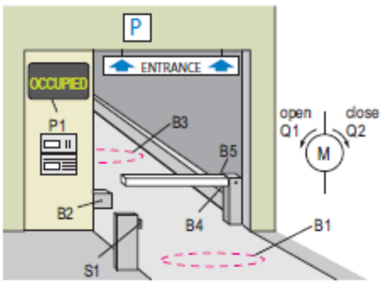
ثانوية التكنولوجيا التطبيقية
Applied Technology High School

PLC Applications

Module 1: Control Task Planning and Implementation



PREPARED BY

Academic Services Unit

January 2013

Module 1: Control Task Planning and Implementation

Module Objectives

Upon successful completion of this module, students will be able to:

- Analyze a control task and define its inputs, outputs, and other technical requirements.
- Configure control task hardware by selecting the proper LOGO! Basic module and its expansion modules.
- Develop control routines and design an Industrial Process control program from beginning to end using LOGO Soft Comfort software.
- Apply interlocks, and latches in simple control applications.

Module Contents:

	Topic	Page No.
1.1	Introduction	3
1.2	Implementation Sequence	4
1.3	Latches	10
1.4	Interlocks	11
1.5	Text messages	13
1.6	Lab Activity 1	15
1.7	Lab Activity 2	19
1.8	Lab Activity 3	20
1.9	Lab Activity 4	21
1.10	Review Exercise	22

1.1 Introduction

Small control tasks can be solved using Siemens LOGO! BASIC module with minimum hardware. As you have studied in PLC Fundamentals, the input signals supply LOGO! with information on the current state of the process and operator commands. The control relays react to these input signals in accordance with a defined program. It then generates output signals, which influence the process in the intended manner through actuators (final control elements).

For example, consider a simple control task described below.

Figure 1.1 shows a conveyor belt system that operates on 220V, and is activated by the master switch S1. Customer orders are assembled on pallets, and the orders ready for dispatch are transported on the conveyor system to the truck ramp. The two start keys or switches S2 and S3 permit transportation of the pallets; the pallets are only transported further if at least one of the start keys is kept pressed.

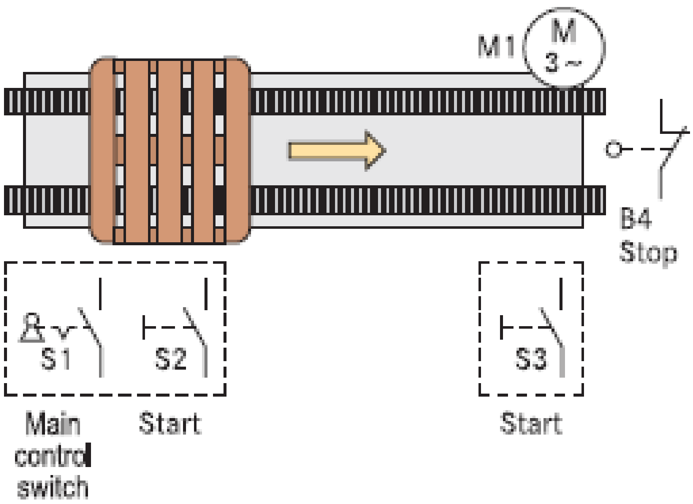


Figure 1.1 Conveyor system

The pallets are transported to the end position where they activate the limit contact B4 ("limit switch"). B4 prevents a pallet from being unintentionally transported beyond the end position and thus falling off.

1.2 Implementation Sequence

What is the right procedure for planning and implementing this control task? An implementation sequence or task flow has to be developed as shown in Fig 1.2.

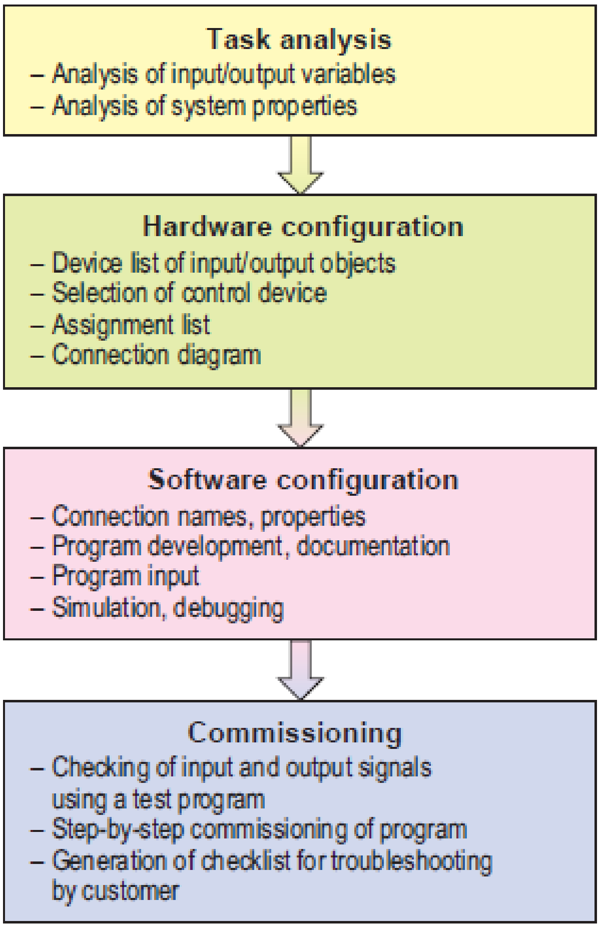


Figure 1.2: Implementation Sequence

Task Analysis:

The first step, task analysis is extremely important, and at this stage the following steps are to be followed:

- Define all inputs and outputs, and assign a variable name for each input and output.
- Analyze the system properties and its technical requirements and conditions.
- Define the relation between the outputs and inputs, and boolean expressions could be used for this purpose.

The inputs and outputs of the conveyor system mentioned before are listed below:

Inputs	
Main control switch	S1
Start pushbutton 1	S2 (N.O)
Start pushbutton 2	S3 (N.O)
Limit switch	B4 (N.C)
Output/s	
Motor	M1

Table 1.1: I/O for conveyor system

The number and type of input (N.O or N.C) and output (active high or active low) objects are based on the technical requirements of the control task. Table 1.2 shows the technical requirements for the conveyor system.

Output /s	ON-requirements	OFF-requirements
Motor (M1)	Main switch (S1) is ON **and** at least one of the start keys is pressed (S2 **or** S3) **and** limit switch (B4) is **not** pressed	Main switch (S1) is OFF **or** Limit switch (B4) is pressed

Table 1.2 Requirements for conveyor system

You must have noticed that under the ON-requirements column we **negate** the OFF requirements. In this example, pressing the limit switch (B4) is under OFF requirements, to shift this under the ON requirements it becomes B4 **not** pressed which is $\overline{B4}$ in boolean. Since the limit switch is already normally closed it becomes $\overline{\overline{B4}} = B4$.

Now to write the logic expression or the Boolean equation that describes the relation between inputs and outputs replace **AND** and **OR** by their symbols as $M1 = S1.(S2 + S3).B4$

Hardware Configuration

The second stage is hardware configuration, and at this stage a device list can be used to assist selection of the LOGO! Controller and expansion modules. In table 1.2 it is recorded that 4 digital inputs and one digital output is required for the conveyor system, and also all inputs are to be connected to 220 V. So basically, a 115/240V LOGO! Basic module has to be selected for this application.

LOGO! basic module									
Versions		Basic	Pure	Clock		Digital			Analog
		With keypad	Without keypad			8 inputs	4 outputs		
		With display	Without display	With	Without		Transistor	Relay	2 inputs
12 V/24 V	AC/DC				■		■		■
24 V	DC			■				■	
12 V/24 V	DC				■		■		
115 V/240 V	AC/DC	X			■	X	■	X	■

■ Selection is not available

Table 1.3: LOGO! Basic module selection

The following expansion modules can be connected to the LOGO! just in case the control task requires more inputs or outputs, either digital or analog.

Name	Power supply	Inputs	Outputs
LOGO! DM 8 12/24R	12/24 V DC	4 digital	4 relays (5A)
LOGO! DM 8 24	24 V DC	4 digital	4 solid state 24 V / 0.3 A
LOGO! DM 8 24R (3)	24 V AC/DC	4 digital	4 relays (5A)
LOGO! DM 8 230 R	115...240 V AC/DC	4 digital (1)	4 relays (5A)
LOGO! AM 2	12/24 V DC	2 analog 0 ... 10 V or 0 ... 20 mA (2)	none
LOGO! AM 2 PT100	12/24 V DC	2 Pt100 –50 °C to +200 °C	none

Table 1.4: Expansion modules

(This table is used when there is a need to select expansion module(s). You do not need to memorize this table)

After selecting the proper LOGO! an **assignment list** should be created, in this list, all input and output objects are addressed and assigned to LOGO! inputs and outputs as shown in table 1.5. A logical assignment of inputs and outputs is necessary for both the installation (hardware connection) and for generation of the program.

Inputs	
Input	Assigned address
S1	I1
S2 (N.O)	I2
S3 (N.O)	I3
B4 (N.C)	I4
Output/s	
Output	Assigned address
M1	Q1

Table 1.5: Assignment list

The third stage is **software configuration**. Before the control program can be developed, the project data should be entered in the "Properties" window, which can be displayed in the pull-down menu "File".

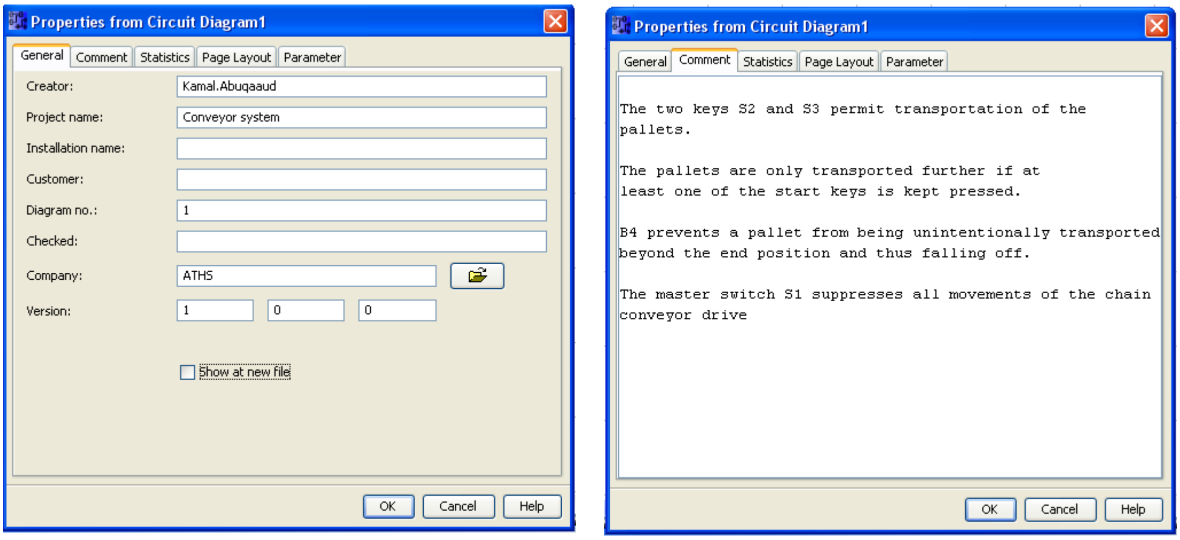


Figure 1.3:Properties window

It is very helpful to produce a connection table (Figure 1.4), but it integrates the assignment list in the project, and names are assigned to the objects which are much more appropriate during generation and checking of the program than the addresses themselves.

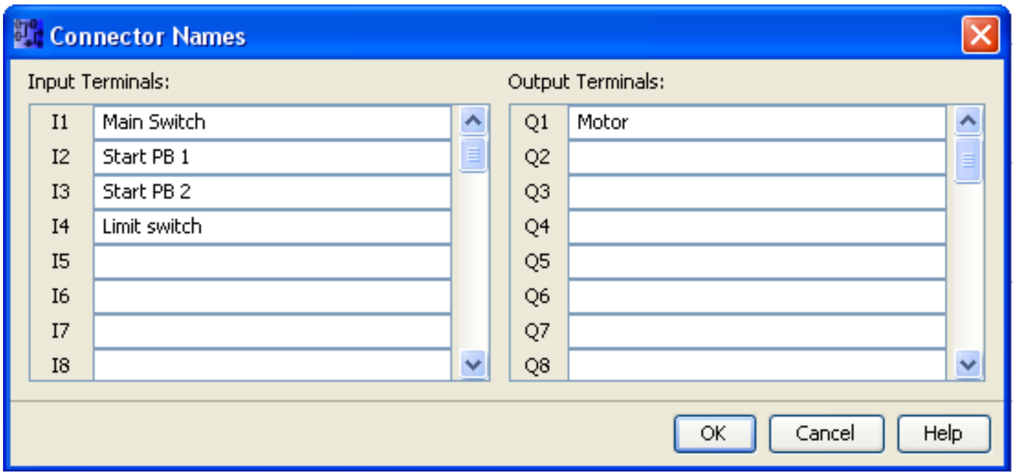


Figure 1.4: Connection table

The connection table is opened by selecting the menu "Edit" and then "Input/Output Names".

In order to program the LOGO! the program is initially developed according to the control task as a draft on paper. However, the program can be produced directly using the LOGO! Soft Comfort software on a PC/notebook. Correct functioning of the program can then be directly checked through simulation. Any errors can thus be corrected in advance.

During simulation, it is very important to simulate the inputs properly by selecting the correct option. In our example starting keys S2 and S3 are make pushbuttons (N.O PBs) while the limit switch B4 is break ushbutton, see Figure 1.5.

