Chapter 10

Programming in C

Lesson 09

C Programming Examples for Timers

Two timers, T0 and T1

- TH0, TL0, THL1 and TL1 for holding time/count values
- 8052 version has three timers T0, T1 and T2 and six registers TH0, TL0, THL1, TL1, TH2 and TL2 for holding bytes for the time/count values.

SFR for control of the timer T0 and T1 functions

- TCON
- It also has the status its for T0 and T1
- T2CON in 8052 for control and status bits for T2 and selects the functions of T2

SFR to defined modes of the timer T0 and T1 functions

- TMOD

Programming a timer

- Programming the TCON and TMOD bits
- Loading the appropriate count variable c0 as per the intervals of the clock inputs *deltaT* to the timer/counter

Program to specify the 4-bits of timer T0 mode 2

```
#include <reg51.h> /* Include header file for the
   registers and SFRs of 8051. */
void main (void)
 {    /* 3rd bit b3 is GATE_T0, 2nd bit b2 is C-T0, 1st
   and 0th bit-pair is for specifying mode */
     TMOD = 0x02; /*Assign the timer T0 start-stop
   external gate pin inactive, timer T0 using  internal
   clock inputs and timer T0 mode = 2*/
}
```

Mode 2 of T0

TH0 loads automatically into TL0 after each overflow

TL0 does 8-bit counting from loaded value to overflow

C statements to specify the timer control bit for timer T0 stop

```
sbit t0Start = TCON^4;/*declare variable t0Start
   address as the fourth bit address in SFR P2. */
t0Start = 0;
         or
TR0 = 0;
```

Calculation for 8-bits in TH1 and TL1 in mode 2 for delay = 220 µs

- *Assume 12 MHz Xtal and classic 8051*
- 8-bit timer Mode 2 is used when TH0 is used to reload after overflow the same value in TL0. Internal clock input period = 1 µs, the maximum delay = 256 µs, it is when timer is loaded with 0x00 to start with.
- Number of clock inputs in 220 µs required = 220 µs /1 µs = 220.
- TH1 and TL1 are loaded same and = (256 –220) = 36 = 0x24. TH1 needs to load counts = 0x24. TL1 needs to load 0x24.

C program

```
# include <reg51>
void main (void)
{/* Write statement for writing TMOD for T1 mode 2.*/
TH1 = 0x24; TL1 = 0x24;
 /* Write statement for starting T1.*/
}
```

Calculations for 16-bits in TH0-TL0 in mode 1 for delay = 1s

- Assume 11.0592 MHz Xtal and classic 8051
- When crystal frequency = 11.0592 MHz, the internal clock input period = 1.085070 µs.
- 16-bit timer Mode 1 used
- Because when the internal clock input period = 1.08507 µs, then the maximum delay 8-bit timer case = 277.78 µs only

Calculations for 16-bits in TH0-TL0 in mode 1 for delay = 1s

- Number of clock inputs in 1 s = 1000000 µs /1.08507 µs = 921600 = 917504 + 4096= (14× 256 × 256 + 4096) = (14 × 0xFFFF + 0x1000)

Calculations for 16-bits in TH0-TL0 in mode 1 for delay = 1s

- 16-bit TH0-TL0 pair is loaded with 0
- Then after 14 overflows, it is loaded with (65736 – 4096) = – 4096 = = – (16 × 256) = (0x10000 – 0x1000) counts = 0xF000 for the case of 1 s delay

C program preprocessor directives

- # include <reg51>
- int numOV;
- int iov;

C program main

- void main (void)
- { /* Write statement for writing TMOD for T0 in mode 1.*/
- numOV = 14;
- iov = 0;
- TH0 = 0x00; TL0 = 0x00; TR0 =1;
- . /* Write statement for setting TR0.*/
- }

Interrupt function

- if (iov <= numOV) /* Condition test for number of overflows less or equal to 14 */
- {iov ++;   /* increment iov */}
- else if (iov= = numOV + 1)
- {TH0 = 0xF0; TL0 = 0x00; iov=++;};
- if (iov = = numOV + 2) {TR0 = 0; iov= 0; }; /* Stop Timer 0. Reset iov */

Alternative Interrupt function

- if (iov <= numOV) /* Condition test for number of overflows less or equal to 14 */
- {iov ++;   /* increment iov */}
- else {
- {TH0 = –16; TL0 = 0; iov = 0;}
- }

C program for an interrupt function for generating square pulses at pin 0 of P2

- Use Timer1ISR and generate pulses at pin 0 of P2 at 1 kHz (pulse interval = 1 ms)
- Counts the number of overflow from timer T1 in mode 2
- T1 in mode 2 overflows after every 250 μs

Preprocessor Directives

- #include <reg51.h> /* Include header file for the registers and SFRs of 8051. */
- sbit pin0P2 = P2^0; /* pin0P2 is SFR bit. It is b0 bit in P2.*/

Main

```
{ unsigned int numOVT0;
 unsigned int num_ms;
 unsigned int num_s; /* Assign initial values 0*/
 numOV = 0;
 /* Code for specifying T0 in mode 2 and
  overflow after every 250 μs.*/
EA = 1; /* Enable all primary level bit*/
```

Enable T1 interrupts

```
ET1 =1; /* Enable timer 1 interrupt bit */
while (1) {  /* Wait endlessly */
 ; }             /* End of the while loop */
}               /* End of the main */
} /* End of the main */
```

Timer 1 ISR interrupt function for T1 with use of the bank 1

```
void timer1ISR (void) interrupt 1 using 1  {
  If (numOV <2) {
  numOV ++;} else  /* Increment numOV */
  {pin0P2 = ~ pin0P2; /* 500 µs over. Therefore
   complement pin0P2 output.  If ANSC  C99
   compiler compliant then statement is pin0P2 =
   ! pin0P2 */
```

Reset Number of Overflow

```
numOV =0; };  ./* Reset numOV for next in-between pulse duration of 500 µs */
        } /* End of interrupt function for timer 0 */}
```

Summary

We learnt

- Programming TMOD
- Programming TCOM
- Start and stop a timer
- C program for 1 s delay
- Assigning a Bank to an Interrupt Function
- Square pulse generation program

End of Lesson 09 on

C Programming Examples for Timers