UNIT 22: PROGRAMMABLE LOGIC CONTROLLERS

Unit code: A/601/1625 QCF level: 4 Credit value: 15

OUTCOME 3 PART 2

This work covers part of outcome 3 of the Edexcel standard module:

Outcome 3 is the most demanding of the outcomes and can only be affectively studied with the use of suitable hardware and/or simulation software such as PneusimPro™ or Bytronics™ simulation software. An industrial background will also be of great benefit to students.

SYLLABUS

3 Be able to apply programmable logic programming techniques

Write programs: use of ladder and logic diagrams; statement lists; Boolean algebra; function diagrams; graphical programming languages; production of a PLC

Advanced functions: less than; greater than;

binary to BCD conversion; proportional feedback control

Producing and storing text: contact labels; rung labels; programming lists; cross-referencing

Test and debug programs: forcing inputs, forcing outputs; changing data; comparing files (tapes, EPROM, disc); displayed error analysis

Associated elements: contacts; coils; timers; counters; override facilities; flip-flops; shift registers; sequencers

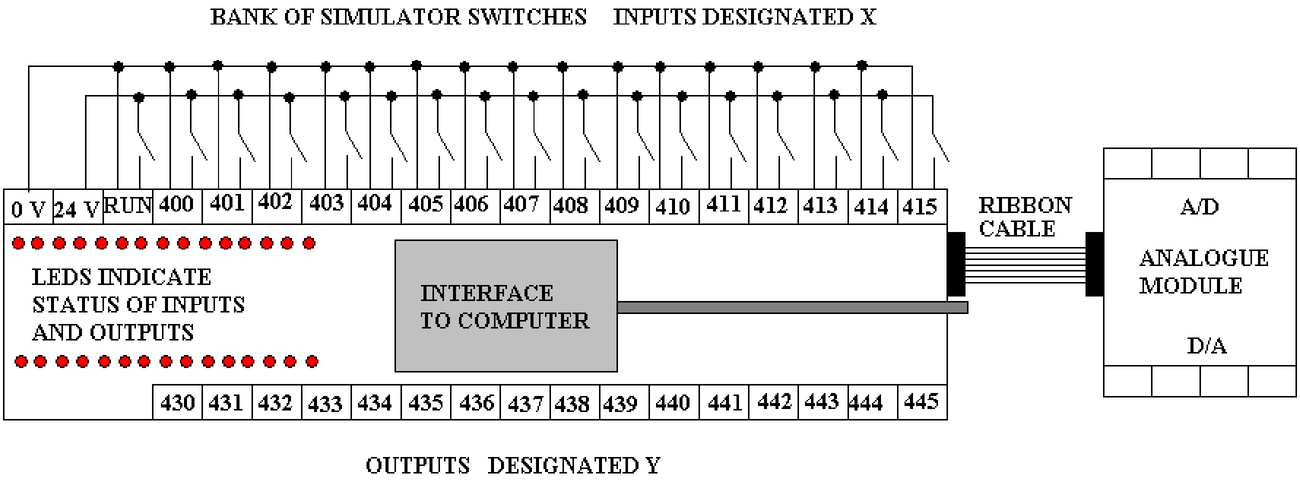
Learning outcomes **On successful completion** **of this unit a learner will:**	**Assessment criteria for pass** **The learner can:**
L03 Be able to apply programmable logic programming techniques	3.1 identify elements associated with the preparation of a programmable logic controller program 3.2 write programs using logic functions based on relay ladder logic 3.3 evaluate the range and type of advanced functions of programmable logic controllers 3.4 use and justify methods of testing and debugging hardware and software

The following work is a set of exercises with explanations based around the Mitsubishi PLC functions and the Medoc programming software. It can be adapted for similar PLCs. Some information on programming the FX series is contained in assignment 3.

CONTENTS

TESTING AND DEBUGGING

The diagram illustrates the layout of a typical PLC. The Light Emitting Diodes are a useful feature to help monitor and debug the programme. The input terminals have a bank of switches attached to it to enable each to be set high or low. These are normally available as standard for simulation, testing and debugging.



