SMIT DIPLOMA ENGG. WING ANKUSHPUR



DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

LAB MANUAL

Year & Semester: 3RD Year, VI Semester Subject Code/Name: PR-4, PLC & AUTOMATION LAB

VISION OF THE DEPARTMENT-

To be a center of excellence in the field of E&TC Engg by providing quality technical education as well as inculcating entrepreneurial skill, self-learning attitude and human values.

MISSION OF THE DEPARTMENT -

- 1) To create an excellent infrastructure and teaching-learning environment for making the student acquire the knowledge needed.
- 2) To inculcate self-learning attitude, entrepreneurial skill and human values.
- 3) To impart knowledge required for recent & advanced Engg.

PROGRAM EDUCATIONAL OBJECTIVE-

- 1) To apply knowledge of E&TC Engg. and principles of basic science & mathematics for analyzing the circuit of different electronics devices.
- 2) To compete for employment both in public and private sector.
- 3) To pursue higher study in E&TC as well as any other related branch for carrier enhancement.
- 4) To establish own assembly or service section with innovative, entrepreneurial ideas.
- 5) To work with team spirit and social ethics.

PROGRAM SPECIFIC OUTCOME-

- 1) Use techniques and skills to design, analyze, synthesize and simulate electronics components and systems.
- 2) Architect, partition and select appropriate technology for implementation of a specified communication system.

PRECAUTIONS:

- 1. Properly connect the microprocessor and microcontroller kit with power supplyterminals.
- 2. Switch on the power supply after checking connections
- 3. Handle the Trainer kit carefully.
- 4. If you smell anything burning, immediately disconnect the power
- 5. Keep your work area dry

COURSE OUTCOME

After the completion of the course the students will be able to-**CO1**-Design PLC ladder logic program.

CO2-Interface the PLC trainer with the PC and application trainer kit with the PLC trainer.

CO3-Simulate the PLC programming and observe the output.

CO4-Install network card, wireless node, internal modem and switches.

CO5-Write and explain the procedure of the experiments conducted.

LIST OF EXPERIMENTS

S.NO.	NAME OF EXPERIMENTS	PAGE NO.
1	Introduction to PLC trainer & its installation with PC	
2	Write and implement a simple ladder logic program using digital inputs and outputs for PLC.	
3	 Write and implementation of simple ladder logic program using timer 1) On delay timer 2) Off delay timer 3) Retentive timer 	
4	 Write and implementation of simple ladder logic program using counter. 1) UP counter 2) Down counter 	
5	Write program on MOVE, control statement, math function, data manipulation technique on PLC.	
6	To study about conveyor control system using PLC	
7	Write and implement ladder logic program to on-off the DC motor using PLC	
8	To study the traffic light controller system by using PLC	
9	Interface SCADA with PLC and associate tags with memory and I/ O and operate the PLC inputs through the switch symbol from the computer screen and view the status of the outputs using lamp and motor graphics symbols in the screen.	

Programming And Logic Control (PLC)

AIM OF THE EXPERIMENT:

Introduction /Familiarization PLC trainer & its instruction with PC.

APPARATUS REQUIRED:

Sl. no.	Name of Equipment
1	Lab Programmable Logic
	Controller (PLC) kit
2	Personal Computer

THEORY:

What is PLC?

A programmable logic controller (PLC) is an industrial computer control system that continuously monitors the state of input device and makes decisions based upon a custom program to control the state of output devices.

What is inside a PLC?

- The central processing unit, the CPU, contains an internal program that tells the PLC how to perform the following functions.
- The CPU combines a microprocessor, an integrated power supply, input and output circuits, built in PROFINET, high speed motion control I/O, and on-board analog input in a compact housing to create a powerful controller.
- After you download your program, the CPU contains the logic required to monitor and control the devices in your application.
- The CPU monitors the input and changes the outputs according to the logic of your user program, which can include Boolean logic, counting, timing, complex math operations and communication with other intelligent devices.
- To communicate with a programming device, the CPU provides a built in PROFINET port.
- With the PROFINET network, the CPU can communicate with HMI panels or another CPU.
- To provide security for your application, every S7-1200 CPU provides password protection that allows you to configure access to the CPU functions.

- The CPU supports only a preformatted SIMATIC memory card. To insert a memory card, open the top CPU door and insert the memory card in the slot.
- > Use the optional SIMATIC memory card either as a program card or as a transfer card.
- Digital inputs in the controller are 14 and voltage is 24. Digital outputs in the controller are 10 and voltage is 24.
- Back panel is available to connect extra I/O modules and communication modules.
- > Analog input in the controller is two.
- ▶ Input power supply to the controller is 120/240 VAC.
- > Three communication modules and eight I/O expansion modules can be used.
- > Modbus communication board, output board and analog input board can be used.

PLC Languages:

The function of all programming languages is to allow the user to communicate with the programmable controller via a programming device. They all convey to the system, by means of instructions, a basic control plan.

- The most common types of languages encountered in programmable controller system design is ;
 - a) Ladder Diagram (LD)
 - b) Function Block Diagram (FBD)

LADDER DIAGRAM (LD)

Traditional ladder logic is graphical programming language. Initially programmed with simple contacts that simulated the opening and closing relays, counters, timers, shift registers etc.

FUNCTION BLOCK DIAGRAM (FBD)

Useful for expressing the interconnection of control system algorithms and logic.

HARDWARE/SOFTWARE REQUIREMENT:

Processor type - Intel Pentium i3, 2.5 GHZ or similar

RAM - 4GB

Available hard disk space - 10 GB on system drive C:\

Operating systems- windows XP professional SP3, windows 2003 server R2 SP2, windows 7

(professional, enterprise, ultimate) SP1, windows 10 pro

Graphics card - 32MB RAM 24-bit color depth

Screen resolution - 1024 x 768

Network - 20 M bit/s Ethernet or faster

Optical drive- DVD-ROM

PROCEDURE: (PLC SETUP)

- 1. First select **TIA portal 14.0** and double click on it.
- 2. Double click on "**Create new project**" then select the "**Project name**" and select the location path to save project and then click "**Create**".
- 3. Double click on "Configure a device".
- 4. Click "Add new device". After device is added click "Controller" and then click on "SIMATIC S71200".
- 5. Click on "CPU" and "CPU 1214 DC/DC/DC", then select the required "MLFB" number and click "Add".
- Now select the signal module AQ1x12 BIT and Add to the CPU and save the project by pressing CTRL+S key.
- The select respective expansion module 3 no's of DI 16/DQ 16x24VDC, 1 no. of DI 8/DQ8 x 24 VDC and 1 no. of AI 4 x 13 BIT/AQ 2 x 14 BIT from the hardware catalogue.
- 8. Double click on the "**RJ 45**" symbol in the controller, now "**Properties**" of the controller will be open and Ethernet configuration also open, if it is not opened just click on the Ethernet address and change IP address and change IP address as required then save the project.
- Then go to "Downloading the program" setting for click on the "Compile" icon and then click on the "Download" icon.
- 10. Then automatically the "Extended to download device" window open. Select the "PN/IE" into the type of the "PG/PC" interface and also select the "Show all compatible devices". Finally click the "Start search" option and then click the "Load" button.
- 11. In software synchronization before loading to a device window in opens and then clicks the **"Continue without synchronization**" option below.

12. "**Stop modules**", "**Stop all**" option and then "**Load**" option and then load option and also goes to "**Load results**" window tick the "Start all" and click the "**Finish**" option.

PROGRAMMING METHOD IN PLC:

- Click "PLC-1" in the project tree then click "Program block" and click "main OB1". Now OB1 is created (object block1).
- 2. Now select the network and double click on "**normally open**" icon (⊣⊢).
- Now normally open is added. Now add the "output coil" (- () -) in the network. Then double click on the both NO contact and output coil to enter their addresses simultaneously.
- 4. After it click on the "**Compile**" icon then "**Download to device**" icon.
- 5. For make an online process goes to click the "**Monitoring on/off**" icon. Then the window shows the online mode of the PLC.

RESULT:

We studied introduction of PLC, its installation with PC, hardware components, building various blocks and determine no. of digital inputs/outputs & analog inputs/outputs.

Write and implement a simple ladder logic program to study and verify and gate using digital inputs and outputs for PLC.

AND-GATE OPERATION

AIM OF THE EXPERIMENT:

Test the truth table of AND gate using PLC software.

APPARATUS REQUIRED:

Sl. no.	Apparatus Name
1.	PLC trainer kit
2.	Personal computer installed with PLC software
3.	Ethernet cable
4.	Patch chords

THEORY:

- 1) In AND_GATE operation is used to make the multiple operations of 2 inputs. Now using A&B are 2 inputs and C is the output.
- 2) Now generating the following formula to create the AND gate operation is given below.

A.B = C

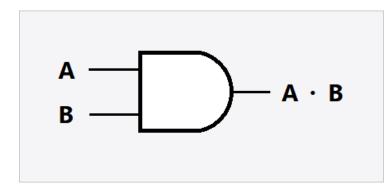
Programming:-



Truth table & Symbol of AND gate:

Α	В	С
0	0	0
0	1	0

0-LOW	; 1-HIGH	[
1	1	1
1	0	0



PROCEDURE:

- Click "PLC-1" in the project tree then click "Program block" and click "Main OB1". Now OB1 is created (object block1).
- 2. Now write an **AND_GATE** operation for make a 2 inputs like A and B. It is created by using two normally open contacts. And their addresses are **A(I0.0)** and **B(I0.1)**.
- 3. And the only one output can be used in this program that is C and their address is **(Q0.0)**.
- 4. After saved the program go to online mode to click **GO Online** icon directly.
- 5. After click the online mode goes to select the **Monitoring ON/OFF** icon.
- 6. Finally the two inputs [(10.0),(10.1)] are goes to HIGH the output[Q0.0] will goes to HIGH.

CONCLUSION:

Thus the AND_GATE operation was studied successfully through PLC Software.

OR-GATE OPERATION

AIM OF THE EXPERIMENT:

Test the truth table of OR gate using PLC software.

APPARATUS REQUIRED:

Sl. no.	Apparatus Name
1.	PLC trainer kit
2.	Personal computer installed with PLC
	software
3.	Ethernet cable
4.	Patch chords

THEORY:

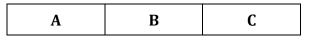
- In OR_GATE operation is used to make the addition operation of 2 inputs. Now using A&B are 2 inputs and C is the output.
- > Now generating the following formula to create the AND_GATE operation is given below.

$\mathbf{A} + \mathbf{B} = \mathbf{C}$

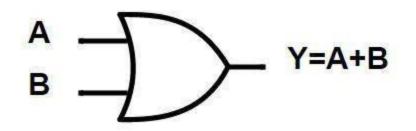
Programming:-



TRUTH TABLE & SYMBOL OF OR_GATE:



0-LOW	/; 1	-HIGH
1	1	1
1	0	1
0	1	1
0	0	0



PROCEDURE:

- **1.** Click PLC-1 in the project tree then click **program block** and then click **main OB1**.
- 2. Now the (object block 1) **OB1** is created.
- 3. Now write an **OR_GATE** operation for make a 2 inputs like A and B.
- 4. It is created by using two normally open contacts parallel connection. And their addresses are A (**I0.0**) and B (**I0.1**).
- 5. And the only one output can be used in this program that is C and their address is **(Q0.0)**.
- 6. Then save the program to press (CTRL+S) function it will be saved.
- 7. After saved the program go to online mode to click **GO Online** icon directly.
- 8. After click the online mode goes to select the **Monitoring ON/OFF** icon.
- Finally the inputs [(I0.0), (I0.1)] are goes to HIGH the output [Q0.0] will goes to HIGH. It can be represented in Green color indication.

CONCLUSION:

Thus the **OR_GATE** operation was studied successfully through **PLC Software**.

<u>EX_OR-GATE</u>

AIM OF THE EXPERIMENT:

Test the truth table of EX- OR gate using PLC software.

APPARATUS REQUIRED:

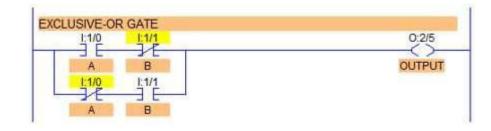
Sl. no.	Apparatus Name
1.	PLC trainer kit
2.	Personal computer installed with PLC software
3.	Ethernet cable
4.	Patch chords

THEORY:

- In EX-OR_GATE operation is used to make the operation of inequality functions of inputs. Now using A,B and C is the output.
- Now generating the following formula to create the EX-OR_GATE operation is given below.

