Welcome to LOGO!

Dear customer,

Thank you for purchasing LOGO!, and congratulations on your decision. In LOGO! you have acquired a logic module that meets the stringent quality requirements of ISO 9001.

LOGO! is universal in application. Its comprehensive functionality and great ease of use make it a highly cost-efficient solution for virtually any application.

LOGO! documentation

This LOGO! manual tells you how to install, progam and use LOGO!.

In addition, the step-by-step graphical guide shipped with LOGO! and the LOGO!Soft online help system provide you with the essentials. LOGO!Soft is a programming package that runs on PCs under Windows™ . It will help you get to know LOGO! and test, print and archive programs independent of LOGO!.

Guide to the manual

We have subdivided this manual into 8 chapters:

- Getting to know LOGO!
- Installing and wiring LOGO!
- Programming LOGO!
- LOGO! program modules
- Parameterizing LOGO!
- LOGO!Soft
- Applications
- Appendices for technical data, facts specific to the AS interface, determining the amount of memory required and abbreviations

Additional support

If you have any questions concerning LOGO!, the dealer from whom you bought it will be glad to help you.

Safety guidelines

This manual contains notices which you should observe to ensure your own personal safety, as well as to protect the product and connected equipment. These notices are highlighted in the manual by a warning triangle and are marked as follows according to the level of danger:

Danger
indicates that death, severe personal injury or substantial property damage **will** result if proper precautions are not taken.

Warning
indicates that death, severe personal injury or substantial property damage **can** result if proper precautions are not taken.

Caution
indicates that minor personal injury or property damage can result if proper precautions are not taken.

Note
draws your attention to particularly important information on the product, handling the product, or to a particular part of the documentation.

Warning
Only **qualified personnel** should be allowed to install and work on this equipment. Qualified persons are defined as persons who are authorized to commission, to ground, and to tag circuits, equipment, and systems in accordance with established safety practices and standards.

Warning
This device and its components may only be used for the applications described in the catalog or the technical description, and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens.
This product can only function correctly and safely if it is transported, stored, set up, and installed correctly, and operated and maintained as recommended.

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

Contents

1 Getting to know LOGO!

What is LOGO! ?

LOGO! is the new universal logic module from Siemens.

LOGO! provides

- Control functions
- An operating and display unit
- A power supply
- An interface for program modules and a PC cable
- Ready-to-use basic functions that are often required in practice, such as functions for on and off delays and pulse relays
- A clock/time switch (LOGO! 230RC, LOGO! 230RCL, LOGO! 24RC)
- And inputs and outputs depending on the device type

You can use LOGO! for domestic and installation engineering tasks (e.g. stairway lighting, external lighting, sun blinds, shutters or shop window lighting) and for mechanical and apparatus engineering (e.g. gate control systems, ventilation systems or rainwater pumps).

What devices are available?

LOGO! is available both for 24 V and 230 V power supply as

- a standard version with 6 inputs and 4 outputs, with the dimensions 72 x 90 x 55 mm
- an ..L version with 12 inputs and 8 outputs and extended functional scope, with the dimensions 126 x 90 x 55 mm
- an ..LB11 version with 12 inputs and 8 outputs, extended functional scope and additional AS interface bus connection over which 4 further inputs and 4 further outputs are available in the bus system. And all this squeezed into dimensions of 126 x 90 x 55 mm.

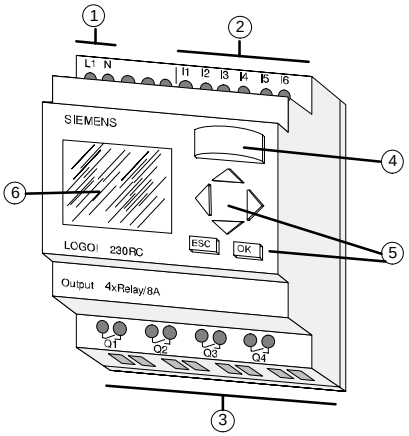
It's your choice

The various versions offer extremely flexible adaptation to your own specific task to be undertaken.

LOGO! offers you solutions ranging from small domestic installation through small automation tasks to extensive tasks integrating the AS interface bus system.

How LOGO! is structured

LOGO!... standard version



① Power supply

② Inputs

③ Outputs

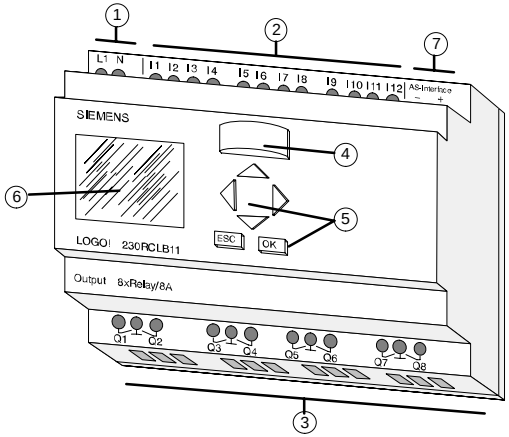
④ Module shaft with cover

⑤ Control panel (keys)

⑥ LCD display

⑦ AS interface connection (optional)

LOGO!...L/LOGO!...LB11



Versions

The following different versions of LOGO! are available:

Symbol	Designation	Relay outputs	Transistor outputs	Clock	ASi bus
	LOGO! 230R	4 * 230 V; 8A			
	LOGO! 230RC	4 * 230 V; 8A		✓	
	LOGO! 24R	4 * 230 V; 8A			
	LOGO! 24RC	4 * 230 V; 8A		✓	
	LOGO! 24		4 * 24 V; 0,3 A		
	LOGO! 230RL	8 * 230 V; 10A			
	LOGO! 230RCL	8 * 230 V; 10A		✓	
	LOGO! 24RL	8 * 230 V; 10A			
	LOGO! 24L		8 * 24 V; 0,3 A		
	LOGO! 230RLB11	8 * 230 V; 10A			✓
	LOGO! 24RLB11	8 * 230 V; 10A			✓
	LOGO! 24LB11		8 * 24 V; 0,3 A		✓

LOGO! has UL, CSA and FM certification, carries CE marking, complies with the VDE 0631 and IEC1131 standards and has interference suppression in accordance with EN 55011 (limit class B).

Certification Society Approval (ABS, BV, DNV, GL, LRS) has been issued or is pending for LOGO! ...L... versions.

LOGO! can therefore be put to use both in industry and in the domestic scene.

How to recognize which LOGO! version you have

LOGO!'s designation contains information about various characteristics:

- 24: 24 V DC version
- 230: 115/230 V AC version
- R: relay outputs
- C: integrated seven-day time switch (clock)
- L: twice the number of outputs and inputs and extended functional scope
- B11: slave with AS interface bus connection

In this description of LOGO!, we also use small pictographs to identify the different types. They are used wherever information refers to only one part of the LOGO! versions:

Standard version with 6 inputs and 4 outputs with dimensions of 72 x 90 x 55 mm

..L version with 12 inputs and 8 outputs with dimensions of 126 x 90 x 55 mm

..LB11 version with 12 inputs and 8 outputs and additional AS interface bus connection with 4 virtual inputs and 4 virtual outputs, with dimensions of 126 x 90 x 55 mm

If details refer to ...C versions or ...R versions only, we will make a specific reference to this fact in the text.

2 Installing and wiring LOGO!

General

We will show you how to install and deinstall LOGO! with the aid of an illustration of the LOGO! 230RC. The measures described also apply to all other LOGO! modules.

You install LOGO! in a distribution box or cabinet, ensuring that the connectors are covered. If they are not, there is a danger of touching live parts.

LOGO! must be installed and wired by a trained technician who knows and complies with both the universally applicable engineering rules and the regulations and standards that apply in specific cases.

Dimensions

The dimensions of LOGO! comply with the DIN 43880 standard for the dimensions of installation equipment.

LOGO! must be snapped onto a DIN rail with a width of 35 mm (DIN EN 50022).

Width of LOGO!:

- LOGO! is 72 mm wide, which corresponds to the size of 4 modules (standard version).
- LOGO!...L is 126 mm wide, which corresponds to the size of 7 modules.
- LOGO!...LB11 is 126 mm wide, which corresponds to the size of 7 modules.

2.1 Installing/deinstalling LOGO!

Installing

You **install** LOGO! on a DIN rail as follows:

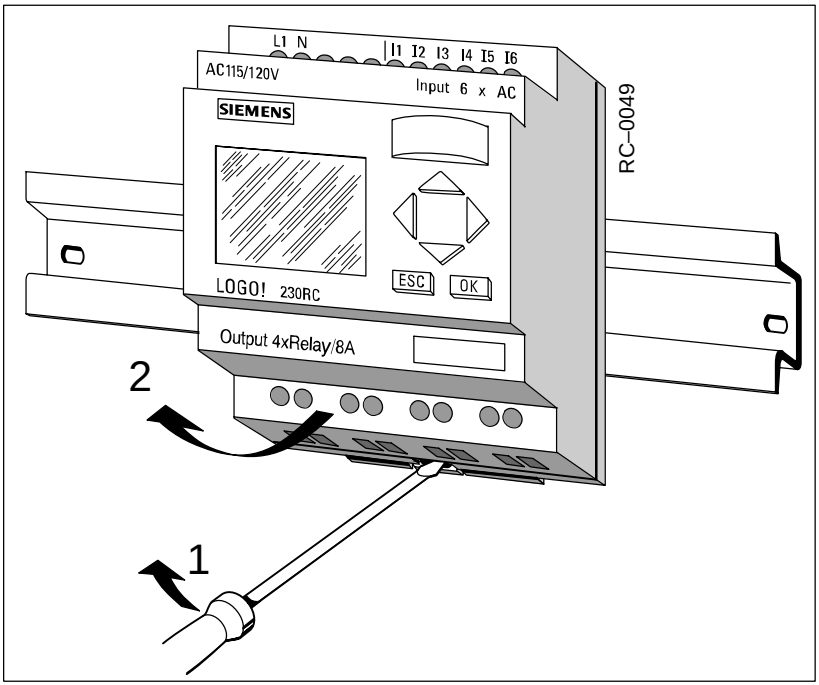
1. Place LOGO! on the rail.
2. Swivel it. The snap catch on the back of LOGO! must engage.

Depending on the type of DIN rail used, the snapping mechanism may be a bit stiff. If it is too stiff and LOGO! will not snap on, you can pull the snap catch down a little, as you do when deinstalling LOGO! as described below.

Deinstalling

You **deinstall** LOGO! as follows:

1. Insert a screwdriver in the hole shown in the picture at the lower end of the snap catch, and pull the snap catch downward.



2. Swivel LOGO! away from the DIN rail.

2.2 Wiring LOGO!

Use a screwdriver with a head 3 mm wide to wire LOGO!.

You do not need wire end ferrules for the connectors. You can use wires up to the following sizes:

- 1 x 2.5 mm^2
- 2 x 1.5 mm^2

2.2.1 Connecting the power supply

LOGO! 230 versions are suitable for line voltages with a rating of 115 V and 230 V; LOGO! 24 versions are suitable for a supply voltage of 24 V DC. Please note the Technical data in appendix A that refer to the permissible voltage tolerances, mains frequencies and current consumptions.

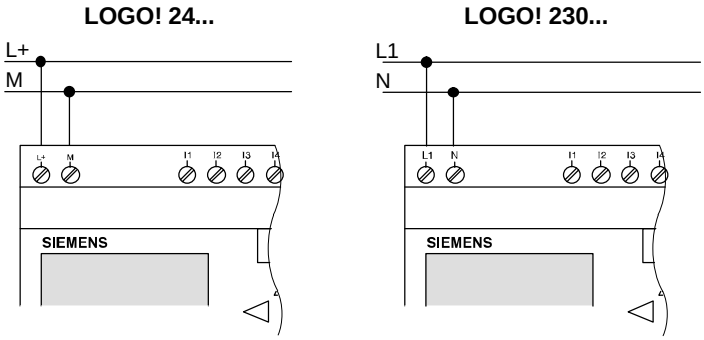
Note

If you are running both LOGO! and its inputs by means of the same power supply, it is possible for an incorrect value to be stored for the remanent functions due to power failure bridging. Under certain circumstances, this can lead to an additional edge after power restoration with edge-triggered special functions.

Make sure that you feed in the power supply for LOGO! and its inputs separately.

Connecting

You connect the sensors to LOGO! as follows:



Note

LOGO! has protective insulation. A ground terminal is not necessary.

2.2.2 Connecting LOGO!'s inputs

Requirements

You connect sensors to the inputs. The sensors may be switches, photo-electric barriers or daylight control switches, for example.

Sensor attributes for LOGO!

	LOGO! 230..	LOGO! 230L..	LOGO! 24..	LOGO! 24..L..
Switch state 0	< 40 V AC	< 40 V AC	< 5 V DC	< 5 V DC
Input current		0.8 ... 1.2 mA		< 1.5 mA
Switch state 1	>79 V AC	>79 V AC	> 15 V DC	> 12 V DC
Input current	typically 0.24 mA	typically 2.5 mA	typically 3 mA	typically 5 mA
Proximity switch	 3-wire 4-wire	 3-wire 4-wire	 3-wire 4-wire	2-wire 3-wire 4-wire
Switches with Glow lamps	yes[1] / no[2]	yes	–	–

1 Glow lamps with a closed-circuit current up to 0.2 mA

2 Glow lamps with a closed-circuit current >0.2 mA possible over relay or with additional N for the glow lamp

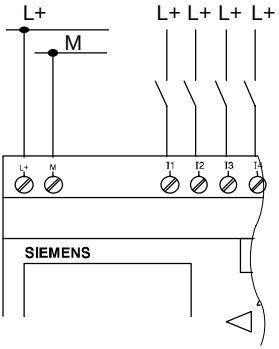
Switch state change 0 ⇨ 1 / 1 ⇨ 0

When the switch state changes from 0 to 1, switch state 1 must exist for at least 50 ms for LOGO! to recognize it. The same applies to state 0 when the change is in the opposite direction.

Connecting

You connect the sensors to LOGO! as follows:

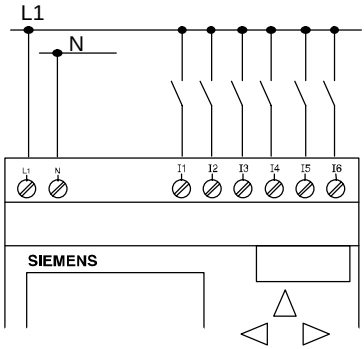
LOGO! 24 ...



LOGO! 24...

The inputs of LOGO! 24... are non-isolated and must therefore be grounded in the same way as the power supply.

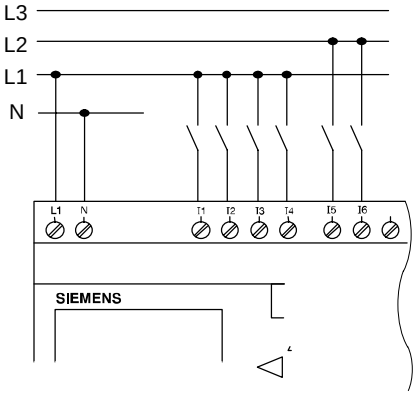
LOGO! 230 ... (standard version)



Warning

Existing safety regulations (VDE 0110, ... IEC 1131, ... and UL and CSA) prohibit the connection of different phases to the inputs of LOGO! 230R/RC.

LOGO! 230 .. L...



LOGO! 230..L...

The inputs of LOGO! ...L.. are grouped in groups of 4 inputs. The same applies to these groups as for the individual inputs of a standard LOGO!. Different phases are possible only between the blocks.

Warning

Existing safety regulations (VDE 0110, ... IEC 1131, ..., and UL and CSA) prohibit the connection of different phases to one input block of LOGO! 230R/RC**L**...

2.2.3 Connecting outputs

LOGO! 230R... and LOGO! 24R...

The outputs of LOGO! 230**R**... and LOGO! 24**R**... are relays. The contacts of the relays are isolated from the power supply and the inputs.

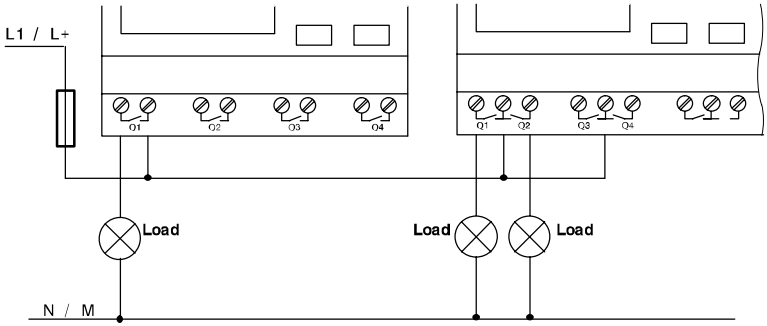
Requirements for the relay outputs

You can connect different loads to the outputs, such as lamps, fluorescent tubes, motors, contactors, etc. The loads connected to LOGO! ...R... must have the following properties:

- The maximum switched current depends on the type of load and the number of operations. You will find more information on this in the technical specifications.
- When switched on (Q = 1), the maximum current is 8 amperes (10 A with LOGO!...RL..) for a non-inductive load and 2 amperes (3A with LOGO!...RL...) for an inductive load.

Connecting

You connect the load to LOGO! ...**R** .... versions as follows:



Protection with automatic circuit breaker (max. 16 A, B16), e.g. power circuit breaker 5SX2 116-6 (if desired)

LOGO! 24... with transistor outputs

LOGO! 24... versions with transistor outputs can be identified by the fact that the letter **R** is missing from their type designation. The outputs short-circuit proof and overload proof. A separate voltage supply to the load is not necessary; LOGO! 24... supplies the load with voltage.

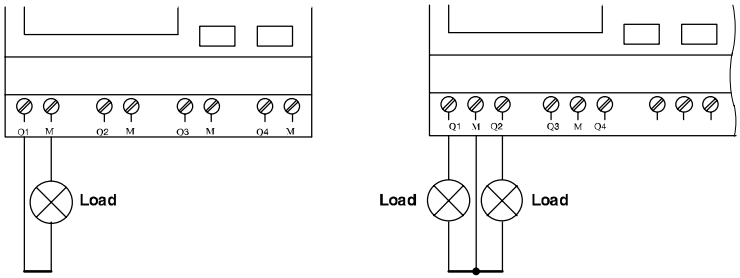
Requirements for transistor outputs

The load connected to LOGO! 24... must have the following properties:

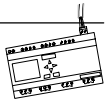
- The maximum switched current is 0.3 amperes per output.
- When switched on Q = 1), the maximum current is 0.3 amperes.

Connecting

You connect the load to LOGO! 24 as follows:



Load: 24 V DC, 0.3 A max.



2.2.4 Connecting the ASi bus (LOGO! ...LB11 only)

This section will be of interest to you if you want to connect LOGO!..LB11 to the ASi bus.

LOGO!...LB11

LOGO!...LB11 can be integrated into a network as an ASi slave. Using a two-wire lead, you can then

- read in and process 4 additional inputs via the ASi bus
- operate 4 additional outputs on one overlaid master of the Asi bus

You configure LOGO!...LB11 in the ASi bus by means of the ASi master you are using.

Requirements for operating LOGO!...LB11 on an ASi master

Please note: LOGO! ...LB11 must be registered in the ASi system, i. e. LOGO! is assigned an address by the bus master. Please read section 2.2.5 to find out how to do this with LOGO!.

Caution

The ASi address can be changed at least 10 times for all LOGO! ...LB11 versions.

We cannot guarantee further changes made.

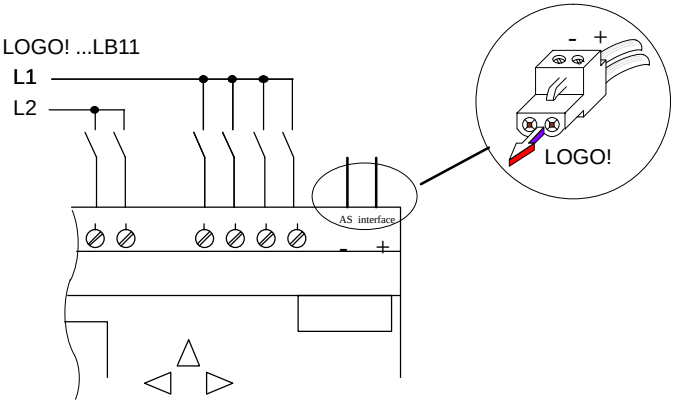
Bus connector

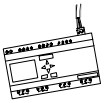
Always use the standard bus connector enclosed in the packaging to connect LOGO!.

Connecting

Connect the bus connector cable to an approved connector in the system, making sure that the polarity is correct.

Then push the wired connector into the interface marked **AS interface**.





2.2.5 LOGO!...LB11 on the ASi bus

LOGO!...LB11 must be known to the bus master if you are to be able to use the ASi functionality. This takes place automatically when you connect LOGO!...LB11 to the bus lead. The master detects the address of the slave.

In the case of LOGO!...LB11, the address preset at the factory = 0. The master assigns a new address that is not equal to 0.

If there are no address conflicts in the system or if only one slave with the address 0 is connected, you do not have to take any further steps.

Note

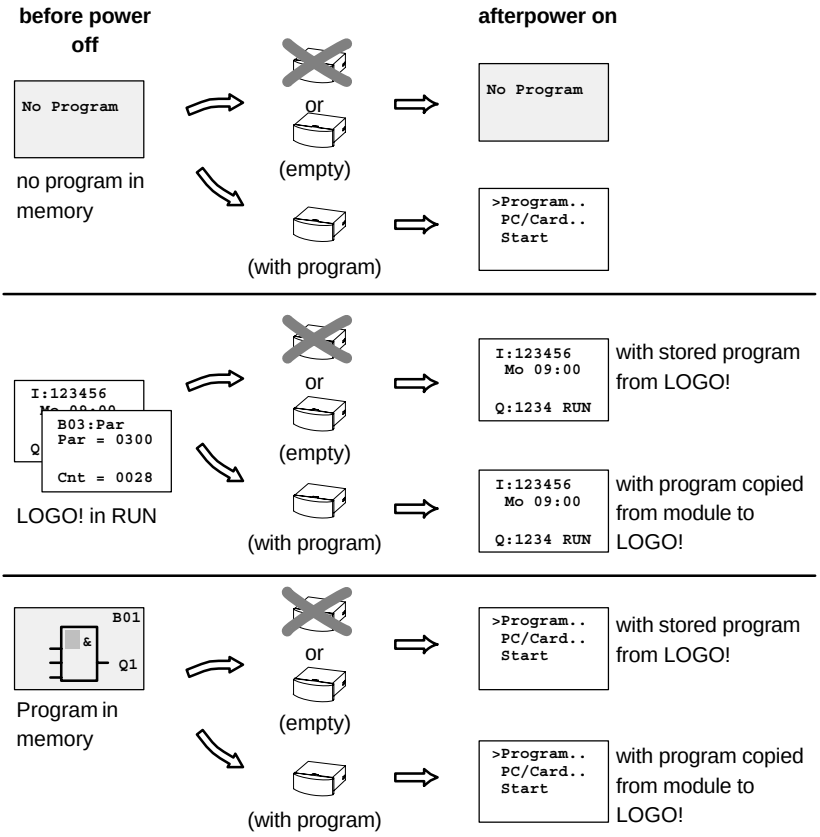
If you are connecting several slave assemblies (e.g. LOGO!...LB11) to the bus at the same time, please read Appendix B.

2.3 Switching LOGO! on/resumption of power

LOGO! does not have a power switch. How LOGO! responds when switched on depends on:

- Whether a program is stored in LOGO!
- Whether a program module is connected
- The state LOGO! was in before power off

The table indicates LOGO!'s responses to the possible situations:



Try to remember the 5 simple rules for starting LOGO!:

1. If there is no program in LOGO! or on the program module connected, LOGO! displays the message: No Program
2. If there is a program on the program module, it is copied to LOGO! automatically. If there is already a program in LOGO!, it is overwritten.
3. If there is a program in LOGO! or on the program module, LOGO! adopts the operating status is had before power off.
4. If you are using a LOGO!...L version with a red or yellow module and have remanence switched on for at least one function or using a function with remanence permanently switched on, its instantaneous values are retained at power off.
5. With all other versions, the times and count values are reset at power off. The program is stored in such a way that it is secure against power failure.

Note

If a power failure occurs while you are entering a program, the program in LOGO! is deleted when the power is restored.

You should therefore back up your original program on a program module (card) before changing it.

LOGO! operating statuses

LOGO! has 2 operating statuses: STOP and RUN

LOGO! is in STOP	LOGO! is in RUN
when 'No Program' is displayed or when you switch LOGO! to programming mode	when the mask for monitoring the inputs and outputs is displayed (after START in the main menu) or when you switch LOGO! to parameterization mode
Action by LOGO!: • the inputs are not read. • the program is not executed. • the relay contacts are always open or the transistor outputs are switched off.	Action by LOGO!: • LOGO! reads the status of the inputs. • LOGO! calculates (with the program) the status of the outputs. • LOGO! switches the relays/transistor outputs on or off.

3 Programming LOGO!

The first steps with LOGO!

By programming, we mean entering a circuit. A LOGO! program is really no more than a circuit diagram represented in a different way.

We have changed the way it is represented to suit LOGO!'s display panel. In this chapter, we will show you how to use LOGO! to turn your applications into LOGO! programs.

In the first section of the chapter, a brief example will help you get to know how to use LOGO!.

- First of all, we will begin by introducing the two basic terms **connector** and **block**, and show you what is meant by these terms.
- In a second step, we will develop a program from a simple, conventional circuit, and ...
- in the third step, you can then enter this program directly in LOGO!.

After reading through only the first few pages of this manual, you will already have stored your first executable program in LOGO!. Using suitable hardware (switches, etc.), you will then be able to carry out your first tests.

What else can you look forward to?

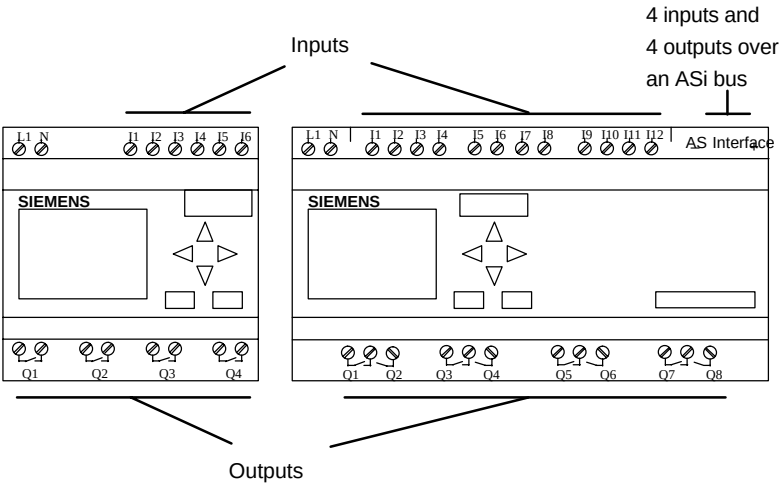
As you progress further through the chapter, you will expand your first program and learn a number of techniques you can use to make changes to an existing program.

