Islamic University of Technology (IUT)
Organization of Islamic Cooperation (OIC)
Department of Electrical and Electronic Engineering (EEE)

Report Submission Date: 17th Oct, 2019

Course : EEE 4606 (Microcontroller Based System Design Lab)

Project No. : 04
Project Name: Stepper Motor Control.

Objective:

1. Familiarization with the AT89C52 Microcontroller.
2. Familiarization with the 4x4 Matrix Keypad.
3. Familiarization with the 16x2 LCD module (LM016L)
4. Familiarization with the ULN2003A motor driver IC.
5. Familiarization with the 28BYJ-48 Stepper Motor.

Section	: C1	
Group	: 02	
Group Members	:	
	160021163	Md. Mohi Uddin Khan
	160021135	Md. Robiul Ferdous

OPERATION

In 4x4 Keypad, user should press any of the 16 keys to rotate Stepper motor according to following table:

Key	Clockwise Rotation
1	45 degree
2	90 degree
3	135 degree
4	180 degree
5	225 degree
6	270 degree
7	315 degree
8	360 degree

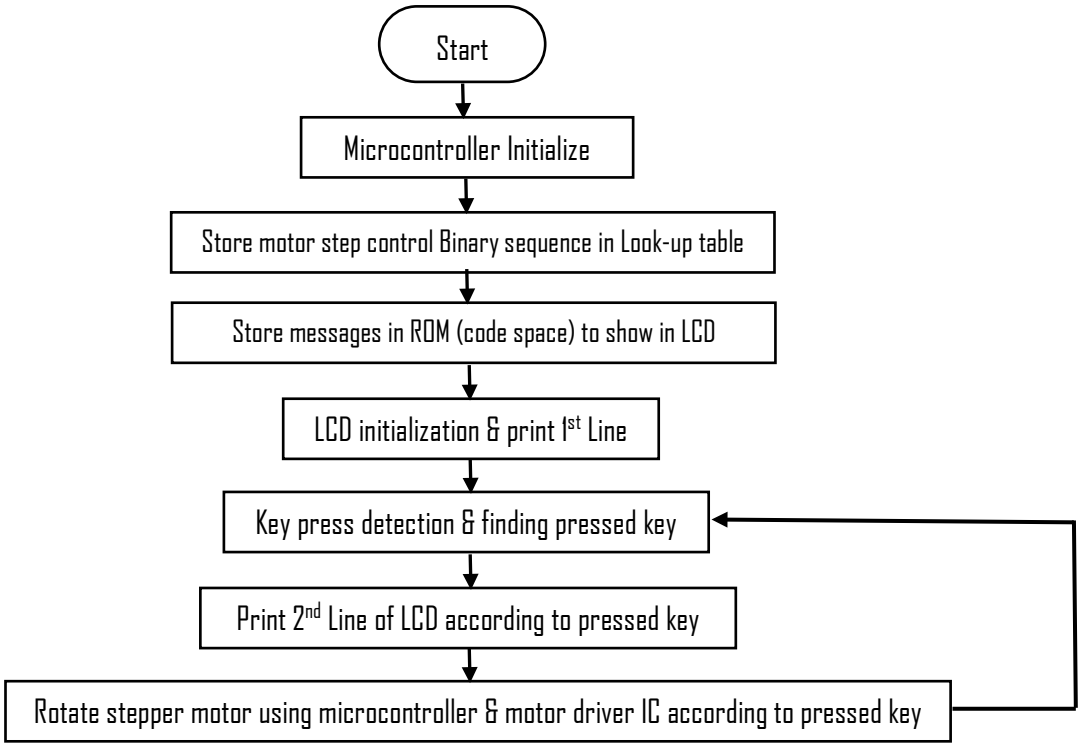
Key	Anti-Clockwise Rotation
9	45 degree
10	90 degree
11	135 degree
12	180 degree
13	225 degree
14	270 degree
15	315 degree
16	360 degree

e.g: User pressed key-2 & motor rotated to +90 degree. Now if user press key-9, motor will rotate -45 degree from +90 degree position; so, now motor is at 90 - 45=45 degree position. Now, if user press key-11, motor will be at 45 – 135 = -90 degree position. Pressing key-2 will turn motor to -90+90 = 0 degree position.

Corresponding rotation angle is displayed on LCD panel.

