NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical&Electronic Engineering

AUTOMATIC CAR WASH SYSTEM WITH PLC

Graduation Project
EE-400

Student: Murtaza Deniz

Supervisor: Özgür C. Özerdem

Nicosia - 2001

ACKNOWLEDGEMENTS

I must first acknowledge to my parents for their support to an upwardly mobile son. To my friends for they helps.

I want to thanks separately to my teachers that during my education they support me, the Dean of Engineering Faculty and Head of Electric & Electronic Department Prof. Dr Fafhreddin MAMEDOV and also to my supervisor Mr. Ozgur C. OZERDEM.

And finally, I want to thanks everybody who helped and suppoted me to come to todays and graduated from this university.

ABSTRACT

PLC (Programmable Logic Controllers) is a thing that programmable with computer support to take more efficiency from time and workers. It it divided into two parts. Hardware and software.

The hardware are parts of machine those are CPU, I / O device and programming device. CPU is basic microprocessor system and it carries out as control sensors, counters, timer functions. CPU carries out stored user program in memory will input information from various sensor circuits and can suitable output to commands and control circuits. There are command which are mathematics process that are comparator processes. These are the main function and feature of software part of PLC.

My project is Automatic Car Wash System with PLC. I must used Compact Type PLC for this system, and the material expences for PLC (Siemens 100U); 944 euro , and for sensors; 24 euro.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I must first acknowledge to my parents for their support to an upwardly mobile son. To my friends for they helps.

I want to thanks separately to my teachers that during my education they support me, the Dean of Engineering Faculty and Head of Electric & Electronic Department Prof. Dr Fafhreddin MAMEDOV and also to my supervisor Mr. Ozgur C. OZERDEM.

And finally, I want to thanks everybody who helped and suppoted me to come to todays and graduated from this university.

ABSTRACT

PLC (Programmable Logic Controllers) is a thing that programmable with computer support to take more efficiency from time and workers. It it divided into two parts. Hardware and software.

The hardware are parts of machine those are CPU, I / O device and programming device. CPU is basic microprocessor system and it carries out as control sensors, counters, timer functions. CPU carries out stored user program in memory will input information from various sensor circuits and can suitable output to commands and control circuits. There are command which are mathematics process that are comparator processes. These are the main function and feature of software part of PLC.

My project is Automatic Car Wash System with PLC. I must used Compact Type PLC for this system, and the material expences for PLC (Siemens 100U); 944 euro , and for sensors; 24 euro.

TABLE OF CONTENTS

INTRODUCTION

In the late 1960's PLCs were first introduced. The primary reason for designing such a device was eliminating the large cost involved in replacing the complicated relay based machine control systems. Bedford Associates (Bedford, MA) proposed something called a Modular Digital Controller (MODJCON) to a major US car manufacturer. Other companies at the time proposed computer based schemes, one of which was based upon the PDP-8. The MODICON 084 brought the world's first PLC into commercial production.

When production requirements changed so did the control system. This becomes very expensive when the change is frequent. Since relays are mechanical devices they also have a limited lifetime which required strict adhesion to maintenance schedules. Troubleshooting was also quite tedious when so many relays are involved. Now picture a machine control panel that included many, possibly hundreds or thousands, of individual relays. The size could be mind boggling. How about the complicated initial wiring of so many individual devices! These relays would be individually wired together in a manner that would yield the desired outcome.

These "new controllers" also had to be easily programmed by maintenance and plant engineers. The lifetime had to be long and programming changes easily performed. They also had to survive the harsh industrial environment. That's a lot to ask! The answers were to use a programming technique most people were already familiar with and replace mechanical parts with solid-state ones.

In the mid70's the dominant PLC technologies were sequencer state-machines and the bit-slice based CPU. The AMD 2901 and 2903 were quite popular in Modicon and A-B PLCs. Conventional microprocessors lacked the power to quickly solve PLC logic in all but the smallest PLCs. As conventional microprocessors evolved, larger and larger PLCs were being based upon them. However, even today some are still based upon the 2903. Modicon has yet to build a faster PLC than there 984A/B/X, which was based upon the 2901.

Communications abilities began to appear in approximately 1973. The first such system was Modicon's Modbus. The PLC could now talk to other PLCs and they could be far away from the actual machine they were controlling. They could also now be used to send and receive varying voltages to allow them to enter the analogue world. Unfortunately, the lack of standardization coupled with continually changing technology has made PLC communications a nightmare of incompatible protocols and physical networks.

The 80's saw an attempt to standardize communications with General Motor's manufacturing automation protocol (MAP). It was also a time for reducing the size of c PLC and making them software programmable through symbolic programming on personal computers instead of dedicated programming terminals or handheld programmers.

The 90's have seen a gradual reduction in the introduction of new protocols, and the modernization of the physical layers of some of the more popular protocols that survived the 1980's. The latest standard (TEC 1131-3) has tried to merge plc-programming languages under one international standard. We now have PLCs that are programmable in function block diagrams, instruction lists, C and structured text all at the same time! PC's are also being used to replace PLCs in some applications. The original company who commissioned the MODICON 084 has actually switched to a PC based control system.

CHAPTER ONE

1. WHAT IS PLC

A programmable logic controller (PLC) is a device that was invented to replace the necessary sequential relay circuits for machine control. The PLC works by looking at its inputs and depending upon their state, turning on / off its outputs. The user enters a program, usually via software, that gives the desired results.

PLC's are used in many real word applications. If there is industry present, chances are good that there is a PLC present. If you are involved in machining, packaging, material handling, automated assembly or countless other industries you are probably already using them. If you are not, you are wasting money and time. Almost any application that needs some type of electrical control has a need for a PLC.

For example, let's assume that when a switch turns on we want turn a solenoid on for 5 seconds and then turn it off regardless of how long the switch is on for. We can do this with a simple external timer. But what if the process included 10 switches and solenoids? We would need 10 external timers. What if the process also needed to count how many times the switches individually turned on? We need a lot of external counters.

As you can see the bigger the process the more of a need we have for a PLC. We can simply program the PLC to count its inputs and turn the solenoids on for the specified time.

This site gives you enough information to be able to write programs far more complicated than the simple one above. We will take a look at what is considered to be the' top 20' PLC instructions. It can be safely estimated that with a firm understanding of these instructions one can solve more than 80 % of the applications inexistence.

1.1. THE TYPES OF PLC

In general, PLC divides to three sections;

*Central Processing unit(CPU)

*The input/output section

*The programming device

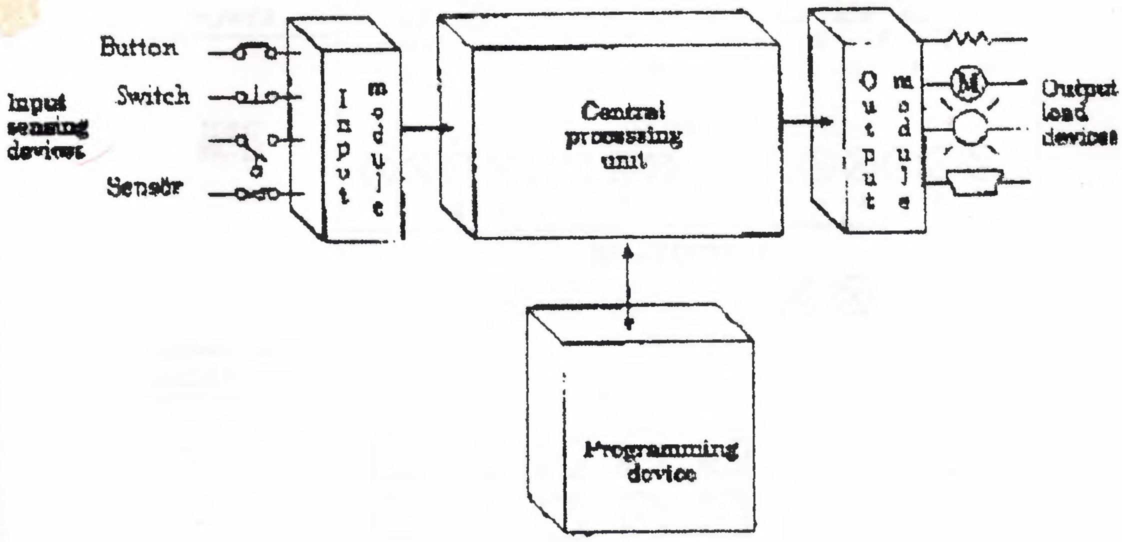


Figure 1.1 PLC sections

(CPU), PLC system and there are various logic circuit gates. CPU is basic microprocessor system and it carries out as control relay, counter, timer functions. CPU carries out user programs stored in memory and read input data from various sensor circuits and can send suitable outputs to commands and to control circuits.

Direct current power supply must be used for the low level voltage that these are using in processor and I/O models. This power supply is a part of CPU. PLC system is independent in its structure and also it can be dependent to its system.

I/O system forms can be connected to controller by other devices. The aim of interface is to send various signals and to take situations to external devices. The output devices for example, motor starters, solenoid valves, indicator lights connected to terminals on the output module.