Finally, in the third and last section of the chapter, we will introduce you to LOGO! in its entirety. This covers:

- all the functions of LOGO!
- introduction to the menu structure of LOGO!

3.1 Connectors

LOGO! has inputs and outputs:



Each input is identified by the letter I with a number. When you look at LOGO! from the front, you see the connectors for the inputs at the top.

Each output is identified by the letter Q with a number. You will see the connectors of the outputs in the figure below.

Note

Inputs and outputs that are made available with LOGO! ...LB11 by means of the AS interface bus connection are not physical inputs on LOGO! itself.

Note that it is the bus master that defines the input and output devices on the ASi bus.

LOGO!'s connectors

Note

The initial letters CO of the term connector will crop up again later on when you are programming circuits in LOGO!.

The term connector refers to all connections and states used in LOGO!.

The inputs and outputs can have the state '0' or '1'. '0' means there is no voltage at the input, and '1' means that there is. But that is unlikely to be new to you.

We introduced the connector hi, lo and x in order to facilitate program entry for you. 'hi' (high) has the fixed state '1', and 'lo' (low) has the fixed state '0'.

If you do not want to wire an input on a block, you use the 'x' connector. If you want to know what a block is, refer to the next page.

LOGO! recognizes the following connectors:

Connectors			
Inputs	I1 to I6	I1 to I12	I1 to I12 and Ia1 to Ia4 (AS interface)
Outputs	Q1 to Q4	Q1 to Q8	Q1 to Q8 and Qa1 to Qa4 (AS interface)
lo	Signal with level '0' (OFF)		
hi	Signal with level '1' (ON)		
x	an existing connection that is not used		

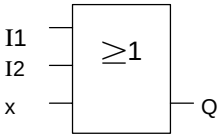
3.2 Blocks and block numbers

A block in LOGO! is a function which converts input information into output information. With earlier versions of LOGO!, you had to wire up the individual elements in the control cabinet or terminal box.

When you program LOGO!, you connect connectors with blocks. To do this, simply select the connection your require from the **Co** menu (Co stands for connector).

The simplest blocks are logic operations:

- AND
- OR
- ...



Inputs I1 and I2 are connected to the OR block. The last input of the block is not used and is therefore marked with an x.

We have made the special functions far more powerful than before:

- pulse relay
- counter
- on-delay
- ....

You can find a complete list of all the functions of LOGO! as of chapter 3.7.

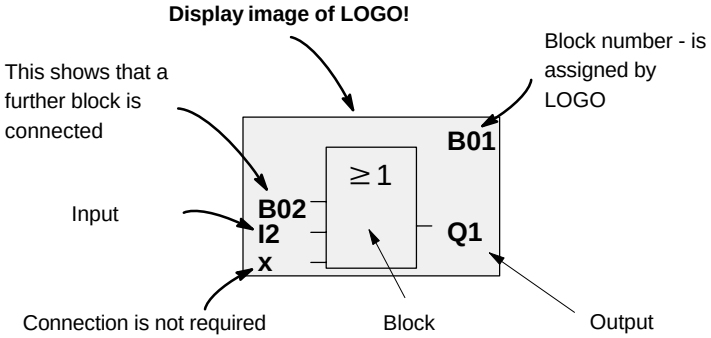
In this chapter, we will introduce you to how you can create extensive circuits with the aid of LOGO!'s elements and how the blocks are linked to each other and to the inputs and outputs.

For this purpose, please turn to the following chapter 3.3. This is where we show you how you turn a conventional circuit into a LOGO! program.

But first of all, you should turn to the information about the block numbers.

Displaying a block in LOGO!'s display

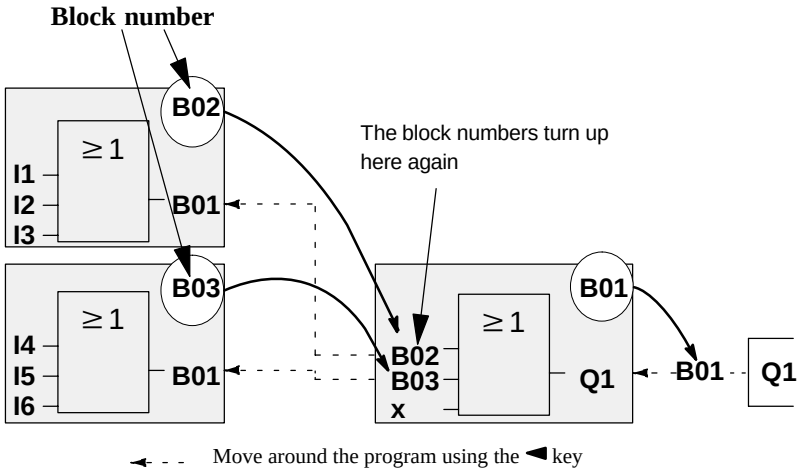
The figure below gives a typical display shown by LOGO!. As you can see, only one block can be depicted at a time. For this reason, we have introduced the use of block numbers, which should help you to keep a check on how the circuit is structured.



Assigning a block number

Whenever you insert a block in a program, LOGO! assigns this block a number, the block number.

LOGO! uses the block number to indicate the connections between blocks. The block numbers are, then, chiefly meant to help you find your way around the program.



The overview display shows you three displays by LOGO!, which together make up the program. As you can see, LOGO! links the blocks with one another by means of the block numbers.

There is, however, one more asset to the block numbers which you can put to good use: you can connect almost any block to an input of the current block by means of its block number. In this way, you can used the interim results of logic or other operations more than once. This saves you the work required to enter things again as well as memory space in LOGO!, and your circuit remains clear and easier to understand. In this case, you have to know how the blocks have been named by LOGO!.

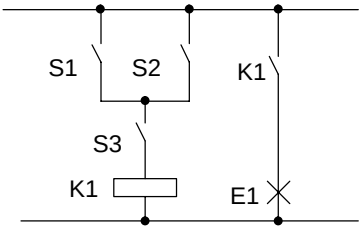
Note

To make working with LOGO! particularly efficient, we recommend that you draw up an overall functional diagram of the program. This will make it a lot easier to generate the program. You can then enter the block number assigned by LOGO! in this diagram.

3.3 From circuit diagram to LOGO!

How a circuit is represented in a circuit diagram

You know, of course, how a circuit is represented in a circuit diagram. Here is an example:

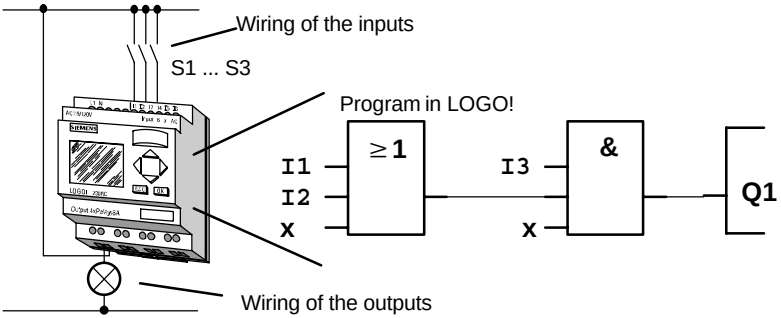


The consumer E1 is switched on and off by means of the switches (S1 **OR** S2) **AND** S3.

The relay K1 picks up when S1 or S2 and also S3 are closed.

Implementing a circuit with LOGO!

You create a circuit in LOGO! by connecting blocks and connectors to each other:

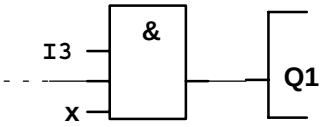


To implement a circuit in LOGO!, begin at the output of the circuit.

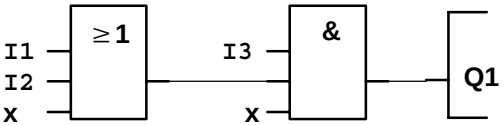
The output is the load or the relay that is supposed to operate.

You convert the circuit to blocks. To do this, you go through the circuit from the output to the input:

Step 1: At output Q1 there is a series connection of the normally open contact S3 with another circuit component. The series connection corresponds to an AND block:



Step 2: S1 and S2 are connected in parallel. The parallel connection corresponds to an OR block:



You have now provided a complete description of the circuit for LOGO!. You now need to connect the inputs and outputs to LOGO!.

Wiring

You connect switches S1 to S3 to the screw connectors of LOGO!:

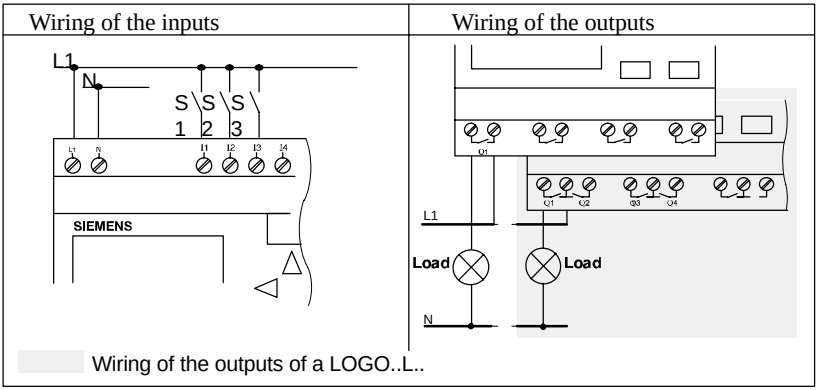
- connect S1 to connector I1 on LOGO!
- connect S2 to connector I2 on LOGO!
- connect S3 to connector I3 on LOGO!

Only 2 inputs of the OR block are used, so the third input must be marked as unused. This is indicated by the **x** next to it.

Likewise, only 2 inputs of the AND block are used. The third input is therefore also marked as 'unused' by an **x** next to it.

The output of the AND block controls the relay at output Q1. Consumer E1 is connected at output Q1.

The following table shows you the wiring on the basis of a 230 V version of LOGO!.



Wiring of the outputs of a LOGO..L..

The remaining steps

Before you enter your first program with us, we would like to introduce you to the 4 most important rules to be followed when working with LOGO!.

3.4 The 4 golden rules for working with LOGO!

Rule 1 - The 3-finger grip 

You enter the circuit in programming mode. You switch to programming mode by pressing the 3 keys ◀, ▶ and **OK** simultaneously.

You change the values of times and parameters in parameterization mode. You switch to parameterization mode by pressing the 2 keys **ESC** and **OK** simultaneously.

Rule 2 - From output to input

You enter a circuit in the following sequence:
From output to input

Rule 3 - Cursor and cursor movement

The following applies when entering a circuit:

- When the cursor appears in the form of an underscore, you can **move the cursor**
  - Use the keys ◀, ▶, ▼ and ▲ to move the cursor in the circuit
  - Press OK to select a connnector/block
  - Press ESC to exit circuit input
- When the cursor appears in the form of a solid block, you **select a connector/block**
  - Use the keys ▼ and ▲ to select a connector/block
  - Press OK to accept a selection
  - Press ESC to go back one step

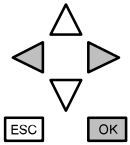
Rule 4 - Planning

Before you enter a circuit, always draw up a complete plan of it on paper.

LOGO! can only store complete programs. If you enter an incomplete program, LOGO! is not able to exit **Programming** mode.

3.5 Overview of LOGO!'s menus

Programming mode



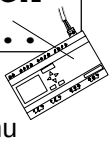
Main menu

```
>Program..
 PC/Card..
 Start
```

OK
ESC

Programming menu

```
>Edit Prg
 Clear Prg
 Set Clock
 ASi_BUS..
```

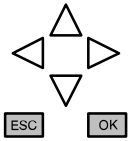


OK
ESC

PC/card menu

```
>PC↔LOGO
 LOGO→Card
 Card→LOGO
```

Parameterization mode



Parameterization menu

```
>Set Clock
 Set Param
```

3.6 Entering and starting a program

You have designed a circuit and now want to enter it in LOGO!. The example below illustrates how to do this.

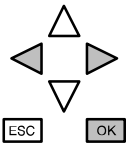
3.6.1 Switching to programming mode

You have connected LOGO! to the mains and switched on the power. The following message appears on the display:

```
No Program
```

Switch LOGO! to programming mode. To do this, press the keys ◀, ▶ and **OK** simultaneously.

```
No Program
```

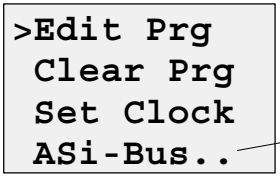


The fact that you have to press the keys simultaneously prevents anyone pressing them and switching to programming mode inadvertently. When you press the keys, LOGO!'s main menu appears:

```
>Program..
 PC/Card..
 Start
```

LOGO!'s main menu

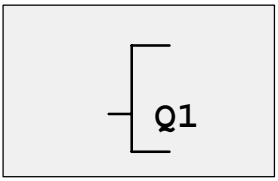
On the left in the first line you will see a ">". You press the ▲ and ▼ keys to move the ">" up and down. Move the ">" to "Program..", and press the **OK** key. LOGO! switches to the programming menu:



LOGO!'s programming menu

The ASi-Bus.. entry only appears with LOGO!...LB11 versions

Here too, you can move the ">" by pressing the ▲ and ▼ keys. Position the ">" on "Edit Prg" (i.e. to enter the program), and press the **OK** key. LOGO! then shows you the first output:

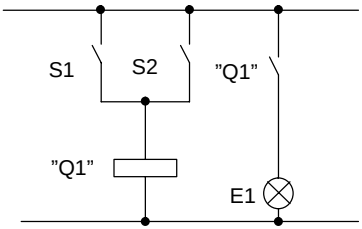


LOGO!'s first output

You can use the ▲ and ▼ keys to select the other outputs. At this point, you begin to enter your circuit.

3.6.2 First program

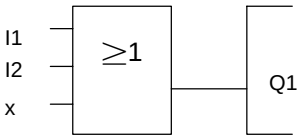
Let's have a look at the following circuit: a parallel connection of two switches. In the circuit diagram, the circuit looks like this:



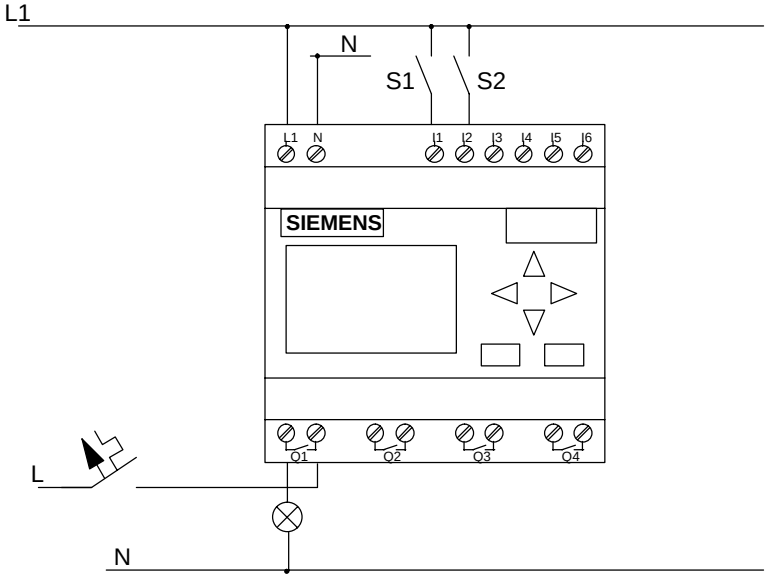
The consumer is switched on by switch S1 or switch S2. As far as LOGO! is concerned, the parallel connection of the switches is an OR block, because S1 **or** S2 switches the output on.

Translated into the LOGO! program, this means: Relay K1 (in LOGO!: Q1) is controlled by an OR block. I1 and I2 are connected to the input of the OR block, S1 to I1 and S2 to I2.

Thus, the program in LOGO! looks like this:



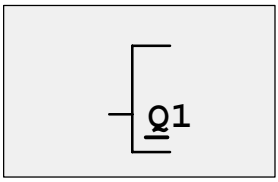
The wiring is as follows:



Switch S1 acts on input I2, and switch S2 acts on input I2. The consumer is connected to relay Q1.

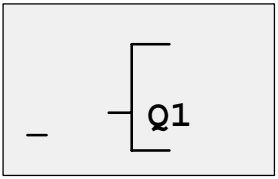
3.6.3 Entering the program

Let's enter the program now (from the output to the input). Initially, LOGO! displays the output:



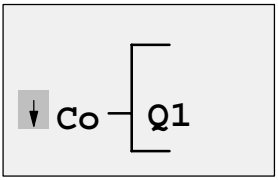
LOGO!'s first output

The Q of Q1 is underlined. This underlining is the **cursor**. The cursor indicates your current position in the program. You can move the cursor by pressing the ▲, ▼, ◀ and ▶ keys. Now press the ◀ key. The cursor moves to the left.



The cursor indicates your position in the program.

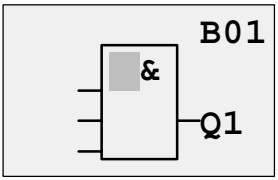
At this point, enter only the first block (the OR block). Press the OK key to switch to input mode.



The cursor appears in the form of a solid block: You can select a connector or block.

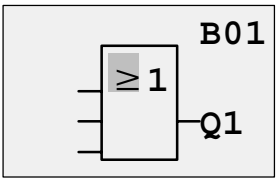
The cursor no longer appears in the form of an underline; instead, it appears as a solid block that flashes on and off. At the same time, LOGO! offers you the first list for selection. Read chapter 3.7 to find out what a list is.

Select the GF list (by pressing the ▼ key until GF appears), and press the **OK** key. LOGO! then displays the first block in the list of basic functions:



The first block in the list of basic functions is AND. The cursor appears in the form of a solid block, indicating that you have to select a block.

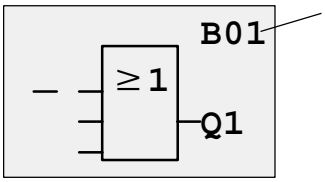
Press the ▲ or ▼ key until the OR block appears in the display:



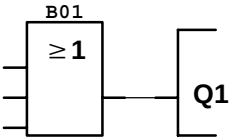
The cursor is still in the block and appears in the form of a solid block.

Press the **OK** key to conclude your selection.

The following appears in the display panel



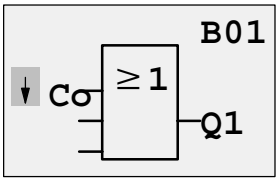
Your entire program looks like this



You have now entered the first block. Every block you enter receives a number, the block number. All you have to do now is wire the inputs of the block. To do this:

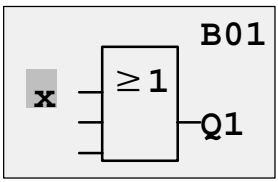
Press the **OK** button:

The following appears in the display panel

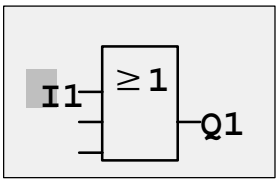


Select the Co list: Press the **OK** key

The following appears in the display panel

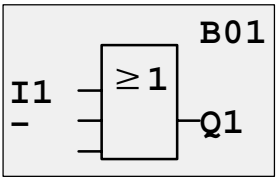


The first item in the Co list is the character for indicating that an input is not used, an ”x”. Use the ▲ or ▼ key to select input I1.

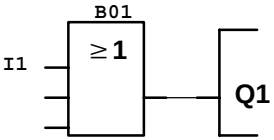


Press the **OK** key: I1 is connected to the input of the OR block. The cursor jumps to the next input of the OR block.

The following appears in the display panel



Your entire program so far looks like this in LOGO!

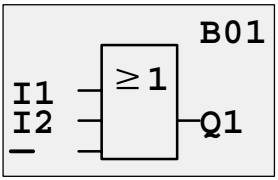


Now connect input I2 to the input of the OR block. You know how to do this already:

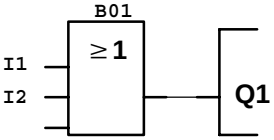
1. Switch to input mode: **OK**
2. Select the Co list: ▲ or ▼
3. Accept the Co list: **OK**
4. Select I2: ▲ or ▼
5. Accept I2: **OK**

Thus, I2 is now connected to the input of the OR block:

The following appears in the display panel



Your entire program so far looks like this in LOGO!

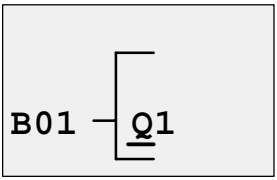


We do not need the last input of the OR block in this program. In a LOGO! program, you mark an input that is not used with an ”x”, so enter the ’x’ now (you know the principle already):

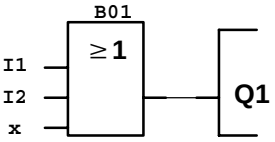
1. Switch to input mode: **OK**
2. Select the Co list: ▲ or ▼
3. Accept the Co list: **OK**
4. Select x: ▲ or ▼
5. Accept x: **OK**

Thus, all the block’s inputs are now wired. As far as LOGO! is concerned, the program is now complete. LOGO! returns to output Q1.

The following appears in the display panel



Your program looks like this



If you want to have another look at your first program, you can use the ◀ or ▶ key or the cursor to move through the program.

But we are going to exit program input now. To do this, proceed as follows:

1. Return to the programming menu: **ESC**

If this does not return you to the programming menu, you have not wired a block completely. LOGO! displays the point in the program at which you forgot something (LOGO! only accepts complete programs, which is very much in your interests). Read also page 46 on this.

Note

LOGO! has now stored your program permanently, so that it will not be lost in the event of a power failure. The program is stored in LOGO! until you expressly delete it by entering the appropriate command.

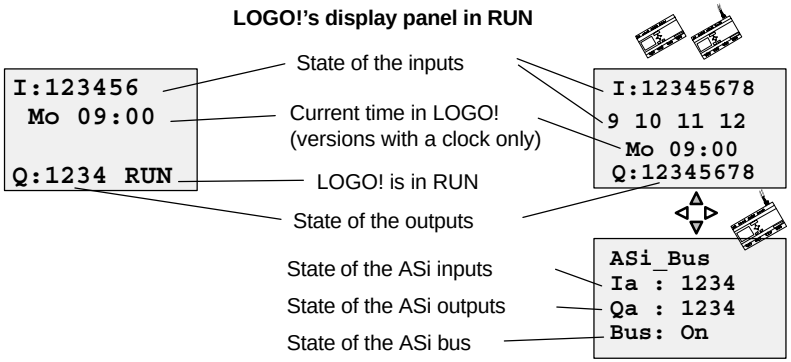
---

2. Return to the main menu: **ESC**

Switching LOGO! to RUN

3. Move '>' to 'Start': ▲ or ▼
4. Accept Start: **OK**

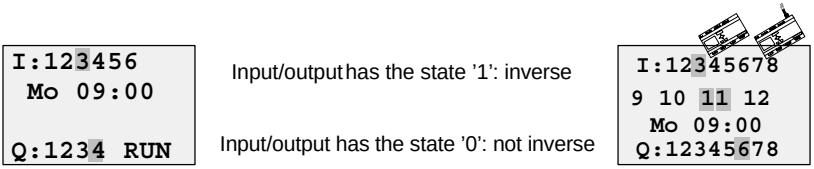
LOGO! switches to RUN. In RUN, LOGO! displays the following:



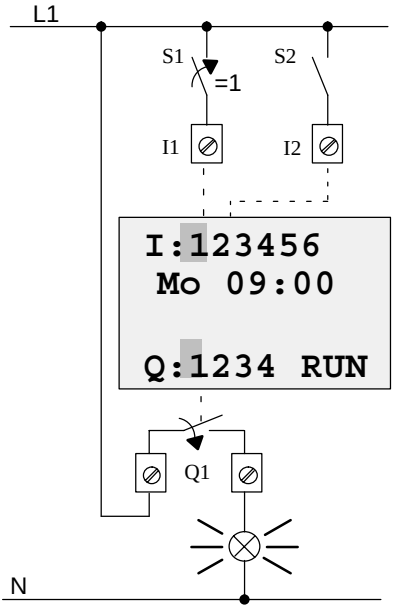
What do we mean when we say "LOGO! is in RUN?"

In RUN, LOGO! executes the program. It reads the states of the inputs, uses the program you have specified to determine the states of the outputs, and switches the relays at the outputs on or off.

LOGO! represents the state of an input or output as follows:



Let’s have a look at that in our example:



When switch S1 is closed, voltage is applied to input I1, which has the state ’1’.

LOGO! uses the program to calculate the states of the outputs.

Output Q1 has the state ’1’ here.

When Q1 has the state ’1’, LOGO! operates relay Q1 and the consumer at Q1 is supplied with voltage.

The next step

You have now successfully entered your first circuit.

In the next chapter, we will show you how to make changes to existing programs and use special functions in them.

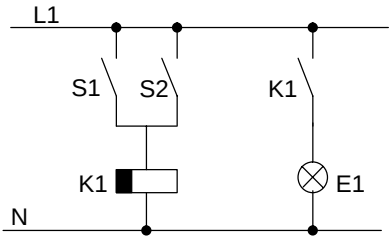
3.6.4 Second program

We use the second program to show you:

- How to insert a block in an existing program
- How to select a block for a special function
- How to enter parameters

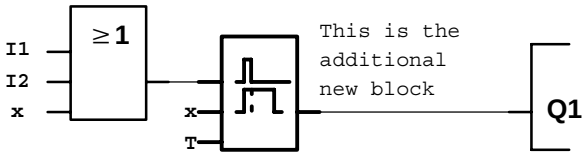
In order to produce the second program, we modify the first one.

Let's begin by looking at the circuit diagram for the second program:



You know the first part of the circuit already. Switches S1 and S2 operate a relay. The relay switches on consumer E1 and switches it off after a delay of 12 minutes.

In LOGO!, the program looks like this:



You will recognize the OR block and the output relay Q1 from the first program. Only the off-delay is new.

You modify your first program as follows:

Switch LOGO! to editing mode.

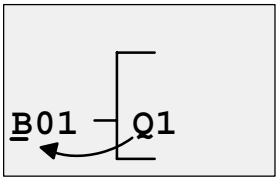
To do this, proceed as follows:

1. Switch LOGO! to programming mode
   (by pressing the ◀, ▶ and **OK** keys simultaneously)
2. Select "Program.." from the main menu
   (by moving '>' to "Program.." and pressing the **OK** key)
3. Select "Edit Prg" in the programming menu
   (by moving '>' to "Edit Prg" and pressing the **OK** button)

You can now modify the existing program.

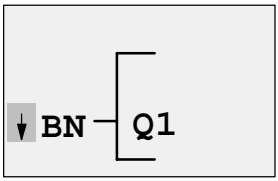
Inserting an additional block in a program

Move the cursor to the B of B01 (B01 is the block number of the OR block).



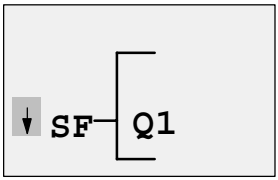
Move the cursor:
Press ◀

At this point we insert the new block. Press the **OK** button:



LOGO! displays the BN list.

