NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

Automation of a Robotic System with PLC

Graduation Project

EE- 400

Student: Hüseyin Önder (961147)

Supervisors: Asst.Prof.Dr.Kadri Bürüncük

Mr. Özgür C.Özerdem

Lefkoşa – 2001

Table of Contents

ACKNOWLEDGEMENTS

I must first acknowledgements some depts, from more distance past. To my parents for their support to an upwardly mobile son. To my friends from to our departments Cemal Kavalcıoğlu, Burak Başeğmezler, Murtaza M. Deniz, Erden Sezer, Murat Yaşar Ertaş, Latif Nizipli , Haşim Üçler and from the Civil Engineering Faculty Erol Kaptı and Aydın Özdemir for his helps.

I want to thanks seperately to my instructors that during my education they support me the Dean of Engineering Faculty Prof. Dr. Fakhreddin MAMEDOV, and everytime from near to me, my dear instructors Mr.Özgür C. ÖZERDEM, Assist. Prof. Dr. Kadri BÜRÜNCÜK, Assist.Prof.Dr. Şenol BEKTAŞ, Prof.Haldun GÜRMEN, Mr. Mehrdad KHALEDI, Dr. Doğan HAKTANIR, and our department secretaries Mrs. Meryem PAŞA, and Mrs. Ayşe YÜRÜN.

And finally, I want to thanks everybody who helped and supported me come today's and graduated from a University.

ABSTRACT

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

The hardware are the parts of machine those are CPU,I/O device and programming device. CPU is basic microprocessor system and it carries out as control sensor,counter,timer function. CPU carries out stored user program in memory will input informations from various sensor circuits and can sending suitable output to commands and control circuits. I/O module receives 120 VAC signal in device or processing device and transform 5 VDC signal form.

There are many specialisation such as timer,counter,master control set,which works data and controls program, master control reset,JMP. Ther are command which are mathematics process that are comparator process. These are the main function and feature of software part of PLC.

INTRODUCTIONS

Now that we understand how inputs and outputs are processed by the PLC, let's look at a variation of our regular outputs. Regular output coils are of course an essential part of our programs but we must remember that they are only true when all instructions before them on the rung are also true.

Think back to the we did a few chapters ago. What would've happened if we couldn 't find a "push on/push off" switch. Then we would've had to keep pressing the button - for as long as we wanted the bell to sound. (A momentary switch) The latching instructions let us use momentary switches and program the PLC so that when we push one the output turns on and when we push another the output turns off. Picture the remote control for your TV .It has a button for on and another for off. When L push the on button the TV turns on. When L push the off button the TV turns off. L don't have to keep pushing the on button to keep the TV on. This would be the function of a latching instruction.The latch instruction is often called a SET or OTL (output latch). The unlatch instruction is often called a RES (reset), OUT (output unlatch) or RST (reset).

CHAPTER 1 : WHAT IS A PLC

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

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

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

We can simply program the PLC to count its inputs and turn the solenoids on for the specified time. This site gives you enough information to be able to write programs far more complicated than the simple one above. We will take a look at what is considered to be the "top 20" PLC instructions. It can be safely estimated that with a firm understanding of these instructions one can solve more than 80% of the applications inexistence.

CHAPTER 2: PLC HISTORY

In the late 1960's PLC 's were first introduced. The primary reason for designing such a device was eliminating the large cost involved in replacing the complicated relay based machine control systems. Bedford Associates (Bedford, MA) proposed something called a Modular Digital Controller (MODICON) to a major US car manufacturer. Other companies at the time proposed computer based 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 1edjous when so many relays are involved. Now picture a machine control pane1 that included many, possibly hundreds or thousands, of individual relays. The size could be mind boggling. How about the complicated initial wiring of so many individual devices! These relays would be individually wired together in a manner that would yield the desired outcome.

These "new controllers" also had to be easily programmed by maintenance and plant engineers. The lifetime had to be long and programming changes easily performed. They also had to survive the harsh industrial environment. That's a lot to ask! The answers were 10 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 PLC 's. Conventional microprocessors lacked the power to quickly solve PLC logic in all but the smallest PLC 's. As conventional microprocessors evolved, larger and larger PLC 's were being based upon them. However, even today some are still based upon the 2903.(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 PLC 's and they could be far away from the actual machine they were controlling. They could also now be used to send and receive varying voltages to allow them to enter the analog world. Unfortunately, the lack of

standardisation coupled with continually changing technology has made PLC communications a nightmare of incompatible protocols and physical networks.
The 80's saw an attempt to standardise 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.

The 90's have seen a gradual reduction in the introduction of new protocols, and the modernisation of the physical layers of some of the more popular protocols that survived the 1980's. The latest standard has tried to merge PLC-programming languages under one international standard. W e now have PLC ' s 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 PLC 's in some applications. The original company who commissioned the MODICON 084 has actually switched to a PC based control system.

CHAPTER 3 : GENAREL PHYSICAL BUILD MECHANISM

PLC 's are separated into two according to their building mechanisms.
The S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. The compact design, expandability, low cost,and powerful instruction set of the S7-200 Micro PLC make a perfect solution for controlling small applications. In addition, the wide variety of CPU sizes and voltages provides you with the flexibility you need to solve your automation problems.

Equipment Requirements

S7-200 Micro PLC system, which includes an S7-200 CPU module, a personal computer, STEP 7-Micro/WIN programming software, and a communications cable. In order to use a personal computer (PC), you must have one of the following sets of equipment:

1. A PC/PPI cable
2. A communications processor (CP) card and multipoint interface (MPI) cable
3. A multipoint interface (MPI) card. A communications cable is provided with the MPI card.

Capabilities of the S7-200 CPUs

The S7-200 family includes a wide variety of CPUs. This variety provides a range of features to aid in designing a cost-effective automation solution.

3.1. Compact PLC 's

Compact PLC 's are manufactured such that all units forming the PLC are placed in a case. They are low price PLC with lower capacity. They are usually preferred by small or medium size machine manufacturers. in some types compact enlargement module is present.

3.2. Modular PLC 's

They are formed by combining separate modules (called RACK) together in a board. They can have different memory capacity , I/0 numbers, Power Supply up to the necessary limits.

Some examples: SIEMENS S5-1 15U, KLOCKNER-MOELLER PS316 OMRON C200H.

CHAPTER 4 : INTERNAL STRUCTURE OF PLC 's:

They have three main units:

1. Input unit
2. Processing unit
3. Output unit

4.1. INPUT UNIT:

Is the that converts the signals coming from the control elements of the system that is going to be contro lled into logic levels. The analog and/or digital signals coming from the sensors or switches showing the systems pressure, humidity, level, etc. enters the PLC through the input unit. Digital signals are converted to 5V dc by this unit which is the internal voltage level of the device.

The parasitic signals are first filtered by RC passive filters and than they pass through up to coupler that has the property to supply galvanised isolation. As a result of this process the signals are send to input display memory .Analog signals pass through this process the signals are send to input display memory .Analog signals pass through frequency converts in some PLC 's. In this way they gain important noise immunity.

4.2. OUTPUT UNITS:

They are suitably manufactured to successfully control the activators in the system to be contro lled. Digital output signals contractor relays, 24 V dc NPN or PNP transistors or Tracs, PLC 's output cannot supply large currents. So by digital output relays and by their contactor groups main contactors or windings are operated. In this way unit like motors, heaters, hydraulic values can be operated.

4.3. PROCESSING UNIT:

It is composed of the sub units given below:

a) CPU (Central Processing Unit) :

It is also given the name processing unit. It processes all the input signals according to the user program instruction order and directs the output signals to the related outputs. This process is contro lled by a microprocessor. Some times instead of microprocessor a micro contro ller or microcomputer can also be-used. The difference of these devices from microprocessor Is that processor; memory and VO interfaces are a ll in one unit. As a memory ROM and RAM is used. Data for Operating System and PLC that cannot be changed are kept in ROM and user program and VQ data are kept in RAM.

b) Memory Retention for the S7-200 CPU

The S7-200 CPU provides several methods to ensure that your program, the program data, and the configuration data for your CPU are properly retained:

- The CPU provides an EEPROM to store permanently all of your program, selected data areas, and the configuration data for your CPU.
- The CPU provides a super capacitor that maintains the integrity of the RAM after power has been removed from the CPU. Depending on the CPU module, the super capacitor can maintain the RAM for several days.
- Some CPU modules support an optional battery cartridge that extends the amount of time that the RAM can be maintained after power has been removed from the CPU. The battery cartridge provides power only after the super capacitor has been drained.

This section discusses the permanent storage and retention of the data in RAM under a variety of circumstances.

c) Downloading and Uploading Your Program

Your program consists of three elements: the user program, the data block (optional), and the CPU configuration (optional). downloading the program stores these elements in the RAM area of the CPU memory. The CPU also automatically copies the user program, data block (DB1), and the CPU configuration to the EEPROM for permanent storage.

CHAPTER 5 : ADVANTAGES

5.1. ACCURACY

In relay control systems logical knowledge's carries in electro mechanical contactors, they can lose data because of mechanical errors. But PLC ' s are microprocessor based system so logical data are carried inside the processor, so that PLC 's are more accurate than relay type of controllers.

5 .2. FLEXIBILITY

When there is need of any change in control, relay type of controllers modification are hard, in PLC this change can be made with PLC programmer equipment.

5.3. COMMUNICATION

PLC' s are computer based systems. So that they can transfers their data to another PC or they can take external inputs from another PC, with this specification we can control the system were they are we can effect the system with OUT PC. With relays this is not possible.

5.4. LOGIC CONTROL OF INDUSTRIAL AUTOMATION

Everyday examp1es of these systems are machines like dishwashers, clothes washers and dryers, and elevators. In these systems, the outputs tend to be 220vac power signals to motors, solenoids, and indicator lights, and the inputs are DC or AC signals from user interface switches, motion limit switches, binary liquid level sensors, etc. Another major function in these types of controllers is timing.

5.5. REALS AND LADDER LOGIC

In the "old days" (ie. before the 1980's) these types of controllers were implemented with relays. Relays are a technology from the early days of electricity in which an electromag net activates an electrical switch. When current flows in the coil, electrically , thermally , and mechanically rugged, easy to design with, cheap, and capable of controlljng very large currents in their ou1put contacts.

Relays can be thought of as logic gates. For example, if two normally open relays are wired in series, and one end of the resulting ou1put circuit is attached to a voltage source, then the two coils form the inputs of a AND gate: only if current is flowing in BOTH input coils will current flow in the ou1put circuit A typical application in a washing machine might be to implement the rule that.

The shape of these diagrams invariably led to the name "Ladder diagrams" and "Ladder Logic" to describe them. The term "Relay Ladder Logic" (RLL) describes this logic notation. By including inter connections between the horizontal rungs, it is possible to create latches ("flip-flops") and implement state transitions. Although LL "state machines" get quite complex and are typically not designed with the convenience of finite state machine theory , they have become widely used and supported by technical workers. Because the logic was implemented in physical wiring, it was difficult to change, as new functions were required.

5.6. SYSTEM OVERVIES

A typical S7-200 system will include an S7-200 base unit which includes the central processing unit, power supply, and discrete input and output points. Expansion module contains additional input or output points and is connected to the base unit bus connectors. The central processing unit has a built-in communications port for programming or ta l king with Intelligent ASCIIdevices..

5.7 CPU OVERVIEW

The S7-200 series is a line of small, compact, micro-programmable logic controllers .- and expansion modules that can be used for a variety of programming applications. There are two types of base units in the S7-200 product line, CPU 212 and CPU 214. Each base unit comes in different models to accommodate the type of power supply , inputs and outputs you require.

5.8. ARCHIECTURE

This section relates to how the S7-200 CPU arranges data and how it executes your program during It's scan cycle.

5.9. MEMORY MAP

The memory space of the S7-200 is divided into five data areas and six data objects. T o reference a memory location for use, you must address that location. The addressing conventions allow memory to be accessed as bits, bytes, words and double words. All addresses are zero-based.

Data space is highly flexible, and it allows read and writes access to all memory areas as bits, bytes, words and double words. Data objects are the memory locations that are associated with devices (such as the current value of a counter or the temperature value of an oven). Access to data objects is more restrictive because the data object can be addressed only according to the intended use of that object.

a) Data Areas

Data memory contains variable memory, and register, and output image register, internal memory bits, and special memory bits. This memory is accessed by a byte bit convention. For example to access bit 3 of Variable Memory byte 25 you would use the address V25.3. The following table shows the identifiers and ranges for each of the data area memory types :

Area Identifier	Data Are	CPU 212	CPU 214
I	Input	I0.0 to 17. 7	10.0 to 17. 7
Q	Output	QO.O to I7.7	QO.O to Q7.7
M	Internal Memory	MO.O to Ml5. 7	MO.O to M31. 7
SM	Special Memory	SMO.O to SM 45. 7	SMO.O to SM 85. 7
V	Variable Memory	VO.OtoVl023.7 V	O.OtoV4095.7

b) Data Objects

The S7-200 has six kinds of devices with associated data: timers, counters, analog inputs, analog outputs, accumulators and high-speed counters. Each device has associated data (data objects). For example, the S7-200 has counter devices. Counters have a data value that maintains the current count value. There is also a bit value, which is set when the current value is greater than or equal to the present value. Since there are multiple devices are numbered from 0 to n. the corresponding data objects and object bits are also numbered. The following table shows the identifiers and ranges for each of the data object memory types:

Object Identifier	Object	CPU 212	CPU214
T	Timers	T0 to T63	T0 to Tl27
C	Counters	C0 to C63	C0 to C 127
AI	Analog Input	AJW0 to AJW30	AIW0 to AJW30
AQ	Analog Output	AQW0 to AQW30	AQW0toAQW30
AC	Accumulator Registers	AC0 to AC3	AC0 to AC3
HC	High-speed Counter Current	HC0	HC0 to HC2

CHAPTER 6
MPS Modular Production System

6.1. Distribution station, pneumatically controlled

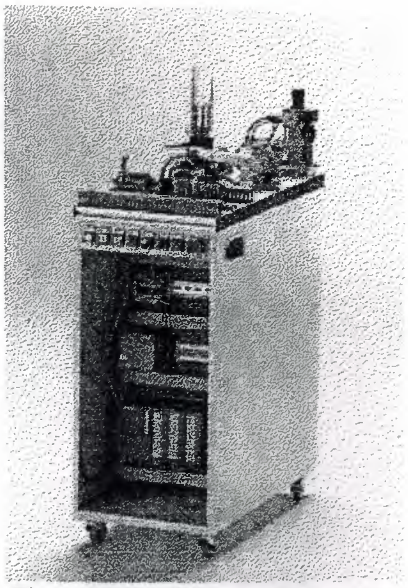


Figure 6.1 (Distribution station)

Training aims

The table in the MPS Overview chapter shows you just how versatile the station is.

Task

-Separating a part from a magazine

-Feeding separated part to a process

Design

-Stack magazine module module (Fig 7.1)

(magazine with double-acting cylinder)

-Changer module (pneumatic rotary actuator) (Fig 7.2)

Technical data

-Operating pressure min 6 bar, max 8 bar

Recommended workbooks:

-Automated Production, a manual for trainee qualification

-Training documentation Distributing Station

Notes:

-The distributing station is the ideal gateway into the Modular Production System. You can begin first of all with the symbolic workpiece and later expand step by step up to the assembly station. The stack magazine is suitable for holding symbolic workpieces and the body of the workpiece that can be assembled.

-TP 100 is required for control.

6.2.Testing station

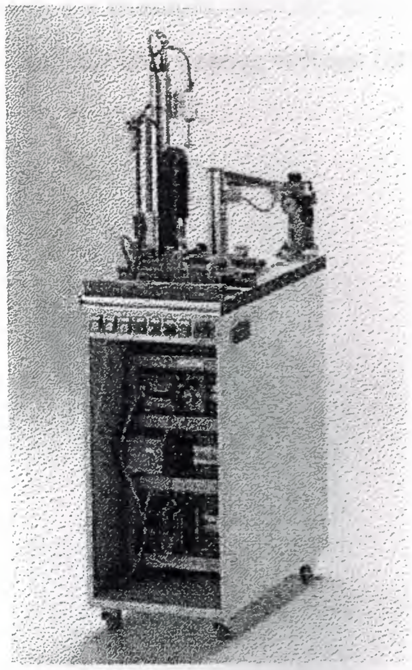


Figure 6.2 (Testing station)

Training aims

The table in the MPS Overview chapter shows you just how versatile the station is.

Task

-Recognizing materials

-Quality control (dimensional accuracy)

Design

-Recognition module (Fig 7.3)

-Lifting module (Fig 7.2)

-Slide module (Fig 7.6)

-Measuring module (Fig 7.5)

Technical data

-Operating pressure min 6 bar, max 8 bar

-No. of I/Os 1 analog input 0...10 V

-No. of I/Os 8 Is/6 Os, binary,for the actuators and sensors

Notes:

-The testing station is particularly suitable for the topics of sensors (binary and analog) and PLC technology (analog processing). You can begin with the symbolic workpiece and later expand step by step up to the assembly station. The station is suitable for holding symbolic workpieces and the body of the workpiece that can be assembled.

-The station can be combined, for example, with the distributing, processing, sorting, buffer or assembly station. -A PLC with analog voltage input is required for PLC control.

6.3. Processing station

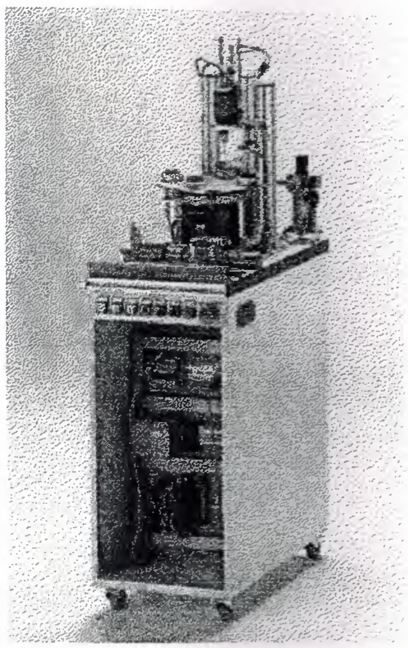


Figure 6.3 (Processing station)

Training aims:The table in the MPS Overview chapter shows you just how versatile the station is.

