

4주차 실험 결과 보고서

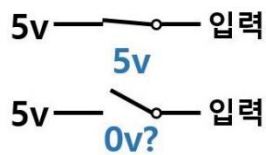
2조 (이지수, 박은재, 유승훈, 고상현, 강민수)

1. 실험 목표

- 1) 스케터 파일의 이해 및 플래시 프로그래밍
- 2) 릴레이 모듈의 이해 및 임베디드 펌웨어를 통한 동작
- 3) 센싱에서 풀링 방식의 이해

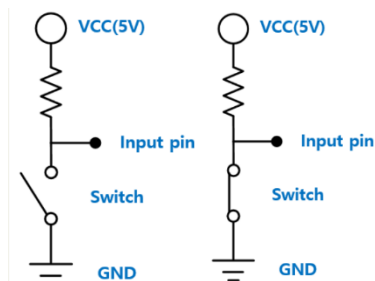
2. 이론적 배경

1) Floating



- 전압을 High / Low 로 보기 힘든 상태
- 아주 작은 노이즈만으로도 High와 low 사이를 빠르게 이동하여 오동작 유발
- 따라서 풀업 저항 또는 풀다운 저항을 사용

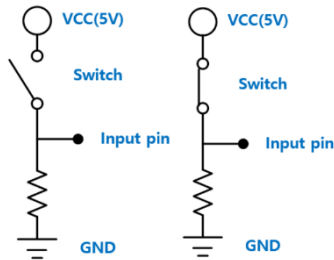
2) Pull up



- VCC에 저항을 연결하는 방법
- 스위치 OFF 시 input에는 High 신호

- 스위치 ON 시 input에는 Low 신호

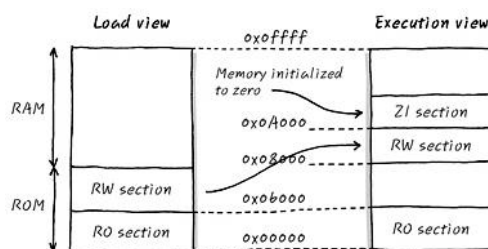
3) Pull down



- GND에 저항을 연결하는 방법
- 스위치 OFF 시 input에는 Low 신호
- 스위치 ON 시 input에는 High 신호

4) Scatter File

- 실행시킬 바이너리 이미지가 메모리에 로드될 때, 바이너리 이미지의 어떤 영역이 어느 주소에 어느 크기만큼 배치되어야 할지 작성한 파일이다.
- 바이너리의 여러 부분을 각각 별개의 메모리 영역에 로드해야 될 때 유용하다.
- 자주 사용되거나 빠른 실행을 요구하는 코드 영역을 접근 시간이 빠른 메모리에 우선 배치하도록 설정할 수 있다.
- IAR EW는 .icf 파일을 스캐터 파일로 이용



- Load view : flash에 실행 image가 담겨 있을 때의 형태
- Execution view : flash에 실행 image가 실행될 때의 형태

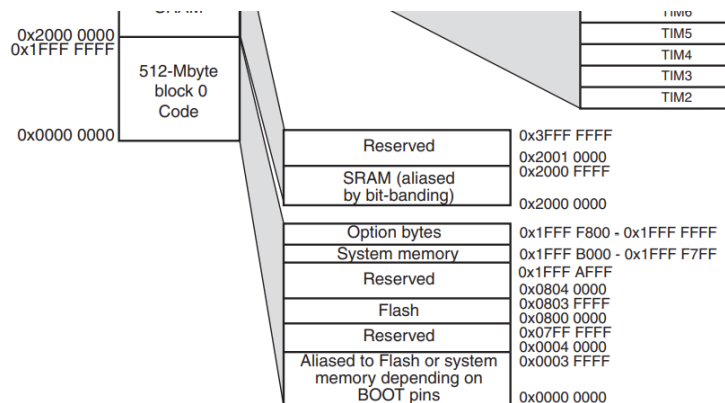
5) Relay Module



- 릴레이를 제어하는 모듈
- 전자기 유도 원리를 이용하여 스위치 역할을 한다.
- 릴레이에 신호를 가하면 출력 상태(ON/OFF)가 변경된다.

3. 실험 과정

1) Scatter File 작성하고 업로드하기



프로젝트를 생성한 후 Scatter File을 수정하여 원하는 만큼 메모리 영역을 할당한다. RAM과 ROM의 시작 주소를 확인해보면 각각 0x2000 0000, 0x0800 0000이다.

```
/*-Memory Regions-*/
define symbol __ICFEDIT_region_ROM_start__ = 0x08000000;
define symbol __ICFEDIT_region_ROM_end__   = 0x08080000;
define symbol __ICFEDIT_region_RAM_start__  = 0x20000000;
define symbol __ICFEDIT_region_RAM_end__    = 0x20008000;
```

RAM에는 0x8000, ROM에는 0x80000 만큼의 메모리를 할당하기 위해 각각의 시작 주소에 크기를 더해준다. 위와 같이 작성한 .icf 파일을 업로드한다.

2) 회로 구성하기



릴레이 모듈의 GND와 VCC를 보드의 GND와 VCC(3.3V)에 연결해주고 Input 핀으로 PD2, PD3을 선택했다.

3) 코드 작성하기

초기화하는 코드는 3주차와 동일하므로 설명은 생략한다. 4주차 실험은 KEY4를 사용하지 않기 때문에 3주차 코드에서 GPIOA 관련 코드는 지웠다.

```
if (~GPIOC_IDR & (0x1 << 4)) { // KEY1

    GPIOD_BSRR |= 0x04; // PD2 ON
    GPIOD_BSRR |= 0x08; // PD3 ON

    delay(10000000);

    GPIOD_BRR |= 0x04; // PD2 OFF
    GPIOD_BRR |= 0x08; // PD3 OFF
}
```

KEY1(PC4)을 눌렀을 때 릴레이 모듈1(PD2), 릴레이 모듈2(PD3)와 연결된 pin을 set 한 뒤 delay() 함수를 호출하고 릴레이 모듈1, 릴레이 모듈2와 연결된 pin을 reset 했다.

```
if (~GPIOB_IDR & (0x1 << 10)) { // KEY2
    GPIOD_BSRR |= 0x04; // PD2 ON

    delay(10000000);

    GPIOD_BRR |= 0x04; // PD2 OFF
}
```

KEY2(PB10)를 눌렀을 때 릴레이 모듈1(PD2)과 연결된 pin을 set 한 뒤 delay() 함수를 호출하고 릴레이 모듈1과 연결된 pin을 reset 했다.

```

if (~GPIOC_IDR & (0x1 << 13)) { // KEY3
    GPIOD_BSRR |= 0x08; // PD3 ON

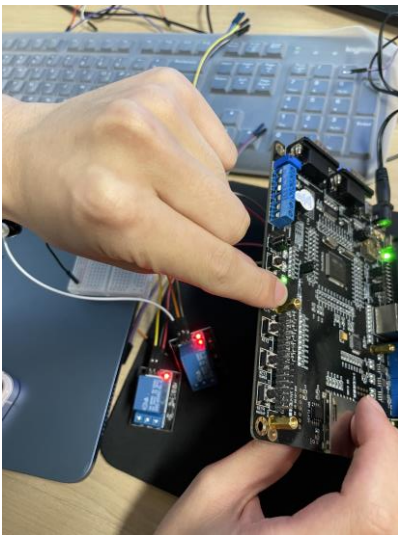
    delay(10000000);

    GPIOD_BRR |= 0x08; // PD3 OFF
}

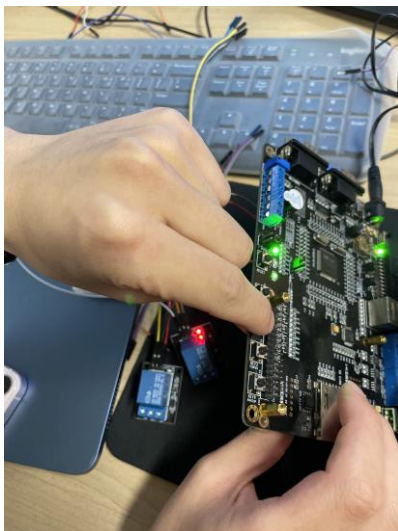
```

KEY3(PC13)을 눌렀을 때 릴레이 모듈2(PD3)와 연결된 pin을 set 한 뒤 delay() 함수를 호출하고 릴레이 모듈2와 연결된 pin을 reset 했다.

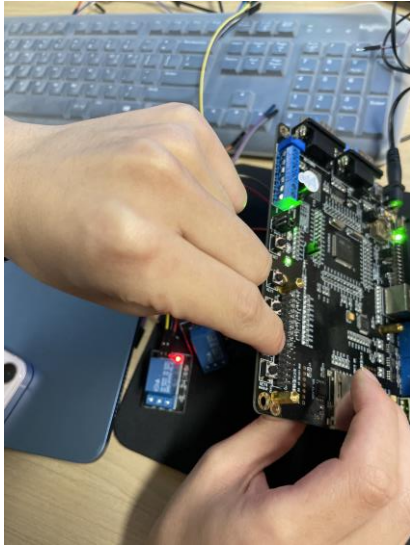
4. 실험 결과



KEY1(PC4)을 누르면 릴레이 모듈1(PD2), 릴레이 모듈2(PD3)에서 딸깍 소리가 난다.



KEY2(PB10)를 누르면 릴레이 모듈1(PD2)에서 딸깍 소리가 난다.



KEY3(PC13)을 누르면 릴레이 모듈2(PD3)에서 딸깍 소리가 난다.