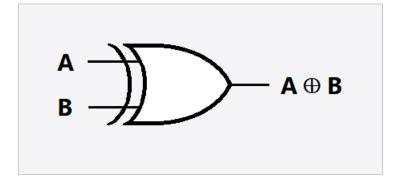
 $\overline{AB} + A.\overline{B} = C$

PROGRAMMING:-



TRUTH TABLE & SYMBOL OF EX-OR_GATE:

Α	В	С
0	0	0
0	1	1
1	0	1
1	1	0
0-LOW	/ ; 1	-HIGH



PROCEDURE:

- 1. Click PLC-1 in the project tree then click **program block** and then click **main OB1**.
- 2. Now the (object block 1) **OB1** is created.
- 3. Now write an **EX-OR_GATE** operation for make a 2 inputs like A and B.
- 4. Then save the program and go to **download** into PLC.
- 5. After saved the program go to online mode to click **GO Online** icon directly.
- 6. Finally **monitoring** all the values of the inputs and outputs.
- 7. All the inputs are **LOW** at that time the output is **LOW**. The inequality functions of Inputs are **HIGH** the Output is **HIGH**.
- 8. Thus the all input values are **HIGH** at that time the Output is **HIGH**.

CONCLUSION:

Thus the EX-OR_GATE operation was studied successfully through PLC Software.

EX NOR-GATE

AIM OF EXPERIMENT:

Test the truth table of EX- OR gate using PLC software.

APPARATUS REQUIRED:

Sl. no.	Apparatus Name
1.	PLC trainer kit
2.	Personal computer installed with PLC software
3.	Ethernet cable
4.	Patch chords

THEORY:

 \triangleright

In **EXNOR_GATE** operation is used to make the inverse operation of the

EXNOR_GATE values. Now using A,B and C is the output.

Now generating the following formula to create the EXNOR_GATE operation is given below.

$$\bar{A}B + A \cdot B = C$$

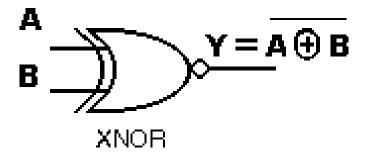
Programming:-

1:1/0	1/1	0.2/6
1		<u>()</u>
A	в	OUTPUT
11/0	11/1	

TRUTH TABLE & SYMBOL OF EXNOR_GATE:

Α	В	С
0	0	1
0	1	0
1	0	0
1	1	1

0-LOW 1-HIGH



PROCEDURE:

- **1.** Click **PLC-1** in the project tree then click **program block** and then click **main OB1**.
- 2. Now the (object block 1) **OB1** is created.
- 3. Now write an **EXNOR_GATE** operation for make a 2 inputs like A and B.
- 4. Then save the program and go to **download** into PLC.

- 5. After saved the program go to online mode to click **GO Online** icon directly.
- 6. Finally **monitoring** all the values of the inputs and outputs.
- 7. All the inputs are **LOW** at that time the output is **HIGH**. The inequality functions of Inputs are **HIGH** the Output is **HIGH**.
- 8. Thus the all input values are **HIGH** at that time the Output is **HIGH**.

CONCLUSION:

Thus the **EX-NOR_GATE** operation was studied successfully through **PLC Software**.

Write and implementation of simple ladder logic program using timer

ON DELAY TIMER

AIM OF THE EXPERIMENT:

To study the ON Delay Timer operation using PLC Software.

APPARATUS REQUIRED:

- PLC Trainer kit
- Personal Computer Installed with PLC Software
- Ethernet cable
- Patch chords

THEORY:

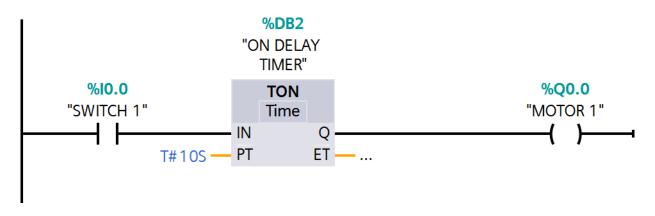
- ON Delay Timer is used to make the Delay operation in processes.
- It can control the inputs and outputs of the field instruments.

PROCEDURE:

- 1. Click PLC-1 in the project tree "1" then click program block "2" and click main OB1 "3" as shown in the below image.
- 2. Now the (Object Block1) OB1 is created.
- 3. Now select the On Delay Timer goes to instructions tab and click the Basic Instructions and select the Timer operations tab and then double click the TON block.
- 4. Now the call options block is displayed. Then change the data block name. And click the automatic option finally select the ok button.
- 5. Now the Timer block is created in network is shown below the diagram. And also that one input and one output coil is created in same network for testing the timer block.
- 6. The input can be connected with IN of the Timer block. Then output coil connected with Q of the Timer block.
- 7. Then given the addresses for input and outputs. And finally given the preset time for the Timer goes to double click the PT in timer block and write the values in sec.
- 8. Then save the program and download after that goes to online mode. And monitoring the program can be displayed in following images.
- 9. Then given the addresses for input and outputs. And finally given the preset time for the Timer goes to double click the PT in timer block and write the values in sec.
- 10. Then save the program and download after that goes to online mode and monitor the program.

11. When the status of the SWITCH 1(I0.0) changes from 0 to 1 the timer instruction will be executed and it will activate the MOTOR 1(Q0.0) after 10s delay.

LADDER LOGIC PROGRAM:-



CONCLUSION:

Thus the ladder logic program of ON Timer was written and implemented successfully using PLC Software.

<mark>OFF DELAY TIMER</mark>

AIM OF THE EXPERIMENT:

To study the OFF Delay Timer operation using PLC Software.

APPARATUS REQUIRED:

- PLC Trainer kit
- Personal Computer Installed with PLC Software
- Ethernet cable
- Patch chords

THEORY:

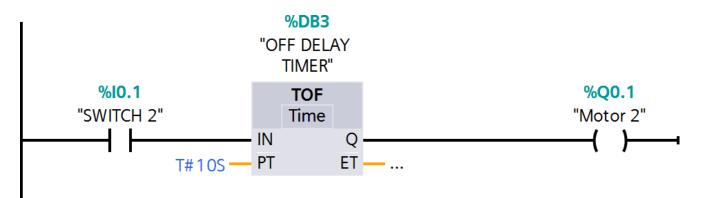
- OFF Delay Timer is used to make the Delay operation in processes.
- It can be control the inputs and outputs of the field instruments.

PROCEDURE:

- 1. Click PLC-1 in the project tree "1" then click program block "2" and click main OB1 "3" as shown in the below image
- 2. Now the below image shows (Object Block1) OB1 is created.
- 3. Now select the OFF Delay Timer goes to instructions tab and click the Basic Instructions and select the Timer operations tab and then double click the TOF block.
- 4. Now the call options block is displayed. Then change the data block name. And click the automatic option finally select the ok button.
- 5. Now the Timer block is created in network. And also that one input and one output coil is created in same network for testing the timer block.
- 6. The input can be connected with IN of the Timer block. Then output coil is connected with Q of the Timer block.
- 7. Then given the addresses for input and outputs. And finally given the preset time for the Timer goes to double click the PT in timer block and write the values in seconds.
- 8. Then save the program and download after that goes to online mode and monitor the program.
- 9. When the status of the SWITCH 2(I0.1) changes from 0 to 1 the timer instruction will be executed and it will activate the MOTOR 2(Q0.1) immediately.

10. When the SWITCH 2(I0.1) status changes back to 0 then programmed time (PT) will start and after time MOTOR 2(Q0.1) will be OFF.

LADDER LOGIC PROGRAM:



CONCLUSION:

Thus the ladder logic program of OFF Timer was written and implemented successfully using PLC Software.

RETENTIVE TIMER

AIM OF THE EXPERIMENT:

Write and implement a simple ladder logic program using Retentive timer.

APPARATUS REQUIRED:

- PLC Trainer kit
- Personal Computer Installed with PLC Software
- Ethernet cable
- Patch chords

THEORY:

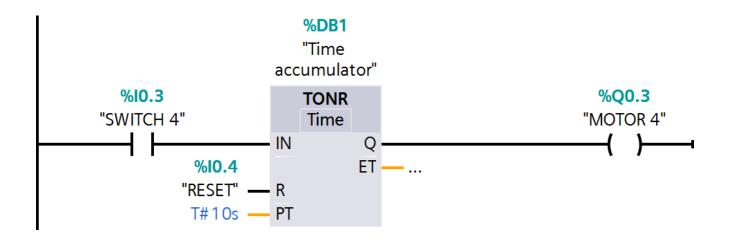
- Counts time base intervals when the instruction is true and retains the accumulated value when the instruction goes false or when power cycle occurs.
- The Retentive Timer instruction is a retentive instruction that begins to count time base intervals when rung conditions become true.
- The Retentive Timer instruction retains its accumulated value when any of the following occurs:
- Rung conditions become false.
- The processor loses power while battery backup is still maintained. And a fault occurs.

PROCEDURE:

- 1. Click PLC-1 in the project tree "1" then click program block "2" and click main OB1 "3" as shown in the below image
- 2. Now the below image shows (Object Block1) OB1 is created.
- 3. Now select the Accumulator timer goes to instructions tab and click the Basic Instructions and select the Timer operations tab and then double click the TONR block.
- 4. Now the call options block is displayed. Then change the data block name. And click the automatic option finally select the ok button.
- 5. Now the Timer block is created in network. And also that one input and one output coil is created in same network for testing the timer block.
- 6. The input can be connected with IN of the Timer block. Then output coil is connected with Q of the Timer block.
- 7. Then given the addresses for input and outputs. And finally given the preset time for the Timer goes to double click the PT in timer block and write the values in seconds.
- 8. Then save the program and download after that goes to online mode and monitor the program.

- 9. When the status of the SWITCH 4(I0.3) changes from 0 to 1 the timer instruction will be executed and MOTOR 4(Q0.3) will start after 10s.
- 10. The MOTOR 4(Q0.2) will remain ON, even when the input status changes back to 0. The Reset (I0.4) is necessary to reset the timer or accumulated time.

LADDER LOGIC PROGRAM-



CONCLUSION:

Thus the ladder logic program of Accumulator Timer was written and implemented successfully using PLC software.

Write and implementation of simple ladder logic program using counter

UP COUNTER

AIM OF THE EXPERIMENT:

To study about the UP Counter operation using PLC Software.

APPARATUS REQUIRED:

- PLC Trainer kit
- Personal Computer Installed with PLC Software
- Ethernet cable
- Patch chords

THEORY:

The counters are mainly used for counting the values from the field equipment. The UP counters can be used to counting the values in upward manner. It can be performing the incremental purpose.

PROCEDURE:

- 1. Click **PLC-1** in the project tree "1" then click **program block** "2" and click **main OB1** "3".
- 2. Now the below image shows (Object Block1) OB1 is created.
- 3. To select the UP Counter block in plc goes to **Instructions** Tab and click the **Basic Instructions** then select the **CTU** block for UP Counter operation.
- 4. Now the UP Counter call options block can be opened. Then change the **data block** name in that Name block. And then click the **ok** button.
- 5. The following details can be entered into the UP Counter block is given below.
- CU Make the counter Input
- Q- Counter Output
- R Reset input
- CV Count Value
- PV Preset value (Set Value)
- 6. Then save the program and goes to online mode and monitoring the counter values before and after enabling the counter block.

LADDER LOGIC PROGRAM:-

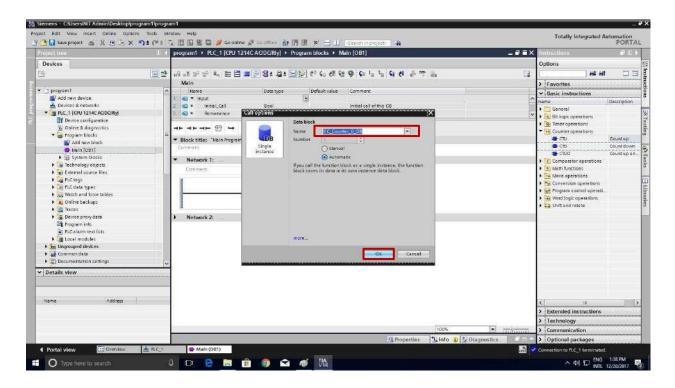
1.

nject trise 🛛 🖉 🐇	program1 + PLC_1 [CPU 1214C A	C/DC/RUy] + Program bloc	is + Main [OB1]		_ • • • ×		100
Devices						Options	
			21 12 19 44 14 14 14 14 14 14 14 14 14 14 14 14	a. ee [12]	3	641 641	r 🗆 🗉
2 (m) #	Main		0. 16 0 - 1 1 - 1 0 1	· · · · · ·	-		
a program 1	hame	Dots type Defaul	tvolue Comment			> Favorites	
Add new device		nois the neuro	Comment.		-	✓ Basic instructions	
Devices & networks	T - C + Initial Call	Bool	autial call of this Of			Hame	Descripti
* B R.C. 1 (OU 1214C ACIDORI)	3 da Remanence	Real	-Tipe, if immanent data are a	adable	100	 Genetal 	
Device configuration			inger tran		17.	* Hitloge aperations	-
S Online & diagnestics						40 -11-	Normally as
• D Program blocks =						HI -H- Normally open	n contact (Shift
Add neurblock	 Block title: "Akin Program Sweep 	(Cycle)"			_	30 -()-	Invertinuu
3- Main (061)	Comment					-0	Assignment Negete acc
) 🕞 Technology objects	 Network 1; 					10 -00	Resetautos
• The External source files	Comment					065	Set output
• 🚂 FLC tags	Comment.					HO SET BP	Setbitfeld
Eg PLC data types	1					HI JESET BP	Recetbitfe
 Watch and force tables 						81 1R	Settreset fi
A Online beckups						£° ≈	Resettiet fil
• 🔄 Traces						8 -17-	Scaniopera
By Device proxy data	the second second					31 -111-	Scan opera
29 Fregram arts	Network 2:					HT-0%-	Set operand
FLC alarm test lists	Caminent					第二 -04)-	Set operani
Lacal modules He Unarouped devices	T.					200.4 15	Scen BLO h
Common deta						ET IL TRG	Scan RO E
Documentation settings						把 10.786	Oetect posi
G Languages & resources						题 f_TRG	Detect negi
						 B Timer operations 	
Details view						• 🔄 Counter operations	
						E Comparator operations	
						Meth functions	
tame Address						Maye operations	
Neme Appress						¢ II	
						Extended instructions	
						> Technology	2011 P.
				100%	and house	> Communication	
			S Propert		100 1 2 3	> Optional packages	

2.

