NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

Programable Logic Controller (PLC)

Graduation Project
EE – 400

Student:Hakan Engin (991059)

Superviser:Asst.Professor
Özgür Cemal Özerdem

Nicosia - 2007

TABLE OF CONTENTS

CHAPTER 11

ACKNOWLEDGMENTS

First I want to thank Asst. Prof. Özgür Cemal Özerdem to be my adviser.Under his guiadance I successfully overcome many difficulties and learn about Programble Logic Controller (PLC).He explained my questions everytime.He gave me the PLC device for making practise in my home.Also he helped me at the laboratory many times and we worked the elevator together.At the other hand all semester he gave me motivation and we talked about everyting in stead of lectures.

Special thanks to Samet Biricik.With his kind help,I started the PLC device succesfully.He spend his time at the weekends for me.

I also want to thank my friends in NEU; Fuat,Remzi,Fatma,Ömer being with them make my 6 years in NEU full of fun.

Finally,I want to thank my family,especially my parents.Without their endless support and love for me.I wish my mother lives happily always,and my father in the heaven be proud of me.

ABSTRACT

Industry has begun to recognize the need for quality improvement and increase in productivity in the sixties and seventies. Flexibility also became a major concern (ability to change a process quickly became very important in order to satisfy consumer needs).

Try to imagine automated industrial production line in the sixties and seventies. There was always a huge electrical board for system controls, and not infrequently it covered an entire wall! Within this board there was a great number of interconnected electromechanical relays to make the whole system work. By word "connected" it was understood that electrician had to connect all relays manually using wires! An engineer would design logic for a system, and electricians would receive a schematic outline of logic that they had to implement with relays. These relay schemas often contained hundreds of relays. The plan that electrician was given was called "ladder schematic". Ladder displayed all switches, sensors, motors, valves, relays, etc. found in the system. Electrician's job was to connect them all together. One of the problems with this type of control was that it was based on mechanical relays. Mechanical instruments were usually the weakest connection in the system due to their moveable parts that could wear out. If one relay stopped working, electrician would have to examine an entire system (system would be out until a cause of the problem was found and corrected).

The other problem with this type of control was in the system's break period when a system had to be turned off, so connections could be made on the electrical board. If a firm decided to change the order of operations (make even a small change), it would turn out to be a major expense and a loss of production time until a system was functional again.

It's not hard to imagine an engineer who makes a few small errors during his project. It is also conceivable that electrician has made a few mistakes in connecting the system. Finally, you can also imagine having a few bad components. The only way to see if everything is all right is to run the system. As systems are usually not perfect with a first try, finding errors was an arduous process. You should also keep in mind that a product could not be made during these corrections and changes in connections. System had to be literally disabled before changes were to be performed. That meant that the entire production staff in that line of production was out of work until the system was fixed up again. Only when electrician was done finding

errors and repairing,, the system was ready for production. Expenditures for this kind of work were too great even for well-to-do companies.

All control and automation studies that have been in industrial applications are the result of the PLC technic.PLC taked its place during the groving up of industrial control.The electronic control systems first started with analog controllers.But after a while they were insufficient.Because of being insufficient it crossed to digital controllers.Digital systems have been more active role in industry because of being faster and making a lot of functions with a small volume.But the essential development had started with programable digital systems and the control with microprocessor.

During the time different companies produced PLCs at various capacities.The companies Mitsubishi and Toshiba produced small types of PLCs.But Siemens,Omron,Allen-Bradley,General Electric and Westinghouse had produced larger PLCs and they extended their systems.

INTRODUCTION

A Programmable Logic Controller, PLC, or Programmable Controller or Logic Box is an electronic device used for automation of industrial processes, such as control of machinery on factory assembly lines. Unlike general-purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or non-volatile memory. A PLC is an example of a real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result.

The main difference from other computers is that PLC are armored for severe condition (dust, moisture, heat, cold, etc) and have the facility for extensive input/output (I/O) arrangements. These connect the PLC to sensors and actuators. PLCs read limit switches, analog process variables (such as temperature and pressure), and the positions of complex positioning systems. Some even use machine vision. On the actuator side, PLCs operate electric motors, pneumatic or hydraulic cylinders, magnetic relays or solenoids, or analog outputs. The input/output arrangements may be built into a simple PLC, or the PLC may have external I/O modules attached to a computer network that plugs into the PLC.

PLCs were invented as replacements for automated systems that would use hundreds or thousands of relays, cam timers, and drum sequencers. Often, a single PLC can be programmed to replace thousands of relays. Programmable controllers were initially adopted by the automotive manufacturing industry, where software revision replaced the re-wiring of hard-wired control panels when production models changed.

Many of the earliest PLCs expressed all decision making logic in simple ladder logic which appeared similar to electrical schematic diagrams. The electricians were quite able to trace out circuit problems with schematic diagrams using ladder logic. This program notation was chosen to reduce training demands for the existing technicians. Other early PLCs used a form of instruction list programming, based on a stack-based logic solver.

The functionality of the PLC has evolved over the years to include sequential relay control, motion control, process control, distributed control systems and networking. The data handling, storage, processing power and communication capabilities of some modern PLCs are approximately equivalent to desktop computers. PLC-like programming combined with remote I/O hardware, allow a general-purpose desktop computer to overlap some PLCs in certain applications.

Under the IEC 61131-3 standard, PLCs can be programmed using standards-based programming languages. A graphical programming notation called Sequential Function Charts is available on certain programmable controllers.

CHAPTER 1

1 PLC HISTORY

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

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

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

In the mid70's the dominant PLC technologies were sequencer state-machines and the bit-slice based CPU. The AMD 2901 and 2903 were quite popular in Modicon and A-B PLCs. Conventional microprocessors lacked the power to quickly solve PLC logic in all but the smallest PLCs. As conventional microprocessors evolved, larger and larger PLCs were being based upon them. However, even today some are still based upon the 2903. (ref A-B's PLC-3) Modicon has yet to build a faster PLC than 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 PLCs and they could be far away from the actual machine they were controlling. They could also now be used to send and receive varying voltages to allow them to enter the analog world. Unfortunately, the lack of standardization coupled with continually changing technology has made PLC communications a nightmare of incompatible protocols and physical networks.

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

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

1.1 Usage Areas Of Plcs

In general PLCs are planned for to use in industry.PLCs are setting out the functions which are writen in digital principles.They check the system groups with system cards and it accomadates counters,timers for to checking.PLCs can be used in feedback control systems after the aritmetical operations are added later.PLC system can check all activities on the field and after that takes its memory by its software.After logical operations the information acted to the field by the elements which are control by the PLC:These information signals can be analog or digital.These signals may come from a transducer or conductor.If the information that coming is analog,the signal can be analyse by a space.If the signal is digital analyse can be made by happen or not happen situation.These feeling activities are made by input cards and interference activities are made by output cards.

The system which will be controlled by PLC can be shown different sizes.We can make only one control of a machine or we can make a control of all factory.The only difference between two system is the capacity of PLC.Todays PLCs are at all sectors.Chemical sector,food sector,production lines,storage systems,markets and rafineries are the example of it.Belong to the development in the electronic sector,PLC technology has been groe up fastly.Because of this all tecnicers have to know something about PLC.

If go to the near past,PLCs were not beeing used much.Because there are two reasons.Microprocessors were very expensive.But after beeing cheaper the cost productivity has been grow up.

We can add PLC application in two groups;General and industrial applications are also beeing used in process industry.PLCs are born in first automotive sector and still it is going like this.Also there are five application field of PLCs by the automotive and food penetrating industry.These are;

1. Sequence Control:PLCs are used a lot in this kind of applications.These applications are very closed to relay workings.In independent machines,conveyors and packing machines are the example of this kind.

2. Movement Control:This is the complete of linear and rotating movement control systems with PLC.We can give metal cutting,metal shaping and fitting machines for an example.

3.Process Control: This application is about the temperature,pressure,momentum,weight parameters.These application needs I/O analog.With the start of PID software,PLC taked on the process of single loop controllers.The example of this is plastic injection machines,repeatable heating ovens and all batch-controls.

4. Datum Control:Collecting datums with PLC is beeing developed last times.

1.2 Advantages of plcs

The same, as well as more complex tasks, can be done with a PLC. Wiring between devices and relay contacts is done in the PLC program. Hard-wiring, though still required to connect field devices, is less intensive. Modifying the application and correcting errors are easier to handle. It is easier to create and change a program in a PLC than it is to wire and rewire a circuit.

Following are just a few of the advantages of PLCs:

- Smaller physical size than hard-wire solutions.
- Easier and faster to make changes.
- PLCs have integrated diagnostics and override functions.
- Diagnostics are centrally available.
- Applications can be immediately documented.
- Applications can be duplicated faster and less expensively.

Siemens PLCs Siemens makes several PLC product lines in the SIMATIC® S7 family. They are: S7-200, S7-300, and S7-400.

S7-200 The S7-200 is referred to as a micro PLC because of its small size. The S7-200 has a brick design which means that the power supply and I/O are on-board. The S7-200 can be used on smaller, stand-alone applications such as elevators, car washes, or mixing machines. It can also be used on more complex industrial applications such as bottling and packaging machines.

S7-300 and S7-400 The S7-300 and S7-400 PLCs are used in more complex applications that support a greater number of I/O points. Both PLCs are modular and expandable. The power supply and I/O consist of separate modules connected to the CPU. Choosing either the S7-300 or S7-400 depends on the complexity of the task and possible future expansion. Your Siemens sales representative can provide you with additional information on any of the Siemens PLCs.

In briefly we can add all advantages at these under groups;

1. **Physical Sizes**:If we compare with their work,PLCs are very small equipments and they don't cover much places.This property supply a perfect usage at every field.

2. **Safety:**Towards to the danger,all equiments are beeing saved.

3. **Cost**:Because of first investments and suppling production benefit,their costs are unimportant.

4. **Environmental resistance:**PLCs are planned especially for industrial environmets so that they show strong abilities in these fields.

5. **Communication Ability:**PLCs can make communication with each other,with computers and with other intellegent devices.

6. **Complex Structure**:PLCs can make the control of a lot of machines at the same time.

7. **Flexibility**:We can make changes in the PLC programs whenever we want and also we can enlarge the memories.

8. **Operation Speed**:PLC can operate the operations that include logical and arithmetical operations.

9. **Image Producing**:We can watch the operations directly from a monitor.Also we can make a failure scan.

1.2.1 The differences between other systems

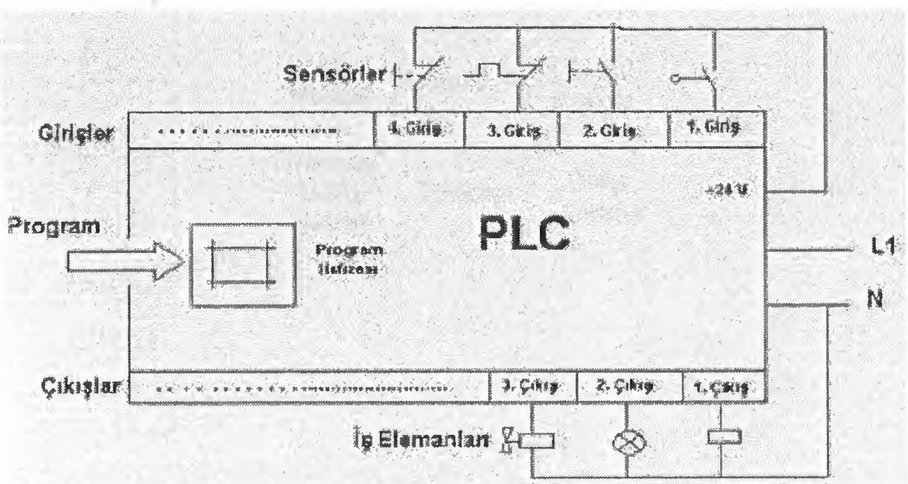
1. We can make high capacity otomation with PLCs.
2. If we are making little control,foundation investments are higher in PLCs.
3. The system with PLCs can work without upkeep much time.
4. The time between failures is more then 8000 hours in relay systems.
5. The systems with PLCs can be adapted to the new systems easier then relay systems.

6. PLC’s cover little spaces and they spend low energy.

CHAPTER 2

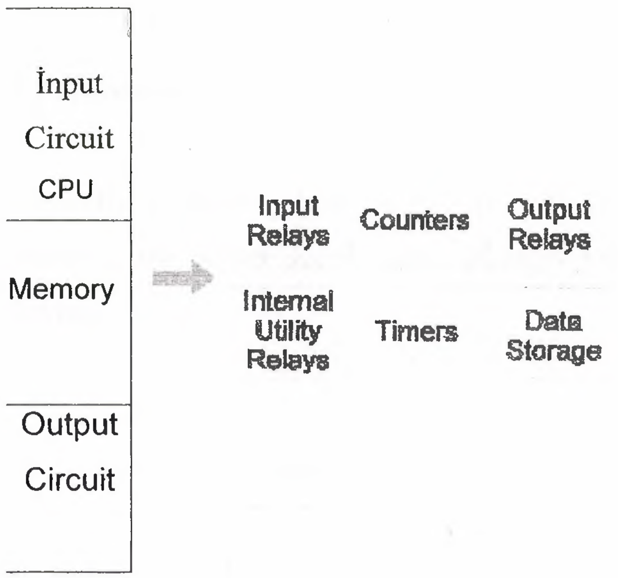
2 PLC INSIDE

The PLC mainly consists of a CPU, memory areas, and appropriate circuits to receive input/output data. We can actually consider the PLC to be a box full of hundreds or thousands of separate relays, counters, timers and data storage locations. Do these counters, timers, etc. really exist? No, they don't "physically" exist but rather they are simulated and can be considered software counters, timers, etc. These internal relays are simulated through bit locations in registers. (more on that later)



2.1 The programmable logic controller

A programmable logic controller (PLC) is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes (Figure 1 .3) and are designed to be operated by engineers with perhaps a limited knowledge of computers and computing languages. They are not designed so that only computer programmers can set up or change the programs. Thus, the designers of the PLC have pre-programmed it so that the control program can be entered using a simple, rather intuitive, form of language. The term logic is used because programming is primarily concerned with implementing logic and switching operations, e.g. if A or B occurs switch on C, if A and B occurs switch on D. Input devices, e.g. sensors such as switches, and output devices in the system being controlled, e.g. motors, valves, etc., are connected to the PLC. The operator then enters a sequence of instructions, i.e. a program, into the memory of the PLC. The controller then monitors the inputs and outputs according to this program and carries out the control rules for which it has been programmed.



PLCs have the great advantage that the same basic controller can be used with a wide range of control systems. To modify a control system and the rules that are to be used, all that is necessary is for an operator to key in a different set of instructions. There is no need to rewire. The result is a flexible, cost effective, system, which can be used with control systems, which vary quite widely in their nature and complexity.

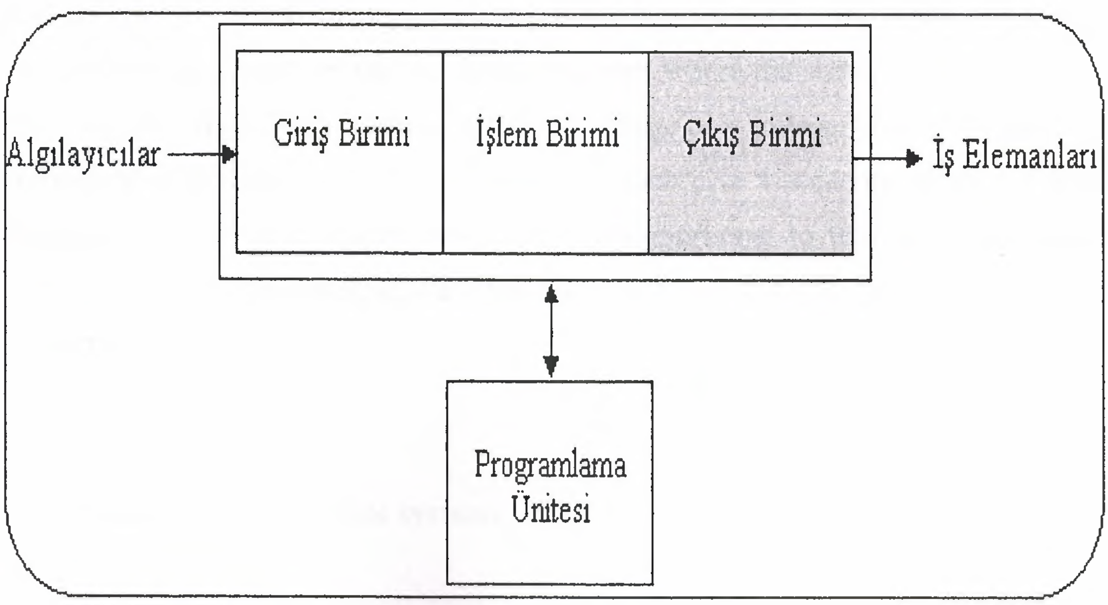
PLCs are similar to computers but whereas computers are optimized for calculation and display tasks, PLCs are optimized for control tasks and the industrial environment. Thus PLCs are:

1. Rugged and designed to withstand vibrations, temperature, humidity and noise.
2. Have interfacing for inputs and outputs already inside the controller.
3. Are easily programmed and have an easily understood programming language which is primarily concerned with logic and switching operations.

The first PLC was developed in 1969. They are now widely used and extend from small self-contained units for use with perhaps 20 digital inputs/outputs to modular systems which can be used for large numbers of inputs/outputs, handle digital or analogue inputs/outputs, and also carry out proportional-integral-derivative control modes.

2.2 Hardware

Typically a PLC system as five basic components. These are the processor unit, memory, the power supply unit, input/output interface section and the programming device.



1.The processor unit or central processing unit (CPU) is the unit containing the microprocessor and this interprets the input signals and carries out the control actions, according to the program stored in its memory, communicating the decisions as action signals to the output

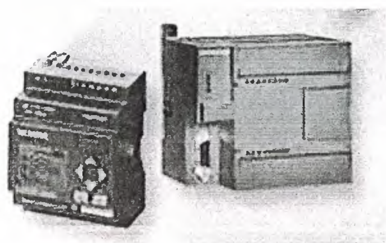
2. The power supply unit is needed to convert the mains a.c. voltage to the low d.c. voltage (5 V) necessary for the processor and the circuits in the input and output interface modules.

3.The programming device is used to enter the required program into the memory of the processor. The program is developed in the device and then transferred to the memory unit of the PLC.
4. The memory unit is where the program is stored that is to be used for the control actions to
be exercised by the microprocessor.

5.The input and output sections are where the processor receives information from external devices and communicates information to external devices. The inputs might thus be from switches, as illustrated in with the automatic drill, or other sensors such as photo- electric cells, as in the counter mechanism temperature sensors, or flow sensors, etc. The outputs might be to motor starter coils, solenoid valves, etc. Input and output devices can be classified as giving signals which are discrete, digital or analogue Devices giving discrete or digital signals are ones where the signals are. Thus a switch is a device giving a discrete signal, either no voltage or a voltage. Digital devices can be considered to be essentially discrete devices, which give a sequence of on_off signals. Analogue devices give signals whose size is proportional to the size of the variable being monitored. For example, a temperature sensor may give a voltage proportional to the temperature.

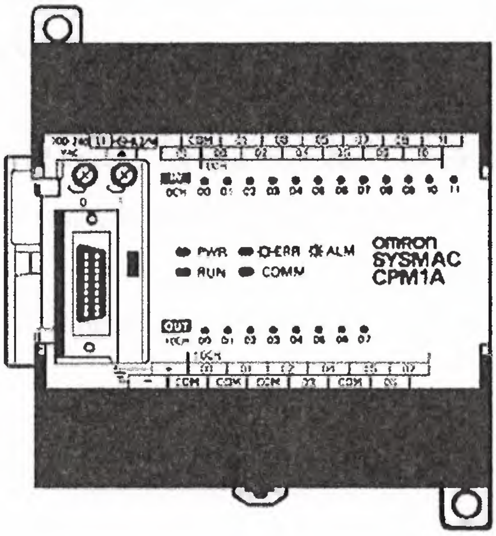
2.3 Mechanical design of plc systems

There are two common types of mechanical design for PLC systems; a single box, and the modular and rack types. The single box type is commonly used for small programmable controllers and is supplied as an integral compact package complete with power supply, processor, memory, and input/output units . Typically such a PLC might have 40 input/output points and a memory, which can store some 300 to 1000 instructions. The modular type consists of separate modules for power supply, processor, etc. which are often mounted on rails within a metal cabinet. The rack type can be used for all sizes of programmable controllers and has the various functional units packaged in individual modules, which can be plugged into sockets in a base rack. The user and the appropriate ones then plugged into the rack decide the mix of modules required for a particular purpose. Thus it is comparatively easy to expand the number of input/output connections by just adding more input/output modules or to expand the memory by adding more memory units.



Programs are entered into a PLC's memory using a program device, which is usually not permanently connected to a particular PLC and can be moved from one controller to the next without disturbing operations. For the operation of the PLC it is not necessary for the programming device to be connected to the PLC since it transfers the program to the PLC memory. Programming devices can be a hand-held device, a desktop console or a computer. Hand-held systems incorporate a small keyboard and liquid crystal display, showing a typical form. Desktop devices are likely to have a visual display unit with a full keyboard and screen display. Personal computers are widely configured as program development workstations. Some PLCs only require the computer to have appropriate software, others special communication cards to interface with the PLC. A major advantage of using a computer is that the program can be stored on the hard disk or a floppy disk and copies easily made. The disadvantage is that the programming often tends to be not so user-friendly. Hand-held programming consoles will normally contain enough memory to allow the unit to retain programs while being carried from one place to another.

