LAB 15

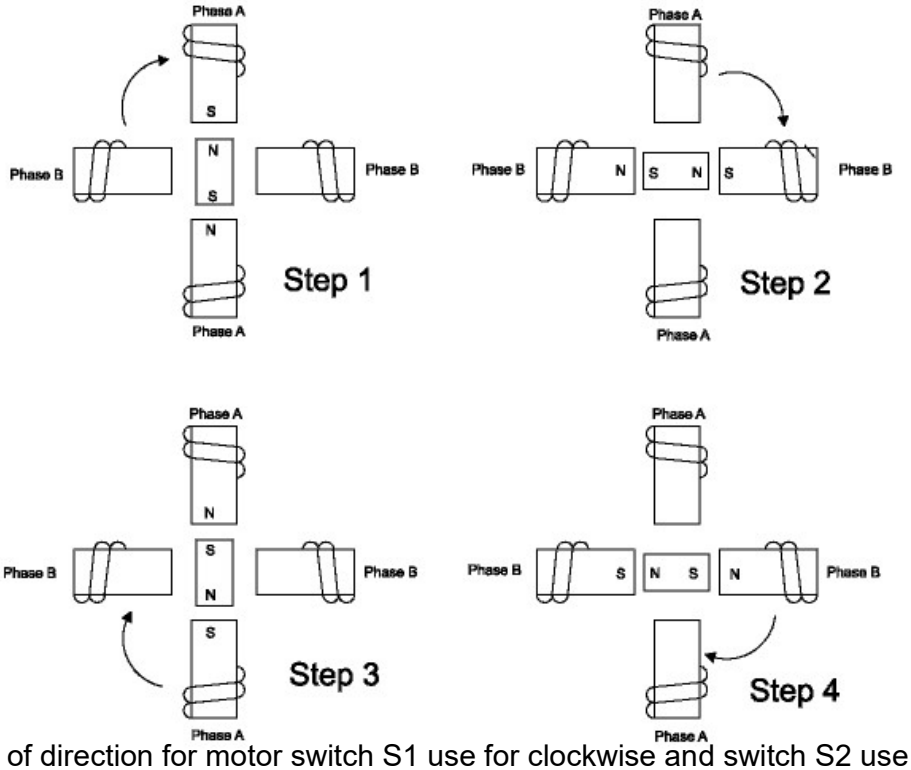
Develop the logic to drive the stepper motor through cyclic interrupt (OEL)

OBJECTIVE:

- To develop the logic in STL to drive the stepper motor.
- To simulate the logic & to download the program in PLC and check the results in real time

TASK:

Develop a logic for the stepper motor in such a way that the motor could rotate in both clockwise & anticlockwise directions with 1-phase excitation.



Selection of direction for motor switch S1 use for clockwise and switch S2 use for counter clockwise direction.

DELIVERABLES:

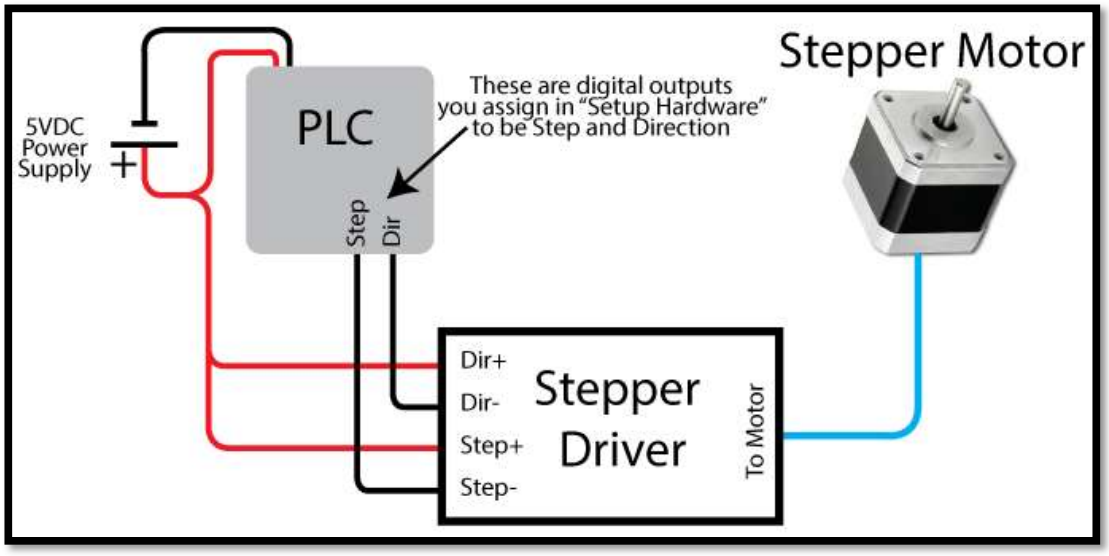
- Develop a logic and show the processing in real time.
- Prepare a report with complete detail and attach with this task.

INTRODUCTION:

A stepper motor is a type of electric motor that moves in precise steps rather than rotating continuously. They are commonly used in CNC machines, robotics, and automation systems. Siemens' Step 7 Simatic Manager is a software program used for programming and configuring PLCs, including those used to control stepper motors. With Step 7 Simatic Manager, engineers can create and edit programs, simulate and test them, and download them to the PLC for execution. Understanding the principles of stepper motor operation and control is essential to effectively use this software for stepper motor control.