And control was an end of the set of the set	• (제비 🖄 🔟 🔟 🗐 🔛 🏓 Go online	972	All the second	here and a second statement of the second statement of				PORT
yect tree	II + program1 + PLC_1 (CPU 12	14C AC/DC/Riy] → Pro	igram blocks ► M	ain (OB1)		_ # # X	nstructions	2.1
Devices							Options	
	B 과 M M 가 가 가 트 클 Main	- - 2 : 2 : 2	19 C° 65 CE 68	। \$\$ 6 1 ₂ 1 ₂ 6 6 8 77	: Q.	8	Favorites 2	1 🗆 🛛
] program t	A Hame	Data type	Default value	Comment			Basic instructions	
Add new device	1 🔄 🕶 Input					1.1	iame	Description
📩 Devices & networks	2 📲 🔹 Initia (_Call	Bool		Initial call of this OB			F General	Description
TI PLC_1 [CPU 1214C ACIDORIA]	3 📶 • Remanence	Bool		-True, if remanent data are availab	ale	17.5	Bit logic operations	
Device configuration			1				Imer operation:	
9 Online & diagnostics	- + + + + + + + + + + + + + + + + + + +	2					Counter operations	
* 🗃 Program blocks	Block title: "Main Program :	(ween (Cycle)"					3 .CU	Countup
i Add new block	Somment	merb felinet					🔹 CID	Count down
🖶 Main (OB1)	Conservation of the second sec						3	Countupar
System blocks	 Network 1: 						Competator operations	
Technology objects	Comment						🗄 Math functions	
External source files External source files							Move operations	
PLC tags PLC date types							🖌 😼 Conversion operations	
Watch and force tables							🕨 🔂 Program control operati.	-
Gig Watch and torce tables Goline backups							Word logic operations	
Contrie backups Traces						2	Shift and rotate	
Taces Taces Taces Taces	Network 2:					_		
Program info	P Network z:							
b) PLC alarm text lists								
Coalam text lots Local modules								
Generation	and the second se							
Common data								
Documentation settings								
- And	×							
Details view								
isme Address								-
an inc							<]I	
							Extended instructions	
						2	Technology	
							Communication	
				Q Properties	🖞 Info 👔 🖞 Diagnostics	and the second se	Optional packages	
Portal view	RLC_1 (081)					🔛 💙 e	Connection to PLC 1 terminated	

3.



4. Now the UP Counter block is created in network is shown below the image.

	program1 + PLC_1 [CPU 1214C AC/DC/Rly] + Program blocks + M	Main [OB1] 🖬 i	i × Instructions	11
Devices			Options	
9	: @ @ 한 한 일 두 다 두 다 한 6 6 6 6 6	2 9 G 1 1 1 G (1 6 7 B	te te	
	Main		> Favorites	
program1	 Nome Detailt value 	Comment	V Basic instructions	-
💕 Add new device	I da ▼ input		A Name Detta	intici
🛔 Devices & networks	2 🚭 🖲 inital_Call Bool	initial call of this GB	F General	dine.
* ELC_1 (CPU 1214C ACIDC/RIy)	3 🚳 = Remanence Bool	=True, if remanent data are available	V Bit logic operation	
Device configuration		and a second	Gillimer operations	
& Online & diagnostics	<u>++++++</u> ₩ ++ -#		Counter operations	
💌 🕁 Program blocks			CTU Counter operations	
💕 Add nevi block	 Block title: "Main Program Sweep (Cycle)" 			nt dow
🖶 Mein [001]	Comment			
System blocks	* ONetwork 1:		CIUD Count Comparator operations	and a
Technology abjects			Comparator operations	
External source files	Connect			
PLC tags	T		Move operations	
Cei PLC data types	5081		Conversion operations	
Watch and force tables	"EC_Counter_ D DB"		Program control operati	
🕨 🙀 Online backups			 Word logic operations 	
• 🕞 Traces	CTU		Shift and rotate	
Device provy data				
Program info	tite a cv-			
PLC alarm text lists				
Local modules	STT: PN			
Engrouped devices				
Common data				
Decumentation settings	Network 2:			
Details view				
Name Address	-		- 22	
name (noores			<	111 I.
			> Extended instructions	
			> Technology	-
		100%	Communication	-
		Q Properties 💃 Info 👔 💆 Diagnostics	> Optional packages	-
Portal view	Amain (OB1)		The project program1 was saved succe.	1

5. Before Enabling the Counter Block

Melin Usano Data type Border Stretch Border Stretch Border Stretch Data type Border Stretch Border Stretch Border Stretch Border Stretch Border Stretch Data type Border St	Options	Description Count up Count down Count down
Melin Melin Import Data Syse Netwick value Data Syse Netwick Value Netwick Value	Favorites Said instructions Said instructions Ident Ide	Description Count up Count down Count down
Main Data type Data type<	Favorites Said instructions Said instructions Ident Ide	Description Count up Count down Count down
Interest & encrotions Interest & encrotions Interest & encrotions Interest & encrotions <	Basic Instructions Basic Instructions Insec Coloral Color	Count up Count down Count up an
Production Production Production Production Production Production Production Production Production Production Production	Alame Alame Central Central Central Central Central Central Central Control Contro Control Control Control Contro	Count up Count down Count up an
Contract American Americ	County of the section of the se	Count up Count down Count up an
Image: Second	Image: specific	Count down Count up an
III (Parks completement) IV Ories & Sempletement	Outer operations Couter operations	Count down Count up an
Bock tide: "Anis Troppen Sides Bock tide: "Anis Troppen Sides (cc/c)". Bock tide: "Anis Troppen Sides (cc/c)". Connect: Bock tide: "Anis Troppen Sides (cc/c)". Bock tide: Troppen Sides (cc/c)". Bock tide: Troppen Sides (cc/c)". Bock tide: Troppen Sides (cc/c)". Bock tide	Counter operations CU Cu	Count down Count up an
Add mere block Benefic (CC)	CTU CTU CTU CTU CTU CTU COmparison operations Miss operations Minage operations Minage operations Minage operations Minage operations Minage operations Minage operations	Count down Count up an
Auf restaurs A	CD CUU Comparator operations May operations May operations Conversion operations Conversion operations Conversion operations Conversion operations	Count down Count up an
Bergers and B	CTUD COUD Comparator operations Mark functions Mark specifions Conversion operations Mark specific operations Mark specific operations Mark specific operations Mark specific operations	Count up an
Levense topological and a second	Comparator operations Math functions Math functions Conversion operations Conversion operations Word logic operations	
The Start Sta	L Math functions Nove operations Conversion operations Program control operations Word logic operations	
Constant of the sables Consables Consables Consables Constant of the sables C	 Nave operations Conversion operations Program control operations Word logic operations 	
No. 0 No. 0 No. 0 10. Work man fore whiles "Sec. Construct". 80,0 °° "Sec. Construct". 80,0 °° "Sec. Construct". 80,0 °° 10. Work man fore whiles "Sec. Construct". 80,0 °° "Sec. Construct". 80,0 °° "Sec. Construct". 80,0 °° 10. Work man fore whiles "Sec. Construct". 80,0 °° "Sec. Construct". 80,0 °° "Sec. Construct". 80,0 °° 10. Work man fore whiles "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° 10. Work man fore whiles "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° 10. Work man fore whiles "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° 10. Work man fore whiles "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °° "Sec. Construct". 90,0 °°	Conversion operations	
************************************	M Program cantrol operations	
• Control for existing 0,0,0,1 • Control for existing 0,0,0,1 • Control for existing 10,0,0,1 • Control for existing 10,0,0,0,1	Word logic operations	È
See There: Tag_1" Int Tag_2" See There are a set of the set of t	Shift and ratate	
The provide a second seco		
All program into		
The form the set of th		
A Local modules Tag_3		
Im Ungrouped devices Image devices Image devices Image devices		
y gy common sets y 10 Documentation setting: y 10 Documentation setting:		
All upcomentation seconds		
Details view		
name didora	1.00	
files Adves	< 11	
	> Extended instructions	
	> Technology	
160%		
S Properties Sinds 20 Section 1	> Communication	

6.After Enabling the Counter Block

		Second second in the second se		And the former of		Teating	
	1000	CARLEN AND DESCRIPTION OF A	The second of the Area and the second	INTERCORD F	to the Control of Cont	1 CONTRACTOR OF THE OWNER	- India
evices		A LA PARA LA PARA				Options	
	自己		III 12 · 22 · 11 12 C* 6o 64	\$₹ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	E81		
	1000	Main				> CPU operator para	o(
] program1 Add new device	- • E	Name	Data type Default value	e comment		✓ Call environment	
Oevices & networks		1 41 * Poput	soot	initial call of this ce	6	No condition defined.	
THE PLC. 1 [CPU 1214C ADDOMM		ALL . Remanders	bool	when, diremanent data are available		A second processing the	Chenge
T Device configuration		Sau - contra monte			19		Contraction
Su critine & diagnostics			2				
* S Program blocks			service states and services and s				
Add new block	1.10	 Block title: "Mein Program : 	Sweep (Cyde)				
🖝 Main (001)		Laser Internet					
System blocks Technology abjects	•	▼ Network 1:				> Breakpoints	
tai technology objects tai totemal source files		ALC: NOT COME				✓ Call hierarchy	
PLC tags		(D)					
PLC data types			10				
· Die watch and force tables			IZD			1	
Collee backupt			Counter_			1	
. Trarat	_	ACCURED STREET	100			1	
B Device proxydata			CIU I	100.0		1	
In Program Info				"Tag_0"		1	
FLC alaren tele lititi	_	cu	9	Proved		1	
Cm Local modules			0			1	
Lei Ungrouped devices	all a la	760.1 7169_3*	CV - Mag 4"			1	
Common data			CV-1 100_1			Intervision et	
Documentation settings	0					No call path	avoitable
letails view	100						
						1	
		Manage State Street as					
		Concernance and					
ame Address	(TA 0108052535205					
						1	
						1	
						1	
				100%	in million	1	

7. After Resetting the Counter Block

Devices Options Image: second and force above			program1 + PLC 1 (CPU 1214C AUDCREV) + Program blocks + Main (OB11	/Teating 🖉 🗊
Image: Section				Options
Import		8		
Bock Utiles " Main Pogean Sweep (Cycle)" Comments Social Indust as the value of the OB Indu	Add new device		Name Data type Default value Comment	✓ Call environment
Block title: "Main Program Surep Kyle?" Bock title: "Main Program Surep Kyle?" Bock title: "Main Program Surep Kyle?" Bock title: "Main Program Surep Kyle?" Comment: Source program Sure title: Comment: Comment:	PLC_1 [CRU 1214C ACIDORIy]			Change .
Connected Connected	 Conline & diagnostics Program blacks Add new black Mein [081] 	•	▼ Block tille: "Jain Pogram Sweep (Cpcle)" Camment	
Image: Section 2 and Sectio	Car Technology objects Car External source files			NAMES AND ADDRESS AND ADDRESS ADDR
	Ser Turk date types Set batch and force tables Set batch and set batch Set batch and set Set batch and set		**E_Country 0.00* *0.0 CDV* *19_1* Mit *19_2* The diamond in the second in the seco	No call path available
ame Addres	ame Address		Network 2:	

CONCLUSION:

Thus the UP Counter operation was successfully performed using PLC software.

<mark>DOWN COUNTER</mark>

AIM OF THE EXPERIMENT:

To study about the Down Counter operation using PLC Software.

APPARATUS REQUIRED:

- PLC Trainer kit
- Personal Computer Installed with PLC Software
- Ethernet cable
- Patch chords

THEORY:

The counters are mainly used for counting the values from the field equipment. The DOWN Counters can be used to counting the values in down ward manner. It can be performing the decremental purpose.

PROCEDURE:

- 1. Click **PLC-1** in the project tree "1" then click **program block** "2" and click **main OB1** "3".
- 2. Now the below image shows (Object Block1) OB1 is created.
- 3. To select the DOWN Counter block in plc goes to **Instructions** Tab and click the **Basic Instructions** then select the **CTD** block for DOWN Counter operation.
- **4.** Now the DOWN Counter call options block can be opened. Then change the **data block** name in that Name block. And then click the **ok** button.
- 5. Now the **DOWN** Counter block is created in network.
- 6. The following details can be entered into the UP Counter block is given below.
 - CD Make the counter Input
 - Q- Counter Output
 - LD load Preset value (After reach the value of 0)
 - CV Count Value
 - PV Preset value (Set Value)
- 7. Then save the program and goes to online mode and monitoring the counter values before and after enabling the counter block.

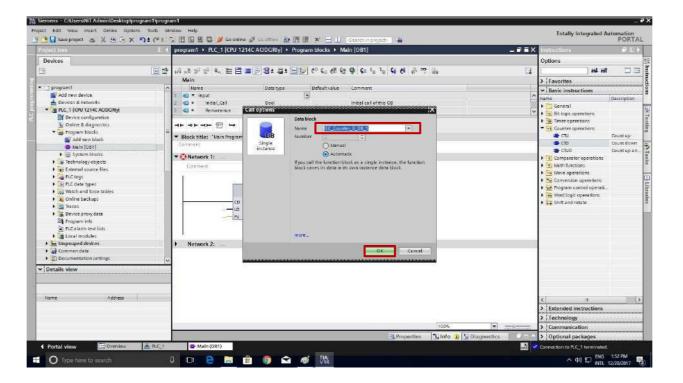
LADDER LOGIC PROGRAM:-

1.

Device Cyclone Image: Second	nject trive 🛛 🕄 🐔	program1 + PLC 1 [CPU 1214C ACID	trund . Designation blacks . L	nia feanal	_ • • • ×	Instructions	1.111
Address Addres Addr	And and the second s		control + Hodi an Diocia + H		and a second	and the second se	
Main Jengonini Jengonini Jengonini Image: Address desise Denses to sensels Jengonini Jengonini Image: Main Denses to sensels Denses to sensels Jengonini Image: Main Image: Main Jengonini Jengonini Image: Mai							
J proprint A Jenne Open pp October pp		品品ができ 私 田田 第日第	1 월 1 월 12 년 ⁰ 60 년 13	9 6 1 2 6 0 0 0 1	3	64 65	n: 🗆 (
Mathematical Mathematical Decise Stewards Decise Stewards	11/09/12-0	Main				> Favoritos	
Mode new device Image: model Image: model <t< td=""><td></td><td>Name D</td><td>lots type Default volue</td><td>Comment</td><td></td><td>Y Rasic Instructions</td><td>And an owner of the local division of the lo</td></t<>		Name D	lots type Default volue	Comment		Y Rasic Instructions	And an owner of the local division of the lo
Bock 18 Sectods Image: Bock 1 and 1 an					A	and a second sec	Descripti
Second Seco		2 -C • mitiat_call #	lool				and a provide state
Import ordigation Import ordigation Works differentiation Import ordigation Import ordigation Import ordigation Import ordin Import ordigation ordioritio		3 - Pemanence B	tao(
So Onine & Biggentitis B. The Conservation B. The All-Solar B. The Conservation B. The All-Solar D0-1 B. The Conservation D0.1 B. The		THE REPORT OF A DESCRIPTION OF A DESCRIP					Hamielbra
We do not book Bit (H)(H) We have book Bit (H)(H) We have book Difference We have book						H -11- A	#1++14
We dance box 0c)- We devolve these 0c)- We devolve these </td <td></td> <td>· Black Hitle: "Elbis Reserve Susan IDe</td> <td></td> <td></td> <td></td> <td>Ht -Incomally ope</td> <td>International Contract (San</td>		· Black Hitle: "Elbis Reserve Susan IDe				Ht -Incomally ope	International Contract (San
Sector Addres 01-0							Assignmer
1 Gib Darradi Indexter Res Channell 2 Change M Change M 1 Gib Darradi Indexter Res Change M 2 Statust and Res substate Change M 3 Statust and Res substate Change M 4 Statust and Res substate Change M 4 Statust and Res substate Change M 5 Statust and Res substate Statust and Res substate 6 Change M Change M 6 Change M Statust and Res substate 7 Statust and Res substate Statust and Res substate 8 Statust and Res substate Statust and Res substate 9 Statust and Res substate Statust and Res substate 9 Statust and Res substate Statust and Res substate 9 Statust and Res substate Statust and Res substate 9 Statust and Res substate Statust and Res substate 9 Statust and Res substate Statust and Res substate 9 Status							Negate as
Addres Connection Addres Connection Addres Connection Addres Connection		 Network 1: 				10 -05	Recetout
Addres		Comment				(3) -(5)	Set output
Address Address Address						O SET BP	Set bit feil
Addres						KI RESET_BE	Resetbitt
Connection Addres Ad							Setireset !
Addres							Repetivet
bit pogram min mit - delimation Bit Russiane maintain Russiane maintain Russiane maintaine Address							Scan open
In Confirment Comment In Confirment El 1-00-							Scen open
I contractive B) -00- I toporque division B) -00- I toporque di listattionos		With States					Set operar
Europage dedexs E /		Comment					Set operar
k ∰ Common data B Disconstructions territys: * @ Languages & resources → Details view + Name Addres =		1					Scan RLO
bill bounderstation settings bill s 2 site Datails sitew bill s 2 site Datails sitew bill s 2 site Name Address 2 and bill s 2 site S and bill s 2 site S and bill s 3 site							ScanRLO
Canguages & Arstaures Control of the staures Control of the st		<u>-</u>					Detect pot
Details view + & berrye starts Consider the starts							Detectives
Fame Addres € Comparison operations Fame Addres € 1 bit American Extended Instructions Extended Instructions Extended Instructions		6 M					
+ D Net Auction + S Net Auction + S Net Auction + S Net Auction - S Extended instructions - S Extended instructions - S Extended instructions	Details view						
Name Address C n S Extended Instantions Technology							
Rame Addres C R S Extended Instructions S Technology							
Extended instantions Technology	Name -					Contract and conversion of the second second	
> Technology	Nevres					and the second se)
						Extended instructions	
100%						> Technology	
					100%	> Communication	
Properties Unite U Diagnostics Optional packages				Properties 1		> Optional packages	

2.

Sjeck tree Devices	L program1 + PLC_1 [CPU 12	14C AC/DC/Rly] > Program block		- F = X Instructions	
SS122				_ ■ ■ X Instructions Options	. PEI
	Test and the second second in the second	and the latter to be a set of the set of the			el el 🗆 🗆
	토말 해 제 위상 특별 Main	■ 🛛 🛛 🗶 🖓 🖓 🖉 🖓 🖉	êl 92 19 €= 1 ₂ 1 ₂ 64 êl 48 19 20	E Favorite:	
] program1	Nome	Data type Default	value Comment	✓ Basic ins	A A A A A A A A A A A A A A A A A A A
Add new device	1 🚭 🕶 input			Name	the second
🖬 🚠 Devices & networks	2 40 * Initial_Cal	800	Initial call of this GB	Fi Genera	Description
FLC_1 (CPU 1214C ACIDC/Rig)	3 🕢 = Remonence	Bool	=True, if remanent data are available	V Genera	
T Device configuration			(and press)	i i i i i i i i i i i i i i i i i i i	
& Online & diagnostics		2			
* 🔐 Program blocks				* Fi Counter	
Add new block	 Block title: "Main Program 	Sweep (Cycle)"		: CU	
🚭 Main [051]	Comment				1801100411
System blocks	Network 1:			a clu	
F Technology objects					rator operations
External source files	Convert			• 💷 Alath Ru	
PLC tags	1			+ 🖂 Klove og	
CE PLC data types					sion operations
Watch and force tables					n control operati
Coline backups				 Word to 	
1 Traces) II Shitan	d rotate
Device provy data	Network 2:				
2 Program info	P HOURSE 21				
PLC alarm text lists					
Im Local modules					
Lingrouped devices					
Common data					
Common cata Common cata Common cata					
Contraction settings	*				
Details view	proved the second se				
Name Address				<	- 14
				> Ferrandar	d instructions
				the low line includes and an end of the	and the second
				> Technolo	2007
			100%	Commun	lication
			🕑 Properties 🛛 🐧 Info 👔 💆 Di		
Portal view	B.C_1 👺 Main (081)			🔛 🎺 Connection to	PLC_1 terminated.



4.

🗄 🖬 new project 🎽 🐰 🖻 🖻 🗙 🕤 1 (* 1							PORTA
ect tree 🛛 🖬 🕯	piogram1 + PEC_1[CPU 1214C	AC/DC/RIy] > Prog	ram blocks 🕨 Ma		_ # # X	Instructions	P E (
Devices						Options	
1	前は空き 4, 10日日日	8:2:5:) eº 60 ell 62	· · · · · · · · · · · · · · · · · · ·	13	ini lie	
	Main					> Favorites	12012201
] program t	Nome	Data type	Default value	Comment		✓ Basic instructions	Indiate de la citada da
Add new device	1 🖅 • input				0	Name	Description
🚠 Devices & networks	2 - initial_Cal	Bool		Initial call of this CG	1	DXMLCL	Description
FLC_1 [CPU 1214C AC/DCIRUy]	3 41 * Remanence	Bool		-True if remanent data are available		• General	
Device configuration				CATE (FIRM)	CAR.	Bit logic operation Timer operations	
W Online & diagnostics							
💌 🕁 Frogram blocks		1000 1100 1000 1000 1000 1000 1000 100				 Fi Counter operations CTU 	Countup
Add new block	 Block title: "Main Program Sweet 	p (Cycle)"				= CTU	and the second se
🚭 Main [081]	Comment						Countdown
 System blocks 	Network 1:					CTUD	Countupen
F 🙀 Technology abjects						Comparator operations	
External source files	Ganmera					Math functions	
PLC tags	T.					Move operations	
CE PLC data types	5002	8				Conversion operations	
Watch and force tables	"EC_Court 0 DB 1	ter <u>s</u> (3				Program control operati.	
🖡 🚺 Online backups						• Word logic operations	
• 🔄 Traces	100.0 CTD			500.0		Shift and rotate	
Device provy data	"Tag_1" Int	100		"Tag_2"			
B Program info		Q					
PLC a larm text lists	2000	WWW10					
Instanting and the second	10.1	CV - "Tag_4"					
Im Ungrouped devices	"Tag_3"						
Common dete	5 — PV						
Decumentation settings		-					
Details view							
Wodule							
	Network 2:						
Name						<	- 100
Device configuration						> ! Extended instructions	
Online & diagnostics	8						
Program blocks					1122	> Technology	
Technology objects				100%	·	> Communication	
Enernal source files				🙁 Properties 🚺 Info 😰 💆 Diag	nostics	> Optional packages	
Portal view	Main (OB1)					The project program1 was saved	succe.

3.