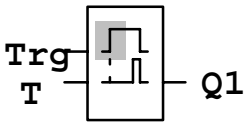
Select the SF list (▼ key).



The SF list contains the blocks for the special functions

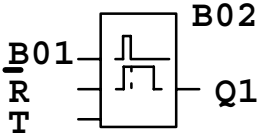
Press the **OK** key.

The block of the first special function appears:



When you select a block for a special or basic function, LOGO! displays the block of the function. The cursor is positioned in the block and itself appears in the form of a solid block. Use the ▼ or ▲ key to select the desired block.

Select the desired block (off-delay, see next diagram), and press the **OK** key:



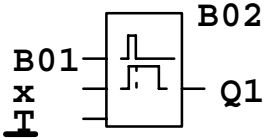
The inserted block receives the block number B02. Block B01, which has been connected up to now to Q1, is connected automatically to the uppermost input of the inserted block. The cursor is positioned at the uppermost input of the inserted block.

The off-delay block has 3 inputs. The uppermost input is the trigger input (Trg). You use this input to start the off-delay. In our example, the off-delay is started by the OR block B01. You reset the time and output by means of the reset input, and you set the time for the off-delay at T. You reset the time and output by means of the reset input, and you set the time for the off-delay by means of T parameter.

In our example, we do not use the reset input of the off-delay. We wire it with 'x'. You learned how to do this in the first program, but just to remind you, here is the procedure again:

1. Position the cursor under the R: ▲ or ▼
2. Switch to input mode: **OK**
3. Select the Co list: ▲ or ▼
4. Accept the Co list: **OK**
5. Select 'x': ▲ or ▼
6. Accept 'x': **OK**

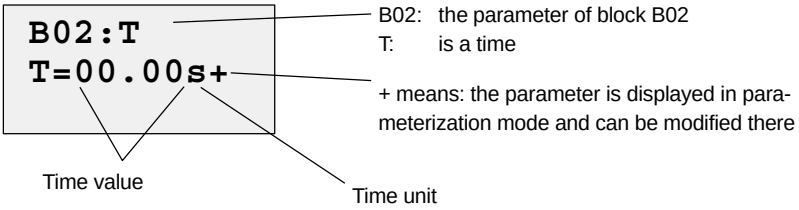
The display should now look like this:



Now enter the time T for the off-delay:

1. If the cursor is not yet under the T, move it there: ▲ or ▼
2. Switch to input mode: **OK**

LOGO! displays the parameter window for parameters:



The cursor appears on the first position of the time value.

To change the time value, proceed as follows:

Use the keys ◀ and ▶ to move the cursor to the different positions.

Use the keys ▲ and ▼ to change the value.

If you have entered the time value, press the **OK** key.

Set the time to 12:00 minutes (T = 12:00):

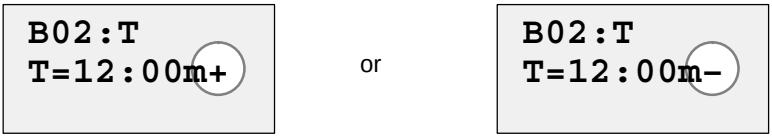
1. Move the cursor to the first position: ◀ or ▶
2. Select '1': ▲ or ▼
3. Move the cursor to the second position: ◀ or ▶
4. Select '2': ▲ or ▼
5. Move the cursor to the unit: ◀ or ▶
6. Select the unit m for minutes: ▲ or ▼

Displaying/hiding a parameter – Type of protection

If you do not want the parameter to be displayed in parameterization mode:

7. Move the cursor to the protection mode: ◀ or ▶
8. Select the protection mode '–': ▲ or ▼

You should now see the following on the display:



Type of protection +: time T can be changed in parameterization mode

Type of protection –: time T cannot be changed in parameterization mode

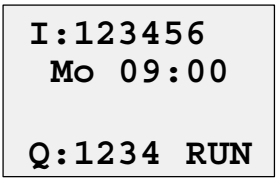
9. Conclude your input: **OK**

This branch of the program for Q1 is now complete. LOGO! displays the Q1 output. You can have another look at the program on the display. Use the keys to move through the program. Use ◄ or ► to move from block to block, and use ▲ and ▼ to move between the inputs on a block.

You exit program input in the same way as you did for the first program, but just to remind you, here is the procedure again:

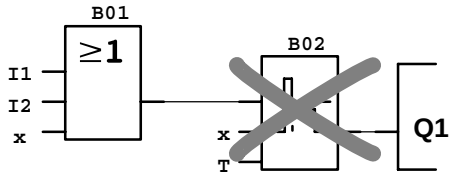
1. Return to the programming menu: **ESC**
2. Return to the main menu: **ESC**
3. Move '>' to 'Start': ▲ or ▼
4. Accept 'Start': **OK**

LOGO! is now in RUN again:



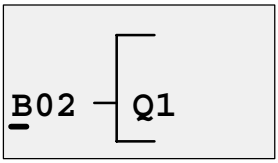
3.6.5 Deleting a block

Let's suppose you want to delete block B02 from the following program and connect B01 directly to Q1.



To do this, proceed as follows:

1. Switch LOGO! to programming mode (3-finger grip).
2. Select 'Edit Prg' by pressing **OK**.
3. Position the cursor at the input of Q1, i.e. under B02, using the ◀ key:

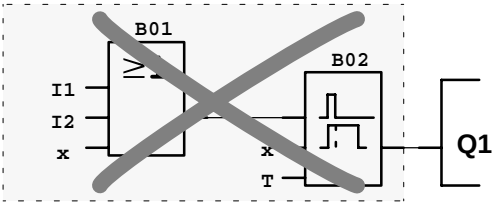


4. Press the **OK** key.
5. Connect block B01 instead of block B02 directly to output Q01:
   Select the BN list, and then press OK.
   Select B01, and then press OK.

Result: Block B02 is now deleted, because it is no longer used anywhere within the entire circuit. Block B01 is now connected directly to the output instead of block B02.

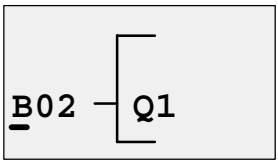
3.6.6 Deleting a number of interconnected blocks

Let's suppose you want to delete blocks B01 and B02 from the following program.



To do this, proceed as follows:

1. Switch LOGO! to programming mode (3-finger grip).
2. Select 'Edit Prg' by pressing **OK**.
3. Position the cursor at the input of Q1, i.e. under B02:



4. Press the **OK** key**.**
5. Set the connector x instead of block B02 at the Q1 output:
   Select the Co list, and then press OK.
   Select x, and then press OK.

Result: Block B02 is now deleted, because it is no longer used anywhere within the entire circuit, and all blocks that are connected to it are deleted with it (i.e. block B01 in the example).

3.6.7 Correcting typing errors

It is easy to correct typing errors in LOGO!:

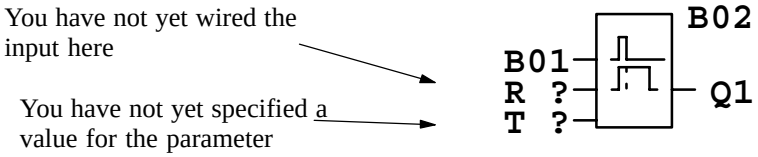
- If you have not yet concluded input, you can use ESC to go back a step
- If you have already concluded input, simply start again:

1. Move the cursor to the location of the error
2. Switch to input mode: **OK**
3. Enter the correct wiring for the input.

You can only replace one block with another if the new block has exactly the same number of inputs as the old one. However, you can delete the old block and insert a new one. You can insert whichever block you like.

3.6.8 "?" on the display

If you have entered a program and want to exit Edit Prg with ESC, LOGO! checks whether you have wired all the inputs of all the blocks correctly. If you have forgotten an input or parameter, LOGO! displays the first place at which you have forgotten something and marks with a question mark all those inputs and parameters that have not been wired.



Wire the input, and enter a value for the parameter. You can then exit Edit Prg by pressing the **ESC** key.

3.6.9 Deleting a program

To delete a program, proceed as follows:

1. Switch LOGO! to programming mode:
   ◀, ▶ and **OK** simultaneously

```
>Program..
 PC/Card..
 Start
```

2. Move the '>' to 'Program..' using the ▲ or ▼ key, and press **OK**

LOGO! switches to the programming menu:

```
>Edit Prg
 Clear Prg
 Set Clock
```

3. Move the '>' to 'Clear Prg': ▲ or ▼
4. Accept 'Clear Prg': **OK**

To prevent you from inadvertently deleting your program, we have included an additional query:

```
 Clear Prg
>No
 Yes
```

If you do not want to delete the program, leave the '>' on 'No', and press the **OK** key.

If you are sure that you want to delete the program stored in LOGO!:

5. Move the '>' to Yes: ▲ or ▼
6. Press **OK**

LOGO! deletes the program and then returns to the programming menu:

```
 Edit Prg
>Clear Prg
 Set Clock
```

3.7 Functions

LOGO! offers you a number of elements in programming mode. So that you don't lose track of things, we have divided these elements into 'lists'. These lists are:

- ↓**Co**: list of connectors for
  - inputs: I1, ...
  - outputs: Q1, ...
  - level: lo, hi
  - not connected: x
- ↓**GF**: list of the basic functions AND, OR, ... (see chapter 3.8)
- ↓**SF**: list of the special functions (see chapter 3.9)
- ↓**BN**: list of the blocks already configured in the circuit and reusable

Contents of the lists

All the lists display elements available in LOGO!. In the normal case, these are all connectors, all basic functions and all special functions that the respective LOGO! version knows. In addition, these elements include all blocks that you have already generated in LOGO! before you call up the ↓**BN** list.

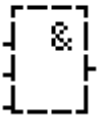
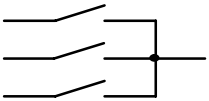
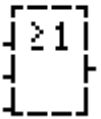
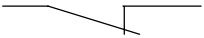
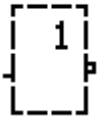
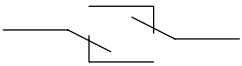
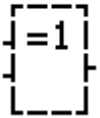
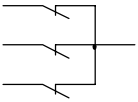
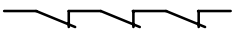
When LOGO! no longer displays everything

LOGO! no longer displays all elements if

- no further block must be inserted.
  In this case, there is either no more memory available or the maximum number of possible blocks has been reached (30).
- a special block would use more memory than is still available in LOGO!
- the resulting number of blocks connected in series would exceed 7.

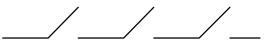
3.8 Basic functions – BF

When you enter a circuit, you will find the blocks for basic functions in the GF list. The following basic functions exist:

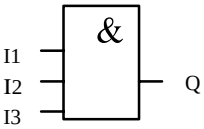
Circuit diagram representation	Representation in LOGO!	Basic function
 Series connection of normally open contacts		AND
 Parallel connection of normally open contacts		OR
 Inverter		NOT
 Double changeover contact		XOR (exclusive or)
 Parallel connection of normally closed contacts		NAND (and not)
 Series connection of normally closed contacts		NOR (or not)

3.8.1 AND

The series connection of a number of normally open contacts is represented in a circuit diagram as follows:



The symbol for AND is as follows:



The block is called AND because its output (Q) has the state 1 only when I1 **and** I2 **and** I3 have the state 1 (i.e. they are closed).

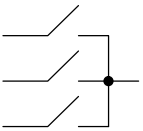
Logic table for AND:

I1	I2	I3	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

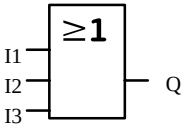
The following applies to AND: x = 1
(x means the input is not used)

3.8.2 OR

The parallel connection of a number of normally open contacts is represented in a circuit diagram as follows:



The symbol for this is as follows:



The block is called OR because its output (Q) always has the state 1 when I1 **or** I2 **or** I3 has the state 1 (i.e. closed). In other words, at least one input must have the state 1.

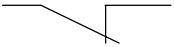
Logic table for OR:

I1	I2	I3	Q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

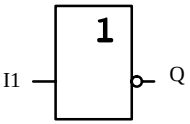
The following applies to OR: x = 0
(x means the input is not used)

3.8.3 NOT

An inverter is represented in a circuit diagram as follows:



In LOGO! the inverter is called NOT: The symbol for this is as follows:



The block is called NOT because the output (Q) has the state 1 when the input has the state 0, and vice versa. In other words, NOT inverts the state at the input.

The advantage of NOT is, for example, that you no longer require any normally closed contacts for LOGO!. You can use a normally open contact and convert it to a normally closed contact using the NOT block. The symbol for NOT is as follows:

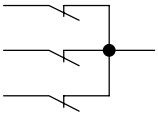
Logic table for the NOT

I1	Q
0	1
1	0

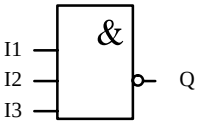
The following applies to NOT: x = 1
(x means the input is not used)

3.8.4 NAND

The parallel connection of a number of normally closed contacts is represented in a circuit diagram as follows::



In LOGO! this is a NAND block. The symbol for it is as follows:



The block is called NAND because its output (Q) only has the state 0 if I1 **and** I2 **and** I3 have the state 1 (i.e. are closed).

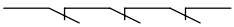
Logic table for NAND

I1	I2	I3	Q
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

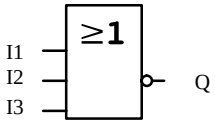
The following applies to NAND: x = 1
(x means the input is not used)

3.8.5 NOR

The series connection of a number of normally closed contacts is represented in a circuit diagram as follows:



In LOGO! this is a NOR block. The symbol for NOR is as follows:



The output of the NOR block is only switched on (state 1) when all the inputs are switched off (state 0). As soon as any of the inputs is switched on (state 1), the output is switched off.

The block is called NOR because its output (Q) only has the state 1 when all the inputs have the state 0. As soon as any of the inputs takes on the state 1, the output of NOR has the state 0.

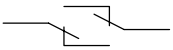
Logic table for NOR

I1	I2	I3	Q
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

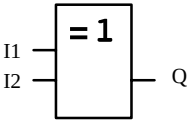
The following applies to NOR: x = 0
(x means the input is not used)

3.8.6 XOR

An XOR in a circuit diagram is a series connection of two change-over contacts:



In LOGO! the symbol for this is as follows:



The output of XOR has the state 1 when the states of the inputs differ.

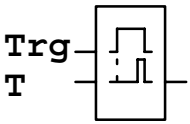
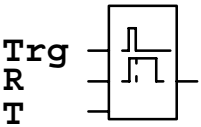
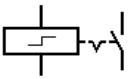
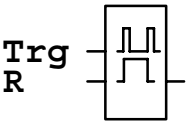
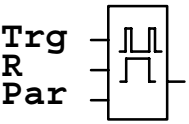
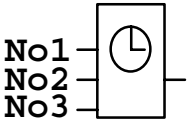
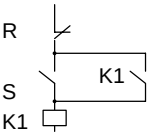
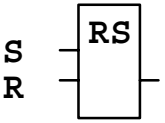
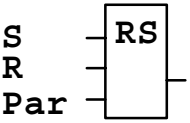
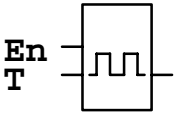
Logic table for XOR

I1	I2	Q
0	0	0
0	1	1
1	0	1
1	1	0

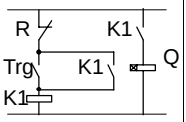
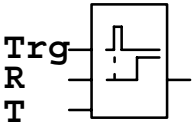
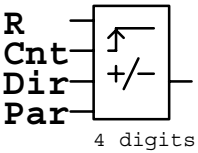
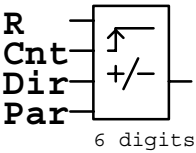
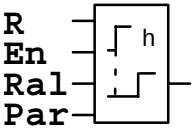
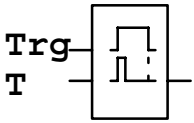
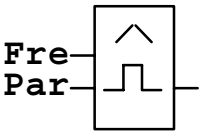
The following applies to XOR: x = 0
(x means the input is not used)

3.9 Special functions – SF

When you enter a program in LOGO!, you will find the blocks for the special functions in the SF list. The following special functions exist:

Function	Circuit diagram representation	Representation in LOGO!...	Representation in LOGO!...L...	Re
On-delay				
Off-delay				
Pulse relay				Re
Clock (time switch)				
Latching relay				Re
Clock pulse generator				

Re If there is a power failure, the state is stored as remanent if a module is inserted for remanence (with LOGO!...-L... only) and the function has been defined as remanent.

Function	Circuit diagram representation	Representation in LOGO!...	Representation in LOGO!...L...	Re
Retentive on-delay				
Up and down counter				**Re**
Operating hours counter				**Re**
Wiping relay / pulse output				
Threshold switch				

Re If there is a power failure, the state is stored as remanent if a module is inserted for remanence (with LOGO!...-L... only) and the function has been defined as remanent.

Note
In all functions, input R has priority over all other inputs.

Remanence

The following applies for **the standard version of LOGO!**:

Note

After a power failure/power restoration, in the case of time functions the time that has elapsed is reset, and in the case of the counter the counted value is reset.

In **LOGO!...-L...,** it is possible with a number of functions to store switch statuses, times and count values as remanent values. For this to be possible,

- the values in question must be defined as remanent
- a yellow or red module must be inserted that permits remanent data storage.

After a power failure, the program continues with those values that were current before the interruption.

Note

If you are running both LOGO! and its inputs by means of the same power supply, it is possible for incorrect values to be stored for the remanent functions due to power failure bridging. Under certain circumstances, this can lead to an additional edge after power restoration with edge-triggered special functions.

Make sure that you feed in the power supply for LOGO! and its inputs separately.

Connector X on the inputs of the special functions

Note

If you wire inputs of special functions to the 'x' connector, these inputs will be assigned the value 0, i.e. a low signal is applied to the inputs.

3.9.1 Accuracy of the time (all variants) and of the clock (LOGO!...C... versions)

Accuracy of T

All electronic components have minute differences. For this reason, small deviations from the time set (T) can occur. In LOGO!, the maximum deviation is 1 %.

Example:

In 1 hour (3600 seconds), the deviation is 1 %, i.e. ± 36 seconds.
In 1 minute, the deviation is therefore only ± 0.6 seconds.

Accuracy of the clock

To ensure that this deviation does not lead to the clock in C versions running inaccurately, the clock (time switch) is regularly compared with a high-precision time base and adjusted accordingly.

This means that the clock has a maximum deviation of + 5 s per day.

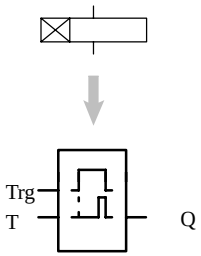
3.9.2 T parameter

With a number of the special functions described below, you have the option of parameterizing a time value T. Please note the following when setting the time:

Note

Always specify a time for T ≥ 0.10 s. For T = 0.05 s and T = 0.00 s, the time T is not defined.

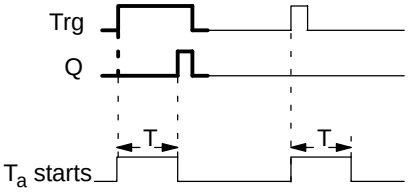
3.9.3 On-delay

Circuit diagram / Symbol in LOGO!	Wiring	Description
	Trg input	You start the time for the on-delay by means of the Trg input (TRG stands for trigger)
	T parameter	T is the time after which the output is switched on (output signal changes from 0 to 1).
	Q output	Q switches on once the parameterized time T has expired, if Trg is still set.

T parameter

Please pay attention to the note in chapter 3.9.2 when specifying the values.

Timing diagram



The bold part of the timing diagram appears in the on-delay symbol.

When the state at the Trg input changes from 0 to 1, the time T_a begins to elapse (T_a is the current time in LOGO!). If the state at the Trg input remains 1 at least for the duration of the parameterized time T, the output is set to 1 after the time T has elapsed (there is a delay between the input being switched on and the output coming on).

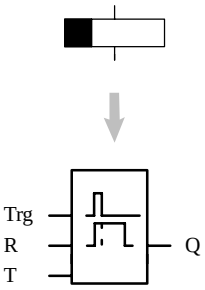
If the state at the Trg input changes back to 0 before the time T elapses, the time is reset.

The output is reset to 0 when the Trg input has the state 0.

Applications

Switch debouncing

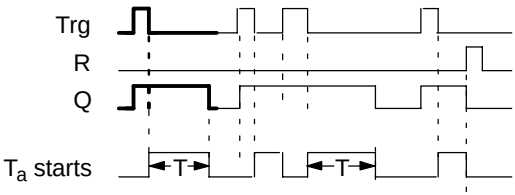
3.9.4 Off-delay

Circuit diagram / Symbol in LOGO!	Wiring	Description
	Trg input	You start the time for the off-delay by means of the Trg input (Trg stands for trigger)
	R input	You reset the time for the off-delay and set the output to 0 via the R (reset) input (R has priority over Trg)
	T parameter	T is the time after which the output is switched off (the output signal changes from 1 to 0).
	Q output	Q switches on when Trg is sent and remains switched on until T expires.

T parameter

Please pay attention to the note in chapter 3.9.2 for parameter T.

Timing diagram



The bold part of the timing diagram appears in the off-delay symbol.

When the Trg input takes on the state 1, the output (Q) switches to 1 immediately. If the state of Trg changes from 1 to 0, LOGO!'s current time T_a is started and the output remains set. If T_a reaches the values set via T (T_a=T), the output (Q) is reset to 0 (off-delay).

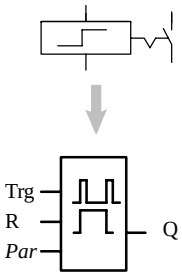
If the Trg input is switched on and off again, the time T_a starts again.

You reset the time T_a and the output via the R (reset) input before the time T_a has elapsed.

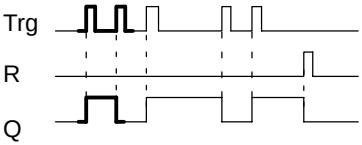
Applications

Automatic stairway lighting system

3.9.5 Pulse relay

Circuit diagram / Symbol in LOGO!	Wiring	Description
	Trg input	You use the Trg input (Trg stands for trigger) to switch the output on and off.
	R input	You use the R input (reset) to reset the pulse relay and set the output to 0 (R has priority over Trg)
	Par parameter 	Par is only available in LOGO!... L... versions. You can use this parameter to switch remanence on and off. Rem: off = no remanence on = the state can be stored as a remanent one
	Q output	Q switches on when Trg is sent and remains switched on until T expires.

Timing diagram

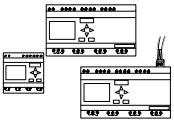
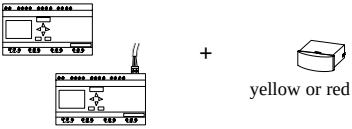


The bold part of the timing diagram appears in the pulse relay symbol.

Every time the state of the Trg input changes from 0 to 1, the state of the output (Q) changes (i.e. it is switched on or off). You reset the pulse relay to its initial state via the R input. After power on or reset, the pulse relay is reset and the output (Q) changes to 0.

Behavior after power on

Behavior after the power supply is switched on depends on the LOGO! version you are using:

	
After power on the pulse relay is always reset and the Q output always set to 0.	If remanence has not been parameterized, the pulse relay is reset and the Q output set to 0 after power on.
	If remanence has been parameterized, the state that was current before LOGO! was switched off is set after power on.

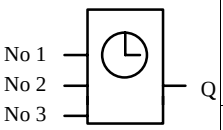
Applications

Hall/corridor lighting

3.9.6 Clock (time switch)

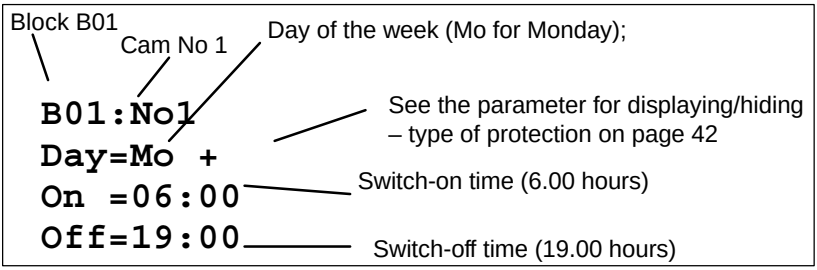
A time switch is only available in LOGO! versions that have the letter C (for clock) in their name (e.g. LOGO! 230 R**C**).

Each clock has 3 cams.

Symbol in LOGO!	Wiring	Description
	Parameter No 1, No 2, No 3	You use the “No” parameters to set the switch-on and switch-off times for the three cams of the clock (see also ”Setting the clock (time switch)”).
	Q output	Q switches on if one of the cams parameterized is switched on.

Parameter No1, No2, No3

The parameter window for cam No1 is as follows, for example:



Day of the week

The following options are available to you for setting the days of the week:

- Su Sunday
- Mo Monday
- Tu Tuesday
- We Wednesday
- Th Thursday
- Fr Friday
- Sa Saturday
- Mo..Fr Every day from Monday to Friday
- Mo..Sa Every day from Monday to Saturday
- Mo..Su Every day from Monday to Sunday (i.e. every day)
- Sa..Su Saturday and Sunday

Switch-on time

Any time between 00:00 and 23:59 hours
—:— means there is no switch-on time

Switch-off time

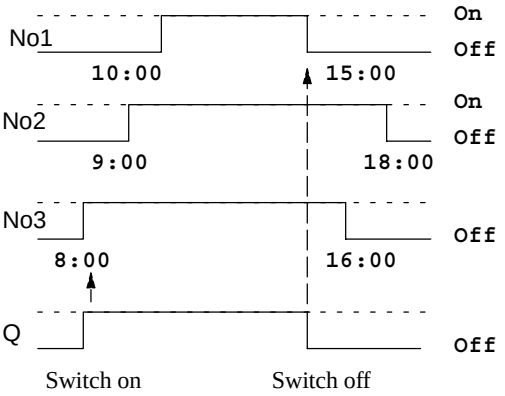
Any time between 00:00 and 23:59 hours
—:— means there is no switch-off time

Clock buffer

In LOGO! ...C, the internal clock continues to run when there is a power failure. In other words, the clock has reserve power. How much reserve power LOGO! ...C has depends on the ambient temperature. At a temperature of 25 °C, it has reserve power for a typical duration of 80 hours.

Cam overlap

You use the cams to set switch-on and switch-off times. At a switch-on time, the clock switches the output on unless it was already on; at a switch-off time, it switches the output off unless it was already off.



Carns entered for the clock

Priorities when setting identical switch-on and switch-off times

If you specify a switch-on time and a switch-off time at the same time for different cams, the switch-on/switch-off times contradict each other. In this case, cam No3 has priority over cam No2, and cam No2 has priority over cam No1.

3.9.7 Setting the clock (time switch)

To enter switching times, proceed as follows:

1. Position the cursor on one of the clock's No parameters (e.g. No1).
2. Press the **OK** key. LOGO! opens the parameter window for the cam. The cursor is positioned on the day of the week.
3. Use the ▲ and ▼ keys to select one or more days of the week.
4. Use the ▶ key to move the cursor to the first position for the switch-on time.
5. Set the switch-on time.
   You use the ▲ and ▼ keys to change the value. To move the cursor from one position to another, you use the ◀ and ▶ keys.
   You can only select the value —:–at the first position (—:— means no switching operation).
6. Use the ▶ key to move the cursor to the first position for the switch-off time.
7. Set the switch-off time (same procedure as for step 5).
8. Conclude your input by pressing the **OK** key.