ALGORITHM

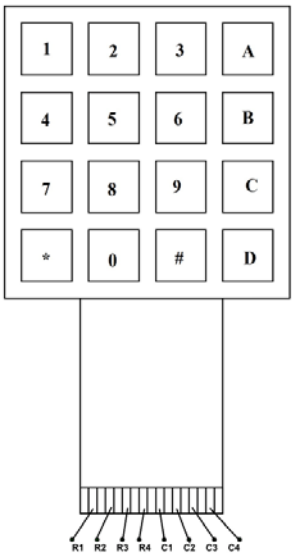
The program is designed according to the following Flow Chart :

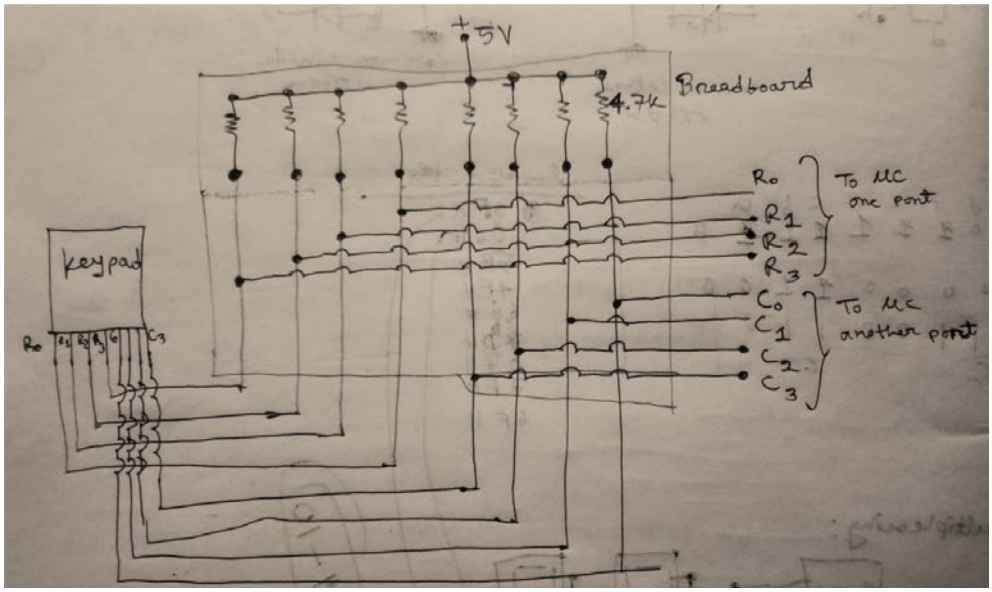


Components Used

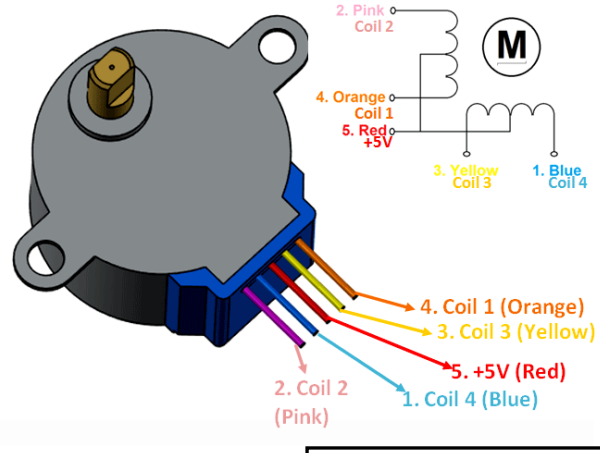
- AT89C52 microcontroller
- Resistors
- Capacitors
- 4x4 matrix keypad
- 16x2 LCD module (LM016L)
- ULN2003A motor driver IC
- 5Volt 28BYJ-48 Micro-Stepper Motor