5. Before Enabling the Counter Block.

pject tree Devices posysm1 A dati net device. Device & networks with pock of networks with pock of networks with pock of networks pock of networks					_ # # X	Instructions Options	# 11 k
Pospani Addinen device To Device a networks To Dev	Main • Nome	THE CONTRACT	😥 १९ ६२ सा ६			Options	
program 1 a Add nev device b Devices & inclusion a Public & Devices & Devices & inclusion a	Main • Nome	THE CONTRACT	😥 🤭 😜 🥶 😒				
program 1 a Add nev device b Devices & inclusion a Public & Devices & Devices & inclusion a	Main • Nome	THE CONTRACT		9 G L L G A & ??		est est	
Add new device Covices & network: Covices &	* Nome	Part of the second s				> Favorites	
Devices & networks Ju PLC_1 [CPU 1214C AC/DC/Rby]	1 del • input		Default value	Comment		Basic instructions	***
PLC_1 (CPU 1214C AC/DCIRIty)			1			and a second sec	1
	2 a initia[_Cal	Bool		initial call of this GE	1	Name	Description
Date:	3 a * Remanence	Bool		=True, if remanent data are available	10	General General	
Device configuration				(ALL (FRAM))		I dit logic operations	
& Online & diagnostics						1 Imer operations	
* 😹 Program blocks	=					Counter operations	-
Add new block	 Block title: "Main Program Swi 	eep (Cycle)"					Count up
🚭 Main [081]	Comment					CTU0	Count down
 System blocks 	Network 1:						Countupen
Technology abjects	and the second sec					Comparator operations Auth functions	
) a) External source files	Economical					Math functions Move operations	
PLC tags	10						
GE PLC data types	508					Conversion operations	
Watch and force tables	"IEC_COU 0 08	inter_				 Program control operation 	
🕨 🙀 Online backups	and the second se	-				• 🐨 Word logic operations	
Traces	100.0 CW			100.0		Shift and retate	
Device provy data				"Tag_2"			
E Program info		Q					
PLC alarm text lists	2000	WWW10					
Local modules	100.1	CV - "Tag_d"					
Engrouped devices	"Tag_8"						
Gommon dete							
Commentation settings	5 — PV						
	2						
Details view	1. B						
Module	Notwork 2:						
Name							
Device configuration						S	
Online & diagnostics						> Extended Instructions	
Program blocks						> Technology	
Technology objects				100%	The second second	> Communication	ALCONCERCT.
External source files	1				1 Diagnostics	> Optional packages	
Portal view 🔄 Crerview 📥 R.C.1	- Main (081)			Stropenes Simo		The project program1 was saved	

6. After Enabling the Counter Block

vices program1 ∰ Add newdevice	(III) II					
💕 Add new device	1 III					Options
💕 Add new device				·····································	a	
💕 Add new device		Main				3 CPU operator panel
		Rathe	Dece type Default vi	alue comment		✓ Call environment
		t -0 * hput	[3]		^	No constition defined.
Devicet & networks		I -O + Intal_Cal	Basi	Initial call of this CIP		
La nuc_1 (Chu 1214C AGDGRb)		3 -cl • Remanence	5001	«Nue, if remanent data are available	Y	Change
T Device configuration				aren aren		
🧏 Online & diagnostics			*			
* 😹 rrogram blocks		Company of the second sec	Managaran Ing			
Add new block		- Block Etle: "Main Fragram Se	weep (Cyde)*			
- Mile (OB1)	•	Comment				
Esstem blocks	•	 Network 1: 				
• La Sechnology chierts		Concentral for				> Breakpoints
> Titernal source fles		SHOWARE				✓ Call hierarchy
+ Cartuc teas		The second s				
 Fig RC dam tipet 						
Go PLC data type: Wetch and faree tobles			0			
		"IEC Cr	aunter			
Online backups		0_0	38_1*			
* mares		360-0	CTD	300 0		
• Device provy data			204	"749.2"		
La Program info		1				
🔚 RC nlarm text lists		and the second sec				
Local modules	2	30.1	5 TANUTO			
🛏 Lingrouped devices		300.1 *Tag_3*	CV ****			
Common data			ev			
Documentation settings	100	the second se				No call path available
Lange and the second	1 M	5PV				
etalls view	and the second second					
dulo						
		Network 2				
		Normore 2				1
me.	100					
vice configuration	0					
line & diagnostics	10					
ogram blocks						
chnology objects				700%	·	
is mel source files				Properties		1
ortal view	H NG1	Main (UE1)		Set Linhauses	and the second se	Connected to FLC_1, via address IP=10

7. After Load Presetting the Counter Block

vices				
				Options
		(M # 순 씨, 는 면 로 한 원 * 원 * 드 달) (* 6, 전 원 및 6, ½ ~ 1 Main	= 9 et & 💬 🔤	> CPU operator panel
program 1 ¹ Add new device		Hame Date type Default value Comment 400 ≠ Input		Call environment
Devices & networks PLC_1 (CPU 1218C ACIDCRM) Device configuration Online & diagnostics Device & diagnostics Device & diagnostics Device & diagnostics Device & diagnostics	=	al • minist_Cail Soci initial Cail of fini al • Remanence Soci -True, if reman • drie - the - finitial - al - finitial - al of finitial - a	n OB ent deta are available	Change
Main (061) System blocks Technology objects External source files	÷	Network 1:		Breakpoints Call hierarchy
A Composition of the set of		1000 1000		Ho call path available
etalls view dule me vice configuration line 8 disgnostics		Network 2:		

8.

	program1 + PCC 1 [CPU 1214C AC/DC/IIIy] + Program Mocka + Main [OII1]	- - - - - - - - - - -	Testing 🖓 D
vices			Options
	· 제 : 최 양 양 후, 돈 프 코 크 영 : 월 : 그 같이 안 \$5 년 원 양 \$4 년 년 일 61 년 유 11 등; Main	3	> CPU operator panel
Program 1 Control of the second secon		4	Call environment No condition defined Change
Device configuration Online & disposition Software Block Add new block Mini (081) Software Blocks	H⊨ HT⊨-O 10 is	1.5	
La chonalogy objects Technology objects Technology objects Technology objects Technology	Network 1:]	> Breakpoints → Call hierarchy No call path available
etails view write write write configuration mine & telaportics oggen Marchs chanslagy objects chanslagy objects	Network 2:	· · · · · · · · · · · · · · · · · · ·	

CONCLUSION:

Thus the DOWN Counter operation was performed successfully using PLC software.

Write program on MOVE, Control statement, math function, data manipulation technique on PLC

MOVE

AIM OF THE EXPERIMENT:

Write a simple ladder logic program using MOVE instruction.

EQUIPMENTSREQUIRED:

- PLC software
- Desktop Computer

THEORY:

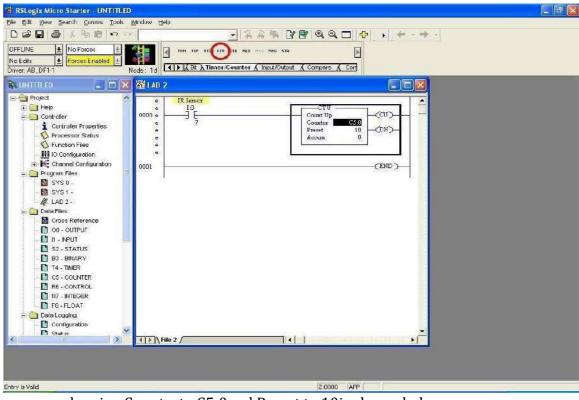
•

	MOV		
_	Move		÷
	Source	????	
		2222	
	Dest	2222	
		2222	

When rung conditions preceding this instruction are true, the MOV instruction moves copy of the source to the destination each scan. The original value remains intactandunchanged in its sourcelocation.

PROCEDURE AND LADDER LOGIC PROGRAM:

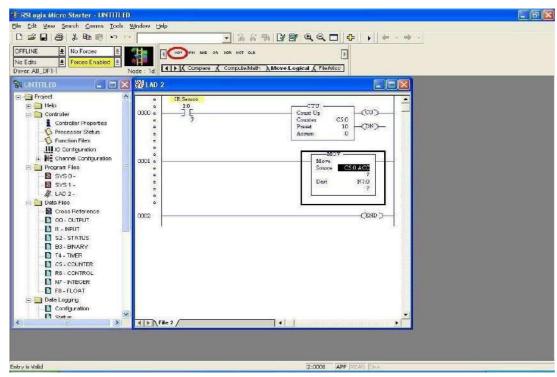
TakeoneNObitandassignaddressI:0/7andcommenttoIRSensorandtakeCount



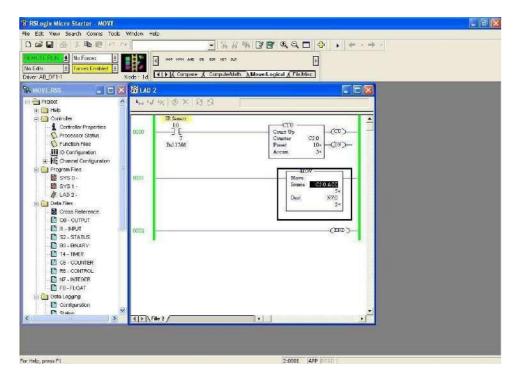
erand assign Counterto C5:0and Preset to 10is shown below.

[34]

• Take Move instruction form Move/Logical Instruction and assign Source toC5:0.ACCand assign Dest. to N7:0 is shownbelow.

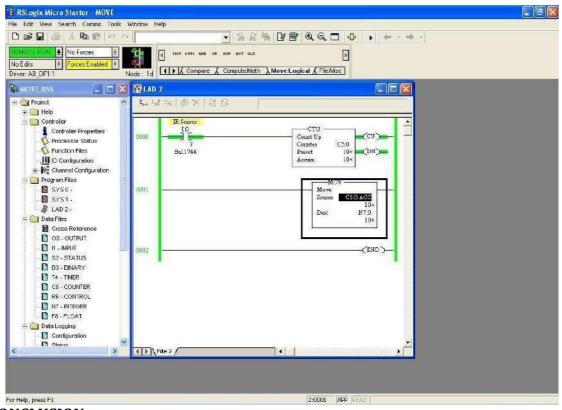


- Verifyanddownloadandruntheprogram.
- Whenany obstacleinfrontofI:0/7(IRSensor)thencounterC5:0willstartcounting . When counter Accumulator value is increase and move instructionMovetheCounteraccumulatorvalue in DestinationN7:0 is shown below.



• Whenany obstacleinfrontofI:0/7(IRSensor)thencounterC5:0willstartcounting. When counter Accumulator value is increase and move

 $instruction Move the Counteraccumulator value\ in\ Destination N7:0\ is\ shown.$



CONCLUSION:

Thus the MOVE operation was studied and performed successfully using PLC software.

CONTROL STATEMENT

AIM OF THE EXPERIMENT:

Write a simple ladder logic program using Control statement (Jump & Label instruction).

EQUIPMENTSREQUIRED:

- PLC software
- Desktop Computer

THEORY:

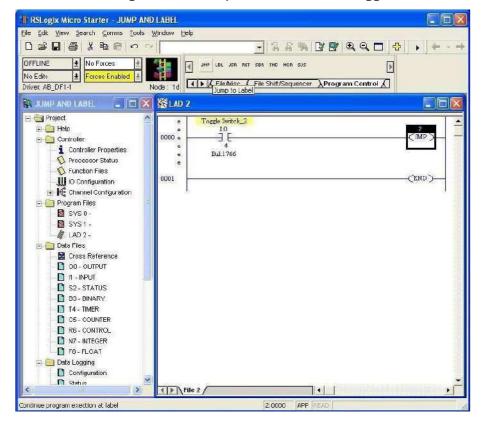
.

Whentherung conditionforthisoutputinstructionistrue, the processor jumps forwardorbackwardtothecorrespondinglabelinstruction(LBL)andresumesprogram execution at the label. More than one JMP instruction can jump to the samelabel. Jumping forward label saves program scan time bv omitting to а а programsegmentuntilneeded.Jumpingbackwardletsthecontrollerexecuteprogramsegmen tsrepeatedly.



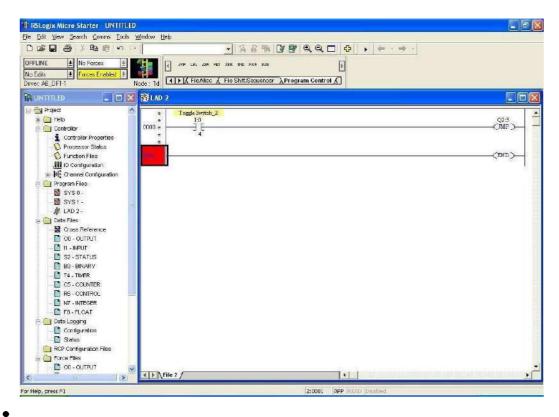
PROCEDURE AND LADDER LOGIC PROGRAM:

TakeNOBitandassignaddressI0:0/4andcommenttoToggleSwitch_2andtakeJ

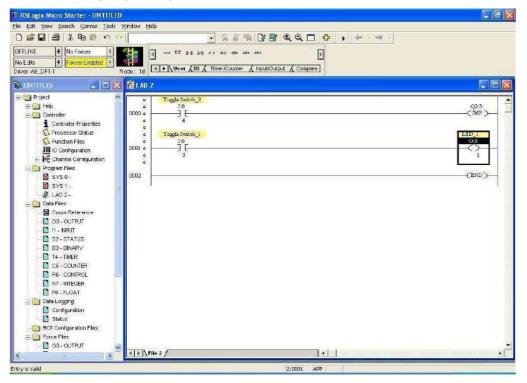


 $MP instruction\ from Program control instruction is\ shown below.$

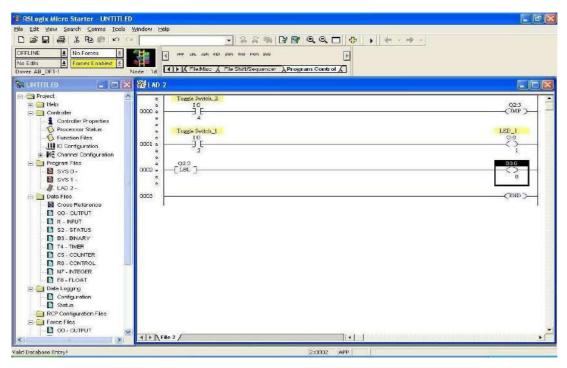
• Assignaddressto JMPinstructionto 3is shownbelow.



TakeNObitandassignaddressI:0/3(ToggleSwitch_1)andtakecoilandassignad dress 00:0/1 (LED_1) is shown below.



• Take LBL instruction from Program Control instruction and assign address Q2:3andtakeCoil and assignaddressB3:0/0 is shown below.



