임베디드 기초 프로그래밍
(마이크로프로세서 실습)
ch4. 기초실험(3)

정보네트워크공학전공
이훈재
hjlee@dongseo.ac.kr
http://crypto.dongseo.ac.kr
http://kowon.dongseo.ac.kr/~hjlee

4.6.1 Mode 0에서의 TIMER 0의 실험
- 실험#5. 회로도

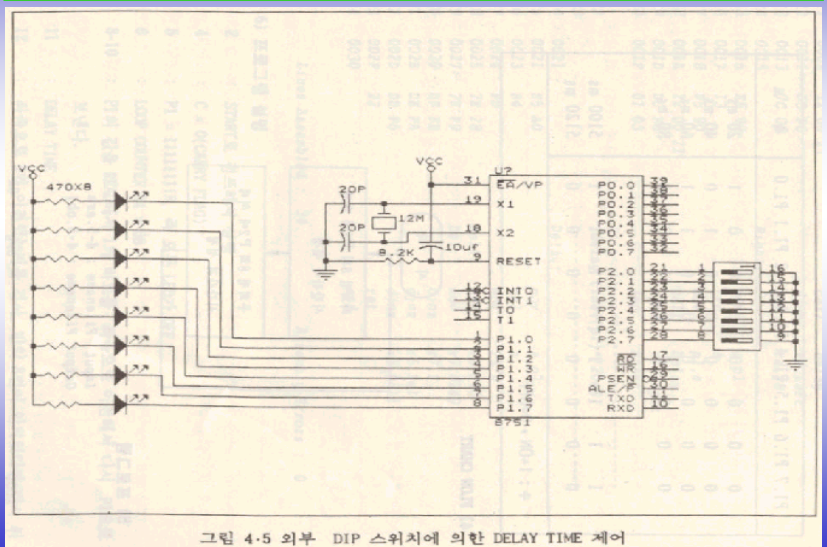
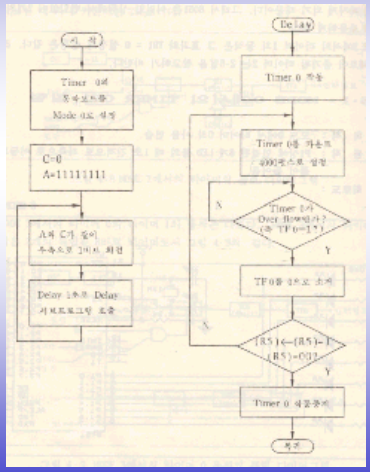


그림 4-5 외부 DIP 스위치에 의한 DELAY TIME 제어

4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험#5. Flow Chart



4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험 #5. ASM Coding

```
  org   0h
  jmp  Start
;
Start:
  mov  tmod,#00000000b  ; 타이머 0 모드
  clr    c  ; CARRY FLAG C = 0
  mov  a,#11111111b
;
;
Loop:
  rlc     a
  mov  p1,a
;
  mov  r5,#250
  call   Delay
;
  jmp   Loop
```

```
;
;---------------------------------------
;       DELAY TIME = r5 X (5msec) @ 12 MHz
;---------------------------------------
Delay:
        setb tr0 ; 타이머 0의 시동
Again:
        mov  t10,#(8192-4000) mod 32
                ;start=4192       ; 나머지(%32)

        mov  th0,#(8192-4000)/32 ; 몫(%32)
Wait:
        jbc  th0,TimeOut  ; 타이머 0 = full ?
        jmp  Wait
TimeOut:
        djnz r5,Again
        clr  tr0  ; 타이머 0의 정지
        ret  ; 호출 프로그램으로 복귀

        end
```

4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험 #5. 타이머 동작모드-TMOD 레지스터

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

GATE	Gating control when set. Timer/Counter "x" is enabled only while "$\overline{INT}$x" pin is high and "TRx" control pin is set. when cleared Timer "x" is enabled whenever "TRx" control bit is set.
C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from in=ternal system clock.) Set for Counter operation (input from "Tx" input pin).

M1	M0	OPERATING
0	0	8048 Timer "TLx" serves as 5-bit prescaler.
0	1	16-bit Timer/Counter "THx" and "TLx" are cascaded; there is no prescaler.
1	0	8-bit auto-reload Timer/Counter "THx" holds a value which is to be reloaded into "TLx" each time it overflows.
1	1	(Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	(Timer 1) Timer/Counter 1 stopped.

SU00534

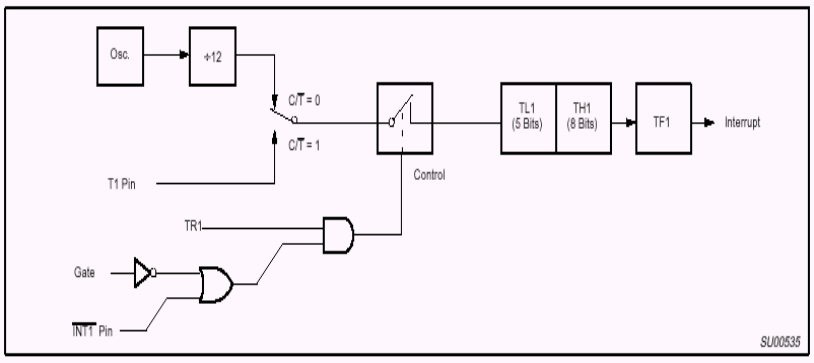
4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험#5. 타이머 동작모드-TCON 레지스터

MSB							LSB
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

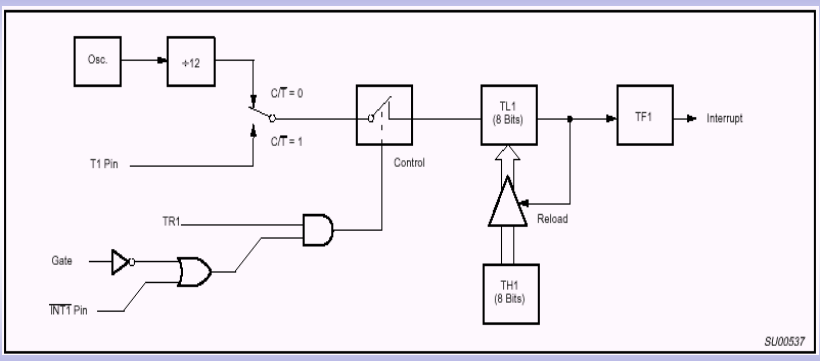
BIT	SYMBOL	FUNCTION
TCON.7	TF1	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or clearing the bit in software.
TCON.6	TR1	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.5	TF0	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine, or by clearing the bit in software.
TCON.4	TR0	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.
TCON.3	IE1	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.2	IT1	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TCON.1	IE0	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TCON.0	IT0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

SU00536

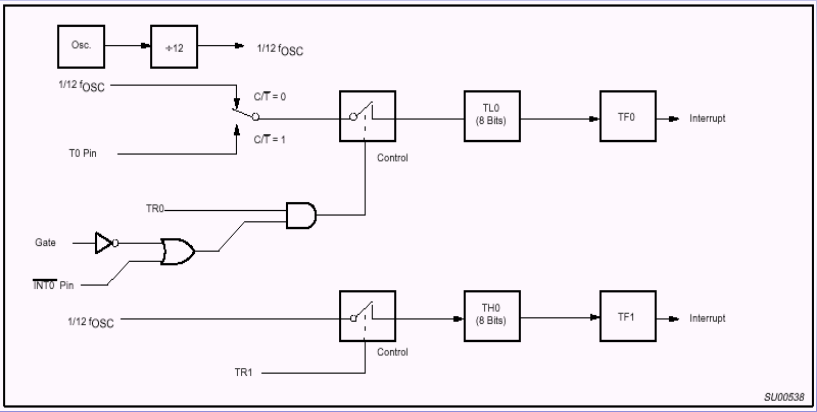
4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험#5. 타이머 동작모드-Mode 0



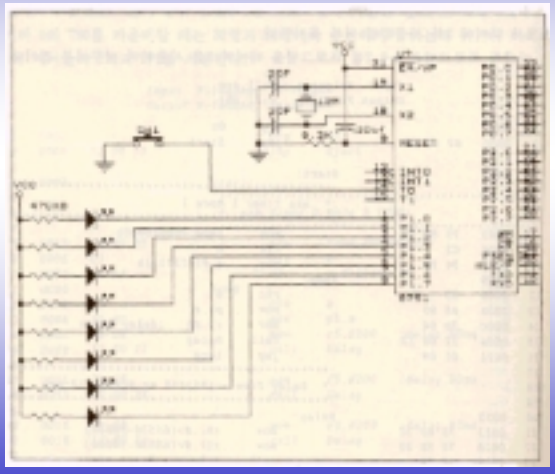
4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험#5. 타이머 동작모드-Mode 2



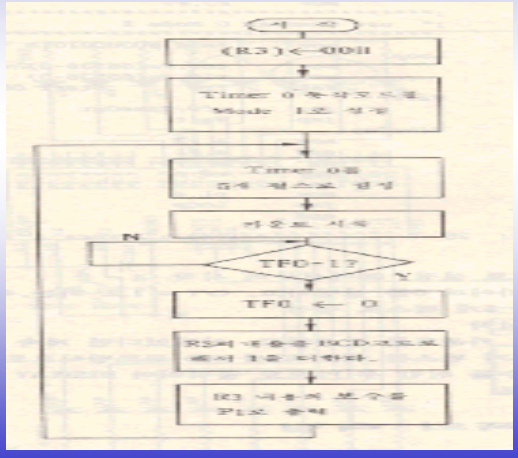
4.6.1 Mode 0에서의 TIMER 0의 실험 - 실험#5. 타이머 동작모드-Mode 3



4.6.5 Mode 1에서의 TIMER 0의 카운터 실험 - 실험#5. 회로도



4.6.5 Mode 1에서의 TIMER 0의 카운터실험 - 실험#5. Flow Chart



4.6.5 Mode 1에서의 TIMER 0의 카운터실험 - 실험#5. ASM Coding

```
   org   0h
   jmp  Start
;
Start:
    mov  r3,#0  ; (R3) = 00H
    mov  tmod,#00000101b  ; 모드 01로 타이머 0;
                                ; C/~T = 1
Loop:
     mov  th0,#(65536 - 5)/256 ; Start =65531
     mov  tl0,#(65536 - 5) mod 256
     setb  tr0                    ; 카운터 action start
Wait:
      jbc   tf0,TimeOut          ;
      jmp  Wait
```

```
TimeOut:
       mov  a,r3  ; 카운터로 5개의 펄스 카운트
       add   a,#1   ; 디스플레이 숫자 증가
       da     a
       mov   r3,a
        cpl    a       ; R3값의 보수, P1 디스플레이
        mov  p1,a
        jmp   Loop  ;


       end
```