1 and S2.

If the input En is not set, then the function issues the analogue value 0 at output AQ.

Analogue output

If you interconnect this special function with a real analogue output, note that the analogue output can only process values between 0 and 1000. To do this, you may

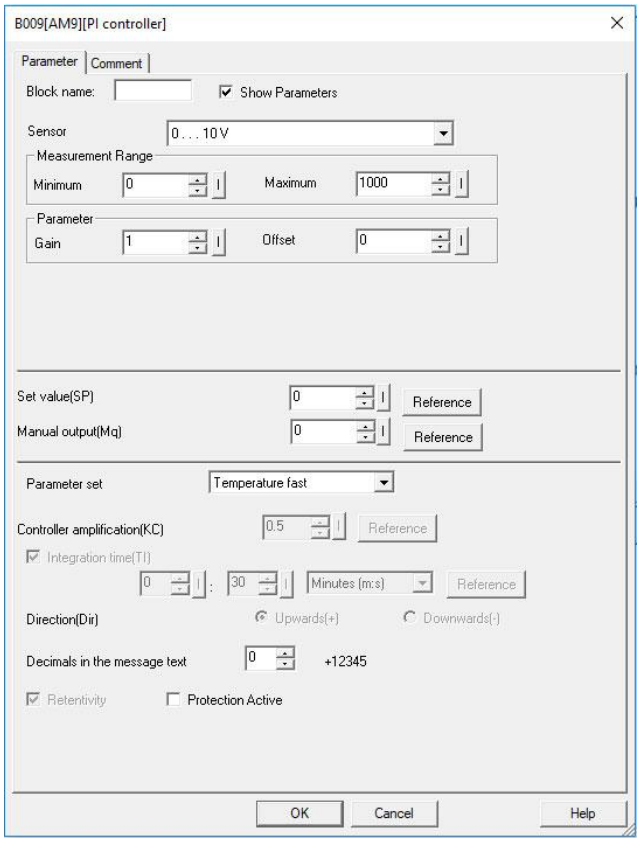
need to connect an additional analogue amplifier block between the analogue output of the special function and the real analogue output. Using this amplifier, you standardize the output range of the special function to a value range of 0 to 1000.

PI Controller

Description of function

It is for proportional-action and integral-action controllers. You can use both types of controller individually or combined.

Connection	Description
Input A/M	Set the mode of the controller: 1: automatic mode 0: manual mode
Input R (—○)	Use the input R to reset the output AQ. As long as this input is set, the input A/M is disabled. The output AQ is set to 0.
Input PV	Analogue value: process value, Influences the Output
Parameter	**Sensor**: Type of sensor being used **Min.**: Minimum value for PV value range: -10,000 to +20,000 **Max.**: Maximum value for PV value range: -10,000 to +20,000 **A**: Gain Value range: +- 10.00 **B**: Offset Value range: +- 10,000 **SP**: Set-value assignment value range: -10,000 to +20,000 **Mq**: Value from AQ with manual mode. Value range: 0 to 1,000 **Parameter sets**: application-related pre-sets for KC, TI and Dir (see below) **KC**: Gain value range: 00.00 to 99.99 **TI**: Integral time value range 00:01 min to 99:59 min **Dir**: Action direction of the controller value range: + or - **p**: Number of decimal places value range: 0, 1, 2, 3
Output AQ	Analog output (manipulated variable) Value range for AQ: 0 to 1,000

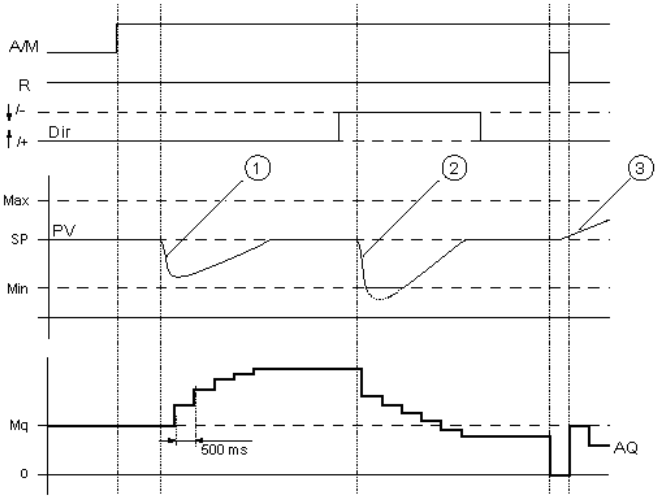


Parameter P (number of decimal places)

Only applies to portray the values from PV, SP, Min. and Max. in a message text.

Timing Diagram

Nature, manner and speed with which the AQ changes depend on the parameters KC and TI. Thus, the course of AQ in the diagram is merely an example. A control action is continuous; therefore, the diagram portrays just an extract.



1.	**A disturbance causes the PV to drop, as Dir is positioned upwards, AQ increases until PV corresponds again to SP.**
2.	A disturbance causes the PV to drop, as Dir is positioned downwards, AQ decreases until PV corresponds again to SP. Dir is coordinated with the basic conduct of a control loop. The direction (dir.) cannot be changed during the term of the function. The change in Dir here is shown for the purposes of clarification.
3.	As AQ is set to 0 by means of the input R, PV changes. This is based on the fact that PV increases, which on account of Dir = upwards causes AQ to drop.

Description of Function

If the input A/M is set to 0, then the special function issues output AQ with the value that you set with parameter Mq.

If the input A/M is set to 1, then automatic mode commences. As an integral sum the value Mq is adopted, the controller function begins the calculations in accordance with the formulas given in Control and

regulated basics. The updated value PV is calculated in the formulas.

Updated value PV = (PV * gain) + offset

If the updated value PV = SP, then the special function does not change the value of AQ.

Dir = upwards/+ (timing diagram numbers 1 and 3)

If the updated value PV > SP, then the special function reduces the value of AQ.

If the updated value PV < SP, then the special function increases the value of AQ.

Dir = downwards/- (timing diagram number 2)

If the updated value PV > SP, then the special function increases the value of AQ.

If the updated value PV < SP, then the special function reduces the value of AQ.

With a disturbance, AQ continues to increase/decrease until the updated value PV again corresponds to SP. The speed with which AQ changes depends on the parameters KC and TI. If the input PV exceeds the parameter Max., then the updated value PV is set to the value of Max. If the PV falls short of the parameter Min., then the updated value PV is set to the value of Min.

If the input R is set to 1, then the AQ output is reset. As long as R is set, the input A/M is disabled.

Sampling Time

The sampling time is fixed at 500 ms.

Parameter sets

In order to simplify the use of the PI controller, the parameters for KC, TI and Dir are predefined for the following applications:

Parameter set	Application example	Parameter KC	Parameter TI (s)	Parameter Dir
Temperature fast	Temperature, cooling control of small spaces; small volumes	0,5	30	+
Temperature slow	Heating, ventilation, temperature, cooling control of large spaces; large volumes	1,0	120	+
Pressure 1	Quick pressure change, compressor control	3,0	5	+
Pressure 2	Slow pressure change, differential pressure control (flow controller)	1,2	12	+
Full level 1	Vat and/or reservoir filling without drain	1,0	1	+
Full level 2	Vat and/or reservoir filling with drain	0,7	20	+

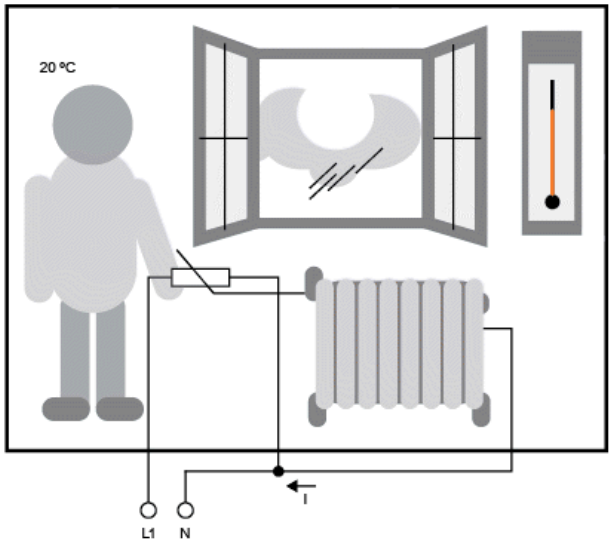
Control and regulated

In engineering, quantities can be both controlled and regulated.

When controlling, a quantity is manipulated without being able to compensate for outside influences. When regulating, a quantity is maintained at a specific value in order to compensate for outside influences.

In the following example, controlling means that the person can set the heat output at a fixed value. The heater cannot compensate for the drop-in room temperature when a window is opened.

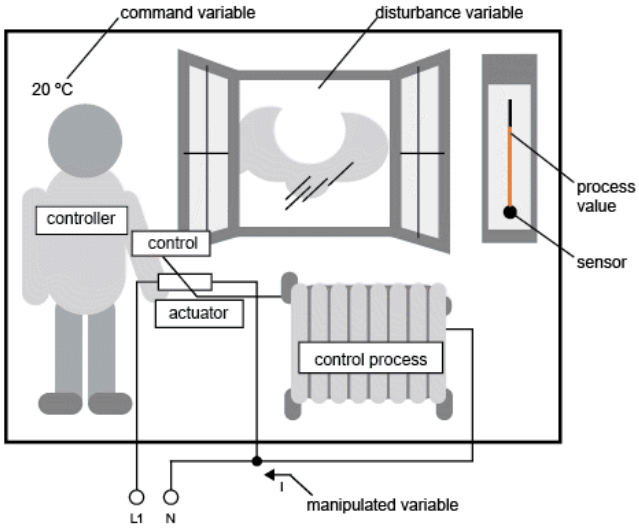
In the example below, regulating means that the person can increase the heat output if the room temperature drops to below 20 °C. If the room temperature rises above 20 °C, the heat output is reduced.



Basic concepts of regulating

In the example, the current for the electric heating is the manipulated variable. The changeable resistance is the actuator. The hand that operates the actuator is the control. The actual room temperature is the controlled variable or the process value. The desired room temperature is the command variable or the setpoint value. The electric heating is the control process. The thermometer is the sensor. The temperature loss from opening the window is the disturbance variable.

So, this means that the person measures the process value (room temperature) with the sensor (thermometer), compares the process value (room temperature) with the command variable (desired room temperature) and uses the actuator (changeable resistance) to manually regulate the manipulated variable (heating current), in order to compensate for the disturbance variable (temperature drop from opening the window). The person is, therefore, the controller.

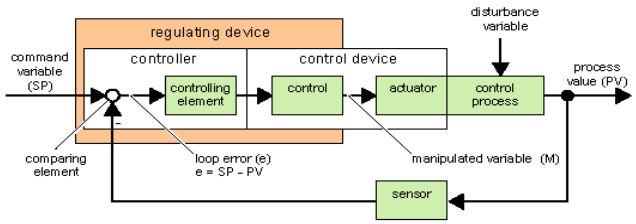


The control device is formed from the actuator and the control.

The control and controller together form the regulating device.

The following picture gives an abstract portrayal of the situation described above.

The comparing element uses the sensor to compare the command variable with the process value. If the command variables and process value deviate from one another, this results in a positive or negative loop error that in turn changes the process value.



Control loop

The process value x influences the manipulated variable M by means of the regulating device. This creates a closed circuit that is also known as a control loop.

If in the example above, the window is opened, the temperature in the room drops. The person must increase

the heat output of the heater. If the heat output is increased too much, it will get too hot. The person must then reduce the heat output.

If the heat output is increased or reduced too quickly, then the control loop starts to sway. The room temperature fluctuates. It is either too hot or too cold. To prevent this, the person must carefully and slowly reduce or increase the heat output.

Loop error

The loop error is the difference between the command variable and the process value. In other words: the deviation of a process value from a set value.

e = SP – PV

The loop error 'e' brings about a change to the manipulated variable M.

The example above illustrates this very well: if the desired temperature is 20 °C (= command value w), the room temperature is 22 °C (= process value PV), this results in the loop error:

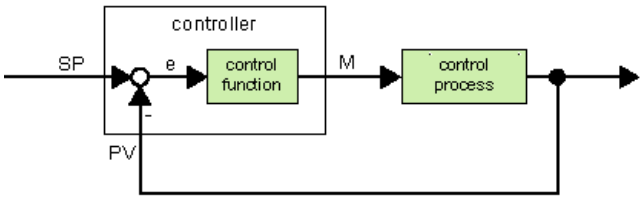
e = SP – PV = 20 °C - 22 °C = -2 °C

In this case, the negative sign indicates a reversing action: the heat output is reduced.

In a control loop's state of equilibrium, the loop error is zero or very small. If the command variable changes or there is a disturbance, a loop error arises. The loop error is corrected by means of the manipulated variable M.

Controller basics

A controller can be simply portrayed as follows:



The comparing element and the controller function describe the conduct of the controller.

The following describes the most important types of controller. A controller's step response tells us a lot about its conduct. The step response describes how a controller reacts to the erratic change in the process value.

There are 3 important basic types of controller:

Proportional-action controller (P controller)

Integral-action controller (I controller)

Differential-action controller (D controller – we're not touching on this here)

These are combined for a real controller. For instance, the PI controller:

P Controller

A proportional-action controller (P controller) changes the manipulated variable M proportional to the loop error. The P controller works immediately. By itself, it cannot drive the loop error to zero.

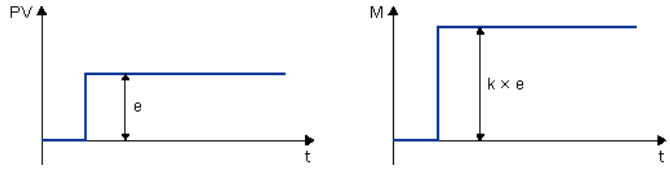
$M_{Pn} = k_P \times e_n$

M_{Pn} : Manipulated variable of the P controller at the time n

k_P: Gain of the P controller

e_n : Loop error at the time n

The following picture shows a jump in process value and step response of the controller:



Summary

The P controller has the following characteristics:

It cannot correct faults with the control process > lasting loop error.

It reacts immediately to a change in the process value.

It is stable.

I Controller

An integral-action controller (I controller) changes the manipulated variable M proportional to the loop error and to the time. The I controller works by delayed action. It completely remedies a loop error.

In order to calculate the value of the manipulated variable at a period of time n, the time up until this period of time must be divided into small time slices. The loop errors at the end of each time slice must be added up (integrated) and they are then entered in the calculation.

$$M_{In} = k_I \times (T_S / T_I) \times (e_n + e_{n-1} + e_{n-2} + e_{n-3} + \dots + e_0) = k_I \times (T_S / T_I) \times e_n + M_{In-1}$$

M_{In}: Manipulated variable of the I controller at the time n

M_{In-1}: Manipulated variable of the I controller at the time n-1; also called the integral sum

k_I: Gain of the I controller

T_S: Sampling time, duration of a time slice

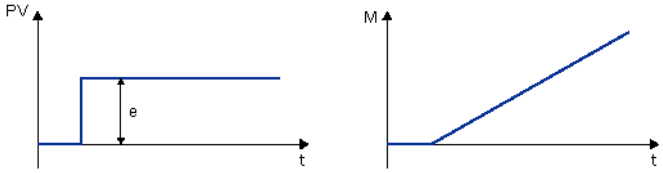
T_I: Integral time: by means of this time, the influence of the integral part is controlled by the manipulated variable, also known as integral-action time

e_n: Loop error at the time n

e_{n-1}: Loop error at the time n-1; etc.

e_0: Loop error at the beginning of the calculations

The following picture shows a jump in process value and step response of the controller:



Summary

The PI controller has the following characteristics:

It sets the process value exactly to the command variable.

By so doing, it tends to oscillate and is unstable.

It requires more time to carry out the control action than the P controller.

PI controller

A PI controller reduces the loop error immediately and will eventually drive the loop error to zero.

$$M_n = M_{Pn} + M_{In} = k_P \times e_n + k_I \times (T_S / T_I) \times e_n + M_{In-1}$$

M_n: Manipulated variable at the time n

M_{Pn}: Proportional part of the manipulated variable

M_{In}: Integral part of the manipulated variable

M_{In-1}: Manipulated variable of the I controller at the time n-1; also called the integral sum

k_P: Gain of the P controller

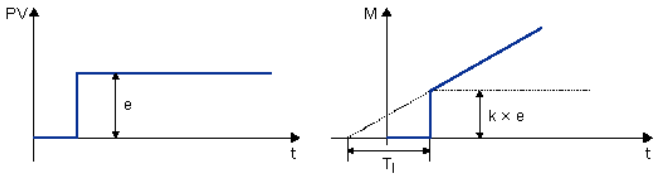
k_I: Gain of the I controller

T_S: Sampling time, duration of a time slice

T_I: Integral time; by means of this time the influence of the integral part is controlled by the manipulated variable, also known as the integral-action time

e_n: Loop error at the time n

The following picture shows a jump in process value and step response of the controller:



Summary

The PI controller has the following characteristics:

The P controller components quickly intercept an occurring loop error.

The I controller components can then remedy the remaining loop error.

The controller components supplement each other so that the PI controller works quickly and precisely.

Description of the individual parameters

Controller parameters	Portrayed in xLogic	Possible value range in the xLogic
Mn Manipulated variable at the time n	The output of the PI controller block	0 to 1,000
kP The gain of the P part **kI The gain of the I part**	In the xLogic, the parameter KC applies as an increase for the I part and the P part of the controller equally. Should you enter KC=0, then the P part of the controller switches off. In this special case, k is automatically set to 1 for the I part. If KC = 0: kP = 0 and kI = 1 If KC <> 0: kP = kI = KC	0.00 to 99.99
Ts Sampling time, duration of a time slice	Fixed	500 ms
TI Integral time	Parameter TI, if you set this parameter to 99:59 min, then you switch off the I part of the controller.	00:01 min to 99.59 min
en Loop error at the time n; generally, applies e = SP – PV	Refer to SP and PV	"
SP	The parameter SP is the set-value assignment w. For this parameter, you can use the analogue output of a different special function.	-10,000 to +20,000
PV	PV is the process value x and is calculated as follows:	"

	PV = (analog value on input * gain) + offset. You can connect the input for example by means of an analogue input with a PT100 sensor.	
	The gain parameter influences PV	0.0 to 10.0
	The offset parameter influences PV	-10,000 to +20,000
	PV is restricted by the parameters Min. and Max.	In each case: -10,000 to +20,000
	The Dir parameter gives the action direction of the controller. Positive means: If the set value > process value then the process value is increased; if the set value < process value then the process value is reduced. Negative means: If the set value > process value then the process value is reduced; if the set value < process value then the process value is increased. e.g. heat regulation: if the set value is greater than the process value (room is too cold), the manipulated variable increases the process value.	- or +

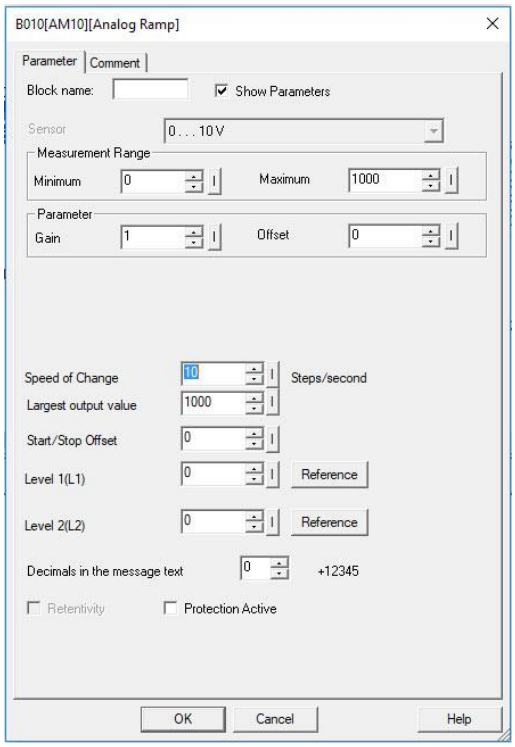
Analogue Ramp

Description of function

The Analogue Ramp instruction allows the output to be changed from the current level to a selected level at a specified rate.

Connection	Description
Input En (—▷)	A change in the status from 0 to 1 at input EN (Enable) applies the start/stop level (Offset "B" + StSp) to the output for 100 ms and starts the ramp operation to the selected level. A change in the status from 1 to 0 immediately sets the current level to Offset "B", which makes output AQ equal to 0.
Input Sel	SeI = 0: The step 1 (level 1) is selected. SeI = 1: The step 2 (level 2) is selected. A change in the status of Sel causes the current level to start changing to the selected level at the specified rate.
Input St	A change in the status from 0 to 1 at input St (Decelerated Stop) causes the current level to decrease at a constant rate until the start/stop level (Offset "B" + StSp) is reached. The start/stop level is maintained for 100 ms and then the current level is set to Offset "B", which makes output AQ equal to 0.
Parameter	**Level1** and **Level2**: Levels to be reached; value range for each level: -10,000 to +20,000 **MaxL**: a Maximum value that must not be exceeded. Value range: -10,000 to +20,000 **StSp**: Start/Stop offset value that is added to Offset "B" to create the start/stop level. If the Start/Stop offset is 0, then the start/stop level is Offset "B"). Value range: 0 to +20,000

	Rate: Speed with which level 1, level 2 or 0ffset is reached. Steps/seconds are issued. Value range: 1 to 10,000 **A**: Gain Value range: 0 to 10,00 **B**: Offset Value range: +- 10.000 **p**: Number of decimal places Value range: 0, 1, 2, 3
Output AQ	The output AQ is scaled using the formula: (Current Level - Offset "B") / Gain "A" Note: When AQ is displayed in parameter mode or message mode, it is displayed as an un-scaled value (engineering units: current level). The value range for AQ: 0...+32767



Parameter p (number of decimal places)

The level parameters Level1 and Level2 can be provided by the value of another already-programmed function:

Analog comparator (Ax – Ay)　Analog threshold trigger (Ax)

Analog amplifier (Ax)　Analog multiplexer (AQ)

Analog ramp (AQ)　Mathematic instruction (AQ)

PI controller (AQ)　Up/down counter (Cnt)

Analog filter (AQ)　Average value (AQ)

Max/Min (AQ)　On-delay (Ta)

Off-delay (Ta)　On-/off-delay (Ta)

Retentive on-delay (Ta)　Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

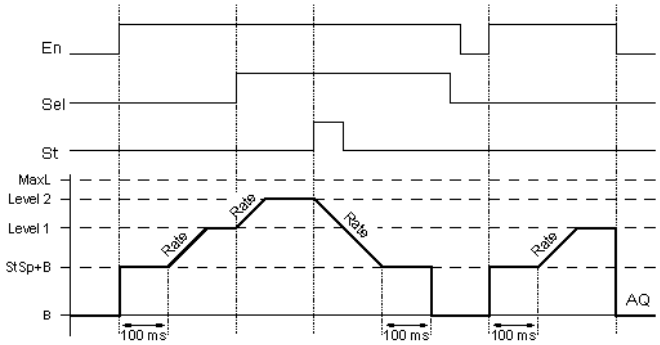
Asynchronous pulse generator (Ta)

Stairway light switch (Ta)　Multiple function switch (Ta)

Stopwatch (Ta)

Parameter p only applies for displaying the values of AQ, level 1, level 2, MaxL, StSp, and Rate in a message text.

A timing diagram for AQ



Description of function

If the input En is set, then the function sets the value StSp + Offset "B" for 100 ms.

Then, depending on the connection of Sel, the function runs from the level StSp + Offset "B" to either level 1 or level 2 at the acceleration set in Rate.

If the input St is set, the function runs to a level of StSp + B at the acceleration set in Rate. Then the function holds the level at StSp + Offset "B" for 100 ms. After 100 ms, the level is set to Offset "B". output AQ. The scaled value (output AQ) is 0.

If the input St is set, the function can only be restarted once the inputs St and En have been reset.

If input Sel has been changed, depending on the connection of Sel, the function runs from the current target level to the new target level at the rate that is specified.

If the input En is reset, the function immediately sets the current level to Offset "B".

The current level is updated every 100 ms. Note the relationship between output AQ and the current level:

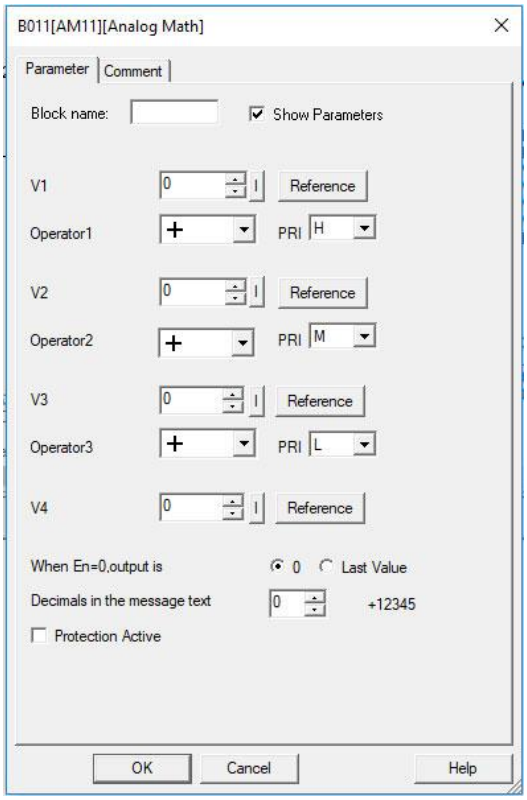
Output AQ = (current level – Offset "B") / Gain "A"

Analogue Math

Description of function

The analogue math block calculates the value AQ of an equation formed from the user-defined operands and operators.

Connection	Description
Input En (—▷)	Enable the analogue math function block.
Parameter	**V1**: Value 1: First operand **V2**: Value 2: Second operand **V3**: Value 3: Third operand **V4**: Value 4: Forth operand **Operator 1**: First operator **Operator 2**: Second operator **Operator 3**: Third operator **Priority 1**: Priority of the first operation **Priority 2**: Priority of the second operation **Priority 3**: Priority of the third operation **P**: number of decimals Range of values: 0,1,2,3
Output AQ	The output AQ is the result of the equation formed from the operand values and operators. AQ will be set to 32767 if a divide by 0 or overflow occurs, and -32768 if a negative overflow (underflow) occurs.



Parameter p (number of decimals)

The values V1, V2, V3, and V4 can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay) Analog threshold trigger (Ax)

Analog amplifier (Ax) Analog multiplexer (AQ)

Analog ramp (AQ) Mathematic instruction (AQ)

PI controller (AQ) Up/down counter (Cnt)

Analog filter (AQ) Average value (AQ)

Max/Min (AQ) On-delay (Ta)

Off-delay (Ta) On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

Parameter p applies to the display of V1, V2, V3, V4, and AQ in a message text.

Description of the function

The analogue math function combines the four operands and three operators to form an equation. The operator can be any one of the four standard operators: +, -, *, or /. For each operator, you must set a unique priority of High ("H"), Medium ("M"), or Low ("L"). The high operation will be performed first, followed by the medium operation, and then by the low operation. You must have exactly one operation of each priority. The operand values can reference another previously defined function to provide the value. The analogue math function rounds the result to the nearest integer value.

The number of operand values is fixed at four and the number of operators is fixed at 3. If you need to use fewer operands, use constructions such as " + 0" or " * 1" to fill the remaining parameters.

You can also configure the behaviour of the function when the Enable parameter "En"=0. The function block can either retain its last value or be set to 0.

Possible errors: Zero division and overflow

If the analogue math function block execution results in zero division or overflow, it sets internal bits that indicate the type of error that occurred. You can program an analogue math error detection function block in your circuit program to detect these errors, and to control the program behaviour as needed. You program one analogue math error detection function block to reference one specific analogue math function block.

Examples

The following tables show some simple example analogue math block parameters, and the resulting equations and output values:

V1	Operator 1	V2	Operator 2	V3	Operator 3	V4
12	+(M)	6	/(H)	3	-(L)	1

Equation: (12 + (6 / 3)) - 1

Result: 13

V1	Operator 1	V2	Operator 2	V3	Operator 3	V4
2	+(L)	3	*(M)	1	+(H)	4

Equation: 2+ (3*(1+4))

Result: 17

V1	Operator 1	V2	Operator 2	V3	Operator 3	V4
100	-(H)	25	/(L)	2	+(M)	1

Equation: (100 – 25) / (2 + 1)

Result: 25

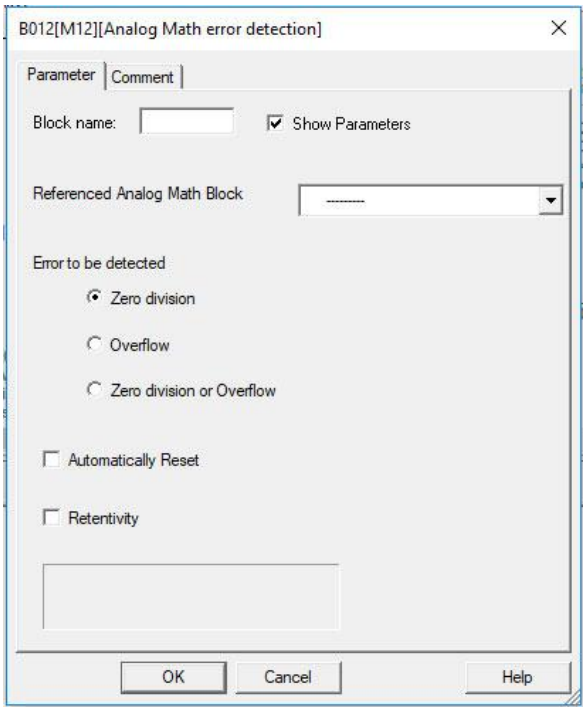
Long Data Math!!

Analogue Math Error Detection

Description of function

The analogue math error detection block sets an output if an error has occurred in the referenced analogue math function block.

Connection	Description
Input En (—▸)	Enable the analogue math error detection function block.
Input R (—○)	Reset the output
Parameter	**Referenced FB**: block number of an analogue math instruction **Error to detect**: Zero division, Overflow, or Zero division OR Overflow. **Auto Reset**: Reset the output when the failure condition clears.
Output Q	Q is set high if the error to detect occurred in the last execution of the referenced analogue math function block.



Parameter Referenced FB

The value for the Referenced FB parameter references the block number of an already-programmed analogue math function block.

Description of the function

The analogue math error detection block sets the output when the referenced analogue math function block has an error. You can program the function to set the output on a zero-division error, an overflow error, or when either type of error occurs.

If you select the automatically reset checkbox, the output is reset prior to the next execution of the function block. If not, the output retains its state until the analogue math error detection block is reset with the R parameter.

In any scanning cycle, if the referenced analogue math function block executes before the analogue math error detection function block, the error is detected in the same scan cycle. if the referenced analogue math function block executes after the analogue math error detection function block, the error is detected in the next scan cycle.

Analogue math error detection logic table

In the table below, Error to Detect represents the parameter of the analogue math error detection instruction that selects which type of error to detect. Zero represents the zero division bit set by the analogue math instruction at the end of its execution: 1 if the error occurred, 0 if not. OF represents the overflow bit set by the analogue math instruction: 1 if the error occurred, 0 if not. Zero division OR Overflow represents the logical OR of the zero-division bit and the overflow bit of the referenced analogue math instruction. Output (Q) represents the output of the analogue math error detection function. An "x" indicates that the bit can be either 0 or 1 with no influence on the output.

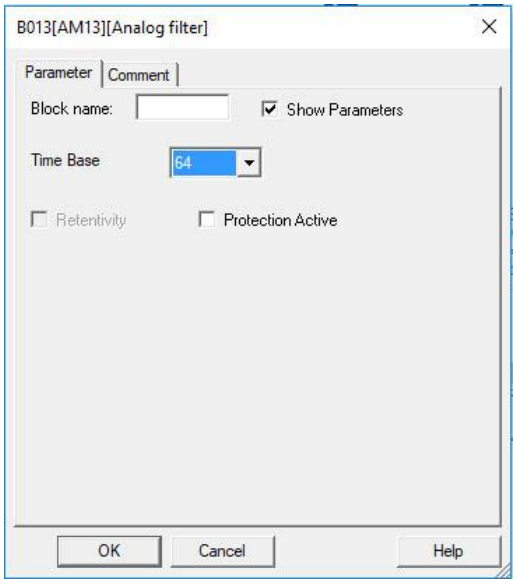
Error to Detect	Zero	Of	Output
Zero division	1	X	1
Zero division	0	X	0
Overflow	X	1	1
Overflow	X	0	0
Zero division OR Overflow	1	0	1
Zero division OR Overflow	0	1	1
Zero division OR Overflow	1	1	1
Zero division OR Overflow	0	0	0

If the Referenced Analog Math FB is null, then the output is always 0.

Analogue Filter

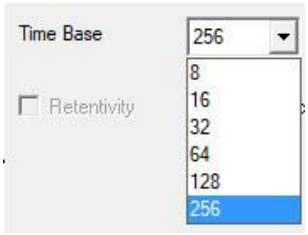
Description of function

Connection	Description
Input Ax	Analogue Inputs Analogue Outputs Analogue Flags The block number of a function with analogue output
Parameter	**Sn** (Number of samples): determines how many analogue values are sampled within the program cycles that are determined by the set number of samples. xLogic samples an analogue value within every program cycle. The number of program cycles is equal to the set number of samples. Possible settings: 8, 16, 32, 64, 128, 256
Output AQ	AQ outputs an average value of the analogue input Ax over the current number of samples, and it is set or reset depending on the analogue input and the number of samples. * Analog inputs: 0 to 10 V corresponds with 0 to 1000 (internal value).



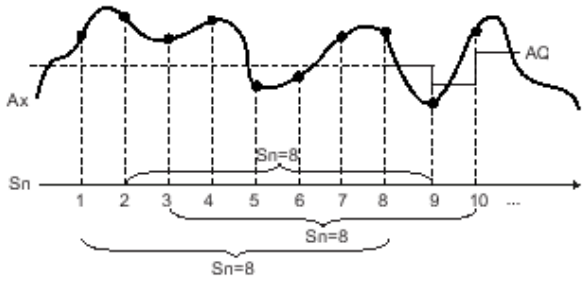
Parameter

You can set the number of samples to the following values:



After you set the parameter, the analogue filter calculates the average value of the samples and assigns this value to AQ.

Timing diagram



Description of function

The function outputs the average value after sampling the analogue input signal according to the set number of samples. This SFB can reduce the error of the analogue input signal.

NOTICE:

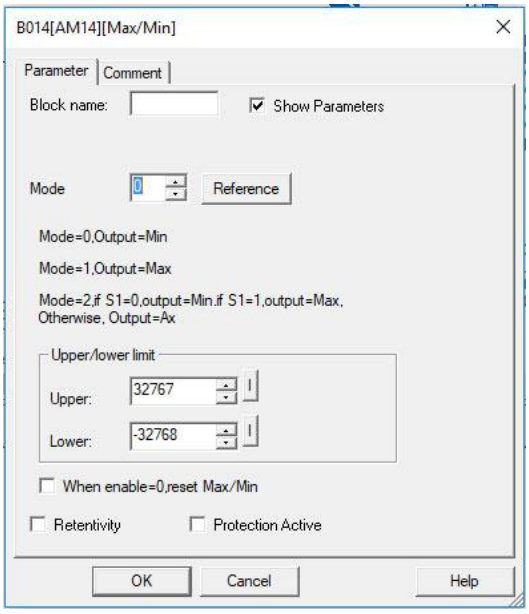
There is a maximum of eight analogue filter function blocks available for use in the circuit program in xLogicsoft.

Max/Min

Description of function

Connection	Description
Input En (—▸)	The function of input En (Enable) depends on the settings of parameter Mode and the selection of checkbox "when En = 0, reset Max/Min".
Input S1	This input is enabled when you set Mode =2: A positive transition (0 to 1) at input S1 sets the output AQ to the maximum value. A negative transition (1 to 0) at input S1 sets the output AQ to the minimum value.
Input Ax	Input Ax is one of the following analogue signals: Analog Inputs Analog Outputs Analogue flags Block number of a function with analogue output
Parameter	**Mode** Possible settings: 0, 1, 2, 3 Mode = 0: AQ = Min Mode = 1: AQ = Max Mode = 2 and S1= 0 (low): AQ = Min Mode = 2 and S1= 1 (high): AQ = Max

	Mode = 3 or a block value is referenced: AQ = Ax
Output AQ	AQ outputs a minimum, maximum, or actual value depending on the inputs, or is reset to 0 is configured to do so when the function is disabled Analog inputs: 0 to 10 V corresponds with 0 to 1000 (internal value).



Parameter Mode

You can set the values for parameter Mode based on the actual values of another already-programmed function:

Analog comparator (Ax – Ay)

Analog threshold trigger (Ax)

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta)

Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

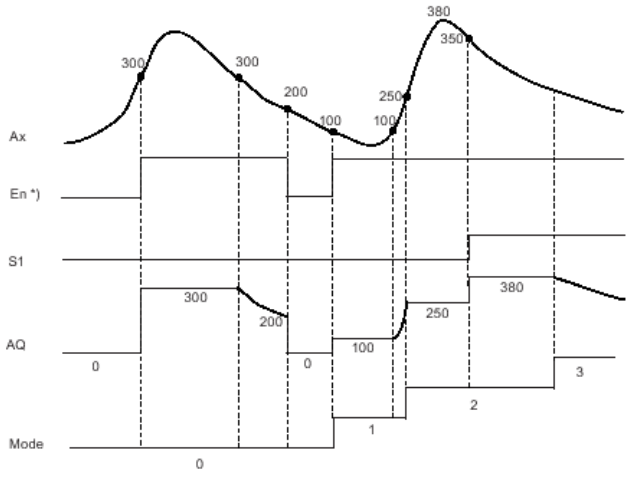
Stairway light switch (Ta)

Multiple function switch (Ta)

Stopwatch (Ta)

You can select the required function by the block number.

Timing diagram



Description of the function

If you select the check box "when En = 0, reset Max/Min":

En = 0: The function sets the AQ value to 0.

En = 1: The function outputs a value at AQ, depending on the settings of Mode and S1.

If you do not select the checkbox "when En = 0, reset Max/Min":

En = 0: The function holds the value of AQ at the current value.

En = 1: The function outputs a value at AQ, depending on the settings of Mode and S1.

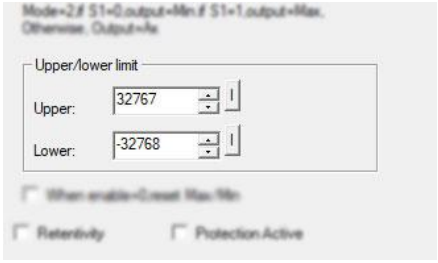
Mode = 0: The function sets AQ to the minimum value

Mode = 1: The function sets AQ to the maximum value

Mode = 2 and S1 = 0: The function sets AQ to the minimum value

Mode = 2 and S1 = 1: The function sets AQ to the maximum value

Mode = 3 or a block value is referenced: The function outputs actual analogue input value.



In the dialogue box of Max/Min block, there is an upper/lower limit setting, when the block output the AQ value is less than the lower value, the AQ shall be equal to the Lower value; While the block output the AQ value is more than the upper value, the AQ shall be equal to the upper value.

If someone wants to use the upper/lower limitation for other function blocks. such upper/lower limit function can be used, then this block can be referenced as other blocks parameters when programming.

Here is an example:

Someone wants to use the panel key to change the on-delay parameters for 1s—10s in the text message block, if the value which user set exceeds such range, then it will crush the machine, hence we must add the upper/lower limitation in the program to avoid such trouble.

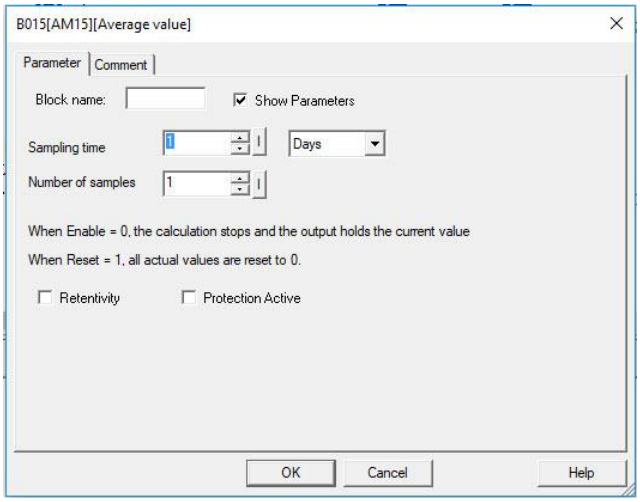
Max/Min block upper/lower function

Average Value

Description of function

The average value function samples the analogue input signal during the configured time period and outputs the average value at AQ.

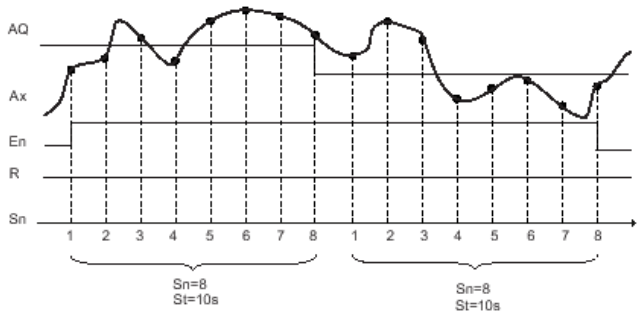
Connection	Description
Input En (—▷)	A positive edge (0 to 1 transition) at input En (Enable) sets the output AQ to the average value of input Ax after the configured time. A negative edge (1 to 0 transition) holds the output at its last calculated value.
Input R (—○)	A positive edge (0 to 1 transition) at input R (Reset) resets the output AQ to 0.
Parameter	**St** (Sampling time): You can set it to Seconds, Days, Hours or Minutes. The range of values: If St = Seconds: 1 to 59 If St = Days: 1 to 365 If St = Hours: 1 to 23 If St = Minutes: 1 to 59 **Sn** (Number of samples): The range of values: If St = Seconds: 1 to St*100 If St = Days: 1 to 32767 If St = Hours: 1 to 32767
	If St = Minutes and St ⩽ 5 minutes: 1 to St*6000 If St = Minutes and St ⩾ 6 minutes: 1 to 32767
Output AQ	AQ outputs the average value over the specified time of sampling. * Analog Inputs: 0 to 10 V corresponds with 0 to 1000 (internal value).



Parameter St and Sn

Parameter St represents the sampling time and parameter Sn represents the number of samples.

Timing diagram



Description of the function

When En = 1, the average value function calculates the average value of the samples during the configured time interval. At the end of the sampling time, this function sets output AQ to this calculated average value.

When En = 0, the calculation stops, and AQ retains the last calculated value. When R = 0, AQ is reset to 0.

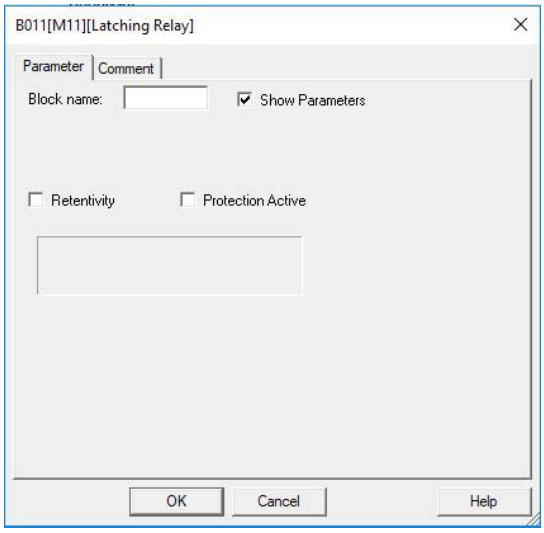
Latching Relay

Description of function

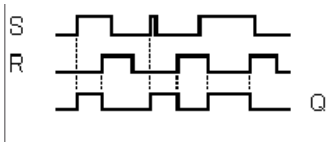
A signal at input S sets output Q. A signal at input R resets output Q.

Connection	Description
Input S (⟶)	Set output Q with a signal at input S (Set).
Input R (⟶○)	Reset output Q with a signal at input R (Reset). Output Q is reset if S and R are both set (reset has priority over set).
Parameter	**Retentivity** set (on) = the status is retentive in memory.
Output Q	Q is set with a signal at input S and remains set until it is reset with a signal at input R.

No parameter of a Latching relay can be set/modified in runtime mode.



Timing diagram



Description of the function

The latching relay represents a simple binary memory logic. The output value depends on the input states and the previous status at the output.

Logic table of the latching relay:

S	R	Q	Remark
0	0	x	Status unchanged
0	1	0	Reset
1	0	1	Set
1	1	0	Reset

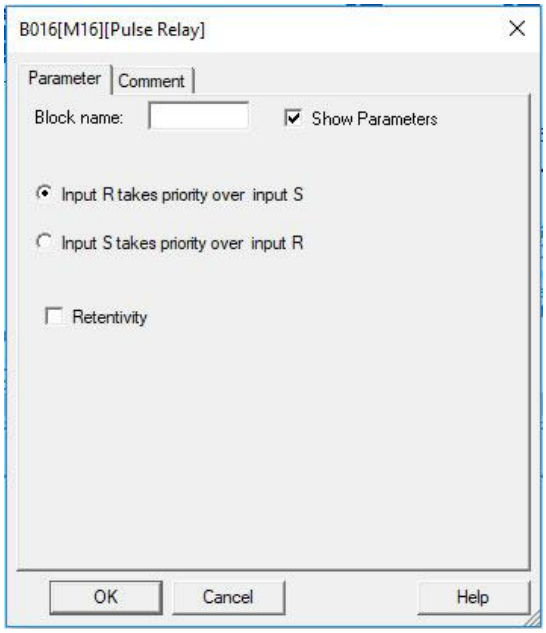
When retentivity is enabled, the output signal corresponds to the signal status prior to the power failure.

Pulse Relay

Description of function

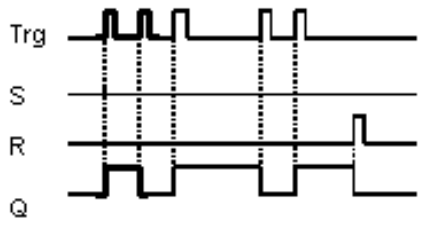
The output is set and reset with a short one-shot at the inputs S and R respectively.

Connection	Description
Trg input (—●)	You switch output Q on or off with a signal at input Trg (Trigger) input.
Input S (—▸)	A one-shot at input S (Set) sets the output to logical 1.
Input R (—○)	A one-shot at input R (Reset) resets the output to logical 0
Parameter	**Selection**: RS (input R priority), or SR (input S priority) **Retentivity** set (on) = the status is retentive in memory.
Output Q	Q is switched on with a signal at Trg and is reset again at the next Trg pulse if both S and R = 0.



No parameter of Latching relay can be set/modified in runtime mode.

Timing diagram



Description of the function

The status of output Q changes with each 0 to 1 transition at input Trg and if both S and R = 0, i.e. the output is switched on or off.

Input Trg does not influence the SFB when S = 1 or R = 1.

A one-shot at input S sets the pulse relay, i.e. the output is set to logical 1.

A one-shot at input R resets the pulse relay to its initial state, i.e. the output is set to logical 0.

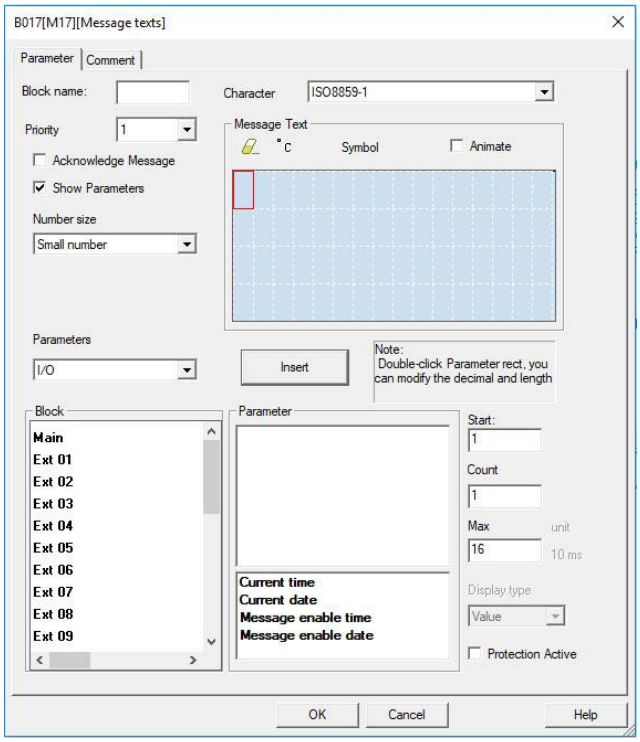
Either the input R takes priority over input S (i.e. the signal at input S has no effect as long as R = 1), or the input S takes priority over input R (i.e. the signal at input R has no effect as long as S = 1), depending on your configuration.

Message Text

Description of function

Display message texts and parameters of other blocks in RUN mode.

Connection	Description
Input En (⟶)	A 0 to 1 transition at En (Enable) triggers the output of the message text.
Input P	P is the priority of the message text. 1 is the lowest, 32 is the highest priority. Quit: Acknowledgement of the message text
Parameter	**Text**: Input of the message text **Par**: Parameter or actual value of another, already configured function (see "Visible parameters or actual values") **Time**: Shows the continuously updated time-of-day **Date**: Shows the continuously updated date **EnTime**: Shows the time of the 0 to 1 transition **EnDate**: Shows the date of the 0 to 1
Output Q	Q remains set as long as the message text is queued.



Description of the function

With a 0 to 1 transition of the signal at input En, the display outputs your configured message text (actual value, text, TOD, date) in RUN mode.

Acknowledgement disabled (Ack = Off):

The message text is hidden with a 0 to 1 signal transition at input En.

Acknowledgement enabled (Ack = On):

After input En is reset to 0, the message text is displayed until acknowledged by pressing the OK button. The message text cannot be acknowledged as long as input En is high.

If several message text functions were triggered with En=1, the message with the highest priority (1 = lowest, 64=highest) is displayed. This also implies that a new message text is only displayed if its priority is higher than that of previously enabled message texts.

After a message text is disabled or acknowledged, the function automatically shows the previously active message text with the next highest priority.

You can press the ▲ and ▼ keys to step through multiple active message texts.

Example

This is how two message texts could be shown:

Display field of xLogic in RUN mode

Input P configuration

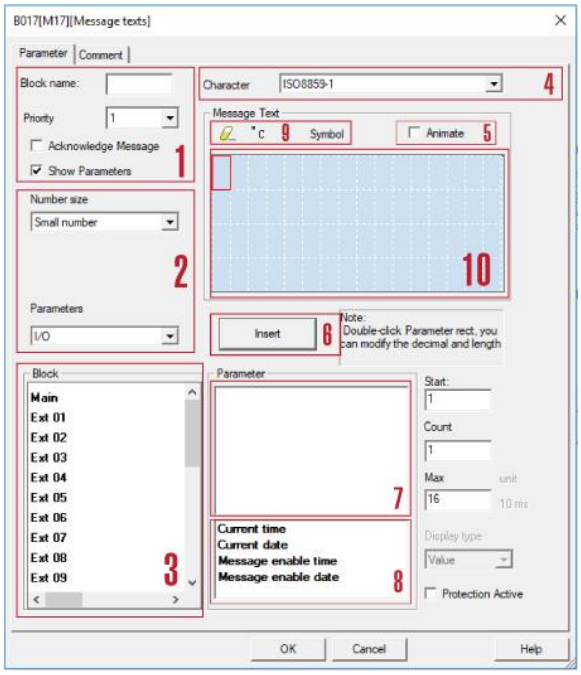
From the input P, you configure the following characteristics of the message text:

- Priority
- Acknowledgement
- Message destination

Restrictions

Up to 128 message text functions are available for PR-12, PR-18/ELC-22/PR-24 and PR-26 CPUs.

Particular characteristics to be noted when configuring



1	**"General" area** **Here you will find the following settings: ·** • **Priority of the message text** • **Check box for message text acknowledgement**
2	Kinds of parameters and the number size selection area IO status, analogue IO values analogue flags and blocks can be inserted into LCD for displaying.
3	"Blocks" area Shows a list of all the circuit program blocks and their parameters.
4	Language selection
5	Animate option This option is used for display animation on the ELC-43TS. See user manual of ELC-43TS for further details.
6	"Insert" button Button for inserting a parameter selected from the "Block parameters" area.
	"Block parameters" or "General parameters" area into the message text.
7	"Block parameters" area Shows the parameters of a block selected from the "Blocks" area which you can output in the message text.
8	"General parameters" area Shows general parameters such as the current date.
9	"Delete" button Button for deleting entries from the "Messages" area "Special characters" button Button for inserting special characters in the "Messages" area
10	"Messages" area You arrange the message text in this area. Information entered in this area corresponds with that on the xLogic display.

To arrange the message text

From the "Blocks" area, select the block whose parameters you want to output.

Drag and drop the parameters required from the "Block parameters" to the "Messages" area. You may also use the "Insert" button to do so.

In the "Messages" area, you can add parameter data as required.

Particular characteristics to be noted when configuring

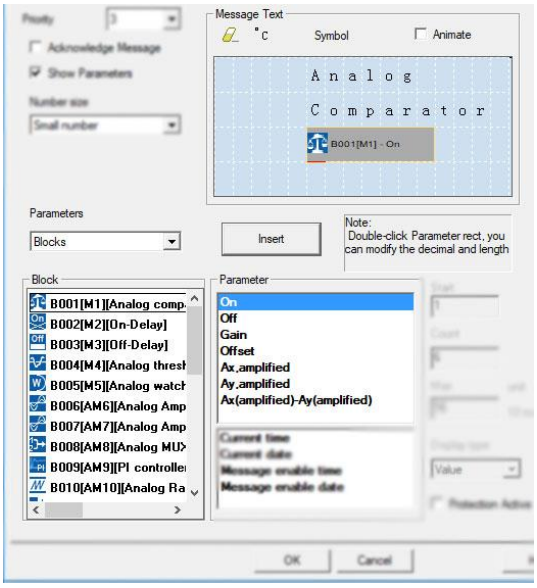
The message text can be configured in the block properties dialogue. You can enter up to 4 lines for each message text (the text display of the xLogic is 4 x 16 characters) and set the priority. You can move to the next

line using the cursor keys or the mouse. Hit the [ENTER] key to confirm all your entries in the block properties dialogue and to close the dialogue.

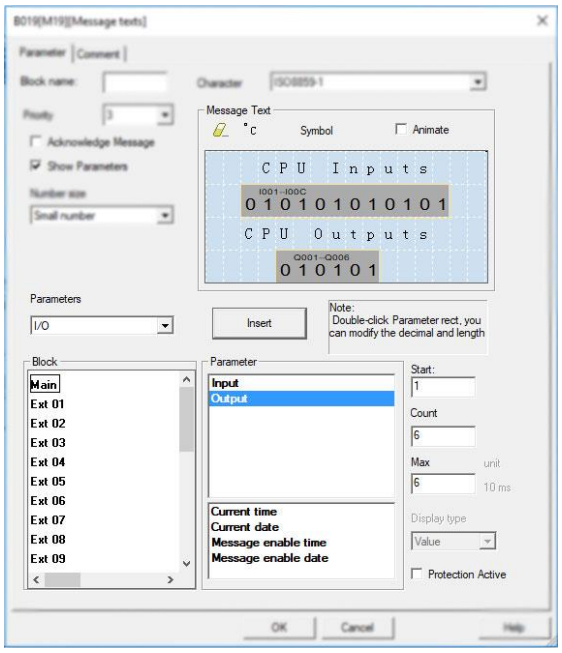
You may also enter the actual values of other blocks in the text lines. To do so, select the relevant block from the Block dialogue. A Parameter dialogue opens to display a list of all parameters available for the selected block. The block parameter you select in this dialogue is written to the selected text line. The actual parameter value is now included when you call the message text.

Set the "Acknowledge message" attribute to specify whether a message is being acknowledged before it is closed.

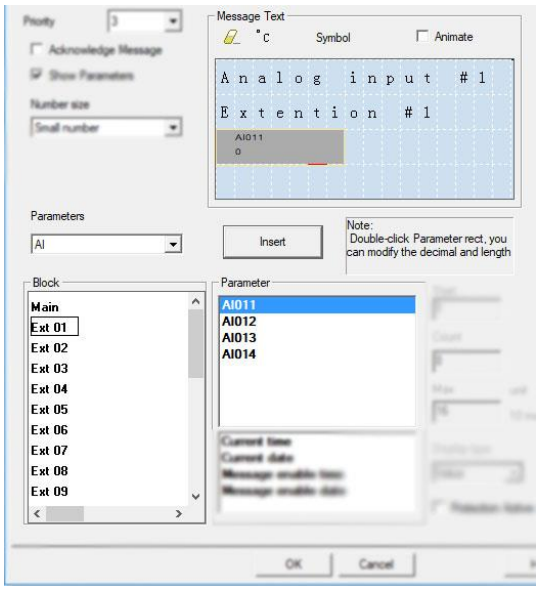
B. Blocks



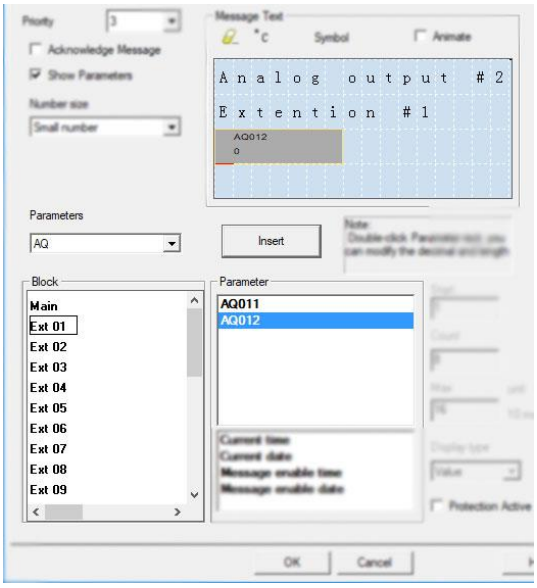
A. I/O status of CPU and extensions



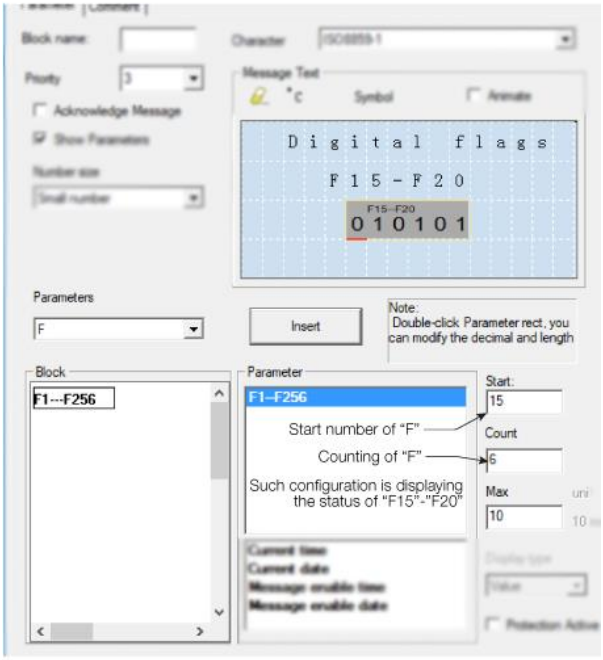
C. Analog input value of CPU and extensions



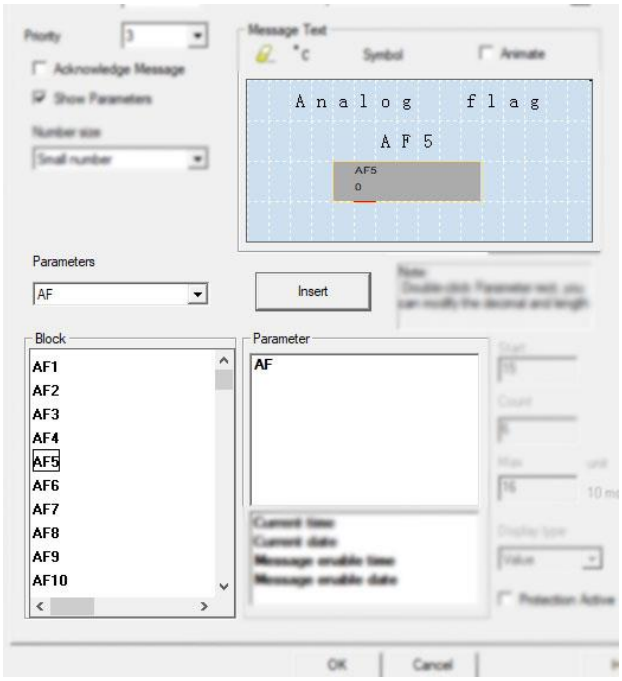
D. The analogue output value of CPU and extensions



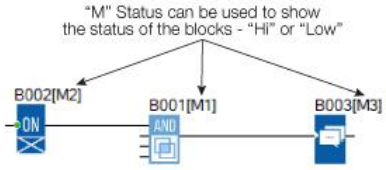
E. F (digital flag) status

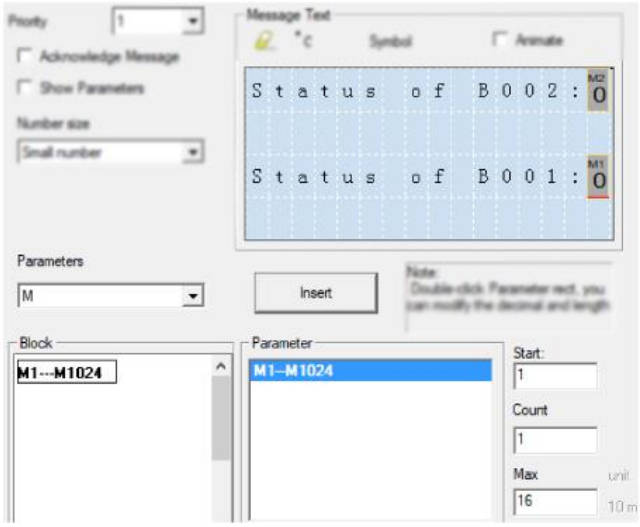


F. AF (analogue flag) value

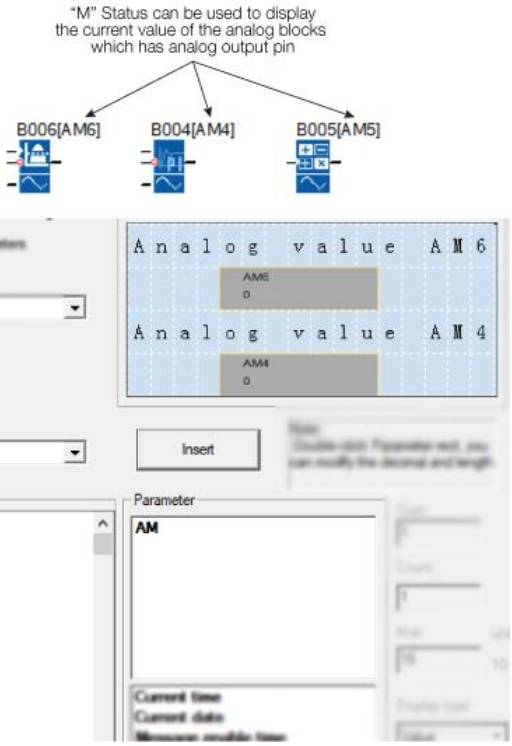


G. M status



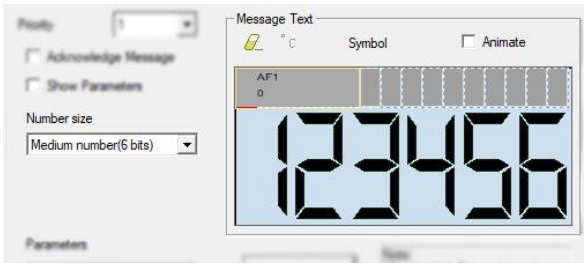


H. AM value

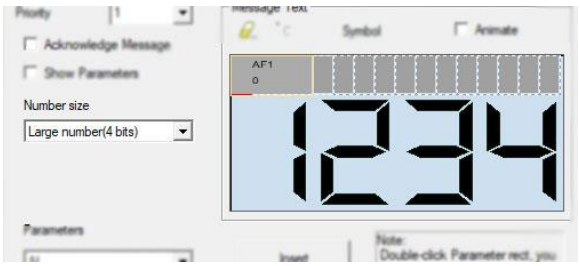


With a text message, the text can be displayed as a large number (6 bits or 4 bits). This feature can only be used in PR-12, PR-18, ELC-22, PR-24 and PR-26 Series CPUs.

6-bit

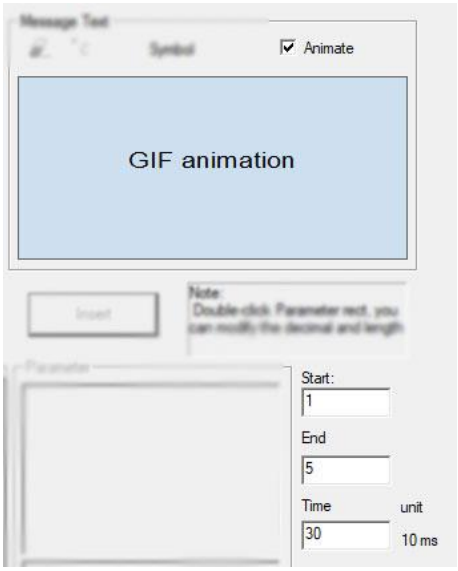


4-bit



Animate option

This option is for the ELC-43TS touch screen, it is used to trigger the pictures saved in the screen, please refer to the ELC-43TS User Guide for detailed information.

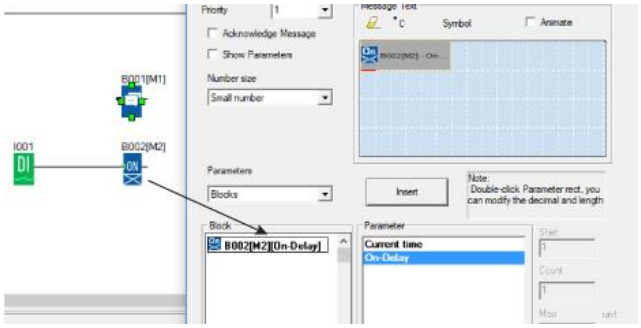


How to change the parameters of blocks in a displayed message?

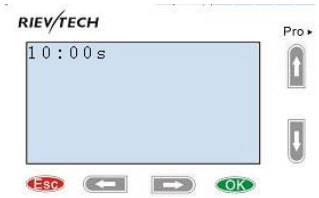
To change the parameters of a displayed message, press "OK" key for 3 seconds.

Steps to Follow:

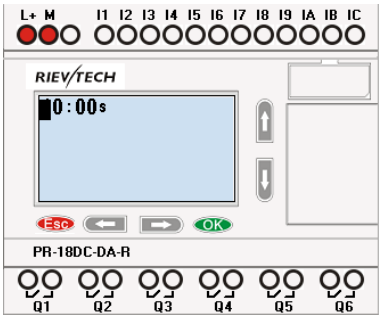
A. Insert the parameters of the block into the message text. (Here is an On-delay).



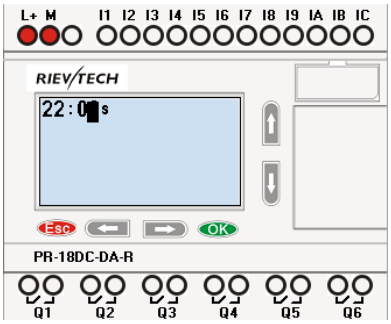
B. Change the page to display the message text contents using the arrow keys.



C. Press "OK" for 3 seconds to enter into edit mode.



D. Modify the value by pressing the arrow keys and confirm with the OK key.



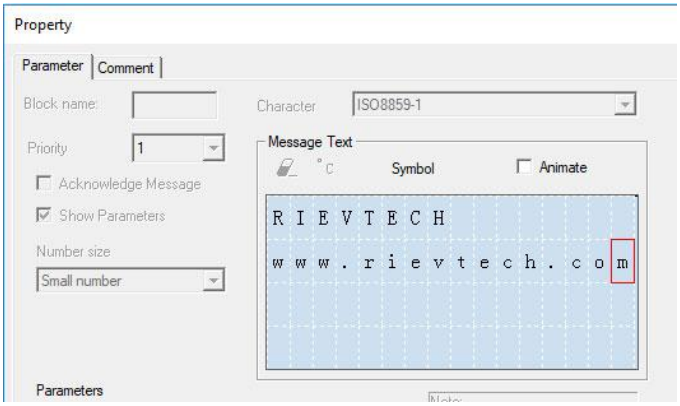
CPU Cover

This block cannot be found in the block list as it is set as default by the system of xLogic. To change the CPU Cover text, use your mouse and click "Tools" menu->select "Edit Cover HMI" in xLogicsoft.

Description of function

Display the status (Run or Stop) of xLogic when power-on or simulation by software.

Particular characteristics to be noted when configuring



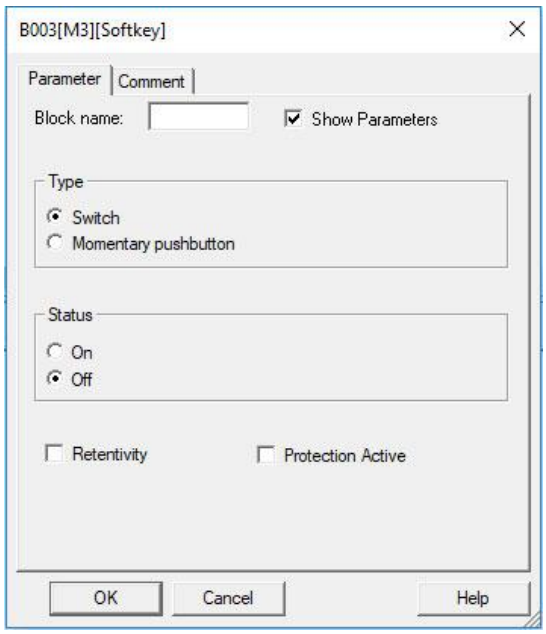
On the PLC's screen:

Softkey Block

Description of function

This SFB provides the action of a mechanical pushbutton or switch.

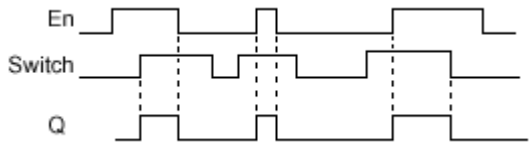
Connection	Description
Input En (⟶)	Output Q is set with a 0 to 1 signal transition at input En (Enable) and if, in addition, 'Status=On' has been confirmed in configuration mode.
Parameter	**Type**: Sets either a pushbutton action for one cycle or a switching action of the function. **Status**: On or Off state that is applied in the initial cycle after program start-up, is retentivity is not set. **Retentivity** set (on) = the status is retentive in memory.
Output Q	Output Q remains set 1, as long as En=1 and the status at the parameter Type = Switch and Status = On. Output Q is set for the duration of one cycle if EN=1 and the status at the parameters Type = momentary (pushbutton) and Status = On.



Factory state

The default of 'Type' is 'momentary action switch'.

Timing diagram



Description of the function

The output is set, when input En is set and the 'Status' parameter is set to 'On' and confirmed with OK. This action is performed irrespective of a configured switch or pushbutton function.

The output is reset to '0' in the following three cases:

1. With a 1 to 0 signal transition at input En.
2. When a pushbutton function is configured, and one cycle has expired after its actuation.
3. When the 'Status' parameter sets the 'Off' status in configuration mode, and this has been confirmed with OK.

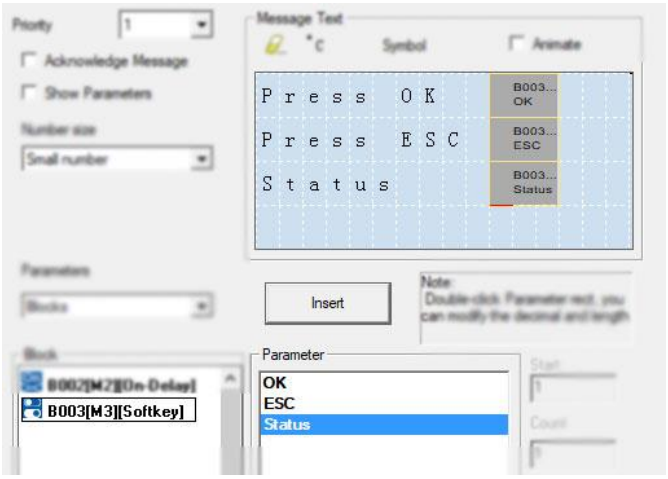
Particular characteristics to be noted when configuring

The softkey can be used both with momentary push button or switching action. At the status parameter, you can define the on (actuated) or off state for the switch/push button.

If the softkey is assigned a push button action, the output is always set for the duration of one cycle with a 0 to 1 transition at input En when the push button is in on state, or if the push button state changes from Off to On when En=1.

Parameters of softkey in message texts block:

1) The "ok" means you can press the ok button to switch the status of the softkey block after its parameter inserted into message texts.
2) The "Esc" means you can press the ESC button to switch the status of the softkey block after its parameter inserted into message texts.
3) The status of softkey block is also can be inserted into the message texts block for displaying.

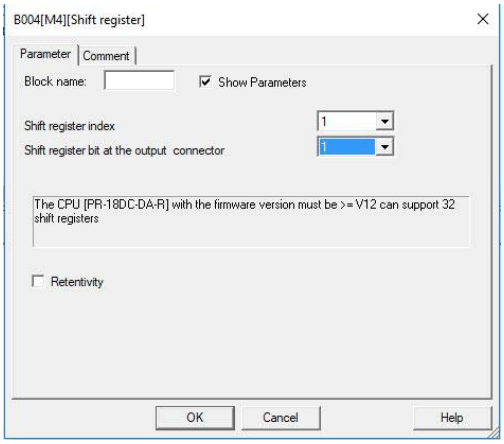


Shift Register

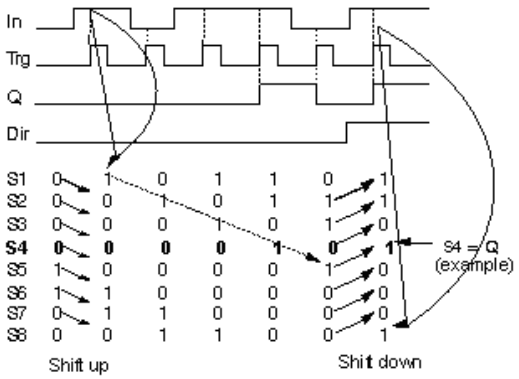
Description of function

The shift register function can be used to read an input value and to shift the bits. The output value corresponds to the configured shift register bit. The shift direction can be changed at a special input.

Connection	Description
Input In	The function when started reads this input value.
Trg input (—●)	The SFB is started with a positive edge (0 t 1 transition) at input Trg (Trigger). A 1 to 0 transition is irrelevant.
Input Dir	You define the shift direction of the shift register bits S1...S8 at the Dir input: Dir = 0: shift up (S1 >> S8) Dir = 1: shift down (S8 >> S1)
Parameter	**Shift register bit** that determines the value of output Q. Possible settings: S1 ... S8 **Retentivity** set (on) = the status is retentive in memory.
Output Q	The output value corresponds to the configured shift register bit.



Timing diagram



Setting the Par parameter

This special function is not available in parameter assignment mode.

Description of function

The function reads the value of input In with a positive edge (0 to 1 transition) at input Trg (Trigger).

This value is written to shift register bits S1 or S8, depending on the set shift direction:

- Shift up: S1 accepts the value of input In; the previous value of S1 is shifted to S2, S2 is shifted to S3, etc.
- Shift down: S8 accepts the value of input In; the previous value of S8 is shifted to S7, S7 is shifted to S6, etc.

Q outputs the value of the configured shift register bits.

If retentivity is not enabled, the shift function restarts at S1 or S8 after a power failure.

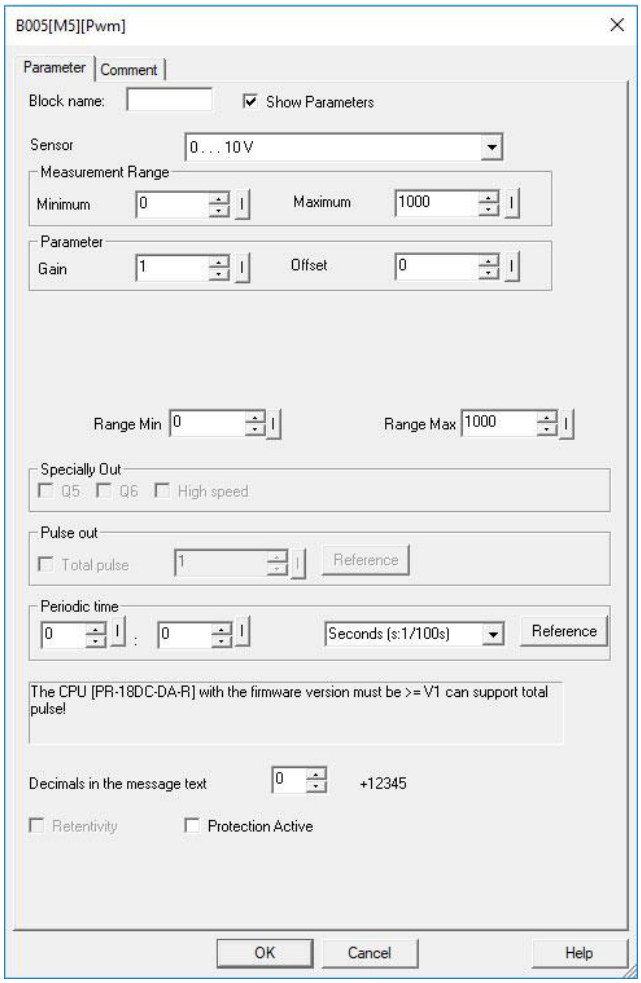
NOTICE:
The special function shift register can be used only once in the circuit program.

Pulse Width Modulator (PWM)

Description of function

The Pulse Width Modulator (PWM) instruction modulates the analogue input value Ax to a pulsed digital output signal. The pulse width is proportional to the analogue value Ax.

Connection	Description
Input En (—▸)	A positive edge (0 to 1 transition) at input En enables the PWM function block.
Input Ax	Analog signal to be modulated to a pulsed digital output signal.
Parameter	**A:** Gain Range of values: +- 10.00 **B:** Zero offset Range of values: +- 10,000 **PT:** Periodic time over which the digital output is modulated **p:** Number of decimals Range of values: 0, 1, 2, 3
Output Q	Q is set or reset for the proportion of each time period according to the proportion of the standardized value Ax to the analogue value range.



Parameter PT

The periodic time PT can be provided by the actual value of another already-programmed function:

Analog comparator (Ax – Ay)

Analog threshold trigger (Ax)

Analog amplifier (Ax)

Analog multiplexer (AQ)

Analog ramp (AQ)

Mathematic instruction (AQ)

PI controller (AQ)

Up/down counter (Cnt)

Analog filter (AQ)

Average value (AQ)

Max/Min (AQ)

On-delay (Ta)

Off-delay (Ta)

On-/off-delay (Ta)

Retentive on-delay (Ta) Wiping relay(pulse output) (Ta)

Edge triggered wiping relay (Ta)

Asynchronous pulse generator (Ta)

Stairway light switch (Ta) Multiple function switch (Ta)

Stopwatch (Ta)

Parameter p (number of decimals)

Parameter p applies only to the display of the Ax value in a message text.

Description of the function

The function reads the value of the signal at the analogue input Ax.

This value is multiplied by the value of parameter A (gain). Parameter B (offset) is added to the product, as follows:

(Ax * Gain) + Offset = Actual value Ax

The function block calculates the proportion of the value Ax to the range. The block sets the digital output Q high for the same proportion of the PT (periodic time) parameter and sets Q low for the remainder of the time period.

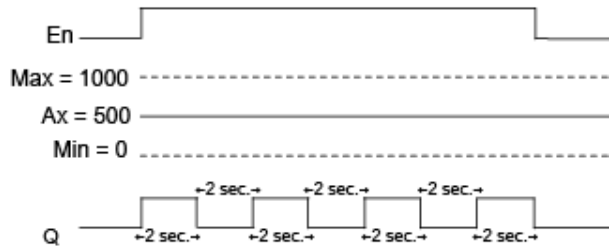
Examples of Timing Diagrams

The following examples show how the PWM instruction modulates a digital output signal from the analogue input value:

Example 1

Analog input value: 500 (range 0...1000)
Periodic time T: 4 seconds

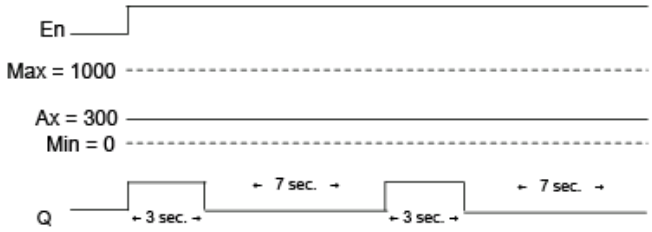
The digital output of the PWM function is 2 seconds high, 2 seconds low, 2 seconds high, 2 seconds low and continues in that pattern as long as parameter "En" = high.



Example 2

Analog input value: 300 (range 0...1000)
Periodic time T: 10 seconds

The digital output of the PWM function is 3 seconds high, 7 seconds low, 3 seconds high, 7 seconds low and continues in that pattern as long as parameter "En" = high.



Calculation rule

Q = 1, for (Ax – Min) / (Max – Min) of time period PT
Q = 0, for PT – [(Ax – Min) / (Max – Min)] of time period PT.

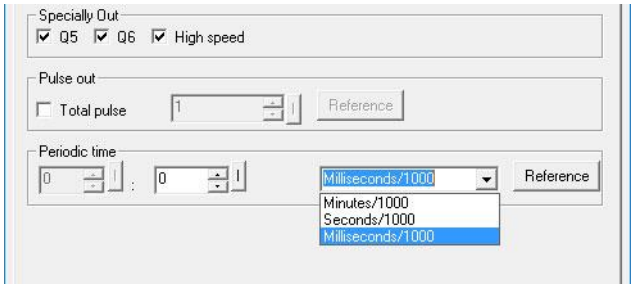
Note: Ax in this calculation refers to the actual value Ax as calculated using the Gain and Offset. Min and Max refer to the minimum and maximum values specified for the range

Special feature.

The standard output frequency of xLogic is up to 30Hz. By configuring the PWM function block as shown below a frequency of up to 10 kHz can be obtained on the following CPU outputs:

- PR-12 – Outputs Q3 and Q4
- PR-18 and ELC-22 (PNP Transistor) – Outputs Q5 and Q6
- PR-24 and PR-26 – Outputs Q7 and Q8

By the following configuration:



If the special output is selected in the property dialogue box of PWM block, then the unit of "periodic time" will be changed from s:1/100s to s:1/1000s, so if you input 3 (1/1000s), then its frequency will be 1000/3 Hz.

NOTICE:

1. The periodic time must be no less than 3 ms.
2. If the specific output is selected in the property dialogue box of PWM block, then the output pin of the PWM function block cannot be linked as an input to other blocks.
3. Q3, Q4 in the above dialogue box correspond to Q3, Q4 of a PR-12 CPU or Q5, Q6 of a PR-18 and ELC-22 CPUs or Q7 and Q8 of the PR-24 and PR-26 CPUs.
4. The output frequencies of the special channels of the CPUs can be different ONLY if ONE box is ticked. If BOTH boxes are ticked, the outputs will have the same maximum frequency.

Modbus Read

Description of function

When there is a high level at En, the Modbus Read block will be activated and the xLogic shall communicate with a peripheral device as a master via RS232 or RS485 interface. Furthermore, the output will be switched on when communication is established successfully. Otherwise, the output (Q pin) remains "off" which means communication has failed.

A signal at input R resets output Q and disables this block at the same time

Connection	Description
Input En (—▷)	A high signal at En input will enable "Modbus Read" function block to be activated
Input R (—○)	To disable the MODBUS read function and set the output to 0 via the R (Reset) input. Reset has priority over En

Parameter	**Slave address**: 1 is a default. **Communication protocol:** Modbus (RTU) **Communication parameter:** baud rate (BPS), Data bits, Stop bits, Parity, Overtime (response timeout) **Comm Type:** RS232 or RS485(Communication interface of xLogic) **Data register Index:** High Low /Low High **Command:** 01 Read coils(0x) 02 Read Discrete Input(1x) 03 Read Holding Registers(4x) 04 Read Input Registers(3x) **Register start address, count**
Output Q	Q is set or reset depending on the communication status. Successful communication, Q=1; Failed communication, Q=0;

NOTICE:

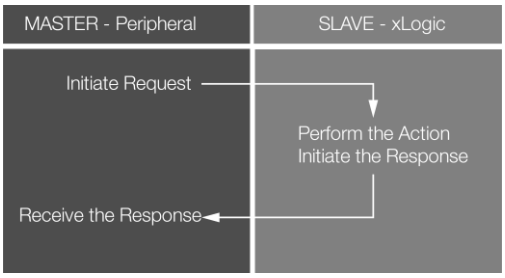
Data register Index: High Low /Low High

For example, when the HighLow index is set, and the data 0x0012 was read and saved to AQ, then AQ= 0X0012; however, when the LowHigh index is set, then AQ= 0x1200.

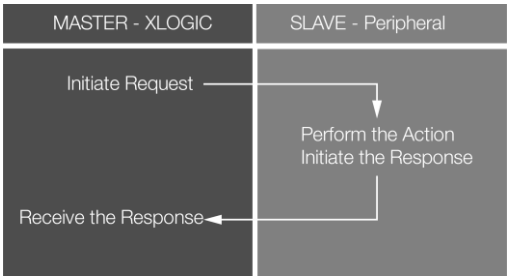
Refer to the Modbus RTU Communication Manual for more information.

Description of the function:

In the configuration of our xLogic communication, the xLogic usually serves as a slave via Modbus RTU Protocol and can communicate with a master directly. That's to say, any device communicating with xLogic sends a command to it, and then its response will be sent out only when the xLogic has received the command, just as the below figure shows:

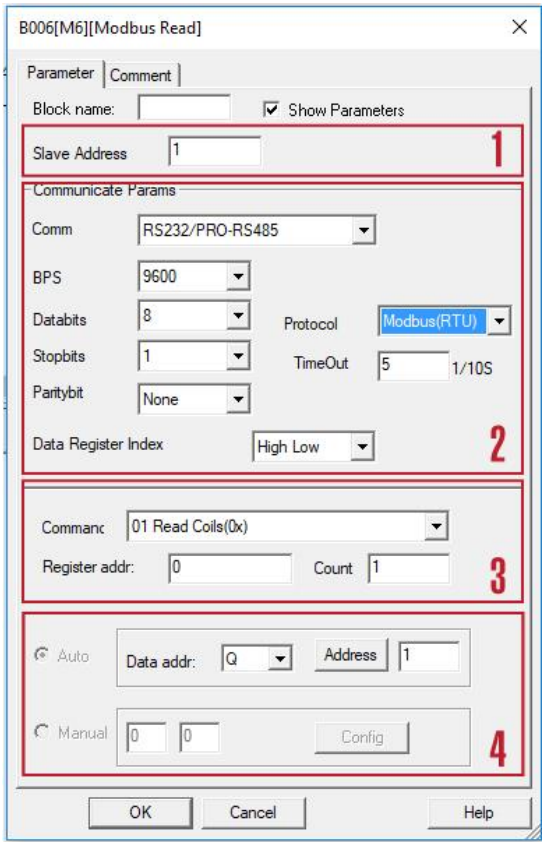


However, the "Modbus Read" or "Modbus Write" (see next chapter) function blocks should be used if xLogic shall be required to play a role of master to communicate with other slave devices. As the following figure shows:



When you put the "Modbus read" or "Modbus Write" function block in your program the xLogic CPU will behave as a Modbus RTU Master.

The properties of "Modbus Read" block are shown below:



1. Slave Address: 1 is the default

2. Communication parameters : BPS is baud rate, Stop bits, Data bits

Communication type: RS232 or RS485 interface of xLogic CPU.

3. Command, register address, and register count

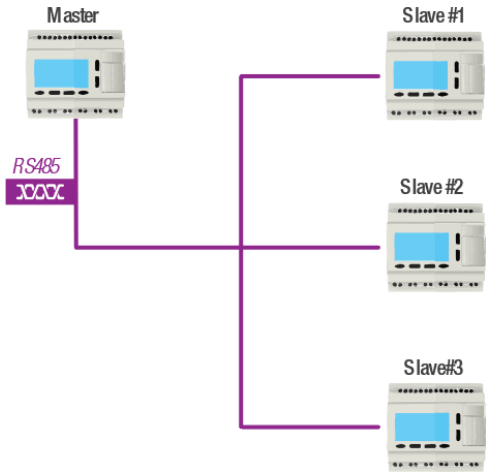
4. Where to save the data read from the Slave.

Command	Function description	Remark
01	Read one group coil status (00000～0XXXX)	Read Coil Status (output)
02	Fetch one group data of the status of switch input (10000～1XXXX)	Read input Status (input relay)
03	Read data of multi-holding register (40000～4XXXX)	Read Holding Registers (Output register)
04	Read data of input registers (3000～3XXXX)	Read Input Registers

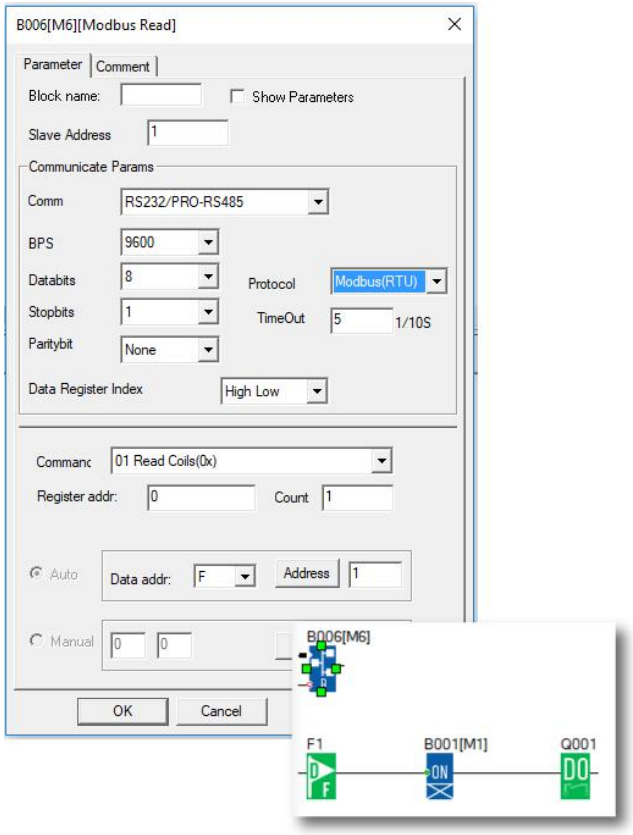
NOTICE:

Please use "03" command to read AI/AO of xLogic.

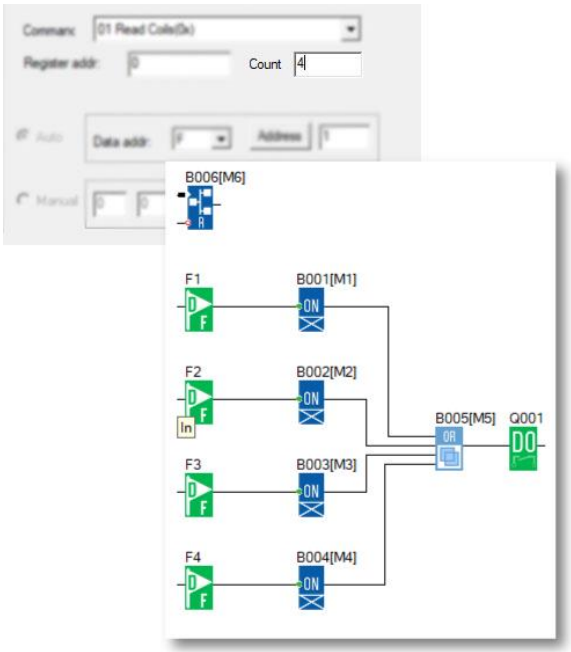
Example: The following shows one xLogic (Master) communicating with three xLogic (Slave) via RS485.



Example 1: Get the Q1 status of SLAVE1(xLogic) and then save the bit status to F1.

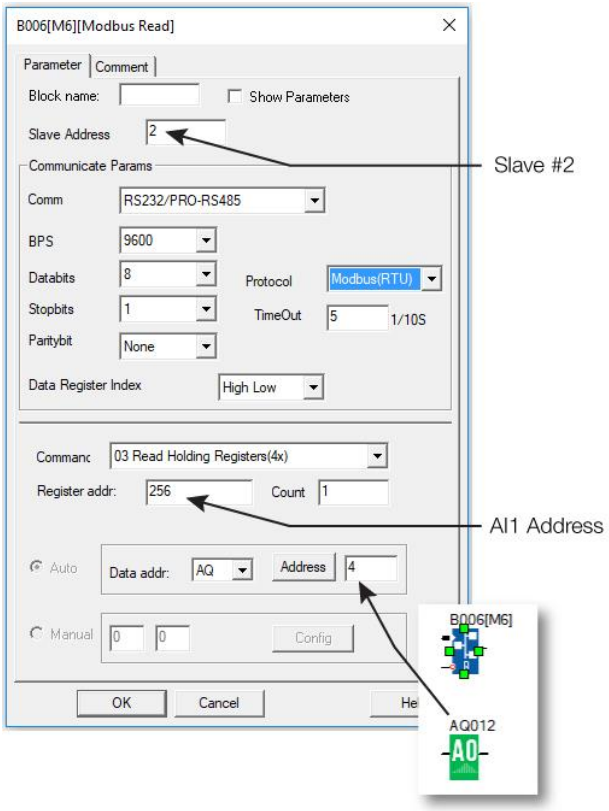


If the count was set to 4, then Q1, Q2, Q3, Q4 of xLogic (Slave No.1) would be read and saved flags F1 to F4 of the Master CPU.



F is bit type flag. It can be used to receive bit data from a slave device.

Example 2: Get AI value from Slave 2 (xLogic with station No.2) and write the data to AQ12, the second analogue output of Ext. 1



The number setting of Q, I, AQ are continuous. AQ12 cannot be set as AQ 12 and should be set AQ 4 as above figure shows.

The following table shows how to set the addresses.

NOTICE:

This table also can be applied to the configuration of the Modbus Write function block.

Model	I, Q, AI, AQ	Dialog box set
CPU	I1-IC	I1-I12
	Q1-Q6	Q1-Q6
	AI1—AI8	AI1—AI8
	AQ1—AQ2	AQ1—AQ2
Expansion1 (Address is 1)	I11-I18	I13-I20
	Q11-Q18	Q7-Q14
	AI11-AI18	AI9—AI12
	AQ11-AQ12	AQ3, AQ4
Expansion2 (Address is 2	I21-I28	I21—I28
	Q21-Q28	Q15—Q22
	AI21-AI28	AI17---AI24
	AQ21-AQ22	AQ5--AQ6
……		

The data format of instruction:

Name	Data Format
F, I, Q	Bit
AF, AI, AQ	Signed Short

Modbus Write

Description of function

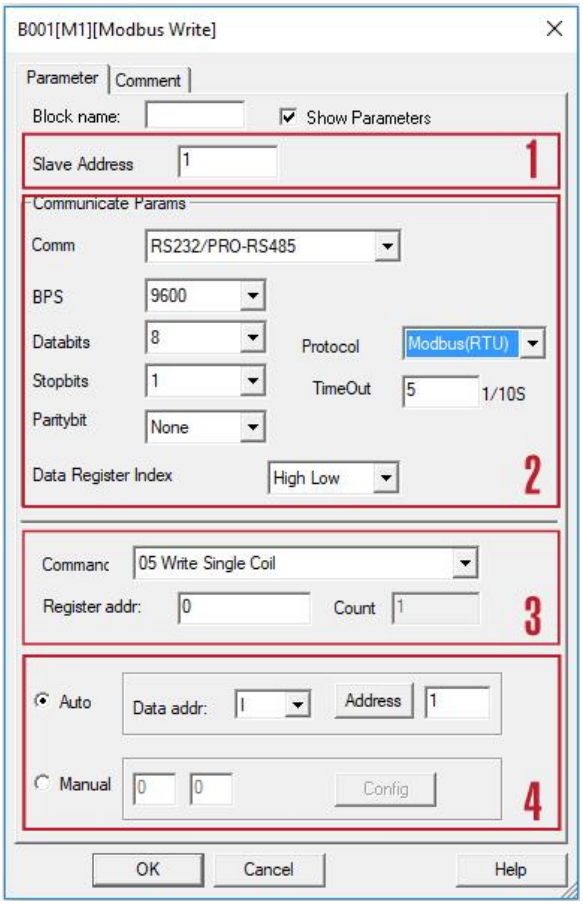
With a high level at En, the Modbus Write block will be activated and the xLogic could communicate with peripheral as a master via RS232 or RS485 interface and the output of the block will be switched on when the communication is established successfully. Otherwise, the output (Q pin) is kept "off" which means communication has failed.

A signal at input R resets output Q and disables this block at the same time.

Connection	Description
Input En (—▸)	A high signal at En input will enable "Modbus Write" function block to be activated
Input R (—○)	Reset the value read from peripheral and set the output to 0 via the R (Reset) input. Reset has priority over En
Parameter	**Slave address**: 1 is a default. **Communication protocol:** Modbus (RTU) **Communication parameter:** baud rate (BPS), Data bits, Stop bits, Parity, Overtime (response timeout) **Comm Type:** RS232 or RS485 (Communication interface of xLogic) **Data register Index:** High Low /Low High

	Command: 05 Write Single Coil 06 Write Single Register 15 Write Multiple Coils 16 Write Multiple Registers **Register start address, count**
Output Q	Q is set or reset depending on the communication status. Successful communication, Q=1; Failed communication, Q=0;

The configuration of a "Modbus Write" block is shown below:



1. Slave Address: 1 is a default

2. Communication parameters: BPS is baud rate, Stop bits, Data bits

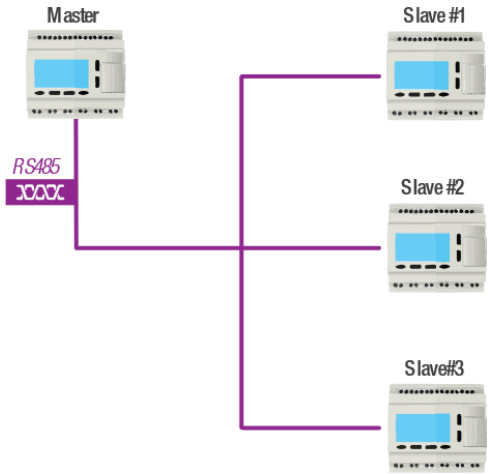
Communication type: RS232 or RS485 interfaces of xLogic CPU.

3. Command, register address, and register count

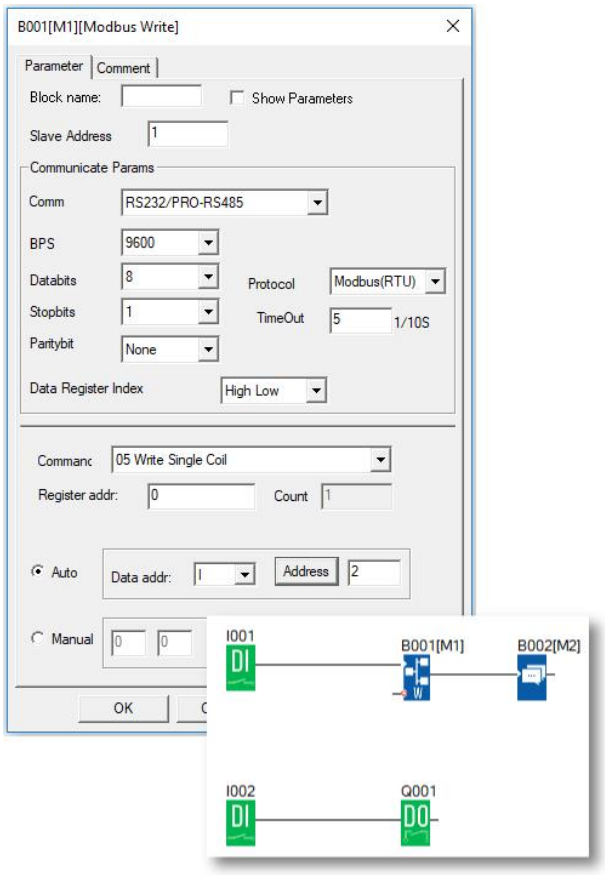
Command	Function description	Remark
05	Force the switch status of single coil (00000-0XXXX)	Force Single Coil (output)
06	Pre-set the data of a single register (40000-4XXXX)	Set single output register
15	Force multi-coils on/off bit (00000-0XXXX)	
16	Write multi-holding registers data (40000-4XXXX)	

4. Data that is to be written to the Slave can be configured in one of two ways. One is in auto mode, this data uses the flags in the program, such as F, M, AF, AM, I, Q and AQ. The second, manual mode is used to set a fixed value or bit status.

Example 1



Write the I2 bit status of Master xLogic to Slave No1 xLogic and control Q1 of the slave via RS485 port. The program of a master would be as follows:

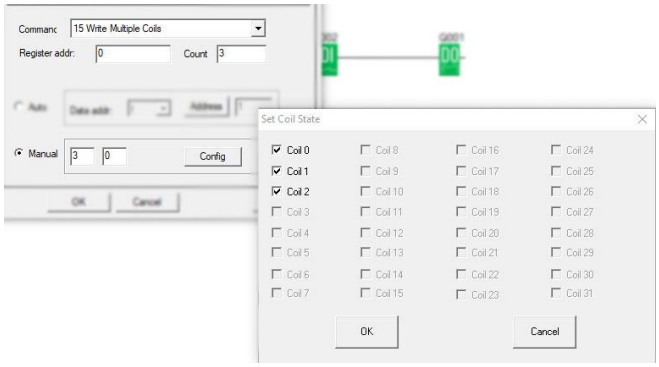


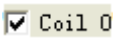
I1 of the master is used to control the communication. If I1 is high and the communication is established successfully, one alarm message (text message block) will be displayed on the LCD. Then Q1 of slave No.1 will be controlled by I2 of the master. If I2 is high, Q1 of slave No.1 would be ON and if I2 is low, Q1 of the slave would be OFF.

NOTICE:

The Q1 of the Slave must be free – No other blocks can be connected to its input pin.

Example 2, manual mode input value



The above configuration is to force Q1, Q2, Q3 of Slave No.1 ON.  "means pre-set the BIT 1 and "means pre-set the BIT 0 "Coil 0" is corresponding to the start address, Here it is Q1.

NOTICE:

The manual input value is Hex data .it contains 4 bytes. If you want to write a decimal value to the register of the slave, it must be converted to Hex format first.

4. The following table shows how to set each output.

Note: This table also applies to the Modbus Read function block

Model	I, Q, AI, AQ	Dialog box set
CPU	I1-IC	I1-I12
	Q1-Q6	Q1-Q6
	AI1—AI8	AI1—AI8
	AQ1—AQ2	AQ1—AQ2
Expansion1 (Address is 1)	I11-I18	I13-I20
	Q11-Q18	Q7-Q14
	AI11-AI18	AI9—AI12
	AQ11-AQ12	AQ3, AQ4
Expansion2 (Address is 2	I21-I28	I21—I28
	Q21-Q28	Q15—Q22
	AI21-AI28	AI17---AI24
	AQ21-AQ22	AQ5--AQ6
.....		

Data format instruction

Name	Data Format
F, I, Q	Bit
AF, AI, AQ	Signed Short

For detailed information about the Modbus addresses of different block types, see the Modbus RTU Protocol User Manual.

Modbus Read Write

Description of function

This function block is for Ethernet PLC which have a built-in webserver. This block is used when the CPU is to act as a Modbus Master (Client).

If the CPU does not have a built-in webserver, then please use the standard Modbus Read and Modbus Write blocks above.

When activated by a high signal on the En pin, this block will enable the CPU to communicate via RS232 or RS485 or Ethernet as a Modbus Master (Client). If the connection to the Slave (Server) device is established, then the output pin of this block will be set high.

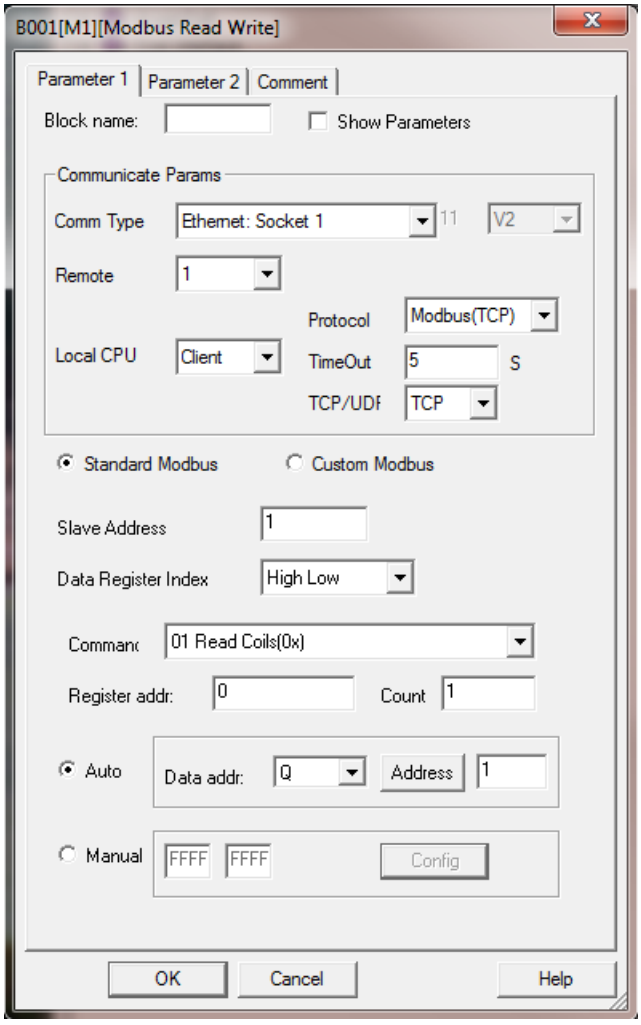
Connection	Description
Input En	A high signal at the En input will enable the Modbus Read Write block to be activated.
Input R	Resets the block with a high signal on the R pin. The CPU will be disconnected from the Slave. Input R has priority over input En.
Parameter 1	**Communicate Parameters** • **Comm Type:** o Com 0 (RS232 or PRO485) o Com 1 (Expansion RS485) o Com 2 (Built-in RS485 o Ethernet (Channels 1 to 8) • **Com 0, 1 or 2 Settings:** o BPS: Baud Rate o Data Bits: 5, 6, 7 or 8 o Stop Bits: 1, 1.5 or 2 o Parity Bit: None, Odd, Even, Mark or Space • **Ethernet Settings:** o Remote: 1 to 8 o Local CPU: Client (Master) or Server (Slave) • **Protocol:** o Modbus RTU or Modbus/TCP • **Timeout:** Connection timeout in seconds. • **TCP/UDP:** Use TCP or UDP • **Modbus Type:** Standard or Custom (See Parameter 2 below) • **Slave Address:** Default is 1 • **Data Register Index:** High/Low or Low/High (Word Swapped) • **Command (Modbus Function):** o 01 Read Coil (0x) o 02 Read Discrete Input (1x) o 03 Read Input Registers (4x) o 04 Read Holding Registers (3x) o 15 Write Multiple Coil o 16 Write Multiple Registers • **Register Address:** Start register • **Count:** Number of registers • **Auto:** o For Modbus Commands 01 and 02, options are: ▪ Q – Physical outputs ▪ F – Digital Flags o For Modbus Command 03, options are: ▪ AF- Analogue flags

	▪ AQ – Analogue outputs ▪ RTC – Real Time Clock ▪ VW – Word blocks o For Modbus Command 04, options are: ▪ AF- Analogue flags ▪ AQ – Analogue outputs ▪ VW – Word blocks o For Modbus Command 15, options are: ▪ Q – Physical outputs ▪ F – Digital Flags ▪ I – Physical inputs o For Modbus Command 16, options are: ▪ AF- Analogue flags ▪ AQ – Analogue outputs ▪ RTC – Real Time Clock ▪ VW – Word blocks ▪ AI – Analogue inputs • **Manual:** Use coils max.32 (FFFF FFFF)
Parameter 2	• **Identification String:** Max. 20-characters • **Data: Start Address and Count (No. of registers) for Types:** o Digital Inputs, Digital Outputs, Analogue Inputs, Analogue Outputs, Digital Flags, Analogue Flags, M Registers and AM Registers

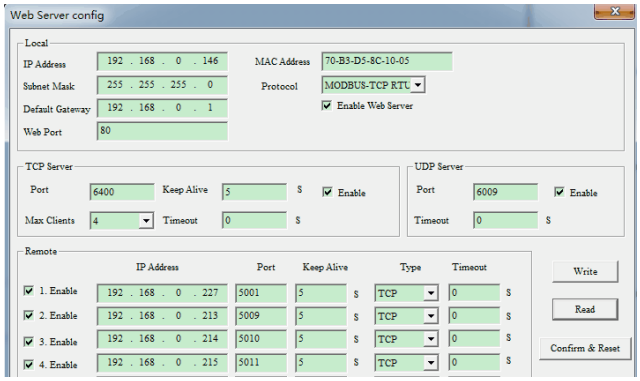
Output Q	When communications established with Slave (Server), the output is set High

Example 1 Local CPU as Master (Client)

The CPU is to work as a Master (Client) using standard Modbus/TCP to read one coil (Count) from the Slave (Server) at address register 0 and write the value to the CPUs first output coil, Q1.



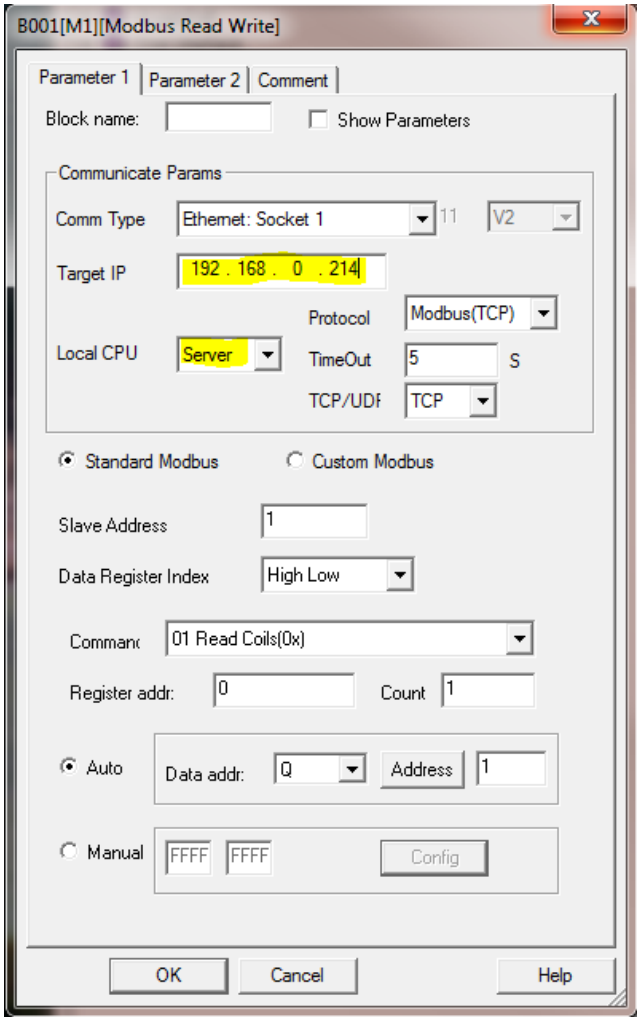
Remote 1 is the channel that belongs to the Slave and its IP address (192.168.0.227) and port number (5001) is configured using Tools → Transfer → Web Server Config:



If we had selected Remote 2, the Slave IP would be 192.168.0.213, Remote 3 then IP would be 192.168.0.124, etc.

Example 2 Local CPU as Server (Slave)

For the CPU to work as a Server (Slave) we only need to configure the IP address of the Client (Master).



Example 3 Local CPU using Customised Modbus

The CPU needs to provide the following information to a Server (Slave) using the communication parameters set in example 1 above:

Identification string: x-Messenger station1

Digital Input states for I3, I4, I5 and I6

Digital Output states for Q1, Q2, Q3 and Q4

Analogue Inputs AI1, AI2, AI3 and AI4

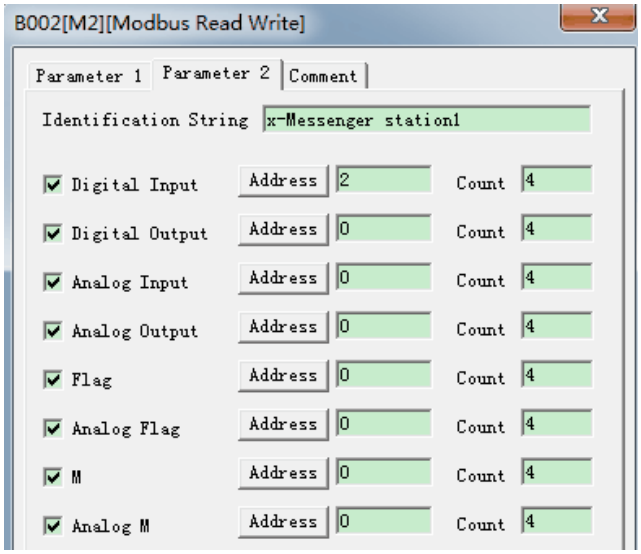
Analogue Outputs AQ1, AQ2, AQ3 and AQ4

Digital Flags F1, F2, F3 and F4

Analogue Flags AF1, AF2, AF3 and AF4

M Coils M1, M2, M3 and M4

AM Coils AM1, AM2, AM3 and AM4



The data received by the Server would be:

00 09 00 00 00 4c 01 8b 49 14 78 2d 4d 65 73 73 65 6e 67 65 72 20 73 74 61 74 69 6f 6e 31 01 01 01 02 01 0d 0b 08 00 80 00 88 00 00 00 00 0c 08 00 00 00 00 00 80 00 88 16 01 02 17 08 01 c2 00 00 02 26 00 00 07 01 04 0d 08 00 00 00 00 00 01 c2

Description of data transmitted:

00 09 00 00 00 4c : MODBUS TCP data prefixed by six bytes

01 :x-Messenger address(the default is 01)

8b : Upload code(fixed, if GPRS Data upload used)

49 :length field =number of bytes following

14 :length field = number of bytes of Identification string

78 2d 4d 65 73 73 65 6e 67 65 72 20 73 74 61 74 69 6f 6e 31 : x-Messenger station1(Identification String)

01 01 01

01 : type code (01 means the digital inputs)

01 : length field = number of bytes of the digital inputs

01 : status of the digital inputs(I3=1,I4=0,I5=0,I6=0)

02 01 0d

02 : type code (02 means the digital outputs)

01 : length field = number of bytes of the digital outputs

0d : status of the digital outputs (Q1=1,Q2=0,Q3=1,Q4=1)

0b 08 00 80 00 88 00 00 00 00

0b : type code (0b means the Analogue Input)

08 :length field = number of bytes of the analogue inputs

00 80 00 88 00 00 00 00 : analogue inputs value (AI1= 00 80 , AI2 = 00 88, AI3= 00 00,AI4= 00 00)

0c 08 00 00 00 00 00 80 00 88

0c : type code (0c means the Analogue output)

08 : length field = number of bytes of the analogue outputs

00 00 00 00 00 80 00 88 : analogue outputs value (AQ1= 00 00 , AQ2 = 00 00, AQ11= 00 80,AQ12= 00 88)

16 01 02

16 : type code (16 means the digital flag)

01 : length field = number of bytes of the digital flag

02 : status of the digital flag (F1=0,F2=1,F3=0,F4=0)

17 08 01 c2 00 00 02 26 00 00

17 : type code (17 means the Analogue flag)

08 : length field = number of bytes of the analogue flag

01 c2 00 00 02 26 00 00 : analogue flag value (AF1= 01 c2 , AF2 = 00 00, AF3= 02 26,AF4= 00 00)

07 01 04

07 : type code (07 means the M status)

01 : length field = number of bytes of M(Function block status(1/0))

04: status of the M (M1=0,M2=10,M3=1,M4=0)

0d 08 00 00 00 00 00 01 c2

0d : type code (0d means the AM value)

08 : length field = number of bytes of AM

00 00 00 00 00 00 01 c2: AM value (AM1=00 00,AM2=00 00,AM3=00 00,AM4=01 c2)

Response expected from Server (Slave)

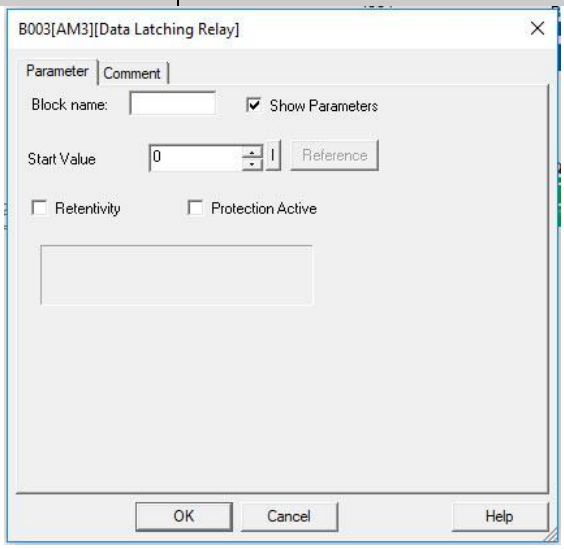
00 00 00 00 00 02 01 8b

Data Latching Relay

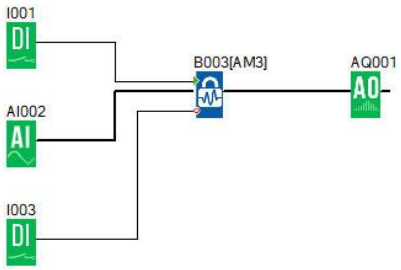
Description of function

This special function saves the process variable of an analogue input to memory and returns it at the analogue output.

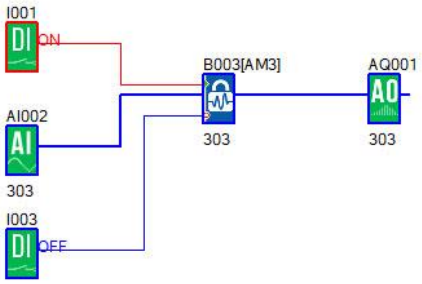
Connection	Description
Input S (—▸)	Save the Ax to memory and return it at the analogue output with a signal at input S (Set).
Input Ax	Input the analogue signal to be amplified at input Ax. Use the analogue inputs, the block number of a function with analogue output, or the analogue outputs.
Input R (—○)	Reset analogue output AQ to 0 with a signal at input R (Reset).
	analogue Output AQ is reset if S and R are both set (reset has priority over set).
Parameter	Value range for **Start value**: -32768...+32767
Output AQ	Analog output Value range for AQ: -32768...+32767



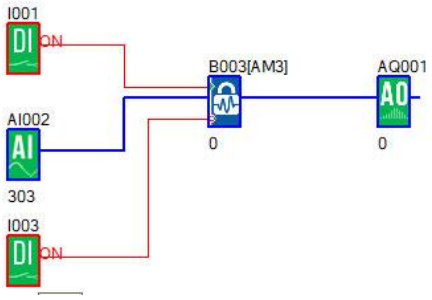
Example



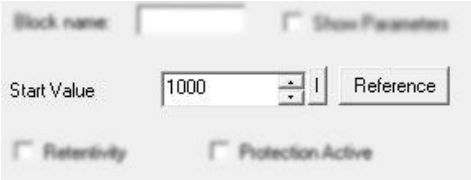
When I1 is HIGH, the value of AI2 will be saved to memory and sent to AQ1 as follows:



When I3 is HIGH, the value of this function block will be reset to 0.

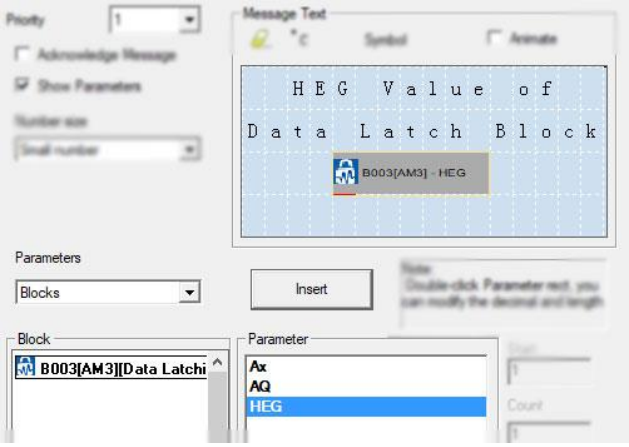


1. Start value – You can set a start value for the data latch relay block, which can be a fixed value or use the value from an existing programmed block by using the 'Reference' button.



2. The HEG value of the data latch relay can be modified at runtime by pressing the OK key for 3 seconds on the CPU front LCD.

The HEG value of data latch relay block also can be displayed with a decimal point if required.



There is a Modbus address for the HEG, so you can also change the HEG value by a Modbus Master (Touchscreen etc.). You can find the address in the Modbus RTU User Manual or written in next to the block in your program.

NOTICE:
If your firmware CPU doesn't support this function, you can download the firmware update package from our website.

Long Data Latching Relay

Description of function

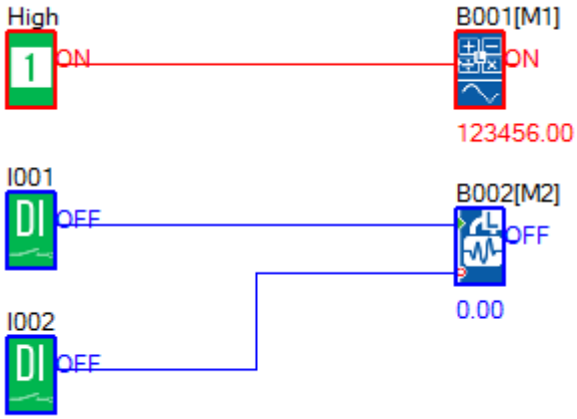
This block can be used to store an analogue value to memory. When the Set pin 'S' is set High, the analogue value referenced by the block will be stored. When the Reset pin 'R' is set High, the value will be reset to 0.

Connection	Description
Input S	Only when there is a low to high trigger at Set pin, the block will store the referenced value to memory.
Parameter	The block referenced which contains the analogue value to be stored. This can be any pre-defined analogue type block such as inputs, flags, timers, maths, etc.
Input R (—○)	A trigger of Low to High on the Reset (Input R) pin will set the memory to 0. Reset has priority over Input S. If both pins are High Pin R has priority over Pin S.

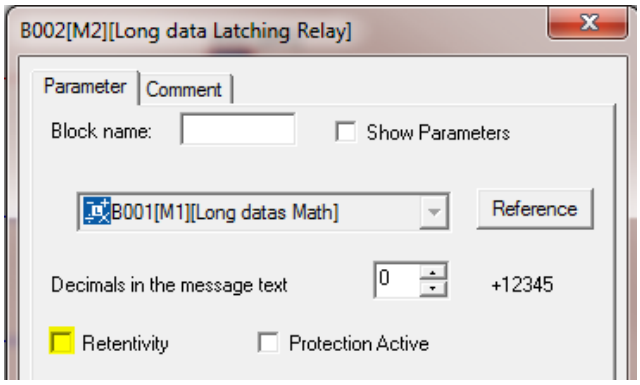
Value Range	The range of value capable of being stored to memory is -2,147,483,648 to 2,147,483,647

Example

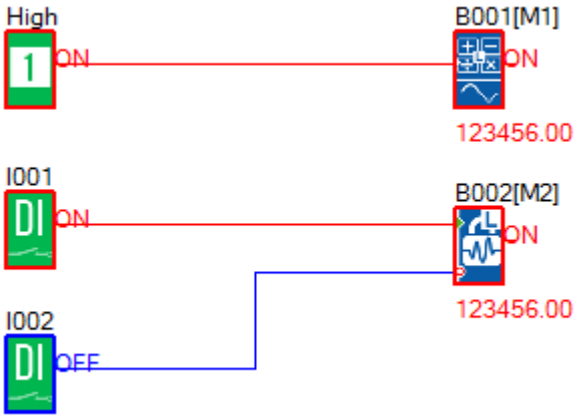
The Long Data Math block (B001) has a value of 123,456.



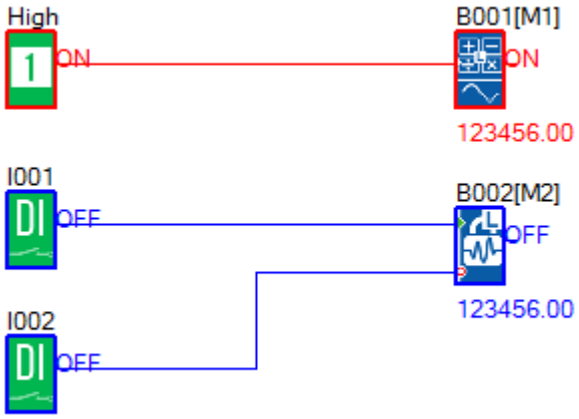
The Long Data Latching Relay block (B002) is configured to reference B001. Note that is Retentivity is checked, then the value in memory will be reserved on power cycle of the PLC. If Protection Active is checked, then user will be unable to change the blocks value via the PLC LCD.



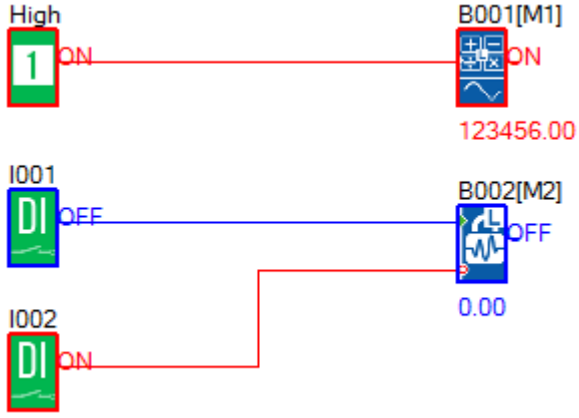
When the Input I001 is High, block B002 stores the current value of block B001.



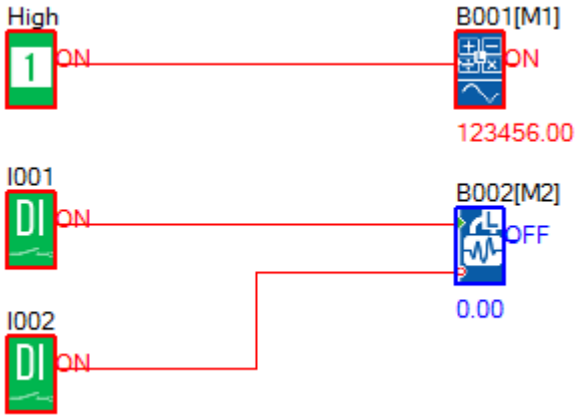
Even if the Set pin returns to Low, the value is latched.



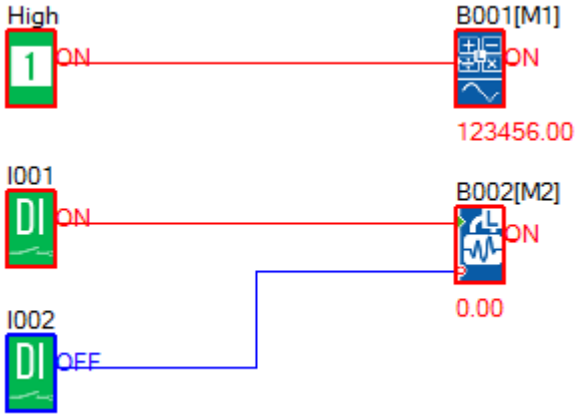
A High state at the Reset pin, resets the value to 0.



If Both the Set and Reset pins are High, the Reset pin takes priority and the value is set to 0.



If the Reset pin is set to OFF, the value retained is still 0 because the Set pin only triggers a storing of the referenced value with a Low to High transition.



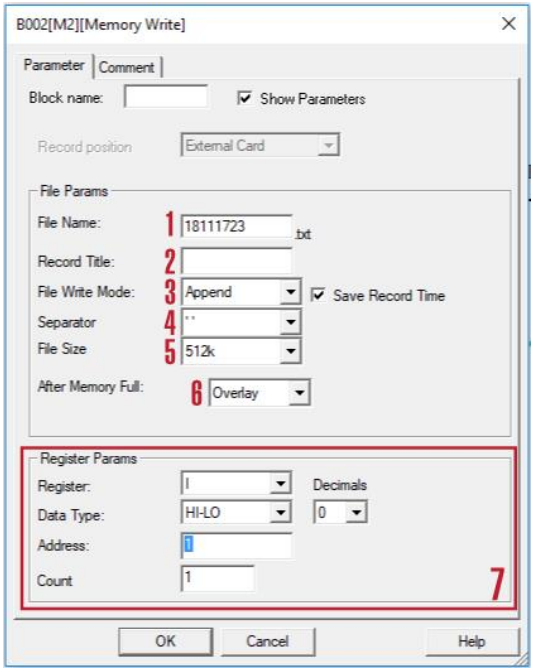
Memory Write

Description of function

Only when there is a low to high trigger at Trg pin, the Memory Write block will be activated and the pre-configured record action will be performed, at the same time the output will switch on if the record action had been done successfully.

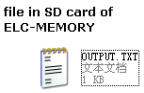
Connection	Description
Trg input (—●)	Only when there is a low to high trigger at Trg pin, the Memory Write block will be activated and the pre-configured record action will be performed. Each trigger only writes once.
Input R (—○)	Reset the Memory Write block and set the output to 0 via the R (Reset) input. Reset has priority over Trg
Output Q	Q switches on only after Write function had been executed correctly.

Description of Memory write block's property dialogue box:



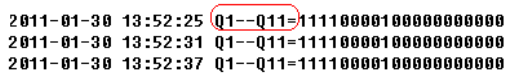
1. Filename

Place where you can set the name of the file used to save the registers' data



Above is an example of the "OUTPUT.TXT"

2. Record title



The above range circled in red is just pre-set contents in the "Record title" of the Memory write block's property dialogue box.

3. File Write Mode

Two options available:

Option A. Append (This option would be selected if a certain file already exists in the Mini SD card inserted in ELC -MEMORY)

B. Create (This option should be chosen, if no file existed or the existing file has a different name from that pre-set in the "filename" in the Mini SD card inserted in ELC-MEMORY

If this box has been ticked, the file content will show the time when the data starts to be recorded.

2011-01-30 13:52:37 Q1--Q11=1111000001000000000000

4. Separator

Such separator shall be required while more than one analogue values would be stored and displayed for easier observation and convenient analysis.

5. File Size

Sets the size of the file to be stored.

6. After memory Full

Two options can be selected after the memory is full (it means the relative file has reached its pre-configured size), one is to over-write and the other is to stop recording.

7. Register params:

This section is for register's parameters setting. The register includes the following choices:

I – Digital Inputs

Q - Digital Outputs

F - Digital Flag

M – Digital Block Statuses

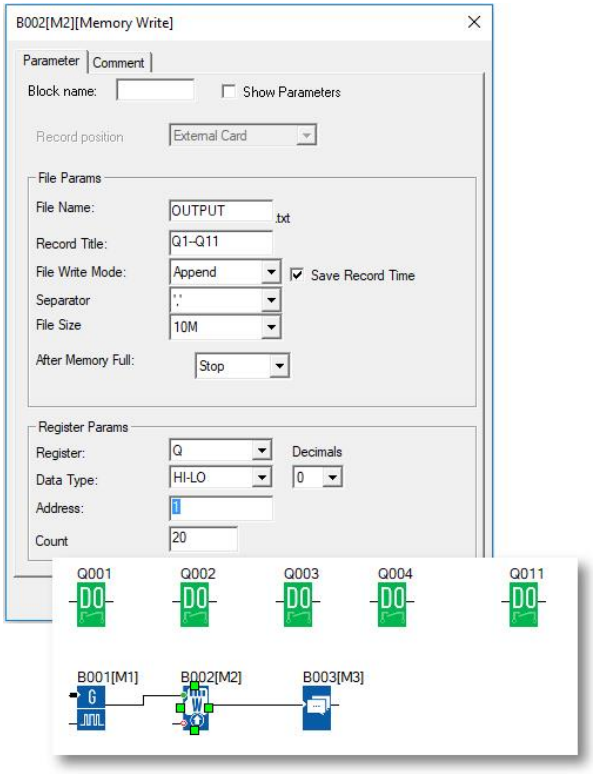
AI - Analogue Inputs

AQ - Analogue Outputs

AF - Analogue Flag

AM – Analog Blocks Values

EXAMPLE:



Please refer the property dialogue box of B002, it can record the output status. The start address is from 1 and it must record the 20 outputs with continuous addresses.

And the record file shows below:

2011-01-30 13:52:25 Q1--Q11=1111000010000000000000
2011-01-30 13:52:31 Q1--Q11=1111000010000000000000
2011-01-30 13:52:37 Q1--Q11=1111000010000000000000
2011-01-30 13:52:43 Q1--Q11=1111000010000000000000
2011-01-30 13:52:49 Q1--Q11=1111000010000000000000
2011-01-30 13:52:55 Q1--Q11=1111000010000000000000
2011-01-30 13:53:01 Q1--Q11=1111000010000000000000

Per program, every 6 seconds the record will do once, and the Q1, Q2, Q3, Q4, Q11 will be all "ON". You can see the

record file and you'll see the recording time and the status of the output.

Q1--Q11=1111000010000000000
Q1-Q4 Q11

NOTICE:

1. The ELC-MEMORY can only be inserted into the RS232 port (programming port) of ELC series CPU.

2. If this function block is working, the RS232 port (programming port) will be occupied, some data will be being transferred, if you want to use the programming port for some purposes (for example download or upload program), you must make sure the Trg pin of this block keeps at Low status or stop the CPU using the panel keys on the LCD.

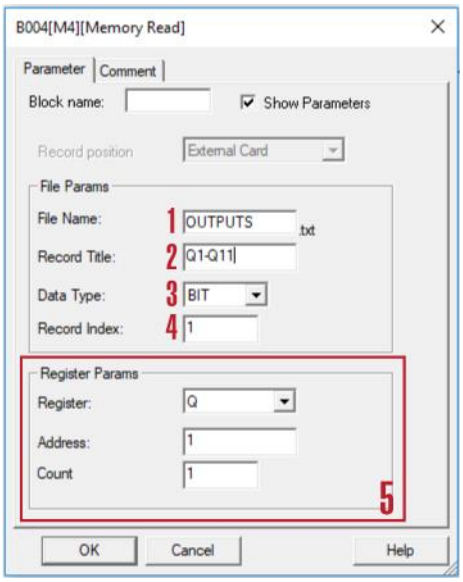
Memory Read

Description of function

Only when there is a low to high trigger at Trg pin, the Memory Read block will be activated once and xLogic CPU will read correlative data (bit or short) to set pre-configured register from the file in the SD card of ELC-MEMORY module, at the same time the output will switch on if the read action had been done successfully.

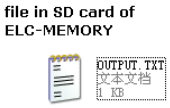
Connection	Description
Trg input (—●)	Only when there is a low to high trigger at Trg pin, the Memory Read block will be activated and xLogic CPU will read some data (bit or short) to set pre-configured register from the file in the SD card of ELC-MEMORY module. Each trigger only writes once.
Input R (—○)	Reset the Memory Read block and set the output to 0 via the R (Reset) input. Reset has priority over Trg
Output Q	Q switches on only after the Read function had been executed correctly, provided.

Description of Memory write block's property dialogue box:



1. Filename

The name of the file which you want to access is stored in the mini-SD card of the ELC-MEMORY module.



Above is an example of the "OUTPUT.TXT"

2. Record Title

```
2011-01-30 13:52:25 Q1--Q11=1111000010000000000
2011-01-30 13:52:31 Q1--Q11=1111000010000000000
2011-01-30 13:52:37 Q1--Q11=1111000010000000000
```

3. Data Type:

Two options are available:

- Option A. BIT (0 or 1, this is used to be set the status of Q or F)
- Option B. WORD (this is used to be set the value of AQ or AF)

4. Record Index:

Here is used to set which line the CPU will access via this Memory Read block

5. Register Params

Here is how to set the parameters of the register, all these registers have a "write" property.

BIT data can be used to set the register "Q" and "F".

Q: Digital Outputs

F: Digital Flag

WORD data can be used to set the register "AQ" and "AF"

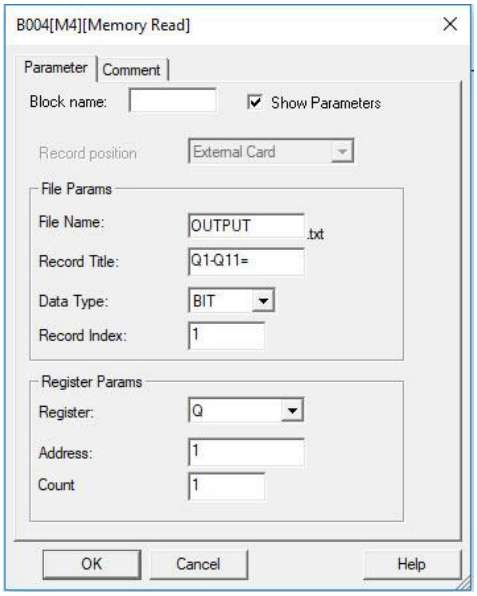
AQ: Analogue Outputs

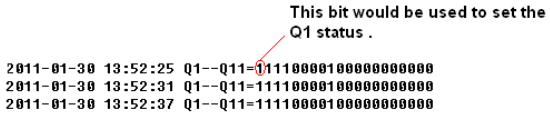
AF: Analogue Flags

Count

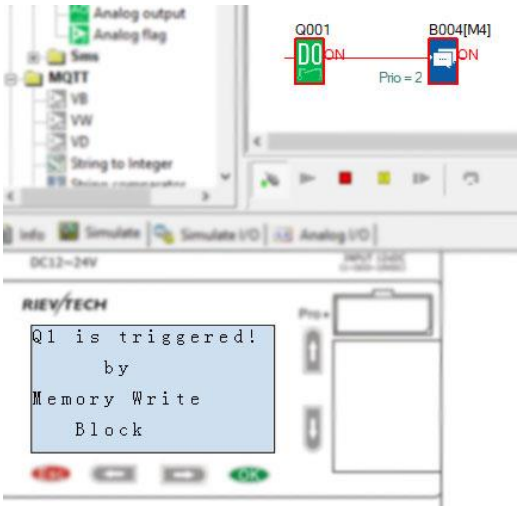
Sets the numbers of registers to be written.

For example





If the Memory Read block had been triggered, the Q1 of PR-12 CPU will be set "1".

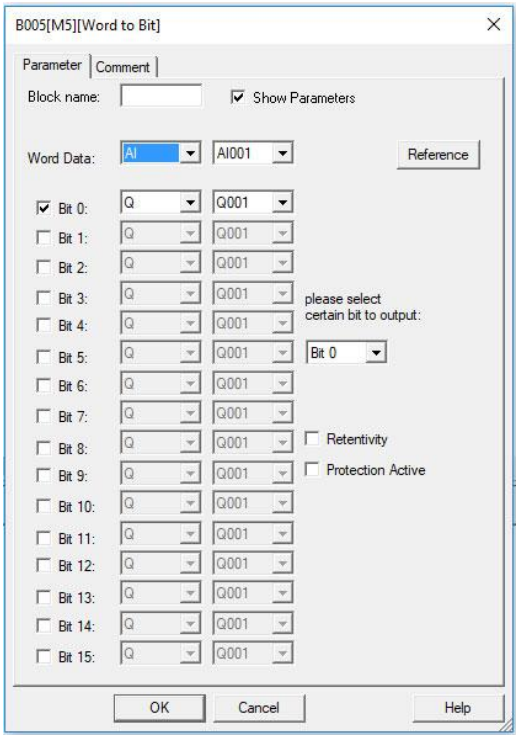


Word to Bit

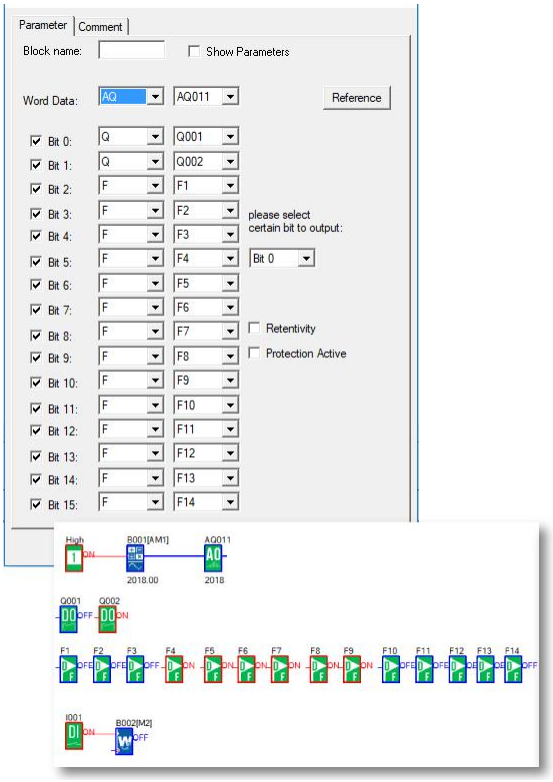
Description of function

This special function is used to transfer the word type data (AI, AF or AQ) to 16-bit status (0 or 1) (F or Q)

Connection	Description
Input En (—▶)	Enable this function.
Input R (—○)	Reset output Q with a signal at input R (Reset).
Parameter	**Retentivity** set (on) = the status is retentive in memory.
Output Q	Q is switched on with a signal at input En and switched off with a low signal at input En.

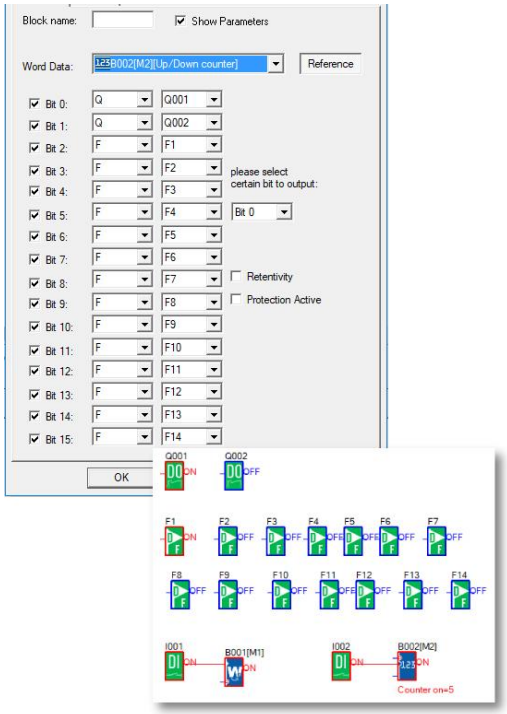


Example1:



Convert the AQ11 (2018) to Q1, Q2 and F1--F14(0000011111100010)

Example2:



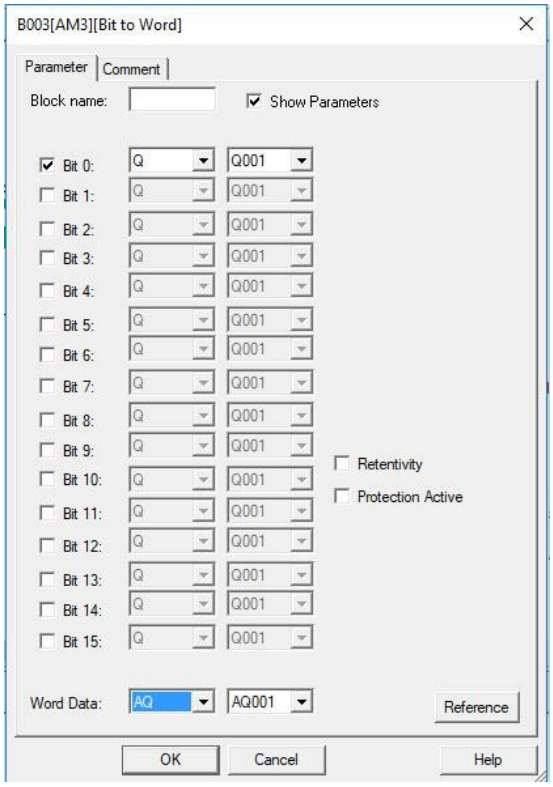
Convert the counter value (5) to Q1, Q2 and F1--F14(0000000000000101)

Bit to Word

Description of function

This special function is used to transfer the 16-Bit status (0 or 1) (F or Q) to word type data (AF or AQ).

Connection	Description
Input En (—▶)	Enable this function.
Input R (—○)	Reset output AQ with a signal at input R (Reset).
Parameter	**Retentivity** set (on) = the status is retentive in memory.
Output AQ	AQ will output the value of the block when En was activated.

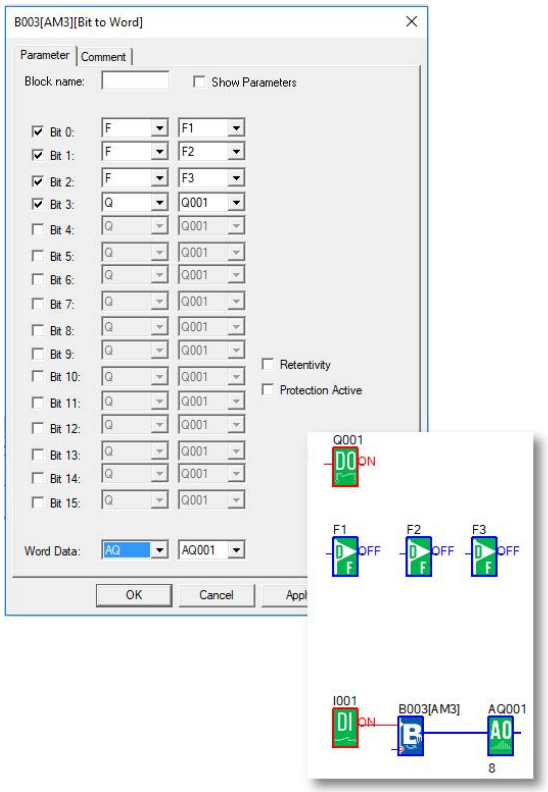


For example

Transfer the F1--F3 and Q1 status to the AQ001. F1 is saved in Bit0, F2 is saved in Bit1, F3 is saved in Bit2, Q1 is saved in Bit3.

F1	F2	F3	Q1	AQ1 value
0	0	0	0	0
1	0	0	0	1
0	1	0	0	2
1	1	0	0	3
0	0	1	0	4
1	0	1	0	5
0	1	1	0	6
1	1	1	0	7
0	0	0	1	8
1	0	0	1	9
0	1	0	1	10
1	1	0	1	11
0	0	1	1	12
1	0	1	1	13

0	1	1	1	14
1	1	1	1	15



NOTICE:

1. The Bit4--Bit15 was not ticked, they are all recognised as 0.

2. With such block, you can modify one bit of the word register in the slave devices together with the code 06/16 in Modbus network.

Device Reset

Description of function

This function block is used to reset the device (Ethernet modem or WIFI modem built-in) in the CPU if there is a trigger at the Trg pin. It can only be used in CPUs with Ethernet module or Wi-Fi module built-in.

Available in the following CPUs:

ELC series type:

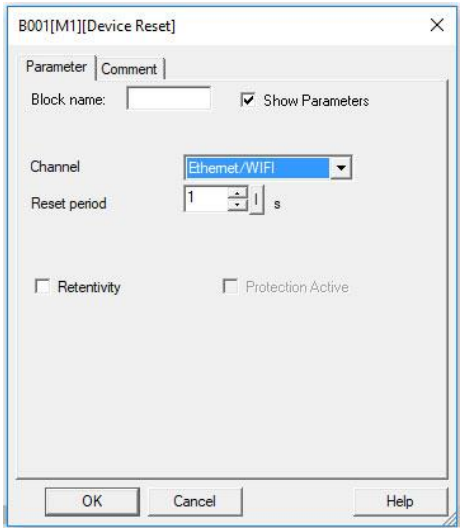
ELC-12DC-DA-R-N

ELC-12AC-R-N

ELC-22DC-DA-R-N

ELC-22AC-R-N

Connection	Description
Trg input (—●)	If there is a trigger at the Trg pin, then the Ethernet modem will be reset, The modem will be start-up again until the time "Reset period" reached.
Parameter	**Channel**: Ethernet **Reset period**:1—100s
Output Q	When the Ethernet module had been reset successfully, this block will output a short trigger.



Description of the function

In order to monitor the communication status of the Ethernet to see if it is normal or continuous (or avoid the Ethernet module is dead by unknown cause), sometimes we need reset the Ethernet module built-in in the CPU when the communication has failed or been a timeout.

Just one parameter to be set is the "Reset period" as follows:

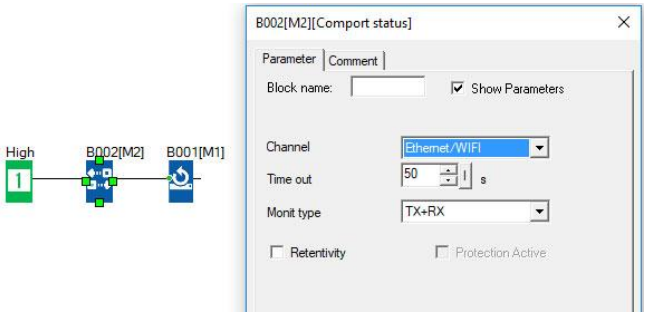
Here are two examples with detailed description on how to use these blocks in the program.

Example 1

Just as program below shows, after the CPU is running, we can push down the digital input1(just need a short trigger to reset the Ethernet modem), after the Reset period (here is 5s) is reached, the Ethernet modem will start up and this block will output a short trigger at the same time.

Example2

We also can use such "device reset" block along with the "Com status" function block together in the program, when there is no data transmission through the Ethernet port while the timeout period(the 50s) is reached, the com port status will output HI signal, and then the Device reset block would be enabled and the Ethernet module will be reset for the Reset period in the device reset block property dialogue box.



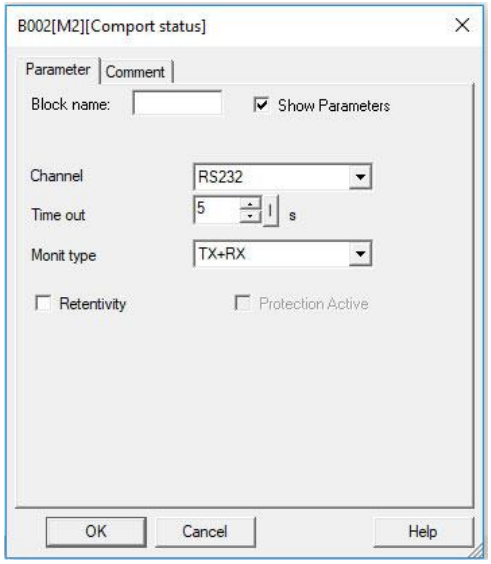
Comport Status

Description of function

This function block is used to monitor the communication status of the RS232 (programming port), RS485 port, Ethernet/WIFI port.

With the text message block, we can insert the com port status from such function block for displaying on the LCD.

Connection	Description
Input En (—▸)	Enable the function block if a HI level at En input pin and if a trigger from HI to LOW, the function would be disabled.
Parameter	**Channel**: RS232 RS485 Ethernet/WIFI **Timeout**: 1—100s **Monitor type**: TX RX TX+RX
Output AQ	When a HI level at the En pin and the timeout is reached the output will be HI, and it will be reset when the En is switched.



Parameters

Channel

RS232: This channel is the programming port; it can be used to monitor or display the communication status of the below accessories:

ELC-RS232 cable

ELC-USB cable

PRO-RS485 cable

ELC-MEMORY

ELC-Copier

RS485: This channel is the RS485 port, it can be used to monitor or display the communication status of the below accessories:

For Standard ELC-12 Series

ELC12-E-RS485

For ELC-12DC-DA-R-N-HMI CPU

EXM-E-RS485

For ELC-18/22/26 and PR-18/24/26 and PR Ethernet Series CPUs

PR-RS485

Ethernet: This channel is the Ethernet port; it can be used to monitor or display the communication status of the LAN port built-in in the CPU:

Timeout: 1—100s

Monitor type:

Tx: Data from CPU to external devices.

Rx: Data from the external device to CPU

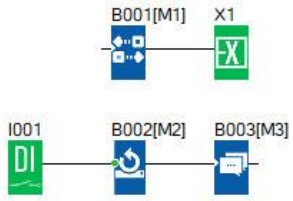
Tx + Rx: Data transmission between external device and CPU.

Description of the function

In order to monitor the communication status of the RS232, RS485 and Ethernet port, we can enable this function block. When the timeout period is exceeded, the block will output a high-level trigger.

How to insert the com port status to the text message for displaying?

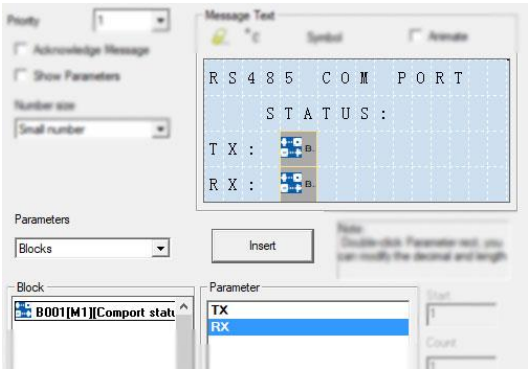
Put the "com port status" into the program.

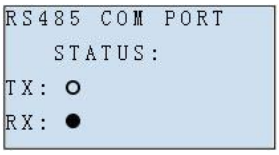


Select "BLOCKS" from the Parameters drop-down list on the Parameters tab.

1. Select TX or Rx and insert into the screen.

2. You can edit the text on the screen, such as TX: RX:





You can view the communication status on the LCD, even if there are no indicators on the ELC-RS232/ELC-USB/PRO-RS485.

If data is being transmitted, the status of the com port on the LCD will be flashing.

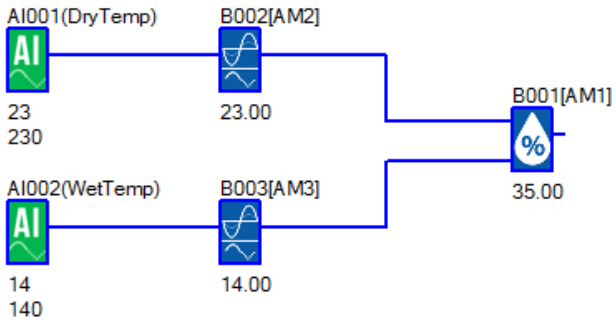
RH Math

Description of function

This block is used to calculate the Relative Humidity given the Wet and Dry bulb temperatures. For the calculation, the pressure is assumed to be 101.3 kPa (Kilopascals) or 1 Atmosphere. If required, this value can be changed to another or can be changed by refencing another block.

Connection	Description
Input Ax (Tdry) (—▶)	The analogue input signal of the Dry temperature. (0 – 10 V corresponds to 0 – 1000 internal PLC values).
Input Ay (Twet) (—)	The analogue input signal of the Wet temperature. (0 – 10 V corresponds to 0 – 1000 internal PLC values).
Parameter	The pressure at which the temperatures are measured. Default is 1 Atmosphere (101.3 kPa). This value is changeable to a fixed value or variable if referenced to another block.
Output AQ	The calculated Relative Humidity. Range of AQ is -32,768 to 32,767

Example



With a measured Dry temperature of 23 °C and Wet temperature of 14 °C at 101.3 kPa gives a Relative Humidity of 35%.

6.1.5. App Functions List – AP

Cam Control

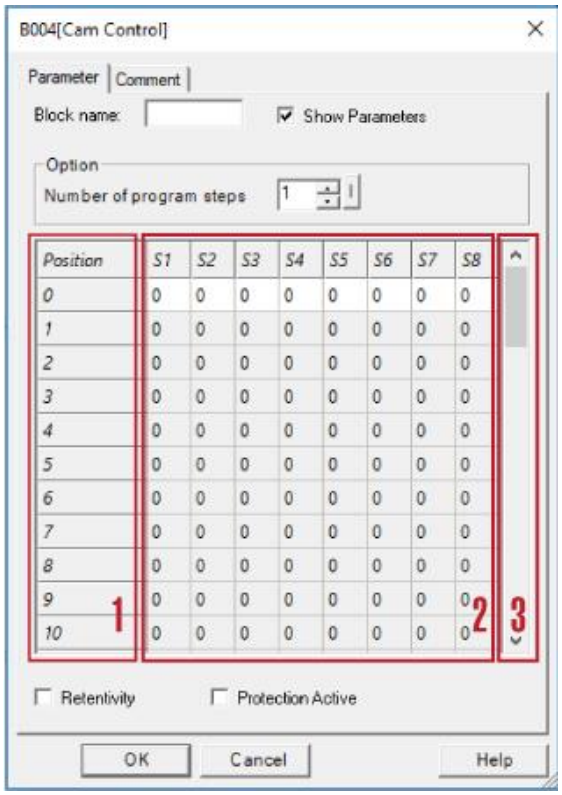
Description of function

The cam programmer function Cam Control is used to control a set of 8 built-in cam wheels.

On its 8 outputs (representing the 8 wheels), the function provides the state corresponding to the current position of the shaft wheels. The cam configuration can be set for each position, the output state is adjustable. Once the maximum value has been reached, the cam restarts from its initial position (output returns to 0).

Connection	Description
Input Forward (—▸)	MOVE FORWARD is the input which is used to control cam progress; it moves one step forward at each rising edge (digital status is changed from 0 to 1).
Input Reverse (—◂)	MOVE BACKWARD is the input which is used to control backward cam movement; it moves one step backwards at each rising edge (digital status is changed from 0 to 1).
Input R (—○)	RESET (initialization): When this input is active, the cam is replaced to its initial position: the POSITION output will be forced to 0.
Parameter	From the property dialogue box, you may adjust:
	The number of program steps: Its value is between 1 and 50. **Output status** [1..8]: for each position of the shaft.
Outputs Q 1-8	The status of the 8 outputs corresponds to the current position of the shaft (representing the 8 wheels).
Output Position	An analogue output corresponding to the current cam position (0 to 49).

The following figure shows an example of a part of the parameters window:



1. Position of the cam

2. Output configuration for each cam position

3. Number of selected steps

NOTICE:

1. The FORWARD input takes priority over the BACKWARD input.

2. If the FORWARD and REVERSE inputs are not connected, they are set to inactive.

When selected, the "Retentivity" enables the current value of the timer to be retrieved following a power failure.

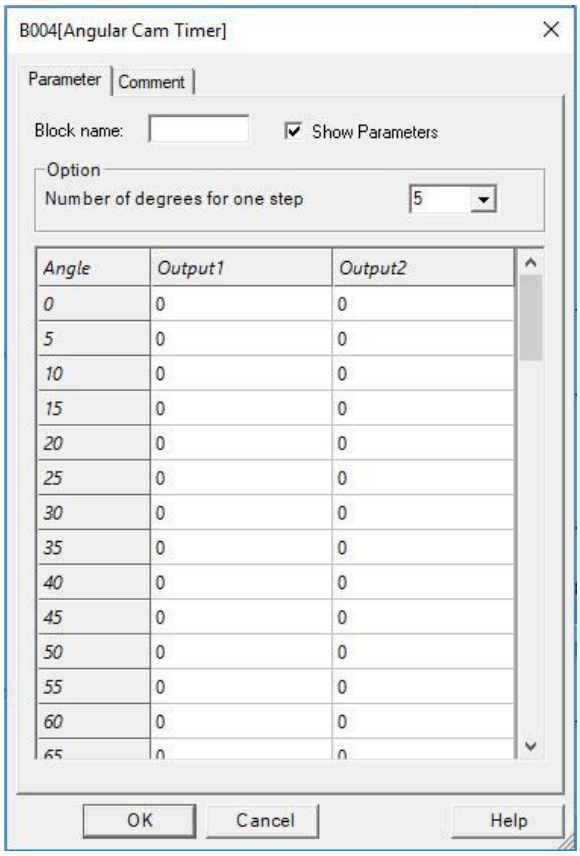
Angular Cam Timer

Description of function

This function block is used to describe the operation of a cam timer based on the angle made by the cams as the analogue input. The number of steps can be selected, and each step corresponds 2 configurable outputs.

Connection	Description
Input Validation (—▸)	Enables the function. If this input is not activated, the function remains inactive. Activated implicitly if it has not been connected.
Input Angle	Timer command input (from 0° to 359°). The outputs vary according to this value and the OUTPUT STATE parameter.
Parameter	**Number of degrees**: Equivalent to the wheel step number (2 to 72 steps of 5° to 180°). **Output states**: Table is listing the output states for each position. These states can be modified by left clicking in the corresponding boxes.
Output Out1	Output 1 is related to the value in the Out 1 column in the output state table. If the value of the angle in the angle input is higher or the same as a value N in the angle column in the output state table and less than the value N+1 in the table, the value of the corresponding Out 1 column is copied to Out 1 (1 => output at ON, 0 => output at OFF).

Output Out2	Output 2 is related to the value in the Out 2 column in the output state table. If the value of the angle in the angle input is higher or the same as a value N in the angle column in the output state table and less than the value N+1 in the table, the value of the corresponding Out 2 column is copied to Out 2 (1 => output at ON, 0 => output at OFF).

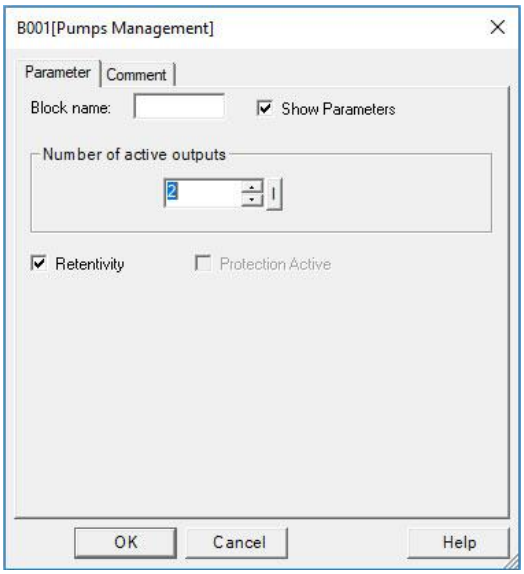


Pump Management

Description of function

Application-specific function: pumps management: (tank management with circular pump changeover).

This function is used to set to ON a maximum of four digital outputs which can be activated (Output 1 ... Output 4). This number is equal to the maximum number of digital inputs (from 2 to 4) in the ON state. In addition, the outputs set to ON are selected so that in the event of prolonged operation, each output will have been set to ON the same number of times.



The ON duration of the outputs is set to equal values by applying the following technique:

As the number of ON inputs increases, the outputs changing to ON are those following the order of the pilot output number: 1 for Out1, 2 for Out 2, 3 for Out 3 and 4 for Out 4. For example, if the "outputs controlled" parameter has the value 4, if pilot output number has the

value 3 and just one input is set to ON, only the Out 3 output will be set to ON. As soon as two inputs change to ON, the Out 3 output remains at ON and the Out 4 output changes to ON. As soon as a third input changes to ON, the Out 3 and Out 4 outputs remain ON and Out 1 changes to ON.

As soon as the number of outputs decreases, the outputs changing to OFF will be those which have been in the ON state the longest. As soon as one output changes to OFF, pilot output number takes the value of the output number after the output(s) which has (have) just been set to OFF. To complete the above example, as soon as one input changes to OFF, the Out 3 output changes to OFF and PILOT output number displays the integer value 4.

The Parameters tab in the property box contains the number of outputs which may change ON depending on the number of inputs which are set to ON. The values of this parameter are fixed at 2, 3 or 4.

If the value of the parameter is fixed at 2, only the Out 1 and Out 2 outputs are used and may, therefore, change to ON. In this case, if more than two inputs change to ON, the Out 1 and Out 2 outputs remain at ON and the Out 3 and Out 4 outputs remain fixed at OFF.

If the value of the parameter is fixed at 3, only the Out 1, Out 2 and Out 3 outputs are used and may, therefore, change ON. In this case, if four inputs change to ON, the Out 1, Out 2 and Out 3 outputs remain at ON and Out 4 remains fixed at OFF.

If the value of the parameter is fixed at 4, only the Out 1, Out 2, Out 3 and Out 4 outputs are used and may, therefore, change to ON.

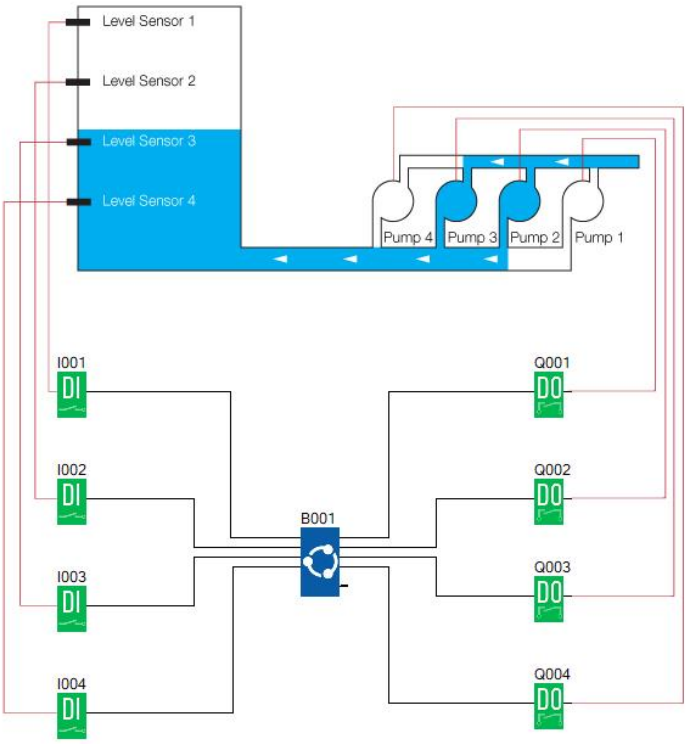
All inputs which are not connected have the value OFF.

When the program is initialized, the pilot output number is fixed at 1.

The Parameters tab contains the default checkbox which re-initializes pilot output number to 1 (and defines the first output activated when the first input changes to 1) after a controller power failure.

Example of use:

Filling a tank with a group of four pumps operating in parallel. The operating duration of each pump is the same.



The "number of outputs controlled" parameter is fixed at 4.

On initialization, pilot output number has the value 1. On initialization, if the tank is in the state indicated and if a sensor above the water is in the ON state when the user program is executed, the INPUT 1 and INPUT 2 inputs are ON, INPUT 3 and INPUT 4 are OFF and Out 2 and Out 3 are ON.

Assuming that the tank is full, Input 2 changes to OFF and Out 1 changes to OFF. Pilot output number indicates the value 2.

Assuming that the tank is empty, Input 2 changes back to ON, Out 3 changes to ON and Out 2 remains ON.

Assuming that the tank refills, Input 2 changes back to OFF, Out 2 changes to OFF and pilot output number indicates the value 3.

Assuming that the tank continues to refill, Input 1 changes to OFF, Out 3 changes to OFF and pilot output number indicates the value 4.

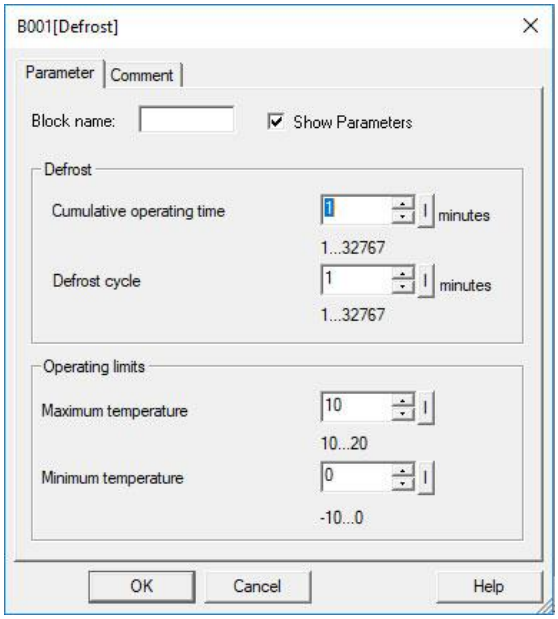
Defrost

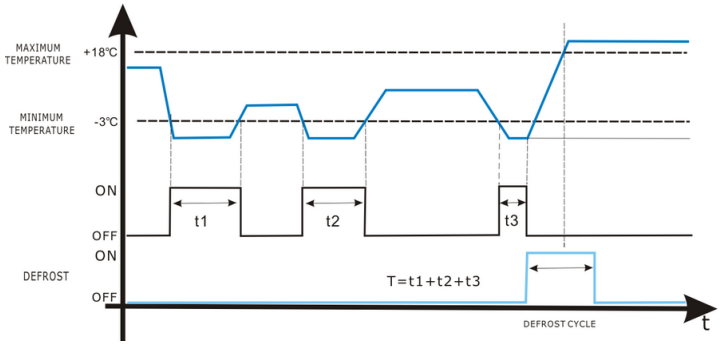
Description of function

The defrost output changes to ON when the input temperature is less than the minimum temperature for a time T (T being the cumulative duration of passages below the minimum temperature). If the temperature rises to higher than the maximum temperature during the defrost cycle, the defrost output reverts to OFF even if it has not finished. This output can be triggered and stopped by means of the corresponding inputs.

Connection	Description
Input Validation (—▸)	Function validation input. The function remains inactive for as long as this input is not activated. Validation is implicitly active if it is not connected.
Input Temperature	Air temperature in °C*100 (-32768°C to 32767°C).
Input Manual defrost On	Sets the defrost output to OFF (Priority stop).
Parameter	**Cumulative operating time**: Time T, in minutes, at the end of which the function triggers defrosting (1 to 32767). **Defrost cycle**: Duration of defrosting in minutes (1 to 32767).

	Maximum temperature: Temperature in °C above which defrosting is stopped (10°C ... 20°C). **Minimum temperature**: Temperature in °C below which the time T is measured (-10°C ... 0°C).
Output DEFROST	The defrost output is at ON when the cumulative operating time has elapsed.
Output cumulative time	Measured duration, in minutes, when the temperature is less than the minimum temperature or duration of the current defrost cycle.





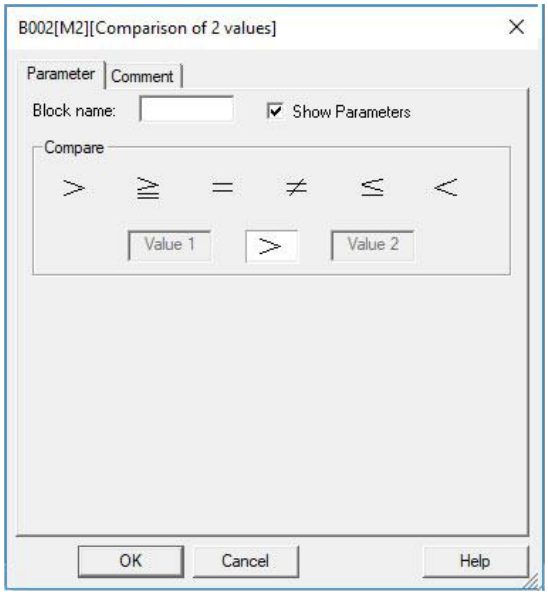
Comparison of Two Values

Description of function

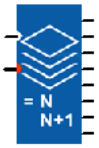
The COMPARE function is used to compare two analogue values.

Connection	Description
Input En (⟶)	Enable function Discrete-type input
Input Value 1 (⟶)	Integer-type input
Input Value 2 (⟶)	Integer-type input
Parameter	The comparison operators that can be chosen from the Parameters window are: Symbol Description > Greater than. ≥ Greater than or equal to. = Equal to. ≠ Different. ≤ Less than or equal to < Less than
Output Q	If the VALUE 1 or VALUE 2 input is not connected, the value is set to 0. The output is active if the result of the comparison between VALUE 1

	and VALUE 2 is true and if the En input is active or not connected. The output does not change state if the En input changes from Active to Inactive state.



Multi-Compare

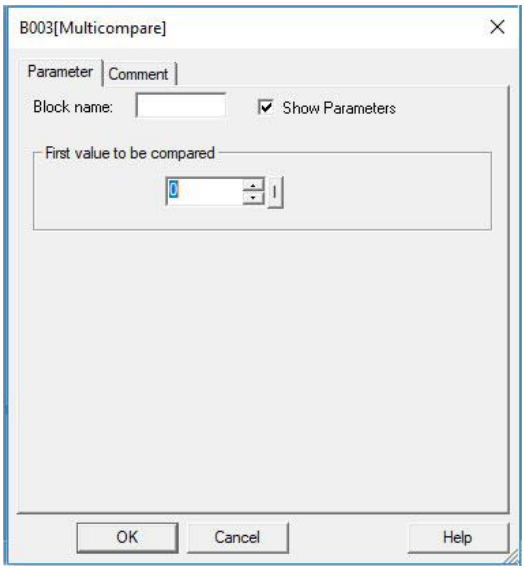


Description of function

This function is used to activate the output corresponding to the value present on the "Value" input.

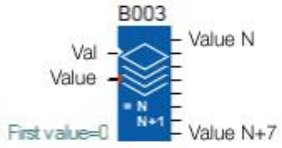
Connection	Description
Input Validation (—▷)	Function validation input. Until this input is activated, the function remains inert. Validation is active implicitly if it has not been connected.
Input Value (—▶)	Value to be compared.
Outputs Value N	**VALUE N:** Output ON if Value = Value N. **VALUE N + 1**: Output ON if Value = Value N + 1. **VALUE N + 2**: Output ON if Value = Value N + 2. **VALUE N + 3**: Output ON if Value = Value N + 3. **VALUE N + 4**: Output ON if Value = Value N + 4. **VALUE N + 5**: Output ON if Value = Value N + 5. **VALUE N + 6**:

	Output ON if Value = Value N + 6. **VALUE N + 7**: Output ON if Value = Value N + 7.



Operation:

The comparison value (Value N) can be configured. It must be between 0 and 32760.



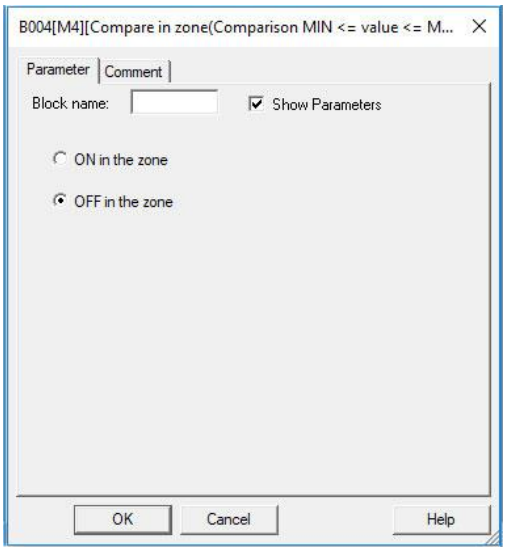
Compare in Zone

Description of function

The COMP IN ZONE comparison function is used to compare one value between two set points (the MIN and MAX values of the zone).

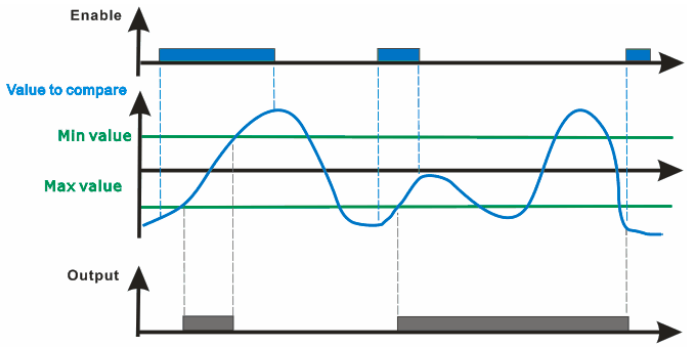
Connection	Description
Input En (⟶)	Discrete enable function input; this input is Active if it is not connected.
Input Value to Compare (⟶)	The value is used to be compared with the min and max value; whose type is Integer.
Input Min Value	A min value input, whose type is Integer
Input Max Value	A max value input, whose type is Integer
Parameter	From the Parameters window, you can select the state of the output according to the result of the comparison: ON in the zone: the output will be active. If the input value is between the two set points (MIN and MAX), OFF in the zone: the output will be inactive if the input value is between the two set points (MIN and MAX). If Min is greater than Max, then for: ON in the zone: the output always remains inactive,

	OFF in the zone: the output always remains active.
Output Q	The output indicates the result of the comparison when the En input is active. The output does not change state when the En input is inactive.



Comparison Function

The diagram below shows the different states the output can take, depending on the input value to compare and the enable input:

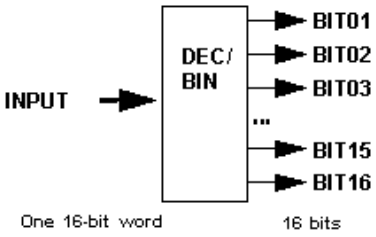


Conversion Word to Bits

Description of function

When the "En" is high, The DEC/BIN function breaks down an integer (16-bit) type input into 16 bit-type outputs.

Illustration:



Inputs/Outputs

This function supports 1 integer type 16-bit input:

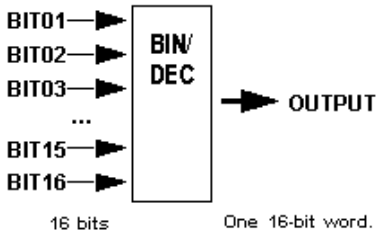
This function supports 16 discrete outputs: BIT01 (least significant byte) ... BIT16 (most significant byte).

Conversion Bits to Word

Description of function

When the "En" is high, the BIN/DEC function produces a 16-bit integer-type output from 16 inputs of the following type: Bit

Illustration:



Inputs/Outputs

This function supports 16 discrete inputs: BIT01 (least significant byte) ... BIT16 (most significant byte).

This function supports one 16-bit integer-type output.

De-Multiplexer

Description of function

This function demultiplexers integers. It is used to route the input value onto one of the 4 OUTPUTS on each rising edge of the VALIDATION input.

A VALUE copied to an output does not revert to 0 when a VALUE is written to another ADDRESS.

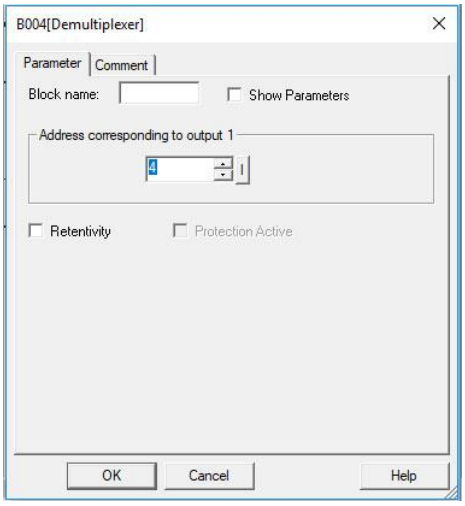
The BASE ADDRESS parameter allows several blocks to be used at the same time to multiple outputs.

The Parameters tab contains:

BASE ADDRESS: Contains the address of the ADDRESS 1 output.

Retentivity: Chooses whether or not the function is reinitialised if the controller power supply is disconnected.

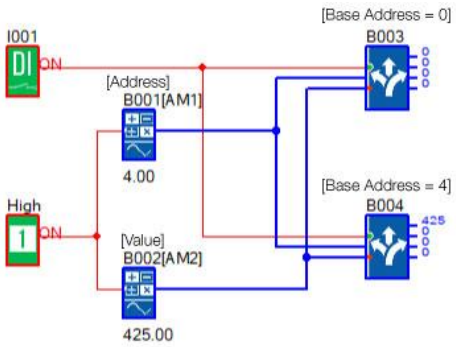
When they are not connected, the ADDRESS and VALUE inputs are set to zero.

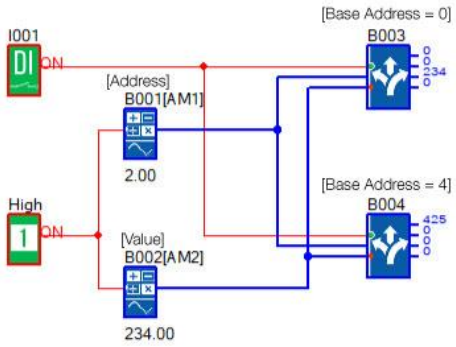


Example:

When the BASE ADDRESS parameter contains the value 0 these outputs have addresses 0, 1, 2, 3 respectively, and in this case, if the ADDRESS input equals 2 the VALUE will be copied to the third output.

If a second block is being used, 8 outputs can be demultiplexed by putting the value 4 as the BASE ADDRESS in the second block and connecting the VALIDATION and ADDRESS inputs to the same source.





Multiplexing

Description of function

The MUX function carries out two input channel multiplexing on the OUTPUT.

Connection	Description
Input CHANNEL A	This is the multiplexer input A; whose type is an integer.
Input CHANNEL B	This is the multiplexer input B; whose type is an integer.
Input SELECTION	This input is used to choose the input channel to apply to the output.
Output AQ	This is the multiplexer output. This value depends upon the state of the **SELECTION** input. If the **SELECTION** input is: inactive: the OUTPUT corresponds to CHANNEL A, active: the OUTPUT corresponds to CHANNEL B.

NOTICE:

1. If the SELECTION input is not connected, then it is considered to be inactive.

2. If CHANNELS A or B are not connected, then they are set to 0.

Multiplexer

Description of function

This function multiplexes the WORD inputs. It is used to route the value of one of the inputs selected by the ADDRESS input to the output. The input is routed to the output on each rising edge of the VALIDATION input.

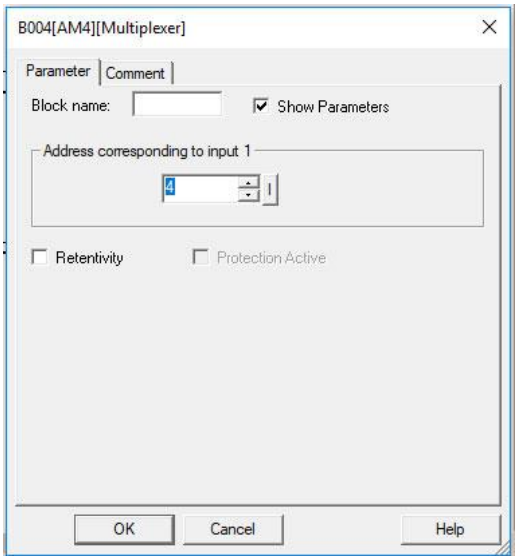
The BASE ADDRESS parameter allows several blocks to be used at the same time to multiple inputs.

The Parameters tab contains:

BASE ADDRESS（corresponding to Input 1): Contains the address of the INPUT 1 input.

Retentivity: Chooses whether or not the function is reinitialized if the controller power supply is disconnected.

When they are not connected, the digital input is in the OFF state and the WORD inputs contain 0.

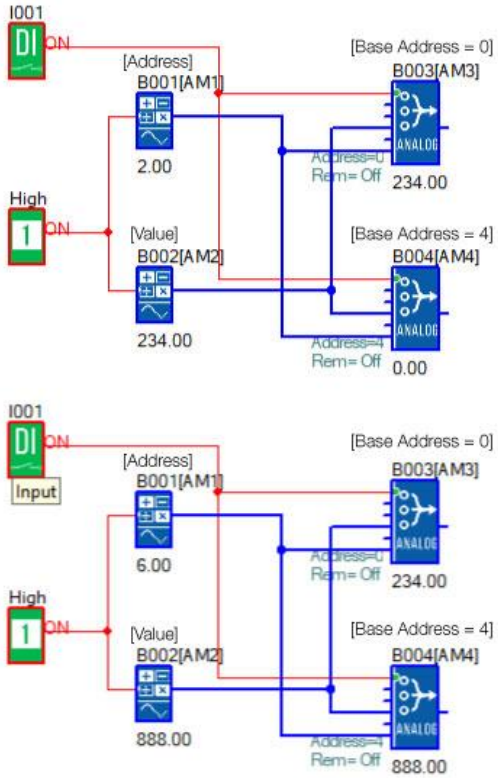


Example:

When the BASE ADDRESS parameter contains the value 0 these inputs have addresses 0, 1, 2, 3 respectively, and in

this case, if the ADDRESS input equals 2 the VALUE of the third input will be copied to the output.

If a second block is being used, 8 inputs can be demultiplexed by putting the value 4 as the BASE ADDRESS in the second block and connecting the VALIDATION and ADDRESS inputs to the same source.



Square Root

Description of function

This function is used to calculate the square root of the number present as an input with accuracy to two decimal points.

Connection	Description
Input Validation (—▸)	Function validation input. Until this input is activated, the function remains inert. Validation is active implicitly if it has not been connected.
Input Calculation	The value must be between 0 and 32767.
Outputs AQ	Calculation output: The result is presented in the format "Root" x 100.

Operation:

Example: When X = 20000 => Root of X = 141.42. The value read as an output of the function is 14142.

If an input number is negative, then the result is 0.

Performance:

The calculation is accurate to 0.01 either way.

Sin & Cos

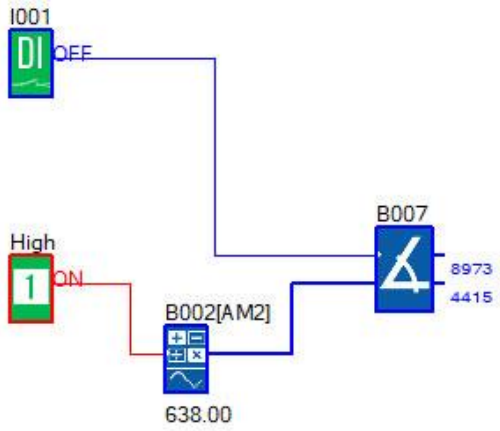
Description of function

This function is used to calculate the cos and sin of an angle between 0° and 90°.

Connection	Description
Input Validation (—▸)	Function validation input. Until this input is activated, the function remains inert. Validation is active implicitly if it has not been connected.
Input Angle	Represents the angle in degrees. Its value must be between 0 and 900 for an angle between 0° and 90°.
Outputs Sin	The result of ("Angle" sin) x 10000.
Output Cos	The result of ("Angle" cos) x 10000

Performance:

The function calculates the Cosine and Sine to the nearest 0.0001 by rounding up or down as appropriate.



Sin (63°8) = 0.8973 and Cos (63°8) = 0.4415

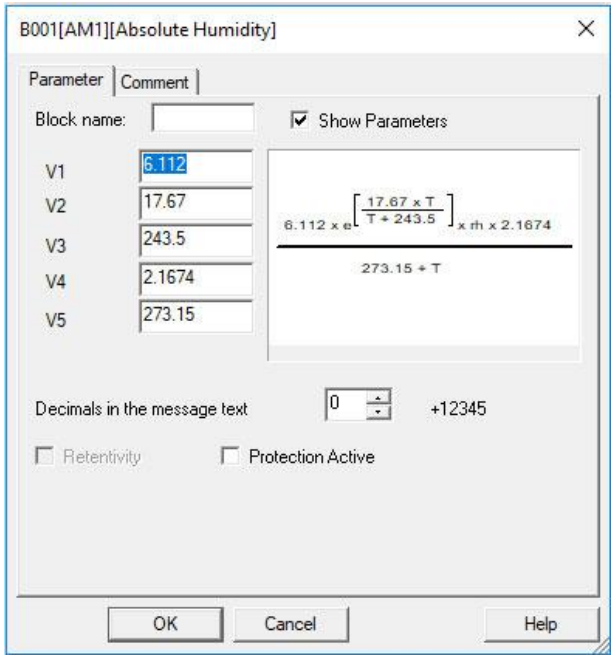
Absolute Humidity

Description of function

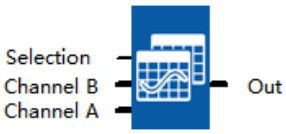
This function is used to calculate the absolute humidity.

Connection	Description
Input En (⟶)	Enable the function
Input T	Temperature
Output RH	Humidity
Output AQ	Absolute humidity

Calculation Rule:



Analogue Table



Description of function

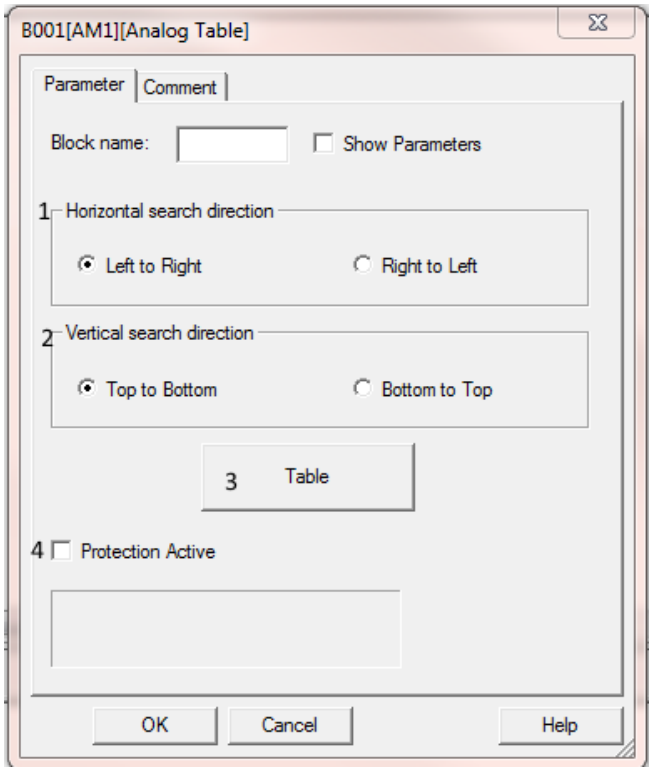
This function provides the ability to output an analogue value based on analogue inputs using a look-up table.

Connection	Description
Selection	When an input trigger is High, this block is enabled.
Channel B	**Analogue Input B value** **Range: -32,768 to +32,768**
Channel A	**Analogue Input A value** **Range: -32,768 to +32,768**
Out	Analogue Output value from table Range: -327.68 to + 327.67

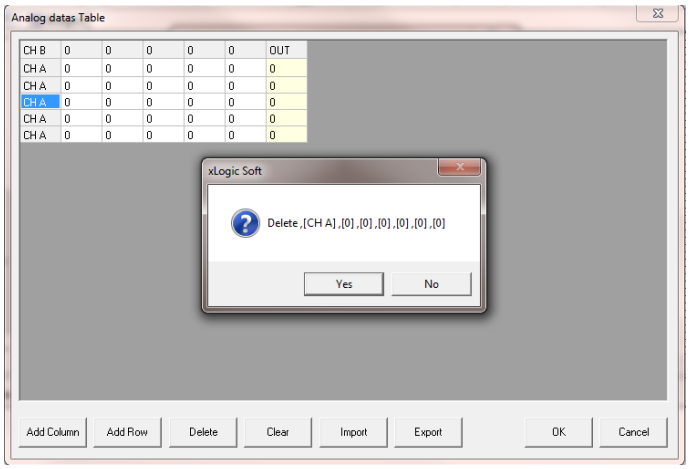
NOTICE:

The output value is multiplied by 100 to provide the actual output. So, a value 76.54 in an output column would result in an analogue value of 7,654 being output.

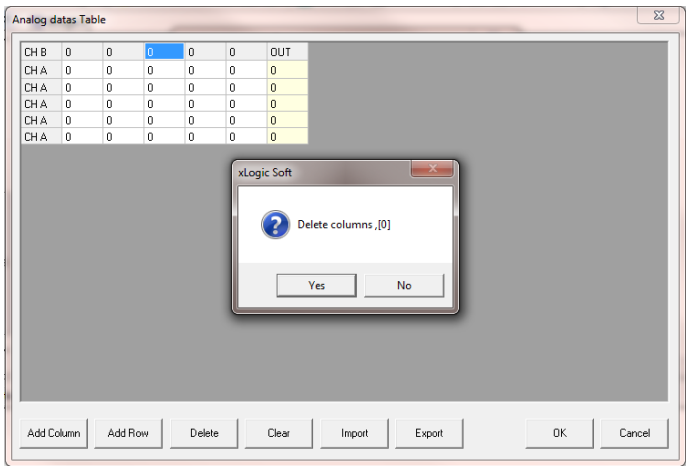
Parameter Description



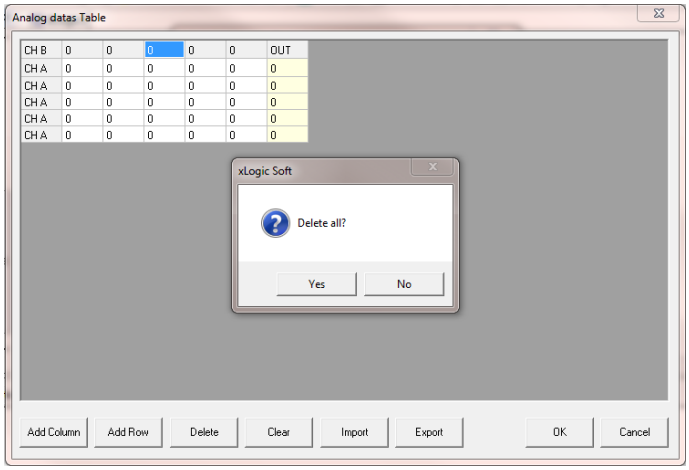
1) Horizontal Search Direction – Channel B Value Search
   a) Left to Right – Search for Channel B value starting from the left-hand column. Note: If the exact value of B is NOT found then the search will **STOP** at the next value which is **GREATER** than B.
   b) Right to Left – Search for Channel B value starting from the right-hand column. Note: If the exact value of B is NOT found then the search will **STOP** at the next value which is **LESS** than B.
2) Vertical Search Direction – Channel A Value Search
   a) Top to Bottom – Search for Channel A value starting from the Top row. Note: If the exact value of A is NOT found then the search will **STOP** at the next value which is **GREATER** than A.
   b) Bottom to Top – Search for Channel A value starting from the Bottom Row. Note: If the exact value of A is NOT found then the search will **STOP** at the next value which is **LESS** than A.
3) Table
   a) Add Column – Add a new column to the table
   b) Add Row – Add a new row to the table
   c) Delete – Select Row or Column to be deleted and press Delete button and confirm operation by clicking Yes or cancel by clicking No.
      i) Delete Row Example



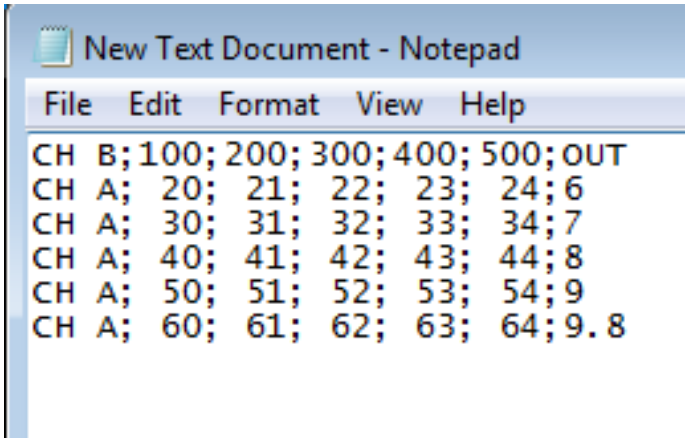
      ii) Delete Column Example



   d) Clear – Delete the whole table. Confirm with Yes or cancel with No.



e) Import – Used to import a CSV file of table data. Table MUST be delimited with a semi-colon between EACH data element.



f) Export – Used to export the current data and save it as a semi-colon delimited CSV file.

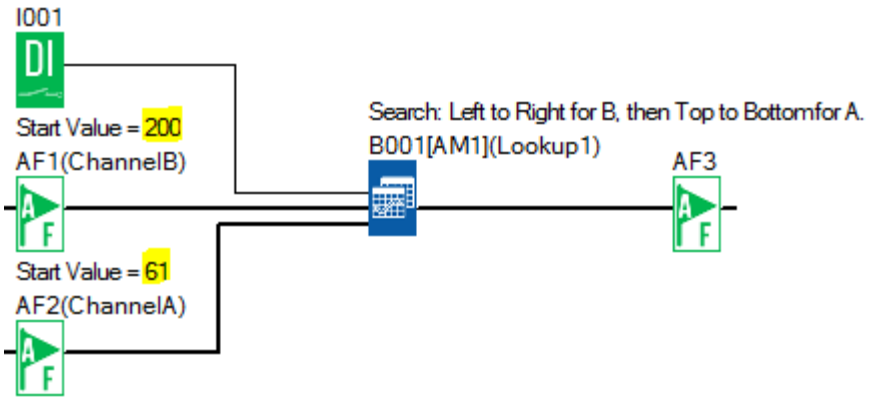
g) OK – To save changes and close table window.

h) Cancel – To close table window without saving changes.

4) Protection Active – If checked, data cannot be altered via the PLC front LCD screen during run time.

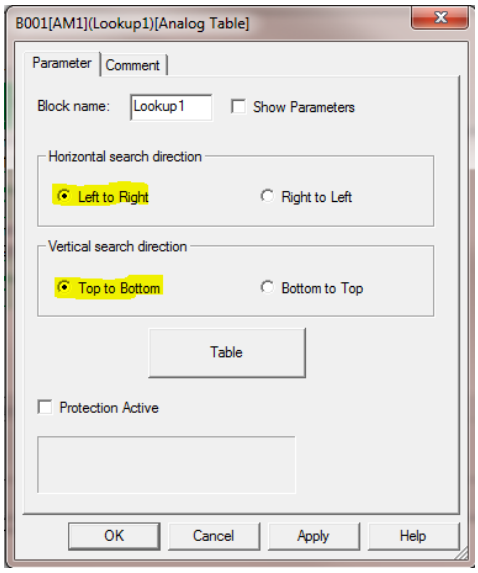
Example Use of Analogue Data Table

Program



In the above program, AF1 has a Start Value of 200 (Channel B) and AF2 has a Start Value of 61 (Channel A). When the digital input I001 is On (High), the analogue table will look-up the values, B followed by A, and output the table value, which will be stored in AF3.

Table Configuration



Analog datas Table

CH B	100	200	300	400	500	OUT
CH A	20	30	40	50	60	5
CH A	30	40	50	60	70	6
CH A	40	50	60	70	80	7
CH A	50	60	70	80	90	8
CH A	60	70	80	90	100	9
CH A	70	80	90	100	110	9.8

When DI1 = On, the table will be searched for the value of Channel B (200) from Left to Right and then for the value of Channel A (61) from Top to Bottom. In this example the value passed to AF3 will be 8 * 100 = 800.

Analog datas Table

CH B	100	200	300	400	500	OUT
CH A	20	30	40	50	60	5
CH A	30	40	50	60	70	6
CH A	40	50	60	70	80	7
CH A	50	60	70	80	90	8
CH A	60	70	80	90	100	9
CH A	70	80	90	100	110	9.8

Example 2

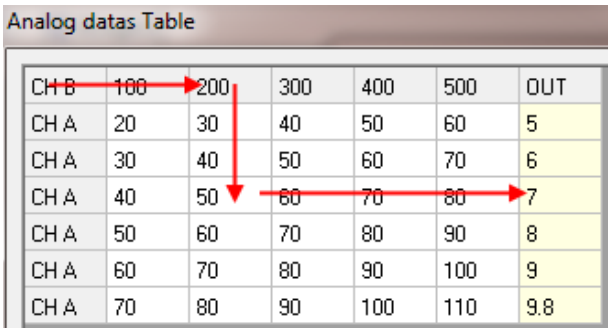
Search: Left to Right and Top to Bottom

B = 250 and A = 55

The search from Left to Right would stop at column headed 200 because this is the nearest value to 250 that is not HIGHER.

The search from Top to Bottom would stop at row with 50 because this is the nearest value to 55 that is not HIGHER.

The output value would be 7 * 100 = 700.



Example 3

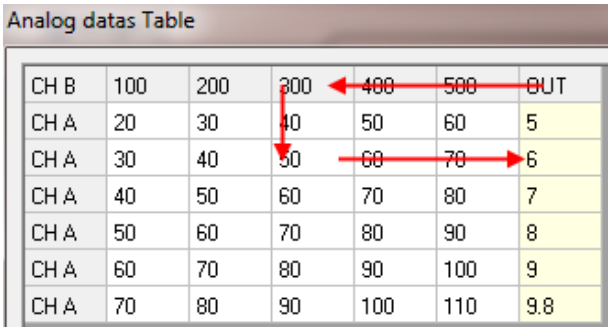
Search: Right to Left and Top to Bottom

B = 250 and A = 55

The search from Right to Left would stop at column headed 300 because this is the nearest value to 250 that is not LOWER.

The search from Top to Bottom would stop at row with 50 because this is the nearest value to 55 that is not HIGHER.

The output value would be 6 * 100 = 600.



Example 4

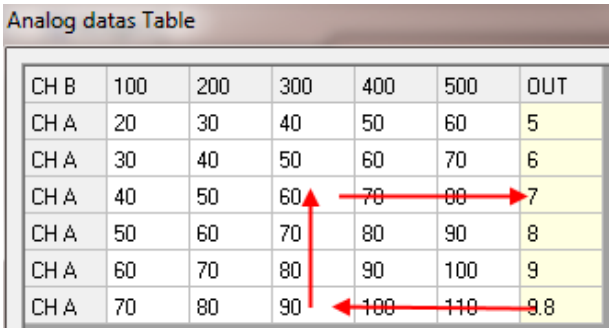
Search: Right to Left and Bottom to Top

B = 250 and A = 50

The search from Right to Left would stop at column headed 300 because this is the nearest value to 250 that is not LOWER.

The search from Bottom to Top would stop at row with 60 because this is the nearest value to 55 that is not LOWER.

The output value would be 7 * 100 = 700.



Example 5

Search: Right to Left and Top to Bottom

B = 680 and A = 167

The search from Right to Left would stop at column headed 500 because this is the LAST value in the row.

The search from Top to Bottom would stop at row with 110 because this is the LAST value in the column.

The output value would be 9.8 * 100 = 980.

Analog datas Table

CH B	100	200	300	400	500	OUT
CH A	20	30	40	50	60	5
CH A	30	40	50	60	70	6
CH A	40	50	60	70	80	7
CH A	50	60	70	80	90	8
CH A	60	70	80	90	100	9
CH A	70	80	90	100	110	9.8

General Data Latching Relay

Description of function

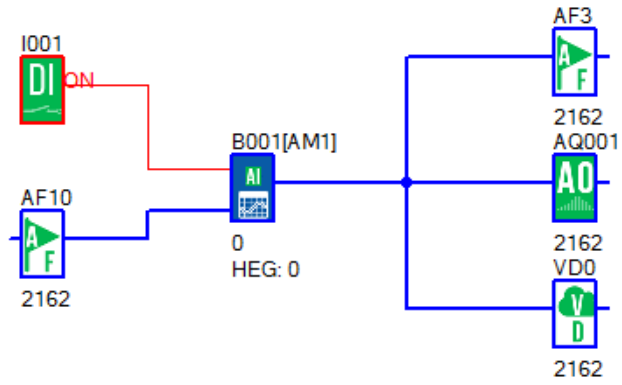
This special function provides the ability to save the process variable of an analogue input to memory and returns it as the analogue output, if a pre-set value is not selected as the output.

Connection	Description
EN	When a rising edge is detected at pin EN, the current Ax input value is latched and output OR the default value is output.
Ax	**Analogue Input Ax value** **Range:** -2147483648 **to** +2147483647
Out	Analogue Output value Range: -2147483648 **to** +2147483647

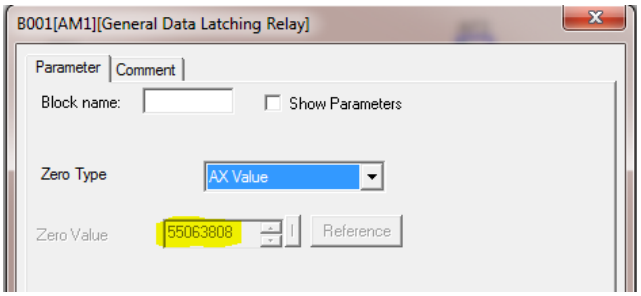
NOTICE:

If the expected value to be output is outside the range -32,768 to +32,767, then a VD type block should be used on the output.

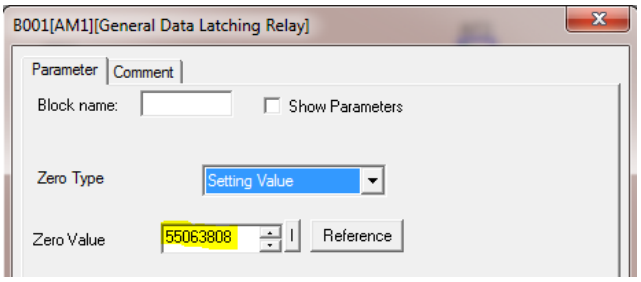
Example

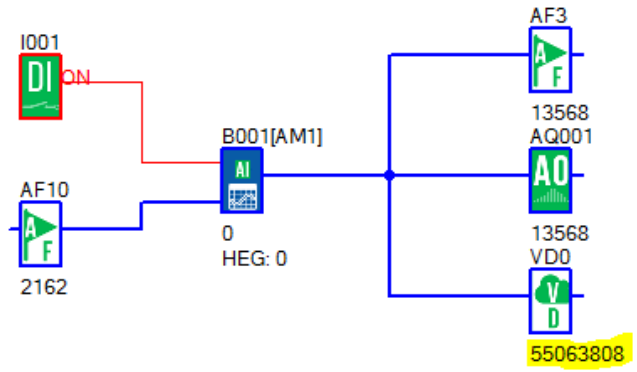


With 'Ax Value' selected the value of AF10 is passed by the block.



With 'Setting Value' selected, the Zero Value setting is passed by the block.





Free Protocol Communication

Description of function

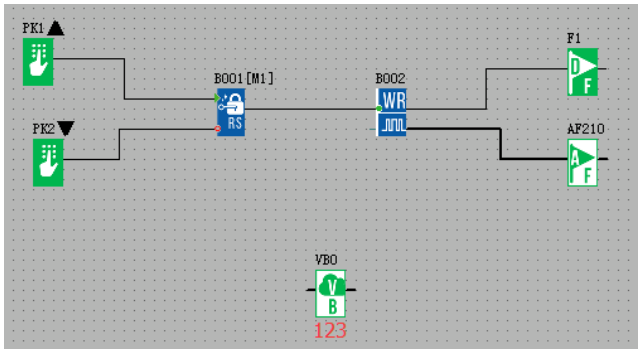
This special function block enables the use of a user defined communication protocol to communicate with multiple smart devices. The block can be used to either Transmit Only or Transmit and Receive.

Connection	Description
EN	When the input to pin EN is High the block is enabled.
Communication	**Comm Type:** **COM 0 (RS232/PRO-RS485)** **COM 1 (Expansion RS485)** **COM 2 (Built-in RS485)** **Ethernet: Sockets 1 to 8** **For Serial COM Ports Settings are:** **Bit per Second: 1200 to 19200** **Data Bits: 5 to 8** **Stop Bits: 1 to 2** **Parity: None, Odd, Even, Mark or Space** **For Ethernet, CPU is Client or Server** **CPU as Client – Configure remote server target in the Web Server Config page settings 1 to 8.** **CPU as Server – Configure target IP as the IP address of the Client.** **BOTH CPU and Client MUST be on the SAME subnet.** **Protocol: TCP or UDP**
Parameter 1	**This parameter page is to configure the CPU to SEND data as either String or HEX format.** **Enable – Check to enable CPU to Send data** **Cycle Send Delay – Time in x10 ms between transmissions** **String or HEX – Data format to be transmitted** **String Append – Chars to Add to end of message:** • **Nothing** • **₩r – Carriage Return (CR)** • **₩n – Line Feed (LF)** • **₩r₩n – CRLF** **Start Data – Characters to transmit at beginning of data string** **Reg Type – CPU registers to be sent:** • **None** • **F – Digital Flags** • **AF – Analogue Flags** • **VB – VB Block data** • **VW – VW Block Data** • **VD – VD Block data** **Reg Start – Start register to be used if something other than None was selected above.** **Reg Count – Number of consecutive registers to transmit.** **Data Word – Byte order of applicable data:**

	• **High Low** • **Low High** **Format – Format of data as Unsigned or Signed Integer** **Data – Word order of applicable data:** • **High Low** • **Low High** **Decimal – Number of decimal places for supported data formats, 0, 1, 2 or 3.** **End Data – Data to be appended to after main data** **Check Mode – Whether a checksum is required or not. Choice of None, CRC or Add.**
Parameter 2	**Timeout – Setting for receive communication timeout. Range 5 to 255 x100ms.** **String or HEX – Data format to be received** **Keyword – Expected message start string if required.** **Data Start Pos – Position of 1st character in received message that contains start of data** **Data Count – No. of characters to be received.** **Reg Type – Type of CPU register in which data is to be stored** **Reg Start – Start flag for received data** **Reg Count – No. of registers to be used to store received data** **Data Word – Byte order of applicable data:** • **High Low** • **Low High** **Data – Word order of applicable data:**
	• **High Low** • **Low High** **Check Mode – Whether a checksum is required or not. Choice of None, CRC or Add.**
Out 1	Digital Output – if set True/High then correct packet received otherwise it is set False/Low
Out 2	Status Code: 0 = Not triggered 1 = Response Timeout 2 = Response Correctly Received

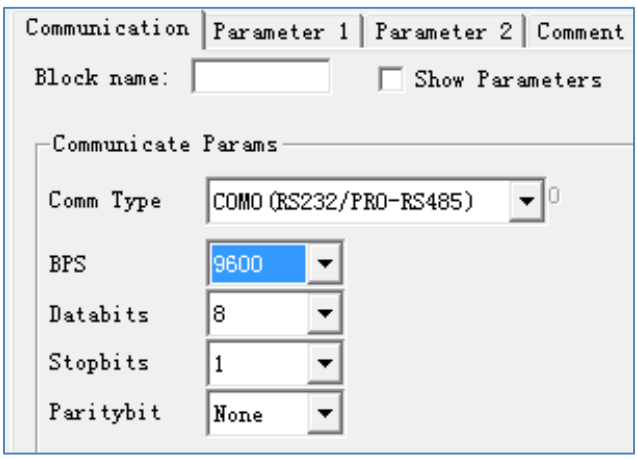
Example 1 – Transmitting Data

Program



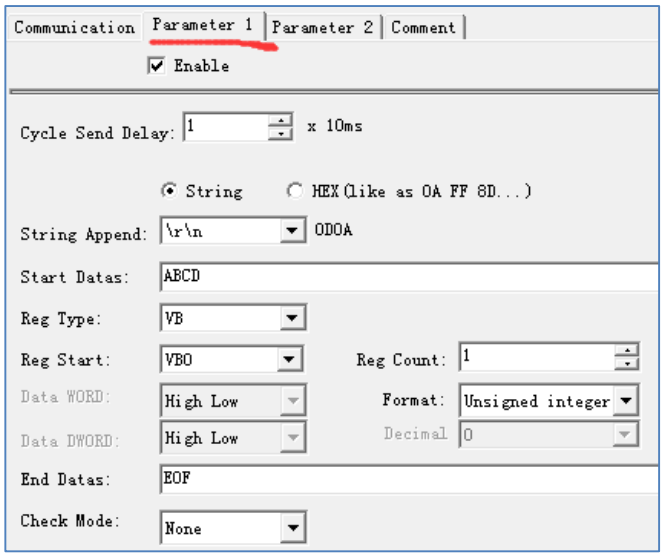
When Up arrow on CPU is pressed, transmitting begins and is stopped by pressing Down arrow on CPU.

Communication Settings



PLC sends data via COM 0.

Parameter Settings



Transmit the characters "ABCD" and the value in block VB0 and "EOF" (End Data) and a Carriage Return with Line Feed via COM 0.

The receiver will receive one frame of data, which is as follows:

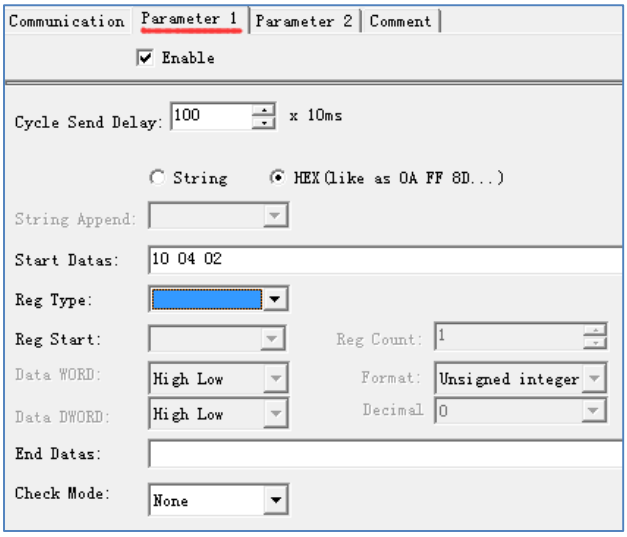
41 42 43 31 32 33 45 4f 46 0D 0A

The above data is the ASCII code representation of the string being sent.

41	42	43	44	31	32	33	45	4F	46	0D	0A
A	B	C	D	1	2	3	E	O	F	\r	\n

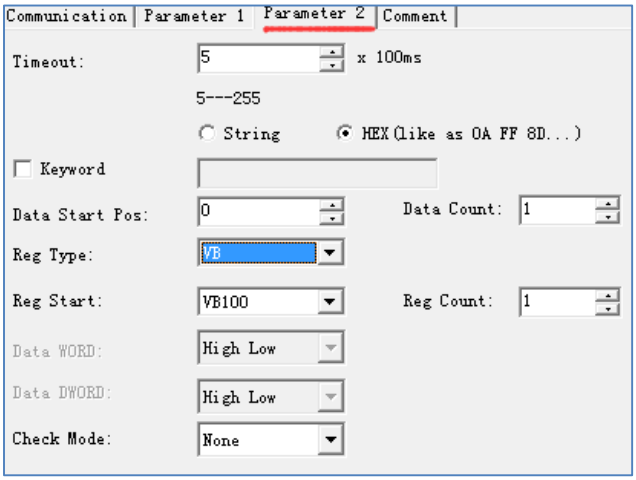
Example 2 – Transmit and Receive Data

Request the status of a model printer. We need to transmit the HEX characters 10 04 02 to the printer, so the Parameter 1 settings are as follows:



After receiving the request from the CPU, the printer will return one byte of data, which is to be stored in VB100.

Parameter 2 settings required are as follows:



General Arithmetic

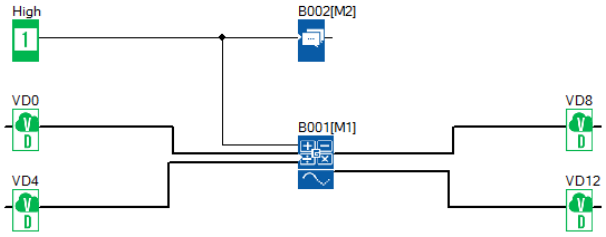
Description of function

This special function can be used perform one of four arithmetic functions on two values. The functions are Addition, Subtraction, Multiplication or Division.

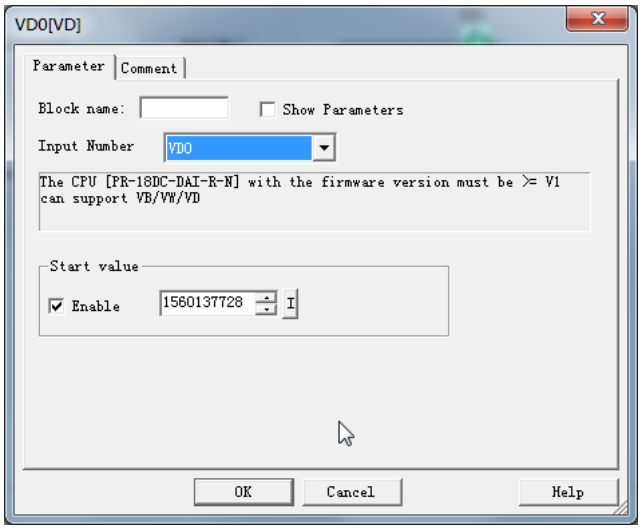
The data types of these two values can be Unsigned Char, Signed Char, unsigned Short, signed Short, unsigned Integer or signed Integer.

