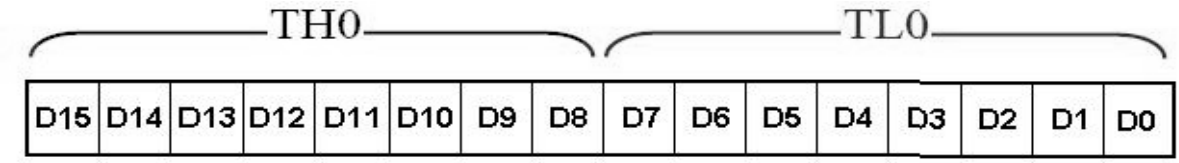
PROGRAMMING 8051 TIMERS

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

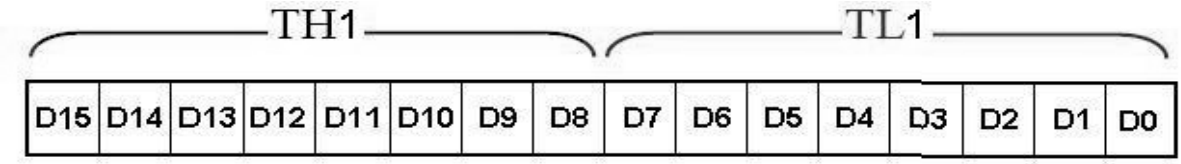
The 16 bit register of timer 0 is accessed as low byte and high byte. The low byte register is called TL0 (timer 0 low byte) and the high byte register is referred to as TH0 (timer 0 high byte). These registers can be accessed like any other register, such as A, B, R0, R1, R2 etc. For example, the instruction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



Fig(1): Timer 0 Registers

TIMER 1 registers

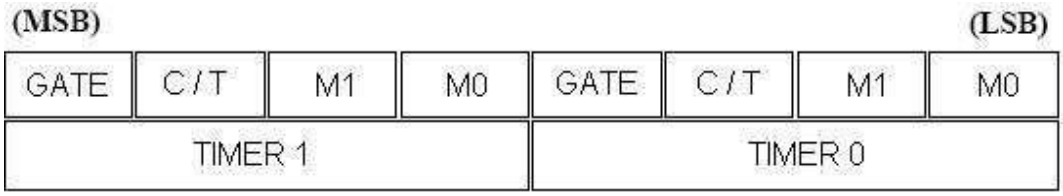
Timer 1 is also 16 bits, and its 16 bit register is split into two bytes, referred to as TL1(timer 1 low byte) and TH1 (timer 1 high byte). These registers are accessible in the same way as the registers of timer 0.



F ig(2): Timer 1 Registers

TMOD (Timer Mode) Register

Both timers 0 and 1 use the same register, called TMOD, to set th 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 "C L R 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 monitored. 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 the 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

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.

TCON and TMOD

- Event counting is used to determine the number of occurrences of an event
- An event is an external stimulus that provides a 1 to 0 transition
- 8051 timers are accessed using six of SFRs (i.e., TCON, TMOD, TL0, TL1, TH0,TH1)
- TMOD register contains two groups of four bits that set the operating mode for Timer 0 and Timer 1
- TMOD is not bit addressable and is loaded once at the beginning of a program to initialize the timer
- TCON contains status bits and control bits for Timer 0 and Timer 1.

TMOD Register

TMOD : Timer/Counter Mode Control Register (Not Bit Addressable)

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
TIMER 1				TIMER 0			

GATE — When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/$\overline{T}$ — Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 — Mode selector bit (NOTE 1).

M0 — Mode selector bit (NOTE 1).

