NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

WATER LEVEL CONTROLLING BY USING PLC

Graduating Project
EE - 400

Student: Mesut BULUT

Supervisor : Assit.Prof.Dr. Özgür C. ÖZERDEM

Nicosia 2006

ACKNOWLEDGEMENTS

It is my pleasure to thank to following people in the ***Near East University*** and My Family Without them, I could never finish my undergraduate program and my Graduation Project would not have been successfully completed on time.

First of all, I would like to thanks to my supervisor ***Assist. Prof. Dr Özgür C. Özerdem*** for supervising my project. Under the guidance of him I successfully overcome many diffficulties and I learned a lot about electric and electronic. In each discussion, he used to explain the problems and answer my questions. He always helped me a lot and I felt remarkable progress during his supervision. I also thank **Prof. Dr.Sezai DİNÇER,** ***Assist. Prof. Dr. Kadri Bürüncük, Mr. Mustafa GÜNDÜZ , Assoc. Prof. Adnan Kashman,*** **Mrs. Filiz Alshanable** that I spent funny times during their lecture times.

Then I also thank to my friends: **Serbulent Karagoz, Mehmet Akca, Aylin Duran** and **Samet Biricik** because they also motivated me when I was preparing My project

Finally and specialally thanks for my family, especially my parents and my elder brother **Mustafa Bulut** for being patientful during my undergraduate degree study. I could never have completed my study without their encouragement and endless support.

ABSTRACT

Water level Controlling is an example that we can solve so many logical problems by using PLC's. Because PLC is a machine that you can get any output on depending on your input conditions, so it can solve so many problem in industry easier than relays. As PLC technology has advanced, so have programming languages and communications capabilities, along with many other important features. Today's PLCs offer faster scan times, space efficient high-density input/output systems, and special interfaces to allow non-traditional devices to be attached directly to the PLC. Not only can they communicate with other control systems, they can also perform reporting functions and diagnose their own failures, as well as the failure of a machine or process. Size is typically used to categorize today's PLC, and is often an indication of the features and types of applications it will accommodate.

In this project CPU 212 8 inputs, 6 outputs, 24 volt DC input 220 Volt AC output PLC is used, and an automation of a Lamp and an industrial motor is realized by PLC programs.

TABLE OF CONTENTS

4 .WATER LEVEL CONTROLLING

1.1 INTRODUCTION

Control Engineering has been evolved day-by-day time. In the past people were the main methods for controlling everything. More recently electricity has been used for control and early electrical control was based on relays. These relays allow power to be switched on and off without a mechanical switch. It is common to use relays to make simple logical control decisions. The development of low cost computer has brought the most recent revolution, the Programmable Logic Controller (PLC).

1.2 WHAT IS A PLC ?

PLC stands for Programable Logic Controllers that PLCs are often defined as miniature industrial computers that contain hardware and software that is used to perform control functions. 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.

A PLC consists of two basic sections: the central processing unit (CPU) and the input/output interface system. The CPU, which controls all PLC activity, can further be broken down into the processor and memory system. The input/output system is physically connected to field devices (e.g., switches, sensors, etc.) and provides the interface between the CPU and the information providers (inputs) and controllable devices (outputs). To operate, the CPU "reads" input data from connected field devices through the use of its input interfaces, and then "executes", or performs the control program that has been stored in its memory system. Programs are typically created in ladder logic, a language that closely resembles a relay-based wiring schematic, and are entered into the CPU's memory prior to operation. Finally, based on the program, the PLC "writes", or updates output devices via the output interfaces. This process, also known as scanning, continues in the same sequence without interruption, and changes only when a change is made to the control program

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 world applications. If there is industry present, changes are good that there is a PLC present. If you are involved in machining, packaging material handling, and 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 PLC.

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

As you see, if the process becomes more complicated, then we have to use a device the simplify that. We use PLC for this process. We can program the PLC to count its inputs and turn the solenoids for the specified time.

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

1.3 HISTORY OF PLC

In the 1960's PLC's 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 (MODICON) to a major US car manufacturer. Other companies at the time proposed computer based upon the PDP – 8. The MODICON 084 brought the world's first PLC into commercial production.

The first PLC can be traced back to 1968 when Bedford Associates, a company in Bedford, MA, developed a device called a Modular Digital Controller for General Motors (GM). The MODICON, as it was known, was developed to help GM eliminate traditional relay-based machine control systems. Because relays are mechanical devices, they have limited lifetimes. They are also cumbersome, especially in large applications where thousands of them may exist. With so many relays to work with, wiring and troubleshooting could be quite complicated. Since the MODICON was an electronic device, not a mechanical one, it was perfect for GM's requirements, as well as for many other manufacturers and users of control equipment. With less wiring, simpler troubleshooting, and easy programming, PLC technology caught on quickly. In the late 1960's PLC's 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 (MODICON) to a major US car manufacturer. Other companies at the time proposed computer 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 that 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, possible 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. The 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 techniques were sequencer state machines and the bit-slice based CPU. The AMD 2901 and 2903 were quite popular in MODICON and A-B PLC's. Conventional microprocessors lacked the power to quickly solve PLC logic in all but the smallest PLC's. As conventional microprocessors evolved, larger and larger PLC's were being based upon them. However, even today some are still based upon the 2903. MODICON has yet the build a faster PLC then their 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 PLC's 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.

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

In The 90's have seen a gradual reduction in the introduction new protocols, and the modernization of the physical layers of same of the more popular protocols that survived the 1980's. The latest standard has tried to merge PLC- programming languages less than one international standard.

1.4 COMPAREING PLC WITH RELAY CONTROLLING

Modern control systems still include relays, but these are rarely used for logic. A relay is a simple device that uses a magnetic field to control a switch, as pictured in Figure 1. When a voltage is applied to the input coil, the resulting current creates a magnetic field. The magnetic field pulls a metal switch (or reed) towards it and the contacts touch, closing the switch. The contact that closes when the coil is energized is called normally open. The normally closed contacts touch when the input coil is not energized. Relays are normally drawn in schematic form using a circle to represent the input coil. The output contacts are shown with two parallel lines. Normally open contacts are shown as two lines, and will be open (non-conducting) when the input is not energized. Normally closed contacts are shown with two lines with a diagonal line through them. When the input coil is not energized the normally closed contacts will be closed (conducting).

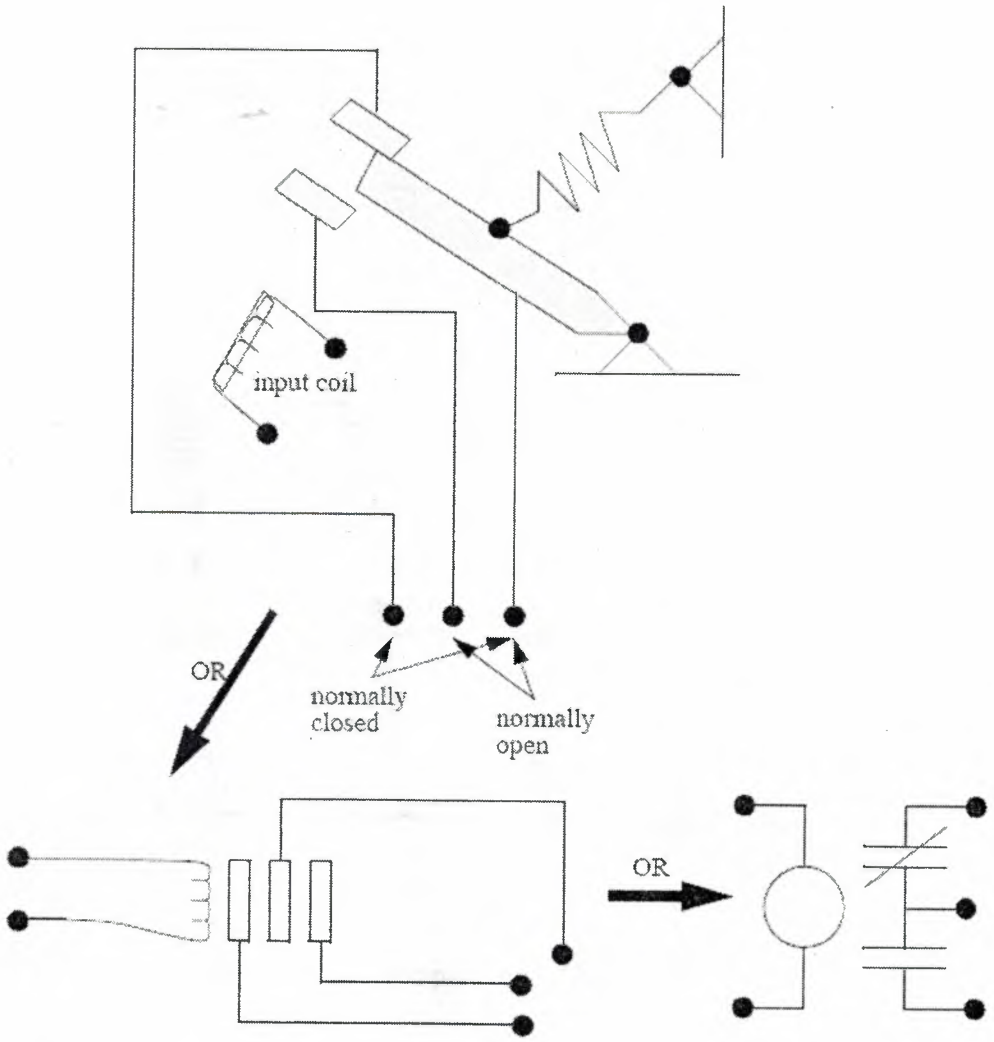


Figure : 1 Simple Relay Layouts and Schematics

Relays are used to let one power source close a switch for another (often high current) power source, while keeping them isolated. An example of a relay in a simple control application is shown in Figure 2. In this system the first relay on the left is used as normally closed, and will allow current to flow until a voltage is applied to the input A. The second relay is normally open and will not allow current to flow until a voltage is applied to the input B. If current is flowing through the first two relays then current will flow through the coil in the third relay, and close the switch for output C. This circuit would normally be drawn in the ladder logic form. This can be read logically as C will be on if A is off and B is on.

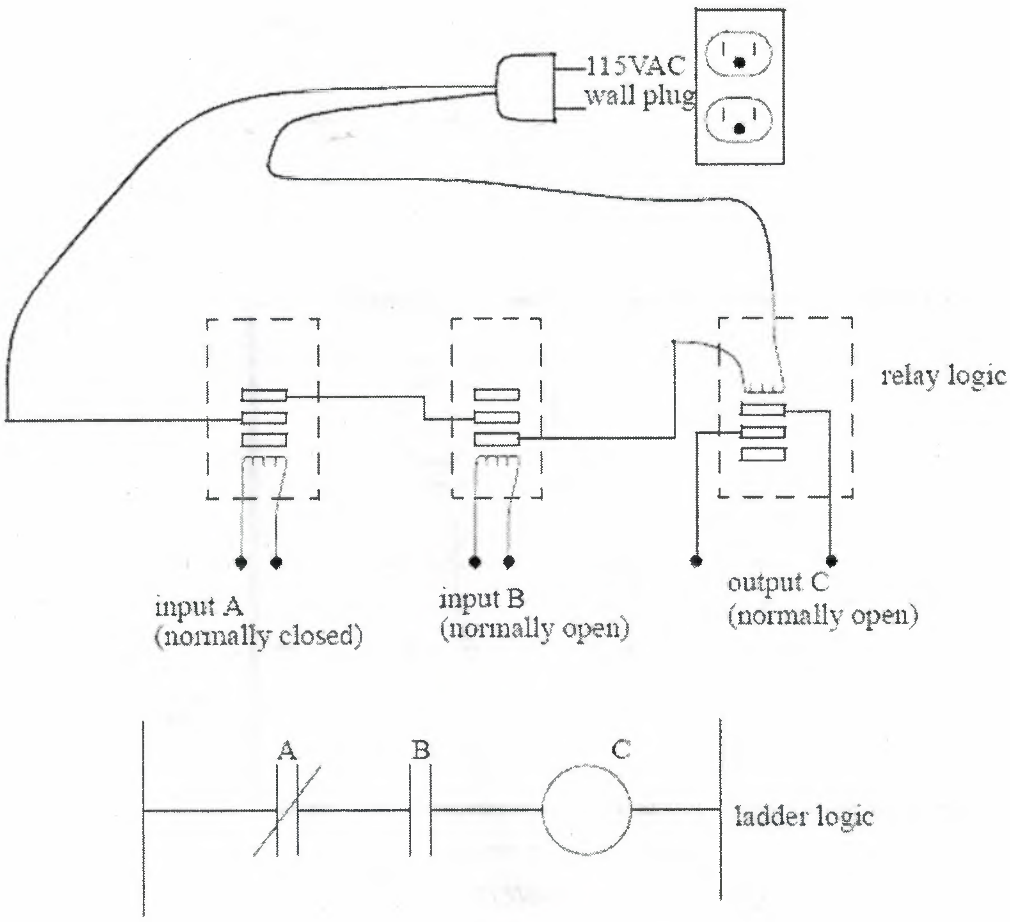


Figure : 2 A Simple Relay Controller

The example in Figure 2 does not show the entire control system, but only the logic. When we consider a PLC there are inputs, outputs, and the logic. Figure 3 shows a more complete representation of the PLC. Here there are two inputs from push buttons. We can imagine the inputs as activating 24V DC relay coils in the PLC. This in turn drives an output relay that switches 115V AC, that will turn on a light. Note, in actual PLCs inputs are never relays, but outputs are often relays. The ladder logic in the PLC is actually a computer program that the user can enter and change. Notice that both of the input push buttons are normally open, but the ladder logic inside the PLC has one normally open contact, and one normally closed contact. Do not think that the ladder logic in the PLC needs to match the inputs or outputs. Many beginners will get caught trying to make the ladder logic match the input types.

