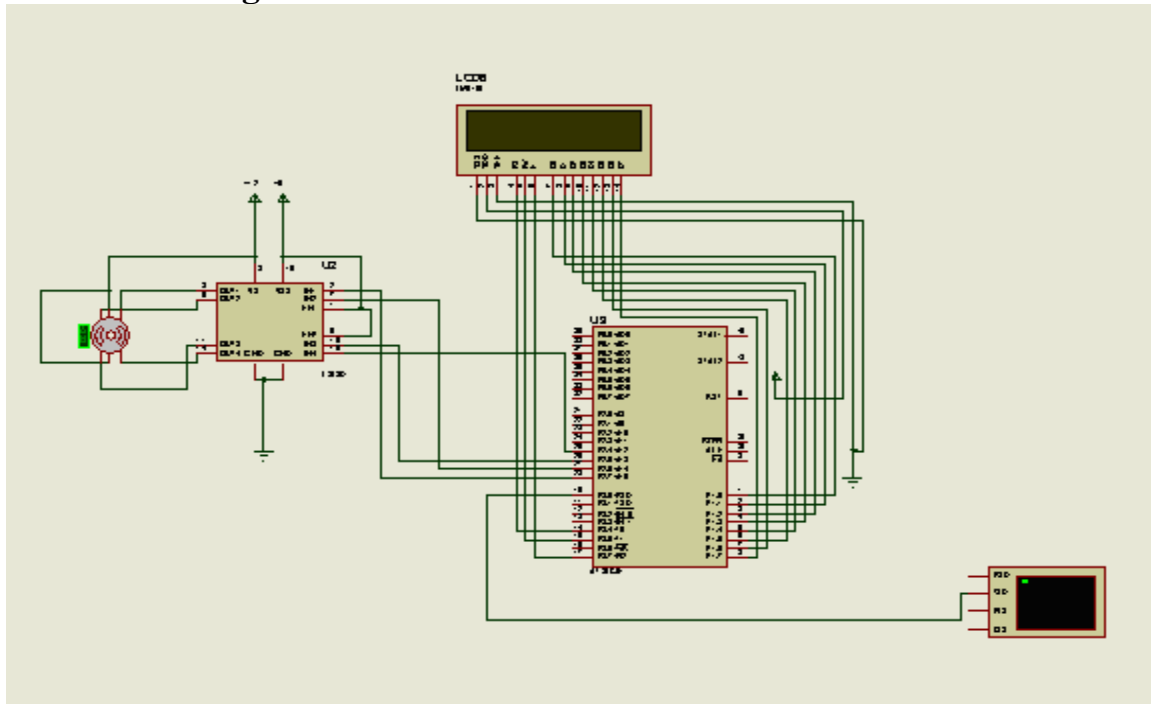


**Project Name:** Rotate Stepper motor in degree where command given from keyboard.

**Solution:** To complete the problem we implement hardware design and write hex code using MIDE-51 and run the stepper motor in different angle.

### Hardware design:



### Software Algorithm:

- Write the software code in mide-51 save it as .asm.
- Compile the F/W by Build.
- Start Proteus and pick from libraries AT89c51 Microcontroller **Start timer**
- Then picked a stepper motor from library connect it in output of I93D taken from library and its input is connected to port 1.
- Click on the AT89c51 and load the program file or hex file
- Play or Run the Proteus file
- Rotation is controlled by the input command

## Software code:

```
MOV    SP,0E0H                ;INITIALIZING STACK POINTER
ORG    00H
MOV    TMOD,#20H              ;TIMER 1, MODE 2
MOV    TH1,#-3h
                                ;9600 BAUD @ 11.0592MHZ
MOV    SCON,#50h
                                ;8 BIT, 1 STOP BIT
SETB   TR1
ACALL  DELAY1                  ;START TIMER 1

SCAN:
JNB    RI,SCAN
ACALL  DELAY23
ACALL  DELAY23
ACALL  DELAY23
ACALL  DELAY23
ACALL  DELAY23
ACALL  DELAY23                ;WAIT FOR CHAR. TO COME IN

MOV    B,SBUF                  ; ;SAVE INCOME BYTE IN B

ACALL  DELAY1

                                CLR RI

                                ;TO SHOW THAT WE RECEIVED THE BYTE
MOV    A,#38H
                                ;INIT LCD 2 LINES 5X7 MATRIX
ACALL  COMMAND
ACALL  DELAY1
                                ;ISSUE COMMAND
MOV    A,#0EH
                                ;LCD ON, CURSOR ON
ACALL  COMMAND
ACALL  DELAY1
                                ;ISSUE COMMAND
MOV    A,#01H
                                ;CLEAR LCD COMMAND
ACALL  COMMAND
ACALL  DELAY1
                                ;ISSUE COMMAND
MOV    A,#01H
                                ;SHIFT CURSOR RIGHT
ACALL  COMMAND
ACALL  DELAY1
                                ;ISSUE COMMAND
MOV    A,#80H
                                ;CURSOR: LINE 1, POS.6
ACALL  COMMAND
ACALL  DELAY1
                                ;ISSUE COMMAND
MOV    A,B
MOV    R7,A
                                ;PC DATA BYTE
ACALL  DATA_DISP
ACALL  DELAY1
ACALL  TRANSMIT

                                ;DISPLAY DATA
```