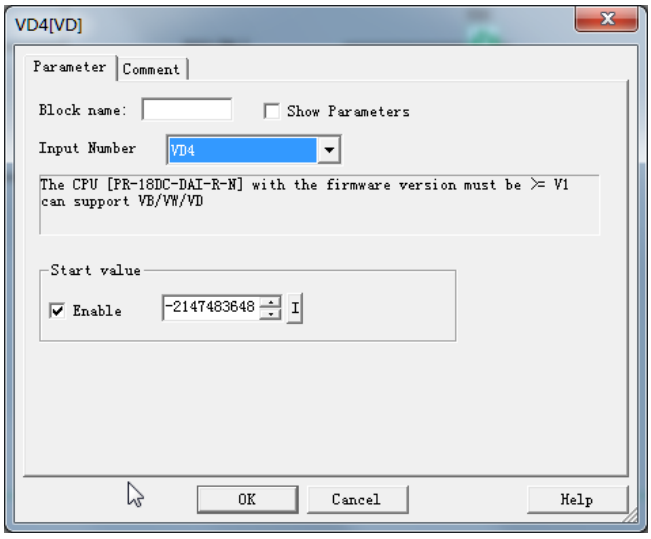
Connection	Description
EN	When the input to pin EN is High the block is enabled.
Value 1	**First analogue value for calculation.** **Range: -2147483648 to +2147483647**
Value 2	**Second analogue value for calculation.** **Range: -2147483648 to +2147483647**
Parameter	**Arithmetic Operand – +, -, x or ÷** **Data Type – Unsigned or Signed Integer** **Value to Output when EN is False – Zero or Last Value calculated** **Decimals in Message Text – 0, 1, 2 or 3. Used to place decimal point in value when displayed on the CPU LCD using Message Text block.**
Out 1	**Analogue output of calculation result.** **Range: -2147483648 to + 2147483647**
Out 2	**Analogue output of status code:** • **0 = No error** • **1 = Division by 0** • **2 = Calculation Overflow**

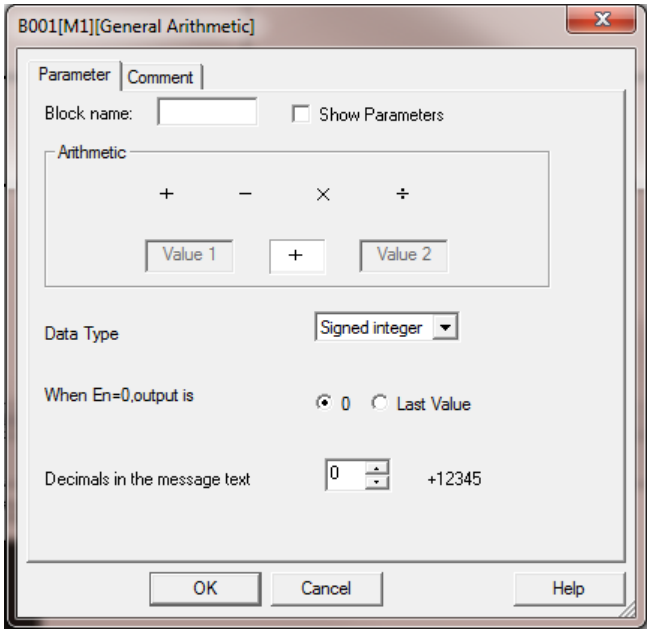
Program



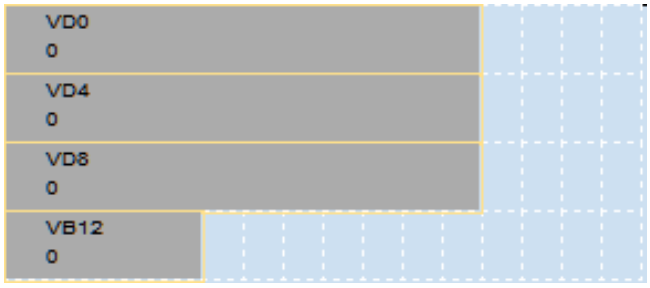
Settings for VD blocks and General Arithmetic block



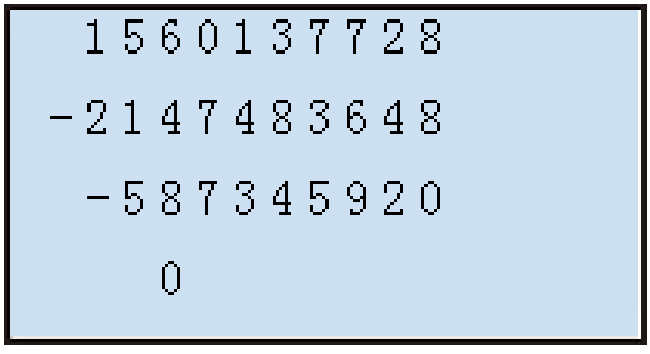




The settings in the Message Text block are:



When the program is downloaded to the CPU and ran, the PLC LCD will show the following information on its LCD panel:



6.1.6. Library Function (LIB)

The LIB Function allows you to create libraries or subroutines, which can be used in other programs.

To create a new library, select 'File….New…Library diagram' from the main menu bar.

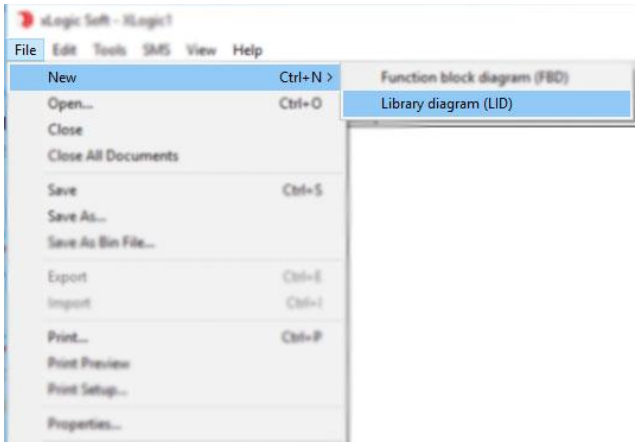


Figure 33 - New Library Diagram (LID)

A new workspace will open, which has 16-input connections and 8-outputs connections. Simply drop the blocks onto the work area that you wish your library function to use and connect them together as you would for a standard program.

For example, you may need something like this 2-Button Start with Inhibit, which switches on two outputs after two push button inputs have been held on for 5-seconds. Pushing the buttons again has no effect as a latching relay inhibits a restart until the Reset input has been switched On. If you need this feature multiple times in your program, you will see it uses a lot of space and may become a little untidy to read, the answer is, save it as a Library and reuse the Library block as often as you need!

The above scenario can be achieved as follows:

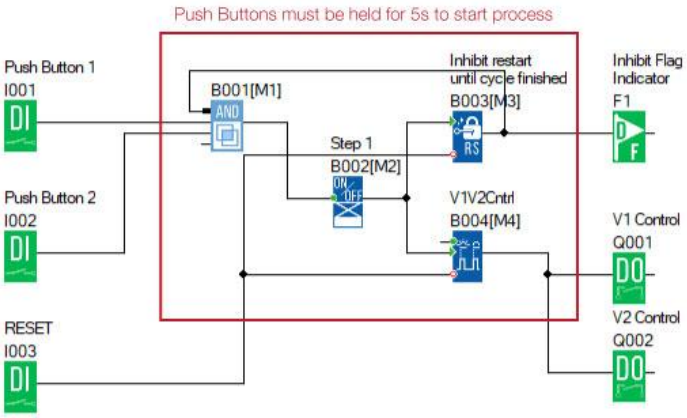


Figure 34 - Library Example Code

The section circled in Red contains the logic blocks to achieve the control scenario required – These would be used to create the Library function:

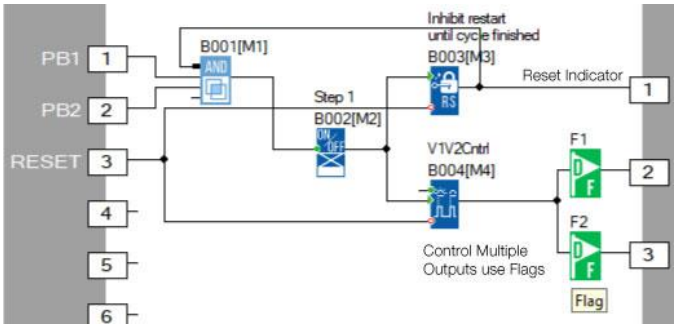


Figure 35 - LID Blocks

As you can see from the above, a block can only be connected to ONE output tab in a Library, so we have had to use two digital flags for the connection from B004 as we need two outputs to be On when it is On.

Library Function Options

There are a number of other features you can configure when creating your LID blocks, from password protection to naming your tabs.

LID Password Protection

As with all of your xLogicsoft programs, you are able to password protect your LID too. Simply select 'File…Properties' from the main menu or click on the 'Page Layout' icon:

Then select the 'Library' tab, tick the 'Program password' box and enter the new password and repeat it. You only need to complete the Old password field if you are changing the password!

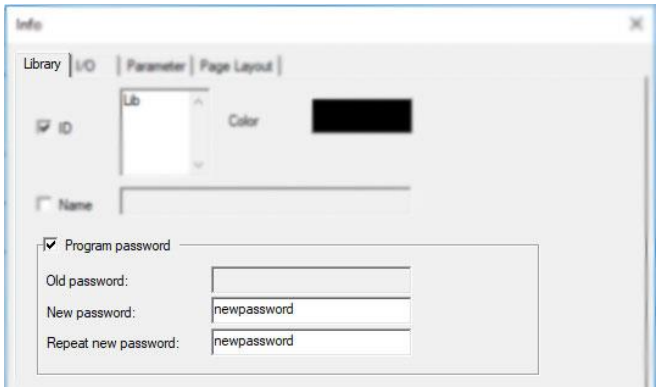


Figure 36 - LID Password Protection

LID Tab Naming

It is a good idea to name the input and output tabs of your LID block so that when you come to use it in a program, you will be reminded of what should be connected to each tab. There are two ways in which this can be achieved.

Method One

Select 'File...Properties' from the main menu or click on the 'Page Layout' icon:

Then select the I/O tab and enter the names that you wish to assign to each tab:

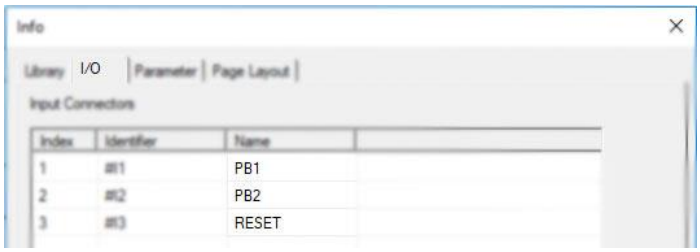


Figure 37 - LID Input Tab Naming

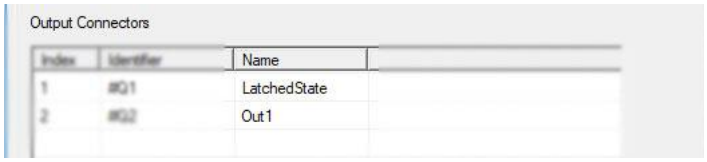


Figure 38 - LID Output Tab Naming

Method Two

Alternatively, you can right-click on a tab and select 'Properties' and a window will open enabling you to configure just that pin:

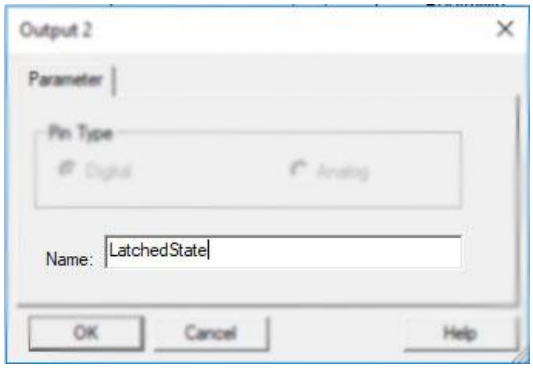


Figure 39 - LID Single Tab Naming

You need to repeat this process for each pin you wish to name.

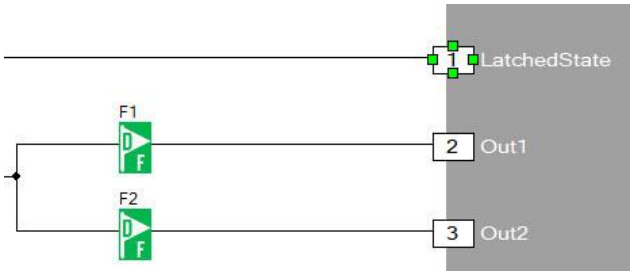


Figure 40 - LID Output Tabs Named

LID Customisation

You can customise your LID block further if you wish by adding your own ID, block image and size. Once saved, every time you use your LID in a program, it will have these properties rather than the standard xLogicsoft defaults.

Select 'File...Properties' from the main menu or click on the 'Page Layout' icon.

Select the 'Library' tab and customise your options:

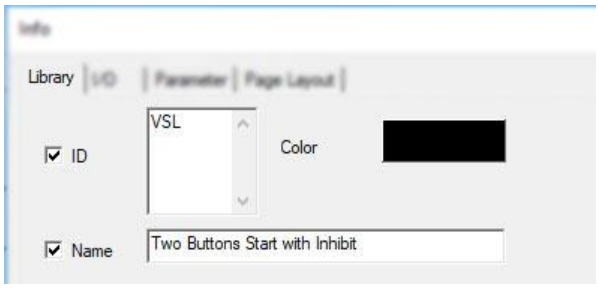


Figure 41 - LID ID Change

- ID – Tick the box and enter the ID you wish your LID to show when used in a program. Three letters give a nice result.
- Colour – You may also select a colour for your ID too.
- Name – You can provide a short name also.

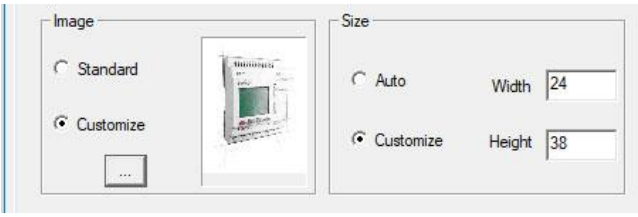


Figure 42 - LID Graphic Change

- Image – Click on the 'Customise' option and browse for the image you wish to select. Note: This MUST be a bitmap (.bmp) image NOT jpeg.
- Size – 'Select Customise' and enter the Width and Height you wish the LID block icon to be. You don't want this to be too big, so something like 24 x 38 for example.

Saving a Library Function

Once you have placed the blocks you require into your new Library block, then simply Save it by clicking 'File…Save' or Ctrl-S and give it a meaningful name. Every time you work on the Library function and re-save it, the software will prompt you to enter a new version number:

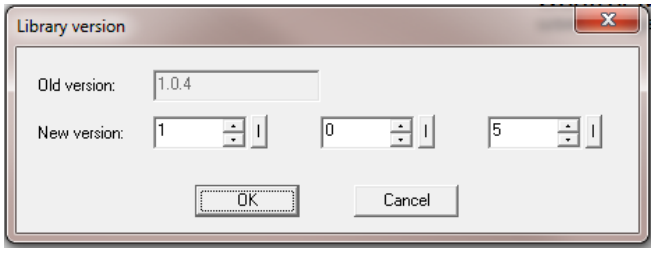


Figure 43 - LID Version Control

Your new Library function will be saved with a '.xid' extension rather than the standard program which uses '.xlg'.

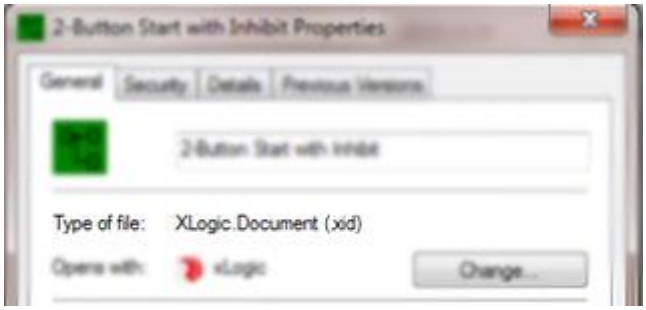
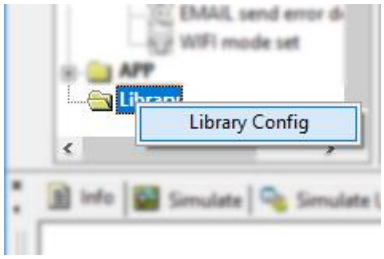


Figure 44 - LID Saved as xid

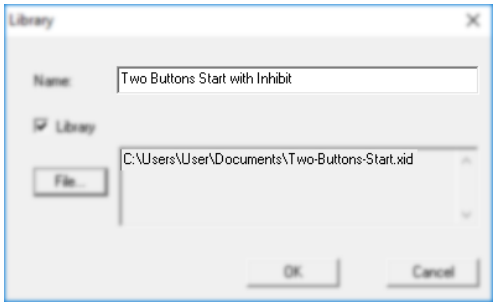
Once you have saved your LID file, you need to Add it to your Library.

1. Right click on Library in the tree menu and select Library Config. Click on Add button in the dialogue window that appeared.



2. Enable the Library check box by placing a tick in it and click File to browse to where you saved your LID block:

3. Give your new LID block a name. make it something meaningful as this will be the name which appears in your Library list:



It will now appear in your xLogicsoft menu tree for use whenever you need it:

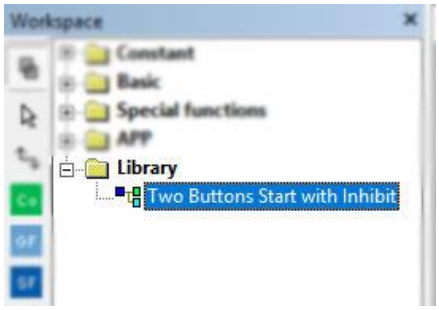


Figure 45 - LID View in Library Tree

Updating the Library Function Block

Open your existing Library function (LID) using the main menu 'File…Open' or Ctrl-O and browse to the directory in which it is saved. By default, xLogicsoft will assume you wish to open a standard program '.xlg', so use the 'File Type' dropdown to select '*.xid' or any option including that extension:

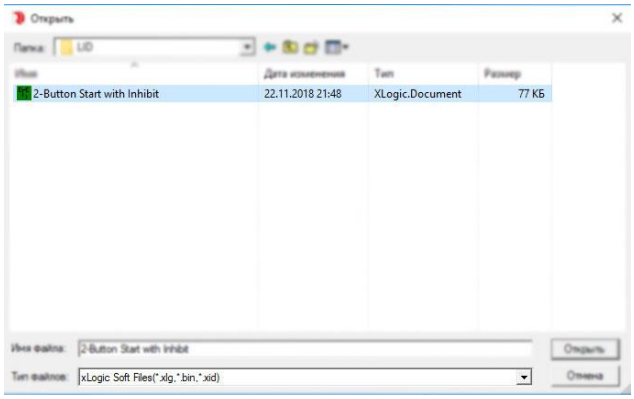


Figure 46 - LID File Selection

Highlight your LID and click Open. You can now freely edit your Library function.

If you update your Library block after using it in an xLogicsoft program, when you open the program again, the library will require updating to the most recent version. This is indicated in your program by a large Red X in the LID block:

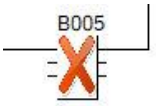


Figure 47 - LID Update Required

To update the LID block, right-click on it and select 'Update Library'.

6.1.7. MQTT Functions

MQTT, Message Queuing Telemetry Transport, is a machine-to-machine (M2M), Internet of Things (IoT) connectivity protocol used over a TCP/IP connection. It is an extremely lightweight publish/subscribe messaging transport and is useful for remote connections where low-bandwidth, high-latency and/or unreliable networks exist.

The xLogicsoft MQTT functions allow you to publish and subscribe to MQTT cloud servers such as Amazon, Ali, Baidu, etc. Users can use MQTT clients to subscribe to the cloud service and when your xLogic publishes a message, it is pushed to ALL subscribers at the same time.

Similarly, if the xLogic is programmed with MQTT Subscribe function, any message published to the cloud service will be received by the xLogic as long as it is connected/subscribing at the time the message is published.

MQTT functions are only available in the following xLogic models:

- PR-12 Series
  - PR-12AC-R-N
  - PR-12DC-DA-R-N
- PR-18 Series
  - PR-18AC-R-N
  - PR-18DC-DAI-R-N
  - PR-18DC-DAI-TN-N
- PR-26 Series
  - PR-26AC-R-N
  - PR-26DC-DAI-RA-N
  - PR-26DC-DAI-RT-N
  - PR-26DC-DAI-RT-2G
  - PR-26DC-DAI-RT-4G
  - PR-26DC-DAI-RT-WIFI
  - PR-26DC-DAI-RT-4GWIFI

Configure xLogic for MQTT Server

There are plenty of MQTT brokers on the internet and some provide their services free. One such broker is Mosquitto and their MQTT Cloud server address is https://test.mosquitto.org.

Your xLogic CPU uses Ethernet to connect to these servers and requires the IP address and not the domain name for its configuration. To obtain the IP address of the cloud server you wish to use, you must Ping the domain name. In our example, the IP address for the cloud server is 37.187.106.16.

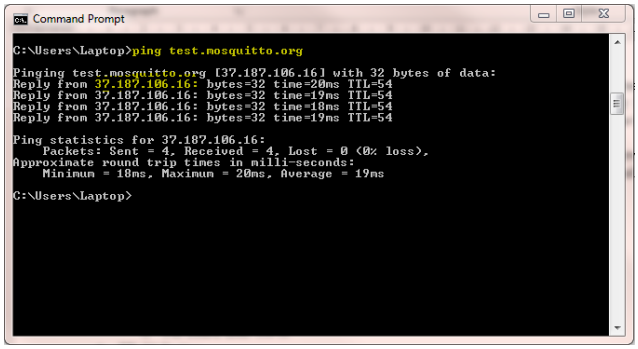


Figure 48 - MQTT Server Ping

Connect to your xLogic PLC using xLogicsoft, here we are using a USB cable on COM1:

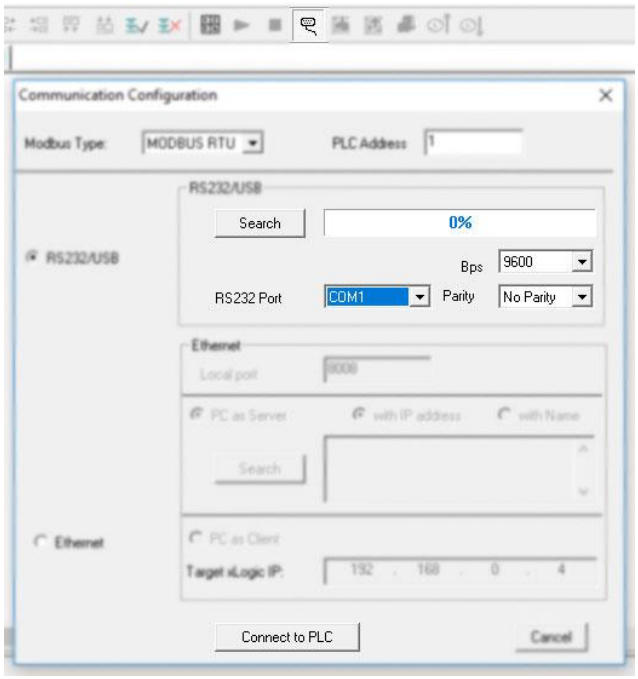
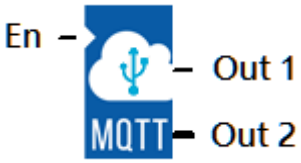


Figure 49 - Serial Connection to PLC

Use the Connect to MQTT Server block to establish a connection to the MQTT Broker of your choice.

Connect to MQTT Server



Description of Function

This function block is used to configure and connect to the MQTT server.

The IP and port information of the MQTT server is no longer obtained from 'Web Server Config'. Instead, get from this block, which triggers this block to connect to the server.

Publish and subscribe function blocks also do not configure the parameters of the MQTT server, they are simplified to have only publish and subscribe features.

Only one Connect to MQTT Server block can be placed in the program.

Connection	Description
En	You enable/disable the block with the signal at input En (1/0).
Parameter	**IP Address:** The IP address of the MQTT server. **Port:** The port of the MQTT server. **Connection timeout:** Timeout setting for cloud connection in seconds. **PING interval time:**
	Ping timer setting in seconds. If there is no message to publish before the timer expires, the PLC will send a ping command to the cloud server to check connection status. **Enable SSL/TLS:** After checking, select to use SSL encryption mode. **CA Certificate File 1:** When using encryption mode, select the SSL certificate by path. **CA Certificate File 2:** Temporarily not supported. **CA Certificate File 3:** Temporarily not supported. **Client ID:** Click the 'Create only one ID' button to generate a unique string of numbers. **Name / Password:** Before you can publish / subscribe a message, you need to configure your cloud server account and 'Topics'. Each cloud server may have different configuration procedures. Check the cloud server instructions on how to do this. **Hide:** Hide the password. You can set a secret password when hiding, and

	you need this secret password when you unhide it. You can also modify the content without a secret password. You can also choose not to set a secret password when hiding.
Out 1	Digital output. Shows if the server is connected successfully. **1: Successfully connected to the MQTT server** **0: The module is not triggered or is connecting to the server**
Out 2	Analog output. The status code of the connection server. **2: Connecting to the target server** **3: MQTT PING is sending PING packet** **13: MQTT PING success** **5: Disconnect** **50: Connection failed** **256: The connection is successful**

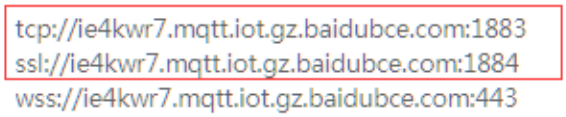
You may now use the MQTT function blocks in your xLogicsoft program.

Setting Up MQTT Server Example

This document uses Baidu Cloud as an example. Different MQTT servers may have different setup methods.

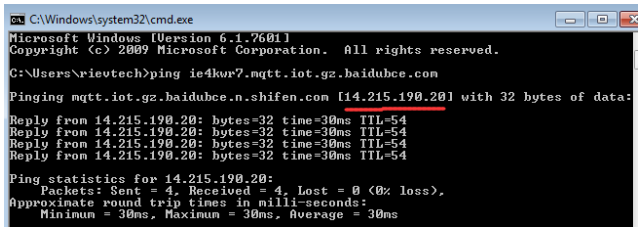
1. Enter Baidu Cloud's MQTT server and create a data-type project.

Follow step by step instructions according to the requirements of the MQTT server, set parameters such as Name, Password, and topic. Finally, the MQTT server will tell you the domain name and port of your MQTT server, as shown in the example below:



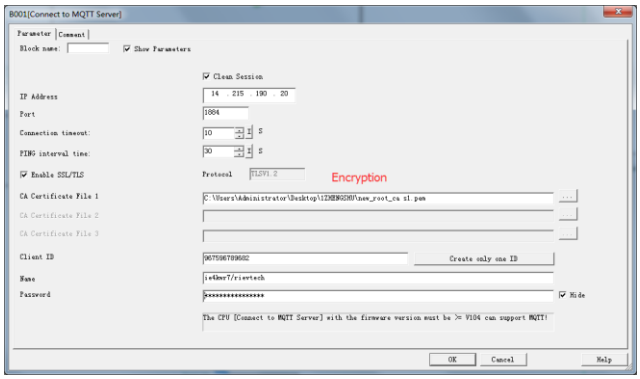
In the red box above, the above method (TCP ://...) is the domain name and port of the unencrypted transmission method. The following method (SSL ://...) is the domain name and port of the SSL encryption transmission method.

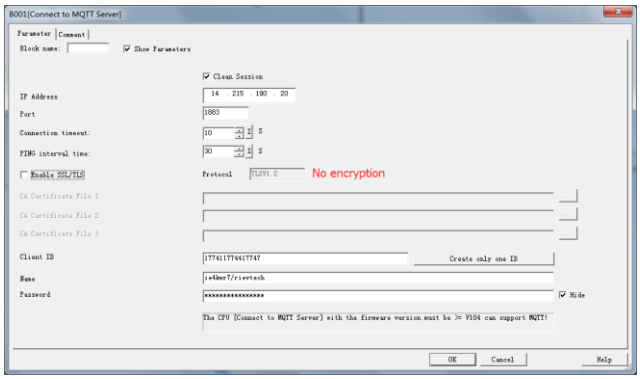
2. You can use the following method to find the IP address of the MQTT server.

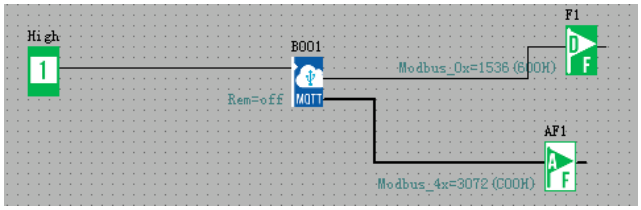


Fill in the IP obtained in the above figure into the Connect to MQTT Server block, as shown below:

The following two figures are configuration diagrams that use encryption and no encryption, respectively.



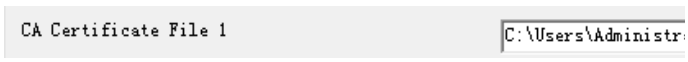




NOTE:

a), 'Name' and 'Password' are obtained/set when creating a project on the MQTT server.

b) The SSL certificate is issued by a trusted digital certificate authority CA (eg Symantec) after verifying the identity of the server. It has server authentication and data transfer encryption.



c) Check the 'Clean Session' option to clear the session cache. Checked by default.

3. When the settings are complete, click the 'OK' button

4. Sample program:

MQTT Data Format in xLogicsoft

All MQTT functions use ASCII Decimal values to represent data to be transmitted/received by the xLogic PLC.

Example:

The string of characters 'ABCD' will be represented as 65 66 67 68, whereas '1234' will be represented as 49 50 51 52. Each character of the ASCII format is 8-bits:

Character	Decimal	Binary	Hex
0	48	0011 0000	30
1	49	0011 0001	31
2	50	0011 0010	32
3	51	0011 0011	33
…	…	…	…
9	57	0011 1001	39
A	65	0100 0001	41
B	66	0100 0010	42
C	67	0100 0011	43
D	68	0100 0100	44
…	…	…	…
Z	90	0101 1010	5A
a	97	0110 0001	61
b	98	0110 0010	62

Figure 50 - MQTT Data Table

VB Block

Description of Function

This function provides for storing 8-bits of data, one character using the Decimal value of the character, which can be used in MQTT communications.

A maximum of 512 VB blocks can be used per xLogic PLC program and the number is reduced with every occurrence of VW or VD blocks used.

This block is used by the Subscribe MQTT function to determine the start character position of a string.

Connection	Description
Input En (—▸)	A high signal at En input will enable this function block to be activated
Output Q	Q is set (1) or reset (0) depending on whether the block is active or not.

Example:

Figure 51 - Setting MQTT Start Byte

Any characters received by the above block will be stored starting in block VB10. The maximum number of characters allowed is 256, i.e. VB265 would store the last possible character.

NOTICE:

VB Blocks use the same memory area as AF blocks, so care must be taken if both types of blocks are used in the same program. AFx will interfere with VBx and VB(x-1), in order to avoidsuchinterference, please use different indexes of VB and AF.

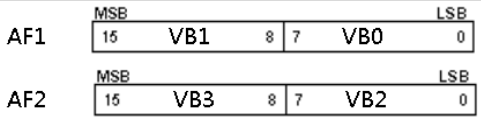


Figure 52 - AF and VB Memory

VW Block

Description of Function

This function provides for storing 16-bits of data, two characters using the Decimal value of each character, which can be used in MQTT communications.

A maximum of 512 VW blocks can be used per xLogic PLC program and the number is reduced with every occurrence of VB or VD blocks used.

This block can be used by the Subscribe MQTT function to determine the start character position of a string to be received.

Connection	Description
Input En (⟶)	A high signal at En input will enable this function block to be activated
Output Q	Q is set (1) or reset (0) depending on whether the block is active or not.
Input En (⟶)	A high signal at En input will enable "Modbus Read" function block to be activated

Example:

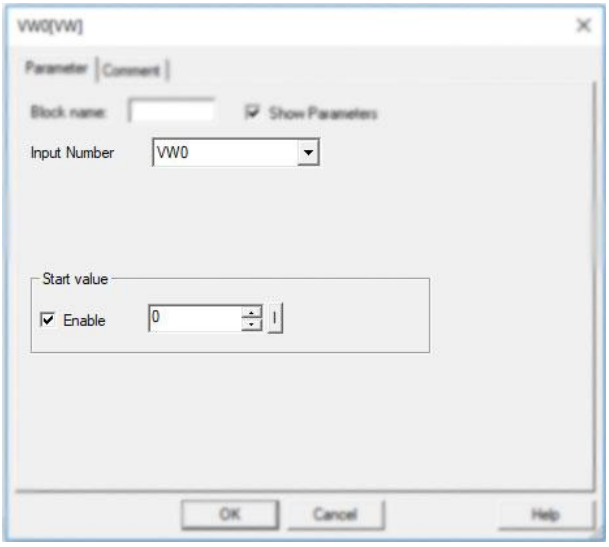


Figure 53 - Storing VW Data

Any characters received by the above block will be stored starting in block VW0. The maximum number of characters allowed is 256. You can also specify a Start value, which could be used as a start of message indicator.

NOTICE:

VW Blocks would show the AF1 value with high and low bytes swapped. VW1 will start with the same value as the end of the previous VW block as shown below. VW use Modbus addresses 0x6800 to 0x7000.

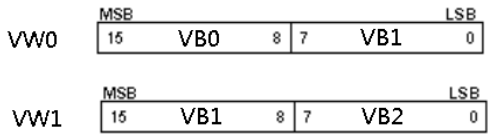


Figure 54 - VW Block Memory Usage

VD Block

Description of Function

This function provides for storing 32-bits of data, four characters using the Decimal value of each character, which can be used in MQTT communications.

A maximum of 509 VD blocks can be used per xLogic PLC program and the number is reduced with every occurrence of VB or VW blocks used.

This block can be used by the Subscribe MQTT function to determine the start character position of a string to be received.

Connection	Description
Input En (—▸)	A high signal at En input will enable this function block to be activated
Output Q	Q is set (1) or reset (0) depending on whether the block is active or not.

Example:

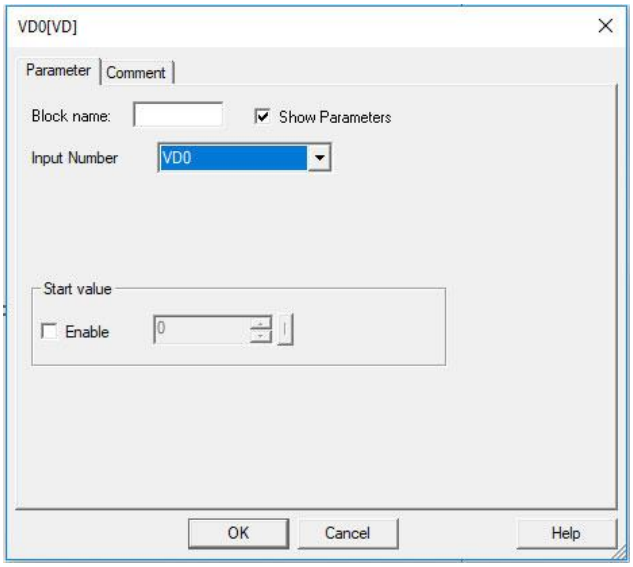


Figure 55 - VD Block Start Address

Any characters received by the above block will be stored starting in block VD20. The maximum number of characters allowed is 256.

NOTICE:

VD Blocks use the same memory area as VB and VW blocks, so care must be taken if both types of blocks are used in the same program. In the example below, VD1 has VB1 as its most significant byte and VB4 as its least. You can use VD blocks as a kind of pointer, where VDx will have VBx as its most significant byte. For example, is you used VD53 as the input number setting, then the most significant byte would be VB53, followed by VB54, VB55 and least significant byte would be VB56.

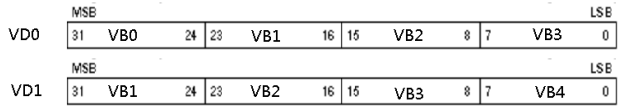


Figure 56 - VD Block Memory

String to Integer

Description of Function

The String to Integer function can be used to convert a string consisting of numbers into an integer value. This can then be used like any other integer within your program such as a new set point, timer value, etc. The function is capable of converting a 16-bit SIGNED integer in the range of -32768 to 32767.

Connection	Description
Input En ( )	A high signal at En input will enable this function block to be activated
Output AQ	AQ is the integer value from the converted string.

Example:

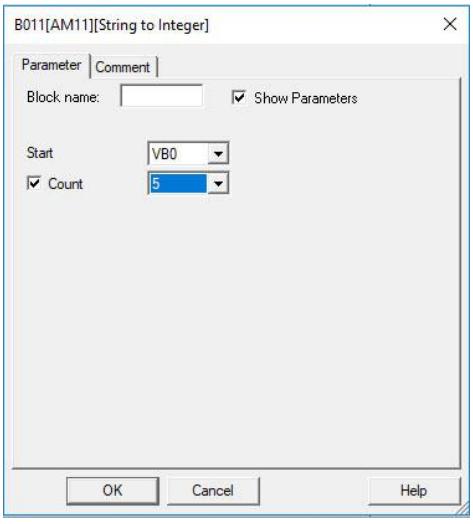
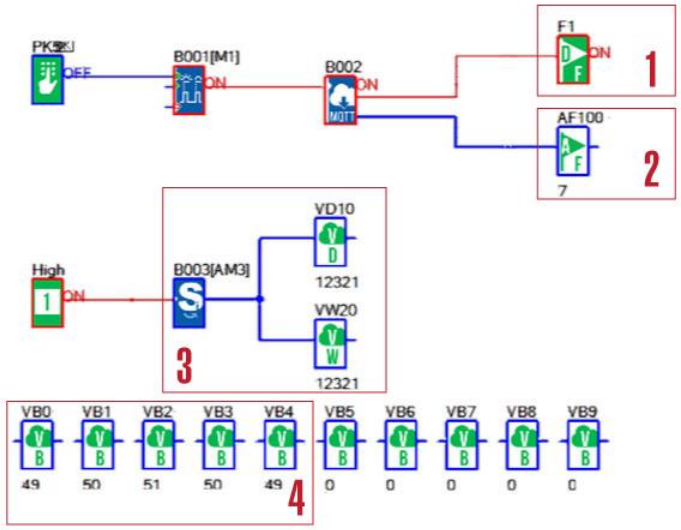


Figure 57 - String To Integer Configuration

The above configuration would convert the numeric value of a string stored in VB0, VB1, VB2, VB3, and VB4.



01: Flag is set to On when a message is received.

02: Flag value indicates the status of the connection to the broker. '7' indicates the PLC is Subscribing successfully to the broker. See Publish Status codes for further details.

03: A VD and VW block showing the value as a number of the string received from the broker.

04: The VB blocks show the string received from the broker as ASCII values, 1=49, 2=50, 3=51, etc.

If the string received from the MQTT broker was "12321" then this would be converted to a numeric value of 12321.

If the string received from the MQTT broker was greater than 32767 or less than -32768, then the string will be converted to a meaningless value. If the string received was a character, a through z, then this will be converted to a value of '0'.

String Comparator

Description of Function

This function is used for finding a character or characters in a message received via MQTT Subscribe. Once you have placed this block in your program, double-click on it to enter its parameters.

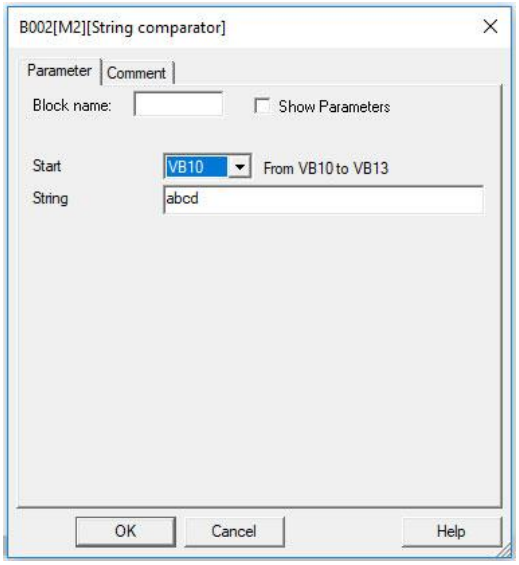
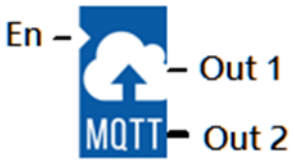


Figure 58 - String Comparator Settings

- Start – Select the VB block number in which you wish to start your comparison.
- String – Enter the string you wish to search for. This is case sensitive.

During program execution, if the String Compare block finds the string then its output is set High. It will remain High until the next search. If a match is not found, then the block will remain set to Low.

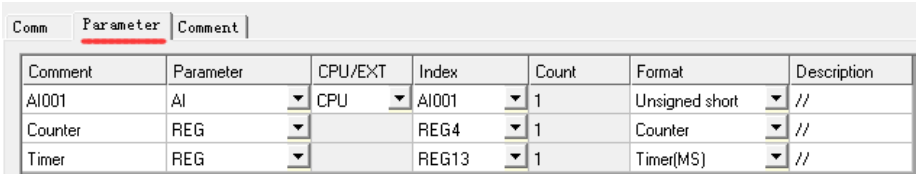
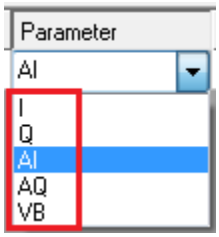
Publish with MQTT



Description of Function

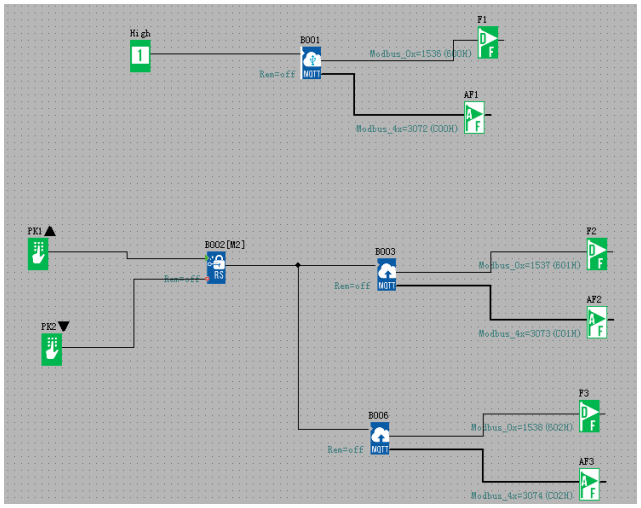
After successfully connecting to the MQTT server, activate this function block to publish the message to a topic in the MQTT server.

Connection	Description
En	You enable/disable the block with the signal at input En (1/0).
Parameter	**Retained:** Typically, if a publisher publishes a message to a topic, when no one subscribes to the topic, then the message will be discarded (not retained) by the MQTT server. However, the publisher can tell the MQTT server to keep the last message of the topic by setting a reserved message flag. When a device subscribed to this topic goes online, the MQTT server sends the retained information to the subscriber. You can check this option to implement this 'information retention' feature. A powered-down device can receive the last message of its subscribed topic after powering on. **QoS Level:** QoS 0(no answer) or QoS 1(answer). If you wish to receive a response from the MQTT server, select 'QoS 1'. **Publish interval time:** Publish timer setting. If enabled, the xLogic will publish the message to the MQTT server every x seconds. If disabled, the message will only be sent once. **Publish topic:** Set when creating a new topic project on the MQTT server.

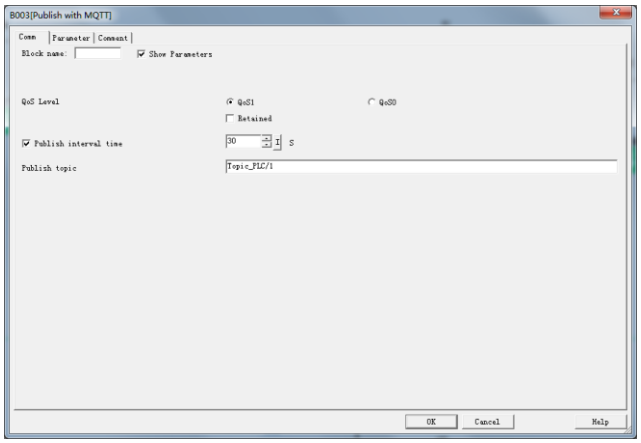
	Parameter: Set the message content.   I – A digital input state Q – A digital output state AI – An analogue input value AQ – An analogue output value VB – Value of VB type data **CPU/EXP:** If using IO, select which unit, CPU or expansion unit it is referring to. **Index:** Select the start index/IO channel of the data. **Count:** Select, AI001 – AI00C, the number of values starting from the index entered. For example, if you selected as per the above picture and changed the Count from 1 to 4, then the message would provide the value of AI001 – AI004. **Format:** Automatically filled in dependant on the Parameter selected. **Description:** A few characters which will be added to the end of the message.
Out 1	Digital output. Indicates whether the information was successfully published. **0: Not triggered or failed to publish** **1: Successfully publish data**
Out 2	Analog output.

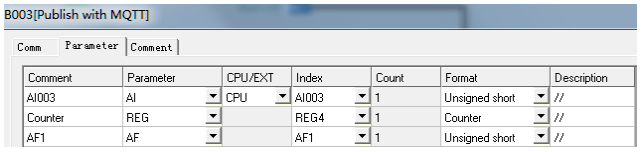
	Publish status code. **0: Block is not triggered** **2: Waiting for the MQTT server to connect** **4: Time to time start send the published data to the server and wait for server. Response (QoS1).** **14: Successful response to server response (QoS1)**

Example



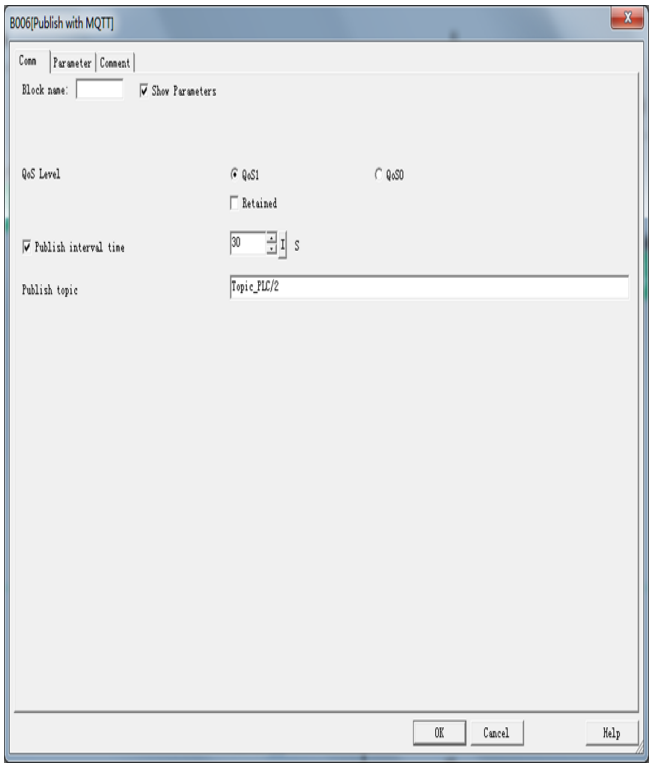
First Publish Block

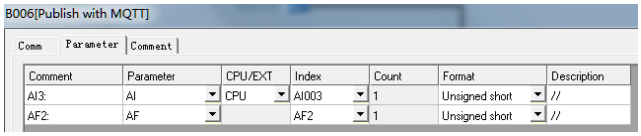




Comment	Parameter	CPU/EXT	Index	Count	Format	Description
AI003	AI	CPU	AI003	1	Unsigned short	//
Counter	REG		REG4	1	Counter	//
AF1	AF		AF1	1	Unsigned short	//

Second Publish Block





Comment	Parameter	CPU/EXT	Index	Count	Format	Description
AI3:	AI	CPU	AI003	1	Unsigned short	//
AF2:	AF		AF2	1	Unsigned short	//

Figure 59 - MQTT Publish Settings

- Publish Topic: Before you can publish a message, you need to configure your MQTT cloud server account and include setting up of 'Topics' usually under the Themes area on your account. Each MQTT cloud server may have different configuration procedures for their Topics. Check the MQTT cloud server instructions on how to do this. There are generally five steps:

1. Create an account on a cloud server.
2. Add a device to your account.
3. Set-up an identity for your account.
4. Create a policy for your account.
5. Create a topic for your account.

Once you have configured the Topic name on the MQTT cloud server you are using, enter it here.

- CPU ID: Each xLogic CPU is required to have a unique cloud identity, which is configured here.
- Name: Enter your Username for the MQTT broker account you created.
- Password: Enter your Password for the MQTT broker account you created.

Param tab settings

This tab is used to enter the messages you wish to publish to the cloud MQTT server for this block.

1. Fill in the first parameter you wish to send. Click the Add button to add more parameters to the message.

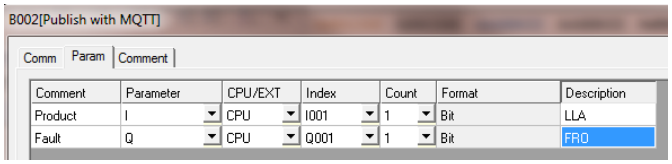


Figure 60 - MQTT Publish Param Settings

The above shows the configuration for publishing a digital input (I001), Product Low-Level Alarm and a digital output (Q001), Fault Relay Output. Please remember MQTT is similar to SMS in that you are limited to the number of characters you can send in each message, so keep your comment and description as concise as possible.

- Comment: This will be the first few characters of your message.
- Parameter: Use the drop-down box to select which xLogic parameter this message refers to. Choices are:
  - I – A digital input state
  - Q – A digital output state
  - AI – An analogue input value
  - AQ – An analogue output value
  - F – A digital flag state
  - AF – An analogue flag value
  - M – A middle coil register state
  - AM – An analogue register value
  - REG – A register value
- CPU/EXP: If using IO, select which unit, CPU or expansion unit it is referring to.
- Index: Select the start index/IO channel of the data.
- Count: Select, 1 – 8, the number of values starting from the index entered.
  - For example, if you selected as per the above picture and changed the Count from 1 to 4, then the message would provide the state of digital inputs, I001 to I004.
- Format: Automatically filled in dependant on the Parameter selected.
- Description: A few characters which will be added to the end of the message.

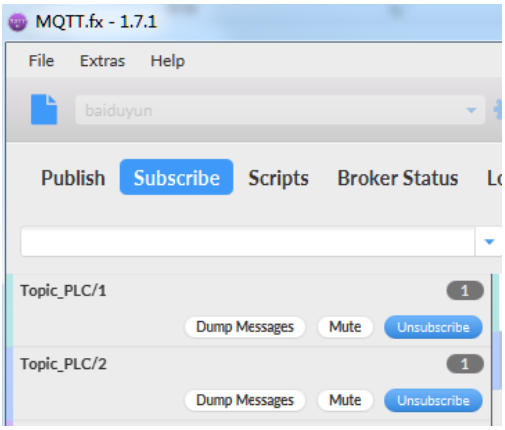
A maximum of 16-parameters can be added to each message.

Click OK button to save your settings.

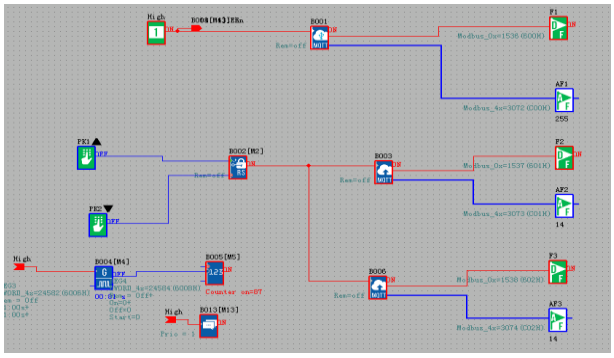
MQTT Publish Status Codes

Code	Description
00	*Connection Accepted*
51	*Connection Refused: unacceptable protocol version*
52	*Connection Refused: identifier rejected*
53	*Connection Refused: server unavailable*
54	*Connection Refused: bad username or password*
55	*Connection Refused: not authorized*
2	*Wait to connect to the server.*
3	*The PING data packet is being sent.*
4	*Time to time start send the published data to the server and wait for server response (QoS1).*
5	*Timeout – Disconnect and retry server connection.*
13	*MQTT PING success*
14	*Successful response to server response (QoS1) .*
255	*Successful connection to the server and ready to Publish.*

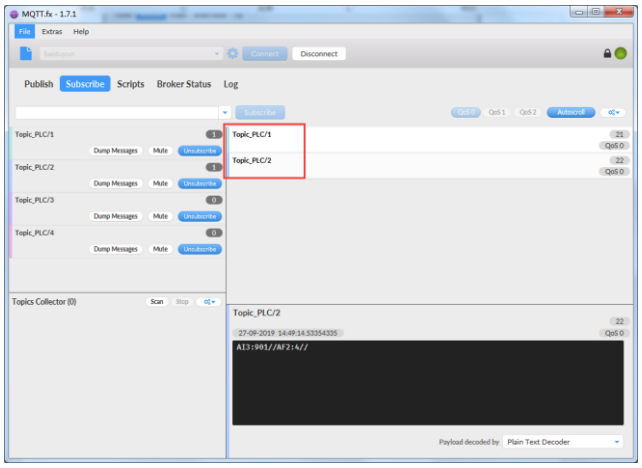
Subscribe to Topic_PLC/1 and Topic_PLC/2 on the MQTT client software MQTT.fx.



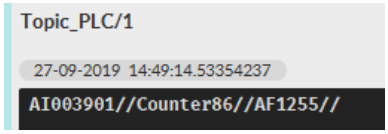
Press the up button on the LCD panel to trigger the publish blocks, see the online monitoring screenshot below:



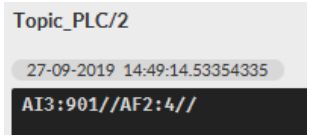
You will receive two messages on the MQTT.fx software:



Topic_PLC/1 content:

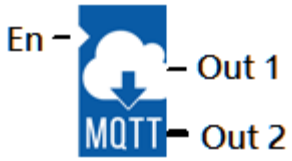


Topic_PLC/2 content：



This indicates that the PLC successfully published information to the MQTT server.

Subscribe with MQTT



Description of Function

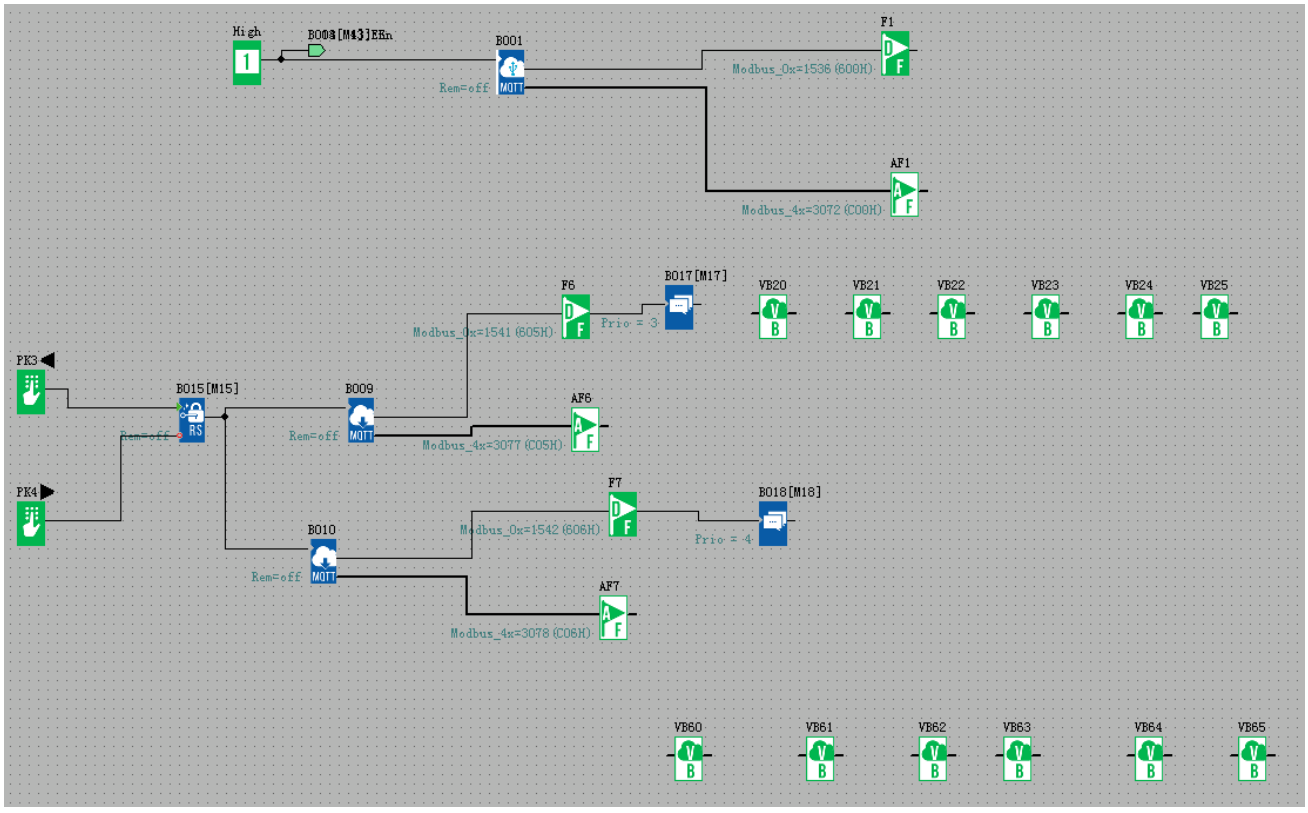
You can use this function to allow the xLogic to subscribe to a message on the MQTT cloud server. Only 8 Subscribe blocks can be placed in one program.

Once you have added it to your program, double-click the block to configure it.

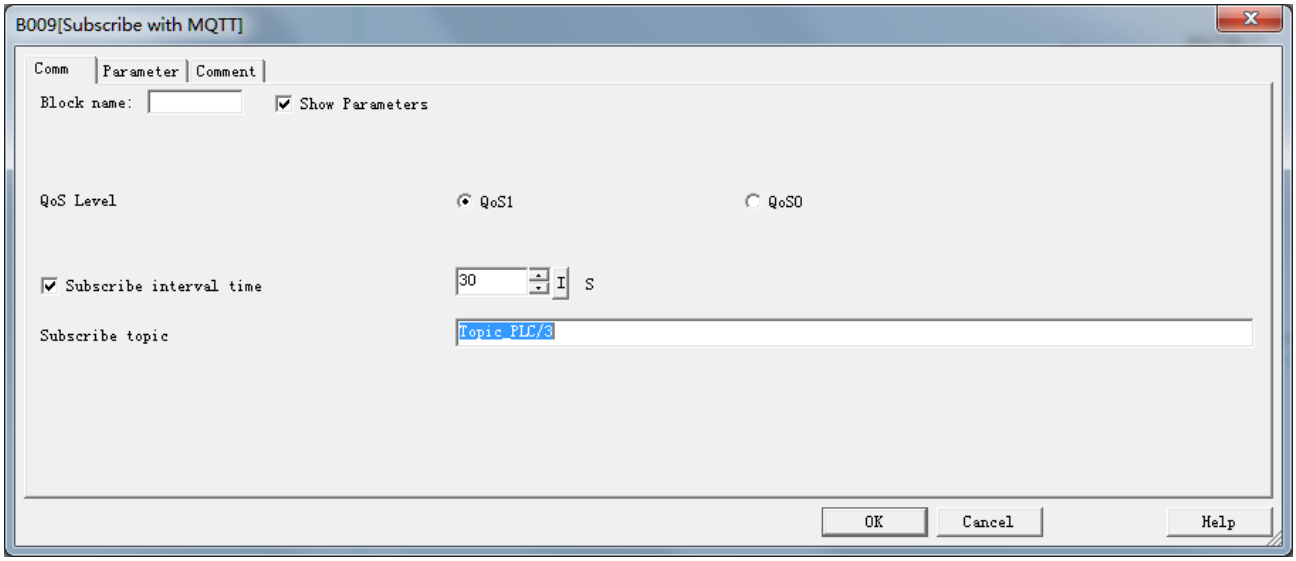
Connection	Description
En	You enable/disable the block with the signal at input En (1/0).
Parameter	**QoS Level:** QoS 0(no answer) or QoS 1(answer). If you wish to receive a response from the cloud server, select 'QoS 1'. **Subscribe interval time:** Subscribe timer setting. If enabled, the xLogic will Subscribe to the message from the cloud server every x seconds. If disabled, the message will only be subscribed to once. **Subscribe topic:** Fill in the topics you want to subscribe to. However, it should be noted that the topic of the multiple subscribe blocks cannot be the same. **Parameter:** Set the start register to store the received information data.
Out 1	Digital output. Indicates whether the information was successfully subscribed. **0: Not triggered or did not receive the topic it subscribed to** **1: Received a subscription topic**
Out 2	Analog output. Subscribe to the status code.

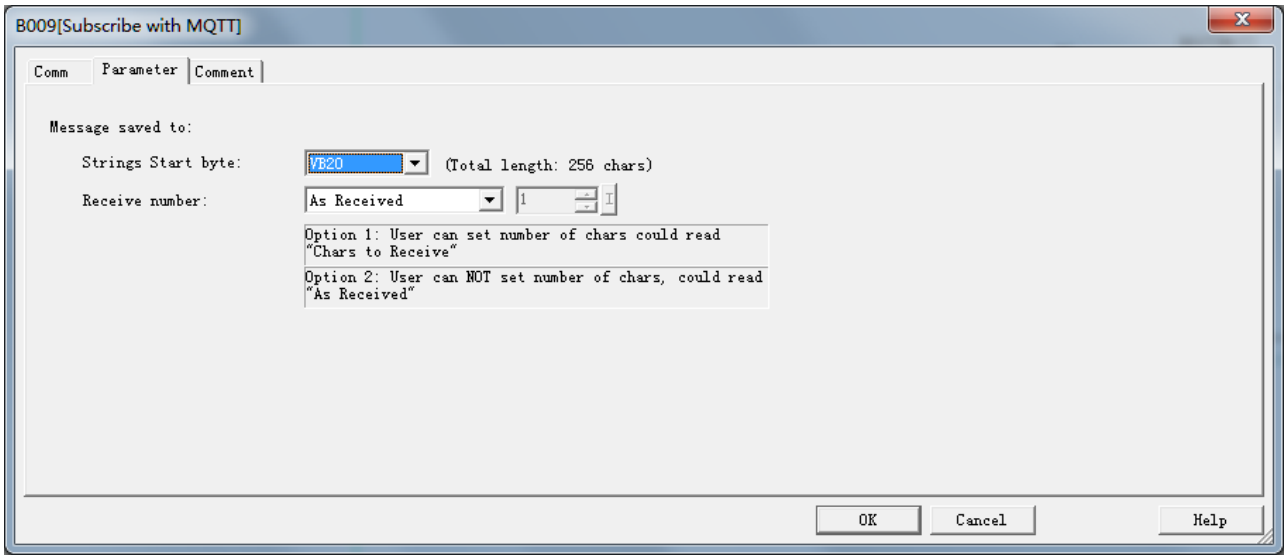
	0: Block is not triggered **1: There are no free subscription channels (maximum 8 subscription blocks).** **2: Wait for the MQTT server to connect or wait for the subscription channel to be idle.** **6: Sending a subscription request** **16: The subscription request is successful, waiting to receive the topic of the subscription.** **26: Received a topic subscribed to by this block** **36: Subscription request failed, waiting for the block subscription timeout** **8: Unsubscribe**

Example

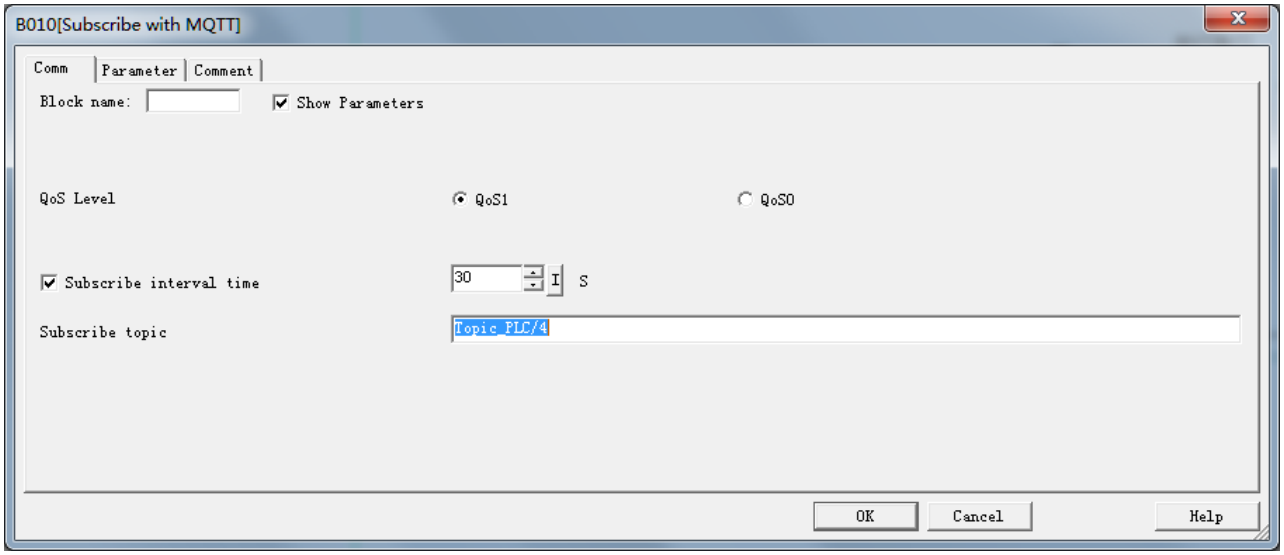


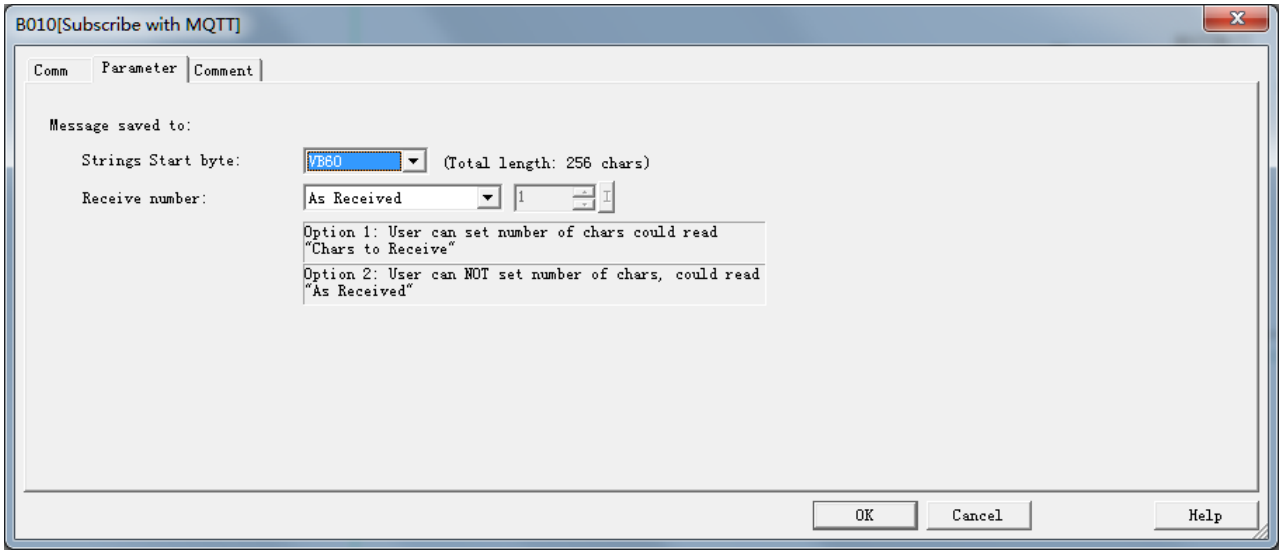
First block:





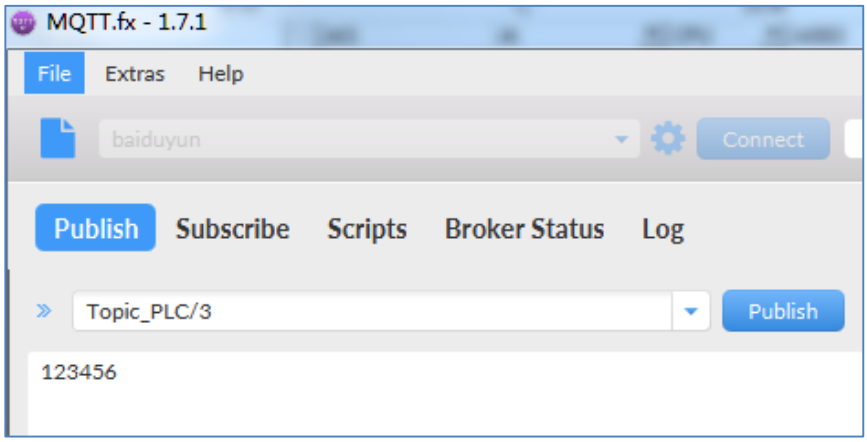
Second block:

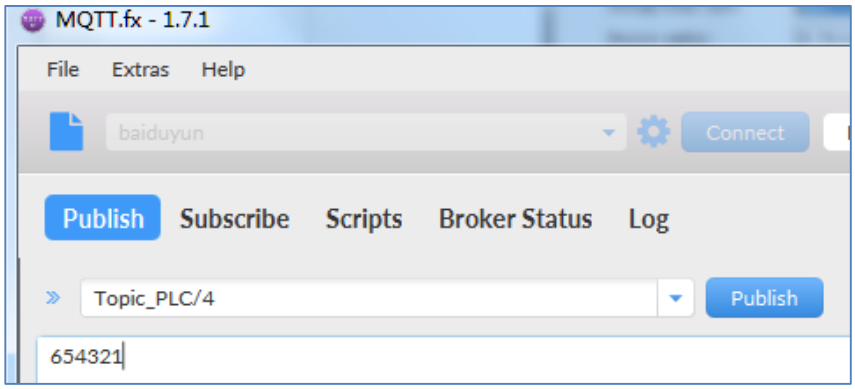


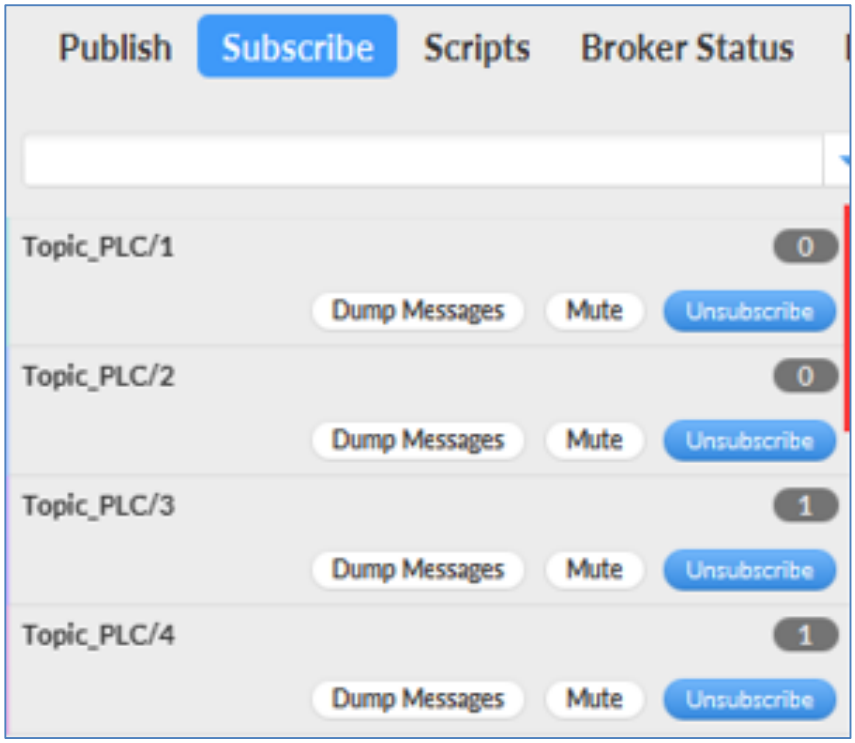


Press the left button on the LCD panel to trigger the subscription function.

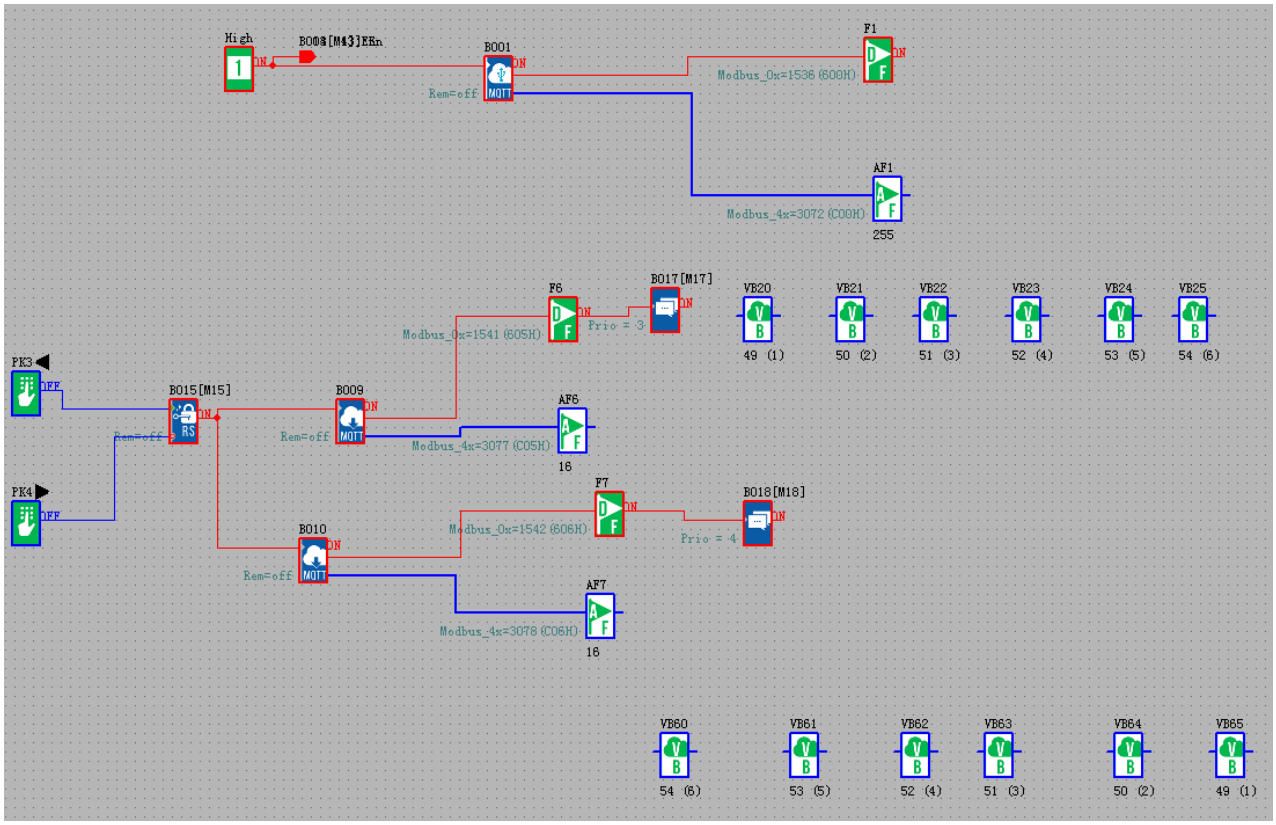
Topic_PLC/3 and Topic_PLC/4 are published on the MQTT client software MQTT.fx. At the same time MQTT.fx also subscribes to Topic_PLC/3 and Topic_PLC/4.