Task : -Transport of material with rotary indexing table

-Processing a part (drilling)

-Quality control (checking drill holes)

Design : -Rotary indexing table module (Fig 7.7)

-Drilling module (Fig 7.9)

-Drill-hole checking module (Fig 7.10)

Technical data : -Operating pressure min 6 bar, max 8 bar

-No. of I/Os 8 Is/6 Os, binary, for the actuators and sensors

Notes:

-The Processing station symbolizes a processing task – such as drilling of a workpiece – with subsequent testing that the process has been successfully completed. The station is based on an electrically operated rotary indexing table. You can start with the symbolic workpiece and later expand step by step to an Assembly station. The station is suitable for holding symbolic workpieces and the body of the workpiece that can be assembled.

-The station can be combined, for example, with the testing, handling (with insertion device), handling (with robot) and assembly stations.

CHAPTER 7 : Modules

7.1. Stack magazine module

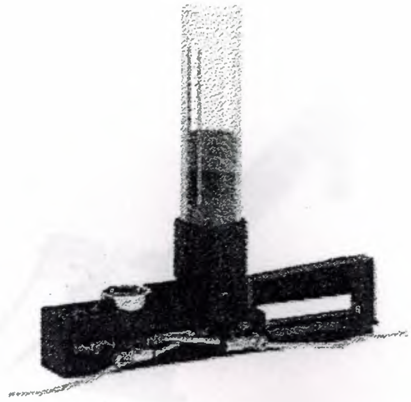


Figure 7.1 (Stack magazine module)

The stack magazine module eparates workpieces from a gravity-feed magazine. The workpieces are then transferred from a defined transfer point to the next station (e.g. by means of the changer module).

The stack magazine is used in various stations. The actual module used (there are 2 models) depends upon the workpiece to be separated (body or cover of the cylinder for assembly).

The position of the thrust cylinder is interrogated by inductive cylinder switches. Both types of the module are supplied complete with pneumatic, quick push-pull connector with one-way flow control valves.

For separating the symbolic workpiece or body of the cylinder for assembly.

-Number of PLC inputs required 3

-Number of outputs required 2

-Cylinder stroke (double-acting cylinder) 80 mm

-Length 290 mm

-Width 60 mm

-Height 280 mm

7.2. Changer Module

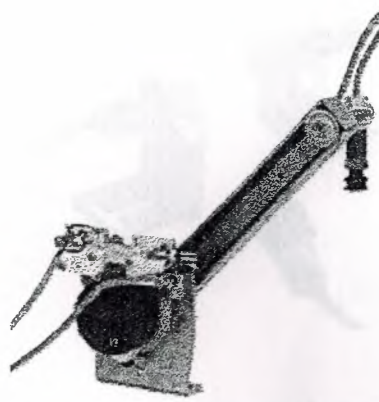


Figure 7.2 (Changer module)

For handling the workpiece.

The workpiece is transferred in a horizontal state to the next station by means of a vacuum suction cup. The module is supplied complete with pneumatic, quick push-pull connector with one-way flow control valves.

-Number of PLC inputs required 2

-If vacuum is also interrogated 3

-Number of outputs required (incl. valve for vacuum) 3

-Angle of rotation of cylinder (rotary cylinder) 180 ° (fully adjustable)

-Length 240 mm

-Width 30 mm

-Height 110 mm

-We recommend the use of a differential pressure switch for interrogating the vacuum

Note:

-If you already work with the changer module and now wish to use the workpiece for assembly, you must exchange the vacuum suction cup.

-Vacuum suction cup for workpiece for assembly

-Adapter

7.3. Recognation module

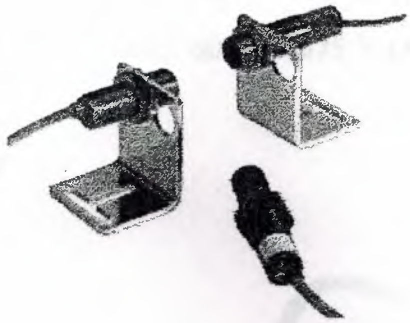


Figure 7.3 (Recognition module)

Description: For recognizing the material and color of the workpiece.The material is determined with the aid of an inductive sensor; an optical sensor distinguishes between the two colors (black or red) and a capacitive sensor signals whether there is a part in the holder.

-Number of PLC inputs required 3

-Number of PLC outputs required 0

-Footprint of sensor bracket 40 x 50 mm

-Supply voltage 24 V DC

-Sensor thread M18

-Sensor output signal (PNP) 0 V/24 V DC

-Recognition module for output signal NPN

7.4. DUO cable for two sensors

Figure 7.4 (Duo cable)

For connenction of two cylinder switches with plugs to an M12 input socket for two input signals.

Plug: Sensor plug

Sensor: M12 external thread

Cable length	0.5m
Conductor cross-section	3x0,25 mm^2
Current carrying capacity	max. 3.8 A
Operating voltage	max. 60 V AC/75 V DC

Voltage supply 24 V DC

7.5. Measuring module

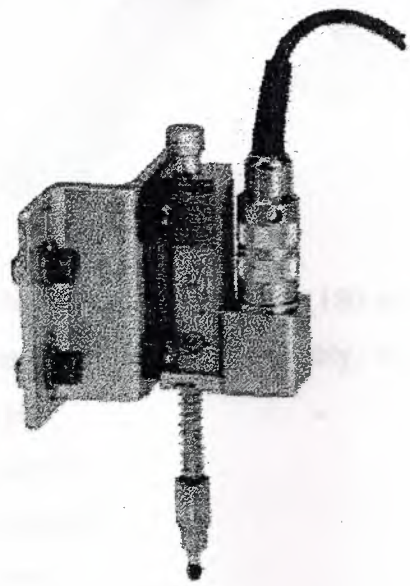


Figure 7.5 (Measuring module)

For measuring the height of the workpieces using an analog sensor with sensor holder.The analog measured value is further processed by a programmable logic controller (PLC) with analog inputs or by an A/D converter. For this, we recommend the R/U converter .

-Number of PLC inputs required 1 (analog)

-Number of PLC outputs required 0

-Supply voltage 24 V DC

-Sensor output signal in conjunction with the position display module 0...10 V DC

7.6. Slide module

Figure 7.6 Slide module (180 mm)

For transporting workpieces (symbolic or for assembly) to the next station.The angle of inclination is infinitely variable.

-*Number of PLC inputs required 0*

-Number of PLC outputs required 0

-Width of slide (inside) 40 mm

7.7 Rotary indexing module

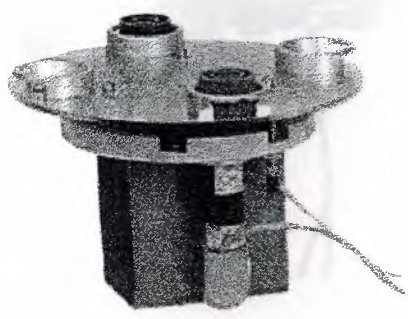


Figure 7.7 (Rotary indexing module)

For holding the workpieces for processing.

The workpiece is machined in four cycles and made available for transfer to the downstream station. The module is supplied complete with inductive proximity switch and optical sensor (for signalling 'workpiece present').

-Number of PLC inputs required 2
-Number of outputs required 1
-Workpiece positions 4
-Diameter 260 mm
-Height 180 mm
-Recommended motor
-*Technical data of DC motor* 24 V
-Nominal voltage 3200 rpm
-Nominal speed 0.3 A
-Nominal current 3.31 W
-Nominal power output 3.0 Ncm
-Starting torque 3.0 Ncm

7.8 Lifting module

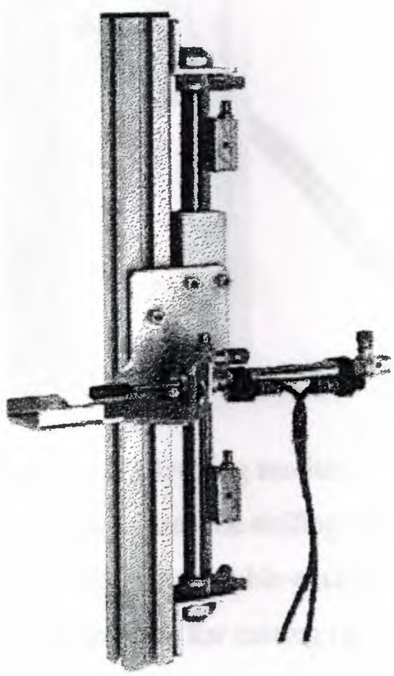


Figure 7.8 (lifting module)

For lifting a workpiece with the aid of a rodless cylinder. In this position, the workpiece can be checked by the measuring module. A further cylinder is then used to eject the workpiece.

