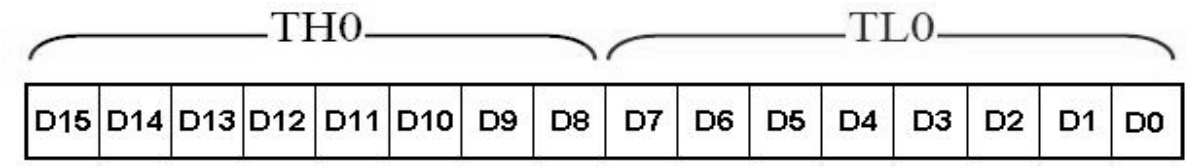
# PROGRAMMING 8051 TIMERS

The 8051 has two timers; timer 0, timer 1. They can be used either as timers or as event counters. Both timer 0 and timer 1 are 16 bits wide. Since the 8051 has an 8-bit architecture , each 16 bit timer is accessed as two separate registers of low byte and high byte.

## TIMER 0 registers

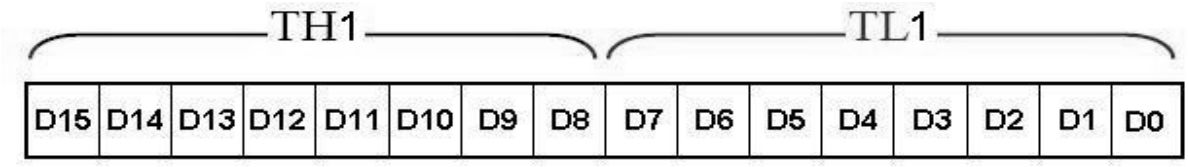
The 16 bit register of timer 0 is accessed as low byte and high byte. The low byte register is called TL0 (timer 0 low byte) and the high byte register is referred to as TH0 ( timer 0 high byte). These registers can be accessed like any other register, such as A, B, R0, R1, R2 etc. For example, the instruction " MOV TL0,#25H" loads the value 25H into TL0.




Fig(1): Timer 0 Registers

## TIMER 1 registers

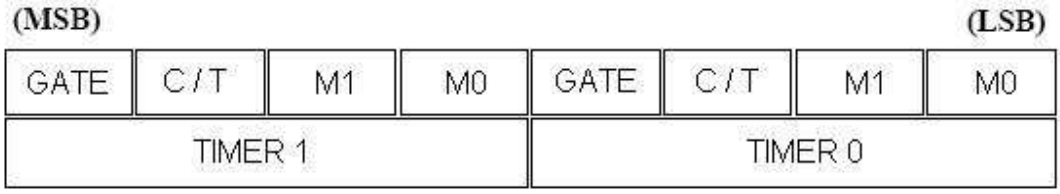
Timer 1 is also 16 bits, and its 16 bit register is split into two bytes, referred to as TL1(timer 1 low byte) and TH1 (timer 1 high byte). These registers are accessible in the same way as the registers of timer 0.




F ig(2): Timer 1 Registers

## TMOD (Timer Mode) Register

Both timers 0 and 1 use the same register, called TMOD, to set the various timer operation modes. TMOD is an 8-bit register in which the lower 4 bits are set aside for timer 0 and the upper4 bits are set aside for timer 1. In each case, the lower 2 bits are used to set the timer mode and the upper 2 bits to specify the operation. TMOD register is shown in fig(3).




Fig(3): TMOD Register

**GATE:** The T MOD register of Fig(3) that both timers 0 and 1 have the GATE bit. Every timer has means of starting and stopping. Some timers do this by software, some by hardware, and some both software and hardware controls. The timers in the 8051 have both. The start and stop of the timer are controlled by way of software by the TR (timer start) bits TR0 and TR1. This is achieved by the instructions "SETB TR1" and " CLR T R1" for timer 1 and " SETB TR0" and " CLR TR0" for time 0. The SETB instruction starts it, and it is stopped by the CLR instruction. These instructions start and stop the timers as long as GATE=0 in the TMOD register.

**M1, M0:** M0 and M1 select the timer mode. As show in the below Table, there are three modes; 0 ,1, and 2. Mode 0 is a 13 bit timer, mode 1 is a 16 bit timer and mode 2 is an 8-bit timer.

| M1 | M2 | MODE |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 2 |
| 1 | 1 | 3 |

**C / T (Clock / Timer):** This bit in the TMOD register is used to decide whether the timer is used as a delay generator or an event counter. If C/T =0, it is used as a timer for time delay generation. The clock source for the time delay is the crystal frequency of the 8051.

## Timer Programming

### Mode 1 Programming

**The following are the characteristics and operations of mode 1:**

1. It is a 16-bit timer, therefore it allows values of 0000 to FFFFH to be loaded into the timer's registers TL and TH.
2. After TH and TL are loaded with a 16-bit initial value, the timer must be started. This is done by "SETB TR0" for Timer 0 and "SETB TR1" for Timer 1.
3. After the timer is started, it starts to count up. It counts up until it reaches its limit of FFFFH. When it rolls over from FFFFH to 0000, it sets high flag bit called TF (timer flag). This timer flag can be monitered. When this timer flag is raised, one option would be to stop the timer with the instructions "CLR TR0" for Timer 0 and "CLR TR1" for Timer 1. Notice that each timer has its own timer flag: TF0 for Timer 0 and TF1 for Timer 1.

4. After the timer reaches its limit and rolls over, in order to repeat the process the registers TH and TL must be reloaded with the original value, and TF must be reset to '0'.

