Embedded System Lab: Session First - Microcontroller

LAB – 3: Programming Timers of 8051/8052 Microcontroller

Objectives

To enable us to write assembly language code for the 8051/8052 micro-controller capable of:

- Applying timers in different timing modes.
- Implementing accurate delays using timers.

Equipment Required

- Hardware: 8051 or 8052 micro-controller development board, Jumper cables
- Simulation Software: KEIL, Vision-Embedded development tool, Proteus Design Suite – Professional PCB layout, circuit design and simulation tool
- In-System Programming (ISP) Software: ProgISP – An in-system-programmable tool to load HEX files in to micro-controller
- Device Drivers: LibUSB – Application controlling data transfer to/from USB devices

Background

The 8051 microcontroller has two 16-bits timers T0 and T1. The general function of timer and counter are they are used in calculating the amounts of time between events, counting events and generating baud rate for serial port. Their application could be in communication for generating rectangle pulses, watchdog timer, in manufacturing industry for counting objects, measuring intervals, etc. There are two different types of timer: Interval timer and Counter. The initial state can be set by user. Timer/Counters can be operated by user with special function registers. To and T1 share two SFRs: TMOD and TCON. Each timer has also two register dedicated for themselves. For timer 1: TH0 and TL0, for timer 2: TH1 and TL1; where TH is high byte register and TL is low byte register. TL and TH registers of a timer are byte addressable only.

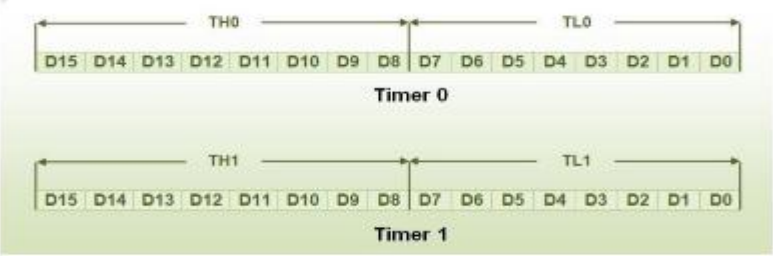

Figure 1: Timer 0 and 1 Registers showing TH and TL; Source: Techtidda

Timer Mode Register (TMOD): This is the 8-bits register available in 8051 microcontroller to select a timer mode of operation. It is byte addressable only, which is loaded at the very beginning of a program to initialize a timer's mode. Both timers use this TMOD register, timer 0 uses the lower 4-bits while timer 1 uses the upper 4-bits.

TMOD - Timer/Counter 0 and 1 Modes

7	6	5	4	3	2	1	0
GATE1	C/T1#	M11	M01	GATE0	C/T0#	M10	M00

Bit Number	Bit Mnemonic	Description
7	GATE1	**Timer 1 Gating Control Bit** Clear to enable timer 1 whenever the TR1 bit is set. Set to enable timer 1 only while the INT1# pin is high and TR1 bit is set.
6	C/T1#	**Timer 1 Counter/Timer Select Bit** Clear for timer operation: timer 1 counts the divided-down system clock. Set for Counter operation: timer 1 counts negative transitions on external pin T1.
5	M11	**Timer 1 Mode Select Bits** M11 M01 Operating mode 0 0 Mode 0: 8-bit timer/counter (TH1) with 5-bit prescaler (TL1). 0 1 Mode 1: 16-bit timer/counter. 1 0 Mode 2: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow. 1 1 Mode 3: timer 1 halted. Retains count.
4	M01	
3	GATE0	**Timer 0 Gating Control Bit** Clear to enable timer 0 whenever the TR0 bit is set. Set to enable timer/counter 0 only while the INT0# pin is high and the TR0 bit is set.
2	C/T0#	**Timer 0 Counter/Timer Select Bit** Clear for timer operation: timer 0 counts the divided-down system clock. Set for counter operation: timer 0 counts negative transitions on external pin T0.
1	M10	**Timer 0 Mode Select Bit** M10 M00 Operating mode 0 0 Mode 0: 8-bit timer/counter (TH0) with 5-bit prescaler (TL0). 0 1 Mode 1: 16-bit timer/counter. 1 0 Mode 2: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow. 1 1 Mode 3: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer using timer 1's TR0 and TF0 bits.
0	M00	

Reset Value = 0000 0000b

Figure 2: TMOD Register with explanation in brief; Source: circuits4you.com

Timer Control Register (TCON): The 8051 microcontroller has one 8-bit register that holds the timer flags, interrupt ags and timer run control bit. This register is bit addressable and is used by both timers as well as interrupts. The timers use the upper 4-bits while interrupts use the lower 4-bits.