-Number of PLC inputs required (incl. analog inputs) 7
-Number of PLC outputs required 4

-Height 290 mm

-Width 220 mm

-Depth 130 mm

7.9. Drilling module

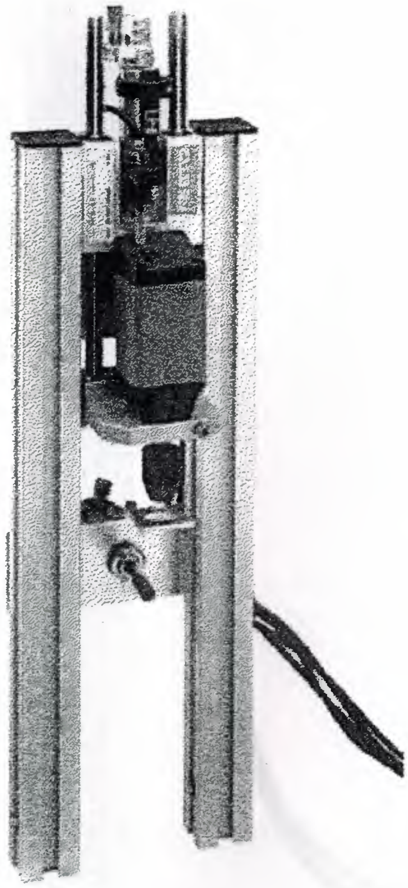


Figure 7.9 (Drilling module)

Description: The drilling module simulates the drilling of a hole in the workpiece. A drill is moved downwards (vertically) by a double-acting cylinder guided by a guide unit. If you wish to use the drilling module for cutting operations, we recommend the additional use of a shock absorber.

-Number of PLC inputs required 2

-Number of PLC outputs required 3

-Supply voltage of motor 24 V DC

-Height 510 mm

-Width 130 mm

-Depth 90 mm

7.10. Drill hole checking

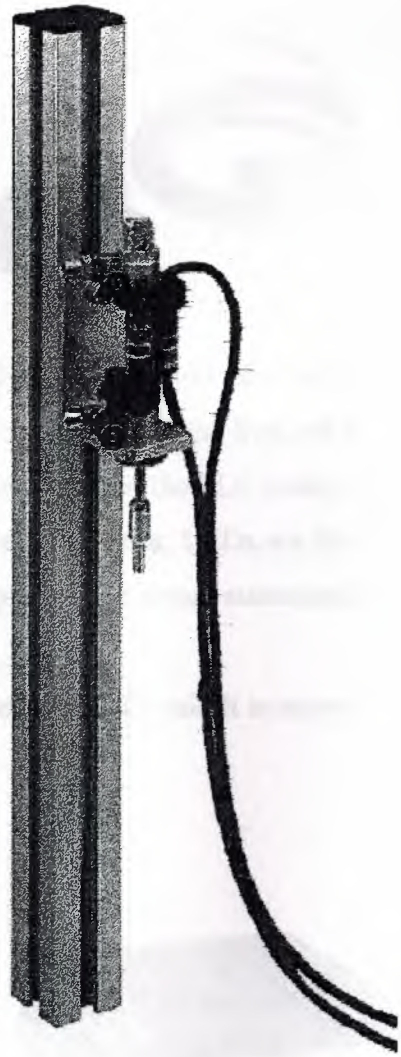


Figure 7.10 (Drill hole checking)

For checking that the 'drilling' process has been performed correctly. The module checks whether there is a drill hole in the workpiece. A double-acting cylinder with test probe is lowered into the work-piece. If the hole is present, the forward end position is reached and the cylinder switch is actuated.

-Number of PLC inputs required 2

-Number of PLC outputs required 2

-Height 400 mm

-Width 32 mm

-Depth 80 mm

7.11. I/O terminal (SysLink)

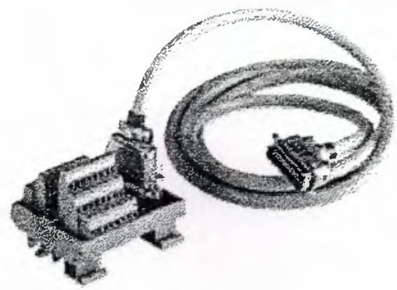


Figure 7.11 (I/O terminal)

The I/O terminal is the central unit of the SysLink concept of MPS. This terminal is used both on the station and on the PLC board, for wiring 8 inputs and 8 outputs that are combined on one connector. LEDs are fitted to the input and output terminals to provide simple display of the circuit states and to enable systematic troubleshooting.

The I/O terminal can be mounted on a DIN rail. It is supplied without a cable.

-Terminals for inputs 8

-Terminals for outputs

7.12. Simulation box

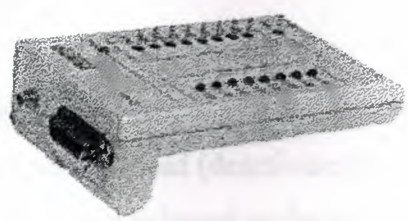


Figure 7.12 (Simulation box)

The simulation box is used to simulate the input/output signals of an MPS station.

Two ways of employment are possible

-Simulation of input signals to test a PLC program.

-Setting of outputs (by using a separate 24 V power supply) to operate an MPS station.

-Connector type IEEE 488 24-pin

CHAPTER 8
Ladder Instruction Set

8.1. Normally Open Contact

Symbol:

Operands:

n (bit): I, Q, M, SM, S, T, C, V

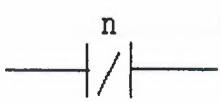
Description of operation:

The Normally Open Contact is closed when the scanned bit value stored at address n is 1. Power flows through a normally open contact when closed (activated).

Used in series, a normally open contact is linked to the next LAD element by AND logic. Used in parallel, it is linked by OR logic.

8.2. Normally Closed Contact

Symbol:



Operands:
n (bit): I, Q, M, SM, S, T, C, V

Description of operation:

The Normally Closed Contact is closed when the bit value stored at address n is 0 . Power flows through the contact when closed (deactivated).

Used in series, a normally closed contact is linked to the next LAD element by AND logic. Used in parallel, it is linked by OR logic.

8.3. Output

Symbol:

Operands:
n (bit): I, Q, M, SM, S, T, C, V

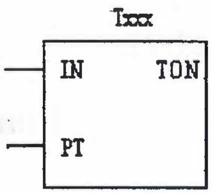
Description of operation:

An Output coil is turned on and the Bit stored at address n is set to 1 when power flows to the coil.

A negated output can be created by placing a NOT (Invert Power Flow) contact before an output coil.

8.4. Timer – On Delay

Symbol:



Operands:

Txx (word): CPU 212: 32-63CPU 214: 32-63, 96-127CPUs 215, 216: 32-63, 96-255

PT (word): VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC

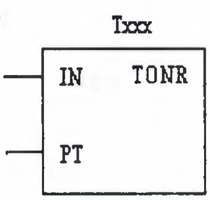
Description of operation:

The On–Delay Timer (TON) box times up to the maximum value when the enabling Input (IN) comes on. When the current value (Txxx) is >= the Preset Time (PT), the timer bit turns on. It resets when the enabling input goes off. Timing stops upon reaching the maximum value.

	CPU 212/214	CPU 214/215/216	CPU 215/216
1 ms	T32	T96	
10 ms	T33-T36	T97-T100	
100 ms	T37-T63	T101-T127	T128-T255

8.5. Timer – Retentive On Delay

Symbol:



Operands:

Txxx (word): CPU 212: 0-31CPU 214/215/216: 0-31, 64-95

PT (word): VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC

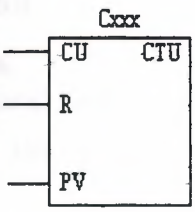
Description of operation:

The Retentive On Delay Timer (TONR) box times up to the maximum value when the enabling Input (IN) comes on. When the current value (Txxx) is >= the Preset Time (PT), the timer bit turns on. Timing stops when the enabling input goes off, or upon reaching the maximum value.

	CPU 212/214	CPU 214/215/216
1 ms	T0	T64
10 ms	T1-T4	T65-T68
100 ms	T5-T31	T69-T95

8.6. Count Up

Symbol:



Operands:

Cxxx (word): CPU 212: 0-63CPU 214: 0-127CPU 215/216: 0-255

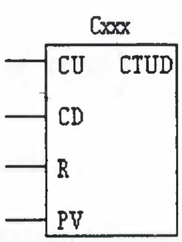
PV (word): VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC

Description of operation:

The Count Up (CTU) box counts up to the maximum value on the rising edges of the Count Up (CU) input. When the current value (Cxxx) is >= to the Preset Value (PV), the counter bit (Cxxx) turns on. It resets when the Reset (R) input turns on. It stops counting upon reaching the maximum value (32,767).

8.7. Count Up / Down

Symbol:



Operands:

Cx: C0 - C3

PV: 0

Description of operation:

The Count Up/Down (CTUD) box counts up on rising edges of the Count Up (CU) input. It counts down on the rising edges of the Count Down (CD) input. It resets *when the Reset (R) input turns on.*

When the current count reaches the maximum value (32,767), the next rising edge at the count-up input will cause the current count to wrap around to the minimum value (-32,768).

When the current count reaches the minimum value (-32,768), the next rising edge at the count-down input will cause the current count to wrap around to the maximum value (32,767).

8.8. End

Symbols:

Conditional End

—(END)

Unconditional End

|—(END)

Operands:

(none)

Description of operation:

The Conditional End coil terminates the main user program based on the condition of the preceding logic.

The Unconditional End coil must be used to terminate the user program.

