Name: Khan Arshad Abdulla **Date – 26/03/2022**
Roll No: 20CO24

Experiment No. 10

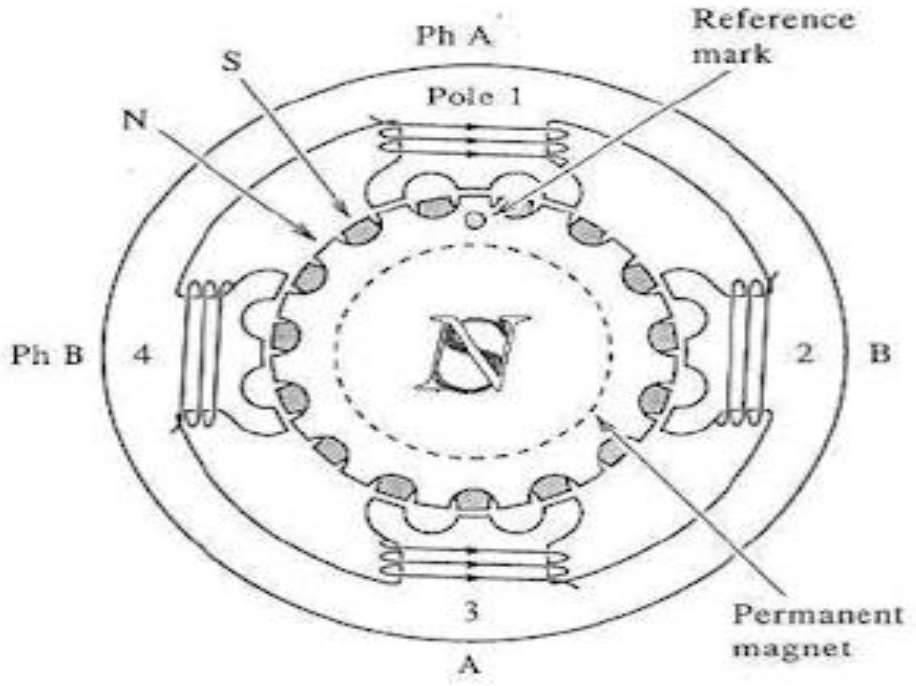
AIM - To Interface Stepper Motor to 8086 using 8255 and write Assembly Language Program to rotate Stepper Motor in Clockwise & Anticlockwise direction.

THEORY:-

Stepper motor is a device used to obtain an accurate position control of rotating shafts. A stepper motor employs rotation of its shaft in terms of steps, rather than continuous rotation as in case of AC or DC motor. To rotate the shaft of the stepper motor, a sequence of pulses is needed to be applied to the windings of the stepper motor, in proper sequence. The numbers of pulses required for complete rotation of the shaft of the stepper motor are equal to the number of internal teeth on its rotor. The stator teeth and the rotor teeth lock with each other to fix a position of the shaft. With a pulse applied to the winding input, the rotor rotates by one teeth position or an angle x. the angle x may be calculated as.

x = 3600 / no. of rotor teeth

After the rotation of the shaft through angle x, the rotor locks it self with the next tooth in the sequence on the internal surface of the stator. The typical schematic of a typical stepper motor with four windings is as shown below.



Cross-section of a two-phase hybrid motor.

The stepper motors have been designed to work with digital circuits. Binary level pulses of 0-5V are required at its winding inputs to obtain the rotation of the shafts. The sequence of the pulses can be decided, depending upon the required motion of the shaft. By suitable sequence of the pulses the motor can be used in three modes of operation.

Name: Khan Arshad Abdulla **Date – 26/03/2022**
Roll No: 20CO24

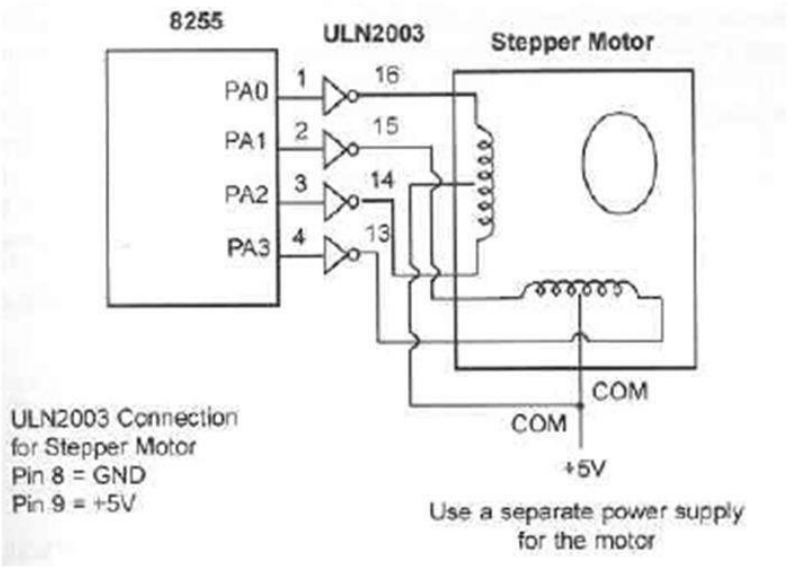
- One phase ON (medium torque)
- Two phase ON (high torque)
- Half stepping (low torque)

