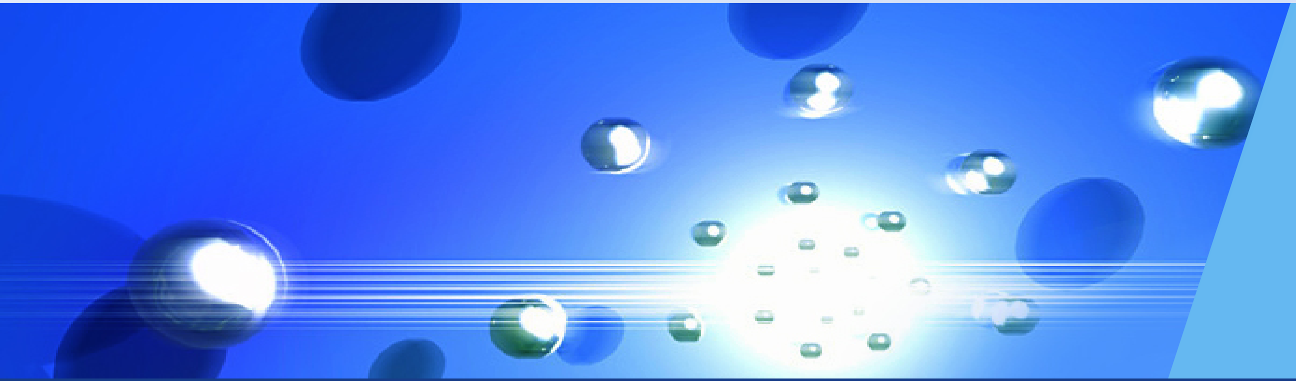
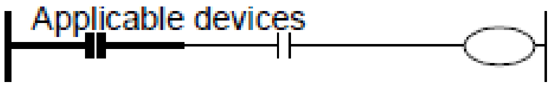
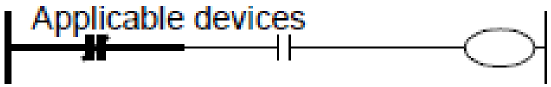
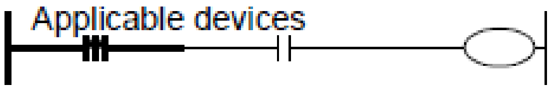
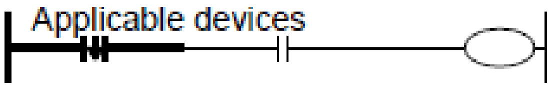
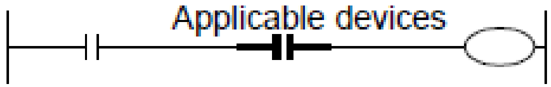
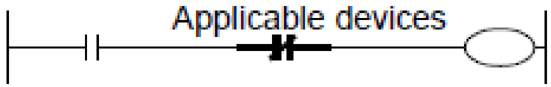
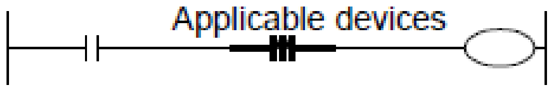


Chapter 8: PLC Programming Language – Basic Instructions

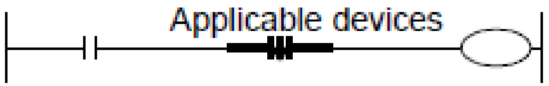
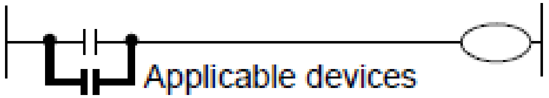
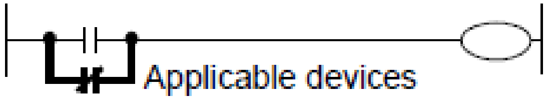
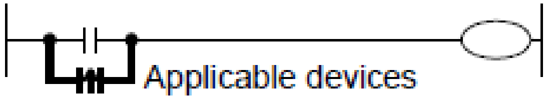
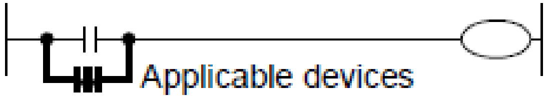
Basic Instructions

Input (Contact) instructions

Mnemonic	Name	Symbol	Function	Applicable devices
LD	Load		Initial logical operation contact type NO (normally open)	X,Y,M,S,D□.b,T,C
LDI	Load Inverse		Initial logical operation contact type NC (normally closed)	X,Y,M,S,D□.b,T,C
LDP	Load Pulse		Initial logical operation of Rising edge pulse	X,Y,M,S,D□.b,T,C
LDF	Load Falling Pulse		Initial logical operation of Falling/trailing edge pulse	X,Y,M,S,D□.b,T,C
AND	AND		Serial connection of NO (normally open) contacts	X,Y,M,S,D□.b,T,C
ANI	AND Inverse		Serial connection of NC (normally closed) contacts	X,Y,M,S,D□.b,T,C
ANDP	AND Pulse		Serial connection of Rising edge pulse	X,Y,M,S,D□.b,T,C

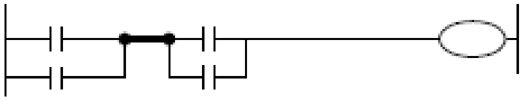
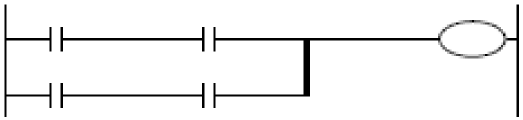
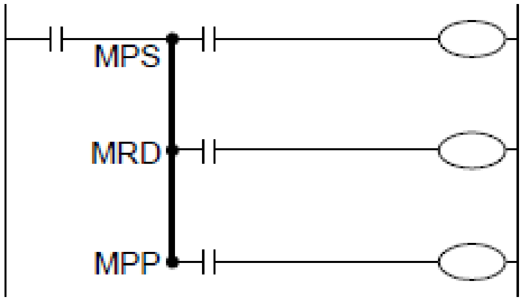
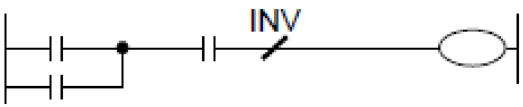
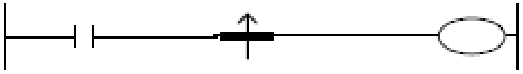
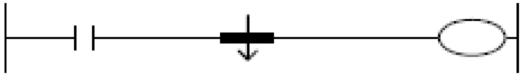
Basic Instructions

❖ Input (Contact) instructions

ANDF	AND Falling Pulse		Serial connection of Falling/trailing edge pulse	X,Y,M,S,D□.b,T,C
OR	OR		Parallel connection of NO (normally open) contacts	X,Y,M,S,D□.b,T,C
ORI	OR Inverse		Parallel connection of NC (normally closed) contacts	X,Y,M,S,D□.b,T,C
ORP	OR Pulse		Parallel connection of Rising edge pulse	X,Y,M,S,D□.b,T,C
ORF	OR Falling Pulse		Parallel connection of Falling/trailing edge pulse	X,Y,M,S,D□.b,T,C

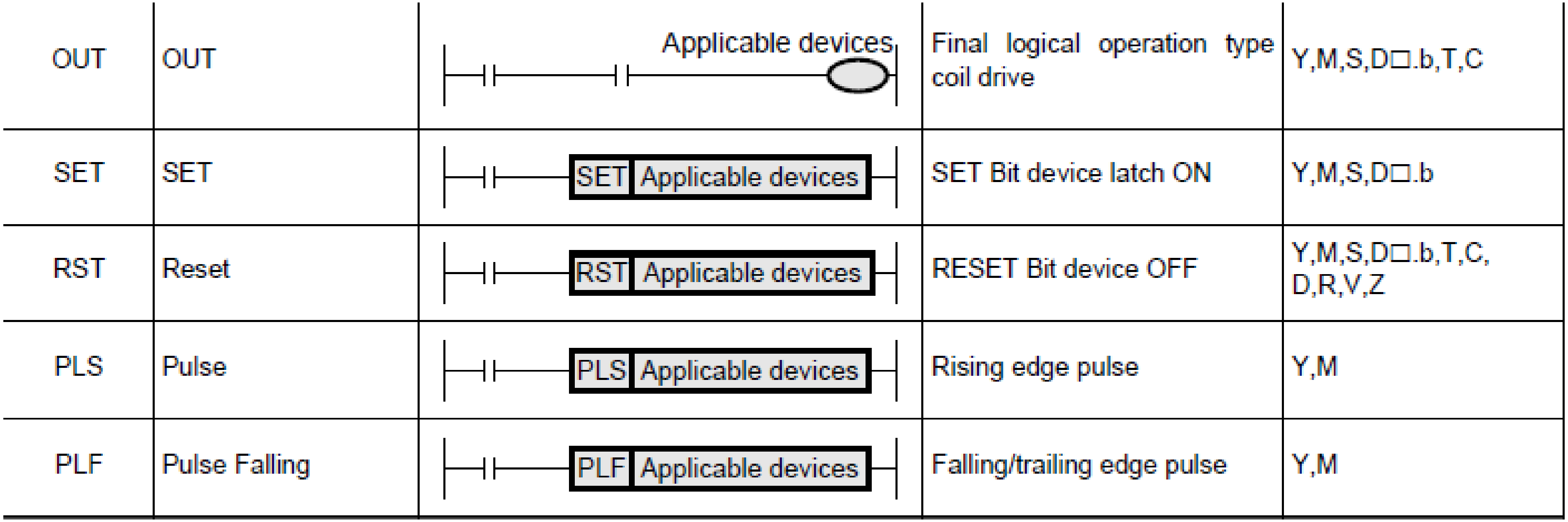
Basic Instructions

Connection instructions

ANB	AND Block		Serial connection of multiple parallel circuits	–
ORB	OR Block		Parallel connection of multiple contact circuits	–
MPS	Memory Point Store		Stores the current result of the internal PLC operations	–
MRD	Memory Read		Reads the current result of the internal PLC operations	
MPP	Memory POP		Pops (recalls and removes) the currently stored result	
INV	Inverse		Invert the current result of the internal PLC operations	–
MEP	MEP		Conversion of operation result to leading edge pulse	–
MEF	MEF		Conversion of operation result to trailing edge pulse	–

Basic Instructions

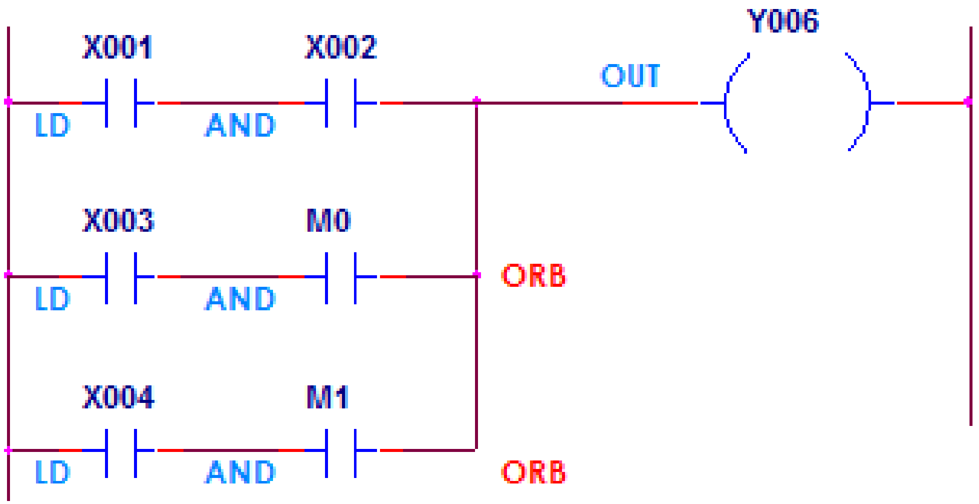
❖ Output instructions



OUT	OUT	Applicable devices	Final logical operation type coil drive	Y,M,S,D□.b,T,C
SET	SET	SET Applicable devices	SET Bit device latch ON	Y,M,S,D□.b
RST	Reset	RST Applicable devices	RESET Bit device OFF	Y,M,S,D□.b,T,C, D,R,V,Z
PLS	Pulse	PLS Applicable devices	Rising edge pulse	Y,M
PLF	Pulse Falling	PLF Applicable devices	Falling/trailing edge pulse	Y,M

Basic Instructions

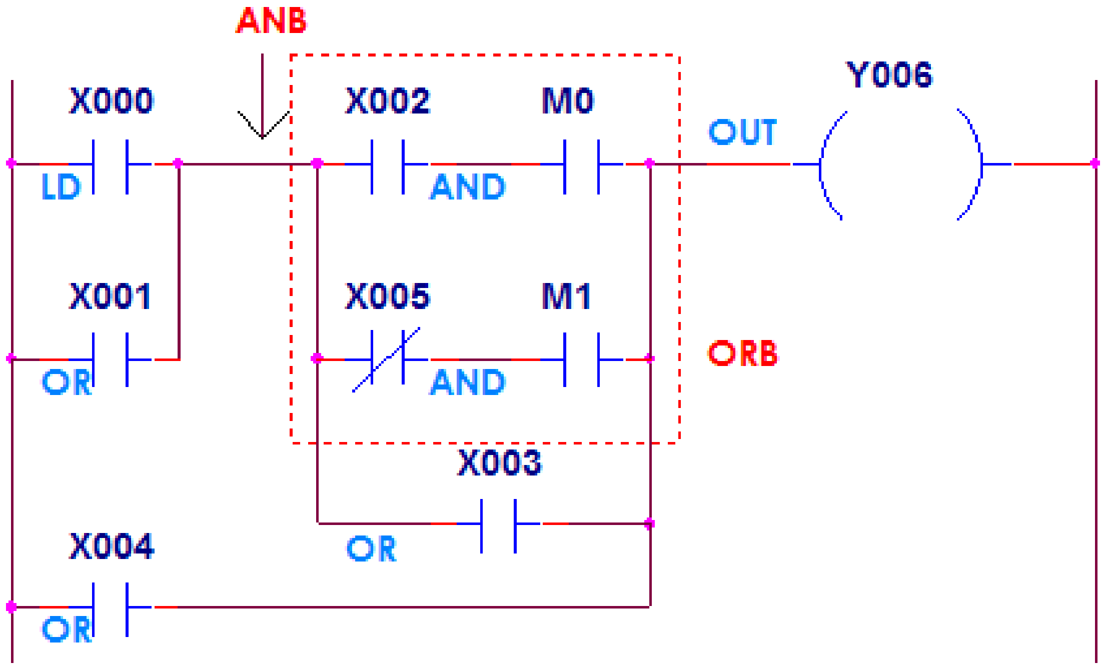
.