THEORY:

A stepper motor is a type of electric motor that rotates in small and precise steps, rather than continuously. It is widely used in various applications, including CNC machines, robotics, and automation systems. The operation of a stepper motor is controlled by a sequence of electrical pulses applied to its windings, which causes the motor shaft to rotate by a fixed angle or step. The number of steps per revolution depends on the motor's construction and can vary from a few steps to thousands of steps per revolution. In order to control a stepper motor, you need a stepper motor driver, which is an electronic device that converts electrical signals from a controller, such as a programmable logic controller (PLC), into the appropriate voltage and current levels for the motor. Step 7 Simatic Manager is a software program used for programming and configuring Siemens PLCs, including those used to control stepper motors. With Step 7 Simatic Manager, you can create and edit programs, simulate and test them, and download them to the PLC for execution. To control a stepper motor using Step 7 Simatic Manager, you will need to program the PLC to generate the appropriate sequence of electrical pulses to the motor driver. This is typically done using a specialized instruction called a pulse train output (PTO) instruction, which generates a series of pulses with precise timing and duration. The PTO instruction can



be configured to control various parameters of the pulse train, including the pulse frequency, pulse width, and direction of rotation. By adjusting these parameters, you can control the speed, acceleration, and direction of the stepper motor. In addition to the PTO instruction, Step 7 Simatic Manager provides a range of other instructions and functions that can be used to program and control stepper motors, including move instructions, interrupt service routines, and data transfer functions. Overall, using Step 7 Simatic Manager to control a stepper motor requires a good understanding of both the software and the hardware involved, as well as the principles of stepper motor operation and control. However, with proper training and experience, it is a powerful and flexible tool for developing and implementing stepper motor control systems.

PROCEDURE:

Here is a general procedure for programming a stepper motor using Step 7 Simatic Manager:

1. Connect the stepper motor to the motor driver and connect the motor driver to the PLC. Make sure all connections are secure and properly grounded.
2. Open Step 7 Simatic Manager and create a new project for your stepper motor control system.
3. Configure the PLC hardware settings and I/O devices to match your motor and motor driver specifications.
4. Write a program using ladder logic or other programming language supported by Step 7 Simatic Manager. The program should include a PTO instruction that generates the necessary pulse train to drive the stepper motor.
5. Configure the PTO instruction to control the speed, acceleration, and direction of the stepper motor. You can adjust various parameters such as pulse frequency, pulse width, and step count to achieve the desired motion.
6. Test the program using the simulation tools provided by Step 7 Simatic Manager. Make sure the motor moves in the expected manner and that there are no errors or warnings in the program.
7. Download the program to the PLC and test the motor in a real-world environment. Make any necessary adjustments to the program or motor driver settings to optimize performance.

SIMATIC MANAGER STEP 7:

- To create a cyclic interrupt using OB35:

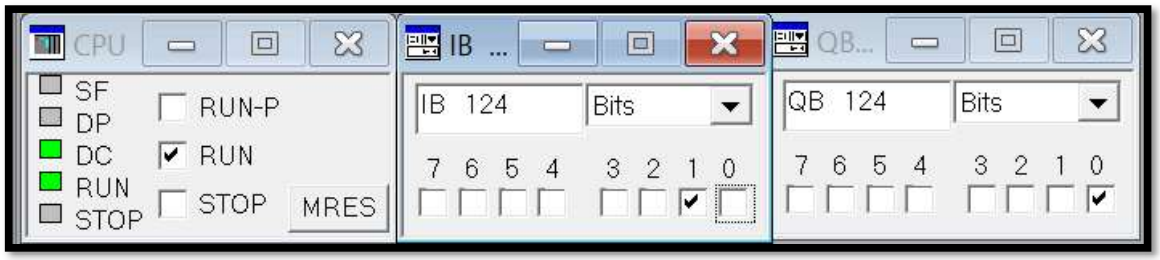
```
OB35 :  "Cyclic Interrupt"
Pushing S1 labelled as CCW to rotate the loaded value in CCW Direction.
Pushing S2 labelled as CW to rotate the loaded value in CW Direction.
Afterwhich
COil-Coil4 Energize.

Network 1: Title:
      A     "CCW"                    I124.1
      JNB   _001
      L     B#16#1
      L     MD     1
      RRD
      T     MD     1
_001: NOP   0

Network 2 : Title:
      A     "CW"                     I124.2
      JNB   _002
      L     B#16#1
      L     MD     1
      RLD
      T     MD     1
_002: NOP   0

Network 3 : Title:
      A     M      3.0
      =     "Coil1"                  Q124.0
```

```
Network 4 : Title:
      A     M      3.1
      =     "Coil2"                  Q124.1

Network 5 : Title:
      A     M      3.2
      =     "Coil3"                  Q124.2

Network 6 : Title:
      A     M      3.3
      =     "Coil4"                  Q124.3
```

- To derive the logic in main using OB1:

```
OB1 :  "Main Program Sweep (Cycle)"
To develop logic to derive stepper motor using OB35

Network 1: Title:
      A     "Start"                  I124.0
      JNB   _001
      L     DW#16#88888888
      T     MD     1
_001: NOP   0
```

SIMULATION RESULTS & VARIABLE TABLE:



	Address	Symbol	Display format	Status value	Modify value
1	I 124.0	"Start"	BOOL	false	
2	I 124.1	"CCW"	BOOL	true	
3	I 124.2	"CW"	BOOL	false	
4	Q 124.0	"Coil1"	BOOL	false	
5	Q 124.1	"Coil2"	BOOL	true	
6	Q 124.2	"Coil3"	BOOL	false	
7	Q 124.3	"Coil4"	BOOL	false	
8	MD 1		HEX	DW#16#22222222	
9	M 3.0		BOOL	false	
10	M 3.1		BOOL	true	
11	M 3.2		BOOL	false	
12	M 3.3		BOOL	false	
13					

CONCLUSION: