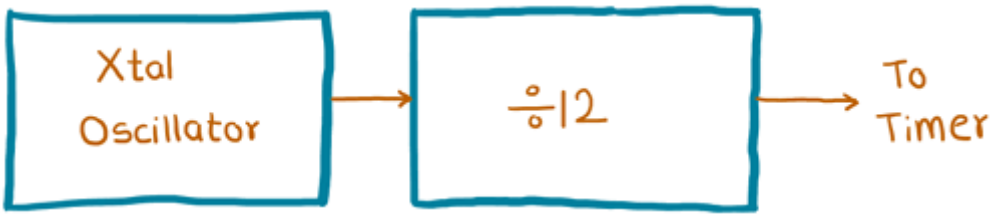
# 8051 Timers

## Introduction

8051 microcontrollers have two timers/counters which work on the clock frequency. Timer/counter can be used for time delay generation, counting external events, etc.

## Clock

Every Timer needs a clock to work, and 8051 provides it from an external crystal which is the main clock source for Timer. The internal circuitry in the 8051 microcontrollers provides a clock source to the timers which is 1/12th of the frequency of crystal attached to the microcontroller, also called Machine cycle frequency.




**8051 Timer Clock**

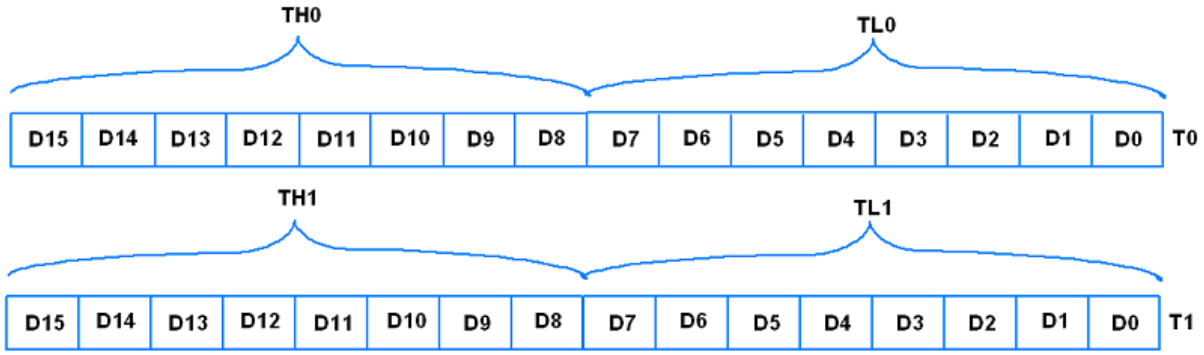
For example, suppose we have a crystal frequency of 11.0592 MHz then the microcontroller will provide 1/12th i.e.

Timer clock frequency= (Xtal Osc.frequency)/12 = (11.0592 MHz)/12 = 921.6 KHz

period T= 1/(921.6 kHz)=1.085 µS

## Timer

8051 has two timers Timer0 (T0) and Timer1 (T1), both are 16-bit wide. Since 8051 has 8-bit architecture, each of these is accessed by two separate 8-bit registers as shown in the figure below. These registers are used to load timer count.

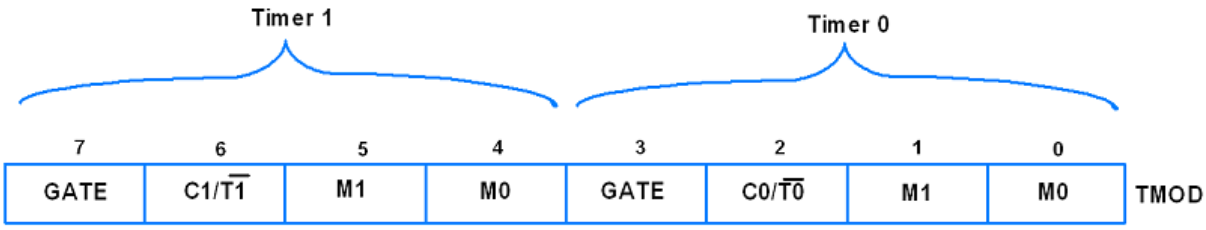



8051 has a Timer Mode Register and Timer Control Register for selecting a mode of operation and controlling purpose.

Let's see these registers,

**TMOD register**

TMOD is an 8-bit register used to set timer mode of timer0 and timer1.




Its lower 4 bits are used for Timer0 and the upper 4 bits are used for Timer1

**Bit 7,3 – GATE:**

**1** = Enable Timer/Counter only when the INT0/INT1 pin is high and TR0/TR1 is set.

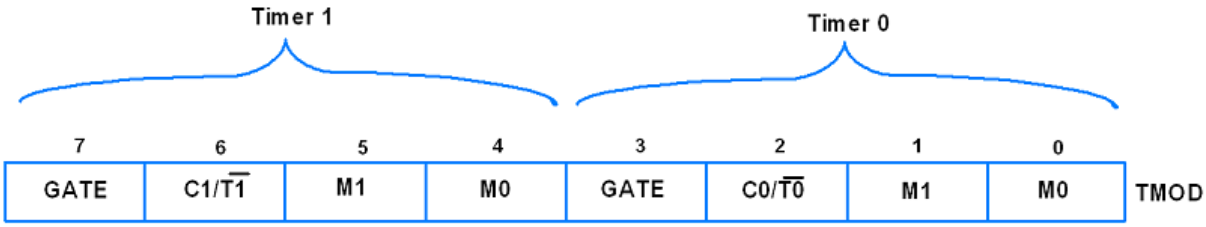
**0** = Enable Timer/Counter when TR0/TR1 is set.

**Bit 6,2 - C/$\overline{T}$ (Counter/Timer):** Timer or Counter select bit

**1** = Use as Counter

**0** = Use as Timer

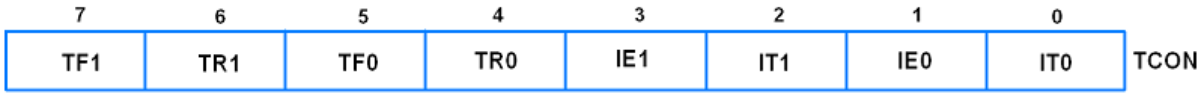
**Bit 5:4 & 1:0 - M1:M0:** Timer/Counter mode select bit




These are Timer/Counter mode select bit as per the below table

| M1 | M0 | Mode | Operation |
|---|---|---|---|
| 0 | 0 | 0 (13-bit timer mode) | 13-bit timer/counter, 8-bit of THx & 5-bit of TLx |
| 0 | 1 | 1 (16-bit timer mode) | 16-bit timer/counter, THx cascaded with TLx |
| 1 | 0 | 2 (8-bit auto-reload mode) | 8-bit timer/counter (auto-reload mode), TLx reload with the value held by THx each time TLx overflow |
| 1 | 1 | 3 (split timer mode) | Split the 16-bit timer into two 8-bit timers i.e. THx and TLx like two 8-bit timer |

**TCON Register**




TCON is an 8-bit control register and contains a timer and interrupt flags.

**Bit 7 - TF1:** Timer1 Overflow Flag

**1** = Timer1 overflow occurred (i.e. Timer1 goes to its max and roll over back to zero).

**0** = Timer1 overflow not occurred.

It is cleared through software. In the Timer1 overflow interrupt service routine, this bit will get cleared automatically while exiting from ISR.

**Bit 6 - TR1:** Timer1 Run Control Bit

**1** = Timer1 start.

**0** = Timer1 stop.

It is set and cleared by software.

**Bit 5 – TF0:** Timer0 Overflow Flag

**1** = Timer0 overflow occurred (i.e. Timer0 goes to its max and roll over back to zero).

**0** = Timer0 overflow not occurred.

It is cleared through software. In the Timer0 overflow interrupt service routine, this bit will get cleared automatically while exiting from ISR.

**Bit 4 – TR0:** Timer0 Run Control Bit

**1** = Timer0 start.

**0** = Timer0 stop.

It is set and cleared by software.

**Bit 3 - IE1:** External Interrupt1 Edge Flag

**1** = External interrupt1 occurred.

**0** = External interrupt1 Processed.

It is set and cleared by hardware.

**Bit 2 - IT1:** External Interrupt1 Trigger Type Select Bit

**1 =** Interrupt occurs on falling edge at INT1 pin.

**0 =** Interrupt occur on a low level at the INT1 pin.

**Bit 1 – IE0:** External Interrupt0 Edge Flag

**1** = External interrupt0 occurred.

**0** = External interrupt0 Processed.

It is set and cleared by hardware.

**Bit 0 – IT0:** External Interrupt0 Trigger Type Select Bit

**1 =** Interrupt occurs on falling edge at INT0 pin.

**0 =** Interrupt occur on a low level at INT0 pin.

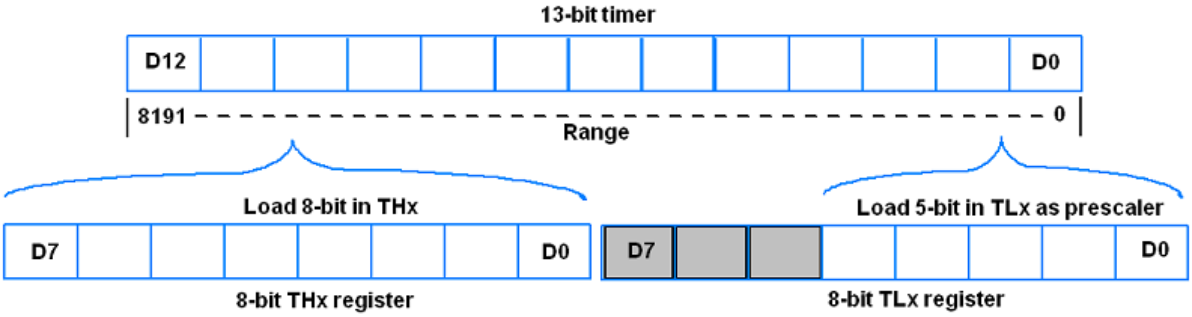
Let's see the timers modes

**Timer Modes**

Timers have their operation modes which are selected in the TMOD register using M0 & M1 bit combinations.

Mode 0 (13-bit timer mode)

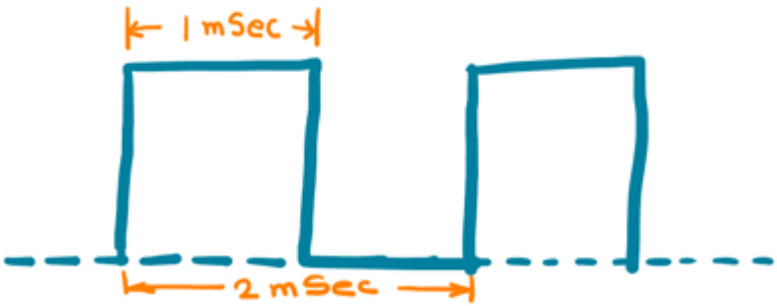
Mode 0 is a 13-bit timer mode for which 8-bit of THx and 5-bit of TLx (as Prescaler) are used. It is mostly used for interfacing possible with old MCS-48 family microcontrollers.




As shown in the above figure, 8-bit of THx and lower 5-bit of TLx used to form a total 13-bit timer. Higher 3-bits of TLx should be written as zero while using timer mode0, or it will affect the result.

**Example**

Let's generate a square wave of 2mSec period using an AT89C51 microcontroller with timer0 in mode0 on the P1.0 pin of port1. Assume xtal oscillator frequency of 11.0592 MHz.




As the Xtal oscillator frequency is 11.0592 MHz we have a machine cycle of 1.085uSec. Hence, the required count to generate a delay of 1mSec. is,

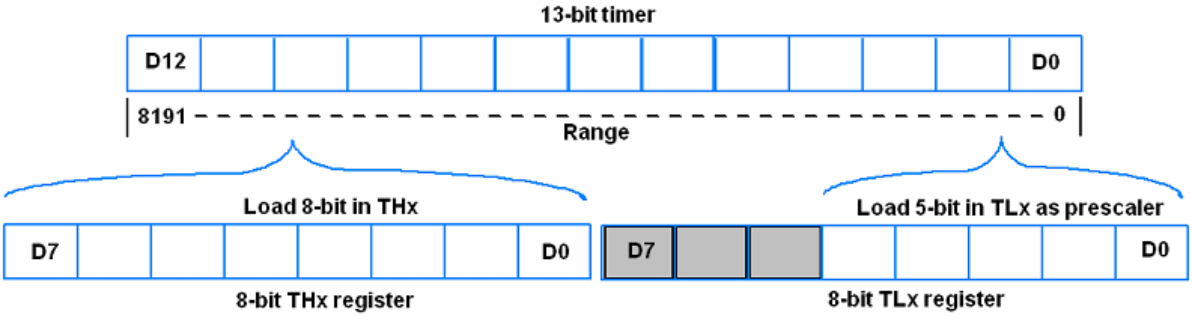
$$\text{Count} = (1\times10^{-3})/(1.085\times10^{-6}) \approx 921$$

The maximum count of Mode0 is 2^13 (0 - 8191) and the Timer0 count will increment from 0 – 8191. So we need to load value which is 921 less from its maximum count i.e. 8191. Also, here in the below program, we need an additional 13 MC (machine cycles) from call to return of delay function. Hence value needed to be loaded is,

$$\text{Value}=(8191-\text{Count})+\text{Function\_MCycles}+1 = 7284 = 0x1C74$$

So we need to load 0x1C74 value in Timer0.

1C74 = 000**1 1100 0111 0100** b, now load lower 5-bit in TL0 and next 8-bit in TH0




so here we get,

TL0 = 000**1 0100** = 0x14 and TH0 = 1110 0011 = 0xE3

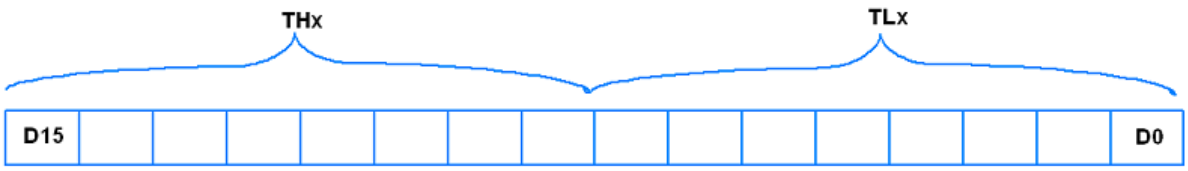
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| TL0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 14H |
| TH0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | E3H |

## Programming steps for delay function

1. Load Tmod register value i.e. TMOD = 0x00 for Timer0/1 mode0 (13-bit timer mode).
2. Load calculated THx value i.e. here TH0 = 0xE3.
3. Load calculated TLx value i.e. here TL0 = 0x14.
4. Start the timer by setting a TRx bit. i.e. here TR0 = 1.
5. Poll TFx flag till it does not get set.
6. Stop the timer by clearing TRx bit. i.e. here TR0 = 0.
7. Clear timer flag TFx bit i.e. here TF0 = 0.
8. Repeat from step 1 to 7 for the delay again.

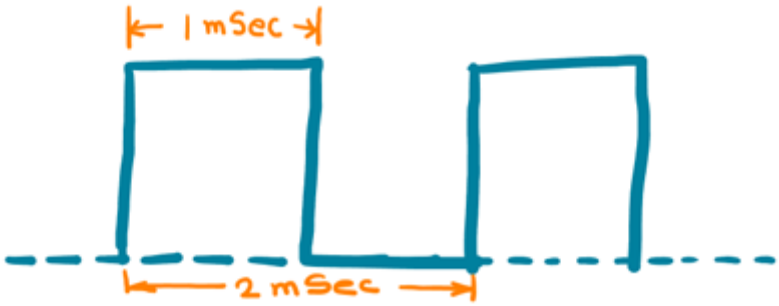
## Mode1 (16-bit timer mode)

Mode 1 is a 16-bit timer mode used to generate a delay, it uses 8-bit of THx and 8-bit of TLx to form a total 16-bit register.



**Example**

Let's generate a square wave of 2mSec time period using an AT89C51 microcontroller with timer0 in mode1 on the P1.0 pin of port1. Assume Xtal oscillator frequency of 11.0592 MHz.




As Xtal is 11.0592 MHz we have a machine cycle of 1.085uSec.

Hence, the required count to generate a delay of 1mSec. is,

$$\text{Count} = (1\times10^{(-3)}) / (1.085\times10^{(-6)}) \approx 921$$

And mode1 has a max count is 2^16 (0 - 65535) and it increments from 0 to 65535 so we need to load value which is 921 less from its max. count i.e. 65535. Also, here in the below program, we need an additional 13 MC (machine cycles) from call to return of delay function. Hence value needed to be loaded is,

$$\text{Value}=(65535-\text{Count})+\text{Function\_MCycles}+1$$

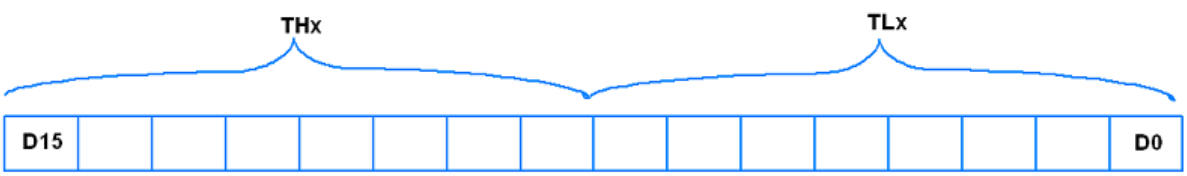
$$=65535-921+13+1=64615= (\text{FC74})\text{Hex}$$

So we need to load FC74 Hex value higher byte in TH0 and lower byte in TL0 as,

FC74=1111 1100 0111 0100 b

TH0 = 0xFC & TL0 = 0x74

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| TH0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | FC |
| TL0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 74 |

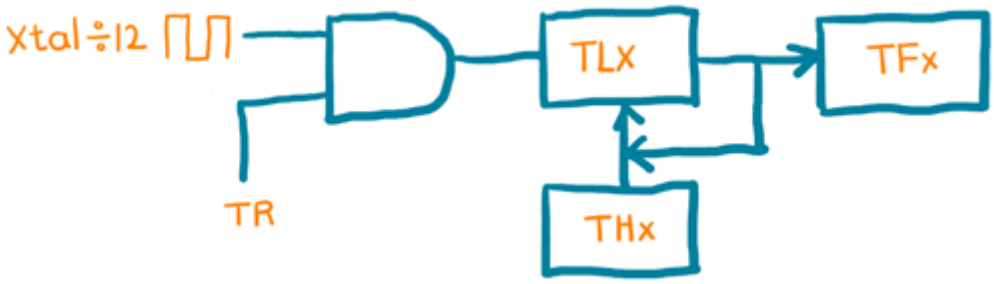


Programming steps for delay function

1. Load Tmod register value i.e. TMOD = 0x01 for Timer0 mode1 (16-bit timer mode).
2. Load calculated THx value i.e. here TH0 = 0xFC.
3. Load calculated TLx value i.e. here TL0 = 0x74.
4. Start the timer by setting a TRx bit. i.e. here TR0 = 1.
5. Poll TFx flag till it does not get set.
6. Stop the timer by clearing TRx bit. i.e. here TR0 = 0.
7. Clear timer flag TFx bit i.e. here TF0 = 0.
8. Repeat from step 1 to 7 for the delay again.

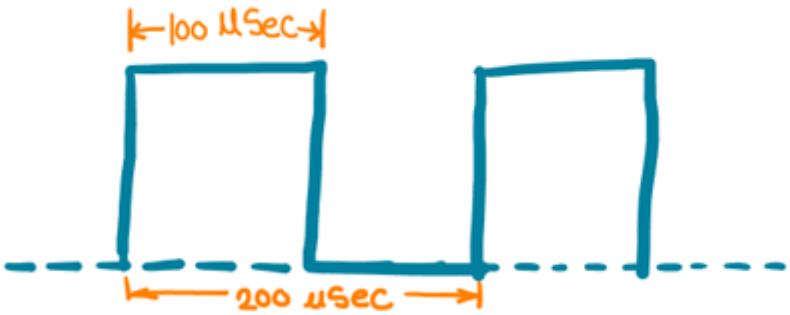
Mode2 (8-bit auto-reload timer mode)

Mode 2 is an 8-bit auto-reload timer mode. In this mode, we have to load the THx-8 bit value only. when the Timer gets started, the THx value gets automatically loaded into the TLx and TLx starts counting from that value. After the value of TLx overflows from the 0xFF to 0x0, the TFx flag gets set and again value from the THx gets automatically loaded into the TLx register. That's why this is called the auto-reload mode.




**Example**

Here we are generating a square wave on PORT1.0 with 200uSec. time period using Timer1 in mode2. We will use 11.0592 MHz Xtal oscillator frequency.



As Xtal is 11.0592 MHz we have a machine cycle of 1.085uSec. Hence, the required count to generate a delay of 1mSec. is,

$$Count = (100\times10^{(-6)}) / (1.085\times10^{(-6)}) \approx 92$$

And mode2 has a max count is 2^8 (0 - 255) and it increment from 0 – 255 so we need to load value which is 92 less from its max. count i.e. 255. Hence value need to be load is,

$$Value=(255-Count)+1 =164= 0xA4$$

So we need to load A4 Hex value in a higher byte as,

TH1 = 0xA4

## Programming steps for delay function

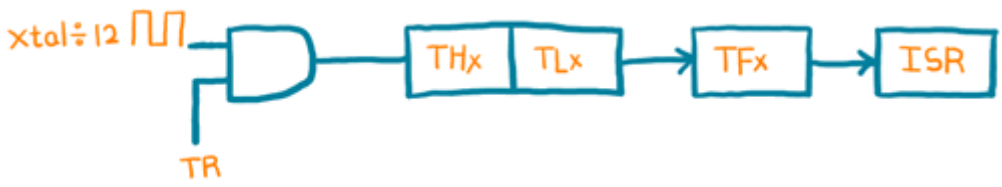
1. Load Tmod register value i.e. TMOD = 1x02 for Timer1 mode2 (8-bit timer auto reload mode).
2. Load calculated THx value i.e. here TH1 = 0xA4.
3. Load same value for TLx i.e. here TL1 = 0xA4.
4. Start the timer by setting a TRx bit. i.e. here TR1 = 1.
5. Poll TFx flag till it does not get set.
6. Clear timer flag TFx bit i.e. here TF1 = 0.
7. Repeat from step 1 to 6 for the delay again.

## Timer interrupt in 8051

8051 has two timer interrupts assigned with different vector address. When Timer count rolls over from its max value to 0, it sets the timer flag TFx. This will interrupt the 8051 microcontroller to serve ISR (interrupt service routine) if global and timer interrupt is enabled.

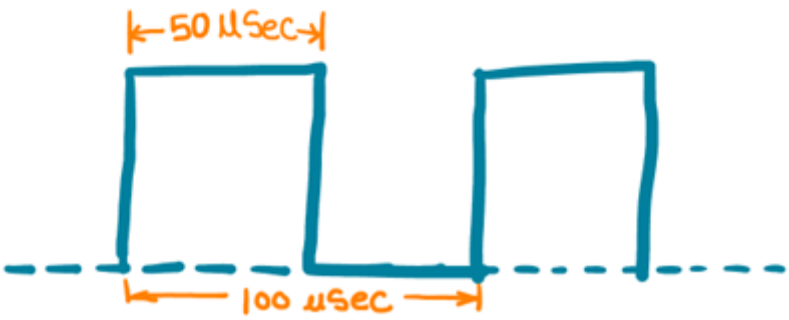
| **Interrupt source** | **Vector address** |
|---|---|
| Timer 0 overflow (TF0) | 000BH |
| Timer 1 overflow (TF1) | 001BH |

The timer overflow interrupt assigned with the vector address shown in the table. 8051 microcontroller jumps directly to the vector address on the occurrence of a corresponding interrupt.




**Example**

Here we will generate a square wave of 10Hz on PORT1.0 using Timer0 interrupt. We will use Timer0 in mode1 with 11.0592 MHz oscillator frequency.




As Xtal is 11.0592 MHz we have a machine cycle of 1.085uSec. Hence, the required count to generate a delay of 50mSec. is,

$$\text{Count} = (50\times10^{(-3)}) / (1.085\times10^{(-6)}) \approx 46080$$

And mode1 has a max count is 2^16 (0 - 65535) and it increment from 0 – 65535 so we need to load value which is 46080 less from its max. count i.e. 65535. Hence value need to be load is,

$$\text{Value}=(65535-\text{Count})+1 = 19456 = (4C00)\text{Hex}$$

So we need to load 4C00 Hex value in a higher byte and lower byte as,

TH0 = 0x4C & TL0 = 0x00

Note that the TF0 flag no need to clear by software as a microcontroller clears it after completing the ISR routine.