8.9. Stop

Symbol:

—(STOP)

Operands:

(none)

Description of operation:

The Stop coil terminates execution of the user program by causing a transition to the stop mode.

8.10. Return

Symbols:

Conditional Return from Subroutine

—(RET)

Unconditional Return from Subroutine

|—(RET)

Operands:
(none)
Description of operation:

The Conditional Return from Subroutine coil terminates a subroutine based on the condition of the preceding logic.

The Unconditional Return from Subroutine coil must be used to terminate each subroutine.

8.11. Read Real Time Clock

Symbol:

```
   +----------+
   | READ_RTC |
 --| EN       |
   |          |
 --| T        |
   +----------+
```

Operands:

T (byte): VB, IB, QB, MB, SMB, SB, *VD, *AC

Description of operation:

The Read Real Time Clock (READ_RTC) box reads the current time and date from the clock and loads it in an 8-byte buffer (T).

Example Memory Data Starting at VB400:

Extended Power Outages:

The time of day clock initializes the following date and time after extended power outages or memory has been lost:

Date: 01-Jan-90 Time: 00:00:00 Day of Week Sunday

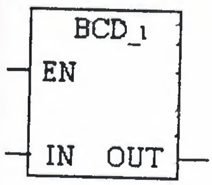
The clock is disabled with this setting until reset.

Note:

Do not use the READ_RTC / SET_RTC instructions in both the main program and in an interrupt routine. If the clock instruction in the main program is executing when the interrupt occurs, then the clock instruction in the interrupt routine is not executed. SM4.5 is then set, indicating that two simultaneous accesses to the clock were attempted.

8.12. BCD to Integer

Symbol:



Operands:

IN (word):

VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC

OUT (word):

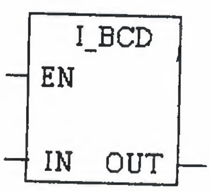
VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC

Description of operation:

The Convert BCD to Integer (BCD_I) box converts the BCD value (IN) to an integer value (OUT). If the input value contains an invalid BCD digit, SM1.6 is set.

8.13. Integer to BCD

Symbol:



Operands:

IN (word):

VW, T, C, IW, QW, MW, SMW, SW, AC, AIW, Constant, *VD, *AC

OUT (word):

VW, T, C, IW, QW, MW, SMW, SW, AC, *VD, *AC

Description of operation:

The Convert Integer to BCD (I_BCD) box converts the integer value (IN) to the BCD value (OUT). If the conversion produces a BCD number greater than 9999, SM1.6 is set.

CHAPTER 9: CONTROLLING THE PROGRAM

9.1. Compiling from the LAD View

Select Compile from the CPU menu, and the Micro/WIN compiler translates your LAD program into an organization (program) block that can be downloaded into an S7-200 PLC.

The Micro/WIN LAD compiler fails if any of these errors exist in your program.

- The network is missing a contact element
- The network is missing an output element
- Short circuit
- Open circuit
- Reverse power flow
- Illegal placement of positive transition, negative transition, or NOT instruction
- Additional outputs or boxes are not allowed with a CTU/CTUD box
- Other elements are not allowed with an unconditional output
- Invalid network
- Interconnections between CTU/CTUD inputs are not allowed
- Network too large to be compiled

The Micro/WIN LAD compiler does not check for an address out-of-range error, nor does it check overall program structure. For example, the LAD compiler does not check for a missing MEND coil or CALLs to subroutine networks that do not exist. These types of errors are identified and reported by your CPU when you attempt a download.

9.2. Clearing a Program

You have the option of clearing a program from the CPU. To clear the CPU from Micro/WIN, you must have the CPU switch in TERM position.

1.Open the Project to be cleared.

2.Select Clear from the CPU menu.

3.Click on the option button to select Clear All.

4.Click on the OK button.

When the CPU is cleared, this means that the following conditions are met.

CPU transitions to Stop mode.

- OB1 (code block) is deleted.
- DB1 (data block) is deleted.
- All configuration parameters, except the CPU address, are cleared.
- All System Memory (SM) bits are set to default state.
- All Memory (M) bits are cleared.
- Analog outputs are frozen.

- ~~All system data~~ memory is set to default state.
- All forced points are ~~cleared and unforced~~.
- All timer/counter current data is deleted.

When a memory ~~cartridge is installed, the memory cartridge data is not changed~~ on a Clear operation. To clear the memory cartridge, copy the cleared CPU to the memory cartridge. Otherwise, a power cycle restores the CPU to its original state; because the contents of the memory cartridge are copied to the CPU on power-up.

CPU Memory Types and Ranges

Accessed by:	CPU 212	Ranges	CPU 214	Ranges
Bit (Byte.bit)	V	0.0 - 1023.7	V	0.0 - 4095.7
	I	0.0 - 7.7	I	0.0 - 7.7
	Q	0.0 - 7.7	Q	0.0 - 7.7
	M	0.0 - 15.7	M	0.0 - 31.7
	SM	0.0 - 45.7	SM	0.0 -5.7
	T	0 – 63	T	0 - 127
	C	0 – 63	C	0 - 127
	S	0.0 - 7.7	S	0.0 - 15,7
Byte	VB	0 –1023	VB	0 - 4095
	IB	0 – 7	IB	0 - 7
	QB	0 – 7	QB	0 - 7
	MB	0 – 15	MB	0 - 31
	SMB	0 – 45	SMB	0 - 85
	AC	0 – 3	AC	0 – 3
	KB (Constant)		KB (Constant)	
	SB	0 – 7	SB	0 - 15
Word	VW	0 – 1022	VW	0 - 4094
	T	0 – 63	T	0 - 127
	C	0 – 63	C	0 - 127
	IW	0 – 6	IW	0 - 6
	QW	0 – 6	QW	0 - 6
	MW	0 – 14	MW	0 – 30
	SMW	0 – 44	SMW	0 - 84
	AC	0 – 3	AC	0 - 3
	AIW	0 – 30	AIW	0 - 30
	AQW	0 - 30	AQW	0 – 30
	KW (Constant)		KW (Constant)	
	SW	0 – 6	SW	0 - 14
Double Word	VD	0 – 1020	VD	0 - 4092
	ID	0 – 4	ID	0 - 4
	QD	0 – 4	QD	0 - 4
	MD	0 – 12	MD	0 - 28
	SMD	0 – 42	SMD	0 - 82
	AC	0 – 3	AC	0 - 3
	HC	0	HC	0 - 2
	KD (Constant)		KD (Constant)	
	SD	0 – 4	SD	0 - 4

Accessed by:	CPU 215	Ranges	CPU 216	Ranges
(Byte.bit)	V	0.0 - 5119.7	V	0.0-19.7
	I	0.0 - 7.7	I	0.0 - 7.7
	Q	0.0 - 7.7	Q	0.0 - 7.7
	M	0.0 - 31.7	M	0.0 - 31.7
	SM	0.0 - 199.7	SM	0.0 - 199.7
	T	0 – 255	T	0 - 255
	C	0 – 255	C	0 - 255
	S	0.0 - 31.7	S	0.0 - 31.7
	VB	0 – 5119	VB	0 - 5119
	IB	0 – 7	IB	0 - 7
	QB	0 – 7	QB	0 - 7
	MB	0 – 31	MB	0 - 31
	SMB	0 – 199	SMB	0 - 199
	AC	0 – 3	AC	0 - 3
	KB (Constant)		KB (Constant)	
	SB	0 – 31	SB	0 - 31
	VW	0 – 5118	VW	0 - 5118
	T	0 – 255	T	0 - 255
	C	0 – 255	C	0 - 255
	IW	0 – 6	IW	0 - 6
	QW	0 – 6	QW	0 – 6
	MW	0 – 30	MW	0 - 30
	SMW	0 – 198	SMW	0 - 198
	AC	0 – 3	AC	0 – 3
	AIW	0 – 30	AIW	0 - 30
	AQW	0 – 30	AQW	0 - 30
	KW (Constant)		KW (Constant)	
	SW	0 – 30	SW	0 - 30
	VD	0 – 5116	VD	0 - 5116
	ID	0 – 4	ID	0 - 4
	QD	0 – 4	QD	0 - 4
	MD	0 – 28	MD	0 - 28
	SMD	0 – 196	SMD	0 - 196
	AC	0 – 3	AC	0 - 3
	HC	0 – 2	HC	0 - 2
	KD (Constant)		*KD (Constant)*	
	SD	0 – 28	SD	0 - 28

CHAPTER 10 : SIMATIC S7-200 / CPU-212 PROGRAMMING
10.1. Ladder Diagram

PROGRAM TITLE COMMENTS

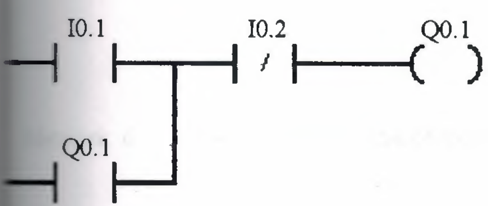
Press F1 for help and example program

Network 1 one workpice is sending the out

When we press the I0.0 one workpiece is sending the out with Q0.0

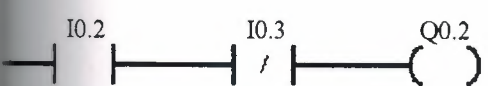
Network 2 Workpiece is sending the out

The workpiece is come to I0.1 switches and turned off and then the Q0.1 is start to work until the I0.2 contacts is opened



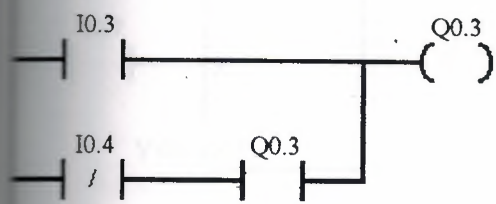
Network 3 The changer module is coming to take the workpice

The changer module is moving to left when the I0.2 open contact is closed and after the open the I0.3 close contact will be opened and then Q0.2 is stopped.



