

4.2 LEARNING IN LADDER LANGUAGE

4.2.1 Getting started

4.2.1.1 The simplicity of Ladder language

Zelio Logic is programmable in Ladder language. This type of programming allows you to carry out combinational logic functions. In this way, you may program your applications with Zelio Soft 2 or from its integrated programming screen and keyboard.

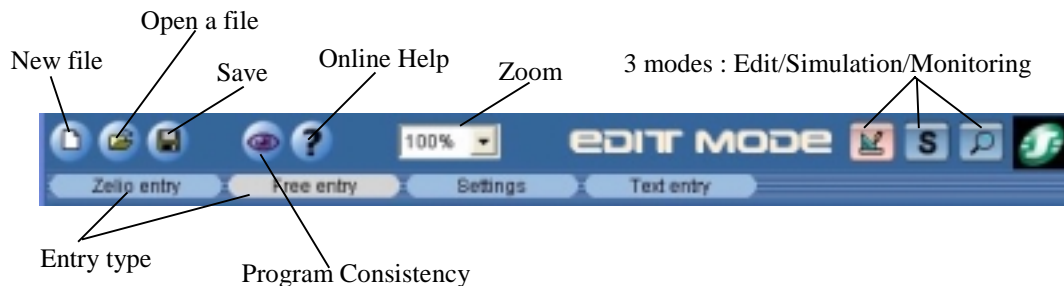
4.2.1.2 Accessing Zelio Soft help

There is help available in the menu bar of Zelio Soft 2 that you may access by clicking on menu ? then on **Help**, or by clicking directly on the ? icon available in the tool bar. To directly access Help concerning a function in use, click on ? in the function's parameter window (which you may access by double-clicking on the function)

4.2.1.3 Toolbar

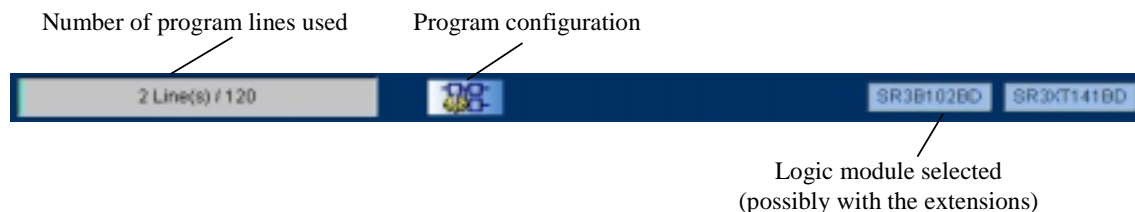
The toolbar contains shortcuts to menu options and offers a **Program coherence** function that is more highly developed. It also allows you to choose the **mode**: Editing, Simulating or Monitoring. Lastly, it offers you 2 types of input mode: Zelio input mode (front panel of smart relay) and freestyle input mode (electric diagram or Ladder diagram)

Hover the mouse arrow over any button to see the action associated with it.



4.2.1.4 Number of lines used and the module chosen

A bar at the bottom of the screen indicates the chosen number of lines of the command diagram as well as the type of smart relay chosen by the program, and the possible extensions. This bar also contains the "Program configuration" icon which allows the different parameters linked to the application to be adjusted.



4.2.2 Writing a program in Ladder language

4.2.2.1 Modes and types of entering

When you have chosen your module and Ladder language, you are ready to build your application.

The selected Zelio Logic reference appears in the lower right (1):



With the software program, you may choose to program in **Manual Data Entry** or in **Zelio Data Entry**.

The default is **Manual Data Entry**: A wiring sheet limiting the areas reserved for the contacts and for the coils (one only at the end of each line) appears on the screen.

Zelio data entry is identical to keyboard data entry in integrated programming. Therefore, the instructions for this type of data entry are the same as for the front panel programming. To select this data entry, click on the corresponding tab (1):



When you are in **Manual data entry**, you may visualize the this diagram in **Ladder symbols** or **Electric symbols** by selecting the desired symbol in the **View** menu.

This program has three modes: **Edition mode** (1), **Simulation mode** (2) and **Monitoring mode** (3). They may be selected in the **Mode** menu or from the toolbar in the upper right. The selected mode appears to the left of the 3 icons (4):