The main monitoring and debugging tools are in the software used to programme the PLC. If the PLC is connected to the computer with a suitable interface, the programmes may be moved either way between the PC and the PLC. The Medoc software highlights the parts of the ladder diagram that are active and the status of timers, counters and registers are displayed at the bottom of the screen.

The following work is presented as a series of exercises based on the Mitsubishi PLC and Medoc software but it may be adapted to other types.

With the Medoc software, you may monitor the PLC as follows.

Enter the programme and go into ladder edit mode using F2 and F8.

The contents of timers and counters are automatically displayed at the bottom of the screen by pressing the F5 key.

You must enter the number of the registers that you wish to monitor e.g. K700 monitors data register D700.

Press F8 to resume monitoring.

Remember that the run switch on the PLC must be closed before anything takes affect.

EXERCISE No.1 - TIMERS

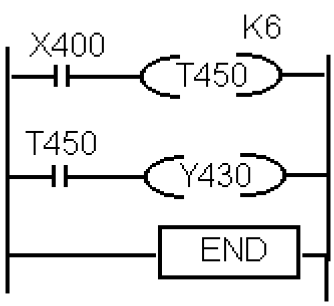
On completion of this exercise you should be able to programme a simple timer sequence into a PLC. You will need a Mitsubishi PLC (F20) linked to a P.C. with Medoc software.

Numeric Designations T50 to 57 and T 450 to 457
Typical Use Timing a heating process.

Consider a machine which has to insert a component into a heat treatment oven for a fixed time and then remove it.

Let the timer used be T450
Let the input used to start the timer be X400
Let the output used to start the removal routine be Y430. (Note it might be a jump to a new routine).
K is the time delay in seconds.

LADDER DIAGRAM



INSTRUCTION SET

LD 400
OUT 450
K 6
LD 450
OUT 430
END

EXPLANATION
When contact 400 is closed the timer starts running. After K seconds (set to 6 here) the timer contact closes and switches on output 430. If the input contacts X400 are open, the timer is reset.

Enter the above programme into your PLC and test it.

EXERCISE No.2 - COUNTERS

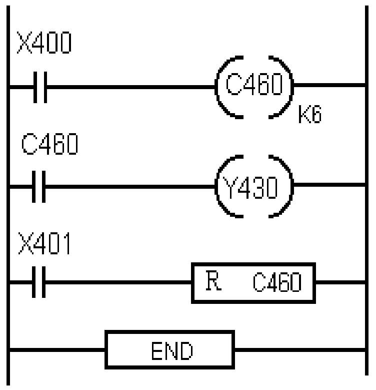
On completion of this exercise you should be able to enter simple counter programme into a PLC. You will need a Mitsubishi PLC (F20) and pc with Medoc software.

Numeric Designations C 60 to 67 and C 460 to 467
Typical Use Batching

Consider a machine which has to inspect a component one at a time, set aside the rejects and stack the good ones in a packing case. Every time K good ones have been stacked, the machine must go into a new routine to pack the components, remove the packing case and then start on a new batch.

Let the counter used be C460
Let the input used to reset the counter be X400
Let the input used to count components be X401
Let the output used to start the packaging routine be Y430. (Note it might be a jump to a new routine).

LADDER DIAGRAM



INSTRUCTION SET

LD 400
OUT 460
K 6
LD 460
OUT 430
LD 401
RST 460
END

EXPLANATION
When contact 401 is closed the counter is reset to zero. Each time contact 400 is closed the count is decremented by 1. When the counter value reaches zero, the counter contacts close. When the counter contact closes, output 430 is switched on.

Enter the above programme into your PLC and test it.

EXERCISE No.3 - REGISTERS

INFORMATION ON MITSUBISHI REGISTERS

Numeric Designations of registers	M100 - 177	general
	M200 - 277	general
	M300 - 377	retentive (battery back up)
	70 - 77	dedicated
	470 - 477	dedicated
	570 - 577	dedicated
	670 - 677	dedicated
	D700 to 777 are data registers.	