   The cursor is positioned at parameter No 2 (cam 2). You can now parameterize another cam (as described under points 2. to 8.).

Note

Please refer to the Technical data in appendix A and chapter 3.9.1 for information regarding the accuracy of the clock (time switch).

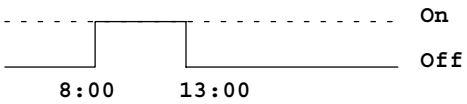
3.9.8 Clock: examples

You can use the clock to combine switch-on and switch-off times however you like. Here are some examples:

Example 1

The clock's output is to be switched on every day (i.e. from Monday to Sunday) from 08:00 hours to 13:00 hours:

B01:No1
Day= Mo..Su
On =08:00
Off=13:00

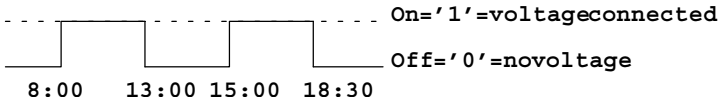


Example 2

The clock's output is to be switched on every day from 08:00 hours to 13:00 hours and from 15:00 hours to 18:30 hours. You need 2 cams for this:

B01:No1
Day= Mo..Su
On =08:00
Off=13:00

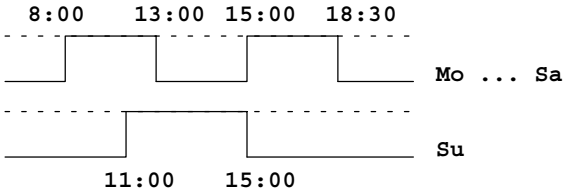
B01:No2
Day= Mo..Su
On =15:00
Off=18:30



Example 3

The clock's output is to be switched on every day from Monday to Saturday from 08:00 hours to 13:00 hours and from 15:00 hours to 18:30 hours. In addition, it is also to be switched on on Sunday between 11:00 hours and 15:00 hours. You need 3 cams for this:

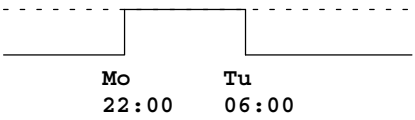
B01:No1	B01:No2	B01:No3
Day= Mo..Sa	Day= Mo..Sa	Day= Su
On =08:00	On =15:00	On =11:00
Off=13:00	Off=18:30	Off=15:00



Example 4

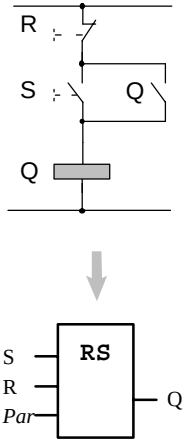
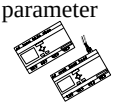
The clock's output is to be switched on on Monday at 22:00 hours and off on Tuesday at 6:00 hours.

B01:No1	B01:No2
Day= Mo	Day= Tu
On =22:00	On =––:––
Off=––:––	Off=06:00



3.9.9 Latching relay

Very often, a circuit is required that retains a switched-on state. This is referred to as latching. Latching is represented in a circuit diagram as follows:

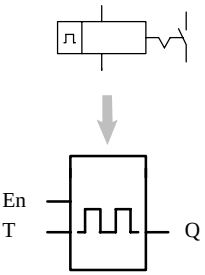
Circuit diagram / Symbol in LOGO!	Wiring	Description
	S input	You set the output (Q) to 1 via the S input (Set).
	R input	You reset the output (Q) to 0 via the R input (Reset). If S and R are both 1 at the same time, the output is reset (resetting takes priority).
	Par parameter 	Par is only available in LOGO!... L... versions. You can use this parameter to switch remanence on and off. Rem: off = no remanence on = the state can be stored as a remanent one
	Q output	Q switches on when S is sent and remains on until the R input is set.

Switching behavior

A latching relay is a simple binary flip-flop. The value of the output depends on the states of the inputs and the previous state of the output. The following table illustrates the logic once more:

S_n	R_n	Q	Note
0	0	State	remains the same
0	1	0	Reset
1	0	1	Set
1	1	0	Reset (resetting has priority over setting)

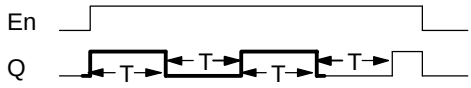
3.9.10 Symmetrical clock pulse generator

Circuit diagram / Symbol in LOGO!	Wiring	Description
	En input	You switch the clock pulse generator on and off via the En input (enable).
	T parameter	T is the time for which the output is switched on or off.
	Q output	Q switches on and off cyclically with the clock time T.

T parameter

Please pay attention to the note in chapter 3.9.2 when specifying the values.

Timing diagram



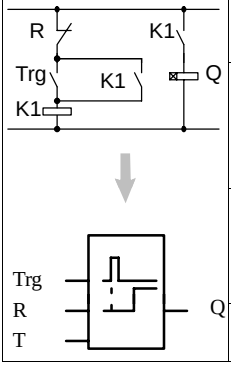
The bold part of the timing diagram appears in the symmetrical clock pulse generator symbol.

You use the T parameter to specify how long the on and off times are to last. You use the En (enable) input to switch the clock pulse generator on. The clock pulse generator sets the output to 1 for the time T, then to 0 for the time T, and so on until the En input is at 0.

Note on the relay outputs Qn:

Relay outputs that switch under load get worn a little with each switching operation. To find out how many switching operations a LOGO! output can execute, refer to the chapter entitled ”Technical data” (see chapter A).

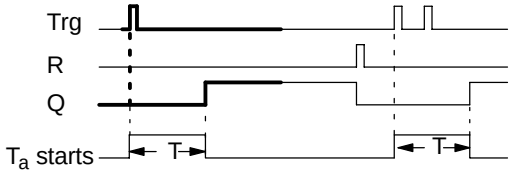
3.9.11 Retentive on-delay

Circuit diagram / Symbol in LOGO!	Wiring	Description
	Trg input	You start the time for the on-delay via the Trg (trigger) input
	R input	You reset the time for the on-delay and set the output to 0 via the R (reset) input (R has priority over Trg)
	T parameter	T is the time after which the output is switched on (the output changes from 0 to 1).
	Q output	Q switches on after the time T expires.

T parameter

Please pay attention to the note in chapter 3.9.2 when specifying the values.

Timing diagram

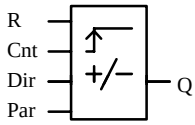


The bold part of the timing diagram appears in the retentive on-relay symbol.

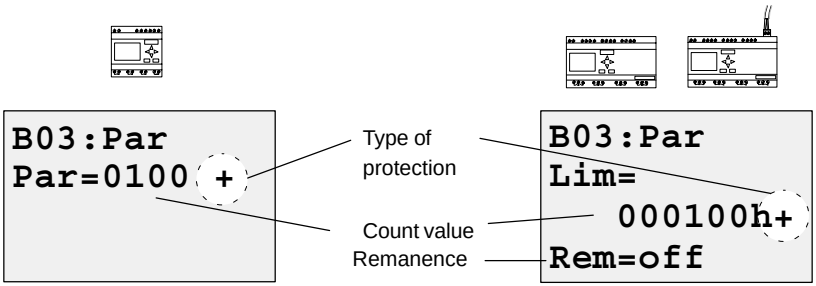
If the state of the Trg input changes from 0 to 1, the current time T_a starts. When T_a reaches the time T, the output (Q) is set to 1. Another switching operation at the Trg input has no effect on T_a.

The output and the time T_a are not reset to 0 until the state of the R input changes to 1 again.

3.9.12 Up and down counter

Symbol in LOGO!	Wiring	Description
	R input	You reset the internal count value and the output to zero via the R (Reset) input (R has priority over Cnt).
	Cnt input	The counter counts the changes from state 0 to state 1 at the Cnt (Count) input. Changes from state 1 to state 0 are not counted. Maximum count frequency at the input connectors: 5 Hz
	Dir input	You specify the count direction via the Dir (Direction) input: Dir = 0: The counter counts up Dir = 1: The counter counts down
	Par parameter	Please read the comments about Par parameter setting that follow this table.
	Q output	Q switches on when the count value (Par parameter or Lim - see below) is reached.

Par parameter setting



If the internal count value is greater than or equal to Par (Parameter) or Lim, the output is set. In the event of overrunning or underrunning, the counter stops.

Par can be anything between 0 and 9999.

Lim can be anything between 0 and 999999.

Rem: This parameter can be used in LOGO!...L... to switch remanence on and off for the internal count value Cnt.

off = no remanence

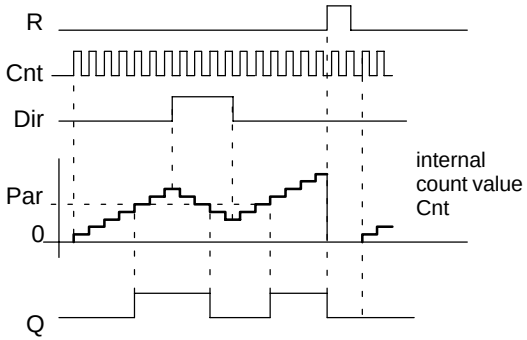
on = the count value Cnt can be stored as a remanent value

Type of protection:

+: The Par or Lim parameter can be changed during operation.

–: The Par or Lim parameter can only be changed at this point during programming. The parameter(s) cannot be changed during operation.

Timing diagram



At each positive edge at the Cnt input, the internal counter is incremented by one (Dir = 0) or decremented by one (Dir = 1). If the internal count value is greater than or equal to the value specified by Par, the output (Q) is set to 1. You can use the reset input to reset the internal count value to ‘0000’ or ’000000’. As long as R=1, the output is 0.

Remanence

Note

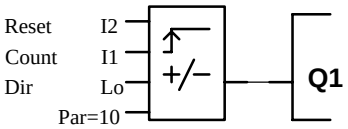
If you switch off the power supply of the standard version of LOGO!, the internal count value is deleted. After power on, the internal count value is always zero (Cnt=0000).

In LOGO!...L..., it is possible with a number of functions to store switch statuses, times and count values as remanent values. For this to be possible,

- the values in question must be defined as remanent
- a yellow or red module must be inserted, that permits remanent data storage.

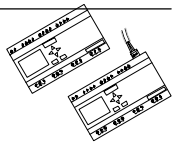
After a power failure, the program recontinues with those values that were current before the interruption.

Example

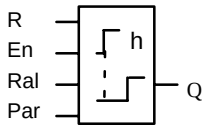


Whenever I1 takes on the state 1, the internal count value is incremented by 1. As soon as the internal count value (Cnt) reaches the value 10 set by means of Par, the output of the counter is set to 1.

3.9.13 Operating hours counter

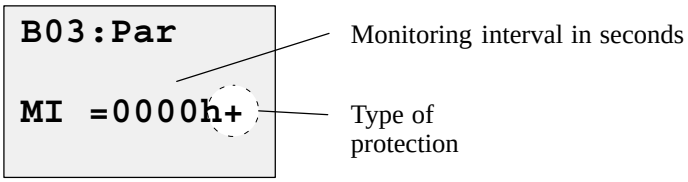


This function is only available in LOGO!...L... versions.

Symbol in LOGO!	Wiring	Description
	R input	**R = 0:** Counting is possible if Ral is not = 1 **R = 1:** The counter is stopped You reset the output via the R (Reset) input. The remaining time of the maintenance interval MN is set to MN = MI.
	En input	En is the monitoring input. LOGO! measures the time in which this input is set.
	Ral input	**Ral = 0:** Counting is possible if R is not = 1 **Ral = 1:** The counter is stopped You reset the counter and the output via the Ral (Reset all) input. I. e.: • the Q output is set to 0 • the operating hours measured (OT) = 0 • the remaining time of the maintenance interval (MN) = MI.
	Par parameter: MI	**MI:** preventive maintenance interval specified in hours. MI can be anything between 0 and 9999 hours.
	Q output	If the remaining time MN = 0 (see timing diagram), the output is set.

MI = parameterized count value
MN = remaining time
OT = overall time expired since the last 1 signal at the Ral input

Par parameter setting

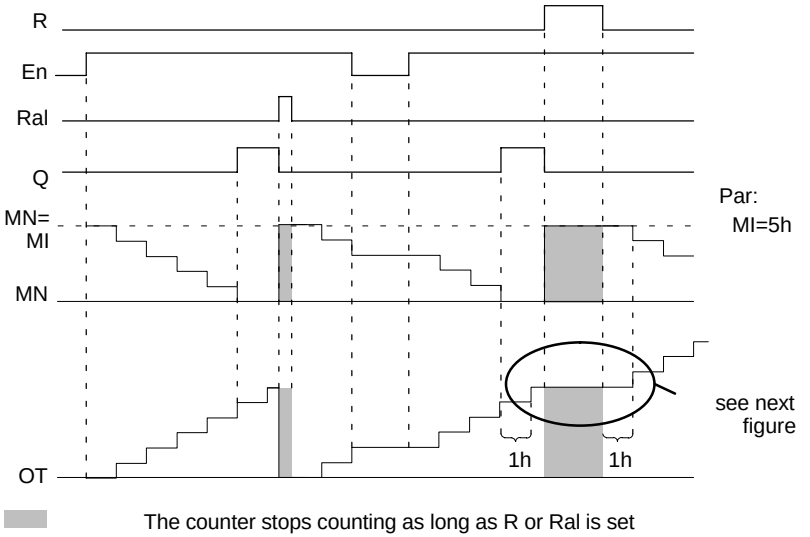


MI is the parameterizable time interval. It can be anything between 0 and 9999.

Type of protection:

+:	The specified monitoring time can be changed during operation.
–:	The specified monitoring time can only be changed at this point during programming. The time cannot be changed during operation.

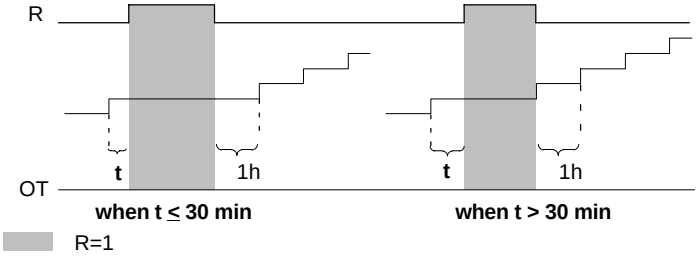
Timing diagram



The counter stops counting as long as R or Ral is set

MI = parameterizable time interval
MN = remaining time
OT = overall time expired since the last 1 signal at the Ral input

Behavior after canceling R



The operating hours counter monitors the En input. As long as this input is 1, LOGO! measures the time expired and the remaining time. LOGO! displays the times in Parameterization mode. If the remaining time is 0, the Q output is set to 1.

You use reset input R to:	You use reset input Ral to:
reset the Q output set the counter for the remaining time to the specified value MI	reset the Q output set the counter for the remaining time to the specified value MI reset the internal counter OT to 0
The internal counter OT remains unchanged	

Limit value for OT

If you reset the operating hours counter by means of the R signal, the operating hours counted in the OT counter are retained. The limit value of the OT counter is 99999 h.

When the operating hours counter reaches this value, no further hours are counted.

Remanence

In LOGO!...L..., the internal count value is always set to remanent. If you want to use this remanence, a yellow or red module must be inserted.

Note

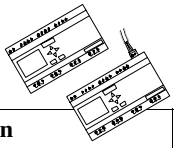
The remanence of the operating hours counter cannot be switched off. As soon as a yellow or red module is inserted, the count values for OT and MN are stored if a power failure occurs.

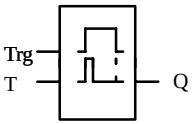
After a power failure, the program continues with those values that were current before the interruption.

Application

Monitoring maintenance intervals

3.9.14 Wiping relay – pulse output

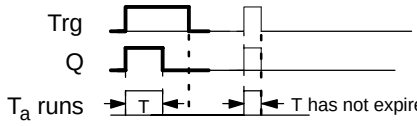


Symbol in LOGO!	Wiring	Description
	Trg input	You start the time for the wiping relay via the Trg (Trg stands for trigger) input
	T parameter	T is the time after which the output is switched off (output signal changes from 1 to 0).
	Q output	Q switches on when Trg is sent and remains switched on until T expires.

T parameter

Please refer to the note in chapter 3.9.2 for the T parameter.

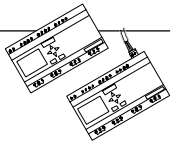
Timing diagram



The section of the timing diagram printed in bold can be found again in the symbol for the wiping relay.

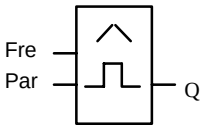
When the Trg input takes on the state 1, the Q output switches immediately to state 1. At the same time, the current time T_a starts in LOGO!, and the output remains set. When T_a reaches the value set by means of T (T_a=T), the Q output is reset to state 0 (pulse output).

If the Trg input changes from 1 to 0 before the time expires, the output also changes immediately from 1 to 0.

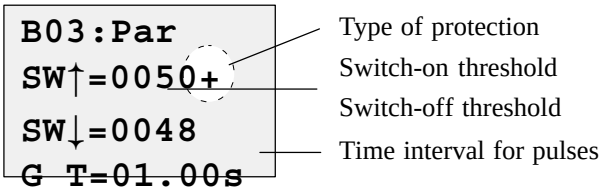


3.9.15 Threshold switch for frequencies

This function is only available in LOGO!...-L... versions.

Symbol in LOGO!	Wiring	Description
	Fre input	You apply the input that supplies the pulses to be counted to the Fre input. Use • input I12 for rapid counts (24 V inputs): max. 150 Hz • any other input or circuit section for low counting frequencies.
	Par parameter: SW↑ SW↓ G_T	SW↑: Switch-on threshold SW↓: Switch-off threshold G_T: Time interval in which the pulses applied are measured
	Q output	Q switches on or off depending on SW↑ and SW↓ (see description below).

Par parameter setting



SW↑ is the switch-on threshold. It can be anything between 0000 and 9999.

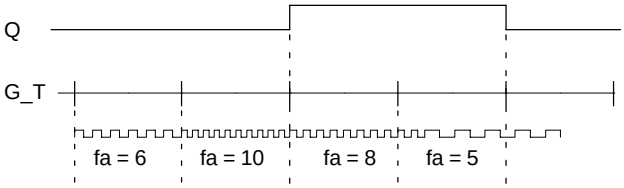
SW↓ is the switch-off threshold. It can be anything between 0000 and 9999.

G_T is the time interval in which the pulses applied to Fre are measured. G_T can be anything between 00.05s and 99.95s.

Type of protection:

+:	The definable switching thresholds can be changed during operation.
–:	The definable switching thresholds can only be changed at this point during programming. The thresholds cannot be changed during operation.

Timing diagram



Switch-on threshold: SW↑ = 9 Switch-off threshold: SW↓ = 5

The threshold switch measures the signals at the Fre input. The pulses are measures during a parameterizable period G_T. If the values measured within the time G_T are **greater** than the switch-on and switch-off thresholds, the Q output switches on.

Q switches off again when the number of pulses measured has **reached or fallen below** the value of the switch-off threshold.

Note

If you specify the time G_T as 1 s, LOGO! returns the current frequency (in Hz) in the fa parameter.

fa is always the sum of the pulses measured per time unit G_T.

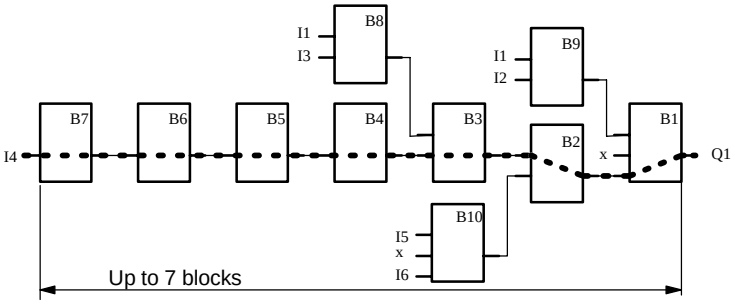
3.10 Memory required and size of a circuit

A program (or circuit diagram, if you prefer) is subject to limitations with regard to:

- The number of blocks connected in series
- The memory available

Number of blocks connected in series

You can insert a series of up to **7 blocks** between an input and an output.



Memory

LOGO! monitors memory utilization and offers in the function lists only those functions for which there actually still is enough memory space available.

At this point, we will mention just a few basic conditions that you should take into consideration:

- a program can contain a maximum of **30 blocks**.

  Just to refresh your memory: a block can be a simple AND function or even a complex special function (e.g. operating hours counter).
- if you use a number of special functions, this reduces the number of blocks possible.

Please read appendix C to find out how to determine the amount of memory that is in use.

4 LOGO!’s program modules

You can copy the program stored in LOGO! to a program module /card. You can insert the program module/card in a different LOGO! and copy the program to it. You can use the program module/card to:

- archive programs
- duplicate programs
- send programs per post
- write and test programs in the office and then transfer them to a different LOGO! in the cabinet.

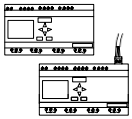
LOGO! is suppplied with cover. You receive the program module/card separately.

Note

You do **not** require a module for permanently storing the program in your LOGO!.

The LOGO! program is already stored permanently when the Programming mode is ended.

We will now introduce you to the three modules that you can buy for LOGO!. All three can accommodate the entire program memory of a LOGO!.

Module	Order number	Use
Standard module	6ED1 056–1AA00–0AA0	All LOGO! versions
Program modules with know-how protection and remanence	6ED1 056–4BA00–0AA0	
Program modules with remanence	6ED1 056–1BA00–0AA0	

4.1 Overview of the modules

We will now introduce you to the characteristics of the different modules in the various LOGO! versions.

The overview table shows you the application options for the modules. Look the symbol of your LOGO! and find out which modules you can use:

LOGO!..	Standard module (cyan)	Module for remanent data (yellow)	Module for protected programs and remanent data (red)
	Reading and writing programs Interchangeable between all versions	Cannot be used	Cannot be used
	Reading and writing programs Interchangeable between all LOGO! ...L versions	Reading and writing programs and remanent data Interchangeable between all LOGO!..L versions	Writing programs Reading and writing remanent data Programs can only be run when module is inserted
	Reading and writing programs Interchangeable between all LOGO! LB11 versions	Reading and writing programs and remanent data Interchangeable between all LOGO!..L versions	Writing programs Reading and writing remanent data Programs can only be run when module is inserted

Note

To enable remanent data to be stored in a red or yellow module, your program must contain functions whose current data can be stored as remanent data. What's more, remanence must be switched on for the parameterizable functions.

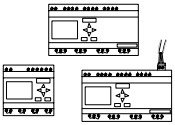
Upward compatibility

The following rule applies: the modules are only upward compatible, i. e. a module

- that has been written in a standard version can be read into all other versions.
- that has been written in a LOGO! ...L version can be read into all other LOGO! ...L versions, but not into a standard version.
- that has been written in a LOGO! ...LB11 version can be read into all other LOGO! ...LB11 versions, but not into a standard version or a LOGO! ...L version.

Standard module

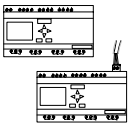
Color: cyan (blue) for



The module can be used in all LOGO! versions. The programs stored can be interchanged in accordance with the rules described above in the **Upward compatibility** section.

Program module with remanence

Color: yellow for



This module can only be used in the LOGO! ...L... versions. The programs stored can be interchanged freely between these models.

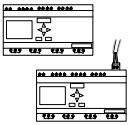
The module must always be inserted so that remanent data can be stored. In the event of a power failure and on power OFF, LOGO! saves the remanent data to the module inserted.

Functions whose data can be stored as remanent data:

Function	Remanent
Pulse relay	The switch state is stored
Latching relay	The switch state is stored
Up and down counter	The internal counter reading is stored
Operating hours counter	The time elapsed is stored

Program module with know-how protection and remanence

Color: red for



This module can only be used in the LOGO! ...L... versions. Once a program has been stored in this module, it can neither be looked at, copied nor changed. I.e. your data is protected.

The module must remain inserted in LOGO! the whole time the system is operating for the program stored in this way to run.

Warning

Make sure that you do not save your program to a module with program protection if you want to edit the program further.

The program of a module with know-how protection can only be started, but not read for the purpose of editing.

In the event of a power failure and on power OFF, LOGO! saves the remanent data to the module inserted.

4.2 Removing and inserting the program module/ card

Whenever you remove a program module, always observe the following points:

Module	Comment
Standard	You can change the program module/card when the power is on and LOGO! is in RUN or the programming mode.
Remanence	The module must be inserted so that the remanent data can be saved in the event of a power failure. The module is not required for the program to be able to run.
Know-how protection and remanence	The program stored on the module can only run if the module is inserted and remains inserted for the entire run time of the program. If the module is removed, LOGO! reports 'no program'.

In any case, however, please heed the following warning:

Warning

Only use LOGO! 230 with the cover or the program module/ card inserted.

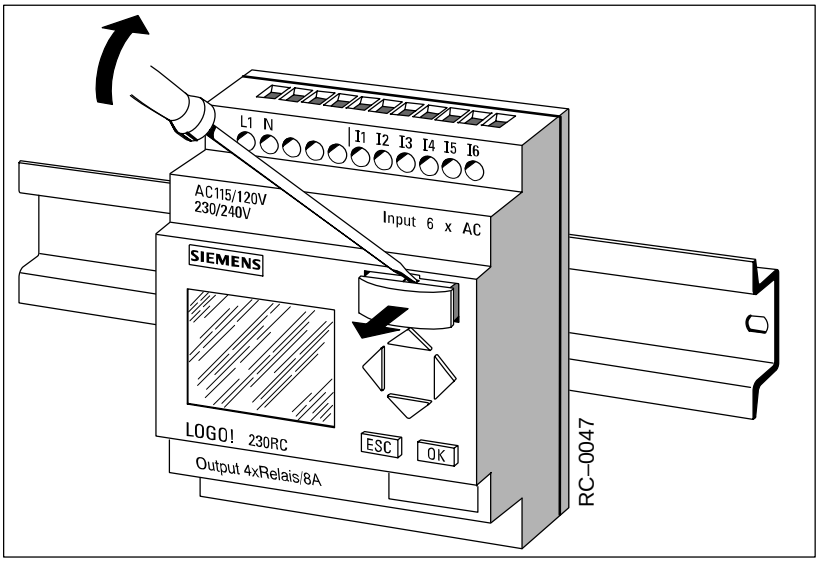
Do not put your finger or an object made of metal or any other conductive material in the open shaft of the program module/ card.

The socket for the program module/card may be live if mistakes have been made with the wiring (L1 and N mixed up).

The program module/card should only be changed by a trained technician.

Removing the module

Remove the program module/card as follows:



Carefully insert a screwdriver into the slot at the upper end of the program module/card, and ease the program module/card out of the shaft a little.

You can now remove the program module/card.

Inserting the program module/card

The shaft for the program module/card is chamfered at the bottom on the right. The program module/card also has a chamfered edge. This prevents you from inserting the program module/card the wrong way around. Insert the program module/card into the shaft until it engages.

4.3 Copying a program from LOGO! to the program module/card

To copy a program to the program module/card, proceed as follows:

1. Insert the program module/card
2. Switch LOGO! to programming mode:
   ◀, ▶ and **OK** simultaneously

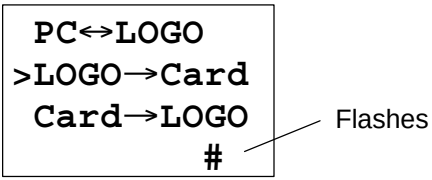
```
>Program..
 PC/Card..
 Start
```

3. Move the '>' to "PC/Card": ▼
4. Press **OK**. The transfer menu appears

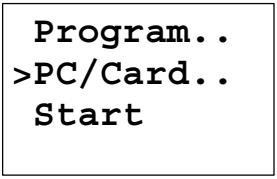
```
>PC↔LOGO
 LOGO→Card
 Card→LOGO
```

5. Move the '>' to 'LOGO → Card': ▼
6. Press **OK**.

LOGO! copies the program to the program module/card. While it is doing this, a '#' flashes on the display:



When LOGO! has finished copying, it returns to the main menu:



The program is now also on the program module/card. You can remove the program module/card. **Do not forget** to replace the cover.

If there is a power failure while LOGO! is copying, you have to copy the program again once the power has been restored.

4.4 Copying a program from the program module/ card to LOGO!

You have a program module/card containing your program. There are 2 ways to copy the program to LOGO!:

- Automatically when LOGO! starts up (power on)
- Via LOGO!'s PC/Card menu

Note

Please note that not all modules can be read in all LOGO! versions. If necessary, read chapter 4.1 once again.

Automatic copying at LOGO! startup

Proceed as follows:

1. Switch LOGO! into programming mode.
2. Switch the power off.
3. Remove the cover from the shaft.
4. Insert the program module/card in the shaft.
5. Switch the power on again.

Result: LOGO! copies the program from the program module/card to LOGO!. While LOGO! is copying, a '#' flashes on the display. As soon as LOGO! has finished copying, LOGO! displays the main menu:

```
>Program..
 PC/Card..
 Start
```

Now you can switch LOGO! to RUN:

Note

Before you switch LOGO! to RUN, you must ensure that the system you are controlling with LOGO! does not represent a source of danger.

---

1. Move the '>' to Start: 2 × ▼
2. Press **OK**

Using the PC/Card menu to copy

Read the note about changing the program module/card.

To copy a program from the program module/card to LOGO!, proceed as follows:

1. Insert the program module/card
2. Switch LOGO! to programming mode:
   ◀, ▶ and **OK** simultaneously

```
>Program..
 PC/Card..
 Start
```

3. Move the '>' to "PC/Card": ▼
4. Press **OK**. The transfer menu appears:

```
 PC↔LOGO
 LOGO→Card
>Card→LOGO
```

5. Move the '>' to 'Card → LOGO': ▲ or ▼
6. Press **OK**.

LOGO! copies the program from the progam module/card to LOGO!. When LOGO! has finished copying, it returns to the main menu:

5 Parameterizing LOGO!

By parameterization we mean setting the parameters of blocks. You can set delay times for time functions, switching times for clocks (time switches), the threshold value of a counter, the monitoring interval of an operating hours counter and the switch-on and switch-off thresholds of the threshold switch.

You can set the parameters:

- In programming mode
- In parameterization mode

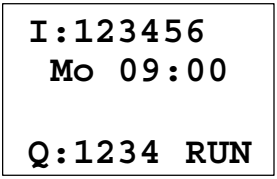
In parameterization mode, the programmer sets a value for a parameter. We introduced parameterization mode so that parameters can be changed without having to change the program. In this way, a caretaker can change times, for example, without having to change into programming mode. The advantage of this is that the program (and thus the circuit) is protected but can still be modified by the user of the circuit to suit requirements.

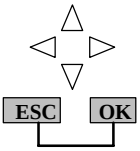
Note

LOGO! continues to execute the program in parameterization mode.

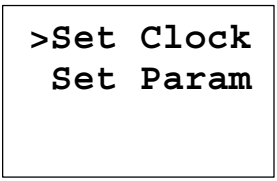
5.1 Switching to parameterization mode

To switch to parameterization mode, press **ESC** and **OK** simultaneously:





LOGO! switches to parameterization mode and displays the parameterization menu:



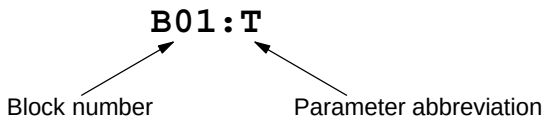
The 'Set Clock' menu item is executed only if your version of LOGO! has a clock/time switch (those versions of LOGO! that have a clock have the letter C in their name, e.g. LOGO 230 RC). 'Set Clock' allows you to set LOGO!'s clock.

5.1.1 Parameters

Parameters can be:

- The delay times of a time relay
- The switching times (cams) of a clock
- The threshold value of a counter
- The monitoring time of an operating hours counter
- The switching thresholds of a threshold switch

Every parameter is identified by the block number and the parameter abbreviation. Examples:

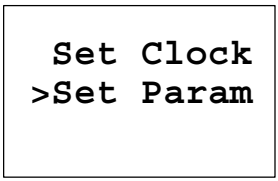


B01:T	A delay time can be set at block B01
B02:No1	Block B02 is a clock block. No1 is the first cam of this clock
B03:Par	Block B03 is a counter. Par is the threshold value of the counter
B04:Par	Block B04 is a counter in LOGO!... L.. . Par stands for a number of parameters that can be monitored.
B05:Par	Block B05 is an operating hours counter. Par stands for a number of parameters that can be monitored.
B06:Par	Block B06 is threshold switch. Par stands for a number of parameters that can be monitored.

5.1.2 Selecting a parameter

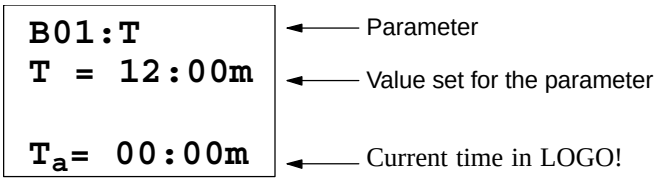
To select a parameter, proceed as follows:

1. Select the 'Set Param' option from the parameterization menu

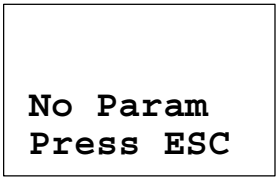


2. Press **OK**

LOGO! displays the first parameter:



If no parameter can be set, LOGO! displays the following:



No parameter can be changed:
ESC returns you to the parametrization menu

3. Select the desired parameter: ▲ or ▼

LOGO! displays a parameter in a separate window.

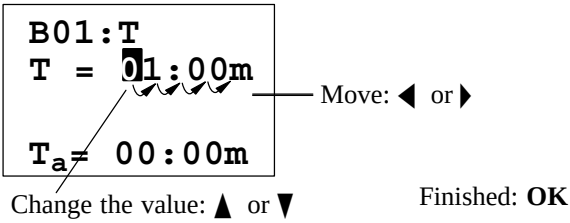
4. To change a parameter, select it and press the **OK** key.

5.1.3 Changing a parameter

To change a parameter, you first have to select it (see ”Selecting a parameter”).

You change the value of the parameter in the same way as you entered it in progamming mode:

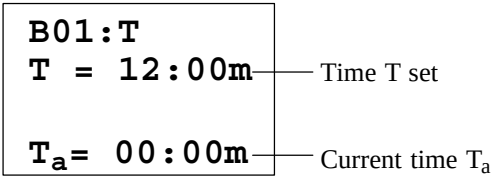
1. Move the cursor to the point at which you want to make the change: ◀ or ▶
2. Change the value: ▲ or ▼
3. Accept the value: **OK**



You cannot change the unit of the delay time for the parameter T in parameterization mode. This is only possible in programming mode.

Current value of a time T

If you view a time T in parameterization mode, it looks like this:



You can change the set time T (see ”Changing a parameter”).

Current value of the clock

If you view a cam of a clock in parameterization mode, it looks like this, for example:

```
B02:No1 1
Day = Su
On =09:00
Off=10:00
```

The switching state of the clock is displayed:

- **0** The clock is off (state '0' at the output)
- **1** The clock is on (state '1' at the output)

LOGO! displays the switching state of the clock rather than the switching state of a cam. The switching state of the clock depends on all three cams (No1, No2 and No3).

Current value of a counter (Par)

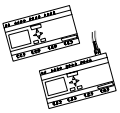
If you view the parameter of a counter in parameterization mode, it looks like this:

```
B03:Par
Par = 0300   <- Switching threshold

Cnt = 0028   <- Current count value
```

Current value of a counter in LOGO!...L...

If you view the parameter of a counter in parameterization mode, it looks like this:

```
B04:Par
Lim=000100   <- Switching threshold

Cnt=000011   <- Current count value
```

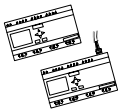


Current value of an operating hours counter

If you view the parameter of an operating hours counter in parameterization mode, it looks like this:

```
B05:Par
MI  = 0050h
MN  = 0017h
OT  =00083h
```

Monitoring time
Remaining time
Operating hours elapsed

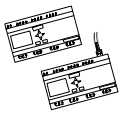


Current value of a threshold switch

If you view the parameter of a threshold switch in parameterization mode, it looks like this:

```
B06:Par
SW↑=0050
SW↓=0048
fa  =0012
```

Switch-on threshold
Switch-off threshold
Measured value



5.2 Setting the time (LOGO! ... C)

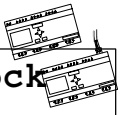
You can set the time:

- In parameterization mode
- In programming mode

Setting the time in parameterization mode:

1. Switch to parameterization mode:
   ESC and **OK** simultaneously
2. Select 'Set Clock', and press **OK**

Set Clock Day =_Mo Time=09:06	The cursor is positioned before the day of the week	Set Clock _Mo 09:06

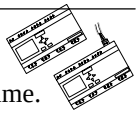


3. Select the day of the week: ▲ or ▼
4. Move the cursor to the next position: ◀ or ▶
5. Change the value at this position: ▲ or ▼
6. Set the clock to the correct time. Repeat steps 4 and 5
7. Conclude your input: **OK**

Setting the time in programming mode:

1. Switch to programming mode:
   ◀, ▶ and **OK**
2. Select 'Program..', and press **OK**
3. Select (using ▼ or ▲) 'Set Clock', and press **OK**

Now you can set the day of the week and the time, as described above (as of step 3).



Switching between summer and winter time:

LOGO!...L... must be in RUN if you want to switch over the time.

1. If necessary, exit programming or parameterization mode and switch your LOGO!...L... to RUN.

```
I:12345678
9 10 11 12
 Mo 09:17
Q:12345678
```

Time displayed

2. Press **OK** and ▲

   The current time is put forward by one hour.

```
I:12345678
9 10 11 12
 Mo 10:17
Q:12345678
```

Time displayed after change

You change the time in the opposite direction in almost the same way:

3. Press **OK** and ▼.

   The current time is put back by one hour.

```
I:12345678
9 10 11 12
 Mo 09:17
Q:12345678
```

Time displayed after change

6 LOGO!Soft

LOGO!Soft V 2.0 is a programming package for use on PCs. The software contains the following functions:

- Offline program generation of your application
- Simulation of your circuit (or your program) on the computer
- Generation and printing of a block diagram of the circuit
- Saving of the program to hard disk or another storage medium
- Program transfer
  - from LOGO! to the PC
  - from the PC to LOGO!

The alternative

LOGO!Soft therefore offers you an alternative to conventional planning:

1. i.e., you develop your applications first at your desk.
2. i.e., you simulate the application in your computer and test whether or not it functions properly before the circuit is actually put to use.
3. i.e., you print out the entire circuit in a block diagram or in a number of block diagrams sorted according to outputs.
4. i.e., you archive your circuits in your PC file system. In this way, you can retrieve a circuit directly if you want to make changes some time in the future.
5. i.e., you transfer the program to LOGO! by pressing just a few buttons. Your LOGO! is ”retooled” within a very short space of time.

6.1 Possible applications for LOGO!Soft

You can run LOGO!Soft both in conjunction with LOGO! (online) and as a standalone solution (offline).

The following requirements must be fulfilled:

LOGO!Soft without connection to LOGO! (offline)	LOGO!Soft with connection to LOGO! (online)
Compatible PC with Windows Version 3.1 or higher, Windows 95 or Windows NT	
• LOGO!Soft Version 2.0 or higher • Free space on the hard disk for full installation: 7 Mbytes	• LOGO!Soft Version 2.0 or higher • Free space on the hard disk for full installation: 7 Mbtyes • LOGO! .... • PC cable for connecting the PC to LOGO!.... .

Installating and using

Before installing LOGO!Soft, read the file Readme.txt on the Installation disk.

To install the software, simply follow the instructions given by the Installation program. This is how you start the Installation program:

1. Select the application SETUP.EXE and start it:
   – in Windows 3.1, e.g. via the File Manager
   – in Windows 95 and Windows NT 4.0 via **Start› Run** and entering A:\Setup in the command line
2. Following the instructions given by the Installation program.

The best way to find out how to use the software is by working with it on your computer. If you get stuck, simply call up the online Help of the software.

The remaining steps

In the next step, we will show you how to connect LOGO! to a PC. Skip this step if at present you only have the software available.

The second step of the chapter describes in note form the special menu items that are not available in Windows software.

6.2 Connecting LOGO! to a PC

Connecting a PC cable

To connect LOGO! to a PC, you need the LOGO! PC cable.

Remove the cover or the program module/card, and connect the cable there.

Switch LOGO! to PC↔LOGO mode

So that the PC can access LOGO!, LOGO! must be in PC ↔ LOGO mode. To switch LOGO! to PC ↔ LOGO mode:

1. Switch LOGO! to programming mode: ◀, ▶ and **OK** simultaneously
2. Select 'PC/Card': ▼ or ▲
3. Press **OK**
4. Select PC ↔ LOGO: ▼ or ▲
5. Press **OK**

LOGO! is now in PC ↔ LOGO mode, and the following appears on the display:

```
PC ↔ LOGO

STOP:
Press ESC
```

The PC can now access LOGO!. The best way to find out how this is done is to go directly into the online Help of LOGO!Soft.

To break the link to the PC, you press ESC.

Switching LOGO! to PC↔LOGO mode at startup

1. Switch the power off
2. Remove the cover or the program module/card, and connect the cable there.
3. Switch the power on

LOGO! goes into PC ↔ LOGO mode automatically

6.3 Using LOGO!Soft with LOGO!

You use LOGO!Soft with LOGO! by means of the menu entries in the LOGO! menu item. This menu item offers you the following entries:

- **Select LOGO!**: you use this entry to set LOGO!Soft to your particular version of LOGO!. This is necessary to ensure that all functions of LOGO! are supported.
- **PC→LOGO!**: you use this entry to transfer a program you have generated in LOGO!Soft to LOGO!.
- **LOGO!→PC!**: you use this entry to transfer a program you have generated in LOGO! to LOGO!Soft.
- **Set up link**: you use this entry to define the serial interface of the PC via which data is be exchanged with LOGO!.

7 Applications

To give you a feeling for the kind of situations in which you can use LOGO!, we have compiled a number of application examples. We have included the circuit diagram of the original solution for each example. For the solutions using LOGO!, we have included the wiring and a diagram.

Solutions for the following tasks are included:

Note

The LOGO! applications are available to our customers free of charge. The examples they contain are not binding and are included to provide general information on how LOGO! can be used. Customer-specific solutions may be different.

The user is responsible for ensuring that the system is run properly. We refer you to the relevant national standards and system-related installation requirements.

Errors are excepted the right to make changes reserved.

These applications – and tips for further applications – can be found in the Internet under the address http://www.AUT.Siemens.DE. Search for LOGO!..

7.1 Stairway, hall or corridor lighting

7.1.1 Demands on stairway lighting

The lighting system of a stairway should fulfill the following requirements:

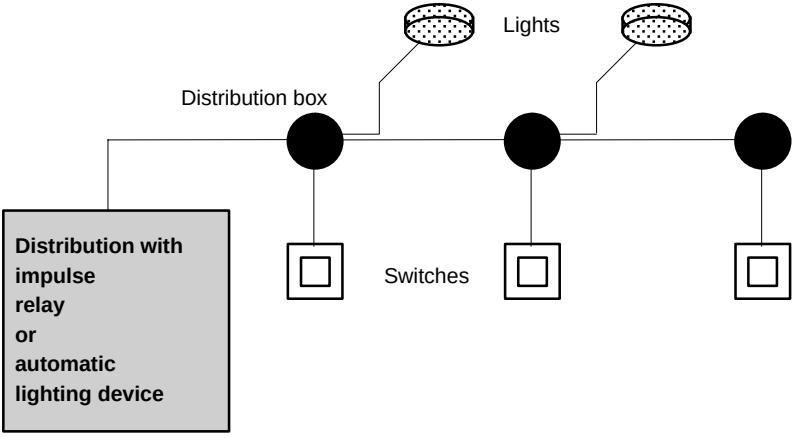
- The light should be on when someone is on the stairway.
- If there is nobody on the stairway, the light should be off to save energy.

7.1.2 Previous solution

Previously there were 2 ways of switching the lighting:

- By means of an impulse relay
- By means of automatic stairway lighting

The wiring for these two lighting systems is the same.



Components used

- Switches
- Automatic lighting device or pulse relay

Lighting system with a pulse relay

When a pulse relay is used, the lighting system behaves as follows:

- When any switch is pressed: The lighting is switched on
- When any switch is pressed again: The lighting is switched off.

Disadvantage: People often forget to switch the light off again.

Lighting system with an automatic lighting device

When an automatic device is used, the lighting system behaves as follows:

- When any switch is pressed: The lighting is switched on.
- After a preset time has elapsed, the lighting is switched off automatically.

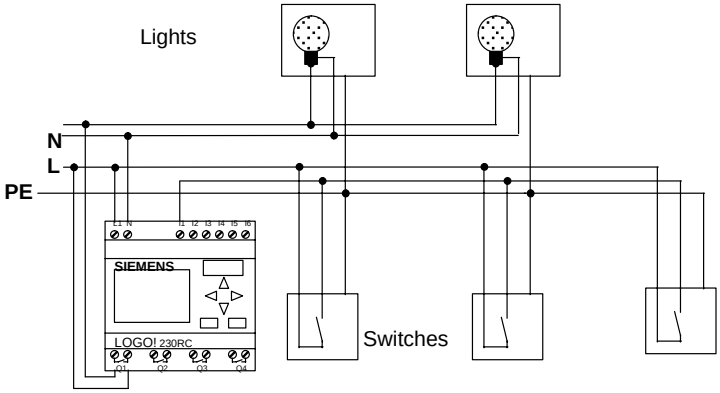
Disadvantage: The lighting cannot be switched on for an extended period of time (e.g. for cleaning purposes). The switch for permanent lighting is usually on the automatic device, which is either impossible or difficult to access.

7.1.3 Lighting system with LOGO!

If you use LOGO!, you can replace the automatic lighting device or the pulse relay. You can implement both functions (time-dependent switching-off and pulse relay) using a single device. You can also include additional functions without changing the wiring. Here are some examples:

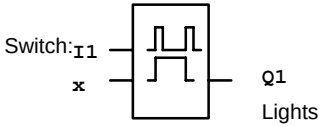
- Impulse relay with LOGO!
- Automatic stairway lighting system with LOGO!
- LOGO! as a multi-function switching system with the following functions:
  - Light on: Press switch
    (Light switches off after the set time elapses)
  - Permanent light on: Press switch twice
  - Light off: Press switch for 2 seconds

Wiring of the lighting system with LOGO! 230RC



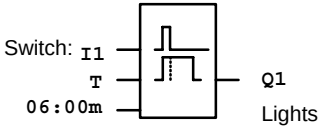
The external wiring of the lighting system with LOGO! is the same as for a conventional hall, corridor or stairway lighting system. The difference is that the automatic lighting device or the pulse relay is replaced. Additional functions are entered directly in LOGO!.

Pulse relay with LOGO!



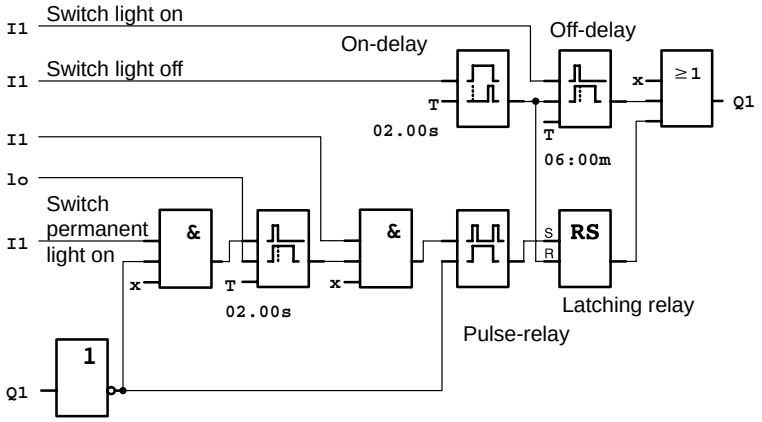
In the event of a gate pulse at input I1, output Q1 switches over.

Automatic stairway lighting system with LOGO!



In the event of a gate pulse at input I1, output Q1 switches on and remains on for 6 minutes.

Multi-functional switch with LOGO!



The diagram shows the circuit for an input with an associated output.

This switch offers the following:

- **When the switch is pressed:** The light is switched on and goes off again after the set time of 6 minutes (T=06:00m) has elapsed (off-delay)
- **When the switch is pressed twice:** The light is switched on permanently (the latching relay is set via the impulse relay).
- **When the switch is pressed for 2 seconds:** The light is switched off (on-delay switches the light off; both the permanent light and the normal light; this branch of the circuit is therefore used twice)

You can enter these circuits several times for the remaining inputs and outputs. Instead of using 4 automatic stairway lighting systems or 4 impulse relays, you thus use only a single LOGO! module. However, you can also use the free inputs and outputs for completely different functions.

7.1.4 Special features and enhancement options

Features such as the following are available for adding functions or saving energy:

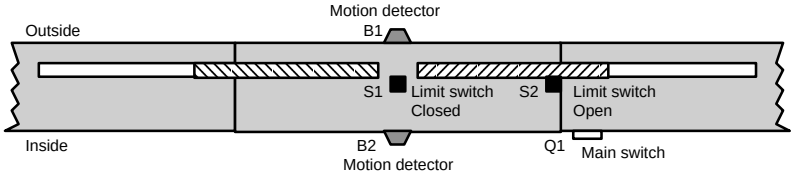
- You can have the light flash before it goes off automatically.
- You can integrate various central functions:
  - Central off
  - Central on (panic button)
  - Control of all lights or individual circuits by a daylight control switch
  - Control by the integrated time switch (clock)
    (e.g. permanent light only until 24.00 hours; no enabling at certain times)
  - Automatic switching off of permanent light after a preset time has elapsed (e.g. 3 hours)

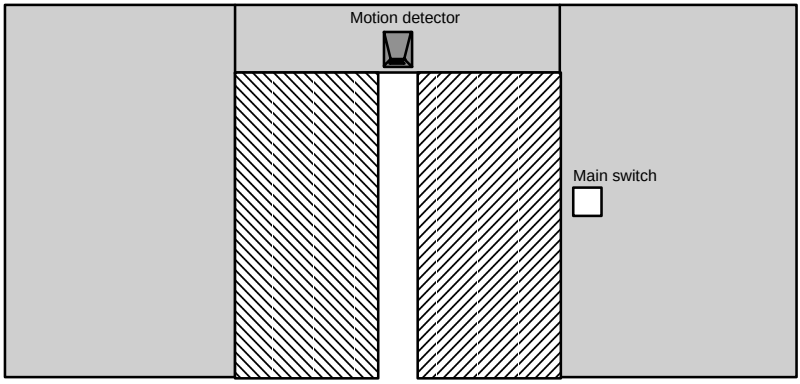
7.2 Automatic door

You often find automatic door control systems at the entrances to supermarkets, public buildings, banks, hospitals, etc.

7.2.1 Demands on an automatic door

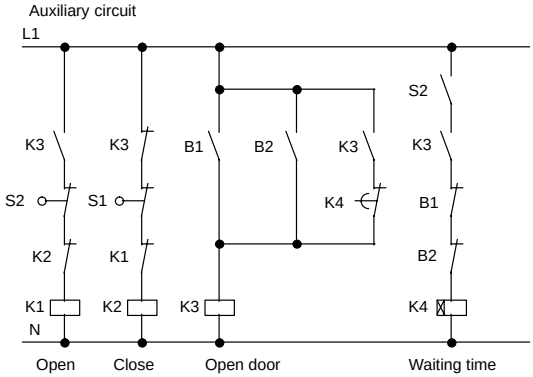
- When somebody approaches, the door must open automatically.
- The door must remain open until there is nobody in the doorway any more.
- If there is nobody in the doorway anymore, it must close automatically after a short time.





The door is generally driven by a motor with a safety clutch. This prevents people from being caught or injured in the door. The control system is connected to the mains via a main switch.

7.2.2 Previous solution



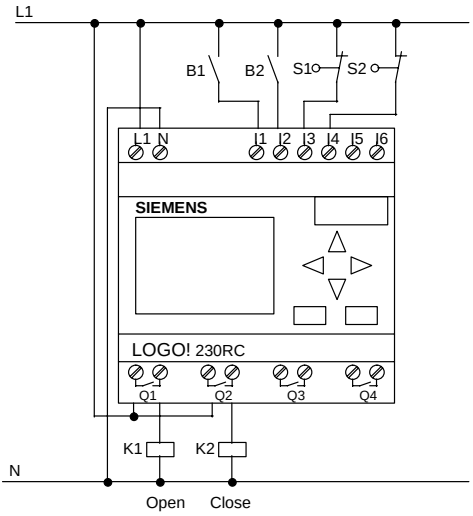
As soon as one of the motion detectors B1 or B2 registers somebody's presence, the door is opened by K3.

If the two motion detectors detect nothing for a minimum period, K4 enables the close operation.

7.2.3 Door control system with LOGO!

LOGO! allows you to considerably simplify the circuit. You need only connect the motion detectors, the limit switches and the master contactors to LOGO!.