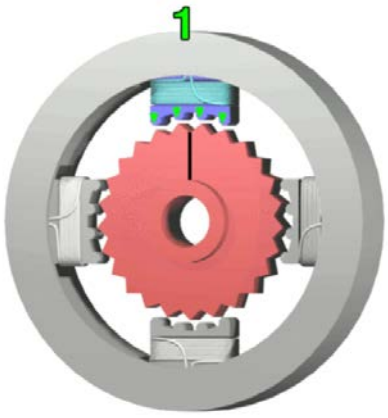
Supporting Figures



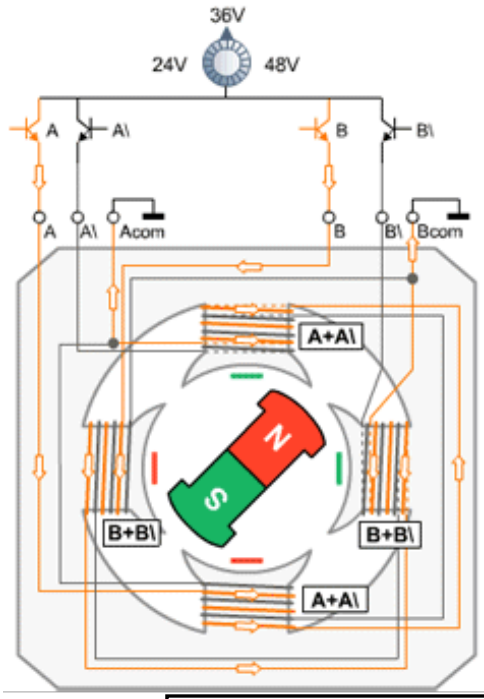


Keypad Hardware connection using Pull-up resistors





Micro-Stepper Motor & it's Pinout diagram



6 Lead Unipolar Driver

Unipolar control is the most simple and cost-effective way to drive a stepper motor, but results in approximately 30% less torque in comparison to the nowadays widely used bipolar drivers. Since the cost advantage is very small today due to cheap integrated circuits, bipolar drivers are now used in most new applications.

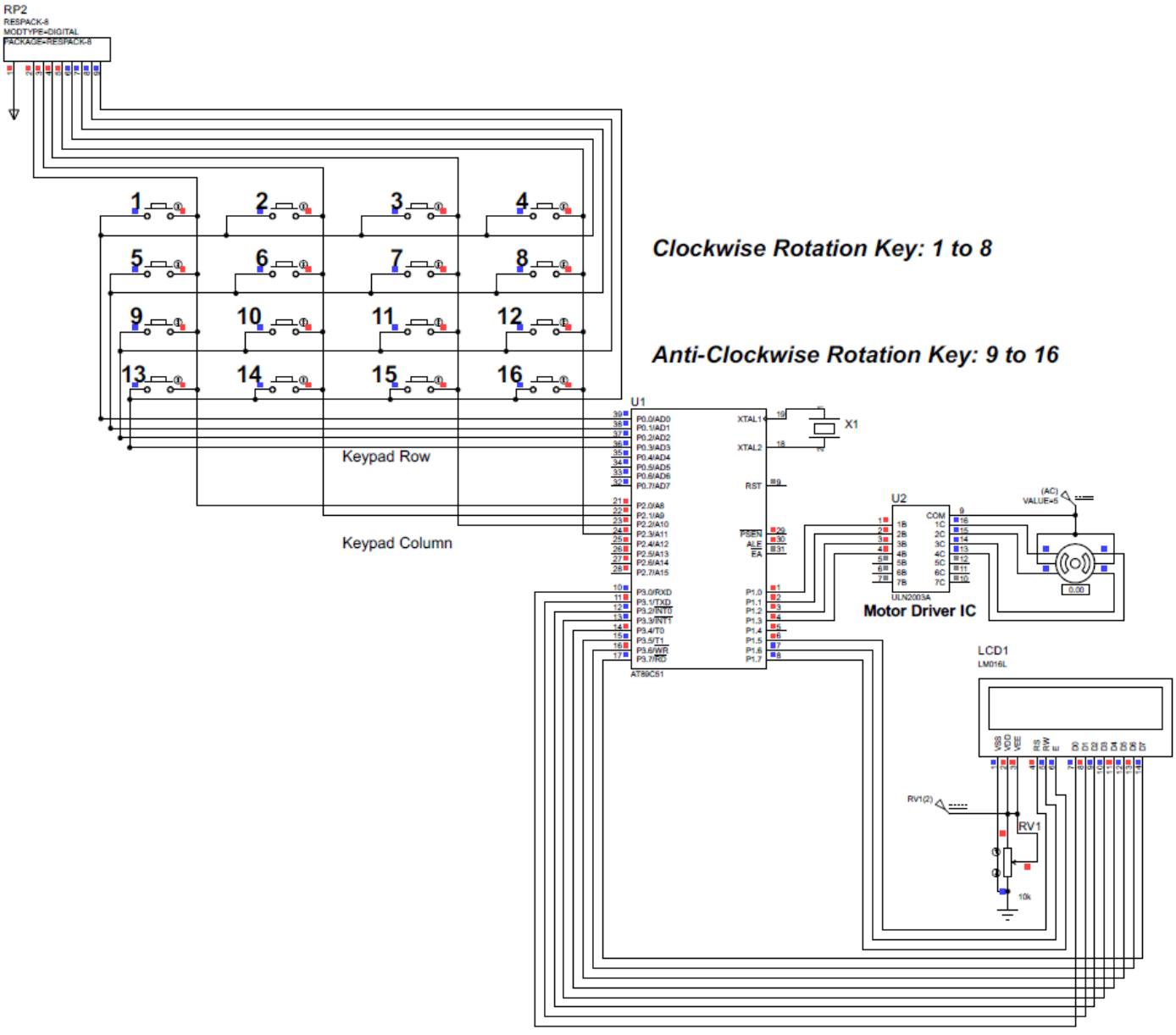
Stepmode								
F	0		1		2		3	
H	0	1	2	3	4	5	6	7
A	1	0	0	0	0	0	1	1
B	1	1	1	0	0	0	0	0
A\	0	0	1	1	1	0	0	0
B\	0	0	0	0	1	1	1	0
dez	12	4	6	2	3	1	9	8