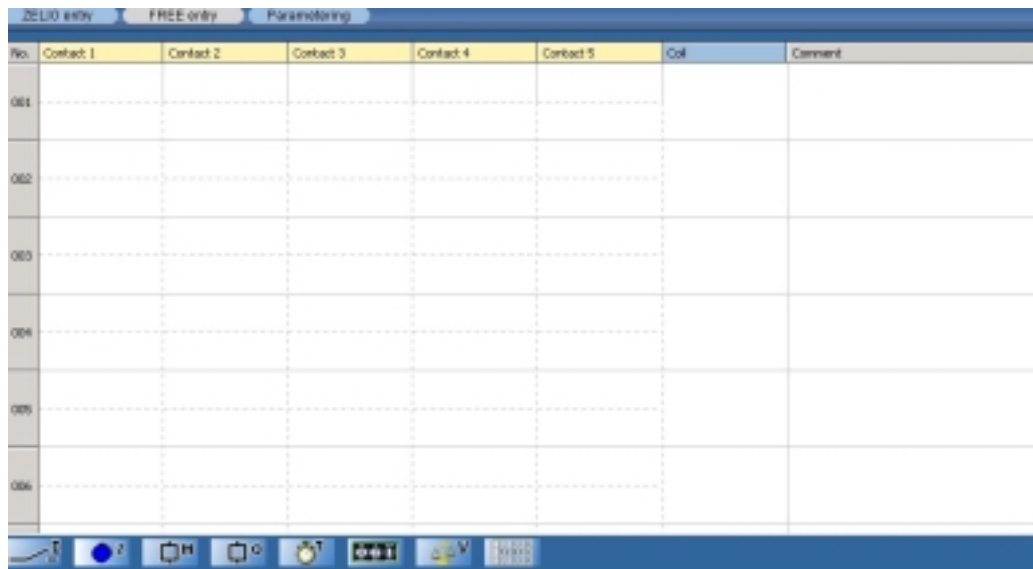
Edition mode is the mode in which you may edit the program and the supervision window. This is the default mode. **Simulation mode** allows you to simulate the program before transferring it to the module. **Monitoring mode** enables you to visualize the input and output statuses of the module in real time.

A **Supervision window** is available for Simulation mode and Monitoring mode. This window offers you the possibility of visualizing the inputs/outputs that you have previously chosen and placed. This allows you to see the essential of the program to ensure an efficient tracking. Drawing functions enable you to illustrate the application.

4.2.2.2 Edit Mode: Programming the application

Entering a program on the wiring sheet

When you have selected your module type and Ladder language, a wiring sheet then appears:



The default is **Edition mode Manual Data Entry**: The diagram is divided in columns, which allow you to distinguish the type of block to be placed. The first five columns are reserved for the contacts (yellow), the sixth allows you to place the output coil (blue). The last column is reserved for entering commentaries associated with each line. The dotted lines are lines where it is possible to wire in order to link functions with each other and to carry out the elementary logic functions **ET** and **OU**

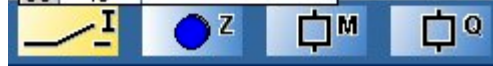
To create a block on the sheet, choose the block type by pointing to the corresponding icon at the bottom of the sheet:



- | | |
|------------------------|-------------------------------------|
| (1) Discrete Input I | (7) Counter comparator |
| (2) Front panel button | (8) Analog comparator |
| (3) Auxiliary relay M | (9) Weekly clock |
| (4) Output Q | (10) Display |
| (5) Timer | (11) Backlighting |
| (6) Counter | (12) Daylight Savings Summer/Winter |

The list of available elements is displayed when you point to each icon:

No.		Comment
01	I1	
02	I2	
03	I3	
04	I4	
05	I5	
06	I6	



The comment cell allows you to associate a name with each element (double click on the zone)

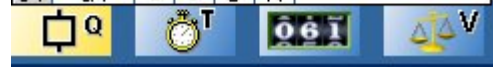
Click and drag the blocks to put them into place on the wiring sheet. The ☹ symbol appears when it is impossible to place the block in a given zone.

For example, if you have just clicked and dragged **12** to place it on the wiring sheet, the ☹ symbol will appear when you try to place it as a coil, indicating that it may only be placed as a contact (also indicated by a color code)

Continue placing the other blocks in the same manner. To wire the reciprocal function (for example **i1** for the reciprocal input **I1**), either select the block by clicking on it (yellow background will appear) and press the space bar, or right click on the mouse and select the reciprocal function. Connections are made by clicking on the dotted line cells that you wish to wire.

The charts corresponding to functions containing different types of inputs/outputs appear as follows:

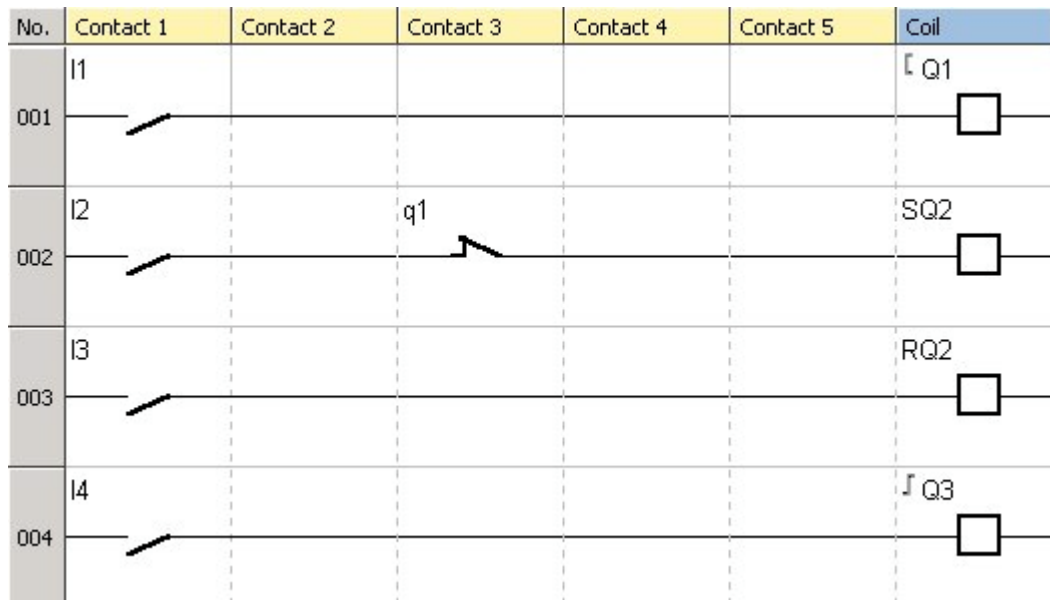
No.					Comment
01	Q1	L	J	S	R
02	Q2	L	J	S	R
03	Q3	L	J	S	R
04	Q4	L	J	S	R



The different input/output-type possibilities appear in the chart. When one of them has been placed and it can be used only once (example: coil reset **RQ2**), the corresponding cell is grayed in and it is impossible to use it again.

Note: On the wiring sheet, it is possible to review the different types of output coils by clicking on the box (yellow background will appear when selected) and pressing the space bar.

For example:



In this example; the 3 types of coils are used: Contactor **Q**, Set/Reset **S/R** and Auto Relay **α**.

Q1 recopies input **I1** status. As for **Q2**, it cannot be activated unless **I2** changes to high status while the **Q1** coil is at rest (**q1** reciprocal function of **Q1**). Press **I3** to deactivate . Finally, **I4** controls impulse relay coil **Q3**.

Click on the link below to access the example:

[\(Ex 01\)](#)

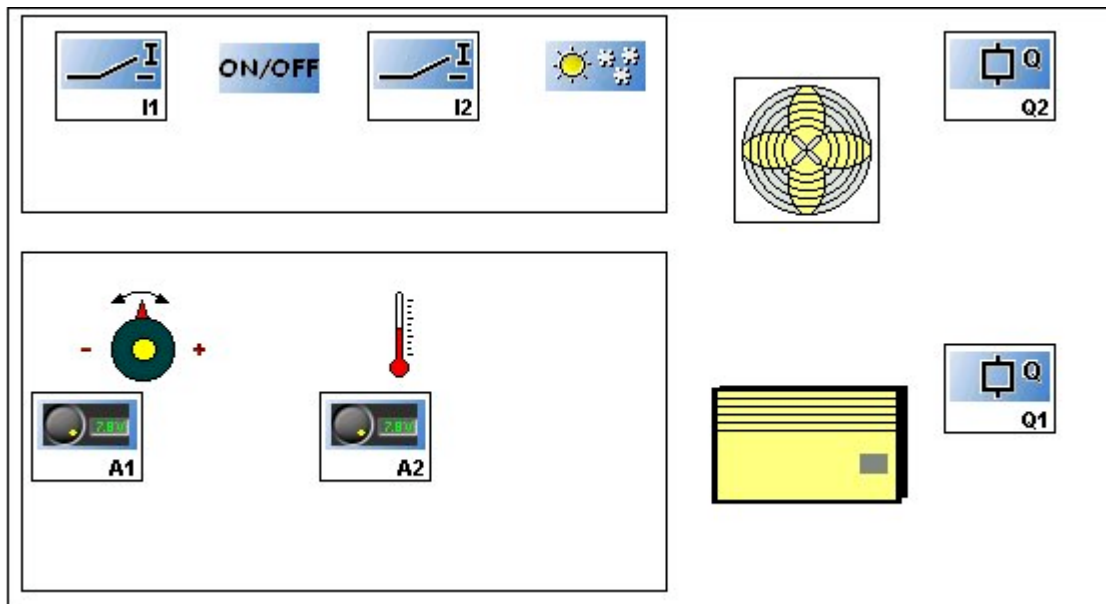


To set parameters for a function (for example a timer), double click on one of the inputs/outputs for this function. A configuration window will then appear. You may choose the following parameters. For further details on each function, refer to **4.2.3 Functions**.

Supervision Window

Select **Window** then **Supervision**. Simply click and drag the inputs/outputs and block function that you have chosen from the wiring sheet to the supervision window. You may illustrate your application using the tools in the **Drawing** menu. You may also choose a background image in Bmp format. This window clearly displays the elements you placed in the wiring sheet in their environment. When you change to simulation or monitoring mode, the inputs and outputs are updated and it is also possible to force an input, as with the wiring sheet.

The following is an example of window supervision using Ladder programming:



The inputs/outputs are located the application as follows.