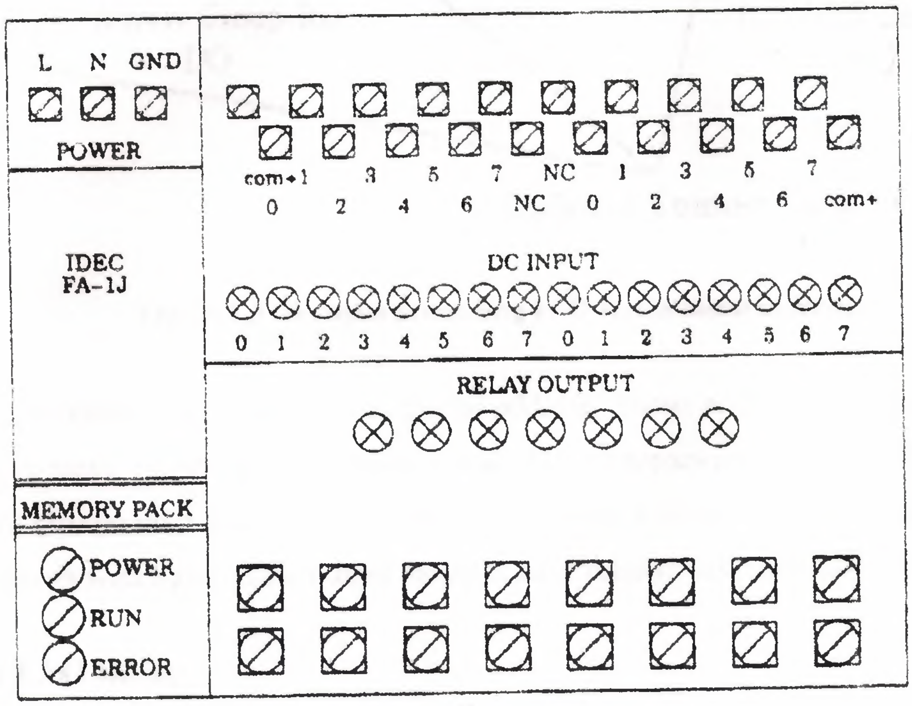


Figure 1.2 In the same structure CPU with PLC I / O unit

Between processor and I / O rack communication different connection cable are permitted. This condition is as the following Figure 1.3 .

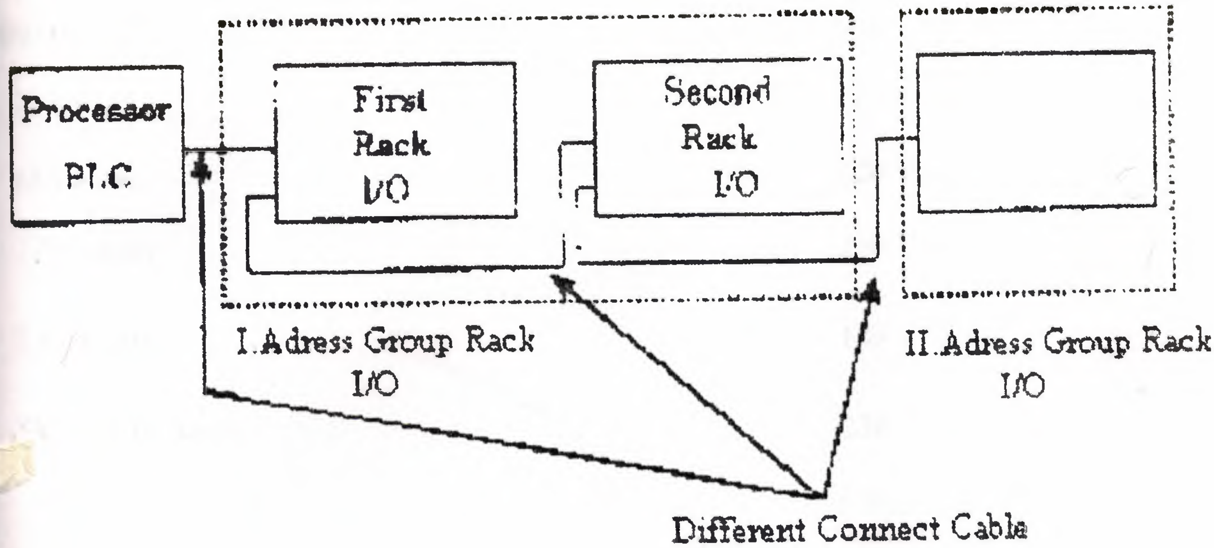


Figure 1.3 Between processing I / O Racks communication

I/O units each input/output has a special address. These addresses are known h~ the processor. To connect output/input an element with 110 or separating is very easy and quick. Furthermore to change with an another module is very easy. ON/OFF condition of 110 circuit each module shows with light. Many output modules have rubbish fuse indicator.

i) Different I / O units:

Many output I/O units are from this type and most useful is interface module. This type interface provides to link of inputs as selector switches push buttons and limits switches. However, output control lights small motor solenoids sensor and motor starters limit it. Which have ON/OFF contacting control. Each different I/O module takes its power from common voltage sources. These voltages can be different size and type. These are showed in the following table.

Input Interface

- 24Vac/dc
- 48Vac/dc
- 120 Vac/dc
- 230 Vac/dc
- 5Vdc (TTL level)

Output Interface

- 12-48 Vac
- 120 Vac
- 230 Vac
- 120 Vdc
- 230 Vdc
- 5 Vdc (TTL level)

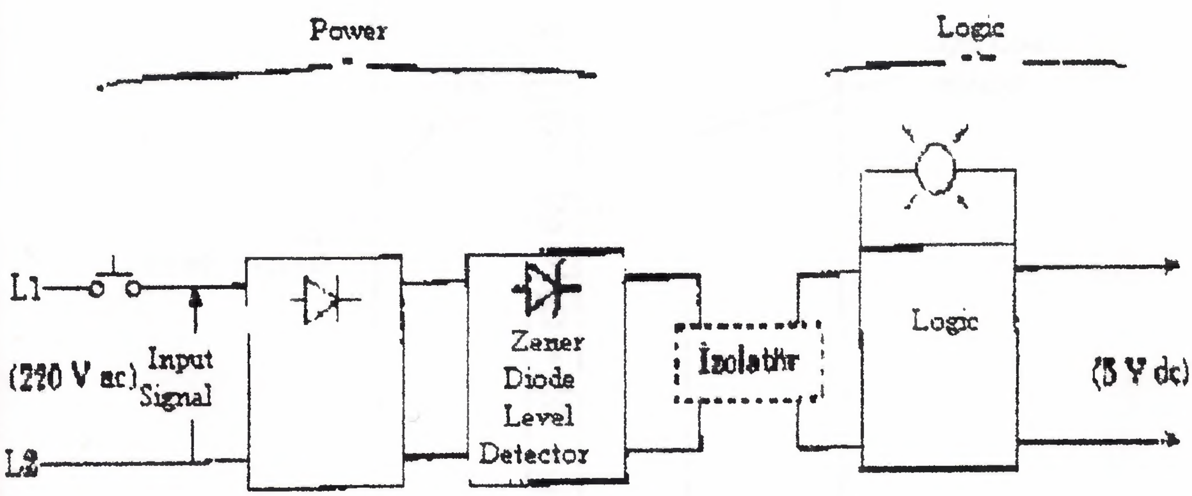


Figure 1.4 AC input interface block diagram

Shows that entries block diagram for an alternative current to input module. 22ut circuit compose of to main section as power and logic section.