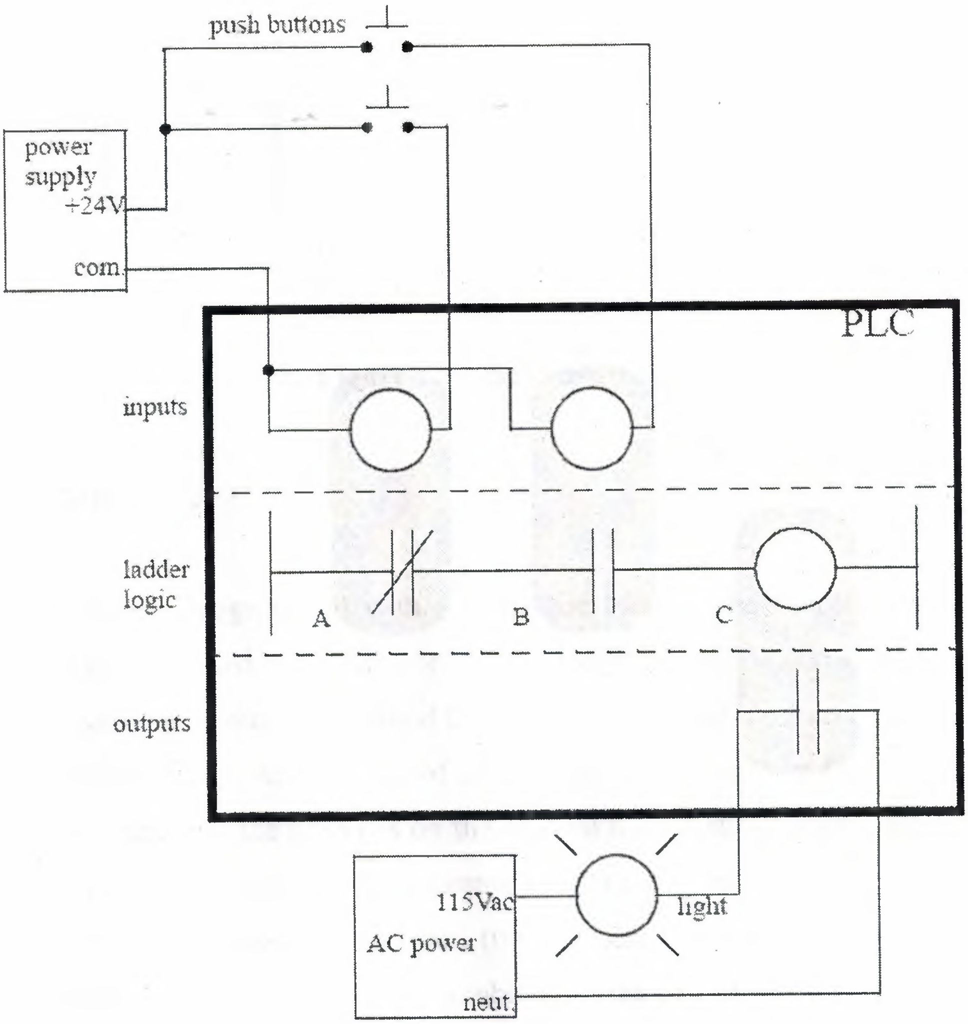


Figure : 3 A PLC Illustrated With Relays

Many relays also have multiple outputs (throws) and this allows an output relay to also be an input simultaneously. The circuit shown in Figure 4 is an example of this, it is called a seal in circuit. In this circuit the current can flow through either branch of the circuit, through the contacts labelled A or B. The input B will only be on when the output B is on. If B is off, and A is energized, then B will turn on. If B turns on then the input B will turn on, and keep output B on even if input A goes off. After B is turned on the output B will not turn off.

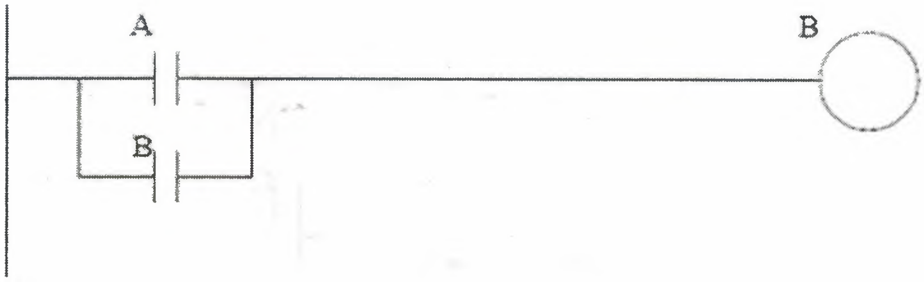


Figure : 4 A Seal-in Circuit

1.4.1 Programming

The first PLCs were programmed with a technique that was based on relay logic wiring schematics. This eliminated the need to teach the electricians, technicians and engineers how to *program* a computer - but, this method has stuck and it is the most common technique for programming PLCs today. An example of ladder logic can be seen in Figure 5. To interpret this diagram imagine that the power is on the vertical line on the left hand side, we call this the hot rail. On the right hand side is the neutral rail. In the figure there are two rungs, and on each rung there are combinations of inputs (two vertical lines) and outputs (circles). If the inputs are opened or closed in the right combination the power can flow from the hot rail, through the inputs, to power the outputs, and finally to the neutral rail. An input can come from a sensor, switch, or any other type of sensor. An output will be some device outside the PLC that is switched on or off, such as lights or motors. In the top rung the contacts are normally open and normally closed. Which means if input *A* is on and input *B* is off, then power will flow through the output and activate it. Any other combination of input values will result in the output *X* being off.

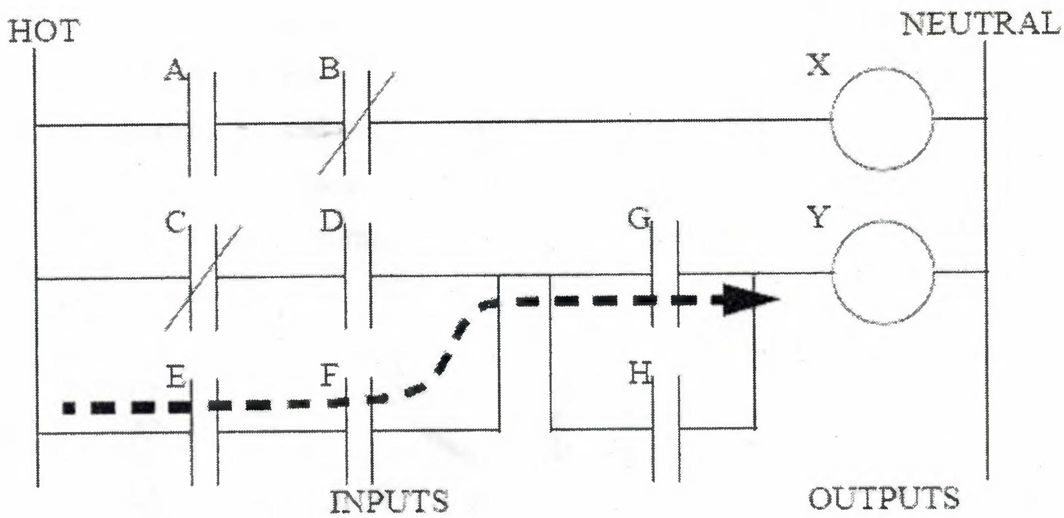


Figure : 5 A Simple Ladder Logic Diagram

The second rung of Figure 5 is more complex, there are actually multiple combinations of inputs that will result in the output *Y* turning on. On the left most part of the rung, power could flow through the top if *C* is off and *D* is on. Power could also (and simultaneously) flow through the bottom if both *E* and *F* are true. This would get power half way across the rung, and then if *G* or *H* is true the power will be delivered to output *Y*. In later chapters we will examine how to interpret and construct these diagrams.

There are other methods for programming PLCs. One of the earliest techniques involved mnemonic instructions. These instructions can be derived directly from the ladder logic diagrams and entered into the PLC through a simple programming terminal. An example of mnemonics is shown in Figure 6. In this example the instructions are read one line at a time from top to bottom. The first line *00000* has the instruction *LDN* (input load and not) for input *00001*. This will examine the input to the PLC and if it is off it will remember a *1* (or true), if it is on it will remember a *0* (or false). The next line uses an *LD* (input load) statement to look at the input. If the input is off it remembers a *0*, if the input is on it remembers a *1* (note: this is the reverse of the *LD*). The *AND* statement recalls the last two numbers remembered and if the are both true the result is a *1*, otherwise the result is a *0*. This result now replaces the two numbers that were recalled, and there is only one number remembered. The process is repeated for lines *00003* and *00004*, but when these are done there are now three numbers remembered. The oldest number is from the *AND*, the newer numbers are from the two *LD* instructions. The *AND* in line *00005* combines the results from the last *LD* instructions and now there are two numbers remembered. The *OR* instruction takes the two numbers now remaining and if either one is a *1* the result is a *1*, otherwise the result is a *0*. This result replaces the two numbers, and there is now a single number there. The last instruction is the *ST* (store output) that will look at the last value stored and if it is *1*, the output will be turned on, if it is *0* the output will be turned off.

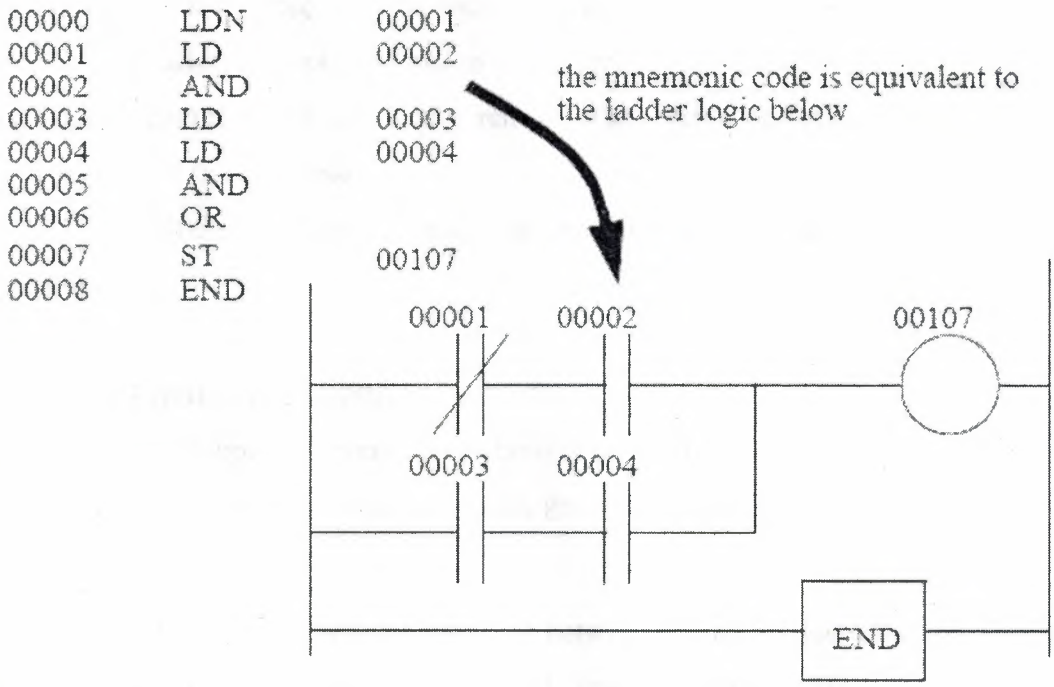


Figure : 6 An Example of a Mnemonic Program and Equivalent Ladder Logic

The ladder logic program in Figure 6, is equivalent to the mnemonic program. Even if you have programmed a PLC with ladder logic, it will be converted to mnemonic form before being used by the PLC. In the past mnemonic programming was the most common, but now it is uncommon for users to even see mnemonic programs.

1.5 PLC Hardware Design

Programmable controllers are purpose-built computers consisting of three functional areas:

- Processing:
- Memory:
- Input / output:

Input conditions to the PLC are sensed and than stored in the memory, where the PLC performs the programmed logic instructions on these input states. Output conditions are then generated to drive associated equipment. The action taken depends totally on the control program held in memory.

In smaller PLC these functions are performed by individual printed circuit cards within a single compact unit, whilst larger PLC's are constructed on a modular basis with function modules slotted in to the back plane connectors of the mounting rack.

This allows simple expansion of the system when necessary. In both these cases the individual circuit board are easily removed and replaced, facilitating rapid repair of the system should faults develop.

In addition a programming unit is necessary to download control programs to the PLC memory.

a) Input output / units

Most PLC'S operate internally at between 5 and 15 V d.c. (Common TTL and CMOS voltages), whilst process signals much greater, typically 24 V d.c. To 240 V a.c. at several amperes.

The I / O units form the interface between the microelectronics of the programmable controller and real world outside, and must therefore provide all, necessary signal conditioning and isolation functions. This often allows a PLC to be directly connected to process actuators and transducers (pumps and valves) without the need for intermediate circuitry and relays.

To provide this signal conversion programmable controllers are available with a choice of input / output units to suit different requirements.

For example:

Inputs:	5 V (TTL level) switched I/ P
	24 V switched I/ P
	110 V switched I/ P
	240 v switched I/ P
Outputs:	24 V 100 mA switched O/ P

110 V 1mA

240 V 1 A a.c. (triac)

240 V 2 A a.c. (relay)

It is standard practice for all I/O channels to electrically isolated from the controlled process, using opto-isolator circuits on the I/O modules. An opto-isolator circuit consists of a light emitting diode and a phototransistor, forming an opto coupled pair that allows small signals to pass through, but will clamp any high voltage spikes or surges down to the same small level. This provides protection against switching transients and power-supply surges, normally up to 1500V.

In small self contained PLC's in which all I/ O points are physically located on the one casing, all inputs will be of one type (e.g. 24 V) and the same for outputs (e.g. all 240 V triac). This is because manufacturers supply on the standard function boards for economic reasons. Modular PLC's have greater flexibility of I/ O, however, since the user can select from several different types and combinations of input and output modules.

In all cases the input/output units are designed with the aim of simplifying the connections of process transducers and actuators to the programmable controller. For these purpose all PLC'S are equipped with standard screw terminals or plugs on every I\O point, allowing the rapid and simple removal and replacement of a faulty I/ O card. Every input\output point has a unique address or channel number, which is using during program development to specify to monitoring of an input or the activating of a particular output within the program. Indication of the status of input\output channels is provided by light-emitting diode (LED's) on the PLC or I/ O unit, making it simple to check the operation of process inputs and outputs from the PLC itself.

b) Central Processing Unit (CPU)

The CPU controls and supervises all operations within the PLC, carrying out programmed instructions stored in memory. An internal communications highway or bus system carries information to and from the CP, memory and I/ O units, under control of the CPU. The CPU is supplied with a clock frequency by an external quartz crystal or RC oscillator, typically between 1 and 8 megahertz depending on the microprocessor used and the area of application.

The clock determines the operating speed of the PLC and provides timing\synchronization for all elements in the system. Virtually all modern programmable controllers are microprocessor based using a micro as a system CPU. Some larger PLC's also employ additional microprocessor to control complex, time-consuming functions such as mathematical processing, three terms PID control.

c) Memory

- For program storage all modern programmable controllers use semiconductor memory devices such as RAM read\write memory, or a programmable read-only memory of the EPROM or EEPROM families.

In the virtually all cases RAM is used for initial program development and testing, as it follows changes to be easily made in program. The current trend is to be providing CMOS RAM because of it is very low power consumption, to provide battery back up to this RAM in order to maintain the contents when the power is removed from the PLC system. This battery has a lifespan of at least one year before replacement is necessary, or alternatively a rechargeable type may be supplied with the system being recharge whenever the main PLC power supply is on.

This feature makes programs stored in RAM virtually permanent. Many users operate their PLC systems on this basis alone, since it permits future program alterations if and when necessary.

After a program is fully developed and tested it may be loaded (blown) into a PROM or EPROM memory chip, which are normally cheaper than RAM devices. PROM programming is usually carried out with a special purpose programming unit, although many programmable controllers now have this facility built-in, allowing programs in the PLC RAM to be down loaded into a PROM IC placed in a socket provided on the PLC itself.

(a) In addition to program storage, a programmable controller may require memory for other functions:

1- Temporary buffer store for input\output channels status- I/ O RAM

2- Temporary storage for status of internal function (timers, counters, marker relays)

Since these consist of changing data they require RAM read\write memory, which may be battery-backed in sections.

d) Memory size

Smaller programmable controllers normally have a fixed memory size, due in part to the physical dimensions of the unit. This varies in capacity between 300 and 1000 instructions depending on the manufacturer. This capacity may not appear large enough to be very useful, but it has been estimated that 90 % of all binary control tasks can be solved using less than 1000 instructions, so there is sufficient space to meet most users needs.

Larger PLC's utilize memory modules of between 1K and 64K in size, allowing the system to be expanded by fitting addition RAM or PROM memory cards to the PLC rack.

As integrated circuit memory costs continue to fall, the PLC manufacturers are providing larger program memories on all products.

e) Logic instruction set

The most common technique for programming small PLC's is to draw s ladder diagram of the logic to be used, and then convert this in to mnemonic instructions, which will be keyed in to programming panel attached to the programmable controller. These instructions are similar in appearance to assembly-type codes, but refer to physical inputs, outputs and functions within the PLC itself.

The instruction set consists of logic instructions (mnemonics) that represent the actions that may be performed within a given programmable controller. Instructions sets vary between PLC's from different manufacturers, but are similar in terms of the control actions performed.

Because the PLC logic instruction set tends to be small, it can be quickly mastered and used by control technicians and engineers.

Each program instructions are made up of two parts: a mnemonic operation component or opcode, and an address or operand component that identifies particular elements

(E.g. outputs) within the PLC.

For example;

Opcode	Operand
OUT	Y430
Device symbol	Identifier

Here the instruction refers to output (Y) number 430

f) Input\output numbering

These instructions are used the program logic control circuits that have been designed in ladder diagram form, by assigning all physical inputs and outputs with an operand suitable to the PLC being used. The numbering system used differs between manufacturers, but certain common terms exist. For example, Texas instrument and Mitsubishi use the symbol X to represent inputs, and Y to label outputs.

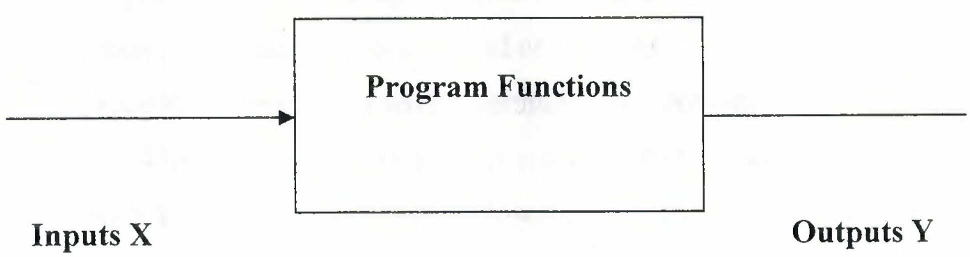


Figure 7 Programmable Controller

A range of addresses will be allocated to particular elements: for example

Mitsubishi F40 PLC: 24 inputs: X400 – 407, 410 – 413

X500 – 507, 510 – 513

16 Outputs: Y430 – 437

Y530 – 537

Inspections of these numbers ranges will reveal that there is no overlap of addresses between functions; that is, 400 must be an input, 533 must be an output. Thus for these programmable controllers the symbol X or Y is redundant, being used purely for the benefit of the user, who is unlikely to remember what element 533 represents. However, for many PLC's both parts of the address are essential, since the I\O number ranges are identical. For example the Klockner-Moeller range of controllers:

Sucos PS 21 PLC: 8 inputs I0 to 7, etc. 8 Outputs Q0 to 7, etc

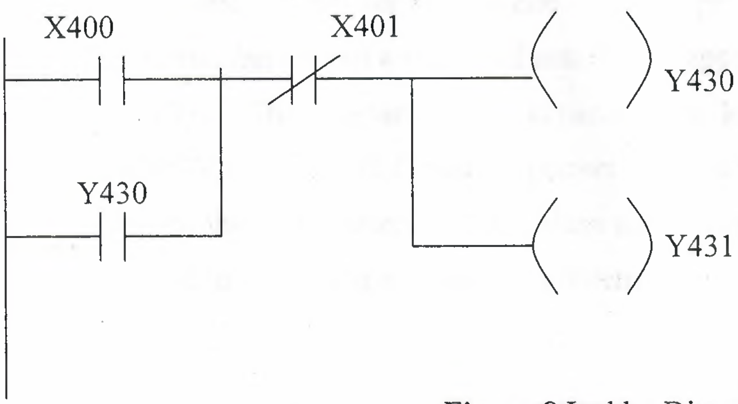


Figure 8 Ladder Diagram

To program the ladder diagram given in figure 1.2, the following code would be written, and then programmed in to a keypad or terminal.

1. LD X400 starts a rung with a normally open contact
2. OR Y430 connect a normally open contact in parallel
3. ANI X401 connect a normally closed contact in series
4. OUT Y430 drive an output channel
5. OUT Y431 drive another channel
6. END end of program-return to start

2. AN OVERVIEW OF SIEMENS S7-200 MICRO-CONTROLLER

2.1. Overview of an S7-200

STEP 7-Micro/WIN supports the S7-200 CPUs by giving you the features to set up and manage your application project. A project consists of the program you enter with STEP 7-Micro/WIN, along with the documentation you write for the program and the configuration you set up for the CPU.

You have the option of selecting either Ladder or Statement List as your programming language. With the S7-200 CPUs, you have a basic program structure that gives you flexibility in setting up any subroutines or interrupts that you program.

2.2. Introduction to the Simatic S7-200 Micro PLC

The Simatic S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. Figure 2.1 shows an S7-200 Micro PLC. The compact design, expandability, low cost, and powerful instruction set of the S7-200 Micro PLC make a perfect solution for controlling small applications. In addition, the wide variety of CPU sizes and voltages provides you with the flexibility you need to solve your automation problems.

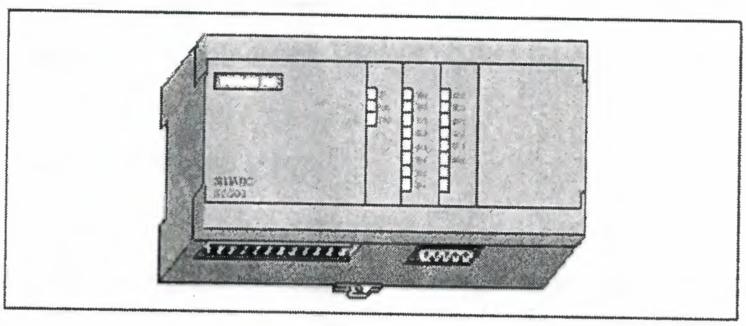


Figure 2.1. S7-200 Micro PLC

2.3. Comparing the Features of the S7-200 Micro PLCs

Feature	CPU 212	CPU 214	CPU 215	CPU 216
Physical Size of Unit	160 mm x 80 mm x 62 mm	197 mm x 80 mm x 62 mm	218 mm x 80 mm x 62 mm	218 mm x 80 mm x 62 mm
Memory				
Program (EEPROM)	512 words	2 Kwords	4 Kwords	4 Kwords
User data	512 words	2 Kwords	2.5 Kwords	2.5 Kwords
Internal memory bits	128	256	256	256
Memory cartridge	None	Yes (EEPROM)	Yes (EEPROM)	Yes (EEPROM)
Optional battery cartridge	None	200 days typical	200 days typical	200 days typical
Backup(super capacitor)	50 hours typical	190 hours typical	190 hours typical	190 hours typical
Inputs/Outputs (I/O)				
Local I/O	8 DI / 6 DQ	14 DI / 10 DQ	14 DI / 10 DQ	24 DI / 16 DQ
Expansion modules (max.)	2 modules	7 modules	7 modules	7 modules
Process-image I/O register	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ
Analog I/O (expansion)	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ
Selectable input filters	No	Yes	Yes	Yes

2.4. Major Components of the S7-200 Micro PLC

An S7-200 Micro PLC consists of an S7-200 CPU module alone or with a variety of optional expansion modules

2.4.1. CPU Module

The S7-200 CPU module combines a central processing unit (CPU), power supply, and discrete I/O points into a compact, stand-alone device.

- The CPU executes the program and stores the data for controlling the automation task or process.
- The power supply provides electrical power for the base unit and for any expansion module that is connected.
- The inputs and outputs are the system control points: the inputs monitor the signals from the field devices (such as sensors and switches), and the outputs control pumps, motors, or other devices in your process.
- The communications port allows you to connect the CPU to a programming device or to other devices. Some S7-200 CPUs have two communications ports.
- Status Lights provide visual information about the CPU mode (RUN or STOP), the current state of the local I/O, and whether a system fault has been detected.

