NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

TRAFFIC LIGHT CONTROL WITH PLC

Graduation Project
EE 400

Student: Mehmet Kinsiz (950288)

Supervisor: Mr. Özgür C. Özerdem

T.R.N.C. - 2000

CONTENTS

ACKNOWLEDGEMENTS

I must first acknowledge some depts from my more distance past. To my parents for their support to an upwardly mobile son. To my friend Recep Kürkçü, Gökhan Ömer Evran, Bora Enis Tuğlu,Bahattin Şanlı and Ali İhsan Taşdemir for his helps.

I want to thanks separately to my teachers that during my education they support me, the Dean of Engineering Faculty Prof. Dr. Halil ISMAILOV, Head of Electrical & Electronic Department Prof. Dr. Fakhreddin MAMEDOV and also Mr. Özgür ÖZERDEM, Mr. Kaan UYAR, Assis. Prof. Dr. Kadri BÜRÜNCÜK and Assis. Prof. Dr. Kamil DİMİLİLER

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

ABSTRACT

PLC (Programmable Logic Controllers) is a thing 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 transforms 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. There are command which are mathematics process that are comparator processes. These are the main function and feature of software part of PLC.

INTRODUCTION

In the late 1960's PLCs were first introduced. The primary reason for designing such a device was eliminating the large cost involved in replacing the complicated relay based machine control systems. Bedford Associates (Bedford, MA) proposed something called a Modular Digital Controller (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. That's a lot to ask! The answers were to use a programming technique most people were already familiar with and replace mechanical parts with solid-state ones.

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

Communications abilities began to appear in approximately 1973. The first such system was Modicon's Modbus. The PLC could now talk to other PLCs and they could be far away from the actual machine they were controlling. They could also now be used to send and receive varying voltages to allow them to enter the analogue world. Unfortunately, the lack of 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 (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.

CHAPTER I

1.1.THE TYPES OF PLC

In general, PLC divides to three sections;

*Central Processing unit(CPU)

*The input/output section

*The programming device

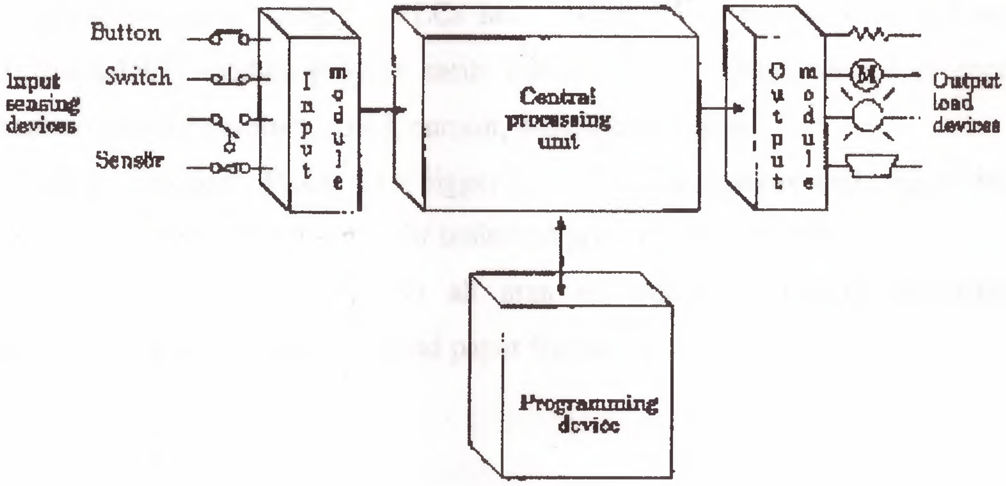


Figure.1.1.1. PLC sections

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

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

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

The desired program loads to processor's memory by programming device or terminal. This program can enter to relay during using ladder logic. Program can be obtained till the main control or machines by sequential processes.

a) PLC size and practice:

There are 3 different categories of PLC; as small, medium and large.

*In small group category, PLC has bigger than input/output of 128 I/O and bigger than memory of 2 KB.

*In medium group category, PLCs have bigger than memory of 32 KB and 2048 I/O. Special I/O module provide easily adaptation in process control practice, analog functions like temperature, press, current, weight and position.

*In large category, PLCs have bigger than 750 KB memory and bigger than input/output of 8192 I/O. This group is for unlimited practice to give force.

Nowadays, PLCs are used in all area of industry along in chemistry, automotive industry production of steel and paper factory.

b) I/O unit:

I/O unit forms is the input/output rack of PLC. I/O unit receives 120 Vac signal in device or processing devices and transforms 5 Vdc signal form. In output units controller signals (5Vdc) are used to devices or processor control as 120 Vac. These output signals provide low current control that used in power electronic elements or optic isolators. Input/output unit in PLC can be put in the same structure or different structure with CPU. This standard input/output unit is in the following shape.

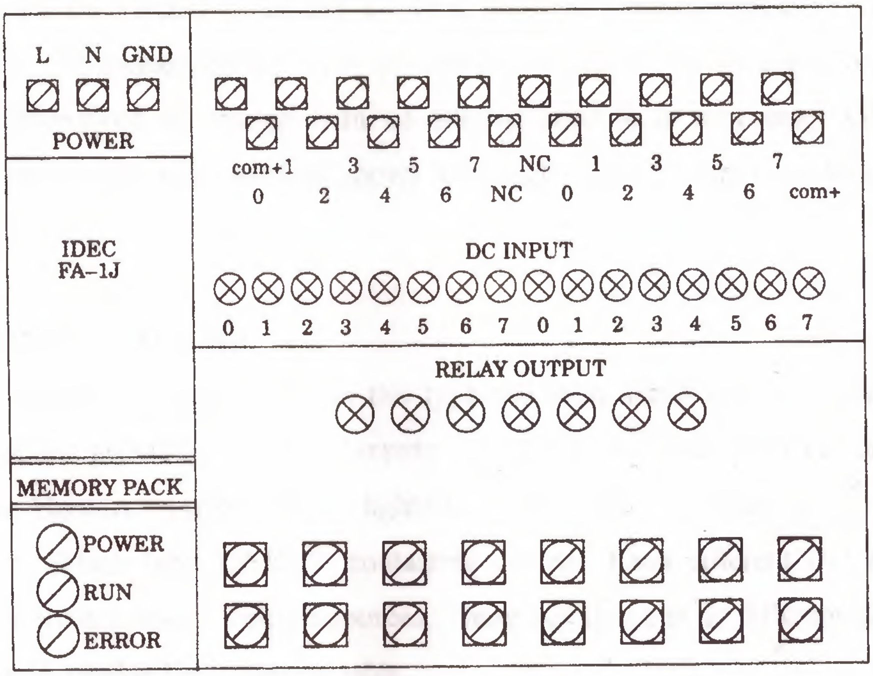


Figure. 1.1.2. In the same structure CPU with PLC I/O unit

Between processor and I/O rack communication different connection cables are permitted. This condition is as the following figure 1.1.3.

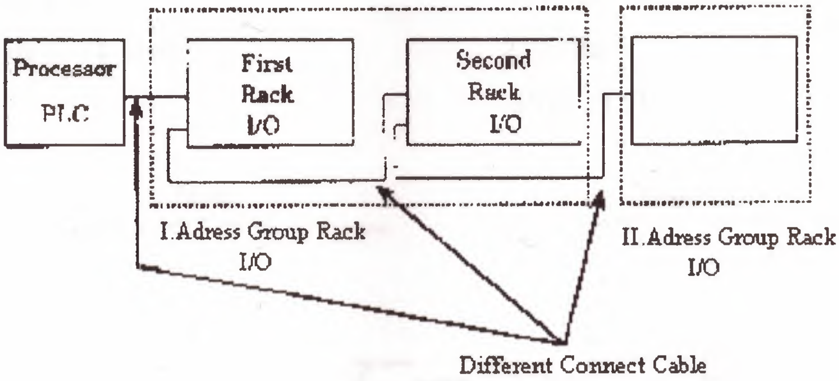


Figure.1.1.3 Between Processing I/O Racks communication

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

c) Different I/O units:

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

Input Interface	Output Interface
24Vac/dc	12-48Vac
48Vac/dc	120Vac
120Vac/dc	230Vac
230Vac/dc	120Vdc
5Vdc (TTL level)	230Vdc
	5Vdc (TTL level)

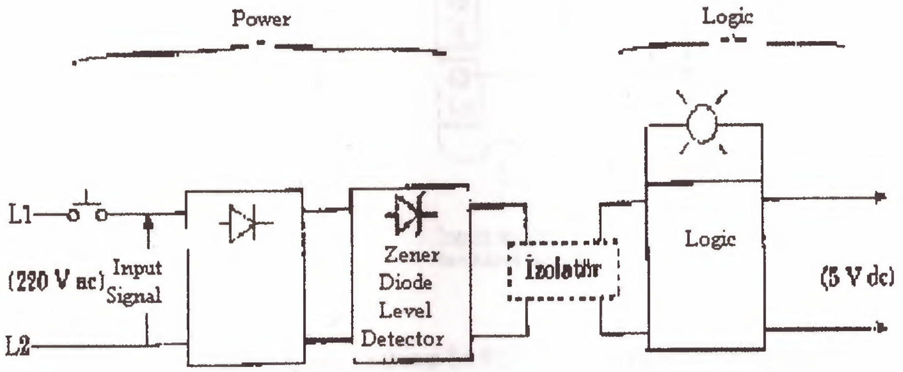


Figure.1.1.4 AC input interface block diagram

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

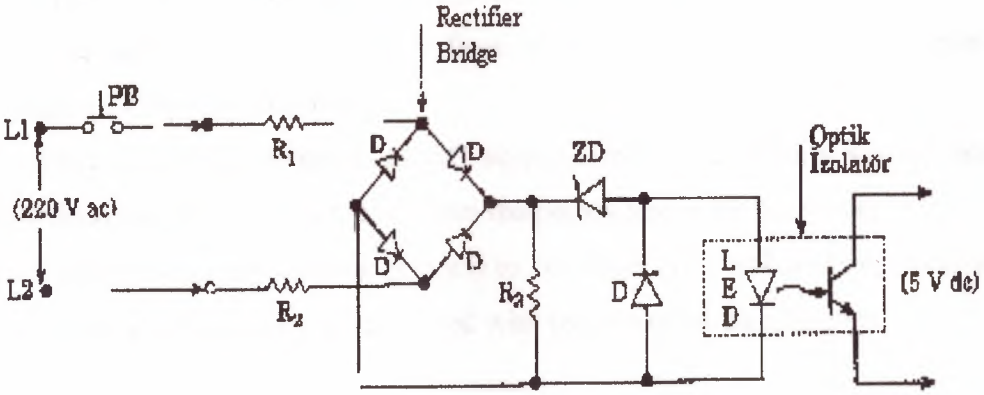


Figure.1.1.5. Simplified Circuit For a AC Module

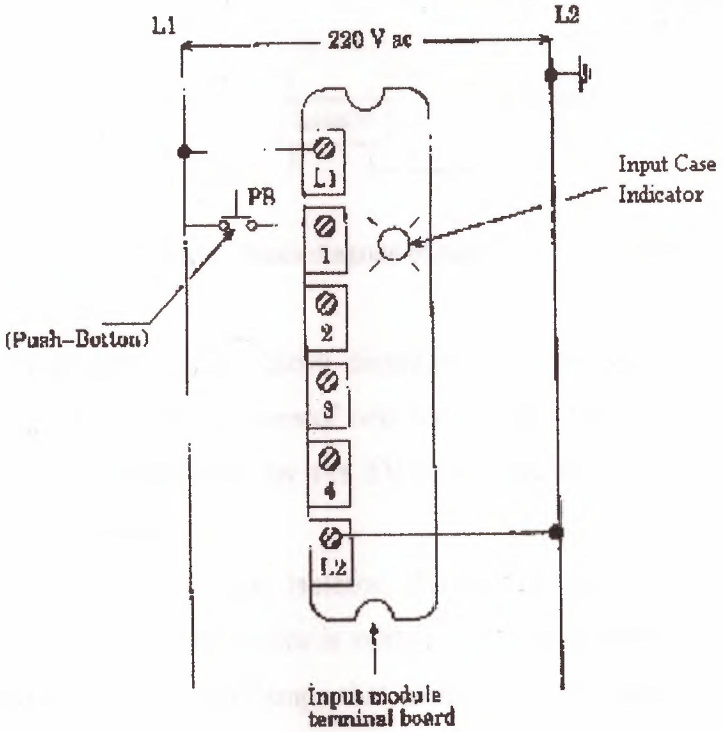


Figure.1.1.6. Linking To PLC Input Unit of 220V Input

Figure 1.1.4 and 1.1.5 shows figural diagram of Ac input module for input, also figure 1.1.6 shows connect terminal.

When push button shuts down, bridge type treatment exercise 220V AC voltage from R_1 and R_2 resistance's.

Zener diode (ZD) voltage limit regulates according to low level voltage.

When light come to processor from led with phototransistor that means low level voltage (SV'dc) is transmitted.

Optic isolator separates high AC voltage from logic circuits also protects to processor from damages, which comes from temporary line voltage change.

Furthermore, optic isolator protects to processor from effect of electrical noise.

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

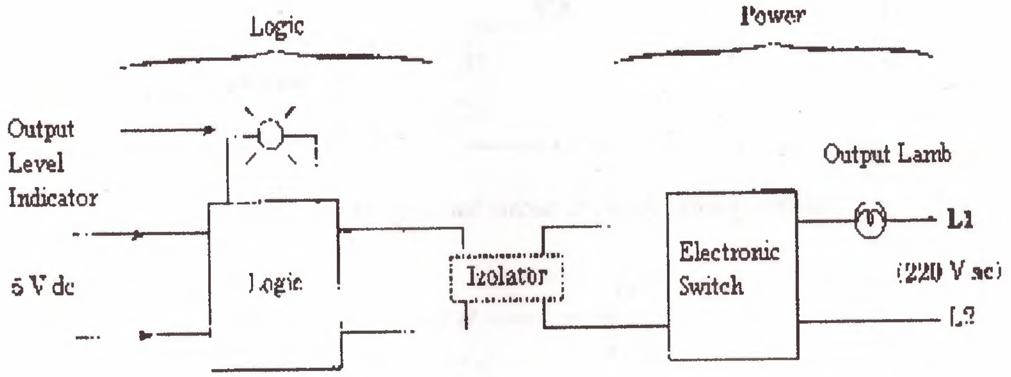


Figure. 1.1.7 typical a block diagram of output interface module.

Figure. 1.1.7 shows typical a block diagram of output interface module. Also output module, as input module, composes of two departments such as power and logic.

Device in output is controlled by the 5V comes from logic unit. In this unit, processor sets output conditions.

When processor, led, in optic isolator, distributing light exercises an output voltage (5V' dc), however, phototransistor is switching and conducting. This means that to detect and conduct of triac, and lamp, that uses as output element, turn on ON condition.

When led in logic unit turn off, logic become 0 condition and phototransistor cannot conduct. If a DC device in output will be controlled, it is carried with circuit.

PLC device will not be damaged from optic isolation that will be from power department.

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

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

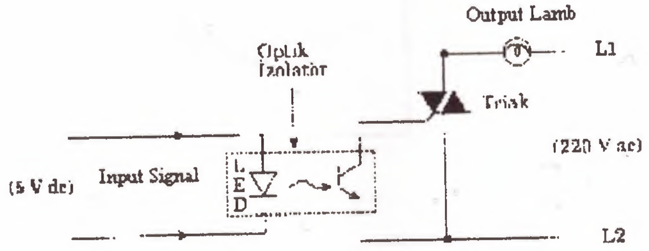


Figure. 1.1.8 Simplified circuit of an AC output module.