SJMP SCAN

;FETCH NEXT BYTE

COMMAND:

ACALL READY

ACALL DELAY1

;CHECK IF LCD READY

MOV P1,A

ACALL DELAY1

;ISSUE COMMAND CODE

CLR P3.4

ACALL DELAY1

;RS=0 FOR COMMAND

CLR P3.5

ACALL DELAY1

;R/W=0 TO WRITE TO LCD

SETB P3.7

ACALL DELAY1

;E=1 FOR H TO L PULSE

CLR P3.7

ACALL DELAY1

;E=0 LATCH IN

RET

;RETURN TO CALLER

DATA\_DISP:

ACALL READY

ACALL DELAY1

;CHECK IF LCD READY

MOV P1,A

ACALL DELAY1

ACALL DELAY1

;ISSUE DATA

SETB P3.4

ACALL DELAY1

;RS=1 FOR DATA

CLR P3.5

ACALL DELAY1

;R/W=0 TO WRITE TO LCD

SETB P3.7

ACALL DELAY1

;E=1 FOR H TO L PULSE

CLR P3.7

;E=0 LATCH IN

RET

;RETURN TO CALLER

READY:

SETB P1.7

;MAKE P1.7 INPUT PORT

CLR P3.4

;RS=0 ACCESS COMMAND REGISTER

SETB P3.5

;R/W=1 READ COMMAND REGISTER

BACK:

CLR P3.7

;E=1 FOR H TO L PULSE

SETB P3.7

;E=0 H TO L PULSE

JB P1.7, BACK

;STAY UNTIL BUSY FLAG=0

RET

DELAY1: MOV R1,#02H

;loads R1=02h

LD2: MOV R5,#0FFH

;loads R5=0FFh

LD1: DJNZ R5,LD1

;decrease R5 until zero

DJNZ R1,LD1

;decrease R1 and jump to LD2 and re-execute

RET

DELAY23:MOV R1,#0FFH

;loads R1=02h

LD4: MOV R5,#0FFH

;loads R5=0FFh

LD3: DJNZ R5,LD3

;decrease R5 until zero

DJNZ R1,LD3

;decrease R1 and jump to LD2 and re-execute

RET

transmit:

mov a,r7

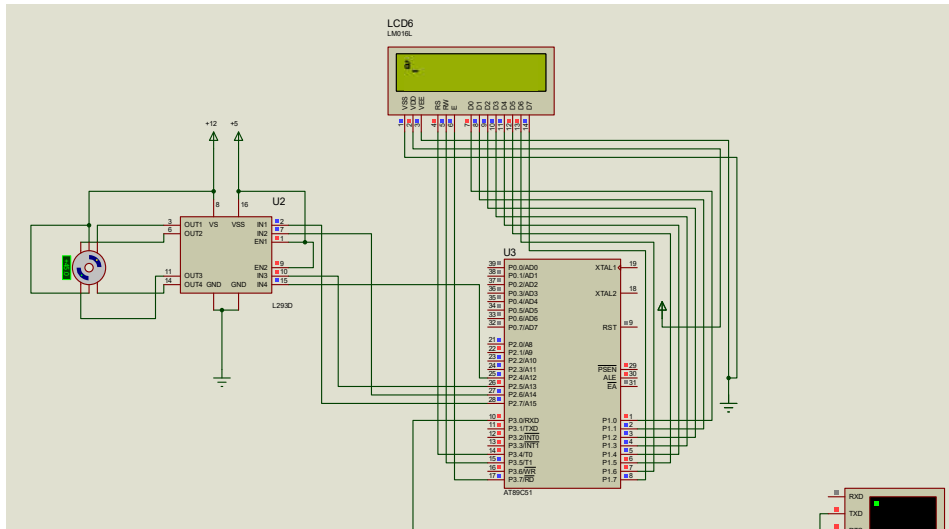
```

mov sbuf,a
clr      ti
CJNE A,#61H,V1
SJMP Rotate1
V1:CJNE A,#62H,V2
SJMP Rotate2
V2:CJNE A,#63H,V3
SJMP Rotate3
V3:CJNE A,#64H,V4
SJMP Rotate4
V4:CJNE A,#65H,V5
SJMP Rotate5
V5:NOP
ret
Rotate1: MOV A, #022H
        MOV P2, A
        ACALL DELAY1
        RR A
        ACALL DELAY1
        RET
Rotate2: MOV A, #033H
        MOV P2, A
        ACALL DELAY1
        RR A
        ACALL DELAY1
        RET
Rotate3: MOV A, #0BBH
        MOV P2, A
        ACALL DELAY1
        RR A
        ACALL DELAY1
        RET
Rotate4: MOV A, #099H
        MOV P2, A
        ACALL DELAY1
        RR A
        ACALL DELAY1
        RET
Rotate5: MOV A, #066H
        MOV P2, A
        ACALL DELAY1
        RR A
        ACALL DELAY1

RET
end                ;RETURN TO CALLER

```

**Result:**  
**When input= a rotation is 45**



## Discussion:

The stepper motor is a form of ac motor that is designed to rotate a specific number of degrees for each electrical pulse applied to its control unit. In this problem we rotate the stepper motor in different angle mode, where angle given from computer. For this we use AT89C51 microcontroller for giving input angle we use serial port and LCD display to see input char. Finally the design and code run properly.