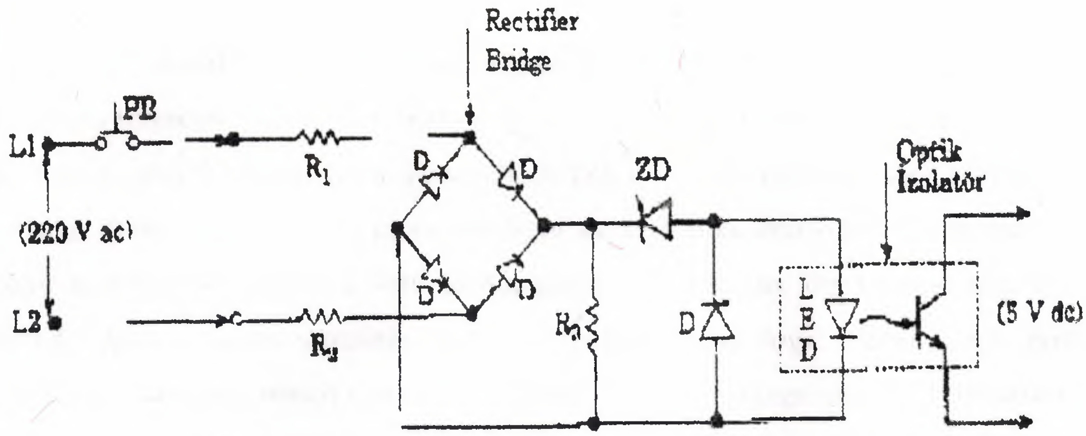


Figure 1.5 Simplified Circuit for an AC Module

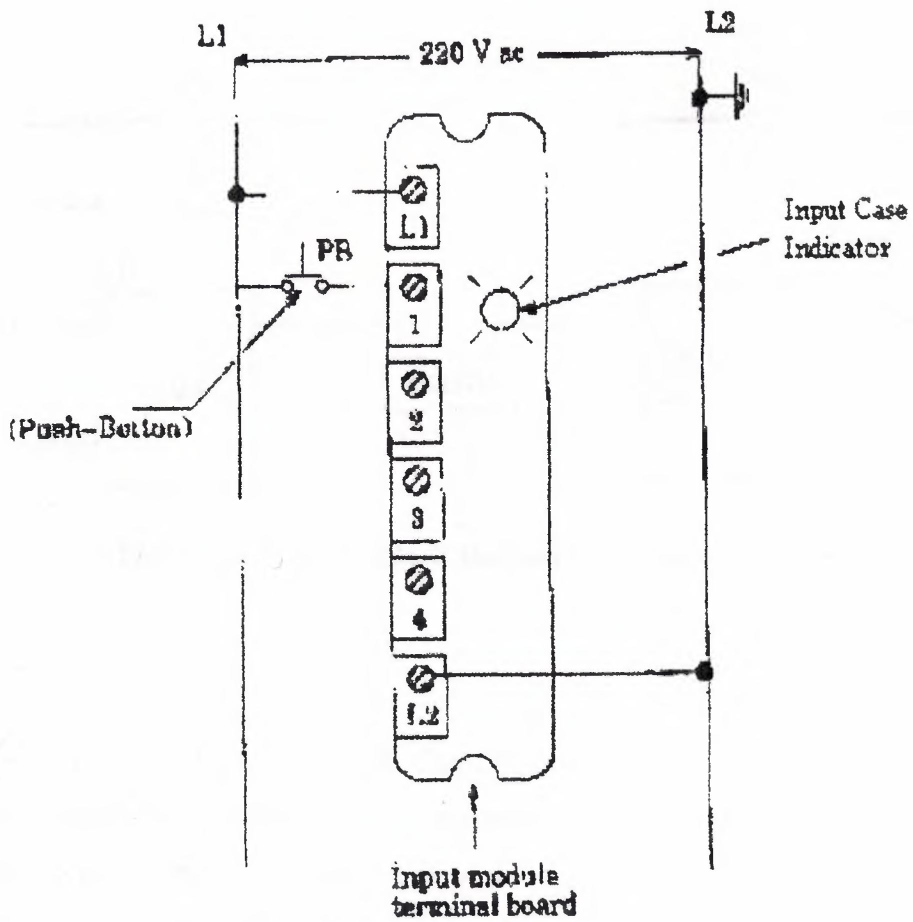


Figure 1.6 Linking to PLC Input Unit of 220V Input

Figure 1.4 and 1.5 shows figural diagram of Ac input module for input, also figure 1.6 shows connect terminal. When push button shuts down, bridge type treatment exercise 220V AC voltage from R_1 and R_2 resistance's. Zener diode (ZD) voltage limit regulates according to low level voltage. When light come to processor from led with phototransistor that means low When light come to processor from led with phototransistor that means low level voltage (SV'dc) is transmitted. Optic isolator separates high AC voltage from logic circuits also protects to processor from damages, which comes from temporary line voltage change. Furthermore, optic isolator protects to processor from effect of electrical noise. Kuplaj and isolation can be created with using a pulse transformation.

Kuplaj and isolation can be created with using a pulse transformation.

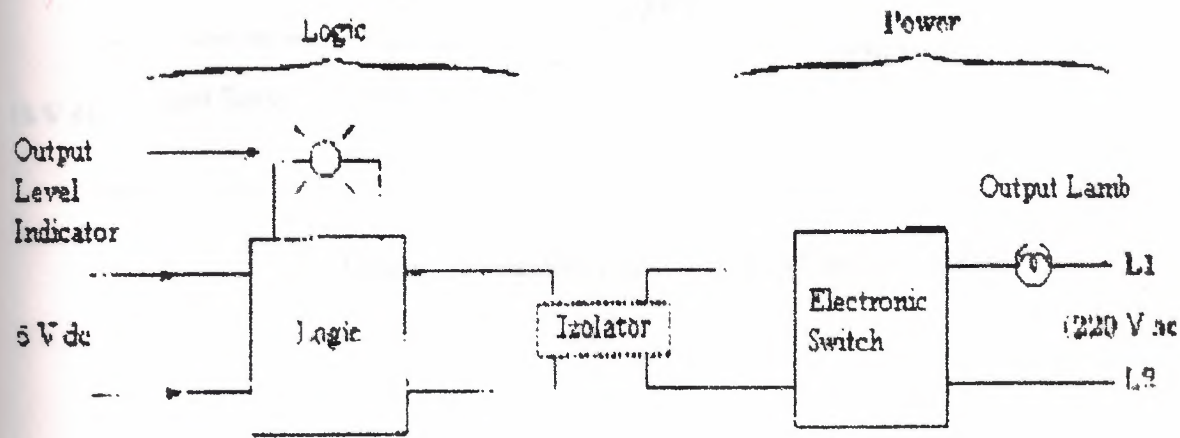


Figure 1.7 Typical a block diagram of output interface module

Figure 1.7 shows typical a block diagram of output interface module. Also output module, as input module, composes of two departments such as power and logic. Device in output is controlled by the 5V comes from logic unit. In this unit, processor sets output conditions. When processor, led, in optic isolator, distributing light exercises an output voltage (5V' dc), however, phototransistor is switching and conducting. This means that to detect and conduct of triac, and lamp, that uses as output element, turn on ON condition. When led in logic unit turn off, logic become 0 condition and phototransistor cannot conduct. If a DC device in output will be controlled, it is carried with circuit.

PLC device will not be damaged from optic isolation that will be from power department. If many high fast ON-OFF is necessary, in right current transistor and also alternative current triac circuits are used. Current cannot pull on PLC from output modules. Maximum current capacity of each device exists in their catalogs of that model.

In high currents instead of triac or other effect elements, standard relay must use as table 6. There are output/input unit as analog/digital translator (ADC) and digital/analog translator (DAC) that it is necessary for feedback control exercises in PLC devices.