Motion	Steps	A	B	C	D	Hex value
Clockwise	1	0	0	1	1	03 H
	2	0	1	1	0	06 H
	3	1	1	0	0	0C H
	4	1	0	0	1	09 H
	5	0	0	1	1	03 H
Anticlockwise	1	0	0	1	1	03 H
	2	1	0	0	1	09 H
	3	1	1	0	0	0C H
	4	0	1	1	0	06 H
	5	0	0	0	0	00 H

WORKING:-
8255 is interfaced with 8086 in I/O mapped I/O. port C (PC0, PC1, PC2, PC3) is used to give pulse sequence to stepper motor. The 8255 provides very less current which will not be able to drive stepper motor coils so each of the winding of a stepper motor needs to be interfaced using high speed switching Darlington transistors with max 1A, 80V rating with heat sink, with the output port of 8255. Output the sequence in correct order to have the desired direction to rotate the motor.

Name: Khan Arshad Abdulla **Date – 26/03/2022**
Roll No: 20CO24

8255 Connection to Stepper Motor



PROGRAM -

```
; this is an example of out instruction.
; it writes values to virtual i/o port
; that controls the stepper-motor.
; c:\emu8086\devices\stepper_motor.exe is on port 7
#start=stepper_motor.exe#
name "stepper"
#make_bin#
steps_before_direction_change = 20h ; 32 (decimal)
jmp start

; ========= data ===============

; bin data for clock-wise
; half-step rotation:
datcw    db 0000_0110b
         db 0000_0100b
         db 0000_0011b
         db 0000_0010b

; bin data for counter-clock-wise
; half-step rotation:
datccw   db 0000_0011b
         db 0000_0001b
         db 0000_0110b
         db 0000_0010b
```

Name: Khan Arshad Abdulla **Date – 26/03/2022**
Roll No: 20CO24

```
; bin data for clock-wise
; full-step rotation:
datcw_fs db 0000_0001b
         db 0000_0011b
         db 0000_0110b
         db 0000_0000b

; bin data for counter-clock-wise
; full-step rotation:
datccw_fs db 0000_0100b
          db 0000_0110b
          db 0000_0011b
          db 0000_0000b

start:
mov bx, offset datcw ; start from clock-wise half-step.
mov si, 0
mov cx, 0 ; step counter
next_step:
; motor sets top bit when it's ready to accept new command
wait:   in al, 7
        test al, 10000000b
        jz wait
mov al, [bx][si]
out 7, al
inc si
cmp si, 4
jb next_step
mov si, 0
inc cx
cmp cx, steps_before_direction_change
jb  next_step
mov cx, 0
add bx, 4 ; next bin data
cmp bx, offset datccw_fs
jbe next_step
mov bx, offset datcw ; return to clock-wise half-step.
jmp next_step
```