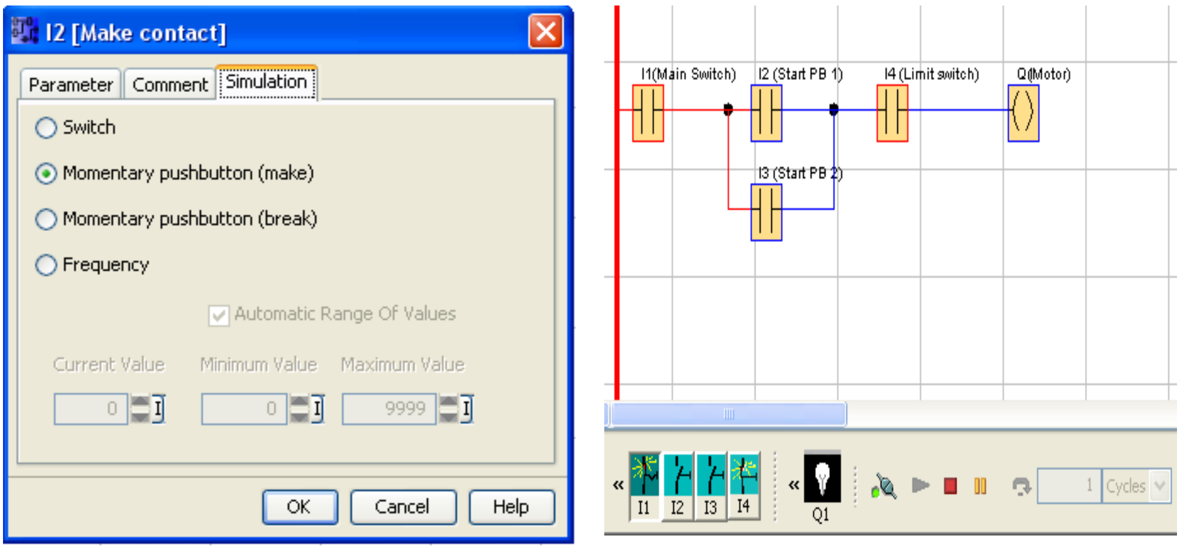


Figure 1.5: Simulation

The relationship between hardware and software is coordinated during the last stage, which is commissioning, and the system is optimized with faults eliminated.

Conduct lab activity 1.

1.3 Latches

In some applications, we need to use the transient close/open buttons to start and stop the equipment. Additionally, to maintain its continuous action, latches are needed. A latch is like a **sticky switch**, when pushed it will turn ON, but stick in place; it must be pulled to turn OFF. A latch in ladder logic uses one contact to latch, and a second to unlatch. Figure 1.6 shows 2 different latches used to control a motor.