Lập trình theo phương pháp tuần tự

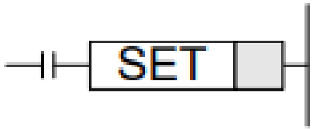
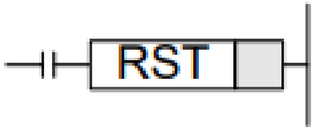
```
LD   X001
AND       X002
LD   X003
AND       M0
ORB
LD   X004
AND       M1
ORB
OUT Y006
```

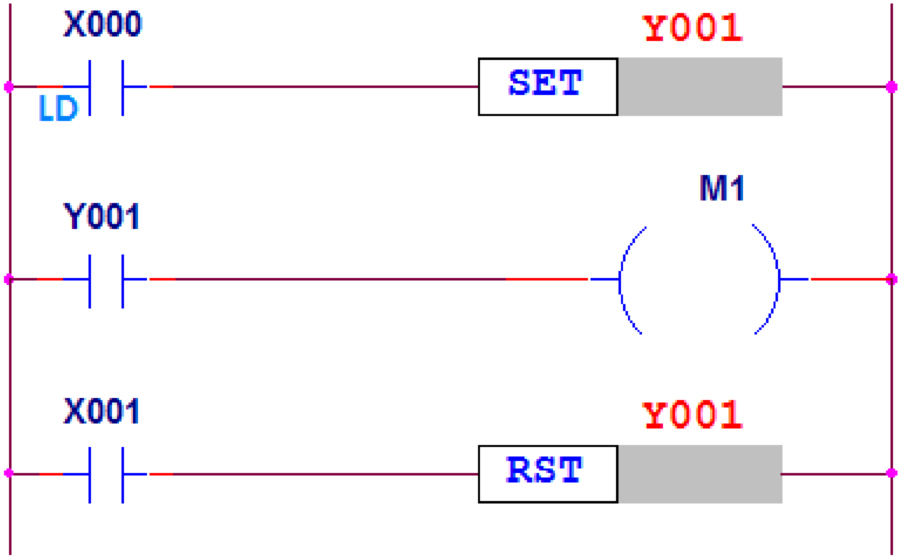
Basic Instructions



```
LD X000
OR X001
LD X002
AND   M0
LDIX005
AND   M1
           ORB
OR X003
           ANB
OR X004
OUT    Y006
```

SET and RESET instructions

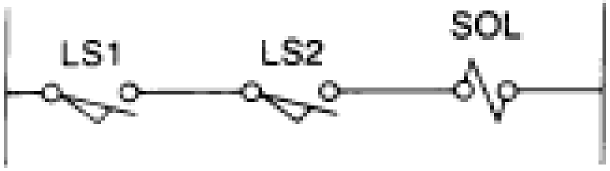
SET *(SET)*	Sets a bit device permanently ON		Y, M, S
RST *(ReSeT)*	Resets a bit device permanently OFF		Y, M, S, D, V, Z (see section 2.16 for timers and counters T,C)



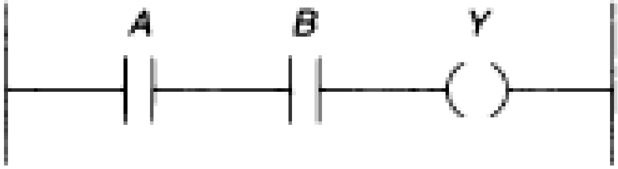
```
LD X000
SET    Y001
LD Y001
OUT    M1
. . .
LD X001
RSTY001
```

Relay schematic vs. Ladder logic program

Relay schematic

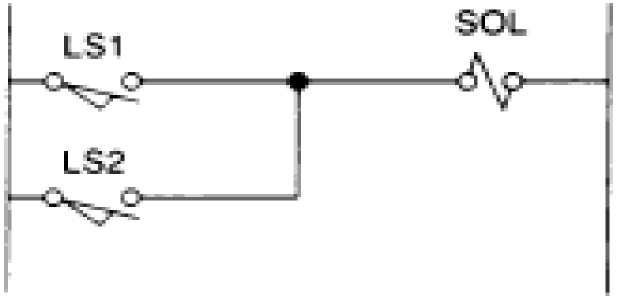


Ladder logic program

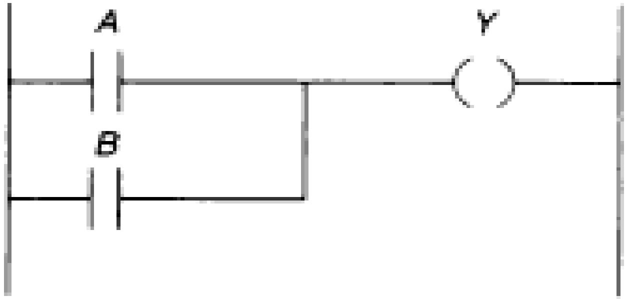


Boolean equation: $AB = Y$

Relay schematic

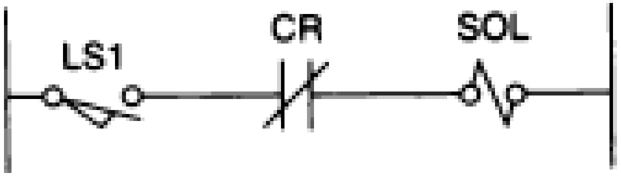


Ladder logic program

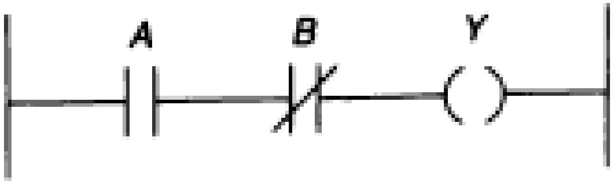


Boolean equation: $A + B = Y$

Relay schematic



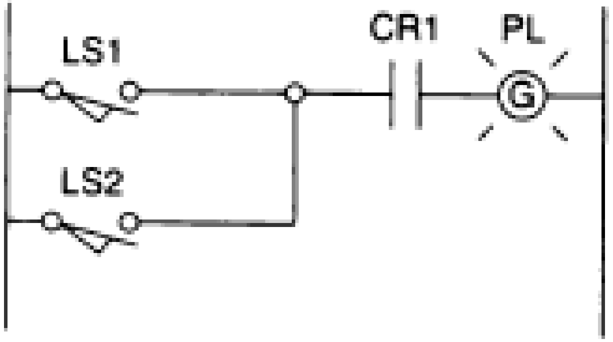
Ladder logic program



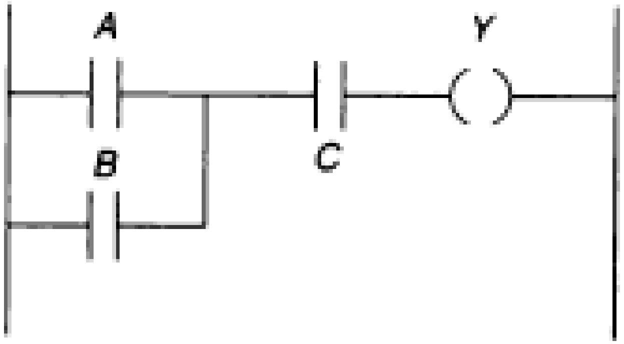
Boolean equation: $A\bar{B} = Y$

Relay schematic vs. Ladder logic program

Relay schematic

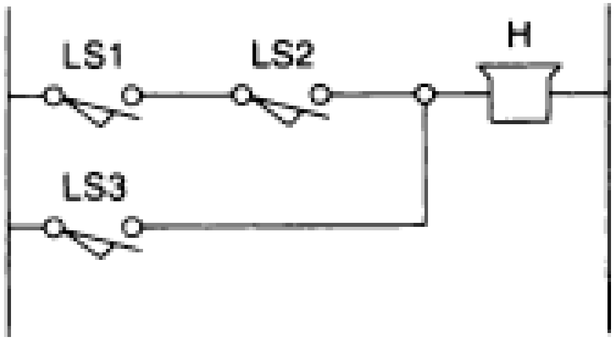


Ladder logic program

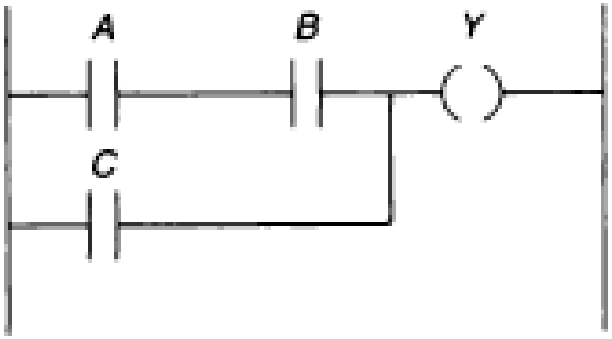


Boolean equation: $(A + B)C = Y$

Relay schematic



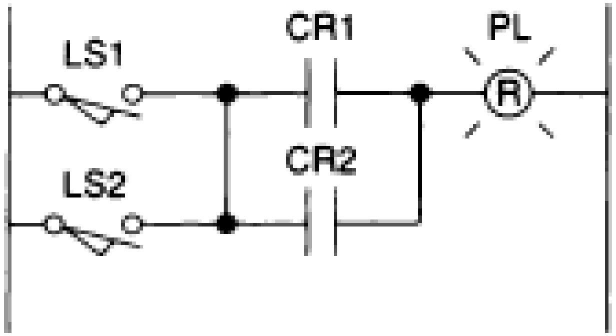
Ladder logic program



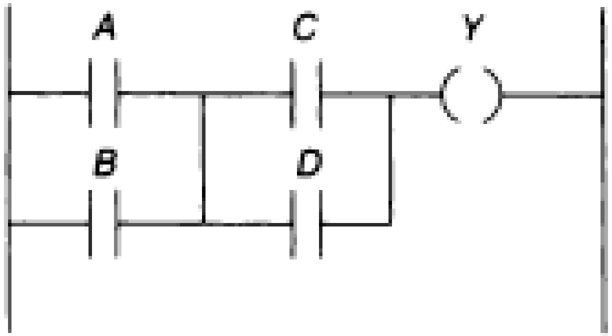
Boolean equation: $(AB) + C = Y$

Relay schematic vs. Ladder logic program

Relay schematic

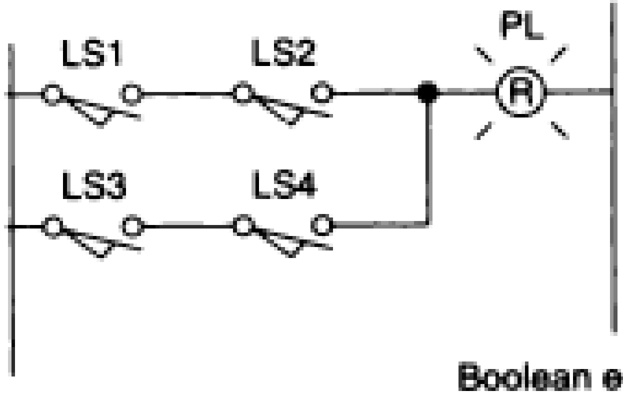


Ladder logic program

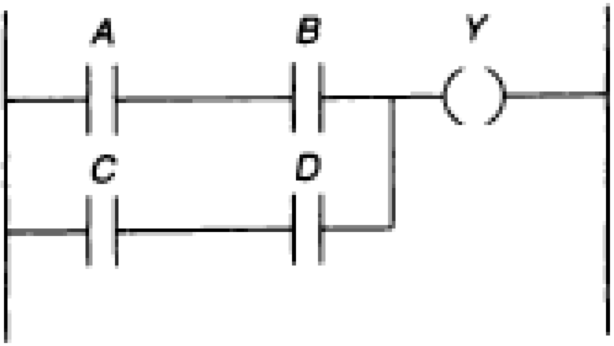


Boolean equation: $(A + B)(C + D) = Y$

Relay schematic



Ladder logic program



Boolean equation: $(AB) + (CD) = Y$

Quiz

- Express each of the following equations as a ladder logic rung. Program each rung into the PLC software (GX Developer) and prove its operation.

