SIEMENS

Stefan Kruse

LOGO! 8

A Practical Introduction, with Circuit Solutions and Example Programs

Stefan Kruse
LOGO! 8

LOGO! 8

A Practical Introduction,
with Circuit Solutions and Example Programs

by Stefan Kruse

with Armin Ruch
and Joachim Zimmermann

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie;
detailed bibliographic data are available on the Internet at http://dnb.d-nb.de.

www.publicis-books.de

Print ISBN 978-3-89578-445-3
ePDF ISBN 978-3-89578-926-7

Editor: Siemens Aktiengesellschaft, Berlin and Munich
Publisher: Publicis Pixelpark, Erlangen

Printed in Germany

Preface

Thanks to the LOGO! universal control module, there has been a reliable, quick, and flexible option for implementing simple automation tasks for many years now. The easy-to-operate hardware unit integrates the controller, an operating and display unit, the power supply, and the interfaces for program modules, networks, and PC connections. The intuitively designed software contains the preconfigured basic functions typically used in practice, e.g. for delayed switching ON/OFF, counting functions, protective relay, comparator levels or arithmetic operations, which can be connected via analog or digital inputs and outputs. Thanks to the modular layout of the hardware, the easy-to-program software, and numerous preset basic functions, the system is not only suited for use in industry, but also for private use and hobbies.

Due to the simple and convenient operability of the programming, the easy integration of the LOGO! into an electrical circuit, and minimal wiring effort, even relatively unskilled amateur technicians can achieve professional results in their small automation projects.

This book explains the basics of information technology and should allow entry-level technicians to take their first step into automation technology using LOGO!. In addition, we present a wide variety of interesting and customizable projects to the passionate amateur technician.

All of the implemented projects have a modular layout. This means that competence gained from everyday examples can easily be transferred to individual projects step-by-step. Even if not all of the ideas that are presented have to be put into practice, many challenges are addressed and a wide variety of solution variants are offered. The circuits that are presented can largely also be implemented using previous LOGO! modules. Some projects, however, adopt the new capabilities of LOGO! 8, such as the SMS function, enhanced functions of the Look & Feel text display, or the integrated web server. In addition to 15 interesting everyday projects, a soldering course, parts lists, and sources of supply are also provided for any required hardware.

We hope you enjoy and wish you much success with interesting Siemens LOGO! projects.

April 2015

Stefan Kruse,
Armin Ruch and
Joachim Zimmermann

Contents

1 Modern information technology

The developments in information technology in the last few decades have changed the lives of people more than any other technological field. Modern information technology encompasses many aspects. It includes the following fields:

- Electrical engineering (research, development, and production of artifacts and methods used in connection with electrical energy)
- Computer engineering (systematic processing of information),
- Technical computer engineering (planning, design, implementation, operation, and analysis of computer engineering systems),
- Communications engineering (recording, transmitting, processing, and saving of information),
- Automation engineering (connecting the overlaps of measurement, control and closed-loop control engineering) and
- the human-computer interaction that has appeared in recent years (planning and design of interactive systems and the human-machine interface).

Fig. 1.1 shows the connection of the disciplines involved in information technology.

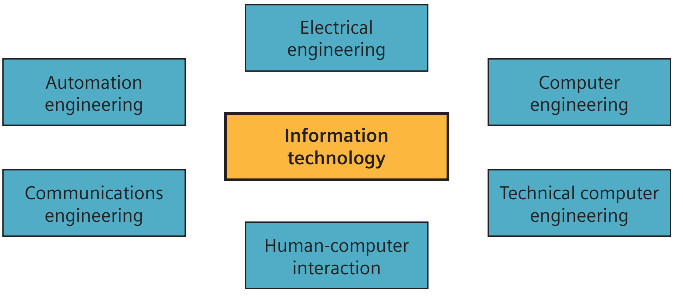


Fig. 1.1 Disciplines involved in information technology

All of the disciplines at the heart of information technology are based on the same operating principle: Most information technology processes can be described in terms of the basic systematic principle of input, processing and output (IPO), both chronologically and spatially. The connection to a classification of the various types of work objects by material, energy and information allows the clear assignment of the information technology elements and a description of the interaction between the system‘s components and its environment (see Table 1.1).

Table 1.1 Combination of the IPO principle with the systematics of the basic technological procedures

Work objects	"IPO" principle Input (I)	 Processing (P)	 Output (O)
Material	e.g. water	e.g. warming	e.g. water vapor
Energy	e.g. solar energy	e.g. heat exchanger	e.g. heat energy
Information	e.g. actual temperature value	e.g. triggering a switching action	e.g. setpoint temperature value

1.1 Fundamentals of electrical engineering

For the planning of an automated circuit, this "IPO" operating principle must be transferred to the project to be processed. In the planned open-loop and closed-loop control system, information is transferred and processed by means of electrical voltage. Process states are detected by sensors, which convert the detected data into information which can be further processed. The acquired information is converted in such a way that it can be processed by the controller as an input variable. After the information is processed, it is converted into information that can be read by the actuator, which converts it into process energy. The flow diagram in Fig. 1.2 illustrates this in an example.

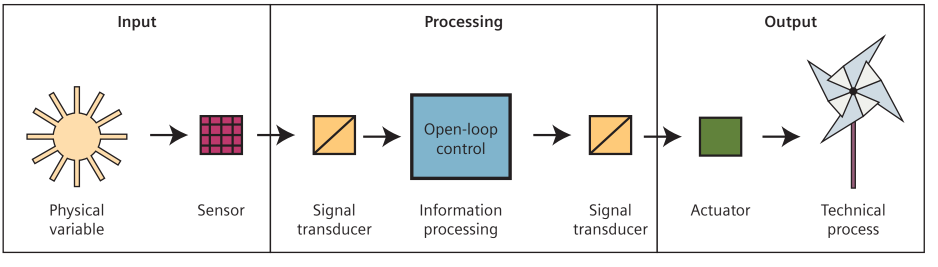


Fig. 1.2 Flow diagram and components of a control system

1.1.1 Sensors

Sensors are used to record measurement signals or physical environmental values. They detect information about properties, states, and processes. Non-electrical measurement variables, such as temperature, pressure, acoustic and optical variables, are converted into electrically analyzable variables. These electrical variables can be processed in a control unit. Thus, via sensors, a system can perceive specific information, properties, states, or processes from its environment.

Just like the sense organs in human beings, sensors can detect different physical and chemical environmental conditions. With the aid of infrared sensors, ultrasonic sensors, or simple touch sensors, a robot can detect obstacles or drop-offs, for example, in order to be able to move about its environment. In addition there are sensors for measuring temperature, air humidity, humidity, radiation, pressure, vibration, gas, motion, light, and sound. Modern robots also have GPS modules which allow them to determine and share accurate information about their location.

All of the sensors are basically subject to the following criteria:

- A sensor should be designed and operable as easy as possible. This concerns the properties of the enclosure, the connection options, and the ability to calibrate the sensor.
- A sensor should be as robust as possible. Sensors are generally subjected to physical stress such as impacts, vibration, or environmental factors such as sunlight or inclement weather. This means that the actual sensor should be protected by an enclosure or some technical measure.
- In sensor design, a distinction is made between an active and passive module. Some sensors need an active power source (e.g. an infrared sensor with an infrared LED). Then, increased consumption of power must be taken into consideration. Passive sensors receive their signals from the environment (e.g. a pushbutton is pushed). The majority of ambient variables such as temperature, humidity, light, vibration, or force are recorded with the aid of passive components.
- The measured values are represented by analog or digital signals. Analog signals are often analyzed based on the principle of the voltage divider: A base voltage is entered in the sensor and a partial voltage is output. The resulting difference is recorded and analyzed in the program using a value table. A digital signal or a pulse is analyzed as a clearly defined input and is further processed.

1.1.2 Actuators

Whereas sensors convert a physical variable into an electric current, as described, actuators do exactly the opposite, as the technical counterpart to sensors. They convert electrical signals from the control unit into a different form of energy such as physical movement or into other physical variables[1] such as light intensity, temperature, or sound. Electrical energy is converted to a different form of energy, e.g. kinetic energy (for movement). In this way, the system can influence its environment.

Actuators can be systematized in the following areas corresponding to their conversion method:

- **Electromagnetic actuators:** Electrical energy is converted to mechanical energy. This can take place in the form of kinetic energy or rotational energy via a switch, slider, or potentiometer. Speakers which generate sound pressure via a diaphragm or motors with rotors that are moved by an electromagnetic field also belong in this category.
- **Piezoelectric actuators:** The converse of the Piezo effect is used here. This is based on the fact that when a voltage is applied to a piezoelectric crystal, a distortion is created in the length or width. Applications of piezoelectric actuators include printers, speakers, or switches.

[1] http://en.wikipedia.org/wiki/Physical_property

- **Optical and opto-electronic actuators:** Light is emitted by electrical energy. These types of actuators are used for the direct output of illumination such as for spotlights or in optical transmission technology, e.g. in optocouplers, displays, or for image recognition.
- **Magnetostrictive actuators:** The variation in length of ferromagnetic materials in magnetic fields is used here. These types of actuators can be used under high pressures and at high temperatures as very precise final controlling elements in the micrometer range.
- **Electrostatic actuators:** Electrostatic fields are used to move small mechanical parts in micro system engineering. This technology is used in speakers, for moving micro mirrors in optical switches, and in printing technology.

The most important actuators in the amateur field include motors, lamps, speakers, pneumatic cylinders, ultrasonic motors, and diaphragm pumps.

1.2 Open-loop and closed-loop control

As a rule, inputs and outputs are used in control technology.

When dealing with automation engineering, the terms “open-loop control and closed-loop control” are often used. However, what do these terms mean in engineering? This is explained in the standard IEC 60050-351. “Open-loop control” or “closed-loop control” are defined there as follows:

“... Controlling is a process in a system in which one or more input variables influence one or more output variables in a specific way. The flow of information for this is open. A signal is entered in the technical system as an input variable and it shows up at another location in the system as an output variable. This output variable does not influence the input variable in any way.”[2]

On the other hand, closed-loop controlling is a process *“... in which a variable and the control variable (the variable to be controlled) are consecutively recorded, compared to another variable (the command variable), and influenced in terms of harmonization with the command variable.”*[3]

1.2.1 Classification of open-loop control systems

Essentially, all control systems have the same operating principle: An input variable influences an output variable in a system in a targeted manner by means of specific laws. Controls do not make any comparison of input variables and output variables; their action flow is open. Depending on the type of control, a distinction is made between command and program controls (Fig. 1.3).

The distinguishing options of commonly used controls are, for example, the energy of the power supply for the system, the type of signal processing, an analog or digital process, or the controlled variable. As shown in Fig. 1.4, the technical representation of a control is implemented using block diagrams.

[2,3] Cf. Beuth (2014): DIN IEC 60050-351 – International Electrotechnical Vocabulary

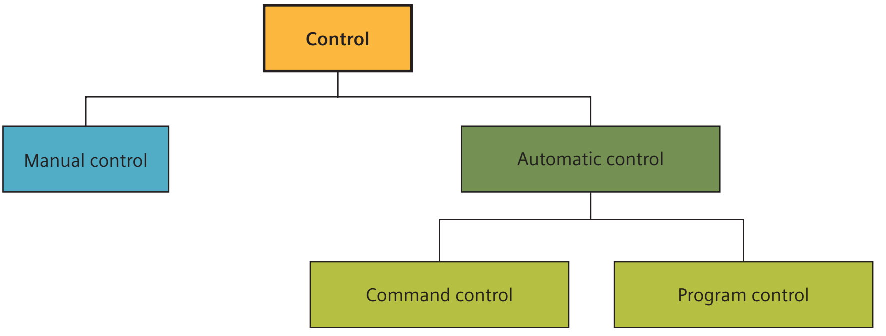


Fig. 1.3 Systematization of the control systems

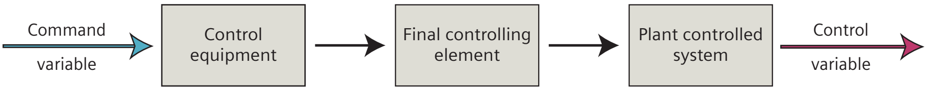


Fig. 1.4 General block diagram of a control system

The simplest form of control is the manual control system. It can be explained using the simple examples of opening a faucet, pressing a button for a doorbell, or steering a bicycle. This is shown in Fig. 1.5.

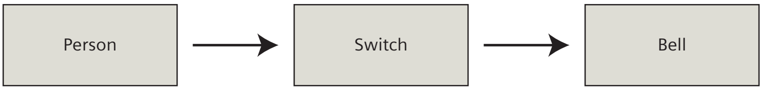


Fig. 1.5 Block diagram of a doorbell system

Automatic controls are more complex and function even without an operator influencing the system. With this type of system, the controlled variables are independently adapted to the specified command variable.

This procedure can be easily illustrated via an illumination system:

As soon if someone enters a building at night, the light is supposed to switch on. The input of the motion detector receives signals. The acquired signals are processed by a circuit. A lamp is connected at the output of the system. Once the detector registers a movement in its detection range, it causes the lamp to switch on. The circuit diagram and block diagram of this system is shown in Fig. 1.6 and 1.7.

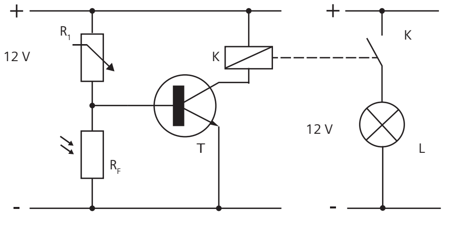


Fig. 1.6 Circuit diagram of an illumination system

Fig. 1.7 Block diagram of an illumination system

1.2.2 Classification of closed-loop control systems

In contrast to an open-loop control system, the variable to be influenced in a closed-loop control system is permanently measured and compared to the setpoint that is saved in the system. When the actual value (control variable) deviates from the setpoint (specified command variable) due to an external disturbance variable, the system responds and corrects itself. To do this, the control variable must be constantly recorded and compared to the command variable:

Control deviation = Command variable – Closed-loop control variable

This task is generally taken over by an electronic control system or a computer between the final controlling element and the output. If a control deviation occurs between the setpoint and actual value, a control signal is automatically output and the deviation of the values is compensated. The operating principle can easily be illustrated using a heating system as an example:

An operator sets the desired room temperature. The desired temperature is a setpoint, meaning that the temperature in the room should remain at this value. A final controlling element now consecutively checks whether the desired temperature has been reached. The actual room temperature is thus an actual value. If the room temperature is too cold, the final controlling element ensures that more hot water is directed into the heating element. Fig. 1.8 and 1.9 show a schematic of this.

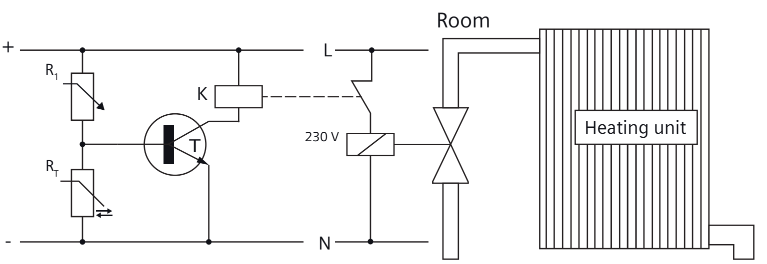


Fig. 1.8 Circuit diagram of a heating control system

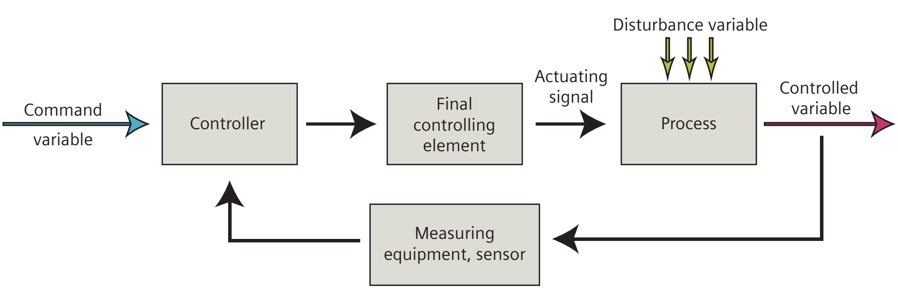


Fig. 1.9 Block diagram of a heating control system

Depending on the type of closed-loop control, a distinction is made between continuous and non-continuous control. For continuous control, an infinitely variable, mostly linear control device is used. For non-continuous control, on the other hand, this is a control unit which, in contrast to the continuous output adjustment, conducts intermittent ON/OFF operation. The diagrams in Fig. 1.10 and 1.11 illustrate this.

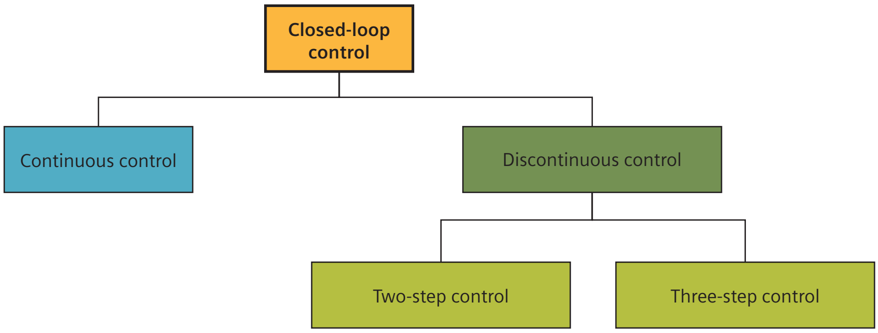


Fig. 1.10 Systematization of closed-loop controls

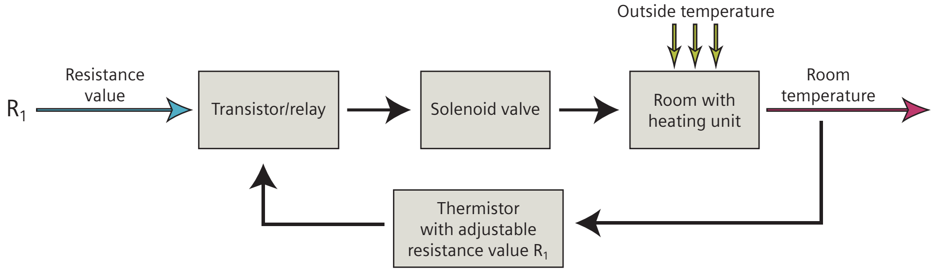


Fig. 1.11 Block diagram of a closed-loop control

In the annex in Chapter 7.1, there is a table displaying signal designations.

1.3 Programmable logic circuits and programmable relays

Automatically executing processes and movements have always been of interest to humans. With the aid of so-called programmable logic controllers (PLCs), a wide variety of processes can be automated and used in many diverse areas of application. More precisely, the LOGO! used in this book (Fig. 1.12) is a mini-controller, a so-called DDC module (Direct Digital Control). This computer-like electronic module is used for open-loop control and closed-loop control tasks for a wide variety of automation tasks.

The control systems of transport systems, elevators, conveyor belts, monitoring systems or house and building technology, heating systems, ventilation systems, and blinds can be implemented using LOGO! with simple technical know-

Fig. 1.12 Basic unit LOGO! 8

how and a minimum of resources. During this, the DDC module has the task of processing signals coming in from the sensors and controlling the actuators depending on the programming, in order to respond to changed environmental conditions.

The program needed for the control is either entered directly on the device via keys or created on a computer with the easy-to-learn and easy-to-use, user-friendly LOGO! Soft Comfort software. When the program is created, a graphical circuit diagram is drawn as function block diagram (FBD) or as ladder logic (LAD) via drag & drop.

The digital logical gates are the basis of the circuit diagram. In addition to the basic digital engineering functions (AND, OR, NOT, etc.), LOGO! Soft Comfort also provides a large number of special functions (e.g. latching relay, ON/OFF delay, operating hours and pulse counter). After a program is created, LOGO! Soft Comfort gives you the option of testing the results in a simulator. If the program meets the requirements, it can be transferred to the hardware via a special interface. Changes and fine adjustments can also be made later using this method or directly on the device via the keys whenever required.

In comparison to other systems or micro-controllers, LOGO! stands out due to the possible cycle frequency, the number of inputs, the individual programming interface, and a multitude of expandable components. The different types and designs of LOGO! and their expansion components can be learned in the Computer based Training (CBS) along with an introduction to working with Siemens LOGO! Soft Comfort. You can find the CBS on the following web site: www.siemens.de/sce/logo

2 Introduction to working with Siemens LOGO! Soft Comfort

The software LOGO! Soft Comfort is used for the creation of programs for LOGO!. The current version is V8.

2.1 The program interface

Three types of projects can be created using the LOGO! Soft Comfort software: Function block diagram (FBD), ladder logic (LAD), and UDP diagram (UDF). Only function block diagrams are created in the later examples. This is explained in more detail in Chapter 4.7. The program interface is comprised of various elements, which are displayed in Fig. 2.1.

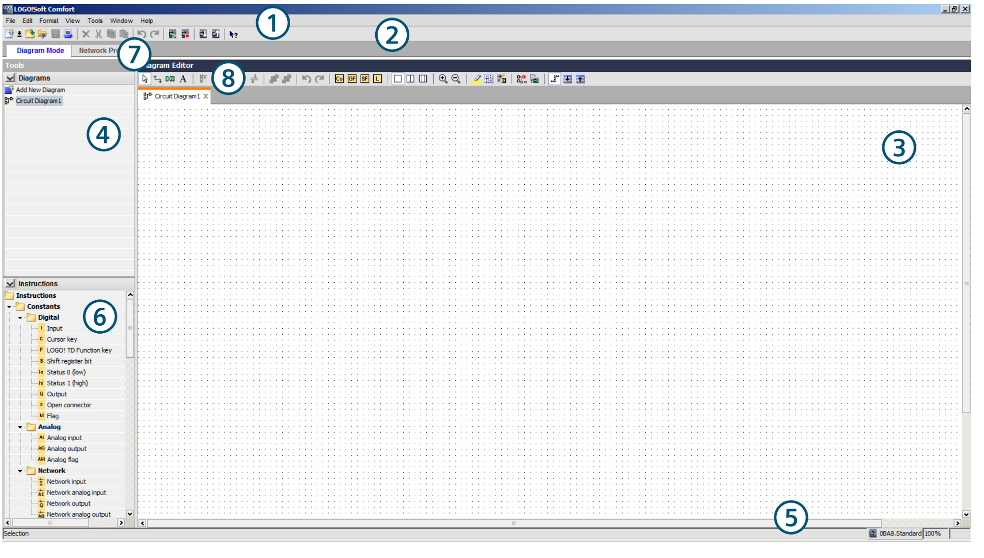


Fig. 2.1 Program interface in LOGO! Soft Comfort

① Menu bar

② Toolbar – Standard functions

③ Programming interface

Here, the individual blocks are inserted from the left window (6) and connected to each other. The program, which subsequently is to be transferred to the LOGO!, is thus written on this interface (3).

④ Info window

All of the messages that are relevant to program execution are displayed in the info window, e.g. project names, overviews, etc.

⑤ Status bar

Among other things, the selection of the LOGO! model (e.g. 8) is displayed in the status bar, including which tool is currently selected, etc.

⑥ Block selection/catalog

All of the blocks to be used can be inserted into the program interface (3) from the selection.

⑦ View mode

Capability of switching between network mode and diagram mode.

⑧ Toolbar – Tools

The individual functions of the tools are displayed by positioning of the cursor on them. Various tools (Fig. 2.2) aid in creating and editing a program.

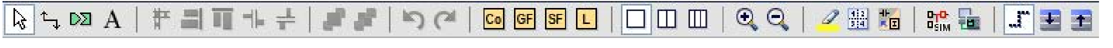


Fig. 2.2 Toolbar – Tools

2.1.1 Developing a program

The execution of a program is essentially comprised of the two states 0 and 1. This is called a binary system (bi = two), which operates using just these two numbers. State 1 describes an active state in which a signal arrives at an input or is pending at an output. State 0 describes an inactive state in which no signal arrives at an input or is pending at an output. The program sequence is only the analysis or passing on of these two states. The analysis of a state can be processed with various blocks. For example, there are logic basic functions in the block selection, the simplest of which will be explained here in brief – AND, OR, NOT, and NAND:

- AND sets the output to state 1 if all inputs have state 1.
- OR sets the output to state 1 if at least one of the inputs has state 1.
- NOT inverts the input state, i.e. during an active state (1) at the input, the output is set to passive (0) and during a passive input state (0), the output is set to active (1).
- NAND sets the output to state 1, if none or at least one input have state 0. If all inputs have state 1, the output is set to 0.

In digital technology, a computation with the states 0 and 1 can be found in every circuit. Computation is done using Boolean algebra. With the aid of Boolean algebra, complex switching programs can be simplified and summarized.

The following program serves as an explanation (Fig. 2.3): A a pushbutton (NO contact) is connected at the LOGO! input I1. When this is pressed, it conducts the electric current, which results in a state change from 0 to 1. If it is not pressed, it does not conduct the electric current and the state at input I1 remains 0.

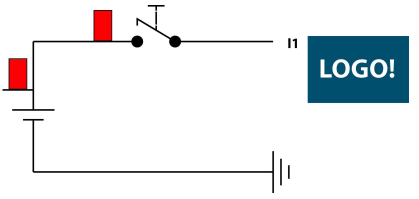


Fig. 2.3 Example of a program sequence

The signal that is pending at I1 should be analyzed: Once the state at input I1 is active, i.e. = 1, at output Q4 a lamp should be activated. The program then looks like the following:

Input I1 and output Q4 are directly connected to each other. The input at Q4 thus always assumes the state of the output of I1 (Fig. 2.3).

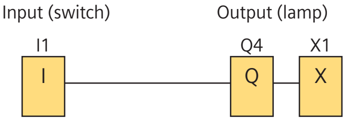


Fig. 2.4 Example of a program sequence

Table 2.1 Function table

Block name	Property	Function in the program
I1	Digital input	Record and forward an input state
Q4	Output	Switching an actuator (lamp)
X1	Open terminal	No open connections

The sequence of a program thus always runs from an input value/state at the digital/analog inputs, an input (I), to an output (Q). To analyze and process a signal for more complex circuits, basic logic functions or signal-analyzing blocks are interconnected between the input and the output (cf. Chapters 4.1 and 4.2). If output Q4, for example, is only to be activated when I1 and I2 simultaneously have status 1, both inputs I1 and I2 have to be interconnected by the basic logic function AND (&) (Fig. 2.5).

Each program sequence takes place in individual analyses of signals, which are sent to an output and then activate it.

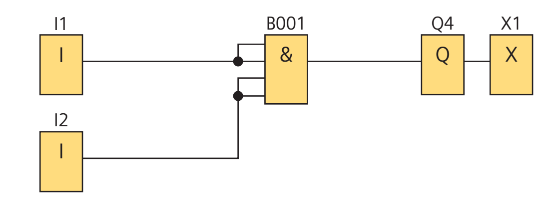


Fig. 2.5 Expansion of the program

2.2 Settings

2.2.1 Block properties

The individual blocks can be edited by selecting them via double-click:

A window in which the parameter and simulation settings for the selected block can be edited opens. The sensor should be connected to terminal I4 of LOGO!, which is why the input number must be selected accordingly (see Fig. 2.6). Comments and simulation settings for the selected block can also be inserted via the setting window.

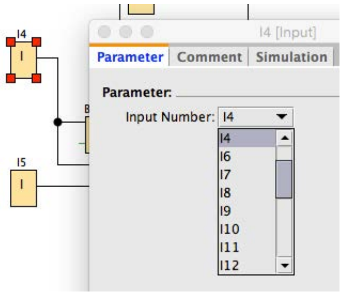


Fig. 2.6 Block properties

Another option for setting individual block properties is provided in the menu bar under *Edit > Block properties (all blocks)*... Here you also find an overview of all of the installed blocks and their properties. A "+" is placed under some blocks. Clicking on "+" opens a window which displays the set parameters or current values. One particularly comfortable function of the pull-down menus is the capability of connecting individual parameters/values to one another:

The counted measurand of the up/down counter block B001 (see Fig. 2.7) can be processed directly in the arithmetic instruction B003. Alternatively – by double-clicking on individual blocks – a reference to the respective other block can be set. This is automatically generated by connecting the values. Settings and parameterizations of various blocks and message texts can be found in the examples in the following chapters.

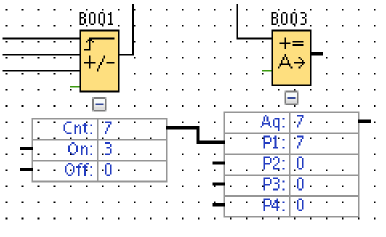


Fig. 2.7 Parameter field opened

2.2.2 Basic program settings

Language settings, etc. can be found in the menu bar under *Tools > Options....* The formatting and settings for the grid of the programming interface, etc. are offered in the menu bar under *Format*. All of the other settings via the menu bar are self-explanatory.

2.3 The help function

The help function is an important component of LOGO! Soft Comfort. The functions of individual blocks are explained using examples. The help function can be called either in the menu bar (*Help*, *Contents*) or by positioning and holding the cursor on the blocks in the block selection: The name of the block where the cursor is positioned appears with a "?" next to it. Clicking on the question mark takes you directly to the help menu. The help menu is very clearly laid out and is self-explanatory. In addition to brief descriptions, you can find detailed explanations there.

LOGO! define is another auxiliary function, which helps in selecting the right hardware and the right model of LOGO!. If writing of a program is finished, the *LOGO! define* function can be opened under *Tools > LOGO! define*. All of the LOGO! models that can be used for the written program are displayed in the info window (see Fig. 2.8).

```
Info Window
The following LOGO! can be used:
0BA6.Standard
0BA6.ES3
0BA6.ES10
0BA7.Standard
0BA7.ES4
0BA8.Standard
```

Fig. 2.8 Specifying the LOGO! info window

2.4 The simulation mode

In simulation mode, written programs can be tested in advance without being transferred to the hardware. The simulation mode is started by pressing the simulation button (Fig. 2.9). The program runs in the same order as it can be used in practice.

Fig. 2.9 Toolbar – Tools

The digital inputs are displayed as action fields, which can be operated by a simple mouse-click. Alternatively, you can click directly on the blocks of the digital inputs. The analog inputs, on the other hand, are displayed with a scroll bar, which changes a value when it changes position. The analog value can also easily be entered via the keyboard in the numeric keypad of the analog input.

Depending on the default settings of the digital inputs, these function as switches, pushbuttons (NO/NC), or a frequency. The pre-settings can be made by double-clicking on the block and selecting the Simulation tab (Fig. 2.10).

Fig. 2.10 Action fields in simulation mode

In simulation mode, individual block parameters and properties can be changed by double-clicking. Thus, various values can also be adapted during simulation mode. Active inputs/outputs are always marked in red during simulation.

3 Connection of LOGO! and transfer of a program

Before integrating a LOGO! in a complex hardware system, it makes sense to check all of the functions used. This is supported by the excellent simulation function of the LOGO! Soft Comfort software. This function allows you to check all of the program sequences. Only pushbuttons must be replaced by switches. In addition to checking the software, it also makes sense to test the hardware to be used. So-called practice or simulation boards are available for this (Fig. 3.1).

3.1 Preassembled practice boards for simulation

Section 5.3.1 explains how to make such a board yourself. If you do not want to make your own board, you can purchase a completely assembled one, e.g. from IKH Didactic Systems (IKHDS). IKHDS offers various combinations of boards and LOGO! versions on its Web page. Additional modules can be added to the IKHDS board via an interface connector. If need be, learning aid cards can be placed on the free space above the switches. This space can also be used, however, for your own notes, diagrams, or sketches.

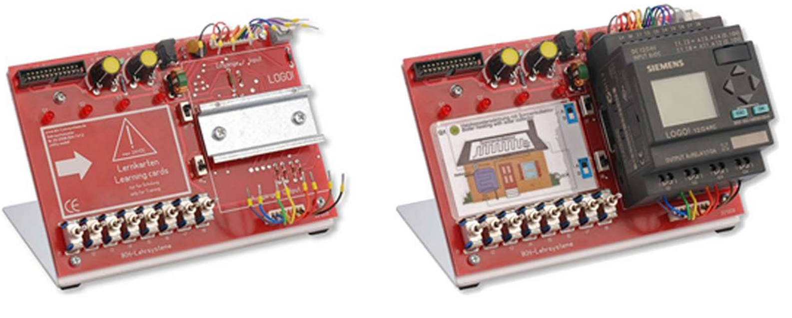


Fig. 3.1 Completely assembled simulation boards with a 0BA7 installed

3.2 The connection of sensors and actuators

Like the predecessor models, LOGO! 8 has eight inputs and four outputs in the basic units. The inputs are digital inputs, four of which can also be used as analog inputs with a range of 0 to 10 V. When digital sensors are connected, especially users in the “amateur field” must be aware that the LOGO! only analyzes a 1 signal at 8.5 V and higher. Values under 5 V are assessed as 0 signals. Thus, it is not possible to use sensors designed for TTL and CMOS, which usually generate a 5 V edge. And edge change from 0 V to 5 V does not lead to a 1 signal for the LOGO! Therefore, it is either necessary to use sensors which generate an edge of more than 8.5 V or to amplify the signals of 5 V sensors so that the LOGO! can analyze them. One option for amplification is described in Section 3.9.3 "Integration of the LOGO! into a 5 V system". The output signal of the sensor then only has to be connected at the screw terminal on the top of the LOGO!.

Analog signals can be processed up to an input voltage of 28.8 V. It is not possible to measure currents with the basic LOGO! unit. Only the analog expansion can measure currents between 0 and 20 mA. The digital inputs I7 and I8 correspond to the analog inputs AI1 and AI2. In addition, the inputs I1 and I2 can be used as analog inputs – the LOGO! detects this via the program.

These inputs are then AI3 and AI4. This is confusing and can lead to errors because it would be more logical to consecutively number the analog inputs corresponding to the digital inputs from left to right. However, this is not the case and must be observed!

Four digital relay outputs are available on the output side. If one of the outputs in the program assumes a high state,[4] the corresponding relay on the basic LOGO! unit is activated. The LOGO! has two screw terminals per output for connecting actuators at the outputs Q. The power source is connected to one of the terminals (marked with 1). The input of the actuator is connected to the other terminal (marked with 2). Switching of the relay then leads to power being supplied to the actuator. Since this is a mechanical switching process, the relay is subjected on the one hand to continuous wear, on the other hand it can only output a limited switching frequency. The manufacturer indicates 10 Hz. It must also be noted that the relay is only designed for 10 A continuous current.

3.3 Power supply

In the examples listed below, only the LOGO! 12/24 V DC is applied (Fig. 3.2). If a LOGO! is installed for stationary use, commercially available 12 V power supply units can be used as power sources (transformers used in model-building should not be used). It is more convenient to use a DIN rail transformer, which can be connected directly next to the LOGO! for a stationary installation.

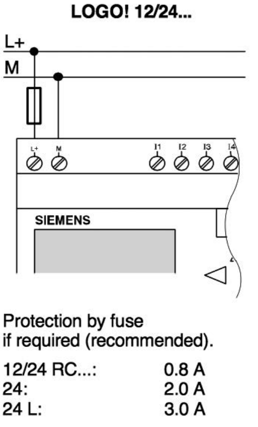


Fig. 3.2 Terminal assignment of LOGO!

If the LOGO! is to be used in a mobile application or if it cannot be connected to the mains (e.g. if installed in a summer house), it is possible to use a 12 V battery or a rechargeable battery (e.g. car battery). Depending on the rated input of the

[4] Two voltage ranges are allowed for digital signals. These are called high state or high level (also H level, high, H), or low state or low level (L level, low, L).

actuators that are connected to the outputs Q1 to Q4 of the LOGO!, you must ensure that the power source has a corresponding rated output. When high-performance actuators, e.g. motors, are used, an additional power source must be used for them. If the motors were connected in parallel to the LOGO!, this could cause a complete power failure for the LOGO!. It is therefore recommended that you use a battery to supply power to the LOGO! and a separate battery to supply power to the actuators.

When connecting the LOGO! to a 230 V mains, you must observe the general safety conditions. It is always recommended, however, to integrate a fuse into the system.

3.4 LOGO! in the network

The desire for a web server has long been expressed in a variety of Internet forums. The LOGO! 8 fulfills this wish. With the web server, Siemens has opened up a variety of options for home users to help them use the LOGO! even more effectively. This chapter will give an introductory explanation of how the LOGO! 8 can be integrated into a network and how it uses the web server.

In this context, only the simplest applications are introduced. However, this should be completely sufficient for a basic understanding and simple applications. At this point, we refer again to the excellent help function in LOGO! Soft Comfort.

3.4.1 The capabilities of a web server

The web server allows the user to access the LOGO! control unit via a browser such as Internet Explorer or Mozilla Firefox. On the one hand, this is possible via the Internet, which allows remote maintenance and checking from anywhere in the world where there is access to the Internet. On the other hand, it also allows it to be used in a Local Area Network (LAN). In a LAN, the computer on which the browser is running and the LOGO! must be located in the same network. Even if the technical prospects are appealing, this book will not go into detail about accessing the LOGO! via the Internet. At this point, it must be noted that the integration of the LOGO! into a home network and the release in the Internet can be a possible point of attack by hackers.

Even without a release in the Internet, the integration into a LAN brings about some interesting possibilities. The following sections go into detail about

- how to control the LOGO! via a browser,
- how to integrate the LOGO! into the network and
- how programs are directly exchanged between the computer and LOGO! via the LAN cable.

Connection and integration of LOGO! in the network

LOGO! is connected to the network or a computer via an LAN cable. The LOGO! can also be equipped with a WLAN antenna as needed, as long as this does not require a driver to be installed. Devices of this type are specially provided for integrating LAN devices, i.e. non-WLAN capable devices, into a WLAN. The only requirement for the computer and the LOGO! to "find" each other is that the computer and LOGO! must be located in the same LAN. Thus, the first three blocks of the IP address must be identical and the last block must be different. The network settings of the PC can be viewed and changed in the "PC settings" menu (Fig. 3.3).

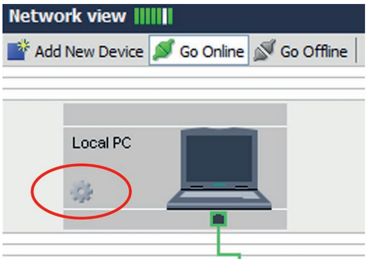


Fig. 3.3 Changing the settings of the local PC

The settings for the LOGO! are intuitive. The online operating manual will help you with any further questions regarding menu navigation. This part of the LOGO! operation is explained in detail there. For the IP address setting on the computer (Fig. 3.4), we refer you to the operating instructions of the respective operating system.

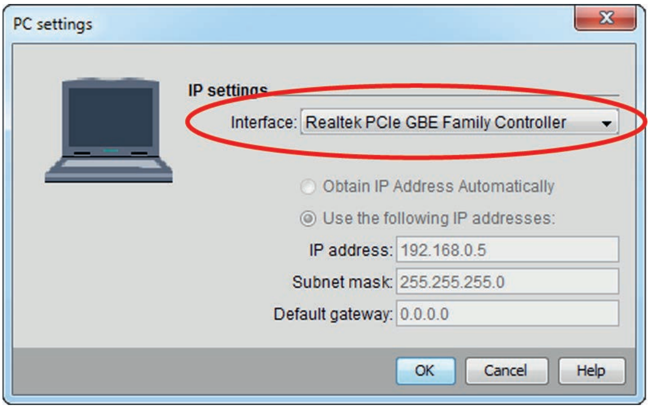


Fig. 3.4 Selection of the suitable network interface

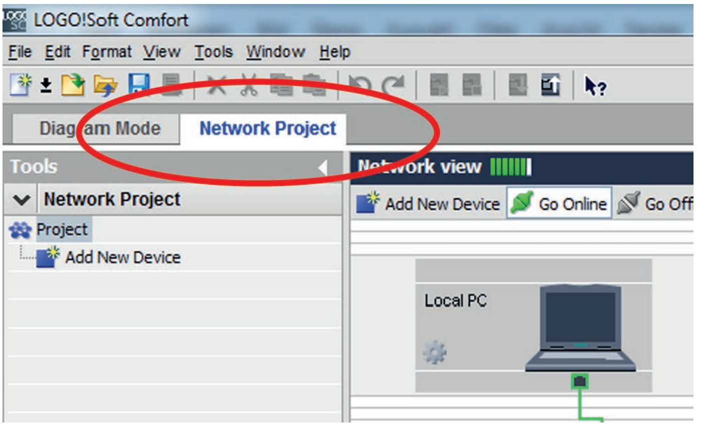


Fig. 3.5 Switching to the "Network Project" view

In LOGO! Soft Comfort you must first switch from diagram mode to network mode. This is done via the “Network Project” tab (Fig. 3.5) in the top left corner of the program window. In this menu, the “Go online” button must be pressed (Fig. 3.6). This causes LOGO! Soft Comfort to search for components in the network.

Fig. 3.6 Going online via LOGO! Soft Comfort

If you press one of the menu keys for controlling the LOGO! and if no unambiguous connection to the computer has been established yet, a dialog in which the settings can be re-checked appears. Here, it is especially important to ensure that the network adapters are correctly designated. By clicking on the round refresh arrows, the network will be searched for IP addresses. If the address for the LOGO! appears, you must click on it once.

After this, a test of the connection is recommended. This should be confirmed by a green checkmark (Figs. 3.7 and 3.8). If no successful connection is achieved, the IP addresses should first be checked for compatibility. In addition, it is possible to enter the IP address of LOGO! manually. If a connection nevertheless fails to be established, we refer you once again to the help function in LOGO! Soft Comfort and Siemens product support.

Changes can be made to the program or data can be read from the log via the network. Thus, it is possible to install a LOGO! without a display and only permit access to the network connection.

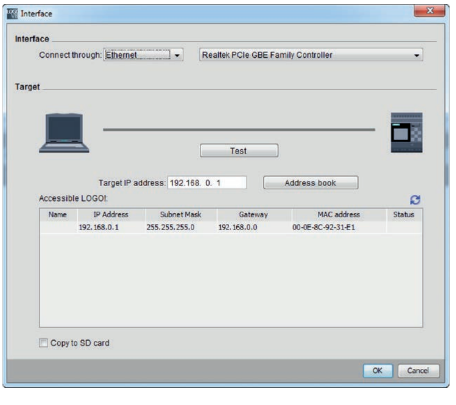


Fig. 3.7 Searching for the connection of a PC with a LOGO!

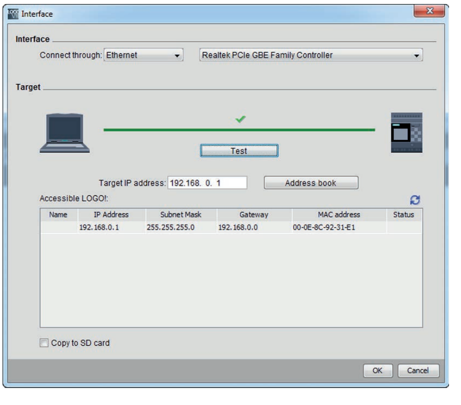


Fig. 3.8 Successful connection of the LOGO! to the PC after pressing the “Test” button

Controlling and reading access to the LOGO! 8 can also be done directly via the web server. Users can access the login page via the Internet browser by entering the IP address. The password must be entered on this page if one has been defined. In addition, languages can be selected (Fig. 3.9).

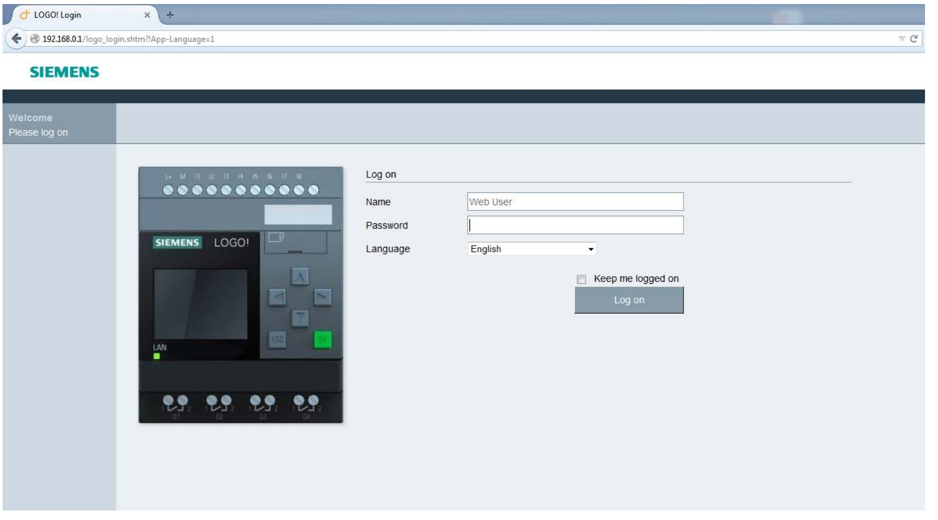


Fig. 3.9 The welcome screen of the new LOGO! web server with login fields

From there, you go to a menu with selection options on the left for status information about the LOGO!, variables and the views as a LOGO! with a display (basic module, BM) and as a text display (TD) are provided. In the BM and TD views, the displayed keys can be used just as they would be on a real LOGO! with a display or a TD. It makes no difference whether or not the installed LOGO! has a display (Figs. 3.10 and 3.11).

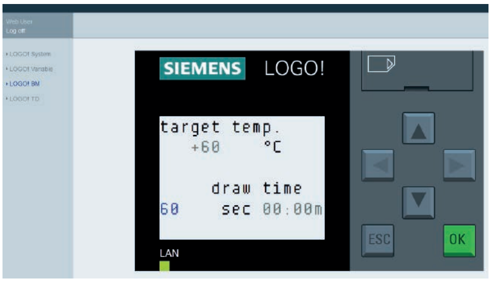


Fig. 3.10 LOGO! basic unit in the Web server

This saves the amateur user from having to buy a TD, because all of the functions can be implemented via the browser. If a LOGO! is to be installed in such a way that the display would not be visible, you can omit the display due to the use of a browser. The keys also do not have to be accessible as long as the network cable is accessible. Information on the display can be called up and adapted from any point in the network.

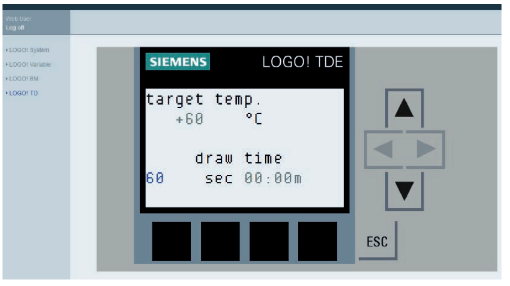


Fig. 3.11 LOGO! TD view

With the functions described it would also be possible to modify the parameters for watering the flowers on the balcony or the lighting in the living room from your holiday resort via the Internet. Nevertheless, we point out once again the risks that exist due to the integration into a home network with Internet access:

The LOGO! can be used as an entry point for hackers. Over-controlling in the LOGO! can have grave consequences in home automation applications. Nevertheless, with this web server, Siemens has succeeded in fulfilling a long-standing request from its customers.

3.5 External storage media

The LOGO! 8 gives the user the capability of saving data on a Micro SD card. Handling was also a primary concern for the designers of the housing for use with the small memory card. The memory card can be inserted from the side into a sliding unit in the top right corner of the housing. The sliding unit can be opened forward using a sharp object, e.g. a screwdriver. It is very advantageous that the sliding unit cannot be removed. This prevents the small component from becoming lost. The memory card is then inserted from the left and latched, just like SD cards are inserted into other devices.

When transferring data from the LOGO! to the memory card, the data is first temporarily saved in an internal memory of the LOGO! and then transferred to the SD card. Data can then be read out via a connected computer. The "DataLog" block is used in the LOGO! program for logging data (Fig. 3.12). During this, you must observe that data can be lost if the LOGO! generates data faster than it can be transferred to the memory card.