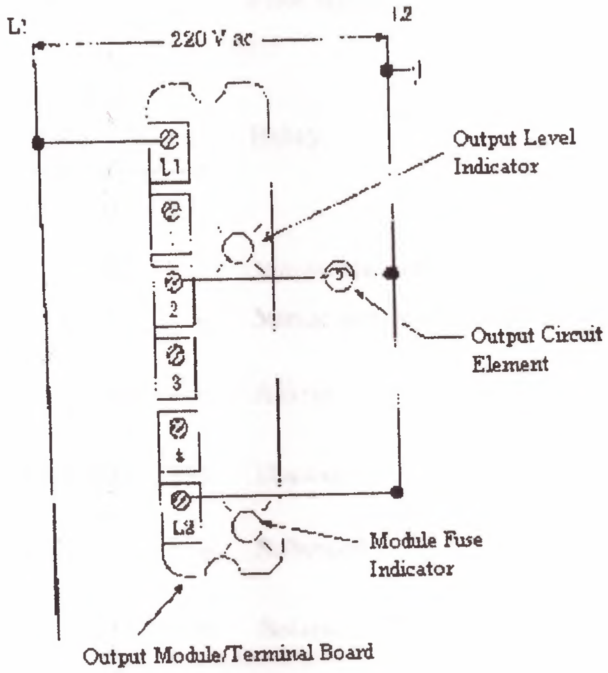


Figure. 1.1.9 Internal wire connection typical an output module

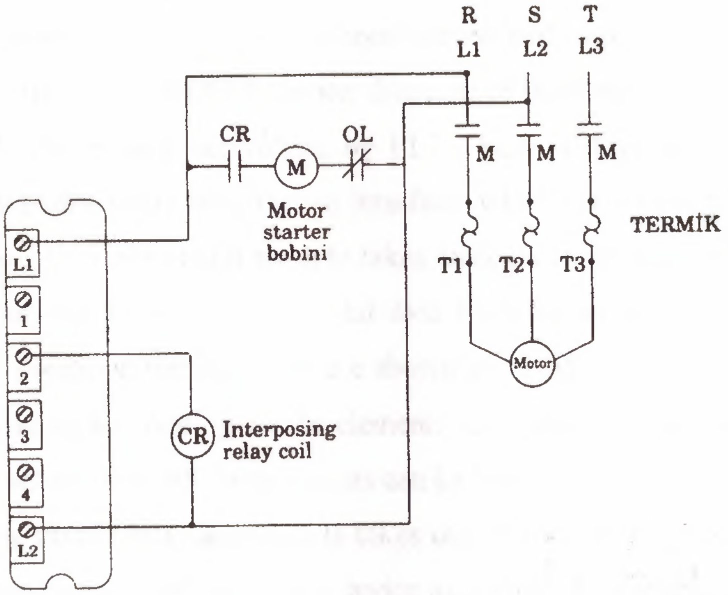


Figure. 1.1.10 Sensor connection points

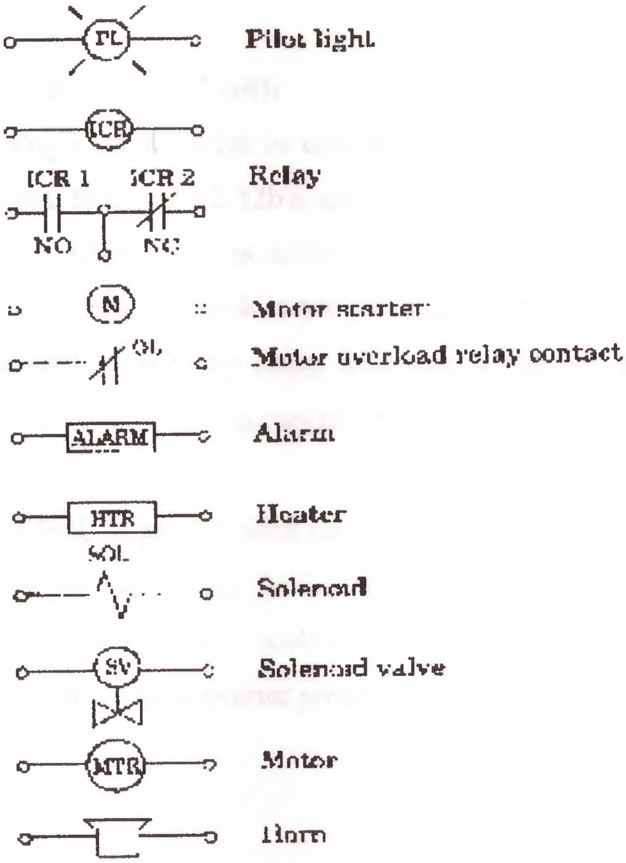


Figure. 1.1.11 Symbols of output control circuit

d) Analog input/output unit (I/O modules):

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

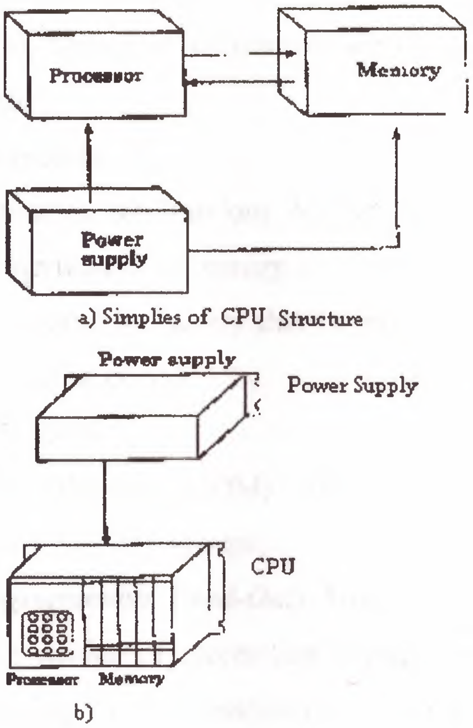
Analog output interface module takes digital data from processor, charges rate with voltage and current and controls a device as analog. As a whole digital data passes from Digital/Analog output device are small motors, valves and analog measure devices.

e) CPU (Central Processing Unit):

Central Processing Unit provides to communicate between power supply and processor memory modules. In figure 1.2.12b it can find covered both of two units.

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

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



a) Simplies of CPU Structure

b)

Figure. 1.1.12 CPU; the elements of central processing unit (a) the structure of simplified CPU (b) power supply unit different from CPU.

f) Processor-Memory Module:

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

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

1.2. MEMORY DESIGN

Memory is used to store data. This stored information is related with which output sign will be store as, which shows input, and the structure of program necessary amount of memory. It stores special information parts, which is named as memory bit. 1 byte = 8 bit, 1024byte = 1kbyte and the number of memory capacity is stated these units.

The memory types are divided into two groups;

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

a) I. Group Memories:

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

b) II. Group Memories:

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

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

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

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

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

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

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

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

1.3. PROGRAMMING DEVICES

The most important one of features of programmable controller is to have programming elements, which are useful. Programming device provides transformation between operator and circuit of controller. (Fig. 1.3.1)

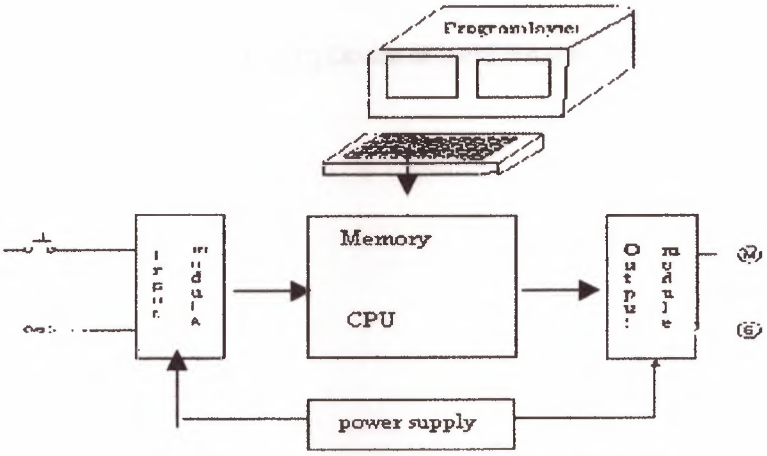


Figure. 1.3.1. Transformation of PLC Circuits

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

Generally, industrial CRT terminals in many devices are used for programmable controllers. These terminals include indicator units, keyboards and CPU and they provide to communicate necessary order.

The advantage of CRT is to check program is easily on monitor.

In small PLCs programming is used cheap, moveable, small and mini programmable devices. The monitor of this type of programming monitor is liquid crystal screen instead of CRT tube, which name LCD. On mini program there are LCD monitor program coding keys and special functions keys. FA2 of programming device IDEC FA1 Junior module is shown at table 1.3.2.

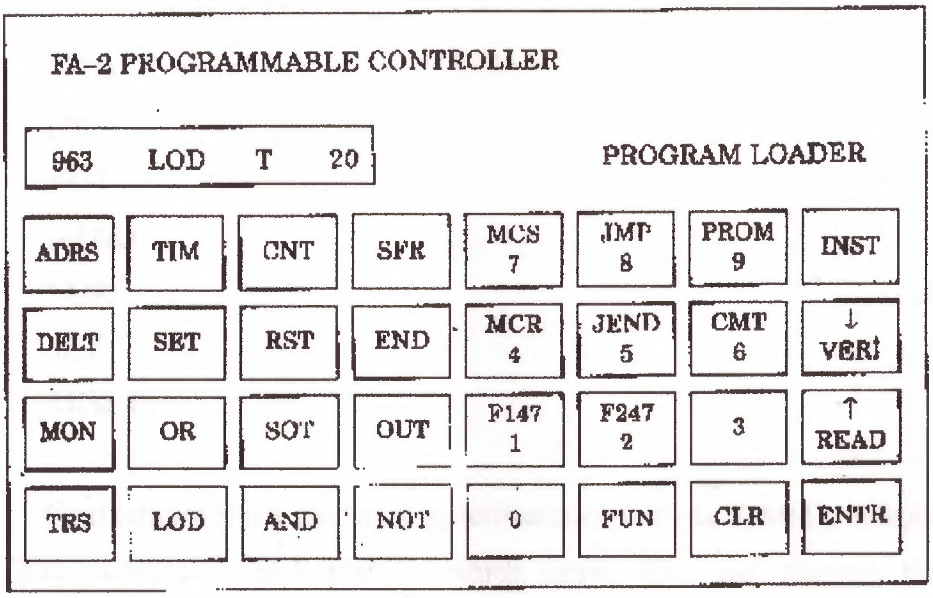


Figure. 1.3.2. Programming Device of IDEC FA-1 PLC.

CHAPTER II

2.1. PLC PROGRAMMING SOFTWARE

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

AND

OR

NOT

NAND

NOR

SET

RESET

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

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

2.2. CREATE OF LEADER DIAGRAM

a) Start Commands:

These commands are first element of program. There are two type contact conditions as at table 2.2.1. First normally is open also second close.

Normally, starting with open contact this program command is to get command as LD IN, LD, LOD A, on PLC device. And also close contact is stated as LDI, LD NOT, LOD NOT, AN.

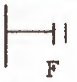
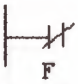
LADDER SYMBOL	COMMAND LINE					
	IDEC	FESTO	AEG	Mitsubishi	Siemens	OMRON
 F Normally open contact	LOD F	LD FLAG F LD IN F	UF	LD F	A F	LD F
 F Normally close contact	LOD NOT F	LD NOT FLAG F LD NOT IN F	UN F	LDI F	AN F	LD NOT F

Table 2.2.1. Load Exercising

Note: in table F value is constant and input/output interval relay, special relay, timer, counter can be SFR number.

According to this table at MITSUBISHI and HITACHI model normally open contact is shown with LD, also close contact is shown with LDI.

Also at AEG PLC, U (UND) command is used for open contact and (UN) UND-NICHT command is used for closed contact.

Also at SIEMENS PLC, A (AND) command is used to open contact and AN (AND-NOT) is used for closed contact.

At OMRON PLC, open contact is shown LD, also close contact is shown with LD NOT.

Also at FESTO PLC, open contact LD FLAG is used for flag load other conditions LD IN command is used to contact load. In normally, also close contact is programmed for flag exercising as LD NOT FLAG... For other contacts are programmed as LD NOT IN...

b) AND and OR Exercising:

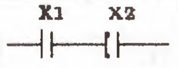
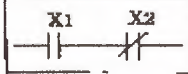
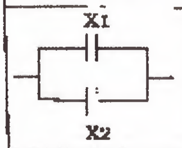
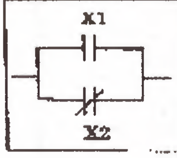
LADDER SYMBOL	COMMAND LINE						
	IDEC	FESTO	AEG	MITSUBISHI	Siemens SIMATIC	OMRON	HITACHI
	LOD X1 AND X2	LD IN X1 AND IN X2	U X1 U X2	LD X1 AND X2	A X1 A X2	LD X1 AND X2	LD X1 AND X2
	LOD X1 AND NOT X2	LD IN X1 AND NOT IN X2	U X1 UN X2	LD X1 ANI X2	A X1 AN X2	LD X1 AND NOT X2	LD X1 ANI X2
	LOD X1 OR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

Table 2.2.2: Symbol and command line AND and OR exercises.

c) Output Stored Exercises:

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

If commands will be observed, there are similarities between PLC devices that output program commands are different. At both output and input functions, X1, X2, are used as addresses.

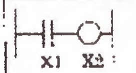
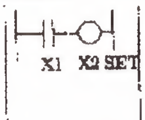
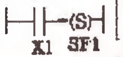
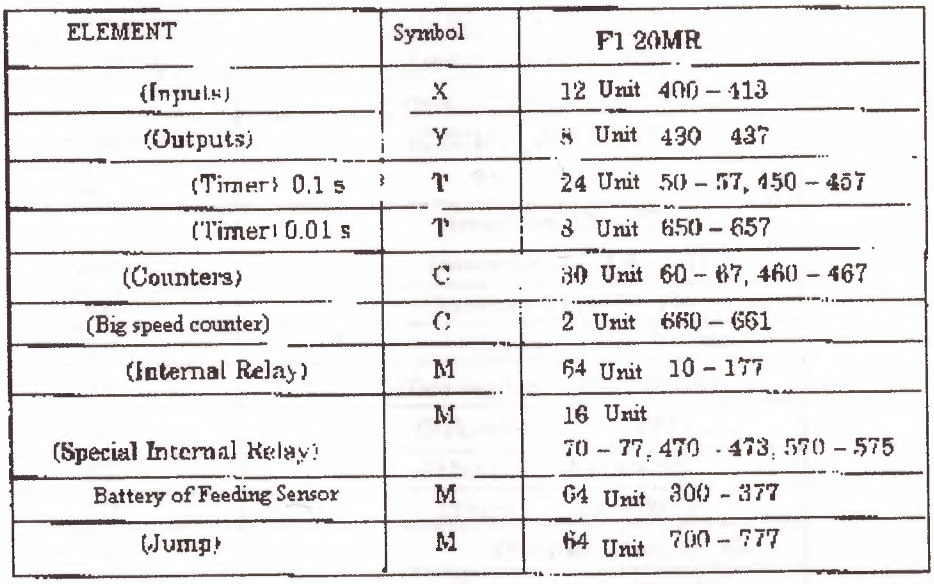
LADDER SYMBOL	COMMAND LINE						
	IDEC	FESTO	AEG	MITSUBISHI	Siemens Simatic	OMRON	HITACHI
	LOD X1 OUT X2	LD IN X1 =OUT X2	U X1 = X2	LD X1 OUT X2	A X1 = X2	LD X1 OUT X2	LD X1 OUT X2
	LOD X1 SET X2	LD IN X1 SE FLAG X2 	U X1 SL X2	LD X1 S X2	A X1 S X2	LD X1 SET X2	LD X1 SET X2

Figure.2.2.3.

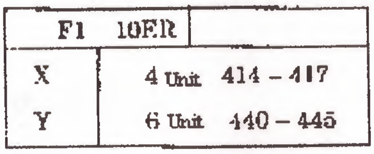
2.3. SPECIFICATION OF EXAMINED PLC

a) Mitsubishi F1 20 MR



ELEMENT	Symbol	F1 20MR
(Inputs)	X	12 Unit 400 – 413
(Outputs)	Y	8 Unit 430 437
(Timer) 0.1 s	T	24 Unit 50 – 57, 450 – 457
(Timer) 0.01 s	T	8 Unit 650 – 657
(Counters)	C	30 Unit 60 – 67, 460 – 467
(Big speed counter)	C	2 Unit 660 – 661
(Internal Relay)	M	64 Unit 10 – 177
(Special Internal Relay)	M	16 Unit 70 – 77, 470 - 473, 570 – 575
Battery of Feeding Sensor	M	64 Unit 300 - 377
(Jump)	M	64 Unit 700 – 777

Table 2.3.1: table of element and element numbers



F1 10ER	
X	4 Unit 414 – 417
Y	6 Unit 440 – 445

Table 2.3.2. Increasing unit

F1 20 MR PLC as 12 inputs 8 outputs, which we use. If more input and output are necessary, input/output-increasing units are plugged to PLC. These units have various numbers output and input. At table 2.3.1, there are 4 inputs 6 outputs for F1 10 ER model.

b) Siemens Simatic S5-90U

Element Name	ELEMENT ADRESS
(Input)	I0.0 – I127.7
(Output)	Q0.0-Q127.7
(Flag)	(retentive) F0.0 – F63.7
(Flag)	(nonretentive) F64.0 – F127.7
Accumulator	ACCUM1 ACCUM2
Timer	T0 – T31
(Counter)	(retentive) C0 – C7 (nonrententive) C8 – C31
KB	(Constant) 1 byte 0 – 255
KC	(Constant count) 0 – 999
KF	(Tam sayılar) - 32768 +32767
KF	(Heksadesimal) 0 – FFFF
KY	(2 byte) 0 - 255 (her bit)
KT	(Timer) 0.0 – 999.3
FB	(Function block) 0 63
DB	(Data block) 2 – 63 [9,10]

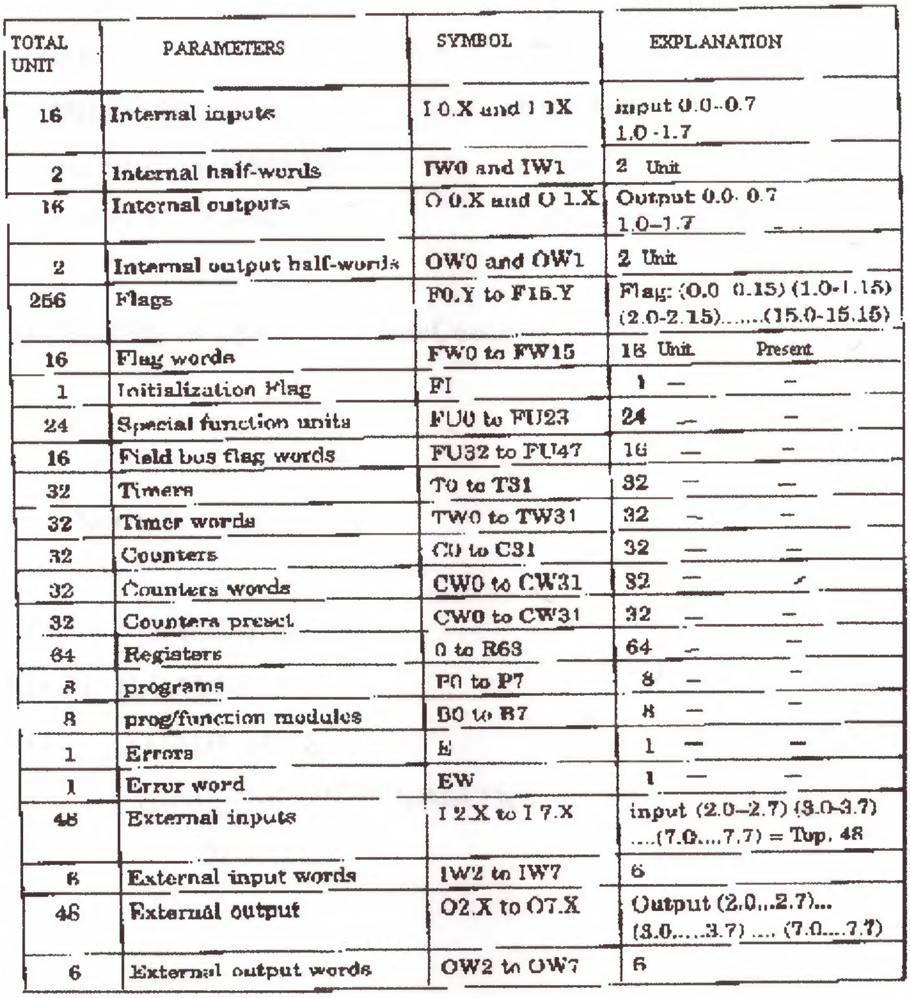
Table 2.3.3: Specifications of S5-90U model Siemens Simatic.