This example concerns temperature control in a room. Regulation may be disabled by the **I1** switch and the hot or cold mode is activated by the **I2** switch (in cold mode, only the fan is activated). The setpoint is compared to the room temperature and if the difference is greater than a given value (Comparators **A1** and **A2**), the fan and possibly the heating are activated (**Q1** and **Q2**).

Note: This application is developed in the application library in Ladder language under the name "Room temperature regulation".

Click on the link below to access the example:

[\(Ex 02\)](#)



Programming in Zelio entry mode or in manual entry mode with the integrated keyboard

For information on programming in Zelio entry mode, please refer to the operating manual. The software program allows you to use the keyboard shortcuts to simplify programming, such as the Shift key for **Shift** (White key) or Enter for **Menu/Ok**.

Coherence Function

The coherence function, represented by the eye icon in the status bar, indicates any coherence problem in the wiring. It also allows wiring verification and correction if necessary.

A blue icon indicates the wiring is correct. A red icon indicates an error. Click on the red icon to see the different wiring errors.



Blue Icon: NTR



Red Icon: Wiring problem, click on the icon to find out more

Program configuration

Program configuration allows you to customize your file by adding the project name and author. It is also possible to adjust certain configuration settings and to choose the date format.


Click on the corresponding icon



in the lower bar to access program configuration.

4.2.3 Functions

Note: The following descriptions are illustrated with functional examples.

Click on the  icon to access the descriptions.

If Zelio Soft 2 software is installed, click on the link to open the program. You may then select simulation mode **(1)** and start the module (**RUN**) **(2)**.



For further details on a function described below, refer to Help: Double-click on the block and then on ?

4.2.3.1 Discrete Inputs



Discrete Inputs I

Discrete-type inputs (**I1, I2, ...**) and mixed inputs (discrete or analog) (**IB, IC...**) In Ladder programming, a mixed input placed on a contact is always discrete. The analog comparator function allows the mixed input to be used as an analog output. All analog inputs take 0 to 10 V input voltage, corresponding to a value of 0 to 255.



Buttons

You may use 4 Zelio Logic front-panel buttons (**Z1**, **Z2**, **Z3**, **Z4**) in your application. Unlike the physical inputs I, they do not have connection terminals.

*Note 1: **Zx** keys can not be used if locked (see APPLICATION TRANSFER for further details).*

*Note 2: When the module is running and you wish to use the **Zx** keys in the program, access the INPUTS-OUTPUTS screen and simultaneously press **Shift** (White key) and **Z1**, **Z2**, **Z3** or **Z4**.*

4.2.3.2 Outputs



Q Outputs

Discrete-type outputs can be used either as coils or contacts.

◆ Use as coil:

[Q (Contactor): The coil is energized if the contacts to which it is connected are closed.

α Q (Impulse relay): Impulse energizing, the coil is energized by a change of state. Its function is identical to that of an impulse relay.

SQ (Set) : "Set" (latch) or triggered coil. This coil is triggered as soon as the contacts that are connected to it are closed. It remains triggered even if the contacts are no longer closed.

RQ (Reset) : "Reset" or (unlatch) or deactivated coil. This coil is deactivated as soon as the contacts that are connected to it are closed. It remains inactive even if the contacts are no longer closed.

◆ Used as contact:

Q (Normal function) or **q** (Reciprocal function): physical output from the smart relay. An output can be used as a contact to determine its state at a given time.

Example 1:

Q1-----[Q2

Q2 duplicates input **Q1** status.

Example 2:

q1-----[Q2

Q2 output will always be reciprocal to **Q1** status.

*Note: The **[** and **α** , **SET** and **RESET** functions must be used once and once only for each coil in a control diagram.*

*Additionally, if you use a **SET** (**S** function), a control diagram line must always be provided for disabling this coil using a **RESET** (**R** function)*

If this is not done, there will be always the risk of generating unexpected switch statuses during operation.



Auxiliary M Relays (or internal memory)

The auxiliary relays operate just like the **Q** output coils. The only difference is that they do not have any connection terminals. They are used to save or forward a state. The saved or forwarded state will then be used as the assigned contact.

For example:

I1-----[M1
M1-----[Q1

Activation of input **I1** activates **Q1** output, via **M1**.

4.2.3.3 Block Functions

Boolean Function






The control diagram entry mode allows you carry out Boolean functions by using the elementary logic functions **AND** and **OR**:

I1—I2———Q1 Associated logic equation: $Q1 = I1 \times I2$, **AND** Logic

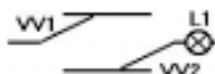
I1—I2———Q1 Associated logic equation: $Q1 = I1 + I2$, **OR** logic
I2—|

Function **i**, the opposite of **I**, produces **NO**. This makes it possible to produce many different functions.

Example of a Boolean function:

No.	Contact 1	Contact 2	Contact 3	Contact 4	Contact 5	Coil
001	I1 		i2 			[Q1 
002	i1 		I2 			

The following logic equation: $Q1 = (I1 \times I2) + (I1 \setminus \times I2) = (I1 \times i2) + (i1 \times I2)$
Corresponds to the following electrical diagram:



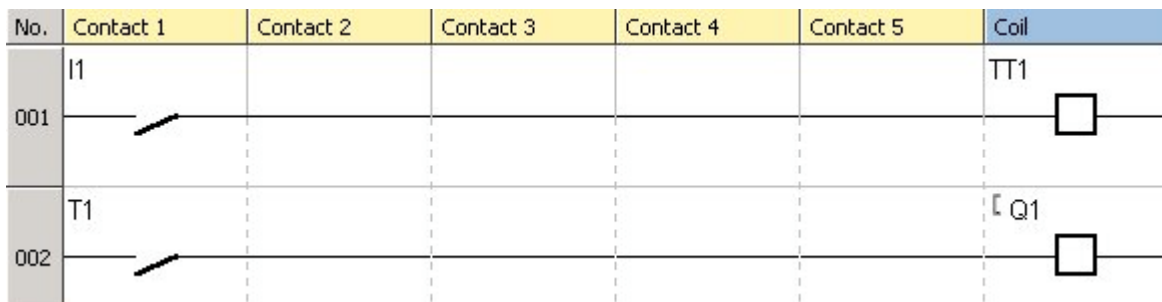
This example illustrates the implementation of a two-way switch.



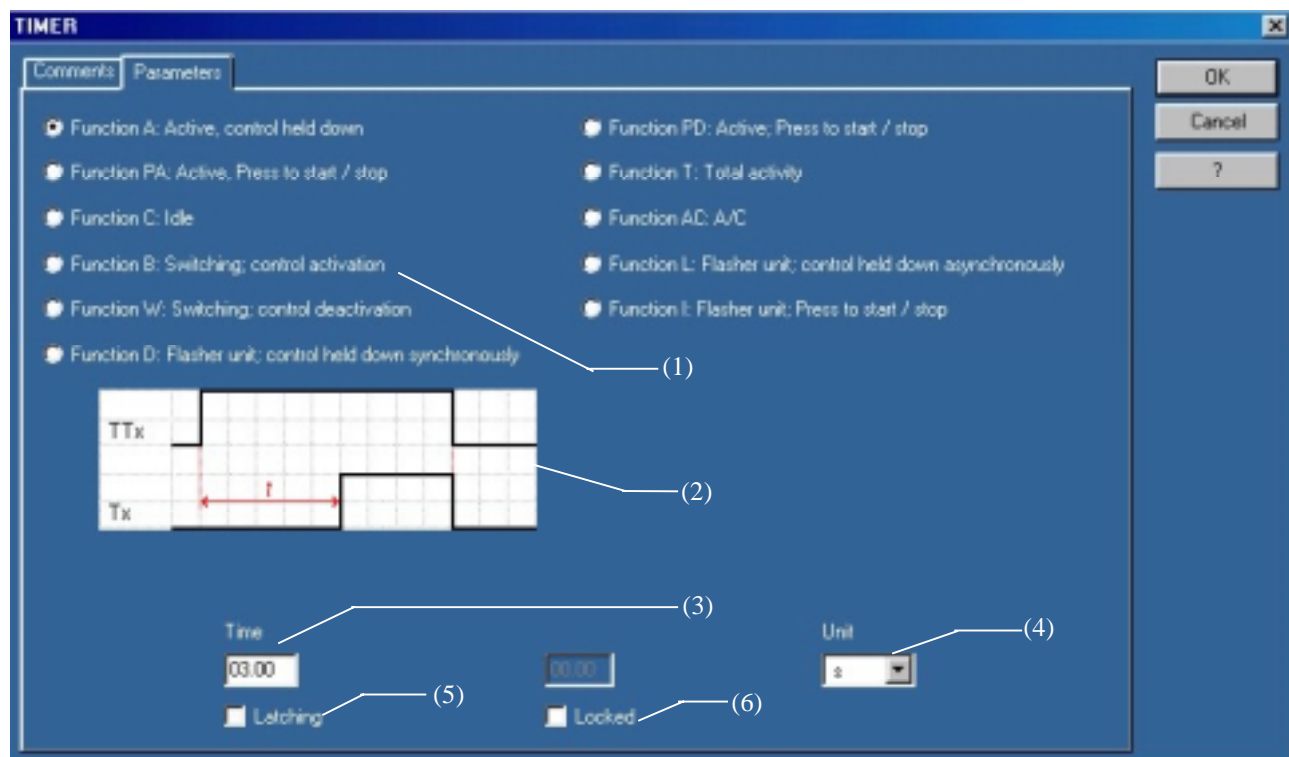
Timer T

The Timer function block is used to delay, prolong and control actions during a set period of time.

Example of a diagram using this function:



Double click on **TT1** or **T1** to display the T1 timer block parameter setting window:



A list of functions **(1)** allows you to choose the type of timer. A diagram **(2)** corresponding to each timer type enables you to find the function you need to use. Zone

(3) enables you to enter the time lag required for each unit (4). Latching is activated by checking the relatching box (5). You may lock the parameters by checking this box (6).