Clock sources for Timers: Using TMOD register, timer operation is selected, and timer is clocked from an oscillator. Frequency for timer is $1/12^{th}$ the frequency of the crystal attached to the 8051 microcontroller, which is equivalent to 921.6 KHz (frequency of an oscillator is 11.0592 MHz). This is so as in 8051 microcontroller, 12 oscillator periods constitute a machine cycle. Hence machine cycle period is 1.085 microseconds.

There are four different timer modes: Mode 0, Mode 1, Mode 2 and Mode 3. **Mode 0** is identical for Timer 0 and Timer 1. Both timers work as 13-bit counters; an interrupt is generated when counter overflows. It takes 8192 input pulses to generate the next interrupt. Timers use 8-bits of THi and 5 lower bits of TLi. After timer overflows TFi (Timer flag in TCON) is set, hence an interrupt occurs.

TCON : Timer/Counter Control Register (Bit Addressable)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
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 when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/flow level triggered External Interrupt.
IE0	TCON.1	External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

Figure 3: TCON Register with explanation in brief; Source: Detube

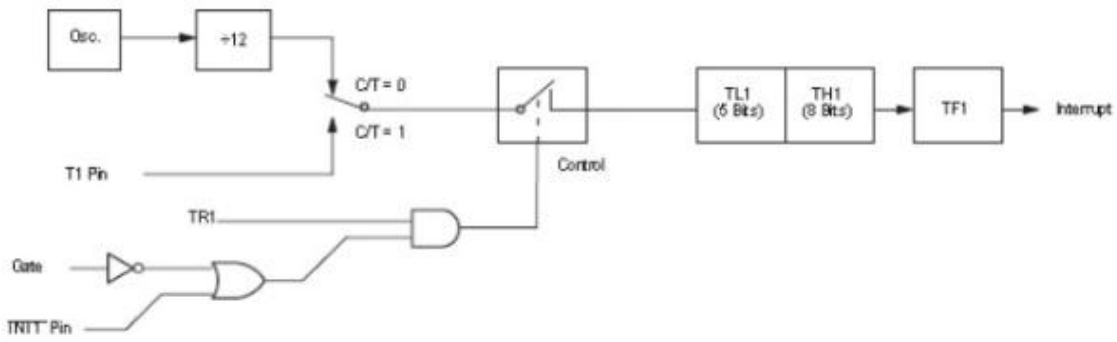

Figure 4: Structure of Timer 1 in mode 0

Mode 1: This mode is similar to mode 0. This timer uses all 8 bits of THi and 8 bits of TLi. So it is a 16-bit counter which can take 65536 input pulses to generate the next interrupt.

Mode 2: In this mode, the timers are 8 bits auto reload type. The timer is operated by TLi, when TLi overflows again it is automatically loaded by THi. So the initial value is loaded to the THi register at first.

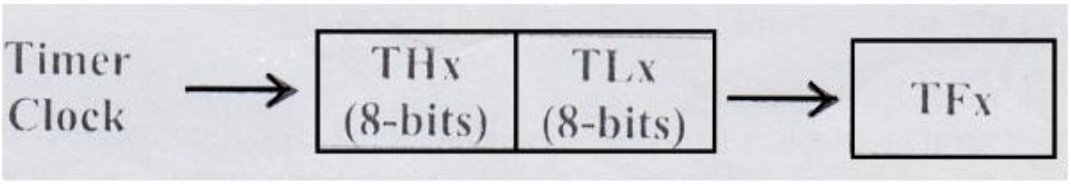

Figure 5: Structure of Timer 0 in mode 1

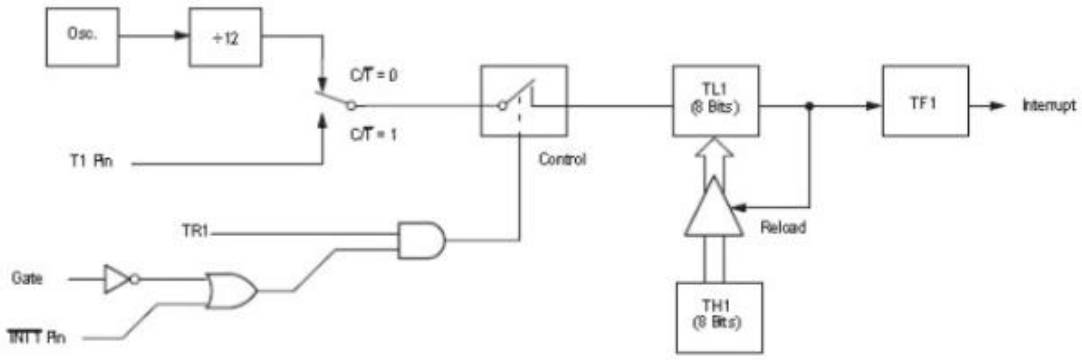

Figure 6: Structure of Timer 1 in mode 2

Mode 3: In this mode, only timer 0 can be used. This is also called split timer mode. Timer 0 operates TL0 and TH0 as two separate 8 bit timers/counters. Timer 0 with TL0 is operated with TF0 and TR0 while timer 0 with TH0 is operated with TF1 and TR1.