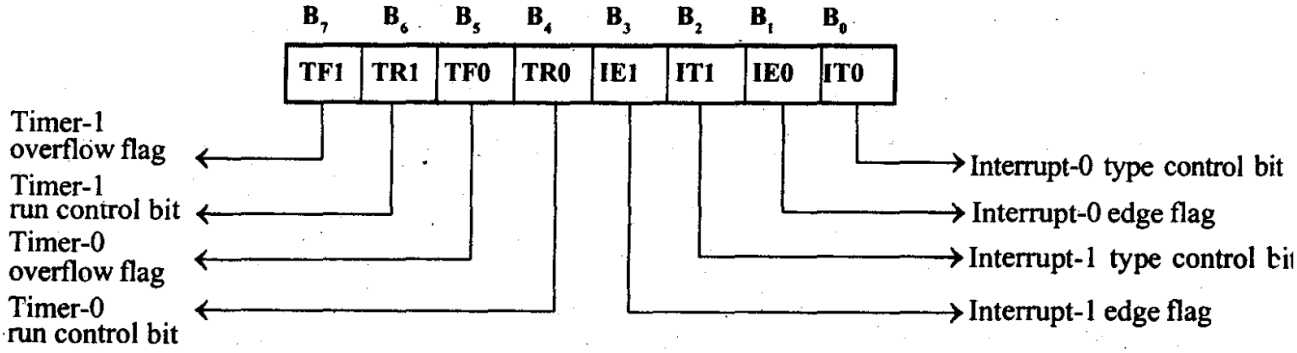
Note 1 :

M1	M0	OPERATING MODE	
0	0	0	13–bit Timer
0	1	1	16–bit Timer/Counter
1	0	2	8–bit Auto–Reload Timer/Counter
1	1	3	(Timer 0) TL0 is an 8–bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8–bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

(MSB) (LSB)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 Overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn timer/counter on/off.
TF0	TCON.5	Timer 0 Overflow Flag. Set by hardware on timer/counter overflow. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn timer/counter on/off.
IE1	TCON.3	Interrupt 1 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 Edge Flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.



TMOD Register

TMOD (Timer Mode) is an SFR. The address of this register is 89H. This is not bit-addressable.

Timer	Timer1 Mode				Timer0 Mode			
Bit Details	Gate (G)	C/T	M1	M0	Gate (G)	C/T	M1	M0

Now, let us see the circuit that controls the running of the timers.

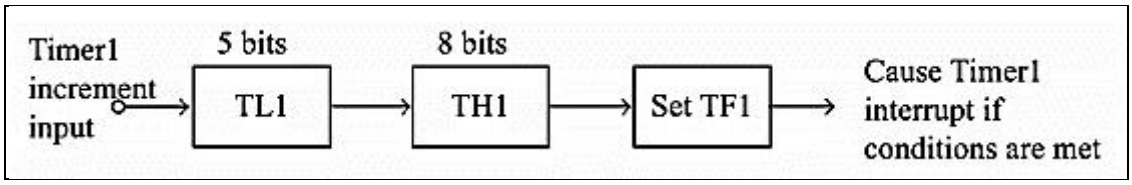
In the following table, we will see the bit details and their different operations for high or low value.

Bit Details	High Value(1)	Low Value(0)
C/**T**	Configure for the Counter operations	Configure for the Timer operations
Gate (G)	Timer0 or Timer1 will be in Run Mode when TRX bit of TCON register is high.	Timer0 or Timer1 will be in Run Mode when TRX bit of TCON register is high and INT0 or INT1 is high.

Bit Details	00	01	10	11
M1 M0	This is for Mode 0. (8-bit timer/counter, with 5-bit pre-scaler)	This is Mode 1. (16-bit timer/counter)	This is Mode 3 (8-bit auto reload-timer/counter)	This is Mode 3 (The function depends on Timer0 or Timer1)

Mode 0 of Timer/Counter

The Mode 0 operation is the 8-bit timer or counter with a 5-bit pre-scaler. So it is a 13-bit timer/counter. It uses 5 bits of TL0 or TL1 and all of the 8-bits of TH0 or TH1.



