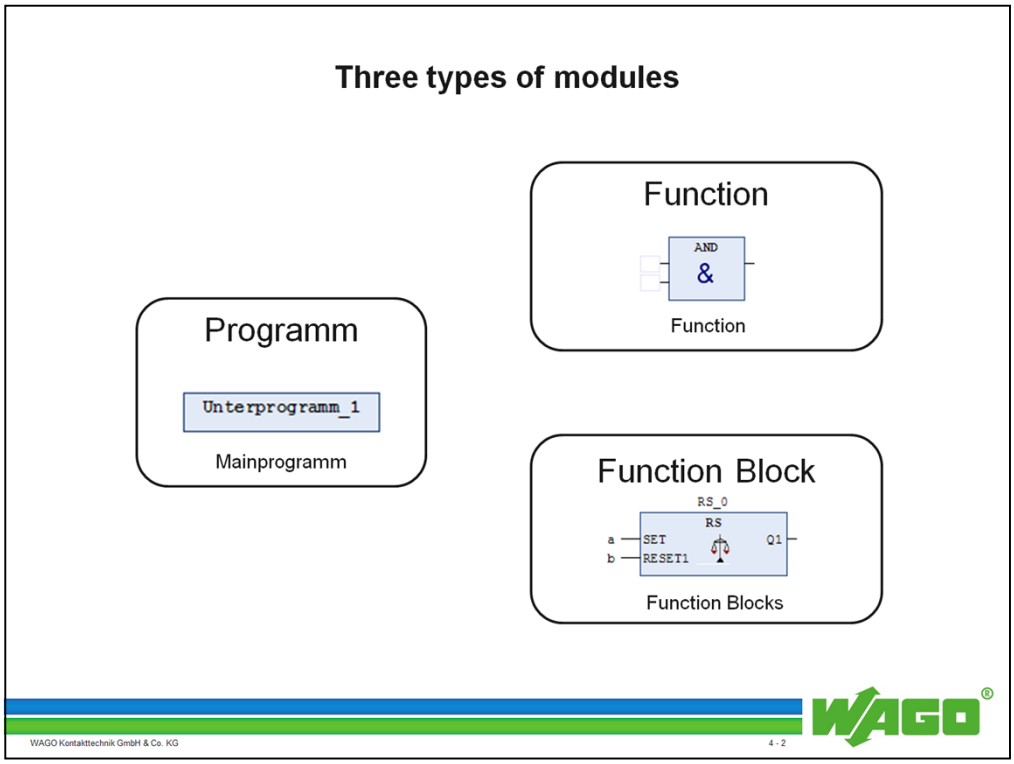
4

Programming

- Function Block types
- Exercises



A project contains different types of objects:

- Modules
- Definitions of data types,
- Display elements (visualization), and
- Resources

The first module that is created in a project is automatically named
PLC_PRG

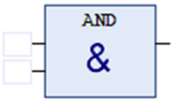
This starts the execution (corresponds to the Main function in a C program), and from here out additional components can be called like

- Programs
- Function Blocks (also called function modules)
- Functions

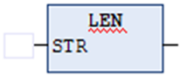
“Function” module type

Function

- A Function is a parameterizable module without memory, i.e. it always delivers the same results from the same input parameters
- A Function delivers its results back to the accumulator
- A Function has a data type
- This means that a function can only deliver a data element as a result



Operator



Function

The function in the POU

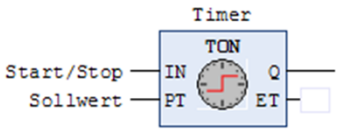
The Function is abbreviated 'FUN' and the key word is 'Function'. The function can be programmed as a parameterizable module in order to thus replace standard functions.

According to IEC 61131, it has been determined that the function may be assigned any number of input values. However, only one function value is returned. Thus, the module is used similar to an IL Instruction. Correspondingly, no output values can be assigned. Also, no Function Blocks can be called by a Function.

“Function Block” module type

Function Block

- A Function Block is a module that delivers one or more values when executed
- It can buffer values
- There are IEC FBs, Manufacturer FBs, and User FBs



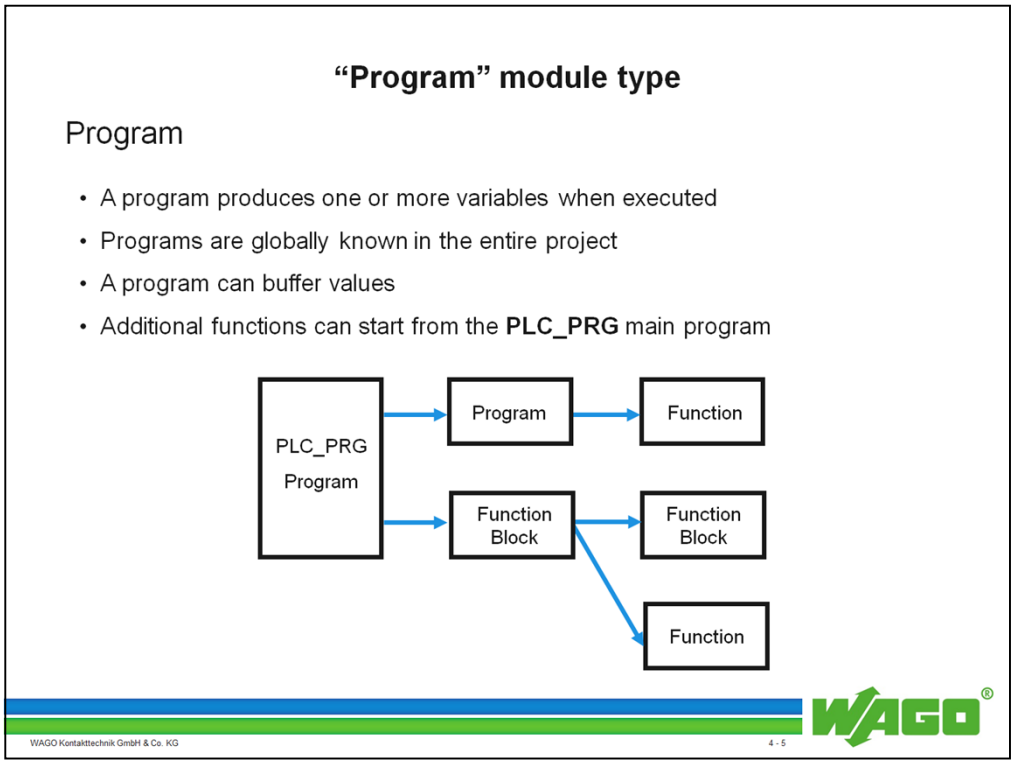
The Function Block in the POU

The Function Block is abbreviated FB. As a key word, the designation 'FUNCTION_BLOCK' is used.

In contrast to *Functions*, Function Blocks can transmit several output and in-out parameters. The parameters must be not "switched" when called (with the exception of the data instance).

Inputs (INPUT), outputs (OUTPUT), and input/output (IN/OUTPUT) signals and values can be transmitted using the function block. A memory, the instance, is assigned to the function block, which is defined according to the module start.

When calling a FB, a data instance must be assigned to it.
The data type of the data instance thereby always corresponds to the name of the FB.



The IEC 61131 modules for programming structure

Three modules types are standard in IEC 61131. The following module types are subordinate to the programming structure, which is designated as the 61131 "Program Organization Unit" (POU) in the IEC. These module types are designated as Program (PROG), Function (FUN), and Function Block (FB). A short description of the module types follows.