The SIMATIC S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. Compact design, low cost, and a powerful instruction set make the

S7-200 controllers a perfect solution for controlling small applications. The wide variety of CPU sizes and voltages, and the windows-based programming tool, give you the flexibility you need to solve your automation problems.

The S7-200 is characterized by the following:

- Easy entry
- Uncomplicated operation
- Peerless real-time characteristics
- Powerful communications capabilities

The S7-200 achieves its full performance potential in distributed automation solutions thanks especially to the integrated ProFiBus-DP connection.

The application area of the SIMATIC S7-200 extends from replacing simple relays and contactors right up to more complex automation tasks.

The S7-200 also covers areas where previously special electronics have been developed for cost reasons. Application areas include:

- Baling processes
- Plaster & Cement mixers
- Suction Plants
- Centralized lubricating systems/flange lubricating systems
- Woodworking machinery
- Gate controls
- Hydraulic lifts
- Conveyor systems
- Food & Drink Industry
- Laboratories
- Modem applications via dial-up, leased-line, or radio remote monitoring (SCADA)
- Electrical Installations

Mechanical features include:

- Rugged, compact plastic housing using SIMATIC's prize-winning design
- Easily accessible wiring and operator control and display elements protected by front covers
- Installs on standard horizontal or vertical DIN rail or direct cabinet mounting with built-in mounting
- Terminal block as permanent wiring assembly (optional)

Design features:

- International standards; Meets the requirements through compliance with VDE, UL, CSA and FM standards.
- The quality management system used during manufacturing has ISO 9001 certification; and Data back up; the user program and the most important parameter settings are stored in the internal EEPROM. A heavy-duty capacitor provides additional back up for all data over longer periods (typically up to 50 or 190 hours). An optional battery module ensures that the data remain stored for 200 days (typically) after power failure.

Benefits of the S7-200:

Complete Automation Solution

The SIMATIC S7-200 Micro PLC is a full-featured programmable logic control system offering stand-alone CPUs, micro-modular expansion capability, and operator interface solutions. Almost any application that requires automation, from basic discrete or analog control, to intelligent networked solutions, can benefit by using the powerful S7-200 family of products.

Value for OEMs

Wherever central controllers or expensive custom electronic control systems are used, the SIMATIC S7-200 offers a significantly more economical alternative. Our off-the-shelf, compact solution, is packed with features, and is accepted around the world as a Micro PLC standard.

Real-time Speed & Versatility

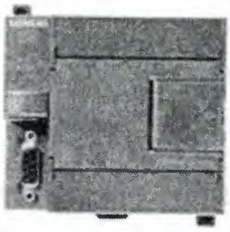
The SIMATIC S7-200 offers real-time control with Boolean processing speeds of 0.37µs per instruction. This fast execution speed, combined with our 20Khz high-speed counters, interrupts, and 20KHz pulse outputs, provide quick responses in demanding real-time applications. The S7-200 has over 200 instructions, including math, PID, For/Next loops, subroutines, sequence control, and more!

Integrated Communications

All S7-200 CPUs offer at least one RS485 communication port with speeds up to 187.5Kbaud. This not only provides fast access for programming and maintenance, but also allows you to build master/slave networks with up to 31 stations.

Non-S7-200 devices, such as bar code readers, intelligent machines, etc. can also be connected by using our FreePort capability. With FreePort, you can easily adapt the S7-200 CPU to virtually any serial ASCII protocol.

S7- 221:



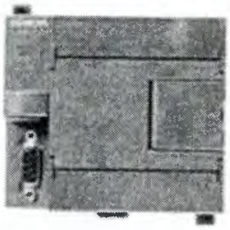
The compact solution that's optimal for first-time users and those needing to change systems. Take the competitive edge and get a kick-start in the right direction with a 221 programmable controller.

Inputs/outputs: 10

Program memory: 4 KByte

Bit processing time: 0.37 µs

S7- 222:



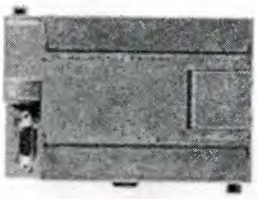
The superior compact solution. Masters all requirements from complex machines right up to small plant solutions.

Inputs/outputs: 14, expandable

Program memory: 4 KByte

Bit processing time: 0.37 µs

S7-224:



The compact high-performance CPU for all those cases where speed, better communication, and more complex programs provide the decisive advantage

Inputs/outputs: 24, expandable

Program memory: 8 KByte

Bit processing time: 0.37 µs

S7-226 & S7-226XM:

The new, compact high-performance CPU. Expandable Inputs/Outputs and additional 485 PPI-interface complex automation tasks.

Inputs/outputs: 40, expandable

226 Program memory: 8 KByte

226XM Program memory: 16 KByte

Additional Data for S7-200 PLCs:

- Simatic Support Pages for S7-200 Hardware FAQ's
- Simatic Support Pages for S7-200 Hardware Manuals
- Simatic Support Pages for S7-200 Hardware Updates (Product Announcements)
- S7-200, the Micro PLC Product Family from Siemens (598 KB)
- Part Numbers for S7-200 PLC's (339 KB)

CHAPTER : 3

TYPES OF PLC

PLCs are separated into two according to their building mechanisms.

3.1 Compact PLCs

Compact PLCs are manufactured such that all units forming the PLC are placed in a case. They are low price PLC with lower capacity. Small or medium size machine manufacturers usually prefer them. In some types compact enlargement module is present.

3.2 Modular PLCs

Combining separate modules together in a board forms them. They can have different memory capacity, I / O numbers, power supply up to the necessary limits.

Some examples: SIEMENS S5-115U, SIEMENS S7-200, MITSUBISHI PC40, TEXAS INSTRUMENT PLC'S, KLOCKNER-MOELLER PS316, OMRON C200H.

3.3 Categories of PLC

The increasing demand from industry for programmable controllers that can be applied to different forms and sizes of control tasks has resulted in most manufacturers producing a range of PLC's with various levels of performance and facilities.

Typical rough definitions of PLC size are given in terms of program memory size and the maximum number of input\output points the system can support. Table 3.1 gives an example of these categories.

Table 3.1 Categories of PLC

PC size	Max I \ O points	Use memory size
Small	40 / 40	1K
Medium	128 / 128	4K
Large	> 128 / > 128	> 4K

Table 3.2 Summary of the S7-200 CPUs

Feature	CPU 212	CPU 214	CPU 215	CPU 216
Physical Size of unit	160mm x 80mm x 62mm	197mm x 80mm x 62mm	218mm x 80mm x 62mm	218mm x 80mm x 62 mm
Memory				
Program (EEPROM)	512 words	2 k words	4 k words	4 k words
User Data	512 words	2 k words	2.5 k words	2.5 k words
Internal Memory Bits	128	256	256	256
Memory Cartridge	None	Yes (EEPROM)	Yes (EEPROM)	Yes (EEPROM)
Optional Battery Cartridge	None	200 Days typical	200 Days typical	200 Days typical
Backup (super capacitor)	50 Hours typical	190 Hours typical	190 Hours typical	190 Hours typical
Inputs/Outputs(I/O)				
Local I/O	8 DI / 6 DQ	14 DI / 10 DQ	14 DI / 10 DQ	24 DI / 16 DQ
Expansion Modules	2 Modules	7 Modules	7 Modules	7 Modules

NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

WATER LEVEL CONTROLLING BY USING PLC

Graduating Project
EE - 400

Student: Mesut BULUT

Supervisor : Assit.Prof.Dr. Özgür C. ÖZERDEM

Nicosia 2006

ACKNOWLEDGEMENTS

It is my pleasure to thank to following people in the ***Near East University*** and My Family Without them, I could never finish my undergraduate program and my Graduation Project would not have been successfully completed on time.

First of all, I would like to thanks to my supervisor ***Assist. Prof. Dr Özgür C. Özerdem*** for supervising my project. Under the guidance of him I successfully overcome many diffficulties and I learned a lot about electric and electronic. In each discussion, he used to explain the problems and answer my questions. He always helped me a lot and I felt remarkable progress during his supervision. I also thank **Prof. Dr.Sezai DİNÇER,** ***Assist. Prof. Dr. Kadri Bürüncük, Mr. Mustafa GÜNDÜZ , Assoc. Prof. Adnan Kashman,*** **Mrs. Filiz Alshanable** that I spent funny times during their lecture times.

Then I also thank to my friends: **Serbulent Karagoz, Mehmet Akca, Aylin Duran** and **Samet Biricik** because they also motivated me when I was preparing My project

Finally and specialally thanks for my family, especially my parents and my elder brother **Mustafa Bulut** for being patientful during my undergraduate degree study. I could never have completed my study without their encouragement and endless support.

ABSTRACT

Water level Controlling is an example that we can solve so many logical problems by using PLC's. Because PLC is a machine that you can get any output on depending on your input conditions, so it can solve so many problem in industry easier than relays.
As PLC technology has advanced, so have programming languages and communications capabilities, along with many other important features. Today's PLCs offer faster scan times, space efficient high-density input/output systems, and special interfaces to allow non-traditional devices to be attached directly to the PLC. Not only can they communicate with other control systems, they can also perform reporting functions and diagnose their own failures, as well as the failure of a machine or process. Size is typically used to categorize today's PLC, and is often an indication of the features and types of applications it will accommodate.

In this project CPU 212 8 inputs, 6 outputs, 24 volt DC input 220 Volt AC output PLC is used, and an automation of a Lamp and an industrial motor is realized by PLC programs.

TABLE OF CONTENTS

4 .WATER LEVEL CONTROLLING

1.1 INTRODUCTION

Control Engineering has been evolved day-by-day time. In the past people were the main methods for controlling everything. More recently electricity has been used for control and early electrical control was based on relays. These relays allow power to be switched on and off without a mechanical switch. It is common to use relays to make simple logical control decisions. The development of low cost computer has brought the most recent revolution, the Programmable Logic Controller (PLC).

1.2 WHAT IS A PLC ?

PLC stands for Programable Logic Controllers that PLCs are often defined as miniature industrial computers that contain hardware and software that is used to perform control functions. 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.

A PLC consists of two basic sections: the central processing unit (CPU) and the input/output interface system. The CPU, which controls all PLC activity, can further be broken down into the processor and memory system. The input/output system is physically connected to field devices (e.g., switches, sensors, etc.) and provides the interface between the CPU and the information providers (inputs) and controllable devices (outputs). To operate, the CPU "reads" input data from connected field devices through the use of its input interfaces, and then "executes", or performs the control program that has been stored in its memory system. Programs are typically created in ladder logic, a language that closely resembles a relay-based wiring schematic, and are entered into the CPU's memory prior to operation. Finally, based on the program, the PLC "writes", or updates output devices via the output interfaces. This process, also known as scanning, continues in the same sequence without interruption, and changes only when a change is made to the control program

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 world applications. If there is industry present, changes are good that there is a PLC present. If you are involved in machining, packaging material handling, and 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 PLC.

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

As you see, if the process becomes more complicated, then we have to use a device the simplify that. We use PLC for this process. We can program the PLC to count its inputs and turn the solenoids for the specified time.

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

1.3 HISTORY OF PLC

In the 1960's PLC's 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 (MODICON) to a major US car manufacturer. Other companies at the time proposed computer based upon the PDP – 8. The MODICON 084 brought the world's first PLC into commercial production.

The first PLC can be traced back to 1968 when Bedford Associates, a company in Bedford, MA, developed a device called a Modular Digital Controller for General Motors (GM). The MODICON, as it was known, was developed to help GM eliminate traditional relay-based machine control systems. Because relays are mechanical devices, they have limited lifetimes. They are also cumbersome, especially in large applications where thousands of them may exist. With so many relays to work with, wiring and troubleshooting could be quite complicated. Since the MODICON was an electronic device, not a mechanical one, it was perfect for GM's requirements, as well as for many other manufacturers and users of control equipment. With less wiring, simpler troubleshooting, and easy programming, PLC technology caught on quickly. In the late 1960's PLC's 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 (MODICON) to a major US car manufacturer. Other companies at the time proposed computer 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 that 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, possible 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. The 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 techniques were sequencer state machines and the bit-slice based CPU. The AMD 2901 and 2903 were quite popular in MODICON and A-B PLC's. Conventional microprocessors lacked the power to quickly solve PLC logic in all but the smallest PLC's. As conventional microprocessors evolved, larger and larger PLC's were being based upon them. However, even today some are still based upon the 2903. MODICON has yet the build a faster PLC then their 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 PLC's 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.

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

In The 90's have seen a gradual reduction in the introduction new protocols, and the modernization of the physical layers of same of the more popular protocols that survived the 1980's. The latest standard has tried to merge PLC- programming languages less than one international standard.

1.4 COMPAREING PLC WITH RELAY CONTROLLING

Modern control systems still include relays, but these are rarely used for logic. A relay is a simple device that uses a magnetic field to control a switch, as pictured in Figure 1. When a voltage is applied to the input coil, the resulting current creates a magnetic field. The magnetic field pulls a metal switch (or reed) towards it and the contacts touch, closing the switch. The contact that closes when the coil is energized is called normally open. The normally closed contacts touch when the input coil is not energized. Relays are normally drawn in schematic form using a circle to represent the input coil. The output contacts are shown with two parallel lines. Normally open contacts are shown as two lines, and will be open (non-conducting) when the input is not energized. Normally closed contacts are shown with two lines with a diagonal line through them. When the input coil is not energized the normally closed contacts will be closed (conducting).

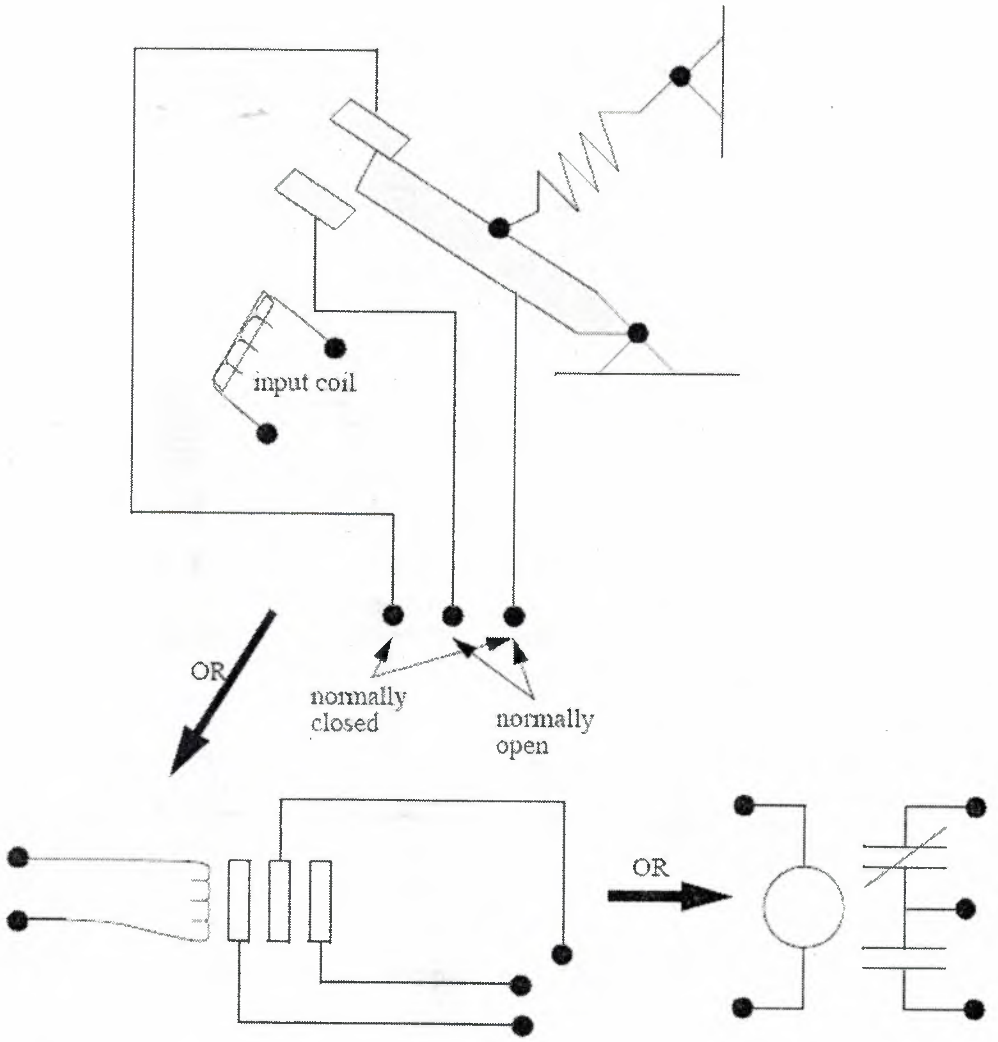


Figure : 1 Simple Relay Layouts and Schematics

Relays are used to let one power source close a switch for another (often high current) power source, while keeping them isolated. An example of a relay in a simple control application is shown in Figure 2. In this system the first relay on the left is used as normally closed, and will allow current to flow until a voltage is applied to the input A. The second relay is normally open and will not allow current to flow until a voltage is applied to the input B. If current is flowing through the first two relays then current will flow through the coil in the third relay, and close the switch for output C. This circuit would normally be drawn in the ladder logic form. This can be read logically as C will be on if A is off and B is on.

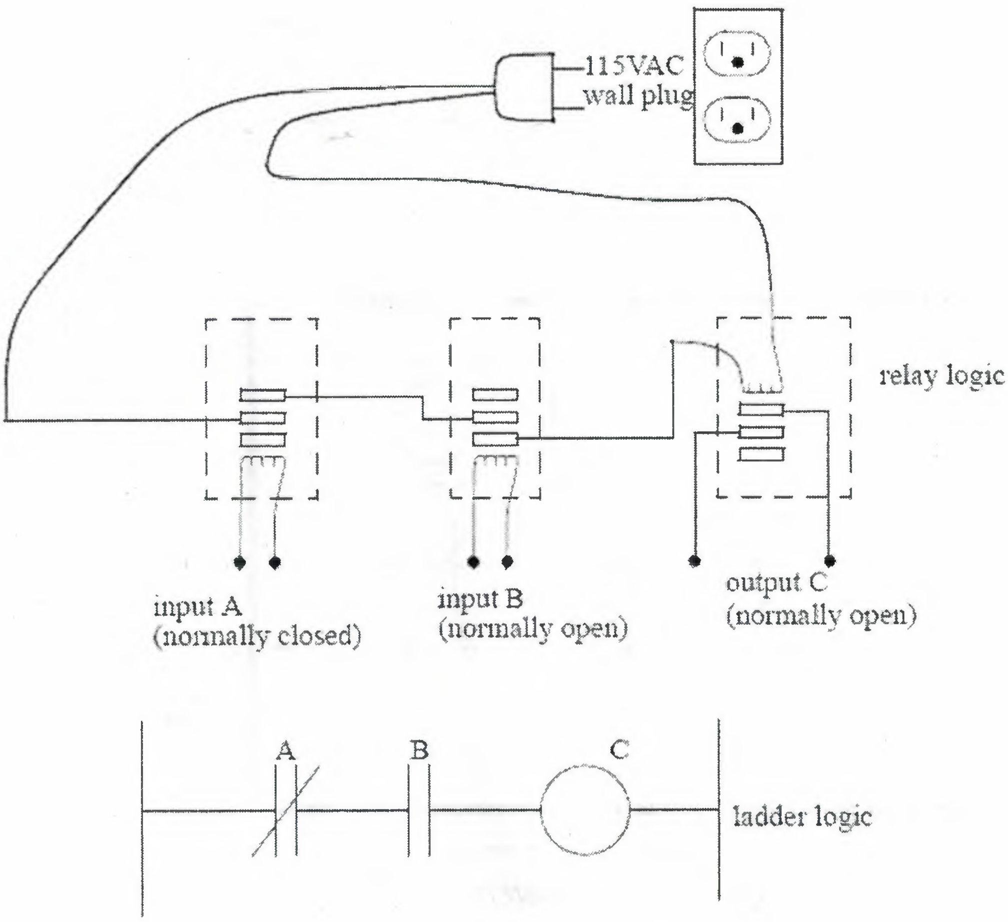


Figure : 2 A Simple Relay Controller

The example in Figure 2 does not show the entire control system, but only the logic. When we consider a PLC there are inputs, outputs, and the logic. Figure 3 shows a more complete representation of the PLC. Here there are two inputs from push buttons. We can imagine the inputs as activating 24V DC relay coils in the PLC. This in turn drives an output relay that switches 115V AC, that will turn on a light. Note, in actual PLCs inputs are never relays, but outputs are often relays. The ladder logic in the PLC is actually a computer program that the user can enter and change. Notice that both of the input push buttons are normally open, but the ladder logic inside the PLC has one normally open contact, and one normally closed contact. Do not think that the ladder logic in the PLC needs to match the inputs or outputs. Many beginners will get caught trying to make the ladder logic match the input types.

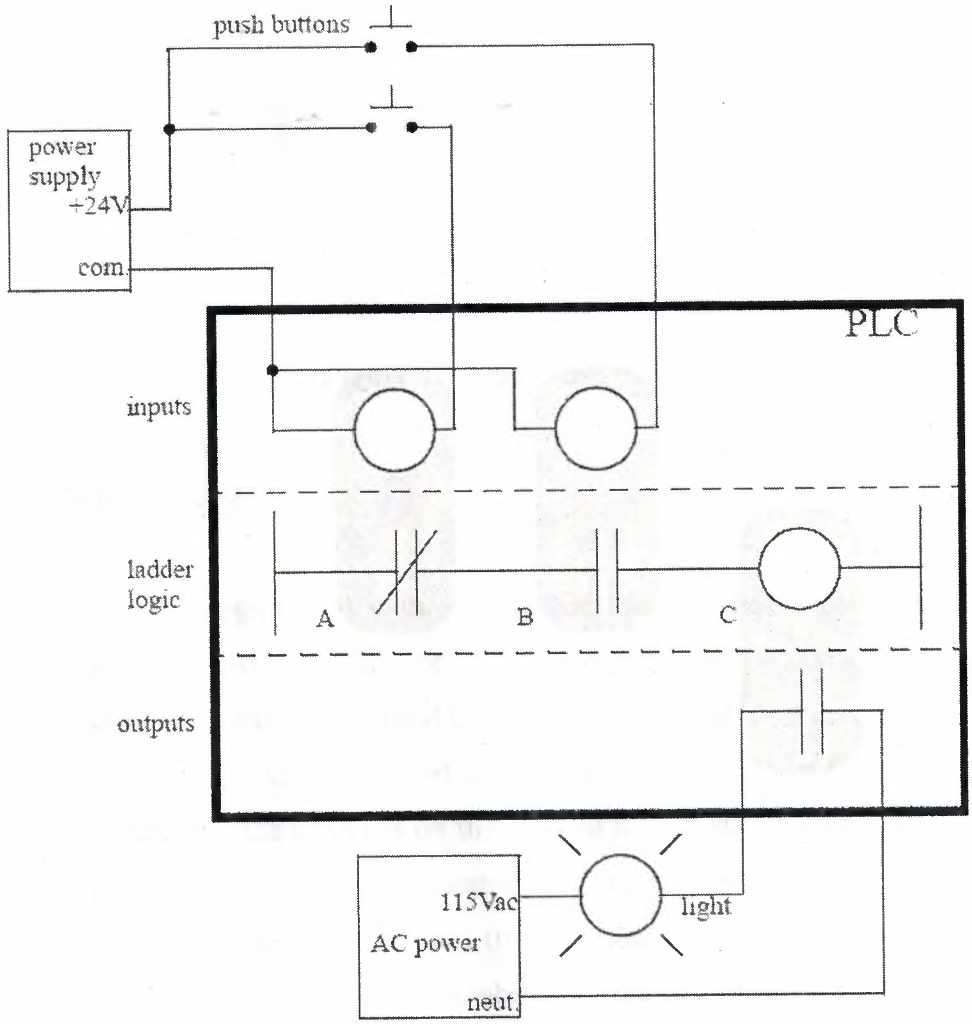


Figure : 3 A PLC Illustrated With Relays

Many relays also have multiple outputs (throws) and this allows an output relay to also be an input simultaneously. The circuit shown in Figure 4 is an example of this, it is called a seal in circuit. In this circuit the current can flow through either branch of the circuit, through the contacts labelled A or B. The input B will only be on when the output B is on. If B is off, and A is energized, then B will turn on. If B turns on then the input B will turn on, and keep output B on even if input A goes off. After B is turned on the output B will not turn off.

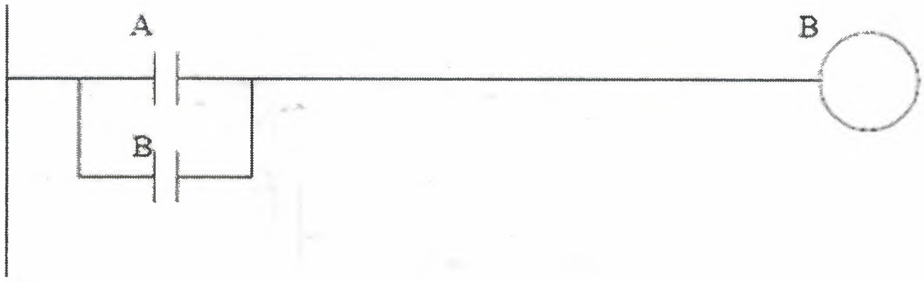


Figure : 4 A Seal-in Circuit

1.4.1 Programming

The first PLCs were programmed with a technique that was based on relay logic wiring schematics. This eliminated the need to teach the electricians, technicians and engineers how to *program* a computer - but, this method has stuck and it is the most common technique for programming PLCs today. An example of ladder logic can be seen in Figure 5. To interpret this diagram imagine that the power is on the vertical line on the left hand side, we call this the hot rail. On the right hand side is the neutral rail. In the figure there are two rungs, and on each rung there are combinations of inputs (two vertical lines) and outputs (circles). If the inputs are opened or closed in the right combination the power can flow from the hot rail, through the inputs, to power the outputs, and finally to the neutral rail. An input can come from a sensor, switch, or any other type of sensor. An output will be some device outside the PLC that is switched on or off, such as lights or motors. In the top rung the contacts are normally open and normally closed. Which means if input *A* is on and input *B* is off, then power will flow through the output and activate it. Any other combination of input values will result in the output *X* being off.

