

## 최범수 10일차 과제

### 1. HW\_001

스위치가 연결되지 않은 상태에서는 전류가 흐르는지 흐르지 않는지 알 수 없는 데 이 상태를 플로팅 현상이라고 한다. 즉, 아무것도 연결되지 않아서 전압을 모르는 상태, 플로팅 상태가 문제가 되는 이유로는 주위의 정전기나 잡음에 의해 오류가 발생할 수 있다는 점이 있다. 이러한 문제점을 해결하기 위해서 풀업 저항이나 풀 다운 저항을 회로에 연결한다. 풀업 저항은 켜져 있을 때 전압이 들어가지 않고 꺼져 있을 때 들어가는 반면 풀다운 저항은 반대로 켜져 있을 때 전압이 들어가고 꺼져 있을 때 들어가지 않는 방식으로 작동한다.

## 2. HW\_002

```
/*
 * test.c
 *
 * Created: 2024-07-26 오전 2:26:49
 * Author : chlqj
 */
#define F_CPU 16000000

#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>

volatile uint8_t led = 0x01; //첫 번째 LED 켜지게

int main(void)
{
    DDRA = 0xFF; // 포트 A를 출력으로
    DDRD = 0x00; // 포트 D를 입력으로

    PORTA = 0xFF; //(LED 꺼짐)
    PORTD = 0xFF;

    //인터럽트
    EIMSK = (1 << INT2) | (1 << INT3);
    //인터럽트
    EICRA = (1 << ISC21) | (0 << ISC20) | (1 << ISC31) | (0 << ISC30);

    sei();

    /* Replace with your application code */
    while (1)
    {
        if((PIND & 0x01) == 0 && (PIND & 0x02) == 0)
        {
            PORTA = 0x00; // LED 켜기
        }
        else
        {
            if ((PIND & 0x01) == 0) // 스위치 1이 눌렸는지 확인
            {
                PORTA = 0xF0; // 4~7번 LED 켜기
            }
            else if ((PIND & 0x02) == 0) // 스위치 2가 눌렸는지 확인
            {
                PORTA = 0x0F; // 0~3번 LED 켜기
            }
            else
            {
                PORTA = 0xFF; //(LED 꺼짐)
                _delay_ms(500);
                PORTA = 0x00; // (LED 켜짐)
                _delay_ms(500);
            }
        }
    }
}
```

```

    }
    }

}

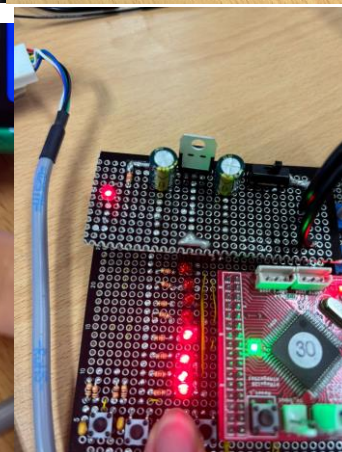
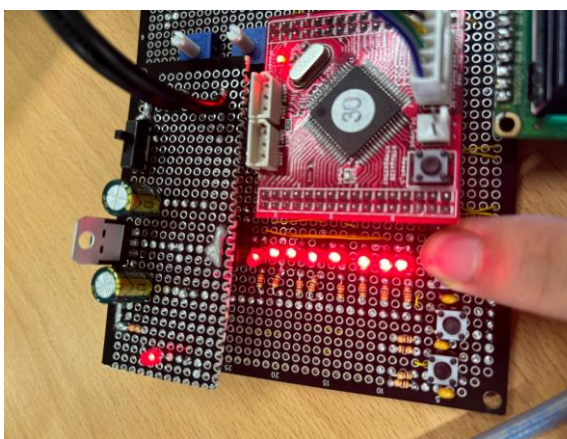
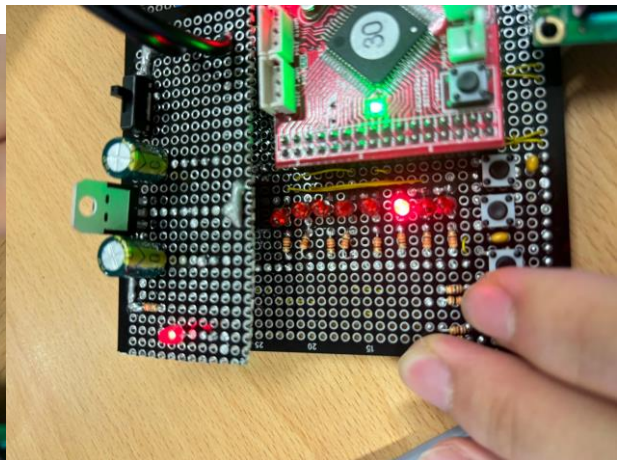
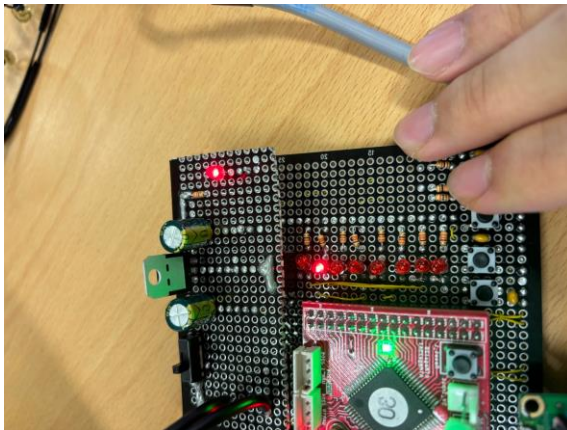
return 0;

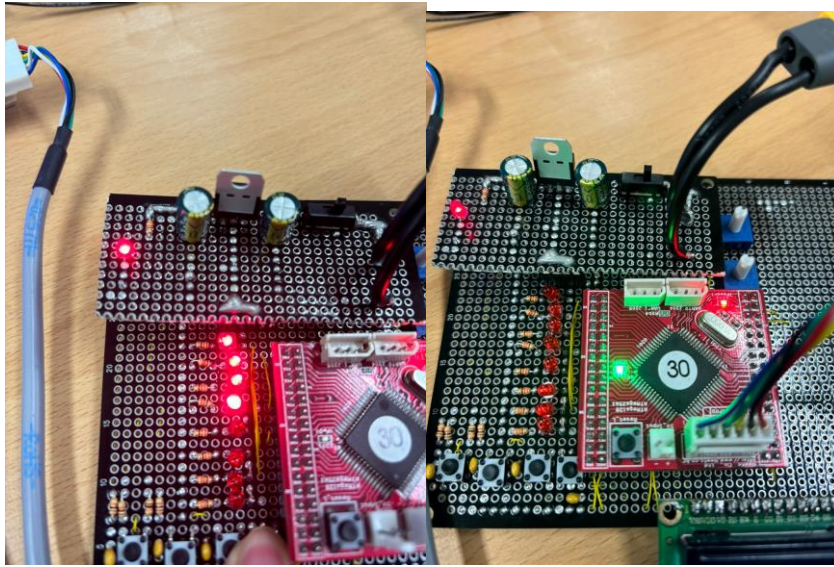
}

ISR(INT2_vect)
{
    // INT2 발생 시 LED 좌측 이동
    if (led & 0x80) {
        led = 0x01;
    } else {
        led <<= 1;
    }
    PORTA = ~led;
}

ISR(INT3_vect)
{
    // INT3 발생 시 LED 우측 이동
    if (led & 0x01) {
        led = 0x80;
    } else {
        led >>= 1;
    }
    PORTA = ~led;
}