Basic Programming Steps:

- First of all, Timer and its mode of operation are selected by loading proper bytes in TMOD register. For eg. to use timer 1 in mode 0, the instruction would be like
  MOV TMOD, #00H
- Then, appropriate value is loaded in Timer Registers according to the mode and timer selected previously. For eg. in same case
  MOV TL1, #1FH
  MOV TH1, #0A9H

- After that TRx (Timer x Run of TCON register) bit is set to start timer to run. For eg: **SETB TR1**
- Check the flag TFx (of TCON register) regularly to get the overflow information for proper timing.

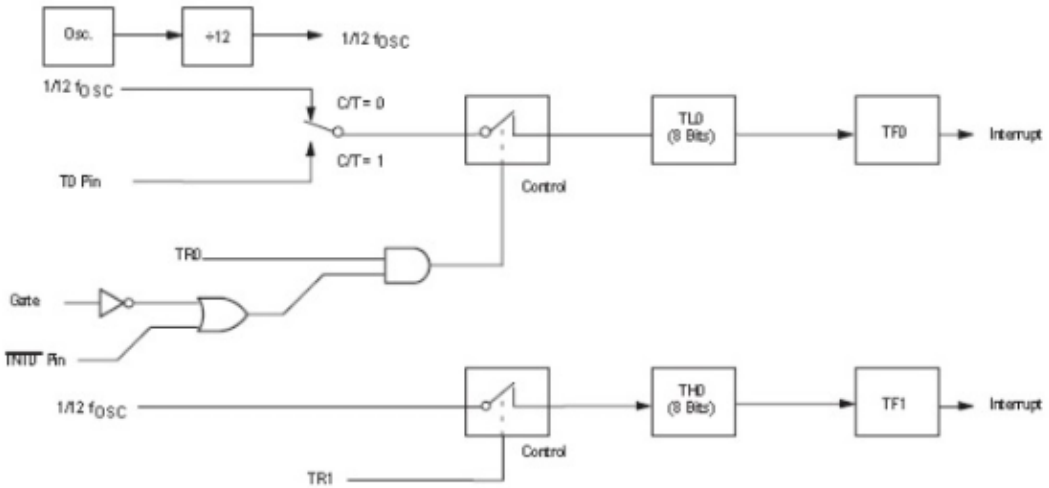

Figure 7: Structure of Timer 0 in mode 3

- After overflow, the Timer Run bit (TRx) and the overflow flag (TFx) is reset or cleared.

Problems

The circuit diagram used for simulation for this lab is shown below:

1. Generate a periodic square wave having a period of 15 ms and a duty cycle of 20%. The waveform should be produced at pin zero of port two (P2.0). The XTAL frequency is 11.0592 MHz. Observe the waveform on an oscilloscope and measure the ON and OFF timers.
   (a) Using Timer 1 in mode 0 (13-bit timer mode)
   (b) Using Timer 0 is mode 1 (16-bit timer mode)
   (c) Using Timer 1 in mode 2 (8-bit auto-reload timer mode)
   (d) Using Timer 0 (TL0) in mode 3 (8-bit split timer mode)

