

# 1.STEPPER MOTOR

```
ORG 0000H
MOV SP,#07H
MOV P0,#00H
START:MOV A,#66H
MOV P0,A
CLKW:MOV R0,#3
CLKW1:MOV R1,#200
CLKW2:RR A
ACALL DELAY
MOV P0,A
DJNZ R1,CLKW2
DJNZ R0,CLKW1
ACLKW:MOV R0,#3
ACLKW1:MOV R1,#200
ACLKW2:RL A
ACALL DELAY
MOV P0,A
DJNZ R1,ACLKW2
DJNZ R0,ACLKW1
SJMP START
DELAY:MOV R2,#100
HERE1:MOV R3,#255
HERE2:DJNZ R3,HERE2
DJNZ R2,HERE1
RET
END
```

# 2.SERIAL COMM

```
org 0000h
mov p0,#0ffh
mov tmod,#20h
mov th1,#0fah
mov scon,#50h
setb tr1
mov dptr,#mydata
l1:clr a
movc a,@a+dptr
jz l1
acall send
inc dptr
sjmp l1
send :mov sbuf,a
c1:jnb ti,c1
clr ti
ret

mydata: db "HAPPY_NEW_YEAR",0
end
```

### 3.STEPPER WITH SERIAL COMM

```
org 00h
mov p0,#00h // port 0 as output
// serial comm initiliazation
mov p3, #0fdh // p3.1 rs 232 transmit pin is output
mov TMOD, #20h // timer1, mode 2
mov TH1, 0fdh // 9600 baud rate
mov scon, #50h // 8 bit, 1 stop, ren enable
setb TR1 // start timer
clr TI
start:mov a,#66h
mov p0,a // drive p0
mov r6,a
acall ser // send "a" value to serial
// clockwise operation 2540 steps
clkw: mov r0,#1
clkwl: mov r1,#254
clkw2: rr a // rotate right
acall delay
mov p0,a
mov r6,a
acall ser
djnz r1,clkw2
djnz r0,clkwl
// anti clockwise operation 2540 steps
aclkw: mov r0,#3
aclkw1: mov r1,#254
aclkw2: rl a // rotate left
acall delay
mov p0,a
mov r6,a
acall ser
djnz r1,aclkw2
djnz r0,aclkw1
sjmp start
delay:mov r2,#100
here1:mov r3,#100
here2:djnz r3,here2
djnz r2,here1
ret
// to transmit the lower nibble data over Serial comm
//// if lower nibble <10 add 30h else add 61h //
ser: mov a,r6
mov r7,a
anl a,#0fh
clr c
subb a,#0ah // subtract 10
jc ser1
add a,#61h
sjmp ser2
ser1:mov a,r7
anl a,#0fh
add a,#30h
ser2:mov sbuf,a
mov a,r6
here:jnb TI, here
clr TI
ret
end
```

## 4.LCD

```
org 0000h
    here: mov a,#38h
          acall cmnd
          mov a,#0eh
          acall cmnd
          mov a,#06h
          acall cmnd
          mov a,#01h
          acall cmnd
          mov a,#083h
          acall cmnd

          mov a,"h"
          acall disp
          mov a,"e"
          acall disp
          mov a,"l"
          acall disp
          mov a,"l"
          acall disp
          mov a,"o"
          acall disp

    cmnd: mov p2,a
          clr p3.7
          clr p3.6
          setb p3.5
          acall delay
          clr p3.5
          RET

    disp: MOV P2,A
          setb p3.7
          clr p3.6
          setb p3.5
          acall delay
          clr p3.5
          RET

    delay:mov r3,#50
    here1:mov r4,#255
    here2:djnz r4,here2
    djnz r3,here1
    ret
    END
```

## 5(a).SINE WAVE

```
org 0000h
again:mov dptr,#table
mov r2,#12

back:movc a,@a+dptr
mov p1,a
clr a
inc dptr
djnz r2,back

sjmp again

org 300h
table: db 128,192,238,255,238,192,128,64,17,0,17,64,128
end
```

## 5(b).SQUARE WAVE

```
org 0000h
mov tmod,#01h
mov tl0,#60h
mov th0,#0b8h
here:cpl p1.0
acall delay
sjmp here

delay:setb tr0
c1: jnb tf0,c1
clr tf0
clr tr0
ret
end
```

## 6.KEYPAD

```
mov dptr, #1000h
mov a, #0ffh
mov p0, #00h
start: mov p1, #0ffh
clr p1.0 // row1 clear
jnb p1.4,n1 // check column1 if p1.4=1 go for next column
verification
mov a, #01h
acall display
n1: jnb p1.5, n2 // check column2 if p1.5=1 go for next column
verification
mov a, #02h
acall display
n2: jnb p1.6, n3 //check column3 if p1.6=1 go for next column
verification
mov a, #03h
acall display
n3: jnb p1.7, n4 //check column4 if p1.7=1
mov a, #04h
acall display
n4: setb p1.0// row1 set
clr p1.1// row 2 clear
jnb p1.4,n5 // check column1 if p1.4=1 go for next column
```

```

verification
    mov a, #05h
    acall display
    n5: jb p1.5, n6 // check column2 if p1.5=1 go for next column
verification
    mov a, #06h
    acall display
    n6: jb p1.6, n7 //check column3 if p1.6=1 go for next column
verification
    mov a, #07h
    acall display
    n7: jb p1.7, n8 //check column4 if p1.7=1
    mov a, #08h
    acall display
    n8: setb p1.1 //row 2 set
    clr p1.2 // row3 clear
    jb p1.4,n9 // check column1 if p1.4=1 go for next column
verification
    mov a, #09h
    acall display
    n9: jb p1.5, n10 // check column2 if p1.5=1 go for next column
verification
    mov a, #10
    acall display
    n10: jb p1.6, n11 //check column3 if p1.6=1 go for next column
verification
    mov a, #11
    acall display
    n11: jb p1.7, n12 //check column4 if p1.7=1
    mov a, #12
    acall display
    n12: setb p1.2 //row 3 set
    clr p1.3 // row4 clear
    jb p1.4,n13 // check column1 if p1.4=1 go for next column
verification
    mov a, #13
    acall display
    n13: jb p1.5, n14 // check column2 if p1.5=1 go for next column
verification
    mov a, #14
    acall display
    n14: jb p1.6, n15 //check column3 if p1.6=1 go for next column
verification
    mov a, #15
    acall display
    n15: jb p1.7, n16 //check column4 if p1.7=1
    mov a, #16
    acall display
    n16:ljmp start
display: movc a,@ a+dptr
    mov p0,a
    ret
    org 1000h
    db 00h,0f9h,0a4h,0b0h,88h,99h, 92h,82h,083h,0f8h,80h,98h,0c6h,
0ffh,0ffh,0ffh,0a1h
    end

```