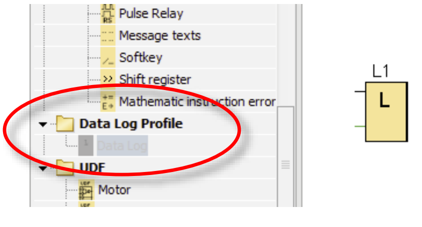


Fig. 3.12 Selection of the data log block

The function is available once per program. After the block is placed, the drag & drop selection changes to gray and can no longer be selected. In the block configuration, you can set which states are to be saved. Respective digital and analog inputs, outputs and bit memories are selectable options for this. The data is always grouped into 8-bit groups. In the case of analog signals, the 8 bits are used by a block.

Digital blocks are grouped into 8-bit groups by the program and can also only be saved in this way. A total of 32 groups can be saved. This data can be transferred in “Stop” mode from the LOGO! to the PC in CSV format and be further processed there. Alternatively, the micro SD card can be removed and read.

The data log function gives the amateur user a powerful tool for gathering data over a long period of time during monitoring, for example, and for providing it for later analysis. This involves such things as weather logs, alarm systems, animal watching, or entry/exit/transit counters.

3.6 SMS function

Communication with the LOGO! was previously limited to the range and accessibility of networks. When operating a LOGO! outside of a network, GPRS communication had to be improvised or additionally purchased from external suppliers. With LOGO! 8, Siemens is also putting the CMR2020 module for SMS communication for the LOGO! onto the market (Fig. 3.13). Thus, there is a “LOGO! family solution” for exchanging information even if no network or cable connection exists or none is possible. The CMR2020 module not only allows the LOGO! to send text messages to its users, it can also receive text messages itself.

Fig. 3.13 The CMR2020 SMS module for LOGO!

In addition, the CMR2020, with its two digital inputs, is capable of responding to digital input signals from a sensor or from a LOGO! which is not connected via the network cable. The CMR2020 can communicate with actuators or control units via two digital outputs.

A network connection is available for connecting to a LOGO!. More complex information than the digital ON/OFF signals can be exchanged via the network connection, which can be sent and received via the digital inputs and outputs: CMR2020 can read and forward the states of individual blocks of a switching program. It is also possible to read specific control commands for the LOGO! from a text message. And system statuses can be sent. The additional Global Positioning System GPS tracking permits LOGO! and CMR2020 to exchange coordinates. It is worth noting that CMR2020 can also function without LOGO! via its inputs and outputs, allowing a number of simple applications (Fig. 3.14).

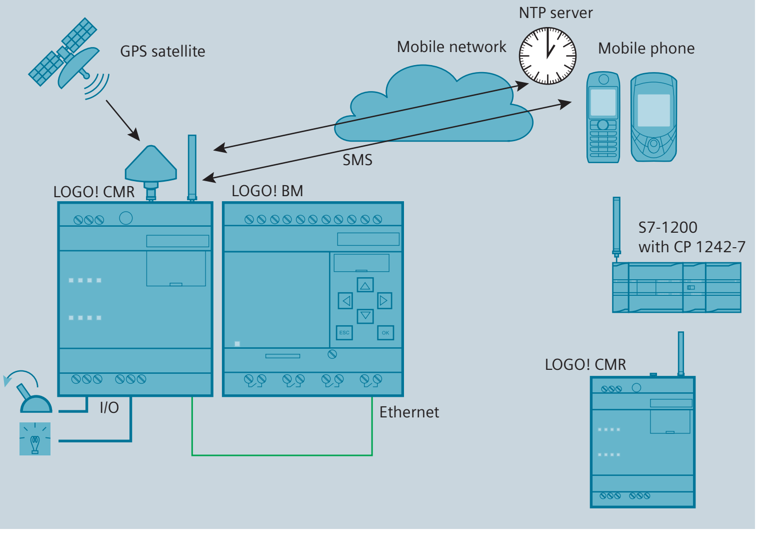


Fig. 3.14 Integration of the CMR2020 SMS module in a project

The CMR2020 is protected against unauthorized "access" by "external" test messages by defining up to 20 users with telephone numbers and authorizations. Only these users can receive or send text messages to the CMR2020 as per the saved authorizations (Fig. 3.15).

A SIM card is required for operation. It can be inserted via a card slot, similar to the SD slot of the LOGO!. Unfortunately, the CMR2020 regarding to the card slot is not as easily solved as in the LOGO! 8. The SIM card should therefore be inserted in an environment where small parts that were dropped can be easily found.

CMR2020 is programmed via the Web browser. To do this, it is necessary to connect the module to a computer using the network cable. The instructions in the manual should be followed for this, because not all of the settings are intuitive. The CMR2020 has LEDs for displaying the status messages, but it does not have a display. Changes can be made to the settings via the browser after the password prompt. Unlike with the LOGO! 8, which can also be protected from program changes via a password, CMR2020 has a Set key on the front side which can be used to restart, shut down, or reset the device to the factory settings. On the one hand, the key is an aid for when the access data have been forgotten or a restart is necessary. On the other hand, the CMR2020 is susceptible to misuse due to the openly accessible Set key. For this reason, we recommend that access to the Reset key be prevented during "openly accessible" operation by means of suitable hardware measures ("firewall", etc.).

During operation, CMR2020 can communicate with a LOGO! and query the status messages of individual blocks, inputs, and outputs and send them via SMS. In addition, the user can intervene in the sequence of LOGO! programs and even set the entire LOGO! to the ON or OFF state. It is convenient that text message commands can be used to query the balance of prepaid cards (rechargeable phone cards) and recharge the cards without having to take them out of

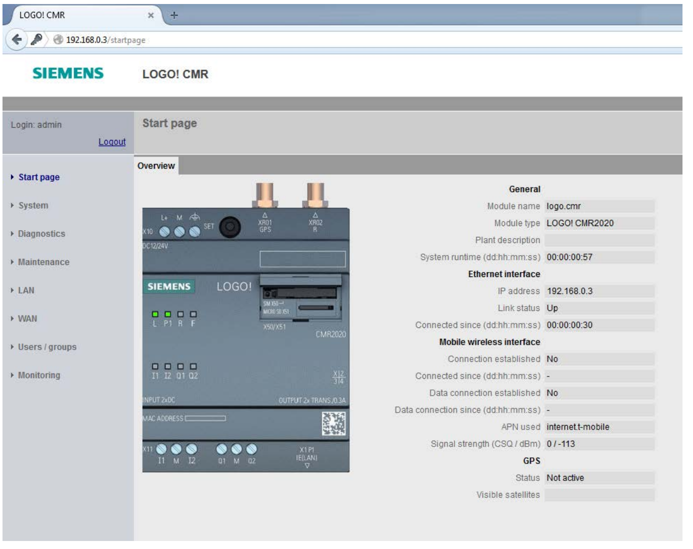


Fig. 3.15 SMS notification

the device. Also changes can be made remotely to the list of authorized users and their telephone numbers (Fig. 3.15).

The possible applications are: all alarm systems, security and monitoring measures of critical processes for which the response should vary depending on various parameters.

For example, as part of a WLAN-free home garden system, the communication module can alert the guards, owners or police via text message whenever suspicious motions are detected and evaluated by a LOGO! via a motion detector. After the SMS alert, the user has the capability of initiating appropriate measures such as lights or sounds, also via SMS message. Of course, it would also be possible to connect a motion detector directly via a digital input, but an additional circuit would then have to rule out the possibility of small animals tripping the alarm as they pass by.

This manual covers in Chapter 6 various circuits in which the use of the CMR2020 can be a helpful augmentation to the program that is presented. In the garden watering circuit (Chapter 6.7), in which an empty cistern or water tank is automatically refilled via a solenoid valve on the water line, the CMR2020 could send a text message about the low level in the tank. Then the user would have the capability, for example if rain was forecast, of disabling the “paid” fill-up from the tap via SMS.

In terms of providing support to persons in need of help (Chapter 6.14), CMR2020 can alert emergency services or family members via a remote signal from a person who fell or is involved in a similar emergency situation at the push of a button. The same would apply to the tripping of the automated alarms. In schools or other institutions, where an alarm system is desired for the entire facility, this can be triggered by an SMS message. The limitation of 20 users can be by-passed by assigning groups of persons. In a case of safeguarding a movable object, such as a car or boat, it may be beneficial to call up the current location via SMS and locate possible thieves. It would also be convenient to preheat, air-condition or ventilate a vacation home via SMS a few hours before you arrive via a connected LOGO!. In the wintertime, an automobile could also be pre-heated or de-iced in this way.

3.6.1 Data exchange between CMR2020 and LOGO!

The CMR2020 provides the amateur user with numerous options for using the LOGO! 8 even more efficiently than before and to retain the capability of intervening at critical points in the program sequence – from the other side of the world if need be. Data exchange with the LOGO! is especially convenient. At this point, we must not neglect to mention that there are also other, possibly more favorable options for controlling and analyzing a LOGO! via SMS – but we have thus far not found a solution as convenient as this one. In addition, the products of other users can only communicate via the inputs and outputs of the LOGO! and thus cannot intercept any information from the program.

Users who need to receive more complex data in the form of analog and digital output values from the LOGO! program, get the capability to do this with the CMR2020. Thanks to the precise adapting and alignment to the LOGO!, a level of security which makes it possible to use it in industrial and “important” applications is provided. In this case, the investment is completely worth it. In the case of “amateur applications”, in which a system failure would have few serious consequences and, in the worst case scenario, would result in a vacation home not being pre-heated upon arrival, you can fall back on a GPRS system from the product group of the 5 V amateur electronics. When tracking maritime containers or integrating senior alert systems, we strongly advise against relying merely on home-made, untested circuits if you do not have solid expert knowledge. In these cases, the CMR2020 provides a proven and certified module that meets industrial standards.

At this point, we must not fail to mention that configuring the CMR2020 also brings limitations with it. Status messages and changes via SMS are done using a very inflexible syntax, in which even uppercase and lowercase must be observed in the SMS message. The CMR2020 also replies using a fixed format, which is not always intuitive. As it is the first device of its kind, the documentation on the Internet and in help forums is not as exhaustive as that for the LOGO! module. Thus, problems that are not yet listed in the operating instructions can lead to longer troubleshooting.

For a comprehensive description of the capabilities of the CMR2020, we refer you to the Siemens online information.

3.7 Expansion components for LOGO!

In some projects, the inputs and outputs of a basic LOGO! unit are not sufficient. Therefore, a large number of expansion modules are available, which can be used to expand the basic unit depending on the application. For example, to control even more complex systems, the LOGO! can be expanded to up to 24 digital inputs, 16 digital outputs, 8 analog inputs, and 2 analog outputs using expansion modules. Furthermore, it can be equipped with a communication module for an actuator-sensor interface (ASi), and for the European Installation

Bus (EIB). As of Version 0BA6, it was already possible to also use an external text display (LOGO! TD). The 0BA7 version, which was introduced in 2011, has a slot for SD cards and an Ethernet interface. Thus, it is possible to interconnect several LOGO!s. The GSM module, which has already been described in Chapter 3.6, allows SMS messages to be sent and is available for LOGO! 8.

The Siemens homepage provides a complete and comprehensive overview of all of the modules. For the projects described in this book, the following modules are primarily used:

- **Digital module**
  The digital module provides four additional inputs and four additional relay outputs. These are the same as the inputs and outputs of the basic unit and can be programmed and used in exactly the same way.
- **Analog modules**
  Whereas the digital module provides four additional inputs and outputs, there are different analog modules that differ in terms of the inputs and outputs of the basic unit. Thus the analog module AM2 only provides two additional analog inputs and no outputs.

  The inputs differ from the basic unit, however, in that they can measure both the voltage and the amperage in a range of 0 to 20 mA. Furthermore, the AM AQ module can be used for the 12/24 V LOGO!. This does not have any inputs, but it has two analog outputs, which can output either a 0 - 10 V or 0 - 20 mA signal.
- **TD text display**
  It also makes sense to integrate a text display for one or the other application. Most outputs can be displayed on the text display of the basic LOGO! unit, but built-in solutions will always require an external TD.

3.8 Trial project for circuit wiring

To test signals at the digital and analog inputs without having to wire the entire project, a simulation module can be helpful. Various, comprehensive and preassembled circuits are commercially available, or you can assemble them yourself as described in Chapter 5.3.1.

3.8.1 Circuit for hardware simulation

The following circuit will help in conducting an initial hardware simulation before your own project has to be wired (Fig. 3.16). Analog measured values are to be displayed and digital signals are to be processed for simulation purposes. If need be, simple 12 V bulbs (or LEDs with 400 Ω of series resistor) can be connected to the outputs of the LOGO!. Alternatively, the acoustic clicking of the relay during a switching process can also be recorded.

The two analog inputs AI3 and AI4 are connected to an analog threshold switch (B002 or B003) so that the output on the threshold switches activates when a defined value is exceeded. Once both values of AI3 and AI4 exceed the defined values, output Q1 is activated (condition given by & block B006). For example, if a switch that is connected to I3 is pressed until the ON delay time of B004 has elapsed and I4 and I5 are also pressed, output Q2 is activated. If a digital input signal arrives at I6, B005 activates output Q3 pulse-wise. If defined values at AI1 OR AI2 are exceeded, output Q4 is activated.

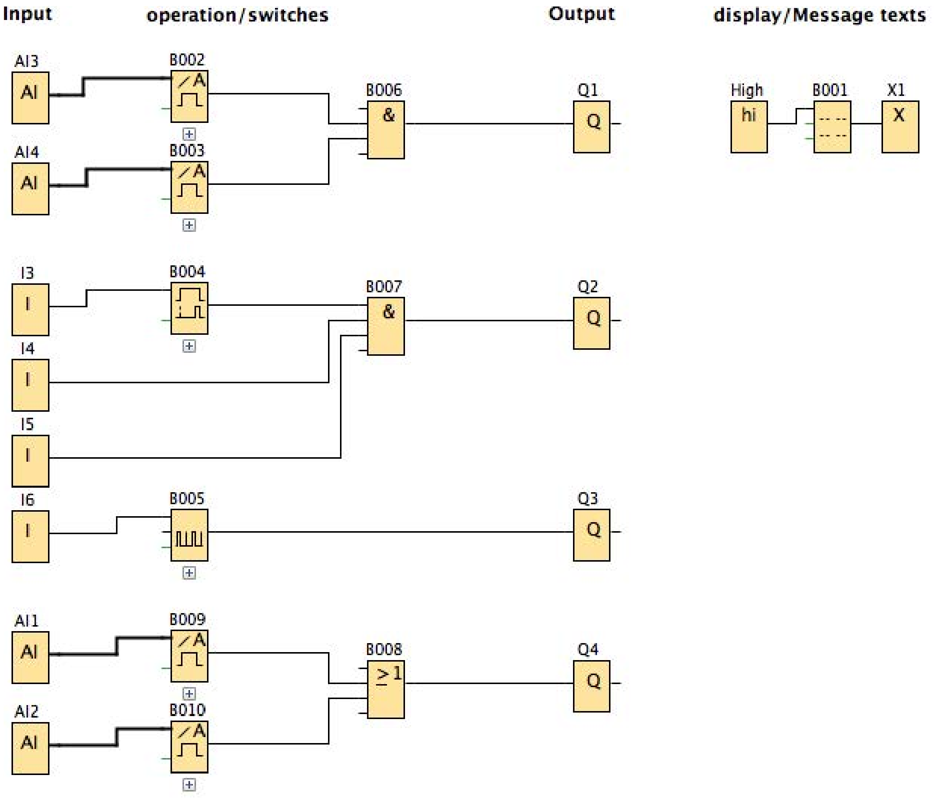


Fig. 3.16 Program for practice board

Table 3.1 shows the overview of all of the blocks used in the program with the respective properties and functions in the program. The table helps the user as an overview for improved orientation and order in the program. In this way, he can better track the switching processes and record the various application options of the blocks. Therefore, such a table can be found in all of the following projects in this manual.

The inputs can also be connected directly to the outputs without the special blocks – except for the analog inputs, which require either a threshold switch (B002, B003, B009, and B010) or another block, which analyzes the analog input signal (Fig. 3.17). In any case, a message text which displays the measured values of the analog inputs is recommended. Thus, for example, the changing light conditions can be connected via an LDR (Light Dependent Resistor) to an analog input and read directly in the message text. In order to be able to use all of the analog connections, corresponding default settings must be made in the program (cf. Chapter 4.2).

Table 3.1 Function table of a project with blocks, properties, and functions

Block name	Property	Function in the program
AI1 – AI4	Analog inputs	Analog signal input of the sensors
I3 – I6	Digital inputs	Digital signal input of the pushbuttons/ sensors
B001	Message text	Display/output of measured values
B002, B003, B009, B010	Analog threshold switch	Evaluation of the analog input signals
B004	ON delay	Time-delayed switching
B005	Incremental encoder	Outputs a pulse signal at the output
B006, B007, B008	Basic logic functions OR and AND	Conditions for activating the outputs
Q1 – Q4	Outputs	Outputs for controlling actuators
X1	Open terminal	Block required for message text
HIGH	State 1	Constant state 1 is always displayed in message text

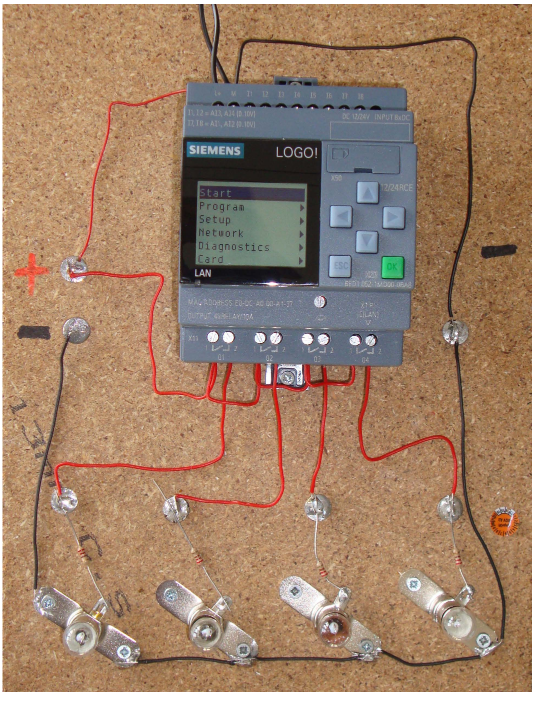


Fig. 3.17 Practice wiring on a wooden board with LOGO! 8

Tip

If the analog signal is too strong, a potentiometer can be connected upstream (cf. 4.2). If the signal is too weak, an analog amplifier block can serve as a signal amplifier.

3.8.2 A simple program: the twilight switch

A simple program for processing an analog value (an LDR is used as sensor) might look like the one in Fig. 3.18. Table 3.2 shows the function table for this.

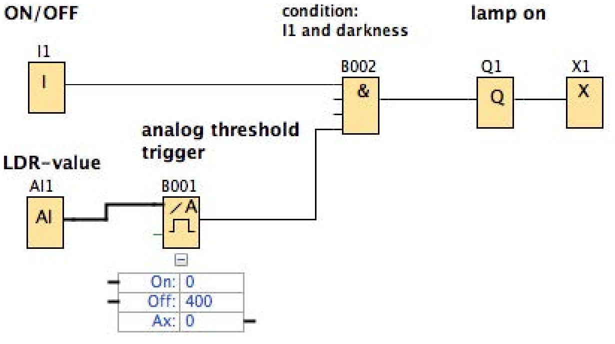


Fig. 3.18 Program for twilight switch

First, the main switch I1 must be set to "ON". If little light shines on the LDR, its resistance value is high. This means that the voltage *V* is lower when there is less light. As the light becomes fainter, the analog value at AI1 decreases. If the value is higher than 600, the analog threshold switch B001 activates its output and with the main switch setting "ON", a lamp for example, which is connected to Q1 could be switched on. If it becomes bright again, the resistance value of the LDR decreases and the voltage *V* increases again. The analog value at AI1 increases and the output at B001 no longer activates, so that the lamp goes out.

Table 3.2 Function table for twilight switch program

Block name	Property	Function in the program
AI1	Analog input (value of the LDR)	Analog signal input of the sensor (LDR)
I1	Digital input	ON/OFF switch, main switch
B001	Analog threshold switch	Monitoring of the LDR value
B002	Logic basic function AND	Condition: I1 On and darkness
Q1	Output	Connection, e.g. for a light source
X1	Open terminal	Circuit termination

3.9 Integration of the LOGO! into 5 V systems

TTL technology [5] is used in many circuits. The blocks are easy to procure and favorably priced for hobbyist users. Even complex sensor tasks, such as color recognition, can be acquired easily. TTL technology is a very good alternative to blocks in home user applications and it satisfies industrial standards and

[5] Transistor-Transistor Logic (TTL) denotes the circuit technology for logical circuits which are designed for operating on a 5 V power supply.

requirements. Even if it would be worth the effort for a amateur user to have a system that would not be out of place in an industrial manufacturing shop, such elements are too expensive for hobby applications.

The disadvantage of TTL technology, as well as the more sophisticated and susceptible CMOS technology[6], is that the high level lies between 2.4 and 5 V and the low level lies between 0.8 and 2.0 V. Thus, a TTL edge change cannot be registered by the inputs of the LOGO!. In order to nevertheless be able to use TTL blocks, an option must be found which amplifies the signal of the TTL sensors enough that it can be registered by the 12 V inputs of the LOGO!.

The remedy is the “L293D” IC circuit[7] in a DIL16 enclosure, which is available for less than 2 euros. With one of these L293D circuits, four currents can be activated independently of one another. The circuit analyzes a TTL input and, depending on the TTL signal, activates a load current of up to 36 V on an output pin. Four of these circuits are installed in an L293D. Diodes that capture induction currents are also installed in the IC. The L293D can process up to 1 A for longer periods.

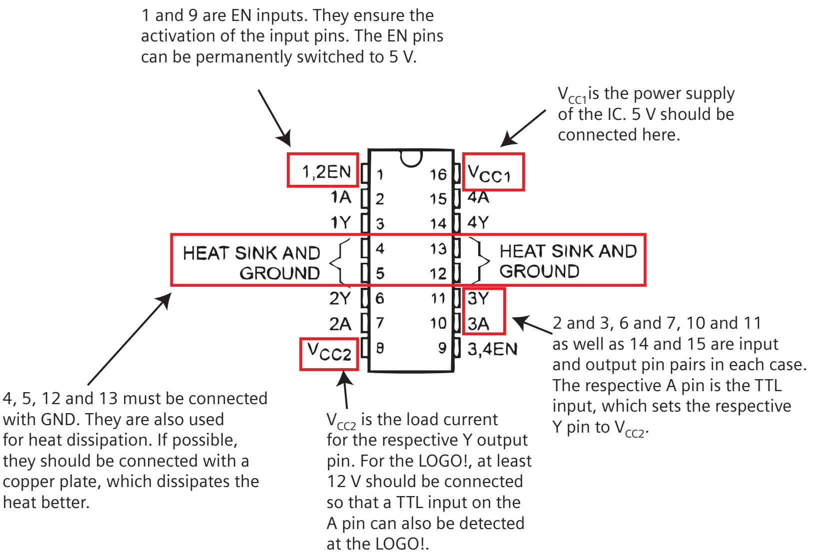


Fig. 3.19 The IC L293D with highlighted pin assignment

An example circuit would then look like the one shown in Fig. 3.20. Note that the data sheet of the IC must be consulted for the correct connections! Fig. 3.19 shows a schematic diagram for a TTL input. As already mentioned, up to four LOGO! inputs can be connected to an L293D.

It is possible to tap the 12 V for VCC2 from the power supply of the LOGO! and to reduce it so that it can simultaneously supply power to the sensors and ICs. For

[6] Complementary metal-oxide-semiconductor (CMOS) technology denotes the logic family of the semiconductor components, which is mainly used for integrated circuits.

[7] An integrated circuit (IC) is an electronic circuit installed on a semiconductor component.

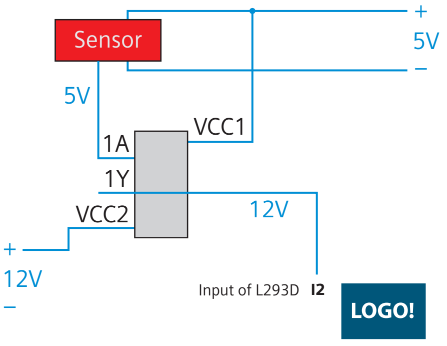


Fig. 3.20 Schematic diagram of the integration of LOGO! in a 5 V sensor system

the most flexible use, it is possible to solder two L293Ds on one PCB (see soldering course, chapter 5.3) and to supply them with 5 and 12 V as described. The A and Y pins can then be connected with terminals or connectors in order to be able to connect sensors to the LOGO! in as modular a way as possible.

3.10 Transfer of programs

There are two options for programming the LOGO!. On the one hand, the LOGO! can be programmed directly on the device. On the other hand, Siemens provides powerful and intuitively comprehensible programming interface in the form of LOGO! Soft Comfort. At this point in the book, we refer you to the option of direct programming. However, we will not describe this in detail. In comparison to direct entry, it seems to us that software programming is so much more convenient that we will only describe the software solution.

3.10.1 Sending programs to the LOGO!

The circuits that have been programmed in LOGO! Soft Comfort can be sent to the control unit directly from the software (Fig. 3.21). To do this, there must be a connection between the LOGO! and the PC. This is possible either via a USB connection, COM connection, or via the LAN. Regardless of the type of connection, a connected LOGO! can be started and stopped via the menu bar. In addition, programs can be sent to and from the LOGO!.

Any running program must be stopped for the transfer to LOGO!. Alternatively, the software asks via a dialog box whether the program should be stopped for the transfer or whether it should continue to run.

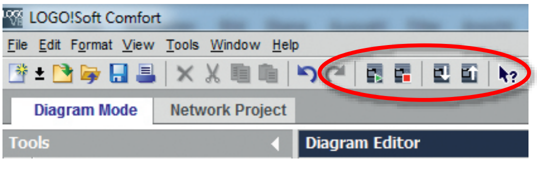


Fig. 3.21 Quick selection keys for the bi-directional transfer of programs between PC and LOGO!

3.10.2 Loading programs from the LOGO!

Just as it is possible to transfer programs into the LOGO!, it is also possible to transfer programs out of the LOGO!. There is also a button in the menu bar for this purpose. As with uploading, the running program must be stopped. Alternatively, the software asks via a dialog box about the way to proceed.

3.10.3 Protection of the LOGO! by the administrator mode

Prior to LOGO! 8, users of older systems complained that it was possible for unauthorized users to use the keys and display to interfere with the program execution directly on the device or to access it via the network. Version 8 is now equipped with an administrator mode. This only allows changes to be made on the LOGO! which do not influence the execution of the program. The password for connecting to the computer must also be known. The default password is "LOGO".

4 Simple exercises with the LOGO!

In this chapter, we want to describe the most basic switching options of the LOGO! in order to become more familiar with them. All of the topics described are used later in Chapter 6 in practical examples.

4.1 Switching by means of digital signals

In digital technology there are the states 0 and 1. State 0 means that no signal is present at an input, whereas state 1 indicates an incoming signal. The limits for the two states can also be defined; thus there is a so-called characteristic line for some components, which clearly draws the border between 0 and 1. We are mainly interested in digital inputs, which we find in the LOGO! Soft Comfort software as a block.

The digital inputs are connected to I1 - I8 on the LOGO!. In order to be able to use all of the inputs I1 - I8 as digital inputs, the corresponding settings must be made in LOGO! Soft Comfort in the "Properties" under "I/O settings" (Fig. 4.1).

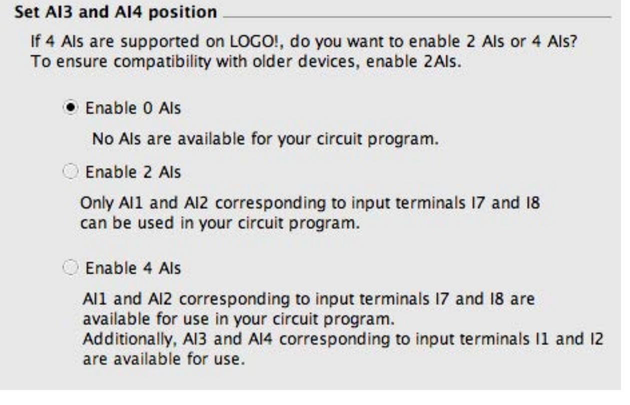


Fig. 4.1 Settings for digital inputs

For example, a pushbutton (NO contact) outputs an unambiguous signal 0 or 1: If it is pressed, signal 1 is output; if the pushbutton is not pressed, its state also does not change, it is in the initial position and no signal is output, the state is 0. The signal would be exactly the opposite for an NC contact.

The output at I1 is activated once a positive signal (state = 1) is present at the digital input of I1 on the LOGO!.

4.2 Switching by means of analog signals

Unlike digital signals, analog signals do not output a state 0 or 1. They are a direct conversion of a non-electric variable into an electrical one. For example, if you take an LDR, which has low electrical resistance in bright sunlight and the highest electrical resistance in darkness, you can recognize the change of state directly from the resistance value. As described previously, analog signals are often analyzed based on the principle of a voltage divider: The sensor must be connected to the power source at the analog input of the LOGO!. A partial voltage, the difference of the applied voltage and the voltage at the electrical resistance of the LDR, is fed into the LOGO! as measured value.

If, however, the signal is to imprecise or too high, a simple circuit with a potentiometer (adjustable resistor) can help (Fig. 4.2).

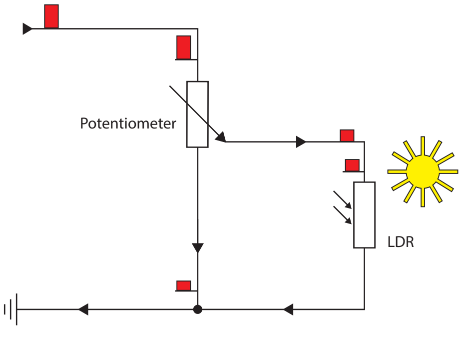


Fig. 4.2 Potentiometer circuit

By connecting the potentiometer upstream, the signal is "throttled", which allows the sensitivity of the sensor to be adjusted. Any sensor can be installed instead of the LDR.

The sensors for the analog signal are connected at either at I7 and I8 and/or at I1 and I2 on the LOGO!. The inputs that can be used must first be set in the software LOGO! Soft Comfort (cf. Chapter 4.1).

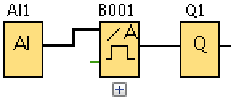


Fig. 4.3 Program example for analog signal analysis

An analog signal must always be analyzed with an analog block, e.g. an analog threshold switch (Fig. 4.3). The practical example of the twilight circuit was described previously in Chapter 3.8.2.

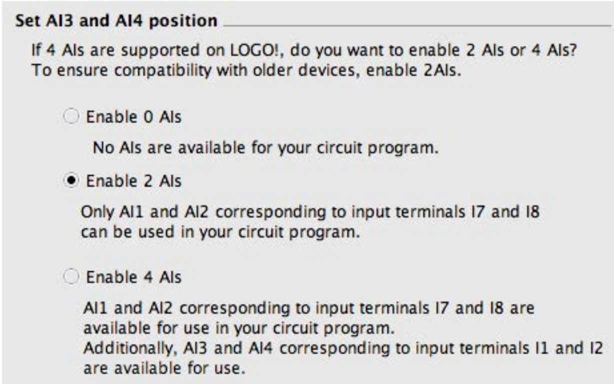
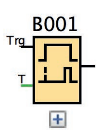
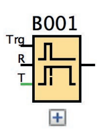
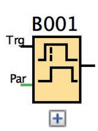
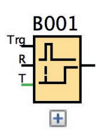
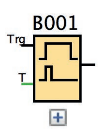
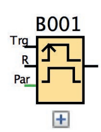


Fig. 4.4 Settings for analog inputs

4.3 Time delays

Many circuits require time delays, for example, in order to allow a pump continue to run, to activate a conveyor belt just for a specific amount of time, to switch off a light in the staircase with a delay, to preheat a heat source, etc. Table 4.1 should give you an overview of the most important time-delayed blocks. In the annex of Chapter 7.2, there is a detailed display of the time delays with associated switching states and descriptions.

Table 4.1 List of various time delay blocks

ON delay	OFF delay	ON/OFF delay	Latching ON delay	Interval relay/ Pulse output	Interval relay, edge-triggered
Time-delayed switching on	Time-delayed switching off	Time-delayed switching on/off	Saves state 1 at the input after a specific time has elapsed	Activates the output after a specific time has elapsed even if status 1 remains the same at the input	Switches on the output if there is a pulse at the input after a specific time has elapsed and switches it off after a specific time has elapsed
					

Example of pump control

You want to build an automatic watering system for your garden, but the water pump that is connected to the rain barrel is so powerful that it pumps too much water onto the flower bed in five minutes. A time delay can be used to prevent flooding. The program for a water pump control system might look like the one shown in Fig. 4.5.

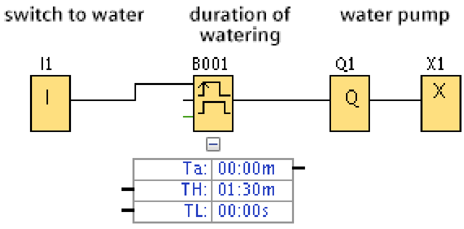


Fig. 4.5 Program for pump control

Once I1 is pressed, the time TH of the edge-triggered interval relay B001 elapses and its output is activated. Water pump Q1 runs. The bed is watered until TH has elapsed. Table 4.2. shows the suitable function table.

Table 4.2 Function table for water pump control

Block name	Property	Function in the program
I1	Digital input	Switches on the pump
B001	Interval relay, edge-triggered	Restriction of the pouring time to X min (TH)
Q1	Output	Water pump
X1	Open terminal	No open connections

4.4 Comparator circuit

In digital technology, it is often necessary to compare values to one another. This is possible both for analog values and for digital values. The simplest form of such a comparison is the 1-bit comparator circuit, which compares one value to another value. There are always only three possible results:

- Value A is greater than value B (A > B)
- Value A is less than value B (A < B) or
- Value A is equal to value B (A = B).

In the case of digital signals, the 1-bit comparison is possible with the AND block without problem. Since the inputs can only accept the values 1 and 0, there is only one possibility for the possibly unequal values. This can easily be shown by negating inputs. The example shown here shows both the possibility of direct negation at the AND block B003 Input 4, and via a NOT block B004 (Fig 4.6).

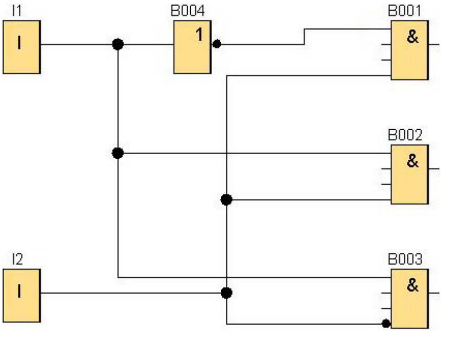


Fig. 4.6 Digital comparator circuit for passing on the signal

Following a similar pattern, analog comparator circuits can also be implemented (Fig. 4.7). Analog threshold switches are used for this. Since these analog inputs do not serve as a permissible signal source, an analog amplifier must be connected in between. The threshold switches are parameterized in such a way that an analog signal is present at Ax and the other analog signal is assessed as both an ON and OFF value. In this way, either one threshold switch or the other will display a positive edge in a relative comparison. Only if both analog values are identical will none of the threshold switches have a positive edge. In this case, the AND block that is negated at both active inputs puts out a positive edge. In this example, the bit memories serve to clearly illustrate the three possible results of the comparator circuit. Due to the use of the analog amplifier, the comparison of the values may require prior computed values. Thus, for example, it can be checked whether a value is twice as large as another by defining the gain of the other value at 2.0. Thanks to "gain", input values can be calculated in a block using multiplication. Both digital and analog comparator circuits can be expanded to more than one bit. However, this would exceed the scope of this chapter. We refer to other literature.

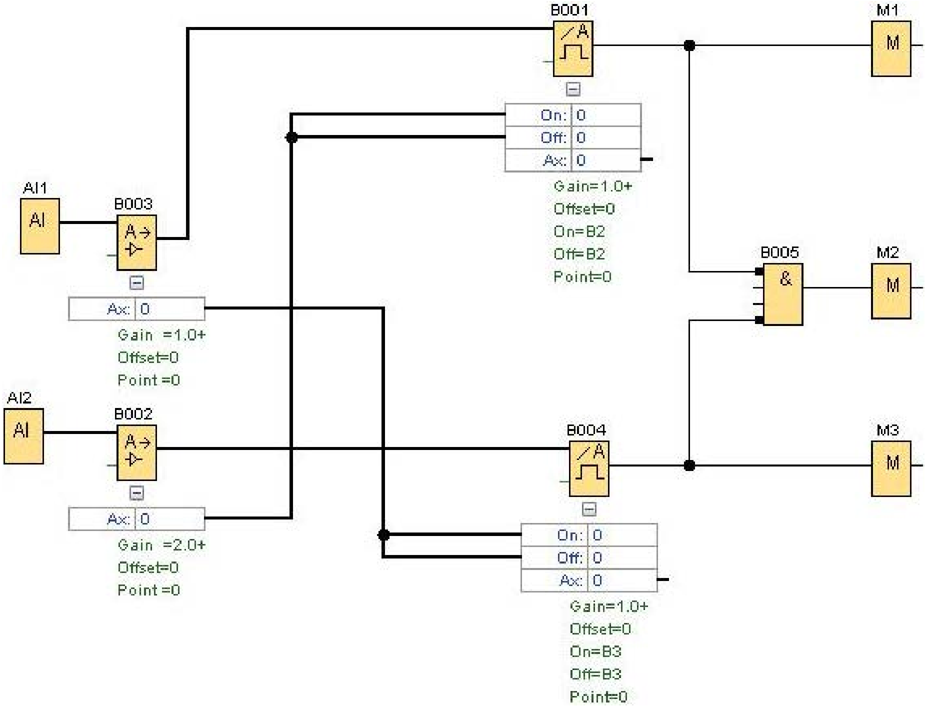


Fig. 4.7 Analog comparator circuit for passing on the signal

4.5 Pulse or frequency measurement

Pulse/frequency detection is used in a variety of circuits. The associated function table is Table 4.3. A simple example is the input of numbers for quantity specifications. For this purpose, an incremental encoder is connected to a digital input of the LOGO!. To generate a pulse in a control project, a timer IC, a reed contact, or the sensor of the scroll wheel of an obsolete computer mouse could be used (Fig. 4.8). Typical applications for frequency measurements might be: anemometers, wind direction indicators, tachometers, or ABS encoders.

Fig. 4.8 Scroll wheel sensor with wheel

Table 4.3 Function table of the pulse/frequency measurement circuit

Block name	Property	Function in the program
I1	Digital input	Pulse/frequency encoder
I2	Digital input	Pushbutton (correction or acknowledgment of the entry)
B001	Message text	Display of the input values
B002	Up/down counter	For recording the number of pulses
B018	ON delay	Allows correction if no pulse during x seconds
B019	Message text	Display of the input values
B020	ON delay	Informational message

In the circuit shown in Fig. 4.9, the counter B002 is positively altered (increased) by the amount corresponding to the number of pulses for each pulse of the pulse/frequency encoder at I1. If pushbutton I2 is also pressed, counter B002 is negatively altered accordingly (decreased). Once I2 is held for longer time and no pulse arrives at I1, the defined countdown of the ON delay B018 is activated. After the defined time has elapsed, counter B002 is reset to zero.

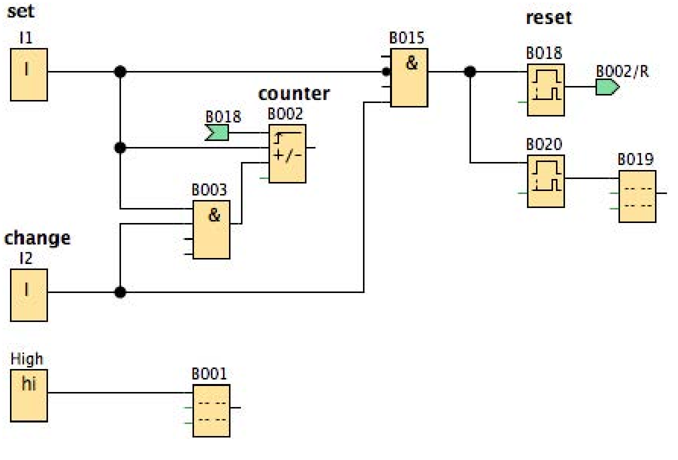


Fig. 4.9 Pulse/frequency measurement circuit

4.6 Information output on one display

One of the most important functions for monitoring and controlling program processes is the message text. It makes it possible to send warnings, information and notes to the user. The message text should thus be seen as supporting the program execution, providing the user with selected information.

The message text on the display of the basic LOGO! unit can encompass up to 96 characters (16 columns × 6 lines) and on the LOGO! TD it can encompass up to 120 characters (20 columns × 6 lines). You can start the parameterization and editing of the text by double-clicking on the message text block. The text is entered via the keyboard after the text box has been selected.

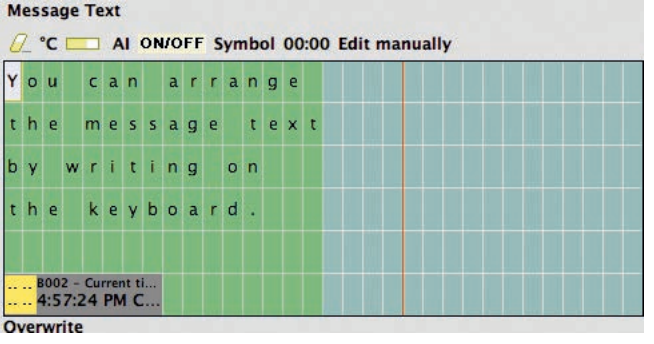


Fig. 4.10 Message text input

Various parameters, such as time output or switching states of blocks, can be placed in the message text directly from the parameter list via drag & drop. Different colors show which area is being displayed: The green area is exclusively for the display of the basic LOGO! unit, whereas the area for the display on the LOGO! TD extends to the red line (see Fig. 4.10). Everything that is placed behind the red line on the exclusively blue field is not displayed at first. To display these characters or to activate special functions for the message text, ticker settings must be set as shown in Fig. 4.11.

4.6.1 Ticker function and colored background lighting

The ticker function allows selected lines to be highlighted. The text to the right of the red line can be displayed flashing or flying into the display by activating the ticker function (line by line or character by character) and selecting the specific line.

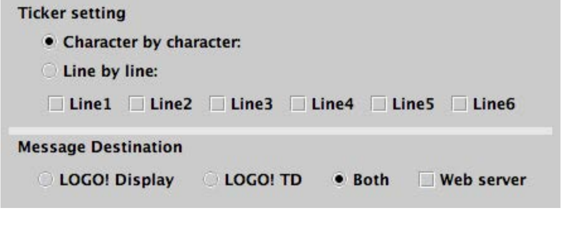


Fig. 4.11 Ticker settings

The ticker time settings, i.e. the frequency of the flashing or the speed of the appearance of the text, are set under *File > Settings for message text... > Ticker time*. Specifying which devices the text is to be displayed on, whether on the LOGO! display, on the LOGO! TD, or on both, is done by setting the message destination.

The variable background lighting is an innovative bonus feature. In addition to the default colors green or blue of the output device, the colors white (M25/26), yellow (M28/30) and red (M29/31) are also available. A color is activated by attaching a bit memory block to the message text (see Fig. 4.12).

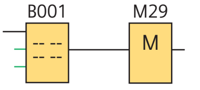


Fig. 4.12 Programming the background color (red)

In addition, the message text provides an acknowledgment function (select by placing a checkmark for "Acknowledge message"), e.g. for a warning message, so that the reading and heeding of the warning message must first be confirmed.

Additionally the priority of a message text can be defined. If two message texts are activated at the same time, the message text with the higher priority (the higher number) is displayed.

4.7 Simplifying complex circuits

Due to their complexity, many programs are so extensive that a user can quickly lose the overview of individual switching sequences. To work more clearly and to better track the individual switching processes, a change to the structure can be helpful. A simple change, such as dividing it into inputs, switches, converters, and outputs, will save you much time and spare your nerves during troubleshooting or a change in the program. Two especially important tools can be found in the software LOGO! Soft Comfort. These play a decisive role in the structuring of the program interface. They will be explained in the following.

4.7.1 Splitting and merging connections

The function of disconnecting connections helps to keep complex connection networks manageable. This function can be found in the tool selection bar in the diagram editor (see Fig. 4.13). When the tool that is marked in red in the figure is selected, the cursor changes and shows an arrow with scissors. If you now click on the connection to be disconnected, only two symbols remain, which are placed on the inputs/outputs of the blocks.

The green arrow in Fig. 4.14 indicates that the connection of the output of the logic block B005 is an outgoing connection. The label on the green arrow shows

Fig. 4.13 The tool selection bar in the diagram editor

where the connection goes: B001/R. Thus, at the destination block, the connection is connected to a “Reset” (B001/R). The green counterpart of the arrow at B005 can be found at the counter block B001 (Fig. 4.15). It is labeled with “B005”, thus indicating the origin of the connection.

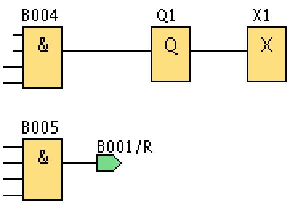


Fig. 4.14 Starting point of the connection

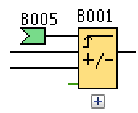


Fig. 4.15 Destination point of the connection

However, if too many blocks are linked, this method will also become unmanageable, as the program becomes increasingly complex. It is recommended that you group important blocks together in a higher-level block. The next section describes how you can design such a block (UDF) on your own.

4.7.2 Creating UDF blocks

The special function UDF (User-Defined Function) is very useful for creating so-called subprograms. Subprograms are like individual chapters that are interconnected. The result is a kind of “book”, a complex program/network with individual “chapters”. The UDFs can be interconnected and each one can be rather comprehensive.

A UDF is created by selecting *File > New > UDP diagram (UDF)* in the tab. A window like the one shown in Fig. 4.16 appears. The desired blocks are inserted into the box bordered by dashed lines. The desired connections to the inputs/outputs then only have to be dragged to the edge of the box. In this way, more complex structures can be grouped in a block.

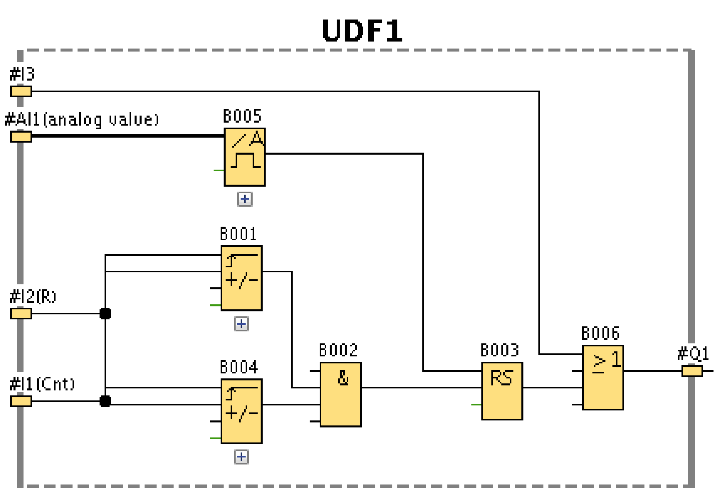


Fig. 4.16 Block selection for the UDF

The UDF block with its created inputs and outputs is saved like a normal project (*File* > *Save as*); the file is called UDF1 and is saved with the extension ".lma". If you would like to integrate the UDF into a project, open an existing/new project and select *Tools* > *Options* in the tab.

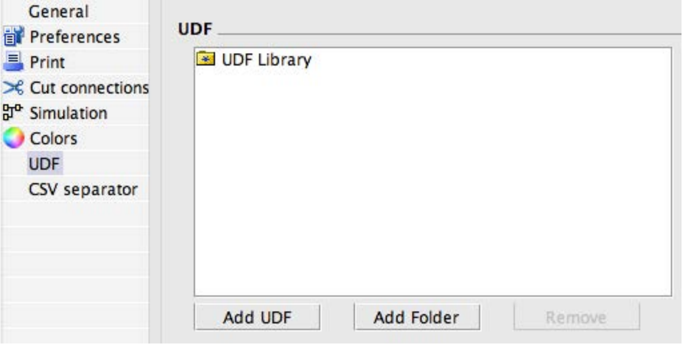


Fig. 4.17 Editing mode for the UDF

The saved UDF now only has to be added to the library. This can be found at the very bottom, under UDF, in the block window from which the blocks are moved into the program (Fig. 4.17). The UDF block is inserted into the program just like all of the other blocks (Fig. 4.18). The inputs and outputs defined in the UDF editing mode can be found at the UDF block. The submenu (the connected counter blocks, the threshold switch, the basic logic functions and the latching relay with their inputs and outputs) is compiled in a block, which gives it structure and makes it significantly more manageable (Fig. 4.19).