c) AEG Teachware modicon A020

Operand Type	Operand	Unit
(inputs)	E1 – E24	24
(outputs)	A1 – A16	16
Analog Input	EWA 1 – EWA 4	4 analog
Analog Output	AWA 1	1 analog
Memory	M1 – M128	128 Unit
Timer	T1 – T16	16 timer
Counter	Z1 - Z16	16 Counter

Table 2.3.4. Specifications of AEG Teachware A020

d) FESTO (FPC 202C)



TOTAL UNIT	PARAMETERS	SYMBOL	EXPLANATION
16	Internal inputs	I 0.X and I 1X	input 0.0–0.7 1.0 -1.7
2	Internal half-words	IW0 and IW1	2 Unit
16	Internal outputs	O 0.X and O 1.X	Output 0.0- 0.7 1.0–1.7
2	Internal output half-words	OW0 and OW1	2 Unit
256	Flags	F0.Y to F15.Y	Flag: (0.0 0.15) (1.0-1.15) (2.0-2.15)......(15.0-15.15)
16	Flag words	FW0 to FW15	16 Unit Present
1	Initialization Flag	FI	1 — –
24	Special function units	FU0 to FU23	24 – –
16	Field bus flag words	FU32 to FU47	16 — –
32	Timers	T0 to T31	32 – –
32	Timer words	TW0 to TW31	32 – –
32	Counters	C0 to C31	32 — –
32	Counters words	CW0 to CW31	32 – –
32	Counters preset	CW0 to CW31	32 – –
64	Registers	0 to R63	64 – –
8	programs	P0 to P7	8 – –
8	prog/function modules	B0 to B7	8 — –
1	Errors	E	1 — —
1	Error word	EW	1 — –
48	External inputs	I 2.X to I 7.X	input (2.0–2.7) (3.0-3.7) ....(7.0....7.7) = Top. 48
6	External input words	IW2 to IW7	6
48	External output	O2.X to O7.X	Output (2.0...2.7)... (3.0.....3.7) .... (7.0....7.7)
6	External output words	OW2 to OW7	6

Table 2.4.8 Specification of FESTO (FPC 202C) Module PLC

In this table, x=(0,1,2,3,4,5,6,7) and y=(0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15) are.

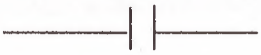
2.4. CREATING COMMAND LINE FOR LOGIC PROCESS

Each process in PLC programming is stated by a command and these commands provides connections of relay and contacts together, designations of outputs, counter, programming of timers and making of arithmetic comparison processes.

In our days, to experience PLC device of all firms are very hard. We will experience five brands. These brands are enough for us.

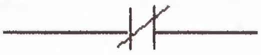
BRAND	**MODEL**
1) IDEC	FA1-JUNIOR (FA1J)
2)FESTO	202-C
3)MITSUBISHI	F120 R
4) SIEMENS-SIMATIC	S5-90U
5) AEG TEACHWARE	MODICON A020

a) Loading of Open and CloseContact:



Normally open contact

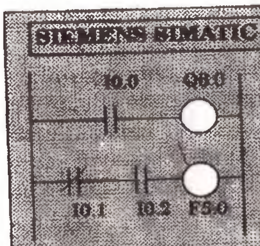
LOD	(LOAD)-IDEC
LD IN	(LOAD)-FESTO
LD	(LOAD)- MITSUBISHI
A	(AND)- SIEMENS-SIMATIC
U	(UND)-AEG



Normally close contact

LOD NOT	(LOAD NOT)-IDEC
LD NOT IN	(LOAD NOT)-FESTO
LDI	(LOAD INVERSE)- MITSUBISHI
AN	(AND NOT)- SIEMENS-SIMATIC
UN	(UND NICHT)-AEG

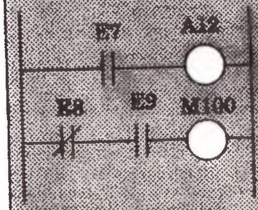
SIEMENS SIMATIC



```
: A   I 0.0
: =   Q 0.0
: AN  I 0.1
: A   I 0.2
: =   F 5.0
: BE
```

Input: I0.0–I127.7
Çıkış Q0.0–Q127.7
Flag F0.0–F63.7
BE, PE, END = PROGRAM SONU

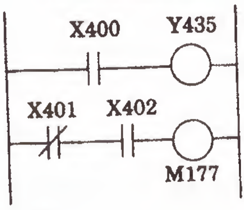
AEG



```
1 U    E7
2 =    A12
3 UN   E8
4 U    E9
5 =    M100
6 PE
```

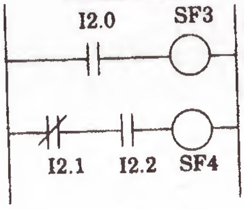
Girişler: E1–E24
Çıkışlar: A1–A16
Saklayıcılar: M1–M128

MITSUBISHI



```
0 LD   X400
1 OUT  Y435
2 LDI  X401
3 AND  X402
4 OUT  M177
```

FESTO



```
0000 LD  IN 2.0
0001  = FLAG 3
0002 LD  IN 2.1
0003 AND IN 2.2
0004  = FLAG 4
```

In here, commands for giving different brand and module normally. Explain to designation of contact and contact numbers are written after command.

In AEG and Siemens PLC, a load command is not used in Siemens Module, open contact command normally is load written A (AND), load process is relazing with AN (AND NOT) command.

In AEG module U (UND) and UN (UND NOT) commands are used for load process. As we know that these commands are used to serial AND and AND NOT exercises.

b) AND exercise:

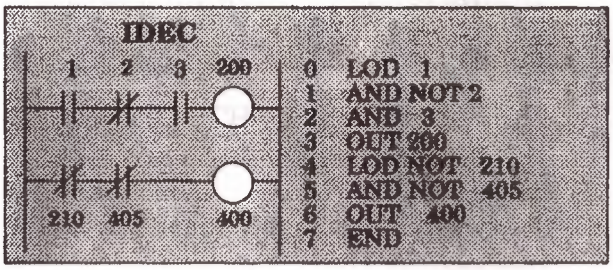
Serial contact linking commands

AND	-(IDEC)
AND IN	-(FESTO)
AND	-(MITSUBISHI)
A(AND)	-(SIEMENS-SIMATIC)
U (UND)	-(AEG)

c) AND NOT exercise:

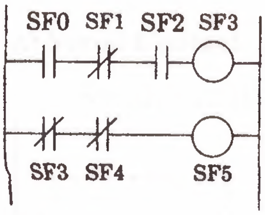
Serial contact linking commands

AND NOT	-(IDEC)
AND NOT IN	-(FESTO)
AND	-(MITSUBISHI)
A(AND)	-(SIEMENS-SIMATIC)
U (UND)	-(AEG)



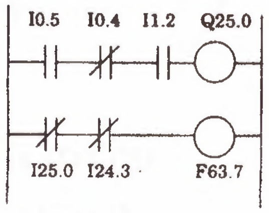
```
0  LOD  1
1  AND NOT 2
2  AND  3
3  OUT 200
4  LOD NOT  210
5  AND NOT  405
6  OUT   400
7  END
```

FESTO



```
0001  LD FLAG  0
0002  AND NOT FLAG  1
0003  AND FLAG  2
0004  = FLAG  3
0005  LD NOT FLAG  3
0006  AND NOT FLAG  4
0007  = FLAG  5
0008  LD PROG  0
```

SIEMENS SIMATIC



```
:   A   I  0.5
:   AN  I  0.4
:   A   I  1.2
:   =   Q  25.0
:   AN  I  25.0
:   AN  I  24.3
:   =   F  63.7
:   BE
```

d) OR exercise:

Parallel contact linking commands

OR	-(IDEC)
OR	-(FESTO)
OR	-(MITSUBISHI)
O(OR)	-(SIEMENS-SIMATIC)
O(ODER)	-(AEG)

e) OR NOT exercise:

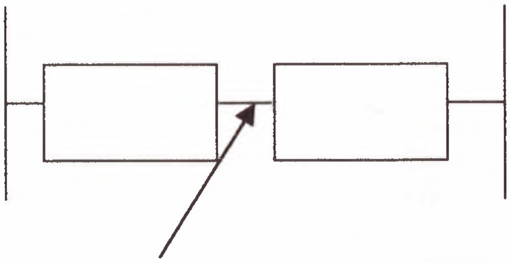
Parallel contact linking commands

OR NOT	-(IDEC)
OR NOT	-(FESTO)
ORI(OR INVERSE)	-(MITSUBISHI)
ON(OR NOT)	-(SIEMENS-SIMATIC)
ON(ODER NICHT)	-(AEG)

2.5. GET COMMUNICATE OF COMMAND BLOCK TOGETHER

a) Serial Contact:

Serial contact



AND LOD -(IDEC)

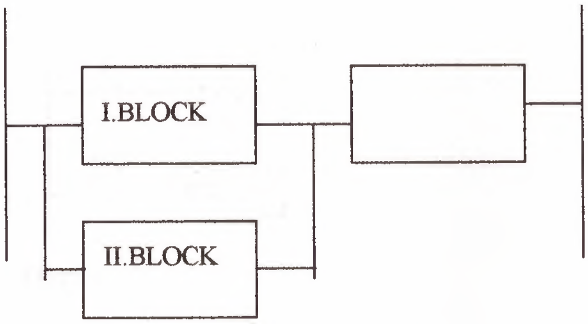
AND LD -(FESTO)

ANB (AND BLOCK) -(MITSUBISHI)

A(................ –(SIEMENS)
)

U(................ –(AEG)
)

b) Parallel Contact:



OR LOD -(IDEC)

```
OR LD                  -(FESTO)
ORB (OR BLOCK)     -(MITSUBISHI)
```

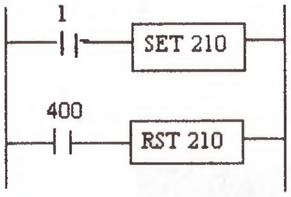
```
A(................          –(SIEMENS)
O
...................
)
```

```
O(                             -(AEG)
)
```

2.6 SET AND RESET INSTRUCTION

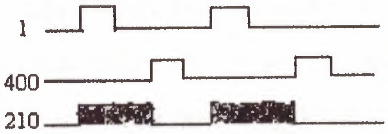
If any of the OFF position relay (eg. Input, output register and internal relay) pass the ON position that is from logic 0 to logic 1. Pass instruction called SET command. RESET command is opposite of SET command that is ON position to OFF position, from logic 1 to logic 0.

Another peculiarity of SET and RESET instructions for working instructions input must be control with relay. It does not require any continuos signal or stroke. That means SET relay always logic 1 position with input relay. If input relay done OFF position does not effect setted relay while that RESET command come.

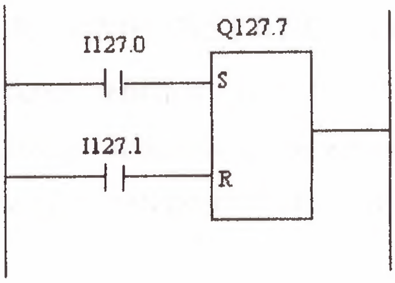


IDEC

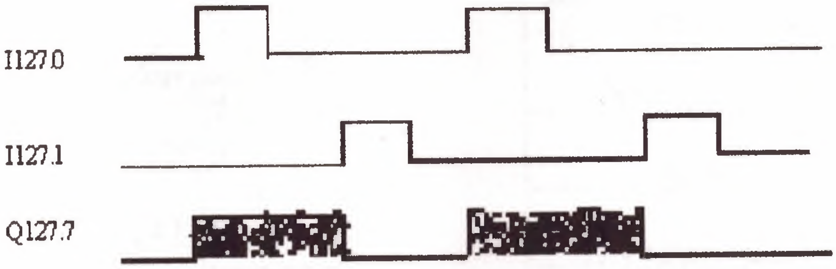
```
0          LOD     1
1          SET     210
2          LOD     400
3          RST     210
4          END
```



SIEMENS



```
A     I    127.0
S     Q    127.7
A     I    127.1
R     Q    127.7
BE
```



2.7.SINGLE OUTPUT INSTRUCTIONS

Our aim is make ON position, on scan time length. With these aim we use two different relays. First one is which makes control, other one is where we take output. The important point is; while controlling relay passing OFF position to ON, where output relay is 1 scan time length mould pass ON position to OFF. It is unimportant that controlling relay is protecting ON position. When the OFF position relay pass to ON position, we take 1 scan time length from output relay.

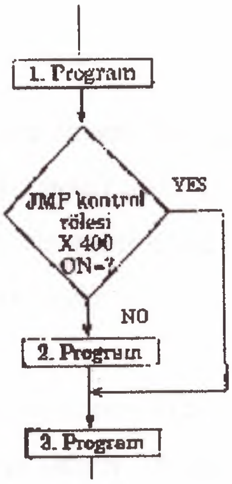
2.8. JUMP INSTRUCTION

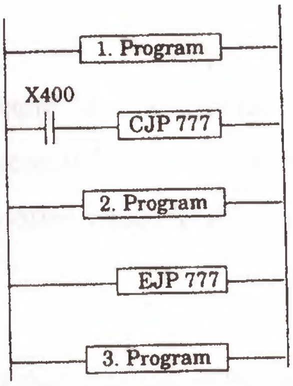
Source peculiarity with JUMP instruction; determined program line or lines makes possive position that jumped by some condition, or conditions. Provided jumped relay is time of the ON position of JUMP command.

MITSUBISHI

CJP (Conditional Jump)

EJP (End of Jump)





Note: JUMP instructions are between 700 – 777

Above program is between the 1. and 2. Programs because of using JUMP instruction, 400-numbered input relay when passed logic 1 position, JUMP instruction come to active condition and 2. program jumped 3. program, and 3. program started to work. Because after the EJP, JUMP ending operation instruction.

With 401 numbered input came logic 1 (ON) jumping operation starts and from CJP 700 until EJP 700 line program line jumps.

Jumping operation goes when X401 OFF. When X401 OFF done program return to work normally and scan operation works line by line.

While X401 OFF position, JMP function does not work. The important point is; before CJP instruction, EJP used must go to last EJP operation. Others will be invalid.

2.9 TIMERS

Let's now see how a timer works. Its exactly what the word says... it is an instruction that waits a set amount of time before doing something. Sounds simple doesn't it.

When we look at the different kinds of timers available the fun begins. As always, different types of timers are available with different manufacturers. Here are most of them:

On-Delay timer-This type of timer simply "delays turning on". In other words, after our sensor (input) turns on we wait x-seconds before activating a solenoid valve (output). This is the most common timer. It is often called TON (timer on-delay), TIM (timer) or TMR (timer).

Off-Delay timer- This type of timer is the opposite of the on-delay timer listed above. This timer simply "delays turning off". After our sensor (input) sees a target we turn on a solenoid (output). When the sensor no longer sees the target we hold the solenoid on for x-seconds before turning it off. It is called a TOF (timer off-delay) and is less common than the on-delay type listed above. (i.e. few manufacturers include this type of timer)

Retentive or Accumulating timer- This type of timer needs 2 inputs. One input starts the timing event (i.e. the clock starts ticking) and the other resets it. The on/off delay timers above would be reset if the input sensor wasn't on/off for the complete timer duration. This timer however holds or retains the current elapsed time when the sensor turns off in mid-stream. For example, we want to know how long a sensor is on for during a 1 hour period. If we use one of the above timers they will keep resetting when the sensor turns off/on. This timer however, will give us a total or accumulated time. It is often called an RTO (retentive timer) or TMRA (accumulating timer).

Let's now see how to use them. We typically need to know 2 things:

What will enable the timer. Typically this is one of the inputs.(a sensor connected to input 0000 for example)

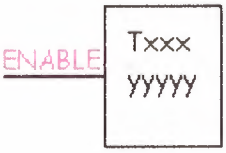
How long we want to delay before we react. Let's wait 5 seconds before we turn on a solenoid, for example.

When the instructions before the timer symbol are true the timer starts "ticking". When the time elapses the timer will automatically close its contacts. When the program is running on the plc the program typically displays the elapsed or "*accumulated*" time for us so we can see the current value. Typically timers can tick from 0 to 9999 or 0 to 65535 times.

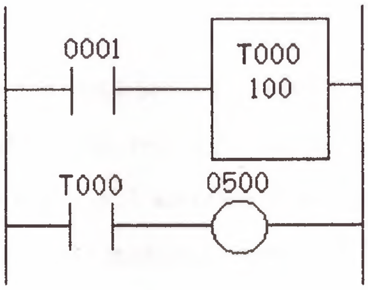
Why the weird numbers? Again its because most PLCs have 16-bit timers. We'll get into what this means in a later chapter but for now suffice it to say that 0-9999 is 16-bit BCD (binary coded decimal) and that 0 to 65535 is 16-bit binary. Each tick of the clock is equal to x-seconds.