Step mode of Stepper motor & Binary Sequences

- To rotate micro-stepper motor to following angles, we need to execute previous page's half-stepmode binary sequence <ABA'B'> 'Total Steps needed' times in a nested-loop, where particular <ABA'B'> is considered as a single step.

Rotation Angle	Total Steps needed = 61H memory data * 62H memory data * 63H memory data [e.g: 2*32*8 = 510]	Loop Counter Registers' Loaded Values		
		61H byte memory	62H byte memory	63H byte memory
45 degree	510	2D	32D	8D
90 degree	1020	2D	64D	8D
135 degree	1530	2D	96D	8D
180 degree	2040	2D	128D	8D
225 degree	2550	2D	160D	8D
270 degree	3060	2D	192D	8D
315 degree	3570	2D	224D	8D
360 degree	4080	2D	255D	8D

Circuit Diagram in PROTEUS



ASSEMBLY CODE

```
ORG 00H
STEPPER EQU P1
LCD    EQU P3
RS     EQU P1.5
RW     EQU P1.6
EN     EQU P1.7

;STORING BINARY SEQUENCE OF STEPPER MOTOR
IN ROM
MOV R0,#51H    ; INDIRECT ADRESSING MODE
CAN BE USED BY ONLY R0 OR R1 REGISTER

MOV @R0,#00001001B
INC R0
MOV @R0,#00001000B
INC R0
MOV @R0,#00001100B
INC R0
MOV @R0,#00000100B
INC R0
MOV @R0,#00000110B
INC R0
MOV @R0,#00000010B
INC R0
MOV @R0,#00000011B
INC R0
MOV @R0,#00000001B

; LCD INITIALIZATION
MOV DPTR,#MSG_1
C1:CLR A
   MOVC A,@A+DPTR
   ACALL COMNWRT
   ACALL DELAY
   INC DPTR
   JZ  SEND_DAT
   SJMP C1

SEND_DAT:MOV DPTR,#MSG_2
D1:      CLR A
         MOVC A,@A+DPTR
         JZ KEYPAD
         ACALL DATAWRT
         ACALL DELAY
         INC DPTR
         SJMP D1

KEYPAD:
MOV P2,#0FFH
K1:  MOV P0,#0
     MOV A,P2
     ANL A,#00001111B
     CJNE A,#00001111B,K1
K2:  LCALL DELAY
     MOV A,P2
     ANL A,#00001111B
     CJNE A,#00001111B,OVER
     SJMP K2
 OVER:LCALL DELAY
      MOV A,P2
      ANL A,#00001111B
      CJNE A,#00001111B,OVER1
      SJMP K2
OVER1: MOV P0,#11111110B
      MOV A,P2
      ANL A,#00001111B
      CJNE A,#00001111B,ROW_0
      MOV P0,#11111101B
      MOV A,P2
      ANL A,#00001111B
      CJNE A,#00001111B,ROW_1
      MOV P0,#11111011B
      MOV A,P2
      ANL A,#00001111B
      CJNE A,#00001111B,ROW_2
      MOV P0,#11110111B
      MOV A,P2
      ANL A,#00001111B
      CJNE A,#00001111B,ROW_3
      LJMP K2
ROW_0:MOV DPTR,#KCODE0
      SJMP FIND
ROW_1:MOV DPTR,#KCODE1
      SJMP FIND
ROW_2:MOV DPTR,#KCODE2
      SJMP FIND
ROW_3:MOV DPTR,#KCODE3
      SJMP FIND
FIND:RRC A
      JNC MATCH
      INC DPTR
      SJMP FIND
MATCH:CLR A
      MOVC A,@A+DPTR
      MOV R7,A
;CODE FOR MICRO-STEPPER MOTOR
MOTOR:
MOV A,#0C0H    ; LCD CURSOR TO 2ND LINE
ACALL COMNWRT
ACALL DELAY

       M_1:   CJNE R7,#01D,M_2
              MOV DPTR,#MSG_3
              D3:      CLR A
              MOVC A,@A+DPTR
              JZ E_1
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D3
              E_1: MOV 60H,#32D   ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD
```

```
       M_2:
              CJNE R7,#02D,M_3

              MOV DPTR,#MSG_4
              D4:      CLR A
              MOVC A,@A+DPTR
              JZ E_2
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D4

              E_2: MOV 60H,#64D     ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD
       M_3:
              CJNE R7,#03D,M_4

              MOV DPTR,#MSG_5
              D5:      CLR A
              MOVC A,@A+DPTR
              JZ E_3
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D5

              E_3: MOV 60H,#96D     ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD
       M_4:
              CJNE R7,#04D,M_5

              MOV DPTR,#MSG_6
              D6:      CLR A
              MOVC A,@A+DPTR
              JZ E_4
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D6

              E_4: MOV 60H,#128D    ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD
       M_5:
              CJNE R7,#05D,M_6

              MOV DPTR,#MSG_7
              D7:      CLR A
              MOVC A,@A+DPTR
              JZ E_5
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D7

              E_5: MOV 60H,#160D    ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD
```

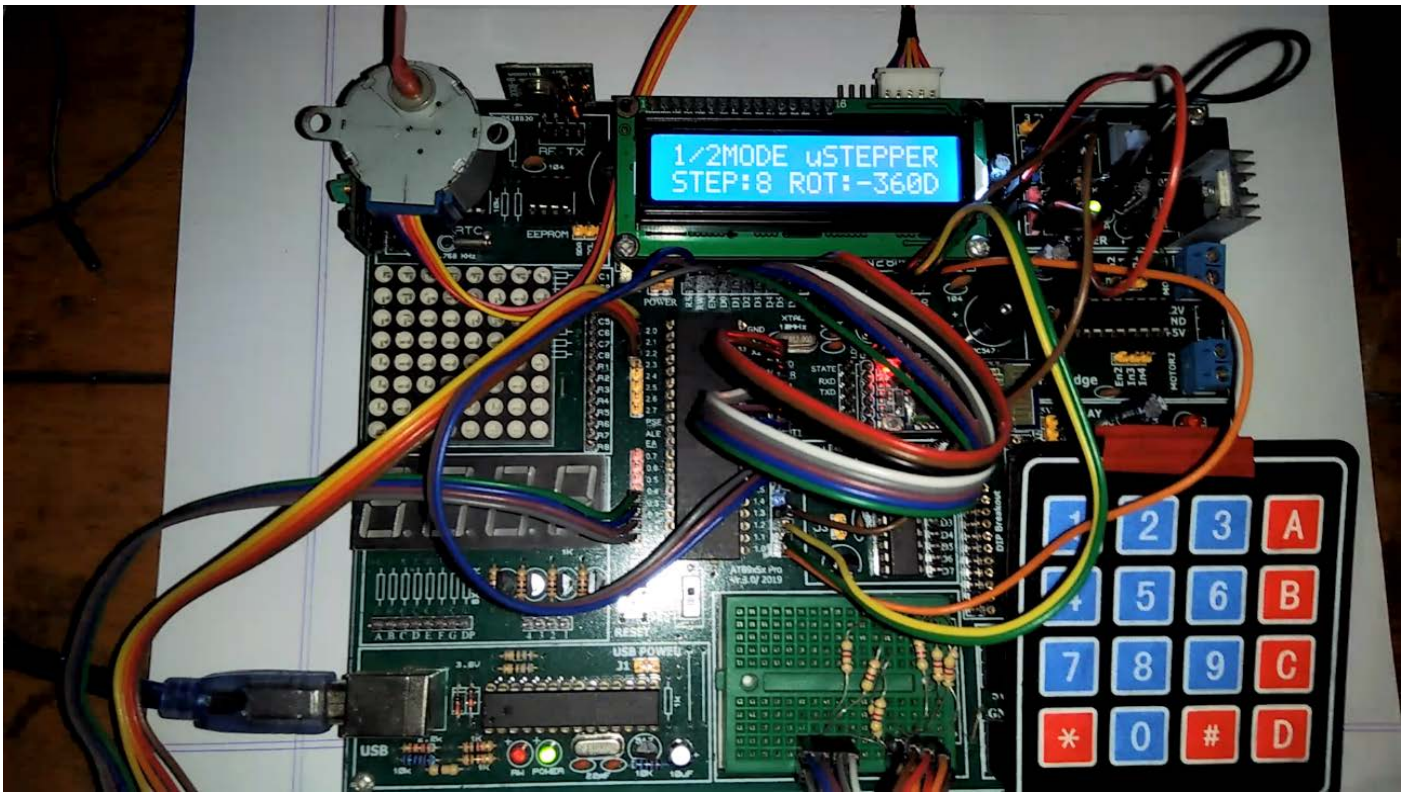
```
       M_6:
              CJNE R7,#06D,M_7

              MOV DPTR,#MSG_8
              D8:      CLR A
              MOVC A,@A+DPTR
              JZ E_6
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D8

              E_6: MOV 60H,#192D    ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD

       M_7:
              CJNE R7,#07D,M_8

              MOV DPTR,#MSG_9
              D9:      CLR A
              MOVC A,@A+DPTR
              JZ E_7
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D9

              E_7: MOV 60H,#224D    ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD

       M_8:
              CJNE R7,#08D,M_9

              MOV DPTR,#MSG_10
              D10:      CLR A
              MOVC A,@A+DPTR
              JZ E_8
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP D10

              E_8: MOV 60H,#255D    ;COUNTER
FOR MOTOR ROTATION LOOP
                   LCALL CLOCKWISE_ROTATION
                   LJMP ENDD

       M_9:   CJNE R7,#09D,M_10

              MOV DPTR,#MSG_M3
              DM3:      CLR A
              MOVC A,@A+DPTR
              JZ E_M1
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM3
```

```
            E_M1: MOV 60H,#32D    ;COUNTER
FOR MOTOR ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD
       M_10:
              CJNE R7,#10D,M_11

              MOV DPTR,#MSG_M4
              DM4:      CLR A
              MOVC A,@A+DPTR
              JZ E_M2
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM4

              E_M2: MOV 60H,#64D    ;COUNTER
FOR MOTOR ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD
       M_11:
              CJNE R7,#11D,M_12

              MOV DPTR,#MSG_M5
              DM5:      CLR A
              MOVC A,@A+DPTR
              JZ E_M3
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM5

              E_M3: MOV 60H,#96D    ;COUNTER
FOR MOTOR ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD
       M_12:
              CJNE R7,#12D,M_13

              MOV DPTR,#MSG_M6
