PROGRAMMABLE LOGIC CONTROLLERS
ASSIGNMENT 3
DESIGN AND OPERATIONAL CHARACTERISTICS
OUTCOME 3 PROGRAMMING TECHNIQUES

At this stage you should have completed all the exercise to do with programming the Mitsubushi/Melsec with Medoc software. These exercises should have taught you some of the programming techniques and advanced functions. You are now in a position to produce a programme for an industrial problem and this is PART 3 of this assignment.

PART 1   EXERCISE

- *Produce evidence of having completed the exercises in Tutorial 3 part 1 and 2.*

List of exercise
1.   Timers
2.   Counters
3.   Registers
4.   A/D and D/A
5.   SFC programme

PART 2   FURTHER STUDIES

On completion of part 3 you will have produced a ladder diagram and an instruction set with full documentation. In addition you need to study some of the advanced features of programming. Here are some extra things you need to find out.

- *Define the terms OPCODE and OPERAND*

Suppose you want to switch a motor Y430 on with a simple switch X400. In Mitsubushi Mnemonics the instruction set would be LD X400   OUT Y430. If you used the programme LD X400   Set Y430 the programme would behave differently.

- *Explain the difference between LoaD and Set and why when you use SET you also need RESET.*

Some programming methods enable you to write plain language statements such as

IF A>B AND B<C THEN SET D ELSE RESET D.

- *Find out about these more advanced forms of instructions and list some of the things that can be done that cannot easily be done with the Mitsubishi system.*

GRAFCET

- *Find out and write a brief but accurate description of the grafcet method of programming and discuss its advantages and disadvantages compared to the other methods.*

OFFLINE AND ONLINE PROGRAMMING

- *Define the meaning of "offline" and "online" programming and discuss the advantages and disadvantages.*

DEBUGGING

You have used debugging techniques with the Mitsubishi and Ladsim software.

- *Discuss methods of debugging your programmes and explain the usefulness of the methods.*

When you transfer your programme from The PC to the PLC or use the programming panel to transfer your programme on an EEPROM, you have the choice of comparing your programmes after transfer.

- *Explain what this means and why it is important.*

LOGIC FUNCTIONS

Explain the following logic functions and draw the European and American symbols for each.

- *And*
- *Or*
- *Exclusive Or*
- *Nand*
- *Nor*

FLOW CHARTS AND TRUTH TABLES AND LOGIC FUNCTION DIAGRAMS

You will be asked to produce a flow chart, truth table and logic function diagram in part 2.

- *Briefly explain the advantages and disadvantages of each method. You might give examples where one method might be more useful than another.*

ADVANCED FEATURES

Study the Mitsubishi manual and describe the following advanced features that are available and what they might be used for. You have already used some of them in your worksheets. You might give some industrial examples (look in the manual).

Register operations

- *Add*
- *Subtract*
- *Decrement*
- *Increment*
- *Right shift*
- *Left shift*
- *Rotate*
- *Compare*
- *Fast counters*
- *Fast timers*

PART 3   INDUSTRIAL PROGRAMMES

The following is written for systems with Mitsubishi PLCs and Medoc software programming but can be adapted to other systems. Three industrial examples are described. You should select ONE that best suits what you have available and complete the programming required to operate it as described in each. If you have no hardware, you might be able to complete one of them with suitable simulation software such as Ladsim and Pneusim.

EXAMPLE 1 – LUBRICATION PUMP

This example is intended for those with a range of small equipment including 6 or 12 V motors, small lamps such as LEDs and a range of switches. Using this you should be able to produce a reasonable model and connect it to a suitable PLC.

The following gives a brief description of a proposed machine system.

The machine will consist of a main drive motor and a lubricant pump system. A single action start button will operate the lubricant pump and an indicator lamp is to be switched on when the pump is operating. The oil reservoir is to be fitted with a mechanical float switch and a thermostatic switch and either will switch the pump off any time the lubricant reservoir falls below a critical level or the oil becomes too hot.

The float switch is a normally open type that closes when the level is high and opens when the level is low.

The thermostat is a normally open switch that closes when the temperature is high and opens when it is low.

There are two start buttons for the main motor located at two different points. Either button may start the motor but not if they are both on at the same time. Both are normally open toggle switches.

The motor must not start unless the pump is running and the guard door is closed. The guard sensor is normally open and closes when the guard is down. The sensor that detects when the pump is running is normally open and closes when the pump is running.

The following tasks are to be completed and a report of the activities compiled together with any required background information. In addition the practical aspects of tasks 4, to 7 will be observed and assessed by your tutor. Instruction manuals will be available for carrying out tasks that are new to you.