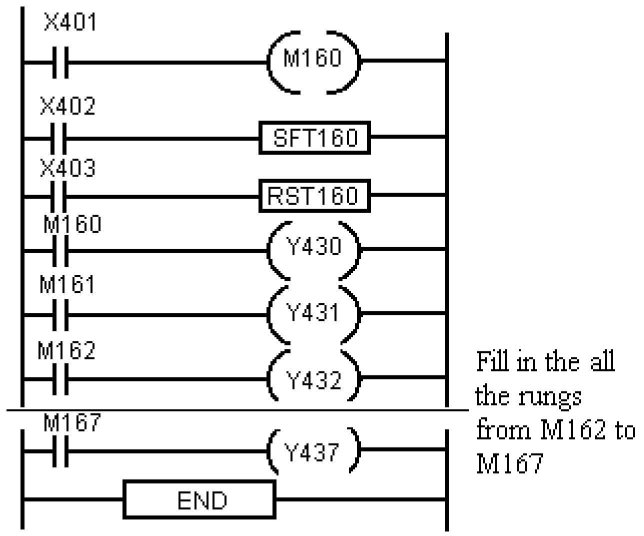
- A register may be thought of as a bank of elements (called auxiliary relays) which are on or off. In our case there are 16 in each bank. Each bank starts with a number ending in zero e.g. M160 and ends with a number ending in 5 e.g. M175.
- Each element may be addressed and used individually and used for flagging operations. These may also be thought of as bits in a binary code.
- If a bank is addressed (e.g. M160) then you cannot address individual elements in it but you can use them as inputs.
- The registers in the range D700 - D777 are data registers for numeric operations in advanced programming (F20 only, not F12).
- A register may be reset with the RST command. This sets the first element on or high.
- A register may be shifted using the SFT command. This moves the pattern of high and low settings along one element.

On completion of this exercise you should be able to do the following.

- Programme a register into a PLC and shift the register.
- Monitor the contents of the register.
- Decrement a register.
- Increment a register.
- Shift a register.
- Understand how a pattern represents a 3 digit binary decimal coded number BCD).

PART 1 SHIFTING AN AUXILIARY REGISTER

Enter the programme into your PLC using Medoc software and test it. It will be used to demonstrate a register shifting.



LD 401
OUT 160
LD 402
SFT 160
LD 403
RST 160
LOAD 160
OUT 430
LD 161
OUT 431
LD 162
OUT 432 and so on to
LD 167
OUT 437
END

EXPLANATION

When input X401 is on the bank of auxiliary relays M160 to M175 are recognised as a register and bit 0 (M160) is set high. Each time input X402 is pulsed, the bit is moved along to M161, M162 and so on.

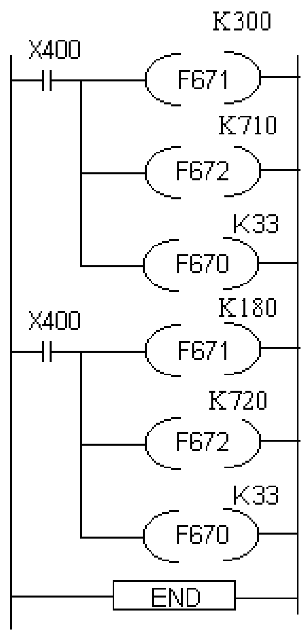
Each individual bit is used to turn on a corresponding output in order to indicate its status. For example if bit M165 is high, then output Y435 is turned on.

Switch on X401 and output 430 should light up.
Each time input X402 is pulsed the bit moves along and the next output light ups.
When input X403 is pulsed, the register is reset.
If X401 is off a low is loaded into M160.
If a shift is performed, the low (light off) is carried along with each shift.
By switching X401 on or off and shifting, it is possible to arrange any pattern on the register.

When you are satisfied that your programme works, produce a binary pattern on the 6 outputs to represent a decimal number of 375

PART 2 PERFORMING OPERATIONS ON A DATA REGISTER

POKING



The next programme pokes numbers direct to a register. Enter the programme into the PLC using Medoc software. Note that data registers are designated by the letter K in the programming.

X400 activates the programme.

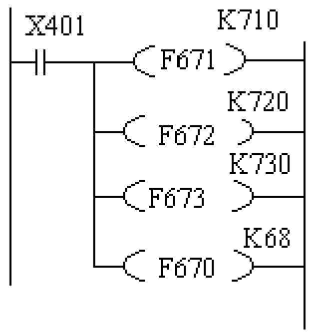
300 is the first number to be stored and it is stored in data register 710.

F670 K33 performs the operation.

180 is the second number to be stored and it is stored in data register 720.

Transfer the programme into the PLC and run it. Monitor the data registers and note that the numbers 300 and 180 are stored in the appropriate register.

SUBTRACTING



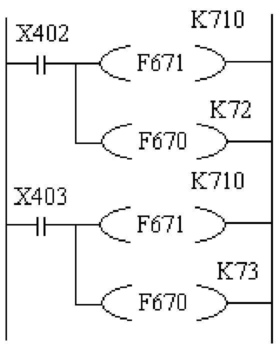
Add the following programme to the last.
When X401 is activated the contents of register 720 is subtracted from the contents of register 710 and the result is placed in register 730.
F671 K710 defines the first data register as D710
F672 K720 defines the second data register as D720.
F670 K68 subtracts one from the other places the result in D730.

Enter the above programme and monitor the contents of the data registers. You should see the result of the subtraction in register K730.

DIVIDING

If the command F670 K83 is used, register 710 is divided by register 720 and displayed in register 730. Change your programme and demonstrate that you can divide say 770 by 11 and display the result.

INCREMENTING AND DECREMENTING



Add the following programme to the existing one and monitor all three data registers in use. Switch off X400 so that it does not overwrite the contents of K710.

Each time X402 is pulsed, the contents of register 710 are increased by 1 and the changed result of the subtraction should be seen. The instruction F670 K72 does this.

Each time X403 is pulsed, the contents of register 710 are reduced by 1 and the changed result of the subtraction should be seen. The instruction F670 K73 does this.

EXERCISE No.4 - ANALOGUE/DIGITAL CONTROL

On completion of this exercise you should be able to do the following.

- Interface an A/D and D/A module to the Mitsubishi PLC.
- Read an analogue signal into a data register and monitor it.
- Write to an analogue port and monitor the output.

In order to complete this assignment you must

- Read the notes and manuals for the Mitsubishi PLC.
- Read the manuals for programming the PLC with a keyboard.
- Read the manual for the Medoc software.
- Read your notes and text books on A/D and D/A conversion.
- Read the RS data sheet on the I/O module.

It is advised that MEDOC programming be used.

The following information should be read carefully before attempting to programme the PLC.

The I/O module is connected to the PLC by a ribbon cable to the socket numbered 400. The module contains 4 input ports numbered 410, 411, 412 and 413. It also contains two output ports numbered 400 and 401.

The ports may be configured with dip switches to use 0 -5 V, 0 -10V, 0 -20 mA or 4 - 20 mA. For this assignment they should be configured for 0 - 10V.

There are 64 data registers starting at address 700 (octal). In medoc programming they are prefixed with a K and they must have an address ending with a zero (e.g. 720 but not 721). This function is only available on the types F_1 20 and F_1 12 made after Sept 1988. Check the serial number.

Data may be placed in these registers from timers, counters, analogue inputs or from the programme. They may be manipulated (incremented, decremented, added, subtracted, divided and so on). The manipulated data may be placed back to the counter, timer, analogue output etc.

The commands to specify the manipulation are carried out with an instruction F670 followed by a K value to specify the type of manipulation.

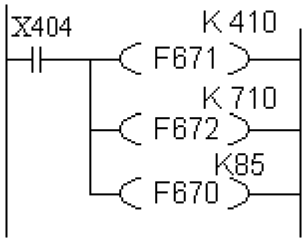
Data to be used in the command is entered with F671, F672 and so on followed by a K number to specify the data or its address.