Typically each manufacturer offers several different ticks. Most manufacturers offer 10 and 100 ms increments (ticks of the clock). An "ms" is a mili-second or *1/1000th* of a second. Several manufacturers also offer 1ms as well as 1 second increments. These different increment timers work the same as above but sometimes they have different names to show their time-base. Some are TMH (high speed timer), TMS (super high speed timer) or TMRAF (accumulating fast timer).

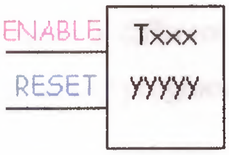
Shown below is a typical timer instruction symbol we will encounter (depending on which manufacturer we choose) and how to use it. Remember that while they may look different they are all used basically the same way. If we can setup one we can setup any of them.



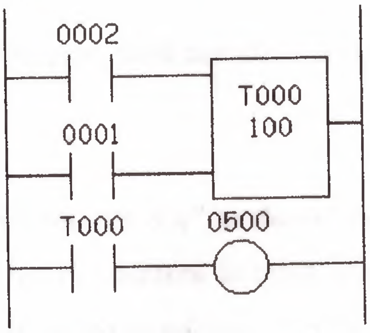
This timer is the on-delay type and is named Txxx. When the enable input is on the timer starts to tick. When it ticks yyyyy (the preset value) times, it will turn on its contacts that we will use later in the program. Remember that the duration of a tick (increment) varies with the vendor and the time-base used. (i.e. a tick might be 1ms or 1 second or...)



In this diagram we wait for input 0001 to turn on. When it does, timer T000 (a 100ms increment timer) starts ticking. It will tick 100 times. Each tick (increment) is 100ms so the timer will be a 10000ms (i.e. 10 second) timer. 100ticks X 100ms = 10,000ms. When 10 seconds have elapsed, the T000 contacts close and 500 turns on. When input 0001 turns off(false) the timer T000 will reset back to 0 causing its contacts to turn off(become false) thereby making output 500 turn back off.



This timer is named Txxx. When the enable input is on the timer starts to tick. When it ticks yyyyy (the preset value) times, it will turn on its contacts that we will use later in the program. Remember that the duration of a tick (increment) varies with the vendor and the time-base used. (i.e. a tick might be 1ms or 1 second or...) If however, the enable input turns off before the timer has completed, the current value will be retained. When the input turns back on, the timer will continue from where it left off. The only way to force the timer back to its preset value to start again is to turn on the reset input.



In this diagram we wait for input 0002 to turn on. When it does timer T000 (a 10ms increment timer) starts ticking. It will tick 100 times. Each tick (increment) is 10ms so the timer will be a 1000ms (i.e. 1 second) timer. 100ticks X 10ms = 1,000ms. When 1 second has elapsed, the T000 contacts close and 500 turns on. If input 0002 turns back off the current elapsed time will be retained. When 0002 turns back on the timer will continue where it left off. When input 0001 turns on (true) the timer T000 will reset back to 0 causing its contacts to turn off (become false) thereby making output 500 turn back off.

2.10. COUNTERS

A counter is a simple device intended to do one simple thing - count. Using them, however, can sometimes be a challenge because every manufacturer (for whatever reason) seems to use them a different way. Rest assured that the following information will let you simply and easily program anybody's counters.

What kinds of counters are there? Well, there are up-counters (they only count up 1,2,3...). These are called CTU,(count up) CNT,C, or CTR. There are down counters (they only count down 9,8,7,...). These are typically called CTD (count down) when they are a separate instruction. There are also up-down counters (they count up and/or down 1,2,3,4,3,2,3,4,5,...) These are typically called UDC(up-down counter) when they are separate instructions.

Many manufacturers have only one or two types of counters but they can be used to count up, down or both. *Confused yet?* Can you say "no standardisation"? Don't worry, the theory is all the same regardless of what the manufacturers call them. A counter is a counter is a counter...

To further confuse the issue, most manufacturers also include a limited number of high-speed counters.

High-speed Counter :

Typically a high-speed counter is a "*hardware*" device. The normal counters listed above are typically "*software*" counters. In other words they don't physically exist in the plc but rather they are simulated in software. Hardware counters do exist in the plc and they are not dependent on scan time.

A good rule of thumb is simply to always use the normal (software) counters unless the pulses you are counting will arrive faster than 2X the scan time. (i.e. if the scan time is 2ms and pulses will be arriving for counting every 4ms or longer then use a software counter. If they arrive faster than every 4ms (3ms for example) then use the hardware (high-speed) counters. (2xscan time = 2x2ms= 4ms)

To use them we must know 3 things:

Where the pulses that we want to count are coming from. Typically this is from one of the inputs.(a sensor connected to input 0000 for example)

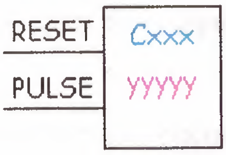
How many pulses we want to count before we react. Let's count 5 widgets before we box them, for example.

When/how we will reset the counter so it can count again. After we count 5 widgets lets reset the counter, for example.

When the program is running on the plc the program typically displays the current or "*accumulated*" value for us so we can see the current count value.

Typically counters can count from 0 to 9999, -32,768 to +32,767 or 0 to 65535. Why the weird numbers? Because most PLCs have 16-bit counters. We'll get into what this means in a later chapter but for now suffice it to say that 0-9999 is 16-bit BCD (binary coded decimal) and that -32,768 to 32767 and 0 to 65535 is 16-bit binary.

Here are some of the instruction symbols we will encounter (depending on which manufacturer we choose) and how to use them. Remember that while they may look different they are all used basically the same way. If we can setup one we can setup any of them.



In this counter we need 2 inputs.

One goes before the reset line. When this input turns on the current (accumulated) count value will return to zero.

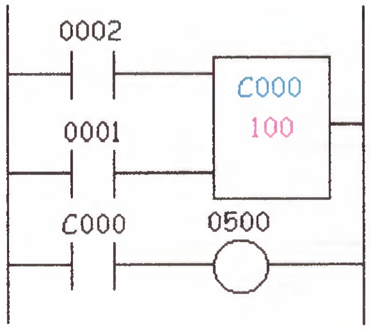
The second input is the address where the pulses we are counting are coming from.

For example, if we are counting how many widgets pass in front of the sensor that is physically connected to input 0001 then we would put normally open contacts with the address 0001 in front of the pulse line.

Cxxx is the name of the counter. If we want to call it counter 000 then we would put "C000" here.

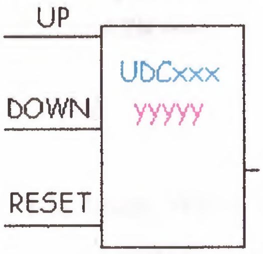
yyyyy is the number of pulses we want to count before doing something. If we want to count 5 widgets before turning on a physical output to box them we would put 5 here. If we wanted to count 100 widgets then we would put 100 here, etc. When the counter is finished (i.e we counted yyyyy widgets) it will turn on a separate set of contacts that we also label Cxxx.

Note that the counter accumulated value ONLY changes at the off to on transition of the pulse input.

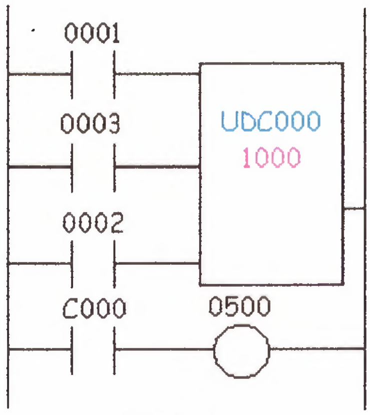


Here's the symbol on a ladder showing how we set up a counter (we'll name it counter 000) to count 100 widgets from input 0001 before turning on output 500. Sensor 0002 resets the counter.

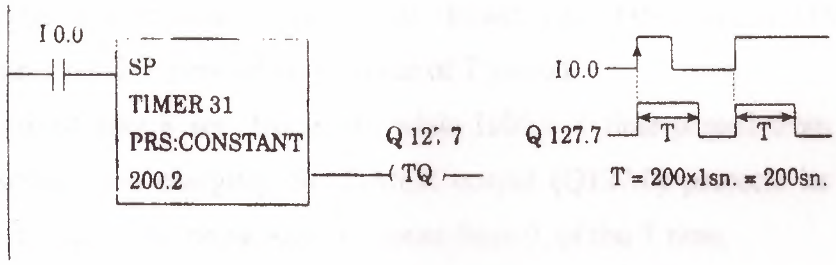
Below is one symbol we may encounter for an up-down counter. We'll use the same abbreviation as we did for the example above.(i.e. UDCxxx and yyyyy)



In this up-down counter we need to assign 3 inputs. The reset input has the same function as above. However, instead of having only one input for the pulse counting we now have 2. One is for counting up and the other is for counting down. In this example we will call the counter UDC000 and we will give it a preset value of 1000. (we'll count 1000 total pulses) For inputs we'll use a sensor which will turn on input 0001 when it sees a target and another sensor at input 0003 will also turn on when it sees a target. When input 0001 turns on we count up and when input 0003 turns on we count down. When we reach 1000 pulses we will turn on output 500. Again note that the counter accumulated value ONLY changes at the off to on transition of the pulse input. The ladder diagram is shown below.



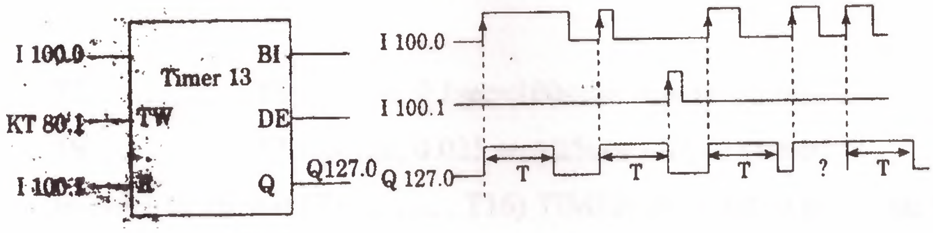
Siemens Simatic : Pulse Timer (SP)



10.0 input sensor works T31 timer. When this sensor takes ON position, settled till 200 sec, Q127.7 out put done 1. Even time over, if input signal I0.0 logic 1, output will reset.

```
: A     I     0.0
: L     KT    200.2
: SP    T     31
: =     Q     127.7
: BE
```

Extended Pulse Timer



```
: A     I     0.0
: L      KT   100.2
: SD     T    12
: A      I    0.1
: R      T    12
: A      T    12
: =      Q    100.0
: BE
```

This kind of timer controls I100.0 input sensor 13 numbered TIMER. When I100.0 sensor was made 1, the sensor which was obliged Q127.0 numbered TIMER pass ON position. The important event is the pass of I100.0 to ON position not the time o this sensors ON position. Even I100.0 1msec stays ON position TIMER protects Q127.0 sensor on ON position by the time of T period.

T must stay 8 sec. But mean while I100.0 T time passed from logic 0 to 1 without second time charging. So TIMER output (Q127.0) protects its ON position again. But it returns beginning again to count from 0, of the T time.

```
:  A     I      100.0
:  L     KT     80.1
:  SE    T      13
:  A     I      100.1
:  R     T      13
:  A     T      13
:  =     Q      127.0
```

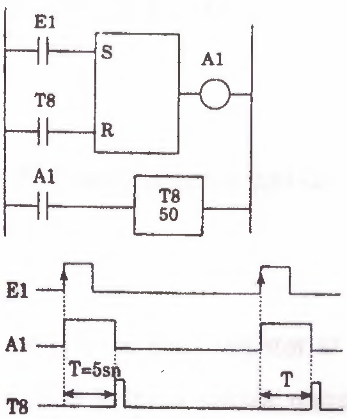
AEG

In the Teachware A020-020 Plus model;

T1....................T8 (8 unit, 0.1sec=100msec rhythm timer)

T9....................T16 (8 unit, 0.025 sec=25msec rhythm timer)

In order to 16 unit (T1..............T16) TIMER there are so programs be smallest and biggest time value is 25 msec which is 110 minutes.



```
1     U     E1
2     SL    A1
3     U     T8
4     RL    A1
5     U     A1
6     =     T8
7           50
8     PE
```

In this example A1 is stetted with E1 output. Reset position is the time of, when T8 pass ON position.

When E1 pass ON position A1 output makes set. By the setting of A1,T8 timer (present value 50x 0.1sec=5sec) count in its inside 5sec and at the end of this time logic done 1. As to program; when T8 is on,A1 output makes resent and T8 output goes OFF position because T8 output is armed reset sensor. The event to care on TIMER present value; chosen TIMER's rhythm times by its number, because of its changes, present value must count right.

The program on above; 413 numbered input sensor and M73 numbered private internal sensor are used to reset 467 numbered counter. Counting input is controlled by 412 numbered input sensor. Present value of counter is showed with K20-20. The input of counter pulse's every present pulse value is lowered 1 degree.

2.11. SHIFT REGISTER

IDEC

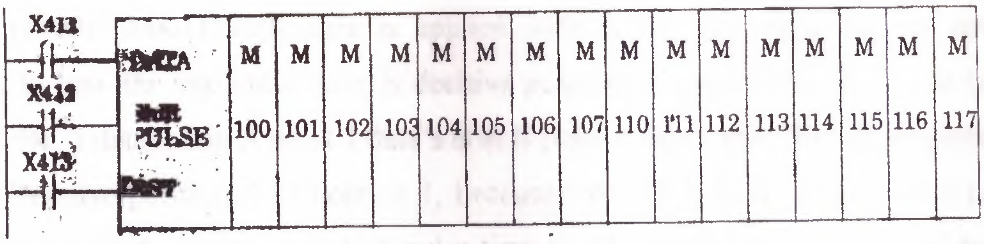
This model in PLC shift register unit has studied extensively.

MITSUBISHI

Internal relay M is used shift register at the some time. So 16 sensor must be 1 group at the same time First helping sensor number, shift register address and following 16 sensor can not use another arm.

Shift Register Addresses

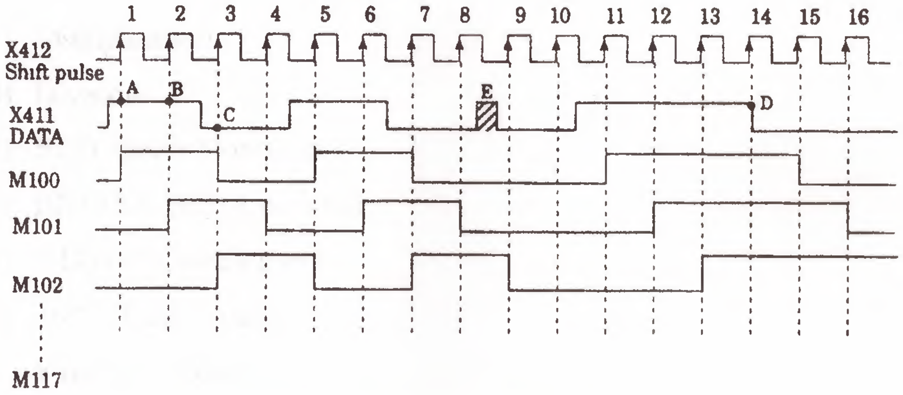
M100 – M117	=	M100..........M107, M110..........M117 = 16 unit
M120 – M137	=	M120..........M127, M130..........M137 = 16 unit
M140 – M157	=	M140..........M147, M150..........M157 = 16 unit
M160 – M177	=	M160..........M167, M170..........M177 = 16 unit
M200 – M217	=	M200..........M207, M210..........M217 = 16 unit
M220 – M237	=	M220..........M227, M230..........M237 = 16 unit
M240 – M257	=	M240..........M247, M250..........M257 = 16 unit
M260 – M277	=	M260..........M267, M270..........M277 = 16 unit
M300 – M317	=	M300..........M307, M310..........M317 = 16 unit
M320 – M337	=	M320..........M327, M330..........M337 = 16 unit
M340 – M357	=	M340..........M347, M350..........M357 = 16 unit
M360 – M377	=	M360..........M367, M370..........M377 = 16 unit

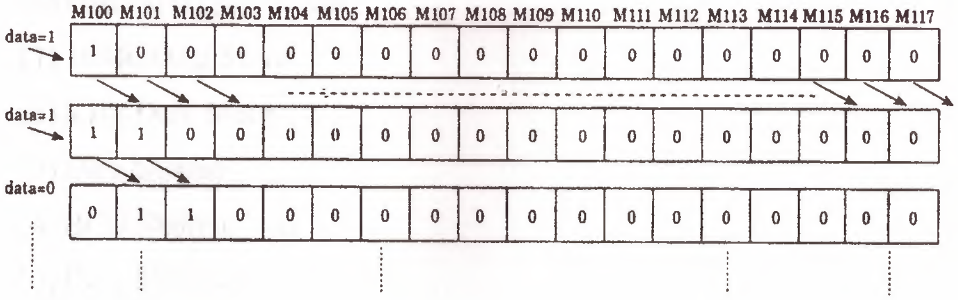


1-Data input: Data signal which must be given to Register, is designed ON-OFF position to X411 sensor. Data, entered to register, firstly 'apply to M100 register. But every shift operation can make by shift pulse.

2-Shift pulse: It is shift input which is transferred to M100 by X411 entered data but while X412 is passing from 0 to 1. It can be used 72 numbered which produces 100msec time pulse or 73 numbered which produces on msec time pulse generator instead of X412.

3-Reset input: X413 input sensor is used for reset of the above. So all the register sensor with X413's passing OFF position to ON position makes reset and pass of position (M100.................M117).





	M100	M101	M102	M103	M104	M105	M106	M107	M108	M109	M110	M111	M112	M113	M114	M115	M116	M117
data=1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
data=1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
data=0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(110011000111100) data is applied with X411 data input on the above example. In here the important thing is decisive position of data m the shift pulse time. For example, 1 data's is in A point 1 data's is in B point 0 data's is in C point examples.