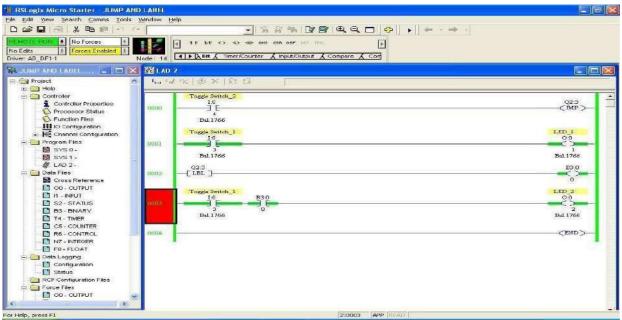
•

TakeNObitandassignaddressI:0/3(ToggleSwitch_1)andtakeanotherNObitan dassignB3:0/0andtakecoilandassign 0:0/2(LED_2)isshown below.

🕆 RSLogix Micro Starter - UNTITLED		🗖 🖬 🔁
Ele Edit Mew Search Comms Lools	<u>Window H</u> elp	
DFFLINE No Forces Forces Enabled Driver: AB_DF1-1 No Forces Enabled Driver: AB_	Image: Second	
🐘 UNTITLED 📃 🗖 🔀	醫LAD 2	
Project Project Processor Status Processor Procesor Processor Procesor Processor Processor	Togle Switch_2 0000 I I I Togle Switch_1 I 0001 I I I 0002 I I	
	())\File 2	
Entry is Valid	2:003 APP	

- Verify, download and run the program of Jumpand Labelinstruction.
- •

WhenyouenabletheI:0/3(ToggleSwitch_1)thenoutputO:0/1(LED_1)andO:0/



2(LED_2)will beON is shown below.

- When youenableI:0/4(ToggleSwitch_2)thenJUMPinstructionisON.
- When you enable the I:0/3 (Toggle Switch_1) then output 0:0/1 is off becauseexecutionofinstructionisJumponnetwork3(0003)meansoutput0:0/2(LED_2) will beONis shown below.

RSLogix Micro Starter - JUMP AND		
Ecit View Search Comms Tools M		
₽	· · · · · · · · · · · · · · · · · · ·	
No Forces 🛓	4 3 E 3/E <> <> <> <> <> <> <> <> <> <> <> <> <>	
dits 🛓 Forces Enabled 🛓		
r: AB_DF1-1 No	de: 1d I A Bit & Timer/Counter & Input/Cutput & Compare & Com	
LIMP AND LABEL	蹬 LAD 2	
Rroject	14 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 Marriell (Marriell
i Help		
🛛 🧰 Controller	Togele Switch_2	
- 1 Controller Properties		023 (JMP)
- 💭 Processor Status	4	2.0
S Function Files	Bul.1766	
IO Configuration	Toggle Switch 1	LED 1
Program Files	19	9:0
SYS0-	0001	C
SVS1-	Bul.1766	Bul 1766
ALAD 2 -	023	E3.0
🗌 🧰 Data Files	0002 [LBL]	
		0
	Togele Switch 1	LED 2
	I:0 B3:0	0:0
S2-STATUS		
B3 - BINARY	Bul.1766	Bul 1766
CS - COUNTER		
BE-CONTROL	0004	(END)
N7 - INTEGER	EXAMPLE IN THE REPORT OF THE REPORT	
F8 - FLOAT		
Data Logging		
- Configuration		
- 🛄 Status		
RCP Configuration Files		
Force Files		
and the second se		
lp, press F1	2:0003 APP READ	

CONCLUSION:

Thus the Control operation was performed and verified successfully using PLC software.

MATH FUNCTION

AIM OF THE EXPERIMENT:

Write a simple ladder logic program using Math instruction (Addition).

EQUIPMENTSREQUIRED:

- PLC software
- Desktop Computer

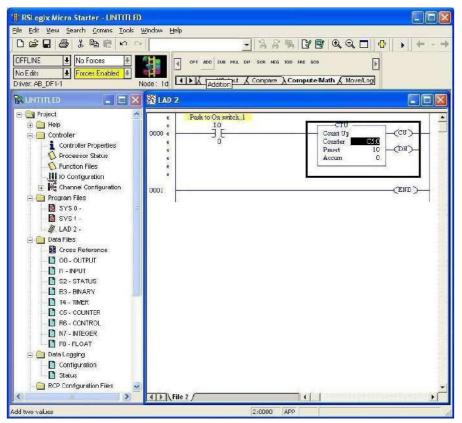
THEORY:

Use the ADD instruction to add one value to another value (Source A to Source B)andplacetheSUM in destination.

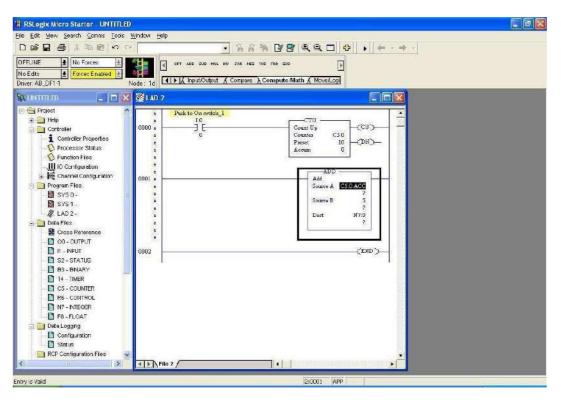
ADD Add	۰.,	
Source A		
Source B		
Dest		
Dest		

PROCEDURE AND LADDER LOGIC PROGRAM:

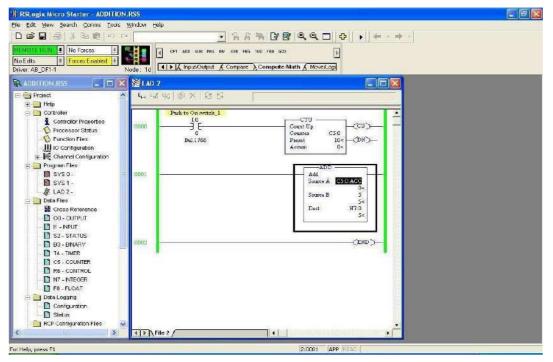
• Take one Examine if closed bit and assign it I: 0/0 (Pushbutton Switch_1) and Take one up counter and assign it Counter to C5:0 and Preset to 10 and Accumto0 is shown below.



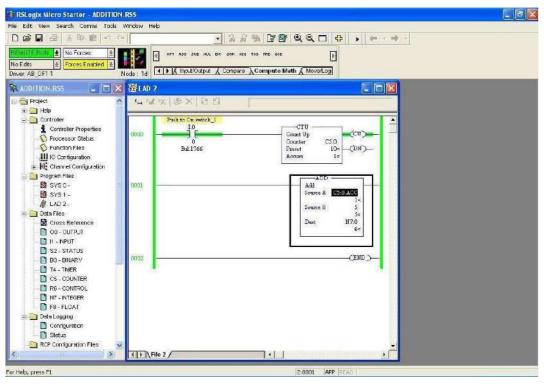
• Take Addition instruction from Compute Math (in instruction Bar) and assign itSource A to C5:0.ACC and Source B to 5 and Destination to N7:0 is shownbelow.



CompileandDownloadandRuntheProgram.



• WhenyoupressI:0/0(PushbuttonSwitch_1)counterC5:0startcountingandadd itionis shown in N7:0.



UseofAdditionInstruction

CONCLUSION:

Thus the ADDITION operation was performed and verified successfully using PLC software.

DATA MANIPULATION

AIM OF THE EXPERIMENT:

Write a simple ladder logic program using data manipulation technique (Bit shift left).

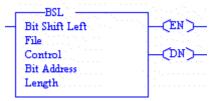
EQUIPMENTSREQUIRED:

- PLC software
- Desktop Computer

THEORY:

BitshiftinstructionisusedtoshiftthebitinanydirectionasuserwantsuchasBitshiftLeft (BSL),Bit Shift Right(BSR).

Bitshift left (BSL): When the rug goesfrom false to true, the controller set the enable bits (Enable Bit 15) and the data block is shifted to the left one bit position. The specified bit at the bit address is shifted into the first bit position. The last bit is shifted out of array and stored in unloaded bit. The shift is completed immediately



The following general information applies to bit shift instruction

Enteringparameter:

Enterthefollowingparameter when programming these instruction

٠

Fileistheaddressofthebitarrayyouwanttomanipulate.Youmustusethefileindi cator(#)in thebit arrayaddress.

• Controlistheaddressofcontrolelementthatstorethestatusbyteofinstruction,the sizeofarray(in number ofbits).

The control element is shown in below

	15	13	11	10	00
Word 0	EN	DN	ER	UL	Notused
Word 1	Sizeof	bit arı	ay(nu	mber of b	its)
Word2	Reser	ved			

Status bitsof controlelementmaybe addressed

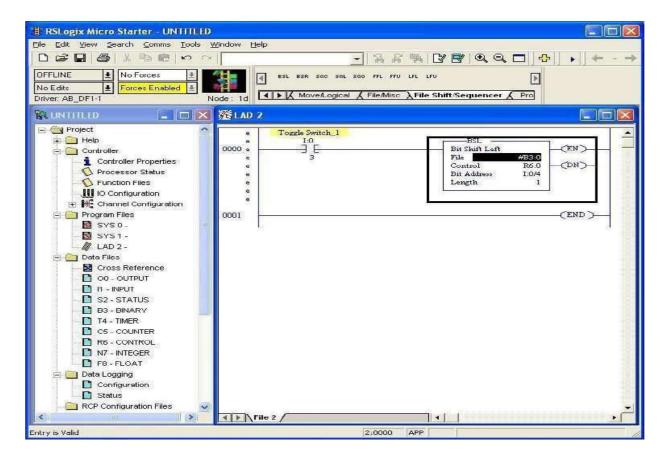
bymnemonics.Theyincluded • Unloadedbit UL(Bit 10)is

theinstructionsoutput.

- ErrorbitER(Bit11)whenset, indicates the instruction detected an error such entering an egative number for length or position.
- DoneBit DN (Bit 13)when setindicate thebitarrayshiftedoneposition.
- Enable bit EN (Bit 15) is set on a false to true transition of the rung and indicate the instruction is enabled.
- Bit address is the address of the source bit. The status of this bit is inserted ineitherthefirst (lowest)bitposition(BSL) orlast (highest)bit position(BSR).
 - Length (size of bit array) is the number of bits in bit array up to 1680 bits.

PROCEDURE AND LADDER LOGIC PROGRAM:

• Take No bit assign I:0/3 (Toggle Switch_1) and select a File shift / sequencerinstruction in Tool bar then take a BSL (Bit Shift Instruction) and assign it Fileto #B3:0 and Control to R6:0 and File Address I:0/4 and Length 1is shownbelow.



Compileanddownloadandrunthe program.

•

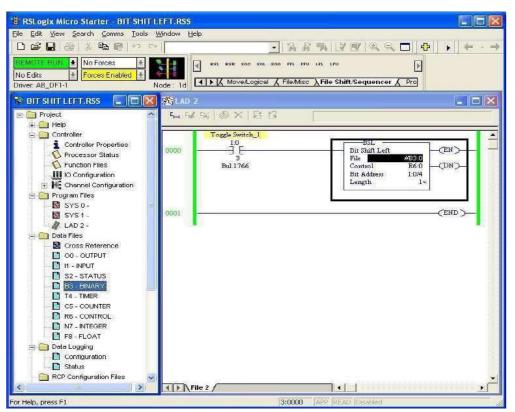
•

RSLogix Micro Starter - BIT SHIT L	LET BSS	
File Edit View Search Comms Tools		
		>
Inemotif Finite Ine Forces Ine Ine Forces Ine Force		
🙀 BIT SHIT LEFT.RSS 🛛 🔳 🖾	🔀 LAD 2	
Project Help Controller Controller Properties Function Files Function Files SYS 0- SYS 1- Constroller Construction Construction Construction State Systement Construction State Systement Construction State Systement Construction State Construction Constructi	Fm Solvital 10 Ist 10	
RCP Configuration Files	File 2	لتے ج
For Help, press F1	2:0000 APP [READ [Disabled	

ClickonB3:0BinaryOptionisshownbelow.AfterclickonB3:0Binarythiswindo wwill open.