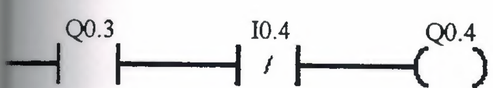
Network 4 working for to hold the workpice

I0.3 open contacts is turn to close after that Q0.3 hold motor will be start to work until the I0.4 contact is opened

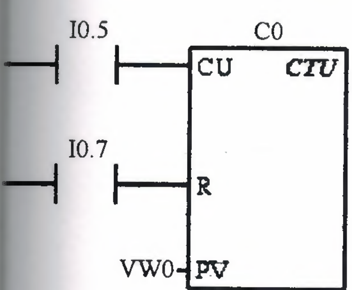


Network 5 changer module is working to give the workpiece the other station

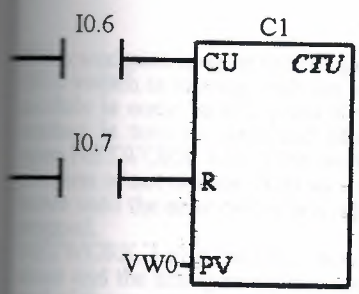
When the hold motor (Q0.3) working and the I0.4 contact is close and the Q0.4 will be active mode (Q0.4 is changer module is moving to right side motor)



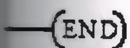
Network 6 working for the count of red workpice with the colour sensor



Network 7 Working for the count of workpice with the colour sensor



Network 8

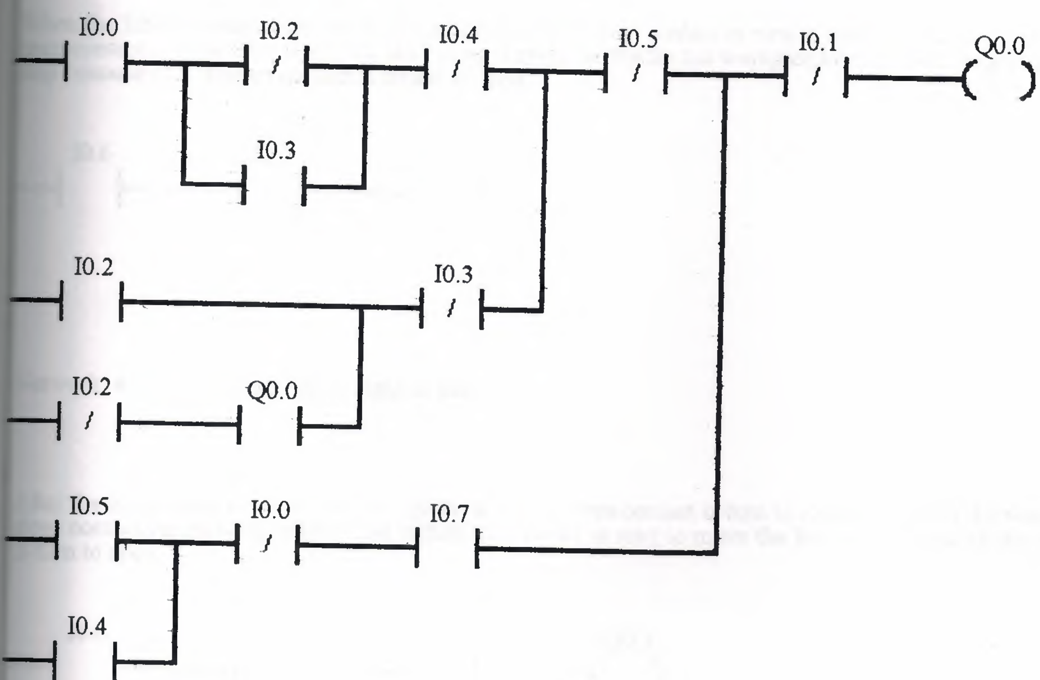


PROGRAM TITLE COMMENTS

Press F1 for help and example program

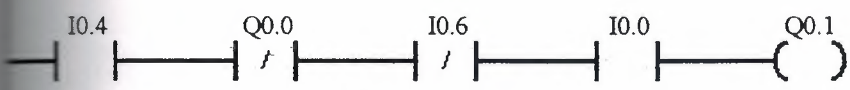
Network 1 NETWORK TITLE (Moving to lifting module up/down)

If the workpice is come to lifting station I0.0 open contacts is turn to close (I0.0 switch is moving with the work piece) and then the lifting module is moving to down stairs to upstairs when the lifting module is come to I0.2 point I0.2 close contact is turn to open and if the workpiece is have got the correct high I0.3 open contact is turn to close and lifting module is start to move upstairs to down stairs until the I0.4 close contact will be open.NETWORK 1.b) If the workpiece is havent got the necessary high I0.2 open contacts is turn to close but I0.3 contact's position is not chance. And so we will be say the workpiece is havent got the correct high and then lifting module is start to move until the error pieces box when the lift is come to error pieces box I0.5 close contact is turn to open and the Q0.0 motor is stopped.
NETWORK 1.c) if the lift is stop to I0.4 or I0.5 point and the workpiece is give to other station I0.0 close contact is return to close and the lifting module is come to home I0.7 open contact turn to close and then lifting module is start to move until the end of the way contacts I0.1 close contact is turn to opened,and the system is wait to other workpice.



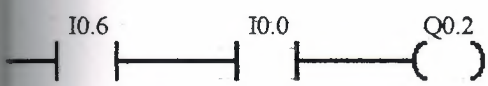
Network 2 The lifting is moving to left to right

If the workpice is have got to the necessary high the lifting down stairs motor (0.0) and I0.4 open contact is turn to close and then lefthen side to righthen side motor(Q0.1) will be start to move to the righten side until the end of the way switches(I0.6) close switches turn to open we put the I0.0 ontacts to the this circuit becouse of when the lifthing is moving to lefthen side I0.6 close contact is return to close and so the program is can not be work correctly but if we put the I0.0 open contact after the workpice is goes to other station returning to open



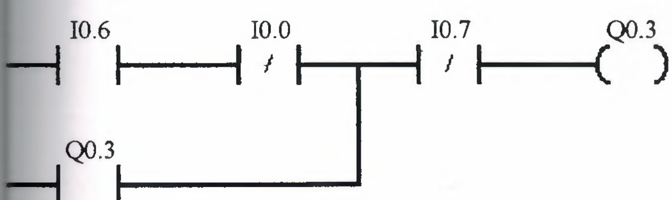
Network 3 Workpice is sending to other station

When the lifting module is come to end of the way I0.6 open contact is turn to close and the workpice is on the lift and so I0.0 open contact is close after that Q0.2 start to send away for the for the workpice,after the workpice is goes on the lift Q0.2 will be stop becouse of I0.0 open contact is return to open



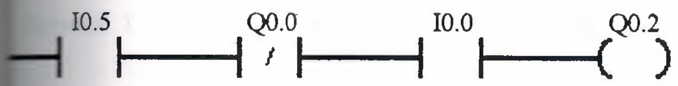
Network 4 Lift is moving to right to left

After the lift is come to completely to righten side I0.6 open contact is turn to close and and if the workpice is not on the lift I0.0 close contact return to close after that lefthen side motor is start to move the lift until the end of the way switches close contact is turn to open.

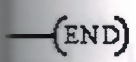


Network 5 Workpiece is sending the error box

If the workpice is havent got the correct high the lift is stopped from the
I0.5 open contact is turn to close and lifting module is not working and the workpice is on the lift and so I0.0 open contact is closed,becouse of the Q0.2 is tsart to send away until the workpiece control switches I0.0 will be open.



Network 6

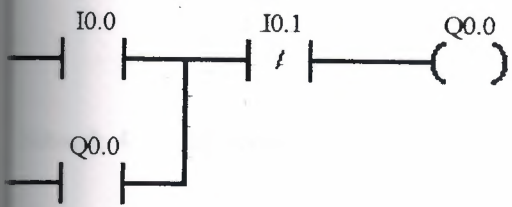


PROGRAM TITLE COMMENTS

Press F1 for help and example program

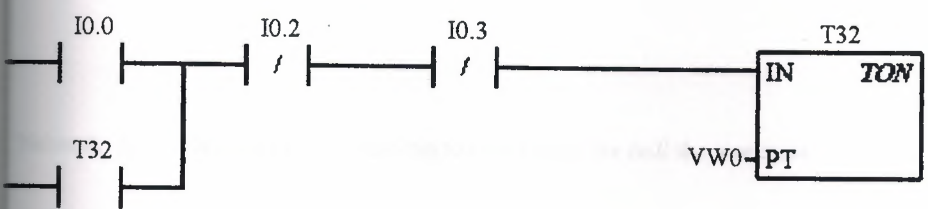
Network 1 Moving to rotary motor from the drilling

When a workpiece is come to dirilling station I0.0 open contact is turn to close and then the rotary indexing module is start to move the workpice until the drilling modules below when the rotary changer module is come to this point I0.1 close contact is turned ON,and then Q0.0 is stopped.