Only when the program has been designed on the programming device and is ready is it transferred to the memory unit of the PLC.



2.3.1 Internal Architecture

Figure shows the basic internal architecture of a PLC. It consists of a central processing unit(CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between I and 8 MIIz. This frequency determines the operating speed of the PLC and provides the timing and synchronisation for all elements in the system. The information within the PLC is carried by means of digital signals. The internal paths along which digital signals flow are called buses. In the physical sense, a bus is just a number of conductors along which electrical signals can flow. It might be tracks on a printed circuit board or wires in a ribbon cable. The CPU uses the data bus for sending data between the constituent elements, the address bus to send the addresses of locations for accessing stored data and the control bus for signals relating to internal control actions. The system bus is used for communications between the input/output ports and the input/output unit.

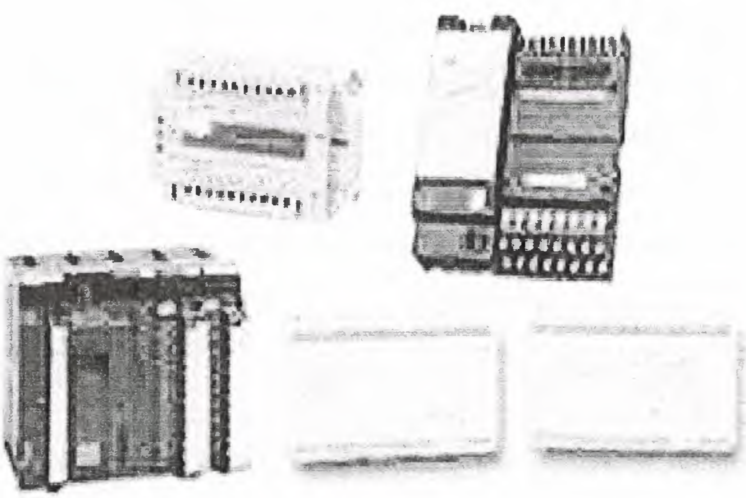
2.3.2 The CPU

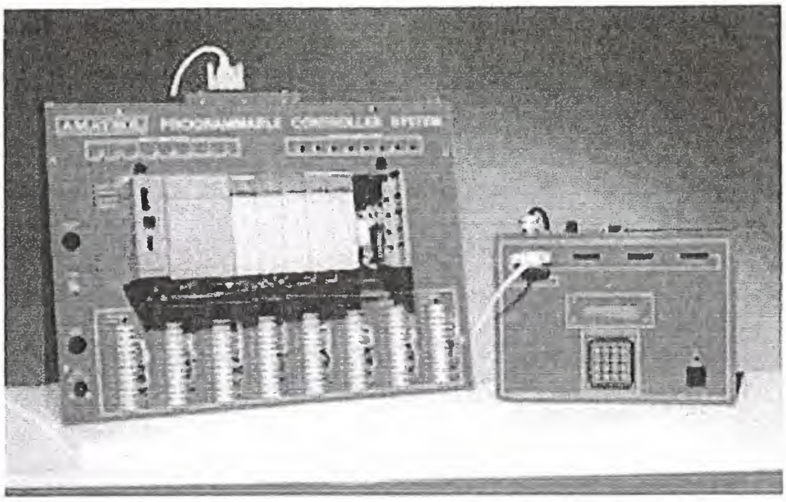
The internal structure of the CPU depends on the microprocessor concerned. In general they have:

1. An arithmetic and logic unit (ALU) which is responsible for data manipulation and carrying out arithmetic operations of addition and subtraction and logic operations of AND, OR, NOT and EXCLUSIVE OR.

2. Memory, termed registers, located within the microprocessor and used to store information involved in program execution.

3. A control unit, which is used to control the timing of operations.





2.3.3 The Buses

The buses are the paths used for communication within the PLC. The information is transmitted in binary form, i.e. as a a group of bits with a bit being a binary digit of 1 or 0, i.e. on/off states. The term word is used for the group of bits constituting some information. Thus an 8-bit word might be the binary number 00100110. Each of the bits is communicated simultaneously along its own parallel wire. The system has four buses:

1. The data bus carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus, which can handle 8-bit numbers. It can thus perform operations between 8-bit numbers and deliver results as S-bit values.

2. The address bus is used to carry the addresses of memory locations. So that each word can be located in the memory, every memory location is given a unique address. Just like houses in a town are each given a distinct address so that they can be located. so each word location is given an address so that data stored at a particular location can be accessed by the CPU either to read data located there or put, i.e. write, data there. It is the address bus, which carries the information indicating which address is to be accessed. If the address bus consists of 8 lines, the number of 8-bit words, and hence number of distinct addresses, is 2^8 = 256. With 16 address lines, 65 536 addresses are possible.

3. The control bus carries the signals used by the CPU for control, e.g. to inform

memory devices whether they are to receive data from an input or output data and to carry timing signals used to synchronize actions.

4. The system bus is used for communications between the input/output ports and the input/output unit

2.3.4 Memory

There are several memory elements in a PLC system

1. System read-only-memory (ROM) to give permanent storage for the operating system and fixed data used by the CPU.

2. Random-access-memory (RAM) for the user's program.

3. Random-access-memory (RAM) for data. This is where information is stored on the status of input and output devices and the values of timers and counters and other internal devices.

The data RAM is sometimes referred to as a data table or register table. Part of this memory, i.e. a block of addresses, will be set aside for input and output addresses and the states of those inputs and outputs. Part will be set aside for preset data and part for storing counter values, timer values, etc.

4. Possibly, as a bolt-on extra module, erasable and programmable read-only-memory (EPROM) for ROMS that can be programmed and then the program made permanent.

The programs and data in RAM can be changed by the user. All PLCs will have some amount of RAM to store programs that have been developed by the user and program data. However, to prevent the loss of programs when the power supply is switched off, a battery is used in the PLC to maintain the RAM contents for a period of time. After a program has been developed in RAM it may be loaded into an EPROM memory chip, often a bolt-on module to the PLC, and so made permanent. In addition there are temporary buffer stores for the input/output channels.

The number of binary words that it can store determines the storage capacity of a memory unit. Thus, if a memory size is 256 words then it can store 256 x 8 = 2048 bits if 8-bit words are used and 256 x 16= 4096 bits if 16-bit words are used. Memory sizes are often specified in terms of the number of storage locations available with 1K representing the number 2^10, i.e. 1024. Manufacturers supply memory chips with the storage locations grouped in groups of 1, 4 and 8 bits. A 4K x 1 memory has 4 X I x 1024 bit locations. A 4K x 8 memory has 4 x 8 x 1024 bit locations. The term byte is used for a word of length 8 bits. Thus the 4K x 8 memory can store 4096 bytes. With a 16-bit address bus we can have 216 different addresses and so, with 8-bit words stored at each address, we can have 216 x 8 storage locations and so use a memory of size 216 x 8/210 = 64K x 8 which we might have in the form of four 16K x 8 bit memory chips

2.3.5 Input/output unit

The input/output unit provides the interface between the system and the outside world, allowing for connections to be made through input/output channels to input devices such as sensors and output devices such as motors and solenoids. It is also through the input/output unit that programs are entered from a program panel. Every input/output point has a unique address, which can be used by the CPU.

The input/output channels provide isolation and signal conditioning functions so that sensors and actuators can often be directly connected to them without the need for other circuitry. Electrical isolation from the external world is usually by means of optoisolators (the term optocoupler is also often used). shows the principle of an optoisolator. When a digital pulse passes through the light-emitting diode, a pulse of infrared radiation is produced. This pulse is detected by the phototransistor and gives rise to a voltage in that circuit. The gap between the light-emitting diode and the photo transistor gives electrical isolation but the arrangement still allows for a digital pulse in one circuit to give rise to a digital pulse in another circuit. The digital signal that is generally compatible with the microprocessor in the PLC is 5 V d.c. However, signal conditioning in the input channel, with isolation, enables a wide range of input signals to be supplied to it. A range of inputs might be available with a larger PLC, e.g. 5 V, 24 V, 110 V and 240 V digital/discrete, i.e. on-off, signals. A small PLC is likely to have just one form of input, e.g. 24 V.shows the basic form a d.c. input channel might take.Outputs are often specified as being of relay type, transistor type or triac type.

1. With the relay type, the signal from the PLC output is used to operate a relay and so is able to switch currents of the order of a few amperes in an external circuit. The relay not only allows small currents to switch much larger currents but also isolates the PLC from the external circuit. Relays are, however, relatively slow to operate. Relay outputs are suitable for ac. and d.c. switching. They can withstand high surge currents and voltage transients. the basic feature of a relay output.

2. The transistor type of output uses a transistor to switch current through the external circuit. This gives a considerably faster switching action. It is, however, strictly for d.c. switching and is destroyed by over current and high reverse voltage. As a protection, either a fuse or built-in electronic protection is used. Optoisolators are used to provide isolation. shows the basic form of such a transistor output channel. Basic form of transistor output

3. Triac outputs, with optoisolators for isolation, can be used to control external loads, which are connected to the ac. power supply. It is strictly for ac. operation and is very easily destroyed by over current. Fuses are virtually always included to protect such outputs.

The output from the input/output unit will be digital with a level of 5 V. However, after signal conditioning with relays, transistors or triacs, the output from the output channel might be a 24 V, 100 mA switching signal, a d.c. voltage of 110 V, 1 A or perhaps 240 V, 1 A ac. or 240 V, 2 A a.c. from a triac output channel ..With a small PLC, all the outputs might be of one type, e.g. 240 V a.c., 1 A. With modular PLCs, however, a range of outputs can be accommodated by selection of the modules to be used.

The following illustrates the types of inputs and outputs available with a small PLC, one of the Mitsubishi F2 series:

Number of inputs 12

Number of outputs 8

CHAPTER 3

3 Input-Output Devices

3.1 Input Devices

Sensors which give digital/discrete, i.e. on-off, outputs can be easily connected to the input ports of PLCs. Sensors, which give analogue signals, have to be converted to digital signals before inputting them to PLC ports. The following are examples of some of the commonly used sensors.

3.1.1. Mechanical switches

A mechanical switch generates an on-off signal or signals as a result of some mechanical input causing the switch to open or close. Such a switch might be used to indicate the presence of a work piece on a machining table, the work piece pressing against the switch and so closing it. The switch being open and its presence indicate the absence of the workpiece by it being closed. Thus, with the arrangement shown in Figure 2.1(a), the input signals to a single input channel of the PLC are thus the logic levels:

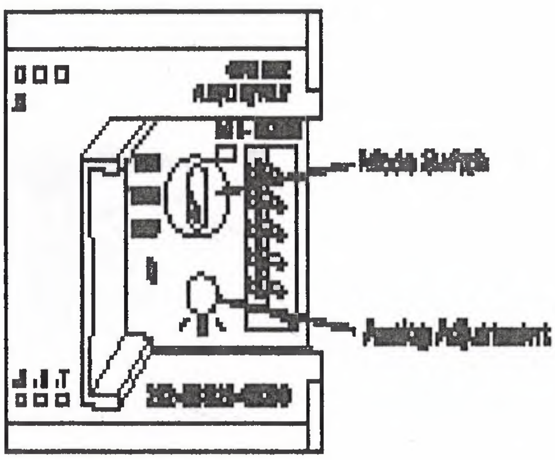
Work piece not present 0
Work piece present 1

The 1 level might correspond to a 24 V d.c. Input, the 0 to a 0 V input..When the switch is open the supply voltage is applied to the PLC input, when the switch is closed channel the input voltage drops to a low value. The logic levels are thus:

Supply Work piece not present 1
Work piece present 0

Switches are available with normally open (NO) or normally closed (NC) contacts or can be configured as either by choice of the relevant contacts. A switch has its contacts open in the absence of a mechanical input and the mechanical input is used to close the switch. An NC switch has its contacts closed in the absence of a mechanical input and the mechanical input is used to open the switch.

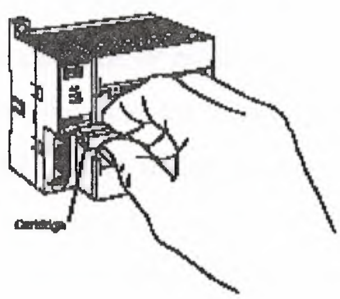
The term limit switch is used for a switch, which is used to detect the presence or passage of a moving part. It can be actuated by a cam. roller or lever. shows . The cam can be rotated at a constant rate and so switch the switch on and off for particular time intervals.



3.1.2 Proximity switches

Proximity switches are used to detect the presence of an item without making contact with it. There are a number of forms of such switches, some being only suitable for metallic objects.

The eddy current type of proximity switch has a coil, which is energized by a constant alternating current and produces a constant alternating magnetic field. When a metallic object is close to it, eddy currents are induced in it .The magnetic field due to these eddy currents induces an e.m.f. back in the coil with the result that the voltage amplitude needed to maintain the constant coil current changes. The voltage amplitude is thus a measure of the proximity of metallic objects. The voltage can be used to activate an electronic switch circuit, basically' a transistor which has its output switched from low to high by the voltage change, and so give an on-off device. The range over which such objects can be detected is typically about 0.5 to 20 mm. Another type, the inductive proximity switch, consists of a coil wound round a ferrous metallic core. When one end of this core is placed near to a ferrous metal object there is effectively a change in the amount of metallic core associated with the coil and so a change in its inductance. This change in inductance can be monitored using a resonant circuit, the presence of the ferrous metal object thus changing the current in that circuit. The current can be used to activate an electronic switch circuit and so give an on-off device. The range over which such objects can be detected is typically about 2 to 15 mm.



Another type is the reed switch. This consists of two overlapping, but not touching, strips of a springy ferromagnetic material sealed in a glass or plastic envelope. When a magnet or current carrying coil is brought close to the switch, the strips become magnetized and attract each other. The contacts then close. The magnet closes the contacts when it is typically about 1 mm from the switch. Such a switch is widely used with burglar alarms to detect when a door is opened; the magnet being in the door and the reed switch in the frame of the door. When the door opens the switch opens.

A proximity switch that can be used with metallic and non-metallic objects is the capacitive proximity switch. The capacitance of a pair of plates separated by some distance depends on the separation, the smaller the separation the higher the capacitance. The sensor of the capacitive proximity switch is just one of the plates of the capacitor, the other plate being the metal object whose proximity is to be detected Thus the proximity of the object is detected by a change in capacitance. The sensor can also be used to detect non-metallic objects since the capacitance of a capacitor depends on the dielectric between its plates. In this case the plates are the sensor and the earth and the non-metallic object is the dielectric. The change in capacitance can be used to activate an electronic

Switch circuit and so give an on-off device: Capacitive proximity switches can be used to detect objects when they are typically between 4 and 60 mm from the sensor head.

3.1.3 Photoelectric sensors and switches

Photoelectric switch devices can either operate as transmissive types where the object being detected breaks a beam of light, usually infrared radiation, and stops it reaching the detector or reflective types where the object being detected reflects a beam of light onto the detector.In both types the radiation emitter is generally a light-emitting diode (LED). The radiation detector might be a phototransistor, often a pair of transistors, known as a Darlington pair. The Darlington pair increases the sensitivity. Depending on the circuit used, the output can be made to switch to either high or low when light strikes the transistor. Such sensors are supplied as packages for sensing the presence of objects at close range, typically at less than about 5 mm. a U-shaped form where the object breaks the light beam.

Another possibility is a photo diode. Depending on the circuit used, the output can be made to switch to either high or low when light strikes the diode. Yet another possibility is a photo conductive cell. The resistance of the photo conductive cell, often cadmium sulphide, depends on the intensity of the light falling on it.

With the above sensors, light is converted to a current, voltage or resistance change. If the output is to be used as a measure of the intensity of the light, rather than just the presence or absence of some object in the light path, the signal will need amplification and then conversion from analogue to digital by an analogue-to-digital converter. An alternative to this is to use a light-to-frequency converter, the light then being converted to a sequence of pulses with the frequency of the pulses being a measure of the light intensity. Integrated circuit sensors are available, e.g. the Texas Instrument TSL22O, incorporating the light sensor and the voltage-to-frequency converter.

3.1.4 Encoders

The term encoder is used for a device that provides a digital output as a result of angular or linear displacement. An increment encoder detects changes in angular or linear displacement from some datum position, while an absolute encoder gives the actual angular or linear position.

A beam of light, from perhaps a light-emitting diode (LED), passes through slots in a disc and is detected by a light sensor, e.g. a photo diode or photo transistor. When the disc rotates, the light beam is alternately transmitted and stopped and so a pulsed output is produced from the light sensor. The number of pulses is proportional to the angle through which the disc has rotated, the resolution being proportional to the number of slots on a disc. With 60 slots then, since one revolution is a rotation of 3600, a movement from one slot to the next is a rotation of 60. By using offset slots it is possible to have over a thousand slots for one revolution and so much higher resolution.

The absolute encoder differs from the incremental encoder in having a pattern of slots, which uniquely defines each angular position. Figure 2.9 shows the form of such an encoder using three sets of slots and so giving a 3-bit output. Typical encoders tend to have up to 10 or 12 tracks. The number of bits in the resulting binary output is equal to the number of tracks. Thus with 3 tracks there will be 3 bits and so the number of positions that can be detected is 2^3= 8, i.e. a resolution of 360/8 =45 degree With 10 tracks there will be 10 bits and the number of positions that can be detected is 2^10= 1024 and the angular resolution is 360/1024 0.35.

3.1.5. Temperature sensors

A simple form of temperature sensor, which can be used to provide an on-off signal when a particular temperature is reached, is the bimetal element. This consists of two strips of different metals, e.g. brass and iron, bonded together..The two metals have different coefficients of expansion. Thus when the temperature of the bimetal strip increases the strip curves, in order that one of the metals can expand more than the other. The higher expansion metal is on the outside of the curve. As the strip cools, the bending effect is reversed. This movement of the strip can be used to make or break electrical contacts and hence, at some particular temperature, give an on-off current in an electrical circuit. The device is not very accurate but is commonly used in domestic central heating thermostats.

Another form of temperature sensor is the resistive temperature detector (RTD). The electrical resistance of metals or semiconductors changes with temperature. In the case of a metal, the ones most commonly used are platinum, nickel or nickel alloys, the resistance of which varies in a linear manner with temperature over a wide range of temperatures, though the actual change in resistance per degree is fairly small. Semiconductors, such as thermistors, show very large changes in resistance with temperature. The change, however, is non-linear. Such detectors can be used as one arm of a Wheatstone bridge and the output of the bridge taken as a measure of the temperature.Another possibility is to use a potential divider circuit with the change in resistance of the thermistor changing the voltage drop across a resistor.The output from either type of circuit is an analogue signal which is a measure of the temperature.

Thermodiodes and thermotransistors are used as temperature sensors since the temperature affects the rate at which electrons and holes diffuse across semiconductor junctions. Integrated circuits are available which combine such a temperature sensitive element with the relevant circuitry to give an output voltage related to temperature. A widely used integrated package is the LM35, which gives an output of 10 mV/^{0}C when the supply voltage is +5 V

A digital temperature switch can be produced with an analogue sensor by feeding the analogue output into a comparator amplifier which compares it with some set value, producing an output giving a logic 1 signal when the temperature voltage input is equal to or greater than the set point and otherwise an output which gives a logic 0 signal. Integrated circuits, e.g. LM3911N, are available combining a thermotransistor temperature sensitive element with an operational amplifier. When the connections to the chip are so made that the amplifier is connected as a comparator then the output will switch as the temperature traverses the set point and so directly give an on-off temperature controller.

Another commonly used temperature sensor is the thermocouple. The thermocouple consists essentially of two dissimilar wires A and B forming a junction .When the junction is heated so that it is at a higher temperature than the other junctions in the circuit, which remain at a constant cold temperature, an e.m.f is produced which is related to the hot junction temperature. The voltage produced by a thermocouple is small and needs amplification before it can be fed to the analogue channel input of a PLC. There is also circuitry required to compensate for the temperature of the cold junction since its temperature affects the value of the e.m.f given by the hot junction. The amplification and compensation, together with filters to reduce the effect of interference from the 50 Hz mains supply, are often combined in a signal-processing unit.

3.1.6 Displacement sensors

A linear or rotary potentiometer can be used to provide a voltage signal related to the position of the sliding contact between the ends of the potentiometer resistance track. The potentiometer thus provides an analogue linear or angular position sensor.

Another form is displacement sensor is the linear variable differential transformer (LVDT), this giving a voltage output related to the position of a ferrous rod. The LVDT consists of three symmetrically placed coils through which the ferrous rod moves.

When an alternating current is applied to the primary coil, alternating voltages are induced in the two secondary coils. When the ferrous rod core is centred between the two secondary coils, the voltages induced in them are equal. The outputs from the two secondary coils are connected so that their combined output is the difference between the two voltages. With the rod central, the two alternating voltages are equal and so there is no output voltage. When the rod is displaced from its central position there is more of the rod in one secondary coil than the other. As a result the size of the alternating voltage induced in one coil is greater than that in the other. The difference between the two secondary coil voltages, i.e. the output, thus depends on the position of the ferrous rod. The output from the LVDT is an alternating voltage. This is usually converted to an analogue d.c. voltage and amplified before inputting to the analogue channel of a PLC.