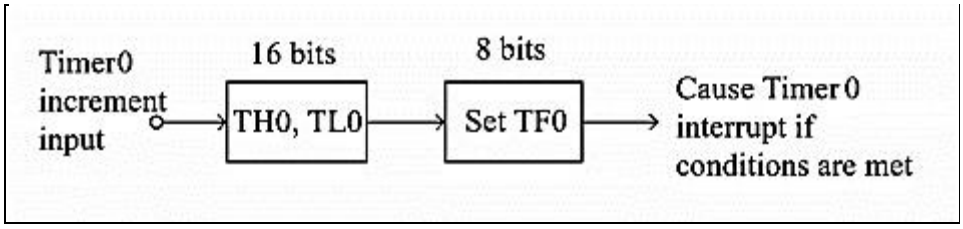
In this example the Timer1is selected, in this case, every 32 (25)event for counter operations or 32 machine cycles for timer operation, the TH1 register will be incremented by 1. When the TH1overflows from FFH to 00H, then the TF1 of TCON register will be high, and it stops the timer/counter. So for an example, we can say that if the TH1 is holding F0H, and it is in timer mode, then TF1will be high after 10H * 32 = 512 machine cycles.

```
MOVTMOD, #00H

MOVTH1, #0F0H

MOVIE, #88H

SETB TR1
```

In the above program, the Timer1 is configured as timer mode 0. In this case Gate = 0. Then the TH1 will be loaded with F0H, then enable the Timer1 interrupt. At last set the TR1 of TCON register, and start the timer.

Mode 1 of Timer/Counter

The Mode 1 operation is the 16-bit timer or counter. In the following diagram, we are using Mode 1 for Timer0.



In this case every event for counter operations or machine cycles for timer operation, the TH0– TL0 register-pair will be incremented by 1. When the register pair

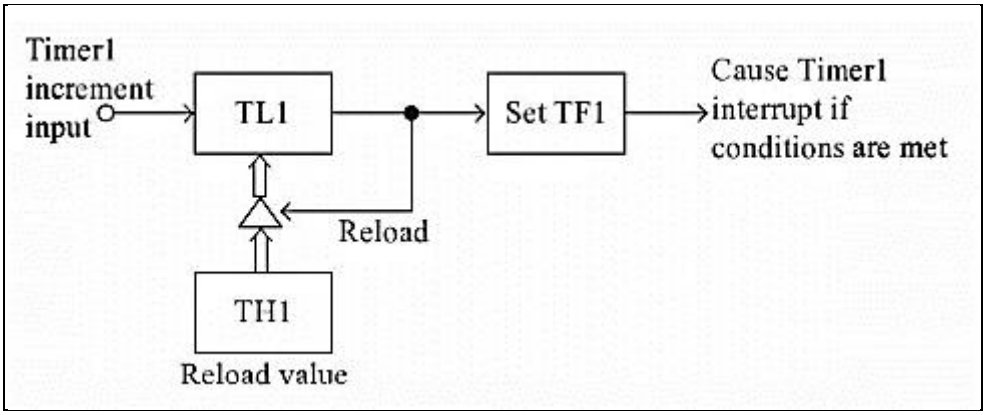
overflows from FFFFH to 0000H, then the TF0 of TCON register will be high, and it stops the timer/counter. So for an example, we can say that if the TH0 – TL0 register pair is holding FFF0H, and it is in timer mode, then TF0 will be high after 10H = 16 machine cycles. When the clock frequency is 12MHz, then the following instructions generate an interrupt 16 μs after Timer0 starts running.

```
MOVTMOD, #01H

MOVTL0, #0F0H

MOVTH0, #0FFH

MOVIE, #82H

SETB TR0
```

In the above program, the Timer0 is configured as timer mode 1. In this case Gate = 0. Then the TL0 will be loaded with F0H and TH0 is loaded with FFH, then enable the Timer0 interrupt. At last set the TR0 of TCON register, and start the timer.

Mode 2 of Timer/Counter

The Mode 2 operation is the 8-bit auto reloads timer or counter. In the following diagram, we are using Mode 2 for Timer1.



In this case every event for counter operations or machine cycles for timer operation, the TL1register will be incremented by 1. When the register pair overflows from FFH

to 00H, then the TF1 of TCON register will be high, also theTL1 will be reloaded with the content of TH1 and starts the operation again.