Decisive position in D point is 1, because while shift pulse going from 0 to 1; data value stayed decisively periods 1 pulse time in ON position, so D point of data's the time of going from 1 to 0, shift pulse which is still formed, can't catch and it can't be seen and examples the time of going from 1 to 0 of 14 pulse.

If you attend E area of data diagram; it can't be exampled by data which is between 8 and 9 pulse and it doesn't accept like this data. According to this, for to load of data's to registers is the time of passing the time of piece of referans shift pulse (OFF→ON)

2.12. COMPUTING FUNCTION

one of the most important peculiarity of PLC system is computing and data embroidery function. As a main structure, PLC has this peculiarity.

Some of these are:

1) Addition
2) Subtraction
3) Multiplication
4) Division
5) BCD Binary Converter
6) BINARY BCD Converter
7) 4 DIGIT Comparation
8) 16 Bit Data Loading
9) 8 Bit Data Loading
10) Data Saving-Decrease
11) 16 Bit Data Store
12) 8 Bit Data Store
13) Data Display
14) BCD Shifting Left
15) Data Shifting

SIEMENS (Simatic) Comparison Function

In comparison operations:

!=F (equal)

≠F (not equal)

>F (big)

>=F (big equal)

<F (small)

<=F (small equal)

Instructions are used for make desired comparison, and if YES decision is reached, Q output will give ON position >F control was done at the above. According this, IB0 value which is in ACCU2 will be compared with IB1 in ACCU1, if ACCU2>ACCU1, Q100 will remove ON position. If this condition is not provide, Q100 will stay OFF position.

Arithmetically +F instruction will provide addition of 2 complete number this instruction add ACCUM1 and ACCUM2, of for –F instruction distinct the 2 number.

From ACCU2's contents will distinct ACC1's contents.

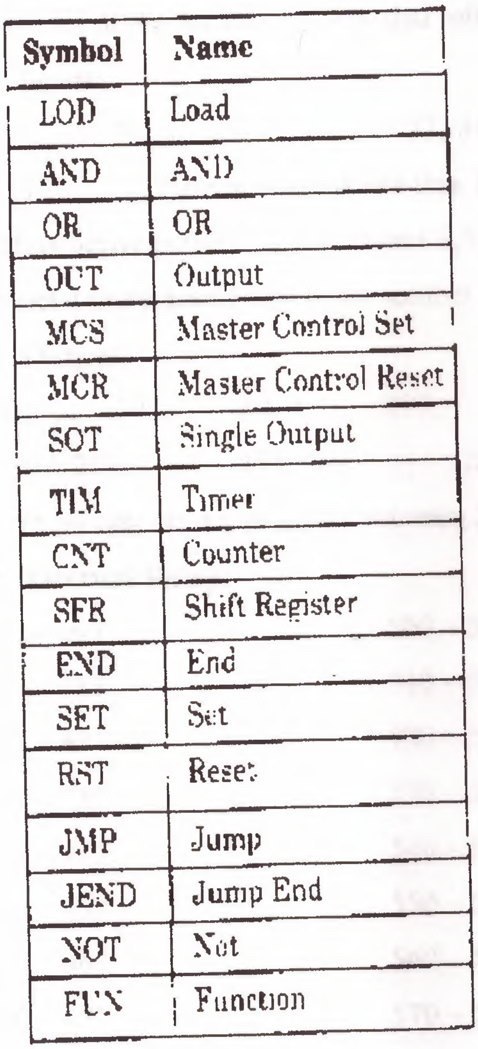
CHAPTER III

DETAIL ANALYSIS OF PROGRAMMING

3.1 BASIC INSTRUCTION WORD

Instruction word list

a) Basic Instructions:



Symbol	Name
LOD	Load
AND	AND
OR	OR
OUT	Output
MCS	Master Control Set
MCR	Master Control Reset
SOT	Single Output
TIM	Timer
CNT	Counter
SFR	Shift Register
END	End
SET	Set
RST	Reset
JMP	Jump
JEND	Jump End
NOT	Not
FUN	Function

b) FUN (Function) Instructions:

We can divide the instructions into 2 parts. These are ;

One – address instruction

Two – address instruction

There are 2 kinds of address instruction. Generally first address is the instruction word. In LOD, AND, OR, OUT, SET, RST, SOT instructions; there is a instruction word and number and addressing is obstructed with this that single addressed instruction.

Two addressed instructions; SFR, SFR NOT, TIM, CNT, FUN 100-146, FUN 200-246, TIM FUN, CNT FUN, FUN 147 and FUN 300. In this instructions first addresses are give instruction word and instruction numbers (Except FUN 147, FUN 300). As for second addresses are present peculiarity according to instruction.

There are some deliver numbers that referenced by FA1J at the below.

c) Input:

0..........7, 10..........17, 20.........27, 30...........37, 40...........47, 50..........57, 60.........67, 70.........77 are numbered like this. In here inputs are considered to OCTAL system which is between 0-77. If you attend 8,9,18,19,28,,29,..........78,79, numbers are not used. In octal there are 64 unit input number between 0-77 (except 8 and 9).

d) Output:

200..........207, 210...........217, 220............227, 230............237, 240...........247, 250.............257, 260.............267 and 270..............277 numbered. Like input there are 64 unit output numbers between 200-277 (except 8 and 9).

e) Internal Relay:

400 – 407
410 – 417
420 – 427
430 – 437
440 – 447
450 – 457
460 – 467
470 – 477
480 – 487
490 – 497

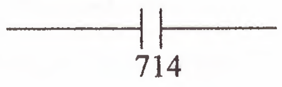
500 – 507
510 – 517
520 – 527
530 – 537
540 – 547
550 – 557
560 – 567
570 – 577
580 – 587
590 – 597

600 – 607
610 – 617
620 – 627
630 – 637
640 – 647
650 – 657
660 – 667
670 – 677
680 – 687
690 – 697

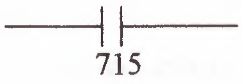
There are 240 units (30x8=240) internal relays between 400 and 697, we can appoint the TIMER, COUNTER or FUN outputs to the any of 240 sensor and then can use of this sensor for take new data or count value.

f) Special Internal Relay:

There are 16 units become 700-707 and 710-717. As an example of these, we can use the signal generator which produces 1 sec clock sign, that means we can use 1 Hz clock pulse sing ready.



We can use the signal generator which produces 0.1 sec clock sign that means 10 Hz clock pulse sign ready.



g) Timer:

There are totally 80 unit timers between 0 and 79. If you attent you can use 8 and 9. You can use any of TIMER that include 0 and 79. In there its enough to know for you that totally there are 80 unit TIMER that include 0-79.

h) Counter:

Totally there are 45 unit counter between 0 and 44. If you attent you can use 8 and 9.

i)Reversible Counter:

It is counter which can be counted forward or review. While other counters can only count forward counters number 45-46 can count forward or review. Counter 45 has up and down pulse input edge yet counter 46 is connected to only one input of up/down situation and when this edge is 1 up and when it be comes 0 it counts down.

j) Shift Register:

There are 128 shift register between 0 and 27 including 8-9.

k) Single Output:

We can use 96 SOT functions between 0 and 95 including 8-9.

l) Data Register:

Between DRO and DR99 and between 800 and 899, we have 100 data register.

3.2 FA1J SERIES ALLOCATION NUMBERS OF SPECIAL RELAYS

As known special relays are 700 and 717 relays except 708 and 704 from these numbers 700 and 705 are unused.

701 and 702 Start Control: When input number 0, which used to start the program is on or if number 500 has been appointed to automatic start process. It starts to turn the program on. Special relays 701 and 702 are off the process of the program is stopped.

703 All Output OFF: All outputs between 200 and 277 are off when special relay 703 turns into ON.

704 Initialize Pulse: Special flag (1 scan time) 704 becomes on as much as the time equalling 1 scan time. When program FA1J started being processed.

704 Numerical Value Error: Is there an error in computing instructions results. 706 becomes on for example; if the result of a subtraction process is lower than –10.000, special relay 706 becomes on. They make sure that the program is correct from the point of view numerical process while they register the programs.

707 Curry and Borrow: It there is carry or borrow in the results at computing instructions. 707 is set for example; in a addition process the total of 2 numbers are higher than 9999,707 is on.

713 1 sec. Timer Reset: When 713 is on special relay 714 is always reset mode.

714 1 sec. Clock: It is possible to take signal generator producing clock sign for one second or clock pulse sign for 1 Hz from special relay 714.

715 100-msec. Clock: We can remove our clock pulse that is for 10 speed by using special relay output of 715 with this sign.

716 Timer/Counter Preset Value Changed: Special relay 716 becomes on when timer counter preset value has been changed into unit of FA1J CPU. It is possible to delete 716 when pressed key of TR S, ENTR and ENTR. If a program is registered in memory.

717 In-operation Output: Relay 717 is always on while FA1J is operating of the program has ended this relay becomes off.

3.3. BASIC INSTRUCTION

Each program written in PLC are started in 2 ways. One at these that we can draw the program with its symbols in the location called Ladder Diagram and load it to the computer as this. The second one is that we can make direct attribution using the key team of PLC. Because of this it will be told example symbol and attribution us. Instructions later whole LOD instruction and the other instructions are being stated.

a) LOD Instructions:

This instructions is used at the beginning of logic diagram lines. It can be used once back by back or more than once to determine the situation at the beginning of the instructions such as AND LOD, OR LOD, SFR, CNT, TIM. As you see below an input relay is wanted to be loaded as a program. Symbol of it is declared as a show in ladder diagram. Program list from the statement.

This program is loaded as 0 LOD 1 and 0 which is seen an address must be given in each line of the end one by one starting from each line of the program. Value is appointed to each line orderly. We have mentioned before which numbers are separated for shift register, output, input, special relay, timer counter. Imaginary internal relay at the machine PLC.

We can divide our load process into 4 groups according to our functions.

b) Input, Output, Internal and Special Relays:

In the examples above example relay circuit of relay in ladder diagram and how the process of key and as a result of this the format seen in deplay was given.

- We can choose a value between 0 and 77 except 8 and 9 in the example of input.
- We can choose a value between 200 and 277 except 8 and 9 in the example of output.
- We can choose a value between 400 and 697 except 8 and 9 in the example of internal relay.

You can use special relay which you need are between 700-717 in the example of special relay for example I use pulse generator of clock for one speed with special relay 714.

c) Timer:

I wanted to use T8 timer from the 80 timers between 0-79 including 8 and 9 here and you see how the load process had been done.

d) Counter:

You can use any counter between 0 and 46 including 8 and 9. Load process is the same as aside.

e) Shift Register:

You can use any register from 128 of them between 0 and 127 including 8 and 9. Shift register numbered 1 was loaded in the next side.

f) AND Instruction:

It is same as AND logic we studied in Logic lessons. Both keys that are connected each other rapidly are on, output is on and is the other situations it becomes OFF in logic. In a multiplying processes both inputs are 1 than output is 1. And had it ended with 2 limit switches and 1 solenoid valve in order to understand the logic better. In diagrams, it is stated as relay ladder diagram and logic diagram. So we can tell that LS1 relay A and LS2 relay is B input and output is Y. In such equality it is that Y=A.B according to the compulsion of Boolean. If both inputs are 1 (ON) Y output will be ON. In other 3 probabilities, output Y will be 0 (OFF). You can see this in the table of truth.

As known, the series of TTL is Logic entegrate containing 4 and gate with 2 inputs in 7408. As in the circuit ¼ has been made equal to ladder diagram by using 7408. In both of them the function of output and working are same.

g) OR Instruction:

Or instruction has the same functions as or gate logic we studied in logic lessons. In here, just only one of the keys are OFF or 1 is enough for output to be 1 as 2 keys are connected in the parallel way. As a result there is addition process and in this process one of the 2 parallel inputs is enough to be one. I gave 2 important information's with or instruction. One as them is out function that is symbolised with 200 in the circle. I will speak about out function 2 or 3 classes later. But now, I gave output of parallel circuit, output 200 for the first time it means that: I mentioned that special relay 704 is a clock pulse generator that has f=1 Hz. You see signal of clock pulse in the diagram. We determined time of 1 and 0 in input relay of 36 by chance now so that nothing will be by chance in the following lessons. Let's accept that there is a time diagram for to learn or let's assume that input 36 is gained by making ON/OFF in the form. If we think that output 200 is connected to a lamb, the situations that lamb will be on are the times that output 200 is 1.

In this example, in order to understand or instructions better firstly, 2 limit switches were connected to each other rapidly and shown a ladder diagram and a solenoid valve control in output of it. And same circuit has been gained Logic equality by using only 1or gate of integrate of 7432. It is enough to make on only one of the inputs for the outputs to be ON in 3 equaliavence circuit to make output OFF. It is necessary to make both parallel inputs OFF. This position was shown in the truths table below.

h) NOT Instructions:

It has the same duty as NOT gate that you studied in the logic lessons. We take the opposite of the sign. If we have a look of the example above, they take the opposite of input relay 1 in PLC. If you carry out 1 logic level to input 1 from the outside, the sign is going to continue from B point as logic 0, because of the instruction of LOD NOT 1.

As it can be seen in time diagram, LOD NOT 1 instructions got the opposite of input 1. It is symbolised

i) AND LOD Instruction:

It is one of the most important of the basic instructions. You have to understand it very well AND LOD instruction, as shown a side is used to connect the instructions to each other in two different blocks. There can be some instructions lines that were mode with or, and, not, input shown before you in the blocks staled before. Watch out that in the programming. Thinks as if you were opening a ladder wall or parentheses. When you are beginning a new block. In each block open the relays and each new block with LOD. If you have write the program and at lost connect the both blocks with AND LOD.

j) OR LOD Instruction:

If you got AND NOT instruction well. It is easy to comprehend OR LOD. As sign in the ladder diagram above, 2 different blocks were connected rapidly this time. PLC this operator in its own memory, it makes it like this in operation register and stock register. There are 1 operation register and maximum 8 stock register that we used to make temporary loading in PLC. Operation register is the register that procedure is mode stock register is assistant register. We have maximum 8 stock register. If we load the I. Block, these instructions are loaded to the register. While II. Block is being loaded. I. Block in operation register is slide to assistant register. Now II. Block was set to operation register and I. Block to stack register. II. Block are connected to each other in a parallel way with OR LOD instruction given later.

3.4.REAL TIME APPLICATIONS

Real Time PLC Fundamentals:

The Real Time PLC is a straightforward software solution that is executed in a Windows environment as an interpreter. The PLC program is executed in a non compiled form, just the same as it is executed in a hardware PLC. The advantage of executing a PLC program with an interpreter is that the PLC status can be displayed in real time without any recompile activities. Diagnosis and testing of a PLC program is much easier and the instruction to be executed can be monitored in real time.

Real Time PLC Operating Systems:

To meet the demands and requirements of our customers and provide flexibility, different hardware and software platforms are available to execute the Real Time PLC. The operating systems Windows 3.1x/95 or Windows NT 4.x can be used. The user also has the ability to install and activate the Real Time PLC on an additional processor board which can be plugged into the motherboard of the PC.

Real Time PLC Compatibility:

As a result of a variety of performance requirements and the need to transfer existing PLC programs, without any modifications into a modern PLC environment, two versions of the Real Time PLC's are available. The Real Time PLC version PLC43 is compatible to the Simatic PLC CPU943. The version PLC45 is compatible to the Simatic PLC CPU945.

Real Time PLC I /O's

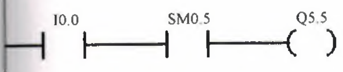
Standard I/O boards may be accessed with the Real Time PLC as well as numerous intelligent hardware boards are available to control bus systems widely used by the industries (e.g. CAN-Bus, Inter Bus, Profi Bus, etc.). There is even a hardware board that will allow you to connect an original SIEMENS S5-115U extension rack, with all possible I/O boards, available. Drivers have been developed to connect bus system interface boards with the Real Time PLC. Drivers for additional boards are easily programmed with Function Blocks using the standard STEP® 5 PLC programming language plus some additional special instructions supplied with the Real Time PLC. Very often only a handful of instructions are needed to realize such a driver.

