

## Format of Tmod register

first(lower four bits) =t1(timer1)

second four=t0

if c/t(bar)=0 it acts as timer

if gate bit=1,we r gng to control hardware ckt

if gate bit=0,we r gng to control software ckt

1 machine cycle consist of 12 clock cycles

timer counts how many machine cycles

MODE1 INTIALIZING

gate	c/t-	m1	m0	gate	c/t-	M1	M0
0	0	0	0	0	0	0	1
TCON							TCON
HIGHER	NIBBLE	FOR	TIMER				
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

M1	M0	Mode	Timer
0	0	Mode0	13 bit timer
0	1	Mode1	16 bit timer
1	0	Mode2	8 bit auto load
1	1	Mode 3	8 bit split timer

## SPECIAL FUNCTION REGISTERS:

TL0 AND TH0 (TIMER ZERO LOWER BYTE AND HIGHER BYTE)

TCON(to start timer and to stop timer)

TR0=1 TO START

TR0=0 TO STOP

WHEN OVERFLOW OCCURS TFO BECOMES 1 (TF0=1)

WHEN COUNT EXCEEDS FFFF IT BECOMES COUNT 0 CARRY 1

for delay

ffff-1000=effff delay in terms of machines cycles

ffff-ffff=0000 for more delay

```
mov tmod,#01h
```

```
mov TL0,#00h
```

```
mov TH0,#00h
```

```
goto: mov p2,#0ffh
```

```
acall delay
```

```
mov p2,#00h
acall delay
sjmp goto
delay :setb TR0
here:jnb TF0,here
clr tf0
clr tr0
ret
end
```

## 2.using switch

```
mov tmod,#00h
mov TL0,#00h
mov TH0,#00h
```

```
check:      jnb p1.0,on
             mov p2,#00h(leds off state)
             acall delay
             sjmp check

on:          mov p2,#0ffh
             acall delay
             sjmp check

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

## 2.using switch LED Scrolling

```
mov tmod,#00h
mov TL0,#00h
mov TH0,#00h
mov a,#01h
mov b,10h
check:      jnb p1.0,rolleft
            sjmp rolright

rolleft:    mov p2,@a
            rl a
            acall delay
            jnb p2.7,rolleft
            sjmp check

rolright:   mov p2,@b
            rr b
            acall delay
            jnb p2.0,rolright
            sjmp check

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