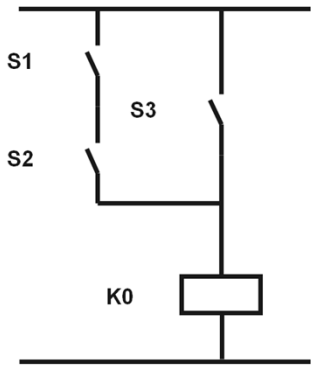
The Program

The most important part of a PLC program according to the IEC is the Program, designated by the abbreviation PROG. The key word has the designation 'PROGRAM'. This module type represents the main program, that was OB1 according to DIN 19239. Functions & Function Blocks are called in the PROG module. At the beginning of the module, the arrangement of inputs/outputs and global data are defined, which the remaining program then accesses.

Exercise 4.1

First program in LD

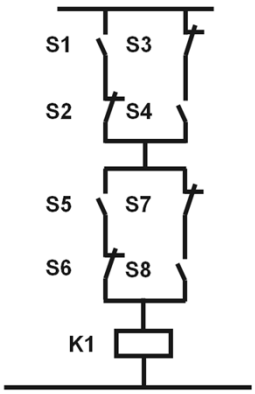
Write a program with the following function



Exercise 4.2

Program in CFC

Write the program in CFC

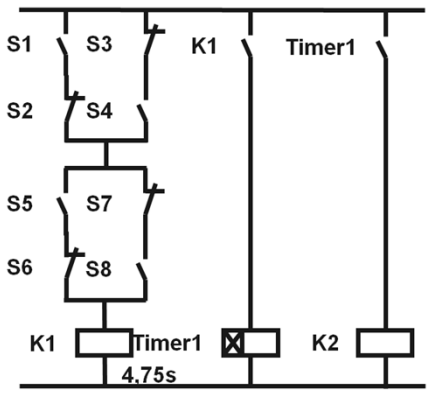


You will find an explanation of the modules "AND" and "OR" in the appendix on page 6-3.

Exercise 4.3

Program with a timer

Add an ON delay for output K2 with a 4.75 s delay time to the program from exercise 4.2.



You will find an explanation of the Timer modules in the appendix on page 6-8.

Exercise 4.4

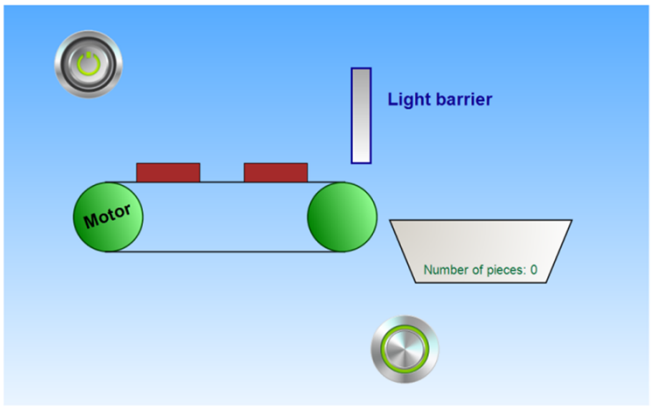
Generate a 'PLC_VISU' with the elements depicted.

The system is switched using 'S1' to "RUN" and using 'S2' to "STOP".

Using 'S3' simulates the "light sensor" that is used for counting the parts.

The "motor" runs as long as the counter has not reached the value 5 and the system is set to "RUN".

The counter is reset by using "S4" (pallet change).



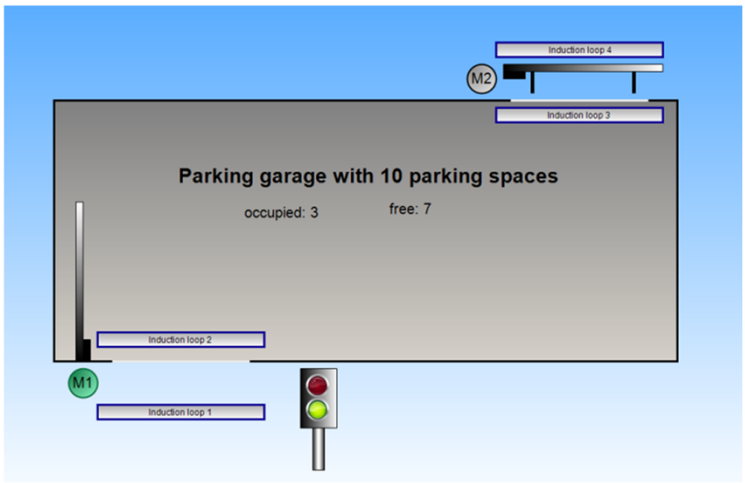
WAGO Kontakttechnik GmbH & Co. KG 4 - 9

Generate a project for controlling a conveyor belt with the following characteristics:

- The system is activated via the RUN button (switch).
- Activating the switch starts the motor for the conveyor belt (if the quantity is below the maximum value).
- The conveyor belt transports packets to a palette, which can accept a maximum of 5 packets.
  The number of packets is determined by means of a light barrier.
- If the maximum number of 5 packets is reached, then the conveyor belt stops.
- The palette is then exchanged by the system monitor for an empty one, and by actuating the palette exchange button (switch), the quantity counter is reset to zero and the conveyor belt is restarted.
- There is a STOP button to shut off the system, which shuts off the conveyor belt and deactivates the packet counter unit (The quantity status is not reset). The RUN button must be activated for restart. The RUN button must be activated for restart.

As an addition to the SPS program, generate a ***e!****COCKPIT* visualization, via which the system can be monitored or alternately operated.

Exercise 4.5 (Optional)



Add to the project for controlling a parking garage with the following characteristics:

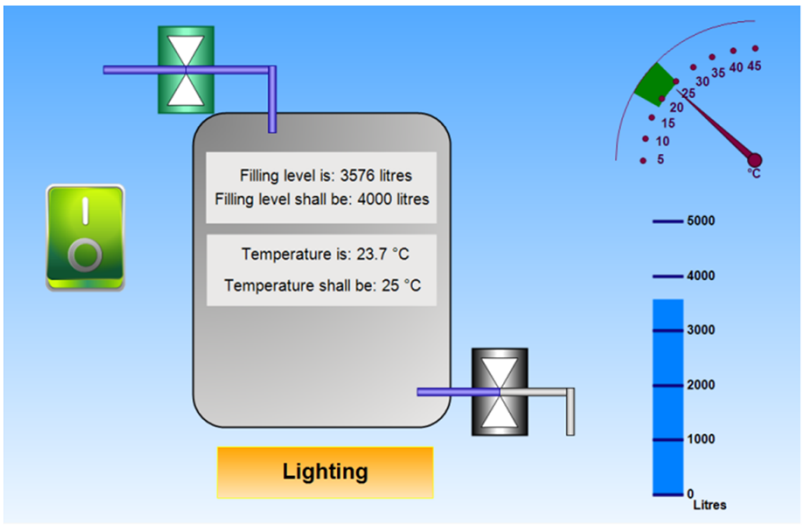
- The first barrier (M1) is opened by actuation of "Induction loop 1". By actuating "Induction loop 2", a counter (number of vehicles) is increased by one and the barrier (M1) is closed again.
- The second barrier (M2) is opened by actuation of "Induction loop 3". By actuating "Induction loop 4", a counter (number of vehicles), is reduced by one and the barrier (M2) is closed again.
- Counting may only function when the barriers are open.
- If the maximum number of 10 vehicles is reached, the traffic light at the entrance to the parking garage switches from green to red and the entrance to the parking garage is blocked.

Additional task:

- Expand the project by a variable specification (input field in VISU) of the maximum number (CAR_MAX) to be between 10-50 vehicles.
- Using the program, determine the number of available parking places and display this at the entrance to the parking garage.

Exercise 4.6

Boiler



Exercise 4.6 (description)

Boiler

Implement a boiler control, by means of which a water boiler can be filled and heated.
The system should be started/stopped using a Start/Stop button.