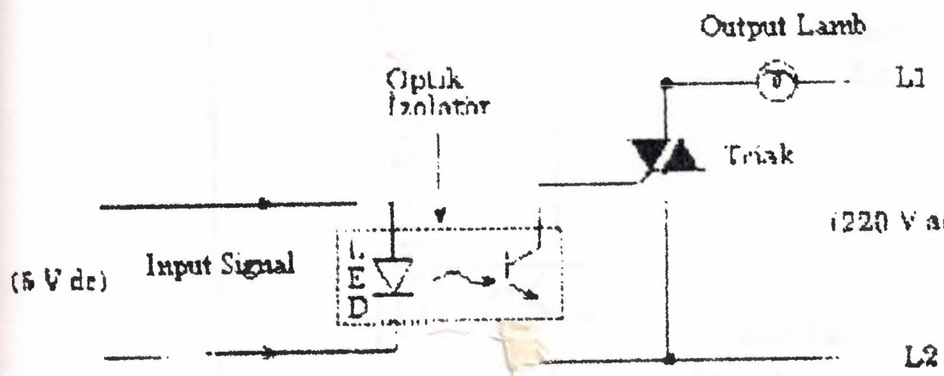


Figure 1.8 Simplified circuit of an AC output module

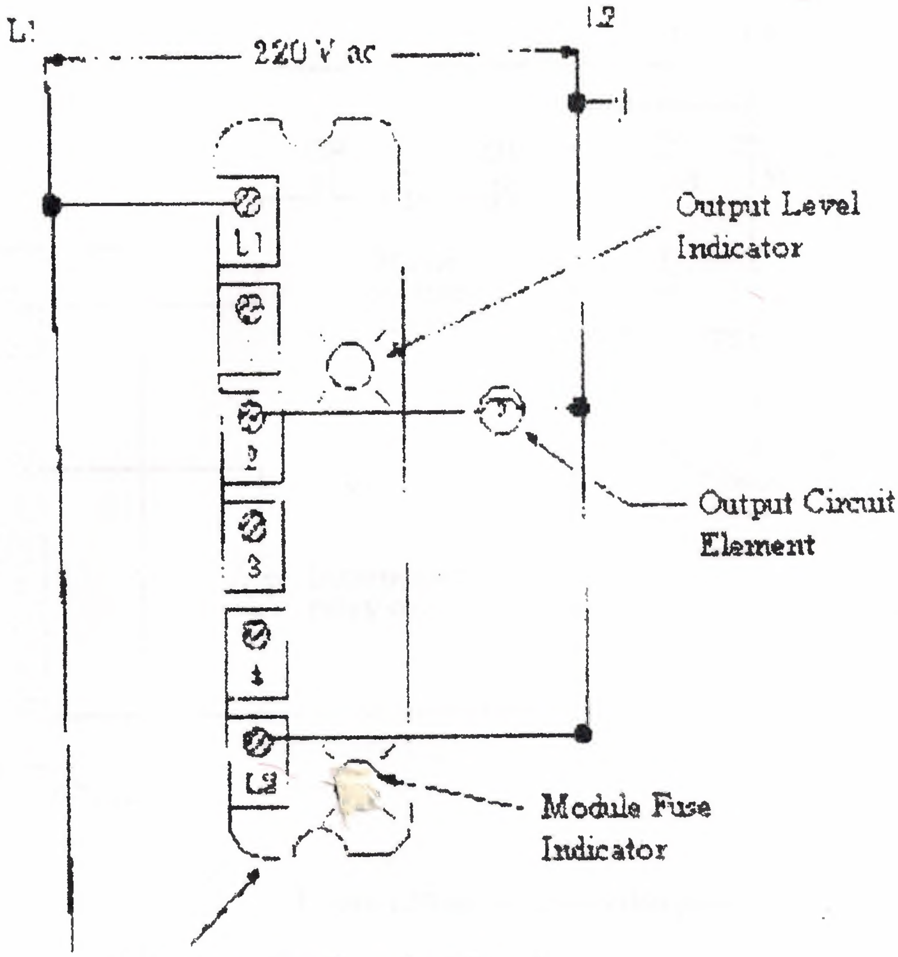


Figure 1.9 Internal wire connection typical an output module

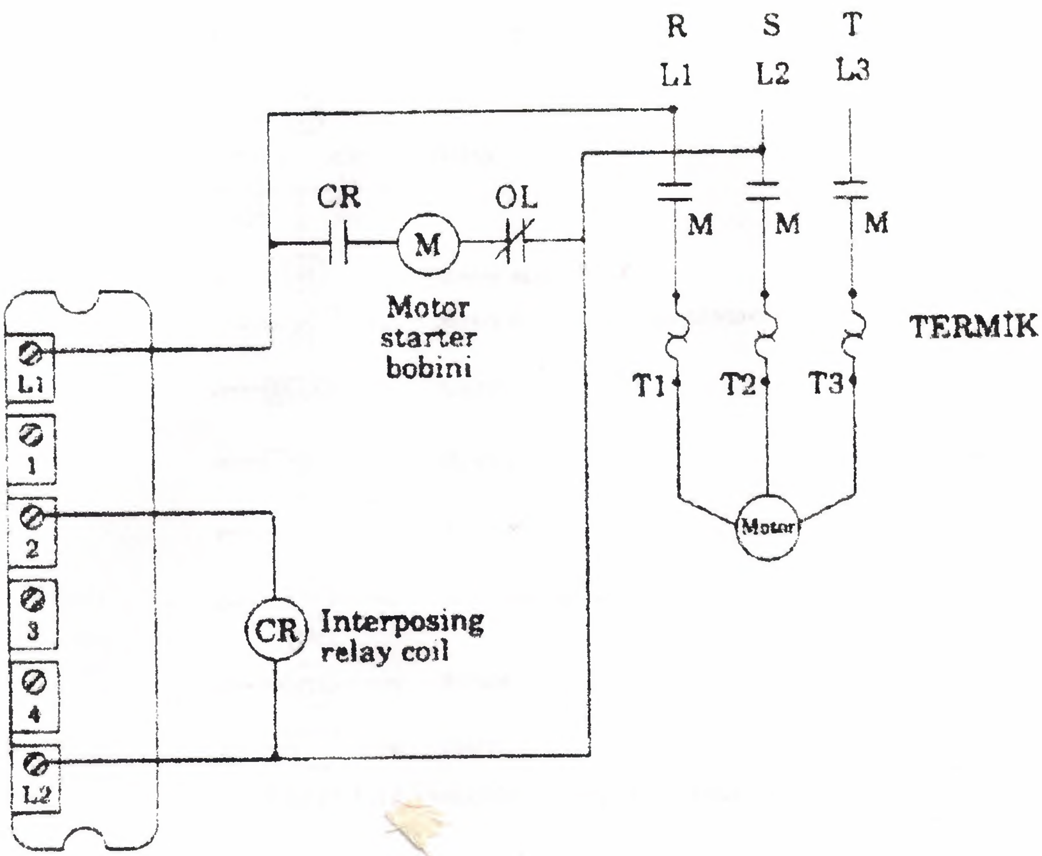


Figure 1.10 sensor connection points

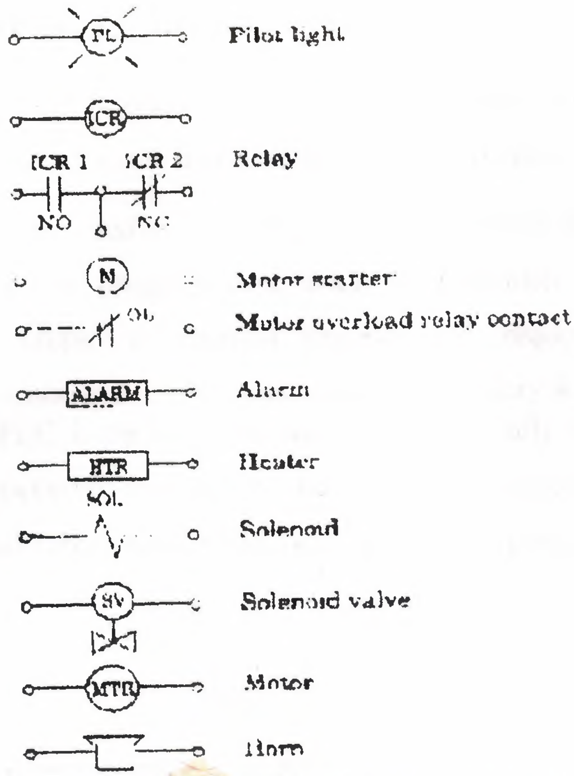


Figure 1.1. Symbols of output control circuit

ii) Analog input/output unit (110 modules):

First produced PLCs only had been limited with separate 110 interfaces which had been allow to link to ON/OFF device. Because of this limitation many of processing exercises could be as part controlling by PLC. Also in days PLCs included analog interface and separate (110) input/output interface, which carries out practically many of control process. An analog input module takes analog current and voltage that is taken off analog input and it changed to digital data form by an Analog Digital Converter (ADC). In this condition turning levels are shown as 12-bit binary or 3 digit BCD that is rates with analog signal. Analog sensor elements are transducers as heat, light, velocity, pressure, and wet sensors. All these sensors can be linked to analog input

Analog output interface module takes digital data from processor, charges rate with voltage and current and controls a device as analog. As a whole digital data passes from Digital/Analog output device are small motors, valves and analog measure devices.

iii) CPU (Central Processing Unit):

Central Processing Unit provides to communicate between power supply and processor memory modules. In figure 1.2. b it can find covered both of two units.

CPU statement is often used as mean of processor statement. Processor-memory creates a big unit of CPU, which is programmable brain of controller. In this unit, there are microprocessor, memory chips, information reading and request data from memory, programming device and communication circuits, which is necessary for processor.

Development of PLC is parallel with increasing especially of CPU. In our day PLC systems carry out logic processing furthermore they have some especially such timer, counter, data storing, main addition-subtraction, multiplication-division processes, compare processes, code converter processes.

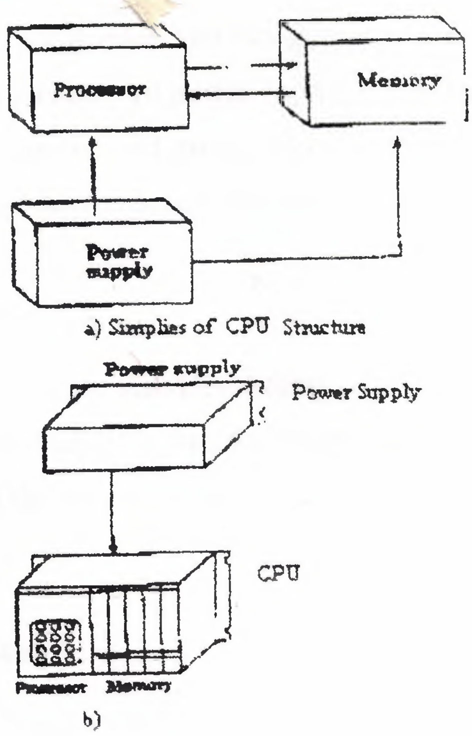


Figure 1.12 CPU; the elements of central processing unit a) the structure of simplified CPU , b) power supply unit different from CPU

iv) Processor-Memory Module:

CPU is the brain of programmable of controller and a big part of CPU family forms from processor memory unit. This module cover microprocessor, memory chips programming device and necessarily communication circuits for processor interface.

Furthermore processor carries out other functions. For example, it carries out timer, counter, compare, keeper and addition, subtraction, multiplication and division functions, which are four main functions of mathematics.

1.2. MEMORY DESIGN

Memory is used to store data. This stored information is related with which output sign will be store as, which shows input, and the structure of program necessary amount of memory. It stores special information parts which is named as memory bit. 1 byte 8 bit, IO24byte 1kbyte and the number of memory capacity is stated these units.

The memory types are divided into two groups;

The first group: the energy of power supply is cut that supplied memory, it means that memory had been erased. Also second group: hide information cannot lose if the energy is cut. But to change of includes of those types of memories, there is a necessary a special system.

i) I. Group Memories:

First group memories are Random Access Memory (RAM) and Read/Write (RJW). In these types memories if the energy is cut, the information is lost. If RAM is supplied program can be stored by battery that battery is in PLC device. When battery energy finishes, program will be erased.

ii) II. Group Memories:

It is Read Only Memory (ROM). The type memory can be erased and programmable. It is divided four into groups;

*1) **PROM (Programmable Read-Only Memory):*** it is a special type of ROM. PROM memory allows to writing of information in chip, these information are provided or there were at the beginning. The information can be written into ROM only one time.

The main disadvantage of PROM is no erasable and no Programmable. In PROM programming is doing as dissolve and pluck logic, for this reason, the erasing of erasable connections is process that there is no to turn back. For this reason, firstly all mistake control process must be finished.

2) ***EPROM (Erasable Programmable Read-Only Memory):*** this type is the memory type that is used in PLC devices. Written programmable firstly, is store in EPROM memory and is sent central processing unit.

3) ***EAROM (Electrically Alterable Read-Only Memory):*** It is like EPROM memory, but to erase and ultraviolet light supply is not necessary. EAROM chip to clean by erasing, an eraser voltage is exercised to suitable pin. When chip erases one time, it can be programmed again.

4) ***EEPROM (Electrically Erasable Programmable Read-Only Memory):*** In EEPROM memory type, when energy is cut, information cannot lose as EPROM. Special device is not necessary in writing and erasing processing. EEPROM or EPROM memories that are mounted to PLC make runs as stored program into records.

Data table stores information's, that are necessary to carry to the program, *which* includes information's such as output and input conditions, timers, and counter results and data records. Includes of table is divided two /groups as conditions data and numbers (or codes) 0 and 1 conditions are ON/OFF conditions of information that records the place of bit. Data table is divided 3 sections. Input view table stores the condition of digital input that relations input interface circuits. As ON/OFF condition, in this unit results of input are stored as zero (0) or one (1).

Output view memory is order of bits that control the digital condition of devices which links interface of output. The logic conditions of output units are stored in this memory and it is taken from this logic level memory and transfers to output unit.

1.3. PROGRAMMING DEVICES

The most important one of features of programmable controller is to have programming elements, which are useful. Programming device provides transformation between operator and circuit of controller. (Fig. 1.13)

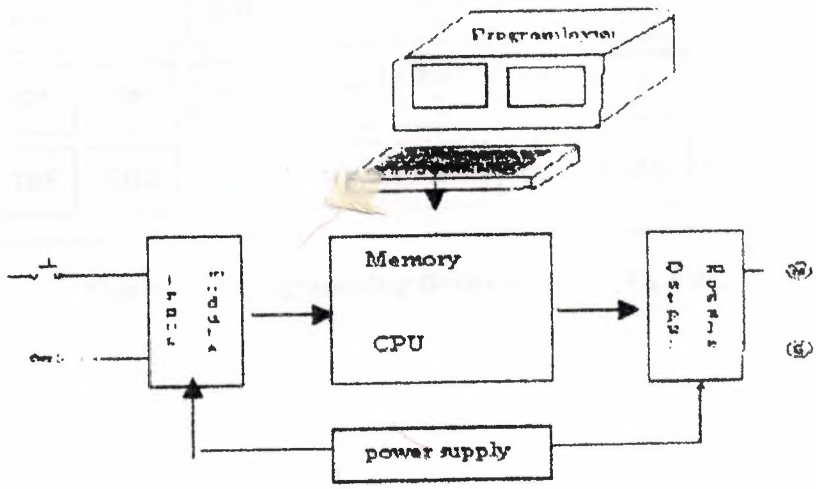


Figure 1.13 Transformation of PLC Circuits

Programming terminal relation between PLC memory and monitor. User sends programming device and PLC control program to device.