For each rung with an F670 command, you must store the data in F671, F672 and so on. These may be used several times in a programme but only once in each ladder rung with an F670 command.

Load the complete programme from exercise 3. Add the following routines to demonstrate use of the A/D and D/A module.

PART 1. READ ANALOGUE INPUT AND PLACE IN REGISTER.

The following routine is used to read an analogue signal and place the digital data into a data register.

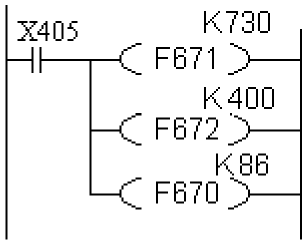


X404 is used to activate the operation.
F671 K410 is the functional instruction to prepare analogue port 410.
F672 K710 is the functional instruction to place the data in data register D710.
F670 K85 is the functional instruction to carry out the operation.

Enter the above routine. Switch off X400 so that it does not overwrite the contents of K710. Connect analogue port 410 to a 0-10V source and monitor the digital value in register 710 using F8 F5 online monitoring. Note that the subtraction routine from the last worksheet still works.

PART 2. WRITES TO ANALOGUE OUTPUT PORT.

The following routine takes data from a data register and places it to the analogue output port.



F671 K730 defines data register D712 as the source.

F672 K400 defines output channel 400 for the analogue data.

F670 K86 performs the operation.

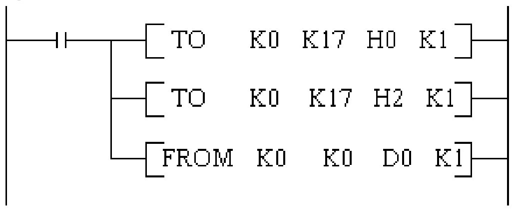
Add the programme above and monitor the output of the port with a voltmeter or ammeter. Change the contents of K710 by changing the analogue input. Note that this changes the result of subtraction in K730 and so the analogue output changes as the contents of K730 changes. Also increment and decrement K710 and note that you can make the analogue output go up or down.

THE MITSUBISHI FX SERIES PROGRAMMES

The following shows how to do the above the exercise with the FX series.

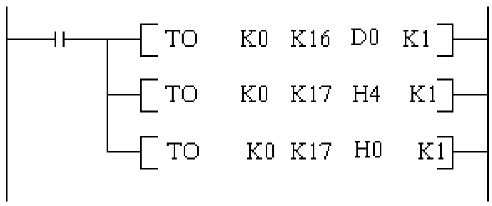
READ FROM A/D PORT

This routine reads the analogue input port. The result is represented by a number in the range 0 – 254 and it is placed in register designated D0.



WRITE TO A/D PORT

This routine takes the digital information in D0 and places it as an analogue signal on the Analogue output port.



P.I.D. and PLC PROGRAMMING

To understand this topic you need to study the stand alone tutorial on 3 term control. It is a major topic and it is doubtful that the module designers intended you to study the topic.

Briefly, when a PLC is used to control a system such as an electric motor, the system error is reduced by using 3 term control. The three terms are Proportional, Integral and Derivative so it is referred to as P.I.D. control. Analogue systems have controllers that provide this function. Usually you must set three constants to optimise the system performance.

Many systems are controlled with a PLC using analogue/digital conversion. The signals are digitally processed. Some types of PLC's have a programming command to calculate the incremental change in the following.

- The change in the error.
- The change in the time integral of the error.
- The change in the rate of change of the error.

Each of these are multiplied by a constant and the result is added to the previous value.

The PLC is programmed with the three constants. Typically this command to invoke the processing is PIDINC. Often the sampling period is part of the command.

The error is found by subtracting the output signal from the input signal (this being the correct value). If these are analogue signals they must be converted into digital form. These values would be read into the PLC and stored. This would be repeated at intervals of T seconds. Three successive samples would be stored and then updated every T seconds. The numerical processing is carried out. A further command to add the incremental change to the last value must also be used and the resulting value placed out on the I/O module as an analogue signal. A typical ladder diagram would contain the sub-routine shown below to do this. The exact diagram will depend upon the type of PLC used.