Chapter 09

Programming in Assembly

Lesson 05

Programming Examples for Timers

Programming TMOD Register

Write instructions to run T0 in Mode 0, external count inputs, internal start/stop control

- ANL TMOD, # F4
- ; Using AND operation upper 4 bits will not change in
- ; TMOD. Lower 4-bits will change = 0100. GATET0 = 0,
- ;C/T0 = 1, M1T0; = 0 and M0 T0 = 0
- ;When defining a mode do not change the mode of T1

Write instructions to run T0 in Mode 1 internal clock inputs and internal start/stop control

- ANL TMOD, # F1
- ; Using AND operation upper 4 bits will not change in
- ; TMOD. Lower 4-bits will change = 0001. GATET0 = 0,
- ;C/TT0 = 0, M1T0; = 0 and M0 T0 = 1.
- ;When defining a mode do not change the mode of T1

Write instructions to run T0 in mode 2 internal clock inputs

- ANL TMOD, # F4; Using AND operation upper 4 bits will not change in
- ; TMOD. Lower 4-bits will change = 0010. GATET0 = 0,
- ;C/TT0 = 0, M1T0; = 1 and M0 T0 = 0
- ;When defining a mode do not change the mode of T1

Write instructions to run TH0 and TL0 independent timer in mode 3 internal clock inputs

- ANL TMOD, # F3
- ; Using AND operation upper and lower 4 bits will change in
- ; TMOD. Lower 4-bits will change = 0011. GATET0 = 0,
- ;C/TT0 = 0, M1T0; = 1 and M0 T0 = 1
- ; When defining a mode do not change the mode of T1

Programming Time Interval for the Overflow and Timer Interrupt using TH0, TL0, TH1 andTL1

Internal clock input intervals and frequency of internal clock to the timers in 8051

- 12 MHz
- Each internal clock interval to the timer will be equal to 1 μs

Internal clock input intervals and frequency of internal clock to the timers in 8051

- Xtal frequency 24 MHz
- Internal clock input intervals 0.5 μs

Internal clock input intervals and frequency of internal clock to the timers in 8051

- XTAL frequency = 11.0592 MHz
- then Internal clock input intervals = 1.085 μs.

Maximum interval for the overflow the timer T0

- 12 MHz Xtal used

Mode 0: $2^{13} \times 1$ μs.

Mode 1: $2^{16} \times 1$ μs.

Mode 2: $2^{8} \times 1$ μs.

Mode 3 TL0: $2^{8} \times 1$ μs.

Mode 3 TH0: $2^{8} \times 1$ μs.

Timer higher byte TH0 with 80H in mode 2

- MOV TH0, #80H; TH0 = 128;
- MOV TL0, #80H; TL0 = 128;
- SETB TR0; Timer T0 runs (28 – 80H) × 1
- ; μs= 128 μs. TL0 overflows in 128 μs
- ;and TH0 reloads into TL0 after every 128 μs [12 MHz XTAL]

Timer with 40H and 80H in mode 3 TH0 and TL0

- MOV TH0, #40H;
- MOV TL0, #80H;
- SETB TR0;
- ;TH0 overflows after $(2^8 - 40H) \times 1\ \mu s$
- ;= 192 μs. TL0 overflows after
- ;$(2^8 - 80H) \times 1\ \mu s$= 128 μs [12 MHz XTAL]

Timer with 8000H in mode 1

- MOV TH0, #80H;
- MOV TL0, #00H;
- SETB TR0
- ;TH0 overflows after (2^{16} – 32768) × 1 μs=
  ;32.768 ms [12 MHz XTAL]

Programming TCON

Using TCON bit for the starting timer T0

- Starting timer T0: SETB TCON.4 or SETB TR0 or SETB 8CH

Using TCON bits for the starting timer T1?

- Starting timer T1: SETB TCON.6 or SETB TR1 or SETB 8EH

Using TCON flag for the overflow of timer T0

- Resetting timer T0: CLR TCON.5; or
- CLR TF0; or
- CLR8DH
- ;Some time a program waits for TF0 to set
  ;and then an instruction clears it after the
  ;necessary action. ISR for T0 is not used in
  ;this case. Resetting the flag enables
  ;response to next T0 interrupt and next
  ;overflow.

Using TCON bits for the starting timer T0

- Resetting timer T1: CLR TCON.7; or
- CLR TF1; or
- CLR8FH
- ;Some time a program waits for TF1 to set ;and then an instruction clears it after the ;necessary action, ISR for T1 is not used in ;this case.
- ;Resetting the flag enables response to next ;T1 interrupt and next overflow

Programming for program for overflow and interrupt after 4.096 ms and setting output at P2.6 pin after the overflow and interrupt after 4.096 ms

Solution

- When Xtal frequency = 12 MHz, the internal clock intervals = 1 μs
- Overflow required after 4.096 ms
- It means after 4.096 ms/ 1μs = 4096 internal clock input = 1000H = $128 \times 32 = 2^7 \times 2^5 = 2^{12}$ clock inputs.
- Use timer T0 mode 0
- ISR for timer T0 will stop the Timer 0 after the overflows and interrupt and set port bit P2.6

Program (Main Program)

(i) ANL TMOD, #0F0 ; Define T0 mode 0, internal clock inputs and internal start/stop

(ii) MOV TH0, #80H ; TH0 overflows after the 128 clock inputs

(iii) MOV TL0, #00H ; TL0 overflows after 32 clock inputs

Program (Main Program)

```
(iv)  SETB EA          ; Enable interrupts
(v)   SETB ET0         ; Enable T0 interrupts
(vi)  SETB TR0         ; Run timer T0
(vii)  SJMP – 2        ; Wait for ever
```

Interrupt Service Routine for T0

(i) CLR TR0 ; Stop T0

(ii) CLR ET0 ; Disable T0 interrupts. en

(iii) SETB P2^6 ; Set bit 0AEH can be used in place of P2^6. 0AEH is bit address of P2.6 pin

(iv) RETI ; Return from Interrupt

Program for a timer in mode 2 for the successive interrupts after every 250 μs and setting and resetting (toggling) of the P2.6 pin output on overflow [Square Pulses]

Program (Main Program)

(i) ANL TMOD, #F2 ; Define T0 mode 2, internal clock inputs and internal start/stop

(ii) MOV TL0, #06H ; TL0 overflows after 150 clock inputs

(iii) MOV TH0, #06H ; TH0 loads in TL0 after 150 clock inputs

Program (Main Program)

```
(iv) SETB EA          ; Enable interrupts
(v) SETB ET0          ; Enable T0 interrupts
(vi) SETB TR0          ; Run timer T0
(vii) SJMP – 2          ; Wait for ever
```

Interrupt Service Routine for T0

(i) CPL P2^6 ; CPL at 0AEH can be used in place of P2^6. 0AEH is bit address of P2.6 pin

(ii) RETI ; Return from Interrupts

Summary

We learnt

- Programming the TMOD register
- Programming the T0 and T1 registers
- Programming the TCON register
- Programs for mode 0 and mode 2

End of Lesson 05 on

Programming Examples for Timers