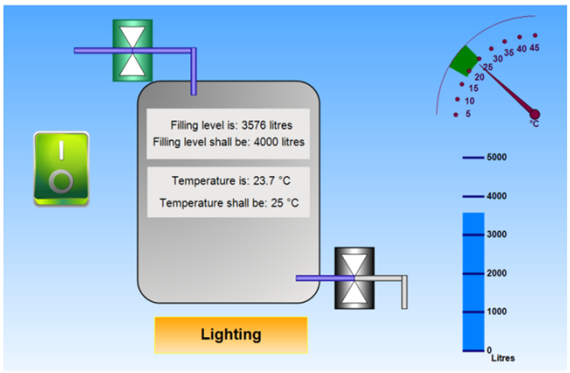
At a filling level of 4000 liters, the supply should be closed. The fill level is determined using a 0-10 V module..

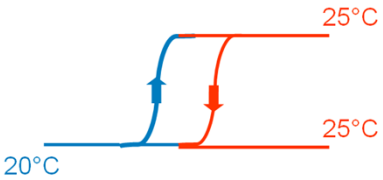
At a temperature above 25°C, boiler firing should be stopped. The temperature is determined by means of a thermocouple.

Exercise 4.7

Addition

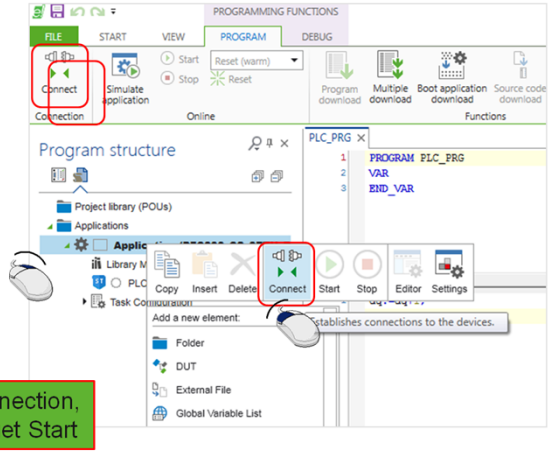
In order to protect the burner, it should be switched via a hysteresis function.





Download the program in device (RAM)

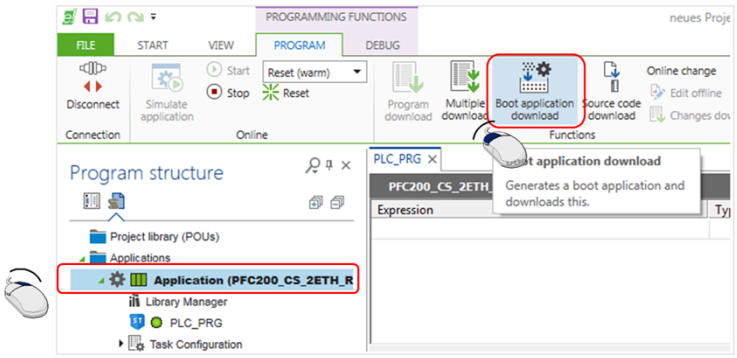
By click of the right mouse button on application, or connection under the program ribbon



After connection, don't forget Start

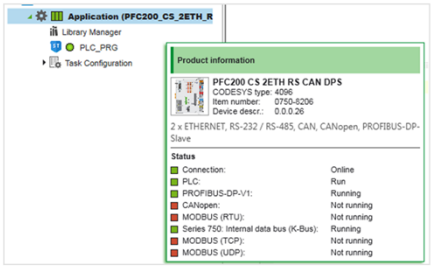
Download the program in device (Flash)

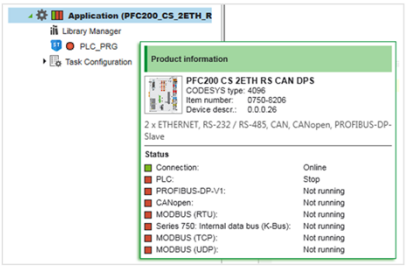
For permanent saving, boot application- download under the program ribbon



Diagnostics

The state of the controller is indicated by the status field





Left: Connection state
Center: Run/ Stop
Right: Fieldbus/ K bus state