Explanation: When I1 is in high status, Q1 will change to high status with a time lag t (here 03.00 s) and it will go back to the low status when I1 is deactivated (A type function)

Click on the link below to access the example:

[\(Ex 03\)](#)



There are 3 main types of timers:

- Type A: Active, Control Held Down



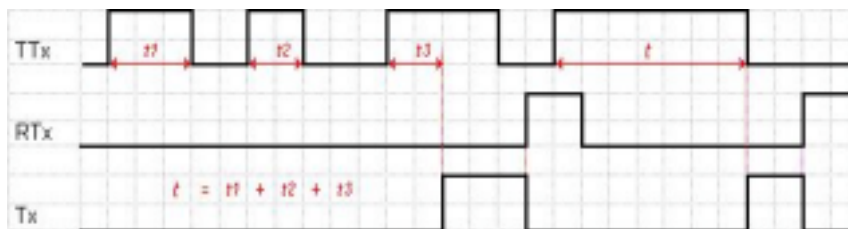
For example: Delay the second motor start-up to limit energy consumption.

Click on the link below to access the example:

[\(Ex 03\)](#)



- Type T: Total Activity Accumulator



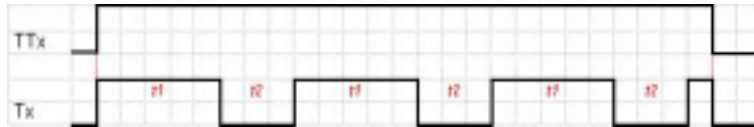
For example: Request replacement of equipment when recommended service life is exceeded.

Click on the link below to access the example:

[\(Ex 04\)](#)



- Type L or Li: Flasher Unit, Control Held Down/Asymmetrical



For example: Control a sound signal and create an alarm sound.

Click on the link below to access the example:

[\(Ex 05\)](#)



Other types of timers may be used (11 types of timers)
Each timer type possesses an input control (**TT**) and an input reset (**RT**).

"LATCHING" Save data function available.

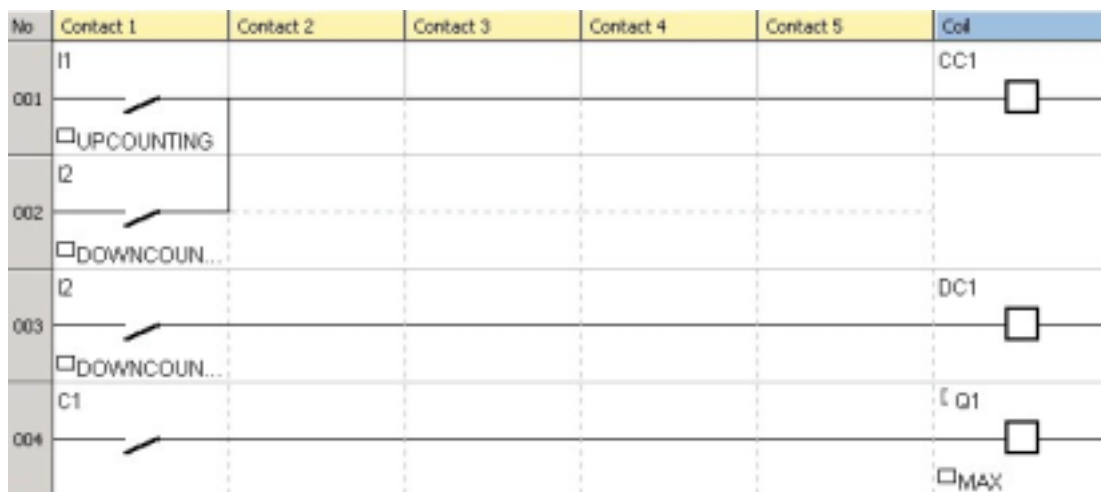


Counter

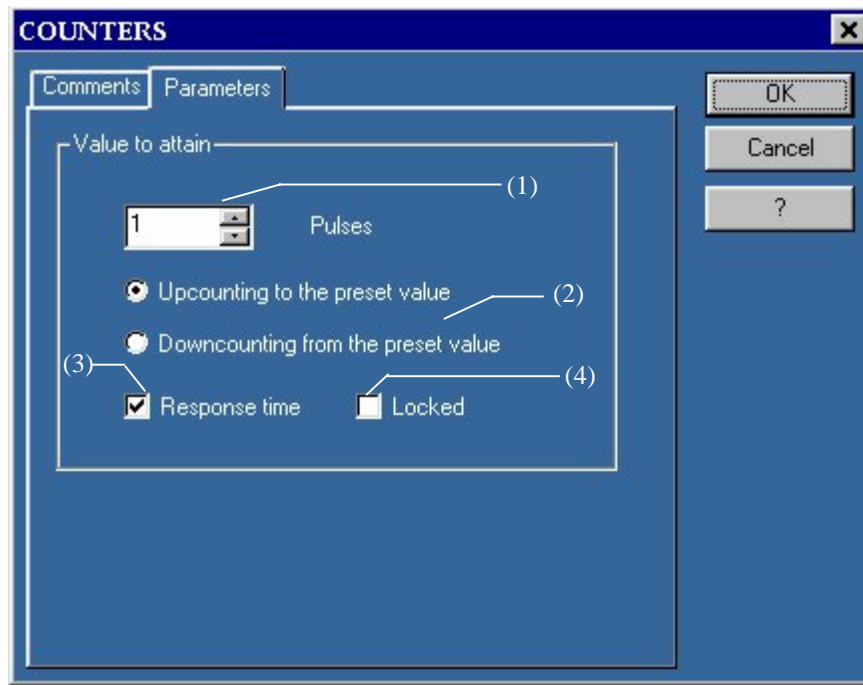
This function enables you to upcount or downcount pulses, until a value preset in the parameter setting window has been reached.