3.1.7 Strain gauges

When a wire or strip of semiconductor is stretched, its resistance changes. The fractional change in resistance is proportional to the fractional change in length, i.e. strain. where AR is the change in resistance for a wire of resistance .R and G is a constant called the gauge factor. For metals the gauge factor is about 2 and for semiconductors about 100. Metal resistance strain gauges are in the put form of a flat coil in order to get a reasonable length of metal in a small age area. Often they are etched from metal foil and attached to a backing of thin plastic film so that they can be stock on surfaces, like postage stamps on an envelope.

The change in resistance of the strain gauge, when subject to strain, is usually converted into a voltage signal by the use of a Wheatstone bridge.A problem that occurs is that the resistance of the strain gauge also changes with temperature and thus some means of temperature compensation has to be used so that the output of the bridge is only a function of the strain. This can be achieved by placing a dummy strain gauge in an opposite arm of the bridge, that gauge not being subject to any strain but only the temperature.

An alternative which is widely used is to use four active gauges as the arms of the bridge and arrange it so that one pair of opposite gauges are in tension and the other pair in compression. This not only gives temperature compensation but also gives a much larger output change when strain is applied. The following paragraph illustrates systems employing such a form of compensation. By attaching strain gauges to other devices, changes, which result in strain of those devices, can be transformed, by the strain gauges, to give voltage changes. They might, for example, be attached to a cantilever to which forces are applied at its free end..The voltage change, resulting from the strain gauges and the Wheatstone bridge, then becomes a measure of the force. Another possibility is to attach strain gauges to a diaphragm which deforms as a result of pressure.The output from the gauges, and associated Wheatstone bridge, then becomes a measure of the pressure.

3.1.8 Pressure sensors

Commonly used pressure sensors, which give responses related to the pressure, are diaphragm and bellows types. The diaphragm type consists of a thin disk of metal or plastic, secured round its edges. When there is a pressure difference between the two sides of the diaphragm, the centre of it deflects. The amount of deflection is related to the pressure difference. This deflection may be detected by strain gauges attached to the diaphragm or by using the deflection to squeeze a piezoelectric crystal..When a piezoelectric crystal is squeezed, there is a relative displacement of positive and negative charges within the crystal and the outer surfaces of the crystal become charged. Hence a potential difference appears across it. An example of such a sensor is the Motorola MPX100AP sensor .

This has a built-in vacuum on one side of the diaphragm and so the deflection of the diaphragm gives a measure of the absolute pressure applied to the other side of the diaphragm. The output is a voltage, which is proportional to the applied pressure with a sensitivity of 0.6 mV/kPa. Other versions are available which have one side of the diaphragm open to the atmosphere and so can be used to measure gauge pressure, others allow pressures to be applied to both sides of the diaphragm and so can be used to measure differential pressures.

Pressure switches are designed to switch on or off at a particular pressure. A typical form involves a diaphragm or bellows, which moves under the action of the pressure and operates a mechanical switch. two possible forms. Diaphragms are less sensitive than bellows but can withstand greater pressures.

3.1.9 Liquid level detector

Pressure sensors may be used to monitor the depth of a liquid in a tank. The pressure due to a height of liquid h above some level is hpg, where p is the density of the liquid and the acceleration due to gravity. Thus a commonly used method of determining the level of liquid in a tank is to measure the pressure due to the liquid above some datum level .Often a sensor is just required to give a signal when the level in some container reaches a particular level. A float switch that is used for this purpose consists of a float containing a magnet, which moves in a housing with a reed switch. As the float rises of falls it turns the reed switch on or off, the reed switch being connected in a circuit, which then switches on or off a voltage.

3.1.10 Fluid flow measurement

A common form of fluid flow meter is that based on measuring the difference in pressure resulting when a fluid flows through a constriction..As a result of the fluid flowing through the orifice, the pressure at A is higher than that at B, the difference in pressure being a measure of the rate of flow. This pressure difference can be monitored by means of a diaphragm pressure gauge and thus becomes a measure of the rate of flow.

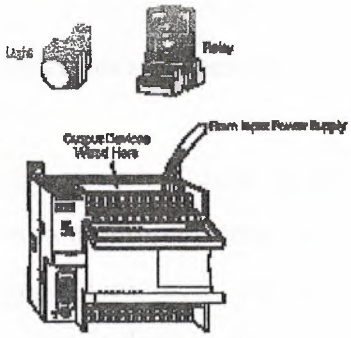
3.1.11 Keypads

Many machines employ keypads to input instructions to set the conditions required for outputs such as temperatures or speeds. Such keypads commonly have buttons which, when pressed, operate conductive silicon rubber pads to make contacts.Rather than have each key wired up separately and so giving 12 inputs, the keys are connected in rows and columns and closing a single key can give a column output and a row output, which is unique to that key. This reduces the number of inputs required to the PLC.

3.1.12 Output devices

The output ports of a PLC are of the relay type or optoisolator with transistor or triac types depending on the devices connected to them, which are to be switched on or off Generally, the digital signal from an output channel of a PLC is used to control an actuator, which in turn controls some process. The term actuator is used for the device, which transforms the electrical signal into some more powerful action, which then results in the control of the process.

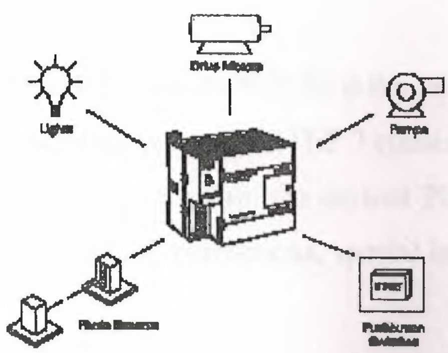
The following are some examples;



3.1.13 Condactor

Solenoids form the basis of a number of output control actuators. When a current passes through a solenoid a magnetic field is produced and this can then attract ferrous metal components in its vicinity. One example of such an actuator is the contactor. When the output from the PLC is switched on, the solenoid magnetic field is produced and pulls on the contacts and so closes a switch or switches.The result is that much larger currents can be switched on. Thus the contactor might be used to switch on the current to a motor.

Essentially a contactor is a form of relay, the difference being that the term relay is used for a device for switching small currents, less than about 10 A, whereas the term contactor is used for a heavy current switching device with currents up to many hundreds of amps.



3.1.14 Directional control valves

Another example of the use of a solenoid as an actuator is a solenoid operated valve. The valve may be used to control the directions of flow of pressurized air or oil and so used to operate other devices such as a piston moving in a cylinder.

Pressurized air or hydraulic fluid is inputted from port P, this being connected to the pressure supply from a pump or compressor and port T is connected to allow hydraulic fluid to return to the supply tank or, in the case of a pneumatic system, to vent the air to the atmosphere. With no current through the solenoid the hydraulic fluid of pressurized air is fed to the right of the piston and exhausted from the left, the result then being the movement of the piston to the left. When a current is passed through the solenoid, the spool valve switches the hydraulic fluid or Pressurised air to the left of the piston and exhausted from the right. The piston them moves to the right. The movement of the piston might be used to push a deflector to deflect items off a conveyor belt or implement some other form of displacement, which requires power.

Direction valves can be used to control the direction of motion of pistons in cylinders, the displacement of the pistons being used to implement the required actions. The term single acting cylinder is used for one which is powered by the pressurised fluid being applied to one side of the piston to give motion in one direction, it being returned in the other direction by possibly an internal spring. The term double acting cylinder is used when the cylinder is powered by fluid for its motion in both piston movement directions. shows how a valve can be used to control the direction of motion of a piston in a single-acting cylinder;.

CHAPTER 4

4 PRORAMMING PLC

STEP 7-Micro/WIN32 is the program software used with the S7-200 PLC to create the PLC operating program. STEP 7 consists of a number of instructions that must be arranged in a logical order to obtain the desired PLC operation. These instructions are divided into three groups: standard instructions, special instructions, and high-speed instructions.

Standard Instructions Standard instructions consists of instructions that are found in most programs. Standard instructions include; timer, counter,math,logical,increment/decrement/invert, move, and block instructions.

Special Instructions Special instructions are used to manipulate data. Special instructions include shift, table, find, conversion, for/next, and real-time instructions.

High-Speed Instructions High-speed instructions allow for events and interrupts to occur

Independent of the PLC scan time. These include high-speed counters, interrupts, output, and transmit instructions.It is not the purpose of this text to explain all of the instructions and capabilities. A few of the more common instructions necessary for a basic understanding of PLC operation will be discussed. PLC operation is limited only by the hardware capabilities and the ingenuity of the person programming it.
Refer to the **SIMATIC S7-200 Programmable Controller System Manual** for detailed information concerning these instructions.

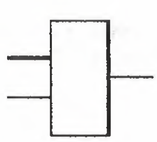
Micro/WIN32 The programming software can be run Off-line or On-line. Offline programming allows the user to edit the ladder diagram and perform a number of maintenance tasks. The PLC does not need to be connected to the programming device in this
mode. On-line programming requires the PLC to be connected to the programming device. In this mode program changes are downloaded to the PLC. In addition, status of the input/output
elements can be monitored. The CPU can be started, stopped, or reset.

Symbols In order to understand the instructions a PLC is to carry out, an understanding of the language is necessary. The language of PLC ladder logic consists of a commonly used set of symbols that represent control components and instructions.

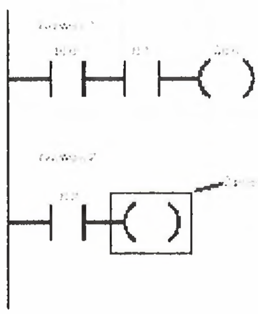
Contacts One of the most confusing aspects of PLC programming for first-time users is the relationship between the device that controls a status bit and the programming function that uses a status bit. Two of the most common programming functions are the normally open (NO) contact and the normally closed (NC) contact. Symbolically, power flows through these contacts when they are closed. The normally open contact (NO) is true (closed) when the input or output status bit controlling the contact is 1. The normally closed contact (NC) is true (closed) when the input or output status bit controlling the contact is 0.

Coils Coils represent relays that are energized when power flows to them. When a coil is energized, it causes a corresponding output to turn on by changing the state of the status bit controlling that output to 1. That same output status bit may be used to control normally open and normally closed contacts elsewhere in the program.

Boxes Boxes represent various instructions or functions that are executed when power flows to the box. Typical box functions are timers, counters, and math operations.

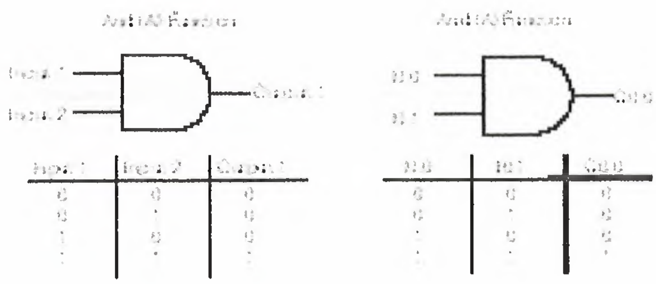


Entering Elements Control elements are entered in the ladder diagram by positioning the cursor and selecting the element from a lists. In the following example the cursor has been placed in the position to the right of I0.2. A coil was selected from a pulldown list and inserted in this position.

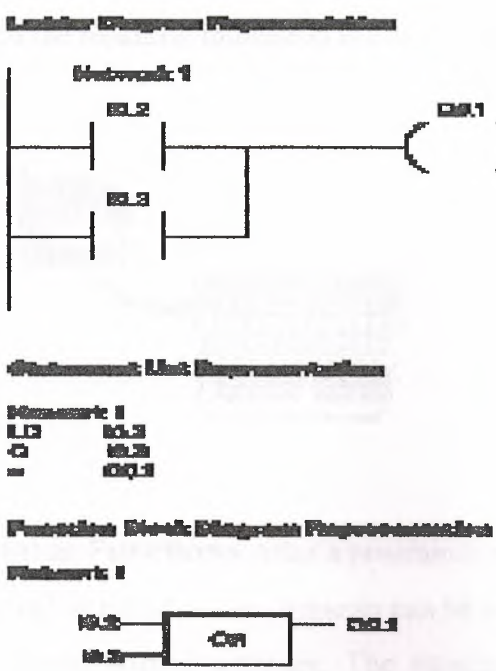


An AND Operation Each rung or network on a ladder represents a logic operation. The following programming example demonstrates an AND operation. Two contact closures and one output coil are placed on network 1. They were assigned addresses I0.0, I0.1, and Q0.0. Note that in the statement list a new logic operation always begins with a load instruction (LD). In this example I0.0 (input 1) and (A in the statement list) I0.1 (input 2) must be true in order for output Q0.0 (output 1) to be true. It can also be seen That I0.0 and I0.1 must be true for Q0.0 to be true by looking at the function block diagram representation.

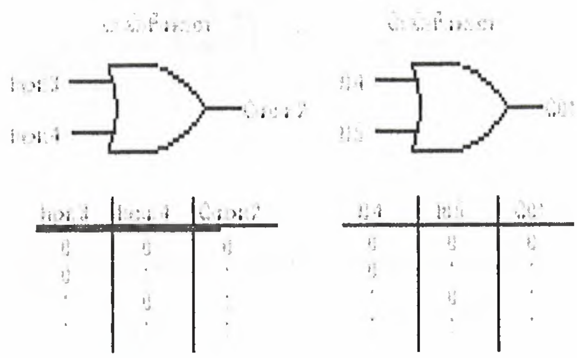
Another way to see how an AND function works is with a Boolean logic diagram. In Boolean logic an AND gate is represented by a number of inputs on the left side. In this case there are two inputs. The output is represented on the right side. It can be seen from the table that both inputs must be a logic 1 in order for the output to be a logic 1.



An OR Operation In this example an OR operation is used in network 1. It can be seen that if either input I0.2 (input 3) or (O in the statement list) input I0.3 (input 4), or both are true, then output Q0.1 (output 2) will be true.

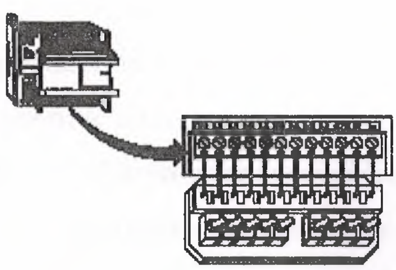


Another way to see how an OR function works is with a Boolean logic diagram. The symbol differs slightly from an AND function. The OR function is represented by a number of inputs on the left side. In this case there are two inputs. The output is represented on the right side. It can be seen from the table that any input can be a logic 1 in order for the output to be a logic 1.

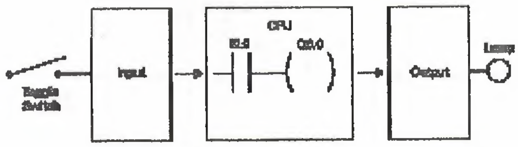


4.1 Testing a Program

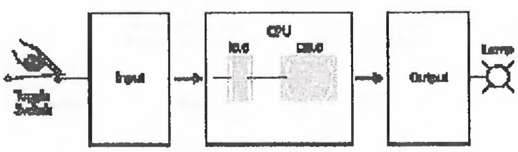
Once a program has been written it needs to be tested and debugged. One way this can be done is to simulate the field inputs with an input simulator, such as the one made for the S7-200. The program is first downloaded from the programming device to the CPU. The selector switch is placed in the RUN position. The simulator switches are operated and the resulting indication is observed on the output status indicator lamps.



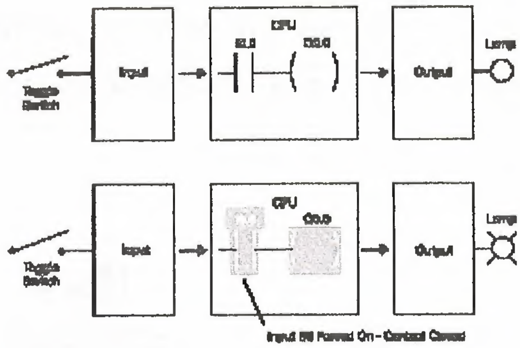
Status Functions After a program has been loaded and is running in the PLC, the actual status of ladder elements can be monitored using STEP 7 Micro/WIN32 software. The standard method of showing a ladder element is by indicating the circuit condition it produces when the device is in the deenergized or non operated state. In the following illustration input 1 (I0.0) is programmed as a normally open (NO) contact. In this condition, power will not flow through the contacts to the output (Q0.0).



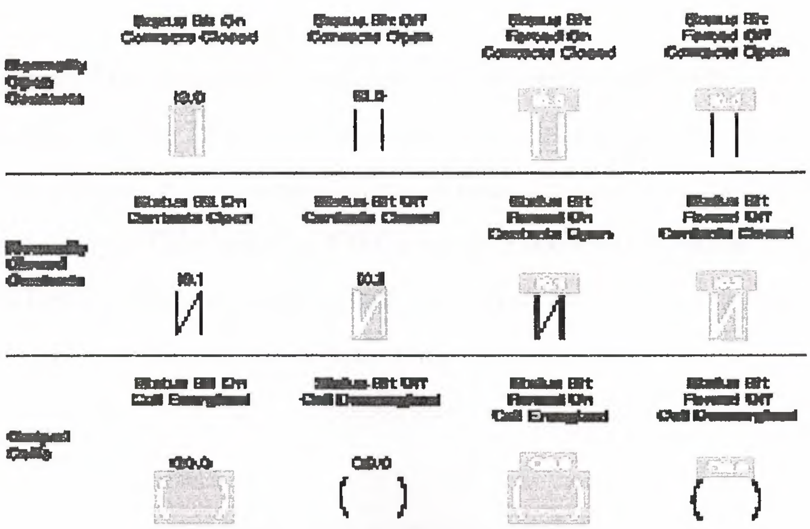
When viewing the ladder diagram in the status mode, control elements that are active, or true (logic 1), are highlighted. In the example shown the toggle switch connected to input 1 has been closed. Power can now flow through the control element associated with input 1 (I0.0) and activate the output (Q0.0). The lamp will illuminate.



Forcing Forcing is another useful tool in the commissioning of an application. It can be used to temporarily override the input or output status of the application in order to test and debug the program. The force function can also be used to override discrete output points. The force function can be used to skip portions of a program by enabling a jump instruction with a forced memory bit. Under normal circumstances the toggle switch, shown in the illustration below, would have to be closed to enable input 1 (I0.0) and turn on the output light. Forcing enables input 1 even though the input toggle switch is open. With input 1 forced high the output light will illuminate. When a function is forced the control bit identifier is highlighted. The element is also highlighted because it is on.

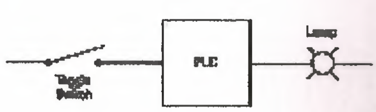


The following table shows the appearance of ladder elements in the Off, forced, and On condition.

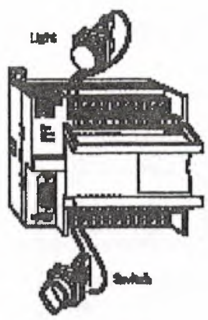


4.2 Discrete Inputs/Outputs

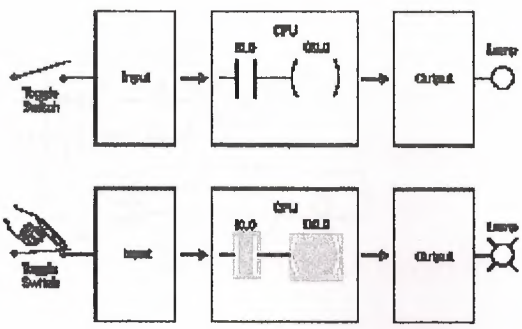
To understand discrete control of a programmable controller the same simple lamp circuit illustrated with forcing will be used. This is only for instructional purposes as a circuit this simple would not require a programmable controller. In this example the lamp is off when the switch is open and on when the switch is closed.



Wiring To accomplish this task, a switch is wired to the input of the PLC and an indicator light is wired to output terminal.

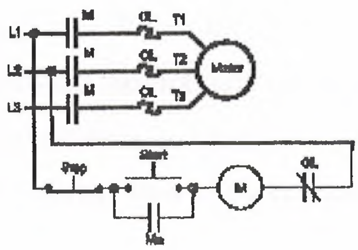


The following drawing illustrates the sequence of events. A switch is wired to the input module of the PLC. A lamp is wired to the output module. The program is in the CPU. The CPU scans the inputs. When it finds the switch open I0.0 receives a binary 0. This instructs Q0.0 to send a binary 0 to the output module. The lamp is off. When it finds the switch closed I0.0 receives a binary 1. This instructs Q0.0 to send a binary 1 to the output module, turning on the lamp.

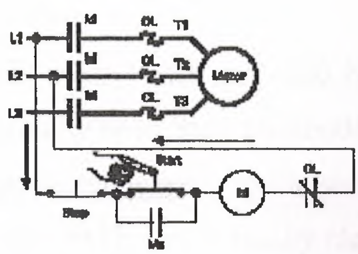


Program Instruction When the switch is open the CPU receives a logic 0 from input I0.0. The CPU sends a logic 0 to output Q0.0 and the light is off.

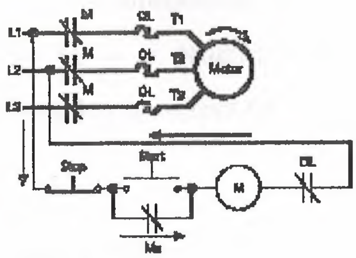
When the switch is closed the CPU receives a logic 1 from input I0.0. The CPU sends a logic 1 to output Q0.0, thus activating Q0.0. The light turns on.