```





### 3. HW\_003

```
/*  
  
 * test5.c  
  
 *  
  
 * Created: 2024-07-28 오전 4:25:21  
  
 * Author : chlqj  
  
 */  
  
#define F_CPU 16000000UL  
  
#include <avr/io.h>  
  
#include <util/delay.h>  
  
#include <avr/interrupt.h>  
  
void pattern1() {  
    uint8_t patterns[] = {248, 241, 227, 199, 143, 31, 62, 124};  
  
    for (int i = 0; i < 2; i++) {  
        for (int j = 0; j < 7; j++) {  
            PORTA = patterns[j];  
            _delay_ms(100);  
        }  
    }  
}
```

```
}
```

```
void pattern2() {
```

```
    uint8_t patterns[] = {31, 143, 199, 227, 241, 248, 124, 62};
```

```
    for (int i = 0; i < 2; i++) {
```

```
        for (int j = 0; j < 8; j++) {
```

```
            PORTA = patterns[j];
```

```
            _delay_ms(100);
```

```
        }
```

```
    }
```

```
}
```

```
void pattern3() {
```

```
    int result_first = 1;
```

```
    for(int i = 0; i < 8; i++) {
```

```
        PORTA = ~result_first;
```

```
        _delay_ms(100);
```

```
        result_first *= 2;
```

```
    }
```

```
    int result_second = 128;
```

```
    for(int i = 0; i < 8; i++) {
```

```
        PORTA = ~result_second;

        _delay_ms(100);

        result_second /= 2;
    }
}
```

```
ISR(INT0_vect) {
    pattern1();
}
```

```
ISR(INT1_vect) {
    pattern2();
}
```

```
ISR(INT2_vect) {
    pattern3();
}
```

```
ISR(INT3_vect) {
}
```

```
void make_Interrupts() {
```

```

EIMSK |= (1 << INT0) | (1 << INT1) | (1 << INT2) | (1 << INT3);

EICRA |= (1 << ISC01) | (1 << ISC11) | (1 << ISC21) | (1 << ISC31);

}

```

```

int main(void) {

    DDRA = 0xFF;

    DDRD = 0x00;


    make_Interrupts();

    sei();


    int x = 255;

    while (1) {

        PORTA = x;

        _delay_ms(100);

        x--;

    }

}

```