So for an example, we can say that if the TH1 and TL1 register both are holding F0H and it is in timer mode, then TF1 will be high after 10H= 16 machine cycles. When the clock frequency is 12MHz this happens after 16 μs, then the following instructions generate an interrupt once every 16 μs after Timer1 starts running.

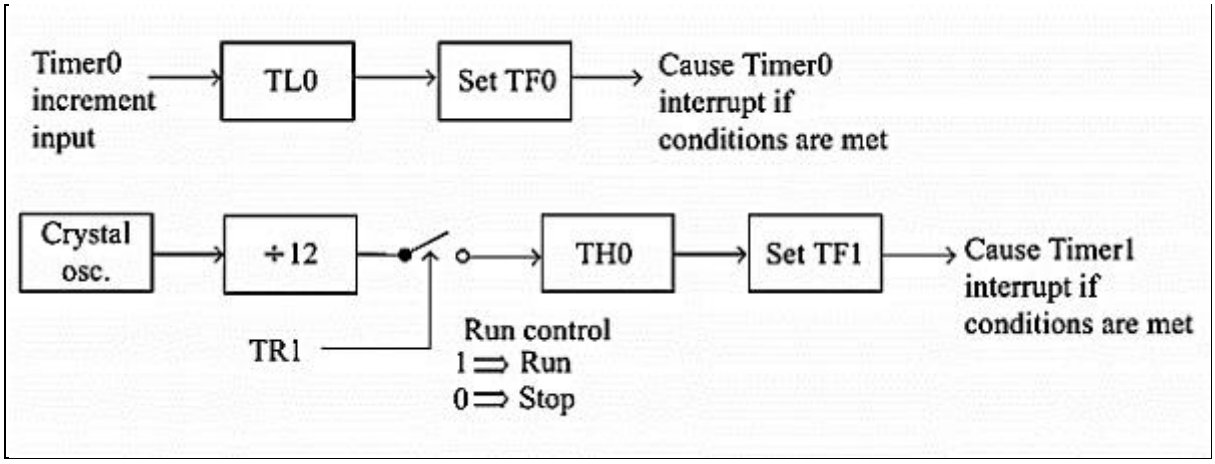
```
MOVTMOD, #20H

MOVTL1, #0F0H

MOVTH1, #0F0H

MOVIE, #88H

SETBTR1
```

In the above program, the Timer1 is configured as timer mode 2. In this case Gate = 0. Then the TL1 and TH1 are loaded with F0H. then enable the Timer1 interrupt. At last set the TR1 of TCON register, and start the timer.

Timer1 in mode 2 generates the desired baud rate when the serial port is working on Mode 1 or 3.

Mode 3 of Timer/Counter

Mode 3 is different for Timer0 and Timer1. When the Timer0 is working in mode 3, the TL0 will be used as an 8-bit timer/counter. It will be controlled by the standard Timer0 control bits, T0 and INT0 inputs. The TH0 is used as an 8-bit timer but not the counter. This is controlled by Timer1 Control bit TR1. When the TH0 overflows from FFH to 00H, then TF1 is set to 1. In the following diagram, we can Timer0 in Mode 3.



When the Timer1 is working in Mode 3, it simply holds the count but does not run. When Timer0 is in mode 3, the Timer1 is configured in one of the mode 0, 1 and 2. In this case, the Timer1 cannot interrupt the microcontroller. When the TF1 is used by TH0 timer, the Timer1 is used as Baud Rate Generator.

The meaning of gate bit in Timer0 and Timer1 for mode 3 is as follows

It controls the running of 8-bit timer/counter TL0 as like Mode 0, 1, or 2. The running of TH0 is controlled by TR1 bit only. So the gate bit in this mode for Timer0 has no specific role.

The mode 3 is present for applications requiring an extra 8-bit timer/counter. In Mode 3 of Timer0, the 8051 has three timers. One 8-bit timer by TH0, another8-bit timer/counter by TL0, and one 16-bit timer/counter by Timer1.

If the Timer0 is in mode3, and Timer1 is working on 0, 1 or 2, then the gun control of the Timer1 is activated when the gate bit is low or INT1 is high. The run control is deactivated when the gate is high and INT1 is low.