**Steps to Program in Mode 1**

1. Load the TMOD value register indicating which timer (Timer 0 or Timer 1) is to be used and which timer mode ( 0 or 1 ) is selected.
2. Load registers TL and TH with initial count values
3. Start the timer.
4. Keep monitoring the timer flag (TF). Get out of the loop when TF becomes high
5. Stop the timer.
6. Clear the TF flag for the next round.
7. Go back to Step 2 to load TH and TL again.

**Example**

| | | |
|---|---|---|
| | MOV TMOD, #01 | Time 0, mode 1 (16-bit mode) |
| HERE: | MOV TL0, #F2H | TL0 = F2H, the Low byte |
| | MOV TH0, #FFH | TH0 = FFH, the High byte |
| | CPL P1.5 | |
| | ACALL DELAY | |
| | SJMP HERE | |

Delay using Timer 0

**DELAY:**

| | | |
|---|---|---|
| | SETB TR0 | Start Timer 0 |
| AGAIN: | JNB TF0, AGAIN | Monitor Timer 0 flag until ir rolls over |
| | CLR TR0 | Stop Timer 0 |
| | CLR TF0 | Clear Timer 0 flag |
| | RET | |
| | | |

## Mode 2 Programming

**The following are the characteristics and operations of mode 2:**

1. It is an 8-bit timer, therefore, it allows only values of 00 to FFH to be loaded into the timer's register TH.
2. After TH is loaded with the 8-bit value, the 8051 gives a copy of it to TL. Then the timer must be started. This is done by the instruction "SETB TR0" for Timer 0 and "SETB TR1" for Timer 1.
3. After the timer is started, it starts to count up by incrementing the TL register. It counts up until it reaches its limit of FFH. When it rolls over from FFH to 00, it sets high the TF (Timer Flag). If we are using Timer 0, TF0 goes high. If we are using Timer 1, TF1 is raised.
4. When the TL register rolls from FFH to 00 and TF is set to 1, TL is reloaded automatically with the original value kept by the TH register. To repeat the process, we must simply clear TF and let it go without any need by the programmer to reload the original value. This makes mode 2 an auto-reload, in contrast with mode 1 in which the programmer has to reload TH and TL

It must be emphasized that mode 2 is an 8-bit timer. However, it has an auto-reloading capability. In auto-reload, TH is loaded with the initial count and a copy of it is given to TL. This reloading leaves TH unchanged, still holding a copy of the original value. This mode has many applications, including setting the baud rate in serial communication.

**Steps to program in mode 2**

To generate the time delay using the timer's mode 2, take the following steps.

1. Load the TMOD value register indicating which timer (Timer 0 or Timer 1) is to be used, and select the timer mode (mode 2)
2. Load the TH registers with the initial count value
3. Start the timer.
4. Keep monitoring the timer flag (TF) with the "JNB TF0, Target" or "JNB TF1, Target" instruction to see whether it is raised. Get out of the loop when TF goes high.
5. Clear the TF flag.
6. Go back to step 4, since mode 2 is auto-reload.

**Program: Assembly Language Program to generate the square on pin P1.0, assuming XTAL = 11.0592 MHz.**

```
        MOV   TMOD, #20H ; T1/ mode 2/ 8-bit/ auto-reload
        MOV   TH1, #05H  ; TH1 = 05
        SETB  TR1        ; Start Timer 1
BACK:   JNB   TF1, BACK  ; stay until timer rolls over
        CPL   P1.0       ; complement P1.0 to get high, low
        CLR   TF1        ; clear Timer 1 flag
        SJMP  BACK       ; mode 2 is auto-reload
```

## COUNTER PROGRAMMING

The timer / counter of the 8051 is used to generate time delays. These timers can also be used as counters counting events happening outside the 8051. As far as the use of a timer as an event counter is concerned, everything that in programming the timer applies to programming it as a counter, except tha source of the frequency. When the timer/counter is used as a timer, the 8051's crystal is used as the source of the frequency. When it is used as a counter, it is a pulse outside the 8051 that increments the TH, TL register.

### C/T bit in TMOD register

The C/T bit in the TMOD register decides the source of the clock for the timer. If C/T=0, the timer gets pulses from the crystal. In contrast, when C/T = 1, the timer is used as a counter and gets its pulses from outside the 8051. Therefore, when C/T = 1, the counter counters up as pulses are fed from pins 14 and 15. These pins are called T0 (Timer 0 input) and T1 (Timer 1 input). These two pins belong to port 3. In the case of Timer 0, when C/T = 1, pin P3.4 provides the clock pulse and the counter counts up for each clock pulse coming from that pin. Similarly, for Timer 1, when C/T = 1 each clock pulse coming in from pin P3.5 makes the counter count up.

**Port 3 Pins Used For Timer 0 and Timer 1**

| Pin | Port Pin | Function | Description |
|---|---|---|---|
| 14 | P3.4 | T0 | Timer / Counter 0 external input |
| 15 | P3.5 | T1 | Timer / Counter 1 external input |

(MSB) (LSB)

| GATE | C/T | M1 | M2 | GATE | C/T | M1 | M2 |
|---|---|---|---|---|---|---|---|
| Timer 1 | | | | Timer 0 | | | |

Fig:TMOD register

**TCON REGISTER**

| TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
|---|---|---|---|---|---|---|---|

Fig : TCON register

TCON register is an 8-bit register. As shown in above figure the upper four bits are used to store the TF and TR bits of both Timer 0 and Timer 1. The lower four bits are set aside for controlling the interrupt bits. TCON is a bit addressable register.