The block function Counter contains a count input (**CC**) (each time the coil is energized, the counter is incremented or decremented by 1 depending on the chosen counter direction), a Reset input (**RC**), a counter direction input (**DC**) (the block downcounts if this input is activated) and a **C** output that enables you to know what level is controlled by the counter. When the preset value has been reached, this output goes to 1 until it is zeroed or the counting direction changes. The counting value and the preset value can be visualized on the module screen.

Diagram produced using this function:



The parameter window is as follows:



Field (1) enables you to enter the value to be reached (preset value). In (2), you may choose between counting up to the preset value or counting down from the preset value. Latching is activated by checking the latching box (3). Check box (4) to lock the parameters.

Explanation: Every time I1 is pressed, the counter is incremented by 1. Pressing on I2 changes the counter direction (DC1), the counter counts down. When the preset value is reached (here 5), Q1 will be in high status, as will Q1 output. In a parking lot for example, each car input activates I1 and each output activates I2. When the parking lot is full, the Q1 output blocks the input.

Click on the link below to access the example:

[\(Ex 06\)](#)



"LATCHING" Save data function available.



Fast Counter



Counter Comparator



Analog Comparator

Available only in modules with analog outputs.

This function block is used in applications using analog data, and enables you to compare a measured analog value and an internal value, or two measured analog values.

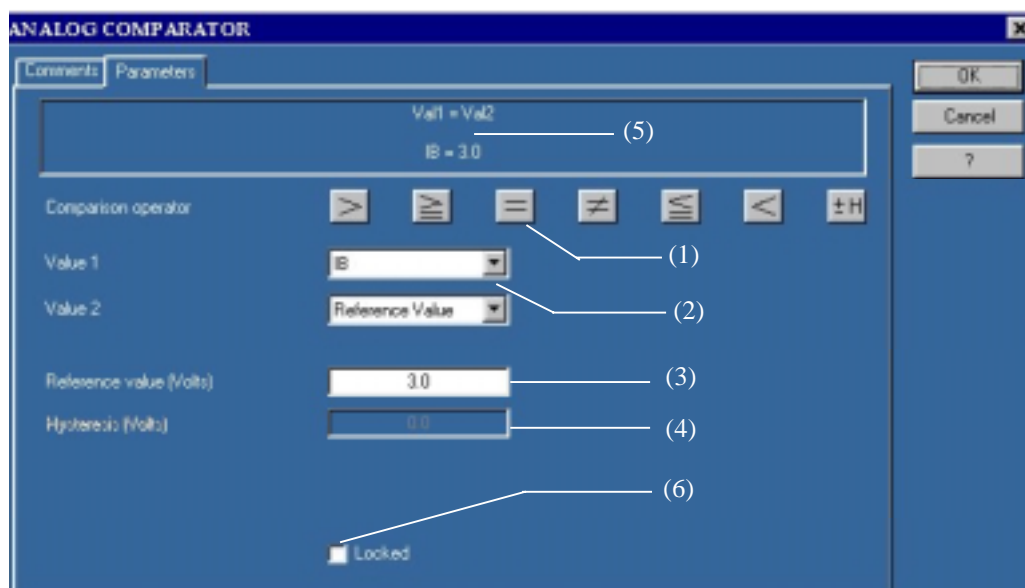
The comparison result is used as a contact.

This function is shown in the diagrams by the letter **A** (**a** for the reciprocal function)

Diagram produced using this function:



Parameters for the **A1** block are shown as follows (double click on **A1**):



Different comparison operators may be used (1). Use fields (2) to select the 2 values to be compared. The values in use are the analog inputs (up to 6 according to the model) and the reference value entered in field (3) (between 0.0 and 9.9 V). Field (4) does not appear until you select the operator "+-H"; this field will then enable you to enter the hysteresis value. Frame (5) summarizes the operation carried out according to the operators and operands chosen. Cell (6) can be used to lock the parameters.

Explanation: Contact **A1** is closed when the value of analog input **IB** equals or exceeds the value of **IC**. Output **Q1** is now active.