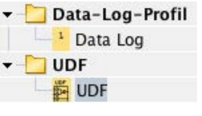


Fig. 4.18 UDF block list

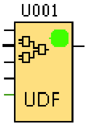


Fig. 4.19 UDF block

As with all other blocks, a UDF can also be connected to the other blocks in the program. Existing UDF blocks can also be installed in other, new UDFs.

5 Planning and carrying out a project in information technology

The LOGO! provides the tinkerer and hobbyist with options for a large number of "easy" to implement projects in a variety of applications. All of the projects have one thing in common: The fact that they can be implemented the quickest, the most effectively, and most cost-effectively if they were planned in careful advance. Answers to goal-oriented questions are helpful in any project start. We are presenting such key questions in Table 5.1.

Table 5.1 Questions regarding sensors and actuators in project planning

Sensors to be used	Actuators to be used
1. Are the sensors digital or analog? 2. How many sensors do I need? Is one basic LOGO! unit enough? 3. What amperage or voltage does the sensor generate in the output – do I have to amplify or reduce the signal? (Chapter 3.2) 4. What frequency does the output of the sensor have – do I have to convert the signal so that it can be recognized by the LOGO!?	1. How many actuators do I need – is one basic LOGO! unit enough? 2. Can the actuators be controlled via a relay or do I need analog outputs – do I need a LOGO! AM2 AQ analog module? (see Chapter 3.7) 3. How high is the voltage – should I use a relay or do I need relays and/or transistors? 4. How high is the current – should I use a relay or do I need relays or transistors? (We would like to point out again that the contents of the book are aimed at tinkerers and amateurs. Voltages above 12 V are life-threatening and must not be worked on by users without corresponding training.)

5.1 Project planning using an example of a fan circuit

We want to clarify this process based on a fan circuit for controlling a fan: The motor is operated with 10 V and 2 A. The fan should always start and run for 5 minutes whenever the room temperature is above 25 °C and a person is standing in front of the fan. In addition, an LED traffic light with a continuous signal is to show that the room temperature

- is under 25 °C: green
- is exactly 25 °C: yellow or
- is higher than 25 °C: red.

The LEDs are operated with 12 V and 10 mA.

1. Defining the sensors and actuators. First, based on the answers from Table 8, the sensors and actuators and the selection of the LOGO! (basic unit or expansion modules) are defined (Table 5.2).

Table 5.2 Sensors and actuators for planning the fan circuit

Sensors to be used	Actuators to be used
5. The temperature sensor is analog and outputs a signal between 0 and 10 V. The motion sensor upstream of the fan is digital and outputs a 5 V edge. 6. Two sensors are needed – one basic LOGO! unit is sufficient. 7. The analog sensor can be connected without any further effort, while the signal of the motion sensor must be amplified to 12 V. 8. The motion sensor has a pre-programmed threshold. If a detected motion exceeds this threshold, the sensor generates a positive edge change. If the motion disappears, the edge drops again. The LOGO! can process this frequency.	1. Four actuators are required: The fan and one red, yellow and green LED each. The four outputs of the LOGO! are sufficient. 2. All of the actuators can be digitally controlled via the relay. 3. All of the actuators run at 12 V and have suitable voltage for direct operation via the LOGO!. 4. The LEDs require 10 mA and are thus clearly under the recommended 1 A continuous load. The fan is above this, however. Therefore, you should fall back on a relay here.

2. Creating the hardware list. The answers to the central questions (Table 8) result in both the hardware list and the capability of naming the LOGO! inputs and outputs in the software. The following hardware is required:

- 12 V fan
- Relay for fan control
- One red, green, yellow 12 V LED each (possibly series resistor)
- Shared power source for fan, LOGO!, LEDs and sensors
- Temperature sensor
- Motion sensor
- 5 V-12 V output amplifier

3. Naming the inputs and outputs. This might seem unnecessary in smaller projects, but it is nevertheless highly advisable. The improved overview facilitates the work – especially in larger projects with many inputs and outputs.

Two inputs are needed for the fan circuit: a digital input I1 on the one hand and an analog input AI1 on the other hand. At this point, we refer you once again to the naming of the analog inputs of the LOGO! (Chapter 3.2). Q1 to Q4 are needed as outputs; Q1 for the fan relay, Q2 to Q4 for the LEDs. In the software, the inputs and outputs are changed in the LOGO! settings in the *Edit/Connection name* menu. The analog inputs are located in the specified list of possible inputs all the way at the bottom (see Figs. 5.1 and 5.2).

4. Function diagram in LOGO! Soft Comfort. First you should give some thought to how much space the function block diagram will require. LOGO! Soft gives you the option of organizing the project into several pages. For complex diagrams with several functional and logical groups, we highly advise you to divide the project into several parts. In this case, this would not be necessary, because the example project is comprised of very few function blocks. Nevertheless, the function of dividing the sheets and the improved overview is demonstrated here. Therefore, the inputs, the fan control, and the temperature traffic light are displayed on different pages. The division takes place via the Page division key in the bar above the function block diagram (Fig. 5.3). Using this, the division of the pages can then be set – in this case 2 × 2 (see Fig. 5.4).

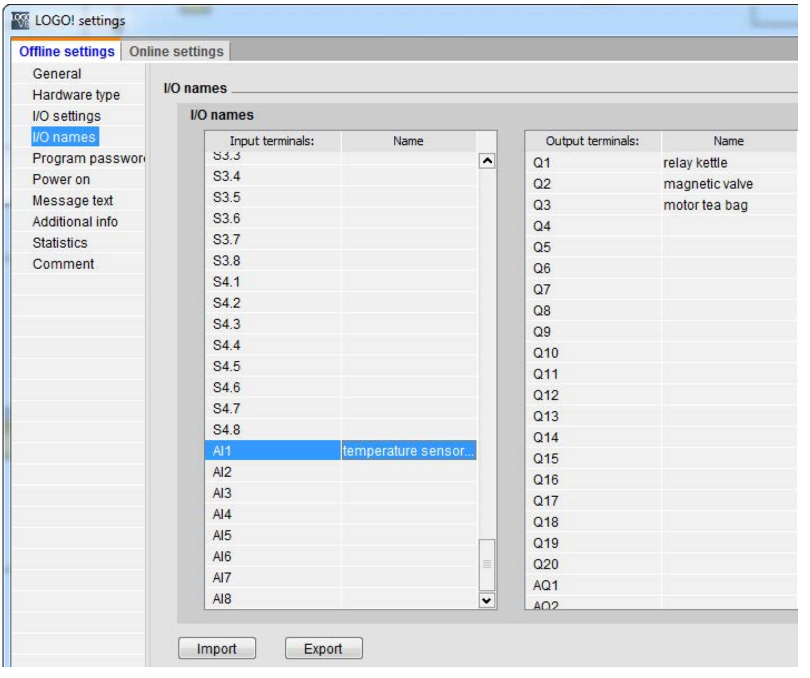


Fig. 5.1 Window for naming the inputs/outputs

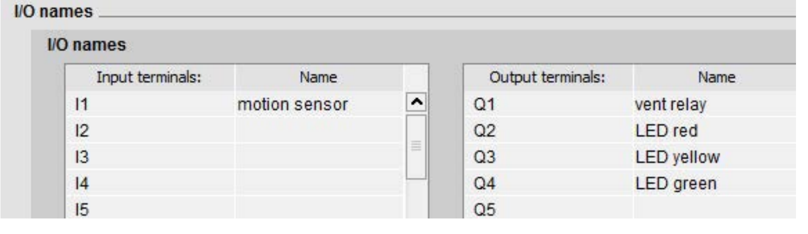


Fig. 5.2 Selecting and naming of analog inputs

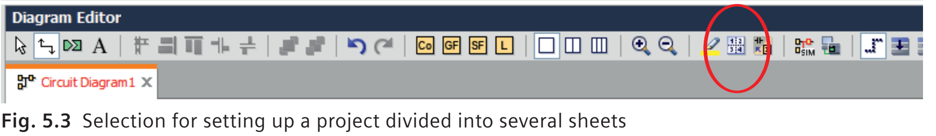


Fig. 5.3 Selection for setting up a project divided into several sheets

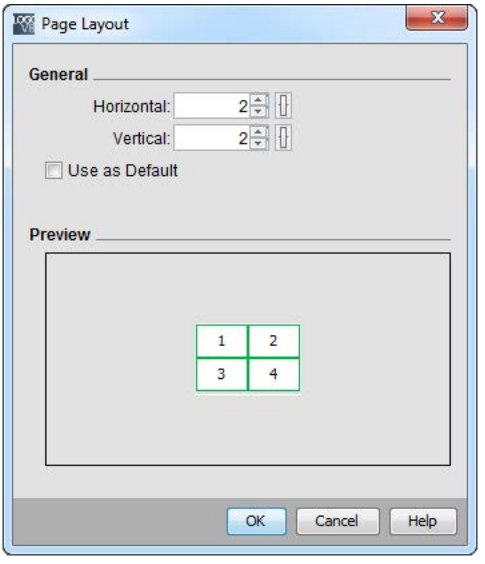


Fig. 5.4 Capability of selecting the arrangement of the project sheets

5. Positioning the inputs and outputs. In our example, we use page 1 for the inputs, page 2 for the fan control, and page 4 for the LED traffic light. Page 3 remains unused.

6. Divide the project into its subprocesses. In the example, four subprocesses can be defined: First, the temperature analysis, second, the motion analysis, third, the fan control and fourth, the LED control. Since the temperature analysis is relevant to both the fan control and the LED control, you should start with it.

The analog input of the LOGO! measures voltage between 0 and 10 V. From the documentation of the temperature sensor, it can be seen in this example that it shows the temperatures in a linear display from -50 °C to 50 °C with 0 to 10 V. This means that a spectrum of 100 °C is covered. If you divide the 10 V by the 100 °C, you get 0.1 V/1 °C.

The following temperatures are relevant to solving the task:

< 25 °C => green LED illuminates,

= 25 °C => yellow LED illuminates,

> 25 °C => red LED illuminates AND fan is operated for 5 minutes.

In Volts, beginning with -50 °C, this means that the LED illuminates at the following values:

- the green LED at values < 7.5 V,
- the yellow LED at 7.5 V and
- the red LED at values > 7.5 V.

When the red LED illuminates, the fan is also operated.

The voltage is output in LOGO! Soft Comfort in values of 1 to 1000, which means that values of < 750, =750 and > 750 are required in the settings.

7. Implementation of the project. The following implementation of the specifications makes no claim to be the best option for the implementation, but is intended to show that there is often a wide variety of equally good solutions for implementation (Fig. 5.5).

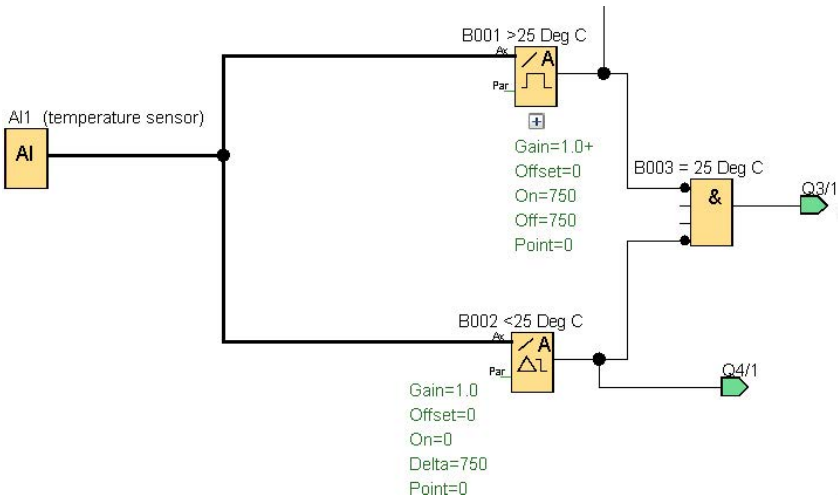


Fig. 5.5 The analysis and further processing of the analog input signal

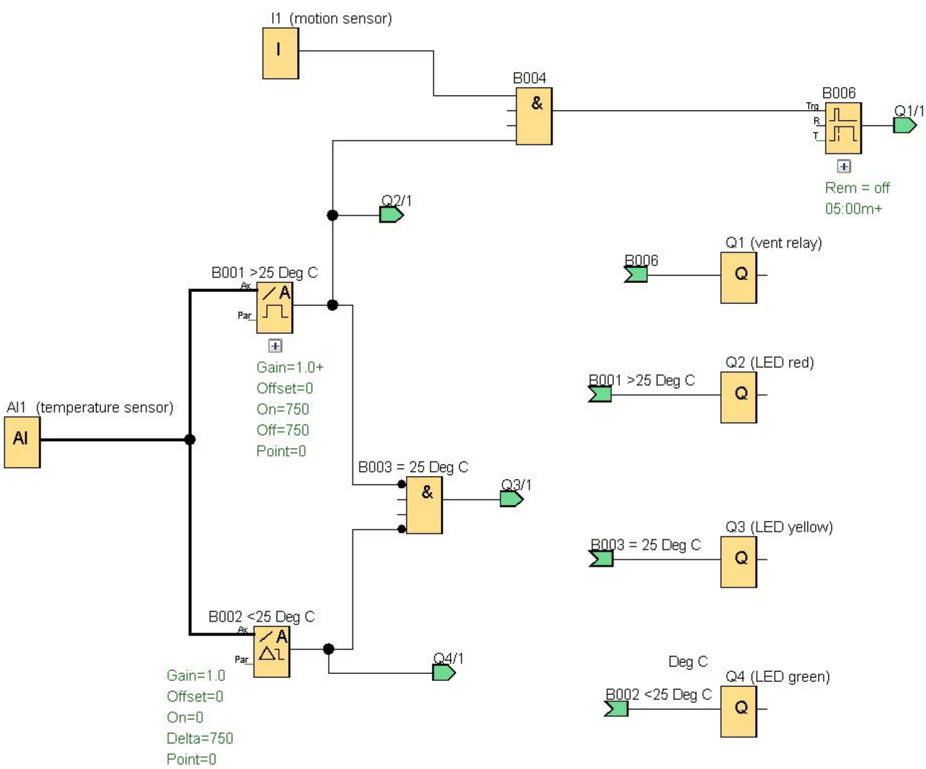


Fig. 5.6 Overview of the finished circuit

The finished project has been compressed onto one page here – the disconnects between the inputs and outputs were left to more clearly illustrate how complex circuits can be divided into several sheets (Fig. 5.6).

8. Simulation. The last step is the simulation of the finished circuit. This is done via the simulation button above the function block diagram and adjusting the parameters under the function block diagram (Fig. 5.7).

At this point, we highly recommend that you read the program help for the simulation. The simulation option is a major advantage of LOGO! Soft Comfort. It provides a number of capabilities for debugging and testing the program before loading it on the hardware. Even if many features seem intuitive, it makes sense to use the exhaustive documentation in the program help to fully utilize the entire range of functions.

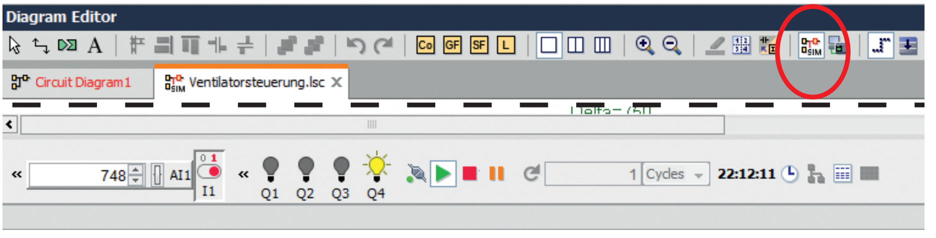


Fig. 5.7 Capability of calling up simulation mode

5.2 Safety notes

Since electric current is not visible or perceptible, everything to do with electricity is a puzzle to many people. Many circuits are based on simple logical associations, however, the structure and wiring of an electrical device is often more simple than expected. Nonetheless, basic safety regulations must be observed when working with electrical circuits. Failure to observe these regulations may result in painful shocks, fire, or even health problems.

Generally speaking, working with 230 V line voltage is reserved for expert technicians. For this reason, all of the circuits presented in this book are implemented using the 12 V version of the LOGO!. Safety precautions must also be observed when handling a transformer or battery.

The following regulations must be observed for all circuits:

- Regularly check the circuits of the electrical devices. In the event that the circuit, housing, or insulation become damaged, have an expert electrician check and repair the parts – also if there are any doubts.
- Ensure that outlets, extension cables, switches, etc., are in perfect condition. Defective plugs and old or damaged cables must be replaced immediately.
- Chafed cables could energize a metal housing or metal tube and cause a short-circuit.

- Avoid winding cables too tightly or bending them. There is a risk that the cable may break.
- Immediately disconnect your circuit from the mains if you smell smoldering. Have it checked by an expert technician, even if it is still functioning.
- Before connecting a circuit to the LOGO!, it must first be disconnected from the mains.
- Wherever possible, a fuse should be installed in the circuit as an active protective measure. Furthermore, there is also the option of protecting individual outlets with mobile or fixed residual current operated circuit breakers or fuses.
- When electrical circuits are installed in the vicinity of water, the power plug must be pulled. The residual current operated circuit breaker should also be used for such an installation.
- Many electrical devices give off heat. Therefore, you must ensure that no flammable objects are located in the vicinity of a circuit that could produce heat. Also ensure there is adequate ventilation. This especially applies to soldering work.
- Especially prevent children from accessing your circuits. Communicate the hazards associated with electrical current.

5.3 A simple soldering course

To connect the LOGO! to individual sensor or actuator solutions, pin boards or even self-made bread boards or strip boards can be used. Plug-in type solutions are suitable for experimental solutions. Bread boards or strip boards are suitable for the regular use of a circuit (Fig. 5.8).

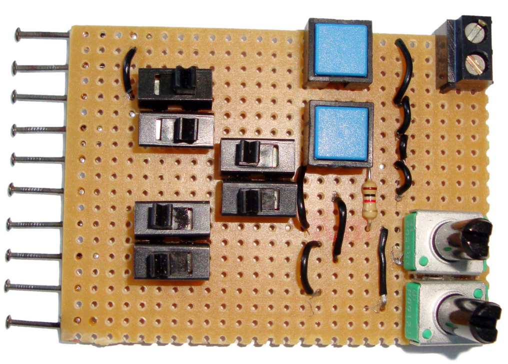


Fig. 5.8 Plan view of the simulation board

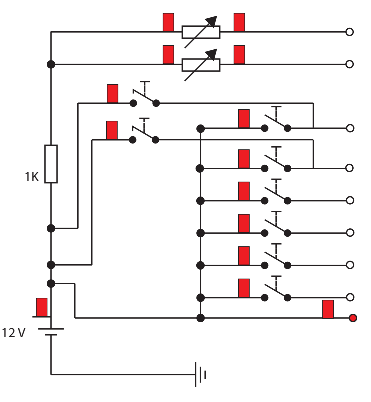


Fig. 5.9 Circuit diagram of the board

The electronic components used in the manufacture of boards are soldered together. The components are plugged into the board for this purpose. There should be a mechanical connection between the two metal parts that are to be connected (securing parts). The connection contacts and the board are heated approx. 1 to 3 seconds using a soldering iron. Then this point is coated with a special metal alloy (solder). The molten solder flows into the columns between the metal parts and bonds them. Usually, a small wisp of smoke will be given off and you will hear some crackling. After the solder has hardened, the soldering point is a brilliant silver and there is an adhesive bond. Excess solder and flux on the soldering tip should immediately be removed using a damp sponge.

5.3.1 Manufacturing a board for the simulation of the LOGO! inputs

In order to be able to easily test the system when planning and implementing a circuit solution, it is possible to make a small simulation board. Both digital and analog inputs can be simulated using the circuit if it is directly connected to the LOGO!. The circuit is also suitable for trainees for practicing how to work with a soldering iron and simple electronic components.

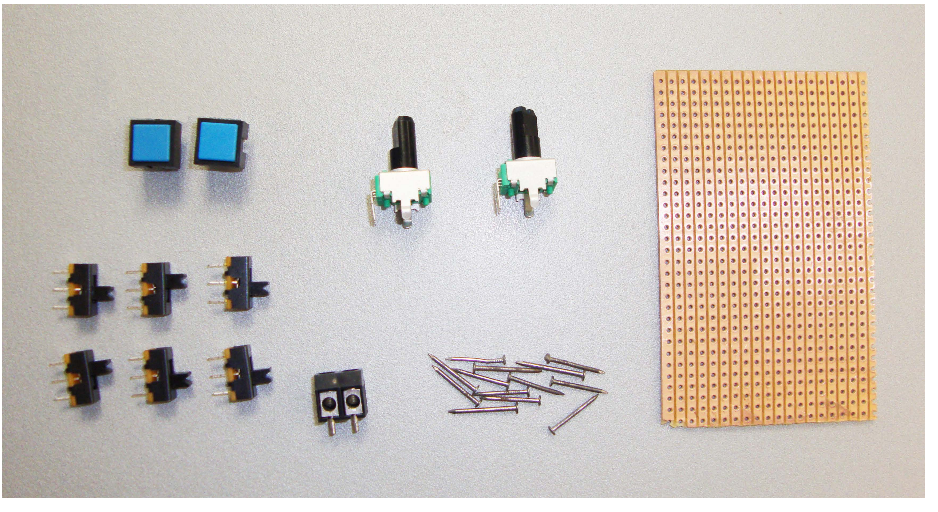


Fig. 5.10 Board components

The circuit consists of six switches and two additional pushbuttons as needed, which can be used to simulate the digital inputs I1 - I6 of the LOGO! (Fig. 5.9). The two analog inputs I7 and I8 of the LOGO! can be controlled using two adjustable resistors. The following components are required for the practice board (Fig. 5.10):

- 1 bread board 160 mm × 100 mm (copper strips, one-side), alternatively one 160 mm × 100 mm perforated board (copper-lined holes, one-side)
- 6 switches (1x switch-over)
- (2 pushbuttons as needed)
- 2 rheostatic resistors 10 kΩ
- 1 resistor 1 kΩ
- 1 connection socket, 2 inputs
- 10 nails

For the circuit shown in Fig. 5.9, a bread board is used. Thin copper strips are affixed to this type of board at a distance of 2.54 mm (grid dimension 1/10 inch) on one or both sides. The paths are pre-perforated. Thus, it does not need to be etched or milled. Proceed as follows (for result, see Fig. 5.11):

1. The ten nails are connected to the board on one side in the direction of the copper paths. The nails serve as contacts of the input switches of the board and the LOGO!. Since the board and LOGO! are structured with the same grid dimension, a nail is soldered onto every second path. Before soldering, ensure that the nails are slightly roughened using sandpaper. The solder will adhere better.

2. Connecting terminal for power supply. On the opposite end of the board, the connecting terminal for the 12 V power supply and the negative pole are soldered onto one side. This is done in such a way that the terminals are connected to the two outer nails directly via the copper paths.

3. The switches are soldered to the board along the copper paths. For changeover switches, it is recommended that you pinch off one of the outer contacts. Not all of the switches are supplied with the right grid dimension, therefore the two remaining contacts might possibly have to be bent open slightly. On the rear of the board, the copper path must be interrupted between the soldered contacts. This is best done using a 4 mm drill in a drill press, a small milling head in a mini-drilling machine, or manually using a countersink or a knife.

4. Power supply for the switches. After fitting the switches, all of the switches should be supplied with power. To do this, the connection of a switch pole to the positive pole (of the outlying copper path) of the 12 V power supply must be established via a wire strand.

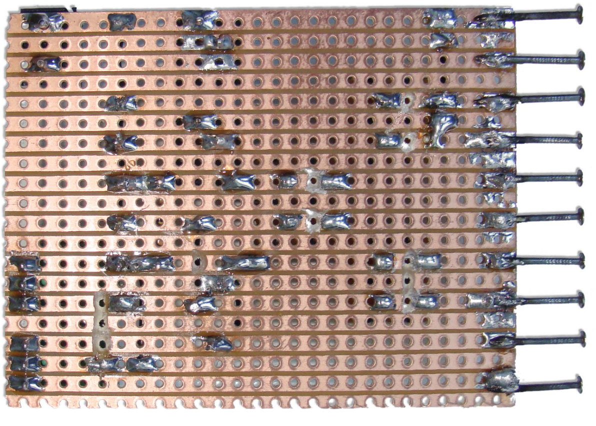


Fig. 5.11 PCB assembly from below

6 Using the LOGO! in day-to-day operations

The following projects are topics taken from day-to-day practice. All of them have been implemented in the presented form or with individual variants based on Siemens LOGO!. Even if you do not want to implement one of the examples, you can at least develop each individual topic as a theoretical example for brain-storming or for self-study. With the aid of the simulation mode or with a practice board, more complex subjects can also be well developed following the motto of “Learning by doing”. In this way, many individually planned automated systems can arise after a practice phase more or less based on the projects described here.

When implementing the applications, we ensured that all other necessary parts, in addition to the required LOGO! components, are easy to procure or can be self-made. During the concrete implementation of a project, the specified program structure is always seen as one option of the implementation. Parts of the different circuits can also be grouped together or complement each other. Each project must be adapted to the user‘s own capabilities, but also to the conditions and the situation on-site.

Furthermore, as already described in Chapter 5.2, we must point out that the safety requirements must always be complied with when implementing a project! Since work may not be carried out on the mains voltage by laymen, all of the projects are designed to allow working with low voltages.

All of the projects are structured according to the same scheme:

1. The idea and the fundamentals for the project are described and illustrated in the topic description in the first section.
2. The contents of the circuit and the blocks used in the program are presented in tabular form in the second part. This gives a quick overview of the topics. When a user is planning his own project, quickly readied circuit components can be found and used for help.
3. In the third part, the structure of the circuit is described, the special procedures for programming individual components are explained, and information is given on the individual adaptation of the project. Depending on the scope of the topic, individual parts are described separately or the contents are based on each other with an increasing degree of complexity.
4. For a better overview, all of the blocks used are named and their properties and function in the program are described in an additional table. The respective property of a block also always corresponds to the name in the program window.
5. Finally, tips and tricks for implementing the project are provided.

6.1 A weather station

The LOGO! is very multi-faceted. Why buy an expensive weather station, for example, when you can observe and record the weather using the LOGO!? The following example should give you ideas on how to convert the LOGO! into a weather station with data analysis and weather information.

A weather station is comprised, for example, of an anemometer, a thermometer, and a rain gage (Fig. 6.1). First, the individual components with their associated programs are explained. Then all of the subsections are merged into an overall circuit (see Table 6.1).

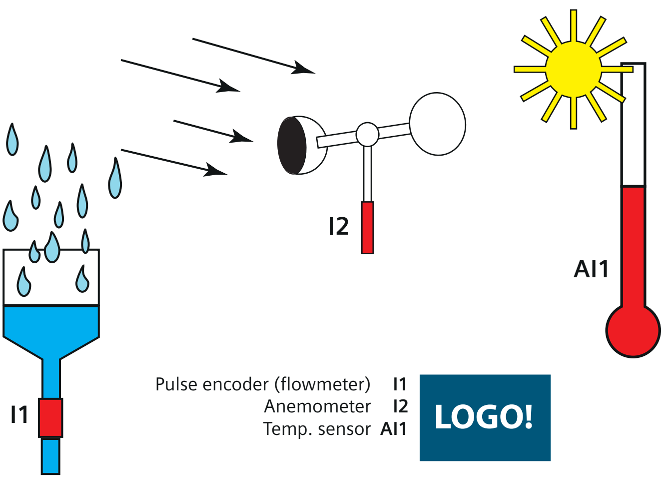


Fig. 6.1 Functions of a weather station

Table 6.1 Contents and functions of the "Weather station" project

Content	Blocks used in the program
✓ Arithmetic instruction ✓ Pulse recording ✓ Message text output ✓ Bar diagram in the message text ✓ Analysis of a temperature sensor (e.g. PT 100) ✓ Display of the temperature by means of message text	✓ Analog input ✓ Digital input ✓ Arithmetic instruction ✓ ON delay ✓ Latching relay ✓ Up/down counter ✓ Analog threshold switch ✓ Message text ✓ Bit memory ✓ Analog value monitoring ✓ Open terminals

A counter module forms the basic block for the anemometer in order to determine the force of the wind and to display the pulses of the wind wheel with a bar diagram.

6.1.1 Anemometer subprogram

Fig. 6.2 shows the program for the anemometer: A pushbutton is connected to I1, which records each rotation of the wind wheel with a pulse. The counter B001 counts each pulse, i.e. each rotation of the wind wheel. Once the wind wheel is rotating, an ON delay is activated by the latching relay B014, which resets the counter block B001 every x minutes. Thus the measured value is always as current as the time interval allows. To ensure that the ON delay is only activated when the wind wheel rotates, a loop is integrated in the program (comprised of the latching relay B014, the ON delay B013, and the bit memory block M1). The bit memory is needed to ensure that a recursion, which is needed for the loop, is possible.

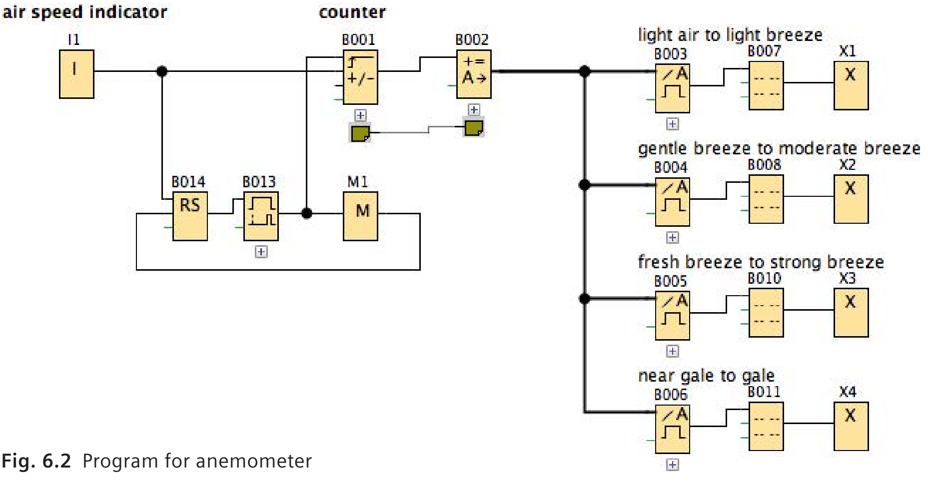


Fig. 6.2 Program for anemometer

The arithmetic instruction B002 divides the pulses of B001 by x min in order to calculate a frequency. For example, for 1 minute of measured time, all of the pulses within that one minute are divided by 60 seconds. Analog threshold switches (B003 – B006) are coupled to the arithmetic instruction. Beyond a specific frequency, these activate the respective message text coupled to them (see Figs. 6.3 and 6.4).

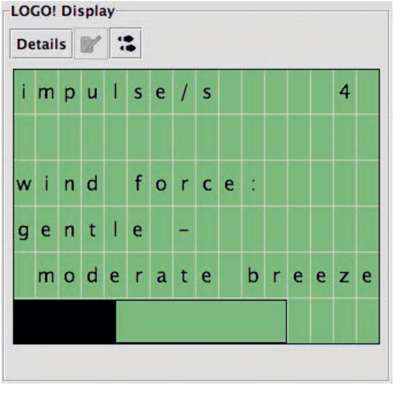


Fig. 6.3 Pulse and wind display

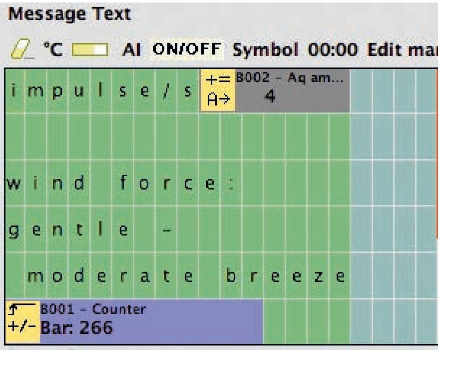


Fig. 6.4 Wind display settings

Depending on the size of the wind wheel, the arithmetic instruction must be adapted. You can read about the wind forces in the respective specialist literature[8]. The wind forces used in the example are fictitious. The measured values can be compared to an anemometer[9] and be adapted to the program.

6.1.2 Temperature measurement subprogram

Every weather station must also have a thermometer. The sensor needed for this is commercially available and favorably priced.[10] A sensor from a disused outside thermometer can also be used. The sensor can be connected directly to an analog input of the LOGO!.

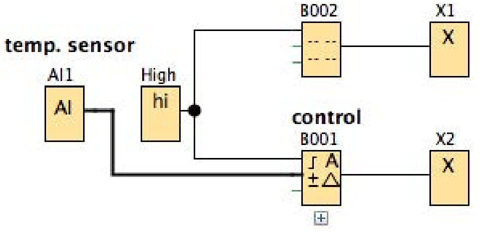


Fig. 6.5 Program for temperature measurement

The program for temperature analysis looks like the following (Fig. 6.5): The temperature sensor is connected to the analog input AI1 of the LOGO!. Its value can be read directly into the analog value monitoring (B001). In the settings of the analog value monitoring (B001), in the *category Sensor*, choose > *Sensor: PT100/1000*. The analysis can be done in °C or °F. To display the temperature directly on the display of the LOGO! or an external device (TD), we have integrated a message text into the program which displays the analog value, in °C in this example (Fig. 6.6).

The message text is programmed (right) in such a way that it always displays the current analog value (left). Depending on the sensitivity (and the purchase price) of the temperature sensor, different value ranges can be measured.

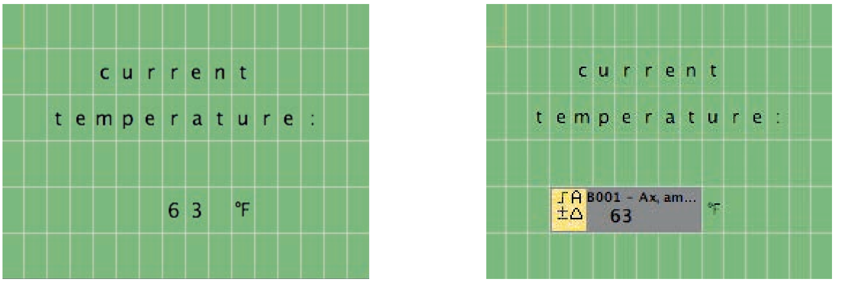


Fig. 6.6 Temperature display and message text setting

8 www.wikipedia.org/wiki/Beaufort_scale
9 e.g. from conrad.com, order number 123204
10 e.g. from conrad.com, order number 172430

6.1.3 Precipitation measurement subprogram

A flow meter should help to measure the precipitation in a specific period of time[11]. Thanks to the turbine measuring principle, the flow meter outputs 2,500 to 8,500 pulses per liter, depending on the size of the inserted nozzle (according to datasheet of the flow meter used). This means that if one liter of water has flowed through, 2,500 to 8,500 pulses have been output. The program measures the flow at specific time intervals and calculates the amount of water in milliliters (Fig. 6.7).

The flow meter is connected to I1 and provides pulses to the up/down counter B002. Once it rains, i.e. I1 outputs a pulse, the latching relay (B003) and thus also the ON delay (B004) are activated. The time period in which the precipita-

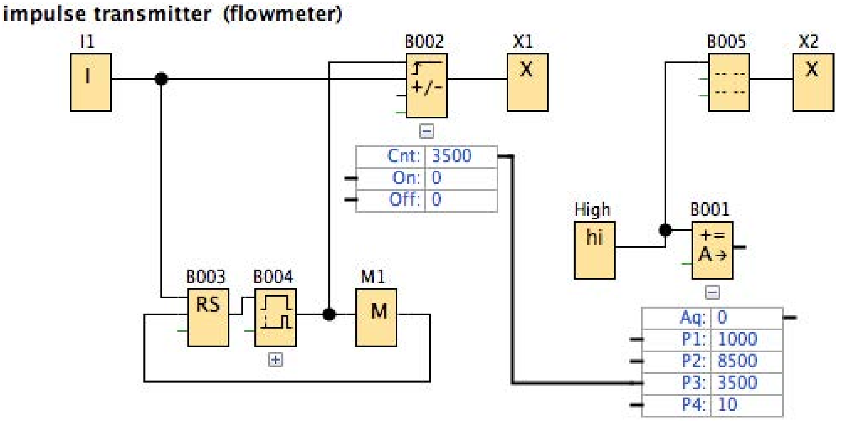


Fig. 6.7 Program for precipitation measurement

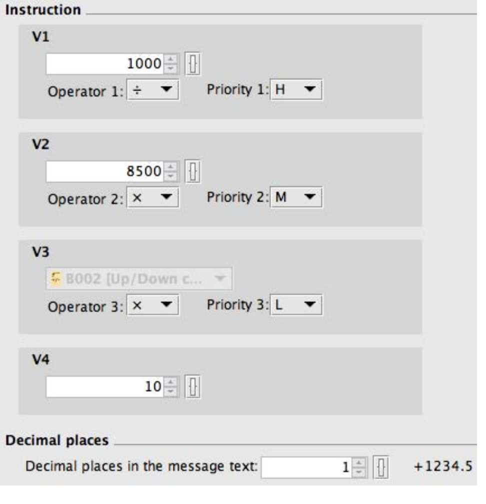


Fig. 6.8 Arithmetic instruction for calculating the amount of precipitation

[11] e.g. from conrad.com, order number: 150392

tion is to be measured can be set at block B004. The bit memory M1 allows a recursion, which causes a reset of the counter (B002) and the ON delay (B003). For one liter of water flow in the defined time interval *X* (ON delay B004), the up/down counter (B002) counts 8,500 pulses in our example. Using the rule of three, we get the following calculation:

The precipitation values are specified in ml/m^2, which is why the calculation for the arithmetic instruction (B001) looks like in Fig. 6.8 in order to get as accurate a result in ml as possible.

The counter pulse is integrated into the arithmetic instruction (B001) via a pointer by B002 (see V3 in Fig. 6.8). To display a result in ml and with one decimal place, the end product must be multiplied by 10 and the number of decimal places must be set (see *V4* and *decimal places* in Fig. 6.8). A message text (B005) is integrated in order to be able to read the amount of water that has flowed through on a display (Fig. 6.9). This is programmed in such a way that it displays the result of the arithmetic instruction (B001). The amount of rain is thus displayed as the end product of the arithmetic instruction (B001) on the message text (B005).

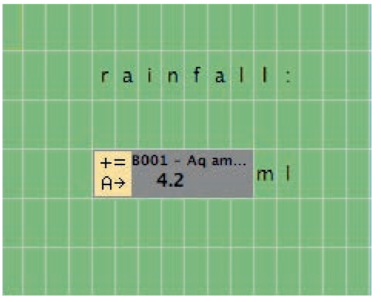


Fig. 6.9 Message text for amount of precipitation

6.1.4 Overall program for weather station

Now, all of the subprograms are merged into one large program using copy and paste. The block numbers change as a result of the merging of the subprograms (Fig. 6.11).

The message texts of the subprograms are deleted. To ensure that the current weather conditions (temperature, rain and wind force) are always displayed, the required information from the individual blocks is inserted into the message texts B015 to B018 from the subprograms, so that a display is always visible like in Fig. 6.10.

Values from B007 in overall program ➔

Values from B001 in overall program ➔

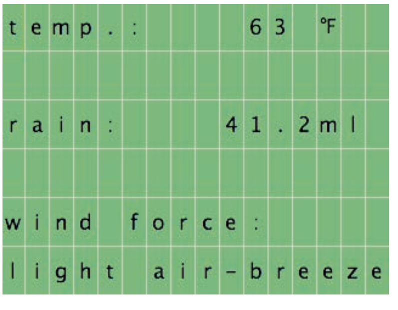


Fig. 6.10 Message text for overall program of the weather station

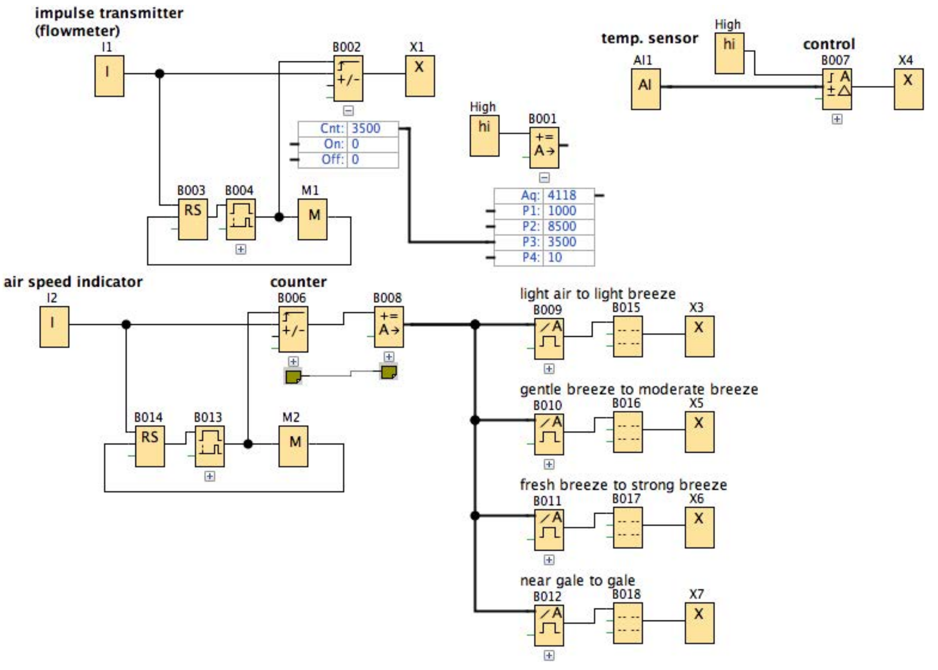


Fig. 6.11 Overall circuit for the weather station

Table 6.2 Blocks, properties and functions of the "Weather station" project

Block name	Property	Function in the program
AI1	Analog input	Temperature sensor values
I1	Digital input	Connection of the flow meter (pulse input)
I2	Digital input	Pulse recording
B001	Arithmetic instruction	Analysis of the pulses of the flow meter, calculation of the amount of rain which flows through the flow meter
B002	Up/down counter	Recording the pulses of the flow meter
B003	Latching relay	Activates the ON delay (B004)
B004	ON delay	Allows measurements in a defined period of time
B006	Up/down counter	Pulse counter
B007	Analog value monitoring	Analysis of the temperature values of the sensor
B008	Arithmetic instruction	Frequency calculation
B009 – B012	Analog threshold switch	Activating the respective wind display
B013	ON delay	Value update
B014	Latching relay	Loop activation for value update
B015 – B018	Message text	Overall weather display
M1, M2	Bit memory	Allows recursions/loops
High	High	Sets the connected blocks to state 1
X1 – X7	Open terminal	No open connections

Tip

To check the system for functional safety, all of the data gathered by the weather station should be compared using calibrated devices or meteorological values from the Internet.

The size of the inserted nozzle must be observed when using the flow meter. Corresponding to this, 2,500/8,500 pulses must be used in the calculation for the arithmetic instruction. The calculation might have to be adapted accordingly.

6.2 An illumination system

An intelligent illumination system with various switching options is generally very expensive to procure. With a system controlled by the LOGO!, it is possible to implement a control system with four different modes (Fig. 6.12). In addition, the user can switch back and forth between four different modes using two switches in a self-programmed menu. The LOGO! 12/24 V RCE is equipped with relay outputs, which unfortunately makes continuous dimming of the output power impossible.

The following program should help to allow different illumination scenes even without a continuous dim function. Various lamps or preset dimmers could be connected to the outputs Q1 to Q4.

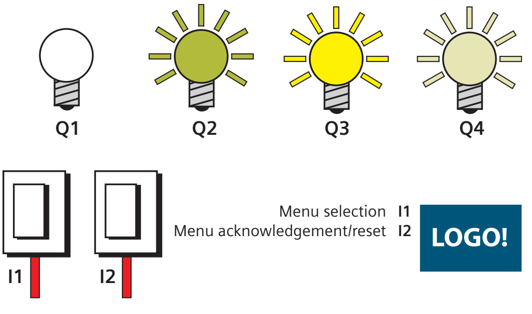


Fig. 6.12 Scenarios for the "Illumination system" project

Table 6.3 Contents and functions of the "Illumination system" project

Content	Blocks used in the program
✓ Programming your own selection menu (display on the LOGO! display) ✓ Loops in a menu ✓ Starting and ending programs via a menu selection ✓ Message text	✓ Digital input ✓ Digital outputs ✓ Up/down counter ✓ Message text ✓ Basic function AND ✓ Latching relay ✓ Arithmetic instruction ✓ Analog threshold switch ✓ ON delay ✓ High

There is a large selection of lamps: From halogen lamps and LED lighting to simple bulbs, which can be connected to the outputs of the LOGO! in order to achieve the desired ambiance.

You can decide on atmospheric lighting according to your needs, with discreet light for a relaxed atmosphere or bright light for working, for example. It would also be possible to connect dimmers that are set to a specific brightness to the outputs and which are connected upstream of the conventional room lights in the circuit. A handheld radio transmitter with an associated receiver module[12] can be connected to the inputs of the LOGO! as an extra feature for controlling the different lighting modes. Radio transmitters that are suitable for this generally have one or more relay outputs on the receiver module, which, in turn, can be connected to the inputs of the LOGO!.

The up/down counters B001 to B004 are set by I1 and activate the respective output (B001 for 0, B002 for 1, etc.) for counting states 0, 1, 2 and 3 and activate the assigned message texts B005 to B008. To ensure that the menu (the successive message texts) begins again from the beginning (B001/B005) after B004/B008, an arithmetic instruction (B009 with reference to counter B004) is necessary with an analog threshold switch (B011). To do this, configure the following settings:

- Sensor: no sensor,
- Measuring range: 0 - 1000,
- Threshold value In: 3, Out: 0

It is set up in such a way that all of the counters are reset by activating the output of B011 when the counting value/state 3 is exceeded. Thus, a menu loop is created, which allows jumping back and forth between the individual scenes.

Briefly pressing once on switch I2 confirms the illumination scene which is displayed on the display and activates the respective latching relay (B016 to B019) and thus the associated output (Q1 to Q4).

If the selected and active illumination scene is to be ended, only the pushbutton at I2 must be held for three seconds (ON delay B020). Now the latching relays are reset.

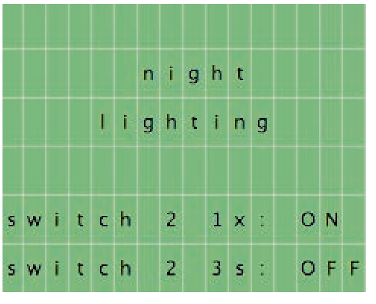


Fig. 6.13 Display, selection menu

In order to change the illumination scene, the user can switch to a different illumination scene while the currently selected illumination scene is active by pressing pushbutton I1 and he can reset the scenes by holding pushbutton I2 for three seconds. Then he confirms this by briefly pressing once on I2 and the selected scene is activated. The basic logic function AND (B012 to B015) prevents several scenes from being activated at the same time. Thus, no other scene can be simultaneously activated wile an illumination scene is being activated. The message text for a scene might look like the one shown in Fig. 6.13.

[12] e.g. from Pollin, order number 550 639

As an extra setting, the 5th and 6th message text line can be a ticker (see Chapter 4.6.1).

Of course, the program can also be used to control four different actuators, which can be easily selected with the aid of the menu selection. Fig. 6.14 shows the program for the illumination control.

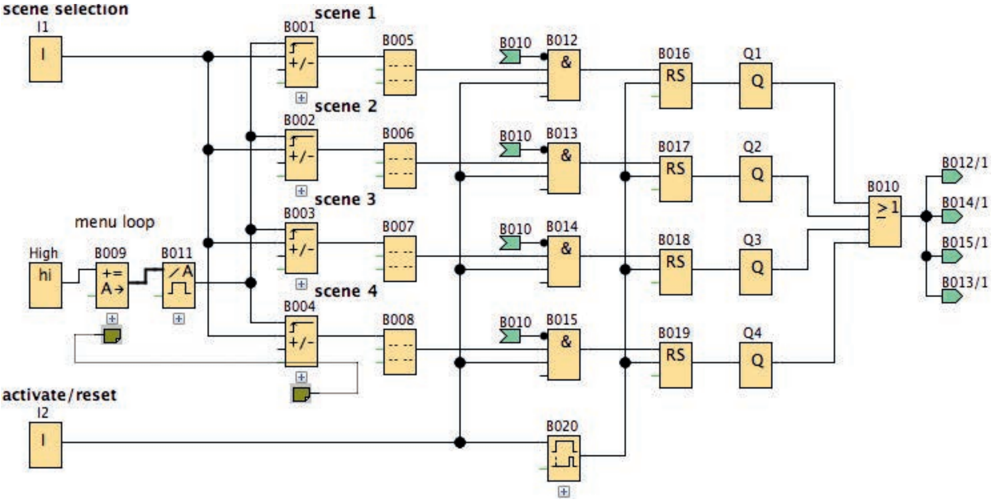


Fig. 6.14 Program for illumination system

Table 6.4 Function table of the “Illumination system” project

Block name	Property	Function in the program
I1	Digital input	Selection between 4 menu items
I2	Digital input	Selecting/clearing an illumination scene
Q1 – Q4	Output	Outputs for various lamps/dimmers
B001 – B004	Up/down counter	Illumination scene selection 1/2/3/4
B005 – B008	Message texts	Display menu: Illumination scene 1/2/3/4
B009	Arithmetic instruction	Control and checking of the menu
B010	Logic basic function OR	Only individual switching of the outputs
B011	Analog threshold switch	Loop function in the menu
B012 – B015	Logic basic function AND	Only individual switching of the outputs
B016 – B019	Latching relay	Switching of outputs Q1 to Q4
B020	ON delay	Reset/switching through 3s activation
High	Status 1 (high)	Activating the loop function in the menu

Tip

Thanks to a well-structured program, more complex elements like loops can easily be implemented. A clear structure helps you to keep the overview.

When controlling with a radio transmitter, the pushbuttons, which are connected to the inputs I1 and I2, can be replaced by the relay outputs of the receiver module.

6.3 A plant station for the window sill

A fully functioning herb or plant box on the outside window sill with fresh herbs that you cultivated yourself – this is the dream for any kitchen. Unfortunately, we often forget to water the plants or we are away from home when the plants start to dry out in the sun. Using simple means, a self-maintaining plant box can be set up with water tank monitoring (float switch), withering protection (via a humidity sensor), drying up protection (via an automatic blind control), and a regular, time-controlled watering function (Fig. 6.15).

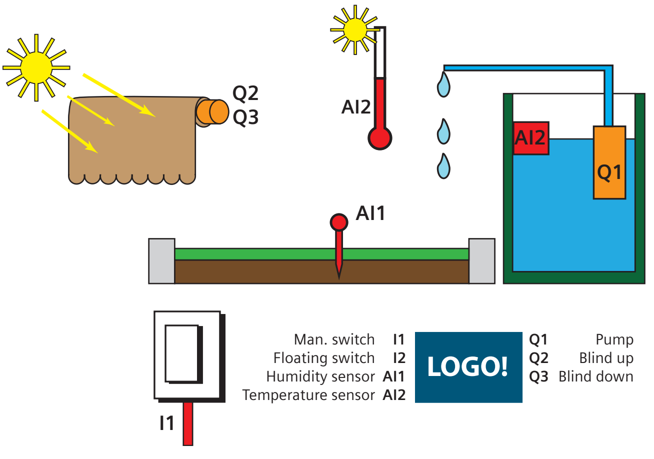


Fig. 6.15 Automating a plant station

Table 6.5 Contents and functions of the "Mini-green house" project

Content	Blocks used in the program
✓ Time-controlled automation ✓ Water level check ✓ Soil humidity recording ✓ Temperature check ✓ Intelligent watering	✓ Digital input and output ✓ Analog input ✓ Week time switch ✓ Analog threshold switch ✓ Basic functions/logic gate: AND, OR, NOT ✓ Retentive ON delay ✓ ON and OFF delay ✓ Latching relay ✓ Bit memory

You can set fixed watering times (B001) on the week timer, alternatively you can do the watering manually (I1). Since the week timer can only be activated for a minimum of 1 minute, the circuit needs an OFF delay (B004), which allows a watering time under 1 minute. In this way, you can individually set how long the watering is to take place (depending on the pumping capacity of the water pump and the desired amount of water).

A water tank could be installed under the plant box and its level could be monitored by a floating switch (I2). Humidity sensor(s) (AI1) in the soil around the plants, which are coupled to threshold switches (B009 and B013), prevent drying out and excess watering of the plants. When a defined value is undershot/exceeded over a long period of time, the analog threshold switches activate.

In addition, a blind (Q2/Q3) can also be installed, which should protect the plants from excessive sunlight. In the event of too much sunlight, the output of the threshold switch (B005) is activated, which moves the blind downward (Q2). In the event of too little sunlight, the output of the threshold switch (B005) is inactive, which moves the blind upward (Q3). Fig. 6.16 shows the program for such a plant station.

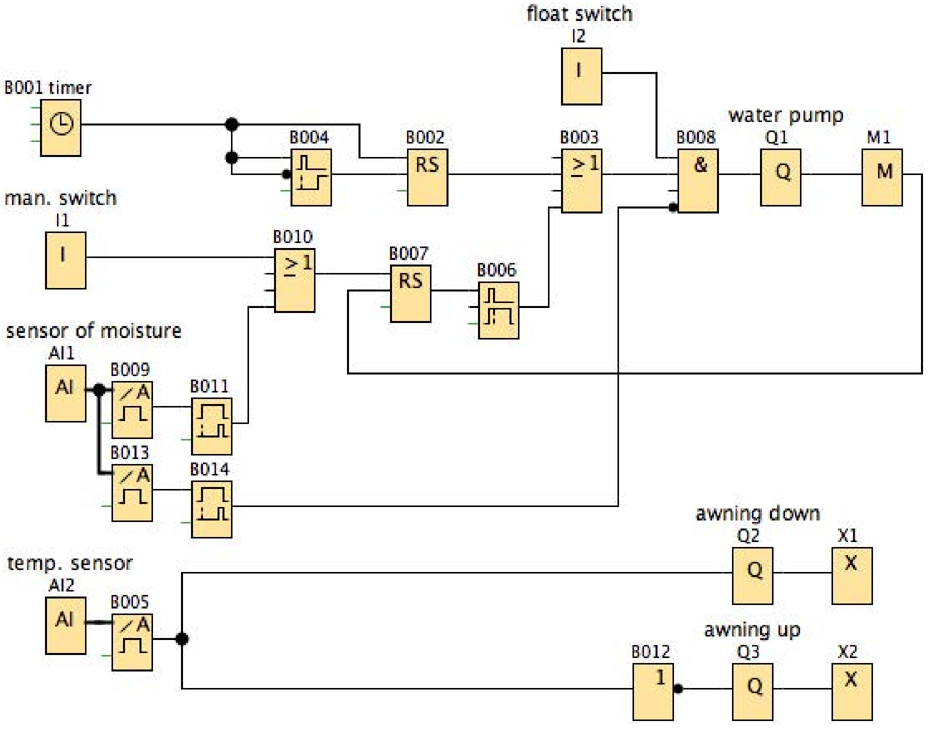


Fig. 6.16 Program for a plant station

Table 6.6 Function table of the “Growing station” project

Block name	Property	Function in the program
I1	Digital input	Manual switching
I2	Digital input	Floating switch (water tank)
AI1	Analog input	Humidity sensor (soil humidity)
AI2	Analog input	Temperature sensor
Q1	Output	Water pump
Q2, Q3	Output	Blinds down/up (shading system)
X1, X2	Open terminals	
M1	Bit memory	Allows recursions/loops
B001	Week time switch	Timer clock with defined watering times

Block name	Property	Function in the program
B002	Latching relay	The output remains activated until the defined time of B004 has elapsed
B003	Logic basic function OR	Activation by timer clock or manually
B004	Retentive ON delay	Allows brief watering (under 1 min)
B005	Analog threshold switch	Controls the shading system (e.g. blinds)
B006	OFF delay	Allows watering for a defined period of time by means of manual switching
B007	Latching relay	Allows one-time pressing of I1 (manual switching)
B008	Logic basic function AND	No pump function when water tank is empty
B009	Analog threshold switch	Initiates watering when too dry
B010	Logic basic function	Switching manually or when too dry
B011	ON delay	Only activates in event of prolonged dry status
B012	Logic basic function NOT	Controls blinds when temperature is too low
B013	Analog threshold switch	Pump will not run if soil humidity is too high
B014	ON delay	Only activates in event of prolonged humid status

Tip

The watering times should be set according to the pumping capacity of the water pump and the desired amount of water.

Instead of the blind, a heat lamp can be installed and connected to Q3 for areas with little light.

12 V DC model pumps, which can be found in various electronics shops, can be installed as water pumps. Of course, the user must observe the amount to be conveyed and the maximum conveying height (depending on what the height difference is between the water tank and the window sill).

The sensitivity of the humidity sensor and the temperature sensor can be set more sensitively by connecting a potentiometer or a variable resistor upstream (cf. Fig. 6.17).

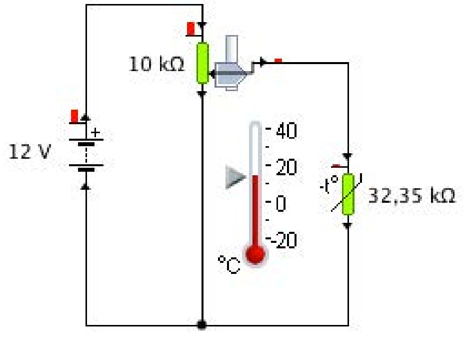


Fig. 6.17 Connecting a potentiometer to regulate the sensitivity of the sensors

6.4 A tea brewer

The last few years have seen a wide variety of innovations for the home brewing of fresh coffee. Modern coffee machines take over the regulating of the brewing time, the temperature of the water, and the dosing of the right amounts. Tea drinkers, on the other hand, still have to boil water in a tea pot, allow it to cool to the right temperature for the respective tea, pour it onto the tea, and then end the brewing process at the right time. With the aid of the LOGO! and a few commercially available electronic components, it is possible to build your own tea machine, which takes over the individual steps of brewing tea, similar to a coffee machine (Fig. 6.18).

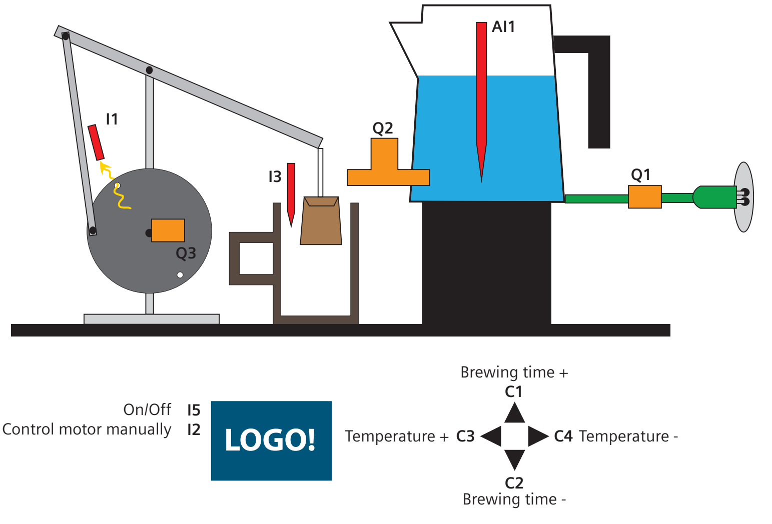


Fig. 6.18 Structure of a "tea machine"

Table 6.7 Contents and functions of the "Tea brewer" project

Content	Blocks used in the program
✓ Controlling a solenoid valve ✓ Controlling a motor ✓ Recording of temperatures ✓ Integration of cursor keys ✓ Integration of relays ✓ Integration of light sensors ✓ Using cursor keys with single entry and quick entry ✓ Integration of the display	✓ Analog threshold switch ✓ Latching relay ✓ AND/OR/High/XOR gate ✓ Negation ✓ ON delay ✓ Incremental encoder ✓ Up/down counter ✓ Message text ✓ Arithmetic instruction ✓ Cursor ✓ Pulse relay ✓ Interval relay/pulse output

The tea machine is comprised of two hardware units, which are controlled via four inter-communicating software groups. The hardware is divided into

"Belonging to the water boiler" and "Belonging to the cup". The software groups are "Set the target temperature", "Set the brewing time", "Implement the time and temperature parameters" and "Tea bag handling".

6.4.1 Water boiler subproject

For amateurs, 12 V devices are affordable and are potentially less dangerous than 230 V devices. However, the load is too great to connect them directly to the LOGO!. Therefore, the 12 V boiler is connected via a relay, which, in turn, is controlled by the LOGO!.

A temperature sensor is needed in the water boiler for measuring the target temperature. In this case, food-safe, tasteless materials should be used for the preparation of foodstuffs. Because tea is prepared with very hot water, it does not make sense to use the usually cheap plastic pumps. It is easier to use a metal solenoid valve. Solenoid valves which withstand boiling water are clearly more affordable and easier to obtain than comparable pumps.

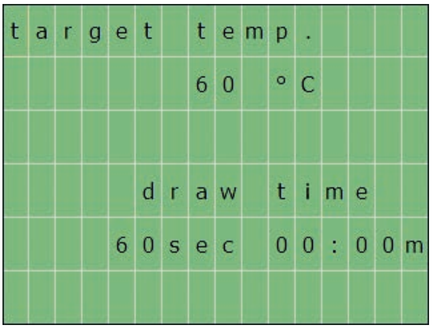


Fig. 6.19 Setting the target temperature and brewing time countdown (60 °C correspond to 140 °F, 0 °C = 32 °F, 100 °C = 212 °F)

If the water has reached the desired target temperature, it must go from the boiler into the cup. If a solenoid valve is built into the housing of the water boiler and is positioned higher than the cup, the hot water can flow into the cup using gravity. Low-cost copper components can be used. A solenoid valve can be controlled directly by the LOGO!. With solenoid valves, a distinction is made between those that are open when activated and close when deactivated and those that are exactly the opposite, i.e. activation causes them to close. In the case of the outlet for the water boiler, the first variant must be used to prevent the contents of the boiler draining out when it is switched off.

6.4.2 Tea cup subproject

The easiest part of this project is the preparation of tea using tea bags. The tea is already portioned. The preparation of the tea does not cause any contamination of the water due to leaking sieves, and the used tea bag can be easily disposed of. The automatic preparation requires a mechanism which removes the tea bag from the cup at the end of the parameterized brewing time. Furthermore, the amount of water must be adaptable to the amount of tea in the tea bag.

A rocker arm, which can be raised and lowered over the cup by a motor, could be used for removing the tea bag. To prevent unnecessary stress on the motor, the redirection can also be implemented via a self-locking worm drive. Only a sensor is needed to detect the fill level of the cup and display it via a contact.

6.4.3 Control system subproject

The control system has the task of opening the solenoid valve when the right temperature is reached in the boiler. To do this, the LOGO! must evaluate the temperature signal and compare it to a target temperature. The target temperature should be easy for the user to adjust, because different types of tea are prepared at different temperatures.

The temperature is measured by an analog temperature sensor. These can be purchased already assembled or as a kit at favorable prices. If necessary, an amplifier or a potentiometer must be connected between the sensor and the LOGO! to make optimal use of the analyzable 0 - 10 V of the LOGO!.

In this case, a LOGO! with a display is used for setting the target temperature and the brewing time. The four cursor keys are used for entries. The display shows the brewing time, the target temperature, and the remaining brewing time after the preparation begins (Fig. 6.19).

Areas b and c of the overall circuit shown in Fig. 6.20 are similar to each other in that each area contains an “Up/down counter” (B011 and B016), which counts the presses of the cursor keys. To count backwards, invert the Dir input with the cursor C2 (shorter brewing time). Pressing C1 without C2 increases the value in B011. The circuit is divided into five sections (a-e):

a) Hardware circuit on the water boiler

b) Target temperature setting

c) Brewing time setting

d) Display of the values on the display

e) Rocker arm circuit for tea bag handling.

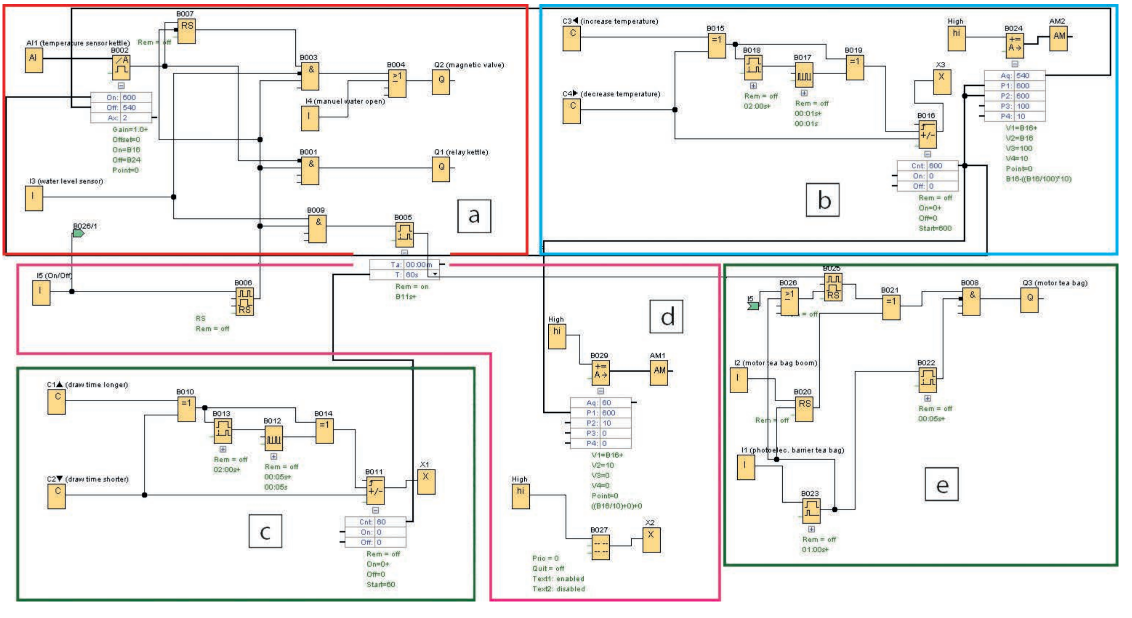


Fig. 6.20 Overall circuit of the automated tea brewer

If there are big time differences, it is tiring if the cursor key must be pressed for each individual step. Combining the blocks B010, B013, B012 and B014 facilitates the entry of higher values (Fig. 6.21). The XOR block B010 ensures that the signal is only further processed when a key is pressed. Both cursors, active and passive, lead to no forwarding of the signal. The ON delay B013 counts down 2 seconds before the signal is forwarded. During this time, the corresponding cursor key must be held permanently. Only then will the signal transmitter B012 be activated. This then switches on and off at a frequency of 10 Hz and thus modifies the values in B011 much more quickly and conveniently than would be possible using single entries. Fine adjustments can be made by releasing the cursor key and pressing it in single steps. The Cnt value is forwarded to B005. B005 later controls the rocker arm for removing the tea bag.

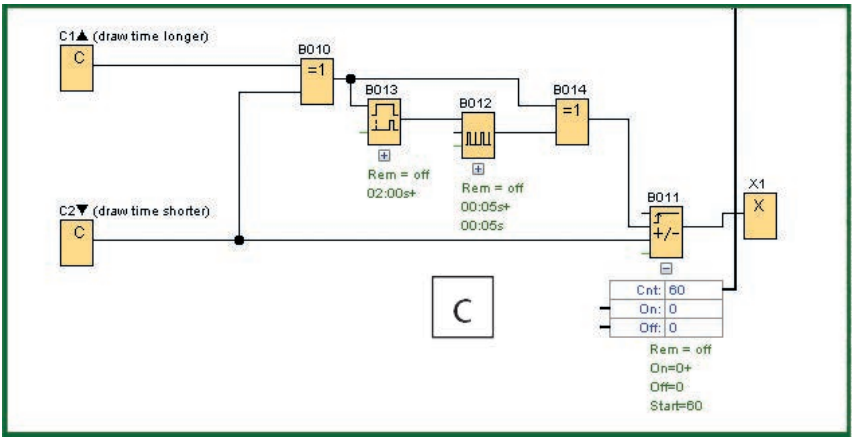


Fig. 6.21 Enlarged area c "Set the brewing time"

The setting of the target temperature is also organized using the same method (area b). Here, however, there is a difference from setting the brewing time: Because the relay which regulates the water boiler is controlled by an "analog threshold switch" (B002), it is necessary to define an ON and OFF value. The ON and OFF value must be on the analog scale 0 - 1000 and should represent the desired temperatures. The value is displayed in the "Up/down counter" B016. In the case of the temperature sensor, 0 degrees corresponds to the value 0 and 100 degrees corresponds to the value 1000. The remaining values correspond to the linear equation:

$$\text{Temperature [°C]} = \text{Value} / 10$$

The OFF value is defined in the circuit as 90% of the ON value.

This calculation takes place automatically by means of the "analog arithmetic" block B024, which is given the current ON value of block B016. B024 and B016 transfer their values to the threshold switch B002, which regulates the switching ON and OFF of the relay for the water boiler (Fig. 6.22). If the temperature in the water boiler lies under the ON value, the relay is activated (Q1).

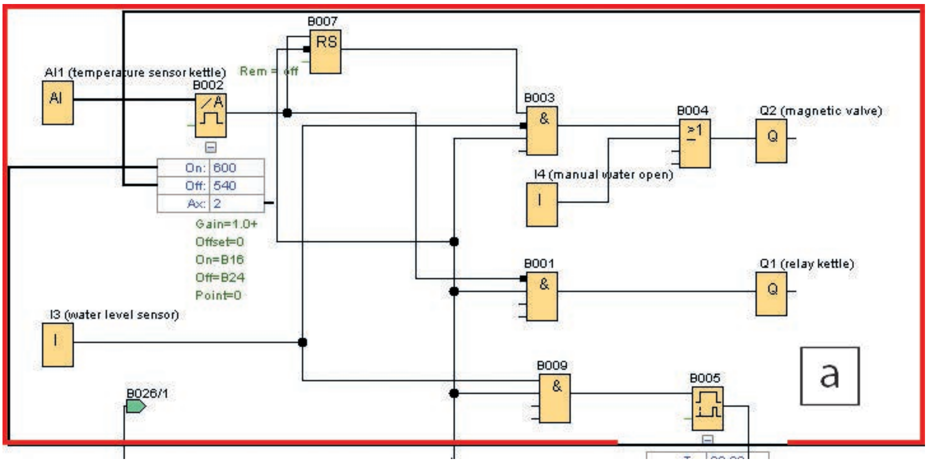


Fig. 6.22 Controlling the water boiler and solenoid valve

If the setpoint is reached, the relay switches off. At the same time, the solenoid valve for the water outlet opens, activated by output Q2. Only if the OFF value is undershot does the relay switch on again, unless the cup has already been filled and thus the solenoid valve has closed.

The setpoint brewing time, the remaining brewing time, and the setpoint water temperature are displayed in the display (B027). Since the value for the target temperature is 10 times the temperature in Celsius, an additional arithmetic block is connected upstream of the display, which divides the value by 10 (B029) (see Figs. 6.23 and 6.24).

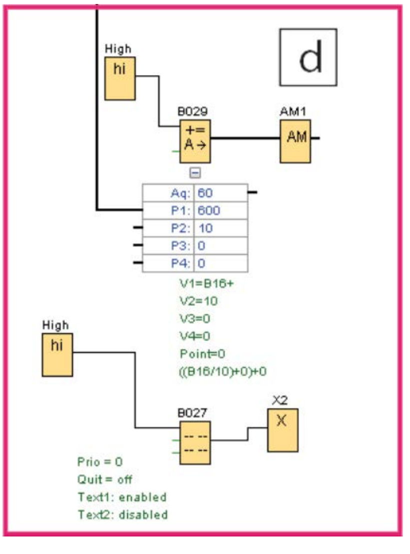


Fig. 6.23 Recalculation of the temperature and activation of the display

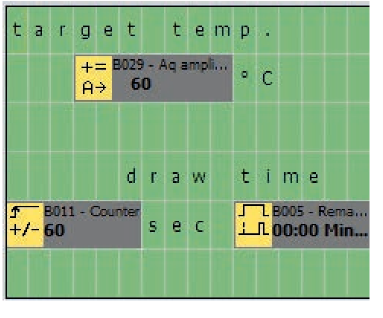


Fig. 6.24 Settings in the message text block

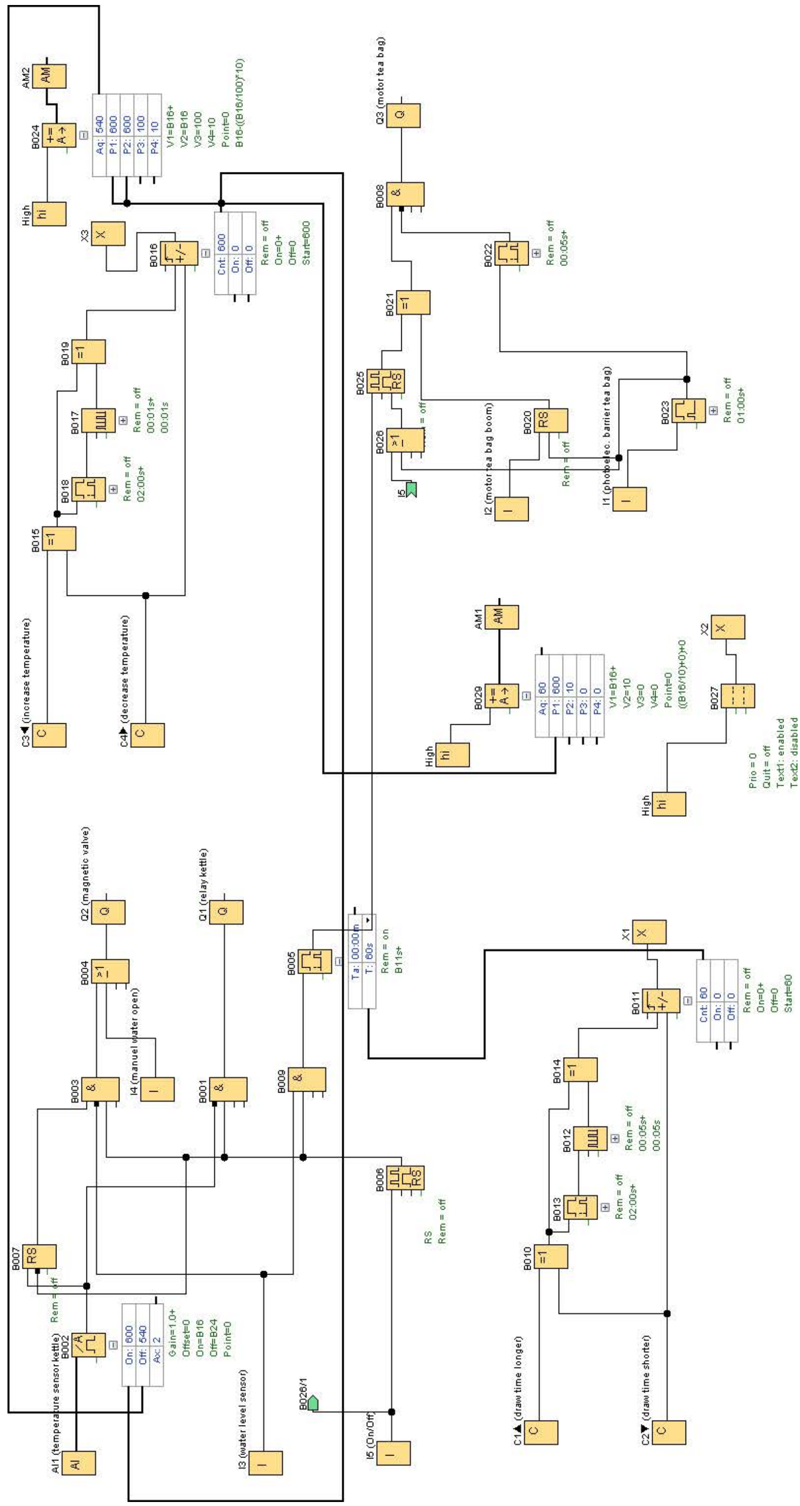


Fig. 6.25 Overall circuit of the automated tea brewer

The tea bag rocker arm is controlled by the blocks in area e. Q3 controls the motor, which moves the tea bag rocker arm via a disk. A 1 state at Q3 causes the motor to move. The motor moves until the light sensor I1 outputs a 1 signal via B023. B023 interrupts this signal after one second in order to ensure that the motor can be put into operation again even when there is an old 1 signal from I1. The disk to which the tea bag rocker arm is connected has a hole in the lowest and highest position, through which the light source on the one side triggers the light sensor on the other side. After each new activation of the motor, the light sensor is covered again, which causes B023 to receive an edge change at the Trg input and when a hole on the light sensor is reached again, it can be activated again for one second in order to stop the motor again.

The overall circuit for this automated tea brewer is shown in Fig. 6.25. The associated functions are compiled in Table 6.8.

Table 6.8 Function table of the project "Tea brewer"

Block name	Property	Function in the program
AI1	Analog input	Temperature sensor boiler
I1	Digital input	Light barrier of tea bag arm
I2	Digital input	Activate motor of tea bag arm
I3	Digital input	Water level sensor
I4	Digital input	Manual water open
I3	Digital input	On/Off
B001	Logic basic function AND	Water boiler is only switched on when ON/OFF = 1
B002	Analog threshold switch	ON and OFF water temperature
B003	Logic basic function AND	Opening of the solenoid valve depending on level and water temperature
B004	Logic basic function OR	Allows manual opening of the solenoid valve
B005	ON delay	Entry of the brewing time until the tea bag is removed
B006	Pulse relay	Switch the system ON and OFF
B007	Latching relay	Forwards the reaching of the target temperature
B008	Logic basic function OR	Allows the manual and automatic control of the tea bag arm
B009	Logic basic function AND	Connection of ON/OFF switch to water level sensor
B010	Logic basic function XOR	Lengthen or shorten brewing time simultaneously
B011	Up/down counter	Saves the desired brewing time and forwards it
B012	Incremental encoder	Signal generation for quick entry via the cursor keys
B013	ON delay	Controls the quick entry when cursor key is pressed
B014	Logic basic function XOR	Rule out simultaneous single-step/quick entry
B015, B019	Logic basic function XOR	Like B010
B016	Up/down counter	Like B011
B017	Incremental encoder	Like B012
B018	ON delay	Like B013
B020	Latching relay	Holds the signal for manual moving of the tea bag rocker arm

Block name	Property	Function in the program
B021	Logic basic function XOR	Either automatic or manual mode
B023	Interval relay/pulse output	Interrupts the 1 signal from I1 after 1 second
B024	Arithmetic instruction	Creates the target temperatures
B025	Pulse relay	Control of the motor of the tea bag arm via Q1 after the brewing time has elapsed
B026	Logic basic function OR	Motor can only be switched if ON/OFF = 1
B027	Message text	Display control
B029	Arithmetic instruction	Outputs the target temperature for the display output
C1, C2	Cursor keys	Lengthen/shorten the brewing time
C3, C4	Cursor keys	Increase/reduce the target temperature
High	High	Activates B029, B024, and B027
Q1	Output	Controls the relay of the load current of the water boiler
Q2	Output	Controls the solenoid valve on the water boiler
Q3	Output	Controls the motor for operating the tea bag holder

Tip

Using the relay, higher loads than those used in this example can also be controlled by the LOGO!. We advise against this, however, because the combination of water and electricity is especially dangerous. It is conceivable that a commercially available robotic arm could be used here as an expansion instead of the tea bag arm shown here. This would have its own control system, but it could receive the signal for immersing or removing the tea bag via the LOGO!. This would have the advantage that the robotic arm could also dispose of the used tea bag. Such robotic arms for hobbies and tinkering can easily be ordered online.

The opening of solenoid valves depending on temperature can just as well be applied for the warming of baby food.

6.5 An electronic lock

There are a number of applications for which it makes sense for a user to be authenticated. Even if the LOGO! is not sufficient for the access control of secured areas and facilities, it can make perfect sense to enter a code for activating or deactivating an alarm system, an application, or a cabinet locking system. This project introduces such an application option with a binary 8-bit code (Fig. 6.26).

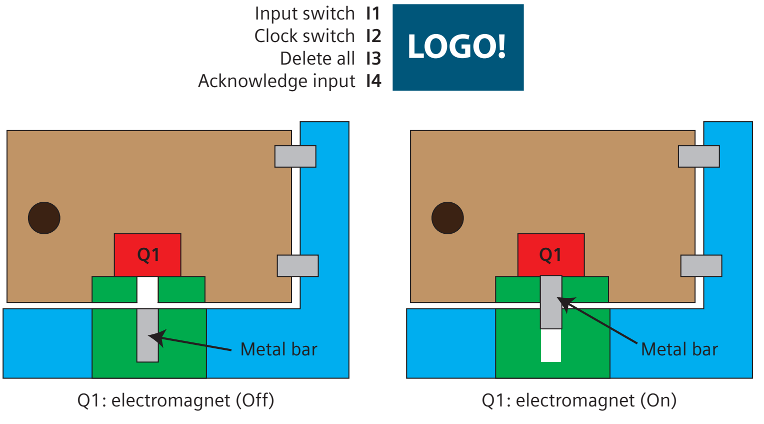


Fig. 6.26 Locking a door by activating an electromagnet and a metal bolt

Table 6.9 Contents and functions of the "Electronic lock" project

Content	Blocks used in the program
✓ Building a flip-flop ✓ Forwarding the status of a bit ✓ Parallel output of linear inputs	✓ Digital inputs ✓ Bit memory ✓ AND ✓ OR ✓ Latching relay

The project shown here is an effective and reliable option for initiating or ending actions of the LOGO! via authentication.

However, we strongly advise against using self-made locking mechanisms on safety-relevant or potentially hazardous devices, objects, and systems!

The locking system that is described is based on an electromagnet holding a locking bolt against the force of gravity. It not only opens when the right code is entered, but also when the LOGO! is disconnected from the power source. Should such a circuit be used, corresponding structural precautions must be made to prevent the LOGO! from being disconnected from the mains.

AND blocks, OR blocks, and bit memories are primarily used when building the binary lock. Furthermore, individual inputs were negated on some of the blocks. A NOT block is also used.

In any case, four hardware pushbutton switches are needed. An actuator, which is activated or deactivated when the right code is entered, is also needed. In case of this project, an electromagnet is used, which is deactivated when the right code is entered.

6.5.1 Shift register function of the LOGO!

Sometimes it may be necessary to output bits of information that arrive successively at an input simultaneously at several outputs. The LOGO! offers the "Shift register" function for this (Fig. 6.27).

A shift register is a sequential logic system with a series of successive (virtual) flip-flops. In electronic circuits, flip-flops can assume two states via a toggle function, either ON (1) or OFF (0). Each flip-flop is thus capable of storing a bit over a long period of time. If you connect several of these flip-flops in succession, this is called a shift register, which gets its name from the fact that the state of the first flip-flop is passed on to the next flip-flop with each cycle. Shift registers can be compared to bucket brigades in which either a full bucket (state 1) or an empty bucket (state 0) is passed on. For each post in the chain, the status of the current bucket can be queried. The states (buckets) can always only be given either in one direction or in the other direction at the same time. Serial signals can thus be saved and output in parallel. Conversely, parallel signals can be converted into a serial sequence. Both functions are possible with the shift register block of the LOGO! (Figs. 6.28 and 6.29).

The number of shift registers is limited in the LOGO! to four blocks per function block diagram. Each register can store 8 bits. Block "S" is necessary for reading out the shift register.

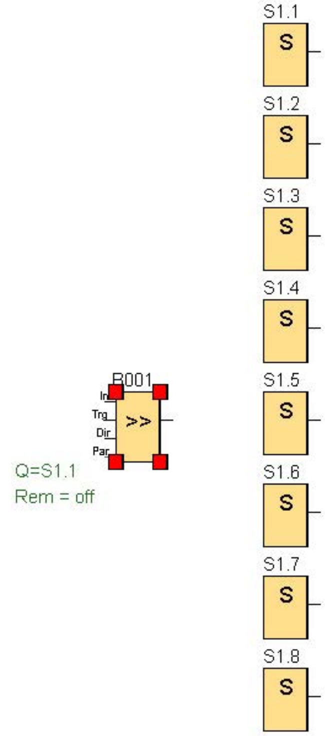


Fig. 6.27
Shift register arranged in parallel

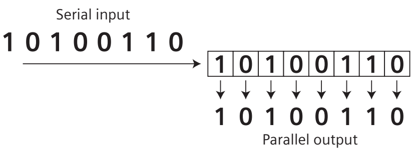


Fig. 6.28 Functioning principle of a parallel-in-serial-out shift register

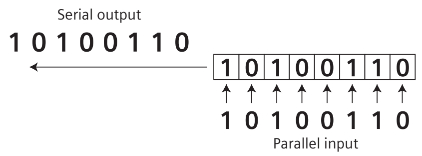


Fig. 6.29 Functioning principle of a serial-in-parallel-out shift register

It is also possible to manually create shift registers. These give the user more leeway with the number of bits which are to be stored in the register. Virtual flip-flops are also simulated when manually setting up shift registers in the LOGO!. By means of a “clock pin” signal of the clock generator of the “bucket brigade”, the bits are “passed on” from flip-flop to flip-flop, during which each flip-flop adopts the state of the preceding 1-bit memory and passes on its previous state. This circuit is a serial-in parallel-out circuit, which is to be used to enter a binary code for a locking mechanism. An 8-bit code is to be implemented with four inputs and only one output. The number of bits could be expanded and reduced, unlike with the default shift register blocks.

6.5.2 Brief description of the function block diagram

The function block diagram in Fig. 6.30 shows a binary 8-bit lock, which controls an electromagnet via output Q1. The function block diagram is divided into the areas a, b and c. The arrangement and function of the remaining blocks correspond to the blocks in area a and form a flip-flop in each case. Area b is designed in such a way that all of the flip-flops are set to the 0 state when input I3 is actuated. Area c compiles the serial information of the individual flip-flops and activates an electromagnet via output Q1 if the combination is correct.

The flip-flops function according to the following principle (Fig. 6.31): The output Q for block B004 “AND with edge evaluation” is set to 1 by pressing the button I2 “Clock Pin”. If I1 “Data input” is also set to 1 at the same time, then the AND condition of B001 is satisfied. B001 forwards the 1 state to the latching relay B003. If I1 equals 0 when I2 is set to 1, then the AND condition of B002 is satisfied, where the input of I1 is negated. Thus, the S input of B003 is not activated as for I1=1. Instead, the R input is activated, which leads to the zero setting of B003. The OR block B029 is interconnected in each flip-flop in order to empty the register if I3 “All bits to 0” is actuated. Clock Pin I2 is connected to all of the flip-flops. The bit memory at the end of the flip-flop buffers the forwarding of the bits.

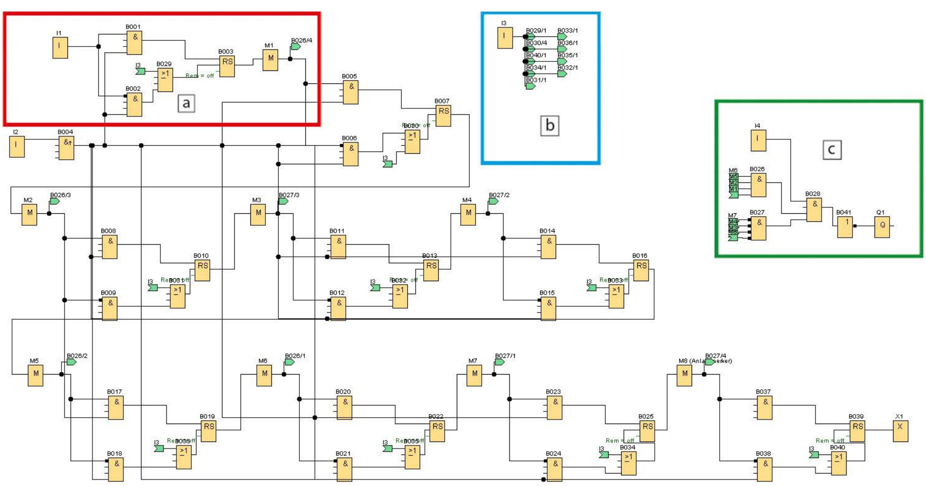


Fig. 6.30 Function block diagram of the binary 8-bit locking mechanism

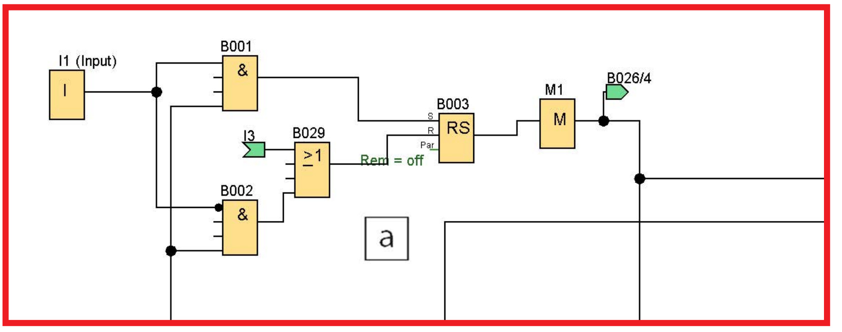


Fig. 6.31 Function block diagram of a flip-flop

Area “b” is organized in a simple manner (Fig. 6.32): If you press I3, all of the latching relays in the shift register are set to 0. The connections are split for a better overview.

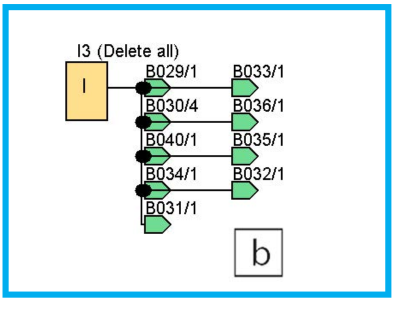


Fig. 6.32 Function block diagram “b”

Area “c” can be explained just as easily (Fig. 6.33). The statuses of the latching relays are collected in the AND blocks B026 and B027. All of the AND blocks are only switched if all of the inputs are correctly activated or negated. The result of this is the binary 8-bit code. The status of the AND blocks is only forwarded when I4 is pressed in order to prevent random combinations from switching output Q1 during “forwarding”. As needed, the right combination of flip-flops can lead to the output being activated or deactivated by inserting a negation. In this case, we decided on deactivation.

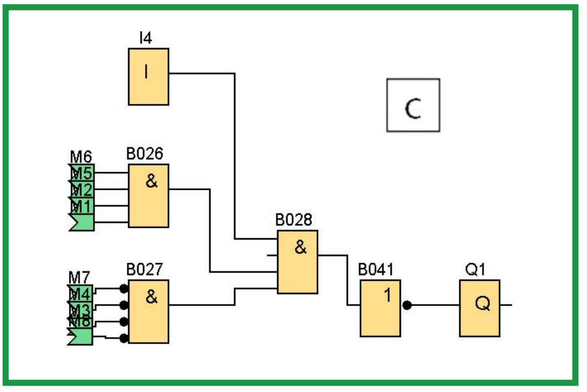


Fig. 6.33 Function block diagram “c”

Table 6.10 Function table of the project “Electronic lock”

Block name	Property	Function in the program
I1	Digital input	Assume either state 1 or 0
I2	Digital input	Assume state of I1
I3	Digital input	Delete all stored entries
I4	Digital input	Input completed
B001, B002	Logic basic function AND	Enters a 1 signal/0 signal in the flip-flop
B003	Latching relay	Holds the state 1 of B001 or is reset by B029
B004	Logic basic function AND (edge)	Ensures that a 1 cycle follows a 0 cycle, and vice versa
B005, B006	Logic basic function AND	First 1 input block/0 input block in the follow-up flip-flop
M1	Bit memory	Holds the state of B003 for one cycle and forwards it to the follow-up flip-flop.
B007 – B027	Correspond to B001 - B006 + B029	The individual flip-flops are logically identical
B028	Logic basic function AND	Connects I4 with the states of the individual flip-flops
B029	Logic basic function OR	Permits the simultaneous deleting of all flip-flops by I3
B030 – B040	Correspond to B001 - B006 + B029	The individual flip-flops are logically identical
B041	Logic basic function NOT	Outputs 1 if B028 does not have 1 state
Q1	Output	Controls the control mechanism

Tip

The authentication via the shift register does not have to end in the execution of a task by an actuator. Rather, it also suited for initiating or ending more complex circuits in the LOGO!. Thus, for example, an alarm system with a motion detector can be activated or deactivated or an alarm signal can be switched on and off like a rotating beacon. If the door is not an entry door to a safety-relevant area, the shift register could also be integrated into a circuit for the motor control of a barrier or of a gate, or into the system, which is described in chapter 6.6.

6.6 Object monitoring

In many cases, the security and protection of important objects is a basic requirement for peace of mind and sustainability. Often, however, comprehensive safeguarding means a loss of convenience. The following project should provide practicable handling, but nevertheless a high degree of security for protecting various possible units. Whether you are monitoring a garden plot, a parked recreational vehicle, the access control of an apartment or a private area: Multi-faceted protection of the object can be achieved using the following circuits and a LOGO! (Fig. 6.34).

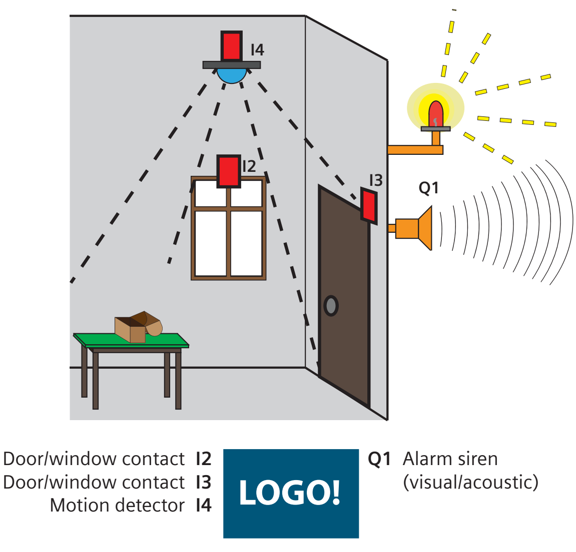


Fig. 6.34 Functions of object monitoring

Table 6.11 Content and functions of the project "Object monitoring"

Content	Blocks used in the program:
✓ Motion detection	✓ Digital inputs and outputs
✓ Time-controlled motion detection	✓ Week time switch
✓ Tilt switch	✓ AND, OR, NOT gate
✓ Time-delayed switching	✓ ON and OFF delay
✓ Message texts	✓ Up/down counter
✓ Message via SMS	✓ Latching and pulse relays
✓ Image recording via a camera	✓ Incremental encoder
✓ Data logging	✓ High
	✓ Data log
	✓ Message text
	✓ Open terminals

The object monitoring system described in the following is modular in design. Different sensors can be used depending on the object to be monitored. In the signal processing area, a decision will be made as to whether a message is to be processed with a time delay, the intensity or frequency with which a signal is to sound an alarm, or which of the actuators is to be activated. Various actuators can be integrated (Fig. 6.35). The system can be configured in any way, from a simple acoustic message or a camera image to a message text on a display or smartphone. Another sensible feature is the capability of saving all of the received signals to be able to read them externally later.