Generally, industrial CRT terminals in many devices are used for programmable controllers. These terminals include indicator units, keyboards and CPU and they provide to communicate necessary order. The advantage of CRT is to check program is easily on monitor.

In small PLCs programming is used cheap, moveable, small and mini programmable devices. The monitor of this type of programming monitor is liquid ~rvstal screen instead of CRT tube, which name LCD. On mini program there are LCD monitor program coding keys and special functions keys. FA2 of programming device DEC FA1 Junior module is shown at table 1.14

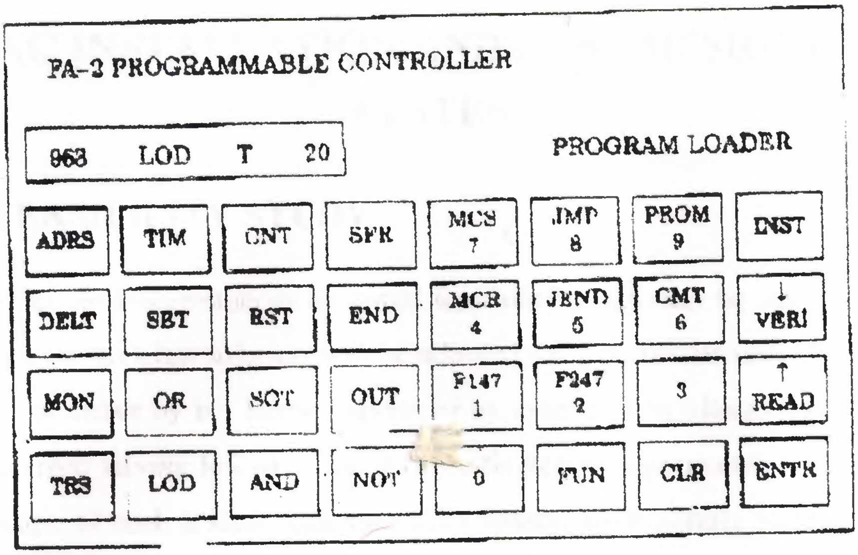


Figure 1.14 Programming Device of IDEC FA-1 PLC

CHAPTER TWO

CHOOSING INSTALLATION AND COMMISSIONING OF PLC SYSTEM

2.1 FEASIBILITY STUDY

Under certain circumstances an initial feasibility study may be suggested or warranted. prior to any decision on what solution will be adopted for a particular task. The feasibility study may be carried out either by in house experts or by external consultants. Often an independent specialist is preferred having few or no ties to specific vendor equipment.

The scope of such a study can vary enormously, from simply stating the feasibility of the proposal through to a comprehensive case analysis with complete equipment recommendations. Typically though, a feasibility study of this nature encompasses several specific areas of investigation:

(i) **economic feasibility,** consisting of the evaluation of possible installation and development costs weighed against the ultimate income or benefits resulting from a developed system;

(ii) **technical feasibility** . where the target process and equipment are studied in terms of function performance and constraints that may relate to achieving an acceptable system;

(iii) **alternatives** . with an investigation and evaluation of alternative approaches to the development of the acceptable system.

Area (i) economic feasibility and worth, can only be addressed fully once the result of areas (ii) and (iii) are available ,with estimated costing, and direct / indirect benefits being considered. Area (ii) is detailed in the following sections, with background information for area (i) usually being compiled through liaison with company personnel. The achievement of a complete technical proposal requires us to know what the present and future company needs are in terms of plant automation and desired information systems.

Once the control function has been accurately defined a suitable programmable control system has to be chosen from the wide range available. Following the identification of a suitable PLC work can begin on aspects of electrical hardware design and software design.

2.2 DESIGN PROCEDURE FOR PLC SYSTEM

Because the programmable controller is based on standard modules the majority of hardware and software design and implementation can be carried out independently of, rut concurrently with, each other.

Developing the hardware and software in parallel brings advantages both in terms of saving time and of maintaining the most flexible an adaptable position regarding the eventual system function. This allows changes in the actual control functions through software, until the final version is placed in the system memory and installed in the PLC. An extremely important aspect of every design project is the documentation.

Accurate and up - to - date documentation of all phases of a project need to be fully documented and updated as the job progresses through to completion. This information will form part of the total system documentation, and can often be invaluable during later stages of commissioning and troubleshooting.

i) Choosing a programmable controller

There is a massive range of PLC systems available today with new additions or replacement continually being produced with enhanced features of one type or another. Advances in technology are quickly adopted by manufacturers in order to improve the performance and market status of their products. However, irrespective of make the majority of PLC s in each size range are very similar in terms of their control facilities. Where significant differences are to be found is in the programming methods and languages together with differing standards of manufacturer support and backup. This latter point is often overlooked when choosing a suitable make of controller, but the value of good reliable manufacturers assistance cannot be overstated both for present and future control needs.

ii) Size and type of PLC system

This may be decided in conjunction with the choice of manufacturer, on the basis that more than one make of machine can satisfy a particular application, but with the ast choice of equipment now available the customer can usually obtain similar systems from several original equipment manufacturers (OEMs). Where the specification requires certain types of function or input / output it can result in one system from a single manufacturer standing out as far superior or cost effective than the competition but this is rarely the case. Once the stage of deciding actual size of the PLC system is reached there are several topics to be considered:

- necessary input / output capacity;
- types of I / 0 required;
- size of memory required;
- speed and power required of the CPU and instruction set.

All this topics are to a large extent interdependent with the memory size being directly tied to the amount of I / 0 as well as program size. As the I / 0 memory size rises this takes longer to process and requires a more powerful faster central processor if scan times are remain acceptable.

The I / 0 sections of a PLC system must be able to contain sufficient modules to connect all signal and control lines for the process. These modules must conform to the basic system specifications as regards voltage levels, loading, etc.,

- The number and type of I /0 points required per module;
- Isolation required between the controller and the target process;
- The need for high speed I / 0 or remote I / 0 or any other special facility;

iii) I / 0 requirements

The I / O sections of a PLC system must be able to contain sufficient modules to connect all signal and control lines for the process. These modules must conform to the basic system specifications as regards voltage levels, loading, etc.,

- The number and type of I / O points required Per module;
- Isolation required between the controller and the target process;
- The need for high speed I / O . or any other special facility;
- Future needs of the plant in terms of both expansion potential and installed spare I / O points.
- Power supply requirements of I / 0 points is an on board PSU needed :c~ drive any transducer or actuators?

In certain cases there may be a need for signal conditioning modules to be included in the system, with obvious space demands on the main or remote racks. When the system is to be installed over a wide area, the use of a remote or decentralized form of I / O working can give significant economies in cabling the sensors and actuators to the

iv) Memory and programming requirements

Depending on the type of programmable controller being considered, the system memory may be implemented on the same card as the CPU, or alternatively on dedicated cards. This ladder method is the more adaptable, allowing memory size to be increased as necessary up to the system maximum, without a reciprocal change in CPU card.

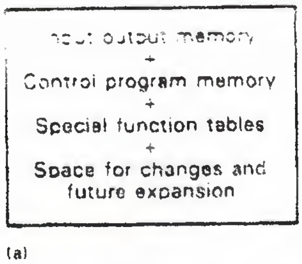
As stated in the previous section, memory size is normally related to the amount of o points required in the system. The other factor that affects the amount of memory required is of course the control program that is to be installed. The exact size of any program cannot be defined until of the software has been designed, encoded, installed and tested. However, it is possible to accurately estimate this size based on average program complexity. A control program with complex ,lengthy interlocking or sequencing routines obviously requires more memory than one for a simple process. Program size is also related to the number of 110 points since it must include instructions for reading from or writing to each point. Special functions are required for the control task may also require memory space in the unit PLC memory map to allow data transfer between cards. Finally additional space should be provided to allow for changes in the program, and for future expansion of the system.

There is often a choice of available memory type — RAM or EPROM. The RAM form is the most common, allowing straightforward and rapid program alterations both before and after the system is installed. RAM contents are made semipermanent by the provision of battery backing on their power supply. RAM must always be used for I / and data functions , as these involve dynamic data.

EPROM memory can be employed for program storage only, and requires the use of special EPROM eraser / programmer to alter the stored code. The use of EPROMS is ideal where several machines are controlled by identical programmable controllers inning the same.

However until a program has been a fully developed and tested RAM storage should be used.

As mentioned in earlier chapters, microcomputers are commonly used as program development stations. The large amounts of RAM and disk storage space provided in these machines allows the development and storage of many PLC programs ,including related text and documentation. Programs can be transferred between the microcomputer and the target PLC for testing and alteration. EPROM programming can so often be carried out via the microcomputer.