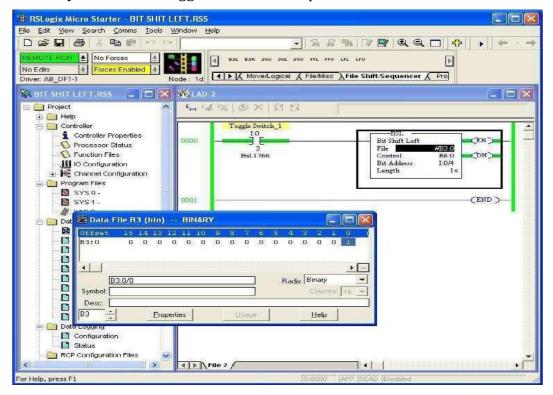
I RSLogix Micro Starter - BIT SHIT L	CLARKER CO.			
File Edit Yew Search Comms Iools !	Vindow Help			II SILA
			raalo	· · · · · · · · · · · · · · · · · · ·
AEMOTE RUN Image: Second	est see so	, soo mi mu uni uni		
	ode: 1d Ale MovelLogic	al <u> </u>	equencer & Pro	
💦 BIT SHITLEFT RSS 🔳 🖬 🔀	KLAD 2			- 0 🛛
🖃 🧰 Project 🔨	S WW OX B	8		
Help Controller	Toggle Switch_1			
Data File 83 (bin)			-BSL	CEN)
	1 10 9 8 7 6 5 4	Fil		
83:0 0 0 0 0		0 0 0 0 B	ntrol R6.0 Address 1.0/4	-CMD-
		Le	ngth 14	
		<u>+</u> =		
83.0/0	Rad	x Binary 💌		-(END)-
Symbol		Columnit 16		- S.
[02 ·] During	L Usage	Help		1.5
I H - NPUT				
D. S2_STATUS				
B3 - BINARY				3 I
CS - COUNTER				
R6 - CONTROL				
F8+FLOAT				
Configuration				
Status				<u>19</u>
RCP Configuration Files				<u></u>
< ×	File 2 /			
For Help, press F1		0.0000 APP READ	CHEADING 1	10

• YouenabletheI:0/4(ToggleSwitch_2)isshown below.



WhenyouenabletheToggleswitch,B3:0bit positionisshownbelow.

•



- WhenyouEnabletheI:0/5manytimes,B3:0BitpositionShiftLeftisshownbelow.
- WhenYouEnableand Disablethe I:0/3 (Toggle Switch_1)bitwill beshiftleft

It RSLogix Micro Starter - BU SHIT Ele Edit View Search Comms Tools D ## III ## A Re III BEMOTE RUN 1 No Forces	Window Help		- □ × + · →
No Edits Driver: AB_DF1-1		Misc \File Shift/Sequencer (Pro	
Controller Properties Processor Status Ui Do Configuration Ni Wi Channel Configuration Program Files SYS0 SYS0 SYS0 SYS0	10000 1000000	BIL File Affine (EN) Control Reso (DN) Bit Address 1:0/4 Length 15	
Dat Z Data File B3 (bin)	BINARY 12 11 10 9 8 7 6 5 4 3		
B3:0/0 Symbol: Desc: B3 Desc: B3 Proc			
For Help, press F1	5 File 2 /	I • I I 000 [APP [READ [Disabled	<u>*</u>

CONCLUSION:

Thus the data manipulation operation was performed and verified successfully using PLC software.

AIM OF THE EXPERIMENT:

To study about conveyor control system using PLC

APPARATUS REQUIRED:

Sl. no.	Apparatus Name
1.	VPAT-24 kit
2.	VPLCT-03 kit
3.	PLC software installed PC
4.	STEPPER MOTOR
5.	Patch chords
6.	Power chord

THEORY:

RUN	STOP									вох
										вох
									1	вох
										BUA
	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	Ø	0/
				ſ				_ ∾	lotor	6
					LOCALC	ONTRO	PANE	L .		0
						E	mergenc	y I		
5	0	00	000	0000				Emergence		Emergency

Conveyor System:

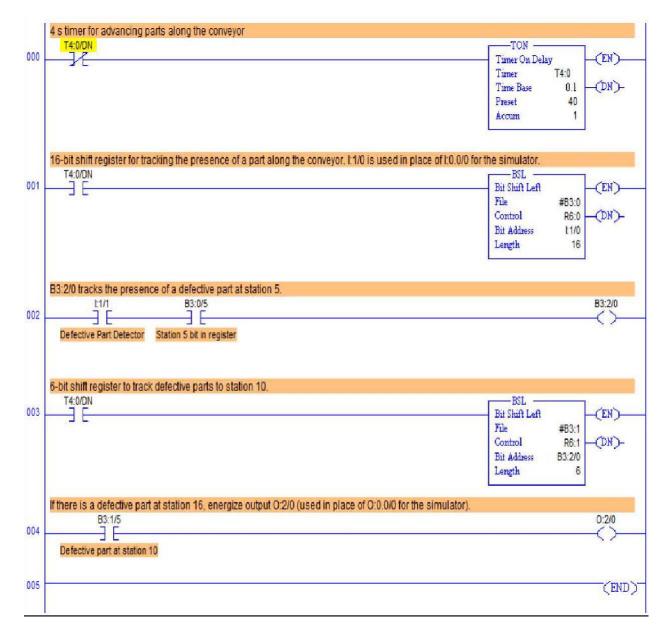
- ✓ A conveyor system is a common piece of mechanical handling equipment that moves materials from one location to another.
- ✓ Conveyors are especially useful in applications involving the transportation of heavy or bulky materials.
- ✓ Conveyor system allows quick and efficient transportation for a wide variety of materials, which makes then very popular in the material handling and packaging industries.

Stages in the System:

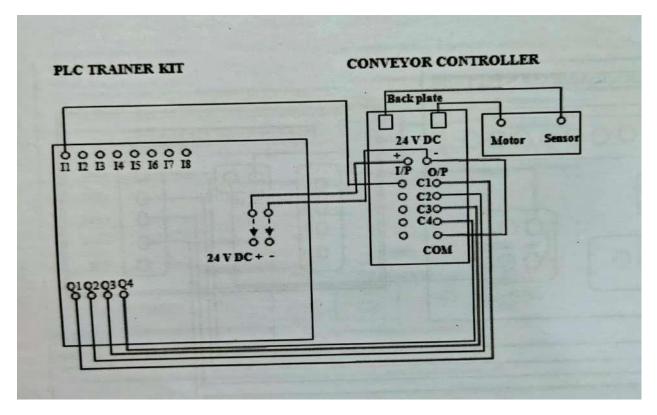
- 1. **Comparator**: Two inputs (Threshold & Feedback); compares the output of detecting network and a preset.
- 2. **Converter/Switching circuit**: Controller; takes decision based on the comparator output.

- 3. **Steeper Motor**: A stepper motor is a brushless, synchronous electric motor that converts digital pulses into mechanical movement.
- 4. **Conveyor belt**: Object translational mechanism.
- 5. **Object presence detector**: To give feedback based on presence of object at the end position.

LADDER LOGIC:-



WIRING DIAGRAM:



PROCEDURE:

- 1. Open the PLC Software and design the ladder diagram.
- 2. Interface the PLC with the system using Ethernet cable.
- 3. Make electrical connections as per wiring diagram.
- 4. Download the program and run it.
- 5. Place the object on the conveyor. If the sensor the object, conveyor stops for 2 seconds and again starts automatically.

CONCLUSION:

Thus the study of conveyor control system had been studied.

AIM OF THE EXPERIMENT:

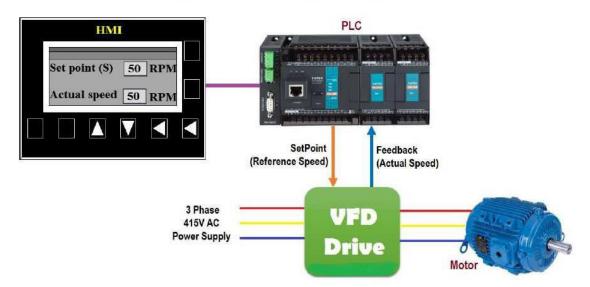
Write and implement ladder logic program to on-off the DC motor using PLC.

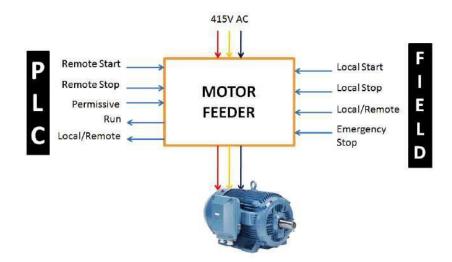
APPARATUS REQUIRED:

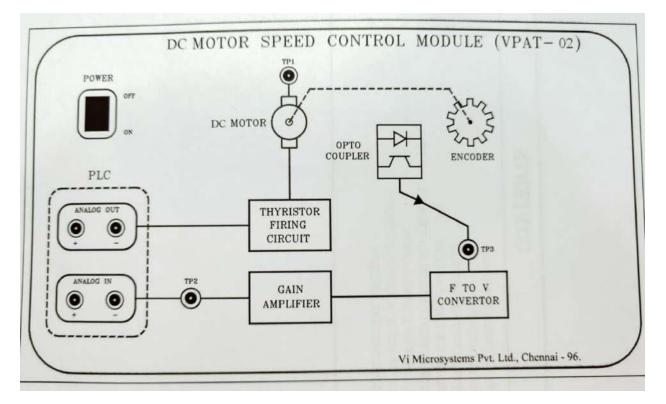
Sl. no.	Apparatus Name
1.	Speed control module trainer (VPAT- 02)
2.	PLC trainer kit
3.	Personal computer installed with TIA portal software
4.	DC Motor
5.	Patch chords

THEORY:

PLC Program to Control Motor Speed using VFD Drive



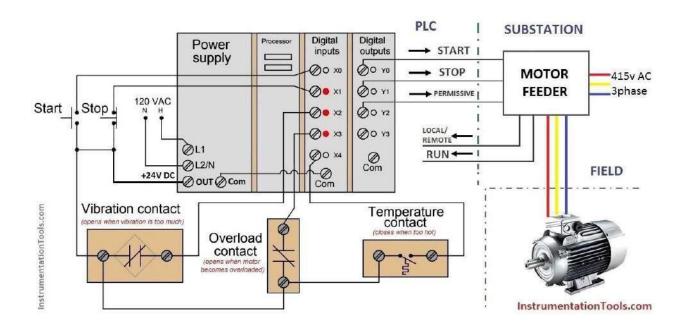




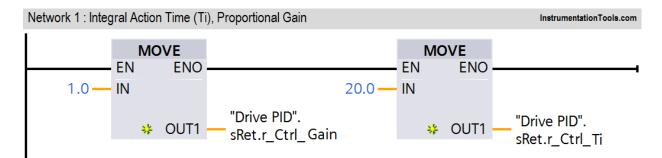
Speed Control Module:

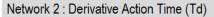
- This module maintains the speed of DC motor to the set point using PLC.
- After giving the set point (SP) the PLC produces the analog output, this will be given to motor.
- So initially the motor starts rotating at some speed.
- The speed of the motor is measured using Opto coupler sensor.

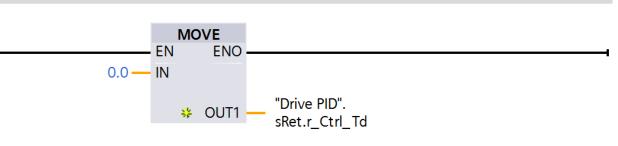
- The output of Opto coupler will be a series of pulses are converted into voltage using frequency to voltage converter.
- This voltage is process variable (PV) and is applied to the analog input of PLC.
- In PLC program PID block reads this PV and compares the both set point (SP) and process variable (PV), it creates error value and produces the control variable (CV) to the motor unit through analog output.
- This controlled output will maintain the speed of AC motor at its set point (SP).
- Likewise it forms the closed loop control using PLC.



PLC Ladder diagram to control Motor Speed

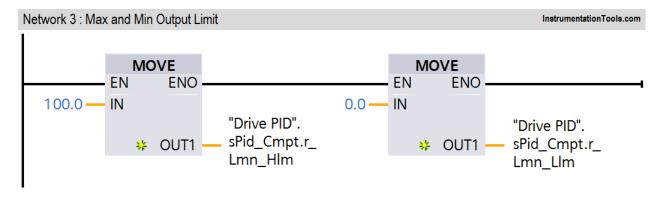


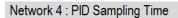


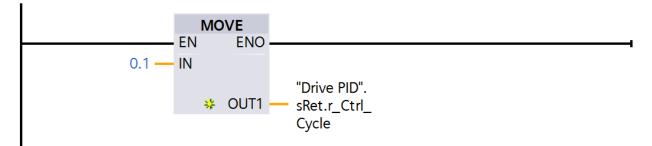


InstrumentationTools.com

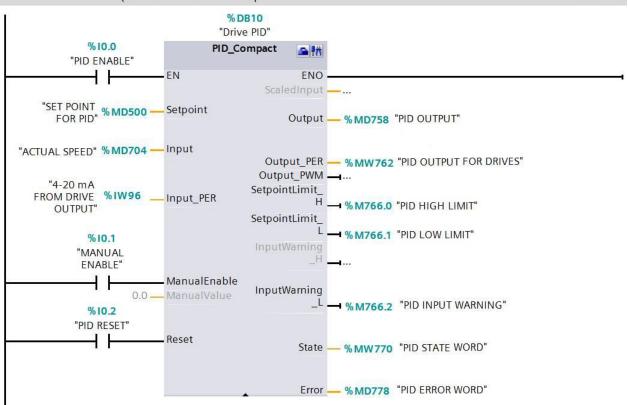
InstrumentationTools.com



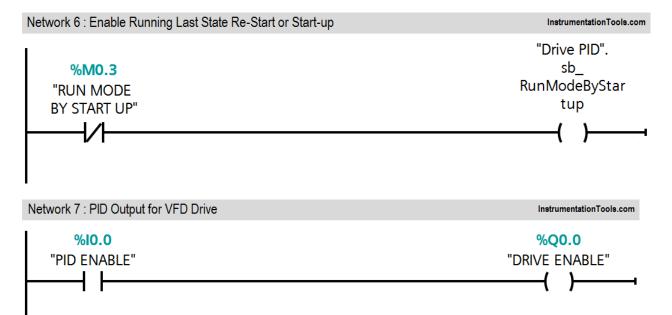




Network 5 : PID Block (Linear PID for Drives Output

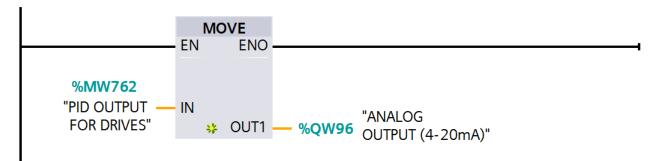


InstrumentationTools.com



Network 8 : PID Output Value Moving to Analog Output Channel

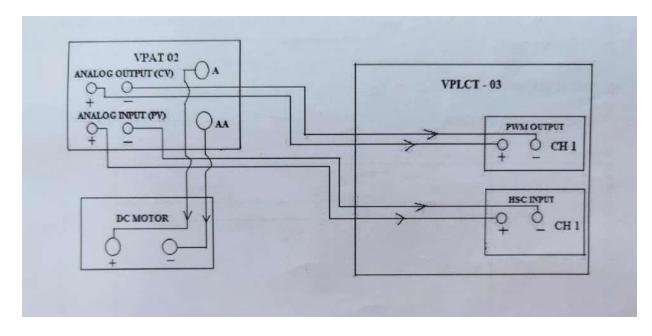
InstrumentationTools.com



Runtime Test Cases:-

Inputs	Outputs	Physical Elements
I0.0=1	Q0.0=1	Drive ON
MD500=50RPM	MW762=13838 (approx.)	Drive speed =50RPM

WIRING DIAGRAM:



PROCEDURE:

- 1. Made the connections based on wiring diagram.
- 2. Switch ON the PLC kit.
- 3. Interface the PC to PLC through Ethernet Communication Cable.
- 4. For configuration of software, programming follows the procedure.
- 5. Switch ON the DC MOTOR SPEED CONTROL kit.
- 6. Download and run the program to PLC.
- 7. Set the set point of motor speed (0-1500 rpm) i.e. 0.0 to 100.0 in PID set point block.
- 8. Based on set point the PID controller enables the PWM output pulses.
- 9. By this pulse the DC Motor starts rotating at some speed.
- 10. Opto coupler sensor senses the feedback of high speed ON pulses.
- 11. That feedback we can read from High speed counter input channel.
- 12. Now the PID controller compares the set point (SP) and Process (PV) (SPEED).
- 13. By varying the Control Output (CV) (PWM pulse) it makes the motor running at set speed.

RESULT:

Thus the ON-OFF operation of DC motor using PLC was studied successfully.

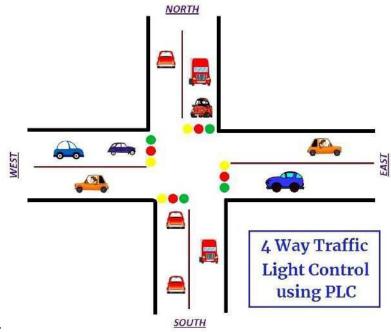
TRAFFIC LIGHT CONTROLLER

AIM OF THE EXPERIMENT:

To study the Traffic light controller system by using PLC.

APPARATUS REQUIRED:

Sl. no.	Apparatus Name
1.	VPAT-03 kit
2.	PLC
3.	PLC software installed PC
4.	ETHERNET cable
5.	Patch chords
6.	Power chord



THEORY:

Density Based Traffic Light Controller System:

- Traffic control system is used to control automatically the vehicle and human beings by using simple program logic.
- Microcontroller system is very complex to programming and wiring. Hence PLC is used to control the traffic.

- For the complex road maps the PLC provides simpler solution by means of ladder programs.
- It is four directions based traffic light controller system, which is depends upon the density of vehicle.
- If an one direction's density of vehicle is HIGH, but another one direction's density of vehicle is very LOW, then we will give more time duration to high density direction half of time duration to middle density direction, the one third (1/3) of time duration to very low density direction as respective as our decision.
- If one direction is in glow green LED [Run-mode], other directions are in red- LED [Stop-mode].

Sl. no	Address	Name	Input/ Output	
1	I:0/0	Start	Input	
2	I:0/1	Stop	Input	
3	B3.0	Memory	Memory	
4	0:0/0	East Green	Output	
5	0:0/1	North Red	Output	
6	0:0/2	West Red	Output	
		South		
7	0:0/3	Yellow	Output	
8	0:0/4	East Yellow	Output	
		North		
9	0:0/5	Yellow	Output	
10	0:0/6	North Green	Output	
11	0:0/7	East Red	Output	
12	0:0/8	West Yellow	Output	
13	0:0/9	West Green	Output	
		South		
14	0:0/10	Yellow	Output	
15	0:0/11	South Green	Output	

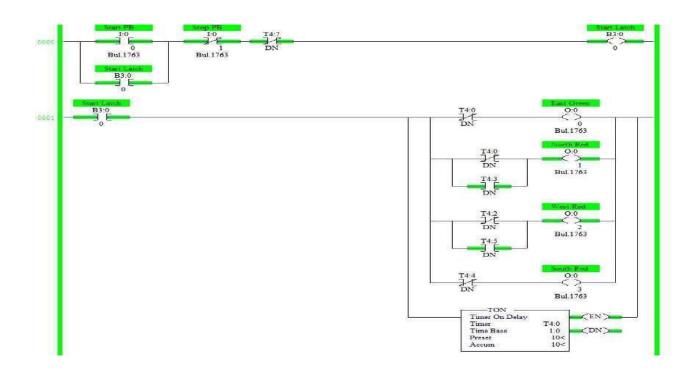
List of Inputs and Outputs

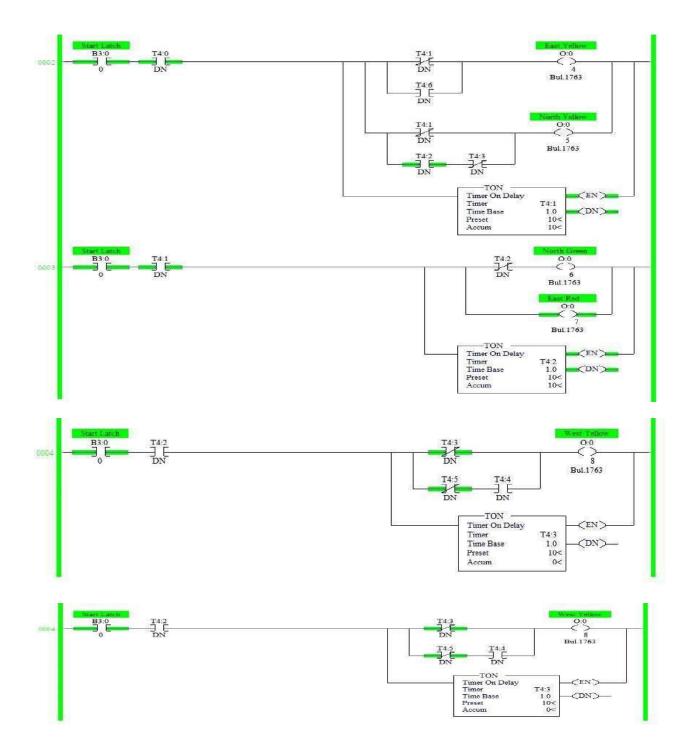
Sequence of Operation

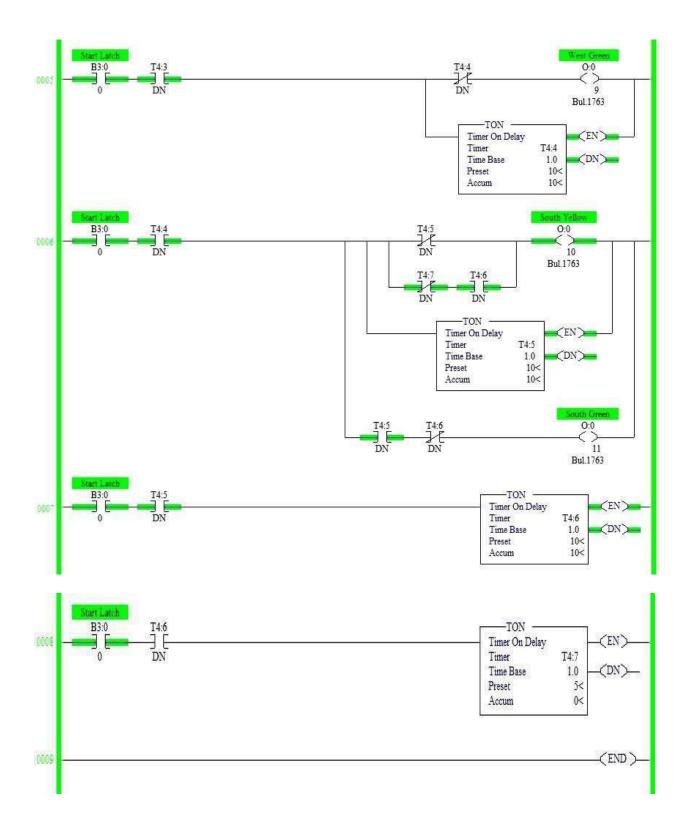
Below tabular column gives the Steps or sequence of outputs to turn ON the traffic system lamps (RED, GREEN, YELLOW)

S.NO	EAST	WEST	NORTH	SOUTH
1	G	R	R	R
2	Y	R	Y	R
3	R	R	G	R
4	R	Y	Y	R
5	R	G	R	R
6	R	Y	R	Y
7	R	R	R	G
8	Y	R	R	Y

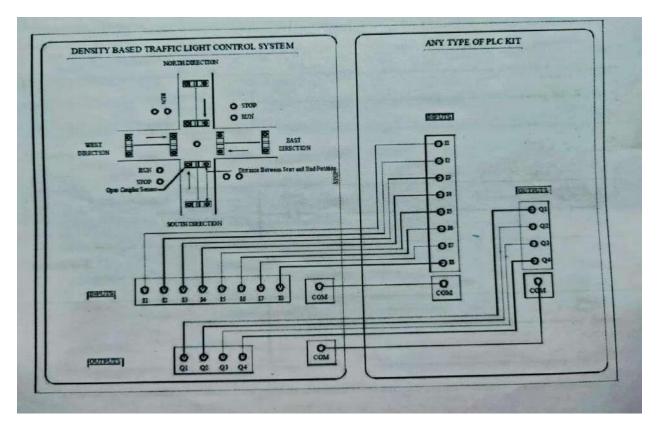
LADDER LOGIC PROGRAM:-







WIRING DIAGRAM:



PROCEDURE:

- 1. By applying start one triggering pulse to input I1.0(I9), which causes for open coilM0.0 energized in Network 1, Due to energization of open coil M0.0 initiates the operation.
- 2. It causes the open coil Q0.0 [Q1 coil] energized in Network 5 on delay timer is also energized. After completing the time delay [5sec], which is depending upon preset Time coil M0.0 will be energized. Due to energization of M0.1 it cuts the Q0.0 coil output and enables the Q0.1 west Direction output.
- 3. Due to Q0.1 (Q2 coil) will energized in Network 7, then Q1 coil Q0.0 de-energized and timer is reset in Network 8. This same process was repeated again in west direction, North Direction, and East direction.
- 4. In west direction, that means Network 8 ON delay timer will be energized depends upon Q2coil [Q0.1] then M0.2 [open coil] is energized, after completed programmed time delay, which is depends upon preset time, then rest the timer. At that time, Q2 coil [Q0.1] deenergized, timer was reset, Q3 coil [Q0.2] was energized.
- 5. In network I0, latch connection Q3 coil [Q0.2] is used to energized for continuously. Hence on delay timer are starts ON due to Q3 contact [Q0.2] energization in run- 8, then M0.3 [Q0.2] will energize. After completion of time delay, which is depends upon preset time, then reset the timer. Then Q4 coil [Q0.3] energized and Q3 coil [Q0.2] are de-energized at that same time.

6. In Network I3 ON delay timer is starts ON, due to contact [Q0.3] energization, then M0.4 coil will be energized, after completion of time delay, rest the timer this time delay depends upon the [M0.4] preset time.

CONCLUSION:

Thus he study of traffic control system was done by using PLC.