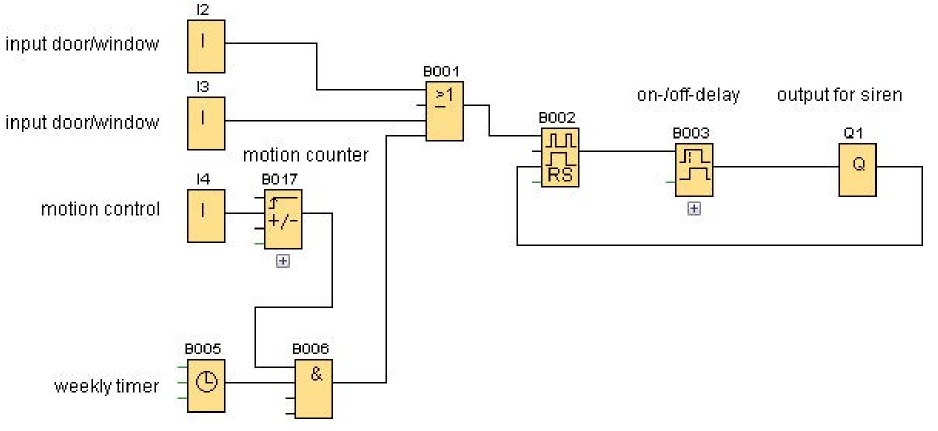


Fig. 6.35 Various alarm inputs and connection of an actuator

The basis of object monitoring is any number of door and window contacts, which are connected directly to the inputs of the LOGO!. Pushbuttons can be used as NO contacts, NC contacts or switches as required. Another sensible supplement is the integration of a motion sensor module. Ready-made 12 V blocks can be purchased from an electronics accessories dealer[13]. You may have to observe whether a digital or analog output signal is created. In the latter case, an additional analog threshold switch must be installed. Since motion detectors often also record unwanted objects (e.g. small animals) or signals are passed on as a result, it is recommended that you define a threshold value via a counter. In this case, the motion counter B017 was set to five messages, for example. In addition, the detection of moving objects was also limited to the night hours by a day-time timer clock B005. All of the signals are compiled by the AND gate B006 and the OR gate (B001).

To activate the output of the signal encoder Q1 or to deactivate it again after a specified time, the signal is controlled via a latching relay B002 and a time delay B003. It must be noted that the latching relay B002 is reset again by the signal at Q1 (Fig. 6.36).

[13] Motion detector block 8 - 12 V, detection range 5 m, e.g. available from conrad.com, order number 190952

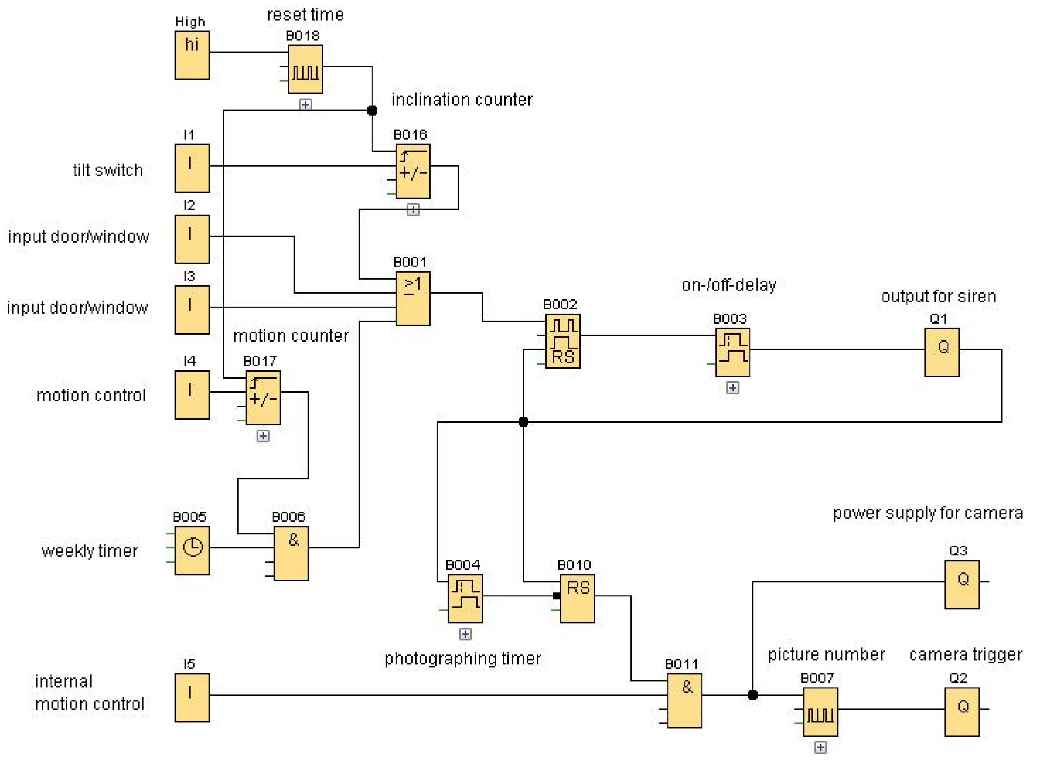


Fig. 6.36 Integration of a tilt switch, a reset function, and a camera module

The system is expanded by a tilt switch in the following[14]. This too should be adapted with a threshold (5 movements in this case) via a tilt counter (B016). To be able to define the threshold function of the tilt switch and the motion detector more precisely, both counters should be reset after a specific amount of time. To do this, a time is specified via an incremental encoder (B018), after which both reset inputs of the counter are reset to zero.

6.6.1 Integration of a camera

Another useful function for monitoring closed objects is the documentation of interior spaces via a camera. Another motion detector is connected to the system for this purpose. Since a malfunction or false detection of the motion detectors inside the object can be ruled out, here, the definition threshold is omitted. The camera itself is only triggered, however, if a signal from the other sensors has activated the output of the alarm siren beforehand. This takes place via a latching relay (B010) and an AND gate B011. The photographing time can be set to 10 seconds, for example, via counter B0004. After the counting process has ended, the reset input of the latching relay B010 is activated.

[14] Tilt switches up to 60 V/0.25 A 1 NO/1 NC, e.g. available from conrad.com, order number 700444

The best cameras for this are cameras that have an external shutter control switch (Fig. 6.37). These types of devices are available as helmet cameras for sports[15] or for wild game monitoring as a hunting accessory. Depending on the device, the camera can either be connected to the continuous pulse of the system or is only activated as required via a separate output of the LOGO! (Q3). You might have to take a warm-up time or boot phase into consideration for the camera. Without an external shutter control switch, the shutter control switch of the camera is connected to the output (Q2) of the LOGO! in order to document images of the interior space to be monitored. In the case of the system presented here, an additional incremental encoder has proven useful. A sequence of images can be created via its time setting.

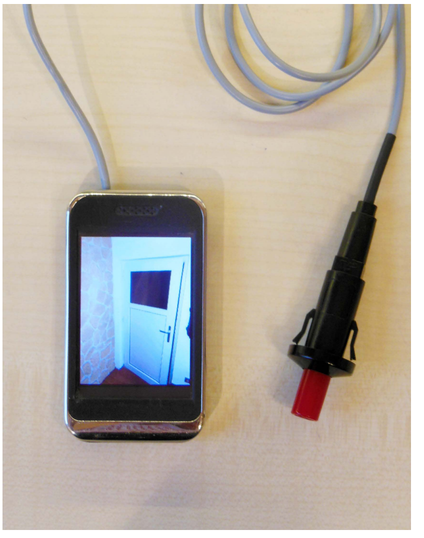


Fig. 6.37 Helmet camera with remote shutter switch and external camera

In order to be able to also control the system functions remotely, the output can be expanded by a simple message text, e.g. a text display connected in the network or the SMS module described in Chapter 3.6 (Fig. 6.39). In order to be able to distinguish between the triggered sensors in the text output, a connection is needed from each individual sensor signal and the alarm signal. For the output of the tilt switch signal, the incoming signal is therefore saved in a latching relay (B014), connected to the output Q1 via an AND gate (B013), and forwarded to the message text. The message text of the contact switches can be determined directly at output Q1. Similar to the method already described, regarding the output of the motion detector signals, the signal is held in a latching relay (B015), connected to the output Q1 via an AND gate (B009), and forwarded to the message text (Fig. 6.38).

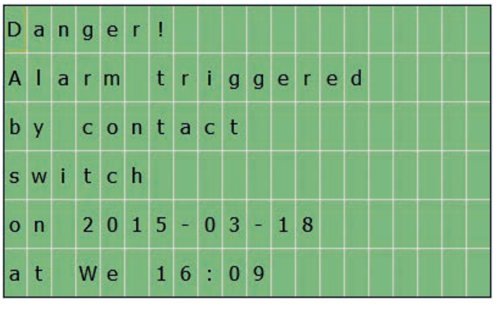


Fig. 6.38 Integration of a data logger and a message text function

[15] DVR II helmet camera. Such cameras are e.g. available from www.helmetcameracentral.com.

To get an overview of all of the incoming alarm messages while operating the system, it is recommended that you also integrate the data log function (described in Chapter 3.5) into the circuit. Only one memory unit (B015) is integrated for recording alarm signals for this purpose. These signals converge in the OR gate (B021). The data logger is then connected to the memory. Alternatively, you can also configure the menu in such a way that the signals of digital inputs I1 - I8 are logged directly (Fig. 6.39).

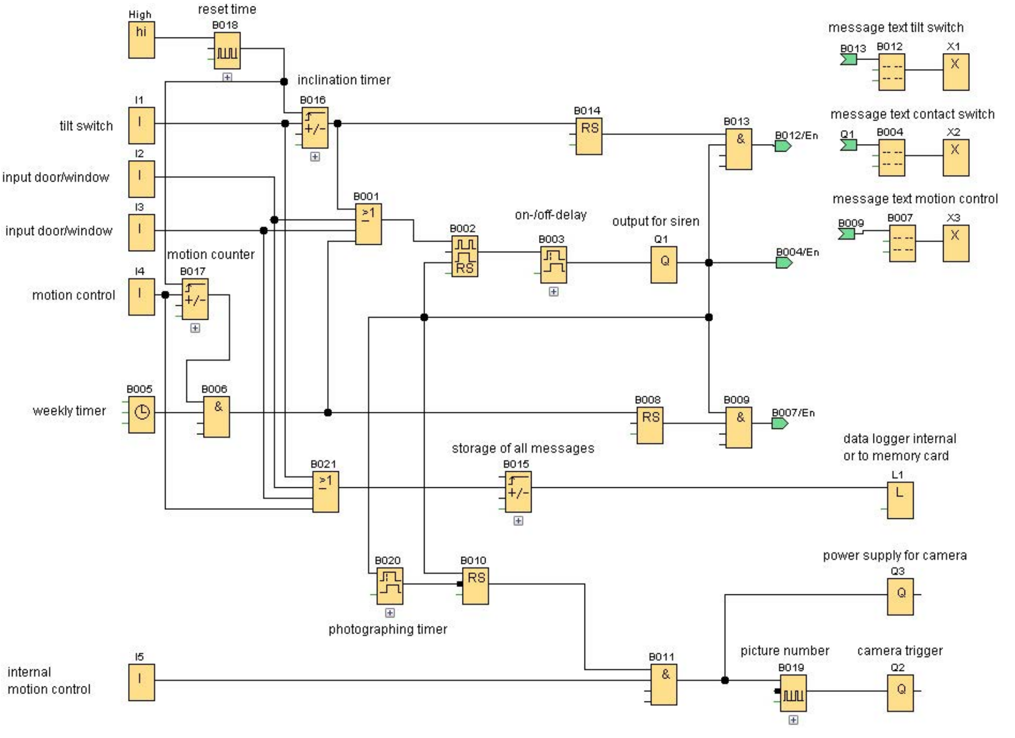


Fig. 6.39 Text message when alarm is tripped via a door or window contact

Table 6.12 Function table of the "Object monitoring" project

Block name	Property	Function in the program
B001	Logic basic function OR	Collects the input signals of the sensors
B002	Pulse relay	Holds the alarm signals of the sensors
B003	On/Off delay	Time delay of the alarm signal
B005	Weekly timer clock	Setting the time of day for the motion detector
B008	Latching relay	Holds the contact triggered by the motion detector
B009	Logic basic function AND	Connects the signal from the motion detector with the alarm signal
B010	Latching relay	Holds the alarm signal for the camera
B011	Logic basic function AND	Connects the alarm signal with the input signal of the internal motion detector
B012, B004, B007	Message texts	Texts for display outputs and SMS module

Block name	Property	Function in the program
B013	Logic basic function AND	Connects the signal from the tilt switch with the alarm signal
B014	Latching relay	Holds the contact triggered by the tilt switch
B015	Up/down counter	Counts all of the triggered alarm signals
B018	Incremental encoder	Time for reset of the counters
B019	Incremental encoder	Trigger time for individual camera images
B020	On/Off delay	Setting the photographing time
B021	Logic basic function OR	Collects the input signals of the sensors without threshold value
High	Signal for incremental encoder	Assume state of I1
I1 – I5	Digital inputs	Input for tilt switches, contacts, and motion detectors
L1	Data log	Storage of all received alarm signals for external evaluation
Q1	Output	Connection for alarm siren
Q2	Output	Connection for shutter control of the camera
Q3	Output	Connection for power supply of the camera
X1 – X3	Open terminals	Output for display

Tip

The following options are available for expanding the system:

The described tilt switch can be self-made with no problems. Operating instructions can be found in the lawn mower project in Chapter 6.15.

To activate the system, a central key switch can be integrated into the power supply of the LOGO!. However, an RFID system is more convenient and is commercially available at favorable prices.

Alternatively, the code lock described in Chapter 6.5 can also be used for switching the system ON/OFF. Depending on the sensors and actuators connected to the system, the inputs or outputs must then be expanded by an additional module.

Another possible expansion could be the integration of a GPS receiver into the system (see Chapter 3.6). For mobile monitoring objects, a message with the current GPS coordinates of the system can be called up via SMS upon request using the receiver. Possible applications would be, for example, boats, vehicles, or other mobile goods.

6.7 An intelligent garden watering system with process water control

Especially in the hot summer months, watering a garden takes up a lot of time and energy. Large manufacturers of garden accessories offer automated watering systems for a modular configuration, and it is also easy to implement an automated watering system using the LOGO! (Fig. 6.40). This allows the costs for the hardware to be kept relatively low. The following control system allows the user to water his garden depending on the soil humidity and to also manage the contents of an underground cistern and water reservoir. A cock can also be integrated as needed. This solution combines sustainable use of process water and intelligent watering. Partial aspects of the overall control system are described in detail in the following in subprojects.

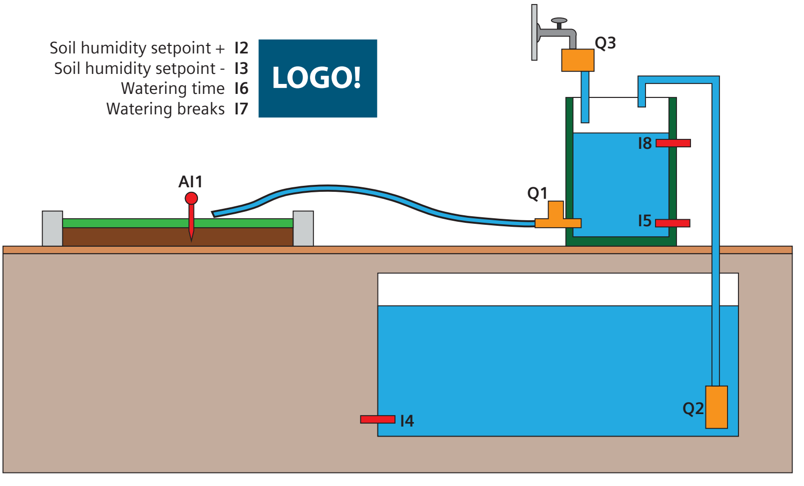


Fig. 6.40 LOGO!-based garden watering with process water

Table 6.13 Content and functions of the “Garden watering” project

Content	Blocks used in the program
✓ Analyzing an analog signal ✓ Two methods for setting and resetting target values ✓ Activating a solenoid valve ✓ Synchronized controlling of a pump ✓ Outputting message texts	✓ Analog and digital input ✓ Arithmetic instruction ✓ AND, XOR, OR gate ✓ On/Off delay ✓ Up/down counter ✓ Latching relay ✓ Incremental encoder ✓ High ✓ Interval relay/pulse output ✓ ON delay ✓ Message text ✓ Analog threshold switch ✓ Analog amplifier

6.7.1 Soil humidity measurement subproject

Industry-standard soil humidity meters are used in our intelligent garden watering system. They inductively measure the water content of the soil. They output an analog signal, which is regulated to the necessary 0 - 10 V with the aid of a potentiometer. Do-it-yourselfers can also measure the degree of moisture in the soil by the resistance between two electrodes. This has the advantage that the sensors are easy and cost-effective to make. However, when used continuously in the garden, they also cause some problems. On the one hand, they are subjected to a high degree of wear and often have to be replaced after only one year. In addition, they cause electrolysis in the soil in which charged particles accumulate on the anode and cathode and thus lead to a change of the measured values. Both problems can be circumvented by pouring the electrodes into plaster, which protects the electrodes from corrosion and charged particles, but also takes up water at the same time, thus changing its resistance depending on the soil humidity. However, the plaster is also subjected to the weather and the manufacture of sensors is considerably more complex. Industry-standard induction soil humidity sensors provide the solution for all of the problems described above.

6.7.2 Watering process subproject

In our example, the water pressure for watering is generated by the weight of the water column in the water reservoir and the difference in height to the hose. For opening the water hose a solenoid valve is sufficient. The solenoid valve could be replaced at any time by an additional pump, which could do the watering without or against gravity. As an alternative to the water tank or the cock in this example, a well or tube well could be used to fill the water tank and the water reservoir and compensate for loss of water due to watering. The solenoid valves open for a 1 signal and close if there is no positive signal. The output Q2 “pump tank” controls a relay, which regulates the pump flow in order to avoid overloading the LOGO!.

Since water needs time to seep into the soil and uniformly dampen the ground after the watering process, it makes sense to define a duration for the watering. After the watering time, it is also necessary to define a break time during which the water has time to seep into the soil before the next measurement. After this, the watering measurement process can be repeated in loops as needed.

6.7.3 Overall circuit for garden watering

The circuit diagram shown in Fig. 6.41 is divided into five sections (a-e). Section “a” is used for setting the desired soil humidity (Fig. 6.42). In section “b”, the values from “a” are processed in such a way that they can be used again. Section “c” activates the actual watering via a solenoid valve depending on the watering and break times which can be set in section “d”. Section “e” regulates the water management.

The setpoint value of the soil humidity is set via a counter, which can be changed up and down using two inputs (I2 and I3). When an input is held longer than 2 seconds, the counting process is accelerated via the incremental encoder B028. To save entry time, the value that is entered in B027 is multiplied by 10 via the arithmetic computation block B032 – the analog values are specified in values between 0 and 1000. Since the water is ultimately regulated via an “analog

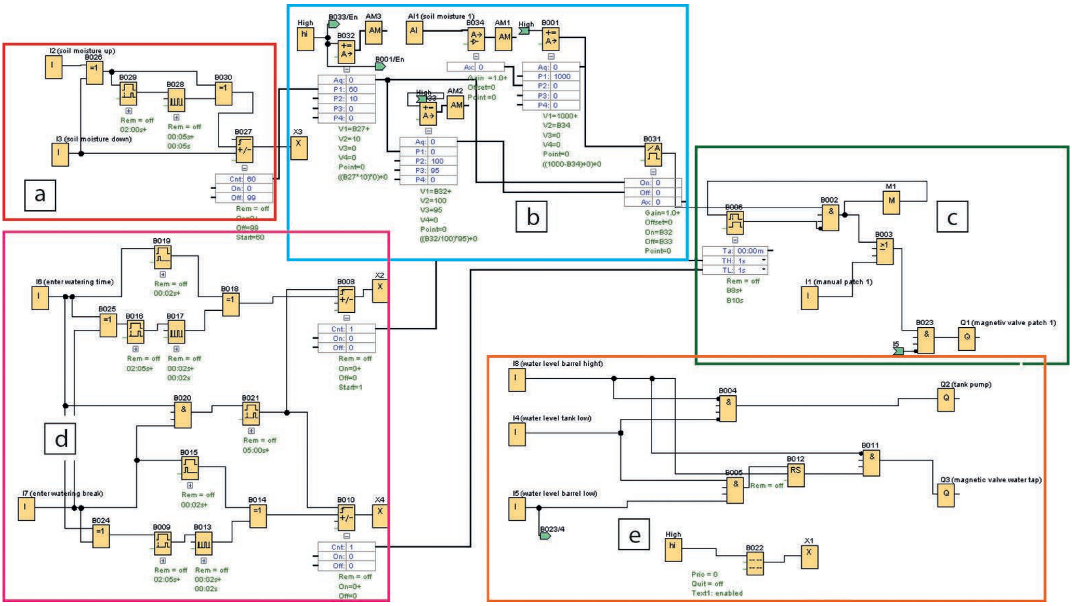


Fig. 6.41 Overall circuit of the garden watering

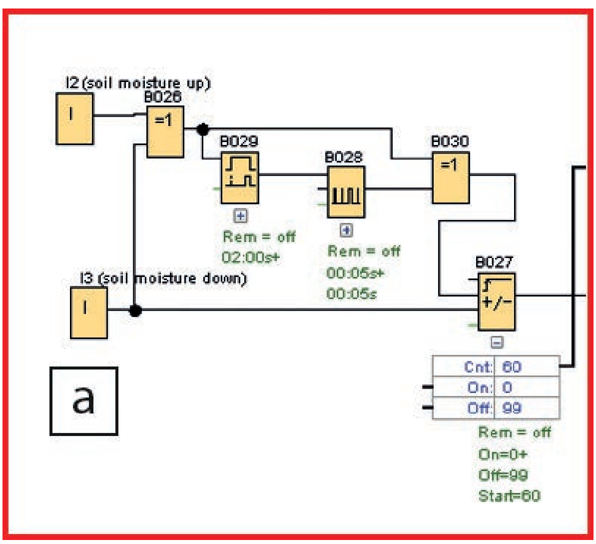


Fig. 6.42 Activation of the solenoid valve

threshold switch" B031, it is necessary to define both an "ON" and an "OFF" value. The ON value comes from the afore-mentioned computation block B032. Depending on that, the OFF value is calculated as 95% of B032 in block B033. The ON and OFF values are forwarded to the threshold switch B031. This part is shown in section "b" (Fig. 6.43).

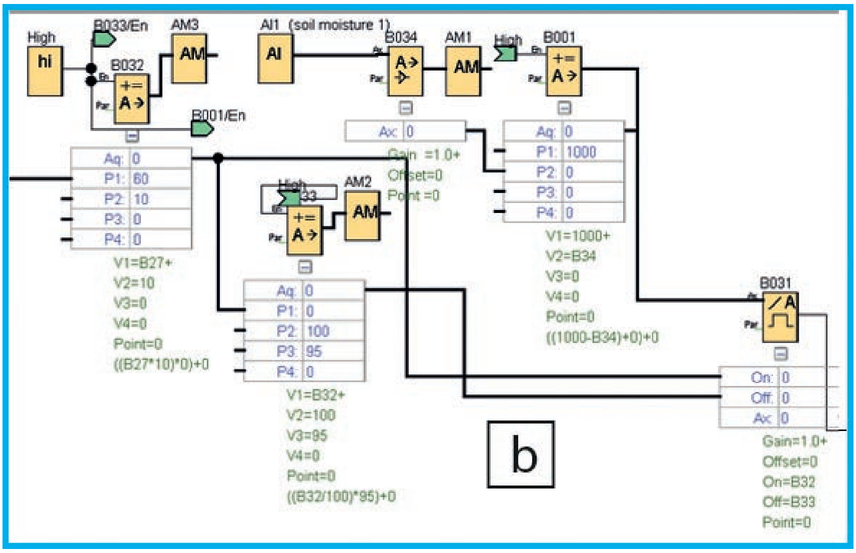


Fig. 6.43 Controlling the watering times and breaks

AI1 “Soil humidity 1” outputs values between 0 and 1000, where 1000 corresponds to very damp and 0 corresponds to dry soil. Since the threshold switch must stride the ON value from the lower value to the higher value and the OFF value must be under the ON value, the analog value of the sensor is subtracted from 1000 in order to get a dryness value. In turn, this value is further processed by B031.

Sections “c” and “d” are connected to each other to the extent that the solenoid valve for the watering is controlled in “c” (Fig. 6.44). The necessary signals come from the “ON/OFF delay” B006, which, in turn, gets its parameters from section “d”. The watering loop, comprised of the watering time and measuring time, runs as long as AI1 indicates that the soil is still too dry.

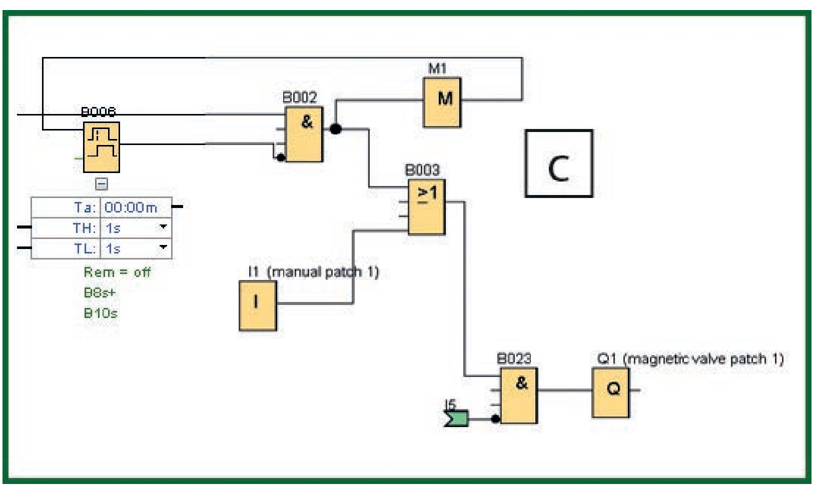


Fig. 6.44 Water level messages with the automated responses

The times, i.e. how long the watering should last and how long the system should wait until the next watering pulse, are entered in section “d” (Fig. 6.45). The controlling in “d” takes place according to a principle similar to the one in “a”. The counters for setting the time differ in one important feature: There is only one input each for the setting of the watering duration and the waiting time. If you enter incorrect values, it is no longer possible to correct the values downward. By pressing both keys for 5 seconds, you can simultaneously activate the ON delay B021, which, in turn, then deletes the time entries for the watering time (B008) and waiting time (B009).

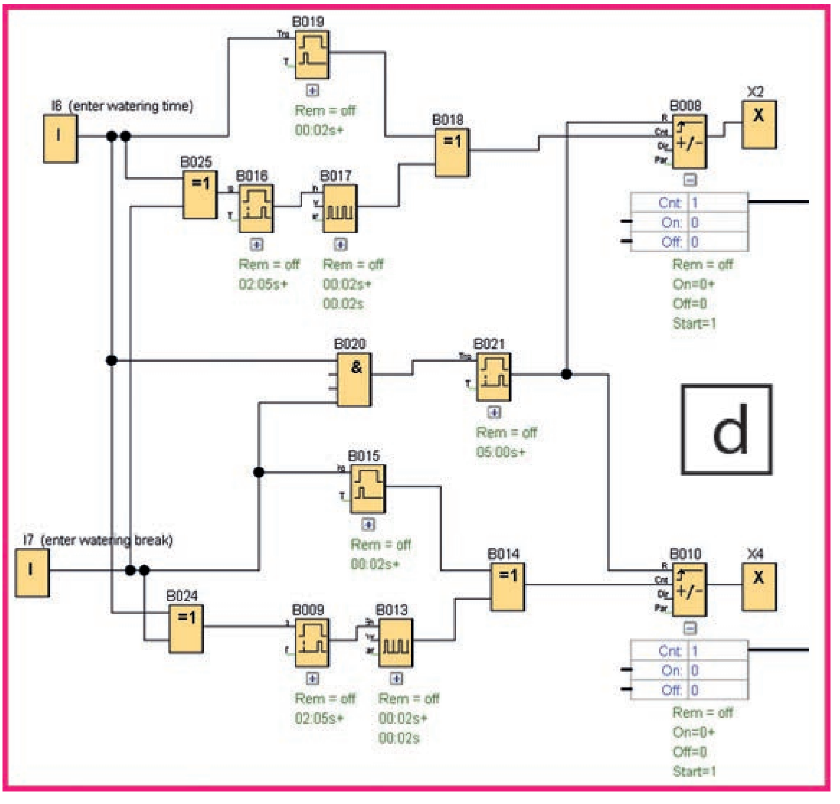


Fig. 6.45 Entry via keys and fast forward

Section “e” is used for regulating the water level in the process water reservoirs. A water reservoir in the garden usually holds between 200 and 500 liters of water. The water is conveniently accessible and is extracted either by a cock or by scooping. It also plays a central role in our intelligent watering system.

Water reservoirs have advantages and disadvantages in the system described above. The 500 liters of water do not last long in hotter months. When there is ample rainfall, they fill up quickly, which means that less water is collected than possible. The remedy for this is underground cisterns, which hold more than 1000 liters and can be fed via rain spouts or infeeds. However, this lacks easy access.

The LOGO! controller in our case detects when the water level drops below the desired “Water level of reservoir high” via sensor I8 in the water reservoir and then activates the “tank pump” Q2. This only happens, however, if the water level in the tank has not dropped below the minimum “Water level of tank low” I4, in order to prevent the pump from running dry. A 12 V pump is available at low cost. In the event that these pumps have a maximum runtime before they should take a break, it is possible to integrate this into the program. If the water level in the cistern is not too low and the reservoir is no longer full, water is pumped from the cisterns into the reservoir until the reservoir is full or the cistern is empty (Fig. 6.46).

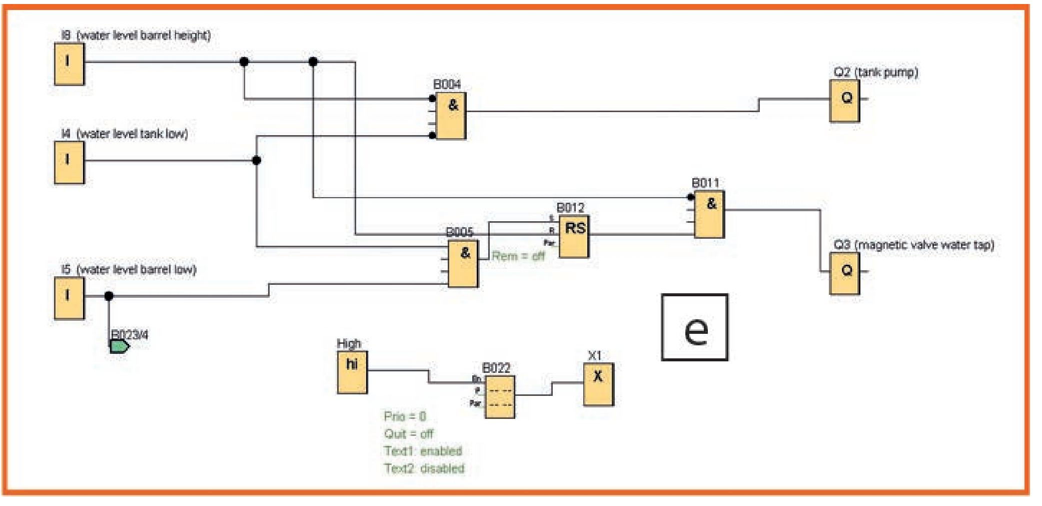


Fig. 6.46 Refilling depending on the water levels

If there are long periods of drought in the hotter months and the cisterns are so empty that water is no longer being pumped, the cock is used as a last reserve. Water is only taken from the water line if there is no more “free” process water. To this end, solenoid valve Q3 activates the water infeed at the cock if

- sensor I4 in the cistern indicates that it is empty,
- sensor I8, as a safeguard, indicates that the reservoir is not filled to the maximum,
- and the water level sensor on the bottom of the reservoir indicates that the reservoir is empty.

To this end, the latching relay B012, which is switched off by I8, is activated when the reservoir is full again. As long as I5 indicates that the reservoir is empty, the solenoid valve Q1, which controls the watering of the garden, cannot be activated.

Table 6.14 Function table of the project "Garden watering"

Block name	Property	Function in the program
AI1	Analog input	Soil humidity sensor
I1	Digital input	Manual watering
I2, I3	Digital inputs	Increase/reduce target soil humidity
I4	Digital input	Water level sensor tank low
I5	Digital input	Water level tank low
I6, I7	Digital inputs	Input of watering duration and breaks
I8	Digital input	Water level tank high
Q1	Output	Solenoid valve bed
Q2	Output	Pump in tank
Q3	Output	Solenoid valve at cock
X1	Open terminals	For message text operation
M1	Bit memory	Needed for recursion
B001	Arithmetic instruction	Calculates the threshold for switching on the watering system
B002	Logic basic function AND	Connects watering times with target soil humidity
B003	Logic basic function OR	Either automatic or manual watering
B004	Logic basic function AND	Only waters when the reservoir and tank have enough water
B005	Logic basic function AND	Only waters when the reservoir has enough water
B006	On/Off delay	Controls the watering and break time
B008	Up/down counter	Calculates the watering time and forwards it
B009	ON delay	Allows quick counting
B010	Up/down counter	Calculates the watering breaks and forwards them
B011	Logic basic function AND	Connection of the cock only in special cases
B012	Latching relay	Holds the water level until the reservoir is full again
B013	Incremental encoder	Generates the signal for quick entry
B014	Logic basic function XOR	Either individual entry or computer entry
B015	Interval relay/pulse output	Individual entry
B016	ON delay	Counts the time until the start of the quick entry
B017	Incremental encoder	Generates pulse for quick input of I6
B018	Logic basic function XOR	Either individual or quick entry
B019	Interval relay/pulse output	Interrupts the input of I6
B020	Logic basic function AND	Allows deletion through joint holding of I6 and I7
B021	ON delay	Delays the deletion for the simultaneous holding of I6 and I7
B022	Message text	Displays the parameters on the display
B023	Logic basic function AND	Excludes watering when water level in reservoir is low.
B026	Logic basic function XOR	Either I2 or I3
B027	Up/down counter	Records the inputs for the soil humidity/forwards them

Block name	Property	Function in the program
B028	Incremental encoder	Generates a pulse for quick entry
B029	ON delay	Activates the incremental encoder B028 after 2 seconds
B030	Logic basic function XOR	Either individual or quick entry
B031	Analog threshold switch	Connects the values of B033 and B032 in one threshold switch
B032	Arithmetic instruction	Calculates the target soil humidity
B033	Arithmetic instruction	Determines a percentile value of B032
B034	Analog amplifier	Amplifies the analog signal by a factor of 10
High	High	Activates the message text B022
High	High	Activates the arithmetic blocks B032, B033 and B001

Tip

The circuit can be expanded by additional soil humidity sensors and water outputs according to the same principle.

A tube well can be used instead of the cock in order to allow the reservoir and cisterns to also be filled during hot weather. This would then make a pump necessary instead of the solenoid valve.

6.8 Speed measurement with the LOGO!

In regulated traffic zones, there are often digital displays which display the speed of a driver's vehicle. Similar systems are also often used on modern ski slopes to measure the descent times. The following system is intended to be a speed display for a specific route, a time recording system for a soap-box derby, or a speed measurement for any objects or persons (Fig. 6.47). The maximum speed for this presented application is 30 km/h (18.7 mph). For reason of easier calculation, in this example we have used “km/h” instead of “mph”.

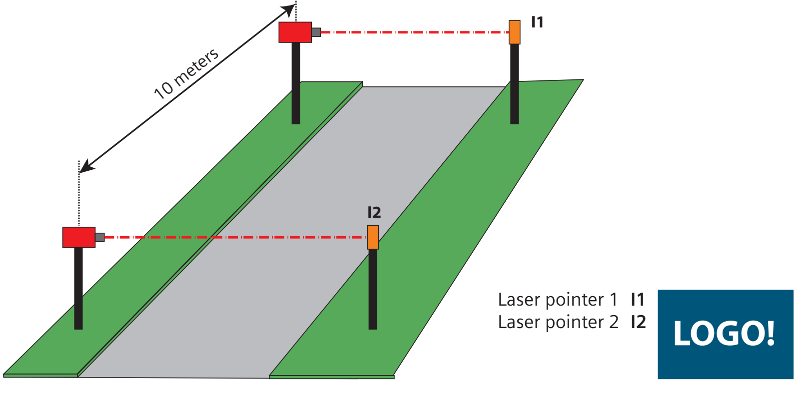


Fig. 6.47 A simple “radar” system

Table 6.15 Content and functions of the "Speed measurement" project

Content	Blocks used in the program
✓ Speed detection ✓ Displays on the internal and external text display ✓ Activation of external circuits ✓ Displays on a 7-segment display ✓ Making a sensor circuit	✓ Digital inputs and outputs ✓ Digital outputs ✓ Pulse relay ✓ Incremental encoder ✓ Up/down counter ✓ Message text

The central basis of the system is the combination of two laser pointers and two light-dependent transistors. Each of the laser pointers is secured to a stand or a pole. The poles are set up at a distance of 10 m on the curb or the desired measurement route. As receivers, on two additional poles the photo-transmitters are mounted. Each of them reacts to the briefest of light pulses and activates a dark circuit. These "receivers" are set up in such a way that they are positioned exactly across from the pointers. The fine adjustment of the system requires finesse (the pointers must meet the transistor).

In order to be able to connect the transistor to the LOGO! so that an analyzable signal can be used, the dark circuit shown in Fig. 6.48 is made from very few components.

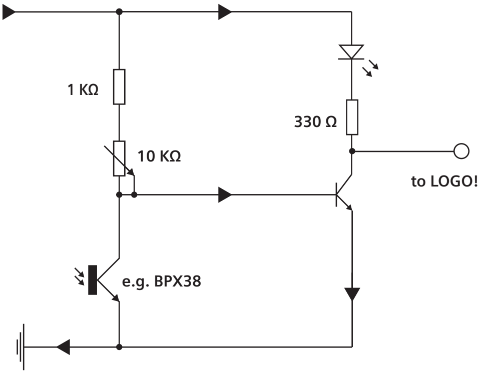


Fig. 6.48 Circuit diagram of dark circuit

On the sensor side, this consists of a series resistor for protecting the transistor, a potentiometer and the photo-transistor. The signal of the photo-transistor is amplified by an additional transistor and then output to the LOGO!. An LED with a series resistor is connected in the load circuit of the transistor for controlling the output signal.

Input I1 is activated either by the interruption of the laser beam or by the pulse sent by the dark circuit. When connecting the receiver circuit, you must ensure that the collector of the transistor is connected to the analog input 7 or 8 of the LOGO!. The pulse activates pulse relay 1 and holds itself. A clock generator is also activated. This sends a pulse every 200 ms. For the simulation mode, this should be set to one second for a better overview. Several counter components now count the pulses every 200 ms (or every second).

A counter block is used per desired time unit for this purpose, i.e. if the counting should go from 0 to 50 km/h in 5 km/h steps, then 10 counters are used. The switch-on time of the individual counters is set according to Table 6.16.

Table 6.16 Switch-on parameters for the counters

Speed	Time for 10 m	Switch ON parameter counter	Digit of the 7-segment display
50 km/h / 31.2 mph	0.72 sec	3.5 (3)	1
45 km/h / 28.1 mph	0.8 sec	4	2
40 km/h / 25.0 mph	0.9 sec	4.5 (5)	3
35 km/h / 21.9 mph	1.02 sec	5.1 (5)	4
30 km/h / 18.7 mph	1.2 sec	6	5
25 km/h / 15.6 mph	1.44 sec	7.2 (7)	6
20 km/h / 12.5 mph	1.8 sec	9	7
15 km/h / 9.3 mph	2 sec	10	8
10 km/h / 6.2 mph	4 sec	20	9
5 km/h / 3.1 mph	7.2 sec	36	0
1 km/h corresponds to 0.624 mph			

For example, if a speed of 20 km/h (12.5 mph) is to be determined, the switch-on parameter of the counter block "20 km/h" must be set to 9 according to the formula $v = s/t$ and by the conversion of m/s to km/h for a pulse of 0.2 seconds. The process is similar for all other speeds. The message text outputs are then connected directly to the counter blocks. These must be arranged by priority, so that the last triggered value is also retained on the display.

If the system is to be used in several pass-throughs, it is possible to label the first counter with priority "0" and the message text "System ready for measurement". During operation, the counter blocks are decreased by one until the second light barrier is interrupted and I2 is activated via the second dark circuit. During this, the counter block with the highest priority is output on the display.

For displaying the speed, the display of the LOGO! or the external TD (text display) (see Chapter 4.6) can be used (Fig. 6.49).

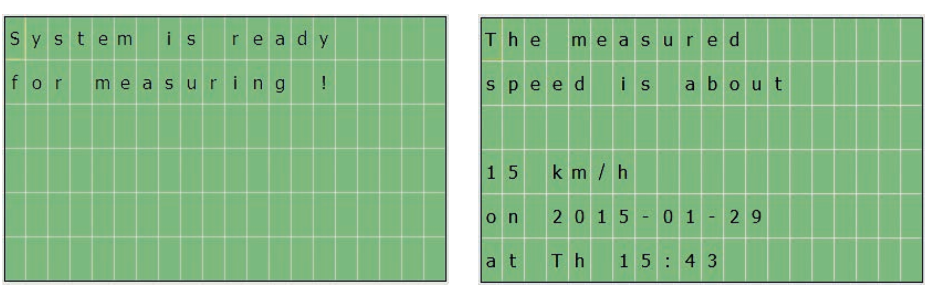


Fig. 6.49 Outputs on the display

The results display can be reset by activating the reset inputs via an external pushbutton on the "Pulse relays 2 through 5" (Fig. 6.50).

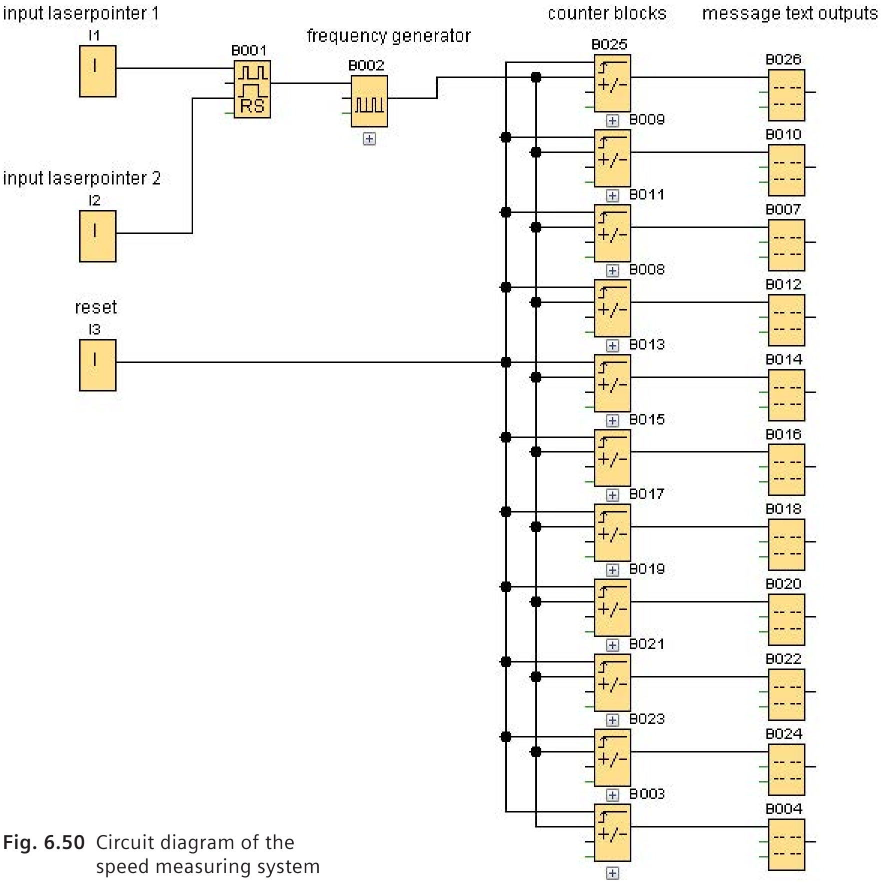


Fig. 6.50 Circuit diagram of the speed measuring system

Another interesting option for displaying the measured results is to connect simple bulbs or LEDs to the LOGO!. However, since the LOGO! only has a limited number of outputs, the information is displayed in BCD code. Table 6.17 helps with decoding. It is more convenient to display the results with a downstream 7-segment display.

To do this, the frequency of the clock pulse generator must be counted in four counter blocks and be converted into a BCD signal. The counter blocks are connected to outputs Q1 to Q4. These outputs can be connected directly to lamps or to the inputs of a BCD-to-7-segment decoding block (74LS47) (see Fig. 6.51). Its outputs are connected to a 7-segment display, which then outputs the interval range of the speed of the vehicle in the form of a numeric value from 0 to 9. When connecting the 74LS47 and the 7-segment display, you must ensure that the necessary voltages are adapted via corresponding serial resistors, for example.

Table 6.17 BCD code assignment

A4	A3	A2	A1	Decimal output of the 7-segment display	Speed
0	0	0	0	1	50 km/h / 31.2 mph
0	0	0	1	2	45 km/h / 28.1 mph
0	0	1	0	3	40 km/h / 25.0 mph
0	0	1	1	4	35 km/h / 21.9 mph
0	1	0	0	5	30 km/h / 18.7 mph
0	1	0	1	6	25 km/h / 15.6 mph
0	1	1	0	7	20 km/h / 12.5 mph
0	1	1	1	8	15 km/h / 9.3 mph
1	0	0	0	9	10 km/h / 6.2 mph
1	0	0	1	0	5 km/h / 3.1 mph
1 km/h corresponds to 0.624 mph					

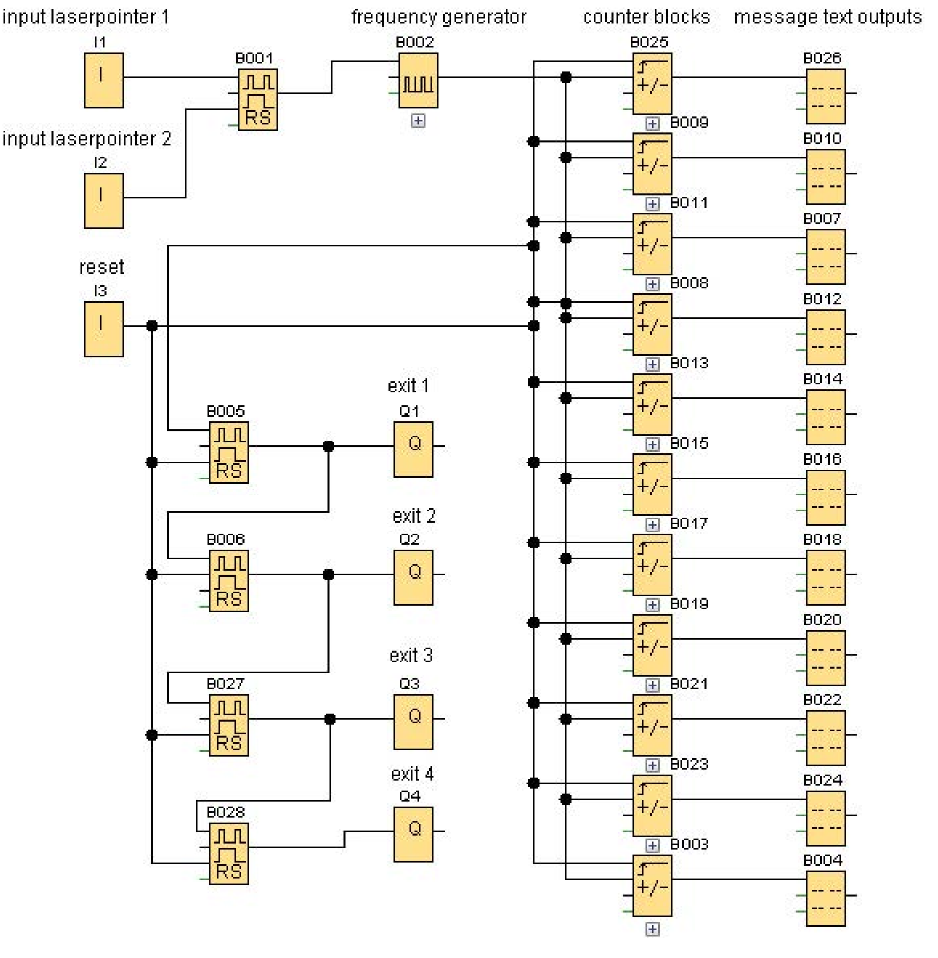


Fig. 6.52 Expansion of the outputs for connecting a 7-segment display

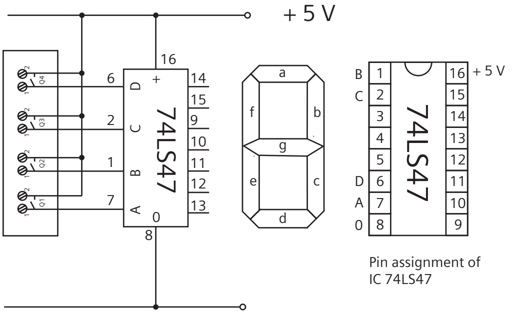


Fig. 6.51 Connection of the 7-segment display

Table 6.18 Function table of the "Radar time measurement" project

Block name	Property	Function in the program
I1	Digital input	Input for dark circuit 1
I2	Digital input	Input for dark circuit 2
I3	Digital input	Reset
B001	Pulse relay	Set the counters and terminate the counting process
B002	Incremental encoder	Output the 0.2 sec (1 sec) pulses for the counters
B003 – B011	Up/down counter	Count the signals of the incremental encoder/output of message texts
B012 – B021	Message text	Speed output
B022 – B025	Pulse relay	Counting in the BCD code
Q1 – Q4	Outputs	Connection of the 7-segment display

Tipp

When performing the fine calibration of the laser beam, it is helpful to put a sheet of white paper behind the photo-transistor. This makes the beam visible and the transmitter can be set more accurately.

Commercially available, low-cost laser pointers are suitable for use as the laser. With a simple pointer, however, it is only possible to replace the operating switch with an external switch with some difficulty. Online, cost-effective laser diodes without switch can be purchased, but only with loose wire ends.

To prevent external interference due to sunlight or other external light source, a dark transparent film or glass from an old pair of sunglasses can be glued in front of the photo-transistors.

6.9 Controlling a photovoltaic system

Photovoltaic systems have almost become the standard for new buildings. However, the market for upgrade components for old residences supplies a wide variety of products for expanding existing components. We are presenting a simple but effective do-it-yourself solution for solar hot water usage with readily available components (Fig. 6.53).

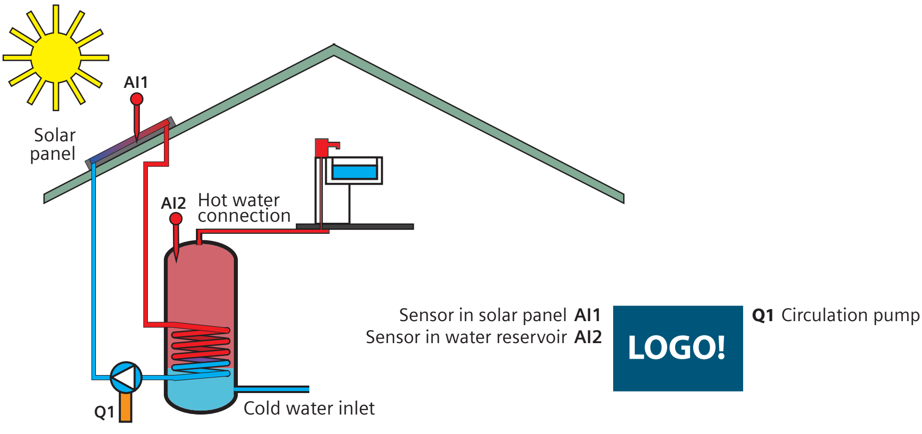


Fig. 6.53 Structure of a photovoltaic system

Table 6.19 Content and functions of the “Photovoltaic system” project

Content	Blocks used in the program
✓ Recording of temperatures ✓ Comparison of temperatures ✓ Display outputs by priority ✓ Display outputs following manual entry	✓ Analog inputs ✓ TD function key ✓ Digital output ✓ Runtime meter ✓ Analog comparator ✓ Message text ✓ Analog threshold switch ✓ Basic functions and logic gate ✓ Bit memory

A photovoltaic system for generating hot water essentially consists of one or more solar collectors (hot water generators), a water pump with water lines, a boiler for storing the hot water, an extension vessel, and the control system. The first components mentioned are available at low cost in hardware stores or plumbing stores and a LOGO! is well-suited for controlling the system.

A temperature sensor is positioned in a solar collector. Sockets are usually available in the collectors for this purpose. An additional sensor is positioned in the hot water tank. There are ferrule sleeves in the tank for this too. The sensors can be slid into the sleeves. Both sensors are connected to an analog input of the LOGO!. The inputs are compared using an analog comparator. The digital output for controlling the water pump is switched on and off depending on a defined difference $A_x - A_y$ and two parameterizable thresholds.

The internal text display of LOGO! is used to monitor the status of the system. An external display, which can be integrated into the structure of the house, is better suited. The temperatures in the solar cell and in the storage unit are displayed as basic functions (Fig. 6.54). The display is set up in such a way that the values are permanently displayed and they also remain visible when the temperature decreases. To do this, the field "Acknowledge message" must be activated in the menu field of the display output.

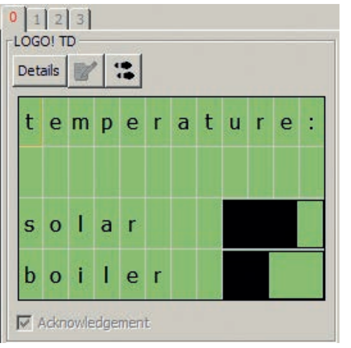


Fig. 6.54 Output of the temperatures in the system

One important safety feature of this system is the display of a warning message in the event of overheating or overtemperature in the solar module or in the hot water reservoir (Fig. 6.55). A corresponding message prompts the user to remove hot water from the system (e.g. to open the cock for hot water). However, the standard overpressure vessel should protect pipes in the system from bursting when steam builds up in the system. Very large-scale systems, so-called "drain-back systems", will even withdraw all of the water from the solar modules when a specific temperature is reached in the hot water reservoir.

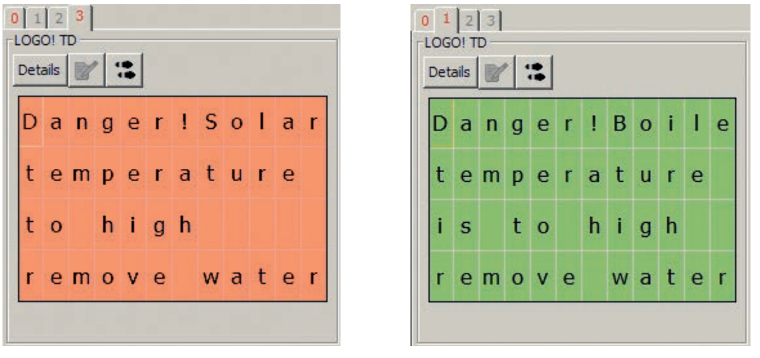


Fig. 6.55 Displays of warning messages for solar module (left) and water reservoir (right)

A digital pushbutton can be installed in the circuit for completing the control capability. By pushing this the runtimes of the solar pump since the system was commissioned and since the beginning of the day's pumping can be displayed (Fig. 6.56). As desired, a permanent or an instantaneous display can then be called up via a switch on the TD or an external switching button next to the display.

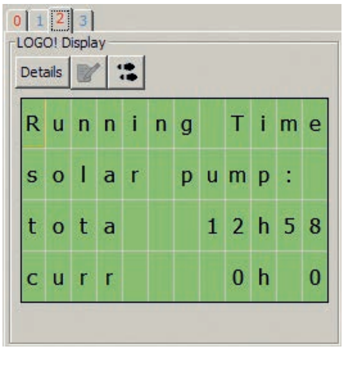


Fig. 6.56 Output of the runtimes of the system

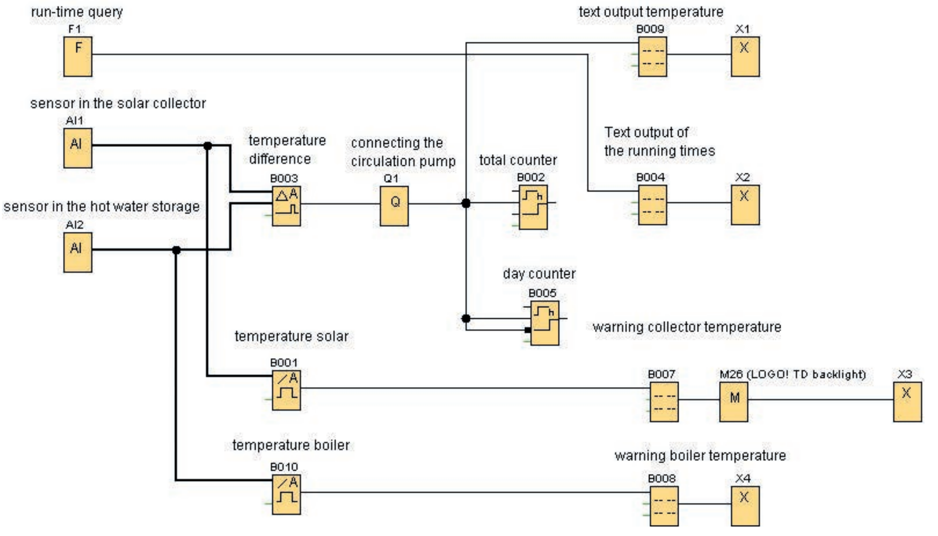


Fig. 6.57 Circuit diagram of the photovoltaic system controller

Table 6.20 Function table of the "Photovoltaic system" project

Block name	Property	Function in the program
A1	Analog input	Temperature sensor in the solar panel
A2	Analog input	Temperature sensor in the heat storage boiler
F1	TD input	Pushbutton for output of the display
Q1	Output	Connection for the motor of the water pump
B001	Analog threshold switch	PT100: Temperature in the solar panel
B010	Analog threshold switch	PT100: Temperature in the boiler
B003	Analog comparator	Temperature difference
B002	Runtime meter	Recording of the runtime of the pump since commissioning
B005	Runtime meter	Recording of the current daily runtime of the pump
B006	Logic basic function	Negation for the reset input on the day counter
B004	Message text	Output of the runtime of the pump
B007	Message text	Warning message of the collector
B008	Message text	Warning message of the boiler
B009	Message text	Output of the current temperatures
M026	Bit memory	Color change in the text display

Tip

The system can be expanded by an additional output, which opens an emergency valve when a set temperature is exceeded and thus, the pressure in the system is reduced. As a rule, however, a sufficiently dimensioned expansion vessel is sufficient.

6.10 Generating hot water using a wood stove

More and more private households have wood stoves. These disperse pleasant heat and light in a room. In most cases, however, a large part of the heat energy that is generated by, for example, wood, charcoal briquettes or pellets, is not used. A considerable part of the energy is lost through the chimney due to the high exhaust gas temperatures.

In modern stoves, a part of the generated heat energy and a part of the exhaust temperature are also converted to energy via water-filled heat exchangers. This can then be used in an existing heating or photovoltaic system in the home. In addition to the reduced power loss of these systems, there is also the advantage that gas or oil heating can be completely dispensed with, particularly in the transitional periods. We present a simple combination of solar hot water usage with a solid fuel system (Fig. 6.58).

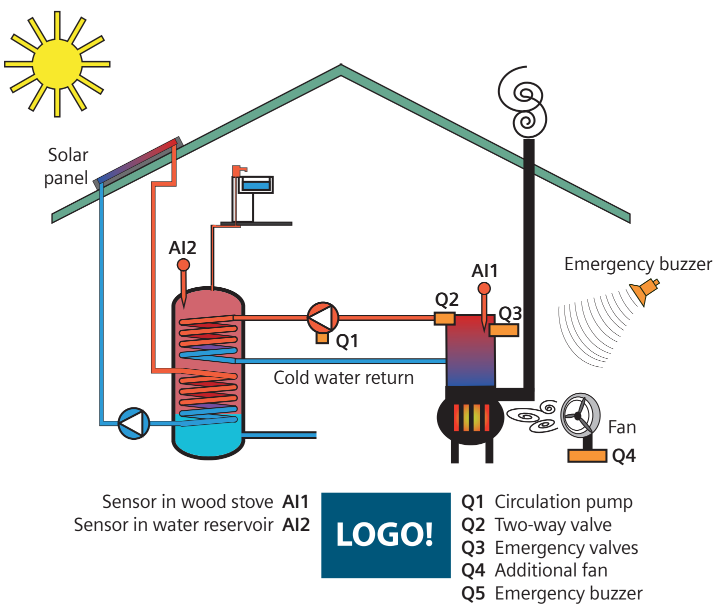


Fig. 6.58 Structure of a heating system with water-filled heat exchanger

Table 6.21 Content and functions of the "Wood stove" project

Content	Blocks used in the program
✓ Recording of temperatures	✓ Digital and analog inputs
✓ Comparison of temperatures	✓ Analog threshold switch
✓ Display outputs by priority	✓ Basic functions and logic gate
✓ Controlling of valves	✓ Message text
✓ Controlling a fan	✓ Digital output
✓ Controlling an acoustic output	✓ TD function keys

A heating system for heating up process water or heating water primarily consists of:

- the heater with integrated water-filled heat exchanger,
- circulation pump with water lines,
- emergency and control valve,
- boiler for storing the hot water,
- expansion vessel and
- control system.

The first three components can be procured at low cost in the hardware or plumbing stores. The storage unit and expansion vessel can generally be used by an existing photovoltaic system. The control of the system is implemented using LOGO!.

During the planning of the system, you must keep in mind that the existing water reservoir needs an additional heating circuit for the water-filled heat exchanger. In many boilers this is already integrated. Otherwise, the existing solar line must be expanded by a heating circuit by adding valves. Since the heating systems are generally delivered with connection diagrams, the system can be installed by an experienced amateur by hard soldering or crimped connections.

Each stove with a water-filled heat exchanger has a connecting socket into which a temperature sensor can be integrated. An additional sensor must be positioned in the boiler or the existing sensor of the photovoltaic system may be used (see circuit diagram in Fig. 6.59).

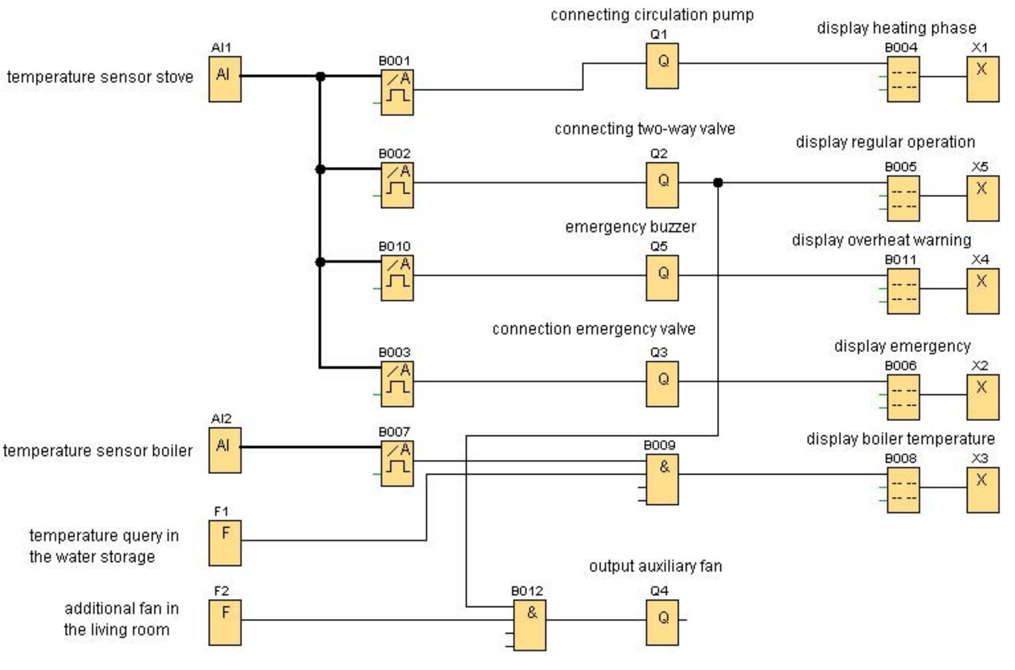


Fig. 6.59 Circuit diagram of the control system of a wood stove

The circulation pump is started via an analog threshold switch beyond an operating temperature in the oven of approx. 30 °C. Beyond an operating temperature of 50 °C, the outlet of the two-way valve is opened via an analog threshold switch in order to begin with the heat exchange in the boiler (message text, see Fig. 6.60). It is important that the stove only begins with the heat exchange beyond 50 °C because otherwise there would be a risk of sooting up. The user can be notified of the function and status of the system at an appropriate location, e.g. in the living room, via a display output. After an increase in the temperature of the system, e.g. to 90 °C, the user is warned via another analog threshold switch by a corresponding output on the display (Fig. 6.61). An output, to which a buzzer is connected, is integrated for this purpose. If a critical temperature is exceeded, two emergency valves are opened and hot water is drained or it is topped off with cold water.

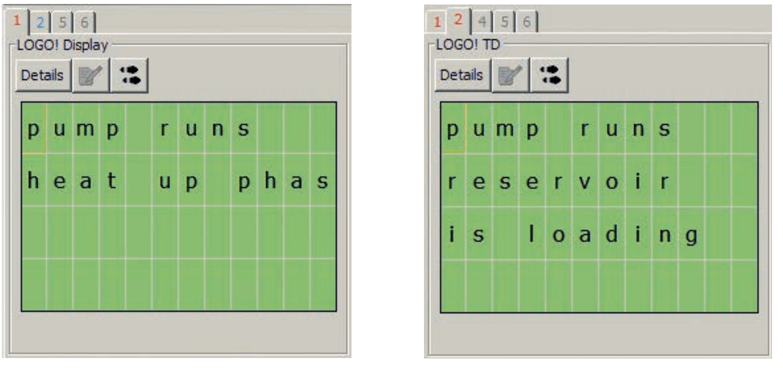


Fig. 6.60 Outputs of the system status

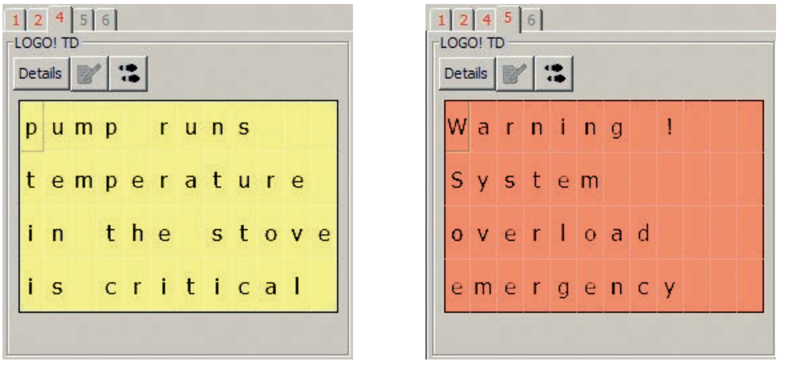


Fig. 6.61 Outputs of the emergency function

A digital input or a function key on the display, which can query the boiler temperature, is used for manually controlling the system (Fig. 6.62). If need be, an additional digital input is used for manually activating a fan under or behind the oven. This has proven its worth in practice, since the room often has to be warmed up first when the system is started and only then should the water of the heating system be heated.

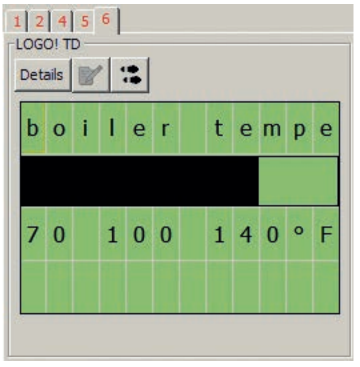


Fig. 6.62 Output of the boiler temperature

Table 6.22 Function table of the "Wood stove" project

Block name	Property	Function in the program
A1/A2	Analog inputs	Temperature sensor in the stove/boiler
F1	LOGO! TD function key	Temperature query in the storage unit
F2	LOGO! TD function key	Manual switching of an additional fan
B001	Analog threshold switch	Output signal for the circulation pump
B002	Analog threshold switch	Output signal for the two-way valve
B010	Analog threshold switch	Output signal for the emergency buzzer
B003	Analog threshold switch	Output signal for the emergency valve
B009	Logic basic function AND	Output signal for the temperature query
B012	Logic basic function AND	Output signal for the additional fan
Q1	Output	Motor for the circulation pump
Q2	Output	Output for the two-way valve
Q5	Output	Output for the emergency buzzer
Q3	Output	Output for the emergency valve
Q3	Output	Output for the additional fan
B004 – B006, B008 – B011	Message text	Messages for the operating state of the system
X1 – X5	Open terminal	End of message text

Tip

For less experienced builders, it is recommended to have a plumber do the installation or the piping of the system. At the highest points in the piping, venting valves and a shut-off valve for filling the system must be integrated into the water lines.

6.11 An (automobile) alarm system

Modern vehicles generally have simple alarm systems. If a door is opened or the ignition is activated and the deactivation of a contact via the remote control of the central lock is omitted, a time-limited, pulsating horn alarm is triggered. For older vehicles or for an individual solution, a comprehensive alarm system can also be self-developed and implemented. The following project shows the structure of a multi-stage alarm system with the aid of the LOGO! (Fig. 6.63). The alarm system can be implemented individually and also be provided with active safety functions as needed.

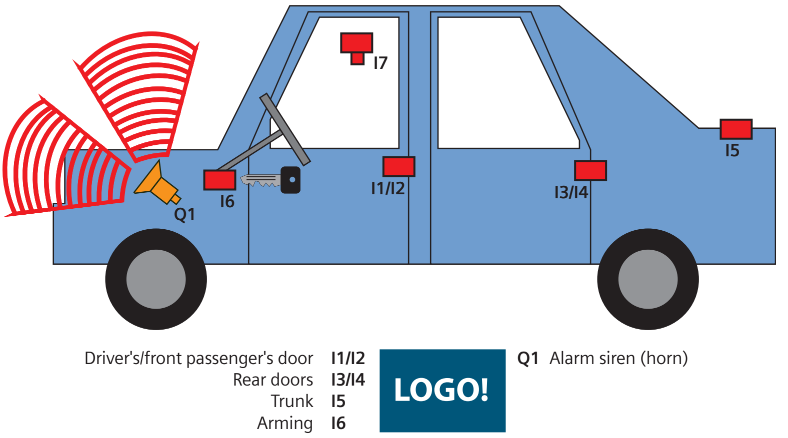


Fig. 6.63 Schematic diagram of a vehicle alarm system

Table 6.23 Content and functions of the "Alarm system" project

Content	Blocks used in the program
✓ Integration of negated contact switches ✓ Integration of a motion detector ✓ Time-delayed output of alarm signals ✓ Time-limited output of alarm signals ✓ Activation of various output relays for active components ✓ Output of message texts ✓ Integration of finished system components (ultrasonic module)	✓ Digital inputs ✓ Digital outputs ✓ Time delays ✓ Incremental encoder ✓ Latching relay ✓ Coupled bit memory ✓ Message text ✓ Sending an SMS notification

Every alarm system generally consists of:

- Sensors for recording an input signal,
- an electronic analysis of the received signals and
- of the activation of one or more actuators for outputting a warning signal.

The alarm circuit shown in the following uses simple contact switches (I1 to I5) for the possible alarm activation on the vehicle doors. Switches are used for this when programming the LOGO!. Before the circuit is installed in a vehicle, the switches in the software are replaced with pushbuttons that have an opening function. For a permanent installation of the system in a vehicle, the existing interior light contacts can be used with a little skill and dexterity. When a door is opened, these provide a positive signal which can be used as an input signal. For mobile applications of the alarm system (see tips), other switches must be used. These can be purchased ready-made or you can simply make them yourself, e.g. from 2-side adhesive tape, which is adhered to aluminum foil. The inputs are merged with one or more OR gates (B002/B004). If desired, the central arming of the system can be implemented via an additional external input. In our example, it is triggered via a negated 12 V signal from the ignition (I6), i.e. the system is deactivated when the ignition is switched on and the system is activated when the motor is shut down. To manually arm the system and then be able to leave the vehicle, an additional switch and a time delay are needed. After activation of the switch I8, the time delay B010 is activated, which sends a positive pulse to an AND gate after a set time, e.g. one minute. The ignition signal (I6) and the OR gates (B001/B004) of the doors and windows are also connected to this gate (Fig. 6.64).

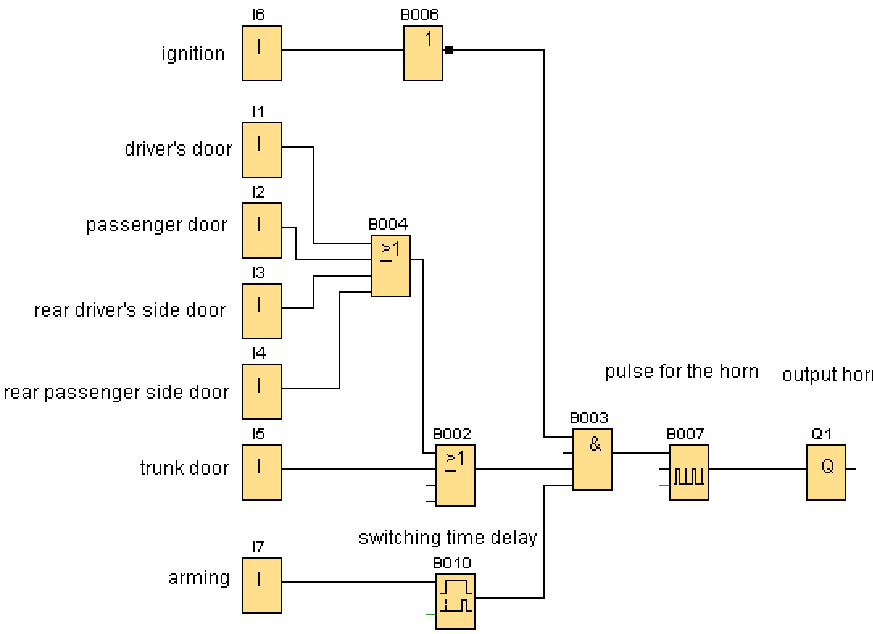


Fig. 6.64 Schematic diagram of a simple vehicle alarm system

The existing vehicle horn can be used as a simple actuator. To do this, the vehicle‘s horn is either connected directly to the output of B003, or for an even more penetrating warning signal, it is provided with an incremental encoder B007, for which a horn interval of 2 seconds is set. The disadvantage of this solution is that the alarm signal can be shut off when the triggered contact is closed again, for example, by closing the opened door again.

To bypass this, a bit memory, a latching relay (B005), is integrated into the circuit (see Expanding the system, Fig. 6.65). If this is set, the system can no longer be deactivated by closing a door. In addition, the system can be shut down via an additional time delay (B001) if it is accidentally initiated. To ensure that the system is automatically deactivated after a specific period of time, the reset input of the latching relay must be activated after a desired period of time. This takes place via an additional latching relay (B008), which is needed by the incremental encoder, and an ON delay (B009).

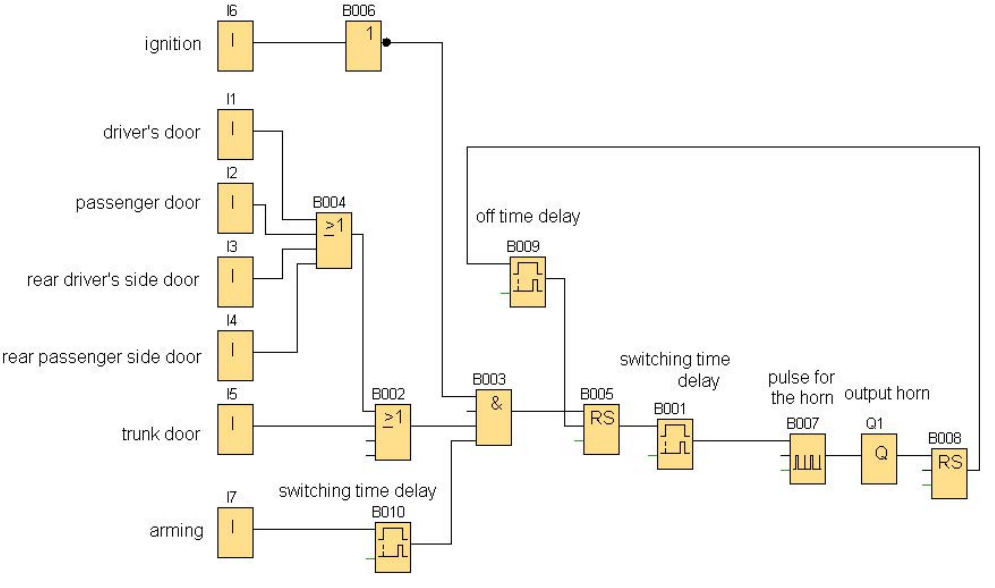


Fig. 6.65 Expansion of the system: automated switch-off and pulsed alarm output

It is recommended that you automate the arming in order to allow the system to run autonomously. To this end, input I7 for the arming is removed and replaced with several memory blocks. Combining B011 and B013 allows the first signal, which is triggered by exiting the vehicle, to be temporarily saved and to only cause an alarm message if the second signal is triggered. To ensure that the system can be deactivated again even without its own switch, it is necessary to connect the ignition signal to the previously described latching relay B005 via an OR gate (B010).

An interior monitoring system can be easily integrated into the system in order to provide more security for a convertible or other open vehicle (Fig. 6.66). For this purpose, an ultrasonic module can be connected to an additional input of the LOGO!. An ultrasonic system generally consists of a transmitter and a receiver. In principle, it would be possible to connect the transmitter and the receiver to the LOGO! and to conduct the analysis of the information with the system as well.

For less experienced information engineers, however, it is recommended that a preassembled block with a transmitter, receiver, and analysis electronics be used. This must be separately supplied with 12 V via an output of the LOGO!

(Q2) and the output signal of the block must be connected to an additional input (I7). When doing this, you must observe whether you are dealing with an analog or digital output signal (analog or digital input of the LOGO!). Input Q2 and the ultrasonic signal of I7 are activated via a signal of the first latching relay B011. We have used a preassembled ultrasonic system when implementing the circuit.

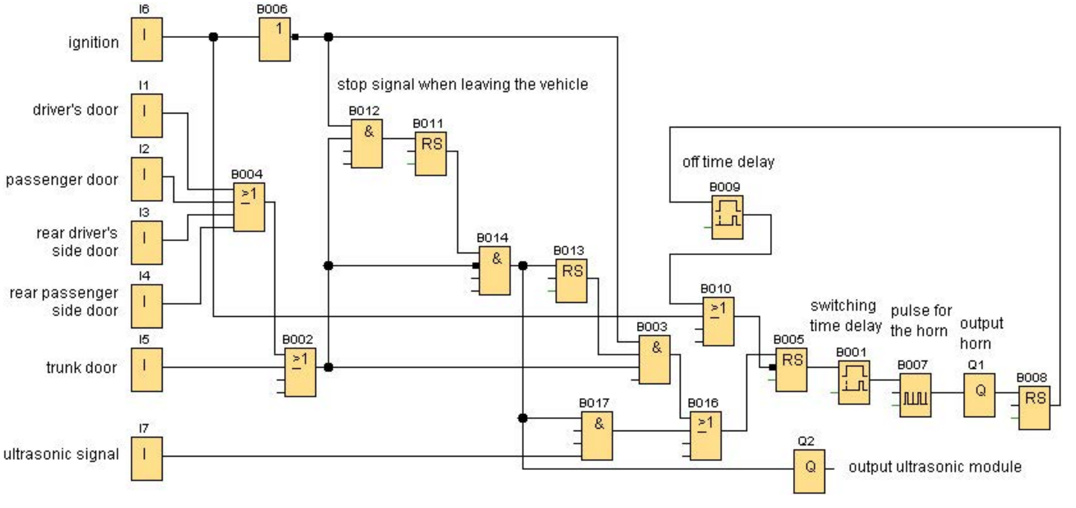


Fig. 6.66 Expansion of the system: Switch-off function and interior monitoring

Lastly, various passive and active theft prevention options are available using the still available outputs of the LOGO!. To this end, additional outputs Q3, etc. are connected to the signal of the ON delay B001.

One option for passively securing an object is to interrupt the ignition circuit via an output (Q3) of the LOGO!. To do this, the ignition signal of the vehicle must be searched for and interrupted. Alternatively, the relay of the starter could also be interrupted.

Active alarm systems can also send the output of an active message text to an external receiver. This function can be implemented with the SMS module (see Chapter 3.6). To this end, a message text (B019) is generated via output Q4 and forwarded to the SMS module. This then sends a text notification to any mobile telephone, which was saved with its call number (Fig. 6.68).

Fig. 6.67 Active security system

Another function, to be used with caution, is the integration of a CS gas actuator. To do this, output Q5 is connected to a strong lifting magnet via an AND gate and an additional safety switch (I8). This is mounted inside a lockable housing on a CS gas cartridge in such a way that the anchor triggers the CS cartridge when there is movement (Fig. 6.67).

At this point, we expressly point out that the use of this system must be comply with local safety regulations or road traffic regulations!

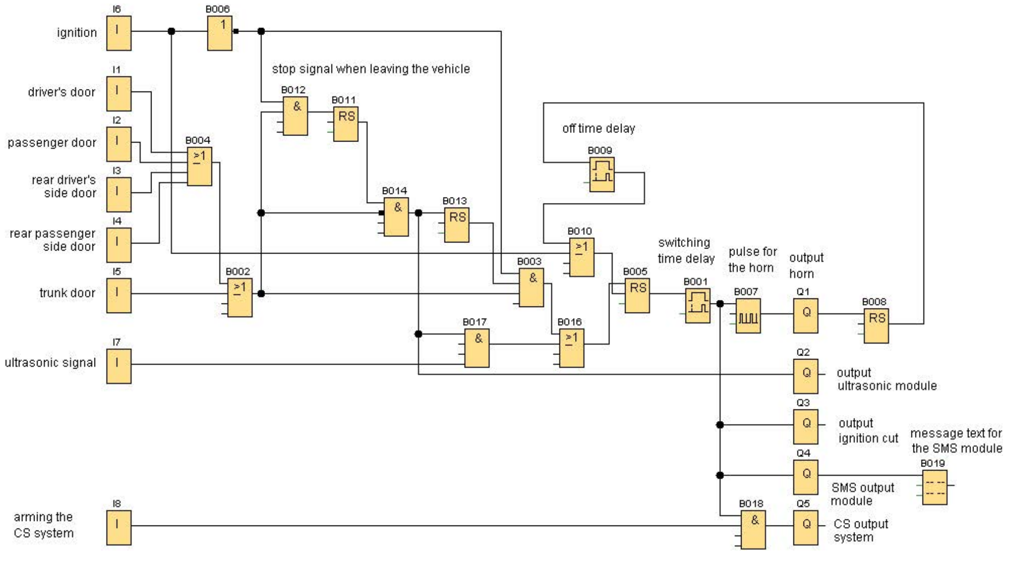


Fig. 6.68 Expansion of the system: active security systems

Table 6.24 Function table of the "Alarm system" project

Block name	Property	Function in the program
I1 – I5	Digital inputs	Contact switches in the vehicle doors
I6, I7	Digital input	Ignition input. Ultrasonic sensor
I8	Digital input	Safety switch of CS gas system
B006	Logic basic function NOT	Negation of the ignition signal
B002, B004	Logic basic function OR	Connects the input signals of the doors
B011	Latching relay	Hold signal when exiting the vehicle
B012, B014	Logic basic function AND	Connects the ignition signal to the door contacts/ and lock
B013	Latching relay	Hold signal when door is opened again
B003	Logic basic function AND	Combining the ignition/hold signal and the door contact
B005	Latching relay	Hold signal of the triggered alarm
B001	ON delay	Time delay for alarm triggering
B007	Incremental encoder	Frequency generator for the horn
Q1	Output	Warning signal for the vehicle horn

Block name	Property	Function in the program
B008	Latching relay	Signal holding of the pulsed horn signal
B009	ON delay	Time delay for switching off the alarm signal
B010	Logic basic function OR	Deactivates the latching relay for the incremental encoder
B017	Logic basic function AND	Connects the held door and ultrasonic signal
B016	Logic basic function OR	Connects ultrasonic solution and door contacts
Q2	Output	Output for the activation of the ultrasonic module
Q3	Output	Output for ignition interruption
Q4	Output	Output for a message text
B019	Message text	Text for an SMS message
Q5	Output	Output for the CS gas system
B018	Logic basic function AND	Connects alarm signal to additional safety switch

Tip

In the mobile application of the alarm system, a flexible power supply must naturally also be provided. The system can be operated via a cigarette lighter in a vehicle (if it is continuously positive), but it is not convenient to connect it with an additional cable. Therefore, the use of a 12 V battery is recommended, e.g. from a bicycle accessories store. When sizing a lifting magnet for triggering the CS cartridge, you must ensure that the possible power output of the battery is sufficient. A power outage leads to a shutdown of the LOGO! and a system crash.

Conrad Electronic offers a special ultrasonic learning package for approx € 15. The preassembled module (order no.: 10108) is well-suited for integration into the system.

If a CS gas actuator is used, it may make sense to integrate an additional time delay into the system. This limits the triggering of the gas to a short period of time (e.g. 5 seconds). Otherwise, the persons involved may be seriously injured.

6.12 Automated feeding machine for an aquarium or terrarium

A simple but effective “automated feeding machine” can automate the feeding of fish and animals in an aquarium or terrarium, for example, when you are away from home or on vacation. The “automated feeding machine” can also be turned into an intelligent flour or muesli dosing machine by undertaking corresponding conversion measures (Fig. 6.69).

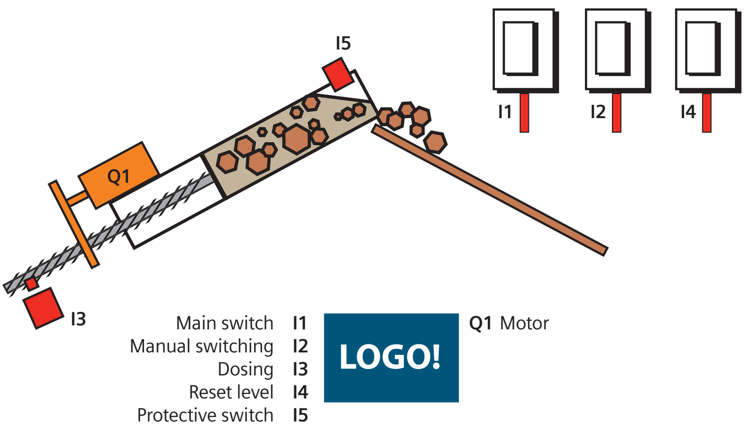


Fig. 6.69 Functions of a feeding station

Table 6.25 Content and functions of the “Automated feeding station” project

Content	Blocks used in the program
✓ Time-controlled automation ✓ Automatic measuring of quantities with the aid of the up/down counter ✓ Arithmetics for quantity checking/inventory checking ✓ Setting the counter block by means of cursor keys directly on the LOGO!	✓ Digital inputs and outputs ✓ Week time switch ✓ Up/down counter ✓ Message text ✓ Arithmetic instruction ✓ Analog threshold switch ✓ Basic functions/logic gate: AND/OR

A tube that is open at the top is available for implementing an “automated feeding machine”, which automatically ensures the correct portioning of the feed. A cover that is form-fitted to the tube is situated on the bottom. Secured to this cover is a threaded rod, which is driven/turned by a strong (geared) motor. The threaded rod must rest in a guide, which is firmly connected to the tube, so that the cover moves up, in the direction of the opening in the tube, when the threaded rod is turned. The feed is thus pushed up and out of the tube and can slide smoothly into the feeding dish via a chute.

A main switch (I1) is used to start the system for the “Automated feeding machine”. Once this is activated, the message text (B008) is displayed as in Fig. 6.70.

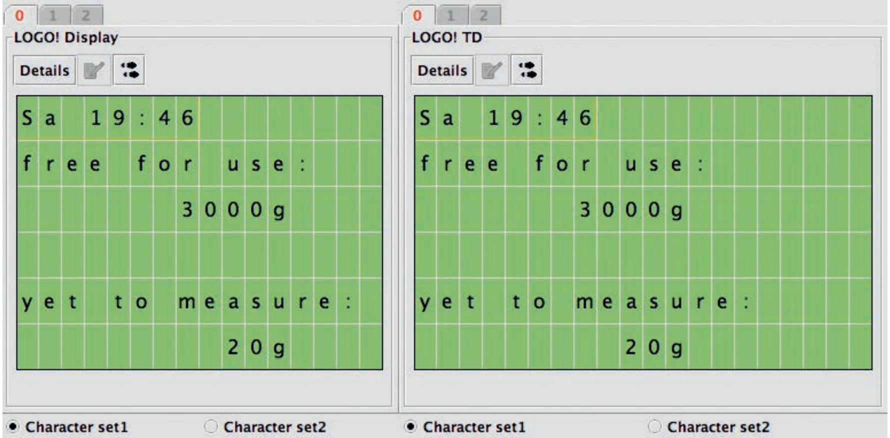


Fig. 6.70 Message text: Fill level overview on LOGO! display (left) or on external text display (right)

In the 1st row, the current day and time is displayed. In the 3rd row, the amount of feed remaining in the container is specified. In the 6th row you can see how much feed will be dosed in the next switching cycle.

The timer clock (B003) in the LOGO! allows the time-controlled, automatic dosing of a fixed quantity (a precise time for the feeding can be set). This is set in the block properties of B001, either directly on the LOGO!, after the program has been transferred, or alternatively in the program prior to the transfer.

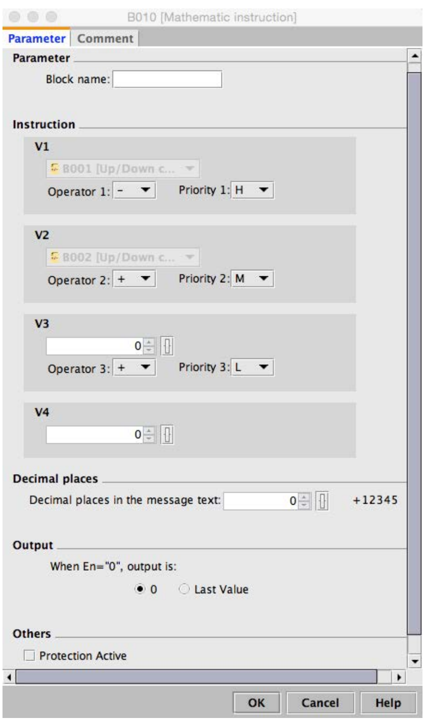


Fig. 6.71 Settings for arithmetic

Once the output B003 (the time for the feeding) is activated, the motor (Q1) that turns the threaded rod is switched on. In addition, a manual switch (I2) is installed in the circuit, which allows unscheduled feeding.

In counter block B001, the ON/OFF limit is set in such a way that the output of this counter is activated when a defined value is undershot and a warning: “Note: refill feed!” (message text B009) is displayed. We have set the ON limit for the up/down counter B001 to 0 and the OFF limit to 21 for this purpose: Once only 20 grams of feed remains in the container, the value is undershot. In addition, arithmetics (B010; Fig. 6.71), coupled with an analog threshold switch (B011; Fig. 6.72), are built into the program, which checks the difference of the actual value (here: 3000 g) with the dosing value (here: 20 g):

B001 – B002 ➔ Actual value – dosing value (cf. Fig. 6.71: V1 – V2).

The analog threshold switch (B011; Fig. 6.72) monitors the difference and activates a warning when a defined value is undershot (message text B013): “Warning! No more feed!”. The threshold switch (B011) is thus a protective device when there is too little feed in the container for the amount to be dosed. It is set as in Fig. 6.72.

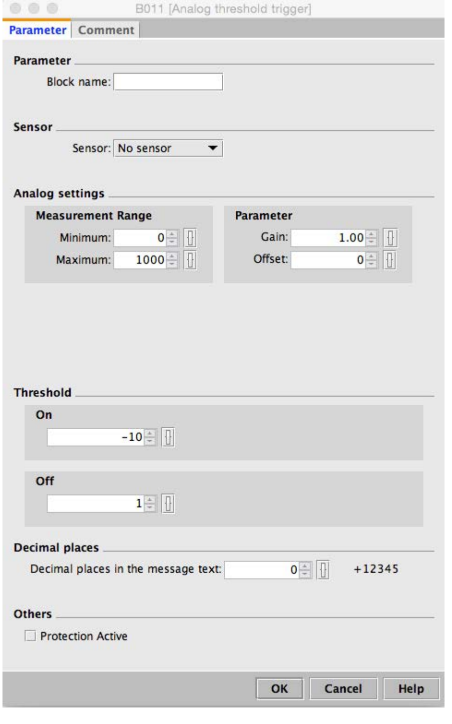


Fig. 6.72 Threshold switch settings

As a final protection, a sensor is also attached to the housing of the feed container (I5). It is mechanically triggered once the container no longer has any feed.

A sensor is connected at digital input I3 for determining the level of feed. It detects how much feed has already been dosed. This can be determined, for example, by the threaded rod sending one pulse per rotation to I3 e.g. by a magnet and a reed switch. Now, it only has to be determined how many grams of feed is dosed per rotation of the threaded rod, i.e. how much feed is pushed upward out of the opening of the container per rotation. If the feed container is refilled, the actual value can be reset (by activating I4). Fig. 6.73 shows the overall circuit.

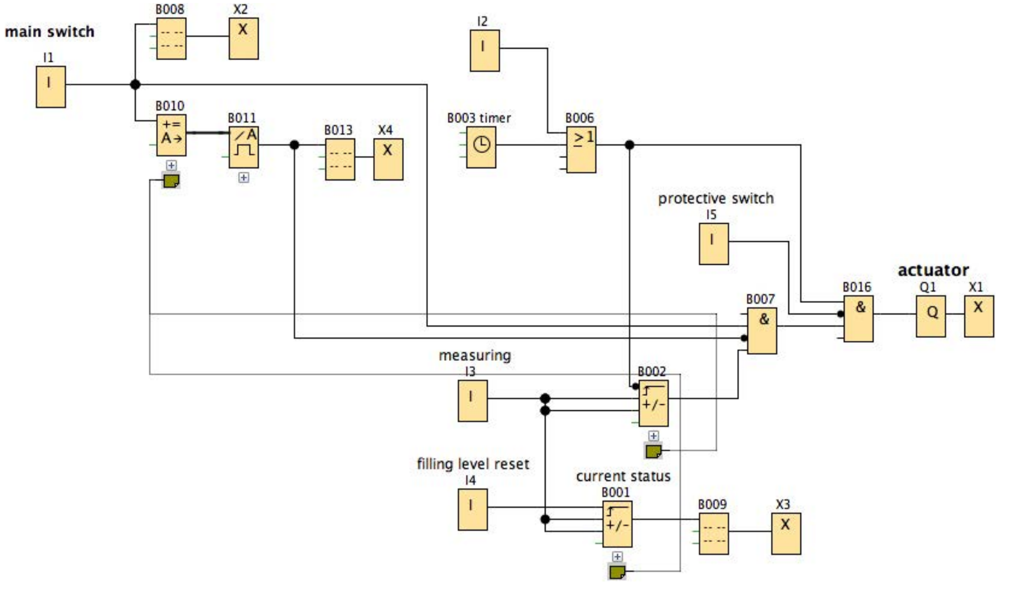


Fig. 6.73 Overall circuit for the “automated feeding machine”

Table 6.26 Function table of the “Automated feeding machine” project

Block name	Property	Function in the program
I1	Digital input	Main switch
I2	Digital input	Manual switching
I3	Digital input	Sensor for dosing checking
I4	Digital input	Reset: Level
I5	Digital input	Protective switch: level empty
Q1	Output	Actuating elements: Motor for feed dosing
X1 – X4	Open terminals	
B001	Up/down counter	Actual value: Level actual value
B002	Up/down counter	Value to be measured off
B003	Week time switch	Timer clock with defined feeding times
B006	Logic basic function	Activation by timer clock or manually
B007	Logic basic function AND	Safeguarding through diverse conditions
B008	Message text	Main menu: overview and monitoring

Block name	Property	Function in the program
B009	Message text	Low level warning
B010	Arithmetic instruction	Checking of setpoint and actual value
B011	Analog threshold switch	Warning if actual value is undershot
B013	Message text	Insufficient level warning
B016	Logic basic function AND	Safeguarding through diverse conditions

Fig. 6.74 Conveying screw

> **Tip**
> Instead of the threaded rod and a form-fitted cover, which push the feed up and out of the opening, you could also purchase a conveying screw (Fig. 6.74) or even make one yourself.

