



Embedded System Software

(Programming on an embedded system 2)

<http://jcnet.co.kr/> 의 자료실의 강의노트를 참조함

Fall, 2017



Outline

- 개관
- 실습 키트와 개발 환경
- LED
- FND
- 스위치 (인터럽트)
- 버저 (Buzzer)
- 광센서
- 온도 센서

EX1 – Switch

```
#include <avr/io.h>
int main(void)
{
    DDRA = 0xff;
    DDRE = 0x00;
    while (1)
    {
        if (((PINE & 0x10) == 0x00))
            PORTA = 0xff;
        else
            PORTA = 0x00;
    }
}
```

Port INput register E

// SW1 = PE bit4
// LED = 'ON'

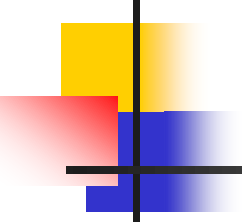
// LED = 'OFF'

EX2 – Switch(Interrupt)

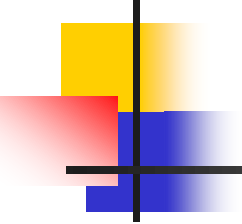
■ 예제 프로그램 개요

- SW1을 누를 때마다 FND의 숫자가 0부터 시작하여 1씩 증가하기 (인터럽트로 처리)

```
#include <avr/io.h>                // ATmega128 register 정의
#include <avr/interrupt.h>          // interrupt 관련
#define F_CPU 16000000UL
#include <util/delay.h>
unsigned char digit[10] = {0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7c, 0x07,
0x7f, 0x67};
unsigned char fnd_sel[4] = {0x01, 0x02, 0x04, 0x08};
volatile int count = 0;            // 전역변수(Global Variable)
ISR(INT4_vect)
{
    count++;
    _delay_ms(10);                // debouncing
}
```



```
void display_fnd(int count)
{
    int i, fnd[4];
    fnd[3] = (count/1000)%10;           // 천 자리
    fnd[2] = (count/100)%10;           // 백 자리
    fnd[1] = (count/10)%10;            // 십 자리
    fnd[0] = count%10;                 // 일 자리
    for (i=0; i<4; i++)
    {
        PORTC = digit[fnd[i]];
        PORTG = fnd_sel[i];
        _delay_ms(2);
    }
}
```



```
int main()
{
    DDRC = 0xff;           // C 포트는 FND 데이터 신호
    DDRG = 0x0f;           // G 포트는 FND 선택 신호
    DDRE = 0xef;           // 0b11101111, PE4(switch1)는 입력
    EICRB = 0x02;          // INT4 = falling edge
    EIMSK = 0x10;          // INT4 interrupt enable
    SREG |= 1<<7;          // SREG bit7 = I (Interrupt Enable)
    while (1)
        display_fnd(count); // FND Display
}
```

sei()와
동일



EX3 – Buzzer

```
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>
int main()
{
    DDRB = 0x10;           // 포트 B의 bit4 를 출력 상태로 세팅
    while(1)               // 500 Hz로 동작
    {
        PORTB = 0x10;      // 1ms 동안 'On' 상태 유지
        _delay_ms(1);
        PORTB = 0x00;      // 1ms 동안 'Off' 상태 유지
        _delay_ms(1);
    }
}
```

EX4 – Buzzer(Prescaler)

```
#include <avr/io.h>
#include <avr/interrupt.h>
#define ON 1
#define OFF 0
#define DO_data 17
volatile int state = OFF;

ISR(TIMERO0_OVF_vect)
{
    if (state == ON)
    {
        PORTB = 0x00;
        state = OFF;
    }
}

else
{
    PORTB = 0x10;
    state = ON;
    TCNT0 = DO_data;
}

int main()
{
    DDRB = 0x10;
    TCCR0 = 0x03;
    TIMSK = 0x01;
    TCNT0 = DO_data;
    sei(); // 전역 인터럽트
    while(1);
}
```

32분주

/ Overflow



Assignments

- I-Class에 다음 슬라이드의 과제 제출
- 제출 파일
 - 소스 코드(1 mandatory + 2 optional)
 - 보고서(.pdf)
- 파일명
 - 11주차_학번_이름(압축하여 하나의 파일로 제출)
- 제출 기한
 - 다음주 화요일 자정까지



Implement 1 (Mandatory)

- Switch와 Buzzer 사용하여 도레미파솔라시도 8가지의 소리 내기
- 두개의 스위치는 인터럽트 사용
 - SW1: 음 변경, SW2: 버저의 ON/OFF 담당
- 버저는 프리스케일러 사용

- 참고
 - 두 번째 스위치 ISR 등록하기
 - ISR(INT5_vect){ ... }
 - DDRE의 5번째 비트 클리어 (interrupt 입력)
 - EICRB의 4번째 비트 세트 (falling edge)
 - EIMSK의 5번째 비트 세트 (interrupt enable)
 - 프리스케일러가 32분주일 때,
 - 도레미...도: 17, 43, 66, 77, 97, 114, 117, 137



Implement 1 (Mandatory)

- Using switches and buzzer, make sound of Do Re Mi Fa Sol La Si Do (CDEFGABC)
- Implement two switches with interrupt
 - SW1: Changes scale, SW2: Turn on/off the buzzer
- Implement the buzzer with prescaler

- Hint
 - Register SW2 ISR
 - `ISR(INT5_vect){ ... }`
 - Clear 5th bit of DDRE (interrupt input)
 - Set 4th bit of EICRB (falling edge)
 - Set 5th bit of EIMSK (interrupt enable)
 - When prescaler = 32,
 - Do Re Mi ... Do(CDE...C): 17, 43, 66, 77, 97, 114, 117, 137



Implement 2 (Optional)

- Switch와 Buzzer, FND 사용하여 1/100초단위 스탑워치 구현하기
- 두개의 스위치는 인터럽트 사용
 - SW1: 시작, 일시정지 / SW2: 시간 기록, 일시정지 시 초기화
- 버저는 각 기능에 따라 다른 소리 내기



Implement 2 (Optional)

- Implement 1/100 seconds stopwatch using switches, buzzer and FND
- Implement two switches with interrupt
 - SW1: start and stop / SW2: recording the time and clear when it is stopped
- Make sound differently according to the functions

- 실습 후 Erase device 해주세요.
- Before you go out, please click “Erase device” button.