For example, in a room, when the temperature (analog input **IB**) exceeds the **IC** setpoint, the **Q1** fan will start up.

Click on the link below to access the example:

[\(Ex 07\)](#)



Here are 2 examples of formulas and their interpretation:

- Value 1 = Value 2

with Value 1=ID and Value 2=Reference Value=5.6V

Contact **A1** is closed when the value of analog input **ID** equals or exceeds the reference voltage entered. In this case, 5.6 V.

Click on the link below to access the example:

[\(Ex 08\)](#)



- Value 1 - H <= Value 2 <= Value 1 + H

with Value 1=ID and Value 2=IC and Hysteresis (H)=2.3 V

Contact **A1** is closed when the value of analog input **IC** is between **ID - 2.3 V** and **ID + 2.3 V**.

Click on the link below to access the example:

[\(Ex 09\)](#)


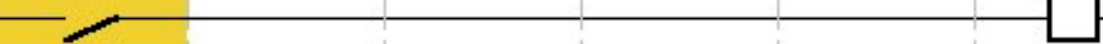


Clock Block Function - Daily and weekly programmer

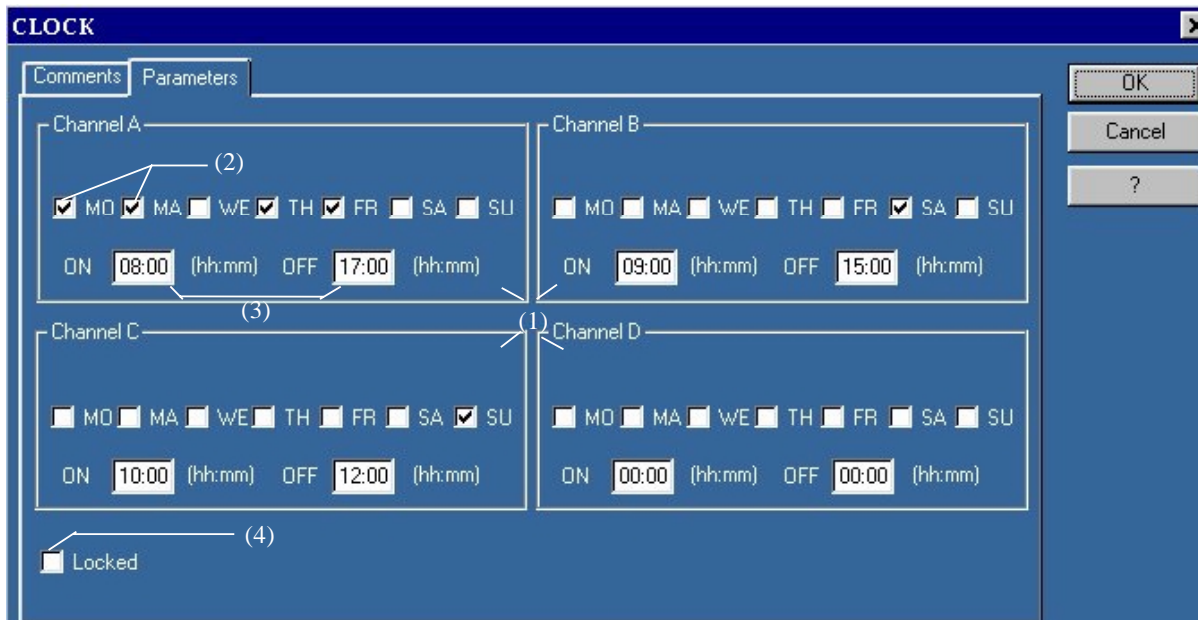
Available only in modules with a clock.

This function is used to activate or deactivate the output at a specific moment during the day or the week. This function is therefore only placed on a contact. This block is event based.

This function is shown in the diagrams by the symbol ⌚ (⌚ for the reciprocal function).
Diagram produced using this function:

No.	Contact 1	Contact 2	Contact 3	Contact 4	Contact 5	Coil
001	 1 					Q1

The parameters of the  1 block appear as follows:



Four separate time slots are available (1). Fields (2) enable you to choose the days of the week when the clock will be activated. The activation and deactivation schedule must be entered in the "ON" and "OFF" fields (3). Cell (4) may be used to lock the parameters.

Explanation: The **Q1** coil will be activated every week from Monday to Friday from 8 a.m. to 5 p.m. (Channel A), Saturday from 9 a.m. to 3 p.m. (Channel B) and Sunday from 10 a.m. to 12 p.m. (Channel C). Channel D was not used in this example. This clock could be used, for example, to define the hours during which a building is open.

Click on the link below to access the example:

[\(Ex 10\)](#)



Display function

Available only in modules with a display system.



Display screen backlighting.

Available only in modules with backlighting.

When activated, the function acts as an output and ensures display lighting.



Daylight Savings Change Summer/Winter

Available only in modules with a clock

The output for this function is OFF when winter daylight saving time applies, and ON when summer time applies. The switch from winter to summer time is displayed on the screen.