6.13 Telemetry for a vacation home

Owners of vacation apartments, RVs or camps know that their property is often unprotected. A telemetry system can replace an expensive caretaker or security service. With little programming effort, a complex remote monitoring system can be configured. Depending on individual needs, the system can be adapted to the local conditions or to the monitoring needs of the user.

Since a permanent Internet connection in the property to be monitored is only found in a few cases, the new GSM connection of the LOGO! is excellent for the cost-effective transfer of information to the home PC or smartphone (Fig. 6.75).

Our project comprises two parts. The first part describes a signaling system for an alarm system in which the alarm message is sent both actively and passively. In the second part, various parameters of the property are queried.

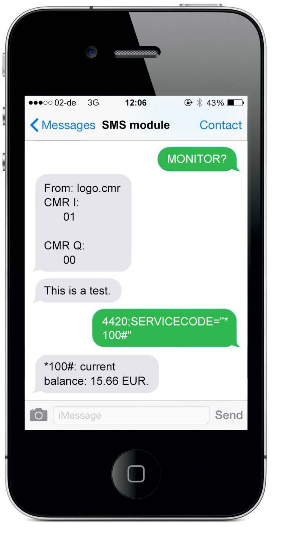


Fig. 6.75 Alarm message from a monitored object to the mobile phone

This is adapted to corresponding deviations from a desired value or can be queried and adapted remotely.

Table 6.27 Content and functions of the "Telemetry" project

Content	Blocks used in the program
✓ Recording of several digital signals	✓ Digital and analog inputs
✓ Time-dependent switching on/off	✓ Basic functions and logic gate
✓ Repeated switching	✓ Digital output
✓ Output of message texts	✓ Pulse relay and latching relay
✓ Output of SMS notifications	✓ Counter
✓ Remote switching	✓ Colored display text
	✓ TD display keys

6.13.1 Subproject: Access point protection

The most dangerous points of a property to be protected are the access points. Therefore, the first part of the circuit is designed in such a way that door and window contacts are installed in an alarm system. They are connected to the digital inputs I1 and I2. The incoming signal is sent via an OR (B001) to a pulse relay (B002) and from there to an ON/OFF delay. This can be set in such a way that the system can be disarmed when entering the room.

Further the duration of the alarm signal can be specified at Q1. This should not be activated for longer than one minute depending on the signal encoder that is connected.

To ensure a more effective alarm signal during a break-in, it is recommended that the signal be activated several times in succession. To do this, the output signal of Q1 is sent to a latching relay (B009) and then back to the OR (B001) via a clock generator (B008). The clock generator is required to generate an edge signal and to specify the renewed switch-on time. Depending on the desired time, the clock generator can be set to a frequency of one minute.

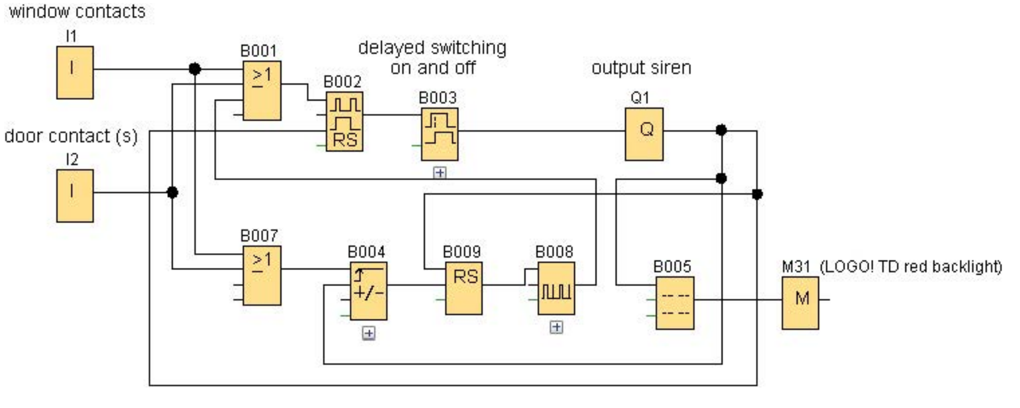


Fig. 6.76 Alarm message via contact switch

To ensure that the system does not sound the alarm forever, the desired alarm frequency can be set via counter B008. In this case, it is recommended that you let the alarm signal repeat 4 to 5 times. The counter is triggered via the signal of output Q1 and is reset via a renewed triggering of the door and/or window contacts. After the counting procedure is finished, the signal of the counter is used to reset the latching relay B009.

If need be, the triggered signal can also be reset via an additional pushbutton (I3). To this end, this is connected to the reset inputs of the pulse relay B002 and latching relay B009 via two additional OR blocks. A main switch in the system, e.g. a key switch for activating the system, replaces this function however.

To display the alarm message on a display via the net or via the SMS function of the LOGO!, a message text (B005) is connected to output Q1. An additional bit memory (M31) connected in series allows the display color of the message text to be changed to red, for example (Fig. 6.76).

6.13.2 Subproject: Frost and humidity monitoring

In the second part of the circuit, the automatic object monitoring is expanded by:

- the interior monitoring functions
- a connected frost monitor and
- humidity detection with a connected air de-humidifier.

As sensors a component with both functions (humidity and temperature sensor) can be used[16]. Please note that there are analog and digital sensors and that the circuit must be adapted accordingly. We have used analog components in our example.

For the function of the frost monitor, a temperature signal from a temperature sensor is analyzed via an analog input A1 from an analog threshold switch B006. With a correspondingly set value, output Q2 is activated and can switch a heating fan. This ensures that water lines, for example, do not freeze (Fig. 6.77).

The function of the humidity sensor is similar[17]. The analog signal of a humidity sensor is analyzed via an analog input A2 by an analog threshold switch B010. With a correspondingly set value, output Q3 is activated and can activate an air de-humidifier. This ensures that the object is reliably protected against the formation of mold, for example. Since simultaneous operation of the devices is not necessary, the outputs are connected via an AND with a negated input. This allows the frost monitor (heating fan) to work autonomously and the air dehumidifier to only commence operating when the frost monitor is deactivated.

If need be, the system can also be expanded by a smoke detector. This function only makes sense if a telemetric output is to take place. To do this, a preassembled signaling device can be modified in such a way that the signal of the Piezo buzzer is used as input signal for the LOGO!. This can then be output via a message text via the webserver.

[16] For example, available for approx. € 35 from conrad.com, order number 503492-62
[17] For example, available for approx. € 3 from conrad.com, order number 183324

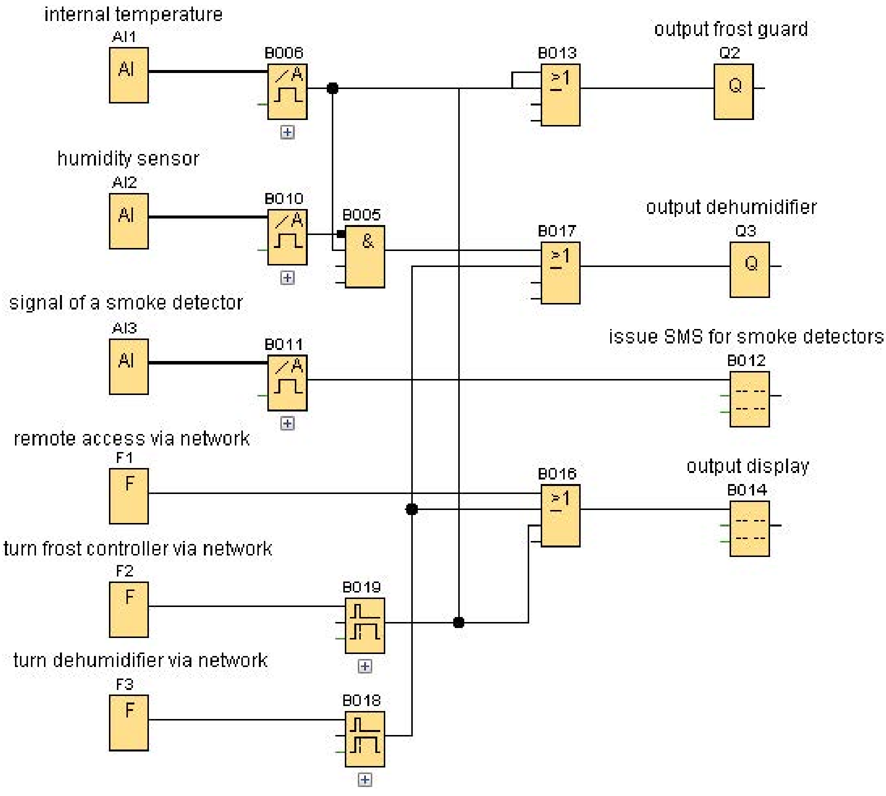


Fig. 6.77 Schematic diagram of the internal monitoring functions

6.13.3 Additional capabilities of the remote display and remote control

If the system is operated via a Local Area Network (LAN) or SMS, it makes sense to integrate the remote display and remote control options. To display a triggered alarm, a message text display is connected to output Q1. A text message is sent when the alarm is triggered. The current values of the object can also be queried remotely: To this end, a function key F1 is integrated into the circuit, which can also be operated by the software for a remote query or on the device itself. If the key is activated, the temperature and humidity data can be queried via a bar diagram, via an ON/OFF display for the window and door, and via an ON/OFF display for frost protection and dehumidifier devices. Depending on the arrangement desired on the display(s), the text lines can be a ticker.

Two additional function keys F2 and F3 allow the devices to be remotely activated via SMS or LAN for air dehumidifying or for frost monitoring. To ensure that the user does not forget to deactivate the devices during manual switching, it is recommended that you integrate the OFF delays B018 and B019. These are connected to the corresponding outputs Q2 and Q3, and via an OR gate with the message text output. Thus, the user gets a status message immediately after activating the devices remotely (Fig. 6.78).

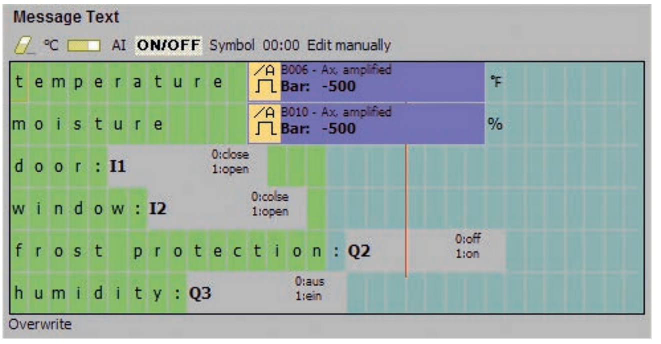


Fig. 6.78 Programming of the output display

Table 6.28 Function table of the "Telemetry" project

Block name	Property	Function in the program
I1, I2	Digital inputs	Input of the window contacts/door contacts
A1, A2	Analog inputs	Connection for interior temperature transmitter/ humidity sensor
A3	Analog input	Connection for the smoke detector
F1, F2, F3	LOGO! TD function keys	Input for the query signal (pushbutton)
B001/ B007	Logic basic function OR	Merging of the door and window contacts
B002	Pulse relay	Holding the signal of the door and window openers
B003	ON and OFF delay	Time-delayed switching on/off of the horn
Q1	Output	Connection of the horn
Q2, Q3	Output	Connection of the frost monitor/air de-humidifier
B015, B014	Message texts	Message via SMS in event of smoke/desired query
B005	Message text	Message for unauthorized opening of the residence
M31	Bit memory	Coloring of the message text
B009	Latching relay	Holding of the alarm signal for the counter
B008	Clock generator	Setting the time for alarm repetitions
B004	Counter	Setting of the alarm repetition
B006, B010, B011	Analog threshold switches	Conversion of the temperature signal/air humidity signal/smoke detector signal
B005	Logic basic function AND	Negating of the fan and de-humidifier operation
B016	Logic basic function OR	Activation of the message text for the remote query
B018, B019	OFF delays	Time-delayed switching off of the manual device activation

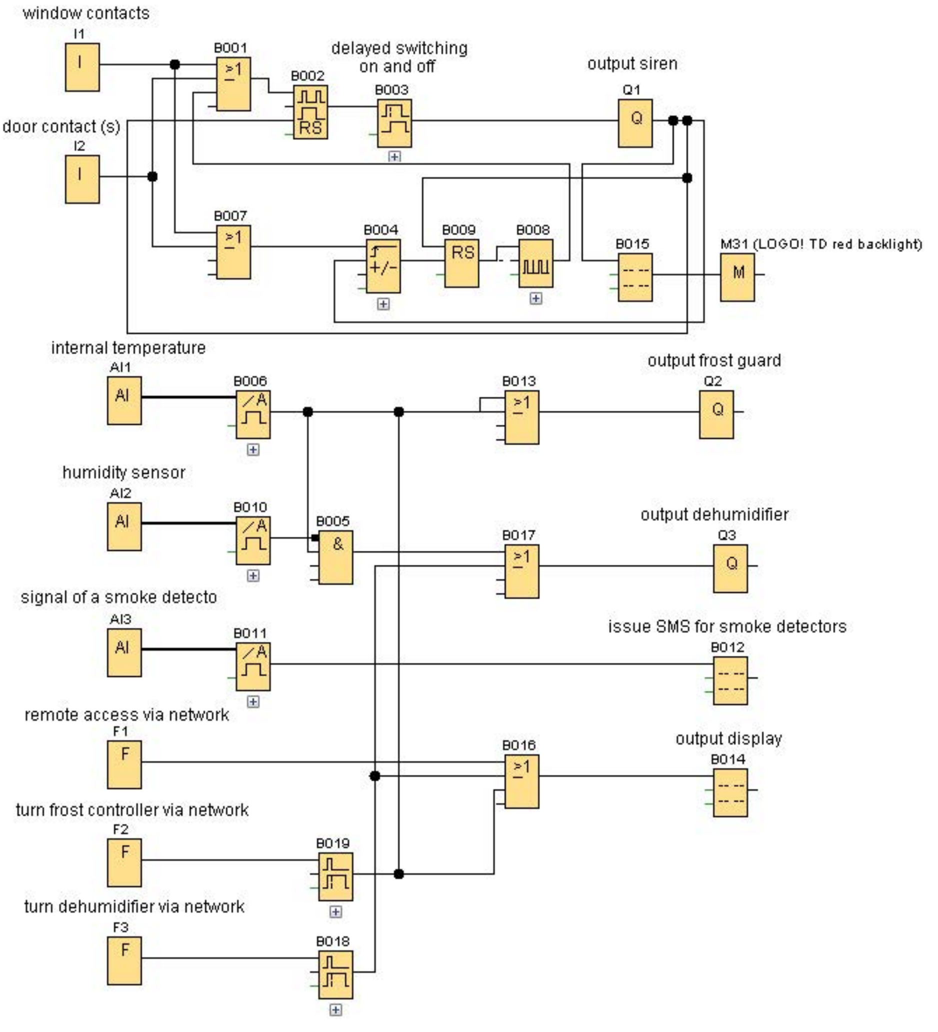


Fig. 6.79 Schematic diagram of the overall system

Tip

If the frost monitor is integrated into the project in the form of a heating fan, you must ensure that it is set up so that it cannot tip over and that no flammable objects are placed in front of the fan.

When setting up the air dehumidifier, ensure that it cannot tip over and that no objects are blocking the air inlet/outlet. The device must also be suitable for tank-free operation, i.e. the resulting water must be able to flow out through a hose into a nearby wash basin. To protect heating tubes from freezing, an antifreeze agent must be added to the heating circuit. Similar to the antifreeze used in windshield wipers in vehicles, the risk of frozen lines is thus prevented depending on the ratio of the mixture.

6.14 A service for seniors

The following project shows some possible options, which, with the aid of the LOGO!, can be used for implementing a sensitive monitoring system and support system for preventing accidents in private rooms (Fig. 6.80).

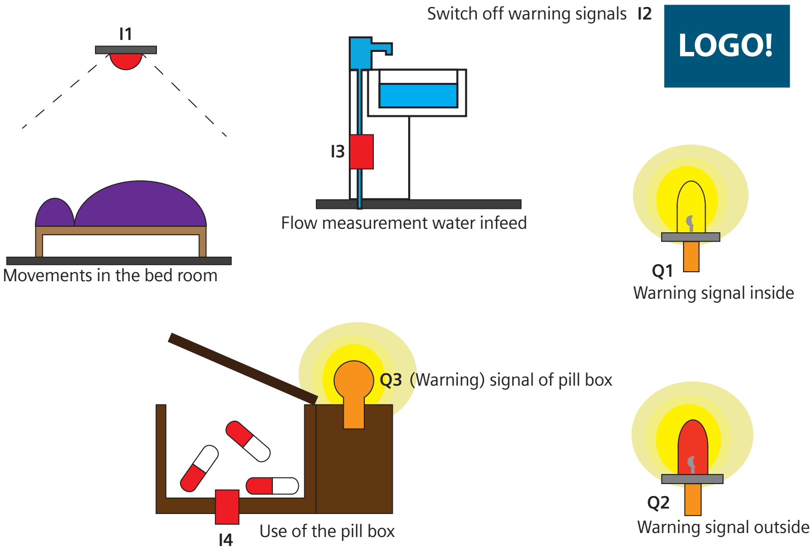


Fig. 6.80 Overview of possible assistance systems for seniors

Table 6.29 Content and functions of the "Senior service" project

Content	Blocks used in the program
✓ Integration of a motion detector ✓ Individual motion suppression ✓ Staggered alarm signals ✓ Integration of an analog flow meter ✓ Integration of a week time switch for time-limiting of the sensors ✓ Activation of an LED	✓ Digital inputs ✓ Bit memory ✓ Week time switch ✓ Up/down counter ✓ NOT ✓ AND ✓ OFF delay ✓ NOT ✓ OR ✓ Latching relay ✓ ON delay ✓ Threshold switch ✓ NAND (edge) ✓ High

This project introduces different approaches for supporting people who require assistance. The options described serve as examples and provide different approaches to the sensitive and careful monitoring of people requiring assistance. The circuits shown here are suggestions. They can easily be expanded, reorganized and combined.

Three independent concepts are the basis for the following circuit: integration of a motion detector, a possible application for a water flow meter, and an “intelligent” dosage of medicine. All three monitoring systems are connected to a graduated alarm. Initially, a first alarm is triggered, which only sounds in the residence of the affected person. After a parameterizable time, during which the alarm can be switched off in the residence, a second alarm is triggered. The graduated alarm is to give the person a chance to intercept an alarm that was unintentionally triggered before it alerts others. It can also be used to remind a person to take his medicine or to “Use more water”. The sensor can only measure how much water is taken from the line, not what the water is used for. The afore-mentioned SMS module can be integrated in an alarm system. This allows the alarm system to function worldwide and it allows certain measures to be introduced.

6.14.1 Motion detector subproject

To implement this, you will need a digital motion detector, a digital water flow meter, a light sensor, and a light-tight box with hinged lid. Depending on how the first and second stage of the graduated alarms are represented, an actuator will be needed for an acoustic and/or visual signal for the first alarm in the residence. A combination of acoustic and visual signals is used in this example. Additional hardware is needed for a second alarm system outside the residence.

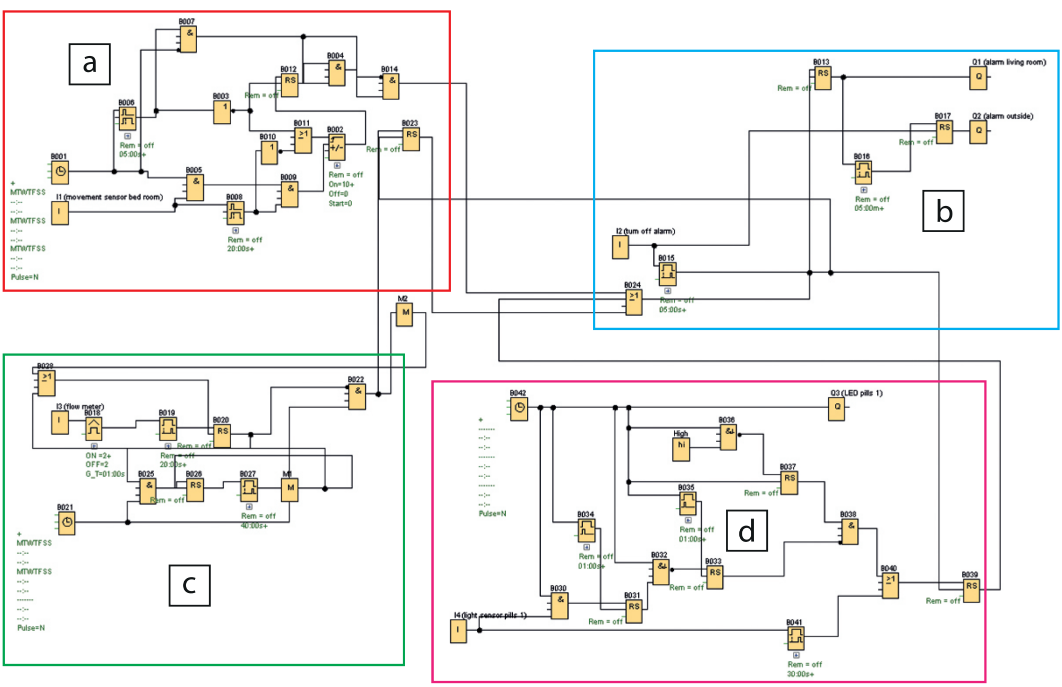


Fig. 6.81 Complete overview of the circuit for the senior service

This example starts with an in-house solution in which alarm stage 2 is structurally identical to the first stage, but is located in a neighboring apartment. Lastly, a switch is needed in the apartment of the person being monitored, which allows the first and second alarm to be deactivated.

Bit memories in the context of recursions are used on logical blocks. An up/down counter is used for analyzing the motion detector. Threshold switches and ON delays are used for measuring water. In addition, numerous latching relays, additional ON delays, the interval relay/pulse output block, as well as OR and AND blocks are used. Week time switches also play a central role for all of the sensors in order to limit the monitoring to specific, relevant times. Fig. 6.81 shows the overall circuit for this project.

The first option that is presented is an infrared motion detector. This registers changes in the reflection of infrared light, i.e. whether a person moves about in the apartment according to specific patterns. This should prevent the person who has fallen or is otherwise immobilized from lying helplessly on the floor. To avoid constant checking, the motion detector only records the morning and evening hours (waking up and going to bed). An alarm is only triggered if no waking up or reposing motions are registered within a parameterized time window in the morning and evening.

The week time switch (B001) can be set so that a signal is sent in a morning time window in which the person to be monitored usually wakes up. This ensures that the motion sensor in the bedroom I1 is only activated during that time. This is achieved via the AND block B005. This fulfills two functions:

- Firstly, an alarm should be triggered if no sufficient activity occurs in the parameterized time frame and
- secondly, the circuit should define whether the motions are sufficient based on the data from the motion detector.

Sleeping movements should not activate the motion detector. Such movements should not be registered as “waking up”. The circuit is programmed in such a way that it only outputs a positive edge if the motion detector is triggered ten times in succession. B002 takes over this function as an up/down counter. To rule out that a person who, while sleeping, moves ten times during the waking window and this is misinterpreted as “waking up”, the 10 edges of the motion detector are linked to two conditions. The first, as already described, is the time window, which is defined via B001. The second condition is an OFF delay, which holds the positive edge for 20 seconds after the negative edge change of the motion sensor I1. It is implemented by B008. The positive edge change at B008, in combination with the correct time in B009, activates the Cnt counting input of B002. At the same time, the positive edge of B008 switches off the negation B010. If a negative edge change occurs in B008, B010 outputs a positive edge. This, in turn, leads to a positive edge in B011, which resets the up/down counter B002 counting the ten movements back to zero. The result of this is that the ten movements and the positive edge change in B002 only take place if the time span between two movements does not last longer than 20 seconds. This would be the case when waking up, but not during restless sleep. Should 20 seconds prove to not be practicable in an individual case, it is possible to adapt the number of movements or time between them without much effort (Fig. 6.82).

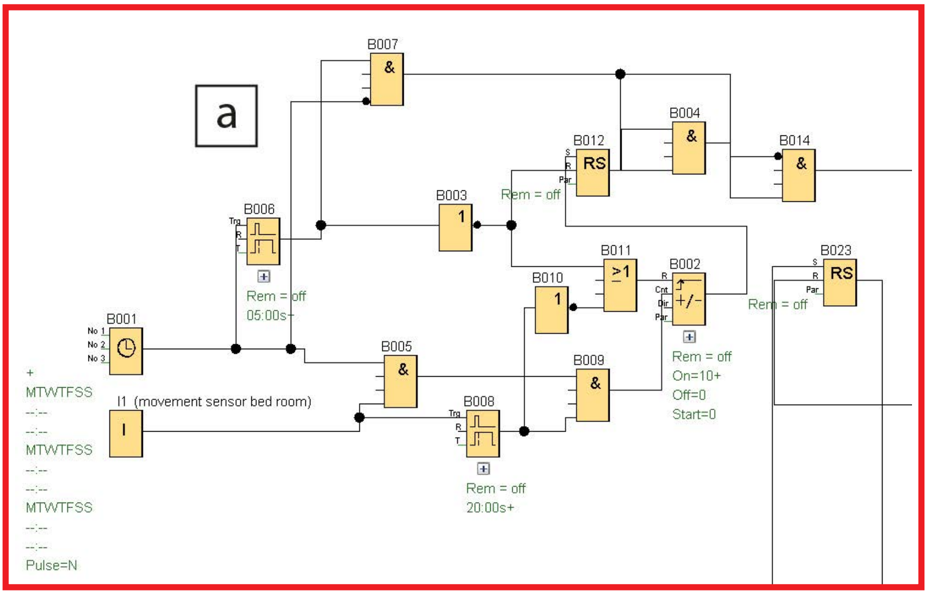


Fig. 6.82 Control of the motion sensor above the bed

If a positive edge change occurs in B002, the latching relay B012 is set to positive. This, in turn, prevents the AND block B014 from triggering the alarm when the time window closes. Regardless of when the monitored person actually gets up from bed in the waking time window, the resulting positive edge is held in B002. The negative edge change of the clock B001, which specifies the end of the time window, deletes the signals saved in B002 up to that point via the negation B003 and the OR block B011.

If a positive edge change occurs in B002, the latching relay B012 is set to positive. This, in turn, prevents the AND block B014 from triggering the alarm when the time window closes. Regardless of when the monitored person actually wakes up in the waking time window, the resulting positive edge is held in B002. The negative edge change of the clock B001, which specifies the end of the time window, deletes the signals saved in B002 up to that point via the negation B003 and the OR block B011.

The end of the time window should generally reset the counter and the relay to zero. But it can also trigger the alarm. This happens if the negated input 1 of B014 specifies a positive edge via a negative edge of B004 and B007 simultaneously signals the end of the time window. B007 does this only after a delay of 5 seconds after the time set in B001. The OFF delay B006 is to ensure that the LOGO! has completed all other switching processes. By combining the negation of B001 with the positive edge of B006, B007 is switched to positive for exactly 5 seconds after the negative edge change of B001. Across the entire time window during which B001 is active, the negative edge of B007 is achieved via the negation at input 4 of B007.

To summarize, an alarm is triggered by the motion detector if there are not ten movements which are less than 20 seconds apart each within a defined period of time. At the end of the time window, the program sequence is reset to the original state. Should it happen that B014 triggers an alarm, this takes place in the way described below.

B024, an OR block, at which the alarm signals of all of the components come together, sets the latching relay B013. This, in turn, forwards the positive edge to Q1, the alarm in the apartment. At the same time, it activates the ON delay B016. This then only adopts a positive state if the positive edge of B013 exists uninterrupted over a time period of five minutes. Only then does B016 output a positive edge, which, in turn, activates the latching relay B017, which then executes the alarm of the second alarm stage (see Fig. 6.83).

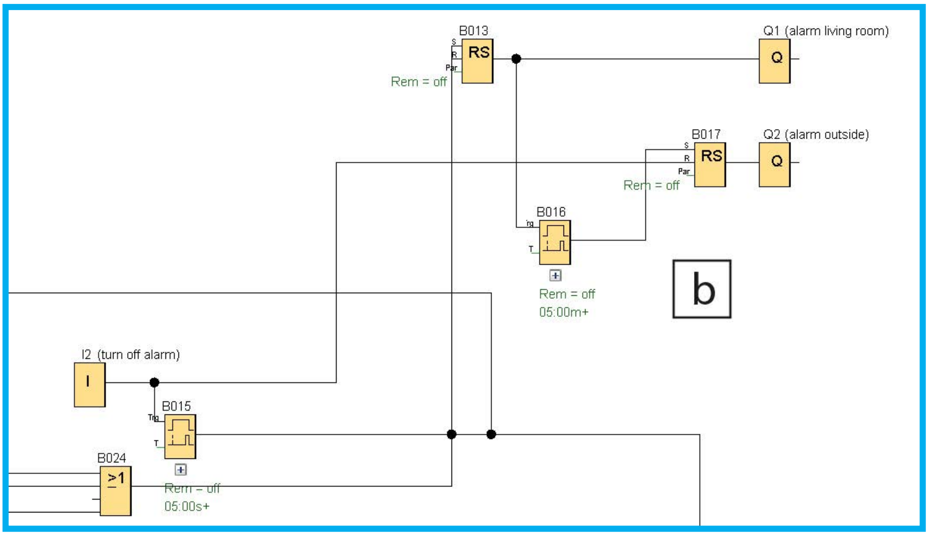


Fig. 6.83 Time-delayed activation of the alarm outputs

This means that between a first alarm in the residence and the activation of the second alarm stage, five minutes remain for switching off the alarm. To do this, I2, the off switch of the warning signal, must output a positive edge change. This signal influences the reset inputs of B013 and B017 and thus also switches off the alarms. To prevent misuse of this function, I2 must be held five seconds before the alarms go out.

6.14.2 Water consumption subproject

The second option for triggering an alarm is via a water flow meter on the kitchen – or bathroom water tap. If no water is drawn during a defined period of time, this can be an indication that the person in the apartment may need help. The commercially available pre-assembled component[18] outputs a digital signal depending on the flow-through per unit of time. This digital signal is summed

[18] Water flow meter, e.g. available from conrad.com, order number 503591

up and analyzed in the circuit described here during a defined time window. When the digital signal is output, you must ensure that the output frequency of the sensor does not exceed the maximum frequency of the LOGO!. This can happen quickly with sensors that were developed for micro-controllers of higher frequency – even at the quick input of the LOGO!.

In this example, the sensor is mounted between the cold water tap and the water line in the kitchen for instance. This is supposed to ensure that the person is active in the apartment and is using water. Alternatively, the hot water connection, the drain of the kitchen sink or bathtub, or the water connection of the toilet tank can also be used. Therefore, the parameters used in the following must be adapted to the respective conditions.

Fig. 6.84 shows the circuit of a clock for monitoring water consumption. The flow meter, which is connected to I3, is analyzed by the threshold switch B018. B018 then outputs a positive edge if I3 outputs a higher frequency than 2 positive edges per second, i.e. 2 Hertz. This corresponds to the flow rate of the water tap used in this example, which is opened mid way. B018 is connected to the Trg input of the ON delay B018. B019 only outputs a positive edge if B018 holds a positive edge over a period of 20 seconds, i.e. cold water flows in the right amount for 20 seconds. This edge sets the latching relay B020. If B020 were not activated via a positive edge of B019, this would become a positive signal via the negation in B022. This positive signal would lead to the alarm being triggered due to the positive edge change of B021 when the time is reached by B022 outputting a positive signal. The other triggering mechanism corresponds to that of the motion detector. Relay B020 must be reset after each time cycle in which it had a positive edge to ensure that the positive edge is not permanently saved by B022 and no more alarms are triggered.

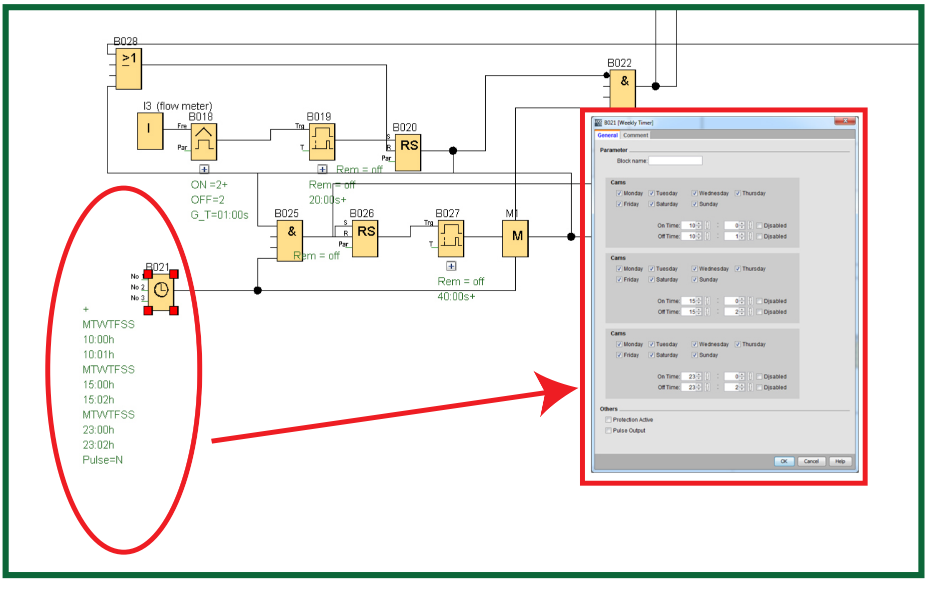


Fig. 6.84 Setting the clock block for defining the recording of water consumption

6.14.3 Medication taking subproject

The third option for triggering the alarm is the monitoring of the taking of medication, if the person is instructed to take medication regularly. Due to the consequences of forgetting about a dose of medication can have, it seems to make sense to double-check to ensure a monitored person does not omit to take his or her medication. It must be pointed out that the solution presented here can only be used to register the opening and closing of the pill box. The circuit cannot show whether the box was intentionally opened and closed to avoid the alarm or whether the person forgot their medication.

The system for checking whether medication was taken is based on the principle of a box with a hinged lid that is impenetrable to light, a digital light sensor, which outputs a signal when it is exposed to light, a time clock logic block, and a signal LED. The signal LED is intended to indicate that a specific time period for taking the pills has been reached. Of course, an acoustic signal can also be added. To avoid triggering the alarm, the pill box must be opened and closed within a parameterized period of time.

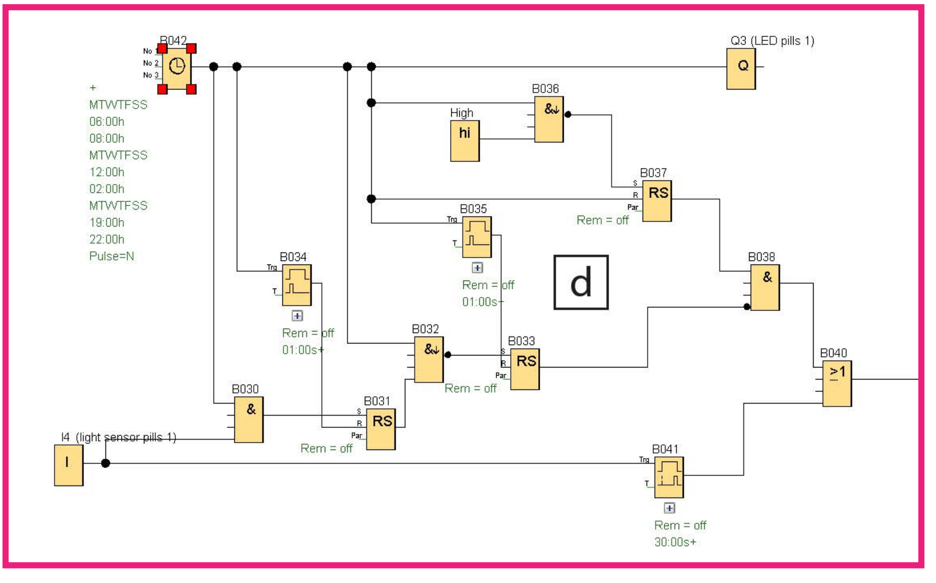


Fig. 6.85 Circuit for monitoring the taking of medication

Fig. 6.85 shows the circuit for this monitoring system: At the start, the high input 4 of NAND block B036 and the negated input 4 of AND block B038 are activated. For this purpose, light sensor I4 is connected to the timer clock B042 in the AND link B030. The LED at Q3, which is directly connected to B042, illuminates to indicate when a new time period is reached.

The latching relay B031 is only activated if B042 outputs a positive edge, i.e. a medication taking time period has started and the light sensor is exposed in this time period. B031 is first reset by interval relay B034 one second after the start of each new time period. I4 is also connected to the ON delay B041. This forwards the positive edge after 30 seconds. If the light sensor remains exposed longer than 30 seconds, i.e. the pillbox is open, the graduated alarm is triggered. The person then has the parameterized forwarding break between the alarms to close the pill box and deactivate the alarm (e.g., 5 minutes). This is intended to prevent the box from being left open. Switching off the alarm also deactivates the latching relay B039, which forwards the alarm.

To ensure that each time period begins "new", the latching relay B033 is also reset by B031 via an interval relay B035 at the start of a new time period. The interval relay comes into effect because a continuous edge at the R input of a latching relay would lead to a situation in which the relay cannot be activated. According to this principle, latching relay B037 prevents it from being activated during a period of time.

If the block B033 is not activated via I4 during a certain period of time, the AND block B038 outputs a positive edge, which, in turn, triggers the graduated alarm via the OR block B040. The positive edge of B038 results from connecting the negated input of B033 and the positive input of B037. This latching relay is set by B036 at each end of a time window when the negative edge change of B042 activates the NAND block B036. According to the same principle, the system also waits at the NAND block B032 until the end of the cycle before the completed opening and closing of the pill box is forwarded. Opening and closing the box multiple times within the time window has no consequences. Also opening and closing the box outside of the time window has no consequences. Only leaving the box open triggers the alarm even outside of a time period.

Table 6.30 Function table of the "Service for seniors" project

Block name	Property	Function in the program
I1	Digital input	Motion detector in the bed room
I2	Digital input	OFF switch for warning signals
I3	Digital input	Flow meter at the water infeed
I4	Digital input	Light sensor in the pillbox
Q1, Q2	Output	Warning signal inside/outside the apartment
M1, M2	Bit memory	Recursion 1/2 for the flow meter
B001	Week time switch	Activates the motion sensor
B002	Up/down counter	Adds up the number of motions
B010, B003	Logic basic function NOT	Resets B002, resets B002 after a set time
B004, B005	Logic basic function AND	Connects the time setting to the motion analysis/sensor
B006	OFF delay	Delays the switch off of the timer clock
B007	Logic basic function AND	Connects the switch-off of the timer clock to B006
B008	OFF delay	Deletes B002 after a set time
B009	Logic basic function AND	Connects the motion time window to the motion sensor
B011	Logic basic function OR	Connects the reset options for B002
B012, B020	Latching relay	Hold the signal of B002/B019
B013, B017	Latching relay	Hold the signal for triggering the alarm of Q2
B014	Logic basic function AND	Connects the time analysis to the motion analysis

Block name	Property	Function in the program
B015	ON delay	For switching off, I2 must be held 5 seconds
B016	ON delay	The warning signal is sent out after 5 minutes
B018	Threshold switch	Flow-dependent timing
B019	ON delay	Requires 20 seconds B018 = 1 for triggering
B021	Week time switch	Gives a time frame for water usage
B022	Logic basic function AND	Connects conditions that trigger alarms
B023, B026	Latching relay	Holds the alarm signal/reset signal for B020 signal
B024	Logic basic function OR	Connects alarm-triggering signals
B025	Logic basic function AND	Connects time settings and flow meter
B027	ON delay	Delays the reset signal beyond the alarm time
B028	Logic basic function OR	Connects the reset signals for B020
B030	Logic basic function AND	Connects time and light sensor
B031, B033	Latching relay	Holds the non-alarm signal
B032	Logic basic function NAND (edge)	Forwards non-alarm signal when B042 is switched off
B034, B035	Interval relay/pulse output	Resets B031 when a pill period starts over
B036	Logic basic function NAND (edge)	Activates B037
B037	Latching relay	Specifies the end of the time window
B038	Logic basic function AND	Triggers alarm if I4 was not active in the time window
B039	Latching relay	Triggers an alarm
B040	Logic basic function OR	Triggers alarm via B041 or B038
B041	ON delay	Triggers alarm if box remains open
B042	Week time switch	Specifies pill taking times
High	High	Provides duration -1 for one of the inputs of B036

Tip

Of course, it is possible to leave out some of the described components.

The motion detection section can be expanded easily.

Temperature sensors have also proven themselves in the monitoring of kitchen or fireplace areas, signaling when a burner on the cooker is not switched off, for example. Water detectors on the floors of the wet areas can be used in order to indicate “flooding” due to water taps not being closed.

6.15 The automatic lawn mower

Robotic lawn mowers are available from many suppliers. The majority of them function according to a similar principle: A transport vehicle transports a mower, which is comprised of a rotating disk with blades mounted on it. The robot travels to the end of the area to be mowed and turns around by itself when the limit is reached. Then it traverses across the area to be mowed again until it reaches a new control wire. Limitations can implemented by means of fixed obstacles, which the robot bumps into, or by an inconspicuous induction loop. Depending on the model, these robotic mowers run across the lawn until the battery is low or dead. Then the device is piloted back to the charging station in various ways, mainly by an induction wire.

Such a lawn mower is to be built in this project. It is equipped with induction sensors, which respond to wires mounted at ground level. Additionally the lawn mower is equipped with tip-over protection (see Fig. 6.86).

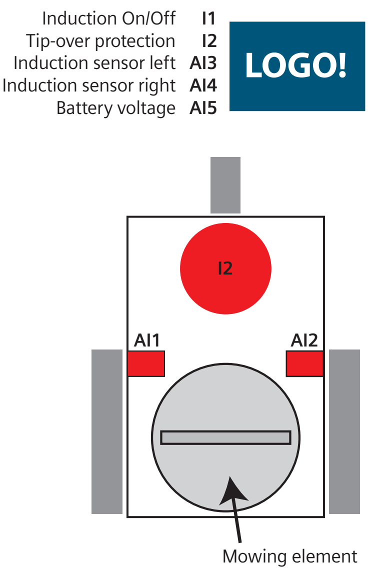


Fig. 6.86 Schematic diagram of a three-wheeled lawn mower

Table 6.31 Content and functions of the "Lawn mower" project

Content	Blocks used in the program
✓ Use of digital and analog inputs ✓ Recursions ✓ UDF blocks ✓ Division of a diagram into several pages ✓ Complex delays ✓ Various complex counting options	A large number of the blocks that are already known are used in this project. Due to the complex interconnections in this project, it is primarily aimed at advanced LOGO! hobbyists who are already familiar with the functions of the individual blocks. The use of UDF blocks must be highlighted.

There are two possible approaches for the carrier for the mower. First, it is possible to mount the lawn mower on a vehicle, which has a steering system that is controlled by a motor. Hence, the drive is not used for steering. The second approach involves structuring the mower according to the well-known wheeled robot model; with two individually controllable large rear wheels and a passive, independently steered front wheel. Both rear wheels are driven by one motor each. Cables, a metal sphere, and a funnel are needed for the tip-over protection. The mowing element can be a disk secured on a geared motor with cutting blades glued to it. The function and the materials needed for the tip-over protection will be described in detail later in this chapter.

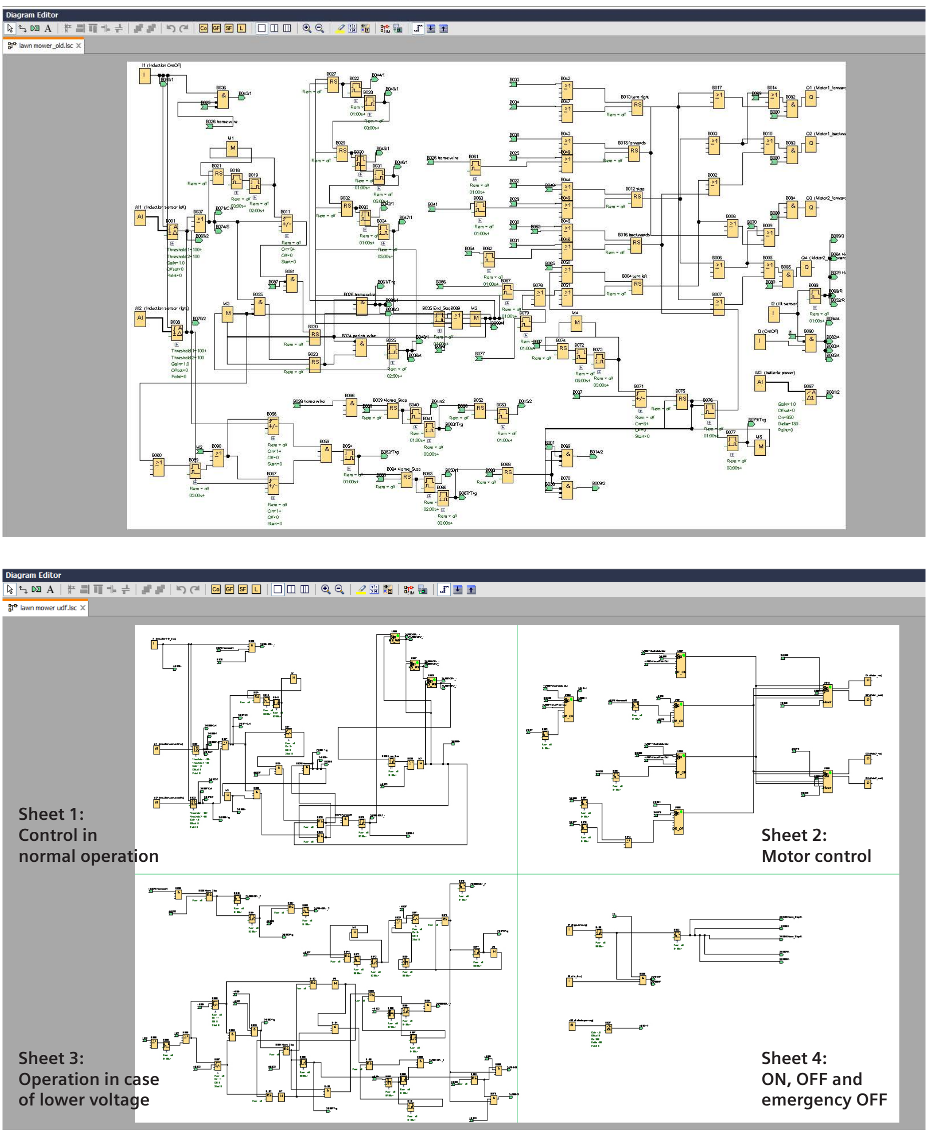


Fig. 6.87 Overall circuit diagram of the automatic lawn mower, above without and below with division into blocks

Before we go into detail about the actual circuit in this section, we want to point out again an important preliminary, the dividing of the diagram into several sheets. For a better overview of this project, we have distributed the program over four pages, which are first displayed individually. From a comparison with the diagram display on one side without UDF blocks, it is clear what sort of contribution UDF and page division can make to the structured design of projects. First we compare the authentic first circuit on one sheet to the final, revised circuit on four sheets (Fig. 6.87).

Please note that the left circuit on one sheet is in a significantly larger scale than the four charts in the right diagram. Dividing the work area into several pages makes it possible to devote individual pages to specific task areas. In this way, debugging is facilitated considerably, because it is easier to trace and assign the connections. The division can be achieved by using the 1,2,3,4 buttons as already shown in Fig. 6.88. The logic blocks can then be moved beyond the borders of the sheets, connected, and disconnected.