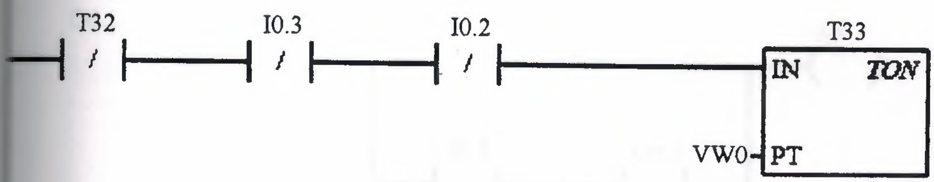
Network 2 timer is controlling the dirill motor to start

when the I0.0 open contact is closed the timer is active mode until the I0.2 or I0.3 close contact open



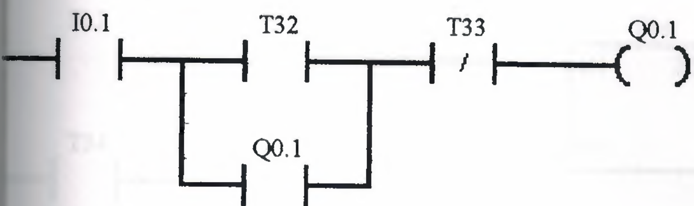
Network 3 Controlling dirill motor for the stop

when the T32 is close and the I0.2 and I0.3 close and then T33 will be active



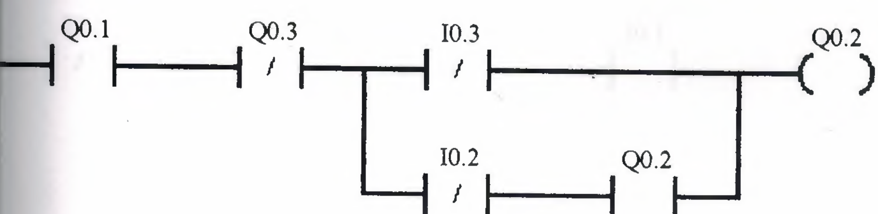
Network 4 drill motor

when the rotary indexing module is come to below to the drill motor T32 is active and T33 is not active and then drill motor is star to work until the T33 will be active.



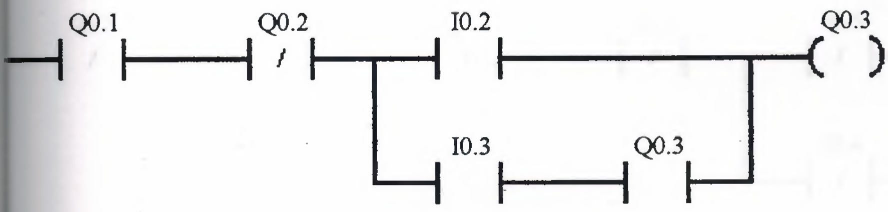
Network 5 Drilling motor is moving to down stairs for drill the workpice

When the drill motor is active Q0.1 open contact is turn to close and then the drill motor is goes to down stairs for drill the workpiece I0.3 is upstairs switch and I0.2 is downstairs switchwh the drill is moveing to down stairs I0.3 close contact is turn to open and then the drill is come to completely to the down stairs I0.2 close contact is turn to open and so Q0.2 downstairs motor is stopped.



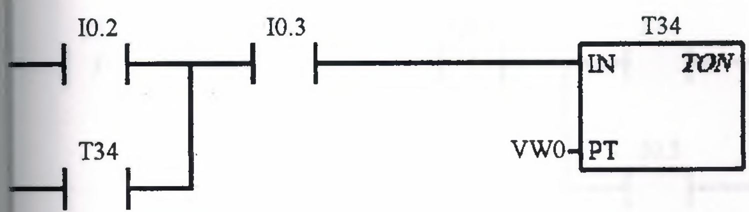
Network 6 Drill is moving to up stairs

When the drill is come and close to I0.2 open contacts and then upstairs motor Q0.3 is start to work until the drill is turn to upstairs and opened the I0.3 open contact(I0.3 open contact is when the drill is from the downstairs closed).



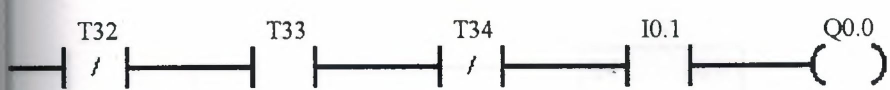
Network 7 Controlling the timer

When the I0.2 and I0.3 open contacts are turn to close the timer is turn to active mode until the drill is come to completely to the upstairs and then I0.3 open contact is return to open.



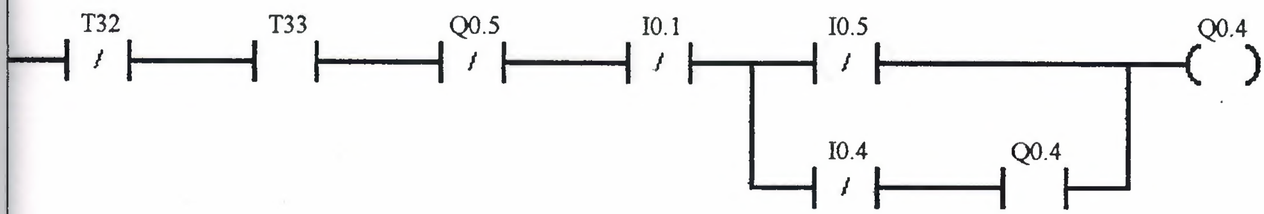
Network 8 Rotary module is work for the drill check module

T32 close T39 open contact is close because from the network3,T32 close and I0.2 and I0.3 is close and so the timer is working.



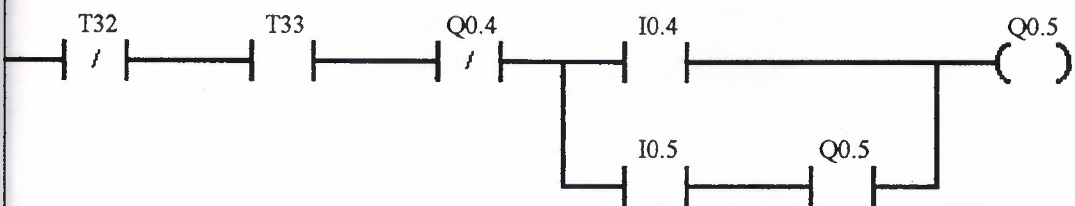
Network 9 drill hole checking module

When the drill is moving to down stairs I0.5 close contact is open and then I0.4 close contact is turn to open Q0.4 output will be stop.



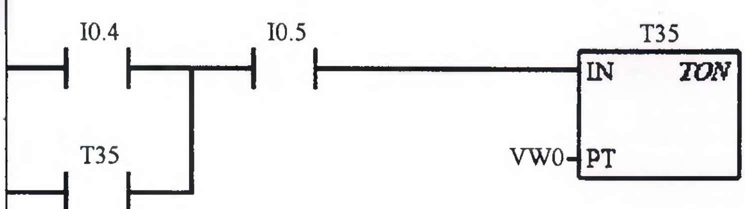
Network 10 drill hole checking module

when the drill check machine come to down stairs I0.4 open contacts is turn to close and after the drill check module is moving to up stairs until the I0.5 open contact return to open.



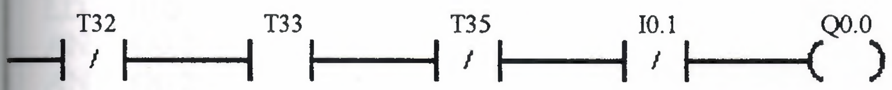
Network 11 controlling rotary modules energy

When the I0.4 and I0.5 open contact is turn to close the timer is start to work firstly I0.4 open contact is open but the timer is taken the energy from the parallel with I0.4 to T35, after the I0.5 contact is turned to open the timer will be stop mode.

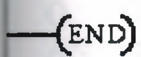


Network 12 rotary module is moving for the give the workpice to the other station

T32 is closed T33 open contact is closed T35 is closed after that I0.1 closed the rotary module is start to move to for the workpice give to the other station



Network 13



11.2. Distributing Station STL Program

NETWORK 1 //one workpice is sending the out
//When we press the I0.0 one workpiece is sending the out with Q0.0

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

NETWORK 2 //Workpiece is sending the out
//one workpiece is come to I0.1 switches and turned off and then the Q0.1 is start to work until the I0.2 contacts is opened

```
LD    I0.1
O     Q0.1
AN    I0.2
=     Q0.1
```

NETWORK 3 //The changer module is coming to take the workpice
//The changer module is moving to left when the I0.2 open contact is closed and after the open the I0.3 close contact will be opened and then Q0.2 is stopped.

```
LD    I0.2
AN    I0.3
=     Q0.2
```

NETWORK 4 //working for to hold the workpice
//I0.3 open contacts is turn to close after that Q0.3 hold motor will be start to work until the I0.4 contact is opened

```
LD    I0.3
LDN   I0.4
A     Q0.3
OLD
=     Q0.3
```

NETWORK 5 //changer module is working to give the workpiece the other station
//When the hold motor (Q0.3) working and the I0.4 contact is close and the Q0.4 will be active mode (Q0.4 is changer module is moving to right side motor)

```
LD    Q0.3
AN    I0.4
=     Q0.4
```

NETWORK 6 //working for the count of red workpice with the colour sensor

```
LD    I0.5
LD    I0.7
CTU   C0, VW0
```

NETWORK 7 //Working for the count of workpice with the colour sensor

```
LD    I0.6
```

```
LD     I0.7
CTU    C1, VW0

NETWORK 8
MEND
```

11.3. Lifting Station STL Program

//PROGRAM TITLE COMMENTS

NETWORK 1 //NETWORK TITLE (Moving to lifting module up/down)

//If the workpice is come to lifting station I0.0 open contacts is turn to close //(I0.0 switch is moving with the work piece) and then the lifting module is moving to down stairs to upstairs when the lifting module is come to I0.2 point I0.2 close contact is turn to open and if the workpiece is have got the correct high I0.3 open contact is turn to close and lifting module is start to move upstairs to down stairs until the I0.4 close contact will be open.

NETWORK 1.b) If the workpiece is havent got the necessary high I0.2 open contacts is turn to close but I0.3 contact's position is not chance. And so we will be say the workpiece is havent got the correct high and then lifting module is start to move until the error pieces box when the lift is come to error pieces box I0.5 close contact is turn to open and the Q0.0 motor is stopped.

//NETWORK 1.c) if the lift is stop to I0.4 or I0.5 point and the workpiece is give to other station I0.0 close contact is return to close and the lifting module is come to home I0.7 open contact turn to close and then lifting module is start to move until the end of the way contacts I0.1 close contact is turn to opened,and the system is wait to other workpice.

```
LD     I0.0
LDN    I0.2
O      I0.3
ALD
AN     I0.4
LD     I0.2
LDN    I0.2
A      Q0.0
OLD
AN     I0.3
OLD
AN     I0.5
LD     I0.5
O      I0.4
AN     I0.0
A      I0.7
OLD
AN     I0.1
=      Q0.0
```

NETWORK 2 //The lifting is moving to left to right

//if the workpice is have got to the necessary high the lifting down stairs motor (0.0) and I0.4 open contact is turn to close and then lefthen side to righthen side motor(Q0.1) will be start to move to the righten side until the end of the way switches(I0.6) close switches turn to open we put the I0.0 ontacts to the this circuit becouse of when the lifthing is moving to lefthen side I0.6 close contact is return to close and so the program is can not be work correctly but if we put the I0.0 open contact after the workpice is goes to other station returning to open

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

NETWORK 3 //Workpice is sending to other station
//When the lifting module is come to end of the way I0.6 open contact is turn to close and the workpice is on the lift and so I0.0 open contact is close after that Q0.2 start to send away for the for the workpice,after the workpice is goes on the lift Q0.2 will be stop becouse of I0.0 open contact is return to open

```
LD   I0.6
A    I0.0
=    Q0.2
```

NETWORK 4 //Lift is moving to right to left
//After the lift is come to completely to righten side I0.6 open contact is turn to close and and if the workpice is not on the lift I0.0 close contact return to close after that lefthen side motor is start to move the lift until the end of the way switches close contact is turn to open.

```
LD   I0.6
AN   I0.0
O    Q0.3
AN   I0.7
=    Q0.3
```

NETWORK 5 //Workpiece is sending the error box
//if the workpice is havent got the correct high the lift is stopped from the
//I0.5 open contact is turn to close and lifting module is not working and the workpice is on the lift and so I0.0 open contact is closed,becouse of the Q0.2 is tsart to send away until the workpiece control switches I0.0 will be open.

```
LD   I0.5
AN   Q0.0
A    I0.0
=    Q0.2
NETWORK 6
MEND
```

11.4.Drilling Station STL Program

NETWORK 1 //Moving to rotary motor from the drilling
//When a workpiece is come to dirilling station I0.0 open contact is turn to close and then the rotary indexing module is start to move the workpice until the drilling modules below when the rotary changer module is come to this point I0.1 close contact is turned ON,and then Q0.0 is stopped.
//

```
LD   I0.0
O    Q0.0
```

```
AN    I0.1
=     Q0.0
```

NETWORK 2 //timer is controlling the dirill motor to start
//when the I0.0 open contact is closed the timer is active mode until the I0.2 or I0.3 close contact open

```
LD    I0.0
O     T32
AN    I0.2
AN    I0.3
TON   T32, VW0
```

NETWORK 3 //Controlling dirill motor for the stop
//when the T32 is close and the I0.2 and I0.3 close and then T33 will be active

```
LDN   T32
AN    I0.3
AN    I0.2
TON   T33, VW0
```

NETWORK 4 //drill motor
//when the rotary indexing module is come to below to the drill motor
//T32 is active and T33 is not active and then drill motor is star to work
//until the T33 will be active.

```
LD    I0.1
LD    T32
O     Q0.1
ALD
AN    T33
=     Q0.1
```

NETWORK 5 //Drilling motor is moving to down stairs for drill the workpice
//When the drill motor is active Q0.1 open contact is turn to close and then the drill motor is goes to down stairs for drill the workpiece I0.3 is upstairs switch and I0.2 is downstairs switchwh the drill is moveing to down stairs I0.3 close contact is turn to open and then the drill is come to completely to the down stairs I0.2 close contact is turn to open and so Q0.2 downstairs motor is stopped.

```
LD    Q0.1
AN    Q0.3
LDN   I0.3
LDN   I0.2
A     Q0.2
OLD
ALD
=     Q0.2
```

NETWORK 6 //Drill is moving to up stairs
/When the drill is come and close to I0.2 open contacts and then upstairs motor Q0.3 is start to work until the drill is turn to upstairs and opened the I0.3 open contact(I0.3 open contact is when the drill is from the downstairs closed).

```
LD   Q0.1
AN   Q0.2
LD   I0.2
LD   I0.3
A    Q0.3
OLD
ALD
=    Q0.3
```

NETWORK 7 //Controlling the timer
//When the I0.2 and I0.3 open contacts are turn to close the timer is
//turn to active mode until the drill is come to completely to the
//upstairs and then I0.3 open contact is return to open.

```
LD   I0.2
O    T34
A    I0.3
TON  T34, VW0
```

NETWORK 8 //Rotary module is work for the drill check module
//T32 close T39 open contact is close because from the network3,T32 close and I0.2 and I0.3 is close and so the timer is working.

```
LDN  T32
A    T33
AN   T34
A    I0.1
=    Q0.0
```

NETWORK 9 //drill hole checking module
//When the drill is moving to down stairs I0.5 close contact is open
//and then I0.4 close contact is turn to open Q0.4 output will be stop.

```
LDN  T32
A    T33
AN   Q0.5
AN   I0.1
LDN  I0.5
LDN  I0.4
A    Q0.4
OLD
ALD
=    Q0.4
```

NETWORK 10 //drill hole checking module
//when the drill check machine come to down stairs I0.4 open contacts
//is turn to close and after the drill check module is moving to up
//stairs until the I0.5 open contact return to open.

```
LDN  T32
A    T33
AN   Q0.4
LD   I0.4
```

```
LD    I0.5
A     Q0.5
OLD
ALD
=     Q0.5
```

NETWORK 11 //controlling rotary modules energy
//When the I0.4 and I0.5 open contact is turn to close the timer is start to work firstly I0.4 open contact is open but the timer is taken the energy from the parallel with I0.4 to T35, after the I0.5 contact is turned to open the timer will be stop mode.

```
LD    I0.4
O     T35
A     I0.5
TON   T35, VW0
```

NETWORK 12 //rotary module is moving for the give the workpice to the other station
//T32 is closed T33 open contact is closed T35 is closed after that I0.1 closed the rotary module is start to move to for the workpice give to the other station

```
LDN   T32
A     T33
AN    T35
AN    I0.1
=     Q0.0
```

NETWORK 13
MEND

CONCLUSION

When developing this project we see that PLC makes our life easier in everyday applications.

With the information observed from our lecture and our researchers for this topic PLC , is a convenient tool with a wide rage of useful ways to be used. Such examples can be mentioned several machines can be used at the same time , easy adjustments from the PLC program can be meet within a few minutes by the keyboard , installed PLC programs can be controlled or checked before within the office and laboratory , even the PLC programs for firm can be meet at home. It is very protective and safe for the workers. Comminication programs of PLC's within each other or during operation is possible. The developed languages have constructed the productivity , security , establishment security fast productivity , quality and we can see that PLC is a very cheap device that can be fundamentally used.

REFERENCES

Mustafa Yağımlı & Feyzi Akar, Programmable Logic Control, BETA,İstantanbul, 1999

Özgür C.Özerdem, Programmable Logic Control,Lecture Note, Near East University, Lefkosa, 2001

World Wide Web : **www.siemens.com**

World Wide Web : www.festo.com\didactic