PROGRAM TITLE COMMENTS

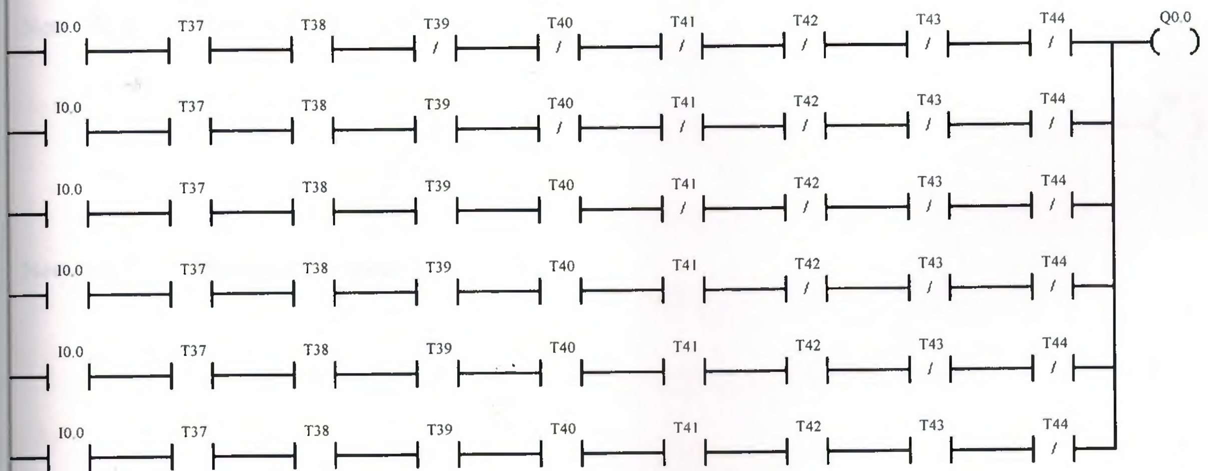
Press F1 for help and example program

Network 1 Flasher

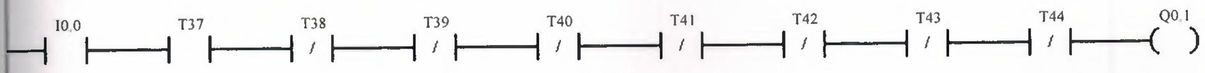
NETWORK COMMENTS



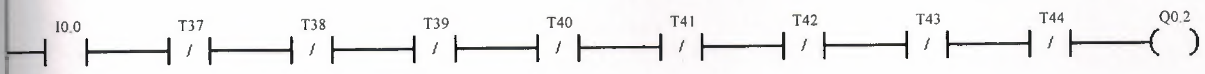
Network 2 Mainroad 1 is red



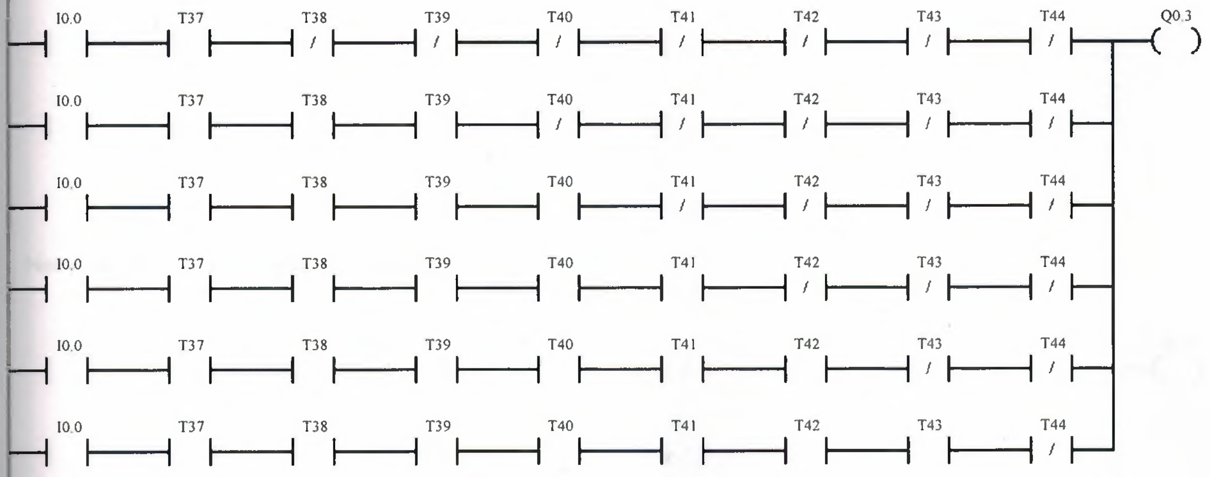
Network 3 Mainroad 1 is yellow



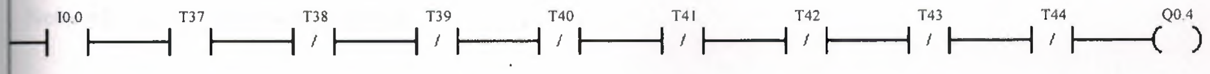
Network 4 Mainroad 1 is green



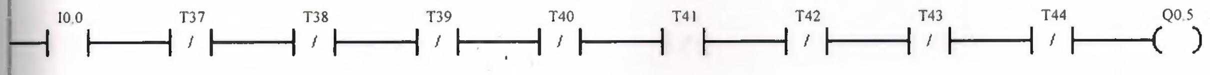
Network 5 Mainroad 2 is red



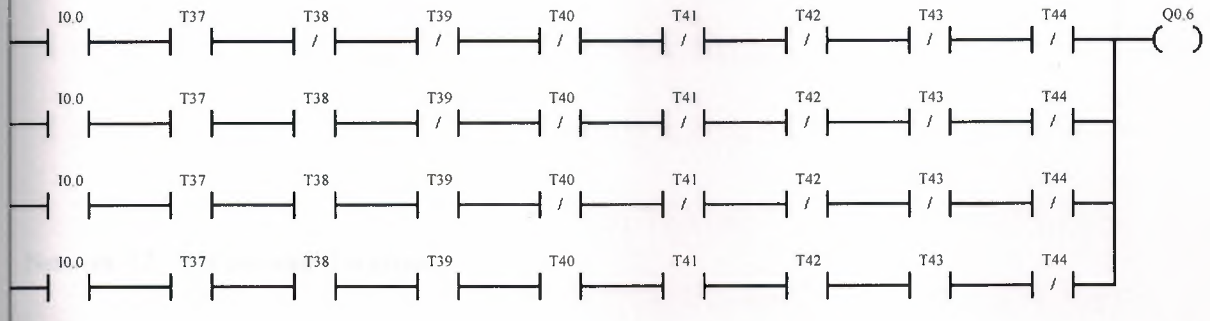
Network 6 Mainroad 2 is yellow



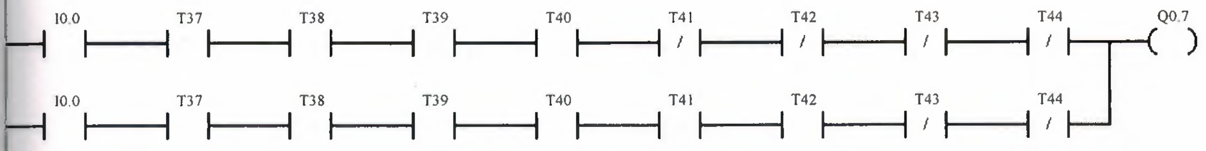
Network 7 Mainroad 2 is green



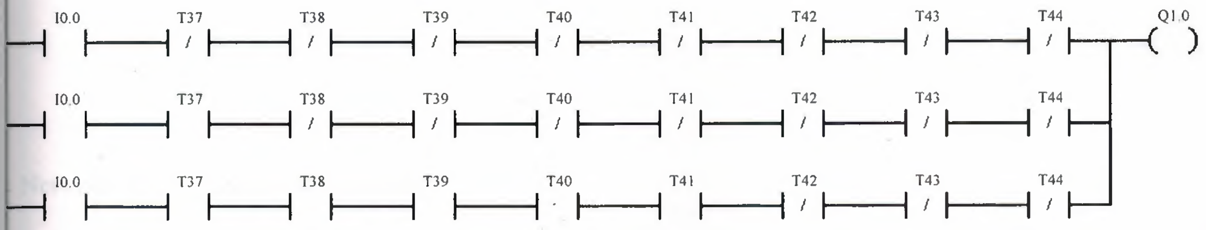
Network 8 Crosswalk 1 is red



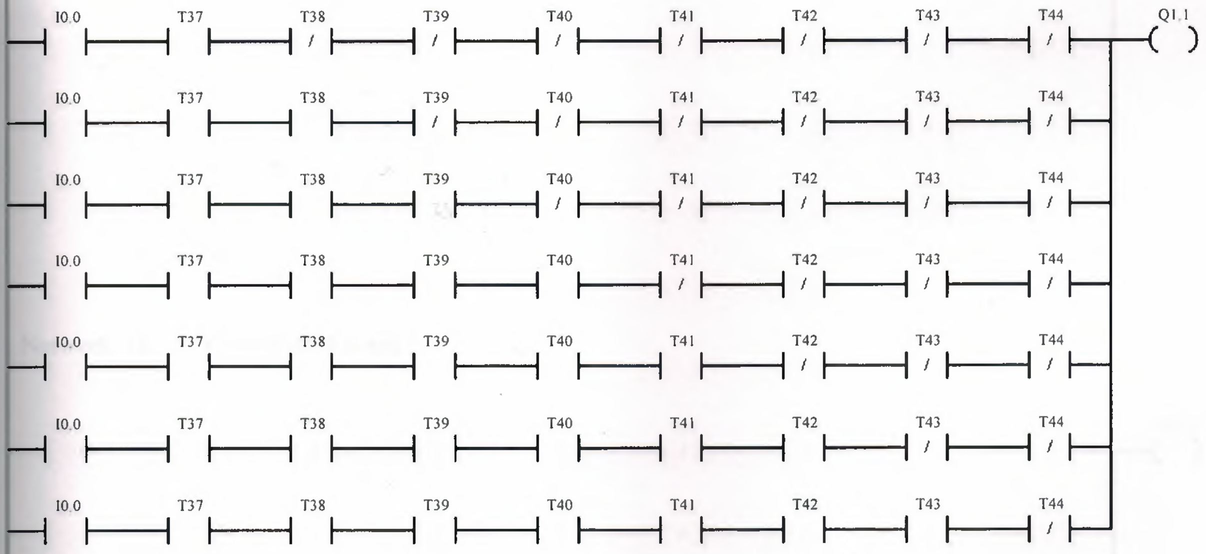
Network 9 Crosswalk 1 is yellow


I0.0 T37 T38 T39 T40 T41 T42 T43 T44 Q0.7
I0.0 T37 T38 T39 T40 T41 T42 T43 T44

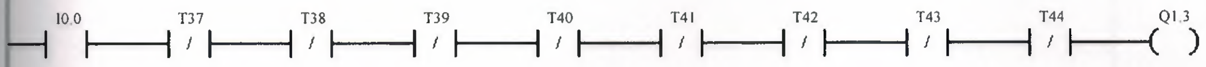
Network 10 Crosswalk 1 is green


I0.0 T37 T38 T39 T40 T41 T42 T43 T44 Q1.0
I0.0 T37 T38 T39 T40 T41 T42 T43 T44
I0.0 T37 T38 T39 T40 T41 T42 T43 T44

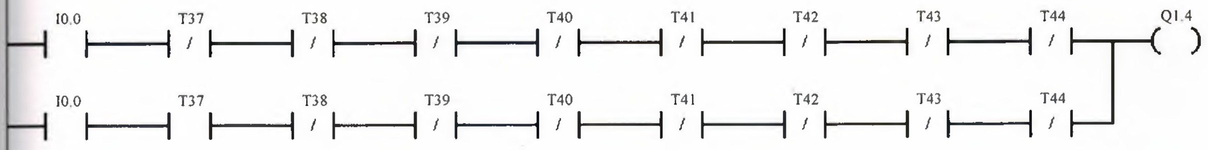
Network 11 Crosswalk 2 is red


I0.0 T37 T38 T39 T40 T41 T42 T43 T44 Q1.1
I0.0 T37 T38 T39 T40 T41 T42 T43 T44
I0.0 T37 T38 T39 T40 T41 T42 T43 T44
I0.0 T37 T38 T39 T40 T41 T42 T43 T44
I0.0 T37 T38 T39 T40 T41 T42 T43 T44
I0.0 T37 T38 T39 T40 T41 T42 T43 T44
I0.0 T37 T38 T39 T40 T41 T42 T43 T44

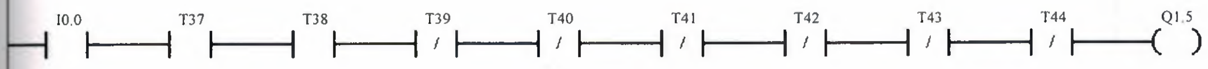
Network 12 Crosswalk 2 is green



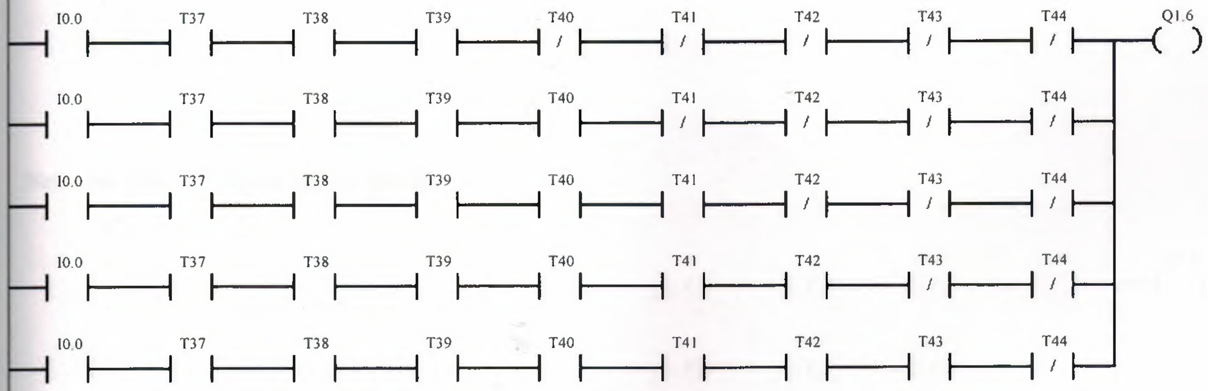
Network 13 Crosswalk 3 is red


I0.0 T37 / T38 / T39 / T40 / T41 / T42 / T43 / T44 / Q1.4
I0.0 T37 T38 / T39 / T40 / T41 / T42 / T43 / T44 /

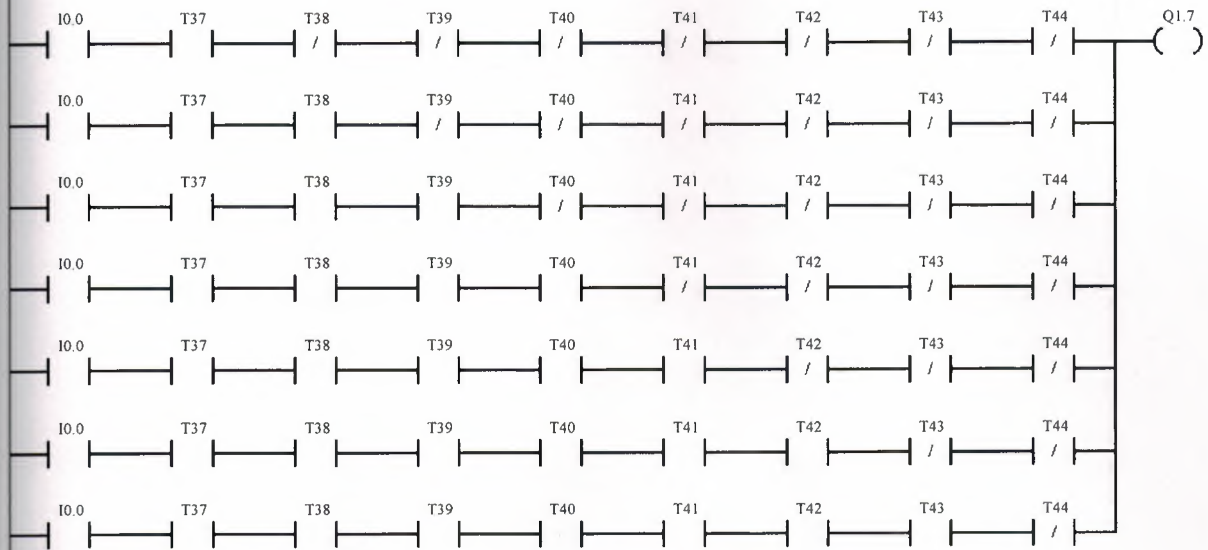
Network 14 Crosswalk 3 is yellow



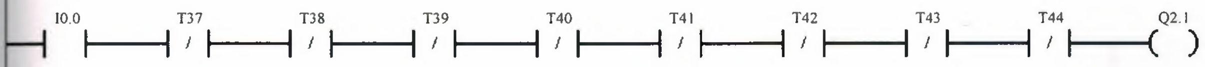
Network 15 Crosswalk 3 is green


I0.0 T37 T38 T39 T40 / T41 / T42 / T43 / T44 / Q1.6
I0.0 T37 T38 T39 T40 T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 T44 /

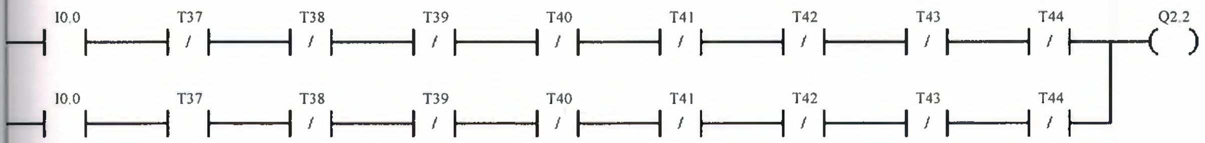
Network 16 Crosswalk 4 is red


I0.0 T37 T38 / T39 / T40 / T41 / T42 / T43 / T44 / Q1.7
I0.0 T37 T38 T39 / T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 T44 /

