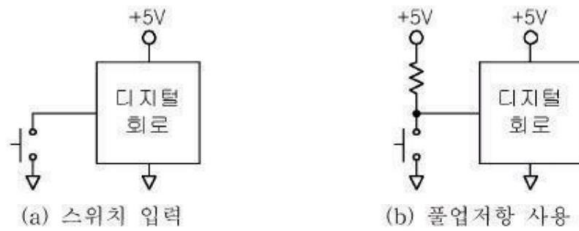
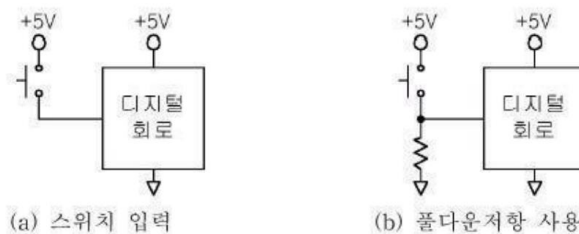


과제 1

플로팅 현상이란: 풀업 스위치 방식에서 스위치가 닫힌 상태에서는 5v가 흐르면서 high 인 것을 알게 된다. 풀다운 스위치 방식에서는 반대로 스위치가 닫힌 상태에서 0v가 흐르면서 low인 것을 알게된다. 하지만 스위치가 열린 상태에서는 두 상황 모두 스위치가 열린건지 0v가 흐르는 low상태인지 구분할수 없다. 이상태를 플로팅 상태라고 한다.



<그림 1> L 스위치 입력과 풀업 저항



<그림 2> H 스위치 입력과 풀다운 저항

풀업 저항, 풀다운 저항으로 각각 전원부, 접지부에 저항으로 연결해 놓으면 전기는 저항 이 낮은곳으로 흐르는 특성을 사용해서 플로팅 문제를 해결할수 있다.

풀업 방식에서 스위치가 열려있을때는 전원부에서 흐르는 5v가 핀으로 흘러 high상태가 된다. 스위치를 닫으면 5v가 접지부로 흘러 스위치에는 0v가 흘러 low가 된다

풀다운 방식에서는 스위치를 열었을 때 0v, 스위치를 닫았을 때 5v가 흐른다

과제 2 소스코드

```
#define F_CPU 16000000

#include <avr/io.h>

#include <util/delay.h>

#include <avr/interrupt.h>
```

```

int main()
{
    DDRA=0xFF;
    DDRD=0x00;

    EIMSK=(1 << INT2)|(1 << INT3);
    EICRA=(ISC21 >> 1)|(ISC20 >> 1)|(ISC31 >> 1)|(ISC30 >> 1);

    sei();

    while (1)
    {
        if((PIND & 0x01)==0)
        {
            PORTA=0x0F;
            if((PIND & 0x02)==0)
            {
                PORTA=0x00;
            }
        }
        else if((PIND & 0x02)==0)

```

```

        {
            PORTA=0xF0;

        }
        else
        {
            PORTA=0xFF;

            _delay_ms(500);

            PORTA=0x00;

            _delay_ms(500);

        }
    }
}

```

ISR(INT2_vect)

```

{
    PORTA= 0xFF;

    unsigned char left=0x80;

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

        _delay_ms(500);

        left=left>>1;

    }
}

```

```
}
```

```
ISR(INT3_vect)
```

```
{
```

```
    PORTA= 0xFF;
```

```
    unsigned char right=0x01;
```

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

```
    {
```

```
        PORTA= right;
```

```
        _delay_ms(500);
```

```
        right=right<<1;
```

```
    }
```

```
}
```

과제 3 소스코드

```
#define F_CPU 16000000
```

```
#include <avr/io.h>
```

```
#include <util/delay.h>
```

```
#include <avr/interrupt.h>
```

```
char counter = 0xff;
```

```

int main()
{
    DDRA = 0xff;

    EIMSK = (1 << INT0) | (1 << INT1) | (1 << INT2) | (1 << INT3);
    EICRA = (1 << ISC01) | (1 << ISC00) | (1 << ISC11) | (1 << ISC10) | (1 << ISC21) | (1
<< ISC20) | (1 << ISC31) | (1 << ISC30);
    sei();

    while(1)
    {
        counter = counter - 0x01;
        _delay_ms(100);
        PORTA = counter;
    }
}

ISR(INT0_vect)
{
    char buff = 0x1f;
    for (int j = 0; j < 2; j++)
    {
        for (int i = 0; i < 7; i++)

```

```

        {

            PORTA = buff;

            _delay_ms(100);

            buff <<= 1;

        }

    }

}

```

ISR(INT1_vect)

```

{

    char buff = 0xf8;

    for (int j = 0; j < 2; j++)

    {

        for (int i = 0; i < 7; i++)

        {

            PORTA = buff;

            _delay_ms(100);

            buff >>= 1;

        }

    }

}

```

ISR(INT2_vect)

```
{  
  
    char buff = 0xfe;  
  
    for (int i = 0; i < 7; i++)  
    {  
  
        PORTA = buff;  
  
        _delay_ms(100);  
  
        buff >>= 1;  
  
    }  
  
    for (int i = 0; i < 7; i++)  
    {  
  
        buff <<= 1;  
  
        PORTA = buff;  
  
        _delay_ms(100);  
  
    }  
  
}
```

ISR(INT3_vect)

```
{  
  
    counter = 0xff;  
  
}
```