Zelio Logic Programming Guide

10/2017

EIO0000002612.01

www.schneider-electric.com

Schneider Electric

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

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

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

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

Table of Contents

Safety Information 11
About the Book 15
Part I Overview of the Programming Software 19
Chapter 1 Overview of the Programming Software 21
Overview of the Programming Software 22
Creating or modifying the configuration of an application 26
Part II Getting Started With the Programming Software . . . 29
Chapter 2 Getting Started With the Programming Software 31
How to Create a New Program 32
How to Program an Application Using the Programming Software . . . 33
Programming an Application from the Smart Relay Front Panel 34
How to Transfer the Program From the PC to the Smart Relay 35
How to Protect the Program Residing in the Smart Relay 36
How to debug an application without loading it onto the smart relay: Simulation 37
How to Monitor and Modify an Application Running on the Smart Relay from the Programming Software: Monitoring 43
How to Monitor and Modify an Application Using the Smart Relay Front Panel 48
What the Error Code Displayed on the Front Panel of the Smart Relay Means 49
How to Connect the Programming Software to the Smart Relay 51
How to Diagnose the Smart Relay State 52
How to Control the Smart Relay from the Programming Software . . . 53
How to Control the Smart Relay from the Front Panel 54
How to Configure an Application from the Front Panel of the Smart Relay 58
How to Modify Program Data Using the Smart Relay Front Panel . . . 59
How to Recover the Smart Relay Program from the Programming Software 60
How to Check an Application Using the Programming Software 61
How to Check the Smart Relay Firmware 65

How to Use the Memory Cartridge . . . 66
How to Configure the Language of the Programming Software and the Smart Relay . . . 67
How the Smart Relay Behaves in the Event of Power Outage . . . 68
How to import an application developed using Zelio Soft 1 into Zelio Soft 2 . . . 70
Part III Functions Accessible from the Front Panel . . . 71
Chapter 3 Overview of the Functions Accessible from the Front Panel . . . 73
Functions Accessible from the Front Panel of the Smart Relay . . . 74
Control Keys on the Front Panel of the Smart Relay . . . 76
Chapter 4 Input/Output Screen . . . 79
Inputs-Outputs Screen . . . 79
Chapter 5 PROGRAMMING Menu . . . 81
Method for Entering a Contact or Coil . . . 83
Entering a Link . . . 86
Entry of Function Block Parameters . . . 88
Deletion and Insertion of Diagram Lines . . . 89
Chapter 6 PARAMETERS Menu . . . 91
PARAMETERS Menu . . . 91
Chapter 7 MONITORING Menu . . . 93
MONITORING Menu . . . 93
Chapter 8 RUN/STOP Menu . . . 95
RUN/STOP Menu . . . 95
Chapter 9 CONFIGURATION Menu . . . 97
PASSWORD Menu . . . 98
FILTER Menu . . . 101
Zx KEYS Menu . . . 102
WATCHDOG CYCLE Menu . . . 103
Chapter 10 CLEAR PROGRAM Menu . . . 105
CLEAR PROG Menu. . . . 105
Chapter 11 TRANSFER Menu . . . 107
TRANSFER Menu . . . 107
Chapter 12 VERSION Menu . . . 111
VERSION Menu . . . 111
Chapter 13 LANGUAGE Menu . . . 113
LANGUAGE Menu . . . 113
Chapter 14 DEFAULT Menu . . . 115
DEFAULT Menu . . . 115

Chapter 15 CHANGE DATE/TIME Menu 117
CHANGE DATE/TIME Menu 117
Chapter 16 CHANGE SUMMER/WINTER Menu 119
CHANGE SUMMER/WINTER Menu 119
Part IV LD Language 121
Chapter 17 Overview of LD language 123
General Overview of Ladder Language. 124
Structure of a Ladder Network. 125
Ladder Network Comment. 127
Ladder Language Graphic Elements. 128
Programming Rules for a Ladder Network 130
Chapter 18 Programming in Ladder using Zelio Soft 2 133
Structuring LD Programs 134
Zelio Entry Mode 135
Ladder Entry Mode 136
Configuration Mode 138
Text Entry Mode 139
Chapter 19 LD Language Elements 143
Discrete Outputs 144
Discrete Inputs. 147
Modbus Inputs/Outputs 149
Auxiliary Relays 151
Zx Keys 155
Counters 157
Counter Comparators 164
Fast Counter 166
Clocks 176
Change to Summer / Winter Time 180
Timers 182
Analog Comparators 192
TEXT 198
LCD Screen Backlighting. 202
Message 203

Chapter 20	**Programming Ladder Using Zelio Soft 2**	211
20.1	Creating an LD Application in the Zelio Soft 2 programming software.	212
	Enter a Contact or a Coil	213
	Enter a Link	215
	Automation Function Configuration	216
	Insert and Delete a Program Line	218
	Copy Parts of a Program	219
	Check Program Consistency	220
20.2	Debugging an LD Application in the Programming Software	222
	Simulation of an Application	223
	Monitoring of an Application	231
Chapter 21	**Example of an LD Application**	237
	Greenhouse Ventilation Panes	237
Part V	**FBD Language**	241
Chapter 22	**Overview of FBD Language**	243
	FBD Program Edit Window	244
	Function Bar	247
Chapter 23	**FBD Language Elements**	249
23.1	Different Input Blocks	250
	Discrete Inputs	251
	Filtered Discrete Input	253
	Analog Input	254
	Filtered Analog Input	256
	Integer Input	257
	Special Inputs in FBD Language	258
	10-Bit Integer Input	260
23.2	Different Output Blocks	261
	Discrete Output	262
	Integer Output	264
	LCD Screen Backlighting Output	265
	SR3XT43BD Extension 10-Bit Integer Output	266
23.3	Modbus Inputs/Outputs	267
	Modbus Inputs-Outputs	267
23.4	Ethernet Inputs and Outputs	269
	Ethernet Inputs-Outputs	269
23.5	Logic Functions	270
	Logical functions	270

23.6 Standard Functions . . . 272
BOOLEAN Equation (Boolean function) . . . 274
SET and RESET Function . . . 275
PRESET COUNT Up/Down Counter . . . 276
H-SPEED COUNT (Fast Counter) . . . 281
UP/DOWN COUNT (Up/Down Counter) . . . 287
TIMER A/C (Timer) . . . 289
TIMER BW (Pulses on Edges) . . . 293
TIMER Li (Cyclic Timing) . . . 294
TIMER B/H (Time out) . . . 297
COMP IN ZONE Comparison . . . 300
PRESET H-METER (Preset Hour Counter) . . . 302
TRIGGER (Schmitt Trigger) . . . 303
COMP IN ZONE (Comparison of two values) . . . 305
GAIN Function . . . 306
DISPLAY (LCD Screen Display) . . . 308
TEXT . . . 311
TIME PROG (Daily, weekly, yearly programmer) . . . 314
BISTABLE (Impulse Relay) . . . 318
MUX (Multiplexing) . . . 319
ADD/SUB (ADD/SUB Arithmetic Function) . . . 320
MUL/DIV (MUL/DIV Arithmetic Function) . . . 322
CAM BLOCK (Cam Programmer) . . . 324
ARCHIVE . . . 327
STATUS . . . 329
CNA (Bit to Word Conversion) . . . 331
CAN (Word to Bit Conversion) . . . 332
SLIn (Serial Port Input) . . . 333
SLOut (Serial Port Output) . . . 335
COM (Message) . . . 337
Sunrise/Sunset . . . 344
Suntrack . . . 346
23.7 SFC Functions . . . 348
Presentation of SFC Functions . . . 349
Using the SFC Steps and Transitions . . . 352
Use of divergences to AND . . . 355
Use of Divergences to OR . . . 357
Use of convergences to AND . . . 361

Use of convergences to OR . . . 363
Use of SFC Loops . . . 367
Initialization of an SFC Chart at the Start of a Program . . . 368
Initialization of SFC Charts . . . 371
Reinitialization of an SFC Chart When Program is Running . . . 372
SFC Functions . . . 375
INIT STEP (SFC Initial Step) . . . 376
RESET INIT (Resettable Initial SFC Step) . . . 377
STEP (SFC Step) . . . 378
DIV AND 2 (Divergence to AND with 2 SFC Branches) . . . 379
CONV AND 2 (Convergence to AND with 2 SFC Branches) . . . 380
DIV OR 2 (Divergence to OR with 2 SFC Branches) . . . 381
CONV OR 2 (Convergence to OR with 2 SFC Branches) . . . 382
Errors and Advisories Detected in an SFC Chart . . . 383
23.8 Application Function Blocks . . . 384
Presentation of AFB Functions . . . 385
PID Presentation . . . 386
PID Function . . . 389
Chapter 24 Programming in FBD Using Zelio Soft 2. . . 393
24.1 Creating an FBD Application in the Zelio Soft 2 Programming Software 394
Configuring FBD Program Editing . . . 395
Inserting Function Blocks . . . 397
Creation of Links Between Function Blocks . . . 399
Function Block Parameters . . . 405
Display Options . . . 407
Draw Function . . . 408
The Find Function . . . 410
24.2 Manipulating FBD Objects . . . 411
How to Select Objects . . . 412
How to Create Composite Objects . . . 413
How to Delete and Duplicate Objects . . . 414
How to Position Objects . . . 416
How to Create or Modify a MACRO . . . 418
24.3 Debugging and Monitoring an FBD Application in the Programming Software . . . 428
Simulation Mode . . . 429
Monitoring Mode . . . 434
Modification and Forcing in Simulation and Monitoring Mode . . . 436

Chapter 25	Example of an FBD Application	439
	Greenhouse Ventilation Panes	439
Part VI	Connection With the Smart Relay	445
Chapter 26	Connection with the Programming Software	447
	Communication Setup Between the Programming Software and the Smart Relay	448
	Transfer the PC Program to the Smart Relay	450
	Transfer the Smart Relay Program to the PC	452
	RUN/STOP Program Run Commands	454
	Compare the Smart Relay Data with the Program	455
	Smart Relay Diagnostics	456
	Access/Modification Protection of the Program Saved on the Smart Relay	457
	Clear the Program Contained in the Smart Relay	458
	Set Smart Relay Clock	459
	Configuring the Smart Relay Language	460
	Update Smart Relay Firmware	461
	Remote Control of Front Panel	462
Chapter 27	Communication Via Modbus Extension	463
	Communication via the Modbus Extension	463
Chapter 28	SR2COM01 Communication Interface	467
	Zelio2 COM Menu	468
	Directories Menu	470
	Configuring the SR2COM01 Communication Interface	476
	Sending an Email via SMS	482
	Description of the Error Codes of the SR2COM01 Communication Interface	483
Chapter 29	Analog Input-Output Extension SR3XT43BD	485
	Analog Input-Output Extension SR3XT43BD	485
Chapter 30	Communication Via Ethernet Extension	487
	At a Glance	488
	Acquiring IP Addresses	491
	Communication on an Ethernet Network	494
	Requests Specific to TCP Diagnostics	499
Part VII	Programming Software Functions	501
Chapter 31	Functions	503
	Program Configuration	504
	Preferences of the Programming Software	508
	Program Check	509

Write Options Window 510
Program Import 512
Conversion of Applications Created with Previous Version of the Software 514
Setting the Clock Display 515
Saving an Application. 516
Printing the Program 517
Page Header and Footer for Application Printing. 519
Description of Smart Relay Errors 521
Splitting the Wiring Sheet. 523
Chapter 32 Description of the Programming Software Menus 527
Description of the Programming Software Menus 527
Appendices 535
Appendix A Compatibility. 537
Compatibility between the version of Zelio Soft 2 software and the version of the firmware on the smart relay 538
Zelio Soft 2 Software Version Versus Functions 540
Compatibility between the memory cartridges and the version of the firmware on the smart relay 541
Glossary 543
Index 547

Safety Information

Important Information

NOTICE

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

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

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

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

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

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

NOTICE
NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

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

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

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

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

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

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

⚠ WARNING
EQUIPMENT OPERATION HAZARD ● Verify that all installation and set up procedures have been completed. ● Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices. ● Remove tools, meters, and debris from equipment. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:
- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book

At a Glance

Document Scope

This manual describes how to use the programming software Zelio Soft 2.

Validity Note

This document has been updated for the release of Zelio Soft 2 V5.1.

The technical characteristics of the devices described in this document also appear online. To access this information online:

Step	Action
1	Go to the Schneider Electric home page *www.schneider-electric.com*.
2	In the **Search** box type the reference of a product or the name of a product range. ● Do not include blank spaces in the reference or product range. ● To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the **Product Datasheets** search results and click on the reference that interests you. If you entered the name of a product range, go to the **Product Ranges** search results and click on the product range that interests you.
4	If more than one reference appears in the **Products** search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the data sheet.
6	To save or print a data sheet as a .pdf file, click **Download XXX product datasheet**.

The characteristics that are presented in this manual should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the manual and online information, use the online information as your reference.

For product compliance and environmental information (RoHS, REACH, PEP, EOLI, etc.), go to *www.schneider-electric.com/green-premium*.

Product Related Information

⚠ WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.[1]
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

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

[1] For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

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

Care must be taken and provisions made for use of the modem functionality as a remote control device to avoid inadvertent consequences of commanded machine operation, smart relay state changes, or alteration of data memory or machine operating parameters.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

- Ensure that there is a local, competent, and qualified observer present when operating from a remote location.
- Configure and install a means of local control over the starting or stopping of the smart relay such that it can be maintained regardless of the remote commands sent to the smart relay.

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

Related Documents

Document title	Reference
Zelio Logic 2 Smart Relay User Manual	*EIO0000002690 (ENG)* *EIO0000002692 (GER)* *EIO0000002691 (FRE)* *EIO0000002693 (SPA)* *EIO0000002694 (ITA)* *EIO0000002695 (POR)*
Zelio Soft 2 Applications Example Guide	*EIO0000002600 (ENG)* *EIO0000002602 (GER)* *EIO0000002601 (FRE)* *EIO0000002603 (SPA)* *EIO0000002604 (ITA)* *EIO0000002605 (POR)*
SR2A••••• / SR2B••••• Instruction Sheet	*1724026_01A55*
SR2D••••• / SR2E••••• Instruction Sheet	*1724028_01A55*
SR3B••••• Instruction Sheet	*1724027_01A55*

You can download these technical publications and other technical information from our website at *http://www.schneider-electric.com/en/download*.

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
EN 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2008	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements

Standard	Description
EN 1088:2008 ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2006	Safety of machinery - Emergency stop - Principles for design
EN/IEC 62061:2005	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2008	Digital data communication for measurement and control: Functional safety field buses.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive* (*2006/42/EC*) and *ISO 12100:2010*.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Part I
Overview of the Programming Software

Chapter 1
Overview of the Programming Software

Subject of this Chapter

This chapter introduces the programming software.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Overview of the Programming Software	22
Creating or modifying the configuration of an application	26

Overview of the Programming Software

Overview

Zelio Soft 2 Programming Software is designed to program the Zelio Logic Smart Relay family of controllers. Zelio Soft 2 allows you to choose between programming languages, display program and parameter data, upload and download applications, and print application documentation.

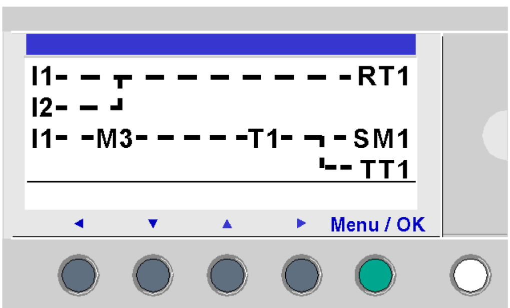
Programming Modes

Smart relays with screen:

There are two ways to begin programming smart relays with screen:

- From the smart relays front panel *(see page 71)*
  This approach is designed for those with experience in programming directly on smart relays.
  NOTE: Only possible in **LD mode**

Illustration:

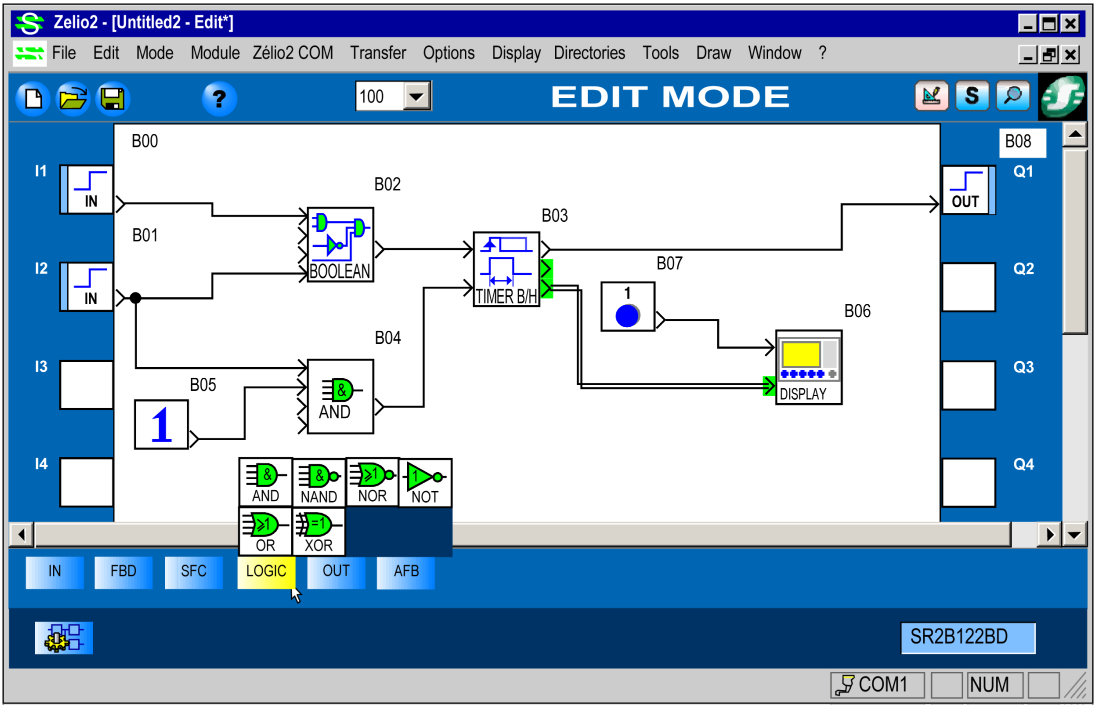


- From the programming software *(see page 211)*

Smart relays without screen:

Smart relays without screen can only be programmed from the programming software.

Main window of the programming software:



Languages Used

The smart relay provides 2 programming modes:

- LD mode: Ladder language
- FBD mode: Function Block Diagram mode

These languages implement:

- Predefined function blocks:
  - Timers,
  - Counters.
- Specific functions:
  - Time management,
  - Character string,
  - Communication, etc.

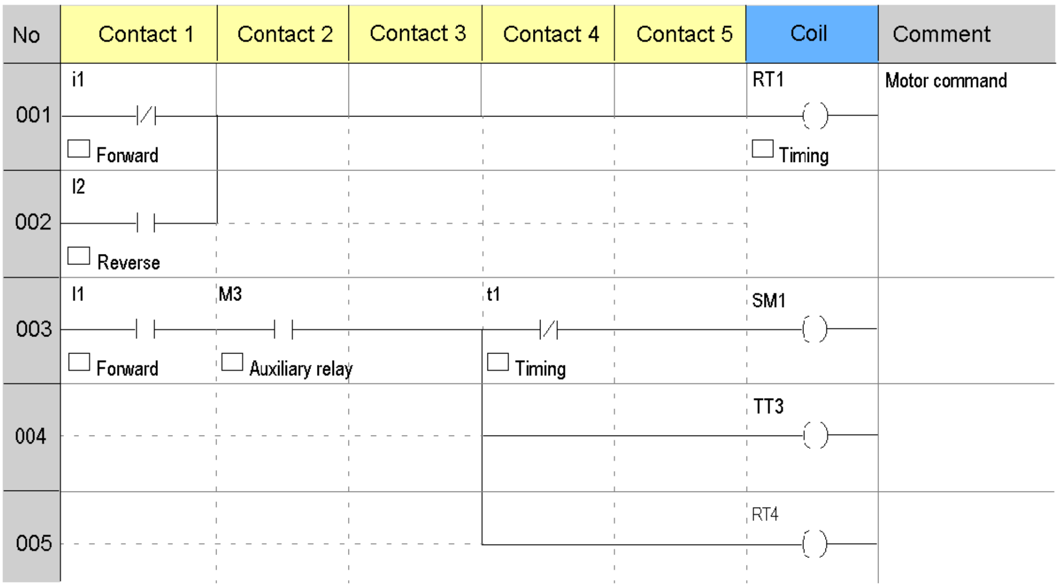
Ladder language

Ladder language (LD) is a graphic language. It can be used to transcribe relay diagrams, and is suited to combined processing.

It provides graphic symbols: contacts, coils, blocks.

Specific calculations can be executed within the operate blocks.

The following is an example of a program in ladder language within Zelio Logic 2:



The maximum number of lines in Ladder language is either:

- 120 lines, if an SR2COM01 communication interface has been selected in the configuration,
- 240 lines, without an SR2COM01 communication interface.

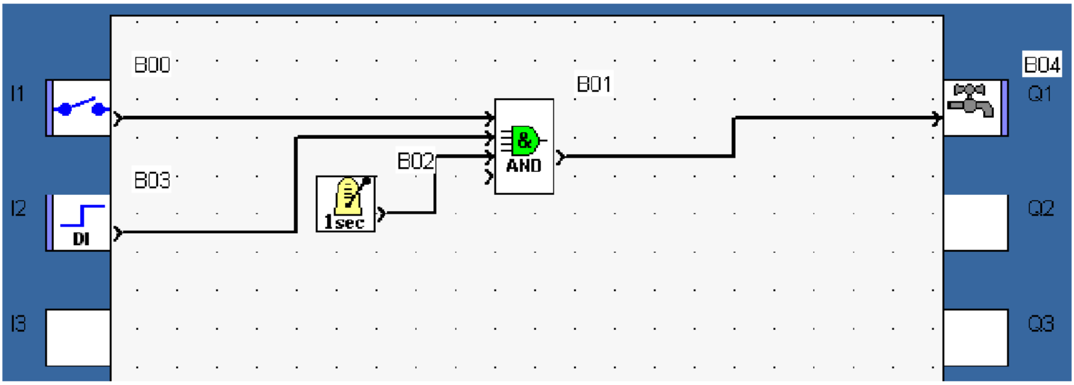
NOTE: The maximum number of program lines also depends on the firmware version *(see page 540)*.

FBD mode

FBD mode allows graphic programming based on the use of predefined function blocks.

It offers a large range of functions: timer, counter, logic, etc.

Example of a program in FBD.



Operating mode

There are two operating modes for the programming software:

- **Entry mode**
  Entry mode is used to construct programs in LD or FBD mode, which corresponds to the development of the application.
- **Debug mode**
  This mode is used to finalize the application, which may be performed:
  - In Simulation mode: The program is executed offline directly in the programming software (simulated on the PC).
    In this mode, each action on the chart (changing the state of an input, output forcing) updates the simulation windows.
  - In Monitoring mode: The program is executed on the smart relay; the programming software is connected to the smart relay.
    The different windows are updated cyclically.

  In these two modes, it is possible to:
  - Display dynamically (in the windows: Edit / Supervision / Front Panel), the output states and program function blocks corresponding to the wiring sheet.
  - Force the inputs/outputs to test program behavior under specific conditions.

Creating or modifying the configuration of an application

Description

This is an important phase, as it determines the configuration of the work environment.

The available functions depend on:

- The hardware configuration (module/extension),
- The type of programming language selected (LD/FBD).

These choices enable configuration of the programming software:

- LD/FBD context for the menus,
- Composition of the windows.

Module Selection/Programming Option

Within the **Module** menu, the **Module Selection/Programming** option is used to modify the module and/or extension type when an application is open in edit mode. If you also modify the programming type of an application open in edit mode, then the open application must be closed and a new application created (user program).

This option displays a suite of three pages within a dialog box:

- The first is used to choose the type of module,
- The second is used to add an extension if necessary,
- The third page is used, where applicable, to select the programming type used for the new FBD or Ladder LD application, if the module is compatible with the 2 types of programming.

The procedure is exactly the same as the procedure used to create a new application and is detailed below (see *Create an application, page 26*).

Create an application

Procedure for creating an application:

Step	Action
1	Select the **File → New** or click on the **Create New Program** icon when launching Zelio Soft 2. **Result**: The **Module Selection** window appears.
2	In the **Select Your Module Category** zone, select the category by clicking on the corresponding checkbox. The modules are grouped by categories corresponding to: ● The number of inputs/outputs, ● The presence or absence of an operator display, ● Whether or not it is possible to connect extensions. **Result:** The list of corresponding modules appears in the **Select the Type of Zelio Module to Program** zone.

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Select the module by clicking on the corresponding line then confirm using the Next >button.
Result: Three following possibilities arise at this stage:
The module does not support extensions and programming in LD mode only:
Go to step 7.
The module does not support extensions and programming in available LD and FBD modes:
Go to step 6.
If the module supports extensions:
Result: 2 new zones appear in the window:
Module selection: summarizing the choices made in steps 2 and 3;
Select extensions: listing the compatible extensions.</td></tr>
<tr><td>4</td><td>In the Select extensions zone, select the extension type to be added in the Compatible extensions list by double-clicking on the corresponding line or by using the Add button.
Result: the selected extension appears in the Selected extensions list.
The extension can be removed from the Selected extensions list by clicking on it then using the Delete button.

NOTE: You may then add a single input/output extension and/or a single bus type extension.</td></tr>
<tr><td>5</td><td>Validate the configuration by clicking on the Next> button.
Result: The program type selection zone appears: Select the programming type.</td></tr>
<tr><td>6</td><td>By default, the programming type is LD. To choose the FBD programming type, click on the associated graphic.
Click on the NEXT> button to confirm.</td></tr>
<tr><td>7</td><td>The edit window appears with a blank wiring sheet.
For an application in FBD mode
With the module type and any extension selected serving as the context, there is a drawing background displayed in the Edit window with specific I/Os arranged about its periphery and a specific set of FBD functions presented in the Tool bar. The names of the module and extensions are displayed below the wiring sheet.
For an application in LD mode
With the module type and any extension selected serving as the context, there is a specific set and number of LD functions presented in the Tool bar. The names of the module and extensions are displayed below the wiring sheet.</td></tr>
</table>

Modify the configuration of an application

Modification procedure for the configuration of an application:

Step	Action
1	Click on the **Module** → **Module Selection/Programming** menu. **Result**: The summary and choose programming type window appears on the screen.
2	Modify the parameters to obtain the configuration required. To: ● Modify the module type: Click two times on the **Previous** button, then proceed in the same manner as for the creation of an application, ● Add, modify or delete an extension: Click two times on the **Previous** button, then proceed in the same manner as for the creation of an application, ● Modify the programming type: Click on the illustration representing the type of programming desired. **NOTE:** If you only want to change the programming type, the previously programmed instructions, if any, will be erased so that you can create the application in the chosen programing type. **NOTE:** No SR2COM01 extension can be added if the Ladder program exceeds 120 lines.
3	Continue, if necessary, to the summary page and the choice of programming type by clicking on **Next**.
4	Confirm the changes by clicking on **Next**. **Result**: An empty wiring sheet is displayed on the page.

Part II
Getting Started With the Programming Software

Chapter 2
Getting Started With the Programming Software

Subject of this Chapter

This chapter explains how to use the Programming Software.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
How to Create a New Program	32
How to Program an Application Using the Programming Software	33
Programming an Application from the Smart Relay Front Panel	34
How to Transfer the Program From the PC to the Smart Relay	35
How to Protect the Program Residing in the Smart Relay	36
How to debug an application without loading it onto the smart relay: Simulation	37
How to Monitor and Modify an Application Running on the Smart Relay from the Programming Software: Monitoring	43
How to Monitor and Modify an Application Using the Smart Relay Front Panel	48
What the Error Code Displayed on the Front Panel of the Smart Relay Means	49
How to Connect the Programming Software to the Smart Relay	51
How to Diagnose the Smart Relay State	52
How to Control the Smart Relay from the Programming Software	53
How to Control the Smart Relay from the Front Panel	54
How to Configure an Application from the Front Panel of the Smart Relay	58
How to Modify Program Data Using the Smart Relay Front Panel	59
How to Recover the Smart Relay Program from the Programming Software	60
How to Check an Application Using the Programming Software	61
How to Check the Smart Relay Firmware	65
How to Use the Memory Cartridge	66
How to Configure the Language of the Programming Software and the Smart Relay	67
How the Smart Relay Behaves in the Event of Power Outage	68
How to import an application developed using Zelio Soft 1 into Zelio Soft 2	70

How to Create a New Program

Description

See Creating an Application *(see page 26)*.

How to Program an Application Using the Programming Software

Description

See LD Programming from the Programming Software *(see page 211)*.

See FBD Programming from the Programming Software *(see page 393)*.

Programming an Application from the Smart Relay Front Panel

Description

See PROGRAMMING Menu *(see page 81)*

How to Transfer the Program From the PC to the Smart Relay

Description

See Transferring the Program from the PC to the Smart Relay *(see page 450)*

How to Protect the Program Residing in the Smart Relay

Description

See Protection of the Program Saved on the Smart Relay *(see page 457)*

How to debug an application without loading it onto the smart relay: Simulation

Description

To help ensure that a program will perform as expected before loading it onto a smart relay, it is possible to simulate execution of the program using the Zelio Soft 2 programming software. This simulation allows you to:

- Temporarily modify or to force any FBD function output, any LD contact, the majority of function parameters as well as any of the keys on the smart relay front panel,
- View the effect of each modification or forcing on the execution of the program by observing the values of the FBD block outputs, LD contacts and coils as well as the displays on the simulated smart relay front panel.

How to Execute a Program in Simulation Mode

After having created a diagram in the wiring sheet or using **Zelio entry** in LD mode, click on the **Simulator**  icon in the toolbar.

To terminate the simulation and return to Edit mode, click on the **Edit**  icon.

After clicking on the **Simulator**  icon, the **SIMULATION MODE** toolbar and a set of icons representing the functions available in simulation mode are displayed. All or some of the following windows can be displayed:

- Using the **Window** menu:
  - The **Edit** window,
  - The **Supervision** window,
  - The Smart Relay Front Panel Simulation window.
- Using the icons in the bar at the bottom of the window:
  - The Simulation time window,
  - The Function blocks window (with application in LD mode only),
  - The Discrete inputs window (with application in LD mode only),
  - The Zx keys window (with application in LD mode only),
  - The Discrete outputs window (with application in LD mode only).

Each action that the user performs on the chart corresponds to a simulation the results of which are displayed in the windows.

The Smart Relay Front Panel Simulation Window

This window allows you to use the mouse to click any of the keys on the smart relay front panel which is depicted in the window.

The keys in the Front Panel window can be controlled as if they were those on the physical front panel of the smart relay. The functions which can be accessed from the front panel can be used with a mouse click.

The result of these actions is then displayed in the simulation on the LCD screen.

Though they are accessible, certain functions are inoperative as they are not meaningful in a simulation environment:

- FILTER,
- WATCHDOG CYCLE,
- TRANSFER.

The Edit Window

Display in LD and FBD:

- Displays programs written on an FBD or LD wiring sheet,
- Shows discrete FBD links, contacts, coils and LD links which are OFF in inactive color (blue by default),
- Shows discrete FBD links, contacts, coils and LD links which are ON in active color (red by default). Active and non-supplied contacts and coils are displayed in orange,
- Shows each active step of an SFC chart in active color (red by default),
- Shows the value of each digital link of an FBD chart,
- Animates the LD contacts and FBD functions that have only one Discrete output, according to the status of its Discrete output,
- Shows the value of the parameters of the FBD functions, by double-clicking on the function block,
- Shows the value of the parameters of the LD functions, by right-clicking with the mouse on each contact or coil and then selecting **Parameters** window in the menu that is displayed.

Forced values are highlighted in the **Edit** and **Supervision** windows by a change in the background color on which they are displayed.

LD Actions:

- Can be used to temporarily modify the state of any LD chart contact, by clicking on it with the mouse (change from ON/OFF),
- Can be used to permanently force the state of any LD chart contact, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering ON or OFF in the **Permanent forcing** window, and then confirming the selections by pressing **OK**,
- Can be used to modify the value of a subset of LD function parameters, by right-clicking on each contact or coil with the mouse, then selecting **Parameters** window in the menu displayed, modifying one or more of the non-grayed out parameters and confirming the selections by pressing **OK**,

- Can be used to release the state of any LD chart contact by right-clicking on it with the mouse and selecting **Release** in the menu displayed,
- Can be used to release the forced outputs or links, by right-clicking with the mouse in the window and selecting **Release all** in the menu displayed.

FBD Actions:

- Can be used to temporarily modify the state of any Discrete or Token output or link of an FBD chart, by clicking on it with the mouse (change from ON/OFF),
- Can be used to temporarily modify the state of any FBD chart output or digital link, by clicking on it with the mouse, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any Discrete or Token output or link of an FBD chart, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering ON or OFF in the **Permanent forcing** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any digital link output of an FBD chart, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to modify the value of a subset of FBD function parameters, by double-clicking on the function block, modifying one or more of the non-grayed out parameters and confirming the selections by pressing **OK**,
- Can be used to release the state of a forced output or link by right-clicking on it with the mouse and selecting **Release** in the menu displayed,
- Can be used to release the forced outputs or links by right-clicking in the window with the mouse and selecting **Release all** in the menu displayed.

Supervision Window

View:

- Displays the LD or FBD edit functions selected in this window as FBD function blocks,
- Shows the discrete FBD function block outputs which are OFF in inactive color (blue by default),
- Shows the discrete FBD function block outputs which are ON in active color (red by default),
- Shows each active step of an SFC chart in active color (red by default),
- Shows the value of each digital output of an FBD function block,
- Animates the FBD function blocks that have only one Discrete output, according to the status of its Discrete output,
- Shows the value of the FBD function block parameters, by double-clicking on the function block with the mouse or right-clicking on each contact or coil with the mouse, and then selecting **Parameters** window in the menu that is displayed.

Forced values are highlighted in the **Edit** and **Supervision** windows by a change in background color.

Actions:

- Can be used to temporarily modify the state of any Discrete or Token output of an FBD function block, by clicking on it with the mouse (change from ON/OFF),
- Can be used to temporarily modify the state of any FBD function block output or digital link, by clicking on it with the mouse, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any Discrete or Token output of an FBD function block, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering ON or OFF in the **Permanent forcing** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any digital output of an FBD function block, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to modify the value of a subset of FBD function block parameters, by double-clicking on the function block with the mouse, modifying one or more non-grayed out parameters, then confirming the selections by pressing **OK**. This action can also be performed by right-clicking each contact or coil with the mouse, then selecting **Parameters** window in the menu displayed, modifying one or more non-grayed out parameters, then confirming the selections by pressing **OK**.
- Can be used to release a forced output by right-clicking on it with the mouse and selecting **Release** in the menu displayed,
- Can be used to release the forced outputs by right-clicking in the window with the mouse and selecting **Release all** in the menu displayed.

Acceleration and Simulation Limits Window

This window is displayed when you click on the simulation time smart relay icon  situated in the bar at the bottom of the simulation window.

View:

- Displays the date and time of the start and end of the simulation.

Action:

- Can be used to modify the date and time of the start and end of the simulation in the **Simulation Limits** window.
- Can be used to accelerate the simulation speed up to 65000 times the original speed by clicking the **>>** key and moving the level of the **min-max** bar.

Function Block Windows

Function Block Windows: Discrete input, discrete outputs and **Zx** keys are accessible only in LD mode.

They are displayed when you click on one of the icons situated in the bar at the bottom of the simulation window.

View:

- Function blocks summarize in table format the function blocks with analog parameters or inputs and their changes,
- The other windows display the state of the inputs, physical outputs and smart relay keys.

Action:

- Can be used to modify a parameter by double-clicking on the box that represents its value in the function blocks window, which triggers display of the function's parameters window,
- Can be used to modify the ON or OFF value of an input or a **Zx** key in the inputs and **Z** keys window, by clicking on its drawing.

Functions not Accessible in Simulation Mode:

In simulation mode, the following functions are not available:

- Graphic editing of the program,
- Reading, writing, comparing and clearing the smart relay program,
- Monitoring,
- Modifying communication parameters,
- Modifying the configuration parameters of the program.

Simulation Mode Toolbar

The simulation bar is used to modify simulation rates or to simulate certain events affecting the smart relay.

NOTE: To display the functions described below, check the box in the **File → Preferences** menu labeled **Show the refresh cycle** and specify the number of cycles for monitoring and simulation. The functions described below are required to carry out a simulation capable of highlighting transient anomalies, in particular upon startup of the application and when power is restored following a power outage.

As the execution of the application on the smart relay is periodic and controllable by a WATCHDOG (**Edit → Program Configuration → Configuration**), the programming software can call the simulator periodically and require that the simulator use a time base that increments the number of milliseconds corresponding to the application execution period (application basic cycle).

This time base will set the rate not only for execution of the functions that depend explicitly or implicitly on time (timers, Filtered FBD inputs, looped FBD functions, etc.) but also changes in the clock and the simulator date on which Clock, Daily programmer and the Summer/Winter time functions depend.

Thus, to highlight the transient anomalies introduced by the program, the program must be executed step by step. The Number of cycles must be set to 1, which will trigger a duration between 2 simulation results equal to the execution period of the application on the smart relay.

On the other hand, to provide a succinct explanation of the application operation, you can raise the number of cycles to 255. If this is not sufficient, then the **Acceleration and simulation limits** window can be used to multiply this duration up to 65000, or at least 46 hours between 2 simulation results.

Other icons and windows:

- Stop, Pause and Power outage,
- Run,
- refresh frequency of the PC windows (modifiable values).

A gray button cannot be used; a colored button can be activated by clicking on it once; a yellow or red button indicates a stop in the simulation (pause) or a stop in the simulated smart relay (stop or power outage). A green button indicates a program whose simulation is in progress (Run).

When you click on **Run**, the switch from **Stop** red to **Run** green triggers initialization of the program and startup of program execution simulation.

When you click on **Stop**, the change from **Run** green to **Stop** red stops the program execution simulation.

The **Pause** button in the simulation bar can be used to stop and restart program execution. This button can only be used in simulation mode.

The **Mains Power Failure** (power outage simulation) button in the simulation bar can be used to stop and restart a warm restart initialization and then program execution. This initialization, which takes into account the state of latching parameters, is only executed on the smart relay when a power outage occurs. This button can only be used in simulation mode.

The refresh frequency corresponds to the frequency at which the output and parameter values are updated in the application windows that are open during simulation mode. The update of this set of values is considered to be the simulation result. The refresh frequency can be used, in the absence of the step counter function, to slowly display the transient changes of the simulated application.

The integer Number of cycles is the number of cycles executed between each simulation result. A number of cycles equal to 1 signifies that the modifications to the input output states displayed correspond to period by period execution (application execution period defined in the configuration) of the smart relay chosen. Management of the date and time is aligned on the number of cycles executed between each simulation result.

If you select a number of cycles that is greater than 1 for each refresh of the simulation results, you might not observe the changes in the inputs and outputs of functions terminating in less time than the cycle time multiplied by the number of cycles for each refresh.

How to Monitor and Modify an Application Running on the Smart Relay from the Programming Software: Monitoring

Description

To monitor or modify the behavior of a program running on a smart relay, you can use the monitoring function. This monitoring allows you to:

- Temporarily modify or permanently force any FBD function output, any LD contact, the majority of function parameters as well as any of the buttons on the smart relay front panel,
- Periodically display program execution showing input/output values of the smart relay, of any extensions and FDB block outputs, LD coils and contacts, and the state of the parameters and the front panel displays of the connected smart relay.

These modifications and display are carried out:

- On the one hand in the programming software windows, which can be accessed:
  - Using the menu **Edit** or **Supervision** on the smart relay front panel,
  - Using the icons of the bar at the bottom of the monitoring window (LD only): Function blocks, Discrete inputs, Zx keys, Discrete outputs.

  The programming software windows display the state of the smart relay inputs and outputs, as well as those of its possible extensions, the states of the program, the FBD function block parameters (including output parameters), the states of the LD contacts and coils and the parameters of the LD diagram corresponding to the program running on the smart relay.
- On the other hand, in the monitoring mode toolbar, where a set of icons can be used to start and stop application execution in the smart relay and the frequency at which output values and parameters are updated in the open application windows.

The obtained value of each link is displayed near the function block output. **Monitoring** mode is independent of the **Run/Stop module** function. If the smart relay is stopped, only modifications to the parameters and the outputs of the buttons on the smart relay front panel are displayed.

NOTE: The values and states displayed in **Monitoring** mode represent those values and states read at the time of the defined refresh period.

Furthermore, when the connected smart relay switches to **Monitoring** mode, the application execution periods (defined in Program configuration) are extended by the communication time between the PC and the smart relay, as does any possible permanent forcing applied to the application. When applications without permanent forcing are executed, the application may run on the smart relay for a time that is much shorter than the refresh period for the PC monitoring windows. The actions observed on the smart relay are not less than twice the monitoring refresh period. Therefore, the real duration of execution periods may vary greatly during this operating mode.

In addition, during the **Monitoring** mode, the WATCHDOG action (defined in WATCHDOG Program Configuration) is deactivated.

⚠ CAUTION
EQUIPMENT OPERATION HAZARD ● Perform real machine or process operational tests before placing this equipment into service. ● Do not rely solely on simulation testing results for the debugging and/or commissioning of your machine or process. **Failure to follow these instructions can result in injury or equipment damage.**

Switch to Monitoring Mode in the Smart Relay and Programming Software

You can only switch to this mode if the smart relay:

- contains a program that is not read/write protected with a password,
- contains a program that is read/write protected with a password known by you,
- contains a program that is identical to the program open in the programming software.

The programming software determines whether a password protects access to a program. If this is the case, the programming software displays the **Password** dialog window.

The chart in the **Edit** window must be in accordance with the smart relay program. The programming software launches the **Compare the Smart Relay Data with the Program** function. If a difference exists, the programming software returns to edit mode without establishing a connection with the smart relay.

Following these validations, click on the  button in the toolbar to switch to monitoring mode.

Monitoring Window of the Front Panel of the Smart Relay

This window allows you to use the mouse to click any of the keys on the smart relay front panel which is depicted in the window. The keys in the Front Panel window can be controlled as if they were those on the physical front panel of the smart relay. The functions which can be accessed from the front panel can be used with a mouse click. The result of these actions is then displayed in the copy on the LCD screen.

The Edit Window

Display:

- Displays programs written on an FBD or LD chart,
- Shows discrete FBD links, contacts, coils and LD links which are OFF in inactive color (blue by default),
- Shows discrete FBD links, contacts, coils and LD links which are ON in active color (red by default). Active and non-supplied contacts and coils are displayed in orange,
- Shows each active step of an SFC chart in active color (red by default),

- Shows the value of each digital link of an FBD chart,
- Animates the LD contacts and FBD functions that have only one Discrete output, according to the status of its Discrete output,
- Shows the value of the parameters of the FBD functions, by double-clicking on the function block,
- Shows the value of the parameters of the LD functions, by right-clicking with the mouse on each contact or coil and then selecting **Parameters** window in the menu that is displayed.

Forced values are highlighted in the **Edit** and **Supervision** windows by a change in background color.

FBD Actions:

- Can be used to temporarily modify the state of any Discrete or Token output or link of an FBD chart, by clicking on it with the mouse (change from ON/OFF),
- Can be used to temporarily modify the state of any FBD chart output or digital link, by clicking on it with the mouse, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any Discrete or Token link or output of an FBD chart, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering ON or OFF in the **Permanent forcing** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any digital link output of an FBD chart, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to modify the value of a subset of FBD function parameters, by double-clicking on the function block, modifying one or more of the non-grayed out parameters and confirming the selections by pressing **OK**,
- Can be used to release a forced output or link by right-clicking on it with the mouse and selecting **Release** in the menu displayed,
- Can be used to release the forced outputs or links by right-clicking in the window with the mouse and selecting **Release all** in the menu displayed.

LD Actions:

- Can be used to temporarily modify the state of any LD chart contact, by clicking on it with the mouse (change from ON/OFF),
- Can be used to permanently force the state of any LD chart contact, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering ON or OFF in the **Permanent forcing** window, and then confirming the selections by pressing **OK**,
- Can be used to modify the value of a subset of LD function parameters, by right-clicking on each contact or coil with the mouse, then selecting **Parameters** window in the menu displayed, modifying one or more of the non-grayed out parameters and confirming the selections by pressing **OK**,

- Can be used to release the state of any LD chart contact by right-clicking on it with the mouse and selecting **Release** in the menu displayed,
- Can be used to release the forced outputs or links by right-clicking in the window with the mouse and selecting **Release all** in the menu displayed.

Supervision Window

Display:

- Displays the LD or FBD edit functions selected in this window as FBD function blocks,
- Shows the Discrete FBD function block outputs which are OFF in inactive color (blue by default),
- Shows All or None FDB functional blocks that are ON in active color (red by default) (discrete outputs and active and non-powered FBD blocks are displayed in orange,
- Shows each active step of an SFC chart in active color (red by default),
- Shows the value of each digital output of an FBD function block,
- Animates the FBD function blocks that have only one Discrete output, according to the status of its Discrete output,
- Shows the value of the FBD function block parameters, by double-clicking on the function block with the mouse or right-clicking on each contact or coil with the mouse, and then selecting **Parameters** window in the menu that is displayed.

Forced values are highlighted in the Edit and Supervision windows by a change in background color.

Actions:

- Can be used to temporarily modify the state of any Discrete or Token output or link of an FBD function block, by clicking on it with the mouse (change from ON/OFF),
- Can be used to temporarily modify the state of any FBD function block output or digital link, by clicking on it with the mouse, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any Discrete or Token output of an FBD function block, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering ON or OFF in the **Permanent forcing** window, and then confirming the selections by pressing **OK**,
- Can be used to permanently force the state of any digital output of an FBD function block, by right-clicking on it with the mouse, selecting **Force and maintain** in the menu displayed, entering a signed whole value in the **Analog value** window, and then confirming the selections by pressing **OK**,
- Can be used to modify the value of a subset of FBD function block parameters, by double-clicking on the function block with the mouse, modifying one or more non-grayed out parameters, then confirming the selections by pressing **OK**. This action can also be performed by right-clicking each contact or coil with the mouse, then selecting **Parameters** window in the menu displayed, modifying one or more non-grayed out parameters, then confirming the selections by pressing **OK**.

- Can be used to release a forced output by right-clicking on it with the mouse and selecting **Release** in the menu displayed,
- Can be used to release the forced outputs by right-clicking in the window with the mouse and selecting **Release all** in the menu displayed.

Function Blocks Windows

Function Blocks Windows: Discrete input, discrete outputs and Zx keys are accessible only in LD mode.

They are displayed when you click on one of the icons situated in the bar at the bottom of the monitoring window.

Display:

- Function blocks summarize in table format the function blocks with analog parameters or inputs and show their changes,
- The other windows display the state of the inputs, physical outputs and smart relay keys.

Action:

- Can be used to modify a parameter by double-clicking on the box that represents its value in the function blocks window, which triggers display of the function's parameters window,
- Can be used to modify the ON or OFF value of an input or a Zx key in the inputs and Z keys window, by clicking on its drawing.

Functions not Accessible in Monitoring Mode

In monitoring mode, the following functions are not available:

- Graphic editing of the program,
- Reading, writing, comparing and clearing the smart relay program,
- Modifying communication parameters,
- Modifying the configuration parameters of the program.

Monitoring Mode Toolbar

The refresh frequency corresponds to the frequency at which the output and parameter values are updated in the application windows that are open during monitoring mode. Reducing this frequency, and consequently, the refresh period, reduces the programming software workload.

The commands that can be used to control monitoring are:

- The **Stop** button,
- The **Run** button,
- The time between 2 displays of smart relay data on the screen (modifiable value).

NOTE: To display the refresh frequency, check the following box in the **File → Preferences** menu: **Show the refresh cycle (simulation and monitoring) and the time between two simulation results**.

How to Monitor and Modify an Application Using the Smart Relay Front Panel

Description

See MONITORING Menu *(see page 93)*

What the Error Code Displayed on the Front Panel of the Smart Relay Means

Description

You can display on the front panel the error or advisory codes detected by the smart relay (WATCHDOG overflow *(see page 103)*, cycle time too long *(see page 505)*, etc.) using the **DEFAULT Menu** *(see page 115)*.

Possible Errors

List of errors:

Code	Type of error
00	**No error**
01	**Error in writing to non-volatile memory** This error defines transfer problems between the memory cartridge and the smart relay. If the error occurs frequently, contact your local Schneider Electric support representative.
02	**Error in writing to the clock** If the error occurs frequently, contact your local Schneider Electric support representative.
04	**Overload on transistor outputs** Once a transistor output reaches the threshold for over-current detection, the group of 4 outputs to which it belongs is deactivated. To make this group of outputs operational, the cause of the over current (short-circuit, etc.) must first be rectified, and then the error cleared from the DEFAULT menu *(see page 115)*.
50	**Smart relay firmware is corrupted,** see Update Smart Relay Firmware *(see page 461)*. Reload the firmware on the smart relay, followed by transferring the application program. If this error persists, contact your local Schneider Electric support representative.
51	**Watchdog overflow** Advisory or error according to the selection made in the configuration menu (smart relay display) or in the configuration window (programming software). The cycle time in the smart relay is too short compared with the application program execution time programmed in the smart relay. If the application requires cycle time or strict sampling of the smart relay inputs/outputs, lengthen the application cycle time in the smart relay. To do this, either set the parameters in the **CONFIGURATION** menu (smart relay display) or in the configuration window (programming software). If the application does not require a maximum cycle time, select: **No Action for the WATCHDOG**.

Code	Type of error
52	**The smart relay has executed an incorrect operation** If the error is permanent, reload the firmware on the smart relay and the user application. If this error persists, contact your local Schneider Electric support representative.
53	**Link error between smart relay and bus-type extension** Verify operation of the extension (connection, power supply and error status).
54	**Link error between smart relay and input/output-type extension** Verify operation of the extension (connection, power supply and error status).
58	**An error is present in the firmware or on a part of the smart relay hardware.** If the error is permanent, reload the firmware on the smart relay and the program. If this error persists, contact your local Schneider Electric support representative.
59	**At the beginning of RUN on the smart relay application: The application cannot switch to RUN as it is incompatible with the smart relay physically connected to the power supply.** If this error occurs, contact your local Schneider Electric support representative.
60	**At the beginning of RUN on the smart relay application: program incompatible with the bus extension physically connected to the power supply.** If this error occurs, contact your local Schneider Electric support representative.
61	**At the beginning of RUN on the smart relay application: program incompatible with the Input/Output extension physically connected to the power supply.** If this error occurs, contact your local Schneider Electric support representative.
62	**Version (or build number) incompatibility when loading a program from the backup memory** If this error occurs, contact your local Schneider Electric support representative.
63	**Hardware configuration incompatibility when loading a program from the backup memory** If this error occurs, contact your local Schneider Electric support representative.

How to Connect the Programming Software to the Smart Relay

Description

See Communication Setup Between the Software and the Smart Relay *(see page 448)*

How to Diagnose the Smart Relay State

Description

See Smart Relay Diagnostics *(see page 456)*

How to Control the Smart Relay from the Programming Software

Description

See RUN/STOP program execution commands *(see page 454)*.

How to Control the Smart Relay from the Front Panel

Description

The LCD display and the command keys can be used to:

- Identify the smart relay and its extensions,
- Monitor the state of the smart relay,
- Configure the smart relay and its extensions (date, time, language, etc.),
- Program an application (program) in LD mode,
- Configure and execute a program,
- Monitor the execution of a program,
- Transfer the programs to and from a memory cartridge.

In order to carry out these actions, you can use:

- Menu screens displayed on the first four lines of the LCD display,
- Contextual information displayed on the 5th line of the LCD display,
- 5 main keys, which are colored, and one white key (**Shift**).

Menu Screens

The menu screens display in the first four lines of the LCD display:

- information, or
- actions
  NOTE: In this case, only the field that flashes can be selected and its selection triggers the action.

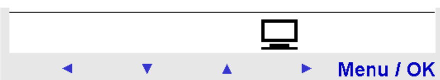
When the information and the actions to perform do not fit in the four lines, symbols ↑ and ↓ on the right side column indicate the presence of information on lines preceding or following those on the screen. These lines can be accessed using the keys located under the screen and designated by the markings ↑ and ↓.

Contextual Information

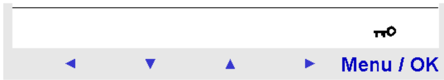
Contextual information belongs to two categories:

- Symbols providing information on the state of smart relay operation:

PC connection

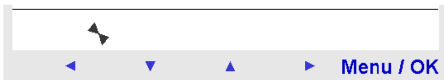


password



alarm

Run/Stop



- Contextual menus located above each key that indicate the action that results when the key is pressed.
  The existence of the contextual menu is shown by a horizontal line displayed at the bottom of the screen and information explaining the functions of the key.

Illustration

Command Keys

The keys belong to two categories:

- Five main colored keys,
- One additional white key (**Shift**).

Illustration

Shift key

The additional white key (**Shift**) is only used for the following actions:

- Showing hidden contextual information:
  - In the **PROGRAMMING** menu in LD mode,
  - In the **MONITORING** menu in LD mode,
  - In the LD TEXT or FBD DISPLAY menu, if **Authorized modification** was checked in the function parameters window.
- In combination with the **Menu/OK** key
  - In the PASSWORD menu, to exit without entering a password,
  - To change to RUN mode between the INPUT-OUTPUT menu and a possible active LD TEXT or FBD DISPLAY menu,
- In combination with the key located under **Param**
  - Modification of the values displayed, including **Modification authorized** in the LD TEXT or FBD DISPLAY menus.

Otherwise, the key is inactive.

Colored keys / Arrow keys

The main colored keys are used for the following actions:

- Gray keys: If no contextual menu is displayed above the key, then the marked symbol applies:
  - ↑: Selection of the preceding line if it exists,
  - ↓: Selection of the following line if it exists,
  - →: Movement to the right of the screen (**PROGRAMMING** and **MONITORING** menu in LD mode), or movement in the screen to each action that can be selected, or inactive key.
  - ←: Movement to the left of the screen (**PROGRAMMING** and **MONITORING** menu in LD mode), or movement in the screen to each action that can be selected, or return to the previous menu.
- Blue key **Menu/OK** display of the menu screen associated with the field selected, or validation of actions or modifications carried out in a menu, or return to the preceding menu when the program is inactive (STOP), or return to the input-output menu or a possible LD TEXT or FBD DISPLAY menu active when the program is executed (**RUN**).

Colored keys / Modification actions

A contextual menu is displayed above the corresponding key:

- **+**: Adds +1 if the selected field (flashing) is a number, or selects another choice if the selected field (flashing) is text.
- **-**: Subtracts 1 if the selected field (flashing) is a number, or selects another choice if the selected field (flashing) is text.
- **Ins**: Inserts an LD diagram line in the **PROGRAMMING** menu in LD mode.
- **Del**: Deletes an LD diagram line in the **PROGRAMMING** menu in LD mode.

Colored keys / Miscellaneous actions

A contextual menu is displayed above the corresponding key:

- **Param**: Can be used to access the menu describing the parameters of an LD function selected in the **PROGRAMMING** or **MONITORING** menus or modification of the displayed modifiable values in LD TEXT or FBD DISPLAY.
- **1, 2, 3, 4**: Pressing the key under the number switches the output of the LD function Zx key or FBD function Zx Button to ON. Releasing the key switches the function back to OFF.

How to Configure an Application from the Front Panel of the Smart Relay

Description

See **PARAMETERS** Menu *(see page 91)*.

Setting the parameters for a program means:

- Changing the daylight saving time switchover settings,
- Defining the conditions under which the program will run (access the **CONFIGURATION** Menu). You may:
  - Modify the program execution frequency,
  - delete or choose a WATCHDOG to monitor program cycle times,
  - modify the smart relay input filters,
  - protect modifications to the program with a password,
  - authorize or disable the **Zx** keys (blue keys) only while program is running in LD mode.

Each function has a number of unique parameters which are not applicable to other functions. Other parameters may, however, apply in the same way across functions. These are:

- Latching. When selected, this parameter enables the data set for a given function to be saved and retrieved after a power outage.
- Locked (only in LD mode): When selected, prevents locked parameters from being displayed and modified using the **PARAMETERS** menu.

NOTE: In FBD mode, locking is a programming option for programming software which locks the front panel button commands, other than the modifications authorized by the **DISPLAY** function blocks.

To modify or display the parameters using the smart relay front panel, you can:

- Access the **PARAMETERS** menu in any operating mode then press the buttons beneath the + and - signs displayed on line 5 on the screen: This displays the parameters used for each program function;
- When the program is running in LD mode (**RUN**), access the **MONITORING** menu then use the blue navigation keys to point to the required function, then press **Shift** (White key). When line 5 on the screen displays **Param**, press the key just below to display the parameter for the selected function.
- When the program is no longer running in LD mode (**STOP**), access the **PROGRAMMING** menu then repeat the procedure described above.

How to Modify Program Data Using the Smart Relay Front Panel

Description

See TEXT, Displaying a Numerical Value *(see page 200)*

See DISPLAY (LCD Screen Display), Parameters *(see page 309)*

How to Recover the Smart Relay Program from the Programming Software

Description

See Transfer the Smart Relay Program to the PC *(see page 452)*

How to Check an Application Using the Programming Software

At a Glance

The **Edit → Check the Program** command launches the program compilation, then the result of the compilation is displayed in the **Compilation results** window.

Two types of verification can be used for an application:

- The first verifies the consistency between LD or FBD diagrams,
- The second verifies the performance of the application, i.e. the suitability of:
  - the memory usage,
  - the execution periods of the application,
  - memory capacities,
  - the execution speed of the smart relay.

Program Consistency Check

If the option: **Display compilation results in simulation mode and when loading** is activated in the programming software preferences *(see page 508)*, the compilation is performed automatically in the following cases:

- Switching from Edit mode to Simulation/Monitoring mode,
- Transferring the program to the smart relay.

Consistency of FBDs

This only concerns SFC network linking errors.

FBD networks behave consistently: inconsistent linking is impossible, and the non-linking of an input sets it to a constant value that does not affect the execution of the function or makes it passive. See the details for the particular function in this document.

Consistency of LDs

The LD network can be simulated, loaded and executed on the smart relay at any time. Hence it can be built and debugged progressively.

However when wiring anomalies are detected (cable without termination, function Reset not connected, etc.) an eye symbol *(see page 220)*  goes from blue to red in the upper panel of the edit window. Double click on the red eye to open a program consistency window which gives details on any detected anomalies.

These notifications are intended to draw your attention to singular wiring instances, which may nevertheless be justified in certain applications.

As a general rule, these anomalies correspond to inputs with incomplete wiring, or no wiring at all (e.g.: Reset function), or to parameters left in their default value, or to certain Clock configurations (where the output stays ON permanently).

Application Performance Check

This appears in the **Compilation result** window in the following cases:

- Activation of the **Edit** → **Check the program** command,
- Switch from Edit mode to Simulation/Monitoring mode (if activated in the programming software preferences *(see page 508)*),
- Transfer of the program to the smart relay (if activated in the programming software preferences *(see page 508)*).

NOTE: When optional, the window is only displayed when the smart relay capacities (memory space and execution speed) are too low in relation to the program being checked.

NOTE: The compilation time for programs that use more than 128 FBDs or SFCs and numerous loops, may exceed several minutes.

Application Estimated Duration Check

The compiler also calculates the estimated duration of the program by adding together the elementary execution periods of each function used.

The application is executed periodically, and its execution period is defined in **Edit** → **Program Configuration** → **Configuration** → **Adjustment of the basic cycle time of the module**, accessible by clicking the  icon in the toolbar (edit, simulation and monitoring mode), and then opening the **Configuration** tab.

This period corresponds to the minimum sampling period of the smart relay inputs (except for: Fast counter function) and the minimum time for modifying the output values. The application response time is therefore twice the duration of this period.

NOTE: Take into account that the compiler arranges FBD functions from inputs to outputs, cutting the loops as close as possible to the outputs and SFCs from each INIT STEP or RESET INIT, to the downstream steps.

Not all automation applications need a WATCHDOG on overrun of the target application execution period. You can therefore choose to use a WATCHDOG *(see page 103)* which will generate an alarm or error, if the application is in RUN mode on the smart relay, and when the application execution period, added to the duration of the processes specific to the operation of the smart relay and any extensions, exceeds the duration of the selected period.

The action of this WATCHDOG (inactive by default) is defined in **Edit** → **Program Configuration** → **Configuration** → **WATCHDOG action (module cycle time control)**, accessible by clicking the  icon in the toolbar (edit, simulation and monitoring mode), and then opening the **Configuration** tab.

A WATCHDOG notification can be returned to an FBD program, enabling you to activate a retrieval sequence in the application, by using the smart relay status *(see page 329)* function.

NOTE: The LD and FBD functions have a defined maximum execution period, with one exception: in FBD, the execution period of the TIME PROG function may vary from 1 to 51 depending on the number of events used.

To determine the duration of the program execution period, you must take into account the estimated duration in the compilation results table.

To have a constant program execution period, you must also perform appropriate tests on the smart relay to verify that Watchdog Overrun does not appear.

Duration of Processes Specific to the Operation of the Smart Relay and any Extensions

In addition to the processing time for the function blocks contained in the application program, there are a number of additional processes during an execution period which can be defined and are taken into account in the calculation of available application execution time (compilation result) of each period.

However, there are others, which can be either occasional or hard to quantify or account for.

Processes hard to quantify:

- Synchronous periodic interruptions which take up an extra 1 millisecond for every 10 milliseconds in a period (i.e. 1 millisecond for a period of 10 milliseconds and 5 milliseconds for a period of 50 milliseconds). These interruptions are used to acquire microprocessor inputs,
- Interruptions relating to the fast counter (H-SPEED COUNT).

Episodic processes:

- Clock management: switch between summer and winter time: 1.60 milliseconds,
- Compensation for clock drift once per week, every Sunday at 01:00: 4.38 milliseconds.

The WATCHDOG has no effect for the period in which one of these processes takes place.

NOTE: If the application presents no requirements in the event of an increase in the execution duration, set the WATCHDOG to inactive. Otherwise, you must verify the maximum execution time.

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION If your application requires a fixed execution period, be sure to take into account the episodic processes. **Failure to follow these instructions can result in death, serious injury, or equipment damage.**

NOTE: Measurements of analog input values on smart relays supplied with an AC voltage (24 Vac and 100-240 Vac) are performed every 20 milliseconds. Consequently, if you choose an execution period of 10 milliseconds, analog input acquisition for these types of smart relays will take place every two cycles.

NOTE: Modifying parameters using the FBD DISPLAY or LD TEXT functions on the front panel of the smart relay increases the application execution period by a variable time. The same applies for the other commands (PARAMETERS, etc.) carried out from the front panel. The WATCHDOG has no effect in this operating mode of the smart relay (Smart Relay Status *(see page 329)*).

NOTE: The display of various data (text, data, time, date) by active FBD DISPLAY functions or an active LD TEXT function, on the smart relay front panel increases the application execution period by a variable duration. This duration depends on the type of data to be displayed and, for the FBD, on the number of DISPLAYS simultaneously active (maximum: 32).

NOTE: In Monitoring mode, the execution times are increased by the communication times between the PC and the smart relay. The real execution time may vary greatly during this operating mode. In addition, during the Monitoring mode, the WATCHDOG action (defined in WATCHDOG Program Configuration) is deactivated (Smart Relay Status *(see page 329)*).

CAUTION

EQUIPMENT OPERATION HAZARD

- Perform real machine or process operational tests before placing this equipment into service.
- Do not rely solely on simulation testing results for the debugging and/or commissioning of your machine or process.

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

How to Check the Smart Relay Firmware

Description

See Smart relay Diagnostics *(see page 456)* and Compatibility between the version of Zelio Soft 2 software and the version of the firmware on the smart relay *(see page 538)*

How to Use the Memory Cartridge

Description

See **TRANSFER** Menu *(see page 107)*

How to Configure the Language of the Programming Software and the Smart Relay

Description

You can configure the language used in the programming software and on that used on the front panel of the smart relay:

- In the programming software: **File → Programming Software Preferences** menu (programming software),
- In the programming software: **Module → Module Language** menu (smart relay),
- From the smart relay front panel: **LANGUAGE** menu (smart relay).

How the Smart Relay Behaves in the Event of Power Outage

Power Outage

A power outage causes the following:

- The application execution is stopped and the buttons are inoperable. The outputs maintain their logical status and values they had prior to the detection of the power outage while they are being saved. After saving, the outputs are turned off (deactivated).
- The links with the PC and any extensions are terminated:
  - The programming software displays the following message: the target peripheral is not reacting. Check the connection,
  - The outputs of the input/output extensions are deactivated after retaining the output values,
  - The Modbus extension normally continues to emit 80 milliseconds after the power outage that affected the smart relay,
  - After 80 milliseconds, the Modbus extension detects the interruption and sets the Modbus status word to Time_OUT_SPI, then loses its power supply.
- The smart relay display is cleared, the backlighting switches off and the smart relay outputs are deactivated.

NOTE: The date and time are maintained during the power outage on smart relays equipped with a clock (battery powered).

Restart Following a Power Outage

The smart relay verifies that its extensions are operating normally, then restores the value of the outputs stored during the power outage, and restarts the application execution with a specific initialization sequence for power return.

This sequence initializes the function inputs and outputs, except the outputs of the functions protected by a checked **latching** parameter.

In this case, these outputs are not reset, and therefore are initialized with the value they had at the time of the power outage.

Functions with **latching** parameters in LD mode:

- Auxiliary Relays *(see page 151)*
- Discrete Outputs *(see page 144)*
- Timers *(see page 182)*
- Counters *(see page 157)*
- Fast Counter *(see page 166)*

Functions with **latching** parameters in FBD mode:

- The ARCHIVE *(see page 327)* function,
- Preset hour counter: PRESET H METER *(see page 302)*
- Timers: TIMER A/C *(see page 289)*, TIMER B/H *(see page 297)*, TIMER Li *(see page 294)*,
- Counters: PRESET COUNT *(see page 276)*,UP_DOWN COUNT *(see page 287)*,
- Fast counter: HI_SPEED COUNT *(see page 281)*,
- CAM block *(see page 324)*.

Special case of SFC components. On restart after a power outage, the positions the status tokens held in SFCs whose continuity was interrupted are:

- re-initialized if the charts did not have RESET-INIT function,
- restored if the charts had a RESET-INIT function.

How to import an application developed using Zelio Soft 1 into Zelio Soft 2

Description

See Conversion of Older Applications using Zelio Soft 2 *(see page 514)*.

Part III
Functions Accessible from the Front Panel

Subject of this Section

This section describes the functions that can be accessed from the front panel of the smart relay in LD and FBD mode.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
3	Overview of the Functions Accessible from the Front Panel	73
4	Input/Output Screen	79
5	PROGRAMMING Menu	81
6	PARAMETERS Menu	91
7	MONITORING Menu	93
8	RUN/STOP Menu	95
9	CONFIGURATION Menu	97
10	CLEAR PROGRAM Menu	105
11	TRANSFER Menu	107
12	VERSION Menu	111
13	LANGUAGE Menu	113
14	DEFAULT Menu	115
15	CHANGE DATE/TIME Menu	117
16	CHANGE SUMMER/WINTER Menu	119

Chapter 3
Overview of the Functions Accessible from the Front Panel

Subject of this Chapter

This chapter describes the different functions that can be accessed from the smart relay front panel, in LD mode.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Functions Accessible from the Front Panel of the Smart Relay	74
Control Keys on the Front Panel of the Smart Relay	76

Functions Accessible from the Front Panel of the Smart Relay

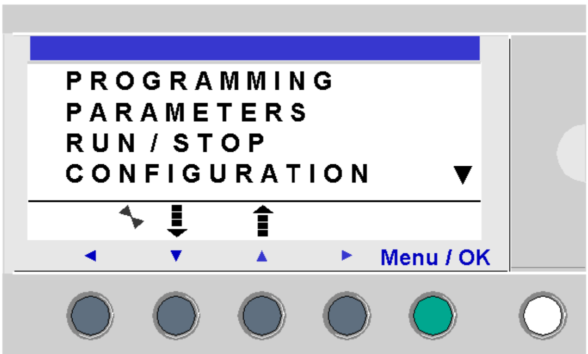
Description

Front panel of the smart relay refers to:
- the front panel of the smart relay itself, as well as
- the display window of the programming software front panel.

In both cases, actions are performed using the keys located on the front panel of the smart relay. These are used for:

- Program (in LD mode),
- Configure,
- Control the application,
- Monitor the performance of the application.

Illustration:



In the programming software, buttons operate:
- **either using the keyboard**: The navigation keys (in gray) are emulated by the navigation keys on the keyboard, the **Menu/OK** key (green) is emulated by the **Enter** key and the **Shift** button (white) by the **Shift** key on the keyboard,
- **Or directly on the representation of the front panel** (front panel window): If you place the mouse cursor over one of the buttons, a hand appears and you may click to validate.

The line flashes to indicate where you are positioned.

The up triangle ▲ on the right side of the LCD screen indicates that possible up options exist. The down triangle ▼ indicates that possible down options exist.

To return to the previous menu, press left navigation key.

NOTE: The LCD screen is illuminated for 30 seconds when a key is pressed on the front panel.

Managing Menus

The inputs-outputs screen is displayed by default whether the mode be **LD** or **FBD**.

Pressing the **Menu/OK** key switches the display from the inputs-outputs screen to the main menu.

The menu on the first row which is selected by default (flashing). The ⬇ and ⬆ navigation keys can be used to place the cursor over the other menus.

Press the green **Menu/OK** key to display the screen corresponding to the selected menu or to move onto the first sub-menu.

Differences Between LD and FBD Modes

Certain menus are specific to either LD or FBD mode.

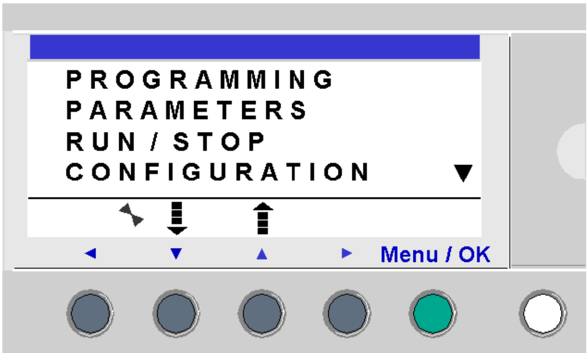
Menu		LD	FBD
PROGRAMMING		✓	
MONITORING		✓	
PARAMETERS		✓	✓
RUN / STOP		✓	✓
CONFIGURATION			
	PASSWORD	✓	✓
	FILTER	✓	✓
	Zx KEYS	✓	
	WATCHDOG CYCLE	✓	✓
CLEAR PROG.		✓	
TRANSFER		✓	✓
VERSION		✓	✓
LANGUAGE		✓	✓
DEFAULT		✓	✓
CHANGE D/T		✓	✓
CHANGE SUMM/WINT		✓	✓

Control Keys on the Front Panel of the Smart Relay

Description

The keys located on the front panel of the smart relay are used to configure, program and control the application and monitor the application's progress.

Illustration:



NOTE: The LCD screen is illuminated for 30 seconds when a key is pressed on the front panel.

Shift Key

The **Shift** key is the white key located on the right side of the LCD screen.

When the **Shift** key is pressed, a contextual menu is displayed above the Z keys.

Menu/OK Key

The **Menu/OK** key is the green key located below the LCD screen on the right side.

This key is used for confirmation of a menu, sub-menu, program, parameter, etc.

Zx Keys

The Zx keys are the gray keys aligned from left (Z1) to right (Z4) and located under the LCD. The arrows indicating the movement direction associated with navigation are marked above the keys.

The navigation keys are used to move left or right, down or up.

The position on the screen appears as a flashing zone:

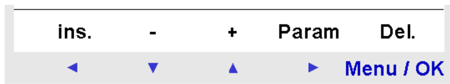
- Square for a position that corresponds to a contact (only in programming menu),
- Round for a link (only in programming menu).

NOTE: When the keys may be used for other actions apart from navigation, a contextual menu bar is displayed (e.g.: 1, 2, 3 and 4 as Zx-type keys).

Contextual Menus

When the cursor is placed on a modifiable parameter, if the **Shift** key is pressed, a contextual menu appears.

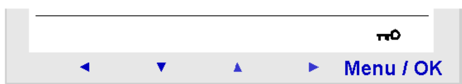
Illustration:



Using the contextual menu functions:

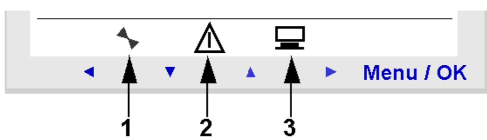
- **+ / -**: Used to scroll through the various possible values of the selected field (types of inputs, outputs, automation functions, numbers, numerical values, etc),
- **Ins.**: Inserts a line,
- **Del.**: Deletes the selected element, or the entire line if it is empty,
- **Param.**: Displays the specific parameter screen for the automation function (visible only if the automation function contains a parameter),
- **← ↑ ↓ →**: Direction of the connection (available only if the cursor is placed over a link box),
- **1 2 3 4**: This line appears when the keys are used as Zx key-type inputs in a program.

Password Protection Illustration:



The key indicates that the program is password-protected.

Other Condition Illustration:



1: Indicates the state of the smart relay. In RUN it is in motion, in STOP it is immobile.
2: Indicates that errors have been detected.
3: The smart relay is physically connected to the programming software.

Chapter 4
Input/Output Screen

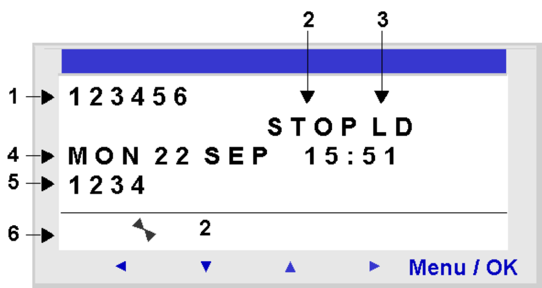
Inputs-Outputs Screen

Description

The inputs-outputs screen is the highest-level interface. It is displayed by default, when no (**TEXT** or **DISPLAY**) display function is active and regardless of:

- the programming type: **LD** or **FBD**,
- the operating mode: **STOP** or **RUN**.

Illustration:



The inputs-outputs screen can be used to view:

1. The state of the inputs: 1 to 9, A to P,
2. The operating mode: RUN / STOP,
3. The programming type used: LD/FBD,
4. The date and time for products with a clock,
5. The state of outputs: 1 to 9, A to G,
6. Z keys: 1 to 4.

In Simulation mode or Monitoring mode when the program is in **RUN**, the active states of the inputs and outputs are indicated in reverse video.

Access to the Main Menu

Pressing the **Menu/OK** key switches the display from the inputs-outputs screen to the main menu:

- PROGRAMMING (**LD** STOP mode),
- MONITORING (**LD** RUN mode),
- PARAMETERS,
- RUN / STOP,
- CONFIGURATION (STOP mode),
- CLEAR PROG. (**LD** STOP mode),
- TRANSFER (STOP mode),
- VERSION,
- LANGUAGE,
- DEFAULT,
- CHANGE D/T,
- CHANGE SUMM/WINT.

The display automatically returns to the inputs-outputs menu on exiting other menus and sub-menus.

Display Functions

The main inputs-outputs screen is replaced by the content of the display functions if:

- In **LD** mode: a **TEXT** function is active.
  If several display functions are active simultaneously, only the last block to be activated is displayed.
- In **FBD** mode: a **DISPLAY** function is active.
  If several display functions are active simultaneously, all the blocks are displayed. If there is overlap between the fields displayed, the DISPLAY for the highest block number is shown.

Switching between the screens

It is possible to go from the **TEXT** (LD) or **DISPLAY** (FBD) screen to the inputs-outputs screen and vice-versa.

To do this, press and hold down the **Shift** key and press the **Menu/OK** key.

Chapter 5
PROGRAMMING Menu

Subject of this Chapter

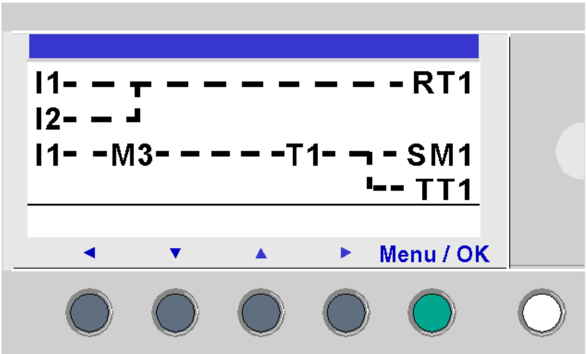
This chapter describes the characteristics of the **PROGRAMMING** menu specific to **LD** mode / smart relay in **STOP** mode.

This function lets you enter the ladder diagrams that will work on the smart relay.

This program is written only using a ladder diagram LD.

The programming software can be used in Zelio Entry mode to construct an LD program *(see page 135)* as if programming by using the buttons on the smart relay front panel.

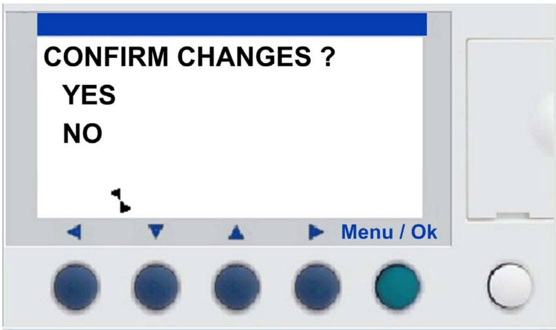
Illustration:



NOTE: The smart relays to which have been added an Input/Output extension are programmable only in **FBD** mode from the programming software.

NOTE: In front panel programming with 240 lines, program memory is stored in two banks. One bank contains lines 1 to 120, and the other bank contains lines 121 to 240. You must do the program modifications in the first part (line 1 to 120) or in the last part (line 121 to 240) and save them before being able to modify the other part.
When the cursor moves from line 120 to 121, the front panel notifies you to save the modification (see picture below) and display the line 121. Then modification could be done on last part of the ladder.

When the cursor moves from line 121 to 120, the front panel notifies you to save the modification (see picture below) and display the line 120. Then modification could be done on first part of the ladder.



NOTE:
No link between the upper part (line 1 to 120) and the lower part (line 121 to 240) could be made in Ladder front panel programming:

- On line 120, it is not possible to insert a descending link (the descending link is displayed in the contextual menu but is ineffective).
- On line 121, it is not possible to insert a ascending link (the ascending link is displayed in the contextual menu but is ineffective).
- Insert a line in the upper part is possible only if line 120 is empty. If a line is inserted in upper part, the lower part is not modified.
- Delete a line in the upper part does not modify the lower part (line 121 does not move to line 120).

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Method for Entering a Contact or Coil	83
Entering a Link	86
Entry of Function Block Parameters	88
Deletion and Insertion of Diagram Lines	89

Method for Entering a Contact or Coil

Description

NOTE: Accessible only in **LD mode** / smart relay in STOP mode.

This section describes the procedures for performing the following operations:

- Entering an element,
- Modifying an element,
- Deleting an element.

This is valid for: contact or coil elements, whether the parameters can be set or not.

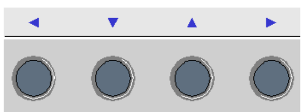
Entering an Element

When entering an element, the following rules must be observed:

- **Contact**: In any column except the last,
- **Coil**: Only in the last column.

The presence of a square, flashing cursor means an element can be inserted.

Entry procedure:

Step	Action
1	Place the flashing cursor at the required location. The navigation keys can be used move the cursor in the direction of the arrows on the navigation keys ◄ ▼ ▲ ►. Illustration: 
2	Press the **Shift** key to display the contextual menu. Illustration:  By simultaneously pressing **Shift** and one of the ▼ ▲ (- and +) keys, the first letter of the element is inserted: **I** for a contact and **Q** for a coil, followed by the number **1**.

Step	Action
3	Choose the type of element desired by pressing simultaneously on **Shift** and **+** or **-**. This makes the different types of elements scroll down cyclically, in the following order: ● For the contacts: I, i, Z, z, N, n, M, m, Q, q, T, t, C, c, K, k, V, v, A, a, H, h, W, w, S, s. ● For the coils: M, N, Q, T, C, K, X, L, S. See the chapter *LD Language Elements, page 143*.
4	Release the **Shift** key to have access to the navigation keys: ◀ ▼ ▲ ▶. Pressing the ▶ key places the cursor over the corresponding number **1**.
5	Simultaneously hold down the **Shift** and **+** keys to increment the number of the element (2, 3, 4,..., 9, A, etc.). **NOTE:** The numbers for functional blocks are limited to the number of blocks of the type available in the smart relay. In the case of extensible smart relays, the inputs and outputs numbers are used to program the extension to maximum size. In entering a contact, once this step is completed, the entry is terminated. In entering a coil, you must additionally select the function of the coil.
6	Release the **Shift** key to have access to the navigation keys: ◀ ▼ ▲ ▶.
7	**Steps 7 to 9 are only necessary when entering a coil.** Position the cursor on the function of the coil by pressing twice on the ◀ key.
8	Select the desired function by pressing simultaneously on the **Shift** key and the **+** or **-** key. This will scroll through the different coil functions available.
9	Release the **Shift** key to have access to the navigation keys: ◀ ▼ ▲ ▶.

NOTE: Confirming some function block coils will bring-up a function block parameter setting screen.

Modifying an element

To modify an existing control diagram element:

- Position the pointer over the element to modify: Step **1** in the previous table,
- Select the desired new element: Steps **3** to **6**.

Modification of the State of a Contact

In the programming software, to modify the state of a contact, position the pointer on it, then:

- With the mouse: Right-click to display a list of possible states (click to confirm),
- With the space bar: Scroll through the possible states.)

Initialization

Status of contacts on program initialization:

- A normally open contact (direct state) is inactive,
- A normally closed contact (reverse state) is active,

Deleting an Element

To delete an element:

- Place the cursor over the element to delete
- Simultaneously press the **Shift** and **Del** (**Menu/OK**) keys.

Two scenarios are possible, depending on the position of the cursor at the time of the deletion:

- Cursor over an element: the element is deleted,
- Cursor over an empty position of the line: the line is deleted.

NOTE: Generally, the deleted element must be replaced by a link.

Entering a Link

Description

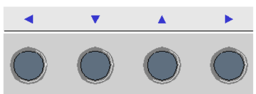
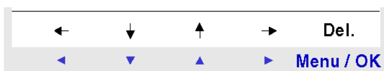
NOTE: Accessible only in **LD mode** / smart relay in STOP mode.

This section describes the procedures for performing the following operations:

- Entering/Modifying links between elements,
- Deleting links between elements,
- Replacing a link with a contact.

Entering/Modifying a Link

Links are entered exclusively using the round flashing cursor.

Step	Action
1	Place the flashing cursor at the required location. The navigation keys can be used move the cursor in the direction of the arrows on the navigation keys ◄ ▼ ▲ ►. Illustration: 
2	Press the **Shift** key to display the contextual menu. Illustration: 
3	Trace connections by simultaneously pressing the **Shift** key and the navigation keys: ← ↑ ↓ →. **Shift** and → to trace a connection to the position of the next contact or to the coil at the end of the line. **Shift** and ↑ ↓ to trace perpendicular connections to the previous or next line. **NOTE:** You cannot add a perpendicular connection between lines 120 and 121.
4	Release the **Shift** key to have access to the navigation keys: ◄ ▼ ▲ ►.
5	Repeat the operation as many times as necessary to create a program.

Deleting a Link

To delete a link, simply:

- Place the cursor over the element to delete.
- Simultaneously press the **Shift** and **Del** (**Menu/OK**) keys.

Two scenarios are possible, depending on the position of the cursor at the time of the deletion:

- Cursor over a link: The link is deleted,
- Over an empty position of the line: The line is deleted.

Replacing a Link with a Contact

Refer to the element entry procedure *(see page 83)*.

Entry of Function Block Parameters

Description

NOTE: Accessible only in **LD mode** / smart relay in STOP mode.

When entering a control diagram, the parameters of the configurable automation functions must be completed.

The automation functions with parameters are the following:

- Auxiliary relays *(see page 151)* (latching),
- Discrete Outputs *(see page 144)* (latching),
- Clocks *(see page 176)*,
- Analog Comparators *(see page 192)*,
- Timers *(see page 182)*,
- Counters *(see page 157)*,
- Fast counters *(see page 166)*.

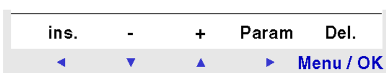
Accessibility of parameters

Function block parameter setting can be accessed:

- When entering the command diagram line,
- From the **PARAMETERS** menu if the block has not been padlocked.

Entering/Modifying Parameters of the Block

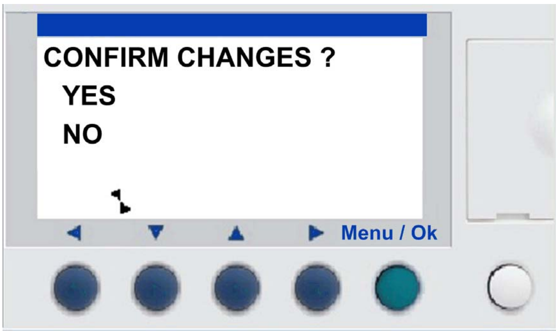
Parameters are entered in the same way, whatever the parameters screen:

Step	Action
1	Place the flashing cursor at the required function. When the function has parameters, **Param** appears in the contextual menu (when the **Shift** key is pressed). Illustration: 
2	Press and hold down the **Shift** key and press on **Param** (key ►). **Result**: The function's parameter screen appears.
3	Use the navigation keys to move to the cursor over the modifiable parameters: ◄ ►.
4	Modify the value of the parameter using the **+** and **-** keys, holding down **Shift**.
5	Confirm the modifications by pressing **Menu/OK**, which will open the confirmation window. Confirm again by pressing the **Menu/OK** key to save.

Deletion and Insertion of Diagram Lines

Introduction

NOTE: In front panel programming with 240 lines, program memory is stored in two banks. One bank contains lines 1 to 120, and the other bank contains lines 121 to 240. You must do the program modifications in the first part (line 1 to 120) or in the last part (line 121 to 240) and save them before being able to modify the other part.
When the cursor moves from line 120 to 121, the front panel notifies you to save the modification (see picture below) and display the line 121. Then modification could be done on last part of the ladder.
When the cursor goes from line 121 to 120, the front panel notifies you to save the modification (see picture below) and display the line 120. Then modification could be done on first part of the ladder.



NOTE:
No link between the upper part (line 1 to 120) and the lower part (line 121 to 240) could be made in Ladder front panel programming:

- On line 120, it is not possible to insert a descending link (the descending link is displayed in the contextual menu but is ineffective).
- On line 121, it is not possible to insert a ascending link (the ascending link is displayed in the contextual menu but is ineffective).
- Insert a line in the upper part is possible only if line 120 is empty. If a line is inserted in upper part, the lower part is not modified.
- Delete a line in the upper part does not modify the lower part (line 121 does not move to line 120).

Deletion

NOTE: Accessible only in **LD mode** / smart relay in STOP mode.

Diagram lines are deleted line-by line. The procedure is the following:

Step	Action
1	Place the cursor over the line to delete.
2	Delete all the elements in the line *(see page 83)*: (Links, contacts and coils) to obtain an empty line.
3	Press the **Shift** key to display the contextual menu. Illustration:  Simultaneously pressing **Shift** and **Del** opens the confirmation window.
4	Confirm by pressing **Menu/OK**.

NOTE: It is possible to delete all diagram lines contained in the smart relay. In order to do this, select the **CLEAR PROG.** option from the main menu, and confirm the deletion of all the control diagram lines.

Insertion

The procedure is the following:

Step	Action
1	Place the cursor over the line located immediately below the line to create.
2	Press the **Shift** key to display the contextual menu.
3	Press the **Ins** key (while holding down the **Shift** key) to create the line.

Chapter 6
PARAMETERS Menu

PARAMETERS Menu

Description

This menu is used to enter and modify the application parameters directly on the screen using the smart relay keys. This function can be accessed in the two modes: **LD** and **FBD**, but the contents will be specific to the mode used.

If there are non-locked parameters to display they are listed in the window; otherwise a **NO PARAMETER** message appears.

LD Mode

Functions with parameters in LD mode:

- Auxiliary relays *(see page 151)* (latching),
- Discrete Outputs *(see page 144)* (latching),
- Clocks *(see page 176)*,
- Analog Comparators *(see page 192)*,
- Timers *(see page 182)*,
- Counters *(see page 157)*,
- Fast counter *(see page 166)*.

Only those functions used in the program and with parameters are listed in the **PARAMETERS** menu.

FBD Mode

Functions with parameters in FBD mode:

- Numerical Constant-Type Inputs *(see page 258)*,
- Clocks *(see page 314)*,
- Gain *(see page 306)*,
- Timers: TIMER A/C *(see page 289)*, TIMER B/H *(see page 297)*, TIMER Li *(see page 294)*,
- Counters: PRESET COUNT *(see page 276)*,
- Fast counter,
- CAM block *(see page 324)*.

To access the parameters of the FBD blocks, you must enter the block number. This number appears on the wiring sheet at the top right corner of the block *(see page 22)*.

Only those functions used in the program and with parameters are listed in the **PARAMETERS** menu.

Parameter Modification

Parameter modification procedure:

Step	Action
1	Place the cursor over the **PARAMETERS** menu in the main menu (PARAMETERS flashing) and confirm by pressing the **Menu/OK** key. **Result**: The parameters window opens to the first parameter.
2	Select the function to modify. To access the required function, scroll through the function block numbers (navigation keys ▼ and ▲) until you reach the one to edit.
3	Select the parameter to modify. The ◄ and ► keys are used to place the cursor over the parameter to modify.
4	Modify the parameter using the **+** and **-** keys (▲ and ▼) of the contextual menu.
5	Confirm the modifications by pressing **Menu/OK**, which will open the confirmation window.
6	Confirm again twice by pressing **Menu/OK** to save. **Result**: The display returns to the INPUTS-OUTPUTS screen in RUN mode and to the MAIN menu in STOP mode.

Parameters in RUN Mode

It is possible to modify parameters in RUN mode as long as they are not locked.

The modifications can be made:

- From the **PARAMETERS** *(see page 91)* menu,
- From the **MONITORING** *(see page 93)* (LD) menu: Move the pointer over the function to be modified using the navigation keys and open the parameters window from the contextual menu (**Shift** key).

Chapter 7
MONITORING Menu

MONITORING Menu

Description

NOTE: Accessible only in **LD mode** / smart relay in **RUN** mode.

MONITORING mode can be used to obtain a dynamic view of the state of the smart relay inputs/outputs.

In this mode the wiring grid appears as it does in the PROGRAMMING *(see page 81)* menu (smart relay in **STOP** mode), but appear in reverse video when inputs or outputs are activated (white on black background).

Illustration:

```
I1-i2----------[Q 1
IB-------------TT1
T1-------------[Q2
H1-------------[M1
```

This mode is also used to dynamically modify the values of automation function parameters if these are not locked.

NOTE: When using the window of the front panel of the programming software, you must click on the window to make it active.

Parameter Modification

To modify the parameters, proceed as follows:

Step	Action
1	Use the navigation keys to move the cursor over the element to modify.
2	Hold down **Shift** key then press the **Param** key to open the parameter window.
3	Use the navigation keys to move to the cursor over the modifiable parameters: ◄ ►.
4	Change the parameter value using the keys **+** and **-**.
5	Confirm the modifications by pressing **Menu/OK**, which will open the confirmation window. Confirm a second time by pressing **Menu/OK** to save.

Step	Action
6	Confirm again with **Menu/OK**. **Result**: Return to the parameter screen.
7	Confirm again with **Menu/OK**. **Result**: Return to the LD diagram screen.

Chapter 8
RUN/STOP Menu

RUN/STOP Menu

Description

This function is used to start or stop the program in the smart relay:

- In **STOP** mode: The program is stopped and the outputs disabled,
- In **RUN** mode (with or without initialization of latching parameters): The program is executed.

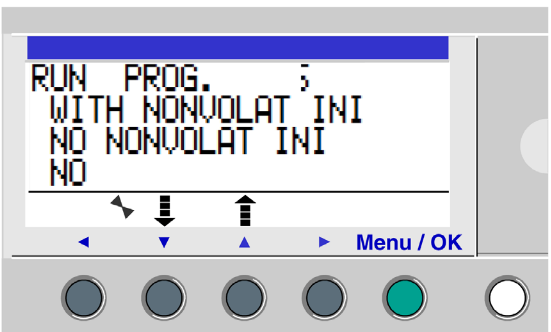
NOTE: The outputs of an extension SR3XT43BD are not disabled if an acceleration time is specified in the extension parameters *(see page 485)*.

Startup

In STOP mode, when accessing the RUN/STOP menu, the interface proposes the following three choices for starting the program:

- **WITH NONVOLAT INI**: All values (counters, timers, etc.) are reset to their initial values before the program starts (default selection),
- **NO NONVOLAT INI**: Values for which the **Latching** option has been activated are kept,
- **NO**: The program does not start.

Illustration:



The navigation keys ▼ ▲ are used to change the selection.

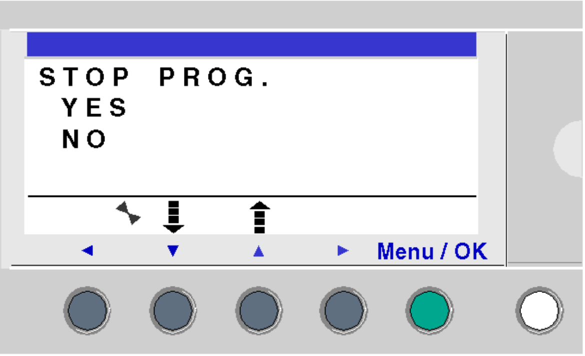
When the setting has been validated with the **Menu/OK** key, the display moves to the **INPUT-OUTPUT** screen.

Off

In RUN mode, when accessing the RUN/STOP menu, you need to confirm the request to stop the program:

- **YES**: The program stops (selected by default),
- **NO**: The program does not stop.

Illustration:



The navigation keys ▼ ▲ are used to change the selection.

When the setting has been confirmed with the **Menu/OK** key, the display moves to the **INPUT-OUTPUT** screen.

Smart Relays Without Screen

For smart modules without screen, a green LED located on the front panel of the module is an indicator light:

- If the LED flashes slowly (3 Hz), the module is in RUN mode (even if there is a recoverable error detected).
- If the LED flashes rapidly (5 Hz), the module is in STOP mode with an error detected.
- If the LED stays illuminated, the module is powered-up and in STOP mode.

NOTE: On power up, the smart relay is in RUN mode, unless there is an error detected.

NOTE: When an error has been detected, eliminate the source of the error and power cycle the module.

Chapter 9
CONFIGURATION Menu

Subject of this Chapter

The **CONFIGURATION** menu provides access to the following 4 functions:
- PASSWORD,
- FILTER,
- Zx KEYS,
- WATCHDOG & CYCLE

This chapter describes the characteristics of these functions.

NOTE: Use the navigation key to return to the main menu ◄.

NOTE: If the program is password-protected, (key displayed in the contextual menu), the user must enter the password before any action can take place in the sub-menus.

NOTE: The **CONFIGURATION** menu is only available in STOP mode.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
PASSWORD Menu	98
FILTER Menu	101
Zx KEYS Menu	102
WATCHDOG CYCLE Menu	103

PASSWORD Menu

Description

If the program is password-protected (key icon appears), you must enter the password to perform certain operations.

The password protects access to the following menus:

- PROGRAMMING (**LD** STOP mode),
- MONITORING (**LD** RUN mode),
- CONFIGURATION (STOP mode),
- CLEAR PROG. (**LD** STOP mode),
- MODULE TRANSFER > MEM (STOP mode).
- MEM TRANSFER > MODULE (STOP mode). In **LD** mode, password protection of this menu is configurable *(see page 507)*.

Activating the password also involves usage limitations in the programming software:

- Modification of the program contained in the smart relay,
- Refreshing of the program contained in the smart relay,
- Overwrite by transferring another program.
- Monitoring

NOTE: If you lose a password, you can overwrite the program from the programming software:

- Transferring/Clearing the program *(see page 458)*,
- UpdateSmart Relay Firmware *(see page 461)*, for more information about the compatibility of the firmware, refer to Compatibility between the version of the programming software and the version of the firmware on the smart relay *(see page 538)*.

NOTE: It is possible to quit the screen without entering a password. Hold down **Shift** key (white key) then press the **Menu/OK** key (green key).

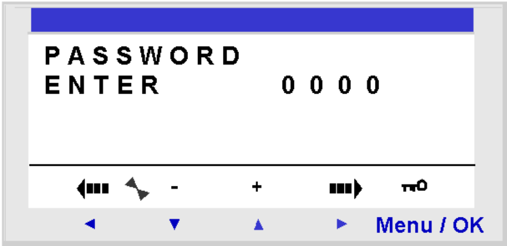
NOTE: To return to the main menu from the **CONFIGURATION** menu, use the navigation key ◄.

Defining Password

Initially, the key is not displayed and each digit is set to 0.

The **ENTER** message appears in the window.

Illustration:



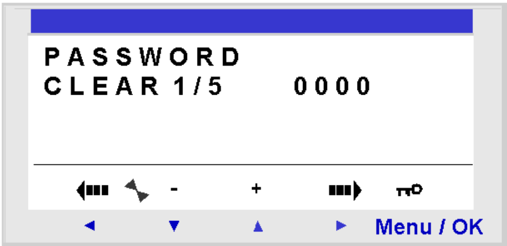
Entry procedure:

Step	Action
1	Use the navigation keys to select the digit to enter: ◄ ►.
2	Select the value of the digit using the **+** and **-** keys of the contextual menu.
3	Confirm the password with the **Menu/OK** key, which opens the confirmation window.
4	Confirm again with the **Menu/OK** key. **Result**: The display returns to the MAIN menu.

NOTE: Henceforth the key is displayed in the contextual menu line.

Removing Password

To inhibit the password, follow the same procedure used to enter it.



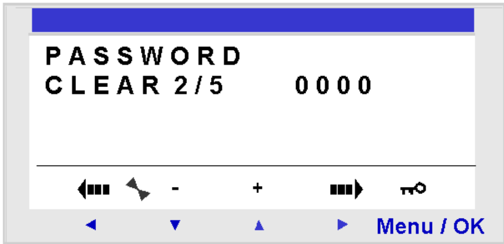
Initially, the key icon is displayed, meaning: Smart relay protected.

The message **CLEAR** and the number of attempts **1 / 5** appear in the window.

The following scenarios may arise:

- **Password correct**: The password is then inhibited, and the smart relay returns to the **PASSWORD** menu,
- **Password incorrect**: The **CLEAR** counter is incremented.

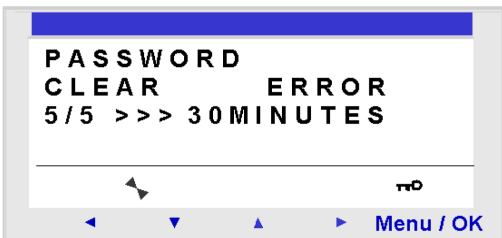
Illustration:



If an incorrect password is entered **5** times consecutively, the security function is locked for 30 minutes.

During this period, if the power supply to the smart relay is interrupted, the downcount will start again on power up.

Illustration:



Modifying Password

To modify the password, remove the password and then enter a new one.

FILTER Menu

Description

This function is used to set the filter time of the inputs. A fast filter detects an input signal change faster than a slow filter; however, a fast filter is more sensitive to disturbances such as signal bounce.

Two choices are available:

- Fast
- Slow

Response time:

Filtering	Commutation	Response time
Slow	ON → OFF	5 milliseconds
	OFF → ON	3 milliseconds
Fast	ON → OFF	0.5 milliseconds
	OFF → ON	0.3 milliseconds

This selection can only be made when the smart relay is in STOP. By default, the smart relays are configured in SLOW.

NOTE: This function is available on smart relays with a direct voltage power supply.

NOTE: To return to the main menu from the CONFIGURATION menu, use the navigation key ◄.

Filter-Type Selection

The type is indicated by the selection symbol (black diamond).

Procedure for selection of filter type:

Step	Action
1	Select the type of filtering using the ▼ ▲ keys (the selection will flash).
2	Confirm with **Menu/OK**. **Result**: the display returns to the MAIN menu.

Zx KEYS Menu

Description

NOTE: Only accessible in **LD mode**.

The **Zx KEYS** option is used to activate or deactivate the use of the navigation keys as pushbuttons.

Different functions can be obtained depending on the state of this option:

- **Inactive**: The keys are only available for setting, configuring and programming the smart relay.
- **Active**: they can also be used in a control diagram.
  In this configuration, they operate as pushbuttons: Zx keys *(see page 155)*, without the need to use a terminal input contact.

NOTE: To return to the main menu from the CONFIGURATION menu, use the navigation key ◄.

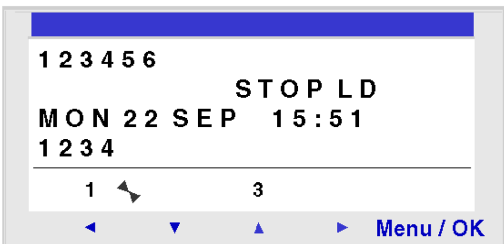
Zx Keys in RUN Mode

By default, the Zx keys are used as navigation keys.

In RUN mode, when the inputs-outputs screen, TEXT screen or DISPLAY screen is active, the numbers of the Zx keys used in the program are displayed in the contextual menu line.

To activate the key, simply select the required key ◄ ▼ ▲ ►.

Illustration:



NOTE: The function is inactive in Parameters mode, Monitoring and all the function block parameter and configuration screens.

WATCHDOG CYCLE Menu

Description

The duration of a program cycle depends on its length and complexity; in particular, the type and number of I/O and the number of extensions.

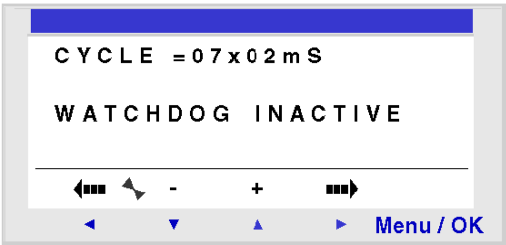
The program is executed periodically at regular time intervals. This time interval is called the **cycle** time.

The program will only execute completely if the cycle time is greater than the program execution time.

The cycle period is configurable in the **CONFIGURATION → WATCHDOG CYCLE** menu. This period may be set from 6 to 90 milliseconds in 2-millisecond steps.

The default value of the cycle period is 14 milliseconds.

Illustration:



NOTE: Make sure that:
- Input variations that are too rapid are not hidden by a cycle time that is too slow.
- The speed of output variations is compatible with system commands.

If the duration of the execution cycle of the program and the embedded software functions exceeds the cycle time value selected by the programmer, the WATCHDOG can be used to operate a specific action.

NOTE: In certain dialog phases, the cycle times are increased by the communication times between the PC and the smart relay. The real cycle times vary greatly during this operating mode. The WATCHDOG is always inhibited in this smart relay operating mode.

NOTE: To return to the main menu from the CONFIGURATION menu, use the navigation key ◄.

Actions

The WATCHDOG can perform the following actions:

- **INACTIVE**: Normal operation
- **ALARM**: An error condition is set and the error code corresponding to **Cycle time overrun** is accessible in the **DEFAULT** menu.
- **ERROR**: The program stops (STOP mode) and the error code corresponding to **Cycle time overrun** is accessible in the **DEFAULT** menu.

Cycle Time

The cycle time may be set from 6 to 90 milliseconds in 2-millisecond steps.

To adjust this period, adjust the 2-millisecond step multiplier factor using the **+** and **-** keys in the contextual menu. This factor is between 3 and 45.

```
CYCLE =07 x02mS
```

The multiplier factor is adjusted depending on the shortest sampling period of the inputs.

WATCHDOG Configuration

Procedure:

Step	Action
1	Configure the **CYCLE** parameter using the **+** and **-** keys in the contextual menu.
2	Confirm the entry using one of the following keys: ◄ or ►. **Result**: The **CYCLE** parameter is confirmed and the **WATCHDOG** parameter is selected (it flashes).
3	Configure the **WATCHDOG** parameter using the **+** and **-** keys in the contextual menu.
4	Confirm your changes by pressing the **Menu/OK** key. **Result**: The display returns to the MAIN menu.

Chapter 10
CLEAR PROGRAM Menu

CLEAR PROG Menu.

Description

NOTE: Accessible only in **LD mode**.

This function is used to clear the entire program.

NOTE: If the program is protected (key displayed), the user must enter the password (see *PASSWORD Menu,* *page 98*) before being able to delete the program.

Clearing the Program

On opening, NO is selected by default.

Procedure:

Step	Action
1	Select the **YES** choice using the navigation keys ▼ and ▲.
2	Confirm the clear command by pressing the **Menu/OK** key. **Result**: the display returns to the MAIN menu.

Chapter 11
TRANSFER Menu

TRANSFER Menu

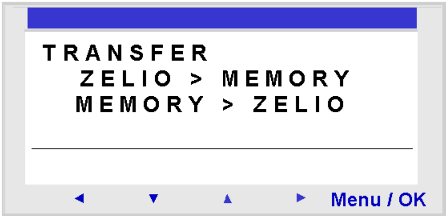
Description

This function is used to:

- Load the firmware and the application contained in the smart relay into the backup memory.
- load firmware and application from the backup memory to the smart relay.

This backup memory can then be used to load the firmware and the application into another smart relay.

Illustration:



NOTE: The backup memory is provided as an option.

NOTE: Insertion and extraction of the backup memory may be performed even when the smart relay is powered up.
For smart relays without screens, detection of the memory may only be performed on power up of the smart relay, if the memory is inserted when the smart relay is powered on, it will not be acknowledged.

NOTE: If the application is protected (key icon displayed), you must enter the password before being able to save the program.

NOTE: If an application is already present in the backup memory, it will be overwritten by the new transfer.

NOTE: It is not possible to directly transfer an application created with version V2 of the programming software from the SR2MEM01 memory to the smart relay if this latter contains version V3 firmware.
In this case, see what action you should take in the section Application incompatible with firmware on the smart relay *(see page 109)*.
For more information about the compatibility of the memory cartridges, see Compatibility between the memory cartridges and the firmware on the smart relay *(see page 541)*.

Module → Backup Memory Transfer

Procedure for transferring the application from the smart relay to the backup memory, for a smart relay with LCD and keyboard:

Step	Action
1	Insert the memory cartridge (SR2MEM02) into the slot provided.
2	Select the transfer type: **ZELIO>MEMORY** using the navigation keys ▼ ▲.
3	Confirm the transfer command with the **Menu/OK** key. (Enter the password if the program is password-protected).
4	Wait for the transfer to end. Display: **> > > MEMORY** then **TRANSFER. OK** when it is completed.
5	Confirm again by pressing **Menu/OK** key to exit the menu. **Result**: The display returns to the INPUTS-OUTPUTS screen in RUN mode and to the MAIN menu in STOP mode.

NOTE: It is not possible to transfer the application from a smart relay without LCD or keyboard. You can transfer the module application to the PC using Zelio Soft *(see page 452)*.

Backup Memory → Module Transfer

The program transfer from one smart relay to another via a memory card is only possible between smart relays with the same reference.

Procedure for transferring the application from the backup memory to the smart relay, for a smart relay with LCD and keyboard:

Step	Action
1	Insert the memory cartridge (SR2MEM02) with the program to be transferred into the slot provided.
2	Select the transfer type: **MEMORY>ZELIO** using the navigation keys ▼ ▲.
3	Confirm the transfer command with the **Menu/OK** key.
4	Wait for the transfer to end. Display: **> > > MODULE** then **TRANSFER. OK** when it is completed.
5	Confirm again by pressing **Menu/OK** to exit the menu. **Result**: the display returns to the INPUTS-OUTPUTS screen in RUN mode and to the MAIN menu in STOP mode.

Procedure for transferring the application from the backup memory to the smart relay, for a smart relay without LCD or keyboard:

Step	Action
1	Since the smart relay **is not powered-on**, insert the memory cartridge (SR2MEM02) into the slot provided.
2	Power up the smart relay. During the transfer, the LED display is off.
3	Wait for the transfer to end. During the transfer, the LED display is off, then at the end of the transfer the LED flashes.
4	● If the flashing is slow (3 Hz), the transfer has been successful, the smart relay is in RUN, remove the memory cartridge (SR2MEM02). ● If the flashing is rapid (5Hz), the transfer has been unsuccessful due to incompatibility between the configuration necessary for the program to be transferred and that of the smart relay.

NOTE: When the smart relay is in STOP mode, the LED display is illuminated and does not flash.

Possible Errors

Below are the possible errors and, for each case, the messages that are displayed:

- Absence of backup memory
  Error message:
  TRANSFER ERROR: NO MEMORY
- Configurations of the hardware and program to transfer incompatible
  Error message:
  TRANSFER ERROR: CONFIG INCOMPAT (hardware or software reference numbers).

For more details, refer to the DEFAULT menu *(see page 115)* chapter.

Application incompatible with firmware on the smart relay

If the application stored in backup memory SR2MEM01 was created with a version of the programming software that is incompatible *(see page 541)* with the firmware of the target smart relay, proceed as follows:

Step	Action
1	Load the application from the backup memory to a smart relay with compatible firmware. **NOTE:** If no smart relay has a firmware that is compatible with the application, use the programming software version that was used to create the application to load a compatible firmware into the target smart relay.
2	Use the version of the programming software that was used to create the application to load it from the smart relay toward the PC.

Step	Action
3	Save the application uploaded in step 2.
4	Launch the latest version of the programming software.
5	Open the application saved in step 3. **Result**: The programming software converts the application.
6	Load the converted application and the associated firmware to the target smart relay.

Use of SR2MEM01 and SR2MEM02

On SR2MEM01, only the program is loaded whereas on SR2MEM02 the program and the corresponding firmware are loaded.

Consequently:

- With the SR2MEM01 memory cartridge, you can perform:
  - A smart relay to memory transfer if the version of the firmware on this relay is strictly lower than 3.09.
  - A memory to smart relay transfer if the program contained in the SR2MEM01 memory cartridge is loaded from a smart relay that has the same version of firmware as the smart relay to which you want to load the cartridge.
- With the SR2MEM02, memory cartridge, you can perform:
  - A smart relay to memory transfer if the version of the firmware on this relay is equal to or greater than 3.09.
  - A memory to smart relay transfer if the version of the firmware on the relay to which you want to load the cartridge is greater than 3.09.

For more information about the compatibility of the memory cartridges, see Compatibility between the memory cartridges and the firmware on the smart relay *(see page 541)*.

Chapter 12
VERSION Menu

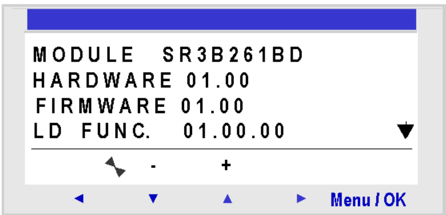
VERSION Menu

Description

This function is used to precisely identify the version of each system component:

- MODULE: smart relay reference,
- HARDWARE: hardware version,
- FIRMWARE: firmware version,
- LD FUNC: language functional level if LD language or
  FBD FUNC: language functional level if FBD language.

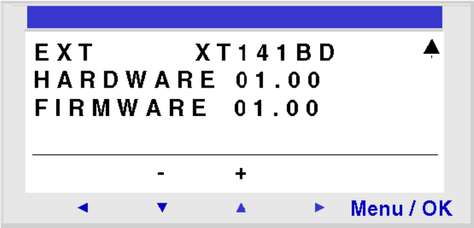
Illustration:



This information is available for the smart relay, but also for the connected extensions.

The ▼ symbol is present in the bottom right, indicating the existence of extensions connected to the smart relay.

Illustration:



To quit, press the **Menu/OK** button, the display returns to the INPUTS-OUTPUTS screen if smart relay is in **RUN** mode and to the MAIN menu if smart relay is in **STOP** mode.

Chapter 13
LANGUAGE Menu

LANGUAGE Menu

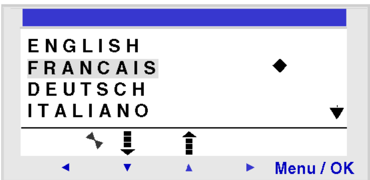
Description

This function is used to select the language used by the smart relay.

All messages may be viewed in 6 languages:

- English,
- French,
- German,
- Italian,
- Spanish,
- Portuguese.

Illustration:



Language Selection

The current language is indicated by the selection symbol (black diamond).

Language selection procedure:

Step	Action
1	Select the language using the navigation keys: ▼ and ▲ (the selection flashes).
2	Confirm with the **Menu/OK** key. **Result**: The display returns to the INPUTS-OUTPUTS screen in RUN mode and to the MAIN menu in STOP mode.

Chapter 14
DEFAULT Menu

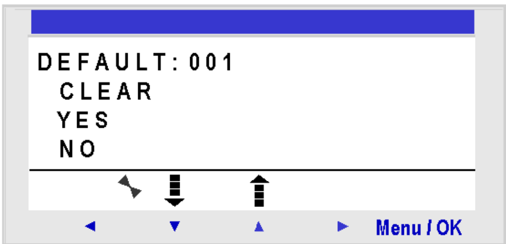
DEFAULT Menu

Description

This function is used to:

- Display on the LCD screen the type of error detected by the firmware of the smart relay (Watchdog overrun, *see WATCHDOG CYCLE Menu,* *page 103*, cycle time too high, etc.),
- Reset the error counter to zero.

Illustration:



Reset to Zero of the Error Counter

To reset the error counter to zero, proceed as follows:

Step	Action
1	Select the **YES** choice using the navigation keys ▼ and ▲.
2	Confirm the clear command by pressing the **Menu/OK** key. **Result**: The display returns to the INPUTS-OUTPUTS screen in RUN mode and to the MAIN menu in STOP mode.

Description of Errors

Description of Smart Relay Errors *(see page 521)*

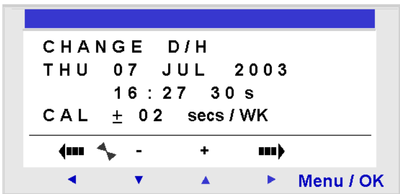
Chapter 15
CHANGE DATE/TIME Menu

CHANGE DATE/TIME Menu

Description

This function is used to configure the date and time of the smart relays that have a clock.

Illustration:



The modifiable parameters are:
- Day / week / month / year,
- Hour, minutes, seconds,
  Values are recorded by pressing the **Menu/Ok** key.
- CAL: Calibration of the internal clock of the smart relay in seconds per week.

Clock Calibration

The quartz that controls the real-time clock of the smart relay has a variable monthly drift depending on the environmental conditions of the smart relay.

The maximum value for this drift is approximately one minute per month.

To estimate this drift, proceed by observing the drift on the smart relay clock with respect to a reference clock for a few weeks or more.

Example:

If you wish to compensate this drift, you can for example make a -15 second correction per week to compensate for a + 60 second drift per month. This compensation is executed on Sunday at 01:00.

NOTE: This correction serves no purpose if the smart relay is subject to long power interruptions or major variations in temperature.

Clock Configuration

Procedure:

Steps	Description
1	Select the parameter to modify using the navigation keys ◄ and ►. **Result**: The selected parameter flashes.
2	Modify the value of the parameter. The **+** and **-** keys of the contextual menu can be used to change the current value.
3	Confirm the changes by pressing the **Menu/Ok** key. **Result**: The display returns to the MAIN menu.

NOTE: The smart relay determines the day of the week when the day of the month in the year is selected.

NOTE: You cannot modify the hour by a product between 2:00 and 3:00 for the days of the change from summer to winter time.

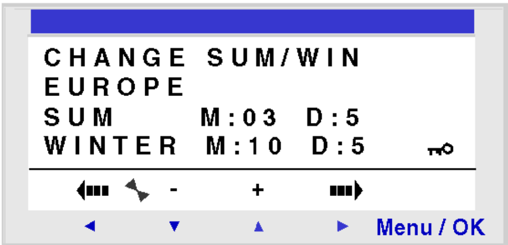
Chapter 16
CHANGE SUMMER/WINTER Menu

CHANGE SUMMER/WINTER Menu

Description

This function is used to change the time range automatically: Summer/winter, for smart relays with a clock.

Illustration:



The following operating modes are possible:

- **NO**: no change,
- **Automatic**: The change takes place automatically, the dates are preset according to the geographic zone:
  - EUROPE: Europe,
  - USA.
- **OTHER ZONE**: (MANUAL) the change takes place automatically, but you must specify, for summer and winter:
  - The month: **M**,
  - The Sunday: **D** (1, 2, 3, 4 or 5) when the change takes place.

Configuration of the Time Change

To configure automatic time change, proceed as follows:

Step	Action
1	Select the parameter to modify using the navigation keys ◄ and ►. **Result**: The selected parameter flashes.
2	Modify the parameter value. The **+** and **-** keys of the contextual menu are used to change the current value.
3	Confirm the changes by pressing the **Menu/OK** key. **Result**: The display returns to the MAIN menu.

Part IV
LD Language

Subject of this Section

This section describes the use of LD (Ladder Diagram) programming language for the smart relay.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
17	Overview of LD language	123
18	Programming in Ladder using Zelio Soft 2	133
19	LD Language Elements	143
20	Programming Ladder Using Zelio Soft 2	211
21	Example of an LD Application	237

Chapter 17
Overview of LD language

Subject of this Chapter

This chapter provides a general description of LD language.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
General Overview of Ladder Language	124
Structure of a Ladder Network	125
Ladder Network Comment	127
Ladder Language Graphic Elements	128
Programming Rules for a Ladder Network	130

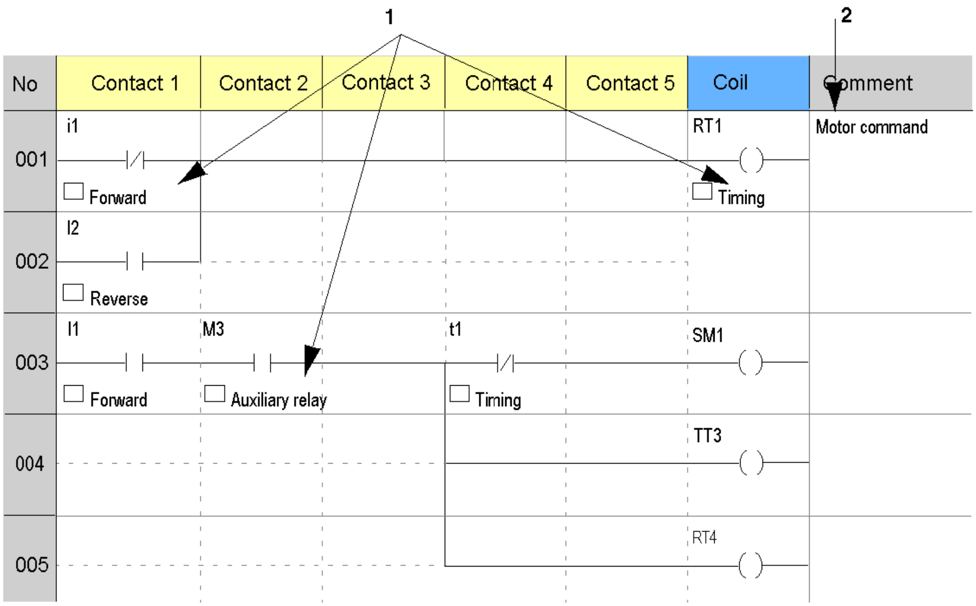
General Overview of Ladder Language

General

A section of program written in Ladder Language is made up of a series of ladder networks executed by the smart relay.

Ladder Network Illustration

The following diagram shows a Zelio Soft 2 wiring sheet in LADDER data entry mode:



Number	Element	Function
1	Graphic elements	These represent: ● The inputs/outputs of the smart relay (push-buttons, sensors, relays, LEDs, etc.), ● Automation functions (timers, counters, etc.), ● Logic operations, ● Internal variables (auxiliary relays) of the smart relay.
2	Comments	For each line of a ladder network (optional).

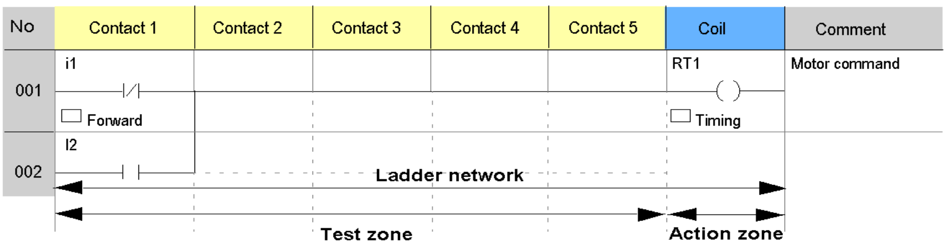
Structure of a Ladder Network

Introduction

The ladder network is between the first "contact" column (**Ladder 1**) and the "**coil**" column.

Illustration

The following diagram describes the structure of a ladder network.



Ladder Network Description

A ladder network is made up of a collection of graphic elements set out over a wiring sheet. The maximum number of program lines depends on two factors:

- the firmware version *(see page 540)*
- whether an SR2COM01 communication interface has been selected in the configuration.

Each program line comprises of a maximum of 5 contacts and a coil.

It is divided into two zones:

- The **test zone**, in which the conditions necessary for triggering an action (contacts) are displayed,
- The **action zone**, which applies the result following a logical test combination (coils).

Definition of an Action

An action is applied to an automation function (timer, counter, etc.), an auxiliary relay or a physical smart relay output.

An action causes a change in status of the specified associated function. For example:

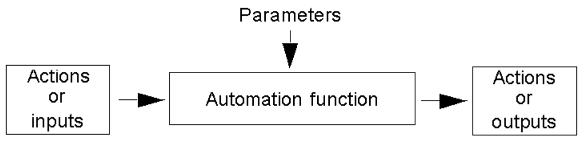
- An RT1 *(see page 182)* action causes a reset of the T1 timer,
- An SM1 *(see page 151)* action causes a set of the M1 auxiliary relay.

Definition of an Automation Function

An automation function (timer, counter, auxiliary relay, etc.) is defined by:

- Input data or actions,
- Output data or states,
- Adjustment parameters.

The following diagram shows the structure of a function:



Ladder Network Comment

General

A comment, though not mandatory, is assigned to a portion of the network and makes it easier to interpret.

Comment Associated With An LD Line

The comment is integrated into the network at the end of the line, in the **Comment** column, and is made up of a maximum of 192 characters.

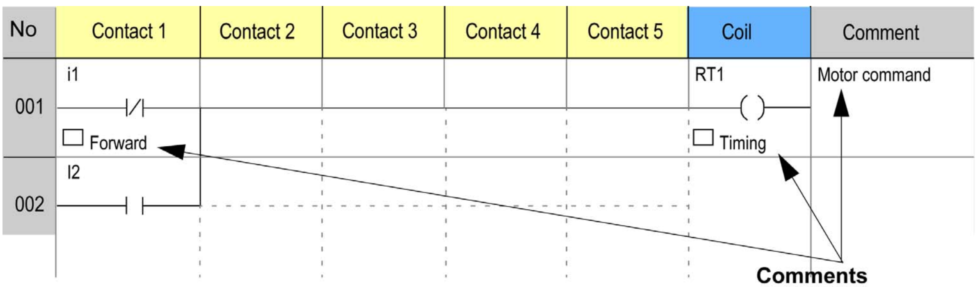
NOTE: Press **Ctrl+Enter** to insert a new line.

Comment Associated with a Graphic Element

The comment is integrated into the network below the associated graphic element (contact or coil).

NOTE: For printing and display reasons, it is advisable not to exceed 25 characters as they will be truncated in both display and printout.

Illustration



NOTE: For the column comment associated with a line, the maximum number of characters displayed varies according to the size of your editing window. The number 192 is provided for a full screen.

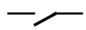
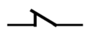
Ladder Language Graphic Elements

General

The graphic elements are ladder language instructions. Used together and arranged on a wiring sheet, they form a ladder network. The logic of the network is solved, leading to an output result. One or more networks then constitute the application program.

Contacts

Graphic elements referred to as contacts are placed in the test zone and take up one cell (one row high by one column wide of the wiring sheet).

Name	Ladder symbol	Electrical symbol	Functions
Normally open contact	—| |—		Conducting contact when its controlling input (switch, sensor, etc.) is active.
Normally closed contact	—|/|—		Conducting contact when its controlling input is inactive.

Linking Elements

Linking graphic elements are used to connect test and action graphic elements.

Name	Graphic representation	Functions
Horizontal connection	———	Used to link test and action graphic elements together between the two potential bars on each side of the wiring sheet.
Vertical connection	|	Used to link test and action graphic elements in parallel.

A horizontal connection represents a logical **AND**; it sends the state of the contact located immediately to its left to the contact located immediately to its right.

A vertical connection represents the logical **OR** of the **active** states of the horizontal connections located to its left, i.e.:

- Inactive if the states of the horizontal contacts located to the left are inactive,
- Active if at least one of the horizontal contacts located to the left is active.

Coils

The graphic elements referred to as coils are placed in the action zone and take up one cell (one row high by one column wide of the wiring sheet).

Name	Ladder symbol	Electrical symbol	Functions
Direct coil	[—( )—	[—□—	The coil is energized if the contacts to which it is connected are conducting (contact mode).
Impulse coil	ʃ —( )—	ʃ —□—	The coil is energized if the contacts to which it is connected change state (impulse relay mode).
Set or latch coil	S —( )—	S —□—	The coil is energized once the contacts to which it is connected are conducting, then stays energized even if later the contacts are no longer conducting (SET mode).
Reset or unlatch coil	R —( )—	R —□—	The coil is de-energized when the contacts to which it is connected are conducting. It remains inactive even if later the contacts are no longer conducting (RESET mode).

NOTE: For reasons of upward compatibility for the programs operating with Zelio 1, the four types of function for any given Q output coil or M auxiliary relay can be used in the same wiring sheet in Zelio 2.

Programming Rules for a Ladder Network

General

Ladder networks are programmed using graphic elements, observing the following programming rules.

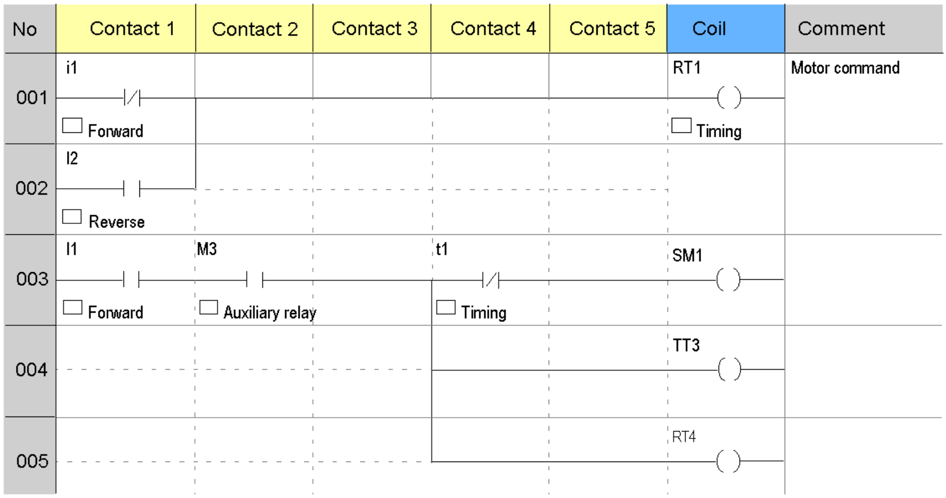
Programming Rules

The programming of a ladder network must obey the following rules:

- Test and action graphic elements each occupy a cell within a network.
- The ladder networks end with at least one action (such as a coil) in the final column.
- Actions are located in the last column.
- A coil corresponds to the triggering of an action assigned to an automation function (timer, counter, auxiliary relay, physical smart relay output, etc.).
- The status of an automation function can be used as a test (contact). The contact then takes on the name of the associated function, e.g.:
  - T1 represents the status of the **T1** timer *(see page 183)*.
  - t1 represents the complementary status of the **T1** timer.
- Links are read (interpreted) from left to right.
- If, in a network, you use an **S** (Set) action for an automation function (output, auxiliary relay, etc.), generally use also an **R** (Reset) action for the same function.
  Exception: An **S** action is used without an **R** action for detecting operating anomalies that can only be reset on receiving a RESET-INIT action from the program.
- The **R** (Reset) actions of an automation function take priority over **S** (Set) actions applied to the same function at the same moment.
- Network tests combine in the same way as an electrical voltage circuit from the left-hand network column (+V) to the right-hand network column (+0v).
- No perpendicular connection is possible between lines 120 and 121.

Example of a Ladder Wiring Sheet

The following screen shows an example of ladder wiring sheet.



Chapter 18
Programming in Ladder using Zelio Soft 2

Subject of this Chapter

This chapter describes simplified examples of the different types of programming in ladder mode.
A detailed description of the programming types is provided in chapters:

- Functions Accessible from the Front Panel *(see page 71)*,
- Programming from the Programming Software *(see page 211)*.

In ladder extended mode (240 lines), there are two parts. Part1 (line 1 to 120) and Part2 (line 121 to 240) are independent. The editor does not allow connection from line 120 to 121.

What Is in This Chapter?

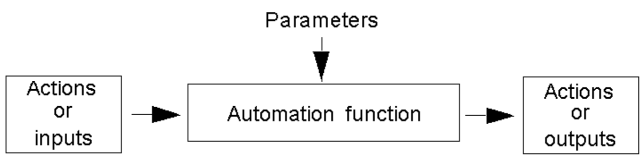
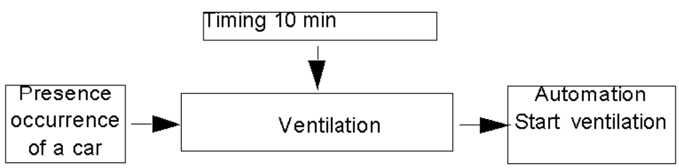
This chapter contains the following topics:

Topic	Page
Structuring LD Programs	134
Zelio Entry Mode	135
Ladder Entry Mode	136
Configuration Mode	138
Text Entry Mode	139

Structuring LD Programs

Procedure

The following table describes the phases in creating an LD program:

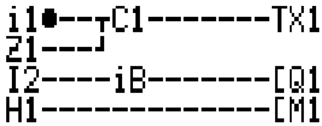
Phase	Description
1	**List the Inputs/Outputs and enter any associated comments.** Example: ● Presence of car, ● Limit-switch, ● Actuator control.
2	**List the automation functions to be performed.** Example: ● Counting of cars, ● Daily programming.
3	**Perform each function taking into account:** ● The input data, ● The output data, ● The adjustment parameters (threshold). The following diagram shows the structure of a function: 
4	**Comment each function.** Example: 
5	**Test each function using the simulation tool:** ● An output is generally activated at a single location in a program. ● A Set output must be accompanied by a RESET output. ● Check the RESET inputs for the Timer, Counter and Text functions.

Zelio Entry Mode

Description

By using the software in **Zelio Entry** mode, it is possible to construct an LD network in the same way as you would by using the buttons on the Zelio front panel.

Illustration



This approach is designed for those with experience in programming directly on smart relays.

This mode can be used to configure, program and control the application using the keys on the front panel *(see page 76)*:

- **Z Keys** ←↑↓→: These keys (in gray), in a row from left (Z1) to right (Z4), are located below the LCD.
  When the keys can be used for other actions than navigation, a contextual menu bar is displayed at the bottom of the screen (if the **Shift** key is pressed).
- **Menu / Ok**: This key (in green) is used for confirmation of menu, sub-menu, program, parameter, etc.
- **Shift**: This key (in white) is used to display a contextual menu above the other buttons.

NOTE: When the cursor is over a modifiable parameter, a contextual menu appears (if the **Shift** key is pressed).

Programming

In this mode, the programming characteristics are displayed on the front panel in **PROGRAM** *(see page 81)* mode.

The front panel of the smart relay is simulated; programming is possible using the buttons displayed in the window.

When you enter this mode, the start of the program is displayed.

From then on, a flashing square appears on the first box to show that it is possible to insert or modify a character.

The four navigation buttons can be used to move the flashing cursor over the boxes of the LCD.

When the cursor is moved on a line, flashing zones appear:

- Squares that show that it is possible to enter contacts and a coil at the end of the line.
- Circles that show that it is possible to enter horizontal and vertical connections.

NOTE: When the cursor is over a modifiable parameter, a contextual menu appears (if the **Shift** key is pressed).

Ladder Entry Mode

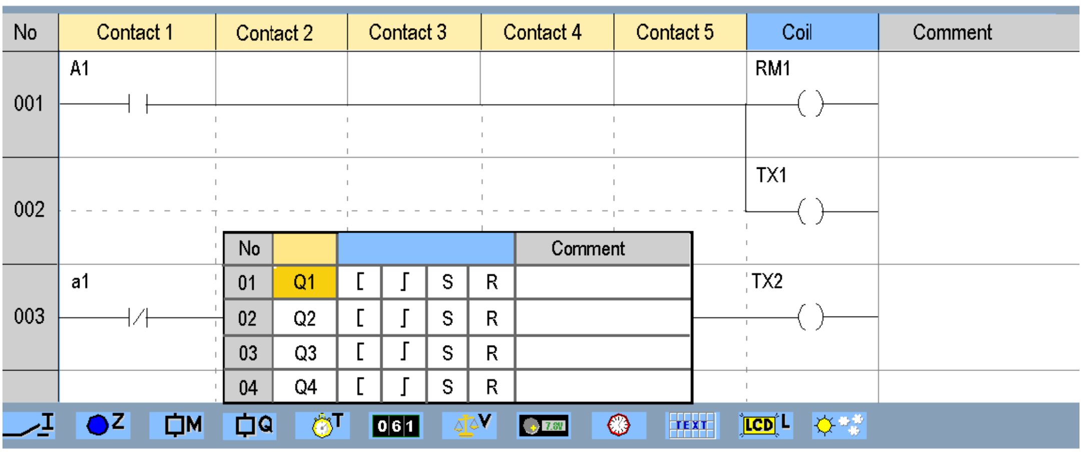
Description

Using the Zelio Soft 2 Programming software in **Ladder Entry** mode allows you to adopt a software-based approach to programming:

- Use of toolbars,
- Creation of the application by dragging and dropping automation functions,
- Use of parameters windows,
- Overall visibility of the application.

With this mode, the workspace is made up of a wiring sheet to which the various automation functions are added.

Illustration:



This mode provides extra functionalities over the Zelio entry mode:

- Choice of symbol types: Ladder / Electrical,
- Possibility of adding comments to each programming line.

Program

Toolbars

Programming is performed on a wiring sheet. The automation functions available appear in the toolbar located at the bottom.

Wiring

Once the function type is selected, the list of available functions appears in the form of a set of lines, in which each line contains the following function elements:

- List of its outputs (or states),
- List of its inputs (or actions),
- Comment associated with the function.

Click once on an output and drag it to a coil cell, or on an input and drag it to a contact cell.

Parameters

For functions that contain parameters (counters, clocks, auxiliary relays, etc.), to open the parameters window:

- On the wiring sheet: Double-click on the element,
- Use the Settings mode *(see page 138)*.

Connections

Connections are made by clicking on the horizontal and vertical connections pre-drawn with a dotted line on the wiring sheet.

Symbols

You can select the type of symbols used in the wiring sheet (ladder, electrical) from the **Display** menu.

Comments

To associate a comment with a variable:

- On the wiring sheet: Double-click on the element in the wiring sheet, select the **Comment** tab and enter the text in the entry zone.
- From the variable selection table: Double-click in the Comment column (on the line corresponding to the element) and enter the text.
- in Text Entry mode *(see page 139)*, enter the text.

When a comment is associated with a variable, it appears in the cell under the variable. If the comment is hidden, an envelope icon is displayed  .

Configuration Mode

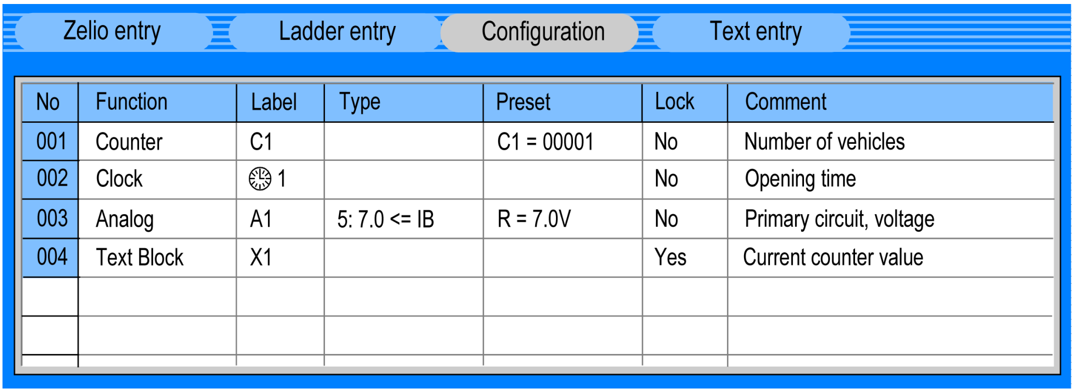
Description

Configuration mode enables you to list the automation functions with parameters used in the application. This mode is accessible from edit mode by clicking the Configuration tab.

The general interface allows to view the information:

- **Function**: Timer, Counter, etc.
- **Label**: Function block ID.
- **Type**: Counter type, Timer type, etc.
- **Preset**: The value to reach for a timer, counter or other function with a preset value.
- **Lock**: Lock the parameters (prevent modification via the front panel).
- **Comment**: Comments associated with the function.

Illustration:



No	Function	Label	Type	Preset	Lock	Comment
001	Counter	C1		C1 = 00001	No	Number of vehicles
002	Clock	1			No	Opening time
003	Analog	A1	5: 7.0 <= IB	R = 7.0V	No	Primary circuit, voltage
004	Text Block	X1			Yes	Current counter value

To adjust the various parameters, double-click on the line.

The configurable automation functions are:

- Clocks *(see page 176)*
- Analog Comparators *(see page 192)*
- Timers *(see page 182)*
- Counters *(see page 157)*
- Texts *(see page 198)*

Text Entry Mode

Description

The **Text entry** interface allows you to have an overall view of the inputs/outputs used in the application, and is intended to allow you to enter/edit comments for each variable object.

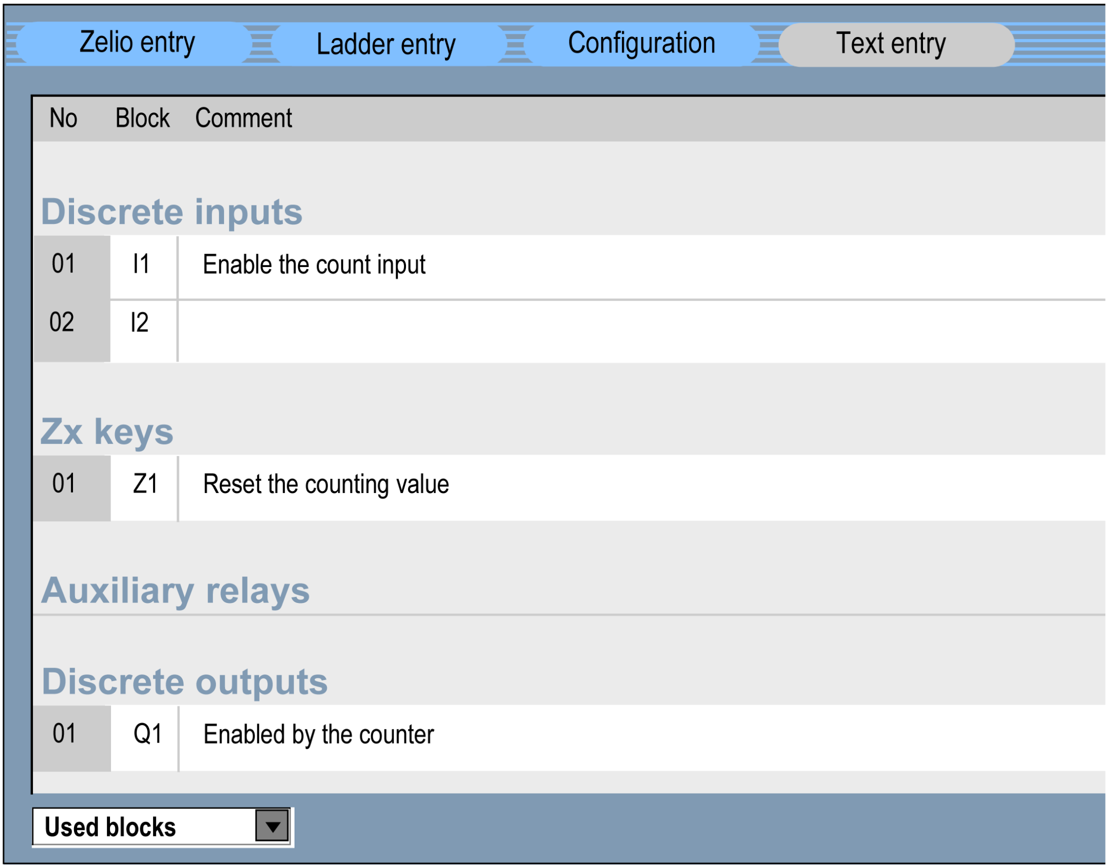
It is important to identify each variable with an explicit comment in order to make the application as clear as possible.

This mode applies to:

- Discrete inputs,
- Zx keys,
- Discrete outputs,
- Auxiliary relays,
- Timers,
- Counters,
- Fast counters,
- Counter comparators,
- Analog comparators,
- Clocks,
- Texts,
- LCD backlighting,
- Summer Winter.

You can choose to display in the window only the blocks that are used, using the drop-down menu below the edit zone.

Illustration:



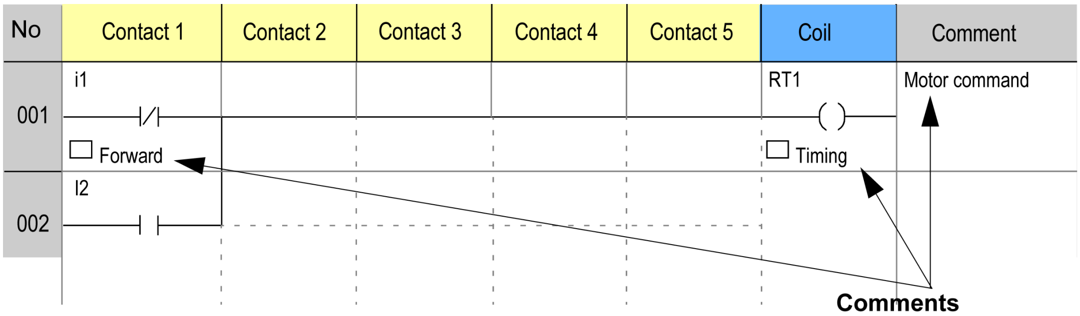
Entering and Modifying Comments

Comments will be displayed under the contact or coil when viewing the program in ladder entry mode.

The comment is integrated into the network below the associated graphic element (contact or coil).

NOTE: For reasons relating to display and printing, it is advised to not exceed 25 characters as they will be truncated in both display and printout.

Illustration:



In **Text Entry** mode, you can double click on the comment zone, to enter and validate a comment.

Chapter 19
LD Language Elements

Subject of this Chapter

This chapter describes the different automation functions of the LD language.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Discrete Outputs	144
Discrete Inputs	147
Modbus Inputs/Outputs	149
Auxiliary Relays	151
Zx Keys	155
Counters	157
Counter Comparators	164
Fast Counter	166
Clocks	176
Change to Summer / Winter Time	180
Timers	182
Analog Comparators	192
TEXT	198
LCD Screen Backlighting	202
Message	203

Discrete Outputs

Description

Discrete Outputs correspond to the smart relay outputs (connected to the actuators). These outputs are numbered from Q1 to Q9, then from QA to QG, according to the smart relay reference and the connected extensions.

Discrete outputs can be used either with a coil (write) or a contact (read) element.

Access

This  function is accessible from the **LD** tool bar.

Use as a Coil

To use a Discrete output as a coil, four types are available:

- Direct coil
- Impulse coil
- Set (latch) coil
- Reset (Unlatch) coil

Direct coil:

Symbol of a Discrete output, used as a Direct coil:

[Q-

The coil is energized if the elements to which it is connected are conducting. Otherwise it is not energized.

Impulse coil:

Symbol of a Discrete output, used as an Impulse coil:

ʃQ-

Pulse energization, the coil changes state on the rising edge of each pulse it receives.

<u>Example:</u> Switching a lamp on and off with a pushbutton:

i1 ——ʃQ1

A push button is connected to input **I1** and a lamp to output **Q1**. Every time the button is pressed, the lamp switches on or off.

Set coil:

Symbol of a Discrete output, used as a Set coil:

SQ-

The Set coil is energized as soon as the elements to which it is connected are conducting, then stays energized even if afterward the elements are no longer conducting.

Reset coil:

Symbol of a Discrete output, used as a Reset coil:

RQ-

The **RESET** coil is deactivated when the elements to which it is connected are conducting. It remains inactive even if afterward the elements are no longer conducting.

Example: Switching a lamp on and off with two pushbuttons:

```
I1 ——— SQ1
I2 ——— RQ1
```

In this example, push button 1 (PB1) is connected to input **I1**. PBI2 to input **I2**. The lamp is controlled by output Q1. The lamp illuminates when pushbutton PBI1 is pressed, and it turns off when pushbutton PBI2 is pressed.

NOTE:
- Generally, an output is only used at one single point in the program as a coil (given the exception of the Set and Reset coils).
- If a SET coil is used for a Discrete output, provide a RESET coil for this output. The **RESET** coil takes priority over the **SET** coil.
  The use of a Set coil on its own is only justified for activating an alarm signal that can be reset only by an INIT + ON action from the program.

Use as a Contact

An output can be used as a contact as many times as necessary.

This contact may use the direct state of the output (normally open contact) or its inverse state (normally closed contact), see below.

Normally open:

Symbol of a Discrete output, used as a contact in normally open:

Q-

If the output is energized, the contact is conducting.

Normally closed:

Symbol of a Discrete output, used as a contact in normally closed:

q-

If the output is energized, the contact is non-conducting.

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a coil or a contact, position the cursor on the element then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):

- Position the cursor on the symbol representing the coil output and press **Shift** key,
- Scroll through the possible coil or contact types (**Q** for a normally open contact, **q** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of contacts on program initialization:

- Normally open (direct state) is inactive,
- Normally closed (reverse state) is active.

Latching

By default, after a power outage, the outputs are in the state that corresponds to program initialization.

Activate latching to restore the state of the output as backed up before the power outage:

- From the front panel: from the **PARAMETERS** menu *(see page 91)*, or
- In the programming software: Enable the **Latching** option in the parameters window associated with the output.

Discrete Inputs

Description

The **Discrete Inputs** can be used exclusively as contacts in the program.

These contacts represent the status of the input for the smart relay connected to a sensor (push button, switch, sensor, etc.).

The contact number corresponds to the number of terminals of the associated input: 1 to 9, then A to R (except for letters I, M and O) according to the smart relay and the possible extension.

Access

This  function is accessible from the **LD** function bar.

Use as a Contact

This contact may use the direct state of the input (normally open contact) or its inverse state (normally closed contact), see below.

Normally open:

Symbol of a normally open contact:

I -

If the input is **supplied**, the contact is **conducting**.

Example:

I1 ——ʃ Q1

If input **1** is supplied, contact **I1** is closed, and coil **Q1** is activated.

Normally closed:

Symbol of a normally closed contact:

i -

If the input is **supplied**, the contact is **non-conducting**.

Example:

i1 —— [Q1

If input **1** is supplied, contact **i1** is open, and coil **Q1** is not activated.

Modification of the state of a contact

In the programming software, to modify the state of a contact position the mouse on it, then:
- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: Scroll through the possible states.

To modify a contact from the front panel of the smart relay (the programming window is displayed on the screen):
- Place the cursor over the letter of the contact,
- Scroll through the possible contact types (**I** for a normally open contact, **i** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of contacts on program initialization:

- The direct state is inactive,
- The reverse state is active.

Modbus Inputs/Outputs

Description

A Modbus **SR3MBU01BD** extension module may be added onto an extensible smart relay.

In **LD** mode, the application cannot access the four 16-bit data exchange words. Data transfer between master and slave is implicit and transparent.

NOTE: The Modbus module only operates as Modbus slave.

Parameters

The Modbus module can be configured only in the programming software.

To access the parameters of the Modbus module, proceed as follows:

Step	Action
1	• Click on **Edit → Program Configuration** command, or • Click on the Program Configuration icon:  **Result**: The program configuration window is displayed.
2	Click on the **Modbus Extension** tab.

When changing to RUN mode, the smart relay initializes the Modbus module.

The Modbus module has 4 parameters:

- The number of wires and frame format on the Modbus network,
- The data transmission speed in bauds,
- The protocol parity,
- The Slave Modbus extension network address.

Words to be sent to master

Writing these words to the master is automatically performed by duplication of the status of the discrete I/Os as follows:

																Modbus Address (Hexa)
IG	IF	IE	ID	IC	IB	IA	I9	I8	I7	I6	I5	I4	I3	I2	I1	0x0014
0	0	0	0	0	0	0	0	IR	IQ	IP	IN	IL	IK	IJ	IH	0x0015
0	0	0	0	0	0	QA	Q9	Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	0x0016
0	0	0	0	0	0	0	0	0	0	QG	QF	QE	QD	QC	QB	0x0017

Most significant byte **Least significant byte**

I1 to IG: discrete input states for the SR3B261BD base.

IH to IR: discrete input states for the SR3XT141BD extension.

Q1 to QA: discrete output states for the SR3B261BD base.

QB to QG: discrete output states for the SR3XT141BD extension.

Words sent by the master

The words sent by the master are not implicitly operated on by the smart relay.

These 4 16-bit words have the following addresses (Hexa): 0x0010 / 0x0011 / 0x0012 / 0x0013.

Auxiliary Relays

Description

Auxiliary relays marked **M** or **N** behave as Discrete Outputs **Q** *(see page 144)*, but do not have an electrical output contact. They can be used as internal variables.

NOTE: The maximum number of auxiliary relays depend on the firmware version and whether an SR2COM01 is included in the configuration *(see page 540)*.

There are 28 **M** auxiliary relays, numbered from M1...M9, and then MA...MV, excluding MI, MM, and MO.

In addition, if no SR2COM01 communication interface has been selected in the configuration, there are 28 **N** auxiliary relays, numbered from N1...N9, and then NA...NV, excluding NI, NM, and NO.

The auxiliary relays can be used in the program, indifferently either as a coil or contact. They can be used to latch a state to be used in the form of the associated contact.

Access

The  function is accessible from the **LD** function bar.

Use as a Coil

To use an auxiliary relay as a coil, 4 types are available:
- Direct coil
- Impulse coil
- Set (latch) coil
- Reset (Unlatch) coil

Direct coil:

Symbol of an auxiliary relay used as a Direct coil:

[M-

The relay is energized if the elements to which it is connected are conducting. Otherwise it is not energized.

Impulse coil:

Symbol of an auxiliary relay used as an Impulse coil:

ʃM-

Pulse energization, the coil changes state on each rising edge it receives.

Set coil:

Symbol of an auxiliary relay used as a Set coil:

SM-

The **SET** coil is energized as soon as the elements to which it is connected are conducting, then stays energized even if afterward the elements are no longer conducting.

Reset coil:

Symbol of an auxiliary relay used as a Reset coil:

RM-

The **RESET** coil is deactivated when the elements to which it is connected are conducting. It remains deactivated even if afterward the elements are no longer conducting.

NOTE: For upward compatibility for programs operating with Zelio 1, the four types of a given output coil (Q) or auxiliary relay (M) can be used in the same wiring sheet in Zelio 2.

Use as a Contact

Auxiliary relays can be used as contacts as many times as necessary.

This contact may use the direct state of the relay (normally open contact) or its inverse state (normally closed contact), see below.

Normally open:

Symbol of an auxiliary relay used as a contact in normally open:

M-

If the relay is **energized**, the contact is **conducting**.

Normally closed:

Symbol of an auxiliary relay used as a contact in normally closed:

m-

If the relay is **energized**, the contact is **non-conducting**.

Example

In the following example turning a lamp on and off is conditioned by the state of the 6 following inputs: I1, I2, I3, I4, I5, and IB.

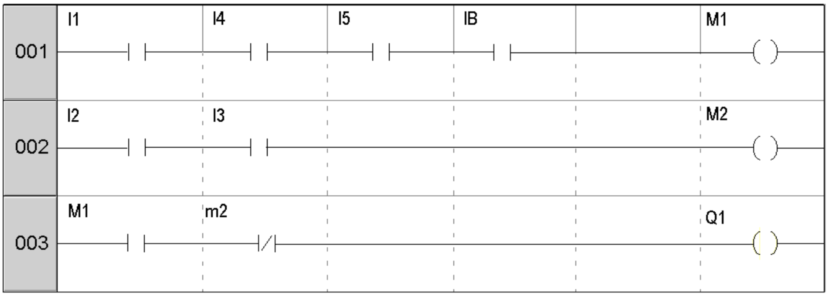
The lamp is on when:

- Inputs I1, I4, I5, and IB are set to 1, and
- Inputs I2 and I3 are set to 0

As the smart relay does not allow more than five contacts on a line, auxiliary relays are used to control the lamp.

We have chosen to latch inputs I1, I4, I5, and IB using auxiliary relay M1 and to latch inputs I2 and I3 using auxiliary relay M2. The lamp is controlled by relays M1 and M2, which are used as a normally open contact and a normally closed contact respectively.

Illustration:



Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a coil or a contact, position the mouse on the element then:

- With the mouse: Right-click to display a list of possible states (click to validate).
- With the space bar: Scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (with the programming window displayed on screen):

- Position the cursor on the symbol representing the coil type or on the letter of the contact.
- Scroll through the possible coil or contact types (**M** for a normally open contact, **m** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

State of contacts on program initialization:

- Normally open (direct state) is inactive.
- Normally closed (reverse state) is active.

Latching

By default, after a power outage, the relay is in the state that corresponds to program initialization.

To restore the state of the output as backed up before the power outage, latching must be activated:

- From the front panel: From the **PARAMETERS** *(see page 91)* menu, or
- In the programming software: Enable the **Latching** option in the parameters window associated with the relay.

Zx Keys

Description

The navigation keys behave like the **I** physical inputs (Discrete inputs). The only difference is that they do not correspond to smart relay connection terminals, but to the four gray keys on the front panel.

They are used as pushbuttons, and can only be used as contacts.

Access

This  function is accessible from the **LD** function bar.

Use as a Contact

This contact may use the direct state of the key (normally open contact) or its inverse state (normally closed contact), see below.

Normally open:

Symbol of the normally open contact, representing a key:

Z-

If the key is **pressed**, the corresponding input is **conducting**.

Normally closed:

Symbol of the normally closed contact, representing a key:

z-

If the key is **pressed**, the corresponding input is **non-conducting**.

Example

Creating a switch operated by the **Z1** key and **Q1** output:

Z1 ——ʃQ1

Each time the Z1 key is pressed, the Q1 output changes state.

Deactivation of Zx Keys

By default the **Zx Keys** are active. They may be deactivated as described here:

- From the smart relay front panel: Using **CONFIGURATION → Zx KEYS** menu, refer to Zx KEYS *(see page 102)*
- From the programming software:
  - in the **Edit → Program Configuration**, Configuration tab *(see page 504)*, by enabling the **Zx keys inactive** option
  - in the **Write options** window *(see page 510)*, by enabling the **Zx keys inactive** option

NOTE: When the smart relay is in RUN mode, if the **Zx Keys** have been deactivated, they cannot be used for inputs in the program, but for navigating the menus.

Modification of the State of a Contact

In the programming software, to modify the state of a contact position the mouse on it, then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: Scroll through the possible states.

To modify a contact from the front panel of the smart relay (the programming window is displayed on the screen):

- Place the mouse over the letter representing the contact,
- Scroll through the possible contact types (**Z** for a normally open contact, **z** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of contacts on program initialization:

- Normally open (direct state) is inactive,
- Normally closed (reverse state) is active.

Counters

Description

The **Counters** function is used to upcount or downcount pulses. The smart relay has either 28 counters, or 16 counters if an SR2COM01 communication interface has been selected in the configuration. They are numbered from 1...9 then from A...V (I, M, O are not used).

NOTE: The maximum number of counters also depends on the firmware version *(see page 540)*.

The **Counters** function can be reset to zero or to the preset value (depending on the chosen parameter) during use.

It may be used as a contact to find out whether:

- The preset value has been reached (upcounting **TO**),
- The counter has reached 0 (downcounting **FROM**).

Access

This  function is accessible from the **LD** function bar.

Use of Coils

Each timer has 3 associated coils:

- Coil **CC**: **Counting Pulse Input**,
- Coil **RC**: **Reset Initial Counter State Input**,
- Coil **DC**: **Counting Direction input**.

The use of these coils is described below.

Counting pulse input:

Symbol of the Counting Pulse Input coil of a timer:

CC-

When used as a coil in a control diagram, this element represents a counting input for the function. Every time the coil is energized, the counter is incremented or decremented by 1 according to the counting direction chosen.

Example: Input counting pulses delivered by counter no. 1.

I1 ——— CC1

Every time input I1 is energized, the counter no. 1 is incremented by 1.

Reset Initial Counter State input:

Symbol of the Reset Initial Counter State Input coil:

RC-

When used as a coil in a control diagram, this element represents an input that resets the counting function to its initial state.

Energizing the coil has the following effect:

- Reset the count value to **zero** if the count type is **TO** (upcounting to the preset value),
- Reset the value to the **preset value** if the count type is **FROM** (downcounting from the preset value).

Example: Counter no. 1 reset to zero by pressing Z1 key.

Z1 ——— RC1

Every time key Z1 is pressed, the counter starts from 0.

Counting direction input:

Symbol of the Counting Direction Input coil of a timer:

DC-

This input determines the counting direction according to its status:

- It downcounts if the coil is energized,
- It upcounts if the coil is not energized.

NOTE: By default, if this input is not wired, the function upcounts.

Example: Up/downcounts, depending on the status of smart relay input I2.

I2 ——— DC1

If the **I2** input is active, the function downcounts.

Use as a Contact

The contact associated with the counter indicates whether the preset value (**TO**) or zero (**FROM**) has been reached.

It may be used as many times as necessary in the program either as normally open or as normally closed:

Normally open:

Symbol of the normally open contact associated with a counter:

C-

The contact is **conducting when**:

- The counter value **has reached** the preset value, if the counter is upcounting (**TO**).
- The counter value **is equal to 0**, if the counter is downcounting (**FROM**).

Normally closed:

Symbol of the normally closed contact associated with a counter:

c-

The contact is **conducting as long as**:

- The counter value **has not reached** the preset value, if the counter is upcounting (**TO**).
- The counter value **is not equal to 0** if the counter is downcounting (**FROM**).

Example: Lighting a LED connected to counter no. 1 output (**TO**).

```
C1 ——— [ Q1
```

When the preset value has been reached: The LED is illuminated; otherwise it is off.

Settings from the Software

Pulses:

This value is between 0 and 32767 (preset value).

Type of counting:

Two settings are available:

- Upcounting to the preset value: Incrementation of the count value,
- Downcounting from the preset value: Decrementation of the count value.

Latching:

By default, after a power outage, the counter is set to the state that corresponds to program initialization.

To restore the state of the counter backed up on power outage, activate latching.

Locked:

Locking prevents locked parameters from being modified from the front panel of the smart relay using the PARAMETERS menu.

Configuration from Front Panel

The block parameter settings can be accessed either when entering the command line or from the **PARAMETERS** menu if the block has not been padlocked.

The parameters to enter are the following:
- Type of counting,
- Preset value,
- Parameter lock,
- Latching.

Type of counting:

Symbol of the type of counting parameter:

T

This parameter is used to select the type of the counter:

- **TO**: upcounting towards the preset value.
  When the counter value is equal to the preset value, contact C of the counter is conducting.
- **FROM**: downcounting from the preset value.
  When the counter value equals 0, counter contact C is conducting.

Preset value:

Symbol of the preset value parameter:

P

This value is between 0 and 32,767, and represents:

- The value to reach when counting to the preset value (**TO**),
- The initial value when downcounting from the preset value (**FROM**).

Parameter lock:

Symbol of the Parameter lock parameter:

Locked Unlocked

Locking prevents the modification of locked parameters from the front panel of the smart relay via the PARAMETERS menu.

Latching:

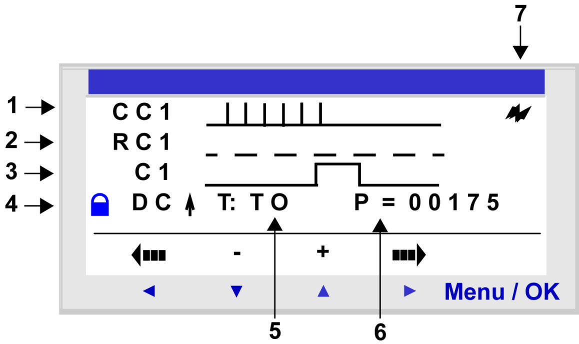
Symbol of the Latching parameter:

Active Inactive

This function is used to save the status of the counter values in the event of a power outage.

In Zelio entry mode, latching will only be activated if the symbol is displayed on the parameter screen.

Illustration: Configuring a counter from the front panel of the smart relay:



Description:

Number	Parameter	Description
1	**Command input**	Control input timing diagram (following pulse).
2	**Reset input**	Counter reset input timing diagram.
3	Counter output	Counter output timing diagram.
4	**Parameter lock** 	This parameter is used to lock the counter parameters. When the block is locked, the preset value no longer appears in the PARAMETERS menu.
5	**Type of counting**	**TO**: upcounting towards the preset value or, **FROM**: downcounting from the preset value.
6	**Preset value**	Counter preset value.
7	**Latching**	Backup of the counter value.

Counter Value

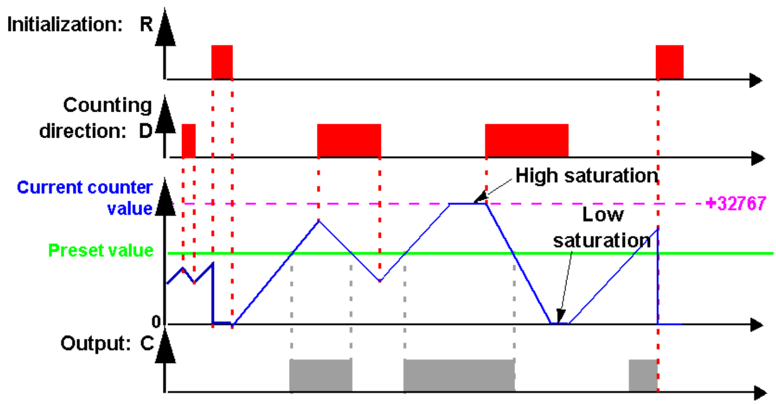
The counter value is the value at a given time resulting from the successive up/down counting actions that have occurred since the last time the counter was reset to its initial state.

This value is between 0 and 32767. Once these limits have been reached, a downcount will leave the value 0 and an upcount will leave the value at + 32767.

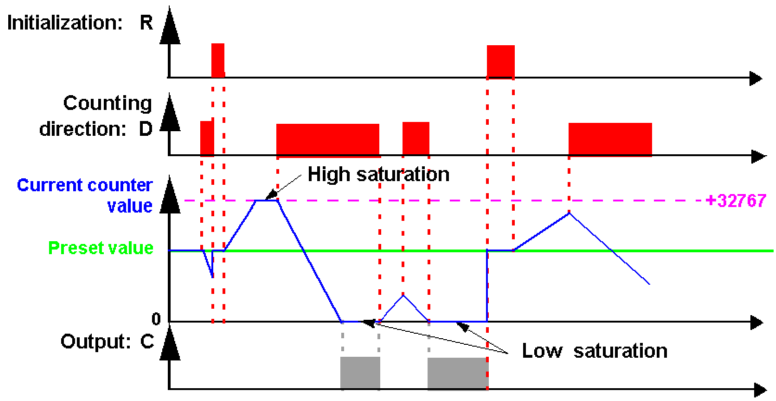
Timing Diagrams

In the timing diagrams, the blue curves represent the value of the counter:

The following figure shows the operation of the counter when upcounting (**TO**) toward the preset value:



The following figure shows the operation of the counter when downcounting (**FROM**) from the preset value:



Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a coil or a contact, position the mouse on the element then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):

- Position the cursor on the symbol representing the coil type or on the letter of the contact,
- Scroll through the possible coil or contact types (**C** for a normally open contact, **c** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of the contacts and value on initialization of the program:

- The **normally open** (direct state) is **inactive**,
- The **normally closed** (inverse state) is **active**,
- The **value** is **zero**.

Examples

Below, three examples of the use of a counter:

Screen	Description
`I1-------------CC1` `I2-------------RC1`	**Upcounting and zero resetting:** The counter is incremented each time input **I1** is activated. The counter is reset each time input **I2** is activated.

`I1-------------CC1`
`               DC1`
`I2-------------RC1`
	Downcounting and Resetting:
	The counter is decremented each time input **I1** is activated.
	The counter is reset each time the **I2** input is activated.

`I1┬------------CC1`
`I3┘`
`I3-------------DC1`
`I2-------------RC1`
	Upcounting, Downcounting and Resetting:
	The counter is incremented each time input **I1** is activated.
	The counter is decremented each time the **I3** input is activated.
	The counter is reset each time the input **I2** is activated.

Counter Comparators

Description

This function is used to compare the values of two counters or of a counter and a constant value.

NOTE: The **Counter Comparators** function block can only be configured from the programming software in **Ladder Entry**.

Access

This  function is accessible from the **LD** function bar.

Use as a Contact

The counter comparator indicates whether the chosen condition is verified. It is used as a contact, in normally open or in normally closed.

Normally open:

Symbol of the counter comparator, in normally open:

V1

The contact is **conducting** when the condition is **verified**.

Normally closed:

Symbol of the counter comparator, in normally closed:

v1

The contact is **conducting** when the condition **is not verified**.

Settings from the Software

The different parameters to fill in are the following:
- Comparison formula,
- Parameter lock.

Comparison formula:

The comparison formula is the following:

Cx + x <Comparison Operator> Cy + y

Where:
- **Cx and Cy**: Represent the counters to compare; these are selected using the associated drop-down menu,
- **x and y**: These are constants (offset) between: - 32,768 and 32,767.

The **comparison operators** that may be chosen are the following:

Symbol	Description
>	Greater than.
≥	Greater than or equal to.
=	Equal to.
≠	Different.
≤	Less than or equal to.
<	Less than.

Parameter lock:

Locking prevents locked parameters from being modified from the front panel of the smart relay using the PARAMETER menu.

Configuration from the Front Panel

The **Counter Comparators** function block cannot be configured from the front panel of the smart relay. This function must be configured from the programming software.

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a contact, position the pointer on it, then:
- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):
- Position the cursor on the symbol representing the coil type or on the letter of the contact,
- Scroll through the possible coil or contact types (**V** for a normally open contact, **v** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of contacts on program initialization:

- Normally open (direct state) is inactive,
- Normally closed (reverse state) is active.

Fast Counter

Description

The **Fast Counter** function is used to count pulses up to a frequency of 1 kHz.

Use of the **K1** contact indicates:

- The preset value has been reached (upcounting),
- The value 0 has been reached (downcounting).

The **Fast counter** inputs are implicitly connected to the **I1** and **I2** smart relay inputs:

- A pulse (rising edge) on the **I1** input increments the counter,
- A pulse (rising edge) on the **I2** input decrements the counter.

These inputs cannot be used in any other context.

The **Fast Counter** function can be reset to zero during use by the **RK1** coil. It is reset to:

- 0 if it is upcounting towards the preset value
- the preset value if it is downcounting from the preset value.

The counter only operates if the **TK1** confirmation coil is active.

Repetitive cycle type can be used with a time-delay value.

NOTE: Limit overrun:

- If the value of the counter exceeds the upper limit: **+ 32,767**, it is set to **- 32,768**,
- if the value of the counter exceeds the lower limit: **-32,768**, it is set to **+32,767**.

NOTE: This function block cannot be simulated.

Access

The function is accessible from the **LD** function bar.

Use of Coils

Two coils are associated with the fast counter:

- coil **TK1**: **Enable function input**,
- coil **RK1**: **Reset initial counter state input**.

The use of these coils is described below.

Enable function input:

Symbol of the Enable Function Input coil of the fast counter:

TK1

This element is used to confirm the counter. When this coil is active, each rising edge on the **I1** input will increment the **Fast counter** and each rising edge on the **I2** input will decrement it.

Reset initial counter state input:

Symbol of the Reset Initial Counter State Input:

RK1

This input resets the counter function to its initial state.

Energizing the coil has the following effect:

- reset the counter value to **zero** if the **count type** is **TO** (upcounting to the preset value).
- reset the counter value to the **preset value** if the count type is **FROM** (downcounting from the preset value).

Example: Counter reset by pressing on the Z1 key:

Z1 —— RK1

Each time the Z1 key is pressed, the counter is reinitialized.

Use as a Contact

The contact associated with the fast counter indicates whether the preset value (**TO**) or zero (**FROM**) has been reached.

It may be used as many times as necessary in the program either as normally open or as normally closed:

Normally open:

Symbol of the normally open contact associated with the fast counter:

K1

The contact is **conducting when**:

- the value of the counter has **reached** the **preset value** (**TO**),
- the value of the counter has **reached 0** (**FROM**).

Normally closed:

Symbol of the normally closed contact associated with the fast counter:

k1

The contact is **conducting as long as**:

- the counter value **has not reached** the preset value, if the counter is upcounting,
- the counter value **has not reached 0**, if the counter is downcounting,

Example: Lighting a LED connected to fast counter no. 1 output (**TO**).

```
K1 ——[ Q1
```

When the preset value has been reached: The LED is illuminated; otherwise it is off.

Settings from the Software

Type of counting:

Two settings are available:

- **TO**: upcounting towards the preset value.
  When the counter value is greater than or equal to the preset value, contact K1 of the fast counter is conducting.
- **FROM**: downcounting from the preset value.
  When the counter value is less than or equal to 0, contact C of the counter is conducting.

Presetting:

The preset value is between 0 and 32,767.

Cycle type:

This parameter determines the behavior of the fast counter when it reaches the preset value (when upcounting **TO**), or when it reaches the value zero (when downcounting **FROM**):

The cycle type may be:

- **Single**: Reaching the preset value (when upcounting **TO**) or the zero value (when downcounting **FROM**) does not affect the value of the counter.
  The counter value changes on an on-going basis. The output is activated when the value is greater than the preset value (when upcounting **TO**) or when it is less than the preset value (when downcounting **FROM**).
- **Repetitive**: when upcounting **TO**, the value is reinitialized when it reaches the preset value and when downcounting **FROM**, it is reset to the preset value when it reaches zero.
  The output is enabled following this reinitialization and remains active for a time that may be configured with the parameter: **Duration of pulse** (from 1 to 32,767 times 100 ms).

Latching:

By default, after a power outage, the counter is set to the state that corresponds to program initialization.

To restore the state of the counter backed up on power outage, activate latching.

Locked:

Locking prevents locked parameters from being modified from the front panel of the smart relay using the PARAMETERS menu.

Configuration from Front Panel

The block parameter settings can be accessed either when entering the command line or from the **PARAMETERS** menu if the block has not been padlocked.

The parameters to enter are the following:
- Cycle type,
- Duration of pulse,
- Preset value,
- Type of counting,
- Parameter lock,
- Latching.

Cycle type:

This parameter determines the behavior of the fast counter when it reaches the preset value (when upcounting **TO**), or when it reaches the value zero (when downcounting **FROM**):

The cycle type may be:
- **Single**: Reaching the preset value (when upcounting **TO**) or the zero value (when downcounting **FROM**) does not affect the value of the counter.
  The counter value changes on an on-going basis. The output is activated when the value is greater than the preset value (when upcounting **TO**) or when it is less than the preset value (when downcounting **FROM**).
- **Repetitive**: when upcounting **TO**, the value is reinitialized when it reaches the preset value and when downcounting **FROM**, it is reset to the preset value when it reaches zero.
  The output is enabled following this reinitialization and remains active for a time that may be configured with the parameter: **Duration of pulse** (from 1 to 32,767 times 100 ms).

Duration of pulse:

Symbol of the **Duration of pulse** parameter:

I

This parameter is only displayed if the cycle is repetitive. It determines the duration during which the fast counter remains active when the value reaches the preset value (when upcounting **TO**), or when it reaches the value zero (when downcounting **FROM**).

This value must be between 1 and 32,767 (x 100 ms).

Preset value:

Symbol of the **Preset value** parameter:

P

This value is between 0 and 32,767, and represents:
- the value to reach when upcounting to the preset value (**TO**),
- the initial value when downcounting from the preset value (**FROM**).

Type of counting:

Symbol of the **Type of counting** parameter:

T

This parameter is used to select the type of the counter:

- **TO**: upcounting towards the preset value.
  When the counter value is greater than or equal to the preset value, contact K1 of the fast counter is conducting.
- **FROM**: downcounting from the preset value.
  When the counter value is less than or equal to 0, contact C of the counter is conducting.

Parameter lock:

Symbol of the **Parameter Lock** parameter:

Locked Unlocked

Locking prevents the modification of locked parameters from the front panel of the smart relay via the PARAMETERS menu.

Latching:

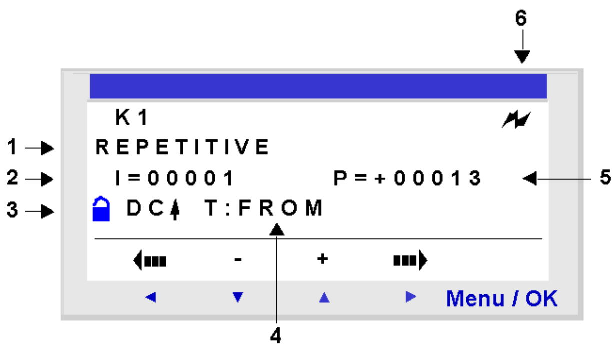
Symbol of the **Latching** parameter:

Active Inactive

This function is used to save the status of the fast counter values in the event of a power outage.

In Zelio entry mode, to activate latching, the symbol must be displayed on the parameter screen.

Illustration: configuring a counter from the front panel of the smart relay:



Description:

Number	Parameter	Description
1	**Cycle type**	Single/Repetitive
2	**Duration of pulse**	Only if the cycle is repetitive
3	**Parameter lock** 	This parameter is used to lock the counter parameters. When the block is locked, the preset value no longer appears in the PARAMETERS menu.
4	**Type of counting**	Counter configuration: Counting to the preset value (**TO**) or from the preset value (**FROM**).
5	**Preset value**	Counter preset value.
6	**Latching**	Backup of the counter value.

Counter Value

Value at a given instant resulting from successive up/down counts since the last counter reset to its initial state.

If the value of the counter exceeds the upper limit: +32,767, it is set to -32,768.

If the value of the counter exceeds the lower limit: -32,768, it is set to +32,767.

Timing Diagrams

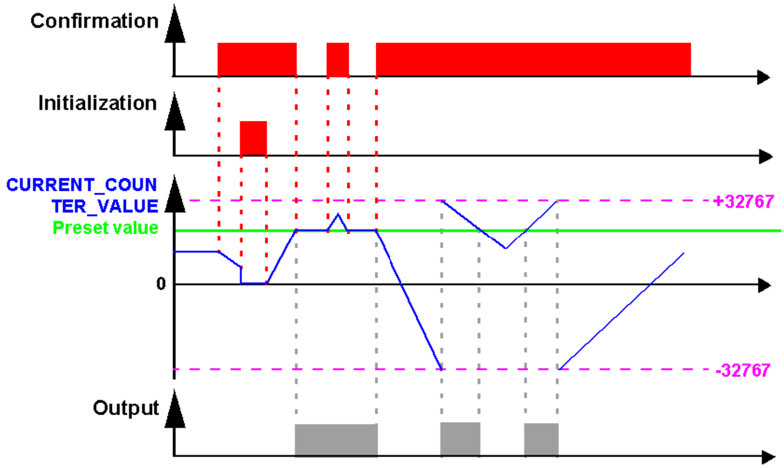
Timing diagrams are provided here to illustrate the various behaviors of the fast counter according to its parameters:

- upcounting function **TO**, in single cycle type,
- downcounting function **FROM**, in single cycle type,
- upcounting function **TO**, in repetitive cycle type,
- downcounting function **FROM**, in repetitive cycle type.

For the following 4 charts, the blue curve represents the value of the counter. When it increases, it is because of pulses on I1 and when it decreases, it is because of pulses on I2.

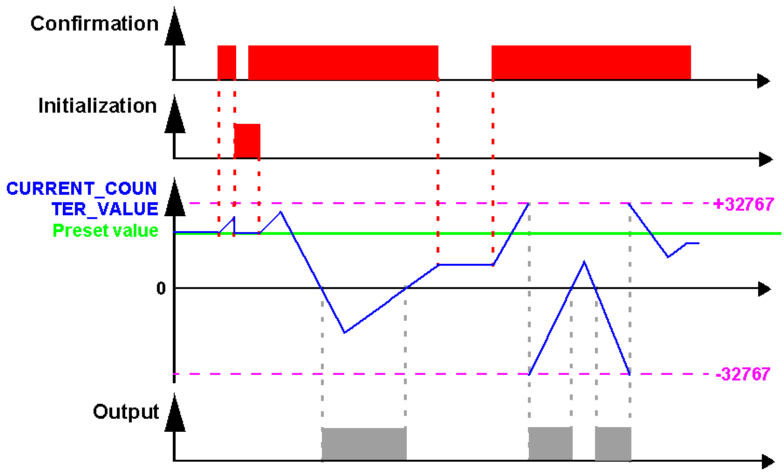
UpCounting in Single Cycle Type:

The figure below illustrates the counter function in upcounting and single cycle type:



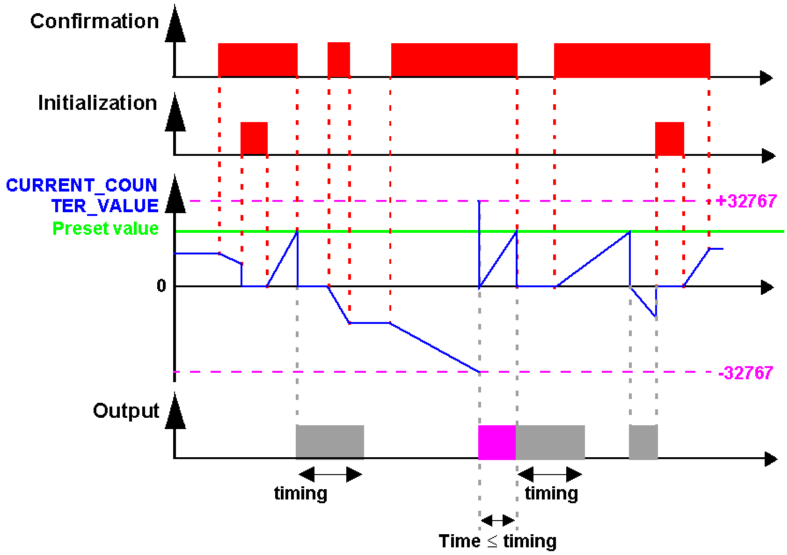
Downcounting in Single Cycle Type:

The figure below illustrates the counter function in downcounting and single cycle type:



Upcounting in Repetitive Cycle Type:

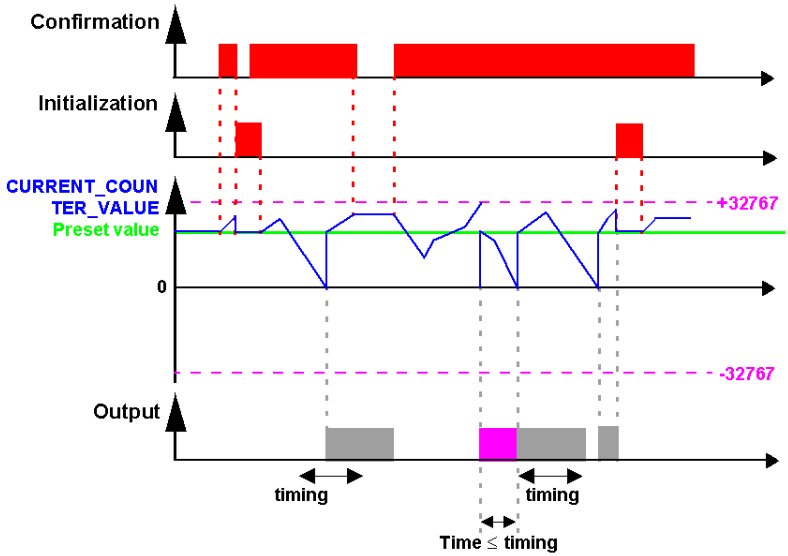
The figure below illustrates the counter function in upcounting and repetitive cycle type:



The output switches to the **Inactive** state when the predefined pulse duration value has elapsed. If the switch condition is Active before the switch to the Inactive state, the output pulse is extended by the **Duration of pulse** (Timing).

Downcounting in Repetitive Cycle Type:

The figure below illustrates the counter function in downcounting and repetitive cycle type:



The output switches to the **Inactive** state when the predefined pulse duration value has elapsed. If the switch condition is Active before the switch to the Inactive state, the output pulse is extended by the **Duration of pulse** (Timing).

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a coil or a contact, position the mouse on the element then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):

- Position the cursor on the symbol representing the coil type or on the letter of the contact,
- Scroll through the possible coil or contact types (**K** for a normally open contact, **k** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of the contacts and value on initialization of the program:

- The **normally open** (direct state) is **inactive**,
- the **normally closed** (inverse state) is **active**,
- The **value** is **zero**.

Example

Below, an example of using a fast counter: output Q1 is set to 1 when the fast counter is set to 1; the counter is activated by input I3 and reset to 0 by input I4.

```
K1-------------[Q1
I3-------------TK1
I4-------------RK1
```

Clocks

Description

Use the **Clocks** function to validate the time ranges during which actions can be executed.

The smart relay has 8 **Clocks** function blocks numbered from 1...8. Each of these has four programming ranges and behaves like a weekly programmer. The **Clocks** function blocks are used like contacts.

Access

This  function is accessible from the **LD** function bar.

Use as a Contact

This contact may use the direct state of the Clock function block (normally open contact) or its inverse state (normally closed contact), see below.

Normally open:

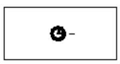
Symbol of the normally open contact, representing a clock:

H-

The contact is conducting when the clock is in a validity period.

Normally closed:

Symbol of the normally closed contact, representing a clock:



h-

The contact is conducting when the clock is not in a validity period.

Settings from the Software

The **Clocks** configuration window consists of 4 blocks, which correspond to the 4 available ranges (or channels): A, B, C, D.

For each range, the days of the week appear, and can be activated by selecting the associated boxes.

Then, the activation time range must be configured by setting the start time: **ON** and the end time: **OFF**.

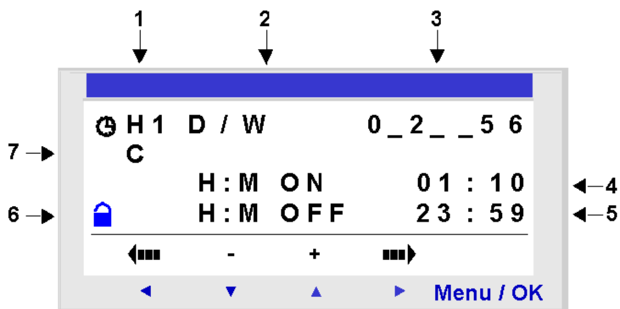
Illustration:

Locked

Locking prevents locked parameters from being modified from the front panel of the smart relay using the **PARAMETERS** menu.

Configuration from Front Panel

Configuration screen of a Clock function block from the front panel of the smart relay:



Number	Parameter	Description
1	Clock module number	8 clocks available, numbered 1...8.
2	Type of date configuration	**D/W**: Days of the Week.
3	Validity day (D/W type)	Validity day: ● **0**: Monday ● **1**: Tuesday ● ... ● **6**: Sunday **Unselected** days are indicated by a _.
4	Start time (D/W type)	This is the start time from 00.00...23:59.

Number	Parameter	Description
5	Stop time (D/W type)	This is the end time from 00.00...23:59.
6	Parameter lock 	Locking prevents locked parameters from being modified from the front panel of the smart relay using the **PARAMETERS** menu.
7	Operating ranges	4 operating ranges are available: A, B, C, D. In operation, these ranges are cumulative: The block is valid over the selected ranges.

Combining Operating Ranges

Operating ranges can be mixed for the same clock.

Example: Using the four operating ranges with different settings.

Operating range	Program
A: time range	Every day from Monday to Friday, start at 8.00 and end at 18.00.
B: Day/Night	Every day from Tuesday to Thursday: start at 22.00 and end the following day at 6.00.
C: interval	Start on Friday at 20.00.
D: Interval	End on Monday at 03.00.

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a contact, position the pointer on it, then:
- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):
- Place the cursor over the letter of the contact,
- Scroll through the possible contact types (**H** for a normally open contact, **h** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Simulation

In Simulation *(see page 223)* mode, the operation of the Clock function block is determined by the configuration of the accelerator *(see page 229)*.

Example

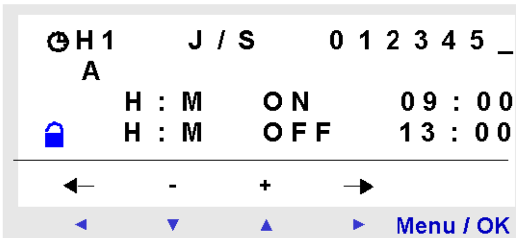
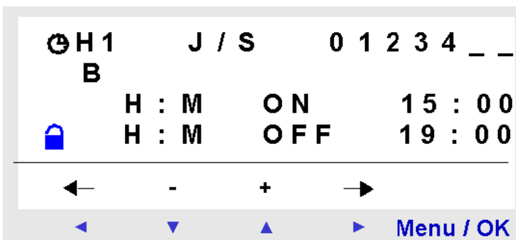
You wish to control a device connected to the Q2 output of the smart relay. You want it to be active on the following two clock ranges:

- from Monday to Saturday, from 09:00 to 13:00.
- from Monday to Friday, from 15:00 to 19:00.

For this, the **H1** Clock block is used and the following wiring sheet is created:

```
H1-------------[Q2
```

When entering the H1, Clock block, configure the operational ranges **A and B** as described in the table below:

Screen	Comment
	**First Slot A**: From Monday to Saturday, from 09:00 to 13:00.
	**Second Slot B**: From Monday to Friday, from 15:00 to 19:00.

Change to Summer / Winter Time

Description

The output of this function is in an OFF state over the entire duration of winter time, and switches to ON for the entire duration of summer time.

By default, there is no change in winter / summer time. This function must be activated, either from the programming software, or from the front panel of the smart relay.

To activate this function, from the programming software, proceed as follows:

- display the **Program configuration** window: **Edit** → **Program Configuration** menu,
- Select the **Date format** tab,
- Select the **Activate Summer/Winter Time Change** box,
- Define the time change dates:
  - Either using one of the predefined geographic zones,
  - Or by manually configuring the date (month/Sunday).

To activate this function from the front panel of the smart relay, proceed as described in chapter CHANGE SUM/WIN Menu *(see page 119)*.

NOTE: This function is only available for smart modules that contain a real-time clock.

Access

This  function is accessible from the **LD** function bar.

Use as a Contact

When used as a contact, this element indicates winter time or summer time.

It may be used as many times as necessary in the program either as normally open or as normally closed:

Normally open:

Symbol of the normally open contact associated with a Change summer / winter time function block:

W 1

The contact is active for the entire duration of summer time.

Normally closed:

Symbol of the normally closed contact associated with a Change summer/winter time function block

w 1

The contact is active for the entire duration of winter time.

Parameters

The following settings are possible:

- **No**: no change,
- Automatic change: Dates are preset according to geographic zone:
  - **EUROPE**: Europe,
  - **USA**.
- **OTHER ZONE**: The change is automatic, but you must specify the month: **M** and the Sunday: **S** (1, 2, 3, 4 or 5) on which the summer/winter change takes place.

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a contact, position the cursor on it, then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: Scroll through the possible states.

To modify a contact from the front panel of the smart relay (the programming window displayed on screen):

- Place the cursor over the letter of the contact,
- Scroll through the possible contact types (**W** for a normally open contact, **w** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

State of the contacts and value on initialization of the program:

- The **normally open** (direct state) is **inactive**,
- The **normally closed** (inverse state) is **active**.

Timers

Description

Use the **Timers** function to delay, prolong and control actions over a predetermined period. Durations can be set using one or two preset values, according to the type of timer.

There are 11 types of timers:

- A: **Active, control held down**,
- a: **Active, Press to start/stop**,
- C: **Off delay**,
- B: **On pulse one shot**: Pulse calibrated on the command input rising edge,
- W: **Timing after pulse**: Pulse calibrated on the command input falling edge,
- D: **Symmetrical flashing**: control held down synchronously,
- PD: **Symmetrical flashing, Start/stop on pulse**,
- T: **Time on addition**,
- AC: **A/C**: Combination of A and C,
- L: **Flasher unit, control held down asynchronously**,
- l: **Flasher unit; Press to start/stop**.

For the description of different types of timers, refer to the Timing Diagrams *(see page 187)*.

The smart relay has either 28 timer function blocks, or 16 timer function blocks if an SR2COM01 communication interface is present. They are numbered from 1...9 then from A...V (I, M, O are not used).

NOTE: The maximum number of timers also depends on the firmware version *(see page 540)*.

Each block has a reset input, a command input and an output used to indicate timer time-out.

Access

This  function is accessible from the **LD** function bar.

Use of Coils

Two coils are associated with each timer:

- Coil **TT**: **Command Input**,
- Coil **RC**: **Reset Input**,

The use of these coils is described below.

Command input:

Symbol of the Command input coil of a timer:

TT-

Each type involves a specific operation, which can be used to manage the possible scenarios in an application.

Reset input:

Symbol of the Reset input coil of a timer:

RT-

Energization of the coil causes a reset of the timer value: contact T is deactivated and the function is ready for a new timer cycle.

NOTE: This coil is only necessary for pulse start/stop type timers.

Use as a Contact

The contact associated with the timer indicates whether the timer has stopped.

It may be used as many times as necessary in the program either as normally open or as normally closed:

Normally open:

Symbol of the normally open contact associated with a timer:

T-

If the output of the Timer function block is **active**, the contact is **conducting**.

Normally closed:

Symbol of the normally closed contact associated with a timer:

t-

If the output of the Timer function block is **active**, the contact is **non-conducting**.

Software Setting

Type of timer:

The type of timer is selected by checking the desired box. The operating diagram of the timer appears below.

Time unit:

The format of the time unit is made using the associated drop-down menu.

Delay:

The delay(s) are entered in the associated fields.

Latching:

This function is used to save the state of the timer values in the event of a power outage.

Locked:

Locking prevents locked parameters from being modified from the front panel of the smart relay using the PARAMETERS menu.

Configuration from Front Panel

The block parameter settings can be accessed either when entering the command line or from the **PARAMETERS** menu if the block has not been padlocked.

The parameters to enter are the following:

- Timer type,
- Preset value(s),
- Time unit,
- Parameter lock,
- Latching.

Type of timer:

This parameter allows you to choose the type of timer function from among the 11 types available. Each type is represented by one or two letters:

- A: **Active, control held down**,
- a: **Active, Press to start/stop**,
- C: **Off delay**,
- B: **On pulse one shot**: Pulse calibrated on the command input rising edge,
- W: **Timing after pulse**: Pulse calibrated on the command input falling edge,
- D: **Symmetrical flashing**: control held down synchronously,
- PD: **Symmetrical flashing, Start/stop on pulse**,
- T: **Time on addition**,
- AC: **A/C**: Combination of A and C,
- L: **Flasher unit, control held down asynchronously**,
- l: **Flasher unit; Press to start/stop**.

Preset value:

Depending on the type of timer, there can be 1 or 2 preset values:

- 1 preset value for the A, a, C, B, W, D, PD and T types:

  t : on-delay or off-delay according to type.

- 2 preset values for the AC, L and I types:

  A : Timer on-delay in the case of AC type; active state in the case of flasher units L and I.

  B : Timer off-delay in the case of AC type; inactive state in the case of flasher units L and I.

Time unit:

This is the time unit for the preset value. There are five possibilities:

Unit	Symbol	Form	Maximum value
1/100 of a second	s	00.00 s	00.00 s
1/10 of a second	S	000.0 s	00.00 s
Minutes : Seconds	M : S	00 : 00	99 : 99
Hour : Minute	H : M	00 : 00	99 : 99
Hours Only for type T.	H	0 000 h	9,999 h

Parameter lock:

Symbol of the **Parameter Lock** parameter:

Locked Unlocked

Locking prevents the modification of locked parameters from the front panel of the smart relay via the PARAMETERS menu.

Latching:

By default, if a power outage occurs while a timer function block is running, the information on time already elapsed is lost. When the supply voltage returns, the time function block is reinitialized and ready for a new operating cycle.

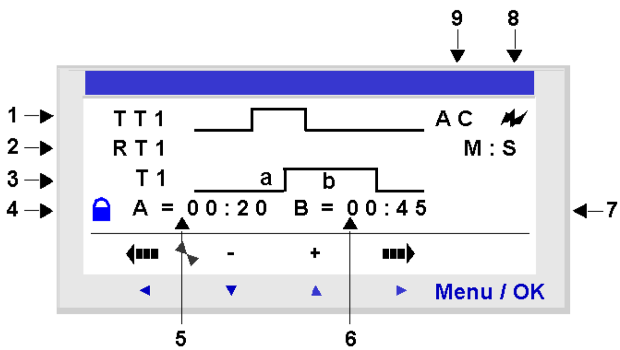
If the application requires it, the time elapsed before the power outage can be memorized using the **Latching** parameter.

Symbol of the **Latching** parameter:

 Active  Inactive

This function is used to save the timer values and memorize the elapsed time in the event of a power outage.

Illustration: Configuring a counter from the front panel of the smart relay:



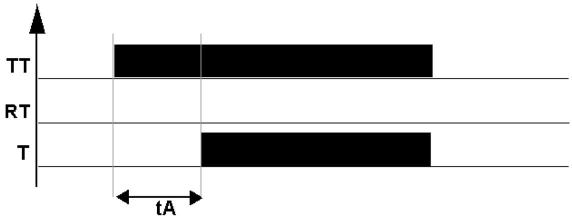
Description:

Number	Parameter	Description
1	**Command input**	Command input timing diagram.
2	**Reset input**	Reset input timing diagram.
3	**Timer output**	Timer output timing diagram.
4	**Parameter lock** 	This parameter is used to lock the counter parameters. When the block is locked, the preset value no longer appears in the PARAMETERS menu.
5	**Timer on-delay**	Timer on-delay of the AC timer.
6	**Timer off-delay**	Timer off-delay of the AC timer.
7	**Time unit**	Time unit for the preset value.
8	**Latching**	Backup of counter value.
9	**Timer type**	Type of timer used.

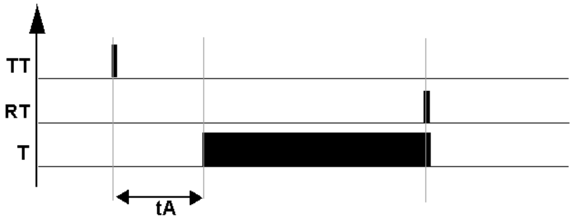
Timing Diagrams

Timing diagrams are provided here to illustrate the various behaviors of the Timer function block, according to the selected type of timer:

Type A is **Active, control held down**. The following diagram shows the operation of the type A timer:

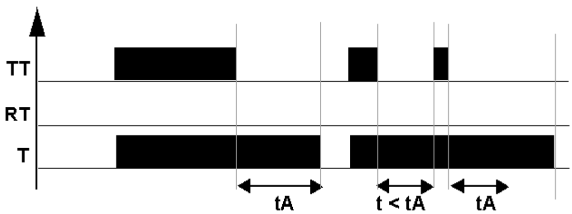


Type a is **Active, Press to start/stop**. The following diagram shows the operation of the type a timer:

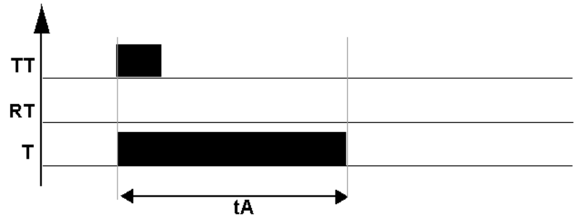


NOTE: Each rising edge on the TTx input resets the timer value to 0.

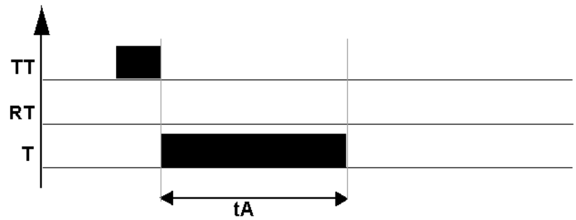
Type C is **Off delay**. The following diagram shows the operation of the type C timer:



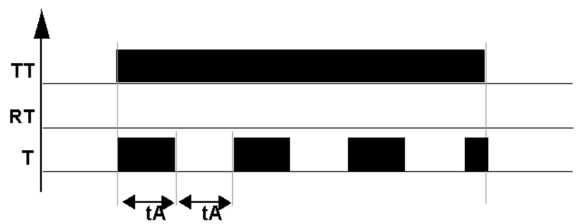
Type B is **On pulse one shot** for a pulse calibrated on the command input rising edge. The following diagram shows the operation of the type B timer:



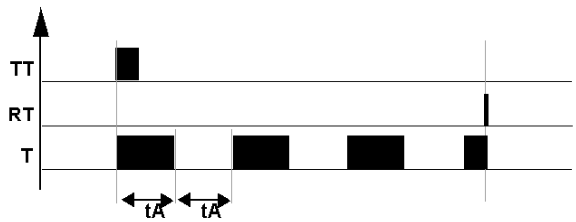
Type W is **Timing after pulse** for a pulse calibrated on the command input falling edge. The following diagram shows the operation of the type W timer:



Type D is **Symmetrical flashing** for control held down synchronously. The following diagram shows the operation of the type D timer:

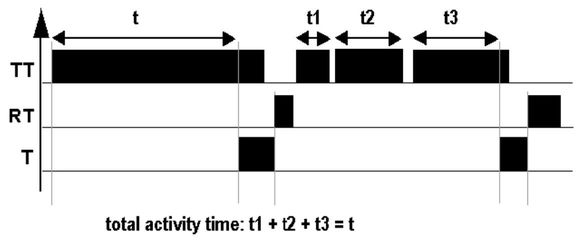


Type PD is **Symmetrical flashing, Start/stop on pulse**. The following diagram shows the operation of the type PD timer:



NOTE: Each rising edge on the TTx input resets the timer value to 0.

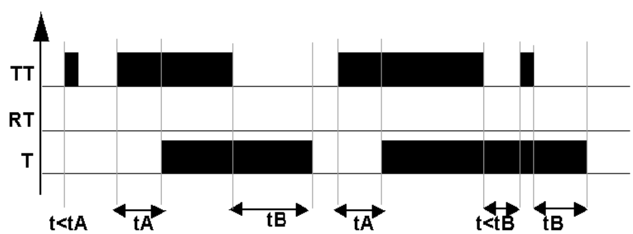
T is **Time on addition**. The following diagram shows the operation of the type T timer:



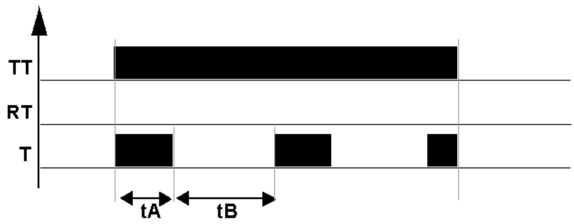
With this type, the preset value can be reached:

- In one step: t,
- In several steps: t1 + t2 + ... + tn.

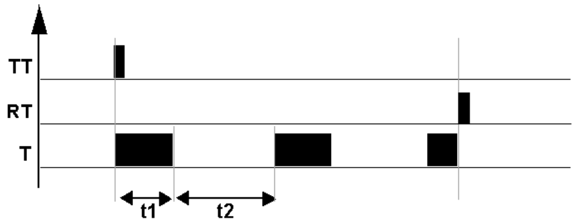
Type AC (**A/C**) is a combination of A and C. The following diagram shows the operation of the type AC timer:



Type L is **Flasher unit, control held down asynchronously**. The following diagram shows the operation of the type L timer:



Type I is **Flasher unit; Press to start/stop**. The following diagram shows the operation of the type I timer:



NOTE: Each rising edge on the TTx input resets the timer value to 0.

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a coil or a contact, position the mouse on the element then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):

- Position the cursor on the symbol representing the coil type or on the letter of the contact,
- Scroll through the possible coil or contact types (**T** for a normally open contact, **t** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

State of the contacts and values on initialization of the program:

- The **normally open** (direct state) is **inactive**,
- the **normally closed** (inverse state) is **active**,
- the **value(s)** is (are) **zero(s)**.

Example 1

Creating a timer device for a stairway.

The stairway light should remain on for two minutes and thirty seconds when one of the push buttons is activated.

On each floor, the buttons are linked to the **I1** input of the smart relay.

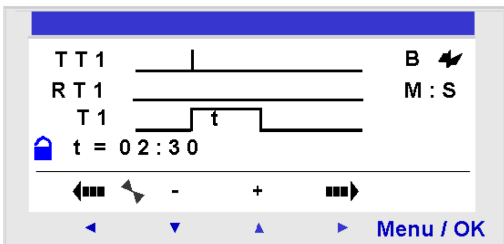
The stairway light is linked to the **Q4** output of the smart relay.

You would write the following program:

```
I1--------------TT1
I2--------------RT1
T1--------------[Q4
```

To obtain the desired operation, you should use a type B timer (On pulse one shot), and configure the duration of the timer for 2 minutes 30 seconds. To thus configure the timer duration, choose the time units **M : S** and enter the value **02:30** for the preset value **t**.

Illustration: Timer's configuration screen:



Analog Comparators

Description

The **Analog Comparators** function block is used to:

- Compare a measured analog value with a reference value.
- Compare two measured analog values.
- Compare two measured analog values with hysteresis parameter.

The result of this comparison is used in the form of a contact.

Analog automation functions can be used for smart relays with a real time clock and DC power supply, and with mixed discrete and analog inputs.

The following indicate the existence of mixed discrete and analog inputs:

- The existence of inputs numbered from **IB** to **IG** (maximum configuration). These inputs are used to receive analog signals from 0.0 V to 9.9 V inclusively.
- The presence of the **Analog Comparators** function in the toolbar of the programming software.

These smart relays have 16 **Analog Comparators** function blocks, numbered from 1 to 9 then from A to G.

Access

This  function is accessible from the **LD** function bar.

Use as a Contact

The contact shows the position of a measured analog value in relation to a reference value or to another measured value.

It may be used as many times as necessary in the program either as normally open or as normally closed:

Normally open:

Symbol of the normally open contact associated with an Analog Comparator:

A-

The contact is **conducting** when the comparison condition is **verified**.

Normally closed:

Symbol of the normally closed contact associated with an analog comparator:

a-

The contact is **conducting** when the condition **is not verified**.

Settings from the Software

Comparison operator:

The simple comparison formula is as follows:

Value1 <Comparison operator> Value2

The <Comparison Operator> is chosen among the following:
- >
- ≥
- =
- ≠
- ≤
- <

The comparison formula, for a comparison with hysteresis:

Value1 - H ≤ Value2 ≤ Value1 + H

In the above formulas, the Value 1 and Value 2 variables are chosen among the analog inputs Ib to Ig or the reference value.

The selection is made by clicking the corresponding button; the formula is displayed above.

Value1 and Value2:

Value1 and Value2 are configured using the associated scroll menus.

Reference and Hysteresis value:

These values are to be entered in the associated fields. The value must be between 0.0 and 9.9.

Locking:

Locking prevents locked parameters from being modified from the front panel of the smart relay using the PARAMETERS menu.

Configuration from Front Panel

The comparison formula is as follows:

xl <Comparison Operator > x2

The comparison formula, for a comparison with hysteresis is as follows:

x1 - H ≤ x2 ≤ x1 + H

The parameters to enter are the following:

- Values to compare,
- Comparison operator,
- Reference value,
- Hysteresis parameter,
- Parameter lock.

Values to compare:

Symbol of values to compare:

x1 **x2**

These variables are chosen from among the following:

- Numbered analog inputs from IB to IG (maximum configuration),
- Reference value R

Comparison operator:

The comparison operator is chosen using the number in the upper right-hand side of the front panel display.

The table below provides the correspondence between this number and the comparison formula that will be used:

Number	Comparison formula
1	x1 > x2
2	x1 ≥ x2
3	x1 = x2
4	x1 ≠ x2
5	x1 ≤ x2
6	x1 < x2
7	comparison with hysteresis: x1 - H ≤ x2 ≤ x1 - H

Reference value:

Symbol of the reference value:

R

The reference value is a constant to which a measured value may be compared. It must be between 0 and 9.9.

Hysteresis parameter:

Symbol of the hysteresis parameter:

H

The hysteresis parameter is a constant used to define an interval in which the 2x variable should be found for the comparator to be active. Its value must be between 0 and 9.9.

Parameter locking:

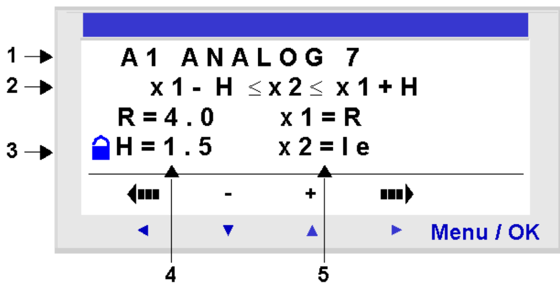
Symbol of the Parameter lock parameter:

Locked Unlocked

Locking prevents the modification of locked parameters from the front panel of the smart relay via the PARAMETERS menu.

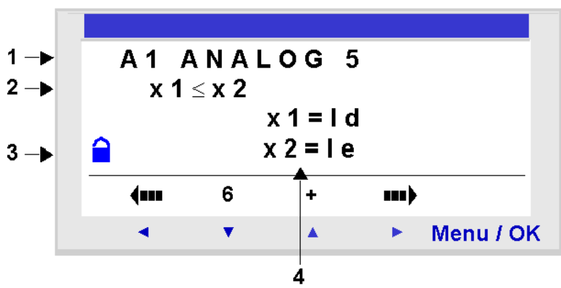
Illustration:

Configuration from the front panel of the smart relay, of the hysteresis-type comparator with constant reference value:



In this case: The comparison condition is verified when the power to the input terminal Ie is between 2.5 V and 5.5 V.

Configuration of a single comparator from the front panel:



Description:

Number	Parameter	Description
1	**Type of comparison**	The number that follows ANALOG corresponds to the selected comparison operator.
2	**Comparison formula**	Formula used for comparison.
3	**Parameter lock**	Locking prevents locked parameters from being modified from the front panel of the smart relay using the PARAMETERS menu.
4	**Parameters of the comparison formula**	Parameters of the comparison formula.

Modifying the State of a Coil or a Contact

In the programming software, to modify a contact, position the cursor on it, then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: Scroll through the possible states.

To modify a contact from the front panel of the smart relay (the programming window is displayed on the screen):

- Place the cursor over the letter of the contact,
- Scroll through the possible contact types (**A** for a normally open contact, **a** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Initialization

Status of the contacts and value on initialization of the program:

- The **normally open** (direct state) is **inactive**,
- The **normally closed** (inverse state) is **active**.

Example

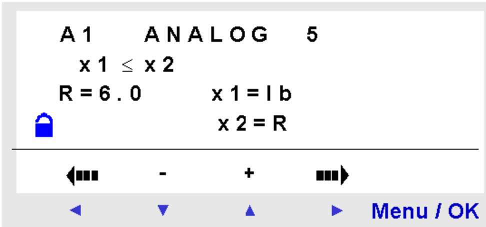
A heating resistance is to be triggered by the smart relay **Q1** output when the temperature is below 20° C.

A temperature probe is used, providing a 0 to 10 volt signal for a -10° to +40° C temperature range. A temperature of 20° C corresponds to a voltage level of 6 volts on the probe.

You would write the following Ladder program:

```
A1-------------[Q1
```

Using the following parameters for the A1 comparator:



The comparison operator 5 is chosen, that is “inferior or equal to”.

The values to compare are chosen: The analog input IB (to which the temperature probe is connected) for the first, the reference value R for the second.

The reference value is set to 6.

The analog comparator is thus active when the power measured on the analog input IB is less than or equal to 6 V. This is when the probe measures a temperature less than or equal to 20 °C.

TEXT

Description

The **TEXT** function is used to display text, a date, a time, and numerical values on the LCD display, instead of the inputs-outputs states.

One single **TEXT** function block is used to define the content of the entire LCD display. The content can be a combination of:

- Text (maximum 72 characters),
- Numerical values corresponding to the output of a function used in the application (for example a counter). These values can include a decimal point.
- Date, time or calibration value from the **Clocks** function.

It is possible to authorize the modification of the content using the keys on the front panel.

The smart relays have 16 **TEXT** blocks, numbered from 1...9 then from A to G. These function blocks are used as coils.

The maximum number of variables that can be displayed per **TEXT** block is 4.

Up to 16 **TEXT** blocks may be used (TX1 to TXG) simultaneously in one program, but only the block which is activated is displayed. If multiple blocks are activated, the block with the highest number is displayed.

To switch the display from the **TEXT** screen to the INPUTS-OUTPUTS screen, hold down the **Shift** key, then press the **Menu/OK** key.

NOTE: The **TEXT** blocks are only programmable from the programming software.

Access

This 

function is accessible from the **LD** function bar.

Used as a Coil

Two coils are associated with each **TEXT** block:

- **Display Activation** coil.
- **Display Deactivation** coil.

The use of these coils is described below.

Display Activation

Symbol of the **Display Activation** coil of a **TEXT** function block:

TX -

This coil displays on the screen the text and/or the values of the associated **TEXT** block when the elements that are connected are conducting.

Display Deactivation

Symbol of the **Display Deactivation** coil of a **TEXT** function block:

```
RX -
```

This coil deactivates the display of the text and/or the values of the associated **TEXT** block when the elements that are connected are conducting. The display returns to the inputs-outputs screen.

<u>Example:</u>

```
I1 ——— TX1
I2 ——— RX1
```

Activation of input **I1** displays the text on the LCD. Activating input **I2** makes the text disappear.

Text Block Identification

Each display function is identified by a **current text** number (TX1 to TXG).

This identifier may be found in the parameters window, in the **Parameters** tab: the number is in the drop-down menu at the top of the window.

The parameters window opens by default on the function block number from which the dialog box is open.

Character String Display

When the parameters window for a new **TEXT** box is opened, the cursor is positioned over the first flashing box.

The cursor is positioned at the start of the string displayed in the window:

- By clicking on the box (which then flashes),
- By using the arrow keys on the computer keyboard.

Description of the entry procedure:

Step	Action
1	Position the cursor at the beginning of the text.
2	Type the text to be displayed using the keyboard.
3	Confirm by clicking **OK**. **Result**: The new **TEXT** block is saved and the parameters window is closed.

NOTE: The character string is limited to the line. If you keep typing characters, each additional character overwrites the one in the last box.

NOTE: ASCII-standard characters, together with accented characters can be used. Characters and symbols are not displayed in the data entry window if they are not supported.

NOTE: If the text entered on a line overwrites an existing numerical value, the latter is deleted. If a numerical value is positioned over text that has already been entered, the characters are overwritten.

Displaying a Numerical Value

Positioning

To position the value on the line, drag and drop it to the edit window.

Selection:

The value to be displayed is selected in the window located above the edit window.

This window lists the following elements:

- **Date**: The value of the internal date (day.month.year) of the device on which the program is executed (smart relay or simulator),
- **Hour**: The time value of the smart relay (hours:minutes),
- **Calibration**: The drift value of the smart relay clock,
- **Values** (current, preset, etc.) belonging to one of the function blocks used in the diagram.

List of values that may be displayed:

- Timer *(see page 182)*: current and preset value,
- Counter *(see page 157)*: current and preset value,
- Analog Comparator *(see page 192)*: Value of analog inputs used in the comparators, hysteresis value.

Display Limitations:

In Ladder, more than 4 variables cannot be displayed simultaneously.

Example of **TEXT** block:

Date: DD.MM.YYYY

Hour: HH.MM Der: CCC

C1C=_C1_C

T1C=

T1_CAUUU cannot be positioned after T1C.

Authorized modification:

The parameters for which the **Authorized modification** option has been enabled (displayed in green in the edit window of the **Parameter** tab) can be modified from the front panel of the smart relay.

Description of the modification procedure for displayed values (active **TEXT** block):

Step	Action
1	Press the (white) **Shift** key and the ▶ key to display the contextual menu. **Result**: The parameter which can be modified flashes and the following contextual menu is displayed: 
2	Select the parameter to be modified using the navigation keys ◀ and ▶ from the contextual menu (the value which are available for modification flash).
3	Modify the parameter value with the + (▲) and - (▼) keys from the contextual menu.
4	Confirm the changes by pressing the **Menu/OK** key. **Result**: The display returns to the inputs-outputs screen or the **TEXT**/**DISPLAY** screen.

Clear Text

Description of the procedure:

Step	Description
1	Select the zone to be cleared.
2	Clear using the **Delete** key on the keyboard.

LCD Screen Backlighting

Description

The **LCD screen backlighting** output is used to control the backlighting of the LCD by a program.

In STOP and RUN modes, the LCD screen is illuminated for 30 seconds when a key is pressed on the front panel.

Access

This 

function is accessible from the **LD** function bar.

Used as a Coil

Used as a coil, it illuminates the LCD when the elements to which it is connected are conducting.

Symbol of the coil of the LCD screen Backlighting function:

TL1

The screen is illuminated if this coil is active.

Message

Description

When activated, the **Message** function block can be used to:

- Send alarm messages to mobile phones, the Zelio Logic Alarm tool or Email addresses via the SR2COM01 communication interface
- Provide remote access to I/O and/or a digital variable for reading or modifying them.

There are 28 **Message** function blocks numbered from S1...S9, then from SA...SV (SI, SM, SO are not used).

NOTE: The **Message** function is only available on smart relays with clocks and when an SR2COM01 communication interface is added.
For further information on the configuration of the communication interface, refer to Configuring the SR2COM01 Communication Interface *(see page 476)*.

Access

This  function is accessible from the **LD** function bar.

Use of the Coil

Command input

Symbol of the Command Input coil of a **Message** function block:

TS-

This coil sends the configured alarm message in the associated **Message** function block, when it is activated.

Depending on the configuration of the **Message** function block, the coil may be activated during detection on its input, by a transition:

- From Inactive to Active State (by default),
- From Active to Inactive State.

For more information, refer to Configuration from the programming software *(see page 205)*.

Use as a Contact

The contact associated with the **Message** function block indicates whether the function block is activated.

It may be used as many times as necessary in the program either as normally open or as normally closed:

Normally open:

Symbol of the normally open contact associated with a **Message** function block:

```
S-
```

The contact is conducting when the function block is activated.

Normally closed:

Symbol of the normally closed contact associated with a **Message** function block:

```
s-
```

The contact is conducting as long as the function block is not activated.

Example: Illuminating an LED connected to the **Message** No.1 function block output

```
S1 ——— [ Q1
```

When function block no. 1 is activated, the associated alarm message is sent and the LED is illuminated, otherwise it is off.

Modifying the State of a Coil or a Contact

In the programming software, to modify the state of a coil or a contact, position the mouse on the element then:

- With the mouse: right-click to display a list of possible states (click to validate),
- With the space bar: scroll through the possible states.

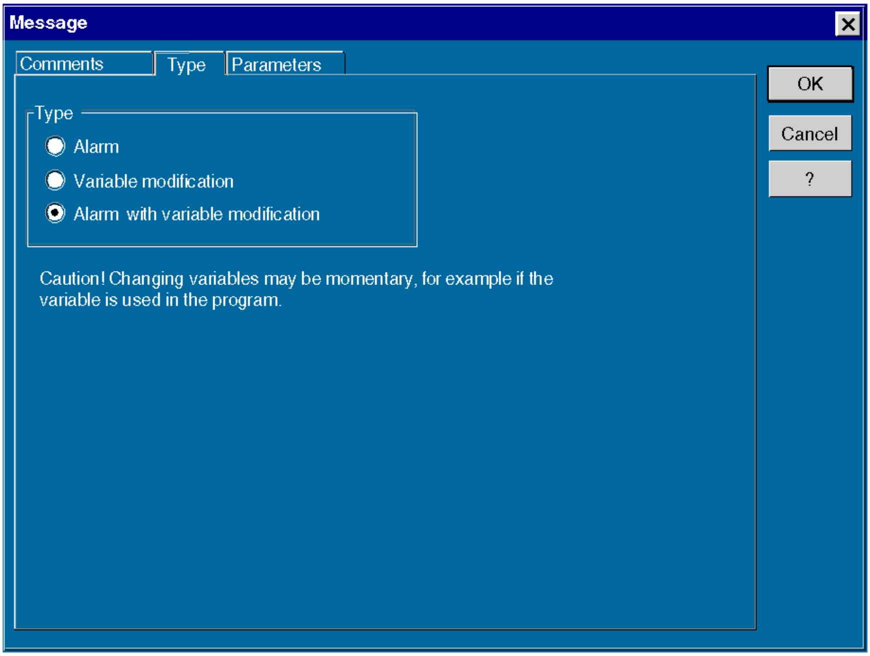
To modify the type of a coil or a contact from the front panel of the smart relay (the programming window displayed on screen):

- Position the cursor on the symbol representing the coil type or on the letter of the contact,
- Scroll through the possible contact types (**S** for a normally open contact, **s** for a normally closed contact).
  For more details, refer to Method for Entering a Contact or Coil *(see page 83)*.

Configuration from the programming software

Double click on the function block to make the **Message** window appear. Use the **Type** and **Parameters** tabs of this window to configure the block.

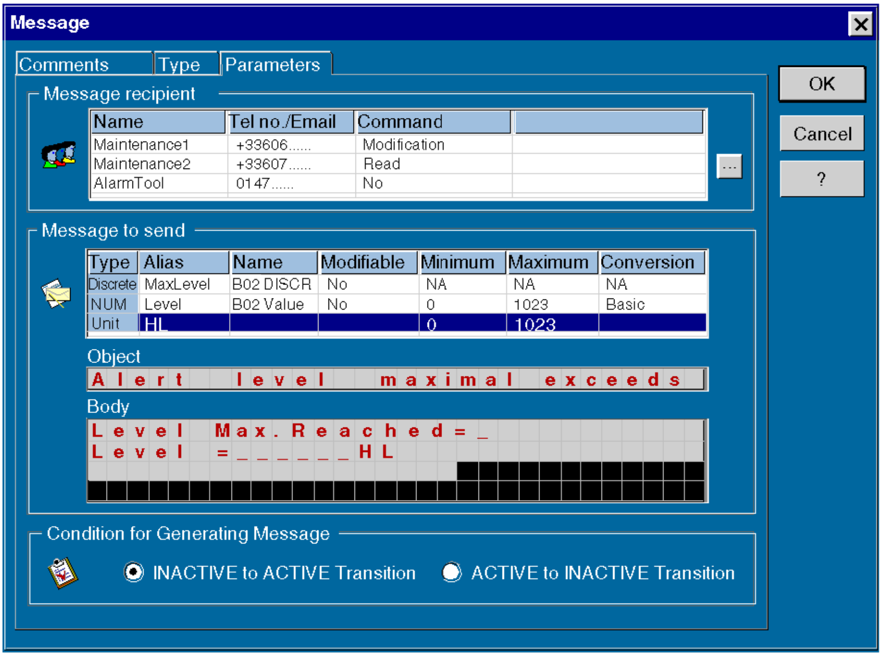
In the Type tab:



Select the **Type** of the alarm message:

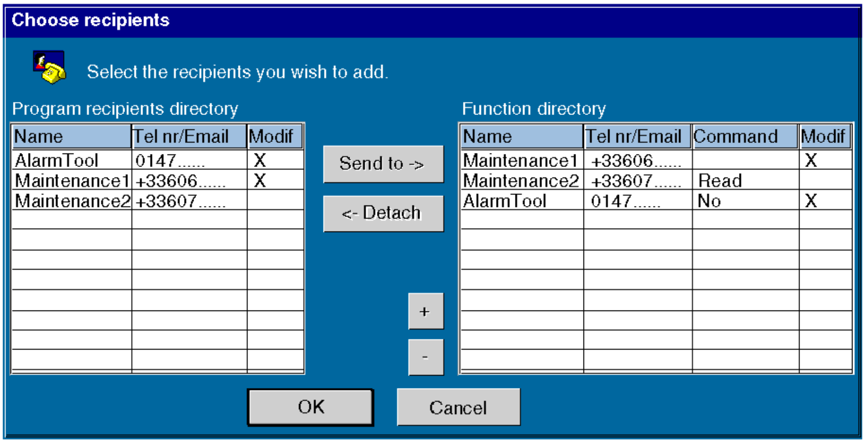
- **Alarm**: used to send an alarm message on activation of the function block. The variables associated with the block may be displayed in the message.
- **Variable modification**: Used to provide access to a I/O and/or a digital variable. Each of these two variables may be declared as readable and modifiable.
- **Alarm with variable modification**: Used to send an alarm message on activation of the function block. The values of variables linked to the block may be displayed in the message and are used to provide access to I/O and/or digital variables, each of these 2 variables may be declared as readable and modifiable.

In the **Parameters** tab:



Specify the alarm **Message recipient** in this block.

To do this, proceed as follows:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Click on the button of the Message recipient zone to add a recipient or modify the list of recipients of this message.
Result: The following window appears:


These recipients are displayed from the Program recipients directory (see page 473).</td></tr>
<tr><td>2</td><td>For each new recipient to be added, select it in the directory of the program and click on the Send to -> button.</td></tr>
<tr><td>3</td><td>Organize the recipients in the order of priority (see page 209) by using the + and - buttons.</td></tr>
<tr><td>4</td><td>Double click on a recipient of the function to specify the types of Command that it will be authorized to execute:
● Choose No to prevent a recipient from accessing variables connected to the Message function block,
● Choose Read to enable a recipient to read commands for variables connected to the Message function block,
● Choose Modify to allow recipients to read and modify variables connected to the Message function block: this choice is only available for recipients authorized to modify variables in the directory.</td></tr>
<tr><td>5</td><td>Confirm by clicking on the OK button.</td></tr>
</table>

Choose the **variables associated** with the **Message** function block. The values of the linked variables may be (according to the configuration of the **Message** function block) displayed in the alarm messages sent, and/or modified using the commands sent from a mobile telephone or from the Zelio Logic Alarm tool. It is possible to link 1 I/O and/or digital variable with each **Message** function block.

To do this, proceed as follows:

Step	Action
1	Click on the button in the second zone in the window to select the variables accessible for this **Message** function block. **Result**: The **Values to Send** window appears.
2	In the **Values to Send**: ● Choose, in the list of I/O variables in the program, the one that will be associated with this **Message** function block. ● Choose, in the list of digital variables in the program, the one that will be associated with this **Message** function block. Confirm by clicking on the **OK** button.
3	Double click on the line of the I/O variable, to access the **Configure** window. This window is used to: ● Modify the **Alias** of the variable, ● Possibly make the variable readable and modifiable. Confirm by clicking on the **OK** button.
4	Double click on the line of the digital variable, to access the **Configure** window. This window is used to: ● Modify the **Alias** of the variable, ● Possibly make the variable readable and modifiable, ● Define a range of possible values for the variable. Then, if a modification command is sent, the new value needs to be found in this page, otherwise the command will not be processed, Confirm by clicking on the **OK** button.

Define the **Object** and **Body** of alarm message (these 2 fields are not available if the type chosen is **Variable modification**).

It is possible to include the value of one or both variables associated with the function block in the body of the message. To do so:
- Select the variable in the list,
- Drag and drop the variable into the zone of the message body.

NOTE: When using GSM modems to send an **Email by SMS**, the syntax used in the subject and body of the message is specific to each telephone operator. Contact the telephone operator of the SIM card of the modem and refer to the section Alarm Message Frame *(see page 482)* for further information.

Define the **Condition for generating message** (this field is not available if the type chosen is **Variable modification**).

Choose:
- **INACTIVE to ACTIVE Transition**, so that the message will be sent when the coil input becomes active,
- **ACTIVE to INACTIVE Transition**, so that the message will be sent when the coil input becomes inactive,

Order of Priority

When sending an alarm message, the SR2COM01 communication interface contacts the message recipients one after the other. The **Choose Recipients** window is used to define the order in which the message recipients are contacted.

Depending on whether the **Recognition** option is activated or not, two types of recipients may be defined:
- **Recipient without recognition**: The alarm message is systematically sent, then the communication interface processes the next recipient,
- **Recipient with recognition** (for mobile-type recipients only): The communication interface sends the alarm message and waits for acknowledgment of the recipient via their mobile telephone:
  - If the recipient with recognition acknowledges the message, the communication interface continues the send sequence only to recipients **without** recognition,
  - If the recipient with recognition does not acknowledge the message in the given time (**Recognition delay**), the communication interface processes the next recipient.

For more information on activating the **Recognition** option, refer to Creating a Recipient *(see page 475)*.

Configuration from the Front Panel

The **Message** function block cannot be configured from the front panel of the smart relay. This function must be configured from the programming software.

Initialization

Status of contacts on program initialization:

- The normally open (direct state) is inactive,
- The normally closed (inverse state) is active.

Chapter 20
Programming Ladder Using Zelio Soft 2

Subject of this Chapter

This chapter describes the different functions that can be accessed from the Zelio Soft 2 programming software in LD mode.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
20.1	Creating an LD Application in the Zelio Soft 2 programming software	212
20.2	Debugging an LD Application in the Programming Software	222

Section 20.1
Creating an LD Application in the Zelio Soft 2 programming software

Subject of this Section

This section describes the different functions linked to programming in the programming software in LD mode.

What Is in This Section?

This section contains the following topics:

Topic	Page
Enter a Contact or a Coil	213
Enter a Link	215
Automation Function Configuration	216
Insert and Delete a Program Line	218
Copy Parts of a Program	219
Check Program Consistency	220

Enter a Contact or a Coil

Description

This section describes the procedures for performing the following operations:

- Entering an element,
- Modifying an element,
- Deleting an element.

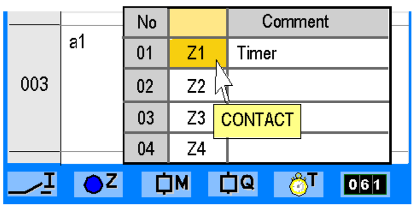
This is valid for either type of element: contact or coil, whether its parameters can be set or not.

Entering an Element

When entering an element, the following rules are observed:

- **Contact**: in any column except the last two,
- **Coil**: in the second-last column (the last column is reserved for comments).

Entry procedure:

Steps	Description
1	Select the type of element required in the toolbar: Illustration  The list of available elements is displayed in the tool bar at the bottom of the edit window. When the mouse is moved over one of the elements, a dialog box appears displaying the list of available variables: ● The number of the element, ● The label of the element, ● The associated comment.
2	If necessary, enter a comment by clicking in the comment zone of the variable to use.
3	Drag the variable from the dialog box and drop it over a cell on the wiring sheet.

Deleting an Element

To delete an element, select the element and use one of the following methods:

- **Delete**,
- **Backspace**,
- right-click and select **Clear**,
- **Ctrl**+X.

Enter a Link

Description

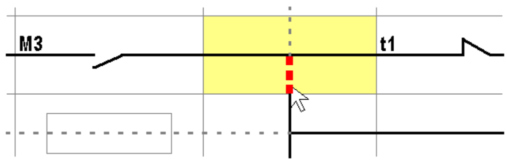
This section describes the procedures for performing the following operations:

- Entering links between elements,
- Deleting links between elements,
- Replacing a link with a contact.

Entering a Link

Links are entered exclusively in cells framed by dotted lines.

Entry procedure:

Steps	Description
1	Select the segment to transform, by placing the mouse pointer over it. Illustration: 
2	Click and hold the segment. It is validated and becomes red.
3	Release the mouse button: The segment is created.
4	Connect the elements of the wiring sheet by clicking on the dotted lines that separate them.

Deleting a Link

To delete the links between elements, click again on the link.

Replacing a Link with a Contact

To replace a link with a contact:

- Follow the element entry *(see page 213)* procedure,
- Place the contact over the segment to modify.

Automation Function Configuration

Description

When entering a control diagram, the parameters of the configurable automation functions must be completed:

- Discrete Outputs *(see page 144)*
- Auxiliary Relays *(see page 151)*
- Clocks *(see page 176)*
- Analog Comparators *(see page 192)*
- Timers *(see page 182)*
- Counters *(see page 157)*,
- Fast Counter *(see page 166)*
- Counter Comparators *(see page 164)*
- Texts *(see page 198)*

Direct access

Once the automation function is entered *(see page 213)* in the wiring sheet, double-click on it and the corresponding parameters window opens.

This window has two tabs:

- **Parameters**: These are the specific parameters associated with the variable.
- **Comments**: The associated comments.

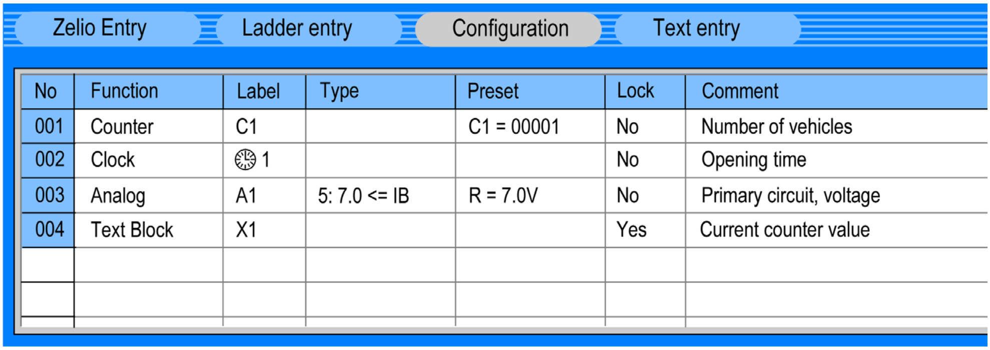
Access via the configuration interface

The **Configuration** mode allows you to list the automation functions with parameters used in the application. This mode is accessible from edit mode by clicking the **Configuration** tab.

You can view the following information:

- **Function**: Timer, Counter, etc.
- **Label**: Function block ID,
- **Type**: Counter type, timer type, etc.,
- **Preset**: The value to reach for a counter,
- **Lock**: Lock the parameters (prevent modification via the front panel),
- **Comment**: Comments associated with the function.

Illustration:


Zelio Entry | Ladder entry | Configuration | Text entry

No	Function	Label	Type	Preset	Lock	Comment
001	Counter	C1		C1 = 00001	No	Number of vehicles
002	Clock	1			No	Opening time
003	Analog	A1	5: 7.0 <= IB	R = 7.0V	No	Primary circuit, voltage
004	Text Block	X1			Yes	Current counter value

To adjust the parameters, double-click on the line.

Parameters in RUN Mode

In the programming software, in **RUN** mode (Simulation, Monitoring, Remote control (emulation) of the front panel), it is possible to modify the parameters (if they are not locked) via:

- the **PARAMETERS** menu on the front panel,
- on the edit sheet, right click on the function block,
- the function blocks command box,
- the supervision window.

List of authorized actions:

Automation functions	Authorized modification
Counter	Preset value.
Timer	The timing duration(s).
Clock	The range, Day of the Week (D/W), and ON/OFF parameters.
Analog	The reference (R) and hysteresis (H) voltages.

Insert and Delete a Program Line

Inserting Lines

Select the line to move down, or one of its cells and use one of the following methods:

- With the keyboard: Press the **Insert** key,
- With the mouse: **Right click/Insert a line** on the number of the line to move down,
- With the menus: Click on **Edit** → **Insert a line**.

Deleting Lines

To delete a line (or a cell), select the line (or a cell), then use one of the following methods:

- With the keyboard: Press the **Delete**, or **Backspace** key, or **Ctrl + X**,
- With the mouse: **Right click/Delete the line**, on the number of the line to delete,
- With the menus: Click on **Edit** → **Delete the line**.

Copy Parts of a Program

Description

It is possible to copy parts of the program:

Steps	Description
1	Select the elements to copy.
2	Right-click and select **Copy** to copy the elements to the clipboard (**Ctrl**+C).
3	Place the cursor over the recipient zone.
4	Right-click and select **Paste** to paste the elements contained in the clipboard (**Ctrl**+V).

NOTE: It is also possible to use the **Cut**, **Copy** and **Paste** commands from the **Edit** menu.

Check Program Consistency

Description

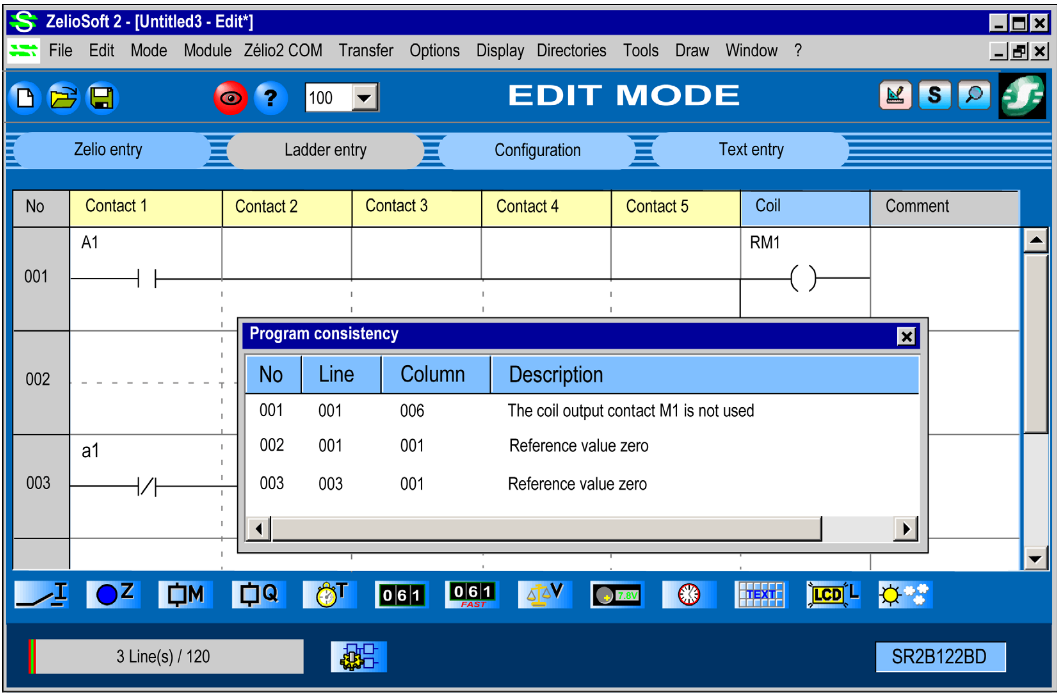
When entering the program, the programming software constantly verifies the consistency of the program, for instance:

- Incomplete lines,
- Non-connected Reset RX coils,
- Non-defined Preset values.

The LD network can be simulated, loaded and executed on the smart relay at any time. Hence it can be built and debugged progressively.

When inconsistencies are detected, the eye symbol changes from blue to red in the upper panel of the edit window.

Illustration



 As soon as the software detects a possible problem, the **Consistency check** icon becomes red. By clicking on it, you can display a dialog box.

The programs consistency window provides the following information:

- Error code,
- Location of the error: line and column,
- Description of the error.

By double-clicking on the error in the window, the position of the problem is highlighted on the wiring sheet.

These anomalies are intended to draw the your attention to singular wiring instances, which may nevertheless be justified in certain applications.

As a general rule, these anomalies correspond to incomplete wiring, either some inputs are not wired, for example a function Reset, parameters are not configured, or there are certain Clock configurations where the output stays ON permanently.

NOTE: Even if the eye is red, the program can still be simulated or executed. This allows for progressive debugging.

Section 20.2
Debugging an LD Application in the Programming Software

Subject of this Section

This section describes the different functions linked to debugging the application in the programming software in LD mode.

What Is in This Section?

This section contains the following topics:

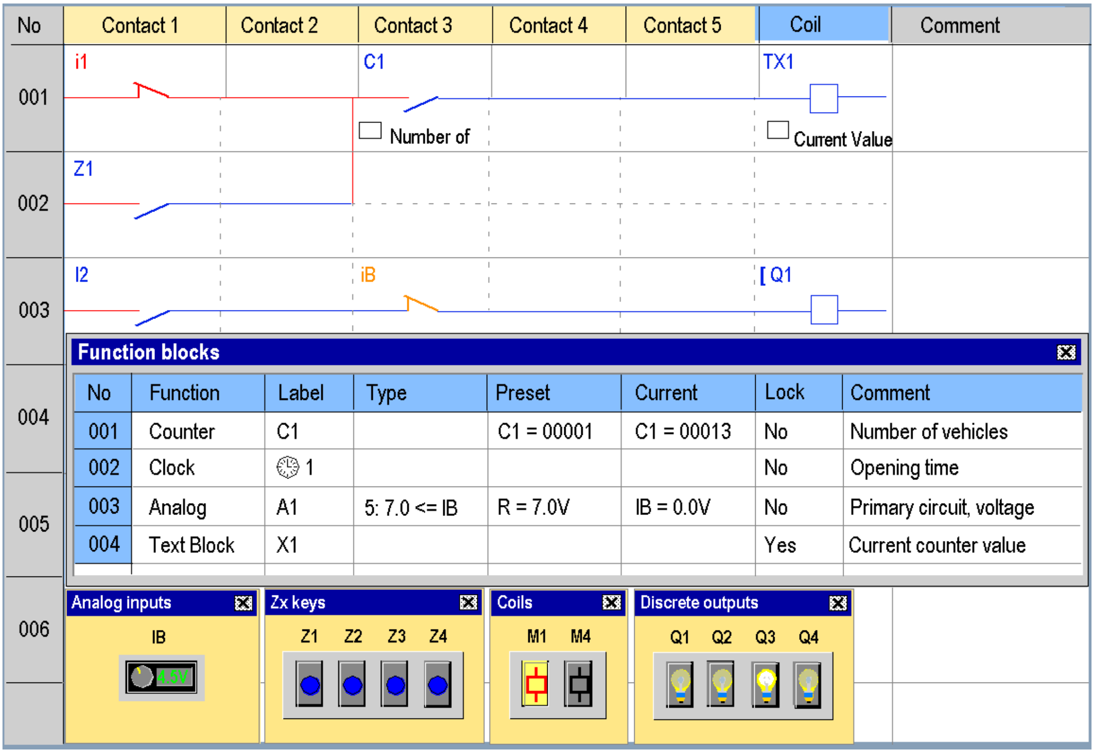
Topic	Page
Simulation of an Application	223
Monitoring of an Application	231

Simulation of an Application

Description

The **Simulation** mode is used to execute the program directly in the software (locally), as part of the application debug procedure.

Illustration:



For the simulation to work, you must switch to **RUN** mode using the corresponding icon.

In **RUN** mode, the active contacts are displayed:

- In red in Ladder entry *(see page 136)* mode,
- In reverse video in Zelio entry *(see page 135)* mode.

Contacts and coils may be displayed in orange if they are active but not supplied.

When switching from **RUN** to **STOP**, the automation functions switch back to zero. Only permanently forced contacts continue to be displayed (highlighted in red). However, in **STOP** mode, permanent or momentary forcing can be positioned in preparation for **RUN** mode.

In **RUN** mode, the following elements are simulated:

- Wiring sheet: Dynamic display (red) of the different active elements of the program.
- Input commands *(see page 226)*
- Auxiliary relay commands *(see page 227)*
- Output commands *(see page 228)*
- Z key commands *(see page 228)*
- Analog input control *(see page 229)*
- Viewing/modifying automation function parameters *(see page 225)*
- Clock simulation *(see page 229)*

The output contacts of the automation functions in the wiring sheet can be forced to test program behavior under specific conditions.

(See How to debug an application without loading it onto the smart relay: Simulation *(see page 37)*).

Access to Simulation Mode

Simulation is accessed by the **Mode** → **Simulation** menu or by using the  icon.

NOTE: By default the **Edit** window is displayed full screen, and the front panel and **Supervision** windows can be accessed:

- From the **Window** menu,
- By minimizing the wiring window.

Program Execution Parameters

NOTE: To display the functions described below, check the box in the **File** → **Preferences** menu: Display the refresh period and the number of cycles for Monitoring and Simulation.

(Refer to Simulation Mode Toolbar *(see page 41)*).

Refresh Period

This is the frequency at which the output values and parameters are updated in the application windows.

In order to be executed by the smart relay, this program is translated as a set of ordered instructions, where each instruction corresponds to a function in the program.

This instruction set is executed periodically, thus at regular time intervals. This time interval is called the **execution period**.

The refresh period for the input values and for the output values is set to `N` times the execution period.

Number of cycles

This corresponds to the number of cycles executed between each refresh of application windows.

Program Commands

Description of program command buttons in Simulation mode:

Active button	Description
	Launches program execution.
	Stops program execution.
	Pause / Run: Stops or relaunches the program flow. (only activated in **RUN** mode).
	Simulation of a power outage *(see page 68)*. (only activated in **RUN** mode).

The color of the icons changes according the application state.

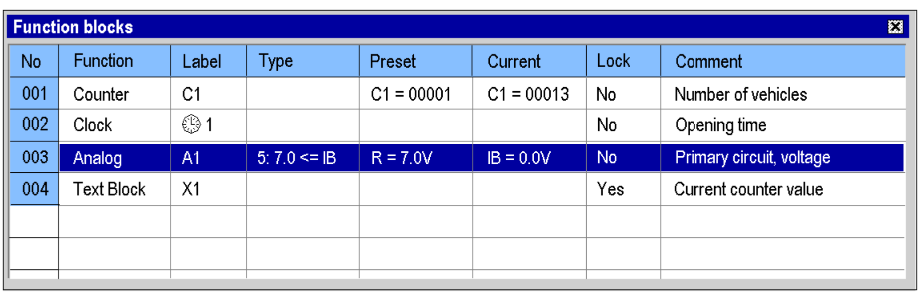
When it is possible to select the icon it is shown in yellow  .

Automation Function Parameters

From the function blocks command box

The  icon is used to display or hide the automation function parameter display box.

Illustration:



Function blocks

No	Function	Label	Type	Preset	Current	Lock	Comment
001	Counter	C1		C1 = 00001	C1 = 00013	No	Number of vehicles
002	Clock	1				No	Opening time
003	Analog	A1	5: 7.0 <= IB	R = 7.0V	IB = 0.0V	No	Primary circuit, voltage
004	Text Block	X1				Yes	Current counter value

In **Simulation** mode, you can:

- Display the values of the different parameters,
- Click on the function to modify the preset value or comment.

The following table shows, for each of the automation functions, what can be displayed or modified:

Automation functions	Display / Function blocks window	Authorized modifications
Counter	• Current value • Preset value • Lock	• Preset value • Count direction • Latching • Lock
Timer	• Timer type • Current value • Preset value • Lock	• Timer type • Timing duration(s) • Unit • Latching • Lock
Clock	• Lock	• Lock
Analog comparator	• Type of comparison • Reference voltage • Hysteresis value • Values measured on the analog inputs • Lock	• Type of comparison • Reference voltage • Hysteresis value • Inputs to be compared • Lock
Counter comparator	–	• Comparison operator • Offset value

From the wiring sheet

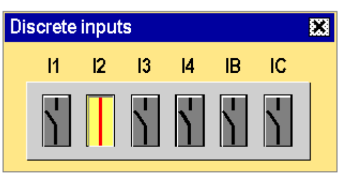
Position the cursor over the element to be modified, then right-click and select **Parameters** window.

Simulation of Discrete Inputs

From the Discrete inputs command box

The  icon is used to display or hide the input command box.

Illustration: input **I2** conducting



Possible actions:

- Permanent forcing: click.
- Momentary forcing: right click,

From the wiring sheet

Possible actions:

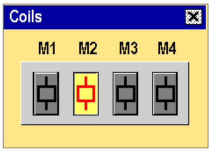
- Permanent forcing: click,
- Momentary forcing: right click,
- Forcing and maintain: right-click, which then locks the input (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this input until a release command has been performed.
- Release: right click,
- Release all: right click.

Simulation of Auxiliary Relays

From the Coils command box

The  icon is used to display or hide the coil command box.

Illustration: Coil **M2** active



Possible actions:

- Momentary forcing: left or right click.

From the wiring sheet

Possible actions:

- Permanent forcing: click,
- Momentary forcing: right click,
- Forcing and maintain: right-click, which then locks the coil (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this coil until a release command has been performed.
- Release: right click,
- Release all: right click.

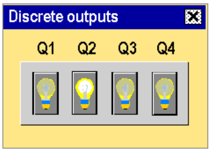
Simulation of Discrete Outputs

From the Discrete outputs command box

The  icon is used to display or hide the output command box.

If an output Q is active in Simulation, the corresponding bulb is lit. This bulb remains lit when a power outage is simulated.

Illustration: Output **Q2** active



Possible actions:

- Permanent forcing: left or right click.

From the wiring sheet

Possible actions:

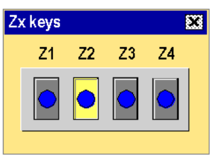
- Permanent forcing: click,
- Momentary forcing: right click,
- Forcing and maintain: right-click, which then locks the output (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this output until a release command has been performed.
- Release: right click,
- Release all: right click.

Simulation of Zx Keys

From the Zx keys command box

The  icon is used to display or hide the **Zx keys** command box.

Illustration:



Possible actions:

- Momentary forcing: left or right click.

From the wiring sheet

Possible actions:

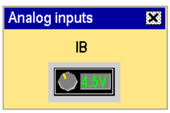
- Permanent forcing: click,
- Momentary forcing: right click,
- Forcing and maintain: right-click, which then locks the key (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this key until a release command has been performed.
- Release: right click,
- Release all: right click.

Simulation of Analog Inputs

The 

icon is used to display or hide the input command box.

Illustration:



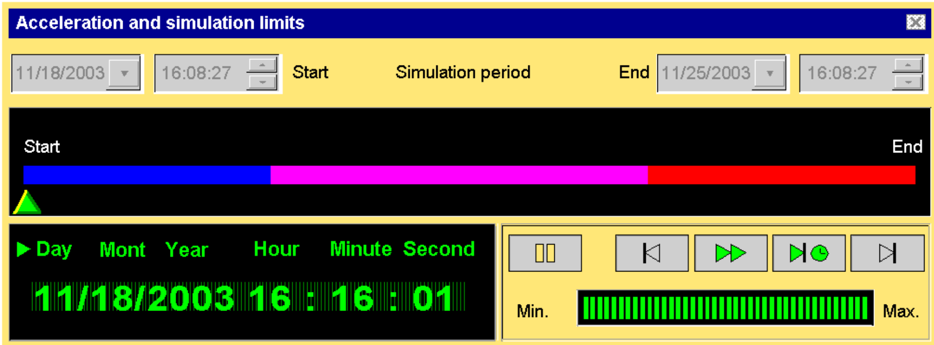
The analog value can be modified by adjusting the potentiometer (click).

Accelerator

The  icon is used to display or hide the accelerator box.

NOTE: This functionality has an influence on operation of the Clock *(see page 176)* function.

Illustration:



Description of Simulation Window:

- Entry and display of simulation period,
- Cursor allowing to change the time by moving the cursor (Stop mode only),
- Display of date and time in Simulation,
- Video-type control panel: Pause, Return to Start (Stop mode only), Fast Forward, Jump to Next Clock Event, End, Time acceleration period adjustment.

This window is displayed when you click on the simulation time smart relay icon situated in the bar at the bottom of the simulation window.

Display:

- Displays the date and time of the start and end of the simulation.

Actions:

- Can be used to modify the date and time of the start and end of the simulation (in **Stop** mode) in the **simulation limits** window.
- Can be used to accelerate the simulation speed up to 65000 times the original speed by clicking the  button and moving the level of the min-max bar.
- Can be used to place the cursor 3 s before the clock event by clicking on the  button.

Simulation of a Power Outage

During a power outage simulation, the simulator clock is frozen. To simulate power return at a given time, the simulation time must be changed. To do this, proceed as follows:

Step	Action
1	Click on the **Simulation → Set Clock** menu.
2	Enter the date and time for power return in the **Date** and **Time** fields of the **Set Clock** window.
3	Confirm by clicking on the **Write in the module** button.

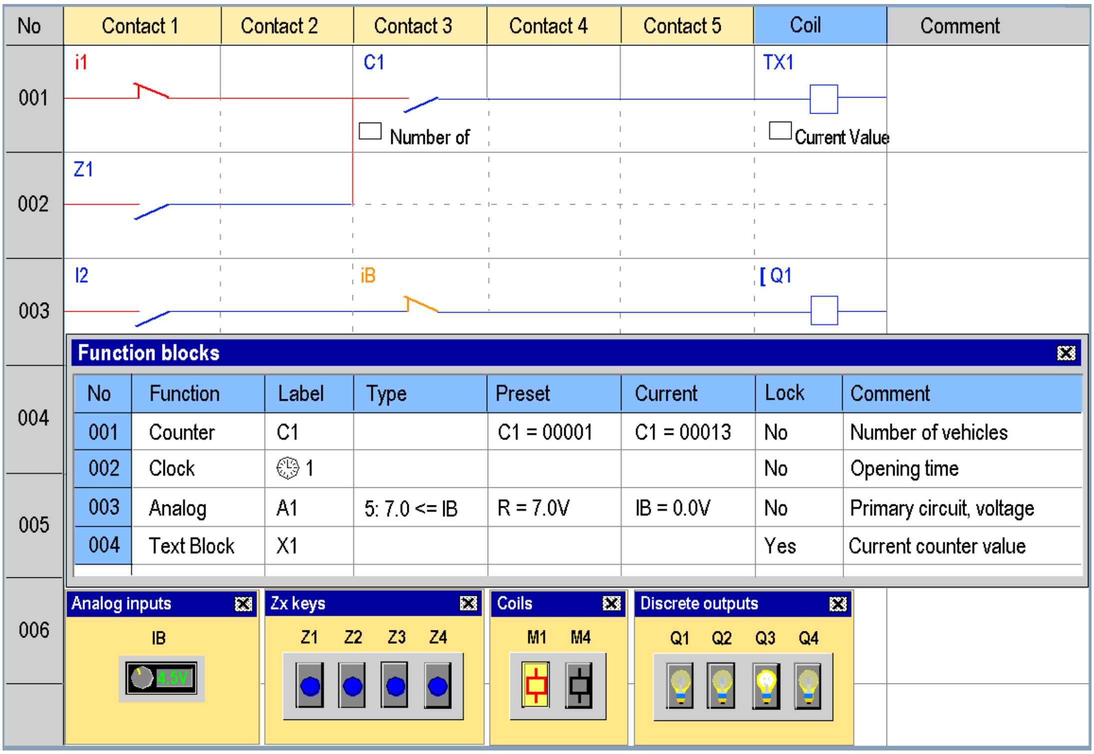
Monitoring of an Application

Description

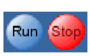
The monitoring function is used while running the program on the smart relay (in online mode) and to display its progress in the software (using a serial link).

The state of the different elements of the application (inputs / outputs and parameters) are updated on each program cycle.

Illustration:



For the monitoring to work, you must switch to **RUN** mode using the corresponding icon.



In **RUN** mode, the active contacts are displayed:

- In red in **Ladder entry** mode *(see page 136)*,
- In reverse video in **Zelio entry** mode *(see page 135)*.

Contacts and coils may be displayed in orange if they are active but not supplied.

When switching from **RUN** to **STOP**, the current automation functions switch back to zero.

In RUN mode, the following elements are displayed:

- Wiring sheet: Dynamic display (in red) of various active program elements.
- Input commands *(see page 233)*,
- Auxiliary relay commands *(see page 234)*,
- Output commands *(see page 235)*,
- Zx key commands *(see page 235)*,
- The viewing/modifying of automation function parameters *(see page 233)*.

It is possible to force certain states from the software and to display the internal states (up to 10 function block outputs simultaneously).

(See *How to monitor and modify an application running on the smart relay from the software: Monitoring (see page 43)*).

Access to Monitoring Mode

Monitoring is accessed by the **Mode** → **Monitoring** menu or by using the  icon.

The following scenarios may arise:

- An application is open in the software: The version on the smart relay is compared with that of the software:
  - If the software application is the same as the one on the smart relay, monitoring mode is started.
  - If the software application is different from the one on the smart relay, the versions must be synchronized by transferring the program from the PC to the smart relay or from the smart relay to the PC.
- No application is open in the software: In this case, the software offers to transfer the application from the smart relay to the PC.
  Once the transfer is complete, the supervision window is displayed.

Program Execution Parameters

NOTE: To display the functions described below, check the box in the **File** → **Preferences** menu: **Show the refresh cycle (simulation and monitoring) and the time between two simulation results**.

(See *Monitoring Mode Toolbar (see page 47)*).

Refresh Cycle

This is the frequency at which the I/O values and parameters are updated in the application windows.

The application program is executed periodically, thus at regular time intervals. This time interval is called the **execution period**.

The refresh cycle of the input values and the refresh cycle of the output values are set to `N` times the **execution period**.

Monitoring parameters

In monitoring mode, you can:

- display the values of the various parameters in the **Function Blocks** window
- click on the block to change the settings.

The following table shows, for each of the automation functions, what can be displayed or modified:

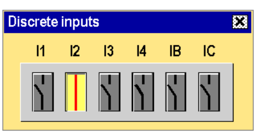
Automation functions	Display / Function blocks window	Authorized modifications
Counter	• Preset value • Lock	• Preset value • Count direction • Latching • Lock
Timer	• Timer type • Preset value • Lock	• Timer type • The timing duration(s) • Unit • Latching • Lock
Clock	• Lock	• Lock
Analog comparator	• Type of comparison • Reference voltage • Hysteresis value • Lock	• Type of comparison • Reference voltage • Hysteresis value • Inputs to be compared • Lock
Counter comparator	–	• Comparison operator • Offset value • Lock

Monitoring of Discrete Inputs

From the Discrete inputs command box

The  icon is used to display or hide the input command box.

Illustration: input **I2** conducting



Possible actions:

- Permanent forcing: click.
- Momentary forcing: right click,

From the wiring sheet

Possible actions:

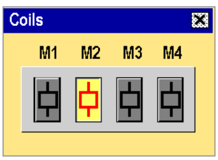
- Permanent forcing: click,
- Momentary forcing: right click,
- Force and maintain: right-click, which then locks the input (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this input until a release command has been performed.
- Release: right click,
- Release all: right click.

Monitoring of Auxiliary Relays

From the Coils command box

The  icon is used to display or hide the coil command box.

Illustration: coil **M2** active



Possible actions:

- Permanent forcing: left or right click.

From the wiring sheet

Possible actions:

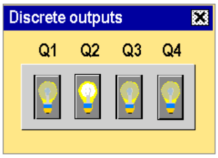
- Permanent forcing: click,
- Momentary forcing: right click,
- Force and maintain: right-click, which then locks the coil (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this coil until a release command has been performed.
- Release: right click,
- Release all: right click.

Monitoring of Discrete Outputs

From the Discrete outputs command box

The  icon is used to display or hide the output command box.

Illustration: output **Q2** active



Possible actions:

- Permanent forcing: left or right click.

From the wiring sheet

Possible actions:

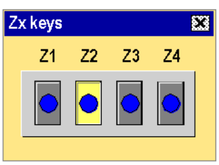
- Permanent forcing: click,
- Momentary forcing: right click,
- Force and maintain: right-click, which then locks the output (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this output until a release command has been performed.
- Release: right click,
- Release all: right click.

Monitoring of Z Keys

From the Zx keys command box

The  icon is used to display or hide the **Zx keys** command box.

Illustration:



Possible actions:

- Momentary forcing: left or right click.

From the wiring sheet

Possible actions:

- Permanent forcing: click,
- Momentary forcing: right click,
- Force and maintain: right-click, which then locks the key (highlighted in red) in the desired state: ON or OFF. When forced, no further action can be carried out on this key until a release command has been performed.
- Release: right click,
- Release all: right click.

Chapter 21
Example of an LD Application

Greenhouse Ventilation Panes

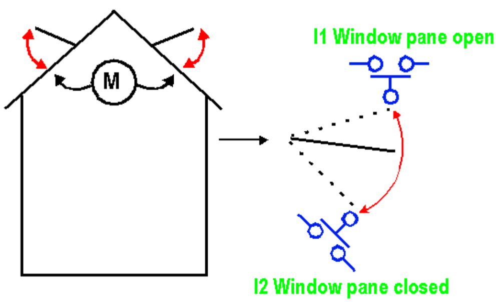
Description

This example describes how greenhouse ventilation panes can be managed automatically.

Specifications

The owner of a greenhouse would like to acquire an installation to manage the opening and closing of the ventilation window panes located on the greenhouse roof.

The greenhouse has two window panes to provide ventilation. The opening of these window panes is controlled by a motor and 2 sensors that indicate whether the window panes are open or closed:



During the day, the window panes open to ventilate the structure from 12:00 to 15:00, at the time of day when, in principle, the temperature is the highest. However, if the temperature is less than 10 °C, the window panes do not open, or when they are already open, they close.

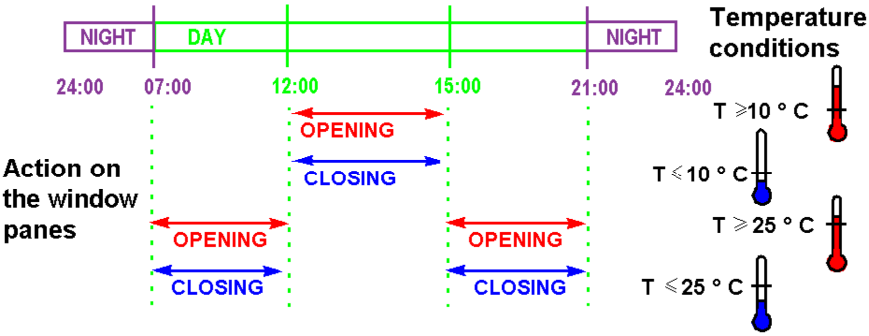
In addition, the window panes open during the day when the temperature reaches 25 °C. If the temperature falls below 25 °C, the window panes must close again.

Finally, at night, the window panes remain closed regardless of the temperature.

Program description, 3 time ranges are used:

- Range 1: Night, from 21:00 to 07:00
- Range 2: Day, from 07:00 to 12:00 and from 15:00 to 21:00
- Range 3: Noon, from 12:00 to 15:00

Summary:



Input/Output Table

Description of the inputs:

Input	Description
I1	Window panes open (Discrete)
I2	Window panes closed (Discrete)
IB	Temperature (analog)

Description of the outputs:

Output	Description
Q1	Opening of the window panes (Discrete)
Q2	Closing of the window panes (Discrete)

The temperature is supplied by a sensor with output voltage of 0 to 10 V.

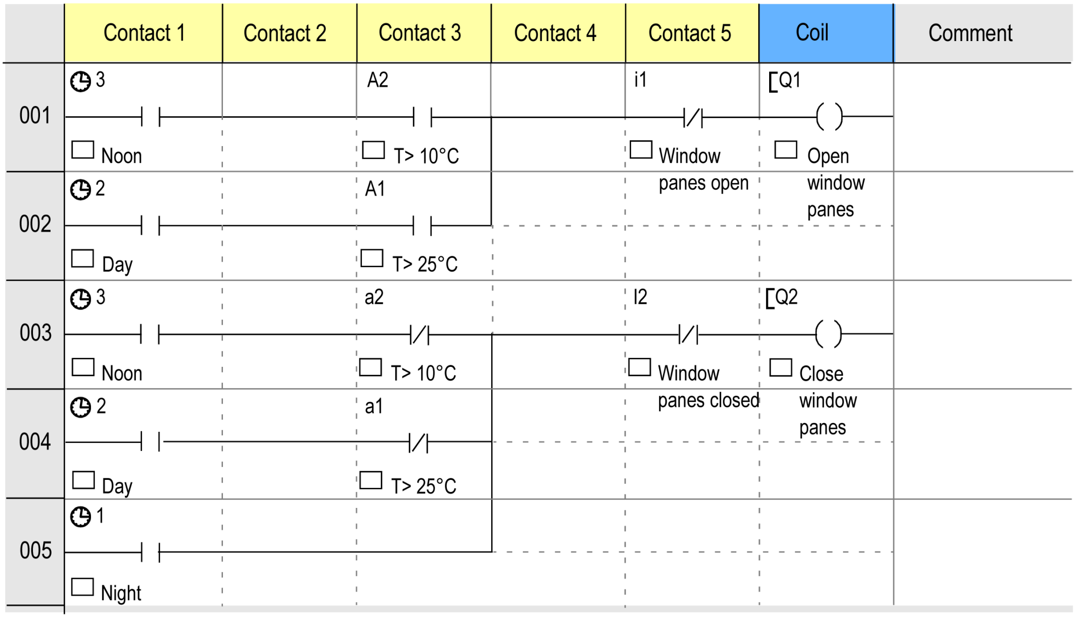
Required Reference

For this application, a smart relay with a clock and analog inputs is required:

- SR2B121BD (24 Vdc),
- SR2B122BD (24 Vdc),
- SR2B121JD (12 Vdc).

The LD Wiring Sheet

This figure shows the example with Ladder symbols display:



Description of the Parameters

Daily programmer H1:

Channel C:

The other channels (A, B, D) are not configured.

Daily programmer H2:

Channel C:

- ON: 07:00 OFF 12:00,
- The other parameters are the same as for programmer H1.

Channel D:

- ON: 15:00 OFF 21:00,
- The other parameters are the same as for programmer H1.

The other channels (A, B) are not configured.

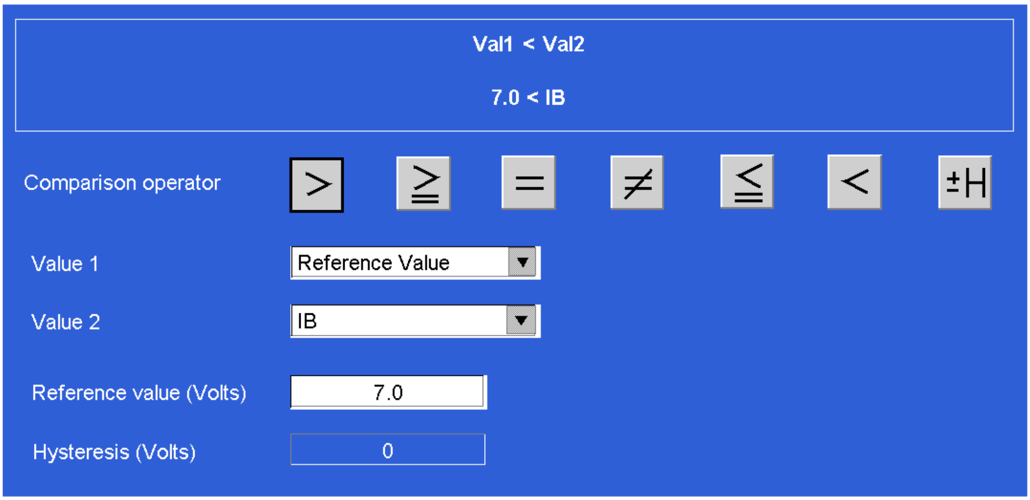
Daily programmer H3:

Channel C:

- ON: 12:00 OFF 15:00,
- The other parameters are the same as for programmer H1.

The other channels (A, B, D) are not configured.

Analog comparator a1



Analog comparator A2

- Reference value: 3 Volts.

The other parameters are the same as for programmer A1.

Part V
FBD Language

Subject of this Section

This section describes the use of FBD (Functional Block Diagram) programming language for the smart relay.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
22	Overview of FBD Language	243
23	FBD Language Elements	249
24	Programming in FBD Using Zelio Soft 2	393
25	Example of an FBD Application	439

Chapter 22
Overview of FBD Language

Subject of this Chapter

This chapter provides a general description of FBD language.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
FBD Program Edit Window	244
Function Bar	247

FBD Program Edit Window

At a Glance

FBD mode allows graphic programming based on the use of predefined function blocks.

In FBD programming, there are three types of windows:

- The **Edit** window.
- The **Supervision** window *(see page 246)*.

Edit Window

FBD programs are created in the edit window. This window can be accessed from the **Mode** → **Edit** menu or by using the **Edit** button  on the toolbar.

The edit window is made up of three zones:

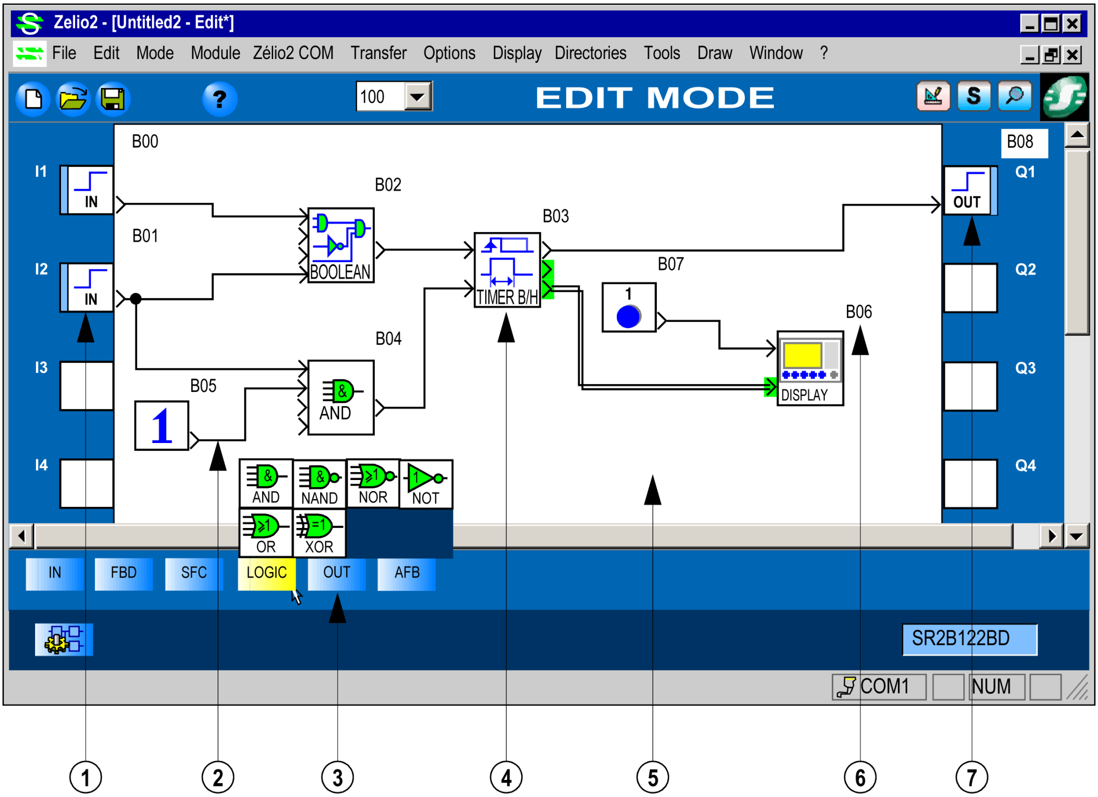
- The wiring sheet, where the functions that make up the program are inserted.
- The Inputs zone on the left of the wiring sheet where the inputs are positioned.
- The Outputs zone on the right of the wiring sheet where the outputs are positioned.

The inputs/outputs are specific to the type of smart relay and extensions chosen by the user.

The program in the edit window corresponds to the program that is:

- Compiled.
- Transferred into the smart relay.
- Compared to the contents of the smart relay.
- Used in simulation mode.
- Used in supervision mode.

The following figure shows an example of a part of an edit window in FBD language:

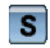


1 Function block input zone
2 Connection between two function blocks
3 Function bar
4 Function block
5 Wiring sheet
6 Function block number
7 Output function block zone

Supervision/Monitoring Window

The supervision/monitoring window is a subset of the **Edit** window.

It can be accessed from:

- **Simulation**: The **Mode/Simulation** menu or using the simulation button  on the toolbar.
- **Monitoring**: The **Mode/Monitoring** menu or using the monitoring button  on the toolbar.

It contains the functions, without their connections, that the programmer extracted (using Drag/Drop or **Copy**/**Paste**) from the **Edit** window.

The window can also contain drawings *(see page 408)*, text and images.

In **Simulation** and **Monitoring** mode, the parameters and outputs of the functions present are updated.

Function Bar

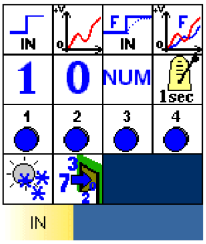
At a Glance

To create an FBD program, the different functions to be inserted in the wiring sheet are available in a function bar. Each of the tabs in the function bar groups a function type.

When the mouse is moved over one of the tabs, the dialog box displays the list of available variables.

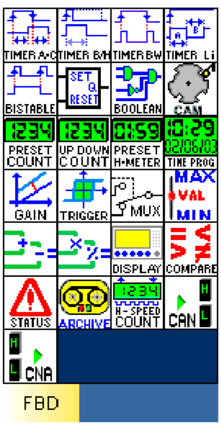
Inputs Bar

The following figure shows the inputs *(see page 250)* bar:



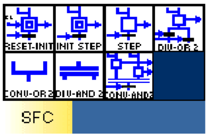
Standard Functions Bar

The following figure shows the Standard Functions *(see page 272)* bar:



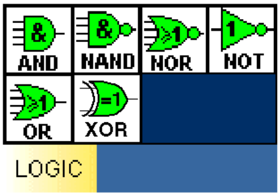
SFC Functions Bar

The following figure shows the SFC Functions *(see page 348)* bar:



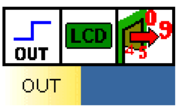
Logic Functions Bar

The following figure shows the Logic Functions *(see page 270)* bar:



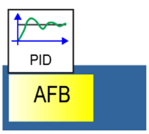
Outputs Bar

The following figure shows the Outputs *(see page 261)* bar:



AFB Bar

The following figure shows the Application Functions Blocks *(see page 384)* bar:



Chapter 23
FBD Language Elements

Subject of this Chapter

This chapter describes the different elements of the FBD language.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
23.1	Different Input Blocks	250
23.2	Different Output Blocks	261
23.3	Modbus Inputs/Outputs	267
23.4	Ethernet Inputs and Outputs	269
23.5	Logic Functions	270
23.6	Standard Functions	272
23.7	SFC Functions	348
23.8	Application Function Blocks	384

Section 23.1
Different Input Blocks

Subject of this Section

This section describes the different input blocks available using FBD language.

What Is in This Section?

This section contains the following topics:

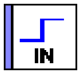
Topic	Page
Discrete Inputs	251
Filtered Discrete Input	253
Analog Input	254
Filtered Analog Input	256
Integer Input	257
Special Inputs in FBD Language	258
10-Bit Integer Input	260

Discrete Inputs

At a Glance

The **Discrete Input** is available for all smart relays. The Discrete inputs can be arranged at any smart relay input.

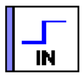
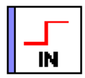
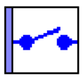
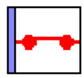
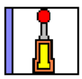
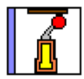
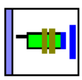
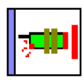
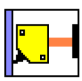
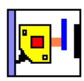
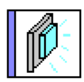
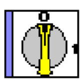
Access

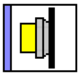
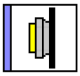
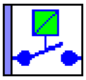
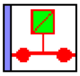
The **Discrete Input** 

function is accessible in the **IN** function bar.

Type of Discrete Inputs

The type of Discrete input can be selected from the **Parameters** window. This is then displayed in the **Edit** and **Supervision** windows.

Type	Display in the Inactive state	Display in the Active state
Discrete input		
Contact		
Limit switch		
Proximity sensor		
Presence sensor		
Illuminated pushbutton		
Selector switch		

Type	Display in the Inactive state	Display in the Active state
Pushbutton		
Normally open relay		

Simulation and Monitoring Modes

In **Simulation** or **Monitoring** modes, it is possible to force Discrete inputs. In this case, the input symbol is displayed as shown in the above table.

Filtered Discrete Input

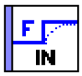
At a Glance

Behind the Discrete input, a filter is added to reduce or even eliminate disturbances such as contact bounce or momentary state changes of the input.

A Discrete input is filtered using a constant level detection algorithm (1 or 0) on the "sensor" signal, measured over a certain time frame. If the signal is stable throughout the detection period, the output of the symbol from the filtered Discrete input takes the value of the measured signal. Otherwise it remains unchanged.

The filtered Discrete inputs can be arranged at any smart relay input.

Access

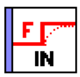
The **Filtered Discrete input** 

function is accessible from the **IN** window.

Parameter

The value of the parameter (between 1 and 255) entered in the **Parameters** window can be used to define the minimum time during which the signal must be stable. This value is a multiple of the smart relay's cycle time.

Simulation and Monitoring Modes

In Simulation or Monitoring modes, it is possible to force filtered Discrete inputs. In this case, the input symbol is displayed as follows 

.

Analog Input

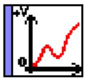
At a Glance

The **Analog input** is available on smart relays supplied with DC voltage.

The analog input voltage is converted into an integer digital value by a 8-bit analog/digital converter. The whole output value is between 0 and 255.

The analog inputs can only be connected to inputs between IB and IG.

Access

The **Analog input**  function is accessible from the **IN** window.

Parameter

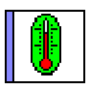
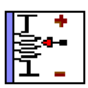
By default, this voltage varies between 0 and 10 Vdc.

The type of electrical connection at the input can be configured in the **Parameters** window:

- 0 - 10 V,
- The potentiometer option is selected if the input is connected to a potentiometric device powered between 0 Volts and the voltage of the smart relay.

Analog input types

Analog input types to be displayed in the edit and supervision windows can be selected from the **Parameters** window.

Type	Display in edit mode
Input (by default)	
Input	
Temperature	
Potentiometer	

Simulation and Monitoring Modes

In Simulation or Monitoring modes, you can force (between 0 and 255) the output of the analog inputs.

Filtered Analog Input

At a Glance

Behind the analog input, a **low pass** filter is added. This function is available on smart relays supplied with a DC voltage.

The analog input voltage is converted into an integer digital value by a 8-bit analog/digital converter. The whole output value is between 0 and 255.

The analog inputs can only be arranged on the inputs between IB and IG.

Low Pass Filter

A **low pass** filter restores the input signal (frequency, amplitude and phase-shift), whose frequency is low, to a typical filter frequency, called a **cut-off frequency**. When the frequency of the input signal approaches the **cut-off frequency**, the output signal of the same frequency becomes increasingly lower and phase-shifted. When the frequency of the input signal is equal to the **cut-off frequency**, the output signal is lowered by approximately 30%, and phase-shifted by 45°. For a frequency superior to the **cut-off frequency**, the reduction is greater (until it reaches total elimination) and the phase-shifting approaches 90°.

Access

The **Filtered Analog input**  function is accessible from the **IN** window.

Parameters

The **Parameters** window is used to define:

- The input voltage. By default, this voltage varies between 0 and 10 Vdc.
  The potentiometer option is selected if the input is connected to a potentiometric device powered between 0 Volts and the voltage of the smart relay.
- The **cut-off frequency** of the **low pass** filter (between 0.06 and 88.25 Hz).

⚠ CAUTION
INCORRECT CUT-OFF FREQUENCY Verify and adjust if necessary the cut-off frequency whenever the cycle time is modified. **Failure to follow these instructions can result in injury or equipment damage.**

Integer Input

At a Glance

This function is used to enter a 16-bit (-32768, +32767) integer from the inputs of certain extensions connected to the smart relay.

Integer inputs can be positioned on the J1XT1 to J4XT1inputs of the extension modules.

Access

The **Integer input**  is accessible from the **IN** window.

Special Inputs in FBD Language

At a Glance

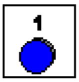
In FBD, various special inputs are available:

- Button
- Discrete constants
- Numerical constants
- Summer time
- Flashing for 1 second

These inputs can be accessed from the **IN** window.

These inputs cannot be inserted in the input squares of the diagram sheet.

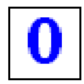
Button-type Inputs



Button-type inputs correspond to the keys available on the front panel of the smart relay. These inputs can be inserted in an FBD diagram and, in Simulation and Monitoring modes, can simulate contacts.

Discrete Constant-Type Inputs

There are two types of Discrete constants: The `1` constant 

and the `0` constant 

. These two constants can be used to set the function inputs to 1 or 0.

In Simulation or Monitoring modes, you can force these inputs in the reverse order. The symbol then appears in red.

Numerical Constant-Type Inputs

The numerical constant `NUM` 

is an integer with a value between -32768 and +32767.

This constant can be used to set values to the functions' non-connected inputs:

- GAIN
- COMP IN ZONE
- TRIGGER

The value of the constant can be set in the **Parameters** window.

In Simulation or Monitoring modes, it is possible to modify the constant.

Summer Time Input

The summer time input function is active  throughout summer time, and inactive throughout winter time  .

NOTE: To confirm this function:

- Display the **Program configuration** window: **Edit** → **Program configuration**.
- Select the **Date format** tab.
- Check the **Activate the summer/winter time change** box.
- Define the dates when the time change takes place:
  - Either using one of the predefined geographic zones.
    or
  - By manually configuring the date (month/Sunday).

Flashing Input

The flashing input function is active every second. Its active symbol is 

and its inactive symbol is 

.

10-Bit Integer Input

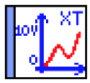
At a Glance

10-bit integer inputs are available on smart relays that are compatible with SR3XT43BD analog input-output extensions.

10-bit integer inputs may be arranged only on IH XT2 and IJ XT2 input contacts of the SR3XT43BD analog input-output extensions.

The analog input voltage is converted into a whole numerical value by a 10-bit analog/digital converter. The whole output value of the converter is between 0 and 1023.

Access

The 10-bit integer input function 

is accessible from the **IN** window.

Parameters

The 10-bit integer inputs are configured from the **Analog Extension** tab of the **Program Configuration** window, see *Analog Input-Output Extension SR3XT43BD, page 485*.

Simulation and Monitoring Modes

In Simulation or Monitoring modes, you can force (between 0 and 1023) the output of the analog inputs.

Section 23.2
Different Output Blocks

Subject of this Section

This section describes the different output blocks available using FBD language.

What Is in This Section?

This section contains the following topics:

Topic	Page
Discrete Output	262
Integer Output	264
LCD Screen Backlighting Output	265
SR3XT43BD Extension 10-Bit Integer Output	266

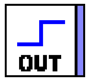
Discrete Output

At a Glance

The smart relays feature two types of Discrete outputs:

- **Solid-state** outputs for certain smart relays supplied with DC voltage.
- **Relay** outputs for smart relays supplied with AC or DC voltage.

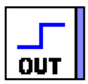
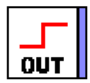
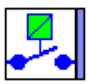
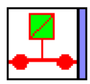
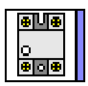
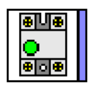
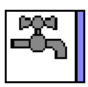
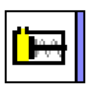
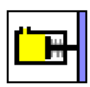
Access

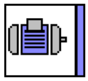
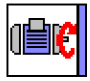
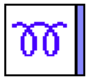
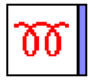
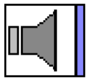
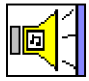
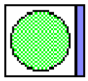
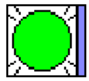
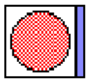
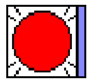
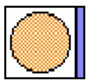
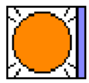
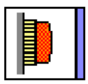
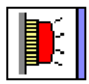
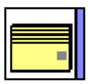
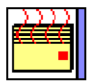
The Discrete output function 

is accessible from the **OUT** window.

Types of Discrete Outputs

The type of Discrete output can be selected from the **Parameters** window. This is then displayed in the edit and supervision windows. The selection is made using the output's inactive-state symbol.

Type	Display in the Inactive state	Display in the Active state
Discrete Output		
Normally open relay		
Lamp		
Solid state relay		
Valve		
Cylinder		

Type	Display in the Inactive state	Display in the Active state
Motor		
Resistance		
Audible signal		
Green indicator lamp		
Red indicator light		
Orange indicator light		
Indicator light		
Heating		
Fan		

Simulation and Monitoring Modes

In Simulation or Monitoring modes, outputs are displayed in the active or inactive state with their corresponding symbols (shown in the table above).

Integer Output

At a Glance

This function is used to create a 16-bit (-32768, +32767) integer output towards the integer outputs of certain extensions connected to the smart relay.

Integer-type outputs can be positioned on the O1XT1 to O4XT1 outputs of the extension modules.

NOTE: If the function input is not connected, the output is 0.

Access

The **Integer output**  function is accessible from the **OUT** window.

LCD Screen Backlighting Output

At a Glance

The **LCD Screen Backlighting Output** is used to control the lighting of the smart relay LCD for each program.

As long as the connected input is active, the backlighting is on.

This function cannot be arranged on the smart relay outputs.

Access

The **LCD Screen Backlighting Output** 

function is accessible from the **OUT** window.

Simulation and Monitoring Modes

The following table lists the symbols of the **LCD Screen Backlighting** function in Simulation or Monitoring modes.

Input State	Symbol in Simulation and Monitoring mode	Description
Inactive		The LCD screen is off.
Active		The LCD screen is back-lit.

SR3XT43BD Extension 10-Bit Integer Output

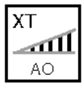
At a Glance

10-bit integer outputs are available on smart relays that are compatible with SR3XT43BD analog input-output extensions.

10-bit analog outputs may be arranged only on QB XT2 and QC XT2 output contacts of the SR3XT43BD analog input-output extensions.

A 10-bit analog/digital converter converts the integer digital value into an output voltage. The analog voltage varies between 0 and 10 V (1023 is equivalent to 10 V).

Access

The **10-bit integer output** function 

is accessible from the **OUT** window.

Parameters

The 10-bit integer outputs are configured from the **Analog Extension** tab of the **Program Configuration** window which can be accessed by clicking on the **XSR3XT43BD** button, see *Analog Input-Output Extension SR3XT43BD, page 485*.

Section 23.3
Modbus Inputs/Outputs

Modbus Inputs-Outputs

Description

A Modbus **SR3MBU01BD** extension module may be added to a **SR3BxxxBD**-type smart relay.

In FBD mode, the 4 (16 bits) input words (from J1XT1 to J4XT1) and the 4 output words (from O1XT1 to O4XT1) can be accessed by the application.

NOTE: The Modbus extension module only operates in Modbus slave mode.

Parameters are set in the workshop, using the: **Edit → Program Configuration** menu, **MODBUS Extension** tab, or by clicking on the  icon.

When changing to RUN mode, the smart relay initializes the Modbus Extension.

The smart relay has 4 parameters:

- Number of UART wires and frame format on the Modbus network,
- Data transmission speed in baud.
- Protocol parity,
- Slave Modbus extension network address.

Modbus Inputs

The Modbus **SR3MBU01BD** extension has 4 (16 bits) inputs:

- **J1XT1** (Hexa 0010 address),
- **J2XT1** (0x0011),
- **J3XT1**(0x0012),
- **J4XT1**(0x0013).

Data downloaded from the master.

Modbus Outputs

The Modbus **SR3MBU01BD** extension has 4 (16 bit) outputs:

- **O1XT1** (Hexa 0x0014 address),
- **O2XT1** (0x0015),
- **O3XT1**(0x0016),
- **O4XT1**(0x0017).

These data are sent to the master.

Section 23.4
Ethernet Inputs and Outputs

Ethernet Inputs-Outputs

Description

An **SR3NET01BD** Ethernet extension may be added to **SR3B261BD**, **SR3B262BD**, **SR3B101BD** and **SR3B102BD** smart relays.

In FBD mode, if the extension is present, then the application can use the four inputs J1 to J4 and the four outputs O1 to O4. Each I/O represents a 16-bit Modbus word.

The Ethernet extension *(see page 488)* exchanges Modbus messages as a server and by using the suite of TCP/IP protocols and Ethernet technology.

Parameters

Parameters are set in the software using the **SR3NET01BD** button in the title block *(see page 495)*.

Ethernet Inputs

The Ethernet extension has four 16-bit inputs:
- **J1** (address 16),
- **J2** (17),
- **J3** (18),
- **J4** (19).

These words may be accessed in read and write modes.

Ethernet Outputs

The Ethernet extension has four 16-bit outputs:
- **O1** (address 20)
- **O2** (21),
- **O3** (22),
- **O4** (23).

These words may be accessed in read mode only.

Section 23.5
Logic Functions

Logical functions

At a Glance

In FBD language, it is possible to use logic functions in the block diagrams. The available functions are:

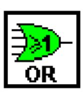
- The **NOT** function,
- The **AND** function,
- The **OR** function,
- The **NAND** function,
- The **NOR** function,
- The **EXCLUSIVE OR** function.

Access

These inputs can be accessed from the **LOGIC** window.

Logical functions

The following table shows the various logic functions:

Function	Symbol	Description	Number of inputs	Input type
NOT		If the input is inactive or not connected, the output is active. If the input is active, the output is inactive.	1	Digital
AND		If the inputs are active or not connected, the output is active. If at least one input is inactive, the output is inactive.	4	Digital
OR		If at least one input is active, the output is active. If the inputs are inactive or not connected, the output is inactive.	4	Digital
NAND		If at least one input is inactive, the output is active. If the inputs are active or not connected, the output is inactive.	4	Digital

Function	Symbol	Description	Number of inputs	Input type
NOR		If the inputs are inactive or not connected, the output is active. If at least one input is active, the output is inactive.	4	Digital
EXCLUSIVE OR		If an input is inactive and the other input is active or not connected, the output is active. If both inputs are active or inactive or not connected, the output is inactive.	2	Digital

Section 23.6
Standard Functions

Subject of this Section

This section describes the different standard functions available using FBD language.

What Is in This Section?

This section contains the following topics:

Topic	Page
BOOLEAN Equation (Boolean function)	274
SET and RESET Function	275
PRESET COUNT Up/Down Counter	276
H-SPEED COUNT (Fast Counter)	281
UP/DOWN COUNT (Up/Down Counter)	287
TIMER A/C (Timer)	289
TIMER BW (Pulses on Edges)	293
TIMER Li (Cyclic Timing)	294
TIMER B/H (Time out)	297
COMP IN ZONE Comparison	300
PRESET H-METER (Preset Hour Counter)	302
TRIGGER (Schmitt Trigger)	303
COMP IN ZONE (Comparison of two values)	305
GAIN Function	306
DISPLAY (LCD Screen Display)	308
TEXT	311
TIME PROG (Daily, weekly, yearly programmer)	314
BISTABLE (Impulse Relay)	318
MUX (Multiplexing)	319
ADD/SUB (ADD/SUB Arithmetic Function)	320
MUL/DIV (MUL/DIV Arithmetic Function)	322
CAM BLOCK (Cam Programmer)	324
ARCHIVE	327
STATUS	329
CNA (Bit to Word Conversion)	331

Topic	Page
CAN (Word to Bit Conversion)	332
SLIn (Serial Port Input)	333
SLOut (Serial Port Output)	335
COM (Message)	337
Sunrise/Sunset	344
Suntrack	346

BOOLEAN Equation (Boolean function)

At a Glance

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 figure shows an example of part of the **Boolean function** truth table:

Input 1	Input 2	Inputs 3	Input 4	Output
0	0	0	0	1
1	0	0	0	1
0	1	0	0	0
1	1	0	0	1

Combinations of inputs

Output values

Access

This 

function is accessible from the **FBD** function bar.

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.

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

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

SET and RESET Function

At a Glance

The **SET and RESET** function operates as follows:

- Activation of the **SET** input activates the output, which remains so even if the **SET** input is then deactivated,
- Activation of the **RESET** input deactivates the output,
- If both inputs are active, the state of the output depends on the configuration of the function:
  - The output is active if the **SET Has Priority** option is configured,
  - The output is inactive if the **RESET Has Priority** option is configured.

Non-connected inputs are set to the **Inactive** state.

Access

This 

function is accessible from the **FBD** function bar.

PRESET COUNT Up/Down Counter

Description

The **Preset Up/Down counter** function is used to up-count from 0 to the preset value, or to down-count from this value to 0.

Several functions are available:

- Up-counting *(see page 278)* and forcing the counter to 0 on initialization,
- Up-counting *(see page 279)* and forcing the counter to 0 on initialization and when the count value has been reached,
- Down-counting *(see page 278)* and forcing the counter to the preset value on initialization,
- Down-counting *(see page 280)* and forcing the counter to the preset value on initialization and when 0 has been reached.

Access

This function 

is accessible from the **FBD** function bar.

Inputs/Outputs

The up/down counter uses:

- A discrete **Up-Count** input,
- A discrete **Down-Count** input,
- A discrete **Initialization** input.

The up/down counter provides:

- A Discrete-type **Output**,
- The **Preset value** (1),
- The **Current Counter Value** (1),
- The output **timer value** (1).
  (1) these integer values are displayed in **Simulation** and **Monitoring** mode.

Parameters

In the programming software

From the **Parameters** window, you can adjust:

- The **Upcounting to the preset value** or **Downcounting from the preset value**,
- The **Preset** or **Setpoint** value (**1**),
- The **Single** cycle for initializing the counter only on initialization,
- The **Repetitive** cycle for initializing the counter on initialization, and when the current count value reaches 0 or the preset value.

For the **Repetitive** cycle, the **Duration of the pulse** (x 100ms) corresponding to the time during which the output is Active.

Following a Power Outage *(see page 68)*, the **Latching** parameter, if selected, enables the current value of the timer to be retrieved.

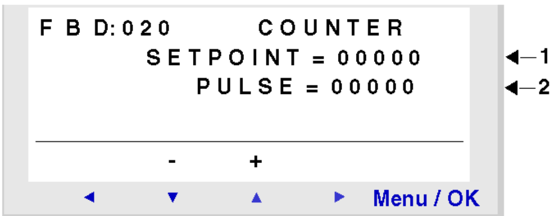
From the front panel

From the **PARAMETERS** menu *(see page 91)*, you can adjust:

- The **Preset** or **Setpoint** value (**1**),
- The **pulse duration** (for a repetitive cycle) (**2**).

Illustration

Illustration: counter parameters



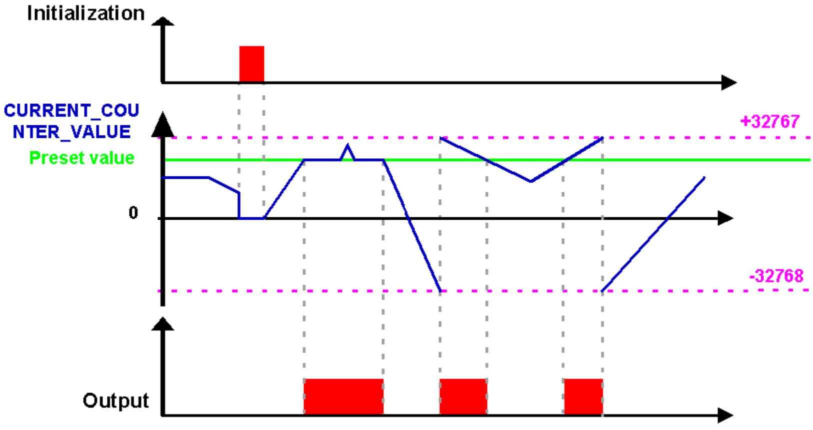
Parameter lock

Locking prevents modification of the parameters of the locked function block from the front panel of the smart relay using the **PARAMETERS** menu.

Up-Counting in Single Cycle Mode

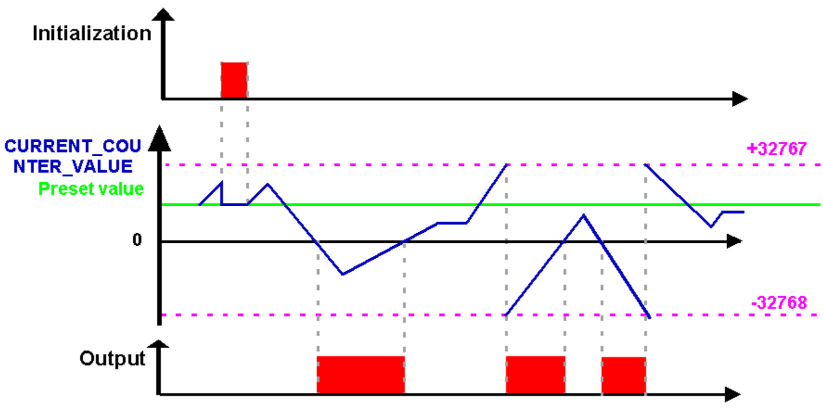
For the following four charts, the blue curve represents the internal counter value, when it increases there are pulses on the upcount input and when it decreases, pulses on the downcount input.

The following diagram shows the operation of the counter with initialization at 0:



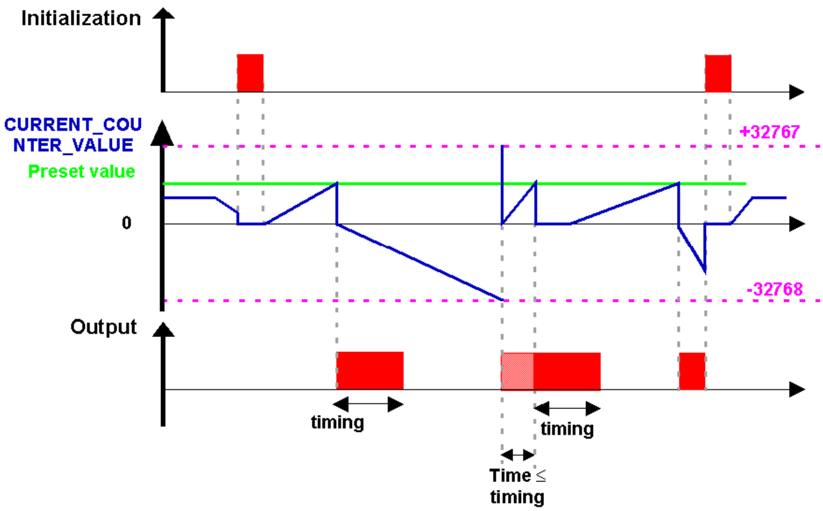
Down-Counting in Single Cycle Mode

The following diagram shows the operation of the down-counter with initialization at the preset value:



Up-counting in Repetitive Cycle Mode

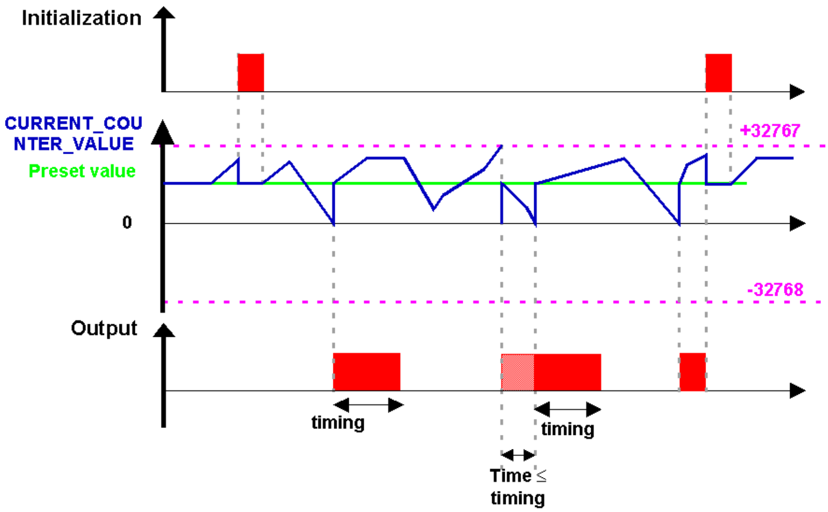
The following diagram shows the operation of the counter with forcing to 0 of the current value on initialization, or when the count value has reached the preset value:



The output switches to the Inactive state when the predefined pulse duration value has run out. If the switch condition is Active before switching to the Inactive state, the output pulse is extended by the **Duration of pulse** (Timing).

Down-Counting Function in Repetitive Cycle Mode

The following diagram shows the operation of the down-counter with forcing to the preset value of the current value on initialization, or when the count value has reached 0:



The output switches to the Inactive state when the predefined pulse duration value has run out. If the switch condition is Active before the switch to the Inactive state, the output pulse is extended by the **Duration of pulse** (Timing).

H-SPEED COUNT (Fast Counter)

Description

The **Fast Counter** function is used to count pulses up to a frequency of 1 kHz.

The counter **Output** indicates whether:

- The preset value has been reached (upcounting),
- The value 0 has been reached (downcounting).

The fast counter inputs are implicitly connected to the **I1** and **I2** smart relay inputs:

- A pulse (rising edge) on the **I1** input increments the counter,
- A pulse (rising edge) on the **I2** input decrements the counter.

These inputs are not used on the wiring sheet.

The Fast Counter function can be reset to zero or the preset value (depending on the parameter selected) during use by using the **Initialization** input.

The counter functions only if the **Enable function** input is active.

Repetitive mode can be used with a time-delay value.

NOTE: If the value of the counter exceeds the upper limit: **+32767**, it goes to **-32768**.
If the value of the counter exceeds the lower limit: **-32767**, it goes to **+32768**.

NOTE: This function block cannot be simulated.

Access

This function 

is accessible from the **FBD** function bar.

Inputs/Outputs

The up/down counter uses:

- A Discrete-type **Enable function** input.
- A discrete **Initialization** input.
- 2 inputs implicitly connected to the **I1** and **I2** smart relay inputs.

The up/down counter provides:

- A Discrete-type **Output**,
- The **Preset value** (1),
- The **Current Counter Value** (1),
- The **Current timer value** for output (1).
  (1) these integer values are displayed in Simulation and Monitoring mode.

Settings from the Software

Type of counting

Two modes are available:

- Upcounting to the preset value: incrementation of the counter value,
- Downcounting from the preset value: decrementation of the counter value.

Preset

This value is between 0 and 32767 (preset value).

Cycle type

Two modes are available:

- **Single cycle** *(see page 283)*: the counter value changes on an on-going basis.
  The output is activated when the counter value is greater than the preset value (counting mode) or when the counter value is less than the preset value (counting mode),
- **Repetitive cycle** *(see page 285)*: the counter value of the counter is reinitialized during counting when the value reaches the preset value or 0.
  The output is enabled following this reinitialization and remains active during a period of time that can be configured with the parameter: **Duration of pulse** (from 1 to 32,767 times 100 ms).

Latching

By default, after a power outage, the counter is set to the state that corresponds to program initialization.

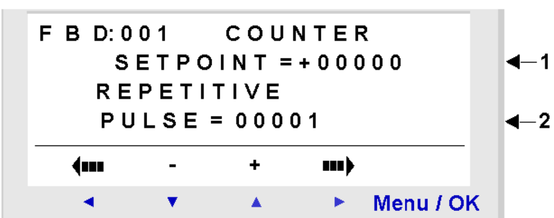
To restore the state of the counter backed up on power outage, activate latching.

Configuration from the Front Panel

From the **PARAMETERS** *(see page 91)* menu, you can adjust:

- The **setpoint** value,
- The **DURATION OF PULSE** (for a repetitive cycle).

Illustration: Parameter screen for a fast counter in Zelio entry / Front panel mode:



Description:

Number	Parameter	Description
1	**Setpoint**	Counter preset value.
2	**Duration of pulse**	Only if the cycle is repetitive

NOTE: This type of cycle cannot be modified from the front panel.

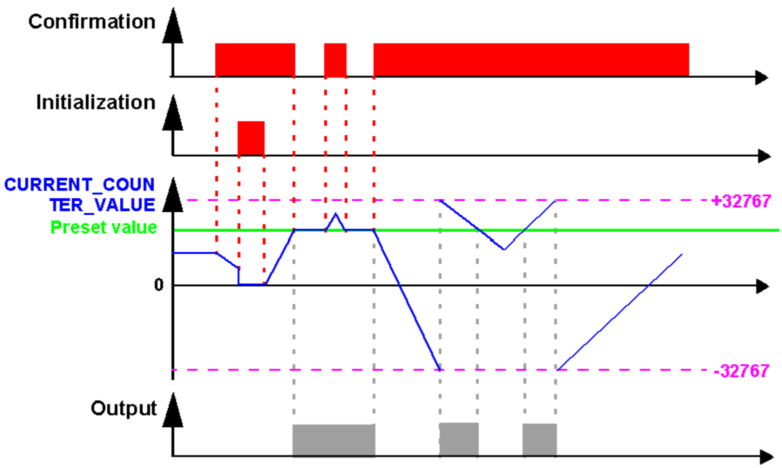
Parameter lock

Locking prevents modification of the parameters of the locked function block from the front panel of the logical module using the PARAMETERS menu.

Up-Counting in Single Cycle Mode

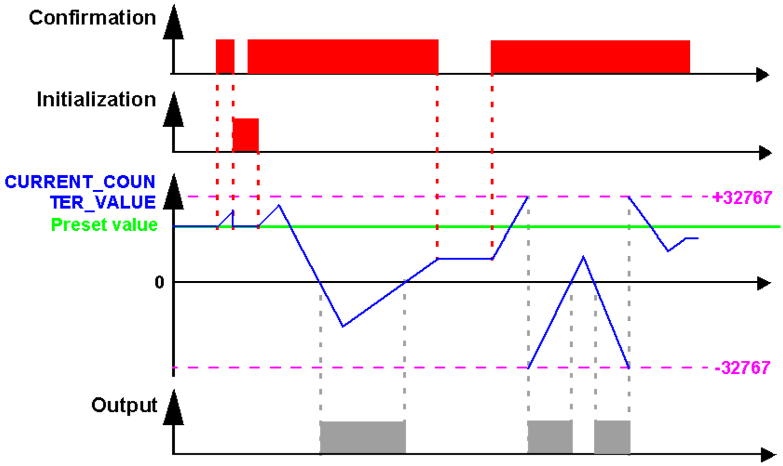
For the following 4 charts, the blue curve represents the counter value, when it increases there are pulses on I1 and when it decreases, pulses on I2.

The following diagram shows the operation of the counter with initialization at 0:



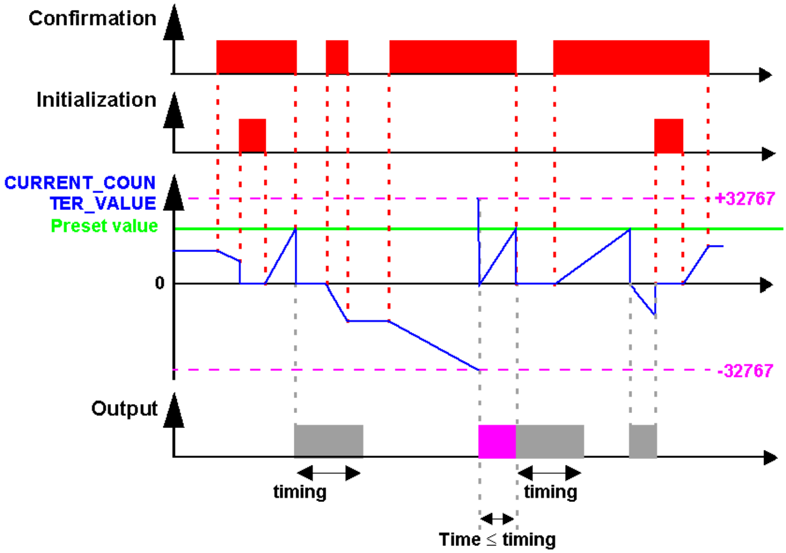
Down-Counting in Single Cycle Mode

The following diagram shows the operation of the down-counter with initialization at the preset value:



Up-counting in Repetitive Cycle Mode

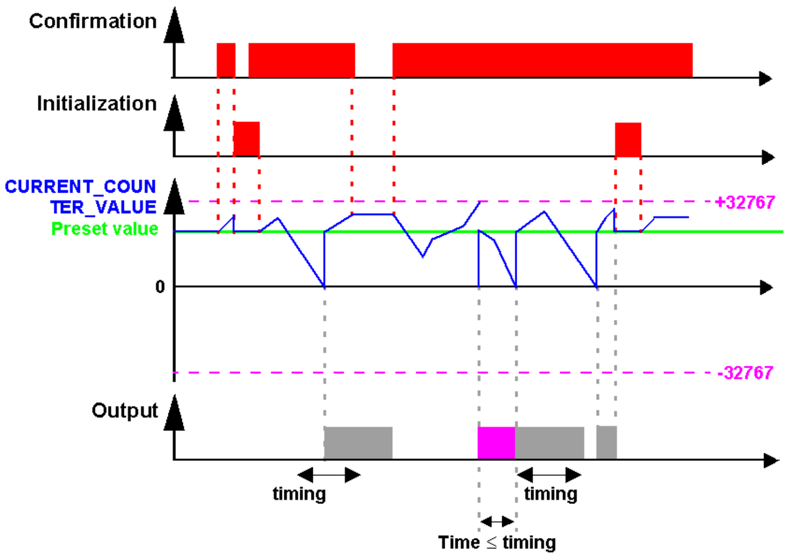
The following diagram shows the operation of the counter with forcing to 0 of the counter value on initialization, or when the counter value has reached the preset value:



The output switches to the **Inactive** state when the predefined pulse duration value has run out. If the switch condition is Active before the switch to the Inactive state, the output pulse is extended by the **Duration of pulse** (Timing).

Down-Counting Function in Repetitive Cycle Mode

The following diagram shows the operation of the down-counter with forcing to the preset value of the counter value on initialization, or when the counter value has reached 0:



The output switches to the **Inactive** state when the predefined pulse duration value has run out. If the switch condition is Active before the switch to the Inactive state, the output pulse is extended by the **Duration of pulse** (Timing).

UP/DOWN COUNT (Up/Down Counter)

Description

The **Up/Down Counter** function is used to up-count or down-count from a preset value resulting from a calculation outside the function.

A level 1 on the **Preset forcing** input is used to change the counter with the value available at the **Preset** input.

The **Preset** input can be connected to the NUM constant, to an analog input, or to any other kind of function block output which delivers an integer-type value.

A rising edge on the:

- **Upcounting**: increments the counter.
- **Downcounting**: decrements the counter.

State of the **Output**:

- **1**: When the counting number has been reached, the **Output** switches to 1 and remains so for as long as the counting number is greater than or equal to the **Preset** value,
- **0**: If the transitions on the **Downcounting** input switch the counting number back to a value less than the **Preset** value.

Activation of the **Reset** or **Preset forcing** inputs enables the counter to be relaunched.

When the **Reset** input is set to 1, the **Output** remains in state 0. When the **Reset** input becomes 0, the up/down counting operation is restarted from zero.

Access

This 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The **Up/Down Counter** uses the following inputs:

- Discrete-type **Upcounting**,
- Discrete-type **Downcounting**,
- Discrete-type **Reset**.
- Discrete-type **Preset forcing**.
- Integer-type **Preset**.

The up/down counter provides the following outputs:

- **Output** (Integer-type),
- **Current value**, integer type, between -32768...32767.

Parameters

The **Latching** parameter, if selected, enables the current value of the timer to be retrieved following a power outage *(see page 68)*.

TIMER A/C (Timer)

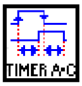
At a Glance

The **Timer** function is used to delay, prolong and control actions over a predetermined time.

The **Timer** has three functions:

- The A function *(see page 290)*: timer on-delay, or timer active,
- The C function *(see page 291)*: timer off-delay, or timer idle,
- The function A/C *(see page 291)*: combination of functions A and C.

Access

The  function is accessible from the **FBD** function bar.

Inputs/Outputs

The **Timer** features:

- A discrete **Command** input,
- A discrete **Reset** input.

The timer provides:

- A Discrete-type output,
- A copy of the activation delay setpoint (1),
- A copy of the deactivation delay setpoint (1),
- The current value of the activation delay (1),
- The current value of the deactivation delay (1).
  (1) these integer values are displayed in Simulation and Monitoring mode.

Parameters

In the programming software

From the **Parameters** window, you can adjust the value of the delays for each of the functions (A, C and A/C).

- **ON delay** for function A,
- **OFF delay** for function C,
- A combination of both the ON and OFF delays can be used to adjust function A/C.

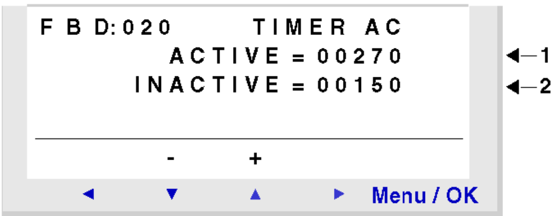
In the event of a power outage *(see page 68)*, the **Latching** parameter, if selected, enables the timer to be restarted at the point where it stopped.

From the front panel

From the **PARAMETERS** *(see page 91)* menu, you can adjust:

- The duration of the pulse **Active state time** (**1**),
- The duration between two pulses **Inactive state time** (**2**),

Illustration:

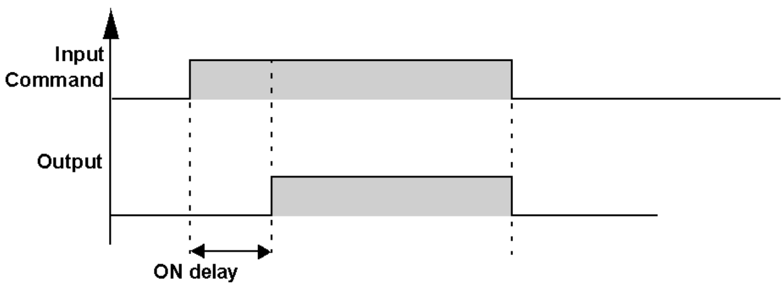


Parameter lock

Locking prevents modification of the parameters of the locked function block from the front panel of the logical module using the PARAMETERS menu.

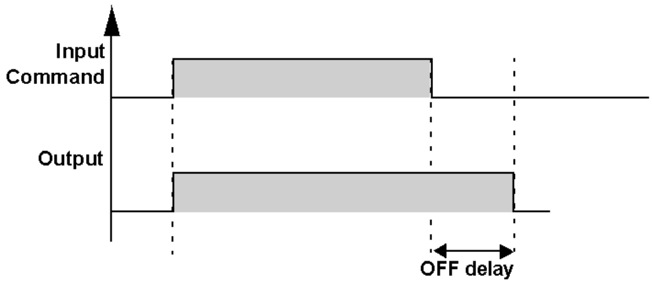
A Function

The following diagram shows the operation of the timer in function A:



C Function

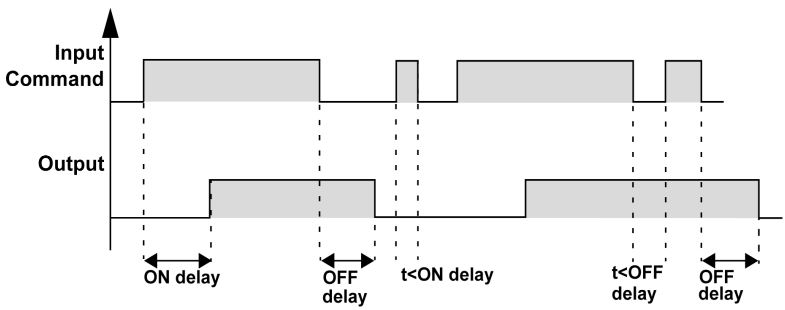
The following diagram shows the operation of the timer in function C:



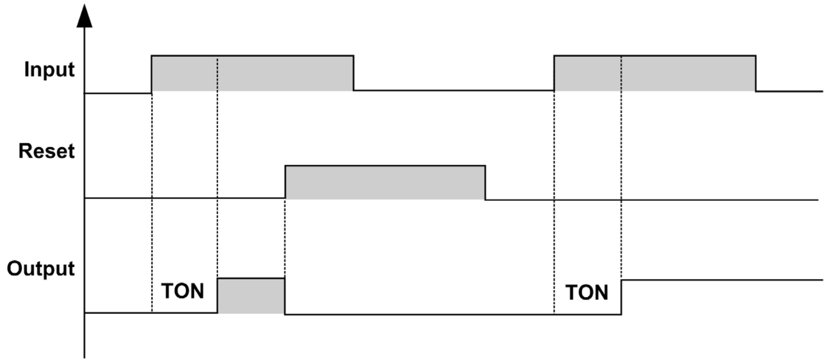
NOTE: Each pulse on the **Command** input of the Timer block resets the current value to 0.

A/C Function

The following diagram shows the operation of the timer in function A/C:



The following diagram illustrates the operation of the timer when the **Reset** input is activated, with the timer in function A/C:



TIMER BW (Pulses on Edges)

At a Glance

The **Pulses on Edges** function is used to create a pulse on an input edge for the duration of a cycle on the output.

The types of input edges taken into account can be the following:

- Rising edge,
- Falling edge,
- Rising and falling edge.

The input and output of the function are of the Discrete type.

Access

This 

function is accessible from the **FBD** function bar.

Parameters

From the **Parameters** window, you can select the type of edge to be processed on the input:

- Inactive to Active for a rising edge,
- Active to Inactive for a falling edge,
- Inactive to Active and Active to Inactive for a rising and falling edge.

TIMER Li (Cyclic Timing)

Description

The **Cyclic timing** function generates pulses (flashes) on the input rising edge.

The duration of the pulse and the duration between each pulse can be set.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The function uses a Discrete **Command** input.

The function provides:

- A Discrete-type **Output**,
- A copy of the setpoint for the pulse duration (1),
- The current value of the duration of the output active state (1),
- A copy of the setpoint for the duration between two pulses (1),
- The current value of the duration for which the output is in the inactive state (1),
- A copy of the setpoint (1):
  - For the number of flashes,
  - Or for the duration of the flash.
- The current value (1):
  - Of the number of flashes since the first pulse,
  - Or of the duration of flashes since the first pulse.

  (1) these integer values are displayed in Simulation and Monitoring mode.

If the **Command** input is inactive, the **Output** is inactive and the current values are set to 0.

Parameters

In the programming software

From the **Parameters** window, you can adjust:

- **On time**: value between 0...32767,
- **Off time**: value between 0...32767,
- **Number of flashes**: value between 0...32767,
- **Duration of flashes**: value between 0...32767,
- The selection for **Continuous** flashing.

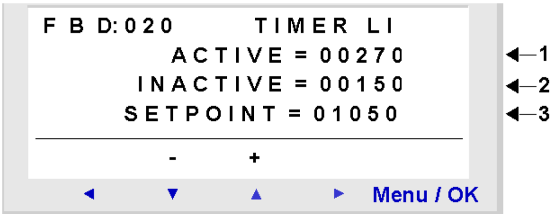
The **Latching** parameter, if selected, enables processing to be restarted at the point where it stopped following a power outage *(see page 68)*.

From the front panel

From the **PARAMETERS** menu *(see page 91)*, you can adjust:

- The duration of the pulse **Active state time** (**1**).
- The duration between two pulses **Inactive state time** (**2**).
- The counting setpoint corresponds to either a duration or a number of pulses (**3**).

Illustration:



Parameter lock

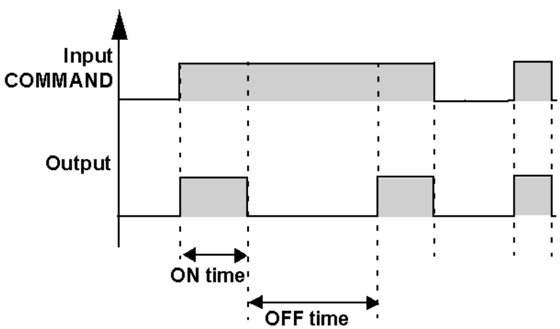
Locking prevents modification of the parameters of the locked function block from the front panel of the logical smart relay using the **PARAMETERS** menu.

Latching

To ensure latching after a power outage in the smart relays, check the **Latching** box in the parameters window.

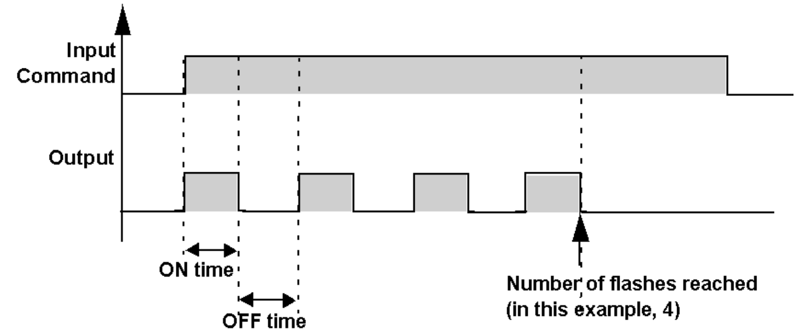
Continuous flashing

The figure below illustrates function operation with continuous flashing:



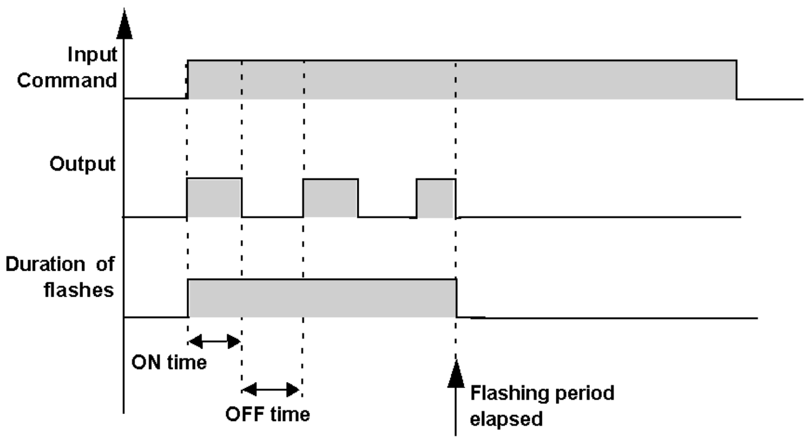
Number of Flashes

The figure below illustrates function operation with a defined number of flashes:



Duration of flashes

The figure below illustrates function operation with predefined flash duration:



TIMER B/H (Time out)

Description

The **Timer B/H** function creates a pulse on the output of the rising edge of the input.

Processing of the **Command** input depends on two types of functions:

- **Function B**: regardless of the duration of the command pulse, the output is active for a duration that has been set,
- **Function H**: the output is inactive at the end of a set time or on the falling edge of the command.

Activation of the **Reset** input allows the output to be deactivated.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The function uses:

- A discrete **Command** input,
- A discrete **Reset** input; this input is inactive if it is not connected.

The function provides:

- A Discrete-type **Output**,
- A copy of the setpoint for the pulse duration (1),
- The current value of the pulse (1).
  (1) these integer values are displayed in **Simulation** and **Monitoring** mode.

Parameters

In the programming software

From the **Parameters** window, you can adjust:

- The duration of the **On time** pulse, which is a value between 0 and 32767 (x 100ms),
- The selection of **Function B** or **Function H**.

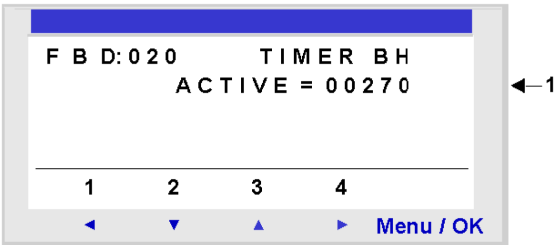
In the case of **Function H**, the **Latching** parameter, if selected, allows processing to be restarted following a power outage *(see page 68)* at the point where it stopped.

From the front panel

From the **PARAMETERS** menu *(see page 91)*, you can adjust:

- The duration of the **Inactive state time** pulse (x 100ms), (**1**),

Illustration:



Parameter lock

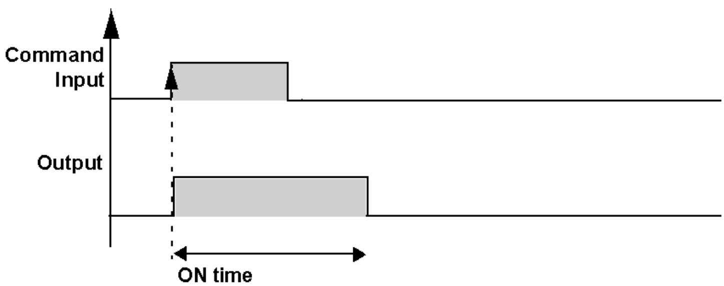
Locking prevents modification of the parameters of the locked function block from the front panel of the logical smart relay using the **PARAMETERS** menu.

Latching

To ensure latching after a power outage in the smart relays, you should check the **Latching** box in the parameters window.

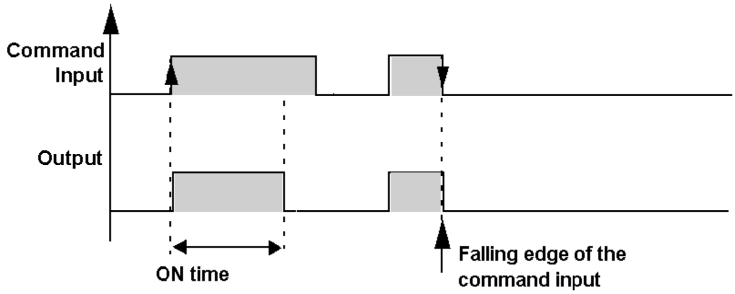
Function B

The following figure illustrates operation with **Function B** set up:



Function H

The following figure illustrates operation with **Function H** set up:



COMP IN ZONE Comparison

Description

The **Comparison** function is used to compare one value between two setpoints (the MIN and MAX values of the zone).

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The comparison function uses:

- A discrete **Enable** input; this input is Active if it is not connected,
- An integer-type **Value to compare** input,
- An integer-type **Min value** input,
- An integer-type **Max value** input,
- A discrete **Output**.

The OUTPUT indicates the result of the comparison when the ENABLE input is active.

The OUTPUT does not change state when the ENABLE input changes from Active to Inactive state.

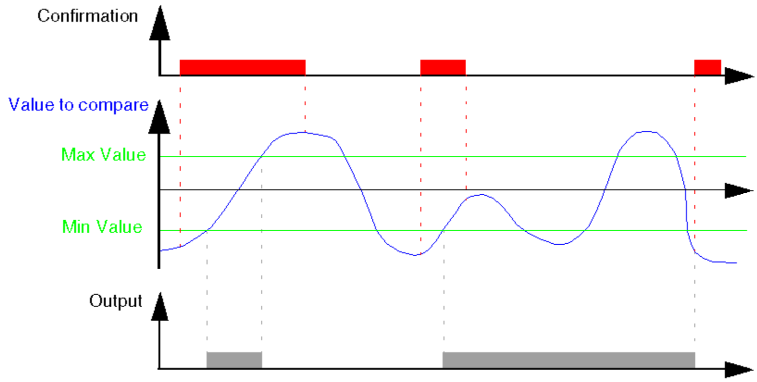
Parameters

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 setpoints (min and max),
- **OFF in the zone**: The output will be inactive if the input value is between the two setpoints (min and max).

Comparison Function

The diagram below shows the different states for the output, depending on the input value to be compared and the enable input:



PRESET H-METER (Preset Hour Counter)

Description

The **Preset hour counter** measures the duration of input activation. When this duration reaches a preset value, the output is activated.

The duration can be set in hours (Maxi 32767) and minutes.

Activation of the **Reset** input deactivates the output.

Access

This function 

is accessible from the **FBD** function bar.

Inputs/Outputs

The counter uses:

- A discrete **Command** input,
- A discrete **Reset** input.

If these two inputs are not connected, they are set respectively to Active and Inactive.

The counter provides:

- A Discrete-type **Output**,
- The copy of the setpoint of the number of hours (1),
- The current value of the number of hours (1),
- The copy of the setpoint of the number of minutes (1),
- The current value of the number of minutes (1),
  (1) these integer values are displayed in **Simulation** and **Monitoring** mode.

Parameters

From the **Parameters** window, you can adjust:

- The preset **Hour** value, which is a value between 0 and 32767,
- The preset **Minute** value, which is a value between 0 and 59.

The **Latching** parameter, if selected, enables the timer to be restarted at the point where it stopped following a power outage *(see page 68)*.

TRIGGER (Schmitt Trigger)

Description

The **Schmitt Trigger** function allows an analog value to be monitored relative to two thresholds.

The output changes state if:

- The input value is less than the minimum value,
- The input value is greater than the maximum value.

If the input is between the two, the output does not change state.

Each of the **From on to off** and **From off to on** setpoints can be set as the minimum or maximum value. This involves reverse operation of the function. These two operations are shown in the diagrams *(see page 304)*.

If the **Enable function** input is in inactive state, the output remains inactive. The output does not change state if the **Enable function** input changes from Active to Inactive state.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

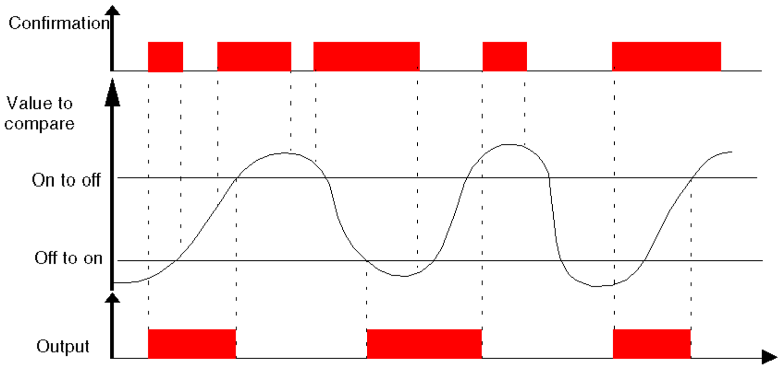
The function uses four inputs:

- An Integer-type **Value to compare** input,
- An Integer-type **On to off setpoint** input,
- An Integer-type **Off to on setpoint** input,
- A Discrete-type **Enable function** input.

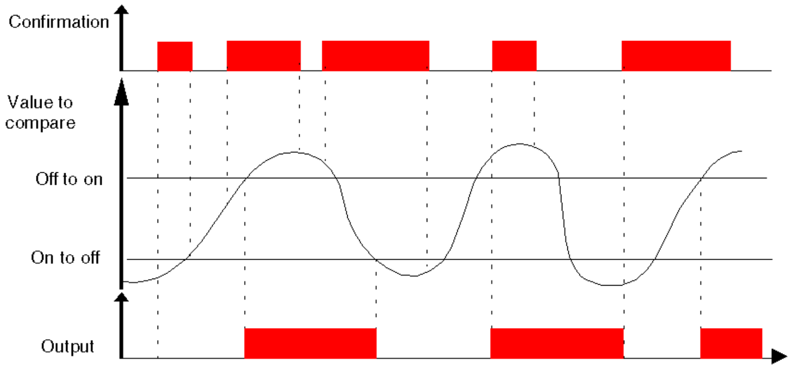
The function provides a discrete-type **Output**.

Operating Diagrams

The figure shows possible output states when the **On to off setpoint** is higher than the **Off to on setpoint**:



The figure shows the possible output states when the **Off to on setpoint** is higher than the **On to off setpoint**:



COMP IN ZONE (Comparison of two values)

Description

The **Comparison of two values** function is used to compare two analog values.

The output is active if the result of the comparison between **Value 1** and **Value 2** is true **and** if the **Enable function** input is active or not connected.

The output does not change state if the **Enable function** input changes from Active to Inactive state.

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.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The function uses:

- A Discrete-type **Enable function** input.
- An Integer-type **Value 1** input,
- An Integer-type **Value 2** input.

If the **Value 1** or **Value 2** input is not connected, the value is set to 0.

The function provides a discrete-type **Output**.

GAIN Function

Description

The **Gain** function enables analog values to be converted by changing the scale and offset.

Gain calculation formula:

```
CALCULATION OUTPUT = A / B * CALCULATION INPUT + C
```

Access

This 

function is accessible from the **FBD** function bar.

Inputs/Outputs

Description of the inputs:

- **Enable function**: This is the gain function input command, whose type is Discrete *(see page 251)*.
  The state of this input determines operation of the block: If the **Enable Function** input is inactive, the **Calculation output** retains the last calculated value.

NOTE: If the **Enable Function** input is not connected, it is considered to be active.

- **Calculation Input**: value of the analog input connected to the gain function.
  This is an integer between -32768 and 32767.

Description of the output:

- **Calculation Output**: this is the output value of the gain function.
  This value depends upon the state of the **Enable function** input.
  If the **Enable function** input is:
  - Inactive: the **Calculation output** is equal to zero,
  - Active: the **Calculation output** is equal to the result of the gain calculation formula.

Parameters

In the Programming Software

From the **Parameters** window, you can adjust:

- **A/B** which corresponds to the **gain** applied by the function with:
  - **A**: being a numerator (from -32768 to 32767),
  - **B**: being a denominator (from -32768 to -1 and from 1 to 32767) (**4**)
- **C** is the **offset** applied by the function, and is an integer between -32768 and 32767(**5**).

In addition, it is possible to define an **operating range** by setting limits for the function output:

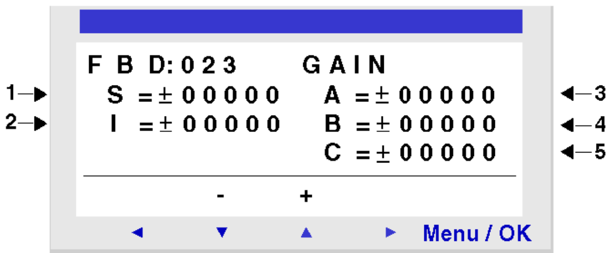
- Lower limit: integers between -32768 and 32767,
- Upper limit: integers between -32768 and 32767.

From the front panel

From the **PARAMETERS** *(see page 91)* menu, you can adjust:

- **S**: Upper limit (**1**),
- **I**: Lower limit (**2**),
- **A**: numerator (**3**),
- **B**: denominator (**4**),
- **C**: offset (**5**).

Illustration:



Parameter Lock

Locking prevents modification of the parameters of the locked function block from the front panel of the logical module using the PARAMETERS menu.

DISPLAY (LCD Screen Display)

Description

The **DISPLAY** function is used to display text, a date, a time, or numerical values on the LCD display, instead of the inputs-outputs states.

Different types of content can be displayed simultaneously, by using up to 32 **DISPLAY** function blocks.

Each function block defines the content to be displayed in a specific position on the LCD display. The content can be either:
- Text (maximum 72 characters),
- Numerical values corresponding to the output of a function used in the application (for example a counter). These values can include a decimal point.
- Date, time or calibration value from the **Clocks** function.

It is possible to authorize the modification of the content using the keys on the front panel.

To switch the display from the **DISPLAY** screen to the INPUTS-OUTPUTS screen, hold down the **Shift** key, then press the **Menu/OK** key.

NOTE: ASCII-standard characters, together with accented characters, can be used.

NOTE: Characters and symbols that are not displayed in the data entry window when keyed are not supported.

Access

The  function is accessible from the **FBD** function bar.

Inputs

- **Enable function**: this is the Discrete *(see page 251)*-type **DISPLAY** function input command. The state of this input determines operation of the block: If the **Enable** function input is active, the information is displayed on the LCD; otherwise, there is no display.

NOTE: If the **Enable** Function input is not connected, it is considered to be active.

- **Value input**: this is the selection input that determines the nature of the information to be displayed, if this input is:
  - Not connected: the display corresponds to the selection made in the **User options** zone.
  - Connected to the output of a function block: the display corresponds to the value emitted by this output in the **Display mode** zone.

Parameters

From the Parameters window, you can adjust:

Value input not connected

The display corresponds to the selection made in the **User options** zone.

Depending on the options chosen, the following can be selected:

- **Text**: a string of characters,
- **Date**: The value of the date of the device on which the program is executed (smart relay or simulator),
- **Time**: The time value,
- **Calibration**: The drift value of the smart relay clock.

Value input connected

The value emitted by the function block output is displayed according to the format indicated in the parameters window.

The integer value present on the input is converted into a string of characters, whose format depends on the option that has been selected:

- **Integer 1/1 - 1/10000**:
  - 1/1 signed integer,
  - 1/10 - 1/10000 signed decimal number; the fractional part represents the number of digits after the decimal separator.
- **Authorized modification**
  Modifications are made using the buttons on the front panel of the smart relay or in the window on the front panel of the Zelio Soft 2 Programming software.
  Enabling this option allows the following to be modified:
  - The integer data connected on the **Value input** of the function if the function can be modified by **DISPLAY**,
  - The smart relay date or time value (execution on a smart relay),
  - The value of the simulator date and time (in simulation mode),
  - The correction of the drift of the smart relay clock (execution on a smart relay). The last action is inoperative in simulation mode.

Description of the modification procedure for displayed values:

Step	Action
1	Press the **Shift** key (white) until **Param** is displayed at the bottom of the screen, keep it pressed and press the ► key once until **Prog** is displayed at the bottom of the screen and then release the **Shift** key. **Result**: The value of parameter flashes.
2	Press the ▼ ▲ keys to modify this value.
3	Press the **Menu/Ok** to validate.

Operating mode

Description of the interface

Each display function is identified by a block number: `BXX`.

This identifier is found:

- On the wiring sheet: The number is located at the top right corner of the block,
- In the parameters window / **Parameters** tab: the number is in the drop-down menu in the top left corner of the window.

The parameters window displays the resulting string from the blocks (`BXX`) used in the wiring sheet.

The parameters window opens by default on the function block number from which the dialog box is open.

For the selected block, text concerning it appears in red.

In the event of overlap, the text appears in red reverse video mode.

The non-overlapping text corresponding to the other selected blocks appear in black.

Entering one of the parameters of a DISPLAY block

Description of the entry procedure:

Step	Description
1	Position the start of the text using the parameters: • **Row**: value between 1 and 4 • **Column**: value between 1 and 18 (The position is stated in relation to the top left box)
2	Select the type of information to be displayed (text, date, value, etc.).
3	Confirm by clicking **OK**. **Result**: The new **DISPLAY** block is saved and the parameters window is closed.

NOTE: If more than 32 blocks are enabled simultaneously, only the first 32 will be displayed.

NOTE: If the strings are superposed, an error is displayed in the grid: The boxes appear in red, valid strings are displayed in black.

TEXT

Description

The **TEXT** function is used to display text, a date, a time, and numerical values on the LCD display, instead of the inputs-outputs states.

One single **TEXT** function block is used to define the content of the entire LCD display.The content can be a combination of:

- Text (maximum 72 characters),
- Numerical values corresponding to the output of a function used in the application (for example a counter).
- Date, time or calibration value from the **Clocks** function.

It is possible to authorize the modification of the content using the keys on the front panel.

The maximum number of variables that can be displayed per **TEXT** block is 4.

Multiple **TEXT** functions can be included in an application and activated using the **SET**/**RESET** validation input. If multiple **TEXT** blocks are activated simultaneously, the block with the highest number is displayed.

To switch the display from the **TEXT** screen to the INPUTS-OUTPUTS screen, hold down the **Shift** key, then press the **Menu/OK** key.

Access

The 

function is accessible from the **FBD** function bar.

Inputs

The **Text** function has two discrete inputs:

- **Set**: Activating the Set input triggers the display.
- **Reset**: Activating the **Reset** input cancels the display. Reset takes priority over Set.

The **Text** function has four 10-bit analog inputs which are values that can be displayed.

- **Value 1**
- **Value 2**
- **Value 3**
- **Value 4**

Character String Display

The cursor is positioned at the start of the string displayed in the window:

- By clicking on the box (which then flashes)
- By using the arrow keys on the computer keyboard

Description of the entry procedure:

Step	Action
1	Position the cursor at the start of the text.
2	Type the text to be displayed using the keyboard.
3	Confirm by clicking **OK**. **Result:** The new **Text** block is saved and the parameters window is closed.

NOTE: The character string is limited to the four lines. Any additional character overwrites the one in the last box.

NOTE: ASCII-standard characters, together with accented characters, can be used. Characters and symbols that are not displayed in the data entry window when keyed are not supported.

NOTE: If the text entered on a line covers an existing numerical value, the latter is deleted.
If a numerical value is positioned over text that has already been entered, the characters it covers are overwritten.

Displaying a Numerical Value

Positioning:

To position the value on the line, drag and drop it to the edit window.

Selection:

The value to be displayed is selected in the window located above the edit window.

This window lists the following elements:

- **Date**: The value of the date (day.month.year) of the device on which the program is executed (smart relay or simulator)
- **Time**: The time value (hours:minutes)
- **Calibration** *(see page 117)*: The drift value of the smart relay clock
- List of values that can be displayed, i.e. the analog inputs of the function
- **Authorized modification**
  Modifications are made using the buttons on the front panel of the smart relay or in the window on the front panel of the Zelio Soft 2 Programming software.
  Enabling this option allows the following to be modified:
  - The integer data connected on the **Value** inputs of the function if the function can be modified by **TEXT**,
  - The smart relay date or time value (execution on a smart relay),
  - The current value of the simulator date and time (in simulation mode),
  - The correction of the drift of the smart relay clock (execution on a smart relay). The last action is inoperative in simulation mode.

Description of the modification procedure for displayed values:

Step	Description
1	Press the **Shift** key (white) until **Param** is displayed at the bottom of the screen, keep it pressed and press the ► key until **Prog** is displayed at the bottom of the screen and then release the **Shift** key. **Result**: The value of parameter flashes.
2	Press the ▼ ▲ keys to modify this value.
3	Press the ► ◄ keys in order to choose another parameter.
4	When all the parameters are at the desired values, press the **Menu/Ok** to validate.

Clear Text

Description of the procedure:

Step	Description
1	Select the zone to be cleared.
2	Clear using the **Delete** key on the keyboard.

TIME PROG (Daily, weekly, yearly programmer)

Description

The **Daily, weekly, yearly programmer** validates the time ranges when actions can be executed.

This function allows a maximum of 51 events to be defined, which are used to control its output.

Access

The `TIME PROG` function 

is accessible from the **FBD** function bar.

Outputs

Output: this is the programmer enable output.

When one of the cycles that has been defined as a parameter is reached, the output is active (the output remains active for the duration of this cycle).

Parameters

In the Programming Software

A cycle is defined by:

- The **type of action**: ON or OFF.
- The **time** at which it will take effect: Hour/Minute.
- The activation **mode**.

Cycles can be activated in different ways:

- **Annual**: Triggering of an event once a year.
  In this case, the month and day must be configured.
- **Monthly**: Triggering of an event once a month.
  In this case only the day must be configured.
- **Date**: Triggering of a single event on a specific date.
  In this case, the day, month and year must be configured.

NOTE: For the 3 previous types, the calendar can be used to configure the date (click on the calendar icon to open the calendar).

- **Periodically**: Triggering of an event on certain weeks of each month (weekly) or certain days of the week (daily).
  In this case, you will have access to a new series of options:
  - Weekly: This option is enabled by default and all weeks are selected, with the possibility of deselecting only certain weeks.
  - Daily: This option is enabled by default and all days are selected, with the possibility of deselecting only certain days (where the Daily option is no longer valid).

NOTE: The weeks indicated in Weekly option do not correspond to calendar weeks (Monday to Sunday), but are instead defined in relation to the number of days since the beginning of the month (the first seven days of the month form the first week).

From the front panel

From the **PARAMETER** *(see page 91)* menu, it is not possible to:

- Add or delete an event.
- Modify the type (**Periodically**, **Annual**, **Monthly**, and **Date**).
- Modify the ON/OFF activation type.

From the front panel, only the value of the parameters can be modified.

Illustration: **Date** activation mode:

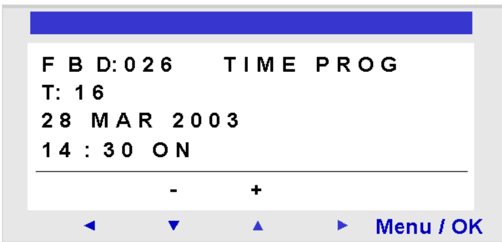
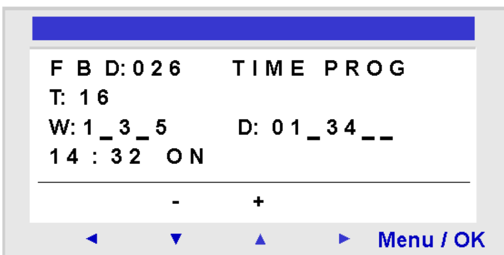


Illustration: **Periodically** activation mode:



Parameter Locking

Locking prevents modification of the parameters of the locked function block from the front panel of the smart relay using the **PARAMETERS** menu.

Creating a Cycle

Procedure for creating a new cycle:

Step	Action
1	Create a new cycle by clicking the **New** button in the **Parameters** tab. **Result:** A new event number appears in the **Current cycle** box.
2	Configure the time when the event should take place: Hour/Minute.
3	Configure the type of action: ON or OFF.
4	Configure the activation mode according to your criteria (by default, the cycle will be triggered every hour at the time indicated).
5	Confirm by clicking **OK**. **Result:** The new cycle is saved and the parameters window is closed.

Modifying a Cycle

Procedure for modifying a cycle:

Step	Action
1	Select the cycle to be modified using the **Current cycle** drop-down menu in the **Parameters** tab. **Result:** The configuration of the selected cycle is opened.
2	Modify the required parameters.
3	Confirm by clicking **OK**. **Result:** The new cycle is saved and the parameters window is closed.

Clearing a Cycle

Procedure for clearing a cycle:

Step	Action
1	Select the cycle to be cleared using the **Current cycle** drop-down menu in the **Parameters** tab. **Result:** The configuration of the selected cycle is opened.
2	Clear the cycle using the **Clear** button. **Result:** The cycle disappears from the drop-down menu.
3	Confirm by clicking **OK**.

Summary of the Configuration

To view the cycles created and the conditions that trigger them, select the **Summary** tab and scan the list of the cycles set up.

The **Clear** button allows you to delete the designated cycle by clicking in the **Summary** tab list.

The **Number** button allows you to assign a new number (not yet used) to a designated event by clicking in the **Summary** tab list.

To modify the characteristics of a cycle, double-click on the desired line. The parameters window opens on the selected cycle.

Simulation and Monitoring Modes

Clock Configuration

In simulation mode, it is the clock specific to the simulator that is taken into account. During the switch to simulation mode this clock is initialized with the time/date of the PC on which the programming software is running.

The clock parameters can then be modified:

- Using the **module** menu Read/Write date and time command.
- Using the CLOCK command in the MISCELLANEOUS option, which can be accessed using the buttons on the front panel.
- Using the accelerator *(see page 432)*.

Modifying the TIME PROG Parameters

These parameters cannot be modified by opening the parameters window in the Simulations and Monitoring modes.

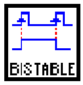
These parameters can be modified from the front panel (using the buttons) in the **PARAMETERS** menu, followed by selecting the number of the block corresponding to `TIME PROG` and the event number to be modified.

BISTABLE (Impulse Relay)

Description

The **Impulse Relay** function switches the **Output** state on each rising edge (change from inactive to active) of the **Command** input.

Access

The impulse relay function 

is accessible from the **FBD** function bar.

Inputs/Outputs

Description of the inputs:

- **Command**: this is the input that controls changes in the output state, whose type is Discrete *(see page 251)*.
- **Reset**: when this command is active, the OUTPUT always remains inactive, regardless of the COMMAND input transitions.

NOTE: If the **Reset** input is not connected, it is considered to be inactive.

Description of the output:

- **Output**: this is the impulse relay output, whose type is Discrete *(see page 262)*.
  This value depends upon the state of the **Reset** input.
  If the **Reset** input is:
  - Inactive: the **Output** changes state in line with the transitions of the **Command** input,
  - Active: the **Output** always remains inactive.

MUX (Multiplexing)

Description

The **Multiplexing** function carries out two input channel multiplexing on the **Output**.

Access

The  function is accessible from the **FBD** function bar.

Inputs/Outputs

Description of the inputs:

- **Channel A**: this is the multiplexer input A, whose type is integer *(see page 254)*.
- **Channel B**: this is the multiplexer input B, whose type is integer *(see page 254)*.
- **Command**: this input is used to choose the input channel to apply to the output.

NOTE: If the **Command** input is not connected, it is considered to be inactive.

NOTE: If channels A or B are not connected, they are set to 0.

Description of the output:

- **Output**: this is the multiplexer output.
  This value depends upon the state of the **Command** input.
  If the **Command** input is:
  - Inactive: the **Output** corresponds to **Channel A**,
  - Active: the **Output** corresponds to **Channel B**.

ADD/SUB (ADD/SUB Arithmetic Function)

Description

The **ADD/SUB** arithmetic function is used to perform simple operations on integers:

- Addition
- Subtraction

Calculation formula:

```
CALCULATION OUTPUT = INPUT1+INPUT2-INPUT3
```

Access

This  function is accessible from the **FBD** function bar.

Inputs/Outputs

Description of the inputs:

- **Input 1**: First input value of the formula (integer *(see page 254)*).
- **Input 2**: Second input value of the formula (integer *(see page 254)*).
- **Input 3**: Third input value of the formula (integer *(see page 254)*).

NOTE: If the inputs are not connected, they are set to 0.

- **Error propagation**: This Discrete *(see page 251)* type input is used to propagate errors (or saturations) from calculation functions (ADD/SUB or MUL/DIV) carried out upstream.

NOTE: If **Error propagation** is set to 1, then the operations are not performed and the **Error/Overrun** output is set to 1.

NOTE: If the **Error propagation** input is not connected, it is set to 0.

Description of the outputs:

- **Calculation Output**: This is the value of the calculation formula output (integer *(see page 262)*).
- **Error/Overrun**: This output, whose type is discrete *(see page 262)*, indicates any presence of saturation errors).
  This output is activated in the following cases.
  - The consequence of the operations is a result that is not included in the interval [-32768 to +32767].
  - The **Error propagation** input is active.

Examples

Simple addition: do not use the **Input 3** input.

Simple subtraction: do not use one of the **Input 1** or **2** inputs.

MUL/DIV (MUL/DIV Arithmetic Function)

Description

The **MUL/DIV** arithmetic function is used to perform simple operations on integers:

- Multiplication
- Division

Calculation formula:

```
CALCULATION OUTPUT = INPUT1*INPUT2/INPUT3
```

Access

This  function is accessible from the **FBD** function bar.

Inputs/Outputs

Description of the inputs:

- **Input1**: First input value of the formula (integer *(see page 254)*).
- **Input2**: Second input value of the formula (integer *(see page 254)*).
- **Input3**: Third input value of the formula (integer *(see page 254)*).

NOTE: If the INPUTS are not connected, they are set to 1.

- **Error propagation**: This Discrete *(see page 251)* type input is used to propagate errors (or saturations) from calculation functions (ADD/SUB or MUL/DIV) carried out upstream.

NOTE: If **Error propagation** is set to 1, then the operations are not performed and the **Error/Overrun** output is set to 1.

NOTE: If the **Error propagation** input is not connected, it is set to 0.

Description of the outputs:

- **Calculation output**: this is the value of the calculation formula output (integer *(see page 262)*).
- **Error/Overrun**: This output, whose type is discrete *(see page 262)*, indicates any presence of saturation errors).
  This output is activated in the following cases:
  - The consequence of the operations is a result that is not included in the interval [-32768 to +32767].
  - The **Error propagation** input is active.
  - The **Input 3** equals 0.

Examples

Simple multiplication: do not use the **Input 3** input.

Simple division: do not use one of the **Input 1** or **2** inputs.

CAM BLOCK (Cam Programmer)

At a Glance

The cam programmer function **CAM BLOCK** controls 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, output state is adjustable.

Once the maximum value has been reached, the cam restarts from its initial position (output returns to 0).

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

Description of the inputs:

- **Forward**: This is the input that controls cam progress; it moves one step forward at each rising edge (change from inactive to active).
- **Reverse**: This is the input that controls backward cam movement; it moves one step backward at each rising edge (change from inactive to active).

NOTE: The **Forward** input takes priority over the **Reverse** input.

NOTE: If the **Forward** and the **Reverse** inputs are not connected, they are set to inactive.

- **Reset** (initialization): When this input is active, the cam is replaced to its initial position: The **Position** output is forced to 1.

NOTE: The **Reset** input takes priority over the **Forward** and **Reverse** inputs.

NOTE: If the **Reset** input is not connected, it is set to inactive.

Description of the outputs:

- **Output 1 to 8**: State corresponding to the current position of the shaft (representing the 8 wheels).
- **Position**: Cam position (1 to 50).

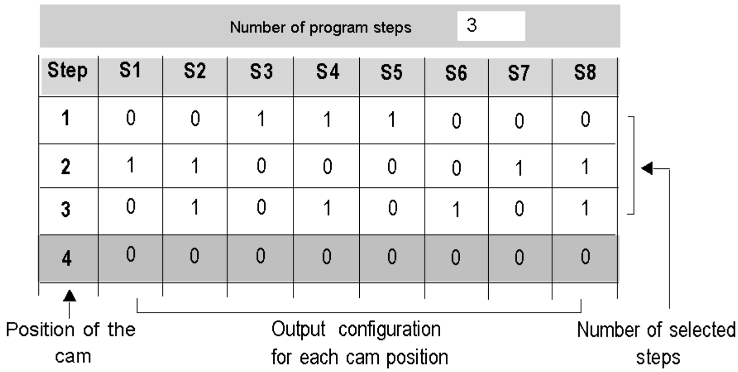
Parameters

From the programming software

From the **Parameters** window, you can adjust:

- **The number of program steps**: Its value is between 1 and 50.
- **Output status [1...8]**: For each position of the shaft.

The following figure illustrates part of the parameters window:



The `Latching` parameter, if selected, enables the current value of the timer to be retrieved following a power outage *(see page 68)*.

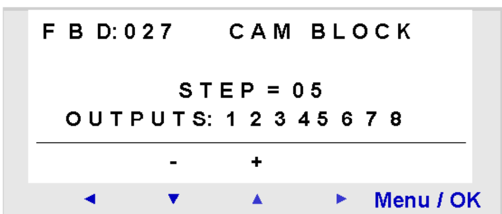
From the front panel

From the **PARAMETERS** *(see page 91)* window, the contents of the cam programmer's steps can be modified by consecutive bits, but it is not possible to modify the number of steps.

After you have entered the block number, then enter:

- **The step number**: Value between [1...50].
- **Output status [1...8]**: The value of each output can be set to inactive (normal display of the number) or active (number in reverse video).

Illustration:



Parameter lock

Locking prevents modification of the parameters of the locked function block from the front panel of the smart relay using the **PARAMETERS** menu.

ARCHIVE

At a Glance

The **Archive** data archiving function enables two values to be saved simultaneously with information relative to their dating.

Access

This  function can be accessed from the **FBD** function bar.

Inputs/Outputs

Input description:

- **Memorisation**: For each archive function command input (Discrete *(see page 251)* type) on each rising edge (transition from inactive to active), the VALUE input is memorized.

NOTE: If the **Memorisation** input is not connected, it is set to inactive.

- **Reset**: When the input (Discrete *(see page 251)* type) is active, it forces the **Valid archive** to inactive. The values that were previously saved remain available.

NOTE: If the **Reset** input is not connected, it is set to inactive.

- **Value 1**: This is the first input that is saved. The value present on this input is saved with information relative to its dating: time and date (this information is available on the outputs).
- **Value 2**: Second input saved.

NOTE: If a **Value 1 or 2** input is not connected, it is set to inactive.

Description of the outputs:

- **Valid archive** output (Discrete *(see page 262)* type) indicates the validity of the storage in process:
  - Inactive: No data available
  - Active: Data available

NOTE: The following outputs are integers.

- **Minute**: Value of the minute of the dating information (0 to 59).
- **Hour**: Value of the hour (0 to 23).
- **Day**: Value of the day (1 to 31).
- **Month**: Value of the month (1 to 12).
- **Year**: Value of the year (0 to 99).
- **ARCHIVE 1**: Whole value present on the **Value 1** input..
- **Archive 2**: Whole value present on the **Value 2** input.

Parameters

From the programming software

The `Latching` parameter, if selected, enables the current value of the timer to be retrieved following a power outage *(see page 68)*.

Storage Mechanism

If the **Memorisation** input is activated several times, only the data concerning the last activation is memorized.

Display of Saved Values

Saved values can be displayed; in order to do this, connect the outputs of the `Archive` function to the DISPLAY blocks.

The DISPLAY function can modify the value displayed if the `Authorized modification` parameter is checked.

NOTE: Any modification may result in inconsistencies of the archived data: Value/Date.

STATUS

Description

The **STATUS** function allows the user to access smart relay statuses and modify the behavior of its FBD and/or SFC program according to these statuses.

Only an alarm status is available (the advisory can be retrieved by the application), as the error causes the application to stop; thus the function block `STATUS` is no longer executed.

Access

The  function is accessible from the **FBD** function bar.

Inputs/Outputs

This function block does not have an input.

The function uses seven outputs:

- **Alarm status**: Active as soon as an error or an alarm is detected in the smart relay. In this case, the corresponding code is available on the **Alarm Number** output. The only way to return this output to inactive status and set the **Alarm Number** to zero is to use the front panel **DEFAULT** menu with the `CLEAR` and `YES` commands. Usage: allows the program to be put into a known “fallback” state in case of error.
- **Run monitoring**: Active when the program is correctly executed on the smart relay and a Monitoring session is activated from the programming software.
  Otherwise, this output is inactive.
  Usage: In this operating mode, the watchdog action in the configuration is systematically overridden regardless of the programmer's initial choice. If in the program, the watchdog action (error / advisory) is essential, this output allows the program to be put into a known state with no changes for the controlled outputs.
- **Run parameters**: Emits a pulse when the program is correctly executed on the smart relay and a parameter modification action is activated either from the programming software, or after execution in the **PARAMETERS** menu on the front panel of the LCD.
  Otherwise, this output is inactive.
  Usage: in this operating mode, the watchdog action in the configuration is systematically overridden regardless of the programmer's initial choice. If in the program, the watchdog action (error/advisory) is essential, this output allows the program to be put into a known state with no changes for the controlled outputs.
- **Cold init**: Is in high level (with no possibility for edge detection) during the first execution cycle of a program when it switches from **STOP** to **RUN** with **INIT**.
  Usage: This output allows you to insert specific initializations in your program, for example, initializing the SFC `RESET-INIT` function, which confers latching in the SFC chart containing it in case of a power outage.

- **Warm init**: Is in high level (with no possibility for edge detection) during the first execution cycle of a program when power is restored following a power outage occurring when the program was in **RUN** mode or when it switches from **STOP** to **RUN** without **INIT**.
  Usage: This output lets you insert specific initializations in your program once the power has been restored.
- **Flash cycle**: Delivers a periodic signal that switches alternatively from ON to OFF at each execution of the program (**RUN** mode). Its period is equal to twice the duration of the cycle time described in the configuration.
- **Alarm number**: Provides the alarm code in signed integer format when the **Alarm status** output is active.

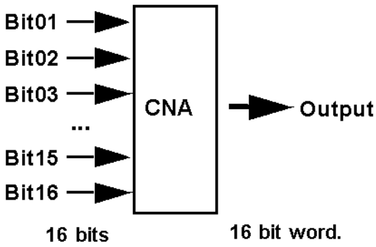
NOTE: In simulation mode, all outputs are significant.
However:
- **Run monitoring** is always active as it simulates functions similar to those of Monitoring.
- **Cold start** corresponds to the simulation of the switch from **STOP** to **RUN**.
- **Warm start** is triggered by the end of a power outage simulation.

CNA (Bit to Word Conversion)

Description

The **Bit to Word Conversion** function produces an integer (16 bits) type output from 16 bit-type inputs.

Illustration:



NOTE: This function can for instance be used to transfer discrete input or function status to a **Modbus** *(see page 267)* type output (O1XT1 ... O4XT1).

Access

The  function is accessible from the **FBD** function bar.

Inputs/Outputs

This function supports 16 discrete entries: **Bit01** (least significant byte) ... **Bit16** (most significant byte).

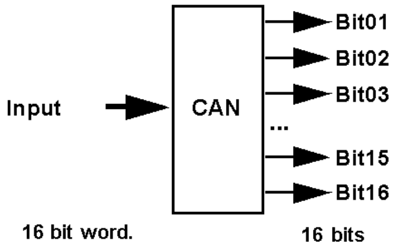
This function supports 1 integer type 16_bit output:

CAN (Word to Bit Conversion)

Description

The **Word to Bit Conversion** function breaks down an integer-type input (16 bits) into 16 bit type outputs. ..

Illustration:



NOTE: this function can be used for example to break down a **Modbus** *(see page 267)* type input (J1XT1 ... J4XT1) and to duplicate these statuses in the discrete outputs.

Access

The  function is accessible from the **FBD** function bar.

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).

SLIn (Serial Port Input)

Description

The **Serial port input** function block sends data via a serial link to fixed address memory locations in the smart relay.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The function delivers eight Integer-type outputs named input 1 to input 8. These outputs enable the application programmed in the smart relay to use the data stored in the selected fixed address memory locations.

Parameters

Select a range of eight addresses from the **Parameters** window. The other address ranges are available are as follows:

- 1 - 8
- 9 - 16
- 17 - 24

The Serial Link

The serial link is configured as follows:

- dialog speed: 115 kilobauds,
- format: 7 bits, even parity, 1 stop bit,

The Write Frame and the Response

Below is the write frame to send to the smart relay:

- Beginning delimiter: " : "
- Slave address: 0x01
- Write command: 0x10
- Data address: 0x00 00 FF xx
  xx is a number between 0x00 and 0x17 inclusively, corresponding to the address of the data to write less 1.
- Number of bytes: 0xnn
  This is the number of data to write. Each value is made up of two bytes.
- Data to write: 0xd1H d1L d2H ... dnnL
  There are the 0xnn bytes to write.

- Checksum: 0xcc
  This is the completed sum increased by 2, of the bytes between the slave address and the last of the date to write.
- End delimiter: " CR " " LF "

The smart relay response is structured as follows:

- Beginning delimiter: " : "
- Slave address: 0x01
- Write command: 0x10
- Data address: 0x00 00 FF xx
- Number of bytes: 0xnn
- Checksum: 0xcc
  This is the completed sum increased by 2, of the bytes between the slave address and the byte number.
- End delimiter: " CR " " LF "

For example

Write to the 3 address the 16-bit value 8569:

8569 corresponds to 0x2179 in hexadecimal format.

Checksum: 0x01+0x10 + 0x00 + 0x00+ 0xFF + 0x02 + 0x02 + 0x21 + 0x79 = 0x1AE of which complement increased by 2 gives for a 0x53 byte

" : " 0x01 0x10 0x00 0x00 0xFF 0x02 0x02 0x21 0x79 0x53 " CR " " LF "

The frame in the format above is used to calculate the checksum. Except for the delimiters, each byte is sent as two ASCII characters. Which gives:

0x3A 0x30 0x31 0x31 0x30 0x30 0x30 0x30 0x30 0x46 0x46 0x30 0x32 0x30 0x32 0x32 0x31 0x37 0x39 0x35 0x33 0x0D 0x0A

Smart relay response: 0x3A 0x30 0x31 0x31 0x30 0x30 0x30 0x30 0x30 0x46 0x46 0x30 0x32 0x30 0x32 0x45 0x44 0x0D 0x0A

Communication interruption

In the event of communication interruption, remove and then reapply power to the smart relay.

SLOut (Serial Port Output)

Description

The **Serial Port Output** function block is used to send data stored in fixed addresses in the smart relay to other equipment via a serial link.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The function provides eight integer-type inputs. These inputs enable the application to write the data that must be sent to the fixed address memory locations.

Parameters

Select a range of eight addresses from the **Parameters** window. The other address ranges are available are as follows:

- 25 - 32
- 33 - 40
- 41 - 48

The Read Frame and the Response

The read frame to be sent to the smart relay is as follows:

- Beginning delimiter: " : "
- Slave address: 0x01
- Read command: 0x03
- Data address: 0x00 00 FF xx
  xx is a number between 0x00 and 0x2F inclusively, corresponding to the address of the first data to read less 1.
- Number of bytes: 0xnn
  This is the number of data to read. Each value is made up of two bytes.
- Checksum: 0xcc
  This is the completed sum increased by 2, of the bytes between the slave address and the number of bytes.
- End delimiter: " CR " " LF "

The smart relay response is structured as follows:

- Beginning delimiter: " : "
- Slave address: 0x01
- Read command: 0x03
- Number of bytes: 0xnn
- Data read: 0xd1H d1L d2H ... dnnL
  These are the 0xnn bytes read.
- Checksum: 0xcc
  This is the completed sum increased by 2, of the bytes between the slave address and the last of the data to read.
- End delimiter: " CR " " LF "

Example

Read 5 16-bit data from address 17:

- Hexadecimal frame before ASCII coding:
  " : " 01 03 00 00 FF 10 0A E4 " CR " " LF "
- Hexadecimal frame after ASCII coding:
  3A 30 31 30 33 30 30 30 30 46 46 31 30 30 41 45 34 0D 0A
- The response will be as follows if the five values equal 0:
  3A 30 31 30 33 30 41 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 46 33 0D 0A

Communication interruption

In the event of communication interruption, remove and then reapply power to the smart relay.

COM (Message)

Description

When activated, the **Message** function block can be used to:

- Send alarm messages to mobile phones, the Zelio Logic Alarm software or Email addresses via the SR2COM01communication interface
- To provide access, remotely, to an I/O and/or a digital variable, to read or modify them.

It is possible to use up to 28 **Message** function blocks in the same program.

NOTE: The **Message** function is only available on smart relays with clocks and when a SR2COM01 communication interface is added to them *(see page 476)*.

Access

The 

function is accessible from the **FBD** function bar.

Inputs/Outputs

The **Message** function block has the **following inputs**:

- **Enable function**, depending on the configuration of the function block, the alarm message is sent when a transition is detected on this input:
  - From Inactive to Active State (by default),
  - From Active to Inactive State.
- **Val1**, I/O variable associated with this **Message** function block
- **Val2**, digital variable associated with this **Message** function block

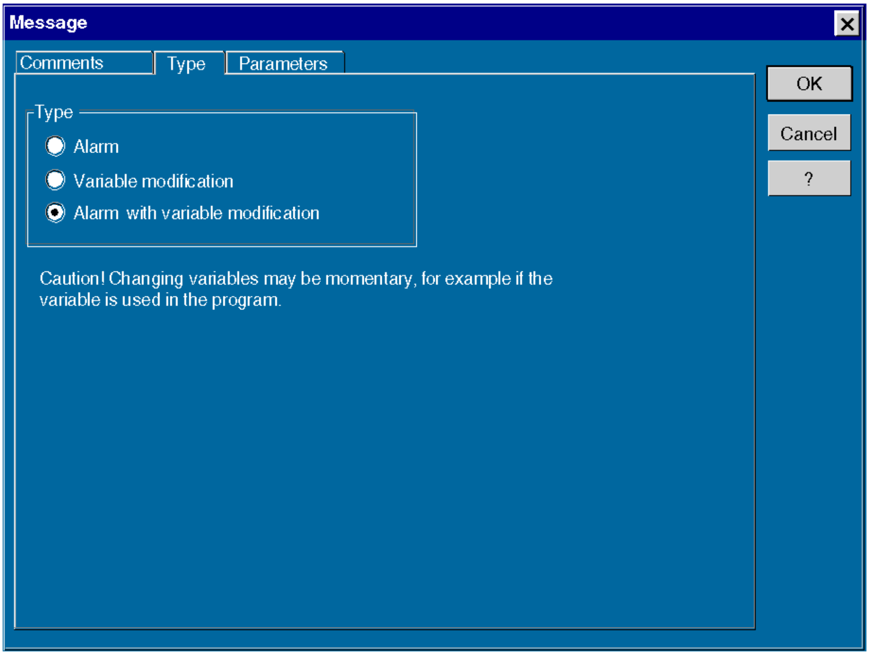
The values of the variables connected to the **Val1** and **Val2** inputs may (according to the configuration of the **Message** function block) be displayed in the alarm messages sent, and/or read and modified using the commands sent from a mobile phone or from the alarms operations tool.

The **Message** function block has an **output**. Each time the function block is enabled, a pulse is sent to this output.

Configuration from the Programming Software

Double click on the function block to make the parameters window appear. Use **Type** and **Parameters** tabs of this window to configure the block.

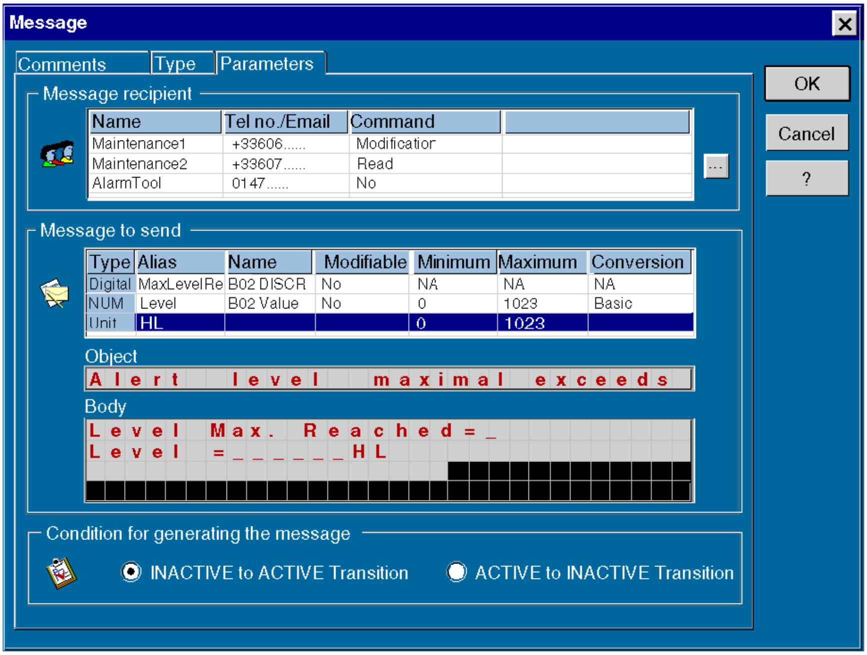
In the Type tab:



Select the **Type** of the alarm message:

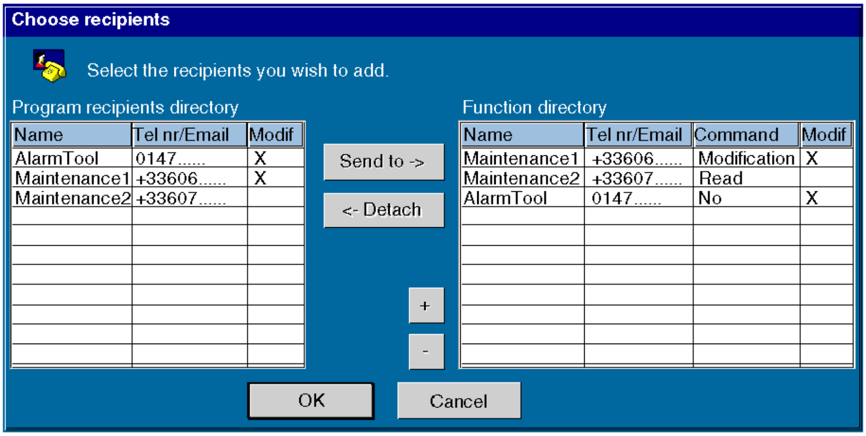
- **Alarm**: Is used to send an alarm message on activation of the function block. The variables associated with the block may be displayed in the message.
- **Variable modification**: Is used to provide access to an I/O and/or a digital variable. Each of these two variables may be declared as readable and modifiable.
- **Alarm with variable modification**: Used to send an alarm message on activation of the function block. The values of variables linked to the block may be displayed in the message and is used to provide access to an I/O variable and/or digital variable, each of these 2 variables may be declared as readable and modifiable.

In the **Parameters** tab:



Specify the alarm **Message recipients** of this block. These recipients are chosen from the Program recipients directory.

To do this, proceed as follows:

Step	Action
1	Click on the button of the **Message recipients** zone to add a recipient or modify the list of recipients of this message. **Result**: The following window appears:  These recipients are chosen from the Program recipients directory *(see page 473)*.
2	For each new recipient to be added, select it in the directory of the program and click on the **Send to →** button.
3	Organize the recipients in the order of priority *(see page 343)* by using the **+** and **-** buttons.
4	Double click on a recipient to specify the types of commands that it will be authorized to execute: ● Choose **No** to prevent a recipient from accessing variables connected to the **Message** function block, ● Choose **Read** to enable a recipient to execute read commands for variables connected to the **Message** function block, ● Choose **Modify** to allow recipients to execute read and modify variables connected to the **Message** function block: this choice is only available for recipients declared authorized to modify variables in the directory. **NOTE:** For more information on sending commands, see the on-line help of the Zelio Logic Alarm software. Confirm by clicking on the **OK** button.
5	Confirm by clicking on the **OK** button.

Configure the **associated variables** for the **Message** function block.

To do this, proceed as follows:

Step	Action
1	Double click on the line of the I/O variable, to access the **Configure** window. This window is used to: ● Modify the **alias** of the variable, ● Possibly make the variable readable and modifiable. Confirm by clicking on the **OK** button.
2	Double click on the line of the digital variable, to access the **Configure** window. This window is used to: ● Modify the **alias** of the variable, ● Possibly make the variable readable and modifiable, ● Define a range of values possible for the variable, if a modify command is sent, the new value ought to be found in this page, otherwise the command will not be processed, ● Possibly define the conversion properties *(see page 342)*. Confirm by clicking on the **OK** button.

Define the **Object and Body of Alarm Message** (these 2 fields are not available if the type chosen is **Variable modification**).

It is possible to include the value of one or both variables associated with the function block in the body of the message. To do so:
- Select the variable in the list,
- Drag and drop the variable into the zone of the message body.

NOTE: When using GSM modems to send an **Email by SMS**, the syntax used in the subject and body of the message is specific to each telephone operator. Contact the telephone operator of the SIM card of the modem and refer to the section Sending an Email via SMS *(see page 482)* for further information.

Define the **Condition for generating message** (this field is not available if the type chosen is **Variable modification**).

Choose:
- **Transition INACTIVE to ACTIVE**, so that the message will be sent when the **Enable function** input of the **Message** function block switches to active
- **Transition ACTIVE to INACTIVE**, so that the message will be sent when the **Enable function** input of the **Message** function block switches to inactive

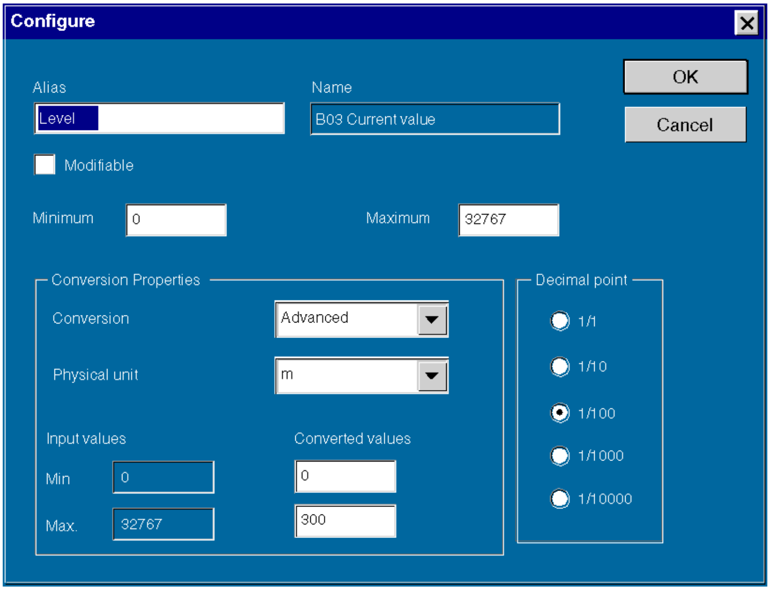
Conversion Properties

The conversion properties are used to facilitate reading the digital value linked to the **Message** function block. They are used, for example, to convert the digital value of the program to express it in a physical unit, and this converted value will be displayed in the message.

Example:

The internal variable **Level** (comprised between 0...32,767) represents the fill level of a tank comprised between 0...3 m.

We can thus use the following conversion parameters:



If the value of **Level** is 32,767, the communication interface converts it into meters. The value that will be sent in the alarm message will be 3.00 m.

If the recipient wants to modify this variable, he should send the Level=1.50 m command. The interface will convert the value and assign the converted value of 16 383 to **Level**.

NOTE: Additional information:
- The field **Physical unit** is used to add a unit of its choice behind the converted value
- The **Decimal point** field is used to move the decimal point in the converted value.
- The basic type of conversion is only available for analog input values and is used to convert the analog input voltage between 0...10 V directly in the desired amount.

Order of Priority

When sending an alarm message, the SR2COM01 communication interface contacts the message recipients one after the other. The **Choose Recipients** window is used to define the order in which the message recipients are contacted.

Depending on whether the **Recognition** option is activated or not, two types of recipients may be defined:

- **Recipient without recognition**: The alarm message is systematically sent, then the communication interface processes the next recipient,
- **Recipient with recognition** (for mobile-type recipients only): The communication interface sends the alarm message and waits for acknowledgment of the recipient via its mobile telephone:
  - If the recipient with recognition acknowledges the message, the communication interface continues the send sequence only to recipients **without** recognition,
  - If the recipient with recognition does not acknowledge the message in the given time (**Recognition delay**), the communication interface processes the next recipient.

For more information on activating the **Recognition** option, refer to Creating a Recipient *(see page 475)*.

Configuration from the Front Panel

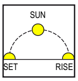
The **Message** function block cannot be configured from the front panel of the smart relay. This function must be configured from the programming software.

Sunrise/Sunset

Description

This function calculates the sunrise and sunset times in relation to the longitude and latitude on the function block inputs. The discrete output **Sun Up or Down** is at a high level when the sun is up and at a low level.

Access

The 

function is accessible from the **FBD** function bar.

Inputs

The **Sunrise/Sunset** function block has the following inputs:

- **Activation**: This is a boolean. Until this input is activated, the discrete-type output is inactive and the four outputs (**Sunrise hour**, **Sunrise minute**, **Sunset hour** and **Sunset minute**) are equal to 0. This input is active if it is not connected.
- **Longitude**: This integer has a value between -18000 and 18000, representing the longitude of the equipment location from 180°00 West to 180°00 East.
- **Latitude**: This integer has a value between -9000 and 9000, representing the latitude of the equipment location from 90°00 South to 90°00 North.
- **Time zone**: This integer represents the time difference in minutes between the UTC and the country where the controller is located.

NOTE: Longitude and **Latitude** must be entered in decimal degrees (hundredths of a degree), not in sexagesimal degrees.
To convert the geographical coordinate **Longitude** (or **Latitude**) **m°n'** of a point from degrees, minutes to **h** in hundreths of degrees, apply the formula **h**=100x(**m**+(**n**/60)):

- if **Longitude** is West (or **Latitude** is South), negate **h**.
- if **h** is fractional, round it to the nearest integer value.

NOTE: Time zone must be entered in minutes, not in hours.

The following table gives examples of converting geographical coordinates from sexagesimal degrees to decimal degrees and UTC to minutes:

Town	Geographical coordinates in sexagesimal degrees	Time zone: UTC	Longitude in decimal degrees	Latitude in decimal degrees	Time zone (min.)
Los Angeles	34°3' N, 118°15' W	-8	-11825	3405	-480
Brasilia	15°30' S, 47°51' W	-3	-4785	-1550	-180
Moscow	55°45' N, 37°37' E	+3	3762	5575	+180
Canberra	35°18' S, 149°8' E	+10	14913	-3530	+600

Outputs

The **Sunrise/Sunset** function block has the following outputs:

- **Sun Up or Down**: This boolean is 1 (sun is up) or 0 (sun is down).
- **Sunrise hour**: This integer represents the sunrise hour (value range: 0..24) in relation to the geographical position and date of the controller.
- **Sunrise minute**: This integer represents the sunrise minutes (after **Sunrise hour**) in relation to the geographical position and date of the controller.
- **Sunset hour**: This integer represents the sunset hour (value range: 0..24) in relation to the geographical position and date of the controller.
- **Sunset minute**: This integer represents the sunset minutes (after **Sunset hour**) in relation to the geographical position and date of the controller.

NOTE:

- These four integer values **Sunrise/sunset hour and minute** are displayed in Simulation and Monitoring modes.
- Date and time must be correctly set in the controller to get correct **Sunrise/sunset hour and minute** values as outputs.

NOTE: In some regions of the globe and at specific dates there is no sunrise nor sunset. In this case, **Sun Up or Down** is inactive and the other four outputs are equal to 0.

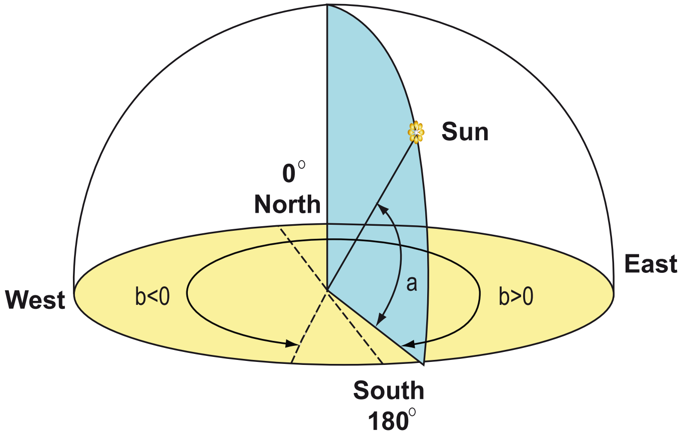
Suntrack

Description

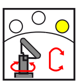
This function calculates the position of the sun. This depends on the two angles calculated by the function:

- a the elevation angle
- b the azimuth angle

The following diagram shows the positions angles (a,b) of the sun:



Access

The  function is accessible from the **FBD** function bar.

Inputs

The **Suntrack** function block has the following inputs:

- **Activation**: This is a boolean. Until this input is activated, the two outputs (**Elevation angle a** and **Azimuth angle b**) are equal to 0. This input is active if it is not connected.
- **Longitude**: This integer has a value between -18000 and 18000, representing the longitude of the equipment location from 180°00 West to 180°00 East.
- **Latitude**: This integer has a value between -9000 and 9000, representing the latitude of the equipment location from 90°00 South to 90°00 North.
- **Time zone**: This integer represents the time difference in minutes between the UTC and the country where the controller is located.

NOTE: Longitude and **Latitude** must be entered in decimal degrees (hundredths of a degree), not in sexagesimal degrees.

To convert the geographical coordinate **Longitude** (or **Latitude**) **m°n’** of a point from degrees, minutes to **h** in hundreths of degrees, apply the formula **h**=100x(**m**+(**n**/60)):
- if **Longitude** is West (or **Latitude** is South), negate **h**.
- if **h** is fractional, round it to the nearest integer value.

NOTE: Time zone must be entered in minutes, not in hours.

The following table gives examples of converting geographical coordinates from sexagesimal degrees to decimal degrees and UTC to minutes:

Town	Geographical coordinates in sexagesimal degrees	Time zone: UTC	Longitude in decimal degrees	Latitude in decimal degrees	Time zone (min.)
Los Angeles	34°3’ N, 118°15’ W	-8	-11825	3405	-480
Brasilia	15°30’ S, 47°51’ W	-3	-4785	-1550	-180
Moscow	55°45’ N, 37°37’ E	+3	3762	5575	+180
Canberra	35°18’ S, 149°8’ E	+10	14913	-3530	+600

Outputs

The **Suntrack** function block has the following outputs:
- **Elevation angle a**: This integer represents the height of the sun (90°00 South to 90°00 North).
  - Positive elevation angle: The sun is above the horizon
  - Negative elevation angle: The sun is below the horizon
- **Azimuth angle b**: This integer represents the rotation needed to position oneself facing the sun from North direction, value between -18000 and 18000 (180°00 West to 180°00 East).

NOTE: These outputs are displayed in both the Simulation and Monitoring modes.

NOTE: Date and time must be correctly set in the controller to get correct **Elevation angle a** and **Azimuth angle b** values as outputs.

Section 23.7
SFC Functions

Subject of this Section

This section provides information on the different SFC (Sequential Function Chart) functions using FBD language.

What Is in This Section?

This section contains the following topics:

Topic	Page
Presentation of SFC Functions	349
Using the SFC Steps and Transitions	352
Use of divergences to AND	355
Use of Divergences to OR	357
Use of convergences to AND	361
Use of convergences to OR	363
Use of SFC Loops	367
Initialization of an SFC Chart at the Start of a Program	368
Initialization of SFC Charts	371
Reinitialization of an SFC Chart When Program is Running	372
SFC Functions	375
INIT STEP (SFC Initial Step)	376
RESET INIT (Resettable Initial SFC Step)	377
STEP (SFC Step)	378
DIV AND 2 (Divergence to AND with 2 SFC Branches)	379
CONV AND 2 (Convergence to AND with 2 SFC Branches)	380
DIV OR 2 (Divergence to OR with 2 SFC Branches)	381
CONV OR 2 (Convergence to OR with 2 SFC Branches)	382
Errors and Advisories Detected in an SFC Chart	383

Presentation of SFC Functions

General

SFC (Sequential Function Chart) functions are similar to Grafcet.

Grafcet is used to represent the functioning of a sequential automation operation in a structured and graphic form.

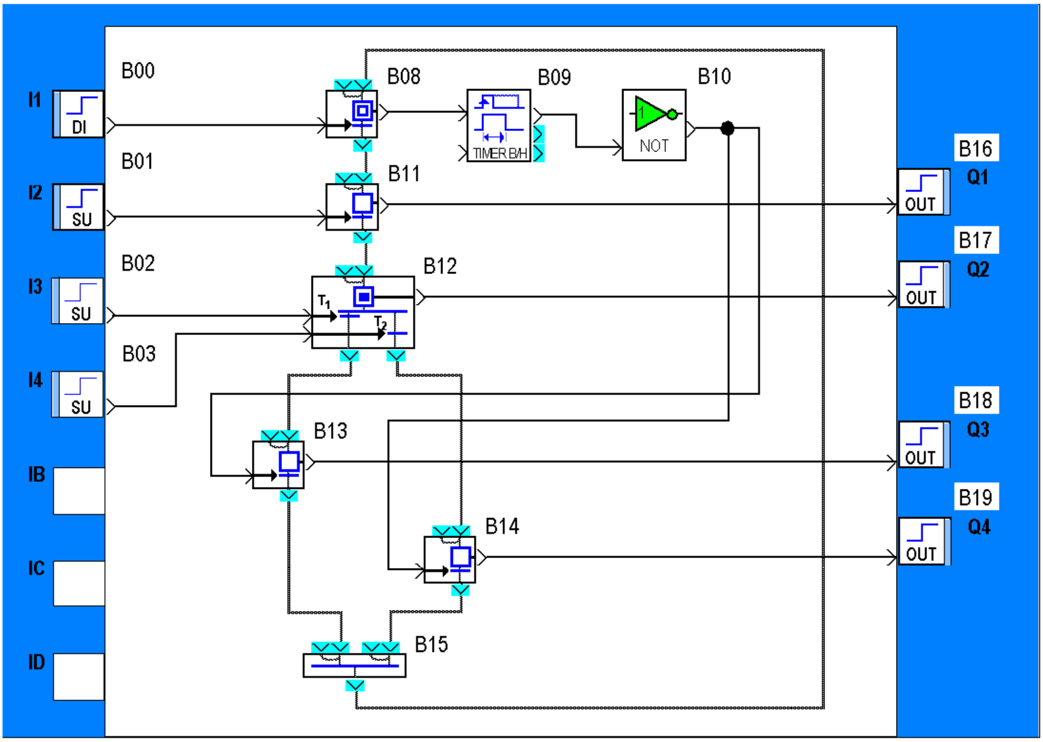
The principle is simple: a graph containing SFC functions is read from top to bottom, and is principally composed of:

- Steps
- Transitions

Steps are placed in succession, and are controlled by transitions. When a step is active, you must wait for the following transition to become active before carrying on to the following step.
Associated with each step is a **Step output** action, which sends orders to other functions (discrete output/logical/standard functions).

FBD Representation

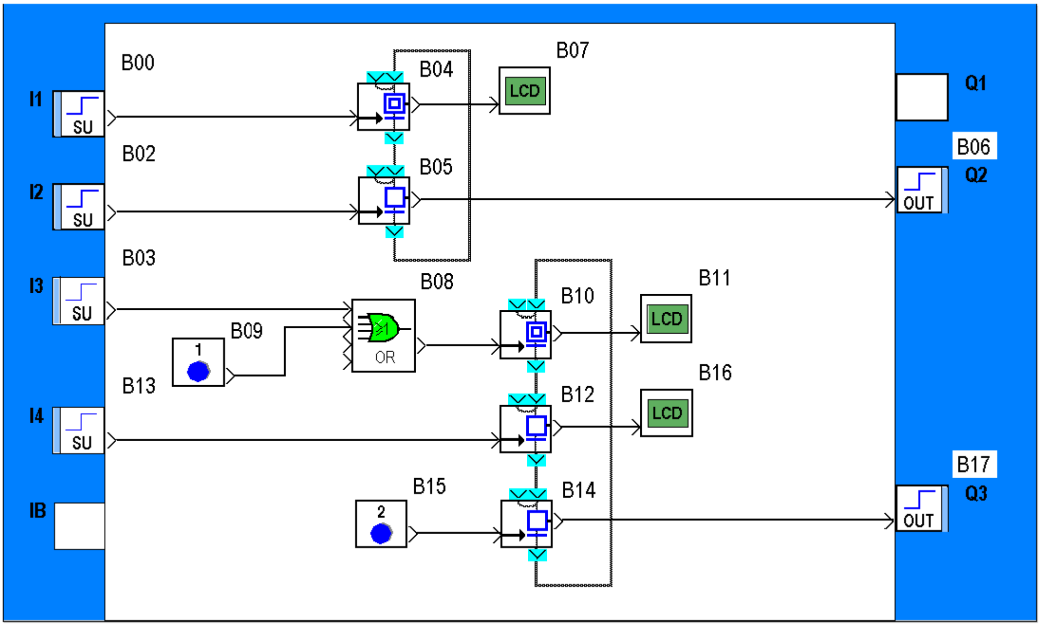
The following diagram shows an example application with SFC functions in FBD language:



Independent Charts

An **independent chart** is a set of SFC functions interconnected by input and output function links. Each chart performs an automation function. In a wiring diagram it is possible to create various independent charts.

The following diagram shows an example of 2 independent charts in a wiring diagram:



Using the SFC Steps and Transitions

Description

Steps and transitions can be used to represent and control consecutive operating phases.

Each operating phase is represented by a symbol called a **step**. When this operating phase takes place, the step is said to be active. In this case the step is said by definition to contain a **status token**.

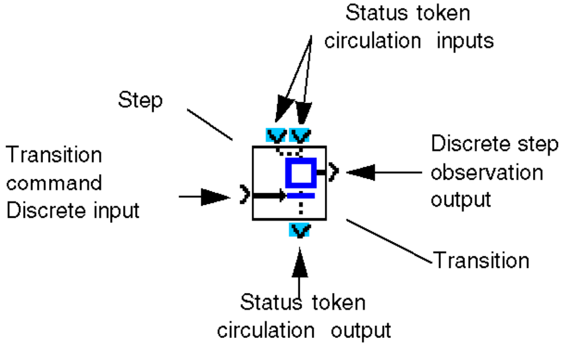
The step's active status is seen by the setting to ON of a **observation Discrete** of the step.

For the operating phase to terminate, the phase ending must be authorized or commanded. For this, a **transition command Discrete** input is set to ON.

The **transition** is then said to be passing and the status token crosses it. It therefore disappears from the step and is led to the **status token circulation output**. Consequently, the observation Discrete is set to OFF.

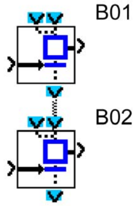
When the operating phase is terminated, the step becomes inactive and the observation Discrete switches to OFF.

Illustration:



The switching off of an operating phase (B01) is immediately followed by the start-up of the following operating phase (B02). The following operating phase is also symbolized by a new step, and its end is also controlled by a transition.

Illustration:



To show the fact that the switching off of operating phase B01 is followed (in sequence) by operating phase B02, the B01 status token circulation output is linked to one of the circulation inputs of the B02 status tokens.

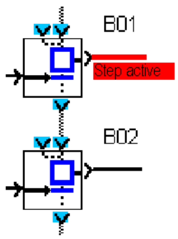
In this case, when the switch to ON of the B01 transition command makes this passing, the token present in the B01 step "falls" through the passing transition to the B02 step, where it stays as long as the Discrete command input of the B02 transition remains set to OFF (blocked transition).

The Discrete observation output for the B02 step activity switches to ON. As soon as the B02 transition becomes passing, the token now present in step B02 escapes by the status token circulation output, the operating phase associated with the step of block B02 ends and the Discrete observation output of step 2 switches to STOP.

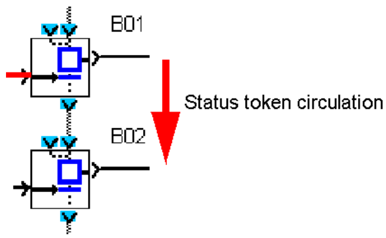
Operation

The mechanism is broken down into four steps.

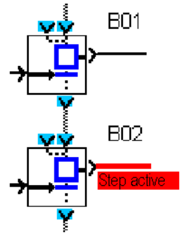
Phase 1, operation in progress: step 1 active (stable status)



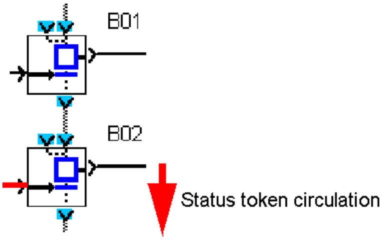
End of operating phase 1: transition 1 active (momentary status)



Phase 2, operation in progress: step 2 active (stable status)



End of operating phase 2: transition 2 active (momentary status)



If step 1 is not active, the associated operating phase (B01) is not in progress, and by definition, the status token is not present in step 1. Therefore, turning ON the Discrete input of transition command 1 that authorizes the transition will have no effect as there is no token in step 1 (the token cannot be passed on).

The Discrete inputs controlling each transition and Discrete outputs that observe each step can be connected to the other FBD blocks with Discrete inputs and outputs.

For example, a Boolean combination of inputs can command transition 1, a button can command transition 2, the step 1 observation Boolean can switch a relay and the step 2 observation Boolean can activate the message display.

Use of divergences to AND

Description

The **divergence to AND** is used to represent and command simultaneous operating phases. This representation of a string of operating phases describes the opposite mechanism to the **AND convergence** *(see page 361)*.

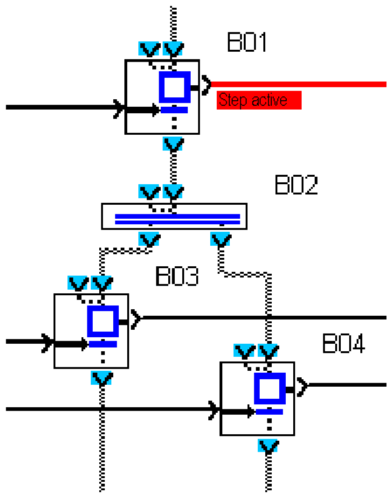
An operating phase (B01) can be followed by two operating phases that take place at the same time, and which assign, for example, two command devices to the same hardware.

To represent this operating mode, a function called **Divergence to AND with 2 SFC branches** (or DIV AND 2) is used, which is linked to two step functions that each symbolize one of the simultaneous operating phases.

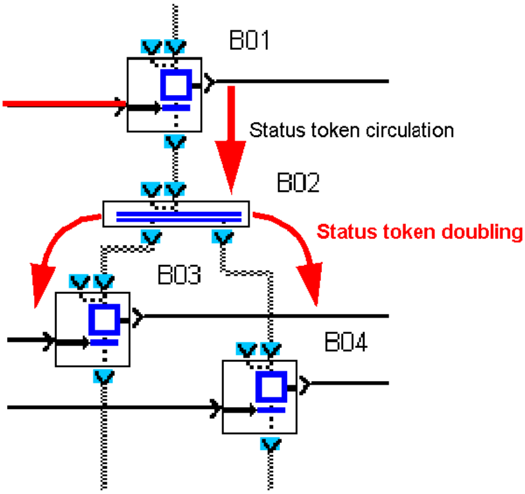
When the transition command input of block B01 is set to ON, the token, if present in step B01, migrates from this step, through transition B01, then doubled into two tokens which, each one falling into steps B03 and B04, show the activation of the two parallel operating phases.

Mechanism

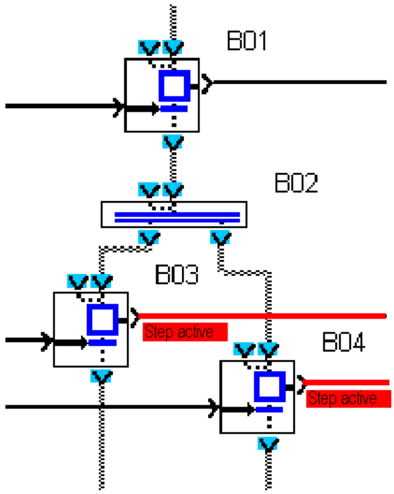
End of operating phase 1 in progress: step B01 active (stable status)



End of operating phase 1: transition 1 active (momentary status)



Operation phases 2 and 3 simultaneously in progress: steps 3 and 4 active (stable statuses)



Use of Divergences to OR

Description

The **divergence to OR** is used to follow up an operating phase with one or two further operating phases from a choice of two possible phases.

This representation of a string of operating phases is the opposite mechanism to that of **convergence to OR** *(see page 363)* (CONV OR 2).

A B01 operating phase can be followed by two operating phases which form a non-exclusive alternative: operating phase B02, B03 or both are activated at the end of operating phase B01.

To represent this operating mode, a function called **Divergence to OR with 2 SFC branches** (or DIV OR 2) is used, which is linked to two step functions that each symbolize one of the two operating phases available (B02 and/or B03).

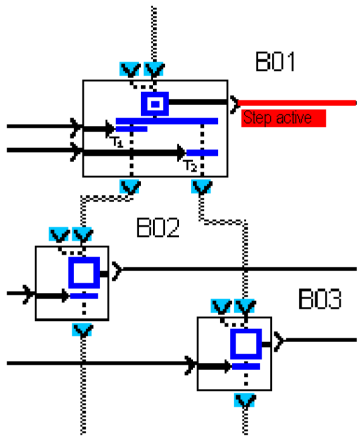
If the status token is present in the step (operating phase B01), the choice is made by forcing to ON one and/or the other of the command inputs of each B01 transition, which are respectively linked downstream to steps B02 and B03.

This therefore causes the end of operating phase B01, the migration of the token from step B01, through the passing transition(s) (with its command input set to ON) to the step connected to it.

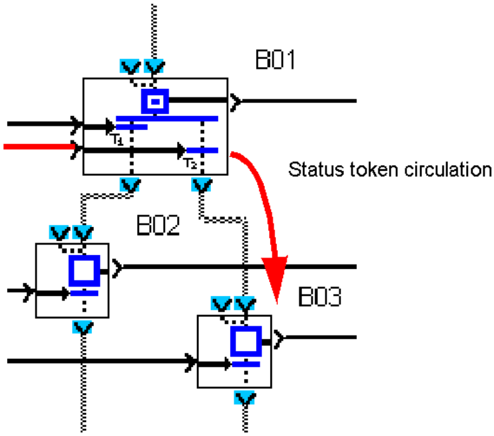
Examples

Example 1: one of the two transitions available is active.

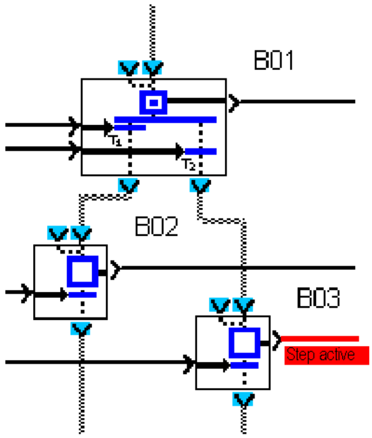
Phase 1, operation in progress: Step B01 active (stable status):



End of operating phase 1: B01 transition 2 active (momentary status):

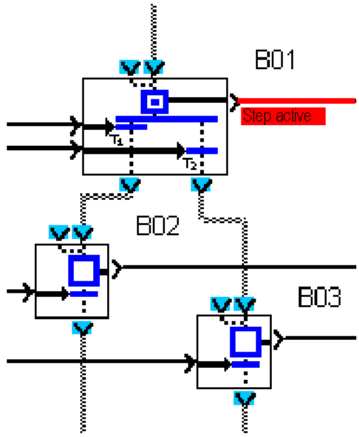


Phase 3, operation in progress: step B03 active (stable status):

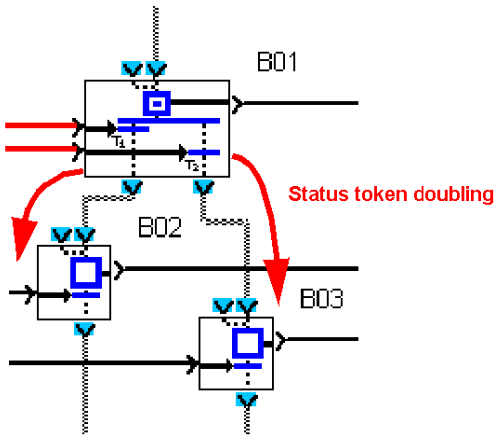


Example 2: both transitions are passing at once.

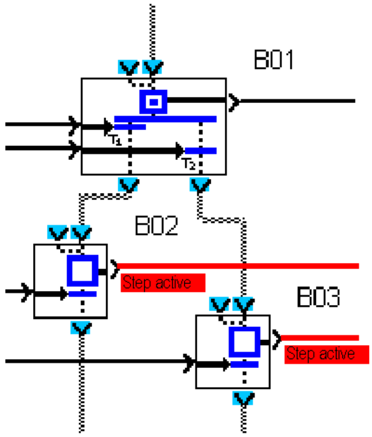
Phase 1, operation in progress: Step B01 active (stable status):



End of operating phase 1: B01 transition 1 and 2 active (momentary status):



Operating phase 2 and 3 in progress: steps B02 and B03 active (stable statuses):



NOTE: If you want the choice between the two following operating phases to be exclusive, one of the two transitions must be commanded by an **AND** combining the command of the first transition with the reverse of the second transition command.

Use of convergences to AND

Description

The **convergence to AND** is used to sequence a single operating phase after simultaneous operating phases. This representation of a string of operating phases describes the opposite mechanism to the **AND divergence** *(see page 355)*.

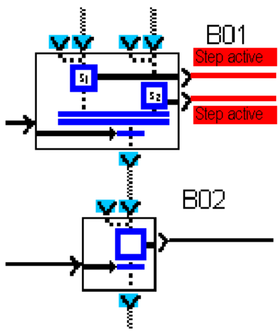
Two simultaneous operating phases (B01 steps 1 and 2) can be followed by a single operating phase, which can only be triggered after the simultaneous end of the two previous phases.

To represent this operating mode, an SFC function called **Convergence to AND with 2 SFC branches** (or CONV AND 2) is used, which is wired to the two upstream step functions, each of which symbolizes one of the simultaneous operating phases, and to a downstream step, which symbolizes the single phase which links to the two previous operating phases.

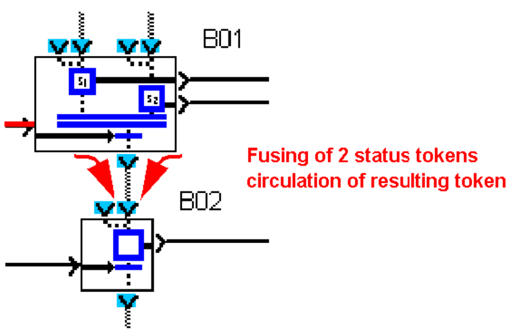
Each of the tokens migrates from its respective step, through its associated transition, fuses into a single token, which, falling into step B02, shows the activation of the next single operating phases.

Mechanism

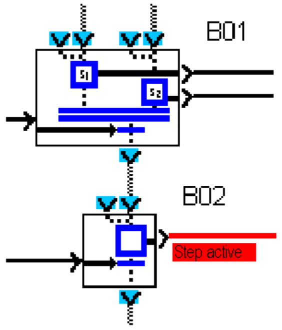
Operating phase 1 and 2 in progress: B01 step 1 and 2 simultaneously active (stable status):



End of operating phase 1 and 2: transition B01 active (momentary status):

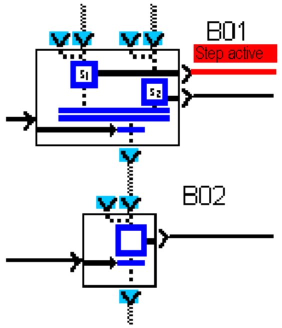


Phase 3, operation in progress: step B02 active (stable status):

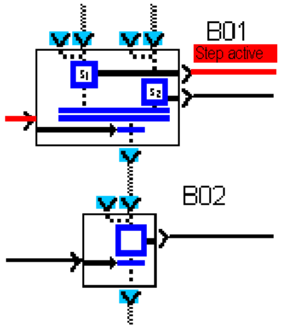


If a single token is present in one of the upstream steps and the other is empty (inactive), then even if the transition is set to ON, nothing happens. The step containing the token stays active (Discrete observation output of the step set to ON) and the downstream step (B03) stays inactive.

Phase 1, operation in progress: only step 1 is active (stable status) but step 2 is inactive:



Phase 1, operation in progress: transition B01 active (stable status):



Use of convergences to OR

Description

Convergence to OR is used to sequence one same operating phase after one or the other of two previous operating phases (simultaneous or not). This representation of a string of operating phases describes the opposite mechanism to the **OR divergence** *(see page 357)* (DIV OR 2).

Two operating phases, simultaneous or not, (steps B01 and/or B02) are followed by a single operating phase which can only be triggered after the end of one of the two previous phases (once transition B01 or B02 is set to ON).

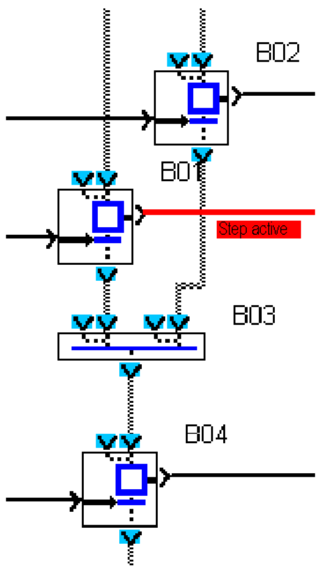
To represent this operating mode, an SFC function is used called **Convergence to OR with 2 SFC branches** (or CONV OR 2), which is linked to the two upstream transitions, each of which controls the end of an operating phase (step B01, step B02), and to a downstream step (B03) which symbolizes the single phase which is linked after one or the other of the two previous operating phases.

The first command input that makes a transition passing while the activation token is present in the associated step lets the token migrate to the downstream step (B03) which symbolizes the commitment of operating phase 3.

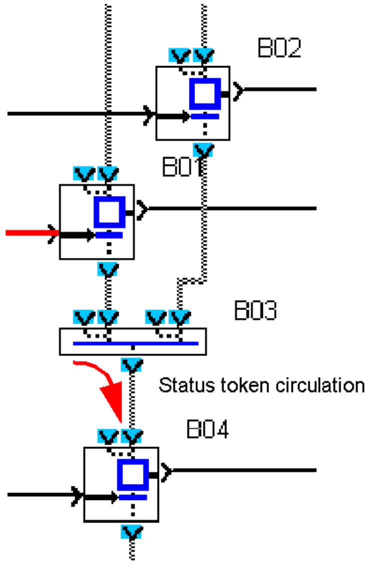
Example

Example 1: transition 1 is made passing while operating phase 1 is in progress.

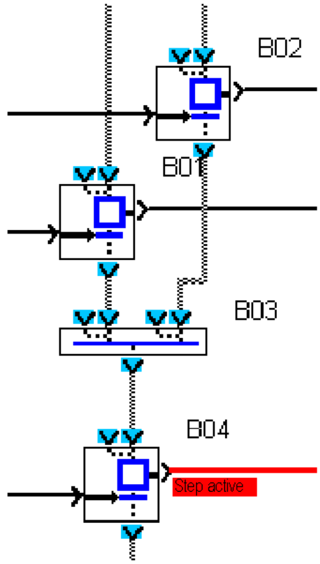
Phase 1, operation in progress: B01 step 1 active (stable status):



End of operating phase 1: transition B01 active (momentary status):

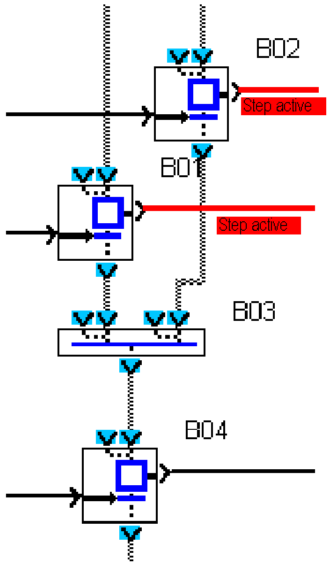


Phase 3, operation in progress: B04 step 1 active (stable status):

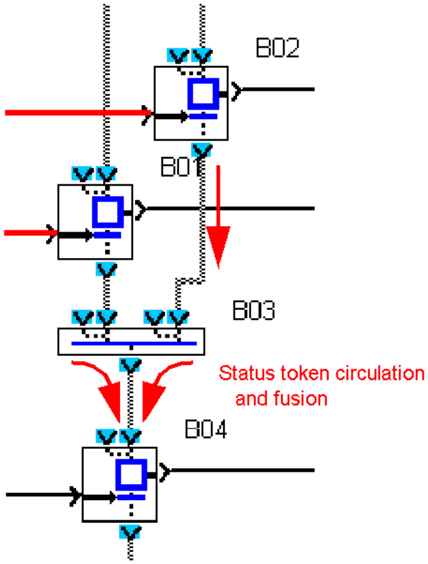


Example 2: transition 1 and transition 2 are made simultaneously passing while operating phases 1 and 2 are simultaneously in progress.

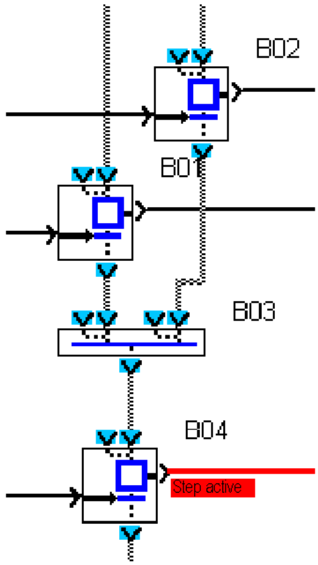
Operating phases 1 and 2 simultaneously in progress: step B01 and B02 simultaneously active (momentary status):



Simultaneous end of operating phases 1 and 2: transition B01 and B02 simultaneously active (momentary status):



Phase 3, operation in progress: Step B04 active (stable status):



Use of SFC Loops

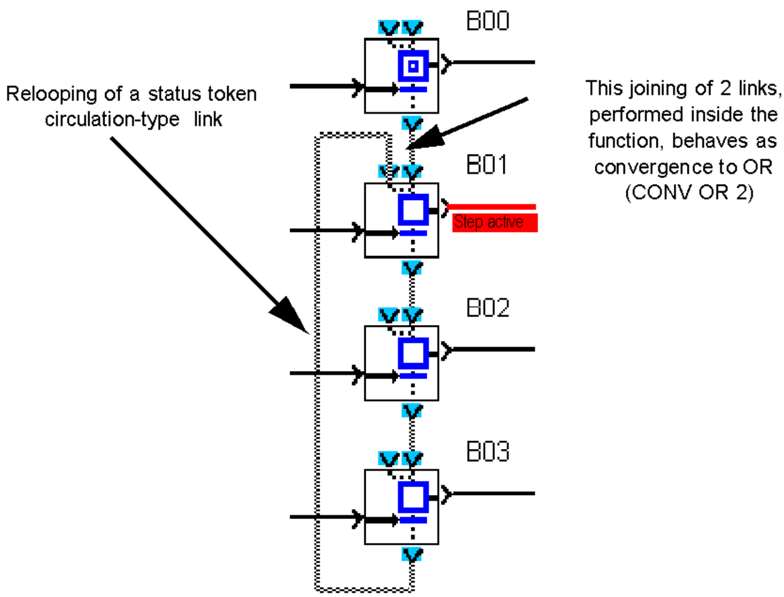
Description

Loops are used to build a sequence of operating phases without end.

Most PLCs are designed to operate by continually linking a sequence of operating phases after an initialization phase. To create this link, the programmer must loop to itself "status token circulation"-type links.

Example

End of operating phase 1 in progress: step B01 active (stable status)



Initialization of an SFC Chart at the Start of a Program

Description

On launching (initializing) the program containing an SFC, you must know which operating phase needs to be activated first, and therefore which step contains a status token at the time of initialization.

To show this step in the chart, it is essential to use at least one SFC function called **Initial SFC step** (`INIT STEP`) or **Resettable initial SFC step** (`RESET-INIT`) per independent SFC.

An independent SFC is a set of SFC functions connected together by links between the token inputs/outputs (circulation of status tokens).

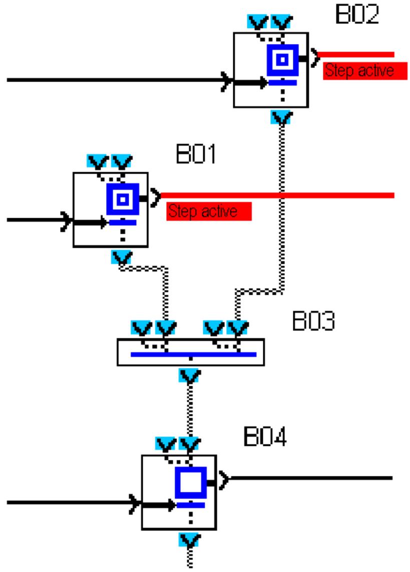
On launching the program (once the INITIALIZE AND SWITCH ON order is executed):

- The charts that contain one or more **Initial SFC step** (`INIT STEP`) functions are automatically initialized. This or these `INIT STEP` functions contain a status token which symbolizes the same number of active operating phases.
  The other steps belonging to the other functions contain no token, and the operating phases they symbolize are inactive.
- This automatic initialization also takes place on restart after a power outage. The positions the status tokens had at the time of the power outage are lost,
- In the charts containing a `RESET-INIT` function, it is mandatory, right at the start of the program, to place an ON signal on the **Reinitialization** input, and disable the smart relay outputs, which may be subject to interference. On restart after a power outage, the positions the status tokens had at the time of the power outage are restored.

Example

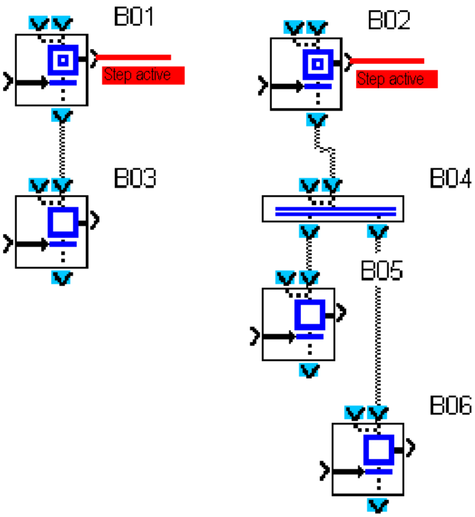
Example 1: SFC with two `INIT STEP` functions.

Initialization and switching on of the program, initial operating phases 1 and 2 simultaneously in progress, steps B01 and B02 simultaneously active (stable statuses)



Example 2: Two independent SFC charts each have an **Initial SFC step** function.

Initialization and switch on of the program, initial operating phases 1 and 2 simultaneously in progress, step B01 and B02 simultaneously active in two independent SFCs (stable statuses)



Initialization of SFC Charts

At a Glance

A program containing one or more SFC charts must be initialized when launched. To perform this initialization you must insert at least one **INIT STEP** *(see page 376)* function or a **RESET INIT** *(see page 377)* function in each of the independent charts.

If a chart contains the **RESET INIT** *(see page 377)* function, it can also be initialized when the program is running.

Initialization at Startup or on Power Return

On program startup, when the INITIALIZE AND SWITCH ON command is executed, or when power is restored, the following occurs:

- The **Step outputs** of the **INIT SFC** or **RESET INIT** functions are activated and the other chart functions are deactivated,
- The former step states are lost.

If a chart contains a **RESET INIT** function, the steps are restored to the states they were in at the time of the power outage.

NOTE: At the start of an SFC chart it is mandatory to connect a RUN *(see page 258)*-type input to the **Reset input** of the RESET INIT function and to disable the outputs of the smart relay which are dependent on the outputs of the SFC chart steps.

NOTE: At the start of an SFC chart it is mandatory to connect the **Cold restart** output of the STATUS *(see page 329)* function to the **Reset** of the RESET INIT function and to disable the outputs of the smart relay which are dependent on the outputs of the SFC chart steps.

Initialization in Progress

When a program containing one or more independent SFC charts is running, a chart containing the **RESET INIT** function can be reset independently of the other SFC charts. This initialization is performed by activating the **Reset** of the **RESET INIT** function which achieves the following:

- The **Step outputs** of the **INIT SFC** and **RESET INIT** functions are activated and the other chart functions are deactivated,
- The functions of the other independent charts are not affected.

As long as the **Reset** is active, the steps are forced as described above regardless of the transition values of the chart functions.

Reinitialization of an SFC Chart When Program is Running

Description

When running a program containing one or more independent SFC charts, a chart containing the **RESET-INIT** function may be reinitialized independently of the other SFC charts. This initialization is triggered by setting to ON the Discrete input called **Reinitialization** of the `RESET-INIT` function.

This input can be connected to the other FBD blocks using Discrete outputs. For example, a Boolean combination of inputs can command this initialization input.

During execution of the program, once the **Reinitialization** input of the `RESET-INIT` function switches to ON, each `INIT STEP` function and the `RESET-INIT` function belonging to the same SFC each contain a status token that symbolizes the same number of active operating phases.

The other steps belonging to the other functions of the same SFC contain no token: the operating phases they symbolize are inactive.

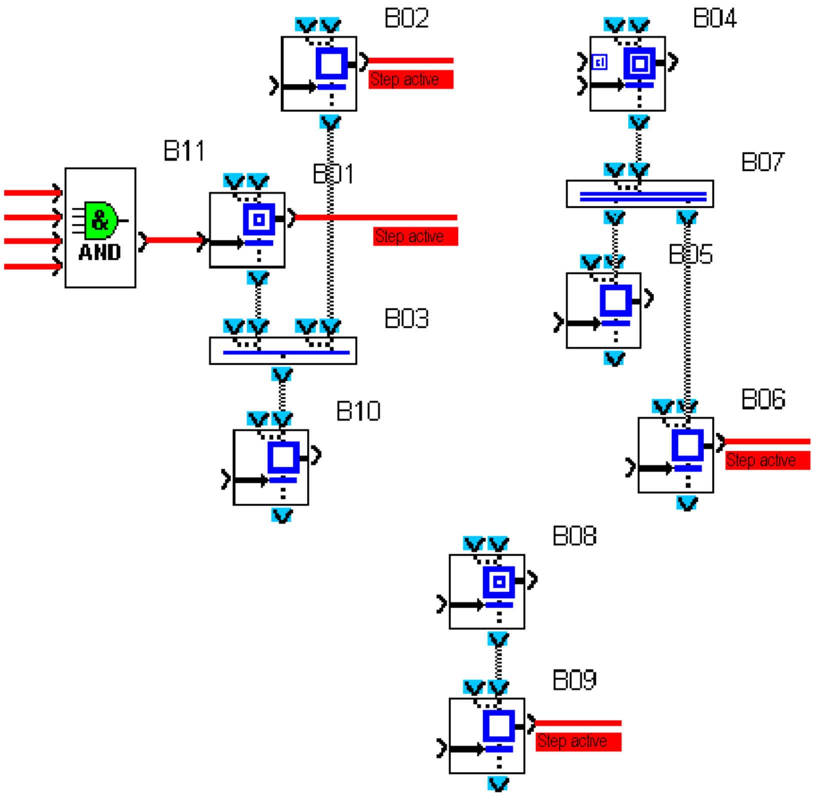
The other functions belonging to other SFCs independent from the previous one are not assigned.

As long as the **Reinitialization** input is set to ON, the steps are forced as described above without taking into account the values applied to the command inputs associated with all transitions of the chart functions.

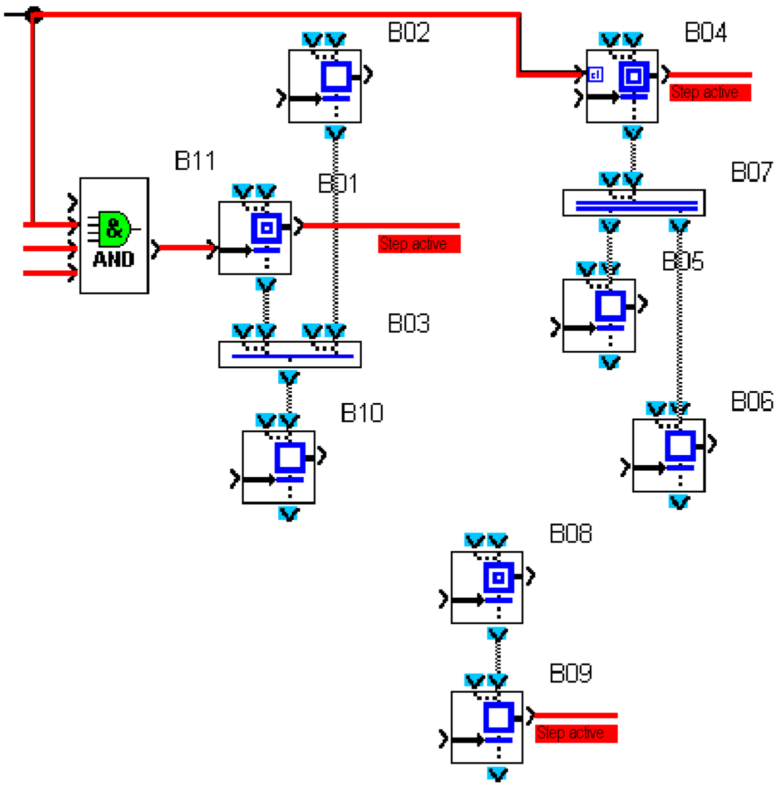
Example

One SFC has an `INIT STEP` function and a `RESET-INIT` function, a second SFC independent from the first contains a single `RESET-INIT` function, a third SFC independent from the first two contains no `INIT STEP` function.

Initialization while the program is on. Initial phases 1 and 2 simultaneously in operation, as Clear input of B01 block set to ON. Steps B01 and B02 simultaneously active in the first SFC (stable status). Steps B06 and B09 active in the other two charts are not affected.



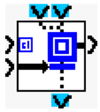
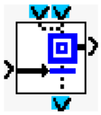
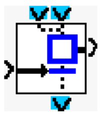
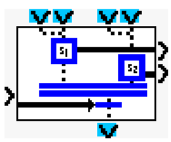
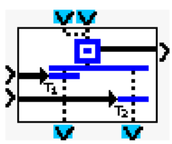
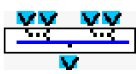
Initialization while the program is on. Initial phase 4 underway, as **Reinitialization** input of B04 block set to ON. Step B04 active in the second SFC (stable status). Steps B03 and B08 active in the other two charts are not affected.



SFC Functions

At a Glance

The following table shows the different functions that make up an SFC program:

Name	Symbol	Description
Initial Step *(see page 376)*		Initial step of an SFC chart.
Resettable initial step *(see page 377)*		Initial step of an SFC chart with initialization of the step by command. Initializes the entire connecting chart containing the reset init.
Step *(see page 378)*		Step which transmits an order to another FBD function.
Divergence to AND *(see page 379)*		Transition of one or two steps toward two steps.
Convergence to AND *(see page 380)*		Transition of two simultaneous steps toward one step.
Divergence to OR *(see page 381)*		Transition of a step toward one or two steps.
Convergence to OR *(see page 382)*		Transition of one to four steps toward a single step.

INIT STEP (SFC Initial Step)

Description

The **INIT STEP** function is an initial step of an SFC chart. It operates normally as follows:

- If **Input 1** or **Input 2** is active, then the **Step output** is activated and remains active even after the inputs have disappeared.
- If the **Transition** input is active, then the **Step output** is deactivated and the **Step transition output** is activated.
- If none of the inputs is active and only the **Step output** is inactive, then the output remains inactive.

NOTE: An SFC chart must contain at least one `INIT STEP` function. Each of the program's independent charts can contain several `INIT STEP` functions.

If there is no `RESET INIT` function in the SFC chart, then the `INIT STEP` function is automatically initialized in the following cases:

- Beginning of a simulation session.
- When switching to **RUN** mode.
- When normal operation is resumed following a power outage.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses:

- Two inputs, **Input 1** and **Input 2** to activate the step output.
- A **Transition** input to activate the step located downstream from this one.

NOTE: If not connected, inputs are in the inactive state.

The function provides:

- A **Step output**
- A **Step transition output**

RESET INIT (Resettable Initial SFC Step)

Description

The **RESET INIT** function can be used, when the **Reset** function is activated:

- To activate the **Step output** for the function, which is the initial step of the SFC chart,
- To reinitialize the other active steps in the chart to which it belongs.

If the **Reset** input is not active, it operates in the following manner:

- If **Input 1** or **Input 2** is active, then the **Step output** is activated and remains active even after the inputs have disappeared,
- If the **Transition** input is active, then the **Step output** is deactivated and the **Step transition output** is activated,
- If none of the inputs is active and only the **Step output** is inactive, then the output remains inactive.

During a power outage, this function enables current values of the chart to be saved and retrieved when power is restored.

NOTE: An SFC chart can only contain a single **RESET INIT** function. Each of the program's independent charts can contain a single **RESET INIT** function.

NOTE: At the start of an SFC chart, it is mandatory to connect the **Cold start** output for the STATUS *(see page 329)* function to the **RESET** input of the RESET INIT function and disable the outputs of the module which are dependent on the outputs of the SFC chart steps.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses:

- Two inputs, **Input 1** and **Input 2** to activate the step output,
- A **Reset** input for the program and its steps,
- A **Transition** input to deactivate the step located downstream from this one.

NOTE: If not connected, inputs other than **Reset** are inactive.

The function provides:

- A **Step output**,
- A **Step transition output**.

STEP (SFC Step)

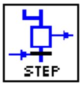
Description

The **STEP** function is a step of an SFC chart. The step symbolizes an operational phase of a control device or PLC.

An action is connected to each **Step output** to transmit commands to other functions (Discrete output, logical, standard functions). It operates in the following manner:

- If **Input 1** or **Input 2** is active, then the **Step output** is activated and remains active even after the inputs have disappeared,
- If the **Transition** input is active, then the **Step output** is deactivated and the **Step transition output** is activated,
- If none of the inputs is active and only the **Step output** is inactive, then the output remains inactive.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses:

- Two inputs, **Input 1** and **Input 2** to activate the step output,
- A **Transition** input to activate the step located downstream from this one.

NOTE: If not connected, inputs are in the inactive state.

The function provides:

- A **Step output**,
- A **Step transition output**.

DIV AND 2 (Divergence to AND with 2 SFC Branches)

Description

The **DIV AND 2** function enables a transition of one or two steps to be simultaneously made toward two steps.

- If **Input 1** or **Input 2 of divergence to AND** is active, then **Output 1** and **Output 2 of divergence to AND** are activated,
- If none of these inputs is active, then **Output 1** and **Output 2 of divergence to AND** are inactive.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses two input s that allow activation of the transition outputs:

- **Input 1 of divergence to AND**,
- **Input 2 of divergence to AND**.

NOTE: If not connected, inputs are in the inactive state.

The function provides two outputs:

- **Output 1 of divergence to AND**,
- **Output 2 of divergence to AND**.

CONV AND 2 (Convergence to AND with 2 SFC Branches)

Description

The **CONV AND 2** function enables a transition of two steps to be simultaneously made toward one step.

- If **Input 1** or **Input 2** is active, then **Step output 1 of convergence to AND** is activated and remains active even after the inputs have disappeared,
- If **Input 3** or **Input 4** is active, then **Step output 2 of convergence to AND** is activated and remains active even after the inputs have disappeared,
- If **Step output 1 of convergence to AND** and **Step output 2 of convergence to AND** are active and the **Transition** input is also active, then:
  - **Output 1** and **Step output 2 of convergence to AND** are deactivated,
  - The **Transition output** is activated.
- If none of these inputs is active, then **Output 1** and **Step output 2 of convergence to AND** are inactive,
- If the **Transition** input is active but **Output 1** or **Step output 2 of convergence to AND** is inactive, **Output 1** or **Step output 2 of convergence to AND** does not change state and the **Transition output** remains inactive.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses:

- Two inputs, **Input 1** and **Input 2** to activate the step output 1,
- Two inputs, **Input 3** and **Input 4** to activate the step output 2,
- A **Transition** input to activate the step located downstream from this one.

NOTE: If not connected, inputs are in the inactive state.

The function provides:

- A **Step output 1 of convergence to AND**,
- A **Step output 2 of convergence to AND**,
- A **Transition output**.

DIV OR 2 (Divergence to OR with 2 SFC Branches)

Description

The **DIV OR 2** function enables a transition of one step to be simultaneously made toward one or two steps.

- If the **Input 1** or **Step input 2** is active, then the **Step output** is activated,
- If the **Transition 1** and the **Step output** is active:
  - The **Step output** is deactivated,
  - **Transition output 2 with divergence to OR** is activated.
- If the **Transition 2** and the **Step output** is active:
  - The **Step output** is deactivated,
  - **Transition output 2 with divergence to OR** is activated.
- If the **Transition 1** and **Transition 2** inputs are active and the **Step output** is active:
  - The **Step output** is deactivated,
  - The **Transition output 1 with divergence to OR** and the **Transition output 2 with divergence to OR** are activated.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses:

- Two inputs, **Input 1** and **Input 2** to activate the step output,
- Two inputs, **Transition 1** and **Transition 2** to activate the transition step output(s).

NOTE: If not connected, inputs are in the inactive state.

The function provides:

- A **Step output**,
- A **Transition output 2 with divergence to OR**,
- A **Transition output 2 with divergence to OR**.

CONV OR 2 (Convergence to OR with 2 SFC Branches)

Description

The **CONV OR 2** function enables a transition of one to four step(s) to be simultaneously made toward one step.

- If **Input 1** or **Input 2** or **Input 3** or **Input 4 of convergence to OR** is active, then **Output 2 of convergence to OR** is activated,
- If none of these inputs is active, then the **Output of convergence to OR** is inactive.

Access

The 

function is accessible from the **SFC** function bar.

Inputs/Outputs

The function uses four input s that allow activation of the transition output.

- **Input 1 of convergence to OR**,
- **Input 2 of convergence to OR**,
- **Input 3 of convergence to OR**,
- **Input 4 of convergence to OR**.

NOTE: If not connected, inputs are in the inactive state.

The function provides an **Output of convergence to OR**.

Errors and Advisories Detected in an SFC Chart

At a Glance

When editing a chart, you can cause structural errors. The Zelio Soft 2 Programming software detects them and generates errors and advisories when:

- Switching from **Edit** mode to **Simulation** mode.
- Switching from **Edit** mode to **Monitoring** mode.
- Using the following commands:
  - **Transfer** → **Transfer Program** → **PC > Module**.
  - **Transfer** → **Compare data from the module using the program**.
  - **Edit** → **Check the program**.

The Zelio Soft 2 Programming software displays a dialog box in the **Compilation results** window with a list of Errors and/or Advisories, and puts a red frame around the function(s) where errors have been found.

The SFC errors are displayed in bold red on the wiring sheet.

Errors

The following table describes the **errors** according to their numbers:

Error code	Description
60	An SFC does not have an initial `INIT STEP` function, and no resettable initial `RESET INIT` function. No step will be active on initialization of the program.
61	An independent SFC has several resettable initial `RESET INIT` functions.

Advisories

The following table describes the **advisories** according to their numbers:

Advisory code	Description
70	This advisory is generated if several advisories of different types are detected.
71	This advisory is generated if an SFC function output is linked directly to several SFC function inputs. The AND Divergence function `DIV AND` can be used to clear this error.
72	This advisory is generated if: • An output from an SFC function is not connected to another function. • None of the inputs from an SFC function except `RESET INIT` and `INIT STEP` are connected to a function.

Section 23.8
Application Function Blocks

Subject of this Section

This section describes the Application Function Blocks (AFBs).

What Is in This Section?

This section contains the following topics:

Topic	Page
Presentation of AFB Functions	385
PID Presentation	386
PID Function	389

Presentation of AFB Functions

General

Application Function Blocks (AFBs) are elements of the FBD language that implement application functions. To save the memory space in the smart relay, the AFBs are only stored in the smart relay memory if they are included in the application.

Using AFB in a Program

Each **AFB** uses space in the smart relay memory, which is characterized by a number of slots.

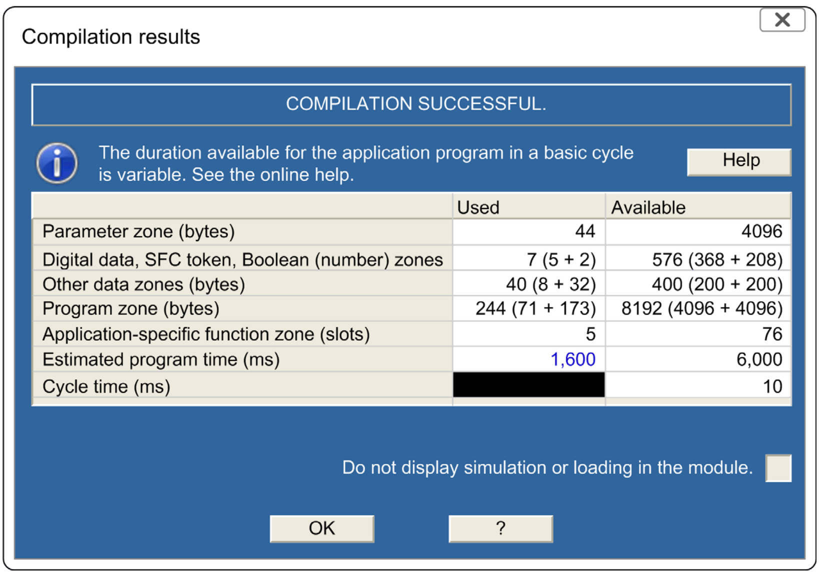
The number of slots used by the **PID** function is 5 slots.

The total number of slots available for the AFBs is 76 slots.

When you place an **AFB** on the wiring sheet, there are two possible scenarios:

- If this is the first time you have included that **AFB** in your program, then the number of slots used increases by the corresponding amount.
- If the **AFB** is already present elsewhere in your program, then the number of slots used does not increase.

The number of slots used and total number of slots are displayed in the **Compilation results** window:



	Used	Available
Parameter zone (bytes)	44	4096
Digital data, SFC token, Boolean (number) zones	7 (5 + 2)	576 (368 + 208)
Other data zones (bytes)	40 (8 + 32)	400 (200 + 200)
Program zone (bytes)	244 (71 + 173)	8192 (4096 + 4096)
Application-specific function zone (slots)	5	76
Estimated program time (ms)	1,600	6,000
Cycle time (ms)		10

PID Presentation

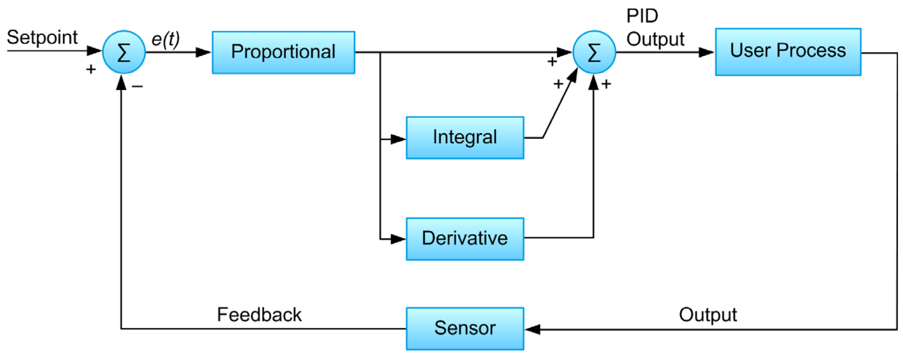
Presentation

The PID function is used for implementing regulation applications.

As a closed loop feedback mechanism, it delivers the controller output and maintains the measured value (feedback) at setpoint level.

If there is any difference between the setpoint and the measured value, the PID algorithm performs mathematical calculations, and a corresponding corrective signal is applied to the process.

The closed loop feedback mechanism is a continuous process. This is illustrated in the following graphic.

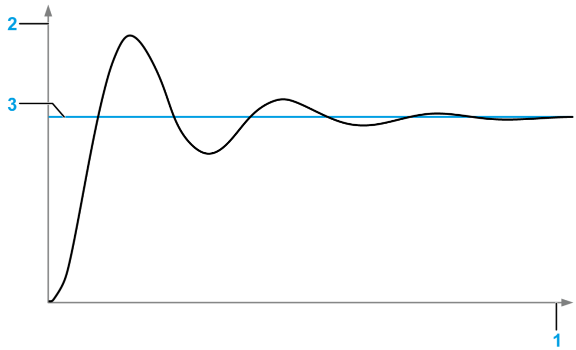


Proportional Control

This control gives output which is proportional to the difference *e(t)* between the desired setpoint and the measured output. It compares the setpoint value with the feedback process value and multiplies the difference by a proportional constant to increase the output towards the setpoint.

When *e(t)* becomes 0, no correction is added to the output. If the output drops below the setpoint, then a correction will be added to increase the output back towards the setpoint. If the output exceeds the setpoint, then a correction will be applied to decrease the output back towards the setpoint.

Proportional control response:

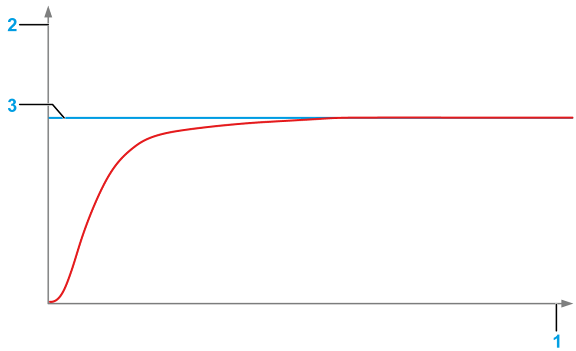


1 Time
2 Output
3 Setpoint

Integral Control

This control integrates the difference *e(t)* between the desired setpoint and the measured output, over a period of time until the difference approaches zero.

Proportional and Integral control response:

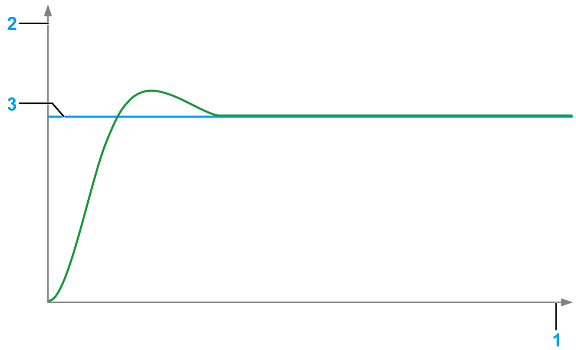


1 Time
2 Output
3 Setpoint

Derivative Control

This control takes into account the rate of change of the difference with respect to time. So it helps anticipate the future behavior of the difference value and, thereby improves the system response.

Proportional, Integral and Derivative control response:



1 Time
2 Output
3 Setpoint

PID Function

General

The PID function is used for implementing regulation applications *(see page 386)*.

Access

The 

function is accessible on the **AFB** function bar.

Inputs

- **Enable**: Enables the PID function input.
- **Measure**: Measures the input (16 bits).
- **Preset Setpoint**: Setpoint value (16 bits). This value is used by the PID function only if the **Setpoint Activation** input is activated.
- **Setpoint Activation**: If active, this input validates the use of **Preset Setpoint** as setpoint for the PID. If not active, the PID **Setpoint Value** is used.

Outputs

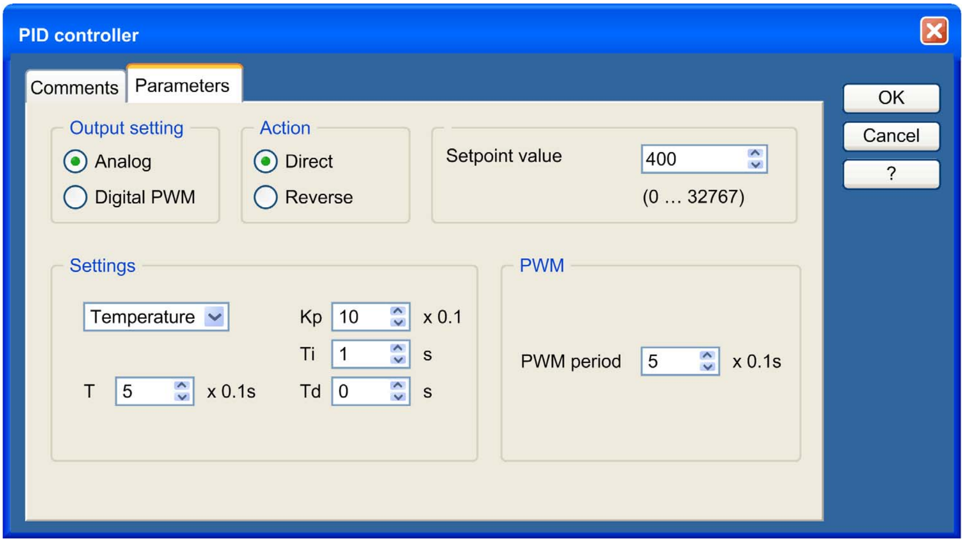
- **Analog Output**: PID analog output [0...1023].
- **PWM Output**: PID PWM output (Boolean).
- K_p: Proportional gain (0.1...100.0).
- T_i: Integral time (1...900 s).
- T_d: Derivative time (0...60 s).
- **Current Setpoint**: Setpoint used by the PID function.

K_p, T_i, T_d are defined as output parameters in order to be modified through the front panel using **TEXT** or **DISPLAY** function.

This is illustrated in the Central Heating System with PWM Valve example. Refer to Zelio Soft 2 Applications Example Guide.

Parameters

The Parameters window is as shown below:



The **Parameters** window is used to define:

- **Output setting**: Select the output type, either Analog or Digital PWM.
- **Action**: Select the action, either Direct or Reverse.
  - **Direct**:
    – If **Measure** < **Setpoint**, the PID function output increases.
    – If **Measure** > **Setpoint**, the PID function output decreases.
    Use this mode for example, in a heating system to increase the temperature when the PID function output increases.
  - **Reverse**:
    – If **Measure** < **Setpoint**, the PID function output decreases.
    – If **Measure** > **Setpoint**, the PID function output increases.
    Use this mode for example, in a cooling system to decrease the temperature when the PID function output increases.
- **Setpoint Value**: It is used if the PID input **Setpoint activation** is not activated. The range of **Setpoint Value** is 0...32767.
- **PWM period**: Period of the PWM output (0.5...10s).
- **Settings**: In this section you can set the individual values for:
  - Proportional gain K_p: 0.1...100.0
  - Integral time T_i: 1...900 s
  - Derivative time T_d: 0...60 s
  - PID sampling time T: 0.5...10 s

You can also choose preset values for these settings depending on your type of application. Five options are available as follows:

Application type	K_p	T_i (s)	T_d (s)	T (s)
Temperature	1.0	180.0	12.0	1.0
Pressure	2.0	30.0	0.0	1.0
Level	1.0	600.0	0.0	1.0
Flow	0.8	12.0	0.0	1.0
Manual settings	1.0	10.0	0.0	1.0

Chapter 24
Programming in FBD Using Zelio Soft 2

Subject of this Chapter

This chapter describes the different functions that can be accessed from the Zelio Soft 2 Programming Software in FBD mode.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
24.1	Creating an FBD Application in the Zelio Soft 2 Programming Software	394
24.2	Manipulating FBD Objects	411
24.3	Debugging and Monitoring an FBD Application in the Programming Software	428

Section 24.1
Creating an FBD Application in the Zelio Soft 2 Programming Software

Subject of this Section

This section describes the different functions linked to programming in the Zelio Soft 2 Programming Software in FBD mode.

What Is in This Section?

This section contains the following topics:

Topic	Page
Configuring FBD Program Editing	395
Inserting Function Blocks	397
Creation of Links Between Function Blocks	399
Function Block Parameters	405
Display Options	407
Draw Function	408
The Find Function	410

Configuring FBD Program Editing

At a Glance

Before creating an FBD program *(see page 26)*, you must first set up several options to facilitate editing, such as:

- Modifying the linking colors.
- Defining the wiring mode.
- Displaying the editing grid.

Linking Colors

The Zelio Soft 2 Programming software can be used to define different colors to display the following:

- Links between function blocks.
- Inputs/Outputs.
- Forced values in Simulation and Monitoring mode.
- The background colors of the edit and supervision windows.

NOTE: It is also possible to configure colors from the File/Preferences *(see page 508)* menu.

Modifying a Color

The following table shows the procedure for modifying colors:

Step	Action
1	From the **Options** menu, select the command **Modify the colors**. **Result**: The **Define the Link Colors** window appears.
2	Click the **Modify** button to the right of the color you wish to modify. **Result**: The **Colors** window appears.
3	Select the new color to apply. **Result**: The **Colors** window appears.
4	Confirm with **OK**.
5	Repeat steps 2 to 4 to modify the other colors.
6	Click **Apply to Document** button to confirm the new configuration.

Wiring Mode

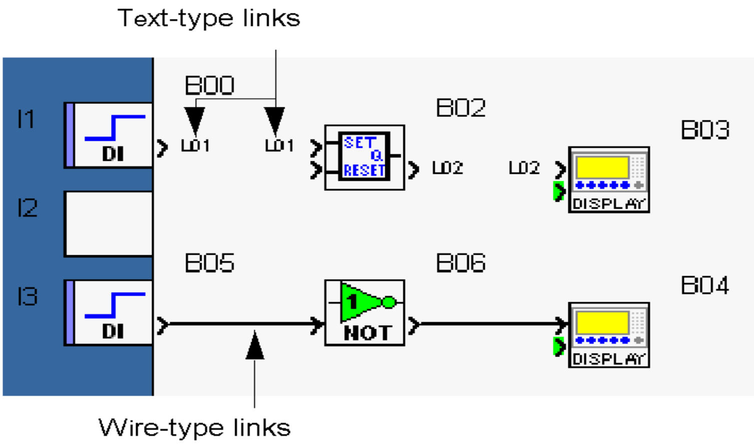
The links between the function blocks can be of the following type:

- **Wire** by clicking **Tools → Wiring Mode → Wire**.
- **Text** by clicking **Tools → Wiring Mode → Text**. The text is inserted by default, and can be modified later.

NOTE: The text displayed at the beginning and end of the link is Lxx type by default (e.g. L04) but can be modified.

Once the type of link is selected, new links will be of the selected type.

The following diagram shows an example of a program with wire- and text-type links:



NOTE: Wire mode linking is the default setting.

Type of Wiring

The Type of Wiring option (wire or text) is only used to modify the selected link:

- Either by clicking **Tools → Type of Wiring**.
  or
- By right clicking.

Displaying the Grid

To help you align blocks in the wiring sheet, you can display a grid by clicking **Display → Grid**.

Inserting Function Blocks

At a Glance

To create an FBD program, you must insert various function blocks in the wiring sheet, then link these together.

The Edit mode is the default mode on opening the application. This is generally accessible by clicking **Mode → Edit** during programming, to switch from one mode to another.

The types of blocks can be placed on the sheet, (including the IN inputs and the OUT outputs).

The only restrictions apply to IN blocks and OUT blocks that can only be positioned on their dedicated contacts.

If there is an incompatibility it will not be possible to place the block. When the contact is empty, and error message is displayed. If the contact already contains a block, a barred circle will appear.

Inserting Function Blocks

The following procedure describes how to insert a function block in a wiring sheet:

Step	Action
1	Select the type of function to insert. ● IN ● FBD ● SFC ● Logic ● OUT ● AFB
2	Click on the icon corresponding to the function to insert.
3	Drag/drop from the function bar to the wiring sheet.
4	Position the function in the required location on the wiring sheet.
5	Repeat steps 2 to 5 to insert all the functions required for the program.

Input Blocks

NOTE: The following input blocks can only be inserted in the input contacts on the left of the wiring sheet:

- Discrete input
- Filtered discrete input
- Analog input
- Filtered analog input
- Integer input

Output Block

NOTE: The following output blocks can only be inserted in the output contacts on the right of the wiring sheet:
- Discrete output
- Integer output

Position of the contacts

It is possible to change the relative positions of the input and output contacts to improve the legibility of the wiring sheet. To do this, proceed as follows:

Step	Action
1	Lengthen the wiring surface if necessary.
2	Designate the contact to move: ● Using the left mouse button, click and hold down on the blue bar of the contact if it contains the diagram of an IN- or OUT- type block. ● If the contact is empty, click and hold down the left mouse button anywhere on contact.
3	Move the contact to the desired location and release the mouse button.

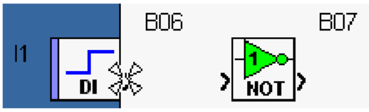
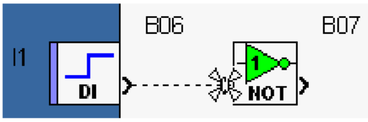
Creation of Links Between Function Blocks

At a Glance

Position the function blocks in the wiring sheet, then link them together. If you have created MACROS *(see page 418)*, they are linked in the same way. You can link one block output (function block or MACRO) to an input of another block or loop an output back to an input of the same block.

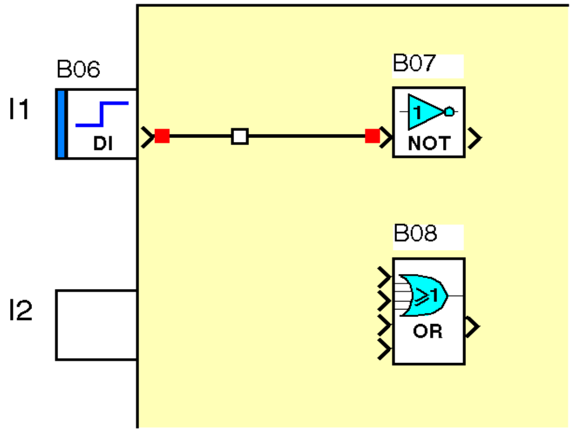
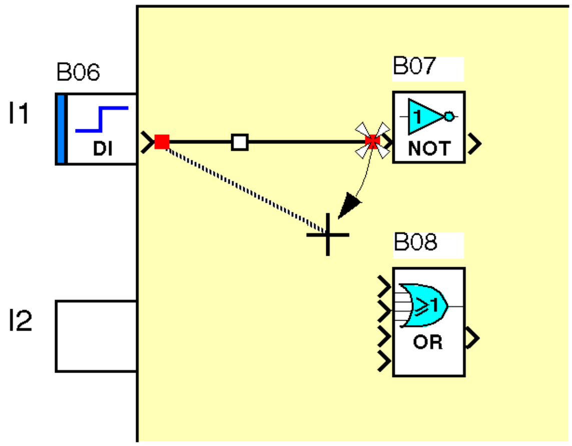
Links between Function Blocks

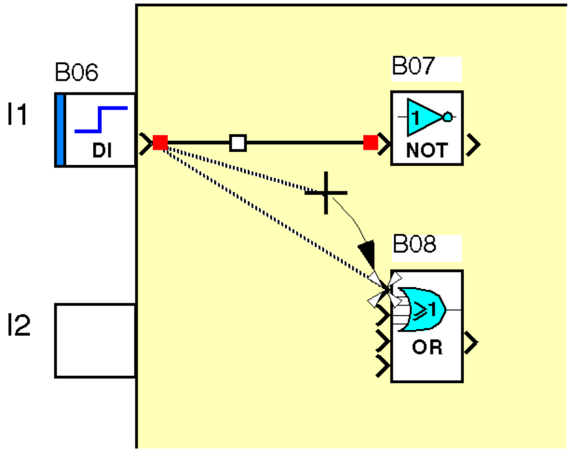
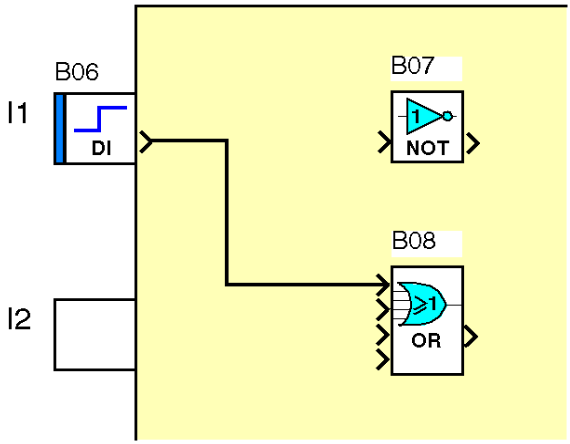
The following procedure describes how to link function blocks together:

Step	Action
1	Click (and hold) the output of the function block. **Result**: the mouse cursor is displayed as a star. 
2	Drag the cursor over a block input. **Result**: the mouse cursor is displayed as a star.  **NOTE:** If the cursor turns into a circle with a line through it when you move it over a block input, this means that the link destination is incorrect (incompatible types).
3	Release the mouse button. **Result**: A line or numbers are shown between the two linked blocks.
4	Repeat steps 1 to 3 to link all the blocks.

How to Change Link Starts and Stops

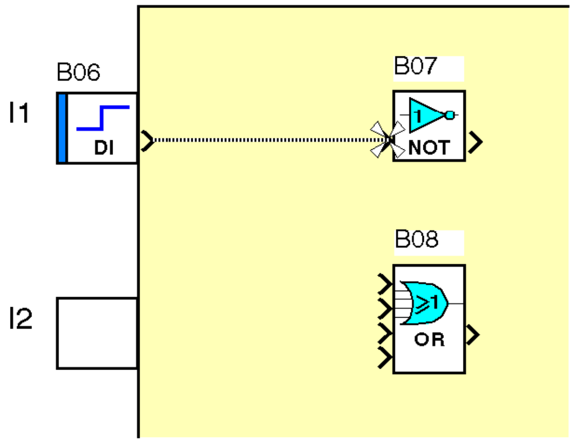
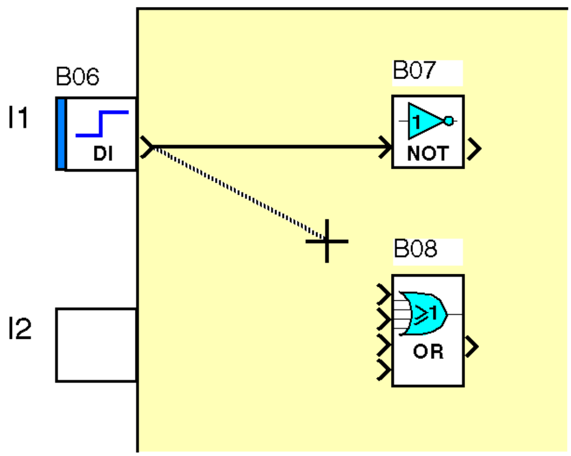
The following procedure describes how to change the start and stop for a link.

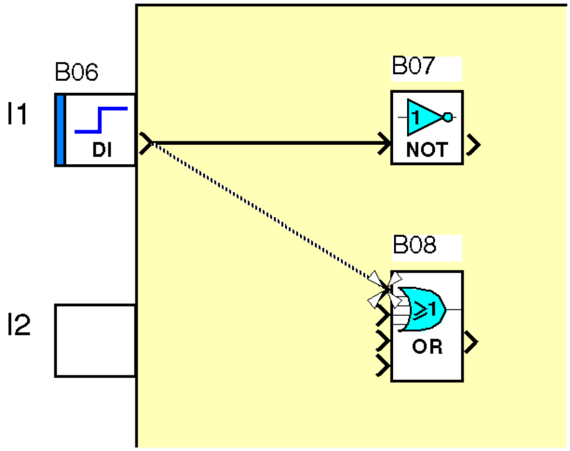
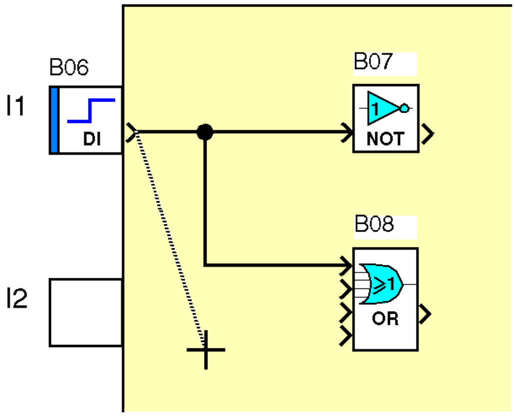
Step	Action
1	Select the link by clicking on it. Illustration 
2	Press the **Shift** key while you click (and hold) the end of the link to modify. **Result:** the cursor is displayed as a star. 

Step	Action
3	**NOTE:** At this step the **Shift** key can be released. Drag the cursor to the input or output of another block. **Result**: the cursor is displayed as a star.  **NOTE:** If the cursor turns into a circle with a line through it when you move it over a block input, this means that the link destination is incorrect (incompatible types).
4	Release the mouse button. **Result**: A line or numbers are shown between the two linked blocks. 

How to Link a Block to Several Others

The following procedure describes how to link a block to several others:

Step	Action
1	Press the **Ctrl** key while clicking on a block output.
2	Drag the cursor to a block input. **Result**: the cursor is displayed as a star. 
3	Hold down the **Ctrl** key and release the mouse button. **Result**: A line or numbers are displayed between the two linked blocks and another dotted line appears. 

Step	Action
4	Hold down the **Ctrl** key and move the cursor to the input of another block. **Result**: the cursor is displayed as a star. 
5	Hold down the **Ctrl** key and click on the input of another block. **Result**: Another link is created.
6	Repeat steps 4 and 5 to create as many links as necessary. 
7	Release the **Ctrl** key and click anywhere on the wiring sheet.

Type of Link

The link is displayed in different ways depending on the type of data transferred via the link:

- Discrete data: Continuous black line,
- Signed integers between -32768 and +32767: Black double line,
- Link between SFC function blocks: Black interwoven lines.

The link display formats described above are the default formats. They can be modified by using the **Options** → **Modify the Colors** → **Link Colors** menu.

Modification of Link Type

The following procedure describes how to change the type of link between function blocks:

Step	Action
1	Select the link to change.
2	Select the **Tools** → **Type of Wiring**.
3	Select the **Wiring** command to change the text link into a wiring link or Select the **Text** command to change the wiring link into a text link. **Result**: The type of the link is modified.

Modification of Link Text

The following procedure describes how to modify the text of the link between function blocks:

Step	Action
1	Click on one of the two link texts to modify.
2	Select the **Tools** → **Type of Wiring** → **Modify text** command. **Result**: The **Modify Link Text** window is displayed.
3	Enter the text.
4	Confirm with **OK**.

Function Block Parameters

At a Glance

Each of the function blocks has a parameters window. This window consists of one, two or three tabs:

- **Comments** tab.
- **Parameters** tab, depending on the function block type (`FBD PRESET COUNT`).
- **Summary** tab, depending on the function block type (`FBD TIME PROG`).

Double-click on the function block to access this window.

Comments Tab

Comment section

In the **Comment** section, you can enter a comment of up to three lines of 30 characters maximum.

On Discrete Inputs *(see page 251)*/Discrete Outputs *(see page 262)* and Analog Inputs *(see page 254)* function blocks, you can also choose the type of function block symbol that will be displayed in the wiring sheet.

When a comment has been added to a function block, an **envelope** symbol is displayed to the bottom right of the block.

Two scenarios may arise:

- If you click on this symbol the content of the comment zone is displayed.
- If the **Display the comment** box of the block is selected the block comment is permanently displayed.

Block number

The following option is also available: **Display the block number**. This option is activated by default.

Symbols used for block

For certain types of block, you can choose specific symbols to be used when shown in the wiring sheet (FBD DI, OUT).

When this function is available, the list of available icons is shown in a menu at the bottom of the window.

To change the icon, double-click on the desired symbol.

Parameters

Most function blocks have a **Parameters** tab. In this tab, you have to set the function block specific parameters. These parameters are described in detail in the help for each of the blocks.

Summary

Some function blocks also have a **Summary** tab (`FBD TIME PROG`). This window lists the actions configured for the block. It represents an overview of the configuration.

Display Options

At a Glance

For an FBD program, several different display options are available with:

- Comments
- Zoom
- Block numbers

Comments

The function blocks can have an associated comment. These comments are displayed above the block in the wiring sheet.

You can choose to display:

- The comment for a block.
- All comments with the command **Display** → **Comment** → **All**.
- No comment with the command **Display** → **Comment** → **None**.

Displaying a Comment

The following table shows the procedure for displaying a function block comment:

Step	Action
1	Select the block. If a comment is associated with the block, the icon is visible.
2	Click on the icon. **Result:** The comment for the block is displayed.

Zoom Function

The command **Display** → **Zoom** allows you to use the zoom to display a part of the program in detail.

Block Numbers

As with the comments, you can choose to display the program function block numbers.

- All of the function block numbers with the command **Display** → **Block numbers** → **All**.
- None of the function block numbers with the command **Display** → **Block numbers** → **None**.

Draw Function

At a Glance

In the edit and supervision sheet, you can create square, ellipse or line forms or text. You can also insert an image in Bitmap format.

The line width (3 widths), line color and background color can also be changed.

Creating a Drawing

The following table shows the procedure for inserting a drawing in the wiring or supervision sheet:

Step	Action
1	Select the **Draw** menu.
2	Select the type of drawing to be created: ● **Row**, ● **Rectangle**, ● **Ellipse**, ● **Text**.
3	Draw the desired form in the wiring or supervision sheet.
4	If you selected **Text**, double-click on the object created and enter the text.

Inserting an Image

The following table shows the procedure for inserting an image in the wiring or supervision sheet:

Step	Action
1	Select the **Draw** menu.
2	Select **Image**. Result: The **Open** window appears.
3	Select the image file in bitmap format.
4	Confirm with **Open**.
5	Left-click on the wiring or supervision sheet. Result: A zone framed by a dotted frame the size of the image appears.
6	Place the zone corresponding to the image on the wiring or supervision sheet.
7	Release the left mouse button. Result: The image appears.

Border

You can create a drawing that is a **rectangle** or **ellipse** with or without a **border**. By default, the border option is selected. If you would like to remove it or confirm your choice, use the **Draw → Border** command. The border color can be modified in the same way as that of a line.

Line Width

The following table shows the procedure for changing a line width or border in a drawing:

Step	Action
1	Select the drawing to modify.
2	Select the **Width** sub-menu from the **Draw** menu.
3	Choose the width type. ● Single line, ● Double line, ● Triple line, Result: The drawing width is modified.

Background Color

The following table shows the procedure for changing the background color of a drawing:

Step	Action
1	Select the drawing to modify.
2	Select the **Background Color** icon. Result: The **Color** window appears.
3	Choose the new background color.
4	Confirm with **OK**.

Line and Border Color

The following table shows the procedure for changing the color of borders and lines in a drawing:

Step	Action
1	Select the drawing to modify.
2	Select the **Line Color** icon. Result: The **Color** window appears.
3	Choose the new line color.
4	Confirm with **OK**.

The Find Function

At a Glance

The **Find** function is used to find the following in the edit and supervision windows:

- A function block, from its comment or name,
- A link, from its name.

Procedure

The following table shows the procedure for using the **Find** function:

Step	Action
1	Select the **Find** command from the **Edit** menu. Result: the **Find** window appears.
2	Enter the string of characters to be found in the **Find** zone.
3	Check the **Find whole word only** box so that the search is carried out only on the string to be found.
4	Check the **Case sensitive** box so that the search takes the case into account (upper and lower case letters).
5	Launch the search by pressing **Next**. Result: ● If the search is successful, the function block is highlighted in the window, ● If the research is not successful, the **No block found** window appears.
6	Relaunch the search by pressing **Next** until the **No other block** window is displayed.

Section 24.2
Manipulating FBD Objects

Subject of this Section

This section describes the manner in which objects in the wiring and supervision sheets should be manipulated: how to select, move, duplicate or delete objects, etc.

What Is in This Section?

This section contains the following topics:

Topic	Page
How to Select Objects	412
How to Create Composite Objects	413
How to Delete and Duplicate Objects	414
How to Position Objects	416
How to Create or Modify a MACRO	418

How to Select Objects

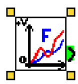
At a Glance

In a wiring or supervision sheet, the function blocks and drawings are objects.

When objects are created, it can be necessary to select certain objects in order to position or group them, etc.

How to Select One or More Objects

The following table describes the operations to carry out in order to select one or more objects.

If you would like to select...	Then
An isolated block.	Click on the block. **Result**: The selected object is highlighted by small yellow squares placed at each corner of the block. 
Several contiguous objects.	Frame the objects to be selected by defining a selection zone. **Result**: The selected objects are highlighted by small yellow squares placed at each corner of the block.
Several non-contiguous objects.	Press and hold the **Shift** key, and click on the objects to be selected. **Result**: The selected objects are highlighted by small yellow squares placed at each corner of the block.

How to Deselect a Block of Selected Objects

The following table describes the operations to carry out in order to deselect a block.

Step	Action
1	Press and hold the **Shift** key.
2	Click the selected block that you would like to deselect. **Result**: The yellow squares associated with the block disappear, showing that the block is no longer a part of the selection.

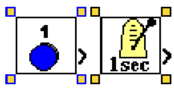
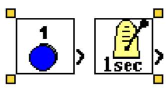
How to Create Composite Objects

At a Glance

The objects in a wiring or supervision sheet can be associated to form a unique composite object. In the same way, it can be necessary to ungroup a composite object into several objects in order to work with them individually.

How to Associate a Group of Objects

The following table describes the operations to carry out when associating a group of objects.

Step	Action
1	Select the objects to associate. **Result**: The selection is highlighted by small yellow squares placed on each element of the selection. 
2	Activate the **Group** command in the **Tools** menu. **Result**: The objects are grouped in a single **composite object**. The resulting object is highlighted by small yellow squares placed at each corner of the object. 

How to Ungroup a Group of Objects

The following table describes the operations to carry out when ungrouping a group of objects.

Step	Action
1	Select the composite object to ungroup. **Result**: The composite object is shown by small yellow squares.
2	Activate the **Ungroup** command in the **Tools** menu. **Result**: The objects contained in the composite object are displayed with their small yellow squares.

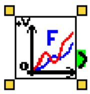
How to Delete and Duplicate Objects

At a Glance

It can be necessary to delete or duplicate an object in the wiring sheet.

How to Delete Objects

The following table describes the operations to carry out in order to delete one or more objects.

Step	Action
1	Select the object(s) to be deleted. **Result**: The selection is highlighted by small yellow squares placed on each corner of the block. 
2	Press the **Delete** or **Backspace** key. **Result**: The selected objects are deleted.

How to Copy Objects Using the Mouse

The following table describes the operations to be carried out in order to copy one or more objects using the mouse.

Step	Action
1	Select the object(s) to be copied.
2	Click on one of the selected objects.
3	Keep the mouse button pressed down and press the **Ctrl** key.
4	Drag the selected object(s) to the chosen spot. **Result**: During the movement, the selection is shown by a dotted zone.
5	Release the mouse button. **Result**: The copy of the selection is positioned at the chosen spot.

How to Cut, Copy or Paste Objects

The following table shows the operations to carry out to cut, copy or paste one or more objects.

Step	Action
1	Select the object(s) to be manipulated. **Result**: The selection is highlighted by small yellow squares placed on each corner of the block.
2	Select the command to execute: ● **Edit → Cut** ● **Edit → Copy** ● **Edit → Paste** **NOTE:** The keyboard shortcuts **Ctrl**+X, **Ctrl**+C and **Ctrl**+V can also be used.

How to Position Objects

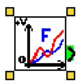
At a Glance

It can be necessary in a wiring or supervision sheet to position an object in relation to another:

- To align objects.
- To center objects.
- To position the objects in the foreground and background in relation to others.

How to Align a Group of Objects

The following table describes the operations to carry out when aligning a group of objects:

Step	Action
1	Select the objects to align. **Result**: The selected objects are highlighted by small yellow squares placed at each corner of the block. 
2	From the **Align** command in the **Tools** menu, select: ● Align left ● Align right ● Align top ● Align bottom **Result**: The selected objects are aligned according to the choice made.

How to Center a Group of Objects

The following table describes the operations to carry out when centering a group of objects:

Step	Action
1	Select the objects to center. **Result**: The selected objects are highlighted by small yellow squares placed at each corner of the block.
2	From the **Align** command in the **Tools** menu, select: ● Center vertically ● Center horizontally **Result**: The selected group of objects is centered.

How to Bring an Object to the Foreground

The following table describes the operations to carry out when bringing an object to the foreground:

Step	Action
1	Select the object to be brought to the foreground. **Result**: The selected object is highlighted by small yellow squares placed at each corner of the block.
2	From the **Order** command in the **Tools** menu, select **Bring to front**. **Result**: The object selected is brought to the foreground.

How to Send an Object to the Background

The following table describes the operations to carry out when sending an object in the background:

Step	Action
1	Select the object to be sent to the background. **Result**: The selected object is highlighted by small yellow squares placed at each corner of the block.
2	From the **Order** command in the **Tools** menu, select **Send to back**. **Result**: The object selected is sent to the background.

How to Create or Modify a MACRO

What is a MACRO?

A MACRO is a group of function blocks. It is characterized by its number, its name, its links, its internal function blocks (255 maximum) and its input/output connections.

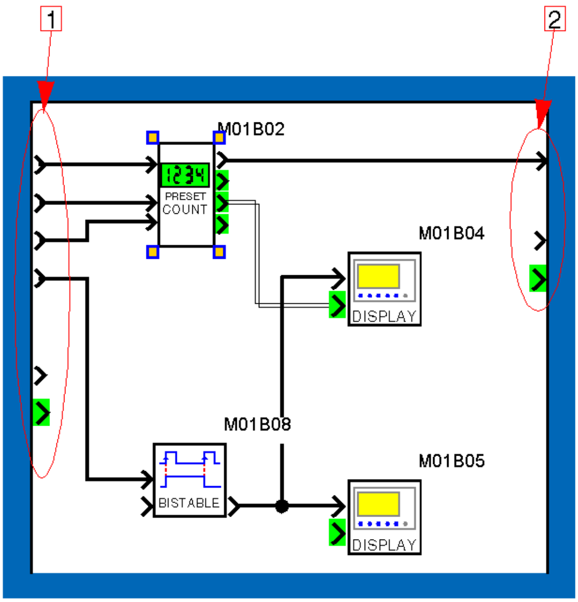
Inside the MACRO:

- Each input connection is attached to one input maximum on the function block.
- Each function block output can be attached to a function block input or an output connection.

From the outside, the MACRO resembles a function block with inputs and/or outputs that may be connected to links *(see page 399)*. However, a MACRO cannot be inserted into another MACRO.

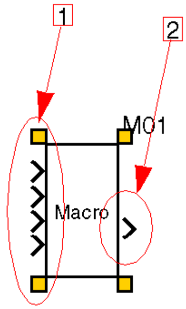
For example:

An inside view of a MACRO:



1 Input connections
2 Output connections

An outside view of the same MACRO in the Edit window.



1 inputs (only current connections are displayed)
2 output (only the current connection is displayed)

Maximum Number of MACROs

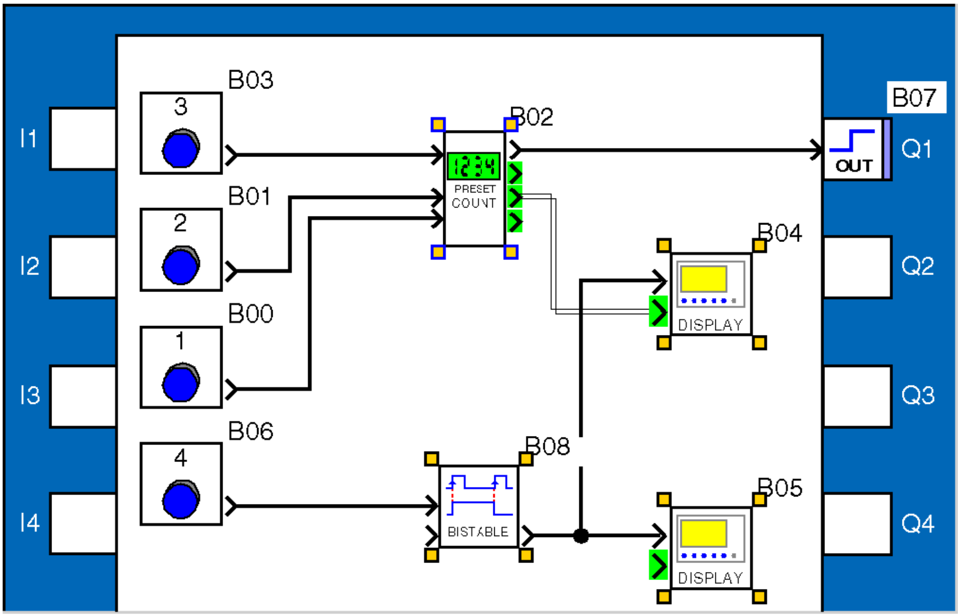
The maximum number of MACROS (including the instances *(see page 422)*) is 64.

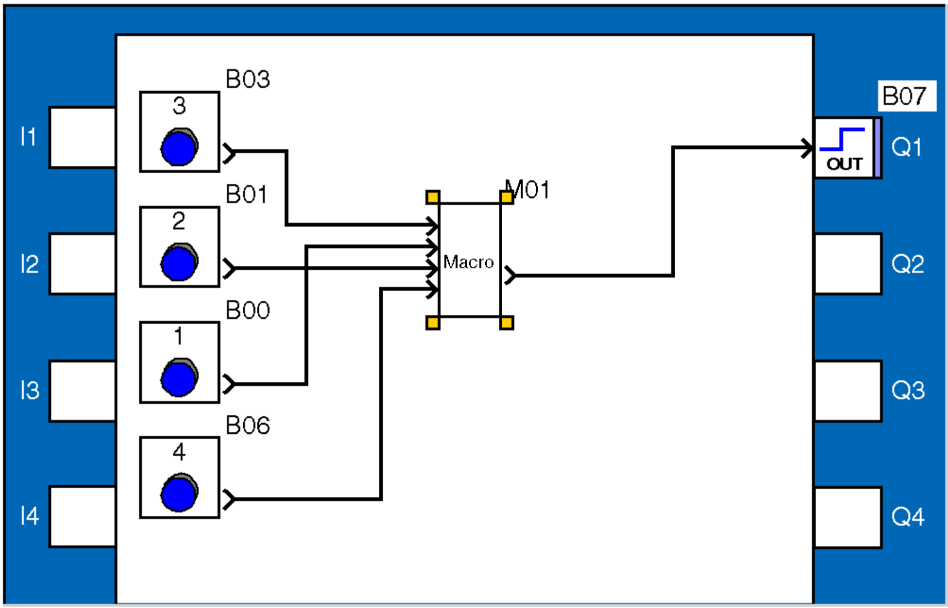
MACRO Backup

A MACRO is backed up when the application to which it belongs is saved. For more details, refer to Saving an Application *(see page 516)*.

How to Create a MACRO

A Macro is created in several steps:

Step	Action
1	Select (see page 412) the function blocks that must appear in the MACRO from the wiring sheet. For example: 
2	Select the **Create a MACRO** command from the contextual menu.
3	Fill in the MACRO Properties (see page 422) dialog box (**MACRO Identifier** is the only mandatory field).

Step	Action
4	Close the dialog box by clicking on **OK**. **Result**: The function blocks selected in step 1 are then represented by a single block (MACRO box) in the edit window. For example: 

Handling a MACRO

Once created, a MACRO may be handled like a function block and in particular it may be:
- Selected *(see page 412)*.
- Associated with other objects *(see page 413)*.
- Copied in the Edit window *(see page 415)*.
- Copied as import *(see page 512)* in the Edit window.
- Copied/Pasted between two programs *(see page 415)*.
- Deleted *(see page 414)*.

MACRO Instances

A MACRO copy is a new instance of the original MACRO.

Modifications made to the graph or properties *(see page 422)* of an instance are automatically applied to other instances of the MACRO. They are equivalent to recompiling the MACRO. On the other hand, the modifications made to comments or parameters of internal function blocks belong to each instance of the MACRO. You can therefore have two instances of the same MACRO with different parameters.

If the last instance of a MACRO is cut or deleted, a message will appear. The operation can then be canceled.

MACRO Properties Dialog Box

The Macro Properties dialog box is used to enter or modify the properties of a MACRO. If the MACRO has been copied, the modifications will be made to all the instances of the MACRO *(see page 422)*.

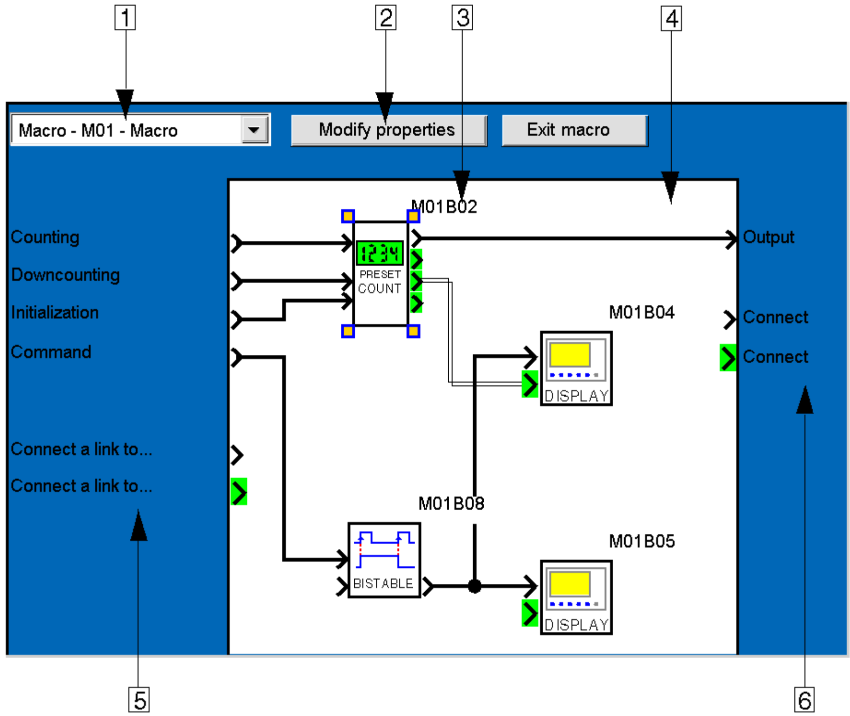
The dialog box can be accessed during MACRO creation or in the contextual menu by selecting **Display the MACRO** then clicking the **Modify Properties** button.

The different parameters of a MACRO are as follows:

- **MACRO Identifier** (1 to 5 characters).
- **Name of the MACRO** (optional).
- **Block symbol**, which is the appearance of the block representing the MACRO in the main wiring sheet, and which may be either:
  - A **Standard Image** (The identifier of the MACRO will be used as the block symbol).
    or
  - A **Custom Image** (To insert an image, click on the button).
- **Name of inputs** (if required, you may modify the input label in the **Label** box of the table.)
- **Name of outputs** (if required, you may modify the label of the output in the **Label** box of the table.)

The MACRO Window

MACROS may be modified in the **MACRO** window accessible from the **Windows** menu (except when protected by a password. For more details, refer to Password Protection *(see page 426)*.



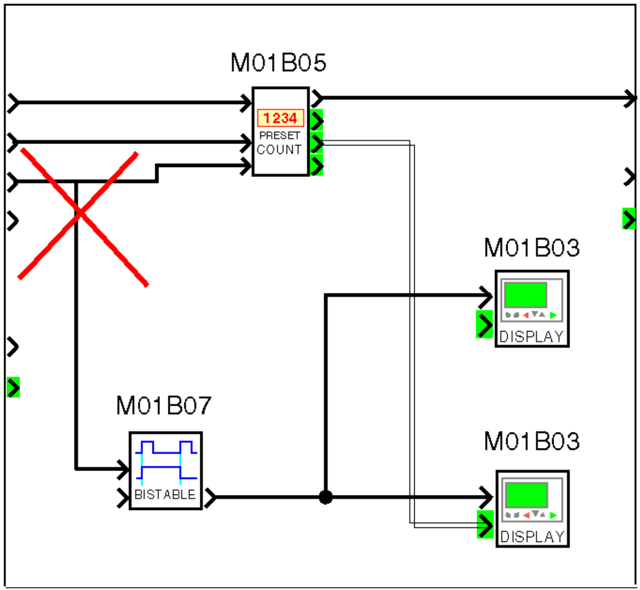
The following table lists the different elements of the **MACRO** window.

Element	Function
1: drop-down list	To select the MACRO from among all the MACROs of the project and the different instances *(see page 422)*.
2: **Modify properties** button	To access the MACRO Properties *(see page 422)* dialog box.
3: MACRO internal function block	Double click on the internal function block to gain access to its parameters. (If the MACRO has been copied, the parameter modifications will be made only to the instance *(see page 422)* of the MACRO in operation.

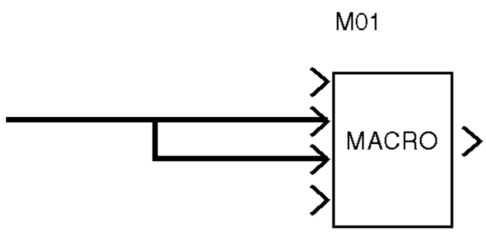
Element	Function
4: MACRO wiring sheet	To modify the MACRO graphics and in particular: ● Add or delete a link between two function blocks. ● Add a function block from the function bar or the edit window. ● Delete a function block. (If the MACRO has been copied, these modifications will be made to all the instances of the MACRO *(see page 422)*. Two input connections cannot be attached to the same function block input.
5: input not connected	To create a new link to a function block input of the wiring sheet. An additional input of the MACRO shall then appear in the edit window. (If the MACRO has been copied, these modifications will be made to all the instances of the MACRO *(see page 422)*.
6: output not connected	To create a new link from the function block output of the wiring sheet. An additional output of the MACRO will then appear in the edit window. (If the MACRO has been copied, these modifications will be made to all the instances of the MACRO *(see page 422)*.

Input Connections

An input connection cannot be attached to two different function block inputs in the same MACRO.



Instead, make these connections outside the MACRO, as indicated in the diagram below:



How to Modify the Graph for a MACRO Instance While Keeping the Other Instances

When the graph for a single MACRO instance is modified, a new MACRO is created. To modify the graph for a MACRO, proceed as follows:

Step	Action
1	Right click on the mouse to select the MACRO instance.
2	Select **Display the MACRO** in the contextual menu.
3	Select **Edit** → **Select all** menu.
4	Select **Edit** → **Copy** menu.
5	Click on the  button to return to the main wiring sheet.
6	Select **Edit** → **Paste** menu.
7	Place the selection in the required position.
8	Select **Create a MACRO** in the contextual menu.
9	Fill in the MACRO Properties *(see page 422)* dialog box (**MACRO Identifier** is the only mandatory field).
10	Close the dialog box by clicking on **OK**.
11	Select **Display the MACRO** in the contextual menu.
12	Use the MACRO window *(see page 423)* to create the input/output connections and make modifications.

How to Modify MACRO Comments

To modify MACRO comments, proceed as follows:

Step	Action
1	Double click on MACRO.
2	Modify the comment.
3	Confirm by clicking on **OK**.

Password Protection

A password can be used to protect the MACROS of a project. It is independent of the application password. It consists of 4 numbers (0000 is not a valid password).

This protection is defined in the program configuration window which can be accessed by clicking on the  button or through **Edit** → **Program Configuration** menu, **Configuration** tab.

The same password protects all MACROS of the project. It is requested when the project is opened.

If the password is not entered when the project is opened the following functions are not available:
- Access the MACRO window
- Copy MACRO
- Print MACROS

Front Panel Display

In the main menu on the front panel of the smart relay, select Parameters by pressing on the **Menu/Ok** button. The first function block with its number will be displayed. You may switch from one function block/MACRO to another by using the up and down navigation keys.

The numbering rules are as follows:

Number	Meaning
R00B•	Function block
R01B	Macro number 1
R02B	Macro number 2

NOTE: Function blocks are listed first.

Section 24.3
Debugging and Monitoring an FBD Application in the Programming Software

Subject of this Section

This section describes the different functions linked to debugging the application in the Programming Software in FBD mode.

What Is in This Section?

This section contains the following topics:

Topic	Page
Simulation Mode	429
Monitoring Mode	434
Modification and Forcing in Simulation and Monitoring Mode	436

Simulation Mode

At a Glance

The Simulation mode of an FBD program allows you to debug the program by simulating its execution on the host computer. In this mode you can perform the following actions from the edit and supervision windows and from the front panel:

- View the states of function block outputs.
- View and modify function block parameters.
- Force the state of function block inputs and outputs.
- Modify the state of the buttons on the front panel.
- Force the state of function block links.

In Edit mode the different windows are updated together. For example, if a function block is placed in the edit and supervision window. When an action is performed on this function block from the edit window it is also updated in the supervision window.

For more information, refer to How to debug an application without loading it onto the smart relay: Simulation *(see page 37)*.

Access to Simulation Mode

Simulation is accessed by the **Mode** → **Simulation** menu or by using the  icon.

NOTE: By default the edit window is displayed full screen, and the front panel and supervision windows can be accessed:
- From the Window menu.
- By minimizing the wiring window.

Program Execution Parameters

NOTE: To display the functions described below, check the box in the **File** → **Preferences** menu: display the refresh period and the number of cycles for monitoring and simulation.

For more information, refer to How to debug an application without loading it onto the smart relay: Simulation *(see page 41)*.

Refresh Period

This is the duration with which the I/O values and parameters are updated in the application windows.

Program Commands

Description of program command buttons in simulation mode:

Active button	Description
	Launches program execution.
	Stops program execution.
	Pause/Run: stops or relaunches the program flow. (only activated in **RUN** mode)
	Simulation of a power outage *(see page 68)*. (only activated in **RUN** mode).

The color of the icons changes according the application state.

When it is possible to select the icon it is shown in yellow 

.

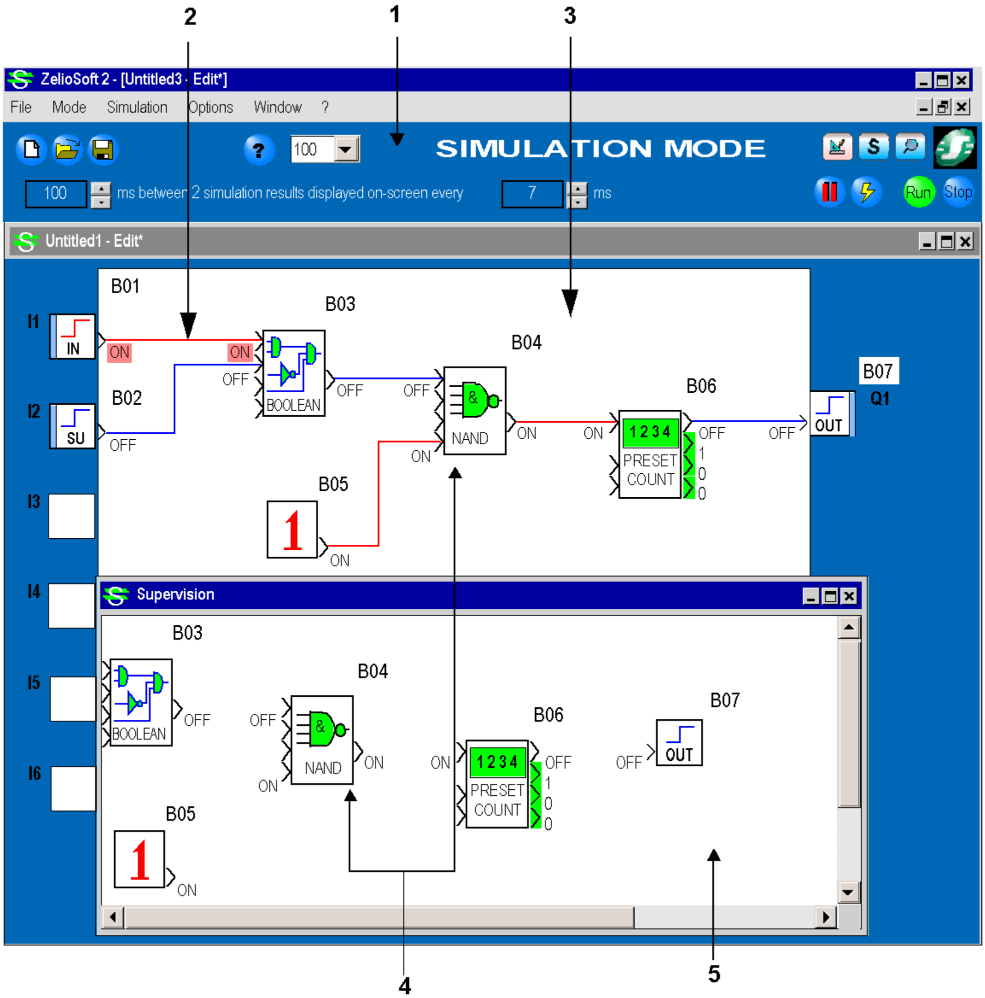
Unavailable Functions

In simulation mode, the following functions are not available:

- Graphic editing of programs.
- Transfer program.
- Clear the program.
- Compare program.
- Switch to Monitoring mode with smart relay connection.
- Modify communication parameters.

Diagram

The following figure shows an example of edit and supervision windows in simulation mode:



Description of Elements

The following table lists the different elements of the edit window:

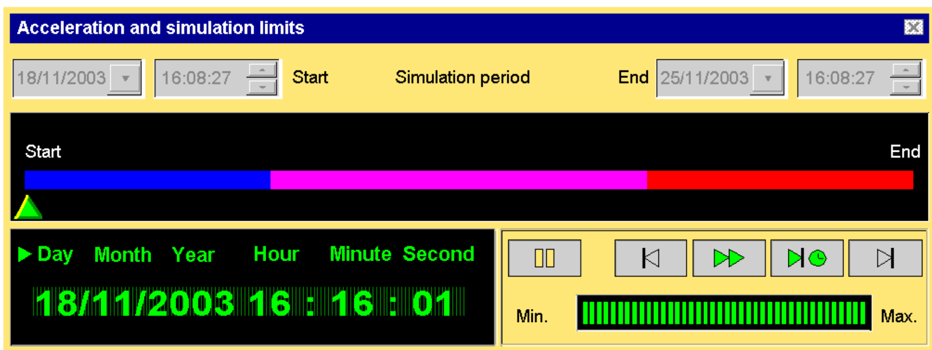
Number	Description
1	Simulation bar: used to modify the number of cycles executed on each simulation step.
2	Link in active state: the color can be configured *(see page 395)* according to state. Active (ON) or Inactive (OFF) state is specified to each side of the link.
3	Simulated wiring sheet.
4	The same function block with animated inputs/outputs and parameters in the edit and supervision windows.
5	Simulated supervision window.

Accelerator

The  icon is used to display or hide the accelerator box.

NOTE: This functionality has an influence on operation of the TIME PROG *(see page 314)* function.

Illustration:



Description of Simulation Window:

- Entry and display of simulation period.
- Cursor allowing to change the time by moving the cursor (Stop mode only).
- Display of date and time in simulation.
- Video-type control panel: Pause, Return to Start (Stop mode only), Fast Forward, Jump to Next `TIME PROG` Function Event, End, Time Acceleration Period Adjustment.

This window is displayed when you click on the simulation time smart relay icon situated in the bar at the bottom of the simulation window.

Display:

- Displays the date and time of the start and end of the simulation.

Actions:

- Can be used to modify the date and time of the start and end of the simulation (in **Stop** mode) in the **Acceleration and simulation limits** window.
- Can be used to accelerate the simulation speed up to 65,000 times the original speed by clicking the  button and moving the level of the **Min - Max** bar.
- Can be used to place the cursor 3 s before the `TIME PROG` function event by clicking on the  button.

Simulation of a power outage

During a power outage simulation, the simulator clock is frozen. To simulate power return at a given time, the simulation time must be changed. To do this, proceed as follows:

Step	Action
1	Click on the **Simulation → Set Clock**.
2	Enter the desired date and time for power return in the **Date** and **Time** fields of the Set Clock window.
3	Confirm by clicking on the **Write in the module** button.

Monitoring Mode

At a Glance

In monitoring mode, the smart relay is linked to the host computer. In this mode you can perform the following actions from the edit and supervision windows and from the front panel:

- View the states of function block outputs.
- View and modify function block parameters.
- Force the state of function blocks inputs and outputs (maximum of 10 function block outputs simultaneously).
- Modify the state of the buttons on the front panel.
- Force the state of function block links.

Monitoring mode can be accessed from the **Mode → Monitoring** menu.

In monitoring mode, the different windows are updated at each cycle. For example, if a function block is placed in the edit and supervision window. When an action is performed on this function block from the edit window it is also updated in the supervision window.

For more details, refer to How to monitor and modify an application running on the smart relay from the Programming Software: monitoring *(see page 43)*.

Unavailable Functions

In monitoring mode, the following functions are not available:

- Graphic editing of programs.
- Transfer program.
- Clear program.
- Compare program.
- Switch to Simulation mode.
- Modify communication parameters.

Access to Monitoring Mode

Monitoring is accessed by the **Mode → Monitoring** menu or by using the  icon.

The following scenarios may arise:

- An application is open in the software: The version present on the smart relay is compared to that of software:
  - If the software application is the same as the one on the smart relay, monitoring mode is started.
  - If the software application is different from the one on the smart relay, the versions must be synchronized by transferring the program from the PC to the smart relay or from the smart relay to the PC.
- No application is open in the software: In this case, the software offers to transfer the application from the smart relay to the PC.
  Once the transfer is complete, the supervision window is displayed.

Diagram

The program states in the application windows are represented the same way as those in Simulation *(see page 431)* mode.

Modification and Forcing in Simulation and Monitoring Mode

At a Glance

In simulation mode, you can modify the parameters of the function blocks and force the function block input and output states.

- Force the state of function block links.
- Force the state of function block inputs and outputs.
- Modify function block parameters.
- Modify the state of the buttons in the front panel of the smart relay.

In monitoring mode, you can modify the parameters of the function blocks and force the link states.

- Force the state of function block links.
- Modify function block parameters.
- Modify the state of the buttons in the front panel of the smart relay.

There are two forcing modes:

- Momentary mode.
- Permanent mode.

Forced values are highlighted by a change in color *(see page 395)* according to the state.

It is possible to force certain states from the Programming Software and to display the program states (maximum of 10 function block outputs simultaneously).

Momentary Mode

Click on the link between the two blocks to modify the state. This forcing works only during the mouse click.

Permanent Mode

Function block discrete and analog inputs and outputs can be permanently forced.

The following table shows the procedure for permanently forcing a discrete selection:

Step	Action
1	Right-click on the link or on the function block input or output. Result: the contextual menu appears.
2	Select the **Force and maintain** command. Result: the **Permanent forcing** window appears.
3	Select: ● ON/Active to switch the selection from inactive to active state. ● OFF/Inactive to switch the selection from active to inactive state.
4	Confirm with **OK**. Result: the selection changes color and displays ON for Active and OFF for inactive.

The following table shows the procedure for permanently forcing an Analog selection:

Step	Action
1	Right-click on the link or on the function block input or output. Result: the contextual menu appears.
2	Select the **Force and maintain** command. Result: the **Analog value** window appears.
3	Enter the analog forcing value.
4	Confirm with **OK**. Result: the selection changes state.

Parameter Modification

If a function block has parameters, you can modify them. The following table shows the procedure to follow:

Step	Action
1	In the edit or supervision window, double-click on the function block to modify.
2	Select the **Parameters** tab.
3	Modify the required parameter(s).
4	Confirm the modifications with **OK**. Result: The new parameters appear next to the function block in the edit and/or supervision window.

Front Panel of the Module

The buttons on the smart relay front panel can be controlled from the application windows as if they were those on the actual front panel of the smart relay. Click on the button to change its state.

Chapter 25
Example of an FBD Application

Greenhouse Ventilation Panes

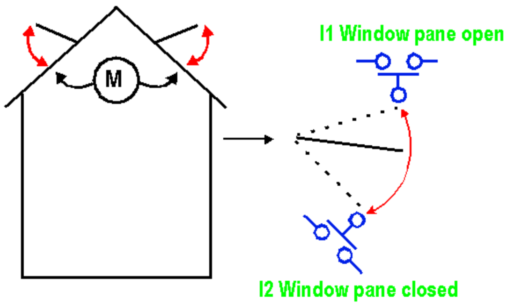
Description

This example describes how greenhouse ventilation panes can be managed automatically.

Specifications

The owner of a greenhouse would like to acquire an installation to manage the opening and closing of the ventilation window panes located on the greenhouse roof.

The greenhouse has two window panes to provide ventilation. The opening of these window panes is controlled by a motor and 2 sensors that indicate whether the window panes are open or closed:



During the day, the window panes open to ventilate the structure from 12:00 to 15:00, at the time of day when, in principle, the temperature is the highest. However, if the temperature is less than 10 °C, the window panes do not open, or when they are already open, they close.

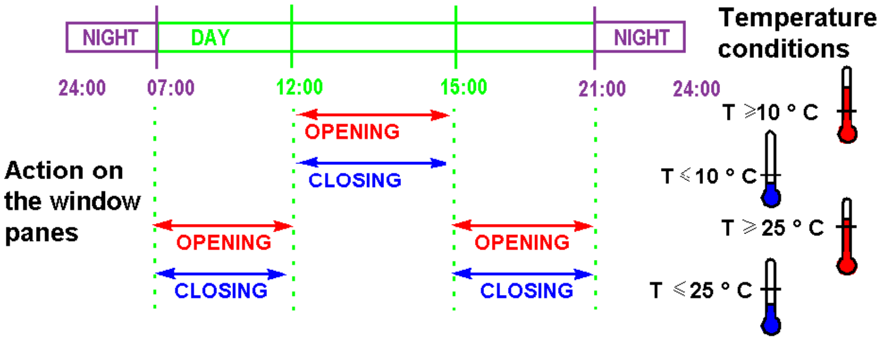
In addition, the window panes open during the day when the temperature reaches 25 °C. If the temperature falls below 25 °C, the window panes must close again.

Finally, at night, the window panes remain closed regardless of the temperature.

Program description, 3 time ranges are used:

- Range 1: Night, from 21:00 to 07:00
- Range 2: Day, from 07:00 to 12:00 and from 15:00 to 21:00
- Range 3: Noon, from 12:00 to 15:00

Summary:



Input/Output Table

Description of the inputs:

Input	Description
I1	Window panes open (Discrete)
I2	Window panes closed (Discrete)
IB	Temperature (analog)

Description of the outputs:

Output	Description
Q1	Opening of the window panes (Discrete)
Q2	Closing of the window panes (Discrete)

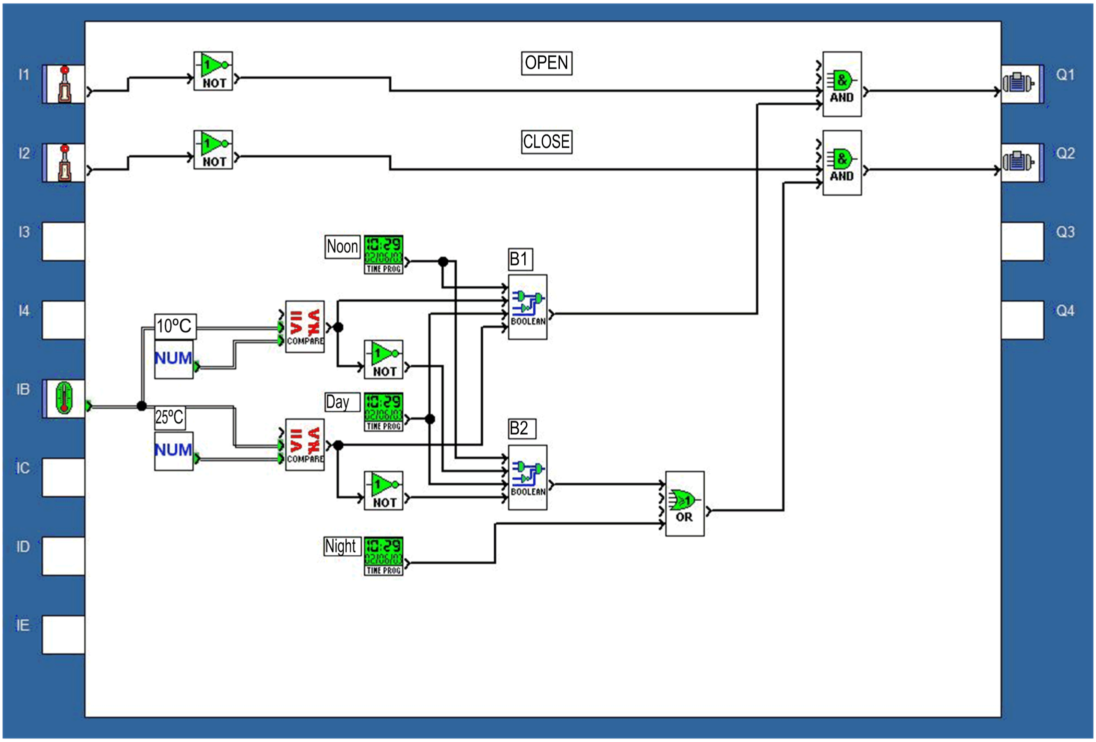
The temperature is supplied by a sensor with output voltage of 0 to 10 V.

Required Reference

For this application, a smart relay with a clock and analog inputs is required:

- SR2B122BD (24 Vdc),
- SR2B121JD (12 Vdc).

FBD wiring sheet



Parameters

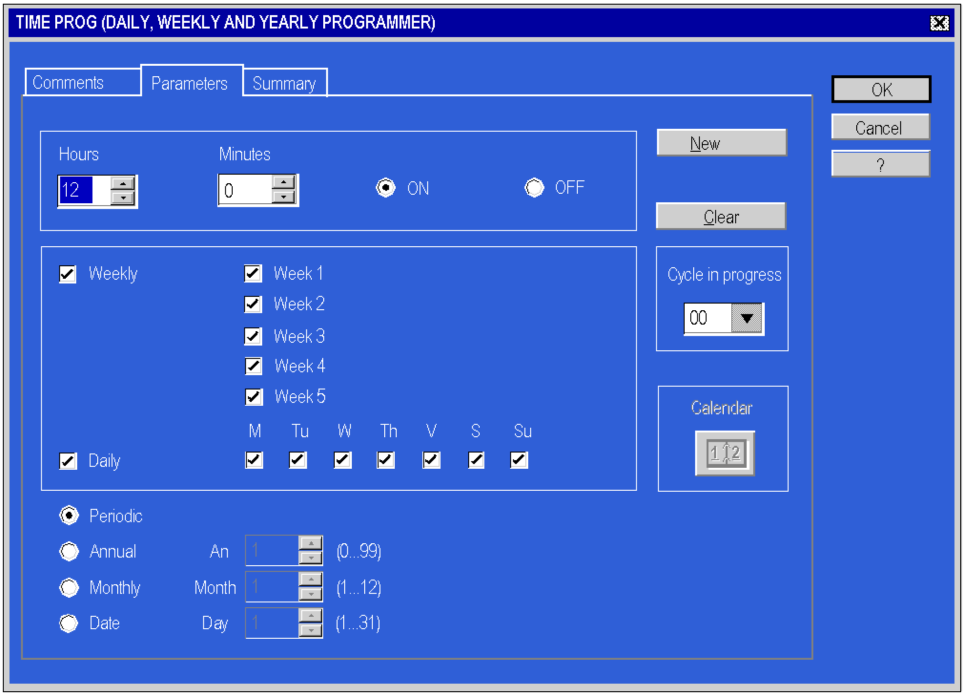
Analog comparator B12

Value1 > Value2

Analog comparator B18

Value1 > Value2

Daily programmer B11



Cycle in progress: 00

- Hour: 15,
- Minute: 00,
- OFF is selected,
- The other parameters are the same as for ON.

Daily programmer B13

Cycle in progress 00

- Hour: 07,
- Minute: 00,
- ON is selected,
- The other parameters are the same as for programmer B11.

Cycle in progress 01

- Hour: 12,
- Minute: 00,
- OFF is selected,
- The other parameters are the same as for programmer B11.

Cycle in progress 02

- Hour: 15,
- Minute: 00,
- ON is selected,
- The other parameters are the same as for programmer B11.

Cycle in progress 03

- Hour: 21,
- Minute: 00,
- OFF is selected,
- The other parameters are the same as for programmer B11.

Daily programmer B19

Cycle in progress 00

- Hour: 21,
- Minute: 00,
- ON is selected,
- The other parameters are the same as for programmer B11.

Cycle in progress 01

- Hour: 7,
- Minute: 00,
- OFF is selected,
- The other parameters are the same as for programmer B11.

Boolean functions

Option

- ○ Output OFF if result is TRUE
- ◉ Output ON if result is TRUE

INPUT 1	INPUT 2	INPUT 3	INPUT 4	OUTPUT
0	0	0	0	0
1	0	0	0	0
0	1	0	0	0
1	1	0	0	1
0	0	1	0	0
1	0	1	0	0
0	1	1	0	0
1	1	1	0	1
0	0	0	1	0
1	0	0	1	0
0	1	0	1	0
1	1	0	1	1
0	0	1	1	1
1	0	1	1	1
0	1	1	1	1
1	1	1	1	1

Part VI
Connection With the Smart Relay

Subject of this Section

This section describes the functions and settings related to connections with the smart relay.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
26	Connection with the Programming Software	447
27	Communication Via Modbus Extension	463
28	SR2COM01 Communication Interface	467
29	Analog Input-Output Extension SR3XT43BD	485
30	Communication Via Ethernet Extension	487

Chapter 26
Connection with the Programming Software

Subject of this Chapter

This chapter describes the functions related to connection of the smart relay to the Programming Software.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Communication Setup Between the Programming Software and the Smart Relay	448
Transfer the PC Program to the Smart Relay	450
Transfer the Smart Relay Program to the PC	452
RUN/STOP Program Run Commands	454
Compare the Smart Relay Data with the Program	455
Smart Relay Diagnostics	456
Access/Modification Protection of the Program Saved on the Smart Relay	457
Clear the Program Contained in the Smart Relay	458
Set Smart Relay Clock	459
Configuring the Smart Relay Language	460
Update Smart Relay Firmware	461
Remote Control of Front Panel	462

Communication Setup Between the Programming Software and the Smart Relay

Description

To establish communication between the Programming Software and the smart relay, one of the following links may be used:

- **Serial link**: Com port
- **Bluetooth link**: Com port
- **USB link**
- **Modem link** (only for smart relays to which the SR2COM01 communication interface has been added).

Prerequired Configurations

Before launching the connection between the Programming Software and the smart relay, the following must be checked:

In the case of...	Make sure that:
Serial or USB links	• The smart relay is physically connected to the Programming Software (PC). • The connection is correctly configured.
bluetooth link	• The bluetooth adapter and its driver are installed on the PC. **NOTE:** The driver associates a com port to the adapter. • Ensure the Bluetooth interface is connected to the smart relay.

Access

The **COMMUNICATION Setup** function can be accessed from the **Transfer** menu.

Communication Configuration

Procedure for configuring the communication:

Step	Action
1	Open the **COMMUNICATION Setup** window from the **Transfer** menu.
2	Choose the type of link: ● **Modem**: Specify: ❍ The modem of the PC to use for communication, ❍ The name of the remote station to which the Programming Software should connect. ● **Com Port**: Specify the port to use: ❍ COM1, ❍ COM2, ❍ USB.
3	Confirm the modifications by clicking **OK**.

NOTE: In specific situations, you must enter information in the COMx window (where x> 2):
- On portable (laptop) computers with no serial link.
- On portable (laptop) computers with USB.
- When a USB-SERIAL converter is used (Win XP provides a COM6 or COM8 for this new peripheral).
- When a bluetooth link is used (specify the com port that the adapter driver has associated to the adapter).

Test the Connection

It is possible to test the connection parameters at the **COMMUNICATION Setup** window level using the **Test** button.

In this case, the Programming Software attempts to connect to the smart relay using the configured parameters. If the connection configuration is not correct, an error message indicates that the device is not responding.

Transfer the PC Program to the Smart Relay

Description

The **Transfer Program** function translates the program developed with the software into data that can be loaded into the smart relay and transfers it from the PC to the smart relay.

This command opens the window **Compilation Results**. The result of the compilation can be:

- **Compilation successful**: The application is transferred to the smart relay.
- **Failed**: The error code appears, the program must be edited to correct the error and the transfer command launched again.

The transfer is only possible if the smart relay:

- Is not blocked by an incorrect password.
- Is stopped.

The program will be written on the smart relay only in the following cases:

- The smart relay does not contain a program.
- The smart relay contains a program that is not read/write protected with a password.
- The smart relay contains a program that is read/write protected with a password, and the password is correctly entered.
  (In this case, the **Password** dialog box appears).

If all conditions are met, the Write options *(see page 510)* dialog box appears.

NOTE: Only an FBD program that has been compiled without any error will be written to the smart relay. All compiled LD programs will be written to the smart relay.

NOTE: The type of smart relay declared in the program must be compatible with the smart relay connected:

- Smart relay version level.
- Version firmware.
- Firmware build number less than or equal to that of the smart relay.
- Same extension.
- Same hardware version and same firmware version as the extension.

NOTE: The firmware can be implicitly updated when an LD (or FBD) program is transferred to a smart relay containing different FBD (or LD) firmware.
The firmware can only be updated if the firmware loaded has been designed for the same smart relay:

- Same hardware version on the smart relay.
- Same boot version and a boot build number less than or equal to the boot of the smart relay to be loaded.

Access

This function is can be accessed from **Transfer → Transfer Program → PC > Module**.

Firmware Error

If a communication error appears during the automatic update of the firmware (3 attempts), then you can try loading the firmware from the menu **Module → Update module FIRMWARE**.

For details about compatibility, refer to Compatibility between the version of the programming software and the version of the firmware on the smart relay *(see page 538)*.

If the firmware update is unsuccessful, verify that the serial link communication is functioning correctly.

Procedure

Procedure for transferring the program to the smart relay:

Step	Action
1	From the **Transfer** menu, click **Transfer Program → PC → Module**. **Result**: The program verification is launched and the **Compilation Results** window opens.
2	Depending on the results of the verification: ● **Compilation successful**: Confirm with **OK**. **Result**: The **Write Options** dialog box appears. ● **Failed**: Correct the errors and then go back to **step 1.**
3	Select the **Write Options** *(see page 510)*: ● Protect reading and modification of the program with a password. Refer to Access/Modification protection of the program saved on the smart relay *(see page 457)*. ● Save modifications before writing. ● Start monitoring mode and switch on the smart relay.
4	Confirm your changes by clicking **OK**. **Result**: The **Write Options** dialog box disappears.
5	Launch the transfer by clicking **OK** in the **Transfer Program → PC → Module** dialog box.

NOTE: When using the SR2COM01 communication interface, in order for the link between the smart relay and the modem to work, you must:
- Restart the communication interface after each change of link type (link with the PC, link with the modem).
- Wait for the end of the initialization cycle of the communication interface.

Transfer the Smart Relay Program to the PC

Description

The function for transferring the application from the smart relay to the PC translates the data contained in the smart relay in order to restore a program that can be edited in the Programming Software.

The Programming Software can read the contents if the smart relay:

- contains a program that is not read/write protected with a password, or
- contains a program that is read/write protected with a password, and the password is correctly entered.
  (In this case, the **Password** dialog box appears).

The data retrieved by reading contains references to the application during its transfer:

- The name of the application file.
- The access path: relative to the (**File/Preferences**) work directory.

NOTE: the access path is limited to a maximum of 128 characters (program name with extension included).
If this limit is exceeded (only the file name and its extension are saved), then a window is displayed to complete the access path.

Access

This function is can be accessed from **Transfer → Transfer Program → Module > PC**.

Restoring the Program

Using the information concerning the application present on the smart relay (name of the source file and location on the PC), the Zelio Logic2 software tries to reload the application file from the PC.

The aim of this search is to retrieve the graphic representations:

- Positions related to the function blocks
- Positions of links between functions
- Comments
- Screen backgrounds
- Drawings

NOTE: modifications may have been made after the write from the application to the smart relay:
- In the Programming Software: The application has been modified.
- In the smart relay: Modification of the parameters using the front panel.

In the case where differences in parameters appear, a dialog box offers to update the Programming Software with the parameters read on the smart relay.

There are certain cases where the program cannot be retrieved:

- The program differences appear between the file containing the program on the PC and the application read on the smart relay.
- The file containing the program on the PC is not accessible.
  To reread the original application saved on the PC, use the path (128 characters) in the configuration of the application loaded on the smart relay or in another directory on the PC. If this is not possible, the reconstruction is made by default, with access to page setup and comments.

In these circumstances, the **Program construction** window opens and suggests an alternative procedure:

- **Construction using the file specified by the user**: enter the file path of the application to be retrieved.
- **Automatic construction of the program**: In this case, the Programming Software regenerates the application from the data retrieved on the smart relay.

NOTE: The program loaded into the smart relay does not contain information concerning page setup (drawing, comment, relative position of the function blocks and links); a default page setup is thus produced.

NOTE: All of the function parameters are retrieved.

RUN/STOP Program Run Commands

Description

These commands can be used to remotely control a smart relay connected to the PC. Once the connection has been made, control can be carried out using the front panel window, with which you can interact as if it was the front panel of the smart relay.

This function is used to start and stop the program in the smart relay:

- **RUN Module**: The values (counters, timers, etc.) are reset to zero before the program starts.
- **RUN Module Without Init of saved data**: The values for which the **Latching** option has been activated are kept.
- **STOP**: The program is no longer executed and the outputs are disabled.

NOTE: The outputs of an extension SR3XT43BD are not disabled if an acceleration time is specified in the extension parameters *(see page 485)*.

Access

The **RUN Module**, **Run Module Without Init of saved data** and **STOP Module** commands can be accessed from the **Transfer** menu.

Module Status Upon Power Outage

In the event of a power outage, the program is immediately stopped, parameters of the type **initialization on power break** or **latching on power break** (Latching) are saved.

(For more details, refer to How the Smart Relay Behaves in the Event of Power Outage *(see page 68)*.

An interruption in the link between the PC and the smart relay is indicated in the Zelio Soft 2 Programming Software by an error message (if the software is in Monitoring mode, it switches to edit mode).

When power is restored, the smart relay executes a **RUN** command, initializing the non-saved data.

Module Status on Blocking Error

If the event of a blocking error (disruption in the link between the smart relay and its extensions), the smart relay places itself in **Stop** mode.

The detected error is displayed on the front panel of the smart relay.

To restart the smart relay, having removed the cause of the blockage, use the **RUN** command.

For more details, refer to Meaning of the error code displayed on the controller front panel *(see page 49)*.

Compare the Smart Relay Data with the Program

Description

This function compares the data contained in the smart relay and the data produced by compiling the Programming Software application.

If the smart relay data is protected by a password, enter it in the **Password** window.

The comparison is carried out on the program (including parameters) contained:

- In the smart relay.
- In the Programming Software edit window on the PC.

Access

The **Compare the program with module data** function can be accessed from the **Transfer** menu.

Smart Relay Diagnostics

Description

The diagnostics function allows you to view the characteristics of the smart relay to which the Programming Software is connected.

The smart relay Diagnostics dialog window can only be accessed if the smart relay is connected to the PC.

The diagnostics window is made up of two tabs:

- **Hardware**: Characteristics of the smart relay (hardware and firmware).
- **Application**: Characteristics of the application (program) of the smart relay.

Access

The **Module diagnostics** function can be accessed from the **Module** menu.

Hardware

The hardware tab provides the following information:

- The smart relay type and version/release of the hardware and firmware.
- Numbers and types of smart relay inputs and outputs.
- Connected extension(s) and version(s)/release(s), only for the extendable smart relays.
- Smart relay status (Run, Stop, Blocked in Error, Advisory).
- Smart relay language.
- Error code (No error, Binary error, Communication error, Target Error or Advisory).

NOTE: The hardware-related information is accessible regardless of whether the program is protected by a password or not.

Application

The application tab provides the following information:

- The name of the program, its author, and version.
- Used or maximum memory capacity.
- Its configuration parameters: Cycle time duration, Z key locking, watchdog action, password, input filtering.
- For LD mode only, the number of LD lines used/available and the number of each function used in the program.

NOTE: The information related to the application is only available if the smart relay contains a program that is not password-protected or if the password has been correctly entered.

Access/Modification Protection of the Program Saved on the Smart Relay

Description

The option for protecting the program transferred to the smart relay can be activated at the end of the procedure for transferring the PC program to the smart relay *(see page 451)*.

The protection is activated in the **Write Options** dialog box that contains the parameter: **Protect reading and modification of the program with a password**: If this option is validated, the password data entry zones are activated.

NOTE:

- After 5 unsuccessful tries, the smart relay is locked for a duration of 30 minutes.
- When the application is protected by a password, it cannot be modified but it can still be overwritten.

Clear the Program Contained in the Smart Relay

Description

The **Clear the Program** function can be used to erase the application loaded on the smart relay, as well as related information (password), but does not affect the smart relay firmware.

This operation can be used for deleting a program whose password you have forgotten.

NOTE: The program clear command is still valid, even if the smart relay is protected by a password.

Access

The **Clear the Program** function can be accessed from the **Transfer** menu.

Set Smart Relay Clock

Description

The **Set Clock** function is used to set the date and time. It is divided into two zones:

- **Date** zone
- **Time** zone

Access

The **Set Clock** function can be accessed from the **Module** menu in Edit mode or from the **Simulation** menu in simulation mode.

Adjustments

The date is configured using the field in the **Date** zone.

The **Time** zone is used to configure the:

- Hours
- Minutes
- Seconds
- Drift value of the smart relay clock (in seconds per week).

Procedure

Smart Relay clock configuration procedure:

Step	Action
1	Open the **Set Clock** window from the **Module** menu.
2	Enter the date and time parameters.
3	Confirm the changes by clicking **Write in the Module**. **Result**: The Programming Software sends the values to the smart relay.

Configuring the Smart Relay Language

Description

This function is used to change the smart relay interface language.

The messages can be displayed in 6 languages:

- English
- French
- German
- Italian
- Spanish
- Portuguese

Access

The **Smart Relay Language** function can be accessed from the **Module** menu.

Procedure

Procedure for updating the smart relay language:

Step	Action
1	Open the **Smart Relay Language** window from the **Module** menu.
2	Select the language from the drop-down menu.
3	Confirm the transfer by clicking **Write in the Module**. **Result:** The Programming Software sends the new value to the smart relay.

Update Smart Relay Firmware

Description

This command allows you to load the firmware (specific to the smart relay) onto the smart relay. It can be used to select the operating mode of the program: FBD/LD mode or to load a different version/release of firmware.

This triggers clearing of the program that was loaded into the smart relay, as well as the smart relay configuration parameters.

This operation can be used to delete a program if you have forgotten the password.

NOTE: The firmware is implicitly updated when an LD (or FBD) program is transferred to a smart relay containing different FBD (or LD) firmware.

Access

The **Update module FIRMWARE** function can be accessed from the **Module** menu.

Procedure

Procedure for updating the smart relay firmware:

Step	Action
1	Open the **Update module FIRMWARE** window from the **Module** menu.
2	Select the firmware to be downloaded using the **Browse** button.
3	Confirm the transfer by clicking **Write in the Module**. **Result:** The Programming Software sends the new firmware to the smart relay.

Remote Control of Front Panel

Description

This function remotely emulates the smart relay front panel.

The **Remote Control of Front Panel → RUN** submenu starts the smart relay front panel simulation.

The **Remote Control of Front Panel → STOP** submenu stops the smart relay front panel simulation.

NOTE: The function is also available for smart relays without any front panel.

Access

The **Remote Control of Front Panel** can be accessed from the **Transfer** menu.

Chapter 27
Communication Via Modbus Extension

Communication via the Modbus Extension

Description

The Modbus protocol is a **master/slave** protocol that allows one, and only one master to request responses from slaves, or to act based on the request.

To use Modbus functions, an **SR3MBU01BD** extension module must be added onto a Zelio 2 **SR3BxxxBD** smart relay.

Modbus communication is possible in the following modes:

- LD *(see page 149)*,
- FBD *(see page 267)*.

NOTE: The Modbus Zelio 2 module only operates in Modbus slave mode.

Functional Description

The Modbus Zelio 2 module has the following characteristics:

- Connection on a Modbus network: 2 or 4-wire,
- Maximum length of the network: 1,000 meters (9600 bauds);)
- Line terminated at each of the 2 ends (Line terminators: 1mF, 10V, 12 ohms, 0.25 W in series),
- Polarized line (Pull Up/Down: 470 ohms/0.25W polarization resistor),
- Use of a shielded cable,
- male RJ45 connectors,
- COMMON signal connected directly to the protective ground (earth) and to a point on the bus.

Parametering

Parameters for the Modbus characteristics of the Zelio 2 module can be set in the Programming Software using the **Edit → Program Configuration** menu, **Modbus Extension** tab, or by clicking on the Program configuration icon  .

Number of wires and format:

- 2-wire, RTU,
- 4-wire, RTU,
- 2-wire, ASCII,
- 4-wire, ASCII.

Speed in bauds

Transmission speed (bauds): 1200, 2400, 4800, 9600, 19200, 28800, 38400 and 57600.

Parity

- None,
- Even,
- Odd

Modbus address of slave:

- Network address: 1 to 247.

Default settings: 2-wire, RTU, even parity, address 1, 19200 bauds.

Data Exchanged

The module has four 8-bit data exchange words, four clock words and one status word.

Data

The data exchanged is specific to programming mode: **LD** *(see page 149)* or **FBD** *(see page 267)*.

Clock

The Modbus extension allows the Modbus master to access (read or write) to the clock.

Every modification to one of the 4 clock words updates the smart relay clock.

Smart relay firmware time update:

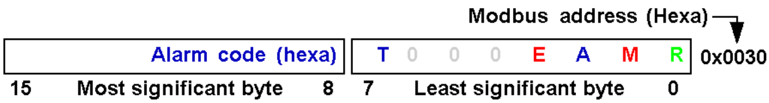
15 Most significant byte 8	7 Least significant byte 0	Modbus Address (Hexa)
Seconds	Day of week	0x0020
Hours	Minutes	0x0021
Month	Day of the month	0x0022
Century	Year	0x0023

The possible values (to be converted into hexadecimal):

- Seconds: 0 to 59
- Minutes: 0 to 59
- Hours: 0 to 23
- The day of the week is calculated automatically,
- Day of the month: 1 to 31,
- Month: 1 to 12,
- Year: 0 to 255 (2000 to 2255),
- Century: 21 (not used).

Status

The status word can only be accessed by the Modbus master.



R State of the smart relay:

- 0: The module is stopped.
- 1: The module is in run mode.

M Monitoring:

- 0: The module is not in Monitoring mode.
- 1: The smart relay is in Monitoring mode.

A alarm status:

- 0: Alarm detection is not active
- 1: Alarm detection is active.

E Error:

- 0: No error activated
- 1: Error activated (blocking error)

T Time Out:

- 0: Time Out period observed
- 1: Time Out period exceeded

The alarm code contains the code of the error detected by the firmware *(see page 521)*.

Wiring

2-wire Modbus. Use of a shielded cable: 1 twisted pair for D1-D0 and a third wire (or 1 twisted pair) for COMMON:

	Modbus Zelio 2 slave	Modbus master
RJ45	Signal	Signal
1	RXD0	N.C.
2	RXD1	N.C.
3	N.C.	N.C.
4	**TXD1**	**D1**
5	**TXD0**	**D0**
6	N.C.	N.C.

	Modbus Zelio 2 slave	Modbus master
7	N.C.	N.C.
8	COMMON	COMMON

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not connect wires to unused terminals and/or terminals indicated as “No Connection (N.C.)”.

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

4-wire Modbus. Use of a shielded cable: 1 twisted pair for RXD1-RXD0, 1 twisted pair for TXD1-TXD0 and a fifth wire (or 1 twisted pair) for COMMON.

	Modbus Zelio 2 slave	Modbus master
RJ45	Signal	Signal
1	**RXD0**	**TXD0**
2	**RXD1**	**TXD1**
3	N.C.	N.C.
4	**TXD1**	**RXD1**
5	**TXD0**	**RXD0**
6	N.C.	N.C.
7	N.C.	N.C.
8	COMMON	COMMON

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Do not connect wires to unused terminals and/or terminals indicated as “No Connection (N.C.)”.

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

Chapter 28
SR2COM01 Communication Interface

Subject of this Chapter

This chapter describes the programming software functions relating to the SR2COM01 communication interface.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Zelio2 COM Menu	468
Directories Menu	470
Configuring the SR2COM01 Communication Interface	476
Sending an Email via SMS	482
Description of the Error Codes of the SR2COM01 Communication Interface	483

Zelio2 COM Menu

Description

This menu groups the functions relating to the connection of the programming software to the SR2COM01 communication interface.

These functions are the following:
- Zelio2 COM Diagnostics *(see page 468)*
- Update Zelio2 COM Firmware *(see page 468)*
- Message ON *(see page 469)*
- Message OFF *(see page 469)*

NOTE: To use these functions, the programming software must be connected to the SR2COM01 communication interface.

Zelio2 COM Diagnostics

Use the **Zelio2 COM Diagnostics** function to view the characteristics of the communication interface to which the programming software is connected. In particular, you can use it to verify the hardware and firmware versions. To do this, proceed as follows:

Step	Action
1	Click on the **Zelio2 COM → Zelio2 COM Diagnostics** menu. **Reminder**: To use this function, the programming software must be connected to the communication interface.

NOTE: After transferring the program, reinitialize communication interface to activate the modem link.

Update the Zelio2 COM Firmware

Use the **Update the Zelio2 COM Firmware** function to select and download the firmware to the communication interface. To do this, proceed as follows:

Step	Action
1	Click on the **Update Zelio2 COM Firmware → Update Zelio2 COM Firmware** menu. **Reminder**: To use this function, the programming software must be connected to the communication interface.
2	Click on the **Yes** button in confirmation request window to begin the transfer to the communication interface.

Message ON

The **Message ON** function is used to **activate** the **Message** function blocks of the application. Once active, the **Message** function blocks send their alarm messages as soon as the associated condition for generation becomes true.

To activate the **Message** function blocks, proceed as follows:

Step	Action
1	Click on the **Zelio2 COM → Message ON** menu. **Reminder**: To use this function, the programming software must be connected to the communication interface.

NOTE: By default, the **Message** function blocks are activated after the program is transferred.

Message OFF

The **Message OFF** function is used to **deactivate** the **Message** function blocks of the application. Even if the condition for generation of the alarm message of a **Message** function block becomes true, the message will not be sent.

This function is used, for example, to avoid alarm messages being sent during debugging of an application.

To deactivate the **Message** function blocks, proceed as follows:

Step	Action
1	Click on the **Zelio2 COM → Message OFF** menu. **Reminder**: To use this function, the programming software must be connected to the communication interface.

Directories Menu

Description

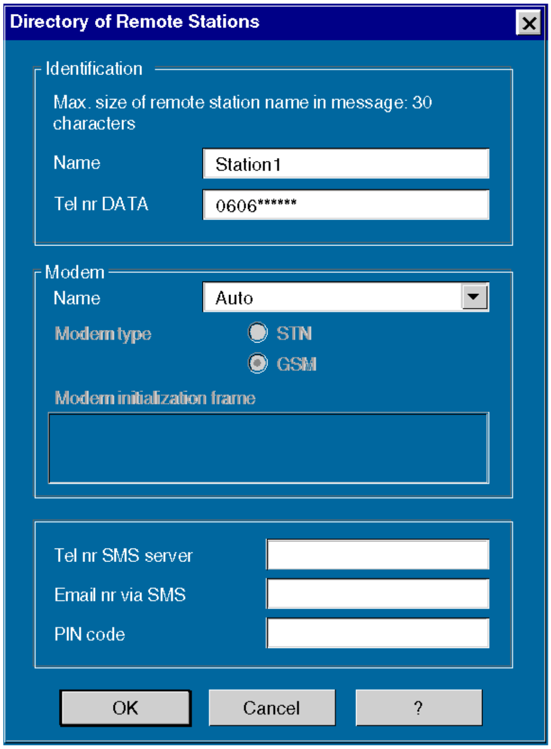
The **Directories** menu lists the programming software functions that are used to create or modify the different directories required to use the SR2COM01 communication interface.

There are three types of directories:
- The **Directory of Remote Stations** *(see page 470)*: Lists target smart relays and provides for each the telephone numbers and configuration parameters for the associated modems.
- The **Program recipients directory** *(see page 473)*: Lists recipients usable in the program being written, provides their telephone number or email address, and specifies the commands that they are allowed to execute.
- The **Recipients General Directory** *(see page 474)*: Lists the recipients regularly used in programs. It can be used so that they do not have to be recreated for another program.

Remote Stations Directory

To create or modify the Remote Stations Directory, proceed as follows:

Step	Action
1	Click on the **Directories → Directories of Remote Stations...** menu. **Result:** The **Directory of Remote Stations** window appears and for each remote station shows the following: ● The **Name** of the remote station. ● The type of **Modem**. ● The SIM card number of your SR2MOD02 in international format (1). ● The **PIN code** (only for GSM modems). ● The SIM card SMS server number of your SR2MOD02 in international format (1). ● The **Email no. via SMS**, used to send an SMS to an Email address (contact the telephone operator of the SIM card to find out if this option is offered).
(1) International format: replace 00 by +.	

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>2</td><td>Click on the Create button to add a remote station to the directory.
Result: The following window appears:


NOTE: To modify an existing remote station, select the remote station and click on the Modify button.</td></tr>
<tr><td>3</td><td>In the Identification section enter the following:
● The name of the remote station.
● The DATA telephone number of the remote station.
NOTE: When using GSM modems to send an Email via SMS, the syntax used in the remote station name is specific to each telephone operator. Contact the telephone operator of the modem SIM card and refer to the section Sending an Email via SMS, page 482 for more information.</td></tr>
<tr><td colspan="2">(1) International format: replace 00 by +.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>4</td><td>In the Modem section, choose the modem used:
Auto: The system will automatically configure the modem detected (SR2MOD01 or the SR2MOD02).
SR2MOD01: Uses the configuration predefined for the RTC reference modem SR2MOD01.
SR2MOD02: Uses the configuration predefined for the GSM reference modem SR2MOD02.
Other: You must manually enter the name, the type of modem, its initialization frame (for more information about the initialization frame, contact the manufacturer) and configure it using the following parameters:
serial speed: 115,200 baud
7 data bits, 1 stop bit, 1 parity bit
even parity
DSR ON
echo deactivated
NOTE: Only one other modem can be created.</td></tr>
<tr><td>5</td><td>For the GSM modem only, fill in the following:
The Tel no. SMS server (required field).
The Email no. via SMS.
The PIN code corresponding to the SIM card of the modem. If no PIN code is associated with the SIM card, leave the field empty.
NOTE: The telephone number of the SMS server and the Email number by SMS are specific to each telephone operator. Contact the telephone operator of the SIM card of this modem to obtain them.</td></tr>
<tr><td>6</td><td>Confirm by clicking on the OK button.</td></tr>
<tr><td>7</td><td>Repeat steps 2 to 6 for each remote station to create or modify.</td></tr>
<tr><td>8</td><td>Confirm by clicking on the OK button.</td></tr>
<tr><td colspan="2">(1) International format: replace 00 by +.</td></tr>
</table>

Program Recipients Directory

To create or modify the Program recipients directory, proceed as follows:

Step	Action
1	Click on the **Directories → Directories of Program Recipients** menu. **Result:** The **Program recipients directory** window appears and for each remote station shows the following: ● The **name** of the recipient. ● The **telephone number** or the **email** address. ● The **type of recipient**. ● The **number of connection attempts** in the event of unsuccessful message (network interruption, busy signal, etc.) before the remote station attempts to contact the recipient following the list. ● The **recognition authorization** (only for GSM mobiles). ● The **recognition delay**: The delay (in minutes) given to the recipient to send back the acknowledgment. ● The **authorization to modify variables**. **NOTE:** For more information on sending variable modification commands, see the on-line help of the Zelio Logic Alarm software.
2	Add Recipients: ● Either by creating them: Click on the **Create** *(see page 475)* button. or ● By importing it from the general directory: Click on the **Gen. Dir.** button, and then: ❍ Select the recipient in the **Recipients general directory** section. ❍ Confirm by clicking on the **OK** button.
3	Confirm by clicking on the **OK** button.

General Directory of Recipients

The Recipients general directory is independent of the program being edited. It can be used to save the information of the recipients used regularly in the programs. To create or modify the Recipients General Directory, proceed as follows:

Step	Action
1	Click on the **Directories** → **Recipients general directory** menu. **Result:** The **Recipients general directory** window appears and for each recipient shows the following: ● The **name** of the recipient. ● The **telephone number** or the **Email** address. ● The **number of connection attempts** in the event of unsuccessful message (network interruption, busy signal, etc.) before the remote station attempts to contact the recipient following the list. ● The **recognition authorization** (only for GSM mobiles). ● The **recognition delay**: The delay (in minutes) given to the recipient to send back the acknowledgment. ● The **authorization to modify variables**. **NOTE:** For more information on sending variable modification commands, see the on-line help of the Zelio Logic Alarm software.
2	It is possible: ● To Add Recipients: Click on the **Create** *(see page 475)* button. ● To Modify a Recipient: Select the recipient then click on the **Modify** button. ● To Delete a Recipient: Select the recipient then click on the **Delete** button. ● To import recipients from a given program: Click on **Import** button, and then: ❍ Select the program and confirm. ❍ Select the recipient in the **Imported Directory** section. ❍ Confirm by clicking on the **OK** button.
3	Confirm by clicking on the **OK** button.

Creating a Recipient

When creating a recipient, after clicking on the **Create** button (in the Recipients general directory or in the Program recipients directory), proceed as follows:

Step	Action
1	Select the type of recipient from among: ● Zelio Logic Alarm ● Mobile phone ● Email
2	Enter recipient name.
3	Enter the telephone number or the Email (use the international format for mobile telephones, for example: +33670••••••). **NOTE:** When using GSM modems to send an **Email via SMS**, the syntax used in the recipient's Email is specific to each telephone operator. Contact the telephone operator of the SIM card of the modem and refer to the section *Sending an Email via SMS, page 482* for more information.
4	Specify the number of connection attempts in the event of unsuccessful message (network interruption, busy signal, etc.) before the remote station attempts to contact the recipient following the list.
5	You can also check the **Variable Modification Authorized** box (only for **Zelio Logic Alarm** and **Mobile**-type recipients) to let them modify certain variables associated with message blocks. **NOTE:** When receiving a modification command, the communication interface identifies the recipient by its telephone or modem number. The recipient should thus make sure that its number is not masked when it sends a command.
6	You can also check the **Recognition Authorized** box, to activate the acknowledge system for this recipient (only for **Mobile**-type recipients). In this case, specify the recognition delay (in minutes).
7	Confirm by clicking on the **OK** button.

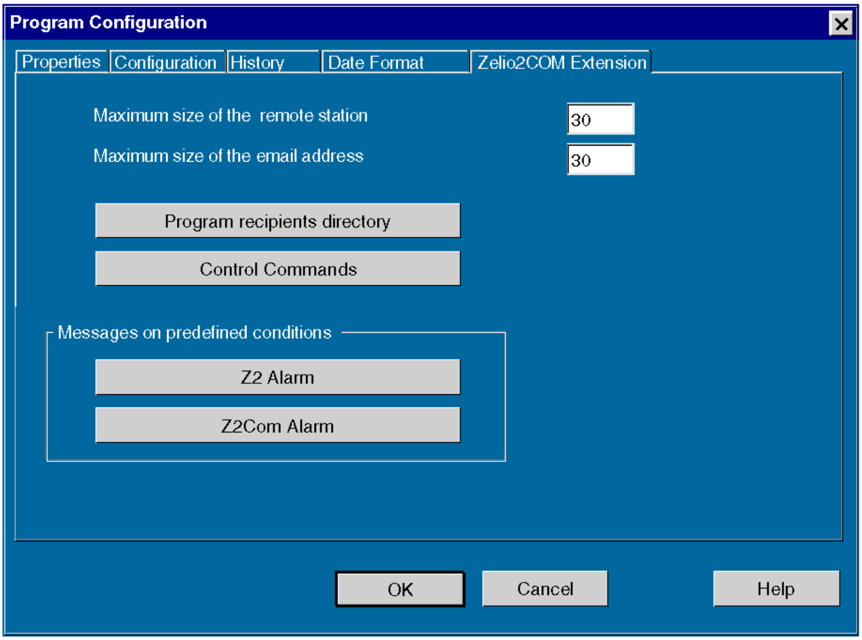
Configuring the SR2COM01 Communication Interface

Description

Detailed information about the parameters required to configure the SR2COM01 communication interface is provided below.

Parameters

The **Zelio2COM Extension** tab of the **Program Configuration** window is used to configure the SR2COM01 communication interface.



The configuration window is accessible using the **Edit→ Program Configuration** menu, or using the  icon located under the edit window.

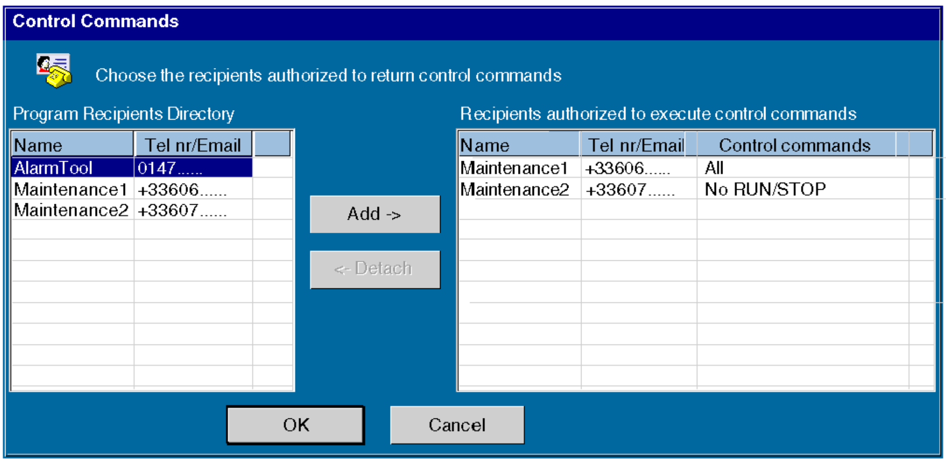
The parameters to enter for configuring the communication interface are the following:

- **Maximum size of the remote station**, the name must be less than 30 characters long (default value),
- **Maximum Size of Email Address**, the name must be less than 30 characters long (default value), this determines the maximum length of email addresses of recipients or alarm messages,
- **Program recipients directory**, this button is used to display and modify the directory, refer to Program recipients directory *(see page 473)*,

- The list of recipients who are authorized to execute **Control Commands**, this button is used to display and modify this list, refer to Control Commands *(see page 477)*.
- The messages on predefined conditions:
  - **Z2 Alarm**: This button is used to define the alarm message sent when the smart relay firmware detects an error, refer to Z2 Alarm *(see page 478)*,
  - **Z2Com Alarm**: This button is used to define the alarm message sent when the SR2COM01 interface firmware detects an error, refer to Z2Com Alarm *(see page 479)*.

Control Commands

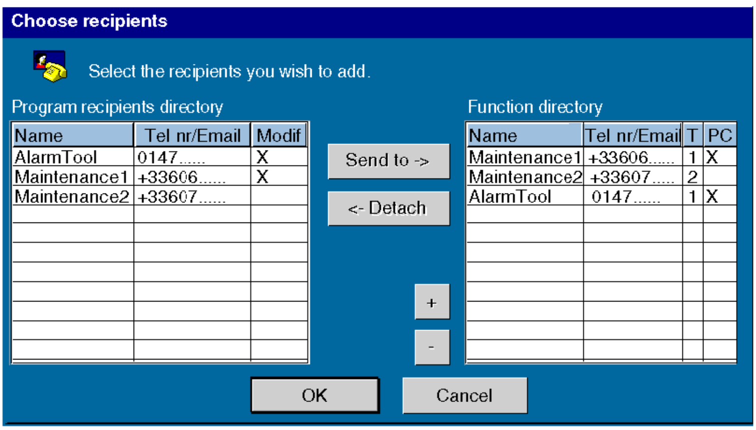
Control commands are used to read/modify the controller configuration settings of the smart relay and monitor its status. To choose among the program recipients the ones that are allowed to execute control commands, proceed as follows:

Step	Action
1	Click on the **Control Commands** button. **Result**: The following window appears: 
2	Select, in the **Recipients Directory**, the recipient to add in the list of the **Recipients authorized to execute control commands**.
3	Click on the **Add→** button.
4	Double click, in the **Recipients authorized to execute control commands** zone, on the recipient just added. In the drop-down list choose: ● **No RUN/STOP**: To authorize this recipient to execute control commands, **except** the **RUN** and **STOP** commands, ● **All**: To authorize this recipient to execute all control commands. **NOTE:** For more information on sending commands, see the on-line help of the Zelio Logic Alarm software.

Step	Action
5	Repeat steps 2 to 4 for each recipient in the list of **Recipients authorized to execute control commands**. **NOTE:** The list of **Recipients authorized to execute control commands** contains a maximum of 10 recipients.
6	Confirm by clicking on the **OK** button.

Z2 Alarm

This button is used to define the alarm message sent when the **smart relay** firmware detects an error, proceed as follows:

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1</td><td>Click on the Z2 Alarm button.
Result: The Z2 Alarm window appears.</td></tr>
<tr><td>2</td><td>Click on the button  of the Message Recipient zone to add a recipient or modify the list of recipients of this message.
Result: The following window appears:


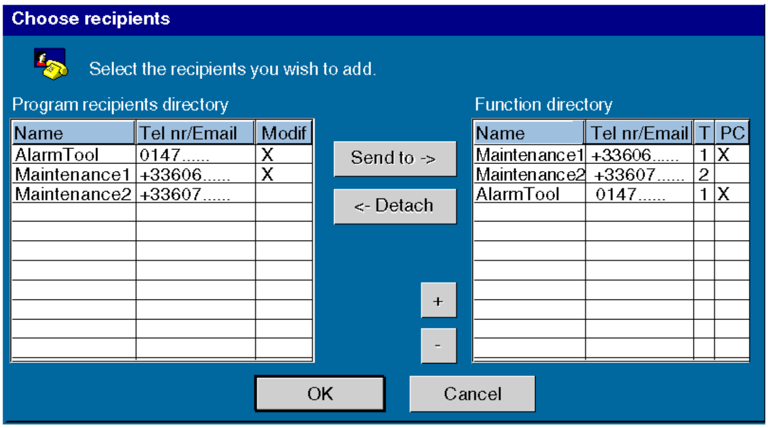
These recipients are chosen from the Program recipients directory (see page 473).</td></tr>
<tr><td>3</td><td>For each new recipient to be added, select it in the directory of the program and click on the Send to → button.</td></tr>
<tr><td>4</td><td>Organize the recipients in the order of priority (see page 481) by using the + and - buttons.</td></tr>
<tr><td>5</td><td>Confirm by clicking on the OK button</td></tr>
</table>

Step	Action
6	In the Message to Transmit zone, enter: ● The message object. ● The message body. **NOTE:** The total length of the message object and body should not exceed 160 characters. **NOTE:** For GSM modems, to send an **Email by SMS**, the syntax to use in the subject and message body is specific to each telephone operator. Contact the telephone operator of the SIM card of the modem and refer to the section Sending an Email via SMS *(see page 482)* for more information.
7	If necessary, check the **Display error code** box to include the code of the error detected in the message body. **NOTE:** ● It is possible to move this text: Select **ERR=___** and drag and drop. ● Refer to Smart Relay Errors *(see page 521)*, for more information on smart relay error codes.
8	Confirm by clicking on the **OK** button.

Z2Com Alarm

To define the alarm message sent when the **communication interface** firmware detects an error, proceed as follows:

Step	Action
1	Click on the **Z2Com Alarm** button. **Result**: The **Z2Com Alarm** window appears.

Step	Action
2	Click on the button ... of the **Message Recipient** zone to add a recipient or modify the list of recipients of this message. **Result**: The following window appears:  These recipients are chosen from the Program recipients directory *(see page 473)*.
3	For each recipient to be added, select it in the directory of the program and click on the **Send to →** button.
4	Organize the recipients in the order of priority *(see page 481)* by using the **+** and **-** buttons.
5	Confirm by clicking on the **OK** button
6	In the Message to Transmit zone, enter: ● The message object. ● The message body. **NOTE:** For GSM modems, to send an **Email by SMS**, the syntax to use in the subject and message body is specific to each telephone operator. Contact the telephone operator of the SIM card of the modem and refer to the section Sending an Email via SMS *(see page 482)* for more information.
7	If necessary, check the **Display error code** box to include the code of the error detected in the message body. **NOTE:** ● It is possible to move this text: Select **ERR=___** and drag and drop. ● Refer to Communication Interface Errors *(see page 483)*, for further information on the smart relay error codes.
8	Select the **Condition for Generating Message** of the alarm. One (or more) boxes may be checked among the following: ● **Power returns after power outage**, ● **Zelio2 - Zelio2 COM link Break** ● **All Zelio2 Com Errors** (if this box is checked, the 2 preceding boxes are automatically checked).
9	Confirm by clicking on the **OK** button.

Order of Priority

When sending an alarm message, the SR2COM01 communication interface contacts the message recipients one after the other. The **Choose Recipients** window is used to define the order in which the message recipients are contacted.

Depending on whether the **Recognition** option is activated or not, two types of recipients may be defined:

- **Recipient without recognition**: The alarm message is systematically sent to it, then the communication interface processes the next recipient via modem,
- **Recipient with recognition** (for mobile-type recipients only): The modem communication interface sends the alarm message and waits for acknowledgment from the recipient via their mobile phone:
  - If the recipient with recognition acknowledges the message, the modem communication interface proceeds with the send sequence only to recipients **without** recognition
  - If the recipient with recognition does not acknowledge the message in the given time (**Recognition delay**), the communication interface processes the next recipient.

For more information on activating the **Recognition** option, refer to Creating a Recipient *(see page 475)*.

Sending an Email via SMS

Description

The following describes the frame of an alarm message based on the parameters of the communication interface and `Message` function blocks.

Message Structure

The structure of the SMS frame sent to the SMS Email server of the SIM card operator is described below.

<Email of recipient><Name of the remote station>␣<Date time>␣<Object of alarm message><Body of the alarm message>

In the frame above:
- The characters *<* and *>* represent the different field limits and are not present in the frame sent.
- The character ␣ represents a space character.

For example:

The remote station **StationPompage1** sends an alarm message to the recipient whose Email is **Maintenance1@•••.com** at 7:35 PM on January 9th, 2006. The object of the alarm message is **Alarm levels max attained**, the message body is **Level=2.80 m**.

The frame of the alarm message sent is thus:

Maintenance1@•••.comStationPompage1 01/06/09 19:35 Alarm levels max attainedLevel=2.80 m

Respect the Syntax Specific to the Operator

Respect the syntax specific to the SMS Email server of the operator of the SIM card of the remote station. For that, use the necessary characters in the affected fields (recipient directory, remote-station directories, message function block, etc.).

Description of the Error Codes of the SR2COM01 Communication Interface

Description

Below is a description of the errors detected by the SR2COM01 communication interface firmware via modem extension.

Error Codes

List of errors:

Code	Type of error
05	**Recipient error** The type of recipient is incorrect. Modify the recipient. **NOTE:** This error is non-blocking.
42	**Smart Relay Application Checksum Error** The application in the smart relay is not correct. Transfer the application in the programming software to the smart relay.
43	**Power supply return** A power outage had been detected. **NOTE:** This error is non-blocking.
44	**PIN Error Codes** Two incorrect PIN codes have been entered. Verify the PIN code on the SIM card of the GSM modem connected to the communication interface.
45	**IS463 Unknown Status** An unknown operation was executed. Reload the firmware and the communication interface.
46	**Program error** The application in the smart relay is absent. Transfer the application in the programming software to the smart relay.
47	**IS498 Unknown Status** An unknown operation was executed. Reload the firmware and the communication interface.
48	**Modem Absent** The modem cannot be detected. Verify: • The link between the SR2COM01 communication interface and the modem. • Modem power supply.
49	**Smart Relay Absent** The smart relay cannot be detected. Verify: • The link between the SR2COM01 communication interface and the smart relay. • The state of the smart relay. **NOTE:** This error is non-blocking.

Code	Type of error
50	**Binary error** The firmware of the communication interface is damaged. Reload the firmware and the communication interface.
51	**Modem parameters incorrect** Verify the modem configuration using the programming software. For more details, refer to Directory of Remote Stations *(see page 470)*.
58	**Watchdog error** Internal watchdog overflow. **NOTE:** This error is non-blocking.
60	**OFF Alarms** Alarm message send is disabled. For more details, refer to Menu Zelio2Com->Message OFF *(see page 469)*.
61	**SIM Erase Error** A SIM card erase error has been detected. Verify the SIM card.

Chapter 29
Analog Input-Output Extension SR3XT43BD

Analog Input-Output Extension SR3XT43BD

Description

The Analog Input-Output extension `SR3XT43BD` allows the use of an additional 2 analog inputs and 2 outputs. These Inputs-Outputs are coded on 10 bits.

Programming Language

If an analog Input-Output extension was added during the Choice of Smart Relay/Programming Type *(see page 26)* process, only the FBD programming language is available.

The extension inputs are represented on the wiring sheet by the contacts:
- `IH XT2`
- `IJ XT2`

The extension outputs are represented on the wiring sheet by the contacts:
- `QB XT2`
- `QC XT2`

Parametering

The features of each of the 2 inputs and 2 outputs of the extension are configurable in the programming software in the **Analog Extension** of the **Program Configuration** window.

The configuration window may be accessed through **Edit → Program Configuration** menu, or by clicking on the  icon.

In the **IH input** section, choose:
- **0-20 mA**: If the input connected on contact `IH XT2` is a current input.
- **0-10 V**: If the input connected on contact `IH XT2` is a voltage input.

In the **IJ input** zone, choose:
- **0-20 mA**: If the input connected on contact `IJ XT2` is a current input.
- **0-10 V**: If the input connected on contact `IJ XT2` is a voltage input.
- **PT100**: If the sensor connected to contact `IJ XT2` is a PT100 type thermoresistor.

In the **QB Output** and **QC Output** section, specify the **acceleration time** of `QB XT2` and `QC XT2` output, according to the dynamic of the system. The acceleration time is the time taken by the output to go from 0 V to 10 V or from 10 V to 0 V. When an acceleration time is specified, the output is not disabled when the smart relay is in **STOP** mode.

Measurement reading delays

In the event of power outage or variations between 0 and the full scale, analog input measurements will be delayed:

- 20 mA: typically 150 milliseconds
- Pt100:typically 4 seconds

Chapter 30
Communication Via Ethernet Extension

Subject of this Chapter

This chapter describes the functions and settings related to communication via the Ethernet Extension.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
At a Glance	488
Acquiring IP Addresses	491
Communication on an Ethernet Network	494
Requests Specific to TCP Diagnostics	499

At a Glance

Introduction

The Ethernet **SR3NET01BD** extension may be added to any of the following reference smart relays:
- SR3B101BD
- SR3B102BD
- SR3B261BD
- SR3B262BD

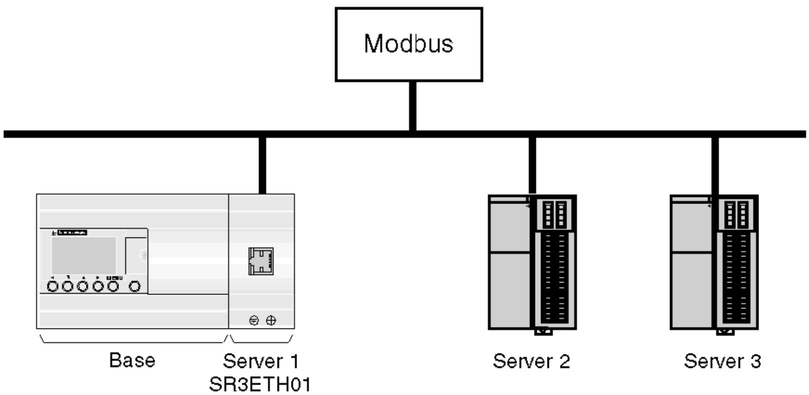
To use the Ethernet connection, choose the FBD programming mode.

The architecture and protocols used are presented in the following section.

Ethernet Network Architecture

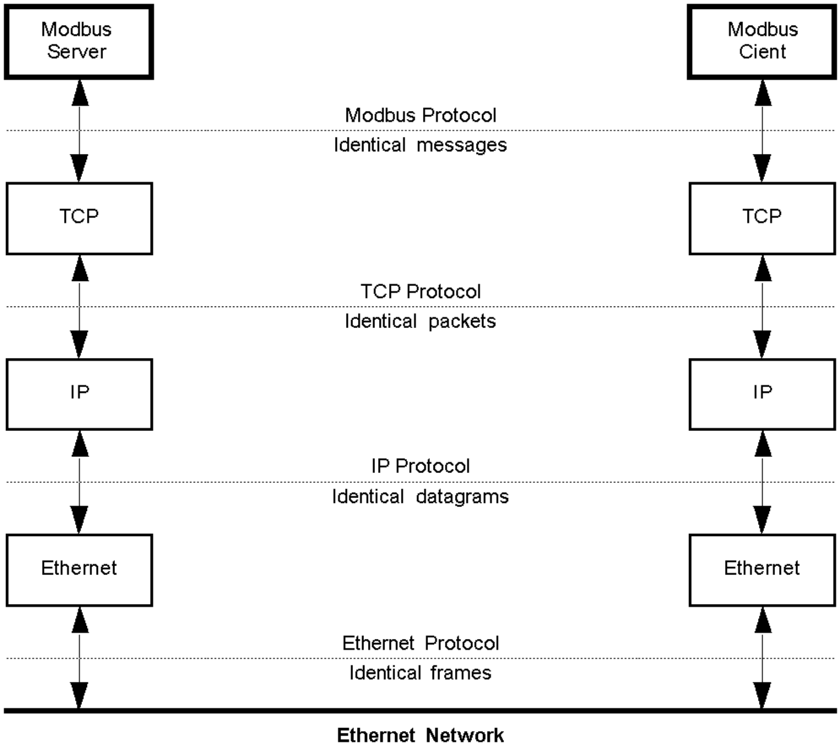
The Ethernet extension is present as a server on the network.

Example of network architecture with an Ethernet extension:



Protocol Stack Architecture

Data is exchanged between a client and a server in different protocol layers in accordance with the following diagram:



Ethernet Protocol

Ethernet is used to exchange data between several devices, known as hosts, connected to each other via the network.

An Ethernet message contains, in particular:
- The recipient address to enable the message to be acquired by the destination device.
- The sender's address to enable a response to be sent to the transmission device.

IP Protocol

IP protocol is used to connect networks to each other and communicate between networks via gateways.

In a heterogeneous network, the various devices which relay messages between the source and target devices may be subject to a limitation on the length of messages that they can transmit. IP (Internet Protocol) fragments the data to be exchanged between the devices into datagrams so that they can be accepted across various devices.

TCP Protocol

TCP (Transmission Control Protocol) controls the reliability and scheduling of the transmission.

TCP works in online mode, providing a virtual point-to-point connection between the communicating devices.

TCP manages the flow of exchanged data. It is used to monitor the arrival of transmitted packets, and then reassemble them for execution.

NOTE: The default Modbus exchange service which acts as a server can be accessed from the TCP port with address 502.

Modbus TCP/IP Protocol

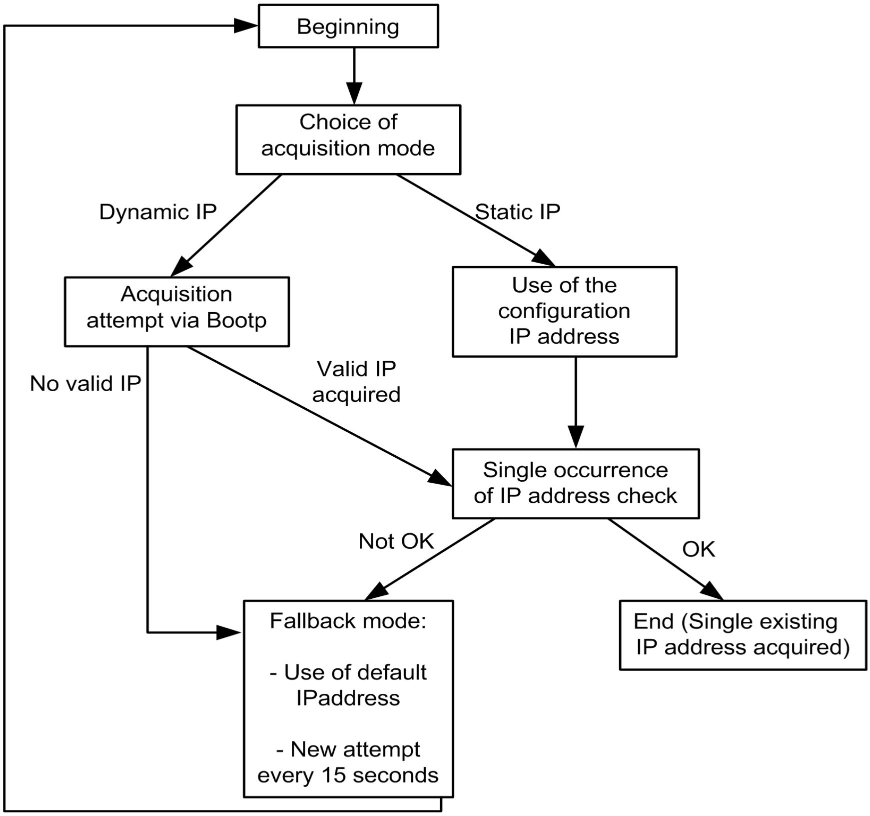
Modbus TCP/IP is based on a client / server model. Each Modbus server has an array of registers from which clients can read or write data.

Acquiring IP Addresses

Introduction

The Ethernet extension only responds to messages that are sent to it. First it needs to be configured with its own IP address and be known by the gateway.

The following graph indicates the general process involved in acquiring extension IP addresses:



There are two possible acquisition modes:

- Static *(see page 492)* (with or without subnet mask *(see page 492)*).
- Dynamic *(see page 493)* (only with a Bootp server).

Whatever the acquisition mode, in some situations the Ethernet extension may go into fallback mode *(see page 493)* and use the default IP address *(see page 493)*.

Static Acquisition Mode

In static acquisition mode:

- The Ethernet extension and gateway IP addresses are derived from the configuration in the programming software.
- The addresses may be masked by a subnet mask *(see page 492)*.
- Duplicates of IP addresses are not allowed on the network. If the Ethernet extension detects duplicates of an address, it switches into fallback mode *(see page 493)* and makes a new attempt after 15 seconds.

Subnet Mask

A subnet mask is used to address several physical networks with a single network address.

The mask separates the subnet address from the host device address as follows:

- The subnet address is obtained by keeping the bits of the IP address which correspond to the mask positions containing 1 as they are, and by replacing the others with 0.
- The subnet host device address is obtained by keeping the bits of the IP address which correspond to the mask positions containing 0 as they are, and by replacing the others with 1.

Example 1:

	Byte 1	Byte 2	Byte 3	Byte 4
IP address	192(11000000)	1(00000001)	17(00010001)	11(00001011)
Subnet mask	255(11111111)	255(11111111)	0(00000000)	0(00000000)
Subnet address	192(11000000)	1(00000001)	0(00000000)	0(00000000)
Host device address	255(11111111)	255(11111111)	17(00010001)	11(00001011)

Example 2:

	Byte 1	Byte 2	Byte 3	Byte 4
IP address	192(11000000)	1(00000000)	17(00010001)	11(00001011)
Subnet mask	255(11111111)	255(11111111)	240(11110000)	0(00000000)
Subnet address	192(11000000)	1(00000000)	16(00010000)	0(00000000)
Host device address	255(11111111)	255(11111111)	241(11110001)	11(00001011)

Dynamic Acquisition Mode

In dynamic acquisition mode:

- A BootP server must be present on the network.
- Configure the BootP server using the MAC address which is written on the side of the Ethernet extension.
- When IP acquisition starts, the Bootp server enables the Ethernet extension to obtain its IP parameters from its MAC address and to be recognized by the gateway.
- If the Ethernet extension does not obtain a valid IP address from the Bootp server or if it detects a duplicate of an IP address on the network, it will switch into fallback mode *(see page 493)* and send a Bootp request to the server every 15 seconds.

Fallback Mode

The Ethernet extension will switch into **fallback mode** if it does not obtain a valid IP address or if it detects duplicates of an IP address on the network. It then provides the FDR (Fast Device Replacement) service and reacts as follows:

Phase	Description
1	The Ethernet extension uses its own default IP address *(see page 493)* and checks that this address exists only once on the network (duplicates of IP addresses are not allowed on the network).
2	• If there are no duplicates, the default IP address will be used. • Otherwise, the IP address field will remain empty.

3	Every 15 seconds
	• In static acquisition mode, the Ethernet extension uses the configuration IP address and checks that this address exists only once on the network:
	  ○ If no duplicates are detected, it switches into `normal operating mode`.
	  ○ Otherwise, it will restart phases 1, 2 and 3.
	• In dynamic acquisition mode, the Ethernet extension sends a Bootp request to the server:
	  ○ If it obtains a valid IP address which exists only once on the network, it will switch into `normal operating mode`.
	  ○ Otherwise, it will restart phases 1, 2 and 3.

Default IP Address

A default IP address is derived from the MAC address of the Ethernet extension. It is made up of 4 bytes. The two first bytes are 85 and 16. The two final bytes are the two last bytes of the MAC address.

Example: MAC address = 00.80.F4.85.00.1A default IP address = 85.16.0.26.

NOTE: A MAC address is written in hexadecimal. An IP address is written in decimal form. Therefore, you must convert the codes.

Communication on an Ethernet Network

At a Glance

The **Ethernet SR3NET01BD** extension may be added to a smart relay with any of the following references:

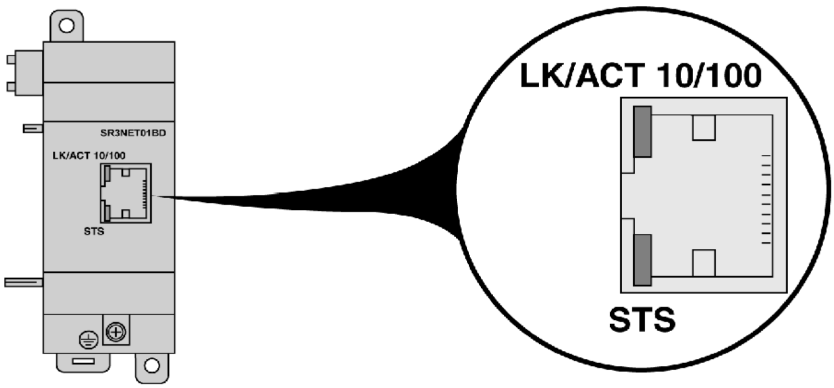
- **SR3B101BD**
- **SR3B102BD**
- **SR3B261BD**
- **SR3B262BD**

To use the Ethernet connection, choose the FBD programming mode.

The Ethernet extension exchanges Modbus message as a server, using the suite of TCP/IP protocols and Ethernet technology. For more details, refer to At a Glance *(see page 488)*.

Ethernet Extension

The Ethernet extension has a RJ45 female connector and two LEDs.



The states of each LED, and what they mean, are described below.

- LK/ACT 10/100
  - Off: No Ethernet connection.
  - Green light: 100 Mbits/s Ethernet connection.
  - Green flashing light: Data exchange at 100 Mbits/s.
  - Yellow light: 10 Mbits/s Ethernet connection.
  - Yellow flashing light: Data exchange at 10 Mbits/s.
- STS
  - Light continually on: Power is being supplied to the Ethernet extension and is ready for communication.
  - Rapid flashing light: Ethernet communication is being initialized.
  - 4 flashes: IP address detected more than once on the network.

- 5 flashes: Get IP Address is in progress (in normal operating mode or before switching to fallback mode *(see page 493)*).
- 6 flashes: Get IP Address in progress after switching to fallback mode *(see page 493)*.

The MAC address is engraved on the side of the Ethernet extension.

Ethernet Network Connections

The Ethernet extension is able to manage a maximum of four simultaneous TCP connections. Client applications must be designed so that this limit will not be exceeded. If a request is made to open a fifth connection, it will be automatically rejected by the Ethernet extension.

A connection is considered to be inactive if no Modbus request was received during a timeout period. The Ethernet extension closes inactive connections that do not come from the **Reserved Address** (It does not take account of the TCP `Keepalive` messages). The timeout period can be configured. Its default value is 10 minutes. When the maximum number of connections has been reached, new connections are authorized when one of the existing ones expires.

Reserved Address

An address may be reserved for a client to which the Ethernet extension must constantly remain connected. The connection between the Ethernet extension and the client with the **Reserved Address** is not subject to a timeout period. This connection is included in the maximum number of four connections.

How to Set the Communication Parameters

To configure the communication parameters, proceed as follows:

Step	Action
1	Select the **Edit** → **Program configuration** menu and the **Ethernet Extension** tab or click on the **SR3NET01BD** button.
2	Does the network have a Bootp server (see *Acquiring IP Addresses, page 491*)? ● If the answer is yes, then select **Dynamic Address** and go to step 6. ● If not, then select **Static Address** and go to step 3.
3	In the **IP Address** field, enter the four bytes of the Ethernet extension IP address in decimal format.
4	If the Ethernet extension and the gateway are part of a sub-network determined by a mask, then enter the four bytes of the subnet mask in the **Subnet Mask** field in decimal format (see *Subnet Mask, page 492*). **NOTE:** 0.0.0.0 indicates that there is no mask.

Step	Action
5	In the **Gateway Address** field, enter the four bytes of the gateway IP address in decimal format. **NOTE:** The gateway is a sub-network device (also called a router) that allows your network segment to access other network segments of your company's overall network, the Internet and a remote Intranet. When installing your new Ethernet extension on the existing network, contact your network administrator for information on the gateways.
6	If the Ethernet extension must be constantly connected to a client, enter the client's address in the Reserved Address *(see page 495)* field.
7	Where applicable, you may modify the timeout period, i.e. the time after which the Ethernet extension must close an inactive connection with the gateway if this connection is not with a **Reserved Address**. **NOTE:** Take into account the Maximum of Four Connections *(see page 495)*.
8	Click on **OK**.

Data Exchanges with the Modbus client

The Ethernet extension can exchange eight data words, four clock words and one status word with the **Modbus TCP/IP** client.

Data

The data exchanged is as follows:

- Four 16-bit input words, each accessible through the application via **J1** to **J4** contacts and open to client(s) in read and write modes. For more details, refer to Ethernet Inputs *(see page 269)*.
- Four 16-bit output words, each accessible through the application via **O1** to **O4** contacts and open to client(s) in read mode only. For more details, refer to Ethernet Outputs *(see page 269)*.

Clock

The Ethernet extension is used by the client to access the smart relay clock in read and write modes.

Each modification to one of the 4 clock words updates the smart relay clock.

Smart relay firmware time update:

Seconds	Day of the week	32
Hours	Minutes	33
Month	Day of the month	34
Century	Year	35
15 Most significant byte 8	7 Least significant byte 0	

The possible values (to be converted into hexadecimal):

- Seconds: 0 to 59
- Minutes: 0 to 59
- Hours: 0 to 23
- The day of the week is calculated automatically
- Day of the month: 1 to 31
- Month: 1 to 12
- Year: 2 to 99 (2002 to 2099)
- Century: 20 (not used)

Status

The status word may be accessed by the client in read mode only.

Alarm code (hexa)	T	0	0	0	E	A	M	R	48
15 Most significant byte 8	7	Least significant byte						0	

R State of the smart relay:

- 0: The smart relay is OFF.
- 1: The smart relay is ON.

M Monitoring:

- 0: The smart relay is not in MONITORING mode.
- 1: The smart relay is in MONITORING mode.

A alarm status:

- 0: Alarm detection is not active.
- 1: Alarm detection is active.

E Error:

- 0: No error activated.
- 1: Error activated (blocking error).

T Time Out:

- 0: Time Out period observed.
- 1: Time Out period exceeded.

The alarm code contains the code of the error detected by the smart relay *(see page 521)*.

Modbus Identification

The Ethernet extension supports the **Read Device Identification** Modbus function, function code 43 MEI. This function is used to identify a remote device and obtain information about its physical and functional description.

When a query is addressed to the Ethernet extension, it provides the **Basic Device Identification** service by sending the answer explained below (x and y are the indices of the software version of the Ethernet extension:

Identifier	Description	Type	Content
0x00	Name of vendor	ASCII string	TELEMECANIQUE
0x01	Product code	ASCII string	SR3NET01
0x02	Major and minor revision	ASCII string	Vx.y

Requests Specific to TCP Diagnostics

Call State

In call state, the TCP request frame received by the Ethernet extension is as follows:

Field	Size	Content
Function Code	1 byte	0x08
Sub-function	2 bytes	0x0015
Data	2 bytes	0x0003

The Ethernet extension reply frame is as follows:

Field	Size	Content
Function Code	1 byte	0x08
Sub-function	2 bytes	0x0015
Operation statistics	2 bytes	0x0003
Function errors	20 bytes	0 (not provided)
Reception statistics	4 bytes	Frame error during reception
	4 bytes	Capacity overflow error during reception
	8 bytes	0 (not provided)
Transmission statistics	20 bytes	0 (not provided)
	4 bytes	Host device IP address

Reset State

In reset state, the TCP request frame received by the Ethernet extension is as follows:

Field	Size	Content
Function Code	1 byte	0x08
Sub-function	2 bytes	0x0015
Data	2 bytes	0x0004

The Ethernet extension reply frame is as follows:

Field	Size	Content
Function Code	1 byte	0x08
Sub-function	2 bytes	0x0015
Data	2 bytes	0x0004

Part VII
Programming Software Functions

Subject of this Section

This section describes the different functions available in the Programming Software.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
31	Functions	503
32	Description of the Programming Software Menus	527

Chapter 31
Functions

Subject of this Chapter

This chapter describes the different functions available in the Zelio Soft 2 Programming Software.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Program Configuration	504
Preferences of the Programming Software	508
Program Check	509
Write Options Window	510
Program Import	512
Conversion of Applications Created with Previous Version of the Software	514
Setting the Clock Display	515
Saving an Application	516
Printing the Program	517
Page Header and Footer for Application Printing	519
Description of Smart Relay Errors	521
Splitting the Wiring Sheet	523

Program Configuration

Description

The program configuration window allows the different parameters linked to the application to be adjusted.

The window is made up of four tabs that include the following parameters:

- **Properties**:
  - Programmer name
  - Program name
  - Version
  - Comment
- **Configuration**:
  - Cycle time *(see page 505)* of the application in the smart relay
  - WATCHDOG *(see page 103)* action (smart relay cycle time control)
  - Type of filtering *(see page 505)* (hardware) for inputs: Slow/fast
  - Password request during transfer of SR2MEM02 to the smart relay (LD mode)
  - **Zx keys inactive** (LD mode) or **Lock smart relay front panel** (FBD mode) - refer to Program Protection *(see page 505)*
  - Activate MACRO password protection, refer to Password Protection *(see page 426)*.
- **History**: This tab is used to follow modifications in the application. The programmer can save the following information for each modification:
  - Date
  - Programmer name
  - Version
  - Comment
- **Date format**:
  - Date format
  - Activate the summer/winter time change

  The use of these parameters is described in Date Format Tab *(see page 506)*.

Some extensions make specific configuration tabs appear when they are added to the smart relay. Refer to Connection With the Smart Relay *(see page 445)*.

NOTE: Once the type of programming has been determined (**LD** or **FBD**) only the corresponding commands are accessible.

Access

The **Program Configuration** function can be accessed from the **Edit** menu or by using the

 icon in the toolbar.

Cycle Time Duration

Description

A program is represented as a circuit wired with components (the functions).

In order to be executed by the smart relay, this program is translated as a set of ordered instructions, where each instruction corresponds to a function in the program.

This instruction set (functions) is executed periodically, thus at regular time intervals. This set time interval is called the **Cycle time of the application in the smart relay** or the cycle period.

This period corresponds to the sampling period of analog data read at the inputs of the smart relay and its extensions and the refresh period of the outputs of the smart relay and its extensions.

Configuration

The cycle time of the application in the smart relay can be configured in the **Configuration** tab of the **Program Configuration** window. This period may be set from 6 to 90 milliseconds in 2-millisecond steps.

NOTE: Make sure that:
- Input variations that are too rapid are not hidden by a cycle time that is too slow.
- The speed of output variations is compatible with system commands.

NOTE: If the cycle time duration is too short it will be adjusted automatically at program verification stage or when the program is transferred from the PC to the smart relay. However, if the cycle time is too long, it will not be adjusted automatically.

NOTE: For discrete and analog **filtered inputs**, the filtering duration and cut-off frequency are recalculated when the corresponding parameters window is opened. You must then verify if the new values of these parameters are compatible with the input signals variation speed.

See How to debug an application without loading it onto the smart relay: Simulation *(see page 41)*

Hardware Input Filtering

This filtering is different from that of the filtered discrete and analog function blocks; it is part of the program configuration:

- **Slow**: 3 milliseconds
- **Fast**: 0.3 milliseconds

Program Protection

Zx keys inactive (LD mode)

In LD mode, if the **Zx Keys Inactive** box is checked, the Zx keys that are used as pushbuttons will be deactivated. The Zx keys can only be used as navigation keys in the menus that are accessible from the smart relay front panel.

This option has the same effect as an action carried out on the smart relay front panel at the **CONFIGURATION\Zx KEYS** *(see page 102)* level.

Lock smart relay front panel (FBD mode)

In FBD mode, if the **Lock smart relay front panel** box is checked, access to the menus from the smart relay is blocked.

In this case, even the password cannot be used to gain access to the menus from the smart relay front panel.

However:
- The Zx pushbutton functions used in the program remain active.
- Modification of the application parameters or data can be carried out using the **DISPLAY** *(see page 308)* function (provided that the **Authorized Modification** option has been selected).

Date Format Tab

This tab allows configuration of:
- The format in which the date will be displayed, to be chosen from the following 3 possibilities:
  - Day/Month/Year
  - Month/Day/Year
  - Year/Month/Day
- Automatic summer/winter time change: For activating or deactivating the automatic time change and choosing the change dates. See procedure detailed below.

Automatic Time Change

To activate the automatic summer/winter time change, proceed as follows:

Step	Action
1	• Select **Edit → Program Configuration** or • Click on the  icon in the toolbar **Result**: The program configuration window is displayed.
2	Click on the **Date Format** tab.
3	Select the **Activate Summer/Winter Time Change** option.
4	Choose the dates for the time change. There are two ways of doing this: • By using the drop-down list next to the **Zone** parameter, select a **geographic zone** from the following two choices: ○ Europe ○ USA For these two zones, the time change dates are pre-configured and do not require any other adjustment. • By choosing **Other** from the drop down list next to the **Zone** parameter, then by manually specifying the month and the Sunday of the two time changes.
5	Click on the **OK** button.

Password request during transfer of SR2MEM02 to the smart relay

In LD mode, if this box is selected and if a password is defined, then this password will be requested to transfer a SR2MEM02 cartridge to the smart relay using the front panel.

The password is defined in the **Write Options** *(see page 510)* window.

Preferences of the Programming Software

Description

The programming software preferences window is used to configure the general characteristics:

- **Language**: Language used in the Programming Software.
- **Simulation language**: The HMI language of the Programming Software front panel (LCD).
- **Working Directory**: Path of the directory where the applications are saved on the PC (the access path is limited to a maximum of 128 characters, including the program name and its extension).
- **Default Colors**:
  - Of the links in simulation and monitoring mode.
  - Of the background.
  - Of the inputs/outputs of the blocks (Specific FBD/SFC Attribute).
- **No longer display the Beginner dialog box at startup**: If this option is checked, the Programming Software is opened empty (without an application), and you must launch a command from one of the menus.
  By default (option not checked), a preliminary window appears in order to:
  - Create a new program
  - Open an existing program
  - Open a recently used program
  - Upload a program from a smart relay
  - Monitoring mode
  - Exit
- **Display compilation results in simulation mode and when loading**: this option enables the window with the program compilation results *(see page 509)* to be automatically displayed.
- **Show the refresh cycle (simulation and monitoring) and the time between two simulation results** (Simulation *(see page 41)* and Monitoring *(see page 47)* bar): This option can be used to display the drop-down menus of the parameters used to control execution of the application:
  - Refresh Period *(see page 224)*,
  - Number of cycles *(see page 224)*.

NOTE: The **Default Color** button is used to define the characteristics of a new project. To modify the colors of an existing project, use the **Options** → **Modify the Colors** menu.

Access

The **Preferences** function can be accessed from the **File** menu.

Program Check

At a Glance

In LD or FBD mode using the **Edit → Check program** command, you launch the compilation (check) of the program. The result of the compilation is displayed in the **Compilation results** window. In this window, the following information appears:

- The result of the program check,
- Resources used and available.

In FBD mode, the compilation is carried out automatically when you:

- Switch from Edit to Simulation mode,
- Switch from Edit to Monitoring mode,
- Transfer the program to the smart relay.

In LD mode; the compilation is carried out automatically if the **Programmable and Configurable from Front Panel** box in the **Compilation Results** window is checked.

Results Window Elements

The available resources depend on the smart relay type. The compiler calculates the volumes of resources used in the different memory zones of the smart relay.

If the values calculated are greater than the available values, they appear in red.

The following table shows the different elements that are displayed in the **Compilation results** window:

Elements	Description
Parameter Zone	The parameters of the function blocks or automation functions. Two bytes for each integer and 1 byte for the other types.
Digital Data, SFC token, etc.	Data in bit format. One bit per digital or Boolean element or per SFC step bit.
Other data zones	Data in byte format. Two bytes for each integer.
Program zone	The number of bytes corresponding to all of the program function blocks and automation functions.
Application-specific function zone (slots)	The total number of slots for the **AFB** functions *(see page 384)*.
Estimated program time	Sum of the basic execution times for each function used.
Cycle time	Configured **cycle** *(see page 504)* time.

Write Options Window

Description

The **Write options** window appears before the application is transferred to the smart relay: **Transfer → Transfer Program → PC > Module**.

This window is used to:
- Protect the smart relay program.
- Save the modifications carried out in the programming software before the program is written in the smart relay.
- Synchronize smart relay and PC date and time.
- Automatically switch the smart relay to **RUN** mode.
- Automatically switch the programming software to monitoring mode.

Program Protection

Use a password to protect reading and modification of the program written to the smart relay.

If the program is password protected (key icon appears), you must enter the password to perform certain operations.

The password protects access to the following menus (front panel):
- PROGRAMMING (**LD RUN** mode).
- MONITORING (**LD RUN** mode).
- CONFIGURATION (**STOP** mode).
- CLEAR PROG. (**LD STOP** mode).
- MODULE TRANSFER > MEM (**STOP** mode).
- MEM TRANSFER > MODULE (LD **STOP** mode depending on the choice of the programmer, FBD **STOP** mode).

Activating the password can also trigger user restrictions in the programming software:
- Modification of the program contained in the smart relay.
- Refreshing the program contained in the smart relay.
- Overwrite by transfer of another program.
- Monitoring

This option has the same effect as an action carried out on the smart relay front panel at the CONFIGURATION /PASSWORD *(see page 98)* level.

Saving Changes

If the **Save modifications before writing** box is selected, the modifications carried out in the programming software will be automatically saved before the program is written in the smart relay.

Date and Time Synchronization

If the **Synchronize module date and time with PC** box is selected, the time and date of the smart relay will be reset to that of the PC.

Automatic Switching to RUN Mode

If the **RUN mode after loading** box is selected, the smart relay will automatically switch to **RUN** mode at the end of transfer.

Automatic Switching to Monitoring Mode

If the **Monitoring mode after loading** box is selected, the programming software will automatically switch to Monitoring Mode at the end of transfer.

Program Import

At a Glance

The **File** → **Import** command is used to import a program or part of a program of the same reference of smart relay as the target smart relay.

Use this command to open the program containing the function blocks (FBD) or the lines and cells (LD) to import, then use copy-paste to perform the import:

- Select the function blocks, or the lines and cells to import.
- Copy and paste them in the desired location in the program being edited.

After each import:

- In LD, an automatic verification is performed.
- In FDB, you must launch the **Edit** → **Check the program** command.

How to Import Function Blocks and FBD MACROS

Follow the steps below to import function blocks and/or MACROS.

Step	Action
1	From the FBD application, select **File** → **Import**.
2	Choose the file containing the function blocks to import and confirm. **NOTE:** The selected file must contain an FBD application for the import to succeed.
3	Select **Window** → **Tile**. **Result**: The windows of the application in progress and the imported application are displayed one below the other. **NOTE:** If the **Imported** application contains MACROS, they cannot be displayed during this step.
4	In the window of the **Imported** application, select the useful function blocks and/or MACROS.
5	Drag and drop the function blocks and/or MACROS into the window of the application in progress. **NOTE:** If a MACRO was placed in the active window, it can now be opened with the **Display the MACRO** contextual menu.

How to Import LD Cells

Follow the steps below to import LD cells and lines.

Step	Action
1	From the LD application, select **File** → **Import**.
2	Choose the file containing the function blocks to import and confirm. **NOTE:** The selected file must contain an LD application for the import to succeed.
3	Select **Window** → **Tile**. **Result**: The windows of the application in progress and the **Imported** application are displayed one below the other.
4	In the window of the **Imported** application, select the required cells or lines.
5	Drag and drop these cells or lines into the window of the application in progress. **Result**: An automatic verification is performed on the application in progress containing the new cells or lines.

Conversion of Applications Created with Previous Version of the Software

At a Glance

Zelio Soft 2 lets you open and convert applications created using previous versions of the Programming Software.

Procedure

The following table shows how to open a Zelio application created with previous version of Zelio Soft 2:

Step	Action
1	Select the **Open** command from the **File** menu.
2	Select the application.
3	Click **Open** to confirm. **Result**: A window confirming the application conversion appears.
4	Click **OK**.

Setting the Clock Display

At a Glance

See Connection with the Smart Relay/ Set Smart Relay Clock *(see page 459)*.

Saving an Application

At a Glance

When it is saved, the user application and its configuration are stored on the PC:

- Program title
- Programmer name
- Program release version
- Cycle time
- Watchdog parameters
- Implicit input filtering
- Locking of parameters windows
- Locking of Z keys (LD mode)
- Date format
- Summer/Winter commutation dates.
- Information related to the smart relay and the extensions for which the application was designed.
- Release version of configuration components (hardware/firmware).

Access

The save function **Save** or **Save As** can be accessed from the **File** menu.

Printing the Program

At a Glance

Printing an application written in LD or FBD language enables you to create the documentation for the application and it consists of:

- An application diagram.
- Wiring sheet(s) of macro(s).
- The content of the supervision window.
- A table with the following for each symbol:
  - A representation of the symbol.
  - Its chart number.
  - The associated comment.
  - The parameter(s) with their values and their descriptions.
- The program run sheet (option available only when an SR2COM01 communication interface is connected to the smart relay).

The print application diagram function can be used to print the screen in A4 format.

NOTE: Taking into consideration the zoom factor in use at the time of printing:

- In FDB mode, you obtain all or part of the schema, depending on the zoom factor in use.
- In LD mode, the zoom factor in use at the time is not taken into consideration.

Commands

The following table lists the commands available from the **File** menu used for printing:

Command	Description
Print	Used to print the document.
Print preview	Used to preview the print job to verify the result.
Print setup	Opens the print setup window.

Print Options

Various print options are available. They can be configured from the **Print configuration** window:

- **Cover page**
  Cover page print of the program properties defined by the **Files** → **Properties...** command.
- **Edit window**
  - Print area: print all, or print a visible part or a selection in the window.
  - Visible part: print 1 (A4), 2 (A3), 4 (A2) pages or free (print several pages using the current zoom factor).
  - Include the background: Choice of whether or not to print the background of the Edit window.

- **Supervision window**
  - Print area: print all (by default), or print a visible part or a selection in the window.
  - Visible part: print 1 (A4), 2 (A3), 4 (A2) pages or free (print several pages using the current zoom factor).
  - Include the background: Choice of whether or not to print the background of the supervision window.
- **Macro window**: Print the Macro wiring sheet, according to the same print area options as the ones in the edit window, (this option is only available if there is at least one macro and if the macro protection *(see page 426)* is not activated).
- **Summary table**
  Print the function summary table. In LD mode, you can decide to print the I/O, texts or other functions; in FBD mode, you can print either everything or nothing at all.
- **Program run sheet**
  Is used to choose and print a program run sheet.
- **Page Setup** to define how the document is to be presented (**Portrait** or **Landscape**).
- **Headers and footers** *(see page 519)*.

NOTE: in the case of printing a visible part or a selected part, the printed zone is adapted based on the format selected. Thus, some blocks next to the selection (or next to the visible part) may appear during printing.

Print Area Options

The print area options for the Edit, Macro, and Supervision windows accessible by **Menu → Print setup** menu are described in the following table:

Option	Description
All	Prints the entire wiring sheet.
Visible part	Prints the visible part of the screen at the time of printing according to, in FDB mode, the zoom factor. The non-visible part of the screen is printed according to the place it takes.
Selection	Prints the selected objects at the time of printing according to, in FDB mode, the zoom factor. Concerning the unselected objects, they are printed according to the place remaining.
Number of sheets (1, 2 or 4 sheets)	Indicates the number of sheets that will be used to print each diagram.
Includes the background	Prints the background of the wiring sheet.

Page Header and Footer for Application Printing

At a Glance

This function is used to insert the following into the printed application document:

- A logo in bmp format
- Text with:
  - Comments
  - The name of the application file
  - The page numbers and number of pages
  - The time and the date (current, last modification)

The window is divided into 2 sets of 3 white boxes. The upper 3 correspond to the header and the lower 3 to the footer.

Several text items or a logo can be inserted into each of the boxes.

The default contents of each of the 6 boxes are the following:

- **Top left:** The name of the project file followed by the version.
- **Top center:** Schneider Electric logo.
- **Top right:** The project name (provided in the **Properties** tab of the **Program Configuration**) window.
- **Bottom left:** The program author's name (provided in the **Properties** tab of the **Program Configuration**) window.
- **Bottom center:** Date of the last record of the project.
- **Bottom right:** The page number and total number of pages.

The steps for customizing these default values are provided in detail below.

NOTE: A logo and text cannot occupy the same box.

Inserting a Logo

The following table shows the procedure for inserting a logo:

Step	Action
1	Select the **Print Setup** command from the **File** menu. **Result**: the **Print Setup** window appears.
2	Click the **Headers and Footers** button. **Result**: the **Select Headers and Footers** window appears.
3	Position the mouse cursor in one of the upper or lower boxes where to place the logo.
4	Check the **Logo** box.
5	Click the **...** button. **Result**: The **Open** window appears.

Step	Action
6	Select the logo .bmp file.
7	Click **Open** to confirm. **Result**: The file path name appears in the selected box.

Inserting Text

The following table shows the procedure for inserting text:

Step	Action
1	Select the **Print Setup** command from the **File** menu. **Result**: the **Print Setup** window appears.
2	Click the **Headers and Footers** button. **Result**: The **Select Headers and Footers** window appears.
3	Position the mouse cursor in one of the upper or lower boxes where to place the text.
4	Check the **Text** box.
5	Click the icon corresponding to the text to insert: ● #: Page number ● Σ: Number of pages ●  : Date ●  : Time ● C:\: File name **Result**: The inserted text appears between { }.
6	Repeat step 5 to insert another text item in the same box or resume the procedure from step 3.
7	Click **OK** to confirm.

Description of Smart Relay Errors

Description

The **Default Menu** *(see page 115)* of the **smart relays with screen** allows display and release of errors or advisories detected by the firmware (WATCHDOG overflow *(see page 103)*, cycle time too long *(see page 505)*, etc.).

To release an error or advisory in a smart relay without screen, power off and then power on.

Error Codes:

List of errors:

Code	Type of error
00	**No error**
01	**Error in writing to non-volatile memory** This error defines transfer problems between the memory cartridge and the smart relay. If the error occurs frequently, contact your local Schneider Electric support representative.
02	**Error in writing to the clock** If the error occurs frequently, contact your local Schneider Electric support representative.
04	**Overload on transistor outputs** Once a transistor output reaches the threshold for overcurrent detection, the group of 4 outputs to which it belongs is deactivated. To make this group of outputs operational, the cause of the over current (short-circuit, etc.) must first be rectified, and then the error cleared from the DEFAULT menu *(see page 115)*.
50	**The smart relay firmware is corrupted** Reload the firmware on the smart relay. If this error persists, contact your local Schneider Electric support representative.
51	**Watchdog overflow** Advisory or error according to the selection made in the configuration menu (smart relay display) or in the configuration window (Zelio Soft 2 Programming Software). The cycle time in the smart relay is too short compared with the application program execution time programmed in the smart relay. If the application requires a strict sampling of the smart relay inputs/outputs, lengthen the cycle time in the smart relay. To do this, configure the information either in the CONFIGURATION menu (smart relay display) or in the configuration window (Zelio Soft 2 Programming Software). If the application does not require a maximum cycle time, select **No Action for the WATCHDOG**.

Code	Type of error
52	**The smart relay has executed an incorrect operation** If the error is permanent, reload the firmware on the smart relay and the user application. If this error persists, contact your local Schneider Electric support representative.
53	**Link error between smart relay and bus extension** Verify operation of the extension (connection, power supply, error status).
54	**Link error between smart relay and input/output extension** Verify operation of the extension (connection, power supply, error status).
58	**An error is present in the firmware (software specific to the smart relay) or on a part of the smart relay hardware.** If the error is permanent, reload the firmware on the smart relay and the program. If this error persists, contact your local Schneider Electric support representative.
59	**At the beginning of RUN on the smart relay application: the application cannot switch to RUN as it is incompatible with the smart relay physically connected to the supply.** If this error occurs, contact your local Schneider Electric support representative.
60	**At the beginning of RUN on the smart relay application: program incompatible with the bus extension physically connected to the power supply.** If this error occurs, contact your local Schneider Electric support representative.
61	**At the beginning of RUN on the smart relay application: program incompatible with the Input/Output extension physically connected to the power supply.** If this error occurs, contact your local Schneider Electric support representative.
62	**Version (or build number) incompatibility when loading a program from the backup memory** If this error occurs, contact your local Schneider Electric support representative.
63	**Hardware configuration incompatibility when loading a program from the backup memory** If this error occurs, contact your local Schneider Electric support representative.

Splitting the Wiring Sheet

Introduction

The wiring sheet may be split in two. The split display is used to display two distinct parts of the wiring sheet on the same screen.

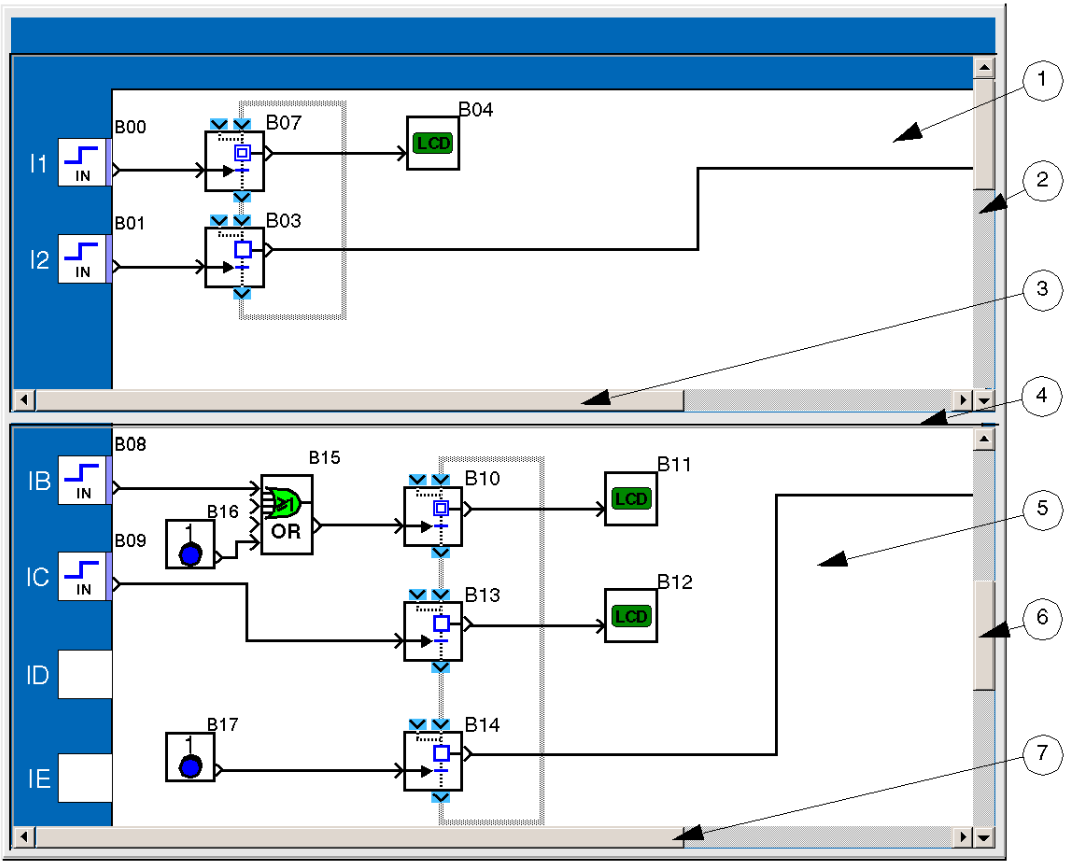
How to Split the Display

To split the display, proceed as follows:

Step	Action
1	Select the **Window → Split Display** menu.
2	Click on the place where to split the display. **Result:** The wiring sheet will split into two displays.

Structure of the Split Wiring Sheet

The split wiring sheet is structured as follows:



The elements of the split wiring sheet are described below:

Number	Element
1	Upper section display.
2	Upper section vertical scroll bar.
3	Upper section horizontal scroll bar.
4	Splitting bar.
5	Lower section display.
6	Lower section vertical scroll bar.
7	Lower section horizontal scroll bar.

Using the Split Wiring Sheet

The split wiring sheet may be used to perform the following actions:

To...	Action
To make desired function blocks appear in the upper section	Scroll the upper section scroll bars.
To make desired function blocks appear in the lower section	Scroll the lower section scroll bars.
To move the splitting bar to the desired location	Click on the location.
To connect the upper and lower section function blocks	● Click (and hold) on the start block output ● Drag to the target block input, if necessary, cross the splitting bar ● Release.

How to Cancel Split Display

To cancel the split display, select the **Window → Cancel Split** menu.

Result:

- The lower display disappears.
- The upper display will display the entire wiring sheet.

NOTE: To cancel the split display, you can also click on the splitting bar and drag it to the scroll bar of the lower display or to the title block.

Chapter 32
Description of the Programming Software Menus

Description of the Programming Software Menus

Description

Description of the Programming Software menus:

- File *(see page 527)*
- Edit *(see page 528)*
- Mode *(see page 529)*
- Module *(see page 529)*
- Zelio2 COM *(see page 529)*
- Transfer *(see page 530)*
- Options *(see page 530)*
- Display *(see page 531)*
- Directories *(see page 531)*
- Tools *(see page 532)* (FBD)
- Draw *(see page 532)* (FBD)
- Window *(see page 533)* (LD)
- Simulation *(see page 533)*
- ? *(see page 533)*

File Menu

Description of commands in the File menu:

Command	Description
New *(see page 26)*	Create a new project.
Open	Open an existing project.
Close	Close the project being edited **(*)**.
Save *(see page 516)*	Save the project being edited **(*)**.
Save As *(see page 516)*	Save the project being edited under another name **(*)**.
Print... *(see page 517)*	Prints the project **(*)**.
Print preview *(see page 517)*	To view the project as it will appear when printed **(*)**.
&Print setup... *(see page 517)*	Configures the print characteristics of the project **(*)**.
(*) Only available if a project file is open in the Programming Software.	

Command	Description
Import... *(see page 512)*	Imports the edit window of another project **(*)**. **NOTE:** Only programs or parts of programs of the same smart relay reference as the target smart relay can be imported.
Preferences... *(see page 508)*	Configures the general characteristics of the Programming Software.
No. file_name.zm2	List of files recently opened.
Exit	Closes the Programming Software.
(*) Only available if a project file is open in the Programming Software.	

Edit Menu

Description of commands in the Edit menu:

Command	Description
Cancel	Cancels the last operation carried out (50 cancellation levels).
Cut	Copies and deletes the selected element (placed in the clipboard).
Copy	Copies the selected elements to the clipboard.
Paste	Pastes the element from the clipboard.
Insert line *(see page 218)*	Inserts a line in the wiring sheet **(*)**.
Delete line *(see page 218)*	Deletes a line in the wiring sheet **(*)**.
Ladder entry *(see page 136)*	Programming from the editor **(*)**.
Zelio entry *(see page 135)*	Programming from the front panel of the smart relay**(*)**.
Settings *(see page 138)*	Displays the Parameters window.
Text Entry *(see page 139)*	Displays the text entry window.
Clear	Clears the contents of the selected boxes.
Select All	Selects the entire wiring sheet.
Search	Searches for a function in the program using its name or an associated comment.
Find Item	Lists the functions used in the project **(*)**.
Program configuration *(see page 504)*	Used to set the different parameters linked to the application.
Check the Program *(see page 509)*	Checks program consistency.
(*) Available only in **LD** mode.	

Mode Menu

Description of commands in the Mode menu:

Command	Description
Edit *(see page 25)*	Used to construct programs in LD or FBD mode, which corresponds to development of the application.
Monitoring *(see page 25)*	The program is executed on the smart relay; the Programming Software is connected to the smart relay.
Simulation *(see page 25)*	The program is executed offline in the Programming Software (simulated on the PC).

Module Menu

Description of Module menu commands:

Command	Description
Choice of Smart Relay/Programming Type *(see page 26)*	Choice of smart relay type with its associated functions and connected extensions.
Smart Relay Diagnostics *(see page 456)*	Allows you to view the smart relay diagnostics.
Set Clock *(see page 459)*	Allows you to set the smart relay clock.
Update Smart Relay FIRMWARE *(see page 461)*	Used to load a new version of the firmware into the smart relay.
Smart Relay Language *(see page 460)*	This function is used to change the smart relay interface language.

Zelio2 COM Menu

Description of commands in the Zelio2 COM menu:

Command	Description
Zelio2 COM Diagnostics *(see page 468)*	Allows you to view the smart relay communication diagnostics.
Update Zelio2 COM Firmware *(see page 468)*	Used to load a new version of the firmware into the SR2COM01 communication interface.
Message ON *(see page 469)*	Used to activate the SR2COM01 communication interface.
Message OFF *(see page 469)*	Used to deactivate the SR2COM01 communication interface.

Transfer Menu

Description of commands in the Transfer menu:

Command	Description
Transfer Program	**PC -> Smart Relay** *(see page 450)* Transfers the application from the PC to the smart relay. **Smart Relay-> PC** *(see page 452)* Transfers the application from the smart relay to the PC.
RUN Module *(see page 454)*	Initializes and starts the program.
RUN Module Without Init of saved data *(see page 454)*	Starts the program without initializing the current values of functions for which the Latching options is activated.
Stop Module *(see page 454)*	Stops the program.
Compare the Program With Smart Relay Data *(see page 455)*	Compares the data contained in the smart relay and the data produced by compiling the Programming Software application.
Clear the Program *(see page 458)*	Clears the program and erases the data in the smart relay.
Remote Control of Front Panel	Used to remotely control a smart relay connected to the PC. **RUN**: Starts the program **STOP**: Stops the program
COMMUNICATION configuration *(see page 448)*	Configures the communication (serial link) between the Programming Software and the smart relay.
Connect	Used to connect to the remote station using a modem link.
Disconnect	Used to disconnect from the remote station in the case of a modem link.

Options Menu

Description of commands in the Options menu:

Command	Description
Modify the Colors *(see page 395)*	Used to define different colors to display in the Programming Software.

Display Menu

Description of commands in the Display menu:

Command	Description
Status bar	Shows or hides the status bar dialog box (at the bottom of the Programming Software window).
Comments	Used to show/hide the program comments (the comments are displayed under the function block) **(**)**.
Block Number	Displays/hides the function block numbers **(**)**.
Grid	Shows/hides the wiring sheet grid **(**)**.
Zoom (Y)	Configures the zoom factor of the wiring sheet (25 to 150%).
Ladder Symbols	Program display in Ladder symbols **(*)**.
Electrical Symbols	Program display in electrical symbols **(*)**.
(*) Available only in **LD** mode. **(**)** Available only in **FBD** mode.	

Directories Menu

Description of commands in the Directories menu:

Command	Description
Directory of Remote Stations... *(see page 470)*	Allows configuration of telephone numbers.
Program Recipients Directory... *(see page 473)*	Allows configuration of telephone numbers, and access rights or recipients associated with a program.
Recipients general directory... *(see page 474)*	Used to create a list of recipients regularly used when writing programs.

Tools Menu

Description of commands in the Tools menu (FBD-specific):

Command	Description
Align *(see page 416)*	Positions the objects in relation to others: • Left • Right • Top • Bottom • Center vertically • Center horizontally
Distribute	Distribute the objects: • Horizontally • Vertically
Order *(see page 416)*	Positions the objects in relation to others: • Bring to front • Send to back
Group *(see page 413)*	Creates composite objects.
Ungroup *(see page 413)*	Ungroups composite objects.
Renumber the functions	Used to reassign the numbers of consecutive blocks starting from number B00.
Renumber the links	Used to reassign numbers of consecutive links.
Wiring Mode *(see page 399)*	Used to change the type of link between the function blocks: • Text • Wiring (This option specifies the type for new links.)
Type of Wiring *(see page 399)*	Used to change the type of link between the function blocks: • Text • Wiring • Modify the text (This option specifies the type only for the active link.)

In the edit and supervision sheet, you can create squares, ellipses, lines or text. You can also insert an image in Bitmap format.

Window Menu

Description of commands in the Window menu:

Command	Description
Cascade	Organizes the windows in a cascade.
Tile	Tiles the windows horizontally.
Arrange icons	Rearranges the windows.
Front panel	Activates the Front panel window.
Supervision	Activates the Supervision window.
Edit	Activates the Edit window.

Simulation Menu

Description of commands in the Simulation menu (specific to simulation mode):

Command	Description
Init and Run	Initializes and launches the program.
Pause	Pause/Run: stops or relaunches program execution.
Stop	Stops the program.
Power Failure	Simulates a power outage.
Power restore	Simulates power restoration.
Set clock *(see page 459)*	Allows you to set the smart relay clock.
Simulation Language	Configures the HMI language of the Programming Software front panel (LCD).

Menu ?

Description of commands in the ? menu:

Command	Description
About the Programming Software	Displays the release version of the Zelio Logic 2 Programming Software and its components.
Help	Provides access to online help.

Appendices

Appendix A
Compatibility

Subject of this Chapter

This appendix provides information on the compatibility between the versions of the firmware, the versions of the programming software, the available functions and the different memory cartridges.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Compatibility between the version of Zelio Soft 2 software and the version of the firmware on the smart relay	538
Zelio Soft 2 Software Version Versus Functions	540
Compatibility between the memory cartridges and the version of the firmware on the smart relay	541

Compatibility between the version of Zelio Soft 2 software and the version of the firmware on the smart relay

Introduction

The section below describes the compatibility between the versions of the programming software and the versions of the firmware on the smart relay.

In the Case of a Transfer of the Program from the PC to the Smart Relay

In the case of a transfer of the PC program to the smart relay, all the versions of the programming software are compatible with all the versions of the firmware on the smart relay.

During transfer of the PC program to the smart relay, the firmware associated with the version of the programming software is transferred to the smart relay.

In the Case of a Transfer of the Program from the Smart Relay to the PC

In the case of a transfer of the program from the smart relay to the PC, compatibility between the version of the programming software and the version of the firmware on the smart relay is as follows:

		Zelio Soft 2 software version												
Firmware version	Language	2.0	2.1	2.2	2.4	3.1	4.1	4.2	4.3	4.4	4.5	4.6	5.0	5.1
2.16	LD, FBD	X	X	X	X	–	–	–	–	–	–	–	–	–
2.17	LD, FBD	X	X	X	X	–	–	–	–	–	–	–	–	–
2.18	FBD	X	X	X	X	–	–	–	–	–	–	–	–	–
2.19	LD	–	–	–	X	–	–	–	–	–	–	–	–	–
3.09	LD, FBD	–	–	–	–	X	–	–	–	–	–	–	–	–
4.01	LD, FBD	–	–	–	–	–	X	X	X	X	X	X	–	–
4.02	LD, FBD	–	–	–	–	–	–	–	X	X	X	X	–	–
4.03	LD, FBD	–	–	–	–	–	–	–	–	X	X	X	–	–
4.04	LD, FBD	–	–	–	–	–	–	–	–	X	X	X	–	–
4.05	FBD	–	–	–	–	–	–	–	–	X	X	X	–	–
4.05	LD	–	–	–	–	–	–	–	–	X	X	X	X	X
4.07	FBD	–	–	–	–	–	–	–	–	X	X	X	X	X
X: Compatible –: Not compatible														

		Zelio Soft 2 software version												
Firmware version	Language	2.0	2.1	2.2	2.4	3.1	4.1	4.2	4.3	4.4	4.5	4.6	5.0	5.1
4.09	LD	–	–	–	–	–	–	–	–	–	–	–	X	X
4.1•	LD, FBD	–	–	–	–	–	–	–	–	–	–	–	–	X
X: Compatible –: Not compatible														

For more information on how to check the firmware version, refer to the **VERSION** Menu *(see page 111)*.

Zelio Soft 2 Software Version Versus Functions

The following table shows the functions available depending on the Zelio Soft 2 software version.

Functions	Zelio Soft 2 software version				
	V2.xx	V3.xx	V4.xx	V5.0	V5.1
LD language					
Maximum number of program lines	–	–	120	240 (1)	240 (1)
Number of auxiliary relays	–	–	28	56 (2)	56 (2)
Number of counters	–	–	16	28 (3)	28 (3)
Number of clocks	–	–	8	8	8
Number of timers	–	–	16	28 (4)	28 (4)
Number of text blocks	–	–	16	16	16
Number of messages	–	–	28	28	28
FBD language					
Maximum number of function blocks	–	–	255	500	500
Logic functions	Yes	Yes	Yes	Yes	Yes
Standard functions except Sunrise/Sunset and Suntrack	Yes	Yes	Yes	Yes	Yes
Sunrise/Sunset	–	–	Yes	Yes	Yes
Suntrack	–	–	Yes	Yes	Yes
SFC functions	–	–	Yes	Yes	Yes
Application functions (PID)	–	–	–	–	Yes
(1) Only if there is no SR2COM01 module in the configuration. Otherwise, the maximum number of lines is 120. (2) Only if there is no SR2COM01 module in the configuration. Otherwise, the maximum number of auxiliary relays is 28. (3) Only if there is no SR2COM01 module in the configuration. Otherwise, the maximum number of counters is 16. (4) Only if there is no SR2COM01 module in the configuration. Otherwise, the maximum number of timers is 16.					

For more information on how to check the firmware version, refer to the **VERSION** Menu *(see page 111)*.

Compatibility between the memory cartridges and the version of the firmware on the smart relay

Introduction

The section below describes the compatibility between the memory cartridges and the versions of the firmware on the smart relay.

Compatibility of the Memory Cartridge with the Version of the firmware

The table below describes the compatibility of the memory cartridges with the version of the firmware:

Type of memory cartridge	Version of firmware compatible
SR2MEM01	LD Language: V2.19 or lower. FBD Language: V2.18 or lower.
SR2MEM02	V3.09 or higher.

Transferring a Program from the SR2MEM01 Memory Cartridge to the Smart Relay

In the case of a transfer of the program from the SR2MEM01 memory cartridge to the smart relay, compatibility is as follows:

		Smart relay firmware language	
		LD	FBD
Memory cartridge program language	LD	Compatible if the versions of the memory cartridge and smart relay match.	The LD firmware version must be transferred to the smart relay.
	FBD	The LD firmware version must be transferred to the smart relay.	Compatible if the versions of the memory cartridge and smart relay match.

Transferring a Program from the SR2MEM02 Memory Cartridge to the Smart Relay

In the case of a transfer of the program from the SR2MEM02 memory cartridge to the smart relay, compatibility depends upon the firmware version of the smart relay that the program was loaded from, and the hardware version of the smart relay that the program is being transferred to:

- If the memory cartridge was loaded from a smart relay using firmware version 4.04 or lower, then transfer is not compatible to smart relays with hardware version 1.0.08 or higher.
- If the memory cartridge was loaded from a smart relay using firmware version 4.05 or higher, then transfer is compatible with all smart relays.

For more information on how to check the firmware or hardware version, refer to the **VERSION** Menu *(see page 111)*.

Glossary

A

Application

Program

D

Diagram

Program diagram in the program window.

Drag/Drop

Operation involving clicking on the left mouse button then moving the mouse while holding down the left button, before releasing it at the required position on the screen.

F

FBD

Functional Block Diagram.

FDR

The FDR (Fast Device Replacement) service is used when a replacement device is physically connected to the network. The service enables the system (including the device) to:

- Provide the replacement device with the same IP address as the previous one,
- Help ensure that the replacement device is functionally compatible with the previous one,
- Restore the application parameters of the replaced device.

G

Gateway

Equipment that links networks with different architectures and which functions on the application layer. This term may refer to a router.

H

HMI Software

Human Machine Interface of the programming software executed on a PC.

L

LCD Display
Screen located on the unit of certain modules whose keys can be operated to provide autonomous use of the smart relay (control, settings, surveillance, and, in LD mode only, programming and monitoring).

LD
Ladder Diagram.

M

MAC address
Media Access Control. Unique (worldwide) hardware address of a network card or peripheral coded on 6 bytes. It is assigned by the device manufacturer.

MACRO
A MACRO is a group of function blocks. It is characterized by its number, name, links, internal function blocks and input/output connections.

MEI
The MEI (Modbus Encapsulated Interface) is an mechanism for encapsulating service requests and invocation methods, as well as their replies, in the Modbus frame.

Module
General name given to differentiate between the different types of Zelio Logic smart relays.

Monitoring
Action used to scan the data and parameters modified in the smart relay from the software on a PC (online mode) or on the LCD of the smart relay (in LD mode only).

P

Program
See application.

R

Recipient
Depending on the situation, a recipient may be a mobile phone, a PC equipped with **Zelio Logic Alarm**, or an electronic mailbox that can receive alarm messages sent by the remote station and/or can control the remote station.

Remote station
This consists of a smart relay and a SR2COM01 communication interface.

S

SFC

Sequential Function Chart, programming mode similar to GRAFCET.

Supervision

Term characterizing the HMI software window displaying the program data and parameters scanned during a simulation or monitoring phase.

U

User Guide

Text file generated by the programming software during program transfer to the remote station. This file contains the information concerning program alarm messages for this remote station and access possibilities.

W

Wiring sheet

Work surface of the Edit window:

- Includes the input and output contacts for an application in FBD mode,
- includes columns for the contacts and a column for the coils of an application in LD mode.

Z

Zelio Logic

Automation smart relay, also called logic smart relay.

Index

Symbols

Communication
  Configuration, *448*

0-9

10-Bit Analog Output, *266*
10-Bit FBD Integer Input, *260*

A

Access/Modification Protection of the Program Saved on the Smart Relay, *457*
ADD/SUB Arithmetic Function, *320*
Analog FDB Input, *254*
Analog Inputs-Outputs, *485*
ARCHIVE Function, *327*
Auxiliary Relays, *151*

C

CAM BLOCK Cam Programmer, *324*
CAN FBD Word to Bit Conversion, *332*
Clear the Program Contained in the Smart Relay, *458*
Clock, *515*
Clocks, *176*
CNA FBD Bit to Word Conversion, *331*
Comment
  Ladder Network, *127*
Communication Setup, *448*
Compare the Smart Relay Data with the Program, *455*
Compatibility
  Firmware, *538*, *541*
  Memory Cartridges, *541*
  Programming Software, *538*
Configuring the Smart Relay Language, *460*
Creating an application, *26*

D

Description of the Programming Software Menus, *527*
Directories, *470*
  General Directory of Recipients, *474*
  Program recipients directory, *473*
  Remote Stations Directory, *470*
Discrete FBD Inputs, *251*
Discrete FBD Outputs, *262*
Discrete Inputs, *147*
Discrete Outputs, *144*

E

Error Codes of the Communication Interface, *483*
Ethernet
  Communication, *494*
  Ethernet Inputs-Outputs, *269*
  Gateway Address, *496*
  IP address, *495*
  Reserved Address, *495*
  Subnet Mask, *495*
  Time Out, *495*
example of a Ladder application , *237*
example of an FBD application, *439*

F

FBD application example
  greenhouse ventilation panes, *439*
FBD Ethernet Inputs-Outputs, *269*
FBD Fast Counter, *281*
FBD Integer Type Input, *257*

FBD Language
  Draw, *408*
  Edit Window, *244*
  Forcing, *436*
  Function Bar, *247*
  Monitoring, *434*
  Positioning Objects, *416*
  Simulation, *429*
  Zoom, *407*
FBD Language Elements
  10-Bit Integer Input, *260*
  10-Bit Integer Output, *266*
  ADD/SUB Arithmetic Function, *320*
  AFB, *385*
  Analog Input, *254*
  ARCHIVE Function, *327*
  Boolean Equation, *274*
  CAM BLOCK Cam Programmer, *324*
FBD language elements
  CAN Word to Bit Conversion, *332*
  CNA Bit to Word Conversion, *331*
FBD Language Elements
  Comparison, *300*, *305*
  Counter, *276*
  Discrete Inputs, *251*
  Discrete Outputs, *262*
  Down Counter, *276*
  Ethernet Inputs-Outputs, *269*
  Fast Counter, *281*
  Filtered Analog Input, *256*
  Filtered Discrete Input, *253*
  Gain, *306*
  Hour Counter, *302*
  Impulse Relay, *318*
  Integer Type Input, *257*
  Integer Type Output, *264*
  LCD DISPLAY Screen Display, *308*
  Logical functions, *270*
  Message, *337*
FBD language elements
  Modbus Inputs-Outputs, *267*
FBD Language Elements
  MUL/DIV Arithmetic Function, *322*
  Multiplexing, *319*
  PID, *386*
  PID Function, *389*
  Pulses, *293*, *294*
  RESET, *275*
  Resettable Initial SFC Step, *377*
  SCHMITT TRIGGER, *303*
  Serial Link, *333*, *335*
  SET, *275*
  SFC, *349*
  SFC Convergence to AND, *380*
  SFC Convergence to OR, *382*
  SFC Divergence to AND, *379*
  SFC Divergence to OR, *381*
  SFC Initial Step, *376*
  SFC Initialization, *371*
  SFC Step, *378*
  Special Inputs, *258*
  STATUS, *329*
  Sunrise/Sunset, *344*
  Suntrack, *346*
  TEXT, *311*
  TIME PROG Programmer, *314*
  Timer B/H, *297*
  Timer Block, *289*
  Up/Down Counter, *287*, *287*
FBD Special Inputs
  Button, *258*
  Discrete Constants, *258*
  Flashing Input, *259*
  Numerical Constants, *258*
  Summer Time Input, *259*
FBD TEXT, *311*
FDB Integer Output, *264*
FDB Language Elements
  LCD Screen Backlighting, *265*
Filtered Analog FBD Input, *256*
Filtered Discrete FBD Input, *253*
Find, *410*
Firmware, *65*
  Compatibility, *538*, *541*

Free Entry
Automation Function Configuration, *216*
Insert/Delete a Program Line, *218*
Free Mode
Check Program Consistency, *220*
Copy Parts of a Program, *219*
Enter a Contact or a Coil, *213*
Enter a Link, *215*
Function Blocks
Comment, *407*
Cut, Copy or Paste, *415*
Delete, *414*
Duplicate, *414*
Group, *413*
Insertion, *397*
Links, *399*
Parameters, *405*
Selection, *412*
Ungroup, *413*

G

Gain, *306*
Getting started
Checking the Smart Relay Firmware, *65*
Getting Started
Behavior of the Smart Relay in the Event of Power Outage, *68*
Checking the Program, *61*
Configure an Application from the Front Panel of the Smart Relay, *58*
Configuring the Language of the Programming Software and the Smart Relay, *67*
Connecting the Programming Software to the Smart Relay, *51*
Controlling the Smart Relay from the Front Panel, *54*
Controlling the Smart Relay from the Programming Software, *53*
Create a New Program, *32*
Debugging an Application without Loading it onto the Smart Relay: Simulation , *37*
Diagnosing the Smart Relay State, *52*
Getting started
Importing an application developed using Zelio Soft 1 into Zelio Soft 2, *70*
Getting Started
Meaning of the Error Code Displayed on the Front Panel of the Smart Relay, *49*
Modifying Program Data Using the Smart Relay Front Panel, *59*
Monitor and Modify an Application Running on the Smart Relay from the Programming Software: Monitoring , *43*
Monitoring and Modifying an Application Using the Smart Relay Front Panel, *48*
Programming an Application from the Smart Relay Front Panel, *34*
Programming an Application Using the Programming Software, *33*
Transferring the Program from the PC to the Smart Relay, *35*
Graphic Elements, *128*

I

Import, *512*
Impulse Relay, *318*
inputs-outputs, *79*

L

Ladder application example
greenhouse ventilation panes, *237*
Ladder Language, *124*
Ladder Network, *125*
LCD DISPLAY Screen Display, *308*
LCD Screen Backlighting Outputs, *265*
LD
Structuring LD Programs, *134*
LD Analog Comparator, *192*
LD Change to Summer / Winter Time, *180*
LD Configuration Mode, *138*
LD Counter Comparators, *164*
LD Counters, *157*

LD Entry Type
Configuration Mode, *138*
Ladder Entry Type, *136*
Text Entry Mode, *139*
Zelio Entry Mode, *135*
LD Fast Counter, *166*
LD Ladder Entry Mode, *136*
LD language Elements
Analog Comparator, *192*
LD Language Elements
Auxiliary Relays, *151*
LD language Elements
Change to Summer / Winter Time, *180*
LD Language Elements
Clocks, *176*
Counter Comparators, *164*
Counters, *157*
Discrete Inputs, *147*
Discrete Outputs, *144*
Fast Counter, *166*
LCD Screen Backlighting, *202*
Message, *203*
Modbus I/Os, *149*
TEXT, *198*
Timers, *182*
Zx Keys, *155*
LD LCD Screen Backlighting, *202*
LD Monitoring of an Application, *231*
LD Simulation of an Application, *223*
LD TEXT, *198*
LD Text Entry Mode, *139*
LD Timers, *182*
Logic Function
AND, *270*, *270*
EXCLUSIVE OR, *270*
NAND, *270*
NO, *270*
NO AND, *270*
NO OR, *270*
NOR, *270*
NOT, *270*
OR, *270*, *270*
XOR, *270*
Logical FBD Functions, *270*

M

MACRO, *418*
Create a MACRO, *420*
Front Panel Display, *427*
Links, *418*
Modify a MACRO, *423*
Modifying an instance, *426*
Modifying MACRO Comments, *426*
Password Protection, *426*
Menu
CHANGE DATE/TIME, *117*
CHANGE SUMMER/WINTER, *119*
CLEAR PROG., *105*
CONFIGURATION, *97*
DEFAULT, *115*
LANGUAGE, *113*
MONITORING, *93*
PARAMETERS, *91*
PASSWORD, *98*
PROGRAMMING, *81*
RUN / STOP, *95*
TRANSFER, *107*
VERSION, *111*
WATCHDOG CYCLE, *103*
Zx KEYS, *102*
Message OFF, *469*
Message ON, *469*
Modbus LD I/Os, *149*
Modbus LD Inputs-Outputs, *267*
Monitoring
Discrete Inputs, *233*
Discrete Outputs, *234*, *235*
Monitoring of an Application, *231*
Z Keys, *235*
MUL/DIV Arithmetic Function, *322*
Multiplexing, *319*

P

Preferences of the Programming Software, *508*
Printing, *517*
Footer, *519*
Header, *519*
Program Check, *509*

Program Configuration, *504*
Programming
    Ladder Network, *130*
Pulses on Edges, *293*

R
RUN/STOP Program Run Commands, *454*

S
Save, *516*
Set Smart Relay Clock, *459*
SFC Advisories, *383*
SFC Errors, *383*
Simulation
    Analog Inputs, *229*
    Automation Function Parameters, *225*
    Clocks, *229*, *432*
    Discrete Inputs, *226*
    Discrete Outputs, *227*, *228*
    Simulation of an Application, *223*
    Zx Keys, *228*
Smart Relay Diagnostics, *456*
Smart Relay Error Codes, *521*
Split Display, *523*
Starting out
    Using the memory cartridge, *66*

T
TIME PROG Programmer, *314*
Timer A/C, *289*
Transfer the PC program to the Smart Relay, *450*
Transfer the Smart Relay program to the PC, *452*

U
Update Smart Relay FIRMWARE, *461*
Update the Zelio2 COM Firmware, *468*

W
Write Options Window, *510*

Z
Zelio LD Entry Mode, *135*
Zelio2 COM Diagnostics, *468*
Zelio2 COM Interface Configuration, *476*
Zelio2 COM Menu, *468*
Zx Keys, *155*