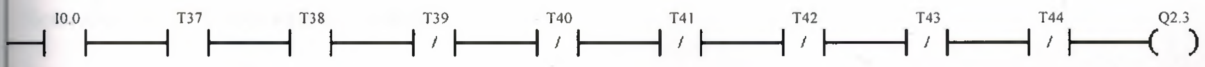
Network 17 Crosswalk 4 is green



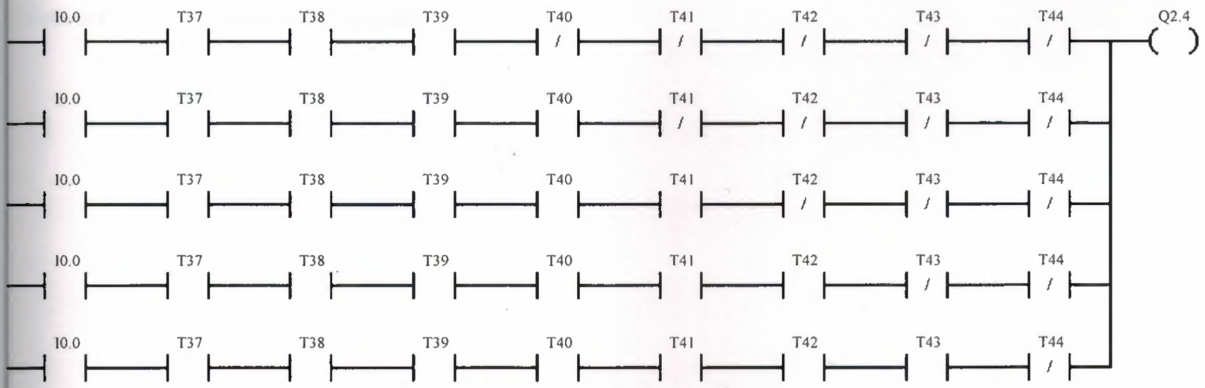
Network 18 Crosswalk 5 is red



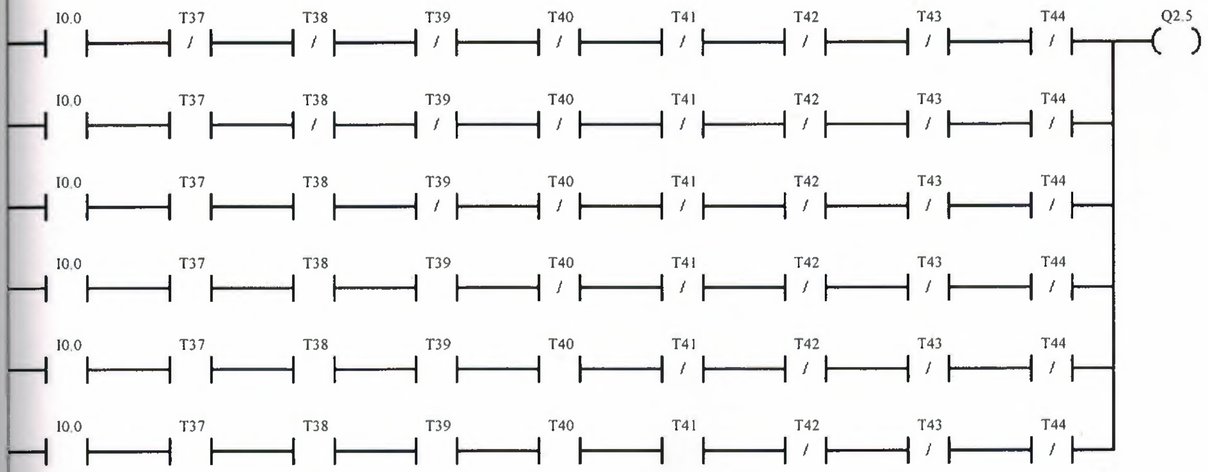
Network 19 Crosstalk 5 is yellow



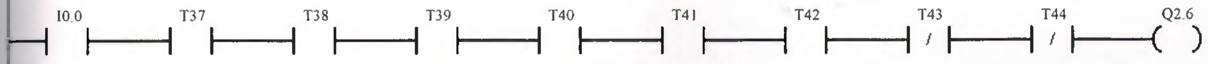
Network 20 Crosswalk 5 is green



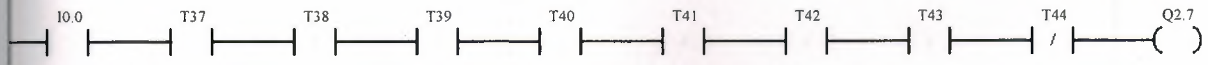
Network 21 Crosswalk 6 is red



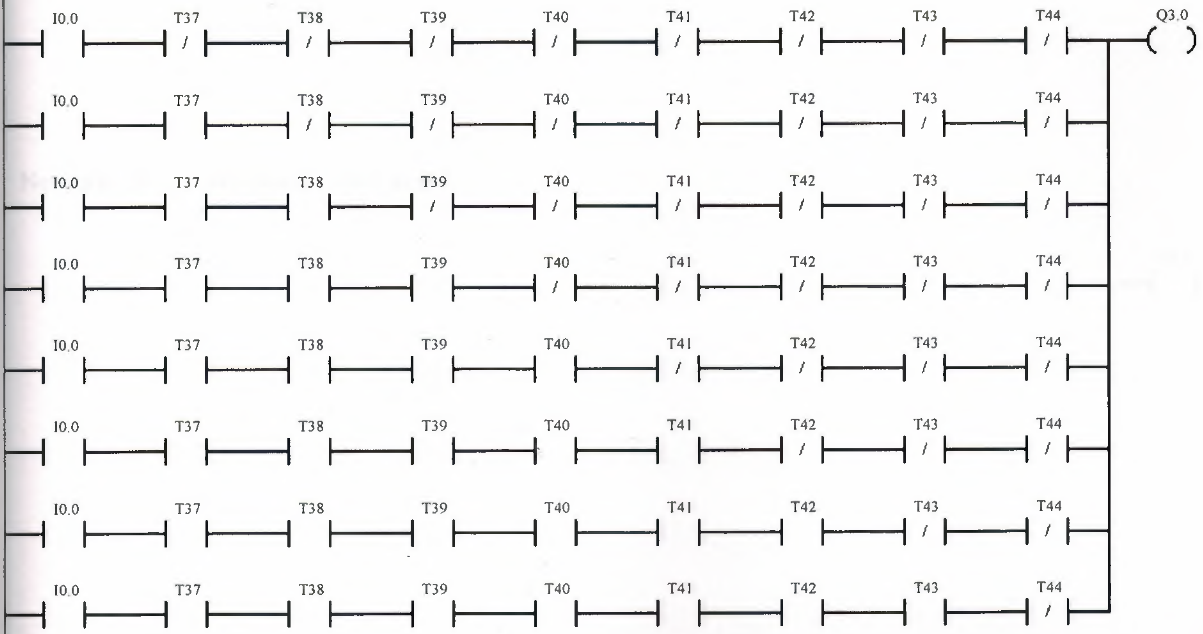
Network 22 Crosswalk 6 is yellow



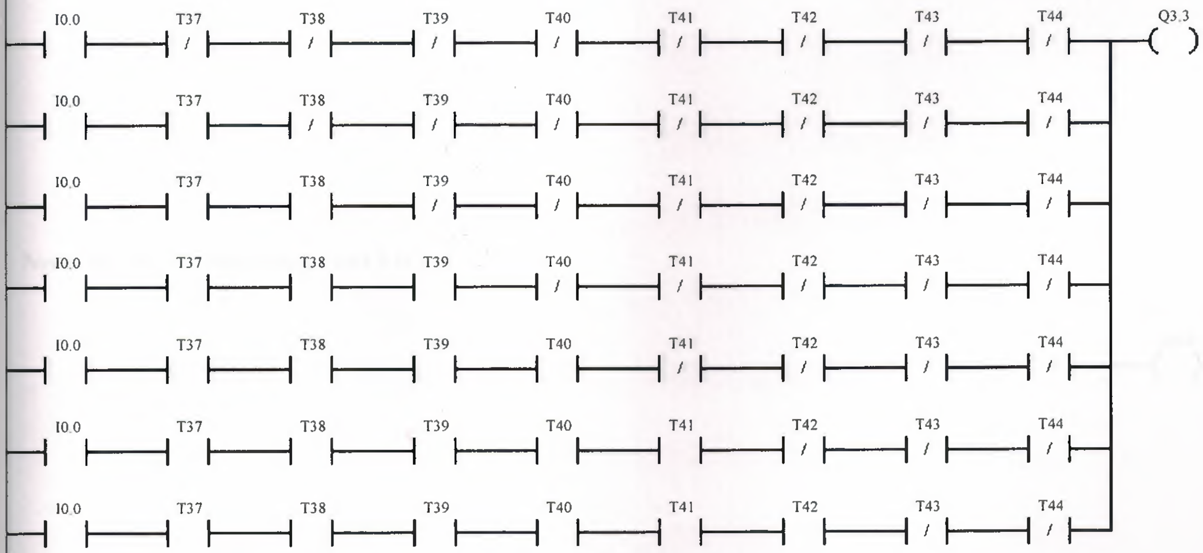
Network 23 Crosswalk 6 is green



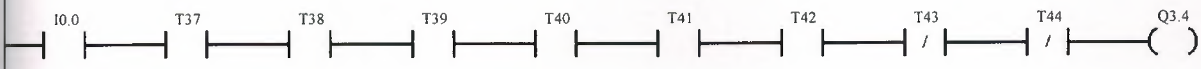
Network 24   Secondary road 1 is red



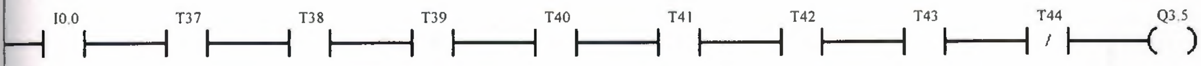
Network 25   Secondary road 2 is red



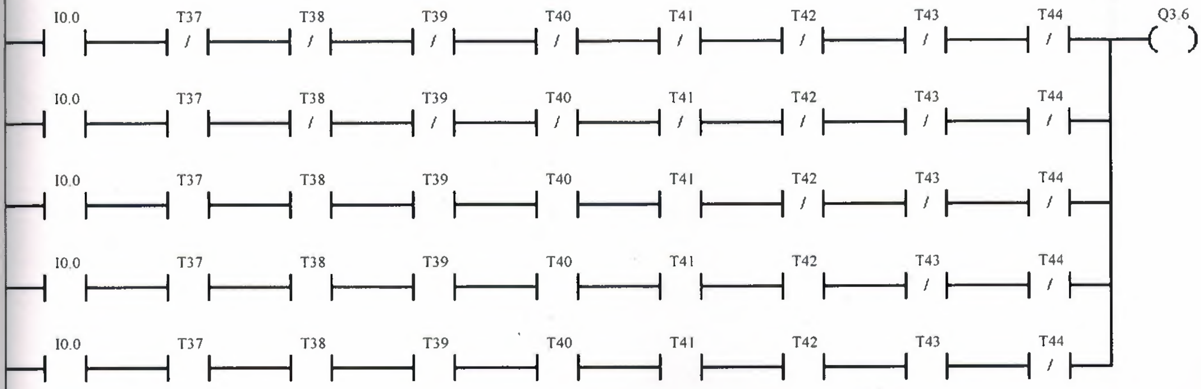
Network 26   Secondary road 2 is yellow



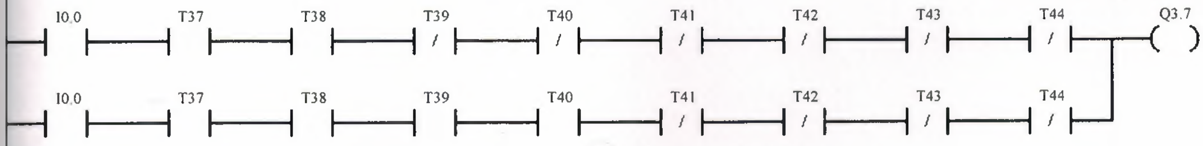
Network 27 Secondary road 2 is green



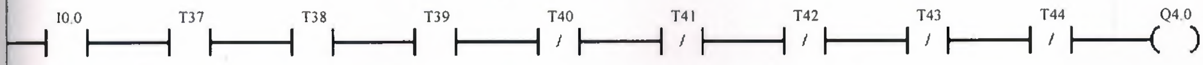
Network 28 Secondary road 3 is red



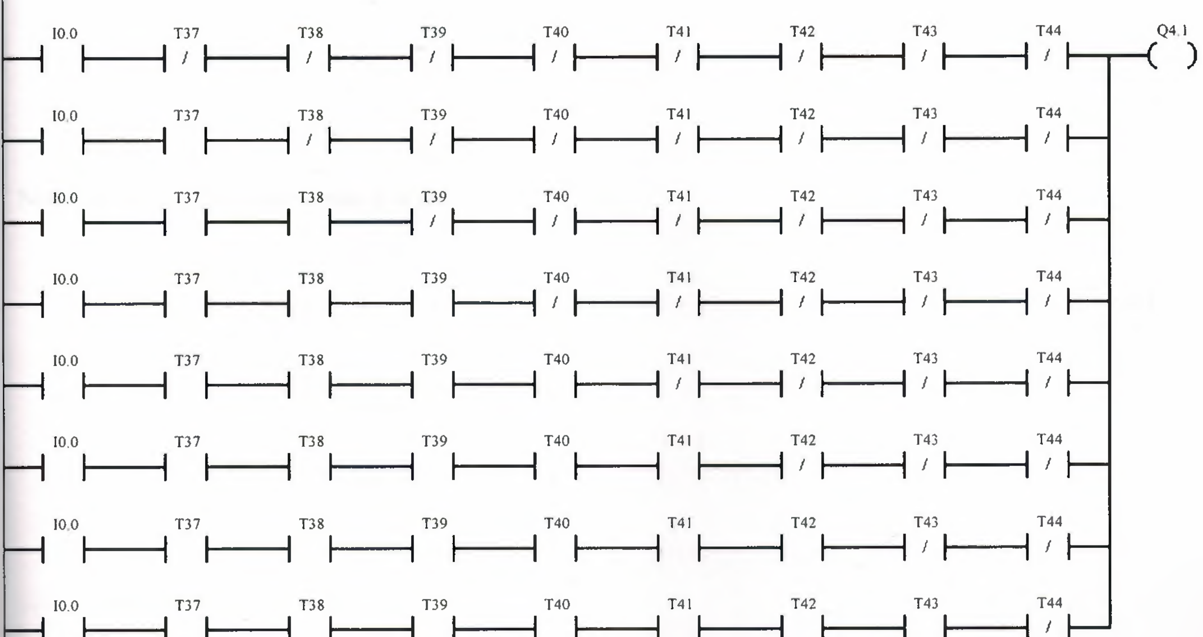
Network 29 Secondary road 3 is yellow



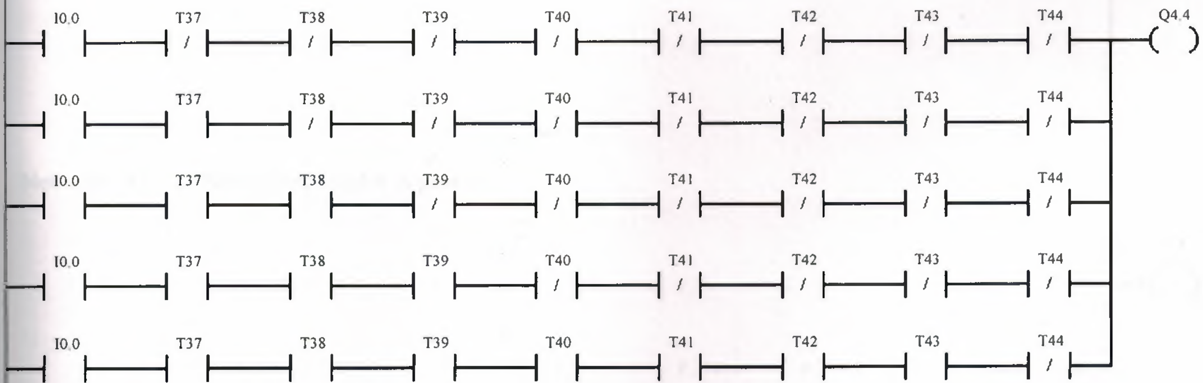
Network 30 Secondary road 3 is red



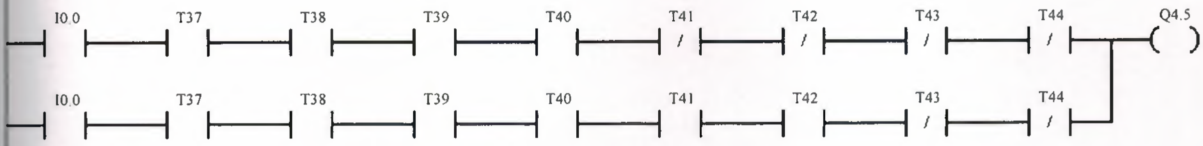
Network 31 Secondary road 4 is red


I0.0 T37 / T38 / T39 / T40 / T41 / T42 / T43 / T44 / Q4.1
I0.0 T37 T38 / T39 / T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 / T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 T44 /

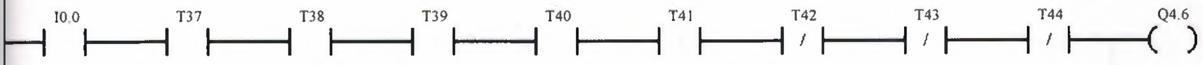
Network 32 Secondary road 5 is red


I0.0 T37 / T38 / T39 / T40 / T41 / T42 / T43 / T44 / Q4.4
I0.0 T37 T38 / T39 / T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 / T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 / T41 / T42 / T43 / T44 /
I0.0 T37 T38 T39 T40 T41 T42 T43 T44 /