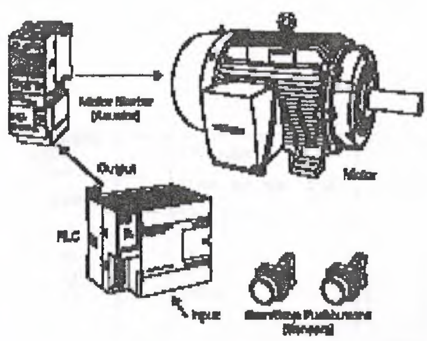
Motor Starter Example The following example involves a motor start and stop circuit. The line diagram illustrates how a normally open and a normally closed pushbutton might be used in a control circuit. In this example a motor started (M) is wired in series with a normally open momentary pushbutton (Start), a normally closed momentary pushbutton (Stop), and the normally closed contacts of an overload relay (OL).



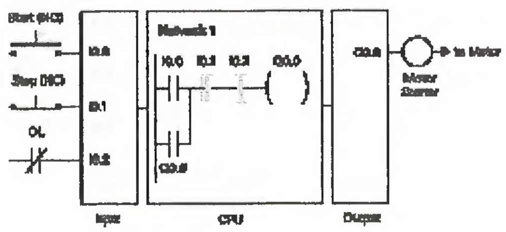
Momentarily depressing the Start pushbutton completes the path of current flow and energizes the motor starter (M).



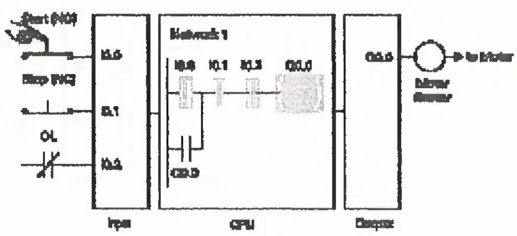
This closes the associated M and Ma (auxiliary contact located in the motor starter) contacts. When the Start button is released a holding circuit exists to the M contactor through the auxiliary contacts Ma. The motor will run until the normally closed Stop button is depressed, or the overload relay opens the OL contacts, breaking the path of current flow to the motor starter and opening the associated M and Ma contacts.



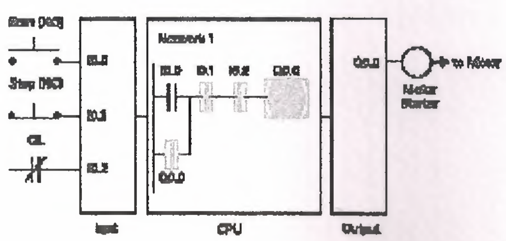
This control task can also be accomplished with a PLC. **Program Instruction** A normally open Start pushbutton is wired to the first input (I0.0), a normally closed Stop pushbutton is wired to the second input (I0.1), and normally closed overload relay contacts (part of the motor starter) are connected to the third input (I0.2). The first input (I0.0), second input (I0.1), and third input (I0.2) form an AND circuit and are used to control normally open programming function contacts on Network 1. I0.1 status bit is a logic 1 because the normally closed (NC) Stop Pushbutton is closed. I0.2 status bit is a logic 1 because the normally closed (NC) overload relay (OL) contacts are closed. Output Q0.0 is also programmed on Network 1. In addition, a normally open set of contacts associated with Q0.0 is programmed on Network 1 to form an OR circuit. A motor starter is connected to output Q0.0.



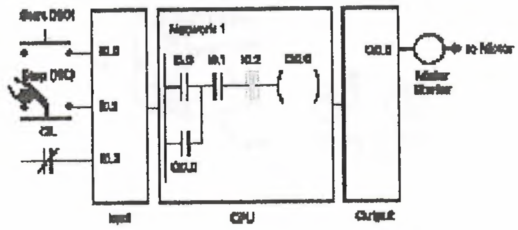
When the Start pushbutton is depressed the CPU receives a logic 1 from input I0.0. This causes the I0.0 contact to close. All three inputs are now a logic 1. The CPU sends a logic 1 to output Q0.0. The motor starter is energized and the motor starts.



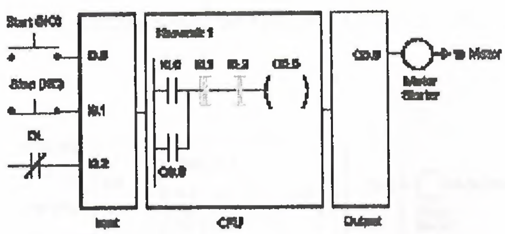
When the Start pushbutton is pressed, output Q0.0 is now true and on the next scan, when normally open contact Q0.0 is solved, the contact will close and output Q0.0 will stay on even if the Start pushbutton has been released.



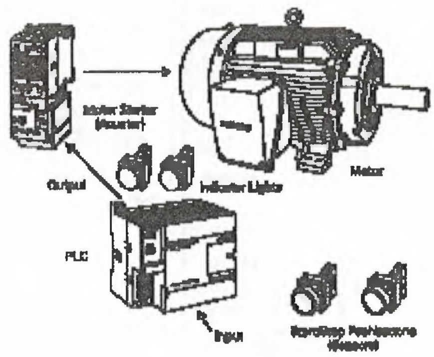
The motor will continue to run until the Stop pushbutton is depressed. Input I0.1 will now be a logic 0 (false). The CPU will send a binary 0 to output Q0.0. The motor will turn off.



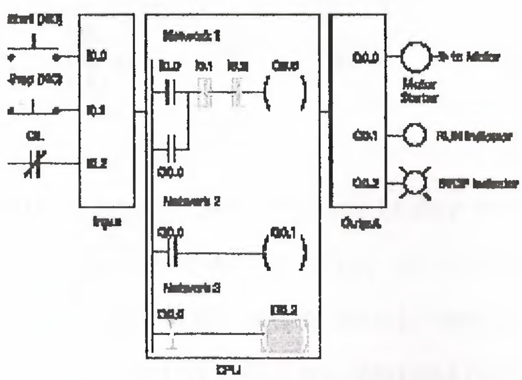
When the Stop pushbutton is released I0.1 logic function will again be true and the program ready for the next time the Start pushbutton is pressed.



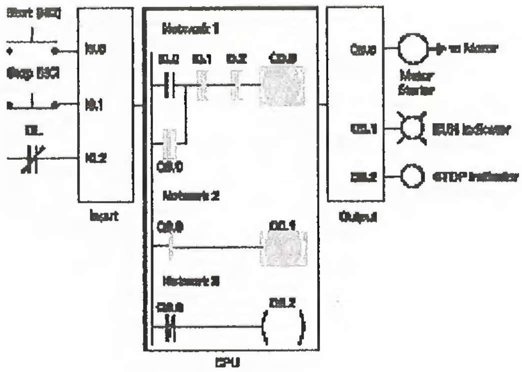
Expanding the Application The application can be easily expanded to include indicator lights for RUN and STOP conditions. In this example a RUN indicator light is connected to output Q0.1 and a STOP indicator light is connected to output Q0.2.



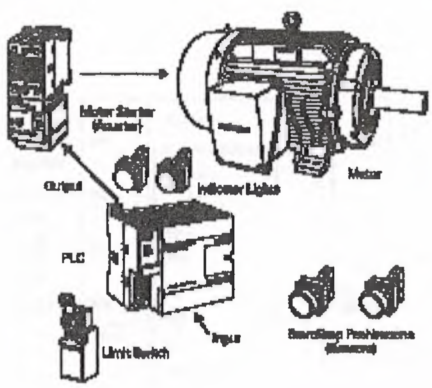
It can be seen from the ladder logic that a normally open output Q0.0 is connected on Network 2 to output Q0.1 and a normally closed Q0.0 contact is connected to output Q0.2 on network 3. In a stopped condition output Q0.0 is off. The normally open Q0.0 contacts on Network 2 are open and the RUN indicator, connected to output Q0.1 light is off. The normally closed Q0.1 on Network 3 lights are closed and the STOP indicator light, connected to output Q0.2 is on.



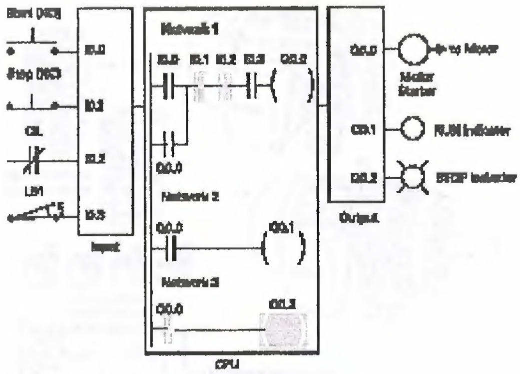
When the PLC starts the motor output Q0.0 is now a logic high (On). The normally open Q0.0 contacts on Network 2 now switch to a logic 1 (closed) and output Q0.1 turns the RUN indicator on. The normally closed Q0.0 contacts on Network 3 switch to a logic 0 (open) and the STOP indicator light connected to output Q0.2 is now off.



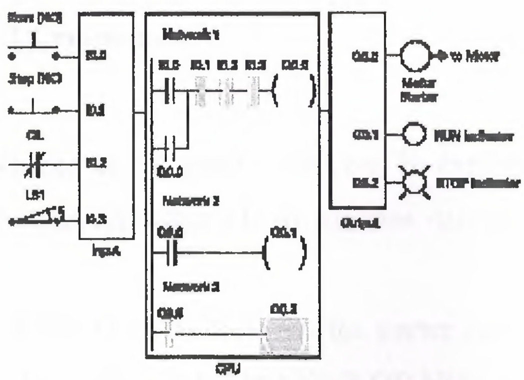
Adding a Limit Switch The application can be further expanded by adding a limit switch with normally open contacts to input I0.3.



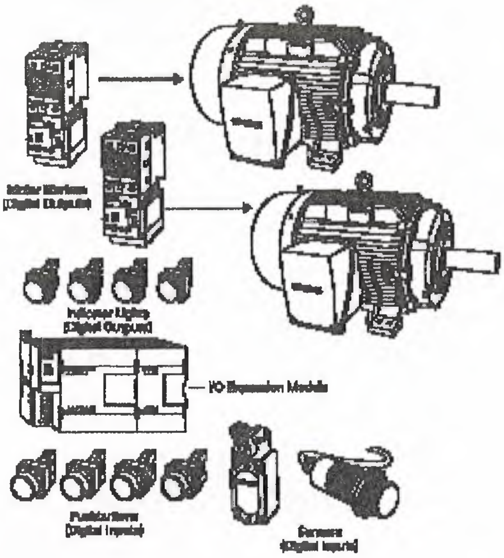
A limit switch could be used to stop the motor or prevent the motor from being started. An access door to the motor, or its associated equipment, is one example of a limit switch's use. If the access door is open, the normally open contacts of LS1 connected to input I0.3 are open and the motor will not start.



When the access door is closed, the normally open contacts on the limit switch (LS1) are closed. Input I0.3 is now on (logic 1), and the motor will start when the Start pushbutton is pressed.



Expansion The PLC program can be expanded to accommodate many commercial and industrial applications. Additional Start/Stop pushbuttons and indicator lights can be added for remote operation, or control of a second motor starter and motor. Overtravel limit switches can be added along with proximity switches for sensing object position. In addition, expansion modules can be added to further increase the I/O capability. The applications are only limited by the number of I/Os and amount of memory available on PLC.



4.3 Commands

Whichever a control circuit can be explain with a logical function.There are three group of commands that a logic function can be operate;

1. LOAD,LOAD NOT are the starter commands.
2. AND,OR,NOT,AND NOT,OR NOT are the operation commands.
3. AN BLOCK,OR BLOCK are the end of the operation commands.
4. OUT is the appoint to the end command.

The table below belongs to various PLCs commands;

KOMUT	Hitachi	Omron	Mitsubishi	Texas ins.	Simatic S7
LOAD	LD	LD	LD	STR	LD
AND	AND	AND	AND	AND	A
OR	OR	OR	OR	OR	O
NOT	NOT	NOT	I	NOT	NOT
LOAD NOT	LDI	LD NOT	LDI	STR NOT	LDN
AND NOT	ANI	AND NOT	ANI	AND NOT	AN
OR NOT	ORI	OR NOT	ORI	OR NOT	ON
AND BLOCK	ANB	AN LD	ANB	AND STR	ALD
OR BLOCK	ORB	OR LD	ORB	OR STR	OLD
OUT	OUT	OUT	OUT	OUT	=
END	END	END	END	END	MEND

We can add timers,counters and checking the program commands to these. Pay attention to these subjects when programming;

1. The signal flow in the PLC control circuit is from left to right.

2. Any elements could not be connect to the distributer directly.

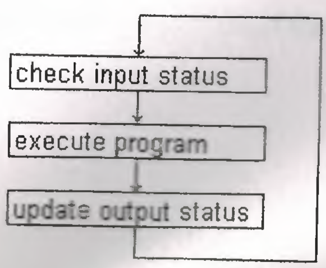
3. We can not connect a short circuit after a relay coil.

4. We can connect more then one relay coil parallel.

5. We have to learn the short circuit and coil numbers from the PLC guide.

4.4 Plc operation

A PLC works by continually scanning a program. We can think of this scan cycle as consisting of 3 important steps. There are typically more than 3 but we can focus on the important parts and not worry about the others. Typically the others are checking the system and updating the current internal counter and timer values.



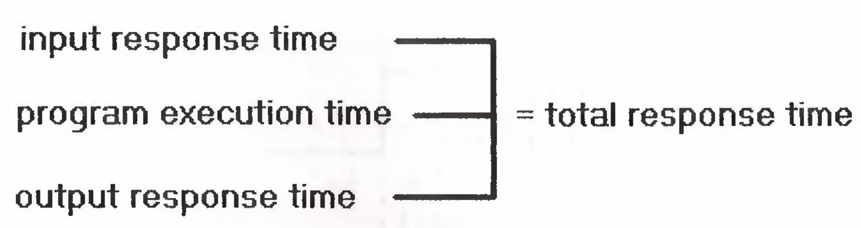
Step 1-CHECK INPUT STATUS-First the PLC takes a look at each input to determine if it is on or off. In other words, is the sensor connected to the first input on? How about the second input? How about the third... It records this data into its memory to be used during the next step.

Step 2-EXECUTE PROGRAM-Next the PLC executes your program one instruction at a time. Maybe your program said that if the first input was on then it should turn on the first output. Since it already knows which inputs are on/off from the previous step it will be able to decide whether the first output should be turned on based on the state of the first input. It will store the execution results for use later during the next step.

Step 3-UPDATE OUTPUT STATUS-Finally the PLC updates the status ofthe outputs. It updates the outputs based on which inputs were on during the first step and the results of executing your program during the second step. Based on the example in step 2 it would now turn on the first output because the first input was on and your program said to turn on the first output when this condition is true.

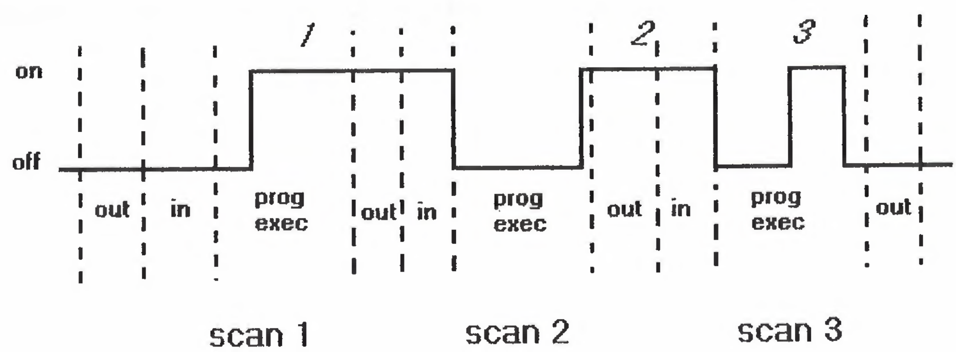
After the third step the PLC goes back to step one and repeats the steps continuously. One scan time is defined as the time it takes to execute the 3 steps listed above.

4.5 Response Time

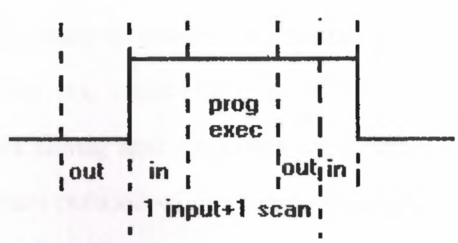


4.5.1 Response Time Concerns

Now that we know about response time, here's what it really means to the application. The PLC can only see an input turn on/off when it's looking. In other words, it only looks at its inputs during the check input status part of the scan.



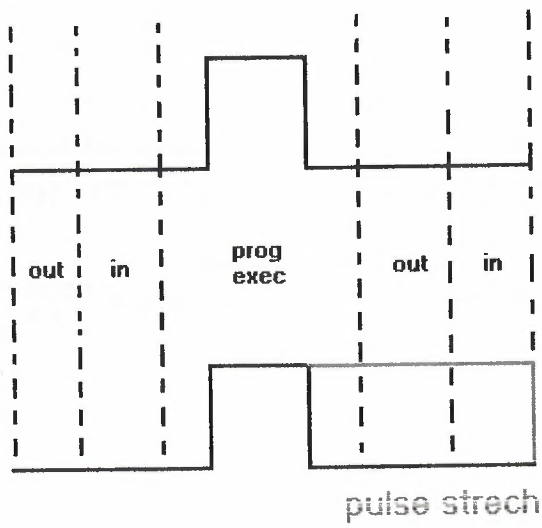
In the diagram, input 1 is not seen until scan 2. This is because when input 1 turned on, scan 1 had already finished looking at the inputs. Input 2 is not seen until scan 3. This is also because when the input turned on scan 2 had already finished looking at the inputs. Input 3 is never seen. This is because when scan 3 was looking at the inputs, signal 3 was not on yet. It turns off before scan 4 looks at the inputs. Therefore signal 3 is never seen by the PLC.



To avoid this we say that the input should be on for at least 1 input delay time + one scan time. But what if it was not possible for the input to be on this long? Then the PLC doesn't see the input turn on. Therefore it becomes a paper weight! Not true... of course there must be a way to get around this. Actually there are 2 ways.

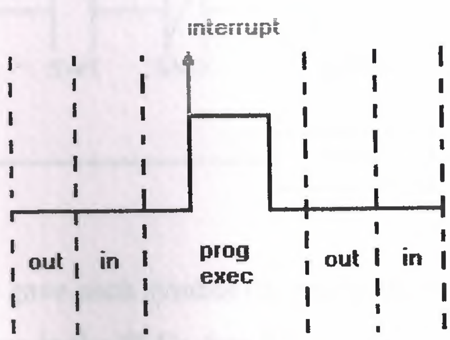
4.5.1 Response Time Concerns

This function extends the length of the input signal until the PLC looks at the inputs during the next scan. (i.e. it stretches the duration of the pulse.)



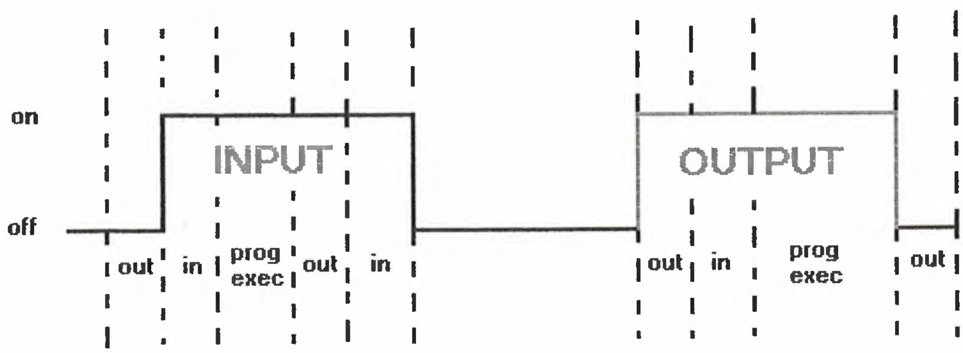
4.5.2 Interrupt function.

This function interrupts the scan to process a special routine that you have written. İ.e. As soon as the input turns on, regardless of where the scan currently is, the PLC immediately stops what its doing and executes an interrupt routine. (A routine can be thought of as a mini program outside of the main program.) After its done executing the interrupt routine, it goes back to the point it left off at and continues on with the normal scan process



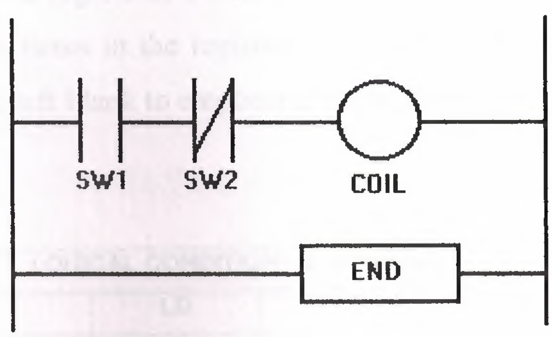
Now let's consider the longest time for an output to actually turn on. Let's assume that when a switch turns on we need to turn on a load connected to the PLC output. The diagram below shows the longest delay (worst case because the input is not seen until scan 2) for the output to turn on after the input has turned on.