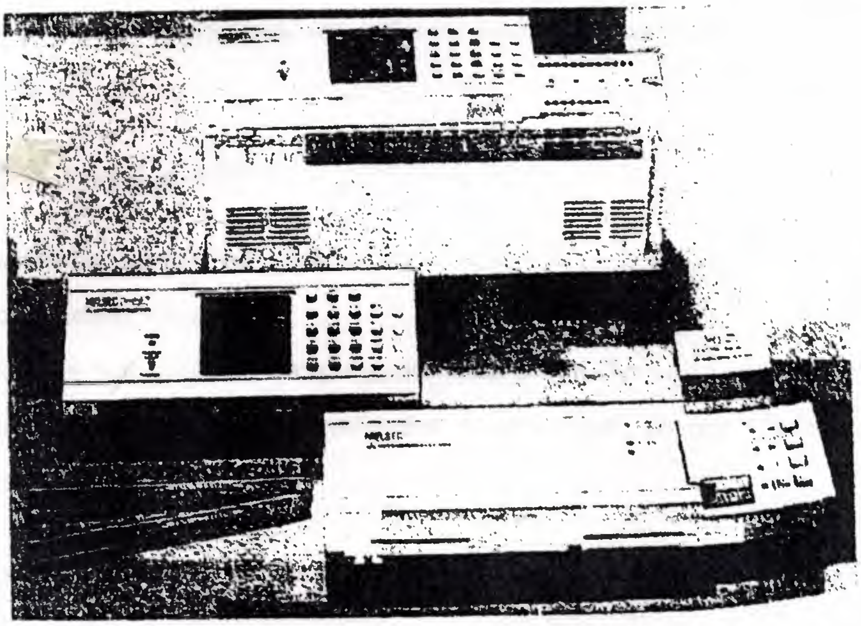


Figure 2.1 a) PLC memory requirements for different tasks.

b) Custom EPROM programmer for a Mitsubishi F series PLC

v) Instruction set / CPU

Whatever else is left undefined, any system to be considered must provide an instruction set that is adequate for the task. Regardless of size, all PLCs can handle logic control, sequencing, etc. Where differences start to emerge are in the areas of data handling, special functions and communications. Larger programmable controllers tend to have more powerful instructions than smaller ones in these areas, but careful scrutiny of small / medium machines can often reveal the capability to perform specific functions at surprisingly good levels of performance.

In modular programmable controllers there may be a choice of CPU card offering different levels of performance in terms of speed and functionality. As the number of I / 0 and function cards increases the demands on the CPU also increase, since there are greater numbers of signals to process each cycle. This may require the use of a faster CPU card if scan time is not to suffer.

Following the selection of the precise units that will make up the programmable controller for a particular application, the software and hardware design functions can be carried out independently.

2.3 INSTALLATION

The hardware installation consists of building up to necessary racks and cubicles, then installing and connecting the cabling.

The cabinet that contains the programmable controller and associated sub racks (see figure 2.2) must be adequate for the intended environment as regards security safety band protection from the elements:

Security in the form of a robust, lockable cabinet;

Safety by providing automatic cut off facilities / alarms if the cabinet door is opened;

Protection from humid or corrosive atmospheres by installation of airtight seals on the cubicle. Further electrostatic shielding by earthling the cubicle body.

For maintenance purposes there must be easy access to the PLC racks for card inspection changing etc. Main on / off and status indicators can be built in to the cabinet doors and glass or Perspex windows fitted to allow visual checking of card status or relay / contactor operation.

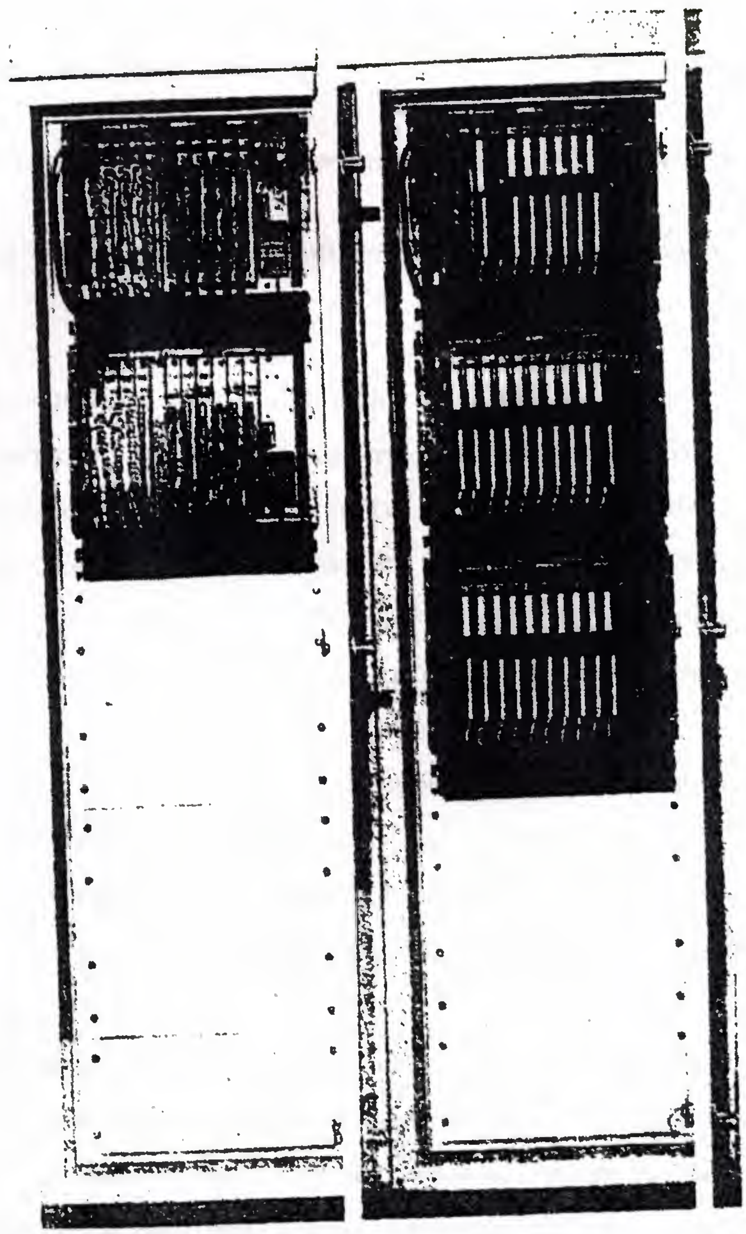


Figure 2.2 Complete PLC installation and cabined

2.4 TESTING AND COMMISSIONING

Once the installation work is completed the next step is to consider the testing and commissioning of the PLC system.

Commissioning comprises two basic stages:

1- Checking the cable connections between the PLC and the plant to be controlled.

2- Installing the completed control software and testing its operation on the target process.

The system interconnections must be thoroughly checked out to ensure all input / output devices are wired to the correct I /0 points. In a conventional control system this would be done by buzzing out the connections with suitable continuity test instruments. With a programmable, however, the programming panel may be used to monitor the status of inputs points directly this is long before the control software is installed, which will only be done after all hardware testing is satisfactorily completed. Before any hardware testing is started, a thorough test of all mains voltages, earthing, etc., must be carried out.

With the programmer attached to the PLC, input points are monitored as the related transducer is operated checking that the correct signal is received by the PLC. The same technique is used to test the various function cards installed in the system. For example analog inputs can be checked by altering the analog signal and observing a corresponding change in the data stored in the memory table.

In turn, the output devices can be forced by instructions from the programming panel checking their connection and operation. The commissioning team must ensure that any operation or misoperation of plant actuators will not result in damage to plant or personnel. Testing of some PLC functions at this stage is not always practical, such as for PD loops and certain communications channel. These require a significant amount of configuring by software before they can be operated, and are preferably tested once the control software has been installed. Some programmable controllers contain in built diagnostic routines that can be used to check out the installed cards, giving error codes on a VDU or integral display screen. These diagnostic are run by commands from the programming panel or from within a control program once the system is fully operational.

i) Software testing and simulation

The preceding sections have outlined the various stages in hardware design and implementation. Over the same period of time the software to control the target process developed. in parallel for the chosen PLC system. These program modules is should be tested and proved individually wherever possible before being linked together to make up the complete applications program. It is highly desirable that any faults or error be removed before the program is installed in the host controller.

The time required to rectit7y faults can be more than doubled once the software is running in the host PLC.

Virtually all programmable controllers, irrespective of size, contain elementary software checking facilities. Typically these can scan through an installed program to check for incorrect labels. Double output coils etc. Listings of all I / 0 points used, counter / timer settings and other information is also provided. The resulting information is available on the programmer screen or as a printout in the figure 2.3 However, this form of testing is only of limited value, since there is no facility to check the operation of the resident program.

In terms of time and cost economies an ideal method for testing program modules is to reproduce the control cycle by simulation, since this activity can be carried out in the design workshop without having the actually connect up to the physical process. Simulation of the process is done in a number of ways, depending on the size of process involved.

When the system is relatively small with only a handful of I / 0 channels, it is often possible to adequately simulate the process by using sets of switches connected up to the PLC as inputs, with outputs represented by connecting arrays of small lambs or relays in the figure 2.4. This allows inputs to be offered to a test - bed controller containing software under test, checking the action of the control program by noting the operation and sequence of the output lambs or relays. By operating the input switches in spcific sequences, it is possible to test sequence routines within a program. Where fast response times are involved ,the tester should use the programming panel to force larger time intervals into the timers concerned, allowing that part of the circuit to be tested by the manual switch method.

Most I / 0 modules have LED indicators that show the status of the channels. These an be used instead of additional test actuators where digital outputs are concerned. Analog inputs can be simulated in part by using potential dividers suitably connected to the input channel and corresponding analog outputs connected either to variable devices such as small motors or to a moving coil meter configured to measure voltage r current. Standard sets of input switches and output actuators are normally available from PLC manufacturers.

When the system is larger with input / output channels and longer, more complex programs, the simple form of simulation described above becomes inadequate. Many larger PLC systems are fitted an integral simulation unit that reads and writes information directly into the I / O memory removing the need to connect external switches, etc. The simulator is controlled from an associated terminal which can force changes in input status and record all changes in output status as the program runs, for later scrutiny by the test team.

The program monitoring facility provided with most programming terminals should be used in virtually all these proceedings since it allows the dynamic checking of all elements in the program including preset and remaining values as the program cycles. In the figure 2.5 illustrates a monitoring display with status information shown on the bottom of the screen.

It is important to realize that the display on the programmer does not up date as rapidly as the control program is executing, due to the delays in transmitting the data across to the terminal. Contacts and other elements that are operated for only a few scans are unlikely to affect the display , but since a human observer could not detect this fast a change, this is not a significant disadvantage. To display all changes , the PLC should be run in single step mode. The monitor display shows a select portion of the ladder program , using standard symbols to depict contacts , output and present functions. All elements within the display are dynamically monitored , indicating their status as shown in the figure 2.6

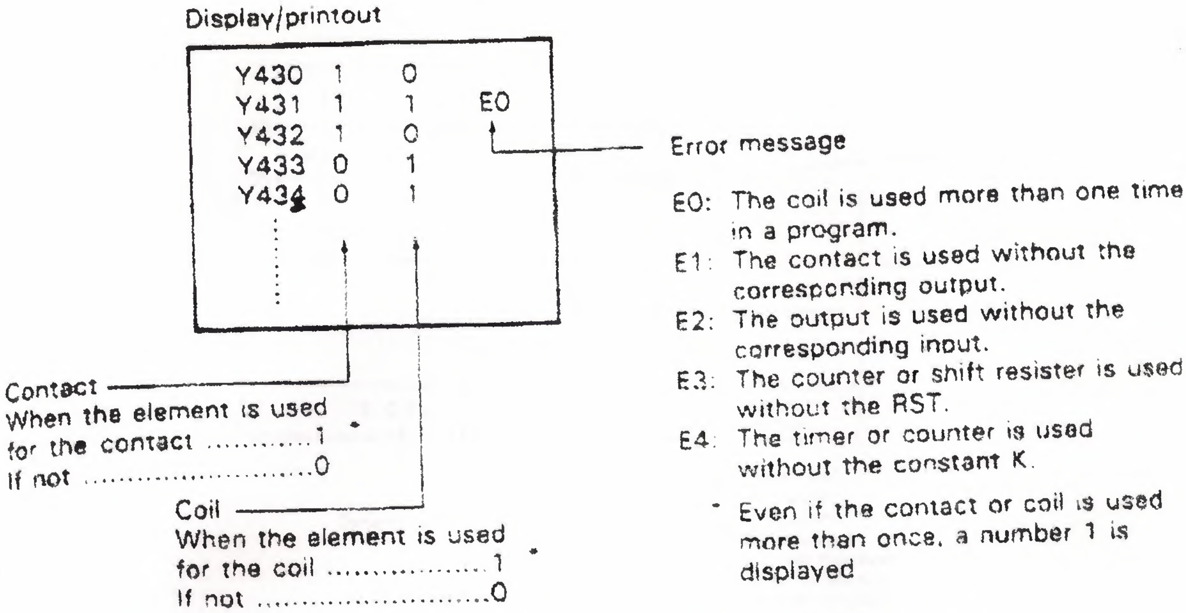


Figure 2.3 PLC printout of I / O static diagnostics information

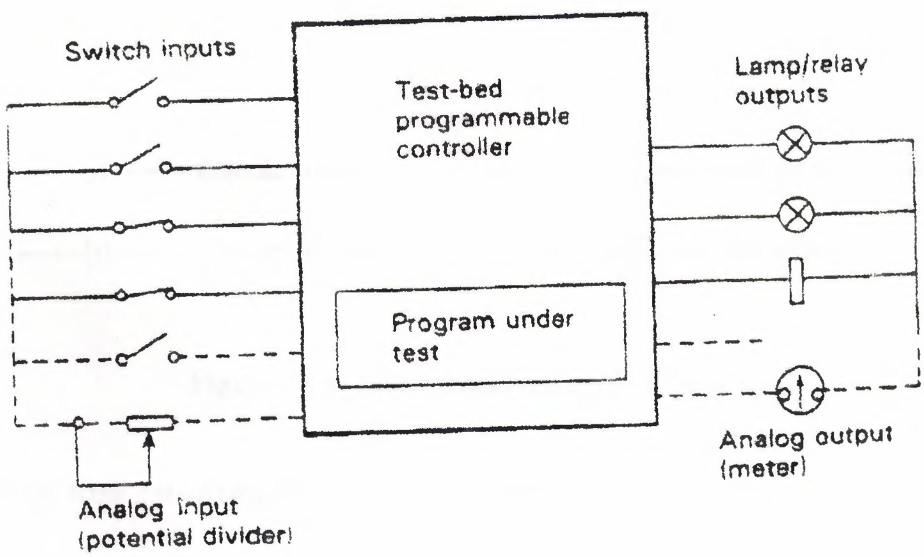


Figure 2.4 Process simulation using switches and lambs

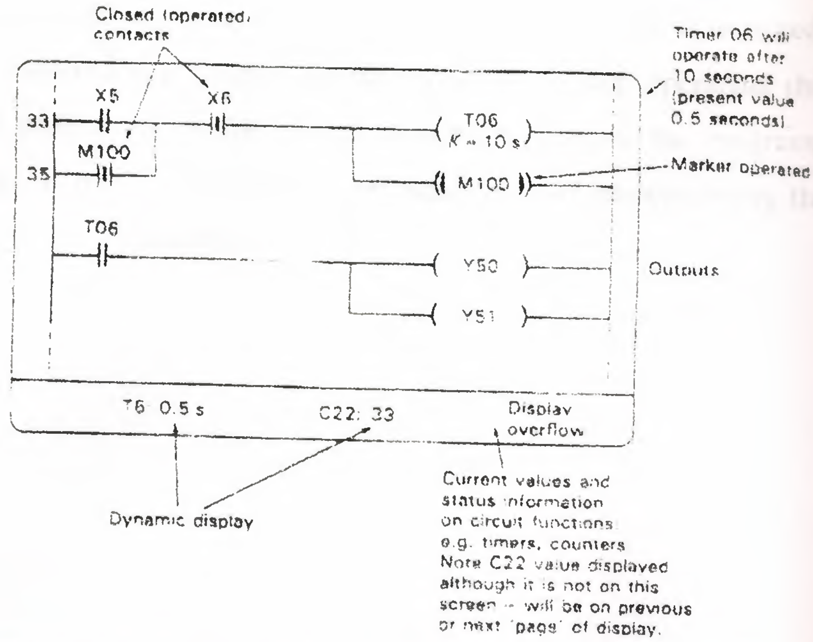


Figure 2.5 Dynamic monitoring of program contacts using a graphic programming display

Figure 2.6 Symbols displayed in monitor mode

ii) Installing and running the user control program

Once control software has been proved as far as possible by the above methods on a test machine, the next step is to try out the program on the tested PLC hardware installation. Ideally each section of code should be downloaded and tested individually, allowing faults to be quickly localized if the plant misoperates during program test. If this subdivided testing is not possible, another method is to include JUMP commands in the complete program to miss out all instructions except those in section to be tested. As each section is proved , the program is amended to place the JUMP instructions so as to select the next section to be tested.

Where a programmable controller supports single — step operation, this can be used examine individual program steps for correct sequencing. Again, the programming terminal should be utilized to monitor I / O status or any other area of interest during these tests, with continuous printouts if this is possible.

CHAPTER THREE
PLC PROGRAMMING SOFTWARE

In this section, PLC programming fundamental is prepared, student's capacity, which met PLC programming, is considered first time.

AND
OR
NOT
NAND
NOR
SET
RESET

Furthermore there are many specialization's such as TIMER, COUNTER, and MASTER CONTROL SET (MCS), which works data and controls PROGRAM, MASTER CONTROL RESET (MCR), JMP. There command which are mathematics process that are comparator processes (=, <, >).

In all PLC systems, to create logic process is programmed as the same are carried out some function. However, the main logic is the same that TIMER, COUNTER and SHIFT REGISTER functions are to get command and programmed but there can be some differences.

3.1. CREATE OF LEADER DIAGRAM

i) Start Commands:

These commands are first element of program. There are two type contact conditions as at table 2.7. First normally is open also second close. Normally, starting with open contact this program command is to get command as LD iN, LD, LOD A, on PLC device. And also close contact is stated as LDI, LD NOT, LOD NOT, AN.

LADDER SYMBOL	COMMAND LINE					
	IDEC	FESTO	AEG	Mitsubishi	Siemens	OMRON
F Normally open contact	LOD F	LD FLAG F LD IN F	UF	LD F	A F	LD F
F Normally close contact	LOD NOT F	LD NOT FLAG F LD NOT IN F	UN F	LDI F	AN F	LD NOT F

Table 3.1 Load Exercising

Note: in table F value is constant and input/output interval relay, special relay, timer, counter can be SFR number. According to this table at MITSUBISHI and HITACHI model normally open contact is shown with LD, also close contact is shown with LDI. Also at AEG PLC, U (UND) command is used for open contact and (UN) IJND-NICHT command is used for closed contact. Also at SIEMENS PLC, A (AND) command is used to open contact and AN (AND-NOT) is used for closed contact. At OMRON PLC, open contact is shown LD, also close contact is shown with LD NOT. Also at FESTO PLC, open contact LD FLAG is used for flag load other conditions LD IN command is used to contact load. In normally, also close contact is programmed for flag exercising as LD NOT FLAG... For other contacts are programmed as LD NOT iN...

ii) AND and OR Exercising:

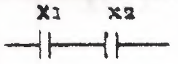
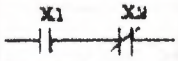
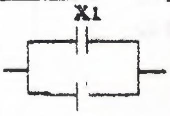
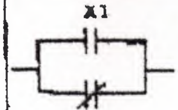
LADDER SYMBOL	COMMAND LINE						
	IDEC	FESTO	AEG	MITSUBISHI	Siemens SIMATIC	OMRON	HITACHI
X1 X2 	LOD X1 AND X2	LD IN X1 AND IN X2	U X1 U X2	LD X1 AND X2	A X1 A X2	LD X1 AND X2	LD X1 AND X2
X1 X2 	LOD X1 AND NOT X2	LD IN X1 AND NOT IN X2	U X1 UN X2	LD X1 ANI X2	A X1 AN X2	LD X1 AND NOT X2	LD X1 ANI X2
X1  X2	LOD X1 OR X2	LD IN X1 OR X2	U X1 O X2	LD X1 OR X2	A X1 O X2	LD X1 OR X2	LD X1 OR X2
X1  X2	LOD X1 OR NOT X2	LD IN X1 OR NOT X2	U X1 ON X2	LD X1 ORI X2	A X1 ON X2	LD X1 OR NOT X2	LD X1 ORI X2

Table 3.2: Symbol and command line AND and OR exercises.

iii) Output Stored Exercises:

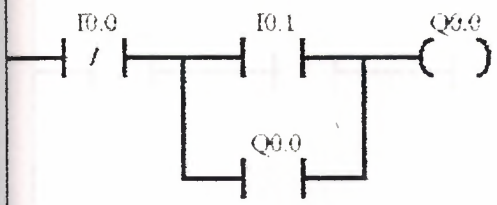
At a PLC system relay, it is used as output function, can be divided into two groups. First group output which charge can be linked to it according to program as (solenoid valves, neon lamb, conductor, led, etc.) are real output. Also second group outputs are internal and image relays. Physical connection cannot link to these relays but outputs of these sensors are transferred to real output and output can be taken.