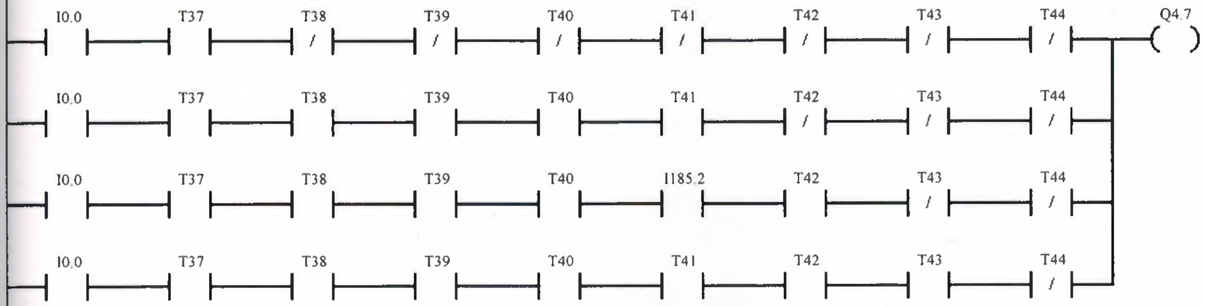
Network 33 Secondary road 5 is yellow


I0.0 T37 T38 T39 T40 T41 / T42 / T43 / T44 / Q4.5
I0.0 T37 T38 T39 T40 T41 T42 T43 / T44 /

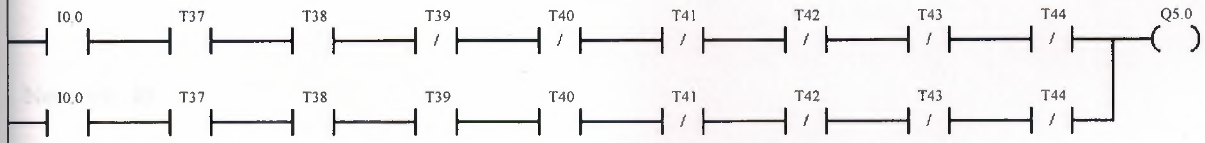
Network 34 Secondary road 5 is green



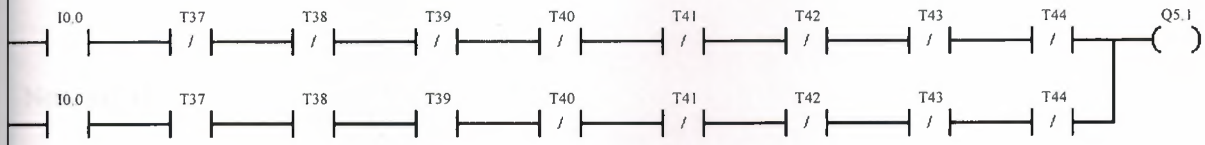
Network 35 Secondary road 6 is red



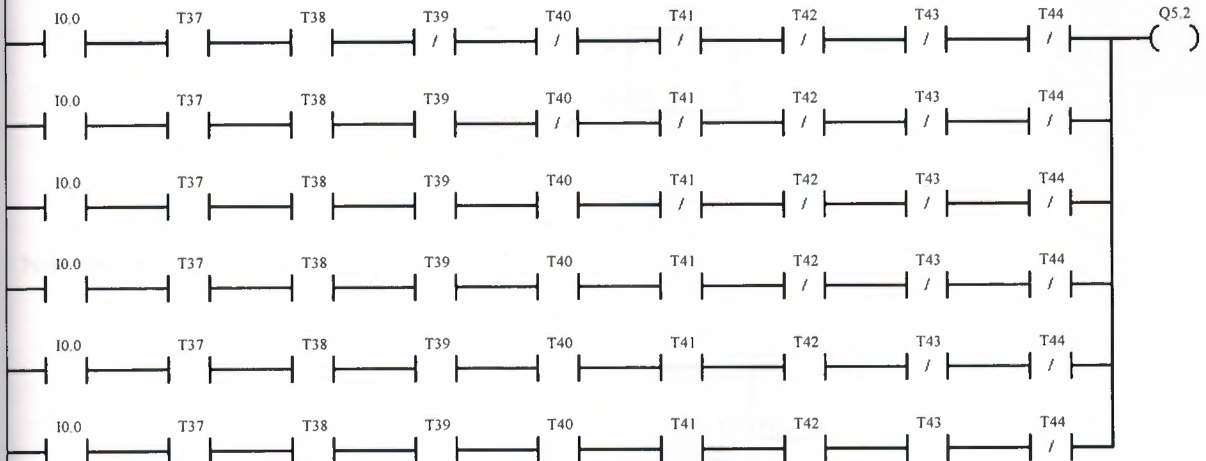
Network 36 Secondary road 6 is yellow



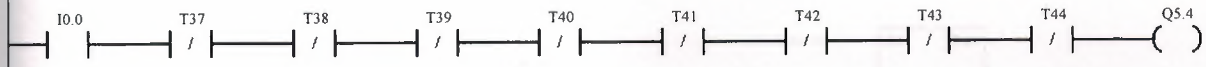
Network 37 Secondary road 6 is green



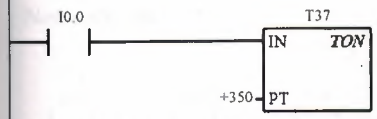
Network 38 Secondary road 7 is red



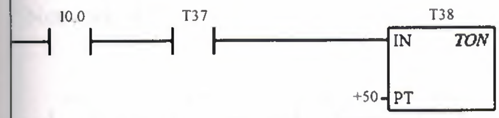
Network 39 Secondary road 7 is green



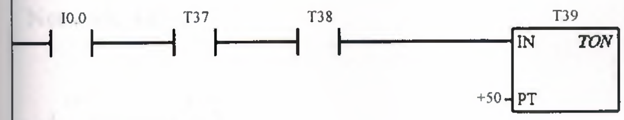
Network 40



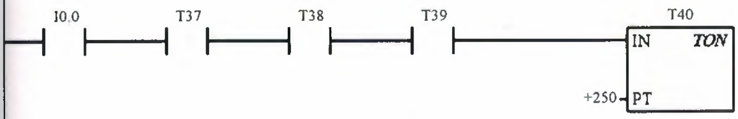
Network 41



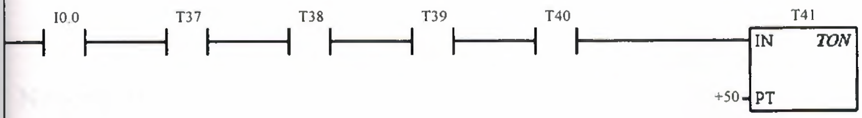
Network 42



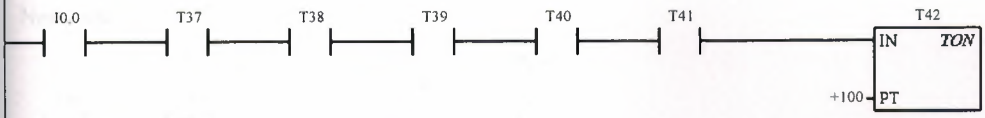
Network 43



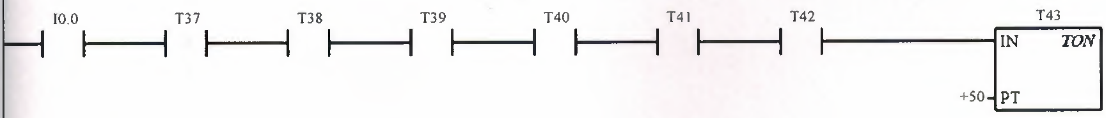
Network 44



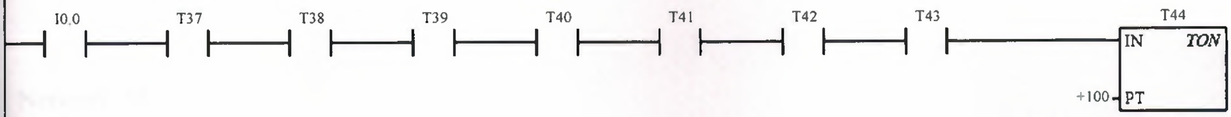
Network 45



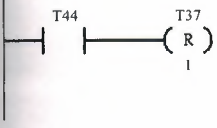
Network 46

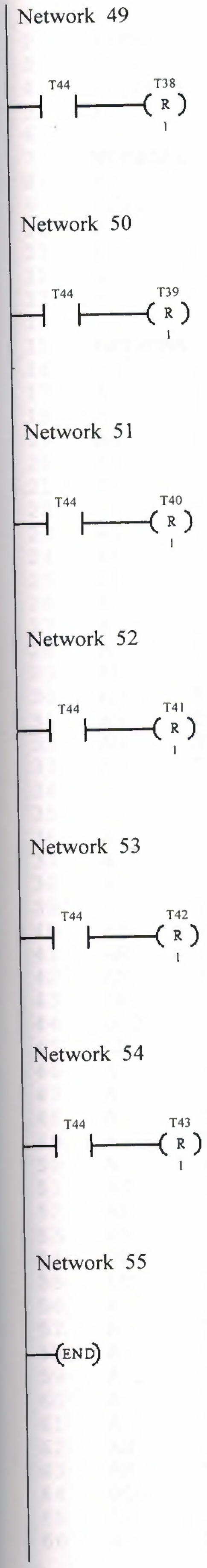


Network 47



Network 48




Network 49
T44
T38
R
1
Network 50
T44
T39
R
1
Network 51
T44
T40
R
1
Network 52
T44
T41
R
1
Network 53
T44
T42
R
1
Network 54
T44
T43
R
1
Network 55
END

```
//
//PROGRAM TITLE COMMENTS
//
//Press F1 for help and example program
//

NETWORK 1    //Flasher
//
//NETWORK COMMENTS
//
LD      I0.0
A       SM0.5
=       Q5.5

NETWORK 2    //Mainroad 1 is red
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
```

```
A      T38
A      T39
A      T40
A      T41
A      T42
A      T43
AN     T44
OLD
=      Q0.0

NETWORK 3    //Mainroad 1 is yellow
LD     I0.0
A      T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
=      Q0.1

NETWORK 4    //Mainroad 1 is green
LD     I0.0
AN     T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
=      Q0.2

NETWORK 5    //Mainroad 2 is red
LD     I0.0
A      T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
A      T38
A      T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
```

```
A      T38
A      T39
A      T40
A      T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
A      T43
AN     T44
OLD
=      Q0.3

NETWORK 6   //Mainroad 2 is yellow
LD     I0.0
A      T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
=      Q0.4

NETWORK 7   //Mainroad 2 is green
LD     I0.0
AN     T37
AN     T38
AN     T39
AN     T40
A      T41
AN     T42
AN     T43
AN     T44
=      Q0.5

NETWORK 8   //Crosswalk 1 is red
LD     I0.0
A      T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
```

```
A      T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
A      T43
AN     T44
OLD
=      Q0.6

NETWORK 9   //Crosswalk 1 is yellow
LD     I0.0
A      T37
A      T38
A      T39
A      T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
AN     T43
AN     T44
OLD
=      Q0.7

NETWORK 10  //Crosswalk 1 is green
LD     I0.0
AN     T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
AN     T38
AN     T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
=       Q1.0

NETWORK 11   //Crosswalk 2 is red
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q1.1

NETWORK 12   //Crosswalk 2 is green
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
=       Q1.3

NETWORK 13   //Crosswalk 3 is red
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
=       Q1.4

NETWORK 14   //Crosswalk 3 is yellow
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
=       Q1.5

NETWORK 15   //Crosswalk 3 is green
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q1.6

NETWORK 16   //Crosswalk 4 is red
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
```

```
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40


A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q1.7

NETWORK 17   //Crosswalk 4 is green
LD      I0.0
```

```
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
=       Q2.1

NETWORK 18   //Crosswalk 5 is red
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
=       Q2.2

NETWORK 19   //Crosstalk 5 is yellow
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
=       Q2.3

NETWORK 20   //Crosswalk 5 is green
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
```

```
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q2.4

NETWORK 21   //Crosswalk 6 is red
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0

A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
=       Q2.5

NETWORK 22   //Crosswalk 6 is yellow
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
=       Q2.6

NETWORK 23   //Crosswalk 6 is green
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
=       Q2.7

NETWORK 24   //Secondary road 1 is red
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
```

```
727  AN     T41
728  AN     T42
729  AN     T43
730  AN     T44
731  OLD
732  LD     I0.0
733  A      T37
734  A      T38
735  AN     T39
736  AN     T40
737  AN     T41
738  AN     T42
739  AN     T43
740  AN     T44
741  OLD
742  LD     I0.0
743  A      T37
744  A      T38
745  A      T39
746  AN     T40
747  AN     T41
748  AN     T42
749  AN     T43
750  AN     T44
751  OLD
752  LD     I0.0
753  A      T37
754  A      T38
755  A      T39
756  A      T40
757  AN     T41
758  AN     T42
759  AN     T43
760  AN     T44
761  OLD
762  LD     I0.0
763  A      T37
764  A      T38
765  A      T39
766  A      T40
767  A      T41
768  AN     T42
769  AN     T43
770  AN     T44
771  OLD
772  LD     I0.0
773  A      T37
774  A      T38
775  A      T39
776  A      T40
777  A      T41
778  A      T42
779  AN     T43
780  AN     T44
781  OLD
782  LD     I0.0
783  A      T37
784  A      T38
785  A      T39
786  A      T40
787  A      T41
788  A      T42
789  A      T43
790  AN     T44
791  OLD
792  =      Q3.0
```

```

NETWORK 25  //Secondary road 2 is red
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
```

```
A      T41
A      T42
AN     T43
AN     T44
OLD
=      Q3.3

NETWORK 26   //Secondary road 2 is yellow
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
AN     T43
AN     T44
=      Q3.4

NETWORK 27   //Secondary road 2 is green
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
A      T43
AN     T44
=      Q3.5

NETWORK 28   //Secondary road 3 is red
LD     I0.0
AN     T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
```

```
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q3.6

NETWORK 29   //Secondary road 3 is yellow
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
=       Q3.7

NETWORK 30   //Secondary road 3 is red
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
=       Q4.0

NETWORK 31   //Secondary road 4 is red
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
```

```
991  AN     T41
992  AN     T42
993  AN     T43
994  AN     T44
995  OLD
996  LD     I0.0
997  A      T37
998  A      T38
999  AN     T39
1000 AN     T40
1001 AN     T41
1002 AN     T42
1003 AN     T43
1004 AN     T44
1005 OLD
1006 LD     I0.0
1007 A      T37
1008 A      T38
1009 A      T39
1010 AN     T40
1011 AN     T41
1012 AN     T42
1013 AN     T43
1014 AN     T44
1015 OLD
1016 LD     I0.0
1017 A      T37
1018 A      T38
1019 A      T39
1020 A      T40
1021 AN     T41
1022 AN     T42
1023 AN     T43
1024 AN     T44
1025 OLD
1026 LD     I0.0
1027 A      T37
1028 A      T38
1029 A      T39
1030 A      T40
1031 A      T41
1032 AN     T42
1033 AN     T43
1034 AN     T44
1035 OLD
1036 LD     I0.0
1037 A      T37
1038 A      T38
1039 A      T39
1040 A      T40
1041 A      T41
1042 A      T42
1043 AN     T43
1044 AN     T44
1045 OLD
1046 LD     I0.0
1047 A      T37
1048 A      T38
1049 A      T39
1050 A      T40
1051 A      T41
1052 A      T42
1053 A      T43
1054 AN     T44
1055 OLD
1056 =      Q4.1
```

```

NETWORK 32  //Secondary road 5 is red
LD     I0.0
AN     T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
AN     T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
AN     T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
AN     T40
AN     T41
AN     T42
AN     T43
AN     T44
OLD
LD     I0.0
A      T37
A      T38
A      T39
A      T40
A      T41
A      T42
A      T43
AN     T44
OLD
=      Q4.4

NETWORK 33  //Secondary road 5 is yellow
LD     I0.0
A      T37
A      T38
A      T39
A      T40
AN     T41
AN     T42
AN     T43
AN     T44
LD     I0.0
A      T37
A      T38
```

```
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
=       Q4.5

NETWORK 34   //Secondary road 5 is green
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
=       Q4.6

NETWORK 35   //Secondary road 6 is red
LD      I0.0
A       T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       I185.2
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q4.7

NETWORK 36   //Secondary road 6 is yellow
LD      I0.0
A       T37
```

```
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
A       T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
=       Q5.0

NETWORK 37   //Secondary road 6 is green
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
=       Q5.1

NETWORK 38   //Secondary road 7 is red
LD      I0.0
A       T37
A       T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
LD      I0.0
A       T37
A       T38
A       T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
```

```
AN      T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
AN      T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
AN      T43
AN      T44
OLD
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
AN      T44
OLD
=       Q5.2

NETWORK 39   //Secondary road 7 is green
LD      I0.0
AN      T37
AN      T38
AN      T39
AN      T40
AN      T41
AN      T42
AN      T43
AN      T44
=       Q5.4

NETWORK 40
LD      I0.0
TON     T37, +350

NETWORK 41
LD      I0.0
A       T37
TON     T38, +50

NETWORK 42
LD      I0.0
A       T37
A       T38
TON     T39, +50

NETWORK 43
LD      I0.0
```

```
A       T37
A       T38
A       T39
TON     T40, +250

NETWORK 44
LD      I0.0
A       T37
A       T38
A       T39
A       T40
TON     T41, +50

NETWORK 45
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
TON     T42, +100

NETWORK 46
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
TON     T43, +50

NETWORK 47
LD      I0.0
A       T37
A       T38
A       T39
A       T40
A       T41
A       T42
A       T43
TON     T44, +100

NETWORK 48
LD      T44
R       T37, 1

NETWORK 49
LD      T44
R       T38, 1

NETWORK 50
LD      T44
R       T39, 1

NETWORK 51
LD      T44
R       T40, 1

NETWORK 52
LD      T44
R       T41, 1

NETWORK 53
LD      T44
R       T42, 1
```

```
NETWORK 54
LD      T44
R       T43, 1

NETWORK 55
MEND
```

CONCLUSION

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

With the information observed from our lecturer and our researchers for this topic PLC, is a convenient tool with a wide rage of useful ways to be used. Such examples can be mentioned several machines can be used at the same time, easy adjustments from the PLC program can be meet within a few minutes by the keyboard, installed 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. Communication programs of PLCs 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 (1999). Programmable Logic Controllers.
Erdoğan Teközgen İstanbul, (1998). PLC ve Uygulamaları
HAKER Soğuk Döküm San. Tic. Lti. Şti. Tansel Sarıçam İZMİR
World Wide Web: www.siemens.com
EGESİM Siemens Ana Bayii. 1204 Sok. No:41/1-l Bulanalp 2 İş Merkezi Yenişehir İZMİR.