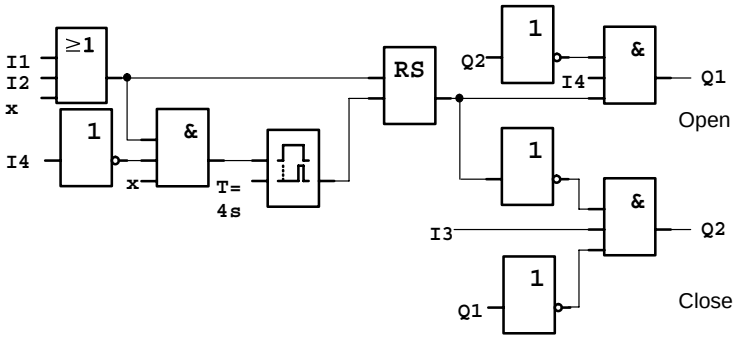
Wiring of the door control system with LOGO! 230RC



Components used

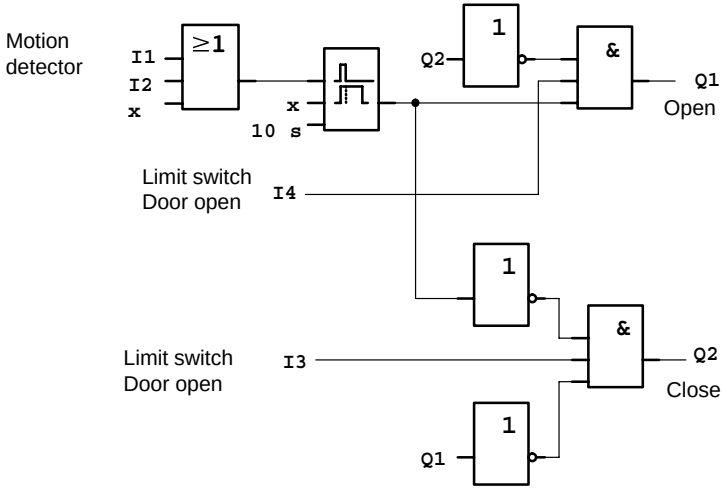
- K1 Master contactor *Open*
- K2 Master contactor *Close*
- S1 (NC contact) Limit switch *Closed*
- S2 (NC contact) Limit switch *Open*
- B1 (NO contact) Infrared motion detector *Outside*
- B2 (NO contact) Infrared motion detector *Inside*

Diagram of the door control system with LOGO!



This is what the functional block diagram that corresponds to the circuit diagram of the conventional solution looks like.

You can simplify this circuit if you make use of LOGO!'s functions. You can use the off-delay to replace the latching relay and on-delay. The following function block diagram illustrates this simplification:



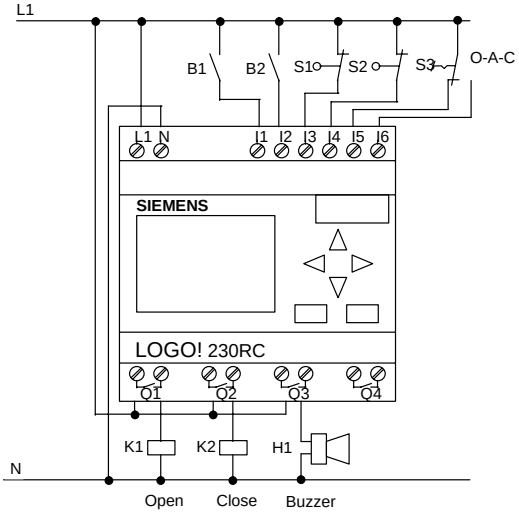
7.2.4 Special features and enhancement options

The functionality and user friendliness can be improved in the following ways, for example:

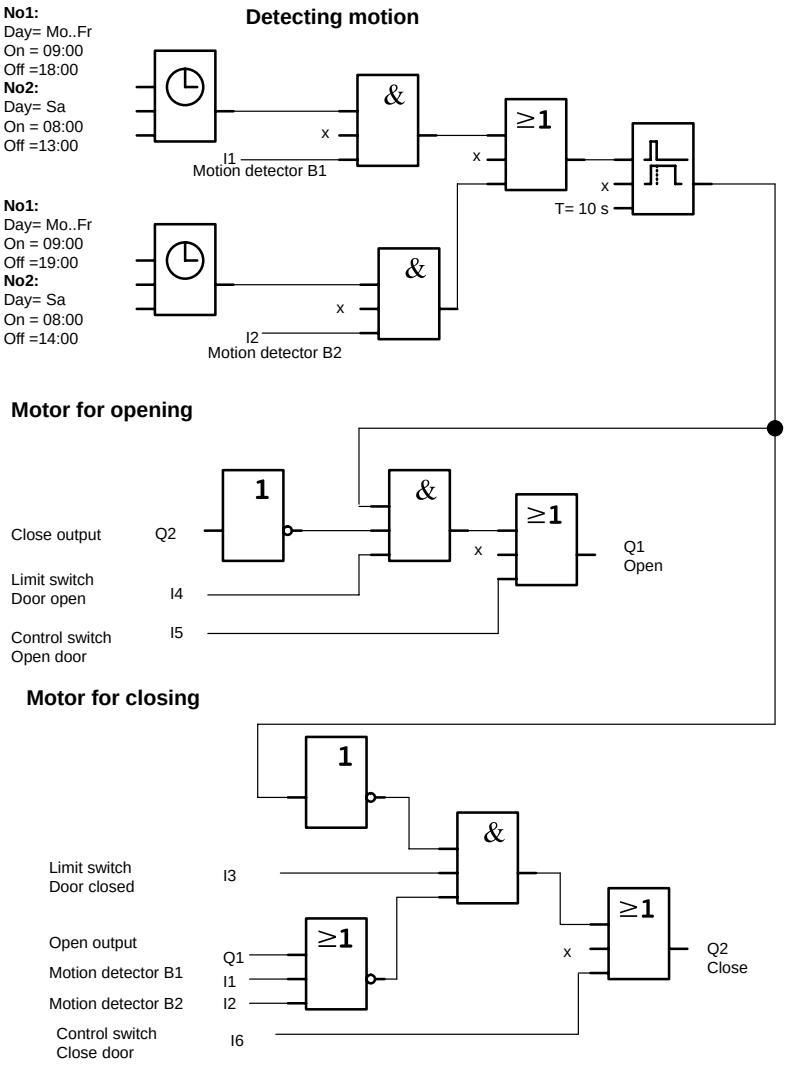
- You can connect an additional control switch: Open – Automatic – Closed (O-A-C)
- You can connect a buzzer to one of LOGO!'s outputs to indicate when the door is about to close.
- You can include time- and direction-dependent enabling of door opening (so that it only opens during shop opening hours and only from the inside to the outside after closing time, for example).

7.2.5 Enhanced LOGO! 230 RC solution

Wiring of the enhanced LOGO! solution



Functional block diagram of the enhanced LOGO! solution



Detecting motion

During business hours, motion detector B1 opens the door as soon as somebody wants to enter the shop from outside. Motion detector B2 opens the door if somebody wants to leave the shop.

After closing time, motion detector B2 continues to open the door for 1 hour so that customers can leave the shop.

Motor for opening

Output Q1 is switched on and opens the door when

- the control switch at I5 is operated (the door is to be constantly open), or
- the motion detectors indicate that somebody is approaching the door, and
- the door is not yet completely open (limit switch at I4).

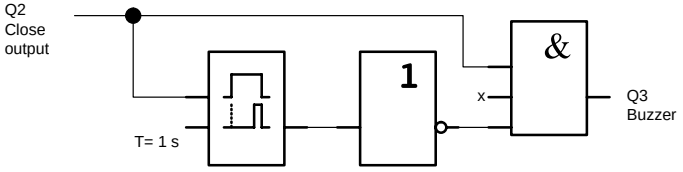
Motor for closing

Output Q2 is switched on and closes the door when

- the control switch at I6 is operated (the door is to be constantly closed), or
- the motion detectors indicate that there is nobody near the door, and
- the door is not yet fully closed (limit switch at I3).

Buzzer

You connect the buzzer to output Q3. The buzzer sounds for a short time (in this case 1 second) when the door is closed. In the block diagram, you enter the following circuit at Q3:



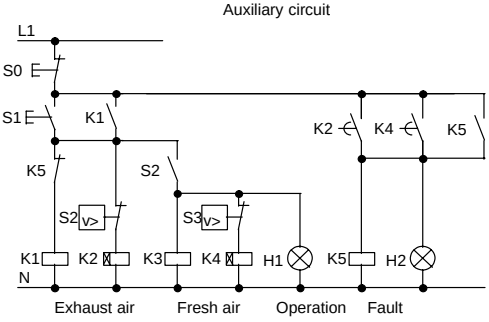
7.3 Ventilation system

7.3.1 Demands on a ventilation system

A ventilation system is used either to feed fresh air into a room or to remove used air from it. Consider the following example:

- The room contains an exhaust ventilator and a fresh air ventilator.
- Both ventilators are controlled by a control monitor.
- At no time must excess pressure develop in the room.
- The fresh air ventilator cannot be switched on unless the flow monitor indicates that the exhaust ventilator is functioning properly.
- A warning light comes on in the event of a ventilator failing.

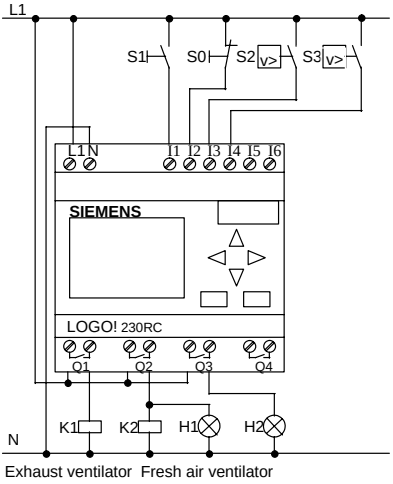
The circuit diagram for the previous solution is as follows:



The ventilators are controlled by flow monitors. If no air flow is detected after a short waiting time has elapsed, the system is switched off and a fault is reported. You acknowledge this by pressing the stop switch.

In addition to the flow monitors, the ventilation monitoring system requires an evaluation circuit with a number of switching devices. The evaluation circuit can be replaced by a single LOGO! module.

Wiring of the ventilation system with LOGO! 230RC

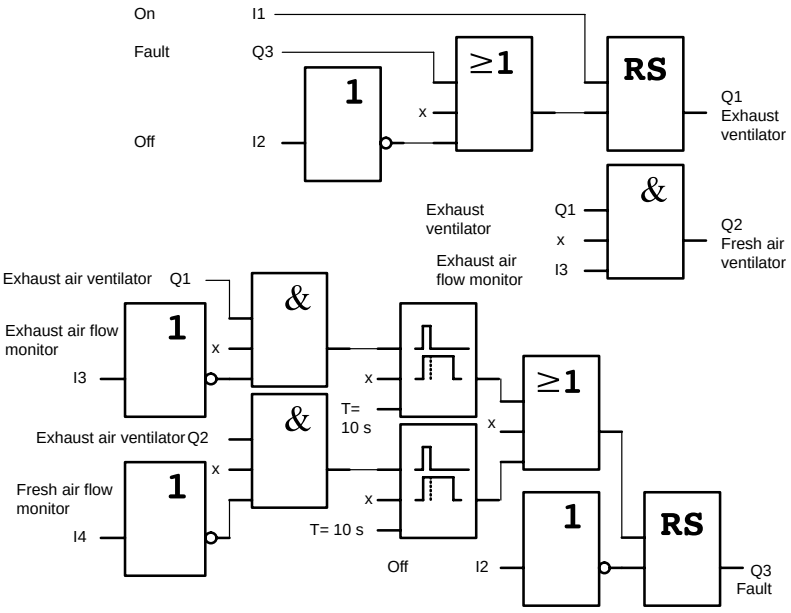


Components used

- K1 Master contactor
- K2 Master contactor
- S0 (NC contact) Stop switch
- S1 (NO contact) Start switch
- S2 (NO contact) Flow monitor
- S3 (NO contact) Flow monitor
- H1 Warning light
- H2 Warning light

Block diagram of the LOGO! solution

The block diagram of the ventilation control system with LOGO! is as follows:



7.3.2 Advantages of using LOGO!

When you use LOGO!, you do not need as many switching devices. Thus, you save on installation time and space in the switch box. You may even be able to use a smaller switch box.

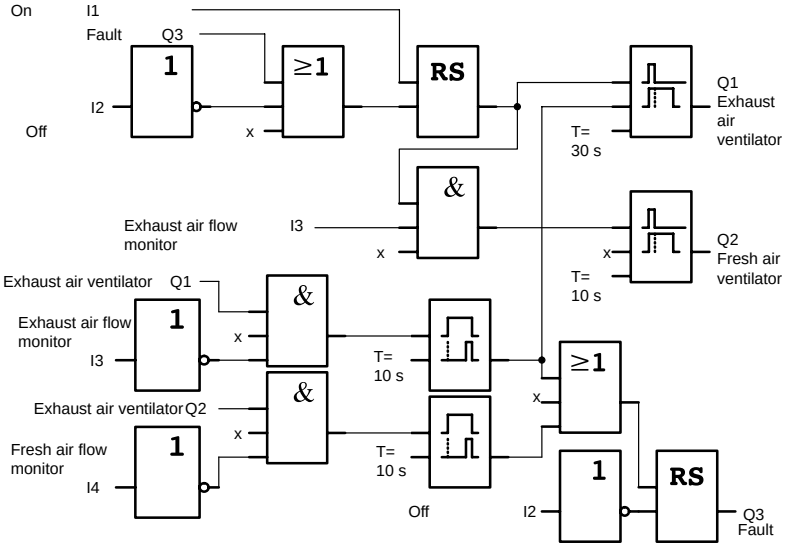
Additional options when using LOGO!

- The free output (Q4) can be used as a potential-free signalling contact in the event of a fault or a power failure.
- It is possible to stagger the switching-off of the ventilators.

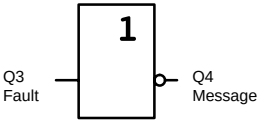
These functions can be implemented without additional switching devices.

Functional diagram of the enhanced LOGO! solution

The ventilators at Q1 and Q2 are switched off as shown in the following circuit:

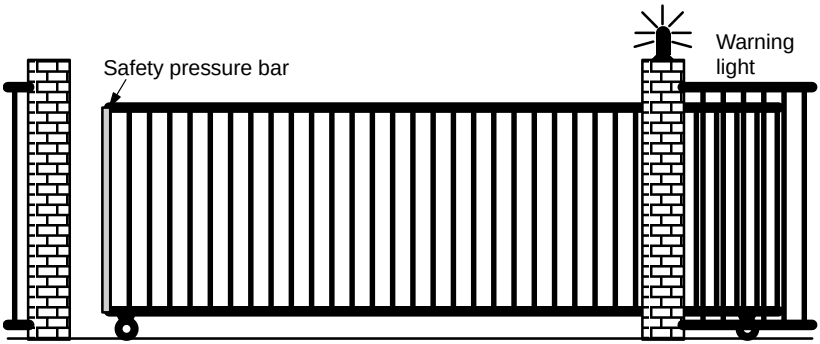


You can also generate a message via output Q4:



The contacts of output Q4 are always closed when the system is running. Relay Q4 does not release unless there is a power failure or a fault in the system. This contact can be used for teleindication, for example.

7.4 Industrial gate



There is often a gate at the entrance to a company's premises. This is only opened to let vehicles in and out.

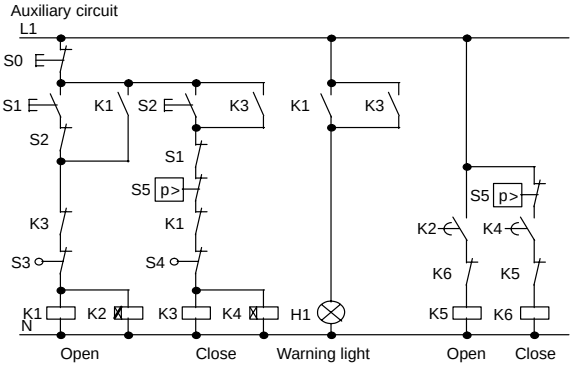
The gate is controlled by the gateman.

7.4.1 Demands on the door control system

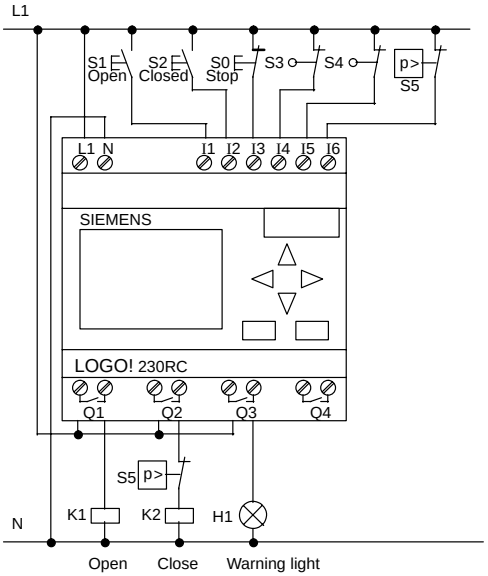
- The gate is opened, closed and monitored by the gateman, who operates it by means of a switch in the gatehouse.
- The gate is normally completely open or completely closed, but its movement can be interrupted at any time.
- A warning light starts flashing on and off 5 seconds before the gate begins to move and continues for as long as the gate is still moving.
- A safety pressure bar ensures that nobody gets injured and nothing gets caught or damaged when the gate closes.

7.4.2 Previous solution

Various kinds of control system are used to drive automatic gates. The circuit diagram shows *one* possible gate control circuit.



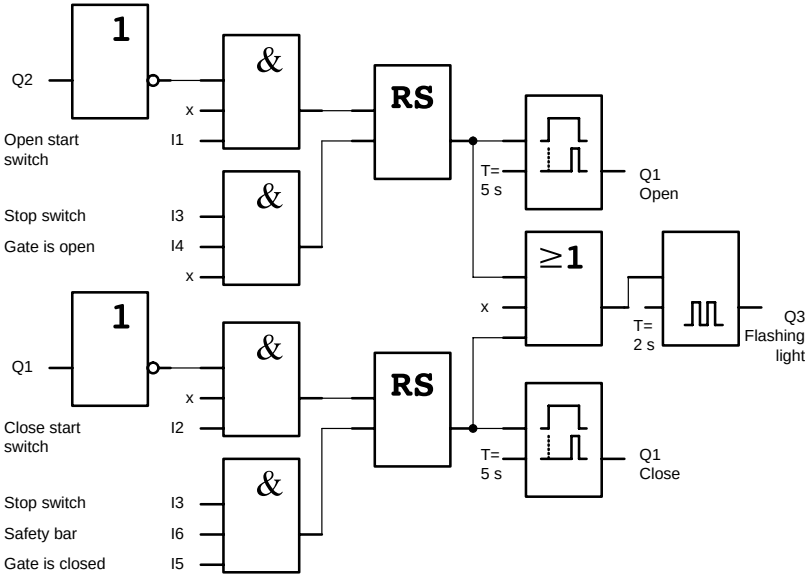
Wiring of the gate control system with LOGO! 230RC



Components used

- K1 Master contactor
- K2 Master contactor
- S0 (NC contact) Stop switch
- S1 (NO contact) Open switch
- S2 (NO contact) Close switch
- S3 (NC contact) Open position switch
- S4 (NC contact) Closed position switch
- S5 (NC contact) Safety pressure bar

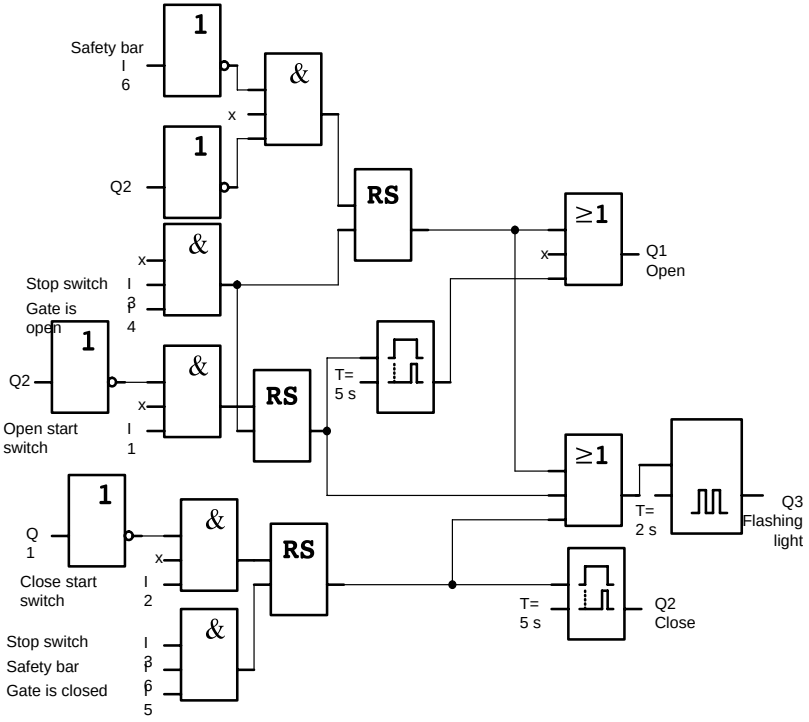
Functional diagram of the LOGO! solution



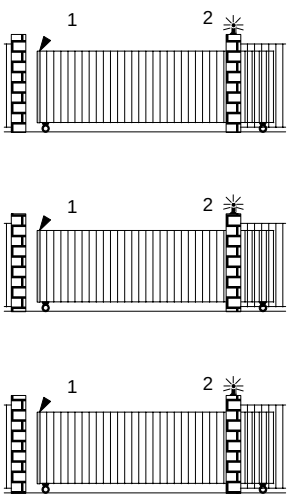
The open and close start switches start the movement of the gate, provided the gate is not currently moving in the opposite direction. The gate stops moving when the stop switch is pressed or when it reaches a limit switch. The gate is also prevented from closing by the safety bar.

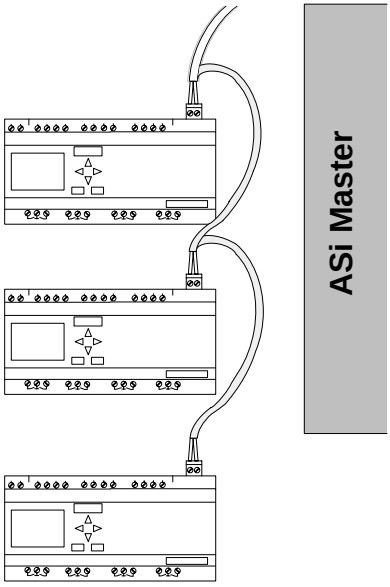
7.4.3 Enhanced LOGO! solution

In our enhanced solution, the gate will automatically open again when the safety bar operates.



7.5 Centralized activation and surveillance/monitoring of several industrial gates





1 Safety pressure bar
2 Flashing warning light

There are often a number of different entrances to a company's premises. Not all gates can always be surveilled and monitored directly by a member of staff. They must therefore be able to surveilled, monitored and operated by a gateman who sits in the a central control room.

In addition, it is obvious that each gate must also be able to be opened and closed immediately at the gate by personnel.

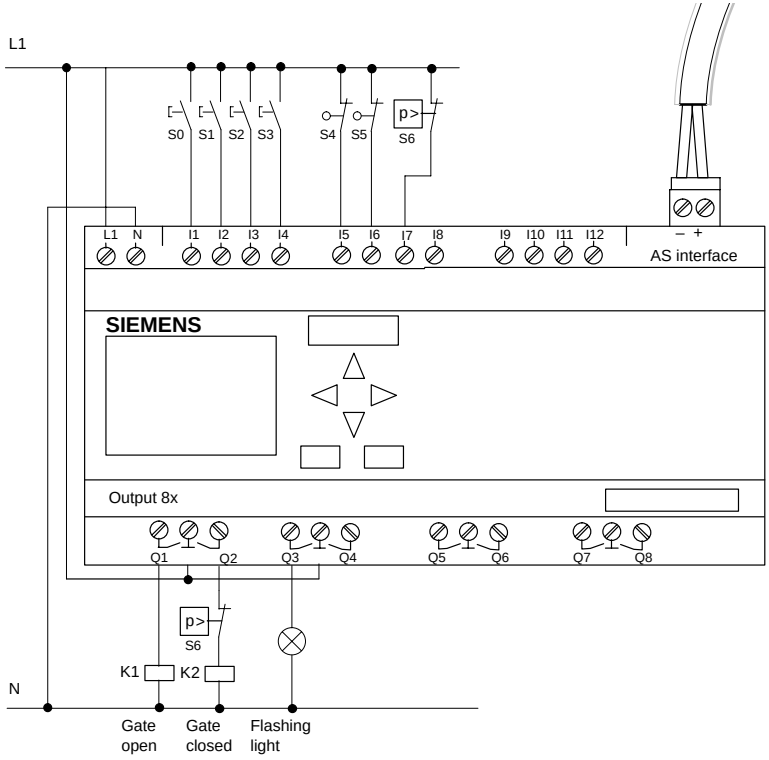
A LOGO!230RLB11 is used for each gate. The modules are linked to each other and an ASi master by means of the ASi bus.

In this chapter, we will describe the gate control system used for a gate. All the other gate control systems are identical.

7.5.1 Demands on the gate control system

- Each gate is opened and closed by means of a pull-cord switch. The gate is always opened and closed completely.
- In addition, each gate can be opened and closed by means of switches at the gate.
- The ASi bus connection enables the gateman to open and close the gate from the gatehouse. The state GATE OPEN or GATE CLOSED is indicated in the gatehouse.
- A flashing warning light starts flashing on and off 5 seconds before the gate begins to move and continues for as long as the gate is still moving.
- A safety pressure bar ensures that nobody gets injured and nothing gets caught or damaged when the gate closes.