What you need to do:
1. *Produce a list of tags and labels for all the inputs and outputs.*
2. *Produce a flow chart style program.*
3. *Produce a truth table showing only the conditions that turn the motors on.*
4. *Produce a logic functional block diagram using European logic symbols*
5. *Produce a circuit diagram for the hardwired control circuit.*
6. *Produce a ladder logic diagram for the control circuit using Medoc software. Transfer it to a test PLC and test the programme.*
7. *Using the modelling system, wire the input switches and output devices to the PLC.*
8. *Transfer the programme from the test PLC to the system PLC using an EEPROM and programming panel (ask for instructions on how to do this).*
9. *Demonstrate the system working.*
10. *After testing a prototype machine, it is found that wear is occurring in the system on start up. In order to overcome this it has been decided that there should be a five-second delay between switching on the pump and starting the main motor. Modify the programme using a programming panel and demonstrate it working.*

EXAMPLE 2 – BYTRONICS SORTING MACHINE.

This part is designed for those having access to the Bytronics conveyor/sorting system widely sold to colleges.
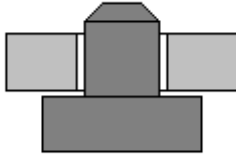
The purpose of the machine is to sort two components, a metal body with a peg, and a plastic ring. The ring is assembled on to the peg. The finished assembly is inspected and components not assembled are rejected.

This work is written for the Mitsubishi PLC but can be changed.

Your overall task is to produce a PLC programme to carry out the above programme. The detailed tasks are listed at the end.

The machine has a start switch and a stop switch as well as an emergency stop that is not connected to the PLC and should not be included in the programme.

The machine has two conveyor belts. One is a chain conveyor on which the two components are placed in any order. The other is the belt conveyor that transports sorted components to the inspection point.

The following details five independent tasks needed to make the machine work. You should devise and test each programme separately and then combine them into a single programme.

Evidence required

1. *a flow chart style program for each section but only if it is a suitable method.*
2. *a truth table for each section but only if it is a suitable method.*
3. *a logic functional block diagram for each section using European logic symbols if applicable but only if it is a suitable method.*
4. *a complete ladder programme and demonstrate it working.*
5. *Use of the name facility on the PLC to allocate tags labels to your inputs and outputs.*
6. *a hard copy of your complete programme.. This should include the ladder and statement version plus the labels and comments..*

PROGRAMME 1   START AND STOP

Devise a simple routine to start the two conveyors when the start switch is pressed and stop when the stop switch is pressed.

The start switch (green button) is a normally open push button and it is designated X406.
The stop switch (red button) is a normally closed push button switch designated X407.
The chain conveyor is driven by a motor connected to the output designated Y430.
The belt conveyor is driven by a motor connected to the output designated Y434.

PROGRAMME 2   SORTING STATION

Devise a routine to detect whether or not the component is a metal peg or a plastic ring. If it is a plastic peg, it must be knocked down the chute into the magazine y the operation of a linear solenoid. If it is a metal peg, it is allowed to pass on and fall down the end chute onto the belt conveyor.
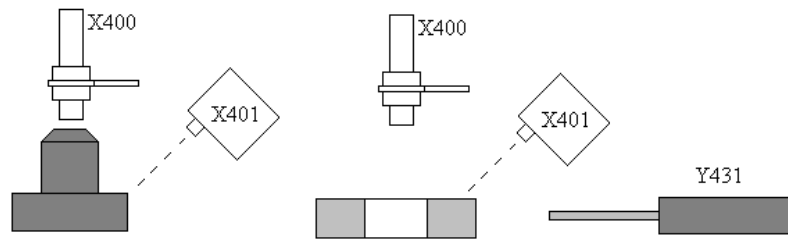
X400 is an inductive proximity switch that comes on when a metal peg is close to it.
X401 is a focussed infra-red detector that comes on when a surface at the focal point reflects light. This comes on when either a plastic ring or a metal peg is passing.
Y431 is the eject mechanism.

PROGRAMME 3   ASSEMBLY STATION
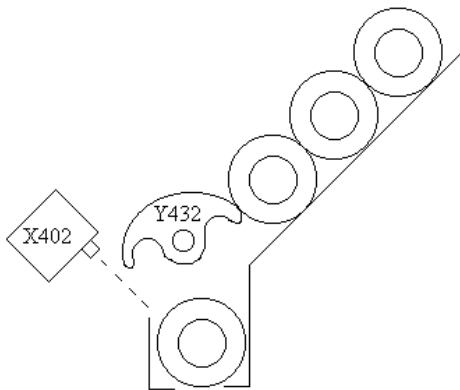
The plastic rings are lined up in the magazine. They are prevented from falling into the assembly station by a rotary solenoid. If the assembly point is empty, the solenoid must operate and allow one ring to enter and close before the next falls on top of it. The rings are automatically dropped onto the metal peg as it passes and carried along with it on the belt conveyor.