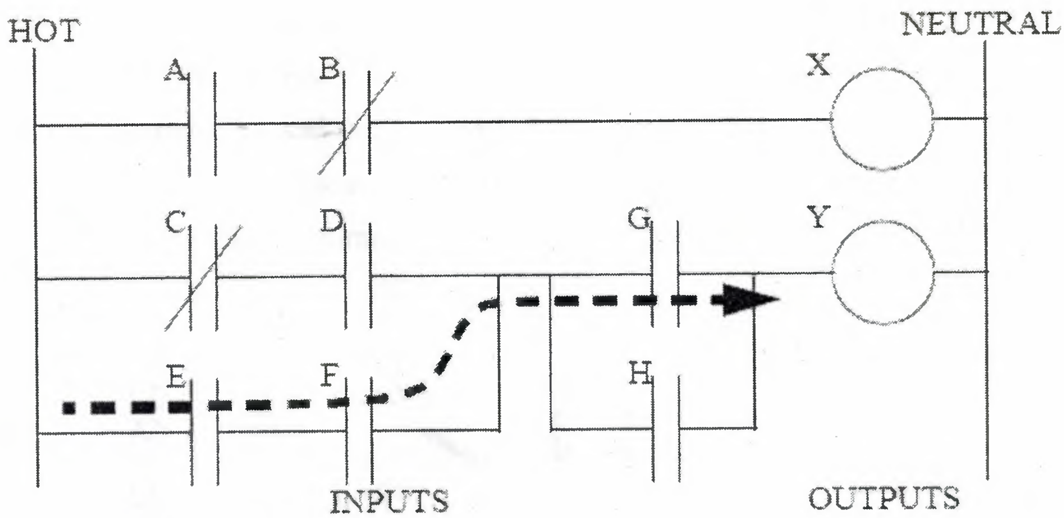


Figure : 5 A Simple Ladder Logic Diagram

The second rung of Figure 5 is more complex, there are actually multiple combinations of inputs that will result in the output *Y* turning on. On the left most part of the rung, power could flow through the top if *C* is off and *D* is on. Power could also (and simultaneously) flow through the bottom if both *E* and *F* are true. This would get power half way across the rung, and then if *G* or *H* is true the power will be delivered to output *Y*. In later chapters we will examine how to interpret and construct these diagrams.

There are other methods for programming PLCs. One of the earliest techniques involved mnemonic instructions. These instructions can be derived directly from the ladder logic diagrams and entered into the PLC through a simple programming terminal. An example of mnemonics is shown in Figure 6. In this example the instructions are read one line at a time from top to bottom. The first line *00000* has the instruction *LDN* (input load and not) for input *00001*. This will examine the input to the PLC and if it is off it will remember a *1* (or true), if it is on it will remember a *0* (or false). The next line uses an *LD* (input load) statement to look at the input. If the input is off it remembers a *0*, if the input is on it remembers a *1* (note: this is the reverse of the *LD*). The *AND* statement recalls the last two numbers remembered and if the are both true the result is a *1*, otherwise the result is a *0*. This result now replaces the two numbers that were recalled, and there is only one number remembered. The process is repeated for lines *00003* and *00004*, but when these are done there are now three numbers remembered. The oldest number is from the *AND*, the newer numbers are from the two *LD* instructions. The *AND* in line *00005* combines the results from the last *LD* instructions and now there are two numbers remembered. The *OR* instruction takes the two numbers now remaining and if either one is a *1* the result is a *1*, otherwise the result is a *0*. This result replaces the two numbers, and there is now a single number there. The last instruction is the *ST* (store output) that will look at the last value stored and if it is *1*, the output will be turned on, if it is *0* the output will be turned off.

00000	LDN	00001
00001	LD	00002
00002	AND	
00003	LD	00003
00004	LD	00004
00005	AND	
00006	OR	
00007	ST	00107
00008	END	

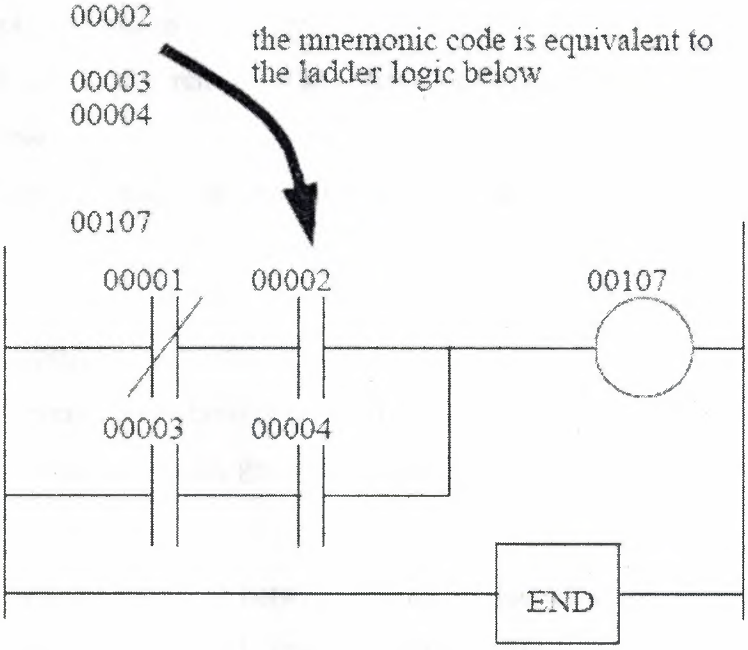


Figure : 6 An Example of a Mnemonic Program and Equivalent Ladder Logic

The ladder logic program in Figure 6, is equivalent to the mnemonic program. Even if you have programmed a PLC with ladder logic, it will be converted to mnemonic form before being used by the PLC. In the past mnemonic programming was the most common, but now it is uncommon for users to even see mnemonic programs.

1.5 PLC Hardware Design

Programmable controllers are purpose-built computers consisting of three functional areas:

- Processing:
- Memory:
- Input / output:

Input conditions to the PLC are sensed and than stored in the memory, where the PLC performs the programmed logic instructions on these input states. Output conditions are then generated to drive associated equipment. The action taken depends totally on the control program held in memory.

In smaller PLC these functions are performed by individual printed circuit cards within a single compact unit, whilst larger PLC's are constructed on a modular basis with function modules slotted in to the back plane connectors of the mounting rack.

This allows simple expansion of the system when necessary. In both these cases the individual circuit board are easily removed and replaced, facilitating rapid repair of the system should faults develop.

In addition a programming unit is necessary to download control programs to the PLC memory.

a) Input output / units

Most PLC'S operate internally at between 5 and 15 V d.c. (Common TTL and CMOS voltages), whilst process signals much greater, typically 24 V d.c. To 240 V a.c. at several amperes.

The I / O units form the interface between the microelectronics of the programmable controller and real world outside, and must therefore provide all, necessary signal conditioning and isolation functions. This often allows a PLC to be directly connected to process actuators and transducers (pumps and valves) without the need for intermediate circuitry and relays.

To provide this signal conversion programmable controllers are available with a choice of input / output units to suit different requirements.

For example:

Inputs:	5 V (TTL level) switched I/ P
	24 V switched I/ P
	110 V switched I/ P
	240 v switched I/ P
Outputs:	24 V 100 mA switched O/ P

110 V 1mA

240 V 1 A a.c. (triac)

240 V 2 A a.c. (relay)

It is standard practice for all I/O channels to electrically isolated from the controlled process, using opto-isolator circuits on the I/O modules. An opto-isolator circuit consists of a light emitting diode and a phototransistor, forming an opto coupled pair that allows small signals to pass through, but will clamp any high voltage spikes or surges down to the same small level. This provides protection against switching transients and power-supply surges, normally up to 1500V.

In small self contained PLC's in which all I/ O points are physically located on the one casing, all inputs will be of one type (e.g. 24 V) and the same for outputs (e.g. all 240 V triac). This is because manufacturers supply on the standard function boards for economic reasons. Modular PLC's have greater flexibility of I/ O, however, since the user can select from several different types and combinations of input and output modules.

In all cases the input/output units are designed with the aim of simplifying the connections of process transducers and actuators to the programmable controller. For these purpose all PLC'S are equipped with standard screw terminals or plugs on every I\O point, allowing the rapid and simple removal and replacement of a faulty I/ O card. Every input\output point has a unique address or channel number, which is using during program development to specify to monitoring of an input or the activating of a particular output within the program. Indication of the status of input\output channels is provided by light-emitting diode (LED's) on the PLC or I/ O unit, making it simple to check the operation of process inputs and outputs from the PLC itself.

b) Central Processing Unit (CPU)

The CPU controls and supervises all operations within the PLC, carrying out programmed instructions stored in memory. An internal communications highway or bus system carries information to and from the CP, memory and I/ O units, under control of the CPU. The CPU is supplied with a clock frequency by an external quartz crystal or RC oscillator, typically between 1 and 8 megahertz depending on the microprocessor used and the area of application.

The clock determines the operating speed of the PLC and provides timing\synchronization for all elements in the system. Virtually all modern programmable controllers are microprocessor based using a micro as a system CPU. Some larger PLC's also employ additional microprocessor to control complex, time-consuming functions such as mathematical processing, three terms PID control.

c) Memory

- For program storage all modern programmable controllers use semiconductor memory devices such as RAM read\write memory, or a programmable read-only memory of the EPROM or EEPROM families.

In the virtually all cases RAM is used for initial program development and testing, as it follows changes to be easily made in program. The current trend is to be providing CMOS RAM because of it is very low power consumption, to provide battery back up to this RAM in order to maintain the contents when the power is removed from the PLC system. This battery has a lifespan of at least one year before replacement is necessary, or alternatively a rechargeable type may be supplied with the system being recharge whenever the main PLC power supply is on.

This feature makes programs stored in RAM virtually permanent. Many users operate their PLC systems on this basis alone, since it permits future program alterations if and when necessary.

After a program is fully developed and tested it may be loaded (blown) into a PROM or EPROM memory chip, which are normally cheaper than RAM devices. PROM programming is usually carried out with a special purpose programming unit, although many programmable controllers now have this facility built-in, allowing programs in the PLC RAM to be down loaded into a PROM IC placed in a socket provided on the PLC itself.

(a) In addition to program storage, a programmable controller may require memory for other functions:

1- Temporary buffer store for input\output channels status- I/ O RAM

2- Temporary storage for status of internal function (timers, counters, marker relays)

Since these consist of changing data they require RAM read\write memory, which may be battery-backed in sections.

d) Memory size

Smaller programmable controllers normally have a fixed memory size, due in part to the physical dimensions of the unit. This varies in capacity between 300 and 1000 instructions depending on the manufacturer. This capacity may not appear large enough to be very useful, but it has been estimated that 90 % of all binary control tasks can be solved using less than 1000 instructions, so there is sufficient space to meet most users needs.

Larger PLC's utilize memory modules of between 1K and 64K in size, allowing the system to be expanded by fitting addition RAM or PROM memory cards to the PLC rack.

As integrated circuit memory costs continue to fall, the PLC manufacturers are providing larger program memories on all products.

e) Logic instruction set

The most common technique for programming small PLC's is to draw s ladder diagram of the logic to be used, and then convert this in to mnemonic instructions, which will be keyed in to programming panel attached to the programmable controller. These instructions are similar in appearance to assembly-type codes, but refer to physical inputs, outputs and functions within the PLC itself.

The instruction set consists of logic instructions (mnemonics) that represent the actions that may be performed within a given programmable controller. Instructions sets vary between PLC's from different manufacturers, but are similar in terms of the control actions performed.

Because the PLC logic instruction set tends to be small, it can be quickly mastered and used by control technicians and engineers.

Each program instructions are made up of two parts: a mnemonic operation component or opcode, and an address or operand component that identifies particular elements

(E.g. outputs) within the PLC.

For example;

Opcode	Operand
OUT	Y430
Device symbol	Identifier

Here the instruction refers to output (Y) number 430

f) Input\output numbering

These instructions are used the program logic control circuits that have been designed in ladder diagram form, by assigning all physical inputs and outputs with an operand suitable to the PLC being used. The numbering system used differs between manufacturers, but certain common terms exist. For example, Texas instrument and Mitsubishi use the symbol X to represent inputs, and Y to label outputs.