Wiring of the gate control system with LOGO! 230RLB11



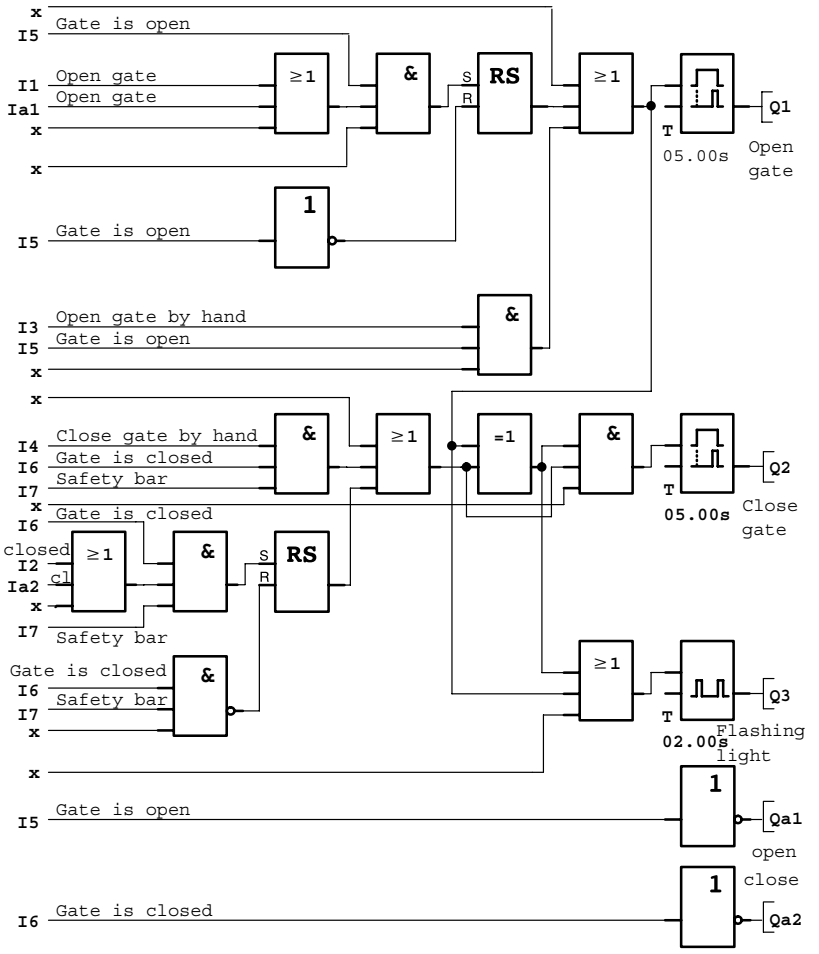
Components used

- K1 Master contactor, opening
- K2 Master contactor, closing
- S0 *(NO contact)* OPEN pull-cord switch
- S1 *(NO contact)* CLOSE pull-cord switch
- S2 *(NO contact)* OPEN switch
- S3 *(NO contact)* CLOSE switch
- S4 *(NC contact)* OPEN GATE position switch
- S5 *(NC contact)* CLOSE GATE position switch
- S6 *(NC contact)* Safety pressure bar

Higher-level control system

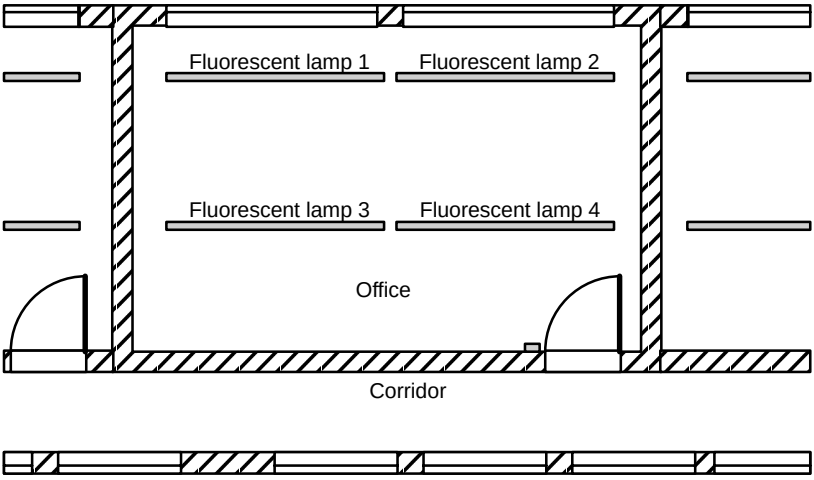
- Qa1 GATE OPEN position switch
- Qa2 GATE CLOSED position switch
- Ia1 External OPEN GATE switch
- Ia2 External CLOSE GATE switch

Functional diagram of the LOGO! solution



The OPEN GATE and CLOSE GATE start switches start movement of the gate provided the gate is not currently moving in the opposite direction. The gate stops moving when it reaches a limit switch. The gate is also prevented from closing by the safety bar.

7.6 Fluorescent lamps

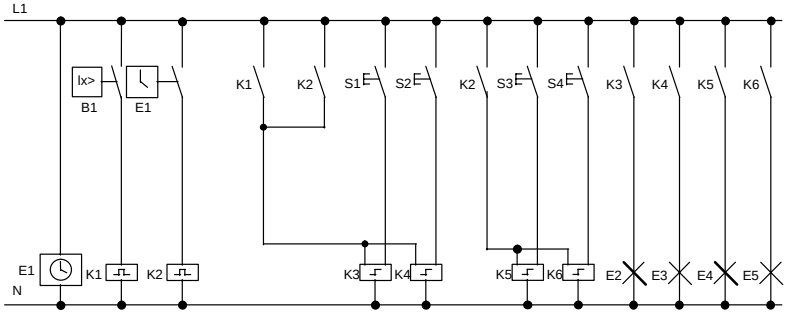


When lighting systems are planned in companies, the type and number of lamps used depends on the level of lighting required. For reasons of cost efficiency, fluorescent lamps arranged in rows of tubes are often used. They are subdivided into switching groups according to how the room is used.

7.6.1 Demands on the lighting system

- The lamps are switched on and off locally.
- If there is sufficient natural light, the lamps on the window side of the room are automatically switched off by means of a brightness-sensitive switch.
- The lights are switched off automatically at 8 o'clock in the evening.
- It must be possible at all times to switch the lights on and off locally.

7.6.2 Previous solution



The lights are operated by means of a pulse relay controlled by the switches at the door. Independently of this, they are reset by the time switch (clock) or by the brightness-sensitive switch via the *central off* input. The switching-off commands must be cut by impulse relays so that it is still possible to switch the lights on and off locally after they have been switched off centrally.

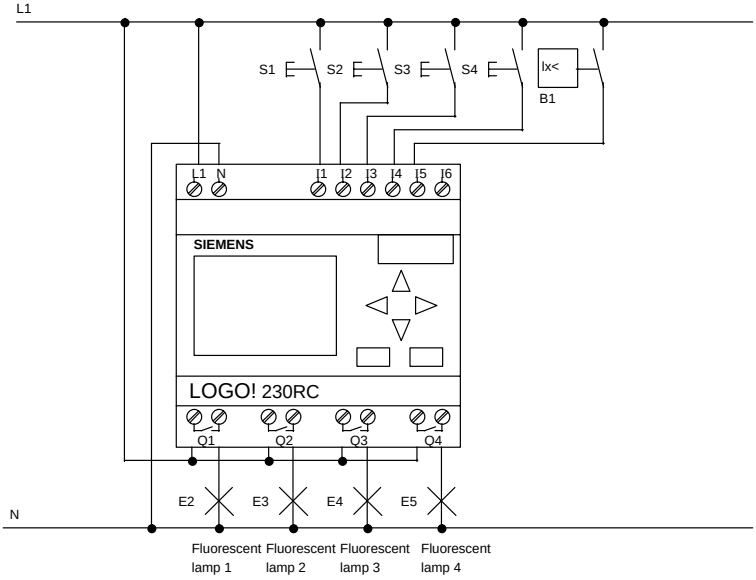
Components required:

- Switches S1 to S4
- Daylight control switch B1
- Time switch (clock) E1
- Impulse relays K1 and K2
- Remote-control switches with central off K3 to K6

Disadvantages of the previous solution

- To implement the required functions, a large amount of circuitry is required.
- The large number of mechanical components means that considerable wear and high maintenance costs can be expected.
- Functional changes are costly to implement.

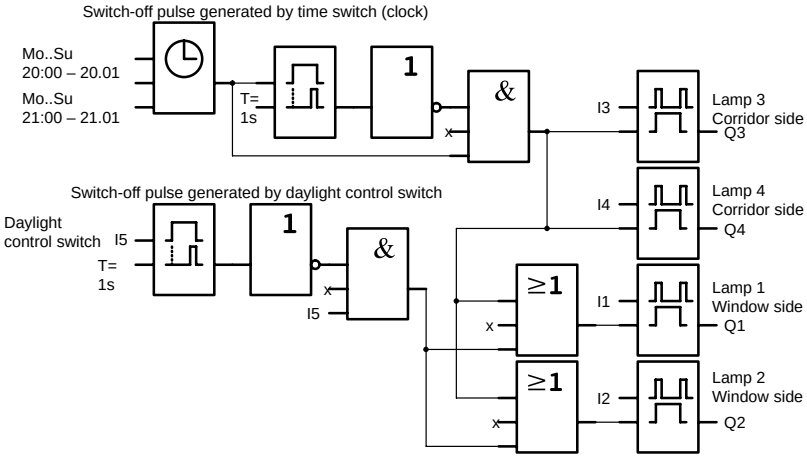
7.6.3 Fluorescent lamp control with LOGO! 230RC



Components used

- S1 to S4 (NO contact) Switches
- B1 (NO contact) Daylight control switch

Functional diagram of the LOGO! solution



Advantages of the LOGO! solution

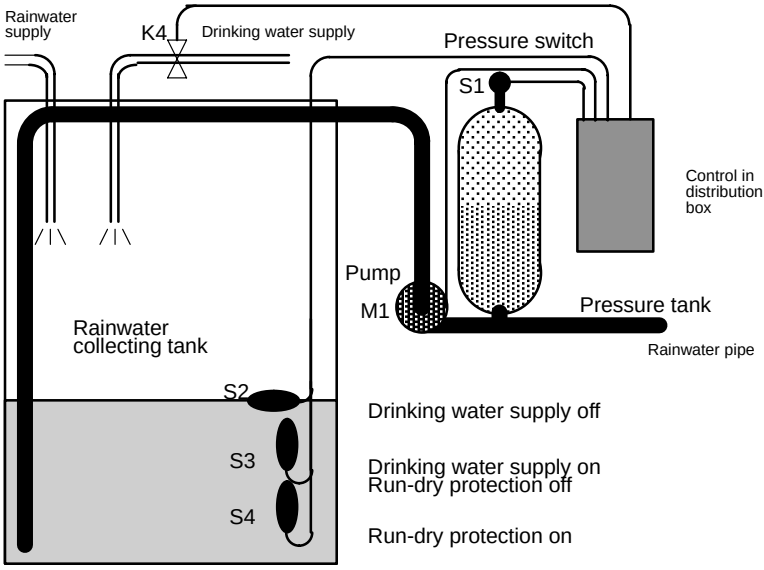
- You can connect the lamps to LOGO! directly provided the switching capacity of the outputs is not exceeded. In the case of greater capacities, you should use a power contactor.
- You connect the brightness-sensitive switch to one of LOGO!'s inputs directly.
- You do not need a time switch; this function is integrated in LOGO!.
- The fact that fewer switching devices are required means you can install a smaller sub-distribution unit and thus save space.
- Fewer devices are required.
- The lighting system can be easily modified.
- Additional switching times can be set as required (staggered switch-off pulses at the end of the day).
- The effect of the brightness-sensitive switch can easily be applied to all lamps or a changed group of lamps.

7.7 Rainwater pump

Rainwater is being used increasingly in homes in addition to drinking water. This saves money and is environment-friendly. You can use rainwater, for example, for:

- Washing clothes
- Watering the garden
- Watering house plants
- Washing the car
- Flushing the toilet

The following drawing illustrates how a system for using rainwater works:

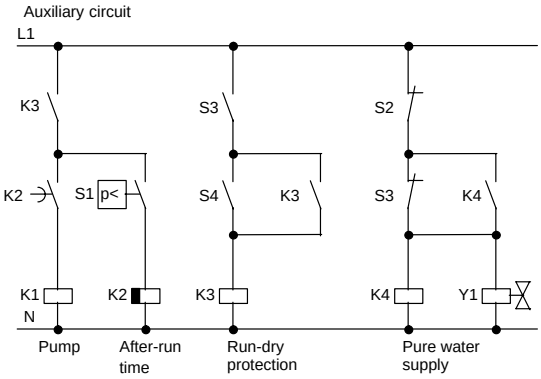


The rainwater is caught in a collecting tank, from which it is pumped into a pipe system. The rainwater can then be taken from this in the same way that drinking water can. If the tank should ever run dry, it can be supplied with drinking water.

7.7.1 Demands on the control system for a rainwater pump

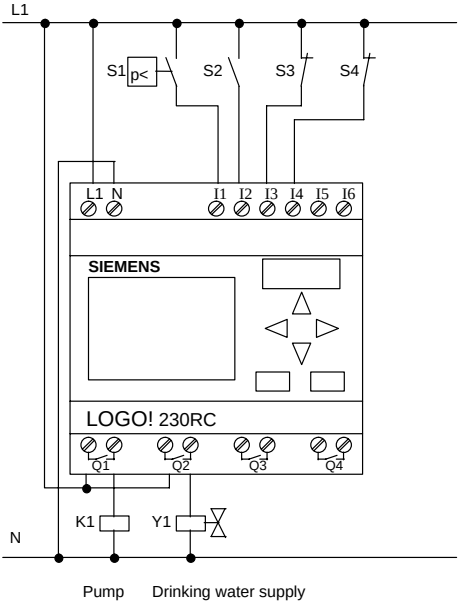
- The water must be available at all times. Whenever necessary, the controller must switch automatically to drinking water.
- When the switch to drinking water takes place, no rainwater must get into the drinking water system.
- If there is not enough water in the rainwater tank, the pump cannot be switched on (run-dry protection).

7.7.2 Previous solution



The pump and a solenoid valve are controlled by a pressure switch and 3 float switches in the rainwater tank. The pump must be switched on when the pressure goes below the minimum permitted. Once the operating pressure is reached, the pump is switched off again after an after-run time of a few seconds. The after-run time prevents the water pump from constantly being switched on and off if the water is drawn for any length of time.

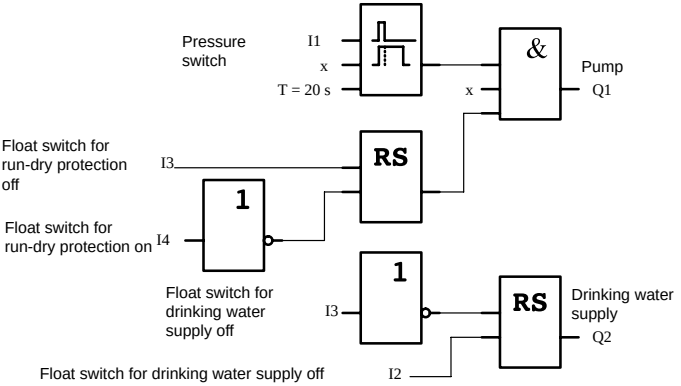
7.7.3 Rainwater pump with LOGO! 230RC



Apart from LOGO!, you need only the pressure switch and the float switches to control the pump. If you are using a three-phase motor, you need a master contactor. If the system has a single-phase motor, you need a contactor if the motor requires more current than output relay Q1 can switch. The consumption of a solenoid valve is so low that you can normally control it directly.

- K1 Master contactor
- Y1 Solenoid valve
- S1 (NO contact) Pressure switch
- S2 (NO contact) Float switch
- S3 (NC contact) Float switch
- S4 (NC contact) Float switch

Functional diagram of the LOGO! solution



7.7.4 Special features and enhancement options

In the functional diagram you can see how to wire the control system for the pump and the solenoid valve. Its structure corresponds to that of the circuit diagram. However, you can also integrate additional functions for specific applications that, with conventional technology, would require additional equipment:

- Enabling of the pump at specific times
- Indication of an imminent or existing water shortage
- Indication of malfunctioning

7.8 Centralized activation and monitoring of pumps

Areas in buildings that are endangered by being flooded by groundwater must be monitored constantly. In most cases, it is enough just to pump away the groundwater as of a certain level.

Each area endangered is equipped with 2 pumps which are controlled by a LOGO! 230RLB11. LOGO! receives all the information is requires from various sensors.

All logic modules are linked with each other and an ASi master by means of the ASi bus. All the areas are monitored in a central control room. Each individual pump can be operated separately via the ASi bus by means of switches.

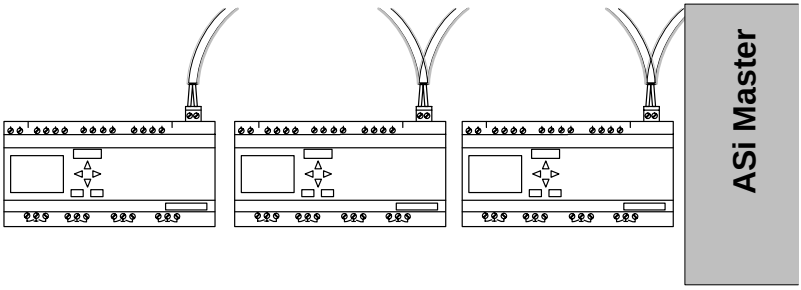
The following operating sequence must be programmed in each LOGO!:

When the maximum permitted water level is reached, pump 1 is switched on. If pump 1 fails, pump 2 is switched on automatically.

If both pumps fail, this ”emergency” state is signaled by a horn.

The program and the wiring of a LOGO! 230RLB11 can be found on the following pages.

You coordinate the individual slave assemblies (LOGO! 230RLB11) in your ASi master assembly.

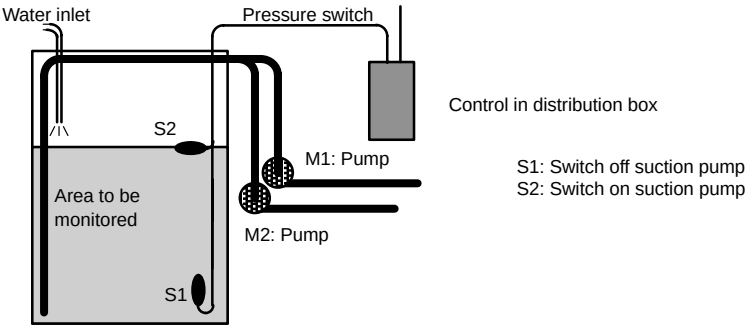


Note

This manual does not contain a description of how you can configure your ASi master and write the interconnecting framework program.

This information is contained in the description of your ASi master assembly.

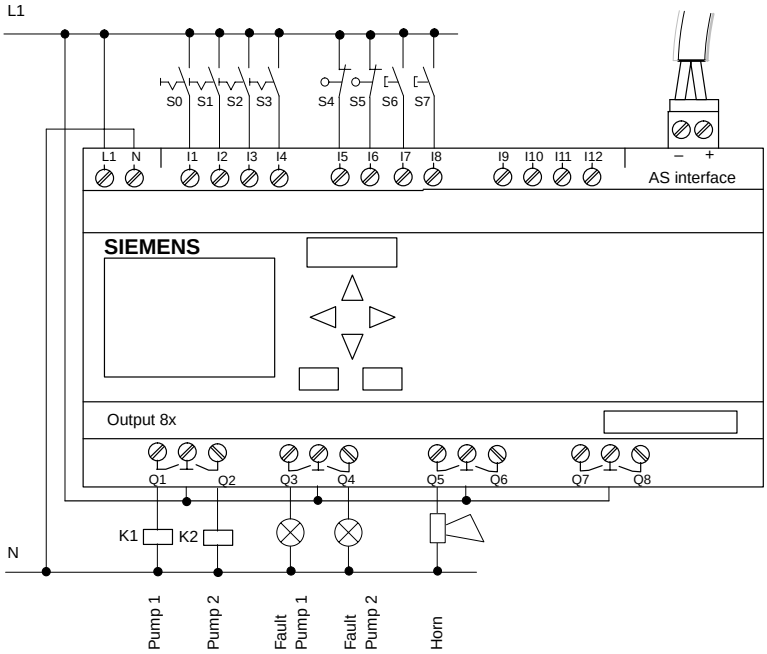
Area monitoring (principle)



7.8.1 Demands on the control system of a tank pump system

- When water level S2 is reached, pump 1 is switched on and continues pumping until the defined switch-off point S1 is reached.
- If pump 1 fails due to a fault while pumping, pump 2 is switched on automatically. The fault is signaled by an indicator light.
- If pump 2 also fails, total failure of the two pumps is signaled by a horn. The fault is also signaled by an indicator light.

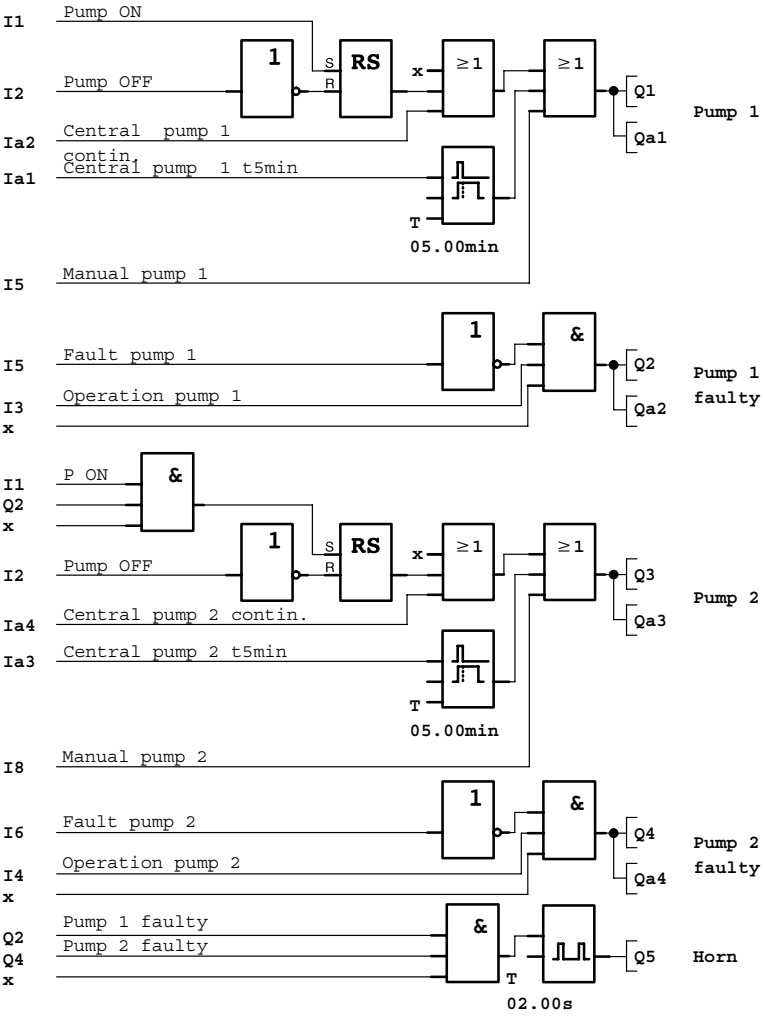
7.8.2 Pump control with LOGO! 230RLB11



In addition to LOGO!, you also require the following components to control the pumps:

- K1, K2 — one master contactor for switching each of the two three-phase motors of pumps 1 and 2
- H1, H2 — one fault indicator light for each pump (pumps 1 and 2)
- H3 — horn for signaling failure of both pumps
- S0 *(NO contact)* — level sensor for switching on the pump
- S1 *(NO contact)* — level sensor for switching off the pump
- S2, S3 *(NO contact)* — one sensor for each pump for signaling that pump 1 or pump 2 is operating
- S4, S5 *(NC contact)* — once sensor for each pump for monitoring pump 1 or pump 2 and signaling a fault
- S6, S7 *(NO contact)* — switches for operating the pumps manually

Functional diagram of the LOGO! solution



A single LOGO! assembly only enables you to perform a limited range of control tasks. If, however, you connect a number of LOGO!..LB11 in an ASi system by means of the AS interface, you have an extensive range of control options open to you.

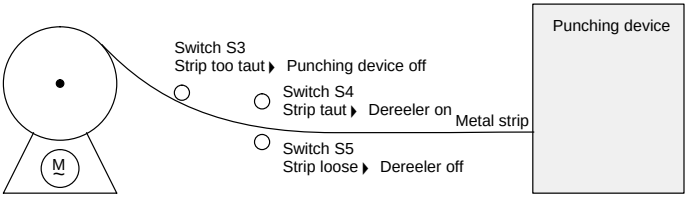
7.9 Dereeler

A metal strip (coil) is fed to a punching device over a dereeler.

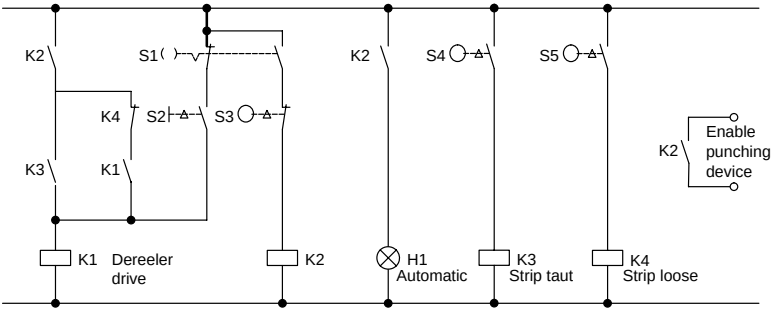
7.9.1 Demands on a dereeler

The following demands are placed on a dereeler:

- The metal strip must not sag.
- The metal fed to the punching device must not exceed a defined maximum tension.
- If the metal strip becomes too taut, the punching device must be switched off.



7.9.2 Previous solution

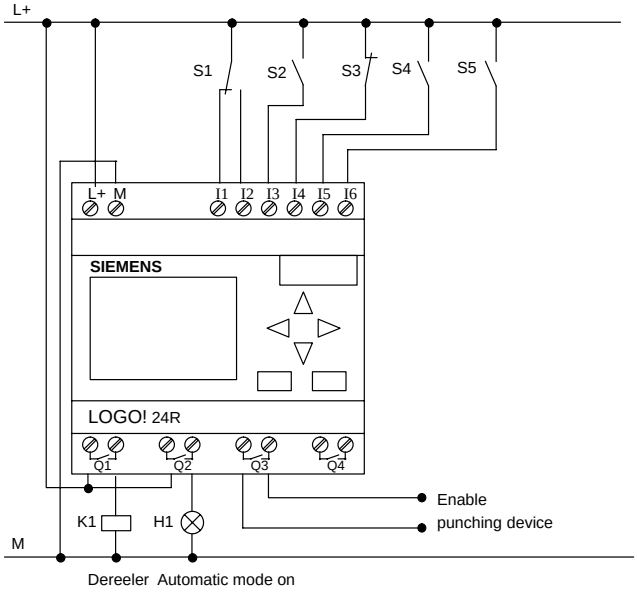


The key-operated switch S1 is the operating mode switch (manual/automatic) for the dereeler. You can use switch S2 to control the motor of the dereeler manually. Switches S4 and S5 monitor the tension of the strip and switch the motor of the dereeler on and off. Switch S3 switches the punching device off when the strip is too taut.

7.9.3 Dereeler with LOGO! 24R

If you use LOGO!, you can make the circuit much simpler. You only have to connect the switches, the indicator light and the main contactor to LOGO!.

Wiring the dereeler with LOGO! 24R

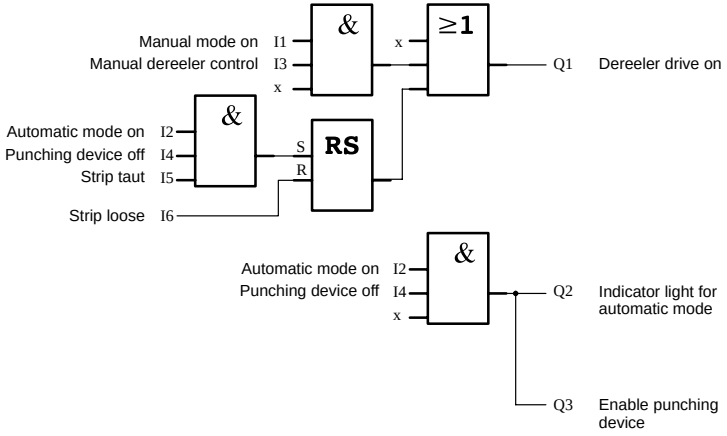


Components used

- S1 Operating mode switch: manual/automatic
- S2 (NO contact) Switch for manual dereeler control
- S3 (NC contact) Switch for switching the punching device off
- S4 (NO contact) Strip taut switch
- S5 (NO contact) Strip loose switch
- K1 Motor contactor
- H1 Indicator light for automatic mode

Functional diagram of the LOGO! solution

The block diagram for controlling the dereeler with LOGO! is as follows:



7.9.4 Advantages of the LOGO! solution

If you use LOGO!, you need fewer switching devices less wiring is involved. You also save on assembly time and space in the switch box. You may even be able to use a smaller switch box.