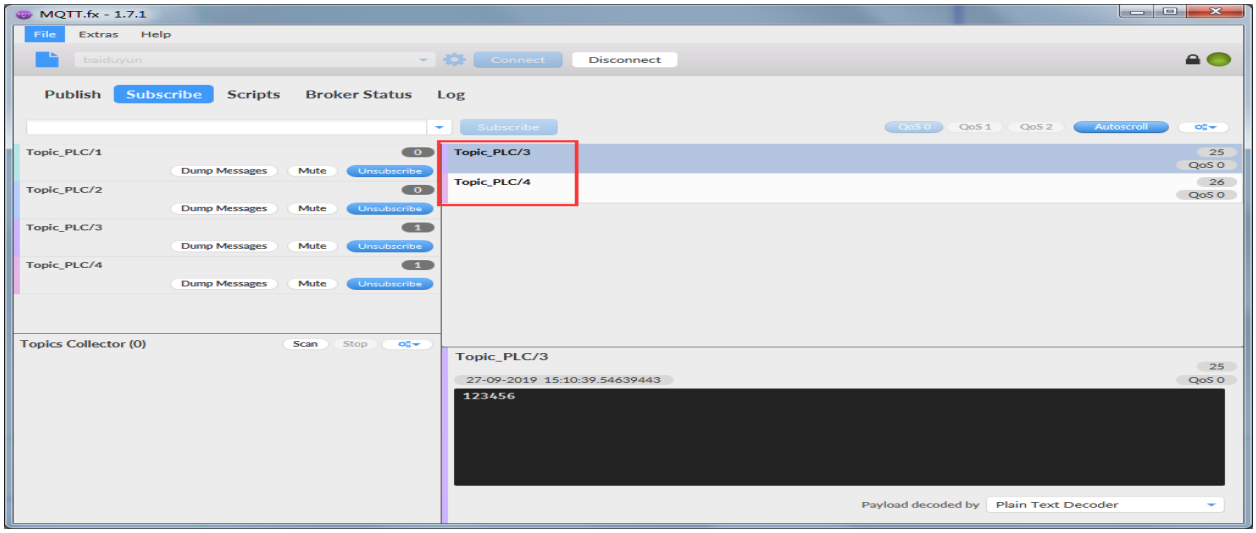


Monitor on xLogic software:

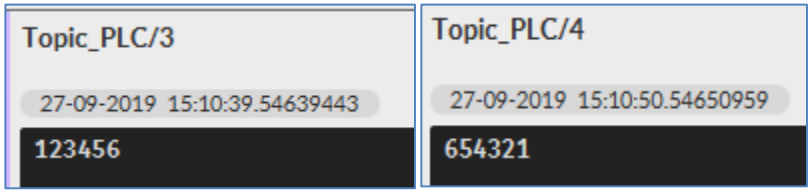


The value received by VB is ASCII code.

This indicates that the subscription to Topic_PLC/3 and Topic_PLC/4 was successful.



Topic_PLC/3 and Topic_PLC/4 content:



7. xLogicsoft

xLogicsoft runs under Windows 95/98, Windows NT 4.0, Windows Me, Windows 2000, Windows XP, Windows Vista, Windows 7 and Windows 10. xLogicsoft is capable of client/server operation and offers you a high degree of freedom for creating your circuit program.

V3.3.6.3 – This is the current version of xLogicsoft. You will find all the functions and the functionality of the devices described in this manual for this version and earlier.

NOTICE:

If a full version is not installed, you can carry out an upgrade as follows:

- Install the software from the CD.
- When the system prompts you for the previous version, place the old xLogicsoft CD in CD drive.

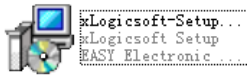
Point your browser to the "...₩Install" directory on the CD.

Updates and Info

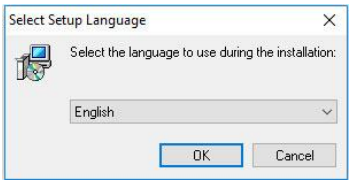
You can download demo versions of the software free of charge from the Internet address specified in the preface.

7.1. Installing xLogicsoft

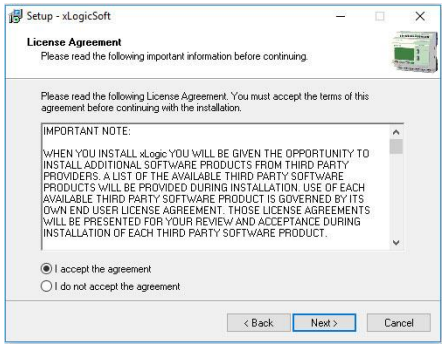
1. Double-click on Setup.exe or left click the "INSTALL" menu directly.



2. Select the language you would like and click OK to confirm

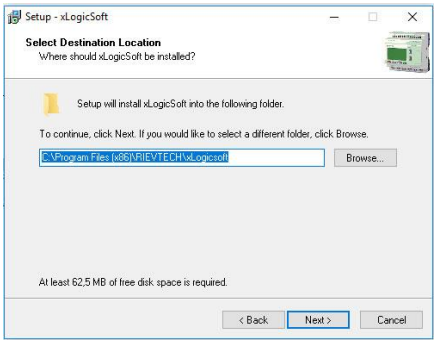


3. If you consent to the license agreement, click Next to confirm.

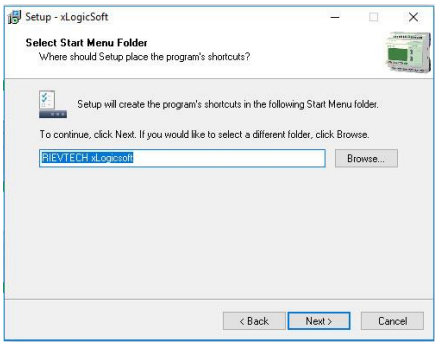


4. Where is the program to be installed? If you do not want to accept the recommended file location:

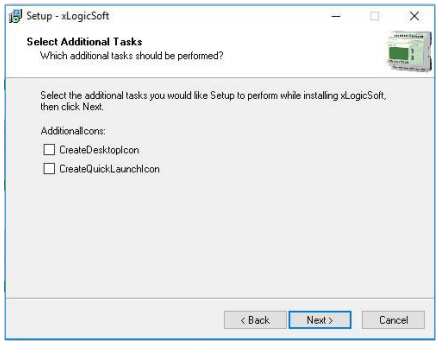
C:₩Program Files₩Rievtech₩xLogicsoft, specify another directory using Browse.



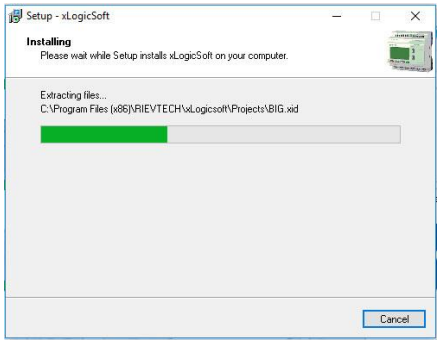
5. If you want to accept the recommended file location, click Next to Confirm.



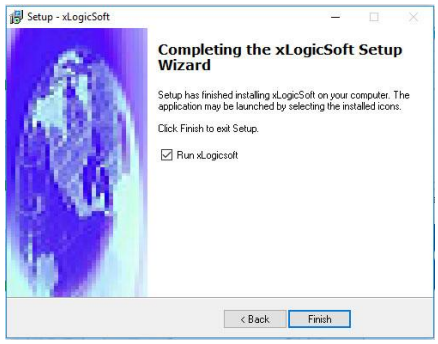
6. In this example, the program icon is to be placed on the desktop. Use Next to proceed.



7. Click Install button to install. The program is being installed. . . . .

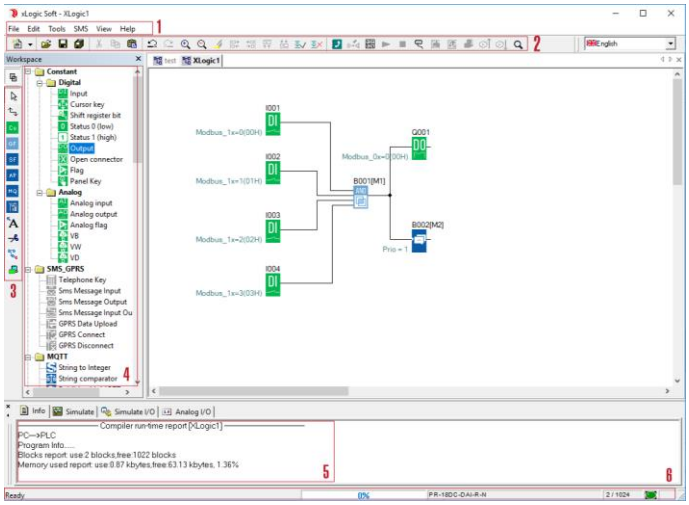


8. The installation is finished. You can start the xLogicsoft immediately or later by double-clicking the icon on the desktop.



You now see the complete user interface of xLogicsoft. The programming interface for creating your circuit programs occupies the greater part of the screen. The icons and logical links of the circuit program are arranged on this programming interface.

To help you to maintain an overview of large circuit programs, the right side and the bottom of the programming interface contains scrollbars, which you can use for vertical and horizontal scrolling of the circuit program.



1. Menu bar

2. Standard Toolbar

3. Programming Toolbar

4. Reference material (Function block list)

5. Infobox (Display memory Info, IO status and analogue IO values under simulation/monitoring mode)

6. Status bar (Including current operation, current CPU model, and the communication status)

7. Programming Interface

NOTICE:

For the detailed operation, please refer to Chapter 10 and Chapter 11 of this part.

7.2. Main Functions

Editing Function

The main function of xLogicsoft is creating programs for xLogic. By using the Logical Function Edition window of xLogicsoft, you can create and edit your desired xLogic programs using various function blocks of xLogic and can also perform file operations such as save, print, program management etc.

Simulation Function

After the program is edited, you can view the program operation result on the computer and conveniently check if the said program meets your control requirements. Here xLogicsoft provides you with a completely new off-line test function, through which you can debug the program without installing the xLogic on site. With this function, many inconveniences of the site test can be avoided.

Real-time Monitoring

xLogicsoft has a Real-Time Monitoring window. You can view the process of the control system and the running conditions of all xLogic and control remote xLogic; by connecting the xLogic communication port (RS232, USB, Ethernet optional) to the computer you can view the process of the control system.

Documentation

The program circuit can be saved and printed. The analogue values also can be saved in an Excel file.

7.3. Operation Instructions

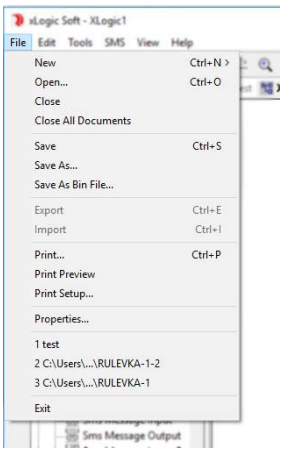
Menu Bar

When xLogicsoft is used to edit xLogic programs under customized mode, some basic operations including files management, opening and closing of the Tool Bar and Status Bar access to Help information are completed by using pull-down menu under File, Edit, Tools, SMS, View, and Help. The Instruction Function list of xLogicsoft is characterized by its flexibility and variation according to the main selection.

It can be changed according to the current operation for the convenience of your specific operations.

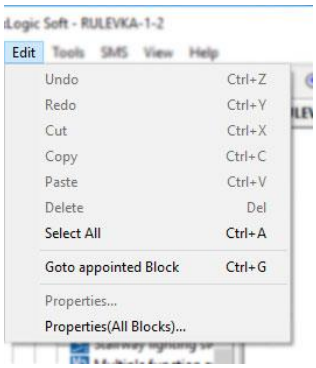
File

The instruction is mainly used for file management, including creation, opening, saving and printing of files.



Instruction Name	Function
New	Open a new file
Open	Open an old file
Close	Close the current active Window
Close All Documents	Close all the currently active Windows
Save	Save a file
Save As	Save current file to a new path and a new file
Print	Print a file
Print Preview	Preview the file before printing
Print Setup	Setup printing format
Property	File property (page, size & model selection)
Exit	Exit the xLogicsoft

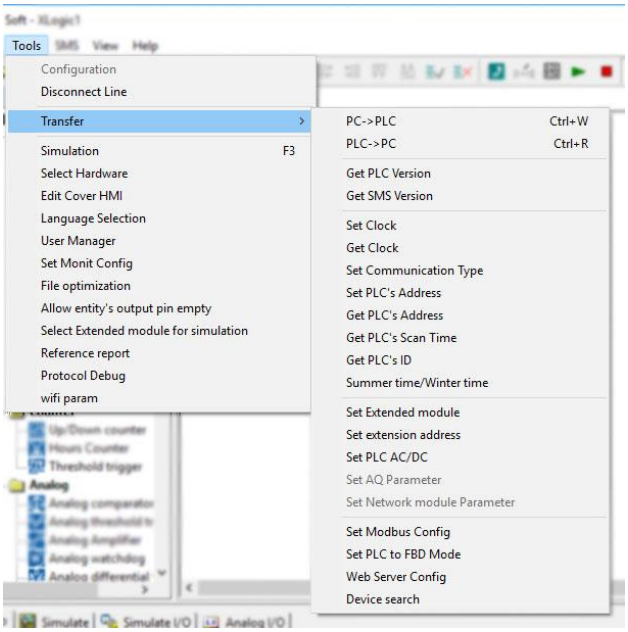
Edit Menu



- Undo: Undo the previous step/s of operation.
- Redo: Recover the contents undone by the previous step/s of operation.
- Cut: Cut the contents in the area highlighted with the cursor.
- Copy: Copy the contents highlighted with the cursor.
- Paste: Paste the contents cut or copied.
- Delete: Delete various graphic components.
- Select All: Select all the contents in the current window editing box.
- Go to Appointed block: Go to the appointed block in the program.
- Property: Open the property box of the selected block.
- Properties (all blocks): Open all the property boxes of the selected blocks.
- Phonebook: Add/Delete phone numbers in the phonebook. See chapter 6.5.2.

Tools Menu

This instruction is mainly used for reading a program from xLogic, writing program to xLogic, diagnosis of the communication to the xLogic, program management and simulation, etc. The pull-down menu of Tools is shown as follows:

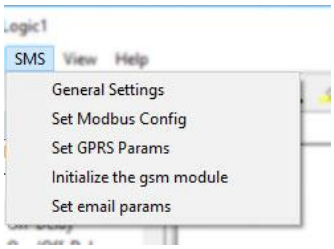


- Configuration: Select and open the Com port.
- Disconnect Line: Cancel the connection between PC and xLogic.
- Transfer: Transfer data between PC and xLogic.
  - PC-> PLC: Download the program to xLogic.
  - PLC->PC: Upload the program from xLogic.
  - Password set: Set Password protection to xLogic.
  - Get PLC Version: Get the current xLogic hardware version.
  - Set Clock: Set RTC for xLogic.
  - Get Clock: Get RTC from xLogic.
  - Get Communication Type: Get current communication type from xLogic.
  - Set Communication Type: Set communication type for xLogic.
  - Set PLC's address: Set the xLogic address.
  - Get PLC's address: Get the address of xLogic.
  - Get PLC's scan time: Get the scan period of the xLogic for the current program.

- Summer time/Wintertime: Enable/Disable Daylight Savings Setting.
- Set Extended module: Enable/Disable the expansion port of xLogic (Only applied to Standard ELC-12 series CPU)

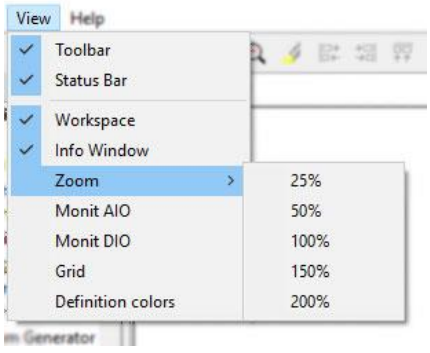
- Simulation: Simulate the xLogic program.
- Select Hardware: Select the model of xLogic for programming.
- Edit Cover HMI: Change the xLogic HMI displayed data.
- User manager: File management, program can be protected with different priorities.
- Set Monit Config: Select elements for monitoring or saving to an Excel file.

SMS Menu



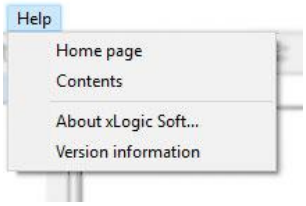
View Menu

This instruction is to display the status bar, workspace, and the Information window, etc. The pull-down menu is shown as the following:



- Status bar: Display Status Bar window.
- Workspace: Display Workspace window.
- Info Window: Display Info window.
- Zoom: Select display zoom level.

Help Menu



- Contents: Help index and detailed contents
- About xLogicsoft: List associated xLogicsoft files and their versions.
- Version: Display xLogicsoft version information.

Standard Toolbar

The icons on the standard toolbar provide quick access to commands that are also available via the menu.

	New		Align Top
	Open		Align Bottom
	Save		Display Block's parameters
	Save All		Hide Block's parameters
	Cut		Phone book
	Copy		Page Layout Tab
	Paste		Set PLC Run
	Undo		Set PLC Stop
	Redo		Open COM port
	Zoom In		Download (PC->xLogic)
	Zoom Out		Upload (xLogic->PC)
	Select Lines		On-line monitor
	Align Left		Get RTC from xLogic

	Align Right		Set RTC to xLogic
			Device Search

Programming Toolbar

The programming toolbar contains integral icons for creating, editing and testing programs. Each one of these tools represents a programming mode, in which mouse operations have different effects.

The editing tools are not available as menu commands.

	Open/Close workspace
	Selection Tool
	Connector Tool
Co	Constants and Terminals
GF	Basic functions
SF	Special functions
AP	App functions
MQ	MQTT functions
	SMS functions
	Text Tool
	Cut/Join
	Simulation Mode
	On-line test

Selection Tool

You can use the selection tool to select and move blocks, text and connecting lines. You can select objects individually with a left click; you can select multiple objects with [Ctrl] +Click, or you can use the mouse as a "lasso" to surround objects with a rectangle and capture them as a selection.

You can call the selection tool in any other tool by pressing the [ESC] key or by clicking on the icon in the programming toolbar.

Text Tool

This tool is used to insert or edit user-defined text objects in the programming interface. You can specify the font type, font size and font colour for each individual label.

Text can be set in Wrap Text format. You can adjust the size of a text typically by dragging the rectangle handles.

Cut/Join

This tool is used to cut and join connections between blocks. To cut a connection, left click to select the relevant line while the Cut/Join tool is active. The connection is replaced at the blocks by a reference to the partner block. The reference is labelled with the page number, block number and the I/O of the partner block.

Simulation Tool and Status Window

A toolbox pops up when you open the simulation mode. It contains:

- Icons (e.g. switches) for operator control of the inputs.

- An icon for the simulation of a power failure, for testing the switching response with reference to retentivity characteristics after a power failure.
- Icons (e.g. bulbs) for monitoring outputs.
- Simulation control icons and
- Time control icons.

Simulation control icons

	Start simulation
	Stop simulation
	Hold simulation (pause)

Time control

If you have programmed a time-sensitive circuit, you should use the time control to monitor the scan time of your circuit program.

	Start simulation for a specific time or number of cycles. Set the period and the number of cycles using the following icons.
1 1 Cycles	Setting the period and the time base for a time-limited simulation or setting a specific number of cycles
2018 / 11 / 24 11 : 53 : 42	Display of the current time in xLogicsoft
	Modify current time in xLogicsoft

Status Display

Layout of Inputs

The inputs are displayed in the form of key or switch icons. The name of the input is displayed below the icon. An open input represents an inactive switch. When you click on the icon, it is made active and the switch is shown in a closed state.

Icon for pushbutton I1, not actuated - open input

Icon for pushbutton I1, actuated - closed input

Icon for pushbutton I2, not actuated - open input

Icon for pushbutton I2, actuated - closed input

The layout of the Outputs

The status of an output is indicated by a light or dark bulb icon. The name of the output in your circuit program is displayed below this icon.

Status display of output Q1 - Output switched off

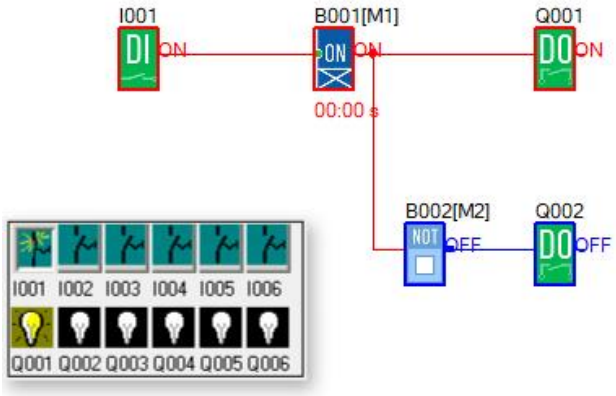
Status display of output Q1 - Output switched on

The output status only indicates the status as such. Here, you cannot switch an output by clicking on an icon. When your circuit program switches an output, the indicator

lamp is switched on; when the output is switched off, the indicator lamp is also switched off.

Prerequisite: The display of signal states and process variables is enabled under **Tools** → **Simulation**.

The coloured indication lets you identify the "1" or "0" status of a connecting line. The default colour for connecting lines carrying a "1" signal is red. The default colour for connecting lines carrying a "0" signal is blue.



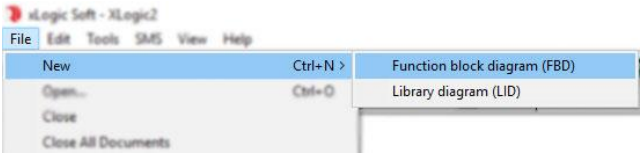
Basic Operation

This chapter will tell you how to write logic function program with xLogicsoft, how to simulate the program you write with xLogicsoft, how to communicate between PC and xLogic with xLogicsoft, how to copy system document of xLogic with xLogicsoft, and how to complete the update of an application and system program code. If you want to write a function block program you need to carry out the following steps:

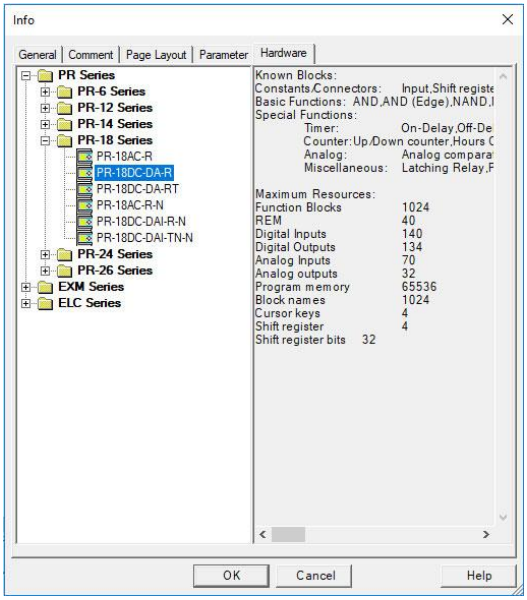
1. Create a new file/project
2. Add the function blocks required by your program to the programming workspace.
3. Set the properties of the blocks added if required.
4. Link the program blocks to in a logical order that your program requires.
5. Test your program using the Simulation mode(F3).

Create New File

To create a new file, click 'New' option from the 'File' menu. As shown in the following fig.



Select the xLogic hardware you are writing the program logic for.



Page Layout

In the **Page Layout** tab, you can specify how and on how many pages to print your logic program. You can preview the pagination in this tab. If you choose more than one program page, the page breaks are indicated by white lines on the programming interface environment.

Open an Existing Document

1. To open a document, click 'Open' of the 'File' menu option, or click once in the toolbar, shortcut key 'Ctrl + O' can also be used.
2. Click 'Open', browse for the file you wish to open.
3. Click the file you want to open with the left button of the mouse, then click "Open" button. After opening the document, you can modify or print the document.

Edit a Function Diagram Program

Place Function Block

When you set up a new document, you can write your control program in the new program interface area.

The method for placing a function block is as follows:

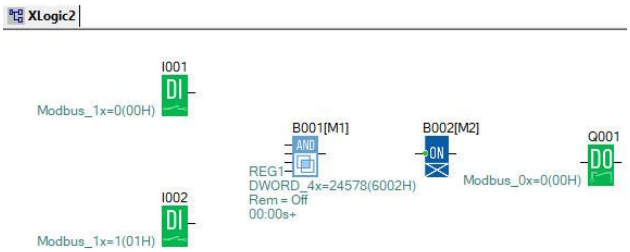
1. Choose a function group Co, GF, SF, AP or MQ from the list on the left-hand side of the workspace window or use the window tree to expand each topic to see the choices available. Details of what each block does can be found in the preceding chapter of this manual.

2. Click the block you need with the left button of the mouse.

3. Move the mouse to a place in the program interface area where you wish to place the block and click with left button to complete the placement.

4. Continue the above steps until you have placed all the blocks you need. You can always add/delete blocks later should you need to.

The program area should look similar to the one below:

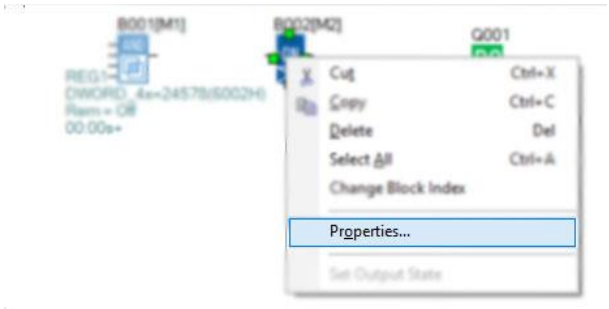


Edit Property of a Function Block

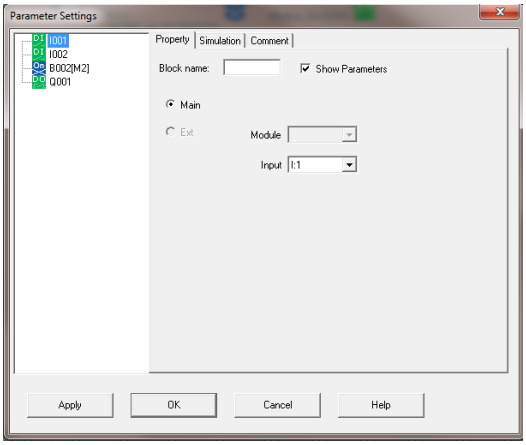
After putting all the function blocks in your work area, it's necessary to set up the properties of each function block. This is the most important step in writing a function block program.

Operation Method:

1. In the function block Program interface, click a block twice with the left button of the mouse, or select a block with a single click and press the right button of the mouse and click "Properties..." from the pop-up menu. As shown below:



2. You also can click the menu Edit-> Properties, if you have selected a single block or Edit-> Properties (All Blocks) to change the settings of the block/s.



For detailed instructions on how to configure a specific function block refer to this manual or you can click the Help button.

NOTICE:

Different blocks have different properties that require setting up, especially the special function blocks. Please read relevant sections above for a detailed explanation of the settings for the block you are using.

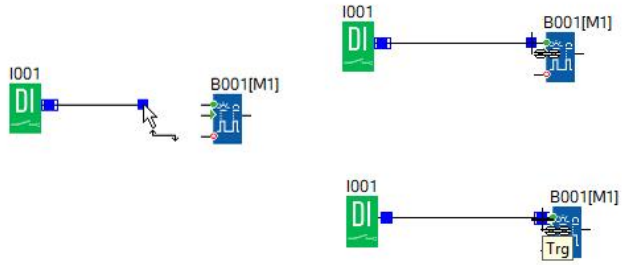
Connecting Blocks

After putting all the blocks needed into your workspace and configuring each block's settings are required, it's necessary to link your blocks together so as the program flow performs the required logical steps for your program.

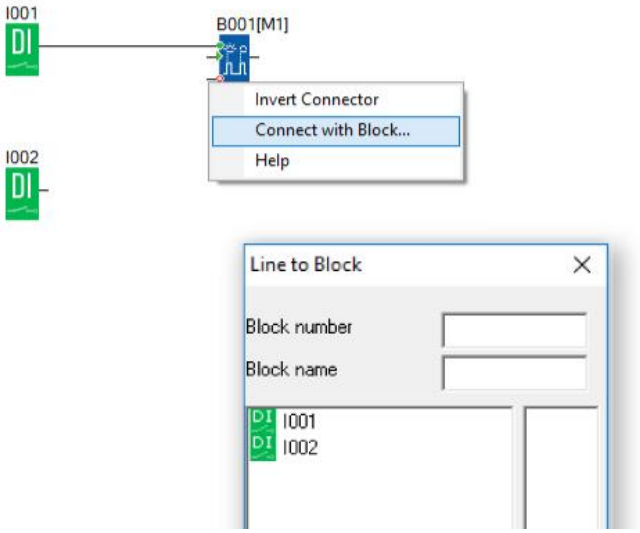
Operation Method:

1. Click on shortcut key on the left-hand side of your workspace. When the mouse icon becomes the shape of a pen, it can be used to link the blocks.

2. Move the mouse pointer to a block input or output pin and press the left mouse button. Keep the mouse button pressed down and then drag the mouse pointer from your selected source terminal to the target terminal. Release the mouse button to anchor the connecting line between blocks. While the connecting line is being drawn, it is shown as a straight line between the first terminal and the mouse pointer. Once it is connected to its target block, it appears as a combination of horizontal and vertical lines. The connections can be tidied up if required by using the Selection Tool and clicking on the connection line.



xLogicsoft offers you a further option of connecting blocks. When you right-click on the input or output of a block a shortcut menu appears, choose the **Connect with Block** command. This calls a selection list that contains all blocks available for your connection. Click on the relevant target block to select the connection and xLogicsoft will draw the connecting line. This method is especially useful for connecting a source to a target block over a greater distance on the programming workspace area.



Tips on Connecting Blocks

1. Hover the mouse pointer over a block and the type of block will be shown. Hovering over the input pins of a block will show the name of each pin.

2. To make it easier for you to connect blocks, a blue frame around the mouse pointer pops when it is in the correct position over the target pin.

Rules for Connecting Blocks

The following rules apply to the connection of blocks:

1. You can connect a single input to multiple outputs.

2. You cannot connect multiple inputs to a single output.

Special function blocks also have green "connectors". These do not represent connecting pins but are used instead for assigning the parameter settings.

3. Analogue I/O cannot be connected to Digital I/O.

Delete Function Block or Delete Link

Sometimes changes are required, and blocks or links need to be removed from your program. They can be deleted as follows:

1. Click on block or link to delete with your mouse.

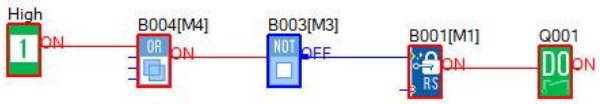
2. Press "Delete" on the keyboard or click the right mouse button and select option "Delete" from the pop-up menu.

Change Block Index

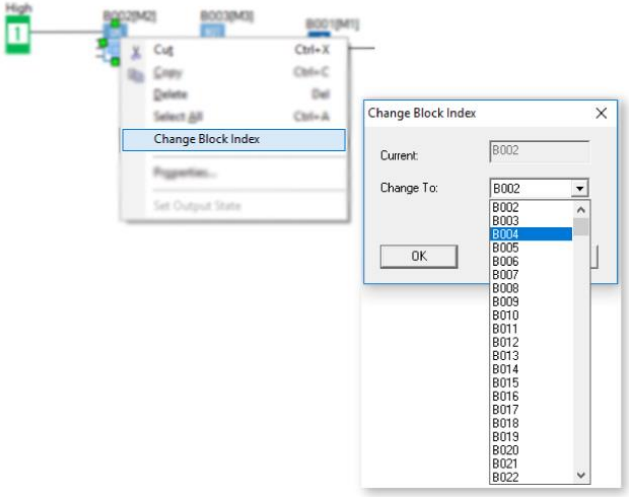
In a program cycle, program blocks are executed in order, starting with the smallest and ending with the largest.

For example, in the program below, the run sequence is B001, B002, B003, so the output Q1 will NEVER turn On because B001 is executed before B002.

If we change B002 to B004, then Q1 will switch On as expected. The program run sequence is B001, B003, and B004.



You can change the block number by right-clicking on the block and selecting the "Change Block Index" option, then select another block number. Note you cannot select a block number that is already used by another block in the program.



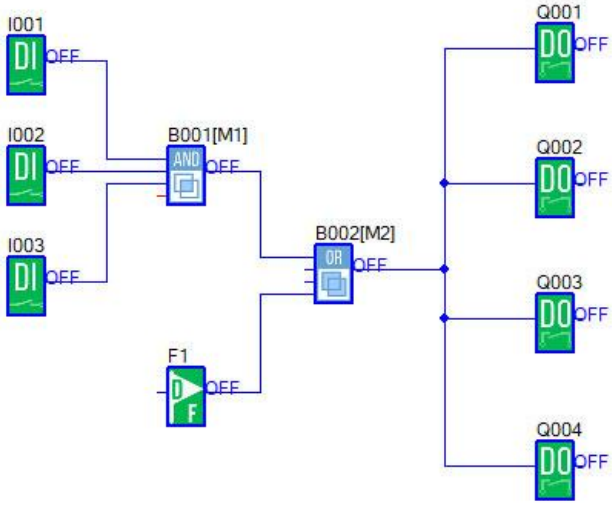
Program Simulation

xLogicsoft is used to create/edit a function diagram program and can also function as a simulation tool.

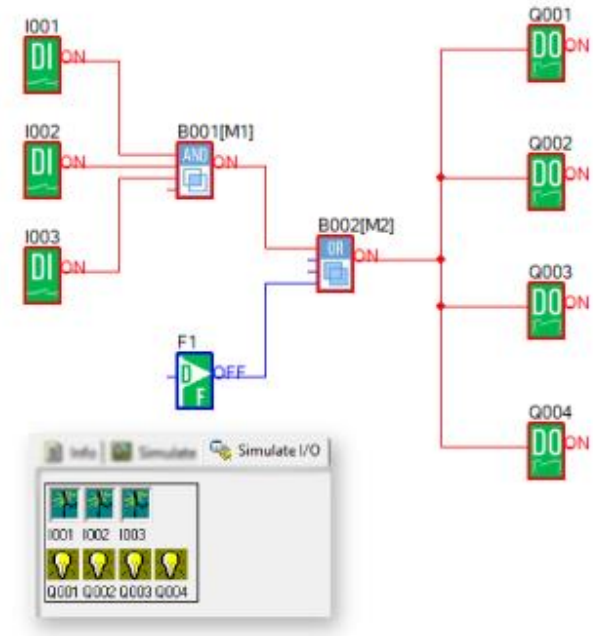
After completing your program, you can use the Simulation Mode to ensure your program is operating as you expected it to.

Operation Method:

1. Click "Simulation" under the "Tools" menu with the left mouse button or click in the simulation toolbar menu on the left-hand side of your workspace or press the F3 function key on your computer keyboard.



2. Clicking on a Digital Input block with the mouse will change the state of the input. The state displays "ON" and "OFF" on the output side of the block, so you can observe the state of input or output.



3. Click the button again, will terminate the simulation of your program.

Attn: In the above program, you can see the "ON" or "OFF" state of inputs and outputs and the output state and the current state of all timing and counting blocks. Through this simulation operation, you can check to ensure your program executes as you expect it to.

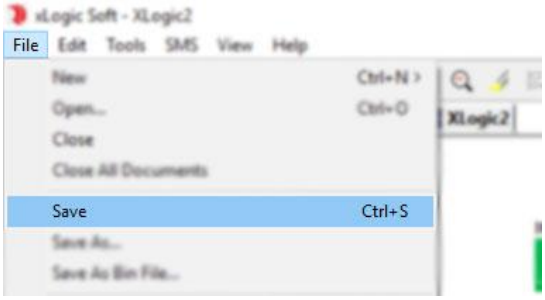
Save and Print

Operation Method of Saving a File

1. To save a Program, click option "Save" or "Save As" under the "File" menu with the left mouse button or click Save icon button on the toolbar.

2. You can select path and filename for your program file.

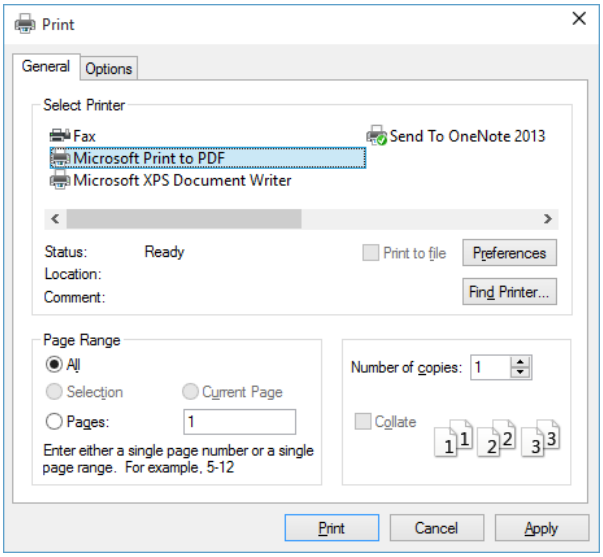
3. Click "Save" to save file.



How to Print Your Program File

1. Select "Print" from the "File" menu with the left mouse button.

2. The dialogue box as shown below appears, select your options and click OK to begin printing.



Modifying Password and Transferring a Program

Operation Method:

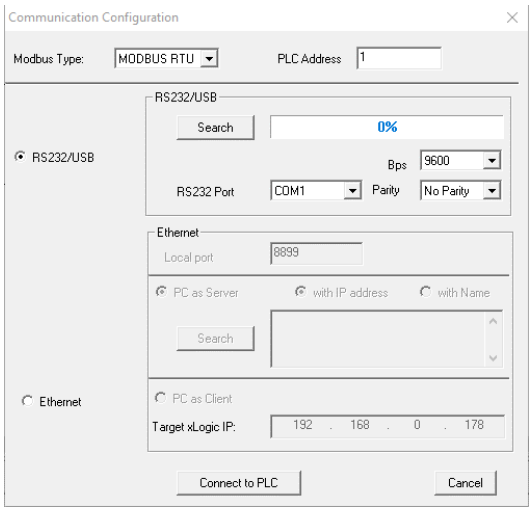
a. Firstly, connect your PC running xLogicsoft to your xLogic PLC using either an RS232 cable, USB cable (USB driver required) or Ethernet cable.

b. Open application software xLogicsoft, create a new project, then open computer communication connection by left-clicking the mouse button on Configuration under the Tools menu. Alternatively, you can click in the toolbar. A dialogue box appears as shown below, select your communication port and speed.

Option A. RS232 – This option is used when you use the ELC-RS232 or ELC-USB (USB driver required) cable.

Option B. Ethernet – This option is used when you use any Ethernet xLogic PLC.

Option C. GPRS – This option is used when you have established the GPRS connection between xLogic and PC.



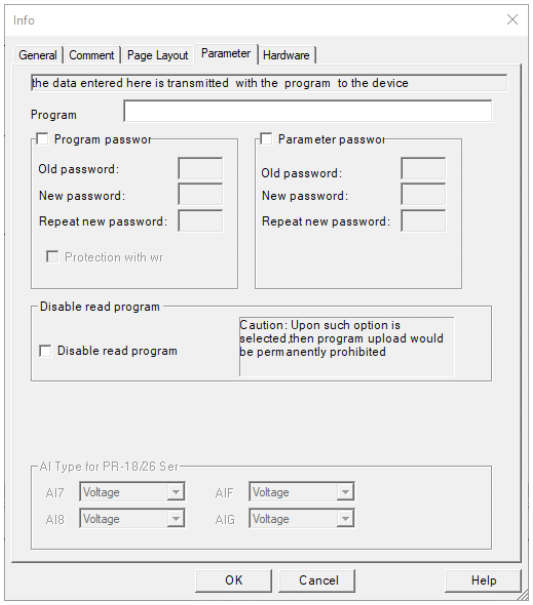
c. Click connect to PLC button to establish communications between PC and PLC.

How to Set a Password and Time and Date

Setting Password

xLogic permits you to set up a password for your Program. Only after entering the correct password, will you be allowed to write, read and modify your program.

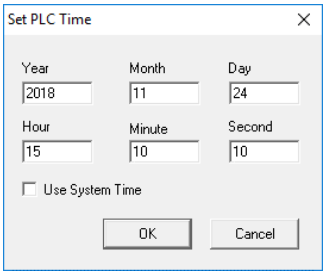
Click "File->Properties" and select "Parameter" tab to see the dialogue box below. You can input a password into the dialogue box as shown below.



Program password and parameter password can be set here. You can also directly set the parameter password by the HMI on the CPU if fitted using "Set.."->" Password"->"Set password".

Setting Time and Date

Click option "Tools->Transfer->Set Clock" with the left mouse button or click on the toolbar to show the time/date dialogue window shown below.



If you check the tick box "Use System Time" then the PC time and date will automatically be entered, otherwise adjust enter parameter as required and when done press the "Ok" button.

If the setting has been successful, the dialogue box below will show "Succeed" message.

Transferring a Program

Downloading a Program to xLogic

After you have created your xLogic control project, you need to download it to the PLC.

a. Connect your PC running xLogicsoft to your xLogic PLC as mentioned above.

b. Click option "Transfer->PC->xLogic" under menu "Tools" or click the icon in the toolbar.

c. Your program will be downloaded to your xLogic PLC and a Transfer Indication will show percentage complete.

NOTICE:

After the program is downloaded, xLogic will run it automatically.

Uploading a Program from xLogic

If you need to retrieve a program currently in the xLogic PLC, you can use the Upload procedure to do so.

a. Connect your PC running xLogicsoft to your xLogic PLC as mentioned above.

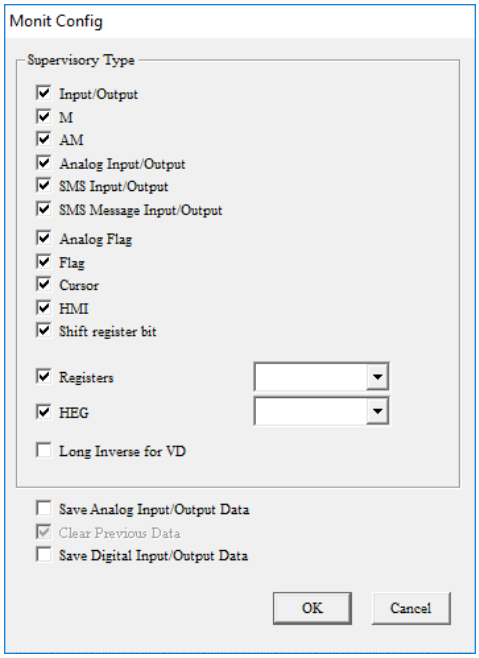
b. Click option "Transfer->xLogic->PC" under menu "Tools" or click the icon in the toolbar.

The program will be uploaded and displayed in your xLogicsoft.

On-line Monitoring and Program Checking

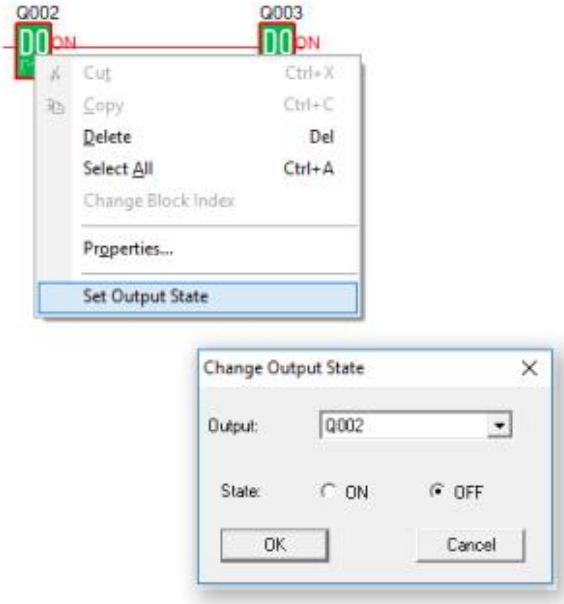
After you are satisfied with your program simulation and have downloaded it to xLogic CPU, you can also perform an online test of the circuit program. An online test is similar to simulation in that you can view inputs and outputs and block parameters. It differs, however, in that you are testing the program running in the xLogic with "live" inputs rather than testing the program on the PC with simulated inputs.

Multiple registers can be accessed individually by clicking "Tools->Set Monitor Config".



Monitor mode:

Under monitor mode, the user can change the state of the spare outputs Q002 and Q003 because the input pin of the block is not connected to any other blocks. This is achieved by Right clicking your mouse on the input side of the block and selecting Set Output State from the pop-up window.



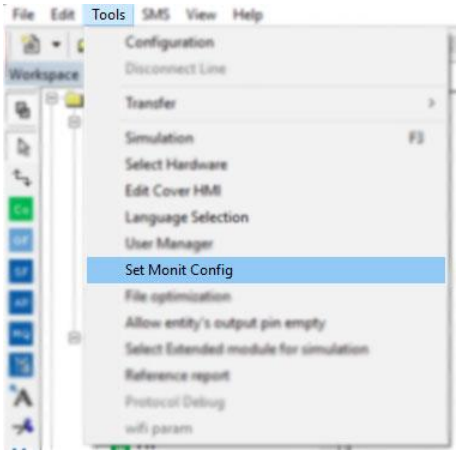
Right click "Q2" and then click "Set Output State".

Click "OK" button and the Q2 of xLogic will be turned off.

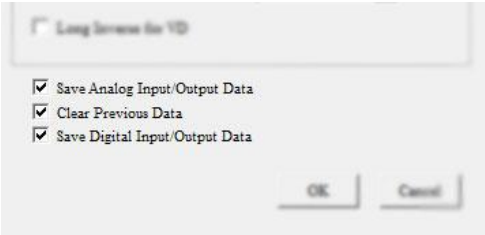
How to Transfer Monitoring data (AI/AQ) to an Excel file?

Example program:

1. Select Tools-> Set Monit Config.



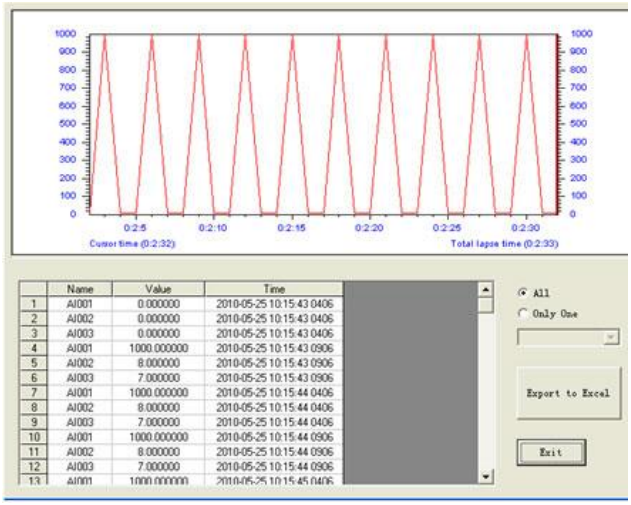
2. Select which data types you wish to save, Analogue, Digital or Both.



Caution: If the "Clear Previous Data" is selected as well, then previously saved data will be over-written. Uncheck this box if you wish to append new data to your existing historical data.

3. Click here to enter Monitor Mode

4. View the data by clicking View-> Monit AIO or Monit DIO to view the live data.



5. Click the to transfer data to an Excel file. The file will contain ALL Analogue or Digital tags, their data and a time and date stamp.

	A	B	C	D	E	F	G
1	Value_Inc	IO_Name	IO_Value	Value_Time			
2	1	AI001	0	2010-05-25	10:00:34	0234	
3	2	AI002	0	2010-05-25	10:00:34	0234	
4	3	AI003	0	2010-05-25	10:00:34	0234	
5	4	AI001	1000	2010-05-25	10:00:34	0734	
6	5	AI002	7	2010-05-25	10:00:34	0734	
7	6	AI003	6	2010-05-25	10:00:34	0734	
8	7	AI001	1000	2010-05-25	10:00:35	0234	
9	8	AI002	7	2010-05-25	10:00:35	0234	
10	9	AI003	6	2010-05-25	10:00:35	0234	
11	10	AI001	1000	2010-05-25	10:00:35	0734	
12	11	AI002	7	2010-05-25	10:00:35	0734	
13	12	AI003	6	2010-05-25	10:00:35	0734	
14	13	AI001	1000	2010-05-25	10:00:36	0234	
15	14	AI002	7	2010-05-25	10:00:36	0234	
16	15	AI003	6	2010-05-25	10:00:36	0234	
17	16	AI001	1000	2010-05-25	10:00:36	0734	
18	17	AI002	7	2010-05-25	10:00:36	0734	
19	18	AI003	6	2010-05-25	10:00:36	0734	
20	19	AI001	1000	2010-05-25	10:00:37	0234	
21	20	AI002	7	2010-05-25	10:00:37	0234	
22	21	AI003	6	2010-05-25	10:00:37	0234	
23	22	AI001	1000	2010-05-25	10:00:37	0734	
24	23	AI002	7	2010-05-25	10:00:37	0734	
25	24	AI003	6	2010-05-25	10:00:37	0734	
26	25	AI001	1000	2010-05-25	10:00:38	0234	
27	26	AI002	7	2010-05-25	10:00:38	0234	
28	27	AI003	6	2010-05-25	10:00:38	0234	
29	28	AI001	1000	2010-05-25	10:00:38	0734	
30	29	AI002	7	2010-05-25	10:00:38	0734	
31	30	AI003	6	2010-05-25	10:00:38	0734	
32	31	AI001	1000	2010-05-25	10:00:39	0234	
33	32	AI002	7	2010-05-25	10:00:39	0234	
34	33	AI003	6	2010-05-25	10:00:39	0234	
35	34	AI001	1000	2010-05-25	10:00:39	0734	
36	35	AI002	7	2010-05-25	10:00:39	0734	
37	36	AI003	6	2010-05-25	10:00:39	0734	

8. How to Configure an Ethernet CPU?

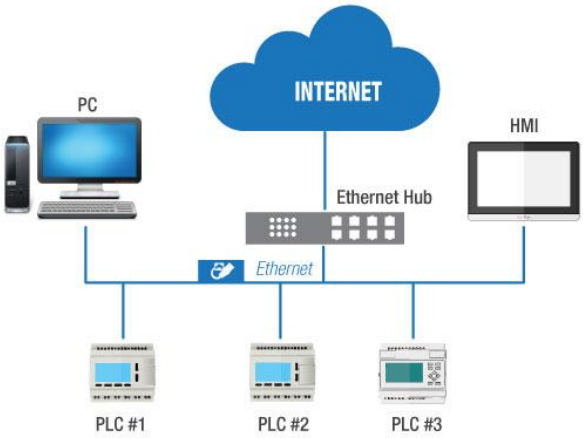
Some of the xLogic CPUs have a built-in Ethernet port and some models also contain a built-in web server. This chapter is concerned with the configuration of an Ethernet CPU, which has not got a built-in web server. The xLogic CPUs this applies to is:

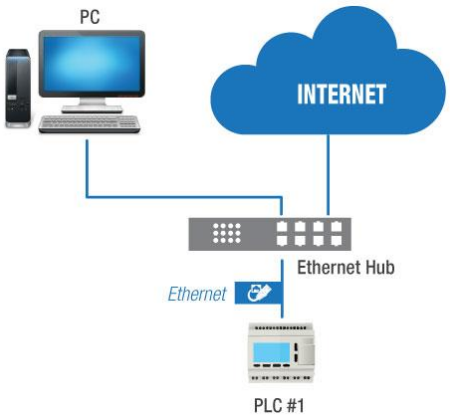
- ELC-12DC-DA-R-N
- ELC-12AC-R-N
- ELC-22DC-DA-R-N

To configure the Ethernet settings for any of the above you will require our free Ethernet Device Manager software. See Configuring Your Ethernet xLogic CPU for details.

Ethernet network

If the application requires a system where more than one main xLogic CPU is needed and these CPUs have to communicate with each other over an Ethernet network, then our Ethernet xLogic CPUs are perfect for such a task. What's more, is the Ethernet network can also be used to upload/download projects from/to the xLogic CPUs. Furthermore, the visualization of the whole system is possible using our FREE SCADA software.



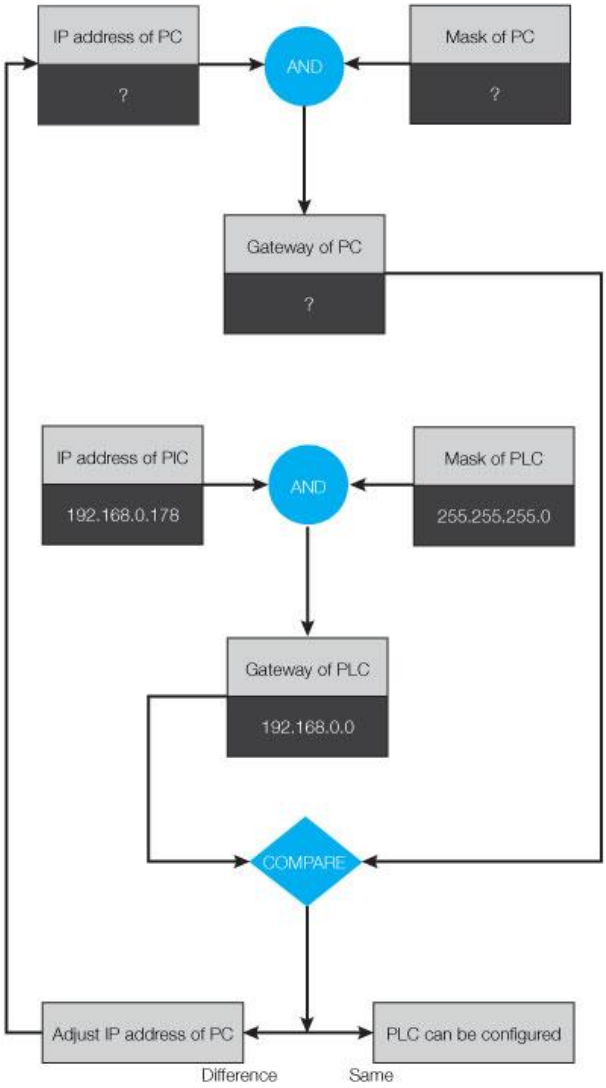


The Ethernet CPU default IP address is 192.168.0.250 and the network mask is 255.255.255.0.

To ensure correct communication with our xLogic Ethernet CPUs, all devices on the network must be on the same network segment and each device on the network must have a different IP address otherwise conflicts will occur.

When using Ethernet Device Manager software's Search function, it will find any xLogic Ethernet CPU connected to the network regardless of its IP settings. Furthermore, the software allows you to configure your xLogic CPU network settings to match your PC's network segment if you wish.

If you wish to use the network segment of the CPU, then you must change the settings of your PC.



Ethernet CPUs with Built-in Webserver

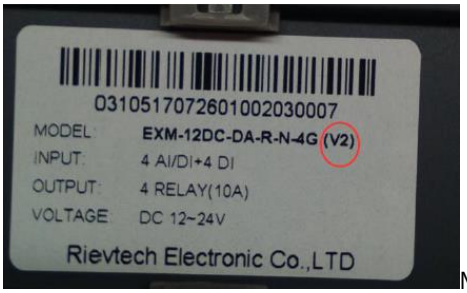
The new range of Ethernet CPUs from Rievtech have a built-in webserver, which is incorporated in the following models:

- ELC-12DC-DA-R-N
- EXM-12DC-DA-R-N(-4G)
- PR-12DC-**-**-N
- PR-12AC-R-N
- PR-18DC-**-**-N
- PR-18AC-R-N
- PR-26AC-R-N
- PR-26DC-***-**-*G
- PR-26DC-***-**-*WIFI

How to Distinguish Between New and Old CPU Versions

Method A

If the label on the bottom of the CPU plastic housing has a part number followed by (V2):



Method 2

Using xLogicsoft software, select Tools…transfer…Get PLC version. If the hardware version is 1 then it is a new version:

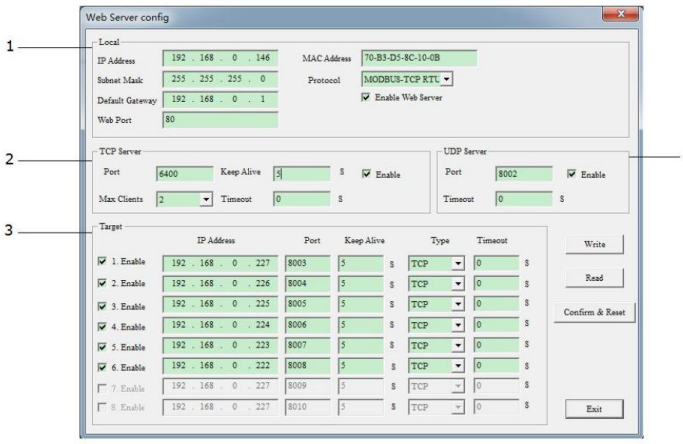
Differences Between Old and New Versions of CPU

Items	New Version	Old Version
Network Configuration Tool	xLogicsoft or eSMSConfig using menu option Tools… Web server config	DeviceManager Software
TCP Server/Client Mode	Can work as BOTH a Server and a Client at the same time	Can work as EITHER a Server or a Client
TCP Connections	Maximum of 8 concurrent connections	Maximum of 1 connection

Built-in Webserver	Yes	No

How to Configure the Network Parameters via Software

You can use xLogicsoft to configure the Ethernet network parameters from the menu option, Tools…transfer…Web server config:



1. Local CPU Network Settings
   a. IP Address
   b. Subnet Mask
   c. Default Gateway
   d. Web Port
   e. MAC Address
   f. Protocol
2. TCP Server Settings – CPU as a TCP Slave
   a. Port for TCP Server
   b. Keep Alive – Not Used
   c. Max Clients – Total TCP connections is 8. If you set this to 4, you can have 4-clients and 4-servers connected simultaneously. If set to 8, then the CPU can no longer work as a Server.
   d. Timeout – 0s means that no timeout is set, and the CPU will remain online. Any other value and the CPU will restart the connection if no transmission of data has occurred during the set period.
   e. Enable – Check to enable the CPU to act as a Server.
3. Target – Details of Server connections. This is determined by the setting in 2c above. If 2c is set to a value of 2, then 6 Server connections will be available for configuration in this section. You need to tick the "enable" option and input the remote server IP address and port number.
   a. Fill in the IP Address of each server (Slave) and the Port number on which communication will occur.

NOTICE:
Each connection MUST have a unique port number from the others that are used.

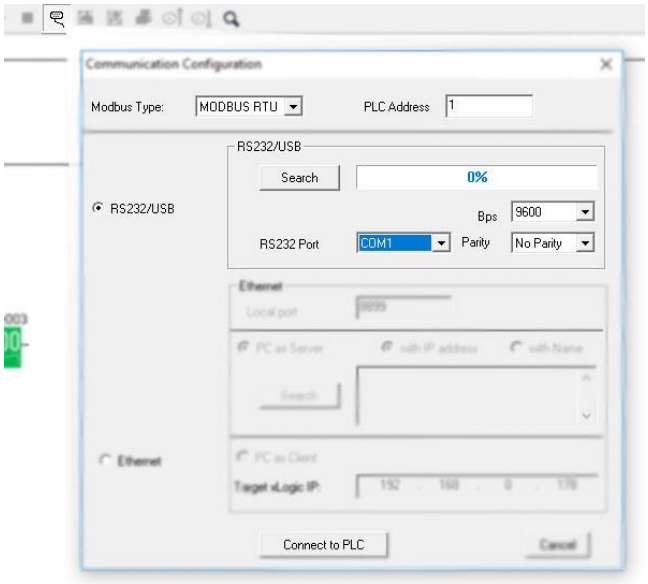
4. UDP Server

The CPU can work as a UDP server as well, you can tick the enable option.

After you have finished the parameter configuration, you need to download the settings into the CPU by either a serial or Ethernet connection as you would to download a program. Finally, you need to click the "Confirm and Reset" button, then the CPU will restart, and the new settings will be available.

Create a connection between the PC and the CPU by a serial cable (RS232/USB cable) or Ethernet (You can check the CPU IP address and server port number on the LCD menu).

Connect the CPU with PC through the USB cable: Click the open com port option, and select the COM port of the USB cable, here the port number is COM3, then click the button "connect to PLC".

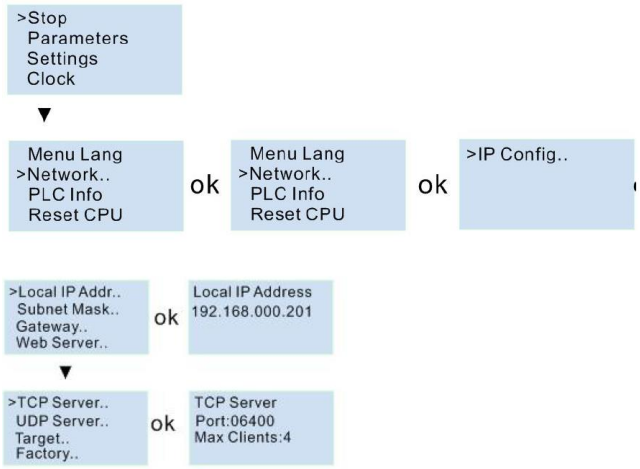


If you have no USB cable there, you can use the Ethernet connection.

First check the PLC address using the LCD menu.

Get the IP address from:

>Network -> IP Config. -> Local IP ADDR. -> Local IP (192.168.0.245)



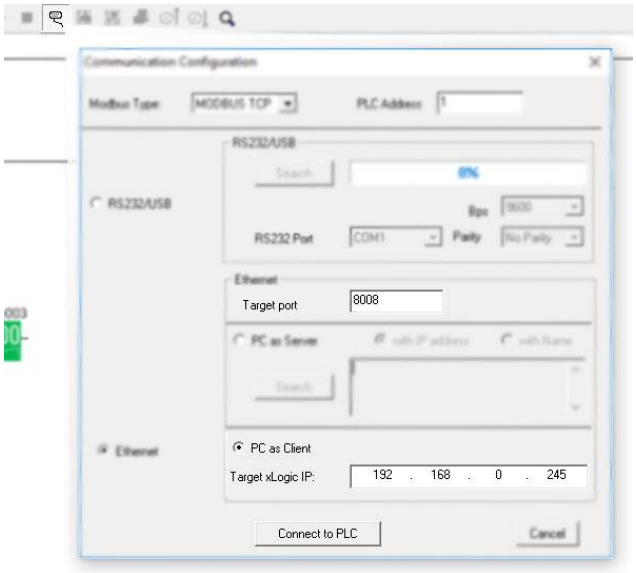
Get the TCP server port from:

>Network -> IP Config. -> TCP Server. -> TCP Server Port (6400).

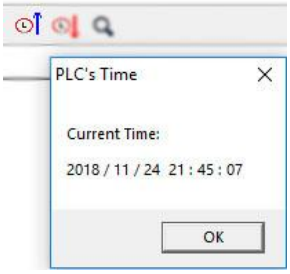
Then you can use the Ethernet option (PC as Client) to connect with CPU.

IP: 192.168.0.245

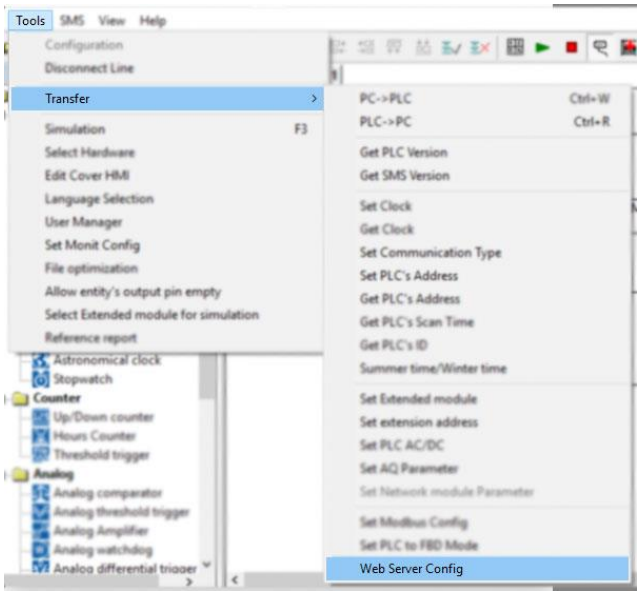
Port: 8008



After the connection is created, you can click Get Time icon to confirm the communication is OK.



Now you can configure the Network parameters by using the menu Tools-> transfer->Web server Config:



You can read the current parameters from the PLC.

How to View and Configure the Ethernet parameters via the LCD Panel

You can view and modify the network parameters through the LCD menu.

You can modify the Local IP address, subnet mask, gateway from the LCD menu:

The web server port also can be modified, and can also be disabled or enabled:

You can view the MAC address from here, but it cannot be modified:

TCP server port settings and allowed TCP clients settings:

View and modify the UDP server:

Set the Target server IP address and port number:

Reset the IP settings to factory defaults:

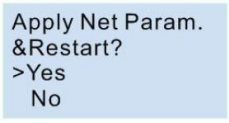
"RST IP Config", the IP address will return to 192.168.0.201, and the TCP server port number will be 6400.

"RST Log on" the web server login name and password will be set back to "admin".

Modify the IP address:

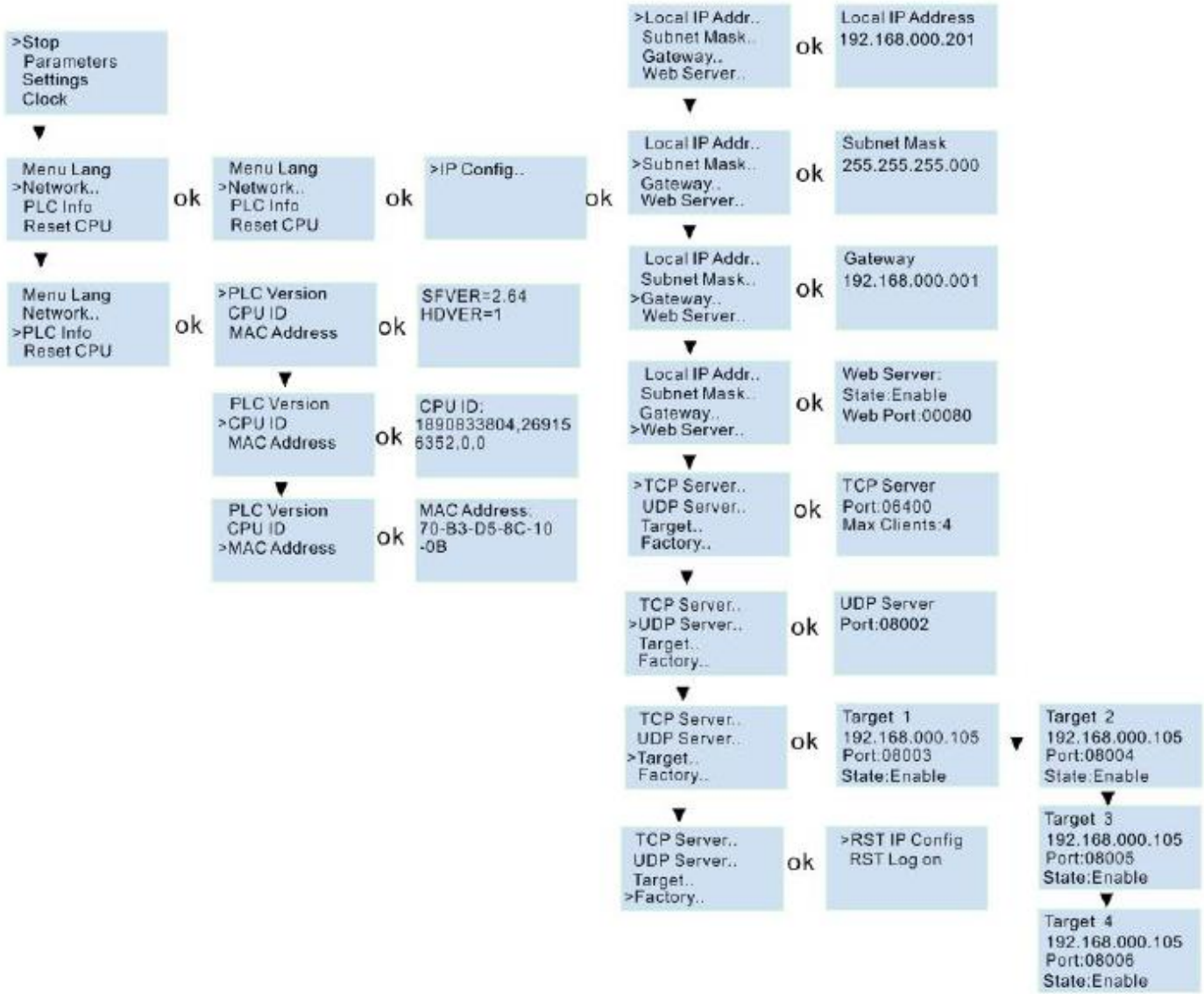
Press OK to enter into the modify mode, the cursor will flash at the start of the address. Then you can move the cursor by using the Left or Right buttons and change the value at the cursor position by pressing the UP/DOWN buttons. Press OK to save and confirm your changes.

After you confirm the parameters with the OK button, the settings are not enabled, until the CPU is restarted. So, when you leave the settings the LCD will show you:



You need to select "Yes" and press OK. The CPU will restart, and the new settings will be applied.

LCD Menu Structure



How to Create Communication between the CPU and a PC via Ethernet?

To communicate with the CPU, you can use the TCP/IP protocol. The CPU can work as a TCP Server and a TCP Client at the same time, and it can also work as a UDP Server or UDP Client as well.

NOTICE:

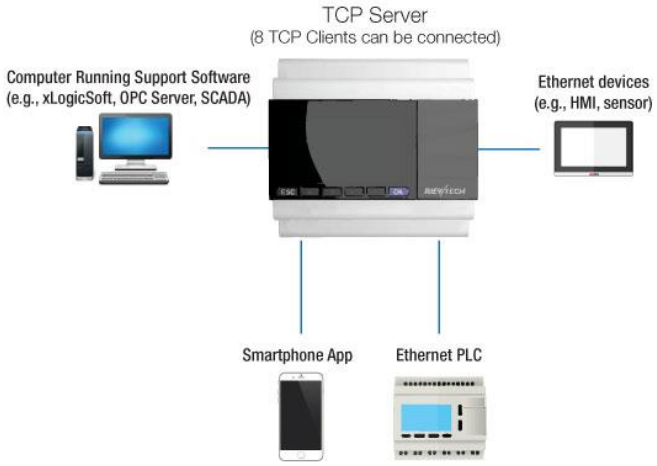
1. The Ethernet CPU can work in TCP and UDP mode. You can create a connection between our Ethernet CPUs, and also create a connection with another Ethernet device based on TCP or UDP.

2. The communication protocol via Ethernet is MODBUS TCP or MODBUS RTU of our PLC, so if you want to communicate with another factory device through Ethernet, you need to make sure the device also supports MODBUS TCP or MODBUS RTU.
3. Our Ethernet PLC (built-in web server version) can work either as a Master or a Slave.

CPU Working as a TCP Server (Slave)

The maximum TCP connections is 8, so one CPU allow a maximum of 8 TCP Clients (Masters) to connect at the same time. And each TCP connection is treated totally independent of the others.

For example:



All the 8 Clients can monitor and control the CPU at the same time.

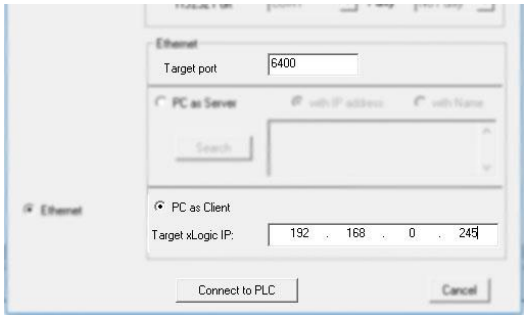
TCP Client 1 - Communication with xLogicsoft (PC is Client)

First check the PLC IP address and TCP Server Port settings using the CPU LCD menu.

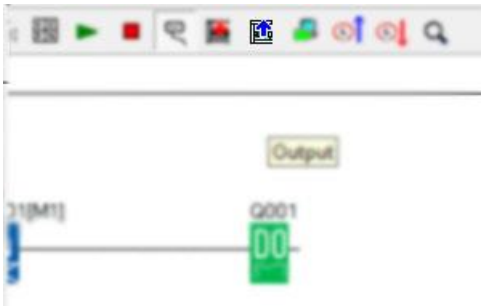
Get the settings from:

>Network -> IP Config. -> Local IP ADDR. -> Local IP (192.168.0.245)

Use these settings in the xLogicsoft Communication Configuration window to enable connection to the CPU from the PC.

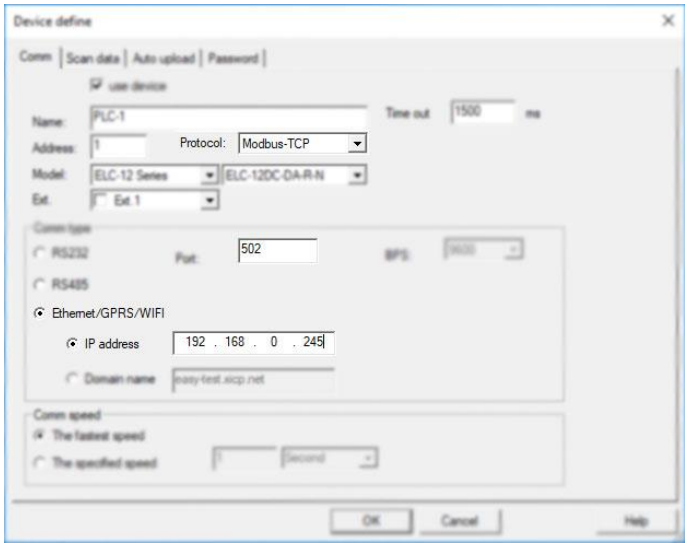


After the connection is established, you can download/upload the program and monitor the program just like using the USB cable connection.



TCP Client 2 - Communication with EasyScada (PC is Client)

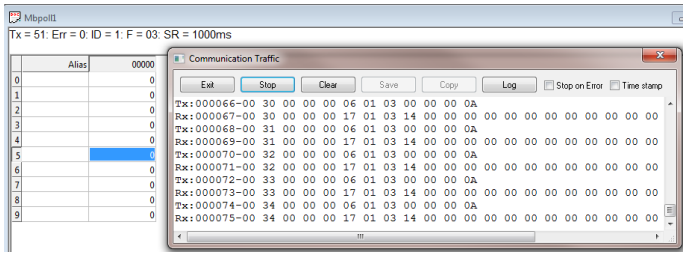
You need set the correct IP address and com port in the Device Define window of the EasySCADA.



TCP Client 3 - Our xLogicApp (Smartphone is TCP Client)

Interface configuration

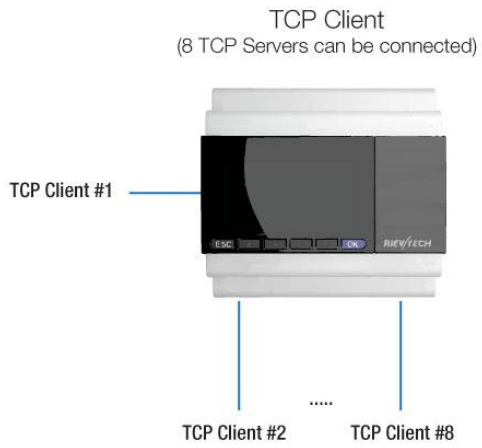
TCP Client 4 - Modbus Poll as TCP Client



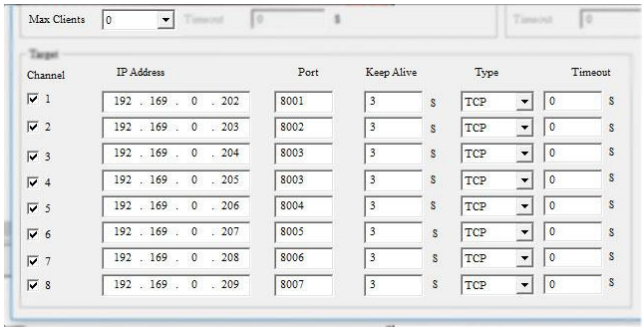
CPU Working as a TCP Client (Master)

The maximum number of TCP connections is 8, so one CPU can allow a maximum of 8 TCP Servers (Slaves) to connect with the CPU at the same time. And each TCP connection is totally separately.

For example:



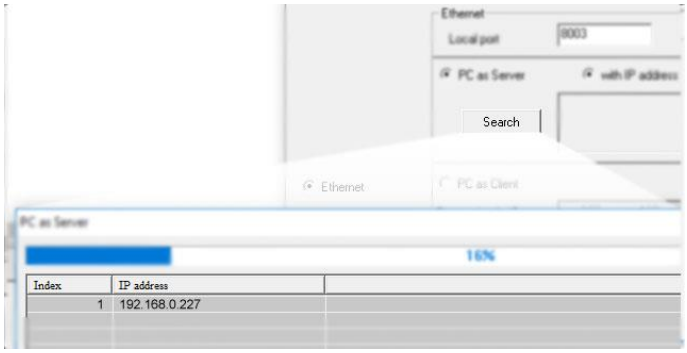
All the 8 Servers can be connected, and the data transmission can be processed separately.



If we set Max Clients to 0, then the CPU will only act as a TCP Client, and 8 TCP servers can be connected at the same time.

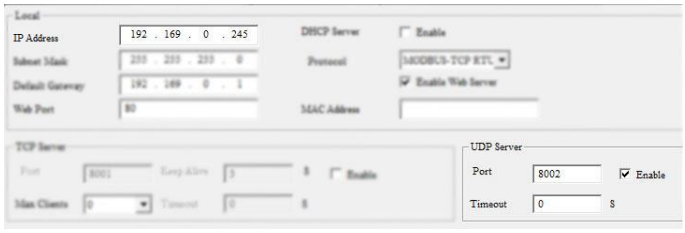
IMPORTANT: EACH TCP SERVER MUST HAVE A UNIQUE PORT NUMBER SETTING. SO, IF YOU HAD A NETWORK WHICH HAD THREE MODBUS/TCP SLAVES, THEN ONLY ONE COULD USE THE MODBUS/TCP DEFAULT PORT 502.

xLogicsoft works as TCP server and waits for the CPU to logon (The PC IP address is 192.168.0.227):



CPU as a UDP Server

The CPU can work as a UDP server, you can use your software to connect with it via UDP or make several CPUs communicate via UDP too.



After the connection is created, the CPU and the software can communicate based on the MODBUS TCP protocol.

How to Logon to the built-in Web Server

Only the new ELC-12DC-DA-R-N and PR-Ethernet series have a built-in web server. You can use a PC, mobile

phone or any device which has a built-in web browser to control the PLC.

Supported browsers:

- Chrome
- IE
- Firefox

Please ensure that your browser is the latest version.

Supported devices:

- PC
- iPhone
- iPad
- Android mobile phone
- Android pad

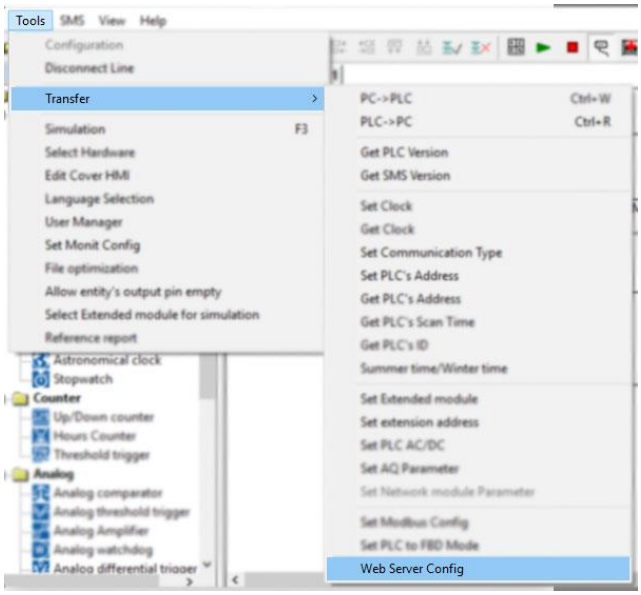
Supported languages:

- English
- Chinese
- German
- French
- Italian
- Spanish

How to Enable the Web Server

Open xLogic software, choose the correct Ethernet model and connect xLogicsoft to the PLC. You can use a serial port or network port to connect to the PLC.

Tools-->Transfer-->Web Server Config



Click Read button to read web server configuration.

The default configuration is as follows:

The web server is enabled by default. If you want to disable the web server access, you need only untick the option "Enable web server" and it will be disabled.

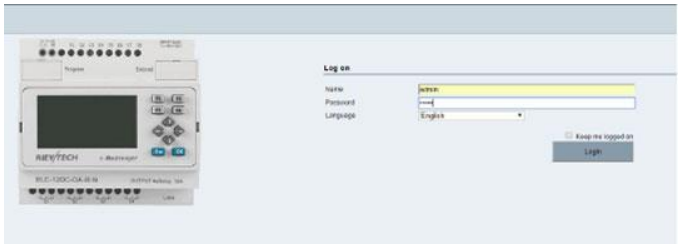
How to Logon to the Web Server

First, obtain the IP address of the PLC and enter into the URL bar of a supported web browser and the login web page will appear.

The default IP of PLC is 192.168.0.245, the Default port of PLC is 8008. So, you can use the default IP address to log on to the web server, you can also use the IP and port of PLC to connect the xlogic software to the PLC and configure web server parameters.

If you have changed the IP address, you can view the IP address by the LCD menu

>Network -> IP Config. -> Local IP ADDR. -> Local IP



Name: The default name is **admin**.

Password: The default password is **admin**.

Language: Our web server supports six languages. You can choose your language.

Keep me logged on: When you choose "Keep me logged on", then you click on "Log on" and you log on to the web server successfully. You will always stay in the login state. If you close the browser, then open it again and input the IP address, you will be automatically logged in. You can click on the "Log off" to exit the login status.

Log on: Click on it to log on to the web server.

Check CPU information

After logging on to the web server, the web server will display the information of the CPU.

CPU name: Current CPU name.

Current status: Display the status of PLC

CPU clock: When you click on the "CPU information", "CPU clock" will display the current time of the PLC. Select the "PC synchronized", then click on "Set", the PC time will be written in PLC.

CPU Addr: Display CPU ID address. You can change the ID of the PLC.

Example:

If you input "5", and click on "Set", the CPU ID will be set to 5.

CPU model: Display CPU model.

Firmware Ver: Display Firmware Version.

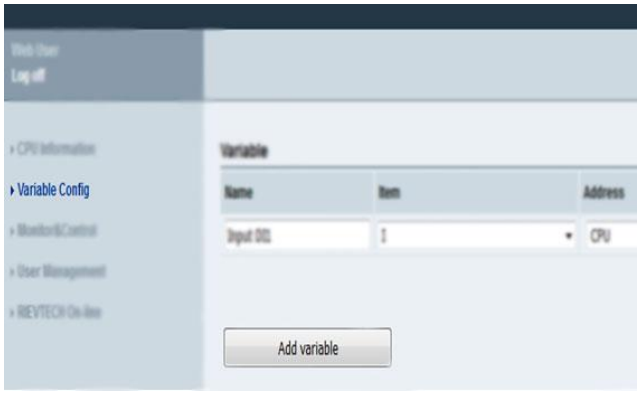
Hardware Ver: Display Hardware Version.

GPRS connection: (For EXM series CPU) Display "Connected" or "Not connected".

GSM signal: (For EXM series CPU) Display GSM signal.

Variable Configure:

"Variable config" interface

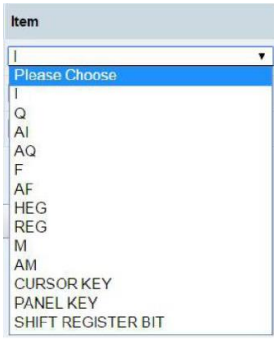


You can click on "Add Variable" to add a variable. And you can change or delete a variable as well.

Up to 16 variables can be configured:

Name: You can define a name for the variable. (Maximum of 8 characters)

Item: You can configure 13 kinds of variables. (These variables are in xLogicsoft)



Address:

Type: The data length of the variable. Includes BIT, WORD, DWORD. If you use a timer or counter, you should choose a TIMER or COUNTER.

Display Format: Display format includes: BOOL, SIGNED, UNSIGNED, HEX, BINARY.

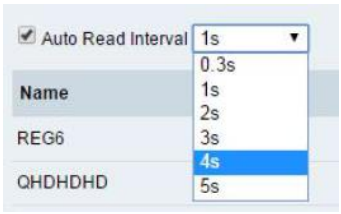
DEL: Click on ✖ to delete variables.

Save Variable: After finishing the configuration of the variables, you have to click on "Save Variable", then you can monitor and control variables using "Monitor & Control".

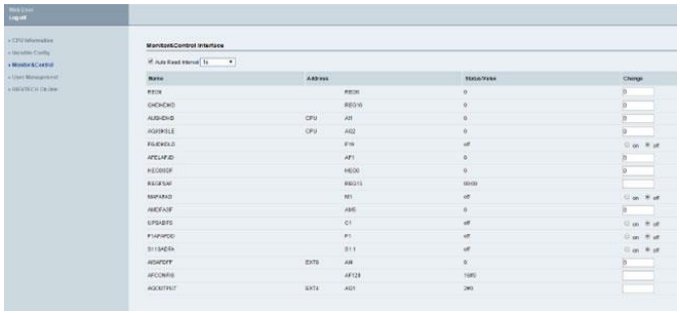
Monitor & Control

On this page, you can monitor and control variables. The web server refreshes data automatically.

Auto refresh: choose "Auto Read Interval" and choose the refresh time.



The web server refreshes the current value of variables every 0.3s-5s. The default auto refresh time is 1 second.



Name: Display the name of a variable.

Address: Display the address of a variable.

Status/Value: Display the current status or value of variables.

Change: Change the current status of a variable. When you change the current

Status: Status of a variable, you have to click on the corresponding "Set" button.

Changing the state of a variable:

Variable I:

You can't change the status of variable I.

Variable Q:

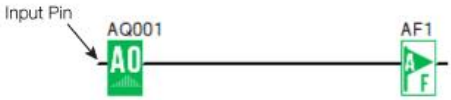
When the input pin of output block Q is not connected, you can change the status of variable Q.

If the input pin of output block Q is already connected with other blocks, you can't change the status of variable Q anymore.

Variable AI:

You can't change the value of variable AI.

Variable AQ:



When the input pin of analogue output block AQ is not connected, you can change the value of the variable AQ.

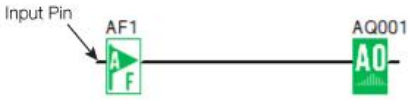
If the input pin of analogue output block AQ is already connected with other blocks, you can't change the value of that variable anymore.

Variable F:

When the input pin of digital flag variable F is not connected, you can change the status of variable F.

If the input pin of the digital flag block F is already connected with other blocks, you can't change the status of that variable anymore.

Variable AF:



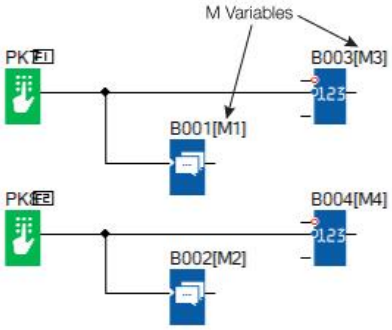
When the input pin of analogue flag block AF is not connected, you can change the value of variable AF.

If the input pin of analogue flag block AF is already connected with other blocks, you can't change the value of that variable anymore.

Variable HEG:

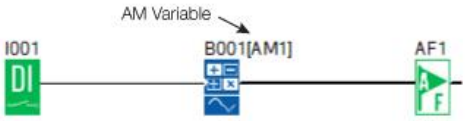
Cannot be set, can only be read.

Variable M:



You can't change the status of a variable M; it can only be read.

Variable AM:



You can't change the value of a variable AM; it can only be read.

Cursor key:

You can't change the status of a cursor key; it can only be read.

Panel key:

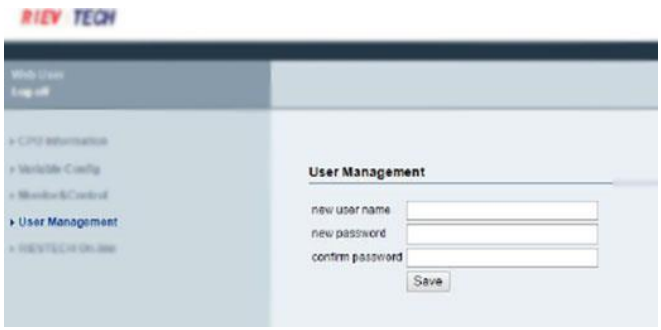
You can't change the status of a panel key; it can only be read.

Shift register bit:

You can't change the status of a shift register bit; it can only be read.

User Management

User management interface



You can set a new username and password on this page. If you have set a new username and password, the old username and password can't be used. The web server just has a username and a password.

If you forget the username or password, you can reset it by using the LCD menu.

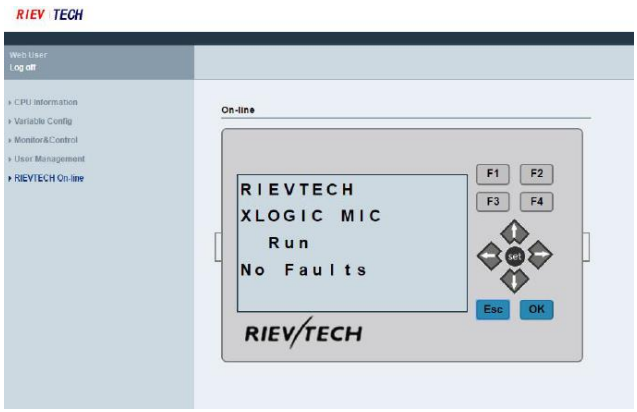
You can reset to factory settings by using the menu option: Network-> IP Config. -> Factory-> RST Log on

After you reset the logon, the username and password will be set back to the default "admin".

RIEVTECH On-line

RIEVTECH On-line Interface

You can monitor and operate the LCD panel remotely.



The CPU only can only get a momentary trigger from the virtual keys, it cannot get a continuous signal, that means if you press the OK key for 3 seconds, the CPU does not know how long it was pressed for, it only knows that is was pressed.

So, if you want to realize the pressing of the OK key for 3 seconds to change the parameters in a text message block on the hardware, you need press the "set" key on the virtual panel instead.

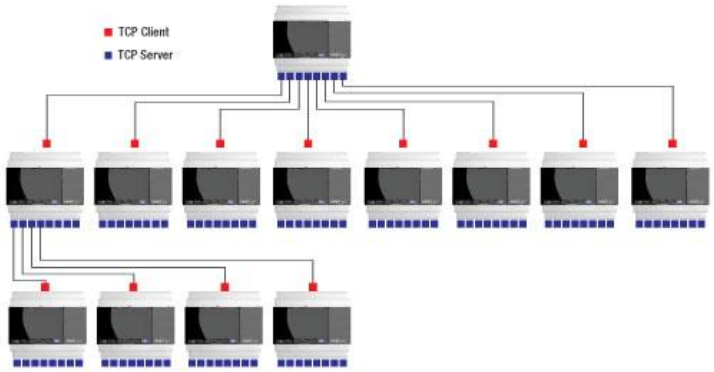
Press "Set" = Press OK key for 3 seconds then the parameters in the text message block can be modified in the usual way using the cursor keys.

How to Establish the Communication between new Ethernet CPUs

We can make an Ethernet connection based on a TCP or UDP connection. This is achieved using the following method.

Configure and Create a Connection (TCP or UDP)

Each CPU has 8 TCP connections, and can work both as a TCP Server and a TCP Client, so a lot of CPUs can be connected in a network.

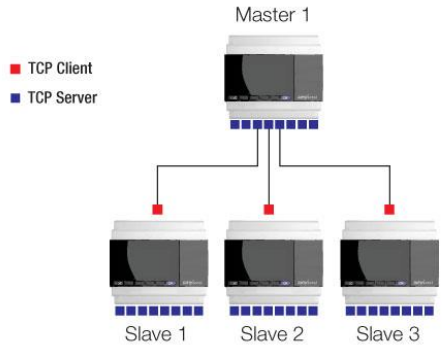


Programming and Data Transmission

We need to use the "Modbus Read and Write" function block in the Master CPU. The TCP server or the TCP client CPU can work as master and/or slave depending on what you need to achieve.

Example 1: One Master CPU (TCP Server) connected to Three Slave CPUs (TCP Clients)

The connection sketch:



Requirement:

1.If I1 of the Master CPU is ON/OFF, then the output Q1 of Slaves 1 to 3 are ON/OFF.

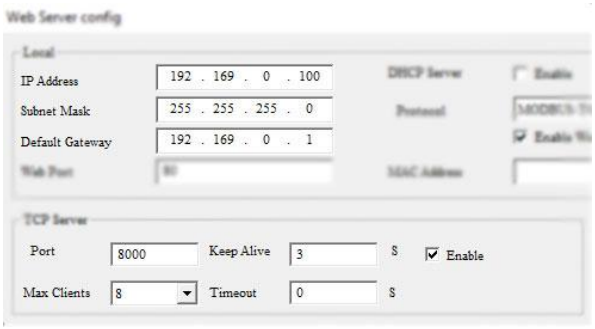
2.Read the analogue input, AI2 value from the slave1 through slave3 and store in variables AF1 to AF3 of the Master CPU and display.

Step1: Configure the IP configuration of the PLCs.

Master

IP: 192.168.0.100

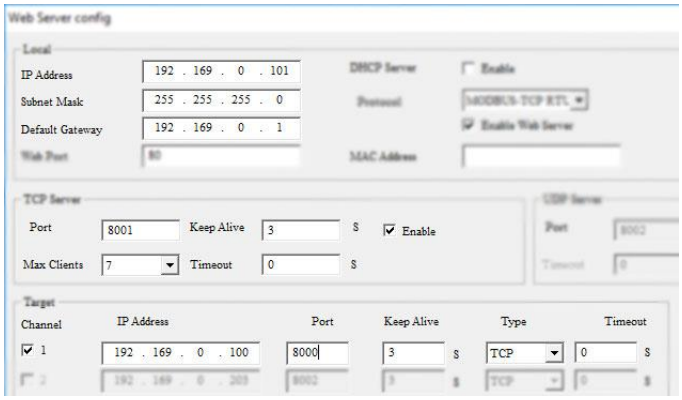
TCP port: 8000



Slave1:

IP: 192.168.0.101 Target server: 192.168.0.100

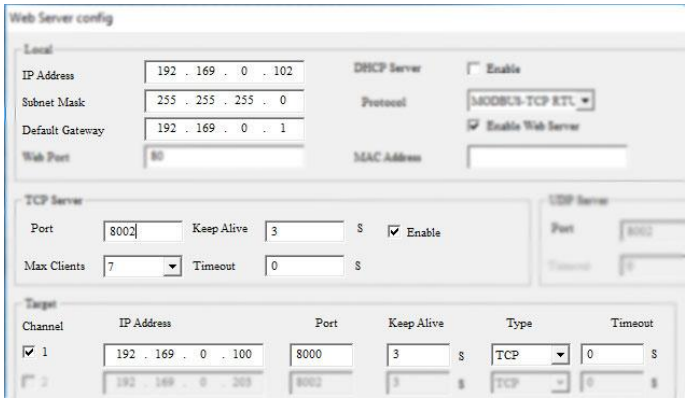
Target Port: 8000



Slave2:

IP: 192.168.0.102 Target server: 192.168.0.100

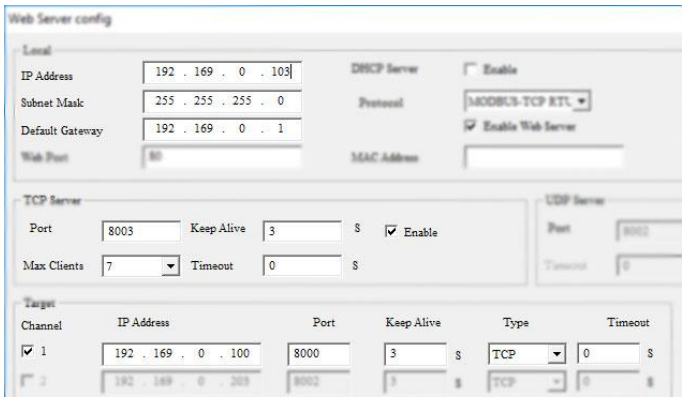
Target Port: 8000



Slave3:

IP: 192.168.0.103 Target server: 192.168.0.100

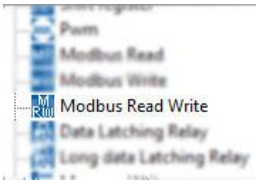
Target Port: 8000



Step 2 Programming the Master and Slave

Master program

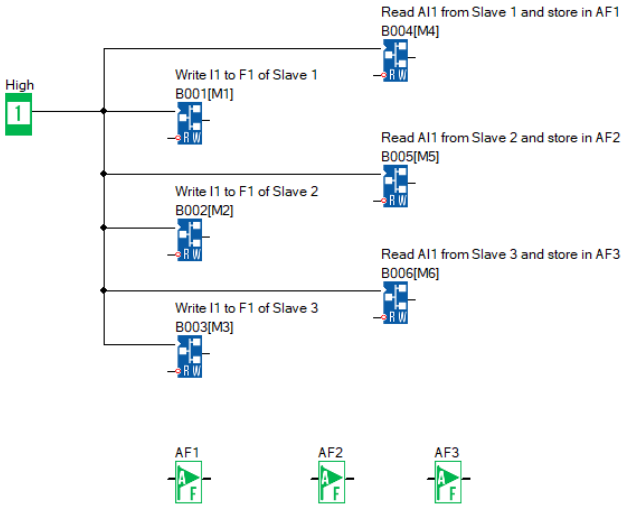
In the master program, you need read and write data to the slaves, the function block is "Modbus Read Write".

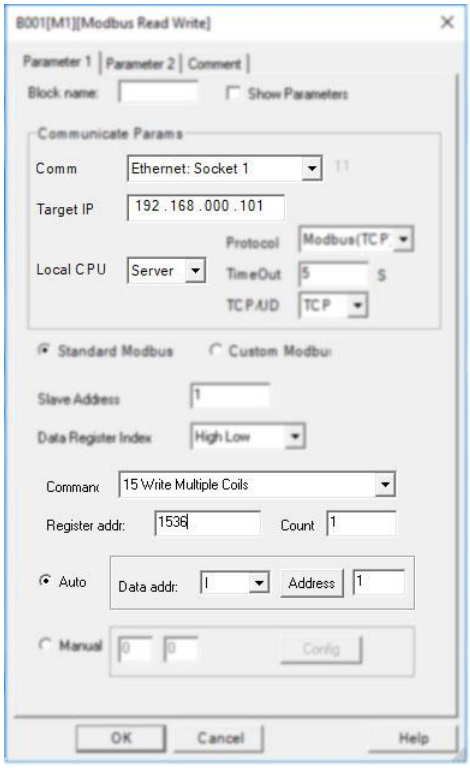


NOTICE:

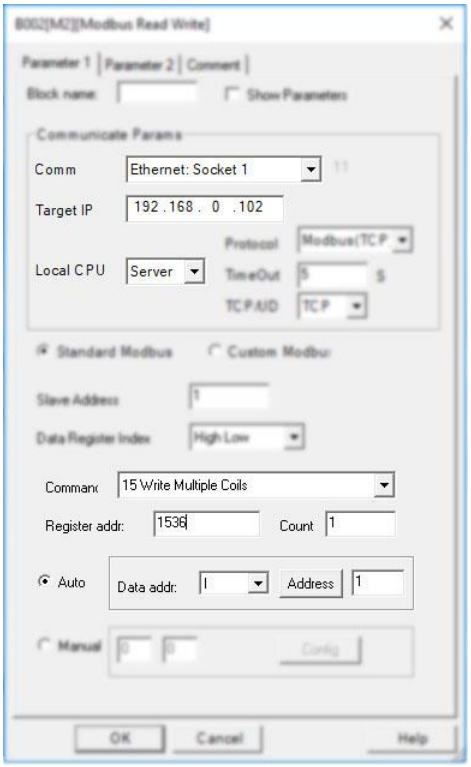
Even if the MODBUS READ and MODBUS WRITE block also have the Ethernet interface, but they only can

be used for the old version Ethernet CPU, it cannot be applied to the new one built-in web server.

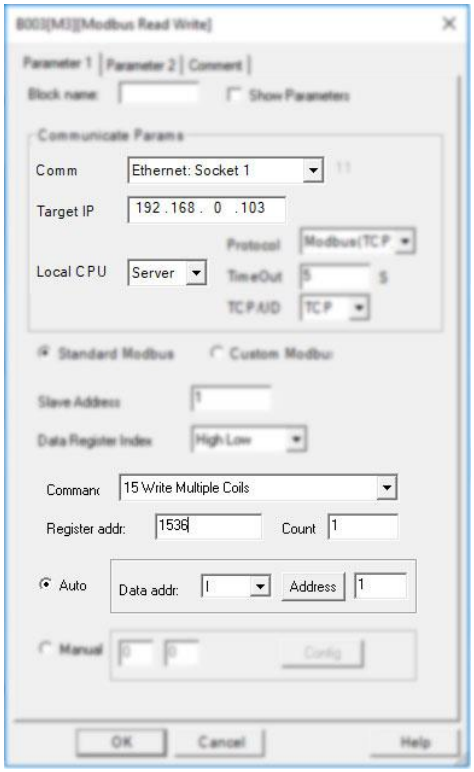




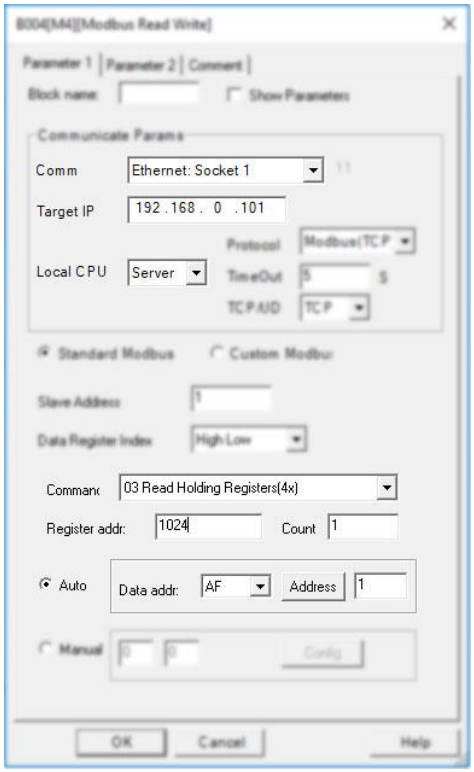
If B001 is enabled, the I1 status of master will transferred to the F1 (Modbus address is 0x1536) slave1 (IP:192.168.0.101).



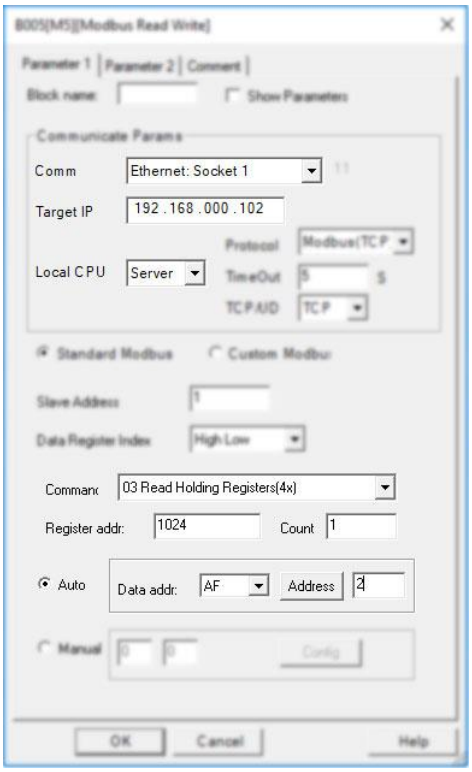
If B002 is enabled, the I1 status of master will transferred to the F1 (Modbus address is 0x1536) slave2 (IP:192.168.0.102).



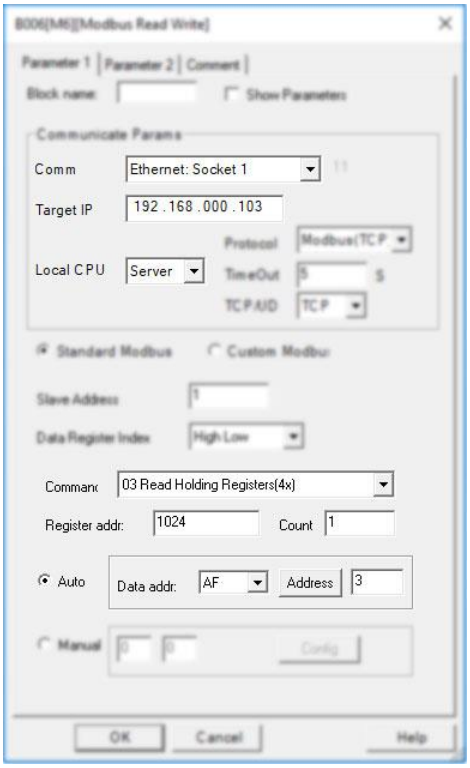
If B003 is enabled, the I1 status of master will transferred to the F1(Modbus address is 0x1536) slave3 (IP:192.168.0.103).



If B004 is enabled, the master will read the AI1 value (Modbus address 4x1024) of slave1 and save into local AF1.

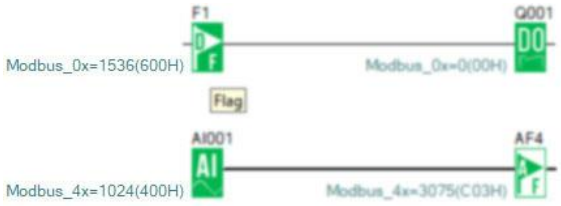


If B005 is enabled, the master will read the AI1 value (Modbus address 4x1024) of slave2 and save into local AF2.



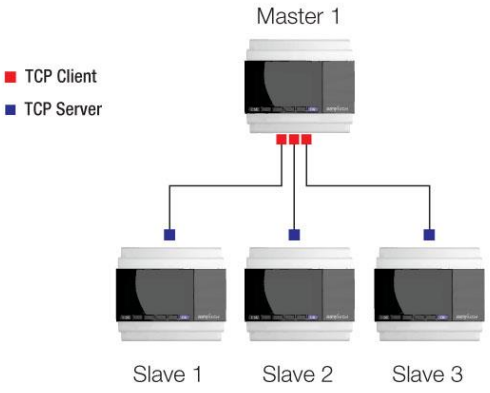
If B006 is enabled, the master will read the AI1 value (Modbus address 4x1024) of slave3 and save into local AF3.

Slave1--Slave3 Program is the same.



Example 2: One master CPU (TCP Client) connect with 3 slaves CPUs (TCP Servers)

Connection Diagram:



Requirement:

1.If I1 of Master is ON/OFF, the Q1 of slave1---slave3 are ON/OFF.

2.Read the AI2 value from the slave1--slave3 to master and display.

Step1: Configure the IP configuration of the PLCs.

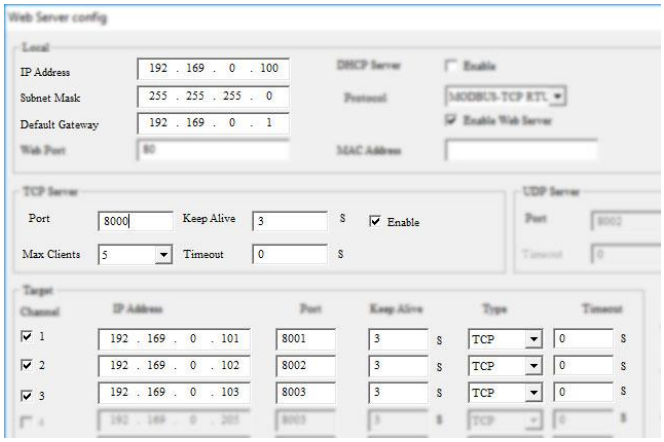
Master

IP: 192.168.0.100

Target server 1 IP :192.168.0.101 port 8001

Target server 2 IP :192.168.0.102 port 8002

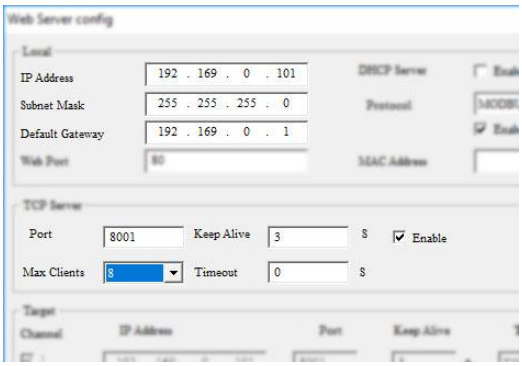
Target server 3 IP :192.168.0.103 port 8003



Slave1:

IP:192.168.0.101

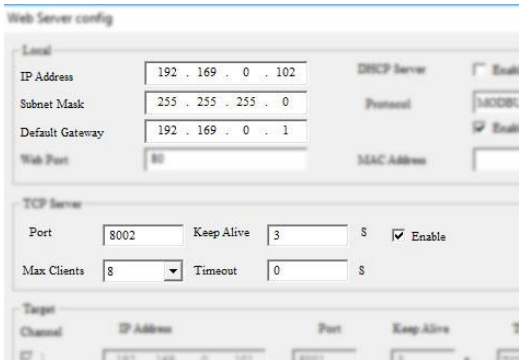
TCP server port:8001



Slave2:

IP: 192.168.0.102

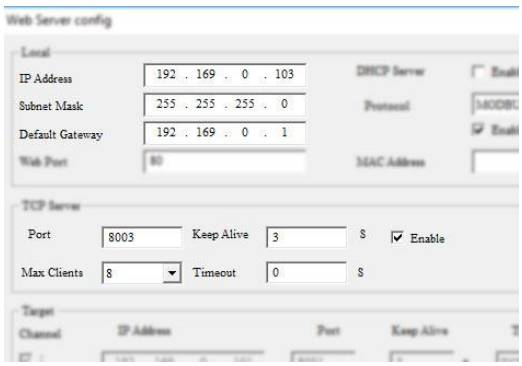
TCP server port: 8002



Slave3:

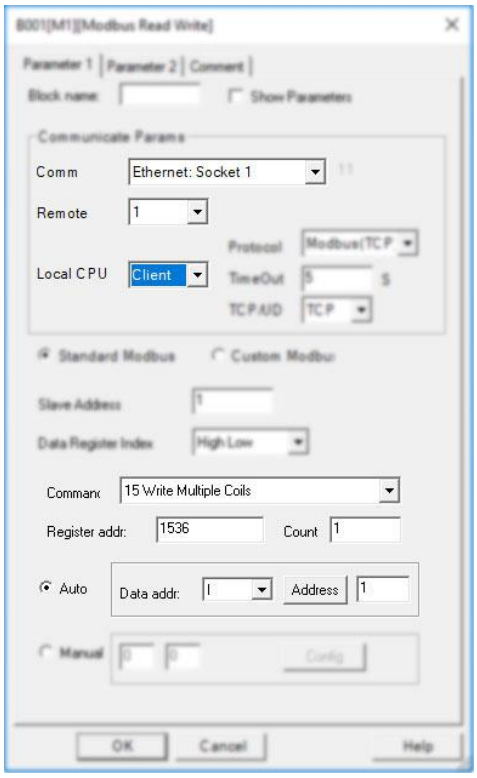
IP: 192.168.0.103

TCP server port: 8003

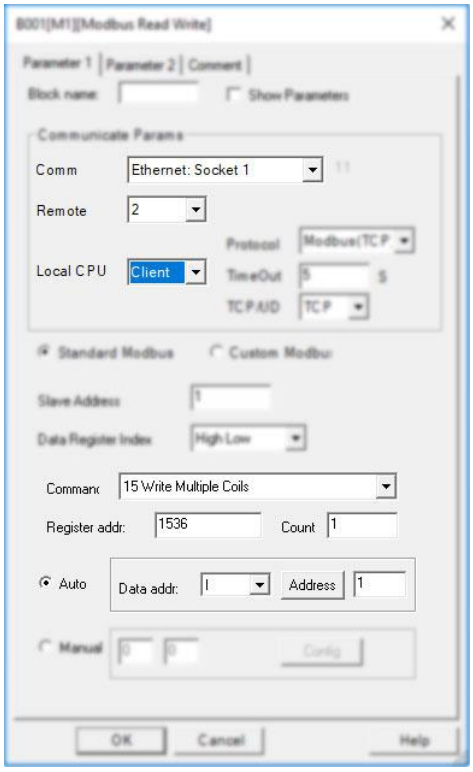


Step 2 Program in the Master and Slave

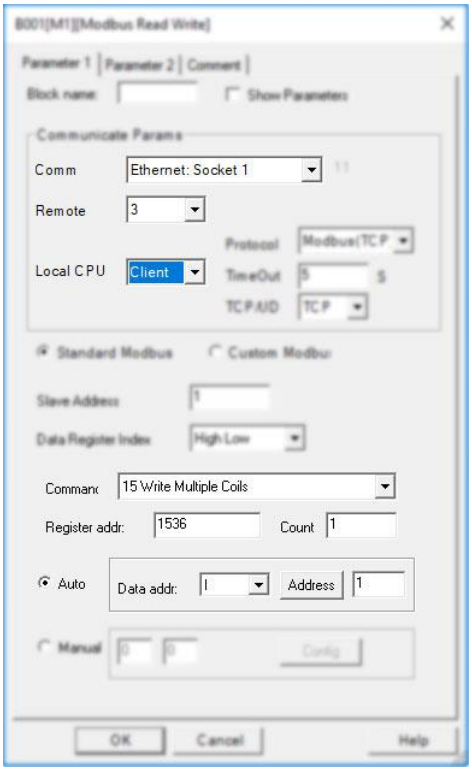
Master program



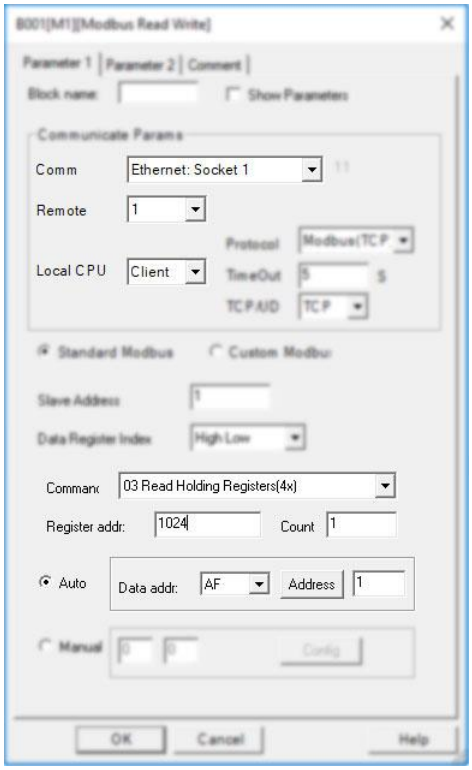
B001 is used to write I1 status to F1 of slave1, the Remote 1 is the target1 in the network parameter settings.



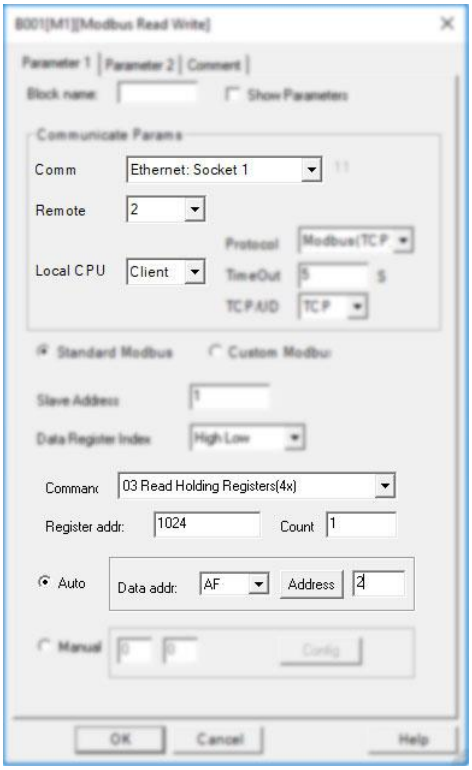
B002 is used to write I1 status to F1 of slave1, the Remote 2 is the target2 in the network parameter settings.



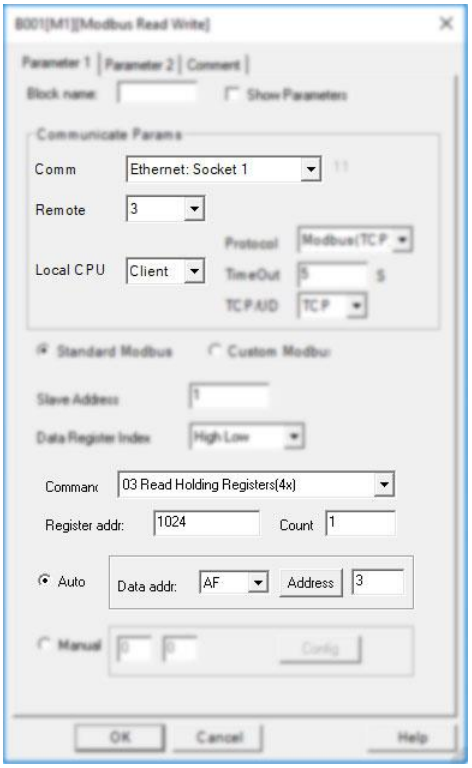
B003 is used to write I1 status to F1 of slave1, the Remote 3 is the target3 in the network parameter settings.



B004 is used to read AI1 value of slave1 then save into AF1, the Remote 1 is the target1 in the network parameter settings.

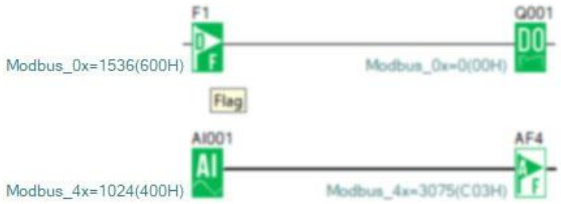


B005 is used to read AI1 value of slave2 then save into AF2, the Remote 2 is the target2 in the network parameter settings.



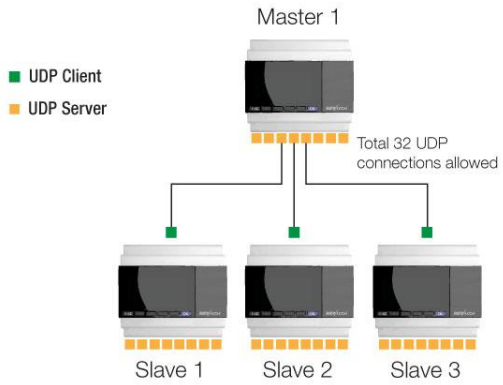
B006 is used to read AI1 value of slave3 then save into AF3, the Remote 3 is the target3 in the network parameter settings.

Slave1--Slave3 Program is the same.



Example3: One Master CPU (UDP Server) connect with 3 Slave CPUs (UDP Clients)

Connection Diagram:



Requirement:

1.If I1 of Master is ON/OFF, the Q1 of slave1---slave3 are ON/OFF.

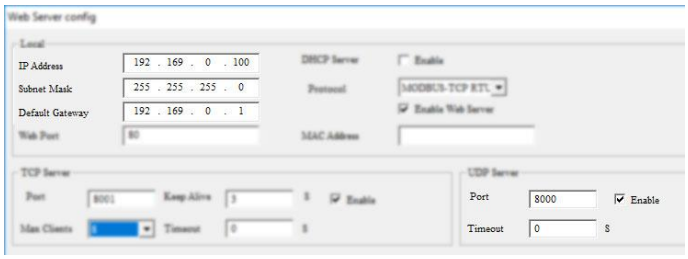
2.Read the AI2 value from the slave1--slave3 to master and display.

Step1: Configure the IP configuration of the PLCs.

Master

IP: 192.168.0.100

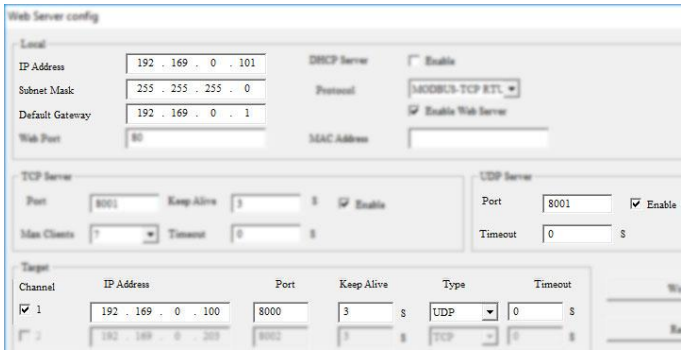
UDP Server port 8000



Slave1:

IP:192.168.0.101 UDP Port 8001

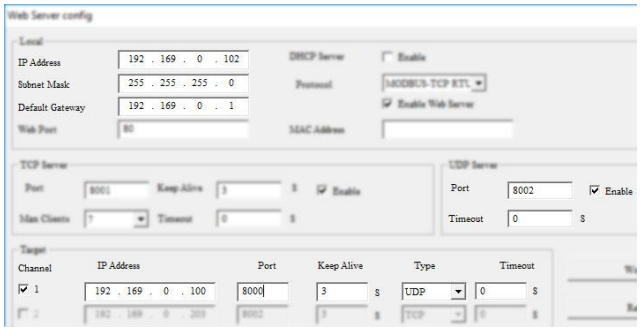
Target1 IP address:192.168.0.100. UDP port 8000



Slave2:

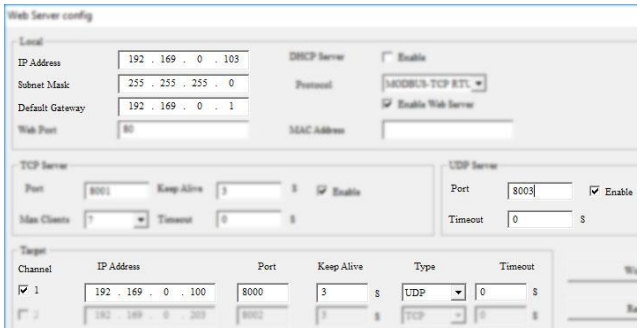
IP:192.168.0.102 UDP port: 8002

Target1: 192.168.0.100 UDP port:8000



Slave3:

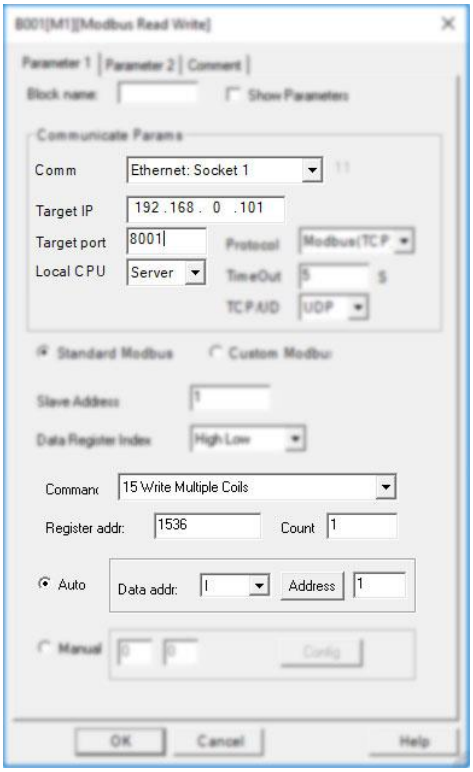
IP:192.168.0.103 UDP port: 8003



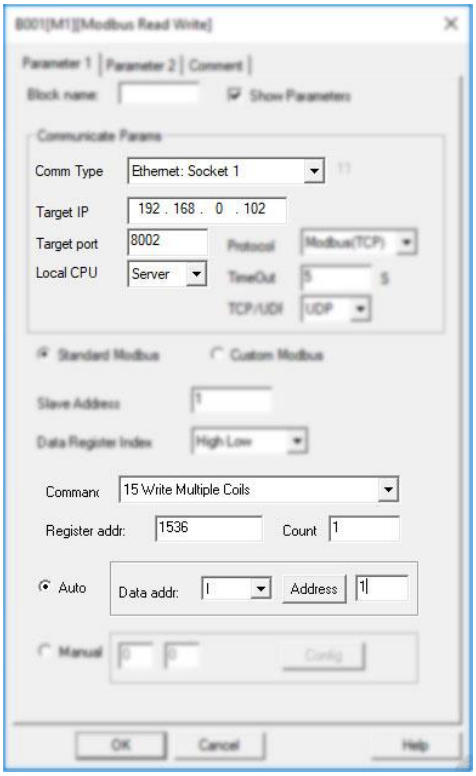
Target1: 192.168.0.100 UDP port:8000

Step 2 Program in master and slave

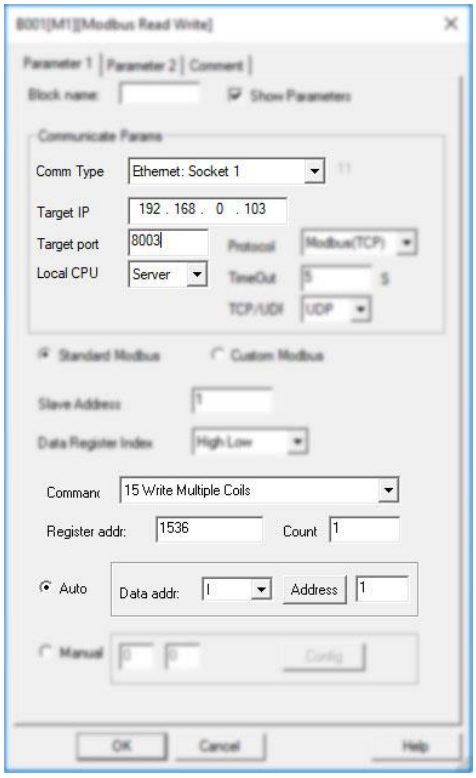
Master program



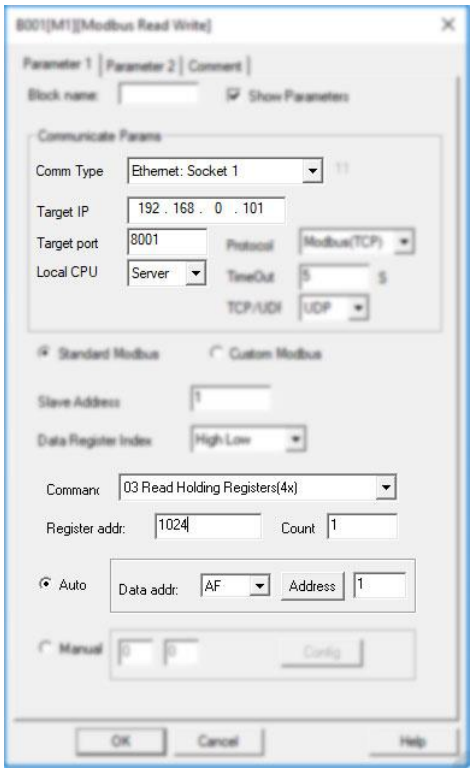
B001 is used to transfer the I1 status to the F1 of slave1 (slave1 IP 192.168.0.101, UDP port is 8001)



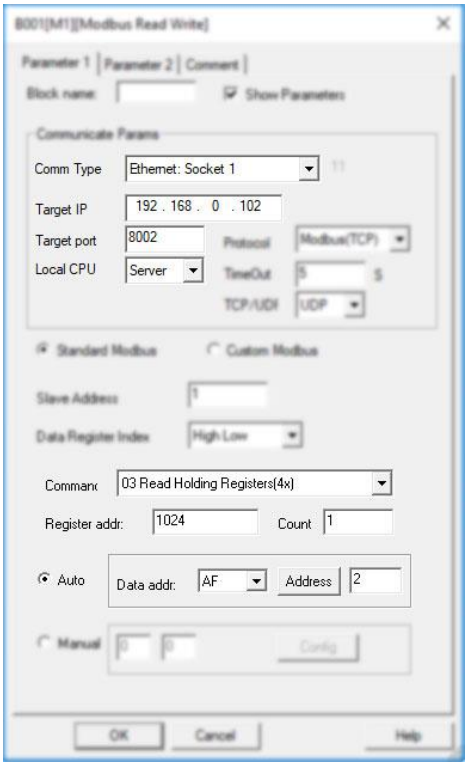
B002 is used to transfer the I1 status to the F1 of slave2 (slave2 IP 192.168.0.102, UDP port is 8002)



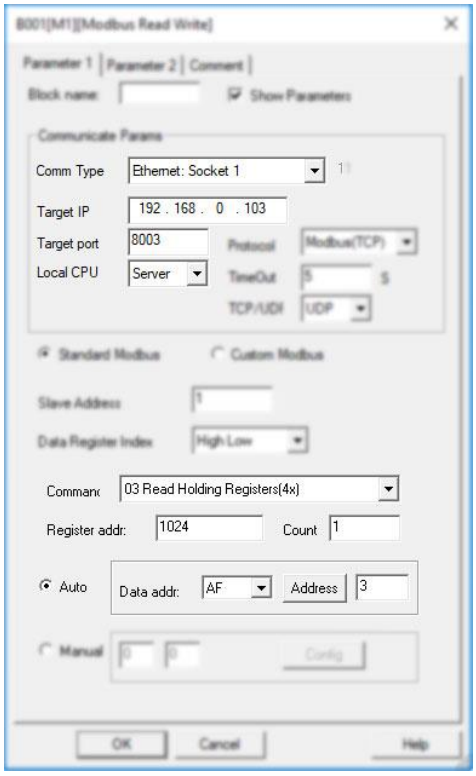
B003 is used to transfer the I1 status to the F1 of slave3 (slave3 IP 192.168.0.103, UDP port is 8003)



B004 is used to read the AI1 value from slave1, then save it into AF1 of master (slave1 IP 192.168.0.101, UDP port is 8001)

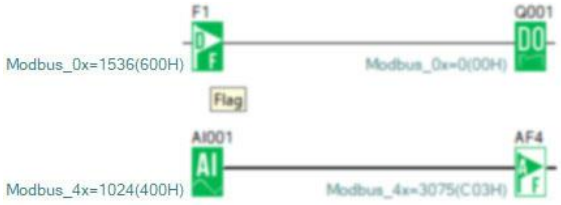


B005 is used to read the AI1 value from slave2, then save it into AF2 of master (slave1 IP 192.168.0.102, UDP port is 8002)

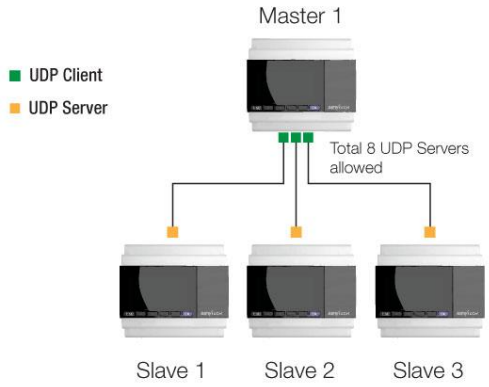


B006 is used to read the AI1 value from slave3, then save it into AF3 of master (slave1 IP 192.168.0.103, UDP port is 8003)

Slave1--Slave3 Program are the same.



Example 4: One Master CPU (TCP Client) Connected with 3 Slave CPUs (UDP Servers)



Requirement:

1.If I1 of Master is ON/OFF, the Q1 of slave1---slave3 are ON/OFF.

2.Read the AI2 value from the slave1--slave3 to master and display.

Step1: Configure the IP configuration of the PLCs.

Master

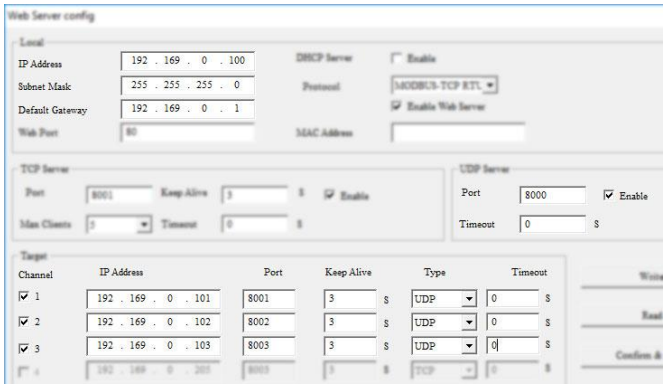
IP: 192.168.0.100

UDP Server port 8000

Target1 UDP 192.168.0.101, Port :8001

Target2 UDP 192.168.0.102, Port :8002

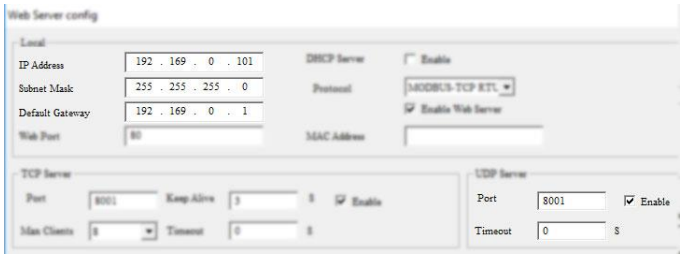
Target3 UDP 192.168.0.103, Port :8003



Slave1

IP: 192.168.0.101

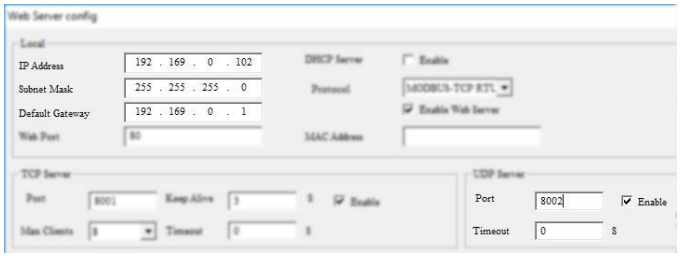
UDP Server port 8001



Slave2

IP: 192.168.0.102

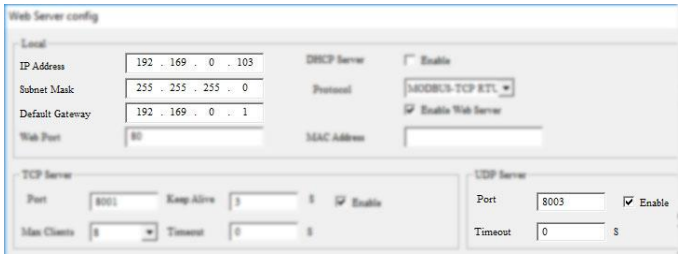
UDP Server port 8002



Slave3

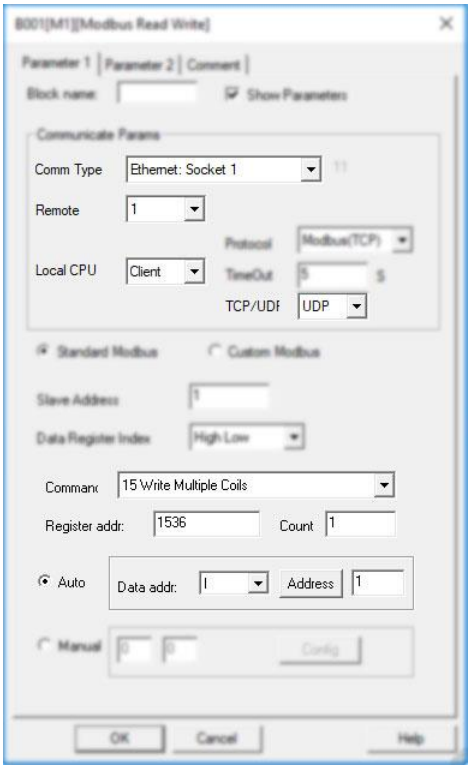
IP: 192.168.0.103

UDP Server port 8003



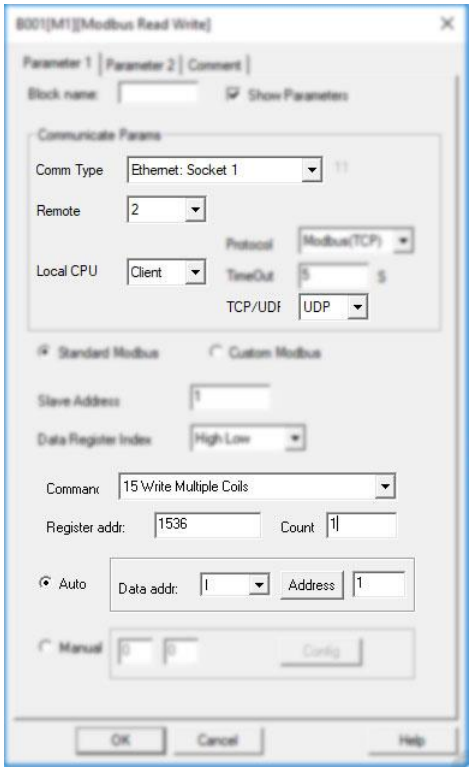
Step 2 Program in master and slave

Master program



B001 is used to transfer I1 status to the F1 of slave1 by UDP connection.

The Remote1 is the target UDP server setting in the network parameter:

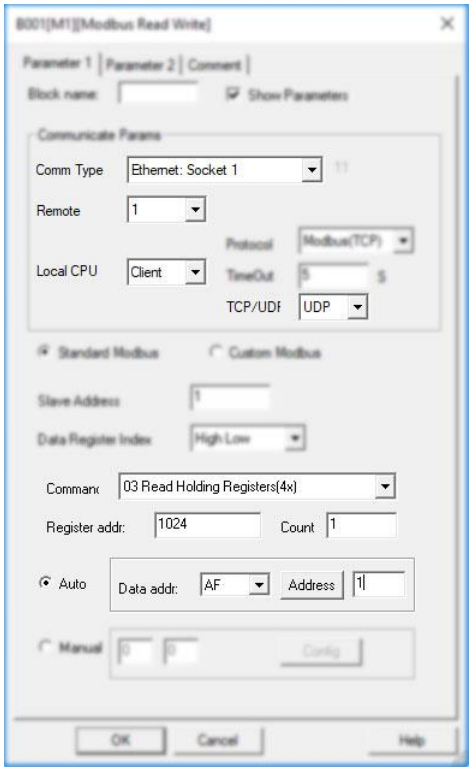


B002 is used to transfer I1 status to the F1 of slave2 by UDP connection.

The Remote2 is the target UDP server setting in the network parameter:

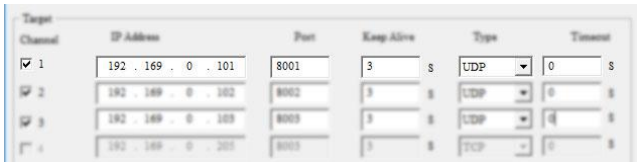
B003 is used to transfer I1 status to the F1 of slave3 by UDP connection.

The Remote3 is the target UDP server setting in the network parameter:



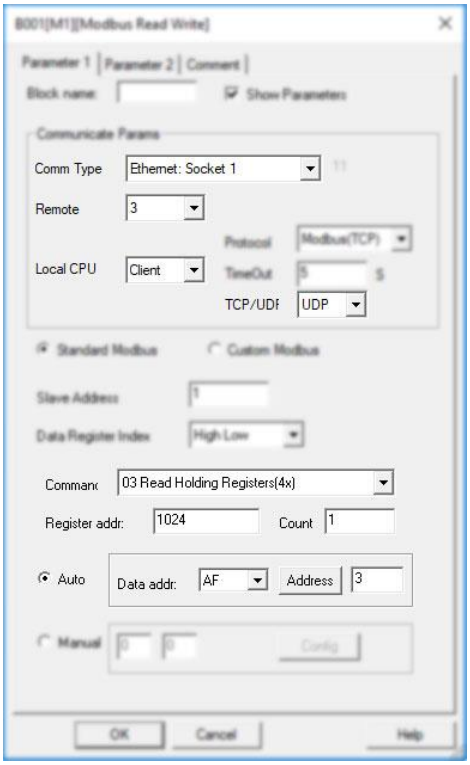
B004 is used to read AI1 value from slave1 and save the value into AF1 by UDP connection.

The Remote1 is the target UDP server setting in the network parameter:



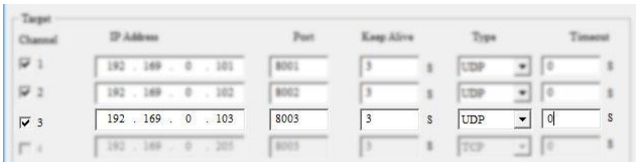
B005 is used to read AI1 value from slave2 and save the value into AF2 by UDP connection.

The Remote2 is the target UDP server setting in the network parameter:

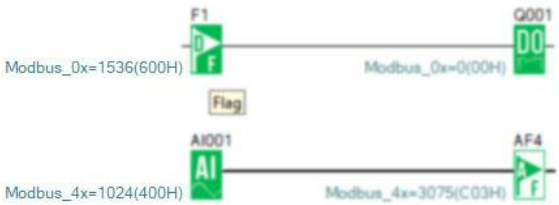


B006 is used to read AI1 value from slave3 and save the value into AF3 by UDP connection.

The Remote3 is the target UDP server setting in the network parameter:



Slave1--Slave3 Program would be same.



8.1. Modification of User Web Pages

The web pages on the following models can be edited and customized to suit your own needs and applications. The current web pages for each model are available for download and modification.

- ELC-12DC-DA-R-N (Version 2)
- PR-12DC-...-N
- PR-18DC-...-N
- PR-26DC-...-N
- PR-26DC-...-G
- PR-26DC-...-WIFI

For each of the Rievtech Ethernet CPUs which have a built-in webserver, there is a folder which contains all the files stored within the CPU.

The folder name is in the format: "**web - PLC Model Name_FBD**". The folder name ***MUST*** remain the same.

Each folder contains the web files used by the CPUs webserver. The folder contains 26-files in total, which are of the following type:

- 6 x HTML web pages (User editable)
- 1 x CSS Style Sheet
- 2 x Java Script Files
- 17 x PNG Graphics Files

The webserver uses xmlhttp to deal with interactions with web pages and therefore your browser must support these types of calls to work properly. The majority of modern web browsers support this method of interaction.

You are able to modify any file within a project folder but it is advised that the Stylesheet and the Javascript files are not modified, as incorrect coding could cause your CPU to stop working and void its warranty.

Modifying the graphics and HTML page files are allowed but the following restrictions must be born in mind when doing so.

NOTICE:

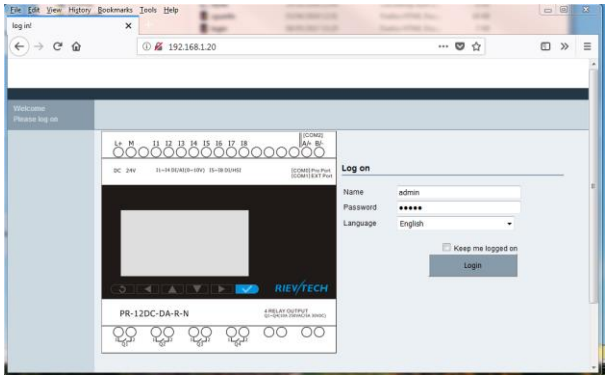
- The image used can be modified/changed but its name MUST remain the same.
- The name of each web page MUST remain the same.
- The name of the project folder MUST remain the same.
- The size of the project build CANNOT exceed 192kb.
- The size of a single file with the project folder CANNOT exceed 100kb,
- The number of web pages in the project folder MUST remain as 6 in total.
- The file format of the web pages MUST remain the same.

Editing Web Pages

As long as the rules above are observed, then you may use any HTML editor to modify the six web pages within a CPUs project folder. The following information describes each HTML page and its properties.

Page 1 – login.html

This is the first web page that is displayed when you enter the IP address of the CPU into a web browser.



The default login details are:

- Username: admin
- Password: admin

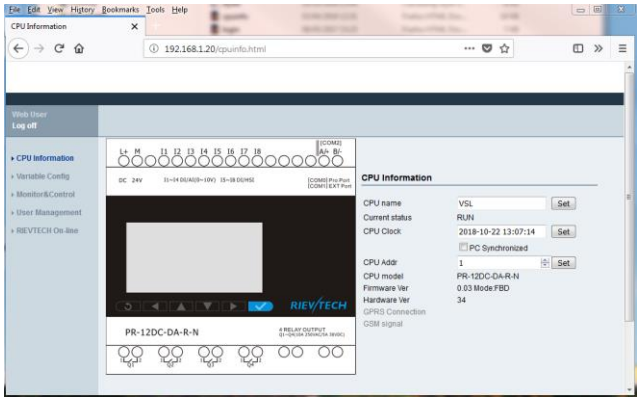
You can select from a choice of five languages, which are:

- English
- German
- French
- Spanish
- Chinese

Checking the 'Keep me logged in' box, you will not be automatically logged out after the inactivity timer expires.

Clicking the 'Login' button will take you to the next page, 'cpuinfo.html'.

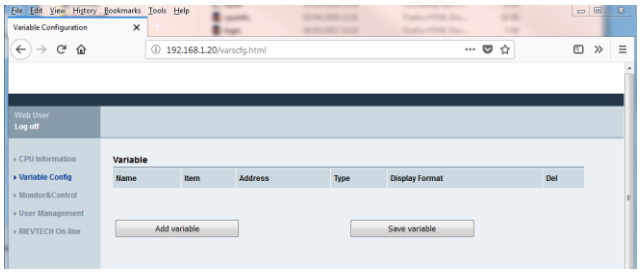
Page 2 – cpuinfo.html

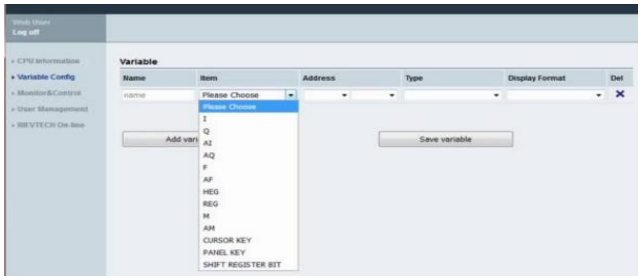


This web page shows the following information by default:

- CPU name – Changeable here by typing new name and pressing 'Set' button
- Current status – Read only
- CPU Clock – Changeable here either by typing of ticking the 'PC Synchronized' box and pressing 'Set' button
- CPU Addr – Changeable here by typing a new number and pressing 'Set' button
- CPU model – Read only
- Firmware Ver – Read only
- Hardware Ver – Read only
- GPRS Connection – Read only on supporting models
- GSM signal – Read only on supporting models

Page 3 – varscfg.html





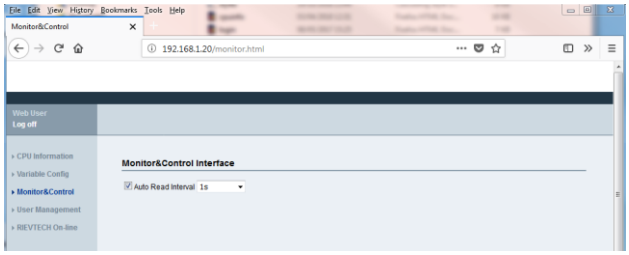
This web page allows the user to add and delete variables they wish to view. By default no variables are set and the user must add them. The list of variables that may be added are:

- I – Digital Input
- Q – Digital Output
- AI – Analogue Input
- AQ – Analogue Output
- F – Digital Flag
- AF – Analogue Flag
- HEG – Frequency Value of a Threshold Trigger
- REG – Holding Register
- M – Middle Coil Value of Logic Block
- AM – Analogue Quantity Buffer
- Cursor Key – CPU Cursor keys e.g. Arrow keys
- Panel Key – CPU Panel keys e.g. OK, ESC, etc.
- Shift Register Bit – Shift Registers – Read Only

A new variable is added by clicking on the 'Add Variable' button and selecting the type of variable required from the 'Item' column dropdown.

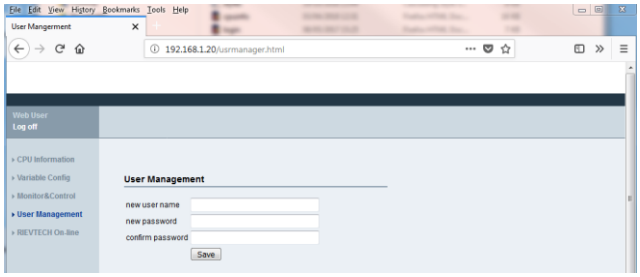
Page 4 – monitor.html

This web page allows the user to change the rate at which data for the browser is refreshed.



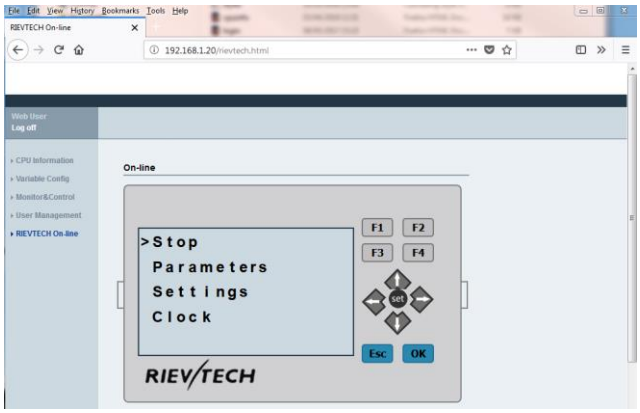
Page 5 – usermanager.html

This web page allows for the addition of new users. Please note that ALL users added will have access to ALL settings on these web pages.



Page 6 – Rievtech.html

This web page shows the user data as if they were via the CPU LCD directly. All buttons function via the web page interface as they would when using the keys on the front of the PLC itself.



You can change any data using the web page that can be changed via the LCD and physical buttons on the screen.

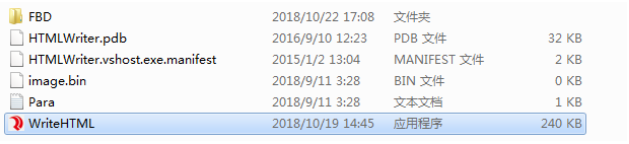
NOTICE:

Great care must be used when accessing remotely, as incorrect changes could disable a connection completely.

Use your favourite HTML editor software to change the web pages to your liking.

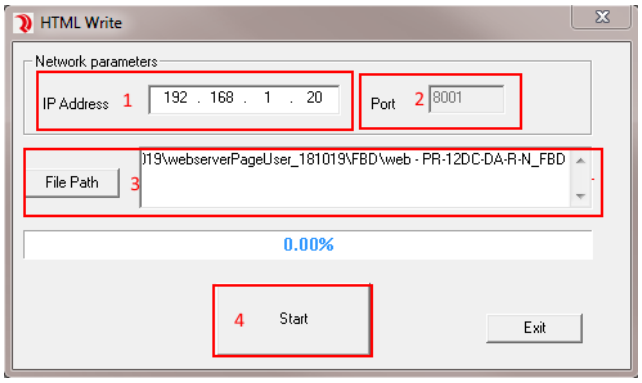
8.2. Downloading Web Pages to CPU

Once you have finished your modifications to the HTML files, you can download it to your CPU using our HTML Writer software, which you will find in our Web Server Page Update Package downloadable from our web site.

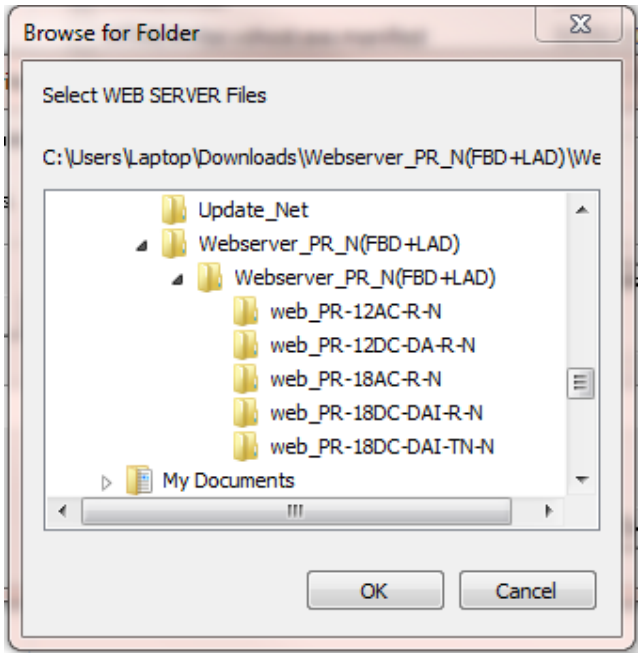


NOTICE:

This software uses port 8001 to write your project, so please ensure the Server Port of your CPU is configured to 8001.



1. Enter the IP address of your CPU
2. Server Port number is 8001
3. Click 'File Path' button to browse for the modified project file directory and select your CPU model



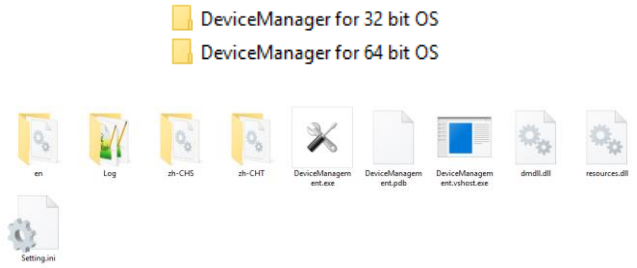
4. Click 'Start' button to begin the download procedure

Once the download has been completed successfully, click the 'OK' button and the CPU will reboot and apply the new web pages.

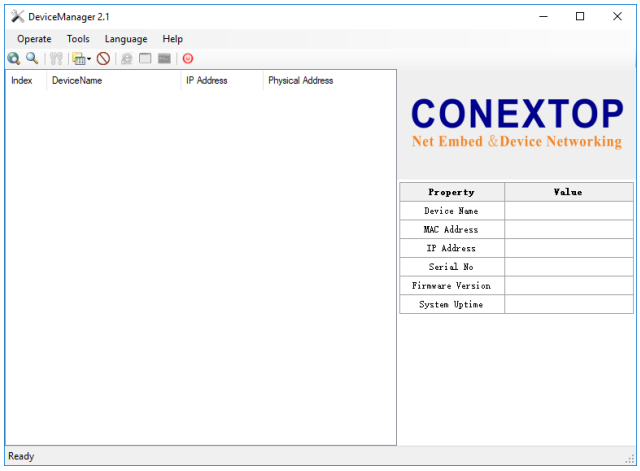
Enter the IP address of the CPU in your web browser to view the new pages.

8.3. Configuration with Ethernet Device Manager

Most people prefer to change the settings of the Ethernet xLogic CPU rather than their PC settings. To do this download our Device Manager for Ethernet CPU software from the website. Unpack the software to a location on your hard drive and open the location using Windows Explorer. There are two versions of the software, a 32-bit and a 64-bit. Select the version which matches your Windows OS system.

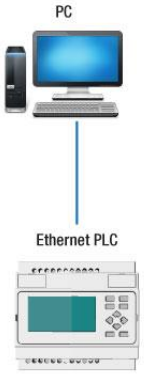


a: Select "DeviceManagement.exe" file and start it by double-clicking with the left mouse button.



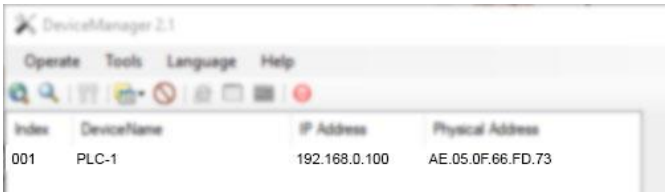
b: Connect your xLogic Ethernet CPU to your network, either via a router/switch or directly to your PC. In our following example, we will use a direct connection to our PC.

Connect diagram:

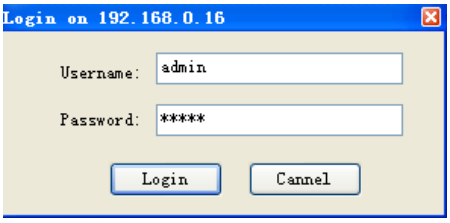


c: Apply power to the xLogic Ethernet CPU and follow the instruction below carefully.

d: Click in the Device Manager software to search for the xLogic Ethernet CPU. Once the search is complete the software will list all devices found with their IP address and MAC address.



e. Double-click the device in the list or highlight it and click the toolbar button.



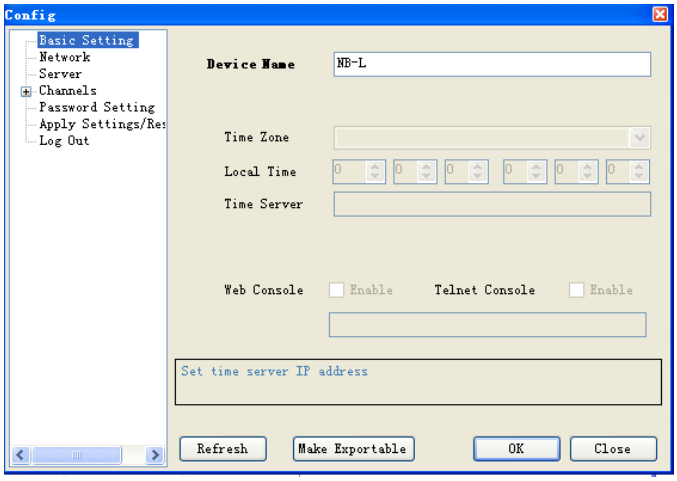
f. Input the username and password to log in. The default is Username: admin; Password is admin. Click the Login button to continue and the configuration dialogue box will appear.

NOTICE:

If you make any changes to any of the following dialogue boxes, you MUST press the OK button in each section to save your settings to the CPU.

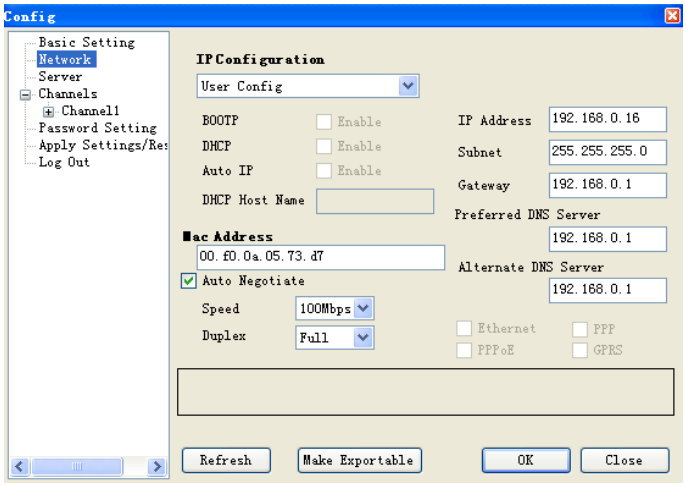
Basic Settings

Here you can enter a name for the device. The default is NB-L.



After you click the "OK" button, the settings in the current dialogue box will be set into the CPU.

Network Settings



In this section, you can configure the Ethernet network settings of the xLogic Ethernet CPU. You can change the following settings:

- IP Configuration: DHCP or User Config.
- IP Address: Default is 192.168.0.250
- Subnet: Default is 255.255.255.0
- Gateway: Default is 192.168.0.1
- Preferred DNS Server
- Alternate (Secondary) DNS Server

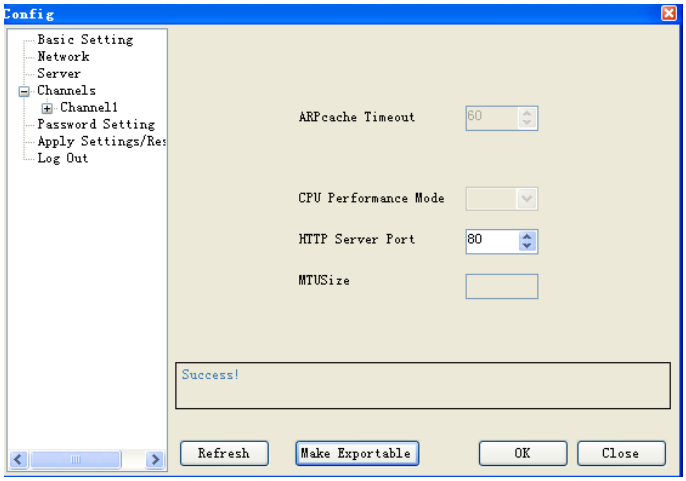
NOTICE:

Please leave **ALL** other settings as Default unless you are fully familiar with Ethernet network configurations.

Ensure any changes made are written to the xLogic CPU by pressing the OK button.

Server Settings

The default settings in this section should be sufficient for most installation and should be left unchanged where possible.



Channels

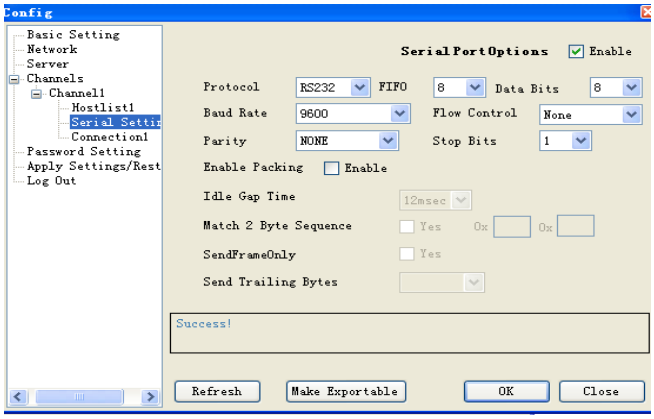
The channels configuration section is used to configure communication speeds and termination settings for the CPU serial ports and the network settings for Server/Client modes of the xLogic CPU.

NOTICE:

Unless there is a specific need to, please do not change any of the settings in the Serial Settings section as this may cause communication problems for your xLogic CPU.

Ensure any changes made are written to the xLogic CPU by pressing the OK button.

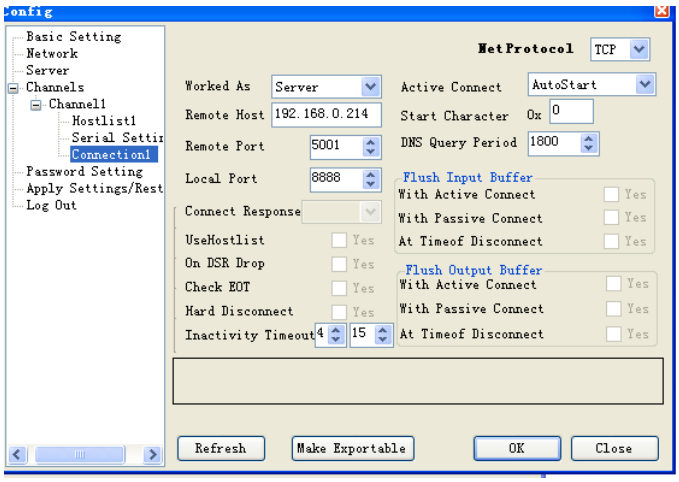
Serial Settings



The above is the default serial settings for the xLogic CPU and should be left as shown above.

Connection Settings

The xLogic Ethernet CPU can work either as a Client or as a Server but ***NOT BOTH***.



The following fields can be adjusted here:

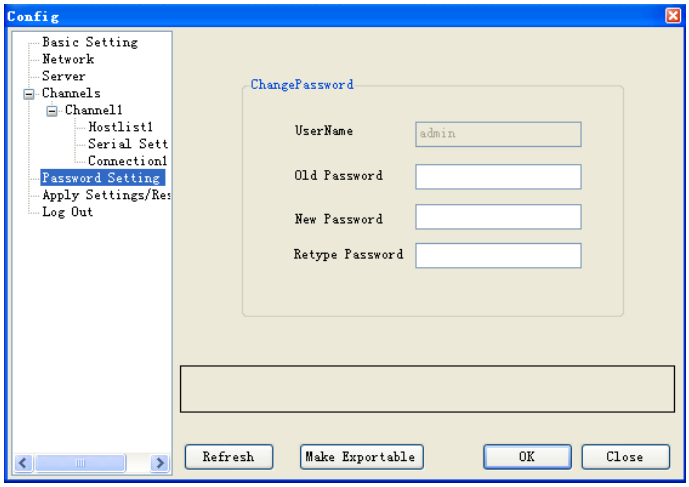
- Net Protocol – Default is TCP and should NOT be changed.
- Worked As – Choose either Server or Client depending on whether the xLogic CPU will be communicating as a Slave (Server) or Master (Client) in the Ethernet network– Default is Server.
- Remote Host: The remote PC IP address. When the xLogic CPU is configured to be a Client then it will automatically try to connect to the IP address entered here.
- Active Connect – Default is Auto Start and should be left unchanged
- Remote Port – When the xLogic CPU is configured to be a Client then it will automatically try to connect on the port number set here.
- Local Port – When the xLogic CPU is configured to be a Server then it will only accept communication on the port number set here. For example, if the xLogic CPU was being used in a Modbus/TCP network as a Server and the Modbus Client used a port setting of 502 (Standard Ethernet Modbus Port number), then this field must be set to 502.

NOTICE:

All other fields should be left with their default values unless you are fully familiar with their settings.

Ensure any changes made are written to the xLogic CPU by pressing the OK button.

Password Setting

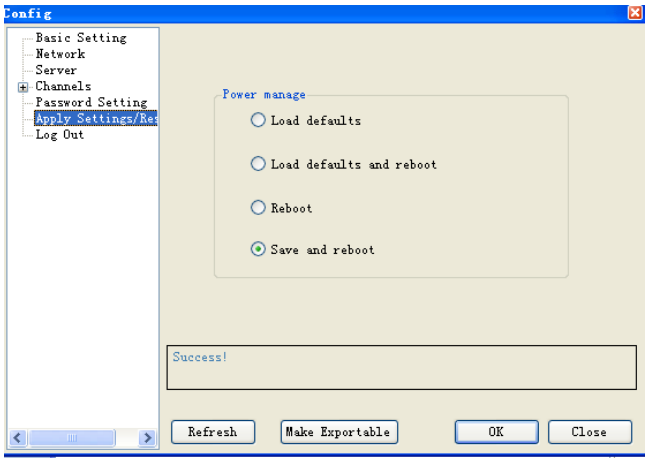


A password must be provided to communicate via the xLogic CPU Ethernet port and can be changed here. The default password is admin. The password must contain at least one character.

To change the Ethernet password, enter the Old Password, enter a New Password and enter the New Password again to confirm, then click the OK button.

Ensure any changes made are written to the xLogic CPU by pressing the OK button.

Apply Settings/Reset



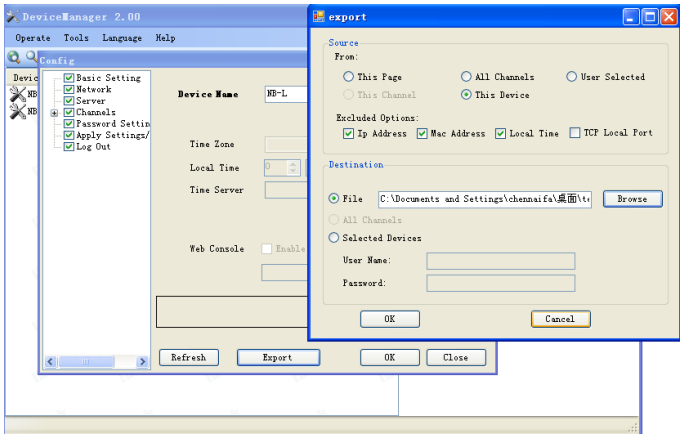
To enable any changes to be permanently saved to the xLogic Ethernet CPU you must select the Save and reboot option and press the OK button.

NOTICE:

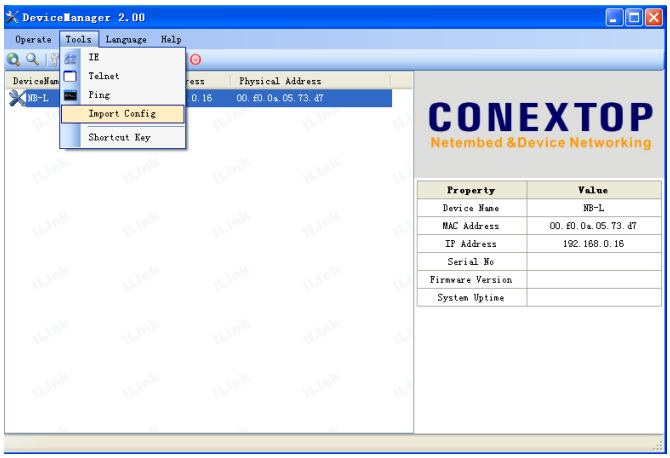
If this step is NOT carried out then the xLogic Ethernet CPU will lose any settings changes when power is cycled to the unit.

Export Your Settings

Click Make Exportable button to export your settings and store them in a .xml file.



You also can import existing configurations from the menu Tools->Import Config



8.4. Establish communication between CPU and xLogicsoft/SCADA via Ethernet.

The Ethernet port of your xLogic Ethernet CPU can be used to communicate with peripheral devices such as a Touchscreen/HMI, PC SCADA using the Modbus/TCP protocol.

This chapter will introduce how to establish the connection between CPU and xLogicsoft via Ethernet using a pre-configured xLogic Ethernet CPU configured as per the previous section.

In xLogicsoft open "COM PORT" by either clicking the icon or using the menu Tools -> Configuration option.

Ethernet CPU as Client and xLogicsoft as Server

a. Select "Ethernet" option

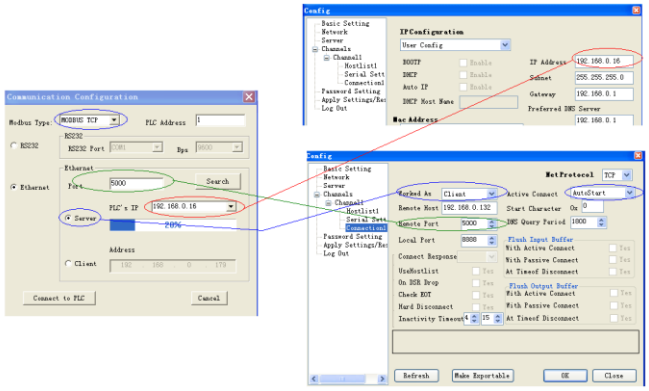
b. Click "Search" button to search for CPU

c. Select found CPU from the drop-down list

d. Click "Connect to PLC" button

NOTICE:

Both xLogicsoft and CPUs ports MUST have the same setting.



xLogicsoft Com port setting dialogue box LAN port settings in Device manager

NOTICE:

1. The remote host IP address is the same as that of the PC in which the xLogicsoft being installed.

2. Modbus Type in xLogicsoft must be MODBUS TCP.

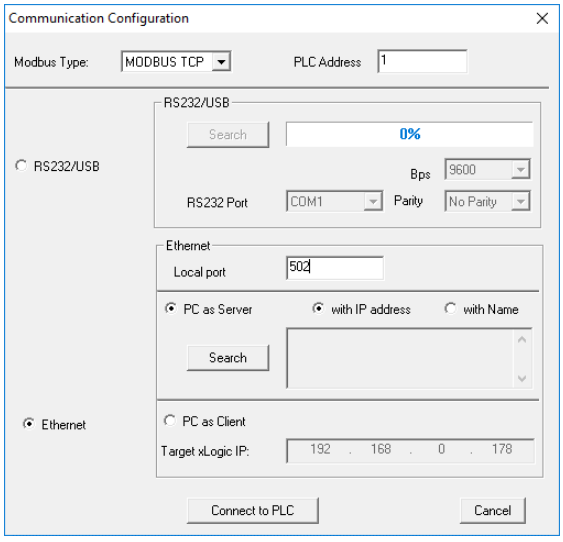
3. Active Connect must be "Auto Start" in device manager.

After the xLogic CPU and PC are connected, you will be able to use the many features of xLogicsoft, e.g. download user program into xLogic CPU module, uploading the program into PC and online monitor (monitor real-time status of xLogic IO) can be done. The only difference is that you are connected via Ethernet rather than a serial connection using the ELC-RS232 or USB cables.

A. Upload program: click

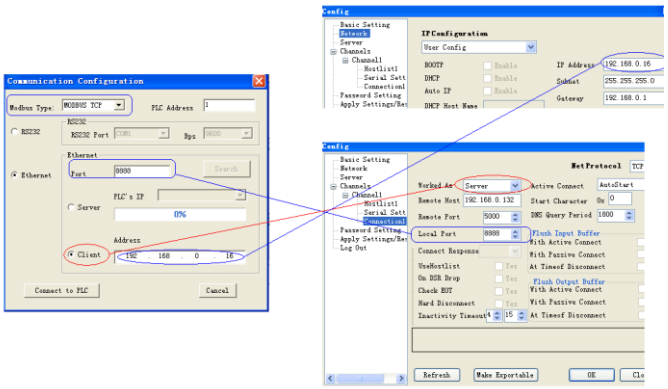
B. Download program: click

C. Monitor program run status: click



Ethernet CPU as Server and xLogicsoft as Client

The connection is established as before except that the option Client is selected in xLogicsoft connection window.

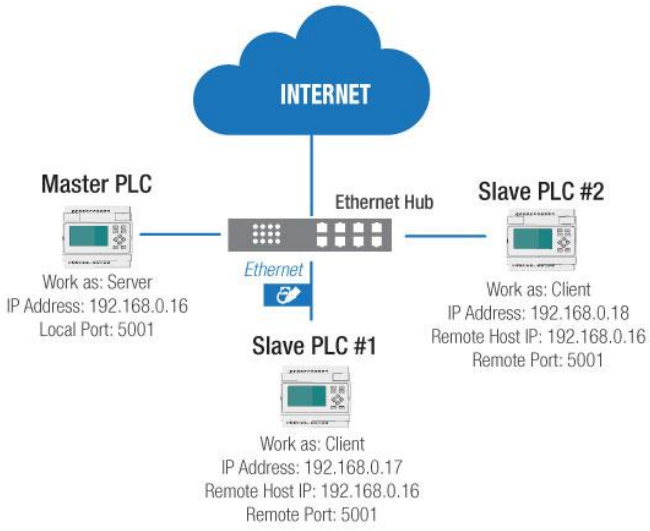


NOTICE:

The configuration of the xLogic Ethernet CPU MUST be configured as above for this type of connection to work.

8.5. How to Establish Communication Between CPUs via Ethernet?

Hardware Connection

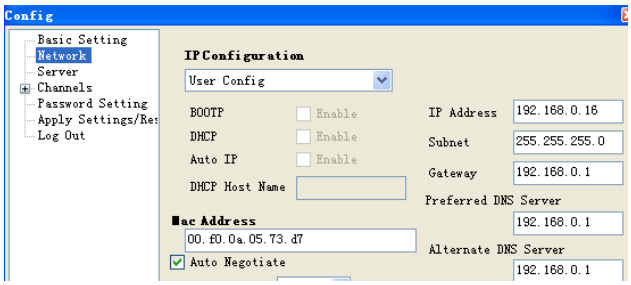


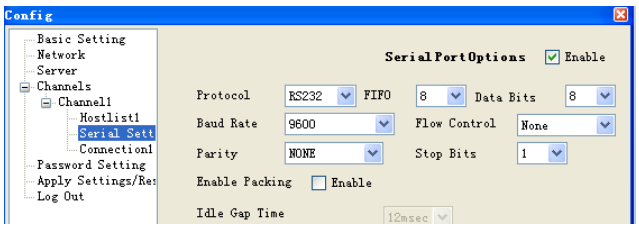
First, you must configure each xLogic Ethernet CPU with its own unique settings using the Device Manager software.

Example:

Master PLC works as Server. Its IP address is 192.168.0.16 and the local port shall be set as 5001.

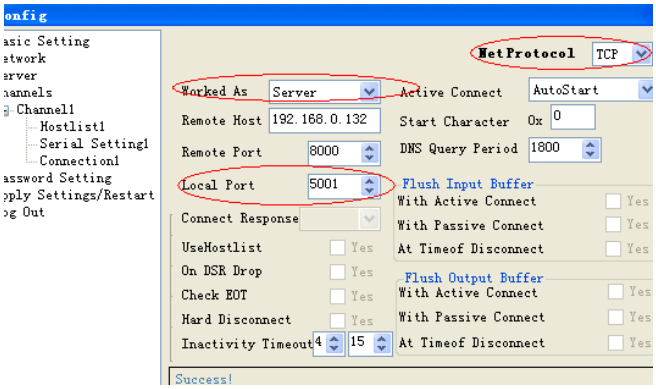
As this configuration shows:





Work as: Server

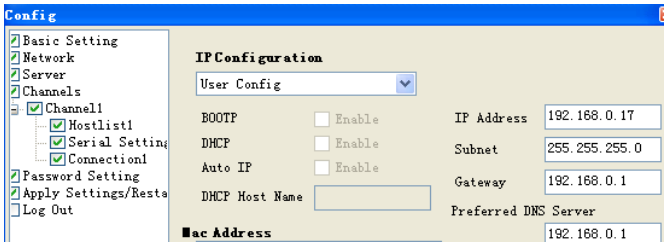
Local port: 5001



The items with a red circle marked must be the same as above.

Slave 1 setting:

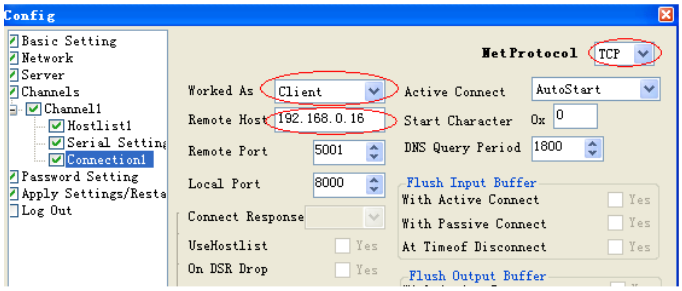
The IP address is 192.168.0.17



Work as: Client

Remote Host: 192.168.0.16

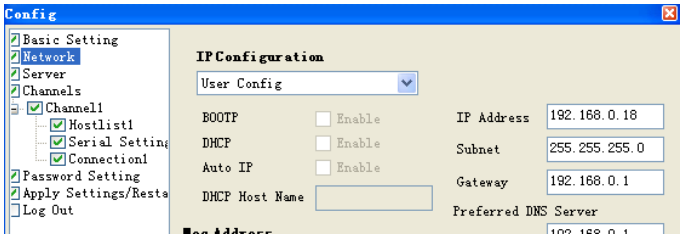
Remote port: 5001



The items with a red circle marked must be the same as above.

Slave 2 settings:

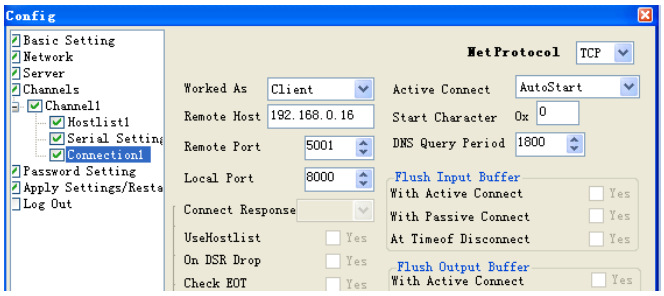
The IP address is 192.168.0.18



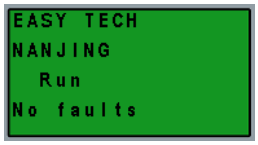
Work as: Client

Remote Host: 192.168.0.16

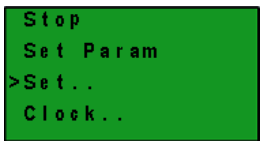
Remote port: 5001



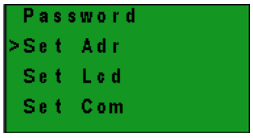
PLC address needs to be changed to 2 (default is 1). Change the CPU address via the HMI on the CPU as follows:



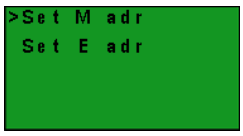
Press Esc, and then press



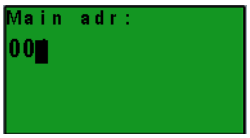
Press Ok



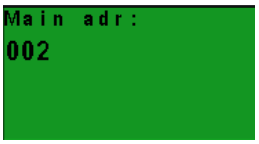
Press Ok



Press Ok



Change address with UP or DOWN button and confirm with OK.



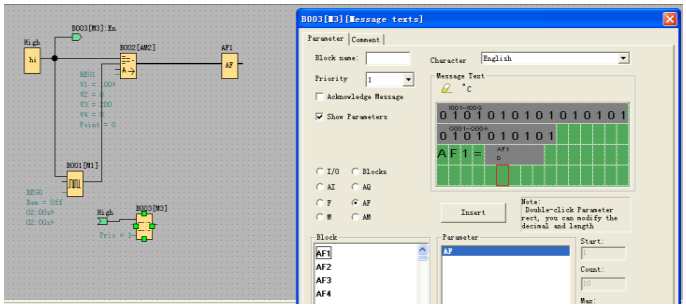
Program Logic Required

1. I1--IA Digital Inputs in the Master is to control the Digital Outputs, Q1—QA, in both Slave1 and Slave2. If I1 is ON in the Master, the corresponding Q1 in the Salves is to be ON; I2 is ON in the Master, the corresponding Q2 in the Slaves is to be ON and so on up to IA/QA. If an input is Off in the Master, the corresponding output in both Slaves must also be Off.

2. Read the Inputs I1--IA statuses of Slave 1 and reflect their states in the Flags F11--F20 in the Master. Read the Inputs I1--IA statuses of Slave 2 and reflect their states in the Flags F21--F30 in the Master.
3. Read the AF1 flag of Slave1 and save it in the AF1 flag of the Master. Read the AF1 flag of Slave2 and save it in the AF2 flag of the Master.

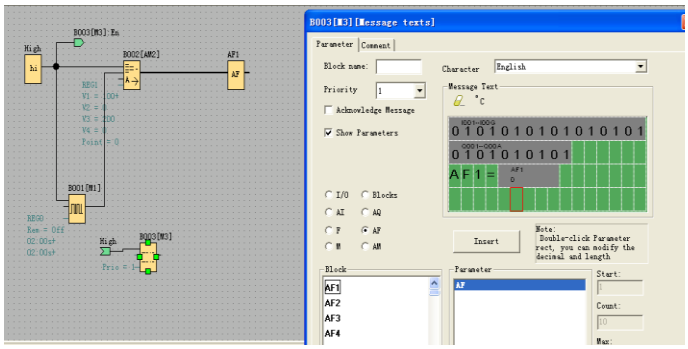
Program in Slave 1

Note: In the program, you can put the input/output block in, but you cannot link the input pin of the output block as this would disable remote control of the output.



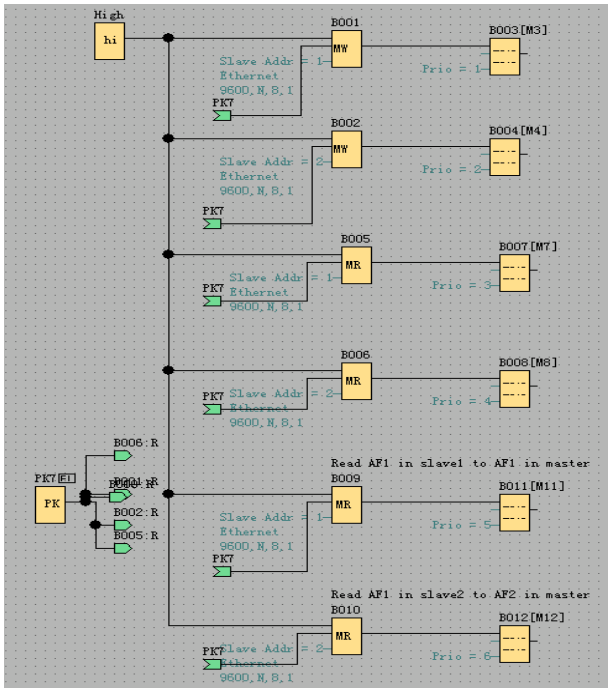
Program in Slave 2

Note: In the program, you can put the input/output block in, but you cannot link the input pin of the output block as this would disable remote control of the output.

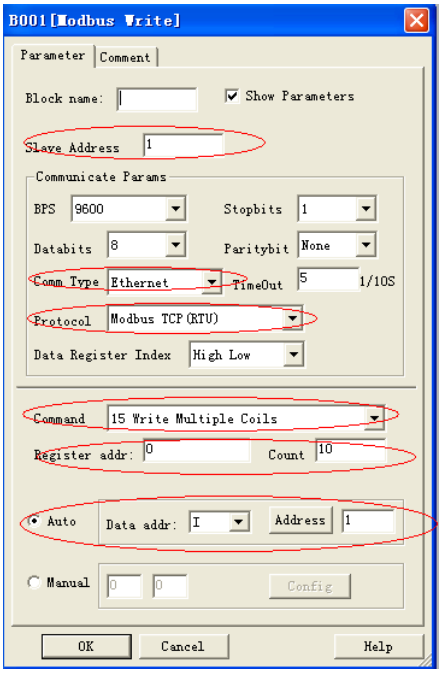


Program in Master

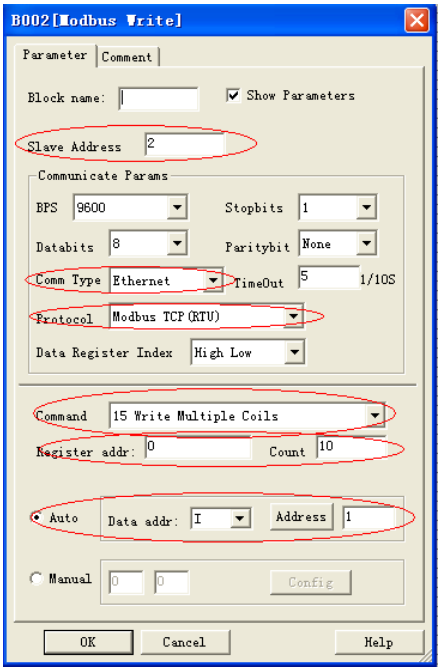
You need to use the MODBUS BLOCK to realize the data transmission between Master and Slaves.



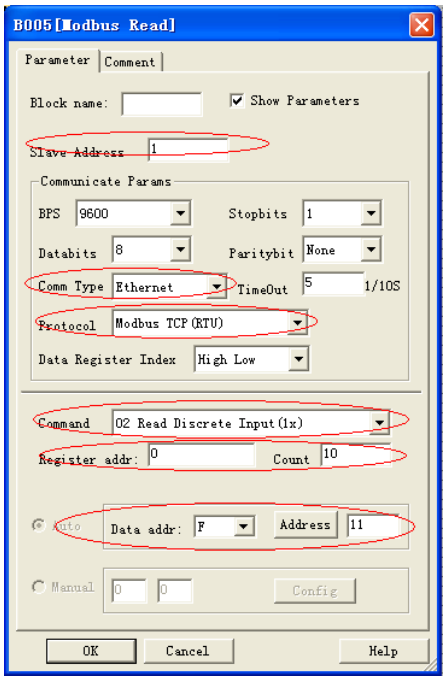
B001: Transfer the I1--IA status of the Master to the Q1-QA of the Slave1. Settings are as follows:



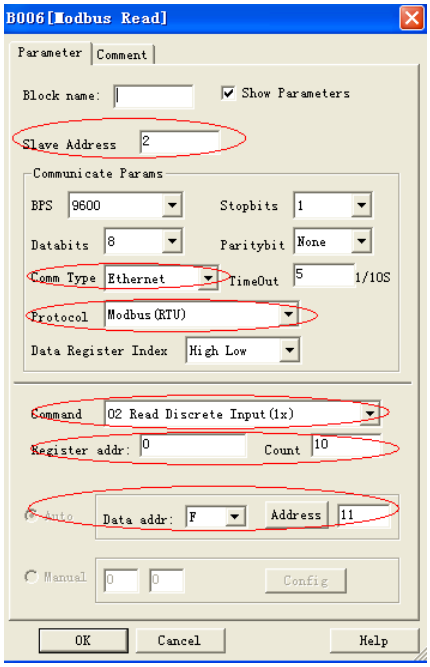
B002: Transfer the I1--IA status of the Master to the Q1-QA of the Slave2. Settings are as follows:



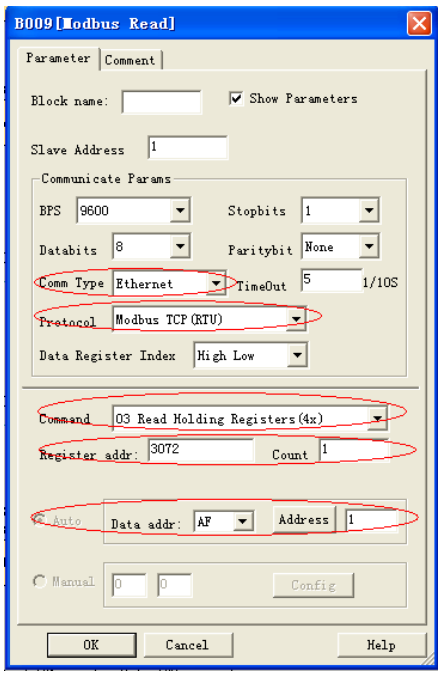
B005: Read the I1--IA status from Slave1 to the F11-F20 flags of the Master. Settings are as follows:



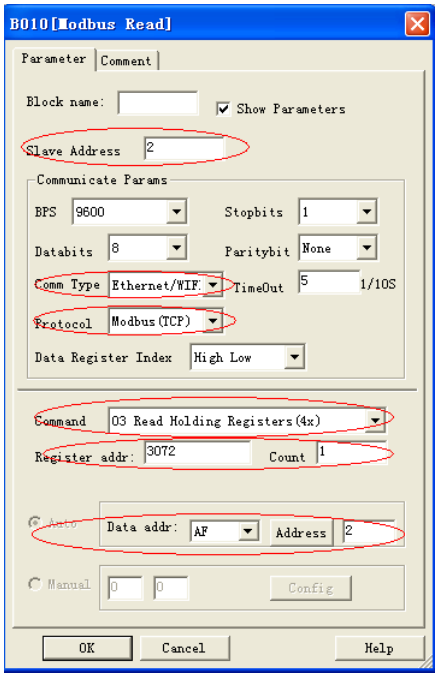
B006: Read the I1--IA status from Slave2 to the F21-F30 flags of the Master. Settings are as follows:



B009: Read the AF1 value from Slave1 to the AF1 flag of the Master. Settings are as follows:



B010: Read the AF1 value from Slave2 to the AF2 flag of the master. Settings are as follows:



NOTICE:

When you do not use the MODBUS blocks in your program, then the xLogic Ethernet CPU shall work as Slave. In this case, you can use the SCADA or a touch screen/HMI as the Master to communicate with the xLogic Ethernet CPU via the LAN port.

However, if MODBUS BLOCKS have been used in your program and the communication type is Ethernet, then the xLogic Ethernet CPU cannot work as a Slave via its Ethernet port.

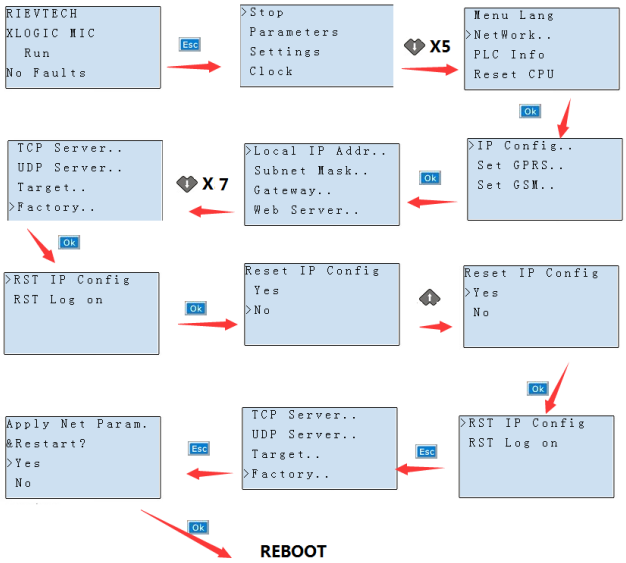
You can use the Modbus read/write blocks in your program to enable the xLogic Ethernet CPU to be a Modbus/TCP Master, which will enable it to talk to other Modbus/TCP Slaves on the Modbus network. These can be xLogic products or other manufacturers products that support Modbus/TCP communications.

8.6. Resetting to Factory Defaults

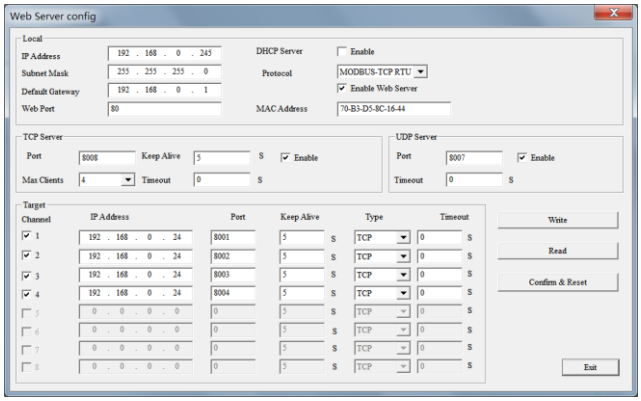
The Ethernet and WIFI configurations can be reset to factory defaults by following the procedures below. Only the communication settings will be reset and any program that is in the CPU will be unaffected.

Resetting Ethernet CPU to Defaults

The Ethernet CPUs are reset to factory defaults using the front HMI panel using the following steps.



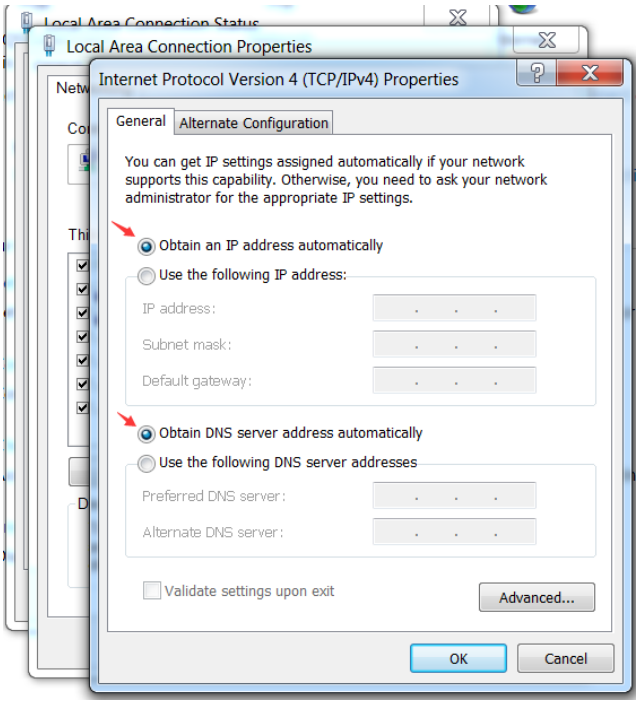
The default CPU Ethernet settings are:



Resetting WIFI CPU to Defaults

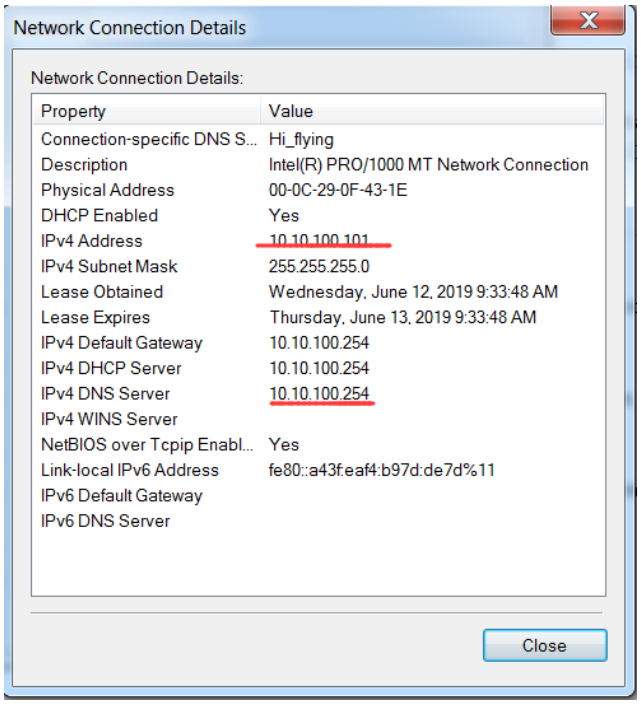
Our WIFI CPUs can be reset to factory defaults using a standard web browser as follows.

1. Configure the IP settings of your PCs Ethernet connection to automatically obtain an IP address:



2. Connect the PC to the RJ45 connection on the WIFI CPU using an Ethernet Crossover cable:

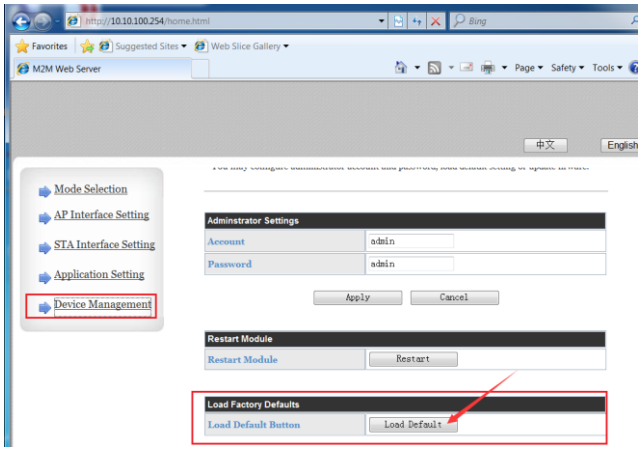
3. This WIFI CPU will provide an IP address to the PC:



4. Open a web browser and enter the IP address of the Default Gateway. In our example, 10.10.100.254 into the address bar of the web page and press Enter.
5. Enter the username and password when prompted in the pop-up window and click OK.
    a. Username: admin
    b. Password: admin

6. Click on the button 'English' if the text is not in that language:

7. Select the option 'Device Management' from the left-hand menu and then choose 'Load Default', which will reset the WIFI CPU to its default communication settings.



8. You will need to cycle the power to the SIFI CPU to restart it for the changes to become effective.

9. Applications

Below we have provided some example projects for which xLogic can be used.

Each instance includes the circuit program of its original solution and the xLogic program solution.

Example Projects:

- Dual-function switch
- Automatic gate
- Ventilation system
- Industry door
- Daylight lamps
- Rainwater pump

NOTICE:

These examples of xLogic are provided as is and free of charge to our clients, their form and function are for demonstration purposes and if used should be fully tested before implementation in a real application.

Also, we have to point out that errors may be present, and we reserve the right to correct and amend these examples without notice.

9.1. Dual-function switch

Requirements for stairway lighting systems.

The basic requirements for a stairway lighting system are as follows:

- When someone is using the stairs, the stairway lights should be on.
- If no one is in the stairway, the lights should go out in order to save energy.

Standard solution

Up to now, two methods have been used to control such a lighting system:

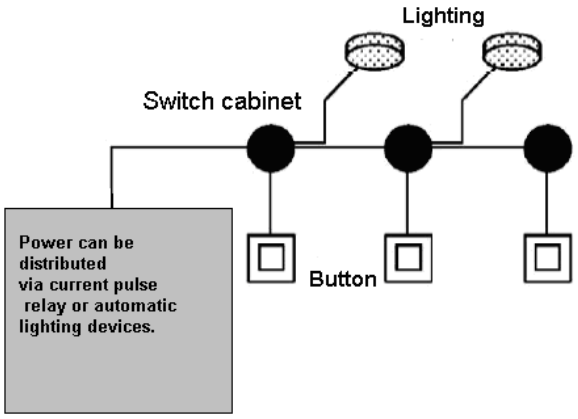
- Pulse relay: When the lights are off, press any of the pushbuttons to switch on the lights. When the lights are on, press any of the pushbuttons to switch off the lights again.

Disadvantage: People often forget to switch off the lights

- Automatic stairway light switch: Press any one of the pushbuttons to switch on the lights. The lights switch off again automatically when a pre-set off delay time has expired.

Disadvantage: You can't keep the lights switched on over an extended period of time. The permanent on switch, usually installed inside the stairway lighting timer unit, may be difficult or impossible to access.

The wiring is the same for both systems.



xLogicsoft solution

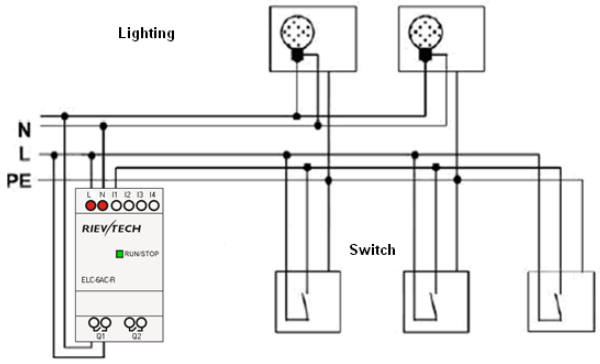
The xLogic system can replace the automatic stairway light switch or the pulse relay.

xLogic also lets you create a simple automatic stairway light switch via the stairway light switch SFB.

You can also implement both functions (off delay timer and pulse relay) in a single unit. What is more, you can incorporate extra functions without making any alterations to the wiring. In our example program, we have combined the advantages of both the current impulse relay and the automatic stairway lighting timer as follows:

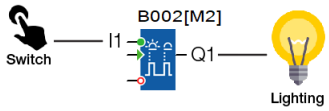
- Actuate the pushbutton - The light is switched on and switched off again on the expiration of a predefined time.
- Hold the pushbutton down - Switches on for continuous lighting
- Press the pushbutton once more - Switches off the lighting

The scheme of xLogic



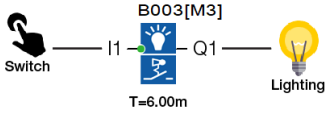
The wiring of a lighting system with xLogic is the same as standard corridor or stairway lighting systems. Only the automatic lighting timer/pulse relay is replaced.

xLogic lets you quickly and easily combine all those functions in a single dual-function switch SFB, without additional wiring and expenditure.



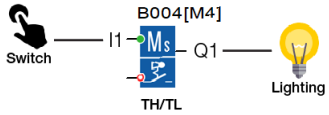
When the input "I1" has a pulse, the output "Q1" will be On.

Automatic stairway lighting system



If input "I1" has a pulse, the output Q1 will turn On for 6 minutes, then turn Off.

Apply xLogic to realize multiple switches



When the input "I1" has a pulse, the output "Q1" will turn On and when the timer "TH" expires then Q1 will turn Off.

If the momentary switch at I1 is held down for the time set in "TL" then the light, output Q1 will remain On.

Using Special Functions

The following selection can be done as a special function for saving energy:

The lighting flickers before it automatically goes off.

You can integrate different central control functions:

Central control Off

Central control On (Emergency button)

Control all lighting or certain single circuitry by lighting control switch.

To control by an integrated timer.

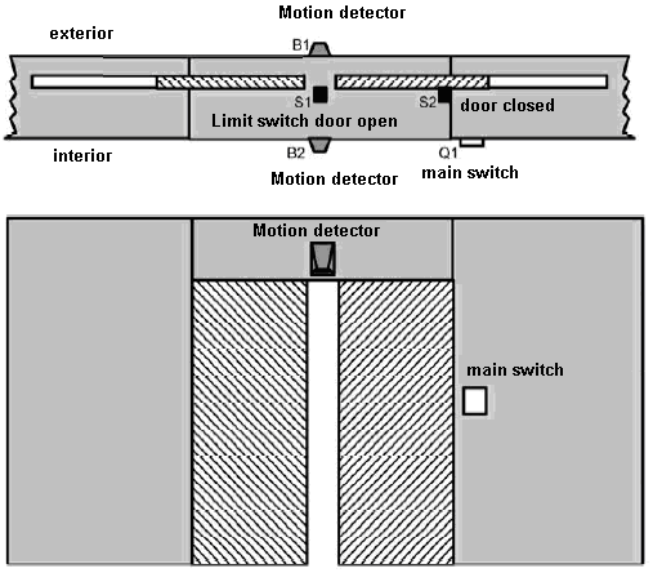
9.2. Automatic Gate

The entrance to many public buildings such as a supermarket、public building、bank、hospital, etc. an automatic entrance is often used.

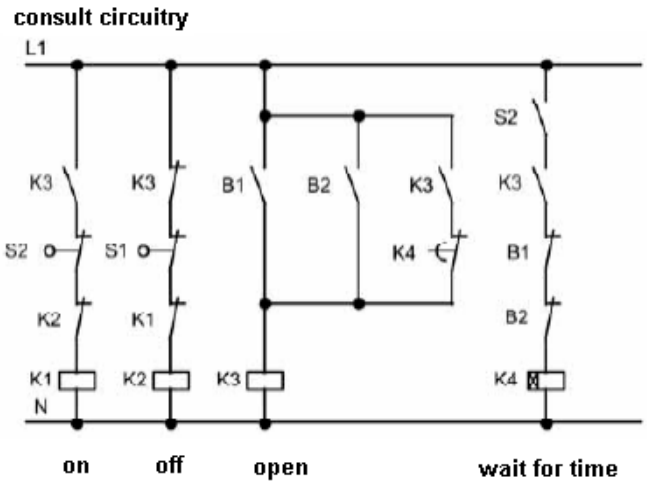
The requirements of the Automatic Gate are:

- When someone approaches the entrance, it will be opened automatically.

- The entrance must remain open until the area is clear of pedestrians.
- If the area is clear, then the gate must close automatically after one minute.



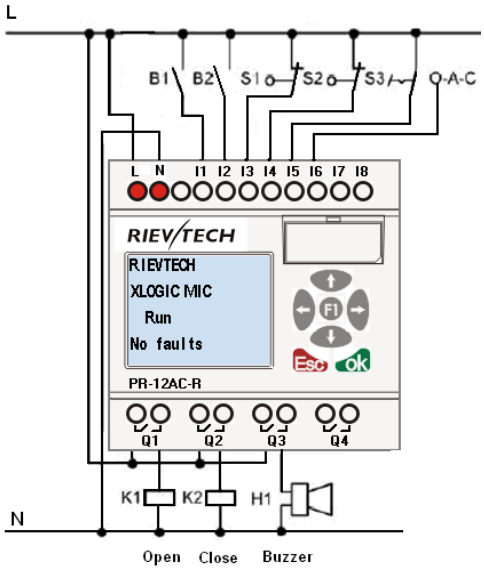
Standard solution



As long as the detector B1 or B2 detects someone approaching, the switch, K3 will be on and open the door.

If the two detectors don't detect a person after a short time, trigger K4 and close the door.

7.2.2 The scheme of xLogic



Required components:

K1 open contactor

K2 close contactor

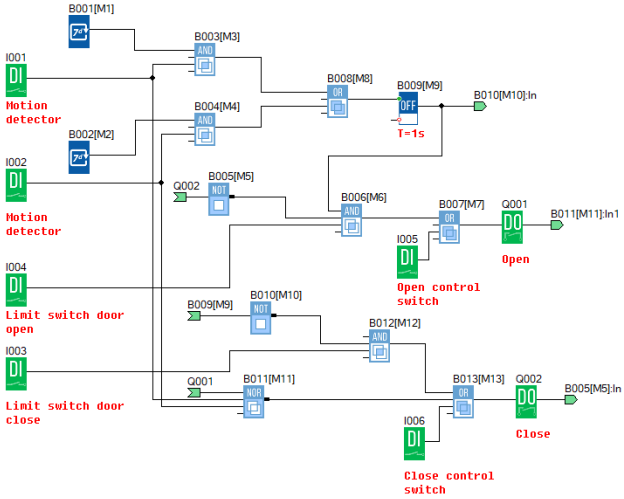
S1 (break contact) close limit switch

S2 (break contact) open limit switch

B1 (make contact) outdoor infrared detector

B2 (make contact) infrared detector inside

xLogic function block circuit program:



Motion Detector

During business hours, if someone enters the store, the detector B1 will trigger an electric motor to open the door, vice versa.

At closing time, the detector B2 controls the electric motor for a further hour to make enough time for customers to leave.

Trigger Electric Motor for Opening the Door

The output Q1 is switched On and triggers electric motor, when:

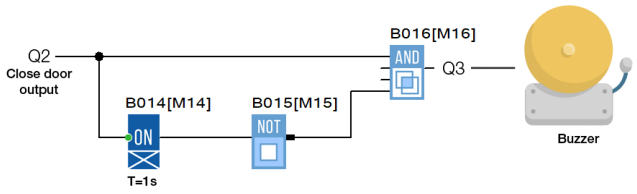
- Manual control switch I5 is On – the door is open all the time.
- Somebody is detected approaching the door.
- The door has not been opened fully – I4 limit switch is not Off.

Trigger Electric Motor for Closing the Door

- Manual control switch I6 is On – the door is closed all the time.
- Nobody is detected approaching the door.
- The door had not been closed fully – I3 limit switch is not Off.

Buzzer

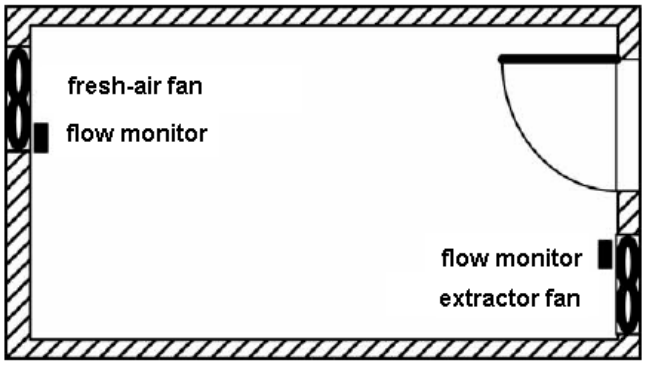
Connect the buzzer to output Q3. When the door is going to be closed; the buzzer gives off sounds for a short time (1s in this example). To attach the buzzer, we need to connect the following circuit program to output Q3.



9.3. Ventilation System

Requirements for a Ventilation system

A Ventilation system supplies fresh air into a room and exhausts the contaminated air. Let us look at the following sample system:



A room contains an extractor fan and a fresh-air fan.

Each fan is monitored by means of a flow sensor.

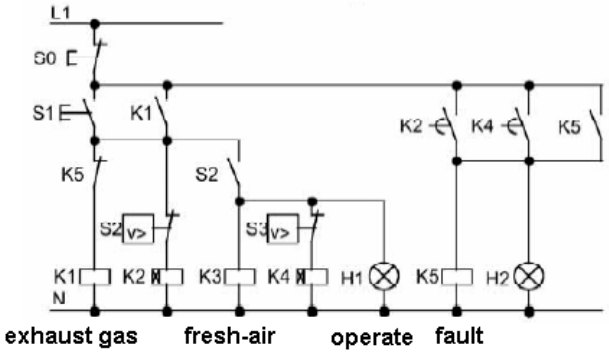
The pressure in the room may rise above atmospheric pressure.

The fresh-air fan may only be switched on if the flow sensor signals the safe operational state of the extractor fan.

A warning lamp indicates failure of one of the fans.

Standard Solution

The control circuit diagram of an example Ventilation system is as follows:

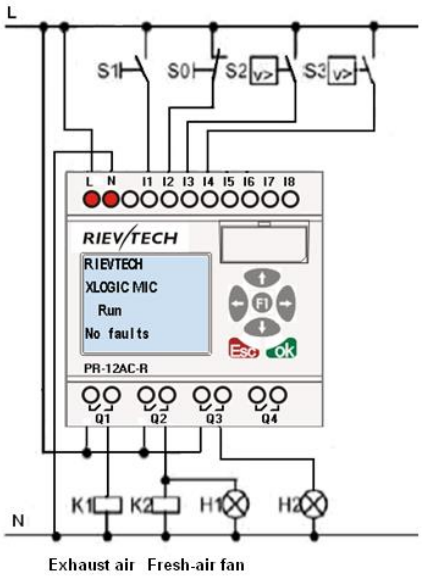


The fans are monitored by means of flow sensors. If no airflow is registered after a short delay time has expired, the system is switched off and an error message is generated, which can be acknowledged by pressing an Off button.

Fan monitoring requires an analyser circuit with several switching devices, in addition to the flow sensors. A single xLogic device can replace this analyser circuit.

The Scheme of xLogic

The circuit diagram of a ventilation system:



Required components:

K1 Main contactor

K2 Main contactor

S0 (make contact) Off button

S1 (make contact) On button

S2 (break contact) Flow monitor

S3 (break contact) Flow monitor

H1 Flashing lamp

H2 Flashing lamp

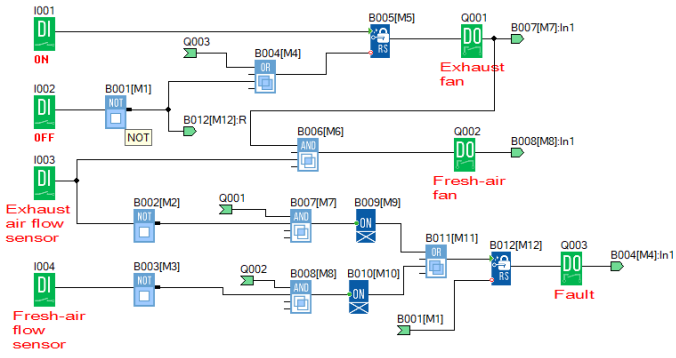
xLogic Solution

The use of xLogic reduces the amount of switchgear. Thus, you save installation time and space in the control cabinet. You may even be able to use as a smaller control cabinet.

With xLogic you can also switch off the fans sequentially after the system is switched off.

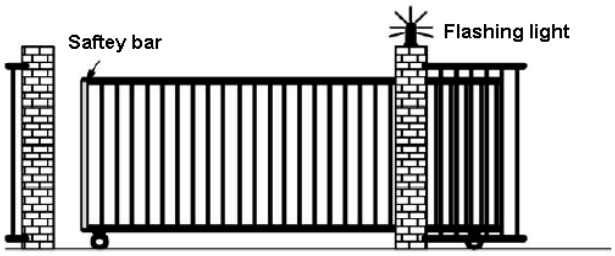
The circuit in xLogicsoft

The system is switched on and off via inputs I1 and I2. The fans are connected to outputs Q1 and Q2. The flow sensors are connected to the inputs I3 and I4. Blocks B07 and B10 are used to set the watchdog times. If any timer expires, the flow sensors should send a signal to the fault output Q3.



You can invert output Q3 to use output messages at Q4. Relay Q4 only drops out if main power is lost or if there is a fault in the system. The output can then be used for a remote message.

9.4. Factory Gate



Requirements for a Gate Control System

In many cases, a factory entrance is closed with automatic gates. Those gates are only opened when vehicles need to enter or leave the factory grounds. The gate is controlled by the porter.

The sliding gate is opened and closed by means of a pushbutton control in the gatehouse. The porter can monitor the gate operation.

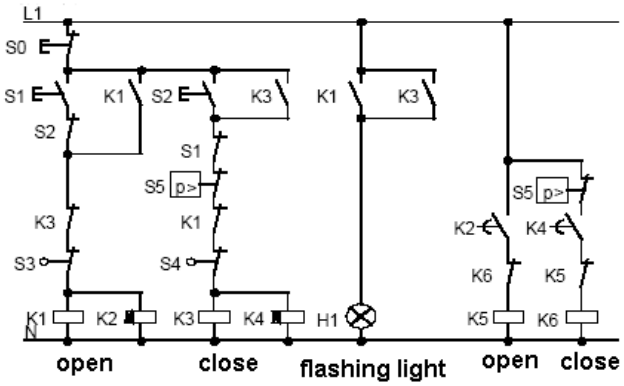
The rolling gate is normally fully opened, or it is closed. However, gate movements can always be interrupted.

A flashing light is activated five seconds before the gate moves, and while the gate is in motion.

A safety pressure strip ensures that people are not injured and that no objects are trapped and damaged when the gate is closing.

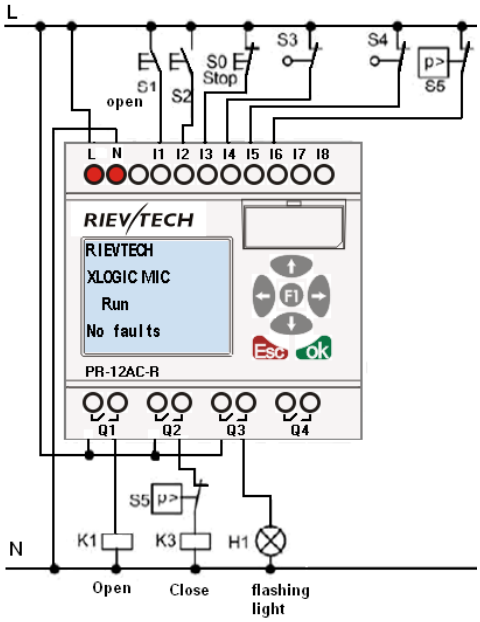
Standard Solution

There are many different control systems for operating automatic gates. The OPEN and CLOSE buttons initiate gate movements in the relevant direction, provided it is not already moving in the opposite direction. Movement of the gate is terminated either by means of the STOP button or the relevant limit switch.



The Scheme of xLogic

The circuit diagram of an industrial gate:



Required components:

K1 Main contactor

K2 Main contactor

S0 (break contact) Off button

S1 (make contact) Open button

S2 (make contact) Shutdown button

S3 (break contact) Open position sensor

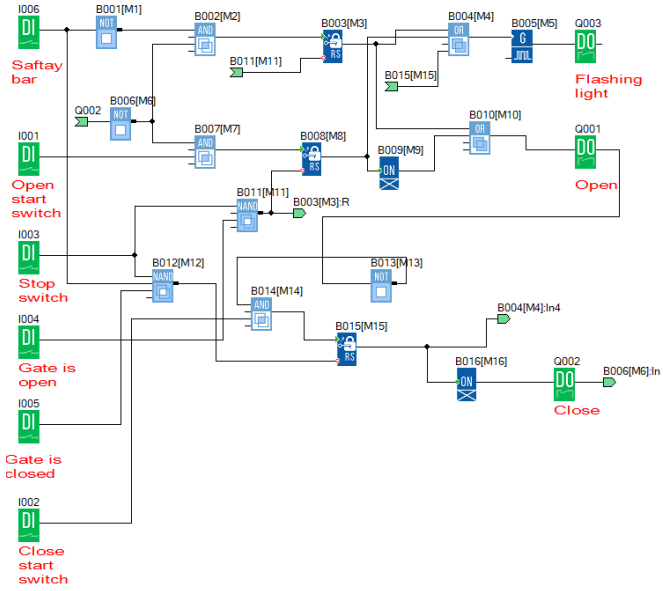
S4 (break contact) Shutdown position sensor

S5 (break contact) Safety bar

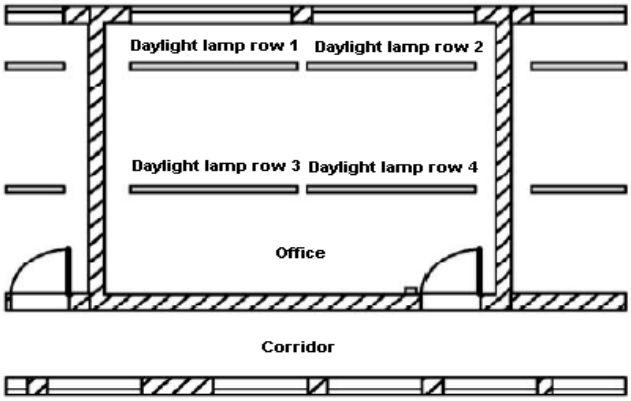
xLogicsoft Solution

The xLogic circuit provides a further feature compared to standard controls: The actuation of a safety bar interrupts the closing motion of the gate. Five seconds before the gate is opened or closed, a flashing light is activated and signals the start of the movement. It continues flashing until the gate has stopped.

In contrast to standard solutions, xLogic offers an economic means of modifying the control system.



9.5. Daylight Lamp System



Requirements for Lighting System

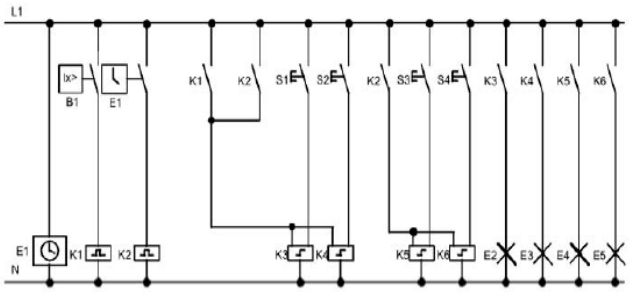
Different daylight lamp rows should be able to be switched on and off easily.

If the window at one side has enough light, the light will be switched off automatically via photosensor switch.

The light will be switched off automatically at 8:00 p.m.

The light can be switched on and off manually at any time.

Standard Solution



The lighting arrays can be operated by a pulse relay controlled by a button beside the door. This can be replaced by means of a timer and photosensor. The pulse relay may shorten the pulse width of the "off command".

Required components:

Button S1—S4

Daylight control switch B1

Timer T1

Pulse relay K1 and K2

Pulse switches K3—K6 able to be switched off collectively

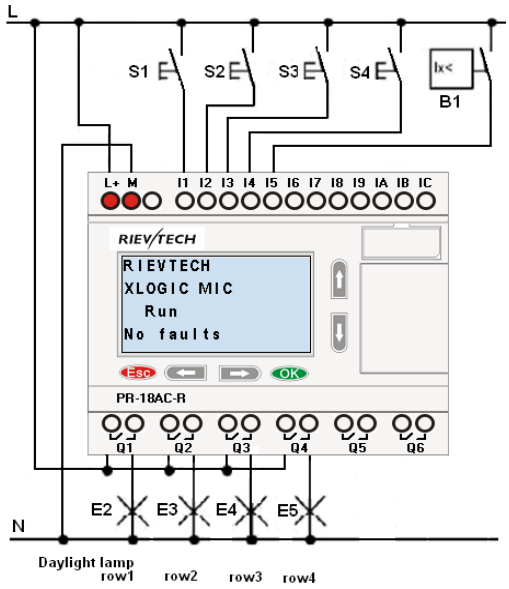
Disadvantages of Traditional Solution:

In order to realize function, it needs plenty of wiring.

Lots of mechanical parts will result in obvious failure over time and high maintenance costs.

Any modifications could be labour intensive.

The Scheme of xLogic

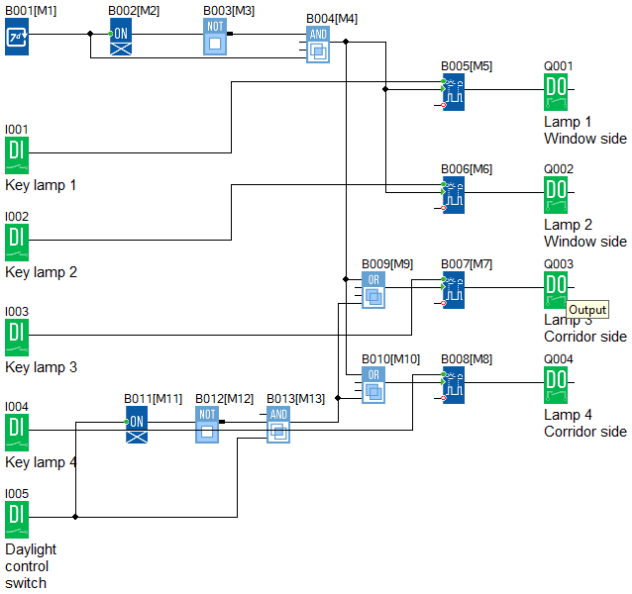


Components:

S1—S4 (make contact) Momentary switch

B1 (make contact) Daylight control switch

Circuit Diagram by xLogicsoft



Benefits:

Whilst the power consumption of load does not exceed the output of the xLogic outputs voltage range, the lamps can be directly connected to the xLogic main module. However, if the power consumption of the load exceeds the output of the switch's voltage range, then an additional interposing relay would be required.

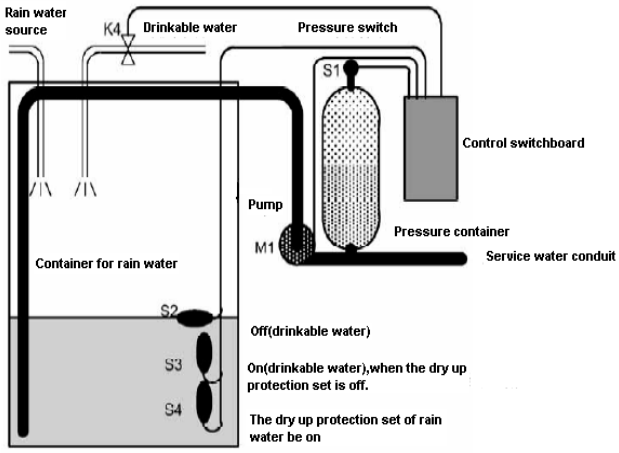
- The photosensor can be directly connected to the input of xLogic.
- Don't need an external timer, as this function has been integrated into xLogic.
- It can be installed in a small-sized cabinet, so quite space-saving.
- Less equipment.
- Quite easy to modify the lighting system.
- Further time controls can be added to provide extra flexibility.

9.6. Rainwater Pump

Nowadays besides drinkable water for your family, rainwater applications are gradually increasing. This can reduce water costs and also improve the environment as well. Rainwater can be used for the following as a small example:

- Washing clothes
- The water system in the garden
- Potted plant water
- Washing car

The following figure is to tell you how to run the rainwater application system:



The rainwater is collected in the container and then pumped to service water pipework via the pump M1. The system allows the tank to also be filled from the drinking water supply when there is not enough rain water.

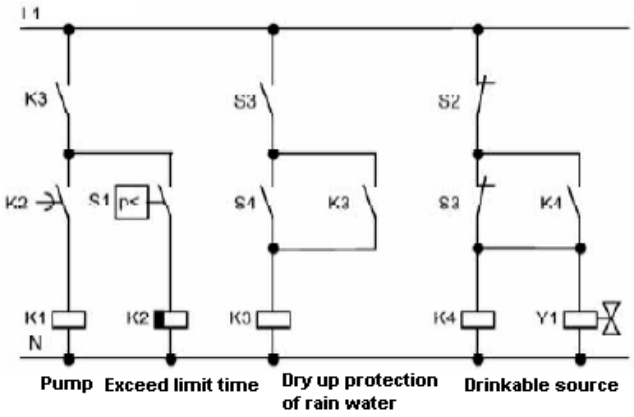
Requirements for the Control System of Service Water Pump

It should provide service water all day as needed. It should also allow for the automatic addition of drinkable water when required.

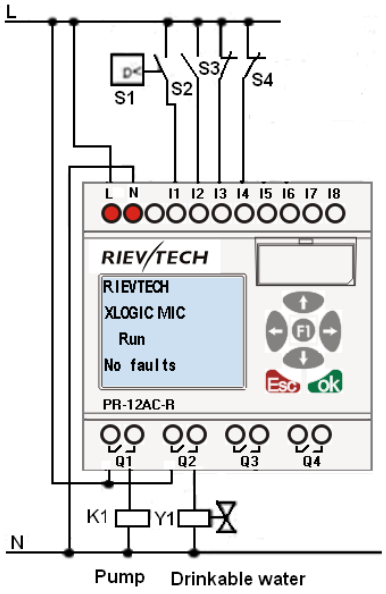
When switching to the drinkable water system, it can't interrupt rain water addition.

If the water level in the tank is too low, then the pump cannot be turned On – Dry run protection.

Standard Solution



The Scheme of xLogic



Components:

K1 main contractor

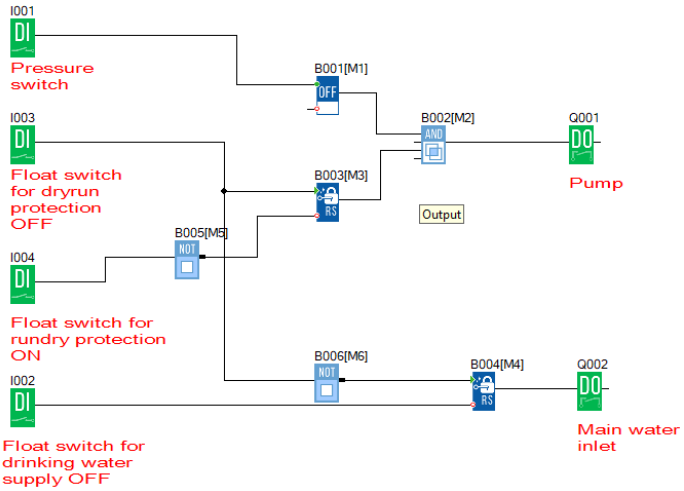
Y1 Solenoid valve

S1 Pressure switch

S2 (make contact) Float switch (water level)

S3—S4 (break contact) Float switch (water level)

Function block diagram:



10. Modbus Function Codes and Register Addresses

10.1. xLogic Supported Modbus Function Codes

The following table contains the Modbus function codes supported by xLogic.

Order code (Hex)	Function Description	Length of the message	Remarks
1	Read one group coil status (00000~0XXXX)	--	Read Coil Status (Output relay)
02	Fetch one group data of the status of switch input (10000~1XXXX)	--	Read input Status (input relay)
03	Read data of multi-holding register (40000~4XXXX)	--	Read Holding Registers (Output register)
05	Force the switch status of single coil (00000~0XXXX)	1	Force Single Coil
06	Pre-set the data of the single register (40000~4XXXX)	80	Set single output register
15	Force multi-coils on/off data (00000~0XXXX)	many	
16	Write multi-holding registers data (40000~4XXXX)		

10.2. Register Addresses of xLogic

Serial communication parameters settings:

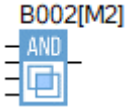
Modbus Communication Protocol: Modbus RTU

Communication Settings:

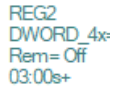
- Baud Rate: 9600
- Data Bits: 8
- Stop Bits: 1
- Parity: None

Name	Modbus Address (Decimal)	Data Format	Attribute
Digital Inputs **Type: 1x** **Modbus Function Code:** **02**	EXM-12(CPU): 0 ~ 7 ELC-12-N (CPU): 0 ~ 7 EXM-E-8 (EXT1): 8 ~ 15 EXM-E-8 (EXT2): 16 ~ 23 EXM-E-8 (EXT8): 64 ~ 71 PR-6 (CPU): 0 ~ 3 PR-12-E (CPU): 0 ~ 7 PR-12 (CPU): 0 ~ 7 ELC-22-N (CPU): 0 ~ 13 PR-E-16 (EXT1):	Bit	R

	16 ~ 23 PR-E-16 (EXT2): 24 ~ 31 PR-E-16 (EXT16): 136 ~ 143 PR-12-N (CPU): 0 ~ 7 PR-18-N (CPU): 0 ~ 11 PR-E-16 (EXT1): 12 ~ 19 PR-E-16 (EXT2): 20 ~ 27 PR-E-16 (EXT16): 132 ~ 139 PR-26-N (CPU): 0 ~ 15 PR-E-16 (EXT1): 16 ~ 23 PR-E-16 (EXT2): 24 ~ 31 PR-E-16 (EXT16): 136 ~ 143 PR-14 (CPU): 0 ~ 9 PR-18 (CPU): 0 ~ 11 PR-E-16 (EXT1): 12 ~ 19 PR-E-16 (EXT2):		
	20 ~ 27 PR-E-16 (EXT16): 132 ~ 139 PR-24 (CPU): 0 ~ 13 PR-E-16 (EXT1): 16 ~ 23 PR-E-16 (EXT2): 24 ~ 31 PR-E-16 (EXT16): 136 ~ 143		
4 Cursor Keys **Type: 1x Modbus Function Code: 02**	C1 – C4: 256 ~ 259	Bit	R
Digital Outputs **Type: 0x Modbus Function Code:** **01 (Read/Write)** **05 (Single Write)**	EXM-12 (CPU): 0 ~ 3 ELC-12-N (CPU): 0 ~ 3 EXM-E-8 (EXT1): 8 ~ 11 EXM-E-8 (EXT2): 16 ~ 19 EXM-E-8 (EXT8): 64 ~ 67 PR-6 (CPU): 0 ~ 1 PR-12-E (CPU): 0 ~ 3	Bit	R/W

15 (Multiple Write)	PR-12 (CPU): 0 ~ 3 ELC-22-N (CPU): 0 ~ 7 PR-E-16 (EXT1): 10 ~ 17 PR-E-16 (EXT2): 18 ~ 25 PR-E-16 (EXT16): 130 ~ 137 PR-12-N (CPU): 0 ~ 3 PR-18-N (CPU): 0 ~ 5 PR-E-16 (EXT1): 8 ~ 15 PR-E-16 (EXT2): 16 ~ 23 PR-E-16 (EXT16): 128 ~ 135 PR-26-N (CPU): 0 ~ 9 PR-E-16 (EXT1): 10 ~ 17 PR-E-16 (EXT2): 18 ~ 26 PR-E-16 (EXT16): 130 ~ 137 PR-14 (CPU): 0 ~ 3 PR-18 (CPU): 0 ~ 5		
	PR-E-16 (EXT1): 8 ~ 15 PR-E-16 (EXT2): 16 ~ 23 PR-E-16 (EXT16): 128 ~ 135 PR-24 (CPU): 0 ~ 9 PR-E-16 (EXT1): 10 ~ 17 PR-E-16 (EXT2): 18 ~ 26 PR-E-16 (EXT16): 130 ~ 137		
Middle Coil  **Type: 0x Modbus Function Code:** **01 (Read)**	EXM-12 Series: 256 ~ 767 ELC-12-N Series: 256 ~ 767 PR-6 Series: 256 ~ 319 PR-12-E Series: 256 ~ 319 PR-12 Series: 256 ~ 767 ELC-22-N Series: 256 ~ 767 PR-12-N Series: 256 ~ 1279 PR-18-N Series: 256 ~ 1279 PR-26-N Series: 256 ~ 1279 PR-14 Series: 256 ~ 1279	Bit	R

	PR-18 Series: 256 ~ 1279 PR-24 Series: 256 ~ 1279		
Digital Flag **Type: 0x Modbus Function Code:** **01 (Read/Write)** **05 (Single Write)** **15 (Multiple Write)**	EXM-12 Series: 1536 ~ 1663 ELC-12-N Series: 1536 ~ 1663 PR-6 Series: 1536 ~ 1567 PR-12-E Series: 1536 ~ 1567 PR-12 Series: 1536 ~ 1791 ELC-22-N Series: 1536 ~ 1663 PR-12-N Series: 1536 ~ 1791 PR-18-N Series: 1536 ~ 1791 PR-26-N Series: 1536 ~ 1791 PR-14 Series: 1536 ~ 1791 PR-18 Series: 1536 ~ 1791 PR-24 Series: 1536 ~ 1791	Bit	R/W
REG	EXM-12 Series: 0 ~ 511 ELC-12-N Series:	DWORD	R/W

Holding Register (Timer)  **Type: 4x Modbus Function Code:** **03 (Read/Write)** **16 (Multiple Write)**	0 ~ 511 PR-6 Series: 0 ~ 63 PR-12-E Series: 0 ~ 63 PR-12 Series: 0 ~ 511 ELC-22-N Series: 0 ~ 511 PR-12-N Series: 0 ~ 511 PR-18-N Series: 0 ~ 1023 PR-26-N Series: 0 ~ 1023 PR-14 Series: 0 ~ 511 PR-18 Series: 0 ~ 1023 PR-24 Series: 0 ~ 1023		
Analogue Input **Type: 4x Modbus Function Code:** **03 (Read)**	EXM-12 (CPU): 1024 ~ 1027 ELC-12-N (CPU): 1024 ~ 1027 EXM-E-8 (EXT1): 1028 ~ 1031 EXM-E-8 (EXT2): 1032 ~ 1035 EXM-E-8 (EXT8): 1080 ~ 1083	Signed Short	R

	PR-6 (CPU): 1024 ~ 1027 PR-12-E (CPU): 1024 ~ 1027 PR-12(CPU): 1024 ~ 1027 ELC-22-N (CPU): 1024 ~ 1031 PR-12-N (CPU): 1024 ~ 1027 PR-18-N (CPU): 1024 ~ 1031 PR-26-N (CPU): 1024 ~ 1039 PR-14 (CPU): 1024 ~ 1029 PR-18 (CPU): 1024 ~ 1029 PR-24 (CPU): 1024 ~ 1029 PR-E-16 (EXT1): 1032 ~ 1035 PR-E-16 (EXT2): 1040 ~ 1043 PR-E-16 (EXT16): 1152 ~ 1155		
Analogue Output **Type: 4x**	EXM-12 (CPU): 0 ELC-12-N (CPU): 0 EXM-E-AQ (EXT1): 1280 ~ 1281	Signed Short	R/W
Modbus Function Code: **03 (Read/Write)** **16 (Multiple Write)**	EXM-E-AQ (EXT2): 1282 ~ 1283 EXM-E-AQ (EXT8): 1294 ~ 1295 PR-6 (CPU): 0 PR-12-E (CPU): 0 PR-12 (CPU): 0 ELC-22-N (CPU): 0 PR-12-N (CPU): 0 PR-18-N (CPU): 0 PR-26-N (CPU): 1280 ~ 1281 PR-14 (CPU): 0 PR-18 (CPU): 0 PR-24 (CPU): 1280 ~ 1281 PR-E-AQ (EXT1): 1282 ~ 1283 PR-E-AQ (EXT2): 1284 ~ 1285 PR-E-AQ (EXT16): 1312 ~ 1313		
Analogue Quantity Buffer	EXM-12 (CPU): 1536 ~ 2047 ELC-12-N (CPU): 1536 ~ 2047	Signed Short	R

B006[AM6] **Type: 4x Modbus Function Code:** **03 (Read)**	PR-6 (CPU): 1536 ~ 1599 PR-12-E (CPU): 1536 ~ 1599 PR-12 (CPU): 1536 ~ 1599 ELC-22-N (CPU): 1536 ~ 2047 PR-12-N (CPU): 1536 ~ 2047 PR-18-N (CPU): 1536 ~ 2558 PR-26-N (CPU): 1536 ~ 2558 PR-14 (CPU): 1536 ~ 2047 PR-18 (CPU): 1536 ~ 2558 PR-24 (CPU): 1536 ~ 2558		
Analogue Flag **Type: 4x Modbus Function Code:** **03 (Read/Write)**	EXM-12 (CPU): 3072 ~ 3327 ELC-12-N (CPU): 3072 ~ 3199 PR-6 (CPU): 3072 ~ 3103 PR-12-E (CPU): 3072 ~ 3103 PR-12 (CPU): 3072 ~ 3327	Signed Short	R/W

16 (Multiple Write)	ELC-22-N (CPU): 3072 ~ 3327 PR-12-N (CPU): 3072 ~ 3327 PR-18-N (CPU): 3072 ~ 3327 PR-26-N (CPU): 3072 ~ 3327 PR-14 (CPU): 3072 ~ 3327 PR-18 (CPU): 3072 ~ 3327 PR-24 (CPU): 3072 ~ 3327		
HEG for Block **Type: 4x Modbus Function Code:** **03 (Read/Write)** **16 (Multiple Write)**	EXM-12 (CPU): 2560 ~ 3071 ELC-12-N (CPU): 2560 ~ 3071 PR-6 (CPU): 2560 ~ 2623 PR-12-E (CPU): 2560 ~ 3103 PR-12 (CPU): 2560 ~ 3327 ELC-22-N (CPU): 2560 ~ 3327	Word	R/W

	PR-12-N (CPU): 2560 ~ 3327 PR-18-N (CPU): 2560 ~ 3071 PR-26-N (CPU): 2560 ~ 3327 PR-14 (CPU): 2560 ~ 3071 PR-18 (CPU): PR-24 (CPU): HEG0 – HEG511: 2560 ~ 3071 HEG512 – HEG1023: 19456 ~ 19967		

Real Time Clock **Type: 4x** **Modbus** **Function Code:** **03 (Read/Write)** **16 (Multiple** **Write)**	**ALL CPU Series** **Year: 3328** **Month: 3329** **Day: 3330** **Hour: 3331** **Minute: 3332** **Sec: 3333**	**Short** **Signed**	**R/W**

Appendices

6
I/O
Not Expandal

PR-6

Model: PR-6AC-R AC

Model: PR-6DC-DA-R DC

Specifications

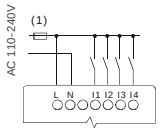
	Item	PR-6AC-R	PR-6DC-DA-R
Power	Nominal voltage	AC 110-240V	DC 12-24V
	Operating limits	AC 85 - 265V	DC 10.8-28.8V
	The main frequency range	47-63Hz	-
	Immunity from micro power	-	Typ. 5 ms
	Max. startup current	-	Max. 0.25A
	Isolation voltage	1780VAC	-
	Max. absorbed power	34 mA (85VAC) 26 mA (265VAC)	1.1W(10.8VDC) 1.2W (28.8V DC)
	Protection against polarity inversions	Yes	
Inputs	Input No.	4 (I1-I4)	
	Digital input	4 (I1-I4)	
	Analogue input	-	4 (I1-I4) (0..10V DC)
Inputs used as digital	Input voltage	AC 110-240V	DC 0-28.8V
	Input signal 0	AC 0-40V <0.03mA	< 5V DC <0.1mA
	Input signal 1	AC 79-240V >0.06mA	>8 V DC >0.3mA
	Input current	-	0.4mA @ 10.8V DC 0.5mA @ 12V DC 1.2mA @ 24V DC 1.5mA @ 28V DC
	Response time	0 to 1 120VAC : Typ. 50 ms 240VAC : Typ. 30 ms 1 to 0 120VAC : Typ. 90 ms 240VAC : Typ. 100 ms	0 to 1 Typ. 1.5 ms 1 to 0 Typ. 1.5 ms
	Maximum counting frequency	Typ: 4 Hz	
	Sensor type	Contact or 3-wire PNP	
	Input type	-	Resistive
	Isolation between power supply and inputs	-	
	Isolation between inpu	-	
	Protection against polarity inversions	Yes	-
Inputs used as analog	Measurement range	-	DC 0-10V
	Input impedance	-	Min. 24KΩ Max. 72KΩ
	Input voltage	-	28.8V DC max.
	Resolution	-	9bit 0.015V
	Accuracy at 25 癈	-	?(Max. 0.03)V
	Accuracy at 55 癈	-	?(Max. 0.06)V
	Isolation between analog inputs and power supp	-	
	Cable length	-	10 m max. shielded & twisted

	Item	PR-6
General data	Memory	64 Functional Blocks
	Data Retentivity	-
	Program Backup	10 years
	RTC	Backup at 25癈 30 days RTC accuracy : MAX ?S/day
	Cycle time	0.6ms - 8.0ms
	Expansion	-
	Communication	TTL interface Program/RS232, Modbus RTU/ASCII only can serve as slaves
	Certificate	CE RoHS cULus LISTED
	Operation Temp.	-20 癈 ~ +55 癈
	Storage Temp.	-40 癈 ~ +70 癈
	Protection	IP20
	Dimensions	48*90*64 (Unit: mm) 2DIN
	Installation	35mm-DIN rail or screw for installation
	Weight	Approx. 180g

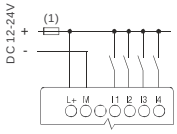
	Item	PR-6AC-R	PR-6DC-DA-R
Outputs	Output No.	2 (Q1-Q2)	
	Output type	Relay output	
	Continuous current	Resistive load 10A/Inductive load 2A	
	Max. breaking voltage	AC 250V DC 110V	
	Max. allowable power force	1250VA 300W	
	Electrical durability expectancy	10^5 Operations at Rated Resistive Load	
	Mechanical life	10^7 Operations at No Load condition	
	Built-in protections	Against short-circuit: None Against overvoltages and overloads: None	
	Response time	Operate Time : 15 ms max. Release Time : 10 ms max.	
Switch frequency	Mechanism	10Hz	
	Resistor/light load	2Hz	
	Sensitive load	0.5Hz	

I/O circuit diagrams

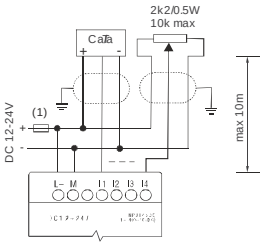
[Digital Input - Power AC]



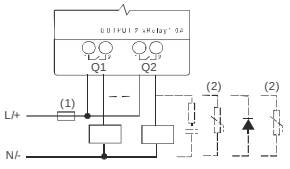
[Digital Input - Power DC]



[DC analog inputs 0-10V - Power DC]



[Relay Outputs]



(1) - Fuse, circuit-breaker or circuit protector
(2) - Inductive load;

12
I/O
Not Expandal

PR-12 Econon

Model: PR-12AC-R- AC
Model: PR-12DC-DA-R DC

PR-12

Model: PR-12AC- AC
Model: PR-12DC-DA- DC
Model: PR-12DC-DA-T DC

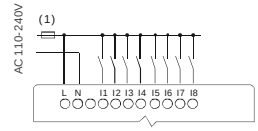
Specifications

	Item	PR-12AC-R-E	PR-12DC-DA-R-E	PR-12AC-R	PR-12DC-DA-R	PR-12DC-DA-TN
General data	Memory	64 Functional Blocks		512 Functional Blocks / 5K Steps		
	Data Retentivity	-		10 years		
	Program Backup	10 years				
	RTC	Backup at 25: 20 days RTC accuracy : MAX ?S/day				
	Cycle time	0.6ms - 8.0ms				
	Expansion	-				
	Communication	1 Program RS232 port Modbus RTU/ASCII only can be as slaves		1 Program RS232 port Modbus RTU/ASCII Can work either as slave or as master in Modbus network.		
	Certificate	CE RoHS cULus LISTED				
	Operation Temp	-20 癈 - +55 癈				
	Storage Temp	-40 癈 - +70 癈				
	Protection	IP20				
	Dimensions	72*90*61 (U,mm) 4DIN				
	Installation	35mm-DIN rail or screw for installation				
	Weight	Approx 800g				

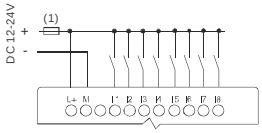
	Item	PR-12AC-R-E	PR-12AC-R	PR-12DC-DA-R-E	PR-12DC-DA-R	PR-12DC-DA-TN
Power	Nominal voltage	AC 110/240V		DC 12-24V		
	Operating limits	AC 85 - 265V		DC 10.8-28.8V		
	The main frequency range	47-63Hz		-		
	Immunity from micro power	-		Typ 5 ms		
	Max startup current	-		Max 0.25A		Max 0.1A
	Isolation voltage	1780VAC		-		
	Max absorbed power	38 mA (85VAC) 30 mA (265VAC)	48.5 mA (85VAC) 35 mA (265VAC)	3.2W (10.8VDC) 3.8W (28.8V DC)	3.5W (10.8VDC) 4W (28.8V DC)	2W (10.8VDC) 2.3W (28.8V DC)
	Protection against polarity inversions	Yes				
Inputs	Input No	8 (I1-I8)				
	Digital input	8 (I1-I8)				
	Analogue input	-		4 (I1-I4) (0..10V DC)		
Inputs used as digital	Input voltage	AC 110-240V		DC 0-28.8V		
	Input signal 0	AC 0-40V <0.03mA		(I1-I4) <0.1mA / (I5-I8) <1mA @ < 5V DC		
	Input signal 1	AC 79-240V >0.06mA		(I1-I4) >0.3mA / (I5-I8) >1.7mA @ > 8V DC		
	Input current	-		(I1-I4) 0.4mA / (I5-I8) 2.3mA @ 10.8V DC (I1-I4) 0.5mA / (I5-I8) 2.6mA @ 12V DC (I1-I4) 1.2mA / (I5-I8) 5.2mA @ 24V DC (I1-I4) 1.5mA / (I5-I8) 6.3mA @ 28.8V DC		
	Response time	0 to 1 120VAC : Typ. 50 ms 240AC : Typ. 30 ms 1 to 0 120VAC : Typ. 90 ms 240AC : Typ. 100 ms		(I1-I4) 0 to 1 Typ. 1.5 ms 1 to 0 Typ. 1.5 ms (I5-I8) 0 to 1 Typ. <1 ms 1 to 0 Typ. <1 ms		
	Maximum counting frequency	Typ: 4 Hz		I1-I4 4 Hz I5-I8 60 kHz		
	Sensor type	Contact or 3-wire PNP				
	Input type	-		Resistive		
	Isolation between power supply and inputs	-				
	Isolation between inpu	-				
	Protection against polarity inversions	Yes		-		
Inputs used as analog	Measurement range	-		DC 0-10V		
	Input impedance	-		Min 24KΩ Max 72KΩ		
	Input voltage	-		28.8V DC max		
	Resolution	-		9bit 0.015V	10bit 0.01V	
	Accuracy at 25 癈	-		?(Max 0.03)V	?(Max 0.02)V	
	Accuracy at 55 癈	-		?(Max 0.06)V	?(Max 0.04)V	
	Isolation between analog inputs and power supp	-				
	Cable length	-		10 m max shielded & twisted		

I/O circuit diagrams

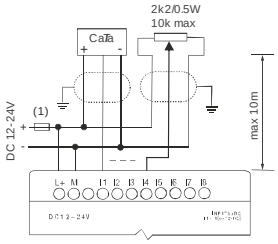
[Digital Input - Power AC]



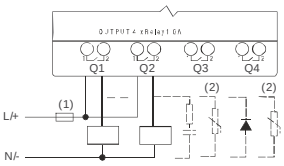
[Digital Input - Power DC]



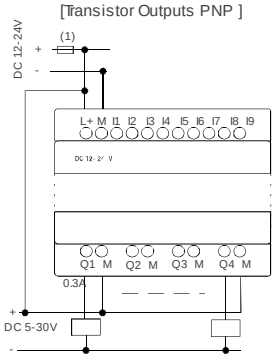
[DC analog inputs 0-10V - Power DC]



[Relay Outputs]



[Transistor Outputs PNP]



(1) - Fuse circuit-breaker or circuit protector
(2) - Inductive load;

Specifications

	Item	PR-12AC-R-E	PR-12DC-DA-R-E	PR-12AC-R	PR-12DC-DA-R	PR-12DC-DA-R-TN
Outputs	Output No	4 (Q1-Q4)				
	Output type	Relay output				Transistor (PNP)
	Continuous current	Resistive load 10A/Inductive load 2A				Max 0.3A per channel
	Max breaking voltage	AC 250V DC 110V				DC 5-30V
	Max breaking current	10A				0.65A
	Voltage drop	-				< 2 V for I = 0.3 (at state 1)
	Galvanic isolation	Yes				-
	Max allowable power force	1250VA 300W				-
	Electrical durability expectancy	105 Operations at Rated Resistive Load				-
	Mechanical life	107 Operations at No Load condition				-
	Built-in protections	Against short-circuit: None Against overvoltages and overloads: None				
	Response time	Operate Time : 15 ms max Release Time : 10 ms max				Make ≤ 1 ms Release ≤ 1 ms
Switch frequency	Mechanism	10Hz				-
	Resistor/light load	2Hz				10Hz
	Sensitive load	0.5Hz				
PWM Output	PWM frequency	-				10kHz
	PWM accuracy at 120H	-				< 0.5 % (20 % - 80 %) load at 10 mA
	PWM accuracy at 500H	-				< 0.5% (20 % - 80 %) load at 10 mA
	Max Breaking current PWM	-				50mA
	Max cable length PWM	-				20m
	PWM cyclic ratio	-				0 to 100 %

14 I/O Expandable

PR-14

Model: PR-14AC-R AC

Model: PR-14DC-DA-R DC

Specifications

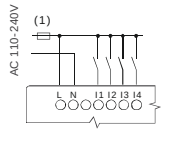
	Item	PR-14AC-R	PR-14DC-DA-R
Power	Nominal voltage	AC 110/240V	DC 12-24V
	Operating limits	AC 85 - 265V	DC 10.8-28.8V
	The main frequency range	47-63Hz	-
	Immunity from micro power	-	Typ 5 ms
	Max startup current	-	Max 0.25A
	Isolation voltage	1780VAC	-
	Max absorbed power	49 mA (85VAC) 37 mA (265VAC)	3.5W(10.8VDC) 4W (28.8V DC)
	Protection against polarity inversions	Yes	
Inputs	Input No	10 (I1-IA)	
	Digital input	10 (I1-IA)	
	Analogue input	-	6 (I1-I6) (0..10V DC)
Inputs used as digital	Input voltage	AC 110-240V	DC 0-28.8V
	Input signal 0	AC 0-40V <0.03mA	(I1-I6) <0.1mA / (I7-IA) <1mA @ < 5V DC
	Input signal 1	AC 79-240V >0.06mA	(I1-I6) >0.3mA / (I7-IA) >1.7mA @ > 8V DC
	Input current	-	(I1-I6) 0.4mA / (I7-IA) 2.3mA @ 10.8V DC (I1-I6) 0.5mA / (I7-IA) 2.6mA @V 12.0 (I1-I6) 1.2mA / (I7-IA) 5.2mA @DC24 (I1-I6) 1.5mA / (I7-IA) 6.3mA @V 28.8
	Response time	0 to 1 120VAC : Typ. 50 ms 240AC : Typ. 30 ms 1 to 0 120VAC : Typ. 90 ms 240AC : Typ.100 ms	(I1-I6)0 to 1 Typ. 1.5 ms 1 to: 0Typ. 1.5 ms (I7-IA)0 to 1 Typ. <1 ms 1 to: 0Typ. <1 ms
	Maximum counting frequency	Typ: 4 Hz	I1-I6: 4 Hz I7-IA: 60 kHz
	Sensor type	Contact or 3-wire PNP	
	Input type	-	Resistive
	Isolation between power supply and inputs	-	
	Isolation between inpu	-	
	Protection against polarity inversions	Yes	-
Inputs used as analog	Measurement range	-	DC 0-10V
	Input impedance	-	Min 24KΩ Max 72KΩ
	Input voltage	-	28.8V DC max
	Resolution	-	10bit 0.01V
	Accuracy at 25 癈	-	?(Max 0.02)V
	Accuracy at 55 癈	-	?(Max 0.04)V
	Isolation between analog inputs and power supp	-	
	Cable length	-	10 m max shielded & twisted

	Item	PR-14
General data	Memory	512 Functional Blocks / 5K Steps
	Data Retentivity	10 years
	Program Backup	10 years
	RTC	Backup at 25: 癈0 days RTC accuracy : MAX ?S/day
	Cycle time	0.6ms - 8.0ms
	Expansion	16 modules (PR-E-16)
	Communication	1 RS232 port & 2 RS485 port (1 built-in) Modbus RTU/ASCII Master or Slave
	Certificate	CE RoHS cULus LISTED
	OperatioTemp	-20 癈 - +55 癈
	StoragTemp	-40 癈 - +70 癈
	Protection	IP20
	Dimensions	95*90*55 (U,mm) 6DIN
	Installation	35mm-DIN rail or screw for installation
	Weight	Approx400g

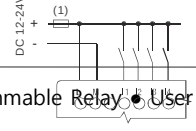
	Item	PR-14AC-R	PR-14DC-DA-R
Outputs	Output No	4 (Q1-Q4)	
	Output type	Relay output	
	Continuous current	Resistive load 10A/Inductive load 2A	
	Max breaking voltage	AC 250V DC 110V	
	Max allowable power force	1250VA 300W	
	Electrical durability expectancy	105 Operations at Rated Resistive Load	
	Mechanical life	107 Operations at No Load condition	
	Built-in protections	Against short-circuit noise Against overvoltages and overloads	
	Response time	OperatTime : 15 ms max ReleasTime : 10 ms max	
Switch frequency	Mechanism	10Hz	
	Resistor/light load	2Hz	
	Sensitive load	0.5Hz	

I/O circuit diagrams

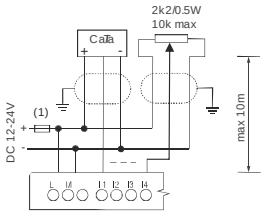
[Digital Input - Power AC]



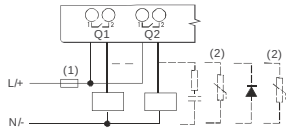
[Digital Input - Power DC]



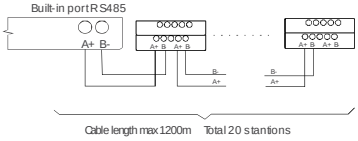
[DC analog inputs 0-10V - Power DC]



[Relay Outputs]



[RS485 Connection]



(1) - Fuse,circuit-breaker or circuit protector
(2) - Inductive load;

18 I/O Expandable

PR-18

Model: PR-18AC-R AC
Model: PR-18DC-DA-R DC
Model: PR-18DC-DA-RT DC

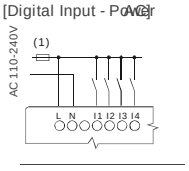
Specifications

	Item	PR-18AC-R	PR-18DC-DA-R	PR-18DC-DA-RT
Power	Nominal voltage	AC 110/240V	DC 12-24V	
	Operating limits	AC 85 - 265V	DC 10.8-28.8V	
	The main frequency range	47-63Hz	-	
	Immunity from micro power	-	Typ 5 ms	
	Max startup current	-	Max 0.25A	
	Isolation voltage	1780VAC	-	
	Max absorbed power	49 mA (85VAC) 37 mA (265VAC)	3.5W(10.8VDC) 4W (28.8V DC)	
	Protection against polarity inversions	Yes		
Inputs	Input No	12 (I1-IC)		
	Digital input	12 (I1-IC)		
	Analogue input	-	6 (I1-I6) (0..10V DC)	
Inputs used as digital	Input voltage	AC 110-240V	DC 0-28.8V	
	Input signal 0	AC 0-40V <0.03mA	(I1-I6) <0.1mA / (I7-IC) <1mA @ < 5V DC	
	Input signal 1	AC 79-240V >0.06mA	(I1-I6) >0.3mA / (I7-IC) >1.7mA @ > 8V DC	
	Input current	-	(I1-I6) 0.4mA / (I7-IC) 2.3mA @ 10.8V DC (I1-I6) 0.5mA / (I7-IC) 2.6mA @ 12.0V DC (I1-I6) 1.2mA / (I7-IC) 5.2mA @ 24.0V DC (I1-I6) 1.5mA / (I7-IC) 6.3mA @ 28.8V DC	
	Response time	0 to 1 120VAC : Typ. 50 ms 240VAC : Typ. 30 ms 1 to 0 120VAC : Typ. 90 ms 240VAC : Typ. 100 ms	(I1-I6) 0 to 1 Typ. 1.5 ms 1 to 0 Typ. 1.5 ms (I7-IC) 0 to 1 Typ. <1 ms 1 to 0 Typ. <1 ms	
	Maximum counting frequency	Typ: 4 Hz	I1-I8 4 Hz I9-IC 60 kHz	
	Sensor type	Contact or 3-wire PNP		
	Input type	-	Resistive	
	Isolation between power supply and inputs	-		
	Isolation between inpu	-		
	Protection against polarity inversions	Yes	-	
Inputs used as analog	Measurement range	-	DC 0-10V	
	Input impedance	-	Min 24KΩ Max 72KΩ	
	Input voltage	-	28.8V DC max	
	Resolution	-	10bit 0.01V	
	Accuracy at 25 癈	-	?(Max 0.02)V	
	Accuracy at 55 癈	-	?(Max 0.04)V	
	Isolation between analog inputs and power supp	-		
	Cable length	-	10 m max shielded & twisted	

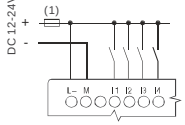
Item	PR-18
Memory	1024 Functional Blocks / 13K Steps
Data Retentivity	10 years
Program Backup	10 years
RTC	Backup at 25: 20 days RTC accuracy : MAX ?S/day
Cycle time	0.6ms - 8.0ms
Expansion	16 modules (PR-E-16)
Communication	1 RS232 port & 1 RS485 port (PR-E-RS485) Modbus RTU/ASCII Master or Slave
Certificate	CE RoHS cULus LISTED
Operation Temp	-20 癈 . +55 癈
Storage Temp	-40 癈 . +70 癈
Protection	IP20
Dimensions	95*90*55 (Unit: mm) 6DIN
Installation	35mm-DIN rail or screw for installation
Weight	Approx. 400g

I/O circuit diagrams

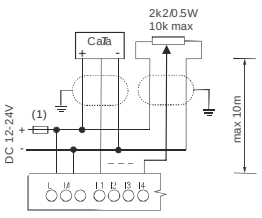
[Digital Input - Power AC]



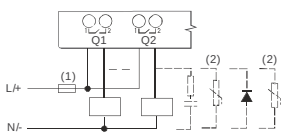
[Digital Input - Power DC]



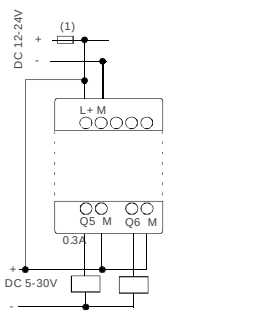
[DC analog inputs 0-10V - Power DC]



[Relay Outputs]



[Transistor Outputs PNP]



(1) - Fuse, circuit-breaker or circuit protector
(2) - Inductive load;

Specifications

	Item	PR-18AC-R	PR-18DC-DA-R	PR-18DC-DA-T	
				Relay	Transistor
Outputs	Output No	6 (Q1-Q6)			
	Output type	Relay output		4 (Q1-Q4) Relay output 2 (Q5-Q6) Transistor output	
	Continuous current	Resistive load 10A/Inductive load 2A			Max 0.3A per channel
	Max breaking voltage	AC 250V DC 110V			DC 5-30V
	Max breaking current	10A			0.65A
	Voltage drop	-			< 2 V for I = 0.3 (at state 1)
	Galvanic isolation	Yes			-
	Max allowable power force	1250VA 300W			-
	Electrical durability expectancy	105 Operations at Rated Resistive Load			-
	Mechanical life	107 Operations at No Load condition			-
	Built-in protections	Against short-circuit noise Against overvoltages and overloads			
	Response time	Operate Time : 15 ms max Release Time : 10 ms max			Make ≤ 1 ms Release ≤ 1 ms
Switch frequency	Mechanism	10Hz			-
	Resistor/light load	2Hz			10Hz
	Sensitive load	0.5Hz			
PWM Output	PWM frequency	-			10kHz Q5, Q6 must have same frequency when PWM works
	PWM accuracy at 120H	-			< 0.5 % (20 % - 80 %) load at 10 mA
	PWM accuracy at 500H	-			< 0.5% (20 % - 80 %) load at 10 mA
	Max Breaking current PWM	-			50mA
	Max cable length PWM	-			20m
	PWM cyclic ratio	-			0 to 100 %

24
I/O
Expandable

PR-24

Model: PR-24AC-R AC
Model: PR-24DC-DA-R DC
Model: PR-24DC-DA-RTA DC

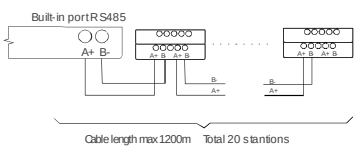
Specifications

	Item	PR-24AC-R	PR-24DC-DA-R	PR-24DC-DA-RTA
Power	Nominal voltage	AC 110-240V	DC 12-24V	
	Operating limits	AC 85 - 265V	DC 10.8-28.8V	
	The main frequency range	47-63Hz	-	
	Immunity from micro power	-	Typ 5 ms	
	Max startup current	-	Max 0.25A	
	Isolation voltage	1780VAC	-	
	Max absorbed power	49 mA (85VAC) 37 mA (265VAC)	3.5W(10.8VDC) 4W (28.8V DC)	
	Protection against polarity inversions	Yes		
Inputs	Input No	14 (I1-IE)		
	Digital input	14 (I1-IE)		12 (I1-I4)(I7-IE)
	Analogue input	-	6 (I1-I6) (0..10V DC)	4 (I1-I4) (0..10V DC) 2 (I5-I6) (0..20mA)
Inputs used as digital	Input voltage	AC 110-240V	DC 0-28.8V	
	Input signal 0	AC 0-40V <0.03mA	(I1-I4) <0.1mA / (I7-IE) <1mA @ < 5V DC	
	Input signal 1	AC 79-240V >0.06mA	(I1-I4) >0.3mA / (I7-IE) >1.7mA @ > 8V DC	
	Input current	-	(I1-I4) 0.4mA / (I7-IE) 2.3mA @ 10.8V DC (I1-I4) 0.5mA / (I7-IE) 2.6mA @ 12V DC (I1-I4) 1.2mA / (I7-IE) 5.2mA @ 24V DC (I1-I4) 1.5mA / (I7-IE) 6.3mA @ 28.8V DC	
	Response time	0 to 1 120VAC : Typ. 50 ms 240VAC : Typ. 30 ms 1 to 0 120VAC : Typ 90 ms 240VAC : Typ.100 ms	(I1-I6)0 to 1 Typ. 1.5 ms 1 to:0Typ. 1.5 ms (I7-IE)0 to 1 Typ. <1 ms 1 to:0Typ. <1 ms	
	Maximum counting frequency	Typ: 4 Hz	I1-I4 I7-I8 ID-IE 4 Hz I9-IC 60 kHz	
	Sensor type	Contact or 3-wire PNP		
	Input type	-	Resistive	
	Isolation between power supply and inputs	-		
	Isolation between inpu	-		
	Protection against polarity inversions	Yes	-	
Inputs used as analog	Measurement range	-	DC 0-10V	
	Input impedance	-	Min 24KΩ Max 72KΩ	
	Input voltage	-	28.8V DC max	
	Resolution	-	10bit 0.01V	
	Accuracy at 25 癈	-	?(Max 0.02)V	
	Accuracy at 55 癈	-	?(Max 0.04)V	
	Isolation between analog inputs and power supp	-		
	Cable length	-	10 m max shielded & twisted	
Current Inputs	Current input No	-		2 (I5-I6)
	Analogue signal	-		0/4?20mA current
	Resolution	-		0.02mA
	Accuracy at 25 癈	-		0.05mA
	Cycle time for analog value generation	-		Typ. 50 ms

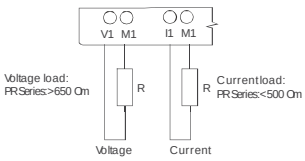
Item	PR-24
Memory	1024 Functional Blocks / 13K Steps
Data Retentivity	10 years
Program Backup	10 years
RTC	Backup at 25癈 30 days RTC accuracy : MAX ?S/day
Cycle time	0.6ms - 8.0ms
Expansion	16 modules (PR-E-16)
Communication	1 RS232 port & 2 RS485 port (1 Built-in) Modbus RTU/ASCII Master or Slave
Certificate	CE RoHS cULus LISTED
Operation Temp	-20 癈 - +55 癈
Storage Temp	-40 癈 - +70 癈
Protection	IP20
Dimensions	133*90*55 (Unit:mm) 10DIN
Installation	35mm-DIN rail or screw for installation
Weight	Approx.500g

Circuit diagrams

[RS485 Connection]

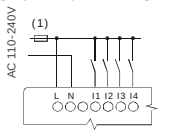


[DCAnalog Outputs - 0-10V / 0-20mA]

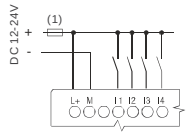


Circuit diagrams

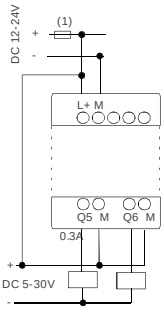
[Digital Input - Power AC]



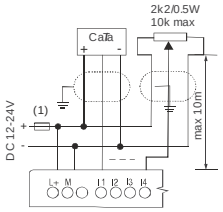
[Digital Input - Power DC]



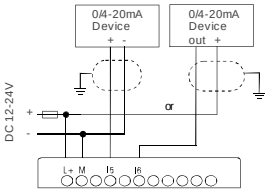
[Transistor Outputs PNP]



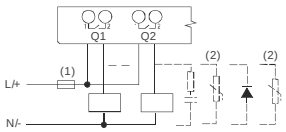
[DC analog inputs 0-10V - Power DC]



[DC analog inputs 0-20mA - Power DC]



[Relay Outputs]



(1) - Fuse,circuit-breaker or circuit protector
(2) - Inductive load;

Specifications

	Item	PR-24AC-R	PR-24DC-DA-R	PR-24DC-DRTA	
				Relay	Transistor
Outputs	Output No	10 (Q1-QA)		9 (Q1-Q8,AQ9)	
	Output type	Relay output		6 (Q1-Q6) Relay output 2 (Q7-Q8) Transistor PNP output 1 (AQ9) Analog output (0-10/0-20mA)	
	Continuous current	Resistive load 10A/Inductive load 2A			Max 0.3A per channel
	Max breaking voltage	AC 250V DC 110V			DC 5-30V
	Max breaking current	10A			0.65A
	Voltage drop	-			< 2 V for I = 0.3 (at state 1)
	Galvanic isolation	Yes			-
	Max allowable power force	1250VA 300W			-
	Electrical durability expectancy	105 Operations at Rated Resistive Load			-
	Mechanical life	107 Operations at No Load condition			-
	Built-in protections	Against short-circuit noise Against overvoltages and overloads			
	Response time	Operate Time : 15 ms max Release Time : 10 ms max			Make ≤ 1 ms Release ≤ 1 ms
Switch frequency	Mechanism	10Hz			-
	Resistor/light load	2Hz			10Hz
	Sensitive load	0.5Hz			
PWM Output	PWM frequency	-			10kHz Q7, Q8 must have same frequency when PWM works
	PWM accuracy at 120Hz	-			< 0.5 % (20 % - 80 %) load at 10 mA
	PWM accuracy at 500Hz	-			< 0.5% (20 % - 80 %) load at 10 mA
	Max Breaking current PWM	-			50mA
	Max cable length PWM	-			20m
	PWM cyclic ratio	-			0 to 100 %
AQ Voltage signal	Output signal	-		DC 0?0V	
	Internal value and signal relationship	-		AQ9 (0..1000)¥1(0...10V)	
	Resolution	-		0.01V	
	Accuracy at 25 癈	-		0.02V	
AQ Current signal	Output signal	-		0..20mA	
	Internal value and signal relationship	-		AQ9 (0..1000) = I1(0...20mA)	
	Resolution	-		0.02mA	
	Accuracy at 25 癈	-		0.05mA	

Extension Modules

PR-E

Model:PR-E-16AC- AC

Model:PR-E-16DC-DA DC

Model:PR-E-16DC-DA-T DC

Model:PR-E-AI- DC

Model:PR-E-PT10 DC

Model:PR-EAQ-VI DC

Model:PR-E-RS48 DC

Specifications

	Item	PR-E-16AC-R	PR-E-16DC-DA-R	PR-E-16DC-DA-T
Power	Nominal voltage	AC 110240V	DC 12-24V	
	Operating limits	AC 85 - 265V	DC 10.8-28.8V	
	The main frequency range	47-63Hz	-	
	Immunity from micro power	-	Typ 5 ms	
	Max startup current	-	Max 0.25A	
	Isolation voltage	1780VAC	-	
	Max absorbed power	53 mA (85AC) 38 mA (265AC)	3.5W(10.8VDC) 4.5W (28.8V DC)	
	Protection against polarity inversions	Yes		
Inputs	Input No	8 (I1-I8)		
	Digital input	8 (I1-I8)		
	Analogue input	-	4 (I1-I4) (0..10VDC)	
Inputs used as digital	Input voltage	AC 110-240V	DC 0-28.8V	
	Input signal 0	AC 0-40V <0.03mA	(I1-I4) <0.1mA / (I5-I8) <1mA @ < 5V DC	
	Input signal 1	AC 79-240V >0.06mA	(I1-I4) >0.3mA / (I5-I8) >1.7mA @ > 8V DC	
	Input current	-	(I1-I4) 0.4mA / (I5-I8) 2.3mA @ 10.8V DC (I1-I4) 0.5mA / (I5-I8) 2.6mA @ 12DC (I1-I4) 1.2mA / (I5-I8) 5.2mA @ 24DC (I1-I4) 1.5mA / (I5-I8) 6.3mA @ 28.8DC	
	Response time	0 to 1 120VAC : Typ. 50 ms 240AC : Typ. 30 ms 1 to 0 120VAC : Typ. 90 ms 240AC : Typ.100 ms	(I1-I4)0 to 1 Typ. 1.5 ms 1 to:0Typ. 1.5 ms (I5-I8)0 to 1 Typ. <1 ms 1 to:0Typ. <1 ms	
	Maximum counting frequency	Typ: 4 Hz	4 Hz	
	Sensor type	Contact or 3-wire PNP		
	Input type	-	Resistive	
	Isolation between power supply and inputs	-		
	Isolation between inpu	-		
	Protection against polarity inversions	Yes	-	
Inputs used as analog	Measurement range	-	DC 0-10V	
	Input impedance	-	Min 24KΩ Max 72KΩ	
	Input voltage	-	28.8V DC max	
	Resolution	-	9bit 0.015V	
	Accuracy at 25 癈	-	?(Max 0.03)V	
	Accuracy at 55 癈	-	?(Max 0.06)V	
	Isolation between analog inputs and power supp	-		
	Cable length	-	10 m max shielded & twisted	

	Item	PR-E-RS485
Power	Nominal voltage	DC 12-24V
	Operating limits	DC 10.8-28.8V
	Immunity from micro power	Typ 5 ms
	Max Startup current	Max 0.1A
	Max absorbed power	1.5W(10.8VDC) 1.8W W (28.8V DC)
	Protection against polarity inversions	Yes

	Item	PR-E
General data	Compatibility	PR-14PR-18PR-24
	Certificate	CE RoHS cULus LISTED
	Operation Temp	-20 癈 - +55 癈
	Storage Temp	-40 癈 - +70 癈
	Protection	IP20
	Dimensions	72*90*61 (Unit:m) 4DIN
	Installation	35mm-DIN rail or screw for installation
	Weight	Approx.300g

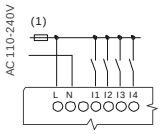
	Item	PR-E-AI-I	PR-E-PT100	PR-EAQ-VI
Power	Nominal voltage	DC 12-24V		
	Operating limits	DC 10.8-28.8V		
	Immunity from micro power	Typ 5 ms		
	Max Startup current	Max 0.25A		
	Max absorbed power	1W		1.8W
	Protection against polarity inversions	Yes		
Inputs	Input No	4 (I1-I4)	3 (I1-I3)	-
	Digital input	-		
	Analogue input	4 (AI1-AI4)	3 (AI1 -AI3)	-
	Analogue signal	0/4?20mA current	RTD PT100	-
	Resolution	0.02mA 10bit	0.3℃ 10bit	-
	Measuring range	-	-50℃ to +200℃	-
	Accuracy at 25 癈	0.05mA	-	-
Outputs	Output No	-		2(AQ1-AQ2)
	Output signal	-		DC 0? 0V or 0..20mA
	Internal value and signal relationship	-		AQ1/2 (0..1000)= V1(0...10V)= I1(0...20mA)
	Resolution	-		0.01V or 0.02mA 10bit
	Accuracy at 25 癈	-		0.02V or 0.05mA
Other	Cycle time for analog value generation	Typ. 50 ms		
	Cable length	10 m max shielded & twisted		

Specifications

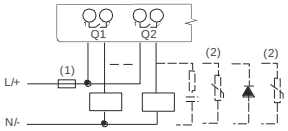
	Item	PR-E-16AC-R	PR-E-16DC-DA-R	PR-E-16DC-DA-TN
Outputs	Output No	8 (Q1-Q8)		
	Output type	Relay output		Transistor PNP output
	Continuous current	Q1-Q4 Resistive load 3A/Inductive load 1A Q5-Q8 Resistive load 10A/Inductive load 2A		Max 0.3A per channel
	Max breaking voltage	AC 250V DC 110V		DC 5-30V
	Max breaking current	10A		0.65A
	Voltage drop	-		< 2 V for I=0.3 (at state 1)
	Galvanic isolation	Yes		-
	Max allowable power force	(Q1-Q4) 500VA 100W (Q5-Q8) 1250VA 300W		-
	Electrical durability expectancy	105 Operations at Rated Resistive Load		-
	Mechanical life	107 Operations at No Load condition		-
	Built-in protections	Against short-circuit: None Against overvoltages and overloads: None		
	Response time	Operate Time : 15 ms max Release Time : 10 ms max		Make ≤ 70 ms Release ≤ 70 ms
Switch frequency	Mechanism	10Hz		-
	Resistor/light load	2Hz		10Hz
	Sensitive load	0.5Hz		

Circuit diagrams

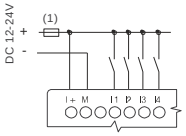
[Digital Input - Power AC]



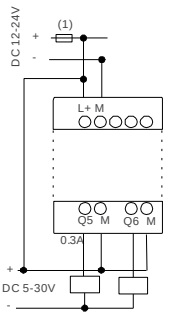
[Relay Outputs]



[Digital Input - Power DC]

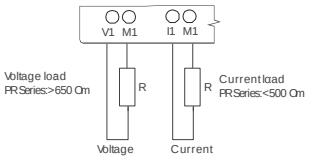


[Transistor Outputs PNP]



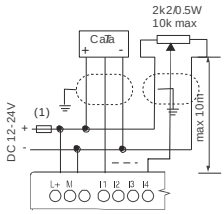
(1) - Fuse, circuit-breaker or circuit protector
(2) - Inductive load;

[DC Analog Outputs - 0-10V / 0-20mA]

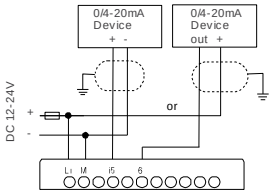


Circuit diagrams

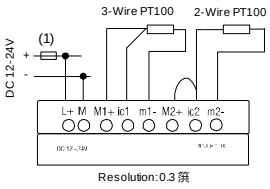
[DC analog inputs 0-10V - Power DC]



[DC analog inputs 0-20mA - Power DC]

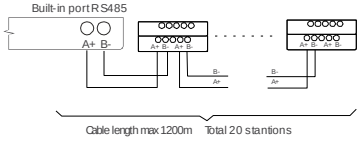


[Pt100 Connection (-50~200)]



(1) - Fuse, circuit-breaker or circuit protector

[RS485 Connection]

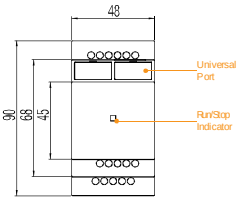


PR-6 Series

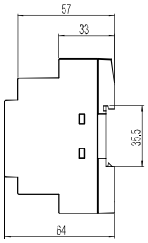
•PR-6AC-R •PR-6DC-DA-R

(Unitmm)

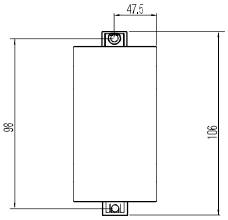
Front View



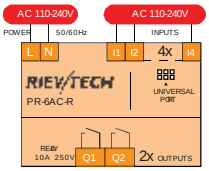
Side View



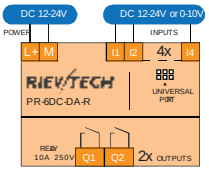
Mounting



PR-6AC-R



PR-6DC-DA-R

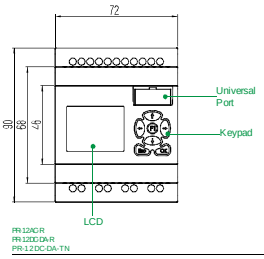


PR-12 Series

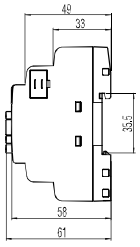
•PR-12AC-R-E •PR-12DC-DA-R-E •PR-12AC-R •PR-12DC-DA-R •PR-12DC-DA-TN

(Unitmm)

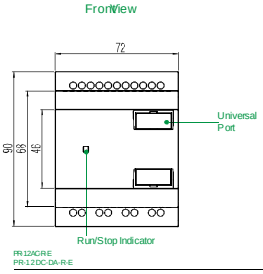
Front View



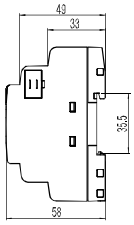
Side View



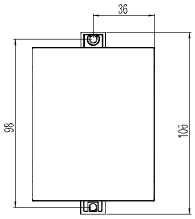
Front View



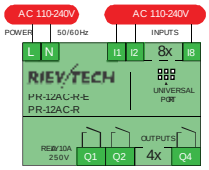
Side View



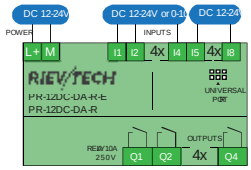
Mounting



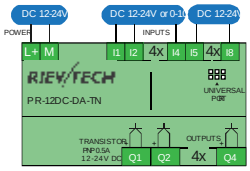
PR-12AC-R-E / PR-12AC-R



PR-12DC-DA-R-E / PR-12DC-DA-R



PR-12DC-DA-TN

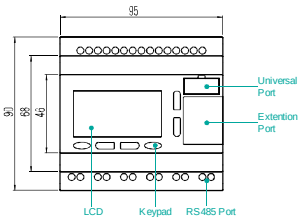


PR-14 Series

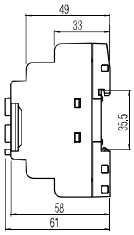
•PR-14AC-R •PR-14DC-DA-R

(Unitmm)

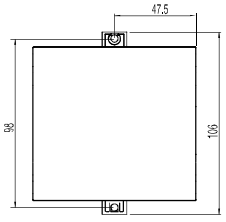
Front View



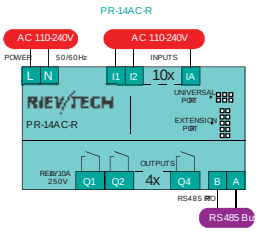
Side View



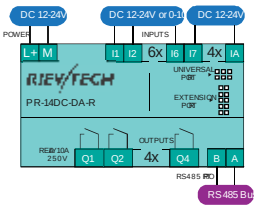
Mounting



PR-14AC-R



PR-14DC-DA-R

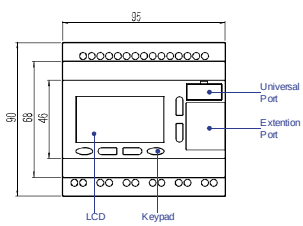


PR-18 Series

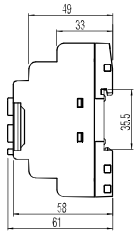
•PR-18AC-R •PR-18DC-DA-R •PR-18DC-DAT-F

(Unitmm)

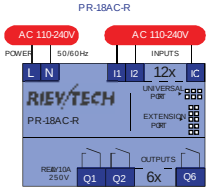
Front View



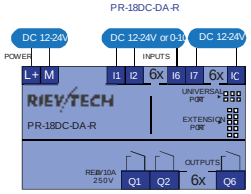
Side View



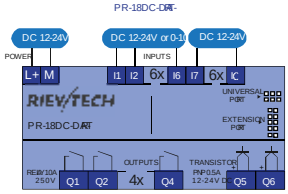
PR-18AC-R



PR-18DC-DA-R



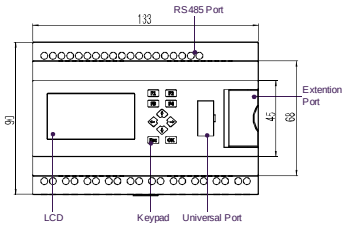
PR-18DC-DAT-F



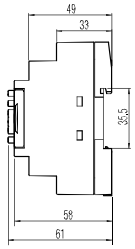
PR-24 Series •PR-24AC-R •PR-24DC-DA-R •PR-24DC-DA-TA

(Unit:mm)

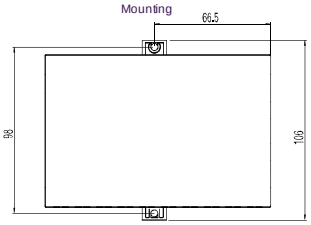
Front View



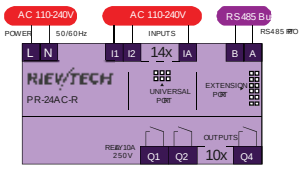
Side View



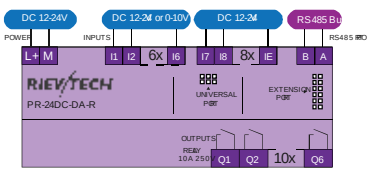
Mounting



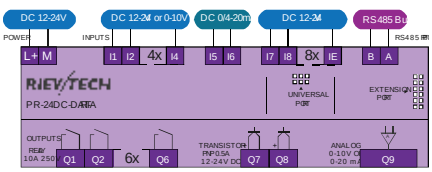
PR-24AC-R



PR-24DC-DA-R



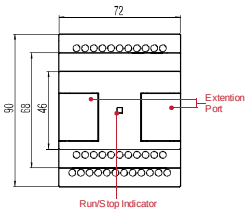
PR-24DC-DA-TA



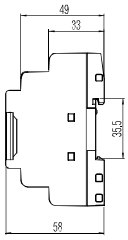
PR-E Expantion Modules •PR-E-16AC-R •PR-E-16DC-DA-R •PR-E-16DC-DA-TN •PR-E-AI-I •PR-E-PT100 •PR-E-AQ-VI •PR-E-RS485

(Unit:mm)

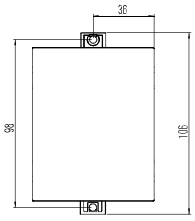
Front View



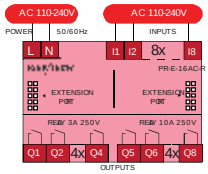
Side View



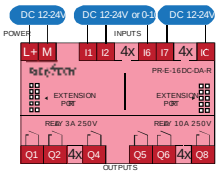
Mounting



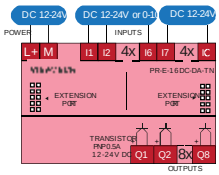
PR-E-16AC-R



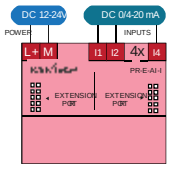
PR-E-16DC-DA-R



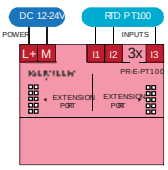
PR-E-16DC-DA-TN



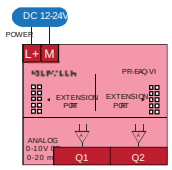
PR-E-AI-I



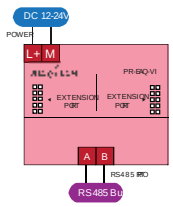
PR-E-PT100



PR-E-AQ-VI



PR-E-RS485



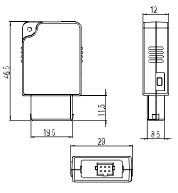
Optional Accessories •RS232 Cable •USB Cable •PRO-RS485 Cable •PR-Copier •PR-Battery •PR-Memory

(Unit:mm)

Cables

Model	Configuration	Compatibility
RS232 Cable	D-Sub connector (9-pin female) — PR-series peripheral connector — Length 2m	PR-6P PR-12P PR-14P PR-18P PR-24
USB Cable	USB type plug connector (male) — PR-series peripheral connector — Length 2m	PR-6P PR-12P PR-14P PR-18P PR-24
PRO-RS485 Cable	Terminal block connector — PR-series peripheral connector — Length 2m	PR-6P PR-12P PR-14P PR-18P PR-24

Accessories

Model	Description	Dimensions	Compatibility
PR-Copier	PR-COPIER can be used to save user program and download program into CPUs.		PR-6P PR-12P PR-14P PR-18P PR-24
PR-Battery	Battery for RTC (backup 3 years)		PR-6P PR-12P PR-14P PR-18P PR-24
PR-Memory	For Data Logging on a microSD Card		PR-6P PR-12P PR-14P PR-18P PR-24

CPU Model

PR-XXXX-XXX-XXX

- Total I/O
- Power supply: AC110-240VAC, DC12-24VDC
- Input type: N/A Only Digital, DA Digital and Analog (0-10V), DAI Digital, Analog (0-10V) and Analog (0/4-20 mA)
- Output type: R Relay, TN Transistor NPN, RT: Relay and Transistor NPN, RTA: Relay, Transistor and Analog

Analog I/O and Communication Extention Module Model

PR-E-XXXXXX-XX

- Module type: AI Analog inputs, PT100 RTD inputs PT100, AQ Analog outputs, RS485 COM Module RS485
- Analog signal type: I Current signal, VI Current and/or voltage signal

Combine I/O Extention Module Model

PR-E-XXXX-XXX-XX

- Total I/O
- Power supply: AC110-240VAC, DC12-24VDC
- Input type: N/A Only Digital, DA Digital and Analog (0-10V)
- Output type: R Relay, TN Transistor PNP

RIEV/TECH

Safety Instructions

- For your safety, please read user's manual thoroughly before operating.
- Contact the nearest authorized service facility for examination, repair, or adjustment.
- Please contact qualified service technician when you need maintenance.
  Do not disassemble or repair by yourself !
- Any maintenance and inspection shall be performed by the personnel having expertise concerned.

Rievtech Electronic Co.,Ltd

Nanjing city, Jiangsu Province, China,
211100
Tel. +86-25-52713690
Tel. +86-25-52713691
Fax +86-25-52713693
info@rievtech.com - General
sales@rievtech.com - Commercial
tech@rievtech.com – Technical

www.rievtech.com

CE FC REACH COMPLIANCE RoHS COMPLIANT UL