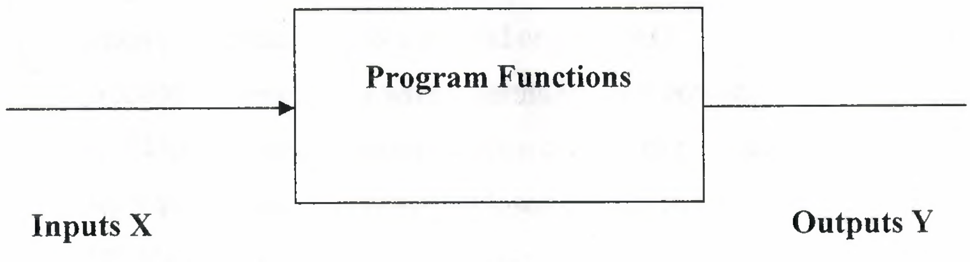


Figure 7 Programmable Controller

A range of addresses will be allocated to particular elements: for example

Mitsubishi F40 PLC: 24 inputs: X400 – 407, 410 – 413

X500 – 507, 510 – 513

16 Outputs: Y430 – 437

Y530 – 537

Inspections of these numbers ranges will reveal that there is no overlap of addresses between functions; that is, 400 must be an input, 533 must be an output. Thus for these programmable controllers the symbol X or Y is redundant, being used purely for the benefit of the user, who is unlikely to remember what element 533 represents. However, for many PLC's both parts of the address are essential, since the I\O number ranges are identical. For example the Klockner-Moeller range of controllers:

Sucos PS 21 PLC: 8 inputs I0 to 7, etc. 8 Outputs Q0 to 7, etc

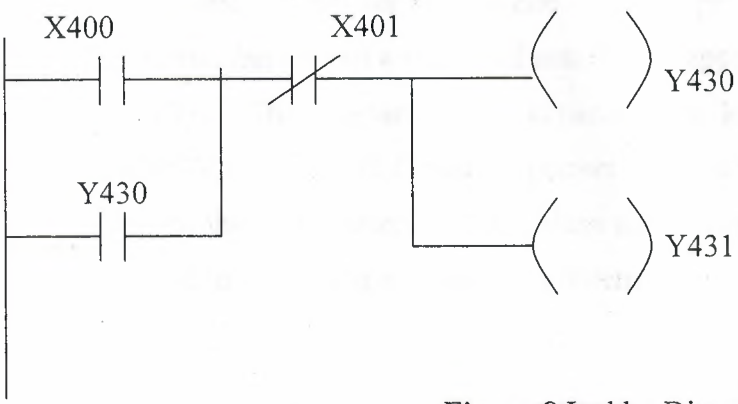


Figure 8 Ladder Diagram

To program the ladder diagram given in figure 1.2, the following code would be written, and then programmed in to a keypad or terminal.

1. LD X400 starts a rung with a normally open contact
2. OR Y430 connect a normally open contact in parallel
3. ANI X401 connect a normally closed contact in series
4. OUT Y430 drive an output channel
5. OUT Y431 drive another channel
6. END end of program-return to start

2. AN OVERVIEW OF SIEMENS S7-200 MICRO-CONTROLLER

2.1. Overview of an S7-200

STEP 7-Micro/WIN supports the S7-200 CPUs by giving you the features to set up and manage your application project. A project consists of the program you enter with STEP 7-Micro/WIN, along with the documentation you write for the program and the configuration you set up for the CPU.

You have the option of selecting either Ladder or Statement List as your programming language. With the S7-200 CPUs, you have a basic program structure that gives you flexibility in setting up any subroutines or interrupts that you program.

2.2. Introduction to the Simatic S7-200 Micro PLC

The Simatic S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. Figure 2.1 shows an S7-200 Micro PLC. The compact design, expandability, low cost, and powerful instruction set of the S7-200 Micro PLC make a perfect solution for controlling small applications. In addition, the wide variety of CPU sizes and voltages provides you with the flexibility you need to solve your automation problems.

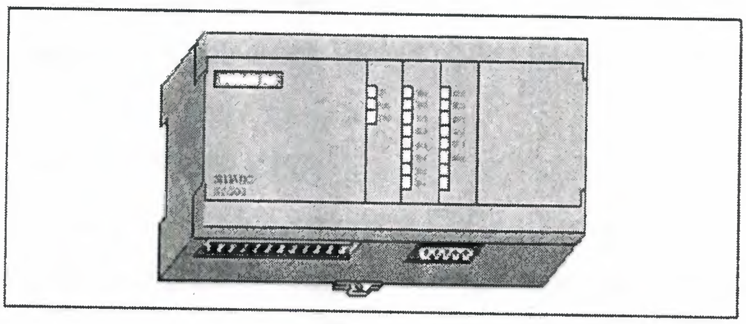


Figure 2.1. S7-200 Micro PLC

2.3. Comparing the Features of the S7-200 Micro PLCs

Feature	CPU 212	CPU 214	CPU 215	CPU 216
Physical Size of Unit	160 mm x 80 mm x 62 mm	197 mm x 80 mm x 62 mm	218 mm x 80 mm x 62 mm	218 mm x 80 mm x 62 mm
Memory				
Program (EEPROM)	512 words	2 Kwords	4 Kwords	4 Kwords
User data	512 words	2 Kwords	2.5 Kwords	2.5 Kwords
Internal memory bits	128	256	256	256
Memory cartridge	None	Yes (EEPROM)	Yes (EEPROM)	Yes (EEPROM)
Optional battery cartridge	None	200 days typical	200 days typical	200 days typical
Backup(super capacitor)	50 hours typical	190 hours typical	190 hours typical	190 hours typical
Inputs/Outputs (I/O)				
Local I/O	8 DI / 6 DQ	14 DI / 10 DQ	14 DI / 10 DQ	24 DI / 16 DQ
Expansion modules (max.)	2 modules	7 modules	7 modules	7 modules
Process-image I/O register	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ
Analog I/O (expansion)	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ
Selectable input filters	No	Yes	Yes	Yes

2.4. Major Components of the S7-200 Micro PLC

An S7-200 Micro PLC consists of an S7-200 CPU module alone or with a variety of optional expansion modules

2.4.1. CPU Module

The S7-200 CPU module combines a central processing unit (CPU), power supply, and discrete I/O points into a compact, stand-alone device.

- The CPU executes the program and stores the data for controlling the automation task or process.
- The power supply provides electrical power for the base unit and for any expansion module that is connected.
- The inputs and outputs are the system control points: the inputs monitor the signals from the field devices (such as sensors and switches), and the outputs control pumps, motors, or other devices in your process.
- The communications port allows you to connect the CPU to a programming device or to other devices. Some S7-200 CPUs have two communications ports.
- Status Lights provide visual information about the CPU mode (RUN or STOP), the current state of the local I/O, and whether a system fault has been detected.

The SIMATIC S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. Compact design, low cost, and a powerful instruction set make the

S7-200 controllers a perfect solution for controlling small applications. The wide variety of CPU sizes and voltages, and the windows-based programming tool, give you the flexibility you need to solve your automation problems.

The S7-200 is characterized by the following:

- Easy entry
- Uncomplicated operation
- Peerless real-time characteristics
- Powerful communications capabilities

The S7-200 achieves its full performance potential in distributed automation solutions thanks especially to the integrated ProFiBus-DP connection.

The application area of the SIMATIC S7-200 extends from replacing simple relays and contactors right up to more complex automation tasks.

The S7-200 also covers areas where previously special electronics have been developed for cost reasons. Application areas include:

- Baling processes
- Plaster & Cement mixers
- Suction Plants
- Centralized lubricating systems/flange lubricating systems
- Woodworking machinery
- Gate controls
- Hydraulic lifts
- Conveyor systems
- Food & Drink Industry
- Laboratories
- Modem applications via dial-up, leased-line, or radio remote monitoring (SCADA)
- Electrical Installations

Mechanical features include:

- Rugged, compact plastic housing using SIMATIC's prize-winning design
- Easily accessible wiring and operator control and display elements protected by front covers
- Installs on standard horizontal or vertical DIN rail or direct cabinet mounting with built-in mounting
- Terminal block as permanent wiring assembly (optional)

Design features:

- International standards; Meets the requirements through compliance with VDE, UL, CSA and FM standards.
- The quality management system used during manufacturing has ISO 9001 certification; and Data back up; the user program and the most important parameter settings are stored in the internal EEPROM. A heavy-duty capacitor provides additional back up for all data over longer periods (typically up to 50 or 190 hours). An optional battery module ensures that the data remain stored for 200 days (typically) after power failure.

Benefits of the S7-200:

<u>Complete Automation Solution</u>

The SIMATIC S7-200 Micro PLC is a full-featured programmable logic control system offering stand-alone CPUs, micro-modular expansion capability, and operator interface solutions. Almost any application that requires automation, from basic discrete or analog control, to intelligent networked solutions, can benefit by using the powerful S7-200 family of products.

<u>Value for OEMs</u>

Wherever central controllers or expensive custom electronic control systems are used, the SIMATIC S7-200 offers a significantly more economical alternative. Our off-the-shelf, compact solution, is packed with features, and is accepted around the world as a Micro PLC standard.

<u>Real-time Speed & Versatility</u>

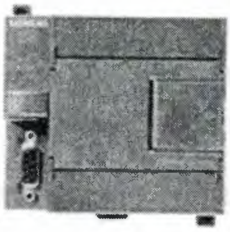
The SIMATIC S7-200 offers real-time control with Boolean processing speeds of 0.37µs per instruction. This fast execution speed, combined with our 20Khz high-speed counters, interrupts, and 20KHz pulse outputs, provide quick responses in demanding real-time applications. The S7-200 has over 200 instructions, including math, PID, For/Next loops, subroutines, sequence control, and more!

Integrated Communications

All S7-200 CPUs offer at least one RS485 communication port with speeds up to 187.5Kbaud. This not only provides fast access for programming and maintenance, but also allows you to build master/slave networks with up to 31 stations.

Non-S7-200 devices, such as bar code readers, intelligent machines, etc. can also be connected by using our FreePort capability. With FreePort, you can easily adapt the S7-200 CPU to virtually any serial ASCII protocol.

S7- 221:



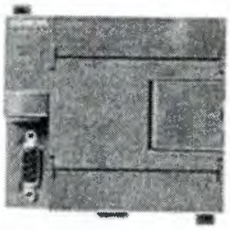
The compact solution that's optimal for first-time users and those needing to change systems. Take the competitive edge and get a kick-start in the right direction with a 221 programmable controller.

Inputs/outputs: 10

Program memory: 4 KByte

Bit processing time: 0.37 µs

S7- 222:



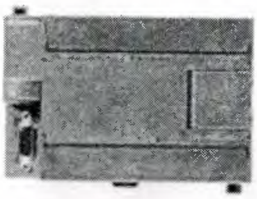
The superior compact solution. Masters all requirements from complex machines right up to small plant solutions.

Inputs/outputs: 14, expandable

Program memory: 4 KByte

Bit processing time: 0.37 µs

S7-224:



The compact high-performance CPU for all those cases where speed, better communication, and more complex programs provide the decisive advantage

Inputs/outputs: 24, expandable

Program memory: 8 KByte

Bit processing time: 0.37 µs

S7-226 & S7-226XM:

The new, compact high-performance CPU. Expandable Inputs/Outputs and additional 485 PPI-interface complex automation tasks.

Inputs/outputs: 40, expandable

226 Program memory: 8 KByte

226XM Program memory: 16 KByte

Additional Data for S7-200 PLCs:

- Simatic Support Pages for S7-200 Hardware FAQ's
- Simatic Support Pages for S7-200 Hardware Manuals
- Simatic Support Pages for S7-200 Hardware Updates (Product Announcements)
- S7-200, the Micro PLC Product Family from Siemens (598 KB)
- Part Numbers for S7-200 PLC's (339 KB)

CHAPTER : 3

TYPES OF PLC

PLCs are separated into two according to their building mechanisms.

3.1 Compact PLCs

Compact PLCs are manufactured such that all units forming the PLC are placed in a case. They are low price PLC with lower capacity. Small or medium size machine manufacturers usually prefer them. In some types compact enlargement module is present.

3.2 Modular PLCs

Combining separate modules together in a board forms them. They can have different memory capacity, I / O numbers, power supply up to the necessary limits.

Some examples: SIEMENS S5-115U, SIEMENS S7-200, MITSUBISHI PC40, TEXAS INSTRUMENT PLC'S, KLOCKNER-MOELLER PS316, OMRON C200H.

3.3 Categories of PLC

The increasing demand from industry for programmable controllers that can be applied to different forms and sizes of control tasks has resulted in most manufacturers producing a range of PLC's with various levels of performance and facilities.

Typical rough definitions of PLC size are given in terms of program memory size and the maximum number of input\output points the system can support. Table 3.1 gives an example of these categories.

Table 3.1 Categories of PLC

PC size	Max I \ O points	Use memory size
Small	40 / 40	1K
Medium	128 / 128	4K
Large	> 128 / > 128	> 4K

Table 3.2 Summary of the S7-200 CPUs

Feature	CPU 212	CPU 214	CPU 215	CPU 216
Physical Size of unit	160mm x 80mm x 62mm	197mm x 80mm x 62mm	218mm x 80mm x 62mm	218mm x 80mm x 62 mm
Memory				
Program (EEPROM)	512 words	2 k words	4 k words	4 k words
User Data	512 words	2 k words	2.5 k words	2.5 k words
Internal Memory Bits	128	256	256	256
Memory Cartridge	None	Yes (EEPROM)	Yes (EEPROM)	Yes (EEPROM)
Optional Battery Cartridge	None	200 Days typical	200 Days typical	200 Days typical
Backup (super capacitor)	50 Hours typical	190 Hours typical	190 Hours typical	190 Hours typical
Inputs/Outputs(I/O)				
Local I/O	8 DI / 6 DQ	14 DI / 10 DQ	14 DI / 10 DQ	24 DI / 16 DQ
Expansion Modules	2 Modules	7 Modules	7 Modules	7 Modules

(max.)				
Process-Image I/O Register	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ
Analog I/O (expansion)	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ
Selectable Input Filter	No	Yes	Yes	Yes

However, to evaluate properly any programmable controller we must consider many dditional features such as its processor, cycle time language facilities, functions, and xpansion capabilities.

A brief outline of the characteristics of small, medium of large programmable controller is iven below, together with typical applications.

3.3.1 Small PLC's

In general, small and 'mini' PLC's are designed as robust, compact units, which can be nounted on or beside the equipment to be controlled. They are mainly used the replaced hard-vired logic relays, timers, counters. That control individual items of plant or machinery, but an also be used to coordinate several machines working in conjunction with each other.

Small programmable controllers can normally have their total I/ O expanded by adding one r two I / O modules, but if any further developments are required this will often mean eplacement of the complete unit. This end of the market is very much concerned with non-pecialist and users, therefore ease of programming and a 'familiar' circuit format are esirable. Competition between manufacturers is extremely fierce in this field, as they vie to btain a maximum share in this partially developed sector of the market.

A single processor is normally used, and programming facilities are kept a fairly basic level, ncluding conventional sequencing controls and simple standard functions: e.g. timers and counters. Programming of small PLC's is by way of logic instruction list (mnemonics) or relay ladder diagrams.

Program storage is given by EPROM or battery-backed RAM. There is now a trend towards EEPROM memory with on-board programming facilities on several controllers.

Table 3.2 Features of a typical small PLC – Mitsubishi F20

Electrical: 240 V a.c. supply;
24 V d.c. on-board for input requirements;
12 input, 8 output points;
LED indicators on all I/ O points;
All I\O Opto-isolated
Choice of output: Relay(240 V 2 a rated)
Triac (240 V 1 A rated)
Transistor (24 V d.c. 1 A)
320 – step memory (CMOS battery-backed RAM)

Programming: Ladder logic or instruction set using hand-held or graphic LCD programmer, with editor, test and monitor facilities;

Facilities: 8 counters, range 1-99 (can be cascaded)
8 timers, range 0.1-99s (can be cascaded)
64 markers\auxiliary relays; can be used individually or in blocks
forming shift registers;
Special function relays;
Jump capability.

3.3.2 Medium-sized PLC'S

In this range modular construction predominates with plug-in modules based around the eurocard 19 inch rack format or another rack mounting system. This construction allows the simple upgrading or expansion of the system. This construction allows the simple upgrading or expansion of the system by fitting additional I/ O cards in to the cards into the rack, since st rack, systems have space for several extra function cards. Boards are usually ggedized' to allow reliable operation over a range of environments.

1 general this type of PLC is applied to logic control tasks that can not be met by small ntrollers due to insufficient I\O provision, or because the control task is likely to be ended in the future. This might require the replacement of a small PLC, where as a dular system can be expanded to a much greater extent, allowing for growth. A medium-ed PLC may therefore be financially more attractive in the long term.

Communications of a single and multi-bit processor are likely within the CPU. For gramming, standard instructions or ladder and logic diagrams are available. Programming normally carried out via a small keypad or a VDU terminal. If different sizes of PLC are chased from a single manufacturer, it is likely that programs and programming panels will compatible between the machines.

.3.3 Large PLC

Vhere control of very large numbers of input and output points is necessary and complex ntrol functions are required, a large programmable controller is the obvious choice. Large C's are designed for use in large plants or on large machines requiring continuous control. ey are also employed as supervisory controllers to monitor and control several other PLC's ntelligent machines. e.g. CNC tools.

Modular construction in Eurocard format is standard, with a wide range of function cards ilable including analogue input output modules. There is a move towards 16-bit processor, l also multi-processor usage in order to efficiently handle a large range of differing control ks.

r example;

- 16-bit processor as main processor for digital arithmetic and text handling.

- Single-bit processor as co-or parallel processor for fast counting, storage etc.
- Peripheral processor for handling additional tasks which are time-dependent or time-critical, such as:

Closed-loop (PID) control
Position controls
Floating-point numerical calculations
Diagnostic and monitoring
Communications for decentralized
Remote input\output racks.

This multi-processor solution optimizes the performance of the overall system as regards versatility and processing speed, allowing to PLC to handle very large programs of 100 K nstructions or more. Memory cards can now provide several megabytes of CMOS RAM or EPROM storage.

3.4 Remote input\output

When large numbers of input / output points are located a considerable distance away from he programmable controller, it is uneconomic to run connecting cables to every point. A olution to this problem is to site a remote I/ O unit near to the desired I/ O points. This acts s a concentrator to monitor all inputs and transmit their status over a single serial ommunications link to the programmable controller. Once output signals have been roduced by the PLC they are feedback along the communications cable to the remote I/ O nit, which converts the serial data into the individual output signals to drive the process.

3.5 Programming large PLC's

Virtually any function can be programmed, using the familiar ladder symbols via a graphics erminal or personal computer. Parameters are passed to relevant modules either by ncorporating constants in to the ladder, or via on screen menus for that module.

There may in addition be computer-oriented languages, which allow programming of unction modules and subroutines.

There is progress towards standardization of programming languages; with programs ecoming easier to over-view through improvement of text handling, hand improved ocumentation facilities. This is assisted by the application of personal computers as vorkstations.

3.6 Developments

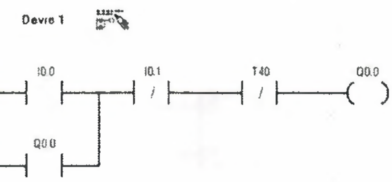
Present trends include the integration of process data from a PLC into management atabases, etc. This allows immediate presentation of information to those involved in cheduling, production and planning.

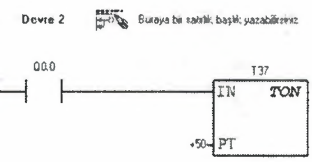
The need to pass process information between PC's and PLC sand other devices within a itomated plants has resulted in the provision of a communications capability on all but the nallest controller. The development of local area networks (LAN) and in particular the cent MAP specification by General Motors (manufacturing automation protocol) provides e communication link to integrate all levels of control systems.

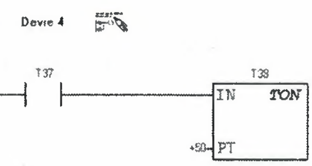
CHAPTER: 4

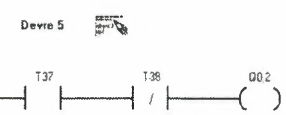
WATER LEVEL CONTROLLING

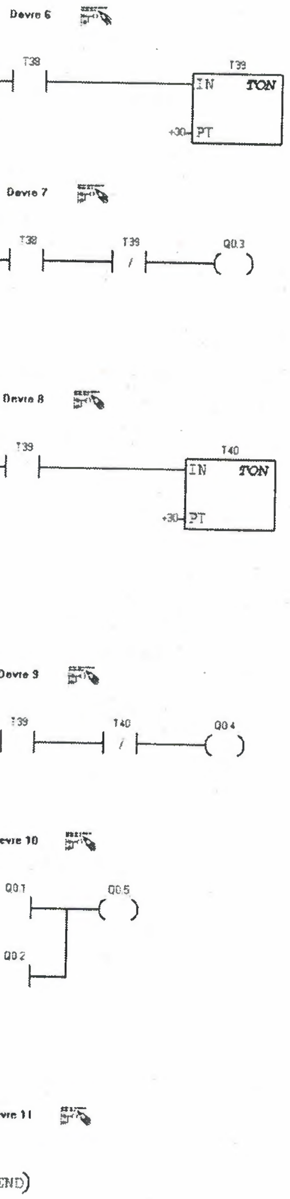
4.1 LADDER DIAGRAM OF PROJECT











STL LIST OF PROGRAM

WORK 1
I0.0
I0.0
I0.1

N T40
Q0.0

ETWORK 2
D I0.0
ON T37, +50

ETWORK 3
D Q0.0
N T37
Q0.1

ETWORK 4
D T37
ON T38, +50

ETWORK 5
T37
N T38
Q0.2

TWORK 6
T38
N T39, +30

TWORK 7
T38
T39
Q0.3

TWORK 8
T39
N T40, +30

TWORK 9
T39
T40
Q0.4

TWORK 10
Q0.1
Q0.2
Q0.5

TWORK 11
ND

WORKING CONDITION OF PROGRAM:

ere the program, when we push the start button, Q0.0 output energizes so the normally n Q 0.0 contact closes so it makes knocking the Q0.0 output

WORK 1
I0.0
I0.0

I0.1
T40
Q0.0

en because of Q0.0 is energized T37 timer starts to operate
VORK 2
I0.0
T37, +50

e 1 starts to operate and let to water flow until timer timing finished, then opens valve
).1) NETWORK 3
Q0.0
T37
Q0.1

 same time Valve2 (Q0.2) starts to operate. During T38 timing Valve 2 operates after
VORK 4
T37
T38, +50

VORK 5
T37
T38
)0.2

me interval finished, mixer(Q0.3) starts to operates immidiatelly depending on T39
ing
VORK 6
T38
T39, +30

VORK 7
T38
T39
0.3

ORK 8
T39
T40, +30
er that Heater (Q0.4) starts to operate until T40 timer timming interval happened
ORK 9
T39
T40
).4

ORK 10
)0.1
0.2
).5

4.4. CONNECTED INSTRUMENTS ON THE PROJECT:

For operate the program I have used following instrments

Name	Number
1. Electromagnetic Valve	2
2. Water Pump	1
3. Mixer	1
4. Heater	1

And Here folowing infomation about these Instruments:

➢ **Electromagnetic Valve:**

An electromagnetic Valve works with 220V-240V , Alternative Current. It has a contact that normally opened, I mean without an applied voltage it does not let to water flow. When we apply the voltage it closes the contact then lets to water flow. Following we can see pictures of double valve and single valve

6

➢ **Water Pump:**

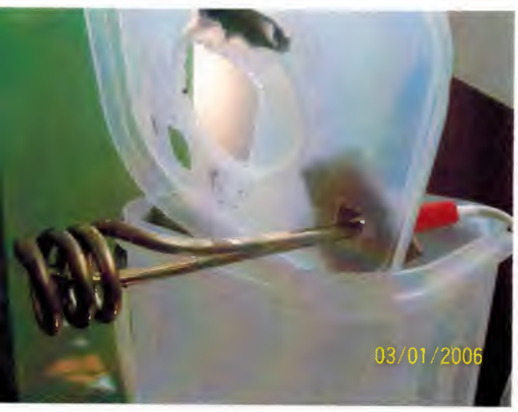
Water Pump helps to water flowing with pressure for send the water to higher place that the plastic can on the project. Here we can see a view of a wate pump.

➢ **Mixer:**

Here in the project, mixer mixes to the water that came from to diffirent valves

➢ **Heater:**

Depending on my condition heater is heatin the water during 3 seconds. You cn see also a view of the heater



CONCLUSION

ne S7-200 is summarized by the following properties:

- Easy entry
- Uncomplicated operation
- Peerless real-time characteristics
- Powerful communications capabilities

In the Laboratory conditions I carried out the operation that is possible. In this eration we controlled the Lamp for this first we connected PC to PLC after this operation : PLC has a 8 inputs, 6 outputs but I used 2 inputs as a I0.0 start button and I0.1 for stop ton, 4 outputs as a Q0.1 for valve 1, Q0. 2 for valve2, Q0.3 for mixer, Q0. 4 for heater I ote a PLC program when I loaded the program to PLC at same time we connected the PLC

For operate the valve we need to pressured water, that I used in Mechanical orotary. After water flowed to plastic jerry can PLC sends information to mixer, after to neater.

We can improve these operations by using the program that is comprehensively in lding Automation and Industrial Automations.

Over the world most of the new technologies are using by that investigate applications 1 PLC, we can say that advertisement panels, Building Automations, and Industrial omations it can be given as an example to this application.

REFERENCES

[1] SIMATIC S7-200 Programmable Controller, System Manual

[2] Özgür Cemal Özerdem, Programmable Logic Controller and Programming, Near East University Press, Lefkoşa 2002

[3] Information about Siemens S-7 200 Micro PLC.
‘‘*http://www.sea.siemens.com/controls/product/s7200/CNs7200.htm*’’

[4] Programmable Logic Controller Recep ÇETİN