              DM6:      CLR A
              MOVC A,@A+DPTR
              JZ E_M4
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM6

              E_M4: MOV 60H,#128D   ;COUNTER
FOR MOTOR ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD
       M_13:
              CJNE R7,#13D,M_14

              MOV DPTR,#MSG_M7
              DM7:      CLR A
              MOVC A,@A+DPTR
              JZ E_M5
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM7
```

```
       E_M5: MOV 60H,#160D   ;COUNTER FOR MOTOR
ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD

       M_14:
              CJNE R7,#14D,M_15

              MOV DPTR,#MSG_M8
              DM8:      CLR A
              MOVC A,@A+DPTR
              JZ E_M6
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM8

              E_M6: MOV 60H,#192D   ;COUNTER
FOR MOTOR ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD

       M_15:
              CJNE R7,#15D,M_16

              MOV DPTR,#MSG_M9
              DM9:      CLR A
              MOVC A,@A+DPTR
              JZ E_M7
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM9

              E_M7: MOV 60H,#224D   ;COUNTER
FOR MOTOR ROTATION LOOP
                  LCALL ANTICLOCKWISE_ROTATION
                  LJMP ENDD

       M_16:
              CJNE R7,#16D,E_M9

              MOV DPTR,#MSG_M10
              DM10:      CLR A
              MOVC A,@A+DPTR
              JZ E_M8
              ACALL DATAWRT
              ACALL DELAY
              INC DPTR
              SJMP DM10

              E_M8:  MOV 60H,#255D
       ;COUNTER FOR MOTOR ROTATION LOOP
                     LCALL
ANTICLOCKWISE_ROTATION
              E_M9:  LJMP ENDD
```

```
COMNWRT: MOV  LCD,A
         CLR  RS
         CLR  RW
         SETB EN
         ACALL DELAY
         CLR  EN
         RET

DATAWRT: MOV  LCD,A
         SETB RS
         CLR  RW
         SETB EN
         ACALL DELAY
         CLR  EN
         RET

; Delay time needs to be sufficiently small
to rotate motor, motor won't rotate in case
of too small or large delay.