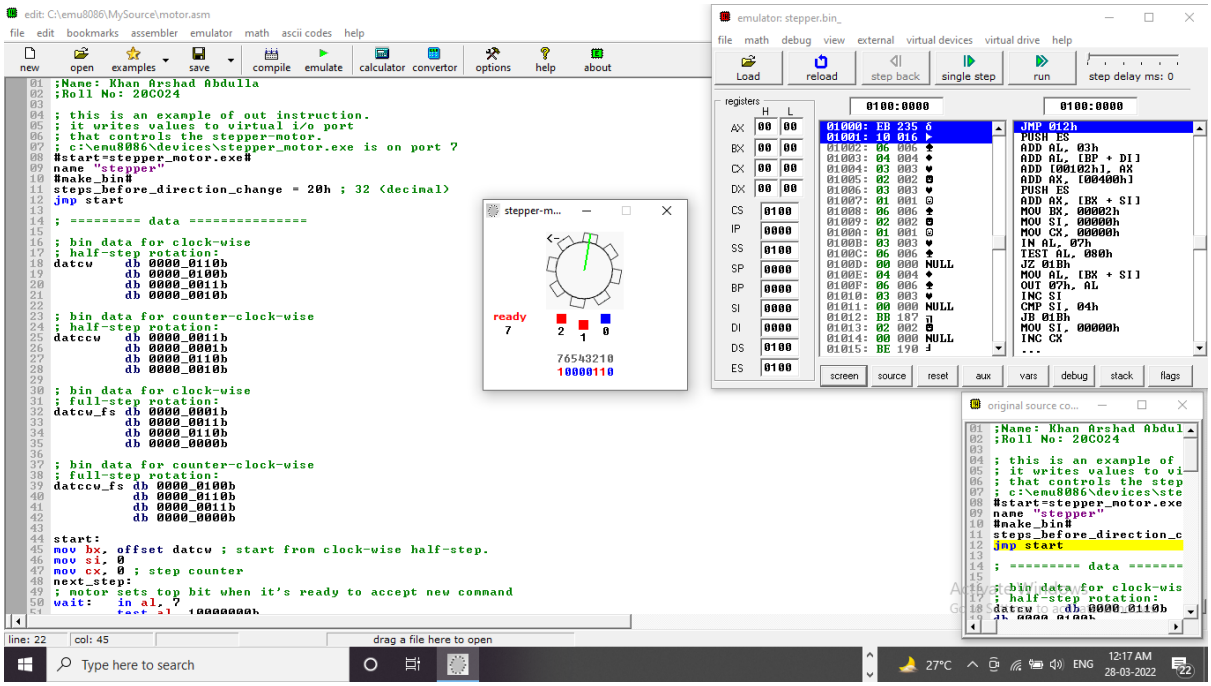
Procedure –

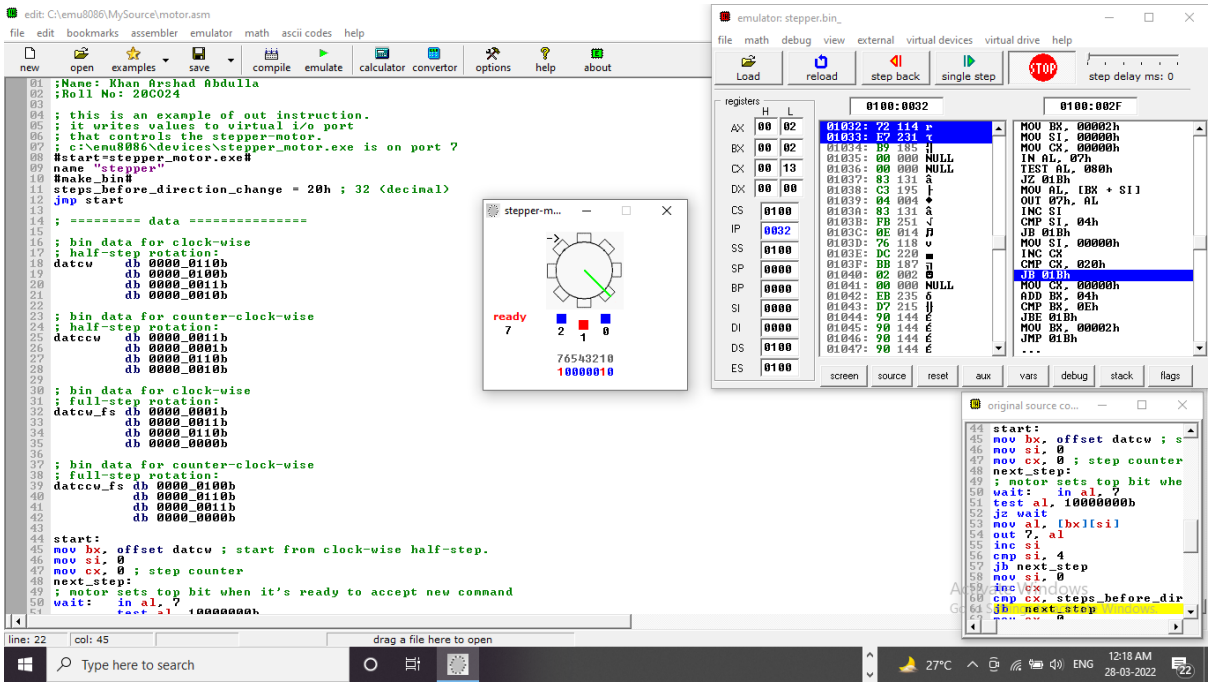
1. **Launch emu8086 IDE** from menu.
2. **Edit** your program , save as file_name.asm
3. **Compile** your program to check for syntax errors, rectify if any error is present. Save and recompile your program.
4. **Run** to observe output of your program.

Output –

Name: Khan Arshad Abdulla **Date – 26/03/2022**

Roll No: 20CO24





Conclusion – To Interface Stepper Motor to 8086 using 8255 and write Assembly Language Program to rotate Stepper Motor in Clockwise & Anticlockwise direction, we use Instructions like TEST, JMP, JBE, etc.

----------end----------