X402 is a focussed infra-red sensor that detects if the ring is in the assembly point. Y432 is the rotary solenoid that must operate in order allow one into the assembly point.

PROGRAMME 4   INSPECTION STATION

The components leave the assembly station and pas an inspection station. Here it must be determined whether the components is assembled or just a single components.
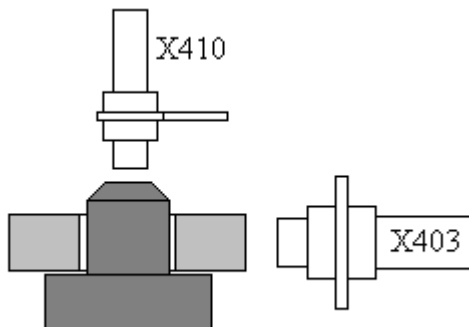
X403 is a capacitive proximity detector at the height of the assembled ring and it is only activated if the ring is present.

X410 is an inductive proximity detector and it is only operated if a metal peg is present.

Note that this programme does not switch anything on physically but it must set some kind of internal flag to tell the eject station to eject when the component passes it.

There is an additional sensor X404 that detects anything passing out of the assembly point.

PROGRAMME 5   REJECTION STATION

The component travels on along the conveyor to the rejection point. Sensor X405 is an infra-red detector that detects the component is at the reject point. If the component is to be rejected (i.e. it's a ring or a peg on its own), then linear solenoid Y433 is operated and knocks it off. If the component is assembled correctly it passes to the end and falls into a bin.

EXTRA WORK - REFINING YOUR PROGRAMME.

There are other sensors present X411 and X412 that detects if the linear solenoids are at the return position. Incorporate these to ensure that neither is stuck on.Use a counter to count the rings in the magazine and inhibit the sort solenoid if it is full so that the rings pass on to the end in the same as the pegs.

EXAMPLE 3 – WATER LEVEL CONTROL

GENERAL DESCRIPTION

The PLC used in this system is a Mitsubishi type FX.

The level of water in the high level tank is to be controlled to any level desired. Water is pumped from the sump into the tank. Water may be emptied from the high level tank to the sump by opening the solenoid valve. The tank has a float switch that is activated when the level is dangerously high.

The level in the tank is measured by an analogue pressure transducer connected to the PLC A/D port.

The level indicator is a 4 – 20 mA meter connected to the PLC D/A port.

The control panel has 6 push button switches that are not latching. Each switch has a lamp built in. In addition there is one extra lamp. The switches are connected to the PLC input terminals and the lamps are connected to the PLC output terminals.
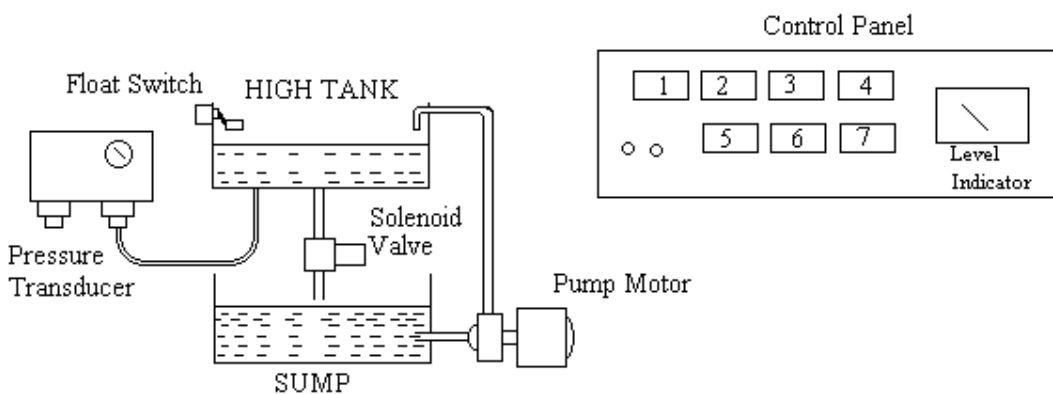
The functions of the Inputs and outputs are as follows. The PLC connection numbers are also shown.

Button 1	**INITIALISE**	X5
Button 2	**LOWER** the level.	X4
Button 3	**RAISE** the level	X3
Button 4	Lamp only	
Button 5	**STOP**	X2
Button 6	Lamp only	
Button 7	**START**	X1
FLOAT SWITCH		X0
Lamp 1	Level too High	Y6
Lamp 2	**LOWER** the level.	Y5
Lamp 3	**RAISE** the level	Y4
Lamp 4	Level too Low	Y3
Lamp 5	**STOP**	Y2
Lamp 6	**HIGH** level.	Y1
Lamp 7	**START**	Y7
PUMP Motor		Y0
SOLENOID Valve		Y8

You are recommended to construct your programme in stages as shown in the following text. Test each routine and use the monitor facility to check that your programme works at each stage.
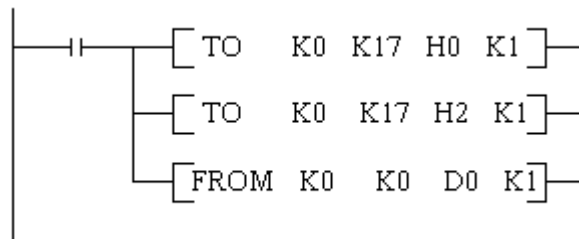
PROGRAMME 1   START AND STOP

Create a routine to start and stop the pump using the non-latching pushbuttons labelled Start and Stop. Arrange that the lamp on start button is on if the pump is running and off when not running. Arrange that the lamp on the stop button be on when the pump is not running and off when it is running.

PROGRAMME 2   HIGH LEVEL CUT OUT

Modify your programme so that when the float switch is operated the pump cuts out and the lamp indicating HIGH level is illuminated.

PROGRAMME 3   READ LEVEL FROM A/D PORT

The routine to read the level from the pressure transducer is shown below. The level is represented by a number in the range 0 – 254 and it is placed in register designated D0.

```
        ┤├─────┬──[ TO    K0  K17  H0  K1 ]──┐
              ├──[ TO    K0  K17  H2  K1 ]──┤
              └──[ FROM  K0   K0   D0  K1 ]──┘
```

PROGRAMME 4   SEND LEVEL TO D/A PORT

The routine to send the level from register D0 to the D/A port is shown below. The meter will receive a current in the range 4 – 20 mA depending on the value in register D0.

```
     ─┤├──┬──[ TO   K0  K16  D0  K1 ]──┐
          ├──[ TO   K0  K17  H4  K1 ]──┤
          └──[ TO   K0  K17  H0  K1 ]──┘
```
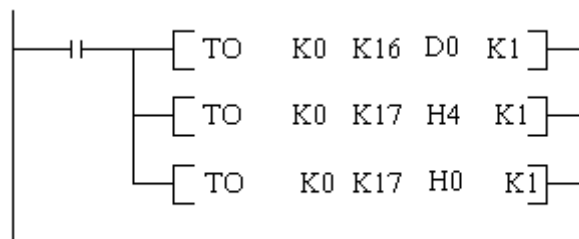
PROGRAMME 5   INITIATE LEVEL

When the initiate button is pressed a middle range figure is placed into register D1. The lamp on the button is illuminated when pressed.

The following command places the number 125 in data register D1.

MOV K125 D1

PROGRAMME 6   RAISE or LOWER LEVEL

The value of register D1 is to be the desired level. To change this use the increment and decrement commands as follows. Use the raise and lower buttons to carry out these commands. The Lamp on the buttons must illuminate when pressed.

INC D1   DEC D1

You will find that the incrementing and decrementing will be too fast. In order to slow it down, use a timer set to 1 second (e.g. T50 K1) and use the timer to initiate the incrementing and decrementing. Make the timer reset itself each time so that the process is repeated so long as the button is pressed.

PROGRAMME 7   DETERMINE IF THE LEVEL IS TOO HIGH OR TOO LOW.

If the value in register D1 is larger than the value in register D0, the level is too high and if it is smaller the level is too low. To determine this use the compare command.

CMP D0   D1   M0

M0 is a flag that comes on when D1<D0
Further flags also are activated as follows.
M1 comes on if D0 = D1
M2 comes on when D1>D0.

Hence M0 indicates the level is too low and M2 indicates the level is too high.

PROGRAMME 8.   CONTROL THE LEVEL

Your final routine must make use of the flags M0 and M2 so that if the level is too high, the pump stops and the valve opens. If the level is too low, the pump must start and the valve must close. This routine will affect earlier routines and you are expected to modify your entire programme as you progress.

Remember that you must demonstrate your programme and produce a fully annotated print off of the work as described earlier.