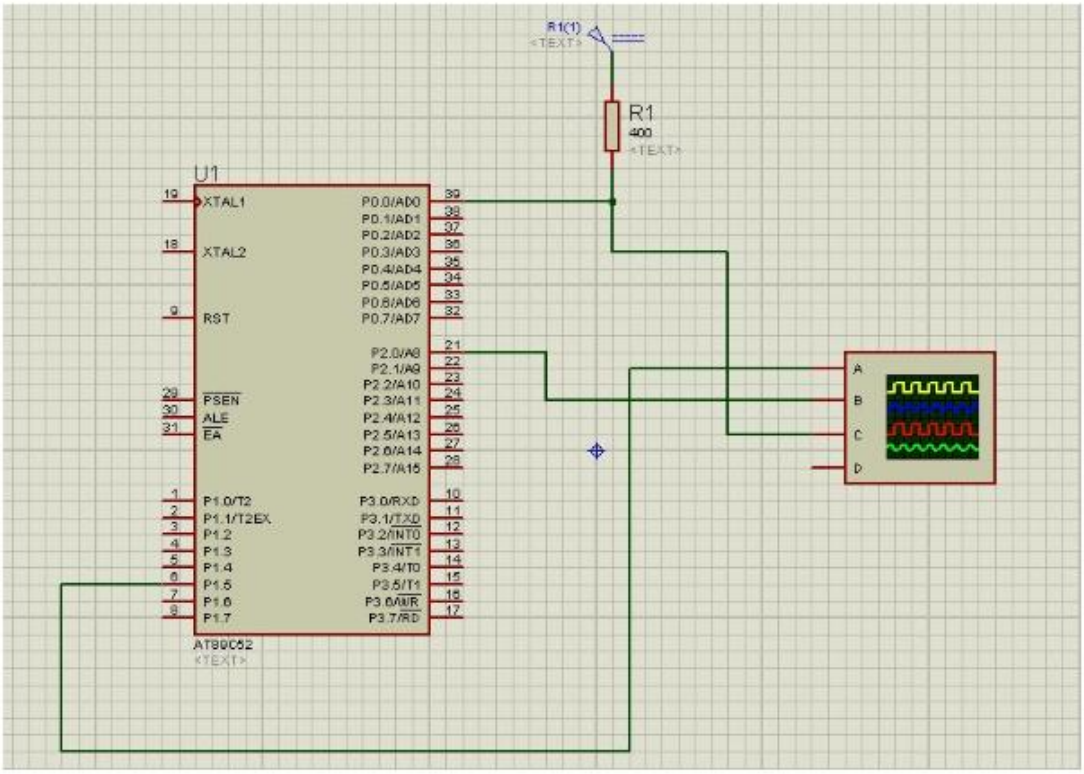

Figure 8: Proteus circuit used to observe result

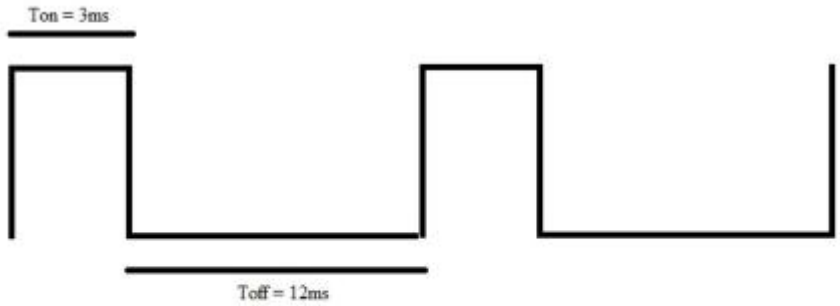

Figure 9: Qno.1 Waveform to be produced

2. Generate the periodic waveform as shown in figure 11. The waveform should be produced at pin zero of port zero (P0.0). The XTAL frequency is 11.0592 MHz. Observe the waveform on an oscilloscope and measure the ON and OFF times.
   (a) Using Timer 0 and mode 0 (13-bit timer mode)
   (b) Using Timer 1 in mode 1 (16-bit timer mode)
   (c) Using Timer 0 in mode 2 (8-bit auto-reload timer mode)
   (d) Using Timer 0 (TH0) in mode 3 (8-bit split timer mode)

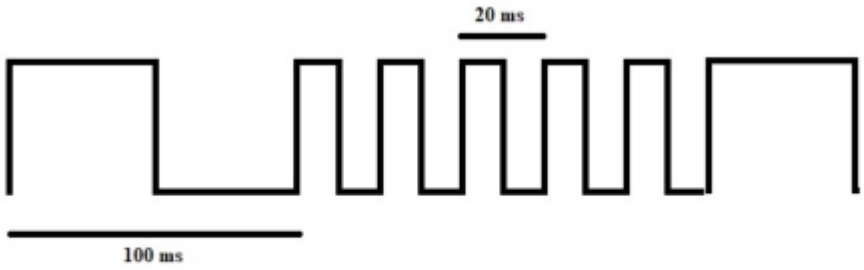

Figure 11: Qno.2 Waveform to be produced

3. Design a digital minutes and seconds in double digit format. The clock should count from 00:00 to 59:59 and repeat. Time should be displayed in decimal format using four 7-segment LED units. A decimal point should separate minutes from seconds. Use an appropriate timer and timer mode. Use port 0 (P0) to send data to 7-segment LED units. Use transistors as switches to activate or deactivate the 7-segment LED units using pins 0, 1, 2 and 3 of port 2 (P2.0, P2.1, P2.2, P2.3).