

실험 11. 시계 기능 구현

기본 기능이 동작되는 조는 시계용 발진기와 SUPER CAPACITOR, DS1302를 받아서 조립한 후에 프로그램하여 완전한 시계 기능을 구현하는 프로그램을 하도록 한다.

1. 표시기 동작시키기

회로의 표시기를 동작시키기 위해서는 scanning 방법이 사용되며, 이를 가장 쉽게 효율적으로 할 수 있는 방법이 interrupt를 이용한 방법이다. 우리가 앞 실험에서 사용하였던 약 2msec 주기의 interrupt를 사용하면 표시뿐 만 아니라 초를 나타내는 dot on/off 시간과 key를 확인하기 위한 key 눌러진 시간 확인용 기준 시간을 만들 수 있다.

- 표시기는 매 interrupt 마다 다음 위치가 지정되도록 표시위치 지정 변수를 증가시킨다.
- 표시되는 값은 특정 변수(표시기변수)에 있는 내용을 RB port에 출력한다. 이 경우 RB port의 값이 개별적으로 변하므로 미리 RB port 변수를 설정하여 변수에서 변경 후 이 값을 RB port로 출력하면 편리하다.
- Key 입력은 앞에서 실험해 보았던 방법을 사용하며되고, 다만 key의 동작 시간을 정하기 위하여 동일 key가 눌러진 시간을 계수(KEY_CNT)하는 변수를 두어, 이 값이 특정 크기 이상이 되면 key로 인식 key input 변수를 설정한다. 그리고 KEY_CNT 변수는 더 이상 증가시키지 않는다.(이유를 이해할 수 있나요? 계속 증가하면 다시 0이 되어 동작이 불규칙해 집니다.) 실습용으로 만들어진 것은 약 0.5초로 확인한다. 그리고 main에서는 key input 변수가 설정되어 있으면, 그 key 에 해당하는 일을 하면 된다. 이때 key가 여러번 인식되는 것을 막기 위해서 KEY_CNT 변수는 clear시킨다.
- 시간 설정시 설정 위치를 알려 주기 위해서는 깜박임을 사용하고 있다.(다른 방법도 가능하다. 다만 한 예로 생각하십시오.) 이것은 특정 key가 들어오면 설정모드로 이동하고 설정 모드의 변수값에 따라서 특정 위치가 깜박

이도록 하면된다.(실습용에서는 약 .1초 단위로 깜박거리게 하였다.) 그리고 그 위치에서 변경 key를 확인하여 값을 변경하고 저장되도록 한다. 설정 모드에서 위치를 선택하는 방법도 7-segment 표시기 동작과 같은 개념으로 하면 된다.

- 이렇게 많은 기준 시간이 필요하며, 이 경우 변수를 서로 다르게 설정하지 말고 INT_CNT 변수를 2의 n 승 단위(bit 단위)로 확인하면 주기적이 되면서 기간을 여러개 만들 수 있다.

2. LED 동작시키기

회로에 구현된 표시해야할 내용은 LED 3개와 4-digit 7-segment로 총 5개이다. 이중 LED는 동작시킬 때 약간은 복잡하다. LED1은 RB4에 직접 연결되어 있으므로 바로 on/off 제어 가능하나, LED2,3은 CD4511의 출력 a,b에 연결되어 있으므로 바로 on/off 제어를 할 수 없다. 즉 RB3,2,1,0의 값(BCD의 출력)이 얼마일 때 a,b의 출력을 개별적으로 on/off 제어 가능한가를 알아야 한다. CD4511의 data sheet를 잘 확인해 보면 다음과 같음을 알 수가 있다.

- LED1: RB4
- LED2: on, LED3: off ---> RB3,2,1,0 = 5
- LED2: off, LED3: on ---> RB3,2,1,0 = 1
- LED2: on, LED3: on ---> RB3,2,1,0 = 7
- LED2: off, LED3: off ---> RB3,2,1,0 = 6

따라서 원하는 LED를 on/off 시키려면 위의 값을 RB4,3,2,1,0에 출력해야한다.

3. 부저 울리기

회로에 사용된 부저는 I/O port의 부족으로 SN74145의 출력에 연결되어 있다. 그리고 부저가 동작하기 위해서는 SN74145의 6번 pin이 low가 되어야 한다. SN74145는 BCD to 10 decoder로 BCD 값이 들어오면 해당 값의 출력이 low가 된다. 즉 부저를 울리기 위해서는 RB7,6,5에 5의 값이 출력되면 된다. 그러나 이 값을 계속해서 출력하면 부저는 소리를 내게되나, 7-segment는 전부 off가 된다. 즉 부저도 scanning에 의해서 소리가 나도록 해야 한다. 이렇게 하면 부저가 울리는 시간이 전체 시간의 1/6에 해당되어 소리가 크게 나오지 않으며, 소리도 불안정하다. 따라서 부저가 동작되는 시간을 늘려 줄 필요가 있다. 가장 쉽게 할 수 있는 방법 중의 하나가 LED, DG1, 부저, DG2, DG3, 부저, DG4, 부저 순으로 scanning 하면, 총 3/8에 해당하는 시간이 부저에 할당되므로 충분히 사용 가능한 소리가 나온다. 부저를 울리지 않기 위해서는 SN74145의 6번 pin이 high가 되어야 한다. 즉 BCD 값으로 6 이상을 출력하면 된다. 여기서 scanning을 총 8개로 한 것은 여러 가지로 편리한 점이 있다.

4. 시계 값 읽어오기

시계의 모든 정보는 DS1302에 들어 있다. DS1302는 외부의 정밀한 수정 발진자를 이용하여 표준시간을 얻고 이것을 분주하여 시계를 만드는 전용 칩이다. 그리고 얻어진 모든 시간은 내부의 register에 저장된다. 따라서 우리는 이 register에 들어 있는 시간값을 읽어와 표시기에 뿌려주면 된다. 먼저 DS1302의 기능을 이해하고 사용하려면 data sheet을 보아야 한다. 모든 것이 눈에 들어오면 이해하기가 쉽다. 따라서 기능을 말로만 설명하는 것이 아니고 그림으로 함축적으로 표현한다. 이것이 block diagram 이다. 2 page의 그림1을 보면 real time clock와 31*8bit의 RAM을 가지며, I/O pin과 SCLK라는 pin을 통하여 직렬 통신함을 알 수가 있다. 그러면 다음으로 데이터를 읽어오려면 통신 방법을 이해해야 한다. 통신은 신호를 순서대로 내 보내므로 이루어지므로 신호의 시간에 따른 신호 파형을 보면 쉽게 이해할 수가 있다. 이런

그림을 timing diagram 이라고 한다. 11 page의 그림5 이다. 먼저 RST가 1로 되어야 통신이 가능함을 알 수 있고, (통신 중임을 나타내는 신호) 그리고 읽고자하는 또는 쓰고자하는 기능 및 주소가(command & address) 출력되고 그것에 따른 결과가 뒤따르게 된다. I/O은 0 bit부터 출력되며, 출력된 값이 DS1302의 내부에 기억되도록하는 신호가 SCLK이며, 그림에서 보면, rising edge에서 일어남을 알 수 있다. 따라서 I/O에 data를 출력하고 SCLK를 0에서 1로 변경시키고 다시 0으로 만든후 다시 다음 데이터를 출력하는 것을 반복하면된다. 이렇게 bit 7까지 출력되면, 하나의 command가 DS1302에 전달된 것이므로 DS1302는 그에 따른 동작을 하게된다. 읽기에서는 데이터가 출력(DS1302 입장에서) 되므로 I/O pin을 input port에서 output으로 변경하고 출력을 하게된다. 따라서 cpu 입장에서는 I/O pin이 연결된 RA2를 output port에서 input port로 변경하고 들어오는 데이터를 읽어야 한다. 이때 중간에 발생하는 port 변화를 잘 이해해야만 정확한 데이터가 전달된다. Data sheet를 잘 확인해보면 command bit 7의 출력이 끝나고 SCLK를 1에서 0으로 떨어뜨리면 DS1302가 output으로 변경되어 신호를 출력한다고 되어 있으므로, 우리는 SCLK가 1일 때 RA2 port를 input으로 변경하고 SCLK를 0으로 변경 후 (DS1302가 데이터를 출력함) 데이터를 읽고 SCLK를 1로 만들거나, SCLK를 1로 만들고 데이터를 읽으면 된다.

5. DS1302에 값 쓰기

DS1302에 시간 값을 기록하는 것은 read 보다는 간편하다. 왜냐하면 I/O port 변경을 시키지 않기 때문이다. 따라서 read 부분의 설명을 참조하기 바란다. 다만 시간용 register 외에 별개의 31개의 8bit RAM이 존재하면 읽고 쓸 수가 있다. 따라서 alarm 시간 설정용 시간값 저장용으로 사용할 수 있으며, 따라서 PIC16F84에 들어 있는 EEPORM를 사용하지 않아도 된다.

6. DS1302를 제어하기 위한 기능

시계 값은 어디에 어떻게 저장되는냐는 7 page의 그림 4에 표시되어 있으며, 이 중 sec의 data 중 bit7인 CH bit는 clock halt flag이며, 이 값이 1로 되어 있으면 시계가 정지되므로 항상 처음에 0으로 기억시켜 주어야 한다. 이때 무작정 0으로 기억시키지 말고 현재의 sec 시간을 읽어서 그중 bit 7만 0으로 만들어 출력해야한다. (이 이유를 알 수 있는 사람은 보고서를 작성시 잘 설명하여 보너스를 받도록 하시오)

그리고 기어된 내용을 변경하려고 하면, 이 명령이 믿을 만한가를 검사해야 비정상적인 동작을 억제 할 수가 있다. 우리가 PIC16F84에서 실험해 본 AA 55를 순서대로 기억시키는 것도 그런 예임을 확인하였는데, DS1302에서는 별도의 write protect bit를 두고 있다. 7 page의 control code의 data bit 7이 이 기능이며, 이 값이 1이면 write protect가 되며, 0이면 write가 가능하다. 즉 '00000000' 과 '10000000' 코드 만을 인식하도록 하고 있다.

또 main 전원이 정전되어도 시계는 정상적으로 동작되도록 하기 위하여 별도의 전원이 필요하다. 여러분이 흔히 사용하는 시계 전원으로는 수은 전지가 사용된다. 수은 전지는 전류 밀도가 높아 적은 부피에 큰 에너지를 저장할 수 있다. 그러나 재충전은 불가능하다. 이는 특정 시간이 지나면 무조건적으로 전원이 없어져 기억되지 못한다는 것을 의미한다. 여러분이 PC 컴퓨터에 들어간 시계가 죽어 또는 CMOS RAM이 지워져 부팅시 귀찮은 경우를 당해 보았겠지요. 그러면 충전이 되는 전지를 사용하면 되지 않을까? 하고 생각되지요. 그러나 충전이 되는 전지는 일반적으로 누설전류가 크게 되며, 이는 사용하지 않아도 내부적으로 전류가 흘러 에너지가 없어진다는 것을 의미합니다. 여러분이 휴대폰 전원이 사용하지 않아도 없어지는 것과 같은 이유이지요. 따라서 항상 상용 전원을 사용하고 정전시만 비상용으로 사용되는 전원이 필요한 경우 사용할 만한 전원 장치가 없을까요? 전기를 직접 저장하는 capacitor가 있지요. 그러나 일반 capacitor는 역시 누설 전류가 커 장기간 사용이 불가능하지요. 따라서 특수하게 만들어진 capacitor를 필요로 합니다. 이것을 super capacitor라고 합니다. 용량도 일반 capacitor와는 비교도 안될 정도로 큼니다. 동일 부피에 약 100 이상은 되며, 따라서 충전시나 방전시 주의해서 사용하지 않으면, 화재 등의 위험이 발생할 수 있습니다. 따라서 DS1302에서도 충전시

충전 전류를 제어하기 위한 회로를 내부에 가지고 있으며, data sheet에서는 trickle charge 라고 명명되어 있지요. 이것도 함부로 설정이 변경되면 문제가 발생할 여지가 있으므로 특별한 코드가 아니면 동작되지 않도록 하고 있습니다. 자세한 것은 data sheet를 잘 읽어 보고, 우리는 conrtol data로 '10101010'를 사용하도록 합시다.

7. 프로그램 작성시 주의 사항

시계를 완벽하게 동작시키기 위해서는 프로그램 길이가 약 1k 정도 가 된다. 그리고 모드변화나 scanning 시 ADDWF PCL,F 라는 명령어를 사용하여 table을 만들어 사용하는 것이 늘어나게 된다. 이때 예상하지 않은(?) 이상 상황이 발생할 수도 있다. 즉 table에서 가라고 한 위치로 가지 않고 엉뚱한 위치로 이동되는 현상이다. 이는 ADDWF PCL,F의 명령어를 완전히 이해하지 못해서 생긴 일이다. 이 명령어는 PC의 low byte(8bit임)인 PCL과 W register의 내용을 더하여 PCL에 넣어라는 것이다. 이때 만약 더한 결과가 자리 올림이 발생하면 어떻게 될까? 자리 올림 값이 PC의 high byte에 자동 저장되면 문제가 발생하지 않을 것이다. 그러나 그렇지 않다. 그냥 없어져 버린다. 따라서 이 경우 앞쪽 번지로 이동되게 된다. 이 것을 해결하기 위해서는 무조건 자리 올림이 발생되지 않도록 해야하며, 따라서 0번지에서 256번지 이내에서 table을 만들어야 한다. 그리고 table에 GOTO 문장을 사용하면 된다. 만약 다른 번지에서 사용하고 쉽다면 PC의 high 내용을 변경시켜 주어야 하며, resister 중 PCLATH가 이 목적으로 있다. 의미를 data sheet에서 확인하고, 사용해 보도록 하세요.

8. DS1302의 내용을 읽어오는 부 프로그램

```

;-----
;;;; DS1302 serial data read : address out & read data (data sheet p11
fig5) ;
;; 들어갈 때 W register=address, 나올 때 W register=DS_RDATA
;; used buffer:(address: DS_ADDRESS, data: DS_RDATA, bit counter:
DS_CNT)
;; 초기조건: RST=0, SCLK=0, I/O=INPUT
DS_READW MOVWF          DS_ADDRESS
          ;BCF  INTCON,GIE ; GLOBAL INTERRUPT DISABLE BIT

          BSF      PORTA,3      ; RST=1
          ; serial address out
          BSF      STATUS,RP0 ; RAM BANK 1 선택
          BCF      TRISA, 2      ; RA2 OUTPUT
          BCF      STATUS,RP0 ; RAM BANK 1 선택
          MOVLW    8              ; CONTROL DATA BIT
          MOVWF    DS_CNT
DS_RALP   BCF      PORTA, 1      ; SCLK=0
          BTFSS    DS_ADDRESS,0 ; 전송 데이터 0 bit 확인
          GOTO     DS_RAO0
          BSF      PORTA, 2      ; DATA OUT LSB
          GOTO     DS_RAO1
DS_RAO0   BCF      PORTA, 2
DS_RAO1   BSF      PORTA, 1      ; SCLK=1
          RRF      DS_ADDRESS,F
          DECFSZ   DS_CNT,F
          GOTO     DS_RALP
          ; serial data read -- port change
          MOVLW    8              ; CONTROL DATA BIT
          MOVWF    DS_CNT
          BSF      STATUS,RP0 ; RAM BANK 1 선택

```


9. DS1302에 기억시키는 부 프로그램

```

;-----
;;; DS1302 serial data write : address out & write data (data sheet p11
fig6) ;
;; used buffer:(address: DS_ADDRESS, data: DS_WDATA, bit counter:
DS_CNT)
;; 초기조건: RST=0, SCLK=0, I/O=INPUT
DS_WRITE  MOVWF          DS_ADDRESS
          BCF    INTCON,GIE ; GLOBAL INTERRUPT DISABLE BIT
          BSF     PORTA,3    ; RST=1
          ; serial address out
          BSF     STATUS,RP0 ; RAM BANK 1 선택
          BCF     TRISA, 2    ; RA2 OUTPUT
          BCF     STATUS,RP0 ; RAM BANK 1 선택
          MOVLW   8           ; CONTROL DATA BIT
          MOVWF   DS_CNT
DS_WALP   BCF     PORTA, 1    ; SCLK=0
          BTFSS   DS_ADDRESS,0 ; 전송 데이터 0bit 확인
          GOTO    DS_WAO0
          BSF     PORTA, 2    ; DATA OUT LSB
          GOTO    DS_WAO1
DS_WAO0   BCF     PORTA, 2
DS_WAO1   BSF     PORTA, 1    ; SCLK=1 (DATA SAVE)
          RRF     DS_ADDRESS,F
          DECFSZ  DS_CNT,F
          GOTO    DS_WALP
          ; serial data write
          MOVLW   8
          MOVWF   DS_CNT
DS_WLP    BCF     PORTA,1     ; SCLK=0
          BTFSS   DS_WDATA, 0 ; 전송 데이터 0bit 확인
          GOTO    DS_WOUT0

```

```

        BSF      PORTA,2
        GOTO     DS_WOUT1
DS_WOUT0 BCF      PORTA,2
DS_WOUT1 BSF      PORTA,1      ; SCLK=1
        RRF      DS_WDATA,F
        DECFSZ   DS_CNT
        GOTO     DS_WLP
        BCF      PORTA,1      ; SCLK=0
        BCF      PORTA,3      ; RST=0
        BSF      STATUS,RP0 ; RAM BANK 1 선택
        BSF      TRISA, 2      ; RA2 IS INPUT
        BCF      STATUS,RP0 ; RAM BANK 1 선택
        BSF      INTCON,GIE ; GLOBAL INTERRUPT ENABLE BIT
        RETURN

```

.....

;; 사용 예

;; 분값을 DS1302에 저장함

```

SAVE_A1_MIN      CALL      DS_WENAB ; DS1302 write enable
                                   ; control out

        MOVF      A1_MIN,W      ; 분값을 저장함
        MOVWF     DS_WDATA
        SWAPF     A1_MIN10,W
        IORWF     DS_WDATA,F
        MOVLW     A1_MIN_WR_CMD
        CALL      DS_WRITE
        CALL      DS_WDISAB ; DS1302 write disable
                                   ; control out

        RETURN

```

.....

10. Display 부 프로그램

DISP ; 전부 OFF

```

        MOVLW          0XFF
        MOVWF          PORTB
; 표시 위치 확인 -- DG_CNT
        MOVF           DG_CNT,W
        ANDLW          07
        ADDWF          PCL,F
        GOTO           DISP_DG0    ; LED
        GOTO           DISP_DG1    ; MSD
        GOTO           DISP_DG5    ; BUZZER
        GOTO           DISP_DG2
        GOTO           DISP_DG3
        GOTO           DISP_DG5    ; BUZZER
        GOTO           DISP_DG4    ; LSD
        GOTO           DISP_DG5    ; BUZZER

```

11. 현재값 선택 표시 부 프로그램

SEL_DISP

; 현재값 표시모드

```

        MOVF           DISP_MODE, W
        ANDLW          07H
        ADDWF          PCL, F
        GOTO           X_MIN        ; 시간+분 HHMM.
        GOTO           X_DATE       ; 월+일   MM.DD
        GOTO           X_YEAR       ; 년     YYYY
        GOTO           A_MIN1       ; ALARM 1 시간+분 HHMM.
        GOTO           A_MIN2       ; ALARM 2 시간+분 HHMM.
        GOTO           X_SEC        ; 초     ?? SS
        CLRF           DISP_MODE
        CLRF           DISP_MODE

```

RETURN

;; 예

.....

X_SEC ; 초를 표시함 ??SS

MOVF SEC, W

MOVWF DG4

MOVF SEC10, W

MOVWF DG3

MOVLW 0X0F

MOVWF DG2

MOVLW 0X0F

MOVWF DG1

; LED2 OFF SET

MOVLW .6 ; LED2 OFF

MOVWF DG0

RETURN

.....

X_MIN ; 시간 분을 표시함 HHMM

CLRF LOOP_TIME ; 시간 증가 확인

CLRF LOOP_TIME1 ; 시간 증가 확인

MOVF MIN, W

MOVWF DG4

; 초 시계 표시용 DOT ON/OFF

BTFSS INT_CNT, SEC_TIME ; DIGIT ON/OFF TIME

GOTO X_MIN_ON

BCF DG4,4

GOTO X_MIN_XX

X_MIN_ON

BSF DG4,4

X_MIN_XX

```
    MOVF      MIN10, W
    MOVWF     DG3
    MOVF      HOUR, W
    MOVWF     DG2
    MOVF      HOUR10, W
    MOVWF     DG1
    ; LED2 OFF SET
    MOVLW     .6           ; LED2 OFF
    MOVWF     DG0
```

RETURN

.....

X_DATE ; 월 일을 표시함 MMDD

```
    MOVF      DATE, W
    MOVWF     DG4
    MOVF      DATE10, W
    MOVWF     DG3
    MOVF      MONTH, W
    MOVWF     DG2
    ; DOT ON SET
    BSF       DG2,4
    MOVF      MONTH10, W
    MOVWF     DG1
    ; LED2 OFF SET
    MOVLW     .6           ; LED2 OFF
    MOVWF     DG0
```

RETURN

.....

X_YEAR ; 년도를 표시함 20YY

```
    MOVF      YEAR, W
    MOVWF     DG4
```

```

MOVF      YEAR10, W
MOVWF     DG3
MOVLW     .0
MOVWF     DG2
MOVLW     .2
MOVWF     DG1
; LED2 OFF SET
MOVLW     .6           ; LED2 OFF
MOVWF     DG0

```

```

RETURN

```

```

.....
.....

```