a) $\mathrm{Y0} = (A + B)CD$

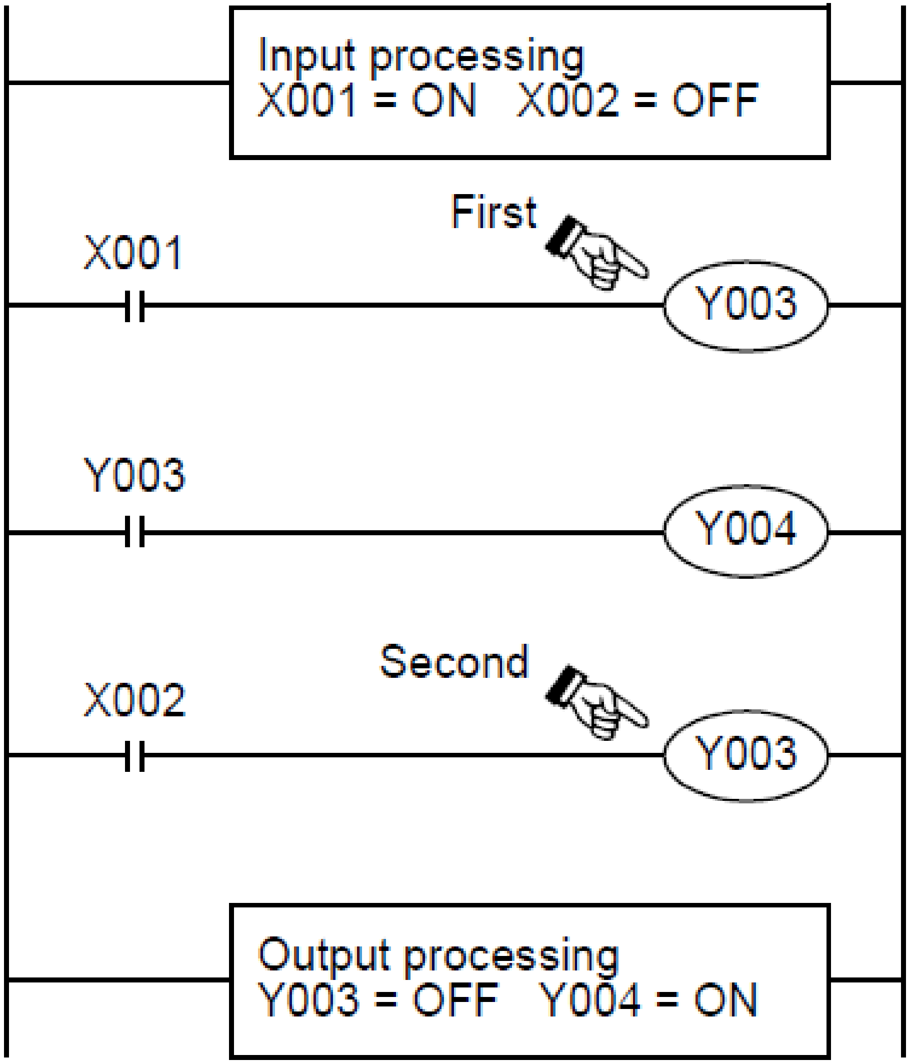
b) $\mathrm{Y0} = (A\bar{B}C) + \bar{D} + E$

c) $\mathrm{Y0} = [(\bar{A} + \bar{B})C] + D\bar{E}$

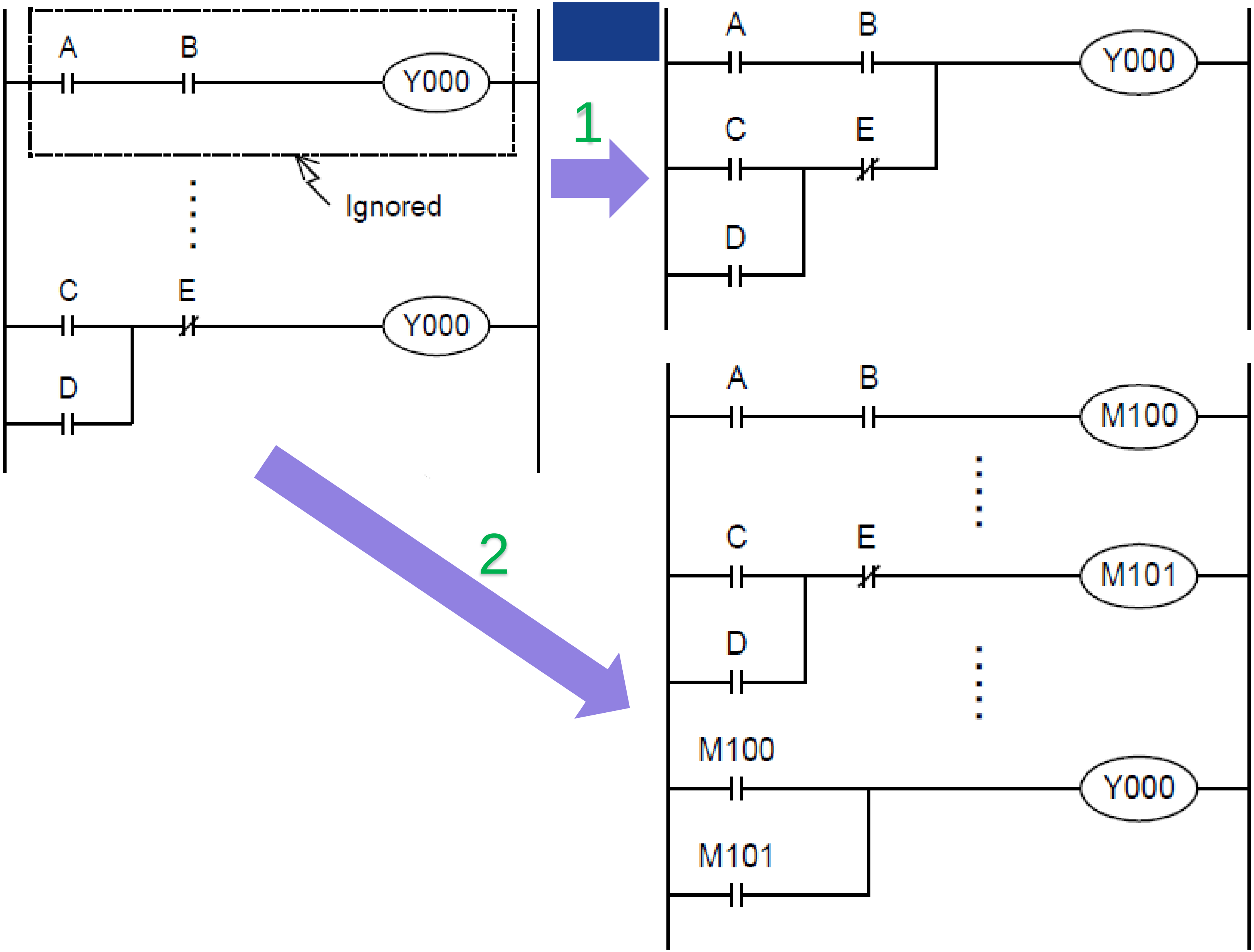
d) $\mathrm{Y0} = (\bar{A}B\bar{C}) + (D\bar{E}F)$

Mistake – Double output (double coil)

- **Double outputs**

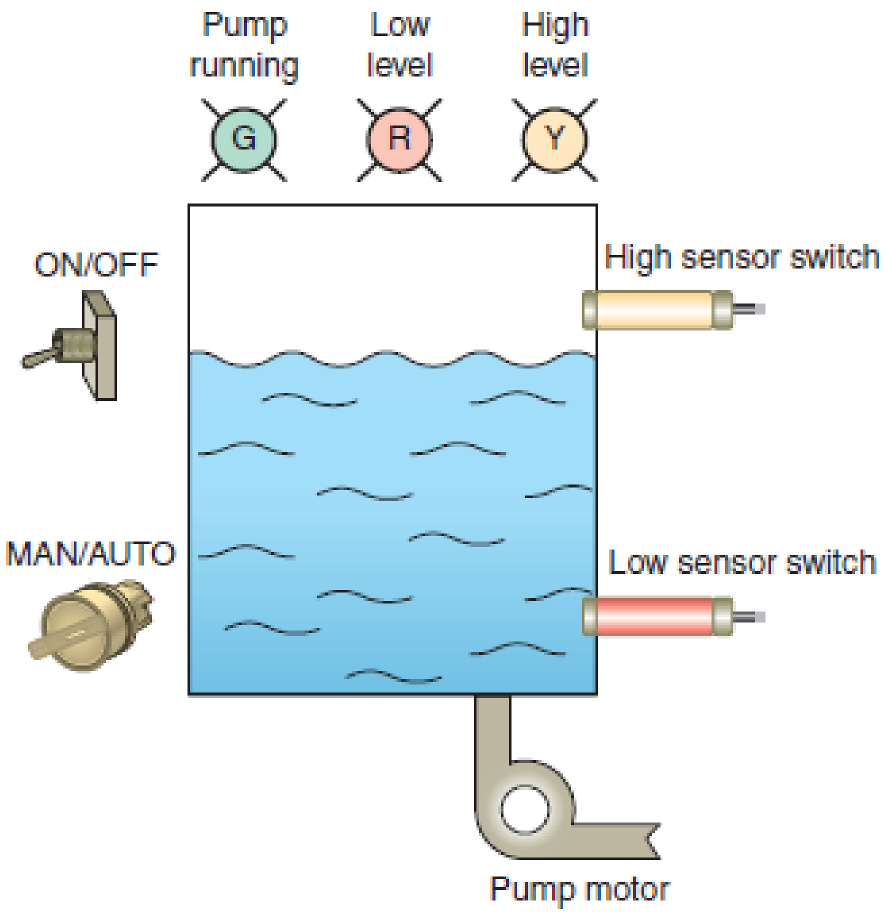


Solutions - Double output (double coil)



Exercises

❖ Exercise 1: Control the level of water in a storage tank.



OFF Position —The water pump will *stop* if it is running and will *not* start if it is stopped.

Manual Mode —The pump will start if the water in the tank is at any level except low.

Automatic Mode —If the level of water in the tank *reaches a high point,* the water pump will *start* so that water can be removed from the tank, thus lowering
the level.

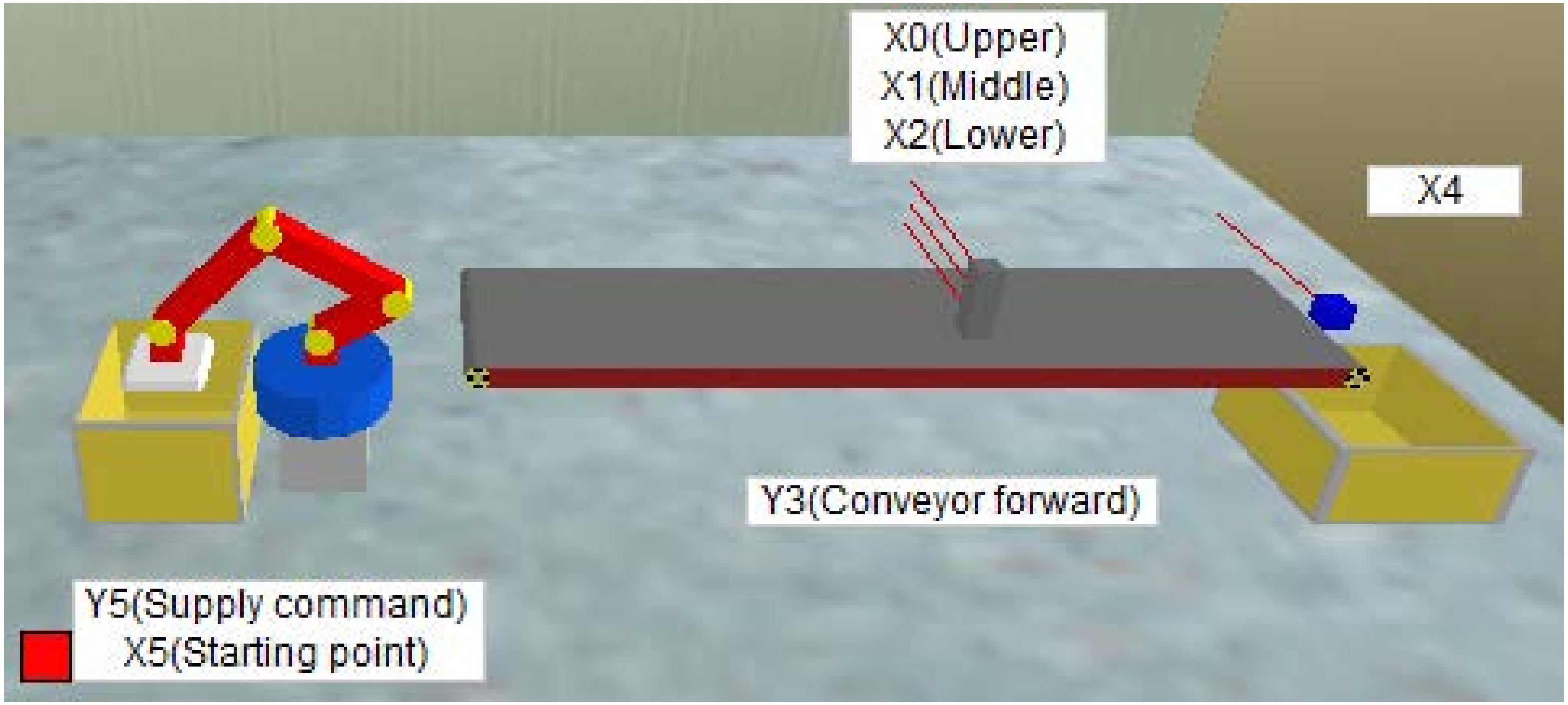
- When the water level *reaches a low point,* the pump will *stop* .

Status Indicating Lights —Water pump running light (green)

- Low water level status light (red).
- High water level status light (yellow)

Exercises

Exercise 2: Part sorting by size – FX Training D4

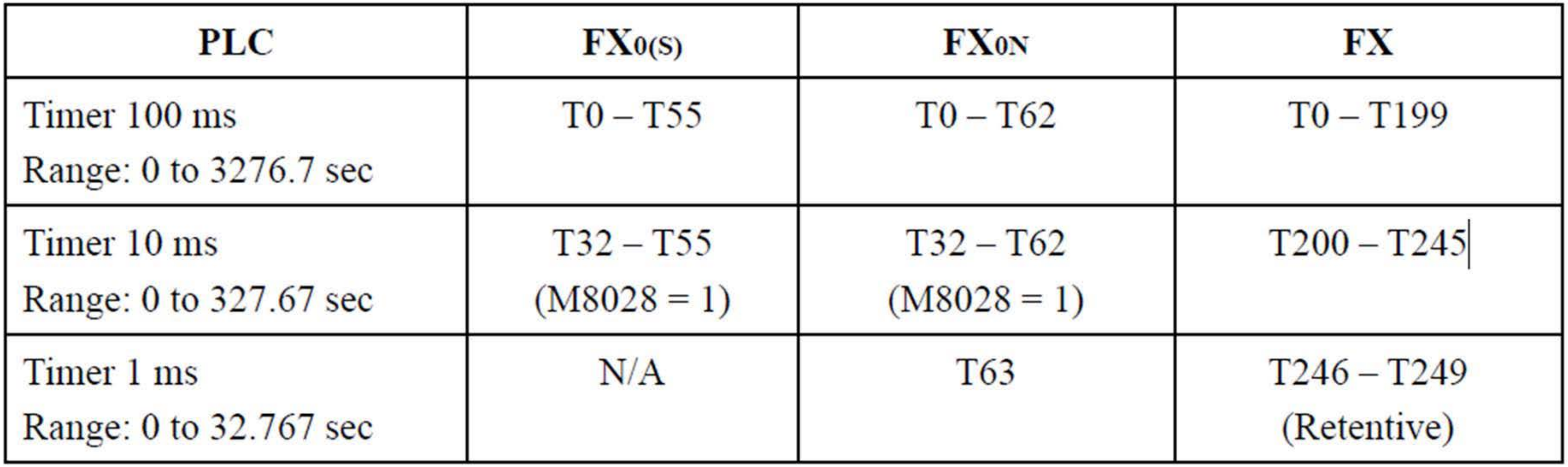


Read the control specifications in FX Training, D4 (Part Sorting by Size I)

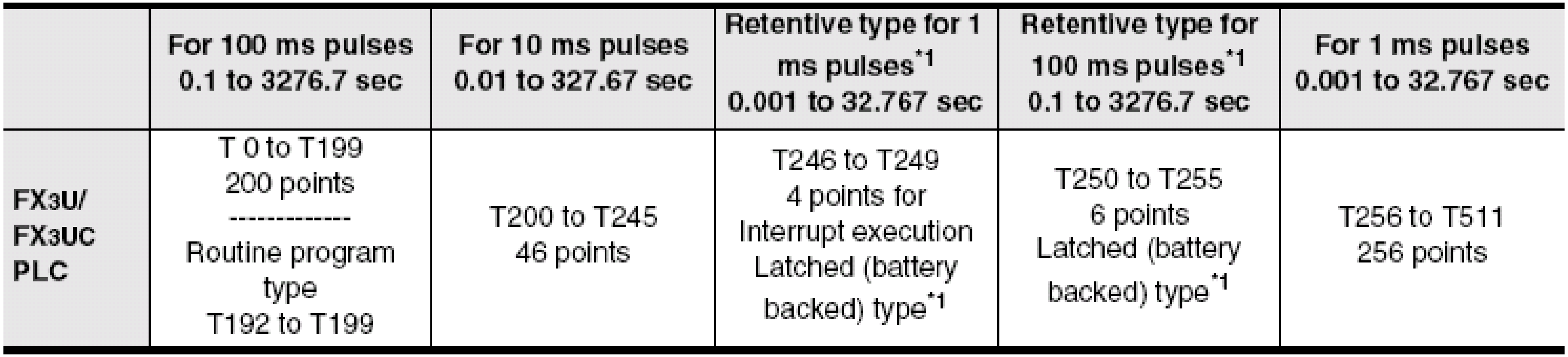
Timer – PLC Mitsubishi (FX Family)

- **Mnemonic of Timer is Ti**
- **Calculate the time delay with the following equation:**
  - ***Time delay = n × clock pulse***
  - ***With n is constant in the program memory (Kn)***
- **This is ON – Delay Timer.**

Timer – PLC Mitsubishi (FX Family)



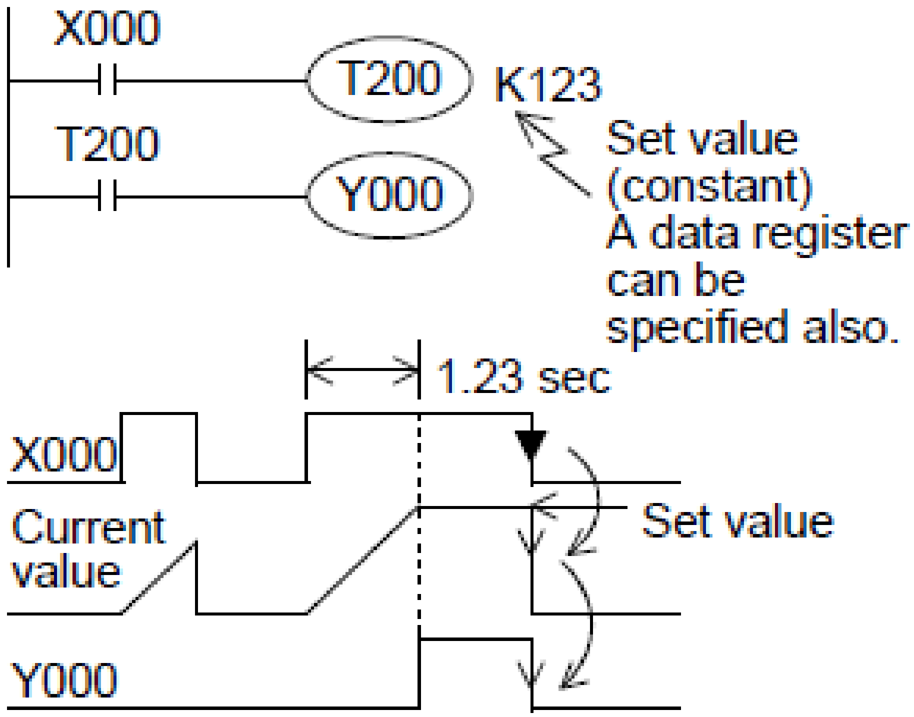
PLC	FX0(S)	FX0N	FX
Timer 100 ms Range: 0 to 3276.7 sec	T0 – T55	T0 – T62	T0 – T199
Timer 10 ms Range: 0 to 327.67 sec	T32 – T55 (M8028 = 1)	T32 – T62 (M8028 = 1)	T200 – T245
Timer 1 ms Range: 0 to 32.767 sec	N/A	T63	T246 – T249 (Retentive)



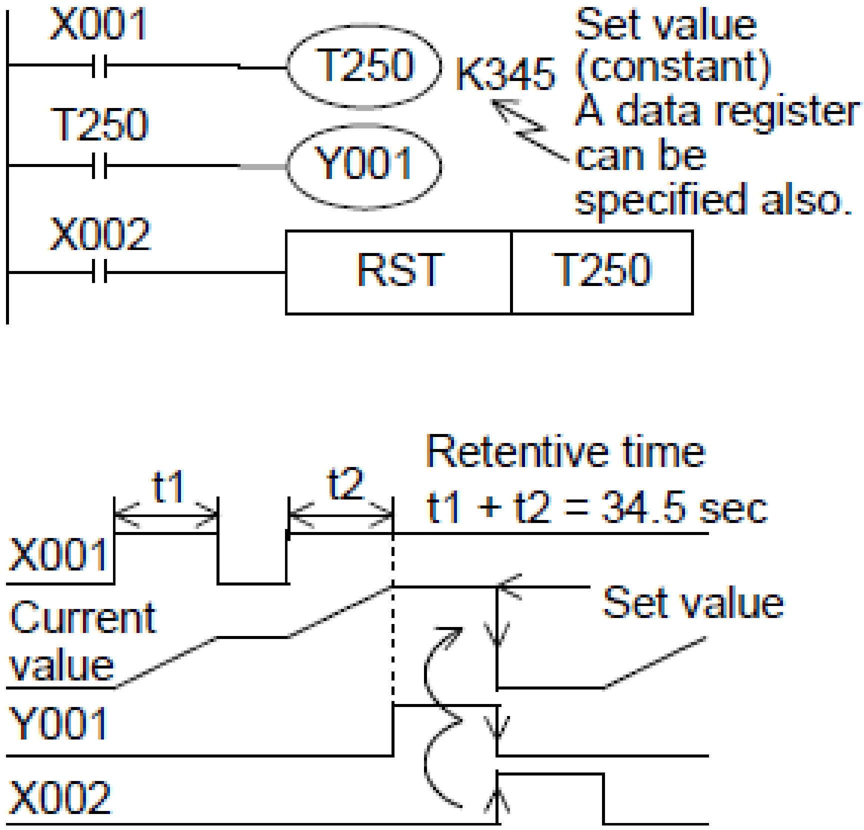
	For 100 ms pulses 0.1 to 3276.7 sec	For 10 ms pulses 0.01 to 327.67 sec	Retentive type for 1 ms pulses[*1] 0.001 to 32.767 sec	Retentive type for 100 ms pulses[*1] 0.1 to 3276.7 sec	For 1 ms pulses 0.001 to 32.767 sec
FX3U/ FX3UC PLC	T 0 to T199 200 points ------------- Routine program type T192 to T199	T200 to T245 46 points	T246 to T249 4 points for Interrupt execution Latched (battery backed) type[*1]	T250 to T255 6 points Latched (battery backed) type[*1]	T256 to T511 256 points

Timer – PLC Mitsubishi (FX Family)

General type

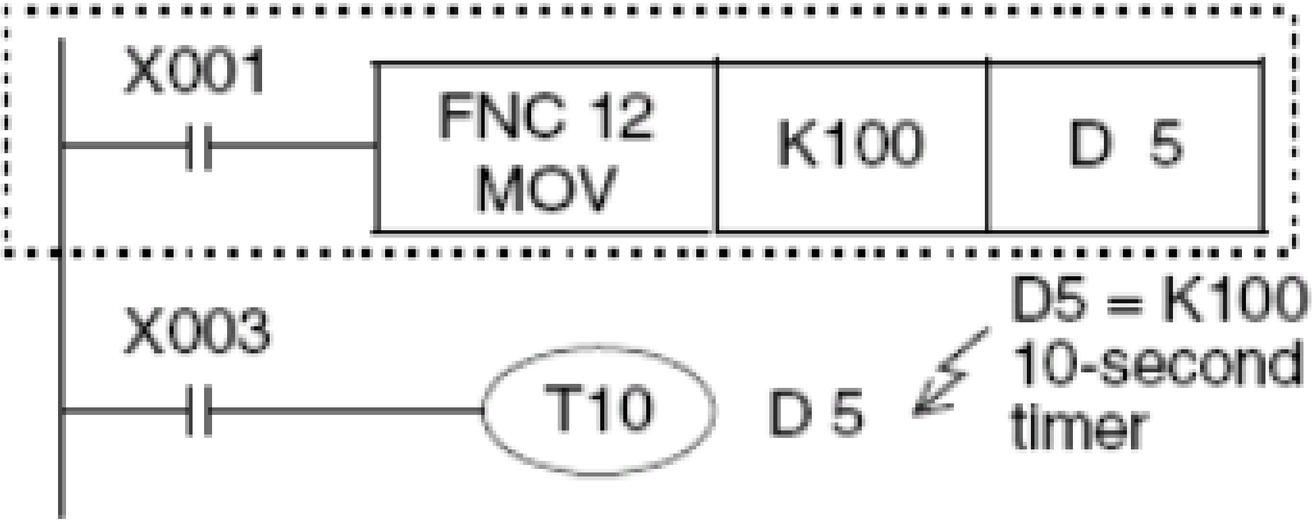


Retentive type



Timer – PLC Mitsubishi (FX Family)

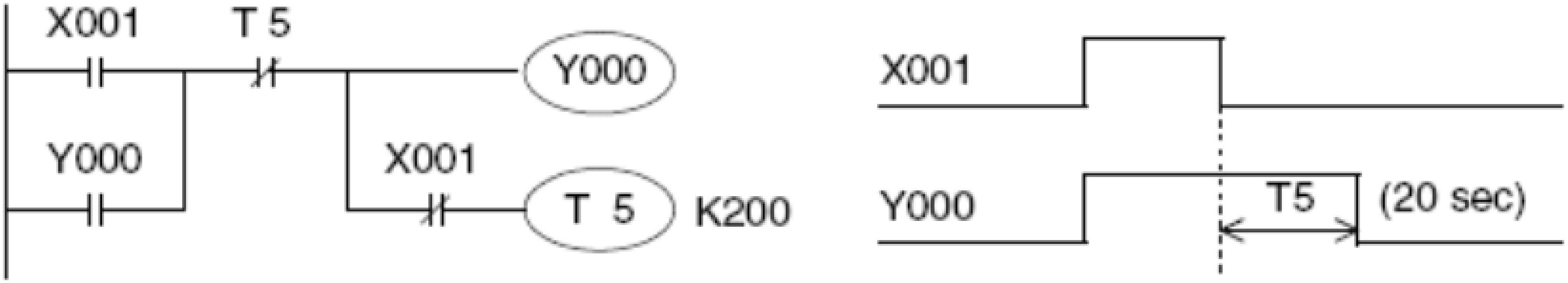
- Time delay is indirectly specified the contents of a data register D



OFF-Delay Timer

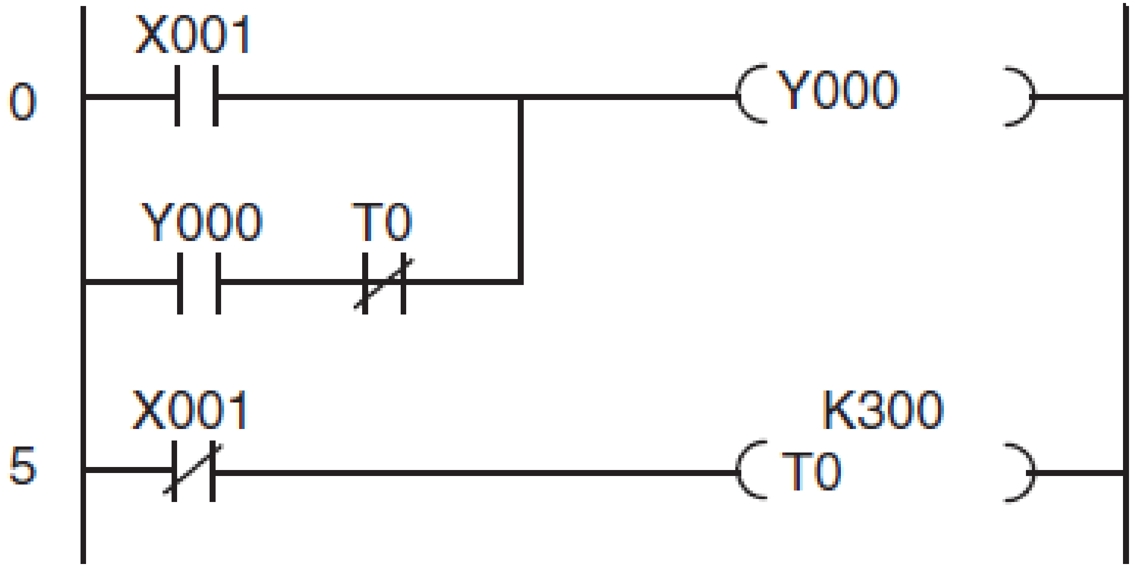
- Because Timer of FX Family PLC is ON-Delay, if we want to program a delayed break operation (an OFF-Delay Timer) → write a program as the following ladder.

Off-delay timer



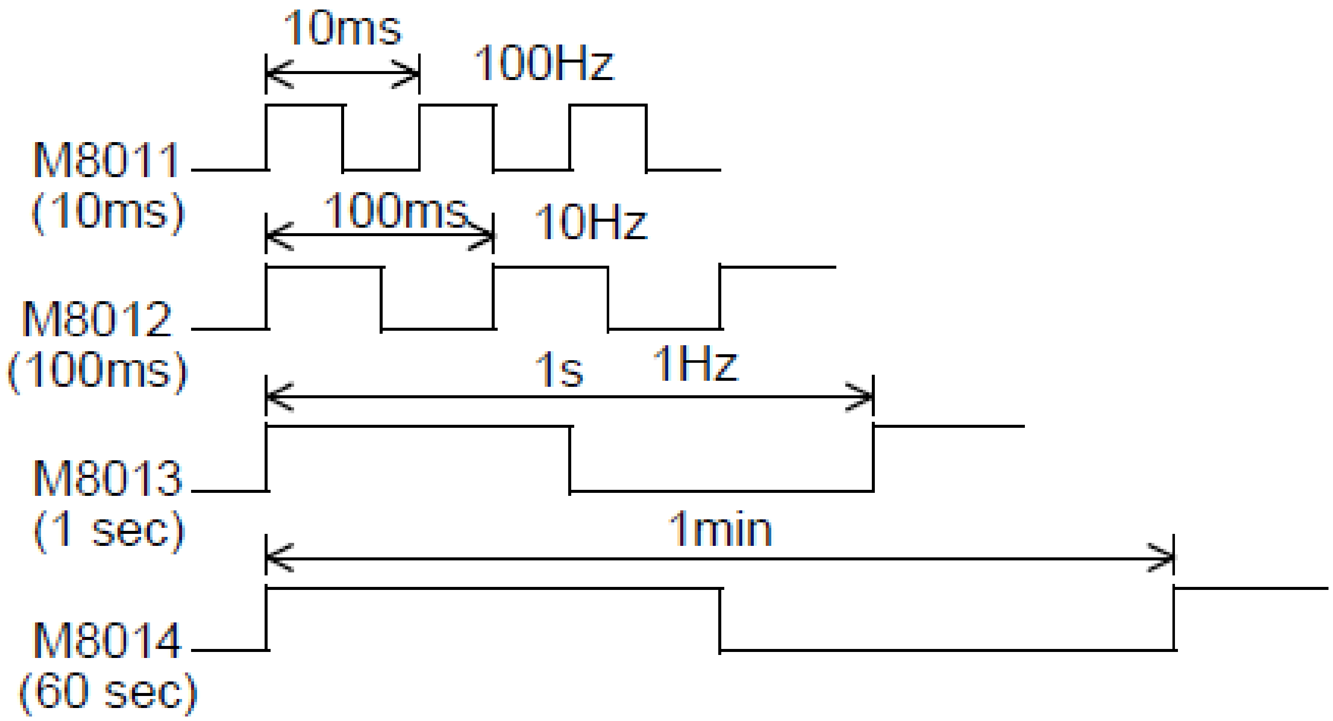
Quiz

- When a light switch (X1) is ON, a fan (Y0) is also ON. However, the latching function ensures that the fan also remains ON after the light has been switched OFF.
- Draw a diagram of signal sequence.
- Rewrite the program using SET/RST instructions.



Clock Signal Generators

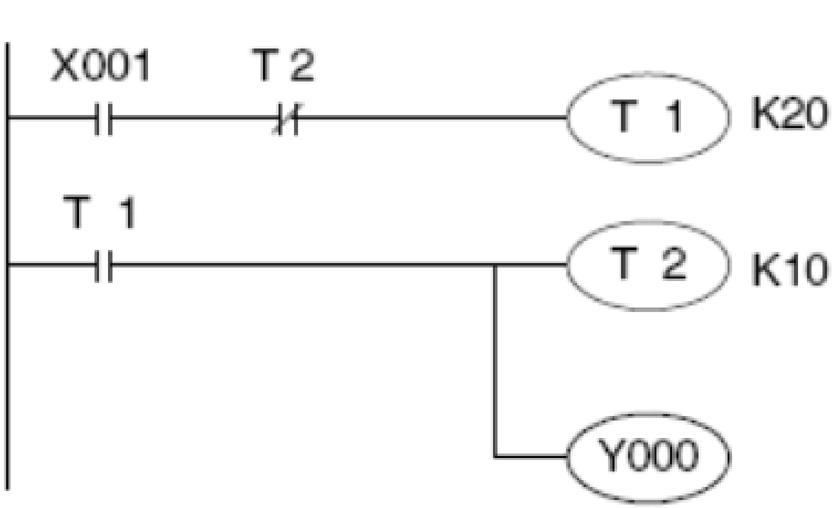
- The PLC has the 4 types of internal clock **M8011, M8012, M8013, M8014** with **10ms**, **100ms**, **1s**, **1min** clock pulse, respectively.

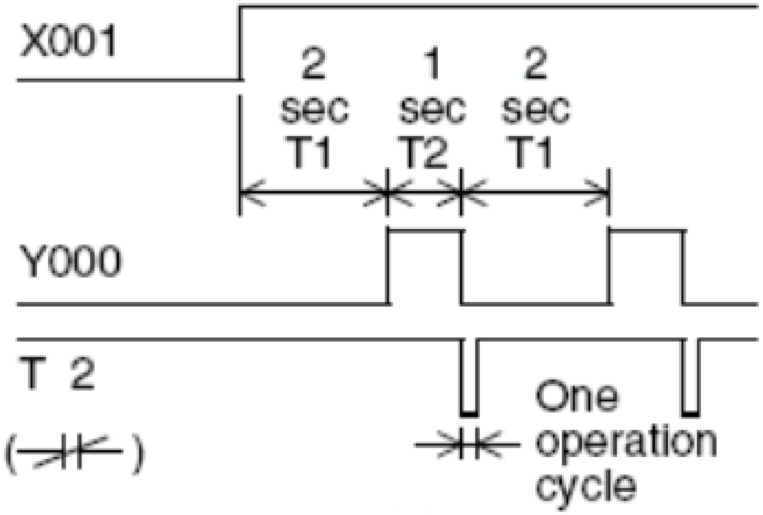


Caution: Even while the PLC is in the **STOP** mode, these clocks are always oscillating.

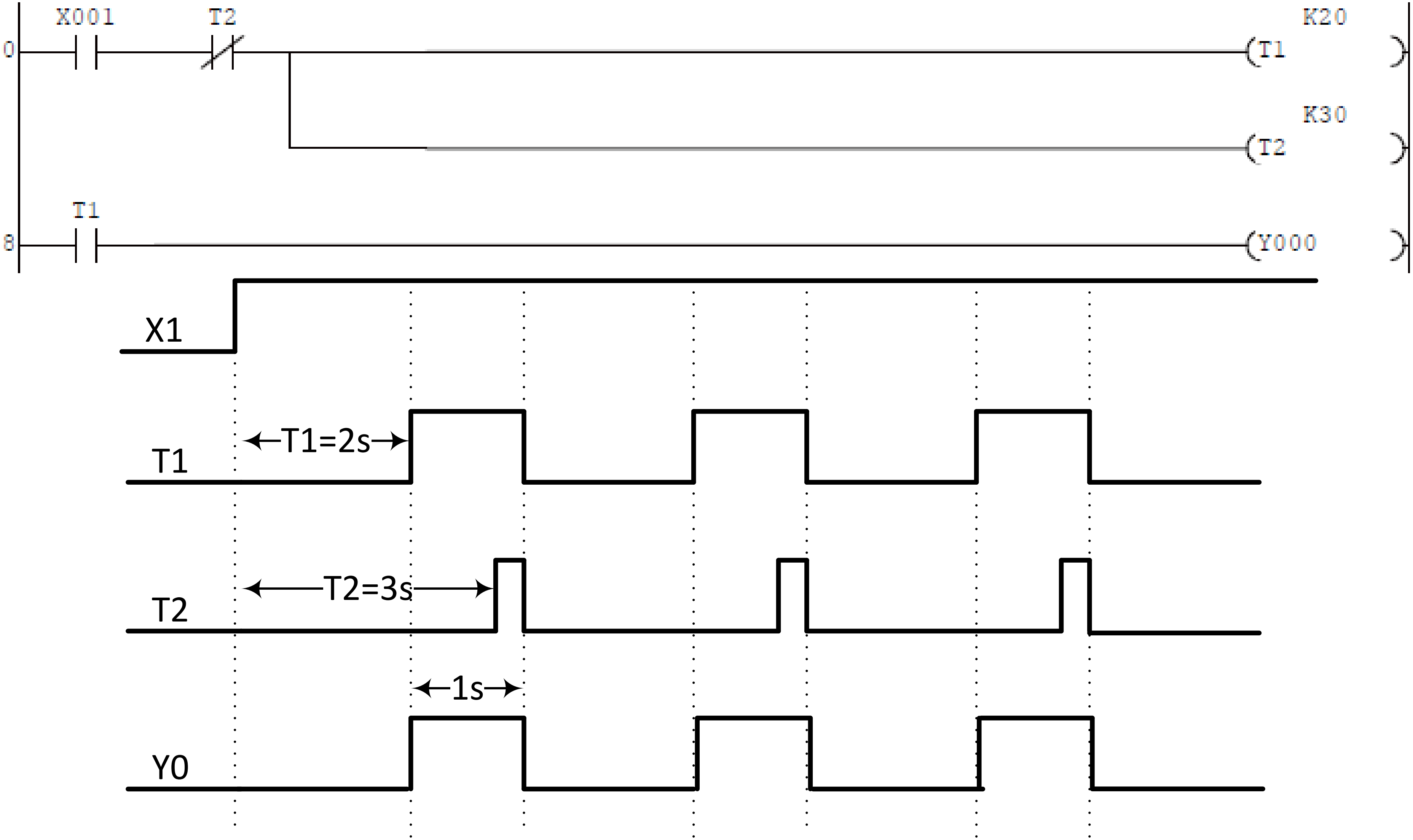
Clock Signal Generators (ON- and OFF-Delay)

- This program uses 2 timers to make pulse waves (square waves).
- The pulse width is highly variable.
- This program can apply to high speed output Y, internal memory M.





Clock Signal Generators



Exercises

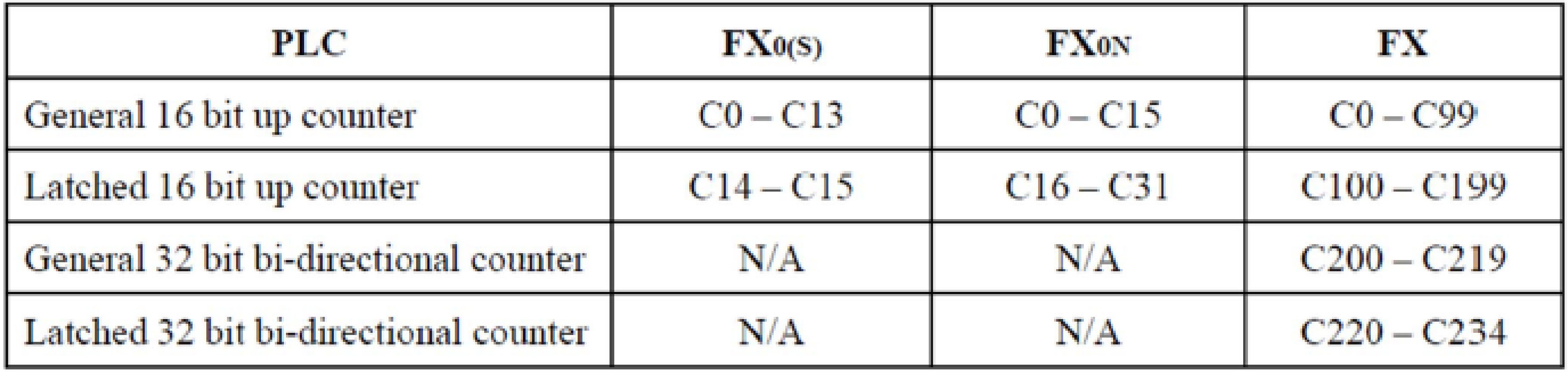
Exercise 3

○ ○ ○

Light 1: Y001 Light 2: Y002 Light 3: Y003

- Start: X000, Stop: X001
- Write a program to control 3 lights with the sequential requirement:
- Start → Light 1 ON in 1s → Light 2 ON in 1s → Light 3 ON in 1s → Light 1 và 3 ON in 2s → Light 2 ON 2s → Return to Light 1 ON in 1s.
- Stop → 3 lights OFF.

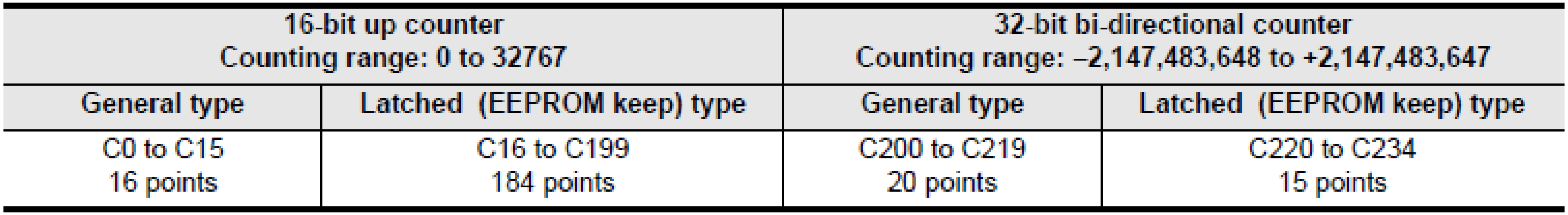
Counter – PLC Mitsubishi (FX Family)



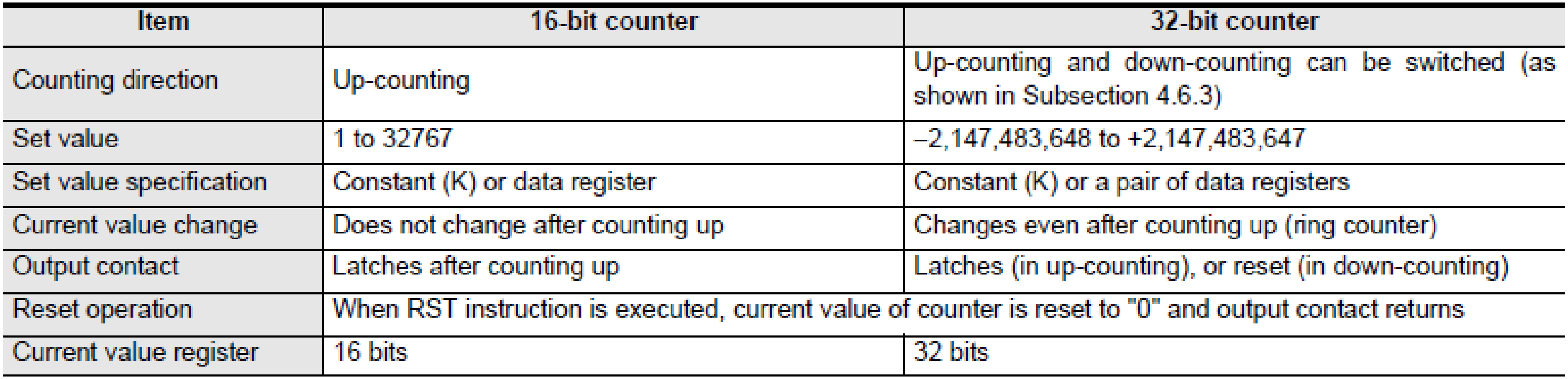
PLC	FX0(S)	FX0N	FX
General 16 bit up counter	C0 – C13	C0 – C15	C0 – C99
Latched 16 bit up counter	C14 – C15	C16 – C31	C100 – C199
General 32 bit bi-directional counter	N/A	N/A	C200 – C219
Latched 32 bit bi-directional counter	N/A	N/A	C220 – C234

Counter – PLC Mitsubishi (FX Family)

❖ Numbers and features of counters (FX3G)



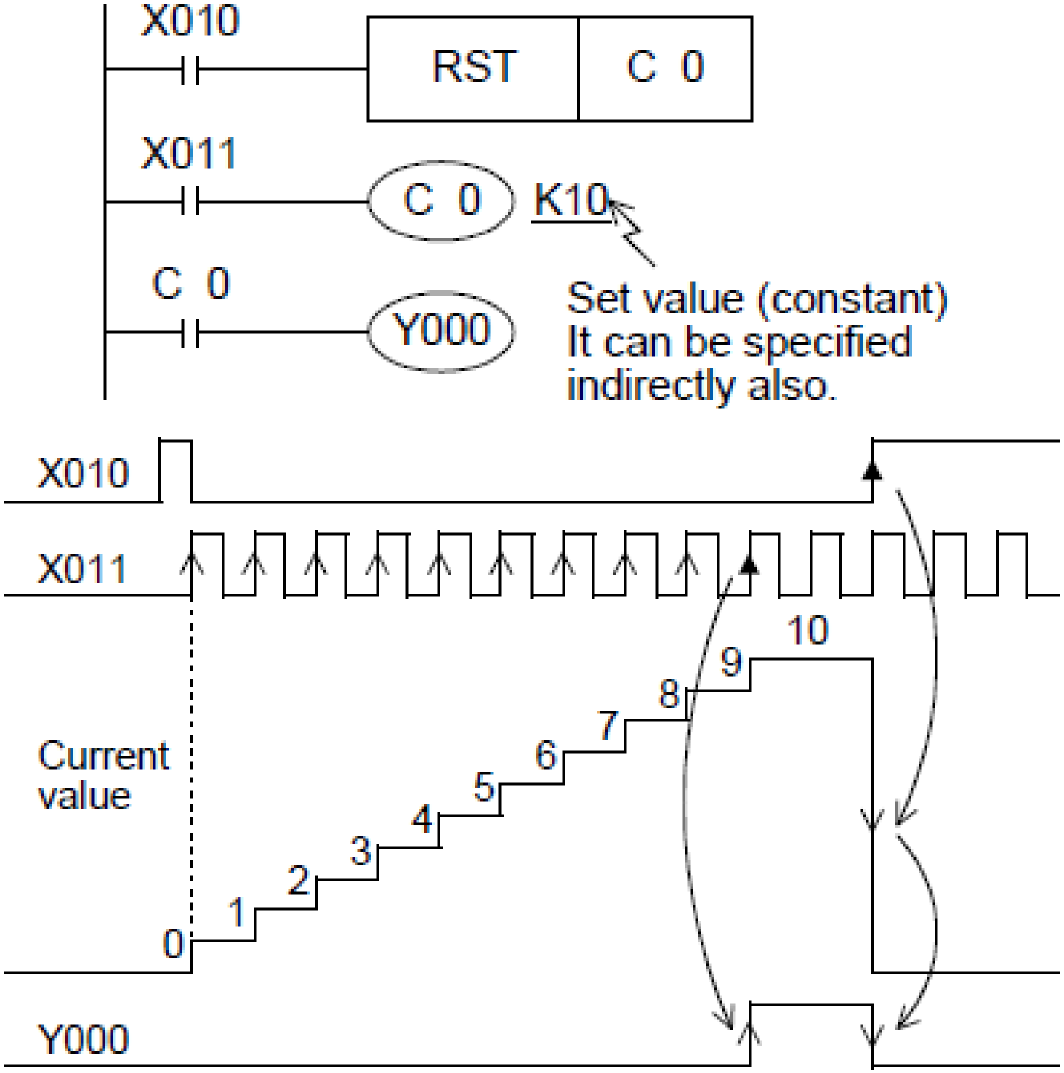
16-bit up counter Counting range: 0 to 32767		32-bit bi-directional counter Counting range: –2,147,483,648 to +2,147,483,647	
General type	Latched (EEPROM keep) type	General type	Latched (EEPROM keep) type
C0 to C15 16 points	C16 to C199 184 points	C200 to C219 20 points	C220 to C234 15 points



Item	16-bit counter	32-bit counter
Counting direction	Up-counting	Up-counting and down-counting can be switched (as shown in Subsection 4.6.3)
Set value	1 to 32767	–2,147,483,648 to +2,147,483,647
Set value specification	Constant (K) or data register	Constant (K) or a pair of data registers
Current value change	Does not change after counting up	Changes even after counting up (ring counter)
Output contact	Latches after counting up	Latches (in up-counting), or reset (in down-counting)
Reset operation	When RST instruction is executed, current value of counter is reset to "0" and output contact returns	
Current value register	16 bits	32 bits

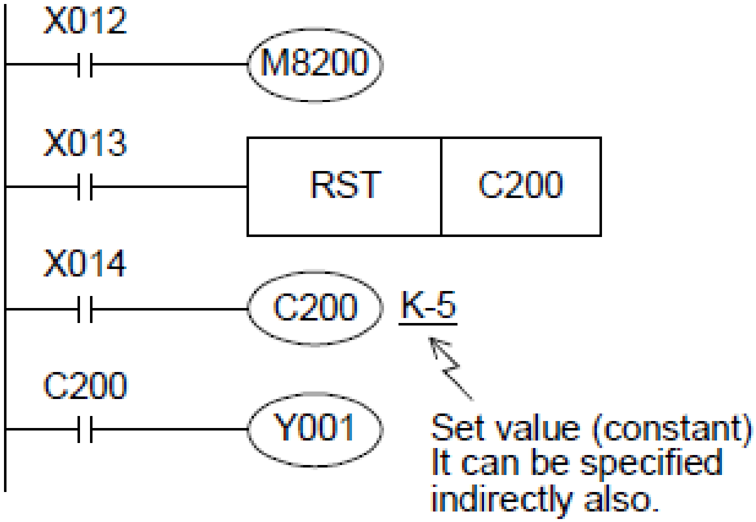
Counter – PLC Mitsubishi (FX Family)

- **General type and latched type 16-bit up counters**

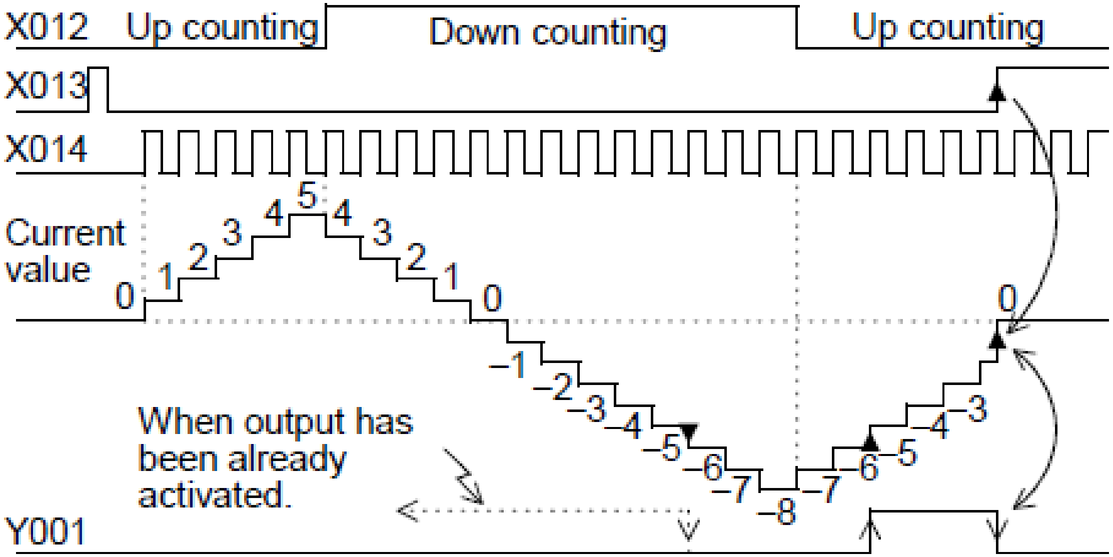


Counter – PLC Mitsubishi (FX Family)

- The counting direction (up or down) is specified by special auxiliary relays from *M8200 to M8234*.
- When M82xx is ON, counter C2xx executes down-counting. When M82xx is OFF, counter C2xx executes up-counting.
- The set value (positive or negative) can be specified by a constant (K) or contents of data register D.



Set value (constant)
It can be specified
indirectly also.



When output has
been already
activated.