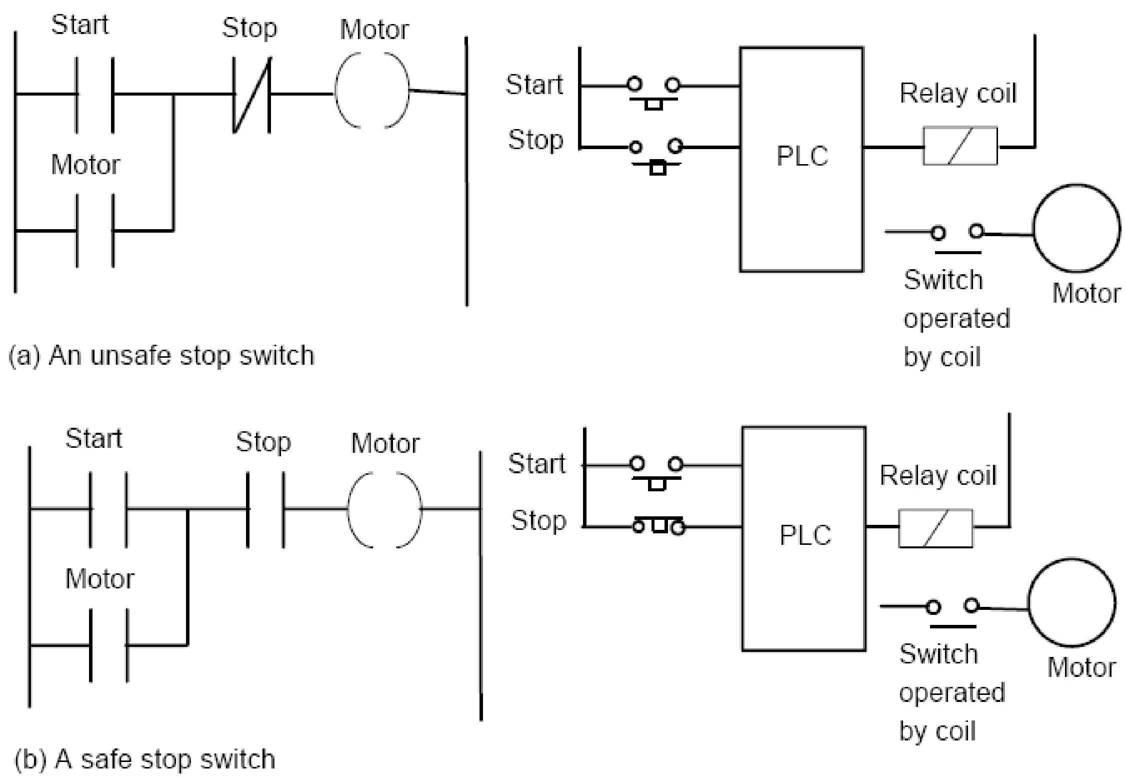


Figure 1.6: Latches used to start and stop a motor

For safety reasons it is preferred to use a normally close stop pushbutton to stop a motor.

Conduct lab activity 2.

1.4 Interlocks

Have you ever tried to insert a coin inside a vending machine and order more than one item? The machine gives you only one item, because it uses interlocks.

Interlocks are used to ensure two incompatible events cannot occur at the same time.

Figure 1.7: Vending machine

Interlocking is holding a system operation until certain conditions are met. These are often required for safety on industrial equipment to protect workers. a good example is in **reversible motor control**, where two motor contactors are wired to switch polarity (or phase sequence) to an electric motor, and we don't want the forward and reverse contactors energized at the same time.

- For the protection of persons, interlocking must be hardware-based.
- For the protection of machines and products, interlocks can be hardware based and/or software-based.

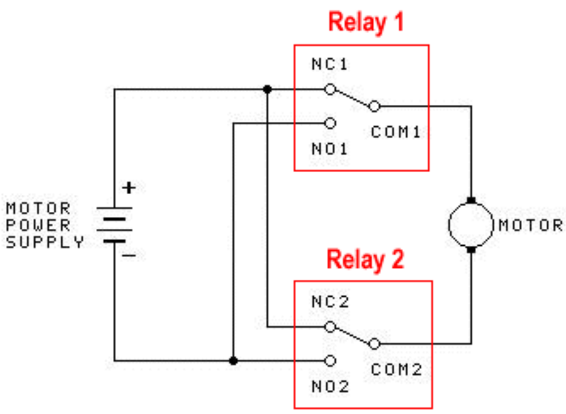


Figure 1.8: Hardware interlock

Software-based Interlock

Consider the application below:

Two normally open pushbuttons are used to turn 12 VDC motor ON; one will run the motor in the forward direction and the other in the reverse direction, and a normally closed pushbutton is used to turn the motor OFF.

Task Analysis:

Inputs	
Input	Assigned address
PB1 (N.O)	I1
PB2 (N.O)	I2
PB3 (N.C)	I3
Output/s	
Output	Assigned address
Forward direction	Q1
Reverse direction	Q2

Table 1.6 Assignment list, reversible Motor

Forward direction requirements	Reverse direction requirements
PB1 is pressed **AND** PB3 is **NOT** pressed and the motor is **NOT** running in the reverse direction.	PB2 is pressed **AND** PB3 is **NOT** pressed and the motor is **NOT** running in the forward direction.

Table 1.7 System requirements

Since we have two outputs we need two Boolean expressions:

$$Q1 = (I1 + Q1).I3.\overline{Q2}$$

$$Q2 = (I2 + Q2).I3.\overline{Q1}$$

Figure 1.9 shows the ladder diagram for reversible motor control task; latches are used in this diagram to maintain continuous forward and reverse actions, while interlocks are used to ensure forward and reverse actions cannot occur at the same time.

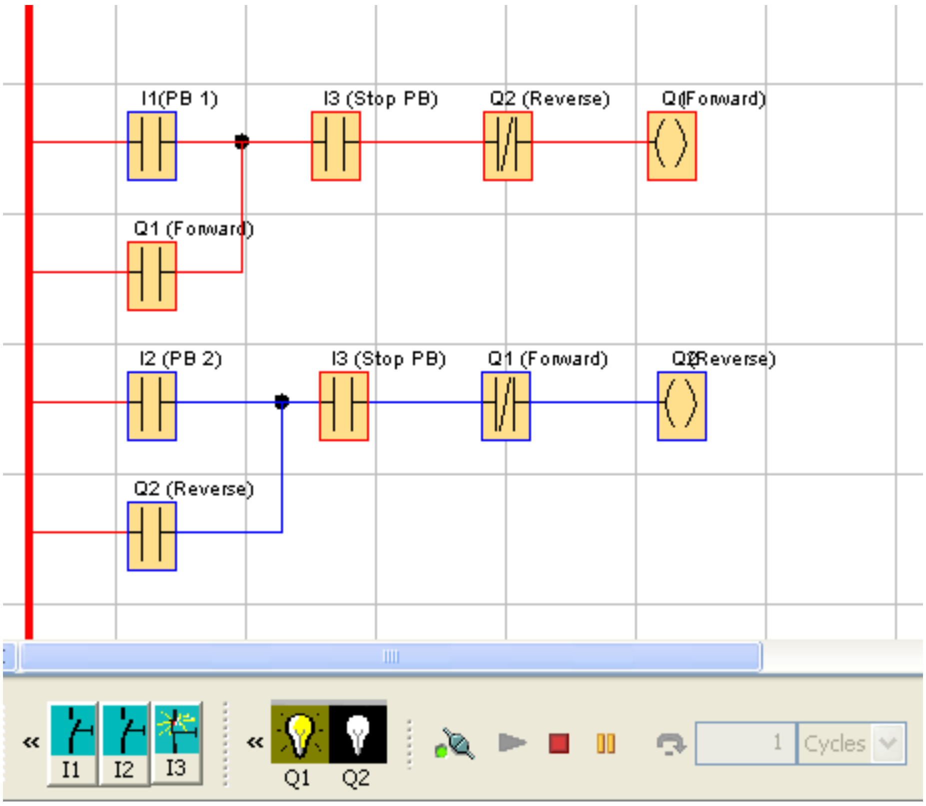


Figure 1.9: LAD for reversible Motor control task

Conduct lab activity 3.

1.5 Text messages

"Message texts" programming block is used to show text messages on the LOGO! display unit, Input **En** triggers the output and shows the message while **P** is the message priority.

Acknowledgement disabled (Ack = Off):

The message text is hidden with a 0 to 1 signal transition at input En.

Acknowledgement enabled (Ack = On):

After input En is reset to 0, the message text is displayed until acknowledged by pressing the OK button. The message text cannot be acknowledged as long as input En is high.

Connection	Description
Input **En**	A 0 to 1 transition at En (Enable) triggers the output of the message text.
Input **P**	P is the priority of the message text. 0 is the lowest, 30 the highest priority. Quit: Acknowledgement of the message text
Parameter	**Text:** Input of the message text **Par:** Parameter or actual value of another, already configured function (see "Visible parameters or actual values") **Time:** Shows the continuously updated time-of-day **Date:** Shows the continuously updated date **EnTime:**Shows the time of the 0 to 1 transition **EnDate:**Shows the 0 to 1 transition of the date
Output **Q**	Q remains set as long as the message text is queued.

Table 1.8 Message texts block

Conduct lab activity 4.

1.6 Lab Activity 1

Objective: Analyze a control task for a sorting machine, and develop and implement a practical solution for the task.

A conveyor belt starts and stops using green and red pushbuttons respectively, green light goes ON to indicate that conveyor belt is moving. If any object is detected by an inductive sensor white light goes ON until the object is sorted out by a branching arm and detected by a fiber optic barrier. The conveyor belt stops if any of the following occurs:

- The stop red pushbutton is pressed.
- The number of the metallic objects detected is 4
- The optical sensor at the end of the conveyor belt detects 5 non-metallic objects.

Use The Edutrainer prototype production line to implement this control task.

1. Create an I/O assignment list.

Inputs	
Input	Address
Outputs	
Output	Address

2. Analyze the system requirements and write the Boolean expression for each output.

System requirements		
Output	Requirements	Boolean expression

3. Draw the FBD for this control task.

   Note : pressing the green pushbutton should reset all counters

4. Use the LOGO! Soft comfort software to solve this task.
   While solving this control task you are required to produce a connection table and to describe the task in the properties window.

5. Simulate the program and fill in the table provided below by writing either **ON** or **OFF**:

   Conditions:
   Green PB is pressed and then released
   White PB is not pressed
   All sensors are inactive

Output	Conveyor belt	Branching arm	Green indicator	White indicator
Status				

   Conditions:
   Green PB is pressed and then released
   White PB is not pressed
   Only inductive sensor is active

Output	Conveyor belt	Branching arm	Green indicator	White indicator
Status				

Conditions:

Green PB is pressed and then released

White PB is not pressed

All sensors are inactive , 4 metallic objects are sorted out

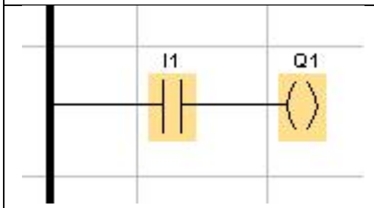
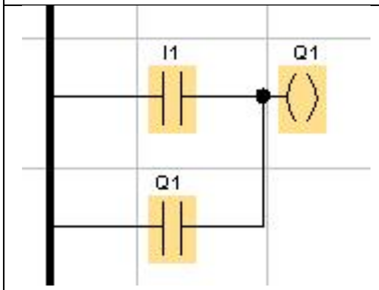
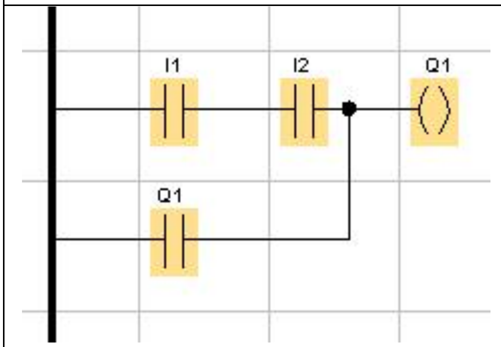
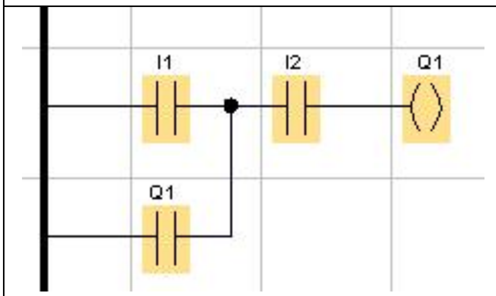
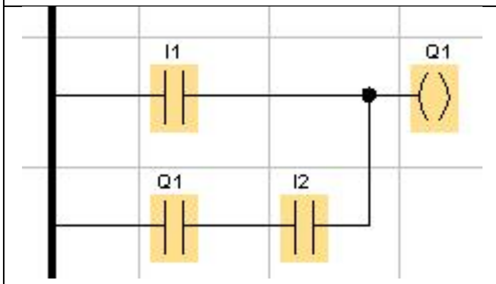
Output	Conveyor belt	Branching arm	Green indicator	White indicator
Status				

6. Run and test the program.

1.7 Lab Activity 2

Objective: Use latches in control applications.

Try the following control routines, and write down your comments:

Ladder diagram	Actions	Comments
	Press and release the green PB.	
	Press and release the green PB.	
	Press and release the green PB. Then press the red one.	
	Press and release the green PB. Then press the red one.	
	Press and release the green PB. Then press the red one.	

1.8 Lab Activity 3

Objective: Use Interlocks in control application.

Use the Edutrainer kit and LOGO! soft comfort software to create and implement the following control task:

Green PB and white PB are used to move the Edutrainer table in the forward and backward directions respectively while the red one is used to stop it, table keeps moving after releasing green or white PB till it reaches any of the proximity switches, once the table is detected by a proximity switch it stops automatically.

Software- based interlock must be used to ensure that forward and backward cannot occur at the same time.

1. Draw the ladder diagram for the previous control task.

2. Run and test your program.

1.9 Lab Activity 4

Objective: Use the message text block.

Repeat the previous task and use the "Message texts" programming block to show "Moving Forward" when table is moving in the forward direction and "Moving backward" when table is moving in the backward direction and "Stop mode" when table is not moving.

1. Draw the ladder diagram for the previous control task.

2. Run and test your program.

1.10 Review Exercise

1. Two barriers are each equipped with a reversible motor (Q1 and Q3 open, Q2 and Q4 close), a pillar with a key switch for the entrance, and the following sensors (NC contacts):
   - B1, B6: induction loops in front of the barriers
   - B2, B7: reflex sensors for detection of objects "underneath" the barriers
   - B3, B8: induction loops behind the barriers

   The barriers are equipped with end contacts (NO contacts) for detection of the bottom (B4, B9) and top end positions (B5, B10). In addition, the entrance has a display P1 "Occupied".

   If a vehicle approaches, the barrier opens if the following conditions are satisfied:

Entrance	• Key switch S1 has been activated and • Induction loop B1 has a "1" signal.
Exit	• Induction loop B6 has a "1" signal.

The barriers open up to the end position and cannot be closed as long as the reflex sensors B2 and B7 detect an object ("1" status). The areas are closed by activation of the induction loops B3 and B8. The signal lamp "Occupied" signals that all spaces are full. The entrance is then blocked.

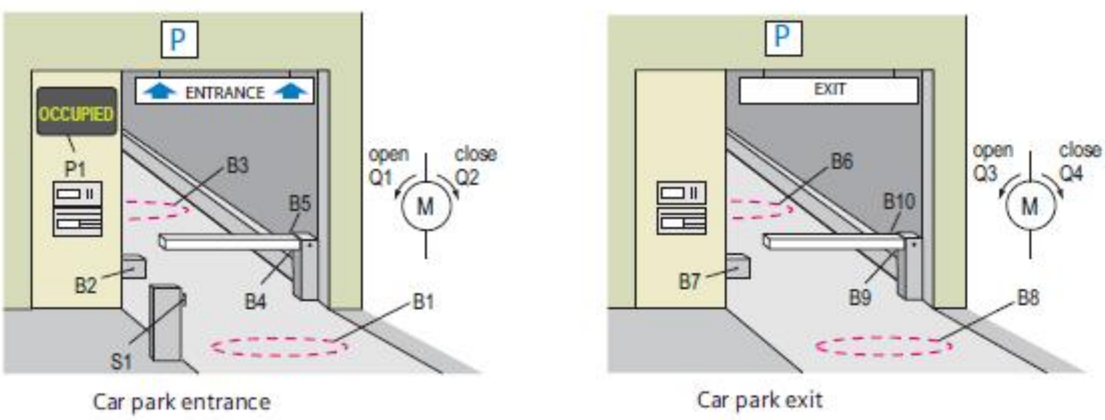


Figure 1.11: Car park with an access control system

1. Create an I/O assignment list.

2. Analyze the system requirements and draw the FBD for this control task.

3. Simulate and test your program using the LOGO! Soft comfort software.

2. In a simple vending machine when you put in a Dirham, the coin detector generates a “high” signal and this allows you to select one item out of 4 different items all are of the same price (1 Dirham) , 4 pushbuttons are used to order these items; one for each item. When you press any button to get something (say, juice can), the PLC turns on the motor connected to the spiral that hold the can. The spiral spins and drops the can. 4 motors are used for items delivery. When the can comes out it is detected by a light sensor. The light sensor then stops the spinning motor.

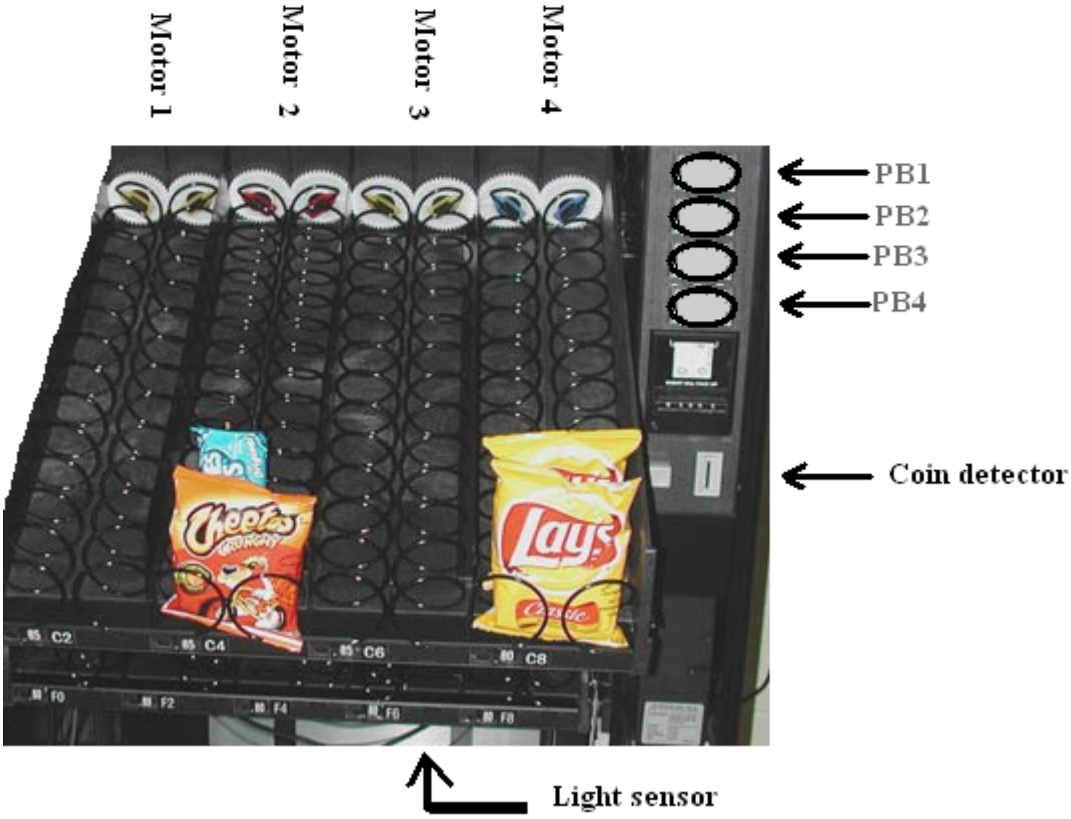


Figure 1.12: Simple vending machine.

1. Create an I/O assignment list.

2. Analyze the system requirements and draw the ladder diagram for this control task.

3. Simulate and test your program using the LOGO! Soft comfort software.