DELAY:MOV R1,#1
      L3:MOV R2,#5
      L2:MOV R3,#100
      L1:DJNZ R3,L1
         DJNZ R2,L2
         DJNZ R1,L3
         RET

CLOCKWISE_ROTATION:
                  MOV 61H,#2D
             LP1: MOV 62H,60H
             LP2: MOV 63H,#8D
                  MOV R0,#51H    ; INDIRECT
ADRESSING MODE CAN BE USED BY ONLY R0 OR R1
REGISTER
             LP3: MOV STEPPER,@R0
                  INC R0
                  LCALL DELAY
                  DJNZ 63H,LP3

                  DJNZ 62H,LP2
                  DJNZ 61H,LP1
                  RET

ANTICLOCKWISE_ROTATION:
                  MOV 61H,#2D
             LM1: MOV 62H,60H
             LM2: MOV 63H,#8D
                  MOV R0,#58H    ; INDIRECT
ADRESSING MODE CAN BE USED BY ONLY R0 OR R1
REGISTER
             LM3: MOV STEPPER,@R0
                  DEC R0
                  LCALL DELAY
                  DJNZ 63H,LM3
                  DJNZ 62H,LM2
                  DJNZ 61H,LM1
                  RET

KCODE0: DB 1D,2D,3D,4D
KCODE1: DB 5D,6D,7D,8D
KCODE2: DB 9D,10D,11D,12D
KCODE3: DB 13D,14D,15D,16D

MSG_1:   DB 38H,0EH,01,06,80H,0
MSG_2:   DB '1/2MODE uSTEPPER',0
MSG_3:   DB 'STEP:1  ROT:45 D',0
MSG_4:   DB 'STEP:2  ROT:90 D',0
MSG_5:   DB 'STEP:3 ROT:135 D',0
MSG_6:   DB 'STEP:4 ROT:180 D',0
MSG_7:   DB 'STEP:5 ROT:225 D',0
MSG_8:   DB 'STEP:6 ROT:270 D',0
MSG_9:   DB 'STEP:7 ROT:315 D',0
MSG_10:  DB 'STEP:8 ROT:360 D',0
MSG_M3:  DB 'STEP:1  ROT:-45D',0
MSG_M4:  DB 'STEP:2  ROT:-90D',0
MSG_M5:  DB 'STEP:3 ROT:-135D',0
MSG_M6:  DB 'STEP:4 ROT:-180D',0
MSG_M7:  DB 'STEP:5 ROT:-225D',0
MSG_M8:  DB 'STEP:6 ROT:-270D',0
MSG_M9:  DB 'STEP:7 ROT:-315D',0
MSG_M10: DB 'STEP:8 ROT:-360D',0

ENDD:
       LJMP KEYPAD

       END
```

Hardware Connection



- **Hardware Connection & Output Video:**
  Scan or Click on the QR code to access the video from Google Drive –

- **Web Reference:**
  1.http://www.8051projects.net/wiki/Stepper_Motor_Tutorial

  2. https://www.seeedstudio.com/blog/2019/03/04/driving-a-28byj-48-stepper-motor-with-a-uln2003-driver-board-and-arduino/