Quiz

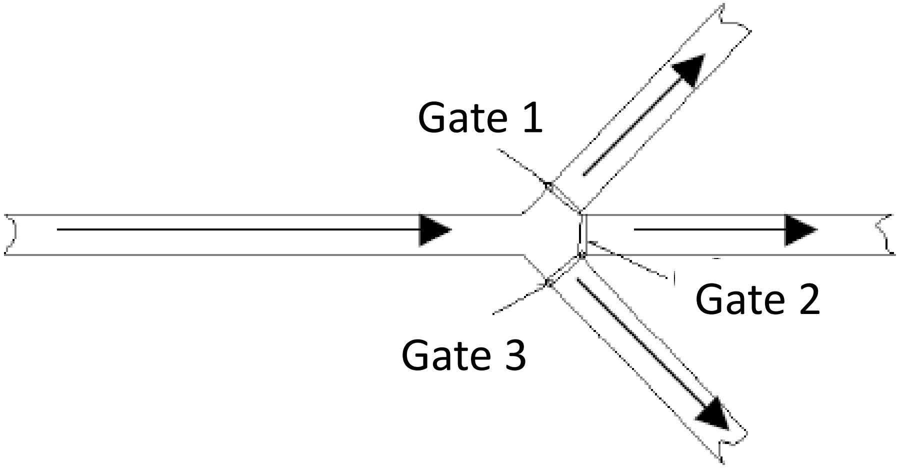
- Write the program to generate 300 pulses with ON duration is 0.1s and OFF duration is 0.2s.

Exercises

Exercise 4:

- A herd of 300 cattle, divides into 3 different cages, each cage of 100 cattle.
- Cattle will follow a common line that will separate 100 cattle for each cage.
- Press Start → Open gate 1 for the first 100 cattle → close gate 1 and open gate 2 for the next 100 cattle → Close gate 2, open gate 3 for the final 100 cattle → Close gate 3.

- Write a control program using PLC Mitsubishi



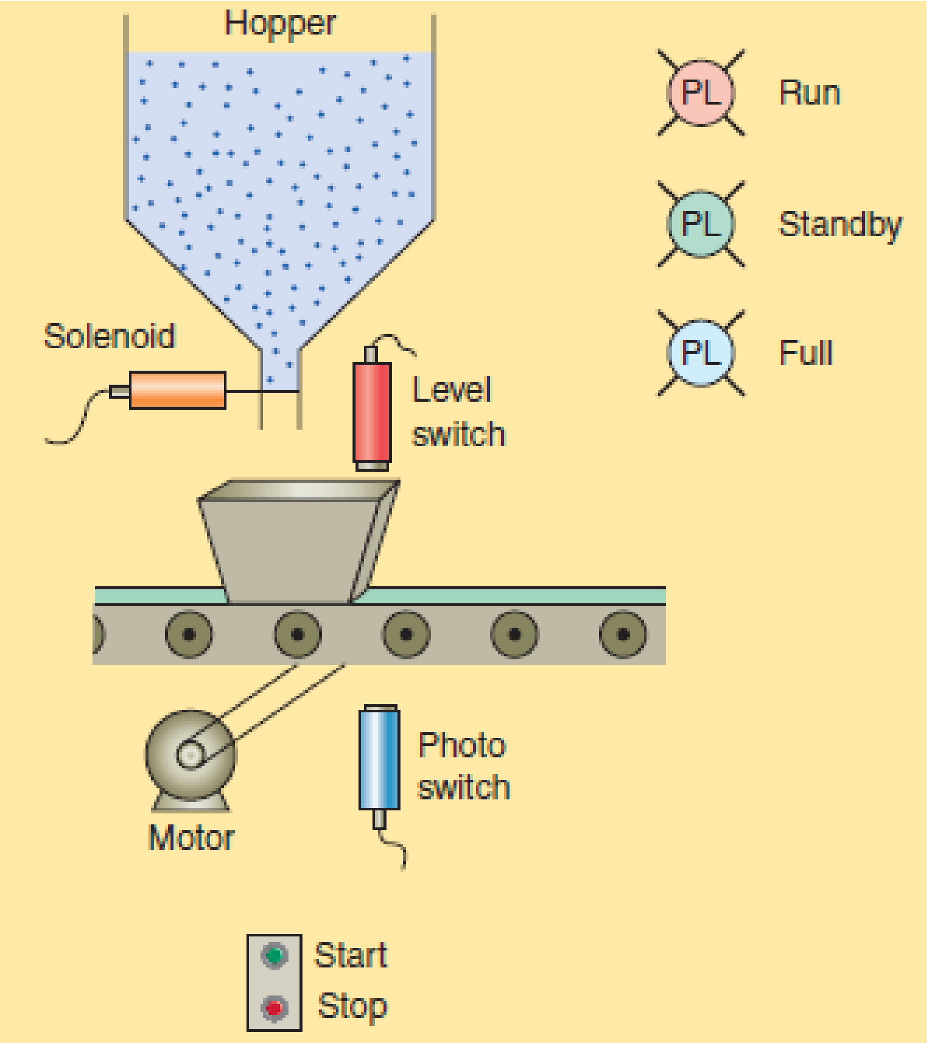
Exercises

- **Exercise 5: Write a program to control traffic lights**

  Green: 15s, Yellow: 5s, Red: 20s.

Exercises

- **Exercise 6: A continuous filling operation. This process requires that boxes moving on a conveyor be automatically positioned and filled.**

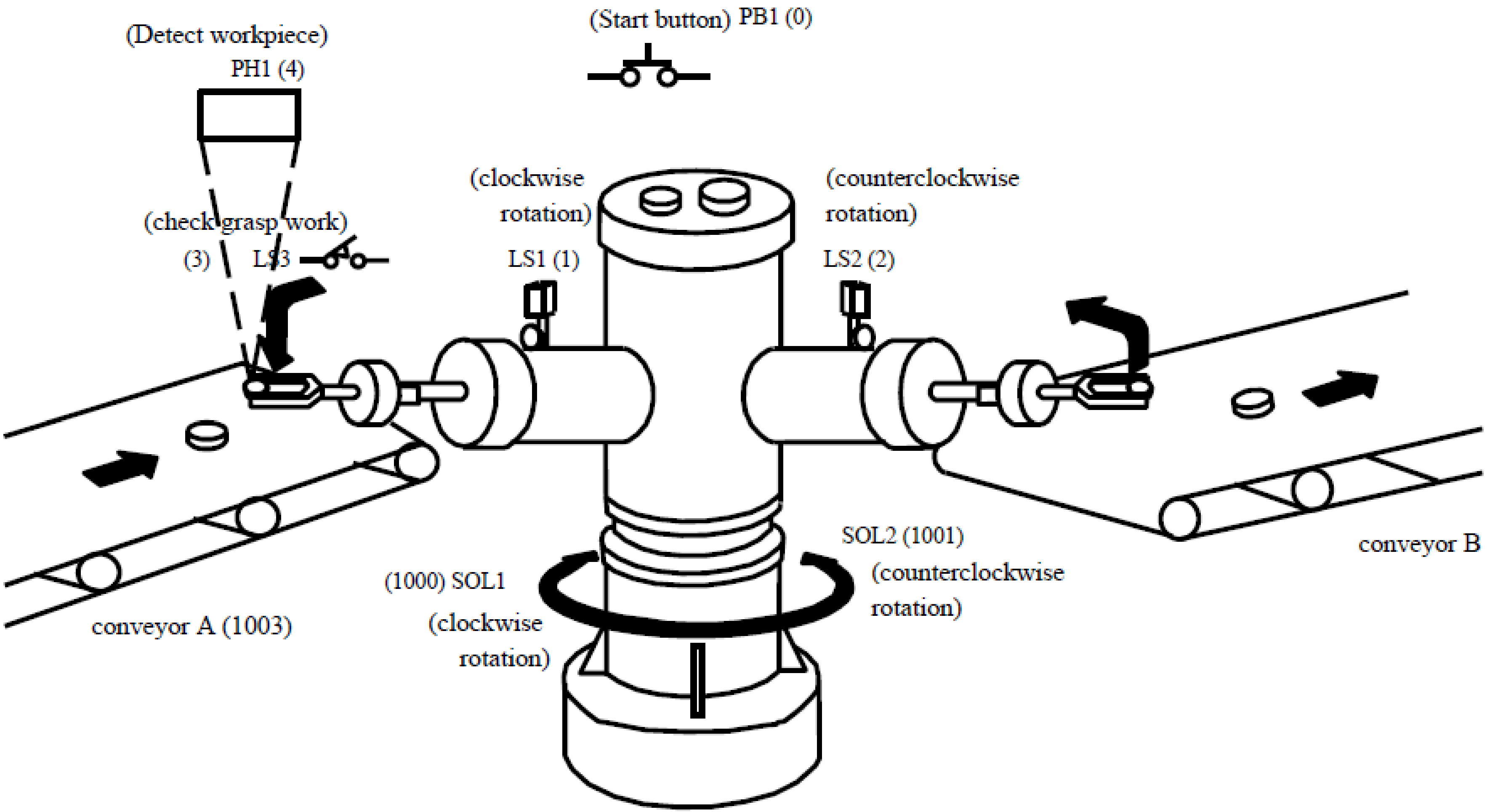


Exercises

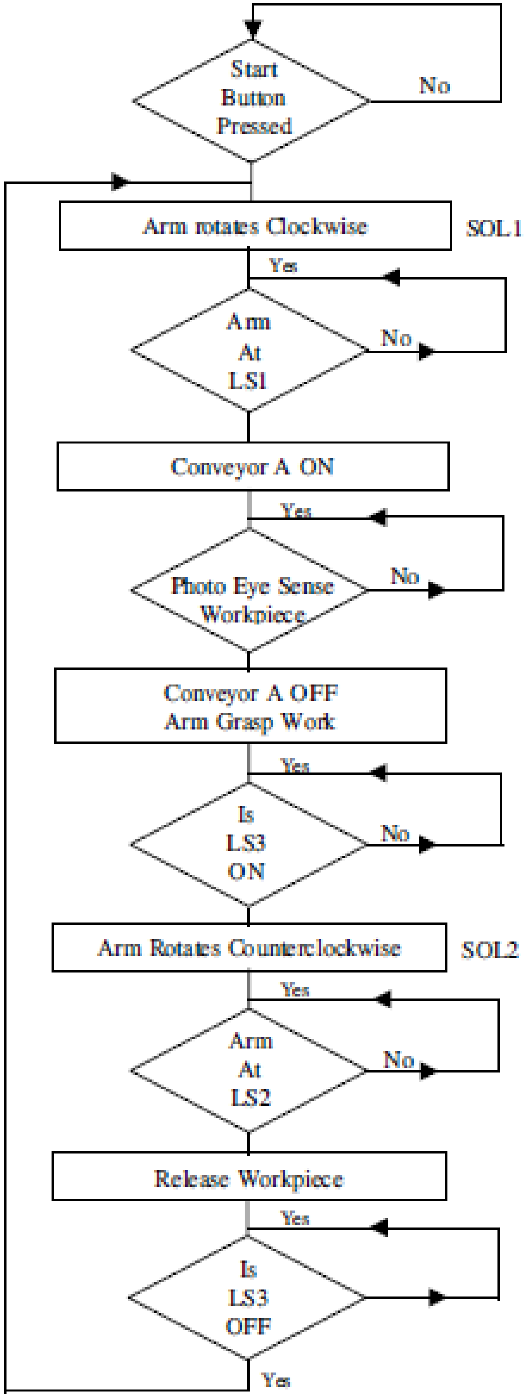
- Start the conveyor when the start button is momentarily pressed.
- Stop the conveyor when the stop button is momentarily pressed.
- Energize the run status light when the process is operating.
- Energize the standby status light when the process is stopped.
- Stop the conveyor when the right edge of the box is first sensed by the photoelectric sensor.
- With the box in position and the conveyor stopped, open the solenoid valve and allow the box to fill. Filling should stop when the level sensor goes true.
- Energize the full light when the box is full. The full light should remain energized until the box is moved clear of the photoelectric sensor.

Exercises

Exercise 7: Control the robot’s movements



Exercises

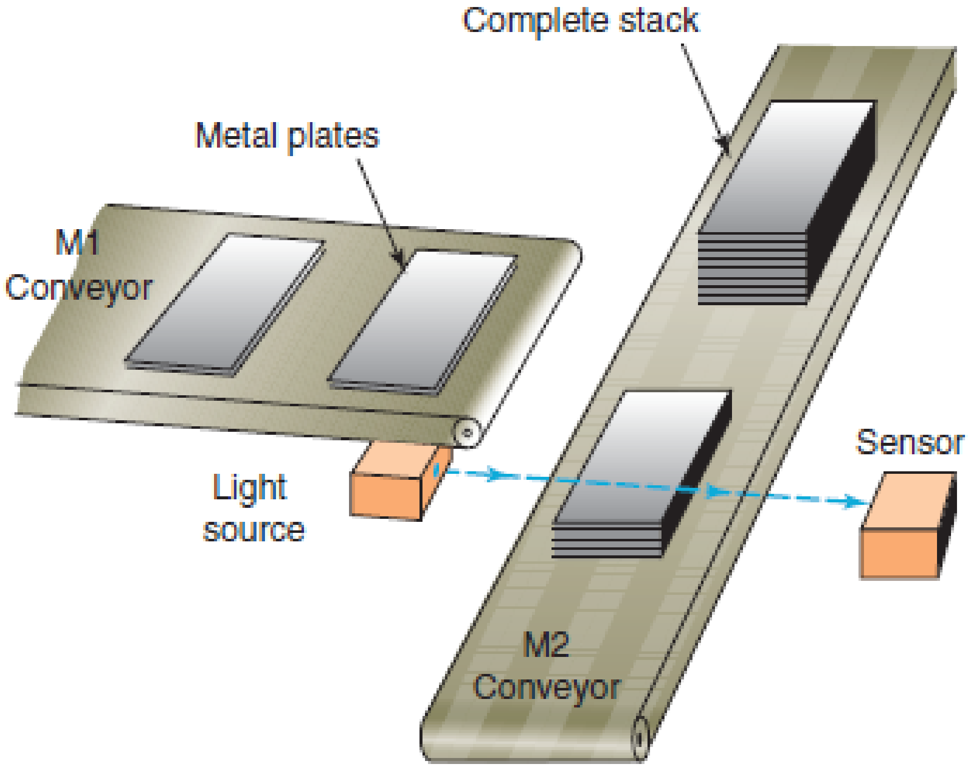

Start Button Pressed
No
Arm rotates Clockwise
SOL1
Yes
Arm At LS1
No
Conveyor A ON
Yes
Photo Eye Sense Workpiece
No
Conveyor A OFF Arm Grasp Work
Yes
Is LS3 ON
No
Arm Rotates Counterclockwise
SOL2
Yes
Arm At LS2
No
Release Workpiece
Yes
Is LS3 OFF
Yes

Exercises

Devices	I/O PLC	Description
Start		Start button
Stop		Stop button
LS1		Clockwise limit switch
LS2		Counterclockwise limit switch
LS3		Check grasp work limit switch
PH1		Detect workpiece sensor
Sol1		Rotate the arm clockwise
Sol2		Rotate the arm counterclockwise
Sol3		Grasp the work
Conveyor A		Operate conveyor A
Conveyor B		Operate conveyor B

Exercises

- **Exercise 8: Write a control for automatic stacking system.**



The operation of the program can be summarized as follows:

• When the start button is pressed, conveyor M1 begins running.

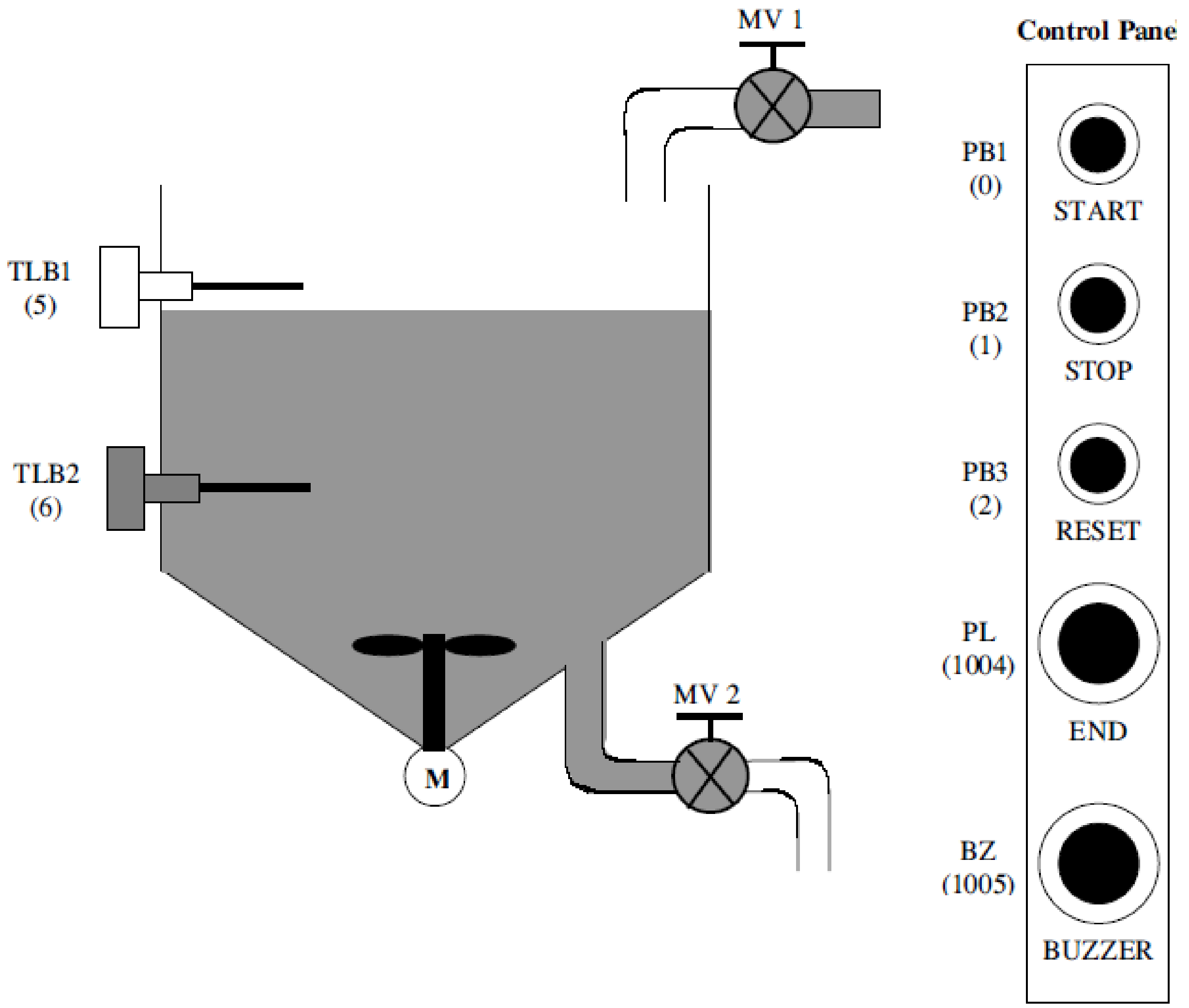
• After 15 plates have been stacked, conveyor M1 stops and conveyor M2 begins running.

• After conveyor M2 has been operated for 5s, it stops and the sequence is repeated automatically.

• The done bit of the timer resets the timer and the counter and provides a momentary pulse to automatically restart conveyor M1.

Exercises

Exercise 9: Filling/Draining control operation.



Exercises

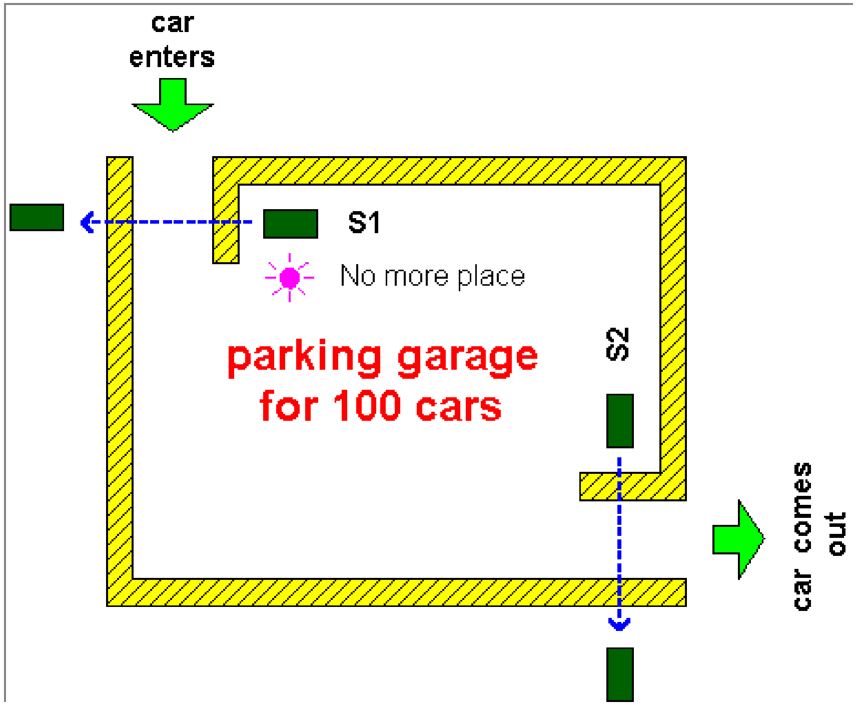
- When the PB1 is pressed, MV1 opens and the water begins to fill the tank. At the same time, the stirring motor M starts.
- When the water level passes TLB2 and reaches TLB1, MV1 closes and the stirring motor stops.
- Next, MV2 opens and starts draining the water. When the water level drops below TLB2, MV2 closes.
- When the cycle of operation has repeated four times, the operation END indicator illuminates, and the filling and draining operation will not restart even if PB1 is pressed.
- Only after pressing the PB3, the filling and draining operation can run when PB1 is pressed.

Exercises

Input		Output	
	Start button (PB1)		Water supply valve (MV1)
	Stop button (PB2)		Drain valve (MV2)
	Reset button (PB3)		Stirring motor (M)
	Upper level switch (TLB1)		End indicator
	Lower level switch (TLB2)		Buzzer

Exercises

Exercise 10: Car parking garage



- Its capacity is 100 cars.
- S1 is a sensor at gate in.
- S2 is a sensor at gate out
- A warning light is used to indicate the parking garage full.

Write a control program.

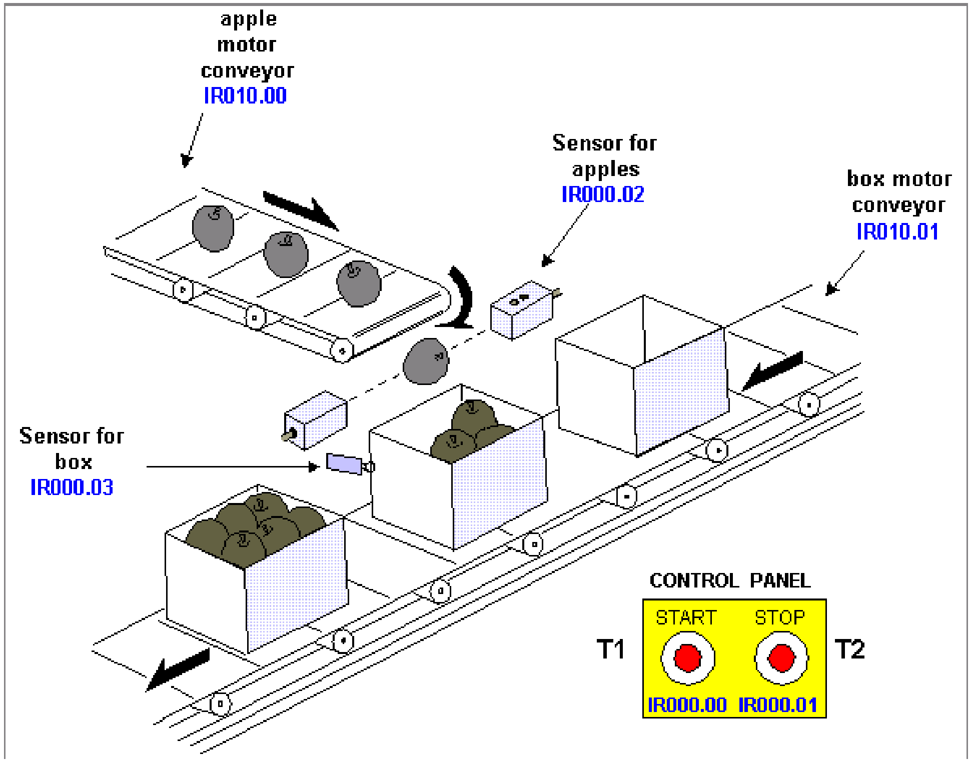
Start → Count the car coming in/out of parking lot → If the parking lot is full (>=100), turn on the warning light.
Stop → Reset system.

Exercises

Input		Output	
	Start		The warning light
	Stop		
	Sensor at gate in		
	Sensor at gate out		

Exercises

Exercise 11: Apple packaging line control



IR000.00: Start button

IR000.01: Stop button

IR000.03: sensor 1, detection of box present

IR000.02: sensor 2, apple counting.

IR010.01: Conveyor 1, the box conveyor

IR010.00: Conveyor 2, the apple conveyor

Exercises

Sensor 1

When Start button is pressed → Conveyor 1 moves → Conveyor

Sensor 2

1 stops, Conveyor 2 moves (supply apple) → Count 10 apples →

Conveyor 2 stops, Conveyor 1 moves again.

Stop button is pressed → The line stops immediately.

Exercises

Input		Output	
	Start		Conveyor 1
	Stop		Conveyor 2
	Apple counting		
	Box detection		

Exercises

Exercise 12: Temperature control in green house

- Temperature sensor: 0 ÷ 100^{0}C, -10 ÷ 10V.
- Connect to channel 1 of module FX2N-4AD 12 bit, slot 0.
- Write a program:
  - Keep the temperature of green house in range 25÷28^{0}C. If t^{0} is less than 25^{0}C → turn on heater; greater than 28^{0}C → turn off heater.
  - Set the conversion speed = 6ms.
  - Show the temperature value.

Exercises

Exercise 13: Temperature and moisture control in green house

- **Temperature = 25÷28^{0}C, moisture = 70 ÷ 75%.**
- **Questions:**
  - **Choose the sensors.**
  - **Draw a flowchart of control system.**
  - **Write a control program (2 sensors connect to channel 1 and 2 of module FX-4AD 12bit ADC, slot 0).**

Exercises

Exercise 14: Control the speed of DC motor

- Have a DC motor with the operating voltage 0-10V which connects to channel 1 of module FX2N-4DA 12bit, slot 1.
- Control the speed of motor in 3 levels: 100%, 70%, 40% with 3 buttons.
- Questions:
  - Determine the Hxxxx value
  - Write a TO instruction for configuration for module FX2N-4DA.
  - Find the relationship equation between the speed of motor and digital value.
  - Write a complete program.

Exercises

- A temperature sensor has 0-100^{0}C, 4-20 mA, connected to CH1 and 1 warning horn.
- Module FX2N-4AD with resolution 12bit at slot 0 is used.
- 2 LED 7 segment are wired to 8 discrete outputs to show the temperature value.
- Problems:
  - Find the linear equation of temperature and digital value.
  - Write a program as follows:
    - Set a configuration for module FX2N-4AD.
    - When START button is pressed, system will read, convert and indicate the temperature value on LED 7 segment.
    - If temperature is greater than 70^{0}C, turn on the horn with ON in 1s and OFF in 1.5s.
    - When STOP button is pressed, the system will stop immediately.