For this very complex lawn mower circuit, we will begin with the description of the outputs (Fig. 6.89).

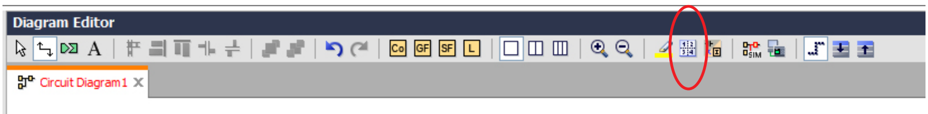


Fig. 6.88 Button for selecting the sheet division options

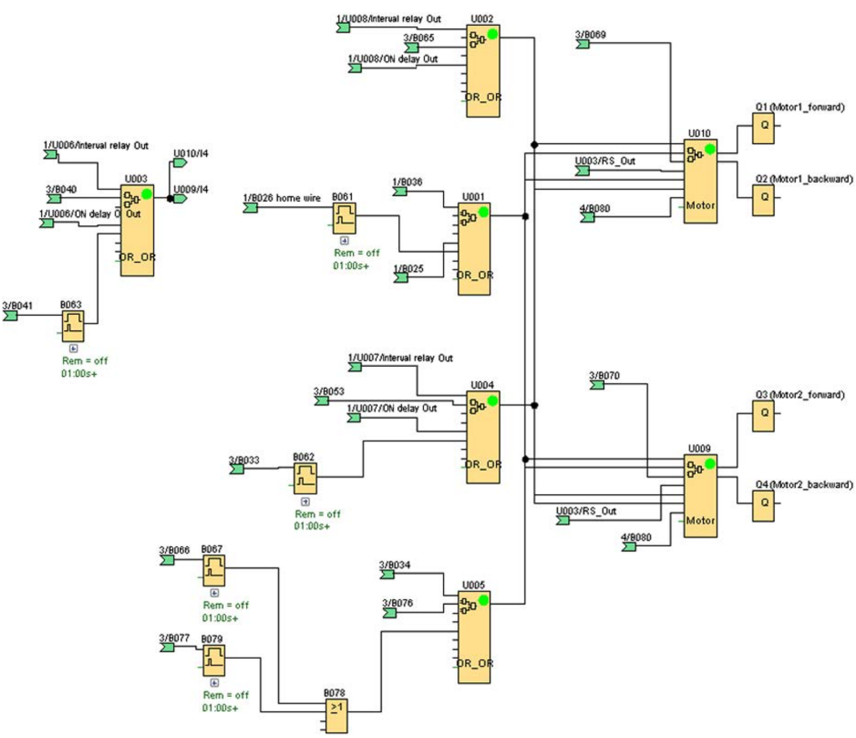


Fig. 6.89 Sheet 2 "Motor control"

6.15.1 Controlling the two motors subproject

The four outputs are distributed to the control of two motors. Q1 and Q2 control the motor for the left wheel, Q3 and Q4 control the right one. A positive signal from Q1 leads to a forward rotation of the left motor, whereas a positive signal from Q2 leads to a reverse rotation of the same motor. If Q1 and Q2 were to simultaneously output a positive edge, the motor would quickly stop. Zero at both outputs leads to no movement of the motor. A negative edge change of the only active output would lead a “coast-down” of the motor. The same applies to Q3 and Q4 for the right motor.

The movements listed in Table 6.32 can be achieved, where: 1 = forward, 0 = no signal, -1 = reverse.

Table 6.32 Circuit logic for direction commands

Motor left	Motor right	Movement
0	0	No movement
1	1	Forward movement
1	0	Wide right curve
1	-1	Tight right curve
0	1	Wide left curve
-1	1	Tight left curve
1 and -1	1 and -1	Active stop

The UDF blocks U009 and U010 are connected upstream of the motors. The UDF blocks permit a better overview of the motor control. They are described in more detail in the following sections.

In the simplest scenario, the lawn mower runs straight ahead, i.e. Q1 and Q3 quickly rotate forward at the same rate. The conditions for this are the switching on of the lawn mower via I3, not triggering the tip-over protection I2, and switching on the induction sensors via I1.

A UDF block U001 then uses a positive edge to control both U009 and U010 in such a way that they activate Q1 and Q3. Following the same pattern, in which U001 activates Q1 and Q3, U004 initiates the reverse movement via Q2 and Q4. U002 and U005 control Q1 and Q4, or Q2 and Q3 (see Table 6.32). Deviating from the logic of U001, U002, U004, and U005, U003 controls all four outputs Q1 to 4 at the same time, in order to bring about the active stop (Fig. 6.90).

The lawn mower traverses the area to be mowed according to a random “billiard ball principle”. It runs straight until one of the induction sensors AI1 and AI2 outputs a signal above the threshold. The signal threshold is parameterized in B001 and B037. The analog value monitoring systems are activated by the already mentioned induction operating switch I1 induction ON/OFF at the En input (Fig. 6.91).

At some point in the lawn mowers forward run, it will pass over the limit cable with one of the induction sensors mounted on the front and thus generate an above threshold value. This leads to a long forwarding of the signal, which, on

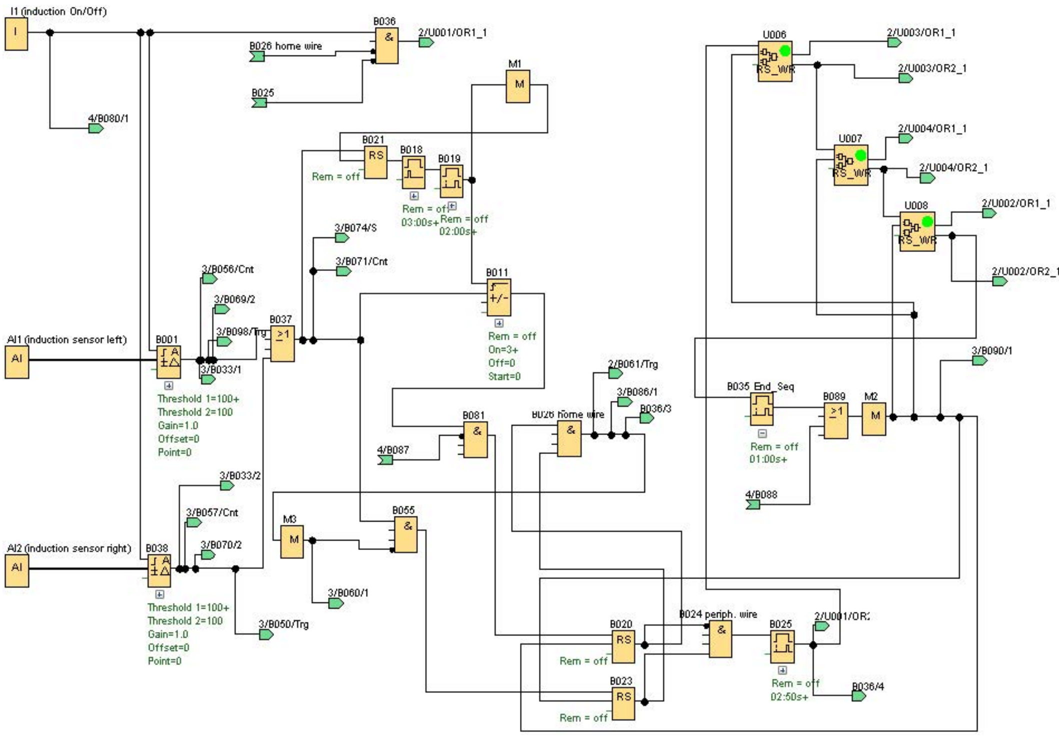


Fig. 6.90 Sheet 1 Control during "normal operation"

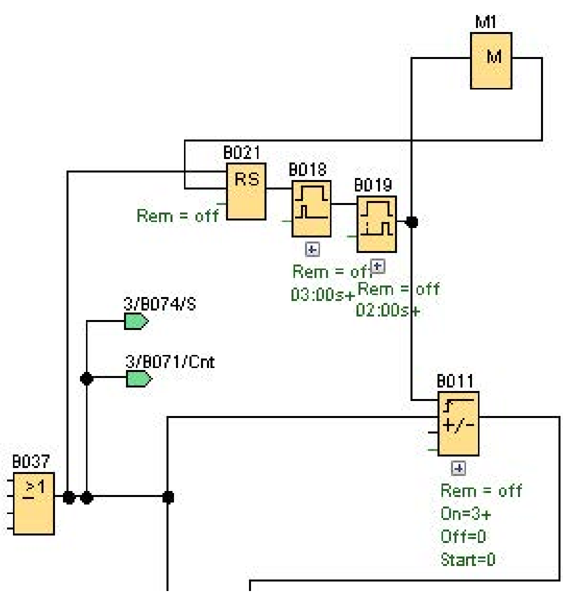


Fig. 6.91 Controlling the counter via the induction sensors

the one hand, puts an end to the forward run by means of the ON delay B025. At the same time, a signal cascade is initiated by B025 involving successive triggering. First, an active stop is triggered, then a reverse run (Q2 and Q4), until finally a right rotation is initiated. Thus, after turning in a direction other than the one it came from, the lawn mower acts similarly to a billiard ball which “bounces” off the edge. The signal cascade, which controls the active stop until the run resumes after the turn, is also based on UDF blocks for simplicity‘s sake: U006, U007 and U008.

On the basis of the previous descriptions, you can calculate how long the lawn mower runs in a zig-zag pattern until the battery runs out of power (Fig. 6.92). In this case, the mower would stop somewhere on the mowing area.

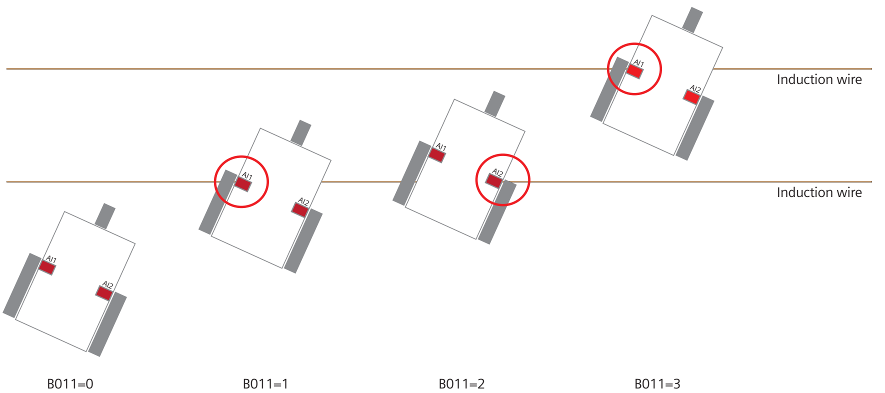


Fig. 6.92 Responses of the mower to passing over a signal wire

Even if it would go beyond the scope of this project to recharge the mower‘s battery at a docking station, we will at least show how the mower can be led back to a specific parking position.

6.15.2 Putting the lawn mower into parking position subproject

To do this, an additional network of wires must be laid in addition to the limit wires on the lawn. These can be laid underground. In our project, we have seen that also firmly pressing the signal wires down at ground level and securing them there leads to the wires being completely overgrown with grass within a very short time. The wires laid in this way should connect the desired parking position to the center of the mowing area. If there is an indication that the battery has reached a critical value via the analog input AI3 battery voltage and the downstream analog differential threshold switch via a missing signal from B087, the mower responds differently to the information from the induction sensors.

As before, the mower counts the number of induction sensor signals within a short and settable period of time. Whereas the limit wire is a single wire, the wire that leads to the parking position is a double wire and thus more closely resembles a home position corridor. The time that the mower is supposed to wait so as to see whether there is a second signal for the continued forward run must be adapted to the speed of the mower in each individual case. If the edge of B087 is not present, which indicates that the battery voltage has gone below a critical value, the mower continues to run in a zig-zag pattern until it receives three signals in succession when it passes over the home wires within the defined period of time: A signal from one sensor at the first wire, then a second signal from the other sensor at the first wire and, lastly, a third signal when passing over the second wire with the first sensor.

This counting function is implemented via the blocks B021, B018, B019, M1, and B011. The positive edge change of B001 or B038 sets the latching relay B021, which, in turn, activates the interval relay B018. B018 generates a signal with a 3 second duration, regardless of the fact that B021 is also active beyond this. With the ON delay B019, this signal leads to a positive signal after 2 seconds. Since recursions are not possible in LOGO!, a bit memory must be connected at block B019, which, in turn, resets B021 at the end of the cycle. B019, the end of the signal cascade, can then reset B011, which is still counting incoming signals from the induction sensor carriers, to zero. B011 can only forward a signal if it counts three signals from the induction sensors within the time arranged in B018 and B019, which is the case when the mower passes over the home wires that are connected in parallel.

If three signals do not occur in this time, the mower turns and continues running in a zig-zag pattern. The times must be adapted to the respective conditions of the home wires for the circuit to succeed.

If, as described previously, the "home wire control" is triggered, the forward signal is first ended by a signal from the interval relay B061 to the UDF input 6 of U001. This procedure was separated for the sake of a clarity and is marked by the separator 2/B061/Trg. Fig. 6.93 shows how a brief reverse run is initiated via the blocks between B086 and B053, with all of the branch-offs and split connections, which is to align the mower for the run to the home position (This connection was also split (3/B086/1). This connection is described in detail below as "B026 control".

For the sake of completeness, the AND block B036 interrupts the induction switch I1, which has an active edge that is a condition for controlling the motor, as described previously. The connection of B026 and B036 was also split for the sake of clarity. The OR block B060 is controlled by the bit memory M3 for the run to the home position. The connection is split (3/B060/1). This circuit is described in detail below as "B060 circuit".

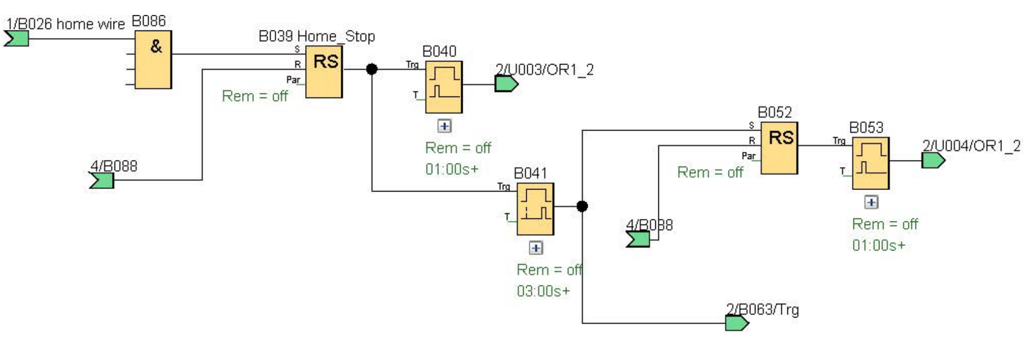


Fig. 6.93 Introduction of a brief reverse run in the "home program"

6.15.3 Lawn mower with low battery power subproject

The "B029 control" (UDF block) is used when the battery power has become so low that B087 no longer outputs a signal and the home wire has been triggered by three induction sensor signals in rapid succession (Fig. 6.94). A latching relay B039 is first activated for the other switching processes. On the one hand, this triggers the active stop U003 via the interval relay B040. On the other hand, a reverse run is initiated using an ON delay B041, which prevents the following blocks from overlapping with the stop of B040. In addition, the active stop is triggered again by 2/B063/Trg.

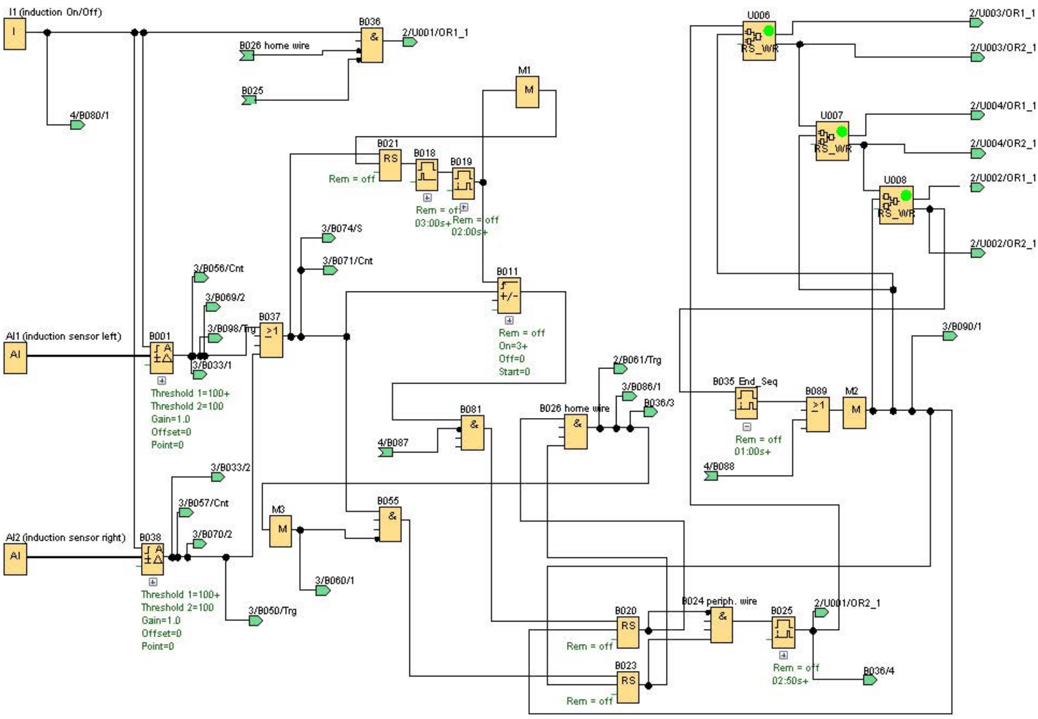


Fig. 6.94 Sheet 3 "Operating at low voltage"

6.15.4 Lawn mower in the parking position subproject

The "B060 circuit" controls the mower's return to home. It is triggered by the bit memory M3. At the time, the mower is in reverse between the two home wires. It should stop when both of its sensors have passed over the home wire.

This includes the up/down counters B056 and B057, each in combination with the sensors 1/B001 and 1/B038, the positive edge change. The AND block B033 only outputs a positive edge after both counters have counted the positive edge change. This triggers an active stop.

The mower should now be realigned in order to travel along the home wire to the parking position. A latching relay B064 is first activated for this purpose. The positive edge is divided into two signals. First the turn is triggered. This is a left or right turn, depending on which induction sensor of the mower passes over the wire first.

In the example in Fig. 6.95, the process is shown as a left turn. Before the final forward run is initiated, the mower turns about its own axis until the sensor that was not triggered during the reverse run passes over the induction wire. For better alignment, the mower then turns one second longer before ending the alignment.

An ON delay B066 also runs for six seconds during the alignment. Then the block initiates the forward run. To this end, a latching relay is activated, which, in turn, directly controls the motors. Whenever only one of the mower‘s sensors passes over the wire, the motor on the same side is switched off for the time of the positive edge. Thus, the mower makes a corrective curve.

The AND block B069 is activated by a positive signal from latching relay B069 and the negated signal of the left induction sensor. B069 is the signal for the forward run in the home wire. If the induction sensor does not have a positive edge and B069 is active, the left motor rotates forward. The same applies the other way around for block B070, which controls the right motor (Fig. 6.95). The corresponding directions of travel result from Table 6.32 for motor circuit combinations.

There is a 50% probability, however, that the mower is traveling along the home wire at the “wrong” angle and thus not to the parking position but is actually moving away from it. To prevent this, we have arranged a triple wire transverse

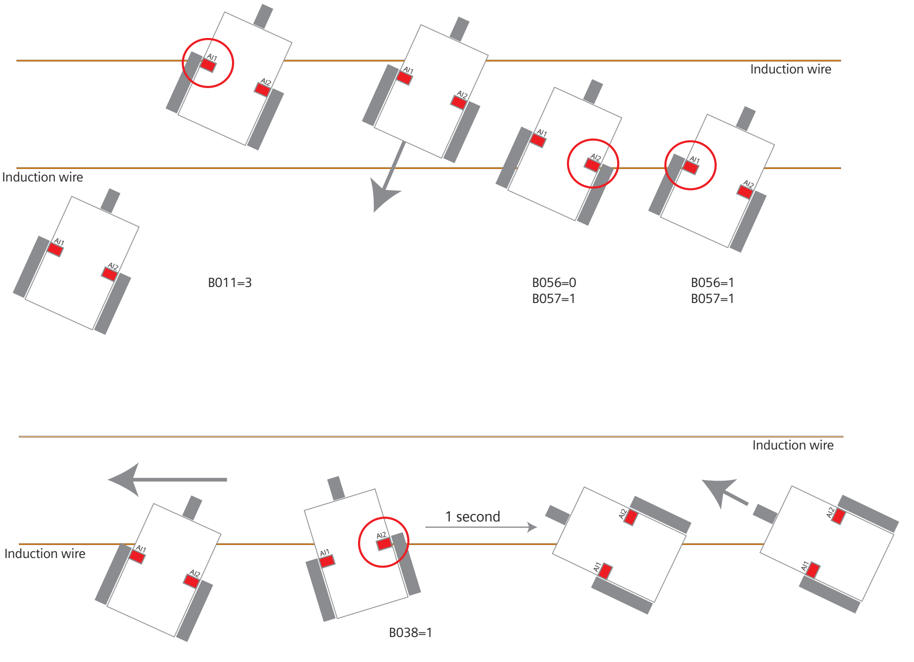


Fig. 6.95 Complex subsequent response depending on different signal wire messages

to the home wire direction at the end of the induction home wire. Following a counting principle similar to the previous one, the mower can be turned around to then traverse the home wire in the other direction to the parking position (Fig. 6.96).

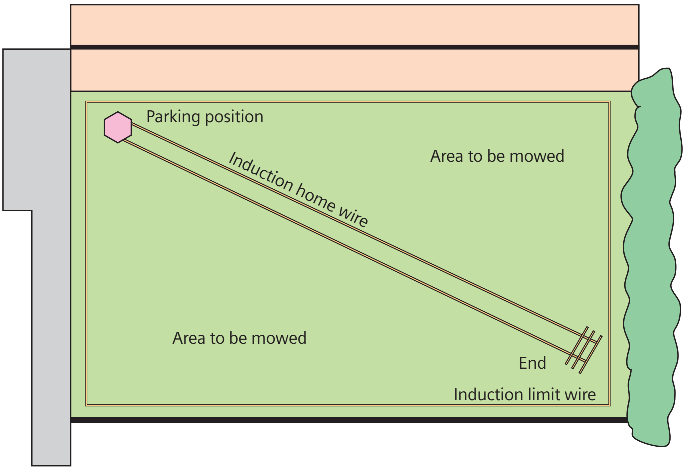


Fig. 6.96 Arrangement of the control wire on the area to be mowed

This circuit is achieved by means of the blocks between the separation 1/B037 and M5. To this end, it receives input from the OR block B037, which connects the analog monitoring systems directly behind the induction sensors AI1 and AI2. The counting process functions according to the same principle as the procedure described before. A time is specified via the latching relay B074, the interval relay B072, and the ON delay B073, in which the signals of B037 are counted. The up/down counter B071 is reset by a positive edge from B073. If B071 reaches the number six before the positive edge from B073, which can be explained by the fact that both sensors pass over three wires each, then the latching relay B075 is set.

The signal splits here. On the one hand, a left turn is initiated by the interval relay B076. On the other hand, the forward motion is prevented. In addition, an ON delay starts, which must be parameterized in such a way that the mower makes an approximate 180° turn. Then the left turn that was initiated by B076 is ended with the positive edge of the ON delay B077 and the latching relay B075, which prevents the forward movement, is reset. To prevent a situation in which the repeated passing over of the end wires on the way to the park position triggers the responses just described, the up/down counter B071 is reset by the bit memory M5 after B077 until the mower has left the end zone again – in this case, the time for this is set to 5 seconds in the OFF delay B031 (Fig. 6.97).

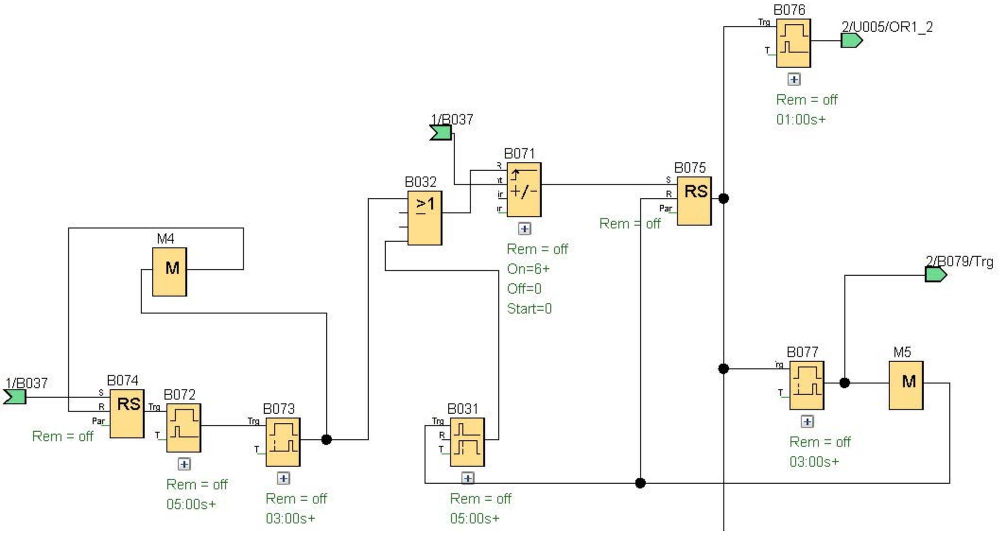


Fig. 6.97 Counting function in a specified time window

6.15.5 Tip-over protection subproject

To understand the behavior of the mower in the parking position, the tip-over protection must first be explained. The lawn mower is a vehicle with very sharp, fast-moving blades on its underside. It is clear even to laymen that this represents a significant hazard. Therefore, at this point, we must explicitly point out that we recommend that the mower never be allowed to mow unattended, even if the tip-over protection safety mechanism is reliably installed.

The mowing element of the lawn mower must be touch-protected at ground level on all sides. At the same time, the operator must ensure that the mowing element no longer rotates if the mower is tipped over. This can happen if the mower is overturned or even if the mower is lifted or threatens to run over a

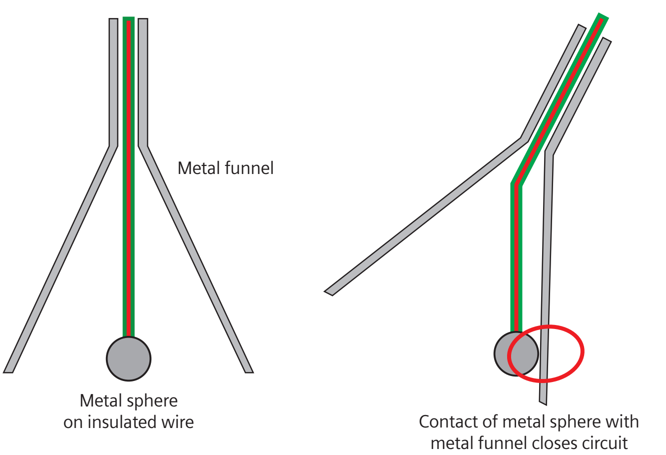


Fig. 6.98 Tip sensor made of a funnel, aluminum foil and a swinging ball

foot or small animal. In addition to the commercially available gyro sensors, which can detect angular inclinations in all directions, there is also a very simple "do-it-yourself", which fulfills exactly the same purpose (Fig. 6.98).

To do this, we have connected a metal sphere to a pendulum with an electrically conductive insulated wire. This pendulum is then suspended in a metal funnel. Commercially available funnels are available for this. Cut-open plastic bottles lined with aluminum foil have also worked well for this. A circuit is closed by the contact of the conductive sphere with the metal wall, which then, in turn, initiates a positive edge at the input of the LOGO!. In order to prevent slight unevenness and a "to and fro" motion of the pendulum from triggering the emergency brake, an ON delay of 0.5 seconds is placed behind the pendulum switch, which then stops all forward movement via an interval relay (Fig. 6.99).

As could already be seen in the applications described previously, all motor movements are controlled using latching relays, which are connected over the R input with the tip-over protection. If the tip-over protection is triggered, all motor movements are stopped and thus the rotation of the mowing element as well. In addition, the positive edge of the ON/OFF switch I3 is only forwarded if the tip-over protection I2 is not positive. This ensures that a mower with rotating blades can only move if it is at ground level. At the end of the home wire, a ramp is waiting for the robot. If the mower goes up it, the tip-over sensor is triggered and the robot stops.

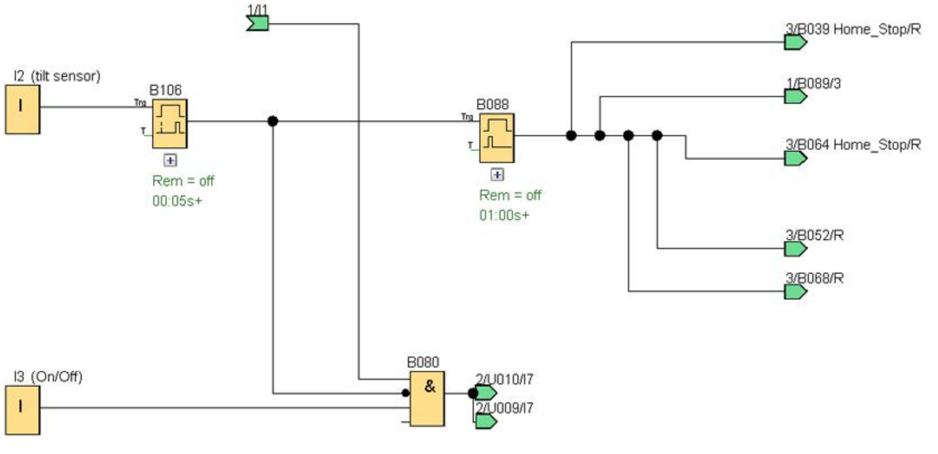


Fig. 6.99 Sheet 4 "On/Off and emergency off"

6.15.6 UDF blocks in the project

UDF blocks are used at three points in this project. These contribute considerably to the clear design of the diagram. In this diagram, a UDF block with the name "Motor" was created, which can control a motor to make it move either forward or backward. A UDF block "OR_OR_RS" was created for controlling two motors, according to the table of the possible directions of travel. A final UDF block "RS_WR_EV" has been programmed for carrying out the turning maneuver (Fig. 6.100).

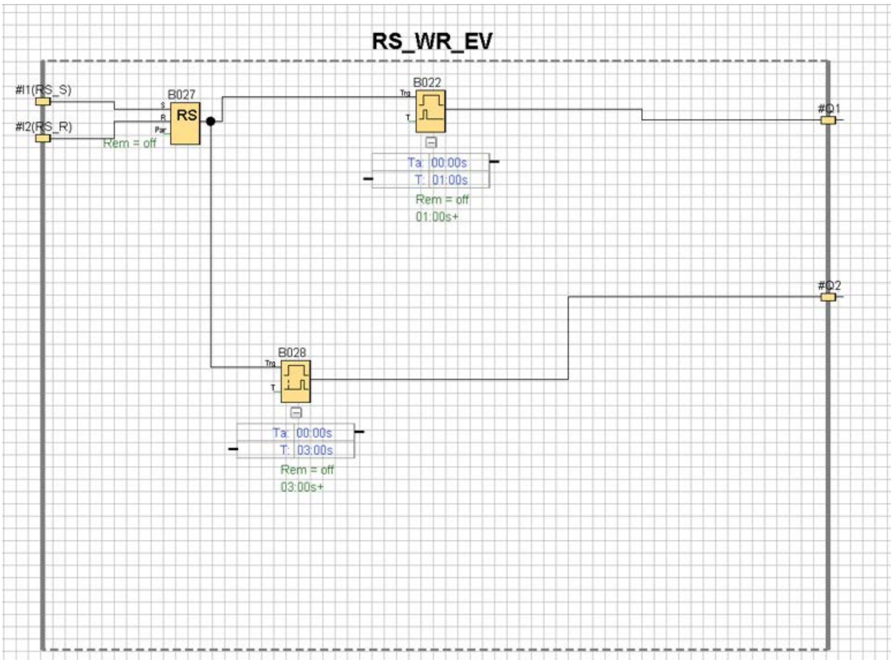


Fig. 6.100 Circuit with RS_WR_EV UDF block

In the case of RS_WR_EV, a latching relay, an interval relay, and an ON delay were connected (Fig. 6.101).

Using a UDF block saves two thirds of the logic blocks in the diagram. In addition, the low number of inputs and outputs makes it easier to correctly link the blocks and have a good overview. The goal of the block is to have two blocks

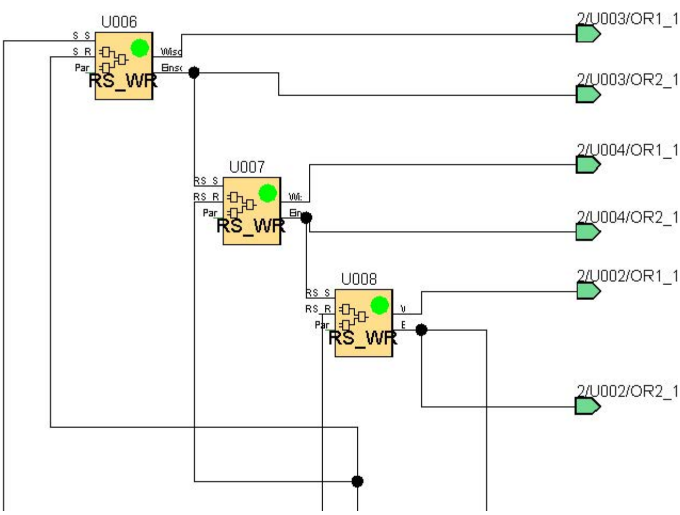


Fig. 6.101 The RS_WR_EV UDF circuit

carry out respective, different tasks chronologically coordinated on one input signal. Since this request is repeated three times, UDF blocks are the obvious choice.

The “OR_OR_RS” UDF block is programmed in such a way that two motors can be controlled via one block (Fig. 102). The combination options are compiled in Table 6.32. The arrangement of the inputs is selected in such a way that the top four inputs activate the latching relay and thus the output, whereas the bottom four inputs reset the latching relay and thus the output. Using a UDF block makes the diagram more manageable and easier to connect (Fig. 6.103).

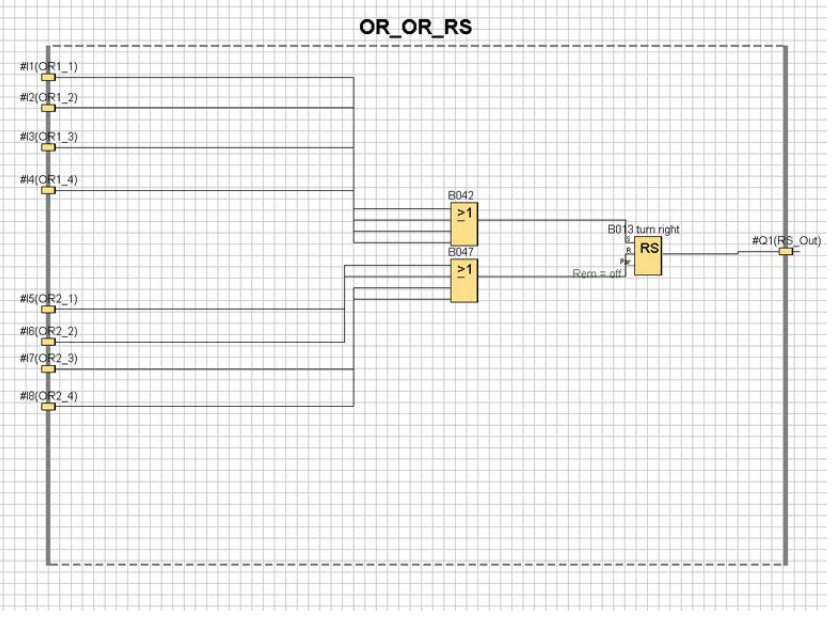


Fig. 6.102 Circuit in the OR_OR_RS UDF block

The last UDF block in the direction of reading (Fig. 6.104) is the “motor” UDF. This permits the control of a wheel in two directions and active stopping. In addition there is the “safeguard” that the motors only receive a positive edge if the ON/OFF switch I3 also has a positive edge. To this end, the signal of I3 is connected and analyzed in the AND blocks B084 and B085 for the respective direction with the results of the respective upstream OR blocks. Due to the OR blocks, it is possible to not connect individual inputs or to add more signals as needed. The number of assigned inputs also varies in this switching program. Nevertheless, the UDF blocks contribute considerably to the simpler design of the diagram (Figs. 6.104 and 6.105).

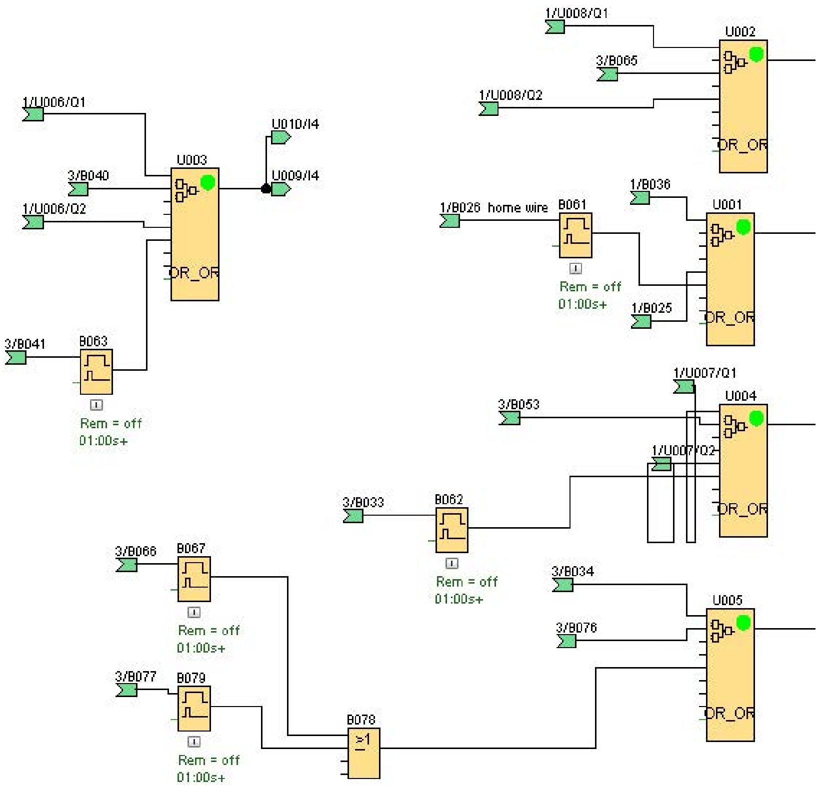


Fig. 6.103 Arrangement of the UDF blocks

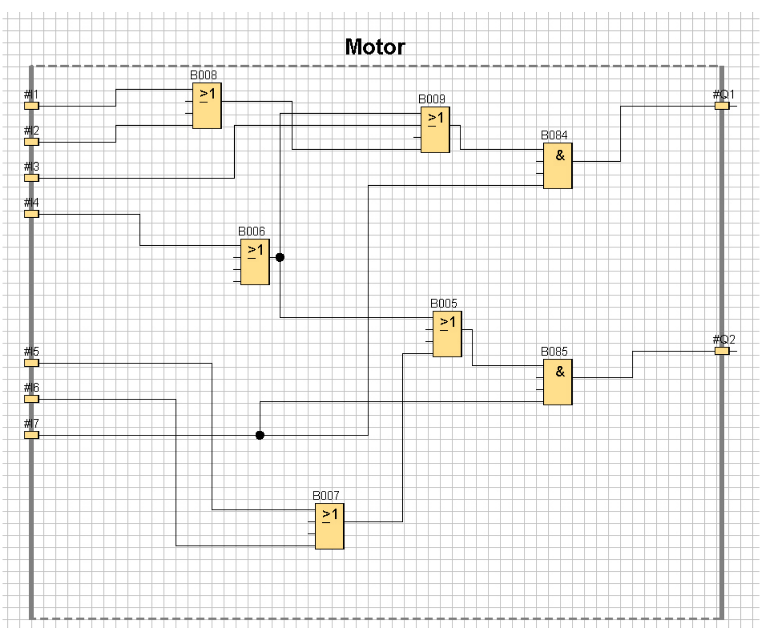


Fig. 6.104 Circuit in the UDF motor block

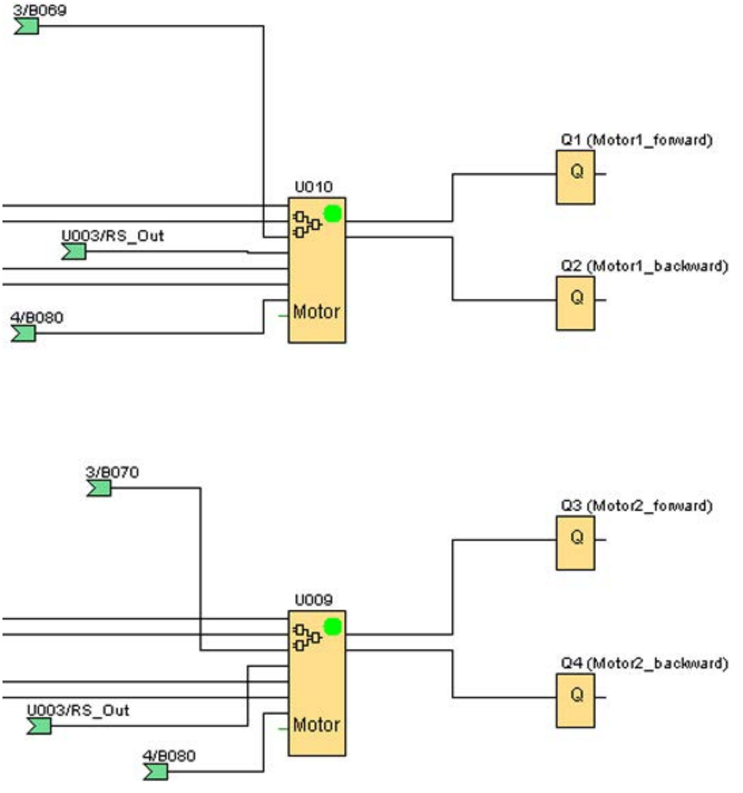


Fig. 6.105 Used UDF motor blocks in the circuit

7 Appendix

The appendix provides a brief overview of important designations of the control technology. The various types of time delays and their output assignments are also displayed.

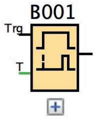
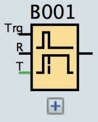
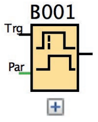
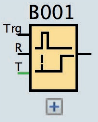
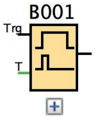
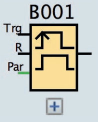
7.1 Signal designations

Table 7.1 Representation of the signal designations

Designation	Characters acc. to ISO	Characters acc. to DIN	Meaning, general and in the example (room temperature control with thermostat valve)
Process	$G_S(s)$		Process whose output variable is controlled, e.g. compressor of an air-conditioning unit with a coolant circuit to the blower
Disturbance variables	d	z	External influences affect the process, e.g. outside temperature, window position, sunlight, etc
Controlled variable	y	x	Controlled process output variable, e.g. room temperature
Measuring element			Variants: thermocouple, heat resistor, pressure load cell, load cell, e.g. thermostat: flexible material element in the thermostat valve
Measured variable	y_M	y_M	Signal: electr. voltage, e.g. thermostat: elongation of the flexible material element
Command variable	w	w	Dynamic signal as an input variable of the control circuit, e.g. thermostat: setting value on the scale
Setpoint			Specific value of the command variable, e.g. 21 °C
Actual value of the controlled variable	y	x	Generally a stationary value of the controlled variable, e.g. 24 °C (= rule deviation 3 °C)
Rule deviation	$e = w - y$	$e = w - x$	Input variable of the controller.
Controller	$G_R(s)$		Types: continuous, discontinuous, analog, digital and specially designed controllers, e.g. thermostat: flexible material element
Control variable	u	y	Output variable of a control system, e.g. position of the control dial on the control element
Final controlling element			Controller output variable and interface controller/process, e.g. thermostat: Valve in the thermostat valve
Manipulated variable	u_R	u_R	Output variable of the controller or a control device, e.g. thermostat: Position of the valve (closed to open)

7.2 Time delays

Table 7.2 List of various time delay blocks

Schematic diagram	Name and function	Output Switching: ON (1)	Output Switching: OFF (0)	Reset
	**ON delay** allows time-delayed switch-on	Trg: 0 ➔ 1 + elapse of T	Trg: 1 ➔ 0	–
	**OFF delay** allows time-delayed switch-off	Trg: 0 ➔ 1	Trg: 1 ➔ 0 + elapse of T or R: 0 ➔ 1	✓
	**ON/OFF delay** allows time-delayed switch-on/off	Trg: 0 ➔ 1 + elapse of TH	Trg: 1 ➔ 0 + elapse of TL	–
	**Retentive ON delay** After T has elapsed it saves the status 1 at Trg until R is actuated	Trg: 0 ➔ 1 + elapse of T	R: 0 ➔ 1	✓
	**Interval relay/pulse output** even when status 1 remains the same at Trg, switches the output to OFF after TL has elapsed	Trg: 0 ➔ 1	Trg: 1 ➔ 0 or elapse of TL	–
	**Interval relay, edge-triggered** If there is a pulse at Trg, it switches the output on after TL has elapsed and switches the output on after TH has elapsed or R is actuated	Trg: pulse + elapse of TL	elapse of TH or R: 0 ➔ 1	✓

Key: T/TL/TH = delay time; Trg = input (trigger); R = reset

References

Beuth: *DIN IEC 60050-351* – International Electrotechnical Vocabulary. Beuth, Berlin 2014

Graune, Uwe; Thielert, Mike; Wenzl Ludwig: *LOGO! Practical Training*. 1st edition. Siemens AG: Erlangen 2009

www.wikipedia.org/wiki/Beaufort_scale

www.conrad.com

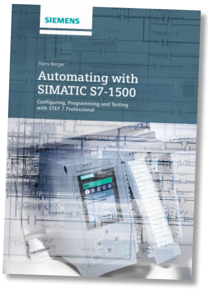
www.helmetcameracentral.com

www.maschinenform.de

www.pollin.de

http://en.wikipedia.org/wiki/Physical_property

Index



Hans Berger

Automating with SIMATIC S7-1500

Configuring, Programming and Testing with STEP 7 Professional

2014, 831 pages,
511 illustrations, 106 tables
ISBN 978-3-89578-404-0, € 89.90

The new SIMATIC S7-1500 controller with its outstanding system performance and with PROFINET as standard interface ensures extremely short system response times and the highest control quality with a maximum of flexibility for the most demanding automation tasks. With PLCopen, the controller provides standardized components for the connection of drives that support PROFIdrive without additional modules. In the book, the new automation system will be presented in detail. It describes the consistent control concept for controllers, HMI, and drives inside the TIA portal.

Nicolai Andler

Tools for Project Management, Workshops and Consulting

A Must-Have Compendium of Essential Tools and Techniques

2nd edition, 2011, 382 pages,
136 illustrations, 55 tables
ISBN 978-3-89578-370-8, € 39.90

This well accepted standard work is a unique reference work and guide for those wanting to learn about or who are active in the fields of consulting, project management and problem solving in general. It presents cookbook-style access to more than 120 most important tools, including a rating of each tool in terms of applicability, ease of use and effectiveness.

SIEMENS

Simplify your education in automation

Unique support for educators and students in educational institutions

Siemens Automation Cooperates with Education

Siemens Automation Cooperates with Education (SCE) focuses on the needs of students by providing educators with the tools and training to simplify their education in automation.
The SCE program delivers value by offering specific courses, training curriculums and exceptional Hardware and Software Trainer Packages.
We are committed to sharing knowledge, resources and tools for teaching automation and drive system technologies through partnerships with educational institutions.

Discover SCE

siemens.com/sce