The maximum delay is thus 2 scan cycles - 1 input delay time.



4.6 PLC Registers

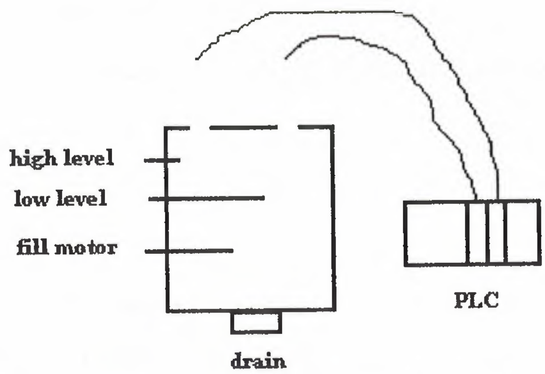
We'll now take the previous example and change switch 2 (SW2) to a normally closed symbol (load bar instruction). SW1 will be physically OFF and SW2 will be physically ON initially. The ladder diagram now looks like



Notice also that we now gave each symbol (or instruction) an address. This address sets aside a certain storage area in the PLCs data files so that the status of the instruction (i.e.

true/false) can be stored. Many PLCs use 16 slot or bit storage locations. In the example above we are using two different storage locations or registers.

REGISTER 00															



15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
														1	0

REGISTER 05															
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
															0

In the tables above we can see that in register 00, bit 00 (i.e. input 0000) was a logic 0 and bit 01 (i.e. input 0001) was a logic 1. Register 05 shows that bit 00 (i.e. output 0500) was a logic O. The logic 0 or 1 indicates whether an instruction is False or True. *Although most of the items in the register tables above are empty, they should each contain a O. They were left blank to emphasize the locations we were concerned with.

LOGICAL CONDITION OF SYMBOL			
LOGIC BITS	**LD**	LDB	OUT
Logic 0	False	True	False
Logic 1	True	False	True

The PLC will only energize an output when all conditions on the rung are TRUE. So, looking at the table above, we see that in the previous example SW1 has to be logic 1 and SW2 must be logic O. Then and ONLY then will the coil be true (i.e. energized). If any of the instructions on the rung before the output (coil) are false then the output (coil) will be false (not energized).

Let's now look at a truth table of our previous program to further illustrate this important point. Our truth table will show ALL possible combinations of the status of the two inputs.

Inputs		Outputs	Register Logic Bits		
SW1 (LD)	SW2(LDB)	COIL(OUT)	SW1 (LD)	SW2(LDB)	COIL(OUT)
False	True	False	0	0	0
False	False	False	0	1	0
True	True	True	1	0	1
True	False	False	1	1	0

Notice from the chart that as the inputs change their states over time, so will the output. The output is only true (energized) when all preceding instructions on the rung are true.

Level Application Example

Now that we've seen how registers work, let's process a program like PLCs do to enhance our understanding of how the program gets *scanned.*

Let's consider the following application:

We are controlling lubricating oil being dispensed from a tank. This is possible by using two sensors. We put one near the bottom and one near the top, as shown in the picture below

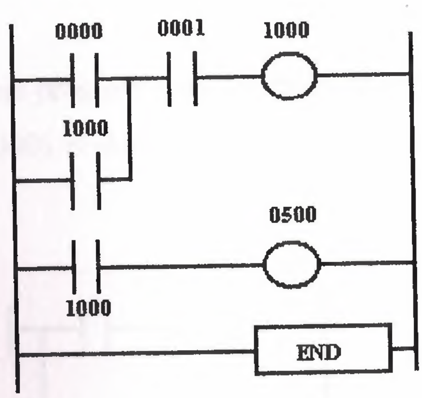
Here, we want the fill motor to pump lubricating oil into the tank until the high level sensor turns on. At that point we want to turn off the motor until the level falls below the low level sensor. Then we should turn on the fill motor and repeat the process.

Here we have a need for 3 I/0 (i.e. Inputs/Outputs). 2 are inputs (the sensors) and 1 is an output (the fill motor). Both of our inputs will be NC (normally closed) fiber-optic level sensors. When they are NOT immersed in liquid they will be ON. When they are immersed in liquid they will be OFF.

We will give each input and output device an address. This lets the PLC know where they are physically connected. The addresses are shown in the following tables:

Inputs	Address	Output	Address	Internal Utility Relay
Low	0000	Motor	0500	1000
High	0001			

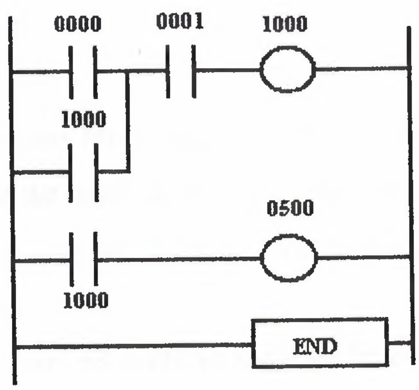
Below is what the ladder diagram will actually look like. Notice that we are using an internal utility relay in this example. You can use the contacts of these relays as many times as required. Here they are used twice to simulate a relay with 2 sets of contacts. Remember, these relays DO NOT physically exist in the PLC but rather they are bits in a register that you can use to SIMULATE a relay.



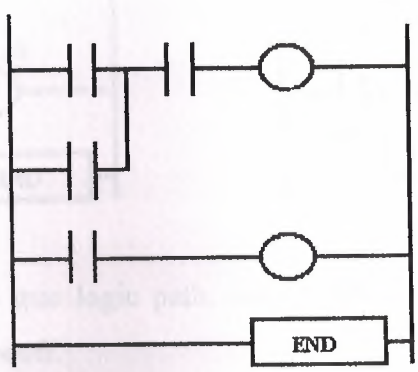
We should always remember that the most common reason for using PLCs in our applications is for replacing real-world relays. The internal utility relays make this action possible. It's impossible to indicate how many internal relays are included with each brand of PLC. Some include 100's while other include 1000's while still others include 10's of 1000's! Typically, PLC size (not physical size but rather I/O size) is the deciding factor. If we are using a micro-PLC with a few I/O we don't need many internal relays. If however, we are using a large PLC with 100's or 1000's of 110 we'll certainly need many more internal relays.

If ever there is a question as to whether or not the manufacturer supplies enough internal relays, consult their specification sheets. In all but the largest of large applications, the supplied amount should be MORE than enough.

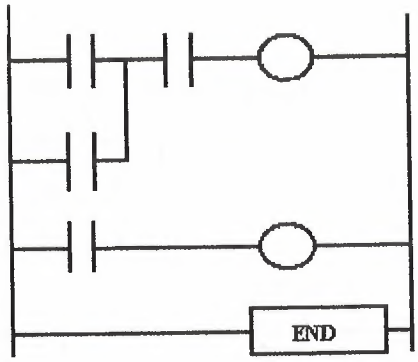
4.6.1 The program scan



Let's watch what happens in this program scan Initially the tank is empty. Therefore, input 0000 is TRUE and input 0001 is also TRUE.



Initially the tank is empty. Therefore, input 0000 is TRUE and input 0001 is also TRUE.

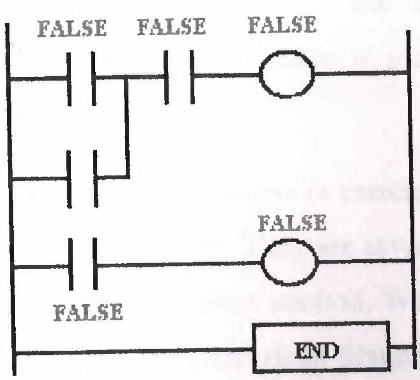


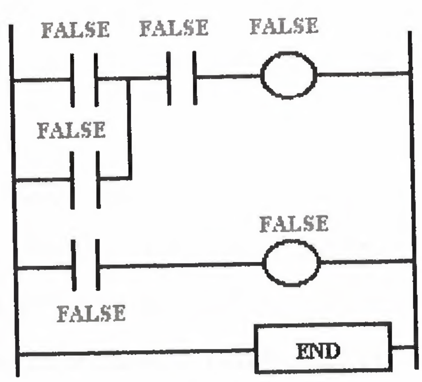
Gradually the tank fills because 500(fill motor) is on.

After 100 scans the oil level rises above the low level sensor and it becomes open. (i.e. FALSE)

Notice that even when the low level sensor is false there is still a path of true logic from left to right. This is why we used an internal relay. Relay 1000 is latching the output (500) on. It will stay this way until there is no true logic path from left to right.(i.e. when 0001 becomes false)

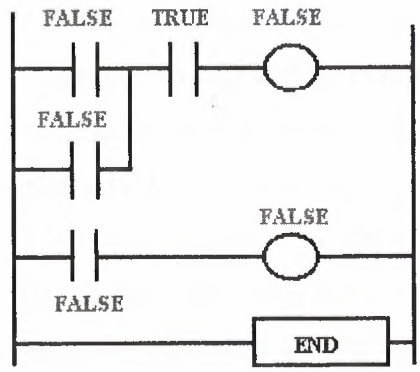
After 1000 scans the oil level rises above the high level sensor at it also becomes open (i.e. false)





Since there is no more true logic path, output 500 is no longer energized (true) and therefore the motor turns off.

After 1050 scans the oil level falls below the high level sensor and it will become true .



Notice that even though the high level sensor became true there still is NO continuous true logic path and therefore coil 1000 remains false!

After 2000 scans the oil level falls below the low level sensor and it will also become true again. At this point the logic will appear the same as SCAN 1 above and the logic will repeat as illustrated above.

CHAPTER 5

5 COMMUNICATION

5.1 RS-232 Communications (hardware)

RS-232 communications is the most popular method of PLC to external device communications. Let's tackle it piece by piece to see how simple it can be when we understand it.

RS-232 is an asynchronous (a marching band must be "in sync" with each other so that when one steps they all step. They are asynchronous in that they follow the band leader to keep their timing) communications method. We use a binary system (1's and 0's) to transmit our data in the ASCII format. (American Standard Code for Information Interchange- pronounced ASS-KEY) This code translates human readable code (letters/numbers) into "computer readable" code (1's and 0's). Our pies serial port is used for transmission/reception of the data. It works by sending/receiving a voltage. A positive voltage is called a MARK and a negative voltage is called a SPACE. Typically, the PLC works with +/-15volts. The voltage between +/- 3 volts is generally not used and is considered noise.

and is considered noise.

There are 2 types of RS-232 devices. The first is called a DTE device. This means Data Terminal Equipment and a common example is a computer. The other type iDCE device. DCE means Data Communications Equipment and a common example is a modem. Your PLC may be either a DTE or DCE device. Check your documentations The PLC serial port works by turning some pins on while turning other off. These pins each are dedicated to a specific purpose. The serial port comes in 2 flavors a 25-pin type and a 9-pin type. The pins and their purposes are shown below. (This chart assumes your PLC is DTE device)

9-PIN	25-PIN	PURPOSE
1	1	frame ground
2	3	receive data (RD)
3	2	transmit data (TO)
4	20	data terminal ready (DTR)
5	7	signal ground
6	6	data set ready (DSR)
7	4	request to send (RTS)
8	5	clear to send (CTS)
9	22	ring indicator (RI) *only for modems*

Each pins purpose in detail:

· frame ground- This pin should be internally connected to the chassis of the device.

· receive data- This pin is where the data from the external device enters the PLC serial port.

· transmit data- This pin is where the data from the PLC serial port leaves the PLC enroute to the external device.

· data terminal ready- This pin is a master control for the external device. When this pin is 1 the external device will not transmit or receive data.

· signal ground- Since data is sent as + or - voltage, this pin is the ground that is referenced.

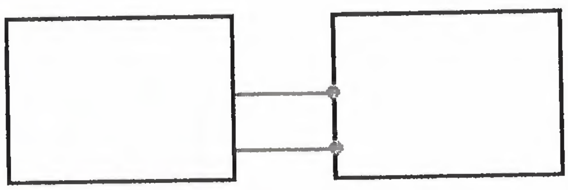
· data set ready- Usually external devices have this pin as a permanent 0 and the PLC basically uses it to determine that the external device is powered up and ready.

· request to send- This is part of hardware handshaking. When the PLC wants to send data to the external device it sets this pin to a 0. In other words, it sets the pin to a 0 and basically says "I want to send you data. Is it ok?" The external device says it's OK to send data by setting its clear to send pin to 0. The PLC then sends the data.

. clear to send- This is the other half of hardware handshaking. As noted above, the external device sets this pin to 0 when it is ready to receive data from the PLC.

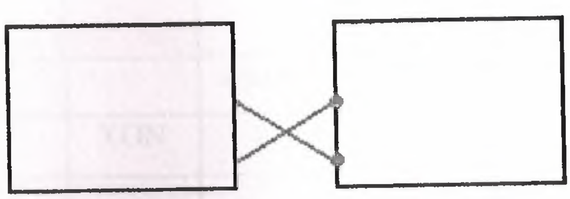
. ring indicator- only used when the PLC is connected to a modem.

What happens when your PLC and external device are both DTE (or both DCE) devices? They can't talk to each other, that's what happens. The picture below shows why 2 same type devices can't communicate with each other.



Notice that in the] ... the first device is connected to the rece ... smit data line (pin3) of the first device is ... vice. Ifs like talking through a phone with the wires reversed. (i.e. your mouth piece is connected directly to the other parties mouthpiece and your ear piece is connected directly to the other parties earpiece.) Obviously, this won't work well!

The solution is to use a null-modem connection as shown below. This is typically done by using a reverse (null-modem) cable to connect the devices.



To summarize everyı ... Both devices are powered up. the PLC ... The external device ı ... vered up and "there". The PLC turns on ... e "are you ready to receive some data?" The external device responds by turning on it's CTS which says it's ok to for the PLC to send data. The PLC sends the data on its TD terminal and the external device receives it on its RD terminal. Some data is sent and received. After a while, the external device can't process the data quick enough. So, it turns off its CTS terminal and the PLC pauses sending data. The external device catches up and then turns its CTS terminal back on. The PLC again starts sending data on its TD terminal and the external device receives it on its RD terminal. The PLC runs out of data to send and turns off its RTS terminal. The external device sits and waits for more data.

5.1.1 RS-232 Communications (software)

Now that we understand the hardware part of the picture, let's dive right into the software part. We'll define a few of the common terms.

ASCII is a human-readable to computer-readable translation code. (i.e. each letter/number is translated to 1's and 0's) It's a 7-bit (a bit is a 1 or a 0) code, so we can translate 128 characters. (2^7 is 128) Character sets that use the 8th bit do exist but they are not true ASCII. Below is an ASCII chart showing its "human-readable" representation. We typically refer to the characters by using hexadecimal terminology. "0" is 30h, "5" is 35h, "E" is 45h, etc. ("h" means hexadecimal)

	Most significant bits								
Least Sig.bits		0	1	2	3	4	5	6	7
	0			space	0	@	P	`	p
	1		XON	!	1	A	Q	a	q
	2	STX		"	2	B	R	b	r
	3	ETX	XOFF	#	3	C	S	c	s
	4			$	4	D	T	d	t
	5		NAK	%	5	E	U	e	u

	6	ACK		&	6	F	V	f	v
	7			'	7	G	W	g	w
	8			(	8	H	X	h	x
	9			)	9	I	Y	i	y
	A	LF		*	:	J	Z	j	z
	B			+	;	K	[	k	{
	C			,	<	L	\	l	|
	D	CR		-	=	M	]	m	}
	E			.	>	N	^	n	~
	F			/	?	O	_	o	

Start bit-In RS-232 the first thing we send is called a start bit.This start bit(invented during WW1 by Kleinschmidt) is a synchronizing bit added just before each character we are sending.This is considered a SPACE or negative voltage or a 0.

Stop bit- The last thing we send is called a stop bit.This stop bit tells us that the last character was just sent.Thing of it as an end-of-character bit.This is considered a MARK or positive voltage or a 1.The start and stop bits are commonly called framing bits because they surround the character we are sending.

Parity bit-Since most PLCs/external equipment are byte-oriented(8 bits=1 byte)it seems natural to handle data as a byte.Althought ASCII is a 7-bit code it is rarely transmitted that way.Typically,the 8^{th} bit is used as a parity bit for error checking.This method of error checking gets its name from the math idea of parity.In simple terms,prity means that all characters will either have an odd number of 1's or an even number of 1's.

Common forms of parity are None,Even, and Odd.(Mark and Space aren't very common so I won't discuss them).Consider these examples;

Send "E" (45h or 1000101(binary)).

In parity of None,the parity bit is always 0 so we send 10001010.

In parity of Even we must have an Even number of 1's in our total character so the original character currentky has 3 1's(1000101)therefore our parity bit we will add must be a 1.(10001011)now we have an evev number of 1's.

In odd paity we need an odd number of 1's.Since our original character already has an odd number of 1's (3 is an odd number) our parity bit will be a 0.(10001010)

During transmission, the sender calculates the parity bit and sends it. The receiver calculates parity for the 7 -bit character and compares the result to the parity bit received. If the calculated and real parity bits don't match, an error occurred an we act appropriately

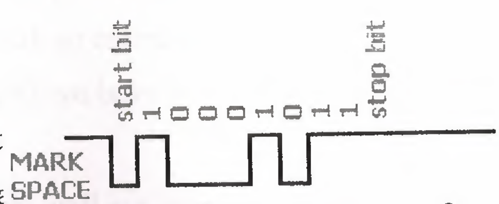
It's strange that this parity method is so popular. The reason is because it's only effective half the time. That is, parity checking can only find errors that effect an odd number of bits. If the error affected 2 or 4 or 6 bits the method is useless. Typically, errors are caused by noise which comes in bursts and rarely effects 1 bit. Block redundancy checks are used in other communication methods to prevent this.

Baud rate- I'll perpetuate the incorrect meaning since it's most commonly used incorrectly. Think of baud rate as referring to the number of bits per second that are being transmitted. So 1200 means 1200 bits per second are being sent and 9600 means 9600 bits are being transmitted every second. Common values (speeds) are 1200, 2400, 4800, 9600, 19200, and 38400.

RS232 data format- (baud rate-data bits-parity-stop bits) This is the way the data format is typically specified. For example, 9600-8-N-1 means a baud rate of 9600,

8 data bits, parity of None, and 1 stop bit.

The picture below shows how data leaves the serial port for the character "E" (45h 100 0101b) and Even parity .



Another important thing t ıre handshaking (flow control). Like the hardwa ous chapter, software handshaking is used to make sure both devices are ready to send/receive data. The most popular "character flow control" is called XON/XOFF. It's very simple to understand. Simply put, the receiver sends the XOFF character when it wants the transmitter to pause sending data. When it's ready to receive data again, it sends the transmitter the XON character. XOFF is sometimes referred to as the hold off character and XON as the release character. The last thing we should know about is delimiters. A delimiter is simply added to the end of a message to tell the receiver to process the data it has received. The most common is the CR or the CR and LF pair. The PLC/external device receives this and knows to take the data from its buffer (where the data is stored temporarily before being processed). An LF (line feed) is also sometimes sent with the CR character. If viewed on a computer screen this would look like what happens on the typewriter when the carriage is returned and the page moves down a line so you don't type over what you just typed.

Sometimes an STX and ETX pair is used for transmission/reception as well. STX is "start of text" and ETX is "end of text". The STX is sent before the data and tells the external device that data is coming. After all the data has been sent, an ETX character is sent.

Finally, we might also come across an ACK/NAK pair. This is rarely used but it should be noted as well. Essentially, the transmitter sends its data. If the receiver gets it without error, it sends back an ACK character. If there was an error, the receiver sends back a NAK character and the transmitter resends the data.

5.1.2 Using RS-232 with Ladder Logic

Now that we understand what RS-232 is/means let's see how to use it with our PLC.We should start out as always, remembering that a PLC is a PLC is a PLC... In other words, understand the theory first and then figure out how our manufacturer of choice "makes it work". Some manufacturers include RS-232 communication capability in the main processor. Some use the "programming port" for this. Others require you to purchase a module to "talk RS-232" with an external device.

To communicate via RS-232 we have to setup a few things. Ask yourself the following questions:

- Where, in data memory, will we store the data to be sent? Essentially we have to store the data we will send. Where, in data memory, will we put the data we receive from the external device?

- How will we tell the PLC when ifs time to send our data (the data we stored in data memory) out the serial port?

- How will we know when we have received data from our external device?

If you know the above, then the rest is easy. If you don't know the above, then make something up and now the rest is easy. Huh??? Simple, pick a memory area to work with and figure out if we can choose the internal relays to use to send and receive data or if the PLC has ones that are dedicated to this

purpose.

1. We assign memory locations DM100 through DM102 to be where we'll put our data before we send it out the serial port. Many PLCs have dedicated areas of memory for this and only this purpose.

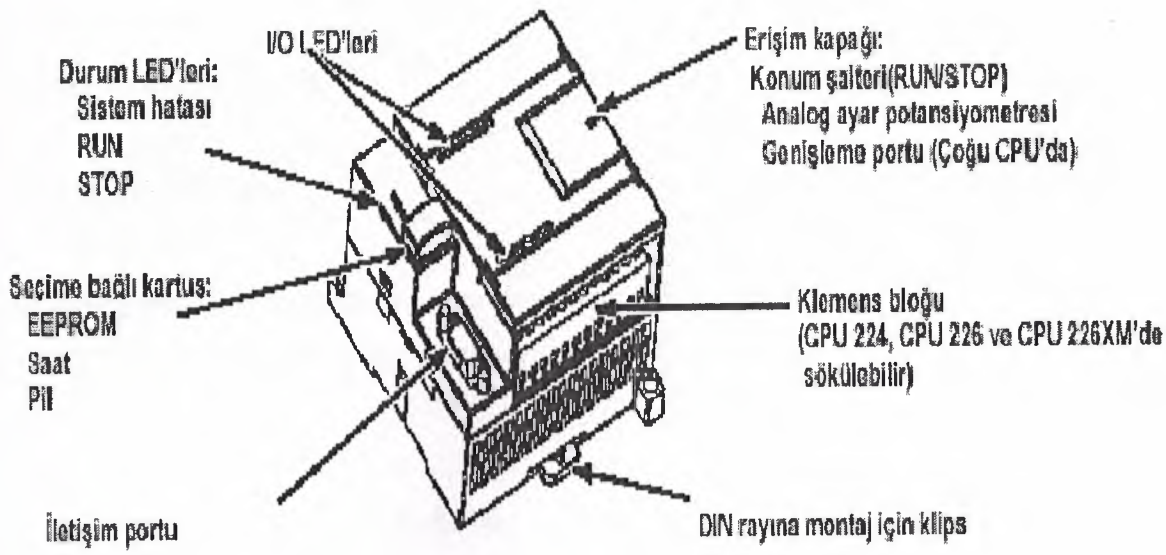
2. We'll assign internal relay 1000 to be our send relay. In other words, when we turn on 1000 the PLC will send the data in DM100-DM102 out the serial port to our external device. Note againMany PLCs have dedicated relays (special utility relays) for this and only this purpose. Ifs great when the manufacturer makes our life easy! We'll send the string "air" out the PLC serial port to an operator interface when our temp sensor input turns on. This means our oven has become too hot. When the operator interface receives this string it will displayed an alarm message for the operator to see. Look back on the ASCII chart and you'll see that "air" is hexadecimal 61, 6C, 72. (a=61, I=6C, r=72) We'll write these ASCII characters (in hexadecimal form) into the individual data memory locations. We'll use DM100-102. Remember the LDA or MOV instruction? We'll turn on our send relay (1000) when our temperature sensor (0000) turns on.

Some PLCs may not have dedicated internal relays that send out our data through the RS-232 port. We may have to assign them manually. Further, some pies will have a special instruction to tell us where the data is stored and when to send the data. This instruction is commonly called AWT (ASCII Write) or RS.Put the data in a memory location and then turn on a relay to send the data.

CHAPTER 6

6 TYPES OF PLCs

S7-200 has a target to be a strong PLC.Because of this it produced like a compact structure microprocessor.It includes power supply,input and output circuits.



CHAPTER 7

7 NUMBER SYSTEMS

Since a PLC is a computer, it stores information in the form of On or Off conditions (1 or 0), referred to as binary digits (bits). Sometimes binary digits are used individually and sometimes they are used to represent numerical values.

7.1 Decimal system

Various number systems are used by PLCs. All number systems
have the same three characteristics: digits, base, weight. The decimal system, which is commonly used in everyday life, has the following characteristics:

Ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base 10

Weights 1, 10, 100, 1000, ...

7.2 Binary system

The binary system is used by programmable controllers. The
binary system has the following characteristics:

Two digits 0, 1

Base 2

Weights Pers of base 2 (1, 2, 4, 8, 16, ...) In the binary system 1s and 0s are arranged into columns. Each column is weighted. The first column has a binary weight of 2_0. This is equivalent to a decimal 1. This is referred to as the least significant bit. The binary weight is doubled with each succeeding column. The next column, for example, has a weight of 2_1, which is equivalent to a decimal 2. The decimal value is doubled in each successive column. The number in the far left
hand column is referred to as the most significant bit. In this example, the most significant bit has a binary weight of 2_7. This is equivalent to a decimal 128

B. SIEMENS S7 PLC'LERİN KARŞILAŞTIRILMALARI

SIEMENS - S7	S7-200 (216)	S7-300 (315)	S7-400(416)	S7-314 IFM
Dijital giriş çıkış sayısı	128 bit	1024 bit	128 K bit	1024 bit (I0.0→I127.7)
Analog giriş, çıkış sayısı	20 bayt	128 bayt	8192 bayt	64 (PIW 128 → 256)
Sayıcı sayısı	256	64	512	64 (Z 0 → Z 63)
Zaman eleman sayısı	256	128	512	128 (T 0 → T 127)
Durum tespit işareti sayısı (merker, yardımcı röle)	256 bit	2048 bit	16384 bit	2048 (M 0.0 → M 255.7)
Bellek durumlar	8 Kbayt 4 1234568ccwr K emirler için 4 K veriler için	48 Kbayt 16 K emirler için 32 K veriler için	512 Kbayt	24 Kbayt 8 K emirler için
Çalışma hızı (1 K bineer emri işleme için)	0.8 ms	0.3 ms	0.08 ms	0.3 ms
Program işleme şekli	Yapısal	Yapısal	Yapısal	Yapısal
Adresleme imkanları	Direkt,indirekt, sembolik	Direkt,indirekt, sembolik	Direkt,indirekt, sembolik	Direkt,indirekt,sembolik
Register durumu	16 Bit - 32 Bit	16 Bit - 32 Bit	16 Bit - 32 Bit	16 Bit - 32 Bit
Haberleşme arabirimi	2xPPI (Point to point)	MPI (Multi Point Int)	MPI PROFIBUS-DP	MPI
Ağ	AS-Interface	AS-Interface PROFIBUS Ind. Ethernet	PROFIBUS Industriel Ethernet	AS-Interface PROFIBUS Ind. Ethernet

Malzeme Cinsi	Sipariş No	Birim Fiyatı EURO/Adet
MODÜLER YENİ NESİL LOGO!BASIC/PURE LOJİK MODÜLLERİ(0BA4)		
LOGO!24 :24/24/24 2AI 8I/4O	6ED1052-1CC00-0BA4	99
LOGO!12/24 RC:12-24/12-24V/RÖLE+SAAT 2AI 8I/4O	6ED1052-1MD00-0BA4	112
LOGO!24 RC(AC/DC) :24V/24V/RÖLE+SAAT 8I/4O	6ED1052-1HB00-0BA4	112
LOGO!230 RC: 230V/230V/RÖLE+SAAT 8I/4O	6ED1052-1FB00-0BA4	115
LOGO!24o :24V/24V/24V 2AI 8I/4O	6ED1052-2CC00-0BA4	85
LOGO!12/24RCo:12-24V/12-24V/RÖLE+SAAT 2AI 8I/4O	6ED1052-2MD00-0BA4	89
LOGO!24RCo(AC): 24V/24V/RÖLE+SAAT 8I/4O	6ED1052-2HB00-0BA4	89
LOGO!230RCo 230V/230V/RÖLE+SAAT 8I/4O	6ED1052-2FB00-0BA4	91

FATEK

. 12 Giriş (24 VDC) ~ 8 Çıkış (Röle)
. 2 Hızlı sayıcı girişi ~ 20 Khz

. 12 Giriş (24 VDC) ~ 8 Çıkış (Transistör)
. 2 Hızlı sayıcı girişi ~ 2 Hızlı Darbe Çıkışı (20 Khz)

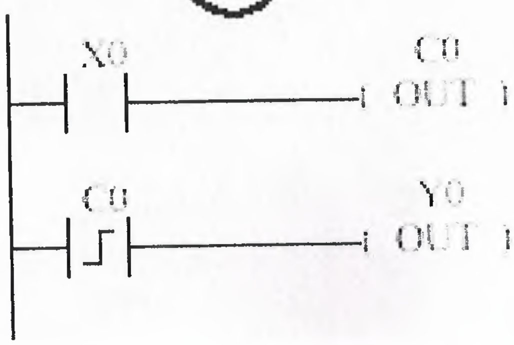
. 16 Giriş (24 VDC) ~ 12 Çıkış (Röle)
. 3 Hızlı sayıcı girişi ~ 20 Khz

. 16 Giriş (24 VDC) ~ 12 Çıkış (Transistör)
. 3 Hızlı sayıcı girişi ~ 3 Hızlı Darbe Çıkışı (20 Khz)

. 24 Giriş (24 VDC) ~ 16 Çıkış (Röle)
. 4 Hızlı sayıcı girişi ~ 20 Khz

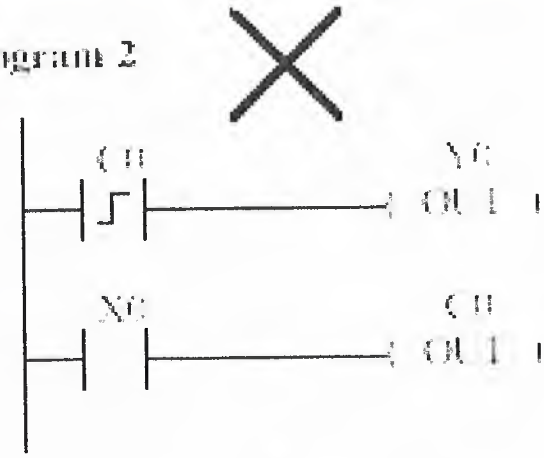
. 24 Giriş (24 VDC) ~ 16 Çıkış (Transistör)
. 4 Hızlı sayıcı girişi ~ 4 Hızlı Darbe Çıkışı (20 Khz)

Program 1

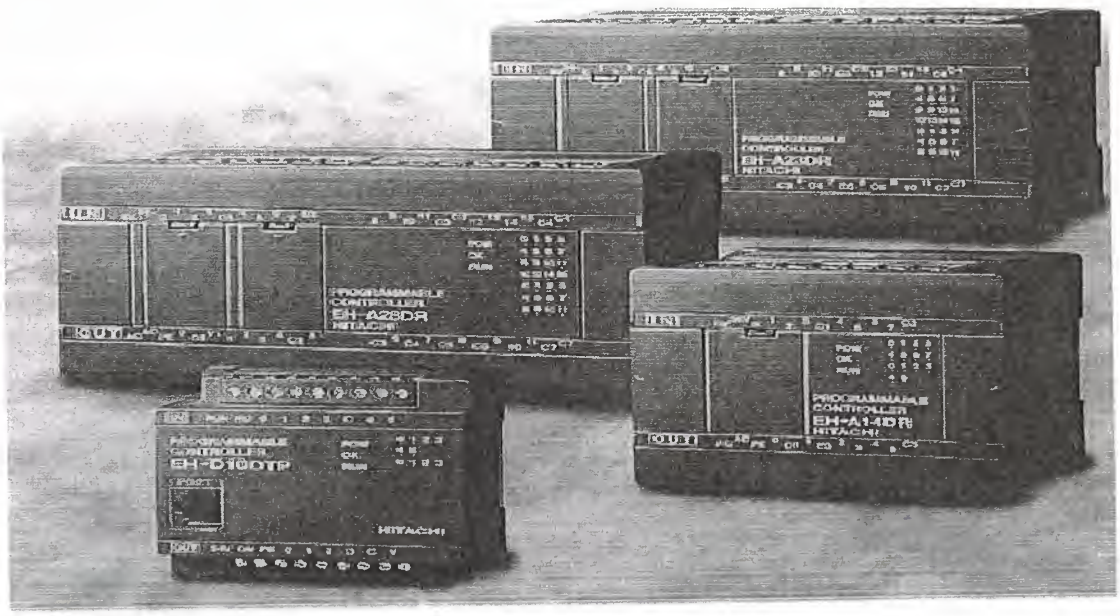


When X0 turns on, Y0 also turns on

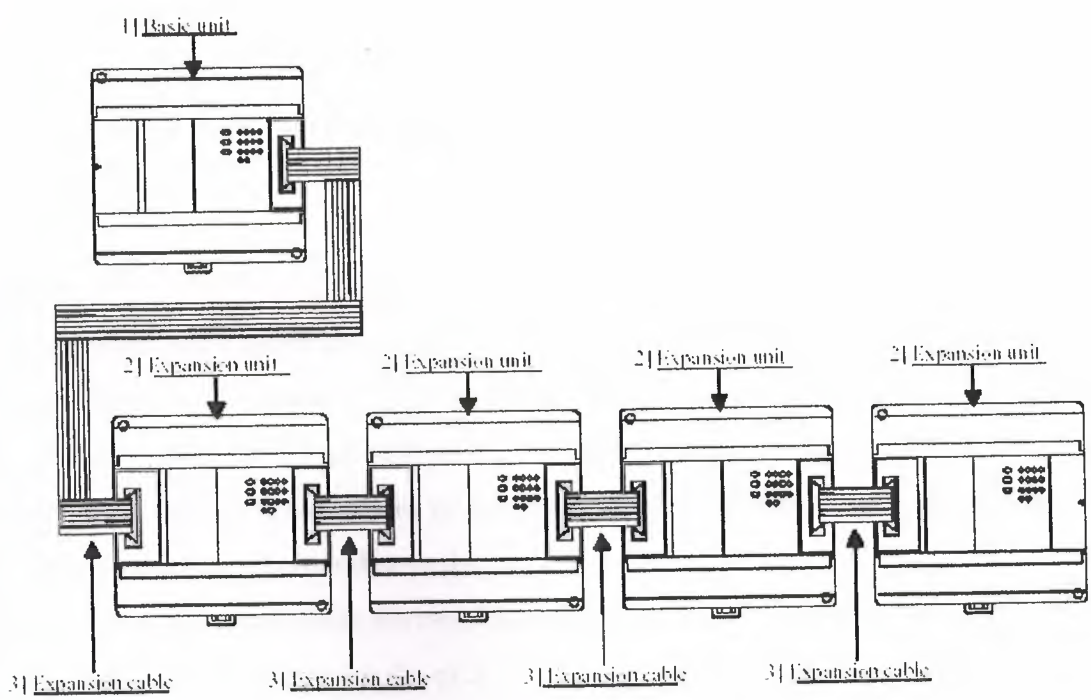
Program 2



Y0 never turns on.



		Compact PLC (Central Units)				Expansion units	
Type		MICRO-EH 10	MICRO-EH 14	MICRO-EH 23	MICRO-EH 28	MICRO-EH 14E	MICRO-EH 28E
Power supply	24V DC	•	•	•	•	•	•
	110 - 240 VAC	–	•	•	•	•	•
Speed	Cycle time for 1 kByte	0.9µs	0.9µs	0.9µs	0.9µs		
Memory	Program (1 step = 4 Byte)	3 k steps max. (12 kByte)	3 k steps max. (12 kByte)	3 k steps max. (12 kByte)	3 k steps max. (12 kByte)		
	Type	FLASH and RAM	FLASH and RAM	FLASH and RAM	FLASH and RAM		
	Pluggable	coming soon	coming soon	coming soon	coming soon		
Functions	Real time clock	–	–	•	•		
	Potentiometer, 10 bit resolution	–	2	2	2		
Modules	Digital inputs	6	8	13	16	8	16
	Type of inputs	DC	DC / AC	DC	DC / AC	DC / AC	DC
	Digital outputs	4	6	10	12	6	12
	Type of outputs	DC	Relay/Trans/Triac	Relay	Relay/Trans/Triac	Relay/Trans/Triac	Relay/Trans
	Analogue inputs			2			
	Type of analogue inputs			mA / Vdc, 12 bit			
	Analogue outputs			1			
	Type of analogue outputs			mA / Vdc, 12 bit			
	Max number of expansion units MICRO-EH 14E	–	4	4	4		
	High-speed counter inputs	3 channels 10 kHz max.	4 channels 10 kHz max.	4 channels 10 kHz max.	4 channels 10 kHz max.		
	Interrupt inputs	3	4	4	4		
	PWM outputs	4 channels 2 kHz	4 channels 2 kHz	4 channels 2 kHz	4 channels 2 kHz		
	Pulse output	4 channels 5 kHz	4 channels 5 kHz	4 channels 5 kHz	4 channels 5 kHz		
Base unit	32 bit RISC processor	•	•	•	•		
Communication	Serial	RS-232 (1 x)	RS-232 (1 x)	RS-232 (1 x) RS-485 (1 x)	RS-232 (1 x) RS-485 (1 x)		
Connections	Removable terminal block	–	•	•	•	•	•
Mounting		DIN rail or wall mounting					



7 NUMBER SYSTEMS

Since a PLC is a computer, it stores information in the form of On or Off conditions (1 or 0), referred to as binary digits (bits). Sometimes binary digits are used individually and sometimes they are used to represent numerical values.

7.1 Decimal system

Various number systems are used by PLCs. All number systems have the same three characteristics: digits, base, weight. The decimal system, which is commonly used in everyday life, has the following characteristics:

Ten digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Base 10

Weights 1, 10, 100, 1000, ...

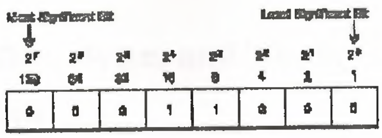
7.2 Binary system

The binary system is used by programmable controllers. The binary system has the following characteristics:

Two digits 0, 1

Base 2

Weights Pers of base 2 (1, 2, 4, 8, 16, ...) In the binary system 1s and 0s are arranged into columns. Each column is weighted. The first column has a binary weight of 2_0. This is equivalent to a decimal 1. This is referred to as the least significant bit. The binary weight is doubled with each succeeding column. The next column, for example, has a weight of 2_1, which is equivalent to a decimal 2. The decimal value is doubled in each successive column. The number in the far left hand column is referred to as the most significant bit. In this example, the most significant bit has a binary weight of 2_7. This is equivalent to a decimal 128

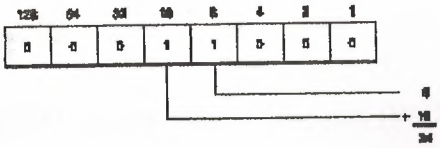


7.2.1 Converting binary

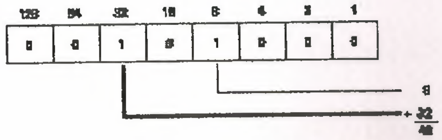
The following steps can be used to interpret a decimal
to Decimal number from a binary value.

1. Search from least to most significant bit for 1s.
2. Write down the decimal representation of each column containing a 1.
3. Add the column values.

In the following example, the fourth and fifth columns from the right contain a 1. The decimal value of the fourth column from the right is 8, and the decimal value of the fifth column from the right is 16. The decimal equivalent of this binary number is 24. The sum of all the weighted columns that contain a 1 is the decimal number that the PLC has stored.

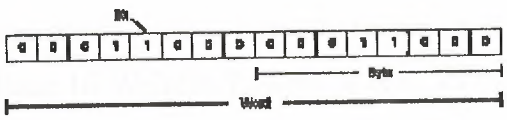


In the following example the fourth and sixth columns from the right contain a 1. The decimal value of the fourth column from the right is 8, and the decimal value of the sixth column from the right is 32. The decimal equivalent of this binary number is 40.

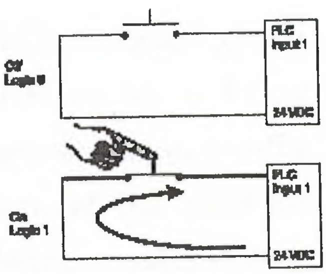


Bits, Bytes, and Words Each binary piece of data is a bit. Eight bits make up one byte.

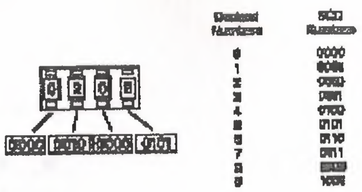
Two bytes, or 16 bits, make up one word.



Logic 0, Logic 1 Programmable controllers can only understand a signal that is On or Off (present or not present). The binary system is a system in which there are only two numbers, 1 and 0. Binary 1 indicates that a signal is present, or the switch is On. Binary 0 indicates that the signal is not present, or the switch is Off.



BCD Binary-Coded Decimal (BCD) are decimal numbers where each digit is represented by a four-bit binary number. BCD is commonly used with input and output devices. A thumbwheel switch is one example of an input device that uses BCD. The binary numbers are broken into groups of four bits, each group representing a decimal equivalent. A four-digit thumbwheel switch, like the one shown here, would control 16 (4 x 4) PLC



Hexadecimal Hexadecimal is another system used in PLCs. The hexadecimal system has the following characteristics: 16 digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F Base 16 Weights Powers of base 16 (1, 16, 256, 4096 ...) The ten digits of the decimal system are used for the first ten

digits of the hexadecimal system. The first six letters of the alphabet are used for the remsining six digits.

A = 10 D = 13

B = 11 E = 14

C = 12 F = 15

The hexadecimal system is used in PLCs because it allows the status of a large number of binary bits to be represented in a small space such as on a computer screen or programming device display. Each hexadecimal digit represents the exact status of four binary bits. To convert a decimal number to a hexadecimal number the decimal number is divided by the base of 16. To convert decimal 28, for example, to hexadecimal:

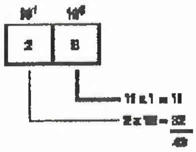
$$\begin{array}{r} 1 \; r\,12 \\ 16\overline{)28} \end{array}$$

Decimal 28 divided by 16 is 1 with a remainder of 12. Twelve is equivalent to C in hexadecimal. The hexadecimal equivalent of decimal 28 is 1C. The decimal value of a hexadecimal number is obtained by multiplying the individual hexadecimal digits by the base 16 weight and then adding the results. In the following example the hexadecimal number 2B is converted to its decimal equivalent of 43.

$16^0 = 1$

$16^1 = 16$

B = 11



7.2.3 Conversion of numbers

The following chart shows a few numeric values in decimal, binary, BCD, and hexadecimal representation.

Decimal	Binary	BCD	Hexadecimal
0	0	0000	0
1	1	0001	1
2	1	0010	2
3	11	0011	3
4	100	0100	4
5	101	0101	5
6	110	0110	6
7	111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	0001 0000	A
11	1011	0001 0001	B
12	1100	0001 0010	C
13	1101	0001 0011	D
14	1110	0001 0100	E
15	1111	0001 0101	F
16	1 0000	0001 0110	10
17	1 0001	0001 0111	11
18	1 0010	0001 1000	12
19	1 0011	0001 1001	13
20	1 0100	0010 0000	14
.	.	.	.
.	.	.	.
126	111 1110	0001 0010 0110	7E
127	111 1111	0001 0010 0111	7F
128	1000 0000	0001 0010 1000	80
.	.	.	.
.	.	.	.
510	1 1111 1110	0101 0001 0000	1FE
511	1 1111 1111	0101 0001 0001	1FF
512	10 0000 0000	0101 0001 0010	200

7.2.3 Conversion of numbers

The following chart shows a few numeric values in decimal, binary, BCD, and hexadecimal representation.

Decimal	Binary	BCD	Hexadecimal
0	0	0000	0
1	1	0001	1
2	1	0010	2
3	11	0011	3
4	100	0100	4
5	101	0101	5
6	110	0110	6
7	111	0111	7
8	1000	1000	8
9	1001	1001	9
10	1010	0001 0000	A
11	1011	0001 0001	B
12	1100	0001 0010	C
13	1101	0001 0011	D
14	1110	0001 0100	E
15	1111	0001 0101	F
16	1 0000	0001 0110	10
17	1 0001	0001 0111	11
18	1 0010	0001 1000	12
19	1 0011	0001 1001	13
20	1 0100	0010 0000	14
.	.	.	.
.	.	.	.
126	111 1110	0001 0010 0110	7E
127	111 1111	0001 0010 0111	7F
128	1000 0000	0001 0010 1000	80
.	.	.	.
.	.	.	.
510	1 1111 1110	0101 0001 0000	1FE
511	1 1111 1111	0101 0001 0001	1FF
512	10 0000 0000	0101 0001 0010	200

CHAPTER 8

8 ANALOG INPUTS AND OUTPUTS

PLCs must also work with continuous or analog signals. Typical analog signals are 0 - 10 VDC or 4 - 20 mA. Analog signals are used to represent changing values such as speed, temperature, weight, and level. A PLC cannot process these signals in an analog form. The PLC must convert the analog signal into a digital representation. An expansion module, capable of converting the analog signal, must be used. The S7-200 analog modules convert standard voltage and current analog values into a 12-bit digital representation. The digital values are transferred to the PLC for use in register or word

Analog outputs are used in applications requiring control capability of field devices which respond to continuous voltage or current levels. Analog outputs may be used as a variable reference for control valves, chart recorders, electric motor drives, analog meters, and pressure transducers. Like analog inputs, analog outputs are generally connected to a controlling device through a transducer. The transducer takes the voltage signal and, depending on the requirement, amplifies, reduces, or changes it into another signal which controls the device. In the following example a 0 - 10 VDC signal controls a 0 - 500 Lbs. scale analog meter.

8.1.1 Timers

Timers are devices that count increments of time. Traffic lights are one example where timers are used. In this example timers are used to control the length of time between signal changes. Timers are represented by boxes in ladder logic. When a timer receives an enable, the timer starts to time. The timer compares its current time with the preset time. The output of the timer is a logic 0 as long as the current time is less than the preset time. When the current time is greater than the preset time the timer output is a logic 1. S7-200 uses three types of timers: On-Delay (TON), Retentive On-Delay (TONR), and Off-Delay (TOF).

S7-200 Timers

S7-200 timers are provided with resolutions of 1 millisecond, 10 milliseconds, and 100 milliseconds. The maximum value of these timers is 32.767 seconds, 327.67 seconds, and 3276.7 seconds, respectively. By adding program elements, logic can be programmed for much greater time intervals.

Hard-Wired Timing Circuit

Timers used with PLCs can be compared to timing circuits used in hard-wired control line diagrams. In the following example, a normally open (NO) switch (S1) is used with a timer (TR1). For this example the timer has been set for 5 seconds. When S1 is closed,

TR1 begins timing. When 5 seconds have elapsed, TR1 will close its associated normally open TR1 contacts,
illuminating pilot light PL1. When S1 is open, deenergizing TR1, the TR1 contacts open, immediately extinguishing PL1. This type of timer is referred to as ON delay. ON delay indicates that once a timer receives an enable signal, a predetermined amount of time (set by the timer) must pass before the timer's contacts
change state.

On-Delay (TON)

When the On-Delay timer (TON) receives an enable (logic 1) at its input (IN), a predetermined amount of time (preset time - PT) passes before the timer bit (T-bit) turns on. The T-bit is a logic function internal to the timer and is not shown on the symbol. The timer resets to the starting time when the enabling input goes to a logic 0.

In the following simple timer example, a switch is connected to input I0.3, and a light is connected to output Q0.1. When the switch is closed input 4 becomes a logic 1, which is loaded into timer T37. T37 has a time base of 100 ms (.100 seconds). The preset time (PT) value has been set to 150. This is equivalent to 15 seconds (.100 x 150). The light will turn on 15 seconds after the input switch is closed. If the switch were opened before 15 seconds had passed, then reclosed, the timer
would again begin timing at 0.

A small sample of the flexibility of PLCs is shown in thefollowing program logic. By reprogramming the T37 contact asa normally closed contact, the function of the circuit is changed to cause the indicator light to turn off only when the timer times
out. This function change was accomplished without changing
or rewiring I/O devices.

Retentive On-Delay (TONR)

The Retentive On-Delay timer (TONR) functions in a similar manner to the On-Delay timer (TON). There is one difference.

The Retentive On-Delay timer times as long as the enabling input is on, but does not reset when the input goes off. The timer must be reset with a RESET (R) instruction The same example used with the On-Delay timer will be used with the Retentive On-Delay timer. When the switch is closed at input I0.3, timer T5 (Retentive timer) begins timing. If, for example, after 10 seconds input I0.3 is opened the timer stops. When input I0.3 is closed the timer will begin timing at 10 seconds. The light will turn on 5 seconds after input I0.3 has been closed the second time. A RESET (R) instruction can be added. Here a pushbutton is connected to input I0.2. If after 10 seconds input I0.3 were opened, T5 can be reset by momentarily closing input I0.2. T5 will be reset to 0 and begin timing from 0 when input I0.3 is closed again.

Off-Delay (TOF) The Off-Delay timer is used to delay an output off for a fixed period of time after the input turns off. When the enabling bit turns on the timer bit turns on immediately and the value is set to 0. When the input turns off, the timer counts until the preset time has elapsed before the timer bit turns off.

S7-200 Timers The S7-200s have 256 timers. The specific T number chosen for the timer determines its time base and whether it is TON, TONR, or TOF.

Timer Example

In the following example a tank will be filled with two chemicals, mixed, and then drained. When the Start Button is pressed at input I0.0, the program starts pump 1 controlled by output Q0.0. Pump 1 runs for 5 seconds, filling the tank with the first chemical, then shuts off. The program then starts pump 2, controlled by output Q0.1. Pump 2 runs for 3 seconds filling the tank with the second chemical. After 3 seconds pump 2 shuts off. The program starts the mixer motor, connected to output Q0.2 and mixes the two chemicals for 60 seconds. The program then opens the drain valve controlled by output Q0.3, and starts pump 3 controlled by output Q0.4. Pump 3 shuts off after 8 seconds and the process stops. A manual Stop switch is also provided at input I0.1.

8.1.2 Counters

Counters used in PLCs serve the same function as mechanical counters. Counters compare an accumulated value to a preset value to control circuit functions. Control applications that commonly use counters include the following:

- Count to a preset value and cause an event to occur
- Cause an event to occur until the count reaches a preset value

A bottling machine, for example, may use a counter to count bottles into groups of six for packaging. Counters are represented by boxes in ladder logic. Counters increment/decrement one count each time the input transitions from off (logic 0) to on (logic 1). The counters are reset when a RESET instruction is executed. S7-200 uses three types of counters: up counter (CTU), down counter (CTD), and up/down counter (CTUD).

S7-200 Counters

There are 256 counters in the S7-200, numbered C0 through C255. The same number cannot be assigned to more than one counter. For example, if an up counter is assigned number 45, a down counter cannot also be assigned number 45. The maximum count value of a counter is ±32,767.

Up Counter

The up counter counts up from a current value to a preset value (PV). Input CU is the count input. Each time CU transitions from a logic 0 to a logic 1 the counter increments by a count of 1. Input R is the reset. A preset count value is stored in PV input. If the current count is equal to or greater than the preset value stored in PV, the output bit (Q) turns on (not shown).

Down Counter

The down counter counts down from the preset value (PV) each time CD transitions from a logic 0 to a logic 1. When the current value is equal to zero the counter output bit (Q) turns on (not shown). The counter resets and loads the current value with the preset value (PV) when the load input (LD) is enabled.

Up/Down Counter

The up/down counter counts up or down from the preset value each time either CD or CU transitions from a logic 0 to a logic 1. When the current value is equal to the preset value, the output QU turns on. When the current value (CV) is equal to zero, the output QD turns on. The counter loads the current value (CV) with the preset value (PV) when the load input (LD) is enabled. Similarly, the counter resets and loads the current value (CV) with zero when the reset (R) is enabled. The counter stops counting when it reaches preset or zero.

Counter Example

A counter might be used to keep track of the number of vehicles in a parking lot. As vehicles enter the lot through an entrance gate, the counter counts up. As vehicles exit the lot through an exit gate, the counter counts down. When the lot is full a sign at the entrance gate turns on indicating the lot is full. Up/down counter C48 is used in this example. A switch, connected to the entrance gate, has been wired to input I0.0.

A switch, connected to the exit gate, has been wired to inputI0.1. A reset switch, located at the collection booth, has been wired to input I0.2. The parking lot has 150 parking spaces. This value has been stored in the preset value (PV). The counter output has been directed to output Q0.1. Output 2 is connected to a "Parking Lot Full" sign. As cars enter the lot the entrance gate opens. Input I0.0 transitions from a logic 0 to a logic 1, incrementing the count by one. As cars leave the lot the exit value, the output QU turns on. When the current value (CV) is equal to zero, the output QD turns on. The counter loads the current value (CV) with the preset value (PV) when the load input (LD) is enabled. Similarly, the counter resets and loads the current value (CV) with zero when the reset (R) is enabled. The counter stops counting when it reaches preset or zero.

Counter Example

A counter might be used to keep track of the number of vehicles in a parking lot. As vehicles enter the lot through an entrance gate, the counter counts up. As vehicles exit the lot through an exit gate, the counter counts down. When the lot is full a sign at the entrance gate turns on indicating the lot is full. Up/down counter C48 is used in this example. A switch, connected to the entrance gate, has been wired to input I0.0.

A switch, connected to the exit gate, has been wired to inputI0.1. A reset switch, located at the collection booth, has been wired to input I0.2. The parking lot has 150 parking spaces. This value has been stored in the preset value (PV). The counter output has been directed to output Q0.1. Output 2 is connected to a "Parking Lot Full" sign. As cars enter the lot the entrance gate opens. Input I0.0 transitions from a logic 0 to a logic 1, incrementing the count by one. As cars leave the lot the exit gate opens. Input I0.1 transitions from a logic 0 to a logic 1, decrementing the count by 1. When the count has reached 150 output Q0.1 transitions from a logic 0 to a logic 1. The "Parking Lot Full" sign illuminates. When a car exits, decrementing the count to 149, the sign turns off.

CHAPTER 9

9 HIGH SPEED INSTRUCTIONS

As discussed earlier, PLCs have a scan time. The scan time depends on the size of the program, the number of I/Os, and the amount of communication required. Events may occur in an application that require a response from the PLC before the scan cycle is complete. For these applications high-speed instructions can be used.

Positioning is one example of an application that can use high-speed counters. In the following illustration a motor is connected through a starter to a PLC output. The motor shaft is connected to an encoder and a positioning actuator. The encoder emits a series of pulses as the motor turns. In this example the program will move an object from position 1 to position 6. Assume the encoder generates 600 pulses per revolution, and it takes 1000 motor revolutions to move the object from one position to another. To move the object from position 1 to position 6 (5 positions) would take 5000 motor revolutions. The counter would count up 30,000 counts (5000 revolutions x 600 pulses per revolution) and stop the motor.

Interrupts

Interrupts are another example of an instruction that must be executed before the PLC has completed the scan cycle. Interrupts in the S7-200 are prioritized in the following order:

1. Communications
2. I/O Interrupts
3. Time-Based Interrupts

PTO

Pulse train output (PTO) is used to provide a series of pulses to an output device, such as a stepper motor driver. The PTO provides a square wave output for a specified number of pulses and a specified cycle time. The number of pulses can be from 1 to 4,294,967,295 pulses. PTOs have a 50% duty cycle. This means the pulse is off for the same amount of time it is on. The number of pulses and the cycle time can be changed with an interrupt. In the following example each pulse is on for 500 ms, and off for 500 ms. After four pulses an interrupt occurs which changes the cycle time to 1000 ms.

operation to be varied to compensate for product variations or mechanical wear.

Transmit

Transmit allows communication with external devices, such as modems, printers, computers, via the serial interface. See the section titled "Connecting External Devices" for examples.

CHAPTER 10

10 SPECIALIZED EXPANSION MODULES

In addition to I/O modules, expansion modules are also available for the S7-200 measure temperature, control positioning applications, and provide various communication functions.

EM 241 In any complex system rapid communication is essential.
Modems are electronic devices used for sending and receiving data over long distances. The EM 241 is an expansion module that supports communication between an S7-200 PLC and STEP 7 Micro/WIN via a modem.The EM 241 provides an international telephone line interface, supports sending numeric and text paging messages, as well as SMS (Short Message Service) messages to cellular phones. This is useful for remote diagnostics and maintenance, machine control, alarm systems, and general communication functions. In addition to CPU-to-CPU communication via a telephone line, the EM 241 also supports the ModBus RTU protocol. Protocols are rules that identify how devices should communicate with each other. ModBus RTU is a protocol originally developed by MODICON, which is now part of Schneider Automation. ModBus RTU has been widely used by other companies.

EM 277 Information flow between intelligent devices such as PLCs,
computers, variable speed drives, actuators, and sensors is often accomplished through a local area network (LAN). LANs are used in office, manufacturing, and industrial areas. In the past, these networks were often proprietary systems

addition to CPU-to-CPU communication via a telephone line, the EM 241 also supports the ModBus RTU protocol. Protocols are rules that identify how devices should communicate with each other. ModBus RTU is a protocol originally developed by MODICON, which is now part of Schneider Automation. ModBus RTU has been widely used by other companies.

EM 277 Information flow between intelligent devices such as PLCs, computers, variable speed drives, actuators, and sensors is often accomplished through a local area network (LAN). LANs are used in office, manufacturing, and industrial areas. In the past, these networks were often proprietary systems designed to a specific vendor's standards. Siemens has been a leader in pushing the trend to open systems based upon international standards developed through industry associations. PROFIBUS-DP and Actuator Sensor Interface (ASi) are examples of these open networks. The PROFIBUS-DP EM 277 module allows connection of the S7-200 CPU to a PROFIBUS-DP network as a slave. The CP 243-2 Communication Processor allows communication between AS-i devices and an S7-200.

PROFIBUS DP PROFIBUS DP is an open bus standard for a wide range of applications in various manufacturing and automation processes. PROFIBUS DP works at the field device level such as power meters, motor protectors, circuit breakers, and lighting controls. Through PROFIBUS DP the features of S7-200 PLCs can be used to their full extent within a distributed system. An advantage to PROFIBUS DP is the ability to communicate between PROFIBUS DP devices of different vendors. This provides uniform communication between all SIMATIC devices on the PROFIBUS DP network as well as devices from other manufacturers.

AS-i Actuator Sensor Interface (AS-i or AS-Interface) is a system for networking binary devices such as sensors. Until recently, extensive parallel control wiring was needed to connect sensors to the controlling device. AS-i replaces complex wiring with a simple 2-core cable. The cable is designed so that devices can only be connected correctly. Several devices can be connected to the cable. PLCs, for example, use I/O modules to receive inputs from binary devices such as sensors. Binary outputs are used to turn on or off a process as the result of an input.

EM 253 Position control describes a range of applications that involve movement with varying degrees of precision. Rotary tables and traversing cars are examples where objects are moved from one position during a product's manufacturing process. The EM 253 is a positioning module that enables the user to control the speed and position for either stepper motors or servo motors. The EM 253 interfaces between an S7-200 PLC and the stepper/servo motor's power control module.

EM 253 Features The EM 253 provides functionality for single-axis, open-loop position control. Features of the module include:

- High-speed control with a range of 12 - 200,000 pulse per second
- Jerk (S curve) or linear acceleration/deceleration
- Configurable measuring system to enter data as engineering units (such as inches or centimeters) or number of pulses
- Configuarble backlash compensation
- Supports absolute, relative, and manual methods of position control
- Continuous operation
- Provides up to 25 motion profiles with up to 4 speed changes per profile
- Four different reference-point seek modes, with a choice of the starting seek direction and final approach direction for each sequence

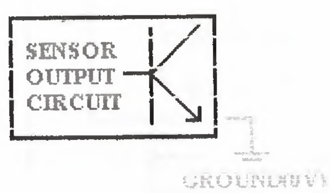
10.1 Dc Inputs

Let's now take a look at how the input circuits of a PLC work. This will give us a better understanding of how we should wire them up.Typically, DC input modules are available that will work with 5, 12,24, and 48 volts.

We'll first look at how the dc inputs work. DC input modules allow us to connect either PNP (sourcing) or NPN (sinking) transistor type devices to them. If we are using a regular switch (i.e. toggle or pushbutton,etc.) we typically don't have to worry about whether we wire it as NPN or PNP. We should note that most PLCs won't let us mix NPN and PNP devices on the same module. When we are using a sensor (photoeye, proximity sensor, etc.) we do, however, have to worry about its output configuration.

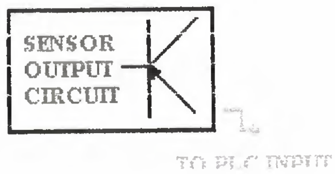
The difference between the two types is whether the load (in our case, the PLC is the load) is switched to ground or positive voltage. An NPN type sensor has the load switched to ground whereas a PNP device has the load switched to positive voltage. Below is what the outputs look like for NPN and PNP sensors.

NPN(SINKING)SENSOR



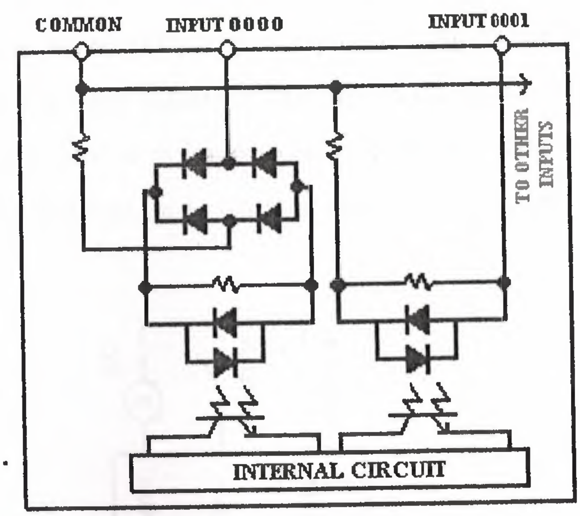
On the NPN sensor we connect one output to the PLCs input and the other output to the power supply ground. If the sensor is not powered from the same supply as the PLC, we should connect both grounds together. Engineers will say that PNP is better (i.e. safer) because the load is switched to ground.

On the PNP sensor we connect one output to positive voltage and the other output to the PLCs input. If the sensor is not powered from the same supply as the PLC, we should connect both V+'s together. PNP sensors are most commonly used in Europe.



PNP(SOURCING)SENSOR

Inside the sensor, the transistor is just acting as a switch. The sensors internal circuit tells the output transistor to turn on when a target is present. The transistor then closes the circuit between the 2 connections shown above. (V+ and PLC input).



The only things accessible to the user are the terminals labeled COMMON, INPUT 0000, INPUT 0001, INPUTxxxx... The common terminal either gets connected to V+ or ground. Where its connected depends upon the type of sensor used. When using an NPN sensor this terminal is connected to V+. When using a PNP sensor this terminal is connected to 0V (ground).

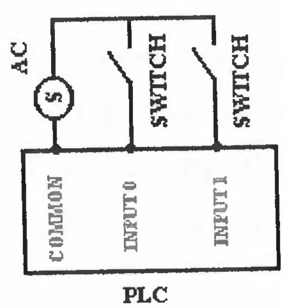
A common switch (i.e. limit switch, pushbutton, toggle, etc.) would be connected to the inputs in a similar fashion. One side of the switch would be connected directly to V+. The other end goes to the PLC input terminal. This assumes the common terminal is connected to 0V (ground). If the common is connected to V+ then simply connect one end of the switch to 0V (ground) and the other end to the PLC input terminal.

The optocouplers are used to isolate the PLCs internal circuit from the inputs. This eliminates the chance of any electrical noise entering the internal circuitry. They work by converting the electrical input signal to light and then by converting the light back to an electrical signal to be processed by the internal circuit.

10.2 Ac ınputs

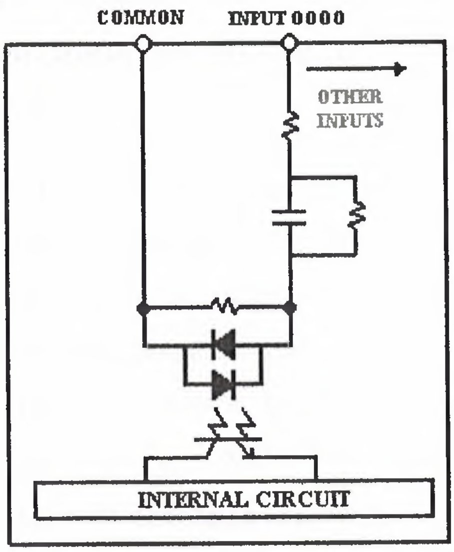
Now that we understand how dc inputs work, let's take a close look at ac inputs. An ac voltage is nonpolarized. Typically, AC input modules are available that will work with 24, 48, 110, and 220 volts.

AC input modules are less common these days than dc input modules. The reason being that today's sensors typically have transistor outputs. A transistor will not work with an ac voltage. Most commonly,the ac voltage is being switched through a limit switch or other switch type. If your applicatipn is using a sensor it probably is operating on a dc voltage.



We typically connect an ac device to our input module as shown above. Commonly the ac "hot" wire is connected to the switch while the "neutral" goes to the PLC common. The ac ground (3rd wire where applicable) should be connected to the frame ground terminal of the PLC.(not shown) As is true with dc, ac connections are typically color coded so that the individual wiring the device knows which wire is which. This coding varies from country to country but in the US is commonly white (neutral), black (hot) and green (3rd wire ground when applicable). Outside the US it's commonly coded as brown (hot), blue (neutral) and green with a yellow stripe (3rd wire ground where applicable).

The PLCs ac input module circuit typically looks like this:



The only things accessible to the user are the terminals labeled COMMON, INPUT 0000, INPUTxxxx... The common terminal gets connected to the neutral wire.

A common switch (i.e. limit switch, pushbutton, toggle, etc.) would be connected to the input terminals directly. One side of the switch would be connected directly to INPUT XXX. The other end goes to the ac hot wire. This assumes the common terminal is connected to neutral. Always check the manufacturers specifications before wiring, to be sure AND SAFE.

The optocouplers are used to isolate the PLCs internal circuit from the inputs. This eliminates the chance of any electrical noise entering the internal circuitry. They work by converting the electrical input signal to light and then by converting the light back to an electrical signal to be processed by the internal circuit.

One last note, typically an ac input takes longer than a dc input for the PLC to see. In most cases it doesn't matter to the programmer because an ac input device is typically a mechanical switch and mechanical devices are slow. It's quite common for a PLC to require that the input be on for 25 or more milliseconds before it's seen. This delay is required because of the filtering, which is needed by the PLC internal circuit.

10.3 Transistor Outputs

The next type of output we should learn about is our transistor type outputs. It is important to note that a transistor can only switch a dc current. For this reason it cannot be used with 'an AC voltage.

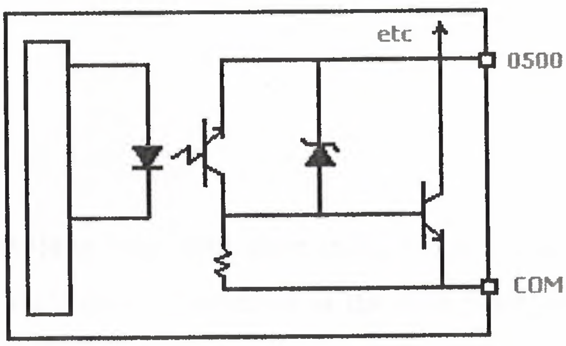
We should also keep in mind that as we saw before with the input circuits, there are generally more than one type of transistor available. Typically a PLC will have either NPN or PNP type outputs. The physical type of transistor used also varies from manufacturer to manufacturer. Some of the common types available are BJT and MOSFET. A BJT type (Bipolar Junction Transistor) often has less switching capacity (i.e. it can switch less current) than a MOS-FET (Metal Oxide Semiconductor- Field Effect Transistor) type. The BJT also has a slightly faster switching time.

Shown above is how we typically connect our output device to the transistor output. Please note that this is an NPN type transistor. If it were a PNP type, the common terminal would most likely be connected to V+ and V- would connect to one end of our load. Note that since this is a DC type output we must always observe proper polarity for the output. One end of the load is connected directly to V+ as shown above.

Let's take a moment and see what happens inside the output circuit. Shown below is a typical output circuit diagram for an NPN type output.

Notice that as we saw with the transistor type inputs, there is a photocoupler isolating the "real world' from the internal circuit. When the ladder diagram calls for it, the internal circuit turns on the photocoupler by applying a small voltage to the LED side of the photocoupler. This makes the LED emit light and the receiving part of the photocoupler will see it and allow current to flow. This small current will turn on the base of the output transistor connected to output 0500. Therefore, whatever is connected between COM and 0500 will turn on. When the ladder tells 0500 to turn off, the LED will stop emitting light and hence the output transistor connected between 0500 and COM will turn off. One other important thing to note is that a transistor typically cannot switch as large a load as a relay. Check the manufacturers specifications to find the largest load it can safely switch. If the load current you need to switch exceeds the specification of the output, you can connect the PLC output to an external relay. Then connect the relay to the large load. You may be thinking, "why not just use a relay in the first place"? The answer is because a relay is not always the correct choice for every output. A transistor gives you the opportunity to use external relays when and only when necessary.

In summary, a transistor is fast, switches a small current, has a long lifetime and works with dc only. Whereas a relay is slow, can switch a large current, has a shorter lifetime and works with ac or dc.



CHAPTER 11

11 GRADUATION PROJECT

11.1 Three floor system

In my project I made a three floor sytem and an elevator.There calling sensors,floor limit sensors for the inputs.At the output part there are three different colour leds,cabin lamp and a bobbin dependent to the ac motor.Sytem is working perfect.There is no fault.

11.1.1 Input part

At the input part there are three calling sensors and three floor limit sensors.Floor limit sensors are the switches that when pressing a calling sensor they are blocked off at that instant up to other operation.

11.1.2 Output part

At the output part there are three leds,cabin lamp and a motor.I used the leds for discussing the elevator's stair condition.When the cabin is at the first floor the red leds will start to operate.For the second floor condition the green leds will be active and last the yellow leds will be active dependent to third floor.When the door open cabin lamp will be active by the time.

11.2 Explanation of the networks that will be used in the operation

NETWORK-1:

When somebody presses to first floor limit calling push button,elevator goes to first floor from which it is present,up to open first floor limit sensor I.03.

11.2 Explanation of the networks that will be used in the operation

NETWORK-1:

When somebody presses to first floor limit calling push button,elevator goes to first floor from which it is present,up to open first floor limit sensor I.03.

NETWORK-2:

When somebody presses to second floor calling push button,if elevator present at first floor it will go to second floor,up to open second floor limit sensor I.04.

NETWORK-3:

When somebody presses to second floor calling push button,if elevator present at third floor it will go to second floor,up to open second floor limit sensor I.04.

NETWORK-4:

When somebody presses to third floor calling push button,which floor that elevator present,it will go to third floor,up to open third floor limit sensor I.05.

NETWORK-5:

When one of the memory contacts M0.3 or M0.5 active,motor moves towards top,up to see limit switch that present at which floor.When door of the elevator open at same time there is no any movement at motor.

NETWORK-6:

When one of the memory contacts M0.2 or M0.4 active,motor moves towards bottom,up to see limit switch that called out.When door of the elevator open,there is no any movement at motor.

NETWORK-7:

Motor that works to any direction or motor does not work,sccording to opening and closing condition of door lamp to be on 15 second.

NETWORK-8:

Motor that works to any direction,M0.0 memory contact operates 15 second and even motors stop,lamp will be active 15 second.

NETWORK-9:

After motor stops M0.0 memory contact will be active and lamp will operate 15 second.

NETWORK-10:

When door will be opened,T38 timer becomes active and lamp operates 20 second.

NETWORK-11:

When door will be closed and motor does not work T39 timer will be active and lamp operates 20 second.

NETWORK-12:

When T38 timer will be active and motor does not operates M01 memory contac becomes active extinguishes lamp 20 second later.

NETWORK-13:

When first floor limit sensor I.0.3 will be active red leds start to operate.If cabin stop at floor led operates contiunal or if cabin leaves from floor le operates 10 second.

NETWORK-14:

When second floor limit sensor I.0.4 will be active green leds start to operate.If cabin stop at floor led operates contiunal or if cabin leaves from floor led operates 10 second.

NETWORK-15:

When third floor limit sensor I.0.5 will be acive yellow leds start to operate.If cabin leaves from floor led operates 10 second.

NETWORK-16:

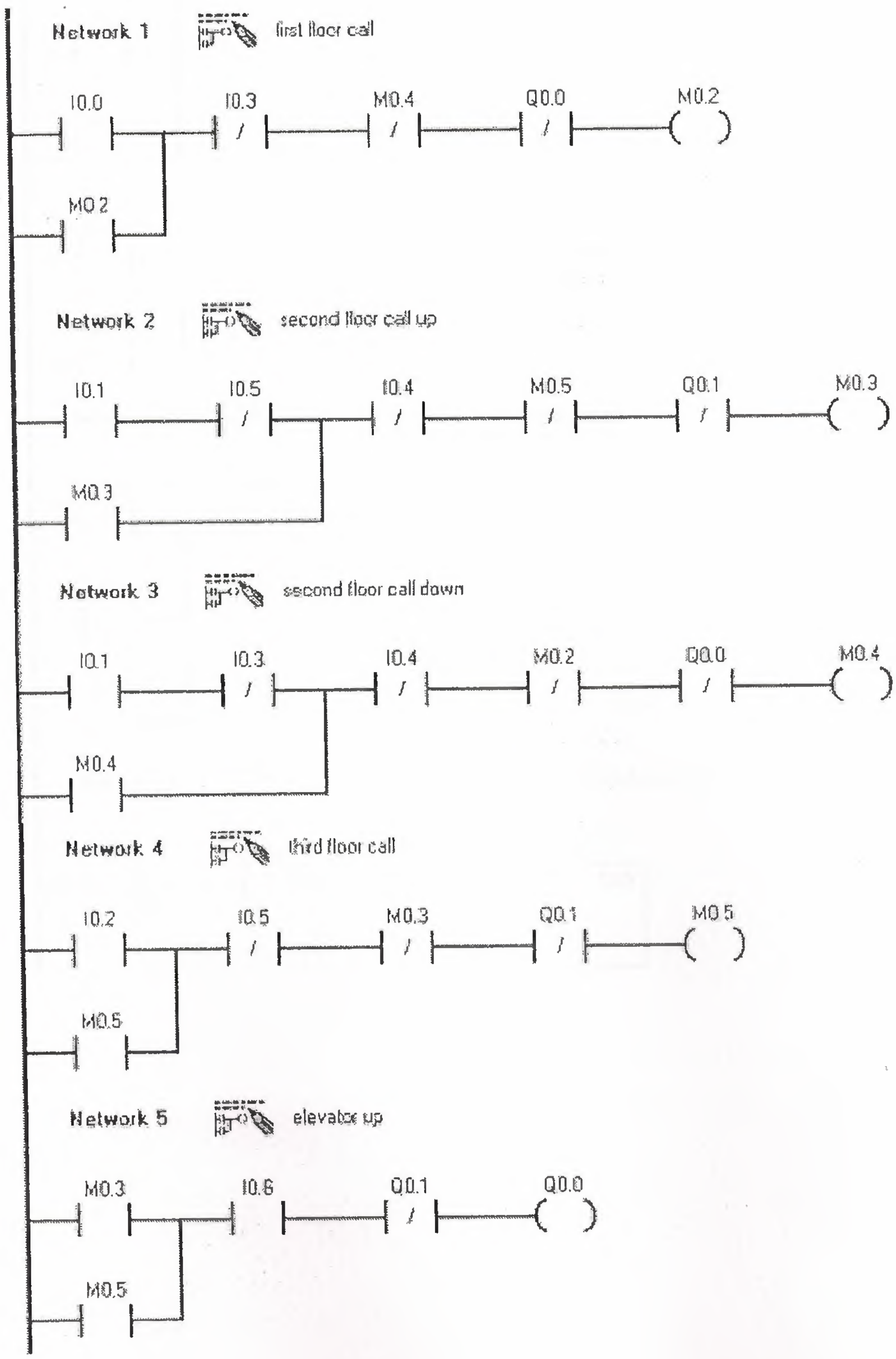
When any of floor lamps starts to operate,T40 timer will be active and changes contacts 10 second later.According to position of cabin,floor closes to determined lights.

NETWORK-17:

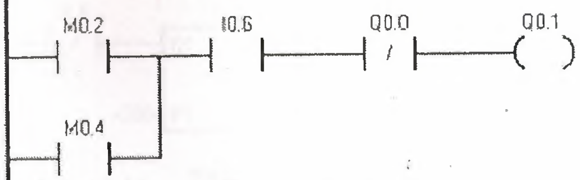
Program stop.

4.3 Lader Diagram and Statement List of the Program

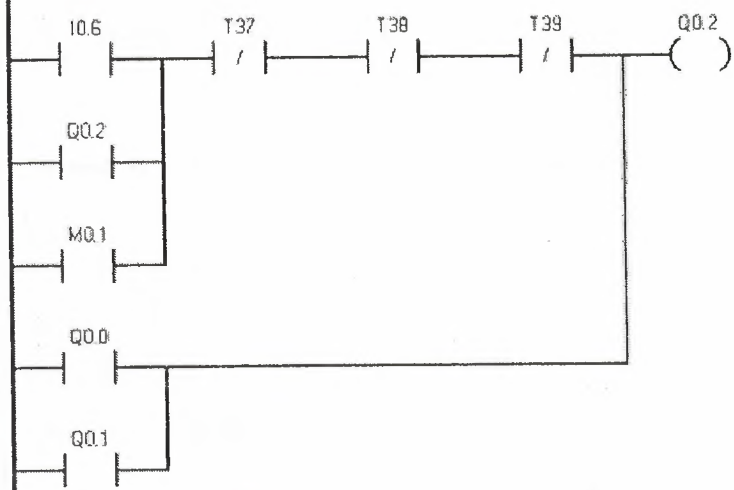
a) Ladder Diagram



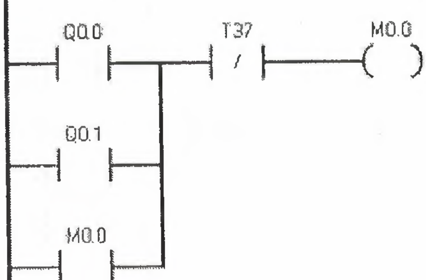
Network 6 elevator down



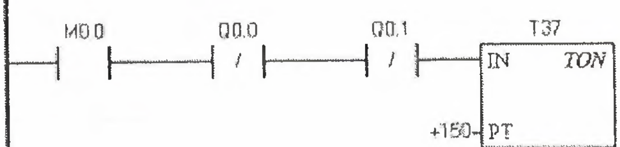
Network 7 elevator lamp control



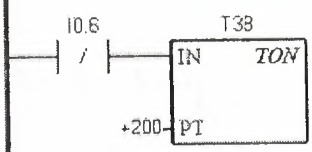
Network 8 lamp control when elevator move and stop



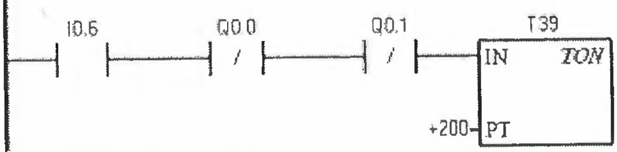
Network 9 when elevator stop,provide the delay for lamp of cabinet



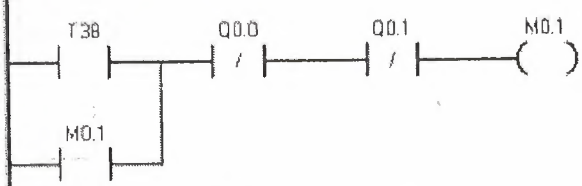
Network 10 when the door is opened light will be on for 20 seconds



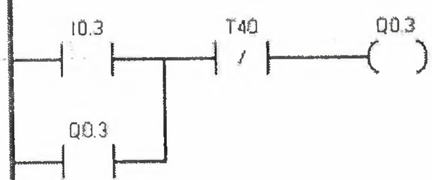
Network 11 when the door close the light is on for 20 seconds



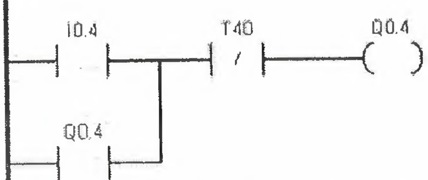
Network 12 when you leave the cabinet light will be off after 20 seconds



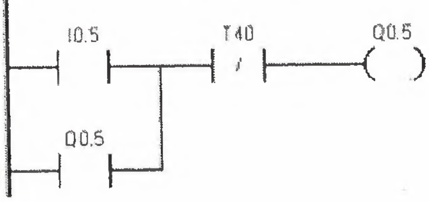
Network 13 first floor indicator lamp will be on



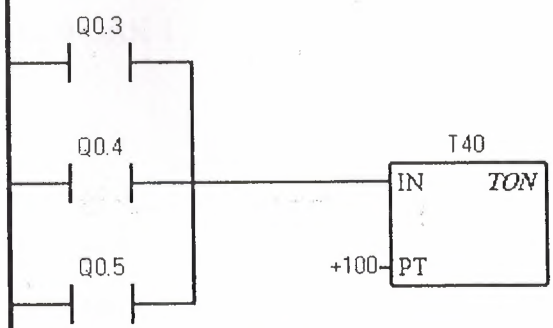
Network 14 second floor indicator lamp will be on



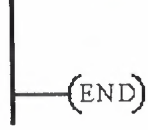
Network 15 third floor indicator lamp will be on



Network 16 when you live any floor the indicator lamp will be off after 10 seconds



Network 17 mend



b) Statement List

NETWORK 1 First floor call

```
LD    I0.0
O     M0.2
AN    I0.3
AN    M0.4
AN    Q0.0
=     M0.2
```

NETWORK 2 Second floor call up

```
LD    I0.1
AN    I0.5
O     M0.3
AN    I0.4
AN    M0.5
AN    Q0.1
=     M0.3
```

NETWORK 3 Second floor call down

```
LD    I0.1
AN    I0.3
O     M0.4
AN    I0.4
AN    M0.2
AN    Q0.0
=     M0.4
```

NETWORK 4 Third floor call

```
LD    I0.2
```

```
O    M0.5
AN   I0.5
AN   M0.3
AN   Q0.1
=    M0.5
```

NETWORK 5   Elevator up

```
LD   M0.3
O    M0.5
A    I0.6
AN   Q0.1
=    Q0.0
```

NETWORK 6   Elevator down

```
LD   M0.2
O    M0.4
A    I0.6
AN   Q0.0
=    Q0.1
```

NETWORK 7   Elevator lamp control

```
LD   I0.6
O    Q0.2
O    M0.1
AN   T37
AN   T38
AN   T39
LD   Q0.0
O    Q0.1
OLD
=    Q0.2
```

NETWORK 8 Lamp control when elevator move and stop

```
LD    Q0.0
O     Q0.1
O     M0.0
AN    T37
=     M0.0
```

NETWORK 9 When elevator stop,provide the delay for lamp of cabinet

```
LD    M0.0
AN    Q0.0
AN    Q0.1
TON   T37, +150
```

NETWORK 10 When the door is opened light will be on for 20 seconds

```
LDN   I0.6
TON   T38, +200
```

NETWORK 11 When the door close the light is on for 20 seconds

```
LD    I0.6
AN    Q0.0
AN    Q0.1
TON   T39, +200
```

NETWORK 12 When you leave the cabinet light will be off after 20 seconds

```
LD    T38
O     M0.1
AN    Q0.0
AN    Q0.1
```

```
= M0.1
```

NETWORK 13 First floor indicator lamp will be on

```
LD  I0.3
O   Q0.3
AN  T40
=   Q0.3
```

NETWORK 14 Second floor indicator lamp will be on

```
LD  I0.4
O   Q0.4
AN  T40
=   Q0.4
```

NETWORK 15 Third floor indicator lamp will be on

```
LD  I0.5
O   Q0.5
AN  T40
=   Q0.5
```

NETWORK 16 When you live any floor the indicator lamp will be off after 10 seconds

```
LD  Q0.3
O   Q0.4
O   Q0.5
TON T40, +100
```

NETWORK 17 Program stop

```
MEND
```

REFERENCES

Warren,H.I.,Control Sytems,Prentice-Hall Inc.,Englewood Cliffs,NJ,1998

Ali Osman Özkan,Muciz Özcan Nobel Yayın Dağıtım Ankara,2002,1.baskı

http://www.plcprogramlama.com

http://www.tr.wikipedia.org/wiki/PLC

http://www.elektrotekno.com

http://www.mekatronikklubu.org