If commands will be observed, there are similarities between PLC devices that output program commands are different. At both output and input functions, Xl, X2, are used as addresses.

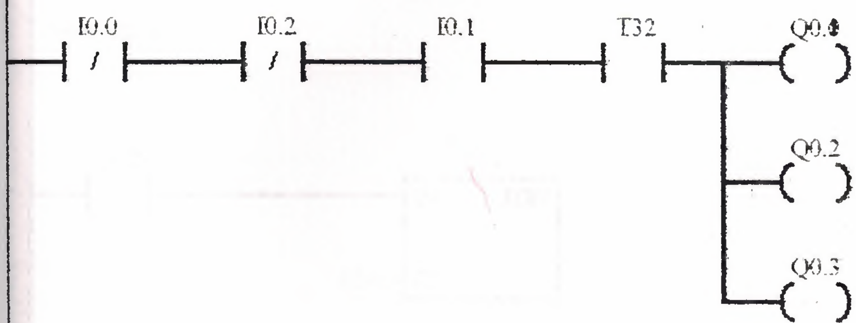
LADDER SYMBOL	COMMAND LINE						
	IDEC	FESTO	AEG	MITSUBISHI	Siemens Simatic	OMRON	HITACHI
 X1 X2	LOD X1 OUT X2	LD IN X1 = OUT X2	U X1 = X2	LD X1 OUT X2	A X1 = X2	LD X1 OUT X2	LD X1 OUT X2
 X1 X2 SET	LOD X1 SET X2	LD IN X1 SE FLAG X2  X1 SF1	U X1 SL X2	LD X1 S X2	A X1 S X2	LD X1 SET X2	LD X1 SET X2

Table 3.3

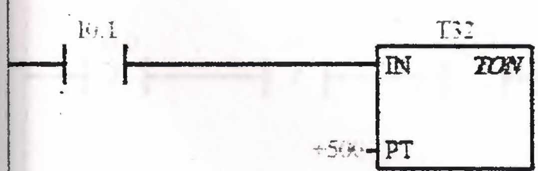
Network 1 sensor 1 considered and rail started



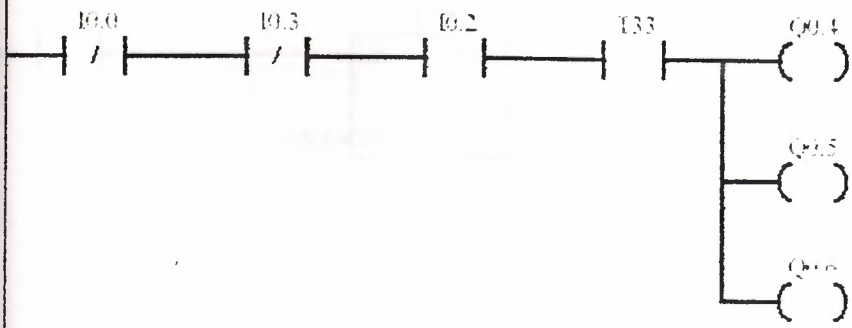
Network 2 sensor 1 considered and, front washing motors, flank brushes and upper brush worked during 50s



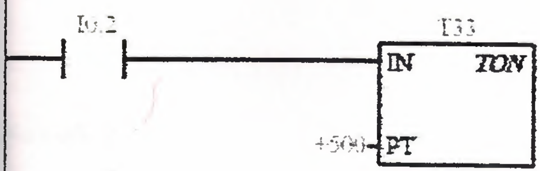
Network 3



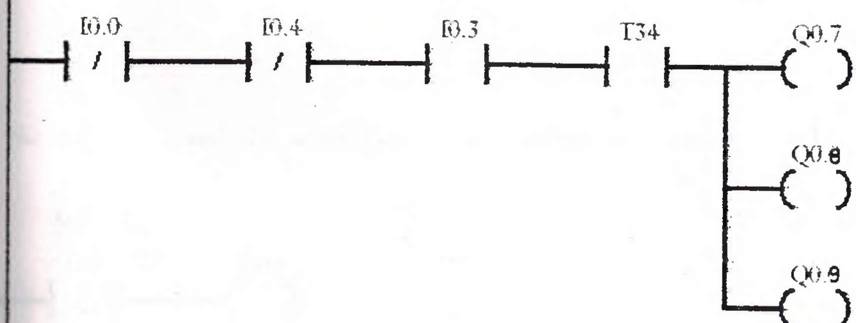
Network 4 sensor 2 considered to car after sensor 1 stopped and shampoo, upeer brush,flank brushes started during 50s



Network 5



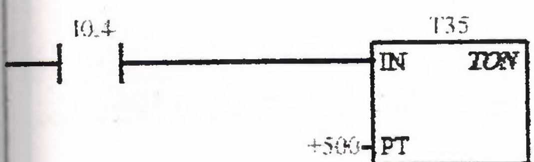
Network 6 after sensor 3 considered to car and shampoo stopped and after rinse, flank brushes, upeer brush worked during 50s



Network 7

Network 8 after sensor 4 considered, rinse stopped and drying started during 50s

Network 9



Network 10 and, after sensor 5 considered to car and, rail, washing stopped

Network 11

(END)

```
NETWORK 1     //sensor 1 considered and rail started

LDN     I0.0
LD      I0.1
O       Q0.0
ALD
=       Q0.0

NETWORK 2     //sensor 1 considered and, front washing motors, flank brushes and
 upper brush worked during 50s
LDN     I0.0
AN      I0.2
A       I0.1
A       T32
=       Q0.0
=       Q0.2
=       Q0.3

NETWORK 3
LD      I0.1
TON     T32, +500

NETWORK 4     //sensor 2 considered to car after sensor 1 stopped and shampoo,
 upper brush,flank brushes started during 50s
LDN     I0.0
AN      I0.3
A       I0.2
A       T33
=       Q0.4
=       Q0.5
=       Q0.6

NETWORK 5
LD      I0.2
TON     T33, +500

NETWORK 6     //after sensor 3 considered to car and shampoo stopped and after
 rinse, flank brushes, upeer brush worked during 50s
LDN     I0.0
AN      I0.4
A       I0.3
A       T34
=       Q0.7
=       Q0.0
=       Q0.0

NETWORK 7
LD      I0.3
TON     T34, +500

NETWORK 8     //after sensor 4 considered, rinse stopped and drying started
 during 50s
LDN     I0.0
A       I0.4
A       T35
=       Q0.0

NETWORK 9
LD      I0.4
TON     T35, +500

NETWORK 10    //and, after sensor 5 considered to car and, rail, washing stopped
LDN     I0.0
=       Q0.0

NETWORK 11
MEND
```

SYMBOL OF KEYS

I 0.0	Button (Emergency Button and Stop Button)
I 0.1	Sensor 1 Contac
I 0.2	Sensor 2 Contac
I 0.3	Sensor 3 Contac
I 0.4	Sensor 4 Contac
Q 0.0	Rail Contac (for contac)
T 32	Timer Contac
T 33	Timer Contac
T 34	Timer Contac
T 35	Timer Contac
Q 0.0	Rail
Q 0.1	Front Washing Motors
Q 0.2	Brush (Upper)
Q 0.3	Brush (Flank)
Q 0.4	Shampoo
Q 0.5	Brush (Upper)
Q 0.6	Brush (Flank)
Q 0.7	Rinse
Q 0.8	Brush (Upper)
Q 0.9	Brush (Flank)
Q 0.10	Drying

CONCLUSION

When developing this project we see that PLC the individual's life easier, which it has gained our interest and notice.

With the information observed from our lecturer and our researchers for this topic PLC, is a convenient tool with a wide rage of useful ways to be used. Such examples can be mentioned several machines can be used at the same time, easy adjustments from the PLC program can be meet within a few minutes by the keyboard, installed PLL programs can be controlled or checked before within the office and laboratory, even the PLC programs for firm can be meet at home. ft is very protective and safe for the workers which they me protected from dagger, communication programs of PLCs within each other or within operates can happen with the PLC; the developed lantues have constructed the productivity, security, establishment security fast productivity, quality and we can see that PLC is a very cheap program that can be fundamentally used.

This PLC devices is automatic car wash system with PLC. Automation is achieved in this machine and this shows, that PLC is a very important device to control complicated processes.

REFERENCES

[1] D.V.Hall, Microprocessors and İnterfacing Programming and Hardware, Mc Graw Hill., 1986.

[2] Yohanson G., Programmable Controllers, Satt Control, 1985.

[3] Simatic S7 – 200 and Industrial Automation, Doc. Dr Salman Kurtulan, ITU Electrical& Electronic Department, 1998.

[4] PLC, Richard Baldry, November, 1999.

[5] Programmable Logic Controllers, Hugh Jack, June 1999.