7.10 Additional application options

It is worth using LOGO! particularly when you:

- Can replace a number of auxiliary switching devices with the integrated functions of LOGO!.
- Want to save yourself wiring and installation work (because the wiring is done in LOGO!).
- Want to reduce the space required by the components in the control cabinet/distribution box. You may be able to use a smaller control cabinet/ distribution box.
- Can add or change functions subsequently without having to install an additional switching device or change the wiring.
- Have to provide your customers with additional functions for their domestic or building installation. Here are some examples:
  - Home security: You can program LOGO! to switch a lamp on regularly or open and close your shutters while you are on holiday.
  - Heating system: You can program LOGO! to run the circulation pump only when water or heat is really required.
  - Cooling systems: You can program LOGO! to thaw your cooling systems automatically on a regular basis to save energy costs.
  - You can illuminate aquaria and terraria automatically on a time-dependent basis.

You can also:

- Use commercially available switches and buttons, which makes it easy to integrate in the installation.
- Connect LOGO! directly to your domestic installation due to its integrated power supply.

Do you have any suggestions?

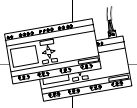
There are many more potential applications for LOGO!. If you know of one, why not write to us? We will collect all the suggestions we receive, and we intend to pass on as many as we can. So drop us a line – no matter how unusual or simple your LOGO! circuit is! We will be delighted to receive all your suggestions.

Write to:

Siemens AG
AUT 1MVM – LOGO!
Postfach 48 48
D-90327 Nuremberg

A Technical data

A.1 General technical data

Criterion	Complies with	Values
Dimensions W×H×D in mm Weight		72 x 90 x 55 with installation mechanism: 72 x 90 x 59 Approx. 190 g
Installation		on 35 mm DIN rail Width of 4 modules
Dimensions W×H×D in mm Weight		126 x 90 x 55 with installation mechanism: 126 x 90 x 59 Approx. 360 g
Installation		on 35 mm DIN rail Width of 7 modules
Ambient climatic conditions		
Ambient temperature Horizontal installation Vertical installation	Cold: IEC 68-2-1 Heat: IEC 68-2-2*	 0 to 55 °C 0 to 55 °C
Storage/transport		−40 °C to +70 °C
Relative humidity	IEC 68-2-30	From 5 to 95% no condensation
Atmospheric pressure		From 795 to 1080 hPa
Pollutants	IEC 68-2-42 IEC 68-2-43	SO_2 10 cm^3 /m^3, 4 days H_2S 1 cm^3 /m^3, 4 days
Ambient mechanical conditions		
Protection type		IP 20
Vibrations	IEC 68-2-6	10 to 57 Hz (constant amplitude 0.15 mm) 57 to 150 Hz (constant acceleration 2 g)

*IEC 68 includes VDE 0631

Criterion	Complies with	Values
Shock	IEC 68-2-27	18 shocks (half-sine 15g/11ms)
Drop	IEC 68-2-31	Drop height 50 mm
Free fall (packaged)	IEC 68-2-32	1 m
Electromagnetic compatibility (EMC)		
Electrostatic discharge	IEC 801-2 Severity 3	8 kV air discharge 6 kV contact discharge
Electromagnetic fields	IEC 801-3	Field strength 10 V/m
Interference suppression	EN 55011	Limit class B group 1 Limit class A for ASi operation
EMC emitted interference	EN 50081-2	
Immunity to interference	EN 50082-2	
Burst pulses	IEC 801-4 Severity 3	2 kV (supply and signal lines) B11 version: in accordance with *ASi Complete Specification V 2.0 dated November 27, 1995*
Energy carriers single pulse (surge) (applies only to LOGO! 230....)	IEC 801-5 Severity 2	0.5 kV (supply lines) symmetrical 1 kV (supply lines) asymmetrical
Information on IEC − /VDE − safety		
Measurement of clearance and creepage distance	IEC 664, IEC 1131, EN 50178 11/94 UL 508, CSA C22.2 No 142 Also VDE 0631 for LOGO! 230R/RC	Fulfilled
Insulation strength	IEC 1131	Fulfilled

A.2 Technical data: LOGO! 230....

	LOGO! 230R **LOGO! 230RC**	**LOGO! 230RL** **LOGO! 230RCL** **LOGO! 230RLB11**
Power supply		
Input voltage: rated value	115 V/120 V/230 V/ 240 V AC	115 V/ 230 V AC
Permissible range in acc. w.		
• VDE 0631:	85 V to 250 V AC	85 V to 250 V AC
• IEC 1131:	85 V to 265 V AC	85 V to 265 V AC
Permissible mains frequency:	47 to 63 Hz	47 to 63 Hz
Power consumption from		
• 115 V AC	Typically 40 mA	Typically 68 mA
• 120 V AC	Typically 40 mA	
• 230 V AC	Typically 26 mA	Typically 45 mA
• 240 V AC	Typically 26 mA	
Voltage failure bridging		
• 115 V AC	Typically 10 ms	Typically 10 ms
• 120 V AC	Typically 10 ms	
• 230 V AC	Typically 20 ms	Typically 20 ms
• 240 V AC	Typically 20 ms	
Power loss at		
• 115 V AC	Typically 2.5 W	
• 120 V AC	Typically 2.5 W	
• 230 V AC	Typically 3 W	Typically 4.5 W
• 240 V AC	Typically 3 W	
Clock buffering at 25 °C	Typically 80 h	Typically 80 h
Accuracy of the real-time clock (LOGO! 230RC; LOGO! 230RCL; LOGO! 230RCLB11)	Maximum ± 5 s/day	Maximum ± 5 s/day

	LOGO! 230R LOGO! 230RC	LOGO! 230RL LOGO! 230RCL LOGO! 230RLB11
Digital inputs		
Number	6	12
Electrical isolation	No	No
In groups of		4
Input voltage L1 at rated value of 115 V/230 V AC		
• Signal 0	0 V to 40 V AC	0 V to 40 V AC
• Signal 1	79 V to 265 V AC	79 V to 265 V AC
Input current at		
• signal 1	Typically 0.24 mA at 230 V AC	Typically 2.5 mA at 230 V AC
• signal 0		Typically 0.8 ... 1.2 mA
Delay time at		
• 0 after 1	Typically 50 ms	Typically 50 ms
• 1 after 0	Typically 50 ms	Typically 50 ms
Line length (unshielded)	100 m	100 m
Sensors: connection of		
• 2-wire Beros	No	No
• incandescent lamps	No	Yes
Digital outputs		
Number	4	8
Output type	Relay outputs	Relay outputs
Electrical isolation	Yes	Yes
In groups of	1	2
Activation of digital input	Yes	Yes
Continuous current I_{th} (per connector)	Maximum 8 A	Maximum 10 A

	LOGO! 230R LOGO! 230RC	LOGO! 230RL LOGO! 230RCL LOGO! 230RLB11
Incandescent lamp load (25,000 switching cycles) at		
230/240 V AC	1000 W	1000 W
115/120 V AC	500 W	500 W
Fluorescent tubes with electr. control gear (25,000 switching cycles)	10 × 58 W (at 230/240 V AC)	10 × 58 W (at 230/240 V AC)
Fluorescent tubes, conventionally compensated (25,000 switching cycles)	1 × 58 W (at 230/240 V AC)	1 × 58 W (at 230/240 V AC)
Fluorescent tubes, uncompensated (25,000 switching cycles)	10 × 58 W (at 230/240 V AC)	10 × 58 W (at 230/240 V AC)
Short-circuit proof cos 1	Power protection B16 600A	Power protection B16 600A
Short-circuit proof cos 0.5 to 0.7	Power protection B16 900A	Power protection B16 900A
Parallel switching of outputs to increase power	Not permitted	Not permitted
Protection of output relay (if desired)	Maximum 16 A, characteristic B16	Maximum 16 A, characteristic B16
Switching rate		
Mechanical	10 Hz	10 Hz
Ohmic load/lamp load	2 Hz	2 Hz
Inductive load	0.5 Hz	0.5 Hz
ASi slave Interfacing (LOGO! 230RLB11 only)		
ASi profile • I/O config • ID code		7.F 7_h F_h
Number of virtual digital inputs	–	4
Number of virtual digital outputs	–	4

	LOGO! 230R LOGO! 230RC	LOGO! 230RL LOGO! 230RCL LOGO! 230RLB11
Input voltage: rated value	–	24 V DC
Power supply	–	ASi power supply unit
Power consumption		Typically 30 mA
Electrical isolation	–	Yes
Polarity reversal protection	–	Yes

A.3 Technical data: LOGO! 24, LOGO! 24R, LOGO! 24RC

	LOGO! 24	LOGO! 24R; LOGO! 24RC
Power supply		
Input voltage: rated value	24 V DC	24 V DC
Permissible range	20.4 V to 28.8 V DC	20.4 V to 28.8 V DC
Power consumption from 24 V DC and per output max. 300 mA (4 * 0.3 A)	Typically 30 mA 1.2 A	Typically 62 mA
Voltage failure bridging		Typically 5 ms
Power loss at 24 V DC	Typically 0.8 W	Typically 1.5 W
Clock buffering at 25 °C (LOGO! 24RC)		Typically 80 h
Accuracy of the real-time clock (LOGO! 24RC)		Maximum ± 5 s/day
Digital inputs		
Number	6	6
Electrical isolation	No	No

	LOGO! 24	**LOGO! 24R; LOGO! 24RC**
Input voltage L+ Rated value • Signal 0 • Signal 1	 24 V DC <5.0 V DC >15.0 V DC	 24 V DC <5.0 V DC >15.0 V DC
Input current at • signal 1 • signal 0	 Typically 3 mA	 Typically 3 mA
Delay time at • 0 after 1 • 1 after 0	 Typically 50 ms Typically 50 ms	 Typically 50 ms Typically 50 ms
Line length (unshielded)	100 m	100 m
Digital outputs		
Number	4	4
Output type	Transistor, current-sourcing	Relay outputs
Electrical isolation	No	Yes
In groups of	4	1
Activation of digital input	Yes	⁄–
Output voltage	≙ Supply voltage	
Output current	Maximum 0.3 A	
Continuous current I_{th}		Maximum 8 A
Incandescent lamp load (25,000 switching cycles)		1000 W
Fluorescent tubes with electr. control gear (25,000 switching cycles)	–	10 × 58 W
Fluorescent tubes, conventionally compensated (25,000 switching cycles)	–	1 × 58 W

	LOGO! 24	LOGO! 24R; LOGO! 24RC
Fluorescent tubes, uncompensated (25,000 switching cycles)	–	10 × 58 W
Short-circuit proof and overload proof	Yes	
Short circuit current limiting	Approx. 1 A	
Derating	None throughout the entire temperature range	
Short-circuit proof cos 1	–	Power protection B16 600A
Short-circuit proof cos 0.5 to 0.7	–	Power protection B16 900A
Parallel switching of outputs to increase power	Not permissible	Not permissible
Protection of output relay (if desired)	–	Maximum 16 A, characteristic B16
Switching rate		
Mechanical	–	10 Hz
Electrical	10 Hz	–
Ohmic load/lamp load	10 Hz / 10 Hz	2 Hz
Inductive load	0.5 Hz	0.5 Hz

A.4 Technical data: LOGO! 24L, LOGO! 24RL, LOGO! 24LB11, LOGO! 24RLB11

	LOGO! 24L, LOGO! 24LB11	LOGO! 24RL, LOGO! 24RLB11
Power supply		
Input voltage: rated value	24 V DC	24 V DC

	LOGO! 24L, LOGO! 24LB11	LOGO! 24RL, LOGO! 24RLB11
Permissible range	20.4 V to 28.2 V DC	20.4 V to 28.2 V DC
Power consumption from 24 V DC at full load of outputs	Typically 2.44 A	Typically 120 mA
Voltage failure bridging		Typically 5 ms
Power loss at 24 V DC	Typically 1 W	Typically 2.9 W
Electrical isolation	No	No
Polarity reversal protection	Yes	Yes
Digital inputs		
Number	12	12
Electrical isolation	No	No
In groups of	12	12
Input voltage L+ Rated value • Signal 0 • Signal 1	 24 V DC <5.0 V DC >12.0 V DC	 24 V DC <5.0 V DC >12.0 V DC
Input current at • signal 1 • signal 0	 Typically 5 mA < 1.5 mA	 Typically 5 mA < 1.5 mA
Delay time at • 0 after 1 • 1 after 0	 Typically 50 ms Typically 50 ms	 Typically 50 ms Typically 50 ms
Line length (unshielded)	100 m	100 m

	LOGO! 24L, LOGO! 24LB11	LOGO! 24RL, LOGO! 24RLB11
Sensors: connection of • 2-wire Beros • incandescent lamps	 Yes –	 Yes –
Digital outputs		
Number	8	8
Output type	Transistor, current-sourcing	Relay outputs
Electrical isolation	No	Yes
In groups of	8	2
Activation of digital input	Yes	Yes
Output voltage	≙ Supply voltage	
Output current	Maximum 0.3 A	
Continuous current I_{th} (per connector)		Maximum 10 A
Incandescent lamp load (25,000 switching cycles)		1000 W
Fluorescent tubes with electr. control gear (25,000 switching cycles)	–	10 × 58W
Fluorescent tubes, conventionally compensated (25,000 switching cycles)	–	1 × 58W
Fluorescent tubes, uncompensated (25,000 switching cycles)	–	10 × 58W
Short-circuit proof and overload proof	Yes	
Short circuit currrent limiting	Approx. 1 A	
Derating	None throughout the entire temperature range	None throughout the entire temperature range

	LOGO! 24L, LOGO! 24LB11	**LOGO! 24RL, LOGO! 24RLB11**
Short-circuit proof cos 1	–	Power protection B16 600A
Short-circuit proof cos 0.5 to 0.7	–	Power protection B16 900A
Parallel switching of outputs to increase power	Not permissible	Not permissible
Protection of output relay (if desired)	–	Maximum 16 A, characteristic B16
Switching rate		
Mechanical	–	10 Hz
Electrical	10 Hz	–
Ohmic load/lamp load	10 Hz / 10 Hz	2 Hz
Inductive load	0.5 Hz	0.5 Hz
ASi slave Interfacing (LOGO! 24RLB11 only)		
ASi profile • I/O config • ID code	7.F 7_h F_h	7.F 7_h F_h
Number of virtual digital inputs	4	4
Number of virtual digital outputs	4	4
Input voltage: rated value	24 V DC	24 V DC
Power supply	ASi power supply unit	ASi power supply unit
Power consumption	Typically 30 mA	Typically 30 mA
Electrical isolation	Yes	Yes
Polarity reversal protection	Yes	Yes

Switching capacity and service life of therelay outputs

Ohmic load

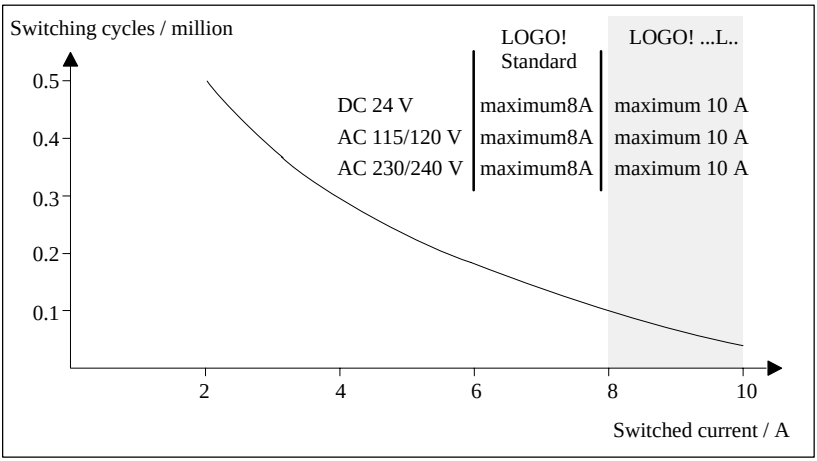


Figure A Switching capacity and service life of the contact at ohmic load (heating)

Inductive load

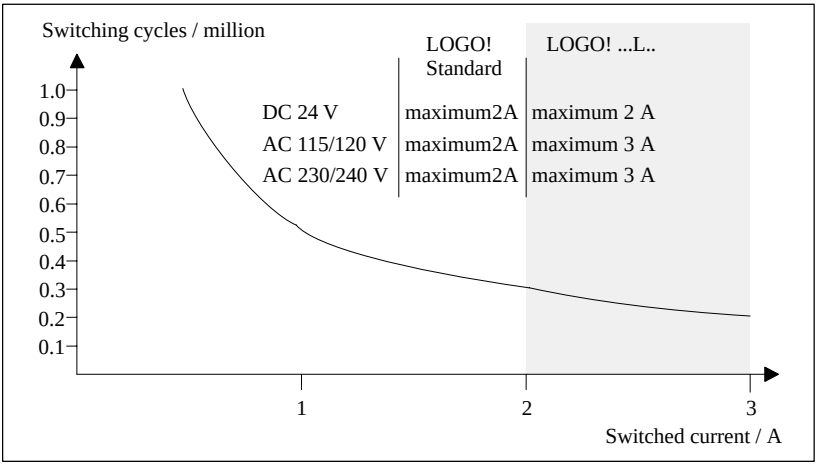


Figure B Switching capacity and service life of the contacts at highly inductive load in accordance with IEC 947-5-1 DC13/AC15 (contactors, solenoid coils, motors)

A.5 LOGO! Power 1.3 and LOGO! Power 2.5

LOGO! Power 1.3 and LOGO!Power 2.5 are switched-mode primary power supply units for the LOGO! ...L... versions.

	LOGO! Power 1.3	**LOGO! Power 2.5**
Input data		
Input voltage: rated value	120/230 V	
Permissible range	85 V ... 264 V (Derating at <93 V)	
Input voltage frequency	47 .. 63 Hz	
Voltage failure bridging	20 (10) ms at 187 (110) V AC	
Making current (25˚C)	< 15 A	
Output data		
Output voltage: rated value	24 V DC	
Output voltage: overall tolerance	+/- 8%	
Output voltage: residual ripple	< 250 mVss	
Output current: rated value	1.3 A	2.5 A
Overcurrent limiting	1.35 A	2.8 A
Outputs idling- and short-circuit proof	Yes	
Efficiency	> 80 %	
Electromagnetic compatibility		
Interference suppression (emitted)	EN 50081-1, EN 55022 Class B	
Immunity to interference	EN 50082-2	
Safety		
Electrical isolation, primary/secondary	Yes, SELV (in acc. w. EN 60950 / VDE 0805)	
Safety class	II (in acc. w. IEC 536 / VDE 0106 T1)	
Protection type	IP 20 (in acc. w. EN 60529 / VDE 470 T1)	
Planned certification	CE, UL/cUL, FM	

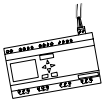
	LOGO! Power 1.3	LOGO! Power 2.5
General details		
Ambient temperature range	0 ... +55˚C, natural convection	
Storage and transport temperature	-40˚C ... +70˚C	
Connections on input	one connector (1x2.5mm^2 or 2x 1.5 mm^2) each for L1 and N	
Connections on output	two connectors (1x2.5mm^2 or 2x 1.5 mm^2) each for L+ and M	
Installation	on 35 mm DIN rail, snap-on	
Dimensions in mm (WxHxD)	72x80x55	126x90x55
Weight	Approx. 0.3 kg	Approx. 0.6 kg

A.6 LOGO! Contact 24 and LOGO! Contact 230

LOGO! Contact 24 and LOGO! Contact 230 are switching modules for direct switching of ohmic loads of up to 20 A and motors of up to 4 kW (without noise emission, hum-free).

	LOGO! Contact 24	LOGO! Contact 230
Operating voltage	24 V DC	230 V AC; 50/60 Hz
Switching capacity		
Utilization category AC-1 Switching of ohmic load at 55˚C Operating current	85 V ... 264 V (derating at <93 V)	
Operating current at 400 V	20 A	
Output of three-phase loads at 400 V	13 kW	
Utilization category AC-2, AC-3 Motors with slipring or squirrel-cage rotor	85 V ... 264 V (derating at <93 V)	
Operating current at 400 V	8.4 A	

	LOGO! Contact 24	**LOGO! Contact 230**
Output of three-phase loads at 400 V	4 kW	
Short-circuit protection: assignement type Type 1	25 A	
Short-circuit protection: assignment type Type 1	10 A	
Connecting leads	finely stranded with connector sleeves single-core 2x(0.75 to 2.5) mm^2 2x(1 to 2.5) mm^2 1x4 mm^2	
Dimensions (WxHxD)	36x72x55	
Ambient temperature	-25˚C to +55˚C	
Storage temperature	-50˚C to +80˚C	



B LOGO! ...LB11: Active-passive switchover

All LOGO! ...LB11 versions are factory-set to address 0.

When the master is assigning addresses, only one active slave may be set to address 0 on the ASi bus at any one time. All other slaves with the address 0 must be passive, that is, unknown on the bus.

Caution

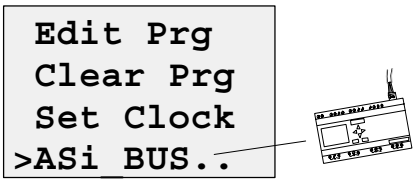
The ASi address can be changed 10 times for all LOGO! ...LB11 versions.

There is no guarantee for more changes than this.

To switch LOGO! ...LB11 to passive, we have built a specific menu item into the programming menu.

Switching LOGO! ...LB11 between active and passive

1. Switch LOGO! ...LB11 to programming mode (using the 3-finger grip) and press **OK** to go directly into the programming menu.
2. Press the ▼ key 3 times.
   The cursor (>) is now positioned at the start of the ASi_Bus.. line.



3. Press **OK**. The following display appears:

4. Switch LOGO! ...LB11 to passive by pressing the ▼ key and then **OK**. The new state is then displayed:

5. As soon as the master has detected an active slave and assigned it an address, you can switch another slave back from **passive** to **active**.

Note

You can exit the menu for switching back and forth between active and passive only if LOGO! is switched **active**.

C Determining the amount of memory required

The maximum number of function blocks in a program is 30. This applies to the basic functions. If you use special functions in your applications, the maximum number of function blocks possible may under certain circumstances be reduced. Please pay attention to the examples given in this chapter.

The function blocks of the special functions in your program require special memory in LOGO!. There are four different memory areas for this in LOGO!. The amount of memory required in the different memory areas varies depending on which function is used.

Memory area	Meaning
△	Area in which your target values are stored (e.g. limit values of the counter) LOGO! has space for **27** units in this memory area.
□	Area in which the current actual values are stored (e.g. current count) LOGO! has space for **24** units in this memory area.
○	Area used by the time functions (e.g. off-delay) LOGO! has space for **10** units in this memory area.
 RE	Area in which current actual values to be stored as **remanent** values are stored (e.g. the count of an operating hours counter). Remanent data storage is only possible in LOGO!...L.. versions LOGO! has space for **7** units in this memory area.

Memory required for thefunctions

The following table provides you with an overview of how much memory each block occupies in each memory area:

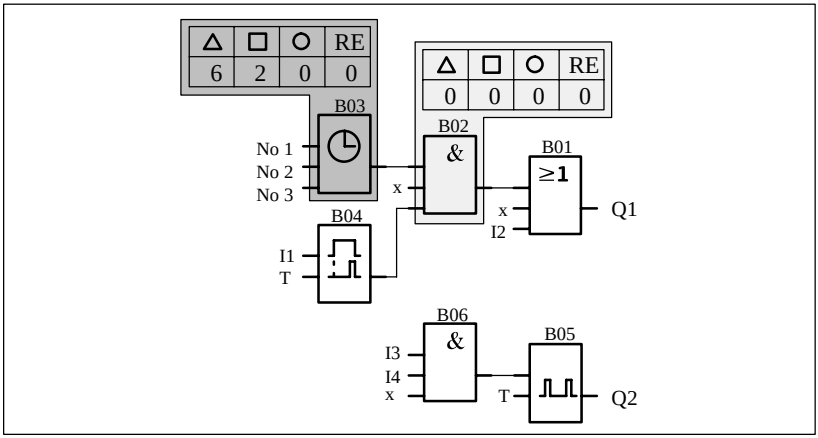
Function	Memory area			
	△	□	○	RE
Basic functions	0	0	0	0
On-delay	1	1	1	0
Off-delay	2	1	1	0
Wiping relay	1	1	1	0
Pulse relay [1]	0	1	0	0
Pulse relay[2] 	0	0	0	1
Clock (time switch)	6	2	0	0
Latching relay [1]	0	1	0	0
Latching relay[2] 	0	0	0	1
Clock pulse generatore	1	1	1	0
Retentive on-delay	2	1	1	0
Counter [1]	2	2	0	0
Counter [2] 	2	0	0	2
Operating hours counter 	2	0	0	4
Threshold switch 	3	3	1	0

[1] Memory occupancy in LOGO!...L... versions without parameterized remanence
[2] Memory occupancy in LOGO!...L... versions with parameterized remanence


The marked functions offer additional functionality which is only available in LOGO!...L.. versions

Always take into account all individual areas of the memory when determining the amount of memory required by a circuit.

Example:



The example program contains:

Block No.	Function / Memory area	△	□	○	RE
B01	OR	0	0	0	0
B02	AND	0	0	0	0
B03	Clock	6	2	0	0
B04	On-delay	1	1	1	0
B05	Clock pulse generator	1	1	1	0
B06	AND	0	0	0	0
	Memory occupied by the program	8	4	2	0
	Memory limits in LOGO!	27	24	10	7
	Memory still free in LOGO!	19	20	8	7

Of the maximum of 30 blocks available in LOGO!, you have used 6 blocks for your program.

The program therefore fits in LOGO!.

If you cannot enter any more blocks when entering a program, this means that a memory area is full. LOGO! offers you only those blocks for which it still has enough space. If there is not enough space in LOGO! for any of the blocks in the list, you can no longer select the list.

When a memory area is full, optimize your circuit or use a second LOGO! module.

Abbreviations

B01	Block number B01
BN	Block number
Cnt	Count (input for counter)
Co	Connector
Dir	Direction (for counter: up or down)
En	Enable (switching on the clock pulse generator)
BF	Basic function
No	Nocke (cam for clock)
Par	Parameter for counter
Par	Parameter list for various functions 
R	Reset
S	Set (setting the latching relay)
SF	Special function
T	Time (parameter)
T_a	Current value of a time (the current value of a time is displayed in the parameterization mode)
Trg	Trigger (parameter)